

# PicoHarp 300

Picosecond Histogram  
Accumulating Real-time Processor



**PICOQUANT**  
Unternehmen für optoelektronische  
Forschung und Entwicklung

Time–Correlated  
Single Photon Counting System  
with USB Interface



User's Manual and Technical Data

Software Version 2.3

# Table of Contents

<b>1. Introduction.....</b>	<b>4</b>
<b>2. Primer on Time-Correlated Single Photon Counting.....</b>	<b>5</b>
<b>3. Hardware and Software Installation.....</b>	<b>13</b>
3.1. General Installation Notes – Read This First.....	13
3.2. Software Installation.....	14
3.3. Device Installation.....	15
3.4. Installation Troubleshooting.....	16
3.5. Uninstalling the Software.....	16
<b>4. Software Overview.....</b>	<b>17</b>
4.1. Starting the Program.....	17
4.2. The Main Window.....	17
4.3. The Toolbar.....	18
4.4. The Control Panel.....	19
4.5. The Axis Panel.....	20
4.6. The Trace Mapping Dialog.....	21
<b>5. Specific Measurement Tasks.....</b>	<b>22</b>
5.1. Setting up the Input Channels.....	22
5.2. Setting up and Running Interactive Measurements.....	25
5.3. Routed Measurements.....	26
5.3.1. General Setup and Operation.....	26
5.3.2. Using the PHR 800 Router.....	28
5.4. Time-Tagged Mode Measurements.....	29
5.5. TTTR Measurements with Real-Time Correlation.....	34
5.6. Time-Resolved Emission Spectra.....	36
<b>6. Controls and Commands Reference.....</b>	<b>40</b>
6.1. Main Window.....	40
6.2. Toolbar.....	41
6.3. Control Panel.....	42
6.4. Axis Panel.....	45
6.5. Trace Mapping Dialog.....	46
6.6. Menus.....	46
6.6.1. File Menu.....	46
6.6.2. Edit Menu.....	49
6.6.3. View Menu.....	49
6.6.4. Help Menu.....	50
6.7. Title and Comment Editor.....	51
6.8. Print Preview Dialog.....	51
<b>7. Problems, Tips &amp; Tricks.....</b>	<b>52</b>
<b>8. Appendix.....</b>	<b>54</b>
8.1. Warnings.....	54
8.2. Data File Formats.....	56

---

8.2.1. Interactive Mode File Format.....	56
8.2.2. T2 Mode File Format.....	59
8.2.3. T3 Mode File Format.....	61
8.2.4. Data Types.....	63
8.3. Hardware Technical Data.....	64
8.3.1. Common Specifications.....	64
8.3.2. Router / Control Port Connectors.....	66
8.3.2.1. PicoHarp 300 Hardware Version 1.0.....	66
8.3.2.2. PicoHarp 300 Hardware Version 2.0.....	68

# 1. Introduction

While fluorescence spectroscopic investigations are fairly common today, extracting additional temporal information from molecules via laser induced fluorescence is a relatively new and much more powerful technique. The temporal analysis can reveal information about the molecule not available from spectral data alone. This is why lifetime analysis of laser induced fluorescence by means of Time-Correlated Single Photon Counting (TCSPC) has gained importance over the recent years. The difference in the fluorescence decay times of appropriate fluorescent dyes provides a powerful discrimination feature to distinguish molecules of interest from background or other species. This has made the technique very interesting for sensitive analysis, even down to the single molecule level. Of course other aspects than lifetime of the temporal pattern of the fluorescence decay are of interest too.

The acquisition of fluorescence decay curves by means of TCSPC provides resolution and sensitivity that cannot be achieved with other methods. In practice it is done by histogramming arrival times of individual photons over many excitation and fluorescence cycles. The arrival times recorded in the histogram are relative times between laser excitation and corresponding fluorescence photon arrival (start-stop-times) ideally resolved down to a few picoseconds. The resulting histogram represents the fluorescence decay. Although fluorescence lifetime analysis is a main field of application for the PicoHarp, it is in no way restricted to this task. Other important applications are e.g. Quantum Cryptography (QC) Time-Of Flight (TOF) and Optical Time-Domain Reflectometry (OTDR) as well as general coincidence correlation.

The PicoHarp 300 is a compact easy-to-use system with Universal Serial Bus (USB) interface. Its new solid state design allows innovative measurement approaches, improves reliability and simplifies calibration. The circuit allows high measurement rates up to 10 million counts per second (Mcps) and provides a time resolution of 4 ps. The modern USB 2.0 interface provides high throughput as well as 'plug and play' installation. The input triggers are programmable for a wide range of input signals. Both have a programmable Constant Fraction Discriminator (CFD). These specifications qualify the PicoHarp 300 for use with most common single photon detectors such as Single Photon Avalanche Photodiodes (SPADs) and Photomultiplier Tube (PMT) modules (via preamplifier). The best time resolution is obtained by using Micro Channel Plate PMTs (MCP-PMT) or modern SPAD detectors. The PicoHarp 300 is perfectly matched to these detectors and the overall Instrument Response Function (IRF) can be as short as 30 ps FWHM. Similarly, inexpensive and easy-to-use diode lasers such as the PDL 800-B with interchangeable laser heads, as well as the most expensive mode locked lasers can be used as an excitation source. This permits lifetime measurements down to under ten picoseconds with deconvolution, e.g. using the FluoFit Global Fluorescence Decay Fit Software.

The PicoHarp 300 can operate in various modes to adapt to different measurement needs. The standard histogram mode performs real-time histogramming in on-board memory. Other modes are available as upgrade options. For instance, two different Time-Tagged-Time-Resolved (TTTR) modes allow to record each photon event separately, with the individual arrival times to provide unlimited flexibility in off-line data analysis such as burst detection and time-gated or lifetime weighted Fluorescence Correlation Spectroscopy (FCS) as well as picosecond coincidence correlation. The PicoHarp 300 is furthermore supported by a variety of accessories such as pre-amplifiers and an optional hardware and software add-on allowing control of a monochromator from within the PicoHarp software. The latter supports automated measurement of Time-Resolved Emission Spectra (TRES)

The PicoHarp hardware is offered in two versions. The standard product PicoHarp 300 provides more features. The PicoHarp 300E is a leaner variant targeting cost sensitive applications where some functionality is not required. The features that distinguish the PicoHarp 300 from the 300E model are support for routing devices for simultaneous measurements with several detectors and interfacing with external hardware via TTL marker signals.

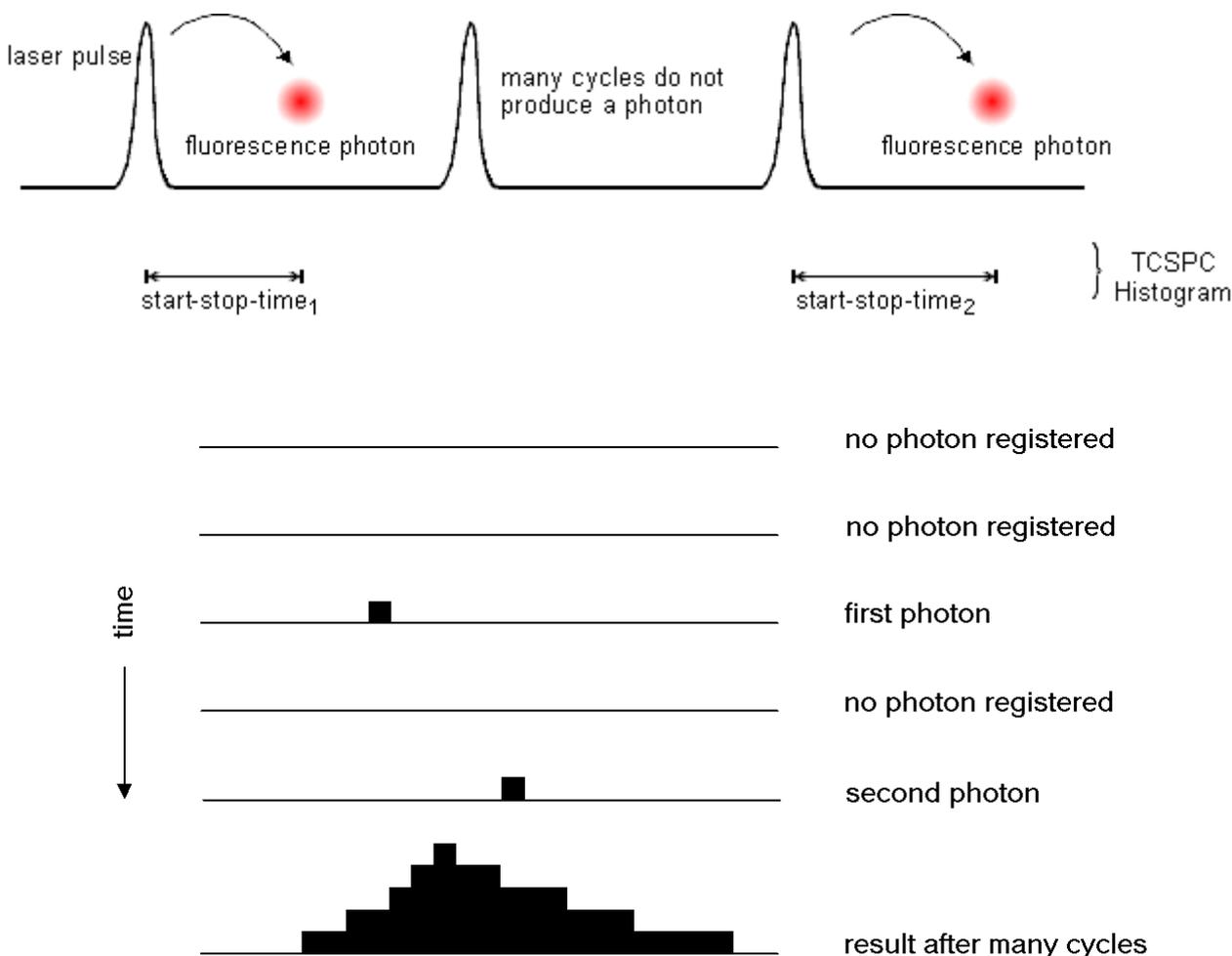
The PicoHarp software runs on all recent 32-bit Windows PC platforms from Windows 2000 to XP and Vista, as well as on the x64 editions. It provides functions such as the setting of measurement parameters, display of measurement results, loading and saving of measurement parameters and decay curves. Important measurement characteristics such as count rate, count maximum and position, histogram width (FWHM) are displayed continuously. Data can conveniently be exported via the clipboard, e.g. for immediate processing by the FluoFit Fluorescence Decay Fit Software. An optional programming library (DLL) enables users to write custom data acquisition programs for the PicoHarp in all modern programming environments.

For details on the Time-Correlated Single Photon Counting method, please read section 2 as well as our TechNote on TCSPC and consult the literature referenced in Further Reading at the end of the next section. Experienced users of the method should be able to work with the PicoHarp straight away. Nevertheless, we recommend carefully reading the section 3 on Hardware and Software Installation to avoid damage. Later, the comprehensive help function of the PicoHarp software will probably let the manual gather dust on the shelf.

## 2. Primer on Time-Correlated Single Photon Counting

In order to make use of a powerful analysis tool such as time-resolved fluorescence spectroscopy, one must record the time dependent intensity profile of the emitted light. While in principle, one could attempt to record the time decay profile of the signal from a single excitation-emission cycle, there are practical problems to prevent such a simple solution in most cases. First of all, the decay to be recorded is very fast. Typical fluorescence from organic fluorophores lasts only a few hundred picoseconds to some hundred nanoseconds. In order to recover fluorescence lifetimes as short as e.g. 200 ps, one must be able to resolve the recorded signal at least to such an extent, that the exponential decay is represented by some tens of sample points in time. This means the transient recorder required would have to sample at e.g. 20 ps time steps. This is hard to achieve with ordinary electronic transient recorders of reasonable dynamic range. Secondly the light available may be simply too weak to sample an analog time decay. Indeed the signal may consist of just single photons per excitation/emission. This is typically the case for single molecule experiments or work with minute sample volumes/concentrations. Then the discrete nature of the signal itself prohibits analog sampling. Even if one has more than just one single molecule and some reserve to increase the excitation power to obtain more fluorescence light, there will be limits, e.g. due to collection optic losses, spectral limits of detector sensitivity or photo-bleaching at higher excitation power. The solution is Time-Correlated Single Photon Counting (TCSPC). Since with periodic excitation (e.g. from a laser) it is possible to extend the data collection over multiple cycles, one can reconstruct the single cycle decay profile from single photon events collected over many cycles.

The method is based on the repetitive, precisely timed registration of single photons of e.g. a fluorescence signal. The reference for the timing is the corresponding excitation pulse. As a single photon sensitive detector a Photo Multiplier Tube (PMT) or a Single Photon Avalanche Diode (SPAD) can be used. Provided that the probability of registering more than one photon per cycle is low, the histogram of photon arrivals per time bin represents the time decay one would have obtained from a single shot time-resolved analog recording. The precondition of single photon probability can (and must) be met by attenuating the light level at the sample if necessary. If the single photon probability condition is met, there will actually be no photons at all in many cycles. The diagrams below illustrate how the histogram is formed over multiple cycles.



The histogram is collected in a block of memory, where one memory cell holds the photon counts for one corresponding time bin. These time bins are often referred to as time channels. In practice, the registration of one photon involves the following steps: first, the time difference between the photon event and the corresponding excitation pulse must be measured. For this purpose both signals are converted to electric signals. For the fluorescence photon this is done via the single photon detector mentioned before. For the excitation pulse it may be done via another detector if there is no electrical sync signal supplied by the laser. Obviously, all conversion to electrical pulses must preserve the precise timing of the signals as accurately as possible. The actual time difference measurement is done by means of fast electronics which provide a digital timing result. This digital timing result is then used to address the histogram memory so that each possible timing value corresponds to one memory cell or histogram channel. Finally, the addressed histogram cell is incremented. All steps are carried out by fast electronics so that the processing time required for each photon event is as short as possible. When sufficient counts have been collected, the histogram memory can be read out. The histogram data can then be used for display and e.g. fluorescence lifetime calculation. In the following we will expand on the various steps involved in the method and associated issues of importance.

### Count Rates and Single Photon Statistics

It was already mentioned that it is necessary to maintain a low probability of registering more than one photon per cycle. This was to guarantee that the histogram of photon arrivals represents the time decay one would have obtained from a single shot time-resolved analog recording (The latter contains the information we are looking for). The reason for this is briefly the following: Due to dead times of detector and electronics for at least some tens of nanoseconds after a photon event, TCSPC systems are usually designed to register only one photon per excitation/emission cycle. If now the number of photons occurring in one excitation cycle were typically  $> 1$ , the system would very often register the first photon but miss the following one or more. This would lead to an over-representation of early photons in the histogram, an effect called 'pile-up'. This leads to distortions of the fluorescence decay, typically the fluorescence lifetime appearing shorter. It is therefore crucial to keep the probability of cycles with more than one photon low.

To quantify this demand, one has to set acceptable error limits and apply some mathematical statistics. For practical purposes one may use the following rule of thumb: In order to maintain single photon statistics, on average only one in 20 to 100 excitation pulses should generate a count at the detector. In other words: the average count rate at the detector should be at most 1 to 5 % of the excitation rate. E.g. with the diode laser PDL 800-B, pulsed at 80 MHz repetition rate, the average detector count rate should not exceed 4 Mcps. This leads to another issue: the count rate the system (of both detector and electronics) can handle. Indeed 4 Mcps may already be stretching the limits of some detectors and usually are beyond the capabilities of conventional TCSPC systems. Nevertheless, one wants high count rates, in order to acquire fluorescence decay histograms quickly. This may be of particular importance where dynamic lifetime changes or fast molecule transitions are to be studied or where large numbers of lifetime samples must be collected (e.g. in 2D scanning configurations). This is why high laser rates (such as 40 or 80 MHz from the PDL 800-B) are important. PMTs can safely handle TCSPC count rates of up to 10 Mcps, standard (passively quenched) SPADs saturate at a few hundred kcps, actively quenched SPADs may operate up to 5 Mcps but some types suffer resolution degradation when operated too fast. Old NIM based TCSPC electronics can handle a maximum of 50 to 500 kcps, newer integrated TCSPC boards may reach peak rates of 3 to 5 Mcps. With the PicoHarp 300, average count rates of 5 Mcps and peak rates up to 10 Mcps can be collected. It is worth noting that the count arrival times are typically random so that there can be bursts of high count rate and periods of low count rates. Bursts of photons may still exceed the average rate. This should be kept in mind when comparing count rates considered here and elsewhere. The specifications for TCSPC systems may interpret their maximum count rates differently in this respect. This is why another parameter, the so called dead-time is also of interest. This quantity describes the time the system cannot register photons while it is processing a previous photon event. The term is applicable both to detectors and electronics. Dead-time or insufficient throughput of the electronics do not usually have a detrimental effect on the decay histogram or, more precisely, the lifetime to be extracted from the latter, as long as single photon statistics are maintained. However, the photon losses prolong the acquisition time or deteriorate the signal to noise ratio (SNR) if the acquisition time remains fixed. In applications where the photon burst density must be evaluated (e.g. for molecule transition detection) dead-times can be a problem. The PicoHarp 300 has an extremely short dead time of typically 90 ns, in fact one of the shortest in any comparable instrument available today.

## Timing Resolution

The most critical component in terms of timing resolution in TCSPC measurements will usually be the detector. However, as opposed to analog transient recording, the time resolution of TCSPC is not limited by the pulse response of the detector. Only the timing accuracy of registering a photon determines the resolution. The timing accuracy is limited by the timing uncertainty that the detector introduces in the conversion from a photon to an electrical pulse. This timing error or uncertainty can be as much as ten times smaller than the detector's pulse response. The timing uncertainties are usually quantified by specifying the r.m.s. error (standard deviation) or the Full Width Half Maximum (FWHM) of the timing distribution or instrument response function (IRF). Note that these two notations are related but not identical. Good, but also very expensive detectors, notably Micro Channel Plate PMTs, can achieve timing uncertainties as small as 30 ps FWHM. Lower cost PMTs or SPADs may introduce uncertainties of 50 to 500 ps FWHM.

The second most critical source of IRF broadening in fluorescence lifetime measurements with TCSPC is usually the excitation source. While many laser sources can provide sufficiently short pulses, it is also necessary to obtain an electrical timing reference signal (sync) for comparison with the fluorescence photon signal. Where this signal is derived from, depends on the excitation source. With gain switched diode lasers (e.g. PDL 800-B) a low jitter electrical sync signal is readily available. The sync signal used here is typically a narrow negative pulse of  $-800$  mV into  $50 \Omega$  (NIM standard). The sharp falling edge is synchronous with the laser pulse (e.g.  $< 3$  ps r.m.s. jitter for the PDL 800-B). With other lasers (e.g. Ti:Sa) a second detector must be used to derive a sync signal from the optical pulse train. This is commonly done with a fast photo diode (APD or PIN diode). The light for this reference detector must be derived from the excitation laser beam e.g. by means of a semi-transparent mirror. The reference detector must be chosen and set up carefully as it also contributes to the overall timing error.

Another source of timing error is the timing jitter of the electronic components used for TCSPC. This is caused by the finite rise/fall-time of the electric signals used for the time measurement. At the trigger point of comparators, logic gates etc. the amplitude noise (thermal noise, interference etc.) always present in these signals is transformed to a corresponding timing error (phase noise). However, the contribution of the electronics to the total timing error is usually small. For the PicoHarp it is typically  $< 10$  ps. Finally, it is always a good idea to keep electrical noise pickup low. This is why signal leads should be properly shielded coax cables and strong sources of electromagnetic interference should be kept away from the TCSPC detector and electronics.

The contribution of the time spread introduced by the individual components of a TCSPC system to the total IRF width strongly depends on their relative magnitude. Strictly, the overall IRF is the convolution of all component IRFs. An estimate of the overall IRF width, assuming independent noise sources, can be obtained from the geometric sum of the individual components as an r.m.s. figure according to statistical error propagation laws:

$$e_{system} = \sqrt{\sum e_{component}^2}$$

Due to the squares in the sum, the total will be dominated by the largest component. It is therefore of little value to improve a component that is already relatively good. If e.g. the detector has an IRF width of 200 ps FWHM, shortening the laser pulse from 50 ps to 40 ps is practically of no effect. However, it is difficult to specify a general lower limit on the fluorescence lifetime that can be measured by a given TCSPC instrument. In addition to the instrument response function and noise, factors such as quantum yield, fluorophore concentration, and decay kinetics will affect the measurement. However, as a limit, one can assume that under favourable conditions lifetimes down to 1/10 of the IRF width (FWHM) can still be recovered via deconvolution.

A final time-resolution related issue worth noting is the channel width of the TCSPC histogram. As outlined above, the analog electronic processing of the timing signals (detector, amplifiers, CFD etc.) creates a continuous distribution around any true time value. In order to form a histogram, at some point the timing results must be quantized. This quantization introduces another random error, if chosen too coarse. The quantization step width (i.e. the resolution) must therefore be small compared with the continuous IRF. From an information theoretical point of view this is governed by sampling theory. As a minimum sampling frequency one would therefore assume the Nyquist frequency. I.e. the signal should be sampled at least at twice the highest frequency contained in it. For practical purposes one may wish to exceed this limit where possible, but there is usually no benefit in sampling the histogram at resolutions much higher than 1/10 of the overall IRF of the analog part of the system. Nevertheless, a good resolution is beneficial in data analysis via iterative reconvolution.

## Photon Counting Detectors

### Photomultiplier Tube (PMT)

A PMT consists of a light-sensitive photocathode that generates electrons when exposed to light. These electrons are directed onto a charged electrode called dynode. The collision of the electrons with the dynode produces additional electrons. Since each electron that strikes the dynode causes several electrons to be emitted, there is a multiplication effect. After further amplification by multiple dynodes, the electrons are collected at the anode of the PMT and output as a current. The current is directly proportional to the intensity of light striking the photocathode. Because of the multiplicative effect of the dynode chain, the PMT is a photo-electron amplifier of high sensitivity and remarkably low noise. The high voltage driving the tube may be varied to change the sensitivity of the PMT. Current PMTs have a wide dynamic range, i.e. they can also measure relatively high levels of light. They are furthermore very fast, so that rapid successive events can be reliably monitored. One photon on the photo-cathode can produce a short output pulse containing millions of photoelectrons. PMTs can therefore be used as single photon detectors. In photon counting mode, individual photons that strike the photocathode of the PMT are registered. Each photon event gives rise to an electrical pulse at the output. The number of pulses, or counts per second, is proportional to the light impinging upon the PMT. As the number of photon events increase at higher light levels, it will become difficult to differentiate between individual pulses and the photon counting detector will become non-linear. This usually occurs at 1 to 10 Mcps, dependent on the model. Similarly, in TCSPC applications, individual photon pulses may merge into one as the count rate increases. This leads to pulse pile-up and distortions of the collected histograms.

The timing uncertainty between photon arrival and electrical output (transit time spread) is usually small enough to permit time-resolved photon counting at a sub-nanosecond scale. In single photon counting mode the tube is typically operated at a constant high voltage where the PMT is most sensitive.

PMTs usually operate between the blue and red regions of the visible spectrum, with greatest quantum efficiency in the blue-green region, depending upon photo-cathode materials. Typical maximum quantum efficiencies are about 25 %. For spectroscopy experiments in the ultraviolet-visible-near-infrared region of the spectrum, a PMT is very well suited.

Because of noise from various sources in the tube, the output of the PMT may contain pulses that are not related to the light input. These are referred to as dark counts. The detection system can to some extent reject these spurious pulses by means of electronic discriminator circuitry. This discrimination is based on the probability that some of the noise generated pulses (those from the dynodes) exhibit lower signal levels than pulses from a true photon event. Thermal emission from the cathode that undergoes the full amplification process can usually not be suppressed this way. In this case cooling of the detector is more helpful.

### Micro Channel Plate PMT (MCP)

A Micro Channel Plate PMT consists of an array of glass capillaries (5 to 25  $\mu\text{m}$  inner diameter) that are coated on the inside with an electron-emissive material. The capillaries are biased at a high voltage. Like in the PMT, an electron that strikes the inside wall of one of the capillaries creates an avalanche of secondary electrons. This cascading effect creates a gain of  $10^3$  to  $10^6$  and produces a current pulse at the output. Due to the narrow and well defined electron path inside the capillaries, the transit time spread of the output pulses is much reduced compared to a normal PMT. The timing jitter of MCPs is therefore sufficiently small to perform time-resolved photon counting on a picosecond scale, usually outperforming PMTs. Good but also expensive MCPs can achieve timing uncertainties as low as 25 ps. Microchannel plates are in this respect the best match for the PicoHarp 300 but they are limited in permitted count rate and provide lower sensitivity towards the red end of the spectrum compared to suitably optimized APDs.

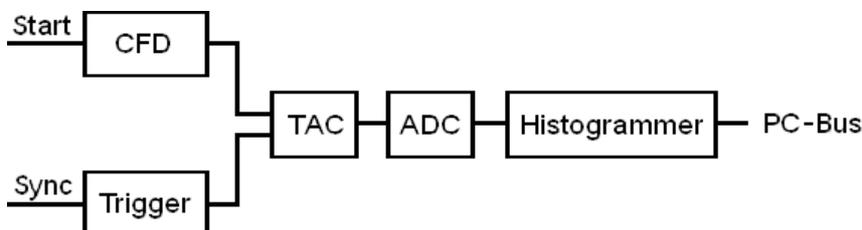
### Avalanche Photo Diode (APD)

APDs are semiconductor devices, usually restricted to operation in the visible to infrared part of the spectrum. Generally, APDs may be used for ultra-low light detection (optical powers  $< 1 \text{ pW}$ ), and can also be used as photon-counters in the so-called "Geiger" mode (biased slightly above the breakdown voltage). In the case of the latter, a single photon may trigger an avalanche of about  $10^8$  carriers. In this mode the device can be used as a detector for photon counting with very accurate timing of the photon arrival. In this context they are also referred to as Single Photon Avalanche Photo Diodes (SPAD). Selected and small devices may achieve timing accuracies down to 50 ps, but small devices are usually hard to align and difficult to focus into. Single-photon detection probabilities of up to approximately 50 % are possible. APDs are often noisier than PMTs, but can have a greater quantum efficiency. Maximum quantum efficiencies reported are about 70 %. The popular commercial devices from Perkin Elmer (SPCM-AQR) provide a timing accuracy of  $\sim 400 \text{ ps}$  (selected) and are specified to have a quantum efficiency of 60 %. These modules are thermoelectrically cooled for low dark count rate and deliver pre-shaped TTL pulses. They are the most common detectors for applications where NIR sensitivity is important, e.g. single molecule detection. To achieve the spe-

cified timing accuracy, exact focusing into the center of the active area is necessary. There may also be a shift and/or broadening of the IRF dependent on count rate.

