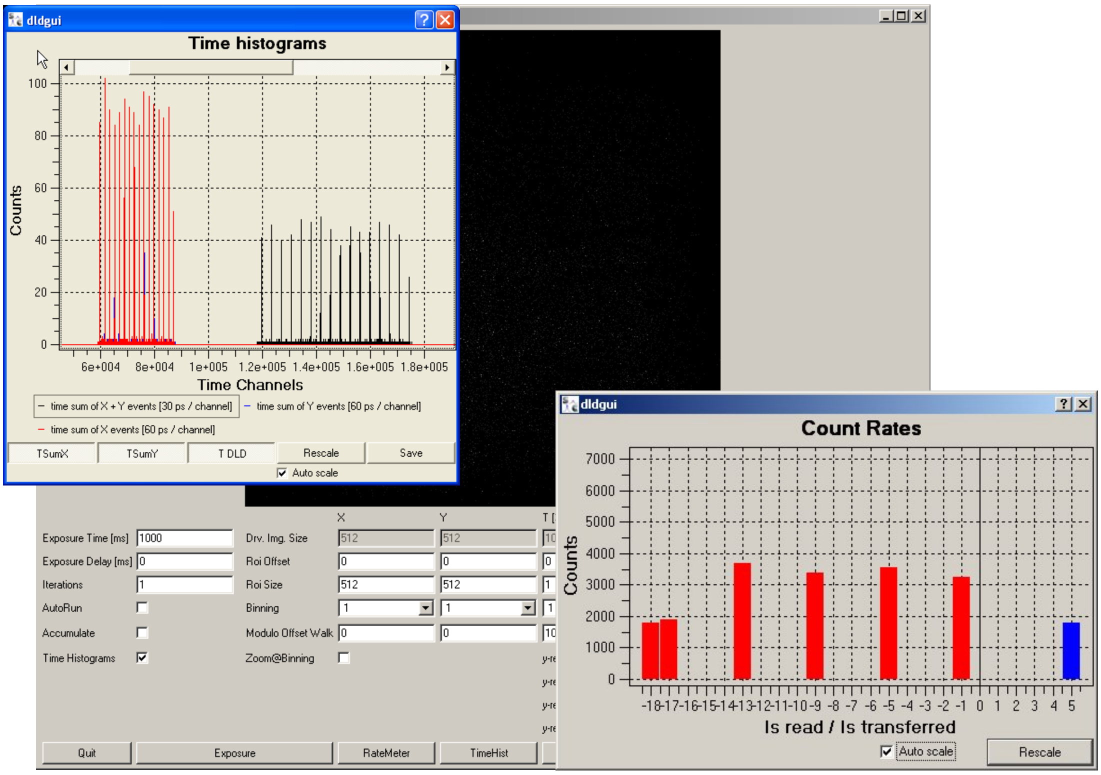
GUI - DLD Software



Manual

Surface Concept GmbH

Staudingerweg 7
55128 Mainz
Germany

Tel. ++49 6131 3923632
Fax: ++49 172 151468621

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

Manual Version: 4.0

Date: 2010-01-04

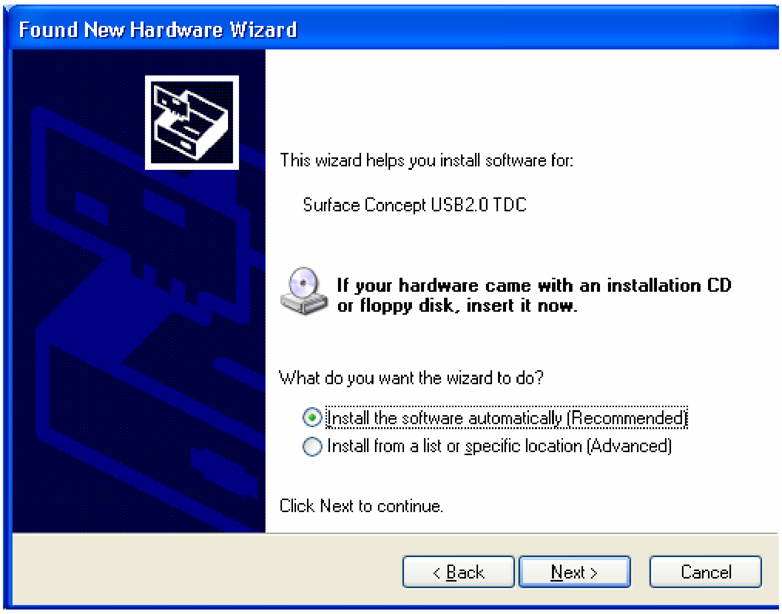
1 Table of Content

2 USB 2.0 driver Installation

Note: **The use of USB2.0 for the readout of the TDC is highly recommended. In principle the readout of the TDC is compatible to USB1.0, but the required data transfer rates are not reached. Do not use PC front panel USB connectors; they are often restricted in performance.**

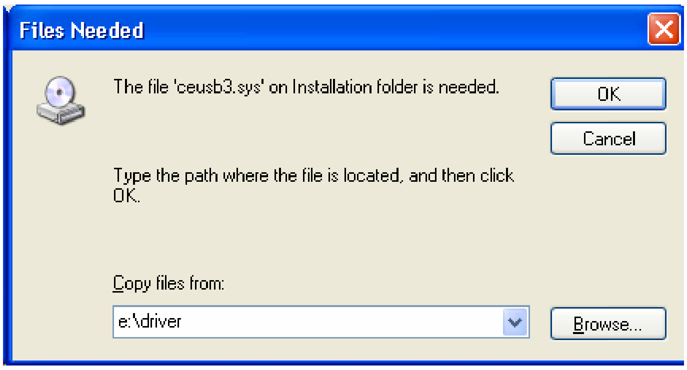
- First, log on as Administrator. Close all applications on your PC. If you are using an anti virus or firewall software, close them (or disable their function). Connect the USB cable of the USB2.0-TDC to your Windows System with USB2.0 turned on. Windows will find a new hardware, and the "Found New Hardware Wizard" will launch. To continue, select “No, not this time” (not looking for windows updates) and click "Next>".

- Insert the CD-ROM, supplied with the delayline detector package into the PC’s CD-ROM drive.
- Select "Install the software automatically (Recommended)" and Click "Next>".

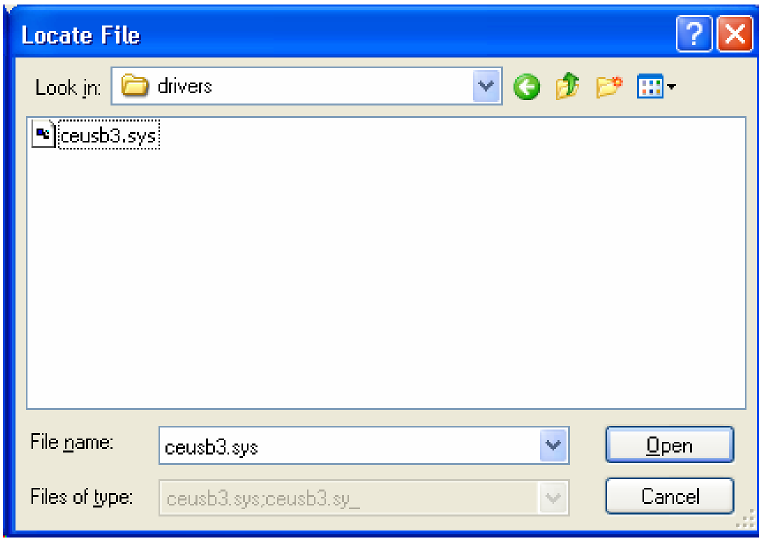


- Continue Installation although the Windows XP capability test failed.

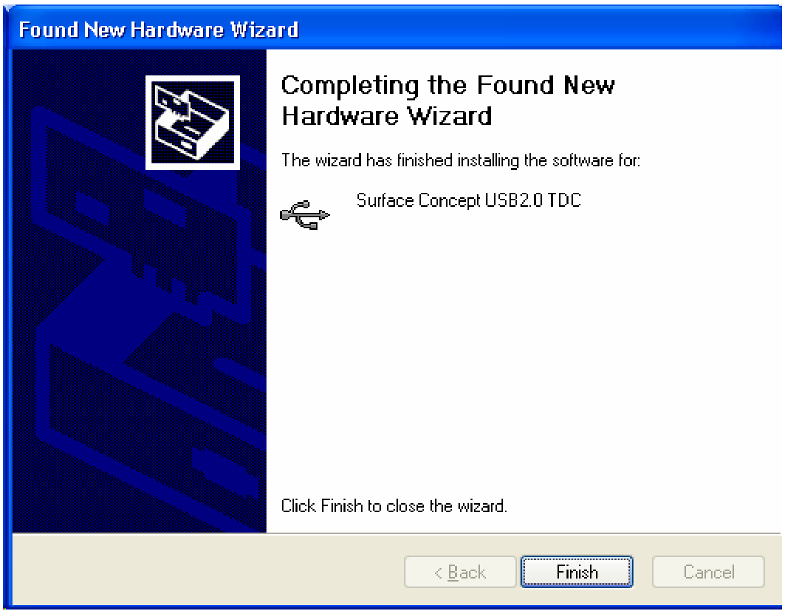
- Enter the path where the driver is located (or browse to it). The driver can be found in the driver folder on the CD-ROM.



