Silicon Photomultiplier characterization Experience

Matthew M., Anosh I., James B., Jackie B.

September 2019 - Prepared as part of the VT-PHYS-4316(S19)

1 SiPM

This experiment uses Silicon Photo-Multiplers (SiPMs) which are an high density matrix ($\sim 10^4/mm^2$) of Silicon avalanche photo-diodes (APDs). Silicon photo-multipliers, often called "SiPM" in the literature, are solid-state single-photon-sensitive devices based on Single-photon avalanche diode (SPAD) implemented on common silicon substrate. The Silicon Photo-multiplier (SiPM) is a sensor that addresses the challenge of sensing, timing and quantifying low-light signals down to the single-photon level.

The silicon photo-multiplier (SiPM) is a radiation detector with extremely high sensitivity, high efficiency, and very low time jitter. It is based on reversed biased https://www.overleaf.com/project/5c758f0ec184c1445ee1719cp-n diodes and can directly detect light from near ultra-violet to near infrared. SiPMs are employed in all those applications where low light/radiation levels must be measured and quantified with high precision.

1.1 SiPMs and APDs - Introduction

When a photon travels through the Silicon material inside either the SiPM or the APD can deposit or absorb energy from any of the bound electrons in the material. The energy absorption will form an electron-hole pair via the photoelectric effect and ejects an electron. Due to energy conservation, photoelectric effect can only take place if the energy of the photon exceeds the binding energy of the electron, typically a few tens of electron volt (eV). The distance that the photon will penetrate the Silicon depends from the incoming photon's energy, and therefore it's wavelength. A graph of absorption depth versus wavelength is shown in Figure 1. As a result of that, how efficient the sensor depends on the wavelength.

A SiPM is comprised of various APDs. A SiPM is externally biased so that the voltage on each of the APDs is above its breakdown voltage. Thus each APD operates in Geiger mode. The difference between the biasing voltage and the breakdown voltage is known as over-voltage — the main adjustable parameter controlling operation of the device. If a SiPM absorbs a photon, the resulting charge carrier (an electron or hole depending on the structure of the device) can trigger an avalanche in the gain region. The avalanche can produce 10^5-10^6 carriers; this constitutes the gain. The gain of a SiPM depends linearly on the over-voltage and the value of junction capacitance. Even though a SiPM is a pixel device, it is not an image sensor. It is an analog device producing a time-varying output signal that is measured in real time.

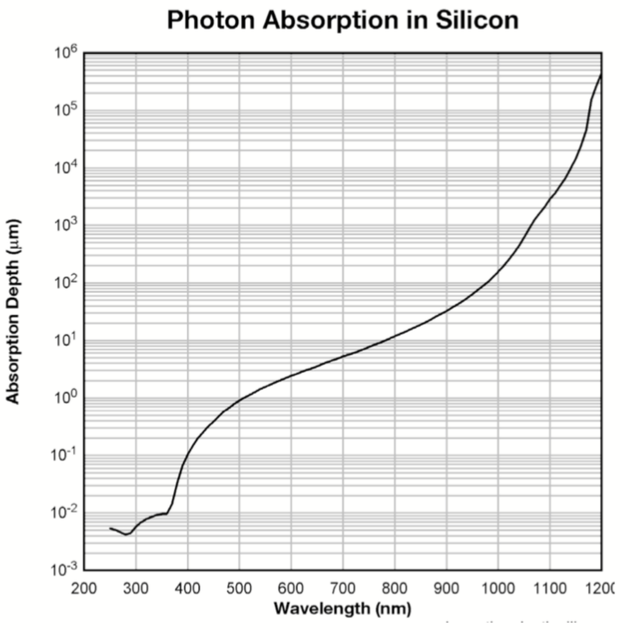


Figure 1: Photon Absorption vs Wavelength in Silicon

An APD is constructed by forming a p-n junction within the Silicon. The p-n junction creates a depletion region where mobile charge carriers are non-existent. The device's goal is to create an avalanche of electrons when an incoming low energy photon struck the silicon structure. Since electrons have a higher probability of causing an avalanche compared to the holes in the electron-hole pair, it is more efficient to construct a p-n junction instead of a n-p junction. Having the photon interact with the p region allows for the most optimal amount of electrons flowing through the depletion region (or the avalanche region).

Every individual APD is connected through a *quenching resistor* and they act independently of all other APDs inside the SiPM. Each APD and quenching resistor constitutes a microcell inside the SiPM which follows a specific cycle that causes the avalanches which we measure. Application of a reverse bias above the breakdown voltage across the p-n junctions generates an electric field across the depletion region. It is this electric field that causes the electron-hole pair to be accelerated thus causing the avalanche. The quenching resistors lower the reverse bias voltage back down under the breakdown voltage and that will cause the avalanche to stop. The halt of the avalanche stops the microcells from detecting new photons while it recharges to the original bias voltage. Once returning to the bias voltage, the microcell can re-detect incoming photons and the cycle, graphically shown in Fig 2, starts again. A very important thing to note is that the avalanche is confined to the microcell that it started in. Every other microcell remains fully operational and able to detect further incoming photons. This guarantee that the detector is active at all time even after firing.

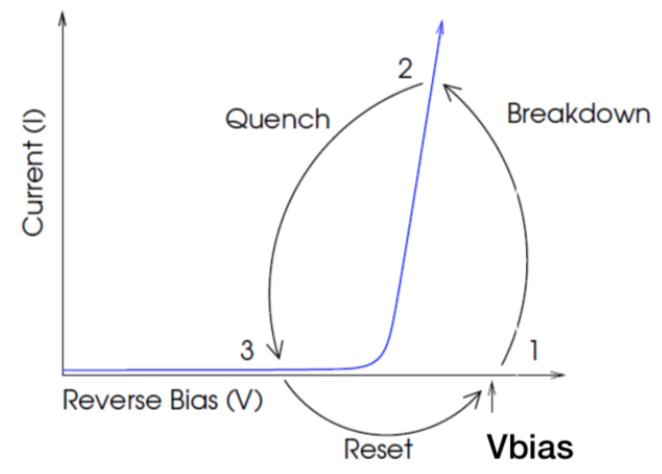


Figure 2: Microcell cycle in the individual APD in a SiPM.

There is an inherent time for the APD's microcell to return to the operating bias voltage after it is quenched, and that is called the recovery time. This recovery time is important when discussing the fill factor of the SiPM. Fill factor is defined as the percentage of the SiPM sensor's surface area that is sensitive to light. Every microcell needs to be separate of the other microcells inside the SiPM for it to function properly. However, the ratio of the separation between microcells and the amount of space taken up by the quenching resistors remains roughly constant. This means that larger microcells correspond to a larger fill factor, higher photon detection efficiency (PDE), and gain. Larger microcells, however, lead to longer recovery times and a lower dynamic range.

1.2 Characteristics of SiPMs and APDs

The performance of SiPMs are dependent upon a set of 6 parameters:

- the breakdown voltage and over-voltage,
- the gain,
- the PDE,
- the dark count rate (DCR),
- the optical crosstalk (OXT),
- the after-pulsing.

We will analyze each of those parameters in the next sub-sections.

Breakdown voltage and over-voltage

The breakdown voltage (V_{br}) is defined as the voltage where the **E**-field across the depletion region is strong enough to cause an avalanche. It is typical for a SiPM to operate at a voltage 10% - 25% higher than V_{br}. The over-voltage (ΔV) is then the difference between V_{br} and the voltage where the SiPM operates. These two quantities allow for the calculation of the bias voltage (V_{bias}) using the simple formula, $V_{bias} = V_{br} + \Delta V$.

Gain

The gain is defined as the amount of charge created for each detected photon. Gain is a function of both the over-voltage and the size of the microcells inside the SiPM. The created charge in the microcells is extremely uniform and quantized. A common way to calculate the gain is by using the capacitance of the microcell (C), ΔV, and the elementary charge, e,

$$G = \frac{C \cdot \Delta V}{e}$$

PDE

The PDE measures the sensitivity of the SiPM and is a function of the wavelength of the incident photon, ΔV, and the fill factor. Another equivalent way of thinking about PDE is that it is the probability that an incoming photon will produce an avalanche inside the APD. PDE is mathematically calculated using the formula, $PDE(\lambda, V) = \eta(\lambda) \cdot \varepsilon(V) \cdot F$, where $\eta(\lambda)$ is the quantum efficiency of Silicon, $\varepsilon(V)$ is the probability of initiating an avalanche, and F is the fill factor.

Dark Count Rate - DCR

DCR is the main source of noise in all SiPMs and is measured in kHz, as a pulse rate, or kHz/mm^2, as pulse rate per unit area. It is, mainly, due to thermally generated electrons inside the microcells and is a function of active area, ΔV, and temperature. An issue with DCR is that the signals produced from thermally generated electrons are identical to electrons that are photon generated. But this can be alleviated to an extent by raising the threshold to a voltage above the thermally generated electrons voltage. This will greatly reduce the amount of noise, but not fully eliminate, that comes from these unimportant electrons. DCR increases with bias voltage so there has to be a trade-off between DCR and PDE that needs to be balanced.

Optical Cross-Talk - OXT

DCR is not the only form of signal noise in SiPMs, OXT plays an important role in the noise level of SiPMs. A function of ΔV, OXT is also highly dependent upon the fill factor. During an avalanche, a charge carrier can emit a photon which can enter another microcell and cause its own avalanche. The probability of this event occurring is known as OXT and it happens instantaneously causing a single photon peak to appear as a peak caused by multiple photons. The OXT can be calculated by taking the ratio of the DCR are two different voltage levels. Similarly to DCR, OXT increases as the PDE increases so the ΔV of the SiPM becomes even more important.

After-pulsing

In Fig. 2, during the breakdown portion of the cycle, a charge carrier can become trapped inside the microcell due to defects in the Silicon. After a very short period of time, typically a few ns, the trapped carrier is released and causes an avalanche in the same microcell that the original carrier passed through. This event is what is known as after-pulsing. After-pulsing becomes important if the release time from being trapped is equal to or longer than the recovery time of the microcell. If the after-pulsing rate is high, it can negatively impact the measurements made from the SiPM. Just like DCR and OXT, the probability of after-pulsing occurring increases and the ΔV increases.

When compared to regular photo-multiplier tubes (PMTs), SiPM tubes have high gain ($\sim 10^6$) and high PDE ($> 50\%$) and their are cheaper and easier to produce. The SiPM are very good for low background application, amount of low background radioactivity in the SiPM material can be more easily controlled than in PMTs where the glass usually generate most of the low background gammas. SiPMs are very compact, rugged, and are magnetically insensitive - a big downside to regular PMTs. The high gain of the SiPM is achieved at a rather low bias voltage and the noise is almost entirely at the level of a single photon. In essence, SiPMs combine the capabilities of PMTs with the benefits found in solid-state sensors which make them popular in a variety of applications from medical imaging to biophotonics to LiDAR.

