CPAN NETWORK
ON INSTRUMENTATION
AND DETECTORS

Exercise Book

TIDPAN-2026

Workshop on Instrumentation and Detectors for Nuclear, Particle and Astroparticle Physics

Exercise Book

TIDPAN 2026
Worksop on Instrumentation and Detectors
for Nuclear, Particle and Astroparticle Physics

May 21, 2026

Exercises

Exercise 1 Neutron detection with scintillation detectors

1.1 Objectives

- To introduce the students to neutron detection using scintillation detectors with neutron–gamma discrimination capability.
- To understand why neutrons cannot be detected directly through electromagnetic ionisation and must instead be detected through nuclear reactions or scattering processes.
- To compare the response of two scintillation detectors sensitive to different neutron-energy ranges: a CLYC-6 detector and a deuterated-stilbene detector.
- To study pulse-shape discrimination (PSD) as a method to distinguish gamma-ray events from neutron events.
- To perform an energy calibration of both detectors using gamma-ray sources.
- To identify neutron and gamma branches in PSD-ratio versus energy matrices.
- To distinguish fast-neutron events from thermal and epithermal neutron events in CLYC-6.
- To estimate the intrinsic detection efficiencies for thermal neutrons in CLYC-6, fast neutrons in CLYC-6, and fast neutrons in deuterated stilbene.

1.2 Professors

- Jorge Lerendegui Marco
- Sebastián Valladares Sánchez
- Javier Balibrea-Correa

1.3 Theoretical Introduction

Neutrons are neutral particles and therefore do not interact with matter through direct electromagnetic ionisation. For this reason, neutron detection is usually based on nuclear reactions or elastic scattering processes that produce charged particles. These charged particles deposit energy in the detector medium and generate a measurable signal.

In this practicum, neutron detection will be studied using two scintillation detectors with neutron–gamma discrimination capability:

- A CLYC-6 scintillator, sensitive to gamma rays, thermal and epithermal neutrons, and also fast neutrons.
- A deuterated-stilbene scintillator, sensitive to gamma rays and fast neutrons.

The comparison between these detectors illustrates how the active material determines the neutron-energy range, the interaction mechanism, the event topology in PSD–energy plots, and the intrinsic detection efficiency.

Pulse-shape discrimination

Many scintillators emit light with different time profiles depending on the type of particle that deposits energy in the crystal. Gamma rays mainly interact through electrons, while neutrons produce heavier charged particles such as recoil protons, recoil deuterons, alpha particles, tritons, or reaction products. These particles have different ionisation densities and therefore produce scintillation pulses with different decay components.

Pulse-shape discrimination exploits this difference. In the charge-integration method, two quantities are calculated for each digitised pulse:

- A total integral, Q_{long}, obtained by integrating the full pulse over a long time window.
- A partial or tail integral, Q_{tail}, obtained by integrating the delayed part of the pulse.

A PSD parameter can then be defined as

$$PSD = \frac{Q_{\text{tail}}}{Q_{\text{long}}}.$$

Equivalently, if two non-overlapping prompt and delayed windows are used, with charges Q_1 and Q_2, the PSD ratio can be written as

$$PSD = \frac{Q_2}{Q_1 + Q_2}.$$

When the PSD ratio is plotted as a function of deposited energy, gamma-ray and neutron events form different branches. The position and separation of these branches depend on the scintillator material, the particle type, the integration windows, and the deposited energy.

CLYC-6 detector

CLYC is an inorganic scintillator with chemical composition Cs_2LiYCl_6:Ce. The CLYC-6 detector used in this practicum is enriched in ^{6}Li and consists of a $10 \times 50 \times 50$ mm^3 crystal from the GN-Vision detector system, coupled to a Hamamatsu photomultiplier tube.

CLYC-6 can detect gamma rays, thermal and epithermal neutrons, and fast neutrons. Thermal and epithermal neutrons are mainly detected through the reaction

$$^6\text{Li} + n \rightarrow \alpha +^3 \text{H} + Q,$$

with

$$Q \simeq 4.78 \text{ MeV}.$$

The alpha particle and triton deposit their energy locally in the crystal. Due to scintillation quenching, the thermal-neutron peak appears at an electron-equivalent energy of approximately 3.2 MeVee. In a PSD-ratio versus energy plot, this contribution appears as a characteristic neutron bump associated with the $^6\text{Li}(n,\alpha)^3\text{H}$ reaction.

Fast neutrons can also be detected in CLYC through reactions on chlorine, mainly

$$^{35}\mathrm{Cl}(n,p)^{35}\mathrm{S}$$

and

$$^{35}\mathrm{Cl}(n,\alpha)^{32}\mathrm{P}.$$

These reactions produce charged particles whose deposited energy is related to the incident neutron energy. Therefore, CLYC can act as a neutron-sensitive detector over a broad energy range. However, its response to thermal and epithermal neutrons is much more efficient in CLYC-6 because of the high ^{6}Li content.

This practicum uses the same CLYC-6 scintillator exploited in the development of GN-Vision, a compact dual gamma-ray and neutron imaging device. GN-Vision is based on two position-sensitive detection planes. The first plane is a CLYC-6 or CLLBC detector, which provides gamma-ray/neutron discrimination and sensitivity to slow neutrons. This detector is the central element that enables dual-particle operation: it can participate in gamma-ray Compton imaging while also detecting and localising neutrons. This practicum uses the same CLYC-6 scintillator as a laboratory detector to study the underlying detection principles: gamma-ray spectroscopy, neutron–gamma pulse-shape discrimination, and efficient thermal/epithermal neutron detection.

Deuterated-stilbene detector

The second detector used in this practicum is a deuterated-stilbene scintillator, also referred to as stilbene-d_{12}. The crystal has dimensions $25 \times 25 \times 50$ mm^3 and it is coupled to a Hamamatsu R11265U photomultiplier tube.

Deuterated stilbene is an organic solid scintillator designed for fast-neutron detection. Fast neutrons are mainly detected through elastic scattering on deuterium nuclei:

$$n + d \rightarrow n + d.$$

The recoil deuterons produce scintillation light. Since neutron-induced pulses and gamma-induced pulses have different time profiles, deuterated stilbene provides excellent neutron–gamma discrimination through PSD.

Unlike CLYC-6, deuterated stilbene does not have a high-efficiency thermal-neutron capture reaction. Therefore, in this practicum it will be used mainly as a fast-neutron detector. Its response to a ^{252}Cf source will be compared with the fast-neutron response of CLYC-6.

The high-purity stilbene-d_{12} detector used in this practicum belongs to the STAR detector, an array of compact stilbene-d_{12} modules are being developed as a new generation of total-energy detectors for neutron-capture time-of-flight measurements at CERN n_TOF. This material is promising alternative to conventional liquid C_6D_6 scintillators: it is non-liquid, non-hygroscopic, has lower chemical handling risks, allows thinner encapsulation, can be implemented in compact geometries, and can be more easily coupled to modern photosensors. In addition, its large light output, fast timing response, neutron–gamma PSD capability, and good recovery after intense prompt gamma flashes make it attractive for experiments performed in high-flux neutron fields.

The ^{252}Cf neutron source

The neutron source used in this practicum is ^{252}Cf. This isotope emits neutrons mainly through spontaneous fission. Each spontaneous-fission event emits several prompt neutrons. The average prompt neutron multiplicity is approximately

$$\bar{\nu} \simeq 3.76 \text{ neutrons/fission}.$$

Only a fraction of the ^{252}Cf decays occur by spontaneous fission. The spontaneous-fission branching ratio is approximately

$$B_{\mathrm{SF}} \simeq 3.1\%.$$

Therefore, for a source activity A, the total neutron emission rate can be estimated as

$$R_n = A\, B_{\mathrm{SF}}\, \bar{\nu}.$$

This neutron emission rate will be used to estimate the intrinsic detection efficiency of the detectors. The value of the source activity used in the calculation must be corrected to the date of the experiment using the half-life of ^{252}Cf and the information from the source certificate.

1.4 Experimental Setup

The experimental setup consists of two scintillation detectors, radioactive sources for calibration and neutron measurements, a digitiser, acquisition software, and ROOT-based analysis scripts.

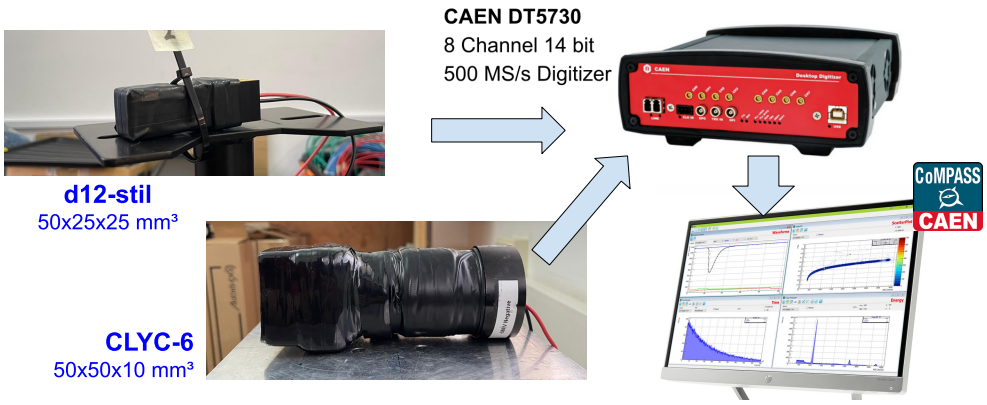


Figure 1: Scintillation detectors with n/g discrimination capabilities, CAEN digitizer and data acquisition system used in the lab session.

Detectors

- **CLYC-6 detector:** $10 \times 50 \times 50$ mm^3 Cs_2LiYCl_6:Ce crystal enriched in ^{6}Li, from the GN-Vision detector, coupled to a Hamamatsu photomultiplier tube. This detector is sensitive to gamma rays, thermal and epithermal neutrons, and fast neutrons.

- **Deuterated-stilbene detector:** $25 \times 25 \times 50$ mm^3 stilbene-d_{12} crystal from the STAR array, coupled to a Hamamatsu R11265U photomultiplier tube. This detector is mainly sensitive to fast neutrons.

Radioactive sources

- ^{137}Cs gamma-ray source.
- ^{207}Bi gamma-ray source.
- ^{252}Cf neutron source.

Electronics and acquisition system

- CAEN DT5730 digitiser.
- Laptop with CoMPASS acquisition software.
- ROOT analysis scripts for energy calibration, PSD analysis, neutron selection, and efficiency estimation.

The CAEN DT5730 is used to digitise the photomultiplier signals. The digitised waveforms are processed to obtain the pulse integrals required for energy estimation and PSD analysis. The acquisition software records event-by-event information, including timestamps, pulse integrals, and other quantities needed for the offline analysis.

1.5 Procedure

Detailed steps to perform the exercise are given below.

1. **Initial setup and signal inspection**
   (a) Identify the two detectors: CLYC-6 and deuterated stilbene.
   (b) Check the optical coupling and PMT connections.
   (c) Connect the PMT outputs to the CAEN DT5730 digitiser.
   (d) Open the CoMPASS acquisition software on the laptop.
   (e) Verify the digitiser settings: sampling, trigger threshold, record length, integration gates, and acquisition mode.
   (f) Display example waveforms from each detector.
   (g) Adjust the acquisition parameters so that the full scintillation pulse is recorded and the baseline is stable.
2. **Energy calibration with gamma-ray sources**
   (a) Place the ^{137}Cs source in front of each detector.
   (b) Acquire a gamma-ray spectrum with the CLYC-6 detector.
   (c) Identify the 662 keV photopeak of ^{137}Cs in the CLYC-6 spectrum.
   (d) Acquire a gamma-ray spectrum with the deuterated-stilbene detector.

(e) Identify the Compton edge associated with the 662 keV gamma ray in the deuterated-stilbene spectrum.

(f) Repeat the measurement using the ^{207}Bi source.

(g) For CLYC-6, use the visible photopeaks to build an energy calibration in electron-equivalent energy.

(h) For deuterated stilbene, use the Compton edges to build an electron-equivalent energy calibration.

(i) Fit the calibration points with a linear calibration function:

$$E = a + bQ,$$

where Q is the total pulse integral.

(j) Save the calibration parameters for later use in the neutron measurements.

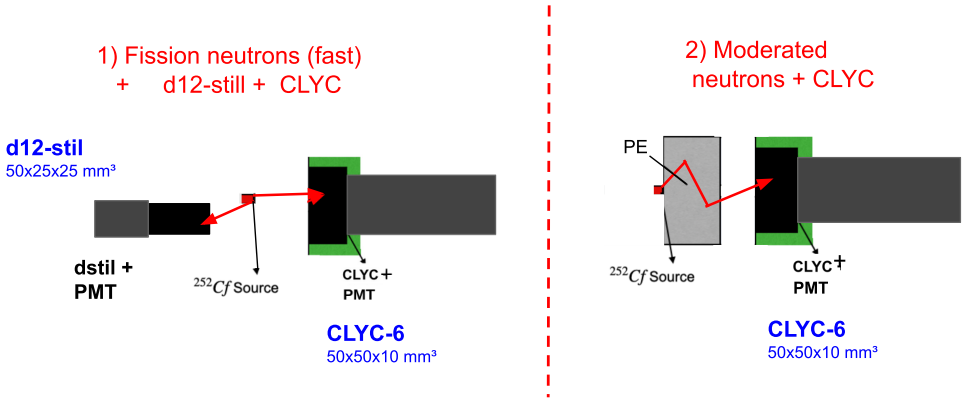


Figure 2: Experimental configurations with the 252Cf neutron source. Naked source for fast neutron detection with CLYC6 and d12-still detectors (left) and moderated 252Cf source for thermal neutron detection with the CLYC6 detector (right).

3. **Fast-neutron detection with a naked ^{252}Cf source**

(a) Place the naked ^{252}Cf source at a distance of 5 cm from the deuterated-stilbene detector.

(b) Acquire data for a fixed time interval.

(c) Repeat the measurement with the CLYC-6 detector placed at 5 cm from the source.

(d) For each detector, build the PSD-ratio versus energy matrix.

(e) Identify the gamma-ray branch and the neutron branch.

(f) Compare the fast-neutron response of deuterated stilbene and CLYC-6.

4. **Thermal and epithermal neutron detection with CLYC-6**

(a) Place the CLYC-6 detector at 5 cm from the ^{252}Cf source.

(b) Insert a 5 cm thick polyethylene slab between the source and the detector.

(c) Acquire data for the same or a comparable time interval as in the naked-source measurement.

(d) Build the PSD-ratio versus energy matrix.

(e) Identify the gamma-ray branch, the fast-neutron branch, and the thermal/epithermal neutron bump.

(f) Locate the thermal/epithermal neutron bump associated with $^{6}\mathrm{Li}(n,\alpha)^{3}\mathrm{H}$ reaction, expected around 3.2 MeV.

(g) Compare the moderated and non-moderated configurations.

5. **PSD analysis**

(a) For each event, calculate the total charge integral Q_{long}.

(b) Calculate the delayed or tail charge integral Q_{tail}.

(c) Compute the PSD ratio:

$$PSD = \frac{Q_{\mathrm{tail}}}{Q_{\mathrm{long}}}.$$

(d) Plot PSD as a function of calibrated energy for each detector and each measurement.

(e) Define graphical or numerical PSD cuts to select gamma rays and neutrons.

(f) For deuterated stilbene, select the fast-neutron branch.

(g) For CLYC-6, select:

- all neutron events,
- fast-neutron events,
- thermal/epithermal neutron events around the $^{6}\mathrm{Li}$ capture bump.

(h) Save the selected spectra and the number of events passing each cut.

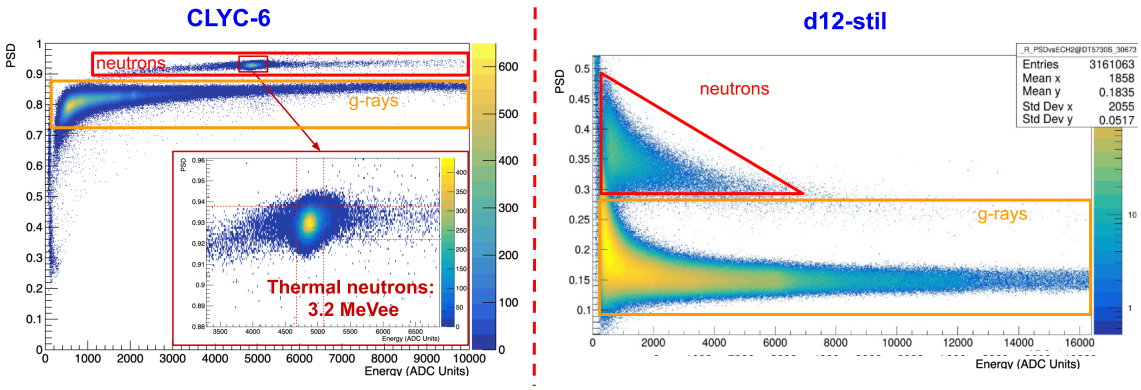


Figure 3: PSD ratio as a function of the electron equivalent energy (ADC channel) obtained for CLYC-6 in an irradiation with a partially moderated 252Cf source (right) and for the d12-stil in an irradiation with a naked 252Cf source (right). The gamma-ray and neutron branches have been highlighted.

6. **PSD and energy cuts**

(a) Implement the PSD and energy cuts using the provided ROOT scripts.

(b) Apply the cuts to select fast neutrons in deuterated stilbene.

(c) Apply the cuts to select all neutron events in CLYC-6.

(d) Apply a dedicated cut to isolate the thermal/epithermal neutron bump in CLYC-6.

(e) Compare the neutron selection obtained with and without the polyethylene moderator.

7. **Intrinsic efficiency estimation**

(a) Determine the number of selected neutron events:

$$N_{\mathrm{det}}.$$

(b) Subtract the background contribution, if a background run is available:

$$N_{\mathrm{net}} = N_{\mathrm{source}} - N_{\mathrm{background}}.$$

(c) Correct for the acquisition live time t_{live}:

$$R_{\mathrm{det}} = \frac{N_{\mathrm{net}}}{t_{\mathrm{live}}}.$$

(d) Calculate the neutron emission rate of the ^{252}Cf source:

$$R_n = A\, B_{\mathrm{SF}}\, \bar{\nu},$$

where A is the source activity, B_{SF} is the spontaneous-fission branching ratio, and $\bar{\nu}$ is the average neutron multiplicity per fission.

(e) Estimate the solid angle subtended by each detector. For a rectangular detector of half-widths a and b placed at distance d from the source, the solid angle can be approximated by:

$$\Omega = 4 \arctan\left(\frac{ab}{d\sqrt{d^2 + a^2 + b^2}}\right).$$

(f) Calculate the geometrical efficiency:

$$\varepsilon_\Omega = \frac{\Omega}{4\pi}.$$

(g) Estimate the number of neutrons emitted towards the detector:

$$R_{n,\Omega} = R_n\, \varepsilon_\Omega.$$

(h) Calculate the intrinsic detection efficiency:

$$\varepsilon_{\mathrm{int}} = \frac{R_{\mathrm{det}}}{R_{n,\Omega}}.$$

(i) Repeat the calculation for:

- thermal/epithermal neutrons detected with CLYC-6 in the moderated configuration,
- fast neutrons detected with CLYC-6 in the naked-source configuration,
- fast neutrons detected with deuterated stilbene in the naked-source configuration.

1.6 Analysis and interpretation of the results

The analysis should focus on comparing the physical response of the two detectors and understanding the role of PSD, energy cuts, moderation, and detector geometry.

1. **Energy calibration**

   (a) Obtain the calibration function for CLYC-6 using photopeaks.

   (b) Obtain the calibration function for deuterated stilbene using Compton edges.

   (c) Compare the calibration procedure for an inorganic scintillator with photopeak response and an organic scintillator dominated by Compton continua.

   .

2. **PSD-ratio versus energy plots**

   For each detector, present the PSD-ratio versus energy matrix. Identify:

   - the gamma-ray branch,
   - the neutron branch,
   - the region used for neutron selection.

   Discuss the quality of the neutron–gamma separation in both detectors. Compare the shape and separation of the branches in deuterated stilbene and CLYC-6.

3. **Fast-neutron detection**

   Compare the fast-neutron response of deuterated stilbene and CLYC-6 using the naked ^{252}Cf measurements. The deuterated-stilbene detector is expected to show a clear fast-neutron branch produced by neutron scattering on deuterium. CLYC-6 is also sensitive to fast neutrons, mainly through reactions on chlorine.

4. **Thermal and epithermal neutron detection with CLYC-6**

   Compare the CLYC-6 spectra acquired with and without polyethylene moderation. The polyethylene slab slows down part of the ^{252}Cf neutron spectrum, increasing the number of thermal and epithermal neutrons reaching the detector. These neutrons are detected efficiently through the $^{6}\mathrm{Li}(n,\alpha)^{3}\mathrm{H}$ reaction and appear as a bump around 3.2 MeVee.

5. **PSD and energy cuts**

   Discuss the cuts used to select:

   - all neutron events in CLYC-6,
   - fast-neutron events in CLYC-6,
   - thermal/epithermal neutron events in CLYC-6,
   - fast-neutron events in deuterated stilbene.

Explain how the selected event populations change when the polyethylene moderator is inserted.

6. **Intrinsic efficiency**

   Estimate the intrinsic efficiencies for the three cases studied:

   - CLYC-6 detection of thermal/epithermal neutrons,
   - CLYC-6 detection of fast neutrons,
   - deuterated-stilbene detection of fast neutrons.

   The calculation must include the activity of the ^{252}Cf source, the spontaneous-fission branching ratio, the average neutron multiplicity per fission, the acquisition live time, the number of selected neutron events, and the solid angle subtended by each detector.

The final report should include the calibrated energy spectra, PSD-ratio versus energy matrices, neutron-selection cuts, selected neutron spectra, and the estimated intrinsic efficiencies for the three requested detection cases.

1.7 Bibliography

- A. Giaz, L. Pellegri, F. Camera, et al., "The CLYC-6 and CLYC-7 response to gamma-rays, fast and thermal neutrons", *Nuclear Instruments and Methods in Physics Research A*, Vol. 810, pp. 132–139, 2016.

- J. Lerendegui-Marco, G. Cisterna, J. Hallam, V. Babiano-Suárez, J. Balibrea-Correa, D. Calvo, I. Ladarescu, G. de la Fuente, B. Gameiro, A. Sanchis-Moltó, P. Torres-Sánchez, and C. Domingo-Pardo, "Imaging neutrons with a position-sensitive monolithic CLYC detector".*Nuclear Instruments and Methods in Physics Research A* 1079,170594 (2025)

- J. Balibrea-Correa, V. Babiano-Suárez, J. Lerendegui-Marco, C. Domingo-Pardo, I. Ladarescu, A. Tarifeño-Saldivia, G. de la Fuente-Rosales, B. Gameiro, N. Zaitseva, et al., "Towards a new generation of solid total-energy detectors for neutron-capture time-of-flight experiments with intense neutron beams", *Nuclear Instruments and Methods in Physics Research A*, Vol. 1072, 170110, 2025.

- F.D. Becchetti, R.O. Torres-Isea, A. Di Fulvio, S.A. Pozzi, J. Nattress, I. Jovanovic, M. Febbraro, N. Zaitseva, and L. Carman, "Deuterated stilbene (stilbene-d_{12}): An improved detector for fast neutrons", *Nuclear Instruments and Methods in Physics Research A*, 2018.

- G.F. Knoll, *Radiation Detection and Measurement*, 4th ed., Wiley & Sons, 2010.

- W.R. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-To Approach*, Springer, 1994.

Exercise 2 Scintillation detector assembly

2.1 Objectives

- To introduce the student to the assembly of scintillation detectors.
- To familiarize the student with the main components of gamma radiation detectors used in spectroscopy, as well as with their characteristics.
- To investigate and compare the relevant physical characteristics of organic (plastic) and inorganic (BGO) scintillators.

2.2 Professors

- Tutor: Luis Caballero (Luis.Caballero@ific.uv.es)
- Assistant: Gustavo Alcalá (Gustavo.Alcala@ific.uv.es)
- Assistant: José Escalante (Jose.Manuel.Escalante@ific.uv.es)

2.3 Theoretical Introduction

When assembling a scintillation detector, the transport and collection of scintillation light must be considered. Losses in light collection can arise from attenuation (absorption) within the scintillator material itself, or from transmission losses at the material boundaries. In our case, since the scintillator volume is small, attenuation losses will not be significant. However, light losses at the scintillator boundaries may be relevant.

It should be recalled that any light ray whose angle of incidence with respect to the normal of the boundary surface (interface) is greater than the critical angle (θ_c), defined as

$$\sin\theta_c = \frac{n_t}{n_i} \ ,$$

will be totally internally reflected (within the material in which the incidence occurs). n_i and n_t are the refractive indices of the material where the light is incident (detector) and transmitted (exterior), respectively. On the contrary, if the angle of incidence (θ_i) is smaller than the critical angle, the ray may be transmitted to the exterior with a transmission angle (θ_t) and a probability given by the Fresnel equations,

$$T_{\parallel} = 1 - \frac{\tan^2(\theta_i - \theta_t)}{\tan^2(\theta_i + \theta_t)} \quad \text{and} \quad T_{\perp} = 1 - \frac{\sin^2(\theta_i - \theta_t)}{\sin^2(\theta_i + \theta_t)} \ ,$$

which depend on the polarization state of the light, also satisfying the Snell's law,

$$n_i \sin\theta_i = n_t \sin\theta_t \ .$$

Therefore, to minimize possible scintillation light losses, the surface of the scintillator material should be covered with an appropriate reflective material. To collect as much scintillation light as possible, the refractive index of the photosensor (photomultiplier tube window) should be as close as possible to that of the scintillator material. In

addition, there must be a good optical coupling between both surfaces (free of air bubbles), which is typically provided by an optical grease whose refractive index is similar to that of both materials.

It is important to note that for regular geometries (e.g., rectangular), a fraction of the light may become trapped and unable to escape from the scintillator. For this reason, it is advisable to break the symmetry of reflections by using a diffuse reflector (i.e., Teflon tape) rather than a specular one (e.g., aluminized foil). Another solution is based on using scintillators with rough (unpolished) surfaces.

2.4 Experimental Setup

The following equipment is available for this laboratory session:

- BC-400 plastic scintillator
- BGO inorganic scintillator
- Teflon tape
- Aluminized foil
- Black insulating tape
- Optical grease
- Hamamatsu photomultiplier tube (PMT)
- Caen PMT preamplifier
- High voltage power supply (TC952)
- Signal amplifier (TC244)
- Palmtop multichannel analyzer (MCA)
- Laptop computer
- Oscilloscope
- ^{137}Cs radioactive source. Alternatives: ^{22}Na or ^{60}Co

2.5 Procedure

Part 1: Organic scintillation detector (Plastic)

In the first part of the laboratory session, we will assemble a plastic scintillation detector (BC-400). We will prepare the scintillator and couple it to a PMT. First, we will wrap the scintillator with Teflon tape, ensuring complete coverage on all faces except the one where the detector is coupled to the PMT. In order to protect the Teflon and ensure optical isolation of the scintillator, it must be wrapped with black insulating tape, taking care not to leave any gaps through which external light could enter. This wrapping maximizes the collection of light produced inside the

plastic scintillator at the PMT, and minimizes any external light that could enter the plastic.

Using the minimum possible amount of optical grease, the plastic scintillator will be coupled to the window (quartz) of the PMT, taking care not to touch either the surfaces or the optical grease with bare hands, so as to avoid leaving air or contaminants between the two surfaces. One surface should be carefully rotated and pressed against the other until the grease is well distributed and good optical coupling is achieved.[1]

Once the plastic is coupled to the PMT, the junction between the scintillator and the tube window must be covered with black insulating tape to prevent external light from entering, avoiding contact between the tape and the PMT window. Likewise, the mechanical robustness of the coupling must be secured using insulating tape to prevent decoupling during measurements (see Figure 4).

Figure 4: Plastic scintillator wrapped and coupled to the PMT.

Next, connect the PMT power cable (red) and the signal output cable of the PMT. Cover the detector with black cloths to shield the system from external light. Connect the PMT signal cable to the oscilloscope as shown in Figure 5.

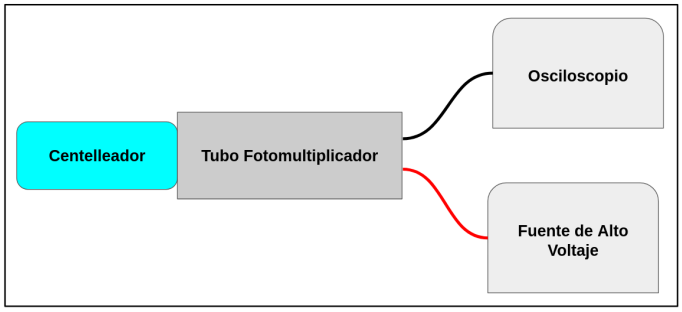


Figure 5: Schematic of the scintillation detector assembly connected to the oscilloscope.

Next, **and only once the detector is completely covered**, switch on the high voltage power supply module (TC952). Check that the "negative polarity LED" is lit, and slowly turn the voltage selector up to -1500 V, verifying the voltage value on the digital display of the module.

Place the ^{137}Cs radioactive source (or ^{22}Na) close to the plastic scintillator and, using the oscilloscope, directly visualize the signals produced by the ^{137}Cs source in the detector.

[1] The purpose of the optical grease is to eliminate the air gap between the scintillator and the PMT, replacing it with a material whose refractive index is similar to that of both surfaces, to minimize internal reflections at this interface.

After verifying the signals from the detector, connect the PMT output to the preamplifier, then to the amplifier (TC244). Connect the amplifier output to the MCA, as indicated in the schematic in Figure 6.

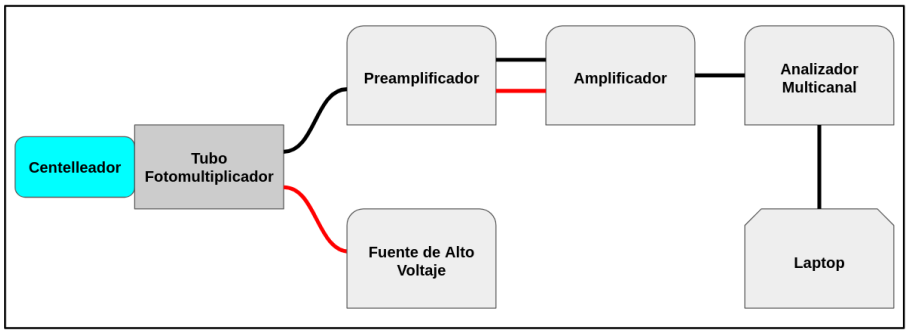


Figure 6: Schematic of the scintillation detector assembly connected to the MCA.

Then, run the "Palmtop MCA" control software on the Windows virtual machine (VirtualBox) on the laptop and record the gamma spectrum. Depending on the range of the acquired spectrum, the amplifier gain will need to be adjusted until a spectrum similar in shape to that shown in Figure 7 is observed.

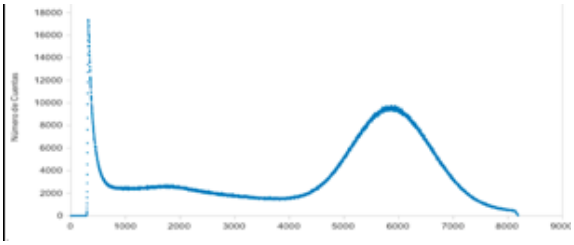


Figure 7: Example of the ^{137}Cs gamma spectrum measured with the MCA.

Once the spectrum has been verified, remove the radioactive source and acquire a background radiation spectrum with an appropriate measurement time. Then, place the ^{137}Cs source in front of the detector and acquire the spectrum again, with an appropriate measurement time.

To compare the detector performance as a function of the type of finish on the scintillator surface, replace the Teflon wrapping with a specular finish (aluminized foil) and an absorbing finish (black cardboard). The procedure is as follows:

1. Remove the radioactive source.
2. Turn the voltage selector until it reads 0 V on the display.
3. Switch off the high voltage (HV) module.
4. Uncover the detector.
5. Replace the previous wrapping with the new one, ensuring optical coupling between the scintillator and the PMT.
6. Cover the detector again to prevent external light from entering.
7. Switch on the HV module.

8. Check that the “negative polarity LED” is lit.

9. Turn the HV module selector to -1500 V.

10. Place the radioactive source in front of the detector and acquire the spectrum with an appropriate measurement time.

11. After each acquisition, **save the spectrum on the computer with a descriptive filename.**

Part 2: Inorganic scintillation detector (BGO)

In the second part of the laboratory session, we will assemble a scintillation detector consisting of a BGO crystal coupled to the PMT. We will compare the detector performance when using different crystal wrappings with varying inner surface finishes: white cardboard, black cardboard, and aluminized foil.

The procedure for both assembling the BGO detector with the oscilloscope and the MCA, and for exchanging each wrapping, is analogous to that followed previously for the plastic detector (see Part 1 of the Laboratory Procedure). Figure 8 shows the BGO detector coupled to the PMT.

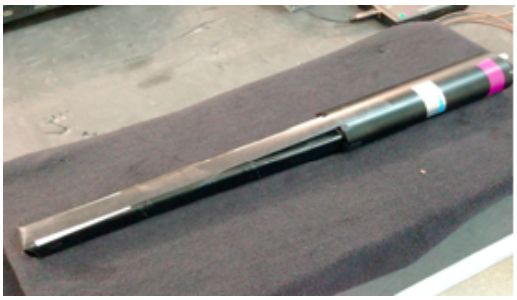


Figure 8: BGO scintillator wrapped and coupled to the PMT.

For this detector, the measurements to be carried out with the different wrappings are analogous to those performed with the organic scintillator. The measurements consist of placing the radioactive source (^{137}Cs) at the same fixed position along the BGO crystal for each wrapping and acquiring the corresponding spectra. Due to the higher efficiency of BGO, the duration of each measurement may be shorter than in the case of the plastic scintillator. Remember to measure the background for this detector.

These measurements allow the study of variations in the spectrum according to the different methods of wrapping the scintillator crystal. As in the first part of the laboratory session, the differences in the spectra are associated with the collection of scintillation light at the PMT window.

Once all measurements are complete, switch off the HV supply to the detector (PMT), disassemble the components, and tidy up the workbench.

2.6 Analysis and interpretation of the results

1. Compare the characteristics of the PMT signals observed on the oscilloscope using the plastic scintillator (BC-400) with the temporal characteristics described in its datasheet.

2. What is the effect of the different reflective wrappings on the plastic scintillator and on the BGO crystal? Explain the underlying reasons and summarize the results obtained for each wrapping used on both scintillators in a table.

3. Rather than using scintillators with all faces left untreated, explain what type of surface finish (coating) applied to the different faces of the crystal would maximize the energy resolution. Justify yourb response.

2.7 Biblography

- PMT datasheet: `https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/etd/PMT_handbook_v4E.pdf`
- Plastic scintillator datasheet: `https://www.hep.phy.cam.ac.uk/~lester/teaching/SparkChamber/SGC_BC400_404_408_412_416_Data_Sheet.pdf`
- BGO datasheet: `https://gammadata.se/wp-content/uploads/2024/01/BGO-data-sheet-v2.pdf`
- W. R. Leo, "Techniques for Nuclear and Particle Physics Experiments". Chapters 7, 8, and 9.
- G. F. Knoll, "Radiation detection and measurement". Chapters 8 and 9.

Exercise 3 High and Medium Resolution Gamma Spectroscopy

3.1 Objectives

- To familiarize students with the main components of gamma radiation detectors used in spectroscopy, particularly scintillation and semiconductor detectors.
- To analyse and compare the energy resolution of two types of radiation detectors: High Purity Germanium (HPGe) and Cerium-activated Lanthanum Bromide ($LaBr_3$:Ce), considering their materials, physical characteristics, and signal production mechanisms.
- To introduce students to digital signal filtering using a Data Acquisition System (DACQ).

3.2 Professors

- David Rodríguez García

3.3 Theoretical Introduction

This section outlines the theoretical framework required to understand the principles of gamma-ray spectroscopy and the operation of the detectors used in this experiment.

3.3.1 Energy deposition

Gamma-ray spectroscopy is based on the measurement of the energy distribution of photons emitted in nuclear transitions. The interaction of gamma radiation with matter is governed primarily by three mechanisms: the photoelectric effect, Compton scattering, and pair production. The relative contribution of each process depends on the photon energy and the atomic number of the absorber.

If we look at the measured energy spectra, in the photoelectric effect the incident photon is fully absorbed, producing a full-energy peak. Compton scattering results in partial energy transfer and generates a continuum, while pair production (for $E_\gamma > 1.022$ MeV) leads to the formation of an electron–positron pair and may produce escape peaks depending on detector geometry.

3.3.2 Differences between scintillation and semiconductor detectors

Two major classes of detectors are employed in gamma spectroscopy: scintillation detectors and semiconductor detectors. Scintillation detectors, such as $LaBr_3$:Ce, convert deposited energy into optical photons. The total number of scintillation photons Nph is approximately proportional to the deposited energy E ($Nph \propto E$), although non-proportionality effects can degrade the linearity and resolution. The light is converted into an electrical signal by a photomultiplier tube (PMT).

In semiconductor detectors such as HPGe, gamma-ray interactions produce secondary electrons, these secondary electrons travel trough the crystal producing electron-hole pairs. The number of charge carriers is given by $N = \frac{E}{\varepsilon}$, where $\varepsilon \approx 2.96$ eV is the average energy required to create one electron–hole pair in germanium. The small value of ε leads to reduced statistical fluctuations and thus superior energy resolution.

3.3.3 Energy resolution

The energy resolution R of a detector is defined as $R = \frac{FWHM}{E_0}$ where FWHM is the Full Width at Half Maximum of a photopeak and E_0 is its centroid energy. The FWHM is not constant but depends on the gamma-ray energy due to different contributing effects. In general, the energy dependence of the resolution can be described by considering three main contributions: statistical fluctuations, electronic noise, and charge collection inefficiencies. These contributions combine quadratically:

$$(FWHM)^2 = (FWHM_{stat})^2 + (FWHM_{noise})^2 + (FWHM_{incomplete})^2$$

Each term has a characteristic dependence on energy:

- The statistical term arises from fluctuations in the number of charge carriers or scintillation photons and scales as: $FWHM_{stat} \propto \sqrt{E}$
- The electronic noise term is largely independent of energy: $FWHM_{noise} \propto const$
- The incomplete charge collection (or non-proportionality) term is often proportional to the deposited energy: $FWHM_{incomplete} \propto E$

Combining these dependencies, the total FWHM as a function of energy is commonly parametrized as:

$$FWHM(E) = \sqrt{a + bE + cE^2}$$

, where:

- a represents the contribution from electronic noise,
- b accounts for statistical fluctuations,
- c describes effects such as incomplete charge collection or non-proportional response.

Accordingly, the energy resolution becomes:

$$R(E) = \frac{FWHM(E)}{E} = \frac{\sqrt{a + bE + cE^2}}{E}$$

This functional form is widely used to fit experimental resolution curves and allows the relative importance of different broadening mechanisms to be quantified for a given detector system.

3.3.4 Signal processing

Signal processing plays a central role in extracting accurate energy information from detector signals. The output of the charge-sensitive preamplifier is typically a step-like pulse with a fast rise time followed by an exponential decay, reflecting the discharge of the feedback circuit. In order to accurately determine the deposited energy, this signal must be shaped to optimize the signal-to-noise ratio, minimize ballistic deficit, and reduce pile-up effects at high count rates.

In modern digital systems, such as the CAEN Data Acquisition (DACQ) modules used in this experiment, the analogue preamplifier signal is directly digitized by a fast analogue-to-digital converter (ADC). The digitized waveform is then processed in real time using digital signal processing (DSP) algorithms implemented in firmware. This approach replaces traditional analogue shaping amplifiers and provides greater flexibility in optimizing the processing parameters.

The Pulse Height Analysis (PHA) firmware developed by CAEN is specifically designed for spectroscopy applications. It implements digital shaping filters, most commonly trapezoidal filters, which approximate the behaviour of ideal rectangular shaping while maintaining finite rise and fall times. The trapezoidal filter is characterized by parameters such as the rise time, flat-top width, and decay (or fall) time. The flat-top region allows for stable amplitude sampling, reducing sensitivity to variations in charge collection time and mitigating ballistic deficit.

In addition to shaping, the PHA firmware includes several essential processing stages: baseline estimation and restoration, pole-zero correction to compensate for the exponential decay of the preamplifier signal, pile-up rejection algorithms, and digital triggering. The pulse amplitude is extracted from the filtered waveform—typically by averaging over the flat-top region—and is proportional to the energy deposited in the detector. The resulting values are accumulated into a histogram to produce the energy spectrum.

The CoMPASS software provides the user interface for configuring the acquisition parameters and visualizing the data in real time. Through CoMPASS, it is possible to adjust key DSP parameters such as shaping time (rise time), flat-top duration, trigger threshold, and gain. These parameters directly influence the trade-off between energy resolution, throughput, and pile-up probability. CoMPASS also allows monitoring of raw and processed waveforms, facilitating optimization of the signal processing chain and diagnostic analysis of detector performance.

A proper selection of the shaping parameters is critical: longer shaping times improve energy resolution by reducing electronic noise contributions, while shorter shaping times allow higher count rates at the expense of resolution. Therefore, the digital processing chain implemented in the CAEN PHA firmware, together with the flexibility provided by CoMPASS, enables precise optimization of detector performance depending on the experimental conditions.

3.4 Experimental Setup

To do this exercise it is necessary the following equipment:

- 1x HPGe detector
- 1x LaBr3:Ce

- 1x NIM crate
- 2x HV power supply
- 1x CAEN DACQ with PHA firmware
- 1x Oscilloscope
- 1x PC with specific software installed (CoMPASS)
- Some radiative sources (^{22}Na, ^{137}Cs, ^{60}Co, ^{152}Eu)

3.5 Procedure

The following steps describe the procedure to follow during the exercise.

1. **Detector setup and high voltage supply**
   Connect both detectors (HPGe and $LaBr_3$:Ce) to their high voltage (HV) power supplies. Set the recommended operating voltage for each detector according to the manufacturer specifications. Ensure that the HPGe detector cooling system is properly functioning before applying HV. Verify stable operation before proceeding.

2. **Signal inspection with oscilloscope**
   Before configuring the digital acquisition system, connect the output signal of each detector to the oscilloscope. Observe the raw signals from both detectors:

   - Identify the characteristic pulse shapes (fast pulses for $LaBr_3$:Ce, slower step-like pulses for HPGe).
   - Measure approximate pulse amplitudes, rise times, and decay constants.
   - Check for noise, distortions, or improper grounding.

3. **Connection to DACQ and CoMPASS configuration**
   Connect the detector outputs to the CAEN digitizer input channels and launch the CoMPASS software. Load the PHA firmware configuration and assign each detector to a dedicated acquisition channel. Configure initial parameters such as input dynamic range, trigger threshold, and polarity according to the observed signals.

4. **Optimization of digital filter parameters**
   Adjust the trapezoidal shaping parameters within CoMPASS:

   - Rising time
   - Flat-top width
   - Pole-zero correction
   - Peaking time position and width

   Perform a systematic study of these parameters to optimize the signal-to-noise ratio and minimize pile-up. Evaluate how changes in filter parameters affect the energy resolution and count rate. Determine a set of optimal parameters for each detector independently.

5. **Data acquisition with radioactive sources**
   Measure several calibrated gamma-ray sources (e.g., ^{22}Na, ^{137}Cs, ^{60}Co, ^{152}Eu) using both detectors. Acquire spectra for sufficient time to obtain well-defined photopeaks with good statistics.

6. **Energy spectrum visualization**
   Use CoMPASS to visualize the energy histograms in real time. Compare the spectra obtained with HPGe and $LaBr_3$:Ce:

   - Identify photopeaks and Compton continua
   - Observe differences in peak width and spectral features

7. **Energy calibration**
   Perform an energy calibration using known gamma-ray energies from the measured sources. Select clear photopeaks and define Regions of Interest (ROI) around them. Fit the peak centroids and establish the calibration curve (channel-to-energy conversion), typically using a linear fit. This step is done using the properly options in CoMPASS.

8. **Resolution determination**
   After calibration, select ROIs around the relevant photopeaks for each detector. Fit the peaks (e.g., Gaussian fits with linear background subtraction) to extract the FWHM and the resolution. Repeat this procedure for different energies to obtain the resolution as a function of energy.

3.6 Analysis and interpretation of the results

The analysis is divided into two parts. The first part is carried out during the experimental procedure, while the second part is completed after the measurements. For the latter, students are required to use their own computers.

3.6.1 During the procedure

1. **Energy calibration**
   Using the acquired spectra, identify well-defined photopeaks from the available radioactive sources (e.g., ^{137}Cs, ^{60}Co, ^{22}Na, ^{152}Eu). Define Regions of Interest (ROI) around each peak and determine their centroid positions (in channels) through Gaussian fitting.

   Construct the energy calibration curve by plotting the known gamma-ray energies versus the measured centroids. Fit the data using an appropriate function (typically linear) and obtain the calibration parameters.

2. **Peak fitting and FWHM extraction**
   For each identified photopeak, perform a fit (Gaussian + background) within the selected ROI. Extract:

   - Centroid energy
   - Full Width at Half Maximum (FWHM)

Ensure that the fits are consistent and that the background is properly accounted for.

3. **Energy resolution calculation**
   Compute the energy resolution for each peak using:

$$R = \frac{FWHM}{E}$$

   Repeat this for multiple energies for both detectors.

3.6.2 After the procedure

1. **Resolution curve determination**
   Plot FWHM as a function of energy for both detectors. Fit the data using:

$$FWHM(E) = \sqrt{a + bE + cE^2}$$

   Determine the parameters a, b, and c, and interpret their physical meaning in terms of noise, statistical fluctuations, and detector-specific effects.

2. **Comparison between detectors**
   Compare the results obtained for HPGe and $LaBr_3$:Ce, plotting both curves in the same graph:

   - Resolution as a function of energy
   - Ability to resolve close gamma lines
   - Differences in peak shape and background

3. **Influence of digital filter parameters**
   Analyse how the variation of shaping parameters (rise time, flat-top, etc.) affects:

   - Energy resolution
   - Peak stability
   - Count rate and pile-up effects

   Identify the optimal parameter configuration for each detector and justify your choice based on the observed data.

4. **Interpretation of results**
   Based on the obtained measurements:

   - Explain the differences in resolution between HPGe and $LaBr_3$:Ce in terms of their physical detection mechanisms.
   - Discuss the dominant contributions to the resolution (noise, statistical, incomplete collection) for each detector.
   - Evaluate the trade-off between resolution and acquisition speed when adjusting the digital filtering parameters.

3.7 Bibliography

- Chapters 2, 4, 8, 9, 10, 11 and 12 from G.F. Knoll: "Radiation detection and measurement"
- K. Debertin and R.G. Helmer: "Gamma- and X-ray spectroscopy with semiconductor detectors"
- CoMPASS Multiparametric DACQ Software for Physics Applications: *https://www.caen.it/products/compass/*
- Manuel CAN DT5725: *https://www.caen.it/download/?filter=DT5725%20/%20DT5725S*
- Radiative source data base: *http://www.lngb.fr/Laraweb/index.php*

Exercise 4 Gamma-ray imaging: PET and Compton

4.1 Objectives

- To introduce the students to two gamma-ray imaging techniques, Positron Emission Tomography (PET) and Compton imaging, using an experimental setup based on position-sensitive gamma-ray detectors operated in time coincidence.
- To understand the physical principles that make it possible to reconstruct the position of a radioactive source from the detection of gamma rays.
- To become familiar with the i-TED experimental system, its detector modules, readout electronics, acquisition software, and event-building procedure.
- To perform an energy calibration of the gamma-ray detectors using a ^{152}Eu source.
- To acquire data with a ^{22}Na source placed at different positions and use these data to reconstruct PET and Compton images.
- To compare PET and Compton imaging in terms of field of view, spatial resolution, event selection, and dependence on gamma-ray energy.
- To analyse the reconstructed images quantitatively using the provided ROOT macros and discuss the main limitations of each technique.

