Teledyne DALSA • 605 McMurray Road • Waterloo, Ontario, N2V 2E9 • Canada
https://www.teledynedalsa.com/

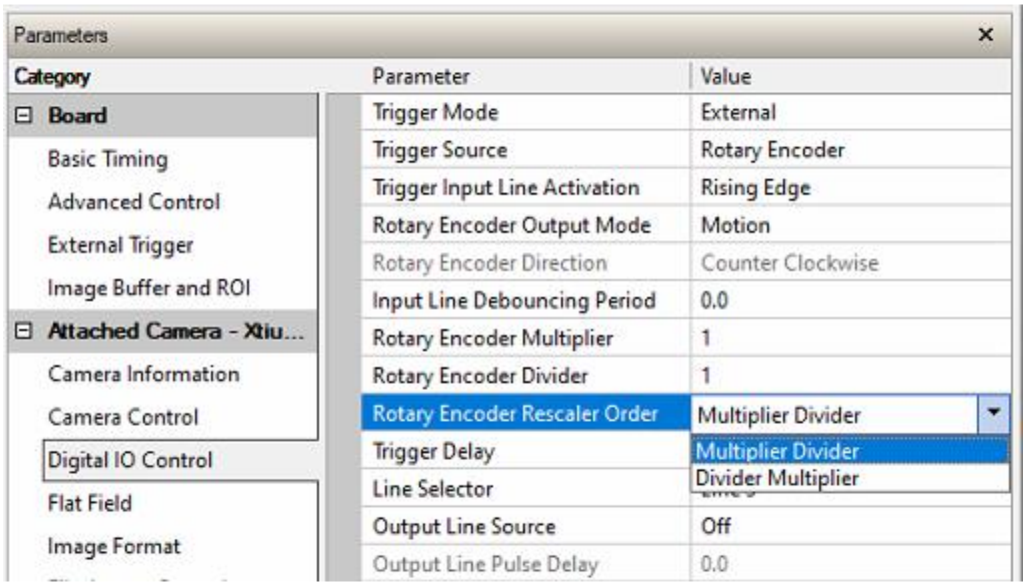
Application Note for Multiplier and Divider

Overview

Have you ever wondered about the two choices you have when dealing with Exsync? They are:

Option 1: Apply multiplier first and then apply divider *(m-d)*.

Option 2: Apply divider first and then apply multiplier *(d-m)*.


Parameters

Category
- Board
  - Basic Timing
  - Advanced Control
  - External Trigger
  - Image Buffer and ROI
- Attached Camera - Xtiu...
  - Camera Information
  - Camera Control
  - Digital IO Control
  - Flat Field
  - Image Format

Parameter	Value
Trigger Mode	External
Trigger Source	Rotary Encoder
Trigger Input Line Activation	Rising Edge
Rotary Encoder Output Mode	Motion
Rotary Encoder Direction	Counter Clockwise
Input Line Debouncing Period	0.0
Rotary Encoder Multiplier	1
Rotary Encoder Divider	1
Rotary Encoder Rescaler Order	Multiplier Divider
Trigger Delay	
Line Selector	[illegible]
Output Line Source	Off
Output Line Pulse Delay	0.0

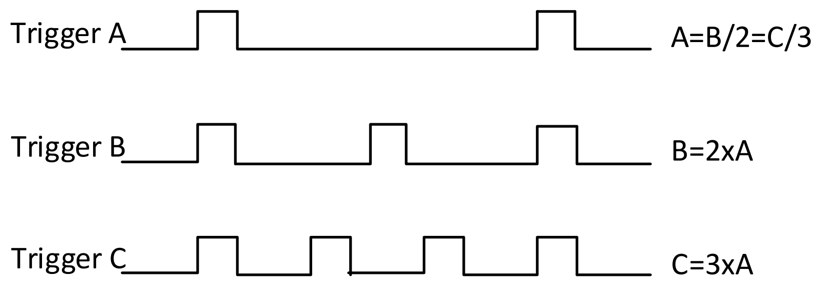
The question arises: What's the difference between them? This document aims to provide the answer to this question, along with application guidance.

How Multiplier and Divider Work?

Multiplier: The camera measures the previous period and inserts a number of (m-1) pulses between every two pulses evenly.

Divider: Takes out every (d-1) pulses.

These two operations are opposites. The diagram below illustrates examples of multiplier and divider operations.



