

## Scope of this document

The Alvium Sequencer is a very powerful tool to run a sequence of predefined settings at full speed. This document describes how to get started and it lists some best practices to successfully work with the Sequencer.

This document is valid for **firmware 13.xx**.

## In a nutshell

Alvium Sequencer in a nutshell:

Save different camera settings in separate sets and connect them with a Sequencer trigger source. Run the sequence and switch sets in realtime.



### Note:

- **Always configure a user set** that contains all non-sequenceable features and load it as a starting point for Sequencer Configuration and Sequencing.
- **Never** change non-sequenceable features during Sequencer Configuration or Sequencing.
- **Only stop the acquisition** if the Sequencer is not transition from one set to another. Use host controlled signals such as SoftwareSignals or Action Commands and/or configure a parking set, see: [Additional step for a stable Acquisition Stop](#).

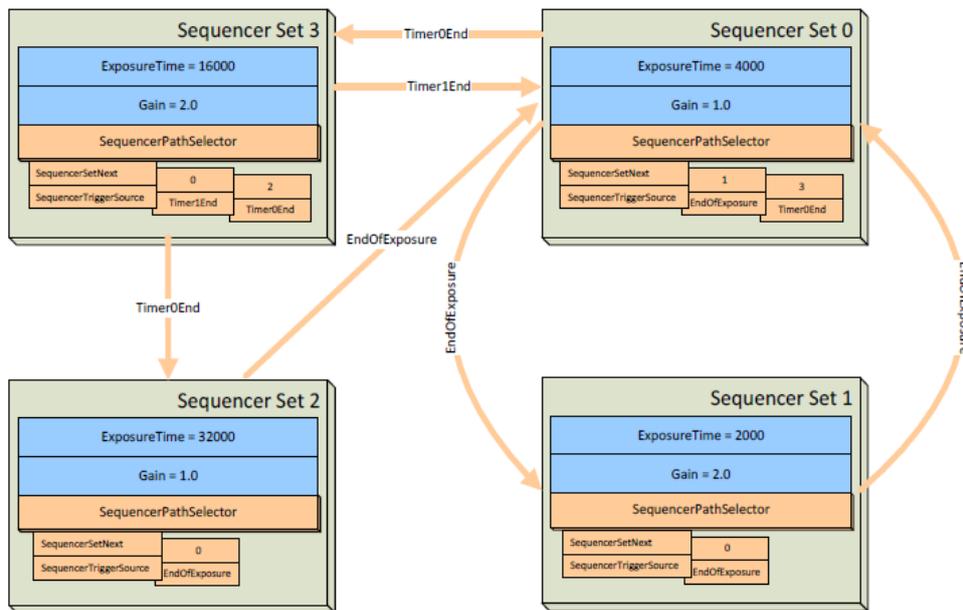
## Technical principles

The Alvium Sequencer is designed to be compliant with the [SFNC](#) (GenICam Standard Naming Convention). You can find a detailed description in the SFNC, especially in chapter *Sequencer Control*.

Additionally to the features defined by the SFNC, Alvium cameras are equipped with the *Sequencer Configuration Reset* command, which deletes all SequencerControl entries and sequencer trigger sources in the camera. Note that *Sequencer Configuration Reset* does not reset the current camera features to default values.

## Sets and paths

The Alvium Sequencer is organized in sets and paths. Each set contains a setting for all sequenceable features. Paths define the order of the sets and the trigger condition to the next set. You can configure a simple sequence with just two sets or more complex sequences as shown in the following image:



Sequencer working diagram (source: SFNC)

## Sequencer features vs. non-Sequencer features

Camera features that are listed in the SequencerControl feature category can be changed within a sequence (feature availability depends on the sensor):

- Gain, ExposureTime
- OffsetX, OffsetY
- AcquisitionFrameRate
- Gamma
- BalanceRatioBlue, BalanceRatioRed
- Hue, Saturation
- ColorTransformationValue, ColorTransformationEnable

**Always configure a user set** that contains all non-sequenceable features **as a starting point for a sequence**. Changing non-sequenceable features during Sequencer Configuration or while Sequencer Mode is *On* may cause a camera crash.

## Sequencer behavior

Additionally to the behavior described in the [SFNC](#), note the following:

- By default, there are no sets in the camera. Save each set to make it available as Start Set and Next Set (by Sequencer default, Set0 is defined as Start Set).
- Whenever a set is entered (including the Start Set), a frame is triggered immediately. Note that the Sequencer deactivates trigger sources that are not Sequencer trigger sources.
- The minimum exposure time within sequences is higher than the regular value. Be aware that extremely short exposure times are unavailable for sequencing.
- If you don't use a host-controllable trigger source such as SoftwareSignal0, see: [Additional step for a stable Acquisition Stop](#)
- Before stopping image acquisition in Sequencer Mode, make sure that the acquired frames are fully transmitted. Consider using a waiting time, which can be determined as follows: *Exposure time + readout time*, where *readout time*  $\approx 1/\text{Acquisition Frame Rate}$

- You can use multiple paths in a sequence. For many use cases, this is not necessary and one path is sufficient. If you do use multiple paths, note:
  - The higher the path number is, the higher is the priority of this path. This is important for time-critical applications, when two triggers may compete.
  - Check all path configurations before saving a set.

