fastAdc Documentation

Release 0.1

CAS

Jul 30, 2025

Contents

Contents:

CHAPTER 1

Fast ADC Application

The Fast ADC application was developed to capture raw data of all ten ADCs channels and to process the information of pulse shaped signals. Each ADC signal is individually process according to the parameters configured by the user and the resulting data available via registers. In addition, 5 virtual channels are available which can be configure to be the sum or subtraction of two (real) ADC channels.

The application was developed under the Simulink environment, using the XFEL Simulink Library. As such, the application can be fully simulated in the Matlab environment using experimental data.

Tests of the system under experimental conditions have indicated that the device can reliably acquire 108000 raw samples, which corresponds to a time window of 1ms, from all 10 channels simultaneously.

In this documentation an overview of the hardware device (taken by the manual by EEE) is presented, as well as its setup in the configuration editor of the Karabo device. For more detail information on the board, firmware and performance results, please visit Fast Electronics Digitizer Overview page.

This chapter will cover the mininum requirements for the device to be operational, which is a **Clock** and **Trigger** signal.

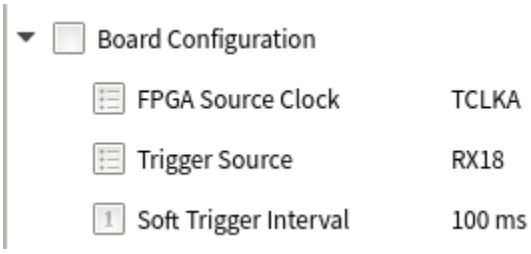

Fig. 1: FastADC Clock and Trigger Karabo parameters

1.1 Clock Sources

The FastADC can accept clock signals from the following sources:

- TclkA (MicroTCA backplane)
- TclkB (MicroTCA backplane)
- Internal Clock (125 MHz)
- Front SMA Connector
- Front Hardlink Connector
- RJ45 SIS8900 RTM Connector

In XFEL, 99% of the FastADC setups use the TclkA or TclkB source, since these lines have a clock signal provided by the Timing Board, which is phase sync with the laser operation at XFEL.

1.2 Trigger Sources

A trigger signal is required to start the operation of the Fast ADC Application. The Fast ADC firmware supports thirteen trigger sources, which can be combine to a single signal: eight from the MLVDS lines in the MicroTCA backplane, four from the Harlink connector, four from SIS8900 RTM RJ45 connector as well as an internal one.

Again, 99% of the XFEL FastADC setups use a trigger source from the MicroTCA Backplane, which is provided by the Timing system.

CHAPTER 2

Raw data

The fast ADC saves raw data from all 10 ADC channels simultaneously. The raw data is saved in the DDR memory available in the SIS8300 board. Some parameters can be tuned by the user to steer the data acquisition, and are presented in the figure below.

The amount of raw data saved with each trigger signal is configured in the *Number of raw samples* property in the Karabo device. It is possible to delay the raw data acquisition by a fixed number of samples after the trigger signal. This parameter is set by adjusting the *Raw Delay* property.

It is also possible to define the period of raw data acquisition (save one sample every N samples), which provides a zoom capability when observing raw data signals. As an example, if this register is set to 1, 2 or 3, the device will show an ADC value only every 2, 3 or 4 samples, respectively. The period of raw data acquisition is configured in the parameter *Skip Samples*. Notice that the parameters delay and period affect all ADCs signals raw data acquisition.

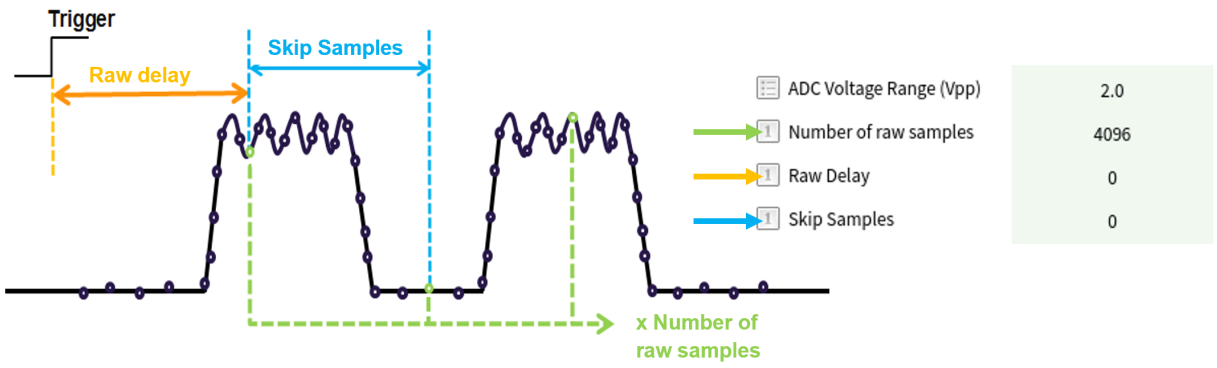

Fig. 1: Example of an ADC signal and configuration parameters for raw data acquisition. The correspondent Karabo parameters are shown on the right.

The previous parameter affects the raw data acquisition for all channels. Nevertheless, to acquire raw data from a specific channel, the correspondent *Enable Raw Data* parameter must be set to TRUE in the channel options. Notice that a *Signal Description* can also be define by the user to easily identify the signal present in that channel.

Fig. 2: Signal description and Enable Raw Data Karabo parameters for Channel 0.

2.1 ADC Range

The ADC range can be configured to be 1.25, 1.5, 1.75 or 2.0 Voltage peak-to-peak. The setting can be changed when the device is not acquiring (STOP state), there is no need to re-initialize the device.

ADC range options.

2.2 Time Axis

A time axis property is available for plotting **Vector XY Graphs** with the Raw data. The values take into account the *Number of raw samples*, *Frequency* and *Skip Samples* values configure in Karabo.

2.3 Virtual Channels

The FastADC includes 5 virtual channels (channel 10 through 14) which can be configured to be the sum or subtraction of two (real) ADC channels. The same features are available in these channels as any other ADC channels (peak integration, bunch pattern peak integration, multibaseline, adc alert, moving average, data voltage conversion, etc.).

