TNT data analysis

Christian Bonnin

IPHC (Institut Pluridisciplinaire Hubert Curien)
<christian.bonnin@iphc.cnrs.fr>

Marc Richer

IPHC (Institut Pluridisciplinaire Hubert Curien)
<marc.richer@iphc.cnrs.fr>

18/06/2013

Table of Contents

List of Figures

Chapter 1. Introduction

Table of Contents

Acquisition cards

The development of TNT (Treatment for Numerical Tracking) data acquisition cards started in 2001 at IPHC (old IReS), CNRS. During 2004 a second card version has been designed (TNT2) which has 4 channels sampled up to 100 Mhz [3].

Softwares

Here is the list of softwares developped to drive and process data from the cards:

- TUC 5 controls one or more cards and can both parameterize them and retrieve their data. It has its own documentation [1] (script: *goTuc5.sh [goTuc5.bat under Windows]*)
- TAN is used to process offline data (script: *goTan.[sh/bat]*)
- DTUC is especially written to handle acquisitions with a lot of TNT cards and pilot distributed installation (scripts: GUI to drive acquisition: *DTUC-goGUI.[sh/bat]*, Data server to drive cards and send data over network: *DTUC-goServer*)
- EventViewer is a small utility used to visualize event files (script: *goEventViewer.[sh/bat]*)

This documentation is dedicated to TAN. Here are the data types it can handle:

- Events
- Histograms
- Bidimensional histograms
- Oscillograms

Online documentation: TUC and TAN have contextual online documentation, i. e. you can press F1 in a dialog box and with the focus on a particular control to get help on this topic

Data formats

The dedicated software TUC (Tnt Usb Control) can write out several data types, among them:

- Energy events
- ASCII 1D histograms
- Bidimensional histograms
- Oscillograms

These formats can be processed off line by the TNT Analyser software or TAN which has been developed at IPHC.

Chapter 2. Events data

Table of Contents

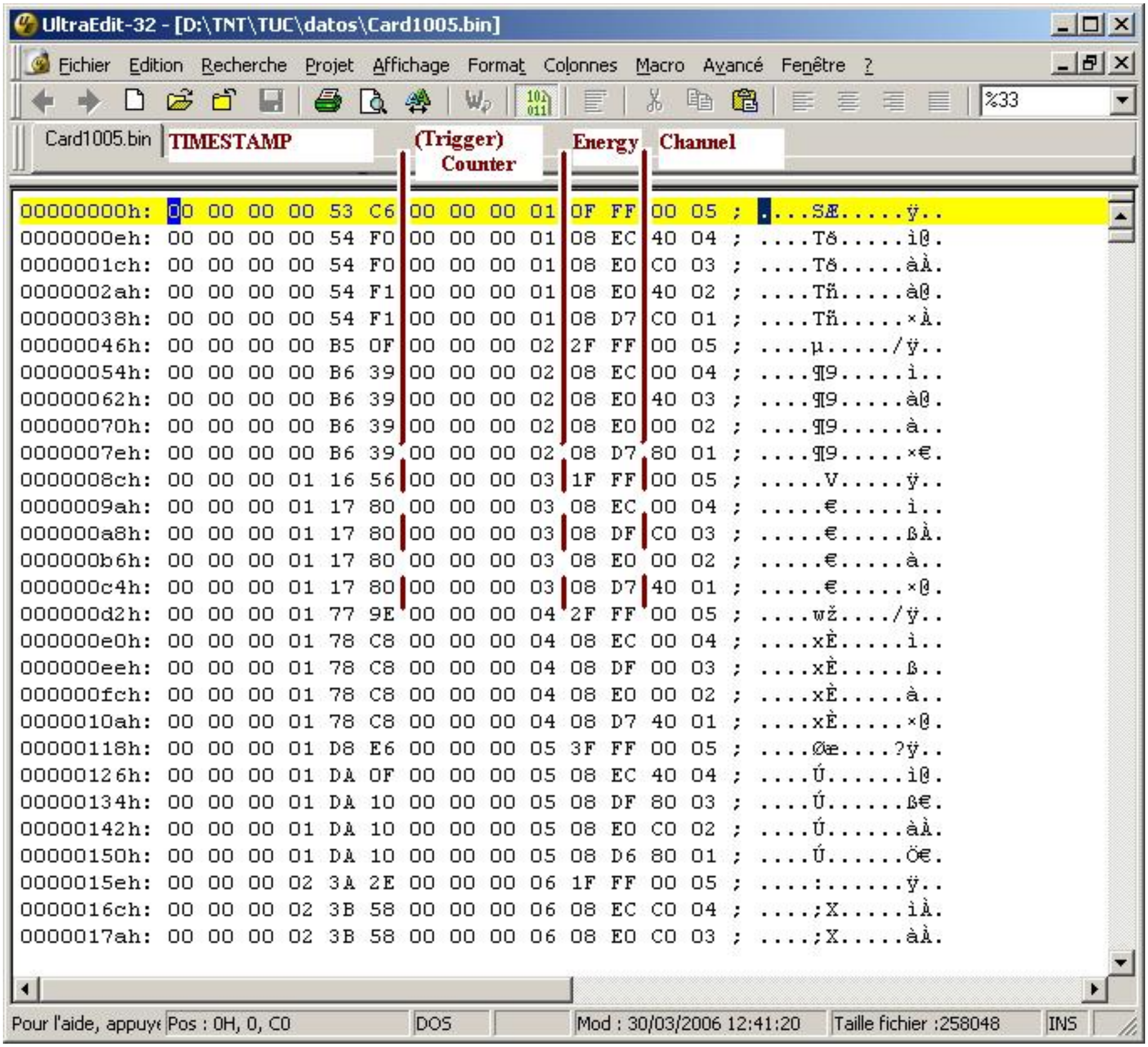
Event files format

In energy acquisition mode each time a energy deposit is detected (a channel's trigger fires), a full 14 bytes event is constructed and stored.

As a result, event data is formatted following the the big endian order which is used for arranging byte order. These fields can be found within a full event (see figure Event format)

- 6 (unsigned) bytes for the timestamp value of trigger occurrence. This is the number of clock periods since the counter starts (default: acquisition start).
- 4 (unsigned) bytes for the trigger counter value which increments each time a trigger occurs
- 2 signed bytes for the energy value [-32768,+32767]
- 2 bytes containing additional informations
    - Channel number 0, (1 to 4) in rightmost bits
    - Pile-up event bit
    - Bit for coincidence/veto using NIM input
    - Bit for coincidence/veto using VETO input
    - ADC out of range bit

Figure 2.1. Event format: Typical binary acquisition file. Each line represents a full 14 bytes event



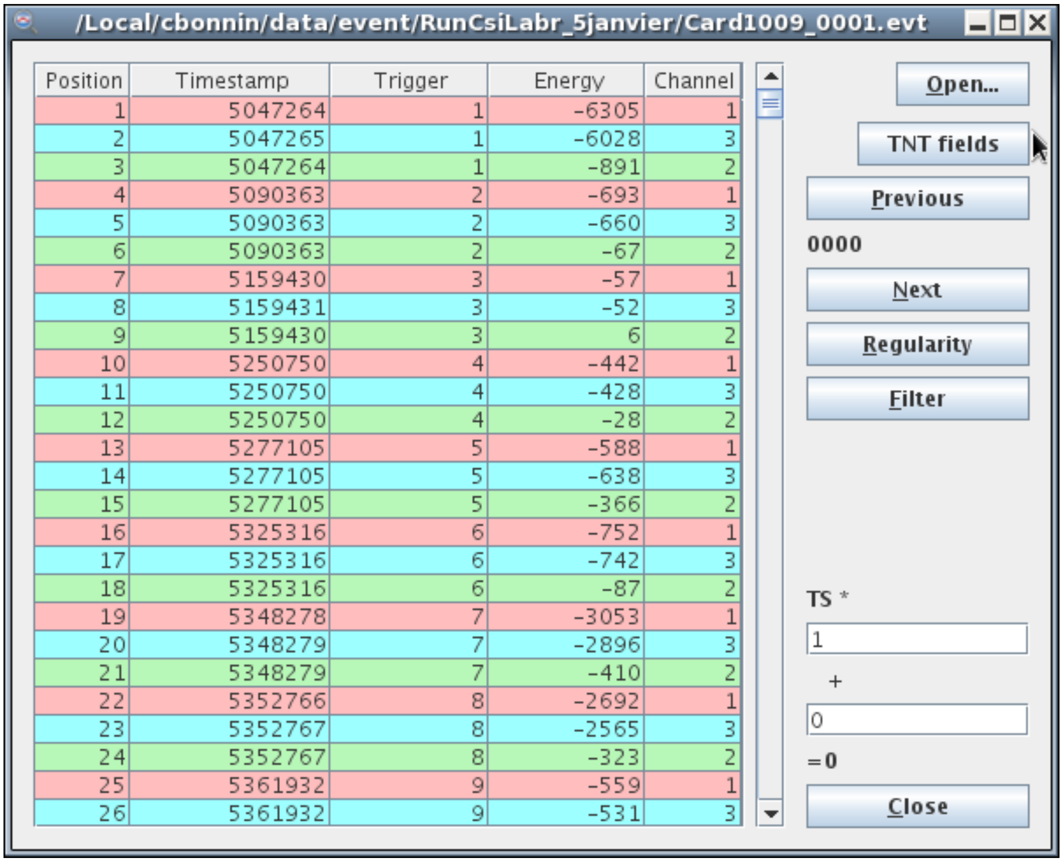
```
00000000h: 00 00 00 00 53 C6 00 00 00 01 0F FF 00 05 ; ....SÆ.....ÿ..
0000000eh: 00 00 00 00 54 F0 00 00 00 01 08 EC 40 04 ; ....Tð.....ì@.
0000001ch: 00 00 00 00 54 F0 00 00 00 01 08 E0 C0 03 ; ....Tð.....àÀ.
0000002ah: 00 00 00 00 54 F1 00 00 00 01 08 E0 40 02 ; ....Tñ.....à@.
00000038h: 00 00 00 00 54 F1 00 00 00 01 08 D7 C0 01 ; ....Tñ.....×À.
00000046h: 00 00 00 00 B5 0F 00 00 00 02 2F FF 00 05 ; ....µ...../ÿ..
00000054h: 00 00 00 00 B6 39 00 00 00 02 08 EC 00 04 ; ....¶9.....ì..
00000062h: 00 00 00 00 B6 39 00 00 00 02 08 E0 40 03 ; ....¶9.....à@.
00000070h: 00 00 00 00 B6 39 00 00 00 02 08 E0 00 02 ; ....¶9.....à..
0000007eh: 00 00 00 00 B6 39 00 00 00 02 08 D7 80 01 ; ....¶9.....×€.
0000008ch: 00 00 00 01 16 56 00 00 00 03 1F FF 00 05 ; .....V.....ÿ..
0000009ah: 00 00 00 01 17 80 00 00 00 03 08 EC 00 04 ; .....€.....ì..
000000a8h: 00 00 00 01 17 80 00 00 00 03 08 DF C0 03 ; .....€.....ßÀ.
000000b6h: 00 00 00 01 17 80 00 00 00 03 08 E0 00 02 ; .....€.....à..
000000c4h: 00 00 00 01 17 80 00 00 00 03 08 D7 40 01 ; .....€.....×@.
000000d2h: 00 00 00 01 77 9E 00 00 00 04 2F FF 00 05 ; ....wž..../ÿ..
000000e0h: 00 00 00 01 78 C8 00 00 00 04 08 EC 00 04 ; ....xÈ.....ì..
000000eeh: 00 00 00 01 78 C8 00 00 00 04 08 DF 00 03 ; ....xÈ.....ß..
000000fch: 00 00 00 01 78 C8 00 00 00 04 08 E0 00 02 ; ....xÈ.....à..
0000010ah: 00 00 00 01 78 C8 00 00 00 04 08 D7 40 01 ; ....xÈ.....×@.
00000118h: 00 00 00 01 D8 E6 00 00 00 05 3F FF 00 05 ; ....Øæ....?ÿ..
00000126h: 00 00 00 01 DA 0F 00 00 00 05 08 EC 40 04 ; ....Ú......ì@.
00000134h: 00 00 00 01 DA 10 00 00 00 05 08 DF 80 03 ; ....Ú......ß€.
00000142h: 00 00 00 01 DA 10 00 00 00 05 08 E0 C0 02 ; ....Ú......àÀ.
00000150h: 00 00 00 01 DA 10 00 00 00 05 08 D6 80 01 ; ....Ú......Ö€.
0000015eh: 00 00 00 02 3A 2E 00 00 00 06 1F FF 00 05 ; ....:......ÿ..
0000016ch: 00 00 00 02 3B 58 00 00 00 06 08 EC C0 04 ; ....;X.....ìÀ.
0000017ah: 00 00 00 02 3B 58 00 00 00 06 08 E0 C0 03 ; ....;X.....àÀ.
```

Event Viewer

This small utility program visualizes the events of a file.

