# picoquant: Software for decoding PicoQuant data formats

Thomas Bischof
tbischof@mit.edu

May 15, 2017

## 1 Picoquant

### 1.1 Purpose

This program decodes binary data from the Picoquant software and outputs the data in human-readable format. Currently, picoquant supports most modes and versions of the Timeharp, Picoharp, and Hydraharp. The board and mode are detected automatically, and if the mode is not supported an error message will explain the details.

Modes and software versions supported:

- Timeharp: v20 (thd), v30 (thd, t3r), v50 (thd), v60 (thd, t3r)
- Picoharp: v20 (phd, pt2, pt3)
- Hydraharp: v10 (hhd, ht2, ht3)

### 1.2 Command-line syntax

#### 1.2.1 Input

The input is either binary stream of data from the timing hardware, or the name of a file containing that data. The details of each format are too varied and verbose to be summarized here, but are laid out in section ??.

#### 1.2.2 Output

The output will either be written to  or the file specified. The output format depends on the mode of the data (interactive, t2, t3), with details given below.

**Interactive** Each record is of the form:

```
struct {
    int curve;
    double left_edge;
    double right_edge;
    int counts;
};
```

printed as:

```
curve number, left edge of time bin, right edge of bin, counts \n
```

**T2** Each record is of the form:

```
struct {
    int channel;
    long long time;
};
```

printed as:

```
channel number, time \n
```

**T3** Each record is of the form:

```
struct {
    int channel;
    long long pulse_number;
    int time;
};
```

printed as:

```
channel number, pulse number, time \n
```

## 1.3 Examples of usage

### 1.3.1 Reading header information

```
> picoquant --file-in data.phd --header-only
Ident = PicoHarp 300
FormatVersion = 2.0
CreatorName = PicoHarp Software
CreatorVersion = 2.3.0.0
FileTime = 14/05/11 17:55:48
Comment = Untitled
NumberOfCurves = 8
   ....
```

Reading these values is often a good way to check the integrity of a file, and to make sure that the correct settings are used in later processing. The keywords used here are consistent with those used in the documentation for the file type as provided by Picoquant, so common values like measurement resolution may not be identical across versions and boards.

### 1.3.2 Obtaining the resolution of a measurement

Resolution values must be multiples of 1 for most of the devices, but for the Timeharp they are integer divisions of 1, leading to non-integer multiples of 1. As such, all resolution values are presented as floats, even those which could be written as integers.

```
> picoquant --file-in data.phd --resolution-only
0,1.280000e+02
1,1.280000e+02
2,1.280000e+02
3,1.280000e+02
4,1.280000e+02
5,5.120000e+02
6,1.280000e+02
```

Interactive mode allows for a large number of curves, so the resolution report gives the resolution for each curve, by index.

```
> picoquant --file-in data.pt2 --resolution-only
1.280000e+02
```

### 1.3.3 Reading interactive data

```
> picoquant --file-in data.phd
0,0.000,0
...
3,4.482,4.484,2620
3,4.484,4.486,2595
3,4.486,4.488,2601
3,4.488,4.490,2604
3,4.490,4.492,2447
3,4.492,4.494,2407
3,4.494,4.496,2418
...
```

### 1.3.4 Reading T2/T3 data

```
> picoquant --file-in data.pt2
0,7128264
0,20957636
0,33684532
```

```
0,36576452
0,42146280
0,42251400
0,65787700
0,75149552
0,86537580
0,109288316
> picoquant --file-in data.pt3
1,103,47360
1,109,47616
1,115,85760
1,115,248832
1,213,55552
1,245,244992
1,254,49920
1,267,69888
1,268,122368
1,274,58624
```

#### 1.3.5 Translating T3 data to T2 data

```
> picoquant --file-in data.pt3 --to-t2
1,41248184
1,43648488
1,46086680
1,46249752
1,85257256
1,98246952
1,101651952
1,106872024
1,107324512
1,109660816
```

## 1.4 Mode- and hardware-specific information

While there are many differences between the formats of the files generated by the different boards and modes, they all follow a common structure:

- a general header identifying the board type and software version
- a board-specific header, identifying hardware and software configuration
- a mode-specific header
- data

As such, the process of streaming data can be broken down into the following steps:

1. Identify the board type (`picoquant.c`)
2. Identify the software version used to generate the file (`hydraharp.c`, `picoharp.c`, `timeharp.c`).
3. Determine the collection mode used (`hydraharp/hh_*.c`, ...).
4. (a) If the run is specified as resolution-only or header-only, print the appropriate values.

   (b) Otherwise, read through the remaining header information and print the data.

In this implementation, while the code used to produce the data and headers is very similar between software versions for a given board, there are some small differences which make a general program difficult to write. As such, each version is hard-coded, and any changes to the overall structure of the program must be rolled out to all versions. Fortunately, many common tasks such as printing of data are centralized in `picoquant.c`, so changes to the output format only require modification of a single function.