4.2 Professors

- Jorge Lerendegui Marco
- Bernardo Bernardino Gameiro

4.3 Theoretical Introduction

Gamma-ray detection systems are commonly used for spectroscopy, where the objective is to determine the energy of the incoming radiation. Some detector systems, however, also allow the reconstruction of the position from which the gamma rays were emitted. This is known as gamma-ray imaging.

In this exercise, two imaging techniques will be studied: Compton imaging and Positron Emission Tomography (PET). Both techniques are based on the detection of gamma rays in coincidence, but they exploit different physical processes and therefore have different requirements and limitations.

Compton imaging

Compton imaging is based on the Compton scattering of an incoming gamma ray in a first detector plane, called the scatterer, followed by the absorption of the scattered gamma ray in a second detector plane, called the absorber (see Fig. 10). If the energy deposited in the scatterer is E_1 and the energy deposited in the absorber is E_2, the initial gamma-ray energy is assumed to be

$$E_\gamma = E_1 + E_2.$$

The interaction positions in the two detector planes, $\vec{r}_1$ and $\vec{r}_2$, define the direction of the scattered gamma ray. The Compton scattering angle θ can be obtained from the Compton formula:

$$\cos\theta = 1 - m_e c^2 \left(\frac{1}{E_2} - \frac{1}{E_1 + E_2} \right),$$

where $m_e c^2 = 511$ keV is the electron rest energy. For each detected coincidence, the source position is constrained to lie on the surface of a cone. By back-projecting many such cones, the most probable source position can be reconstructed from the region where the cones overlap.

The angular resolution of a Compton camera depends mainly on the energy resolution of the detectors and on the precision with which the interaction positions are reconstructed. Therefore, good energy resolution, good timing resolution, and position sensitivity are essential for Compton imaging.

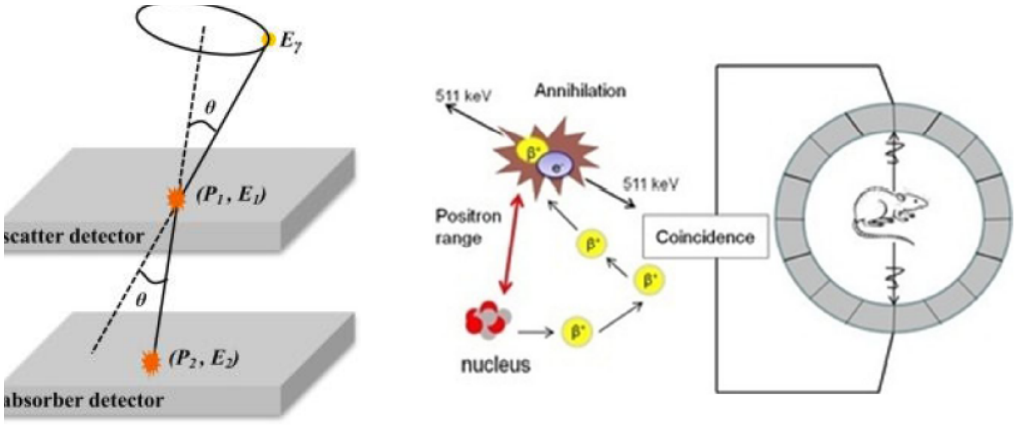


Figure 9: Schematic representation of the basic operating principles of PET imaging (right) and Compton imaging (left).

PET imaging

PET is based on the detection of the two 511 keV photons emitted after the annihilation of a positron with an electron. A positron-emitting isotope, such as ^{22}Na, emits a positron that loses energy in the medium and eventually annihilates. The annihilation produces two gamma rays of 511 keV emitted approximately back-to-back, at 180° (see Fig. 10).

If the two 511 keV photons are detected in coincidence in two opposite detectors, the source is assumed to lie along the line joining the two interaction positions. This line is called the line of response. In a clinical PET scanner, many detectors surround the patient. In this laboratory setup, two position-sensitive detector modules are placed facing each other, allowing PET images to be reconstructed from the coincidence lines between the two detector planes.

Unlike Compton imaging, PET requires a positron-emitting source and is therefore restricted to β^+ emitters. However, PET generally provides better spatial resolution when the source lies within the common field of view of the two opposed detectors.

4.4 Experimental Setup

The experiment uses the i-TED detector system, originally developed as a Compton imaging system for neutron-capture experiments and adapted here to perform simultaneous PET and Compton imaging.

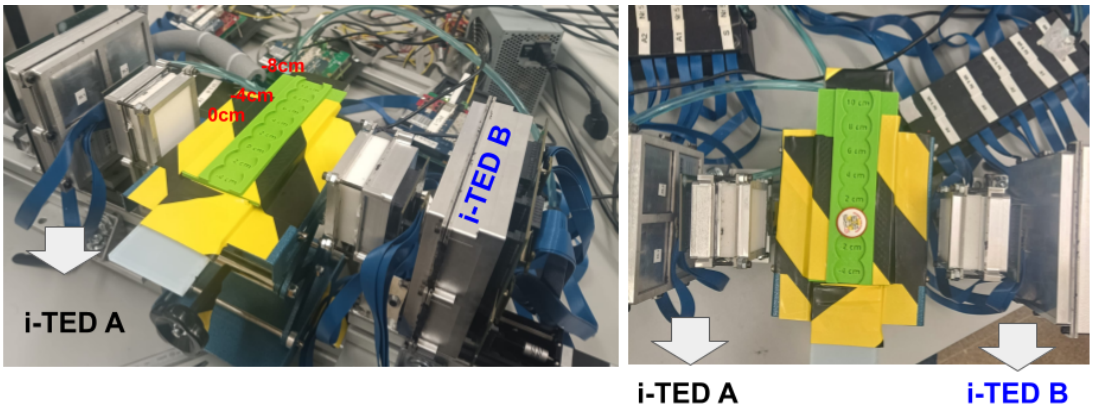


Figure 10: Laboratory setup at IFIC used in this practicum, consisting of two opposed Compton cameras (right). Top view from the setup with a ^{22}Na source located in one of the measuring positions (left).

The setup used in the practicum, depicted in Fig. 10, consists of two opposed i-TED Compton cameras, referred to as i-TED A and i-TED B. Each camera is composed of two detector planes:

- One scatterer detector: a monolithic $LaCl_3(Ce)$ scintillation crystal of dimensions $50 \times 50 \times 10$ mm^3.
- Four absorber detectors: monolithic $LaCl_3(Ce)$ scintillation crystals of dimensions $50 \times 50 \times 25$ mm^3.

Each crystal is coupled to an 8×8 pixelated silicon photomultiplier array, giving 64 readout channels per crystal. The pixelated readout provides position sensitivity by measuring the light distribution produced in the scintillator after a gamma-ray interaction.

The signals are read out using PETsys electronics. The acquisition system records, for each fired pixel, a channel identifier, a timestamp, and an integrated charge proportional to the scintillation light collected by that pixel. These raw data are later processed to build detector-level events.

The main components of the setup are:

- Two i-TED Compton cameras facing each other.
- $LaCl_3(Ce)$ scintillation crystals used as scatterer and absorber detectors.
- Pixelated SiPM arrays coupled to each crystal.
- PETsys front-end electronics and acquisition software.
- A computer with the acquisition and reconstruction software.

- A positioning system for placing the radioactive source at different positions.
- Radioactive sources: ^{152}Eu for energy calibration and ^{22}Na for PET and Compton imaging.
- ROOT-based analysis macros for the reconstruction and quantitative analysis of the images.

The detectors used for each of the imaging techniques are shown in the right panel of Fig. 11. For Compton imaging, coincidences are selected within the same i-TED module (i-TED A will be used), between the scatterer and the absorber plane (see right panel of Fig. 11). For PET imaging, coincidences are selected between detectors belonging to the two opposite i-TED modules. This allows 25 PET coincidence lines to be built from the possible combinations between the five detectors of i-TED A and the five detectors of i-TED B.

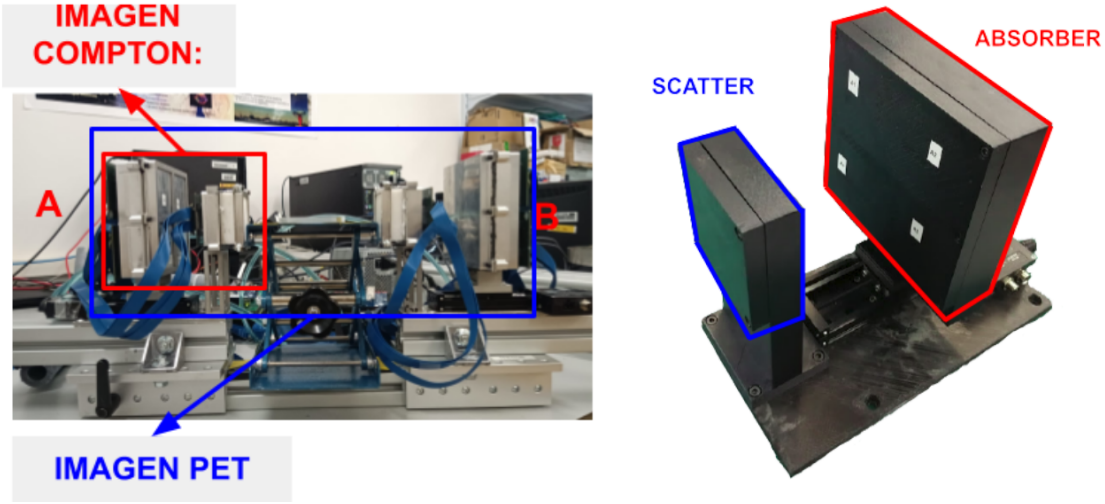


Figure 11: Selection of detectors used for each imaging technique. One of the Compton cameras is shown on the right, highlighting the two detection planes: the scatterer plane and absorber plane.

4.5 Procedure

Detailed steps to perform the exercise are given below.

1. **Introduction to the setup and acquisition system**
   (a) Identify the two i-TED modules used in the experiment.
   (b) Locate the scatterer and absorber planes in each module.
   (c) Identify the radioactive source position and the reference axes used for the reconstructed images.
   (d) Open the HYMNS–iTED–PETsys acquisition software.
   (e) Start the server after checking that the detector system is powered.
   (f) Configure the acquisition by setting the isotope name, measurement duration, and acquisition options.

(g) Select the option to automatically build singles files after acquisition.

2. **Energy calibration with ^{152}Eu**

   (a) Place the ^{152}Eu calibration source between the scatterer and absorber planes of one i-TED module.

   (b) Acquire a short calibration measurement, typically of about 5 minutes.

   (c) After the acquisition and processing have finished, locate the generated ROOT file.

   (d) Inspect the energy spectra of the individual detectors.

   (e) Identify the main ^{152}Eu gamma-ray peaks used for calibration.

   (f) Fit the peak positions and obtain the calibration function that relates detector channel or charge to deposited energy.

   (g) Discuss the energy resolution as a function of gamma-ray energy.

3. **Acquisition of PET and Compton data with ^{22}Na**

   (a) Place the ^{22}Na source on the positioning support between i-TED A and i-TED B.

   (b) Measure and record the distance between the source and the detector modules, since this is an important input for image reconstruction.

   (c) Acquire data for several source positions along the positioning axis. The measurements are typically taken at different positions separated by a few centimetres.

   (d) For each position, acquire data for approximately 10 minutes.

   (e) After each acquisition, verify that the ROOT file containing the processed events has been created.

   (f) Inspect a short spectrum and identify the 511 keV annihilation peak and the 1274 keV gamma ray from ^{22}Na.

4. **Event building**

   (a) Starting from the raw PETsys data, build single-crystal events.

   (b) A single event is obtained by grouping the pixels fired within a time window of approximately 100 ns from the first pixel hit.

   (c) For each single event, calculate the deposited energy as the sum of the charges collected by all fired pixels.

   (d) Assign the event time as the timestamp of the first fired pixel.

   (e) Store the charge map, total charge or energy, and timestamp for each crystal.

   (f) Build coincidence events by selecting single events occurring in different crystals within the coincidence time window, typically 10 ns.

   (g) Store the relevant quantities for the coincidence event, including detector identifiers, energies, timestamps, and pixel charge maps.

5. **PET image reconstruction**

   (a) Select coincidences between detectors belonging to opposite modules, i-TED A and i-TED B.

   (b) Apply a PET energy selection corresponding to the two annihilation photons. Initially, a broad energy window may be used, for example:

   $$E_A + E_B = 0.5\text{–}1.2 \text{ MeV}.$$

   (c) Repeat the reconstruction using a narrower energy window around the 1022 keV sum peak:

   $$E_A + E_B = 0.9\text{–}1.15 \text{ MeV}.$$

   (d) Reconstruct the individual PET images corresponding to the 25 possible coincidence lines between the detectors of the two i-TED modules.

   (e) Sum the 25 individual PET images to obtain the final PET image.

   (f) Use the provided ROOT macro to obtain the summed image, the X and Y projections, and the quantitative spatial resolution.

6. **Compton image reconstruction**

   (a) Select coincidences within the same i-TED module, between the scatterer and one of the absorber detectors.

   (b) Reconstruct the interaction position in each detector from the fitted light distribution measured by the pixelated SiPM.

   (c) Calculate the deposited energies E_1 and E_2 and the total gamma-ray energy $E_\gamma = E_1 + E_2$.

   (d) Apply an energy window to select one of the gamma rays emitted by ^{22}Na.

   (e) For the 511 keV annihilation line, use a window such as:

   $$E_1 + E_2 = 0.4\text{–}0.6 \text{ MeV}.$$

   (f) For the 1274 keV gamma ray, use a window such as:

   $$E_1 + E_2 = 1.2\text{–}1.4 \text{ MeV}.$$

   (g) Reconstruct the Compton image using the back-projection algorithm.

   (h) Generate the image and the X and Y projections using the provided ROOT macro.

   (i) Repeat the reconstruction for different source positions and for the two selected gamma-ray energies.

7. **Saving and organising results**

   (a) Save the reconstructed PET and Compton images.

   (b) Save the corresponding energy spectra used for the event selection.

   (c) Save the X and Y projections of each image.

   (d) Save the text files containing the analysed spatial resolutions.

4.6 Analysis and interpretation of the results

The analysis should focus on understanding how the event selection and the reconstruction method affect the final images. The students should use the reconstructed ROOT files, images, projections, and resolution values obtained during the session.

The following points should be addressed:

1. **Event selection for PET and Compton imaging**

   Discuss the conditions applied to reconstruct each type of image:

   - Detector selection.
   - Time-coincidence window.
   - Energy cuts.
   - Geometrical conditions.

   For PET, the selected events are coincidences between detectors of opposite i-TED modules, with energies compatible with the detection of two 511 keV annihilation photons. For Compton imaging, the selected events are coincidences within the same module, between scatterer and absorber detectors, with a total deposited energy compatible with the selected gamma-ray energy.

2. **Comparison between PET and Compton images**

   Compare the reconstructed PET and Compton images for the same source position. Discuss which technique gives the better spatial resolution and under which conditions. PET is expected to provide better spatial resolution when the source is located within the common field of view of the two opposed detector modules. Compton imaging, however, has a larger angular field of view and can reconstruct gamma rays that do not necessarily produce two back-to-back 511 keV photons.

3. **Field of view limitations**

   Discuss why PET images may fail to reconstruct the source when it is located outside the geometrical acceptance of the two opposed detector planes. In such cases, the two annihilation photons may not be detected in coincidence by opposite detectors. Compton imaging can still reconstruct the source if enough scatterer–absorber coincidences are recorded within one Compton camera.

4. **Effect of gamma-ray energy in Compton imaging**

   Compare the Compton images reconstructed using the 511 keV and 1274 keV gamma-ray lines from ^{22}Na. Discuss which energy gives better image resolution and why. The result should be interpreted in terms of Compton kinematics, energy resolution, interaction probability, and the uncertainty in the reconstructed Compton angle.

5. **Effect of the PET energy window**

   Compare PET images reconstructed with the broad energy window and with the narrower window around the 1022 keV sum peak. Discuss whether the optimized energy selection improves the image resolution, reduces background, or reduces the number of selected events.

6. **Quantitative resolution analysis**

   Use the provided ROOT macros to extract the X and Y projections of the images. Fit the projections around the reconstructed source position and determine the spatial resolution. Compare:

   - PET versus Compton resolution.
   - Compton resolution at 511 keV versus 1274 keV.
   - The effect of different source positions.
   - The effect of different energy-selection windows.

7. **Discussion of uncertainties and limitations**

   Comment on the main sources of uncertainty in the experiment:

   - Finite detector size and segmentation.
   - Uncertainty in the reconstructed interaction position.
   - Energy resolution of the $LaCl_3$(Ce) detectors.
   - Time-coincidence selection.
   - Limited statistics.
   - Source positioning uncertainties.
   - Background events and accidental coincidences.

The final report should be concise and should justify the answers using the figures obtained during the practicum: reconstructed PET and Compton images, energy spectra, image projections, and extracted spatial-resolution values.

4.7 Bibliography

- J.J. Vaquero and P. Kinahan, "Positron Emission Tomography: Current Challenges and Opportunities for Technological Advances in Clinical and Preclinical Imaging Systems", *Annual Review of Biomedical Engineering*, Vol. 17, pp. 385–414, 2015.

- L. Mihailescu, K.M. Vetter, M.T. Burks, E.L. Hull, and W.W. Craig, "SPEIR: A Ge Compton camera", *Nuclear Instruments and Methods in Physics Research A*, Vol. 570, No. 1, pp. 89–100, 2007.

- J. Balibrea, J. Lerendegui-Marco, et al., "Machine Learning aided 3D-position reconstruction in large $LaCl_3$ crystals", *Nuclear Instruments and Methods in Physics Research A*, Vol. 931, pp. 1–22, 2021.

- V. Babiano-Suárez, "i-TED: a Compton imager with Dynamic Electronic Collimation", *Nuclear Instruments and Methods in Physics Research A*, Vol. 953, 163228, 2020. doi: `10.1016/j.nima.2019.163228`.

- B. Gameiro et al., "Towards Real Time Compton Imaging in Demanding Conditions", *Lecture Notes in Computer Science*, Vol. 15386, 2025.

- J. Balibrea-Correa, J. Lerendegui-Marco, I. Ladarescu, et al., "Hybrid in-beam PET- and Compton prompt-gamma imaging aimed at enhanced proton-range verification", *European Physical Journal Plus*, Vol. 137, 1258, 2022.
- J. Lerendegui-Marco, J. Balibrea-Correa, et al., "First Pilot Tests of Compton Imaging and Boron Concentration Measurements in BNCT Using i-TED", *Applied Radiation and Isotopes*, Vol. 225, 112009, 2025.
- V. Babiano-Suárez, J. Balibrea-Correa, I. Ladarescu, J. Lerendegui-Marco, J.L. Leganés-Nieto, and C. Domingo-Pardo, "A computer-vision aided Compton-imaging system for radioactive waste characterization and decommissioning of nuclear power plants", *Nuclear Instruments and Methods in Physics Research A*, Vol. 1076, 170449, 2025.

Exercise 5 System of chip for system health monitoring

5.1 Objectives

General Objective: The general objective of this practicum is to introduce students to the design and deployment of a remote health-monitoring data acquisition system based on a System-on-Chip FPGA. Using the ZUBoard platform, students will integrate programmable logic and embedded software to acquire temperature data from a sensor (STTS22HTR), manage communication through a serial interface, and stream the data to a database infrastructure. The practicum aims to provide hands-on experience with hardware–software co-design, embedded monitoring, and standard data visualization tools widely used in high-energy physics instrumentation and large-scale experimental facilities.

Specific Objectives:

- Understand the SoC FPGA architecture and its relevance for DAQ systems.
- Implement sensor readout logic in programmable hardware.
- Develop embedded software to interface with hardware peripherals.
- Configure serial communication between the SoC and a host computer.
- Implement a Python script for periodic data acquisition.
- Store time-series data in InfluxDB.
- Visualize system health metrics using Grafana dashboards.
- Appreciate remote monitoring concepts in distributed experiments.

5.2 Professors

- Alberto Valero
- David Hernández

5.3 Theoretical Introduction

Modern particle physics experiments rely on complex and distributed detector systems where continuous monitoring is essential to ensure data quality and reliability. Parameters such as temperature, voltage, current, and environmental conditions—known as system health or slow control variables—are not part of the physics data itself but are critical for detecting performance degradation, failures, or unsafe operation. As a result, health monitoring is a fundamental component of modern data acquisition (DAQ) systems. System-on-Chip (SoC) FPGAs are widely used in instrumentation because they integrate programmable logic with embedded processors on a single device. In platforms such as the Xilinx Zynq UltraScale+, time-critical and parallel tasks are implemented in hardware, while control, communication, and data

management are handled by software running on ARM cores. This combination is particularly suitable for DAQ systems, where low-latency interfaces must coexist with flexible control and networking. In monitoring applications, programmable logic interfaces directly with sensors (e.g., via I2C or SPI), ensuring precise timing and low CPU usage. The embedded processor reads these measurements through memory-mapped registers and manages data flow and communication. This separation highlights a key concept: hardware performs deterministic acquisition, while software handles control and data processing. For remote monitoring, measurements must be transmitted to external systems. Serial communication (UART) provides a simple and robust method for streaming data to a host computer, especially in prototyping environments. On the host side, Python is commonly used to read, process, and forward data to storage systems, enabling rapid development and flexibility.

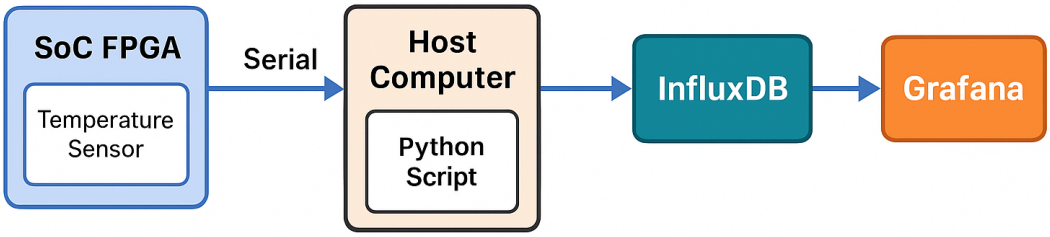


Figure 12: Block diagram of the System-On-Chip monitoring system.

Time-series databases such as InfluxDB efficiently store timestamped data and support temporal analysis, making them ideal for monitoring applications. Visualization tools like Grafana allow real-time and historical inspection through dynamic dashboards, providing intuitive insight into system behavior (Figure 12). Together, these components form a complete monitoring chain—from sensor acquisition to visualization—reflecting architectures used in real high-energy physics experiments.

5.4 Experimental Setup

5.4.1 Introduction

The experimental setup described in this practicum is designed to demonstrate a complete and realistic system-health monitoring chain based on a System-on-Chip (SoC) FPGA platform (Figure 13). The objective is to understand a full data path, starting from physical sensor acquisition and ending with remote visualization of system parameters using monitoring tools. Although the setup focuses on temperature as a representative variable, the same architecture can be extended to other slow-control parameters commonly monitored in high-energy physics experiments, such as voltages, currents, and environmental conditions. The system is organized around a ZUBoard, which integrates programmable logic and embedded processing on a single device. The SoC periodically reads a temperature sensor and transmits the measurement to an external host computer using a serial communication interface. On the host side, a Python script performs data parsing, timestamping, and

forwarding to a time-series database. The collected data are then stored in InfluxDB and visualized in Grafana through interactive dashboards.

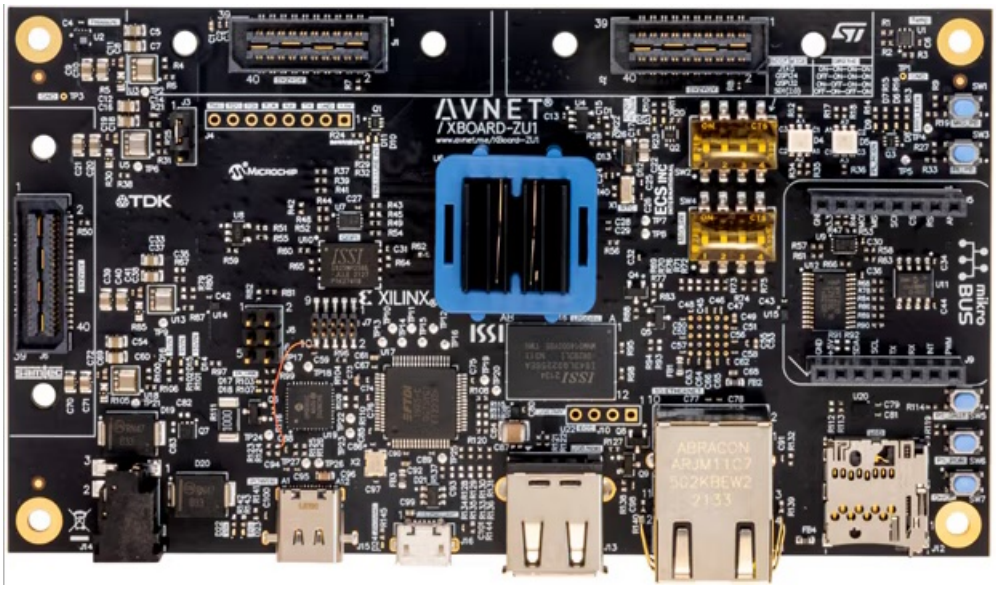


Figure 13: Picture of the Zuboard development board.

5.4.2 Hardware setup

The core hardware element of the experimental setup is the ZUBoard, a development platform based on a Xilinx Zynq UltraScale+ System-on-Chip FPGA. SoC FPGAs combine two traditionally separate computing paradigms within a single silicon device: a high-performance programmable logic fabric and one or more embedded general-purpose processors (Figure 14). This tight integration enables efficient hardware–software co-design and makes SoC FPGAs particularly well suited for data acquisition and monitoring applications. The **processing system** (PS) of the Zynq UltraScale+ device is based on ARM Cortex-A class processors, typically Cortex-A53 cores, capable of running bare-metal firmware or embedded Linux. The PS includes standard peripherals such as UARTs, I²C controllers, SPI controllers, timers, and memory controllers, which can be accessed directly from **software**. In this setup, the PS is responsible for system control, communication with the host computer, and coordination of data transfers from the programmable logic. The **programmable logic** (PL) corresponds to the FPGA fabric, composed of configurable logic blocks, block RAMs, DSP slices, and routing resources. The PL is used to implement **custom** digital hardware, such as sensor interfaces, data formatting logic, and memory-mapped registers. In the context of this practicum, the PL implements the low-level interface to the temperature sensor (STTS22HTR), either through a standard protocol such as I²C or via an internal monitoring block. The PL exposes the acquired temperature value to the PS through an AXI-based interconnect, allowing the processor to read the measurement as if it were a memory-mapped peripheral.

The ZUBoard provides accessible I²C interfaces through its connectors, making it straightforward to attach sensors. Serial communication between the SoC and the host computer is implemented using a UART interface. UART is chosen for its simplicity, robustness, and ease of debugging. The serial link allows the SoC to periodically transmit temperature values in a structured or human-readable format.

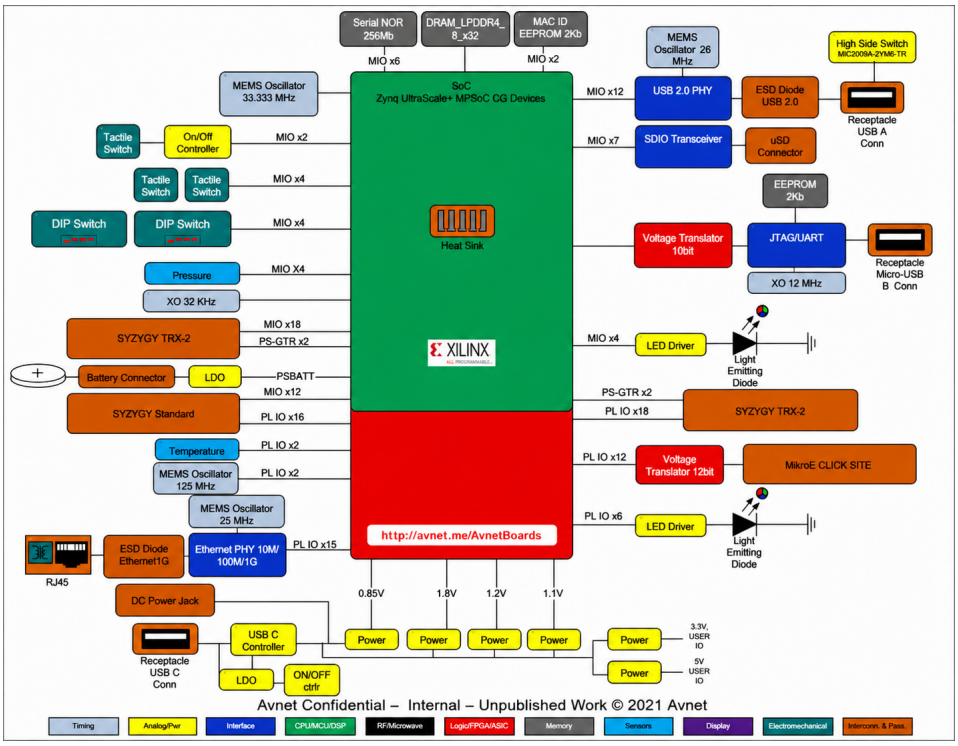


Figure 14: Block diagram of the ZUboard FPGA and peripherals connections. The green block corresponds to the processing system (PS) whereas the read block is the programmable logic (PL) part of the FPGA.

5.4.3 Embedded System Design and Firmware Development

The firmware development in this practicum follows a hardware–software co-design approach, where system functionality arises from the interaction between programmable logic and embedded software in a SoC FPGA. Understanding this interaction is a key objective.

Hardware Description Languages (HDLs) Digital hardware in the FPGA is described using HDLs such as VHDL or Verilog, which define concurrent circuit behavior rather than sequential algorithms. In this setup, HDL implements the temperature sensor interface, including communication (e.g., I²C), timing control, and data storage in registers. These registers are exposed to the processor via a standard bus, enabling clear hardware–software communication.

IP Cores and System Integration FPGA development relies on pre-designed IP cores that implement common functions like communication interfaces and interconnects. In this practicum, IP cores (e.g., AXI, UART, I²C) are combined with custom HDL blocks, allowing students to focus on system architecture while illustrating integration within a single SoC design.

Xilinx Vivado: Hardware Design Environment Vivado is used to configure the processing system, integrate IP cores and custom hardware, and define interconnections via AXI. It also performs synthesis, implementation, and bitstream generation, translating the design into a configuration for the FPGA.

Software Development with Vitis After hardware design, the platform is exported to Vitis for software development on the ARM processor. The software, written in C, handles initialization, peripheral communication, and system control. In this setup, it reads temperature data from memory-mapped registers and transmits it via UART, with acquisition and communication fully controlled in software.

5.4.4 Grafical Interface

The graphical interface of the monitoring system is built using InfluxDB and Grafana, which provide time-series data storage and visualization. This layer transforms raw sensor data into accessible information for users. InfluxDB is a time-series database designed to store large volumes of timestamped data efficiently. In this setup, temperature measurements are written to the database with timestamps, supporting aggregation, filtering, and retention policies. These features make it well suited for slow-control and system-health monitoring, where long-term trends are important. Grafana is used for visualization and dashboarding. It connects to InfluxDB to create interactive plots showing temperature evolution over time. Dashboards can be customized with different time ranges and thresholds, providing real-time feedback and helping students correlate system behavior with hardware or software changes. Together, InfluxDB and Grafana form a monitoring interface similar to those used in large-scale experiments, highlighting the importance of data visualization alongside hardware and firmware development.

5.5 Procedure

5.5.1 Part I: SoC Hardware Design Using Vivado

Step1: Create a New Vivado Project

Open a *Terminal*, setup the environment and launch Vivado 2023.2:

```
> source /opt/Xilinx/Vivado/2023.2/settings64.sh
> vivado
```

Now we are going to create the Vivado project for our setup.

1. *File → Project → New*
2. *Follow the project creation wizard*

   Choose a project name and location.
3. *Select RTL Project. Do not add sources at this stage.*
4. *Select the correct ZUBoard from the board list.*

We are defining the target hardware platform so Vivado knows which FPGA resources, pins, and peripherals are available. **Confirm that the selected device matches the ZUBoard used in the lab**. An incorrect device selection will lead to invalid pin assignments later.

Step 2: Create a Block Design and Add the Processing System Now we need to instantiate the ARM-based processing system that will run the embedded software and manage communication with the outside world.

In the Flow Navigator:

1. *Select IP Integrator → Create Block Design*
2. *Give the design a name (e.g. soc_temp_monitor)*

   Add the processing system:

3. *Click Add IP*
4. *Search for Zynq UltraScale+ MPSoC. Double-click to add it to the diagram.*
5. *Run Block Automation.*

We have now included the ARM Processor and defined the peripherals connected to it. The *Run Block Automation* automatically interconnects the system signals according to the selected board.

Open the PS configuration and **verify that a UART peripheral is enabled and assigned to the correct MIO pins** according to the board documentation.

Step 3: Add the I²C Interface in the Programmable Logic At this point, pause to explicitly define the system architecture:

- PS: system control, software execution, UART communication
- PL: temperature sensor interface via I²C. Although the PS includes I²C controllers, we intentionally implement I²C in the PL to illustrate custom hardware design and hardware–software interaction.

1. *Click Add IP*
2. *Search for AXI IIC and add it to the block design.*

   These are pre-verified, reusable functional modules used to speed up FPGA and SoC design. You can also implement your custom I²C block if you wish.

3. *Run Connection Automation*

   This will automatically connect the AXI interface of the I²C block to the PS via an AXI Interconnect.

4. *If the output port (IIC) of the I²C block is not connected to an output port, then right-click on the IIC port of the AXI IIC block and select "Make External"*

   This will connect the IIC port to external ports of the FPGA. You can rename the port (for instance: "sensor_i2c_pl"). In any case the name should match the name of the ports used later in the constraints file .xdc (in the example we will use: "sensor_i2c_pl").

5. Verify that the I²C IP is clocked and reset correctly and that it appears in the address map. The final block diagram should look like Figure 15.

What is the AXI Bus?

On the Zynq UltraScale+ MPSoC, the AXI bus provides the standard interface between the Processing System (PS) and the Programmable Logic (PL). AXI enables memory-mapped communication for control and data exchange, allowing software running on the ARM processors to access custom hardware accelerators and peripherals implemented in the FPGA fabric. In our case, it is used to communicate the I²C block with the Processor.

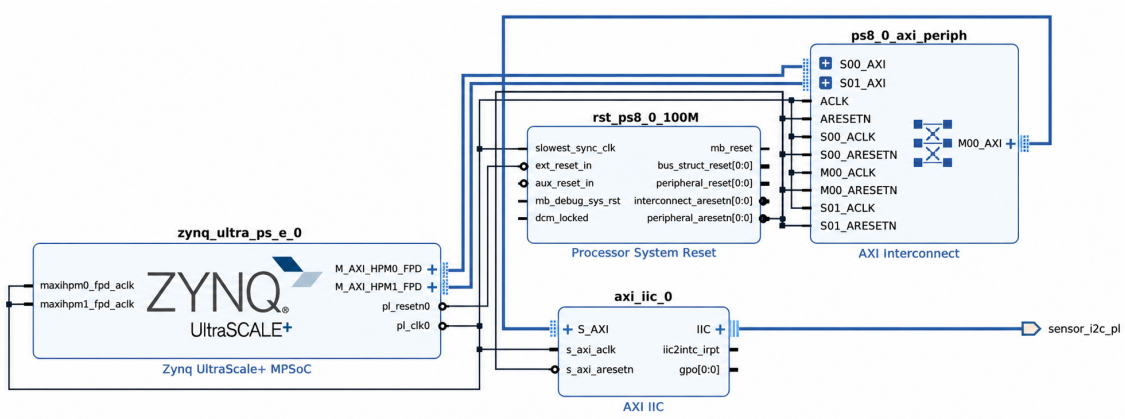


Figure 15: Snapshot of the Vivado Block Design.

Step 4: Assign I²C Pins Using the Schematics The I²C signals in the IIC port (SCL and SDA) must be connected to physical FPGA pins. We need to specify which pins of the FPGA we want to use. Check the ZUboard schematics to know to which pins of the FPGA is the temperature sensor (STTS22HTR) connected.

1. *Open Constraints → Add or Edit Constraints*

2. *Create an .xdc file and paste the following code*

   Replace the *YY* with the correct pins connected to SCL and SDA in the sensor.

```
set_property PACKAGE_PIN YY [get_ports sensor_i2c_pl_scl_io]
set_property IOSTANDARD LVCMOS18 [get_ports sensor_i2c_pl_scl_io]
set_property PULLUP true [get_ports sensor_i2c_pl_scl_io]

set_property PACKAGE_PIN YY [get_ports sensor_i2c_pl_sda_io]
set_property IOSTANDARD LVCMOS18 [get_ports sensor_i2c_pl_sda_io]
set_property PULLUP true [get_ports sensor_i2c_pl_sda_io]
```

3. In the Source tab, right-click in the block and select *Create HDL Wrapper* (Select "Let Vivado manage wrapper and auto-update").

This creates a VHDL block including all the components and interface ports. Check the VHDL code generated to identify the bi-directional ports that we have in the system (double-click on the wrapper).

Step 5: Validate and Generate the Hardware Design

1. *Run Synthesis*: The synthesis step translates the HDL design into a gate-level representation by mapping the described logic into FPGA primitives such as LUTs, flip-flops, and DSP blocks.

   Resolve any errors or warnings.

2. *Run Implementation*: During implementation, the synthesized design is optimized, placed, and routed onto the physical FPGA resources, ensuring timing and resource constraints are satisfied.

3. *Generate Bitstream*: This step produces the final configuration file that programs the FPGA, defining how the hardware will operate on the device.

   Once completed:

4. *File → Export Hardware* Include the bitstream.

We are converting the abstract hardware design into a configuration file that programs the FPGA and defines the hardware platform for software development. At this stage, the **SoC hardware** is complete and ready to be used in the next part of the procedure, where software will be developed using Vitis.

5.5.2 Part II: Vitis Bare-Metal Software Application

Now we have the hardware project implemented. We need to develop the software that will run in the Processor. It should implement the interface with the Sensor through I²C and with the computer through a UART port. **Step1: Create Vitis**

Platform and Application Components

1. *Vivado → Tools → Launch Vitis IDE*
2. *Create Platform Component*
3. *Hardware Design → Select the .XSA file exported previously on Vivado*
4. *Operating System : Standalone. Processor: psu_costexa53_0*
5. *Select generate PMU-FW*
6. *Check in Vitis components that the Platform has been created.*
7. *Settings (json) → standalone_psu_cortexa53_0 → Board Support Package → Regenerate BSP*
8. Select the Platform in VITIS Components → Build

   Check that no errors or warnings appear

9. Create Application Project.*File → New Component → Application*
10. *Select platform*
11. *Sources → src → Add File (read_temperature.c)*

    The program configures the STTS22HTR sensor and then reads out the temperature continuously. The obtained value is transmitted through the serial port.
12. *Build the application. Check that there are no errors.*

Check in the STTS22HTR datasheet the configuration used in the program. Verify that the I²C is correct.

Step2: Configure the FPGA to run the program

1. Connect the power cable and the USB serial cable to the computer.
2. *Power on the board pressing SW7 button*
3. *Vitis → Program Device*

   This will load the .bit file to program the FPGA with the firmware part. Wait for the program to finish
4. *VITIS Componentes → Select the Application → Run*
5. Open a terminal and run a serial console (*sudo gtkterm)*

   Verify the serial port configuration both in gtkterm and in the MPSoC UART moddule. You should see the temperature readings on the screeng.

5.5.3 Part III: Readout software and Monitoring Application

This final part of the practicum completes the monitoring chain by implementing the host-side software and the visualization framework. The objective is to take the temperature values transmitted by the ZUBoard through the UART interface, store them as time-series data in InfluxDB, and visualize their evolution over time using Grafana.

Step1: Python script for serial readout and InfluxDB interface

1. Install required python libraries: *pip3 install pyserial influxdb-client*
2. Implement a python script to readout every second the temperature from Serial port and send it to InFluxDB.

How to readout from Serial port:

```
import serial

# SERIAL CONFIG
SERIAL_PORT = "/dev/ttyUSB1"
BAUDRATE = 115200
# SERIAL SETUP
```

```
ser = serial.Serial(SERIAL_PORT, BAUDRATE, timeout=1)
while True:
        line = ser.readline().decode(errors="ignore").strip()
        time.sleep(1)
```

Convert the RAW value to decimal and send it to InfluxDB:

```
from influxdb_client import InfluxDBClient, Point
from influxdb_client.client.write_api import SYNCHRONOUS

# INFLUXDB CONFIG
INFLUX_URL = "http://localhost:8086"
INFLUX_TOKEN = "my-super-secret-token"
INFLUX_ORG = "zuboard"
INFLUX_BUCKET = "temperature"

# INFLUXDB SETUP
client = InfluxDBClient(
    url=INFLUX_URL,
    token=INFLUX_TOKEN,
    org=INFLUX_ORG
)
write_api = client.write_api(write_options=SYNCHRONOUS)

while True:
     # Create InfluxDB point
            point = (
                Point("temperature")
                .tag("sensor", "zuboard")
                .field("value", temperature)
            )

            write_api.write(
                bucket=INFLUX_BUCKET,
                org=INFLUX_ORG,
                record=point
            )
```

Step2: Deploy InfluxDB and Grafana for temperature monitoring

1. *Create a Docker Compose Configuration*

   Create a folder called *grafana_stack* and copy the file *"dockercompose.yml"*.

   Check the InfluxDB configuration parameters in *"dockercompose.yml"*.

2. *Start InfluxDB and Grafana services*

   ```
   > cd grafana_stack
   > docker compose up
   ```

   Once started, the services are accessible at:

   - InfluxDB: http://localhost:8086
   - Grafana: http://localhost:3000

3. *Configure Grafana Dashboard*

   Open a web browser and navigate to: http://localhost:3000

   User: admin

   Password: admin

   You can also check the database:

   Open a web browser and navigate to: http://localhost:8086

   User: admin

   Password: admin123

4. *Add InfluxDB as a Data Source*

   Open Settings → Data Sources

   Click Add data source and select InfluxDB

   Configure: Query language: Flux

   URL: http://influxdb:8086

   Organization: zuboard

   Token: my-super-secret-token

   Default bucket: temperature

   Click Save & Test

5. *Create a Temperature Plot*

   Click → Dashboard → Add new panel

   Select InfluxDB as the data source

   Enter the following Flux query:

```
from(bucket: "temperature")
  |> range(start: -10m)
  |> filter(fn: (r) => r["_measurement"] ==
     "temperature")
  |> filter(fn: (r) => r["_field"] == "value")
  |> aggregateWindow(every: 1s, fn: mean, createEmpty:
     false)
  |> yield(name: "mean")
```

   Set visualization to Time series.

   Set unit to Celsius (°C).

   Set dashboard refresh interval to 1 s.

   Apply the changes to finalize the dashboard.

At this point, the complete monitoring system is operational, demonstrating a realistic slow-control and health-monitoring chain based on an SoC FPGA, host-side software, and professional data visualization tools Figure 16. You can check if the data is being stored in InfluxDB by accesing from a browser http://localhost:8086.

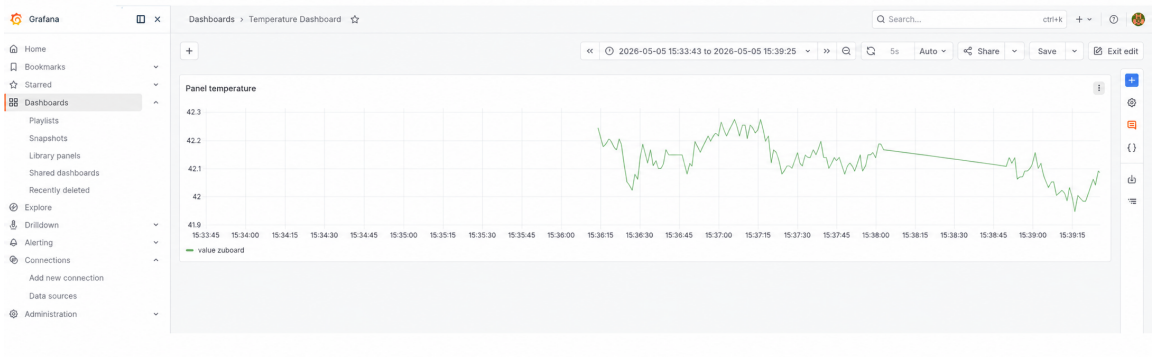


Figure 16: Snapshot of the Grafana monitoring dashboard.

5.6 Analysis and interpretation of the results

Now that the complete process to implement a monitoring system has been introduced, a set of exercises is proposed to extend its functionality. These activities aim to reinforce the key concepts explored throughout the practicum.

1. *Temperature alarm configuration* Program the temperature sensor alarm threshold using its configuration registers and verify its behavior by triggering and monitoring the alarm condition.

2. *Pressure sensor readout (PS-based)* Implement the readout of the pressure sensor available on the ZUBoard, which is directly connected to the Processing System (PS), and integrate its data into the monitoring chain.

3. *Grafana alert configuration* Configure a Grafana dashboard to generate an alert when the measured temperature exceeds a defined threshold, and test the alert mechanism.

4. *Multi-sensor monitoring* Extend the system to simultaneously acquire and display multiple sensor variables (e.g., temperature and pressure) within the same dashboard and the internal SYSMON (on-chip) temperature and voltage sensors.

Exercise 6 Cryogenic Characterization and Calibration of Silicon Photomultipliers

6.1 Objectives

General objective. The goal of this exercise is to perform a quantitative and critical characterization of *silicon photomultipliers* (SiPMs) operating at cryogenic temperature (77 K), using realistic datasets acquired with a setup developed for the large-scale characterization of SiPMs in the context of the DUNE experiment, and to extract the main performance parameters relevant for photon detection in large-scale particle physics experiments.

Specific objectives. By the end of this exercise, students should be able to:

- Understand the physical principles governing SiPM operation in Geiger mode, including gain formation, avalanche processes, and the role of overvoltage.
- Reconstruct and process digitized waveforms from a triggered data acquisition system, including baseline estimation, pulse identification, and charge integration.
- Extract the SiPM gain from single photoelectron spectra, and evaluate the impact of noise and analysis choices on its determination.
- Measure the Dark Count Rate (DCR) at cryogenic temperature and assess its statistical properties and limitations.
- Identify and quantify correlated noise contributions, including optical cross-talk and afterpulsing, using both amplitude and time-domain information.
- Analyse time correlations between events in order to distinguish uncorrelated dark noise from correlated processes and non-ideal detector behavior.
- Recognize and interpret non-Poissonian features in the data, such as burst-like structures, and evaluate their impact on detector performance metrics.
- Critically assess systematic uncertainties associated with the measurement and analysis procedures, including threshold effects, integration windows, and selection criteria.
- Relate the extracted detector parameters (gain, DCR, correlated noise) to the performance requirements of large-scale experiments such as DUNE, and discuss their implications for detector design and operation.

