

---

# **bunchPatternConfigurator**

*Release 1.0*

**CTRLS**

**Nov 07, 2024**



## CONTENTS

<b>1</b>	<b>BunchPatternConfigurator</b>	<b>3</b>
<b>2</b>	<b>Indices and tables</b>	<b>9</b>



Contents:



## BUNCHPATTERNCONFIGURATOR

This device provides an interface from Karabo to the DOOCS **Bunch Pattern Server**.

The DOOCS Bunch Pattern Server provides the main controls for the timing pattern. Those controls for the SASE1, SASE2 and SASE3 bunch patterns are found under the XFEL.UTIL/BUNCH\_PATTERN/CONTROL DOOCS location. Similar DOOCS locations exist in test environments such as CAS-LAB and EEE lab. Using this location, the XFEL beam can be allowed or inhibited, the number of bunches can be changed, beam limits can be set or removed, and dark current suppression can be enabled or disabled (see more in the DOOCS Bunch Pattern Server documentation:

[Bunch Pattern Documentation](#)

The BunchPatternConfigurator provides an interface for a very limited subset of the properties available at the DOOCS Bunch Pattern Server location. The BunchPatternConfigurator establishes a connection with a DOOCS location of the Bunch Pattern Server (using DoocsGate from karabo.doocsapi) and enables users to read/write a subset of the location's properties. The primary distinction from the Karabo PulsePatternDecoder device lies in its direct communication with the DOOCS Bunch Pattern Server, contrary to the PulsePatternDecoder which reads the properties indirectly from the TimeServer. Given that the device grants read/write access to the DOOCS location properties, it is important to first read its documentation:

[Bunch Pattern Documentation](#)

### 1.1 Connection to the DOOCS Location and Clones

The connection is established to the DOOCS location provided by the user. Each of the three tunnel topics (SASE1, SASE2, SASE3) has one instance of the BunchPatternConfigurator. Each instance is connected to the same DOOCS location. Instruments have clones to the devices instantiated in their respective tunnels, for example FXE and SPB instruments access and modify the bunch pattern using devices that clone the one running in SASE1 topic.

Each instance operates only with the Bunch subpatterns specified by the user in an init-only parameter "Active SASE" with allowed values "SASE1", "SASE2" and "SASE3".

### 1.2 Bunch Pattern Terms

In the context of this device, the terms **bunch** and **pulse** are used interchangeably. Throughout all the devices's properties and their descriptions, the term 'pulse' is consistently employed to align with the established terminology of pulse pattern structures. However, the karabo device is named 'BunchPatternConfigurator' to reflect its similarity to the DOOCS counterpart software, namely the 'Bunch Pattern Server'. This naming convention serves to differentiate this device from the PulsePatternDecoder, which uses a TimeServer to obtain the current pulse pattern structure.

**Operation Mode:** When a user sets the number of pulses via the BunchPatternConfigurator, the DOOCS Bunch Pattern Server checks the request against the current operating mode of the accelerator, which can be "Mode Zero", "Mode Single", "Mode Short", "Mode Medium", "Mode Full", and applies some limitations on the number of pulses

if necessary. each mode, only a limited number of pulses are allowed to be set. This means that if a user requests number of pulses that exceeds the limit active for the current operating mode, then this request will not result in an actual change of number of pulses. The DOOCS Bunch Pattern Server keeps a list of these limits and enforces them - ensuring that no more than the allowed number of pulses can actually be set. The BunchPatternConfigurator device only reflects the result of the request and notifies the user in its status in case the requested number of pulses was not set. Additionally, the BunchPatternConfigurator, separately from the DOOCS software, checks the EPS allowance for each operation mode (see “EPS allowance check” section).

### **1.3 Monitoring of the Bunch Pattern Structure**

On successful connection to the DOOCS location the device starts monitoring (with some time intervals set by the “Update Period” parameter) the properties of the bunch pattern structure. Some properties of the karabo device are calculated based on these monitored values.

The properties read directly are:



Table 1.1: The properties read directly from the DOOCS location

Property	key	DOOCS property	Description
Beam Allowed	beamAllowed	BEAM_ALLOWED	
User Allow SA1	userAllowSA1	ALLOW_USER_BUNCH_CONTROL_1	Read if user control over the number of bunches is allowed or not for SA1. If the value is nonzero the number of bunches can be modified
User Allow SA2	userAllowSA2	ALLOW_USER_BUNCH_CONTROL_2	Read if number of bunches can be modified for SA2
User Allow SA3	userAllowSA3	ALLOW_USER_BUNCH_CONTROL_3	Read if number of bunches can be modified for SA3
Bunch Current Total	bunchCurrentTotal	NUM_BUNCHES_CURRENT_TOTAL	Current (actual) number of bunches in total
Bunch Current SA1	bunchCurrentSA1	NUM_BUNCHES_CURRENT_1	Current (actual) number of bunches for SA1 pattern
Bunch Current SA2	bunchCurrentSA2	NUM_BUNCHES_CURRENT_2	Current (actual) number of bunches for SA2 pattern
Bunch Current SA3	bunchCurrentSA3	NUM_BUNCHES_CURRENT_3	Current (actual) number of bunches for SA3 pattern
Pattern Title	patternTitle	PATTERN_SERIALIZATION	PAT- TERN_SERIALIZATION contains a string with information about the current pattern setup. Among them it contains Pattern Title.

