

---

# pplPattern

*Release 1.0*

**CAS**

**Nov 07, 2024**



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Server and Device Startup</b>	<b>5</b>
<b>3</b>	<b>Running the Device</b>	<b>7</b>
<b>4</b>	<b>Device Scenes</b>	<b>11</b>
<b>5</b>	<b>Troubleshooting</b>	<b>15</b>
<b>6</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



## INTRODUCTION

The pplPattern middlelayer device is designed to provide an easy interface to configure the pattern of the pulse probe laser (PPL).

A sequence of up to four patterns can be saved per XFEL train, according to the needs of the experiment being run in the instruments. In addition, a *train configuration* can be executed multiple times, and two distinct train configurations can be set.

The PPL pattern configuration is saved in the DOOCS server XFEL.UTIL/BUNCH\_PATTERN/PATTERN\_BUILDER.

Two device scenes are available in the package to setup either a periodic or an arbitrary pattern. The first one is configured by using the scene “Periodic SubPattern” which allows to generate patterns always consistent with the PPL grid. The scene “Arbitrary SubPattern” allows the user to configure very generic patterns, also not in the PPL grid.

This grid is the reference regular sequence of pulses and of empty bunches generated by the frontend hardware of the pulse-probe laser; a base grid element is considered to be the sequence made of one pulse and of its subsequent empty bunches (before the next pulse). The pulse frequency is set in the frontend by choosing there the so called ‘base frequency divider’ which reduces the laser pulse frequency generated by the frontend from 4.5 MHz to a submultiple value.

Being a user-selected pattern on the PPL grid means eventually that the laser to be delivered to the instruments should have the position of pulses aligned with the pulses in the reference pattern as defined by the frontend configuration. In case a user-selected pattern is not on the grid, the pulses of the pattern might not be amplified in the laser system, resulting in a laser pattern delivered to the instruments different from what expected. Patterns of only empty bunches are considered to be on the grid if their length is an integer multiple of the base grid element.

The condition for patterns to be on the PPL grid condition is always fulfilled when using the scene for periodic patterns. This is not the case when building patterns with the other legacy scene.





## SERVER AND DEVICE STARTUP

Due to a specific requirement by LAS group, each instrument must be able to modify only its relevant bit in the 32-bits pattern associated to each electron bunch running in the XFEL machine. This should reduce the possibility of modifying by mistake a ppl pattern configured for a different instrument. This feature is achieved by letting a dedicated server in the laser systems LA1, LA2 and LA3 start automatically the needed pattern devices with the proper laser bits. The devices relevant to each instrument are then cloned in the corresponding Karabo system (topic). As soon as the control host where the pattern server runs comes online, that server is automatically started, instantiating its configured pattern devices. In case those devices do not show up in LAS systems this is a symptom of other major problems affecting the Karabo system.

Being the pattern devices instantiated directly from their servers, they should not show up in the instruments projects; there only the device scenes can be saved if needed. In addition to the case of scenes saved in a project, as always with Karabo devices the default scene can be opened directly by double left-clicking on the device name in the 'System Topology'. All scenes can be also be directly opened by right-clicking on the the device name, clicking on 'Open Device Scene' menu item, and selecting the desired scene in the pop-up list.



## RUNNING THE DEVICE

No special action should be taken by the user to instantiate the device; this is done automatically by the Karabo system.

An example of the device configuration editor after it comes online, is presented in [Fig. 3.1](#):

After initialization the users can cross-check the correctness of the PPL bit in use:

- **Laser Pattern:** The laser pattern which can be used by an instrument. Every pattern corresponds to a specific bit in the 32-bits pattern associated to each electron bunch. One instrument user should have only one pattern, except LAS group which can instead modify two bits for testing purposes. The pattern bits available in the system are the following:
  - LP\_LAS1: For LAS,
  - LP\_LAS2: For LAS,
  - LP\_SPB: For SPB instrument,
  - LP\_FXE: For FXE instrument,
  - LP\_SQS: For SQS instrument,
  - LP\_SCS: For SCS instrument,
  - LP\_SASE2: For SASE2 instruments.

