

Task 14: Photon reconstruction and gas gain measurement with a CAST-type GridPix detector

Markus Gruber
December 06, 2019

Abstract

This is an instruction on the process of measuring and reconstructing X-ray photons with a CAST-type GridPix detector. First the detector is briefly introduced continuing with the explanation of the setup and measurement principle.

1 Setup

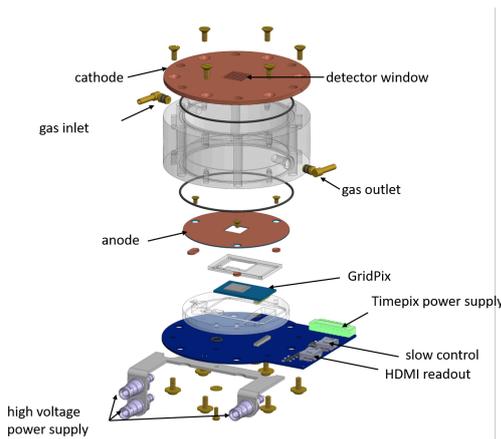


Figure 1: Sketch of the CAST-type GridPix detector.

The detector consists of a cylindrical gas volume with a cathode and an anode with the chip and the readout PCB as end caps. In the center of the cathode there is a window such that soft X-rays can enter the detector volume. The window is based on a $3\ \mu\text{m}$ thin Mylar folie. In the center of the anode there is a quadratic hole such that the anode surrounds the GridPix which is plugged on the readout PCB. This PCB offers all connectors for the detector operation (two times HDMI for the chip slow control and the readout, one low voltage connector for the chip power supply, several high voltage connectors for the grid, the anode, the cathode and a potential field cage).

The two HDMI cables are connected to an adapter card which is connected with a SRS-FEC (Scalabe Readout System - Front End Card). It receives the data from the chip, compresses it by deleting information from pixels without detected charge and sends it via an ethernet cable to the PC.

2 Detector principle

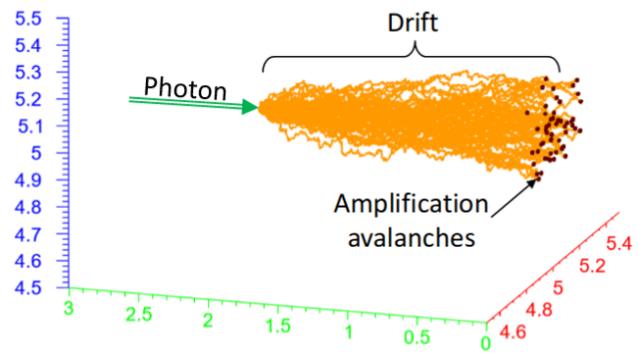


Figure 2: Simulation of a typical photon event in the GridPix detector. The photon leads to ionization of the gas and the electrons drift towards the grid of the GridPix. The electrons drift through the holes of the grid and are amplified by the high electric field between the grid and the readout plane.

If a photon enters the gas volume of the detector through the detector window it can interact with the gas molecules. For soft X-rays the dominant effect in this interaction is the photoeffect. This leads to an emission of a photoelectron which carries most of the photon energy and further ionizes the gas. The number of created electron-ion pairs depends on the energy of the incident photon and the mean ionization energies of the gas components. Due to the electric field between the cathode and the anode the electrons and ions do not recombine but drift apart. Based on the electric potentials the electrons drift towards the anode and the GridPix. During this drift the electrons collide with gas molecules which leads to longitudinal and transversal diffusion. The electrons can then enter the amplification stage through the holes of the GridPix grid. Due to the high electric field in the amplification stage the electrons lead to avalanches of electrons which are detected by the Timepix pixels. In the ideal case the diffusion leads to a separation of the primary electrons such that only one electron enters one grid hole. In this case one can count the active pixels to reconstruct the energy of the incident photon.

3 Measurement

To operate the detector it is flushed with a mixture of argon (97.7%) and isobutane (2.3%). The mean ionization energies of these gases are: $W_I^{\text{Ar}} = 26 \text{ eV}$ and $W_I^{\text{C}_4\text{H}_{10}} = 26 \text{ eV}$. The chip is powered with two times 2.2 V and the cathode, anode, grid and field cage are connected to the high voltage power supply ($V_{\text{cathode}} = 1860 \text{ V}$, $V_{\text{field-cage-cathode}} = 1810 \text{ V}$, $V_{\text{field-cage-anode}} = 410 \text{ V}$, $V_{\text{anode}} = 360 \text{ V}$, $V_{\text{grid}} = 310 \text{ V}$). During the measurement a ^{55}Fe source is used. ^{55}Fe decays via electron capture to an excited state of ^{55}Mn . The deexcitation leads to photons with $E_{K_\alpha} = 5.90 \text{ keV}$ and $E_{K_\beta} = 6.49 \text{ keV}$.