2 CAEN SiPM Kit Experience

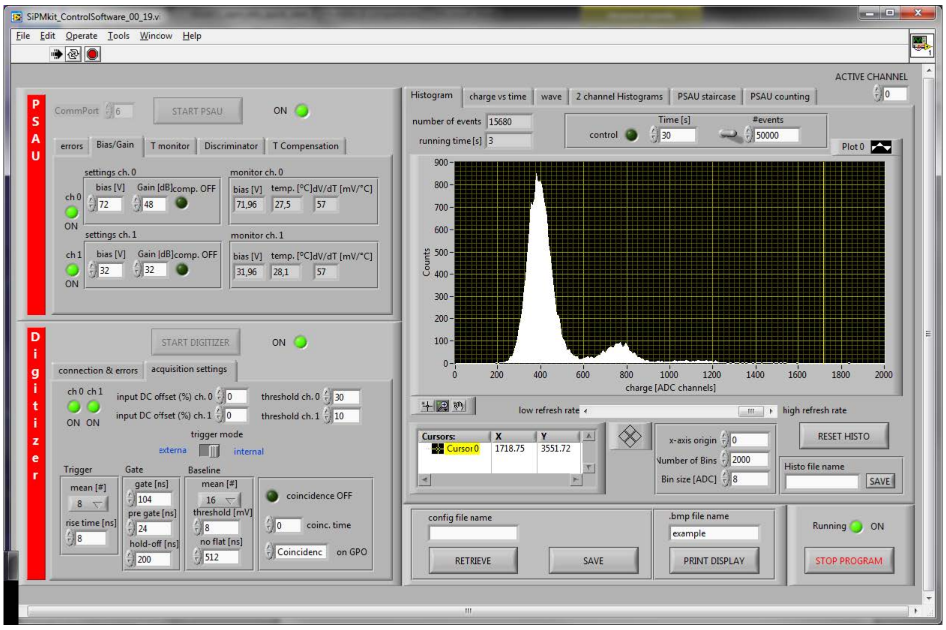
2.1 CAEN software

The GUI main panel is structured as a "virtual instrument": its appearance and operations imitate a real central system (Fig. 3). Three sub-panels may be identified:

- Upper-left panel: tabs refer to the Power Supply and Amplification Unit, PSAU SP5600;
- Bottom-left panel: tabs address the Digitizer DT5720;This is done by setting the display persistence to infinite as before and measuring the position corresponding to the first and second pulse. The difference between the two pulses will be the peak-peak voltage.In order to measure the peak-peak voltage use the oscilloscope menu of measurements and display the horizontal measuring lines. Adjust the two horizontal lines to the position of the peaks. In Fig. 19 the peaks are respectively represented by the orange and yellow pulses. The difference in mV between the position of the two horizontal lines will be the value of the peak-peak voltage.Once the value of the peak-peak is determined adjust the oscilloscope to trigger on half the the value of the first line as determined previously, the V_{pp}, and measure the frequency, this will correspond to the 0.5 Dark Count Rate (DCR0.5) The 1.5 Dark Count Rate (DCR1.5) can be found by triggering the oscilloscope on 3/2 the V_{pp}.The optical cross talk at a given gain can be calculated as follows

  $$OXT = \frac{DCR_{1.5}}{DCR_{0.5}}$$

  Modifying the SiPM gain will change both the Dark Count Rates and the Optical Cross Talk. This can be used to determine the noise level corresponding to a particular gain
- Right-hand side panel: visualization of the measurements and data storage.



CommPort 6 ON

settings ch. 0: ch 0 ON; bias [V] 72; Gain [dB] 48; comp. OFF
monitor ch. 0: bias [V] 71,96; temp. [°C] 27,5; dV/dT [mV/°C] 57
settings ch. 1: ch 1 ON; bias [V] 32; Gain [dB] 32; comp. OFF
monitor ch. 1: bias [V] 31,96; temp. [°C] 28,1; dV/dT [mV/°C] 57

Digitizer ON
ch 0 ON, ch 1 ON
input DC offset (%) ch. 0: 0
input DC offset (%) ch. 1: 0
threshold ch. 0: 30
threshold ch. 1: 10
trigger mode: externa / internal
Trigger: mean [#] 8; rise time [ns] 8
Gate: gate [ns] 104; pre gate [ns] 24; hold-off [ns] 200
Baseline: mean [#] 16; threshold [mV] 8; no flat [ns] 512
coincidence OFF; coinc. time 0; Coincidenc on GPO

ACTIVE CHANNEL 0
number of events 15680
running time [s] 3
control; Time [s] 30; #events 50000

Cursors:	X	Y
Cursor 0	1718.75	3551.72

x-axis origin 0
Number of Bins 2000
Bin size [ADC] 8
Histo file name
config file name
.bmp file name: example
Running ON

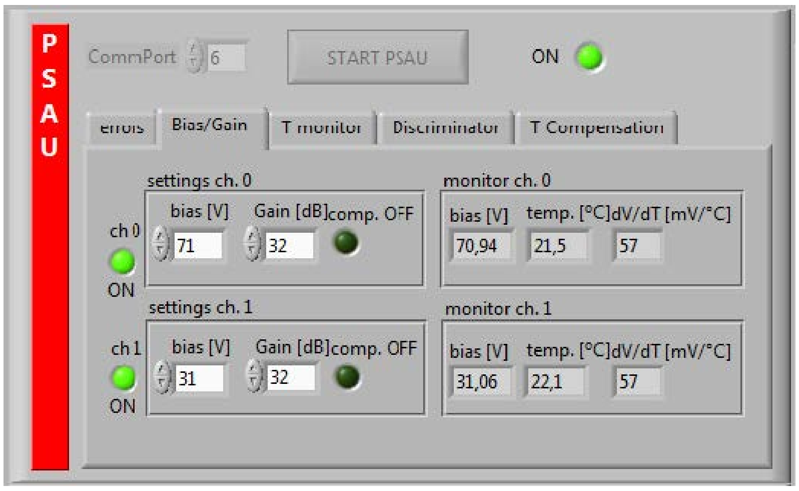
Figure 3: CAEN GUI main panel

The bottom-right side of the GUI allows the user to retrieve or save a configuration file with all the set parameters for the digitizer and the PSAU, to save the current visualized data on the visualization tabs and the plots in the active tab as image files.

2.1.1 The PSAU panel

The PSAU tabs allow the user to set and monitor all the PSAU parameters. They become active after the communication with the PSAU is started, through the “START PSAU” button. This button opens the communication with the PSAU by the selected COMM Port.

The “Bias/Gain” tab (Fig. 4) provides the switchers for the two channels enabling the settings of the bias, the gain and the temperature compensation (Fig. 5).



CommPort 6
ON

settings ch. 0: ch 0 ON; bias [V] 71; Gain [dB] 32; comp. OFF
monitor ch. 0: bias [V] 70,94; temp. [°C] 21,5; dV/dT [mV/°C] 57

settings ch. 1: ch 1 ON; bias [V] 31; Gain [dB] 32; comp. OFF
monitor ch. 1: bias [V] 31,06; temp. [°C] 22,1; dV/dT [mV/°C] 57

Figure 4: PSAU Bias/Gain main panel

The “T monitor” tab (Fig. 5) shows the temperature of the two SiPM.

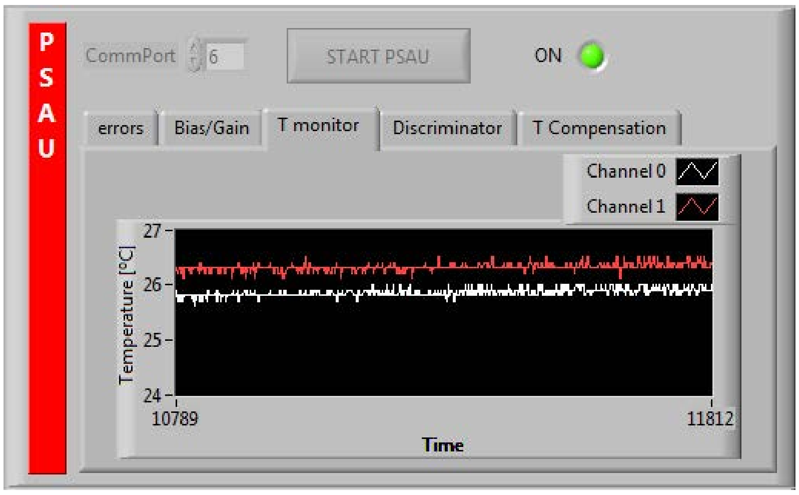
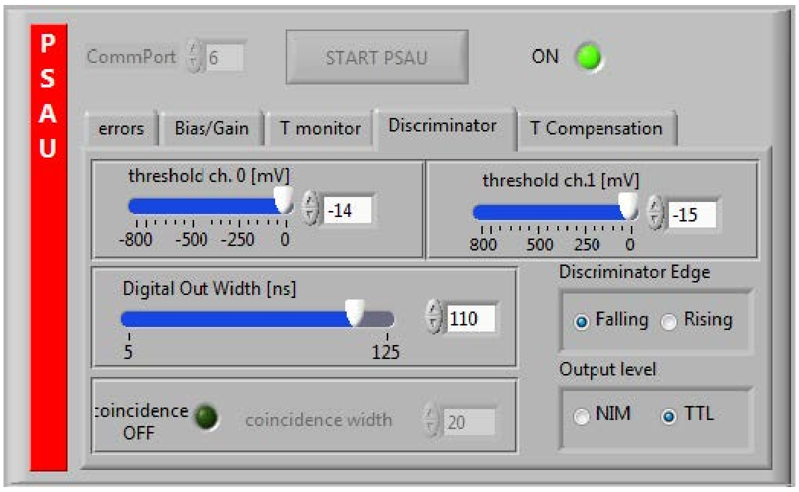


Figure 5: PSAU T monitor panel

The “Discriminator” tab (Fig. 6) allows the settings of the threshold of the discriminators and the width of both signals produced as outputs. The output level can be set as NIM or TTL. The coincidence is active if the two PSAU channels are switched on. The coincidence signal is provided on the digital output 0 (DOUT 0).

The “T compensation” tab (Fig. 7) allows the setting of the coefficient dV/dT for both the channels. The compensation acts on the bias of the sensor to keep its gain constant, according to a linear VT dependence.

The “Errors” tab (Fig. 8) contains the PSAU firmware release and the error code of the library which the PSAU stands on.



CommPort 6

threshold ch. 0 [mV] -14

threshold ch.1 [mV] -15

Digital Out Width [ns] 110

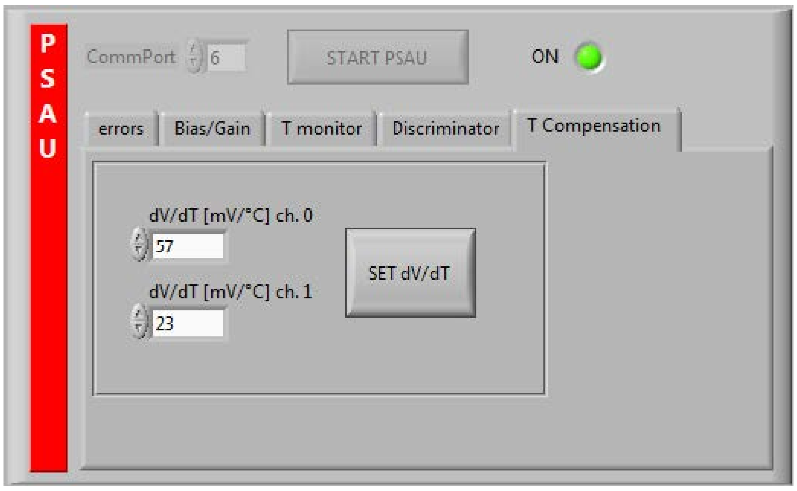
Discriminator Edge: [✓ Falling] Rising

Output level: NIM [✓ TTL]

coincidence OFF

coincidence width 20

Figure 6: PSAU discriminator panel



CommPort 6

dV/dT [mV/°C] ch. 0 57

dV/dT [mV/°C] ch. 1 23

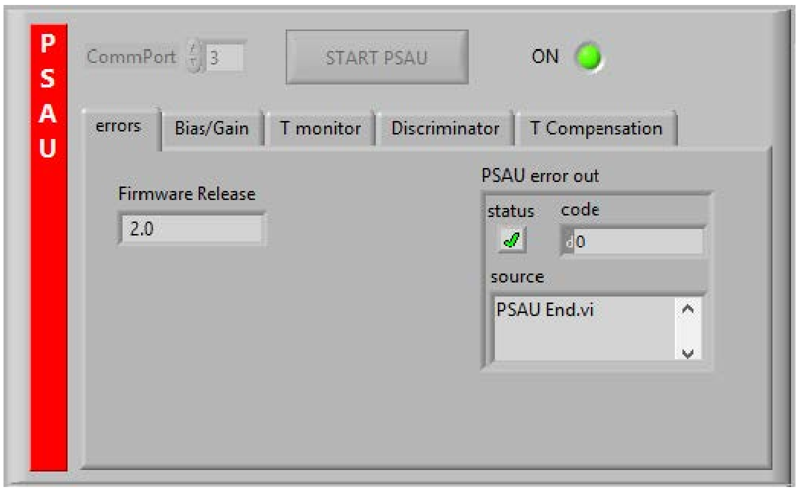
Figure 7: PSAU T compensation panel

2.1.2 The Digitizer panel

In the digitizer tab (Fig. 9) is possible to set all the parameters requested by the digitizer: the active channel, the input DC offset, the channel threshold, the trigger mode, the trigger, the gate and the baseline parameters and the settings for the coincidence. The tab include also information about connection and errors.