It can be launched using the *goEventViewer.sh* script (*goEventViewer.bat* under Windows).

Figure 2.2. Event Viewer



/Local/cbonnin/data/event/RunCsiLabr_5janvier/Card1009_0001.evt

Position	Timestamp	Trigger	Energy	Channel
1	5047264	1	-6305	1
2	5047265	1	-6028	3
3	5047264	1	-891	2
4	5090363	2	-693	1
5	5090363	2	-660	3
6	5090363	2	-67	2
7	5159430	3	-57	1
8	5159431	3	-52	3
9	5159430	3	6	2
10	5250750	4	-442	1
11	5250750	4	-428	3
12	5250750	4	-28	2
13	5277105	5	-588	1
14	5277105	5	-638	3
15	5277105	5	-366	2
16	5325316	6	-752	1
17	5325316	6	-742	3
18	5325316	6	-87	2
19	5348278	7	-3053	1
20	5348279	7	-2896	3
21	5348279	7	-410	2
22	5352766	8	-2692	1
23	5352767	8	-2565	3
24	5352767	8	-323	2
25	5361932	9	-559	1
26	5361932	9	-531	3

The line colours depend on the channel numbers

Files can be open by clicking on the **Open** button or by adding the path to the command line.

The **TNT fields** toggle button shows or hides the special bits and the dummy field which is filled in merged files

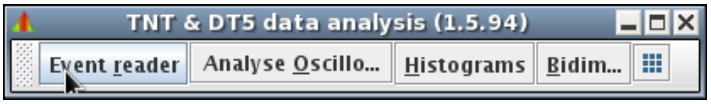
The **Previous** and **Next** buttons move to the previous or next event with the same channel number as the selected one

The **Filter** popup menu allows to exclude channel numbers from visualisation

Event reader

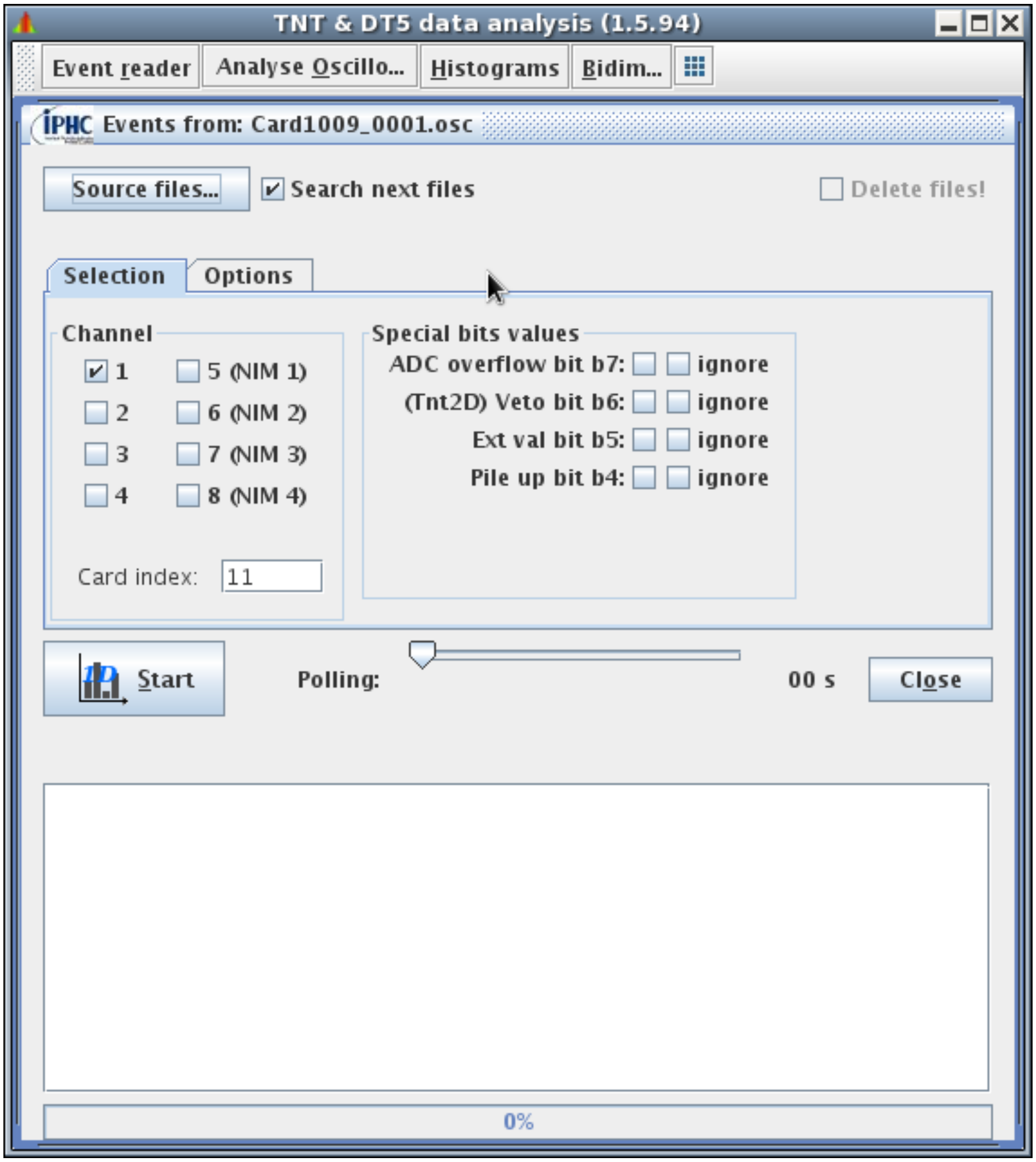
Event files usually have an *.evt* extension and are processed using the EventReader button.

Figure 2.3. EventReader button



1D histograms

Figure 2.4. EventReader window



Events from: Card1009_0001.osc

☑ Search next files □ Delete files!

Channel

☑ 1 □ 5 (NIM 1)
□ 2 □ 6 (NIM 2)
□ 3 □ 7 (NIM 3)
□ 4 □ 8 (NIM 4)

Card index: 11

Special bits values

ADC overflow bit b7: □ □ ignore
(Tnt2D) Veto bit b6: □ □ ignore
Ext val bit b5: □ □ ignore
Pile up bit b4: □ □ ignore

Polling: 00 s

0%

Click the **Source files** button to choose one or more event files.

Select the **Search next files** check box if you want to process not yet generated files. This allows to add new generated files into histograms as soon as they are complete.

Channel selection

Select the desired channel check box. TNT cards have up to 4 channels but Logical Synchronization Events (LSE) corresponding to NIM inputs can also be generated.

The **card index** field is used when a file contains the data of several cards. Those files are merged files, they usually have an `.merged` extension and share the same format as event files.

Special bits

When an event has one or more special bits set (Overflow, Veto, No external validation, Pile up), it is by default excluded from the treatment.

If one or more check boxes of the left column are checked then only events with these bits set are processed. The right column is used to ignore a special bit and process all events. Two boxes of the same line cannot be checked.

Options

The options tab check boxes allow the user to:

- Generate histograms that will also contain negative energy values (range is [-32768, 32767])
- Transform energy values into their opposite (useful when they are negative because of negative pulses)
- Calculate an Effective Number Of Bits using the formula ENOB = 14 - $\log_2(\sigma * \sqrt{12})$ where σ is the standard deviation of a white noise. In this case, energy values are considered as sampled ADC values (an oscillogram file can be opened instead of the event file).

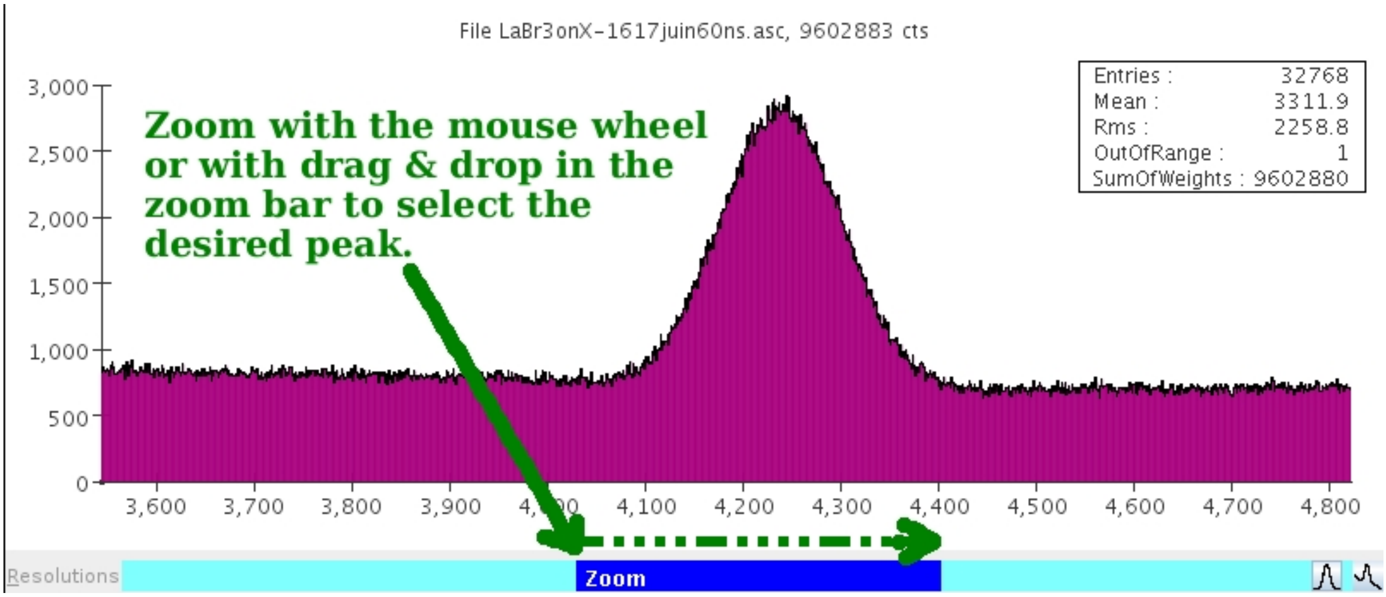
Operations on 1D histograms

The **Start** button opens a window containing an histogram in which the energy values are added as soon as the events are read from the files. This process can be interrupted at any moment by clicking the **Stop** button.

The user can zoom in and out with the mouse wheel. Dragging the horizontal rule near its center moves the histogram while dragging it nears its ends stretches the histogram.