6.2 Teachers

- Carlos G. Benítez Montiel
- Justo Martín-Albo Simón
- Julio Ureña González

6.3 Theoretical Introduction

6.3.1 Basic operation of SiPMs

A silicon photomultiplier (SiPM) is an array of avalanche photodiode (APD) microcells operated in Geiger mode. Each microcell consists of a p–n junction biased above its breakdown voltage, V_{bd}, and connected in series with a quenching resistor, R_q.

When a charge carrier (either thermally generated or produced by photon absorption) initiates an avalanche, the multiplication process becomes self-sustaining. The avalanche is passively quenched by the voltage drop across the quenching resistor: as the avalanche current increases, the voltage drop across R_q reduces the effective bias across the diode below the breakdown voltage, thereby stopping the discharge.

As a consequence, each microcell produces a signal with approximately fixed charge, independent of the number of primary carriers that triggered the avalanche. The total charge measured at the output of the SiPM is therefore proportional to the number of fired microcells.

The charge released in a single avalanche is given by:

$$Q = C_{\text{cell}} \cdot V_{OV}, \tag{1}$$

where C_{cell} is the capacitance of the microcell and $V_{OV} = V_{\text{bias}} - V_{bd}$ is the overvoltage. The corresponding gain is:

$$G = \frac{Q}{e} = \frac{C_{\text{cell}} \cdot V_{OV}}{e}. \tag{2}$$

This linear dependence of the gain on the overvoltage is a key feature exploited in SiPM calibration.

After an avalanche, the microcell requires a finite recovery time, determined by the RC time constant of the cell:

$$\tau \sim R_q \, C_{\text{cell}}, \tag{3}$$

where τ is the characteristic recovery time of the microcell. During this recovery process, the cell is partially or fully insensitive to new events. This behavior plays an important role in shaping the time structure of the signals and in the appearance of correlated noise.

A more detailed description of SiPM operation, including device structure and performance parameters, can be found in the Hamamatsu technical guide to SiPMs [1].

6.3.2 Single photoelectron spectrum and gain calibration

Since each SiPM microcell releases approximately the same charge when firing, the total charge measured at the output of the sensor is approximately proportional to the number of fired microcells. As a consequence, the charge distribution of low-light signals exhibits a discrete multi-peak structure corresponding to different numbers of detected photoelectrons (p.e.), as shown in Figure 17 (left panel).

In the absence of correlated noise and saturation effects, the number of detected photoelectrons follows approximately Poisson statistics. The measured charge spectrum therefore consists of a pedestal peak (0 p.e.), followed by peaks corresponding to 1 p.e., 2 p.e., and higher photoelectron multiplicities.

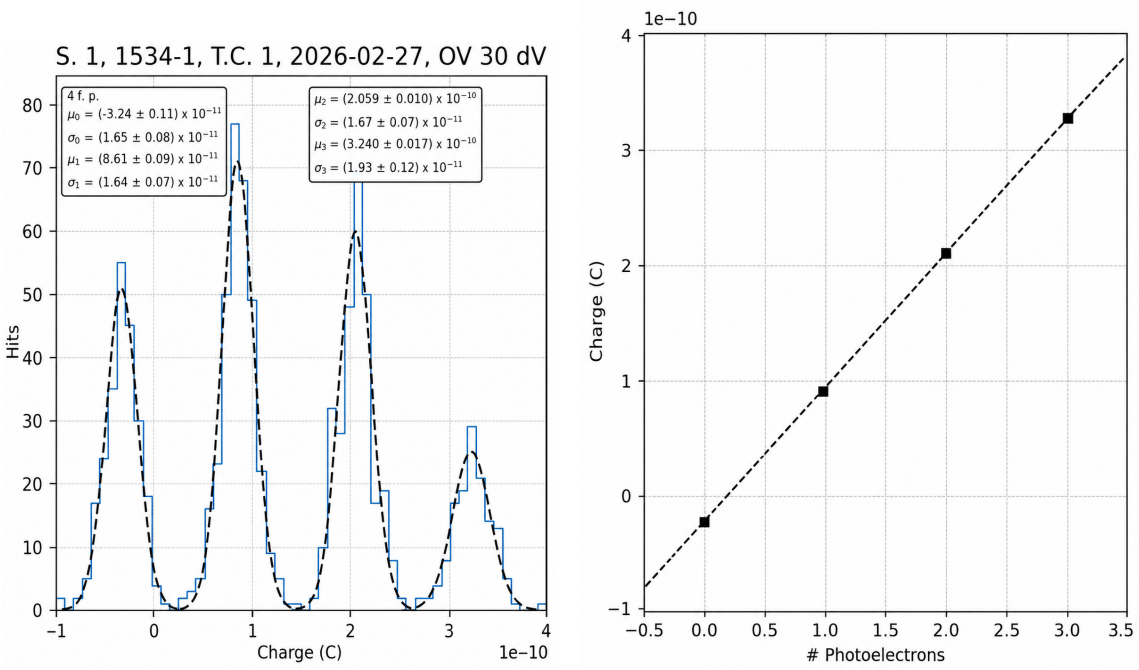


Figure 17: Example of a single photoelectron spectrum (left) and linear fit of the peak positions as a function of the photoelectron multiplicity (right). The slope of the fit provides the SiPM gain in integrated charge units.

Experimentally, the charge associated with each event is reconstructed by integrating the waveform over a suitable time window:

$$Q \propto \sum_i \left(ADC_i - ADC_{\text{baseline}} \right), \tag{4}$$

where ADC_i is the digitized waveform sample and ADC_{baseline} is the baseline level estimated from the pre-trigger region of the waveform.

The gain of the SiPM can then be extracted from the separation between consecutive photoelectron peaks. In particular, the gain is proportional to the distance between the centroids of the 1 p.e. peak and the pedestal:

$$G \propto \mu_{1\text{ p.e.}} - \mu_{\text{pedestal}}, \tag{5}$$

where $\mu_{1\text{ p.e.}}$ and μ_{pedestal} are the mean values obtained from a fit to the charge spectrum.

More generally, the charge associated with the n-photoelectron peak can be modeled as:

$$Q_n = Q_0 + n \cdot G, \tag{6}$$

where Q_0 is the pedestal charge and G is the SiPM gain expressed in integrated charge units. Therefore, the gain can be extracted as the slope of a linear fit to the peak positions as a function of the photoelectron multiplicity (see Fig. 17, right).

In practice, the width of the peaks is determined by several effects, including electronic noise, gain fluctuations, baseline uncertainty, and correlated noise. In particular, optical cross-talk produces an excess of events at higher photoelectron multiplicities, while afterpulsing modifies the time structure of the signals.

The ability to resolve individual photoelectron peaks is one of the main advantages of SiPMs and plays a central role in their calibration and characterization.

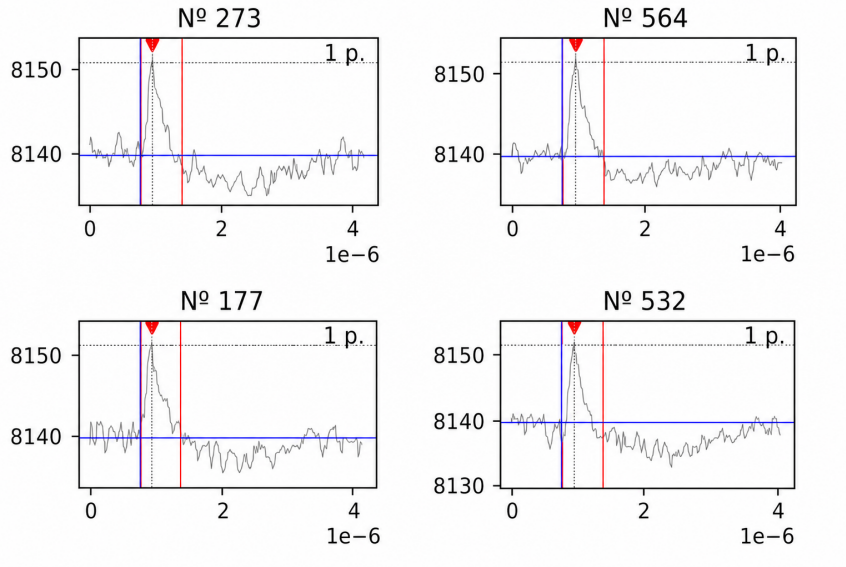


Figure 18: Example of four different single SiPM pulses generated by dark noise and recorded in MASSIBO.

6.3.3 Dark noise and correlated noise

Even in the absence of external illumination, SiPMs produce spontaneous signals (see Figure 18) generated by intrinsic noise processes. These contributions are particularly important in low-background and low-light applications, where they can limit the detector performance and affect the calibration procedure.

The dominant source of uncorrelated noise is the *dark count rate* (DCR), defined as the rate of pulses produced in the absence of incident light. Dark counts are mainly generated by thermally produced carriers or by carrier tunneling processes inside the silicon. At cryogenic temperature, thermal generation is strongly suppressed and the DCR is expected to decrease significantly.

Experimentally, the DCR is measured by counting the number of events above a given threshold during a time interval T in the absence of light:

$$\mathrm{DCR} = \frac{N_{\mathrm{events}}}{T}. \tag{7}$$

In an ideal detector, dark counts would follow Poisson statistics and would therefore appear as isolated events randomly distributed in time. However, additional correlated noise mechanisms modify both the amplitude and time structure of the signals.

One important correlated process is *optical cross-talk*. During an avalanche, accelerated carriers can emit secondary photons which may trigger avalanches in neighboring microcells. Since this process occurs almost instantaneously, cross-talk produces signals with amplitudes corresponding to multiple photoelectrons, even when only a single primary carrier initiated the event.

Another important effect is *afterpulsing*. During the avalanche process, some carriers may become trapped in defects within the silicon lattice and released after a delay. If the microcell has partially or fully recovered when the trapped carrier is released, a secondary avalanche can be triggered. Afterpulsing therefore introduces delayed correlated signals, typically on timescales ranging from tens of nanoseconds to several microseconds.

The analysis of the time difference between consecutive events,

$$\Delta t_i = t_{i+1} - t_i, \tag{8}$$

provides a powerful tool to distinguish different noise contributions. Uncorrelated dark counts produce an approximately exponential distribution in Δt, while correlated processes introduce excesses at short time intervals.

In addition to these standard noise sources, SiPMs operated at cryogenic temperature may exhibit more complex non-Poissonian behavior, including burst-like structures consisting of large numbers of correlated events clustered in time. Such effects have been observed in dedicated characterization campaigns for the DUNE photon detection system [2] and represent an important example of realistic detector behavior beyond idealized models.

6.4 Experimental setup

The measurements in this exercise are performed using MASSIBO, a cryogenic test stand developed for the large-scale characterization of SiPMs in the context of the DUNE photon detection system [2]. During the laboratory session, the students will be introduced to the setup and its operation, including the cryogenic infrastructure, readout electronics, and data acquisition system.

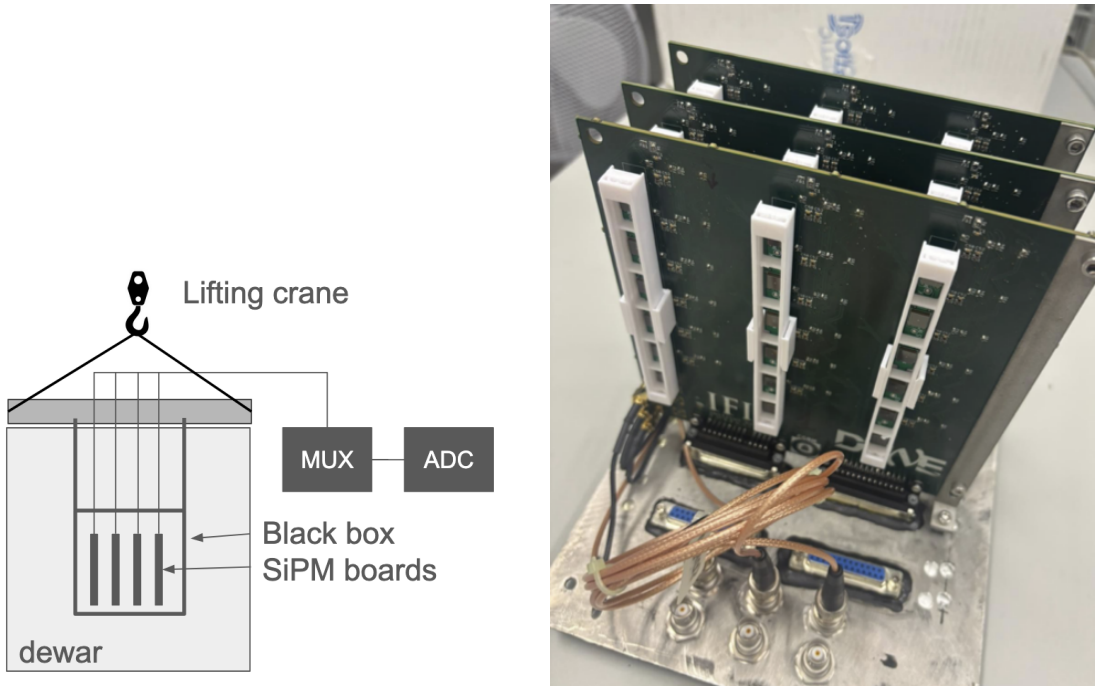


Figure 19: Overview of the MASSIBO cryogenic characterization setup (left) and detail of the cryogenic electronics and SiPM carrier boards (right).

6.4.1 Cryogenic and mechanical setup.

The sensors are operated at liquid-nitrogen temperature (77 K) inside a stainless-steel dewar equipped with a custom light-tight lid. The SiPMs are mounted inside a black aluminium enclosure suspended from the lid and immersed in the cryogenic environment.

This enclosure provides several important functions:

- protection against external light,
- reduction of thermal shock during immersion,
- mitigation of humidity condensation during warm-up.

The SiPMs are mounted on dedicated carrier boards and enclosed within individual optical shields designed to minimize inter-channel optical cross-talk and environmental effects associated with the liquid nitrogen volume.

The cooldown procedure is performed gradually by slowly lowering the instrumented structure into the liquid nitrogen bath. This controlled thermalization improves the reproducibility and stability of the measurements.

6.4.2 Readout electronics and DAQ

Each SiPM is connected to a cryogenic amplification stage designed to operate at liquid-nitrogen temperature. Since the SiPM behaves as a current source, the front-end electronics employ a transimpedance amplifier (TIA) to convert the sensor current into a voltage signal suitable for digitization.

A second amplification stage converts the signal into differential form in order to reduce external electromagnetic noise pickup during signal transmission from the cryogenic volume to the room-temperature electronics.

The system is designed for the parallel characterization of multiple sensors. A multiplexing stage allows different SiPM channels to be selected sequentially while sharing the same digitization chain.

Signal digitization is performed using the DAPHNE V1 board, an FPGA-based digitizer developed for the DUNE experiment [2]. The waveforms are acquired using self-trigger mode with configurable thresholds.

The digitized data consist of collections of triggered events. Each event contains:

- a timestamp identifying the trigger time,
- a waveform of 251 ADC samples.

The sampling frequency is 62.5 MHz, corresponding to a sampling interval of 16 ns and a total waveform duration of approximately 4 μs.

The data used in this exercise correspond to pre-recorded measurements acquired under dark conditions. The datasets are provided in NumPy format and analysed offline using Python and SciPy.

Additional details concerning the mechanical design, cryogenic operation, electronics architecture, multiplexing strategy, and DAQ implementation can be found in Ref. [2].

6.5 Procedure

The laboratory session combines a hands-on introduction to the MASSIBO characterization setup with the offline analysis of previously acquired datasets.

6.5.1 Setup preparation and cooldown.

At the beginning of the session, the students will participate in the preparation of the experimental setup. This includes mounting a set of SiPMs on the carrier boards, installing the sensors inside the cryogenic enclosure, and preparing the system for operation.

Once the setup is assembled, the instrumented structure will be gradually immersed into the liquid-nitrogen bath. During this stage, the students will be introduced to the operation of the cryogenic system, the front-end electronics, and the DAQ infrastructure.

After the cooldown process, the acquisition system will be configured and a dark-noise data-taking run will be started using self-trigger mode.

6.5.2 Offline data analysis.

Since a complete acquisition cycle typically lasts several hours, the datasets analysed during the laboratory session will correspond to measurements acquired previously under equivalent operating conditions.

The students will analyse the data using a Python-based analysis framework developed for this exercise. The software package, together with installation instructions and analysis examples, will be provided through a public GitHub repository:

`https://github.com/ific-neutrinos/TIDPAN2026.git`

The analysis workflow includes:

- waveform visualization and baseline estimation,
- charge reconstruction and gain extraction,
- pulse finding,
- time-difference analysis,
- estimation of DCR, cross-talk, and afterpulsing probabilities.

The students are encouraged to inspect and modify the analysis code in order to explore the impact of different analysis choices and selection criteria on the extracted detector parameters.

6.5.3 Software requirements

The analysis software used in this exercise is written in Python and relies on standard scientific libraries, including NumPy, SciPy, and Matplotlib. Students are expected to have access to a working Python environment on their laptops.

Detailed installation instructions and example notebooks are provided in the GitHub repository associated with the exercise.

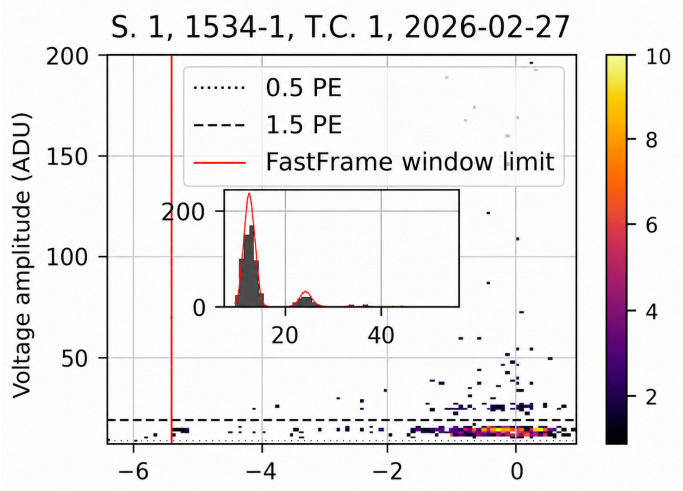


Figure 20: Example of a charge spectrum as a function of the time difference between consecutive events.

6.6 Expected results and discussion

A successful analysis of the provided datasets should allow the students to resolve individual photoelectron peaks in the reconstructed charge spectrum and extract the SiPM gain from their separation (as shown in Figs. 17 and 18).

Typical charge spectra acquired at cryogenic temperature exhibit a well-defined pedestal peak together with clearly separated 1 p.e. and, in some cases, higher-order photoelectron peaks. The quality of the peak separation depends on several factors, including electronic noise, gain fluctuations, and the choice of the waveform integration window.

The analysis of the time distribution between consecutive pulses should reveal an approximately exponential component associated with uncorrelated dark counts, together with excess populations at short time intervals produced by correlated noise mechanisms such as afterpulsing (see Figure 20).

Depending on the dataset, additional non-Poissonian structures may also be observed, including burst-like sequences of correlated pulses. These effects illustrate the complexity of realistic detector behavior and the importance of robust analysis criteria in large-scale characterization campaigns.

The extracted values of gain, dark count rate, cross-talk probability, and afterpulsing probability should be critically compared with the expected behavior of cryogenic SiPMs and interpreted in the context of the detector requirements of large-scale experiments such as DUNE.

6.7 Safety considerations

The operation of cryogenic systems involves several safety risks associated with liquid nitrogen, including cold burns, oxygen displacement, and overpressure hazards. During the laboratory session, students must follow the instructions provided by the instructors at all times.

Particular attention should be paid to:

- the use of appropriate personal protective equipment (PPE),

- safe handling of cryogenic liquids,
- avoiding direct contact with cold surfaces,
- proper ventilation of the laboratory area.

The setup also includes low-voltage and high-voltage electronic components that must not be manipulated without instructor supervision.

References

[1] A. Ghassemi *et al.*, *Technical Guide to Silicon Photomultipliers (MPPC)*, Hamamatsu Photonics K.K., 2022.

[2] M. Andreotti *et al.*, *Cryogenic characterization of Hamamatsu HWB MPPCs for the DUNE photon detection system*, JINST **19** (2024) T01007.

[3] C. Benítez *et al.*, *A system for the mass characterization of Hamamatsu SiPMs at cryogenic temperatures with multiplexed parallel readout*, in preparation (2026).

Exercise 7 Basic Principles of Silicon Microstrip Detectors

7.1 Objectives

- The general objective of this exercise is to understand the basic operating principles of silicon microstrip tracking detectors, including charge collection and its dependence on the bias voltage, charge sharing between adjacent strips, and the determination of the spatial resolution.
- Specific objectives:
  - Observe the noise of a silicon strip detector as a function of the bias voltage.
  - Observe the signal spectra produced by a minimum ionizing particle or a laser,
  - Determine the charge collection efficiency of a silicon detector.
  - Determine the full depletion voltage.
  - Observe charge sharing between strips and determination of the spatial resolution.
  - Illustrate the structure of a typical microstrip detector.

7.2 Professors

- Carmen García
- Carlos Escobar

7.3 Theoretical Introduction

7.3.1 Semiconductors and P-N junctions

Semiconductors are materials with electrical properties between those of conductors and insulators; they are neither good conductors nor good insulators. A key characteristic is that their conductivity can be controlled through doping, the intentional addition of impurities.

In terms of energy bands, semiconductors have a valence band and a conduction band separated by a bandgap of a few electronvolts. If the bandgap is larger than about 4 eV, the material behaves as an insulator. Electrical current is carried by free electrons in the conduction band and by holes in the valence band. When an electron gains enough thermal energy or is introduced through doping, it can move to the conduction band, leaving behind a hole that acts as a positively charged carrier.

In intrinsic semiconductors, electron–hole pairs are generated thermally and recombine continuously, reaching an equilibrium concentration. In extrinsic semiconductors, doping determines the dominant carriers. Silicon, which has four valence electrons, can be doped with group V elements (such as arsenic), providing an extra electron and creating n-type material. Doping with group III elements (such as boron) creates holes, producing p-type material. These impurities introduce donor

or acceptor energy levels within the bandgap, facilitating carrier excitation at room temperature.

Carrier transport occurs through drift and diffusion. Under an applied electric field (V), electrons and holes acquire a drift velocity proportional to the field, determined by their mobilities. The current density depends on carrier concentration and mobility, while diffusion arises from concentration gradients and is also related to mobility.

When p-type and n-type materials are joined, a p–n junction is formed. Due to carrier diffusion across the junction, electrons and holes recombine near the interface, creating a depletion region free of mobile carriers. This region contains fixed charges that produce an internal electric field and a built-in potential. The depletion layer also exhibits a capacitance that depends on its width and the material properties, according to the following equation:

$$d = \sqrt{\frac{2\varepsilon(V_{bi} + V)}{qN_{\text{eff}}}}$$

where ε is the silicon dielectric constant, V_{bi} is the build-in voltage (typically the applied reverse bias $V >> V_{bi}$, so V_{bi} is often neglected) and N_{eff} is the doping effective concentration given by:

$$N_{\text{eff}} = \frac{N_D N_A}{N_D + N_A}$$

N_D and N_A are the donor and acceptor impurity concentrations, respectively. The capacitance can be calculated as a function of the applied voltage.

Applying a reverse-bias voltage increases the width of the depletion region. As the voltage rises, the material can become fully depleted at a specific depletion voltage (V_{dep}). Beyond this point, the capacitance remains constant, and the electric field is approximately uniform. When the applied voltage is below full depletion, the field does not extend over the entire bulk. Its maximum remains at the junction, so only a fraction of the sensor is depleted. Consequently, charge collection is inefficient in this case, as carriers outside the electric field do not move effectively.

Since a minimum ionizing particle deposits energy approximately uniformly along its path in silicon, the collected charge is proportional to the depleted thickness of the detector. Full depletion is important in silicon detectors to ensure efficient charge collection. For a detector of thickness D, the depletion width as a function of the applied voltage can be written as:

$$d_c = D\sqrt{\frac{V}{V_{dep}}}$$

Thus, the efficiency of a silicon detector is given by:

$$\begin{cases} \sqrt{V/V_{\text{dep}}} & \text{for } 0 \leq V \leq V_{\text{dep}} \\ 1 & \text{for } V > V_{\text{dep}} \end{cases}$$

7.3.2 Silicon Microstrip detector

Silicon detectors are based on one or more p–n junctions.. Starting from a planar structure of doped silicon, traditionally n-type, one of the surfaces is doped with the

opposite type, traditionally p-type. This implanted layer is much more highly doped than the bulk and is very thin compared to the detector thickness.

A silicon micro-strip detector is built by segmenting the top layer into strips. Figure 21 shows a schematic of the micro-strip detector used in the ATLAS experiment.

P-on-n strip detectors typically consist of an n-type silicon bulk about 300 μm. On one side, a thin layer is segmented into strips of p-type material, while the back implant is n+. Abrupt junctions are formed between the strips and the bulk silicon. The depletion zone grows from the strips toward the backplane, so that the device can operate partially depleted, with the movement of holes generating the signal.

The silicon sensor requires several additional elements for proper operation. An oxide layer (1–4 μm) above the implanted strips electrically isolates the detector surface from the readout electronics. Each strip is capacitively coupled to a metal (Al) strip above the p implants, where the signal is read out. A common bias rail connects the front strips via polysilicon resistors to apply the bias voltage. On the back, a low-resistance ohmic contact with an n^+ implant and metal layer allows the application of high voltage while preventing the depletion region from reaching the metallization. Finally, the guard ring structure minimizes edge leakage current and prevents electrical breakdown, compensating for crystal damage at the cutting edges that could dramatically increase leakage. These features can be seen in the photograph of the silicon micro-strip detector shown in Figure 22.

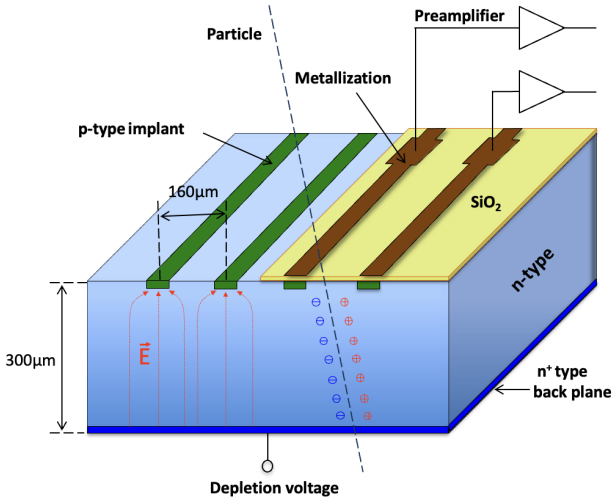


Figure 21: Sketch of a micro-strip p-on-n silicon detector.

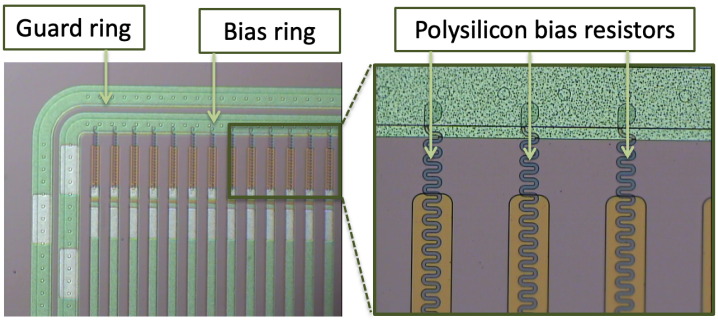


Figure 22: Sensor view showing the guard ring, the bias ring, and polysilicon resistors.

A minimum ionizing particle (MIP) traversing a silicon detector primarily loses energy through ionization. As it passes through the silicon lattice, it liberates valence electrons, creating electron-hole pairs, while occasional interactions with more tightly bound electrons produce high-energy secondary ionization, leading to rare, large energy deposits. These rare events cause the deposited energy to follow a non-symmetric Landau distribution, often modeled as a Landau curve convoluted with a Gaussian.

Because of this asymmetry, the mean energy loss is higher than the most probable energy loss. For a 300 μm thick silicon detector, a MIP deposits about 116 keV on average (mean) but 81 keV most probably, generating roughly 32,000 electrons (mean) or 23,000 electrons (most probable, 3.6 fC). The most probable energy loss is typically used to estimate the collected charge.

7.3.3 Noise

The output signal of a silicon sensor is a short current pulse, proportional to the particle energy deposition, with a duration of approximately 0.1–30 ns for a 10–500 μm thick sensor.

In the EASY system, the signal is read out by the BEETLE chip, originally designed for the LHCb experiment. Consequently, the chip operates under Large Hadron Collider (LHC) conditions, where beam crossings occur every 25 ns.

The signal generated in silicon detectors is generally of small amplitude (around 24,000 electrons); therefore, amplification and shaping stages are required for further processing. These stages are integrated into the BEETLE chip.

A charge-sensitive preamplifier is typically used, which avoids any dependence on changes in the detector capacitance with temperature. Inevitably, noise is introduced by the readout electronics, affecting the charge measurement. The total noise of the system is measured as an equivalent noise charge (ENC) at the input of the amplifier. The input charge that produces a signal amplitude equal to the RMS noise, ENC, at the output is given by:

$$\mathrm{ENC}_{\mathrm{TOT}}^2 = \mathrm{ENC}_{\mathrm{PA}}^2 + \mathrm{ENC}_{\mathrm{I}}^2 + \mathrm{ENC}_{\mathrm{RP}}^2 + \mathrm{ENC}_{\mathrm{RS}}^2$$

where $\mathrm{ENC}_{\mathrm{PA}}$ is the preamplifier noise, which depends on the capacitive load at the input of the preamplifier (C_{load}). $\mathrm{ENC}_{\mathrm{I}}$ is due to the detector leakage current I. $\mathrm{ENC}_{\mathrm{RP}}$ is the thermal noise from the parallel resistor. Finally, $\mathrm{ENC}_{\mathrm{RS}}$ is the noise from the series resistance in the circuit. The two main noise sources that dominate in silicon detectors are the detector capacitances ($\mathrm{ENC}_{\mathrm{PA}}$) and the leakage currents ($\mathrm{ENC}_{\mathrm{I}}$).

7.3.4 Signal

Before analysing the data, the student should understand how the signal is extracted from the raw data, and therefore, be familiar with the concepts of pedestal, noise (and its dependence on the reverse bias voltage), and common-mode noise.

The signal collected by each strip is digitized by an Analog-to-Digital Converter (ADC) and expressed in ADC counts. The pedestal of the ADC output is an effective offset, defined as the ADC count measured in the absence of input signal. In the EASY system, this offset is set near mid-scale (about 500 ADC counts) to allow the readout of both positive and negative signals, depending on the sensor type. When a particle passes through the detector, the trigger initiates the acquisition, thereby defining an event. The corresponding ADC count (raw data) can be decomposed into three components and expressed as follows:

$$ADC_{i,k} = P_{i,k} + D_k + S_{i,k}$$

where i is the strip number and k is the event number. $P_{i,k}$ is the pedestal, which fluctuates around a baseline. These fluctuations are due to electronic non-uniformities in each strip. D_k is the common-mode shift, resulting from electronic parasitic pick-up affecting all channels of the chip, it is specific to the chip and varies with time, fluctuating from event to event. $S_{i,k}$ represents the signal, the collected charge plus the residual noise.

When no signal is present (as in a pedestal run) $S_{i,k} = 0$, the pedestals for strip i is computed as the mean ADC value of N events, since the average common noise is then zero.

$$P_i = \frac{1}{N} \sum_{k=1}^{N} ADC_{i,k}$$

Known the pedestal for each strip P_i, the common-mode shift for a given event k is calculated as the mean ADC charge of the 128 channels after pedestal subtraction:

$$D_k = \frac{1}{128} \sum_{i=1}^{128} (ADC_{i,k} - P_i)$$

Finally, the electronic noise of each channel is defined as the fluctuation of the pedestal after common noise subtraction. That is, the standard deviation (rms) of the pedestal after correcting for each event for the common noise, $P_i^c = P_i - D_k$:

$$N_i = \sqrt{\frac{1}{N-1} \sum_{k=1}^{N} \left(P_{i,k}^c - \overline{P_i^c}\right)^2}$$

The electronic noise depends on the capacitance at the input of the preamplifier, the shaping time, the resistances of the detector and the readout chip, the temperature, and the leakage current. In this experiment, since the shaping time and temperature remain constant, the electronic noise mainly depends on the reverse bias voltage through its effect on the detector capacitance and leakage current.

7.3.5 Spatial resolution

The signal pulse collected by the strips is amplified and integrated by the readout electronics in order to measure the total charge deposited in the microstrip detector. Since all strips are read out, the strip receiving the signal provides information about one coordinate of the traversing particle. Therefore, a microstrip detector is a position-sensitive device and can be used as a tracking detector.

The spatial resolution of a silicon microstrip detector is determined by several factors, including the strip pitch, the diffusion of the charge carriers in the silicon, and the sharing of charge between adjacent strips.

If only the information of the strip collecting the largest signal is used (binary readout), the particle position is assigned to the centre of that strip. In this case, the difference between the reconstructed position and the true particle position follows a uniform distribution within one strip pitch p. The corresponding spatial resolution is therefore

$$\sigma = \frac{p}{\sqrt{12}}.$$

In practice, the charge generated by the particle can be shared between neighbouring strips due to carrier diffusion and the geometry of the electric field inside

the detector. When this occurs, the position of the particle can be reconstructed more precisely by using the charge measured in several strips.

A common method is the centre-of-gravity algorithm (or η-algorithm), and the reconstructed position can be calculated as

$$x_R = x_L + \frac{\mathrm{ADC}_R}{\mathrm{ADC}_R + \mathrm{ADC}_L}\, p$$

where ADC_R and ADC_L are the charges induced in the strips to the right and left of the incident position, respectively, and x_L is the position of the left adjacent strip. The spatial resolution dependents on the ADC/Noise ratio, and the detector pitch according to the approximation, for a lineal distribution of charge

$$\sigma \approx \frac{p}{\mathrm{ADC/Noise}}$$

7.4 Experimental Setup

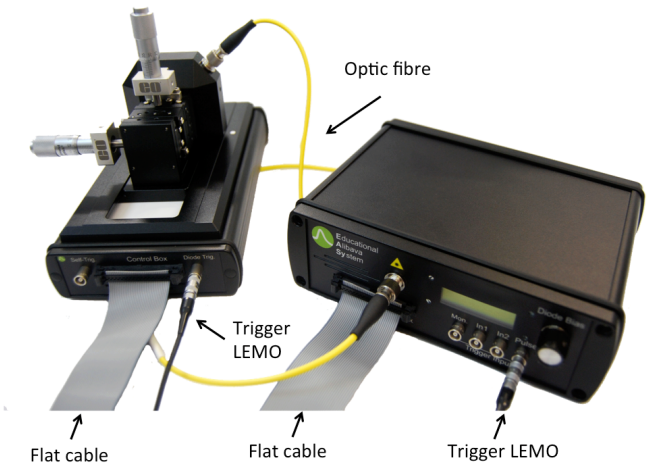


Figure 23: EASY system with the Control Unit and the Sensor Unit.

The experimental system, called the Alibava Educational System (EASY [2]), is a complete instrumentation setup dedicated to silicon micro-strip radiation detectors.

The system can operate either with a laser source or with radioactive sources. The components of the EASY system are the Control Unit (the system's main board) and the Sensor Unit (which contains the detector board). The full system is shown in Figure 23.

The Sensor Unit incorporates an electronic board, shown in Figure 24, designed to accommodate the BEETLE readout chip (ASIC). This chip provides 128 analogue input channels with a 40 MHz clocked analogue pipeline and a maximum programmable latency of 4 μs. The BEETLE has a dynamic range of $\pm$100,000 electrons (electron-equivalent charge). The system also incorporates an Analog-to-Digital Converter (ADC) that digitizes the pulses into arbitrary ADC counts and Time Digital Converter (TDC) with a range of 100ns and a system clock with a period of 25 ns.

The unit also generates a trigger signal to indicate that a particle has crossed the microstrip sensor and to trigger the readout . There are two trigger modes: diode mode and self-trigger mode. The diode trigger mode consists of discriminating a signal generated by a low-noise silicon diode. The silicon diode is mounted underneath the microstrip sensor, separated by a gap of a few millimetres.

The system incorporates a 980 nm laser (5 nm pulse width) to induce charge in the detector. The laser has a peak power of 0.6 mW and a spot size of approximately

[2] https://alibavasystems.com/producto/easy/

20 μm. The laser is housed in the Control Unit and connected to the detector board via an optical fibre.

The system also includes dedicated data acquisition software and data analysis tools (*Alibava-gui*) to demonstrate and visualize the operating principles of the silicon strip detector. There are five run types:

Calibration: The software programs the BEETLE chips to inject calibration pulses into all channels to characterize the ASIC response and determine its gain.

Laser Synchronization: A trigger pulse is sent to the laser while scanning the delay between the laser and the acquisition in order to determine the optimal sampling point.

Pedestal Run: An internal trigger is generated to measure the pedestal baseline and its fluctuations (noise).

Laser Run: Charge is injected into the detector using a laser. Prior laser synchronization is required to ensure optimal signal readout.

Radioactive Source Run: The acquisition is triggered by signals above a predefined threshold, suitable for radioactive source or cosmic-ray measurements.

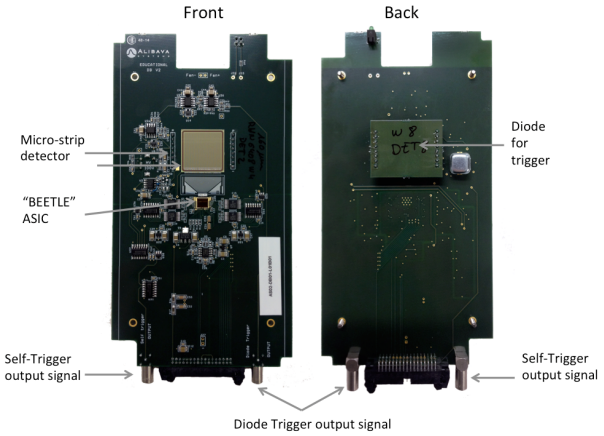


Figure 24: Sensor Unit holding the micro-strip detector and the BEETLE in the front side and the Diode for trigger purposes in the back. The board has two connectors for the trigger signal: Diode Trigger and Self-Trigger.

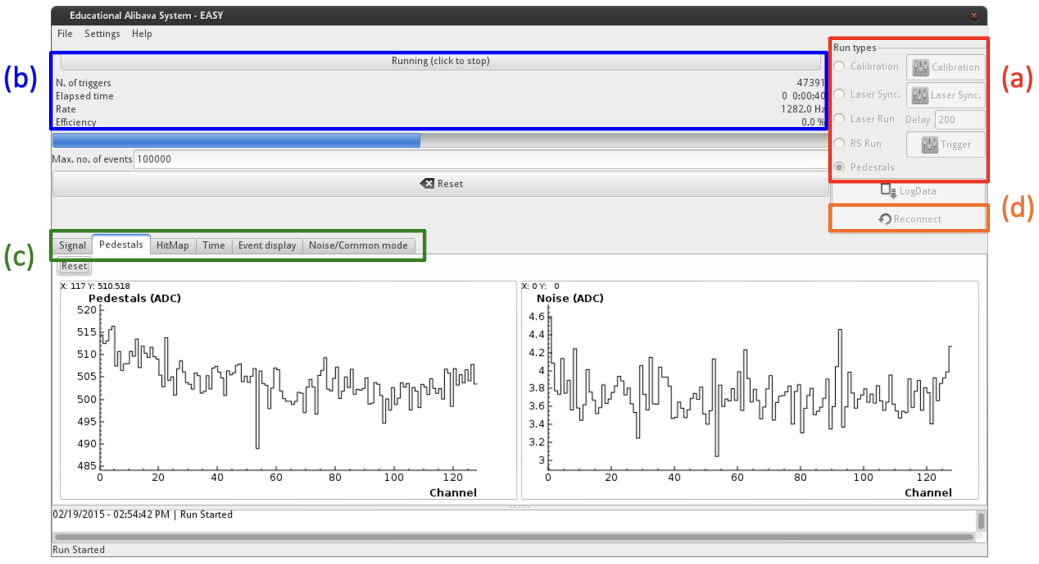


Figure 25: Main graphical user interface of the Alibava-gui during a pedestal run.

When the *Alibava-gui* is launched, the main window shown in Figure 25 is displayed. In area (a), the run type is selected, while area (b) contains the *Start* and *Stop* buttons used to control the acquisition. This section also displays the number

of events, the elapsed time, the acquisition rate, and the acquisition efficiency (i.e., the fraction of events that satisfy the analysis criteria defined in the Analysis dialog).

Area (c) comprises a set of tabs for monitoring the data during acquisition: *Signal, Pedestal, HitMap, Temperature, Time* and *Noise/Common* Mode. The *LogData* button opens a dialog window that allows the user to save the data to a file; the output file must be defined before starting the run.

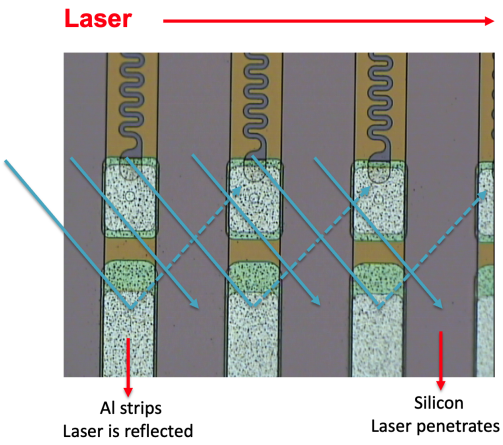


Figure 26: Laser movement across the sensor strips.

7.5 Procedure

The exercise consists of two parts. The first aims to study how the charge collection varies with the depletion voltage and to determine the full depletion voltage. The second part focuses on how the charge is shared between adjacent strips and on determining the spatial resolution of the detector.

The first steps consist of evaluating the pedestals and noise levels, as well as performing the laser synchronization. This part is common to both exercises and is carried out only at the beginning.

7.5.1 Pedestal, electronic noise and common-mode noise study

- Connect the flat cable between the Control Unit and the Sensor Unit, and the USB cable between the computer and the Control Unit. Then power on the Control Unit.
- Start the *Alibava-gui*.
- Select *Pedestal* in the Run Type window and start a run. Observe the pedestal (*Pedestal* window) and noise histograms (*Noise/Common Mode* window).
- Slowly increase the sensor reverse bias voltage and observe the effect on the pedestals and the noise. It is important to increase the first 10 V very slowly in order to clearly see the effect.
- Take a pedestal run and save it for further analysis:
  - Stop the previous pedestal run.
  - Set the sensor reverse bias voltage to 130 V.
  - Select the output file to log the raw data using *LogData* in the main window and start a pedestal run.

7.5.2 Laser Synchronization

- Connect the laser optical fibre between the Control Unit and the Sensor Unit.

- Set the sensor reverse bias voltage to 130 V.
- Select *Laser Sync* in the Run Type window and start a run.
- Observe the charge deposition as a function of time in the *Signal* window and adjust the laser delay (*Delay* in the Run Type window) to the time at which the laser signal reaches its maximum.

7.5.3 Charge collection using laser light

- Set the sensor reverse bias voltage to 10 V.
- Select the output file to log the raw data using *LogData* in the main window.
- Start a laser run.
- Repeat the previous steps, increasing the reverse bias voltage from 10 V to 200 V in 10 V increments. Save the data for further analysis (always before starting the run).
- Observe how the signal changes with voltage in the *Signal* window while taking data.

7.5.4 Inter-strip charge sharing

- Set the sensor reverse bias voltage to 130 V.
- Take several laser runs (remember to save the data before starting each run), moving the laser in 10 μm steps perpendicular to the strip direction (using the horizontal screw). Cover at least a distance of 350 μm. The laser will move as shown in Figure 26. Note that the laser is reflected when it strikes the aluminum of a strip and therefore does not generate charge, whereas it generates charge when it strikes the region between strips.
- Observe how the signal changes in the *HitMap* window while taking data.

7.6 Analysis and interpretation of the results

For the data analysis, the student will need a personal computer and a Google Drive account. Then:

- Copy this notebook [3] to your Google Drive account *(File → Save a copy in Drive)*.
- Go to your *Drive* and open the copied file located in your *My Drive* directory.
- Execute the first cell to mount your Drive at */content/gdrive/*.
- Execute the second cell to create the folder *My Drive/CNID-EASy/Laser* and verify that it has been created correctly.

[3] `https://colab.research.google.com/drive/15jmZqeMP44Ga_x9Y5N5RjeVlhDnTwlPt?usp=drive_link`

- Upload your data to *My Drive/CNID-EASy/Laser*.

Figure 27 shows the general flow of the analysis program; you can navigate through it by opening and running the various cells.

7.6.1 Setting the signal-to-noise cut

- Run all cells until the *Options* cell.
- Select the input file.
- Set `options_seed_cut` to values between 1 and 10.
- Process the data and generate the plots.
- From the information in the plots, select the optimal `options_seed_cut` value to reduce the effect of noise.

7.6.2 Obtaining Charge Collection, Charge Collection Efficiency and Depletion Voltage

- Process the runs corresponding to different bias voltages by entering them one by one in the *Options* cell.
- Run the *Options*, *Read data*, and *Make the plots* cells.
- For each bias voltage, obtain the mean collected charge and its standard deviation.
- Fill the arrays in the section *Mean charge for Depletion voltage scan.*
- Produce the charge versus bias voltage plot and the Charge Collection Efficiency plot.
- Calculate the full depletion voltage.

7.6.3 Inter-strip charge sharing, sensor parameters and spatial resolution

Run the loop over the data files corresponding to different laser incidence positions at a fixed depletion voltage. Identify the strips with signal and use their strip numbers as input parameters of the program in order to obtain the correct plot of charge versus strip number. Answer the following questions:

- In the *Options* cell, set `options_file_prefix` to the prefix of your file names (e.g. "Laser") and define the number of input files using `options_n_files`. Your files should therefore be named from `Laser-1.h5` to `Laser-n.h5`, with $n =$ `options_n_files` $- 1$.
- Run the *Options* cell to set the parameters.
- From the plots, identify the channels with signal and set the parameters `options_chan_min` and `options_chan_max` in the *Options* cell.
- Run the *Options* cell again.

- Run the *Charge sharing* cell to obtain the plot of mean charge versus channel number for the strips defined by `options_chan_min` and `options_chan_max`.
- Describe and interpret the plot.
- Calculate:
  - The strip width.
  - The sensor pitch.
  - The size of the laser beam.
- Determine the spatial resolution for a binary readout.
- Propose a method to improve the spatial resolution using the charge-sharing information and determine the resulting spatial resolution.

