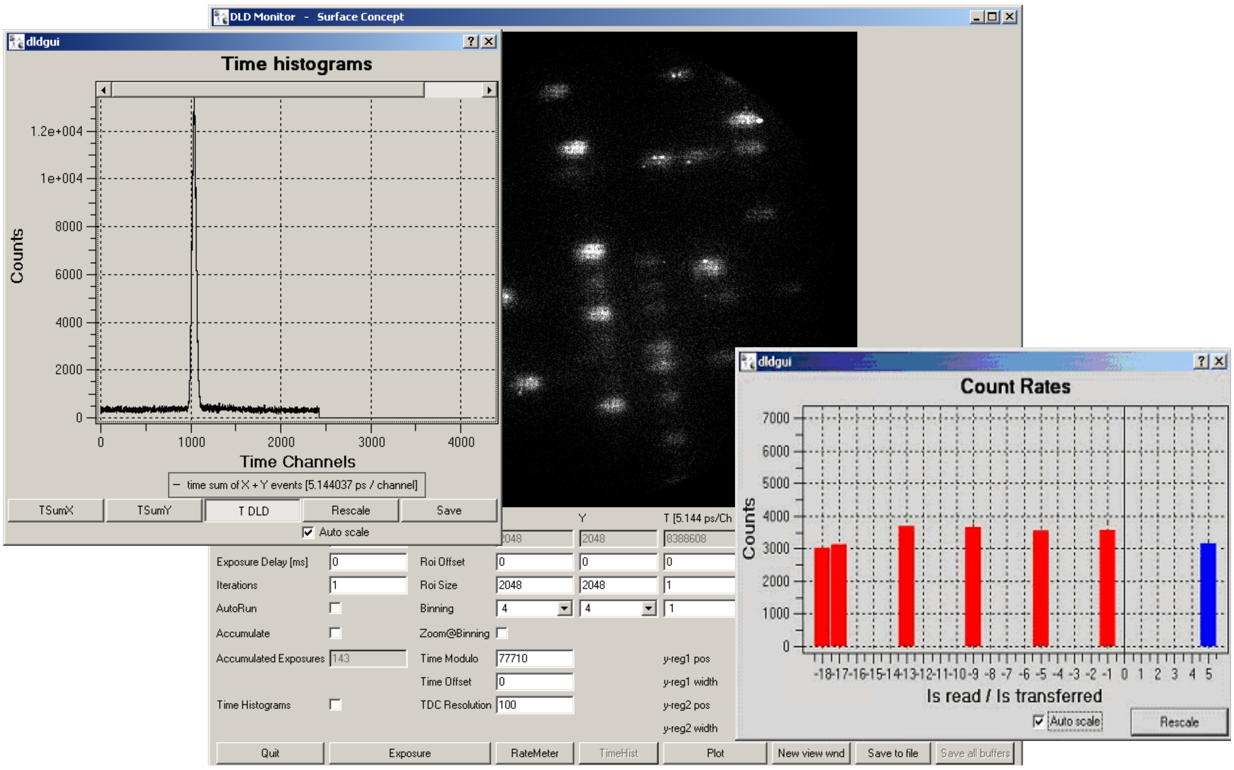
GUI - DLD Software



Manual

Surface Concept GmbH

Am Sägewerk 23a
55124 Mainz
Germany

Tel. ++49 6131 62716-0
Fax: ++49 6131 62716-29

www.surface-concept.com,
support@surface-concept.de

Manual Version: 7.4

Date: 2012-03-19

1 Table of Content

2 GUI monitor software

2.1 Software Installation and Hardware Requirements

There is no installation needed for the GUI software. Copy the "GUI software"-folder to your hard disc. To work with the GUI monitor, the following system requirements are highly recommended:

- Processor: 1.6 Ghz
- RAM: 1GB
- Windows 2000 / Windows XP / Windows Vista 32bit
- USB2.0 (no front panel connector)
- Monitor resolution: in Y min. 864 pixels (most critical), in X min. 1024 pixels

All software parameters are set for a prompt image acquisition.

2.2 Readout Modes of the TDC & Pair Mode Data Formats

The TDC can be operated in two different readout modes: the RawData Mode and the Pair Mode. To switch between the two modes, change the switch "iPairNumber" in the dld_gpx3.ini file (see Figure 1). The dld_gpx3.ini file is part of the DLD software and is located in the GUI main folder.

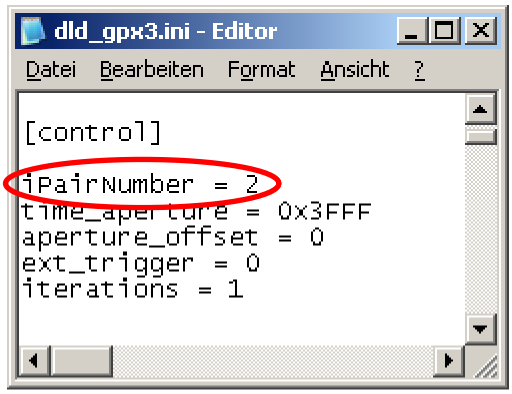


Figure 1: Screenshot of the dld_gpx3.ini file (opened with the MS text editor). The switch "iPairNumber" can be changed between RawData mode (=0) and Pair mode (=2).

If the iPairNumber is set to 0, the readout mode of the TDC is the RawData Mode. In this case the results of all DLD readout channels are transferred unprocessed to the PC. All data processing is done by the software

on the PC. For example calculations of the time differences and the time sums that form the X- and Y- position of the DLD events. For operations like binning or RoI settings the calculations are also done by the software. In the RawData Mode a maximum count rate of about 1.8 MCPS is possible due to the limitation of the USB transfer rate.
To reduce the data load on the USB, the data processing can be done by the FPGA of the USB2.0-TDC. This is the so called Pair Mode. Switch to the Pair Mode by setting iPairNumber = 2.
The calculations of the time differences and the time sums as well as the binning and settings of RoIs are now made within the FPGA before the data is transferred to the PC.
The Pair Mode restricts the usable value range for time measurements, depending on the data format. The data format can be set by the switch "Data_Format" within the dld_gpx3.ini file. There are 4 different data formats:

	dynamic range for x and y	dynamic range for t
Data_Format = 0	9 bit/ 9 bit (≡ 512x512 pixels)	14 bit (≡ 16384 time channels)
Data_Format = 1	10 bit/10 bit (≡ 1024x1024 pixels)	12 bit (≡ 4096 time channels)
Data_Format = 2	11 bit/11 bit (≡ 2048x2048 pixels)	10 bit (≡ 1024 time channels)
Data_Format = 3	12 bit/12 bit (≡ 4096x4096 pixels)	8 bit (≡ 256 time channels)

Note

Save the ini file after each modification and restart the GUI software for the changes to take effect.

2.3 Start and Quitting of GUI Software

Turn on the USB2.0-TDC before starting the GUI. To start the GUI software double-click the dldgui.exe file in your "GUI software" folder. The GUI software opens up with its Main Interface as shown in Figure 2. To end the program, click the "Quit" button.

Note

The USB2.0-TDC must be turned on, before the GUI is started.

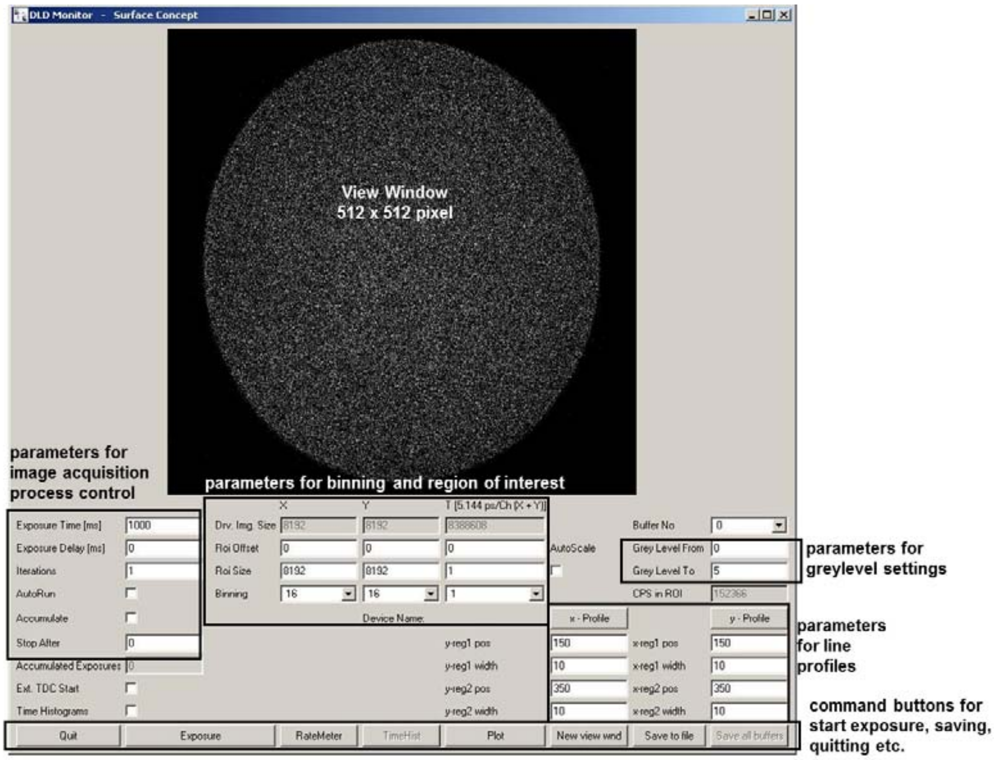


Figure 2: GUI Main Interface

2.3.1 GUI Main Interface for small PC Monitors

Some PC monitors do not reach the minimum resolution needed, see chapter 2.1. In this case the GUI Main Interface is not displayed completely and some of the buttons and entry fields cannot be accessed. In this case it is possible to open a smaller GUI Main Interface. This modified interface is reduced in size by removing some special advanced functions and entry fields. All basic functions for standard operation are still available.

The corresponding switch to change between the normal and the modified GUI Main Interface is in the dldgui.ini file, located in the main folder of the GUI software. Open the dldgui.ini file with a text editor software and change the value of the parameter "small_screen" from "0" to "1" (see Figure 3).

Note

Avoid adding any other characters to the dldgui.ini file, specially no additional space characters.
After modification of the dldgui.ini file, the file has to be saved and the GUI software must be restarted (if already in operation).

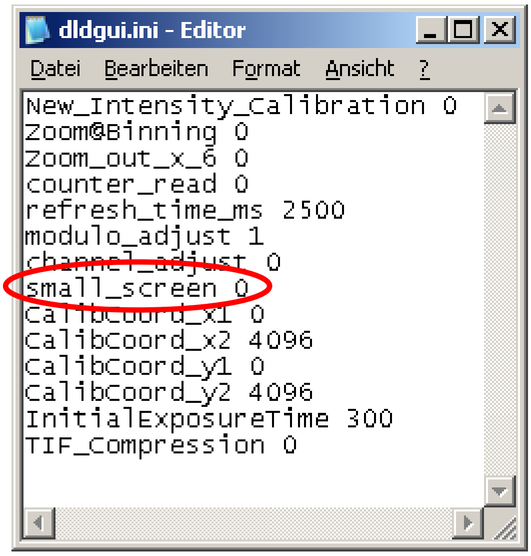


Figure 3: screenshot of the content of the dldgui.ini file (opened with the MS text editor). The switch for changing the size of the GUI Main Interface is marked red.

The GUI software should open up now with the modified Main Interface as shown in Figure 4.

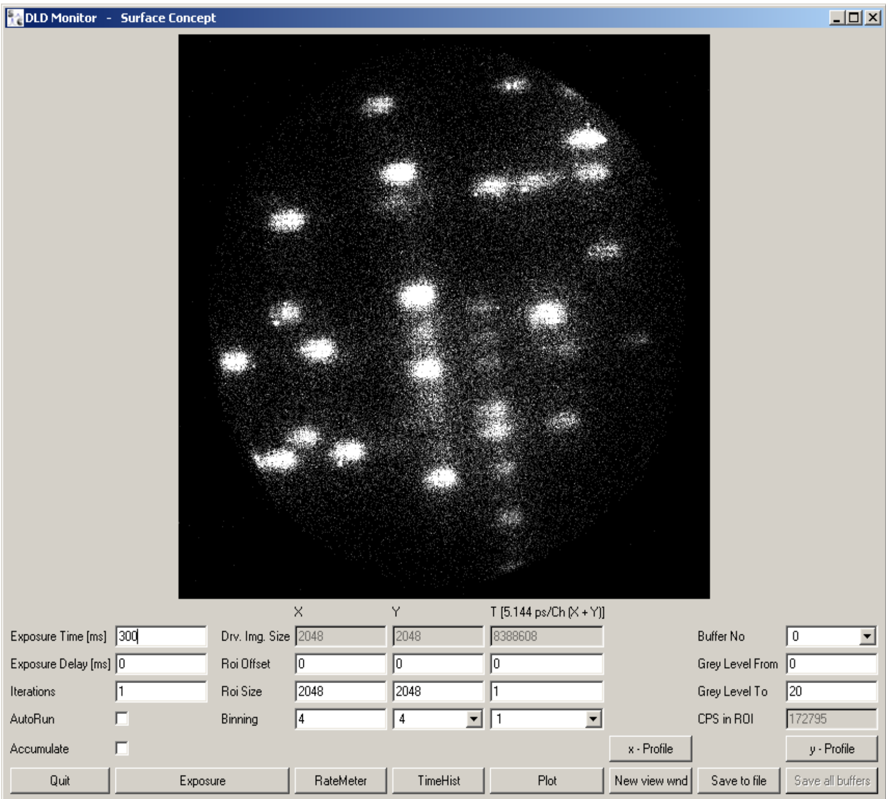


Figure 4: Modified GUI Main Interface

2.4 First Image and basic acquisition functions

2.4.1 First Image

The software acts similar to a CCD camera monitor respective to image acquisition, real-time monitoring, exposure times, pixel binning, regions of interest etc.

Press the exposure button to start the image acquisition.

The exposure time is set to 300 ms by default. It can be changed in the field “Exposure Time”. The exposure time is given in units of ms.
The number in the field “Exposure time” is counting down to indicate the acquisition process, after the exposure has been started.
The exposure time can be varied from 11ms up to 1193h.
The default value of the exposure time can be defined in the dldgui.ini file (see chapter 3 for details).

Note

The elapsing exposure time as displayed during acquisition is based on a software timer and is not very accurate. It is meant to indicate the exposure process in the first place.
This elapsing time cannot be compared with the real exposure time of the measurement, which is high precisely quartz-controlled by the measurement hardware.

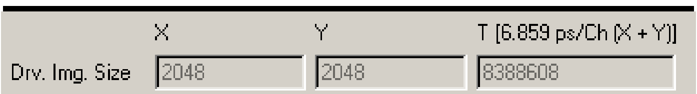
2.4.2 View Window and Image Size

An acquired image is displayed in the view window in the top center of the GUI main interface. The size of an image is either defined by the data format, when operating the TDC in Pair Mode (see chapter 2.2), or by the variables named “image_size_x” and “image_size_y” within the dld_gpx3.ini file, when operating the TDC in RawData Mode.

Note

The data format and the entries for image_size_x and image_size_y are adjusted to fit to the active area of the DLD. <u>Do not change these values.</u>

The image size is displayed in the fields named “Drv. Img. Size X, Y, T” of the GUI Main Interface.



The size of the view window of the GUI Main Interface is fixed to 512x512 pixels. In most cases, the real image size is much larger (depending on the active area of the DLD). In such a case, only the 512x512 pixels around the acquired image center are displayed and scrollbars appear on the left and top side of the view window (see Figure 5). Use them to scroll around. The binning function (see chapter 2.4.11 for further details) can be used, to display the full image in the view window.

scrollbars

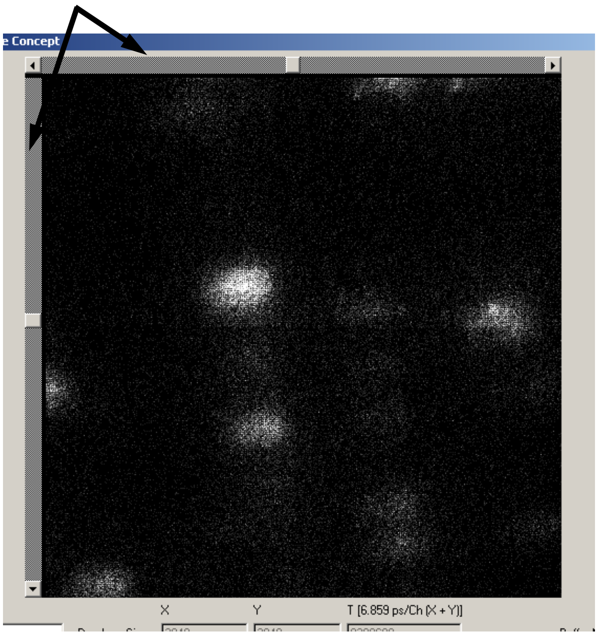


Figure 5: View window with scrollbars. Scrollbars appear when an acquired image is larger than the view window size.

Note

The full image is always acquired and saved independently, even if the image size is larger than the view window in the GUI.

2.4.3 Image Zoom

It is possible to zoom in and out of that part of the image which is shown in the view window. Use the mouse wheel or PageUp/PageDown to zoom in and out. It is also possible to zoom only in 1 dimension (x or y). Use the mouse wheel and "ctrl" to zoom in x direction and use the mouse wheel and "shift" to zoom in y direction.

2.4.4 Image Origin

The image origin (0/0 position) is located in the upper left corner of an acquired image (see Figure 6).

