Berlin, 7/24/2024

PMS-800

4 Channel Gated Photon Counter

USER MANUAL

Becker & Hickl

1 Revision history

Date	Version	Changes	Author
24.07.2024	1.0	Initial revision	Thomas Meures

2 Table of Contents

3 Introduction

The PMS-800 is a Gated Photon Counter and Multiscaler. It is the successor of the PMS-400 card with more channels, extra features and much improved performance parameters. In different operation modes, it can be utilized to count photons in 4 ns time-bins or larger at a rate of 800 MHz without imposing any dead time on the measurements and allows for large time scale TCSPC measurements with the new Triggered Accumulation Mode. The PMS-800 features a PCI Express bus to transfer either timing histograms, built in the PMS-800, or streamed events to a host computer at extremely high rates. *Table 3-1* shows a summarized comparison of the PMS-800 and its predecessor.

Feature	PMS-400	New PMS-800
inputs	2 channels, 2 gates	4 channels, 4 gates (1 flexible trigger-gate)
Max. count rate	800 MHz	800 MHz
Min. timing resolution	250 ns	4 ns
Dead-time	50 ns for each bin	none
Max. event stream count	32768	unlimited
PC interface	PCI	PCI Express Gen 2

Table 3-1 Performance Parameters of the PMS-800 and PMS-400 in comparison.

4 Related Resources

There are various documents and software packages available for download at www.becker-hickl.com on the PMS-800 Product page.

- The PMS-800 Datasheet summarizes the performance parameters and electrical characteristics of the PMS-800.
- The Connection Scheme provides a summarized picture of the PMS-800 frontend.
- The BH-PMS800 GUI software allows for quick measurement setup and recording with the PMS-800. It comes with full control of the PMS-800 and a range of data analysis and processing features to cover most experimenters needs.
- The bhPy package supplies a python wrapper for the PMS-800 DLL and some coding examples for Python developers.
- The **PMS-800 DLL** with software reference gives developers full control over the PMS-800 functionality and allows to integrate the card into your own measurement software. The DLL is included with the BH-PMS800 GUI software, the bhPy package or can be retrieved by contacting info@becker-hickl.com.
- The **Processing DLL** with software reference provides functionality to unpack the PMS data-stream for further analysis. The DLL is included with the BH-PMS800 GUI software, the bhPy package or can be retrieved by contacting info@becker-hickl.com.
- Please contact info@becker-hickl.com for software examples and any questions regarding the PMS-800.

5 PMS-800 Functionality

5.1 Input Description and control

The PMS-800 features 4 timing and counting channels accompanied by an associated gate input for each channel. These gates allow the user to limit pulse counting to desired time windows and avoid background signals. The fourth gate can be simultaneously used as experiment trigger to either start a measurement or to reset the time base in the Triggered Accumulation mode. *Table 5-1* shows the

channels with their function, as they are ordered on the front panel of the PMS-800. The Connection Scheme document illustrates this interface in further details.

Channel function	Front panel label	DLL channel ID
Counting Channel 1	1	0
Gate 1	G	4
Counting Channel 2	2	1
Gate 2	G	5
Counting Channel 3	3	2
Gate 3	G	6
Counting Channel 4	4	3
Gate 4 / Trigger Input	G T	7

Table 5-1 Channel function and front panel labels description.

When a channel is **enabled**, incoming pulses are counted at rates of up to 800 MHz into time-bins, according to their arrival time relative to the beginning of a measurement. To register the pulses, discriminators are used which can be set to detect rising or falling edges at **threshold** levels between -500 mV and +500 mV. The rising or falling edge can be selected by using the **polarity** setting.

For the gates, the **polarity** setting determines if they are considered open when the input signal is above the set **threshold** or when below.

The trigger is registered in the same way as the 4 counting channels when crossing the threshold in the selected direction.

For the counting channels, the user can choose between 3 **input-modes**:

- Mode 0: All incoming pulses are registered according to specifications.
- Mode 1: Pulses are only registered when the associated gate is open.
- Mode 2: An internal calibration signal will be used as channel input instead of the discriminator pulses. This is mostly useful during software development and testing. The **calibration signal** needs to be enabled in a separate setting.