- The internal name of the Surface Concept USB2.0 TDC driver is called “ceusb3.sys”. Select it and press “Open”.



- To continue, click "OK". The driver for the Surface Concept USB2.0 TDC will be installed.
- After a few seconds, a finishing dialog should appear as below. To finish, click "Finish".



Note

The installation routine might start again for a second time, after finishing for the first time. Go through the routine again a second time completely. The driver installation will be complete only after the second installation.
The driver has to be installed again, when the USB cable is connected to a different USB port. In this case the driver installation should start automatically. No driver CD will be needed anymore.

3 GUI monitor software

3.1 Software Installation and Hardware Requirements

There is no real installation procedure for the GUI software. Just copy the complete folder "GUI software" to a folder of your choice on the hard disc of your PC. Check the following system requirements which are highly recommended to work with the GUI monitor:

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

After software installation, all parameters are already set for a straight away image acquisition. To start the GUI software double-click the dldgui.exe file in your "GUI software" folder. The GUI software opens up with the window shown in Figure 1. For ending the program click the "Quit" button.

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

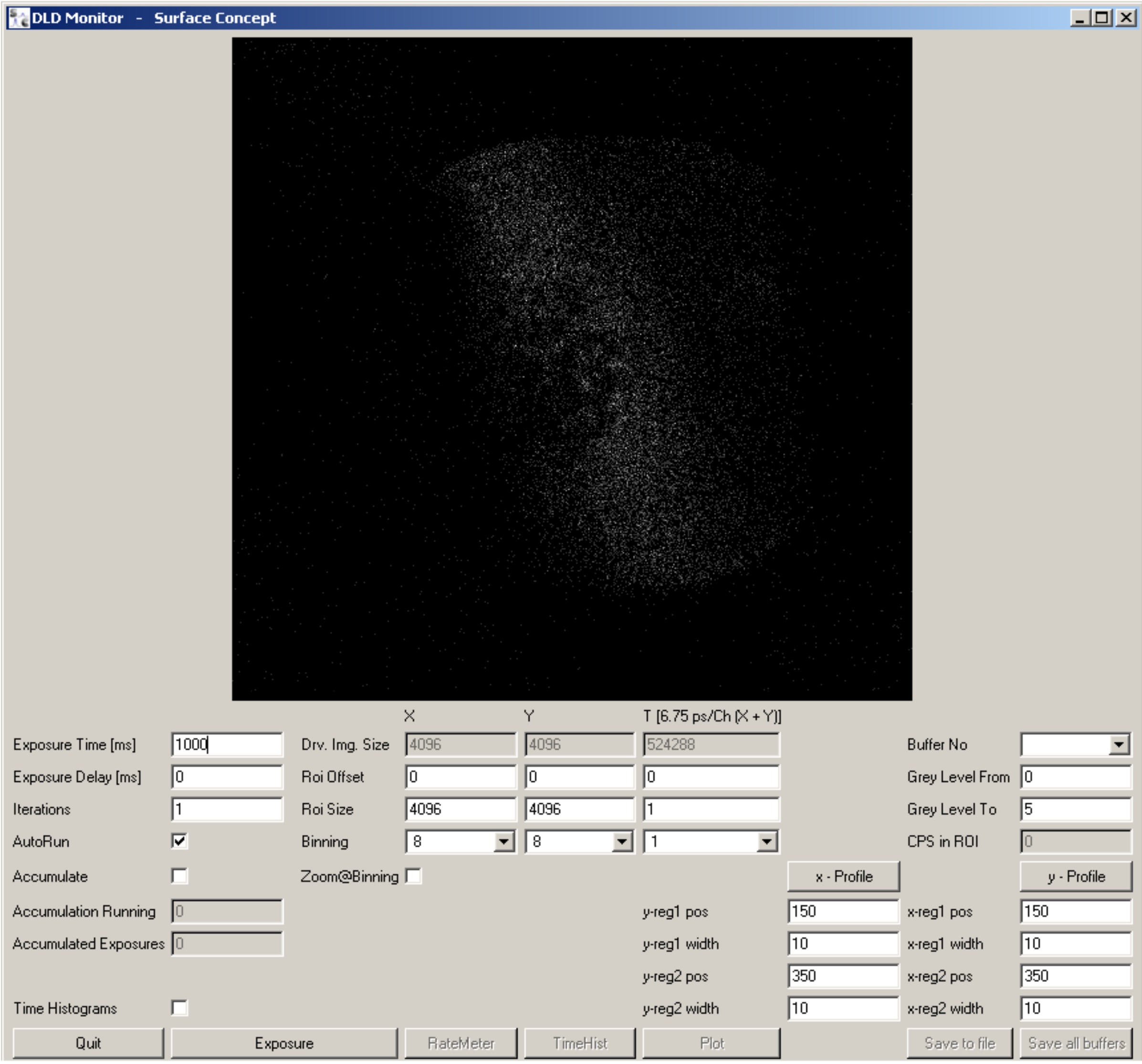


Figure 1: GUI Main Interface

3.2 First Image and basic acquisition functions

3.2.1 First Image

The software acts similar to a CCD camera monitor in terms of image acquisition, exposure times, pixel binning, regions of interest etc.
The exposure time in ms is entered in the field “Exposure Time”. By default, the exposure time is set to 1000 ms and AutoRun mode is activated (the box named “AutoRun” is ticked). 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 4).

Press the exposure button to start the image acquisition. The time in the field “Exposure time” is counting down to indicate the acquisition process.

Note

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

3.2.2 Image Origin

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

3.2.3 Image Update

Image update occurs after the exposure time has ended and every 2500 ms (default value, changeable in the dldgui.ini file, see chapter 4) in case that the exposure time is larger than this value.

3.2.4 Stop Exposure

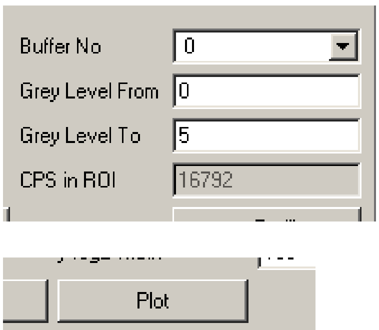
The name of the button "Exposure" changes to the name "Stop Exposure" after the first image update, From now on the exposure can be stopped by clicking on "Stop Exposure". The current image is displayed. If the image appears overexposed or to dark, change the image dynamic range as mentioned in the next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

3.2.5 AutoRun Mode

In AutoRun Mode (default setting), acquisition starts over and over again anew automatically each time the exposure time has ended. The acquired image is displayed until the next image update (each 2500 ms and/or each exposure time). The AutoRun mode is turned on/off by clicking the box named "AutoRun" (turned on when box is ticked).

"AutoRun" can be turned off any time during the acquisition process, but the image acquisition does not stop until the exposure time has ended or "stop exposure" is being clicked. The last image is being displayed.

3.2.6 Dynamic range of displayed image



The dynamic range of each pixel within an image corresponds directly to the number of counts measured during the acquisition time in this pixel.
The default range of the displayed dynamic range 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 or far to dark. In this case you can change the displayed dynamic range by entering higher or lower values in the "Grey Levels" input fields respectively. The displayed dynamic range can be changed during acquisition as well as after acquisition. Press the "Plot" button for the changes to take place, in cases when the dynamic range has been changed after an image has been taken. When the changing occurs during acquisition, the changes take place automatically with the next image update.

Note

Changes of the dynamic range only affect the displaying of images. It does not affect the dynamic range images are recorded in. Images are always saved with a dynamic range of 16 bit.

3.2.7 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 must be pressed additionally, for the changes to take place.