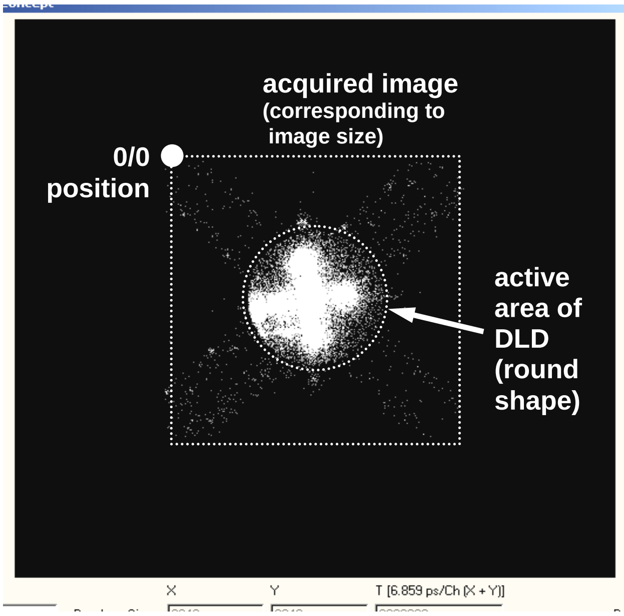


Figure 6: 0/0 position of an image. Image acquisition with the image size of 2048x2048 pixels and a binning of 8 in X and Y each.

2.4.5 Image Refresh Time

Image refreshing occurs after the acquisition has ended and in case that the exposure time is larger than 2500 ms, after every 2500 ms. This refresh time of 2500 ms is a default setting. It can be managed by changing the value of the variable named “refresh_time_ms” in the dldgui.ini file (see Figure 3 as well as chapter 3 for further details.

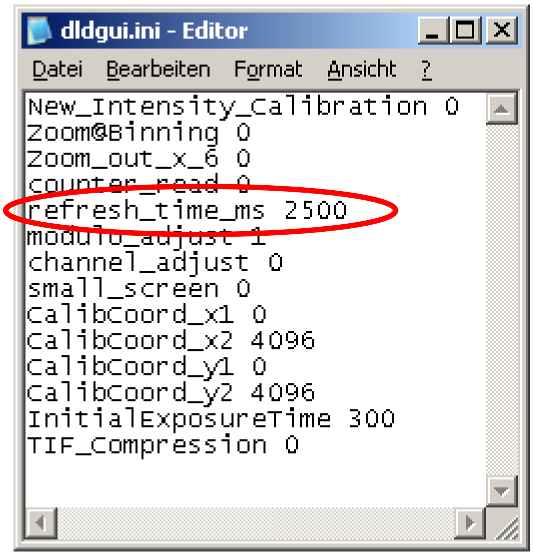


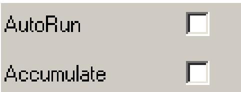
Figure 7: screenshot of contents of the dldgui.ini file (opened with MS text editor). The parameter for setting the refresh time is marked red.

2.4.6 Stop Exposure

The name of the button "Exposure" changes to "Stop Exposure" after the first image update. The exposure can be stopped by clicking on "Stop Exposure". The current image is displayed.
If the image appears overexposed or too dark, change the image dynamic range as mentioned in chapter 2.4.8.

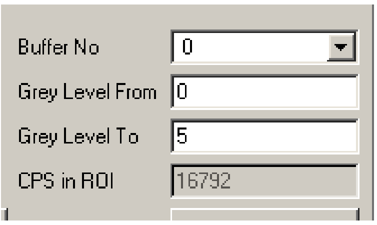
2.4.7 AutoRun Mode

In "AutoRun"-mode the acquisition process will restart automatically each time the exposure time has ended. It is turned on/off by clicking the box named "AutoRun" (turned on when box is ticked). AutoRun Mode is turned on as a default setting.



AutoRun can be turned off any time during the acquisition process and the image acquisition ends when the exposure time is over or "stop exposure" is clicked. The latest image is then displayed.

2.4.8 Dynamic range of displayed image



The grey level of each pixel within an image corresponds directly to the number of counts in this pixel measured during the acquisition time.
The default range of the displayed grey levels is set from 0 to 5 (fields "Grey Level From" and "Grey Level To"). Depending on the number of detected counts during an acquisition, the displayed image can appear overexposed. In this case you can change the displayed grey levels respectively by entering higher or lower values in the "Grey Levels" input fields. The displayed grey levels can be changed during and after the acquisition. In case that it must be changed after an image was taken, press the "Plot" button for the changes to take effect. If the values are changed during acquisition, then the grey levels of the image are updated with the next image.

Note

Changes in the displayed grey levels only affect the way the images are displayed. They do not affect the dynamic range of the images in which they are recorded. Images are always saved with a dynamic range of 16 bit.

2.4.9 The Plot Button

In nearly all cases, changes to the image display (e.g. dynamic range, zoom@binning) can be made during as well as after image acquisition. During image acquisition the changes take place with the next image update. After image acquisition, the "Plot" button has to be pressed additionally for the changes to take place.

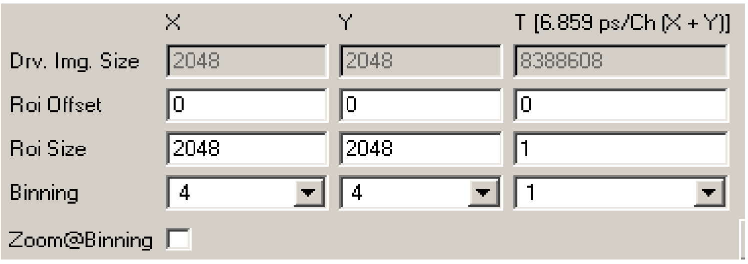
2.4.10 Save Images

To save an image, click the button “Save to file”. A standard Microsoft® window opens. Select the folder for the image and enter a filename.

Note

Images are saved as 16-bit tiff files. Adequate software is needed to open and process the files. The GUI software does not offer the option to reload images.

2.4.11 Binning



With the “Binning” function, a selected number of neighboring pixels are added together and the sum of their counts is given out as one pixel. This function can be useful in cases where the image intensity is not high enough for a proper identification of image structures, especially when working with lower exposure times. Pixel binning leads to a decrease in the number of pixels within an image and to a loss in spatial resolution. The number of neighboring pixels which are going to be added together can be selected individually for the X- and the Y-direction in the corresponding fields “Binning X” and “Binning Y”.
A binning in T can also be defined. In this case the counts within neighboring time channels are added together and a new time channel with a corresponding larger bin time is created (see also chapter 2.8).

Note

Binning affects the recording of data, not just the image display (as e.g. the change of the dynamic range). It cannot be changed afterwards. The binning cannot be changed while acquisition is running; especially not when accumulation is activated.

The data memory for an image buffer is based on e.g. on the binning settings and is initiated always with the beginning of an exposure.

2.4.12 Zoom@Binning

Zoom@Binning is a special helping tool, implemented for instrument adjustment. In many cases instrument adjustment occurs under conditions of very low intensities. The image structures must be visible somehow for the adjustment process. Therefore the only way to combine image structure identification with low intensities is by increasing the pixel binning. The disadvantage of pixel binning (especially a high pixel binning) is that the displayed image is getting very small. This can lead to images hard to identify, especially when the adjustment of the instrument occurs in some distance from the PC monitor. Zoom@Binning has been implemented for these very situations. Zoom@Binning virtually “blows up” the image size by enlarging the size of a displayed pixel on the PC monitor, for an easier identification of small images from a distance.
Tick the box named “Zoom@Binning” to activate the function. To return to the normal image display, tick “Zoom@Binning” again and press the “Plot” button again.

Note

Zoom@Binning virtually increases the image size to fit to the 512x512 pixels of the view window. It <u>is only active</u>, when the image size is <u>smaller</u> than 512x512 pixels (either by initial settings or by binning).
Zoom@Binning <u>is not working</u>, if the image size is larger than or equal to 512x512 pixels.

For example see the following Figure 8.

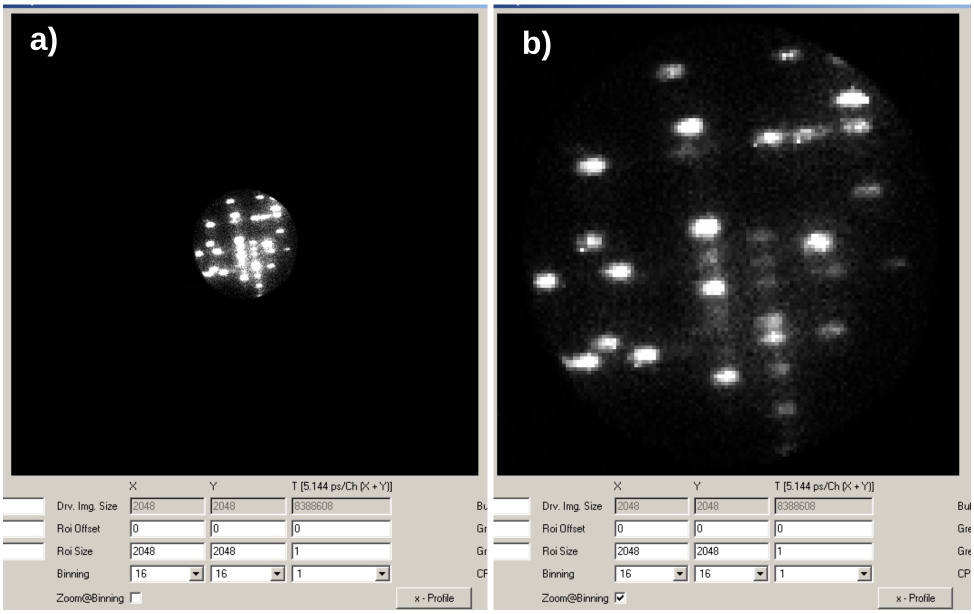
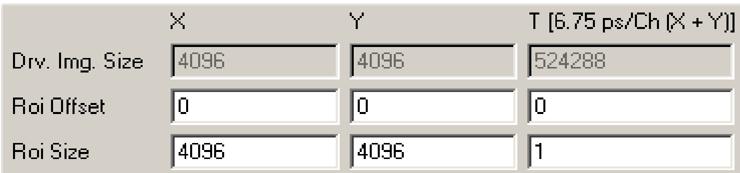


Figure 8: Effect of Zoom@Binning. a) Acquired DLD image with original size of 2048x2048 pixels, reduced to a size of 128x128 pixels by binning of 16. b) Virtually increased image to a size of 512x512 pixels by using Zoom@Binning.

2.4.13 Regions of Interest (RoI)



A Region of Interest (RoI) can be defined to display an image detail. Therefore the RoI Size (in pixel) and the RoI Position (in pixel) can be defined individually within the corresponding fields "RoI Size" and "RoI Offset" for the X- and Y-direction. The default values for the RoI are set in the way that it corresponds to the complete image size (RoI Size = Image Size, RoI Offset = 0/0).
Figure 9 shows an example for a RoI set up around a significant image structure (red marking).

Note

The RoI is always centered to the middle of the view window, regardless of the position the image has with respect to the boarders of the full image.

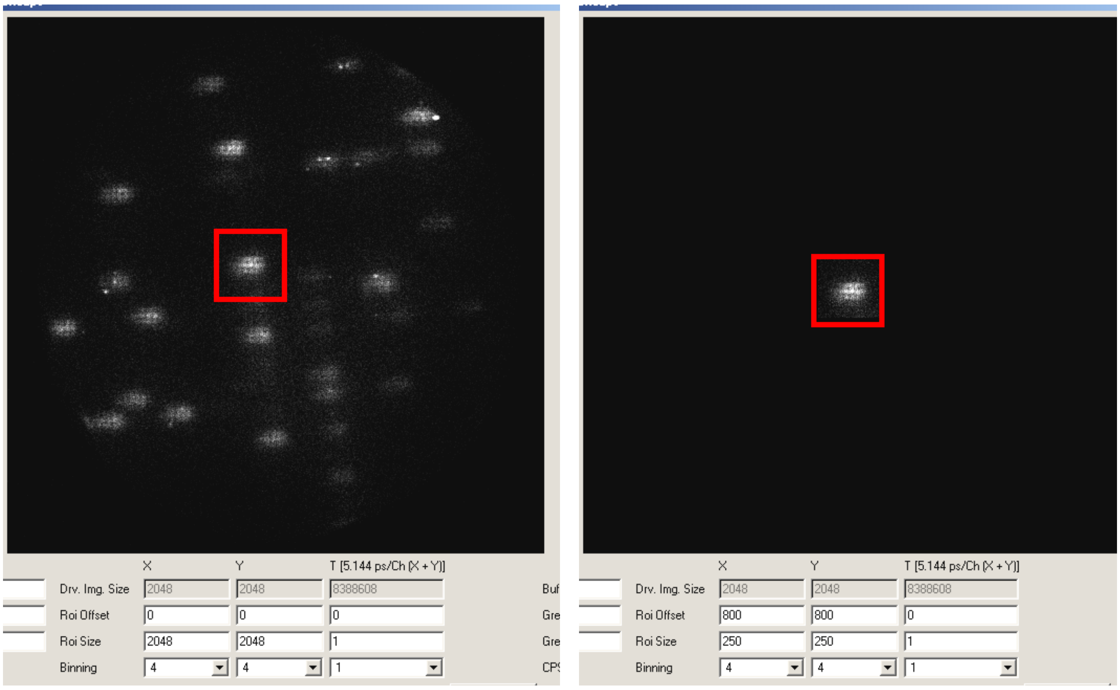


Figure 9: Measurement with a) an RoI = full image size and b) a small RoI around a significant image structure (red marking).

Note

The recording of data is affected by the settings of the RoI. It does not only affect the image display (as e.g. the change of the dynamic range). It cannot be changed afterwards. The RoI cannot be changed while acquisition is running; particularly not when accumulation is activated.

The data memory for an image buffer orientates e.g. on the RoI settings and is always initiated at the beginning of an exposure.

RoIs in time can also be defined. This is extremely important for the measurement of real 3D image stacks (for further information, see also chapter 2.8). The default setting for the RoI in time is RoI offset = 0 and RoI Size = 1. That default setting means that no image buffer for 3D image stacks is attributed, in opposite to the default RoI setting in space, where the complete image size is covered. Only the data for the 1D time histogram as well as for the integral image memory (integrated over t) is attributed. The actual reason behind this is the size of the image buffer for 3D image stacks. This size increases with an increasing number of images within a 3D stack, slowing down the complete software significantly.

For example
The number of transferred time channels into the PC, when working in PairMode with data format = 2, is 1024 (10bit) (see chapter 2.2 for details). Taking the full range as default setting for the RoI in t together with a default setting for binning in t = 1 would result in the need to allocate memory for a 3D image stack with 1024 images. The memory needed for each single image is 2048x2048x32bit, which leads to a total size for the allocated memory for a 3D image stack of 17 GByte. This is far too high to be handled by the software in an adequate time.

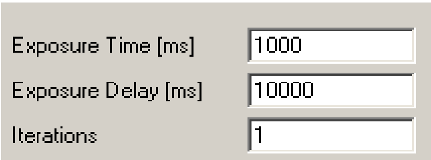
For time resolved measurements especially for the recording of 3D image stacks one has to identify the relevant features within the 1D time histogram first and then set an appropriate RoI in t as well as an appropriate binning in t for a good agreement of the number of images within the 3D image stack and the resolving of information/structures in t. See chapter 2.8 for more detailed information.

2.5 Additional acquisition functions

2.5.1 Accumulation

Each subsequent measurement is accumulated to the previous ones, when ticking the box “Accumulate”. The total result is displayed. The number of accumulated exposures is given in the field “Accumulated Exposures”. Tick the box “Accumulate” again to turn off “Accumulation”. “AutoRun” is also turned off automatically, if it was activated before. “AutoRun” cannot be turned off independently from “Accumulate”. The accumulation of images can be continued after being stopped, by activating “Accumulate” again before starting the next exposure. If “Accumulate” should be continued together with “AutoRun”, “Accumulate” must be activated first without “AutoRun” and one single exposure must be started. After this procedure “AutoRun” can also be activated and exposure can be continued. If “AutoRun” is activated together with “Accumulate”, a completely new measurement is started and already accumulated data is being deleted.

2.5.2 Delayed Exposure

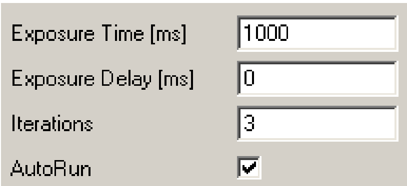


A delay time (in ms) can be entered in the field “Exposure Delay”, if acquisition should not start directly but delayed. The elapsing time displayed in the field “Exposure time” is not connected directly to the real exposure time of the hardware. This becomes obvious especially when starting an exposure with exposure delay set.

Note

The displayed time elapsed in “Exposure time” during acquisition is based on a software timer and is not very accurate. It is meant to indicate the exposure process. This elapsing “Exposure time” cannot be compared with the real exposure time of the measurement or the exposure delay. These times are highly precisely quartz controlled by the measurement hardware.

2.5.3 Running Average



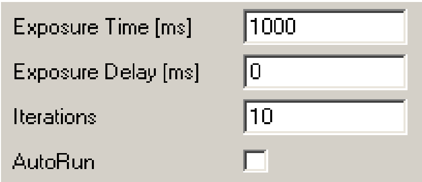
A running average over up to five images can be displayed in Auto-Run mode. Enter the number of images to be averaged into the field “Iterations”. A number higher than five is interpreted automatically as five images.

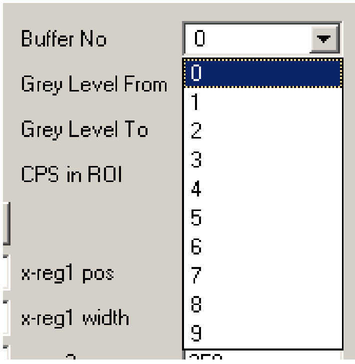
2.5.4 Multiple Exposures

Note

“Multiple exposures” is not working in Auto-Run mode. Make sure that the Auto-Run mode is deactivated, as otherwise a running average will be measured.