Acquisition settings sub tab details:

- START DIGITIZER: the digitizer tab is inactive till the digitizer is on. Pushing the "START DIGITIZER" button the software create a link with the physical device. After the green light the parameters tab become active.
- CHANNEL ON/OFF: the channels involved in the data readout of the digitizer. When one channel is active, the correspondent "input DC offset" and "threshold" controls are enabled. - Please note that the channel to be analyzed is selected also in the upper right corner of the digitizer window - Under the ACTIVE Channel tab.
- INPUT DC OFFSET: it is a percentage shift of the input range scale (=2 Vpp), allowing the dynamic range to be shifted from −2.0/0 V up to 0/2.0 V. -50% is its minimum value and it corresponds to

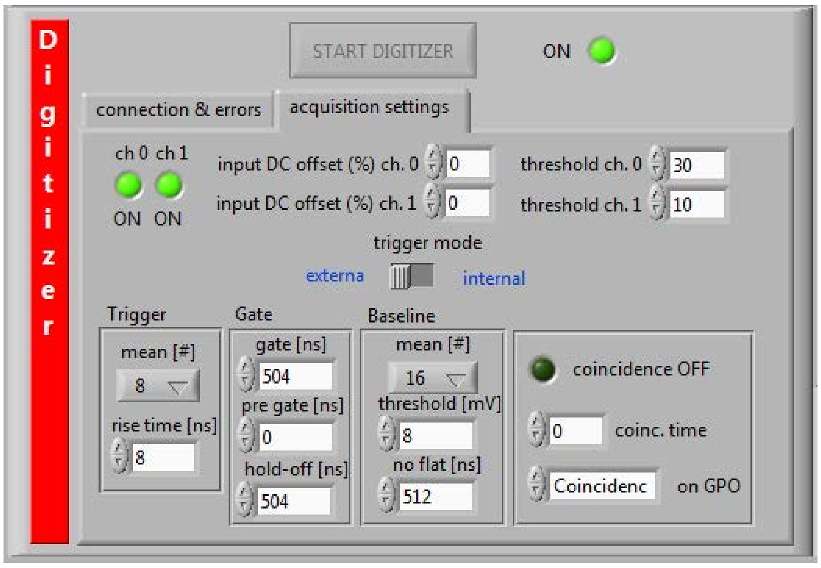


CommPort 3

Firmware Release 2.0

PSAU error out
status ✓ code 0
source PSAU End.vi

Figure 8: PSAU Errors panel



ch 0 ON, ch 1 ON
input DC offset (%) ch. 0: 0
input DC offset (%) ch. 1: 0
threshold ch. 0: 30
threshold ch. 1: 10
trigger mode: externa / internal

Trigger: mean [#] 8; rise time [ns] 8
Gate: gate [ns] 504; pre gate [ns] 0; hold-off [ns] 504
Baseline: mean [#] 16; threshold [mV] 8; no flat [ns] 512
coincidence OFF; coinc. time 0; Coincidenc on GPO

Figure 9: Digitizer panel

$-2.0/0$ V dynamic range. 0% corresponds to a $-1.0/+1.0$ V dynamic range. The control is enabled when the correspondent channel is active.

- CHANNEL THRESHOLD: it represents the threshold over delta, allowing the detection of the pulses, in auto-trigger mode. The delta is the difference between the current sample, i.e. the input signal sampled at time t, and the average of a samples digitized "rise time" ns before t. The control is enabled when the correspondent channel is active.

- TRIGGER MODE: if "internal trigger mode" is selected, the digitizer is able to self-detect the signals, according to the trigger parameters. If "external trigger mode" is selected, the digitizer wait a trigger signal on the LEMO "TRG IN" connector on the front panel.

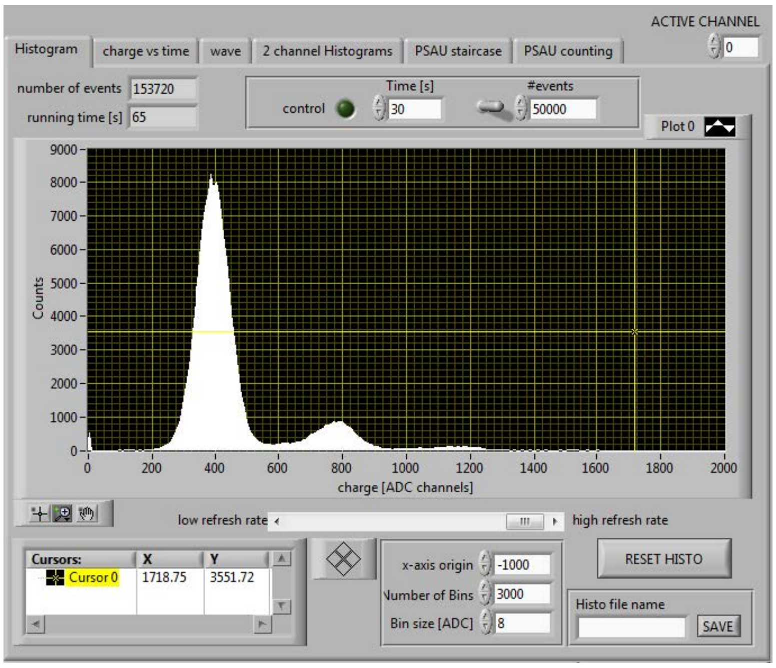
- TRIGGER PARAMETERS: Before the calculation of the delta, the input signal is filtered in order to reduce the high frequency noise, using a low pass filter that averages a certain number of samples within a moving window. "mean" represents the number of double sampling periods used by the average window; allowed values for the parameter are 1, 2, 4, 8, 16 and 32. "RISE TIME" is the rise time of the input signal, used in the calculation of the signal delta.

- GATE PARAMETERS: the "gate" represents the width of the gate signals, the "pre gate" is the advance between the gate generation and the trigger leading edge, while the "hold-off" is a veto for the generation of other gates.
- BASELINE PARAMETERS: the "threshold" represents the value on delta, over that the baseline calculation is frozen. The "mean" parameter is the number of samples for the average calculation of the baseline. 0 disables the baseline restoration. "no flat" is the veto for the calculation of baseline.
- COINCIDENCE PARAMETERS: the coincidence can be selected if both the channels are switched on. "Coincidence time" represents the width of the discriminator signal of each channel. Two signals are in coincidence if all of them exceed their own threshold during this time width. "on GPO" allows to chose the output on the digitizer front panel GPO between: Coincidence, Gate and Discriminator.
- CONNECTION HANDLE: once the device is opened, the function returns a handle that becomes the unique identifier of that device; any access operation to the device will take place according to its handle, thus making transparent the physical channel.
- ROC & AMC FIRMWARE RELEASE: these fields contain the current firmware release running on the mainboard (i.e. on the ROC FPGA) and on the mezzanine (i.e. on the AMC PFGA).
- ERROR OUT: any error given back by the CAEN Digitizer library which the program stands on, is reported in the field code.

2.2 The visualization tab

The visualization tabs allow the user to manage and visualize the signals of the detectors. Please note that the active channel tab on the right upper corner should be set to the corresponding channel of the digitizer to analyze.

- The "Histogram", "charge vs time", "wave" and "2 channel Histograms" tabs refer to the digitizer. The other two, "PSAU Staircase" and "PSAU counting", refer to the PSAU. The "Histogram" tab shows the histogram of the active channel according to the PSAU and digitizer settings. The user can change the refresh rate in the meaning of the access to the buffer of the digitizer: high refresh rate means high access rate to the digitizer and low number of integral signals transferred; low refresh rate means low access rate to the digitizer, but a big amount of data transferred for each access. The properties of the X scale of the histogram can be selected by the user: the origin of the histogram (in the meaning of the minimum plotted charge value), the number of bins (which determine the end of the plotting window) and the bin size. These values determine the range of the histogram that will be stored pushing the "SAVE" button. The prefix of the histogram output file saved can be written by the user in the "Histo file name" field.
- The graph palette of the histogram allows the user to change the visualization of the spectrum, i.e. enlarging the histogram, zooming it, etc.
- The "charge vs time" tab plots the charge versus time. The user can zoom in and zoom out the plot, and change the number of charges for the plotted mean. The plot can be stored pushing the "Save_ChargevsTime" button.
- The "wave" tab shows the trace of the analog and digital signals read from the digitizer. The analog signals are the trace of the input and the virtual probe, which, in internal trigger mode, can be the delta or the baseline signal. If external trigger mode is selected, the virtual probe automatically switches on baseline only. The digital signals are the Gate (red), the Time Over Threshold (yellow), the Hold off time (blu), the Flat signal i.e. the signal that visualize the baseline calculation (violet):
  - The gate, flat and over threshold signals represent the digitizer selected parameters;



ACTIVE CHANNEL 0

number of events 153720

running time [s] 65

control Time [s] 30 #events 50000

Cursors:	X	Y
Cursor 0	1718.75	3551.72

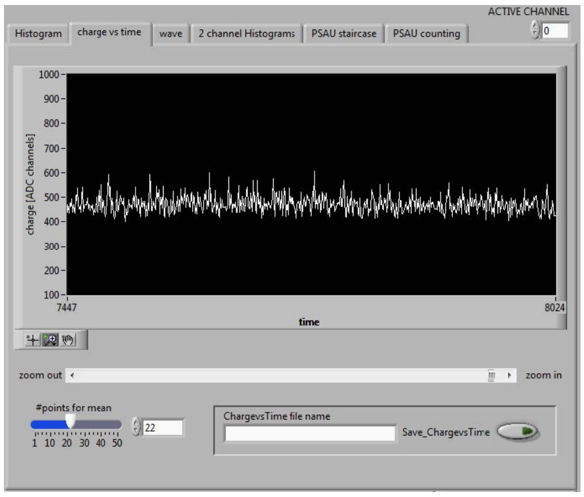
x-axis origin -1000

Number of Bins 3000

Bin size [ADC] 8

Histo file name

Figure 10: visualization panel - Histogram Tab



ACTIVE CHANNEL 0

#points for mean 22

ChargevsTime file name

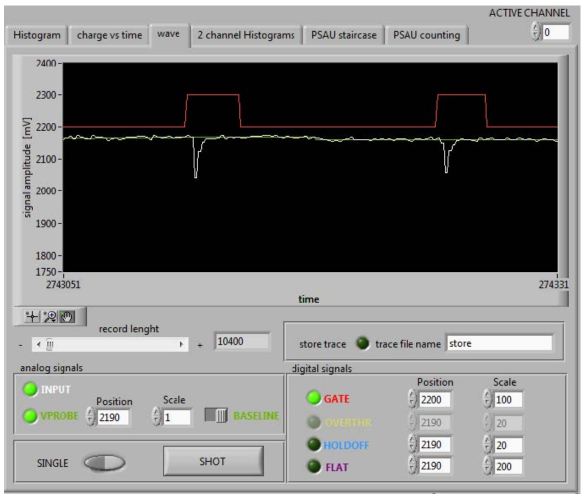
Figure 11: visualization panel - charge vs time Tab

  – The over threshold is generated when the signal is over the set threshold;

All the trace can be amplified with the Scale control and moved in vertical position with the Position control. The switch on the bottom-left side of the tab changes the plotting mode from a continuous stream of data to single shot. In single shot mode the update of the plot stops, and the user has to push “SHOT” button for visualize another triggered signal.

- The “2 channel Histograms” tab allows for managing the histogram plots from the two channels of the digitizer simultaneously. In the three sub tabs it is possible to plot, reset and save the two histograms, the histogram sum and the correlation.

