CMOS sensor & ILC group

IPHC
Institut Pluridisciplinaire Hubert Curien
STRASBOURG

Calibration demonstration setup manual for an analog output CMOS pixel sensor

J.Baudot, 2011, January 26

1 Goals of the experiment

Interconnect all the elements necessary to operate and acquire data from a CMOS pixel sensor issue from the MIMOSA series: MIMOSA 18.

Estimate the charge calibration factor (ADC units per electrons collected in a pixel) and the equivalent noise charge of the sensor. Indeed any acquisition system provides output values in arbitrary units named ADC units. We are nevertheless interested in relating the output measured to a number of charges generated in the sensor. This calibration factor can be obtained using a monochromatic radioactive source which generates always the same amount of charges in the sensor (for instance the ^{55}Fe source has two X rays 5.90 and 6.49 kEV).

Estimate the ratio of the number of charges collected on pixels over the number of charges genuinely generated by a single particle. This is the charge collection efficiency of the pixel sensor.

2 MIMOSA 18

The detector used in this exercise is MIMOSA 18, a CMOS sensor designed by IPHC in 200 and fabricated in 200. The chip features 4 equivalent submatrices of 256x256 pixels with a 18 µm pitch.
Four outputs convey the signal of each pixel from the 4 sub-matrices out of the sensor. With a 10 MHz clock frequency, the whole matrix (262 144 pixels) is read in 6.6 ms which is also the integration time of the sensor.

3 Doing the calibration

The calibration goal is reach through the following steps.

- Setup the experiment (see section 5 and 6).
- Adjust the auxiliary board settings (requires to start a noise run without source, the corresponding output files can be deleted afterward).
- Start a noise run (without source) over 400 events, see the section 4. This will be your reference noise run for the subsequent acquisition. You will see the pedestal and noise computation convergence on the online displays. Thus, the output file "Init_" in the data_demo directory should not be deleted!
- Start a run (with a different run number) with the source placed on top. The online display will show you the different cluster characteristics as reconstructed by the clustering algorithm, see an example in the window below (note there are three additional plots in the current version).
- Observe and interpret the various histograms. Especially important ate the peaks seen in the distribution of the charge collected by the seed pixel (plot indicated Qseed below) and of the sum of the charges collected on all the pixels of the cluster (plot indicated Qclusters below). The position of these peaks will allow you to estimate:
    - the charge-to-adc unit calibration factor,
    - the charge collection efficiency of the sensor for a given cluster size.