Blue bar: A precise zoom can be performed by dragging the mouse pointer in the blue bar under the histogram.

Figure 2.5. Zoom in histograms



File LaBr3onX-1617juin60ns.asc, 9602883 cts

Entries : 32768
Mean : 3311.9
Rms : 2258.8
OutOfRange : 1
SumOfWeights : 9602880

Zoom with the mouse wheel or with drag & drop in the zoom bar to select the desired peak.

This blue bar also contains buttons to add fit functions to the histogram peaks. First zoom so that the peak to be fitted fills all the visible area.

- The ⋀ button allows to add a gaussian function based on zero. This gaussian curve can then be moved or stretched by dragging its handles.
- The ⋀ button allows to add a 'Gaussian with pedestal' fit function that allows to take some background noise into account. The heights of the 'pedestal' are taken at the left and the right of the visible area. This fit function has no handle.

After one or more fit functions have been added, the **Resolutions** button at the left of the blue bar shows theirs characteristics and allows to modify their parameters.

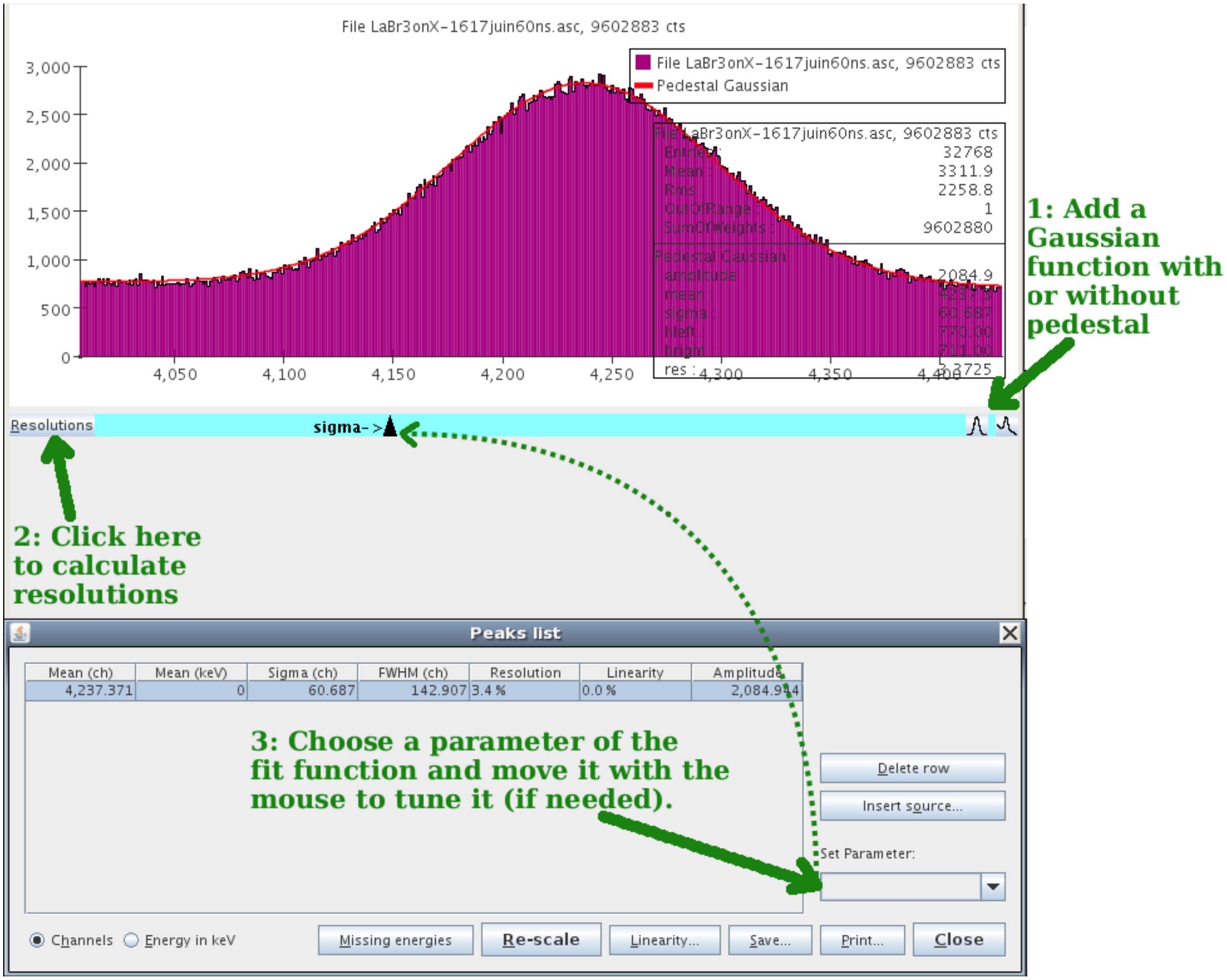
The Gaussian function formula is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\,\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The Gaussian with pedestal merely adds a affine function ($ax+b$) as background.

The Full Width Half Maximum of a Gaussian function is given by :
$H = 2\sqrt{2\,\ln(2)}\;\sigma \simeq 2,3548\sigma$

Figure 2.6. Fit functions



1: Add a Gaussian function with or without pedestal

2: Click here to calculate resolutions

3: Choose a parameter of the fit function and move it with the mouse to tune it (if needed).

Mean (ch)	Mean (keV)	Sigma (ch)	FWHM (ch)	Resolution	Linearity	Amplitude
4,237.371	0	60.687	142.907	3.4 %	0.0 %	2,084.944

To modify the parameter of a fit function, select its corresponding row in the table,then select the parameter in the combo box at right hand. You can choose the:

- amplitude
- mean
- sigma (gaussian width)
- left pedestal height (hleft: only for pedestal gaussian)
- right pedestal height (hright: only for pedestal gaussian)

A cursor appears in the blue bar below the histogram that can be dragged to adjust the parameter value. A χ^2 value is computed at the right hand. Minimize it to choose the right position.

In the Peaks list table (**Resolutions** button) the second **Mean** column is editable allows to enter the energy in keV corresponding to an energy value. The **Insert source** fills it automatically with a given common radioactive source peak value (select the right row first). These common values can be edited and stored on your local disk.

If more than two Mean values in keV are entered, their **Linearity** against the channel value can be drawn. The **Missing energies** buttons fills the 0 keV energy values with the one taken on the linear regression of the other peaks.

The **Re-scale** button fills another histogram with keV energy values on the X axis.

The **Save** and **Print** buttons respectively save and print the table values

Besides, a right click on the blue bar opens a pop-up menu with the following commands:

- **Back to origin** cancels the zoom and returns to original X axis limits
- **Save to file** export the histogram into an ASCII histogram file
- **Sum of weights** computes the sum of the bin weights within the visible area
- **Rebin** refills the histogram bin heights into a lower number of bins.

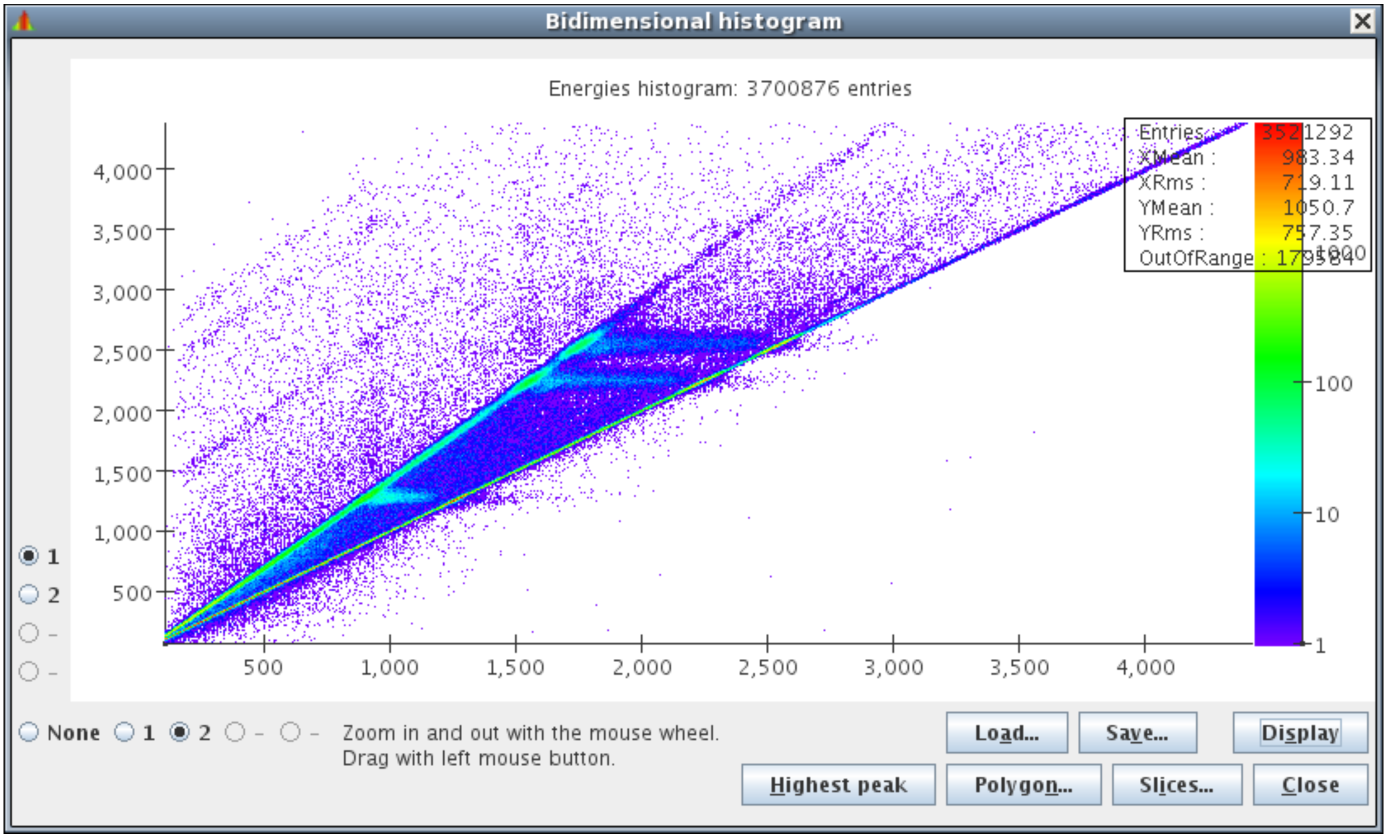
Bidimensional histograms

If two event channels are selected, a bidimensional histogram will be filled with one channel on each axis (X, Y). Two events make a plot when their timestamps are close enough (difference < 10). The two axis can be inverted with the radio buttons near the bottom left corner. Click on the **Display** button to see the events plotted as they are read from files.

Tip

An automatic refresh can be programmed by hitting 'p' when the **Display** button has the focus.

Figure 2.7. Bidimensional histogram



Energies histogram: 3700876 entries

```
Entries :      3521292
XMean :         983.34
XRms :          719.11
YMean :         1050.7
YRms :          757.35
OutOfRange :    179584
```

● 1
○ 2
○ -
○ -

○ None ○ 1 ● 2 ○ - ○ -

Zoom in and out with the mouse wheel.
Drag with left mouse button.

This histogram can be moved by dragging it with the left mouse button, and zoomed in and out with the mouse wheel.