## Preparing Sequencer usage

1. Before configuring or activating a sequence, configure all non-Sequencer camera features according to your use case, including the optional chunk configuration.
2. Save your settings as a user set.
3. Always load this user set before Sequencer configuration and before starting a sequence.

If you already have experimented with sequencing and want a fresh start, you can delete all sets in the camera. To do this, use the *Sequencer Configuration Reset* command.

## Configuring a sequence

### Sequencer Configuration

To configure a sequence:

1. Load the user set for your sequence.
2. Switch Configuration Mode *On* and make sure Sequencer Mode is switched *Off*.
3. Configure a sequence (see [Example: Configure a simple sequence](#) ).  
**Never** change non-Sequencer features and their values during Sequencer configuration.

## Example: Configure a simple sequence step by step

To configure a simple sequence (here: two sets with different Gain values), follow these steps:

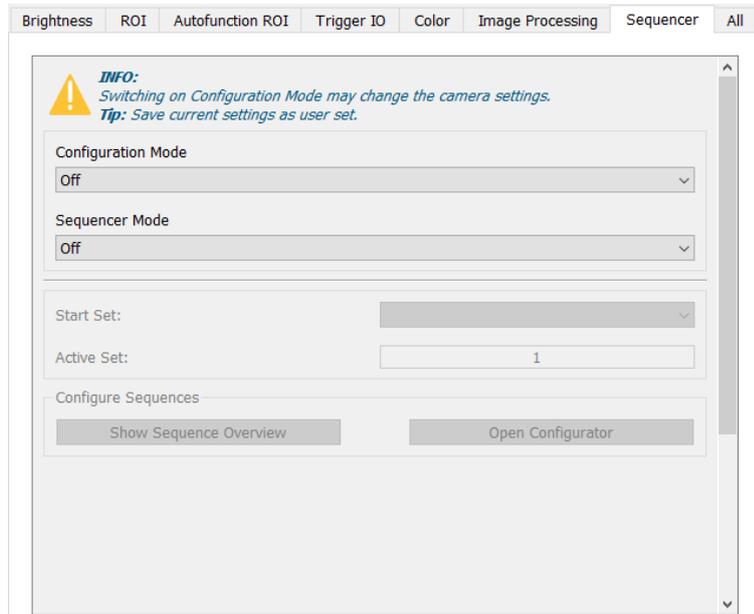
1. Configure, save and load a user set as a starting point for your sequence.
2. In the SequencerControl feature category:
  - a. Switch Sequencer Configuration Mode *On* (make sure Sequencer Mode is switched *Off*).
  - b. By default, Sequencer Set Start is 0. You can keep this value.
  - c. Select a set: Sequencer Set Selector = 0
  - d. Load this set: Sequencer Set Load -> Execute
3. In the AnalogControl feature category: Set Gain = 0
4. In the SequencerPathControl feature category:
  - a. Sequencer Set Next = 1
  - b. Sequencer Trigger Source = Switch *On* and select SoftwareSignal0
5. In the SequencerControl feature category: Execute the *Sequencer Set Save* command.
6. Repeat the steps from 3c. and apply them to Set1. Select Set 0 as the next set and change the Gain value.
7. Switch Sequencer Configuration Mode *Off* and Sequencer Mode *On*.
8. Start image acquisition.
9. Execute SoftwareSignal0 to control the sequence.
10. Stop image acquisition and switch Sequencer Mode *Off*.

Note: In order to set your camera to the state of a particular set, select this set and execute the *Load* command.

## Sequencer Configuration with Vimba X Viewer

**Note:** Even if you use the Sequencer tab, please read [Additional step for a stable Acquisition Stop](#).

To configure a sequence with Vimba X Viewer, use the Sequencer tab. The tab is available as of Vimba X 2023-4.



