x2TimerML

Release 1.0

CAS

Jun 01, 2021

Contents

Contents:

CHAPTER 1

x2TimerML Introduction

The x2TimeML middlelayer device is designed to provide an easy interface, per trigger channel, to the x2Timer device. The latter, in turns, provides an interface to a μTCA [1] crate controlled by the DOOCS system, and provides access up to 21 trigger channels. An example of the x2Timer device installed in the laser lab is presented in Fig. 1.1:

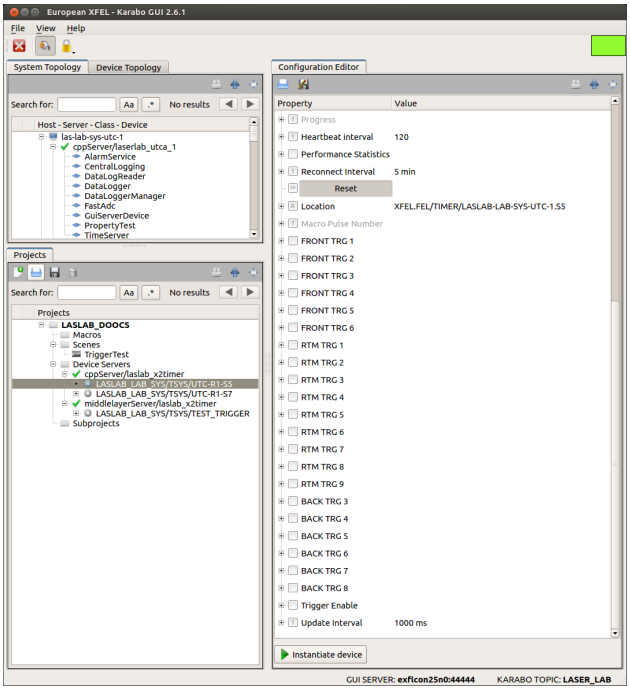

Fig. 1.1: The configuration editor of the x2Timer device provides an interface to the trigger channels available in a μTCA crate.

In the given example the karabo device is connected to the μTCA at DOOCS address XFEL.FEL/TIMER/LASLAB-LAB-SYS-UTC-1.S5. To help the user to easily configure the triggers, the middlelayer device x2TimerML was designed, allowing to configure a particular channel.

[1] MicroTCA Technology Lab, https://techlab.desy.de/

CHAPTER 2

Device Initialization

At initialization phase the x2TimerML device expects two mandatory parameters, Fig. 2.1:

- **X2 Timer**: The name of the x2Timer karabo device connected to a specific μTCA crate;
- **Channel Name**: The trigger in the μTCA crate, which we want to use. The trigger can be selected from the available triggers using a predifined drop-down list, Fig. 2.2.

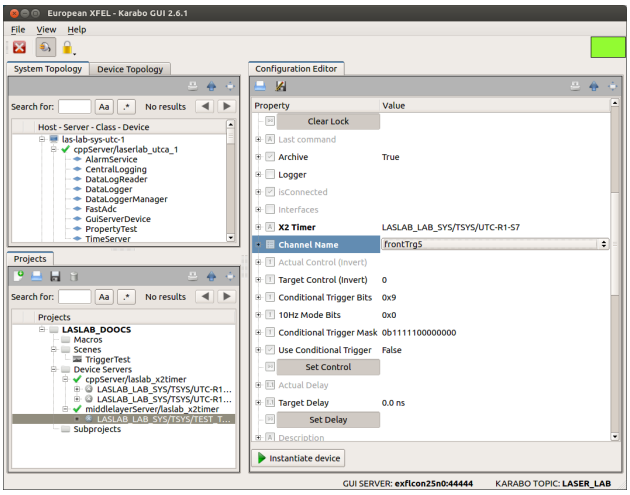

Fig. 2.1: Before initialization mandatory parameters (in bold font in the device configuration editor) must be set, otherwise the device will not be started.

Failing to properly provide a valid instance for a correct x2Timer device name will result in the device remaining in the **INIT** state, Fig. 2.3:

In case the connection with the x2Timer could be established the middlelayer device should reach the state **ON**, Fig. 2.4:

When the device is online the parameters “X2 Timer” and “Channel Name” cannot be changed; this is only possible by first shutting the device down. The device, when online, can be controlled also by its auto-generate scene, which can be opened by double-clicking on the device name in both the “System Topology” and “Projects” panel in the karabo gui client, Fig. 2.5:

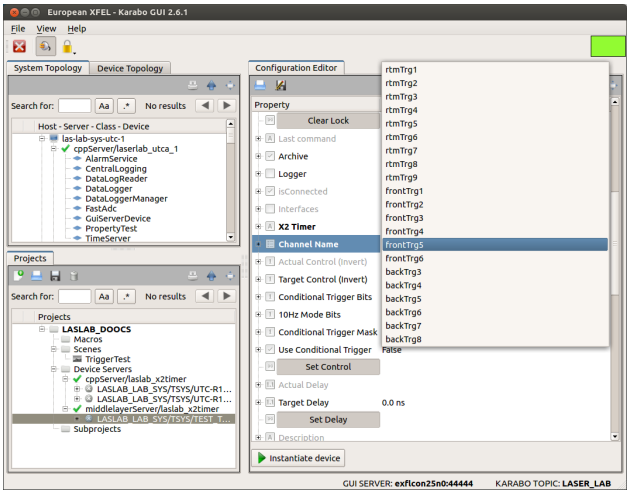

Fig. 2.2: The trigger can be chosen using an available drop-down list.