The status of electron bunches in the XFEL machine is periodically retrieved. The interval to wait before polling for an update from DOOCS can be tuned by changing the value of the parameter **Charge Update Interval** (its default value is 2s). Each instrument will have visualized only the sequence of bunches emitting X-rays in their corresponding SASE tunnel. In case of LAS or Control user, at moment only SASE1 is displayed; in the next releases the possibility to choose at run time which tunnel to monitor will be given. The base frequency for the bunches (**XFEL Bunch Base Frequency**) is regularly updated from DOOCS.

The variable **Selected Pattern** allows to configure the ppl patterns in a train. At the moment, up to eight types of train configurations can be set, named as *A, B, ..., H*, which can be selected via its drop-down menu.

A train configuration can be repeated *N* times in sequence by configuring the variable **Target Pattern Sequence**. Pattern types not inserted in the sequence will not be used in the PPL firing sequence. The currently running sequence is shown in the variable **Actual Pattern Sequence**. Let us suppose we have configured, as an example, the target configuration *N[A]M[B]*. This will translate in firing the PPL in sequence the pattern *A* for *N* consequent trains, and soon after firing the pattern *B* for the next *M* trains. At the end of the last train, the sequence will restart from the beginning. In order to make the target sequence active (i.e. to save it in DOOCS), the slot **Write Pattern Sequence to Doocs** has to be called.

Each train can be then configured with up to four consequent subpatterns, independently configurable, exposed as nodes in the device configuration editor, [Fig. 3.2](#).

For each node (sub-pattern) the following variables can be set:

- **Nr. of Laser Pulses:** The number of laser pulses in a sequence;