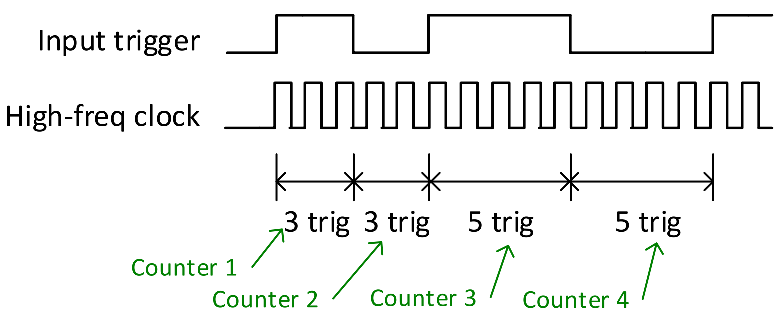
Inserting one (2-1) pulse between every two pulses of trigger A becomes trigger B; inserting two (3-1) pulses between every two pulses of trigger A evenly creates trigger C. Frequency-wise, they result in B=2xA and C=3xA. This is how the multiplier works.

Reversing this operation, removing every other pulse from B or removing two pulses from every three pulses of C, results in trigger A. Frequency-wise, they result in A=B/2 and A=C/3. This is how the divider works.

Does the Order Matter?

From a mathematical point of view, (m-d) or (d-m) does not make any difference; in the world of electronics, however, the two orders can lead to different results.
The (m-d) can track faster because more pulses are added, reacting quickly. On the other hand, the (d-m) may be good for preventing jitters because unnecessary pulses (excessively crowded pulses) are taken out in advance. Both options may have limitations in use, and good results can be achieved when used appropriately.
The camera operates the triggers in this way: first, the camera maps some ongoing periods of the input trigger with an internal high-frequency clock (e.g., 90 MHz) periods; next, it remembers and calculates

the numbers of high-frequency periods mapped to each period with counters. The following drawing explains the mapping principle.



In the above example, the first input period is mapped to 6 (3x2) high-frequency pulses, and the second input period is mapped to 10 (5x2) high-frequency pulses. All pulses' counting is stored in counters cycle by cycle.

Let's look at a real example. One period of a 10 KHz trigger is mapped to 9,000 (90,000,000/10,000) of 90 MHz periods, and this number is remembered by a counter or so and used for later calculations. The accuracy of above mapping is 1/9000. When the input trigger frequency is quite fast, say 100 KHz, the accuracy will drop to 1/900, and if you apply a multiplier of 128 (maximum available), the accuracy will drop further to 1/7. This is a disadvantage when you apply (m-d) to a high-frequency input trigger. Another problem here could be counters' clipping issue as counters also have a limit to the size of numbers they can handle.

On the other hand, if the input is comparatively slow, for example, 2000 Hz, and you divide it by 255 (maximum available), then the period would be 127.5 ms. When it comes to machine vision systems that require precise inspection, this cycle is not short. There is no guarantee that the objects' movement is uniform within such a large period. In other words, in this case, the inserted pulses may be spaced unevenly. There could be counters clipping issue as well in this scenario because the number of clocks could be extremely large. Note that there is no clear-cut watershed between the high and low frequencies covered here; as a rule of thumb, however, when the input trigger frequency is lower than 100 KHz, you may consider choosing (m-d); otherwise, you may consider choosing (d-m). Test it!

How to Approach the Targeted Linerate?

There are exceptions, of course, but most applications deal with images as a 1:1 aspect ratio, aka square images, because it is intuitive and easy to process. In the case of TDI imaging, a square image is not an option but a necessity.

Let’s look at an example. Assume you already know that 1.75x(input trigger) produces a square image. So, you need to find the best-fit parameters, the m and the d. The following table shows some possible options.

Target	x1.75							
m	2	4	8	16	32	64	m_1=128	m_2=256
d	1	3	5	9	18	37	d_1=73	d_2=146
Result	2	1.33	1.6	1.78	1.77	1.7297	1.7534	1.7534

As you can see, the closest result is 1.7534, and two sets of parameters, (m1, d1) and (m2, d2), can generate that same result. Therefore, you have four options:

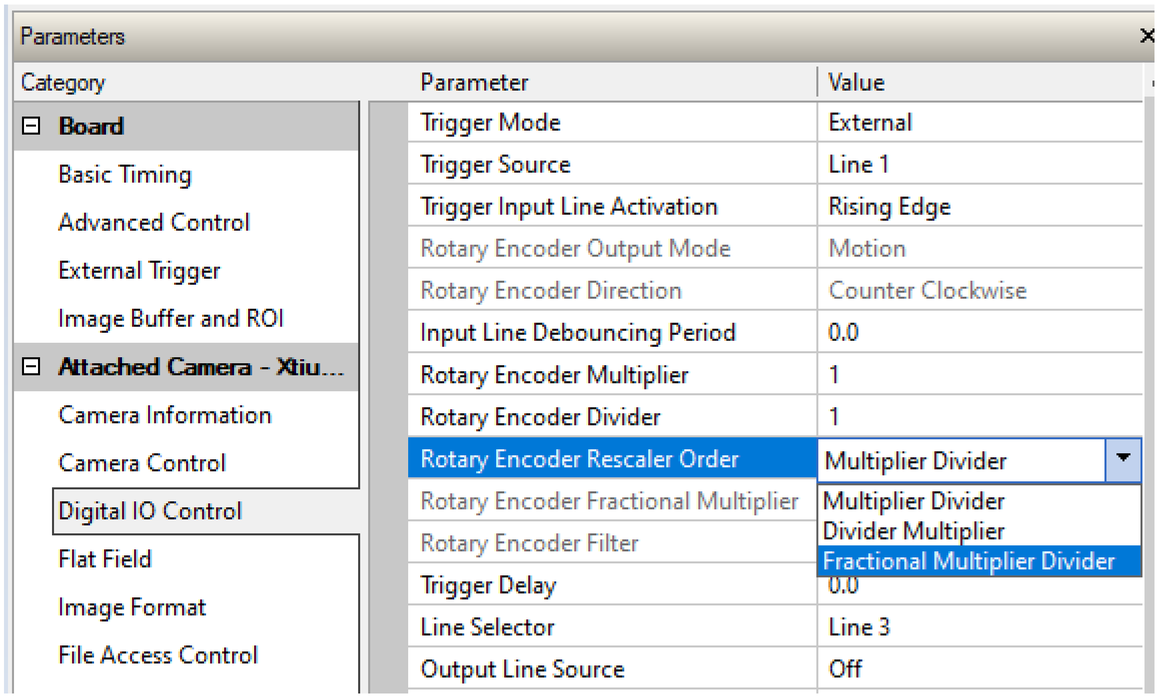
1. (m1 – d1), in a comparatively low frequency, this can be the first option.
2. (d1 – m1), in a comparatively high frequency, this should be the first option.
3. (m2 – d2), do not use it when smaller parameters are available because smaller parameters cause fewer problems.
4. (d2 – m2), again, do not use it when smaller parameters are available, same reason as above.

Let's assume that the target coefficient this time is 1.72. The following table shows some combinations of multipliers and dividers.

Target	x1.72							
m	2	4	8	16	32	64	128	256
d	1	3	5	9	18	37	74	149
Result	2	1.33	1.6	1.78	1.77	1.7297	1.7297	1.71812

As you can see, the combination, (m=256, d=149), gives the best results, so, of course, you should choose that one. Here, the value of m is the maximum, so it would be better to choose (d-m) other than (m-d), which may create more jitters and clip counters. Again, test it!
You may have realized that all the multiplier and divider numbers you can choose are integers. As such, the accuracy of the (m-d) or (d-m) is zero decimal places. As the above example shows, you are looking for 1.72, but the system never gives you the exact 1.72. If this error is outside the allowable range, it becomes a problem.

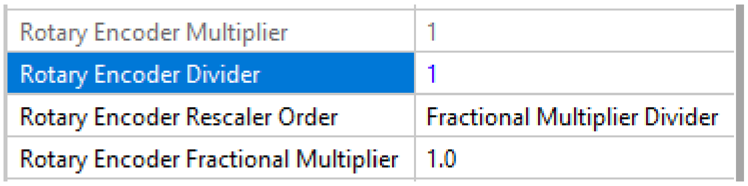
Fortunately, there is a solution to this problem, that is, the so-called **fractional multiplier** (*fm*). The Linea ML and HL family cameras are equipped with it. This function not only increases the accuracy of the trigger to **two decimal places** but also makes it easier to use.



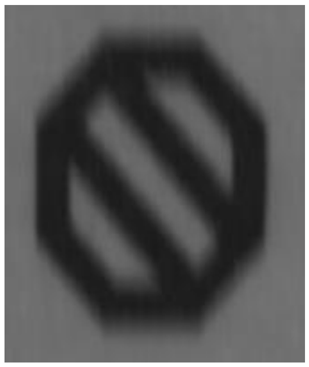
Parameter	Value
Trigger Mode	External
Trigger Source	Line 1
Trigger Input Line Activation	Rising Edge
Rotary Encoder Output Mode	Motion
Rotary Encoder Direction	Counter Clockwise
Input Line Debouncing Period	0.0
Rotary Encoder Multiplier	1
Rotary Encoder Divider	1
Rotary Encoder Rescaler Order	Multiplier Divider
Rotary Encoder Fractional Multiplier	
Rotary Encoder Filter	
Trigger Delay	0.0
Line Selector	Line 3
Output Line Source	Off

With the fractional multiplier activated, it makes your work to search an optimistic parameter much easier than ever. You no longer need to struggle with multiple multiplier and divider combinations, don't even need to worry about the order of (m-d) and (d-m) because the camera handles the complicated tasks in the background. Instead, just need you to type in your target value, for example, 1.72, directly to the Rotary Encoder Fractional Multiplier dialog box. Of course, finding the optimal target value requires repeated tests.