Enter the number of requested exposures into the field “Iterations” to acquire multiple exposures. Enter a delay time (in ms) in the field “Exposure Delay”, if the acquisition of the next image should start delayed to the acquisition of the previous one. For each image a new buffer is created automatically and the image is written into it (e.g. 10 buffers are created, when “Iterations” is set to 10). 1024 iterations can be set at maximum. After the acquisition of multiple exposures is finished, each image can be displayed by selecting the related buffer from the field “Buffer No” and pressing the “Plot” button.





To save multiple exposures press the button “Save all Buffers” (This button is only activated after acquisition of multiple exposures) and enter a filename. All images are saved under this filename and a consecutive number.

The function of measuring multiple exposures was especially developed to work with the synchronization trigger “SYNC IN” of the USB2.0-TDC. The acquisition process of the DLD can be synchronized to an external trigger signal. To do so, the trigger signal has to be applied as TTL signal to the “SYNC IN” BNC socket of USB2.0-TDC. The value of the variable named “ext_trigger” in the dld_gpx3.ini file must be set to “1” (see Figure 10), otherwise the TDC ignores external trigger signals. The TDC provides a TTL signal on the “SYNC OUT” BNC socket after each acquisition, independent on settings in the dld_gpx3.ini file.

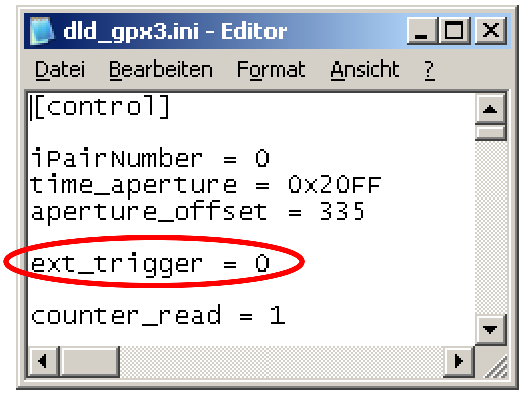


Figure 10: Screenshot of the dld_gpx3.ini file (opened with MS editor software) showing the switch that allows measurement synchronization to an external synchronization signal (=1) or not (=0).

The dld_gpx3.ini file is located in the main folder of your GUI software. Do not forget to save the dld_gpx3.ini after every change and to restart the GUI software.

The combination of measuring multiple exposures and the use of an external synchronization trigger allows an automatic data acquisition with respect to the step wise change of external measurement parameters (e.g. measurement of an intensity distribution of electrons in dependence of the electron kinetic energy in an electron spectrometer).

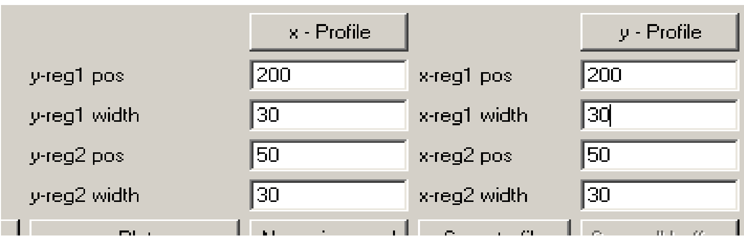
2.6 Line Profiles

2.6.1 Defining a Line Profile

The GUI offers the possibility to display line profiles for both directions X and Y. The corresponding fields for defining the exact parameters of the line profiles can be found in the GUI Main Interface. Up to two line profiles (1 and 2) can defined for each direction (X and Y). For the two line profiles in X, the positions in Y are entered in the fields “y-reg1 pos” and “y-reg2 pos” in pixels. The width for each line profile is entered in the field “y-reg1 width” and “y-reg2 width” respectively. The length of the line profiles corresponds to the RoI size.

<u>For example</u>
A line profile in X with values for “y-reg1 pos” of 200 and for “y-reg1 width” of 30 covers an area from pixel number 185 to pixel number 215 in Y - direction.

The origin 0/0 is in the upper left corner of the displayed image.



2.6.2 Display of Line Profiles

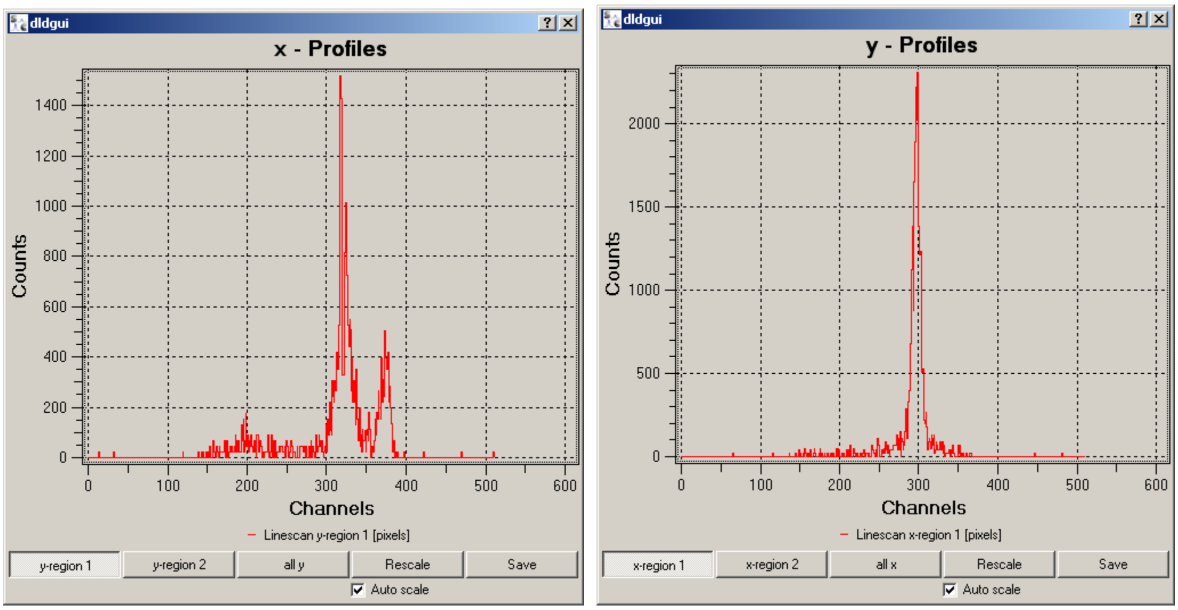


Figure 11: Profile windows for line profile in x and y.

Press the buttons "x - Profile" and "y - Profile" respectively to display the corresponding line profile. Within the profile window for x/ y each of the two profiles can be displayed individually or together by pressing the corresponding buttons "y/ x-region 1" and "y/ x-region 2". Press the button "all y/ x" to get a line scan with a width of the complete image in y/ x.
When the "Auto Run Mode" is activated, the line profiles are displayed online.

Saving a line profile

Click the "Save" button to save the activated line profiles as an ASCII data file. A standard Microsoft® window opens up.

Close a profile window by clicking on the **[x]** in the top right corner.

2.7 Count rate and count rate control

2.7.1 RoI count rate

The count rate in counts per second (cps) within a defined RoI is displayed in the field "CPS in RoI". The count rate of the whole image is displayed, when the RoI is set to the size of the whole image (default value).

2.7.2 Count rate control with the Rate Meter (for 3D Delayline Detectors)

The GUI software contains a rate meter, which opens up in an additional window when pressing the button "RateMeter". The rate meter window can only be opened after an image has been taken.

Note

The rate meter does not display a count rate directly. It displays the number of measured counts per set exposure time. Therefore the displayed number of counts must be divided by your exposure time to get the real count rate.

For example
The rate meter displays 3000 counts after a measurement with exposure time of 1000 ms. The corresponding count rate is than 3000 counts per sec. The corresponding count rate is 10.000 counts, if the exposure time was set to 300 ms.

Counts are updated after each exposure, allowing an online observation of the count rates when "AutoRun" is activated. If the count rate exceeds the displayed range, press the "Rescale" button or tick the box named "AutoScale" for an automatic scaling of the y-axis.

Figure 12 shows a screenshots of the rate meter window for a TDC running in RawData Mode with a detailed description of the individual features. Figure 13 shows two screenshots of the rate meter window to compare a TDC running in RawData Mode (upper screenshot) and in Pair Mode (lower screenshot).

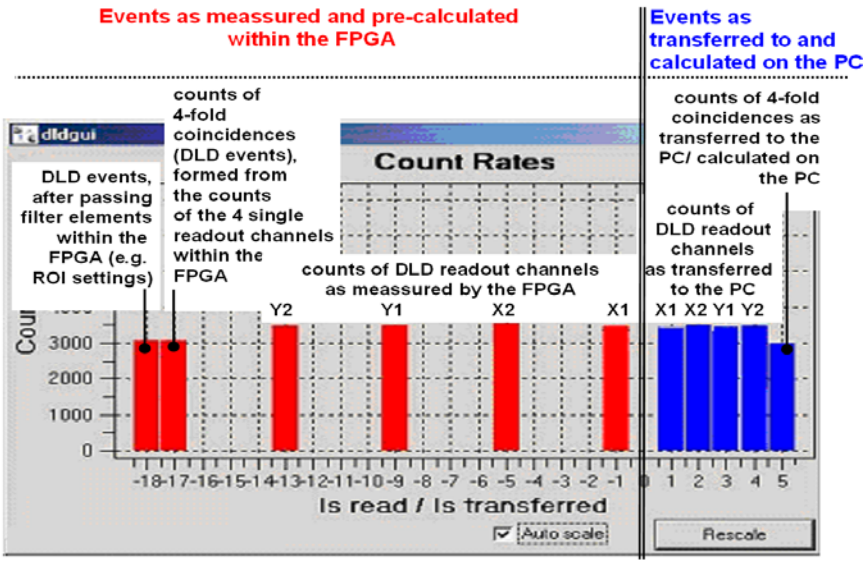


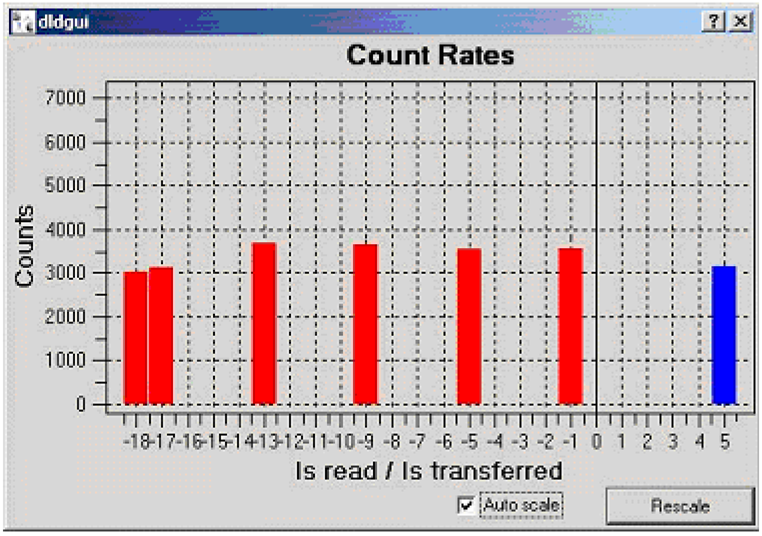
Figure 12: Rate meter window for a TDC operating in RawData Mode with detailed description of single features.

The RateMeter holds 23 different channels. These channels are divided into two groups which are marked by different colors and by a different label in the numbering.

The red channels (labeled -1 to -18) display counts, which have been read and pre-calculated by the FPGA. The counts of the four DLD readout channels (x1, x2, y1, y2) are displayed in the red channels -1, -5, -9 and -13 (for x1, x2, y1, y2). The counts of matched 4-fold coincidences (DLD events) as calculated from the results of the 4 readout channels are given in channel -17. Channel -18 shows the DLD events after passing filter elements within the FPGA (*e.g. ROI Settings*). The channels -17 and -18 display counts, which have been pre-calculated within the FPGA to reduce the transfer rate over the USB. Typically the amount of counts in channel -17 is around 80% - 90% of the amount of each of the four single DLD readout channels, but the counts in channel -18 are slightly less than in channel -17.

The blue channels (labeled 1 to 5) display counts, which were transferred to the PC. The blue channels 1 to 4 display the unprocessed counts of the four DLD readout channels (x1, x2, y1, y2) as they were transferred to

the PC. The counts displayed in channel 5 correspond directly to the events displayed in the image. The interpretation of the counts in channel 5 depends on the readout mode of the TDC (RawData or Pair).
In Pair Mode ("iPairNumber" = 2 in the dld_gpx3.ini file, see chapter 3) only the red channels, as described above and the blue channel no. 5 are displayed (see upper image in Figure 13). Here the blue channel no. 5 displays DLD events as transferred to the PC after pre-calculation within the FPGA. Therefore the channels -18 and 5 should show the same number of counts.
In RawData Mode ("iPairNumber" = 0 in the dld_gpx3.ini file, see chapter 3) the red channels as described above as well as the blue channels 1- 5 are displayed (see lower image in Figure 13). Counts from the single DLD readout channels are pre-calculated by the FPGA in RawData Mode, but the pre-calculated events are not transferred. In opposite to the Pair Mode only the pure raw (unprocessed) data from the four single DLD readout channels is transferred (blue channels 1 – 4 and the number of counts in the blue channel no. 5 corresponds to the 4-fold coincidences (DLD events) as calculated from the counts within the blue channels 1 - 4 by the software on the PC.



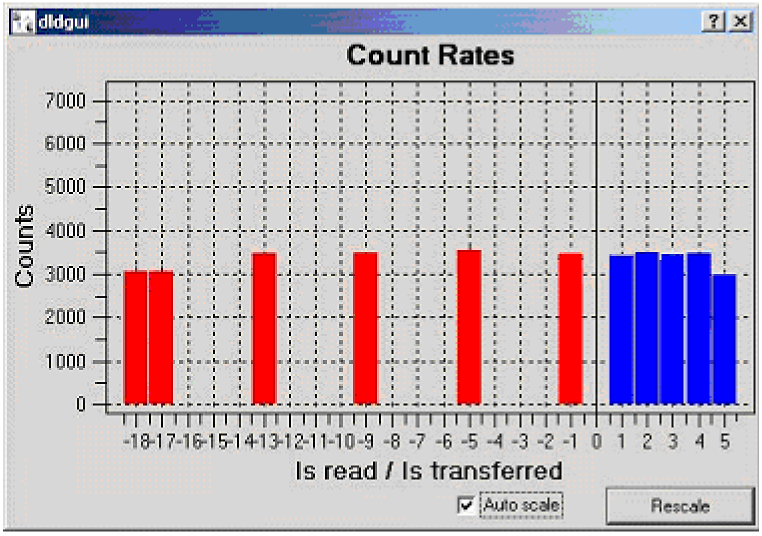


Figure 13: Rate meter window for a TDC operating in Pair Mode (upper image) and RawData Mode (lower image).

Note

In Pair mode only the red channels as well as channel 5 are displayed. In RawData mode the red channels are still displayed together with the blue channels 1 – 5, because also in RawData mode the measurement data is always pre-calculated within the FPGA. But in RawData mode only the unprocessed data is transferred to the PC and not the pre-calculated one.

In RawData mode, number of counts displayed in the channels -1, -5, -9, -13, and 1, 2, 3 and 4 respectively should be equal and also the counts in channel -18 should be comparable to the counts in channel 5, as long as the transferred rate is smaller than the transfer limit of the USB. The increase of rates higher than the transfer limit of the USB will result in the effect, that the counts displayed in the channels 1, 2, 3, 4 and 5 will stay unchanged, while the counts displayed in channels -1, -5, -9, -13, -17 and -18 will further increase.

Note

The count ratio between single channels and the 4-fold coincidence (counts in channel no. -17/-18 compared to counts in channel no. -1, -5, -9, -13 and counts in channel no. 5 compared to counts in channel no. 1, 2, 3,and 4 respectively) will decrease when going to high count rates (up to 10E6 and more) due to an increase in multi hit events and connected to this a loss of 4-fold coincidences.

The rate meter is a powerful tool mainly used for diagnostics, as e.g. input control of the four DLD readout channels. The ratio between the counts of the four readout channels and of the 4-fold coincidence is a measure for the detection efficiency of the detector. For a well-adjusted detector system the number of counts within the single readout channels should be of almost equal height and the counts of the 4-fold coincidence should be of 80 % or more of the number of counts of the single readout channels.

Close the rate meter window by clicking on the **[x]** in the top right corner.

2.7.3 Count rate control with the Rate Meter (for 4 Quadrant Delayline Detectors)

The rate meter is looking quite different, when operation the GUI together with a 4 Quadrant DLD. In this case all red bars (-1 to -18) and all blue bars (1 to 5) are in use to plot the single measured counts as well as the measured 4-fold coincidences independently for all 4 quadrants. A screenshot of the rate meter window of the GUI running in the "4Q_DLd mode" is displayed in Figure 14 with a detailed description of the individual features.

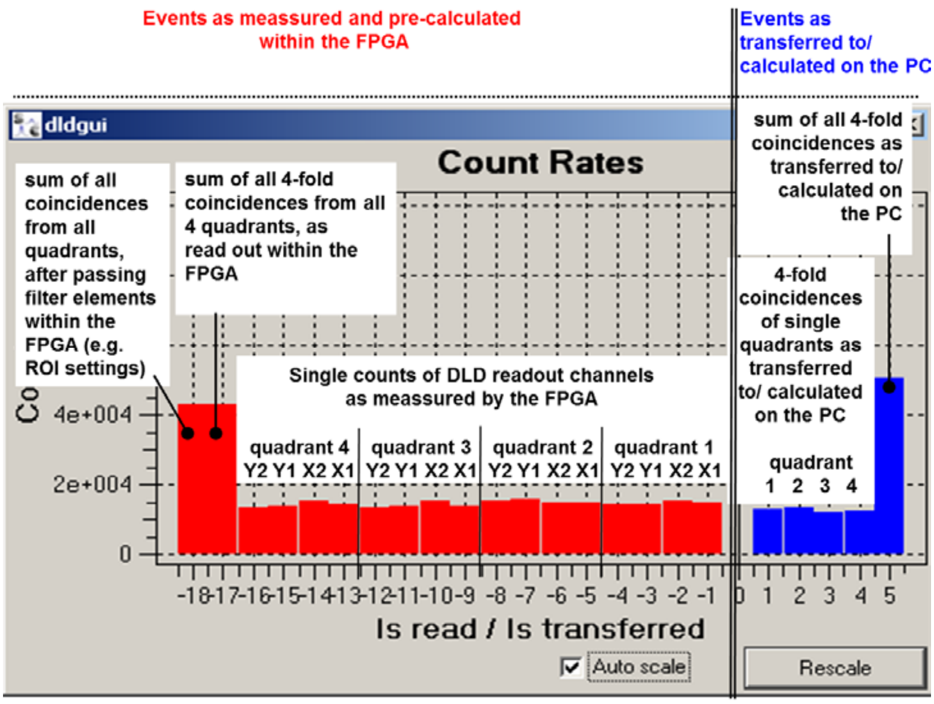


Figure 14: Rate meter window for a 4 Quadrant DLD with detailed description of single features.