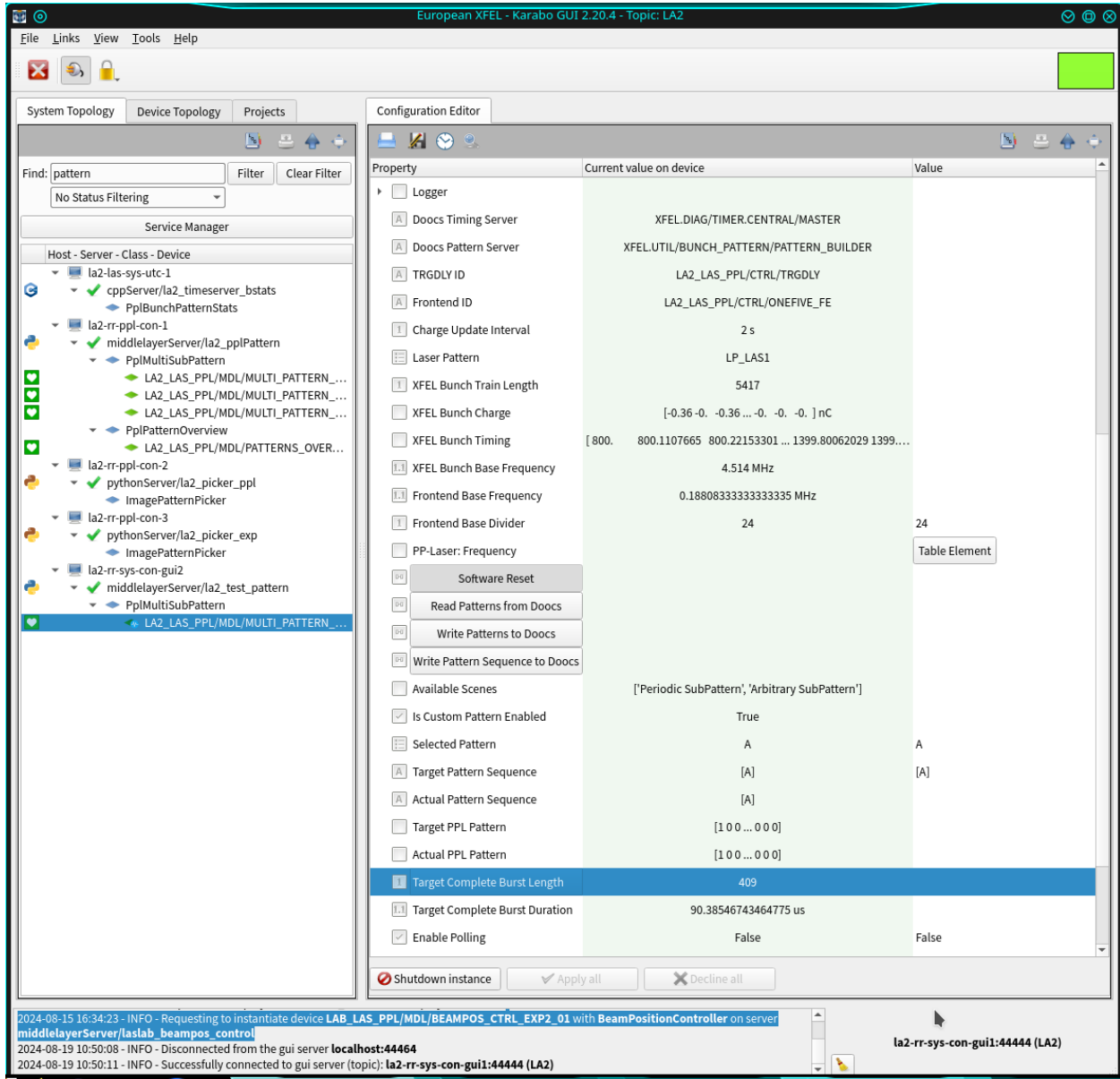


Fig. 3.1: Fig. 3.1: After successful initialization the users can visualize the PPL pattern they are allowed to work with.