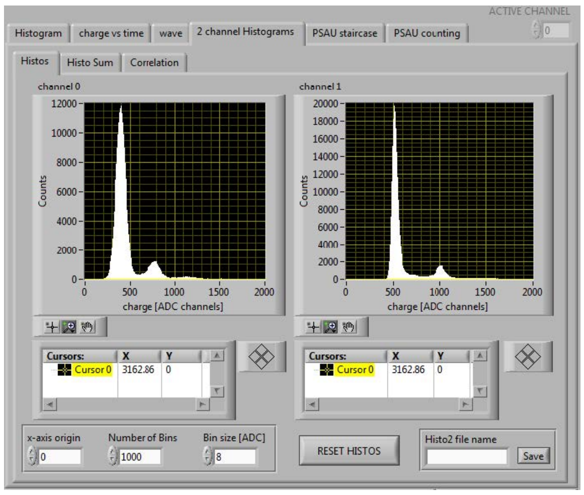
  The “Histos” sub tab contains two histogram plots, where it is possible to set the x-axis origin, the number of bins and the bin size in a dedicated menu common to both the channels, the reset histograms



ACTIVE CHANNEL 0
record lenght 10400
store trace, trace file name: store
VPROBE Position 2190, Scale 1

	Position	Scale
GATE	2200	100
OVERTHR	2190	20
HOLDOFF	2190	20
FLAT	2190	200

Figure 12: visualization panel - wave Tab



ACTIVE CHANNEL 0

Cursors:	X	Y
Cursor 0	3162.86	0

Cursors:	X	Y
Cursor 0	3162.86	0

x-axis origin 0
Number of Bins 1000
Bin size [ADC] 8
Histo2 file name

Figure 13: visualization panel - 2 channel histogram Tab

button and the saving menu (output file prefix, common to both the files, and the SAVE button to perform the saving).

- The “Histo Sum” sub tab contains the plot of the histogram sum. This histogram results from adding channel0’s histogram to channel1’s histogram multiplied by an alpha factor. The parameter alpha is configurable by 0.001/step. Histogram reset can be performed by the “RESET HISTO” button and the plot can be saved by the “Save” button with a user-defined prefix written in the “HistoSum file name” field.

- The “Correlation” sub tab shows a scatter plot of the signals from the 2 sensors, after being integrated in the specified time window. It may be of help for specific applications relying on a simultaneous use of the 2 detectors, e.g. when using the scintillator tiles for cosmic ray experiments or 2 spectrometry heads for ^{22}Na positron annihilation detection.

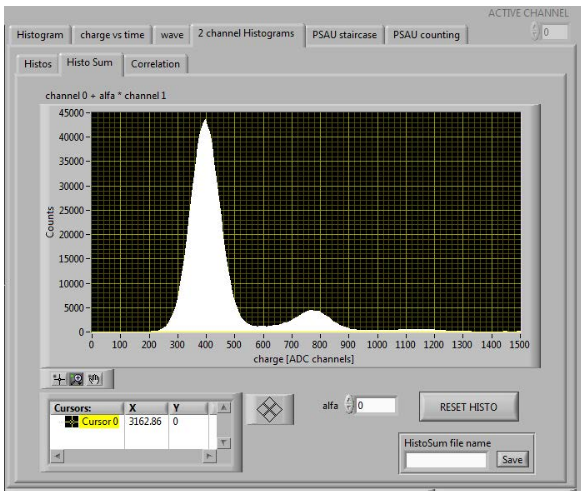


Figure 14: visualization panel - Histo Sum sub-Tab

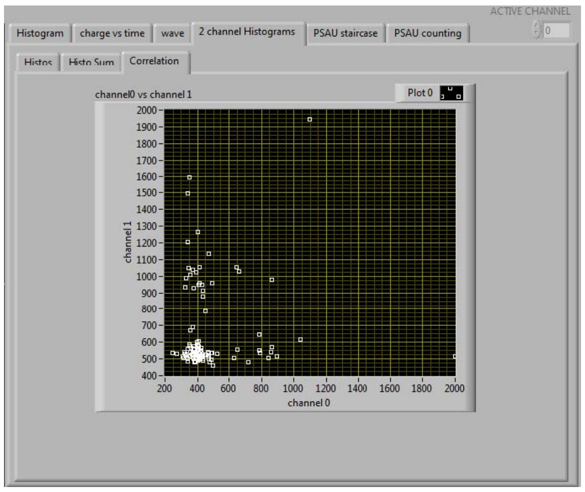
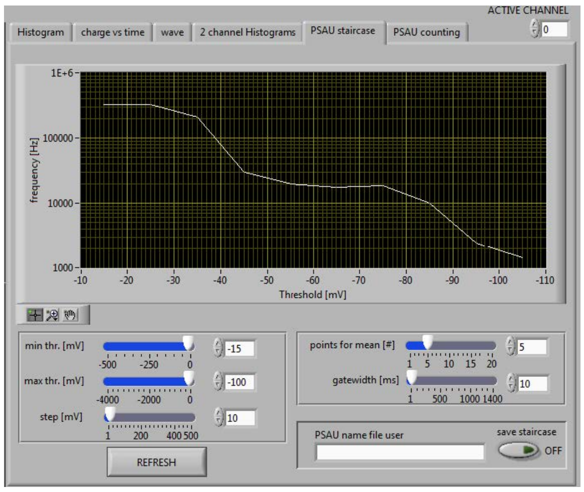


Figure 15: visualization panel - Correlation Tab

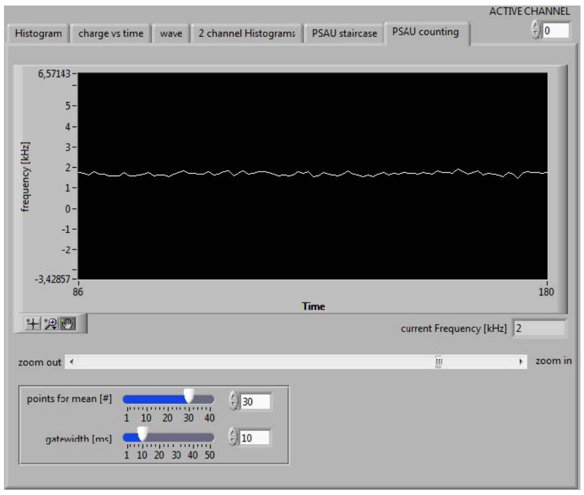
- The “PSAU staircase” tab allows the interaction with the PSAU in order to produce the so-called SiPM staircase: the plot shows the frequency of the signals which are over the threshold, during a threshold scan from the “min thr” value up to “max thr.” value. The user can change these limits, the step, the number of read point which produce the mean plotted value and the gate width for the counting.



ACTIVE CHANNEL 0
min thr. [mV] -15
max thr. [mV] -100
step [mV] 10
points for mean [#] 5
gatewidth [ms] 10
save staircase OFF

Figure 16: visualization panel - PSAU staircase Tab

- The "PSAU counting" tab plots the frequency of the signals over the threshold set in the "Discriminator" tab for the active channel. The user can change the number of points for the plotted mean value and the gate width for the counting.



ACTIVE CHANNEL 0
current Frequency [kHz] 2
points for mean [#] 30
gatewidth [ms] 10

Figure 17: visualization panel - PSAU Counting Tab

3 Basic Measurements

Overview

The purpose of this experiment is to understand the use and characterization of a SiPM detector (Silicon Photo multiplier Detector).

The dark count, Peak-Peak value, and optical cross talk will be evaluated as a function of the detector gain. Next, an led will be used to generate a multi-photon peak spectrum using a CAEN 12 bit, 250 MS/s digitizer.

A digitizer allows for the conversion of analog waves signal to a digital readout. As the waveform is collected by the digitizer, analog-to-digital converters create a digitized wave which is stored in a buffer until a computer is ready to process the wave. From the digitized signal in ADC the program will generate and display a histogram with number of counts as function of ADC values.

3.1 Measuring Dark Count Rate

3.1.1 Configuration

1. Within the GUI top portion of the main panel that controls the PSAU ensure that the correct Comm-Port number is selected. This value should correspond to the Port number which the PSAU is registered to the computer. The CommPort setup box looks like the one shown below in Fig. 18.

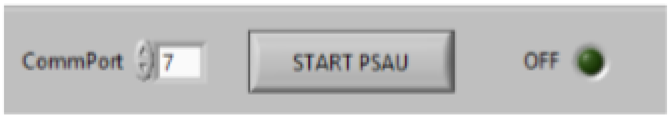


Figure 18: CommPort box on the accompanying software

2. In case of issues: Open device manager, locate the "Ports" drop down, and ensure that the CommPort number on the accompanying software matches the SP5600 USB to Serial Converter port number. Refer to the Fig. 19. The correct CommPort number will only appear after turning on the PSAU. If different, simply type in the correct number into the setup box shown in Fig. 18 on the left and press enter. The message 'CHECK CONNECTIONS' will appear to the right of the 'START PSAU' button shown above if the CommPort numbers do not match.

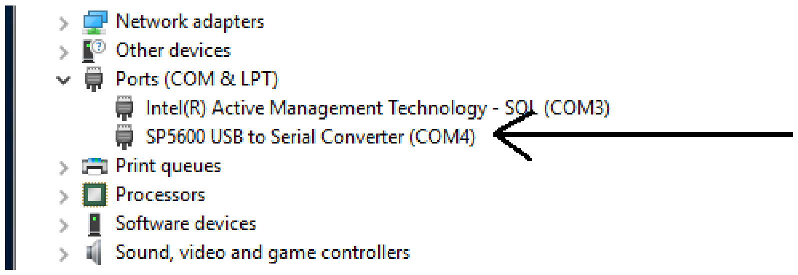

Network adapters
Other devices
Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
SP5600 USB to Serial Converter (COM4)
Print queues
Processors
Software devices
Sound, video and game controllers

Figure 19: Port number in device manager (4)

3.1.2 SiPM Spectrum and Dark Count Rate

1. Begin by ensuring that the power adaptor is connected to the digitizer, LED driver, and PSAU. **This adaptor should be plugged into the isolation transformer along with the power to the oscilloscope.**

2. Check the following connections between the digitizer, LED driver, PSAU, and the oscilloscope.

   (a) There is a T-connector attached to Channel 1 of the oscilloscope whose ends are connected to the front of the digitizer and the back of the PSAU. The channels where the cables are connected should be the same as the one the SiPM is attached to on the front of the PSAU.

   (b) There is a T-connector attached to Channel 2 of the oscilloscope whose ends are connected to the 'Trig In' of the digitizer and 'Trig Out' of the LED driver.

   (c) The fiber optic cable is securely connected to both the SiPM on the PSAU and the front of the LED driver.

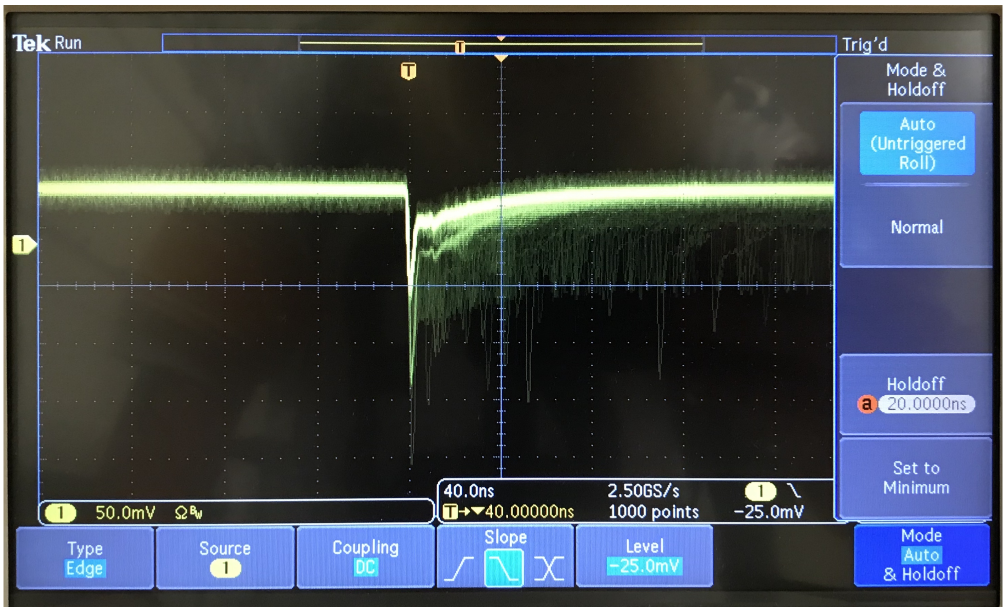
3. To function optimally, the SiPM needs to be properly biased and the gain must be set to prevent saturating the PSAU amplifier. Please check the label for the SiPM serial number to choose the proper bias voltage. There is also a sticker on each PSAU for cross referencing.

4. If not previously done, turn on both the digitizer and the PSAU. Press the 'START PSAU' button, set the correct bias voltage for the channel the SiPM is connected to on the PSAU and press the button for that channel to turn it on. Then set the gain to 40dB.