The measured counts for each readout channel (X1, X2, Y1, and Y2) for each quadrant 1 – 4 are displayed in the red bars -1 to -16 (e.g. red bars -1 to -4 show measured counts for X1, X2, Y1, and Y2 of the first

quadrant and so on). Here again the red channels (labeled -1 to -18) display counts, which have been read and pre-calculated by the FPGA. The sum of matched 4-fold coincidences (DLD events) from all 4 quadrants is given in channel -17, while channel -18 shows the sum of all DLD events after passing filter elements within the FPGA (*e.g. ROI Settings*).
The channels -17 and -18 display counts, which have been pre-calculated within the FPGA to reduce the transfer rate over the USB.
Equivalent to the description above (rate meter for 3D DLDs), also here the blue channels (labeled 1 to 5) display counts, which have been transferred to the PC. The blue channels 1 to 4 now display the 4-fold coincidences of the single quadrants 1 to 4 as transferred to/ calculated on the PC. The counts displayed in channel 5 correspond directly to the sum of the counts displayed in channel 1 to 4 (sum of all 4-fold coincidences). Although there is no visible difference in the rate meter when operating the TDC in RawData or Pair mode, the meaning of the counts displayed in channels 1 – 4 is slightly different.
In Pair Mode ("iPairNumber" = 2 in the dld_gpx3.ini file, see chapter 3) the blue channels no. 1 to 4 display the DLD events as transferred to the PC after pre-calculation within the FPGA. Therefore the channels -18 and 5 should show the same number of counts.
In RawData Mode ("iPairNumber" = 0 in the dld_gpx3.ini file, see chapter 3) the blue channels 1 to 4 display the DLD events as calculated by the GUI software on the PC after the transfer of the pure raw (unprocessed) data from the 16 single DLD readout channels. In RawData mode the channels -18 and 5 must not necessarily show the same number of counts (this depends on the measured count rate and the transfer speed to the PC).

Note

4Q_Dld = 1 must be set in the QDLD section of the dld_gpx3.ini file to use the GUI software to read out a 4Q DLD (this is working only with a 4Q detector system, of course).

2.8 Time resolved measurements

The following list gives you a step-wise overview on how to perform time resolved measurements

1. **Apply an external start signal to TDC and turn off the TDC's internal start signal**
2. **Measure a 1D time histogram to identify interesting features within the defined time range**
3. **Set an appropriate RoI and binning in time, to cover interesting features within a 3D image stack**

2.8.1 Preparation for Measurements with an External Start Signal

Time resolved measurements can only be performed when an external start signal is given to the start input of the USB2.0-TDC and when the internal start signal has been turned off.
The external start signal must be applied as a standard TTL signal to the "TDC-start" input (BNC socket). Additionally, the internal start signal has to be turned off by ticking the box "Ext. TDC Start" in the main interface of the GUI software. A warning message (Figure 15) appears when deactivating the internal start signal. Follow the instructions given within the warning message before confirming by clicking "o.k." Otherwise click "cancel" ("Abbrechen").

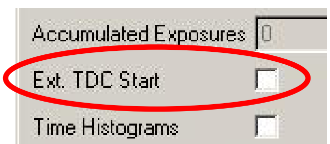


Figure 15: Warning message when ticking the „Ext. TDC. Start" box in the GUI Main Interface.

Untick the box "Ext. TDC Start" to change back to the internal start signal. Again a warning message appears (Figure 16). Follow the instructions given within this warning message before confirming by clicking "o.k." Otherwise click "cancel" ("Abbrechen").

Figure 16: Warning message when unticking the „Ext. TDC. Start" box in the GUI Main Interface.

Note

Take care that measurements are performed either with the internal start signal turned on and <u>no signal</u> applied to the TDC-start input (BNC socket) or with the internal start signal turned off (Ext_Gpx_Start = 1) and an external start signal applied to the TDC-start input (BNC socket).
In all other cases/ combinations the TDC is not working correctly.

Note

For further information about the hardware connection of the start input signal, see your hardware manual (DLD, TDC etc.).

Note

The temporal resolution is influenced mainly by the quality of the start signal. The TDC measures the time in a leading edge determination. Therefore, the start signal must be processed by means of a constant-fraction-discriminator or similar external electronics components in case that it is varying in time.

2.8.2 Measurement of 1D Time Histograms

Before any 3D image stacks can be measured, the size of the memory buffer for the stack must be defined. The 3D image buffer increases dramatically and very quickly with increasing number of images, slowing down the complete software significantly. Therefore there are no default settings for the image buffer. It must be defined individually by the user by setting an appropriate RoI and binning in time (see chapters 2.4.11 and 2.4.13).
To find the correct settings (RoI and binning in time) for an optimized measurement of a 3D image stack, one should identify the position of interesting features in time by observing a 1D time histogram, first. Therefore the next step is to measure a 1D time histogram.

a) <u>Layout of 1D Time Histograms</u>

To measure a time histogram tick the box named "Time Histograms" first and then acquire an image. With ticking the "Time Histograms" box also the button named "TimeHist" becomes active.

Note

There is no box named "Time Histograms", when working with the GUI Main Interface modified for smaller screens. Here the functionality of the box "Time Histograms" is initially activated and stays activated as long as the GUI software is running.

Click the button "TimeHist" at the bottom of the main interface to open the 1D time histogram window shown in Figure 17.

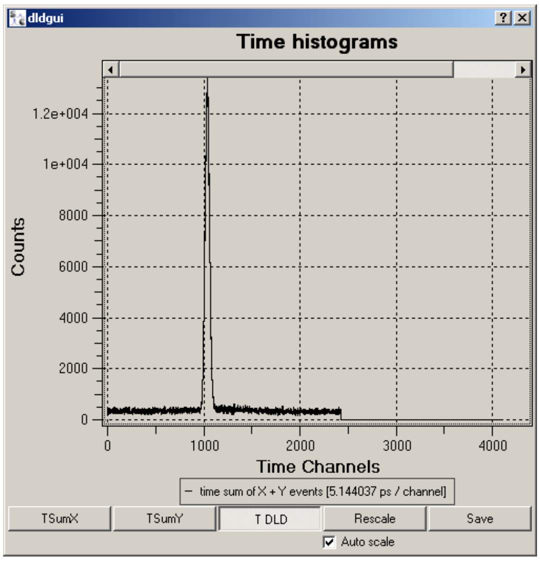


Figure 17: 1D time histogram window

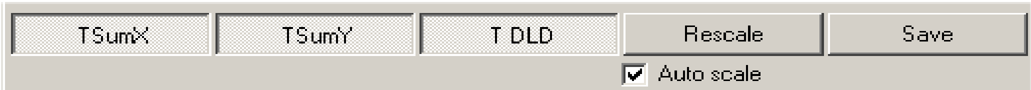
As default setting, the time sum of the 4-fold coincidence, the total time sum of X+Y events ("T DLD"), is displayed as integrated signal over the complete RoI in x and y. Therefore it does not show any lateral resolved information initially. The time structures of individual sample positions can be investigated by defining an appropriate RoI in x and y (see chapter 2.4.11). The 1D time histogram presents the number of measured events with respect to a channel number.

Note

The size of a single channel depends on the operation mode of the TDC as well as on the binning in time and is displayed within the histogram. The value range for events of the total time sum is expanded by a factor of four due to the 4-fold measurement of an event. Therefore the size of a single channel within the histogram is reduced by a factor of four with respect to the original time bin size of the TDC operation mode.

For example
The time bin size in G mode is 20.576148 ps. Therefore the size of a single channel for the total time sum with a binning in time of 1 is 5.144037 ps (= 20.576148/4).

If "AutoRun" mode is activated, the time histogram is updated automatically at the end of each new exposure, allowing an online investigation for small exposure times.
If the number of measured counts exceeds the displayed range, press the "Rescale" button or tick the box named "AutoScale" for an automatic scaling of the y-axis.

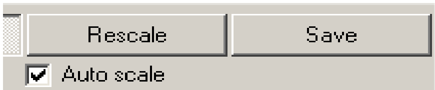


Beside the total time sum "T DLD", also the time sum in x ("TSumX") and the time sum in y ("TSumY") can be displayed as individual 1D time histograms. Press the corresponding buttons at the bottom of the 1D time histogram window for displaying.

Note

The size of a single channel for "TSumX" and "TSumY" is by a factor of 2 larger than the channel size for "T DLD", because "TSumX" and "TSumY" are formed from a 2-fold measurement.
"TSumX" and "TSumY" do not hold any additional information for a time resolved measurement than "T DLD". But they do hold some additional information about the performance of the detector. Therefore they have some diagnostic use. For time resolved measurements it is absolutely sufficient to work only with "T DLD".

b) Zooming in and out



The complete time range (see chapter 2.8.3) is displayed by default, when opening the 1D time histogram window. Point with the mouse onto the time histogram window and move the mouse while holding the left mouse button pressed, to draw a rectangular field. The part of the histogram within the rectangle is displayed enlarged ("zoom in") in the complete histogram window after releasing the left mouse button. Press the "Rescale" button (sometimes several times) to "zoom out".

c) Saving and closing of 1D Time histograms

To save a time histogram, press the button "Save". A standard Microsoft™ window opens up, where a filename can be entered. Always all three histograms ("T DLD", "TSumX" and "TSumY") are saved in an ASCII data file.

Note

The GUI software does not offer the option to reload time histograms.

Close the time histogram window by clicking the **[x]** in the top right corner.

2.8.3 Time Range of 1D Time Histograms

The displayed time range (number of displayed channels) of a 1D time histogram is defined in the field "1D_histo_size" within the dld_gpx3.ini file, by entering the number of displayed channels in hex (see Figure 18).

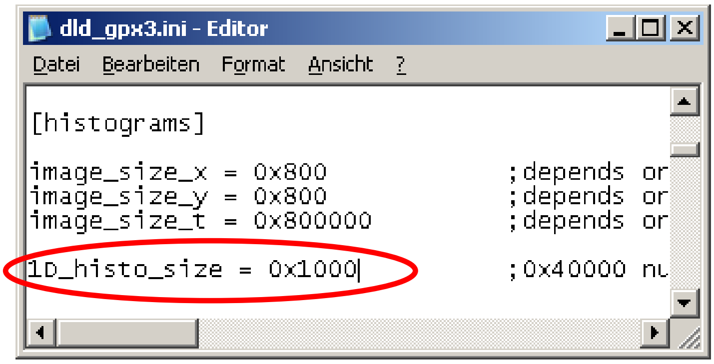


Figure 18: Screenshot of the dld_gpx3.ini file (opened with MS editor software), showing the entry field to define the size of the 1D time histogram.

Note

The displayed time range is not necessarily the same as the measured time range. It can be smaller or larger.
Do not work with a 1D histogram size larger than a few hundred thousand channels. The PC system starts to slow down significantly with a larger number of channels.
The maximum number of measured time channels depends on the operation mode of the TDC.

a) Time Range in RawData Mode

In RawData mode, the number of measured time channels is defined by the operation mode of the TDC itself (*e.g. R mode or G mode, for further details see the manual of your delay line detector*).

For example
The maximum time range, which can be measured in R mode, is 40µs. Therefore the maximum number of measured time channels for the total time sum "T DLD" is about 5839416 (40µs/6.85ps).

By the way, the 6.85ps channel width is calculated from the time bin size in R mode of 27.4ps divided by 4 due to the 4-fold measurement of "T DLD". In G mode the bin size of T DLD is 5.14ps (41.1523ps divided by 8, due to an 8-fold measurement, because in G mode the falling and rising etch of each of the four delay line signals are measured).

Note

The memory buffer needed for a 1D time histogram with about 6 million channels can easily reach a size of a few hundred Mbytes. This can slow down a PC system dramatically, depending on the hardware and free memory of the PC. It is highly recommended to work with a binning in time to reduce the number of channels to a few hundred thousand.

It is not unlikely that measurement data is arriving with a time offset in respect to the start trigger, which is larger than the displayed time range. This is the main reason for an empty 1D time histogram when measuring in RawData mode. The easiest way to deal with this is to use a binning in time (see chapter 2.4.11 for details) to evaluate the time offset. First try to do so without increasing the 1D_histo_size. Only increase the 1D_histo_size, if no time distribution is shown in the 1D time histogram even if the largest binning in time is activated.
With the time distribution displayed in the histogram, one can now identify the time offset. It can be subtracted from all results by entering the offset value in the corresponding field in the dld_gpx3.ini file or alternatively in the entry field "Offset" in the GUI main interface. See chapter 0 for further details.

Note

Take account of the binning in time when evaluating the time offset.

The binning in time can be reduced again after removing the offset and the relevant time distribution should still be displayed in the 1D time histogram.

For example
Figure 19 shows two time histograms with 80000 time channels each. The first histogram (a) has been measured with a binning in time of 1, showing no data at all, while the second histogram (b) shows a measured time distribution. It was measured with a binning of 4. After subtracting the time offset (c) one can reduce the binning in time back to 1 (d).

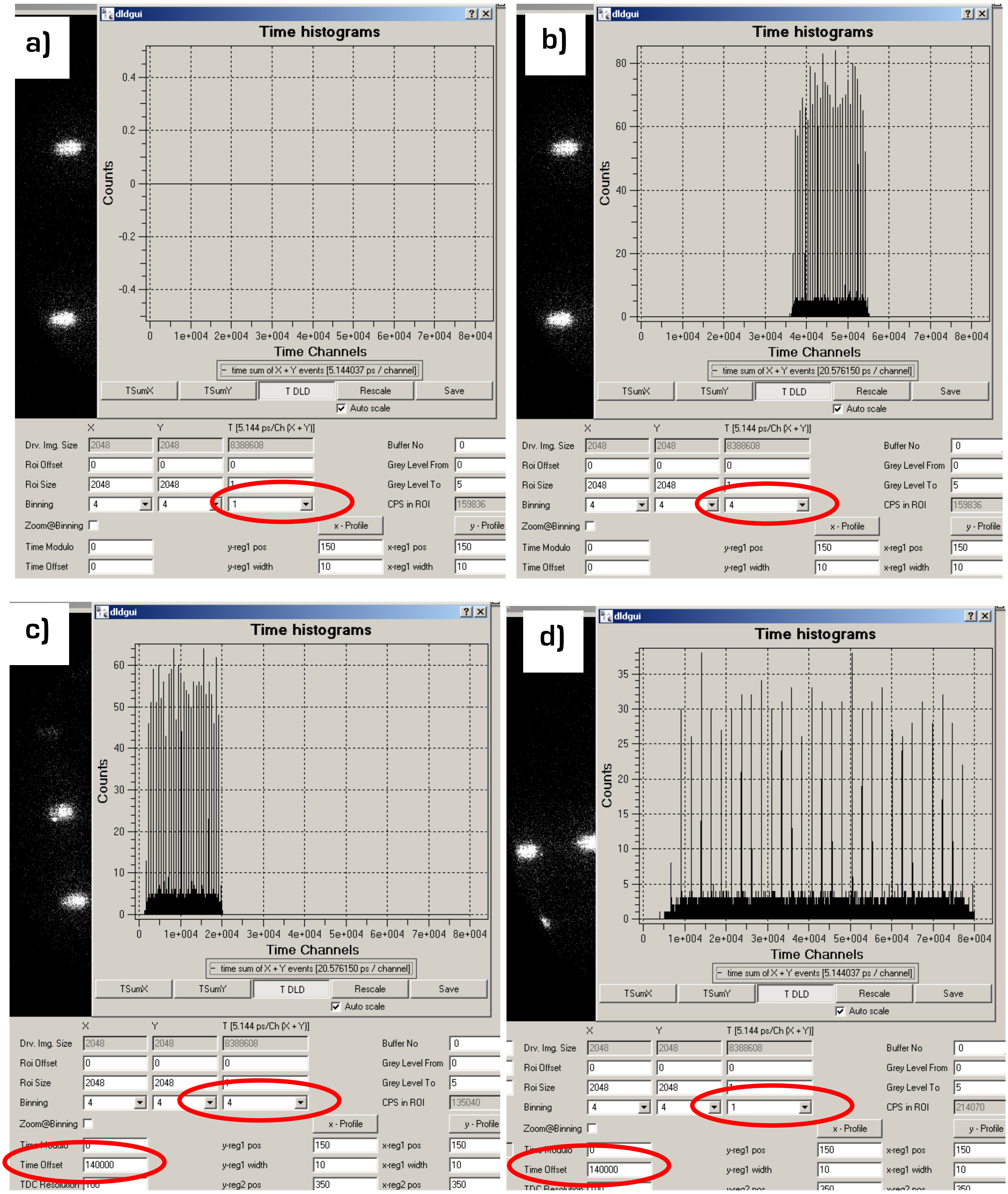


Figure 19: 1D time histogram with a size of 80000 time channels and a) a binning in time of 1, showing no data at all and b) a binning in time of 4, showing a measured time distribution. 1D time histogram after removing an offset of 140000 channels measured at a binning in time of c) 4 and d) 1.