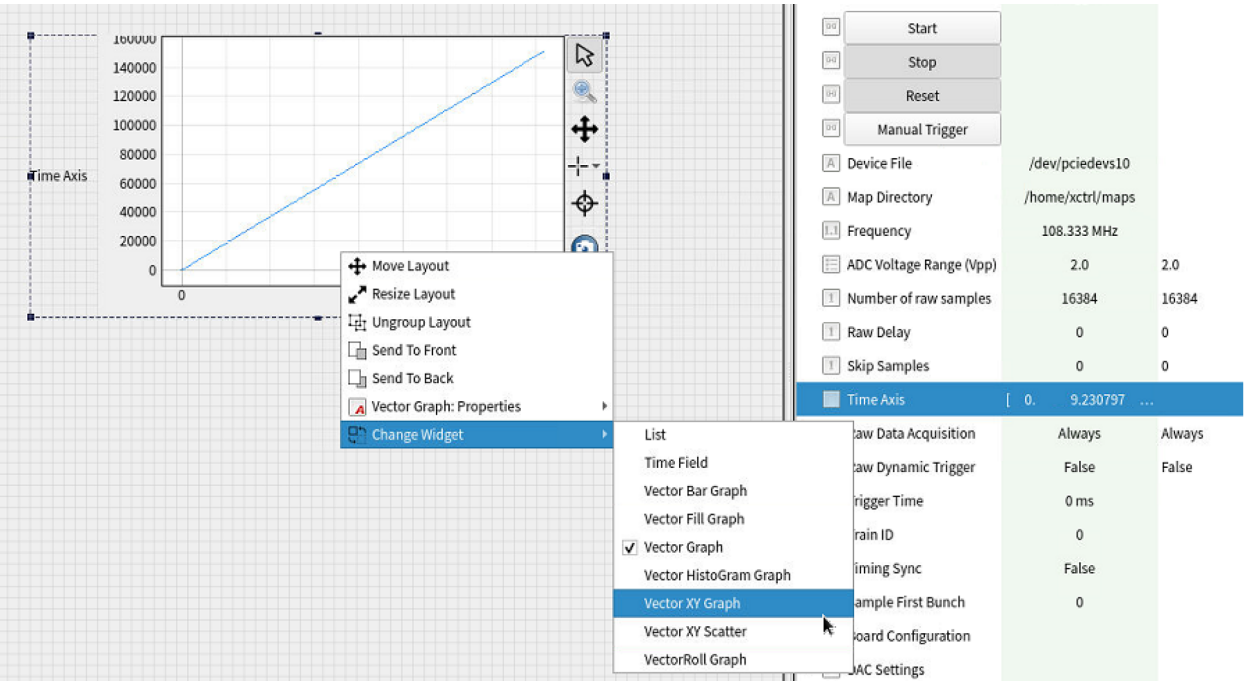

Fig. 3: Time Axis property show in Karabo and the Vector XY Graph option in the Widget. After selecting this option, simply drag the Time axis parameter on top of the widget for the X axis to be updated with the time values.

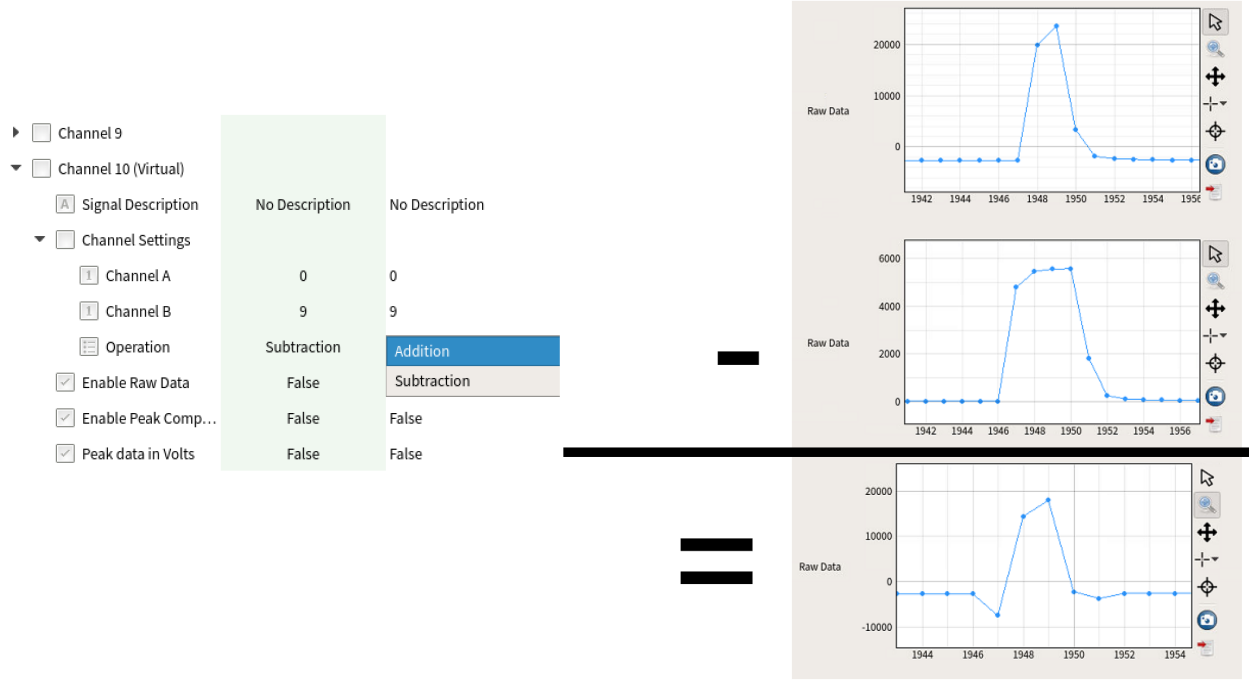

Fig. 4: Configuration of Channel 10 (Virtual). The channel is configure to be subtraction of the ADC signal in channel 0 and channel 9. The raw data graphs are display on the right.

CHAPTER 3

Integrator Processing Module

The fast ADC application includes, per ADC channel, an integrator processing module that can calculate the peak values of periodic ADC signals, or can be configured to integrate based on the current bunch pattern as described in the *Bunch Pattern Decoding* section. Each module can be individually configured. The following functionalities are available per ADC channel.

3.1 Convert data to voltage units

Per channel, the Raw and Peak Integration data can be displayed in Voltage levels in Karabo, which taking into account the *ADC Range* configuration. Take note that **the raw values are always saved in the DAQ**.

3.2 ADC Alert

An ADC alert can be configured per channel, which will be raised if the ADC signal goes above/below the user-specified threshold. To clear the alert, either disable it or reconfigure the threshold.

There is a *Global ADC alert* property which is true if any ADC channel alert is raised. To see which channel raised the alert, check the values in *ADC Channels Alert*.

3.3 Signal Integration

To enable an Integrator module for a specific ADC channel, *Enable Peak Computation* must be set to True for that channel. Once enabled, the module waits for a trigger signal to start the calculation.

Peak signal calculation starts with the sample where the trigger signal is detected. It is possible to delay the calculation by a specific amount of data samples, by writing the desired value in the *Pulse Delay* property. For each peak, the module sums up as many ADC samples as specified in the *Peak Samples* property. The *Number of pulses* property configures the number of pulses to process after receiving a trigger, while the *Pulse Period* property specifies the number of samples expected between pulses (thus disentangling between pulses).