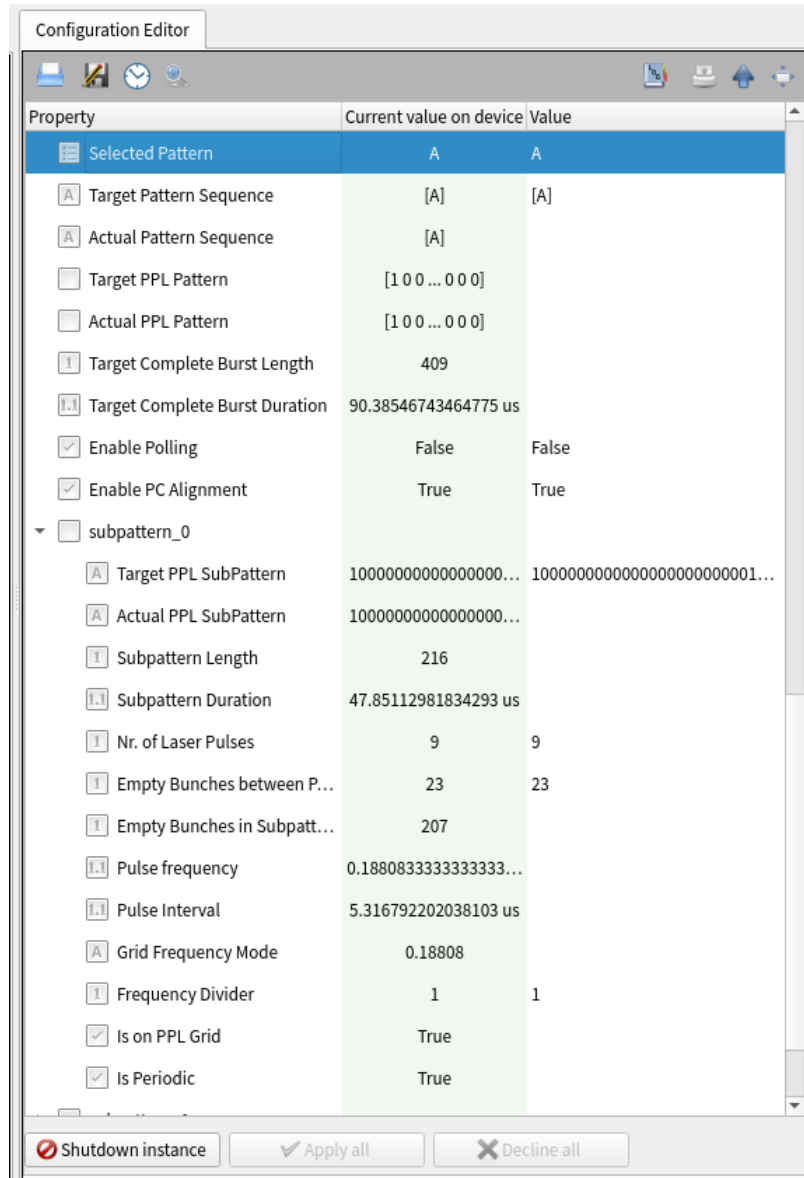


Fig. 3.2: Fig. 3.2: Up to four consequent sub-patterns can be configured in each train.

- **Frequency Divider:** The integer to select a submultiple of the base frequency generated by the the laser frontend (FE) hardware. The pulse frequency and the interval between pulses in the specific sequence will change accordingly.

The frontend base frequency **Base FE Frequency** is determined by a frequency divider internal to the hardware, whose value change is always notified to the pattern device, and shown in the property **Frontend Base Divider**; the latter is not shown directly in the scene to avoid accidental changes of this critical variable.

When the frontend divider changes the target pattern is not modified, while the pattern divider is recalculated according to the newly received frontend divider and the pulse frequency in the target pattern. In case this pulse frequency results no more to be a submultiple of the frontend frequency the divider is defaulted to unity, and a notification is shown to the users via the device status property displayed in the device scene in a ‘Text Log’ widget.

Everytime the frontend divider changes in the device its value is saved online, and can be retrieved after device restart in case the connection with the frontend hardware cannot be established via Karabo. In this scenario LAS users can also change its value according to a new configuration set in the frontend hardware, thus allowing building laser patterns consistent with the pulse frequency set in the frontend hardware (referred to as ‘pattern being on the PPL grid’).

When the frontend divider changes in the pattern device a new lookup table **PP-Laser: Frequency** of allowed pulse frequency values is generated. The table is compiled only up to the first 44 frequency divider values, and its aim is to help the user visualizing the pattern properties.

After each selection (*Laser Pulses* or *Frequency Divider*) is *entered* in the subpattern editor, the variable **Target Ppl Pattern** will be updated. The total subpattern length (in unit of XFEL bunches) and interval, including all empty bunches, will be also updated. In case of subpatterns made only of empty bunches the subpattern length is given by the number of hypothetical inter-pulse empty bunches (allowed by the chosen divider value) plus one unit (the missing initial pulse). This configuration will not be automatically transferred to the DOOCS server; this will be done only after pressing the slot (button) **Write Pattern to Doocs**. The new pattern stored in DOOCS (**Doocs: Ppl SubPattern**) will be then updated accordingly. Note that the patterns saved in DOOCS are regularly monitored every ten seconds if the Boolean property “Enable Polling” is set; it is by default True for devices with instrument PPL bit, False with LAS1/LAS2 bits.

Each time a pattern or a pattern sequence is pushed to DOOCS the proper duration for the PPL Pockels cell is calculated, to let the correct number of pulses propagate to the instrument; its trigger is configured accordingly for allowing this. The value is displayed in the device status property and is shown in the device scene. If needed, this behaviour can be disabled by unsetting the Boolean property **Enable PC Alignment**.

To retrieve the current patterns the slot **Read Pattern from Doocs** can be called. An automatic readout from DOOCS is done soon after patterns have been pushed to DOOCS.

The variable **Target Complete Burst Duration** shows (for each selected pattern type) the duration of the corresponding complete PPL burst, from the first to the last pulse.

## DEVICE SCENES

At the moment, two scenes are auto-generated by the devices, *Arbitrary SubPattern* and *Periodic SubPattern*. They have been probably been saved in the instrument project by the Control group. In case a new version is available, it can be opened by right-clicking on the device name, and selecting from the drop-down menu the item *Open device scene*.