Note

Take account of the time offset, if measuring absolute time distances for structures within a time distribution, displayed in the 1D time histogram.

b) *Time Range in Pair Mode*

The number of measured time channels in Pair mode is defined by the DataFormat (see chapter 2.2 for details). Hereby the maximum possible number of time channels is 14 bit in data format 1.

Note

A large number of time channels goes hand in hand with a reduced number of pixels in x and y. (e.g. data format 1 allows 14 bit of time channels but restricts number of pixels in x and y to 9 bit each).
The entry in the field "1D_histo_size" should fit to the time range selected with DataFormat when working in Pair mode.

Events, which appear with a larger time distance than defined by DataFormat, are sorted into the defined time window automatically.

For example:
Events are measured with DataFormat = 1 which corresponds to 16384 channels, which again corresponds to a time range of about 112 ns for R mode. Events which would appear in a channel no. 16385 are sorted to the events in channel 1, events from a channel no. 16386 to the events in channel 2 and so on up to the events which would appear in a channel no. 32769 which are again sorted to the events in channel 1.

One way of dealing with the incorrect display of time distributions in the 1D histogram is to restrict the measured range of events in time by defining a RoI in time. Only results within the defined RoI are displayed. Now the RoI can be moved stepwise along the time axis for a sub sequential scanning of the time distribution.

2.8.4 3D Image stacks in time

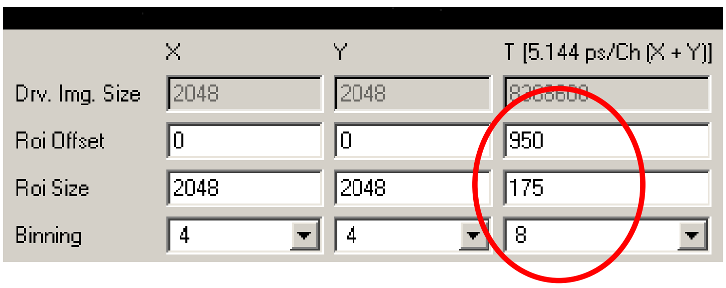
3D (x,y,t) image stacks combine time and lateral resolved measurements. To record a 3D image stack an appropriate RoI in time must be defined first, in order to allocate the memory needed for the image buffer.

Note

Note the information given in chapter 2.4.13 concerning the size of the image buffer for a large number of images within the 3D image stack. Therefore it is very important to set an appropriate RoI in t as well as an appropriate binning in t for a good agreement with the number of images within the 3D image stack and the resolving of information/ structures in t.

The appropriate settings for the RoI as well as for the binning can be found with the 1D time histogram.

For example:
Figure 20 shows a 1D time histogram. The interesting feature within the time distribution appears between time channel 950 and 1125 (indicated by the red lines in Figure 20). Therefore an appropriate setting for the RoI in time would be 950 for RoI Offset and 175 for RoI Size. To further reduce the number of images of 175, one can use an appropriate binning in time. A binning of 8 would reduce the number of images to about 22.



Note

Do not use a time histogram, which has been binned in time to evaluate RoI Offset and RoI Size. This must always happen from a 1D time histogram measured with a binning in time of 1. Otherwise the RoI is set to a wrong position in time.
Use a binning in time only after evaluating RoI Offset and RoI Size, to further reduce the number of images if needed.

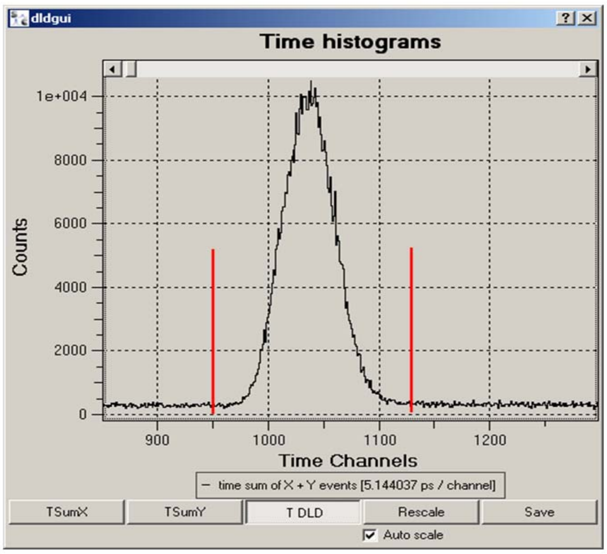


Figure 20: 1D time histogram window

After the values for RoI Offset, RoI Size and binning in time are set, start the acquisition of a 3D image stack by pressing exposure. One can either set an appropriate exposure time, but one can also work with accumulation in combination with or without “AutoRun” mode.

Note **All images within a 3D image stack are measured parallel. Not sequential.**

The total time sum is given out as a result on the screen during and at the end of the exposure.

Saving of 3D image stacks

3D image stacks are saved in the same way as a single detector image by clicking on the “Save to file” button. When saving a 3D image stack, all acquired 16bit tiff-data files are saved individually. A consecutive numbering is added to the file name. Additionally also the total time sum image, as already plotted on the screen and an error image is saved. The total number of images saved is given by RoI size/selected binning in time +2.

Note **It is recommended to save image stacks always into separate folders.**

2.9 Multiple View Windows

A measurement can be displayed in several view windows. Press the button named “New view wnd” to open up a new view window.

This new view window gets its data from a new and independent image buffer which allows image modification without affecting the original view window of the GUI main interface or other new view windows. Measurement results are updated in all view windows simultaneously and in real-time.

Note

Several new view windows can be opened parallel, but the PC is slowed down significantly with increasing number of view windows.

The additional view windows contain viewing functionalities that are not available for the original GUI view window. Figure 21 shows a screenshot of an additional view window.

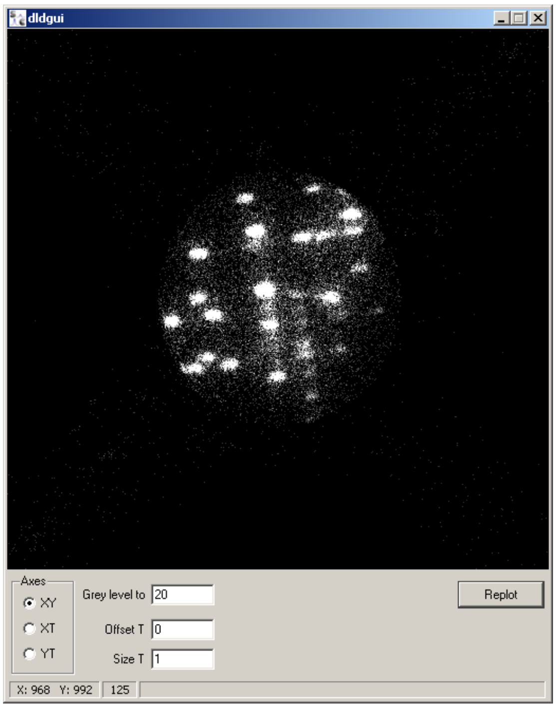


Figure 21: Screenshot of the additional view window

The image size of the additional view windows is set to 512x512 pixels. Image sizes are reduced automatically when working with larger images. Hereby size reduction is reached by simply removing lines and rows. This can lead to a significantly decrease of displayed intensity.

<u>For example:</u> A detector delivers an image size of 2048x2048 pixels. This image is displayed in a new view window of 512x512 pixel size. 3 out of 4 lines and rows are simply deleted to fit the image into the view window when working with binning 1x1. This leads to a displayed intensity of only 1/16 of the real measured intensity,

This effect can be compensated, when working with higher binning values. Image binning takes effect before loading the image data into the image buffer of a new view window. Therefore binning also affects the image, which is displayed in a new view window.

For example: *A detector delivers an image size of 2048x2048 pixels. The image size is reduced to 512x512 pixels when working with binning 4x4. In this case the image fits into a new view window without any deleting lines and rows and without any reduction in the displayed intensity.*

2.9.1 Grey level adjustment

In each view window the grey levels of the displayed image can be modified separately. Press the Replot-button to show the new set grey level. This is not necessary, if the GUI is updating images continuously in AutoRun mode.

2.9.2 Mouse pointer coordinates and pixel intensity

The new view window gives information about the x and y coordinates and the pixel intensity of the image at the actual mouse position. They are shown in the bottom part of the new view window.

2.9.3 Image Zoom

Use the mouse wheel or PageUp/PageDown to zoom in and out of the image at the actual mouse position. It is also possible to zoom only in 1 dimension (x or y). Use the mouse wheel and “ctrl” to zoom in x direction and use the mouse wheel and “shift” to zoom in y direction.

2.9.4 2D views and individual RoI settings

It is possible to select different 2D views (XY, XT, or YT) and an individual RoI can be defined for the third dimension. The default settings for the RoI are Offset = 0 and Size = 1 and correspond to the full time range/ full image size. Entries for X and Y settings (in XT and YT view) are given in pixels and entries for T (XY view) are given in channel numbers.

Especially the selection of different 2D views with one time axis is a very helpful tool for the analysis of time slots in real-time. Figure 22 shows all three possible 2D views of one and the same measurement with the RoI of the third dimension set to the full range.

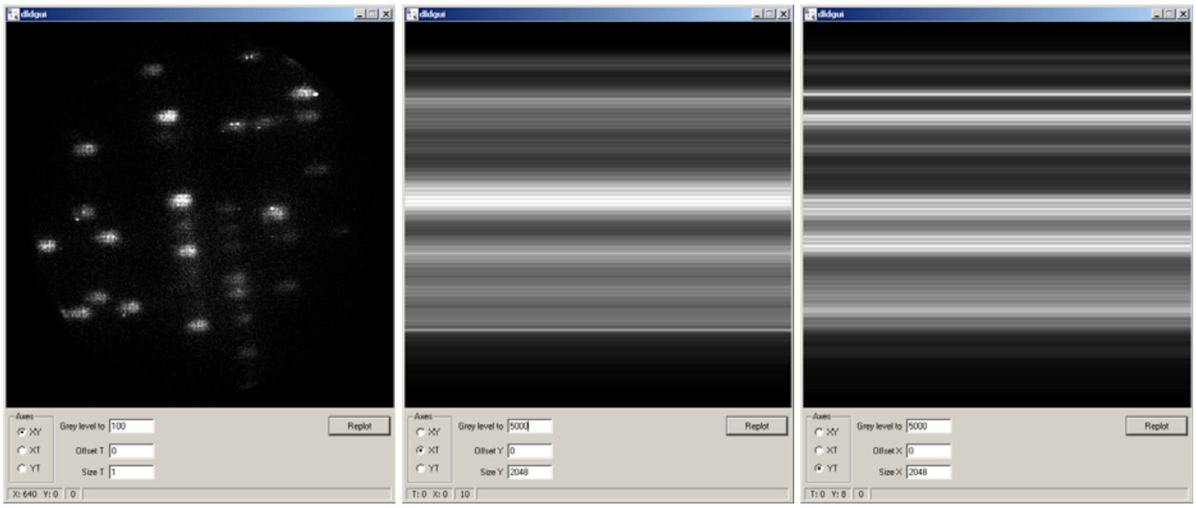


Figure 22: All three possible 2D views of one and the same measurement.

2.10 Data Saving Formats

2.10.1 Standard data formats

The standard data format used to save the acquired images is the 16-bit tiff-file format and for the 1D time histograms or Line Profiles the ascii-file format.

2.10.2 List Mode Files (count file)

In the GUI software it is also possible to save the acquired data in a binary file sorted in columns by X, Y, TsumX and TsumY, the so called list or count mode. The list mode file option must first be activated in the dld_gpx3.ini file (see chapter 3) by setting the switch “Listmodefile” to “1”. This will create an additional box named “Listmodefile” in the GUI Main Interface. Tick the box to save the acquired data additionally in a file named “counts_xxx.dat” to the folder of the GUI software. The triple x indicates the added string to the file name (e.g. counts_1233306575.dat). It is the time of the operation system and is added automatically by the GUI software. One additional major aspect of the List Mode File is its header structure. It contains all important information about measurement parameters and TDC settings e.g. set ROI, binning, calibration, modulo value, TDC mode, time bin resolution etc. With each change in one of these settings a new list mode file with updated header information is created.

Additional options concerning the list mode file can also be set within the dld_gpx3.ini file:

Set “GUI_LMF_default_on” to “1” to have the Listmodefile checkbox in the GUI Main Interface ticked by default when starting the GUI Software.

Measurement data can be written to the list mode file with or without respecting any binning setting made in the GUI Software. Set “ListModeAsBinned” to “1” to respect the binning settings when writing data to the list mode file.

Measurement data can be written to the list mode file with or without respecting any ROI setting made in the GUI Software. Set “ListModeAsIn_ROI” to “1” to respect the binning settings when writing data to the list mode file.

The file path of the list mode file can be defined in “ListModeFilePath = "c:\\temp"”.

Many TDCs are equipped with an synchronization trigger input to synchronize the start of a measurement to an external trigger signal. “ext_trigger” must be set to “1” within the dld_gpx3.ini file to activate this BNC synch trigger input for such hardware triggered exposures (trigger signal must be TTL or LVTTL in default cases).
A separate list mode file can be written with each hardware triggered exposure when setting “SingleLMFPerExposure” to “1”.

2.10.3 Bulk Files

A third method to save data is the Bulk File. The Bulk File contains the pure raw data stream as measured by the TDC unaffected by the settings of ROI, Binning etc., but affected by the channel delay values. To save the data stream, set **both** switches “SaveBulk” and “Debug” in the dld_gpx3.ini file to “1” (see chapter 3).

! Note

The GUI software will start in the so called debug mode, when “debug” is set to “1”. In debug mode a window with additional information appears at the start of the software.

The TDC data stream is stored in a file named “bulk.txt” in the GUI software folder. All incoming TDC data is added consecutively to this file unaffected by stopping and restarting of measurements. To save only one specific measurement, any existing bulk.txt should be deleted from the GUI software folder just before starting the measurement and it should be renamed directly after the end of the measurement.

Each bulk file can be analyzed in the simulation mode (see chapter 2.12) by using the bulk file as a simulation file. The relevant bulk file hast to be renamed to “sim_input.txt” and be stored in the GUI software folder.
Now it is possible to extract 1D time histograms and do measurements of a certain RoI etc. with the data from the original measurement.

! Note

Each bulk.txt file can be used for simulation mode by renaming it into sim_input.txt. Now it can be revised with the GUI software after measurement is done.
A backup of any existing sim_input.txt files and bulk.txt files should be created before.

! Note

There are 4 individual bulk files when operating a 4-Quadrant DLD with a 4-fold Quad Channel USB2.0-TDC. One bulk file for each of the 4 TDC units within the 4-fold Quad Channel USB2.0-TDC.

2.11 Calibrated Measurements

Each measurement can be calibrated for x, y, t distortions as well as for intensity inhomogeneous (flat field correction) directly during a measurement using appropriate calibration files.

2.11.1 Flat Field Correction

Each detector comes with a certain inhomogeneity regarding the detection of a homogeneous intensity distribution. This can be corrected directly during a measurement using intensity calibration.

The intensity calibration algorithm can be activated by setting “cal_intensity” to “1” in the dld_gpx3.ini file (see chapter 3). Additionally an adequate calibration file named “cal_i.tiff”, stored in the GUI software folder is needed. Otherwise the intensity calibration cannot be implented. The intensity calibration is always done at the end of a complete measurement (e.g. when accumulate is activated, the intensity calibration is performed on the accumulated image after the measurement has been stopped).

2.11.2 Calibration of spatial distortions (x, y calibration)

Each detector has a certain spatial distortion (the misfit in imaging a straight line). Also this behavior can be corrected during a measurement using x, y calibration.
The x, y calibration algorithm can be activated by setting “cal_xyt” to “1” in the dld_gpx3.ini file (see chapter 3). Additionally two adequate calibration files named “cal_x.tiff” and “cal_y.tiff” have to be available in the GUI software folder. If one of the two files is missing, the x, y calibration cannot be performed.
The x, y calibration is executed on each single detected event in contrast to the intensity calibration.

2.11.3 Calibration of distortions in time (t calibration)

Similar to the spatial distortions, each detector has a certain distortion in time (the difference in a time measurement of the same event in time but hitting different spatial detector positions), which again can be corrected directly during a measurement using t calibration.
The t calibration algorithm is activated (together with the x, y calibration) by setting “cal_xyt” to “1” in the dld_gpx3.ini file (see chapter 3). An adequate calibration file named “cal_t.tiff” must be available in the GUI software folder in addition. The time calibration works irrespective of the x, y calibration although they are always activated together in the dld_gpx3.ini file. Calibration files for the x, y calibration are not needed for a calibration in time and vice versa. The t calibration is also performed on each single detected event.

Note

A calibration file for the flat field correction can only be used under that same conditions as it has been measured in respect of x, y and/or t correction. This means, that a calibration file measured without a correction in x, y and/or t is not exchangeable with a calibration file that has been measured with an activated correction in x, y and/or t.

Note

There are 4 individual sets of calibration files when operating a 4-Quadrant DLD with a 4-fold Quad Channel USB2.0-TDC. One set for each of the 4 TDC units within the 4-fold Quad Channel USB2.0-TDC.

2.12 Simulation Mode

The simulation mode offers the possibility to work with the GUI software and to learn its basic functions without a running hardware.It also allows analyzing measured data after the measurement has been done (see chapter 2.10.3, bulk file).
The simulation mode is activated by setting the value of the parameter “Simulation” in the dld_gpx3.ini to “1” (see chapter 3). The name of the file used for simulation can also be defined in the dld_gpx3.ini file via the parameter SimulationDataFile, e.g. (SimulationDataFile = ".\simulation_dot.txt"). The name “sim_input.txt” is used as default, if no other file name is defined. Make sure that no hardware is connected to the PC before starting the GUI software.

Note

The simulation mode is only running with no hardware (TDC) connected to the PC.
The simulation mode does not respond to the exposure time, because this is hardware driven.