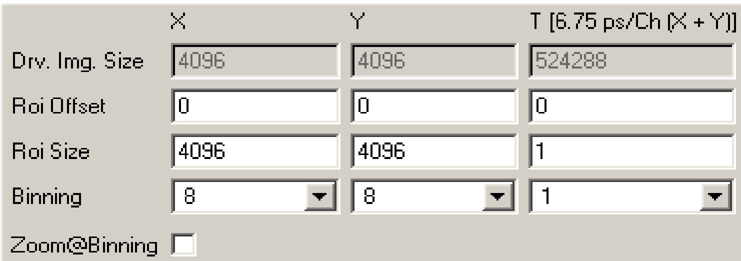
3.2.8 Image Saving

To save an image, click the button "Save to file". A standard Microsoft® window opens. Select the folder to which the image should be saved and enter a filename.

Note

Images are saved as 16-bit tiff files. Adequate software is needed to open them again. The GUI software does not offer the option of reloading images.

3.2.9 Pixel Binning and Zoom@Binning



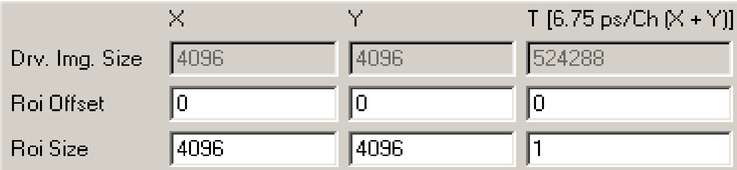
With the "Binning" function, a selected number of neighboring pixels are added together and the sum of their intensities is given out as one pixel. This function can be useful in cases where the image intensity is not high enough for a proper recognition of image structures, especially when working with a lower exposition time. Of course 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".
Pixel binning can also be used to display an enlarged image. Tick the box named "Zoom@Binning". The pixel size of the active image is virtually increased. "Zoom@Binning" can be used during acquisition as well as after acquisition. Press the "Plot" button, when using "Zoom@Binning" on an image after the acquisition process has ended. To return to the normal image display, tick "Zoom@Binning" again and also press the "Plot" button again.

Note

Pixel binning affects the recording of data. It does not only affect the image display (as e.g. the change of the dynamic range). Pixel binning can not be changed afterwards.

Pixel binning cannot be changed while acquisition is running; especially not when accumulation is activated (the memory for an image buffer orientates e.g. on the binning setting and is initiated always with the beginning of an exposure).

3.2.10 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 within the corresponding fields “Roi Size” and “Roi Offset” individually for the X- and the Y-direction. The default values for the RoI are set in the way that it corresponds to the whole image (Roi Size = Image Size, Roi Offset = 0/0).

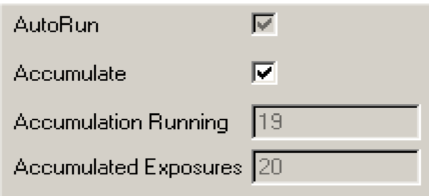
Note

The setting of a RoI affects the recording of data. It does not only affect the image display (as e.g. the change of the dynamic range). It can not be changed afterwards.

The RoI cannot be changed while acquisition is running; especially not when accumulation is activated.

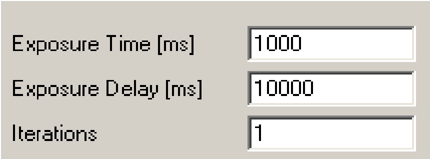
3.3 Additional acquisition functions

3.3.1 Accumulation



When ticking the box “Accumulate” each following measurement is accumulated to the previous ones and 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 the next exposure is started. 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 complete new measurement is started and already accumulated data are being deleted.

3.3.2 Delayed Exposure

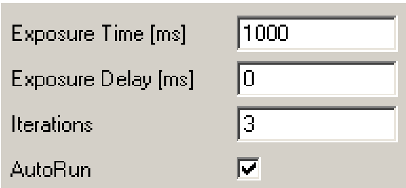


A delay time (in ms) can be entered in the field "Exposure Delay", if acquisition should not start directly but delayed. The elapsing time displayed in the field "Exposure time" is not connected directly to the real exposure time of the hardware. This becomes obvious especially when starting an exposure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

Note

The displayed time elapsed in "Exposure time" during acquisition is based on a software timer and not very accurate. It is rather meant to indicate the exposure process. This elapsing "Exposure time" is not to 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.

3.3.3 Running Average



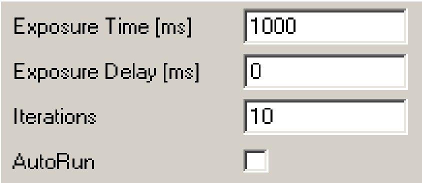
In Auto-Run mode a running average over up to five images can be displayed. Enter the number of images to be averaged into the field "Iterations". A number higher than five will be interpreted as five images automatically.

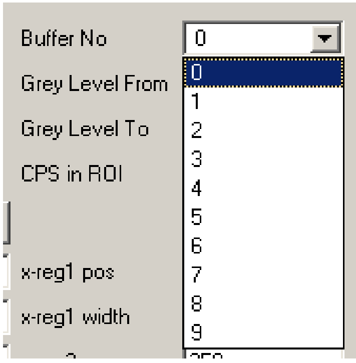
3.3.4 Multiple Exposures

Note

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

To acquire multiple exposures, enter the number of wished exposures into the field "Iterations". Enter a delay time (in ms) in the field "Exposure Delay", if the acquisition of the next image should not start directly after ending of the previous one, but should be delayed. For each image a new buffer is being created automatically and the image is written into it (e.g. 10 buffers are been created, when "Iterations" is set to 10). 1024 iterations can be set at maximum. After the acquisition of multiple exposures has been finished, each image can be displayed by selecting the wished 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.

3.4 Count rate and count rate control

3.4.1 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).

3.4.2 Count rate control with the Rate Meter

The GUI software comprises 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 as been taken. Figure 2 shows a screenshot of the rate meter window. The channels -1 to -16 (red bars) display count rates, which have been measured by the TDC. The count rates of the four detection channels (x1, x2, y1, y2) are displayed in the channels -1, -5, -9 and -13 (for x1, x2, y1, y2). The count rate of matched 4-fold coincidences (calculated from the results of the 4 detection channels) is given in channel -17. Channel -18 shows the count rate of 4-fold coincidences after passing filter elements within the FPGA (e.g. ROI Settings). The channels -17 and -18 display count rates, which have been pre-calculated within the FPGA to reduce the transfer rate over the USB. This happens in the so called "Pair" mode ("iPairNumber" = 2 in the dld_gpx3.ini file, see chapter 4). Typically the count rate in channel -17 is a bit less than the count rate within each of the four single detection channels and the count rate in channel -18 is a bit less than in channel -17.
The channels 1 to 4 (blue bars) display count rates, which have been transferred directly and unprocessed by the FPGA to the PC, where the data processing happens. This is the so called RawData mode ("iPairNumber" = 0 in the dld_gpx3.ini file, see chapter 4). The number of counts in channel 5 corresponds directly to the

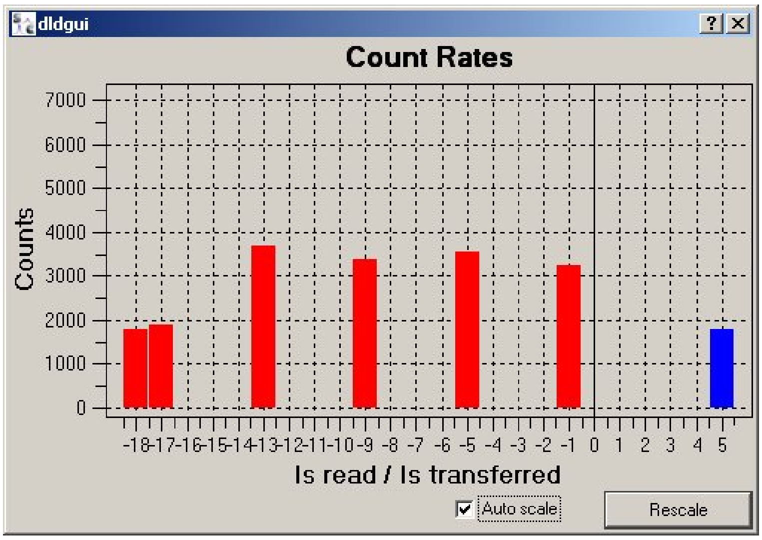
number of counts within the displayed image. Count rates are given always in counts per sec.

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 are always pre-calculated within the FPGA. But in RawData mode the unprocessed data are transferred to the PC and not the pre-calculated ones.

