Allied Vision

TECHNICAL PAPER

Using Counters and Timers with Alvium Cameras

V1.0.0
Sept 2023

Scope of this document

This document provides an overview of Counter and Timer functionalities with Alvium cameras and Vimba X. For an easy start, Vimba X Viewer provides presets for PWM (Pulse Width Modulation) and a Counter on Line0.

Note: Counters and Timers are unavailable if an Alvium CSI-2 camera is used with V4L2.

Use cases

Use cases for counters and timers described in this document:

- PWM (Pulse Width Modulation), see code snippet on page 3
- A pulse synchronized with a frame using auto exposure, see page 7
- Triggering multiple frames during a Timer signal, see page 12

Use cases for counters and timers available as preset in Vimba X Viewer:

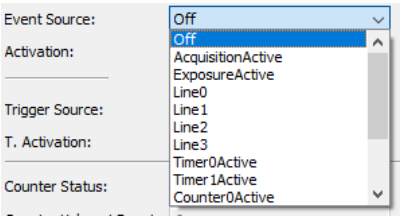
- Pulse Width Modulation with 1 kHz and 50% Duty Cycle:
- Counter IO Line0, counts RisingEdge occurrences on Line0

Other typical use cases for counters and timers in machine vision and embedded vision applications:

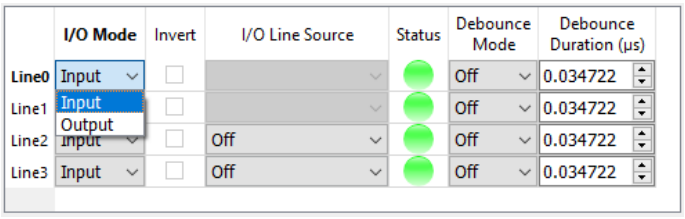
- Delayed triggering of external devices such as another camera or a strobe.
- Image acquisition, triggered through a timer, e.g., every 30 seconds.
- Tracking how often a device was triggered.
- … and more.

Counters and Timers in Alvium cameras

Alvium cameras with GenAPI support are equipped with four Counters. For each Counter, the user can choose between 14 possible Counter Event Sources. The two Timers can be triggered by one of 14 different sources. For details, see the Alvium Features Reference.



The I/O Mode of the four available lines can be switched, so that most of them can serve as input or output (depending on the camera interface).



Getting started with Counters and Timers

Prerequisites

For an easy start with Counters and Timers, you need:

- Alvium GigE, USB, or 1800-C cameras with firmware 11.xx or higher with counter and timer features
- Vimba X SDK
- Optional: Cable for triggering external devices such as a strobe

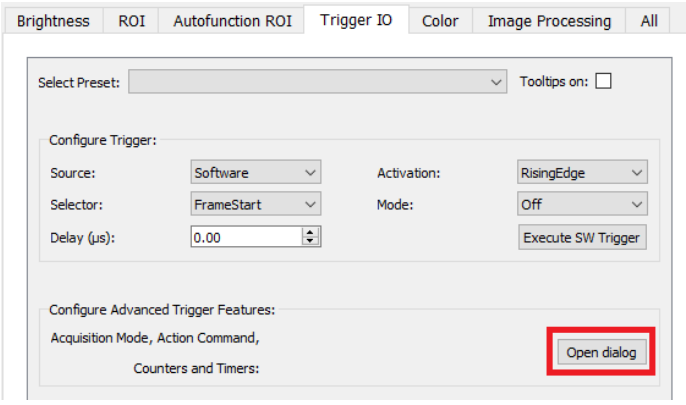
Get started with Vimba X Viewer

Step 1: Get your camera up and running

- Check the firmware version with Vimba X Firmware Updater or Vimba X Viewer. If a newer firmware is available, update it. We recommend using the latest firmware.
- Start Vimba X Viewer, acquire some images, and apply the basic camera settings for your application such as the exposure time. GigE cameras: For best performance, follow the instructions of the user guide, chapter Configuring the host computer.

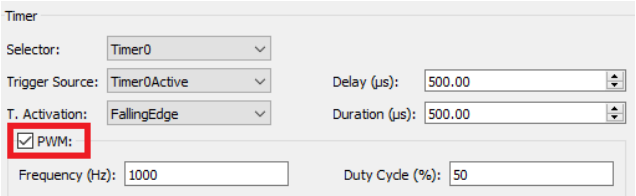
Step 2: Explore the presets for Counter and Timer

On the *Trigger IO tab*, select Open dialog .