## Principles Behind the TCSPC Electronics

Conventional TCSPC systems consist of the following building blocks:



The CFD is used to extract precise timing information from the electrical detector pulses that may vary in amplitude. This way the overall system IRF may be tuned to become narrower and some of the random background signal can be suppressed. The same could not be achieved with a simple threshold detector (comparator). Especially with PMTs, constant fraction discrimination is very important, because their pulse amplitudes vary significantly. Particularly pulses originating from random electrons generated at the dynodes of the PMT can be suppressed because their avalanches had less time to amplify, and their corresponding output pulses are small.

The principle of a classical CFD (as found in PicoHarp models 1.0) is the comparison of the original detector signal with an amplified, inverted and delayed version of itself. The signal derived from this comparison changes its polarity exactly when a constant fraction of the detector pulse height is reached. The zero crossing point of this signal is therefore suitable to derive a timing signal independent from the amplitude of the input pulse. In practice the comparison is done by a summation. The timing is done by a subsequent threshold trigger of the sum signal using a settable level, the so called zero cross trigger. Newer CFD designs achieve the same objective by differentiating the input signal and triggering on the zero crossing of the differentiated signal. This is the case for PicoHarp models 2.0.

Making the zero cross level adjustable (slightly above zero) allows to adapt to the noise levels in the given signal, since otherwise an infinitely small signal could trigger the zero cross comparator. Typical CFDs furthermore permit setting of a discriminator threshold that determines the lower limit the detector pulse amplitude must pass. This is primarily used to suppress dynode noise from PMTs.

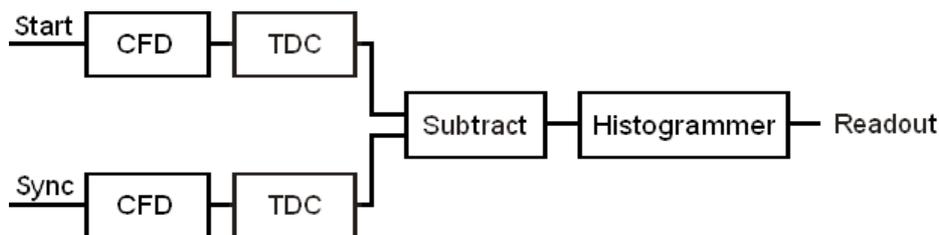
Similar as for the detector signal, the sync signal must be made available to the timing circuitry. Since the sync pulses are usually of well-defined amplitude and shape, a simple settable comparator (level trigger) is often sufficient to adapt to different sync sources. Nevertheless, it can be valuable to have a CFD on both input channels, if e.g. coincidence correlation experiments between two detectors are carried out. A suitably designed CFD is also usable with pre-shaped signals and simply becomes a level trigger in this case. This is the case also for the PicoHarp 300. Note that the input signals must have sufficiently fast rise/fall times.

The signals from the two input discriminators/triggers are in conventional systems fed to a Time to Amplitude Converter (TAC). This circuit is essentially a highly linear ramp generator that is started by one signal and stopped by the other. The result is a voltage proportional to the time difference between the two signals. In conventional systems the voltage obtained from the TAC is then fed to an Analog to Digital Converter (ADC) which provides the digital timing value used to address the histogrammer. The ADC must be very fast in order to keep the dead time of the system short. Furthermore it must guarantee a very good linearity (over the full range as well as differentially). These are criteria difficult to meet simultaneously, particularly with ADCs of high resolution (e.g. 12 bits) as is desirable for TCSPC over many histogram channels.

The histogrammer has to increment each histogram memory cell, whose digital address in the histogram memory it receives from the ADC. This is commonly done by fast digital logic e.g. in the form of Field Programmable Gate Arrays (FPGA) or a microprocessor. Since the histogram memory must also be available for data readout, the histogrammer must occasionally stop processing incoming data. This prevents continuous data collection. Sophisticated TCSPC systems solve this problem by switching between two or more memory blocks, so that one is always available for incoming data.

While this section so far outlined the typical structure of conventional TCSPC systems, it is worth noting that the design of the PicoHarp 300 is somewhat different. Today, it is state-of-the-art that the tasks conventionally performed by TAC and ADC are carried out by a so called Time to Digital Converter (TDC). These circuits allow not only picosecond timing but can also extend the measurable time span to virtually any length by means of digital counters. The PicoHarp 300 does not use one such circuit but two. They independently work on each input channel and provide picosecond arrival times that then can be processed further, with a lot more options than in conventional TCSPC systems. In the case of classical TCSPC, this processing con-

sists of a subtraction of the two time figures and histogramming of the differences. This is identical to the classical start-stop measurements of the conventional TAC approach.



The full strength of the PicoHarp design is exploited by collecting the unprocessed independent arrival times as a continuous data stream for more advanced analysis. Details on such advanced analysis can be found in the literature. In this case the on-board memory is reconfigured as a large data buffer (FiFo) so that count rate bursts and irregular data transfer are decoupled. This permits uninterrupted continuous data collection with high throughput. This mode of operation is called Time-Tagged-Time-Resolved (TTTR) mode. For the PicoHarp, corresponding firmware upgrades must be purchased to make use of these features. Details are described in section 5.4 on Time-Tagged Mode Measurements.

### Forward and Reverse Start-Stop Mode

For simplicity it is most convenient to assume that the time delay measurement is directly causal, i.e. the laser pulse causes a photon event and one measures the time delay between laser pulse and the subsequent photon event. However, most conventional TCSPC systems need to give up this logical concept because of the high repetition rates of the typical excitation lasers: Since the time measurement circuit cannot know in advance whether there will be a fluorescence photon, it would have to start a time measurement upon each laser pulse. Considering that conventionally conversion times are in the region of 0.3 to 2  $\mu\text{s}$ , any excitation rate in excess of 0.5 to 3 MHz would overrun the time measurement circuits. In fact they would most of the time be busy with conversions that never complete, because there is no photon event at all in most cycles. By reversing the start and stop signals in the time measurement, the conversion rates are only as high as the actual photon rates generated by the fluorescent sample. These are (and must be) only about 1 to 5 % of the excitation rate and can therefore be handled easily by the TAC/ADC or TDC. The consequence of this approach, however, is that the times measured are not those between laser pulse and corresponding photon event, but those between photon event and the next laser pulse. This still works (by software data reversing) but is inconvenient in two ways:

1. Having to reverse the data leads to unpleasant relocation of the data displayed on screen when the time resolution is changed.
2. Changing between slow and fast excitation sources requires reconnecting to different inputs, possibly causing trouble when there is no CFD at one of the inputs.

The PicoHarp 300 is revolutionary in this respect, as it allows to work in forward start stop mode, even with fast lasers. This is facilitated by two design features:

1. The independent operation of the two time digitizers, and
2. a programmable divider in front of the sync input.

The latter allows to reduce the input rate so that the period is at least as long as the dead time. Internal logic determines the sync period and re-calculates the sync signals that were divided out. It must be noted that this only works with stable sync sources that provide a constant pulse-to-pulse period, but all fast laser sources known today meet this requirement within an error band of a few picoseconds. Note: for slow sync sources (< 1 MHz) the sync divider must not be used (set to None). Similarly, the divider must not be used if the channel 0 input is used for a photon detector (coincidence correlation measurements etc.).

In summary: The PicoHarp 300 is designed to always work in forward start-stop mode.

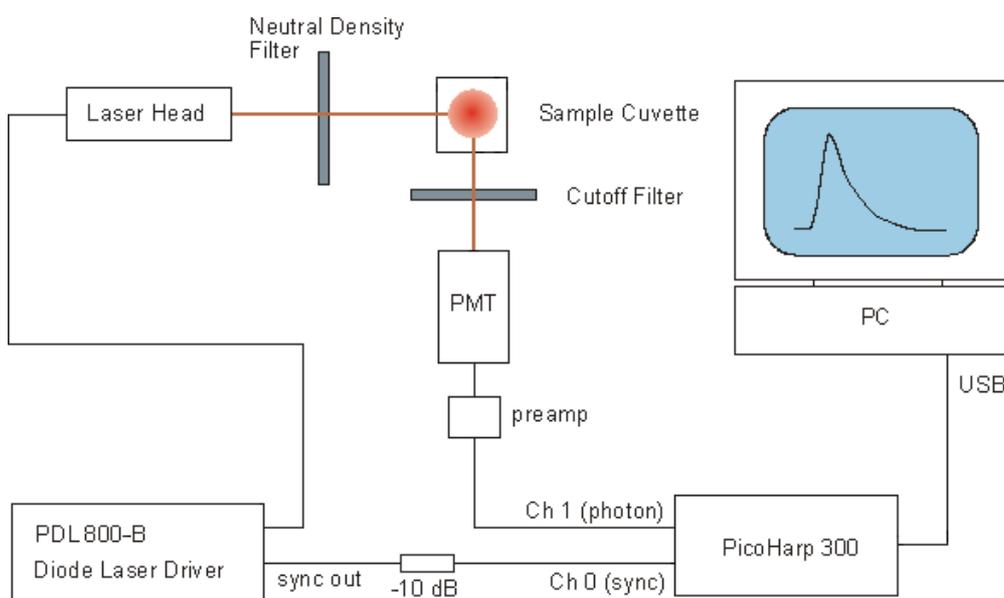
### Experimental Setup for Fluorescence Decay Measurements with TCSPC

The figure below shows a typical setup for fluorescence lifetime measurements with TCSPC. The picosecond diode laser (PDL 800-B) is triggered by its internal oscillator (settable at 2.5, 5, 10, 20 and 40 MHz). The driver unit is physically separate from the actual laser head, which is attached via a flexible cable. This permits to place the small laser head conveniently anywhere in the optical setup. The light pulses of typically < 70 ps FWHM, are directed toward the sample cuvette via appropriate optics. A neutral density filter can be used to attenuate the light levels if necessary. Upon excitation, the fluorescent sample will emit light at a longer wavelength than that of the excitation light. The fluorescence light is filtered out from scattered excitation light by means of an optical cutoff filter (other configurations may utilize a monochromator here). Then it

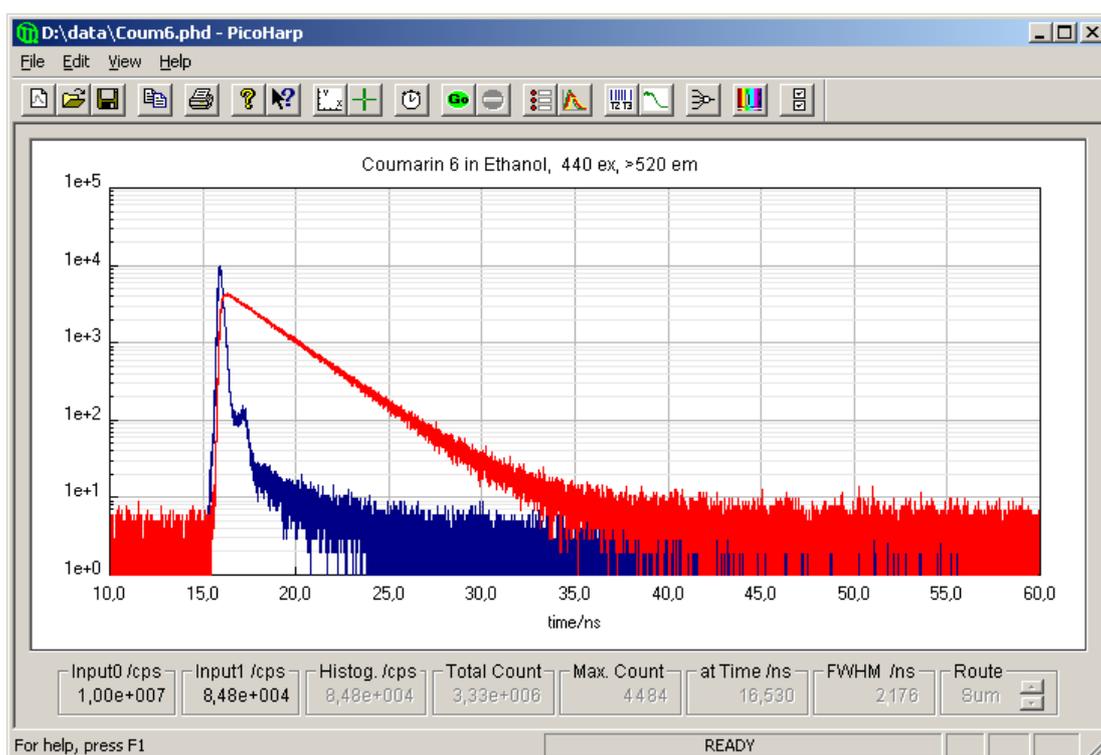
is directed to the photon detector, again possibly via some appropriate collection optics, e.g. a microscope objective or a lens.

As a photon detector the H5783 PMT from Hamamatsu is very convenient. It only needs a 12 V supply and permits an instrument response of < 250 ps, allowing lifetime measurements even much shorter than this via deconvolution. If a higher time resolution is required, the detector of choice is an MCP-PMT. The electrical signal obtained from the detector (small negative pulses of typically -10 to -50 mV) is fed to the TCSPC electronics via a preamplifier (e.g. PAM 102 from PicoQuant). This gives pulses of -100 to -500 mV. Cabling is double shielded 50 Ω coax cable. If the detector is a SPAD module with TTL output (e.g. SPCM-AQR from Perkin Elmer) then an inverter/attenuator (SIA 400) must be inserted.

The PDL 800-B laser driver directly provides the electric sync signal needed for the photon arrival time measurement. This signal (a narrow negative pulse) is also fed to the TCSPC electronics via a high quality 50 Ω coax cable. When using the PicoHarp 300 in combination with the PDL 800-B, the sync pulse adapter LTT 100 (brass cube with SMA connectors) or a 10 dB attenuator should be inserted directly at the sync output of the laser driver. This reduces crosstalk into the relatively weak detector signals.



If the laser does not provide an electrical sync signal (e.g. Ti:Sa lasers), a sync detector (photo diode) such as the TDA 200 must be used. The following figure shows TCSPC histograms obtained with this setup. Excitation source was a PDL 800-B with a 440 nm laser head running at 10 MHz repetition rate.



The narrower (blue) curve represents the system IRF, here dominated by the detector (PMT). The other curve (red) is the fluorescence decay from a solution of Coumarin 6 in ethanol, a fluorescent dye with fairly short fluorescence lifetime (~2.5 ns).

Note that the count rate was adjusted to < 1% of the laser rate to prevent pile-up. The plot in a logarithmic scale shows the perfect exponential nature of the decay curve, as one would expect it. Note that this is obtained even without a deconvolution of the IRF. The approximate mono-exponential fluorescence lifetime can be obtained from a simple comparison of two points in the curve with count rates in the ratio of  $e : 1$  (e.g. 27,183 : 10,000). In this particular experiment this resulted in a lifetime estimate of 2.5 ns in perfect agreement with published results for this dye. For a precise measurement one would perform a numerical exponential fit with IRF deconvolution (typically implemented as an iterative reconvolution). This would result in slightly shorter lifetimes since the IRF broadens the decay. Indeed one can measure lifetimes significantly smaller than the IRF with this method. Additionally, the r.m.s. residue from the fit can be used to assess the quality of the fit and thereby the reliability of the lifetime measurement. The FluoFit decay fit software package from PicoQuant provides this functionality.

### Further Reading

1. Lakowicz, J. R.,  
Principles of Fluorescence Spectroscopy, 3<sup>rd</sup> Edition  
Springer Science+Business Media, New York, 2006
2. O'Connor, D.V.O., Phillips, D.,  
Time-correlated Single Photon Counting  
Academic Press, London, 1984
3. O'Connor, D.V.O., Ware, W.R., Andre, J.C.,  
Deconvolution of fluorescence decay curves. A critical comparison of techniques.  
*Journal of Physical Chemistry*. 83, 1333-1343, 1979
4. Becker, W.,  
Advanced Time-Correlated Single Photon Counting Techniques  
Springer, Berlin, 2005
5. Wahl M., Rahn H.-J., Gregor I., Erdmann R., Enderlein J.,  
Dead-time optimized time-correlated photon counting instrument with synchronized, independent timing channels.  
*Review of Scientific Instruments*, Vol.78, 033106 (2007)
6. [http://www.picoquant.com/\\_biblio.htm](http://www.picoquant.com/_biblio.htm)  
Bibliography listing all publications with work based on PicoQuant instruments
7. <http://www.picoquant.com/showpapers.php?sid=7>  
Bibliography of publications of work using the PicoHarp 300

## 3. Hardware and Software Installation

### 3.1. General Installation Notes – Read This First

The PicoHarp software version 2.3 is compatible with Windows 2000, XP and Vista<sup>1</sup>.

When handling the PicoHarp 300 system, be sure to avoid electrostatic discharges, especially when connecting cables.

In order to use the PicoHarp software with the PicoHarp 300 hardware, a compatible device driver must be installed. The PicoHarp 300 is a USB 2.0 'plug and play' device, this means the necessary resources and drivers are allocated automatically by the operating system. Windows will automatically recognize when such a device is connected and prompt the user for the appropriate driver disk. Note that the software setup for the PicoHarp 300 pre-installs the driver in the application directory in a subfolder `driver` so that windows will find it there. This requires that the software setup is performed before connecting the device. In case you are prompted for the driver location, please point the driver installation wizard to that folder. Just in case, the driver folder is also provided on the installation media. Some windows versions verbosely warn about drivers not being validated by Microsoft. You can ignore these warnings.

On some Windows versions you will need administrator status to perform software setup and deinstallation. You can install the software for use by all users on that computer even if they have limited access rights. The PicoHarp software will maintain individual settings for each user in the Windows registry. When switching between users with sessions still running in the background you cannot run the PicoHarp software in multiple sessions at the same time.

The PicoHarp software 2.3 is also suitable for x64 versions of Windows XP and Vista, where a suitable 64 bit device driver will be installed. The PicoHarp software as such runs as a 32 bit application on WOW64.

#### Disclaimer

PicoQuant GmbH disclaims all warranties with regard to the supplied software and documentation including all implied warranties of merchantability and fitness for a particular purpose. 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, inability to use, or performance of this software and associated documentation.

#### License and Copyright Notice

With this product you have purchased a license to use the PicoHarp 300 software on a single PC. You have not purchased any other rights to the software itself. The software is protected by copyright and intellectual property laws. You may not distribute the software to third parties or reverse engineer, decompile or disassemble the software or part thereof. You may use and modify demo code to create your own software. Original or modified demo code may be re-distributed, provided that the original disclaimer and copyright notes are not removed from it. Copyright of this manual and on-line documentation belongs to PicoQuant GmbH. No parts of it may be reproduced, translated or transferred to third parties without written permission of PicoQuant GmbH.

PicoHarp is a registered trademark of PicoQuant GmbH. Other products and corporate names appearing in this manual may or may not be registered trademarks or subject to copyrights of their respective owners. PicoQuant GmbH claims no rights to any such trademarks. They are used here only for identification or explanation and to the owner's benefit, without intent to infringe.

---

<sup>1</sup> Trademarks of Microsoft Inc.

## 3.2. Software Installation

Before installing the new software you should (if applicable) back-up all PicoHarp data files you created in any older PicoHarp directory (previous versions etc.) as well as any critical data on your PC. For a clean start we also recommend to uninstall all older PicoHarp software.

The PicoHarp 300 setup files are normally provided on a CD supplied with your PicoHarp device. The installer program file containing the complete distribution is `setup.exe`. If you received the software package by email or any other means of electronic distribution, it will probably be packed in a ZIP-File. In this case you can unzip this file to a temporary hard disk location of your choice and run the setup from there. Note that if the distribution was by email, the installer file may have been renamed to `setup.ex_` in order to let it pass strict email filters. In that case, please rename it back to `setup.exe`.

In order to use the PicoHarp software with the PicoHarp 300 hardware, a suitable device driver must be installed. The PicoHarp 300 is a USB 2.0 'plug and play' device, this means the necessary resources and drivers are allocated automatically by the operating system. Windows will automatically recognize when such a device is connected and prompt the user for the appropriate driver disk. Note that the software setup for the PicoHarp 300 preinstalls the driver so that upon connecting the device it is readily available for the Windows plug&play device management and you should not be prompted for the driver location. This requires that the software setup is performed before connecting the device. Dependent on your version of Windows you may still be prompted to confirm the final driver installation. Some windows versions verbosely warn about drivers not being validated by Microsoft. You can ignore these warnings.

When upgrading from an earlier version of the PicoHarp software on a machine where the device driver is already installed, you may also want to perform a driver update via the Windows device manager. For that purpose, the device must be connected but no software should be using it.

On some Windows versions you will need administrator status to perform the setup. You can run the setup program directly from the CD or from the hard disk location where you unpacked the electronic distribution.

To perform the installation, insert the installation disk in your CD drive. Open the CD location either from the Windows desktop or with the Windows Explorer. If you unpacked the setup file to a hard disk location, open that location. Run `Setup.exe`. The setup program will guide you through the installation process step by step.



When asked for a destination folder for the new software, please accept the default path or select another according to your program storage policies. This is where the PicoHarp files will be installed. To avoid confusion, make sure not to specify the path of an older PicoHarp version that you have not uninstalled or that of any other program on your PC.

The default location is: `\Program Files\PicoQuant\PicoHarp300v23`.

Note that dependent on your Windows installation the folder `Program Files` may be spelled in your national language.

Setup will also create a dedicated "program folder" for the new PicoHarp 300 software that will later appear in the Start Menu. You can accept the default folder name or select another according to your own naming policies. However, you should make sure not to specify the folder name of an older PicoHarp version that you have not uninstalled or the dedicated folder of any other program.

The setup program will install the following files in the chosen destination folder:

- `PicoHarp.exe`            the main executable program
- `PicoHarp.chm`            the PicoHarp compiled help file
- `Manual.pdf`              the manual in Adobe PDF format
- `Readme.txt`              a text file with program information
- `unins000.exe`            the uninstaller program file
- `unins000.dat`            the uninstaller information file

Setup will also create a subdirectory `DRIVER` which contains the device drivers. A further subdirectory named `FILEDEMO` contains demo source code for access to PicoHarp 300 data files in various programming languages. Another folder `SAMPLEDATA` will be created with samples of PicoHarp data files.

After the installation the PicoHarp software should be available under the Windows Start Menu under: `Programs/PicoQuant-PicoHarp 300 v2.3` (or the folder name you chose during setup). However, before running the software, please proceed to the hardware installation step.

### 3.3. Device Installation

Make sure to avoid electrostatic discharge when handling the PicoHarp system, especially when connecting cables. Note that PMT detectors operate with high voltage that may discharge through the signal cable. Make sure such detectors are switched off and fully discharged before connecting them.

To connect the PicoHarp device to the PC, first consult your PC manual as to whether and where it provides high speed USB 2.0 connectors. The PicoHarp system will NOT work through USB 1.0. All current PCs should provide USB 2.0 connectivity. Make sure the drivers for the USB 2.0 host controller are correctly installed. If this is not ensured, USB 2.0 interfaces may only run as USB 1.0 interfaces. Older versions of Windows may not have the appropriate drivers, so you may need to install them from the mainboard driver disk. For older PCs without USB 2.0 support it is possible to install USB 2.0 adaptor cards. However, such computers may often be outdated in many other respects, so if possible, consider replacing them (See section 8.3 on Hardware Technical Data for requirements/recommendations). Always use a USB cable rated for high-speed USB 2.0. The cable length allowed by USB 2.0 must not exceed 5 metres (~16 ft). For best reliability we recommended to use the provided cable of 3 metres length. Note that the USB specification does not allow cable extensions other than dedicated active extension cables or HUBs. The PicoHarp device should work flawlessly through USB 2.0 high-speed HUBs. This is also a valid way of extending the maximum cable length beyond 5 metres. After a HUB, another cable of up to 5 metres is allowed. Note, however, that chains of multiple HUBS may lower the data throughput. For the same reasons it is recommended not to connect other bandwidth demanding devices to the same HUB.

The inputs for the detector/sync signals are two SMA connectors located on the front panel of the PicoHarp 300, labelled Channel 0 and Channel 1. In case of time resolved fluorescence experiments with a pulsed excitation source, the sync signal must be connected to Channel 0 and the detector signal must be connected to Channel 1. If coincidence correlation experiments between two detector signals are to be carried out, connect one detector each to the two inputs. Both inputs are dynamically terminated with 50  $\Omega$  internally. Note that they need negative going input signals. Use quality 50  $\Omega$  coax leads with appropriate connectors. For interfacing to BNC connectors use standard adaptors. Carefully screw on the SMA connectors for sync and detector until they are moderately tight. Do not use wrenches. Connect the cable ends to the appropriate signal sources (50  $\Omega$ ) in your experimental setup. The PicoHarp 300 inputs accept negative pulses going from zero to max. -1 V. Both inputs should be operated with similar pulse amplitudes to minimize crosstalk. The optimum range is -100 to -500 mV. Most PMT and MCP detectors will require a pre-amplifier to reach this level. TTL-SPAD-detectors (Perkin-Elmer SPCM-AQR) must be connected through a pulse inverter (PicoQuant SIA 400). Weak PMT detectors should be connected through a 20 dB high speed pre-amplifier. MCP-PMT detectors should be connected through an amplifier with slightly higher gain. Suitable devices are available from PicoQuant. Be sure to switch the high voltage supply of PMTs off and allow their electrodes to discharge before connecting them. Their high voltage charge may damage the pre-amplifier. Observe the allowed maximum ratings for the input signal levels. If you are not sure what signals your setup delivers, use a fast oscilloscope to check the signal level and shape before connecting them to the PicoHarp. All signals should have rise times of no more than 2 ns. Slower signals may not be seen by the PicoHarp and will certainly degrade timing accuracy.

When using the PicoHarp 300 in combination with the PDL 800-B, the sync pulses from the laser driver should be attenuated by 10 to 15 dB to fall in the optimum range for smallest crosstalk.

The PicoHarp hardware is offered in two versions. The standard product PicoHarp 300 provides more features. The PicoHarp 300E is a leaner variant targeting cost sensitive applications where some functionality is not required. The features that distinguish the PicoHarp 300 from the 300E model are support for routing devices for simultaneous measurements with several detectors and interfacing with external hardware via TTL marker signals. The reduced functionality is physically evident through the presence of two multi-pin connectors at the PicoHarp 300 front panel that are not present at the 300E model. If these connectors are present, do not connect anything other than dedicated hardware to these connectors. They are provided for hardware expansion (notably routing and experiment control) and must not be used otherwise. See section 8.3 on Hardware Technical Data for pin assignments. It is recommended to start instrument setup without anything connected to the control ports.