In RawData mode, the count rates of the channels -1, -5, -9 and -13 and 1, 2, 3 and 4 respectively should be equal. And the count rate in channel -18 should be comparable to the count rate of channel 5.

If the count rate exceeds the displayed range, press the "Rescale" button or tick the box named "AutoScale".



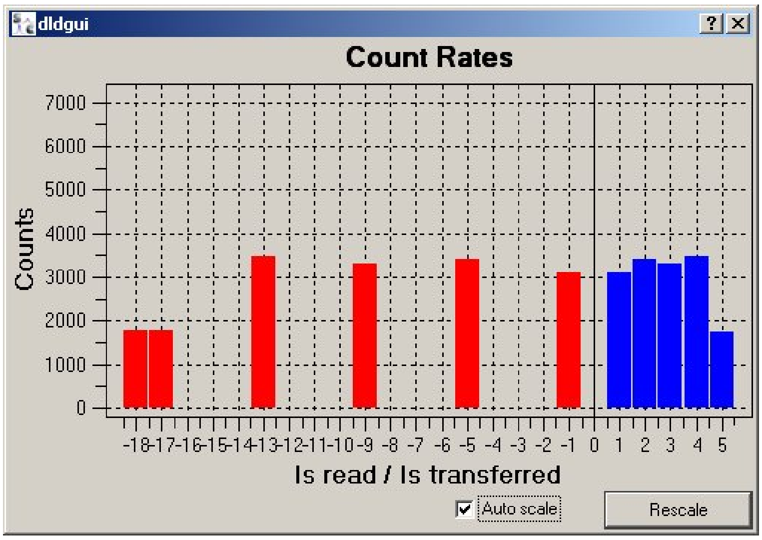


Figure 2: Rate meter windows for a measurement in Pair mode and RawData mode.

The rate meter is a powerful tool mainly used for diagnostics, as e.g. input control of the four detection channels. The ratio between the count rates of the detection channels and of the 4-fold coincidence (channel 5) is a measure for the detection efficiency of the detector. For a well adjusted detector system the number of counts within the single detection channels should be of almost equal height and the count rate within the image (4-fold coincidence) should be of about ¾ or more of the number of counts of the single detection channels.

Note

The count rate ratio between single channels and the 4-fold coincidence will decrease when going to high count rates (up to 10^6 and more) due to an increase in multi hit events and connected to this a loss of 4-fold coincidences.

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

3.5 Time resolved measurements

3.5.1 1D Time Histograms

General
Time Histograms can only be measured when an external start signal is given to the start input of the USB2.0-TDC and the appropriate switch (Ext_Gpx_Start) in the dld_gpx3.ini file has been set to "1" (for further details see chapter 4).

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. Click the button "TimeHist" to open the 1D time histogram window as shown in Figure 3.

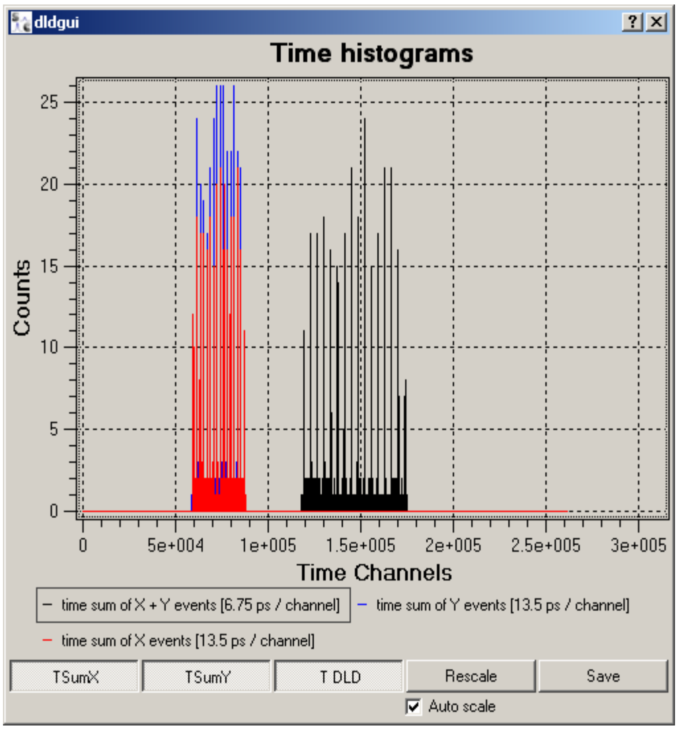
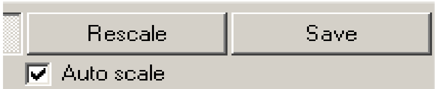


Figure 3: Time histogram window

The time sums of the X events, of the Y events and of the 4-fold coincidence (the total time sum of X+Y events) are displayed in red, blue and black colours respectively as integrated signal over the complete RoI. Therefore they do not show any lateral resolved information initially. Of course the time structures of individual

sample positions can be investigated by defining an appropriate RoI.
The three histograms can be turned off and on individually by pressing the buttons “TsumX”, “TsumY” and “T DLD”. Time histograms are also displayed and updated during the AutoRun mode.

Zooming in and out



The complete time range as defined in the dld_gpx3.ini file (either by the field “1D_histo_size” for RawData Mode or by the field “Data_Format” in Pair Mode) is displayed as default, when opening the 1D time histogram window. Point with the mouse on the time histogram window, press the left mouse button, keep it pressed and move the mouse to reduce the displayed number of time channels (“zooming in”). A rectangular field is drawn which marks the area displayed when letting go the left mouse button. For “zooming out” press the “Rescale” button (sometimes several times).

Saving of time histograms

To save a time histogram, press the button “Save”. A standard Microsoft™ window opens up, where a filename can be entered. All three histograms are saved in an ASCII data file.

Note **The GUI software does not offer the option of reloading time histograms.**

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

3.5.2 Setups for 1D Time Histograms – Important Notes

Depending on the settings in the dld_gpx3.ini file as well as the circumstances of your experiment, it can happen that a measured time distribution is displayed wrong or not at all in the 1D time histogram. This usually happens, when measured events are outside of the time window defined by the dld_gpx3.ini file. The following description gives a method on how to display a measured time distribution in the 1D time histogram correctly.

1.) Measurements done in RawData mode (binary switch in the dld_gpx3.ini file, see chapter 4)

The number of time channels displayed in the 1D time histogram is defined in RawData mode via the variable named “1D_histo_size” in the dld_gpx3.ini file (the number of measured time channels is only restricted by the TDC chip itself). This variable has a max. size of 262000 (≡ 1768.5 ns). It can happen not unlikely, that measurement data are arriving with a larger time offset in respect to the start trigger. This is the main reason for a 1D time histogram with no entries for RawData mode. The easiest way to deal with this is to use a binning in time (see chapter 3.5.3 for details) to evaluate the time offset. First try to do so without increasing the 1D_histo_size. Do only increase the 1D_histo_size, when no time distribution is shown in the 1D time histogram even with the largest binning in time being activated. See chapter 4 for details on changes in the dld_gpx3.ini file.

The offset can be subtracted from all results after evaluation by entering its value in the field “time offset” (see chapter 3.7 for details). The binning in time can now be reduced again and the relevant time distribution should still be displayed in the 1D time histogram.

Note

Do not forget to take in account the time offset when measuring absolute time distances for structures within a time distribution displayed in the 1D time histogram.

2.) Measurements done in Pair mode (binary switch in the dld_gpx3.ini file, see chapter 4)

The number of displayed time channels in a time histogram is defined in Pair mode also via the variable named "1D_histo_size" in the dld_gpx3.ini file. But the number of measured time channels is defined via the "data format" variable in the dld_gpx3.ini file (see chapter 4 for details). The maximum number of measured time channels in Pair mode is 14 bit (≡ 16384 channels ≡ 110.592 ns) in data format 1. A large bit number for time channels goes hand in hand with a small bit number for the pixel numbers 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).

Note

A large number of time channels goes hand in hand with a reduced number of pixels in x and y.

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

Example for data format 1:
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 being again sorted to the events in channel 1. Therefore the time distribution, which is displayed in the 1D time histogram, shows an overlapping of results, actually belonging to different times.
There are two ways of dealing with this:

- The problem with overlapping structures does not occur for a rather sharp time distribution, which is smaller than the defined time window. Such time distributions are displayed in the 1D time histogram in principal correctly from the beginning. The only question is the actual distance to the zero point (absolute time scale). Here the use of subtracting a time offset (see chapter 3.7 for details) can help to evaluate the absolute position in time. Increase the offset step by step until the time distribution starts to vanish to the left side of the 1D time histogram but does not reappear on the right side of the histogram. Now the time distribution has reached the zero position of the time axis and the absolute position in time is given directly by the time offset + the displayed time channel.

