HowToPulsePicker Documentation

Release 1.0.0

gabriele.giovanetti@xfel.eu

Dec 16, 2021

Contents

Contents:

CHAPTER 1

How to run scans with the Pulse Picker

1.1 Pulse picker specific devices

1.1.1 The Pulse Picker

The Pulse Picker is a hardware device that controls a shutter. It is designed to let a single pulse out of a train pass through (no matter which pulse). It's currently installed in SPB. From the Karabo point of view it looks like a bunch of Beckhoff digital inputs/outputs. We use two of them to control it: We set SPB_XTD9_PPU/DCTRL/TRIG *ON* to start the pulse picker (and it goes back to *OFF* when it is done) We read back SPB_XTD9_PPU/DCTRL/STEP2 as its raising edge is the best approximation we have of the time of the passing pulse. *Note:* motor SPB_XTD9_PPU/MOTOR/ROTOR is used behind the scenes in follow-target mode. Do not mess with its settings. Just in case, it may help to know that slit is open when motor actual position is (2 * N) * 30 for any N, closed when it is (2 * N + 1) * 30. So resting value for actual position should be any odd multiple of 30, and *stepSize* value should be 30.

1.1.2 PulsePickerTrigger

It is a middlelayer device. It repeatedly activates Pulse Picker for a given *Acquisition Time* at a given *Repetition Rate* (thus letting *Number of Pulses* pass through). As a pulse passes through, the device writes on and output channel a boolean with the same train Id of the pulse (as described in *Pulse Picker*). It is meant to be *Karabacon*-friendly.

When the "Pass-Through" option is enabled, it opens the pulse picker slit (assuming it is closed at the beginning), keep it open for at least `acquisitionTime`, then closes it again. At the same time *ImagePicker* is disabled.

In both cases if averagers names are set, it takes care of resetting averaged scan values.

1.1.3 ImagePicker

Yet another device from the *imageProcessor* family. It gets an image stream in input (e.g. from a camera), and the train id on a second one (as provided from *PulsePickerTrigger*). It writes on an output channel only the images whose train id matches the one receiced by *PulsePickerTrigger*. Users can set an offset value to delay or anticipate the image

train id with respect to the pulse one. The ImagePicker can be disabled by setting to `True` the **Disabled** boolean. This way the images in input are simply copied to ouput, as if the ImagePicker was not present.

1.2 Other devices needed for Pulse Picker scans

1.2.1 LimaBaslerCamera

Used to acquire beam image. It is supposed to acquire at 10 Hz.

1.2.2 ImageProcessor

Used to evaluate beam width (4 * sigmaX) and height (4 * sigma Y) on the relevant images (*ImagePicker* Output).

1.2.3 GenericAverage

This device is used to average beam width and height read from *ImageProcessor* over values. N must be <= *PulsePickerTrigger* *Number of Pulses*. In the ideal case (i.e. all pulses lead to a valid value) they are equal. Devices and properties to be averaged can be defined at configuration time. Output averaged values have the form of nodes whith `average` and `nOfValues` properties. The name of the node can be configured, **Note that such names must match the hard-coded interface of Karabacon** Average values have the form of nodes, whose name can be selected at con

1.2.4 Karabacon

Meant to scan over a motor position, use *PulsePickerTrigger* as trigger and *GenericAverage* as a data source.

CHAPTER 2

Example setup for Pulse Picker scans

The following informations are updated to 11-10-2018. They can anyway be useful for refernece as an example.

In order to test pulse picker scans the following system was set up. It may be found in subproject *SCANTOOL* in domain *SPB*.

Device Class	Device ID
LimaBaslerCamera	`MY_FAVOURITE_CAMERA`
ImagePicker	`SPB_XTD9_PPU/PROC/IMPICK`
ImageProcessor	`SPB_XTD9_PPU/PROC/IMPROC`
PulsePickerTrigger	`SPB_XTD9_PPU/MDL/TRIGGER`
GenericAverage	`SPB_XTD9_PPU/MDL/BW_AVG`
Karabacon	`SPB_KARABACON`

2.1 Devices configuration remarks

- ImagePicker `SPB_XTD9_PPU/PROC/IMPICK`
  - **Input image**: connected output channel = `MY_FAVOURITE_CAMERA:output`

    The camera name here is all you need to change when switching image soource
  - **Input train id**: connected output channel = `SPB_XTD9_PPU/MDL/TRIGGER:output`
- ImageProcessor `SPB_XTD9_PPU/PROC/IMPROC`
  - **Input**: connected output channel = `SPB_XTD9_PPU/PROC/IMPICK:output`
- PulsePickerTrigger `SPB_XTD9_PPU/MDL/TRIGGER`
  - **averageNames** = `['SPB_XTD9_PPU/MDL/BW_AVG']`
  - **repetition rate** and **acquisition time** can be configured to achieve the desired number of pulses per scan point
- GenericAverage `SPB_XTD9_PPU/MDL/BW_AVG`

- **properties table**

deviceName	propertyName	averageOutputName
SPB_XTD9_PPU/PROC/IMPROC	beamWidth1d	foursigmax
SPB_XTD9_PPU/PROC/IMPROC	beamHeight1d	foursigmay

Note: averageOutputName values can be changed at configuration time **but** they have to match the hardcoded values in Karabacon interface, so do not change them if you want to perform a scan.

- **Window Size**: it could make sense to have this matching the PulsePickerTrigger numberOfPulses but it is not strict

- Karabacon SPB_KARABACON
  - Main control scene:
    - Motor environment: the motor you want to scan on (axis: default)
    - Data sources:

Alias	DeviceId	value
[e.g.] beamwidthX	SPB_XTD9_PPU/MDL/BW_AVG	foursigmax
[e.g.] beamwidthY	SPB_XTD9_PPU/MDL/BW_AVG	foursigmay

    - Trigger Sources: put here the device id of PulsePickerTrigger (i.e. SPB_XTD9_PPU/MDL/TRIGGER)

CHAPTER 3

Indices and tables

- genindex
- modindex
- search