After connecting the PicoHarp to the PC via USB, Windows should detect the device and install the driver. Note that the software setup for the PicoHarp 300 pre-installs the driver so that it should be found automatically. When prompted for the driver location, please point the driver installation wizard to the folder `DRIVER` in the installation folder or the equivalent folder on your installation media. If you downloaded and unzipped the software to a temporary disk location, please select the driver folder in that location. After driver installation Windows should report the device as ready to use. If you wish to verify correct installation you can check the presence of 'PicoHarp 300' as a USB device in the Windows device manager. For troubleshooting see the section 3.4 .

### 3.4. Installation Troubleshooting

After completion of the software setup, the PicoHarp device should be listed in the Windows Device Manager under the USB tree. Right click *My Computer* > left click *Properties* > *Device Manager* to check if the device is free of conflicts and/or if the device driver is installed correctly. Under *USB* look for a device named *PicoHarp 300* and inspect its *Properties*.

You can also use the Windows system information facilities (*Start* > *Run* > *msinfo32*). In the System Information utility inspect *Software environment* > *System Drivers* to check if the PicoHarp device driver PQUSB.SYS (PQUSB64.SYS on x64) is correctly installed and running.

You can also repeat the driver and software installation if necessary. Make sure the software is not installed in multiple places. If problems persist, disconnect the device, uninstall the software and repeat the setup procedure. If this does not resolve the problem, see section 7 on Problems, Tips & Tricks for support.

### 3.5. Uninstalling the Software

Before uninstalling the PicoHarp software you should back-up all PicoHarp data files you created in the PicoHarp directory.

Do not manually delete any program files from the installation folder as it will render a clean uninstall impossible.

To uninstall the PicoHarp software from your PC you may need administrator rights (dep. on Windows version and security settings). Choose *Start* > *Settings* > *Controlpanel* > *Software* and select *PicoQuant - PicoHarp300 vx.x* for uninstallation. This will remove all files that were installed by the PicoHarp setup program but not the user data that may have been stored. If there was user data in the program folders or any subfolders, these will not be deleted by the uninstall program. If intended, you need to delete these files or folders manually. Nevertheless, it is recommended to back-up valuable measurement data before uninstalling the software.

## 4. Software Overview

### 4.1. Starting the Program

After correct installation the Start Menu *Programs* contains a shortcut to the PicoHarp software. To start the PicoHarp software select *PicoQuant - PicoHarp 300 v2.3*. Note that after switching the PicoHarp on, you need to allow a warm-up period of at least 20 minutes before using the instrument for serious measurements. You can use this time for set-up and preliminary measurements.

If the PicoHarp software cannot find the device (or if there are driver problems) it will display a notification message, but it can still start. Device dependent toolbar buttons and functions of the program will then remain disabled. This allows you to use the software without the PicoHarp hardware, e.g. to view or print files on another computer.

If the PicoHarp 300 is correctly installed and there are still hardware related errors, you can use the Windows device manager for troubleshooting (see section 3.4). If problems cannot be resolved, see section 7 on Problems, Tips & Tricks for support. If possible, try the device on another computer.

For regular use of the PicoHarp you may want to create an icon for it on your Windows desktop. You can do this conveniently via a right mouse click on the start menu entry (*Send To > Desktop - create shortcut*). Alternatively use the Windows Explorer to locate the file `PicoHarp.exe` in the directory you selected for installation and drag the icon onto the desktop. This will create a link to the PicoHarp executable file. You can now start it conveniently by double-clicking the icon on the desktop.

You can also start up the PicoHarp software directly from a PicoHarp histogram file (`*.phd`) by double clicking on the file or dragging it onto the PicoHarp icon.

### 4.2. The Main Window

The PicoHarp software provides a measurement control interface to the PicoHarp hardware and an online histogram display. The PicoHarp main window accommodates the histogram display area. The histogram 'curves' are displayed here.

Above the display area is the toolbar. Here you can access frequently used commands by simple mouse click. Above the toolbar is the menu bar for access to additional commands. At the bottom of the histogram display area is a set of 'panel meters' showing count rates, count sums and histogram peak characteristics. These will be updated continuously, some only when a measurement is in progress. They can be enlarged by double-clicking, which is useful for optical alignment etc. when the PC is at some distance.

In the top center of the display area a title line is shown. This can be double-clicked to edit the title. When editing the title, note that only the first line will appear in the display. The remaining lines are meant to be used as a file comment. All lines will be stored with the file data, but the maximum total is 256 characters.

The main window is resizeable and the actual histogram display will adapt its size accordingly. If you make the window smaller than the minimum histogram display, two scrollbars will permit access to hidden window areas. Note that the actual size of your PicoHarp main window depends on the system font selected. It will scale itself to the size of the current display font. On systems with display resolutions of less than 800x600 the full window may fit on the screen only if the small display font is selected. Screen resolutions under 800x600 are not suitable for serious work with the program.

Note that the position and size of the main window on the screen will be stored in the Windows registry and retrieved during the next program start. The registry settings are kept separately for each user.

Consult the section 6.1 Main Window for further command descriptions. Toolbar, Menus, panel meters etc. will be explained in the next sections.

At the very bottom of the main window there is a status bar. The leftmost area of the status bar describes actions of menu items as you navigate through menus. Similarly it shows messages that describe the actions of toolbar buttons. The second status bar area from the left shows the current measurement status of the PicoHarp. Next to it you may find a warnings icon that you can click for details (see also section 8.1).

When the PicoHarp software is running with functional hardware it continuously collects information about the input signals and the current acquisition settings. If these settings together with the input rates indicate possible errors, the software will activate the warning icon. The warning icon can be clicked to display a list of current warnings together with a brief explanation of each warning (see also section 8.1).



The rightmost area of the status bar indicates if the `<Tab>` and `<NumLock>` keys are latched down.

## 4.3. The Toolbar

The toolbar is displayed across the top of the PicoHarp main window, below the menu bar. The toolbar provides quick mouse access to frequently used commands and tools. Note that some buttons may be grayed out (disabled) dependent on the installed software components and/or the state of the hardware.

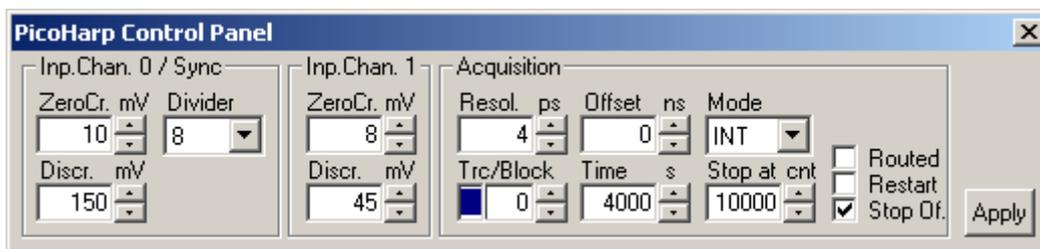


To hide or display the Toolbar, choose Toolbar from the View menu ( $\langle ALT \rangle + V, T$ ). The toolbar is capable of 'docking' i.e. you can move it to another edge of your window or even have it displayed as a separate window. The following table explains the individual buttons.

Click...	To
	Open a blank histogram with default control panel settings
	Open an existing histogram file. Displays the <i>Open</i> dialog box, in which you can locate and open the desired file.
	Save the current histogram data with its current name. If you have not named the file, the <i>Save As</i> dialog box is displayed.
	Print the currently displayed histogram curves.
	Activate context help.
	Copy the currently displayed curves to the clipboard (ASCII export).
	Launch the axis panel.
	Launch the data cursor dialog.
	Calibrate the time measurement hardware
	Start measurement based on current PicoHarp control panel settings
	Stop measurement and histogram accumulation.
	Launch the PicoHarp control panel
	Launch the trace mapping dialog.
	Launch the TTTR mode dialog.
	Launch the TTTR mode real-time correlator dialog.
	Launch the dialog for PHR 800 router setup.
	Launch the dialog for monochromator control and TRES setup..
	Launch the software settings dialog.

## 4.4. The Control Panel

The PicoHarp control panel is a dialog box for setting the parameters for hardware adjustment and data acquisition. It is implemented as a 'non-modal' dialog box, i.e. it does not have to be closed before the main window can continue. This way you can make changes to your settings in the control panel and watch their effect on a running measurement in the main window immediately. Nevertheless, you may close the control panel and restore it at any time by clicking the control panel button on the toolbar or pressing <Alt>+C. Note that the control panel memorizes its last position on the screen, even across sessions. It will re-open where it was last moved.



### Input Channel 0 / Sync

Zero Cross	edit box and spin control
Discriminator Level	edit box and spin control
Sync Divider	edit box and spin control

### Input Channel 1

Zero Cross	edit box and spin control
Discriminator Level	edit box and spin control

### Acquisition

Resolution	edit box and spin control
Offset	edit box and spin control
Time	edit box and spin control
Trc/Block	trace colour indicator, edit box and spin control
Stop At	edit box and spin control
Mode	selection box (OSC, INT, TRES)
Routed	check box
Restart	check box
Stop on Overflow	check box

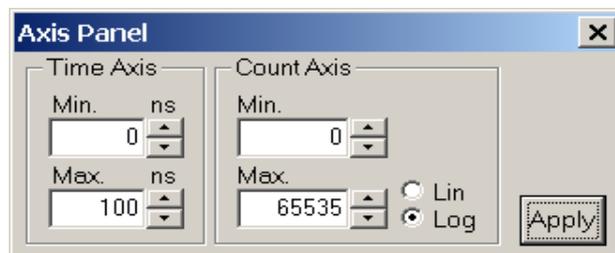
The edit boxes are for keyboard entry. The values must be confirmed with either the <Enter> key or the *Apply* button. The spin controls can be used to increment or decrement the edit box values. In this case the changes take effect immediately without need for <Enter> or *Apply*. Check boxes have their denoted effect when the tick is shown. They can be toggled with a mouse click. Groups of radio buttons are like check boxes but mutually exclusive.

Note that the settings of the control panel as well as the positions of control panel and main window on the screen will be stored in the Windows registry and retrieved during the next program start. The registry settings are kept separately per user. When a PicoHarp data file is loaded, the control panel settings will change according to the settings in that file.

The individual control panel items are discussed in the section *Control Panel* in the *Controls and Commands Reference*.

## 4.5. The Axis Panel

The axis settings panel is a dialog box for setting the axis range for the histogram display in the main window. It is implemented as a 'non-modal' dialog box, i.e. it does not have to be closed before the main window can continue. This way you can make changes in the axis panel and watch their effect in the main window immediately. Nevertheless, you may close the axis panel and restore it at any time by clicking the axis panel button on the toolbar or pressing <ALT>+A. The panel will also open if you double click the axes in the main window.



The axis panel consists of two groups containing edit boxes and other controls for related parameters. These groups and their respective controls are:

### Time Axis

<i>Minimum</i>	edit box and spin control
<i>Maximum</i>	edit box and spin control

### Count Axis

<i>Minimum</i>	edit box and spin control
<i>Maximum</i>	edit box and spin control
<i>Linear</i>	radio button
<i>Logarithmic</i>	radio button

The edit boxes are for keyboard entry. The values must be confirmed with either the <Enter>-key or the *Apply* button. The spin controls can be used to increment or decrement the edit box values. In this case the changes take effect immediately without need for <Enter> or *Apply*. Check boxes have their denoted effect when the tick is shown. They can be toggled with a mouse click. Groups of radio buttons are like check boxes but mutually exclusive. They work like on an old radio: one that was pressed pops out when another one is pressed.

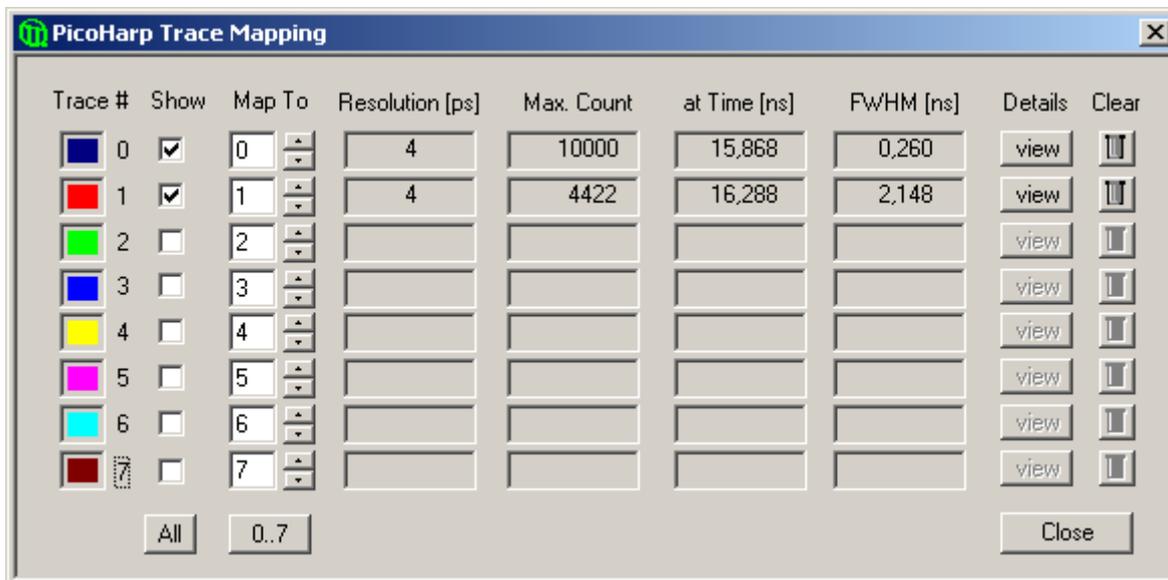
Note that the settings of the axis panel as well as the positions of the panel on the screen will be stored in the Windows registry and retrieved during the next program start. The registry settings are kept separately per user.

The individual axis panel items are discussed in the section 6.4 Axis Panel in the Controls and Commands Reference.

To begin, use logarithmic display to make sure even weak signals can be seen.

## 4.6. The Trace Mapping Dialog

The PicoHarp software can measure histograms in up to 512 memory blocks. Out of these up to 8 curves can be displayed. The *Trace Mapping* dialog is used to select the curves to display. It also allows to view curve details and cleanup. You can use the *Trace Map* button on the Toolbar or click the curve colour indicator in the control panel to launch the dialog.



In the *Trace Mapping* dialog you can tick the individual boxes 'Show' to display a curve. Select the number of the memory block you wish to map the individual display curve to. Choose the active memory block you wish to use for the next measurement in the PicoHarp control panel with the *Map To* edit box and spin control.

The dialog also provides some statistics on each curve (central matrix of figures). *Resolution* is the channel width (bin width) of the histogram in picoseconds. Next is the *Max. Count*, the count in the highest point of the curve. The column *at Time* shows the time corresponding to the *Max. Count* channel. Leftmost there is the Full Width Half Maximum (*FWHM*) of the curve peak (usually for IRF traces).

There are also some buttons for frequently required actions: The button 'view' can be clicked to see more detailed curve information such as time of recording, acquisition settings and count rates. The button *All* can be clicked to tick all traces as shown. The button *0..7* can be clicked to set the default mapping of trace 0..7 to block 0..7. The *Clear* buttons (trash cans) can be clicked to delete the contents of individual blocks.

Note that the Trace Mapping Dialog is non-modal. This means the dialog can remain open while a measurement is in progress, so that adjustments can be made under immediate visual control, similar to the operation of the control panel. Note also that a measurement can be running in a block that is not mapped or shown.

The individual trace mapping dialog items are discussed in section 6.5 "Trace Mapping Dialog" in the Controls and Commands Reference.

## 5. Specific Measurement Tasks

### 5.1. Setting up the Input Channels

If you are running TCSPC measurements for the first time we strongly recommend you read our TechNote on TCSPC first. The document is provided on your distribution CD. It will help you getting a background on the concepts and terms used here.

In order to acquire any data, the two input channels Channel 0 (usually sync) and Channel 1 (usually photon detector) of the PicoHarp 300 must be set to match their electrical input signals. The PicoHarp input channels are designed identically. Both are designed as a Constant Fraction Discriminator (CFD). They can handle an input pulse peak range from 0 to  $-1$  V for regular operation. The absolute maximum rating (beyond which damage may occur) is  $-2$  V.

In case of coincidence correlation experiments between two detectors, the channels are used for one detector each. In time-resolved fluorescence with a pulsed excitation source (typically a laser) Channel 0 is used as the sync input from the laser and Channel 1 for the detector. Here we focus on the latter, more common case. Perform the following steps to set up the input discriminators.

#### Setting Up the Sync Input (Input Channel 0)

For typical fluorescence decay measurements, the PicoHarp needs an electrical sync signal from the light source. The PDL 800 family of diode lasers from PicoQuant provide this signal directly. If the laser does not provide an electrical sync signal (e.g. Ti:Sa lasers), a sync detector (photo diode) such as the TDA 200 must be used. The sync signal must consist of pulses with falling active edge, e.g. a NIM type signal is appropriate. This is a steep negative pulse (0.5 to 10 ns wide) of typically  $-800$  mV into  $50 \Omega$ . The PicoHarp can actually handle  $-1$  V but amplitudes in excess of  $-800$  mV may cause excess interference and crosstalk between the input channels. Amplitudes around  $-200$  to  $-800$  mV are best in terms of timing accuracy and lowest histogram ripple. Another rule of thumb for lowest signal distortion is to use signals of similar amplitude on both channels. SMA inline attenuators of suitable bandwidth can be used to adjust this. The adjustable CFD level is restricted to  $-800$  to  $0$  mV. Initially you should set it to half of the sync pulse amplitude. The zero cross setting should be set to  $10$  mV initially.

Unless you are sure what kind of signal your sync source delivers, use a fast oscilloscope ( $50 \Omega$  input) to check the pulse shape, polarity and amplitude. The leading edge should be steep, there should also be no excessive overshoot or ringing. All signals should have rise times of no more than  $2$  ns. Slower signals may not be seen by the PicoHarp and will certainly degrade timing accuracy. The pulse width should be  $0.5$  to  $30$  ns. The upper limit is not critical, but the pulse width should be as small as necessary to fit in half the sync period.

If the signal is satisfactory, connect the signal to the Channel 0 input and start the PicoHarp software if it is not yet running. A detector signal (Channel 1) is not required at this point, but it does no harm if it is also connected. Open the PicoHarp control panel. Leftmost, there is a group of controls for Channel 0 / sync. Find the edit box and spin control for the CFD discriminator level. The discriminator level should be set to a value around half of the amplitude of the sync pulses. Note that the control panel shows absolute numbers from  $0$  to  $800$  despite the real signals are negative going. Next, set the Zero Cross level to  $10$  mV. Then look at the Sync Divider. It must be set so that the sync rate divided by the shown divider value does not exceed  $10$  MHz. E.g. for a diode laser with  $80$  MHz repetition rate it must be set to  $8$ . For a slow source such as a flash lamp it must be set to 'None'. Generally it is recommended to use the lowest suitable divider.

If the sync source is active, the Input 0 rate will now be displayed at the bottom left rate meter in the PicoHarp main window. Note that the rate meter internally corrects for the chosen divider setting, so that the meter always shows the undivided input rate. The meter display should therefore match the rate delivered from the source. The rate meter will be refreshed every  $0.1$  to  $1$  seconds, determined by the value in the Software Settings Dialog. The Input 0 (sync) rate should be displayed very accurately. Large fluctuations or occasional zeros indicate an incorrect discriminator level setting or an unstable sync signal. Try to vary the discriminator level to obtain a stable rate meter reading. If the sync rate is stable and as expected, you can proceed to the set-up of the other input. A last fine tuning of the discriminator and zero cross level can be done when the detector input is up and running.

#### Setting Up the Channel 1 Input

As noted before, the PicoHarp 300 input channels are designed electrically identical, both with a CFD. They can handle an input voltage range of  $0$  to  $-1$  V. TTL-SPAD-detectors (Perkin-Elmer SPCM-AQR) deliver positive pulses of  $\sim 3$  V and must therefore be connected through a pulse inverter with attenuation (PicoQuant SIA 400). Note that this is not required when using a router. PMTs should be connected through a

preamplifier (available from PicoQuant). MCP–PMT detectors should be connected through an amplifier with slightly higher gain. Be sure to switch the high voltage supply of PMTs off and allow their electrodes to discharge. Their high voltage charge may damage the pre-amplifier. Observe the allowed input signal levels including those of the pre-amplifiers. Again, in a new experimental setup, to be absolutely sure, please check your detector pulses as well as the pre-amplifier output with a fast oscilloscope. Do not over-illuminate the detector to avoid damage. Start timing is on the leading edge, so it should be steep. Ringing and overshoot should be as small as possible. All signals should have rise times of no more than 2 ns. Slower signals will degrade timing accuracy. Very slow may not be seen by the PicoHarp at all.

If the signals are appropriate, connect the detector. Preliminary adjustments can actually be done with uncorrelated light, e.g. daylight. For safety of your detector, use strongly attenuated light, even when the detector is off. In the beginning, do not use a router to keep things simple. Start the PicoHarp software and open the Control Panel. In the Control Panel section for Channel 1, initially set the CFD Discriminator Level to 30 mV and the CFD Zero Cross level to 10 mV.

The rate meter next to the sync rate meter will now show the channel 1 input rate. Make sure that there is actually a signal coming in to the Channel 1 input (some strongly attenuated light on the detector) and try to adjust the CFD Discriminator Level. To monitor this, watch the count rate meter. The adjustment range is 0 to 800 mV for the CFD discriminator level and 0 to 20 mV for the CFD zero cross level. The initial setting of the discriminator level of 30 mV should be approximately right to suppress any electrical noise. If you lower it, you should see an increase in count rate (probably mostly noise), if you raise it you should see a reduced rate. Stay closely above the noise level. If you still see millions of counts per second at levels over 50 mV, reduce the detector illumination. Your detector may be at risk at such high rates. You can then also try out responsivity to illumination changes. The Zero Cross level is not yet meaningful. It can be adjusted for optimum timing later.

To actually collect histograms, select a measurement range large enough (determined by the chosen resolution) and a display range to cover your sync interval (i.e.  $1 / f_{\text{sync}}$ ) if possible. Initially set *Offset* = 0 and *StopAt* = 65535. Start a measurement in oscilloscope mode with e.g. 1 second acquisition time (see section 5.2 Setting up and Running Interactive Measurements for how to start a measurement). If there are counts coming in, you can try to set an upper limit of the count axis so that the histogram is scaled for best viewing.

If so far you used uncorrelated light, you may now want to move to time-correlated measurements. With a detector signal e.g. laser induced, or a 'fake' electrically derived from the laser sync, you can try to obtain a histogram that should be a narrow peak, as opposed to the flat distribution of uncorrelated light. This requires an experimental setup that delivers the signals at the two input channels with a more or less constant relative delay. For the PicoHarp in histogramming mode, this delay must be such that the Channel 1 (detector) signal comes later than the Channel 0 (sync) signal (forward start-stop mode). In addition, the delay must be chosen so that it fits in the measurement range of the histogrammer (max. 33  $\mu\text{s}$ ). To obtain such a timing it may be necessary to make adjustments by inserting cable delays of appropriate lengths in either channel, dependent on the delays in detector, laser sync or any other signal paths, electrical as well as optical.

If you have an appropriate setup installed and a measurement running, you can slightly vary all levels (including sync) for best timing response. If there is no clear optimum for any level, return to the center of its stable working range. For optimization purposes you also may want to switch to a higher resolution. If the rough setup is complete and a time-correlated signal peak is present in the histogram, the Full Width Half Maximum (FWHM) of the signal peak is displayed on-line as a figure of merit. You can also adjust your axis limits to optimize the display. Note that you can double-click all the rate meters to enlarge them as separate windows. This is useful for optical alignment work, if the PC screen is in some distance from your optical table.

## Input Troubleshooting

Whenever there is a problem, first check your cabling, detector power supply and sync source. To be absolutely sure, check the signals with a fast oscilloscope (50  $\Omega$  input!). Never try to deliver the signals to multiple 50  $\Omega$  loads in parallel, using simple T-pads. If you have a router for multiple multiple detectors, try to test your setup without the router first but note the different signal levels.

If you cannot get a stable sync rate reading of expected figures, there may be several reasons:

- there is no proper sync signal (voltage, polarity, pulse width, frequency)
- the Channel 0 CFD level setting is inappropriate
- the sync divider setting is inappropriate

Note that at small rates the meter display may fluctuate between two discrete values.

Reasons for zero Channel 1 counts are:

- no or inappropriate signal (voltage, polarity, pulse width, frequency)
- inappropriate CFD levels
- preamplifier or detector failure
- router connected but not used (bypassed)

Note that all meters are updated at the display refresh rate (it can be set between 0.1 and 1 s for integration mode and it equals the acquisition time in oscilloscope mode). In oscilloscope mode, if the time is chosen very long, this may lead to apparently “nothing happening” for accordingly long times. The refresh rate for Integration mode can be selected in the *Software Settings* Dialog. This setting does not equally apply in oscilloscope mode. If you see no counts, make sure you are not running oscilloscope mode with acquisition times longer than a few seconds. Once Channel 1 count rates > 0 are being displayed you should then also see counts appearing in the histogramming rate meter, provided your measurement range is large enough. Make sure the offset is set to 0. If there are histogramming counts but no histogram is building up on the screen, check the time and count axis bounds. It is best to start with a wide display range and then narrow it down. Similarly, a logarithmic count axis setting is the safest way to start, because you see even small histograms. If your FWHM figures are wider than expected, make sure the histograms are not overflowing (peak count 65535). A shorter acquisition time and/or the setting “*stop on overflow*” helps to avoid this. Do not set the “*stop at*” level to less than 65536 unless you have experimental reason to limit the count height.

In the process of set-up you should observe the warning icon that may appear at the bottom right of the main window. When the PicoHarp software is running with functional hardware it continuously collects information about the input signals and the current acquisition settings. If these settings together with the input rates indicate possible errors, the software will activate the warning icon.



The warning icon can be clicked to display a list of current warnings together with a brief explanation of each warning (see also section 8.1). Note that the software can detect only a small subset of possible error conditions. It is therefore not safe to assume “all is right” just by seeing no warning. On the opposite, if any of the warnings turns out to be an unnecessary nuisance, e.g. because your specific measurement conditions will expectedly cause it, you can disable that warning via the software settings dialog available through the toolbar (see section 6.2).

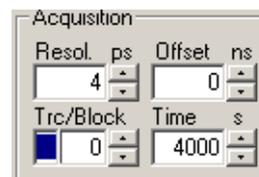
## 5.2. Setting up and Running Interactive Measurements

If you are running TCSPC measurements for the first time we strongly recommend you read our TechNote on TCSPC first (see your distribution CD). This will help you getting started much quicker. This manual cannot replace a textbook.