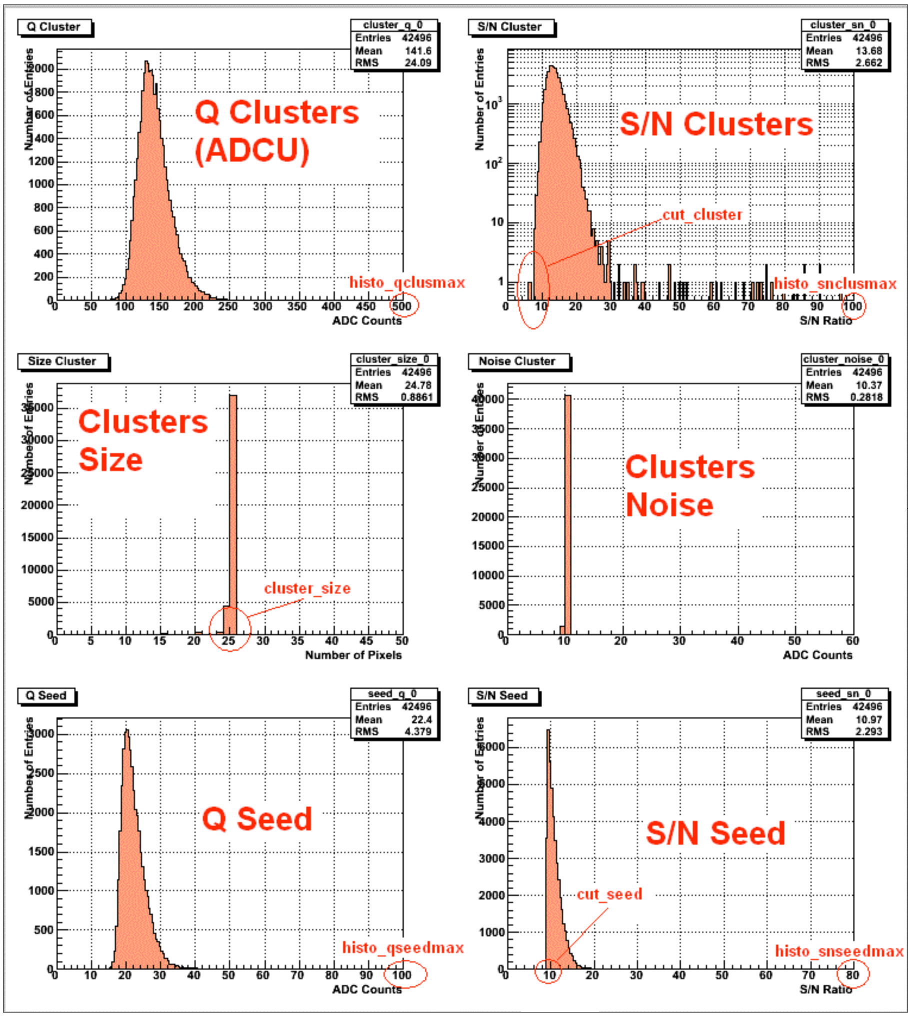

Figure 1 Example of online display during an acquisition.

4 Running

To run (any type of data acquisition), you shall always execute the following sequence of operation in the same order:

a. Place the radioactive source where it should be,

b. DAQ and IPNL-DAQ are not running,

c. Delete any file or directory in the ramdisk directory (you may double-click the clean_RAMDISK.bat script),

d. Update the runInfo.txt file with the desired parameters (see section 6.1),

e. Launch IPNL-DAQ, a window pops-up saying “waiting for DAQ-IPHC”,

f. Launch DAQ and set properly the parameters (see section 6.3), pay attention to the run number and number of events,

g. Push the “start run” button in the “Run” tab and you are running.

Displays open automatically and you cannot adjust them while the acquisition is running. The run stops either because the required number of events is reached or because you pushed the "stop run" button on the DAQ interface.
You shall quit the IPNL-DAQ and DAQ programs by a closing their windows.

The output of the run is a set of root files in the data_demo directory.

5 Installing the setup

5.1 The hardware

You shall have the following parts in your hands:

- two power supplies, ±8 V (could be a single supply with two outputs),
- a USB-Imager card inserted in a VME crate,
- a PC running Windows with the software installed,
- a standalone board (named auxiliary board),
- a set of 4 cables labeled (PC-USB, CLOCK, DATA, power),
- an aluminum box containing a MIMOSA 18 sensor,
- a ^{55}Fe soft X-ray source.

Connect all the parts together then proceed as follow:

a. switch on the power supplies, you should observe the following current:
    - ≤ 500 mA on the +8 V and ≤ 200 mA on the -8 V,
      (note that the +8V powers the auxiliary board and the sensor whereas the -8V powers only the board)

b. switch on the USB-Imager board.

You are ready for the next step.

5.2 The software

In case you are using the IPHC fixed computer, choose the following login parameters:

- Computer = sbgat100 (cet ordinateur),
- Login = mgoffe_adm,
- Password = tyuiop.

Four programs are needed to acquire and analyze the data on the Windows system:

- the Root package from CERN,
- the control interface to the USB-Imager card,
- the online analysis and display package named DAQ-IPNL,
- an emulator for a virtual disk used for the communication between the online analysis and the USB-Imager card,

We consider these packages are already installed. You have to check the three following short cuts are available from the Desktop:

- DAQ: C:\ccmos_daq\v3.4.1\daq.bat
- DAQ-IPNL: C:\ipnl_daq\daq_ipnl_program_test\bin_DAQIPNL\DAQ-IPNL.exe

- runInfo.txt (data card file which sets the analysis parameters): C:\ipnl_daq\daq_ipnl_program_test\bin_DAQIPNL\run_infos.txt,
- data_demo: D:\data_2010\data_demo,
- ramdisk(V): V:\,
- clean_RAMDISK.bat is not a shortcut but is a small script stored directly on the Desktop.

6 Settings & launching the programs

6.1 Auxiliary board settings

Start a noise run (without source) and by checking with a voltmeter set the following values, see the picture below:

- Vcc = 3.3 V
- V+chip = 3.3 V
- InRef ≈ 1 V
- VmR ≈ 1V
- VmRef ≈ 1.25 V
- Ref0, Ref1, Ref2, Ref3 between 1.5 to 2,1 V.

The last voltages Refx, control the pedestal of each output. Adjusting this level impact the dynamic. Knowing the digitizer uses 12 bits, the largest possible value is 4096. An equal dynamic for positive and negative signals corresponds to a pedestal around 2024.