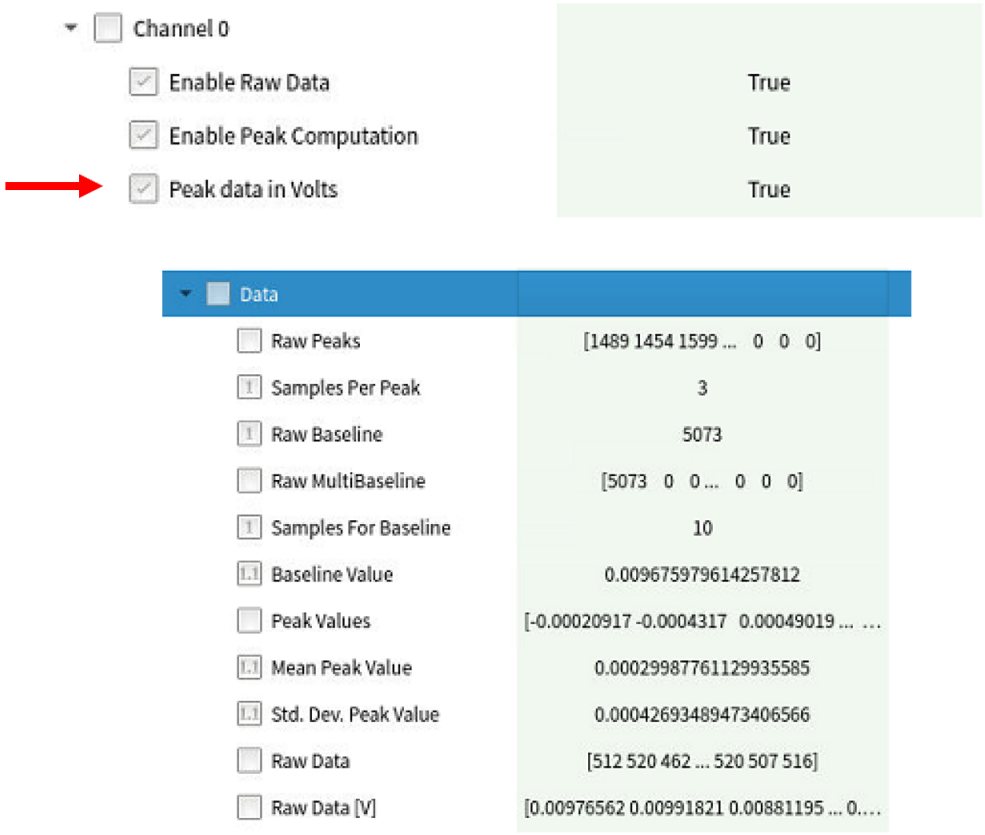

Fig. 1: Property to enable conversion of data to Voltage levels. Notice that all **Raw** properties are not converted.

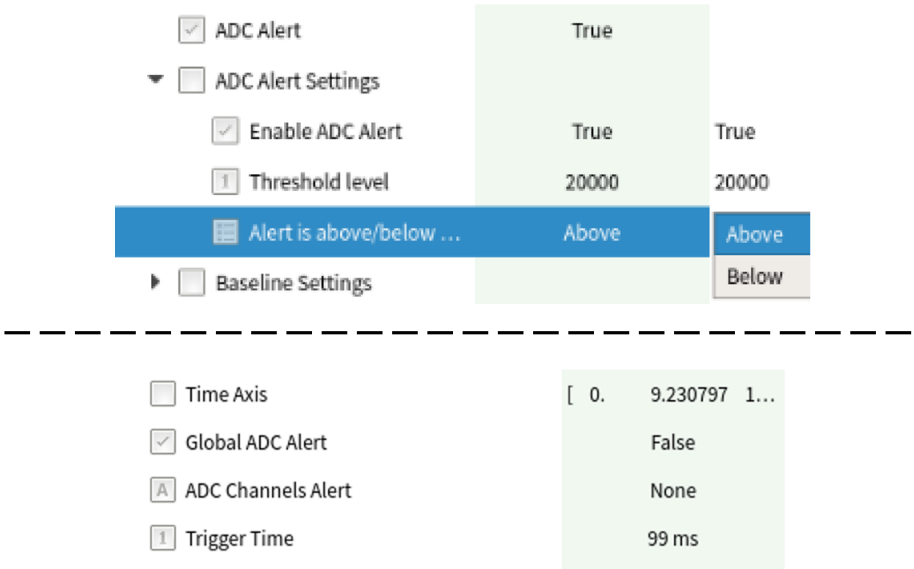

Fig. 2: ADC Alert Settings per channel (top) and Global ADC Alert (bottom).

The calculated values are available in the output channel *Channel X* > *Output* > *Schema* > *Data*. Basic statistics are calculated in Karabo based on these values, including the mean and standard deviation of the peak values. The hardware also provides *Max. ADC Sample* vector, which contains the sample with highest (absolute) ADC value for each integrated peaks.

Another hardware calculated value, **which is not saved in the output channel**, is the *Measured Peak range* (located below the *Pulse Period* parameter). This value shows the difference (in counts or voltage) between the highest and lowest integrated peak.

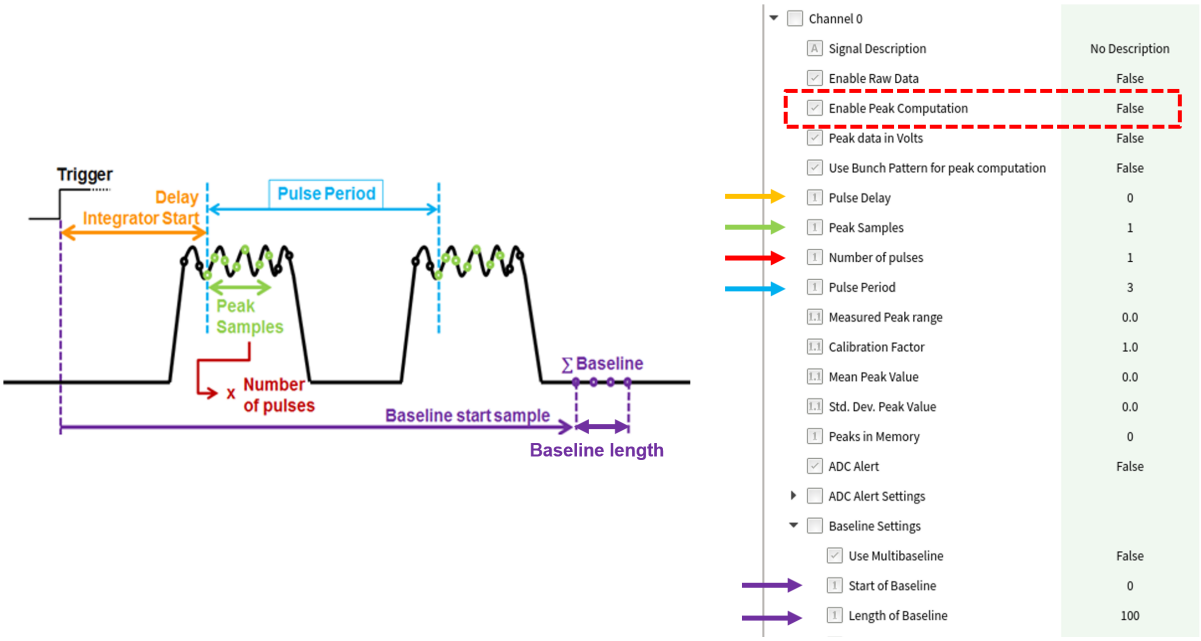

Fig. 3: Example of an ADC signal and of the configuration parameters for the integrator module.