To set up measurement parameters use the PicoHarp control panel. The control panel can be opened by clicking the control panel button on the toolbar or by pressing <ALT>+C.



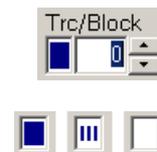
In the control panel section 'Acquisition' you can set the resolution (time per bin), the offset, the measurement time, and the block of memory to use for this measurement. To begin, use a measurement time of 1 second and an offset of 0. There are always 65536 time channels (bins) per PicoHarp histogram. Histograms can be measured in 512 memory blocks. Out of these up to 8 curves can be displayed and one 'active curve' can be used for a measurement. You can designate the active memory block you wish to use for the next measurement by selecting the block number in the control panel. Never forget to select a new block when collecting new data and old data is intended to be preserved.



Use the trace mapping dialog to select up to 8 curves for display. Make sure the memory block you measure into is mapped to a display trace that is switched on, so that you can see the trace. You can reach the trace mapping dialog from the toolbar or by clicking on the trace colour indicator next to the block selector.



The colour indicator shows the trace colour the chosen block is currently mapped to. If it is a solid square, the curve is mapped and shown. If it is mapped but not shown, the indicator shows a small striped square. If the curve is not even mapped for display the indicator remains white.



To run a measurement you will need to set up the appropriate CFD levels as outlined in section 5.1 Setting up the Input Channels

There are two basic histogramming modes for interactive measurements: Oscilloscope and Integration mode. Oscilloscope mode repeatedly collects histograms with a fixed measurement time and displays them on the screen. This lets you see fast changes in the histogram, e.g. for optical setup and adjustments. Usually this only makes sense with relatively strong signals and short acquisition times. Integration mode is usually operated with longer acquisition times. In this case the histogram continues to grow over a longer time and the display is updated at regular intervals, so that the accumulation process can be observed.



To start a measurement with the current control panel settings, use the *start* button (GO) on the toolbar or press <ALT>+G.



To stop a measurement use the *stop* button on the toolbar or press <ALT>+S.

Note that a measurement may automatically stop and/or restart, dependent on the current settings of 'stop on overflow' and 'restart' in the control panel.



Allow a warming-up period of 20 to 30 minutes dependent on lab temperature before using the PicoHarp for final measurements. You can use this time for set-up and preliminary measurements. Click the calibrate button after warming up.



The rate meters (bottom of main window) permit visual control of the data acquisition. They are often too small to view from further away, e.g. when adjusting the optical setup. You can in this case simply double-click the rate meter of interest. This opens a large meter display that you can then size and move on the screen as you like.

Once you have established standard settings for your experimental setup you may want to save them to a file. The control panel settings can then be recalled at any time. The program stores all settings together with the histogram data of those curves in memory (max. 512) that have been filled by a measurement. In addition to this, all settings are memorized in the Windows registry, so that at program startup you find the control panel as it was when you last closed the PicoHarp software.

## 5.3. Routed Measurements

### 5.3.1. General Setup and Operation

Classic TCSPC measurements are performed with one photon detector only. However, it is often desirable to have multiple detector channels in parallel, without having to install multiple TCSPC electronics. This can be achieved by detector routing into one set of TCSPC electronics.

The PicoHarp 300 (but not the low cost model PicoHarp 300E) supports routing of up to four detectors. Detectors with TTL output (usually Perkin Elmer SPCM–AQR) can be routed through the router module PHR 402. Up to four SPAD detectors with NIM output (usually PDM series from Micro Photon Devices, Italy) can be routed through the PHR 403 router. For ultimate flexibility the universal router PHR 800 can be used (PicoHarp hardware version 2.0 only). It supports not only TTL and NIM but also any custom signal levels and polarities. It can also be fitted with CFDs for use with PMT or MCP detectors. All routers must be ordered as add-on options.

The routing principle of the PHR routers is 'signal driven multiplexing'. It is also referred to as 'statistical time sharing'<sup>1</sup>. I.e. the four router channels share one timing circuit on a 'first come first serve' principle. This works fine because the probability of receiving a photon in one sync period must be kept on the order of 1 or 2 % anyway, which is a prerequisite for avoiding pile-up in TCSPC. If two photons arrive during the same sync period, only the first one will be accepted. The router recognizes in which channel it received a photon and delivers that information to the PicoHarp via a dedicated router/control port. The next photon can be on any other channel, whichever comes first. Because of the inherent count rate limitations in TCSPC this method of multiplexing can in fact increase the overall counting efficiency. This is because pile-up effects do not occur across channels, and the overall count rate can thereby be larger than the permitted 1 to 2 % per channel. Even with mutually inflicted dead time losses the counting efficiency of each channel remains close to 1 and the efficiency over all channels is substantially larger than that of a single channel system. The only limitation multiplexing/routing implies is that the total count rate the time digitizer can handle must be shared by the four channels.

Because of the single photon conditions there are consequently low probabilities of photons impinging on two or more detectors simultaneously. The router uses a time window of some ns during which a decision is made as to whether a count is valid. This is the case only if there was a signal from a single detector. In the case of multiple simultaneous signals, the events will be discarded. This mechanism ensures a cross talk between channels of typically < 0.01 % at 500,000 counts/sec. In order to keep the crosstalk low it is recommended not to exceed 500,000 counts/sec per channel.

Because of the multiplexing nature of the routing mechanism and the TCSPC dead time it is generally not possible to perform short distance timing between individual routing channels, e.g. for coincidence correlation experiments. However, cross correlations on the time tags (100 ns resolution) of individual channels can be performed very elegantly from a single data file. This is of great value in Fluorescence Correlation Spectroscopy (FCS).

Using a router does not normally change the time resolution or the IRF width, although at high count rates some degradation may be observed. Also, due to dead time, the count rates recorded on individual channels will drop, as the count rates on other channels increase to very high numbers. Therefore, routing must be used with extra care in regard to single photon statistics. This is usually not an issue at kHz count rates e.g. in single molecule spectroscopy.

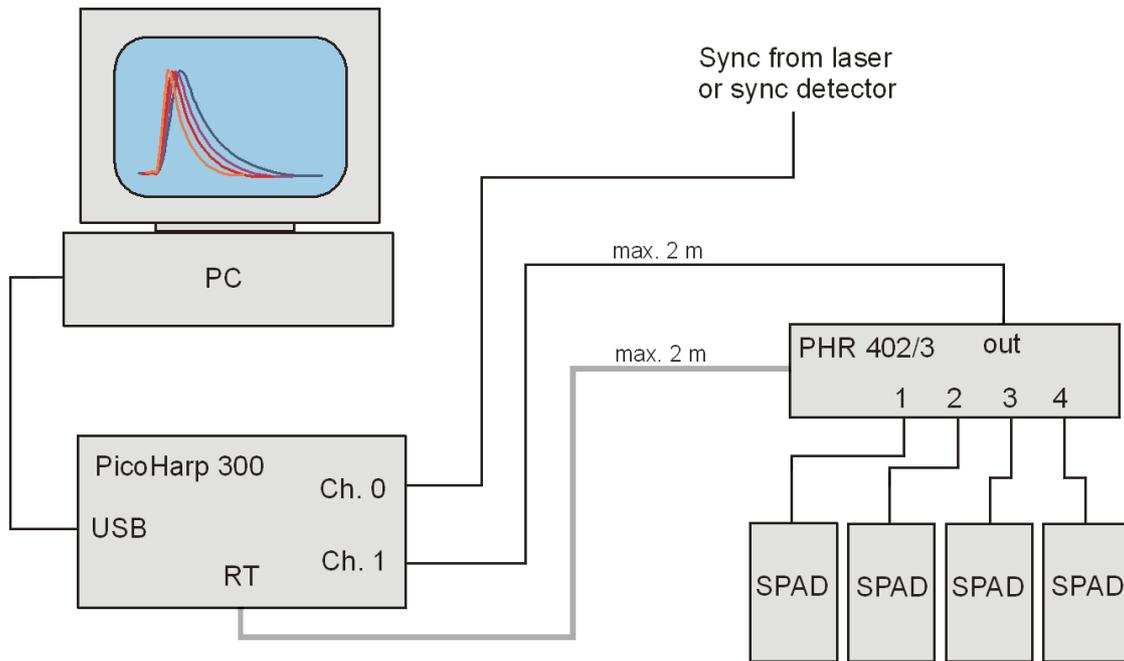
In a routed setup the detectors are each directly connected to one router input via 50  $\Omega$  SMA cables (no attenuator). The router output is directly connected to the PicoHarp input channel 1 (no attenuator). The routers must also be connected to the PicoHarp 300 via a dedicated cable for power supply and control purposes (connector RT/CTRL). Both cables must but be of approximately equal length (max. 2 m).

The PHR 40x routers must be connected to the 'RT' connector through a 15 pin high density Sub-D cable. The PHR 800 can only be used with a PicoHarp model 2.0 and must be connected to the 'CTRL' connector through a dedicated cable (DVI-I).

If you wish to use routing together with other devices (marker input etc.), the other (free) control port can be used (see section 8.3.2 on Router / Control Port Connectors). Alternatively, a suitable Y-cable for the simultaneous connection to the RT port can be used (available from PicoQuant). Note that this cable must not be extended. Only required marker signals may be extended to connect the marker source. The PHR 800 provides its own inputs for marker signals.

The following figure shows the connections required in a routing setup with four SPAD detectors and the PHR 402/403. For the PHR 800 the setup is nearly the same, only that the CTRL port is used.

<sup>1</sup> Birch, Holmes, Imhof, Nadolski and Cooper, "Multiplexed time-correlated single photon counting", SPIE Vol. 909 Time-Resolved Laser Spectroscopy in BioChemistry (1988)



The Router will show its power state via a green LED and its trigger state (signals at any input) via a yellow LED. If the router is connected, the PicoHarp software will detect its presence at start time. In this case the 'Routed' check box  Routed in the control panel will be active (otherwise disabled and grey:  Routed ). When using a PHR router, the PicoHarp's CFD level in the control panel should be set between 50 and 350 mV. The precise figures should be optimized experimentally.

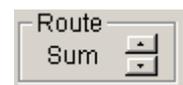
The PHR 40x routers are configured for fixed input signals (TTL or NIM, respectively). The PHR 800 allows adjustment of input level and polarity. See the corresponding section 5.3.2 on Using the PHR 800 Router further below.

If a router is correctly connected you can switch to routed data acquisition by ticking the  Routed check box. In routed measurements the counts from each detector are collected in different blocks of on-board memory, thereby forming one histogram for each detector. This can be used e.g. to collect different polarization directions or colours at the same time. Note that the router acts as a multiplexer, so that the operation is only 'quasi' parallel. In particular, the four channels must share the channel 1 input of the PicoHarp 300, so that the total count rate cannot exceed that permitted by the PicoHarp channel.

When routing is enabled  Routed, the collection of histograms will operate on four consecutive blocks of memory simultaneously. Recall that there are 512 blocks of memory in total. In routed operation, the block number shown in the control panel indicates the first block of four belonging to one measurement. This is why in routed operation you can only change the block number in steps of four. In order to see the data being collected in all four histograms, you need to make sure the active block in the control panel and the next three blocks are mapped to display curves and switched on ('show') in the trace mapping dialog.



The rate meters at the bottom of the main PicoHarp window also have a slightly different meaning when routing is enabled. Only the input rate meter is as usual: it shows the count rate at the PicoHarp's channel 1 input, which is common for all routing channels. For the other meters there is a selector switch on the very right of the rate meters. It allows to assign the rate meters to either the sum of all routing channels or to one single routing channel (click the up/down arrows to make the selection).



When you open the *Trace Mapping Dialog* you will notice that after a routed measurement four curves have been filled with data and their details view button has become active. The details shown in this dialog correspond to the information stored in the "curve header" of a \*.phd file. If a router was used in the measurement, the router model (PHR xxx) will be shown in the details panel.



If you take measurements with the router connected but routing disabled, the counts from all detectors will go into the active block selected in the control panel. Note that in this case the router settings in the curve details display are partially marked 'n.a.' because they are no longer directly mappable to one input.

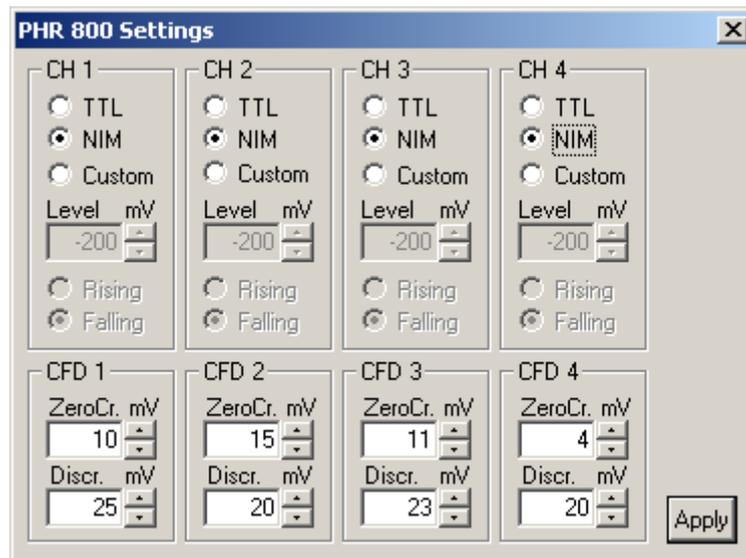
TTR mode measurements (see section 5.4 on Time-Tagged Mode Measurements) always implicitly provide routing information if a router is used. The routing information from the PHR 40x is represented by two bits in each TTR record, allowing the distinction of up to four routing channels. Routing does not affect TTR throughput in any way other than given by the limits in total count rate.

### 5.3.2. Using the PHR 800 Router

The PHR 800 allows very fine control of its input settings. For this purpose, the PicoHarp software provides a dedicated dialog. It can be launched through the *PHR 800* button on the Toolbar. The button will remain disabled (grey) if no PHR 800 was detected at startup.



The control dialog for the PHR 800 is shown below. It has four sections for the input configuration and four sections for CFDs. The router is modular and can be purchased without the full set of CFDs. In this case the corresponding dialog sections are not accessible (greyed out).



The input channel sections provide radio buttons for the input type, the level and the active edge. For TTL and NIM the input levels and edges are fixed. For custom input types they can be adjusted as required. Note that it is important to select valid combinations of level and edge. Otherwise (especially if no detector is connected) the input can be locked in a forbidden state, where all other channels will also be locked. Please use the standard settings for TTL and NIM unless you really need custom settings.

If CFDs are used, they must be physically connected at the front panel of the PHR 800. The software does not recognize if and how such connections exist. Since the software layout suggests a direct match of channel number and CFD number, it is strongly recommended to connect the physical devices accordingly. All other situations would very likely lead to confusion. When CFDs are used, the corresponding input channels must be set for NIM.

The CFDs are identical to those of the PicoHarp's own built in CFDs. Their operation therefore follows the same guidelines and conventions. The adjustment ranges are also the same. Note that due to the polarity convention used in the PicoHarp's control panel, all CFD settings are shown without sign, although the actual electrical signals are negative. This is different from the router input level setting, where both positive and negative signals may occur, so that they must be shown accordingly, including the sign.

The router settings are kept in the registry, so that at the next start of the software they will still be there. Similarly, they are kept in the measurement files, so that they can be retrieved that way. The router settings for each measured curve are also kept in the data files. In this case they are only for the routing channel used during the recording of that curve. They can be viewed through the *Trace Mapping* dialog. Note that the router settings are ambiguous if a router was used but not enabled (routing off). The router settings are then marked as 'n.a.' in the curve details view.

## 5.4. Time-Tagged Mode Measurements

Time-Tagged–Time-Resolved (TTTR) mode allows the recording of individual count events directly to hard disk without on-board forming of histograms. It originates from the TimeHarp TCSPC systems and has been further improved for the PicoHarp 300.

In TTTR mode, in addition to the start-stop timing with picosecond resolution, the timing of the events with respect to the beginning of the whole measurement is recorded in the event records. This mode is particularly interesting where the dynamics in a fluorescence process are to be investigated. The availability of the time-tags permits photon burst identification, which is of great value e.g. for Single Molecule Spectroscopy (SMS) in a liquid flow. Other typical applications are Fluorescence Correlation Spectroscopy (FCS) and Burst Integrated Fluorescence Lifetime (BIFL) measurements. Together with an appropriate scan controller, TTTR mode is also suitable for ultra fast Fluorescence Lifetime Imaging (FLIM). Applications beyond fluorescence spectroscopy are e.g. in quantum information processing, notably quantum key distribution and related basic research.

The PicoHarp 300 actually supports two different Time-Tagging modes, T2 and T3 mode, which will be explained further below. When referring to both modes together we use the general term TTTR here.

TTTR mode measurements always implicitly provide routing information if a router is used. Routing information is represented by four bits in each TTTR record, that allow up to four routing channels to be distinguished. However, these bits are only meaningful if a router was actually used for the measurement. Routing will probably not be useful in T2 mode, where the main objective is typically coincidence correlation. With all current routers, only two routing bits are used (4 channels). Throughput constraints and crosstalk considerations apply the same way as in interactive routed measurements.

Note: TTTR mode is not part of the standard PicoHarp product. It must be purchased as a separate firmware option that gets burned into the ROM of the PicoHarp. An upgrade is possible at any time. It always includes both T2 and T3 mode.

### System Requirements

In cases where the Time-Tagging modes are to be used with high continuous count rates (> 500 kcps) the PC system must meet some special performance criteria, that usually are insignificant in standard interactive use of the PicoHarp. The reason for this is the relatively large amount of data being generated in TTTR mode. In order to prevent an overflow in the recording, this data must be transferred to the hard disk in real-time. This requires a modern PC with a fast I/O subsystem. A CPU running at least at 1 GHz is recommended. For sufficient performance in TTTR mode a modern IDE or SATA hard disk with 7200 rpm and 8 MB cache is a good and economical choice (typically 160 GB and more). Also suitable are modern SCSI hard disks with PCIe or on-board controller. Another option for maximum throughput are RAID systems which, however, must be carefully chosen and configured. At the same time, the USB subsystem must be configured correctly. In order to deliver maximum throughput, the PicoHarp 300 uses state-of-the-art USB 2.0 bulk transfers. This is why the PicoHarp must rely on having a working USB 2.0 host interface. USB 2.0 host controllers of modern PCs are usually integrated on the mainboard. For older PCs they may be upgraded as PCI cards. Throughput is usually limited by the host controller and operating system, not the PicoHarp. Make sure that the host controller drivers are correctly installed, otherwise they may be running only in USB 1.0 mode. This may require using the mainboard manufacturer's driver disk rather than relying on Windows drivers alone. Do not run other bandwidth demanding devices on the same USB interface when working with the PicoHarp.

### T2 Mode

In T2 mode both signal inputs of the PicoHarp 300 are functionally identical. There is no dedication of input channel 0 to a sync signal. Usually both inputs are used to connect photon detectors. The events from both channels are recorded independently and treated equally. In each case an event record is generated that contains information about the channel it came from and the arrival time of the event with respect to the overall measurement start. The timing is recorded with 4 ps resolution. Each T2 mode event record consists of 32 bits. There are 4 bits for the channel number and 28 bits for the time-tag. If the time tag overflows, a special overflow marker record is inserted in the data stream, so that upon processing of the data stream a theoretically infinite time span can be recovered at full resolution. Only for practical reasons of the current software design the number of records is limited to 2147483648, which corresponds to 8 GBytes of data.

Dead times exist only within each channel (90 ns typ.) but not across the channels. Therefore, cross correlations can be calculated down to zero lag time. This allows powerful new application such as FCS with lag times from picoseconds to hours. Autocorrelations can also be calculated at the full resolution but of course only starting from lag times larger than the dead time.

The 32 bit event records are queued in a FIFO (First In First Out) buffer capable of holding up to 256 k event records. The FIFO input is fast enough to accept records at the full speed of the time digitizers (up to 10 Mcps each). This means, even during a fast burst no events will be dropped except those lost in the dead time anyhow. The FIFO output is continuously read by the host PC, thereby making room for fresh incoming events. Even if the average read rate of the host PC is limited, bursts with much higher rate can be recorded for some time. Only if the average count rate over a long period of time exceeds the readout speed of the PC, a FIFO overrun could occur. In case of a FIFO overrun the measurement must be aborted because data integrity cannot be maintained. However, on a modern and well configured PC a sustained average count rate of 5 Mcps is possible. This total transfer rate must be shared by the two input channels. For all practically relevant fluorescence detection applications the effective rate per channel is more than sufficient.

For maximum throughput, T2 mode data streams are normally written directly to disk. There is no immediate data visualization during a plain T2 mode measurement, except count rate and progress display. However, for the special case of Fluorescence Correlation Spectroscopy, there is a real-time correlator for T2 data. Using custom software it is also possible to analyze incoming data "on the fly". Even on-line correlation can be implemented. Obviously this requires efficient processing and possible restrictions in average count rate. The PicoHarp software installation CD contains demo programs to show how T2 mode files can be read by custom software. The implementation of custom measurement programs requires the PicoHarp programming library, which is available as a separate option.

### T3 Mode

In T3 mode channel 0 is dedicated to a sync signal. As far as the experimental setup is concerned, this is similar to histogramming mode. The main objective is to allow high sync rates (up to 84 MHz) which could not be handled in T2 mode. Accommodating the high sync rates in T3 mode is achieved as follows: First, the sync divider is employed as in histogramming mode. This reduces the sync rate so that the channel dead time is no longer a problem. The remaining problem is now that even with a divider of 8, the sync rate is still too high for collecting all individual sync events like ordinary T2 mode events. Considering that sync events are not of primary interest, the solution is to record them only if they arrive in the context of a photon event on channel 1. This is the same as the original TTTR mode concept as it was first conceived for the TimeHarp system. The event record is then composed of two timing figures:

1. The start-stop timing difference between the photon event and the last sync event, and
2. the arrival time of the event pair on the overall experiment time scale (the time tag).

The latter was originally obtained from an independent asynchronous clock. This made it difficult to combine the start-stop timing with the time tag or to know the sync period the event belonged to. For the PicoHarp's T3 mode a smarter approach was chosen: The time tag is now obtained by simply counting sync pulses. From the T3 mode event records it is therefore possible to precisely determine which sync period a photon event belongs to. Since the sync period is also known precisely, this furthermore allows to reconstruct the arrival time of the photon with respect to the overall experiment time.

Each T3 mode event record consists of 32 bits. There are 4 bits for the channel number, 12 bits for the start-stop time and 16 bits for the sync counter. If the counter overflows, a special overflow marker record is inserted in the data stream, so that upon processing of the data stream a theoretically infinite time span can be recovered. Only for practical reasons of the current software design the number of records is limited to 2147483648, which corresponds to 8 GBytes of data. The 12 bits for the start-stop time difference cover a time span of  $4096 \cdot R$  where  $R$  is the chosen resolution. At the highest possible resolution of 4 ps this results in a span of 16 ns. If the time difference between photon and the last sync event is larger, the photon event cannot be recorded. This is the same as in histogramming mode, where the number of bins is also finite. However, by choosing a suitable sync rate and a compatible resolution  $R$ , it should be possible to reasonably accommodate all relevant experiment scenarios.  $R$  can be chosen in doubling steps between 4 ps and 512 ps.

Dead time in T3 mode is the same as in the other modes (90 ns typ.) Since there is only one photon channel, only autocorrelations can be calculated (unless a router is used). This can be done at the resolution  $R$  but of course only starting from lag times larger than the dead time.

The 32 bit event records are queued in a FIFO (First In First Out) buffer capable of holding up to 256 k event records. The FIFO input is fast enough to accept records at the full speed of the time-to-digital converters (up to 10 Mcps each). This means, even during a fast burst no events will be dropped except those lost in the dead time anyhow. The FIFO output is continuously read by the host PC, thereby making room for fresh incoming events. Even if the average read rate of the host PC is limited, bursts with much higher rate can be recorded for some time. Only if the average count rate over a long period of time exceeds the readout speed of the PC, a FIFO overrun could occur. In case of a FIFO overrun the measurement must be aborted because data integrity cannot be maintained. However, on a modern and well configured PC a sustained average count rate of 5 Mcps is possible. This total transfer rate is fully available for input channel 1. For all practically relevant fluorescence detection applications this is more than sufficient.

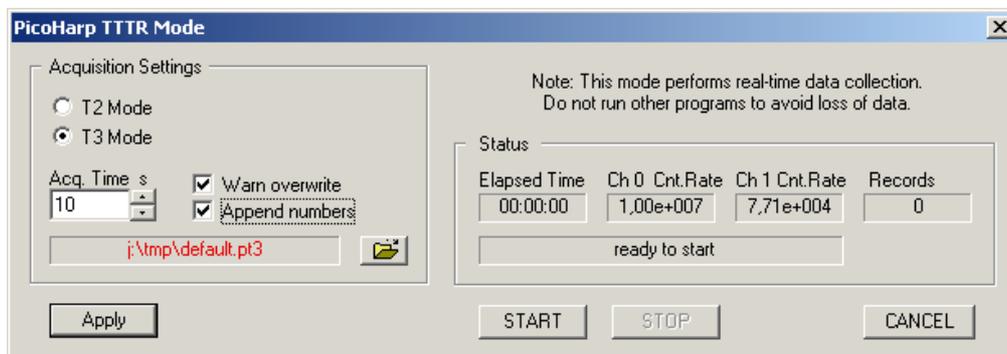
For maximum throughput, T3 mode data streams are normally written directly to disk. There is no immediate data visualization during a plain T3 mode measurement, except count rate and progress display. However, for the special case of Fluorescence Correlation Spectroscopy, there is a real-time correlator for T3 data. Using custom software it is of course possible to implement other algorithms for the analysis of incoming data “on the fly”. Obviously, any real-time analysis requires efficient processing and possible restrictions in average count rate. The PicoHarp software installation CD contains demo programs showing how T3 mode files can be read by custom software. The implementation of custom measurement programs requires the PicoHarp programming library, which is available as a separate option.

## Running a plain TTTR Mode Measurement

A TTTR mode measurement (T2 or T3 mode) will typically be started after all control panel settings have been tested in normal interactive histogramming mode (oscilloscope or integration). The acquisition time (measurement time) and the file for saving the data are the only parameters that can be set separately.

A typical approach to set up a TTTR mode measurement would be by first starting oscilloscope mode with an acquisition time of e.g. 1 second. Then all control panel settings should be optimized to reliably obtain the expected data.

Once all settings are satisfactory, click the *TTTR Mode Button*  on the toolbar. This will bring up the *TTTR Mode Dialog*.



The dialog section *Acquisition Settings* is for the selection of the measurement mode (T2/T3), an overall acquisition time and a file name. Note that switching between T2 and T3 mode takes some time because the hardware must be reconfigured. Normally such a switching should not occur often because the two modes usually require a very different experiment setup.