An example of scene for the periodic pattern configuration is presented in Fig. 4.1:

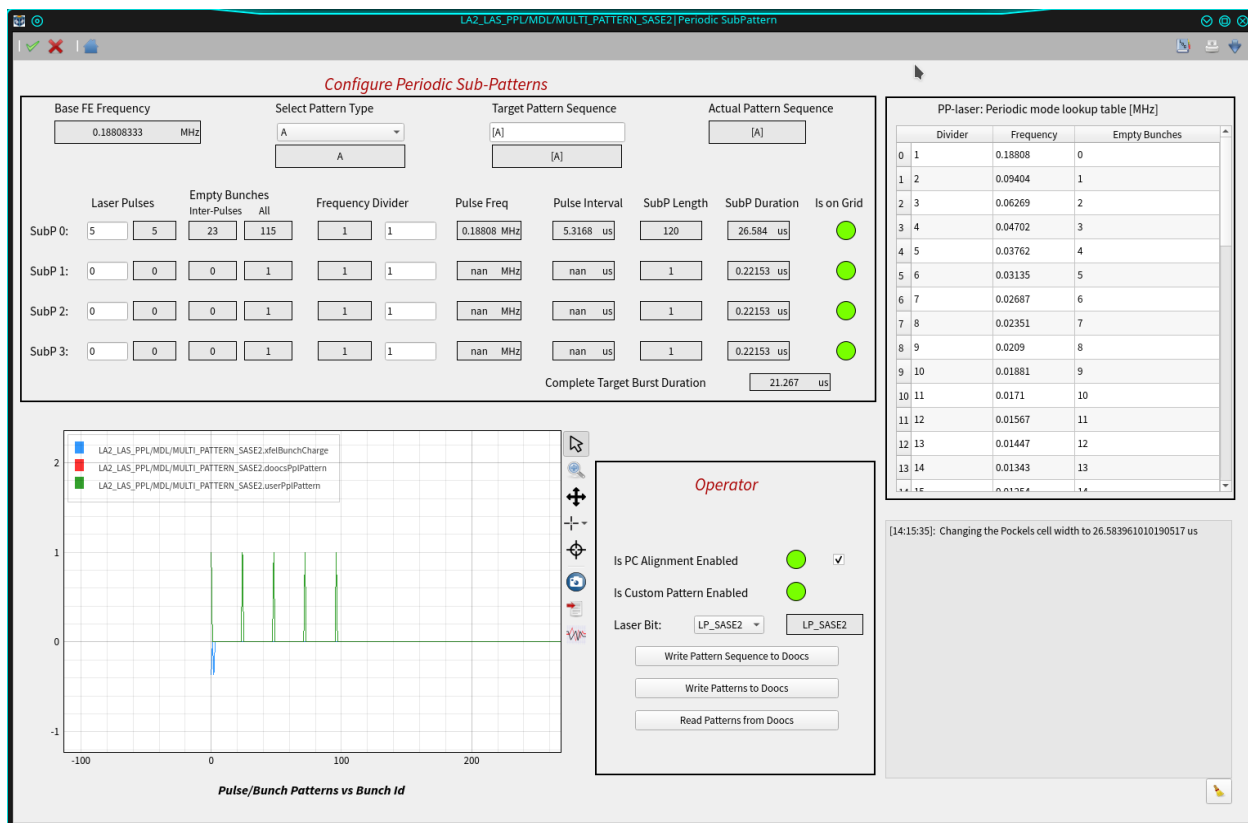


Fig. 4.1: Fig. 4.1: The scene for periodic configuration of the PPL subpatterns.

In the graph three patterns are at the moment presented:

- **Target PPL Pattern:** The entire target PPL pattern (KRB key: 'userPplPattern') for the selected pattern type configured by the user,
- **Actual PPL SubPattern:** The entire actual PPL pattern (KRB key: 'docsPplPattern') saved in DOOCS,
- **XFEL Bunch Charge:** The charge pattern for bunches emitting x-rays (KRB key: 'xfelBunchCharge'), in the instrument SASE.