3.4 Baseline Configuration

The integrated values and statistics are calculated taking into account a baseline value. Multiple options for this value are available in the **Baseline Settings** node.

If a fixed baseline is desired, the value of the baseline should be input in the *Fixed Baseline* property, and the *Enable fixed Baseline* boolean should be set to true. Otherwise, the baseline will be calculated over a section of the signal.

A signal based Baseline can be calculate in the following ways:

Standard :: A single baseline value is calculated for the entire train. The *Start of Baseline* value delays the baseline calculation by a set value after the trigger signal.

Dynamic :: This setting is only available when using **Bunch Pattern** for Integration (see *Bunch Pattern Decoding* section). A single baseline value is calculated for the entire train. The *Start of Baseline* value delays the baseline calculation by a set value before the *Sample First Bunch*.

Multi value :: A baseline value is calculated for every pulse integrated. The *Start of Baseline* value delays the baseline calculation by a set value before the first sample used for integrated the pulse.

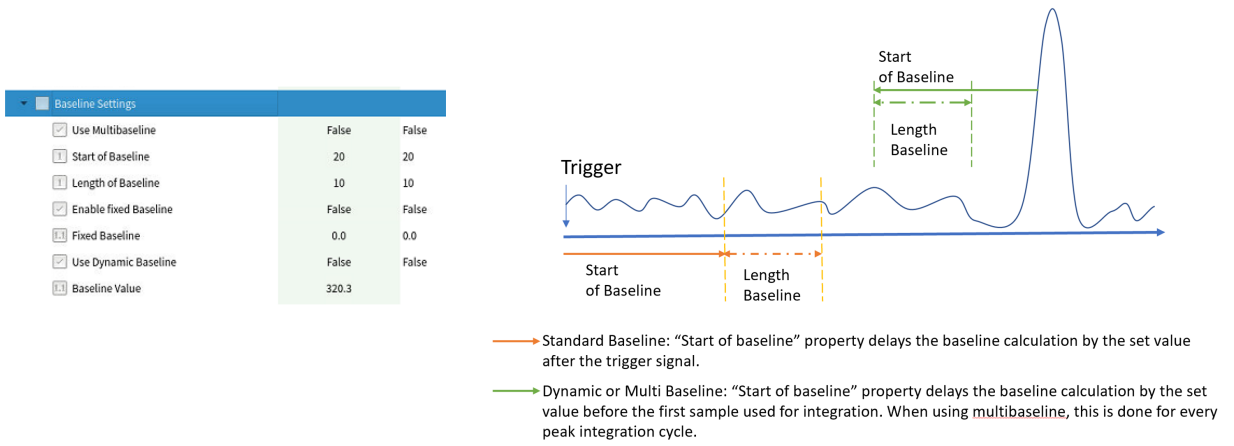

Fig. 4: Baseline calculation and related Karabo Parameters

3.5 Moving Average

The fast ADC firmware implements, per ADC channel, a 128 step moving average filter. The output of this filter, available in the *Moving average settings* node under the name **Moving average**, gives an indication of the order of magnitude of the ADC baseline value. To enable this filter, *Enable Moving average* must be set to '1'.

The firmware also provides values concerning the latest train of pulses received, which are referred to as train statistics. These values get updated whenever a new trigger signal is received. The calculated values are:

- Pulse delay (number of samples between trigger and first pulse),
- Minimum pulse width (in samples),
- Minimum pulse period (in samples),
- Number of pulses in last train.

and are presented in the figure here below:

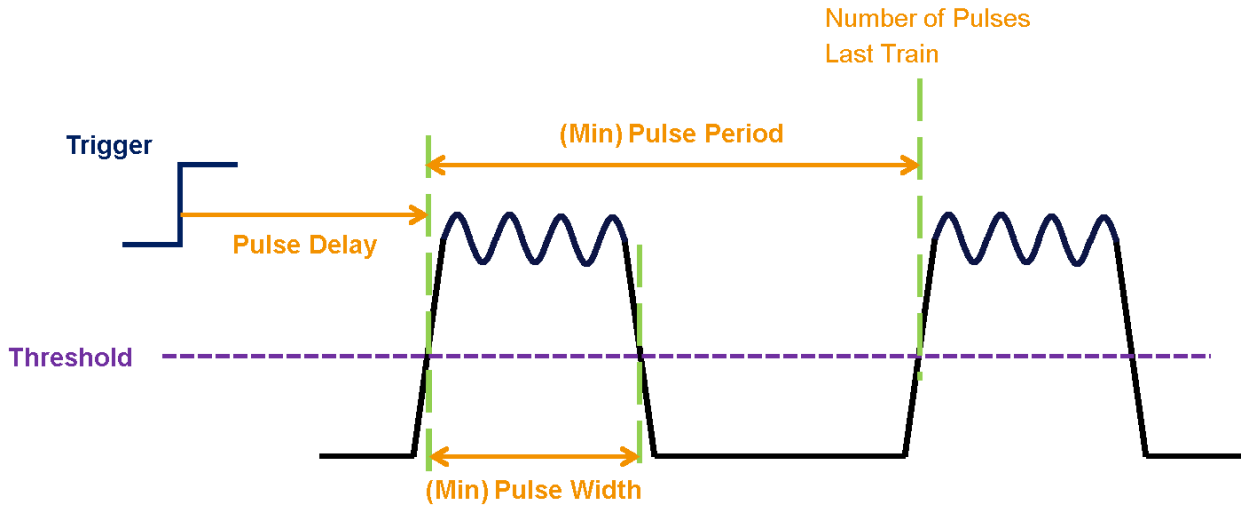

Fig. 5: Values calculated in each train.

To calculate these values, the moving average filter must be enabled and a threshold value configured in the 'ADC Threshold magnitude' property. Whenever the ADC signal goes above or below the threshold value (depending on whether the Moving average value), the firmware starts calculating the aforementioned values.

CHAPTER 4

Bunch Pattern Decoding

If desired, the FastADC can receive and decode the bunch pattern to know which bunches in a train are going to a specific beamline and/or have a pulse probe laser (PPL). Users can also specify a Max/Min of acceptable bunch charges. When enable and configure, the Bunch IDs and Charges are saved in the DAQ.

In addition, it is also possible to configure the FastADC to use this information to do *Automatic Peak Integration* or *Conditional and/or Dynamic Raw acquisition*.