## Additional step for a stable Acquisition Stop

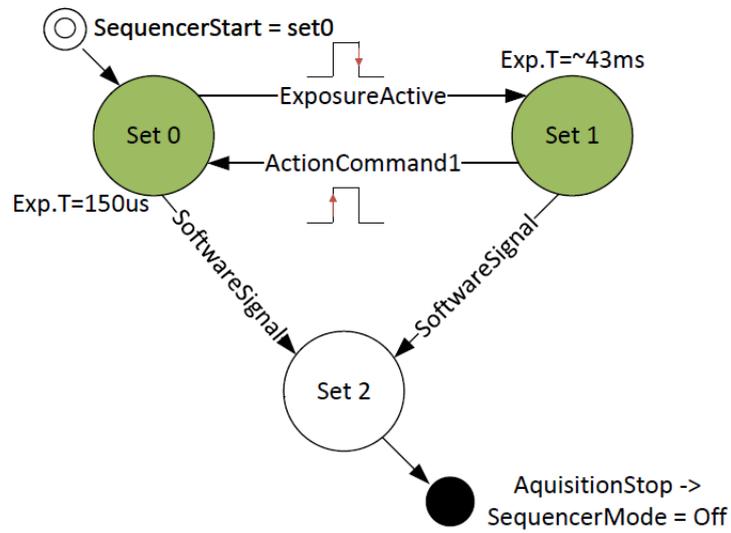
In some cases (depending on the Sequencer trigger source), stopping the acquisition during Sequencer operation may cause a camera crash. This is camera behavior with firmware 13xx.

To avoid this, we recommend using a host-controllable Sequencer trigger source (e.g., `SoftwareSignal` or `ActionCommand`). With a Sequencer trigger source from the camera (e.g., `ExposureActive` or `FrameActive`) please configure an additional step.

For a stable Acquisition Stop, the Sequencer must not be in transition from one set to another. If you experience this behavior:

1. Prepare and configure the sequence as described before.
2. Save one additional set as “parking set”. The Sequencer feature settings do not matter, but configure the parking set with the minimum exposure time (see [Before stopping image acquisition](#))
3. Configure a transition to this parking set with an unused host controllable signal, for example, `SoftwareSignal1` from each set of your sequence.
4. In your application, transition to the “parking set” via an additional path before executing the `AcquisitionStop` command. Make sure this path has the highest path number in your sequence, so that it is executed with the highest priority.

The following image shows how to configure a sequence with two sets and an additional “parking set”:



The following pseudo code example shows in detail how to configure a sequence with two sets and an additional “parking set”:

```

{
  "sets": [
    {
      /* This pseudo code can be used as a starting point for a sequence.
      It shows how to prevent a crash caused by StopImageAcquisition.
      Set0, Set1, and Set2 contain a feature value change (Gain).
      Set3 serves as "parking set" to savely stop image acquisition.
      */
      "SequencerSetSelector": 0,
      "features": {
        "Gain": 3.0
      },
      "paths": [
        {
          /* AcquisitionStop during ExposureActive may cause crash
          "SequencerPathSelector": 0,
          "SequencerSetNext": 1, // Next Set in Path0 = Set1
          "SequencerTriggerActivation": "FallingEdge",
          "SequencerTriggerSource": "ExposureActive"

          },
          {
            /* Additional path leads to parking set to prevent crash.
            Host-controllable trigger source transitions to the parking set
            */
            "SequencerPathSelector": 1,
            "SequencerSetNext": 3, // Next parking set in Path1 = Set3
            "SequencerTriggerActivation": "RisingEdge",
            "SequencerTriggerSource": "SoftwareSignal1"
          }
        ]
      },
      {
        "SequencerSetSelector": 1,
        "features": {
          "Gain": 0.0
        },
        "paths": [
          {
            "SequencerPathSelector": 0,
            "SequencerSetNext": 2, // Next Set in Path0 = Set2
            "SequencerTriggerActivation": "FallingEdge",
            "SequencerTriggerSource": "ExposureActive"
          },
          {
            "SequencerPathSelector": 1,
            "SequencerSetNext": 3, // Next parking set in Path1 = Set3
            "SequencerTriggerActivation": "RisingEdge",
            "SequencerTriggerSource": "SoftwareSignal1"
          }
        ]
      }
    ]
  },
}

```

```
{
  "SequencerSetSelector": 2,
  "features": {
    "Gain": 6.0
  },
  "paths": [
    {
      "SequencerPathSelector": 0,
      "SequencerSetNext": 0, // Next Set in Path0 = Set0
      "SequencerTriggerActivation": "FallingEdge",
      "SequencerTriggerSource": "ExposureActive"
    },
    {
      "SequencerPathSelector": 1,
      "SequencerSetNext": 3, // Next parking set in Path1 = Set3
      "SequencerTriggerActivation": "RisingEdge",
      "SequencerTriggerSource": "SoftwareSignal1"
    }
  ]
},
{
  /* In your application, transition with a host-controllable signal
  (e.g., SoftwareSignal1) to this parking set before executing
  the AcquisitionStop command.
  */
  "SequencerSetSelector": 3,
  "features": {
    "Gain": 0.0
  },
  "paths": [
  ]
}
]
```