The section *Acquisition Settings* also has two tick boxes for the handling of existing files. You can turn on a warning and/or automatically have numbers appended to the file name, so that you can conveniently perform series of measurements. The file name is shown in red if the file already exists. The button with the file icon will open a standard Windows file dialog. You can select an existing file or choose a new name. The PicoHarp TTTR mode files have the extension “.pt2” or “.pt3” respectively. For maximum count rate throughput you should choose a file destination on a fast local hard disk as outlined above. Network drives are often too slow.

The dialog section *Status* shows elapsed time, the count rates and the number of collected records. Note that this is limited to 2147483648 records (8 GBytes of data). Below these figures there is a status line showing what is currently happening. Further below there are buttons for *Start*, *Stop* and *Cancel*. *Start* and *Stop* control the actual TTTR measurement run. *Cancel* is for leaving the TTTR mode dialog.

When the TTTR mode dialog is running it continuously collects information about the input signals and the current acquisition settings. If these settings together with the input rates indicate possible errors, the software will activate the warning icon. The warning icon can be clicked to display a list of current warnings together with a brief explanation of each warning (see also section 8.1). In order to fix wrong settings you may have to close the TTTR mode dialog and return to the control panel.



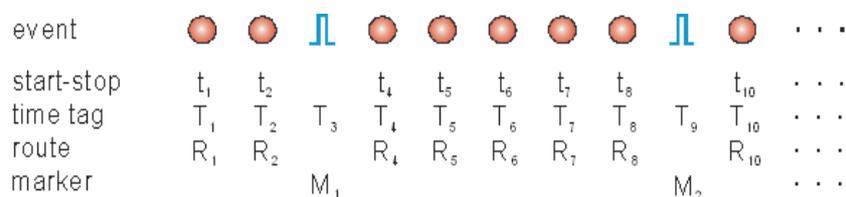
The TTTR mode measurement will start as soon as you click the *Start* button. You will then be able to watch the progress of your measurement in the status boxes. The leftmost status box shows the elapsed measurement time. The two boxes in the center show the count rates. The rightmost box will show the number of events that have been recorded. Note that this includes overflow and marker records. Therefore, the number of records will be somewhat larger than the true photon counts. The overflow and marker records will be removed by the data analysis / processing software.

A measurement can be stopped at any time by clicking the *Stop* button (the middle button). The data recorded up to this point will be in the file. When the measurement has completed, the *Stop* button will change to grey (disabled). Use the *Cancel* button to return to the normal interactive mode. Again, this will take some time for hardware reconfiguration.

### External Markers<sup>1</sup>

Often it is desirable to synchronize TCSPC measurements with other information or processes of complex measurement tasks. In order to perform e.g. Fluorescence Lifetime Imaging, the spatial origin of the photons must be recorded as well. For this purpose one needs a mechanism to assign external synchronization information to the TCSPC data. For the special case of Fluorescence Lifetime Imaging, conventional systems use large arrays of on-board memory and switch to new blocks of memory upon arrival of e.g. a pixel clock pulse. To accommodate the large amount of data generated due to the 3-dimensional matrix of pixel co-ordinates and lifetime histogram channels causes serious problems. Even with modern memory chips, this approach is limited in image size and/or number. In addition it is expensive, and implies loss of information about the individual photon arrival times. To solve the problem much more elegantly, the TTTR data stream generated by the PicoHarp 300 can contain markers for synchronization information derived from an imaging device, e.g. a scan controller. For this purpose the control ports of the PicoHarp 300 provide three TTL inputs for synchronization signals (in addition to those for routing). Note that the low cost model PicoHarp 300E does not have this feature.

The figure below illustrates how the external marker signals are recorded in the data stream.



Bullets denote a photon, blue pulses denote a marker signal. The external markers are treated almost as if they were regular photon event records. A special channel code allows to distinguish true photon records from the marker records. Software reading the TTTR file can thereby filter out the markers, e.g. for line and frame clock in imaging applications. This makes possible to reconstruct the 2D image from the stream of TTTR records, since the relevant XY position of e.g. a scanner can be determined during the data analysis. The data generated is nearly free of redundancy and can therefore be transferred in real-time. The image size is unlimited both in XY and in count depth. Since there are up to three such synchronization signals, all imaging applications can be implemented and even other experiment control signals can be recorded. This marker scheme is a very special feature of the PicoQuant TCSPC electronics. To date, the PicoQuant TimeHarp and PicoHarp are the only TCSPC systems that supports this powerful measurement mode. It may be worth noting that this technology enabled PicoQuant GmbH to develop the leading edge MicroTime 200 Fluorescence Lifetime Microscope.

Section 8.3.2 contains the detailed pin assignments of the PicoHarp's router/control ports (two sub-D connectors labeled 'RT' and 'CTRL'). Here we focus only on the signals of relevance for TTTR mode and markers:

Three TTL compatible inputs accept the synchronization signals that will be recorded as markers. The active edges of these signals can be chosen in the software settings dialog available through the Toolbar. Both high and low state must be at least 100 ns long. The period may therefore (in principle) be as short as 200 ns but data bus throughput constraints will apply. Each marker creates an additional TTTR record, so that one must ensure not to swamp the data stream with too many marker records. Markers also take precedence over photon records, so that excess marker traffic can suppress photon records. In fast imaging applications it is therefore recommended not to use a pixel clock but a line clock only. Because each photon has a time tag it is usually not necessary to use an additional pixel clock.

There is another TTL input that can be used to externally enable/disable the recording of incoming counts (see section 8.3.2). Counting is enabled when this input is held high. This mechanism can be used to blank out periods where e.g. an external scanner is moving to the next position. The count enable signal is also active in interactive mode. It may be used to synchronize the start of acquisition with external events. If external disabling is not desired, the pin must be held high or left open. Note that if a router is connected, this pin is not available for user signals.

<sup>1</sup> not available for PicoHarp 300E

A TTL output 'ctc active' is provided to signal the start of a measurement. This can be used to notify or trigger other hardware. If you wish to use markers, the marker signals must be fed in via the 'CTRL' or 'RT' port (see section 8.3.2). If routing is used together with markers, the free port can be used for feeding in markers.

If necessary, a dedicated Y-cable for the simultaneous connection to the 'RT' port can be used (available from PicoQuant). Note that this cable must not be extended. Only required marker signals may be extended to connect the marker source. The PHR 800 has its own marker inputs, so that simultaneous connection to the 'CTRL' port of a PicoHarp model '2.0' should never be necessary.

### Using TTTR Mode Data Files

For diagnostic purposes you may re-load a T3 mode file into the PicoHarp software. The limitation is that you will only be able to form a histogram over the start-stop times in your T3 mode data. The time-tag information will not be used here. The PicoHarp software will recognize that you are loading a T3 mode file and how many records there are contained in it. It will then prompt you for a range to use for histogramming. Make sure to keep this range limited to a reasonable size, since the histogram channels are limited, so overflow may occur. The histogram will go to block 0 of the histogram memory. A TTTR mode file also contains all control panel settings that were active at the time of the measurement run. However, it does not contain any count rate information or pulse FWHM. This is why the respective meters will show zero after loading such a file. After loading a TTTR mode file, you will find the document title reading "*Histogram from...*". If you choose to save such data you will have to give it a new file name (\*.phd). This is because now a histogram has been formed, and saving it to the same file name would destroy the original TTTR mode file. However, you may save the previously formed histogram as if it were obtained in normal interactive mode, to a standard PicoHarp data file (\*.phd).

Reloading T3 mode files serves as a quick diagnostic tool only. For T2 mode files such a feature is not available. Further processing or analysis of TTTR mode data must therefore be performed through external data analysis software. Such software is available from PicoQuant for a wide range of analysis tasks (SymPho-Time). Further specialized analysis can be performed by dedicated custom software. If you wish to save the cost for the commercial TTTR analysis software or if you require special analysis algorithms you may want to program your own analysis software. For development of your own custom programs, please refer to the demo code provided. Demo source code is contained on your PicoHarp installation disks and will be installed by the software setup. Also see section 8.1 for the file format specifications. The paragraph below can only give an outline.

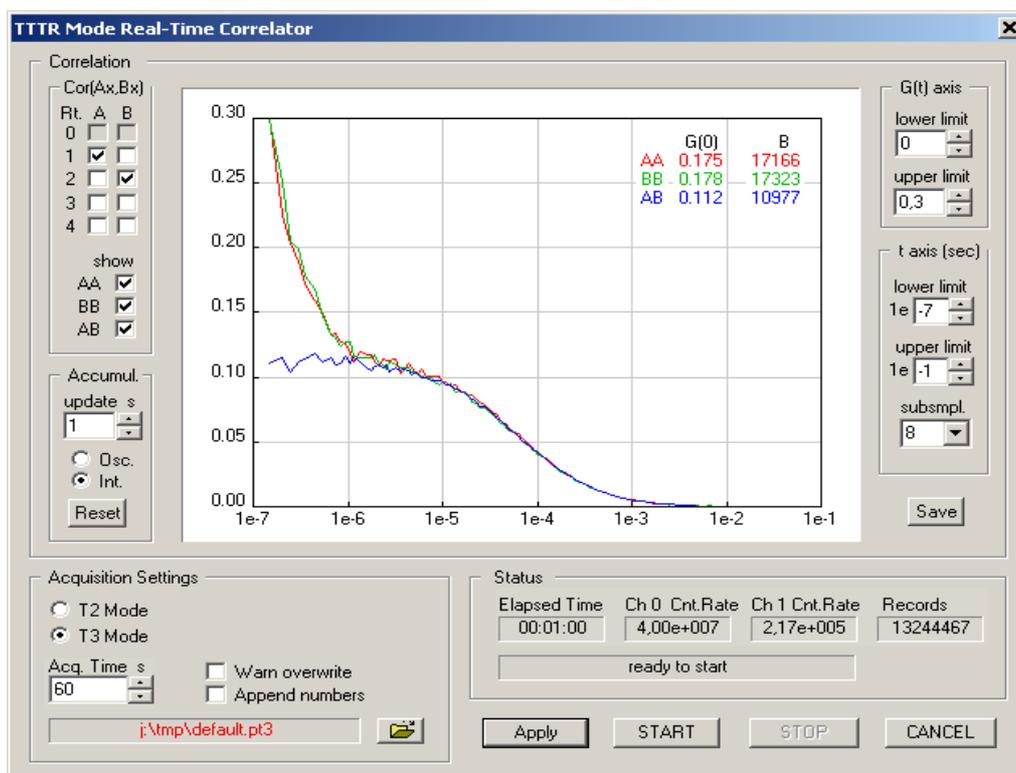
The first part of a TTTR mode file is a header with the basic setup information, similar to that of the other modes. What follows after the header is a sequence of 32 bit TTTR records. The TTTR records in the file consist of different pieces of information in groups of bits. These pieces of information must be extracted by a bit masking/shifting operation. Their specific layout is different for T2 and T3 mode. In both cases, in addition to extracting the bit fields, the software must step through the whole file and interpret the overflow records and correct the overall time axis accordingly. Details on how this can be done should be looked up in the demo programs that are installed as part of the software distribution.

## 5.5. TTTR Measurements with Real-Time Correlation

An important application of the PicoHarp and its TTTR mode is Fluorescence Correlation Spectroscopy. While TTTR data collection and off-line software correlation has been used for quite some time, PCs have only recently become fast enough to calculate the correlation function “on the fly” while the data is being collected. This capability can be very useful in setting up and monitoring of FCS experiments. The PicoHarp software provides such a real-time software correlator for T2 and T3 mode. It is available if the TTTR mode option has been purchased.

Although it collects the same T2 or T3 mode files, the real-time correlator has its own button on the Toolbar. 

The next figure shows the TTTR mode correlator dialog. Its lower section provides the same file handling and progress indicators elements as in plain TTTR mode. Please refer to section 5.4 for details. The upper section of the dialog contains the correlator display and various control elements.



The dialog section *Cor(Ax,Bx)* allows to select individual routing channels for the correlation. A and B are the virtual input channels to be correlated. Provided that a router is installed, all routing channels that are ticked in columns A resp. B will be used in the corresponding correlation channel. If no router is installed, only Rt 1 is meaningful. If your selection is incomplete, the heading of the section *Cor(Ax,Bx)* will turn red to indicate the problem. If you proceed anyway you will get error messages upon start.

The correlator always calculates the autocorrelations of the two virtual channels (AA and BB) as well as the cross-correlation (AB). There are three tick boxes that select which of these are shown.

The dialog section *Accumul.* allows to select an update time for the correlation display. Furthermore it allows to select between repetitive updates (*Osc.*) and cumulative collection (*Int.*). If the latter is chosen one can manually reset the correlator by clicking the *Reset* button. Note that all these settings affect only the real-time correlator. The raw T2 or T3 mode data will be collected continuously and completely independent from the correlator settings.

The *Save* button under the *Accumul.* section allows to save the correlator results as they were last shown. The saved result is an ASCII file with some header information. The format is self-explanatory. Note that the real-time correlator is primarily a tool for preview during measurement setup and optimization. It does not allow to correlate data from TTTR files. If you need the correlation over the entire stream of collected data you need to make sure you run in Integrating mode and do not press reset during the measurement. If you need the correlation results you should save them here. If you need to perform further or more thorough FCS analysis on the collected data you can use the SymphoTime software from PicoQuant.

On the right hand side there are controls for the axis ranges of the correlator display. Note that they affect only the display. Collected data is always complete, independent from the axis settings. In contrast, the selection box *subsmpl* has an effect on the correlator results. It determines how many tau points are calculated. The correlator works in a logarithmic multiple tau scheme and *subsmpl* specifies the number of linear subsamples in each log stage. A higher number of subsamples increases the resolution of the correlation curve. Calculating more points is more time consuming and therefore it may lead to lower count rate limits that can be handled. In this case you may get FIFO overruns. The default of *subsmpl* = 8 is a reasonable tradeoff between speed and resolution. On faster computers the setting of *subsmpl* = 16 is a good choice.

Note that the starting point and spacing of the Tau sampling points also depends on the time tag resolution. In T3 mode this corresponds to the sync period. In T2 mode the native resolution of 4 ps is binned down to a time tag resolution of 25 ns in order to make the data manageable for the real-time correlation algorithm.

The correlation curve display can be shown with a grid (see software settings dialog, accessible from the toolbar of the main window). It also shows two useful figures obtained from the collected data (top right of curve window). The first figure is an approximation of  $G(0)$ . In classic FCS experiments this corresponds to the inverse of the number of particles in the focal volume. It is continuously updated together with the curve display, which can be useful for system adjustment, notably when using the repetitive accumulation mode (Osc.). Note that the approximation is a simple averaging over the Tau points between 0.5 and 10  $\mu\text{s}$ . The reason for not starting at 0 is that detector afterpulsing is often creating artefacts below 0.5  $\mu\text{s}$ . The figure B is an indicator for molecular brightness. It is also updated continuously. Note that it is also only an approximation obtained by multiplying the  $G(0)$  approximation with the average count rate on both virtual correlation channels A and B. Dependent on the chosen routing channels in A and B this may lead to figures that are not truly molecular brightnesses, however, as an adjustment aid they should be useful in any case.

The real-time correlation is very demanding in terms of CPU load. The correlator is implemented in a way that will make use of multiple processor cores. It is therefore beneficial to use one of the recent dual or quad core CPUs.

All other aspects of TTR data collection with correlator preview are the same as in plain TTR mode. Topics such as using external markers or how to use the data files should be looked up there.

## 5.6. Time-Resolved Emission Spectra

In time-resolved fluorescence research, it is often of great interest to observe time-dependent spectral shifts and decay changes in the context of solvent relaxation dynamics and general spectral evolution. This requires recording Time-Resolved Emission Spectra (TRES), ideally with automated wavelength scan and TCSPC data collection under full software control.

In order to measure emission spectra in combination with fluorescence lifetime, the PicoHarp software provides an automated TRES measurement mode. This mode allows to control a monochromator via a stepper motor and automated collection of spectrally resolved lifetime histograms. Data is collected as in standard Integration Mode and saved in different blocks of memory for each wavelength. Routing is not supported in TRES mode.

If a monochromator with appropriate stepper motor and associated drivers is installed, a specialized dynamic link library (`Mono.dll`) can be installed together with the PicoHarp software. The presence of this DLL will be detected by the software. In this case the button for Monochromator and TRES control on the toolbar will be enabled. Note that most of the monochromators can be controlled only through dedicated stepper motor hardware from PicoQuant, usually installed as part of a complete PicoQuant spectrometer. Custom configurations can only be supported upon special request.



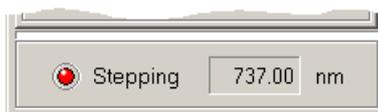
Clicking this button will launch a dialog for manual monochromator control and TRES setup. There you can set up parameters such as the start, step and end of the wavelength scan. There will be error messages if the monochromator/stepper and associated drivers are not configured properly.

### Monochromator / TRES Dialog

The monochromator used for TRES measurements is controlled by a special dialog. It shows a tab control and a status panel at the bottom of the dialog. The three tabs give access to the *Initialization*, *TRES* and *Manual* page of the dialog.



The status panel shows the current monochromator position in *nm* and whether the monochromator is currently moving (the red "LED" labelled *Stepping* is flashing, when it moves).



The dialog is "non modal", i.e. it does not need to be closed before other windows can be accessed. It has its own icon in the Windows Taskbar, from where it can be brought back to the top, should it have been covered by other windows.

### Monochromator Models

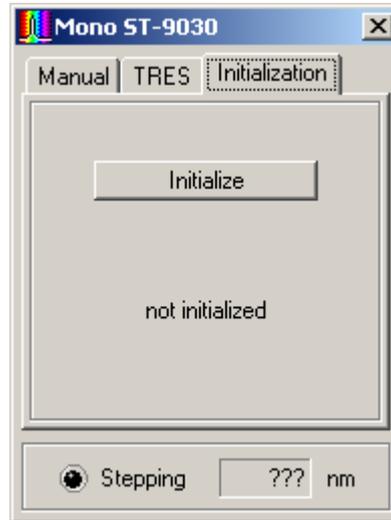
The type of the current monochromator is displayed in the title bar of the dialog. Apart from a simulation mode, as of this release the following monochromator models are supported: Sciencetech ST-9030, ST-9030DS and ST-9055, Acton Research SP-2150, SP-2155, SP-150 and SP-275. Some other Acton Research models are partially supported. Note that some monochromators can only be controlled through dedicated stepper motor hardware from PicoQuant or the monochromator supplier. Usually these are installed as part of a complete PicoQuant spectrometer. Please contact PicoQuant for details.

If your system is not configured for use of a monochromator, the Monochromator/TRES dialog shows the caption *Simulation*. This is for demonstration purposes only. It will work the same way as the "real" ST-9030 monochromator described below, except that pressing the *Initialize* button you will be prompted to decide, whether the initialization should be successful or not. To continue with the demonstration, press Yes. Of course the simulated monochromator movements will not have any effect on the measured TCSPC histograms.

## Monochromator Initialization

Before use, the monochromator must be initialized. This means, it has to seek its mechanical reference position (determined by a micro switch). Physical calibration values are kept in a monochromator configuration file (`monochromator.cfg`) which is generated by the system supplier at the time of delivery. Do not edit this file unless you have exact instructions for doing so.

Position initialization is required each time the software is started. It can be controlled on the Initialization page of the dialog. At first startup the dialog shows the state "not initialized". This means the software has no information about the current monochromator/stepper position.



The position initialization is started by the *Initialize* button. If the initialization is running, the caption of the button will change to *Stop* and the LED will start flashing. You should also hear the stepper motor. If the initialization is successful, the string "not initialized" will change to "Initialization OK", otherwise it will change to "Initialization failed!". In this case, check cabling and power supply of the system. This may happen if either the *Stop* button was pressed or an error occurred during the initialization process. The *Manual* and *TRES* pages will not be available unless the initialization was successfully performed. If for any reason the initialization becomes invalid (e.g. if the monochromator was turned manually and subsequently accidentally reaches the end point switch) the *Initialization* page will be activated and access to the *Manual* and *TRES* pages will be blocked.

## Manual Monochromator Control

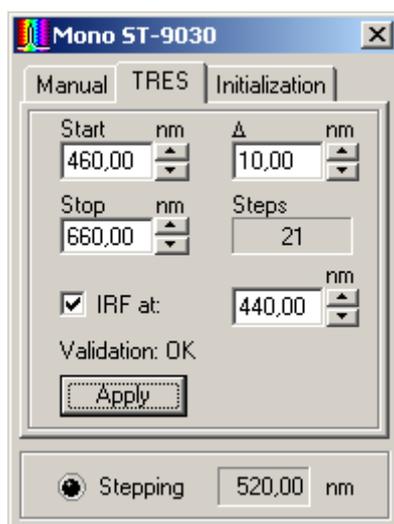
The *Manual* page of the dialog can be used to position the monochromator manually at a given wavelength. The desired position can be entered in the *Position* edit box. Clicking *Apply* or Pressing `<Enter>` will cause the monochromator to move to the specified position. Alternatively, the spin buttons right to the edit box can be used to modify the position. The step width of this action is defined in the  $\Delta$  (delta) edit box. Pressing arrow buttons will make the change take effect immediately; entering values by keyboard require pressing *Apply* or `<Enter>` to show an effect.



If there is a large difference between the current monochromator position and the position entered, the movement will take some time. If you wish to interrupt this process, press the *Stop* button. The monochromator immediately will slow down and stop. The deceleration is necessary to preserve the calibration.

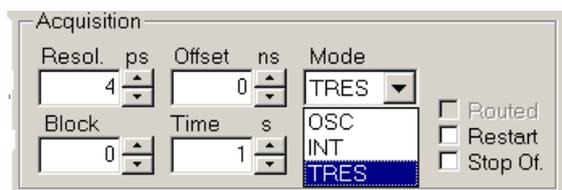
### Running TRES Measurements

The *TRES* page of the dialog contains the controls used to set up an automated scanning of the monochromator. The start and stop wavelengths of the scan can be entered in the corresponding edit boxes, the  $\Delta$  (delta) box defines the step width of the scan. The scan starts always exactly at the *Start* value. If the *Stop* position cannot be reached by an integer multiple of the step width, the scan stops at the last position within the start/stop range. If the start position is larger than the stop position, the scanning direction will be reversed. The maximum range and number of wavelength steps is determined by the physical characteristics of the monochromator as well as the maximum number of memory blocks to record histograms in (512). Again, pressing the spin buttons will apply the change in the corresponding edit boxes immediately; entering values manually require pressing `<Enter>` or the *Apply* button.



The IRF can be measured automatically as part of the scan, at a separately chosen wavelength. To do so, check the box entitled IRF at and enter the wavelength at which to measure the IRF in the corresponding edit box. The monochromator will go to the IRF position at the beginning of the TRES measurement. After the IRF was measured, the actual TRES scan will be performed. The IRF will always be recorded in memory block 0, while the wavelength scan always starts in block 1. Note that starting a TRES measurement will clear and subsequently overwrite all curves in memory that may have been recorded previously. If the curves in memory were loaded from a file, the actual file on disk will not be overwritten until you save to the file (Save menu or corresponding toolbar button). Use Save As and select a new file name if you want to preserve the old file.

The TRES measurement is selected in the PicoHarp control panel. You need to use the selection box for the measurement mode to select TRES mode (instead of OSC or INT).



A TRES measurement is started or canceled as usual via the toolbar buttons for start and stop. The data acquisition will be performed like a standard Integration Mode measurement, while the block number is incremented for each new wavelength.

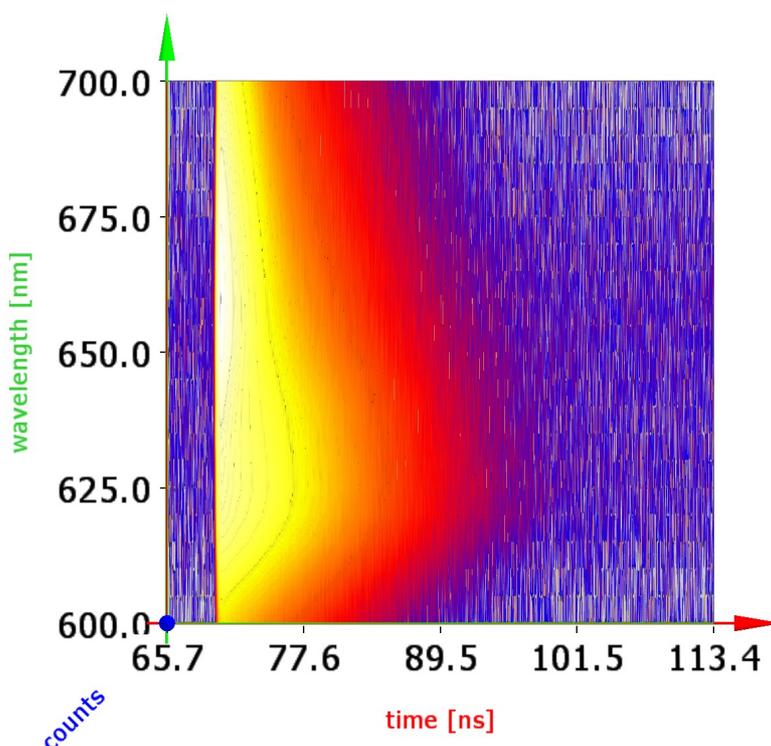
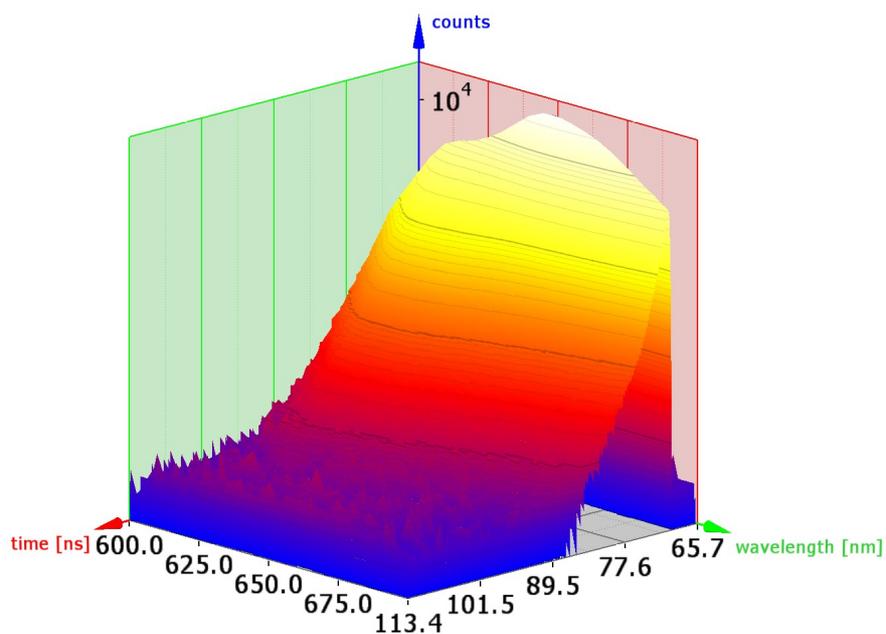


At the beginning of each TRES run, the active curve (block) will be set to 0, or to 1 if no IRF is collected. All curves that were previously collected and not saved to a file will be overwritten with new data. During a TRES run, the currently collected data will always be shown as Trace 0 (dark blue), even though the block (Trace) number is actually incremented at each wavelength step. This is to overcome the limitation of only 8 display curves being available. You can watch the block number in the control panel to see the curve currently being collected. Manual entry will be disabled during the run. Also, you can watch the status bar to see

the current wavelength and stepping activity. Note that for a TRES measurement the control panel setting 'Stop Of' applies as in Integration mode, while 'Restart' is meaningless in TRES mode.