Relevant settings for input channels	Related PMS-800 DLL function
Enabled/Disabled	`set_channel_enable()`
Threshold	`set_input_threshold()`
Polarity	`set_channel_polarity()`
Input-mode	`set_channel_inputmode()`
Enable calibration signal	`set_pulsgenerator_enable()`

Table 5-2 Relevant settings for the input channels with their related DLL function.

5.2 Data acquisition settings

The PMS-800 can be operated in different modes and measurements can be taken with varying time-bin sizes, measurement duration, trigger settings and others.

The main setting to decide on for each measurement is which **measurement mode** to use. The user has a choice between a rather general **Event Streaming** mode and more performance optimized modes like the **Multiscaler** mode and the **Triggered Accumulation** mode. Further details about the measurement modes are given in *Section 5.3*. Depending on the selected mode, some settings are predefined and cannot be changed by the user. *Table 5-4* illustrates the availability of settings in the different measurement modes.

The user has multiple choices to start and stop a measurement. The PMS-800 is armed for measurement by using the `run_data_collection()` function from the PMS-800 DLL. The user can require an **external trigger** to start the measurement**,** when a well-defined start time is desired. If no external trigger is required, the measurement will start as soon as possible after above command has been executed. To stop the measurement, the user can either specifically send a **stop measurement** command or use a software timer in `run_data_collection()` to determine the measurement duration. Furthermore, the hardware has an internal **hardware counter**, which can be used to set the measurement duration with extremely high precision. This counter is limited to a time of approximately 17 s. A fourth way to determine the end of a measurement is to require a **fixed count of triggers** as an exit condition. When an external trigger signal is supplied, the PMS-800 can be configured to count the triggers and to stop a measurement at the desired number.

The user also needs to configure the Time to Digital converter (TDC) before starting a measurement. Generally, incoming hits are counted into time-bins of a given size and with a well-defined edge time relative to the beginning of the measurement. The bin-size and number of bins can be selected for each measurement. A detailed explanation of these parameters is given in *Section 5.3*.

Relevant data acquisition settings	Related PMS-800 DLL functions
Measurement mode	`set_measurement_configuration()`
Start a measurement	`run_data_collection()`
External trigger	`set_external_trigger_enable()`
Stop a measurement	`stop_measurement()`
Hardware counter to determine measurement duration	`set_hardware_countdown_enable()` `set_hardware_countdown_time()`
Trigger counter to determine measurement duration	`set_trigger_countdown_enable()` `set_max_trigger_count()`

Table 5-3 Relevant settings for the data acquisition with their related DLL function.

5.3 Measurement Modes

The PMS-800 can be operated in 3 different modes: The **Event Streaming** mode, the **Multiscaler** mode and the **Triggered Accumulation** mode. The Event Streaming mode is the most versatile mode of operation while the Multiscaler and Triggered Accumulation modes are optimized for maximum performance in various measurement situations.

For each measurement, the user typically has the choice of a **time range** and a **resolution (number of bins)** for the latter. This ultimately dictates the **time-bin size** in the Multiscaler and Triggered Accumulation modes. In the Event Streaming mode, one instead specifies the time-bin size, while the time range is virtually unlimited.

The minimum bin size is 4 ns. The maximum size, as well as the bin count and time range, depends on the measurement mode. The possible parameter settings are shown in *Table 5-4*.

Setting/Parameter	Event Streaming mode	Multiscaler mode	Triggered Acc. mode
External trigger	available	available	Always enabled
Hardware counter	available	used if < time range	available
Trigger counter	available	used if < time range	Available
Time range	Not applicable	256 ns to 1.07 s	256 ns to 67 ms
Resolution (bin count)	Not applicable	64 to 65536	64 to 4096
Bin size	4 to 128 ns	dictated by time range and resolution (4 to 16384 ns)	dictated by time range and resolution (4 to 16384 ns)

Table 5-4 Data acquisition and TDC settings and their availability in different measurement modes

5.3.1 Event-Streaming Mode

The Event-Streaming Mode allows to stream incoming pulses in so-called events. The events contain information about the amount of pulses which were detected within the selected time-bin, the start time of the time-bin relative to the beginning of the measurement and the channel ID of the originating channel. Events are only generated, if the hit count exceeds a set **event count threshold** value. This threshold allows the user to specify a minimum count between 1 and 127 for time-bins to be recorded as events by the PMS-800. This can help to avoid streaming a high number of background noise events and save memory and bandwidth. *Figure 5-1* illustrates the generation of events from input pulses.