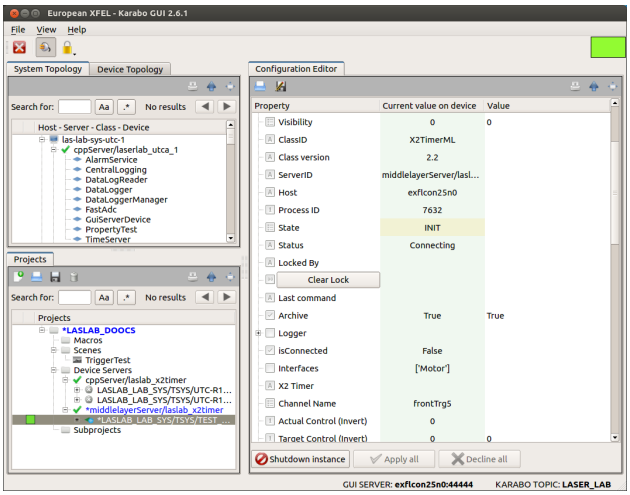

Fig. 2.3: The device remains in INIT state when trying to connect to an x2Timer which is not online.

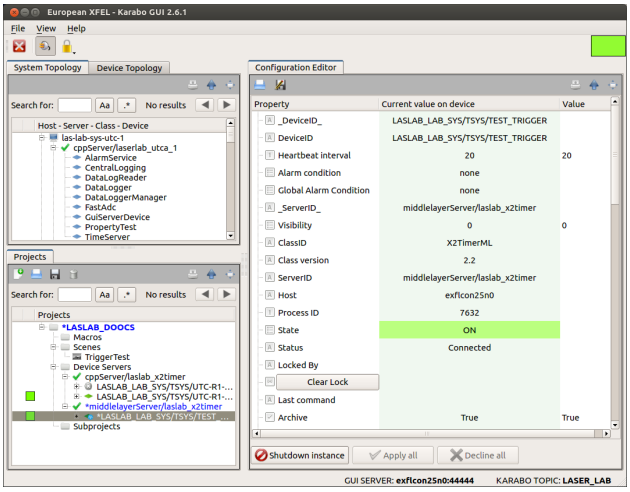

Fig. 2.4: If the initialization phase was successful, the device should eventually reach the state ON.

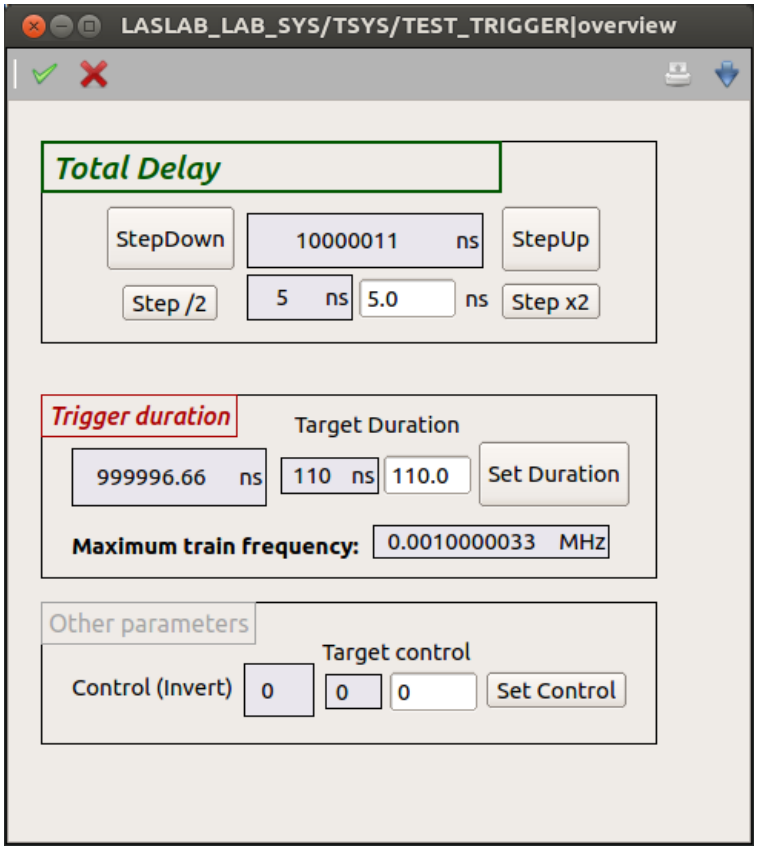

Fig. 2.5: An auto-generated scene allows the user to configure the device.