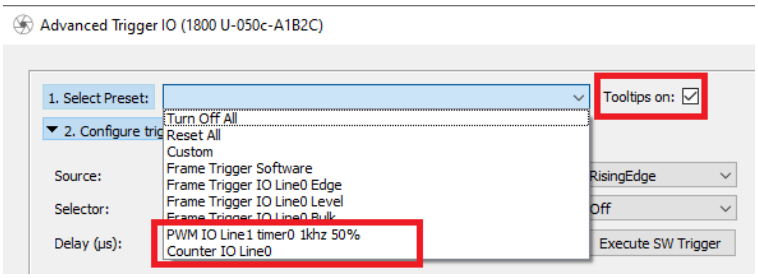


On the **Advanced Trigger dialog** (not available for CSI-2, please use the **All** tab), select the PWM preset or the Counter IO preset. Please note that PWM is not a specific camera feature, but a special configuration of the Timer, where falling edge of the same timer is used as trigger activation. You can

use the Advanced Trigger IO dialog to activate PWM.



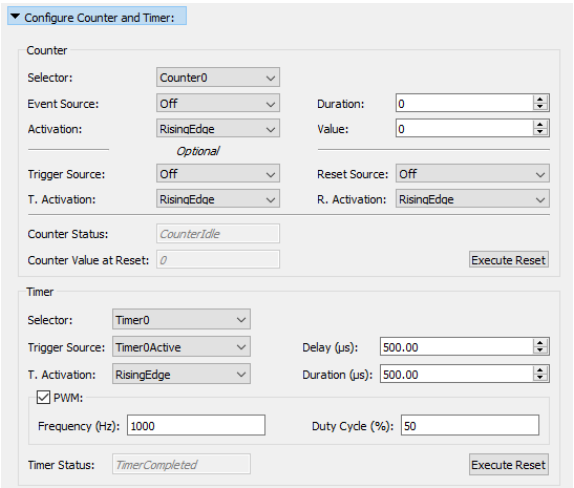
To see feature descriptions as you hover, select *Tooltips on*.



Or skip the presets and proceed with the next step.

Step 3: Adjust the settings

Expand the Counter and Timer section, try the preset, and adapt the settings according to your use case.



Get started with programming

On the following pages, you can find an **example for configuring a Timer with PWM output and a Counter for this signal**.

To simplify the example, no image acquisition is included and only limited error handling is done.

Examples including images acquisition and error handling come with the Vimba X SDK.

```
 // Configure a Timer with PWM output and a Counter for this signal
#include <iostream>
#include "VmbCPP/VmbCPP.h"
using namespace VmbCPP;

int main() {

    VmbSystem& system = VmbSystem::GetInstance();
    if (VmbErrorSuccess == system.Startup())
    {
        CameraPtrVector cameras;
        if (VmbErrorSuccess == system.GetCameras(cameras))
        {
            CameraPtr camera = cameras[0];
            if (VmbErrorSuccess == camera->Open(VmbAccessModeFull))
            {
                std::cout << "Camera is configured..." << std::endl;
                // Set up Timer features
                double timerDuration = 5000;
                double timerDelay = 5000;
                FeaturePtr pfeature;

                camera->GetFeatureByName("TimerSelector", pfeature);
                pfeature->SetValue("Timer0");  // Select Timer0
                camera->GetFeatureByName("TimerTriggerSource", pfeature);
                pfeature->SetValue("Timer0Active");
                camera->GetFeatureByName("TimerTriggerActivation", pfeature);
                pfeature->SetValue("FallingEdge");  // For PWM generation, Timer gets triggered with
                falling edge  from itself.
                camera->GetFeatureByName("TimerDuration", pfeature);
                pfeature->SetValue(timerDuration);
                camera->GetFeatureByName("TimerDelay", pfeature);
                pfeature->SetValue(timerDelay);  // 100Hz 50% Duyty Cycle

                // Set up signal on I/O Line0
                camera->GetFeatureByName("LineSelector", pfeature);
                pfeature->SetValue("Line0");  //  Select Line0
                camera->GetFeatureByName("LineMode", pfeature);
                pfeature->SetValue("Output");
                camera->GetFeatureByName("LineSource", pfeature);
                pfeature->SetValue("Timer0Active");  // Output Timer0 signal
```

```
// Set up Counter features
int counterDuration = 10000;

camera->GetFeatureByName("CounterSelector", pfeature);
pfeature->SetValue("Counter0");
camera->GetFeatureByName("CounterEventSource", pfeature);
pfeature->SetValue("Timer0Active");  // Line0 can be counted even if it is output
camera->GetFeatureByName("CounterEventActivation", pfeature);
pfeature->SetValue("RisingEdge");
camera->GetFeatureByName("CounterTriggerSource", pfeature);
pfeature->SetValue("Off");  // If Off, reset feature starts the counter
camera->GetFeatureByName("CounterDuration", pfeature);
pfeature->SetValue(counterDuration);  // Counter will end at this. Value must be set!
camera->GetFeatureByName("CounterReset", pfeature);
pfeature->RunCommand();  // Start counter

camera->GetFeatureByName("TimerReset", pfeature);
pfeature->RunCommand(); // Start PWM Signal
```

```
                // Output counter Values
                std::cout << "Camera runs PWM on Timer0. It can be measured on Line0 and rising edges of the PWM Signal are counted."<< std::endl;
                std::cout << "Camera will stop counting at: " <<counterDuration << std::endl;
                std::cout << "Press 's' to show counter value." << std::endl;
                std::cout << "Press 'r' to reset counter." << std::endl;
                std::cout << "Press 'q' to leave." << std::endl;
                std::string input;
                VmbInt64_t counterValue;
                while (true) {
                    std::getline(std::cin, input);
                    if (input == "s") {
                        camera->GetFeatureByName("CounterValue", pfeature);
                        pfeature->GetValue(counterValue);
                        std::cout << "Counter value: " << counterValue << std::endl;
                    }
                    if (input == "r") {
                        camera->GetFeatureByName("CounterReset", pfeature);
                        pfeature->RunCommand();  // Start counter
                        camera->GetFeatureByName("CounterValueAtReset", pfeature);
                        pfeature->GetValue(counterValue);
                        std::cout << "Counter value at reset: " << counterValue << std::endl;
                    }
                    if (input == "q") {
                        break;
                    }
                }
            }
            camera->Close();
        }
    }
    system.Shutdown();
    return 0;
}
```