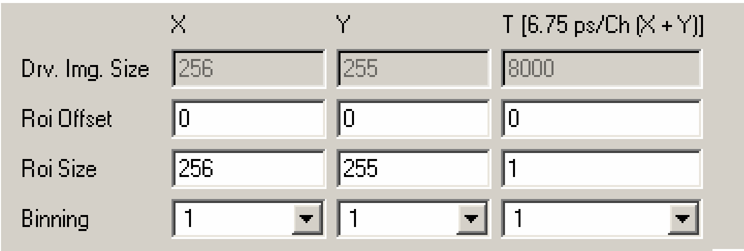
Example: Assume a time distribution given by a single peak at a distance of 20380 channels with a FWHM of 20 channels. This peak is displayed in channel 3996 of the 1D time histogram (assuming data format 1). Start to subtract 3996 channels. The peak moves to the left side of the histogram and will vanish partly. The one half of the peak, which has vanished, appears now on the right side of the histogram again, because it is located in the channels 16374 – 16384. Increasing the value for the time offset will move the peak further to the left. At one stage the complete peak appears on the right side of the histogram and moves again further to the left with an increased time offset. For a time offset of 20380 the peak has vanished again on the left side of the histogram. But now it does not reappear on the right side. This states, that the peak has been moved across the zero position of the time axis and that the value of the time offset corresponds to the absolute time position.

- The second way of dealing with the incorrect display of time distributions in the 1D histogram is to restrict the measured range if events in time by defining a RoI in time. Only results within a defined RoI are displayed. A restricted time window can be created with the use of defining a RoI. This can than be moved stepwise along the time axis for a sub sequential scanning of a measured time distribution.

3.5.3 3D Image stacks in time

3D (x,y,t) image stacks combine time and lateral resolved measurements. First of all, to record a 3D image stack a RoI in time must be defined. A RoI in time is defined similar to a RoI in X or Y by giving a position for the RoI to start from and the size of the RoI. For the RoI in time these values are entered in units of time channels in the fields “RoI Offset” and “RoI Size” of the time column “T”. The time bin size (size of one time channel) in ps is given in the header behind the naming of the T column in the main GUI interface. Select the favored Binning from the field “Binning” in the “T (6.75ps/Ch(X+Y))” column. As default the RoI in time is set to the first time channel (“RoI Offset” = 0 and “RoI Size” = 1).

For example: An image stack should be measured starting from channel 1700 up to channel 1763. “RoI Offset” is set to 1700 and “RoI Size” to 64. With an additional binning of 4, the image stack will contain 16 different time slices.



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.

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

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 No. of channels/ selected binning no. + 2.

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

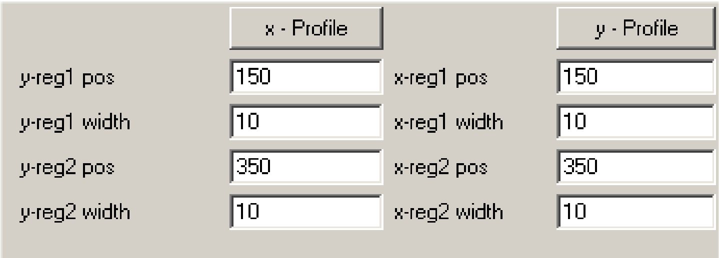
3.6 Line Profiles

3.6.1 Defining of Line Profiles

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 correspond to the RoI size.

For example: A line profile in X with values for “y-reg1 pos” of 120 and for “y-reg1 width” of 10 covers an area from pixel number 115 to pixel number 125 in Y - direction.

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



3.6.2 Display of Line Profiles

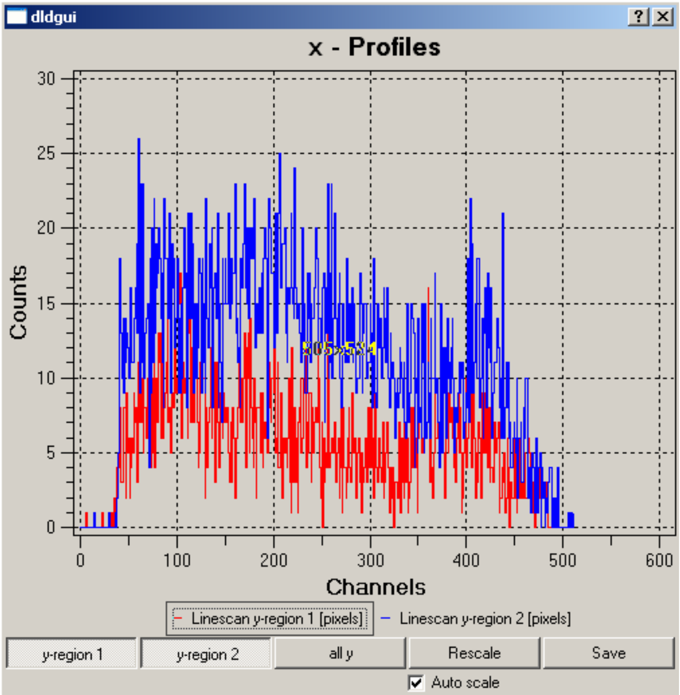


Figure 4: Profile window for x- Profiles

To display a line profile open the corresponding profile window by pressing the buttons "x - Profile" and "y - Profile" respectively. 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.

<u>Saving a line profile</u>
Click the "Save" button to save the activated line profiles as an ASCII data file. A standard Microsoft™ window opens up, where the wished filename can be entered.

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

3.7 Measurement and Device Adjustment

The adjustment of a measurement and or the TDC is in many things done within the dld_gpx3.ini file (see also chapter 4). None the less it is more convenient for a final adjustment, if different parameters can be changed directly via the GUI Main Interface instead of working in 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 4) 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 5 shows the GUI Main Interface with the additional entry fields activated.

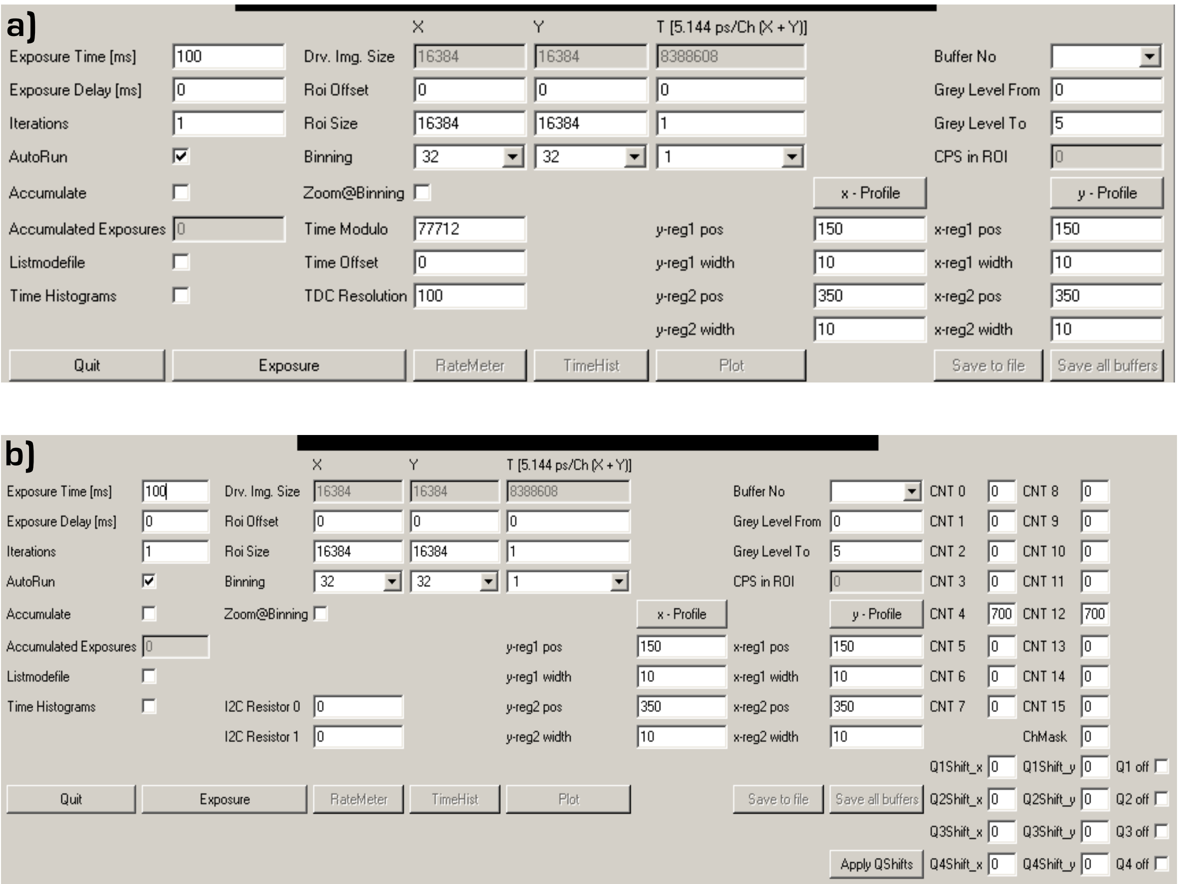


Figure 5: GUI Main Interface with all additional entry fields displayed when setting (a) modulo_adjust and (b) channel_adjust to "1".