	SiPM 1 (Serial Number 15665)	SiPM 2 (Serial Number 13585)
Bias	55.64 V	54.99 V
Gain	40 dB	40 dB

5. Next hit the 'START DIGITIZER' button, turn on the correct channel, set it to internal trigger, and change the channel threshold to 30.
6. Finally, ensure that the 'active channel' in the top right corner of the software display matches the channel that the SiPM is attached to on the PSAU. Consult Fig. 3 for location.
7. On the oscilloscope:
   (a) Within the Channel 1 menu, ensure the following settings:
      i. Termination resistance - 50Ω
      ii. Coupling - DC
      iii. Invert - Off
      iv. Bandwidth - 250MHz
   (b) Have the following settings on the trigger menu:
      i. Trigger - Edge
      ii. Trigger Channel - Channel 1
      iii. Coupling - DC
      iv. Slope - Falling
      v. Mode Auto & Hold off - 20ns
   (c) Change the trigger level until the cleanest separation between the pulses is achieved.

   If done correctly, one should get a similar figure to Fig. 20 as shown below.



Holdoff 20.0000ns
1 50.0mV
40.0ns 2.50GS/s
T→40.00000ns 1000 points
1 -25.0mV
Type Edge
Source 1
Coupling DC
Level -25.0mV
Mode Auto & Holdoff

Figure 20: Example of the SiPM output signal at a gain of 40 dB for a sensor with no LED light. Horizontal scale is set to 40 ns (trigger menu - horizontal scale and vertical scale is set to 50 mV (use channel menu to set it).

3.1.3 Calculating the peak-to-peak voltage at a given gain

1. Locate the *cursor* button and press and hold to open up the menu.
2. Within the cursor menu, make sure that the cursors are set to *screen* and not *waveform* and that they are not linked together.
3. Adjust both of the multipurpose knobs until the cursors are at the location of the two different peaks in the graph. A box will appear on the screen that shows ΔV which is the peak-to-peak voltage (V_{pp}) needed below.

   Calculating V_{pp} is important for a variety of reasons. Primarily, it is a fair estimation of the overall system gain. It is also useful in calculating the DCR and the optical cross talk (OXT). The system gain is in turn useful for:

   (a) Measuring the gain dependence of the over-voltage with respect to the breakdown voltage
   (b) Setting amplification factors and avoiding over saturation effects
   (c) Setting the discrimination threshold which is used in counting experiments

   The OXT at a given gain can be calculated using the following equation:

   $$OXT = \frac{DCR_{1.5}}{DCR_{0.5}}$$

   Modifying the SiPM gain will change both the DCR and the OXT. This can be used to determine the noise level corresponding to that gain.

4. $DCR_{0.5}$ is found by changing the trigger level on the oscilloscope to half of the measured V_{pp} value from step 3 and remeasuring the ΔV value. $DCR_{1.5}$ is similarly found by changing the trigger level to 3/2 the V_{pp} value instead of 1/2 the value.

3.1.4 Data Collection

1. Calculate the OXT at different gains ranging from 30 dB to 50 dB.
2. Using the collected data, plot both the OXT and DCR as functions of gain. Make note of what value of the gain gives the least amount of OXT.

3.2 SiPM and LED; integration and triggering

Using the LED driver we can acquire a Multi-Photon spectrum that will help us characterizing the SiPM gain, noise, photon number resolving capability, DCR, and cross talk. It also allows us to characterize the statistics of the emitted photons from the LED driver itself.

3.2.1 Amplification and Intensity Tuning

Using the **LED Driver, PSAU, and Oscilloscope**:

1. Ensure the setup is the same as in Section 4.1.2
2. Have the channels settings for channels 1 and 2 the same
3. Turn on the LED Driver and set it to INTERNAL trigger and low frequency. This is done by toggling the switches on the back of the driver.
4. On the front of the LED driver, locate the dial that changes the pulse light intensity. Set this to be around 5.0. This will cause the pulse frequency to be roughly 10 kHz. The range of the low frequency is adjustable within the range of 500 Hz to 80 kHz.

Fig. 21 shows an example of what the oscilloscope should show if done properly.

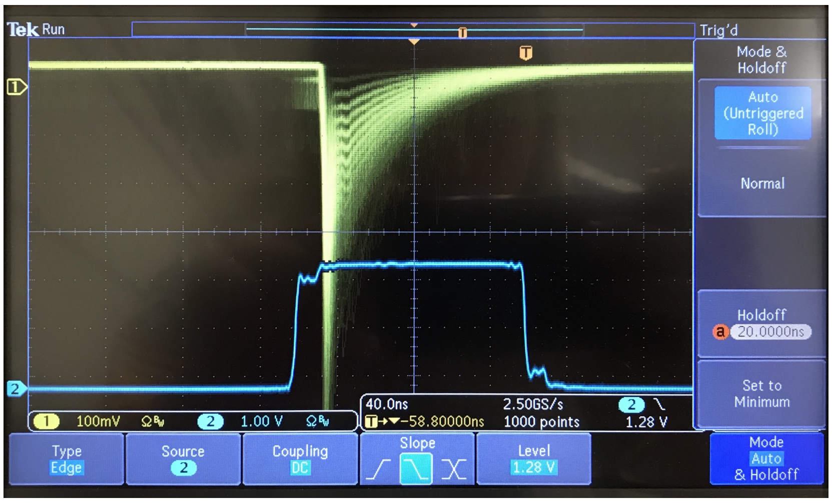


Figure 21: A multi-photon peak spectrum with triggering on the LED driver output signal on channel 2. Channel 1 is set to 100 mV scale, channel 2 is set to 1 V scale and the horizontal scale is 40 ns. Trigger threshold is set to 1.28V on channel 2 in the example here.

After achieving a graph similar to that of Fig. 21, change the value of the LED intensity both above and below the previous value.

Doing so will allow you to explore the dynamic range of the response of the SiPM. The peak height will change horizontally as the intensity is varied. Eventually, one will find the saturation region of the SiPM. The saturation region is defined as the regime where variation in the LED intensity change the peak width rather than the peak height. An example of the saturation region is shown below in Fig. 22.

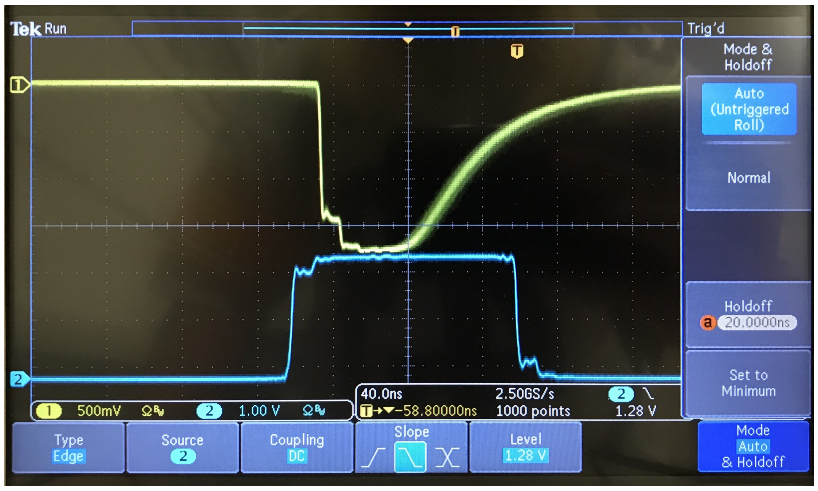
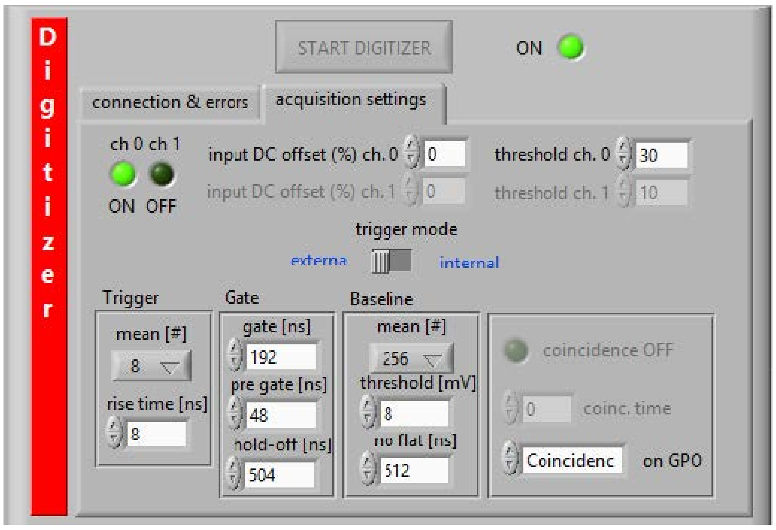


Figure 22: Example of the saturation region of the SiPM. Channel 1 is set to 500 mV scale, channel 2 is set to 1 V scale and the horizontal scale is 40 ns. Trigger threshold is set to 1.28V on channel 2 in the example here.

3.2.2 Signal Digitization

Setup the Digitizer as followed:

1. Select EXTERNAL trigger mode.
2. Select the active channel (either 0 or 1 depending on which channel the digitizer is connected to).
3. Accept the default values for Trigger, Gate, and Baseline sub-panels. They should be set as the following (see Fig. 23):
   (a) Trigger: Mean=8, rise time=8 ns
   (b) Gate: gate = 192 ns, pre-gate=48 ns, hold-off=504 ns
   (c) Baseline: mean=256, threshold=8 mV, no-flat=512 ns


Digitizer

START DIGITIZER ON

connection & errors | acquisition settings

ch 0 ch 1
ON OFF

input DC offset (%) ch. 0: 0
input DC offset (%) ch. 1: 0
threshold ch. 0: 30
threshold ch. 1: 10

trigger mode: externa / internal

Trigger
mean [#]: 8
rise time [ns]: 8

Gate
gate [ns]: 192
pre gate [ns]: 48
hold-off [ns]: 504

Baseline
mean [#]: 256
threshold [mV]: 8
no flat [ns]: 512

coincidence OFF
0 coinc. time
Coincidenc on GPO

Figure 23: Example of waveform digitizer panel settings

4. Switch to the Wave panel and the following components should be visible: **digitized analog input, signal baseline,** and **integration gate**.

   The gate is what properly defines the integration time and the edge is triggered in a variety of ways. The opened gate is defined by three parameters: the **gate**[**ns**]**, pre-gate** and the **hold-off**. Modification of these parameters gives a better integration which in turns leads to a better set of data that is collected.

   - The **gate**[**ns**] represents the width of the gate signals.
   - The **pre-gate** defines the position of the gate with respect to the trigger edge.
   - The **hold-off** is the user defined veto following the gate opening.

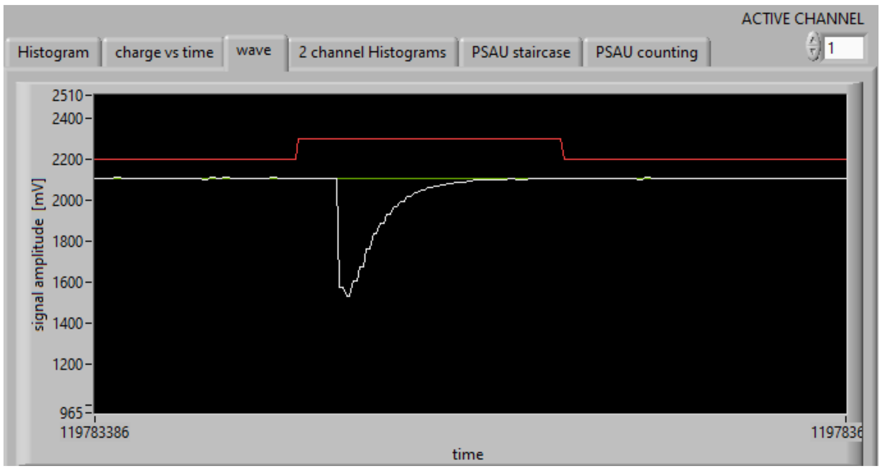


Figure 24: Digitized Wave output corresponding to appropriate settings.

5. Adjust the **pre-gate** and **gate** values So that they encapsulate the entire triggered wave. An example of what a proper wave form and integration gate should look like is show above in Fig. 24. The system will be ready to record and integrate the spectrum and can be viewed in the Histogram tab of the GUI.

6. Switch to the Histogram tab. It is possible to run the data acquisition based on time or number of events. The longer the run time and/or the higher number of events selected will lead to better data acquisition and therefore better analysis. Fig. 25 shown below is an example set of data run for an appropriate amount of time.