Use case: Pulse synchronized with a frame

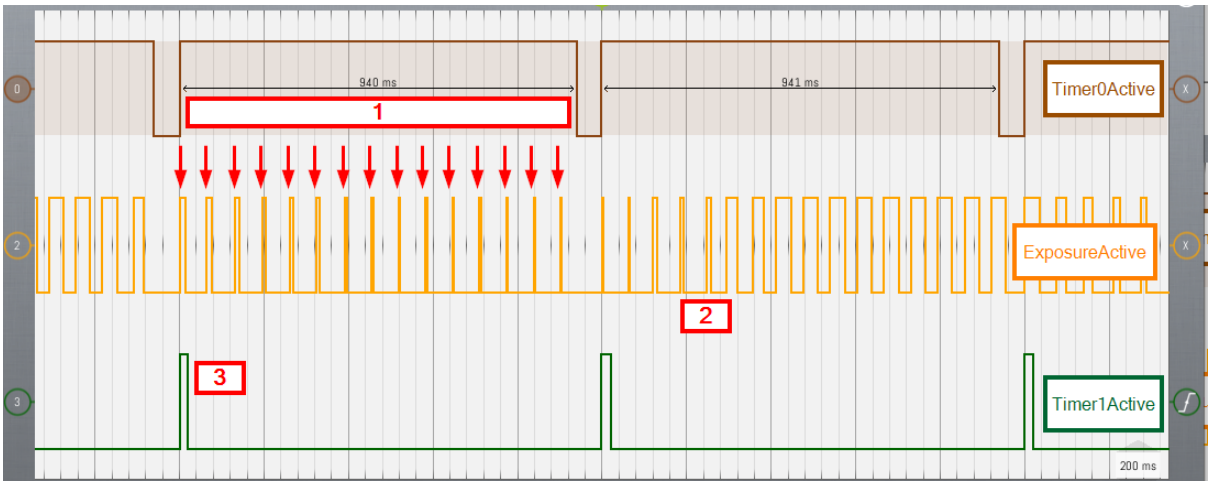
In this use case, we have one pulse per second and simultaneuously an image with auto exposure.

Tip:

Auto exposure works best if it updates often, so acquiring with just 1 fps is not recommended. In this concrete example, the camera (an Alvium U-1236) runs at its free-run rate, which is approximately 15.7 fps. Of course other frame rates work as well, if you have a different camera model.

Only the image synchronized to the pulse is used, the other images are ignored.