4.1 Settings

The decoding configuration is done in the device node **Bunch Pattern Settings**. The source of the X-ray bunch pattern to decode (*Light Source* parameter) can be any of SASE1, SASE2, SASE3, SASE1+3 or None. If bunch pattern decoding is enabled and None is selected, only the PPL bunch pattern will be considered. The Bunch pattern logic option defines whether a bunch ID is considered when there are bunches in both the X-ray beam and the PPL patterns (AND) or when there are bunches in either (OR).

Maximum and Minimum bunch charges can also be define.

If configure, during acquisition the Karabo device will update the *First Bunch ID* and *Number of Bunches* parameters in the current Train.

4.2 Output

An output channel is also available, wherein a list containing the of bunch Ids determined by the conditions specified in the “Bunch Pattern Settings” node are output. This is DAQ compatible, but if DAQ recording is required it must be requested separately from the slow data of the device (request “<DeviceId>:bunchPatternNode.output” to be added to the data group.

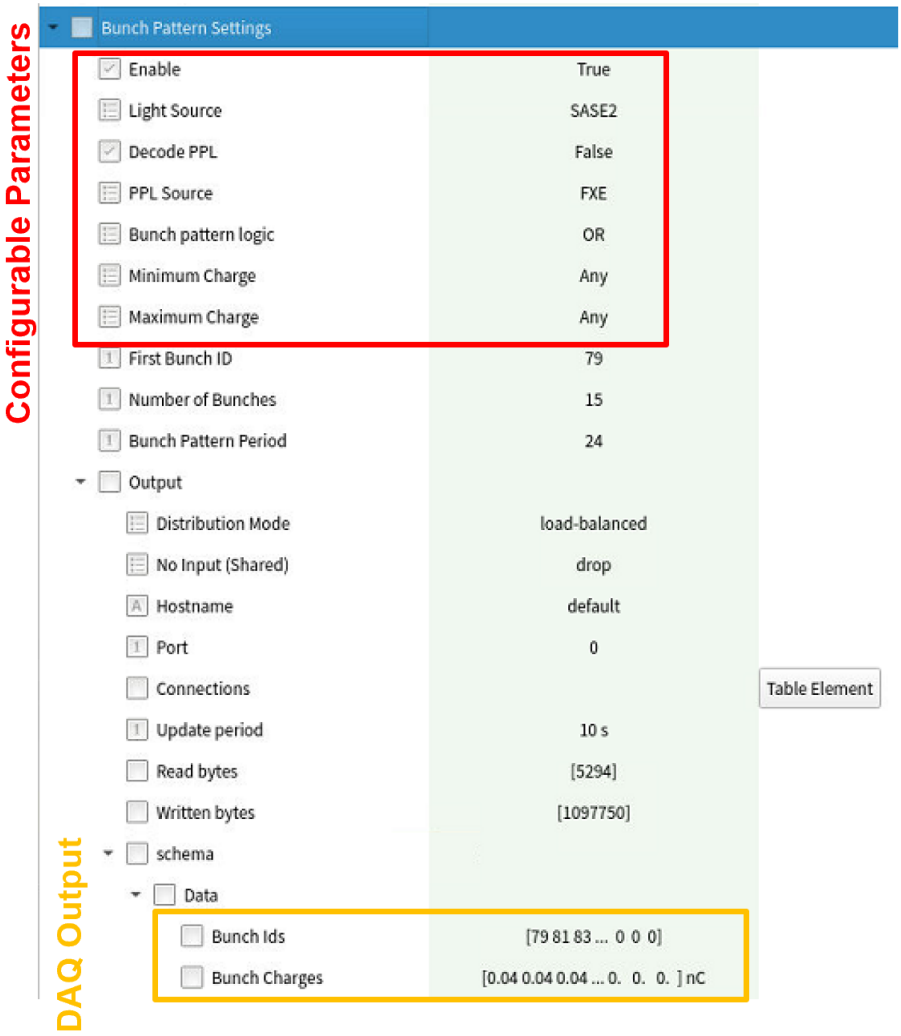

Fig. 1: Bunch pattern settings in the FastADC and correspondent Output channel.

4.3 FastADC Calibration

To use the features described in the following sections, it is require to enable and configure the Bunch Pattern and calibrate the **FastADC**. This is done performing the following steps:

1. Stop acquisition of device
2. Set *Raw Delay* and all *Peak Delay* parameters to 0
3. Configure the Bunch Pattern according to your requirements
4. Configure the *Pre Train Samples* parameter if require (see *Calibration for early bunches*)
5. Start acquisition. Take note of the value in the property *Sample First Bunch*
6. Open the **Karabo Trigger Middle Layer Device** of the trigger used by the FastADC (check *Board Configuration* > *Trigger Source* property. When in doubt, contact Control and/or EEE colleagues)
7. Configure **Macro P-Event** property to be *Standard Trigger*
8. Change the *Target Delay* so that first peak sample (as desired for the peak integration) of the first bunch in the raw trace matches the *Sample First Bunch* value

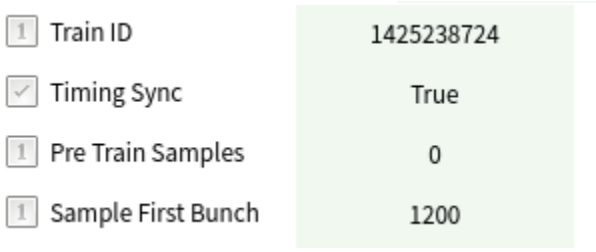

Fig. 2: Pre Train Samples and Sample First Bunch parameter.

4.3.1 Calibration for early bunches

For bunches which are present at the very beginning of the train, like PPL pulses, calibrating the FastADC following the previous section will result in a raw trace with very few samples before the train. This might not be desirable for setups which require data before the train arrives or use a signal based baseline(s).

To surpass this, the FastADC parameter *Pre Train Samples* can be used to configure an offset number of samples to acquire before the train.

4.4 Automatic Peak Integration

Once the FastADC is calibrated (see previous section), channels can enable the *Use the Bunch Pattern for peak computation* feature. In this configuration, the FastADC automatically updates the Peak Integration parameters to integrate all the Bunches present in the Train that match the configuration of the Bunch Pattern.