### TRES Data Analysis and Visualization

Having collected a complete TRES data set you can save it to a regular \*.phd file. You can also inspect individual curves via the Trace Mapping dialog. You can furthermore use the software tool FluoPlot from PicoQuant to visualize and analyze the data in various 2D and 3D representations with a multitude of options for colouring, scaling and changing view aspects in 3D. The figures below show FluoPlot visualizations from a TRES measurement of mixed oxazine dyes in ethanol.



FluoPlot is provided on your software installation disk(s). Just run FluoPlotSetup.exe from there. Note that the program requires modern hardware acceleration with support for OpenGL version 1.5 or higher. Speedy handling of large data files requires sufficient memory. Further data analysis may need to be performed by dedicated software, either custom programs or specialized solutions available from PicoQuant.

## 6. Controls and Commands Reference

### 6.1. Main Window

#### The Title Bar

The title bar is located along the top of a window. To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars. The title bar may contain the following elements:

System Menu	
<Filename> - PicoHarp	
Minimize Button	
Maximize or Restore Button, resp.	
Close Button	

#### Scroll bars

Displayed at the right and bottom edges of the PicoHarp window if a certain minimum window size is reached. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the display area. You can use the mouse to scroll to other parts of the window.

#### Size command

Use this command to resize the active window.

Note: The command is unavailable for already maximized or minimized windows.

##### Shortcut

Mouse: Drag the corners or edges of the window.

Keys: <Alt>+<Space> S

#### Minimize command

Use this command to reduce the PicoHarp window to an icon. Running measurements will continue.

Note: The command is unavailable for already minimized windows.

##### Shortcut

Mouse: Click the minimize button  on the title bar.

Keys: <Alt>+<Space> N

#### Maximize command

Use this command to enlarge the active window to fill the available space.

Note: The command is unavailable for already maximized windows.

##### Shortcut

Mouse: Click the maximize button  on the title bar; or double-click the title bar of a not-enlarged window.

Keys: <Alt>+<Space> X

#### Restore command

Use this command to return from an enlarged window to the previous size.

Note: The command is only available for already maximized or minimized windows.

##### Shortcut

Mouse: Click the maximize button  on the title bar; or double-click the title bar of the enlarged window.

Keys: <Alt>+<Space> R

#### Close command

Use this command to close the active window or dialog box.

Double-clicking a system menu box is the same as choosing the Close command.

##### Shortcuts

Mouse: Click the close button  on the title bar.

Keys: <Alt>+<F4>

<Alt>+<Space> C

## 6.2. Toolbar

Toolbar buttons of special interest are explained in some more detail here:



### Context Help

When you choose the *Context Help* button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the PicoHarp window, such as on another toolbar button. The help topic will be shown for the item you clicked.



### Axis Settings button

Clicking this button opens the Axis Settings Panel. This panel provides controls for the axis limits and lin / log display switching. Double-clicking the axes opens the same dialog. The dialog is non-modal, so it can remain permanently open. The panel will remember its last position when closed. When it is opened the next time it will be at that screen position again.



### Data Cursor Button

Opens the data cursor dialog box. You can use this dialog to mark and retrieve the data values of individual histogram channels. A pair of cross hairs will be provided for marking data points with the mouse or keyboard. When the data cursor dialog is activated, the cross hairs will appear in the histogram display area. By clicking on or near a data point, the current cross hair (black) will snap to that point. At the same time a second cross hair (grey) will jump to the previously marked point. The data cursor dialog then shows the current count and time values for these points as well as the corresponding differences (deltas). The data cursor always applies to one "active" curve only. You can select this active curve at the top of the dialog where it is shown with its corresponding curve colour. While normally the grey cross hair jumps to the previous data point, you can modify this behaviour by holding the <SHIFT> key down while clicking on curve points. The grey cross hair will then remain at its previous position. This allows finding a particular point while keeping the other one fixed.

Another advanced mode of operation of the data cursor is possible via the <CurLeft> and <CurRight> arrows of the keyboard. Pressing these keys will direct the black cursor to the next left or right point in the curve. Again, the behaviour of the marker for the previous point (grey cross) can be controlled via the <SHIFT> key.



### Control Panel Button

Launches the PicoHarp control panel. If the control panel is already open, subsequent clicking of this button will just make the control panel the active window. The dialog is non-modal, so it can remain permanently open. The panel will remember its last position when closed. When it is opened the next time it will be at that screen position again, even in the next PicoHarp session.



### Calibrate Button

Calibrates the time measurement circuits. This is necessary whenever temperature changes > 5 K occurred. Calibrate after warming up of the instrument before starting serious measurements. Warming up takes about 20 to 30 minutes dependent on the lab temperature. It is completed when the cooling fan starts to run for the first time. Calibration takes only a few seconds. During calibration the status bar shows the message *CALIBRATING*. Calibration is only important for serious measurements. You can use the instrument during the warming-up period for set-up and preliminary measurements. For very long measurements, allow some more time for thermal stabilization, calibrate immediately before the measurement and try to maintain a stable room temperature during the measurement. The permissible ambient temperature is 15 to 35°C. Do not obstruct the cooling fan at the back and the air inlets at the bottom of the housing.



### Trace Map Button

Launches the trace mapping dialog. The PicoHarp can measure histograms in up to 512 memory blocks. Out of these, up to 8 curves can be displayed. The trace map dialog is used to select the curves to display. Tick the individual "show" boxes to display a curve. Select the number of the memory block you wish to map the individual display curve to. Choose the active memory block you wish to use for the next measurement in the PicoHarp control panel.



### TTTR Mode Button (disabled if this mode is not available)

Opens the TTTR mode dialog box. Use this dialog to enter the acquisition time and the destination file for a TTTR mode run. Make sure all other measurement parameters have been set and tested in interactive mode before entering TTTR mode. Note: TTTR mode is not part of the standard product and must be purchased separately. This includes T2 and T3 Mode. If the option is not installed the button will remain disabled (grey).



### TTTR Real-Time Correlator Button (disabled if TTTR mode is not available)

Opens the TTTR mode real-time correlator dialog box. Use this dialog for FCS preview during a T2 or T3 mode run. Make sure all other measurement parameters have been set and tested in interactive mode before entering TTTR correlator mode. Changing to this mode takes a few seconds for reconfiguration of the hardware. Note: TTTR mode is not part of the standard product and must be purchased separately. See section 5.6 on Time-Resolved Emission Spectra.



### PHR 800 Router Setup Button

If a PHR 800 router is connected to the PicoHarp this button will be enabled. It launches a dialog for setting up the inputs and the built-in CFDs for the four router channels. The settings will be stored in the Windows registry and the dialog does not need to be used again once satisfactory settings for a given measurement setup have been established. Note that the settings are also retrieved upon loading a data file.



### Monochromator / TRES Button (disabled if device is not available)

In order to measure Time-Resolved Emission Spectra, the PicoHarp software provides a TRES measurement mode, that allows to control a monochromator via stepper motors and automated collection of spectrally resolved lifetime histograms. Clicking this button will launch the dialog for manual monochromator control and TRES setup. There you can set up parameters such as the start, step and end of the wavelength scan. There will be appropriate error messages if the monochromator/stepper and associated drivers are not configured properly.



### Software Settings Button

Opens the software settings dialog box. Use this dialog to change standard settings of the PicoHarp software. Notably these are: *Display rate* (0.1 to 1 s), *Draw mode* (lines, stairs), *Prompt overwrite* (warning before overwriting existing data) and *TTTR Marker Edges*. The control connector of the PicoHarp 300 provides TTL inputs for synchronization signals. The markers can be recorded at the rising or falling edge of the corresponding TTL signal. The active edges can be chosen here. The dialog also allows selective disabling of warnings that you do not wish to receive. All settings will be kept in the Windows registry and will be retrieved at the next program start. On multi-user systems (2000, XP, Vista) they will be individual to each user.

## 6.3. Control Panel

### Input Channel 0 - Zero Cross edit box and spin control

Here the zero cross level of the Constant Fraction Discriminator (CFD) for the Channel 0 (sync) input can be set. Units are millivolts (mV), the permitted range is 0 to 20. Type the value as an integer in the edit box and press <Enter>. Alternatively, use the spin control (next to it) to increment/decrement the current value. In this case changes take effect immediately without pressing <Enter>. See section 5.1 *Setting up the Input Channels* to get started.

See also Input Channel 0 - Discriminator Level

**Input Channel 0 - Discriminator Level** edit box and spin control

Here the discriminator level of the Constant Fraction Discriminator (CFD) for the Channel 0 (sync) input can be set. Units are millivolts (mV), the permitted range is 0 to 800. This corresponds to real input voltage levels of 0 to -800 mV. Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control next to it, to increment/decrement the current value. In this case changes take effect immediately without pressing <Enter>. See *Setting up the Input channels* to get started.

See also: Input Channel 0 - Zero Cross

**Input Channel 0 - Sync Divider** selection box

Here the programmable divider of the sync input can be set. This allows to reduce the sync input rate so that the period is at least as long as the dead time. This is required for fast sync sources (> 10 MHz) Internal logic determines the sync period and re-calculates the sync signals that were divided out. It must be noted that this only works with stable sync sources that provide a constant pulse-to-pulse period, but all fast laser sources known today meet this requirement within an error band of a few picoseconds. With random and low rate signals (< 1 MHz), e.g. in coincidence correlation experiments, the divider must be set to 'None'.

**Input Channel 1 - Zero Cross** edit box and spin control

Here the zero cross level of the Constant Fraction Discriminator (CFD) for the Channel 1 input can be set. Units are millivolts (mV), the permitted range is 0 to 20. Type the value as an integer in the edit box and press <Enter>. Alternatively, use the spin control (next to it) to increment/decrement the current value. In this case changes take effect immediately without pressing <Enter>. See *Setting up the Input Channels* to get started.

See also: Input Channel 1 - Discriminator Level

**Input Channel 1 - Discriminator Level** edit box and spin control

Here the discriminator level of the Constant Fraction Discriminator (CFD) for the Channel 1 input can be set. Units are millivolts (mV), the permitted range is 0 to 800. This corresponds to real input voltage levels of 0 to -800 mV. Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control next to it, to increment/decrement the current value. In this case changes take effect immediately without pressing <Enter>. See *Setting up the Input Channels* to get started.

See also: Input Channel 1 - Zero Cross

**Resolution** edit box and spin control

Use this set of input controls to specify the acquisition resolution. Units are picoseconds (ps). Possible choices are the base resolution (4 ps) and binary multiples of this resolution (8, 16, 32, ... 512 ps). Type the desired resolution value (ps) as an integer in the edit box and press <Enter>. Alternatively, use the spin control (next to it) to increment/decrement the current value. In this case, changes take effect immediately without pressing <Enter>. In case of entering values other than valid resolutions, the next suitable resolution step is chosen automatically.

There are always  $2^{16} = 65536$  channels in one histogram. With the different resolutions, the approximate time ranges covered are between  $65536 \cdot 4 \text{ ps} = 262 \text{ ns}$  up to  $65536 \cdot 512 \text{ ps} = 33.5 \text{ } \mu\text{s}$ . The choice of range must compromise between resolution and time span covered. The smallest range offers the best resolution and the shortest span (vice versa for the largest range). For sync rates > 5 MHz the highest resolution is usually most appropriate, since the smallest range still covers the full sync period. For lower sync rates the histogram range is too small to cover the full sync period. The decay curve region of interest may therefore lie outside the acquisition window of 65536 channels. Apart from switching to a lower resolution it is possible to shift the acquisition window relative to the sync frame by means of the offset. Note that this offset cannot replace cable delays for appropriate placement of the signal peak within in the sync span.

**Offset** edit box and spin control

For sync rates > 5 MHz the highest resolution is usually most appropriate, since the smallest range still covers the full sync period. The offset should in these cases always be set to 0.

For lower sync rates the histogram range may be too small to cover the full sync period. The decay curve region of interest may therefore lie outside the acquisition window of 65536 channels. Apart from switching to a lower resolution it is possible to shift the acquisition window relative to the sync frame by means of setting the offset > 0.

Use this set of input controls to specify the desired offset. Units are Nanoseconds (ns). Possible offsets are in the range from 0 to 100000 ns. Type the value as an integer in the edit box and press <Enter>. The offset value can be entered in steps of one nanosecond. Alternatively, use the spin control next to the edit box to increment/decrement the current value. In this case changes take effect immediately without pressing <Enter>.

Internally the offset is subtracted from each start-stop measurement before it is used to address the histogram channel to be incremented. Therefore, increasing the offset means shifting the signal on the screen to the left, towards earlier times.

Note that the offset has no effect on the relative timing of laser pulses and photon events. It merely shifts the region of interest where data is to be collected. The relative timing of laser pulses and photon events can only be controlled by means of cable delays (e.g. in order to shift your fluorescence decay within the sync frame so that it is not truncated)

#### **Acquisition Time** edit box and spin control

Set the desired measurement time here. Units are seconds (s). The permitted range is 0.001 to 360000 in steps of 0.001. Type the desired value in the edit box and press <Enter>. Alternatively, use the spin control (next to it) to increment/decrement the current value. In this case, changes take effect immediately without pressing <Enter>. The default increment/decrement per mouse click is logarithmic.

#### **Trc./Block** (trace colour indicator, edit box and spin control)

The PicoHarp can measure histograms in up to 512 memory blocks. Out of these, up to 8 curves can be displayed and one "active block" can be used for measurements. Choose the active memory block you wish to use for the next measurement here. The trace mapping dialog is used to select the curves for display. Make sure the curve you are using is switched on (*shown*). You can open this dialog directly by clicking on the trace colour indicator. The colour indicator shows the trace colour the chosen block is currently mapped to. If it is a solid square, the curve is mapped and shown. If it is mapped but not shown, the indicator shows a small striped square. If the curve is not even mapped for display the indicator remains white. Make sure not to overwrite existing data in a block that was used before. In order to warn you if a trace is used, the heading of the Trc/Block selector will turn red.

#### **Routed** check box

Enable/Disable Routing. The check box will remain disabled if no router is found at startup.

#### **Restart** check box

If this box is ticked (checked) the measurement will automatically restart after the acquisition time has elapsed. Toggle the current setting with a mouse click. The setting is without effect in TRES mode.

#### **Stop on Overflow** check box

If this box is ticked (checked) the measurement will stop on overflow of any histogram bin. The overflow limit is set by the "StopAt" parameter. Toggle the current setting with a mouse click.

#### **Oscilloscope Mode** (selection box "Mode")

The selection *Oscilloscope* from the selection box "Mode" selects the current acquisition mode *Oscilloscope* for the PicoHarp. In oscilloscope mode the acquisition and display, once started, repeats at intervals given by the current acquisition time setting. The histogram accumulation always starts from scratch. This is useful for monitoring fast changes during optical alignment etc.

#### **Integration Mode** (selection box "Mode")

As opposed to oscilloscope mode, the histogram acquisition in integration mode is not reset to zero with each display refresh. The histogram continues to grow while the display is updated every 0.1 to 1 seconds determined by the refresh rate value selected in the *Software Settings Dialog*. Acquisition can be manually started and stopped, additionally the option "stop on overflow" will stop the acquisition when the maximum count of any channel overflows. The option "restart" will cause the acquisition to start again after the acquisition time has elapsed. This is similar to oscilloscope mode but allows watching the histogram grow if the acquisition time is fairly long.

**TRES Mode** (selection box "Mode")

Selects the measurement mode TRES (Time-Resolved Emission Spectra). This requires a monochromator and corresponding driver hardware/software to be installed. See section 5.6 on Time-Resolved Emission Spectra.

**Stop At** edit box and spin control

Here the stop count level for histogramming can be set. The setting is only meaningful if the check box "Stop Off" is active. Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control (next to it) to increment or decrement the current value. In this case changes take effect immediately without pressing <Enter>. The minimum is 0. If the entered value exceeds the allowed maximum of 65536, an error message appears. When using the spin control exceeding the maximum is simply not allowed. Instead of incrementing the value further, the system beeps a warning.

## 6.4. Axis Panel

**Time Axis Minimum** edit box and spin control

Here the start of the displayed time axis can be set. Units are nanoseconds (ns). Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control (next to it) to increment or decrement the current value. In this case changes take effect immediately without pressing <Enter>. The minimum is 0. If the entered value exceeds the current time axis maximum, an error message appears. When using the spin control exceeding the current time axis maximum is simply not allowed. Instead of incrementing the minimum value further, the system beeps a warning.

Note that the time axis labelling is precise in terms of relative information only. At *offset* = 0 the absolute time difference between the inputs is close to the shown times but may be off by some tens of picoseconds. Offsets > 0 change the placement of the acquisition window and the true time differences are then off accordingly.

The time axis settings will determine the number of channels exported via clipboard copy/paste. Only the channels within the displayed time range will be used.

**Time Axis Maximum** edit box and spin control

Here the end of the displayed time axis can be set. Units are nanoseconds (ns). Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control (next to it) to increment or decrement the current value. In this case changes take effect immediately without pressing <Enter>. The maximum allowed here is 33555 ns. If the entered value becomes smaller than the current time axis minimum, an error message appears. When using the spin control, violating the current time axis minimum is automatically prevented. Instead of decrementing the minimum value further, the system beeps a warning. Observe the notes on time axis interpretation given in the paragraph *TimeAxis Minimum* above.

The time axis settings will determine the number of channels exported via clipboard copy/paste. Only the channels within the displayed time range will be used.

**Count Axis Minimum** edit box and spin control

Here the start of the displayed count axis can be set. Units are counts. Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control next to it to increment/decrement the current value. In this case, changes take effect immediately without pressing <Enter>. In linear scale mode the default increment/decrement is 100. In log scale mode it is one decade. If the entered value exceeds the current count axis maximum, an error message appears. When using the spin control, exceeding the current count axis maximum is automatically prevented. Instead of incrementing the minimum value further, the system beeps a warning.

**Count Axis Maximum** edit box and spin control

Here the upper end of the displayed count axis can be set. Units are counts. Type the value as an integer in the edit box and press <Enter>. Alternatively use the spin control (next to it) to increment or decrement the current value. In this case, changes take effect immediately without pressing <Enter>. In linear scale mode the default increment/decrement is 100. In log scale mode it is one decade. If the entered value becomes smaller than the current time axis minimum, an error message appears. When using the spin control, violating the current count axis minimum is prevented automatically. Instead of decrementing the minimum value further, the system beeps a warning.

## 6.5. Trace Mapping Dialog

The PicoHarp software can measure histograms in up to 512 memory blocks. Out of these up to 8 curves can be displayed. The trace mapping dialog is used to select the curves to display. Tick the individual boxes 'Show' to display a curve. Select the number of the memory block you wish to map the individual display curve to.

The Trace Mapping Dialog also provides some statistics on each curve. These items are:

- **FWHM:** The Full Width Half Maximum of the curve peak (usually for IRF traces)
- **Max Count:** The count in the highest point of the curve
- **At Time:** The time corresponding to the Max Count channel
- **Resolution:** The channel width (bin width) of the curve

Furthermore there are several buttons:

- **Details:** Can be clicked to see more curve information
- **All:** Can be clicked to tick all traces as shown
- **0..7:** Can be clicked to set the default mapping of trace 0..7 to block 0..7
- **Clear (trash can):** Can be clicked to delete the contents of individual blocks

The trace mapping dialog can be launched from the corresponding button on the toolbar as well as through the trace colour indicator on the control panel.

## 6.6. Menus

### 6.6.1. File Menu

The *File* menu offers the following commands:

<i>New</i>	Opens a new data file, clears all histogram data and restores default settings..
<i>Open</i>	Opens an existing histogram file.
<i>Save</i>	Saves a histogram file.
<i>Save As...</i>	Saves an opened histogram file to a specified file name.
<i>Print</i>	Prints the currently displayed histogram.
<i>Page Setup</i>	Allows modifying the page layout for printing.
<i>Print Preview</i>	Displays the layout as it would appear printed.
<i>Print Setup</i>	Selects a printer and printer connection.
<i>1...4 &lt;Recent Filename&gt;</i>	Opens one of the four last recently opened files
<i>Exit</i>	Exits PicoHarp software.

#### New command

Use this command to create a blank histogram with the last default settings. You can open an existing histogram file with the *Open* command.

#### Shortcuts

Toolbar:



Keys:

<Ctrl>+N  
<Alt>+F N

## Open command

Use this command to open an existing histogram file. All control panel settings will be restored also. You can also open PicoHarp data files (\*.phd) by double clicking them or dragging them onto the PicoHarp icon. You can revert to a blank histogram and default control panel settings with the New command. You can also select other file types for loading as histogram data, notably T3 Mode files. Loading such files as histogram data is for diagnostics only and cannot use their full information content.

### Shortcuts

Toolbar: 

Keys: <Ctrl>+O  
<Alt>+F O

### File Open dialog box

The following options allow you to specify which file to open:

- Look in:* Select (i.e. browse into) the directory in which the file that you want to open resides.
- Main box* In this box you see the content of the chosen directory, filtered by *Files of type*.
- File name:* Type or select the filename you want to open. This box lists files with the extension you select in the *Files of type* box.
- Files of type* Select the type of file you want to open. The default PicoHarp extension is .phd

## Save command

Use this command to save the current histogram data to its current name and directory. When you save a histogram for the first time, PicoHarp displays the Save As... dialog box so you can name your file. This command is unavailable if a measurement is running. If you want to change the name and directory of an existing file before you save it, choose the Save As... command.

### Shortcuts

Toolbar: 

Keys: <Ctrl>+S  
<Alt>+F S

## Save As... command

Use this command to (re-)name the current histogram data and save. The software displays the Save As... dialog box so you can name your file. To save a file with its existing name and directory, use the Save command.

### File Save As... dialog box

The following fields allow you to specify the name and location of the file you are about to save:

- Save in:* Select (i.e. browse into) the directory where you want to store the file.
- Main box* In this box you see the content of the chosen directory, filtered by *Save as type*.
- File name:* Type a new filename to save a histogram with a different name. The software automatically adds the extension you specify in the *Save as type* box.
- Save as type:* Selects a filter on the directory chosen in the *Save in:* box. Only files that pass the filter are shown in the dialogs main box. Additionally the software adds the extension associated with the chosen filter to a given file name if it was given without extension.

### Print command

Use this command to print the currently displayed histogram curves. This command presents a Print dialog box, where you may specify the number of copies, the destination printer, the paper orientation and other printer setup options.

#### Shortcuts

Toolbar:



Keys:

<Ctrl>+P  
<Alt>+F P

### Page Setup command

Use this command to change the print layout. This command presents the page setup dialog box, where you may select or deselect various items to appear in the print. You can use print preview to check the resulting layout.

#### Shortcut

Keys: <Alt>+F U

### Print Preview command

Use this command to display the layout as it would appear when printed. The main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

#### Shortcut

Keys: <Alt>+F V

### Print Setup command

Use this command to select a printer and a printer connection. This command presents a Print Setup dialog box, where you specify the printer and its connection.

#### Shortcut

Keys: <Alt>+F R

### 1, 2, 3, 4 command (most recently used files)

Use the numbers and filenames listed at the bottom of the *File* menu to open the last four files you closed. Choose the number that corresponds with the file you want to open.

#### Shortcuts

Keys: <Alt>+F 1  
<Alt>+F 2  
<Alt>+F 3  
<Alt>+F 4

### Exit command

Use this command to end your PicoHarp session. Save your data before exiting.

#### Shortcuts

Mouse: Click the close button  on the title bar.

Keys: <Alt>+<F4>  
<Alt>+F X

## 6.6.2. Edit Menu

This menu offers only one command:

### Copy command

Copies the currently displayed curves in ASCII format to the clipboard. This can be used to export histogram data to spreadsheet or data analysis software, e.g. the FluoFit Global Fluorescence Decay Fit Software. Note that only the currently selected curves within the current display limits are copied. If the curves in the display are of different resolution then all 65536 channels will be copied. Copying data to the clipboard replaces the contents previously stored there.

The exported data is plain `<Tab>` separated ASCII and can be read by most of the common spreadsheet programs or data analysis software.

This command is unavailable while a measurement is running.

### Shortcuts

Keys:            `<Alt>+E C`  
                   `<Ctrl>+C`

## 6.6.3. View Menu

The *View* menu offers the following commands:

<i>Toolbar</i>	Shows or hides the toolbar.
<i>Status Bar</i>	Shows or hides the status bar.
<i>Axis Panel</i>	Shows the axis settings panel.
<i>Control Panel</i>	Shows the control panel.
<i>Trace Mapping</i>	Shows the trace mapping dialog.

### Toolbar command

Use this command to display and hide the toolbar, which includes buttons for some of the most common commands, such as *File Open*. A check mark appears next to the menu item when the toolbar is displayed.

### Status Bar command

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, the current measurement activity and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed. The status bar is displayed at the bottom of the PicoHarp main window. To display or hide the status bar, use the Status Bar command in the View menu. The left area of the status bar describes actions of menu items as you navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you do not wish to execute the command, then release the mouse button while the mouse pointer is not over the toolbar button. The second area from the left shows the current measurement status of the PicoHarp. Next to it you may find a warnings icon that you can click for details (see also section 8.1). The rightmost area of the status bar indicates if the `<Tab>` and `<NumLock>` keys are latched down.

### Axis Panel command

Use this command to display the axis settings panel. The result is the same as the axis panel button in the toolbar or double clicking the on the histogram axes. The axis settings panel can be closed by clicking its close button.

### Control Panel command

Use this command to display the control panel. This is the same as the control panel button in the toolbar. The control panel can be closed by clicking its close button.

## Trace Mapping command

Use this command to display the trace mapping dialog. This is the same as the corresponding button in the toolbar. The trace mapping dialog can be closed by clicking its close button.