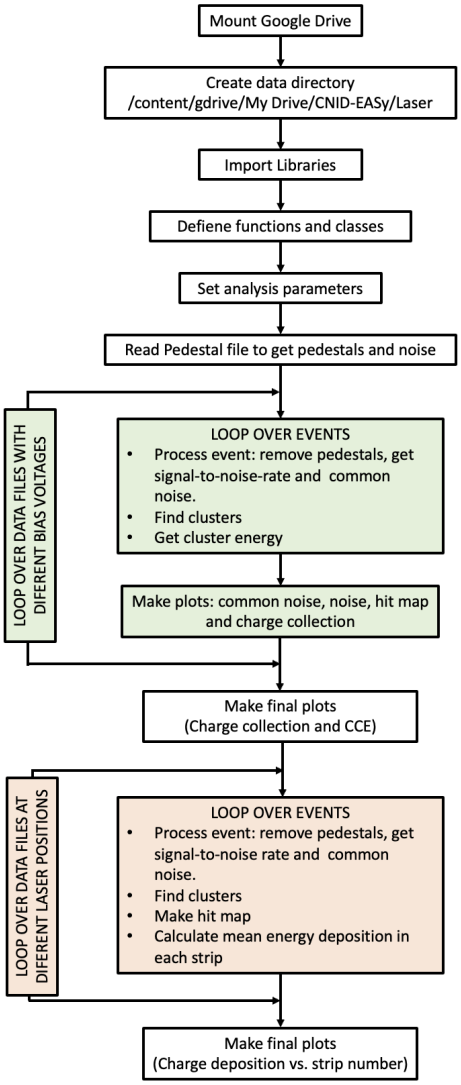


Figure 27: Flowchart of the analysis program

Exercise 8 Transient Current Technique: Measurement of Charge Carrier Mobility in Silicon

8.1 Objectives

- The general objective of this exercise is to understand the Transient Current Technique (TCT) as a method to study charge transport phenomena in silicon detectors.

- Specific objectives:

  - Understand how electron–hole pairs are generated in silicon using short laser pulses.
  - Understand signal formation in silicon detectors using the Shockley–Ramo theorem.
  - Observe transient current signals induced by drifting charge carriers.
  - Study the dependence of carrier drift velocity on the applied electric field.
  - Determine the charge carrier mobility from time-resolved current measurements.

8.2 Professors

- Neil Moffat

- Carlos Marinas

8.3 Theoretical Introduction

8.3.1 Charge generation in silicon

Silicon detectors operate as reverse-biased p–n junctions. When energy is deposited in the silicon bulk, electron–hole pairs are generated. On average, one electron–hole pair is created for every 3.6 eV of deposited energy. For ionizing radiation, this results in approximately 80 electron–hole pairs per micrometer of silicon.

In this experiment, charge carriers are generated optically using a pulsed blue laser with a wavelength of 404 nm. Since the photon energy exceeds the silicon band gap of 1.12 eV, absorption of a photon produces a single electron–hole pair (Single Photon Absorption). Unlike charged particles, photons are absorbed at a well-defined depth that depends on the wavelength.

Blue light has a very high absorption coefficient in silicon and therefore a very short penetration depth. As a consequence, 404 nm light generates charge carriers close to the illuminated surface of the detector. This provides localized surface injection and a well-defined starting position for the drifting charge carriers.

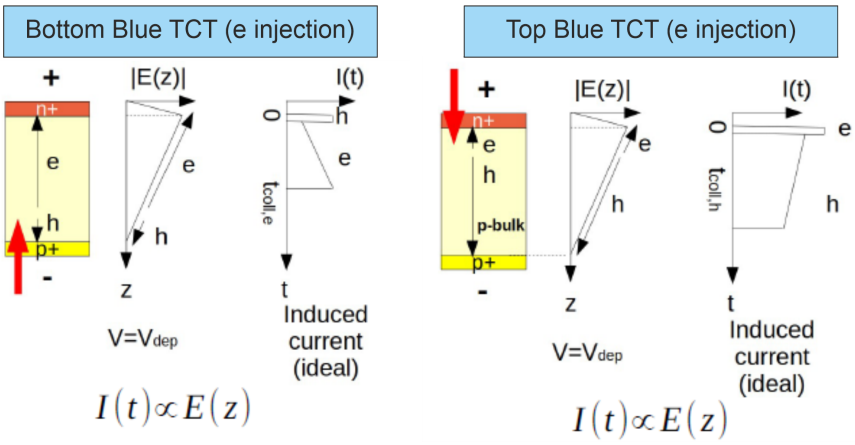


Figure 28: Blue top/bottom allows to "decouple" electrons from holes. Mobility differences can be measured experimentally.

8.3.2 Electric field and charge transport

When a reverse bias voltage is applied to a silicon detector, an electric field is established across the depleted region. Charge carriers move under the combined effect of drift and diffusion. In the presence of a sufficiently strong electric field, drift dominates the transport.

The average drift velocity v of charge carriers is related to the electric field E by the mobility μ:

$$v = \mu(E, T)\, E$$

where μ depends on the electric field strength and temperature. At low electric fields, the drift velocity increases linearly with E, while at high electric fields it approaches a saturation value due to increased scattering.

Typical room-temperature low-field mobility values in silicon are approximately:

$$\mu_e \approx 1400 \text{ cm}^2/\text{Vs}, \qquad \mu_h \approx 480 \text{ cm}^2/\text{Vs}$$

8.3.3 Signal formation and Shockley–Ramo theorem

As charge carriers drift inside the detector, they induce a current on the readout electrode. The induced current is described by the Shockley–Ramo theorem:

$$I(t) = q\, \vec{v}(t) \cdot \vec{E}_w(\vec{r}(t)),$$

where q is the carrier charge, $\vec{v}(t)$ is the drift velocity, and $\vec{E}_w$ is the weighting field, which depends only on the detector geometry.

A key result of the Shockley-Ramo theorem is that the induced current appears as soon as the carriers start moving and does not require them to reach the electrode. The time integral of the induced current corresponds to the induced charge, which depends on the initial and final carrier positions through the weighting potential.

For a planar pad detector, the weighting field is approximately uniform, and the transient current signal directly reflects the carrier drift velocity. This makes the Transient Current Technique a powerful method to study charge transport properties.

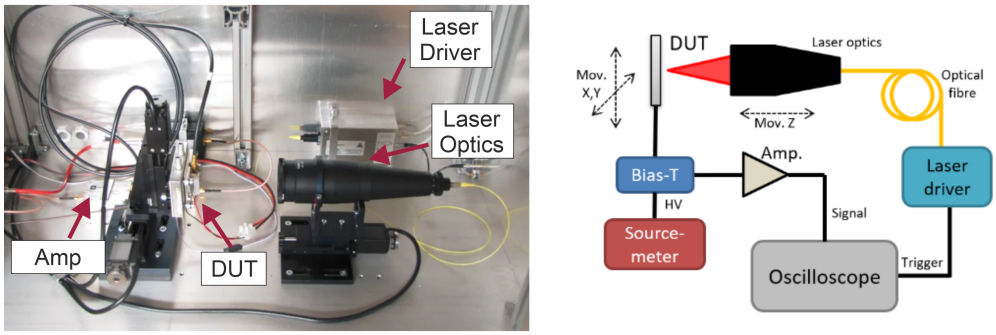


Figure 29: Schematic of the TCT setup

8.3.4 Transient Current Technique

In the Transient Current Technique, charge carriers are generated using very short laser pulses whose duration is much shorter than the carrier drift time. The generated charge cloud can therefore be considered instantaneous.

The duration and shape of the induced current pulse depend on the detector thickness, the applied electric field, and the carrier mobility. Typical pulse durations range from a few nanoseconds to several tens of nanoseconds.

The signal measured by the oscilloscope is a convolution of:

- the induced current generated by the drifting charge carriers,
- the temporal shape of the laser pulse,
- the transfer function of the readout electronics (detector capacitance, amplifier, bias-tee, oscilloscope bandwidth).

If the laser pulse is sufficiently short and the electronics bandwidth is adequate, the measured signal closely approximates the induced current.

8.4 Experimental Setup

The experimental setup consists of:

- a reverse-biased n on p silicon diode,
- a pulsed blue laser source with wavelength $\lambda = 404\,\text{nm}$,
- a high-voltage power supply for detector biasing,
- a current-sensitive amplifier and a bias-tee,
- a fast digital oscilloscope for waveform acquisition,
- optical components for laser delivery and focusing.

The laser is synchronized with the data acquisition system to allow time-resolved measurement of the transient current signals.

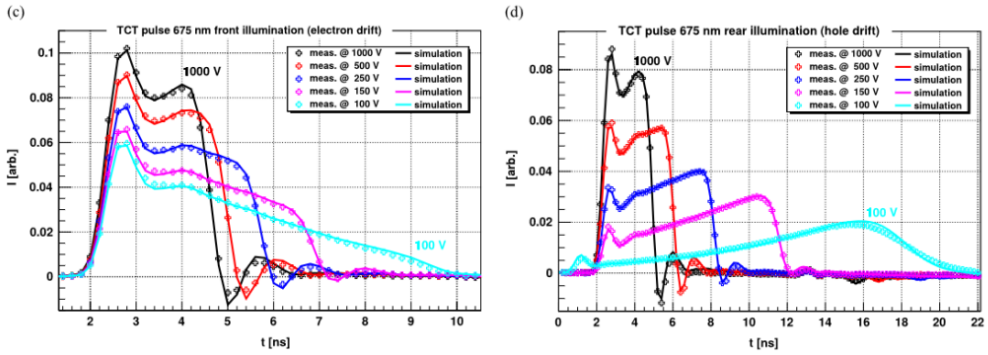


Figure 30: Typical waveforms produced from the injection of low penetrating light with front and back side illumination. Type injection is clear from the change in slope between front and back illumination. Taken from `https://indico.cern.ch/event/1466985/contributions/6379426/attachments/3024839/5338303/signal_formation.pdf`

8.5 Procedure

All measurements in this exercise are performed using 404 nm blue laser illumination in order to generate charge carriers close to the detector surface (Front and Back illumniation).

- Apply a reverse bias voltage to the silicon detector and verify stable operation.
- Acquire noise waveforms without laser illumination to characterize the electronic noise and baseline.
- Illuminate the detector with short blue laser pulses and observe the transient current signal on the oscilloscope.
- Record transient current waveforms for several values of the applied bias voltage.
- For each voltage, acquire multiple waveforms to allow averaging and noise reduction.
- This should be repeated for back side illumination and the resulting analysis

Expected result should be similar to that of the waveforms included in figure 30, where the hole and electron injection are clearly present depending on the illumination direction and applied bias voltage.

8.6 Analysis and interpretation of the results

8.6.1 Waveform extraction and noise treatment

Before extracting physical quantities, the raw waveforms must be processed. Noise waveforms acquired without laser illumination are used to determine the baseline and RMS noise. The baseline is calculated as the mean voltage in a pre-signal time window and subtracted from each signal waveform.

Signal waveforms recorded under identical conditions should be averaged to improve the signal-to-noise ratio.

The TCT data are stored in .tct file. A ROOT macro is provided to convert the stored waveforms into a plain text format (time–voltage pairs), allowing further analysis using Python or PyROOT. Students are free to choose the analysis environment, provided that the full analysis procedure is documented.

8.6.2 Mobility determination from transient signals

The carrier transit time t_{tr} is extracted from the averaged transient current waveform. This can be done, for example, by measuring the pulse duration above a fixed threshold or between two chosen fractions of the maximum pulse amplitude.

The average drift velocity is calculated as:

$$v = \frac{d}{t_{\mathrm{tr}}},$$

where d is the detector thickness.

Assuming an approximately uniform electric field:

$$E = \frac{V_{\mathrm{bias}}}{d},$$

the mobility is obtained from:

$$\mu = \frac{v}{E}.$$

8.6.3 Worked numerical example

Assume:

- Detector thickness: $d = 300\,\mu\mathrm{m}$
- Applied bias voltage: $V_{\mathrm{bias}} = 150\,\mathrm{V}$
- Measured transit time: $t_{\mathrm{tr}} = 6.0\,\mathrm{ns}$

The drift velocity is:

$$v = \frac{3.0 \times 10^{-4}}{6.0 \times 10^{-9}} = 5.0 \times 10^{4}\,\mathrm{m/s}$$

The electric field is:

$$E = \frac{150}{3.0 \times 10^{-4}} = 5.0 \times 10^{5}\,\mathrm{V/m}$$

The mobility is:

$$\mu = \frac{v}{E} = 1.0 \times 10^{-1}\,\mathrm{m^2/Vs} = 1000\,\mathrm{cm^2/Vs}$$

8.6.4 Drift velocity and mobility

Determine the carrier transit time from the pulse duration. Using the known detector thickness, calculate the average drift velocity. Plot the drift velocity as a function of the electric field and extract the carrier mobility from the linear region of the curve. To obtain a reliable mobility value:

- Repeat the analysis for several bias voltages.
- Plot the drift velocity as a function of the electric field.
- Extract the mobility from the slope of the linear region of the $v(E)$ curve.

Expected outcomes:

- representative noise and signal waveforms,
- the method used to determine the transit time,
- the drift velocity versus electric field plot,
- the extracted mobility and associated uncertainties,
- a comparison with reference values for silicon.

8.6.5 Discussion

Compare the obtained mobility values with typical literature values for silicon at room temperature. Discuss possible sources of systematic uncertainty, such as finite laser pulse width, electronic bandwidth, and assumptions regarding field uniformity.

Exercise 9 Pulse processing electronics. Study case: A SiPM readout Front End ASIC

9.1 Objectives

- The **general objective** of this laboratory is to provide students with a comprehensive understanding of the architecture and operation of front-end Application Specific Integrated Circuits (ASICs) used for pulse processing in particle and nuclear physics experiments, with a particular focus on the readout of Silicon Photomultipliers (SiPMs). Through hands-on interaction with a representative front-end ASIC, students will learn how analog detector signals are conditioned, digitized, and prepared for further data acquisition, and how such ASICs are configured and operated in a realistic experimental context.

  Specific objectives. By the end of this laboratory session, students will be able to:

- Identify and describe the main functional building blocks of a front-end ASIC, including preamplifiers, shapers, discriminators, and readout circuitry.

- Understand the signal processing chain from the SiPM output to the processed analog and digital signals, including charge amplification, shaping, discrimination, and timing extraction.

- Configure the ASIC operating parameters using control registers or configuration interfaces, and understand the impact of these parameters on detector performance.

- Gain practical experience with the operation and calibration of discriminators, including threshold scans, rate scans, and the interpretation of noise and signal distributions.

- Understand the role of analog shaping, shaping time selection, and baseline control, and analyze their effects on noise, signal amplitude, and rate capability.

- Perform basic signal acquisition tasks and analyze the acquired data to assess performance metrics such as noise level, trigger efficiency, and threshold dispersion.

- Develop familiarity with standard calibration and characterization procedures used in SiPM readout electronics, bridging the gap between theoretical concepts and practical implementation.

9.2 Professors

- Daniel Guberman

- David Mazzanti

- Eduardo Picatoste

- David Gascon

9.3 Theoretical Introduction

Modern particle and nuclear physics experiments rely on fast, low-noise, and highly integrated electronics to extract physical information from detector signals. An introduction to pulse processing electronics for radiation detectors can be found in [1] and [2]. Technical design of ASIC for particle and nuclear physics is discussed in detail in [3] and [4]. While this laboratory uses the readout of SiPMs [5] as a concrete study case, the pulse-processing front-end ASIC concepts addressed here are of general validity and apply to a wide range of radiation and particle detectors. The principles of signal amplification, shaping, discrimination, and readout discussed in this lab are common to many detector technologies used in high-energy physics, medical imaging, astroparticle, and nuclear instrumentation.

9.3.1 Building blocks

A front-end ASIC integrates multiple analog and mixed-signal blocks optimized for processing detector signals close to the sensor, minimizing noise and power consumption while enabling high channel density. In a typical SiPM readout ASIC, the signal processing chain includes:

1. **Input Stage and Preamplifier**. The input stage converts the sensor current pulse into a voltage or charge signal. Charge-sensitive or transimpedance preamplifiers are commonly used to provide sufficient gain while preserving timing information and minimizing electronic noise.

2. **Pulse Shaping**.The preamplified signal is shaped using analog filters to optimize the signal-to-noise ratio and to adapt the pulse width to the requirements of triggering and digitization. The shaping stage reduces high-frequency noise, defines the pulse duration, and influences pile-up and rate capability. Adjustable shaping times allow the user to trade timing resolution against noise performance.

3. **Discrimination and Trigger Formation**. Discriminators compare the shaped signal to a configurable threshold and generate digital pulses when the signal exceeds this level. Discriminator operation is central to triggering, time measurements, and noise rejection. Threshold settings must be carefully calibrated to ensure uniform response across channels and to balance detection efficiency against noise rates.

4. **Analog and Digital Readout**. Depending on the ASIC architecture, the processed signal may be read out either as an analog pulse for further external digitization, or as a digital representation of different parameters of the signal. If the parameters a related to the amplitude of the pulse (usually proportional to the deposited energy) an Analog-to-Digital Converter (ADC) is used. If the parameters are related to the Time-of-Arrival (ToA) of the pulse, a Time-to-Digital Converter (TDC) is used. The digital outputs enable energy, timing, and counting measurements.

9.3.2 Architectures

Figure 31 summarizes the main architectures of FE ASICs. In the simplest implementation (Figure 31-1) , the output of the preamplifier is fed into a discriminator, which compares it to a threshold. When the threshold is crossed, the output becomes a binary signal. For a binary signal the amplitude is quantized ("0/1"), while the timing of the edges preserves the relevant information (this is not the case for a pure digital signal). If we add a digital counter (Figure 31-2), we can compute the hit rate and the output becomes digital.

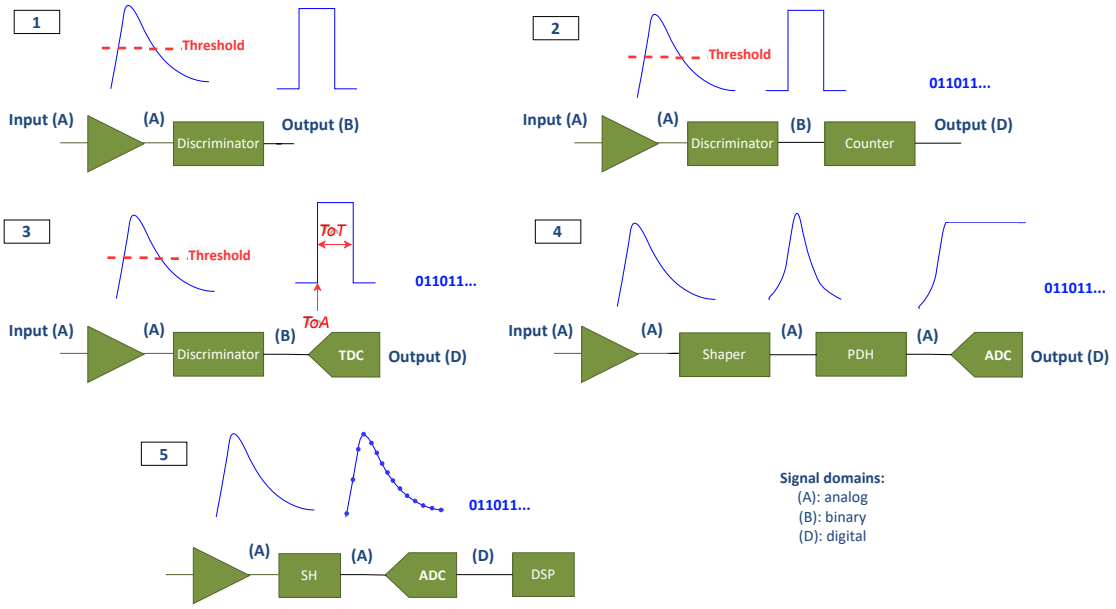


Figure 31: Most common ASIC architectures: 1) binary front end for pulse detection (yes/no), 2) pulse counting, 3) binary front en for time measurements, 4) classic front-end for amplitude (deposited energy measurements) and 5) waveform sampling.

In Figure 31-3, we add a TDC at the output of the discriminator. The leading edge marks the arrival time and the TDC digitizes this edge to output a digital ToA value. The trailing edge can also be digitized, yielding Time-Over-Threshold, which correlates (non-linearly) with pulse amplitude or deposited energy.

Figure 31-4 represents a classical signal processing chain to measure the charge or the amplitude of the incident pulse, which is proportional to the energy deposited on the detector. The shaper is a filter used to optimize the signal to noise ratio of the signal and to re-shape the pulse. For instance, in some cases the signals present long tails from slow decay time constants which results in pulse-up and therefore in measurement errors. The shaper is often used to define the pulse duration and its design is highly related to the expected event rate. After the shaper, a Peak Detector and Hold (PDH) or a Sampel and Hold (SH) circuit samples the peak of the signal. Finally, the ADC will provide a digital representation of the value sampled by the PDH.

Finally, waveform sampling (Figure 31-5) is a unified alternative for detector readout. Signals are amplified, sampled, digitized, and processed through a chain of sample-and-hold, ADC, and Digital Signal Processing (DSP) blocks. By sampling at rates much higher than the signal bandwidth, a single electronics system can extract all relevant information. Typically, SAR, Flash ADCs, or switched-capacitor arrays

are used. Digital signal processing then derives key quantities such as event rate, time of arrival, energy, and, in some detectors, pulse shape for particle identification.

9.3.3 An example: BETA ASIC

In this lab we will use the BETA ASIC [6]. It is a fully programmable chip designed to amplify, shape and digitize the signal of up to 64 SiPM channels, with a power consumption of about ∼1 mW/channel. Thanks to its dual path gain, the BETA chip is capable of resolving single photoelectrons (phes) with SNR>5, while at the same time achieving a dynamic range of ∼4000 photoelectrons. In this way, BETA can provide a cost-effective solution for the readout of SiPM in space missions and low rate (10 kHz) applications. In this paper we describe the key characteristics of the BETA ASIC, and present the evaluation of the performance of 16-channel version which is implemented in a 130 nm technology. The ASIC also contains 2 discriminators that can provide trigger signals with time jitter down to 200 ps r.m.s. for 10 photo-electrons. The linearity error of the charge bi-gain measurement is below 2% for a dynamic range as large as 15 bits. A block diagram of the 16-ch version of the BETA ASIC is shown in Fig. 32. The front-end architecture is based on a dual-gain channel system to process the input current from each SiPM anode node. This current is converted into voltage by means of a resistor divider. Each channel includes a High-Gain (HG) and a Low-Gain (LG) pre-amplifier.

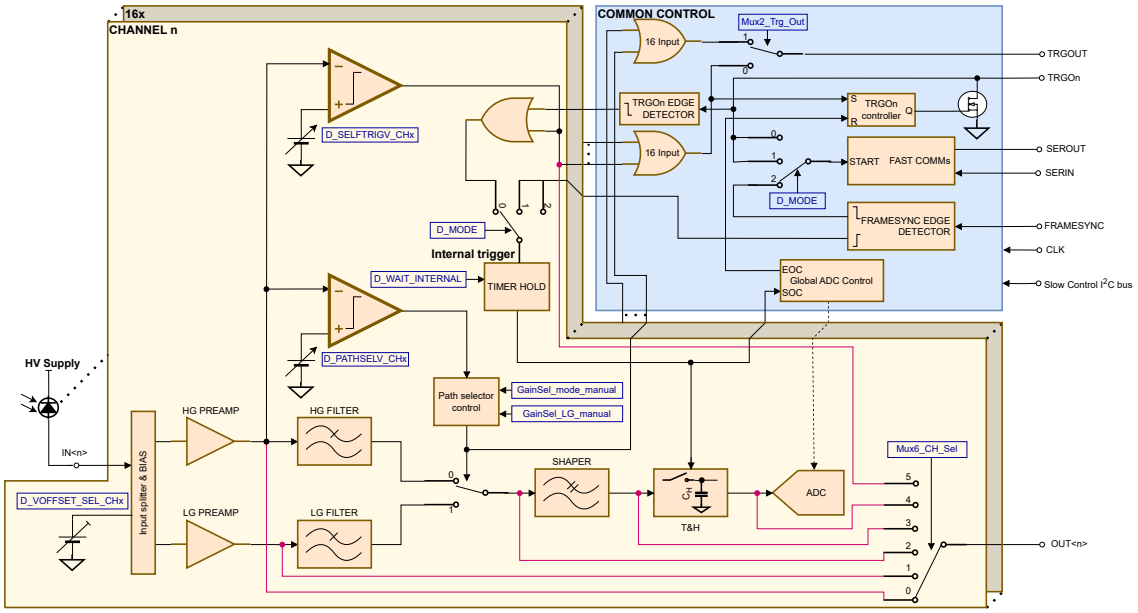


Figure 32: Functional Block diagram of the 16-channel version of BETA ASIC.

The output of the preamplifier is used to generate trigger and gain (or path) binary signals and for charge measurements. Both binary signals are achieved by a voltage rail-to-rail comparator with hysteresis providing a non-linear Time-over-Threshold response. The threshold of the trigger comparator will be typically set to a level corresponding to the signal of few photo-electrons. A global trigger signal is generated as a fast-or of the discriminated trigger output of every channel. In the 16 ch version of BETA ASIC, the individual trigger signal of each channel is provided as output signal. The discriminated path selection output is used to internally select

the HG or LG signal processing chain depending on the amount of detected events and thus avoid saturation. The threshold of the path selection comparator will be typically set to a level corresponding to the signal of 50 to 100 photo-electrons. This sets the dynamic range of the HG path. Higher signals will be processed by the LG path. The charge measurement chain consists on analog shaping, peak detection and digitization. A shaping stage is divided into an initial first-order Low-Pass Filter (LPF) for each of the HG and LG pre-amplifiers. This LPF slows down the signal to give time to the digital circuitry to select between HG and LG signal paths by means of an analog switch. The selected signal path is followed by a single second order LPF implemented using a Sallen-Key topology. The peak amplitude of the shaper is captured using a flip-around Track and Hold (T&H) circuit and then digitized.

9.4 Experimental Setup

The experimental setup scheme includes pulsed LED, a SiPM array, and the readout electronics. The readout electronics employ two boards, one with the BETA ASIC, one with an FPGA that is used to control the ASIC and the trigger. The readout electronics are connected to a PC via USB. We will also use an oscilloscope to monitor the SiPM signals processed by the BETA ASIC (Fig. 33).

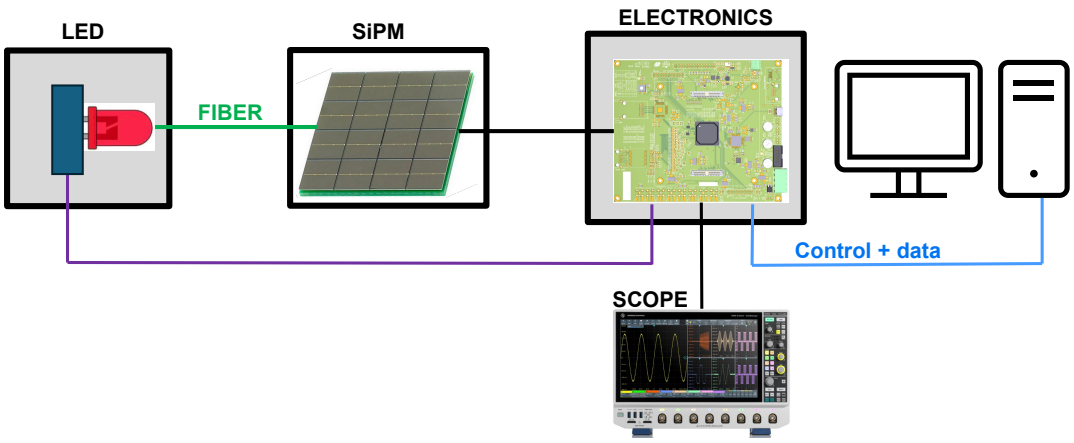


Figure 33: Pulse processing laboratory setup scheme.

The BETA ASIC is configured through a custom-made software that includes a Graphical User Interface (GUI). With this GUI you will be able to navigate over the different parts of the experiment (Fig. 34).

9.4.1 Trigger

The first stages of the analog chain (up to the shaper) are continuously processing the SiPM signals. You will always be able to see ambient photons, dark counts or electronic noise through the ANALOG PROBE that you can configure using the GUI. But the signal will only be digitized when a predefined trigger condition is fulfilled. The ASIC allows using different trigger modes, and you will also be able to monitor the trigger output in the oscilloscope. A binary signal is produced every time a trigger condition is fulfilled. In this session we will work with two trigger modes: Internal and FPGA-generated trigger.

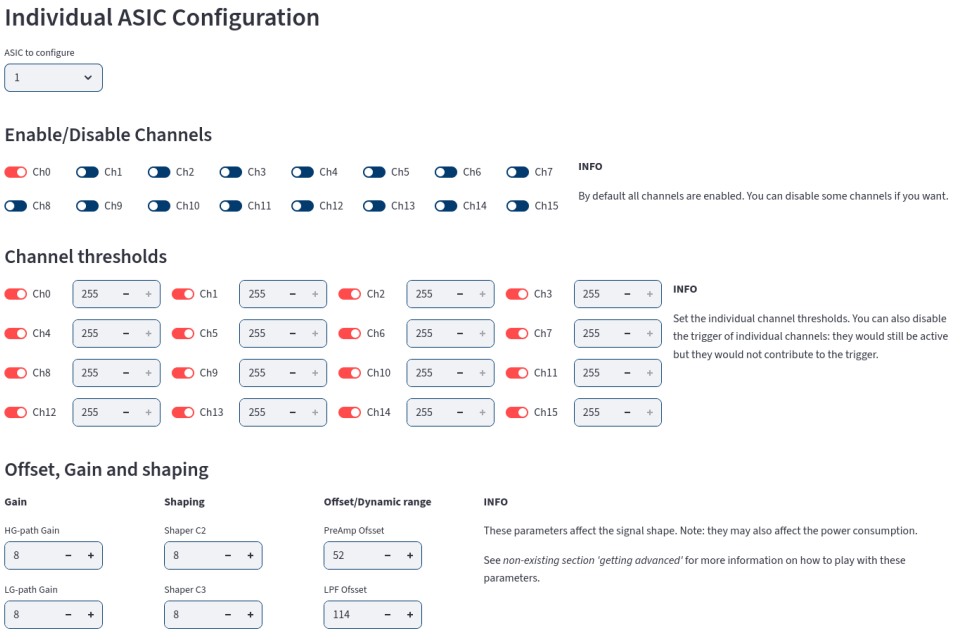


Figure 34: Graphical User Interface front panel used to control the setup.

Internal Trigger In this mode the ASIC is triggered when the signal in one of the channels is above its threshold (Figure 35). You can control the individual channel thresholds and eventually disable some of them.

FPGA-generated Trigger In this mode the FPGA of the electronic module generates a periodic signal that triggers both the LED driver and the BETA ASIC (Fig. 36).

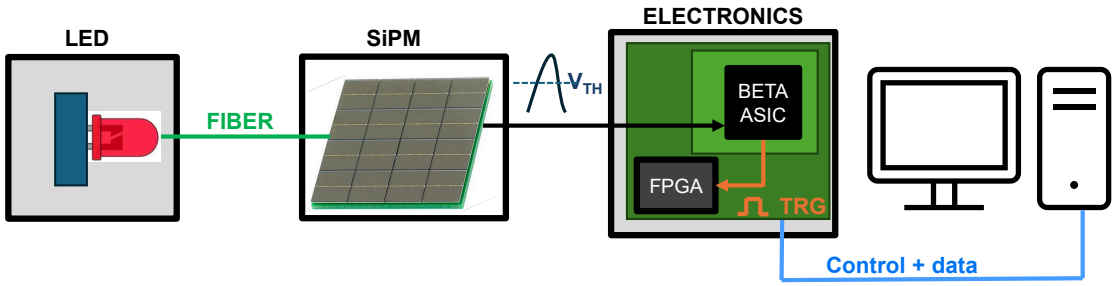


Figure 35: Internal trigger scheme.

9.5 Procedure

9.5.1 Threshold Scan: The S-Curve

A **threshold scan**, commonly referred to as an **S-curve scan**, is a fundamental calibration technique used in particle physics, medical imaging, and digital signal processing to characterize the behavior of a discriminator. This procedure allows

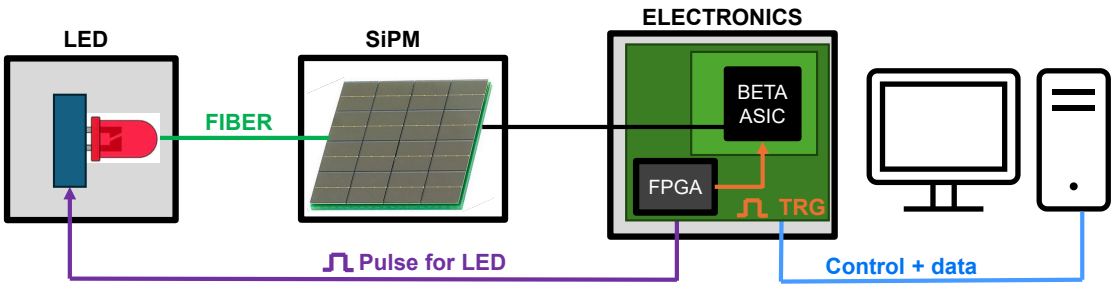


Figure 36: FPGA trigger generation scheme.

for the precise determination of the **pedestal** (the baseline voltage offset) and the **RMS noise** (σ) of a readout channel.

In an ideal, noiseless system, a discriminator behaves as a Heaviside step function:

$$f(V_{in}) = \begin{cases} 1 & \text{if } V_{in} > V_{th} \\ 0 & \text{if } V_{in} \leq V_{th} \end{cases} \tag{9}$$

In many cases, electronic noise follows a Gaussian distribution (this could not be true if a deterministic interference is present). When the threshold V_{th} is set near the baseline, noise fluctuations randomly trigger the discriminator. As the threshold is swept through the baseline, the probability of firing changes from 100% (full occupancy) to 0%, resulting in a characteristic sigmoid or "S-curve" shape.

The calibration involves the following systematic steps:

1. **Stimulus Selection:** The system is either left at its inherent "dark" baseline (to measure intrinsic pedestal) or a fixed-amplitude test pulse is injected via a calibration capacitor.

2. **Threshold Sweeping:** The threshold voltage V_{th} is incremented in discrete steps across the expected range of the baseline.

3. **Data Acquisition:** At each threshold step, the number of discriminator fires (hits) is counted over a fixed number of clock cycles or trigger gates.

4. **Normalization:** The efficiency (or occupancy) ϵ is calculated:

$$\epsilon(V_{th}) = \frac{N_{hits}}{N_{total}} \tag{10}$$

Use the GUI to perform a threshold scan in all the channels with SiPM bias voltage OFF. Now, switch ON the SiPM bias voltage and repeat the scan. This test is performed with LED pulser OFF.

Q: What are the main differences? Can you identify 2 different types of behaviour?

9.5.2 Scan rate

A scan rate will be performed with the SiPM bias voltage ON A critical performance metric for SiPMs is the **Dark Count Rate (DCR)**, which refers to the frequency

of avalanche events triggered by thermal generation or tunneling rather than incident photons [5]. A threshold scan is the primary method used to quantify DCR and characterize the device's gain and crosstalk levels.

In a SiPM, the output pulse amplitude is proportional to the number of cells (pixels) that fire simultaneously.

- **1 p.e. (photo-electron):** A single cell fires (thermal noise or one photon).
- **2 p.e. or more** Multi-cell events caused by Optical Crosstalk (OCT) or Afterpulsing.

As a threshold V_{th} is increased, the discriminator will exclude lower-amplitude pulses. The resulting DCR vs. Threshold plot appears as a "staircase" function.

1. **Setup:** The SiPM is biased at a specific Overvoltage ($V_{OV} = V_{bias} - V_{breakdown}$) and kept in total darkness within a temperature-controlled environment.
2. **Threshold Sweep:** A discriminator threshold is swept from below the 1 p.e. level to several p.e. levels.
3. **Frequency Measurement:** At each threshold, a frequency counter records the rate of pulses (counts per second, Hz) that exceed the threshold.

The "plateaus" in the staircase represent the gaps between p.e. peaks. The distance between the mid-points of successive plateaus (or the peaks of the derivative of the scan) provides a measurement of the **Gain** of the SiPM in terms of ADC counts or Volts per photo-electron.

Switch ON the SiPM bias voltage and perform a rate scan. This test is performed with LED pulser OFF.

9.5.3 Analog signal processing

In this part we will play with **some** of the ASIC functionalities and watch their effect on the signal shape in the oscilloscope. The Analog Probe of the BETA ASIC lets you watch the different stages of the analog chain described in section 9.3.3. You will be able to watch the effect of changing some of the ASIC registers in the different stages of the analog chain.

Waveform visualization

Configure the ASIC setting the Trigger mode as Internal. Disable all channels, but Ch 0. Try to configure the Channel Threshold of Ch 0 at the single-photoelectron level. You can infer this threshold from the rate scan you have just made. Set the Gain Selection at High Gain and the Analog probe to show the High Gain output. Unless specified, ignore all other parameters.

Q: Is the ASIC triggering? With which amplitudes? What happens if you increase the threshold?

Detecting LED pulses

Now configure the TRIGGER MODE to the FPGA-TRIGGERED one. Try to identify the detected LED pulses in the oscilloscope. You can play with the LED intensity.

Q: What is the amplitude of a single-phe signal? And of a signal of 2 phes? And of a signal of 10 phes? What happens if you change the SiPM overvoltage?

Now play with the GAIN of the High-Gain path.

Q: What happened to the signal? What's the amplitude of the single-phe now?

Shaping

Now switch the ANALOG PROBE to the SHAPER. If you modify again the Gain of the high-gain path you should see that the shaper amplitude changes. Now we will modify the signal shape. Play with the SHAPER C2 and SHAPER C3 parameters and see how the shaper signal changes.

Dynamic range

If you did not do it yet, if you increase the LED intensity you will reach a point at which the high-gain path saturates. The ASIC has a dual-gain path to have good resolution to events of a few photons and keep a wide dynamic range to detect events involving thousands of photons. When the GAIN SELECTION is set in AUTOPATH, the ASIC decides which of the two paths should be shaped and digitized depending on the signal amplitude. Set back the LED intensity to a few photons, put the GAIN SELECTION in AUTOPATH and the analog probe at the SHAPER. If the LED signal is low enough, the shaper should be processing the High-gain path. Now slowly increase the LED intensity until you see that it starts processing the Low-gain path.

Q: At how many phes was the path-selection threshold set?

9.5.4 Data acquisition

Now that you gained experience with the ASIC configuration, we will digitize the data and analyze the acquired data. Keep the TRIGGER mode set to FPGA TRIGGERED, we will try to acquire the detected LED pulses. Set the LED intensity at the level of a few photons and be sure that there are events in which you do not detect photons at all.

Delay optimization

Jump into ACQUISITION and acquire data for 10 seconds. Can you identify the events of a single phe? Probably not, since the acquisition delay has not been optimized yet. The ASIC digitizes the shaper output at a fixed time delay from the moment in which it receives the trigger. Ideally, you would like to digitize the peak of the shaper signal. That delay can be configured through the TRIGGER DELAY button.

To optimize the trigger delay use one of the oscilloscope markers to track the peak of the SHAPER output. Then set the ANALOG PROBE to the TRACK AND

HOLD. Play with the TRIGGER DELAY until the HOLD matches the marker you set. Then acquire again. Can you identify the single phe peak now?

Shaping, overvoltage and single-phe resolution

Different SiPMs produce different pulse shapes. The ASIC gives flexibility by allowing different shaping configurations. The user must play with these configurations and find the one that is more suitable for its sensor and application. Play again with the shaper and gain of the high-gain path and see how this affects your single-phe resolution.

9.6 Analysis and interpretation of the results

9.6.1 S-curve analysis: pedestal and noise

The S-curve data is typically fitted using a **Complementary Error Function (erfc)**, which represents the integral of a Gaussian noise distribution. The probability P of a trigger at a given threshold V_{th} is modeled as:

$$P(V_{th}) = \frac{1}{2}\text{erfc}\left(\frac{V_{th} - \mu}{\sqrt{2}\sigma}\right) \tag{11}$$

Where:

- **The Pedestal (μ):** The threshold value where the efficiency is exactly 50%. This represents the effective DC baseline of the electronic channel.
- **The Noise (σ):** The width of the transition region. It is the standard deviation of the electronic noise. A steeper slope indicates lower noise.

Calculate the pedestal and noise for each channel. Compare what happens if your switch ON and OFF the SiPM bias voltage.

9.6.2 Histogram analysis: signal-to-noise

Calculate the signal-to-noise ratio (SNR) for single phes as:

$$SNR_{1phe} = \frac{\mu_{1phe} - \mu_{ped}}{\sigma_{1phe}} \tag{12}$$

where $\mu_{1phe} - \mu_{ped}$ is the distance between the single-phe and the pedestal peaks and σ_{1phe} is the standard deviation of the single-phe peak. How does it change in the different shaping configurations tested? Which ones gives the best SNR?

References

[1] G. Knoll, *Radiation Detection and Measurement*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2010.

[2] W. R. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-To Approach*. Berlin: Springer-Verlag, 1994.

[3] A. Rivetti, *CMOS Front-End Electronics for Radiation Sensors*, CRC Press, Taylor & Francis Group, Boca Raton, 2015.

[4] R. Turchetta (ed.), *Analog Electronics for Radiation Detection: Devices, Circuits, and Systems*, CRC Press, Taylor & Francis Group, Boca Raton, 2016.

[5] F. Acerbi and S. Gundacker, *Understanding and simulating SiPMs*, Nuclear Instruments and Methods in Physics Research Section A, vol.926, pp.16–35, 2019.

[6] Sanmukh, A., Gómez, S., Comerma, A., Mauricio, J., Manera, R., Sanuy, A., Guberman, D., Català, R., Espinya, A., Orta, M., de la Torre, O., Gascón, D., & HERD Collaboration *Low-power SiPM readout BETA ASIC for space applications*. Nuclear Science and Techniques, 35, Article 59 (2024). https://doi.org/10.1007/s41365-024-01419-z

Exercise 10 Introduction to Hierarchical Trigger Systems: From FPGA Level-1 to High-Level Software

10.1 Objectives

- Understand the physics-driven requirements and technological constraints that motivate the implementation of trigger systems in modern particle physics experiments.
- Identify the experimental constraints and needs that determine the architecture and performance of trigger systems, including interaction rate, detector occupancy, latency, bandwidth and storage limitations.
- Gain familiarity with modern hardware architectures used in trigger and data acquisition systems.
- Implement and validate trigger algorithms with different levels of complexity under strict latency, logic, and computational constraints.
- Analyze trigger performance in terms of efficiency, latency, purity, throughput and background rejection.

10.2 Professors

- Álvaro Navarro Tobar
- Javier Prado
- Cristina F. Bedoya
- Santiago Folgueras

10.3 Theoretical Introduction

Modern particle physics experiments generate data streams that often exceed the available bandwidth for data transfer, storage and offline processing.

This may result from very large event rates, as in high-luminosity collider experiments, but also from detectors with many readout channels and detailed waveform sampling, where large amounts of information are acquired for every event, for example, through Flash-ADC digitization of detector signals.

Trigger and online event reconstruction systems are therefore essential to reduce the data flow in real time while preserving events with potential physics interest.

Although some modern experiments are described as triggerless, they still rely on online event selection and real-time data reduction. In these architectures, the trigger functionality is shifted from dedicated hardware decisions to software-based online reconstruction and filtering stages. More generally, whenever the data throughput exceeds the available storage or offline processing capabilities, some form of real-time event selection becomes unavoidable.

A trigger system is therefore an online event-selection and data-reduction chain that brings the data flow down to manageable levels while preserving events with potential physics interest.

In this laboratory session, we will study a simplified example of a hierarchical trigger system similar to those employed in large modern particle physics experiments. In a hierarchical architecture, the event selection process is divided into multiple stages with increasing levels of complexity and processing time:

- **Level 1 (L1) hardware trigger:** Early stages are typically implemented in dedicated real-time hardware such as FPGAs or ASICs. They use reduced detector information and simple algorithms to perform fast, coarse event rejection. They may also generate a Level-1 trigger signal to select the events retained by the front-end pipelines (or readout electronics) when those systems cannot sustain the full readout bandwidth.

- **High-Level Software Trigger (HLT):** Subsequent stages apply more sophisticated reconstruction and selection algorithms to a smaller subset of candidate events, using richer information and more selective decision logic. These systems usually rely on software-based processing running on large computing farms built from CPUs, GPUs, or both. Their goal is to reduce the amount of data that will be stored offline, either by rejecting more events or by storing a more compact representation of them.

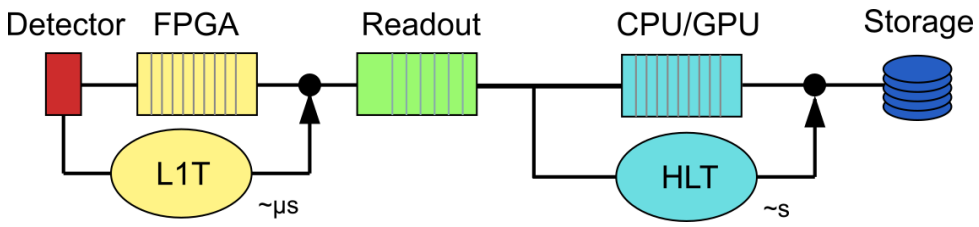


Figure 37: Simplified hierarchical trigger and data acquisition architecture inspired by LHC experiments.

Some terms are particularly useful in the context of trigger systems:

- **latency:** time between the arrival of detector data and the trigger decision; it is limited by the depth of the front-end buffers.

- **throughput:** amount of data or number of events that can be transported or processed per unit time.

- **compute budget:** processing resources and algorithmic complexity that can be spent on each event.

- **pile-up:** multiple interactions overlapping in the same acquisition window, increasing occupancy and selection difficulty.

- **prescale:** reduction factor that records only a fraction of the events accepted by a trigger path.

- **rate:** number of events processed, accepted, or recorded per unit time.

- **pipeline:** processing scheme in which several stages work simultaneously on different events, enabling high throughput.

The global acceptance can be written as:

$$P(\text{accept}) = P(\text{L1 accept}) \times P(\text{HLT accept} \mid \text{L1 accept}). \tag{13}$$

For each class of events (signal or background), we define:

$$\epsilon = \frac{N_{\text{accepted}}}{N_{\text{total}}}, \tag{14}$$

where ϵ is the trigger efficiency. Data reduction is quantified by

$$R = \frac{N_{\text{input}}}{N_{\text{output}}}. \tag{15}$$

In practice, trigger design is a multi-objective optimization: maximize signal efficiency, improve purity, and keep latency and rate within hardware limits.

Examples from the LHC (Run 3 and Upgrade). The table below summarizes representative trigger architectures and design targets for several LHC experiments in Run 3 and in their upgrade configurations. It is intended as a compact comparison of how different experiments balance latency, rate reduction, and software complexity.

Table 1: Representative trigger design values at the LHC (Run 3 and Upgrade eras).