CHAPTER 3

Trigger Parameters

At run time the trigger accessed by the middlelayer device can be configured by setting its parameters: Fig. 3.1:

- **Delay**: The trigger delay in nanosecond units with respect to the reference timing in XFEL. A warning is issued if the value is larger than 0.1 s, as you might be off by a train;
- **Pico-Delay**: A finer tuning of the trigger delay, in picosecond units. To help the user, the device shows also the calculated total trigger delay currently set ("Total Delay");
- **Duration**: The trigger duration (or "width") in nanosecond units. The maximum train frequency ("Max Train Freq.") allowed by this value is automatically calculated as

$$f = 1000/\Delta T \quad MHz$$

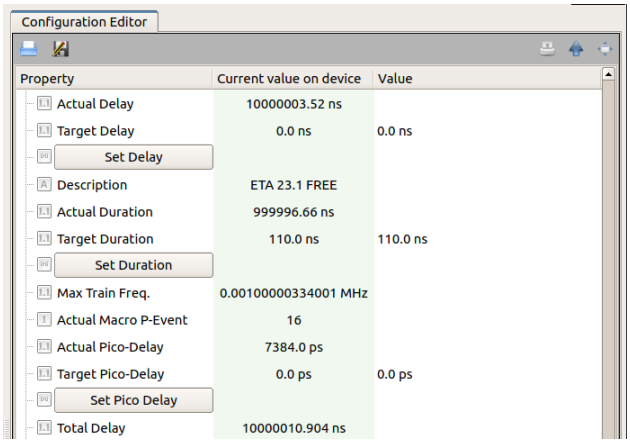

Fig. 3.1: The trigger time parameters can be configured at run-time.

Note that x2Timer device configures the trigger timing parameters using, as defined in DOOCS, multiples of the base step 9.23 or 923 for nanoseconds or picoseconds, respectively. Differently from that method, the middlelayer device x2TimerML allows the user to easily choose a parameter in terms of real time units. In case the selected value was not an exact multiple of the base step, a rounding to the closest multiple is done.

As usual in many karabo devices, a parameter is described by its actual value (e.g., **Actual Delay**) and by the value which the user is interested in (e.g., **Target Delay**). To set the desired value of a trigger parameter the corresponding slot (e.g., **Set Delay**) has to be eventually clicked.

The karabo middlelayer is exposing to the users some of the trigger parameters configurable directly in the DOOCS system, and by changing these parameters, the trigger configuration is actually changed in DOOCS. An example of the DOOCS panel comprising the timing parameters of a trigger channel is presented in Fig. 3.2,

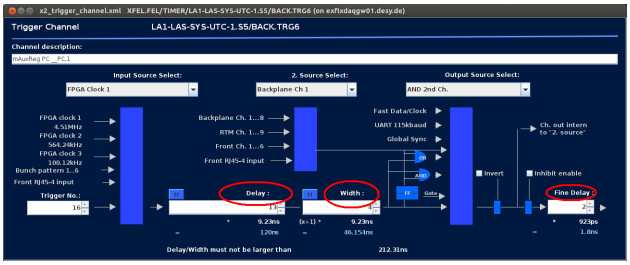

Fig. 3.2: Doocs panel to configure a specific trigger output channel in a μTCA board.

An interesting feature of the device is the possibility to allow *pulse on demand* triggers; a trigger is output when a pulse condition is generated. This is achieved by setting a logical "*and*" condition between the configurable μTCA trigger handled by the device and a second configured trigger channel, "*2. Source Select*" in Fig. 3.2. In the karabo device this is done by setting to *True* the boolean **Use Conditional Trigger**, Fig. 3.3, enabling this variable the **Conditional Trigger Mask** will be set, generating the above mentioned logical *and* condition. Please, check with EEE experts before changing this mask.

Fig. 3.3: Out of the initial 10 Hz triggers, the user can select the ones corresponding to specific conditions, for example in pulse on demand situations.

The device provides the user with some useful features which might be used in case of a scan of the trigger delay, Fig. 3.4. A step value (**Step Size**) can be defined, and the delay can be increased (decreased) of one step unit using the slot **StepUp** (**StepDown**). The coarse and fine delay values will be set according to the closest multiple of a basic step. The size of the step value can be doubled or halved using the slots **Step x2** or **Step /2**, respectively.

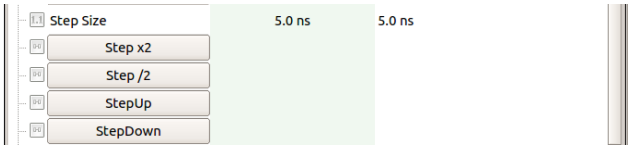

Fig. 3.4: Scan capabilities are provided allowing to modify the trigger delay by predefined steps.

CHAPTER 4

Troubleshooting

The proper working of the middlelayer x2TimerML device depends on the X2Timer device, which in turns relies on the good working of the DOOCS server. One of the typical encountered problems appears when an expected device is not reachable, either because its karabo name was wrongly typed or because the device is not online yet (e.g., it was shutdown). This situation shows up with the middlelayer device remaining in INIT state, and can be cleared by making the expected connected device indeed reachable.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search