It can be saved into a *.bidim* file with the **Save...** button then loaded again with **Load...**

Slices

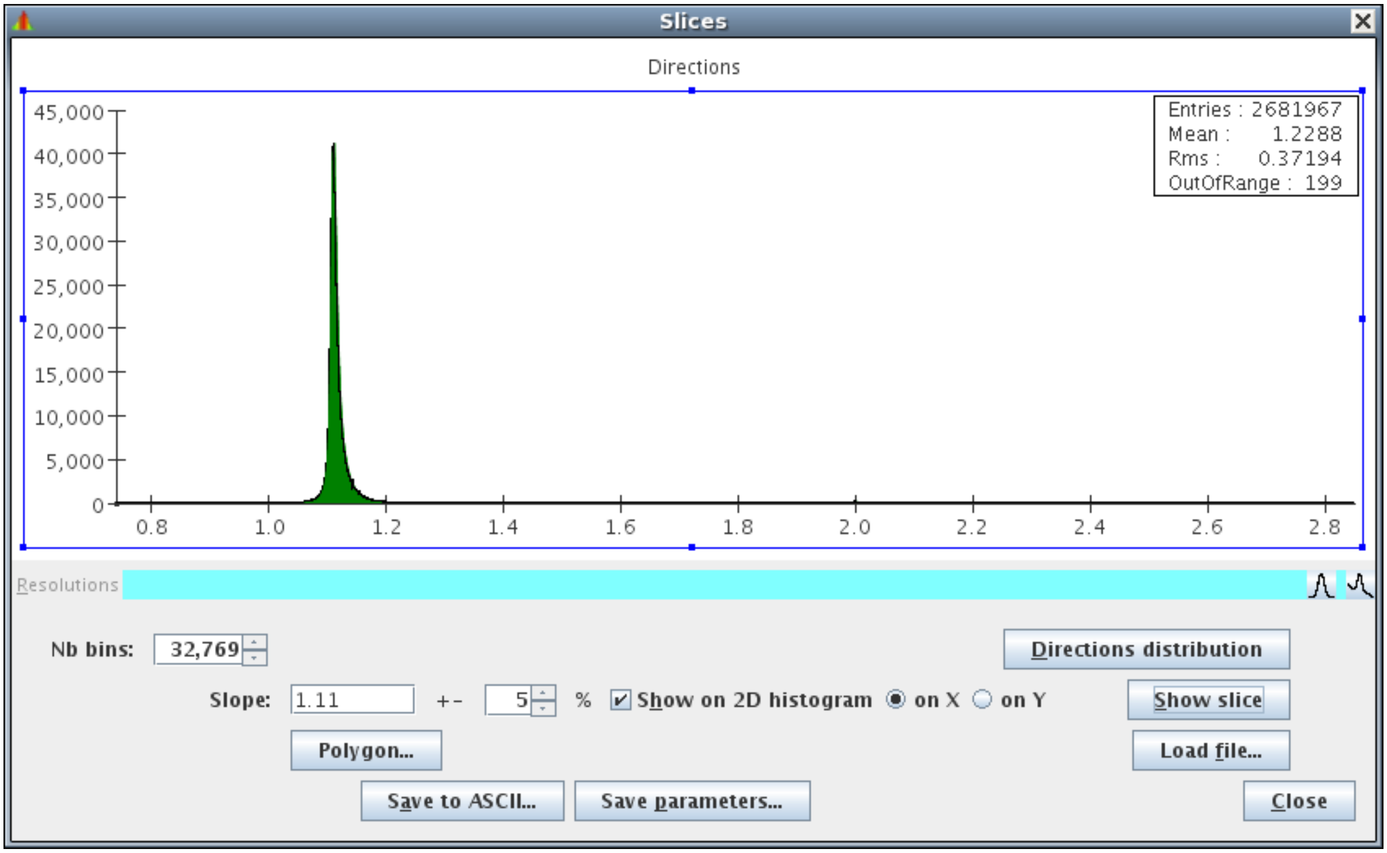
Parts of a bidimensional histogram can be projected on an axis or the other. As they often follow a line, they are names slices.

For example, in the figure Bidimensional histogram, we would like to select the lowest branch of the chart.

Click on the **Slices** button, then on **Directions distribution**. This will fill a 1D histogram with the directions of each point of the bidimensional histogram from the origin (i.e. the quotient X/Y) and allow to determine the slopes of the two branches, assuming that they contain the (0,0) point.

We can now enter the desired direction in the **Slope** edit zone (check **Show on 2D histogram** to display the slice limits in red above the bidimensional histogram), choose the projection axis (X or Y) and click on the **Show slice** button to fill a 1D histogram with the projection of the points from the selected area.

Figure 2.8. Slice from bidimensional histogram



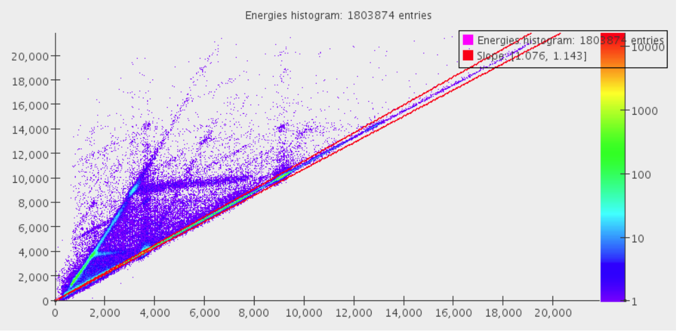
Nb bins: 32,769

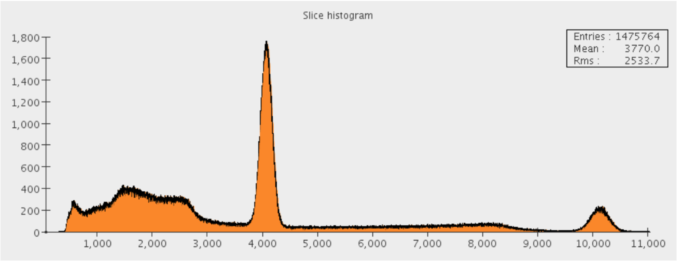
Slope: 1.11 +- 5 % ☑ Show on 2D histogram ● on X ○ on Y

This window also contains a blue bar which allows to add fit functions, calculate resolutions, and plot linearity charts or rescaled histograms.

From this window, ASCII histogram files (with extension *.asc*) can also be saved (**Save to ASCII** button) or loaded (**Load file** button).

The button **Highest peak** in the bidimensional histogram window automatically selects the slope with the highest number of points (+- 3 %) and projects its points on the X axis. The bidimensional histogram must be fully unzoomed (range: [0,32767]).



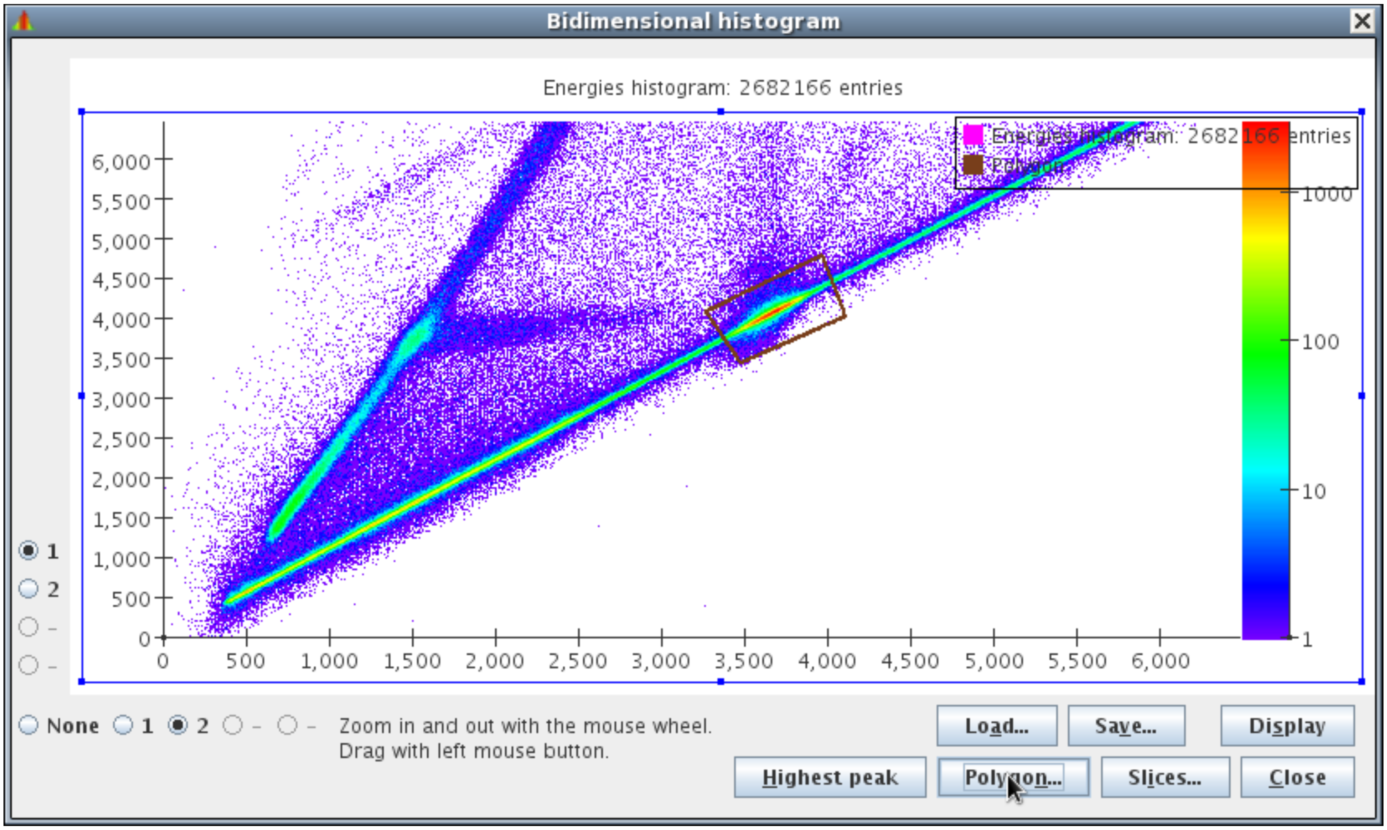


Polygons

An area of the bidimensional histogram can also be delimited by a polygon.

To create a polygon, click on the **Polygon** button then **Create** and follow the instructions. You will be asked to click on two points on the vertical and horizontal rules to calibrate the image on the screen then the points of the desired polygon then the **End** button. The polygon is now displayed in maroon colour and you can give it a file name (extension *.filter*). This polygon can be displayed on any bidimensional histogram (button **Polygon** then **Load**) and can be used from the **Slices** window to delimit an area instead of a slope.

Figure 2.9. Polygon creation



Zoom in and out with the mouse wheel.
Drag with left mouse button.

Chapter 3. Histogram data

An ASCII histogram file (extension *.asc*) is a text file with only a bin height per line. Example:

```
8680.0
8839.0
9044.0
9464.0
9578.0
9733.0
10003.0
9950.0
...
```

Click on the **Histograms** button of the main window to open one or more histogram files.

Figure 3.1. Histograms button

One window will be opened per file, each containing a blue bar to perform the usual operations on the 1D histograms.

Chapter 4. Bidimensional histogram data

The bidimensional histogram files (extension *.bidim*) contain 32768 x 32768 count numbers (one for each couple of energy values).

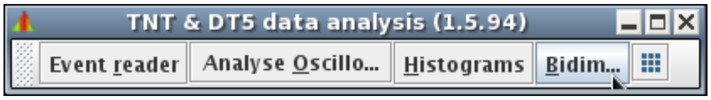
As one count number takes 2 bytes and to avoid handling 2GB files, the data is compressed into a ZIP archive.

The uncompressed data is sorted by columns.

Since TAN version 1.6, the count number values are stored over **3** bytes allowing them to reach 16 millions. TAN 1.6 and greater can handle both formats but do not try to read new format bidim files with an older version!