Patterns can be selected/deselected from the graph legend.

Note that the displayed grid of the charge bunch pattern reflects the fundamental XFEL repetition rate  $9\text{ MHz}$ , while the PPL pattern grid follows the grid in the relative DOOCS server ( $4.5\text{ MHz}$  in the shown example). This results in an artificial rescaling of the charge pattern w.r.t the ppl patterns. In the next releases this mismatch will be fixed, and the grid of bunch charge will be properly rescaled, to help the user to correctly compare the grid of the two types of patterns.

A lookup table is provided by the laser group to help selecting in a subpattern the frequency divider value to properly deliver PPL pulses with the required frequency.

For arbitrary pattern selection a dedicated scene exists, Fig. 4.2:

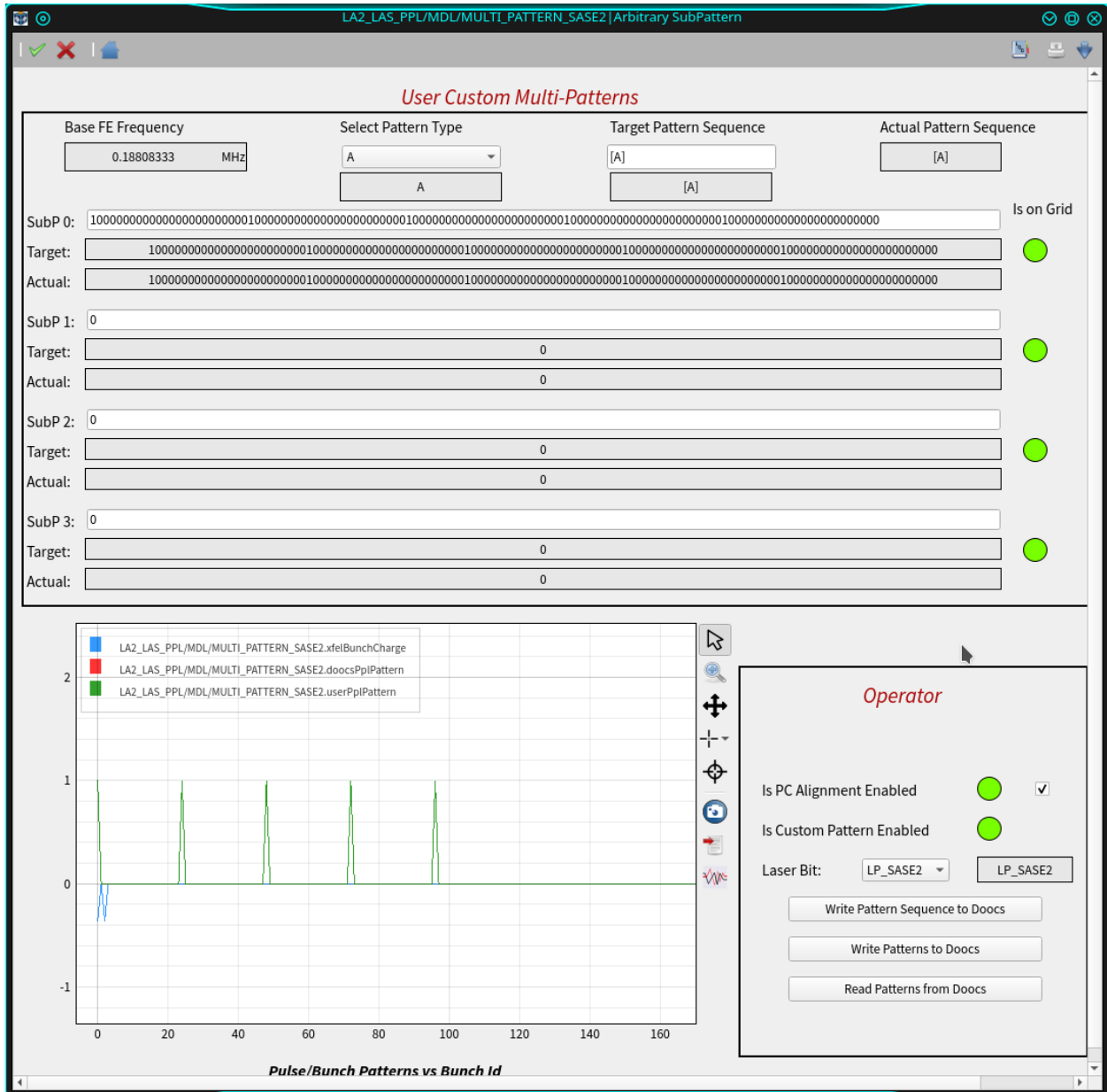


Fig. 4.2: Fig. 4.2: Generic fancy PPL subpatterns can be set using a dedicated scene.

No check is done on the consistency of the patterns here written by the user. Any change done in one scene will be



reflected in the other one because they both describe the same parameter configuration in DOOCS. Thus, for standard (no fancy) patterns users are advised to use the scene optimized for periodic subpatterns, which makes sure that patterns are correctly built on the PPL grid.



---

## TROUBLESHOOTING

Some typical errors have been identified up to now:

- Pattern device suddenly crashed, and could not be re-instantiated due to continuous initialization failures. This is typically due to the device encountering unexpected property changes in the DOOCS side, e.g. reset of patterns.

**Possible solution:** In this type of cases the instrument can do really nothing, except getting in contact with the DOOCS responsables via DOC (9-8089), or LAS colleagues,

- The configured PPL pattern appears not to be applied. After pushing it to DOOCS the patterns displayed in the graphs are not changed. Possible scenarios are:

- It happens sometime that the slot *Write Pattern Sequence to Doocs* is pressed instead of *Write Patterns to Doocs*.

**Possible solution:** activate the configured pattern by pressing the slot *Write Patterns to Doocs*,

- Everything was done correctly, but the Boolean *Is Custom Pattern Enabled* in the scene is false. In this case the machine is running in the so called *Legacy mode* which does not follow the user setting. This should not happen during run operations; if that is the case please contact the DOOCS responsables,