Note **There are 4 individual simulation files when operating a 4-Quadrant DLD with a 4-fold Quad Channel USB2.0-TDC. One simulation file for each of the 4 TDC units within the 4-fold Quad Channel USB2.0-TDC.**

2.13 Measurement and Device Adjustment

The adjustment of a measurement and/or the TDC is often done within the dld_gpx3.ini file (see also chapter 3). Nevertheless it is more convenient for a final adjustment, if different parameters can be changed directly via the GUI Main Interface instead of working with the ini-files. Therefore a couple of additional entry fields can be displayed in the GUI Main Interface. These fields are Time Modulo, Time Offset, I2C Resistor 0 and 1, the channel delay adjustment fields 0–15, and the Q-shift fields. Change the value for "modulo_adjust" from 0 to 1 in the dldgui.ini file (see also chapter 3) to display the entry fields of Time Modulo, Time Offset and TDC Resolution and change the value for "Channel_adjust" from 0 to 1 to display the entry fields of I2C Resistor 0 and 1, for the 16 channel delay adjust fields (Ch 0 – 15) and the Q-shift fields. Figure 23 shows the GUI Main Interface with the additional entry fields activated.

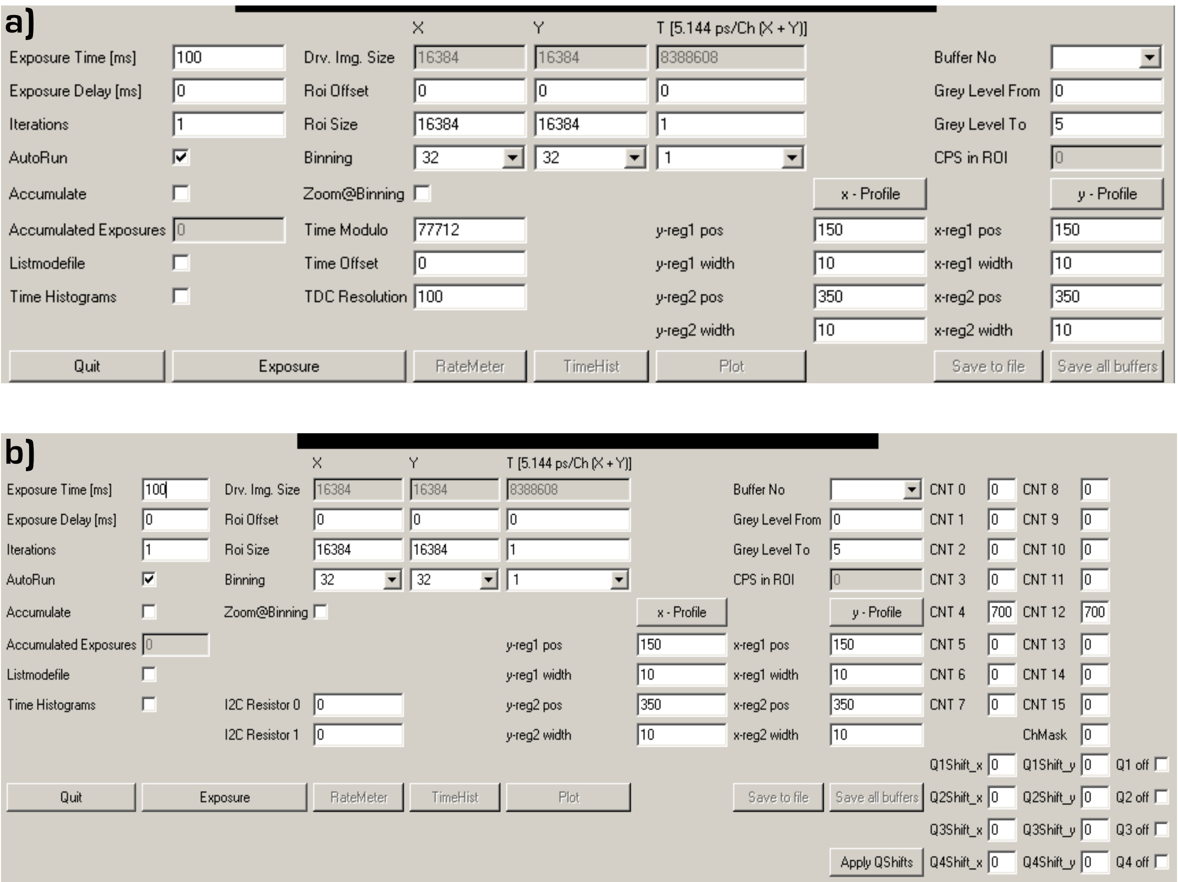


Figure 23: GUI Main Interface with all additional entry fields displayed if (a) modulo_adjust and (b) channel_adjust are set to "1".

Note **The entry field "TDC Resolution" will not be displayed, if both "modulo_adjust" and "channel_adjust" are activated.**

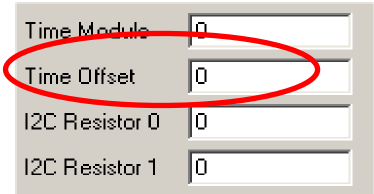
All additional fields are basically used for measurement/device adjustments at the beginning of the measurements. Later on these values can be entered directly in the dld_gpx3.ini file and the additional entry fields can be turned off. The display of all additional entry fields is turned off by default.

Note

A value entered in the additional entry field is <u>not</u> automatically transferred to the dld_gpx3.ini file. The value must be changed with each start of the GUI software if it is not entered and saved in the dld_gpx3.ini file.

2.13.1 Time Offset

The parameter “Time Offset” allows to define a cut-off area for a given time distribution. For example this is useful, if working with a periodic time distribution (for modulo operation) with a non-periodical start region or if the results in time are shifted far outside of the size of the 1D time histogram.



The value for “Time Offset” has to be expressed in channels of the total time sum (T DLD)/4. It can also be defined via the variable “time_offset” in the dld_gpx3.ini file (see chapter 3).

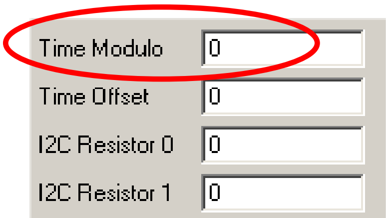
2.13.2 Modulo Operation

The GUI software offers the possibility of overlapping periodical time structures into one defined time window to form one single time structure. The modulo operation is essential when working with a frequency divider for the external start signals, as the use of a frequency divider results in a multiple time histogram.

Note

The USB2.0-TDCs are not operating with start signals of frequencies larger than 7 MHz. Larger start pulse frequencies have to be reduced by using an appropriate frequency divider (for further details see your delay line detector manual).

Figure 24 gives an example for a 16-fold periodical time structure (e.g. created by a frequency divider with dividing factor of 16). Modulo operation can now be used to overlap all time structures to form the original single one.
The value of the modulo-operation can be entered into the corresponding entry field “Time Modulo” in the GUI Main Interface. Later on the value of the variable “time_modulo” in the dld_gpx3.ini file (see chapter 3) can be set directly.



The value of the modulo-operation corresponds to the period of a measured time distribution and is entered in picoseconds (ps). It can be calculated in two different ways:

1.) Measure a time distribution of the total time sum “T DLD” (modulo operation turned off). Identify the average distance of a major structure, which is repeating periodically and multiply this value with a factor of 32. The result is the modulo-value.

For example: A distance of 54713 was measured between the first and the 16th peak for the multiple time histograms (black ones) given in Figure 24. This value is divided by 15 to calculate the average distance between two histograms, and then multiplied by 32. The result is a modulo-value of 116722.

2.) You can begin directly with the period of external start signal. Divide this period by the time bin size of the total time sum "T DLD" (e.g. 6.75 ps) and multiply this value again by 32. The time bin size of the total time sum "T" is given in the GUI Main Interface in the header of the "T (6.75ps/Ch(X+Y))" column, or is displayed when opening the 1D time histogram.

For example: The external start signal goes from a 40 MHz laser. The period of the 40 MHz is 25ns. 25ns divided by 6.75 ps and multiplied by 32 gives a modulo value of 118519,

The first method is more accurate and should be preferred. In most cases a fine adjustment of the modulo-value is necessary and therefore highly recommended. Modify the value slightly and check the width of the time histogram. The best modulo value should result in the smallest width.

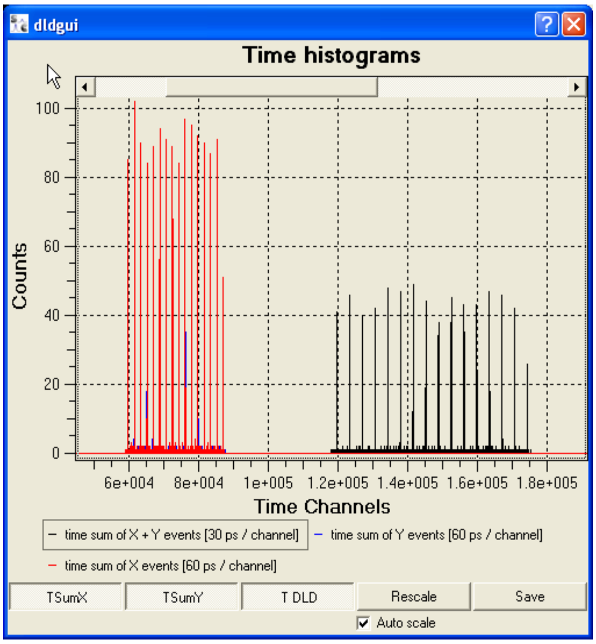


Figure 24: Time histograms measured with a 16-fold divided start signal

Note

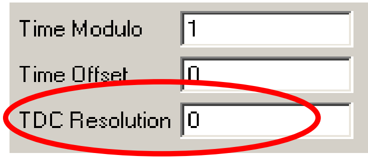
The modulo-operation is turned off by entering a value of 0.

2.13.3 TDC Resolution

When starting the High Resolution-, the Dual Channel- or the Quad Channel- USB2.0-TDC (devices operating in the R mode), a so called PLO loop (an internal TDC measurement routine to reach the 27 ps digital time bin resolution) is started. It can happen that the PLO loop is not locked after starting the TDC. This depends mainly on minor fabrication variations of the TDC chips itself. The TDC gives out a warning message that the resolution is not reached, if the PLO loop does not lock. The warning message appears after pressing "exposure" for the first time after the TDC is turned on.
If the PLO loop does not lock, turn off the TDC, wait a couple of seconds and turn it on again. If the PLO loop still does not lock, increase the digital time bin size slightly by adjusting the value of the variable "time_precision" in the dld_gpx3.ini file (see chapter 3). This variable is given in percentage of an initial value for the internal definition of the digital time bin resolution named "hsdiv". The proportionality between the digital time bin resolution and hsdiv is given by:

Bin $\propto$ 1 / hsdiv

When entering a value of smaller than 100, this will decrease the initial value for hsdiv and will increase the digital time bin resolution.
Alternatively the value for "time_precision" can be changed by entering the value in the "TDC Resolution" entry field of the GUI Main Interface.



2.13.4 I2C Resistor 0/ 1

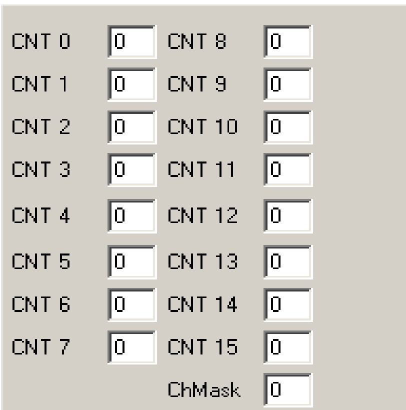
Some single devices (TDC and/or analogue readout electronics) are equipped with electrical potentiometers (e.g. for high voltage power supply control). The communication to these potentiometers is realized via I2C. The resistor values are set via the I2C Resistor entry fields. If not especially stated in the operating manual of the DLD, your devices are **not** equipped with electrical potentiometers and these entry fields are of no functionality.

Note

The I2C connection has to be turned on in the dld_gpx3.ini file explicitly for a proper functioning (see chapter 3).

2.13.5 Stop Channel Correction Values and Channel Deactivation

The GUI software offers the possibility to enter additional correction values individually for each stop channel. This correction value is added to the time result, which has been measured by the TDC (e.g. to compensate different cable delays). Correction values for 16 channels (0 - 15) can be entered in total. Entering can be done either directly in the dld_gpx3.ini file (see chapter 3) or in the corresponding entry fields in the GUI Main Interface after being activated in the dldgui.ini file.



The entry fields for the correction values are connected with the different stop channels as follows (depending on TDC type):

USB2.0-TDC (8 stop channels)	
X1 (Channels -1/1 in rate meter)	0/ 4 (free of choice, 0 is recommended)
X2 (Channels -5/2 in rate meter)	1/ 5 (free of choice, 5 is recommended)
Y1 (Channels -9/3 in rate meter)	2/ 6 (free of choice, 2 is recommended)
Y2 (Channels -13/4 in rate meter)	3/ 7 (free of choice, 7 is recommended)

Double USB2.0-TDC (16 stop channels)	
X1 (Channels -1/1 in rate meter)	0/ 4/ 8/ 12 (free of choice, 0 is recommended)
X2 (Channels -5/2 in rate meter)	1/ 5/ 9/ 13 (free of choice, 5 is recommended)
Y1 (Channels -9/3 in rate meter)	2/ 6/ 10/ 14 (free of choice, 10 is recommended)
Y2 (Channels -13/4 in rate meter)	3/ 7/ 11/ 15 (free of choice, 15 is recommended)
High Resolution USB2.0-TDC (4 stop channels)	**Entry field for delay time**
X1 (Channels -1/1 in rate meter)	0
X2 (Channels -5/2 in rate meter)	4
Y1 (Channels -9/3 in rate meter)	8
Y2 (Channels -13/4 in rate meter)	12
Dual Channel USB2.0-TDC (2 stop channels)	**Entry field for delay time**
X1 (Channels -1/1 in rate meter)	0
X2 (Channels -5/2 in rate meter)	4
Quad Channel USB2.0-TDC (4 stop channels)	**Entry field for delay time**
X1 (Channels -1/1 in rate meter)	0
X2 (Channels -5/2 in rate meter)	4
Y1 (Channels -9/3 in rate meter)	8
Y2 (Channels -13/4 in rate meter)	12

Single channels can also be deactivated by entering a corresponding value in the entry field “ChMask” of the GUI Main Interface or alternatively for the parameter “Chmask” in the dldgui.ini file (see chapter 3). Channels will be deactivated by entering “1” in a bit mask. A “0” entry activates the corresponding channel.

Note

Not the bit mask itself, but its corresponding decimal number has to be entered in the field “ChMask”.

For example: The bit mask for all channels activated is 0000000000000000 and corresponds to 0.
The bit mask for the channels 0, 2, 12 and 15 is 1010000000001001 and corresponds to 40969.
The bit mask for all channels deactivated is 1111111111111111 and corresponds to 65535

2.13.6 Q-Shift entry fields (4-Quadrant Detectors only)

The Q-shift values are needed for adjustments of the Surface Concept 4-Quadrant Delay Line Detectors only. They do not have any relevance to other DLD types. These values are adjusted to the best performance of the 4-Q DLD and should not be changed.

3 The INI Files

3.1 TDC and GUI configuration using “INI” files

The configurations of the TDC and the GUI are adjusted in the files dld_gpx3.ini and dldgui.ini. These files are located in the same folder as the GUI executable file. Both ini-files are loaded once each time the GUI software starts. Changes to the ini-files do not have any effect while the GUI is running. Close and restart the GUI software. Make sure to save the ini-files after a change of an entry. Otherwise the change is not implemented in the GUI after restart.
The dld_gpx3.ini file deals with the configuration of the TDC (e.g. TDC types, TDC chip operation modes, data formats etc.) as well as with the configuration of the FPGA programming and operation (e.g. RoI settings, data pre-arithmetic modes etc.) and is therefore the more important file.

Note

The dld_gpx3.ini is very complex. It contains entry fields for many different devices in different operation modes. Not all entries are relevant for a single device and therefore may have no meaning for your particular device.
Due to the constant development of our software packages for different devices your specific ini file can contain more entries than described in this manual.
It takes some time and effort to get a complete overview over the relevant entries in the ini file due to its complexity. Therefore it is recommended to create a backup file before changes to the ini-file are made.

The dldgui.ini file is used to activate/deactivate certain entry fields of the GUI Main Interface as well as to adjust image display functions. The dldgui.ini is not as complex as the dld_gpx3.ini file.

Note

For the dldgui.ini file: the structure and number of the entries must not be changed, particularly do not add any character at the end of the last entry. Do not change the line position of any entry; the program only takes the entries at pre-determined line positions.

3.2 Description of the dld_gpx3.ini file

Parameter and Entry example | *Parameter Description*

device	
class = 1	The TDC can be equipped with different FPGA boards, depending on the customer specific application. Those different setups are distinguished by the software via this variable.