The parameters *Number of Pulses*, *Pulse Period* and *Pulse Delay* are ignore by the device, since these will be updated by the hardware. The user only needs to specify:

- how many samples per peak the device should consider
- baseline configuration (see Baseline Configuration section)

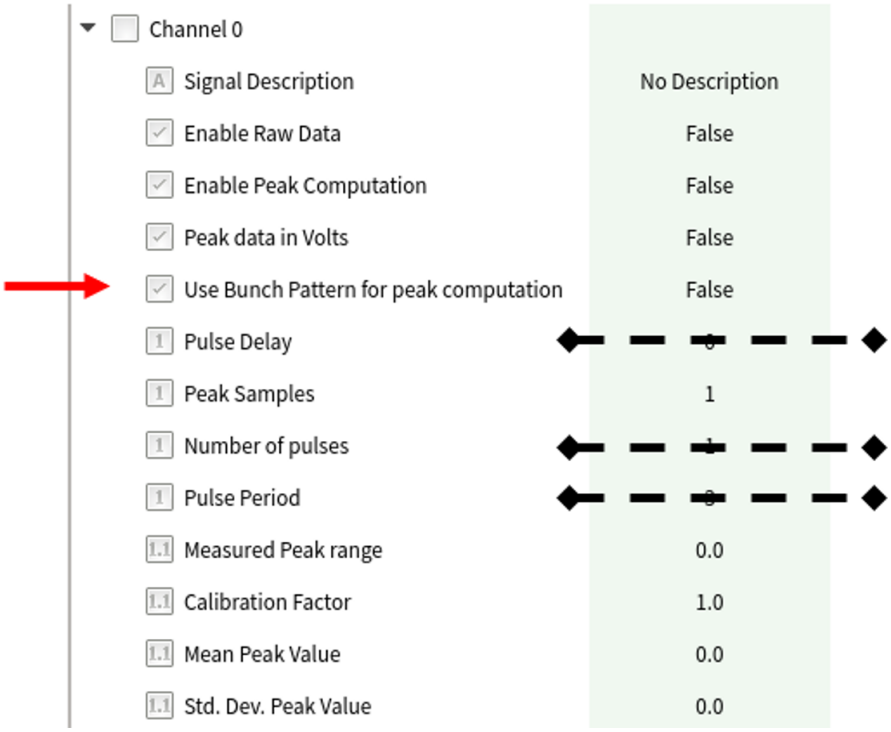

Fig. 3: Enable peak integration with the Bunch Pattern. Crossed are the parameters which are ignore in this setting

4.5 Conditional and/or Dynamic Raw acquisition

With the FastADC calibrated (see *FastADC Calibration*), the property *Raw Data Acquisition* can be set to **Conditional**, meaning that raw data will only be taken if there are bunches in the train which match the configuration of the Bunch Pattern (please note that this applies to **all channels**). This parameter can also be set to *Never*, in case the user is only interested in values from peak integration.

When *Raw Dynamic trigger* is true, raw data acquisition will always start at sample number (*Sample First Bunch - Raw Delay*), meaning that acquisition will have a fix relation (time wise) with the first bunch in the train. This is similar to use the dynamic trigger from the timing system.

Using these features have **no affect** in the Automatic Peak Integration.

Fig. 4: Conditional and Dynamic Raw Karabo properties

CHAPTER 5

Configure a *fastAdc* device

Please consider that the following configuration steps should be performed by experts; a wrong setting will result in a device not properly working.

5.1 Configuration step by step

- Configuration
- Set **Device File** to the latest version of **/dev/pciedevs?**. Now we have **pciedevs6** for SA2 and **pciedevs7** for SCS;
- Set **Map Directory** to **/home/xctrl/maps**. This folder contains the configuration xml-files of the firmware registers.
- Change **Board Configuration / FPGA Source Clock** to **TCLKA**.
- Instantiate the *fastAdc* device. Now **Train ID** should be updating. If not or the value is some unreasonable number, please contact AE.
- By default, the channels are closed. To enable a channel, set **Enable Peak Computation** and **Enable Raw Data** to **True**.
- After you have done the previous steps correctly, **Baseline Value** and **Mean Peak Value** should be updating. If you have a scene, then you should see some noise. However, if nothing for the channel is updating and there is only a flat line in the scene (a single number), you are trapped by a bug in the firmware. Some discussions can be found in the redmine ticket #28462. You can fix it contacting AE. Anyway, the latest device release provides in the configuration editor the option to reset the DDR2 memory (in the SIS8300 board, where the raw ADC data are stored) and/or the ADC chip, clearing the above mentioned firmware bug.

CHAPTER 6

FastAdc

6.1 Commands

Key	Displayed Name	Description	Alias	Access Level	Allowed States
dacNode.dacconfig	Update DAC parameters/memory	Update all DAC parameters in the hardware.		USER	ON
reset	Reset	Resets the device in case of an error		USER	ERROR
start	Start	Instructs device to go to started state		USER	ON
stop	Stop	Instructs device to go to stopped state		USER	ACQUIRING
trigger	Manual Trigger	Sends a software trigger to the hardware (always possible, independent of chosen trigger mode)		USER	ON