Method A: Using Timer0 and Timer1



1 Timer0 enables frames to be produced while it is active. The first frame is started at rising edge of Timer0. Timer0 has a loop time of 1 s (to an accuracy of +/- 100 ns, roughly).

2 The width of ExposureActive changes because auto exposure is on and light conditions vary.

3 Timer1 outputs a 20 ms pulse at rising edge of Timer0. If the rising edge of Timer0 is used, this is optional in most cases.

Tip:

Timer0 'off' time (Timer0Delay) should be roughly one frame time to ensure that there is synchronisation between the first frame and the rising edge of Timer0. This may need to be tweaked for a different camera, it just needs to be long enough to prevent a new frame occurring too close to the rising edge of Timer0.

In general, if your frame rate is NN.FFF frames/s, then (0.FFF * Frame time) gives a first pass at the Timer0 'off' (Timer0Delay) time. You may need to make it a little longer to prevent errors.

Timer settings of this example (please change according to your camera and use case), changes from default settings are highlighted:

CounterAndTimerControl

Counter Duration	0
Counter Event Activation	RisingEdge
Counter Event Source	Off
Counter Reset	[COMMAND]
Counter Reset Activation	RisingEdge
Counter Reset Source	Off
Counter Selector	Counter0
Counter Status	CounterIdle
Counter Trigger Activation	RisingEdge
Counter Trigger Source	Off
Counter Value	0
Counter Value At Reset	0
Timer Delay	60000
Timer Duration	940000
Timer Reset	[COMMAND]
Timer Selector	Timer0
Timer Status	TimerActive
Timer Trigger Activation	LevelLow
Timer Trigger Source	Timer0Active

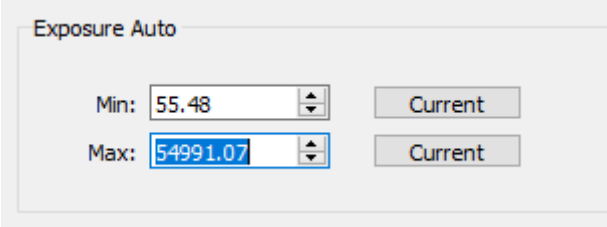
Timer0 Setup

CounterAndTimerControl

Counter Duration	0
Counter Event Activation	RisingEdge
Counter Event Source	Off
Counter Reset	[COMMAND]
Counter Reset Activation	RisingEdge
Counter Reset Source	Off
Counter Selector	Counter0
Counter Status	CounterIdle
Counter Trigger Activation	RisingEdge
Counter Trigger Source	Off
Counter Value	0
Counter Value At Reset	0
Timer Delay	0
Timer Duration	20000
Timer Reset	[COMMAND]
Timer Selector	Timer1
Timer Status	TimerCompleted
Timer Trigger Activation	RisingEdge
Timer Trigger Source	Timer0Active

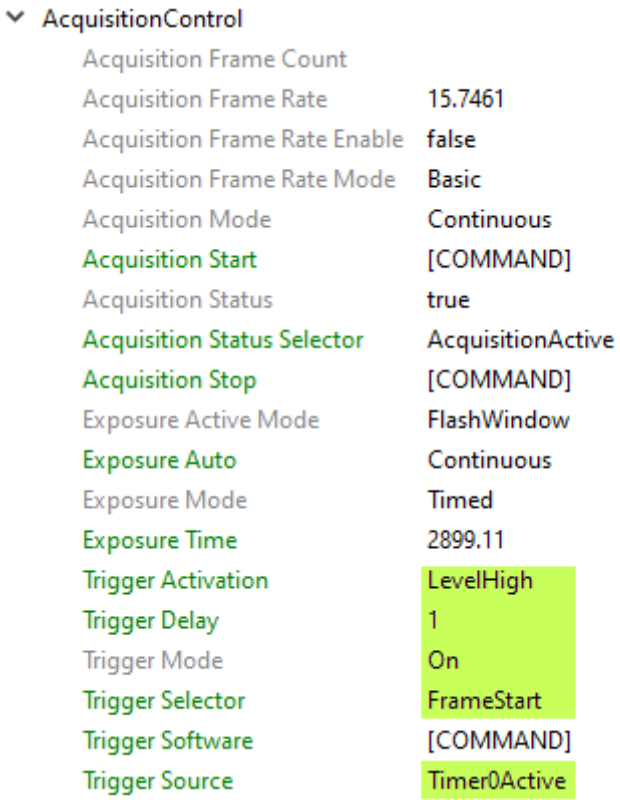
Timer1 Setup

Make sure that the maximum exposure time in the Auto exposure settings prevent frames being extended possibly affecting the overall timing. In this case, we used 55 ms:

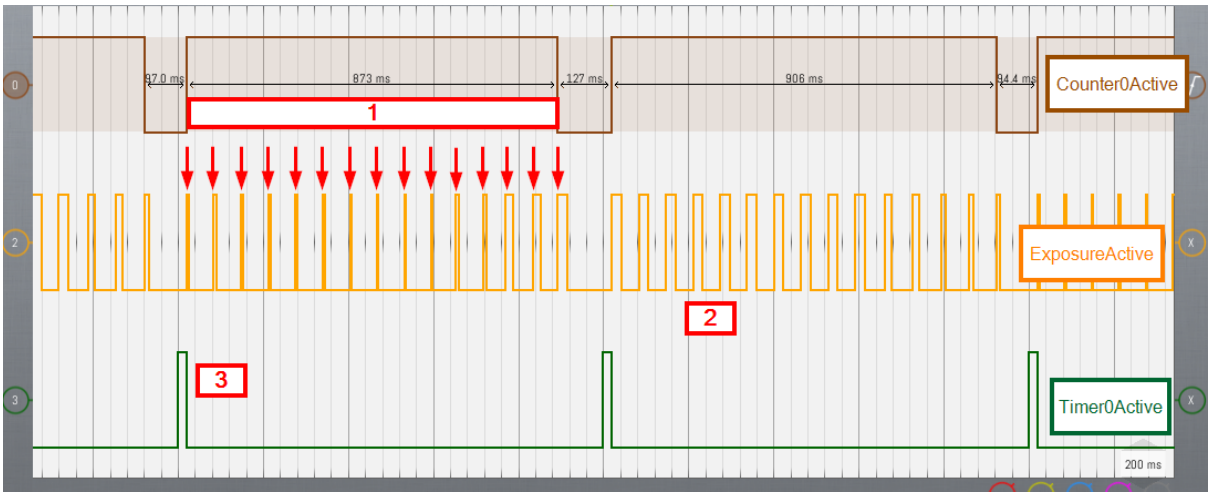


Optionally, you can set Timer1 to give a 20 ms pulse when Timer0 becomes active. Depending on your hardware and use case, this may not always be needed.

Note that Trigger Delay = 1. This is needed, otherwise the trigger signal will miss the internal clock transition.



Method B: Using a Counter and a Timer



1 Counter0 enables frames to be produced while it is active. Counter0 starts to count at the rising edge of Timer0Active and continues to count until it gets to 15. Once it gets to 15, it stops counting and becomes inactive until the next rising edge of Timer0.

2 The width of ExposureActive changes because auto exposure is on and light conditions vary.

3 Timer0 is set to emit a 20 ms pulse every second. Timer0 has two roles. Firstly, to control Counter0 and secondly, to output one pulse per second. The length of this pulse is not critical but, as the falling edge set Counter0 off, it should not be too long. If the frame rate is NN.FFF frames/s then it should be shorter than (0.FFF * Frame time). In this case, at 15.7461 frames/s, it should be shorter than (0.7461 * 1/15.7461) s = 0.7461 * 0.0635 s = 47.4 ms

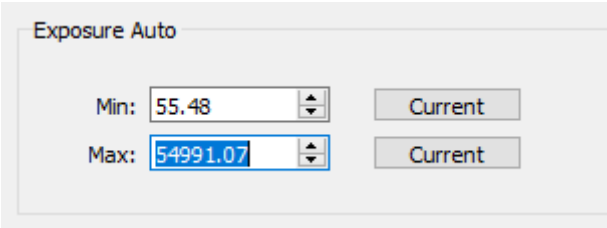
Note: The target for Counter0 needs to be set to the number of complete frames that the camera will output a second. If the frame rate is NN.FFF frames/s, then NN is the target for the counter. In this example, the frame rate is 15.7461 frames/s so 15 is Counter0's target.