For data taking the software TOS (Timepix Operating Software) is used. It controls the chip and reads and saves the data of the chip. With the command ‘run’ data taking is started based on settings in a config file. During the data taking the electronic shutter of the chip is opened and the charge is measured. The charge is measured in number clock cycles which are needed to discharge a capacitor. A calibration based on charge injection via a testpulse circuit on the chip can be used to calculate the charge in electrons based on the charge in clock cycles.

4 Data visualization



Figure 3: Grafana [2] event display and online spectra.

During data taking the data is stored in a TimescaleDB [1] database which supports SQL queries. For each frame a new column is created in the database which contains the framenummer, the runnumber, the number of active pixels, the timestamp of the frame and a two dimensional array of the x, y positions and charges of all active pixels. The data in the database is monitored by Grafana [2] which offers various possibilities to plot the data. The dashboard which will be used consists of a event display and histograms of active pixels per frame, total collected charge per frame and charge per pixel. Furthermore some variables are adjustable which act as cuts on the data. For example one can cut on the number of active pixels in frames or on the mean position of the pixel cluster in every frame.

5 Analysis

For the analysis several python scripts are used which access the database with a query, stores the returned data in an array and iterates through this array to fill the data into histograms which are plotted with ROOT [3]. In the query two SQL functions are used. ‘UNNEST’ creates a list of values based on all entries of an array and ‘AVG(a)’ calculates the average of all elements in ‘a’. With the ‘SELECT’ statement it is defined which data is returned by the database and with the ‘WHERE’ statement filters on the data are defined. In the scripts you can find variables which are used for cuts on the number of active pixels and the mean position of the pixels in the frame. The effect of the cuts shall be investigated and discussed. The scripts for the pixelspectrum and the chargespectrum contain a predefined double gaussian fit function. The charge-per-pixel script contains a predefined polya distribution which is often used to describe phenomenologically the gas gain distribution. Scripts which rely on the charge measurement contain the needed information for the charge calibration and perform it automatically on the data.

References

- [1] <https://www.timescale.com/>
- [2] <https://grafana.com/>
- [3] <https://root.cern.ch>

Timepix operation

Preparation

Procedure

1. Change to the TOS (Timepix Operation Software) folder: `cd ~/Readout/Timepix/EDIT`
2. Start TOS: `./TOS`
3. Load the existing Configuration (Number of chips, boards, FECs) via entering `yes`
4. Reset the chip: `1`
5. Load the settings for the DACs (FSR): `LoadFSR` or `lf`
6. Setup uniform pixel settings: `um`
 - `P0 = 1`
 - `P1 = 0` (P0 and P1 set the operation mode of the pixels which is TOT here)
 - `Mask = 1` (Sets that the pixels are active)
 - `Test = 1` (Sets that the pixels can receive test pulses)
 - `Threshold = 8` (Sets the pixel threshold)
7. Load the equalized pixel thresholds: `lt` (As every pixel is slightly different this makes sure that the response of the pixels is uniform)
8. Write the pixel settings to the chip: `4`
9. Read the result back - it should show 65536 pixels: `3`
10. Write the DAC settings - the chip ID (G4-W66) should be returned: `5`

Troubleshooting

- If steps 9 or 10 don't show the expected result do again steps 8 to 10.
-

Finding the threshold

Procedure

1. Start a threshold scan (It scans through a range of thresholds and shows the number of active pixels):
`THLScan`
 - Set the coarse threshold to 7 (middle of its range)
 - Set the start THL to 300
 - Set the end THL to 500 (The threshold is usually in this range)
 - Set the number of frames to 4 (to get some statistics)
 - Select the fastclock with 1 (Increases the TOT resolution)
 - Select the Long shutter range with 1
 - Set the shutter time to 3 (the resulting shutter time is 883 μ s)
2. In the result look up the threshold at which the number of pixels decreased to 0 to 2
3. Set this value as new threshold: `SetDAC 6 1 0 threshold`
4. Write the new threshold to the chip: `5`

Troubleshooting

- If the number of pixels in the threshold scan keeps high something went wrong during the preparation. So start again with the preparation
-

Readout

TOS

1. Check if the run parameters are correct: `ShowRunParameters`
 - TerminationMode should be 1 (the readout is active until a defined number of frames is recorded)
 - NumberOfFrames_Time should be 0 (infinite frames are recorded - the run is terminated by the user)
 - ExternalTrigger should be 0 (An external trigger can be used but there is none at this setup)
 - FastClock should be 1 (High TOT resolution)
 - ShutterRange should be long (Shutter time between 294.40 μ s and 75.07 ms)
 - ShutterTime should be 3 (Shutter time = 883 μ s)
 - ZeroSuppressed should be 0 (Only active pixels are stored in the frames and not all pixels)
2. If the run parameters are wrong set them with `SetRunParameters`
3. Start the run with `run`
 - Before the run starts you are asked for a run number. Just use the default number by pressing `ENTER`
 - Furthermore the run parameters are displayed again. Check them and if they are correct start the run by pressing `ENTER`
4. You can stop the run by pressing `q` and `ENTER` successively.
5. If some pixels are continuously active (You can see them in the event display) they are noisy / broken. Write down the coordinates of the pixels, stop the run with `q` and `ENTER` and perform the steps in section Deactivate pixels. Start then again with step 3 in Readout.