## 6.6.4. Help Menu

The *Help* menu offers the following commands, which provide you assistance with this application:

<i>Help Topics</i>	Offers you the contents list of topics on which you can get help.
<i>Help Index</i>	Offers you an index to topics on which you can get help.
<i>Help Search</i>	Offers you a means of full text search on all help topics.
<i>Activate Context Help</i>	Switches the cursor mode to "point to item" for context help.
<i>About PicoHarp...</i>	Displays version information of the PicoHarp software.

Note: Online help (context help) on most functions, dialogs, control items etc. is available via the <F1> key.

### Help Topics command

This command opens a browser offering a tree view like list of contents, and a help page. Browse through the contents with your mouse or the cursor keys. <Enter> opens / closes chapters, <CurUp> / <CurDown> turns the pages. The help page to the right of the browser will always be held up-to-date.

#### Shortcuts

Keys:           <F1>                   (if online help was last recently opened in topics mode)  
                  <Alt>+H T

### Help Index command

This command opens a browser offering a index list of keywords, and a help page. Browse through the contents with your mouse or the cursor keys. <Enter> opens the current item, <CurUp> / <CurDown> navigates through the index.

#### Shortcuts

Keys:           <F1>                   (if online help was last recently opened in index mode)  
                  <Alt>+H I

### Help Search command

This command opens a browser offering a means of a full text search on words or word groups over the whole online help volume. You can enter more specific search pattern, using the operators "AND", "OR", "NEAR" and "NOT".

#### Shortcuts

Keys:           <F1>                   (if online help was last recently opened in search mode)  
                  <Alt>+H S

### Activate Context Help

Changes the mouse cursor mode to context help mode. You can choose context sensitive help for a visual element in the GUI by simply clicking on it.

#### Shortcuts

Keys:           <Shift>+<F1>  
                  <Alt>+H C

### About PicoHarp... command

Use this command to display the copyright notice and version number of your PicoHarp software and hardware, if installed.

#### Shortcut

Keys:           <Alt>+H A

## 6.7. Title and Comment Editor

You can use this dialog to edit the file title/comment. It can be opened via double-click on the title, via the File menu or via the Print Preview Toolbar. The text you enter here will be stored in the data file. The first line will be displayed as the file title over the histogram display area. The text you can enter here is limited to 4 lines and 255 characters. Upon loading of a data file, the title and comment will also be retrieved. It will also be included in prints, if the corresponding check mark is set in the page setup dialog box available through the *File* menu.

## 6.8. Print Preview Dialog

The Print Preview command (File menu) opens the Print Preview Dialog. Use this Dialog to preview the layout as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job. It also offers a *Page Setup* button that opens a dialog for the print layout and editing the file comment.

### Print Preview Dialog Toolbar

The print preview toolbar offers you the following options:

<i>Print</i>	Bring up the print dialog box, to start a print job.
<i>Next Page</i>	Preview the next printed page.
<i>Prev Page</i>	Preview the previous printed page.
<i>One Page / Two Page</i>	Preview one or two printed pages at a time.
<i>Zoom In</i>	Take a closer look at the printed page.
<i>Zoom Out</i>	Take a larger look at the printed page.
<i>Page Setup</i>	Change the layout of the printed page.
<i>Close</i>	Return from print preview to the editing window.

## 7. Problems, Tips & Tricks

### PC Performance Issues

The PicoHarp device and its software interface are a complex real-time measurement system demanding appropriate performance both from the host PC and the operating system. This is why a fairly modern CPU and sufficient memory are required. A modern graphics interface is also recommended. Screen resolution should be at least 800x600. At least a 1 GHz processor, 512 MB of memory and a fast hard disk running in DMA mode are recommended.

In order to maintain correct interaction between the measurement hardware and the display of histogram curves and user input, the operating system's message passing mechanism is used. It is recommended not to overload the system by running other processes in the background while measuring with the PicoHarp. The PC's own occasional network activity should be no problem but running the machine e.g. as a server for other PCs is not recommended. In principle, of course any kind of background activity is allowed. Should the system become overloaded the PicoHarp histograms will not be "wrong" or distorted, but measurement times and display rates may become irregular. You can even iconize the PicoHarp during measurements without doing damage. This may be of interest for lengthy measurements in integration mode, where one is only interested in accumulating a certain amount of counts without need for permanent monitoring. The "panel meters" showing the current count rates are not meant to be 100 % accurate. They merely serve as an aid for setting up the system. Some of them may suffer from system overload. Accordingly the values shown in the curve details (trace mapping dialog) are subject to such tolerances.

In TTTR mode, system overload will manifest itself in loss of data. Here no other background processing should be allowed. The faster the PC, the less these issues will matter. TTTR mode performance (i.e. max. throughput without loss) may also be improved by defragmenting the hard disk before long measurements. The real-time correlation is very demanding in terms of CPU load. The correlator is implemented in a way that will make use of multiple processor cores. It is therefore beneficial to use one of the recent dual or quad core CPUs.

### USB Interface

In order to deliver maximum throughput, the PicoHarp 300 uses state-of-the-art USB 2.0 bulk transfers. This is why the PicoHarp must rely on having a working USB 2.0 host interface. USB 2.0 host controllers of modern PCs are usually integrated on the mainboard. For older PCs they may be upgraded as PCI cards. Throughput is usually limited by the host controller and operating system, not the PicoHarp. Make sure that the host controller drivers are correctly installed, otherwise they may be running only in USB 1.0 mode. This may require using the mainboard manufacturer's driver disk rather than relying on Windows drivers alone. Do not run other bandwidth demanding devices on the same USB interface when working with the PicoHarp. USB cables must be rated for USB 2.0 high speed. Older cables do not meet this requirement and can lead to errors and malfunction. Similarly, some PCs have poor internal USB cabling, so that USB sockets at the front of the PC are sometimes unreliable.

### Histogram Artefacts

Disturbing histogram ripple is strongly dependent on the quality of the input signals. Try to deliver the best possible signal quality with clean and reasonably steep active edges and without overshoot and ringing. It is recommended to use detector and sync signals of similar amplitude to avoid excess crosstalk. If the PDL 800-B picosecond diode laser is used, attenuating the sync pulses by 10 dB immediately at the laser driver may reduce histogram ripple (set CFD level accordingly). Always use good quality 50  $\Omega$  components and coax leads with proper shielding and correct termination. We recommend cables of type RG223 with double braided and silver plated screen. Check your setup for ground loops. Grounding different system components (PC, detector, detector power supply, diode laser driver, etc.) at different points can induce considerable noise in the ground lines. Because signal return paths may share the same ground lines, this noise is copied into the signal lines and causes increased timing jitter and/or histogram ripple. Network cables and mobile phones may also be sources of surprising noise. Use properly impedance matched power splitters (reflection-free T-pads) if signals must be fed to multiple 50  $\Omega$  inputs. Never use ordinary Tees. All accessories are available from PicoQuant. PMT detectors should be connected through a suitable high speed preamplifier (available from PicoQuant). MCP-PMT detectors should be connected through an amplifier with slightly higher gain (e.g. PAM 102-M from PicoQuant). TTL-SPAD-detectors (Perkin-Elmer SPCM-AQR) must be connected through a pulse inverter (PicoQuant SIA 400).

A problem often overlooked is pile-up and dead time related histogram distortion. It becomes noticeable if the detector count rate exceeds 2 % of the sync rate. This is why high excitation rates are so important. Pile-up is an inherent problem of TCSPC and not a fault of the hardware. The only way to handle it safely is to maintain count rates <1 to 2 % of the sync rate. Pile-up may to some extent also be corrected for, but this can be extremely difficult in practice. For further details see our TechNote on TCSPC or the literature given in section 2.

### Warming Up and Calibration

Observe the warm-up period of at least 20 minutes (dep. on ambient temperature) before using the PicoHarp for serious measurements. Warming up takes about 20 to 30 minutes dependent on the lab temperature. It is completed when the cooling fan starts to run for the first time. You can use this time for set-up and preliminary measurements. Calibrate one more time before starting serious measurements. The calibration takes only a few seconds. For very long measurements allow some more time for thermal stabilization and avoid temperature changes > 5 K in the environment of the instrument. The permissible ambient temperature is 15 to 35°C. Do not obstruct the cooling fan at the back and the air inlets at the bottom of the housing. Make sure that the cooling air can circulate freely and no other hot instrument is under the cooling inlets.

### Custom Programming of the PicoHarp

A programmer's library (DLL) for custom Windows software development is available to build your own applications e.g. in LabVIEW, C++, Delphi, Lazarus, Matlab or Visual Basic. A rich set of demo code is provided for an easy start. A Library for Linux is also available. It provides the same interface, so that applications can easily be ported between the platforms.

### Software Updates

We constantly improve and update the software for our instruments. This includes updates of the configurable hardware (FPGA). Such updates are important as they may affect reliability and interoperability with other products.

The software updates are free of charge, unless major new functionality is added. It is strongly recommended that you register for email notification on software updates. If you wish to receive such notification, please email your name and the serial number of your PicoQuant product(s) to [info@picoquant.com](mailto:info@picoquant.com). You will then receive update information with links for download of any new software release.

### Bug Reports and Support

The PicoHarp 300 system has gone through extensive testing. Nevertheless, it is a very cutting edge product and we would like to offer you our support in any case of problems with the system. Do not hesitate to contact PicoQuant in case of difficulties with your PicoHarp.

Should you observe errors or bugs caused by the PicoHarp system please try to find a reproducible error situation. E-mail a detailed description of the problem and all relevant circumstances, especially other hardware installed in your PC, to [info@picoquant.com](mailto:info@picoquant.com). Please run *msinfo32* to obtain a listing of your PC configuration and attach the summary file to your error report. Your feedback will help us to improve the product and documentation.

When you submit an error report referring to measurement data you have taken, attach the original data file. If the file is too large for email (>10 MB) please provide access to it on a public download server.

If you have serious problems that require the device to be sent in for inspection / repair, please request an RMA number before shipping the hardware. Observe precautions against static discharge under all circumstances in handling, packaging and shipping. Use original or equally protective packaging material.

Of course we also appreciate good news: If you have obtained exciting results with one of our systems, please let us know, and where appropriate, please mention the instrument in your publications. At our Website we maintain a large bibliography of publications related to our instruments. It may serve as a reference for you and other potential users. See [http://www.picoquant.com/\\_biblio.htm](http://www.picoquant.com/_biblio.htm). Please submit your publications for addition to this list.

## 8. Appendix

### 8.1. Warnings

When the PicoHarp software is running with functional hardware it continuously collects information about the input signals and the current acquisition settings. If these settings together with the input rates indicate possible errors, the software will indicate this by means of a warning icon.



While the software is running in interactive histogramming mode the warning icon is displayed at the bottom of the main window in the status bar. In the case of TTTR mode, it will appear directly in the TTTR mode dialog. The icon can be clicked to display a list of current warnings together with a brief explanation of each warning.

Note that the software can detect only a subset of all possible error conditions. It is therefore not safe to assume “all is right” just by seeing no warning. On the opposite, if any of the warnings turns out to be an unnecessary nuisance, e.g. because your specific measurement conditions knowingly force it, you can disable that warning via the software settings dialog available through the toolbar (see section 6.2).

The warnings are to some extent dependent on the current measurement mode. Not all warnings will occur in all measurement modes. Also, count rate limits for a specific warning may be different in different modes. The following table lists the possible warnings in the three measurement modes and gives some explanation as to their possible cause and consequences.

Warning	Histo Mode	T2 Mode	T3 Mode
<b>WARNING_INP0_RATE_ZERO</b> No counts are detected at input channel 0. In histogramming and T3 mode this is the sync channel and the measurement will not work without that signal.	√		√
<b>WARNING_INP0_RATE_TOO_LOW</b> The pulse rate at input channel 0 is below 1 kHz and the sync divider is >1. In histogramming and T3 mode this is the sync channel and the measurement will not work without a signal <1 kHz if the sync divider is >1. If your sync rate is really so low then you do not need a sync divider >1.	√		√
<b>WARNING_INP0_RATE_TOO_HIGH</b> You have selected T2 mode and the pulse rate at input channel 0 is higher than 5 MHz. The measurement will inevitably lead to a FiFo overrun. There are rare measurement scenarios where this condition is expected and the warning can be disabled. Examples are very short measurements where the FiFo can absorb all data.		√	
<b>WARNING_INP1_RATE_ZERO</b> No counts are detected at input channel 1. In histogramming and T3 mode this is the photon event channel and the measurement will yield nothing without this signal. You may sporadically see this warning if your detector has a very low dark count rate. In that case you may want to disable this warning.	√		√

Warning	Histo Mode	T2 Mode	T3 Mode
<p><b>WARNING_INP1_RATE_TOO_HIGH</b></p> <p>If you have selected T2 mode then this warning means the pulse rate at input channel 1 is higher than 5 MHz. The measurement will inevitably lead to a FiFo overrun. There are rare measurement scenarios where this condition is expected and the warning can be disabled. Examples are very short measurements where the FiFo can absorb all data.</p> <p>In histogramming and T3 mode this warning is issued when the input rate is &gt;10 MHz. This will probably lead to deadtime artefacts. There are rare measurement scenarios where this condition is expected and the warning can be disabled.</p>	√	√	√
<p><b>WARNING_INP_RATE_RATIO</b></p> <p>This warning is issued in histogramming and T3 mode when the rate at input 1 is over 5% of the rate at input 0. This is the classical pile-up criterion. It will lead to noticeable dead-time artefacts. There are rare measurement scenarios where this condition is expected and the warning can be disabled. Examples are antibunching measurements.</p>	√		√
<p><b>WARNING_DIVIDER_GREATER_ONE</b></p> <p>You have selected T2 mode and the sync divider is set larger than 1. This is probably not intended. The sync divider is designed primarily for high sync rates from lasers and requires a fixed pulse rate at channel 0. In that case you should use T3 mode. If the signal at channel 0 is from a photon detector (coincidence correlation etc.) a divider &gt;1 will lead to unexpected results. There are rare measurement scenarios where this condition is intentional and the warning can be disabled.</p>		√	
<p><b>WARNING_TIME_SPAN_TOO_SMALL</b></p> <p>This warning is issued in histogramming and T3 mode when the sync period (<math>1/\text{Rate}[\text{ch0}]</math>) is longer than the start to stop time span that can be covered by the histogram or by the T3 mode records. You can calculate this time span as follows:  Histogramming mode: <math>\text{Span} = \text{Resolution} * 65536</math>  T3 mode: <math>\text{Span} = \text{Resolution} * 4096</math>  Events outside this span will not be recorded. There are some measurement scenarios where this condition is intentional and the warning can be disabled.</p>	√		√
<p><b>WARNING_OFFSET_UNNECESSARY</b></p> <p>This warning is issued in histogramming and T3 mode when an offset &gt;0 is set even though the sync period (<math>1/\text{Rate0}</math>) can be covered by the measurement time span without using an offset. The offset may lead to events getting discarded. There are some measurement scenarios where this condition is intentional and the warning can be disabled.</p>	√		√

If any of the warnings you receive indicate wrong pulse rates, the cause may be inappropriate input settings, wrong pulse polarities, poor pulse shapes or bad connections. If in doubt, check all signals with a scope.

Note that all count rate dependent warnings relate to the PicoHarp's native input channels. If you are using a router, remember that all routing channels share the PicoHarp's input channel 1 and hence create a cumulated common rate at that channel.

## 8.2. Data File Formats

While for many purposes the ASCII export function via the clipboard is sufficient and easy, you also may want to access the PicoHarp data files via custom programs. This section provides the file structure specifications for your reference. Note that the file structures of PicoHarp 300 files are different from TimeHarp / NanoHarp files. Programs that need to process the files should check the string *Ident* in the file header. In order to safely recognize a PicoHarp file this string should be checked in any case. It is also necessary to check the version strings and read the data according to the format specification of that version. File formats in future software releases are subject to change without notification.

### 8.2.1. Interactive Mode File Format

The standard PicoHarp data files from interactive mode (\*.phd) are binary files which contain both the setup parameters and the actual histogram data. In order to safely recognize an interactive mode file (apart from the file extension) a program reading in these files must read and interpret the value *Measurement-Mode* which is 0 in interactive mode. Format version 2.0 \*.phd files are structured as shown in the table below. Note that the format version is not necessarily the same as that of the creating software version.

Data Item	Type	Description
Ident	char [16]	the string 'PicoHarp 300'
FormatVersion	char [6]	currently '2.0' (since Sept 2006)
CreatorName	char [18]	for this software 'PicoHarp Software'
CreatorVersion	char [12]	currently '2.3.0.0' (April 2009)
FileTime	char [18]	the file creation date and time
CR/LF	char [2]	0x0D,0x0A carriage return, line feed
Comment	char [256]	any ASCII string
NumberOfCurves**	int32	number of curves (histograms) stored in this file
BitsPerHistogBin**	int32	normally 32
RoutingChannels	int32	1 or 4
NumberOfBoards**	int32	reserved, now 1
ActiveCurve	int32	curve no. used for last measurement
MeasurementMode**	int32	0=interactive, 1=reserved, 2=T2, 3=T3
SubMode	int32	0=oscilloscope, 1=integration, 2=TRES
RangeNo*	int32	0=base resol., 1=x2, 2=x4, 3=x8, 4=x16, ....., 7=x128
Offset*	int32	offset in ns
AcquisitionTime*	int32	acquisition time in ms
StopAt	int32	stop at this many counts
StopOnOvfl	int32	stop if overflow occurred = 1, otherwise 0
Restart	int32	restart = 1, otherwise 0
DisplayLinLog*	int32	lin=0, log=1
DisplayTimeAxisFrom*	int32	lower time axis bound of display in ns
DisplayTimeAxisTo*	int32	upper time axis bound of display in ns
DisplayCountAxisFrom*	int32	lower count axis bound of display
DisplayCountAxisTo*	int32	upper count axis bound of display
The following block will appear in the file for each i = 1 to 8		
DisplayCurve[i].MapTo	int32	number of memory block to map this curve to
DisplayCurve[i].Show	int32	0=don't show, 1= show
end of block		
The following block will appear in the file for each i = 1 to 3		
Param[i].Start	float	used for automated measurements, e.g. TRES
Param[i].Step	float	reserved for automated measurements
Param[i].End	float	reserved for automated measurements
end of block		
RepeatMode	int32	reserved for automated measurements
RepeatsPerCurve	int32	reserved for automated measurements
RepeatTime	int32	reserved for automated measurements

RepeatWaitTime	int32	reserved for automated measurements
ScriptName	char[20]	reserved for automated measurements
The following block will appear in the file for each i = 1 to <NumberOfBoards> [currently this must be 1]		
Brd[i].HardwareIdent	char[16]	currently 'PicoHarp 300'
Brd[i].HardwareVersion	char[8]	currently '1.0' or '2.0'
Brd[i].HardwareSerial*	int32	hardware serial number
Brd[i].SyncDivider*	int32	input channel 0 (Sync) input divider
Brd[i].CFDZeroCross0*	int32	input channel 0 CFD zero cross level in mV
Brd[i].CFDLevel0*	int32	input channel 0 CFD discriminator level in mV
Brd[i].CFDZeroCross1*	int32	input channel 1 CFD zero cross level in mV
Brd[i].CFDLevel1*	int32	input channel 1 CFD discriminator level in mV
Brd[i].Resolution	float	resolution in ns
Brd[i].RouterModelCode	int32	0=none, 1=PHR_402, 2=PHR_403, 3=PHR_800
Brd[i].RouterEnabled	int32	1=enabled, 0=disabled (routing off)
The following block will appear in the file for each j = 1 to <NumberOfRouterChannels> [currently this must be 4]		
Brd[i].RtCh[j].InputType	int32	0=custom, 1=NIM, 2=TTL
Brd[i].RtCh[j].InputLevel	int32	in mV
Brd[i].RtCh[j].InputEdge	int32	1=rising, 0=falling
Brd[i].RtCh[j].CFDPresent	int32	0=not present, 1=present
Brd[i].RtCh[j].CFDLevel	int32	in mV
Brd[i].RtCh[j].CFDZCross	int32	in mV
end of block		
end of block		
The following block will appear in the file for each i = 1 to <NumberOfCurves>		
Crv[i].CurveIndex	int32	curve index (0..511)
Crv[i].TimeOfRecording	time_t	time of recording this curve
Crv[i].HardwareIdent	char[16]	hardware identifier string
Crv[i].HardwareVersion	char[8]	hardware version string
Crv[i].HardwareSerial	int32	hardware serial number
Crv[i].SyncDivider	int32	input channel 0 (Sync) input divider
Crv[i].CFDZeroCross0	int32	input channel 0 CFD zero cross level in mV
Crv[i].CFDLevel0	int32	input channel 0 CFD discriminator level in mV
Crv[i].CFDZeroCross1	int32	input channel 1 CFD zero cross level in mV
Crv[i].CFDLevel1	int32	input channel 1 CFD discriminator level in mV
Crv[i].Offset	int32	offset in ns
Crv[i].RoutingChannel	int32	0..3
Crv[i].ExtDevices	int32	external devices, bitwise coded
Crv[i].MeasMode	int32	0=interactive
Crv[i].SubMode	int32	oscilloscope=0, integration=1, TRES=2
Crv[i].P1	float	used for automated measurements, e.g. TRES (nm)
Crv[i].P2	float	reserved for automated measurements
Crv[i].P3	float	reserved for automated measurements
Crv[i].RangeNo	int32	0=base resol., 1=x2, 2=x4, 3=x8, 4=x16, ....., 7=x128
Crv[i].Resolution	float	resolution in ns
Crv[i].Channels**	int32	number of channels (bins) in this curve
Crv[i].AcquisitionTime	int32	Acquisition Time in ms
Crv[i].StopAfter	int32	stopped after this many ms
Crv[i].StopReason	int32	0=timeover, 1>manual, 2=overflow
Crv[i].InpRate0	int32	input 0 count rate as displayed by rate meter
Crv[i].InpRate1	int32	input 1 count rate as displayed by rate meter
Crv[i].HistCountRate	int32	histogramming count rate as displayed by rate meter
Crv[i].IntegralCount	int64	total count as displayed by meter
Crv[i].Reserved	int32	reserved
Crv[i].DataOffset**	int32	file offset of corresponding histogram data
Crv[i].RouterModelCode	int32	0=none, 1=PHR_402, 2=PHR_403, 3=PHR_800
Crv[i].RouterEnabled	int32	1=enabled, 0=disabled (routing off)

Crv[i].RtCh_InputType	int32	0=custom, 1=NIM, 2=TTL
Crv[i].RtCh_InputLevel	int32	in mV
Crv[i].RtCh_InputEdge	int32	1=rising, 0=falling
Crv[i].RtCh_CFDPresent	int32	0=not present, 1=present
Crv[i].RtCh_CFDLevel	int32	in mV
Crv[i].RtCh_CFDZeroCross	int32	in mV
end of block		
The following block will appear in the file for each i = 1 to <NumberOfCurves>, j = 1 to <Channels> as indicated in the curve header		
Counts[i][j]	uint32	the actual histogram data, leftmost channel first
end of block		

\* These items are stored as they were set / displayed last. They may not relate to the data stored.

\*\* These items may vary and thereby change the file structure dynamically. A program must interpret these before reading in further data.

See section 8.2.4 for the data types used. Also, please note that the software setup will install a ready to use set of examples in C, Pascal/Delphi/Lazarus and Matlab for PicoHarp \*.phd file access. They can be found in the subfolder \Filedemo in your chosen installation directory.

## 8.2.2. T2 Mode File Format

PicoHarp data files from T2 Mode (\*.pt2) are binary files which contain both the setup parameters and the actual event data. The setup data is in the file header which is in the same format as in standard interactive mode files. This is to allow common reading routines for the header. Note that this leads to some header values that are not meaningful in T2 Mode.

In order to safely recognize a T2 Mode file (apart from the file extension) a program reading in these files must read and interpret the value *MeasurementMode* which is 2 in T2 Mode. Format version 2.0 \*.pt2 files are structured as shown in the table below. Note that the format version is not necessarily the same as that of the creating software version.

Data Item	Type	Description
Ident	char [16]	the string 'PicoHarp 300'
FormatVersion	char [6]	currently '2.0' (since Sept 2006)
CreatorName	char [18]	for this software 'PicoHarp Software'
CreatorVersion	char [12]	currently '2.3.0.0' (April 2009)
FileTime	char [18]	the file creation date and time
CR/LF	char [2]	0x0D,0x0A carriage return, line feed
Comment	char [256]	any ASCII string
NumberOfCurves	int32	not meaningful in T2 Mode
BitsPerRecord	int32	32
RoutingChannels	int32	1 or 4
NumberOfBoards**	int32	reserved, now 1
ActiveCurve	int32	curve no. used for last measurement
MeasurementMode**	int32	0=interactive, 1=reserved, 2=T2, 3=T3
SubMode	int32	0=oscilloscope, 1=integration, 2=TRES
RangeNo*	int32	0=base resol., 1=x2, 2=x4, 3=x8, 4=x16, ....., 7=x128
Offset*	int32	offset in ns
AcquisitionTime*	int32	acquisition time in ms
StopAt	int32	stop at this many counts
StopOnOvfl	int32	stop if overflow occurred = 1, otherwise 0
Restart	int32	restart = 1, otherwise 0
DisplayLinLog*	int32	lin=0, log=1
DisplayTimeAxisFrom*	int32	lower time axis bound of display in ns
DisplayTimeAxisTo*	int32	upper time axis bound of display in ns
DisplayCountAxisFrom*	int32	lower count axis bound of display
DisplayCountAxisTo*	int32	upper count axis bound of display
The following block will appear in the file for each i = 1 to 8		
DisplayCurve[i].MapTo	int32	number of memory block to map this curve to
DisplayCurve[i].Show	int32	0=don't show, 1= show
end of block		
The following block will appear in the file for each i = 1 to 3		
Param[i].Start	float	used for automated measurements, e.g. TRES
Param[i].Step	float	reserved for automated measurements
Param[i].End	float	reserved for automated measurements
end of block		
RepeatMode	int32	reserved for automated measurements
RepeatsPerCurve	int32	reserved for automated measurements
RepeatTime	int32	reserved for automated measurements
RepeatWaitTime	int32	reserved for automated measurements
ScriptName	char[20]	reserved for automated measurements
The following block will appear in the file for each i = 1 to <NumberOfBoards> [currently this must be 1]		
Brd[i].HardwareIdent	char[16]	currently 'PicoHarp 300'
Brd[i].HardwareVersion	char[8]	currently '1.0' or '2.0'
Brd[i].HardwareSerial*	int32	hardware serial number

Brd[i].SyncDivider*	int32	input channel 0 (Sync) input divider
Brd[i].CFDZeroCross0*	int32	input channel 0 CFD zero cross level in mV
Brd[i].CFDLevel0*	int32	input channel 0 CFD discriminator level in mV
Brd[i].CFDZeroCross1*	int32	input channel 1 CFD zero cross level in mV
Brd[i].CFDLevel1*	int32	input channel 1 CFD discriminator level in mV
Brd[i].Resolution	float	resolution in ns
Brd[i].RouterModelCode	int32	0=none, 1=PHR_402 , 2=PHR_403, 3=PHR_800
Brd[i].RouterEnabled	int32	1=enabled, 0=disabled (routing off)
The following block will appear in the file for each j = 1 to <NumberOfRouterChannels> [currently this must be 4]		
Brd[i].RtCh[j].InputType	int32	0=custom, 1=NIM, 2=TTL
Brd[i].RtCh[j].InputLevel	int32	in mV
Brd[i].RtCh[j].InputEdge	int32	1=rising, 0=falling
Brd[i].RtCh[j].CFDPresent	int32	0=not present, 1=present
Brd[i].RtCh[j].CFDLevel	int32	in mV
Brd[i].RtCh[j].CFDZCross	int32	in mV
end of block		
The following is a T2/T3 Mode specific header		
ExtDevices	int32	external devices, bitwise coded
Reserved	int32	reserved
Reserved	int32	reserved
InpRate0	int32	input 0 count rate as displayed by rate meter
InpRate1	int32	input 1 count rate as displayed by rate meter
StopAfter	int32	stopped after this many ms
StopReason	int32	0=timeover, 1>manual, 2=overflow
NumRecords**	int32	number of T2 event records
ImgHdrSize**	int32	size of special header to follow before the records start
end of block		
The following is a special header that may not be present at all: for each i = 0 to <ImgHdrSize>		
Reserved[i]	uint32	reserved
end of block		
The following are the T2 records: for each i = 1 to NumRecords		
T2Record[i]	uint32	<p>The bit allocation in the record is, starting from the MSB:</p> <p>channel : 4</p> <p>time : 28</p> <p>The channel code 15 (all bits ones) marks a special record.</p> <p>Special records can be overflows or external markers. To differentiate this, the lower 4 bits of time must be checked.</p> <ul style="list-style-type: none"> <li>• If they are all zero, the record marks an overflow.</li> <li>• If they are &gt; = 1, they are external markers.</li> </ul> <p>See the demo programs for further details.</p>
end of block		

\* These items are stored as they were set / displayed last. They may not relate to the data stored.