Once adjusted properly, the “rawdata ” plot should be around 2000 ADC units.

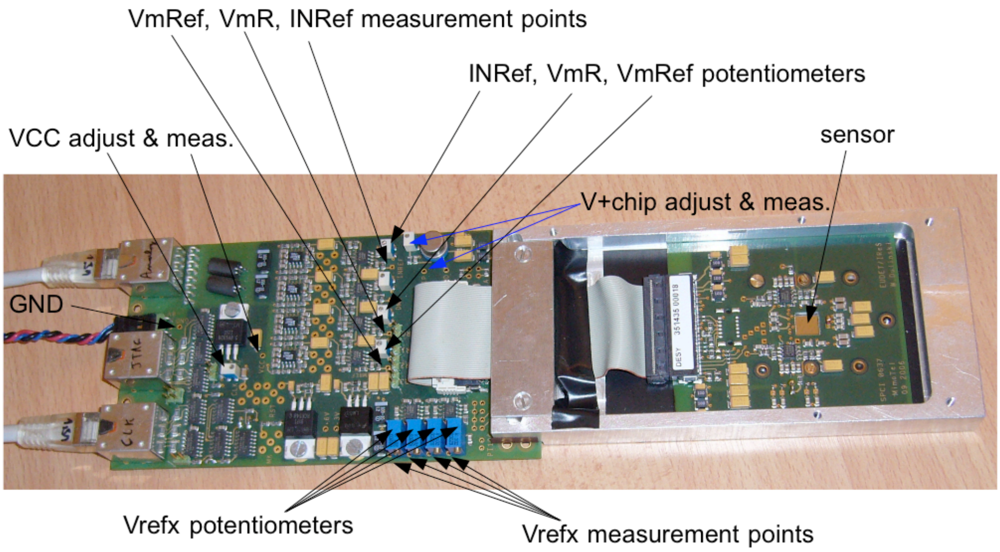

Figure 2 Locations of the measurement points and potentiometers for the biasing voltages on the auxiliary board.

6.2 Analysis parameters settings (IPNL-DAQ configuration)

Double click on the runInfo.txt shortcut to edit the file. For basic operation, you have only to modify the following parameters:

- **Run,**
- **Pedestal_run.**

To launch the software, just double click on the IPNL-DAQ shortcut. A root window will open and nothing more until you start the DAQ itself. Then the online display Windows will pop-up.

However, for the sake of completeness, this is the list of all parameters.

Chip
Indicate which sensor is used.
value: mimosa18

Inpath
Directory where raw files from DAQ-IPHC are saved (i.e. Run_XX.rz). Equivalent to the "Run directory path" field in the DAQ interface.

Outpath
Directory where output files of DAQ-IPNL are saved. If this directory doesn't exist, DAQ-IPNL will stop and ask you to create it.
value: V:\data

Run
ID number of the data acquisition you are about to run. Equivalent to the "run number" field in the DAQ interface.
value: something like 18xxx

Pedestal_run
Run number of the data acquisition with the reference pedestal and noise computation when Pedestal_run is different from the Run parameter.
If the two parameters Run and Pedestal_run are equal, the software will first initialise, that is: recompute the pedestals and noises of each pixel using the number of events specified by Init_event.
Note that if this field is not set, the pedestal run is equal to the run number.
value: something like 18xxx

Init_events
Number of events of the initialisation step (for example, 250 evts).

Total_events
Total number of events in the run (including init step). If you choose online_daq mod (see later), you should write a high value, for example 100000, in order to stop the run only when you click on "Stop" button in DAQ-IPHC.

Online_daq

MANDATORY value : 1
It forces the analysis to run "online" mode, i.e. the *.rz files saved by DAQ-IPHC will be automatically removed by DAQ-IPNL. Then it is possible to perform a very long acquisition without filling the disk space. It is the preferred mod to operate because the most flexible. Note that for this mode, the field **"Events per file" must be set to 1** in DAQ-IPHC.
If you choose "0", then the analysis is offline and the *.rz files are not deleted. It is useful if you want to compare the effects of the cuts on the same events.

Write_raw_frame

Possible values: 0 or 1
Default value: 0
If you choose "1", the raw S/N frames are saved in the ouput file "RawFrames_RunXX.raw". Useful to do a customized offline analysis of the events without having to reprocess pedestal, noise, etc.
Note that if a Region Of Interest is defined (see later), only this region will be saved in the file.

Base_value_type

MANDATORY value: cdsinv
Set the sign of the correlated double sampling to frame0-frame1 used for MIMOSA 18.