Some properties of the DOOCS Bunch Pattern Server have the form of a “time-domain spectrum” that displays a certain aspect of the current pattern in an oscilloscope-like manner, e.g. bunches from injector laser 1 or bunches belonging to a certain subtrain. The various PATTERN\_VIEW\_n properties are vertically offset to that they can be plotted in a common plot:

Table 1.2: Calculated properties

Property	key	DOOCS property	Description
Beam Allowed	beamAllowed	BEAM_ALLOWED	
First Slot SA1	firstSlotSA1	calculated using PATTERN_VIEW_3	is calculated using PATTERN_VIEW_3 that contains the ‘time-domain’ spectrum that display the ‘Subtrain SA1’ current pattern
First Slot SA2	firstSlotSA2	calculated using PATTERN_VIEW_4	is calculated using PATTERN_VIEW_4 that contains the ‘time-domain’ spectrum that display the ‘Subtrain SA2’ current pattern
First Slot SA3	firstSlotSA3	calculated using PATTERN_VIEW_5	is calculated using PATTERN_VIEW_5 that contains the ‘time-domain’ spectrum that display the ‘Subtrain SA3’ current pattern
Delta21	delta21	calculated using PATTERN_VIEW_6	Difference between first and second bunch index numbers. Is calculated using PATTERN_VIEW_6 that contains the ‘time-domain’ spectrum that display the ‘Subtrain ALL’ current pattern

## 1.4 Requesting of the Number of the Pulses

Table 1.3: The properties to set number of pulses :widths: 25 25 50 25

Property	key	DOOCS property	Description
Number of Pulses	numPulses	USER_NUM_BUNCHES_REQUESTED_1 or USER_NUM_BUNCHES_REQUESTED_2 or USER_NUM_BUNCHES_REQUESTED_3	Requested number of pulses, the corresponding DOOCS location property is USER_NUM_BUNCHES_REQUESTED_1 (SASE1 case) or USER_NUM_BUNCHES_REQUESTED_2 (SASE2 case) or USER_NUM_BUNCHES_REQUESTED_3 (SASE3 case). If due to some limitations (see “Wrong number of pulses”) the requested number cannot be set the USER_NUM_BUNCHES_REQUESTED_1 (for SASE1 case) will differ from NUM_BUNCHES_CURRENT_1 (Karabo device property bunchCurrentSA1)

## 1.5 Protection Against Wrong Number of Pulses

The DOOCS device has built-in logic to validate if the number of bunches requested by a user for a specific sub-pattern is valid and can actually be set. For example, if the requested value exceeds the upper limit for a number of bunches, the DOOCS device will not allow it to be set. This is reflected by the fact that there are two sets of properties in both Karabo device and DOOCS Server : “Requested” numbers of bunches and “Current” numbers for each sub-pattern and in total.

## 1.6 EPS Allowance Check

There is additional check of the allowance from EPS (Equipment Protection System) side to proceed with the setting of the pulse number requested by the user. The EPS PLC loops in each tunnel topic have dedicated BeckhoffDigitalOutput devices representing the actual operating mode (see more on **Operating Mode** term in “BunchPattern terms” section). The karabo device checks if the BeckhoffDigitalOutput devices of the EPS PLC loops in corresponding tunnel topic are in state ON to proceed with setting of pulse number requested by the user. There are 4 DCTRLS to be checked, one for each mode:

- **Mode Single** DigitalOutput, for example in SASE1 it is the device “SA1\_SYS\_EPS2MPS/DCTRL/MODE1”
- **Mode Short** DigitalOutput, for example in SASE1 it is the device “SA1\_SYS\_EPS2MPS/DCTRL/MODES”
- **Mode Medium** DigitalOutput, for example in SASE1 it is the device “SA1\_SYS\_EPS2MPS/DCTRL/MODEM”
- **Mode Full** DigitalOutput, for example in SASE1 it is the device “SA1\_SYS\_EPS2MPS/DCTRL/MODEF”

If the user requests the number of bunches that can be provided by the specific operating mode, then only the DCTRL of this mode is to be checked. If its state is OFF, the BunchPatternConfigurator will not request the setting of the number on DOOCS side and will notify the user about the restriction.

## 1.7 Instruments Interlock Check

Similarly to the EPS check there are the checks on the instruments side. There are 4 similar DCTRLs per instrument that are running in instruments’ topics and represent the allowance of the operation mode for the specific instrument. For the time being these additional ‘interlock’ checks are done only for the SASE1 BunchPatternConfigurator instance that checks the operation modes allowance on SPB side. Similarly, as with EPS check, 4 DCTRLs per instrument are checked if they are in state ON. Each request for the number of pulses from the user is checked against specific operating mode DCTRL. If its state is OFF the BunchPatternConfigurator does not proceed with the request and notifies the user about it.

## 1.8 “Train on demand” Functionality

The BunchPatternConfigurator device allows “Train on demand” requests: setting the **TrainN** property to N and calling the slot **Train on demand** will result in consecutive N trains.

This functionality is performed using the remote DoocsPulseKicker device. Its Karabo address is mandatory to be specified in property **doocsPulseKickerDev** before instantiation of the BunchPatternConfigurator device. When performing ‘Train on demand’ DoocsPulseKicker calculates the delay time needed to allow the required number of trains.



## INDICES AND TABLES

- genindex
- modindex
- search