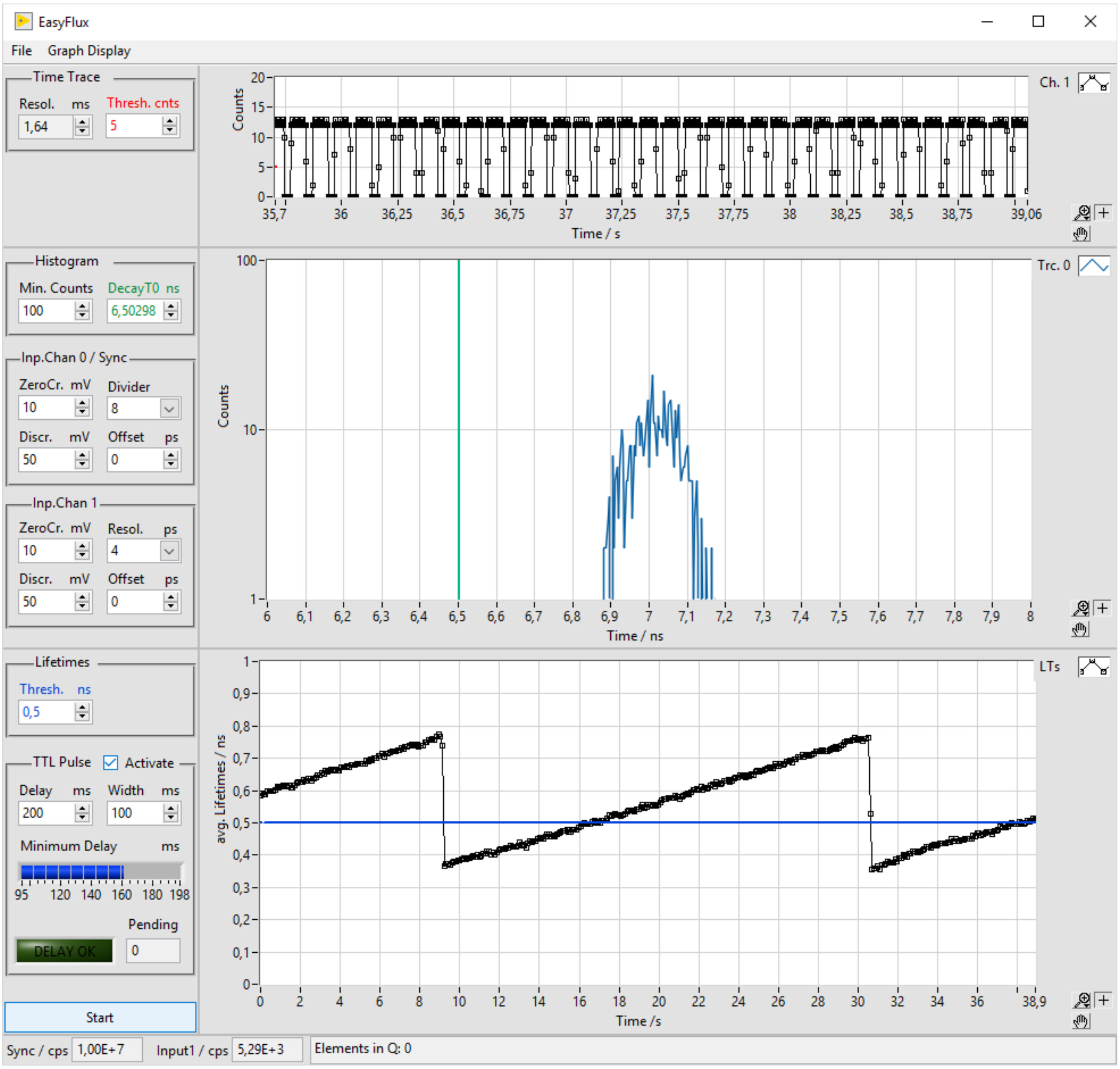
EasyFlux Manual

1. Introduction

EasyFlux.vi is a 64bit LabVIEW application for measuring photon streams from PicoHarp hardware in T3 mode. It extracts a time trace and corresponding histograms from that stream and calculates average lifetimes and timing information from the histograms and the stream, respectively. By implementing a TTL puls generator one can use these information for e.g. sorting fluorophores from a liquid stream based on their lifetimes in “quasi real-time”.

It is recommended to run the application on a Windows 10 operating system with a PC having at least four (logical) cores and sufficient memory.