Counter and Timer settings of this example (please change according to your camera and use case), changes from default settings are highlighted:

Counter Duration	15	Counter0 Setup
Counter Event Activation	RisingEdge	
Counter Event Source	ExposureActive	
Counter Reset	[COMMAND]	
Counter Reset Activation	RisingEdge	
Counter Reset Source	Timer0Active	
Counter Selector	Counter0	
Counter Status	CounterActive	
Counter Trigger Activation	FallingEdge	
Counter Trigger Source	Timer0Active	
Counter Value	7	
Counter Value At Reset	15	
Timer Delay	980000	Timer0 Setup
Timer Duration	20000	
Timer Reset	[COMMAND]	
Timer Selector	Timer0	
Timer Status	TimerDelay	
Timer Trigger Activation	LevelLow	
Timer Trigger Source	Timer0Active	

Alternatively, you can set CounterTriggerActivation to RisingEdge (at the same time as CounterResetActivation being set to RisingEdge).

Just like with Method A, make sure that the maximum exposure time in the Auto exposure settings prevent frames being extended possibly affecting the overall timing. In this case, we used 55 ms:

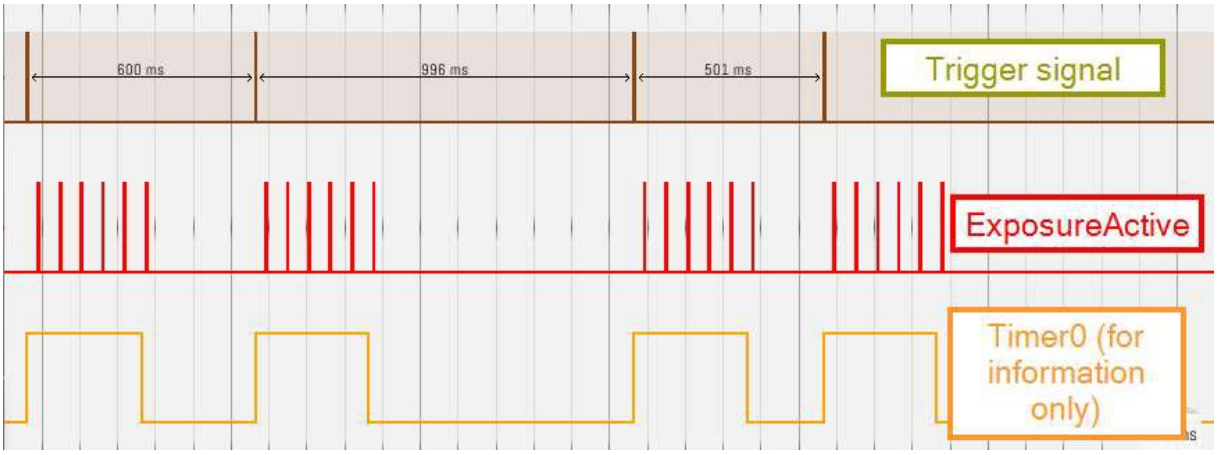


Allied Vision

Camera	
AcquisitionControl	
Acquisition Frame Count	
Acquisition Frame Rate	15.7461
Acquisition Frame Rate Enable	false
Acquisition Frame Rate Mode	Basic
Acquisition Mode	Continuous
Acquisition Start	[COMMAND]
Acquisition Status	true
Acquisition Status Selector	AcquisitionActive
Acquisition Stop	[COMMAND]
Exposure Active Mode	FlashWindow
Exposure Auto	Continuous
Exposure Mode	Timed
Exposure Time	2033.66
Trigger Activation	LevelHigh
Trigger Delay	0
Trigger Mode	On
Trigger Selector	FrameStart
Trigger Software	[COMMAND]
Trigger Source	Counter0Active

Use case: Triggering multiple frames during a Timer signal

Here you can see an example of how features are set when a random trigger signal is used to acquire frames during a Timer signal. Please adapt the settings and values to your camera model and use case.



CounterAndTimerControl	
Timer Delay	40
Timer Duration	300000
Timer Reset	[COMMAND]
Timer Selector	Timer0
Timer Status	TimerCompleted
Timer Trigger Activation	RisingEdge
Timer Trigger Source	Line1

Feature	Value
AcquisitionControl	
Acquisition Frame Count	
Acquisition Frame Rate	17.5429
Acquisition Frame Rate Enable	false
Acquisition Frame Rate Mode	Basic
Acquisition Mode	Continuous
Acquisition Start	[COMMAND]
Acquisition Status	true
Acquisition Status Selector	AcquisitionActive
Acquisition Stop	[COMMAND]
Exposure Active Mode	FlashWindow
Exposure Auto	Off
Exposure Mode	Timed
Exposure Time	28913
Trigger Activation	LevelHigh
Trigger Delay	1
Trigger Mode	On
Trigger Selector	FrameStart
Trigger Software	[COMMAND]
Trigger Source	Timer0End