Experiment	**First stage**	**Second stage**	**Typical output target**
CMS (Run 3)	Hardware L1, 3.8 μs, up to 100 kHz	CPU HLT with GPU acceleration for selected reconstruction tasks	1–1.5 kHz to permanent storage
CMS (Phase-2)	Track-trigger L1, 12.5 μs, 750 kHz design (about 1 MHz margin)	More selective HLT on a heterogeneous CPU/GPU farm	About 5–7.5 kHz
ATLAS (Run 3)	Hardware L1 trigger, about 2.5 μs, up to 100 kHz	Software Event Filter on a large CPU farm	About 1–2 kHz
ATLAS (Phase-II)	Hardware L0, about 1 MHz with about 10 μs latency; L1 track-trigger refinements studied	Software Event Filter; accelerator use under study	About 10 kHz
LHCb (Run 3)	Triggerless 30 MHz readout	GPU HLT1 (Allen) followed by CPU HLT2	HLT1 about 1–2 MHz to disk, final about 10 kHz
LHCb (Upgrade)	Triggerless O(30 MHz) readout at higher luminosity	Expanded GPU HLT1 plus CPU HLT2	Higher throughput than Run 3

These examples show the same core idea with different implementations: hierarchical selection is always a co-design problem between physics goals, detector technology, data links, and computing infrastructure.

10.4 Experimental Setup

This exercise is built around a compact trigger demonstrator that emulates a tracker-like detector exposed to two main classes of structured activity (such as the MiniPIX detector): alpha-like compact deposits and muon-like trajectories with different curvatures. The physics target is the identification of high-energy muons while maintaining sufficient rejection power against diffuse background, isolated noise hits, and lower-momentum topologies. Your task is to design a two-stage trigger chain that achieves a good compromise between efficiency, purity, and implementation cost. The exercise hardware is composed of these two elements:

- a Boolean Board FPGA platform (Fig. 38, label A) that runs the Level-1 trigger and the event-generation/control logic, processing events in real time at 10 MHz with a simple low-latency algorithm;

- an ESP32 module (Fig. 38, label F) with integrated 0.42 inch OLED display that runs the HLT in C++, receives the lower-rate event flow accepted by the FPGA, and provides event visualization and summary readout.

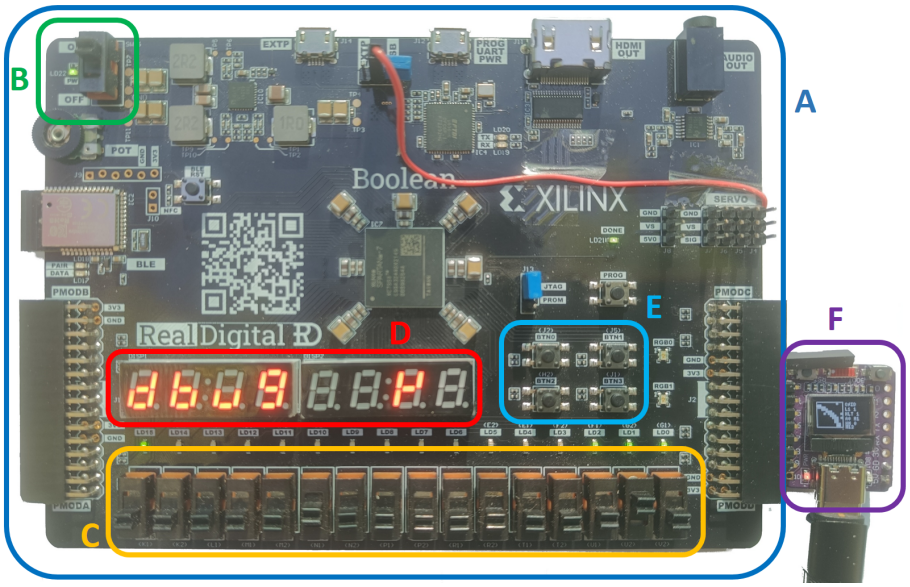


Figure 38: Hardware setup used in the trigger exercise. **Label A**: Boolean Board. **Label B**: power switch and status LED. **Label C**: switches and indicator LEDs used to control the operating mode. **Label D**: 7-segment displays for operating mode indication. **Label E**: action buttons used either to advance events in debug mode or to launch a complete data-taking sequence in run mode. **Label F**: ESP32 module with its OLED display.

As in a real experiment, the trigger chain is first prototyped in a software emulator (Jupyter notebook), where you can validate candidate algorithms on a large sample of events before moving to the actual implementation. The practical information needed to access the notebooks, connect to the remote firmware compilation server, and prepare the ESP32 programming environment on your computer will be distributed separately in an annex with additional instructions.

Hardware workflow

During the session, the FPGA and the ESP32 operate as a two-stage trigger chain. The FPGA applies the first low-latency selection (L1) and forwards the surviving events to the ESP32, where a software HLT refines the decision. The system can be used in two complementary operating modes:

- **run mode:** the chain processes a complete data-taking sequence and the OLED presents the resulting efficiency figures at the end of the run;
- **debug mode:** the chain advances event by event, and the OLED shows the detector matrix together with the information indicating whether the event is accepted by L1 and, if so, whether it is also accepted by the HLT.

The mode-selection switches on the Boolean Board choose between these operating modes. In `DEBUG` mode, the board switches can also be used to filter events by category (noise, alpha, or muon), allowing you to focus on specific patterns while testing the behavior of the trigger logic. The push-buttons are then used according to the selected mode: in `DEBUG` they advance the visualization of stored events, whereas in `RUN` they start a full processing sequence.

Emulator workflow

The same event model is explored in the notebook before it is exercised on the hardware. Events are represented as binary hit matrices from a synthetic tracker detector. The sample contains background/noise patterns, alpha-like compact clusters, muon-like tracks of both low and high energy, and mixed alpha+muon events.

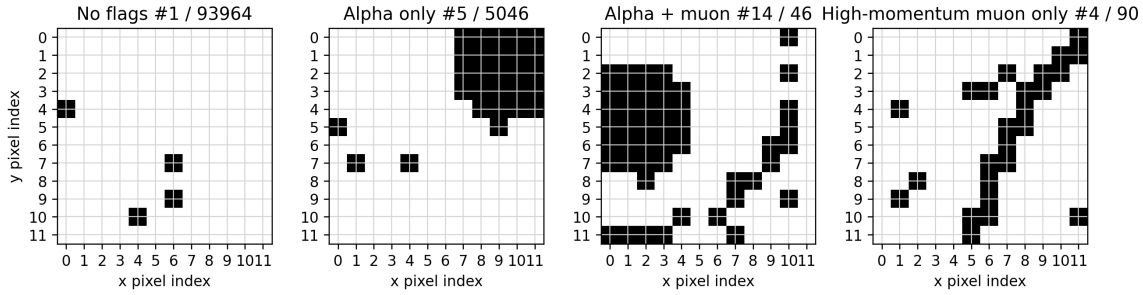


Figure 39: Representative event categories used in the notebook stage: background/noise, alpha-like activity, alpha+muon overlap, and a high-energy muon candidate.

The notebook plays the role of a trigger emulator. In a real experiment, trigger algorithms are first evaluated in an emulation environment running on Monte Carlo samples, because that is where you can test the decision logic, estimate efficiencies and rejection power, and study whether the design is likely to satisfy the physics goals before moving to the online hardware. In this exercise, the notebook lets you identify the limitations of the baseline trigger, test better alternatives, and decide which ideas are realistic enough to transfer to the FPGA and ESP32 implementations.

10.5 Procedure

10.5.1 Emulator exploration and algorithm prototyping.

The goals of this stage are:

- Familiarize yourself with the event data and identify which patterns correspond to sparse noise, compact alpha-like deposits, low- and high-energy muon candidates, and mixed alpha+muon events.
- Understand how the relative frequency of these categories favors the implementation of the trigger system as a two-stage chain rather than as a single-stage selection.
- Run the provided baseline functions and confirm that their efficiency and rejection power are not yet satisfactory.
- Understand how implementation cost and algorithmic performance interact in determining whether the physics goals can be met.
- Modify the notebook L1 and HLT trigger algorithms and test the effect of the changes on the efficiency, purity, and event reduction.
- Understand that algorithm prototyping in the emulator is constrained by what can realistically be implemented in hardware, so the notebook is not a free playground for arbitrarily complex algorithms, but a tool for testing realistic ideas before moving them to the actual trigger chain.

Open the Jupyter notebook supplied for the exercise and follow the proposed steps to explore the event categories, test the baseline trigger functions, and prototype improved algorithms. When the time comes to propose improvements, favor simple features that can be implemented in the FPGA with a small number of logic elements and a simple data flow, or in the ESP32 with a straightforward C++ implementation. Avoid complex observables that would require large look-up tables, floating-point operations, or non-local event information. Since it may be difficult to foresee the exact implementation cost of a given algorithm, it is better to start with simple ideas and test their performance before attempting more complex ones. If, during the hardware stage, you find that the exact notebook prototype is too complex or does not fit the real data flow, go back to the notebook, test a more realistic version, and re-evaluate its performance before deploying it. Trigger design is an iterative process in which you move back and forth between emulator and hardware until you reach a reasonable compromise between performance and implementation cost.

The last cells of the notebook allow you to paste and analyze the run summary information dumped by the ESP32 to the serial port. You can use this output to evaluate the algorithms implemented on the hardware and compare their performance with the estimates obtained in the notebook. After deploying the final trigger architecture, complete that last part of the notebook with the focus on physics performance rather than implementation details.

10.5.2 Hardware execution of the baseline chain.

The purpose of this stage is to see how the same trigger logic behaves when executed as a complete acquisition sequence rather than as isolated debug events.

Once you have an initial notebook-level idea, turn to the hardware chain.

1. Power the setup on the Boolean Board and verify that the ESP32 OLED initializes correctly.

2. Use the mode-selection switches to choose `DEBUG` mode.

3. Step through several events in different categories with the corresponding push-button and compare the OLED view with the event categories studied in the notebook, and the L1/HLT decisions expected from the baseline functions.

4. Compare, qualitatively, the performance of the baseline trigger algorithms on the hardware with the behavior observed in the notebook.

After the event-by-event inspection, switch the board to `RUN` mode.

1. Start a complete run using the corresponding push-button and wait for it to finish.

2. Read the efficiency results shown on the OLED at the end of the run, which correspond to the baseline trigger chain implemented in the FPGA and ESP32.

3. Compare, quantitatively, these results with the notebook estimates for the same algorithm. A reduced summary of the run is displayed on the OLED, but the full information is also dumped to the serial port and can be copied from the serial monitor of the programming environment and pasted into the notebook for a more detailed analysis.

Note that the hardware run is executed over a much larger sample of events than the notebook, so the observed efficiency figures should be more stable and closer to the true performance of the algorithm. The same situation arises in real experiments, where generating Monte Carlo samples is computationally expensive, so only a limited number of representative events can usually be used during the prototyping stage, while the final performance is measured only after deployment on large data samples. In our case, the events delivered by the hardware are slightly different from those in the notebook, so the observed performance may differ from the notebook estimates even for the same algorithm. However, the relative comparison between baseline and improved algorithms should remain broadly consistent between notebook and hardware.

10.5.3 FPGA Level-1 implementation of an improved algorithm and validation on hardware.

The next task is to improve the Level-1 trigger implemented in the FPGA. The aim is to transfer a simple but well-motivated algorithmic idea into a realistic hardware implementation. Favor clear logic based on counts, local patterns, or compact topological rules rather than operations that would only be practical offline.

Do not worry if you are not familiar with FPGA programming. For this exercise, it is enough to understand that programming an FPGA does not mean writing a sequence of commands to be executed by a CPU; instead, you are describing a digital circuit. Because the FPGA is a reconfigurable array of simple logic elements, complex algebraic operations such as trigonometric functions or floating-point arithmetic are generally not practical. As the complexity of the logic increases, two things happen: resource utilization grows and timing becomes more difficult to close. A more complex algorithm requires more hardware resources, so it may exceed what is available on the device. At the same time, a larger circuit increases the signal

propagation time and may therefore exceed the latency budget, which in this exercise is set by the clock period. For that reason, when designing algorithms for the FPGA, you should favor simple logic based on counts, local patterns, or compact topological rules.

The VHDL code provided is intended to be readable and should give you enough guidance to begin modifying the L1 logic. Start with small, controlled edits to the available code. If the hardware implementation you obtain differs from what you expected in the emulator, go back to the notebook and re-evaluate the performance of the implementation you can actually deploy. If you run into difficulties with the implementation or the workflow, ask the professors for help.

If, after a reasonable attempt, your own implementation is not performing satisfactorily, you may continue with the provided "spoiler" VHDL module and its corresponding notebook emulation function.

The workflow is as follows:

1. Connect by SSH to the remote server indicated in the additional instructions annex and set up the environment for firmware compilation.
2. Locate the VHDL module provided for the L1 algorithm and understand how it works.
3. Replace the baseline logic with a slightly better implementation derived from the notebook study.
4. Launch the remote build using the script provided for firmware compilation.
5. Wait for the synthesis and implementation flow to finish and confirm that a valid bitstream has been produced.
6. Ask the professors to load the generated FPGA firmware through the board programming USB connector.

Once the updated FPGA firmware has been loaded, repeat the two operating modes.

- In `DEBUG` mode, verify event by event that the new L1 accepts and rejects patterns in the expected way.
- In `RUN` mode, repeat the sequence and record the efficiency figures reported by the OLED.
- Compare the new hardware behavior against both the notebook expectation and the baseline firmware.

If the observed behavior differs significantly from the notebook expectation, revisit the assumptions of the chosen feature or threshold before moving to the HLT stage.

10.5.4 ESP32 HLT implementation of the improved algorithm and validation on hardware.

At this point in the exercise, the input reaching the ESP32 has already been filtered by the FPGA. The HLT therefore runs on a much lower-rate stream than the original detector output, allowing more selective logic than in the L1 stage.

The main limitation is no longer fixed latency, but execution time. The HLT does not need to return a decision in a strictly constant time for every event, but it must still be fast enough to sustain the average rate delivered by L1. In practice, this means you can consider richer decision logic than in the FPGA, while keeping the code compact, readable, and efficient.

The HLT is ordinary C++ code running on a microcontroller rather than hardware logic. That gives you more freedom when choosing observables or combining conditions, but every additional operation has a direct execution-time cost. The goal is not to reproduce an unrestricted offline analysis, but to build a software trigger that improves the end-to-end chain while remaining compatible with the incoming rate.

The provided source code should be readable enough to let you identify where the decision is taken and how the result propagates through the demonstrator. Start with small, controlled edits and validate them frequently. If the uploaded firmware behaves differently from what you expected in the emulator, return to the notebook and compare the intended algorithm with the one actually deployed. As in the FPGA stage, if you are unable to obtain a satisfactory implementation in the available time, you may continue with the provided "spoiler" algorithm and its corresponding hidden emulation function.

The workflow is as follows:

1. Prepare the ESP32 programming environment following the additional instructions distributed outside this document.

2. Open the firmware project and locate the source file that implements the HLT decision.

3. Replace the baseline HLT with an improved version guided by the notebook observations and the behavior of the updated L1.

4. Build and upload the firmware to the ESP32 from your workstation.

5. Restart the demonstrator and confirm that the OLED, communication link, and event display work as expected.

Once the updated ESP32 firmware has been loaded, repeat the two operating modes.

- In `DEBUG` mode, verify event by event that the new HLT accepts and rejects patterns in the expected way once the event has passed L1.

- In `RUN` mode, repeat the sequence and record the efficiency figures reported by the OLED for the full L1+HLT chain.

- Compare the new end-to-end behavior against both the notebook expectation and the previous firmware.

If the observed behavior differs significantly from the notebook expectation, revisit the assumptions of the chosen HLT features or thresholds and check whether the discrepancy comes from the software stage itself or from the interaction between L1 and HLT.

10.6 Analysis and interpretation of the results

Paste the final run summary information from the serial monitor into the indicated cell of the notebook and execute the corresponding analysis section. This will guide you through the final evaluation of the trigger chain, including the case in which you continued with the provided “spoiler” implementation.

Compare the final efficiency figures with the notebook estimates and with those measured for the previous firmware versions. Discuss how the L1 and HLT improvements contributed to the overall performance, whether the results match the expectations formed during the emulator study, and how your choices in algorithm complexity, feature selection, and threshold tuning affected the balance between efficiency, purity, and data reduction.

Interpret the trigger performance in physics terms. How does the efficiency for high-energy muons compare with the rejection achieved for background and low-energy events? How would this balance affect the physics reach of an experiment relying on such a trigger system, and what would further improvements cost in complexity, resources, or execution time?

Keep in mind that this type of final analysis is usually not possible in a real experiment, because the true event categories are not known and rejected events never reach the readout. Many experiments therefore reserve a small fraction of the bandwidth for a “monitoring stream” that records a random subset of events independently of the trigger decision. In this exercise, by contrast, the labeled event categories and the larger hardware sample make the comparison with the notebook expectations much more direct.

Exercise 11 Real time signal reconstruction in FPGA with Machine Learning

11.1 Objectives

General Objective: Design, train, evaluate, and deploy a low-latency FPGA-based machine-learning model for reconstructing pulse amplitude and timing from digitized calorimeter signals.

Specific Objectives:

- Understand how calorimeter pulses are formed and digitized by ADCs.
- Convert ADC streams into fixed-size time windows (events) suitable for supervised learning.
- Implement and train Neural Networks models to predict amplitude and time from the windowed inputs.
- Evaluate models using metrics like MAE, MSE, R^2, and bias.
- Perform fixed-point quantization of trained models and configure FPGA pipelining.
- Compare FPGAs resources and adapt model complexity accordingly.
- Synthesize the HLS design and package it as an IP core for FPGA implementation.

11.2 Professors

- Alberto Valero

11.3 Theoretical Introduction

Modern high-energy physics experiments operate at extremely high interaction rates, requiring fast and accurate reconstruction of detector signals. In calorimeter systems, particles deposit energy that is converted into electrical pulses, digitized, and processed to extract key quantities such as amplitude (energy) and time of arrival. Traditionally, these quantities are reconstructed using analytical methods that rely on predefined pulse shapes. However, in realistic experimental conditions, especially under high pile-up, these assumptions break down, degrading performance. Machine learning models, and in particular neural networks, offer a data-driven alternative capable of learning complex signal patterns directly from examples. This enables improved reconstruction accuracy even in the presence of noise, distortions, and overlapping signals. In this context, the challenge is not only achieving accurate reconstruction, but doing so under strict real-time conditions. This motivates the use of specialized hardware platforms, such as FPGAs, where neural networks can be deployed for low-latency inference.

11.3.1 Pulse Formation and Shaping

Figure 40 illustrates the full analog front-end readout chain, from sensor to trigger, together with representative signal shapes at each stage.

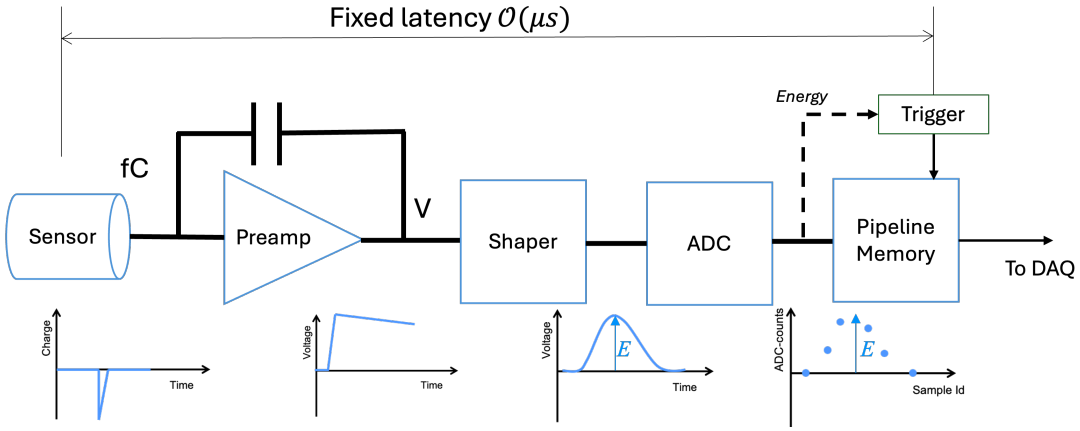
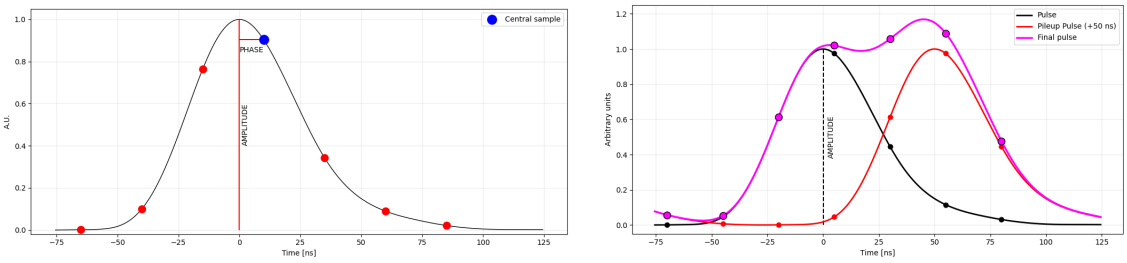


Figure 40: Block diagram of a standard calorimeter data acquisition system.

After light is converted into an electrical current, a preamplifier integrates and amplifies the signal. The subsequent shaping stage transforms the fast current spike into a unipolar or bipolar voltage pulse with a characteristic fast rise and slower exponential decay, typically extending over hundreds of nanoseconds. Different experiments sample such shaped pulses with different frequency and usually with a clock synchronous with the bunch crossing which permits the measurement of the deposition energy and time with respect to the collision (Figure41a). However, at high collision rates (high luminosity), pile-up causes multiple pulses to overlap in time. Electronics noise and pile-up distort the waveform (Figure41b). The goal is to extract the true pulse amplitude (energy) and arrival time from a noisy sample of the waveform. Classical methods like optimal filtering or peak finding assume a known pulse shape and linear summation, which degrades in dense pile-up. In contrast, a properly trained neural network can learn to disentangle overlapping pulses by exploiting subtle waveform features.



(a) (b)

Figure 41: (a)Nominal pulse shape after shaping with 7 digital samples every 25 ns.(b) Illustration of out-of-time pile-up: a second pulse shifted by 50 ns is added to the nominal one, producing a distorted combined waveform.

11.3.2 Real time signal reconstruction constraints for trigger

The implementation of real-time reconstruction algorithms in trigger systems is constrained by strict timing and determinism requirements. At facilities such as the LHC, detector signals are sampled continuously at fixed intervals (typically every 25 ns), and trigger decisions must be delivered within a latency budget of only a few microseconds. This implies that the processing chain must operate as a fully pipelined system, where each stage processes new data at every clock cycle without introducing variability or stalls. As a result, algorithms must be designed with fixed computational complexity and predictable execution time. FPGAs are particularly well suited to this task because they enable deeply pipelined and highly parallel architectures with cycle-accurate behavior. However, deploying neural networks in this environment introduces additional constraints, including limited hardware resources, the need for fixed-point arithmetic, and careful control of latency versus resource usage. Therefore, models must be optimized not only for accuracy, but also for efficient hardware implementation.

11.3.3 Fixed-Point Arithmetic and ap_fixed

FPGAs do not have native IEEE 754 floating-point units (or have very few). Instead, Xilinx HLS provides the ap_fixed<W,I> type, where W is the total word width in bits and I is the number of bits devoted to the integer part (including the sign bit). The remaining W-I bits represent the fractional part. For example, ap_fixed<16,6> uses 16 bits total: 6 for the integer part (range -32 to +32) and 10 for the fraction (resolution » 0.001). Choosing the right precision requires balancing two competing constraints:

- **Dynamic range**: The integer part must be wide enough to represent the largest expected value without overflow. For normalized inputs with values typically in [-4, +4], 4 integer bits (after sign) are usually sufficient.
- **Resolution**: The fractional part must be fine enough that quantization noise does not dominate the reconstruction error. More fractional bits give better resolution but cost more FPGA resources (wider multipliers → more DSPs).

During training we mirror the FPGA quantization by converting z-score normalized inputs to int16 (Q15 format) using a fixed scale factor S derived from the maximum absolute value in the training set. This minimizes the gap between floating-point training accuracy and fixed-point inference accuracy.

11.3.4 The hls4ml Workflow

hls4ml is an open-source Python library that automates the translation of trained neural networks (from PyTorch, Keras/TensorFlow, or ONNX) into synthesizable HLS C++ code targeting Xilinx and Intel FPGAs. The workflow has four stages:

- **Configuration**: A configuration dictionary specifies the fixed-point precision for weights, biases, and intermediate results, the reuse factor (how many times each multiplier is reused across different computations), and the target FPGA part and clock period.

- **Conversion**: hls4ml parses the PyTorch model graph, maps each layer to an HLS template (Dense, Conv1D, ReLU, MaxPool, etc.), and instantiates the fixed-point types.

- **Compilation and functional verification**: The generated C++ code is compiled with a CPU backend so predictions can be compared against the original PyTorch model. Large differences indicate precision or conversion issues.

- **HLS Synthesis**: Vivado/Vitis HLS synthesizes the C++ into a Register Transfer Level (RTL) netlist, estimating DSP, LUT, FF, and BRAM usage, and reporting the critical path latency. The reuse factor RF controls a fundamental latency–area trade-off. With RF=1, every multiply-accumulate in a layer is implemented as a separate hardware unit operating in parallel (fully unrolled), giving minimum latency but maximum resource usage. With RF=N (the number of weights), a single multiplier is reused N times sequentially, reducing area at the cost of N× higher latency. Intermediate values allow partial parallelism.

11.4 Experimental Setup

The exercise uses a purely simulated environment. Python notebooks with guidelines and datasets to implement and test the neural network models are provided:

Datastream.txt: Text file of ADC values and true labels. Roughly 10^5–10^6 consecutive samples corresponding to a calorimeter cell signal digitized every 25 ns. It includes three columns with Sample value, true amplitude and true time (not used in this exercise).

HLS4ML_Tutorial_PartI.ipynb. Python notebook with data modeling, neural model implementation, training, test and validation.

HLS4ML_Tutorial_PartII.ipynb. Python notebook that performs the conversion to HLS of the trained models, performs resources estimation and synthesizes the code for FPGA implementation.

models.py Description of the neural network models to be used.

To support the development, training, and hardware deployment of the neural-network models used in this exercise, a dedicated software environment is provided. The tools in this environment allows to explore the full workflow, from data inspection and preprocessing, to model design and training, and finally to the translation of trained networks into FPGA-compatible firmware. The package includes the following resources:

- **Dataset**: Datastream with monte-carlo simulated energy depositions for a single calorimeter channels. It provides true amplitude and time and the digital signal samples.

- **Software environment**: The framework is developed in Python v3 with scientific libraries required already pre-installed in the workstation. The main libraries used in this exercise are:

  Panda: Dataset management and data formatting.

  Pytorch: Neural Network implementation, training, test and validation.

HLS4ML: Used to translate the trained model into High Level Synthesis (HDL) code. It supports quantization, pipelining, and integration with FPGA toolchain environment.

- **FPGA targets**: We compare three Xilinx UltraScale+ families to illustrate resource constraints:

  Artix UltraScale+: (low-end, low-power) – limited logic and DSP count.

  Kintex UltraScale+: (mid-range) – more DSPs and memory.

  Virtex UltraScale+:(high-end) – massive DSP and memory.

- **FPGA toolchain**: Vivado/Vitis HLS and Vivado Design Suite for synthesis. We create an HLS project (through hls4ml) and then invoke Vivado HLS to synthesize an IP core. Resource reports (DSP, LUT, FF, BRAM usage) are examined.

11.5 Procedure

This section follows the sequence of the provided notebooks, with detailed steps and corresponding notebook cells listed. The two notebooks are Part I (data prep, training) and Part II (HLS conversion, synthesis). Open the Python notebook with Visual Code Studio (presintalled) and select a Kernel (Select Kernel–>Jupyter Kernel–>Python (System)).

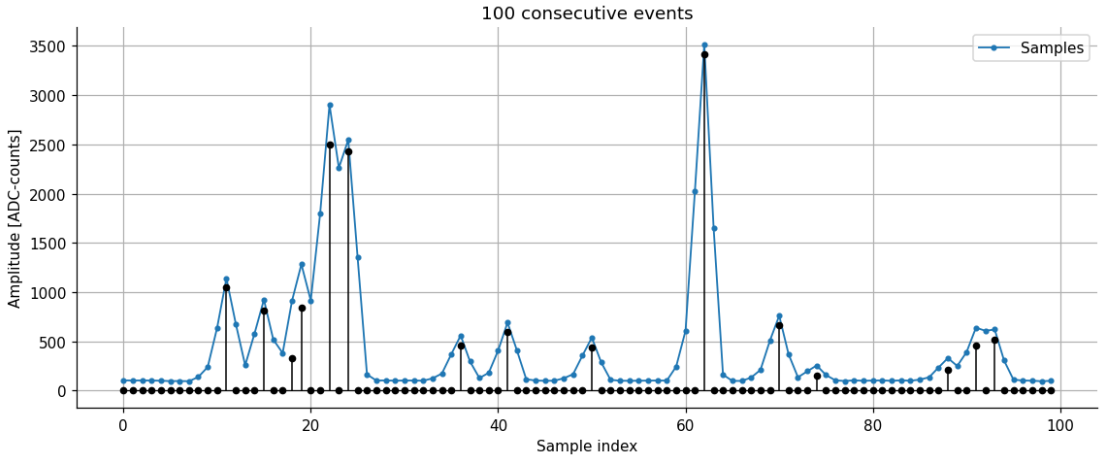


Figure 42: Example of the first 100 consecutive samples with the corresponding true amplitude for each bunch crossing.

11.5.1 Part I : Data preparation, model implementation and training

- **Step 1: Load and Inspect Data** (Part I, Cells 1 - 2) In Part I, Cell 1 executes code to load the input file: it reads the three-column data (ADC samples with true amplitude/time) into arrays. Immediately after, Cell 2 dumps or plots sample windows to verify data integrity. Typical checks include plotting a few windows of samples and confirming that the labels align with pulse peaks. Students should verify no missing values and inspect sample waveforms to understand the signal shape. This step ensures the raw waveform format is as

expected before further processing. As a verification step the first 100 events are plotted in Cell 2 (Figure 42).

> **Question 1:** Can you identify what are black and blue dots in Figure 42? What are the Y-axis units? what is its relationship with the energy deposited in the cell?.

- **Step 2: Format training data** (Part I, Cells 3 - 4) The code in Cell 3 (markdown header) and Cell 4 (code) formats the raw ADC stream into training tensors. It implements the sliding-window logic described above. The Window size in declared inside the models library. By default 9 samples are used, therefore Windows of length 9 are extracted, and the corresponding trueAmplitue and trueTime at the center of each window are assigned as labels. Students should confirm, e.g. by printing the first few windows, that each input-output pair is correct (done in Cell 4). In Cell 3 it is also selected which portion of windows are used for training the model, validate and test.

  > **Question 2:** Can you identify the sliding window id=7 (trueA=1048) in Figure 42? If there are 10k events, how many complete windows do we have? Why?

- **Step 3: Scale and normalize** (Part I, Cell 5) This cell prepares the input data so that the neural network can be trained reliably and later run in hardware under the same conditions. First, the signals are normalized so that all input values have a similar scale. This helps the network learn faster and more stably, because no single sample position dominates just because its numerical values are larger. The same normalization is applied to training, validation, and test data to ensure fair comparisons. We force all the normalization to min-max in the [0,1] range.

  Next, the normalized values are converted into integers. This step mimics what will happen in the FPGA, where floating-point arithmetic is not used. By scaling and storing the data as 16-bit integers, we make training and hardware execution consistent and avoid surprises when the model is deployed. This conversion is done by choosing how many bits represent the integer part and how many represent the decimal (fractional) part of each value. The total number of bits fixes the available range and precision.

  For example, with 16 total bits, allocating more bits to the integer part allows larger signal values but reduces precision, while allocating more bits to the decimal part improves precision but limits the maximum value. A scale factor is computed so that the largest expected normalized value fits safely within the chosen integer range. All values are then multiplied by this scale, rounded, and clipped if necessary. In this way, real-valued signals are represented as integers while preserving as much useful information as possible for FPGA processing.

- **Step 4: Train MLP and CNN (Part I, Cell 6-7)** Both models are trained with MSE loss and the Adam optimizer. A batch size of 128 and a fixed random seed (42) are used for reproducibility. Training runs on CPU only to match the FPGA inference environment. The training loop records train and validation MSE at each epoch.By default, the training is run for 5 epochs. An epoch corresponds to one complete pass of the entire training dataset through

Metric	MLP (float32)	CNN (float32)
MAE (ADC counts)		
RMSE (ADC counts)		
R2		
Bias (ADC counts)		

Table 2: Validation performance of Neural Network models.

the neural network. During each epoch, the model sees all training examples, computes the prediction errors, and updates its internal parameters (weights and biases) to reduce those errors. Using multiple epochs allows the network to gradually improve its reconstruction performance. In this exercise, the number of epochs is intentionally kept small to limit training time and to emphasize the workflow rather than exhaustive model optimization.After training, plot the learning curves (train vs. validation MSE) to diagnose underfitting or overfitting.

Question 3: Inspect the training and validation learning curves for both models. Are 5 epochs sufficient for the loss to converge, or do the curves suggest that additional epochs could further improve performance? Justify your answer based on the shape and trend of the curves.

- **Step 5: Evaluate Performance and save training model (Part I, Cell 8-10)** Then run inference on the held-out test set and compute four metrics for the amplitude channel: **MAE** (mean absolute error, in ADC counts), **RMSE** (root mean squared error), $\mathbf{R}^2$ (coefficient of determination; 1.0 = perfect), and **Bias** (mean signed error; non-zero bias indicates a systematic offset). Scatter plots of true vs. reconstructed amplitude reveal whether the model is unbiased across the full amplitude range. Cell 9 saves the trained model and Cell 10 prints the number of parameters per layer for both models. Once we have identified an optimum model.

Question 4: Interpret the obtained results and compare the performance of the MLP and CNN models on the validation or test set. Which model performs better for amplitude and/or time reconstruction, and how can you tell? Refer to quantitative metrics (e.g. MSE, MAE, or bias) and qualitative indicators (e.g. scatter plots). Complete Table 2. If the number of training epochs is increased, do both models improve in the same way?

Question 5: Examine the architecture of the MLP and CNN models (number of layers, neurons, filters, and trainable parameters). How do the differences in structure and total number of parameters relate to their reconstruction performance and to their suitability for FPGA implementation? Discuss which model is more efficient in terms of performance versus complexity and why.

11.5.2 Part II : HLS model conversion and FPGA implementation

Once a neural network has been designed, trained, and validated in software, the next step is to translate it into a hardware implementation. In this part, the model is converted to HLS, configured for fixed-point arithmetic and deterministic latency,

and prepared as an FPGA IP core for real-time deployment. We will focus on the MLP model which showed a slighly better performance.

- **Step 6: Configure hls4ml(Part II, Cell 11-12)** Reload the saved trained model and weights. Configure HLS4ML implementation parameters. The precision settings define how many bits are used to represent weights, activations, and internal accumulations, directly controlling numerical accuracy versus hardware cost. The reuse factor specifies how often arithmetic units (such as multipliers) are reused: a low reuse factor gives low latency but consumes more DSP resources, while a higher reuse factor saves resources at the cost of longer latency. Other options control pipelining and parallelism, ensuring deterministic, fixed-latency behavior. Together, these parameters allow the same model to be adapted to different FPGA families and resource budgets. By default, we start using a low-cost FPGA model from Xilinx Zynq family (part number : 'xc7z020clg400-1')

- **Step 7: Convert, compile and functional validation (Part II, Cell 13-15)**

  Use hls4ml library to generate the HLS project. Compile the project with invoking the Xilinx Vitis HLS C-simulation backend). Run a numerical comparison between software (floating point arithmetic) and firmware (fixed point arithmetic) : the maximum absolute difference between PyTorch and HLS predictions should be small but depends on the precision used in the fixed point arithmetic).

  Question 6: Are the predictions in the FPGA compatible with the CPU results? What is the largest difference? Is the regression compatible with the CPU?

- **Step 8: Synthesize the project(Part II, Cell 16)** Finally, call hls_mlp.build to invoke Vivado HLS synthesis. This will convert the HLS code into FPGA logic blocks. Examine the report to analyze the result of the implementation in the FPGA and complete the resources Table 3.

Resource	**MLP RF=1**	**MLP RF=32**	Available (Zynq)
DSP			220
FF			53,200
LUT			106,400
BRAM			140
Latency (ns)			250 ns (goal)

Table 3: Resources usage table.

Question 7: Complete Table 3.Can we fit the models in the selected low-cost FPGA? Modify the HLS4ML implementation parameters to increase the Reuse Factor to 32. What is the impact in the latency and in the resources usage? Can we now fit the model in the FPGA? Do we meet the latency requirement (250ns)?

11.6 Analysis and interpretation of the results

Presentation and analysis of the results obtained.

1. Implement the model in medium (Kintex Ultrascale+ xcku19p-ffvj1760-2-e) and advanced (Virtex Ultrascale+ xcvu19p-fsva3824-2-e) FPGA models. Evaluate the resources usage in these cases.

2. Modify the models architecture to reduce the number of parameters and the implementation configuration to find a model that can fit into the Zynq-FPGa model (xc7z020-clg400-1) and still meeting the latency requirements. Evaluate and compare the performance of the new model both in CPU and FPGA implementation.

3. Implement a model that can fit into the Kintex Ultrascale+ FPGA (xcku19p-ffvj1760-2-e) to reconstruct 100 channels in parallel with a maximum latency of 200 ns. Evaluate the performance of the model and the implementation in the FPGA.

4. Taking the full size MLP network with 64 neurons per layer, improve the performance of the FPGA implemented model changing the HLS4ML configuration parameters. Which configuration parameter affects the precision of the FPGA implementation? How it affects the quantity of resources used?

5. *Bonus*: Evaluate if the CNN model can be implemented in any of the FPGAs models and evaluate its performance.

Exercise 12 Gaseous Time Projection Chambers

12.1 Objectives

- **General Objective**: To understand the fundamental principles, operation, and signal analysis of a gaseous Time Projection Chamber (TPC) for particle detection.
- **Specific Objectives**:
  - To assemble and operate a basic cylindrical gas TPC.
  - To understand the generation of primary scintillation (S1) and electroluminescence (EL or S2) signals using an alpha radioactive source.
  - To learn about gas handling by flowing atmospheric-pressure argon.
  - To read out, trigger, and acquire signals using an oscilloscope.
  - To develop a Python-based analysis framework to evaluate S1 and S2 signals, measure drift times, and build an EL gain curve.

12.2 Professors

- Gonzalo Martínez
- Pau Novella
- Josh Renner
- Ander Simón

12.3 Theoretical Introduction

This section presents the theoretical basis necessary to understand the operation of the TPC.

12.3.1 Principles of TPC Operation

The TPC is a versatile particle detector capable of providing 3D track reconstruction and energy measurement. When an ionizing particle, such as an alpha particle (α), travels through a noble gas like argon, it scatters off the gas atoms, depositing its energy. This energy deposition occurs via two main processes:

1. **Excitation**: Argon atoms are excited and form excimers, which decay and emit vacuum ultraviolet (VUV) photons. This prompt light is called the **S1** signal.
2. **Ionization**: The particle removes electrons from the argon atoms, creating electron-ion pairs.

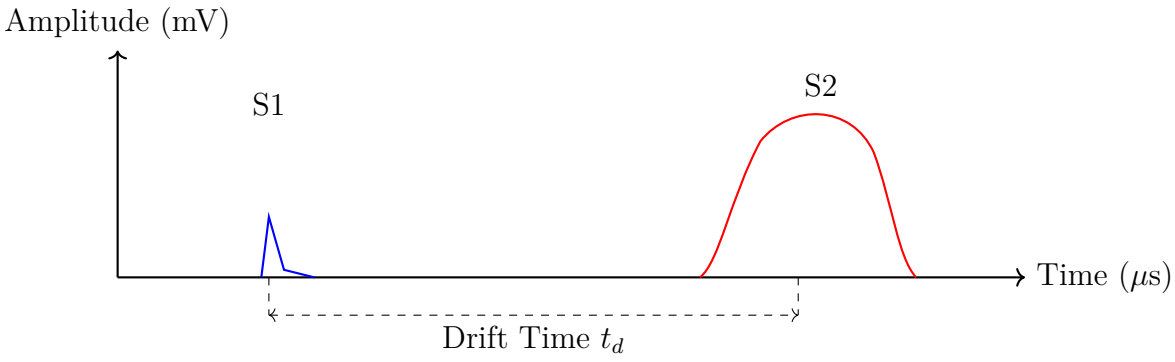
Some of the particle energy is also deposited as kinetic energy in the gas and does not result in excitation or ionization. An electric field applied across the *drift region* (between the cathode and the gate mesh) sweeps the freed electrons upwards toward the *amplification region* (between the gate and the anode mesh). Ions drift slowly towards the cathode.

12.3.2 S1, S2, and Electroluminescence

As the drifted electrons cross the gate mesh and enter the amplification region, they experience a much higher electric field. In this activity, the main amplification mechanism is **electroluminescence** (EL), also called proportional scintillation. The electrons gain enough energy between collisions to excite argon atoms without necessarily producing an avalanche. The excited argon then emits VUV scintillation photons through the same excimer decay process that produces the prompt **S1** light. The light is therefore essentially the same argon scintillation light as S1, but it is produced later and by a different mechanism: drifting ionization electrons are accelerated in the high-field EL region.

The delayed EL light is called the **S2** signal. Its size is approximately proportional to the number of drifted electrons and to the EL field between the gate and anode, until instrumental effects such as SiPM saturation or field non-uniformities become important. The time difference between the prompt S1 signal and the delayed S2 signal corresponds to the drift time of the electrons. Given a constant drift velocity v_d (dependent on the applied electric field and gas pressure), the vertical (z) position of the initial interaction can be determined:

$$z = v_d \times (t_{S2} - t_{S1}) \tag{16}$$



12.4 Experimental Setup

The experimental equipment consists of a small-scale, custom-built cylindrical TPC and standard laboratory read-out electronics:

- **Cylindrical Enclosure:** A cylinder with a 6 cm diameter serves as the gas containment vessel.
- **Electrode Meshes:** Three circular wire meshes define the electric field regions. They are maintained in position by long insulating supports.
  - **Cathode:** Placed near the bottom.
  - **Gate:** Placed 2 cm above the cathode, defining the *drift region*.
  - **Anode:** Placed 0.5 cm above the gate, defining the *amplification region*.
- **Alpha Source:** A disc-shaped radioactive α-emitter is placed directly on the cathode.

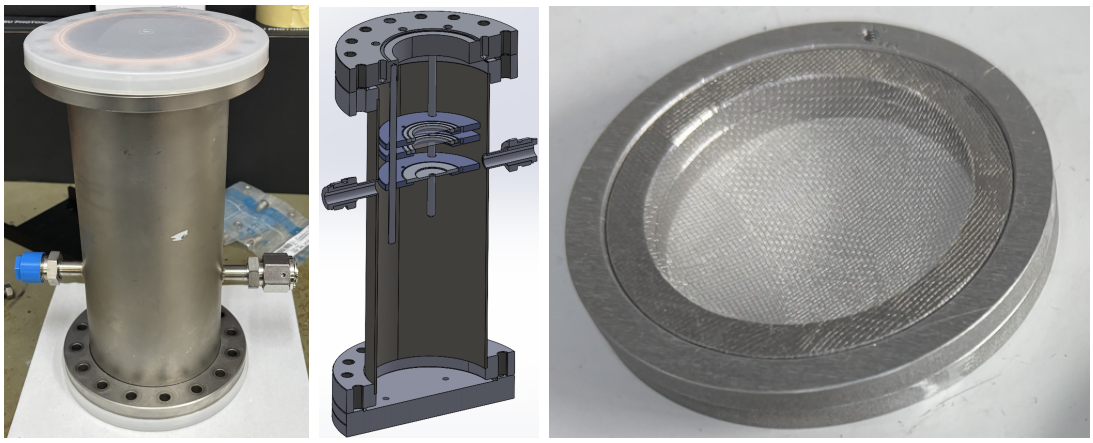


Figure 43: Cylindrical enclosure (left), CAD drawing showing the TPC interior (center) and one mesh structure (right)

- **Gas System:** A compressed bottle of Argon gas. The Ar gas is flowed into the chamber through one side port at atmospheric pressure and allowed to escape through the opposite side port.
- **Signal Readout:** Feedthroughs provide the High Voltage (HV) biases and extract the electrical signals from the meshes. The signals are fed into an oscilloscope for triggering and digitization, prior to saving to a computer for off-line analysis.

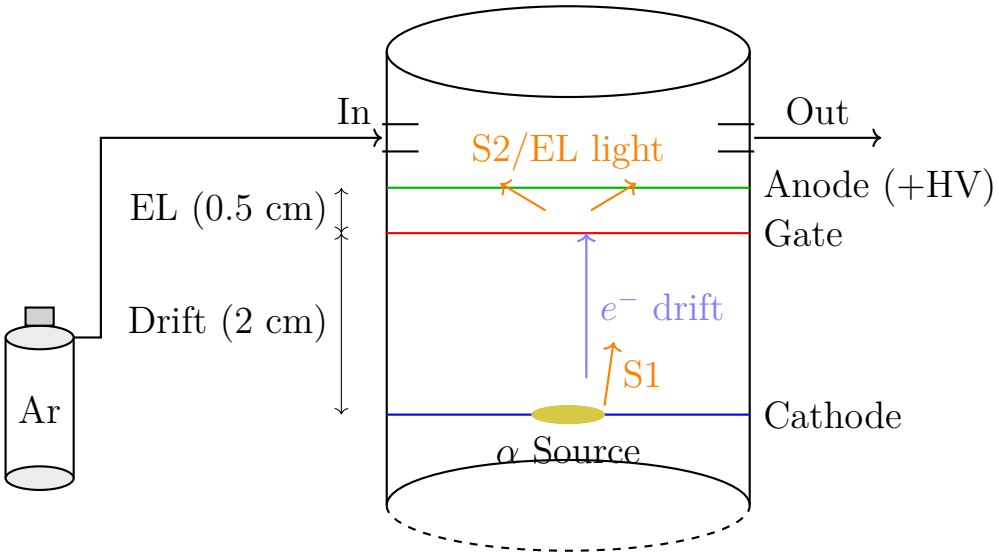


Figure 44: Schematic of the TPC

12.5 Procedure

The exercise should be carried out following these steps:

1. **Assembly & Gas Purging:**

- Ensure the chamber is sealed and the internal supports, meshes, and the α-source are correctly positioned.
- Connect the gas inlet to the argon bottle. Open the valve to start flowing argon gas through the system. Allow the gas to flow for some time to purge the internal air/oxygen out of the system, as impurities can attach to drifting electrons.

2. **High Voltage Biasing:**
   - With the gas flowing, slowly ramp up the high voltage for the cathode, gate, and anode.
   - Ensure the electric field in the drift region ($\sim$100s V/cm) is lower than the field in the amplification region (few kV/cm).
3. **Oscilloscope Setup:**
   - Connect the readout channel to the oscilloscope (see appendix 12.6 for details).
4. **Data Acquisition:**
   - Observe the waveforms on the oscilloscope. Identify the fast prompt S1 peak and the wider, delayed S2 peak.
   - Record a dataset of many waveforms (e.g., 1000 events).
5. **EL Gain Scan:**
   - Repeat the acquisition for several gate–anode voltages while keeping the drift field and gas flow fixed.
   - Start from a voltage where the S2 signal is visible but small, then increase the EL field in safe steps. At each setting, save the voltage values, oscilloscope settings, and a comparable number of events.
   - Watch the SiPM waveform shape as the light yield grows. If the peak flattens, clips, or stops increasing linearly, reduce the voltage or gain because the sensors or readout may be saturating.