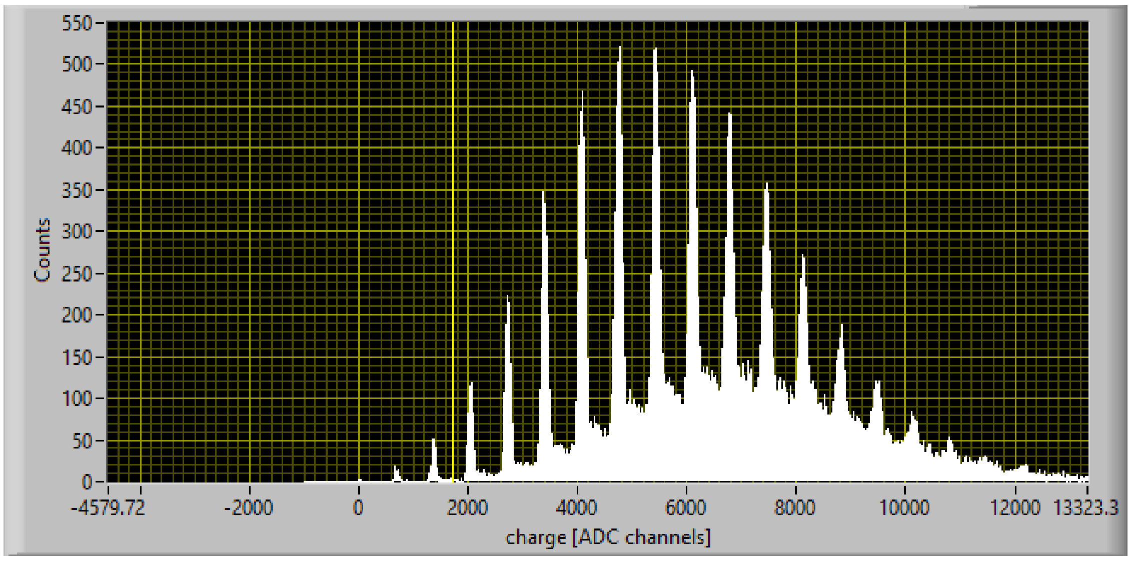


Figure 25: Multi-photon peak spectrum at different LED intensities

3.2.3 Data Acquisition

After recording the Multi-Photon Peak spectrum, several characteristics can be measured and calculated:

1. The SiPM multiplication factor can be measured from the peak-peak distance, using the fact that the system has a charge of 40 fC/ADC, and that the signal is amplified by a factor set by the user.
2. The Gain of the SiPM can be calculated as follows

$$Gain = \frac{\Delta PP(ADC_{ch})^{*} \times ADC_{c.r.}}{e}$$

   where $ADC_{c.r.}$ is the ADC conversion factor and is equal to $.4192 \times 10^{-15}$ at $25°C$, $\Delta PP(ADCch)^{*}$ is the horizontal distance between peaks of the histogram in Fig. 25, and e is the elementary charge.
3. A multi-peak photon spectrum fit for the histogram can and should be performed. This can be done in any coding program such as Python, C++, Java, etc. From this fit, you can:
   (a) Find the width of the Gaussian peak. It will have a trend of the form $\sqrt{\sigma_0^2 + \sigma_1^2 \times n^2}$[i]
   (b) If accounted for, measure the DCR and OXT from the fitted data.
   (c) Determine any statistics about the emitted photons which are usually of a Poissionian form.
4. The photon number resolving power can be attained at a glance.

To save data from Histogram, choose the desired file name and click the save button before the histogram is collecting data. Once the save button has been selected, run a controlled data collection (either time or event based). After the sample has been collected the data file can be found under the data file within the CAEN program files.

[i] σ_0 denotes the system noise and σ_1 denotes the cell-to-cell variation

Analysis

After the multi-photon peak spectrum has been imported into the data analysis program of your choice, in this case Python, various different forms of analysis can be conducted. Figure 26 shows a scatter-plot of the peak spectrum data.

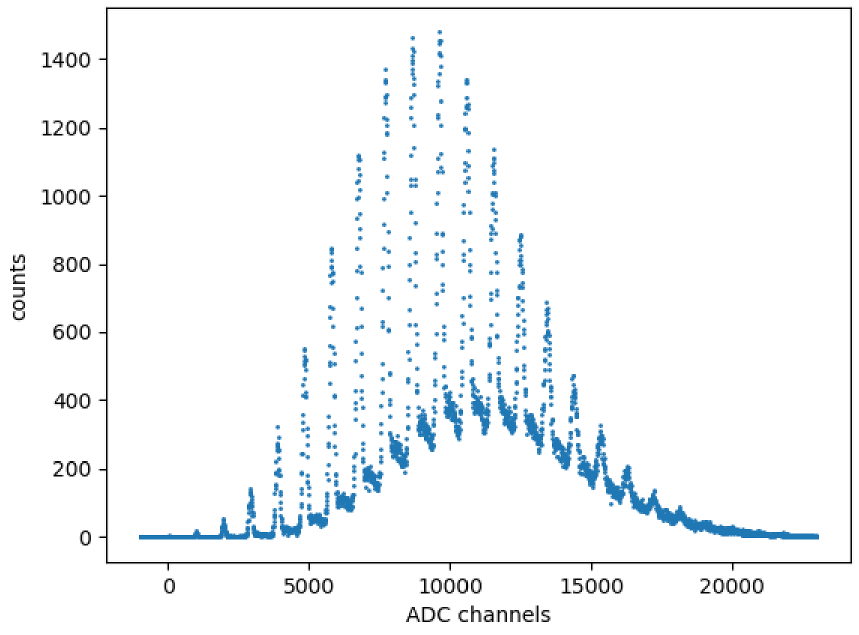


Figure 26: Scatter plot of spectrum

Using some of the analytical libraries that Python has to offer, it is possible to find all of the peaks within the data set. This is particularly important for calculating the peak-to-peak distance used in calculating the gain. Fig. 27 shows a graph of the peaks in the spectra from Fig. 26.

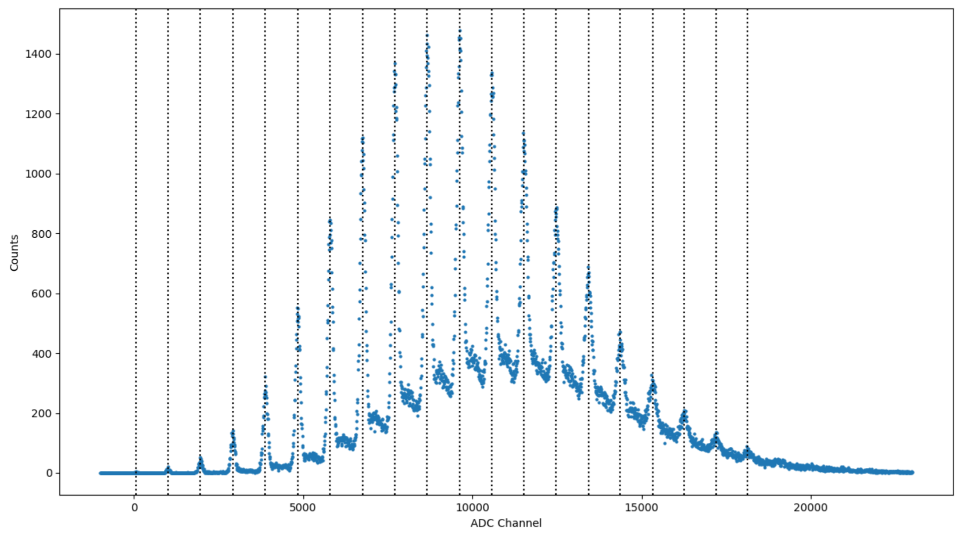


Figure 27: Peaks of the spectrum

Unfortunately, peaks from noise are possible. An example of this can be seen in Fig. 27 at the left most dashed line above 0 on the horizontal axis. Using these peaks, it is possible to plot the entire data set as a sum of individual Gaussian functions. It is also possible to plot each individual peak to its own Gaussian, which are shown in Fig. 28 and, partially, in Fig. 29 respectively.

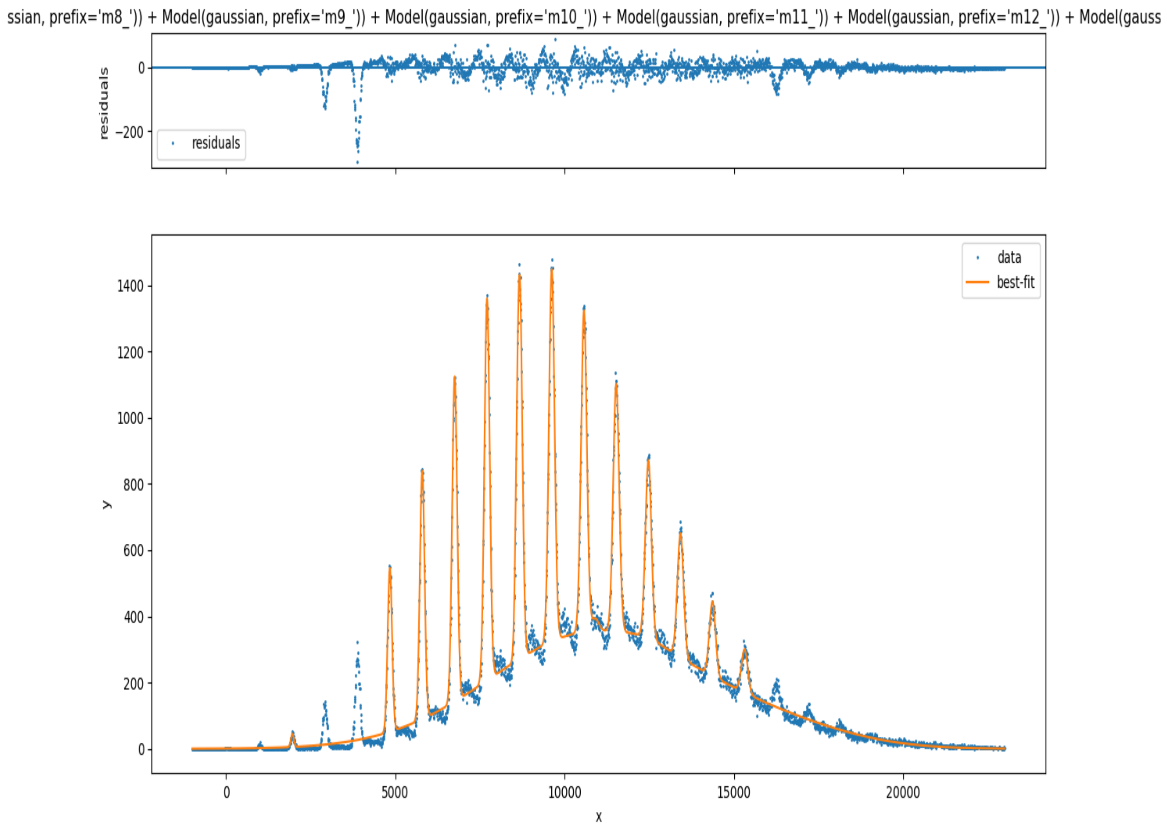


Figure 28: Curve fit for the data

The coded sum of the Gaussian functions is unable to plot the full data set perfectly as can be seen above in Fig. 28. The residual, or error, in each of the data set peaks against the Gaussian plotted peaks is shown in the top plot of Fig. 28