Noisy_pix

Possible values: any positive value.
Default value: 100.
Pixels with noise > noisy_pix will not be used in the analysis neither in the clusters constructions. The cut has to be set by looking to the noise distribution at the end of the initialisation step.

Dead_pix

Possible values: any positive value.
Default value: 0.
Pixels with noise < dead_pix are considered as dead. They will not be used in the analysis neither in the clusters constructions. The cut has to be set by looking to the noise distribution at the end of the initialisation step.

Cluster_size

Possible values: 9, 25 or 49.
Default value: 25.
Clusters are defined as squares of N pixels around a seed (the central and highest S/N pixel).

Cut_seed

Possible values: any positive value.
Default value: 5.
A pixel can be considered as a seed if S/N > cut_seed and if no other pixel in the cluster square has a highest S/N ratio (to avoid over-mapping problem).

Cut_cluster

Possible values: any positive value.
Default value: 0.
A cluster is saved and fill the histograms if the S/N ratio is > cut_cluster.

Use_ROI

Possible values: 0 or 1.
Default value: 0.
If "0", the whole frame is read, analyzed and saved on the disk (if the corresponding options are set).
If you are interesting in looking at a local phenomenon, it is useful to define a Region Of Interest (choice "1"). Then only clusters contained in this area (see ranges below) are taken into account in the histograms. If the region of interest mode is set, a new canvas is opened and displays the positions of the clusters in the defined area (see description of the canvas "Region Of Interest").
Please note that the range is defined in the DAQ-IPNL frame (by looking the front-side of the chip 0;0 is the bottom-left corner, X refers to the column index and Y refers to the line index). If you are running on mimosa5 with 4 matrices, the position refers to the absolute position on the chip (i.e. 0-> 1024), not on the matrix.

Roi_xmin

Possible values: from 0 to frame width value.

Roi_xmax

Possible values: from 0 to frame width value.

Roi_ymin

Possible values: from 0 to frame height value.

Roi_ymax

Possible values: from 0 to frame height value.

Histo_noisemax

Possible values: any positive value.
Default value: 10

Histo_qclusmax

Possible values: any positive value.
Default value: 400.

Histo_snclusmax

Possible values: any positive value.
Default value: 50

Histo_qseedmax

Possible values: any positive value.

Default value: 300

Histo_snseedmax

Possible values: any positive value.
Default value: 200

Histo_log_qclus

Possible values: 0 or 1.
Default value:0 (linear scale)

Histo_log_snclus

Possible values: 0 or 1.
Default value:0 (linear scale)

Histo_log_qseed

Possible values: 0 or 1.
Default value:0 (linear scale)

Histo_log_snseed

Possible values: 0 or 1.
Default value:0 (linear scale)

ROI_nbins

Possible values: any positive value.
Default value: 100.
This option defines the number of bins used in the histograms related to the clusters COG positions and drawn in the ROI canvas. Only used if *use_ROI =1.*

6.3 Data acquisition settings (DAQ configuration)

Launch the DAQ program. Click "Board Testing" and wait for the final OK message. You can then close this window and there is no need to "test" again.

The first settings chose which type of sensor you are reading and at which frequency. You set it once for all.
In the left panel of the window chose the mode "mi18_sc_a0a1a2a3_10MHz" which corresponds to reading the 4 sub-matrices at 10 MHz with a timing tuned for a short cable between the sensor and the USB-Imager card.
The proper configuration is loaded and there is nothing more to do.

The second settings control were the data will be stored and deal with the right panel of the window, precisely the tab "Run Parameters".
Set the following fields:

- "Run directory" to "v:\data",
- "Event per file" to "1"
- check the checkbox "save".

Those are permanent settings you won't need to change later on.

To launch a run, go again to the right panel of the window, precisely the tab “Run Parameters” and sets:

- “events” to something high like 50000,
- “Run number” to something like 18xxx. **This should be the same run number as the Run parameter from the runInfo.txt file.**

Then click the “**Start Run**” button. You may stop the run by clicking the “**Stop Run**” button.

Your DAQ window should look like the following picture.