Note

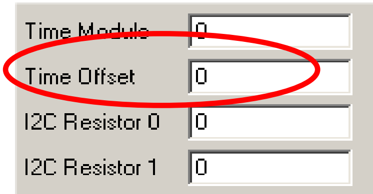
The entry field "TDC Resolution" will not be displayed, when "modulo_adjust" as well as "channel_adjust" is activated.

All additional fields are basically used for measurement/ device adjustment at the beginning. Later on these values can be entered directly in the dld_gpx3.ini file and the display of these entry fields can be turned off again. 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 taken over by the dld_gpx3.ini file. Therefore the value must be entered anew with each start of the GUI software if it has not been entered and saved in the dld_gpx3.ini file.**

3.7.1 Time Offset

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



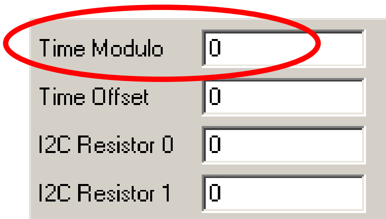
The value for "Time Offset" must be given in channels of the total time sum (T DLD)/4. It can also be defined via the variable "time_offset" in the dld_gpx3.ini file (see chapter 4).

3.7.2 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 must be reduced by using an appropriate frequency divider (for further details see your delayline detector manual).**

Figure 6 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 for 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 4) can be set directly.



The value for 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 delivers the modulo value.

<u>For example:</u> A distance of 54713 was measured between the first and the 16th peak for the multiple time histograms (black ones) given in Figure 6. 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 start off 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 the more accurate one and it should be preferred. None the less, a fine adjusting of the modulo value is necessary in most cases and therefore highly recommended. Modify the value slightly and check the width of the time histogram. The best modulo value should deliver the smallest width.

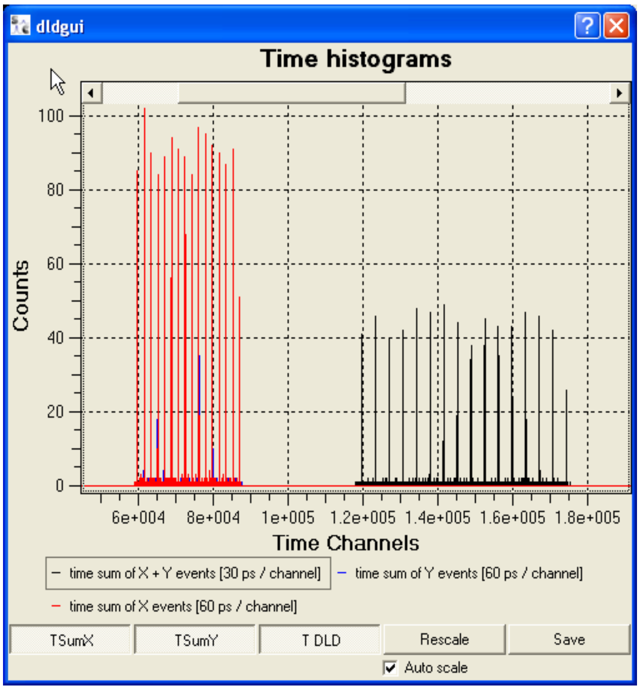


Figure 6: Time histograms measured with a 16-fold divided start signal

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

3.7.3 TDC Resolution

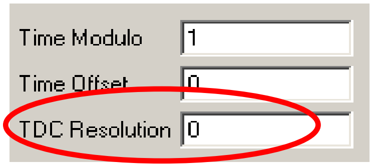
When starting the High Resolution-, the Dual Channel- or the Quad Channel- USB2.0-TDC (devices operating in the R mode), a so called PLO loop (an internal TDC measurement routine to reach the 27 ps digital time bin resolution) is started. It can happen that the PLO loop is not locked after starting the TDC. This depends mainly on minor fabrication variations of the TDC chips itself. The TDC gives out a warning massage that the resolution is not reached, if the PLO loop does not lock. The warning massage appears after pressing "exposure" for the first time after the TDC has been 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, than the digital time bin size should be slightly increased. Do so by adjusting the value of the variable "time_precision" in the dld_gpx3.ini file (see chapter 4). 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:

$$\text{Bin} \propto 1 / \text{hsdiv}$$

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.



3.7.4 I2C Resistor 0/ 1

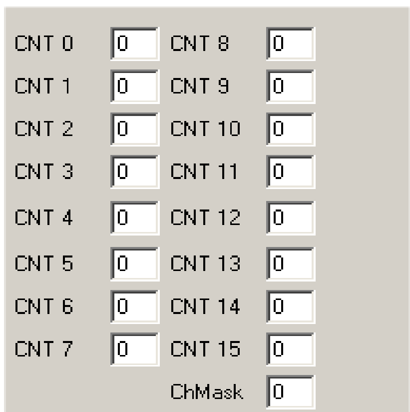
Some single devices (TDC and/ or analogue readout electronics) are equipped with electrical potentiometers (e.g. for high voltage power supply control). The communication to these potentiometers is realized via I2C. The resistor values are set via the I2C Resistor entry fields. If not especially stated 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 must be turned on in the dld_gpx3.ini file explicitly for a proper functioning (see chapter 4).

3.7.5 Channel Delay Times and Channel Deactivation

The GUI software offers the possibility to enter an additional delay time individually for each stop channel. This delay time is added to the time, which has been measured by the TDC (e.g. to compensate different cable delays). Delay times for 16 channels (0 - 15) can be entered in total. Entering can be done either directly in the dld_gpx3.ini file (see chapter 4) or in the corresponding entry fields in the GUI Main Interface after being activated in the dldgui.ini file.



The entry fields of channel delays 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 4). Channels will be deactivated by entering “1” in a bit mask. A “0” entry activates the corresponding channel.

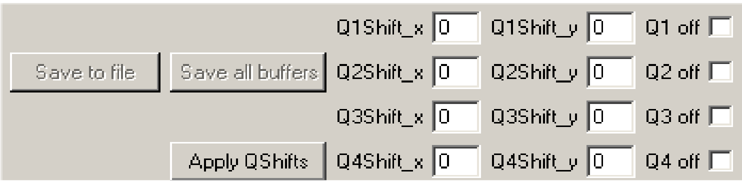
Note

Not the bit mask itself, but its corresponding number in decimal must 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

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

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



3.8 Data Saving Formats

3.8.1 Standard saving formats

The standard data format for saving acquired images is 16-bit tiff files and for 1D time histograms or Line Profiles simple ascii files.

3.8.2 List Mode Files (count file)

The GUI software also allows saving acquired data in a binary sorted in columns by X, Y, TsumX and TsumY, the so called list mode or count file. The list mode file option must first be activated in the dld_gpx3.ini file (see chapter 4) by setting the switch “Listmodefile” to “1”. This will create an additional box named “Listmodefile” in the GUI Main Interface. Tick the box for additional saving of the acquired data in a file named “counts_xxx.dat” in the folder of the GUI software. The triple x stands for the time of the operational system, which is added automatically by the GUI software converted in form of a string (e.g. counts_1233306575.dat). 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.

3.8.3 Bulk Files

A third method of saving data is the Bulk File. The Bulk File contains the pure raw data stream as measured by the TDC unaffected by settings of ROI, Binning etc, but affected by the channel delay values. The saving of the data stream can only be activated by setting the **both** switches "SaveBulk" and "Debug" in the dld_gpx3.ini file (see chapter 4) to "1".