12.6 Analysis and interpretation of the results

The last part of the exercise involves offline analysis of the acquired oscilloscope waveforms using Python over the saved data arrays. A typical analysis of TPC data consists of:

1. **Baseline subtraction:** A baseline value is computed for each waveform (using an early time window where no signal is present), and this value is subtracted from the entire waveform.
2. **Peak finding:** The S1 and S2 peaks are identified via a peak-finding algorithm.
3. **Pulse area integration:** The total charge/light collected in S1 and S2 is determined by integrating the pulses.

Using the provided code, the data can be saved in an "n-tuple" format containing key quantities, such as the determined baseline, S1 and S2 start times, and S1 and S2 integrals. The data can be explored further:

- **EL gain curve:** For each gate–anode voltage setting, plot the integrated S2 area versus the EL voltage or field.
    - Does the S2 area increase approximately linearly over part of the scan?
    - At high light levels, do the largest pulses show signs of SiPM saturation or oscilloscope clipping?
- **Drift time measurement:** Calculate the time difference $\Delta t = t_{S2} - t_{S1}$ for each event.
    - What is the distribution of drift times?
    - Given the drift length, estimate the electron drift velocity.

Appendix

Oscilloscope operation

The scope used during this exercise is shown in Figure 45. Each group of buttons and the channels are highlighted in colors. Connect each sensor to a different channel. Press the channel buttons on the right to select a channel and press again to show/hide its waveform in the display.

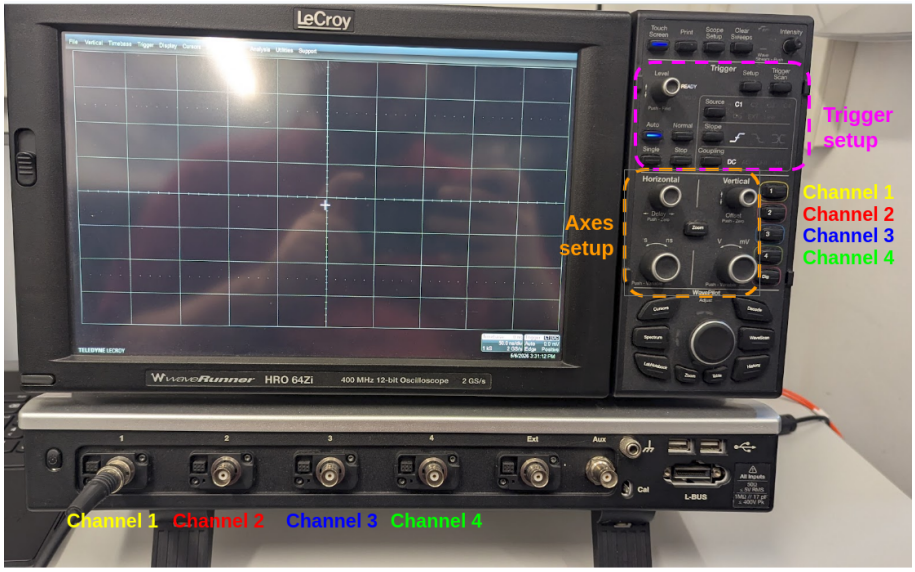


Figure 45: Scope channels and controls.

In the trigger setup section, select the channel to trigger on by repeatedly pressing the "Source" button. Select the first trigger mode using the "Slope" button. Select the DC coupling using the "Coupling" button. Adjust the trigger threshold using the "Level" knob until you observe similar waveforms being displayed. You can also adjust the trigger position in the screen using the "Horizontal knob" in the axes setup section.

Select an appropriate visualization window by using the lower knobs in the axes setup section. The left knob is used to adjust the range in the horizontal/time axis (Figure 46), while the right knob is used to adjust the range in the vertical/voltage axis (Figure 47). The horizontal axis is common for all channels, while the vertical one is specific to each channel.

Data acquisition

To acquire data automatically with the scope, click on the file tab and select "Save Setup..." as shown in Figure 48. This will open a menu at the bottom of the screen. Select "Waveform", "File", "All displayed" and "Excel" as shown in the Figure. Go to the "Auto Save" tab and click on the top "Browse" button. Navigate to your storage device and create a folder to store the data and pick a *meaningful* file name (e.g. a run number, voltages used, etc.). The scope will add a number to the file name to store multiple waveforms automatically. Finally, tick the box next to "Waveform" and click on the "Fill" button on the right of the screen. The scope will store each waveform in a separate file. To stop the acquisition, press the "Off" botton.

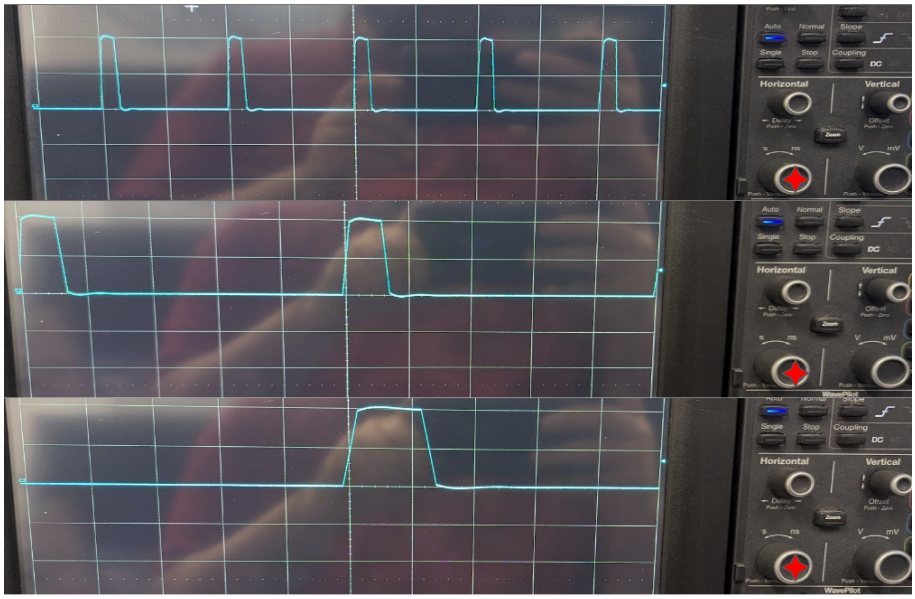


Figure 46: Display of the same waveform with different acquisition windows.

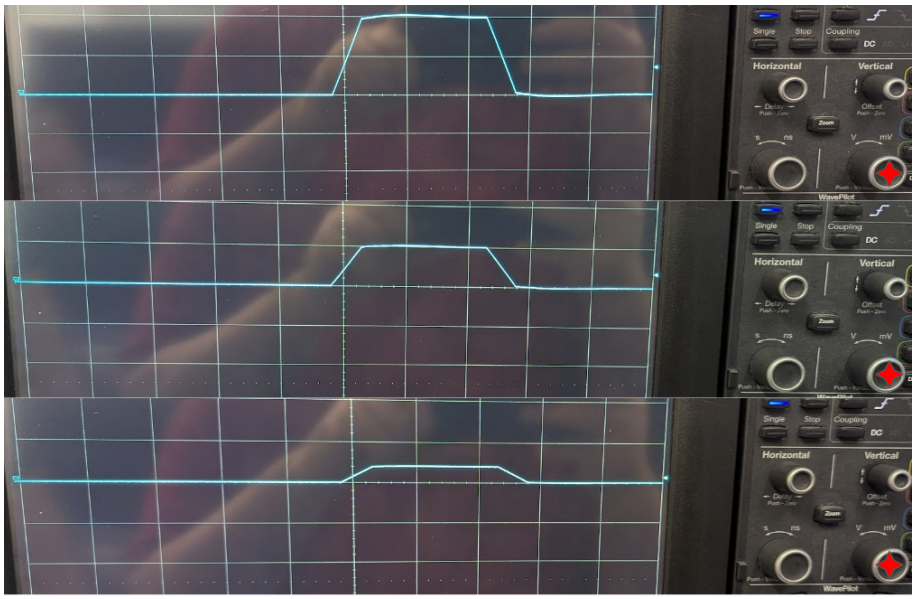


Figure 47: Display of the same waveform with different voltage ranges.

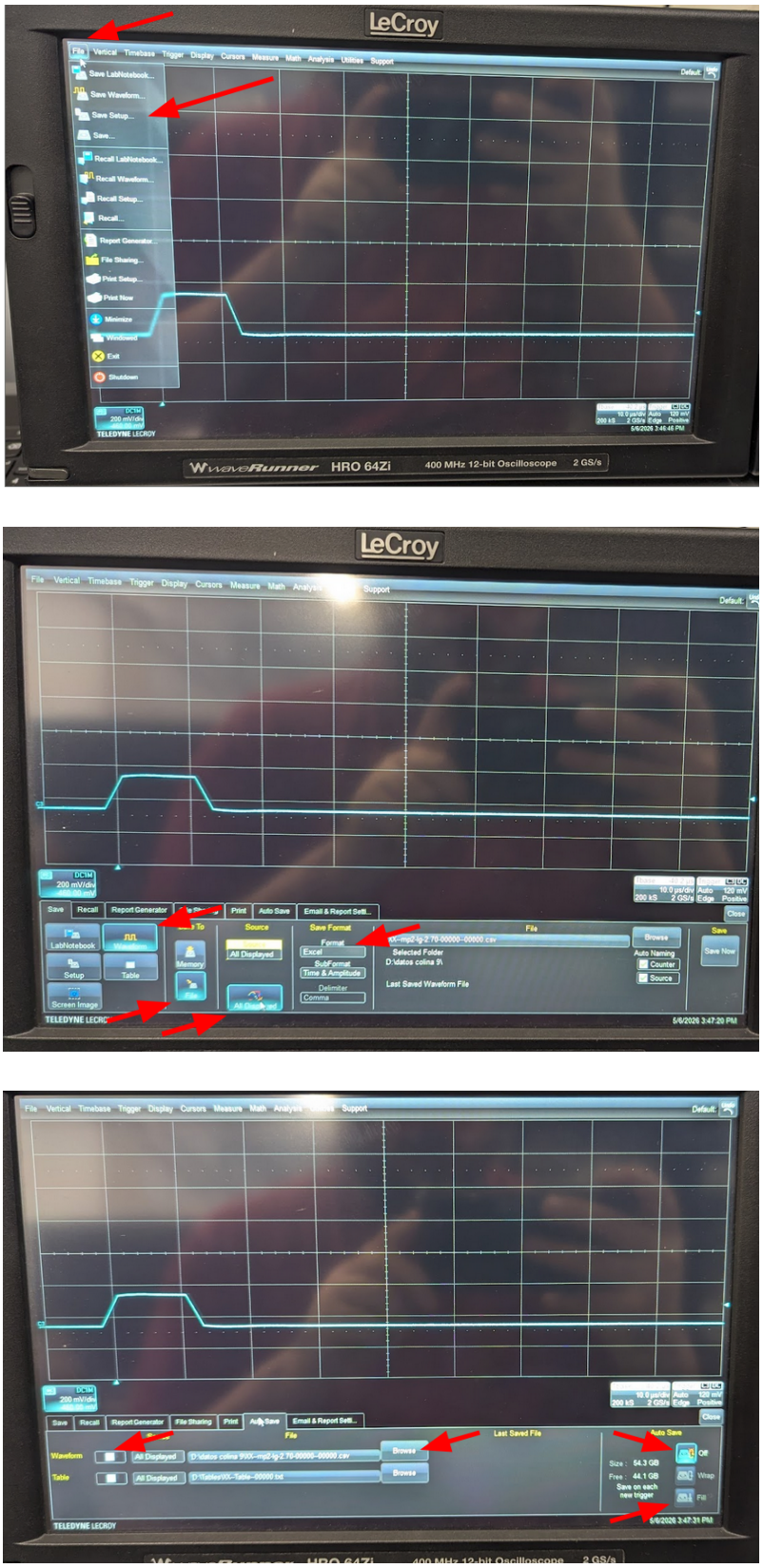


Figure 48: Acquisition setup

Exercise 13 Monte Carlo-based experiment simulation: silicon microstrip detector

13.1 Objectives

The aim of this exercise is to introduce students to particle physics simulations based on Monte Carlo methods, using a simplified but physically meaningful Geant4 application. The silicon microstrip detector from Exercise 7 is used here as a case study to illustrate how a real detector measurement can be translated into a computational model.

The purpose of this exercise is therefore not to provide a complete treatment of silicon-detector physics, which is already covered in other practical sessions, but to show how a detector problem is encoded in a Monte Carlo application, how the different software components are organized, and how the output of the simulation is interpreted.

At the end of the exercise, students should be able to:

- identify the main building blocks of a detector Monte Carlo simulation;
- understand the role of geometry, materials, primary generation, transport, detector response, and analysis in a simulation workflow;
- distinguish between what is simulated directly by Geant4 and what must be introduced through an additional detector-response model;
- run and modify a Geant4-based application inside a Docker environment;
- use a simplified simulation to reproduce qualitatively the response of a silicon microstrip detector under a scanned source;
- analyse the output of the simulation and extract simple observables from it;
- discuss the limitations of a simplified Monte Carlo model and propose possible improvements.

13.2 Professors

- Jordi Duarte-Campderros

13.3 Theoretical Introduction

13.3.1 Monte Carlo simulation in particle physics

Monte Carlo methods are numerical techniques based on random sampling. They are used to solve problems for which an exact analytical treatment is impractical or impossible, either because the underlying processes are intrinsically probabilistic or because the system is too complex to describe in closed form. In particle and detector physics, both situations occur naturally: the interaction of particles with matter is stochastic, and realistic experimental setups involve complicated geometries, multiple materials, many coupled physical processes, and detector effects that must all be taken into account simultaneously.

For this reason, Monte Carlo simulation has become one of the standard tools of experimental particle physics. It is used throughout the full life cycle of an experiment: to design detectors, optimize their geometry, estimate acceptance and efficiency, study backgrounds, validate reconstruction strategies, and interpret measured data. In many cases, Monte Carlo simulation is the only practical way to connect an underlying physical process — such as the passage of a particle through a detector — with the actual observables recorded by the readout system.

From a computational point of view, a Monte Carlo simulation replaces an intractable exact calculation by the generation of many individual random realizations, or *events*, sampled from the relevant probability distributions. Each event represents one possible history of the physical system. Statistical properties of the ensemble, such as mean values, fluctuations, efficiencies, or response distributions, are then estimated from the generated sample. In this sense, Monte Carlo simulation provides both a predictive model and a controlled numerical laboratory in which the behaviour of a detector can be explored.

Once a particle-physics problem has been physically motivated, it must be translated into a simulation model. In practice, this usually involves constructing a simplified but meaningful representation of the system, keeping only the ingredients that are relevant for the observables of interest. As illustrated in Fig. 49, the workflow typically proceeds from a physical case study to an abstraction of the problem, then to its implementation in a simulation code, followed by the production of output data and their post-processing. The interpretation of the results may in turn reveal limitations of the model and suggest possible improvements.

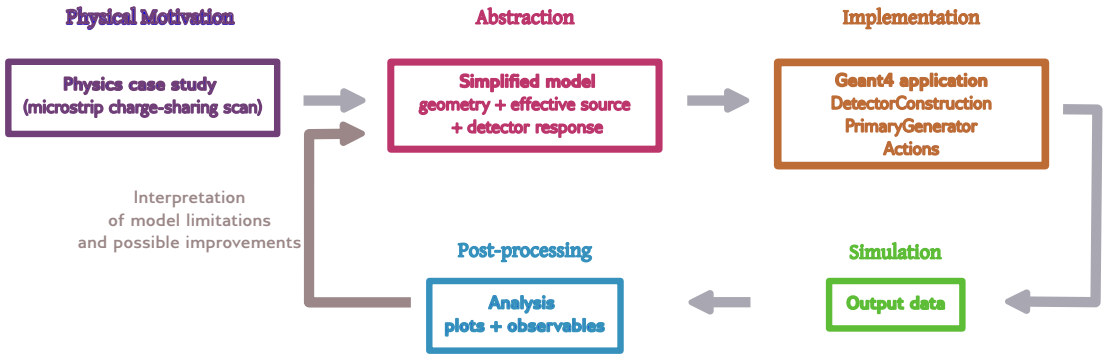


Figure 49: Workflow of the Monte Carlo exercise. The microstrip experiment is used as a case study to teach abstraction, implementation and interpretation.

In detector applications, this translation from a physics problem to a simulation model usually requires the definition of:

- the detector geometry;
- the materials;
- the primary particles or excitation source;
- the physical processes to be considered for each particle type;
- the quantities to be scored or stored;

- the analysis procedure used to convert the simulation output into physics observables.

The purpose of this exercise is therefore not to provide a complete treatment of silicon-detector physics, which is already covered in other practical sessions, but to show how a detector problem is encoded in a Monte Carlo application, how the different software components are organized, and how the output of the simulation is interpreted.

13.3.2 Main building blocks of a Geant4 application

Geant4 is a general-purpose toolkit for the simulation of the passage of particles through matter. Rather than being a single-purpose program, it provides a modular framework in which the user defines the ingredients of the problem to be simulated. A Geant4 application is therefore built by combining a number of software components, each associated with a specific part of the physical or computational model.

The first essential component is the *geometry*. The user must define the world volume, the detector volumes, their dimensions, relative positions, and the materials from which they are made. This step translates the physical layout of the detector into a computational representation on which particle transport can be performed.

A second component is the *primary event generation*. Here, the user specifies what enters the detector: for example, a particle beam, radioactive decays, cosmic rays, optical excitation, or, as in this exercise, an effective source representing a scanned laser beam. The primary generator therefore defines the initial conditions of each simulated event, such as particle type, energy, direction, and position.

A third component is the *physics configuration*. Geant4 provides libraries of physical processes describing electromagnetic, hadronic, optical, and decay interactions. These are usually grouped into a *physics list*, which determines which processes are active in the simulation for each particle type and in which energy range they are valid.

The execution of the simulation is controlled through a set of *user actions*, which allow the user to intervene at different levels of the event loop, for example at the beginning or end of a run, at the event level, or during the stepping of particles through matter. These actions are commonly used to initialize output, inspect the event flow, accumulate observables, and write results to file.

In many detector studies, the quantities of interest are not simply the raw transport variables delivered by Geant4. Energy deposits, hit positions, or particle trajectories often have to be converted into more detector-like observables through an additional response model. This may include charge sharing, digitization, thresholds, electronic noise, or reconstruction algorithms. For that reason, a complete detector simulation often combines the Geant4 transport layer with a separate detector-response and analysis layer.

Figure 50 summarizes these software components and shows how the Geant4 transport layer is complemented by a simplified detector-response and analysis chain in the present exercise.

13.3.3 Transport, detector response, and analysis

Geant4 handles particle transport through the detector geometry and the interaction of particles with materials. However, the detector signal seen by the readout

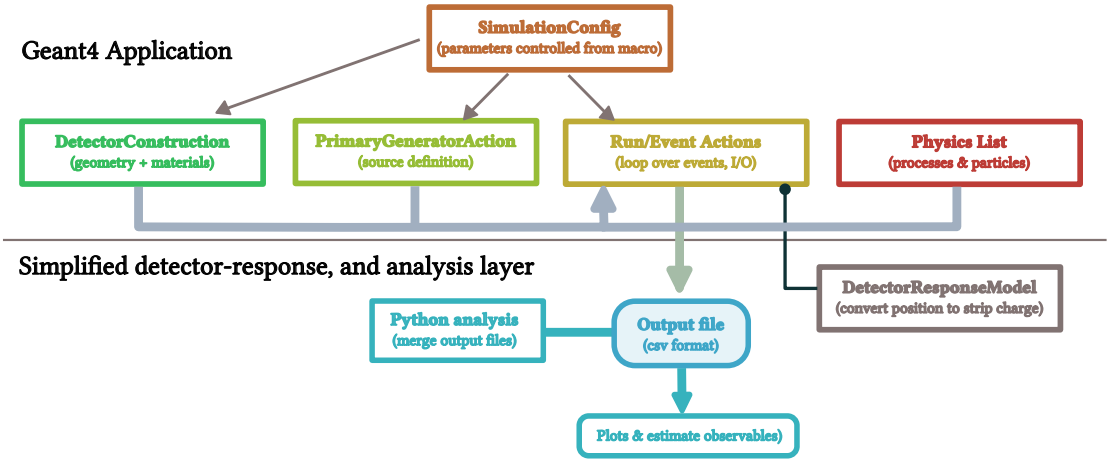


Figure 50: Main software blocks of the Geant4 case study. The diagram separates the Geant4 application layer from the simplified detector-response and analysis layers.

electronics is not always obtained directly from transport quantities alone. In many realistic cases, an additional detector-response model is required in order to translate the simulated interaction into the measured observable.

This separation between transport and detector response is a key concept in detector simulations and is one of the central learning goals of the present exercise.

13.4 Experimental Setup

In this exercise, the "experimental setup" is a simulation environment implemented with Geant4 and distributed through a Docker container. The goal is not to reproduce the full experimental system of Exercise 7 in all its complexity, but to construct a simplified simulation model that retains the ingredients necessary to understand the structure of a detector Monte Carlo application and to obtain an interpretable output.

13.4.1 Scope of the simulated problem

The physical case study considered here is inspired by the scanned-laser charge-sharing measurement of a silicon microstrip detector. The detector physics and the corresponding experimental workflow are treated in detail in Exercise 7; in the present exercise, that system is used only as a convenient example with a simple geometry and an easily interpretable response.

The simulation aims to reproduce, in a simplified way, the variation of the collected charge in neighbouring strips as a function of the source position. The main output observable is therefore the collected charge as a function of scan position. From that output, simple derived quantities such as strip pitch, effective strip width, and binary spatial resolution can be estimated.

In the corresponding experiment, a red laser is scanned across the detector surface along the direction perpendicular to the strips. Since the light is absorbed close to the sensor surface, it generates charge carriers locally in the illuminated region. Depending on the incidence position, the generated charge may be collected mainly by a single strip or shared between neighbouring strips. By recording the collected charge

as a function of the scan position, one obtains a characteristic charge-sharing pattern: the response is reduced when the laser is incident on the strip metallization, while charge is generated and collected when the laser illuminates the inter-strip region. From this position-dependent response, one can identify geometrical quantities such as the strip pitch and the effective strip width, and also study the spatial resolution expected from the detector. Further details on the experimental method and its physical interpretation are given in Exercise 7.

13.4.2 Geometry and materials

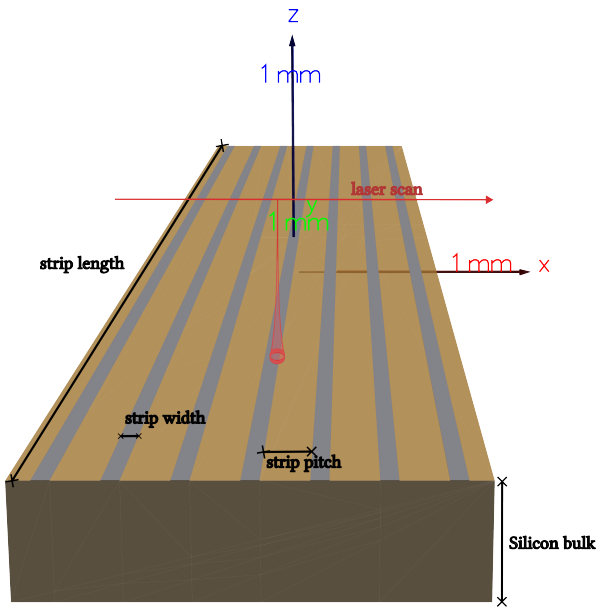


Figure 51: Simplified detector geometry and scan configuration used in Exercise 13.

The simulated geometry contains only the detector elements required for the case study:

- a world volume;
- a silicon sensor bulk;
- several parallel aluminium strips placed on the sensor surface.

The geometry is implemented in the `DetectorConstruction` class, which defines the detector dimensions, the relative placement of the volumes, and the corresponding materials. Figure 51 illustrates the simplified detector geometry used in the exercise.

13.4.3 Source model and simplifying assumptions

The source is represented by an effective scanned excitation rather than by a detailed simulation of the full laser system. In particular, the exercise does not attempt to model the complete optical transport of the 980 nm laser used in the experimental setup. Instead, the source is described through a simplified model characterized by its scan position and an effective transverse beam size.

This source model is implemented in the `PrimaryGeneratorAction` class. Its purpose is not to describe the full laser hardware, but to provide a controlled and configurable input that can be used to study how the detector response varies with position.

13.4.4 Detector-response model

The quantities of interest in this exercise are not the raw transport quantities delivered by Geant4, but strip-level observables that resemble the response of a detector readout. For that reason, a simplified detector-response model is introduced in addition to the Geant4 transport layer.

This model converts the source position into collected charge in neighbouring strips. In particular, it is used to describe, in an effective way, effects such as the finite source width, the sharing of charge between adjacent strips, and the reduced or suppressed response associated with the strip metallization. The aim is not to reproduce the full experimental readout chain, but to provide a minimal and physically meaningful mapping between the simulated source and the final observable.

13.4.5 Software structure of the application

The main software blocks of the application are summarized in Fig. 50. The detector geometry and materials are implemented in the `DetectorConstruction` class, the source model is defined in the `PrimaryGeneratorAction`, and the execution flow and output handling are controlled through the different user actions. The simplified detector-response model is implemented separately and is used to convert the simulated source position into strip- level observables that can be analysed and compared.

The `SimulationConfig` class provides the central configuration point of the application. It contains the messenger interface, which allows the user to modify selected simulation parameters through Geant4 macros rather than by recompiling the code. In this way, the main parameters of the simulation can be adjusted interactively while keeping the software structure unchanged.

13.4.6 Nominal simulation parameters and output

The nominal detector and scan parameters are chosen so as to resemble the geometry used in the corresponding microstrip exercise:

- silicon bulk thickness: $300\,\mu$m;
- strip pitch: $160\,\mu$m;
- aluminium strip width: $44\,\mu$m;
- scan step: $10\,\mu$m;
- total scan range: at least $350\,\mu$m.

The simulation produces output files containing, for each scan position, the charge associated with neighbouring strips. These files are then processed by a Python analysis script, which combines the individual outputs, generates the corresponding plots, and estimates a few simple observables.

The software environment is distributed through Docker. Students are assumed to have Docker installed on their laptops, and no local Geant4 installation is required.

13.5 Procedure

The procedure is divided into four parts: understanding the simulation structure, running the nominal model, modifying selected parameters, and analysing the results.

Part A: Understanding the simulation application

1. Start the Docker container and verify that the Geant4 executable and the Python analysis scripts are available.
2. Inspect the repository structure.
3. Identify the files responsible for:
    - detector geometry,
    - source definition,
    - detector-response model,
    - run/event control,
    - output writing,
    - analysis.
4. Run the visualization macro and inspect the detector geometry.
5. Identify in the geometry:
    - the silicon bulk,
    - the strip direction,
    - the strip pitch,
    - the aluminium strip regions,
    - the scan direction of the source.

Part B: Running the nominal simulation

1. Locate in the application where the nominal simulation parameters are defined.
2. Identify which parameters can be modified through Geant4 macros.
3. Run the default simulation corresponding to the nominal scan.
4. Produce the output files containing, for each scan position, the charge collected in several neighbouring strips.
5. Run the analysis script to plot the mean collected charge as a function of scan position.

Part C: Modifying the model

1. Modify the effective source size and repeat the scan.
2. Compare the resulting output with the nominal one.
3. Optionally, modify one geometry parameter, such as strip width or pitch, and repeat the simulation.

Part D: Interpreting the output

1. Identify the main features of the output curves.
2. Determine which features are controlled mainly by geometry and which are controlled by the detector-response model.
3. Estimate a few simple observables from the output:
   - strip pitch,
   - effective strip width,
   - binary spatial resolution.

13.6 Analysis and interpretation of the results

Using the simulation output, answer the following questions:

1. Which parts of the simulation are defined explicitly in Geant4?
2. Which parts of the final observable are introduced through a simplified detector-response model?
3. What is the main output quantity produced by the simulation?
4. Describe the shape of the collected-charge distributions for neighbouring strips.
5. Identify the minima associated with the aluminium strips.
6. Estimate the strip pitch from the distance between repeated features in neighbouring strips.
7. Estimate the effective strip width from the width of the suppressed-signal region.
8. Determine the binary spatial resolution using the strip pitch.
9. State clearly the main simplifications of the model.
10. Propose one improvement that would make the simulation more realistic.

13.7 Bibliography

- Recommended references for the general concepts of Monte Carlo methods and particle transport:
  - N. Metropolis and S. Ulam, *The Monte Carlo Method*, Journal of the American Statistical Association **44** (1949) 335–341.
  - J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen, London, 1964.
  - M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, 2nd ed., Wiley-VCH, 2008.
  - J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*, Dover Publications, New York, 2008.
  - F. Salvat, J. M. Fernández-Varea and J. Sempau, *PENELOPE – A Code System for Monte Carlo Simulation of Electron and Photon Transport*, OECD Nuclear Energy Agency, Issy-les-Moulineaux, 2011.
- Recommended introductory material for students with no previous Geant4 experience: JLab lecture, ELI-NP lecture, and official Geant4 tutorials.
- Recommended reference for students who want to go beyond the present exercise and understand in more detail the structure of a Geant4 application: Geant4 User's Guide for Application Developers.
- Official Geant4 references:
  - S. Agostinelli et al., *Geant4: a simulation toolkit*, Nucl. Instrum. Meth. A **506** (2003) 250–303.
  - J. Allison et al., *Geant4 developments and applications*, IEEE Trans. Nucl. Sci. **53** (2006) 270–278.
  - J. Allison et al., *Recent developments in Geant4*, Nucl. Instrum. Meth. A **835** (2016) 186–225.
- Recommended general references on detector techniques:
  - W. R. Leo, *Techniques for Nuclear and Particle Physics Experiments.*
  - G. F. Knoll, *Radiation Detection and Measurement.*
- Exercise 7 of this practicum, *Basic Principles of Silicon Microstrip Detectors.*

Exercise 14 Applications of Artificial Intelligence to simulation and reconstruction

14.1 Objectives

This exercise provides an introduction to the application of modern **Machine Learning** (ML) techniques to common tasks in particle physics such as the simulation of the response of a detector or the process of track reconstruction. These tasks are critical for particle physics experiments and have been solved with traditional methods over the years. Methods based on ML can present some advantages mostly because of their data-driven nature and speed. Since the number of ML algorithms and possible applications is very large, emphasis will be put on the underlying key concepts, with the hope that after this exercise, you will be able to easily apply other ML algorithms on your own working areas. This exercise will be done over notebooks on the google-colab platform, as a consequence, this text will focus more on the theoretical introduction since the notebooks will guide you through the exercise. The sub-objectives of this exercise are the following:

1. Training a **Generative Adversarial Neural Network** (GANN) to simulate the response of a **Low-Gain Avalanche Diode** (LGAD) and its readout ASIC.

2. Training a **Graph Neural Network** to perform track reconstruction on data from a simulated tracker.

14.2 Professors

- Pablo Martínez Ruiz del Árbol

14.3 Theoretical Introduction

Artificial intelligence (AI) is one of the boldest concepts in our collective consciousness. However, the notion is elusive because the concept of intelligence is not well-defined. The confusion grows even more when the general public identifies AI with machine learning (ML) algorithms, which only represent a particular strategy or subfield within AI. There is a lot of hype surrounding ML in our society, where firms associated with this technology have escalated to the pole position in terms of value. The debate about the impact of ML is ubiquitous, flourishing among all kinds of techno-tribes who embrace positions ranging from apocalyptic visions to transhumanist delusions.

Immersed in this chaos, it turns out that these methods work surprisingly well for many applications and particle physics is not an exception. In its most basic expression, ML can be seen as a sophisticated procedure for fitting data. That good-old linear regression that you used back in your first year of university contains all the basic concepts, power, and defects of ML algorithms.

14.3.1 The Machine Learning paradigm

Consider a set of variables or **features** $x = (x^0, ..., x^N)$ mapped to another set of variables or **labels** $y = (y^0, ..., y^M)$ through some unknown function f in such a way that $y = f(x)$. Machine Learning techniques try to approximate f using a model $m(\theta^0, ..., \theta^K)$ that depends on the parameters $\theta = (\theta^0, ..., \theta^K)$. Supervised ML methods estimate the model m by minimizing an **objective or loss function** $L(y, x; \theta)$ that provides a measurement of how close the prediction $m(x)$ is to the expected result. Supervised ML methods require a **training dataset** for which both the features and the labels are known $\{x_i, y_i\}$. The labels of the training dataset are usually referred to as **ground truth**. Training the model is equivalent to finding the parameters θ that minimize the Loss function for the training dataset:

$$\theta^{min} = arg \min_{\theta} L(x, y; \theta) \tag{17}$$

Supervised ML methods are extensively applied to tasks such as classification or regression among many others.

On the contrary, unsupervised ML methods, do not require a training dataset with known ground truth. The Loss function of these methods exploits internal requirements to learn from the features by trying to fulfil those requirements. Clustering is one of the most classic applications of unsupervised ML.

After training, ML workflows perform validation and testing. Both processes require two additional datasets independent from the training dataset: the **validation dataset** and the **testing dataset**. The validation dataset is used to tune the so-called **hyperparameters** of the model such as the number of parameters, the learning rates, or other external parameters defining the model m or the training process. Once the hyperparameters have been fixed, the testing dataset is used to evaluate the performance of the system. Since the testing dataset is independent from the training dataset, the testing phase quantifies the ability of the method to **generalize** the knowledge acquired to new samples.

14.3.2 Gradient descend and stochastic gradient descend

Machine Learning methods rely on the ability to find the minimum of complicated, multi-variable functions. Indeed, part of the huge success of these methods in the last couple of decades is related to the evolution of hardware architectures such as **Graphical Processing Units** (GPU) that can be extremely efficient to perform this task.

The most widely-used methods to minimize complicated functions use the concept of the **gradient**. The gradient ∇_θ of a function $-L(\theta)$ is a vector of components $\frac{\partial L}{\theta_i}$, that points in the direction of maximum descend of the function L at a given point. This means that a step $\Delta\theta$ in the parameter space given in the direction of the gradient is the most efficient way to approach the minimum.

The algorithm of **gradient descend** uses the gradient of the function to lead the trajectory towards the minimum. In addition to the gradient, this algorithm

requires to establish an strategy to define how large the step in the direction of the gradient will be. This strategy is usually called the **optimization strategy**. Once this is defined, the algorithm is simple: the gradient is calculated at a point θ_0, the next point is found as $\theta_1 = \theta_0 - s\nabla_\theta$ where s is the size of the step given by the optimization strategy, and the process is repeated until the variation of the function is so small that it can be considered to be at the minimum.

Gradient descend is affected by two problems. On the one hand, for very large datasets, the evaluation of the Loss function is challenging since it requires to load the whole dataset into memory, which may be impossible. On the second hand, complicated Loss functions may have local minima where the gradient descend may get stuck.

These complications lead to the introduction of the **stochastic gradient descend**. In this algorithm, the gradient is not calculated for the full Loss function but only for the Loss function evaluated in a sub-set or **batch** of the total training sample. With this procedure the problem of the memory is trivially solved since only the batch has to be loaded in memory. Surprisingly, this modification helps as well to escape from local minima. The gradient estimated on a batch of the training sample, points only approximately in the direction of the maximum descend. This adds an stochastic component that very often helps the optimization strategy to jump out of the local minima.

Modern ML frameworks such as *Tensorflow* or *Pytorch* implement stochastic gradient descend according to the following scheme. First, the full dataset is read in blocks corresponding to the batch size. For each batch the loss function is evaluated, the gradient estimated, and the parameters updated. This process repeated for the full dataset defines what is called an **epoch**. The full process is repeated for a large number of epochs to increase the chances of finding the global minima.

14.3.3 Typical problems of Machine Learning methods

A very common problem observed in ML methods is related to the lack of **capacity** of the model. This happens when the complexity of the function f is larger than the complexity of the model $m(\theta)$, which usually means that the number of parameters is too low to account for all the characteristics of the function. In this situation, the value of the Loss function after minimizing is very large because even in the minimum $m(\theta)$ is very different to f. This problem is usually referred to as **bias problem**.

Sometimes the capacity of the model is too large for the complexity of f. When this happens, the model will tend to fit not only the overall trend dictated by f but also data fluctuations. In this situation, the value of the Loss function is very small after training because the model is actually accounting for statistical fluctuations. However, the value of the Loss function explodes in the validation and testing datasets, since the model is hyper-adapted to the training dataset and does not generalize well with new samples. This problem is usually referred to as **overfitting**.

14.3.4 Artificial Neural Networks

The model $m(\theta)$ used in ML methods is not constrained to any particular functional form. Particular choices on the typology and structure of $m(\theta)$ give place to different techniques that usually share a common characteristic: the calculation of the gradient is very fast. **Artificial Neural Networks** (ANNs) are probably the most popular choice nowadays. Its name emerges from a certain resemblance to brain neurons although probably the name hides more marketing than physical or medical sense. These networks are formed by several layers of nodes or **neurons**. A neuron is a system that receives as input a vector of features $x = (x^0, ..., x^N)$ and performs a linear transformation $z = w^0 + \sum_i w^i x^i$, followed by a non-linear transformation $o = \sigma(z)$ where w^i are parameters of the models or **weights** and $\sigma(z)$ is the so-called **activation function**. Common activation functions are the sigmoid function, the hyperbolic tangent function, or the **ReLU** defined as $ReLU(z) = z$ if $z > 0$ and $ReLU(z) = 0$ otherwise. All these functions have derivatives that can be calculated efficiently.

Neurons in a layer receive input from the neurons in the previous layer and send their output to the next. Let o_{j-1} be a vector containing the output of each of the neurons in layer $j-1$ expanded with a 1 as follows: $o_{j-1} = (1, o^1_{j-1}, ..., o^{N_j}_{j-1})$. If this vector is given as input to the neuron i of the layer j, the output will be $o^i_j = \sigma(w^i_j o_{j-1})$ where w^i_j is the vector of weights for neuron i in layer j. This operation can be condensed in a matrix operation if all the w^i_j are considered as the rows of a matrix w_j, in such a way that $o_j = \sigma(w_j o_{j-1})$. In this scheme o_0 is defined to be the input vector to the neural network and o_M the output, where M is the number of layers. An ANN is just a collection of matrix multiplications followed by the action of a non-linear activation function. Indeed, the final output $o_M = \sigma(w_M o_{M-1}) = \sigma(w_M \sigma(w_{M-1} o_{M-2})) = \sigma(w_M \sigma(w_{M-1} \sigma(...\sigma(w_1 o_0))))$.

The Loss function of a model based on a ANN should measure the difference between the ground truth expressed by the labels y^i of the training dataset and the predictions o^i_M. This means that the Loss function is $L(y^i, o^i_M)$. In order to minimize this function, the gradient $\nabla_{w^i_j} L$ is required. But given the structure of the calculation, the chain rule of derivatives can be applied in such a way that the gradient can be easily estimated as a product of matrices:

$$\nabla_{w^i_j} L(y^k, o^k_M) = \frac{dL}{do^k_M}\frac{o^k_M}{dw^i_j} = \frac{dL}{do^k_M}\frac{o^k_M}{z^k_M}\frac{dz^k_M}{w^i_j} = \frac{dL}{do^k_M}\frac{o^k_M}{z^k_M}\frac{dz^k_M}{do^k_{M-1}}\frac{do^k_{M-1}}{w^i_j}... \tag{18}$$

This algorithm is known in the literature as the **back-propagation algorithm** and it is at the heart of the ANNs success.

14.3.5 Generative ML: Adversarial Neural Networks

Artificial Neural Networks were traditionally used to perform tasks such as regression or classification. In the last two decades, ANNs have been used for other higher-level tasks such as the generation of synthetic datasets. This paradigm is called **Generative ML**. There are different methods implementing generative ML but all of them share a central concept: **the latent space**. This space expands the

number of features of the ANN with a set of additional features that receive random numbers. This is a strategy to input randomness to the ANN and opens the possibility for the ANN to generate new events according to a given, learned distribution.

For the shake of brevity this text will only discuss the case of **Generative Adversarial Neural Networks** (GANNs) [1]. This architecture involves two different ANNs: a **Generator** and a **Discriminator**. The Generator takes as input a list of features and a list of random numbers from the latent space and it is supposed to output numbers that follow a given distribution. The Discriminator plays the role of a judge in the sense that it is exposed to both samples produced by the Generator and real samples from the training dataset and has to determine to which category they belong to. The Loss functions of the Generator and the Discriminator are connected in such a way that, as the Generator is trained it produces more and more realistic samples to try to cheat the Discriminator. At the same time, the Discriminator learns to distinguish between real and synthetic samples, forcing the Generator to be more and more refined.

The structure of the Loss function and the training procedure is complicated in GANNs. Several approaches are available in the market. This text will focus in one of the simplest forms of training. Let $G(x)$ be the name of the Generator in such a way that given a feature vector combining real features and features from the latent space, x, $o = G(x)$ is a synthetic sample or fake event. Let $D(o)$ be the name of the Discriminator in such a way that given one event o, $y = D(o)$ is a single label that tells whether o is a real or fake sample. In this context, the training process will have two sequential steps: first the Discriminator will be optimized while the Generator is static and then the Generator will be optimized while the Discriminator is static. For a batch of real events o^i, the Loss associated with the Discriminator step is $L = E[D(G(x^i))] - E[D(o^i_{real})]$, where $E[]$ denotes the expected value. The Loss function associated with the Generator will be simply $L = -E[D(M(x^i))]$. Using this logic, the minimization of the Discriminator will try to assign negative values when applied on fake samples and positive values when applied on real samples. The minimization of the Generator, on the other hand, will try to produce fake samples that produce positive values in the discriminator. Therefore, there will be a competition between both Loss functions.

14.3.6 Other ML architectures: Graph Neural Networks

In this introduction, ML methods have been only applied to numerical vectors as features and labels. These methods can be applied to other kinds of data such as pictures (matrices of pixels) or to mathematical **graphs** among others. The former case gives place to the so-called **Convolutional Neural Networks** where additional concepts such as the **Convolution** are introduced. The latter case gives place to the so-called **Graph Neural Networks** (GNN).

The input to a GNN is a graph. A graph is an object composed by two sets V and E and denoted as $G = \{V, E\}$. The set V contains a group of vertices or **nodes** and the set E contains a group of edges that connect pairs of nodes. Each of the nodes may contain a set of features x and belongs to a given category denoted by

label y. The edges can contain a weight that quantifies its importance.

There are several types of GNN in the market. For the shake of brevity, this text will only discuss **Graph Convolutional Networks** (GCN)[2] and the variety that will be used in this exercise, **SAGEGraph**. Graph Convolutional Networks receive as input all the information of a given graph: the nodes, the features for each node, the labels and the collection of edges. The flow of the information has two stages generically called **encoding** and **decoding**.

In the encoding phase the input is passed through a series of regular neuron layers with the aim of producing and output that is usually referred to as **embedding** and forms a more suitable representation of the initial data. The only difference between a GCN and a regular ANN is the fact that the input features to every neuron are combined with the input features of nodes that are connected to one in particular. This mathematical operation that in simple cases may be just a weighted mean gives the name **Convolutional**.

The decoding phase involves a regular ANN that uses the embedding to perform some kind of prediction. There are different working modes: some GCN make predictions on the labels of the nodes, others predict the weights of the edges, etc. The SAGEGraph architecture is a GCN that uses a particular function in order to perform the convolution. The details of that function go beyond the spirit of this introduction and will be only given as a reference [3].

14.4 Experimental Setup

This exercise is divided in two parts both of them available as google-colab notebooks.

14.4.1 GANN to simulate the response of an LGAD sensor

The first exercise aims at training a GANN to produce realistic simulations of the response of a detector. The setup contains a traditional simulator that will be used to produce the training dataset and also the skeleton of a GANN that has to be trained to produce events similar to the simulator.

Low-Gain Avalanche Diodes (LGADs) are silicon-based sensors with very good timing capabilities. Both CMS and ATLAS are considering layers of LGADs as part of their timing detectors for the HL-LHC. In CMS, the endcaps of the timing detector, known as the **Endcap Timing Layer** (ETL) will be instrumented with LGADs. The nominal time resolutions of these sensors are in the order of 35 ps. These values have been already confirmed in test beams. More details about the sensors can be found in [4].

In CMS, the LGAD sensors are read-out with an ASIC called ETROCv2. Both the sensor and the ASIC produce a characteristic signal shape when a charged particle crosses the active volume of the LGAD. The ETROCv2 ASIC imposes a detection threshold and extracts the two relevant time observables: the **Time Of Arrival** (TOA) and the **Time Over Threshold** (TOT).

CMS simulates the response of these sensors using a semi-empirical, **surrogate** model. The momentum, direction, and time of the incoming particle is obtained from GEANT4 together with the deposited energy. The model uses a pre-defined template of the signal shape and scales it with the total amount of deposited energy. The TOA and the TOT are estimated for a given threshold and they are further smeared to emulate different effects: the jitter (the effect of the electronic noise in the signal shape), the skewness of the TDC, or the effect of the inhomogeneous ionization. This simulation process has a clear weakness: it relies on a characteristic signal template that may be different for different sensors or ASICs. In this exercise, a surrogate model based on a GANN will be device to produce realistic simulations with a training based on data.

The notebooks in google-colab provide a basic simulator of the LGAD-ETROC response. It implements the same semi-empirical model in CMSSW with the exception that the energy deposition is directly extracted from a Landau distribution and the signal shape uses an analytical form: $A(t) = A_0 e^{\frac{-t}{\tau_1}}(1 - e^{\frac{-t}{\tau_2}})$. Parameters such as the threshold, the signal-to-noise ratio, and others can be configured to produce slightly different simulations. This setup will be used to produce a dataset from which the GANN will learn.

The notebooks contain the skeleton of a GANN including the definition of the Generator, the Discriminator, the loss functions, and the algorithm to perform the training. Everything is implemented using PyTorch, one of the most popular ML packages.

14.4.2 A GNN to perform track reconstruction

The second exercise aims at performing track reconstruction in a simulated tracker that will produce the hits associated with the generated particles. The GNN will find which of the hits belong to a same track. The notebooks contain a generic tracker simulator that emulates most of the basic components of the a tracker software. This includes the geometry, the propagation of the tracks, the navigation system and also some plotting tools. The geometry is highly configurable and relies on the concept of **detection plane** and **module** (a detection plane with boundaries). Propagation is performed assuming a constant magnetic field but includes also the effect of multiple scattering and single detector resolution. The navigation system guides the propagators to the most probable detectors in order to increase efficiency. It should be noted that *python* is not the best language to perform a real simulation of a tracker because of speed issues, however this framework provides a pedagogic environment in which to understand the basic principles of a tracking system. In this exercise, this simulator will be used to create the hits that will be reconstructed.

The notebooks contain already an skeleton of a GNN aimed at performing the association of the hits with the corresponding tracks. The code implements a **SAGE Graph Convolutional Neural Network** [3] using *PyTorch Geometric*, the PyTorch library specialized in GNNs. The graph assumes that edges can only exist between hits in contiguous layers. In addition, edges won't be permitted between hits that differ in the transverse angle more than a quantity $\Delta\phi$. These two conditions are

given to simplify the graph although they are reasonable for relatively straight tracks. In the notebook you will find how the input to the GNN is actually implemented. The features of every hit (node) are their spatial position in the detector. The last piece in the setup is a code prepared to perform the reconstruction of a non-connected graph and also to measure metrics such as the efficiency.