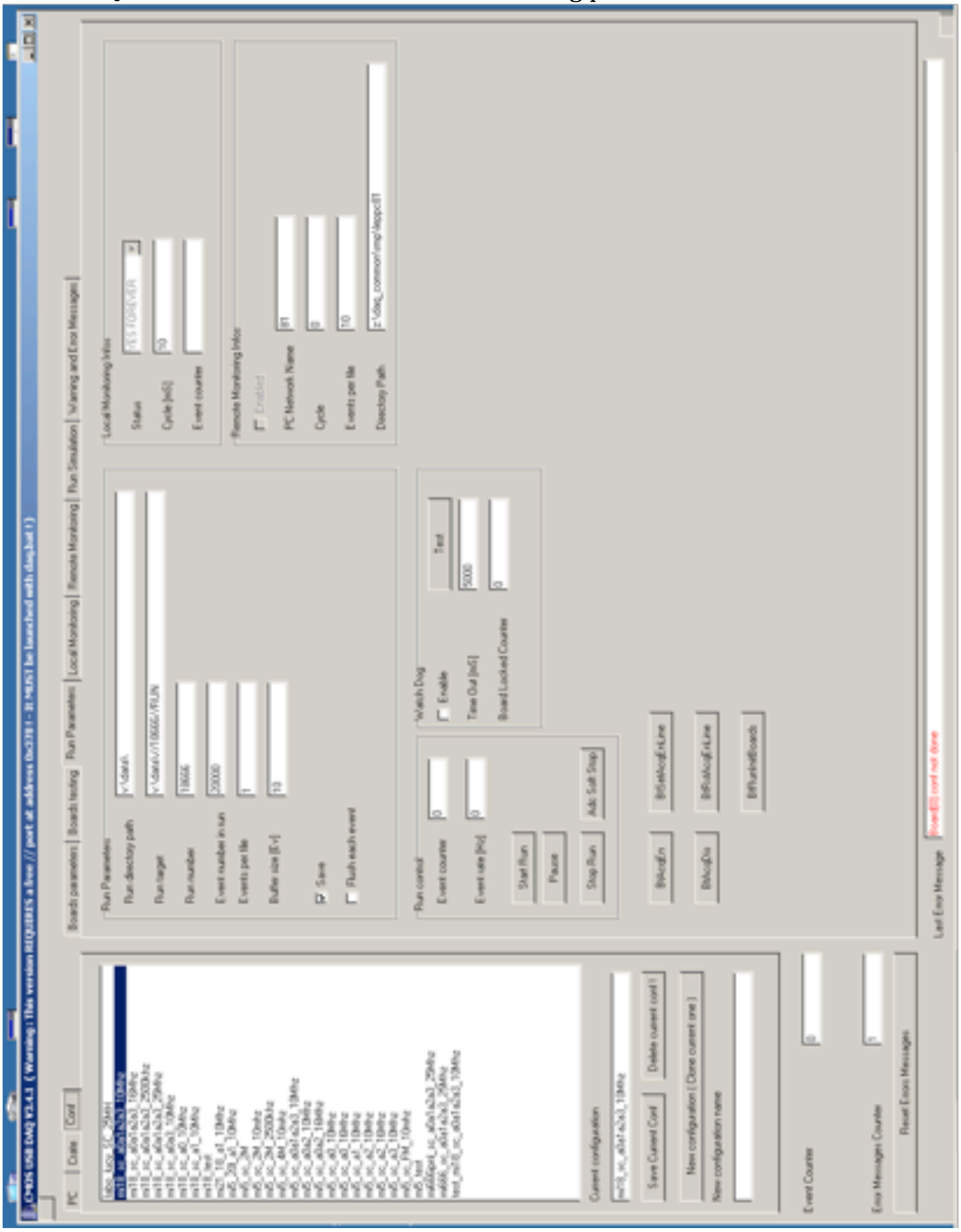

Figure 3 Interface to the USB-imager card with proper settings.

7 Troubleshooting

7.1 USB-Imager hanged

Close the DAQ program interface and power Off/On the card itself. Relaunch the DAQ interface and test again.
If it is still hanged, switch the card and the power as well and retry.

7.2 Severe PC crash

7.3 Sensor dead

Use the spare sensor (only one available). Unscrew the top of the aluminum box and unplug the flat cable. Unscrew the proximity board on which the sensor is bonded and replace it by the spare. Fix the board with the screw, plug the cable and screw the box top.
With a new sensor, you may have to change the Refx voltages, see section 6.1.

8 Acknowledgments

The experimental setup was originally designed by the IPHC group with:

- Mathieu Goffe for the hardware,
- Gilles Clauss for the DAQ software.

The online analysis code IPNL-DAQ was created by Nicolas Estre from IPNL.
The MIMOSA 18 sensor was designed by Wojciech Dulinsky within the IPHC group and fabricated within the EUDET collaboration (European Frame Programme 7) in order to equip a beam telescope.