Note **The GUI software will start in the so called debug mode, when "debug" is set to "1". In debug mode an additional information window appears with 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 are added consecutively to this file independent on 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 of the measurement and it should be renamed directly after ending of the measurement.
Each bulk file can be analyzed in respect of extracting 1D time histograms, measurement of a certain RoI etc. after measurement has been done, by working in simulation mode (see chapter 3.10) and using the bulk file as a simulation file. The relevant bulk file must only be renamed to "sim_input.txt" and be stored in the GUI software folder.

Note **Each bulk.txt file can be used for simulation mode by renaming it into sim_input.txt. By this it can be reworked with the GUI software after measurement has been done.
A backup of any existing sim_input.txt file as well as of any used bulk.txt file should be created before.**

3.9 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.

3.9.1 Flat Field Correction

Each detector comes with a certain inhomogeneous 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 4). Additionally also an adequate calibration file named "cal_i.tiff" and stored in the GUI software folder is needed. Otherwise the intensity calibration won't be performed. The intensity calibration is always performed 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).

3.9.2 Calibration of spatial distortions (x, y calibration)

Each detector comes with a certain spatial distortion (the misfit in imaging a straight line). Also this behavior can be corrected directly 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 4). Additionally also two adequate calibration files named "cal_x.tiff" and "cal_y.tiff" must be available in the GUI software folder. If one of the two files is missing, the x, y calibration won't be performed.
The x, y calibration is performed on each single detected event in contrast to the intensity calibration.

3.9.3 Calibration of distortions in time (t calibration)

Similar to the spatial distortions, each detector comes with 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 4). An adequate calibration file named "cal_t.tiff" must be available in the GUI software folder in addition. The time calibration is working independently 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. Also the t calibration is 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.

3.10 Simulation Mode

The simulation mode offers the possibility to work with the GUI software and to get to know its basic functions without a running hardware. Under certain circumstances, it also allows analyzing measured data after the measurement has been done (see chapter 3.8.3, bulk file).
The simulation mode is activated by setting the value of the parameter "Simulation" in the dld_gpx3.ini to "1" (see chapter 4). 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.
Start the GUI software.

! Note

The simulation mode does not respond to the exposure time, because this is hardware driven.

4 The INI Files

4.1 TDC and GUI configuration using "INI" files

The configuration of the TDC and of the GUI is made in the files dld_gpx3.ini and dldgui.ini. These files are located in the same folder as the GUI executable. Both ini files will be loaded once each time when the GUI software is started. Therefore changes to the ini files while the GUI is running have no effects until the GUI software is closed and started again. Make sure to save an ini file after entries have been changed, otherwise the changes will not be accepted.
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 complexity of the dld_gpx3.ini is fairly large and it takes some time and effort to get a complete overview. Therefore it is recommended that a backup file is created 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 deal with a couple of functions concerning image display. The complexity of the dldgui.ini file is rather small In contrast to 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 append any sign 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.**

4.2 Description of the dld_gpx3.ini file

Parameter and Entry example	*Parameter Description*
control	
iPairNumber = 2	Measurement results undergo a pre-arithmetic within the FPGA of the USB2.0-TDC, (2) reduces data load on the USB by pre-calculation of 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) or (6), than calculations of time differences are done within the GUI software (the so called raw data mode).

time_aperture = 0x3FFF	Quadrupel finder condition for raw data mode: bit 13 (the left 2) switches on/off, the number behind is channel size aperture.
aperture_offset = -98	Offset for quadrupel finder condition for raw data mode
ext_trigger = 0	Activates BNC synch trigger input for hardware triggered exposures when set to (1). Trigger signal must be TTL. **Do not exchange with ext. start (see TDC section)!**
counter_read = 0	for diagnostics of multi-anode DLDs with segments only.
NumChannels = 0	for use of multi-anode DLDs with segments only.
ChannelBinning = 16	for use of multi-channel DLDs only.
flim_mode = 0	Activates readout mode for optical FLIM detectors when set to (1).
debug = 0	When activated (1), an additional window opens when the GUI is started. This window contains information for diagnostic procedures.
simulation = 0	If turned ON (1), a set of test data is read from the hard disk. No hardware readout is performed. Caution, does not work with connected hardware and in raw data mode only.
Listmode = 0	listmode (counts data channel) on/off.
Listmodefile = 0	listmode file writing (only valid for DLD_DATA_COUNTS channel, see interface description).
LMF_data_number = 1000000	listmode transfer block size (only valid for DLD_DATA_COUNTS channel, see interface description).
SaveBulk = 0	only for software debugging: saves TDC stream into "bulk.txt".
MulitHitMode = 1	Turns the multihit algorithm on when set to (1). It works currently only in raw data mode due to performance limits. **Caution, time aperture matters!**
MulitHitDepth = 2	Sets the depth of the multihit recognition algorithm. **Caution, a high depth significantly effects the processing time!**
TDC_F1 = 0	TDC F1 switch. Set to (1) when working with a F1.
F1_highres = 0	switches on the highres mode for the F1 TDC.
histograms	
image_size_x = 0x4000	Adapts readout to real detector size in x. Depends on detector readout, caution, should fit to data_format.
image_size_y = 0x4000	Adapts readout to real detector size in y. Depends on detector readout, caution, should fit to data_format.
image_size_t = 0x80000	Restricts time slicing due to memory limits. Depends on detector readout, caution, 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 = 0.022	
PixelSizeY = 0.024	
ChannelSizeT = 0.015	
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	offset removement in 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.
max_time_channels = 0x7FFFF	max. number of time channels available from measurement device (0x7FFFF for single TDC, 0x01FFFFFF for double TDC).
max_timehist_len = 0x7F000	max. recommended number of time channels to use in time histograms.
mirror_x = 0	mirrors image at vertical middle axis, currently in raw data mode only due to performance optimization.
mirror_y = 0	mirrors image at horizontal middle axis, currently in raw data mode only due to performance optimization.
SumY_versus_Y = 0	for time sum diagnostics, in raw data mode only.
SumX_versus_X = 0	for time sum diagnostics, in raw data mode only.
cal_xyt = 0	3D linearity correction (requires detector depending correction data set).
cal_intensity = 0	Intensity correction, flatfield (requires detector depending data set).
TDC	

;firmware = ".\usb3gpx_LimitedData_NoDCM5.rbt" firmware = ".\usb3gpx_R_new.rbt" ;firmware = ".\usb3gpx_1d.rbt" ;firmware = ".\usb3gpx_NoDCM_I.rbt" ;firmware = ".\usb3gpx4q_old.rbt" firmware = ".\usb3gpx.bit"	Setting of the firmware file. **Do not change!**

SimulationDataFile = “sim_input.txt”	Defines the name of the simulation file used in simulation mode. “Sim_input.txt” is used as default name, if no other name is defined,
BulkDataFile = "bulk.txt"	File name of the BulkDataFile. This file can directly be used as simulation data file.
TimeCalDataFile = "t_cal.txt"	
TimeCalRun = 0	
RDN_WRN_Adjust = 1	enables individually adjustments for TDC optimization.
RDN1Start = 2	Individually adjusted for TDC hardware. **Do not change!**
RDN1Stop = 4	
RDN2Start = 2	
RDN2Stop = 4	
WRNStart = 2	
WRNStop = 2	
One_Gpx = 0	Setting depends on the hardware layout of the TDC. Enter (1) when the TDC is equipped with one GPX-TDC chip and (0) when the TDC is equipped with two GPX-TDC chips.
I_Mode = 0	82 ps resolution mode (8 stop channels per TDC) or if off: 27 ps mode (2 stop channels per TDC).
G_Mode = 0	Activates the G_Mode.
G_x1_corr = 515	Correction values for G-Mode given in thousands in time interval between rising and falling edge. **Do not change!**
G_x2_corr = 485	
G_y1_corr = 510	
G_y2_corr = 490	
G_x_slope = 180	
G_y_slope = -30	
G_x_frequ = 600	
G_x_phase = 0	
G_x_ampli = 0	
G_y_frequ = 500	
G_y_phase = 0	
G_y_ampli = 0	
X_Correct = 100	percentage of image X coordinate re-scaling (in raw data mode only).
Y_Correct = 100	percentage of image Y coordinate re-scaling (in raw data mode only).
ChronoStack = 0	TDC results chronostack on or off (off may improve multihit capability).
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. **Data Format works only in pair mode!**
TTL_Inputs_R_mode = 0	use TTL inputs for High Resolution TDC, dual channel TDC and quad channel TDC (multi-channel DLDs).