Click on the **Bidim...** button of the main window then choose a file to open a bidimensional histogram sub-window.

Figure 4.1. Bidim button



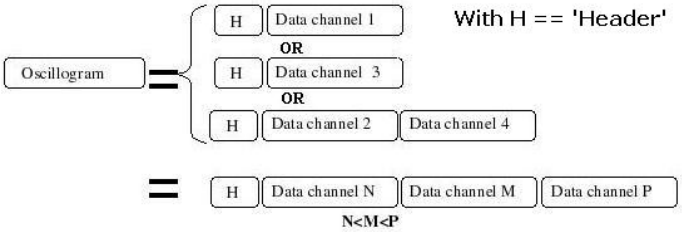
Chapter 5. Oscillogram data

Table of Contents

Oscillogram files format

Successive data files are structured in this manner:

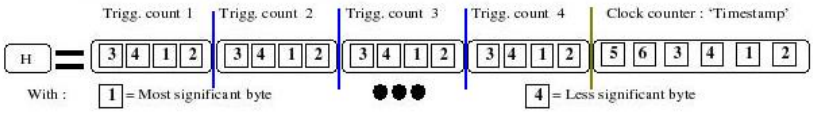
Figure 5.1. Oscillogram files format



The content of an oscillogram has 15 possibles configurations depending of which channels the user has asked (channels are arranged in ascending order)

An oscillogram starts always with the header informations. A header contains always the four trigger counters even if only one channel has been asked for readout. The header is 22 bytes long and contains 5 counter values:

Figure 5.2. Oscillogram header format



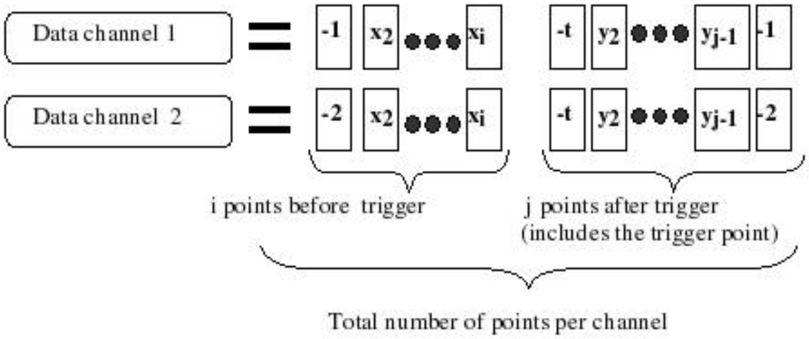
- Number of internal triggering for channel 1 (even if this channel has not been chosen for data readout). This counter is encoded using 4 bytes which gives a range from 0 to 4 294 967 295 (0xFFFF FFFF) on 10 digits
- Number of internal triggering for channel 2 (same as for channel 1)
- Number of internal triggering for channel 3 (same as for channel 1)
- Number of internal triggering for channel 4 (same as for channel 1)
- Number of ADC clock ticks since start of acquisition or since last reset (see parameters above). This counter is encoded using 6 bytes (48 bits) which gives a range from 0 to 281 474 976 710655 (0x FFFF FFFF FFFF) on 15 digits. Having a sampling period of 10 nanoseconds, this gives some maximal duration about 50 days before it will reset.

During an acquisition, the four trigger counters can easily be compared with the number of events which have been really readout during the same duration. It is then possible to get informations about how much events has been lost because of the dead time needed for reading out the previous events or because of pileups or ADC out of range. The ADC’s sample data on 14 bits, but a sample point is readouted over 16 bits (one word) by respecting these ranges of values :

- The bits b0-b13 contain the ADC value (between 0 and 16383) in a two-complement representation: positive values are from 0x0000 = 0 to 0x1FFF = 8191, negative from 0x2000 = 8192 = −8192(in signed 14 bits notation) to 0x3FFF = 16383 = −1(signed 14 bits notation)

- When the input signal is not in the normal ADC range [-1.1 V, +1.1 V], the card delivers some constant value over 14 bits plus the bit n◦ 15, b14, that will be set in order to notify this overflow state :
  - If the ADC is in positive overflow, the sampled point has the maximum 14 bit positive value with the 15th bit (bit b14) set to 1 : 0x5FFF = 24575
  - If the ADC is in negative overflow, the sampled point has the smallest 14 bit negative value with the 15th bit (bit b14) set to 1 : 0x6000 = 24576
- The bytes ordering type within a word is a Big Endian type: MSB, LSB
- The trigger point has his original value with the 16th bit (bit b15) set to one. Decoding whole 16 bits together, in a two’s complement notation, the trigger point would have a negative value.
- The first and last point of a channel have special mark values :
  - For channel 1 : 0xFFFF=65535 = -1 (in signed notation over 16 bits).
  - For channel 2 : 0xFFFE=65534 = -2.
  - For channel 3 : 0xFFFD=65533 = -3.
  - For channel 4 : 0xFFFC=65532 = -4.

Figure 5.3. Oscillogram channel data format



To summarize, decoding point values in a signed way over 16 bits, gives range as follows:

- 0 to 16383 : normal values
- 24575 : positive overflow
- 24576 : negative overflow

- -1 : first and last point of channel 1
- -2 : first and last point of channel 2
- -3 : first and last point of channel 3
- -4 : first and last point of channel 4

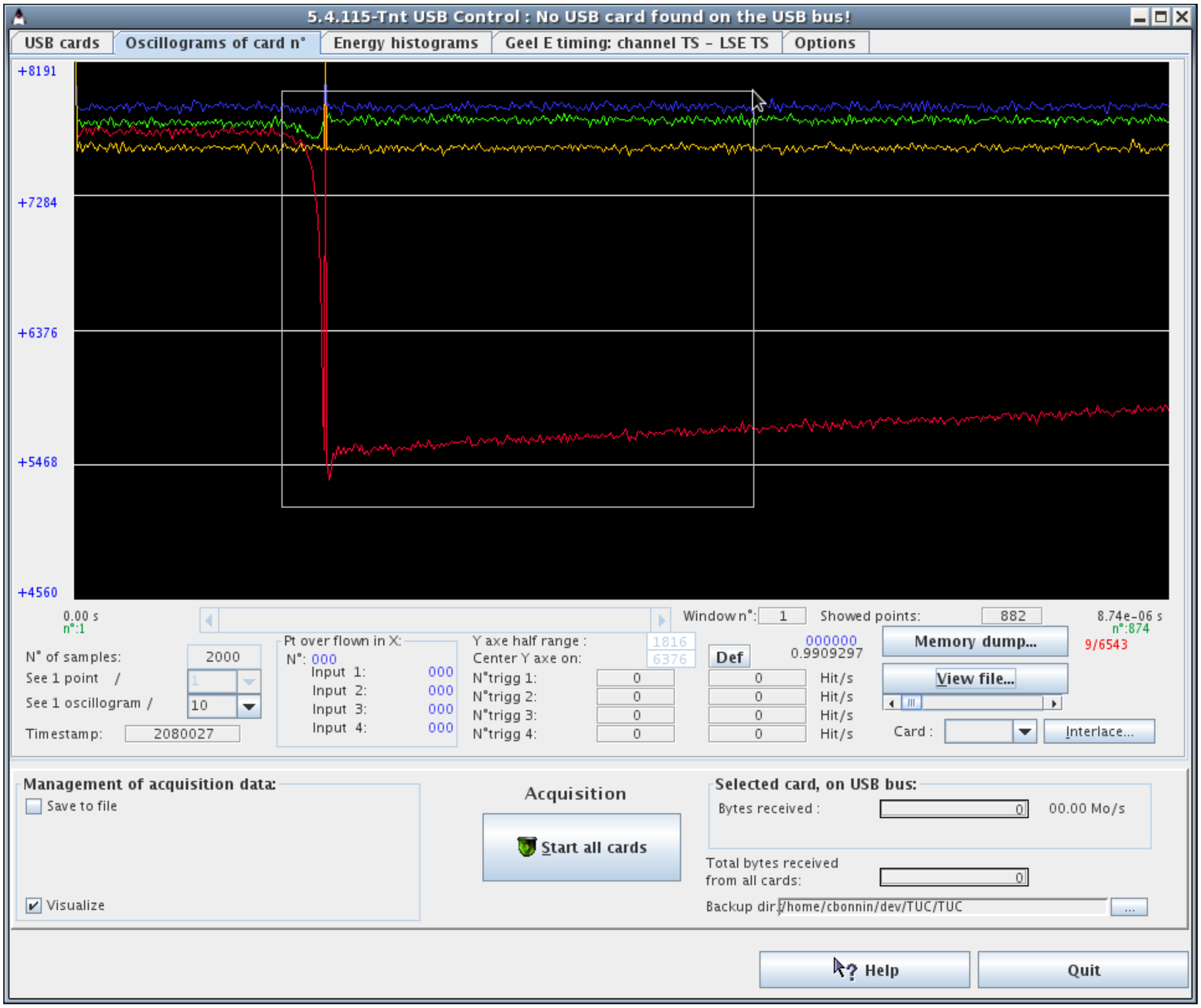
Any other negative point (only one per channel and per event) should be the trigger point.

Oscillograms visualization

The best way to visualize oscillogram files is by using the TUC 5 software as it allows to see up to 4 channels on the same screen.

In the **Oscillograms** tab, click on **View file** button and choose an *.osc* file.

Figure 5.4. Oscillograms visualization



0.00 s n°:1

Window n°: 1

Showed points: 882

8.74e-06 s n°:874

9/6543

N° of samples: 2000

See 1 point / 1

See 1 oscillogram / 10

Timestamp: 2080027

Pt over flown in X:
N°: 000
Input 1: 000
Input 2: 000
Input 3: 000
Input 4: 000

Y axe half range : 1816

Center Y axe on: 6376

000000
0.9909297

N°trigg 1: 0 0 Hit/s
N°trigg 2: 0 0 Hit/s
N°trigg 3: 0 0 Hit/s
N°trigg 4: 0 0 Hit/s

Card :

Management of acquisition data:
□ Save to file
☑ Visualize

Selected card, on USB bus:
Bytes received : 0 00.00 Mo/s

Total bytes received from all cards: 0

Backup dir: /home/cbonnin/dev/TUC/TUC

The scroll bar below the **View file** button allows to navigate through oscillograms in the whole file and the scroll bar just below the black screen is to scroll horizontally in the current oscillogram.

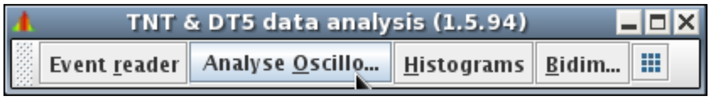
You can zoom by drawing a rectangle with the left mouse button then scroll vertically with the mouse wheel and horizontally with the mouse wheel while holding the SHIFT key down.

Click on the screen to cancel the zoom.

Oscillogram data processing

In the TAN software main window, click on the **Analyse oscillo** button.