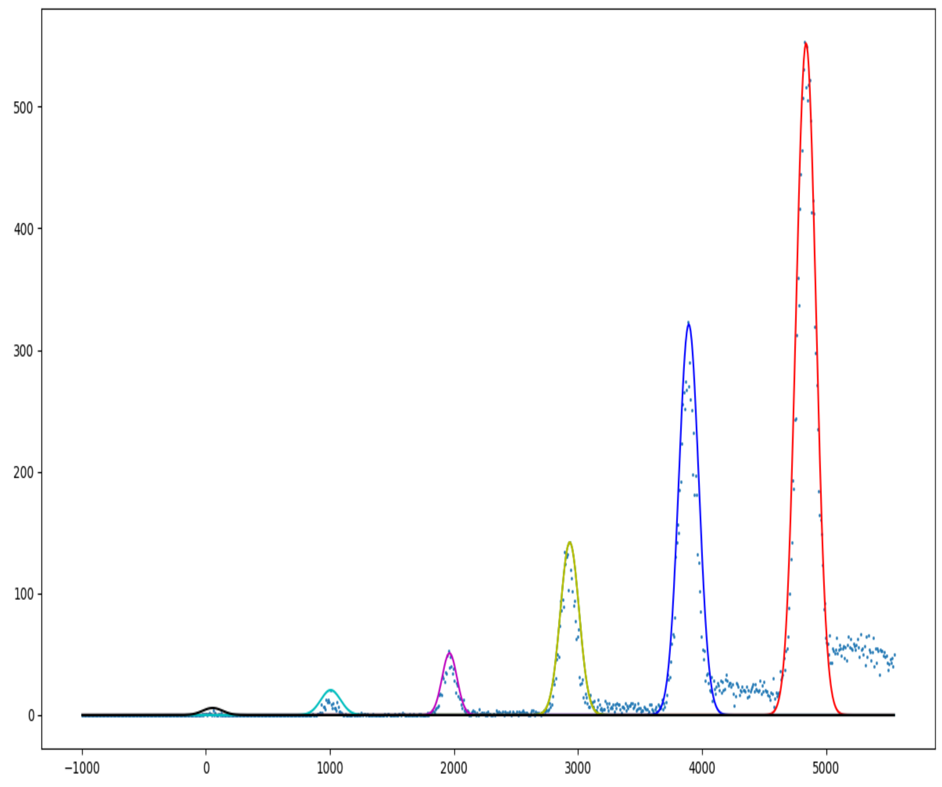


Figure 29: Graph of Gaussians against the original Data

Fig. 29 above shows the first 6 peaks in the full set of data plotted as individual Gaussian functions. This can be translated to any other portion of the full data set. This is of importance because it illustrates that these peaks are fit near perfectly by Gaussian functions. Thereby justifying our use of Gaussian functions in modeling the data set.

3.3 Checking the gamma spectrum

3.3.1 First configuration

For the first configuration, locate the **PSAU, Digitizer, Spectrometer,** and **the Oscilloscope**. Ensure that the PSAU and Digitizer are connected to the desktop and that the Spectrometer is connected to the PSAU in either channel (Ch 0 or Ch 1).

1. After powering everything on, select the correct CommPort number if this has not been done previously.
2. In the PSAU panel, set the bias to the value reported on the sensor ID card. The gain can be set to a high value, i.e. 40 dB, without worry.
3. The temperature variations of the SiPM can be disabled for clarity sake
4. Before taking a γ source spectrum, the DCR must be quantified as a counting frequency. Make the threshold for the frequency 0.5 of a single photo-electron peak ($DCR_{0.5}$).
5. Navigate to the "PSAU Staircase" tab to find the discriminator threshold. This is necessary because the system can become blind to the radioactive source due to a high DCR.
6. Run the PSAU staircase twice, once with a source and once without it. Using these two staircase, it is possible to accurately determine the cut-off or discriminator threshold.

3.3.2 Second configuration

For the second configuration, locate, in addition to what was required for the first configuration, the **splitter** and **a radioactive source**. Before beginning, open up the spectrometer and choose a scintillating crystal to use. Spread an even coat of the optical grease on the open face (non polished) face of the chosen scintillating crystal. Place the greased side of the crystal into the spectrometer and seal it back up.

1. The first thing to do is to properly bias the SiPM. This could be redundant if the same crystal was used as in the first configuration. Leave the gain as it was in the first configuration setup.
2. Navigate to the digitizer panel
   (a) Select external trigger mode
   (b) Select the active channel. This should be the same as it was in the first configuration
   (c) Leave the default values for the Gate and Baseline sub-panels
3. We must now find the "correct" external threshold value.
   (a) Keep the digitizer set to external trigger mode.
   (b) Run a PSAU staircase from -5mV to -110mV. A sharp drop should be noted which stabilizes when the DCR drops to 0.
   (c) Take note of the threshold voltage a safe distance after said sharp drop.
4. The threshold voltage can also be found if the digitizer is set to internal trigger mode. The trigger parameters must be set by hand this way. Set both the mean and rice time to 8. However, it is highly recommended to leave the digitizer on external trigger.

3.4 Can you see the beta particle?

3.4.1 Configuration

- Before beginning, make sure that the scintillating tile inside the SP5608 is removed before conducting the experiment.
- Connect the cables in the given format, as stated in the SP5600AN Educational Kit manual on p. 47.
- First thing to do is to make sure that the detector is properly biased after start up. At this time, also set the gain of the PSAU amplifier. (In our case, we use a bias of 55.6V on the detector, and a gain of 32dB on the PSAU.
- The SiPM gain for temperature variations is turned off, for clarity.
- it is possible, by putting the oscilloscope to signal output before, to distinguish optical cross-talk in the detector.
- Before starting any measurements with a β source, make sure to conduct a DCR measurement to provide a baseline threshold for the detector. (To take into account the spurious avalanches due to thermally generated carriers). It is standard procedure to quantify the DCR with a threshold corresponding to 0.5 x single photo-electron peak ($DCR_{0.5}$). The DCR of the mini spectrometer SiPM is between 2-6 MHz.
- Since we don't want the system to be blind to the radioactive source due to the high DCR, a proper cut-off threshold must be selected. The DCR vs discriminator threshold can be precisely measured by using the "PSAU Staircase" tab in the Control Software.
- Once the SiPM is biased, and a convenient gain chosen, the threshold scan for the signals frequency can be performed.
- This graph should look like an inverse logarithmic graph, or a decreasing straight line on a logarithmic graph scale, as shown in the figure below.

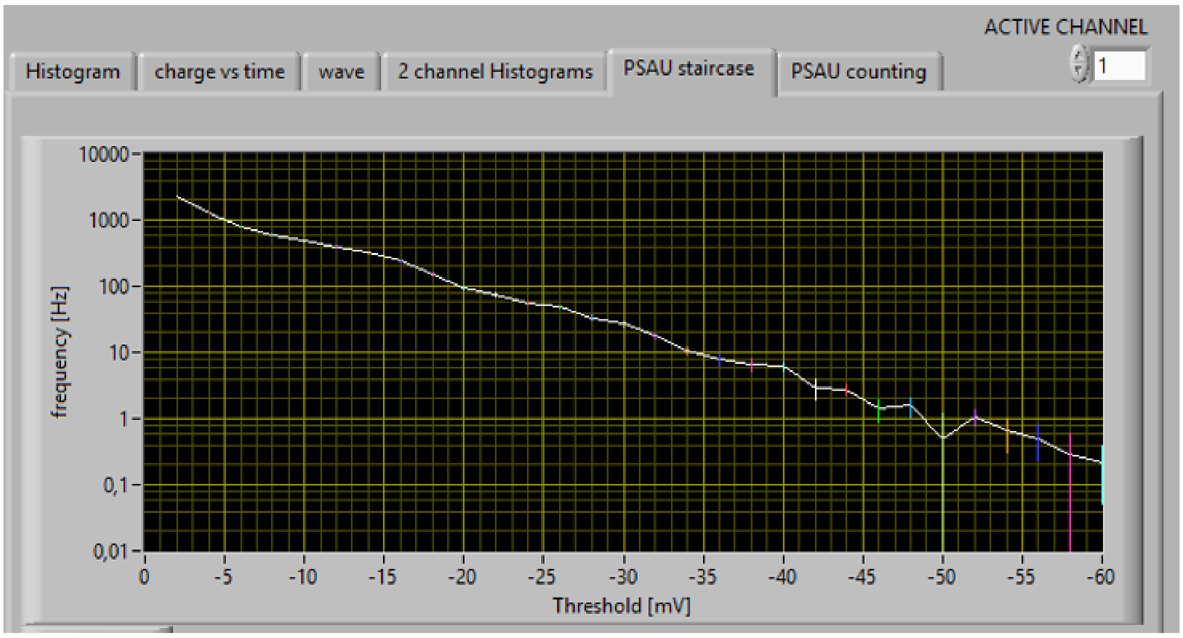


Figure 30: The DCR as a function of discriminator threshold. Notice it's logarithmic nature and scale

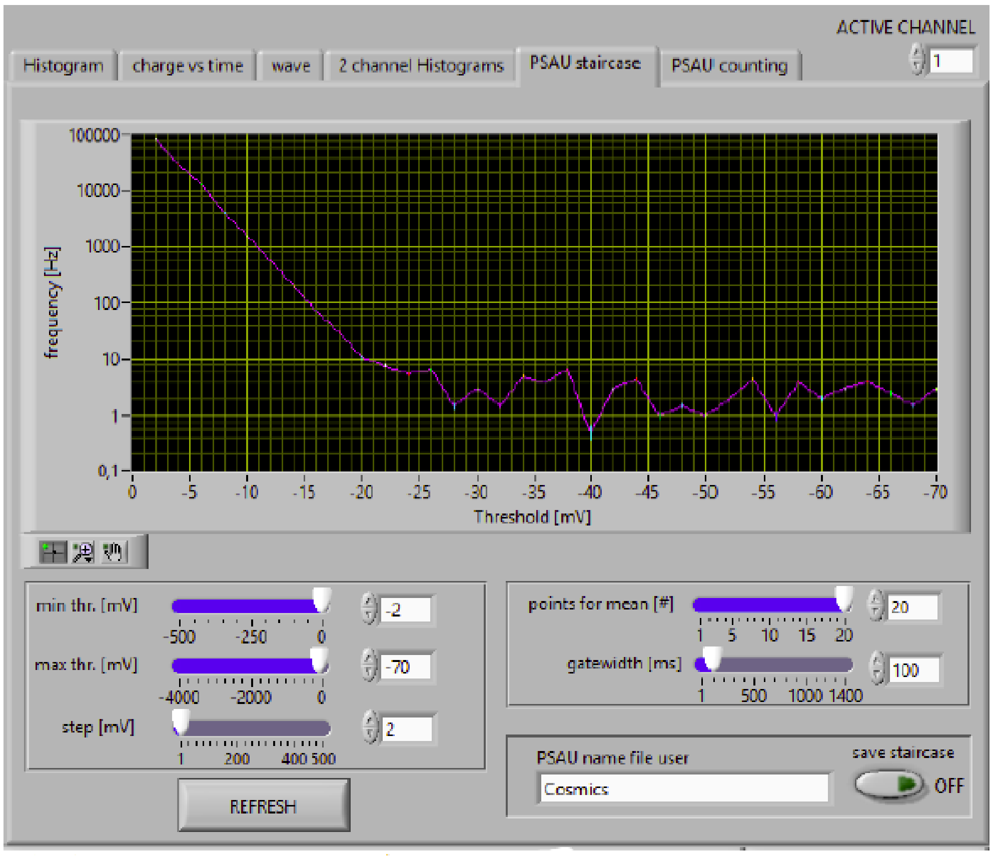
- Once you hit this part of the Setup, remove the second kit, and in the same manner as above, open up the SP5608 and spread optical grease on the sensor. Then Insert the scintillating inside the housing and make sure it matches up perfectly with the sensor. Then put the source holder on top of the tile, and insert the radioactive source inside the holder. Then close up the SP5608 and start the application.
- configure all cabling as noted in the SP5600AN Educational Kit p. 50.
- By using the "PSAU Staircase" tab in the accompanying software, we can observe the beta source contribution to the frequency.
- Apply a threshold scan for the signal frequencies via the "PSAU staircase" tab in the right part of the GUI.
- Using the two staircases, DCR and β source, we can choose a correct cut-off threshold to acquire the spectrum. Running the staircase from roughly -2mV to -100mV, the DCR decreases from .1MHz to below Hz level. Depending on the source, we can select an appropriate threshold. The frequency of the source with this threshold will remain constant, while the DCR drops to zero.

3.5 Cosmic Rays

3.5.1 Configuration

Begin by opening the SP5608 and spread optical grease on the sensor. Once thoroughly coated, insert the Scintillating tile and close the SP5608. The kit elements can now be powered on, RUN the application and start the PSAU and Digitizer. Once the system is started activate the PSAU and Digitizer channels.

Under the "PSAU Staircase" tab in the SiPM Control Software the frequency scanning in the discriminator threshold allows for the observation of the cosmic contribution as seen below.