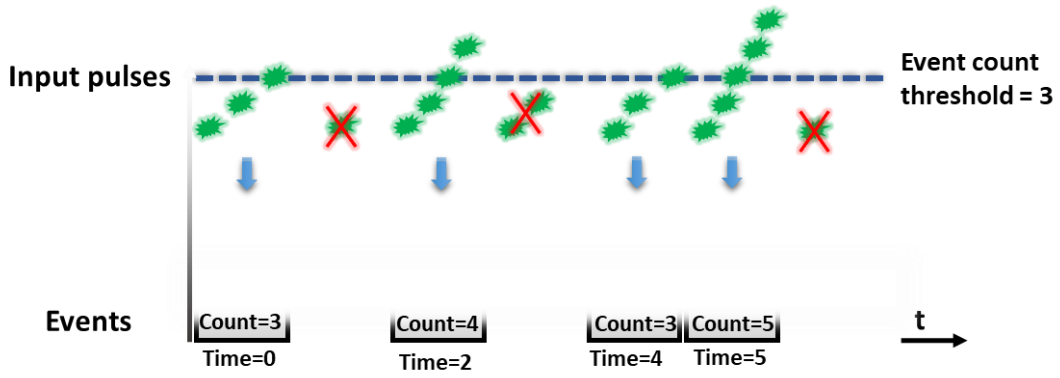

Figure 5-1 Schematic view of events generated from input pulses in the Event Streaming mode. Event Times are given in multiples of the time-bin size.

The above-mentioned information of time, channel ID and hit count is encoded in a 16-bit data package for each event. The exact protocol is illustrated in *Table 7-1*. The Processing DLL provides functionality to unpack the data protocol and convert it into easily usable arrays.

To stop a measurement in Event Streaming mode, the user either needs to provide a software run-time through the PMS-800 DLL, send the stop measurement command through the DLL or use one of the hardware exit conditions as described in *Section 5.2*.

The data transfer rate in Event Streaming mode is only limited by the PCI Express bus and allows for a total event transfer rate of 180 MHz for all channels combined. At the smallest bin sizes this can infer a limit on the rate of incoming pulses and the Multiscaler mode might be a better choice for the given measurement, since it does not have limitations on the count rate or bin-size.

The number of produced events per channel in the Event Streaming mode is limited by the time-bin size and **not necessarily equals the input count-rate**. High input count-rates in principle only lead to high counts in the time-bins. However, a high count-rate can lead to a high occupancy of time-bins, meaning that the maximum event-rate is more likely reached with high input counts.

If at a time-bin size of 4 ns, all time-bins are occupied with pulses, events will be produced at a rate of 250 MHz which leads to data loss. In this case, the input counts need to be limited such that not all time-bins are occupied and events are being generated at a lower rate of 180 MHz. The same is true, for 8 ns and 16 ns bins when more than one or two channels respectively are recording pulses. *Table 3-1* illustrates, which pulse rates can be supported depending on the selected time-bin size in Event Streaming mode. For a bin size of **32 ns and larger**, the peak-rate of 800 MHz is always supported for all channels.

Time-bin size	Maximum event rate produced per channel	Possible number of channels running at 800 MHz input rate	remaining event transfer bandwidth
4 ns	250 MHz	None	180 MHz
8 ns	125 MHz	1	55 MHz
16 ns	62.5 MHz	2	55 MHz
32 ns	**31.25 MHz**	**4 (all channels)**	**N/A**
>32 ns	**<31.25 MHz**	**4 (all channels)**	**N/A**

Table 5-5 The allowed channel rates for different bin-sizes. The channel input rates need to be adjusted for a channel to not generate events exceeding the transfer bandwidth.