Figure 5.5. Analyse oscillograms

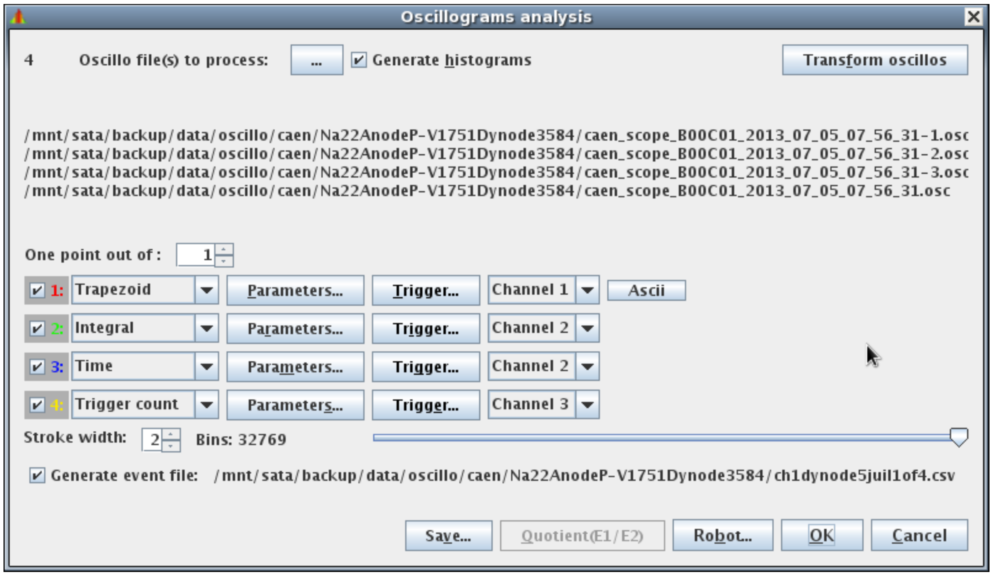


Note

If you want to calculate an ENOB from an oscillogram file, you will have to use an option of the Event Reader as it is an histogram functionality.

The first screen allows to select oscillogram files to be processed (button **"..."**)

Figure 5.6. Oscillogram analysis parameters



4 Oscillo file(s) to process: ☑ Generate histograms

```
/mnt/sata/backup/data/oscillo/caen/Na22AnodeP-V1751Dynode3584/caen_scope_B00C01_2013_07_05_07_56_31-1.osc
/mnt/sata/backup/data/oscillo/caen/Na22AnodeP-V1751Dynode3584/caen_scope_B00C01_2013_07_05_07_56_31-2.osc
/mnt/sata/backup/data/oscillo/caen/Na22AnodeP-V1751Dynode3584/caen_scope_B00C01_2013_07_05_07_56_31-3.osc
/mnt/sata/backup/data/oscillo/caen/Na22AnodeP-V1751Dynode3584/caen_scope_B00C01_2013_07_05_07_56_31.osc
```

One point out of : 1

☑ 1: Trapezoid — Channel 1

☑ 2: Integral — Channel 2

☑ 3: Time — Channel 2

☑ 4: Trigger count — Channel 3

Stroke width: 2 Bins: 32769

☑ Generate event file: /mnt/sata/backup/data/oscillo/caen/Na22AnodeP-V1751Dynode3584/ch1dynode5juil1of4.csv

Select **One point out of** N to divide the sampling frequency.

Up to 4 processes can be programmed on the oscillograms. They can be of type:

- Trapezoid: applies the Jordanov's algorithm [2] on oscillograms to compute energy values (like the algorithm used inside the FPGA on TNT cards)
- Integral: computes the sum of sampled values within given intervals.
- Trigger count: count trigged points according to given trigger parameters (this functionality is now better integrated in TUC 5 and named trigger tuning)
- Time: applies Bardelli's algorithm [4] to compute the precise time of the pulse

The values computed can be used to fill an histogram (**Generate histograms** check box). If it is the case, then the whole oscillogram files will have to be processed. The number of bins in the histogram can be tuned by setting the **Bins** slider.

If needed, an event file can be generated to store the computed values (**Generate event file** check box).

The event output format can be:

- .evt: TNT event file
- .dt5: DT5 event file containing TDC fields
- .csv: Comma Separated Values file with following fields:

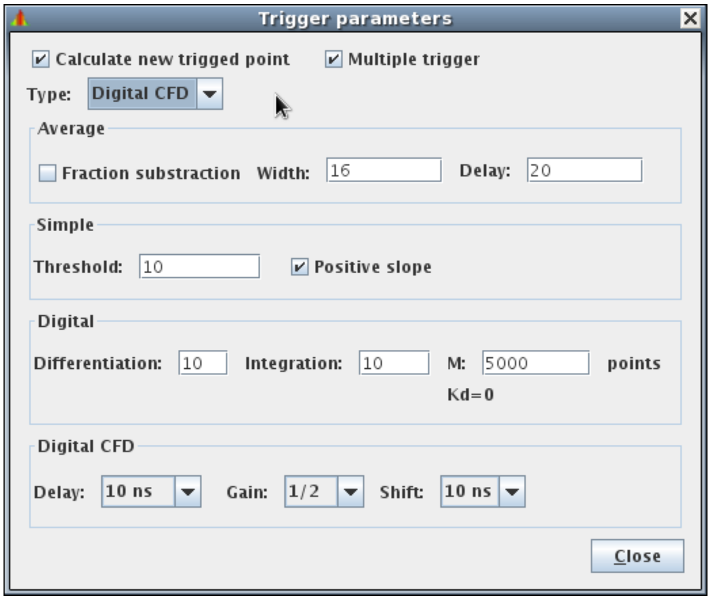
  ```
  <energy value> ; <timestamp>
  ```

Other curves can be drawn on the same screen as the input signal. Their width can be changed by the **Stroke width** editing zone (useful when taking screen shots).

Triggering

Each process has a **Trigger** button to specify parameters that can be used to re-trigger the oscillograms.

Figure 5.7. Trigger parameters



Trigger parameters

☑ Calculate new trigged point ☑ Multiple trigger

Type: Digital CFD

Average

☐ Fraction substraction Width: 16 Delay: 20

Simple

Threshold: 10 ☑ Positive slope

Digital

Differentiation: 10 Integration: 10 M: 5000 points

Kd=0

Digital CFD

Delay: 10 ns Gain: 1/2 Shift: 10 ns

To keep the point trigged by the acquisition card, just ensure that the **Calculate new trigged point** check box is not checked (by default).

Multiple trigger is to allow several trigged points in one oscillogram.

The parameter on this screen are those available in TUC 5 for the different types of triggers [2]:

- Simple: single comparison with a threshold value
- Digital: differentiation and integration stages then comparison with the threshold
- Digital CFD: same as Digital with a Constant Fraction Discriminator (CFD).

The **Average** frame is a test for a new trigger algorithm that better detects pile-up. Set the values to 0 to get the default algorithm.

Close the Trigger parameters window then click **OK** in the Oscillogram analysis window.

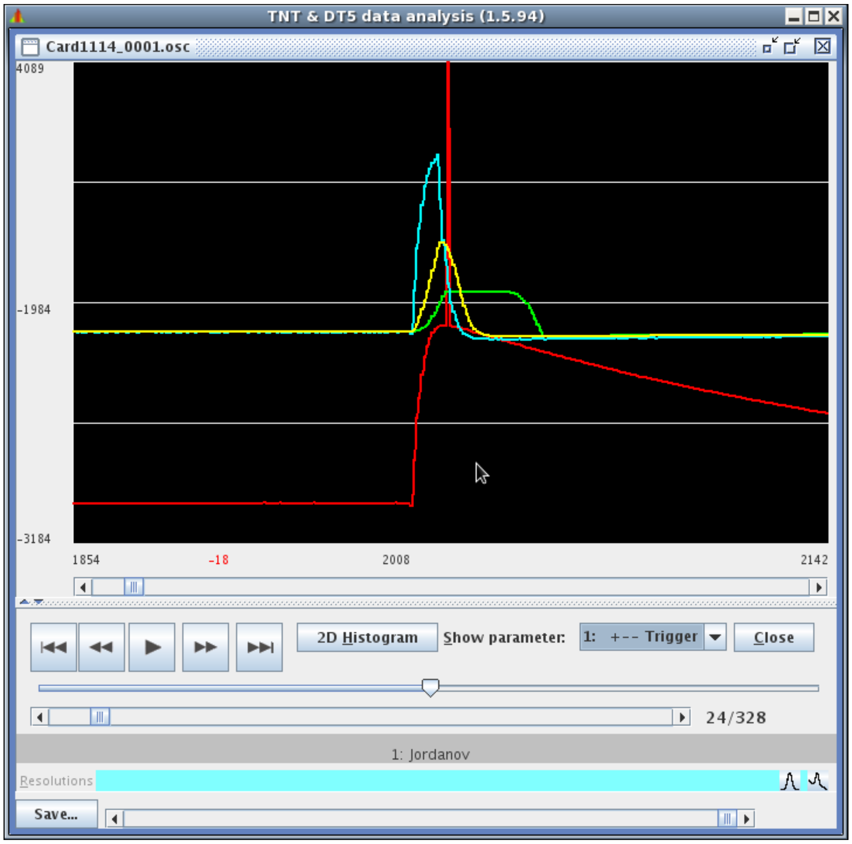
Oscillogram single channel visualization and processing result

This window allows to "play" oscillogram files by showing the successive oscillograms at different speeds.

The trigger algorithm has its intermediate curves that can be drawn above the input signal. To see them, select the Trigger line in the **Show parameter** combo box (Legend: Colors: Input, Differentiation, Threshold, TFA).

If you are not interested in filling the histogram, you can uncheck the **Generate histograms** check box in the previous dialog box and move down the separation line (split pane) that is below the first horizontal scroll bar to increase the size of the oscilloscope screen.

Figure 5.8. Trigger intermediate curves



The first horizontal scroll bar scrolls horizontally within the current oscillogram.

The square buttons respectively

- change to previous oscillogram file
- decrease the speed (duration between two oscillogram displays)
- play / pause the oscillogram succession
- increase the speed
- change to next oscillogram file

The horizontal slider changes the playing speed.

The second horizontal scroll bar (below the slider) navigates through the current file oscillograms. The current oscillogram number and the total number of oscillograms in the file is written at right hand.

By double clicking on this label you can load a CSV index file and sort the oscillograms by their computed value. The index file format is:

```
<value> ; <timestamp>
```

Such a file can be generated in the previous screen by giving a filename ending with ".csv" in the **Generate event file** field.

The **Show parameter** combo box contains for each parameter set <N> configured:

- one head line named *`Param <N>`* that displays the input signal (asked channel)
- one line *`<N>: <type>`* where <type> is one of (Jordanov, Integral, Trigger count) which displays Jordanov's trapezoid or integration areas
- one line *`<N>: +-- Details`* which displays intermediates curves
- one line *`<N>: +-- Trigger`* which displays the trigger intermediate curves if the trigger is enabled (via the **Calculate new trigged point** check box) for this parameter set.

If asked, the histogram at the bottom of the screen is filled with the computed values for all oscillograms of all files.

To temporarily modify a parameter, it is possible to click on the **Show parameter** label. The displayed curves will be affected but not the already filled histograms.