Ext_Gpx_Start = 0	When set to (0) the TDC uses its internal clock for time measurements (imaging only). Set to (1) when using external start signals for time resolved measurements (3D mode).
ResynchronTime = 0	Soft reset time of GPX chips for re-synchronization in ms (checks for gathered exposure times, communication overhead times do not count).
Hexanode = 0	Roentdek Hexanode readout mode (raw data mode only).
DLD_1D = 0	for diagnostics of multi-channel detectors only.
TimeSlicing = 0	time-slicing detector mode (for raw data mode only, max. 16 time slices (x,y) in one image).
TimeSliceWindowSize0 = 0x1FFFF	time-slicing multi-channel detectors only.
TimeSliceWindowSize1 = 0x1FFFF	
TimeSliceZero0 = 0x0	
TimeSliceZero1 = 0x20000	
Ch0 = 0	stop channel corrections values in time bin channel units, for an individual DLD adjustment. **Do not change!**
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.
TimeDif1Min = 0xF000	parameters for FPGA quadrupel finder.
TimeDif1Min = 0x0FFF	
TimeDif2Min = 0xF000	
TimeDif2Max = 0x0FFF	
SumDifMin = 0xFF00	
SumDifMax = 0x00FF	
QDLD	
4Q_Dld = 0	4 quadrant detector mode on (only for 4 fold DLDs).
shift_x1 = 110	quadrant image positioning values.
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	
<u>I2C</u>	
I2C = 0	Turns I2C communication ON (1) or OFF (2).
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".

4.3 Description of the dldgui.ini file

Parameter and Entry example	***Parameter Description***
New_Intensity_Calibration 0	When set to 1, the next taken image is used for intensity calibration. Do only start exposure. Image saving is done automatically. Do not forget to set back to 0 when the calibration image was taken.
Zoom@Binning 0	Screen output expanded to 512x512 pixels if on (1) when binning provides less than 512 pixels, no effect to saved data.
Zoom_out_x_6 0	If turned ON (1) then a 6 x 6 binning on low level in data acquisition is established for diagnostics.
counter_read 0	(1) enables 9 additional counter (TTL) inputs for extensional use (**optional devices**).
refresh_time_ms 2500	Refresh time defines a time interval during exposure after that the actual image is given out periodically.
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).
Channel_adjust 0	When set to (1), additional entry fields are displayed in the GUI Main Interface for adjusting delay values for Ch 0 to Ch 15.
small_screen 0	Reduces the GUI Main Interface to fit to computer screens with a smaller resolution than 864 pixel in Y.
CalibCoord_x1 0 CalibCoord_x2 512 CalibCoord_y1 0 CalibCoord_y2 512	Internal calibration coordinates. **<u>Do not change!</u>**

InitialExposureTime 1000	Sets the default exposure time in ms, which is used when starting the GUI software.

5 Delayline Detector DLL

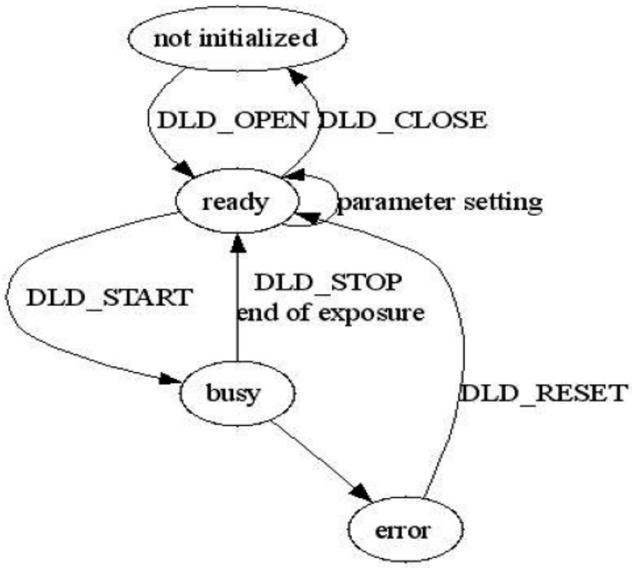
5.1 The basic working principle

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

- The user opens the driver and gets an ID for the use of the device.
- The parameters for the measurement mode and the measurement mode itself 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.

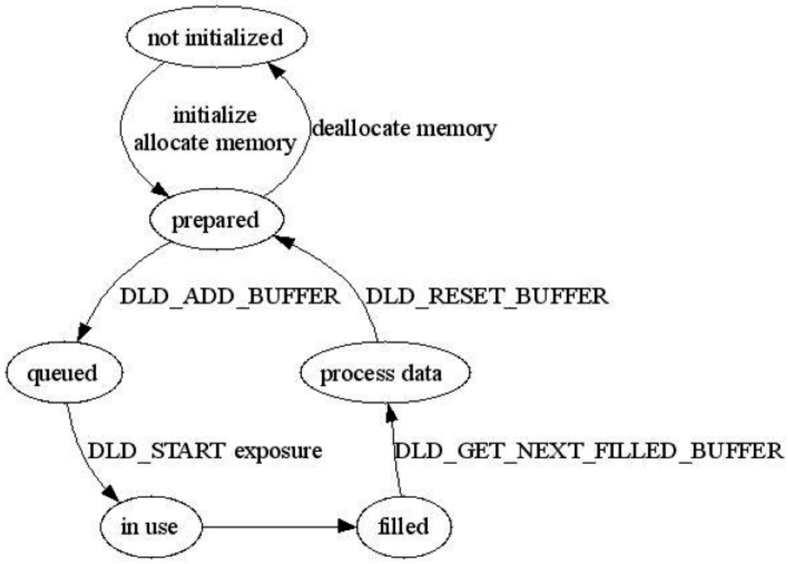
5.2 Interface Description

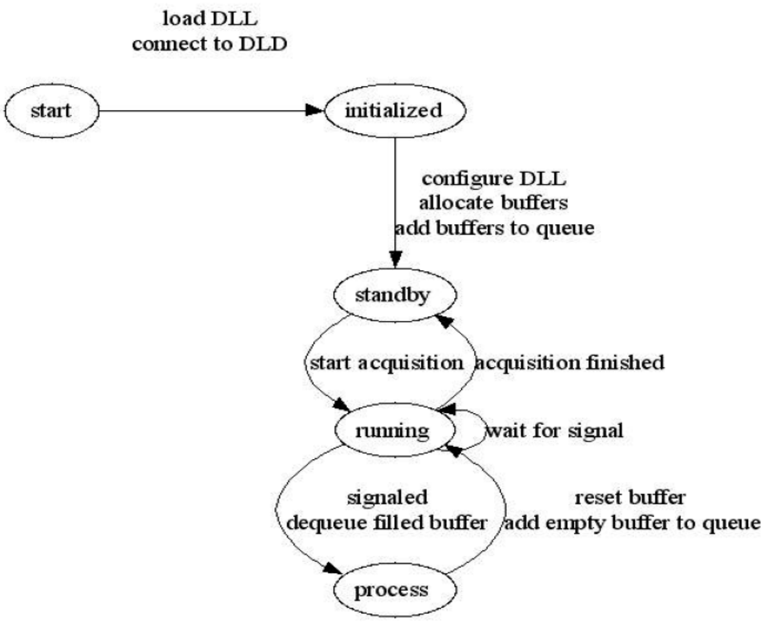
5.2.1 Life cycle of the DLL



5.2.2 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.





5.2.3 Image data

The image data channel of the DLD can be configurated 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.

5.3 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	# delivers the momentarily set ROIs
DLD_SET_IMG_ROI	# places ROIs
DLD_GET_IMG_BINNING_INFO_MAX	# number of possible binning values
DLD_GET_IMG_BINNING_INFO	# list of possible binning values
DLD_GET_IMG_BINNING	# delivers the active binning setting
DLD_SET_IMG_BINNING	# 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	# sets the transformation table for image data
DLD_GET_IMG_THRESHOLD	# gets the threshold for the delayline
DLD_SET_IMG_THRESHOLD	# sets the threshold for the delayline
DLD_GET_CHANNEL_THRESHOLD	# gets the threshold for the channel detector
DLD_SET_CHANNEL_THRESHOLD	# sets the threshold for the channel detector

Methods for data acquisition	
DLD_START	# starts data acquisition
DLD_STOP	# stops data acquisition
DLD_ABORT	# 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	# deletes the EmptyBufferQueue
DLD_RESET_BUFFER_QUEUE	# deletes all bufferqueues

5.4 DLL Conventions

5.4.1 Calling convention

As default, the subroutines deliver back 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 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 also the maximal available memory size is handed over for return values with variable length. In general the client should allocate and clear memory.

5.4.2 Compiler

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

5.5 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

6 List of Figure