In principle, the code can be collected into a nicer data-streaming object which masks the translation process and yields only the resulting data stream. This is probably the most convenient way to deal directly with developing custom tools for data processing, but for most purposes passing the data through pipes should be sufficient. If such an interface is desired, the low-level translation functions (found in `*/*_v*.c`) should be sufficient when wrapped with higher-level logical routines like those used to determine the board identity and version. The function structure is uniform across all versions and boards, which should simplify the wrapping process.

The remainder of this section is devoted to a discussion of the details important to each board and measurement type. Most of this information can be found in the manuals included with the hardware, but there is a significant amount of information which is documented in more scattered locations, such as the sample data code. This summary includes the details vital to understanding how the raw data are actually translated into the general data streams, and how various design decisions affect the quality and precision of the result.

### 1.4.1 A word about external markers

Many of the timing boards have a feature which allows the insertion of an external timing pulse into the signal, for use in TTTR modes to designate a raster scan or other time-dependent behavior. These records are not handled directly by picoquant, but instead a message is passed to  indicating that such a record exists. If different behavior is desired (perhaps assignment of the marker to a non-existent channel), modify the function `external_marker` in `picoquant.c`. This will cause problems with the other programs, which assume every record originates from the true signal stream, so such an alteration must be accompanied by code which appropriately splices the signal stream, for example by halting collection into a histogram and initiation of a new one.

### 1.4.2 T3 timing carries units of histogram bins, not time

In T3 modes–including T3R for the Timeharp–the pulse dimension carries units of pulse number, but the time is not strictly a time with fixed units. Instead, the specified resolution of the interactive mode is the unit of time, such that the measurement is affected directly by this choice of resolution. As a result, the T3 record more directly reflects the index of the bin the record would have fallen into in interactive mode. In picoquant this value is converted to units of implicitly, but do be aware of this distinction.

### 1.4.3 Translation of T3 to T2 data

In principle, if the sync source of a T3-mode experiment arrives with perfectly uniform spacing, the pulse number can be said to represent some constant amount of time, and a T2-like record recovered. To do this, pass the flag `--to-t2` for an input of T3 data, but consider the implications of how this translation is performed (see section **??**).

In picoquant, the average rate of sync pulses is used to determine the time unit a pulse carries.

### 1.4.4 Timeharp

The Timeharp is the least sophisticated of the three boards discussed here, and is fundamentally a histogramming board. It features two input channels, and its three modes are:

- interactive (thd): Collection of a histogram, using channel 0 as the sync and channel 1 as the signal.
- continuous interactive (thc): Identical to normal interactive mode, except that the histogram is reported at the end of a user-specified time interval, repeating until halted. This feature is used for experiments requiring time resolution of a time-dependent feature, such as a fluctuating fluorescence lifetime. This feature is not supported by the existing software, due to lack of data or interest in using this mode (it is not present in the other timing boards).
- time-tagged time-resolved (t3r): TTTR mode, equivalent to T3 mode, with channel 0 as the sync and channel 1 as the signal. Instead of reporting the delay time directly, this mode reports the index of the histogram bin an event would fall into. The time delay represented by the delay can be recovered from the header information, but for homogeneity of the treatment of T3-like data this step is not performed automatically.

### 1.4.5 Picoharp

The Picoharp features two input channels. Its three modes are:

- interactive (phd): Standard interactive mode, using channel 0 as the sync and channel 1 as the signal.
- t2 (pt2): A T2 mode, with both channels treated equally and all pulse arrivals recorded.
- t3 (pt3): A T3 mdoe, with channel 0 as the sync and channel 1 as the signal.

In both the T2 and T3 mode, internal clocks of limited precision (<32 bits) are used to record the passage of time or pulses. As such, many records are devoted to recording the occurrence of an integer overflow. These records are treated by picoquant automatically, producing data streams of the form specified in section **??**.

### 1.4.6 Hydraharp

The Hydraharp features a dedicated sync channel and four input channels. Its three modes are:

- interactive (hhd): Standard interactive mode, with four separate histograms assigned to the four input channels.
- t2 (ht2): A T2 mode, with all channels treated equally, including the sync channel. If the sync channel is active, these events will be recorded, and for convenience picoquant will output each event as arriving on channel 4. This channel index can be altered by modifying value of the global variable `HH_SYNC_CHANNEL` (`hydraharp.h`) at compilation time.
- t3 (ht3): A standard t3 mode.

Future versions of the Hydraharp are being developed with extra input channels. If your model features extra channels, it should be sufficient to modify the value of `HH_SYNC_CHANNEL`, as the number of channels and modules is handled dynamically by picoquant.