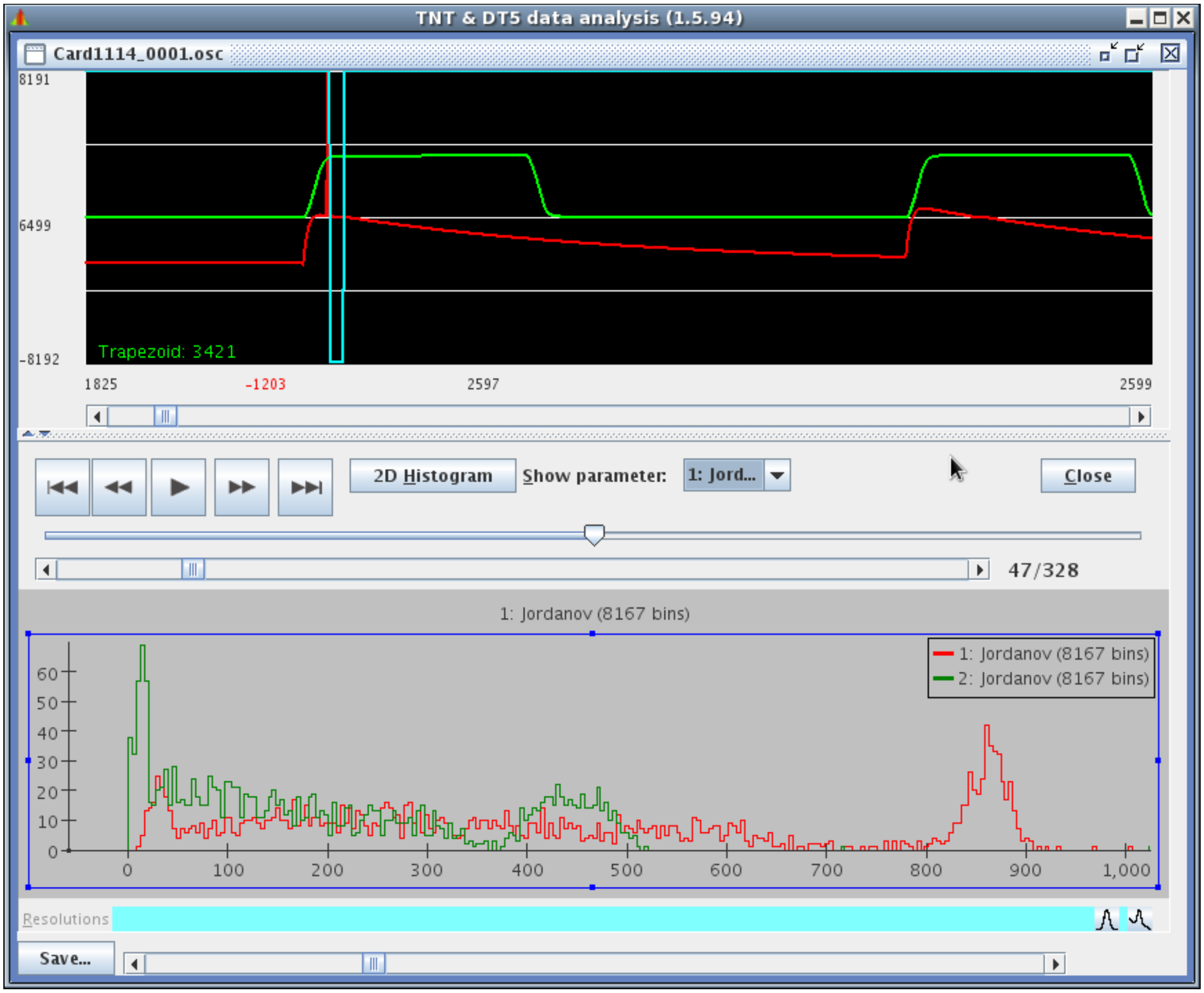
A blue bar allows the usual operations on the histogram.

When at least two parameter sets are asked, it is possible to fill a bidimensional histogram by hitting the **2D Histogram** button.

Jordanov's trapezoid algorithm

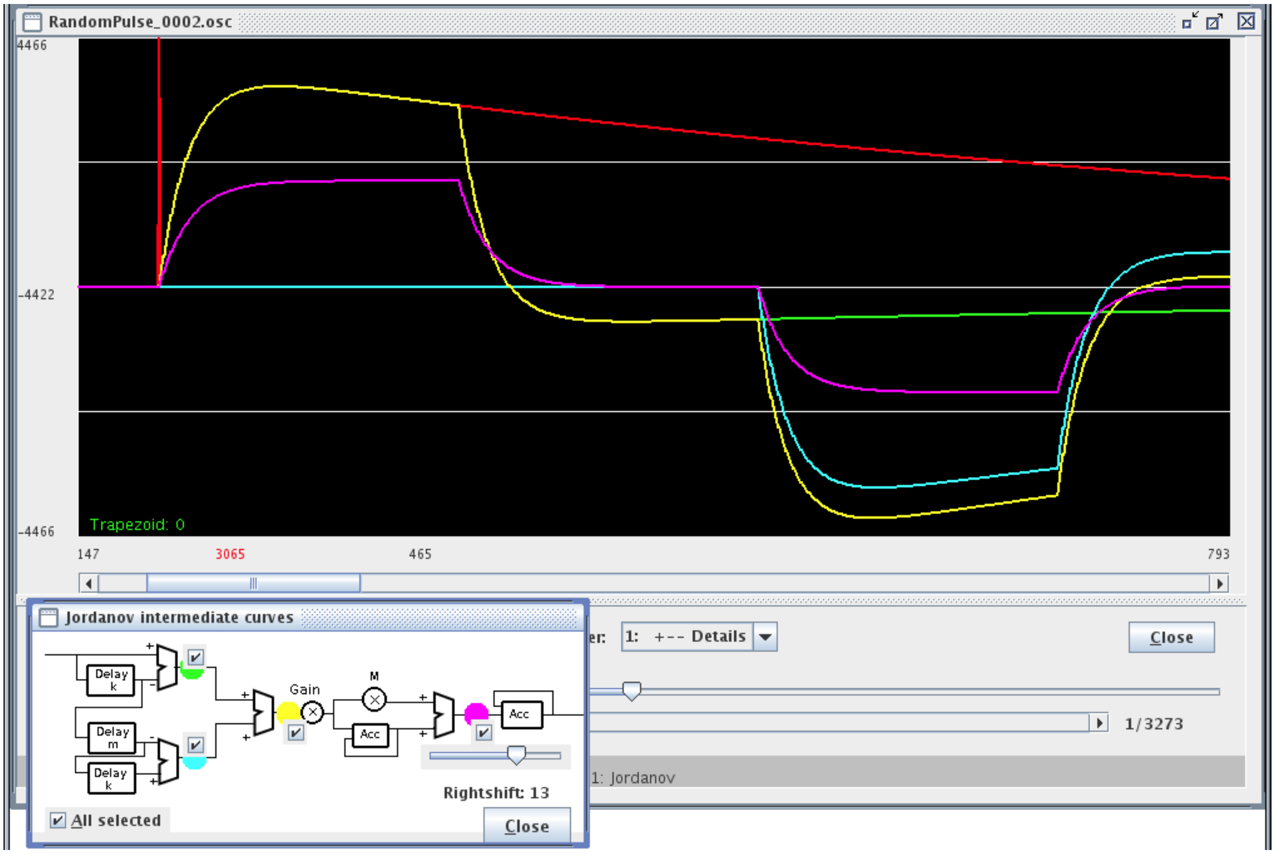
On this example, two Trapezoid where asked on the same input signal. One of them is drawn in green and the input signal in red. The blue lines represent the moving window where the average energy value is computed by Jordanov's algorithm.

Figure 5.9. Histograms filled with trapezoid values



You can also display Jordanov's trapezoid intermediate curves by selecting the `<N>: +--Details` line in the **Show parameter** combo box.

Figure 5.10. Jordanov's trapezoid intermediate curves



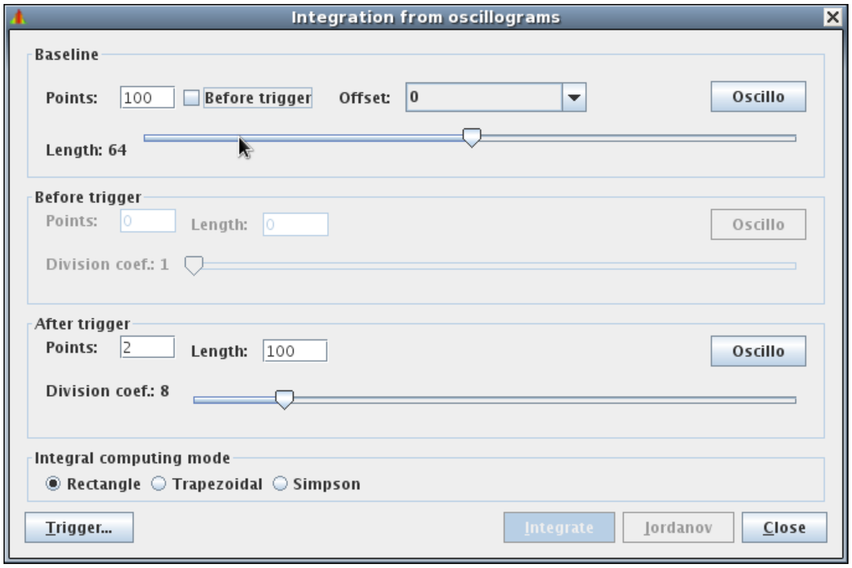
A small dialog box appears that allows to select the displayed curves and gives a legend for different colours.

The **Rightshift** slider is a way to divide the last step values so that they can be displayed on the same screen.

Integral

The Integral parameter type allows to configure an integration area where the sampled values will be summed.

Figure 5.11. Integral parameters



Baseline

Points: 100 □ Before trigger Offset: 0

Length: 64

Before trigger

Points: 0 Length: 0

Division coef.: 1

After trigger

Points: 2 Length: 100

Division coef.: 8

Integral computing mode

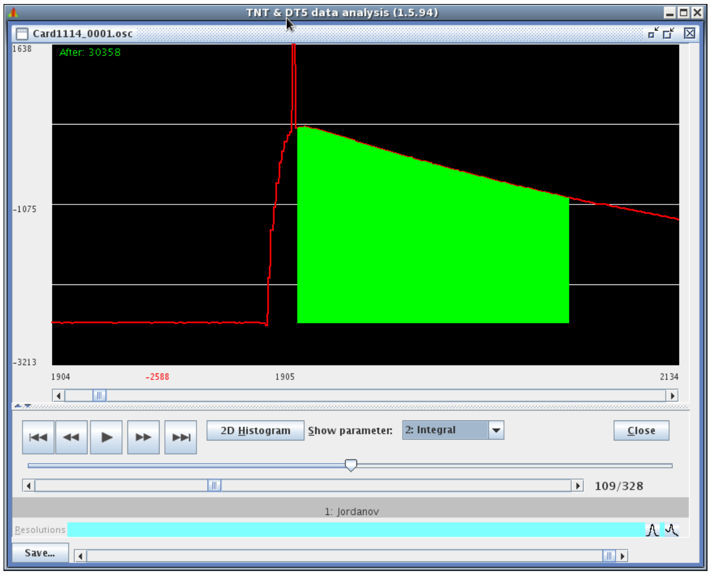
● Rectangle ○ Trapezoidal ○ Simpson

The **Baseline** frame configures the position and the length of the area where the baseline is computed.

The **After trigger** frame configures the position and the length of the integration area. The number of points is counted after the trigged point.

The **Division coefficient** is a power of 2 and is used to divide the sampled points sum. The result is not actually an average because the FPGA performs only right shifts instead of integer division to spare its resources.

Figure 5.12. Displayed integration area



After: 30358

Show parameter: 2: Integral

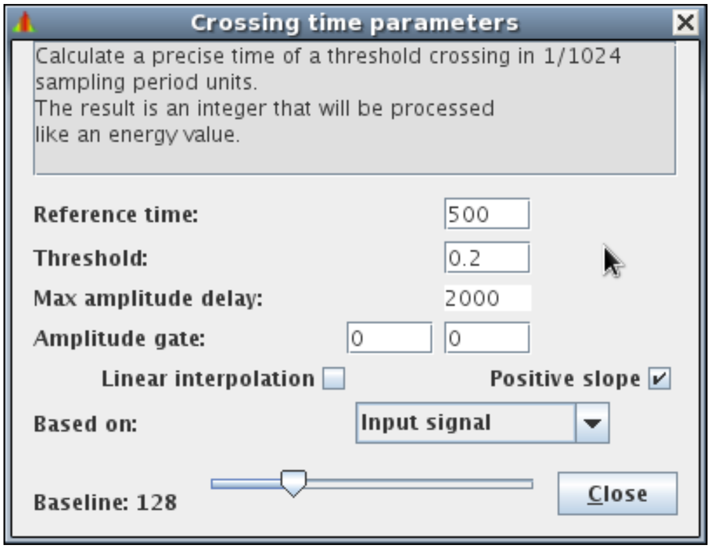
109/328

1: Jordanov

Crossing time

The precise time (in 1024th of the sampling period) of the pulse is computed by determining the crossing time of a polynomial interpolation of the input signal with a threshold which is a fraction of the pulse maximum amplitude.[4]

Figure 5.13. Crossing time parameters



Calculate a precise time of a threshold crossing in 1/1024
sampling period units.
The result is an integer that will be processed
like an energy value.