Relevant settings for the Event Streaming mode	Related PMS-800 DLL functions
Time-bin size	`set_measurement_configuration()`
Event count threshold	`set_event_count_thresholds()`

Table 5-6 Relevant settings for the Event Streaming mode with their related DLL function.

5.3.2 Multiscaler Mode

In the Multiscaler Mode, up to 65536 time-bins can be recorded with sizes from 4 to 16384 ns. This allows for a total measurement time range of up to 1.07s. The data cannot be filtered but it can be recorded at peak rate of 800 MHz for any bin size settings for all channels in parallel without loss.

The data acquisition is automatically stopped when the time range is expired. There is no need for another stop mechanism. It can however be used, if a measurement duration shorter than the time range is desired.

This mode is appropriate for measurements, where a high resolution is required and extremely high average input rates are expected at each channel. Timing histograms are constructed directly in the PMS-800 card. The recorded data will be transferred as compressed histograms to the computer and does not require a large bandwidth. The exact protocol is depicted in *Section 7.1*. The Processing DLL provides functionality to unpack the compressed data and convert it into easily usable arrays.

Relevant settings for the Multiscaler mode	Related PMS-800 DLL functions
Time range	`set_measurement_configuration()`
Resolution (bin count)	`set_measurement_configuration()`

Table 5-7 Relevant settings for the Multiscaler mode with their related DLL function.

5.3.3 Triggered-Accumulation Mode

This mode is in principle a specialized version of the Multiscaler Mode which allows to restart measurements upon triggers coming in at a rate of up to 50 MHz and to accumulate the measured timing distributions of following events at peak rate on the PMS-800 card. This allows to perform TCSPC like measurements for time ranges of up to 67 ms in up to 4096 time-bins. The external trigger is always enabled in this mode and the user only needs to select the trigger edge (**polarity**) and **threshold** for input **7**.

In this mode, the measurement will always start on the external trigger, but the user can choose between all exit conditions, described in *Section 5.2* to terminate the data acquisition.

Relevant settings for the Triggered Accumulation mode	**Related PMS-800 DLL functions**
Time ramge	`set_measurement_configuration()`
Resolution (bin count)	`set_measurement_configuration()`

Table 5-8 Relevant settings for the Triggered Accumulation mode with their related DLL function.

6 Software Control

There are two ways to control the PMS-800 via software. One can use the BH-PMS800 GUI software which has full control of the PMS-800 card and provides easy access to all settings explained above. Furthermore, it comes with a wide range of data handling capabilities. The software, together with a user guide can be downloaded at www.becker-hickl.com.

A second option is to use the PMS-800 DLL, which gives the user direct control over the card and allows for integration with their own data acquisition software.

Becker & Hickl provides the software reference, readymade Python wrappers and example code in C for the DLL.

7 Data Protocol

The data protocol changes depending on the mode, the card is operated in. As mentioned above, the user can choose between three different operating modes: Multiscaler, Triggered Accumulation and Event Streaming mode. The incoming data will be structured as follows.

7.1 In Multiscaler and Triggered accumulation mode

The histograms accumulated in hardware in the two modes are transferred in 32bit vectors. The first 3 vectors contain header information as depicted in the table below.

Vector	Name	Bits					
1	Header 1	31 – 28: Channel ID	27 – 24: Transfer condition	23-20: Time-rollover	19 – 7: N_bins	6 – 4: 0x0	3 – 0: MSB_idx = N_bits-1
2	Header 2	31 – 24: N_occ		23 – 12: 0x0 (Reserved)		11 – 0: N_data	
3	Header 3	31 – 0: event count					
4 → 4+(N_occ-1)	Occupancy array	Occupancy vectors (big endian)					
4+N_occ → 4+N_occ+(N_data-1)	Data vectors	Data vectors (big endian)					
4+N_occ+N_data	Final Padding 1	31 – 0: 0xffffffff					

5+N_occ+N_data (optional)	Final Padding 2 (optional)	31 – 0: 0xffffffff

Header 1:

1. **Channel ID:** The channel number associated with the transmitted data.
2. **Transfer condition:** The condition which triggered the transfer of the histogram. This could be:
    a. Bit3: Due to a trigger
    b. Bit2: Due to the end of the measurement
    c. Bit1: Due to a time-rollover (Multiscaler mode only)
    d. Bit0: Reserved
3. **Time-rollover:** During a Multiscaler measurement, these are the bits 15-12 of the time measurement for the transferred memory block. NOTE: The memory blocks can only hold 4096 bins each. A measurement, containing more bins will be segmented into the appropriate number of 4096-bin blocks. There can be up to 16 blocks.
4. **N_bins:** The number of occupied bins in the transferred memory block.
5. **MSB_idx:** This is the index of the MSB of the maximum bin count. The total number of necessary bits to encode the value in the memory block is MSB_idx+1.

Header 2:

1. **N_occ**: The number of occupancy vectors, to be transferred (see below for details).
2. **N_data**: The number of data vectors to be transferred (see below for details).

Header 3: The total event count in the transferred memory block.

The memory content is compressed before transfer. An array of occupancy bits is sent first to indicate which memory addresses have a content larger than 0. This array has one bit for each memory location and is packed into 32bit vectors.

After that, the content of those memory bins which are not empty is transferred, encoded with the bits necessary to contain the maximum count in the histogram. This data is also packed into 32bit vectors. The last vector is padded with zeros if necessary.

Finally, one or two padding vectors of 0xffffffff are sent in the end of the transfer. These are making sure that every transfer contains an even number of 32bit vectors, which simplifies the memory interface usage. If the number of vectors in the package is even, two padding vectors are sent, otherwise one suffices.

7.1.1 Example

If a 64-bin memory of channel 0 has been transferred upon an incoming trigger and has the following hit content:

Bin 10	Count = 10
Bin 15	Count = 5
Bin 20	Count = 3
Bin 25	Count = 1
Bin 29	Count = 9
All other bins are empty	

The three header vectors will be:

Vector	Bits					
1	31 – 28: 0x0	27 – 24: 0x4	23-20: 0x0	19 – 7: 0x5	6 – 4: 0x0	3 – 0: 0x3
2	31 – 24: 0x2		23 – 12: 0x0		11 – 0: 0x1	
3	31 – 0: 0x1C					

The occupancy array will be 2 x 32bit vector (Big endian):

Vector	Bit 31	Bit 30	…																										Bit 0
4	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The data will be encoded with 4 bits (10 = 1010) and we have 5 bins with content. The data vectors will be:

Vector	Bits 31-20	Bits 19-16	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0
6	0x0 (Padding)	0x9	0x1	0x3	0x5	0xA

Now there will be two padding vectors since we have an even number of 32-bit vectors in the package:

Vector	Bits 31-0
7	0xffffffff
8	0xffffffff

This completes the transfer.

7.2 In Event Streaming Mode

In this mode, “events” and Macrotime-Overflows are transferred individually as 16bit vectors with the following structure:

Type	**Header: 15 - 12**				**Bin count: 11 – 5**	**Event time: 4 – 0**
Event	MTOF	GAP	Channel_ID[1]	Channel_ID[0]	Count in given time bin Is always >0	Event time after last MTOF
MTOF	1	GAP	0	0	0x0	0x0

Table 7-1 The structure of a 16-bit event package in Event Streaming mode

1. **Header:**
   - **MTOF:** Indicates a Macrotime-Overflow. Such an event will be sent regularly after 32 times the set bin width has passed and is used to keep track of the event time.
   - **GAP:** Indicates if the data transfer has been interrupted and the time offset of all following events is not guaranteed anymore.
   - **Channel_ID:** The channel in which the transferred Bin count has been recorded
2. **Bin count** indicates the number of hits, occurring within the transferred time bin. Up to 127 hits per bin can be stored in this mode.

3. **Event time:** The elapsed time T_i since the last Macrotime-Overflow.

With the above information, the timing of every event can be calculated relative to the beginning of the measurement by counting the events with MTOF set high and adding the event time as follows

$$T_{event} = N_{MTOF} \cdot 32 + T_i.$$

T_{event} is expressed in multiples of the set bin-width.