When starting from scratch, you can start with the initial set, (m=1, d=1, fm=1.0), with which, you will probably get shrunk or inflated images.



Rotary Encoder Multiplier	1
Rotary Encoder Divider	1
Rotary Encoder Rescaler Order	Fractional Multiplier Divider
Rotary Encoder Fractional Multiplier	1.0

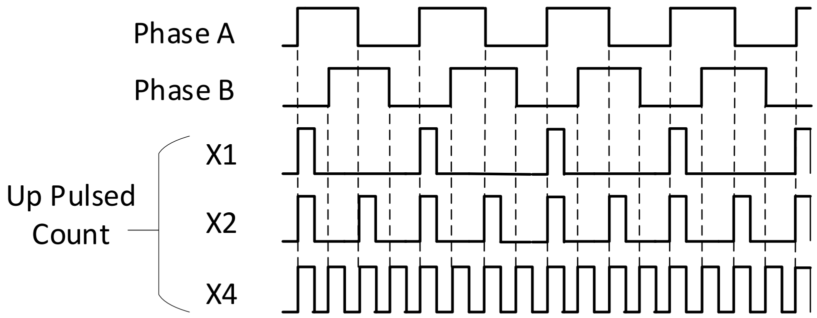


To get a square image, require you to elaborate the trigger frequencies. To do so, you need to follow below two steps:

Step 1: Rough adjustment – adjust the trigger source, encoder for example.

Step 2: Fine-tune – adjust the fm.

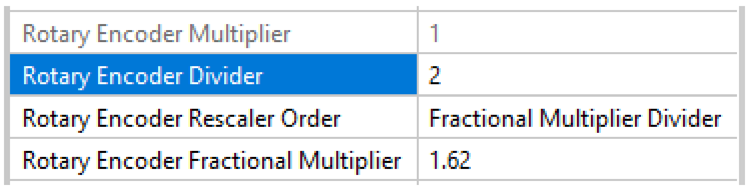
In the case of Line trigger, the frequencies can be doubled with Any Edge (using both rising and falling edges) option compared to using only a single edge (rising or falling edge). In the case of the encoder, the frequencies can be quadrupled with two phases and both edges.



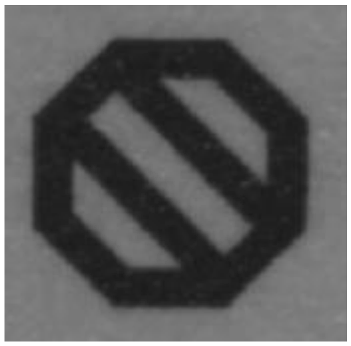
The following is an example showing the process of finding the optimal parameters.

1. (Encoder with Phase A/B). Any Edge, and fm=1.0. This gives an inflated image – the trigger frequency is too high.
2. Rough adjustment - Change the Any Edge to Rising Edge in both Phase A and B, fm remains unchanged. The image still is inflated.
3. Rough adjustment - Change the divider from 1 to 2, fm remains unchanged. This time the image is shrunk.
4. Fine adjustment - Try different fm, e.g., 1.50, 1.60, 1.70…, among them 1.60 gives the best result.
5. Fine-tune. Try 1.61, 1.62, 1.63, …, among them 1.62 gives the best result, ergo, the optimal parameter is found.

As a result, a square image is generated with the optimal parameter set, (m=1, d=2, fm=1.62).



Rotary Encoder Multiplier	1
Rotary Encoder Divider	2
Rotary Encoder Rescaler Order	Fractional Multiplier Divider
Rotary Encoder Fractional Multiplier	1.62



Note:

1) Currently (as of 2023) the fractional multiplier is implemented only in Linea ML and Linea HL family cameras, and it works only when the Exsync is applied to the camera I/O directly.
2) Currently the fractional multiplier does not work if the Exsync is applied via a frame grabber.

Further Supports

Should you have any questions, please feel free to contact your local TCS (Technical Customer Support) teams.

DATE	Revision	Action	Originator	Note
15/12/23	00	Initial Release	Jake Liu	