Reference time: 500

Threshold: 0.2

Max amplitude delay: 2000

Amplitude gate: 0 0

Linear interpolation □ Positive slope ☑

Based on: Input signal

Baseline: 128

The **Reference time** is the approximate time in sampling period of the crossing. The aim is to avoid the result to be greater than 32767

Threshold contains the fraction of maximum amplitude used for crossing calculation

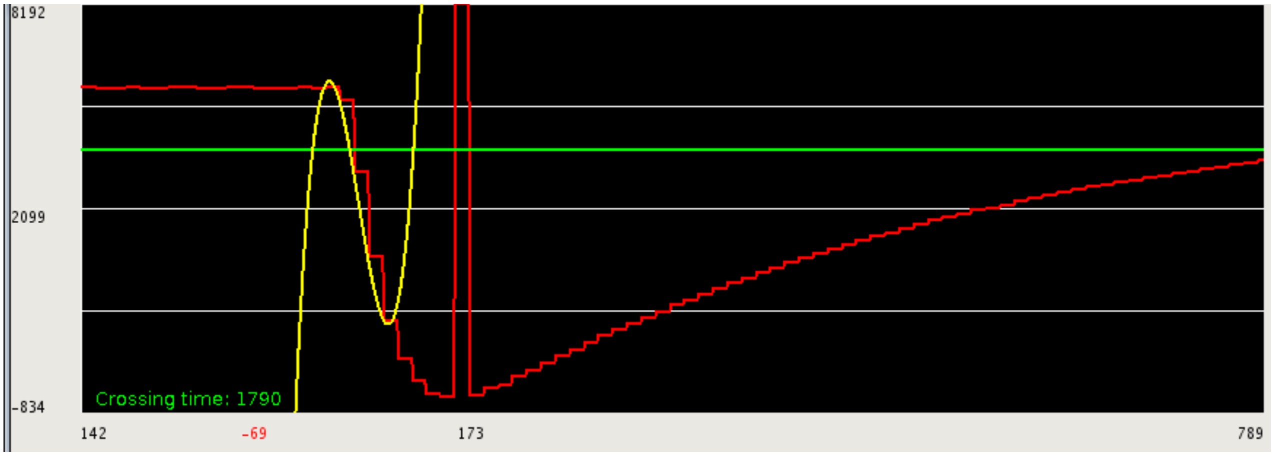
Max amplitude delay: Delay after trigged point used to search for the maximum amplitude

The **Amplitude gate** is an amplitude filter (minimum and maximum) for the pulse to be considered

If **Linear interpolation** is checked, a line will be used instead of a 3rd degree polynom to interpolate the input signal

Baseline: Number of baseline correction points to be considered from the beginning of oscillograms

Figure 5.14. Crossing time computing



The threshold is in green and the polynomial interpolation in yellow

Robot: histograms for parameter values varying in a range

The Robot functionality (from oscillogram analysis) can generate automatically a list of histograms by taking the values of a given parameter in a range.

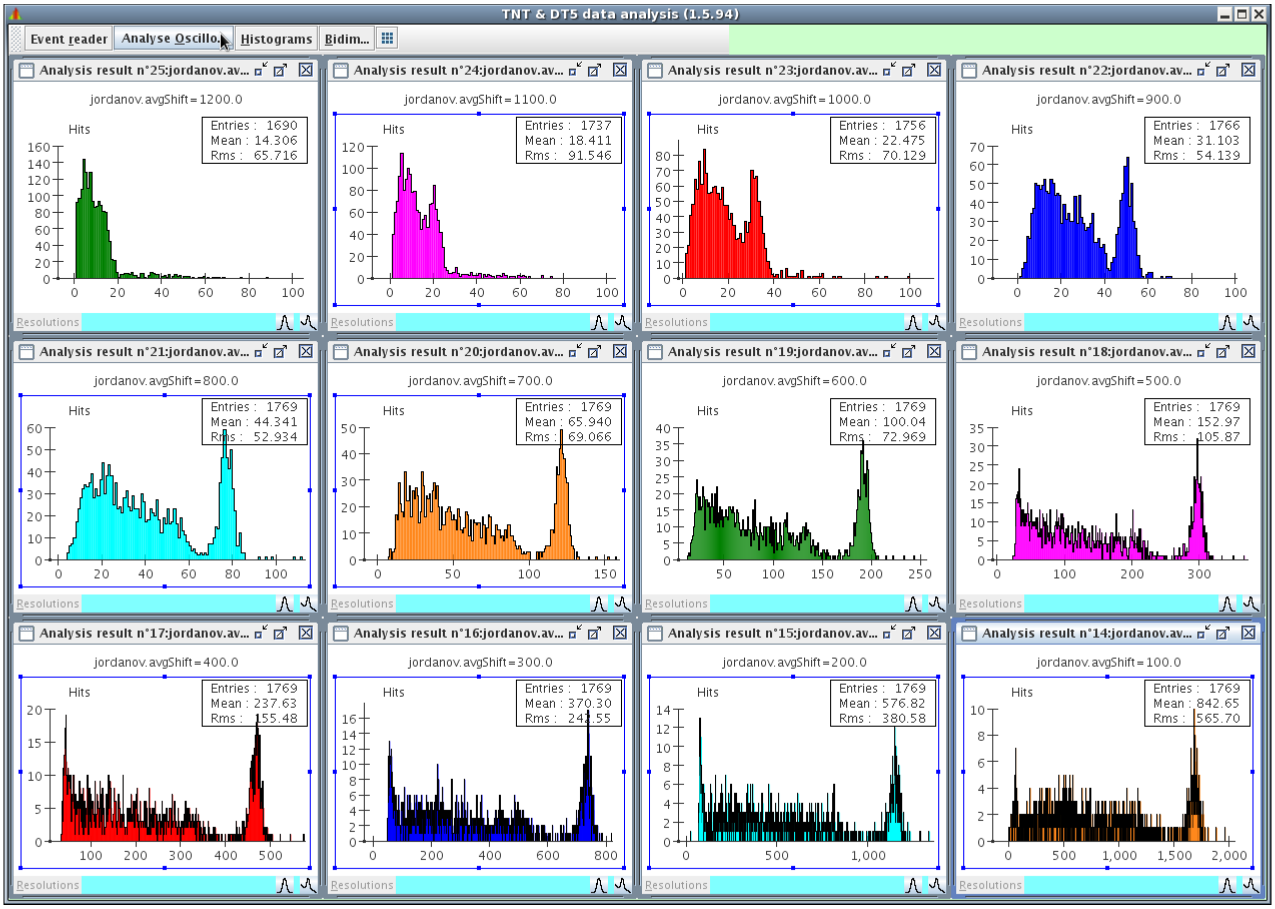
The **Robot** button asks for a string like `<parameter>.<path>=<begin>:<step>:<end>` where:

- <parameter> is one of Jordanov, Integral or Trigger
- <path> is a variable name
  - For Jordanov, it can be: k, m , M, avgShift, avgWidth, baselineCorrection, gain, xFactor, begin
  - For Integral, it can be: iBaseline (base line length), iBaselinePt (base line position), iAfterPt, iAfterLen, iAfterCoef
  - For Trigger, it can be: delay, gain, shift, differentiation, integration, threshold, avgWidth, avgDelay, decayConstant
- <begin> is the beginning of the range
- <step> is the interval between two values
- <end> is the end of the range

The following example makes the Jordanov's trapezoid **Average shift** take the values 100, 200, 300, ..., 1200. The other variables remain at their specified value.

```
jordanov.avgShift=100:100:1200
```

Figure 5.15. Automatic generation of histograms for different values of avgShift

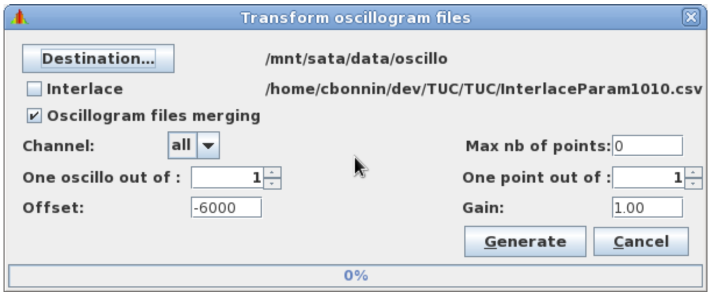


Oscillogram files transformations

The **Transform oscillos** button allows to convert oscillogram files into other oscillogram files with extension .osc and TUC oscillo format

The source files are those selected in the previous screen. They can be .osc files of CAEN binary .bhx files.

Figure 5.16. Transformation of oscillogram files



Transform oscillogram files

Destination... /mnt/sata/data/oscillo

☐ Interlace /home/cbonnin/dev/TUC/TUC/InterlaceParam1010.csv

☑ Oscillogram files merging

Channel: all

Max nb of points: 0

One oscillo out of : 1

One point out of : 1

Offset: -6000

Gain: 1.00

The destination directory is chosen with the **Destination** button.

The files can be interlaced off-line using TUC interlace parameters CSV files.

If a **Channel** is selected, only it will be taken into account.

The user can choose to keep only **one oscillo out of** N to reduce the size or keep **one point out of** N to reduce the sampling frequency

The **Rightshift** is a way to divide or multiply (with a negative number) the sampled value by applying a power of 2 gain.

The oscillograms size (**Max nb of points**) can be limited. This is necessary when converting CAEN binary files (*.bhx).

Oscillogram files merging

This checkbox allows to merge oscillogram files from several cards into one file as if they were from one single card.

The first oscillo from each card gives its offset which is subtracted from the timestamps. Thus the timestamps of the result file are sorted.

Example:

Figure 5.17. Timestamps from source oscillo files to merged result

	Source files				Result file
	Card 1001	Card 1002	Card 1010	Card 1013	
timestamps	1	2	3	4	0
					0
					0
					0
	25	11	23	14	9 (11 - 2)
					10 (14-4)
					11 (15-4)
					20 (23-3)
	28	31	24	15	21 (24-3)
					24 (25-1)
					27 (28-1)
					29 (31-2)

The Most Significant Byte of the fourth trigger counter of each oscillogram header is replaced by a number assigned to each card in the merging process to know which card it is from.

Bibliography

[1] Christian Bonnin. Marc Richer. Cayetano Santos. *TNT2 Digital Pulse Processor Functionalities & TUC control software*.

[2] Valentin T. Jordanov and Glenn F. Knoll. *Digital synthesis of pulse shapes in real time for high resolution radiation spectroscopy*. Nucl. Instr. Meth. A345. 337-345. (1994).

[3] P. Medina et al. *TNT Digital Pulse Processor*. Nuclear Science, IEEE Transactions.

Vol.53. Issue 3. 723-728. (2006). 10.1109/TNS.2006.873712.

[4] L. Bardelli et al. *Time measurements by means of digital sampling techniques: a study case of 100 ps FWHM time resolution with a 100 MSample/s, 12 bit digitizer*. Nucl. Instr. Meth. A521. Vol.521. Issue 2-3. 480-492. (2004). 10.1016/j.nima.2003.10.106.