	DO NOT CHANGE
sn = ""	This entry field allows allocating individual information (e.g. a serial number) into the TDC. This can be used for example to allow the software to distinguish between different TDCs in case that one is working with two or more TDCs in parallel. DO NOT CHANGE
dev_index = 0	This switch is used to write individual information into the TDC. DO NOT TOUCH
control	
iPairNumber = 2	Measurement results pass through a pre-arithmetic calculation within the FPGA of the USB2.0-TDC, (2) reduces data load over the USB by pre-calculating the time differences and sums within the TDC device (the so called pair mode). This mode restricts the usable value range of time measurements to 3.8 µs. If set to (0) than calculations of time differences are done within the GUI software (the so called raw data mode).
time_aperture = 0x3FFF	Quadruple finder condition for raw data mode: bit 13 (the left 2) switches on/off, the number behind is channel size aperture. Note: The time aperture is active independent off the time aperture being switched on or off, when the multihit algorithm has been turned on (see below). IN RAW DATA MODE ONLY
aperture_offset = -98	Offset for quadruple finder condition for raw data mode IN RAW DATA MODE ONLY
ext_trigger = 0	Activates BNC synch trigger input for hardware triggered exposures when set to (1). Trigger signal must be TTL or LVTTL. DO NOT MISTAKE WITH EXT. START (see TDC section)
iterations = 1	Sets the number of multiple exposures. Allows taking multiple exposures at the TDC hardware level (reduces the dead times produced by the additional start signals from the software)
HighSpeedCamMode = 0	Time axis base taken from exposure time using multiple exposure iterations
ImgAutoSave = 0	Activates auto-save mode when set to (1). TIF files are saved for each exposure automatically.
counter_read = 0	For diagnostics of multi-anode DLDs with segments only.
Histo_1Dx8 = 0	
NumChannels = 0	FOR MULTI-ANODE DLDS WITH SEGMENTS ONLY
ChannelBinning = 16	FOR MULTI-CHANNEL DLDS ONLY
flim_mode = 0	Activates readout mode for optical FLIM detectors when set to (1). FOR FLIM DETECTORS ONLY
debug = 0	If activated (1), an additional window opens after the start of the GUI. This window contains information for diagnostic procedures. It also provides additional diagnostic data within the GUI software. FOR DIAGNOSTICS ONLY

DebugLevel = 0	Defines the debug level. Different debug levels contain different additional information. 0 is the lowest debug level. FOR DIAGNOSTICS ONLY AND FOR WRITING BULK FILES
simulation = 0	Allows reading bulk files from the hard disk if turned on (1) and can be used for data analysis with the GUI software belated. It does not work with connected hardware. See chapter 2.12 for more details. IN RAW DATA MODE ONLY. NEED BULK FILES.
Listmode = 0	Activates the listmode (counts data channel) when set to (1). Allows writing the results from the DLD_DATA_COUNTS CHANNEL into the memory buffer of a user specific readout software. The memory to which the results are written to must be assigned by the users software beforehand. FOR DLD_DATA_COUNTS CHANNEL ONLY (see interface description)
Listmodefile = 0	Listmode files are wirtten when set to (1).
SingleLMFPerExposure = 0	Activates writing of separate listmode files for each hardware triggered exposure when set to (1) (each exposure one file).
GUI_LMF_default_on = 0	The default configuration of the LMF (list mode file) checkbox in the GUI main interface is on when set to (1). FOR DLDGUI SOFTWARE ONLY
ListModeAsBinned = 1	Data as binned are written into the listmode file when set to (1).
ListModeAsIn_ROI = 1	Only data within the defined RoI are written into the listmode file when set to (1).
ListModeFilePath = "c:\\temp"	Defines the file path for the listmode file.
LMF_data_number = 1000000	Defines the transfer block size for the listmode. FOR DLD_DATA_COUNTS CHANNEL ONLY (see interface description)
TimeStampMode = 0	Activates writing of TDC timestamp into the TimeSumX of the counts buffer (counts TDC-restarts since exposure start) RELIABLE FOR LOW RATES ONLY
MultiPulseMode = 0	FOR PARTICULAR "MULTI-PULSE" HIGH RESOLUTION DEVICES ONLY
SaveBulk = 0	Allows saving TDC stream into "bulk.txt". Works only if debug is set to (1). See chapter 2.10 for more details.
MulitHitMode = 1	Turns on the multihit algorithm when set to (1). Note: Switching on the multihit algorithm also activates the time aperture as defined above, independent on the time aperture being switched of or not. IN RAW DATA MODE ONLY
MulitHitDepth = 2	Sets the depth of the multihit recognition algorithm. A HIGH DEPTH EFFECTS THE PROCESSING TIME SIGNIFICANTLY
TDC_F1 = 0	Set to (1) when working with a F1 TDC.
F1_highres = 0	Set to (1) when activating the highres mode for the F1 TDC.
histograms	

image_size_x = 0x4000	Defines the image size in x. Depends on detector readout, SHOULD FIT TO DATA_FORMAT
image_size_y = 0x4000	Defines the image size in y. Depends on detector readout, SHOULD FIT TO DATA_FORMAT
image_size_t = 0x80000	Restricts time slicing due to memory limits. Depends on detector readout, SHOULD FIT TO DATA_FORMAT
1D_histo_size = 0x20000	Number of time channels used for the x,y-integral time histogram.
min_roi_offs_x = 0	Restricts RoIs.
max_roi_size_x = 0x4000	Restricts RoIs.
min_roi_offs_y = 0	Restricts RoIs.
max_roi_size_y = 0x4000	Restricts RoIs.
min_roi_offs_t = 0	Restricts RoIs.
max_roi_size_t = 0x20000	Restricts RoIs.
PixelSizeX = 0.02222	Pixel size in x given in mm. Corresponds to the value given in the specification sheet. DETECTOR SPECIFIC VALUE
PixelSizeY = 0.02425	Pixel size in y given in mm. Corresponds to the value given in the specification sheet. DETECTOR SPECIFIC VALUE
ChannelSizeT = 0.006858	Channel size in T given in ns. TDC SPECIFIC VALUE
RealSegmentSize = 0.0	1D DLD specific real segment size in mm. DETECTOR SPECIFIC VALUE
time_precision = 100	TDC resolution in %.
time_asymmetry = 0	TDC resolution asymmetry GPX1 and GPX2 (double TDC only, when positive GPX1 gets higher resolution, GPX2 lower, max. +-30%).
time_modulo = 0	Modulo value (should be channels per period * 32, 32 is the default modulo resolution factor, see below).
time_offset = 0	Defines an offset which is removed from the time distribution (in timesum channels/4, default is set to correct an artificial hardware offset value).
modulo_resolution_factor = 32	Modulo resolution factor to be read out with API function DLD_GET_DEVICE_INFO.
SumDif_Modulo_OnPC = 0	Activates the modulo operation on PC. For firmware versions without any build-in modulo functions in pair mode.
max_time_channels = 0x7FFFF	Maximum number of time channels available from measurement device (0x7FFFF for single TDC, 0x01FFFFFF for double TDC).
max_timehist_len = 0x7F000	Maximum recommended number of time channels to use in time

	histograms.
mirror_x = 0	Mirrors the image at vertical middle axis when set to (1). IN RAW DATA MODE ONLY
mirror_y = 0	Mirrors image at horizontal middle axis when set to (1). IN RAW DATA MODE ONLY
SumY_versus_Y = 0	Plots the time sum of Y versus the y axis (special display of measured data) when set to (1). FOR DIAGNOSTICS AND IN RAW DATA MODE ONLY
SumX_versus_X = 0	Plots the time sum of X versus the X axis (special display of measured data) when set to (1). FOR DIAGNOSTICS AND IN RAW DATA MODE ONLY
cal_xyt = 0	Activates a 3D linearity correction when set to (1) (requires detector depending correction data set). See chapter 0 for detailed information.
cal_smooth = 0	Activates an x/y smoothing algorithm after the x/y calibration when set to (1).
cal_intensity = 0	Activates an intensity/ flat field correction when set to (1) (requires detector depending data set). See chapter 0 for detailed information.
TDC	
Ext_Gpx_Start = 0	The TDC uses its internal clock for time measurements (imaging only) when set to (0). Set to (1) when using external start signals for time resolved measurements (3D mode). See chapter 2.8 and the DLD manual for detailed information.
Start_Falling_Edge = 0;	The TDC start takes reference from falling edge when set to (1). Otherwise it takes reference from rising edge.
Data_Format = 2	Different formats for data transfer. The following bits are reserved for x/y differences and for t sums, when data format is set to: (0) 9bit/ 9bit/14bit, (1) 10bit/10bit/12bits, (2) 11bit/11bit/10bits, (3) 12bit/12bit/8bit. See chapter 2.2 for detailed information. PAIR MODE ONLY
ResynchronTime = 0	Activates the soft reset time of GPX chips for re-synchronization in ms when set to (1) (checks for gathered exposure times, communication overhead times do not count).
TTL_Inputs_R_mode = 0	Allows using TTL inputs for High Resolution TDC, Dual Channel TDC and Quad Channel TDC (multi-channel DLDs) when set to (1).
ChronoStack = 0	Activates the chronostack algorithm of the TDC when set to (1) (off may improve multihit capability).
Data_Flow_Off = 0	Allows to deactivate the "normal" data stream when set to (1) and to keep the service data in stream only (for virtual segment mode).
;firmware = ".\usb3gpx_LimitedData_NoDCM5.rbt" firmware = "usb3gpx_R_new.rbt" ;firmware = "usb3gpx_vs.rbt" ;firmware = "usb3gpx8m.rbt" ;firmware = ".\usb3gpx_1d.rbt"	Assigns the firmware for the TDC. The firmware without the ";" at the beginning is the current firmware in use DO NOT CHANGE

;firmware = ".\usb3gpx_NoDCM_I.rbt" ;firmware = ".\usb3gpx4q_old.rbt" ;firmware = ".\usb3gpx.bit"	
SimulationDataFile = "bulk_final.txt"	Defines the file which is used for simulation. See chapter 2.12 for detailed information.
BulkDataFile = "bulk.txt"	Defines the file name for the bulk file. See chapter 2.10 for detailed information.
TimeCalDataFile = "t_cal.txt"	Defines the file which is used for time calibration. See chapter 0 for detailed information.
TimeCalRun = 0	
RDN_WRN_Adjust = 1	Enables individual adjustments for TDC optimization. DO NOT CHANGE
RDN1Start = 2 RDN1Stop = 4 RDN2Start = 2 RDN2Stop = 4 WRNStart = 2 WRNStop = 2	Individual adjustments for TDC hardware. DO NOT CHANGE
One_Gpx = 0	Setting depends on the hardware layout of the TDC. Is set to (1) when the TDC is equipped with one GPX-TDC chip and to (0) when the TDC is equipped with two GPX-TDC chips. DO NOT CHANGE
I_Mode = 0	Setting of TDC mode. Activates the I mode when set to (1). Otherwise the TDC is running in R mode. INTERFERENCE WITH G_MODE = 1
G_Mode = 0	Setting of TDC readout mode. Activates the G-Mode when set to (1). Otherwise the TDC is running in R mode. INTERFERENCE WITH I_MODE = 1
G_x1_corr = 515 G_x2_corr = 485 G_y1_corr = 510 G_y2_corr = 490 G_x_slope = 180 G_y_slope = -30 G_slope_threshold = 0;350	Correction values for G mode given in thousands in the time interval between rising and falling edge. DO NOT CHANGE
The next parameters control cutoffs in pulse height distributions for two channels A and B (0: x1, 1: x2, 2:y1, ; 3:y2)	
G_A_Channel = 2	channel 0,1,2,3
G_A_LeftCut = 0	;set to 0 to open
G_A_Width = 1000	;set to 500 to open
G_B_Channel = 3	;channel 0,1,2,3
G_B_LeftCut = 0	;set to 0 to open

G_B_Width = 1000	;set to 500 to open
X_Correct = 100	percentage of image X coordinate re-scaling RAW DATA MODE ONLY
Y_Correct = 100	percentage of image Y coordinate re-scaling RAW DATA MODE ONLY
StartOff1 = 0x2710	TDC Register 5 StartOffset In R-Mode x3 DO NOT CHANGE
RefClkDiv = 7	BIN(I-Mode) = (25ns x 2^RefClkDiv) / (216 x HSDiv) DO NOT CHANGE
HSDiv = 180	BIN(R-Mode) = BIN(I-Mode)/3; BIN(G-Mode) = BIN(I-Mode)/2 DO NOT CHANGE
StartTimer = 512	StartPeriod = StartTimer * 12.5 ns DO NOT CHANGE
ReferrenceMeasurement = 0	Activates the extended time range mode when set to (1). I MODE ONLY
ReferenceChannel = 6	Defines the 0 - 15 reference channel for extended time range.
ExtentedTimeRangeBinning = 14	Time base = 2^(this number[0..20]) * 20.5 ps (14 means (2^14)*20.5 ps = 0.336 µs per time step)
SetTimeRangeAsImageOrdinate = 0	Activates the x-t image mode in extended time range mode when set to (1).
AluTriggerPartialReset = 0	Enables partial GPX reset when set to (1). The AluTrigger must be wired in the hardware and supplied via "Synch In". WITH SPECIAL HARDWARE WIRING ONLY
ExtendedTimeRange = 0	Activates the extended time range mode when set to (1). CLASS2 DRIVER IMPLEMENTATION ONLY (ask back or set to 0)
SeparationNumber = 20000	
SeparationTimer = 40	
Hexanode = 0	Roentdek Hexanode readout mode. RAW DATA MODE ONLY
DLD_1D = 0	FOR DIAGNOSTICS OF MULTI-CHANNEL DETCTORS ONLY
TimeSlicing = 0	Activates the time-slicing detector mode when set to (1) (maximum 16 time slices (x,y) in one image). FOR RAW DATA MODE ONLY
TimeSliceWindowSize0 = 0x1FFFF	FOR TIME-SLICING MULTI_CHANNEL DETECTORS ONLY
TimeSliceWindowSize1 = 0x1FFFF	
TimeSliceZero0 = 0x0	
TimeSliceZero1 = 0x20000	
Ch0 = 0	Stop channel correction values in time bin channel units, for an individual DLD adjustment. See chapter 0 for detailed information.
Ch1 = 0	

Ch2 = 0	
Ch3 = 0	
Ch4 = 0	
Ch5 = 0	
Ch6 = 0	
Ch7 = 0	
Ch8 = 0	
Ch9 = 0	
Ch10 = 0	
Ch11 = 0	
Ch12 = 0	
Ch13 = 0	
Ch14 = 0	
Ch15 = 0	
chmask = 0x0000	Switch-off mask for stop-channels. See chapter 0 for detailed information.
CommonShift = 0	
TimeDif1Min = 0xF000	Defines parameters for FPGA quadruple finder. DO NOT CHANGE
TimeDif1Min = 0x0FFF	
TimeDif2Min = 0xF000	
TimeDif2Max = 0x0FFF	
SumDifMin = 0xFF00	
SumDifMax = 0x00FF	
QDLD	
4Q_Dld = 0	Activates the 4 quadrant detector mode when set to (1) FOR 4-QUADRANT DLDS ONLY
shift_x1 = 110	Defines the positioning of the 4 single quadrants of a 4 quadrant detector. FOR 4-QUADRANT DLDS ONLY
shift_y1 = -55	
shift_x2 = -70	
shift_y2 = -70	
shift_x3 = 78	
shift_y3 = 83	
shift_x4 = -115	
shift_y4 = 75	
min_roi_offs_x = 145	
max_roi_size_x = 220	
min_roi_offs_y = 210	
max_roi_size_y = 105	
q1_left_limit = -650	Defines the quadrant cut-offs (currently raw data mode only) FOR 4-QUADRANT DLDS ONLY
q1_lower_limit = 600	
q2_right_limit = 600	
q2_lower_limit = 600	
q3_right_limit = 600	
q3_upper_limit = -600	
q4_left_limit = -650	
q4_upper_limit = -600	
adj_x8 = 100	Percentage of coordinate stretching or compressing, careful use: that may move quadrant imaging positions. FOR 4-QUADRANT DLDS ONLY
adj_y8 = 100	
adj_x9 = 100	
adj_y9 = 100	
adj_x10 = 100	
adj_y10 = 100	
adj_x11 = 100	

adj_y11 = 100	
shift_t8 = 0 shift_t9 = 0 shift_t10 = 0 shift_t11 = 0	Time channel correction for quadrant time offset correction. FOR 4-QUADRANT DLDS ONLY
sn8 = "11050000WX" sn9 = "11050000X2" sn10 = "11050000V7" sn11 = "11050000VT"	Serial numbers of the 4 individual TDC units within the 4-fold Quad Channel USB2.0-TDC (to be used to allow the software to distinguish between the different TDC units). ONLY RELEVANT WHEN OPERATING A 4-QUADRANT DLD WITH THE 4-FOLD QUD CHANNEL USB2.0-TDC. DO NOT CHANGE.
SimulationDataFile8 = "bulk8_raN.txt" BulkDataFile8 = "bulk8.txt" SimulationDataFile9 = "bulk9_raN.txt" BulkDataFile9 = "bulk9.txt" SimulationDataFile10 = "bulk10_raN.txt" BulkDataFile10 = "bulk10.txt" SimulationDataFile11 = "bulk11_raN.txt" BulkDataFile11 = "bulk11.txt"	Defines the file names for the 4 simulation files and the 4 bulk files of the 4 TDC units within the 4-fold Quad Channel USB2.0-TDC (see chapter 2.10 for detailed information). ONLY RELEVANT WHEN OPERATING A 4-QUADRANT DLD WITH THE 4-FOLD QUD CHANNEL USB2.0-TDC.
TimeDif1Min8 = -730 TimeDif1Max8 = 620 TimeDif2Min8 = -700 TimeDif2Max8 = 690 SumDifMin8 = -20 SumDifMax8 = 20 TimeDif1Min9 = -630 TimeDif1Max9 = 640 TimeDif2Min9 = -660 TimeDif2Max9 = 730 SumDifMin9 = -20 SumDifMax9 = 20 TimeDif1Min10 = -730 TimeDif1Max10 = 64 TimeDif2Min10 = -750 TimeDif2Max10 = 750 SumDifMin10 = -20 SumDifMax10 = 20 TimeDif1Min11 = -700 TimeDif1Max11 = 600 TimeDif2Min11 = -780 TimeDif2Max11 = 600 SumDifMin11 = -20 SumDifMax11 = 20	Parameters for FPGA quadruple finder. 4 times for each of the 4 individual TDC units within the 4-fold Quad Channel USB2.0-TDC. ONLY RELEVANT WHEN OPERATING A 4-QUADRANT DLD WITH THE 4-FOLD QUAD CHANNEL USB2.0-TDC. DO NOT CHANGE.
RDN1Start8 = 3 RDN1Stop8 = 4 RDN2Start8 = 1 RDN2Stop8 = 5 WRNStart8 = 2 WRNStop8 = 4 RDN1Start9 = 3	Individual adjustments for the 4 TDC units within the 4-fold Quad Channel USB2.0-TDC. ONLY RELEVANT WHEN OPERATING A 4-QUADRANT DLD WITH THE 4-FOLD QUAD CHANNEL USB2.0-TDC. DO NOT CHANGE.