6.2 Properties

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_0.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_0.output_schema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_0.output_schema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_0.output_schema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_0.output_schema.data.peaks	Peak Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_0.output_schema.data.rawBaseline	Raw Baseline	Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_0.output_schema.data.rawData	Raw Data	Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_0.output_schema.data.rawPeaks	Raw Peaks	Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_0.output_schema.data.samplesForBaseline	Samples For Baseline	Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_0.output_schema.data.samplesPerPeak	Samples Per Peak	Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_0.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_0.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_1.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_1.output_schema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_1.outputSchema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_1.outputSchema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_1.outputSchema.data.peaks	Peaks Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_1.outputSchema.data.rawBaseline	Raw Baseline	Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_1.outputSchema.data.rawData	Raw Data	Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_1.outputSchema.data.rawPeaks	Raw Peaks	Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_1.output_schema.data.samplesForBaseline	Samples For Baseline	Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_1.output_schema.data.samplesPerPeak	Samples Per Peak	Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_1.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_1.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_2.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_2.output_schema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_2.output_schema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_2.output.schema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_2.output.schema.data.peaks	Peak Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_2.output.schema.data.rawBaseline	Raw Baseline	Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_2.output.schema.data.rawData	Raw Data	Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_2.output.schema.data.rawPeaks	Raw Peaks	Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_2.output.schema.data.samplesForBaseline	Samples For Baseline	Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_2.output.schema.data.samplesPerPeak	Samples Per Peak	Number of samples per peak		UInt32	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_2.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_2.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_3.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_3.outputSchema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_3.outputSchema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_3.outputSchema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_3.output.peaks	Peak Values	schema.data.peaks Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_3.output.rawBaseline	Raw Baseline	schema.data.rawBaseline Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_3.output.rawData	Raw Data	schema.data.rawData Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_3.output.rawPeaks	Raw Peaks	schema.data.rawPeaks Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_3.output.samplesForBaseline	Samples For Baseline	schema.data.samplesForBaseline Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_3.output.samplesPerPeak	Samples Per Peak	schema.data.samplesPerPeak Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_3.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_3.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_4.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_4.outputSchema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_4.outputSchema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_4.outputSchema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_4.outputSchema.data.peaks	Peak Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_4.output.rawBaseline	Raw Baseline	data.rawBaseline Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_4.output.rawData	Raw Data	data.rawData Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_4.output.rawPeaks	Raw Peaks	data.rawPeaks Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_4.output.samplesForBaseline	Samples For Baseline	data.samplesForBaseline Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_4.output.samplesPerPeak	Samples Per Peak	data.samplesPerPeak Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_4.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_4.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_5.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_5.outputSchema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_5.outputSchema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_5.outputSchema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_5.outputSchema.data.peaks	Peak Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_5.outputSchema.data.rawBaseline	Raw Baseline	Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_5.outputSchema.data.rawData	Raw Data	Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_5.output.rawPeaks	Raw Peaks	fastAdcSchema.data.rawPeaks Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_5.output.samplesForBaseline	Samples For Baseline	fastAdcSchema.data.samplesForBaseline Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_5.output.samplesPerPeak	Samples Per Peak	fastAdcSchema.data.samplesPerPeak Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_5.peak.mean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_5.peak.std	Std. Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_6.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_6.output.baseline	Baseline Value	fastAdcSchema.data.baseline Baseline Value.		Double	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_6.output.peakMean	Mean Peak Value	data.peakMean Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_6.output.peakStd	Std Dev. Peak Value	data.peakStd Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_6.output.peaks	Peak Values	data.peaks Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_6.output.rawBaseline	Raw Baseline	data.rawBaseline Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_6.output.rawData	Raw Data	data.rawData Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_6.output.rawPeaks	Raw Peaks	data.rawPeaks Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_6.output.schema.data.samplesForBaseline	Samples For Baseline	Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_6.output.schema.data.samplesPerPeak	Samples Per Peak	Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_6.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_6.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_7.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_7.output.schema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_7.output.schema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_7.output.schema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_7.output.schema.data.peaks	Peak Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_7.output.schema.data.rawBaseline	Raw Baseline	Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_7.output.schema.data.rawData	Raw Data	Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_7.output.schema.data.rawPeaks	Raw Peaks	Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_7.output.schema.data.samplesForBaseline	Samples For Baseline	Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_7.output.schema.data.samplesPerPeak	Samples Per Peak	Number of samples per peak		UInt32	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_7.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_7.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_8.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_8.outputSchema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_8.outputSchema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_8.outputSchema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_8.output.peakValues	Peak Values	channelSchema.data.peaks Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	
channel_8.output.rawBaseline	Raw Baseline	channelSchema.data.rawBaseline Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_8.output.rawData	Raw Data	channelSchema.data.rawData Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_8.output.rawPeaks	Raw Peaks	channelSchema.data.rawPeaks Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_8.output.samplesForBaseline	Samples For Baseline	channelSchema.data.samplesForBaseline Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_8.output.samplesPerPeak	Samples Per Peak	channelSchema.data.samplesPerPeak Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_8.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_8.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	
channel_9.baseline	Baseline Value	Baseline Value.		Float	OBSERVER	READONLY	
channel_9.output_schema.data.baseline	Baseline Value	Baseline Value.		Double	OBSERVER	READONLY	
channel_9.output_schema.data.peakMean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Double	OBSERVER	READONLY	
channel_9.output_schema.data.peakStd	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Double	OBSERVER	READONLY	
channel_9.output_schema.data.peaks	Peak Values	Vector of all peak values (with base line correction).		VectorFloat	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_9.output.rawBaseline	Raw Baseline	data.rawBaseline Sums of baseline values from hardware		UInt32	OBSERVER	READONLY	
channel_9.output.rawData	Raw Data	data.rawData Raw data from ADC.		VectorUInt16	OBSERVER	READONLY	
channel_9.output.rawPeaks	Raw Peaks	data.rawPeaks Sums of raw samples of selected peaks		VectorUInt32	OBSERVER	READONLY	
channel_9.output.samplesForBaseline	Samples For Baseline	data.samplesForBaseline Number of samples in rawBaseline		UInt32	OBSERVER	READONLY	
channel_9.output.samplesPerPeak	Samples Per Peak	data.samplesPerPeak Number of samples per peak		UInt32	OBSERVER	READONLY	
channel_9.peak.mean	Mean Peak Value	Mean of the Peak pulse (with base line correction).		Float	OBSERVER	READONLY	
channel_9.peak.std	Std Dev. Peak Value	Standard deviation of the Peak pulse values (with base line correction).		Float	OBSERVER	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
dacNode.dacData	DAC Data	DAC data memory.		VectorInt32	OBSERVER	READONLY	
dacNode.dacVoltageData	DAC Voltage Data	DAC data memory converted to Voltage		VectorDouble	OBSERVER	READONLY	
progress	Progress	The progress of the current action		Int32	OBSERVER	READONLY	
trainId	Train ID	Current train ID as read from the FPGA		UInt64	OBSERVER	READONLY	
triggerTime	Trigger Time	Time between Triggers		Int32	OBSERVER	READONLY	
triggerTimeStat	Trigger Histogram	Histogram of time between Triggers		VectorUInt16	OBSERVER	READONLY	
connection.brokers	Brokers	Brokers must be provided as URLs of format: tcp://<host>:<port>. Extra URLs serve as fallback.		VectorString	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_0.baseline.start	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_0.baseline.stop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_0.calibration.factor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	RUNNING
channel_0.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	RUNNING
channel_0.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	RUNNING

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_0.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	RUNNING
channel_0.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	RUNNING
channel_0.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_0.numberOfPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_0.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_0.output_hostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_0.output_no_input	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	
channel_0.peak_samples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	
channel_0.pulse_period	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	
channel_1.baseline_start	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	
channel_1.baseline_stop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_1.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_1.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_1.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON
channel_1.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_1.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_1.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_1.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_1.outputDistributionMode	Output Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	
channel_1.outputHostname	Output Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_1.outputNoInputShared	Output No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_1.peakSamples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	ON
channel_1.pulsePeriod	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	ON
channel_2.baselineStart	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	ON
channel_2.baselineStop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	ON
channel_2.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_2.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_2.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_2.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_2.fixedBaselineEna	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON
channel_2.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_2.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_2.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_2.output.hostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_2.output.no_input	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	
channel_2.peak_samples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	
channel_2.pulse_period	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	
channel_3.baseline.start	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	
channel_3.baseline.stop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_3.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_3.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_3.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON
channel_3.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_3.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_3.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_3.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_3.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	
channel_3.outputHostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_3.outputNoInputShared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_3.peak_samples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_3.pulse_period	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_4.baseline_start	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_4.baseline_stop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_4.calibration_factor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	RUNNABLE
channel_4.enable_peak_computation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	RUNNABLE
channel_4.enable_raw_data_streaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	RUNNABLE

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_4.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_4.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON
channel_4.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_4.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_4.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_4.output.hostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_4.output.noInputShared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	
channel_4.peakSamples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	
channel_4.pulsePeriod	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	
channel_5.baselineStart	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	
channel_5.baselineStop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_5.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_5.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_5.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON
channel_5.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_5.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_5.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_5.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_5.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	
channel_5.outputHostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_5.outputNoInputShared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_5.peakSamples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	ON
channel_5.pulsePeriod	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	ON
channel_6.baselineStart	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	ON
channel_6.baselineStop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	ON
channel_6.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_6.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_6.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_6.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_6.fixedBaselineEna	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON
channel_6.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_6.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_6.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_6.output.hostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_6.output.no_input_shared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	
channel_6.peak_samples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	
channel_6.pulse_period	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	
channel_7.baseline.start	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	
channel_7.baseline.stop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_7.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_7.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_7.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON
channel_7.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_7.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_7.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_7.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_7.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	
channel_7.outputHostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_7.outputNoInputShared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_7.peakSamples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_7.pulsePeriod	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_8.baselineStart	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_8.baselineStop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	RUNNING
channel_8.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	RUNNING
channel_8.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	RUNNING
channel_8.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	RUNNING