14.5 Procedure

To follow the exercise you should make a copy of the google colab notebooks and directories that will be provided during the session. It is important that you work in your own copy of the code so you do not interfere with others' developments.

14.5.1 A GANN to simulate the response of an LGAD sensor

1. Production of LGAD response datasets.

   (a) Go to the LGAD simulation notebook. For this part you do not need to choose GPU support in google colab.

   (b) Explore the code including also the imports that contain the empirical model. Make sure you understand all the aspects.

   (c) Write a snippet of code that draws the signal for two different sets of parameters.

   (d) Produce two datasets of 500K events with the two different sets of parameters.

2. Training of the GANN.

   (a) Go to the GANN training notebook. For this part you NEED to choose GPU support in google colab.

   (b) Explore the code including also the imports. Make sure you understand all the aspects.

   (c) Finish the training procedure trying to understand all the steps.

   (d) Run the training with the proposed hyper-parameters.

   (e) Plot the evolution of the G and D loss functions against the epoch number.

3. Synthetic dataset production and comparison.

   (a) Go to the GANN production notebook. For this part you NEED to choose GPU support in google colab.

   (b) Explore the code including also the imports. Make sure you understand all the aspects.

   (c) Produce a dataset of synthetic data with the GANN.

   (d) Use the plotting tools to compare the synthetic and the original dataset.

14.5.2 A GNN as a hit-track associator

1. Simulation of an event in the tracker simulator.
   (a) Go to the tracker simulator notebook. For this part you do not need to choose GPU support in google colab.
   (b) Explore the code including also the imports that contain the tracker simulator. Make sure you understand all the aspects.
   (c) Generate an event with a certain number of tracks and visualize it.
   (d) Generate and save 2 events with 50, 100 and 500 tracks.
2. Training of the GNN.
   (a) Go to the GNN training notebook. For this part you need to choose GPU support in google colab.
   (b) Explore the code including the SAGE model. Make sure you understand all the aspects.
   (c) Train the GNN using one of the 50, 100 and 500 tracks events.
3. Track reconstruction on a bunch of hits.
   (a) Go to the GNN application notebook. For this part you need to choose GPU support in google colab.
   (b) Apply the model to the second event with 50, 100, and 500 tracks.
   (c) Measure the track reconstruction efficiency.

14.6 Analysis and interpretation of the results

All through the notebooks you will be asked to produce a set of plots and to answer specific questions. The most important central questions are whether the GANN managed to learn from the empirical model and how well the GNN was able to reconstruct the tracks.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, *Generative Adversarial Networks*, https://doi.org/10.48550/arXiv.1406.2661.

[2] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, Kilian Q. Weinberger, *Simplifying Graph Convolutional Networks*, https://doi.org/10.48550/arXiv.1902.07153.

[3] William L. Hamilton, Rex Ying, Jure Leskovec, *Inductive Representation Learning on Large Graphs*, https://doi.org/10.48550/arXiv.1706.02216.

[4] G. Pellegrini, et al., *Technology developments and first measurements of Low Gain Avalanche Detectors (LGAD) for high energy physics applications*, Nucl. Inst. Meth. A, Volume 765, 2014, 12-16, https://doi.org/10.1016/j.nima.2014.06.008.

Exercise 15 Signal Integrity Measurements: TDR Analysis and Impedance Matching in Transmission Lines

15.1 Objectives

This laboratory is an introductory session to signal integrity foundation techniques and transmission lines. It addresses the experimental characterization of transmission lines using Time Domain Reflectometry (TDR) and the evaluation of impedance matching techniques. These methodologies are essential in high-speed digital systems, RF engineering, and power electronics, where signal integrity constraints directly impact system performance. The main objective of this activity is to guide the student through experimental techniques for TDR that help in evaluating the impact of impedance matching and signal integrity in high-speed interconnects. At the end of the exercise, students should be able to:

- Understand the concept of transmission line behavior and its dependence on distributed parameters.
- Interpret TDR waveforms and relate them to impedance discontinuities.
- Determine the characteristic impedance (Z_0) and load impedance (Z_L) from time-domain measurements.
- Identify and localize reflections and discontinuities in transmission lines.
- Evaluate the effect of impedance mismatch on signal integrity (ringing, overshoot, distortion).
- Implement and validate basic impedance matching techniques.
- Establish the link between time-domain (TDR) and frequency-domain (S-parameter) analysis.

15.2 Professors

- Fernando Arteche
- Francisco Javier Galindo
- Álvaro Pradas
- Mateo Iglesias

15.3 Theoretical Introduction

15.3.1 Transmission Lines and Signal Propagation

A transmission line is a distributed electrical structure designed to guide electromagnetic waves or signals from one point to another with controlled impedance and

minimal distortion. When the electrical length of an interconnect (cable, pcb, transmission line, fiber...) becomes comparable to the signal rise time or the wavelength of the highest significant frequency component, it must be treated as a transmission line rather than a lumped element. The behavior of the line is governed by four distributed parameters per unit length:

- Series Resistance R (Ω/m):Accounting for conductor and skin-effect losses
- Series Inductance L (H/m):Representing magnetic energy stored in the surrounding field
- Shunt Capacitance C (F/m):Modeling electric energy stored between conductors
- Shunt Conductance G (S/m):Modeling dielectric leakage losses

Figure 1 shows the canonical lumped-element model for a differential section Δx of a two-wire transmission line

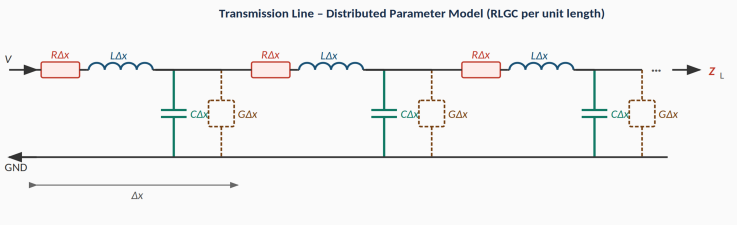


Figure 52: Distributed RLGC model for an infinitesimal section Δx of a transmission line.

15.3.2 Characteristic Impedance

The characteristic impedance Z_0 is the ratio of voltage to current for a wave travelling along an infinite, matched line. It represents the impedance seen by a wave propagating along the line in the absence of reflections. In the general (lossy) case:

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

For a low-loss line (R « ωL and G « ωC), the common engineering approximation at high frequencies is:

$$Z_0 \approx \sqrt{\frac{L}{C}}$$

Typical values: 50 Ω (coaxial RF), 75 Ω (broadcast coaxial), 100 Ω (twisted-pair differential), 90–130 Ω (PCB microstrip/stripline depending on geometry).

15.3.3 Propagation Velocity and Delay

In transmission line and electromagnetic theory, μ is the **magnetic permeability** of the medium. It describes how the material supports magnetic fields, with units of henry per meter (H/m): $\mu = \mu_o \mu_r$. ε is the **electric permittivity** of the medium.

It describes how the material supports electric fields, with units of farad per meter (F/m): $\varepsilon = \varepsilon_o \varepsilon_r$. They determine the wave propagation velocity:

$$v_p = \frac{c}{\sqrt{\varepsilon_r \mu_r}}$$

With μ_0 and ε_0 for free space as $\mu_0 = 4\pi \times 10^{-7}$ H/m and $\varepsilon_0 \approx 8.854 \times 10^{-12}$ F/m

Now, let's consider a TEM wave in a transmission line. The propagation velocity (often denoted v_p) refers to the speed at which the electromagnetic wave travels along the line. In this ideal TEM wave, the phase velocity, this is, the speed of a constant phase point (e.g., a sinusoidal peak) matches the propagation velocity. In dispersive or non-TEM cases, phase velocity can differ from other velocities (e.g., group velocity), but for ideal TEM transmission lines, they are identical. The terms are effectively equivalent under standard conditions, and the phase velocity of a TEM wave on the line is:

$$v_p = \frac{\omega}{\beta} = \frac{1}{\sqrt{LC}} = \frac{c}{\sqrt{\varepsilon_r \mu_r}}$$

where γ is the propagation constant, α (Np/m) is the attenuation constant, and β the phase constant (rad/m). They relate as:

$$\gamma = \alpha + j\beta \qquad \text{with} \qquad \beta = \frac{2\pi}{\lambda}$$

λ is the wavelength of the electromagnetic wave, i.e., the physical distance over which the wave completes one full cycle, and c is the speed of light in vacuum. ε_r is the relative permittivity of the insulating medium ($\mu_r \approx 1$ for non-magnetic materials).Common velocity factors are: Solid PTFE: $\approx 0.70 \cdot c$, Foam PE: $\approx 0.82 \cdot c$, FR-4 PCB: ≈ 0.54–$0.59 \cdot c$. The one-way propagation delay of a line of physical length d is

$$T_D = \frac{d}{v_p}$$

Thus, it quantifies the temporal latency introduced by the line.

15.3.4 Reflection Coefficient and Impedance Mismatch

When a travelling wave in a transmission line reaches a load Z_L that differs from Z_0, a partial reflection occurs. The voltage reflection coefficient Γ is defined as the ratio of reflected to incident voltage and can be expressed in terms of the load and line impedance as:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0}$$

Conversely, the load impedance can be recovered from a measured Γ:

$$Z_L = Z_0 \cdot \frac{1 + \Gamma}{1 - \Gamma}$$

There are three special cases:

- $\Gamma = +1$: open circuit, $Z_L = \infty$
- $\Gamma = -1$: short circuit, $Z_L = 0$

- $\Gamma = 0$: matched load, $Z_L = Z_0$

Figure 2 depicts Γ as a function of the normalized load impedance Z_L/Z_0, it tends in the limits to a short in the origin, and an open for very large impedances. We can also depict these singular points in the Smith Chart. The Smith chart is a graphical tool used in RF and microwave engineering to represent complex impedance, admittance, and reflection coefficients of transmission lines. It maps normalized complex impedances onto a unit circle in the complex Γ-plane. The center of the chart corresponds to $Z_L = Z_0$ - $\Gamma = 0$, this is matched condition and no reflection. The outer circle corresponds to $\Gamma = 1$, total reflection (purely reactive, open, or short loads). The rightmost point is $\Gamma = 1$, open circuit. The leftmost point is $\Gamma = -1$, short circuit. The chart contains , constant resistance circles and constant reactance arcs which allow graphical determination of impedance matching, reflection coefficient, VSWR, return loss, transmission line transformations, stub tuning. It provides an intuitive visualization of how impedances transform along a transmission line as a function of electrical length

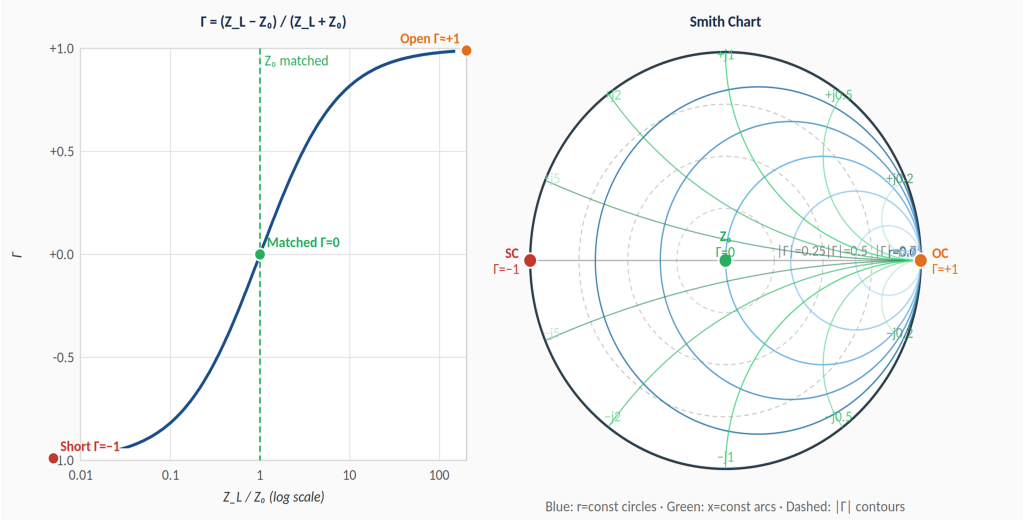


Figure 53: Left: reflection coefficient Γ vs. normalized load Z_L/Z_0. Right: Smith Chart with constant resistance (blue) and reactance (green) circles and $|\Gamma|$contours (dashed).

The Return Loss (RL) quantifies how much power is reflected back due to impedance mismatch in a transmission line or RF system. It is defined from the reflection coefficient Γ:

$$RL = -20 \log_{10} |\Gamma| [dB]$$

RL indicates the quality of the impedance match being large RL $\rightarrow$ small reflection $\rightarrow$ good matching and small RL $\rightarrow$ large reflection $\rightarrow$ poor matching. Return loss is widely used in RF, microwave, antennas, and high-speed interconnect characterization.

Other common term is **Voltage Standing Wave Ratio (VSWR)** that measures the degree of impedance mismatch on a transmission line by comparing the maximum and minimum voltages of the standing wave pattern and can be related to the reflection coefficient as

$$VSWR = \frac{1 + |\Gamma|}{1 - |\Gamma|}$$

Some characteristic values are ($VSWR = 1$) that corresponds to a perfect match ($\Gamma = 0$) and $VSWR \to \infty$ representing a total reflection ($|\Gamma| = 1$).

15.3.5 Time Domain Reflectometry (TDR)

TDR (Time-Domain Reflectometry) is a measurement technique used to characterize transmission lines and detect impedance discontinuities by observing reflected signals in the time domain. TDR injects a fast voltage step at one end of the line and records the voltage as a function of time at the same port. Reflections from impedance discontinuities return after a round-trip delay, making it possible to identify the nature of the discontinuity (from the reflection polarity) and localize it (from the time delay).

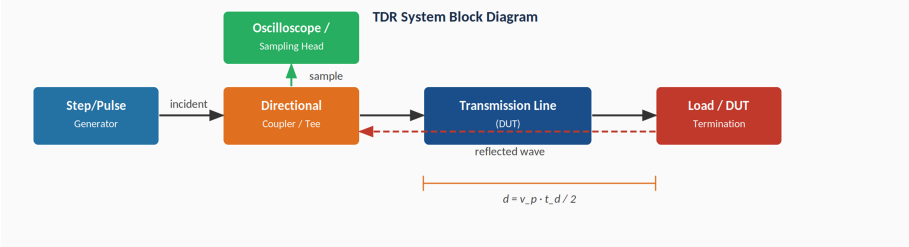


Figure 54: TDR system block diagram. The directional coupler separates incident and reflected waves; the oscilloscope records the combined waveform.

Localization of Discontinuities The physical distance d to a discontinuity is obtained from the round-trip delay time td of the reflected edge:

$$d = \frac{v_p t_d}{2}$$

The spatial resolution Δx is the minimum separation between two resolvable discontinuities and is determined by the 10%–90% rise time t_r of the step stimulus:

$$\Delta x \approx \frac{v_p t_r}{2}$$

Example: for a cable with vp $= 0.66 \cdot$ c and a TDR with $t_r = 100ps$, the spatial resolution is $\Delta x \approx 10mm$.

Graphical Interpretation In a TDR measurement, the observed waveform is the superposition of the incident step and all reflections generated by impedance discontinuities along the transmission line. A perfectly matched line terminated with $Z_L = Z_0$, produces a flat waveform after the initial rise because no energy is reflected back. If the signal encounters an impedance higher than the characteristic impedance, the reflected voltage is positive, creating an upward step or bump in the waveform; this is typical of open circuits, widened traces, inductive vias, or connector discontinuities. In the extreme case of an open circuit, the reflection coefficient becomes $\Gamma = +1$, and the reflected wave doubles the measured voltage at the end of the line. Conversely, when the impedance decreases below Z_0, the reflection becomes negative, producing a downward dip or step. This occurs with capacitive discontinuities, narrowed traces, excessive loading, or short circuits. For a perfect short circuit, $\Gamma = -1$, the reflected voltage cancels the incident step. It

is possible to depict all this information in voltage graphs along a given point in the line. The horizontal axis of the TDR waveform corresponds to propagation time from the measurement point, allowing the physical location of each discontinuity to be estimated from the propagation velocity of the line. In practical systems, waveform transitions are not perfectly sharp because conductor losses, dielectric losses, dispersion, and limited instrument bandwidth smooth and attenuate the reflections. Consequently, TDR waveforms provide both qualitative and quantitative information about impedance uniformity, fault location, connector quality, PCB interconnect behavior, and overall signal integrity. The measured voltage at the TDR port is the superposition of the incident step and all subsequent reflected waves. Figure 4 illustrates the characteristic signatures for four canonical terminations:

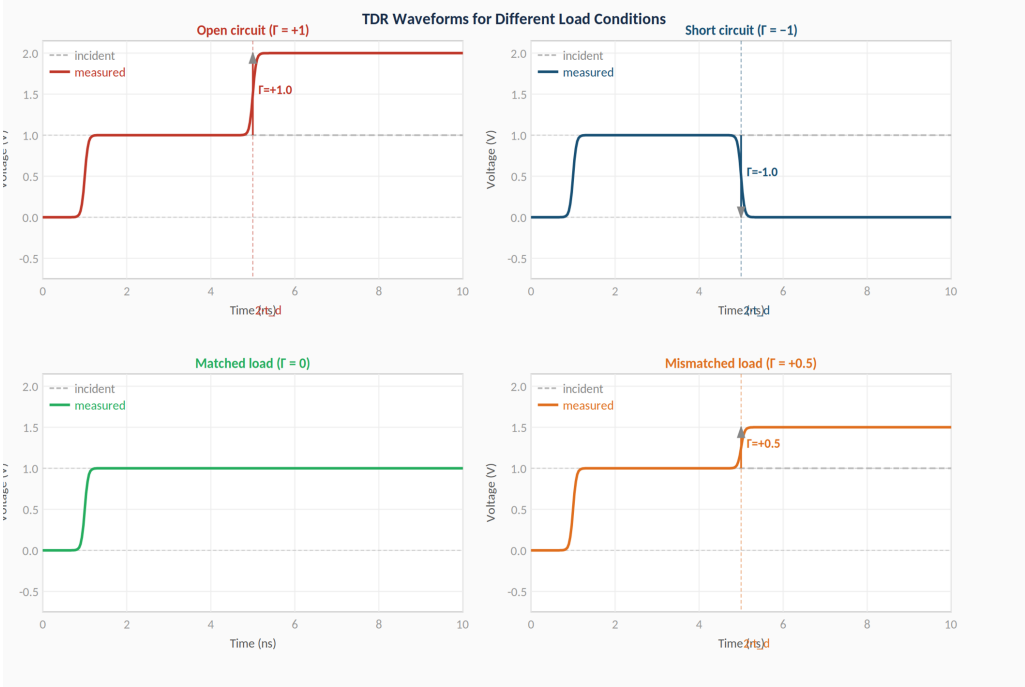


Figure 55: – TDR waveforms for open circuit ($\Gamma = +1$), short circuit ($\Gamma = -11$), matched load ($\Gamma = 0$) and mismatched load ($\Gamma = +0.5$). Dashed = incident step; solid = measured waveform..

Reading rule: a step going UP after the reflection epoch indicates $Z_L > Z_0$ and a step going down indicates $Z_L < Z_0$. A smooth monotonic rise with no subsequent disturbance indicates a matched condition. The reflections accumulate with time until the line reaches an steady state

Jitter due to Reflections Reflections cause the transition instant to shift in time depending on the data pattern history (data-dependent jitter, DDJ). Each time a reflected wave arrives at the receiver, it adds to or subtracts from the signal just before a new edge, displacing the threshold-crossing time. Figure 5 shows overlaid rising edges for matched and unmatched configurations.

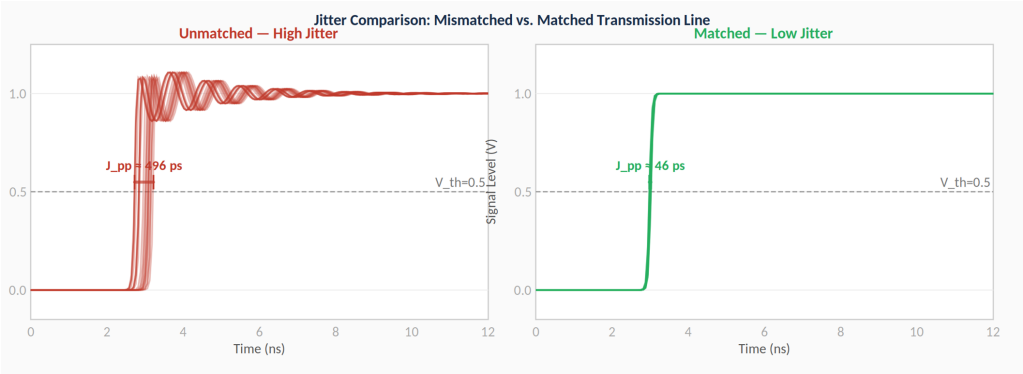


Figure 56: Jitter comparison. Mismatched line (left): multiple reflections shift the threshold crossing randomly, producing high peak-to-peak jitter J_pp. Matched line (right): stable edge timing, J_pp reduced by an order of magnitude.

Effect on the Eye Diagram The eye diagram is constructed by folding the signal waveform into 2-UI windows and overlaying all traces. In a mismatched system, reflections close the eye both vertically (reduced noise margin) and horizontally (increased jitter). Figure 6 illustrates the dramatic difference between matched and unmatched conditions.

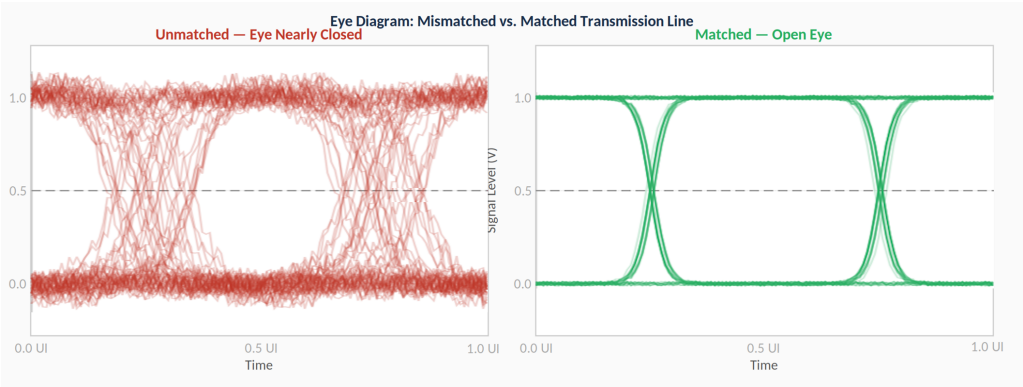


Figure 57: Eye diagram comparison. Mismatched line (left): eye nearly closed due to inter-symbol interference and jitter from reflections. Matched line (right): wide open eye with clear noise margins.

15.4 Experimental Setup- Instrumentation

The following instrumentation and materials are required for the laboratory exercise:

- Step/pulse generator with low rise time $t_r \leq 200\,\text{ps}$, high-bandwidth sampling oscilloscope, Vector Network Analyzer calibrated to the port reference plane, and SOLT calibration kit matching the connector type.
- RG-58 coaxial cable ($50\,\Omega$, $v_p \approx 0.66 \cdot c$) in various lengths, twisted-pair cable ($100\,\Omega$ differential) such as Cat 5e or equivalent, and PCB controlled-impedance traces (microstrip and/or stripline).
- Open-circuit and short-circuit terminations, precision $50\,\Omega$ and $75\,\Omega$ terminations, resistive loads ($25\,\Omega$, $75\,\Omega$, $100\,\Omega$), and a variable decade resistor box.

15.5 Procedure

Step 1 – Initial Setup and Signal Verification

The TDR measurement is performed with a T-type power splitter (or coaxial T-connector) inserted between the step generator and the Device Under Test (DUT). One port of the T-junction connects to the pulse generator (50Ω output), the second port connects to the DUT, and the third port is used to monitor the signal via the oscilloscope (high-impedance input). This arrangement is equivalent to the block diagram in Figure 3.

IMPORTANT: The oscilloscope must be set to high-impedance termination (1MΩ) and connected either directly to the T-splitter or via an electrically short cable. This minimizes parasitic effects and ensures the measurement system does not significantly load the circuit. Verify the generator baseline: connect the 50Ω precision load directly at the T-splitter output port (DUT). The oscilloscope should show a clean step with no reflections and a flat top. Measure the rise time t_r of the incident step. Record V_0 (the steady-state amplitude of the incident step).

Step 2 – Open and Short Characterization

- Connect an open-circuit termination. Record the reflected waveform.
- Connect a short-circuit termination. Record the reflected waveform
- Compare the polarity, reflection amplitude and shape. Expected: $\Gamma_{open} = +1$, $\Gamma_{short} = -1$.
- Identify all components and check connector types. Connect one of T splitter ports

Step 3 – Propagation Velocity Measurement

- Connect a cable as DUT and measure the round-trip delay t_d, defined as the time between the incident and reflected edge measured at 50 % amplitude. Calculate the propagation velocity as: $v_p = \frac{2d}{t_d}$
- Derive the velocity factor $VF = \frac{v_p}{c}$ and compare it with the cable datasheet. Calculate the relative permittivity of the dielectric as:$\varepsilon_r = \frac{1}{VF^2}$

$$d = \frac{v_p \cdot t_d}{2}$$

Step 4 – Impedance Mismatch Analysis

- Connect known loads (25 Ω, 75 Ω, 100 Ω) one at a time. For each load, measure the reflected voltage amplitude V_{ref}.
- Calculate the reflection coefficient $\Gamma = \frac{V_{ref}}{V_{incident}}$ and deduce the load impedance $Z_L = Z_0 \cdot \frac{1+\Gamma}{1-\Gamma}$ .Compare the calculated Z_L with the nominal resistor value.

Step 5 – Discontinuity Localization

- Introduce a controlled discontinuity by connecting two cables in series via a T-junction. Identify all reflection events and note their time coordinates t_1,t_2,..
- Calculate the physical position of each discontinuity: $d_i = v_p.t_i/2$. Verify by physically measuring the cable lengths.

Step 6 – Impedance Matching

- Configure a 50Ω line driving a 100Ω load without matching. Record TDR waveform
- Add a series resistor $R_S = 50\Omega$ at the source end. Record waveform.
- Replace with a 100Ω parallel resistor at the load. Record waveform.
- Compare all three waveforms; evaluate reduction in reflection amplitude and settling time

15.6 Analysis and Interpretation of Results

15.6.1 Reflection Analysis and Impedance calculations

For each load in Step 4, fill the following table. V_0 is the amplitude of the incident step; V_{TDR} is the steady-state measured voltage after the reflection epoch.

$$\Gamma = \frac{V_{TDR} - V_0}{V_0}$$

Estimate the measurement uncertainty in Z_L from the oscilloscope amplitude resolution and noise floor. Discuss systematic errors (non-ideal T-splitter, cable losses, connector reflections).

$$Z_L = Z_0 \cdot \frac{1 + \Gamma}{1 - \Gamma}$$

15.6.2 Discontinuity Localization

Tabulate the expected and measured positions of each discontinuity from Step 5. Calculate the absolute and percentage position error for each event. Discuss the amplitude reduction of the second reflection: if the T-junction presents a reflection coefficient Γ_j at the junction, the transmitted wave amplitude is scaled by $(1 - \Gamma_j^2)$. Estimate Γ_j from the waveform amplitudes.

15.6.3 Propagation Velocity and Dielectric Permittivity

From Step 3, tabulate: t_d [ps], v_p [m/s], VF, ε_r. Compute the percentage error between your measured ε_r and the datasheet value. A residual error below 5 % is acceptable for standard RG-58 at room temperature.

Discuss the effect of the finite rise time of the generator on the accuracy of t_d measurement (the 50 % crossing method minimises this error for symmetric rising edges, but can introduce bias for asymmetric edges due to probe attenuation at high frequency).

15.6.4 Effect of Impedance Mismatch

For each configuration in Step 6, measure:

- Overshoot:$O = \frac{V_{\max} - V_{\text{final}}}{V_{\text{final}}} \times 100\,[\%]$
- Settling time:t_s = time for the waveform to remain within $\pm\,5\,\%$ of V_{final}
- Ringing frequency:f_{ring} measured as $frac$1period of oscillation Compare with $\frac{1}{2T_D}$

Express the improvement in Return Loss achieved by each matching technique. Discuss the trade-off: series matching preserves signal amplitude at the far end at steady state but introduces a half-amplitude step during the first traversal; parallel matching achieves immediate clean termination but draws continuous current.

15.7 Quick references

Quantity	Formula	Notes
Characteristic impedance (lossless)	$Z_0 = \sqrt{\dfrac{L}{C}}$	L, C per-unit-length params [H/m, F/m]
Characteristic impedance (lossy)	$Z_0 = \sqrt{\dfrac{R + j\omega L}{G + j\omega C}}$	General dispersive line
Propagation velocity	$v_p = \dfrac{1}{\sqrt{LC}} = \dfrac{c}{\sqrt{\varepsilon_r \cdot \mu_r}}$	$c = 3 \times 10^8$ m/s
One-way delay	$T_D = \dfrac{d}{v_p}$	d = physical length
Reflection coefficient	$\Gamma = \dfrac{Z_L - Z_0}{Z_L + Z_0}$	$\Gamma \in [-1, +1]$
Load from Γ	$Z_L = Z_0 \cdot \dfrac{1 + \Gamma}{1 - \Gamma}$	Requires $\Gamma \neq 1$
Distance to fault	$d = \dfrac{v_p \cdot t_d}{2}$	t_d = round-trip time
Spatial resolution	$\Delta x \approx \dfrac{v_p \cdot t_r}{2}$	t_r = 10–90 % rise time
Return loss	$RL = -20 \cdot \log_{10} \lvert\Gamma\rvert$ [dB]	Large RL ⇒ good match
VSWR	$VSWR = \dfrac{1 + \lvert\Gamma\rvert}{1 - \lvert\Gamma\rvert}$	$VSWR = 1$ ⇒ perfect match
Velocity factor	$VF = \dfrac{v_p}{c} = \dfrac{1}{\sqrt{\varepsilon_r}} \quad (\mu_r \approx 1)$	Dimensionless
Permittivity from VF	$\varepsilon_r = \dfrac{1}{VF^2}$	Relative permittivity of dielectric

Exercise 16 Design of an Optical Active Target

16.1 Objectives

General objective: Introduce students to the design and operation of an optical active target using a Multilayer Thick Gas Electron Multiplier (MThGEM) as a light amplification structure, combining primary and secondary scintillation readout via a photomultiplier tube and secondary electroluminescence imaging via a digital camera (CCD, CMOS or Timepix-based device). The detector operates with a low-pressure gas mixture (namely, $ArCF_4$), in which the gas acts simultaneously as the target material, the ionisation medium, and the scintillator.

Specific objectives:

- Understand the operating principle of an optical active target and its potential over conventional passive-target detector systems.
- Understand the physical origin of primary (S1) and secondary (S2) scintillation in a low-pressure gas mixture and identify the role of each signal in the two-signal readout scheme.
- Identify the function of each detector layer in the vertical stack: photomultiplier tube, cathode, drift region, MThGEM, charge readout layer, optical viewport, and digital camera.
- Understand how the electric field configuration in the drift region and across the MThGEM controls electron transport, light production, and spatial resolution.
- Acquire and interpret PMT waveforms containing S1 and S2 pulses, and extract from them the gain, the electron drift velocity, and ionisation information.
- Use camera images of the electroluminescence produced at the MThGEM to reconstruct the projected topology of particle tracks generated by a radioactive source.
- Discuss the main design trade-offs of an optical active target: gas pressure, electric field strength, light gain, longitudinal and transverse diffusion, and spatial resolution.

16.2 Professors

- Yassid Ayyad
- Cristina Cabo

16.3 Theoretical Introduction

16.3.1 The active target concept

In a conventional nuclear or particle physics experiment, the target and the detector are separate objects. The target material — a foil, a gas cell, or a liquid — is placed

upstream of a detector system that measures the reaction products. This configuration suffers from an intrinsic limitation: low-energy recoils produced in the reaction may stop inside the target itself before reaching the detector, making them invisible to the measurement. The minimum detectable recoil energy is therefore limited by the energy loss in the target thickness needed to provide sufficient luminosity.

An active target overcomes this limitation by using the detector medium itself as the target material. Every reaction that occurs anywhere in the sensitive volume produces recoils that immediately begin to ionise and excite the surrounding medium, generating a detectable signal. There is no dead layer between the interaction point and the detector. This makes active targets particularly powerful for experiments with low-energy recoils, inverse-kinematics reactions, or extended vertex topologies.

In the optical active target studied in this exercise, the sensitive medium is a low-pressure gas. The gas simultaneously provides the target nuclei for interactions with the radioactive source, generates ionisation and scintillation light along the particle tracks, and transports the ionisation electrons toward the amplification structure under the action of an applied electric field.

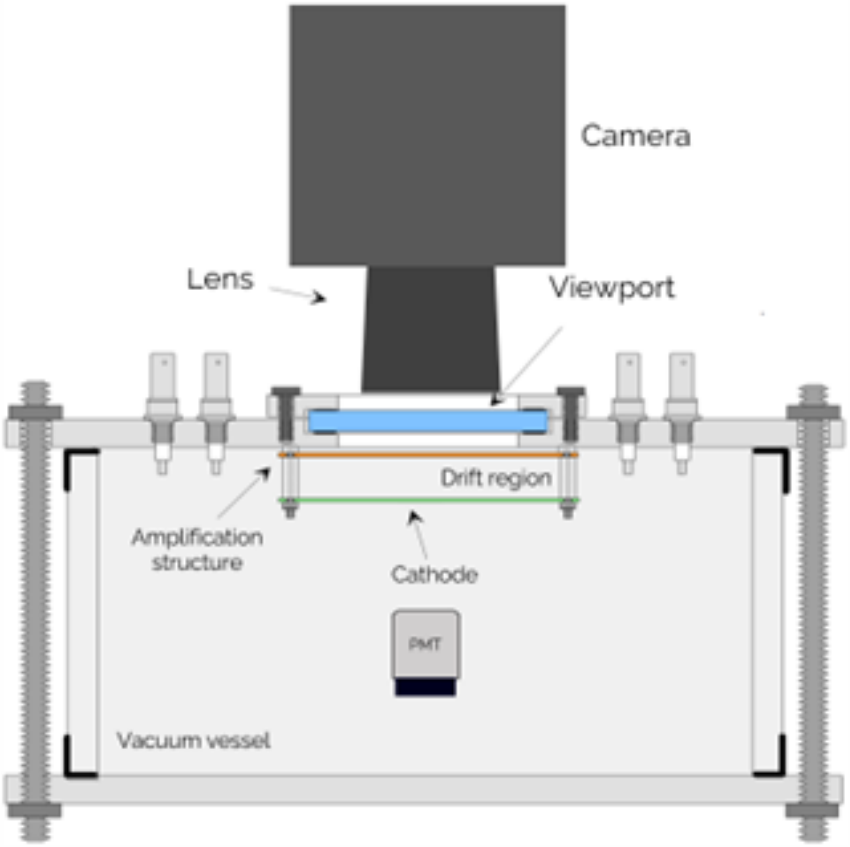


Figure 58: Scheme of the proposed setup.

Optical active targets based on gas TPCs have been developed and used in several nuclear and particle physics contexts. Notable examples include the Active Target Time Projection Chamber (AT-TPC) developed at the National Superconducting Cyclotron Laboratory (NSCL/FRIB) at Michigan State University, used for studies of nuclear reactions in inverse kinematics with rare isotope beams, and various optical TPC designs proposed and operated for rare-event searches such as directional dark matter detection. These detectors share the common feature of combining gas as both

target and sensitive medium with an optical or electronic readout of the ionisation and scintillation produced by the reaction products.

16.3.2 Scintillation in different gas mixtures

When a charged particle traverses a gas, it loses energy through ionisation and excitation of the gas molecules along its path. A fraction of the deposited energy goes into the excitation of atomic and molecular states that subsequently decay radiatively, emitting scintillation photons. This prompt light signal, produced directly by the primary particle track, is referred to as the primary scintillation signal, or S1.

Some gases produce scintillation in the vacuum ultraviolet (VUV) or UV region, which is not directly detectable by standard photosensors. The addition of molecular admixtures can shift a fraction of the emission to the visible range. In the case of argon, the addition of CF_4 molecules shifts the emission to the visible region, allowing its detection with conventional light detectors. A mixture of $ArCF_4$ 90/10 at low pressure provides a scintillation spectrum accessible to standard borosilicate-window photomultiplier tubes and CMOS sensors without the need for wavelength-shifting coatings.

Table 4 summarises the key scintillation properties of gases commonly considered for this type of detector.

Table 4: Key scintillation properties of common gas mixtures for optical active targets.

Gas / Mixture	**Emission region & wavelength**	**Range**	**PMT/CMOS?**	**WLS?**
Pure Ar	VUV	∼128 nm	No	Yes
Pure CF_4	Visible / UV	200–700 nm	Yes (partial)	No
$ArCF_4$ 90/10	Visible	300–650 nm	Yes	No
Pure He	VUV	∼60–80 nm	No	Yes
He + isobutane	Visible / UV	300–450 nm	Partial	Depends

I would recommend Option B as the cleaner solution — tabularx stretches the second column to fill the text width automatically, keeping everything within margins without rotating text. Which would you prefer? Figure 59 shows an example of light emission wavelength ranges for different gas mixtures together with the transmission of a standard borosilicate viewport and the detection range of silicon-based cameras. The use of wavelength shifters (WLS) can partially overcome the mismatch between VUV emission and standard photosensor sensitivity, although this approach carries additional drawbacks that must be taken into account depending on the specific experiment.

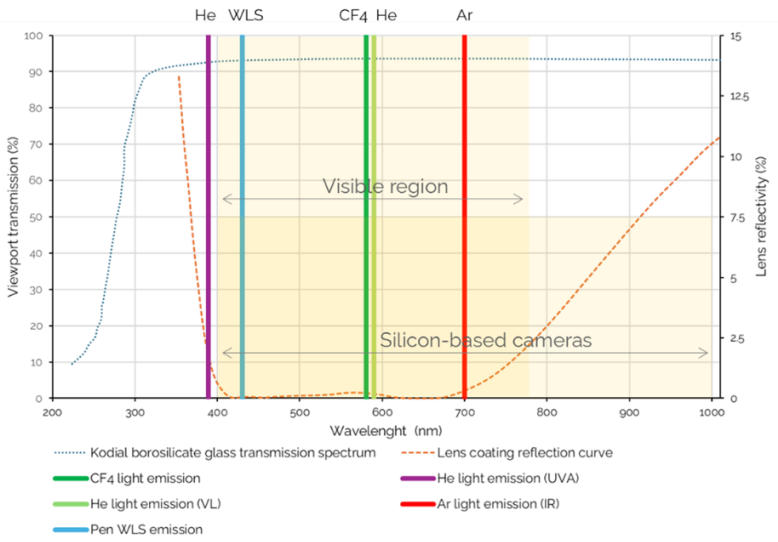


Figure 59: Example of different gas light emission wavelength ranges, borosilicate viewport transmission, and silicon-based camera detection range.

Once the scintillation is produced, the S1 signal is prompt, with a rise time determined by the excitation and de-excitation kinetics of the gas mixture, and its intensity is proportional to the energy deposited by the primary particle along the track segment seen by the photosensor (Figure 60). In the geometry of this exercise, the S1 light is collected by the PMT located below the cathode, looking upward into the active volume through the cathode mesh (see Figure 58).

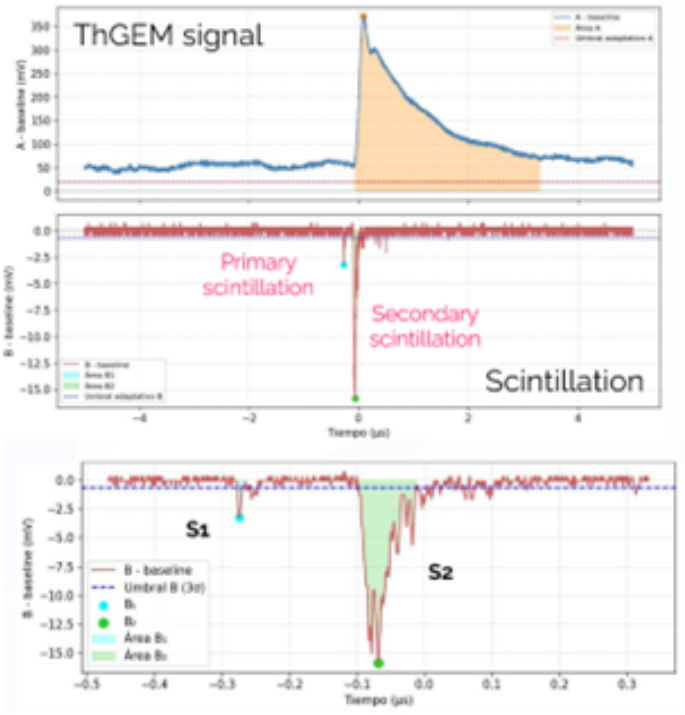


Figure 60: Example of a MThGEM/PMT waveform showing the prompt primary scintillation signal (S1) and the delayed secondary electroluminescence signal (S2).

16.3.3 Electron drift and longitudinal coordinate reconstruction

The ionisation electrons produced along the primary track are separated from the positive ions by the drift electric field applied between the cathode (at negative high voltage) and the top electrode of the MThGEM (at ground potential or at a less negative potential). Under this field, the electrons drift downward toward the MThGEM at a velocity v_d that depends on the gas mixture, pressure, and reduced electric field E/p.

The reduced electric field E/p is the ratio of the electric field E (in $\mathrm{V\,cm^{-1}}$) to the gas pressure p (in bar or Torr). It is used instead of E alone because the microscopic physics of electron transport — how frequently electrons collide with gas molecules and how much energy they gain between collisions — depends not on E or p separately, but on their ratio. At a given E/p, doubling the pressure doubles the collision frequency but also halves the mean free path, leaving the electron energy distribution unchanged. Using E/p therefore allows detector behaviour to be compared across different operating pressures in a physically meaningful way.

The time elapsed between the detection of the S1 signal and the arrival of the electrons at the MThGEM, where they produce the secondary signal S2, is the electron drift time Δt. Since the drift velocity is approximately uniform across the active volume for a uniform field, the longitudinal coordinate z of the interaction point along the drift direction can be reconstructed as:

$$z = v_d \cdot \Delta t \tag{19}$$

This is the fundamental principle of a time projection chamber (TPC). The S1 signal provides the time reference t_0, and the S2 signal provides the arrival time of the electrons. The combination of the two signals therefore gives access to the full three-dimensional information of the event: the projected track image in the transverse plane is obtained from the camera, while the longitudinal coordinate is obtained from the drift time.

During drift, the electron cloud spreads due to diffusion. The transverse diffusion coefficient D_T determines the broadening of the cloud in the plane perpendicular to the drift direction, while the longitudinal diffusion coefficient D_L broadens the cloud along the drift direction. Both coefficients depend on the gas mixture, pressure, and electric field, and set a fundamental limit on the spatial resolution of the detector independently of the readout granularity.

A further effect that must be considered is electron attachment: electronegative molecules present in the gas (including trace impurities such as O_2 or H_2O, as well as CF_4 itself at certain field conditions) can capture drifting electrons, removing them from the signal. Attachment reduces the number of electrons that reach the MThGEM and therefore attenuates the S2 signal. High gas purity and careful vessel preparation are therefore important for detector performance.

In practice, D_T is not a monotonic function of E/p. For many gas mixtures, D_T presents a minimum at a particular value of E/p, and operating the drift field at or near this minimum is therefore the preferred strategy to preserve the sharpness of the track image recorded by the camera. However, this choice involves a compromise: the field that minimises D_T does not in general coincide with the field that maximises the electron drift velocity or optimises the electron extraction efficiency into the MThGEM holes. The selection of the drift field is therefore a

multi-parameter optimisation, balancing track image resolution, signal amplitude, and electron transport efficiency simultaneously.

16.3.4 Micropattern gas detectors: from GEM to MThGEM

Before describing the MThGEM in detail, it is useful to introduce the broader family of devices to which it belongs. Micropattern gas detectors (MPGDs) are a class of gas-based particle detectors in which the charge amplification and signal readout are performed by microscopic electrode structures, typically fabricated using printed circuit board (PCB) technology. The most widely used MPGD is the Gas Electron Multiplier (GEM), introduced by Sauli in 1997 [2].

A standard GEM consists of a thin polymer foil (typically 50 μm of polyimide) clad with copper on both faces and perforated with a regular matrix of holes, typically 70 μm in diameter and spaced 140 μm apart. When a voltage difference of a few hundred volts is applied between the two copper faces, a strong electric field is created inside the holes. Electrons entering the holes from one side are accelerated and produce avalanche multiplication, emerging from the other side with a charge gain of typically 10 to 100 per GEM layer. By stacking several GEM layers, total gains of 10^4 to 10^6 can be achieved.

The Thick GEM (THGEM) is a robust variant of the GEM in which the substrate is thicker (typically 0.4 to 1 mm of standard PCB material), the holes are larger (0.3 to 1 mm diameter), and the electrode is manufactured using standard PCB drilling rather than chemical etching. The increased hole size and substrate thickness make the THGEM mechanically robust and straightforward to fabricate, at the cost of a somewhat lower spatial resolution compared to the standard GEM.

The Multilayer Thick GEM (MThGEM) used in this exercise is a stack of four THGEM-like layers, each independently polarised. By distributing the total voltage gain across four stages rather than one, the MThGEM achieves higher total light yield while keeping the field in any single layer below the level that would cause electrical breakdown. In this exercise the MThGEM is operated primarily in the electroluminescence regime, meaning that the field in the holes is sufficient to produce secondary scintillation but is kept below the threshold for avalanche multiplication. This regime provides a linear and stable response, well suited for energy measurement and track imaging.

16.3.5 Secondary scintillation and electroluminescence at the MThGEM

When the drifting electrons reach the region of high electric field inside the MThGEM holes, they gain sufficient energy to excite the gas molecules without necessarily producing additional ionisation. The excited molecules subsequently decay radiatively, emitting secondary scintillation photons. This process is called electroluminescence, and the resulting light signal is referred to as the secondary scintillation signal, or S2 (see Figure 60).

The number of electroluminescence photons produced per drifting electron per unit path length in the high-field region is approximately proportional to the reduced electric field E/p above a threshold value, and can be expressed as:

$$\frac{dN_\gamma}{dx} = A\left(\frac{E}{p} - B\right)\cdot p \tag{20}$$

where A and B are gas-dependent constants. This proportionality ensures that the S2 signal is linearly related to the number of primary electrons arriving at the MThGEM, which is in turn proportional to the ionisation deposited along the corresponding segment of the track.

The four-layer MThGEM used in this exercise consists of four successive PCB substrates, each with a regular matrix of holes metallised on both faces. Each consecutive layer is polarised at a progressively less negative voltage, from the bottom face (most negative, facing the PMT and the drift region) through to the top face of the last layer (least negative or at ground), creating a strong electric field localised inside the holes at each stage. Electrons entering the holes from the drift region are accelerated and produce electroluminescence at each layer.

It should be noted that this scheme requires a cathode with the highest optical transparency possible. The use of metallic meshes is a practical option to maximise transparency while still allowing polarisation of the electrode.