- The pattern was configured for a pattern type, e.g. *A*, which is not present in the running pattern sequence, e.g. *[B]2[C]*.

**Possible solution:** activate the proper pattern sequence which contains the configured pattern type,

- The generated patterns differ from what expected from the pattern configuration saved in DOOCS; sometimes also zero pulses are observed in the instrument hutch or in the LAS diagnostics. The pulse frequency is not being compliant with what allowed by the frontend base frequency shown in the device. Look at the frequency lookup table for the allowed frequencies to be used, according to the base frontend frequency. Possible scenarios are:

- The base frontend frequency was modified in the hardware and the currently active pattern is no more in the grid.

**Possible solution:** If you are sure that the new displayed frontend frequency is correct

then the patterns should be reconfigured with respect to that value.

- No communication could be established by Karabo to the frontend hardware, and a wrong value for the frontend frequency was accidentally set in the pplPattern device. The active pattern appears mistakenly to be not in the grid.

**Possible solution:** The correct frontend frequency value should be set by LAS colleagues in the device,

- The generated patterns do not show pulses expected at the end of the configured pattern. This is typically due to the property *Complete Target Burst Duration* being larger than 600 microseconds, which is the maximum interval the Pockels cell can stay with voltage high to let pulses propagate and get amplified and delivered to the instruments.

**Possible solution:** Configure patterns with smaller burst duration,

- The subpattern pulse frequency (*Pulse Freq.*) is displayed as a NaN (not a number). In the current code implementation this is not an error, is normal and is due to the number of laser pulses being smaller than two. In this case, also the interval between two pulses (*Pulse Interval*) and the number of empty bunches between pulses (*Inter-Pulses Empty Bunches*) cannot be measured and are set to NaN and zero, respectively.

## INDICES AND TABLES

- [genindex](#)
- [modindex](#)
- [search](#)