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_8.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_8.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON
channel_8.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	ON
channel_8.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	ON
channel_8.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_8.output_hostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_8.output_nonexist_input_shared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	
channel_8.peak_samples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	
channel_8.pulse_period	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	
channel_9.baseline_start	Start of Baseline	Starting Sample to calculate the Baseline.		UInt32	USER	RECONFIGURABLE	
channel_9.baseline_stop	Stop of Baseline	Ending Sample of the Baseline calculation.		UInt32	USER	RECONFIGURABLE	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_9.calibrationFactor	Calibration Factor	Factor to be used with all peak values and the related mean and std values		Double	USER	RECONFIGURABLE	ON
channel_9.enablePeakComputation	Enable Peak Computation	Enable peak computation on the FPGA.		Bool	USER	RECONFIGURABLE	ON
channel_9.enableRawDataStreaming	Enable Raw Data	Enable streaming out of raw data.		Bool	USER	RECONFIGURABLE	ON
channel_9.fixedBaseline	Fixed Baseline	If fixed baseline is enabled, this value will be used for calculations instead of the base line from the h/w.		Double	USER	RECONFIGURABLE	ON
channel_9.fixedBaselineEnable	Enable fixed Baseline	Enables the use of a fixed baseline value.		Bool	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_9.initialDelay	Initial Delay	Time delay between trigger and start of processing algorithm.		UInt32	USER	RECONFIGURABLE	
channel_9.numPulses	Number of pulses	Number of pulses expected in each trigger.		UInt32	USER	RECONFIGURABLE	
channel_9.outputDistributionMode	Distribution Mode	Describes the policy of how to fan-out data to multiple (shared) input channels		String	USER	INITONLY	
channel_9.outputHostname	Hostname	The hostname to which connecting clients will be routed to		String	USER	INITONLY	
channel_9.outputNoInputShared	No Input (Shared)	What to do if currently no share-input channel is available for writing to		String	USER	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_9.peakSamples	Peak Samples	Number of peak samples in each pulse.		UInt32	USER	RECONFIGURABLE	
channel_9.pulsePeriod	Pulse Period	Number of samples between each pulse.		UInt32	USER	RECONFIGURABLE	
config.fpgaClock	FPGA Source Clock	Source Clock to FPGA operations.		String	USER	INITONLY	
config.softTriggerInt	Software Trigger Interval	Interval between software generated triggers in milli seconds.		UInt32	USER	RECONFIGURABLE	
config.triggerSource	Trigger Source	Source of trigger for algorithm (RX17 to TX20 - Backplane; Front1-4 - Harlink Front Panel).		String	USER	INITONLY	
dacNode.dacCycleSamples	DAC Cycles per Samples	Number of clock cycles per Samples.		UInt32	USER	RECONFIGURABLE	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
dacNode.dacDataMode	DAC Data Mode	False: binary offset; True: 2 complement		Bool	USER	RECONFIGURABLE	ON
dacNode.dacFile	DAC File	File with DAC values.		String	USER	RECONFIGURABLE	ON
dacNode.dacSkipSamples	DAC Skip Samples	Number of DAC samples to be in output.		UInt32	USER	RECONFIGURABLE	ON
dacNode.dacTriggerEn	DAC Internal Trigger	Enable DAC Internal Trigger.		Bool	USER	RECONFIGURABLE	ON
dacNode.dacTriggerPeriod	DAC Internal Trigger Period	Period of internal DAC Trigger.		UInt32	USER	RECONFIGURABLE	ON
dacNode.enableDac	Enable DAC	Enable DAC channel.		Bool	USER	RECONFIGURABLE	ON
dacNode.voltageIntercept	Voltage Intercept Conversion (y-intercept)	Value of intercept for converting DAC to Voltage value		Double	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
dacNode.voltageSlope	Voltage Conversion (slope)	Value of slope for converting DAC to Voltage value		Double	USER	RECONFIGURABLE	ON
delay	Raw Delay	Time delay between trigger and start of raw data acquisition.		UInt32	USER	RECONFIGURABLE	ON
deviceFile	Device File	Device driver file to access the hardware (e.g. /dev/pciedevs9).		String	USER	RECONFIGURABLE	ERROR
mapDirectory	Map Directory	Folder where all xml mapping files are located.		String	USER	RECONFIGURABLE	ERROR
numberRawSamples	Number of raw samples	Number of raw samples to aquire, per channel, with each start of raw data acquisition.		UInt32	USER	RECONFIGURABLE	ON

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
skipSamples	Skip Samples	If 1/2/3/..., show only every 2nd/3rd/4th/... raw ADC value (i.e. "zoom out").		UInt32	USER	RECONFIGURABLE	RUNNABLE
channel_0.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_0.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_1.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_1.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_2.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_2.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_3.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_3.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_4.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_4.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_5.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_5.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_6.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_6.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_7.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_7.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_8.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	
channel_8.output.port	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
channel_9.output.compression	Compression	Configures when the data is compressed (-1 = off, 0 = always, >0 = threshold in MB		Int32	EXPERT	INITONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
channel_9.outputPort	Port	Port number for TCP connection		UInt32	EXPERT	INITONLY	
interfaces	interfaces			VectorString	EXPERT	READONLY	
performanceStatistics.enable	Enable Performance Indicators	Enables some statistics to follow the performance of an individual device		Bool	EXPERT	RECONFIGURABLE	
performanceStatistics.maxEventLoopLatency	Maximum event loop latency	Maximum time interval between posting a message on the central event loop and processing it within averaging interval.		UInt32	EXPERT	READONLY	
performanceStatistics.maxProcessingLatency	Maximum processing latency	Maximum processing latency within averaging interval.		UInt32	EXPERT	READONLY	

Continued on next page

Table 1 – continued from previous page

Key	Displayed Name	Description	Alias	Type	Access Level	Access Mode	Allowed States
performanceStatistics.messagingProblems	Messaging problems	If true, there is a problem consuming broker messages		Bool	EXPERT	READONLY	
performanceStatistics.numberOfMessages	Number of messages	Number of messages received within averaging interval.		UInt32	EXPERT	READONLY	
performanceStatistics.processingLatency	Processing latency	Average time interval between remote message sending and processing it in this device.		Float	EXPERT	READONLY	
useTimeserver	Use Time-server	Unused - whether device connects to time server is configured via 'time-ServerId'		Bool	ADMIN	INITONLY	

CHAPTER 7

Indices and tables