\*\* These items may vary and thereby change the file structure dynamically. A program must interpret these before reading in further data.

See section 8.2.4 for the data types used. Note that the T2Records must be read sequentially so that time overflows can be corrected for, based on the overflow records. The software setup will install ready to use demos showing how to do this. There is a set of examples in C, Pascal/Delphi/Lazarus and Matlab for PicoHarp \*.pt2 file access. They can be found in the subfolder \Filedemo in your chosen installation directory. Note that these are only demos. The intention is not to routinely convert TTTR data to ASCII. This would be highly inefficient. It increases the file sizes enormously and adds no useful information. TTTR data is not really human readable. It requires application specific analysis software in any case and the demos should be used as a starting point for writing such software.

### 8.2.3. T3 Mode File Format

PicoHarp data files from T3 Mode (\*.pt3) are binary files which contain both the setup parameters and the actual event data. The setup data is in the file header which is in the same format as in standard interactive mode and T2 Mode files. This is to allow common reading routines for the headers. Note that this leads to some header values that are not meaningful in T3 Mode.

In order to safely recognize a T3 Mode file (apart from the file extension) a program reading in these files must read and interpret the value *MeasurementMode* which is 3 in T3 Mode. Format version 2.0 \*.pt3 files are structured as shown in the table below. Note that the format version is not necessarily the same as that of the creating software version.

Data Item	Type	Description
Ident	char [16]	the string 'PicoHarp 300'
FormatVersion	char [6]	currently '2.0' (since Sept 2006)
CreatorName	char [18]	for this software 'PicoHarp Software'
CreatorVersion	char [12]	currently '2.3.0.0' (April 2009)
FileTime	char [18]	the file creation date and time
CR/LF	char [2]	0x0D,0x0A carriage return, line feed
Comment	char [256]	any ASCII string
NumberOfCurves	int32	not meaningful in T3 Mode
BitsPerRecord	int32	32
RoutingChannels	int32	1 or 4
NumberOfBoards**	int32	reserved, now 1
ActiveCurve	int32	curve no. used for last measurement
MeasurementMode**	int32	0=interactive, 1=reserved, 2=T2, 3=T3
SubMode	int32	0=oscilloscope, 1=integration, 2=TRES
RangeNo*	int32	0=base resol., 1=x2, 2=x4, 3=x8, 4=x16, ....., 7=x128
Offset*	int32	offset in ns
AcquisitionTime*	int32	acquisition time in ms
StopAt	int32	stop at this many counts
StopOnOvfl	int32	stop if overflow occurred = 1, otherwise 0
Restart	int32	restart = 1, otherwise 0
DisplayLinLog*	int32	lin=0, log=1
DisplayTimeAxisFrom*	int32	lower time axis bound of display in ns
DisplayTimeAxisTo*	int32	upper time axis bound of display in ns
DisplayCountAxisFrom*	int32	lower count axis bound of display
DisplayCountAxisTo*	int32	upper count axis bound of display
The following block will appear in the file for each i = 1 to 8		
DisplayCurve[i].MapTo	int32	number of memory block to map this curve to
DisplayCurve[i].Show	int32	0=don't show, 1= show
end of block		
The following block will appear in the file for each i = 1 to 3		
Param[i].Start	float	used for automated measurements, e.g. TRES
Param[i].Step	float	reserved for automated measurements
Param[i].End	float	reserved for automated measurements
end of block		
RepeatMode	int32	reserved for automated measurements
RepeatsPerCurve	int32	reserved for automated measurements
RepeatTime	int32	reserved for automated measurements
RepeatWaitTime	int32	reserved for automated measurements
ScriptName	char[20]	reserved for automated measurements
The following block will appear in the file for each i = 1 to <NumberOfBoards> [currently this must be 1]		
Brd[i].HardwareIdent	char[16]	currently 'PicoHarp 300'
Brd[i].HardwareVersion	char[8]	currently '1.0' or '2.0'
Brd[i].HardwareSerial*	int32	hardware serial number
Brd[i].SyncDivider*	int32	input channel 0 (Sync) input divider

Brd[i].CFDZeroCross0*	int32	input channel 0 CFD zero cross level in mV
Brd[i].CFDLevel0*	int32	input channel 0 CFD discriminator level in mV
Brd[i].CFDZeroCross1*	int32	input channel 1 CFD zero cross level in mV
Brd[i].CFDLevel1*	int32	input channel 1 CFD discriminator level in mV
Brd[i].Resolution	float	resolution in ns
Brd[i].RouterModelCode	int32	0=none, 1=PHR_402, 2=PHR_403, 3=PHR_800
Brd[i].RouterEnabled	int32	1=enabled, 0=disabled (routing off)
The following block will appear in the file for each j = 1 to <NumberOfRouterChannels> [currently this must be 4]		
Brd[i].RtCh[j].InputType	int32	0=custom, 1=NIM, 2=TTL
Brd[i].RtCh[j].InputLevel	int32	in mV
Brd[i].RtCh[j].InputEdge	int32	1=rising, 0=falling
Brd[i].RtCh[j].CFDPresent	int32	0=not present, 1=present
Brd[i].RtCh[j].CFDLevel	int32	in mV
Brd[i].RtCh[j].CFDZCross	int32	in mV
end of block		
The following is a T2/T3 Mode specific header		
ExtDevices	int32	external devices, bitwise coded
Reserved	int32	reserved
Reserved	int32	reserved
InpRate0	int32	input 0 count rate as displayed by rate meter
InpRate1	int32	input 1 count rate as displayed by rate meter
StopAfter	int32	stopped after this many ms
StopReason	int32	0=timeover, 1>manual, 2=overflow
NumRecords**	int32	number of T3 event records
ImgHdrSize**	int32	size of special header to follow before the records start
end of block		
The following is a special header that may not be present at all: for each i = 0 to <ImgHdrSize>		
Reserved[i]	uint32	reserved
end of block		
The following are the T3 records: for each i = 1 to NumRecords		
T3Record[i]	uint32	<p>The bit allocation in the record is, starting from the MSB:</p> <pre> channel      : 4 dtime       : 12 nsync       : 16 </pre> <p>The channel code 15 (all bits ones) marks a special record.</p> <p>Special records can be overflows or external markers. To differentiate this, dtime must be checked.</p> <ul style="list-style-type: none"> <li>If it is zero, the record marks an overflow.</li> <li>If it is &gt;= 1, the individual bits are external markers.</li> </ul> <p>See the demo programs for further details.</p>
end of block		

\* These items are stored as they were set / displayed last. They may not relate to the T3 data stored.

\*\* These items may vary and thereby change the file structure dynamically. A program must interpret these before reading in further data.

See section 8.2.4 for the data types used. Note that the T3Records must be read sequentially so that time overflows can be corrected for, based on the special records. The software setup will install ready to use demos showing how to do this. There is a set of examples in C, Pascal/Delphi/Lazarus and Matlab for PicoHarp \*.pt3 file access. They can be found in the subfolder \Filedemo in your chosen installation directory. Note that these are only demos. The intention is not to routinely convert TTTR data to ASCII. This would be highly inefficient. It increases the file sizes enormously and adds no useful information. TTTR data is not really human readable. It requires application specific analysis software in any case and the demos should be used as a starting point for writing such software.

## 8.2.4. Data Types

The PicoHarp software is written in C++ and the data types used in the files correspond to standard C/C++ data types as follows:

char	byte, characters in ASCII
int32	32 bit signed integer
int64	64 bit signed integer
uint16	16 bit unsigned integer
uint32	32 bit unsigned integer
float	32 bit floating point number
time_t	a 32 bit time value as defined in most C libraries

Note that the formats for date and decimal point may depend on your Windows settings at run-time of the PicoHarp software (usually national language dependent).

Note also that on platforms other than x86 machines byte swapping may occur when the PicoHarp data files are read there for further processing. We recommend using the native x86 environment consistently.

The distribution pack includes a set of demo programs (source code) for various programming languages to show how access to PicoHarp data files can be implemented. They will be installed in the subfolder `\Filedemo` under the chosen installation folder.

## 8.3. Hardware Technical Data

### 8.3.1. Common Specifications

All information given here is reliable to our best knowledge. However, no responsibility is assumed for possible inaccuracies or omissions. Specifications and external appearance are subject to change without notice.

#### Input Channels

Trigger principle	Constant Fraction Discriminator
Trigger point	Falling edge
Input voltage operating range (pulse peak, 50 $\Omega$ ) optimum:	0 mV ... -1 000 mV -200 mV ... -400 mV
Input voltage max. range (damage level)	$\pm 2\,000$ mV
CFD level adjust	0 mV ... -800 mV
CFD zero cross adjust	0 mV ... 20 mV
Trigger pulse width	$\geq 0.5$ ns
Trigger pulse rise/fall time	$\leq 2$ ns
Max. count rate	$10.0 \cdot 10^6$ cps
Max. sync rate without divider	10.0 MHz
Max. sync rate with divider 1/8	84.0 MHz

#### Time to Digital Converter

Base resolution (min. time bin width)	4 ps
Timing uncertainty (rms)	$< 12 \text{ ps} + 10^{-8} \cdot \Delta t$
Time base calibration error	$< 50$ ppm
Differential non-linearity (full measurement range)	$< 5.0\%$ p.p. $< 1.0\%$ rms
Dead time	$< 95$ ns
Max. conversion rate	10.0 MHz

#### Histogrammer

Maximum number of time bins	65 536
Bin widths	4, 8, 16, ..., 512 ps
Measurement ranges	$65\,536 \cdot 4 \text{ ps} \dots 65\,536 \cdot 512 \text{ ps}$
Acquisition time	1 ms ... 100 h

#### TTTR Engine

T2 mode resolution	4 ps
T3 mode resolution	4, 8, 16, ..., 512 ps
FIFO buffer depth (records)	262 144
Sustained throughput (sum of channels)	typ. $5 \cdot 10^6$ events/s
Acquisition time	1 ms ... 100 h

**Operating Enviroments**

Ambient temperature	15°C ... 35°C 59°F ... 95°F
Warm-up period	≥ 20 min
Bus interface	USB rev. 2.0
Recommended PC specs	
CPU clock	≥ 1 GHz
RAM	≥ 512 MB
Operating system	Windows™ 2000/XP/Vista <sup>1)</sup>

**Power Consumption**

Line 100 V ... 240 V AC	≤ 25W
-------------------------	-------

**Dimensions**

Table top rack	310 × 235 × 140 mm
----------------	--------------------

**Retraction of old Devices**

Waste electrical products must not be disposed of with household waste. This equipment should be taken to your local recycling centre for safe treatment.



WEEE-Reg.-Nr. DE 96457402

<sup>1)</sup> Trademarks of Microsoft Corp.

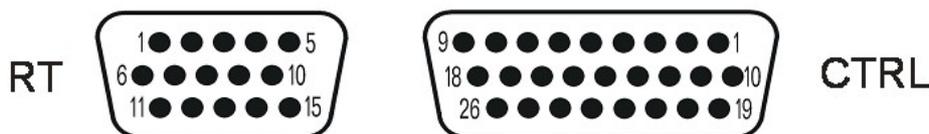
### 8.3.2. Router / Control Port Connectors

Note: The low cost model PicoHarp 300E does not have these connectors.

#### 8.3.2.1. PicoHarp 300 Hardware Version 1.0

The figure below shows the pin layout of the PicoHarp 300 router/control ports as of PicoHarp 300 hardware rev. 1.0. There is a 15-pin male sub-D connector labeled 'RT'. This is the connector primarily for the PHR 402/403 routers. The larger female 26 pin connector labeled 'CTRL' is for generic control signals.

Note that several signals are present on both connectors (wired through) but NOT on the same pin numbers.



PicoHarp 300 rev 1.0 router and control connectors, viewed from the outside of the chassis.

#### RT Connector

The 15-pin male sub-D connector labeled 'RT' is the connector for the PHR 402/403 routers. See section 5.3 on Routed Measurements for details on routing and router connection. It also carries marker signals for TTTR mode but these signals can also be fed in via the 'CTRL' connector. Therefore, if you wish to use routing together with marker input, the marker signals should be fed in via the 'CTRL' port. If necessary, a dedicated Y-cable for the simultaneous connection to the 'RT' port can be used (available from PicoQuant). Note that this cable must not be extended. Only required marker signals may be extended to connect the marker source

The following table shows the RT connector's pin assignments.

Pin #	I/O	Description
1	TTL in	count enable
4	TTL in	marker 1 input (TTTR mode only)
5	TTL in	marker 2 input (TTTR mode only)
9	TTL in	marker 0 input (TTTR mode only)
11	out	+ 5V $\pm$ 0.5V / $\leq$ 200 mA
12	GND	ground
14	GND	ground

Pin 1 accepts a TTL signal that can be used to externally enable/disable recording of incoming counts. Counting is enabled when pin 1 is held high. This mechanism can be used to blank out periods where e.g. an external scanner is moving to the next position. The count enable signal is effective in all modes. It may be used to synchronize the acquisition with external conditions. If external disabling is not desired the pin must be held high or left open. This signal is used by the router to identify valid counts. It can not be used by external applications if a router is connected.

Pins 4, 5 and 9 accept the TTL compatible synchronization signals that will be recorded as markers in TTTR mode. The pins are internally pulled down, so that they are inactive when left unconnected. The active edge is chosen in the software settings dialog. Rise/fall times must be 10 ns or faster. Both high and low state must be at least 100 ns long. The clock period may therefore (in principle) be as short as 200 ns but data bus throughput constraints will apply. Each marker creates an additional TTTR record, so that one must ensure not to swamp the data stream with too many marker records. Markers also take precedence over photon records, so that excess marker traffic can suppress photon records. In fast imaging applications it is therefore recommended not to use a pixel clock but a line clock only. Because each photon has a time tag, it is usually not necessary to use an additional pixel clock. Note that the same signals can also be fed in via the 'CTRL' connector, so that they can be used even if a router is connected.

Pins 11 provides a 5 V DC supply voltage external electronics can use. Under no circumstances must this line be shorted or loaded in excess of the specified maximum current. Note that the maximum current must be shared with any loads connected at the 5V line of the 'CTRL' connector.

Pins 12 and 14 are the common ground for the TTL signals and the DC supply line. They are also connected to the chassis.

All other pins are used for communication with dedicated accessories (PHR 40x routers etc.) and must not be connected otherwise. Appropriate cables are available from PicoQuant.

### CTRL Connector

While the RT connector is primarily for the connection of routers, the larger female 26 pin connector labeled 'CTRL' is for generic control signals such as markers in TTTR mode. The following table shows its pin assignments.

Pin #	I/O	Description
1, 9, 10	GND	ground
2, 8	out	+ 5V $\pm$ 0.5V / $\leq$ 200 mA
3, 7	out	+ 3.3V $\pm$ 0.5V / $\leq$ 200 mA
12	TTL in	marker 1 input (TTTR mode only)
11	TTL in	marker 2 input (TTTR mode only)
17	TTL in	marker 0 input (TTTR mode only)
18	TTL in	count enable
26	TTL out	ctc active

Pins 1, 9 and 10 are the common ground for the TTL signals and the DC supply lines. They are also connected to the chassis.

Pins 2 and 8 provide a 5 V DC supply voltage external electronics can use. Under no circumstances must this lines be shorted or (in total) loaded in excess of the specified maximum current.

Note that the maximum current on the 5 V line must be shared with any loads connected at the 5 V line of the 'RT' connector.

Pins 3 and 7 provide a 3.3 V DC supply voltage external electronics can use. Under no circumstances must this lines be shorted or (in total) loaded in excess of the specified maximum current.

Pin 18 accepts a TTL signal that is used by the router to mark counts as valid. It can also be used to externally enable/disable recording of incoming counts. Counting is enabled when pin 1 is held high. This mechanism can be used to blank out periods where e.g. an external scanner is moving to the next position. The count enable signal is effective in all modes. It may be used to synchronize the acquisition with external conditions. If external disabling is not desired the pin must be held high or left open. This pin can not be used if a router is connected.

Pins 11, 12 and 17 accept the TTL compatible synchronization signals that will be recorded as markers in TTTR mode. The pins are internally pulled down, so that they are inactive when left unconnected. The active edge is chosen in the software settings dialog. Rise/fall times must be 10 ns or faster. Both high and low state must be at least 100 ns long. The clock period may therefore (in principle) be as short as 200 ns but data bus throughput constraints will apply. Each marker creates an additional TTTR record, so that one must ensure not to swamp the data stream with too many marker records. Markers also take precedence over photon records, so that excess marker traffic can suppress photon records. In fast imaging applications it is therefore recommended not to use a pixel clock but a line clock only. Because each photon has a time tag, it is usually not necessary to use an additional pixel clock.

Pin 26 is a TTL output that signals when the ctc is active, i.e. when a measurement has started. Note that this is a standard TTL output, so it can NOT drive 50  $\Omega$  loads.

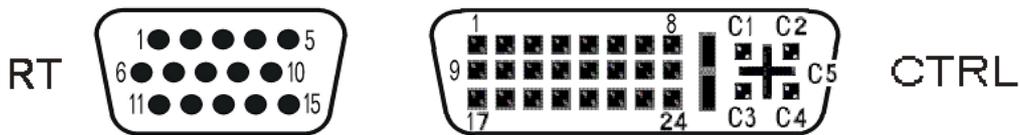
All other pins are used for communication with dedicated accessories or future extensions and must not be connected otherwise.

### 8.3.2.2. PicoHarp 300 Hardware Version 2.0

For the benefit of technical progress, the PicoHarp hardware is undergoing further development. Revision 2.0 has two new connectors. One is a 9 pin connector for CAN bus devices. The other is a new differential routing and control port connector that replaces the old 26 pin control port connector. It carries the routing information as differential signals, thereby improving signal integrity and reducing crosstalk.

The CAN connector is designed for PicoQuant accessories, such as monochromator/stepper motor controllers. It must not be used for any other purposes. If custom CAN nodes are to be controlled, suitable software support is also required, so that we recommend contacting PicoQuant for a tailored solution.

The figure below shows the pin layout of the PicoHarp 300 router/control ports as of PicoHarp hardware rev. 2.0. There is a 15-pin male sub-D connector labeled 'RT'. This is the connector primarily for the PHR 402/403 routers. The larger DVI-I connector labeled 'CTRL' is for generic control signals as well as routing with the PHR 800. Note that several signals are present on both connectors (wired through) but NOT on the same pin numbers.



PicoHarp 300 rev 2.0 router and control connectors, viewed from the outside of the chassis.

#### RT Connector

The 15-pin male sub-D connector labeled 'RT' is the connector for the PHR 402/403 routers. See section 5.3 on Routed Measurements for details on routing and router connection. It also carries marker signals for TTTR mode. Therefore, if you wish to use a PHR 40x together with marker input, a dedicated Y-cable for the simultaneous connection to the 'RT' port can be used (available from PicoQuant). Note that this cable must not be extended. Only required marker signals may be extended to connect the marker source. In principle, these signals can also be fed in via the 'CTRL' connector, but a special DVI plug will be needed.

The following table shows the RT connector's pin assignments.

Pin #	I/O	Description
1	TTL in	count enable
3	TTL out	ctc active
4	TTL in	marker 1 input (TTTR mode only)
5	TTL in	marker 2 input (TTTR mode only)
9	TTL in	marker 0 input (TTTR mode only)
11	out	+ 5V ± 0.5V / ≤ 200 mA
12	GND	ground
14	GND	ground

Pin 1 accepts a TTL signal that is used by the router to mark counts as valid. It can also be used to externally enable/disable recording of incoming counts. Counting is enabled when pin 1 is held high. This mechanism can be used to blank out periods where e.g. an external scanner is moving to the next position. The count enable signal is effective in all modes. It may be used to synchronize the acquisition with external conditions. If external disabling is not desired the pin must be held high or left open. This signal is used by the router to identify valid counts. It can not be used by external applications if a router is connected.

Pins 3 is a TTL output that indicates that the ctc is active (measurement running)

Pins 4,5 and 9 accept the TTL compatible synchronization signals that will be recorded as markers in TTTR mode. The pins are internally pulled down, so that they are inactive when left unconnected. The active edge is chosen in the software settings dialog. Rise/fall times must be 10 ns or faster. Both high and low state must be at least 100 ns long. The clock period may therefore (in principle) be as short as 200 ns but data bus throughput constraints will apply. Each marker creates an additional TTTR record, so that one must ensure not to swamp the data stream with too many marker records. Markers also take precedence over photon records, so that excess marker traffic can suppress photon records. In fast imaging applications it is therefore recommended not to use a pixel clock but a line clock only. Because each photon has a time tag, it

is usually not necessary to use an additional pixel clock. Note that the same signals can also be fed in via the 'CTRL' connector, so that they can be used even if a router is connected.

Pins 11 provides a 5 V DC supply voltage external electronics can use. Under no circumstances must this line be shorted or loaded in excess of the specified maximum current. Note that the maximum current must be shared with any loads connected at the 5 V line of the 'CTRL' connector.

Pins 12 and 14 are the common ground for the TTL signals and the DC supply line. They are also connected to the chassis.

All other pins are used for communication with dedicated accessories (PHR 40x routers etc.) and must not be connected otherwise. Appropriate cables are available from PicoQuant.

### CTRL Connector

While the RT connector is primarily for the connection of legacy routers, the larger DVI-I connector labeled 'CTRL' is for connecting the PHR 800 router. It also carries generic control signals such as markers in TTTR mode. The following table shows its pin assignments. Since most of them require low voltage differential signals (LVDS), it is recommended not to use this port for custom control signals.

Pin #	I/O	Description
1, 2	LVDS in	route 3
3	GND	ground
8	TTL out	ctc active
9, 10	LVDS in	route 2
11	GND	ground
14	out	+ 5V $\pm$ 0.5V / $\leq$ 200 mA
15	GND	ground
16	TTL in	router detection
17, 18	LVDS in	route 1
19	GND	ground
22	GND	ground
23, 24	LVDS in	count enable
C1	TTL in	marker bit 0 input (TTTR mode only)
C2	TTL in	marker bit 1 input (TTTR mode only)
C3	TTL in	marker bit 2 input (TTTR mode only)
C4	TTL in	marker bit 3 input (TTTR mode only)
C5	GND	ground

connector: DVI-I 24+4 (Molex 0743232033), available e.g. from Farnell

All ground pins are common ground for the TTL signals and the DC supply lines. They are also connected to the chassis.

Pins 14 provides a 5 V DC supply voltage external electronics can use. Under no circumstances must this line be shorted or (in total) loaded in excess of the specified maximum current. Note that the maximum current on the 5 V line must be shared with any loads connected at the 5 V line of the 'RT' connector.

Pin 8 is a TTL output that indicates that the ctc is active (measurement running). Note that this is a standard TTL output, so it can NOT drive 50  $\Omega$  loads.

Pin 16 is pulled high by the PHR 800 to indicate that routing signals are to be accepted through the CTRL connector rather than the RT connector.

Pins 23 and 24 accept a LVDS signal that is used by the router to mark counts as valid. It can also be used to externally enable/disable recording of incoming counts. This mechanism can be used to blank out periods where e.g. an external scanner is moving to the next position. The count enable signal is effective in all modes. It may be used to synchronize the acquisition with external conditions. These pins can not be used if a router is connected.

Pins C1 through C4 accept TTL compatible synchronization signals that will be recorded as markers in TTTR mode. The pins are internally pulled down, so that they are inactive when left unconnected. The active edge is chosen in the software settings dialog. Rise/fall times must be 10 ns or faster. Both high and low state must be at least 100 ns long. The clock period may therefore (in principle) be as short as 200 ns but data bus throughput constraints will apply. Each marker creates an additional TTTR record, so that one must ensure not to swamp the data stream with too many marker records. Markers also take precedence over photon records, so that excess marker traffic can suppress photon records. In fast imaging applications it is therefore recommended not to use a pixel clock but a line clock only. Because each photon has a time tag, it is usually not necessary to use an additional pixel clock.

All other pins are used for communication with dedicated accessories or future extensions and must not be connected otherwise.

All information given in this manual is reliable to our best knowledge. However, no responsibility is assumed for possible inaccuracies or omissions. Specifications and external appearance are subject to change without notice.



PicoQuant GmbH  
Unternehmen für optoelektronische Forschung und Entwicklung  
Rudower Chaussee 29 (IGZ), 12489 Berlin, Germany  
Telephone: +49 / (0)30 / 6392 6560  
Fax: +49 / (0)30 / 6392 6561  
e-mail: [info@picoquant.com](mailto:info@picoquant.com)  
WWW: <http://www.picoquant.com>