Time Trace
Resol. ms: 1,64
Thresh. cnts: 5

Histogram
Min. Counts: 100
DecayT0 ns: 6,50298

Inp.Chan 0 / Sync
ZeroCr. mV: 10
Divider: 8
Discr. mV: 50
Offset ps: 0

Inp.Chan 1
ZeroCr. mV: 10
Resol. ps: 4
Discr. mV: 50
Offset ps: 0

Lifetimes
Thresh. ns: 0,5

TTL Pulse ✓ Activate
Delay ms: 200
Width ms: 100
Minimum Delay ms
Pending: 0
DELAY OK

Sync / cps: 1,00E+7
Input1 / cps: 5,29E+3
Elements in Q: 0

Fig. 1: EasyFlux.vi screenshot showing the main GUI elements. Data shown here were generated with a pulse generator for simulating a line scan. Settings: sync. rate: 10 MHz; average count rate: 6000 cps.

2. Usage

When starting the VI all front-panel controls and indicators are disabled and grayed out until the hardware is initialized. If an error occurs during initialization an error message is displayed and initialization is aborted. After successful initialization the info field of the status bar will display '*Initialized'* and all GUI elements will be activated. At that point the GUI controls are initialized with values read from an INI file (EasyFlux.ini in same folder as EasyFlux.vi). By selecting the menu entry *File → Open Config File…* the INI file can be viewed and changed. Changes will affect the initial control values only after a restart of the application.

Pressing the Start button (bottom of left panel) starts an experiment. If no sync rate could be detected the start is aborted and the info field of the status bar displays *'Sync rate not sufficient!'*. The current sync rate is also displayed in the status bar (cf. Fig. 2).

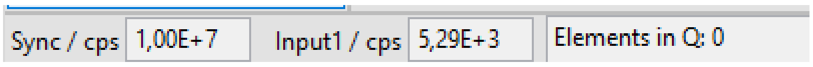

Sync / cps 1,00E+7
Input1 / cps 5,29E+3
Elements in Q: 0

Fig. 2: Sync rate display, count rate display and info field of status bar.

During a running experiment the info field in the status bar shows the number of elements in the queue used for transferring data from the data acquisition thread to the data processing thread inside the software. This value should always be near zero. If the value of *'Elements in Q'* goes up, the amount of acquired data is to high for 'real-time' processing and the software will eventually crash due to 'out-of-memory'.

After starting an experiment the uppermost graph on the right panel displays the time trace extracted from the photon stream. 2048 points are displayed in a chart like manner with a time span depending on the resolution. The resolution (in ms) can be changed in multiples of power(2) times the sync period (e.g. sync rate = 10MHz → sync period = 100 ns; e.g. 2^14 = 16.384 → x 100 ns = 1,64 ms; cf. Fig. 3). The value of which the power(2) is taken (i.e. 14 in the above example) is stored in the INI file as key 'BitShift'. The resolution should not be changed during the actual sorting (i.e. when TTL Pulse is active; cf. Fig. 7).

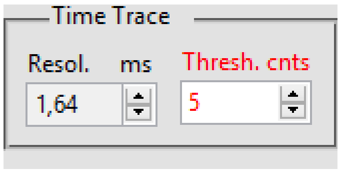
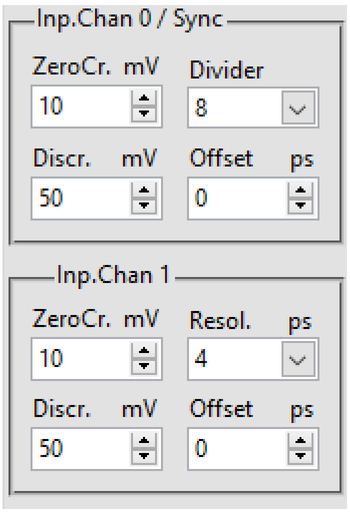

Time Trace
Resol. ms 1,64
Thresh. cnts 5

Fig. 3: Controls for setting time trace resolution and threshold used for histogramming.

Inside the time trace chart a threshold can be set by either dragging the red cursor or setting the threshold control on the left panel (cf. Fig. 3). This threshold affects the histogram which is displayed in the middle graph on the right panel: Every time a rising edge of the time trace crosses the threshold a new histogram is started and every time a falling edge crosses the threshold the histogramming is stopped and the data are transferred inside the program for further processing.
Further settings affecting the histogramming can be made by the controls shown in Fig. 4. Please refer to the PicoHarp manual for their meaning.



Inp.Chan 0 / Sync

ZeroCr. mV	Divider
10	8

Discr. mV	Offset ps
50	0

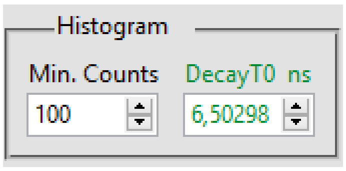
Inp.Chan 1

ZeroCr. mV	Resol. ps
10	4

Discr. mV	Offset ps
50	0

Fig. 4: Controls for PicoHarp hardware settings. During the actual sorting (i.e. when TTL Pulse is active; cf. Fig. 7) the sync divider must not be changed!