The S2 light propagates both downward, where it is collected by the PMT together with the S1 signal, and upward through the optical viewport to the camera. The charge signal produced by the electrons exiting the top layer of the MThGEM is collected on the grounded anode, which is the uppermost face of the last MThGEM layer. Extracting the signal from this grounded electrode is an important design choice: it ensures that the charge-sensitive preamplifier connected to the anode is not exposed to the high voltage present on the lower MThGEM layers, thereby protecting it from discharges that could cause irreversible damage to the readout electronics.

16.3.6 The two-signal readout scheme

The detector operated in this exercise produces two distinct signals for each event:

- **S1 — Primary scintillation:** Produced promptly along the entire particle track at the moment of the interaction. Collected by the PMT. Provides the event time reference t_0 and a measure of the total energy deposited in the active volume. The S1 signal is relatively faint, as the number of primary scintillation photons per unit energy is limited by the gas pressure and the solid angle subtended by the PMT.

- **S2 — Secondary electroluminescence:** Produced at the MThGEM when the drifting electrons arrive and are accelerated in the high-field region of the holes. Collected simultaneously by the PMT (delayed with respect to S1 by the drift time Δt) and by the digital camera (which images the light emitted upward through the viewport). The S2 signal is brighter than S1, as each primary electron can produce many electroluminescence photons. It carries the spatial information about the projected track topology in the transverse plane.

The PMT waveform therefore contains two temporally separated structures: the prompt S1 pulse and the delayed S2 pulse (or a series of S2 pulses if the track extends over a range of drift distances). The time separation between S1 and S2 gives the drift time of the electrons produced at the interaction vertex, from which the longitudinal coordinate z is reconstructed via Equation 19.

The W-value and its role in signal interpretation. To connect the measured S1 and S2 amplitudes to the deposited energy, it is necessary to introduce the concept of the W-value, defined as the mean energy required to produce one electron–ion pair in the gas:

$$W = \frac{E_{\text{dep}}}{N_e}, \tag{21}$$

where E_{dep} is the total energy deposited by the particle and N_e is the mean number of ion pairs produced. The W-value is a property of the gas and is typically in the range of 20 to 40 eV for common gas mixtures. For $ArCF_4$, $W \approx 27\,\text{eV}$. This means that, for example, a 5.5 MeV alpha particle from a ^{241}Am source deposits enough energy to produce roughly 200 000 primary electron–ion pairs in the gas (before any attachment losses).

The S1 signal is related to the number of primary scintillation photons N_{ph}, which is proportional to E_{dep} through the scintillation yield of the gas. The S2 signal is related to N_e through the electroluminescence yield of the MThGEM. Since both N_{ph} and N_e are proportional to E_{dep}, their ratio S2/S1 is a property of the gas and the operating conditions rather than of the individual event. The effective optical gain G of the detector can therefore be defined as:

$$G = \frac{\text{S2}}{\text{S1}} \tag{22}$$

and reflects the combined effect of the ionisation yield of the gas, the electron transport and extraction efficiency, and the electroluminescence yield per electron in the MThGEM high-field region. Monitoring G as a function of the operating parameters is a sensitive diagnostic of detector performance and stability.

16.3.7 Electric field configuration

The electric field in the detector is defined by the cathode and the four MThGEM layers. The cathode is held at the most negative potential (V_c), establishing the lower boundary of the drift field. The four MThGEM layers are polarised at progressively less negative voltages V_4, V_3, V_2, and V_1 from bottom to top, with the top layer (anode) at ground. The electrode stack is summarised in Table 5.

Table 5: Electrode stack, indicative voltage ranges, and roles.

Electrode	Voltage	Role
Cathode	V_c (e.g. -1000 to $-2000\,\text{V}$)	Drift field boundary
MThGEM layer 1 (bottom)	V_4 (e.g. -600 to $-800\,\text{V}$)	Electron entry face
MThGEM layer 2	V_3 (e.g. -400 to $-600\,\text{V}$)	2nd EL stage
MThGEM layer 3	V_2 (e.g. -200 to $-400\,\text{V}$)	3rd EL stage
MThGEM layer 4 (anode)	$V_1 \approx 0\,\text{V}$	Signal collection

The drift field in the active volume between the cathode and the bottom face of the MThGEM is:

$$E_{\text{drift}} = \frac{|V_c - V_4|}{d_{\text{drift}}} \tag{23}$$

where d_{drift} is the distance between the cathode and the bottom face of the MThGEM. As a general reference, the reduced electric field E/p in the drift region typically

ranges from hundreds of $\mathrm{V\,cm^{-1}\,bar^{-1}}$, while the field inside the MThGEM holes reaches values in the range of tens to hundreds of $\mathrm{kV\,cm^{-1}\,bar^{-1}}$ — several orders of magnitude higher than in the drift region.

16.3.8 Key quantities and expected signals

Before entering the laboratory, it is useful to develop an intuition for the orders of magnitude of the signals expected from this detector. The following estimates are based on $ArCF_4$ 90/10 at a representative operating pressure of 100 mbar, with a 5.5 MeV alpha source (^{241}Am) and a drift length of 10 cm. They are approximate and serve only as a guide for judging whether the detector is behaving correctly.

- **Primary ionisation:** $W \approx 27\,\mathrm{eV} \Rightarrow \sim$200 000 electron–ion pairs per 5.5 MeV alpha track.
- **S1 photons collected by PMT:** typically 10^2 to 10^3 (limited by solid angle and gas pressure).
- **S2 photons per primary electron:** typically 10 to 100 (depends on MThGEM voltage and E/p).
- **Total S2 photons:** $\sim 10^6$ to 10^7 (before optical losses).
- **Electron drift velocity** in $ArCF_4$ at $\sim 100\,\mathrm{V\,cm^{-1}\,bar^{-1}}$: $\sim$5 to 10 $\mathrm{cm\,\mu s^{-1}}$.
- **Expected S1–S2 time separation** for 10 cm drift: $\sim$1 to 20 μs depending on drift field.
- **MThGEM signal charge at anode:** of order fC to pC, requiring a charge-sensitive preamplifier.
- **Camera exposure time:** typically 1 to 100 ms for single-event imaging at low pressure.

These numbers imply that the S1 pulse will be small and fast (nanosecond rise time), while the S2 signal will be larger and arrive several microseconds later. The oscilloscope record length must be set to capture both within the same acquisition window. If the S2 pulse is not visible, the most likely causes are: MThGEM voltage too low, gas purity insufficient, or oscilloscope time window too short.

16.4 Experimental Setup

Safety note — read before starting

- **HIGH VOLTAGE HAZARD:** The cathode and MThGEM layers operate at voltages up to several kV. Always ramp HV slowly and never touch any electrode or feedthrough while HV is applied.
- **DISCHARGE HAZARD:** If a discharge occurs (sudden current spike, audible click, or loss of HV), ramp all voltages to zero immediately and wait at least 60 seconds before attempting to ramp up again.
- **GAS HANDLING:** $ArCF_4$ is an asphyxiation hazard in enclosed spaces. Ensure adequate ventilation. CF_4 is also a potent greenhouse gas — minimise releases and close all gas lines when not in use.
- **PREAMPLIFIER PROTECTION:** Never apply HV to the MThGEM before verifying that the anode signal is taken from the grounded top layer. Connecting the preamplifier to a biased electrode will destroy it.
- **RADIOACTIVE SOURCE:** Handle the radioactive source only as instructed. Do not remove it from the vessel. Notify the supervisor immediately if the source is dropped or damaged.

16.4.1 Detector vessel and gas system

The detector is housed in a gas-tight vessel capable of operating at low pressure, in the range of 10 to 500 mbar. The vessel provides electrical feedthroughs for the high voltage connections to the cathode and all four MThGEM layers, signal feedthroughs for the PMT and anode readout, a gas inlet and outlet for filling and optional circulation, and an optical viewport on the top (or side) flange for the camera.

The gas used in this exercise is $ArCF_4$ in a 90/10 volumetric ratio at the operating pressure chosen for the session. The gas system includes a pressure gauge to monitor and stabilise the filling pressure. Figure 61 shows a schematic of the assembled setup.

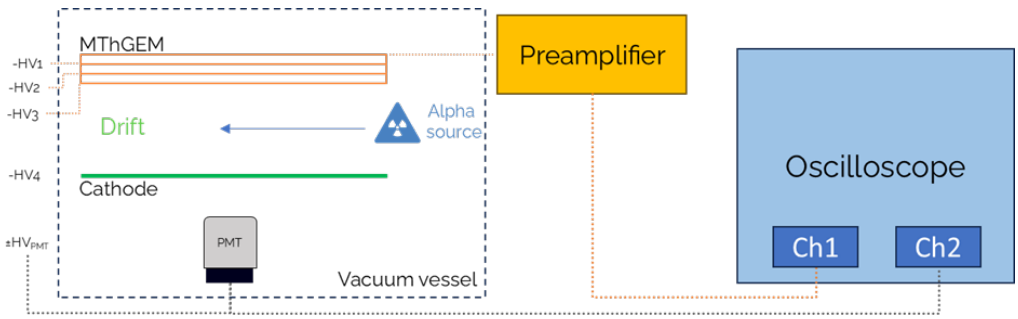


Figure 61: Schematic of the detector assembly showing the main components and their relative positions.

16.4.2 Detector layer stack

The detector is assembled as a vertical stack of functional layers. Starting from the bottom and moving upward, the layers are the following.

Photomultiplier tube (PMT): Installed at the bottom of the vessel, with its photocathode facing upward into the active volume. It collects both the S1 primary scintillation light produced along the track and the downward-propagating component of the S2 electroluminescence light produced at the MThGEM. The PMT must have a spectral response covering the visible emission band of the gas mixture, and its window must be compatible with the operating gas environment.

Cathode: Immediately above the PMT, the cathode is a thin metallic mesh held at the most negative high voltage in the system (V_c). It defines the lower boundary of the drift electric field and must be sufficiently transparent to allow the S1 and downward S2 photons to reach the PMT.

Drift region: The active gas volume between the cathode and the bottom face of the MThGEM constitutes the drift region. A radioactive source is mounted on the side wall of the vessel at a fixed position within the drift region, so that the particles it emits traverse the active gas and produce ionisation and scintillation tracks at a well-defined location. The source is oriented so that the tracks are directed into the active volume rather than toward the vessel walls.

MThGEM (four-layer): The four-layer Multilayer Thick GEM is positioned above the drift region. It consists of four successive PCB substrates, each with a regular matrix of holes metallised on both faces. The four layers are polarised at progressively less negative voltages V_4 through V_1, with the top face at or near ground. Electrons drifting upward enter the holes of the first layer and are progressively accelerated through each subsequent layer, producing electroluminescence at each stage.

Anode readout layer: The grounded top face of the fourth MThGEM layer constitutes the anode. The charge signal is collected here and provides information on the track position. The signal is extracted from this grounded electrode specifically to protect the downstream preamplifier from high-voltage discharges.

Charge-sensitive preamplifier: The charge-sensitive preamplifier (like an Ortec 142) integrates the input charge onto a feedback capacitor, producing a voltage step at its output proportional to the collected charge. The output signal decays exponentially with a time constant of typically tens to hundreds of microseconds. The preamplifier must be connected only to the grounded anode of the MThGEM top layer; it must never be connected to any biased electrode.

Optical viewport: The top (or side) flange of the vessel incorporates a transparent window, compatible with the operating gas and pressure, that transmits the upward-propagating S2 electroluminescence light out of the vessel. The window material must be chosen to match the emission spectrum of the gas mixture.

Digital camera: A digital camera (CCD, CMOS or Timepix-based device) is mounted externally above the viewport, looking downward through the window at the top face of the MThGEM. It images the spatial distribution of electroluminescence produced in the MThGEM holes, providing a two-dimensional projection of the track. The camera is triggered by the S1 signal from the PMT.

The complete detector layer stack is summarised in Table 6.

Table 6: Detector layer stack, signals, and high voltage assignments.

Position	Component	Signal	HV
Bottom	PMT	S1 + S2 (optical)	PMT bias
Above PMT	Cathode (mesh)	—	V_c (most negative)
Middle	Drift region + lateral source	—	Field defined by boundaries
Above drift	MThGEM layer 1 (bottom)	—	V_4
	MThGEM layer 2	—	V_3
	MThGEM layer 3	—	V_2
	MThGEM layer 4 top (anode)	Charge → ORTEC 142 → oscilloscope	V_1 / Ground
Top flange	Optical viewport	—	—
External top	Digital camera	2D track image (S2 optical)	—

16.4.3 Electronics and data acquisition

The DAQ system comprises the following components:

- High voltage power supply for the cathode, with slow ramp-up capability and current monitoring.
- High voltage power supply for the MThGEM, with four independent output channels for layers V_4 through V_1, each with individual current monitoring to detect instabilities or discharges in any single layer.
- PMT bias supply.
- Charge-sensitive preamplifier (ORTEC 142 or equivalent), connected to the grounded anode of the MThGEM top layer. The preamplifier output is a slow, exponentially decaying voltage pulse whose amplitude is proportional to the collected charge.
- Oscilloscope with data acquisition and waveform export capabilities, with sufficient sampling rate and record length to capture both the S1 and S2 pulses within the same acquisition window, and allowing waveform data to be saved to file for offline analysis.
- Digital camera.
- A laptop or PC for waveform export, image capture, and offline analysis.

16.5 Procedure

The procedure is divided into seven steps. Steps 1 through 3 are essential commissioning steps that must be completed before any data can be taken. Steps 4 through 6 are the primary measurement steps. Step 7 is an exploratory parameter scan. The expected total duration of the session is approximately four hours.

Step 1 — Gas filling and pressure setting

- Evacuate the vessel to a base pressure using a roughing pump with the setup inside.
- Fill the vessel with $ArCF_4$ 90/10 to the target operating pressure (gas and pressure to be defined at the session).
- Verify the pressure reading on the gauge and close the gas inlet.
- Allow the gas to stabilise for several minutes before applying high voltage.

Step 2 — High voltage ramp-up and stability check

- Ramp the PMT bias to its operating voltage (considering the operating pressure, the operating voltage may vary).
- Slowly ramp the cathode voltage to the target drift value, monitoring the current.
- Slowly ramp the MThGEM voltages, monitoring the current for signs of discharge or instability while increasing the cathode voltage to keep the drift value.
- Record the stable operating voltages and currents.

Step 3 — PMT signal inspection on the oscilloscope

- Connect the PMT output to the oscilloscope.
- Before looking at real data, sketch on paper the waveform you expect to see: draw the S1 pulse at time t_0, the S2 pulse delayed by Δt, and label the relevant timescales based on the order-of-magnitude estimates in Section **??**.
- Identify the S1 pulse: prompt, relatively small amplitude. Compare with your sketch.
- Identify the S2 pulse: delayed with respect to S1 by the electron drift time, larger amplitude.
- Measure the S1–S2 time separation Δt and estimate the mean electron drift velocity using the known source position and drift length.

Step 4 — PMT waveform acquisition and S1/S2 analysis

- Connect the PMT output to the waveform digitiser.
- Acquire a run of PMT waveforms with the DAQ software.
- For each event, integrate the S1 pulse area and the S2 pulse area separately.
- Build histograms of the S1 integral, the S2 integral, and the S2/S1 ratio.
- Identify the alpha particle peak in the S1 and S2 spectra.
- Measure the mean drift time $\langle \Delta t \rangle$ from the S1–S2 time difference distribution.

Step 5 — MThGEM charge readout

- Read out the charge collected on the MThGEM anode in coincidence with the PMT trigger.
- Produce the one-dimensional charge profile along the strip direction and identify the projected track position.

Step 6 — Camera imaging

- Configure the camera with the appropriate exposure time.
- Acquire a set of single-event images.
- Identify the track topology in the camera images: straight alpha tracks from the source, and any secondary delta-electron tracks.
- Accumulate multiple events to build a summed image showing the average track distribution.

Step 7 — Parameter variation

- Vary the MThGEM voltage by a small increment and observe the effect on the S2 pulse amplitude and the camera image brightness.
- Vary the drift field by adjusting the cathode voltage and observe the effect on the S1–S2 time separation and on the transverse diffusion visible in the camera images.
- Record all parameter settings systematically for use in the analysis.

16.6 Analysis and Interpretation of the Results

1. Electron drift velocity

From the measured mean S1–S2 time separation $\langle \Delta t \rangle$ and the known distance d between the source position and the MThGEM, estimate the electron drift velocity:

$$v_d = \frac{d}{\langle \Delta t \rangle} \tag{24}$$

Compare the result with published drift velocity values for $ArCF_4$ at the operating pressure and reduced electric field E/p.

2. Energy information from S1 and S2

Discuss what physical quantities are encoded in the S1 and S2 integrals. Discuss the meaning of the S2/S1 ratio and what detector properties it reflects. Using the known alpha energy and the W-value for $ArCF_4$, estimate the number of primary electron–ion pairs produced and compare with the S2 amplitude observed.

3. Track length and range in the gas

From the camera images, measure the projected length of the alpha particle tracks. The energy loss of a heavy charged particle in matter is described by the Bethe–Bloch formula, which predicts a slowly varying energy loss along most of the track followed by a sharp maximum near the end of the range, known as the Bragg peak. Using the known source energy and the operating gas pressure, calculate the expected range from tabulated stopping power data (using Lise++) and compare with the measured projected track length. Discuss any discrepancy in terms of the projection geometry.

4. Three-dimensional reconstruction

Describe how the full three-dimensional track would be reconstructed from the combination of the camera image (transverse projection) and the drift time information (longitudinal coordinate). Identify which coordinate is provided by each readout channel and what reference is needed for the longitudinal reconstruction via Equation 19.

5. Spatial resolution

Discuss the contributions to the spatial resolution of the detector:

- Transverse diffusion of the electron cloud during drift, characterised by:

$$\sigma_T = \sqrt{2\, D_T\, t_{\text{drift}}} \tag{25}$$

- MThGEM hole pitch, which sets the minimum granularity of the electroluminescence pattern.
- Camera pixel size and optical magnification factor.
- Source size and angular spread.

Estimate the dominant contribution at the operating conditions of the session.

6. Effect of operating parameters

Using the data acquired during the parameter variation steps, discuss:

- How the S2 amplitude depends on the MThGEM voltage, and what this implies for the electroluminescence yield.
- How the drift field affects the electron drift velocity and the transverse diffusion.
- What would happen to the detector performance if the gas pressure were doubled or halved.

16.7 Bibliography

References

[1] F. Sauli, *Gaseous Radiation Detectors: Fundamentals and Applications*, Cambridge University Press, 2014.

[2] F. Sauli, "GEM: A new concept for electron amplification in gas detectors", *Nuclear Instruments and Methods in Physics Research A*, Vol. 386, pp. 531–534, 1997.

[3] F. Sauli, "The gas electron multiplier (GEM): Operating principles and applications", *Nuclear Instruments and Methods in Physics Research A*, Vol. 805, pp. 2–24, 2016.

[4] M. Cortesi, R. Lagomarsino, S. Nessi, C. Pistillo, P. Campagnolo, V. Peskov, and A. Breskin, "Investigation of a thick GEM-like multiplier operated at low temperatures", *Journal of Instrumentation*, Vol. 4, P08001, 2009.

[5] M. Cortesi, V. Peskov, G. Bartesaghi, J. Miyamoto, S. Kobayashi, T. Namba, T. Nakaya, M. Yokoyama, K. Ankowski, R. Sulej, H. Hamar, and A. Breskin, "THGEM operation in Ar and Ar-CH_4 gas mixtures", *Journal of Instrumentation*, Vol. 4, P08001, 2009.

[6] M. Cortesi et al., "Optical readout of a ThGEM-based detector for charged particle tracking", *Journal of Instrumentation*, to be confirmed with the professors.

[7] G.F. Knoll, *Radiation Detection and Measurement*, 4th ed., John Wiley and Sons, 2010.

[8] W.R. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-To Approach*, Springer, 1994.

Exercise 17 Artificial Intelligence for Radon Monitoring: Alpha-Particle Detection in a Cloud Chamber

17.1 Objectives

The objective of this exercise is to study the operation of a diffusion cloud chamber and use it to visualize charged particle tracks from cosmic radiation and natural radioactive sources. By analyzing the characteristics of the observed tracks, particularly α particles, we aim to identify different types of radiation and relate their properties to the underlying particle physics.

Additionally, the experiment aims to estimate the atmospheric radon concentration by counting α-particle events and applying geometric and efficiency corrections. Finally, the performance of AI-based track identification tools is evaluated and compared with manual event classification to assess their reliability in radiation detection tasks.

17.2 Professors

- Luca Fiorini
- José Enrique Garcia Navarro

17.3 Introduction

Purpose

The diffusion cloud chamber is a unique device for visualizing the tracks of different kinds of radiation and identifying them from differences in their length, thickness (ionization density or *stopping power*), and topology. Therefore, the basic aim of this session is to directly visualize and distinguish the tracks of α particles, protons, electrons, photons, and even muons that reach the chamber from cosmic rays or natural terrestrial radioactivity, directly illustrating the consequences of the Bethe–Bloch formula as well as photon interaction mechanisms. The discrimination between different radiation types will be used to measure the concentration of atmospheric radon. In addition, the exercise introduces the technique of image scanning (in this case a video recording) for the search for events, a technique that led to many of the great discoveries in Particle Physics, such as the positron (the first antiparticle discovered) and the muon. Modern Artificial Intelligence (AI) tools applied to automatic visual recognition and counting of α particles are introduced.

Figure 62: The diffusion cloud chamber used in the practice session.

Background

The diffusion cloud chamber is a particle detector that allows direct observation of the trajectories of charged particles as they pass through the active volume of the chamber. With this chamber we can observe different kinds of tracks associated with cosmic and terrestrial radiation, as well as with artificial radioactive sources like ^{90}Sr.

The diffusion chamber is an improved version of the Wilson cloud chamber, which historically played a fundamental role in the development of Nuclear and Particle Physics. Both are based on the same principle: when an ionizing particle passes through an atmosphere of air supersaturated with water or alcohol vapor, the vapor condenses around the ions formed by the radiation, and the particle track becomes visible.

In Wilson's original chamber, the air inside the sealed device is saturated with water vapor, and a diaphragm is used to expand the air inside the chamber (adiabatic expansion). This cools the air and the water vapor begins to condense around the ions formed by the radiation. The tiny liquid droplets formed along the radiation path can be photographed. If the chamber is also placed inside an electric or magnetic field, the trajectories appear curved by that field, making it possible to determine the charge of the incident radiation. Examples include Anderson's discovery of the positron in 1932 and the muon in 1936, using a cloud chamber in a magnetic field exposed to cosmic rays (Figure 63).

The diffusion cloud chamber was developed in 1936 by Alexander Langsdorf. This chamber differs from Wilson's expansion chamber in that it is continuously sensitive to radiation, since no expansion cooling is used. Instead, a thermal gradient is generated by cooling the bottom of the chamber with a refrigeration unit or dry ice, thus permanently maintaining the supersaturated mixture in this part of the chamber.

17.4 Description of the chamber

The diffusion cloud chamber used in this laboratory session is located in the hall of the IFIC laboratory building (Figure 62). It consists of a base and an observation

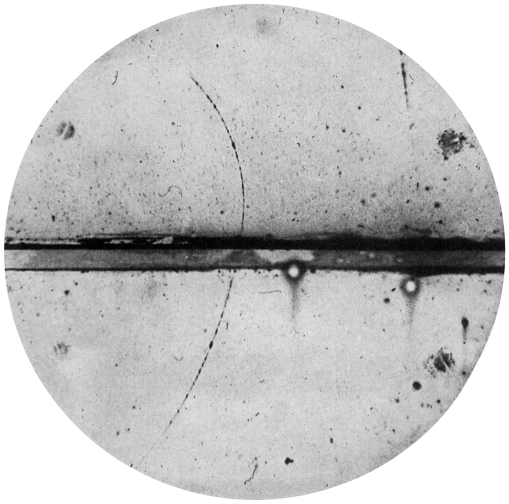


Figure 63: Discovery of antimatter from the track of a positron observed by Anderson in 1932 in a Wilson diffusion chamber. The positron enters the chamber from below and crosses the central lead plate (magnet support), losing energy, so its curvature due to the magnetic field increases.

chamber. The chamber base contains the refrigeration unit, the power supply, the alcohol tank, and the alcohol pump. The observation chamber is located above the base. Figure 64 shows a schematic of the chamber. The bottom of the observation chamber is a solid black metal plate (surface area 45 cm $\times$ 45 cm) cooled uniformly over its entire surface by the refrigeration unit (approximately -35°C). The roof and walls of the observation chamber consist of two superposed glass covers. Between the two covers there is a mesh of thin heating wires used to warm this part of the chamber (up to about $+35$°C), creating the necessary temperature gradient and preventing fogging. The wires also form a high-voltage grid for ion removal. At the top, just below the glass cover, there is an electrically heated channel running all around the cover, inside which isopropyl alcohol drips from a small nozzle. The alcohol evaporates and diffuses from the warm upper region toward the cold bottom. The alcohol condenses at the bottom and returns to the tank.

Since the height of the observation volume is about 15 cm, the temperature gradient is ≈ 4.7°C/cm. This value is above the threshold of 3.6°C/cm required to form a layer of supersaturated vapor in the air–alcohol mixture at 1 atm pressure (the chamber is not sealed and is therefore at atmospheric pressure), above the thin liquid layer covering the bottom. Only in this region, with a thickness of only about 5 mm, do charged particles or photons coming from outside or inside the chamber generate ions along their trajectories, forming condensation nuclei around which the alcohol droplets appear, producing the fog tracks visible to the observer (Figure 65). The length, thickness, and topology (shape) of the tracks make it possible to study some of the properties of the particles that produce them.

17.5 Track visualization

The diffusion cloud chamber allows visualization of all charged particles (electrons/-positrons, protons, α particles, and muons). Photons can be visualized indirectly through the production of photoelectrons, Compton electrons, or even pair produc-

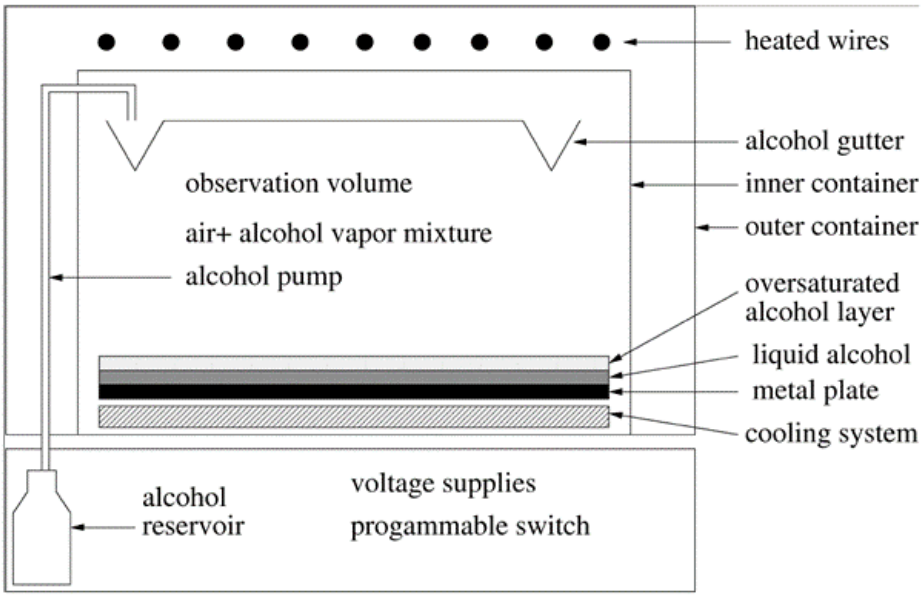


Figure 64: Schematic cross-sectional view of the diffusion cloud chamber.

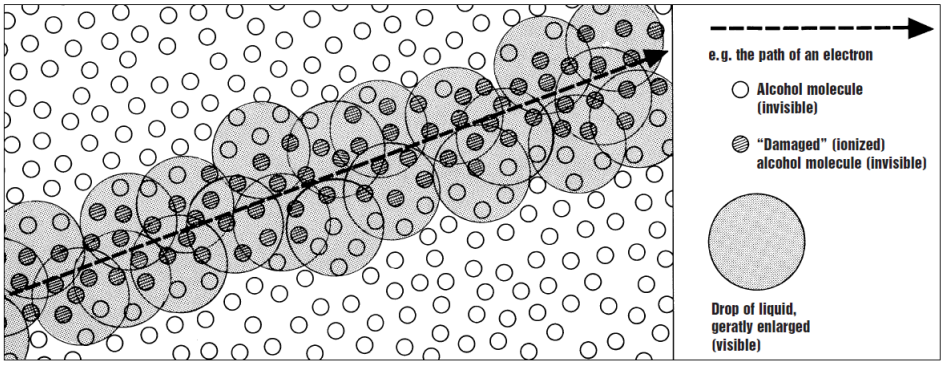


Figure 65: Track formation in the cloud chamber.

tion when the photon energy exceeds twice the electron mass. Finally, neutrons can produce a nuclear reaction with nuclei in the chamber volume, and the charged particles and nuclei produced will leave their traces in it.

Bethe–Bloch formula

The relationship between range and energy at intermediate and high energies is obtained from the quantum electrodynamics (QED) calculation of the Coulomb collision process (excitation and ionization) with the atomic electrons of the medium (Bethe–Bloch formula) [2, 3]:

$$S_{\text{col}} = \frac{1}{\rho}\left(-\frac{dE}{dx}\right)_{\text{col}} = K\frac{Z}{A}\frac{z^2}{\beta^2}\left[\ln\left(\frac{2m_e c^2\beta^2\gamma^2 E_{\max}}{I^2}\right) - 2\beta^2 - \delta - 2\frac{C}{Z}\right] \text{ MeV cm}^2\,\text{g}^{-1}\,, \tag{26}$$

where $K = 2\pi N_{\text{A}} r_e^2 m_e c^2 = 0.1535$ MeV cm^2 g^{-1}, r_e is the classical electron radius, m_e the electron mass, and N_{A} Avogadro's number. The mean excitation potential I can be parameterized as

$$I \approx \begin{cases} 12Z + 7 \text{ [eV]} \ , & Z < 13 \\ 9.76Z + \frac{58.8}{Z^{0.19}} \text{ [eV]} \ , & Z \geq 13 \end{cases} \quad \text{[eV]}, \tag{27}$$

where Z is the atomic number of the material, A the atomic weight of the material, ρ the density of the material, z the particle charge, $E_{\max} = 2m_e c^2 \beta^2 \gamma^2$ the maximum energy transferred ($M >> m_e$), β the particle speed, δ the density correction (shielding effect, ultrarelativistic particles in dense media), and C the shell correction (when the particle speed is much greater than the orbital speed of the atomic electrons). At low energies the values are obtained from experimental stopping-power data. Their values for different materials and energies can be obtained from the NIST database [4].

The ionization produced in a gas depends only slightly on the gas type. In air, the average ionization energy is 34 eV [2, 3].

The α particles

The α particles from the decay of radionuclides in the radon series have a typical energy of 5 MeV, with a range in most solids below 100 μm and in air of about 5 cm. An α particle of 5 MeV produces on the order of 150,000 ions over 5 cm, which generates substantial condensation and leads to a thick, bright track, as shown in Figure 66. The trajectory of the α particle will deviate only when it interacts with the nuclei of the air–alcohol vapor mixture (multiple scattering). When the α particle enters the supersaturated vapor layer vertically, we will only be able to see a thick point as the particle track.

Figure 66: α particle (oblique track) photographed in the IFIC chamber.

Protons

Energetic protons from secondary cosmic rays can cross the glass and enter the chamber. The Bethe–Bloch formula predicts that protons with four times less energy than α particles have the same range in a given material (scaling rules). Therefore, a proton entering the chamber with 1.25 MeV will produce four times less ionization than a 5 MeV α particle over the same distance, leading to a somewhat thinner track, as illustrated in Figure 67. The length of proton tracks varies with their energy. Above 1 GeV, protons are minimum-ionizing particles (MIPs), and the track would be much fainter and much longer. Below 6 MeV, protons lose all their energy in the chamber, since their range is about 40 cm, which is the length of the chamber, provided they enter the active volume horizontally. As in the case of α particles, protons follow straight trajectories except when they undergo multiple scattering.

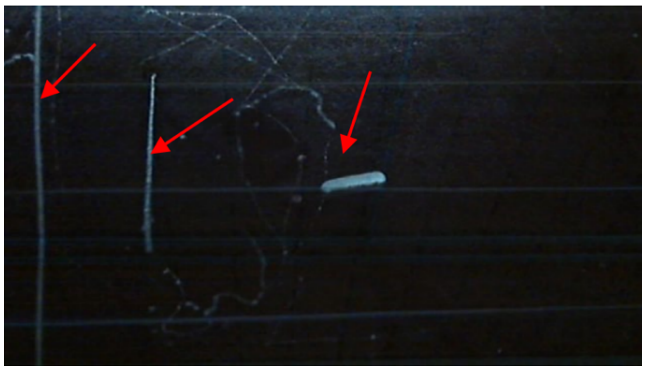


Figure 67: Two very energetic vertical particles and one horizontal α particle photographed in the IFIC chamber.

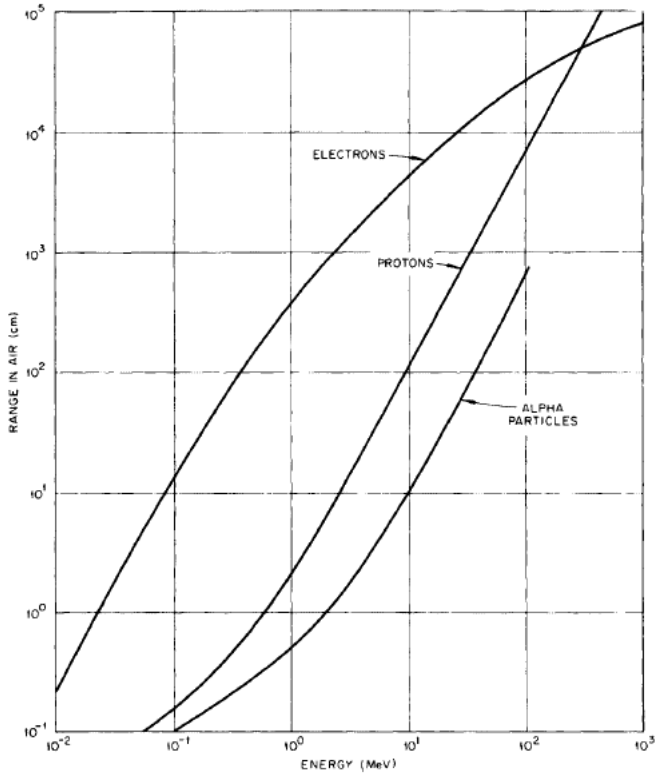


Figure 68: Range–energy relation for electrons, protons, and α particles in air under standard temperature and pressure conditions. Oak Ridge National Laboratory.

17.6 Radon gas

A common feature of the natural uranium–radium, thorium, and actinium chains is the existence of radon isotopes. Radon ($Z = 86$) is the heaviest monoatomic gas found in nature. It is colorless, odorless, and tasteless, and has virtually no chemical activity with respect to other elements in the periodic table. Because of its gaseous nature, it diffuses easily through different media. Three isotopes of radon are known: ^{222}Rn, ^{220}Rn, and ^{219}Rn, called radon, thoron, and actinon, respectively. They are unstable, and each belongs to a different natural radioactive chain. Radon belongs to the uranium–radium series (Figure 69), thoron to the thorium series, and actinon to the actinium series. These radioactive isotopes are the only gaseous elements in

their respective radioactive chains, and when they decay they produce isotopes of heavy metals (Po, Pb, Bi).

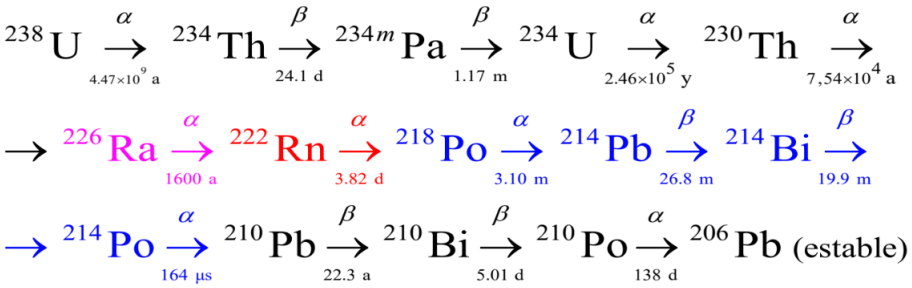


Figure 69: Uranium–radium series. The colored section shows the decay chain of ^{222}Rn and its short-lived daughters. Note that ^{222}Rn is produced by the decay of ^{226}Ra.

The most stable isotope is radon (^{222}Rn), with $T_{1/2} = 3.82$ d. ^{222}Rn is a direct descendant of ^{226}Ra through an α decay. Its decay products form a series of radionuclides with short decay periods, which decay within a few hours to ^{210}Pb, whose period is 22 years. Because of their short half-lives, radon daughters quickly approach secular equilibrium with their parent.

Actinon (^{219}Rn), due to its decay period ($T_{1/2} = 3.96$ s), has limited migration capability. This, together with the isotopic abundance of ^{235}U in natural uranium (0.7%), makes its presence in the atmosphere negligible. The same applies to thoron (^{220}Rn), with a short half-life ($T_{1/2} = 55.6$ s) that inhibits its mobility with respect to that of ^{222}Rn. Therefore, in any atmosphere, the dominant radon isotope will be ^{222}Rn. As illustrated in Figure 70, once in the atmosphere, the decay of ^{222}Rn produces isotopes of polonium, lead, and bismuth. These are heavy metals, highly chemically active, which quickly form molecules in the condensed phase. Attached to dust particles, they form radioactive aerosols that can be inhaled. The danger of exposure to high radon concentrations therefore does not come from the gas itself, but from the next four isotopes in the series, which are short-lived and whose particles become fixed in lung tissue, irradiating it. Since the element with the longest period is 27 minutes, the whole sequence of decays can be completed before the various cleansing systems of the human body can eliminate them, so the radiation is dissipated within cellular matter, with the health problems this may entail. Reference [5] contains a detailed description of this mechanism.

17.6.1 Measurement of atmospheric radon concentration

The concentration of radon in the atmosphere strongly depends on the region and on ventilation and environmental conditions. For example, the average in the Stockholm Museum of Science is 220 Bq/m^3, varying from the canonical value of 40 Bq/m^3 during the day to 700 Bq/m^3 when ventilation is turned off at night. The IFIC diffusion cloud chamber is located in a large hall on the first floor of the laboratory building, with no direct outside ventilation (windows), but with good artificial ventilation and surrounded by offices with good natural ventilation. Ventilation is turned off at night and on weekends. These data are important for the interpretation of the results when comparing with typical values in Europe and in Valencia, in the range 30–40 Bq/m^3.

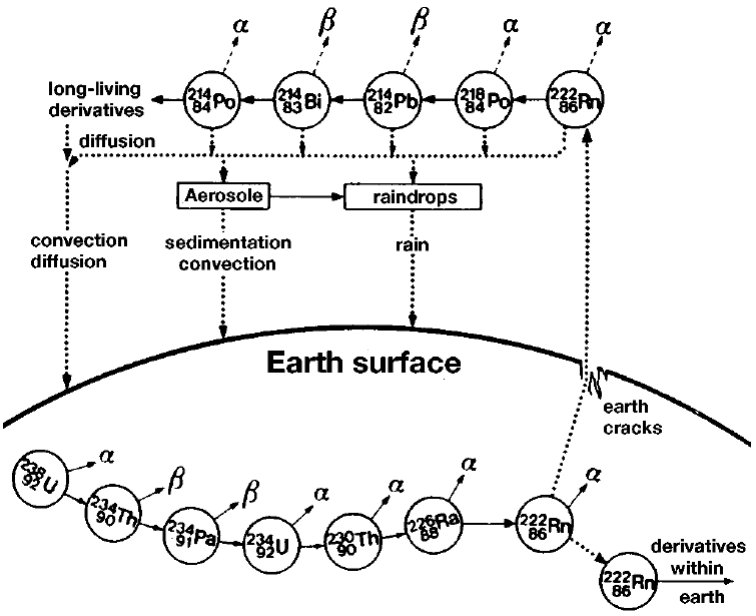


Figure 70: Illustration of the production of ^{222}Rn and its progeny in the lower layers of the atmosphere.

Since ^{222}Rn takes only about 3.5 hours to reach secular equilibrium with its short-lived daughters (^{218}Po, ^{214}Pb, ^{214}Bi, and ^{214}Po), we should observe the α particles from ^{222}Rn itself (5.49 MeV), from ^{218}Po (6.00 MeV), and from ^{214}Po (7.69 MeV) with the same frequency. Lead and bismuth isotopes decay through β processes with short half-lives and their tracks are very different (thin). ^{214}Po produces ^{210}Pb, which has a period of 22 years, so for our practical purposes we can consider it stable.

The diffusion chamber allows us to visualize the tracks of α particles produced in its active volume, a volume that we can estimate because we know the side lengths ($L_x = L_y = 40 \pm 2$ cm) and the thickness of the supersaturated air layer, $\epsilon_z = 5.0 \pm 2.0$ mm. If during a time interval of $t_0 = 3$ minutes we count the number N_0 of thick and bright tracks shorter than about 7 cm, we can obtain an estimate of the number of α particles per unit time and volume. We can compare this value with the data from reference [6], which provide a measurement of radon levels in Valencia and nearby areas.

To convert the measured number of α tracks into radon concentration, we must take into account the dimensions L_x, L_y, and ϵ_z of the active chamber volume. In addition, we must keep in mind that, since ϵ_z is only a few mm, most of the observed tracks come from outside the active volume. Since we are only interested in tracks produced within the active volume, we must introduce a geometric correction factor $\epsilon_{\rm vol}$. This factor can be obtained analytically,

$$\epsilon_{\rm vol} = \frac{\epsilon_z}{\epsilon_z + L/4} \ , \tag{28}$$

where L is the maximum length of α tracks in air. Alternatively, a Monte Carlo simulation technique can be used that explicitly takes into account the finite size of the solid metal plate (L_x, L_y finite). In our case, the simulation gives a value $\epsilon_{\rm vol} \approx 30\%$, which agrees with the analytical calculation, as expected since $\epsilon_z << L_x, L_y$.

For α energies of 5.5, 6.0, and 7.7 MeV, and assuming that the material in the active volume is dry air (the alcohol vapor is mixed with air because the chamber

volume is not sealed), the NIST database [4] gives mass ranges (*Projected range*) of 5.010×10^{-3}, 5.734×10^{-3}, and 8.517×10^{-3} g/cm^2. If we divide by the air density at about 25 °C, $\rho = 1.20 \times 10^{-3}$ g/cm^3, we obtain for L the values 4.2, 4.8, and 7.1 cm, respectively. The last two isotopes are produced in the decay of the first and quickly reach secular equilibrium, so the activity of the three is essentially the same. This equilibrium may be affected in confined areas, but as a good approximation we may assume it holds, so we can take the activity per radon decay of the three isotopes to be the same, $A_{\text{iso}} = 3.0 \pm 0.5$, and on average $L = 5.3 \pm 1.5$ cm.

With all this, we can evaluate the activity of radon in air per unit volume (concentration) as

$$A = \frac{\epsilon_{\text{vol}}}{\epsilon_{\text{obs}}} \frac{N_0}{t_0} \frac{1}{L_x L_y \epsilon_z} \frac{1}{A_{\text{iso}}} = \frac{1}{\epsilon_{\text{obs}}} \frac{N_0}{t_0} \frac{1}{L_x L_y} \frac{1}{\epsilon_z + L/4} \frac{1}{A_{\text{iso}}} \ , \tag{29}$$

where $\epsilon_{\text{obs}} = (100 \pm 10)\%$ is the overall observation efficiency for α particles in the chamber.

17.7 Radon concentration using AI tools

In this section automated image recognition techniques are used to identify α particle tracks in a cloud chamber recording and to determine the atmospheric radon concentration. The performance of the AI algorithm is evaluated through comparison with a manual analysis. In addition, manually labelled images are produced to validate and improve the identification of α decays.

Procedure

1. Divide the provided video recording into six time intervals of duration $t_0 =$ 3 min, each treated as an independent measurement.

2. Process the video using the Artemisa [8] or Colab Notebook [7]:

   - load the trained model and run the full analysis
   - obtain the annotated video with identified α candidates

3. For each interval, record the number of detected candidates N_{AI}, and use the first interval to evaluate the AI performance. Identify α tracks as thick, bright, short tracks with length $L \lesssim 7$ cm, and compare with the AI output to classify events as TP, FP, and FN. From these, compute

$$\text{Efficiency} = \frac{TP}{TP + FN}, \tag{30}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{31}$$

   and correct the counts as

$$N_{\text{corr}} = N_{\text{AI}} \frac{\text{Efficiency}}{\text{Precision}}. \tag{32}$$

4. Compute the radon activity per unit volume:

$$A = \frac{1}{\epsilon_{\rm obs}} \frac{N_{\rm corr}}{t_0} \frac{1}{L_x L_y} \frac{1}{\epsilon_z + L/4} \frac{1}{A_{\rm iso}}, \tag{33}$$

   where all quantities are defined in Section 17.6.1.

5. Repeat the calculation for all intervals and determine the mean value of A and its dispersion.

6. Compare the result with typical values $A \simeq 30$–40 Bq/m^3.

7. Label the assigned cloud chamber images:

   - connect to the server `tical52.ific.uv.es`
   - open the image set assigned to your group
   - identify and label the visible α decays
   - avoid labelling ambiguous tracks unless they can be clearly classified as α candidates

Discussion

Discuss the impact of efficiency and precision on the final result, the stability of the measurements across time intervals, the agreement with expected environmental values, and the dominant sources of uncertainty. Comment on the role of labelled data in improving AI-based track identification. Optionally, compare the results obtained using the AI-based analysis with those obtained from a purely manual count.

References

[1] T.K. Gaisser and T. Stanev, *Review on Cosmic Rays* in *The Review of Particle Physics (2010)*, K. Nakamura *et al.* (Particle Data Group), J. Phys. G 37, 075021 (2010), `http://pdg.lbl.gov/2010/reviews/rpp2010-rev-cosmic-rays.pdf`

[2] K.S. Krane, *Introductory Nuclear Physics*, John Wiley & Sons Inc. (1988).

[3] G.F. Knoll, *Radiation Detection and Measurement*, John-Wiley & Sons Inc. (2000).

[4] M.J. Berger, J.S. Coursey, M.A. Zucker and J. Chang, *Stopping-Power & Range Tables for Electrons, Protons, and Helium Ions*, NIST Standard Reference Database 124, `https://www.nist.gov/pml/stopping-power-range-tables-electrons-protons-and-helium-ions`

[5] J.J. Llerena Cristobo, *Measurement of the concentration of Rn-222 gas inside buildings*, DEA thesis (2006), Faculty of Physics, University of Santiago de Compostela.

[6] R. Amorós *et al.*, *Radon levels in the Valencian Community (Spain). Results in the province of Valencia (1990-1991)*, J. Environ. Radioactitity, vol. 27, 125-131 (1995).

[7] Google Colaboratory, `https://colab.research.google.com/notebooks/welcome.ipynb?hl=en`

[8] Artemisa Notebook, `https://artemisahub.ific.uv.es`

[9] R. Cases, E. Ros, J. Zúñiga, *Measuring radon concentration in air using a diffusion cloud chamber*, Am. J. Phys. 79 (9) (2011). `http://aapt.org/ajp`

Funded by CEX2023-001292-S
ED2024-153684-T

CIPROM/2022/070