RDN1Stop9 = 4	
RDN2Start9 = 0	
RDN2Stop9 = 5	
WRNStart9 = 2	
WRNStop9 = 4	
RDN1Start10 = 3	
RDN1Stop10 = 4	
RDN2Start10 = 0	
RDN2Stop10 = 5	
WRNStart10 = 2	
WRNStop10 = 4	
RDN1Start11 = 2	
RDN1Stop11 = 5	
RDN2Start11 = 0	
RDN2Stop11 = 5	
WRNStart11 = 2	
WRNStop11 = 4	
I2C	
I2C = 0	Activates the I2C communication when set to [1].
clock_divider = 0x5F	Clock divider value for I2C interface.
HVModuleAddress = 0x80	Address for HV module digital potentiometer, 0x80 = No Module.
HVModuleType = "DS1805"	HV module digital potentiometer type. Possible values: "DS3902" - non-volatile and "DS1805" – volatile.
SegmentsSwitchAddress = 0x40	Address for digital potentiometer for segment switching, 0x80 = No Module.
SegmentsSwitchType = "DS3902"	Segment switching digital potentiometer type. Possible values: "DS3902".

3.3 Description of the dldgui.ini file

Parameter and Entry example	*Parameter Description*
New_Intensity_Calibration 0	The next image is used for intensity calibration when set to 1. Do only start exposure. Image saving is done automatically. Do not forget to set back to 0 after calibration image measurement.
Zoom@Binning 0	Screen output expands to 512x512 pixels when binning provides less than 512 pixels when set to [1]. Setting has no effect on saved data. See chapter 2.4.12 for detailed information.
Zoom_out_x_6 0	A 6 x 6 binning on low level in data acquisition is established for diagnostics when set to [1].
counter_read 0	Enables 9 additional counter (TTL) inputs for extensional use when set to [1]. OPTIONAL HARDWARE DEVICE

refresh_time_ms 2500	Defines the refresh time in ms after which an image is refreshed during a long time exposure. See chapter 2.4.5 for detailed information.
modulo_adjust 0	Activates three additional entry fields in the GUI Main Interface for adjusting time_modulo, time_offset and TDC_resolution when set to (1). See chapter 0 for detailed information.
Channel_adjust 0	Additional entry fields are displayed in the GUI Main Interface for adjusting delay values for Ch 0 to Ch 15 when set to (1). See chapter 0 for detailed information.
small_screen 0	Reduces the GUI Main Interface to fit to computer screens with a smaller resolution than 864 pixel in Y when set to (1). See chapter 2.3.1 for detailed information.
CalibCoord_x1 0 CalibCoord_x2 512 CalibCoord_y1 0 CalibCoord_y2 512	Internal calibration coordinates. DO NOT CHANGE
InitialExposureTime 1000	Sets the default exposure time in ms, which is used when starting the GUI software.

4 Delayline Detector DLL

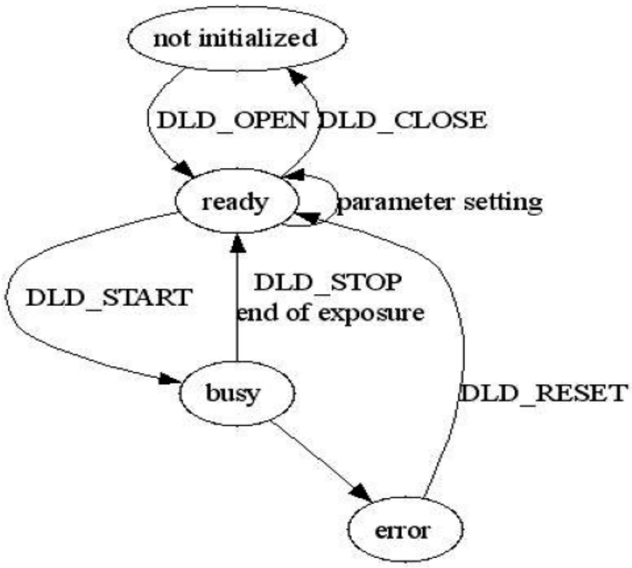
4.1 The basic working principle

The basic operating principle of the driver (delayline.dll) is:

- The user opens the driver and gets an ID for the use of the device.
- The measurement mode and its corresponding parameters are settled from the user interface.
- Each measurement action is called "exposure" like for camera systems.
- The user allocates an appropriate memory space for the exposure results and adds this "buffer" to the DLL interface for filling it up at the next "exposure", this exchange is done by exchange of pointers to the allocated memory space ("buffer").
- The DLL sends a windows signal, when the current exposure is done or the user may ask whether the driver is ready with it.
- The DLL releases its access to the "buffer" after the exposure for a defined experiment, so that the user can work with this result buffer.

4.2 Interface Description

4.2.1 Life cycle of the DLL



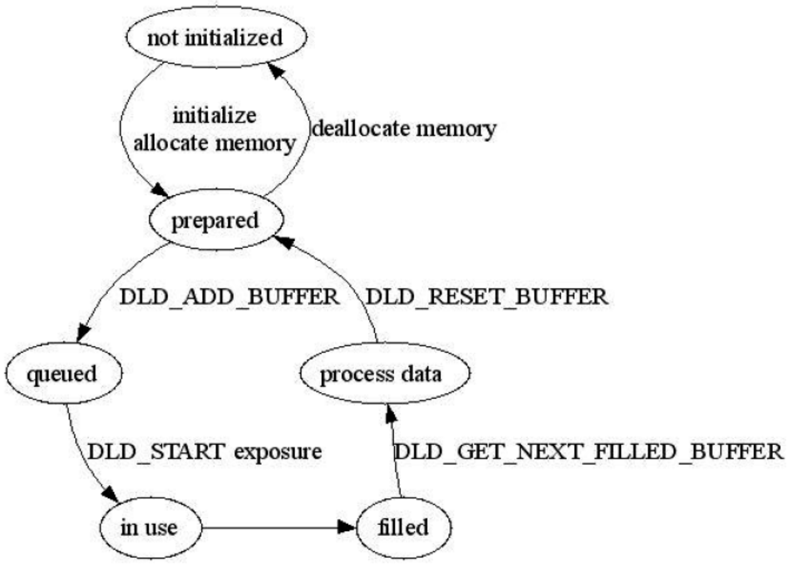
4.2.2 Data model and the process of a measurement

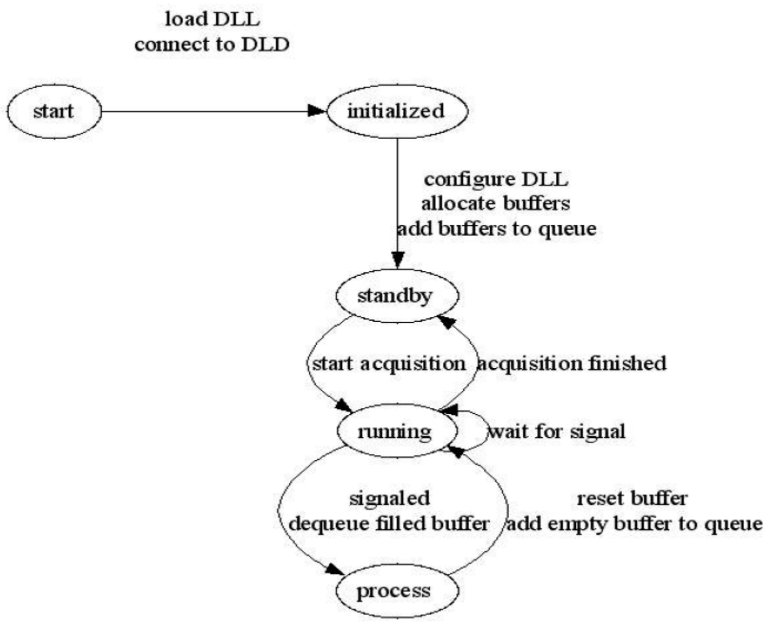
The DLD offers different configurations with different data channels (e.g. image data and channel data). The description of the single channels and the channel ID can be prompted. The required data channels can be selected during the configuration step. The client provides buffer, which contain at least data buffer for the selected channels. Each buffer contains information about its status and about the sizes of the contained data buffer.

The DLL administrates the buffers in queues (FIFOs). The client hands over empty buffers to the DLL with DLD_ADD_BUFFER. These buffers are administrated by the DLL in EmptyBufferQueue. After data acquisition is finished, the first buffer is removed from EmptyBufferQueue, the measured results are added to the buffer and queued to FilledBufferQueue. Data acquisition may be started anew already parallel to this process.

The client is going to inform Event-signaling or Polling that one or more buffer in the FilledBufferQueue are available for processing. The client can collect process and reset the buffers with DLD_GET_NEXT_FILLED_BUFFER and it can send them back to the DLL with DLD_ADD_BUFFER.

Before changing the channel selection, the EmptyBufferQueue must be emptied, to ensure that this Queue does not contain wrong configured buffer.





4.2.3 Image data

The image data channel of the DLD can be configured by binning and setting of a ROI. Doing this, the following relationship has to be considered:

MaxROI = MaxImageSize / Binning

and

ImageSize = ROI

A change of the binning occurs in the reset of MaxROI and with this of the ImageSize.

4.3 API

The functions available for the dll are described in the following. The corresponding syntax is given in the file "delayline.h" and the used data structures for parameters and data exchange buffers are defined in the file "DelayLineDefs.h". Both files can be found on the CD.

Methods for the initializing procedure	
DLD_OPEN DLD_CLOSE DLD_RESET	# explicit initializing and # de-initializing of the DLL # resets to default values
DLD_GET_DEBUG_LEVEL DLE_SET_DEBUG_LEVEL	# gets debug level # sets debug level
Enquiring of information	
DLD_GET_ERROR_STRING DLD_GET_INFO_STRING DLD_GET_DEVICE_TYPE	# delivers an error message to an error code # delivers a description of a specified device # delivers the type of a specified device
Methods for configuration	
DLD_GET_DEVICE_INFO	# delivers a set of device parameters (max. image size, max. numbers of channels, bits per pixel, pixel size in y and y, numbers of detector channels, min. and max. exposure time, min. and max. exposure delay, ..., min. and max. dwell time.
DLD_GET_IMAGE_SIZE	# delivers the size of the active image dependent on ROIs and binning
DLD_GET_IMG_ROI DLD_SET_IMG_ROI	# delivers the momentarily set ROIs # places ROIs
DLD_GET_IMG_BINNING_INFO_MAX DLD_GET_IMG_BINNING_INFO DLD_GET_IMG_BINNING DLD_SET_IMG_BINNING	# number of possible binning values # list of possible binning values # delivers the active binning setting # sets the binning
DLD_GET_EXPOSURE_TIME DLD_SET_EXPOSURE_TIME	# delivers / sets acquisition, waiting and repetition time
DLD_GET_DATA_INFOS	# delivers info of the data available (image, raw image, cannel data, time stamp)
DLD_GET_DATA_SELECTION DLD_SET_DATA_SELECTION	# delivers / set the selection of data; the DLD delivers the selected data to the provided buffers corresponding to the selected option
DLD_GET_IMG_TRANSFORMATION	# delivers the active transformation table for image data
DLD_SET_IMG_TRANSFORMATION DLD_GET_IMG_THRESHOLD DLD_SET_IMG_THRESHOLD DLD_GET_CHANNEL_THRESHOLD DLD_SET_CHANNEL_THRESHOLD	# sets the transformation table for image data # gets the threshold for the delayline # sets the threshold for the delayline # gets the threshold for the channel detector # sets the threshold for the channel detector

Methods for data acquisition	
DLD_START DLD_STOP DLD_ABORT	# starts data acquisition # stops data acquisition # abort of data acquisition
DLD_SUSPEND DLD_RESUME	# suspend / resume a running data acquisition, e. g. at long acquisition times
DLD_GET_STATUS	#
DLD_ADD_BUFFER DLD_REMOVE_BUFFER DLD_GET_BUFFER_STATUS	# adds / removes prepared buffer for the empty buffer queue
DLD_GET_MAX_BUFFERS	# delivers the max. number of buffer queued to the bufferqueue
DLD_GET_EMPTY_BUFFER_NUM	# delivers the numbers of momentarily empty buffer to the bufferqueue
DLD_GET_FILLED_BUFFER_NUM	# delivers the numbers of momentarily filled buffer to the bufferqueue
DLD_GET_NEXT_FILLED_BUFFER	# delivers the first filled buffer from the queue
DLD_RESET_EMPTY_BUFFER_QUEUE DLD_RESET_BUFFER_QUEUE	# deletes the EmptyBufferQueue # deletes all bufferqueues

4.4 DLL Conventions

4.4.1 Calling convention

As default, the subroutines deliver an integer value (32 bit). The return value "0" indicates an error free execution. Negative values indicate errors, positive values indicate warnings. The error codes are defined as macros and are within the DLL well-defined. A textual description is given for every error code via a special routine.

The return values are handed over to the routine by a pointer. Additionally the maximal available memory size is handed over for return values with variable length. In general the client should allocate and clear memory.

4.4.2 Compiler

The DLL is compiled with MS Visual C++ 6.0.
The DLL is compiled as a multithreaded DLL.

4.5 Example of the DLL usage by external software

In the following an example for a user sequence is given in C:

1. *Open the device driver, get an ID:*

```
DLD_ID* dld_id;
DLD_DLL_MODE dll_mode = 1;
   // 1 means simulation mode, 0 is hardware mode
DLD_OPEN_EXT (dld_id, dll_mode);
   // if always hardware is used: DLD_OPEN(dld_id)
```

2. *Optional: ask for parameters, e.g.*

```
dld_properties* dld_prop;
   //see structure in delaylinedef.h
DLD_GET_DEVICE_INFO(*dld_id, dld_prop);
```

3. *Set parameters:*

3.1. TDC parameter initialization (defaults are used if not)

```
dld_dev_params* dev_params;
dev_params.
   //see definitions in delaylinedef.h
DLD_GET_DEV_PARAMS (dld_id, dev_params);
```

3.2. Exposure parameter initialization

```
dld_exposure exp;
exp.iterations      = 1;   //one image
exp.exposure        = 1000; //1000 ms exp. time
exp.delay                    = 0;      //no delay
DLD_SET_EXPOSURE_TIME (dld_id, exposure);
//if needed:
DLD_SET_IMG_ROI (dld_id, dld_image_roi roi);
      // defines region of interest
DLD_SET_IMG_BINNING (dld_id, dld_image_binning binnig);
      // defines binning
```

3.3. Define kind of measurement

```
long selection      = DLD_DATA_IMAGE_INT;
      // the normal imaging mode
DLD_SET_DATA_SELECTION (dld_id, selection);
```

4. Define one or some (i) appropriate buffers for the results, using the "dld_buf" structure:

```
dld_buf* _buffers;
int _bufCount            = exp .iterations;
_buffers                 = new dld_buf[_bufCount];

__int64 imgSize          = (roi.size_x / bin.bin_x) * (roi.size_y / bin.bin_y) *
                           ((roi .size_t/ bin.bin_t) +2);

for ( unsigned int i = 0; i < _bufCount; i++ ) {
  _buffers [i] .buf_id              = i;
  _buffers [i] .prepared_for        = selection;
  _buffers [i] .seq_id              = 0;
  _buffers [i] .timestamp           = 0;
  _buffers [i] .channels            = 0;
  _buffers [i] .image               = new int[imgSize];
  _buffers [i] .raw_image           = 0;
  _buffers [i] .counts              = new dld_counts;
  _buffers [i] .counts->counts      = new dld_count[num_lmf];
  _buffers [i] .counts->len         = num_lmf;
  _buffers [i] .counts->overflow    = false;
  _buffers [i] .iSumHistSize        = 0x10000;
  _buffers [i] .pSumHistX           = new double[_buffers[i].iSumHistSize];
  _buffers [i] .pSumHistY           = new double[_buffers[i].iSumHistSize];
  _buffers [i] .pSumHistXplusY      = new double[2*_buffers[i].iSumHistSize];
  memset( _buffers [i] .image, 0, imgSize * 4);
  memset( _buffers [i] .counts->counts, 0, num_lmf * 10);
  }
```

5. Add the defined buffer to the driver queue:

```
for (unsigned int i = 0; i < _bufCount; i++ )
  DLD_ADD_BUFFER(dld_id, &_buffers[i]);
```

6. Start the exposure

```
HANDLE buf_event;
  // the signaled event
DLD_START (DLD_ID dld_id, buf_event);
```

7. Wait for signal (end of exposure for buf_event) or poll for still running or not running with:

```
DLD_GET_RUNNING (dld_id, int * running);
```

8. Use the result buffer "_buffers [i] .image" freely, it holds now the exposure results and it is now given free by the driver

5 Error Messages

There are different error messages which are given out by the GUI DLD Software under certain conditions:

Both error messages appear following each other.

The GUI Software was not able to detect the TDC.

Check whether the TDC is switched on and/or whether the USB2.0 cable has been connected properly between TDC and PC.

Both error messages appear following each other.

The GUI software is missing the dld_gpx3.ini file. Check the GUI software folder for the dld_gpx3.ini and the dldgui.ini file.
The ini files can always be found on the CD which came together with the DLD.

Contact Surface Concept in case that one of the two ini files cannot be restored.

The GUI software cannot find the TDC firmware file as defined within the dld_gpx3.ini file. Either because the firmware file is missing or because it is not correctly assigned within the dld_gpx3.ini file

Check the dld_gpx3.ini file within the GUI Software folder for the assigned TDC firmware file (see chapter 3 for details). Secondly check for the assigned firmware file within the GUI Software folder.

Contact Surface Concept in case that the firmware file cannot be restored.

6 List of Figure