Inside the histogram graph a time zero (in ns) can be set by either dragging the green cursor or setting the appropriate control on the left panel (DecayT0; cf. Fig. 5). This time zero represents the reference time which is used for calculating the average lifetimes. It should be set right in the middle of the rising edge of a decay histogram (unlike in Fig. 1) and not be changed during the actual sorting (i.e. when TTL Pulse is active; cf. Fig. 7). If it has to be changed during a sorting experiment due to e.g. a thermal induced temporal drift of the sync pulse, it should be changed with care because it directly affects the absolute values of the average lifetimes. Consider also changing the sync offset (cf. Fig. 4) in such a case.



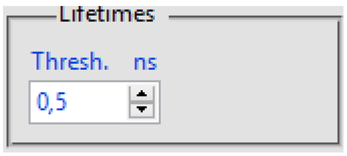
Histogram

Min. Counts	DecayT0 ns
100	6,50298

Fig. 5: Controls for histogram settings affecting the calculation of average lifetimes.

Calculation of average lifetimes is also affected by the Control ‘Min. Counts’ (cf. Fig. 5): Only if the integral counts of a histogram exceed this value, an average lifetime is calculated. This allows for rejection of arbitrary noise peaks crossing the threshold in the time trace.

The calculated average lifetimes are accumulated and displayed in the lowermost graph on the right panel. Inside this graph a threshold can be set by either dragging the blue cursor or setting the threshold control on the left panel (cf. Fig. 6). This threshold serves as a criteria for firing a TTL pulse. The software implementation of this criteria has to be changed by the user in accordance with his experiment. At the moment it is implemented such that if the average lifetime is above the threshold the TTL pulse will be triggered. The implementation of the actual TTL generating hardware has to be done by the user as well.



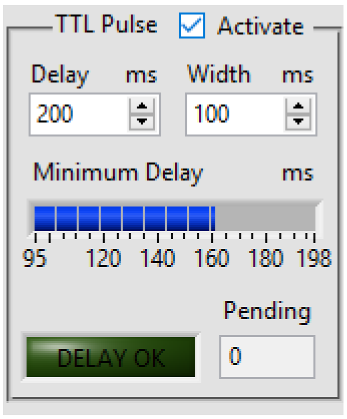
Lifetimes

Thresh. ns: 0,5

Fig. 6: Control for setting a lifetime threshold as a criteria for firing a TTL pulse.

Further settings for the TTL pulse generation are already prepared in this software but not all of them are implemented (cf. Fig. 7). The value in the control ‘Delay’ (in ms) will delay the TTL pulse

with respect to the time the rising edge of the corresponding signal has crossed the threshold in the time trace. Due to certain latencies in acquisition and processing of the data this value can not be made arbitrarily small. The slider indicator ‘Minimum Delay’ gives an estimate of the minimum value the Delay must have. The lower and upper boundary values of this slider represent the range in between all calculated minimum delays fall since the experiment has started. Change these boundaries (increase lower boundary and decrease upper boundary until the software automatically changes these values again) if all other settings have been made and the experiment runs stable. The value which is set for ‘Delay’ should be at least above the upper boundary to ensure that no sorting pulse is missed during the experiment. If for some reason a delay value is to short, this will be indicated by the LED indicator ‘DELAY OK’. The indicator ‘Pending’ shows the number of TTL events in the corresponding queue that have not yet been triggered.
The control ‘Width’ (in ms) will set the duration of the TTL pulse. This functionality has to be implemented by the user as well.


TTL Pulse ☑ Activate

Delay ms: 200
Width ms: 100
Minimum Delay ms

Pending: 0

Fig. 7: Controls for setting properties for the TTL pulse generation.

A user implemented TTL pulse is only triggered if the checkbox ‘Activate’ next to TTL Pulse on the left panel is checked (cf. Fig. 7). This allows for optimization of all relevant settings without generating trigger events. So the general workflow would be:

1. Press the Start button to start data acquisition (make sure a sync signal is present)
2. Optimize all parameters described above for the particular experiment
3. Stop the acquisition or alternatively clear the average lifetime graph by selecting the appropriate menu entry
4. Check TTL Pulse Activate
5. Start data acquisition if it was stopped in 3.

3. Disclaimer

PicoQuant GmbH disclaims all warranties with regard to this software including all implied warranties of merchantability and fitness. In no case shall PicoQuant GmbH be liable for any direct, indirect or consequential damages or any material or immaterial damages whatsoever resulting from loss of data, time or profits arising from use or performance of this software. This software including its source code is provided 'as is' without any warranty whatsoever. By installing the software you agree to these terms.