ACTIVE CHANNEL 1

min thr. [mV] -2

max thr. [mV] -70

step [mV] 2

points for mean [#] 20

gatewidth [ms] 100

PSAU name file user: Cosmics

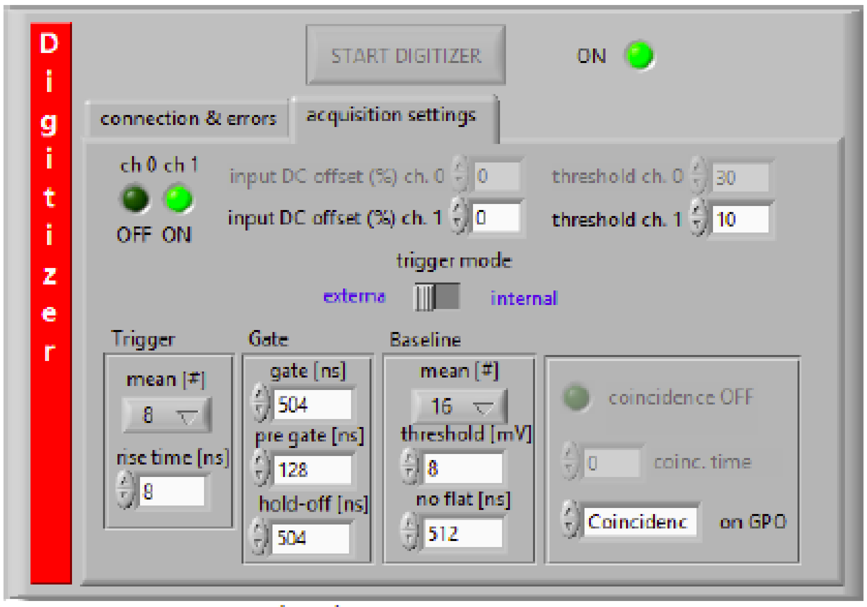
Figure 31: Counting Frequency of Cosmic Rays

The above figure will show the cosmic rays staircase after the SiPM has been properly biased and a the gain set to a convenient value. Using both the Cosmic and DCR bands the right cut-off threshold can be chosen to acquire the spectrum of cosmic rays

3.5.2 Acquiring the Cosmic Spectrum

The Cosmic Spectrum can be triggered on the digitizer externally. For this to happen ensure that the LEMO cable is connected between the PSAU and the trigger input of the digitizer.
Under the "Digitizer Tab" ensure that the Gate and Baseline parameters are tuned according to the signal waveform displayed in the "WAVE" tab. Note that the signal time development is dominated by both the sensor response and the decay time of the scintillating crystal.

Once properly set the system will be able to record the spectrum in the Histogram tab.



ON

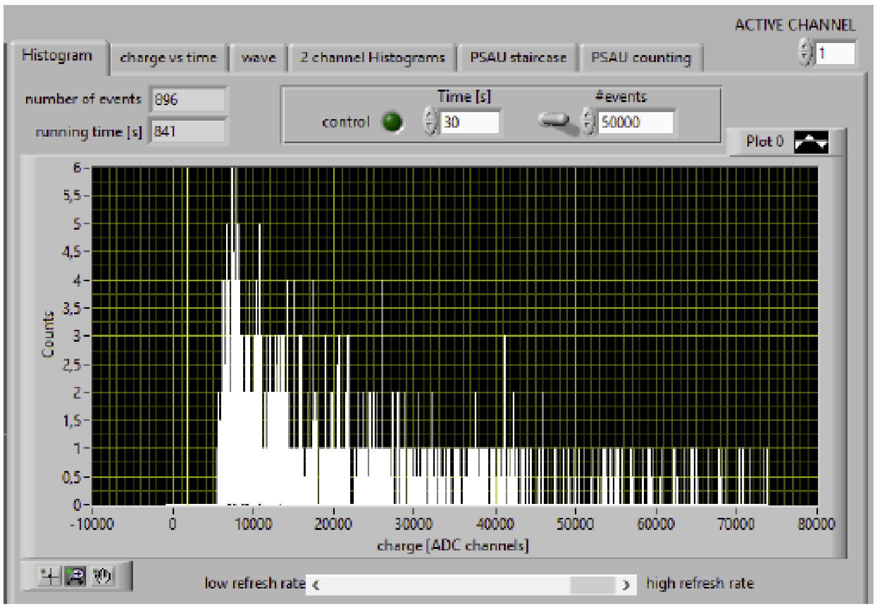
ch 0 ch 1
OFF ON

input DC offset (%) ch. 0: 0
input DC offset (%) ch. 1: 0
threshold ch. 0: 30
threshold ch. 1: 10

Trigger: mean [#] 8; rise time [ns] 8
Gate: gate [ns] 504; pre gate [ns] 128; hold-off [ns] 504
Baseline: mean [#] 16; threshold [mV] 8; no flat [ns] 512

coincidence OFF
coinc. time: 0
Coincidenc on GPO

Figure 32: Caption



ACTIVE CHANNEL: 1
number of events: 896
running time [s]: 841
control
Time [s]: 30
#events: 50000

Figure 33: Observed Cosmic Spectrum

4 Python Code for multi-peak fit

Authors PIDs: anoshi1, mattmc5

Created by - Anosh Irani and Matthew McNeal

Final Edit - 05/04/2019

The code found below is meant to analyze a set of data gained from the SiPM and fit said data with Gaussian curves. It simultaneously finds the gain of the data using a function described in detail below. The basis for this code comes from the article 'Peak fitting XRD data with Python' by Chris Ostrouchov found at the link here: `https://chrisostrouchov.com/post/peak_fit_xrd_python/` .

We begin by importing a variety of Python libraries that will help us in the plotting and analysis of the peaks from the SiPM data. We import in this code: numpy for mathematical calculations, matplotlib.pyplot for any and all plotting necessities, and from scipy we import constants (for gain calculation) and signal where we use a subfunction to find the peaks from our array of numbers.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal, constants

x, y = np.loadtxt("Hist_Data.txt", unpack = True)
```

We unpack the data in the above line and store it for later use

```
X = []
Y = []
counter = 0
while y[counter] <= 700:
    X.append(x[counter])
    Y.append(y[counter])
    counter = counter + 1
X = X[:-60]
Y = Y[:-60]
```

The above 9 lines including the while loop are designed to take the original set of data and find the first few peaks that the data gives. The while loop number can be changed depending on the strength of your data. In the 1st test case, there was a max strength of 1500 so to get 5 peaks we have the while loop going to 700. In a second set of test data, the max peak strength was 500 so the whilecounter was set to 70. The list stripping that takes place in lines 35 and 36 are meant to remove any partial peaks in the smaller set of data.

4.1 FUNCTIONS

This function finds the peaks in the data given. The argument 'model_indicies' needs to be changed to fit any set of data. It can be seen in lines 182 and 183 that it is a list that contains the number of peaks that data appears to have in integer steps starting at 1 going until the number of peaks believed to be in the data.

```
def peak_specs(some_spec, model_indicies, peak_widths=(10, 25), **kwargs):
    x = some_spec['x']
    y = some_spec['y']
    x_range = np.max(x) - np.min(x)
    peak_indicies = signal.find_peaks_cwt(y, peak_widths)
    np.random.shuffle(peak_indicies)
    for peak_indicie, model_indicie in zip(peak_indicies.tolist(),
    model_indicies):
        model = some_spec['model'][model_indicie]
        if model['type'] in ['GaussianModel']:
            params = {'height': y[peak_indicie], 'sigma': x_range
            / len(x) * np.min(peak_widths),
                      'center': x[peak_indicie]}
            if 'params' in model:
                model.update(params)
            else:
                model['params'] = params
        else:
            raise NotImplemented(f'model {basis_func["type"]}
            not implemented yet')
    return peak_indicies
```

This function is used to reject outliners from the data. It is used when calculating the gain mostly so we get the most accurate data possible

```
def reject_outliners(data, m=.3):
    return data[abs(data - np.mean(data)) < m * np.std(data)]
```

```
Defines the Gaussian function that is needed in later functions
```

```
def Gaussian(x,height,c_pos,width_std):
    return height * np.exp(-(x-c_pos)**2/(2*width_std**2))
```

This is the function that will actually calculate the gain of the data that we have accumulated. It takes the x-axis location of all of the peaks and sorts them into a new list. It will then reject any points that are outside of a certain range and calculate the gain using the formula that this function is returning. Said function is given in the lab handout

```
def calculate_gain(peaks):
    q = np.sort(peaks)
    p = []
    p.append(q[0])
    deltaList = np.diff(q)
    remOut = reject_outliers(deltaList)
    meanDiff = np.mean(remOut)
    return round(meanDiff * (.492*10**-15)/
    constants.value('elementary charge'),2)
```

This is the function that will actually draw the Gaussian on top of the plots of the data. It requires an argument that is an object from matplotlib.pyplot

```
def Draw_Gaussians(spec, peak_index, plotter):
    centers_x_val = []
    half_height = []
    full_height = []
    fw_at_half = []
    peaks1 = []
    np.sort(peak_index)
    for i in peak_index:
        peaks1.append(i)
    for i in peaks1:
        centers_x_val.append(spec['x'][i])
    for i in peaks1:
        half_height.append(spec['y'][i] * .5)
        full_height.append(spec['y'][i])
    zero_counter = 0

    while zero_counter < len(half_height):
        if half_height[zero_counter] < 2.5:
            half_height.pop(zero_counter)
            full_height.pop(zero_counter)
            centers_x_val.pop(zero_counter)
            peaks1.pop(zero_counter)
        else:
            zero_counter = zero_counter + 1

    count = 0
    for i in peaks1:
        wrk_list = spec['x'][i:]
        y_list = spec['y'][i:]
        counter1 = 0
        for m in y_list:
            if m <=half_height[count]:
                temp = wrk_list[counter1] - wrk_list[0]
                fw_at_half.append(temp)
                break
            counter1 = counter1 + 1
        count = count + 1

    for i in range(len(half_height)):
        plotter.plot(spec['x'], Gaussian(spec['x'], full_height[i],
        centers_x_val[i], fw_at_half[i]))
```

This is where the specs for the model that is used in plotting. If the number of peaks that you have in the 'model' argument of the peak_specs function is greater than the preset 20 and 6, you must add a corresponding type:Gaussian dictionary entry to the 'model' dictionary sub-dictionary to account for this.

```
full_spec = {'x' : x, 'y' : y, 'model' : [
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'},
        {'type': 'GaussianModel'}, {'type': 'GaussianModel'}]}

first_spec = {'x' : X, 'y' : Y, 'model' : [
        {'type' : 'GaussianModel'}, {'type' : 'GaussianModel'},
        {'type' : 'GaussianModel'}, {'type' : 'GaussianModel'},
        {'type' : 'GaussianModel'}, {'type' : 'GaussianModel'}]}
```

4.2 Analysis and plots

This section is where all of the calling of the aforementioned functions takes place. It calls the functions to calculate gain and plot the Gaussian. There is commented out code that will plot with dotted lines where the peaks are in the plot. This is useful if you want to ensure that the code finds all of the peaks it is supposed to, but for now it is commented out. It also sets the labels to the plots.

```
all_peaks = peak_specs(full_spec , [0,1,2,3,4,5,6,7,8,9,10,11,12,13,
14,15,16,17,18,19,20], peak_widths=(15,))
first_peaks = peak_specs(first_spec , [0,1,2,3,4,5], peak_widhts=(15,))

fig_a , ax = plt.subplots()
ax.set_xlabel("ADC channel")
ax.set_ylabel("Counts")
fig_b , bx = plt.subplots()
bx.set_xlabel("ADC channel")
bx.set_ylabel("Counts")

ax.scatter(full_spec['x'], full_spec['y'], s=1)
bx.scatter(first_spec['x'], first_spec['y'], s=1)


print('Begin')
print('Gain: ', calculate_gain(all_peaks))

#all_peaks = np.sort(all_peaks)
#first_peaks = np.sort(first_peaks)

#for peak in all_peaks:
#    ax.axvline(x=full_spec['x'][peak], c = 'black', linestyle='dotted')
#for peak in first_peaks:
#    bx.axvline(x=first_spec['x'][peak], c = 'black', linestyle='dotted')

Draw_Gaussians(full_spec , all_peaks ,ax)
Draw_Gaussians(first_spec , first_peaks ,bx)

plt.show()
print('End')
```