Monitoring with Grafana

1. Start Firefox
2. Go to the address `localhost:3000`
3. You will see the default dashboard which shows the following information:
 - On the top right you can see the event display. It shows the active pixels of the newest frame
 - On the top left you can see the the online pixel spectrum. It shows the number of active pixels per frame
 - On the bottom left you can see a histogram of the total charge collected by all pixels per frame
 - On the bottom right you can see a histogram of the charge per pixel per frame
4. On top of the page you can set cuts on the number of pixels if you do so all plots show only frames which fit this range. Additionally you can set the refreshing rate at the top right of the page.

Troubleshooting

- If the run stops only after a few amount of frames there is a problem with the firmware on the SRS-FEC. Perform the following steps:
 1. Kill TOS by pressing `CTRL` and `C`
 2. Switch off the SRS-FEC with the power switch on the front-side.
 3. Wait until all red power LEDs on the power supply are off.

4. Power on the SRS-FEC with the power switch.
 5. Wait until one of the green SRS-FEC LEDs starts blinking while the other one is off
 6. Make sure that the PC is connected to the Timepix SRS network
 7. Do the steps 1 to 4 described in [Preparation](#)
 8. Kill TOS again by pressing **CTRL** and **C**
 9. Start again with the [Preparation](#) and perform all following steps. The run should now work
- If all frames show 0 active pixels it is possible, that the high voltage tripped. To check this perform the following steps
 1. Stop the run with **q** and **Enter**
 2. Open a second terminal
 3. Change to the directory `~/LabTools/isegControl-build`
 4. Start the HV interface with `./isegControl`
 5. Look if the channels 1 to 5 are on and that the measured voltages are the set voltages. If this is not the case let the tutors know.
-

Deactivate pixels

1. Start editing the pixel configuration: **em**
 - Enter 3 for editing individual pixels
 - Enter the coordinates of the first pixel
 - P0 = 1
 - P1 = 0 (P0 and P1 set the operation mode of the pixels which is TOT here)
 - Mask = 0 (Sets that the pixel is inactive)
 - Test = 0 (Sets that the pixel can't receive test pulses)
 - Threshold = 8 (Sets the pixel threshold)
 - If you want to edit additional pixel enter **yes** else **no**
 2. Load the equalized pixel thresholds: **lt**
 3. Write the pixel settings to the chip: **4**
 4. Read the result back - it should show 65536 pixels: **3**
 5. Write the DAC settings - the chip ID should be returned: **5**
-

Analysis

With Grafana

1. Start Firefox
2. Go to the address `localhost:3000`
3. On the top left of the page change from "TOS Monitoring (Online)" to "TOS Monitoring (Offline)"
4. Select your run number on the top left of the page
5. You can set again the range of active pixels but also the minimum distance of the mean cluster postilion to the edge of the chip. Additionally you can select single frames within this range which are then displayed in the event display

With Python

1. Change to the analysis folder: `cd ~/Analysis/EDIT2020/groupname`

2. You find five python scripts for plotting the data. You run them with `python scriptname.py runnumber`. The scripts are:
 - `pixelspectrum.py` - Plots a histogram of active pixels per frame
 - `chargespectrum.py` - Plots a histogram of the collected charge per frame
 - `charge_per_pixel.py` - Plots a histogram of the charge per pixel per frame
 - `occupancy.py` - Plots a 2D histogram of active pixels during the whole run
 - `centers.py` - Plots a 2D histogram of the mean pixel position of each frame
3. Within the script you can add cuts on the number of active pixels and the mean pixel position
4. Additionally you can apply fits to the histograms.

GridPix tasks

1. Perform the steps described in [Preparation](#), [Finding the threshold](#) and [Readout](#) to record ^{55}Fe data. To collect enough data the run should last about 30 minutes.
2. During the run use the Grafana monitoring tool to understand the data and the plots.
3. Have also a look on the python scripts in `~/Analysis/EDIT2020/groupname` and understand how they work and what they are plotting.
4. Use the python scripts to plot the data without cuts. Explain the results.
5. Investigate the effect of the cuts (min and max pixels, center_distance) on the different histograms. Why is a cut on the position of the clusters important?
6. Fit the pixelspectrum, the chargespectrum and the charge_per_pixel spectrum with the predefined functions. You may have to adjust the start values of the fit parameters and the fit range. If the [doublegauss](#) does not work properly use the [singlegauss](#) to fit the peaks separately.
7. Use the fit results to determine the energy resolution in the pixelspectrum as well as in the chargespectrum.
8. Use the fit result of the charge_per_pixel spectrum to determine the gas gain.
9. How could one use the fit results of the pixel and the chargespectrum to determine the gas gain. Which assumptions are needed?