



Haptics Generator v3.9

Contents

1	Introduction	2
2	Parameter Description	3
2.1	Device	3
2.2	Graph Settings	3
2.3	Parameter (Graph)	3
2.3.1	Forward / Backward	3
2.3.2	Velocity Compensation	4
2.4	Lookup Table Info	4
3	Export / Import	4
4	Graph Editing	4
5	Other Applications	5
5.1	HAPTICORE Customer Test App	5
6	Troubleshooting	6

1 Introduction

The *HAPTICORE Haptics Generator* is an application for graphically modeling and configuring repetitive haptic feedback patterns. This way, it's possible to create custom haptics for both the forward and backward rotation of the HAPTICORE rotary encoder. The desired pattern is then repeated based on a set rotation angle. It's also possible to configure velocity compensation, which influences the strength of the haptic feedback based on the HAPTICORE's rotation speed.

You can either transfer your created configurations directly to the HAPTICORE target device for testing, or export them as C source code to integrate them into the firmware.

Figure 1 shows an overview of the application window. Figure 2 depicts the selected haptic pattern in respect to a full, 360 degree rotation of the knob.

1. Device: Select the target device from the list and *Download* to transfer data to it.
2. Graph Settings: Settings, which have an influence on the graphs in general.
3. Parameter (Forward / Backward / Velocity Compensation): These parameters target one graph specifically.
4. Lookup Table Info: Displays metrics regarding the size of each graph in the firmware's lookup table.
5. Export / Import: Persist graphs and settings by exporting / importing or export directly to C code for direct use within firmware.
6. Tabs to select the different graphs available.
7. Display information about the application or close it.

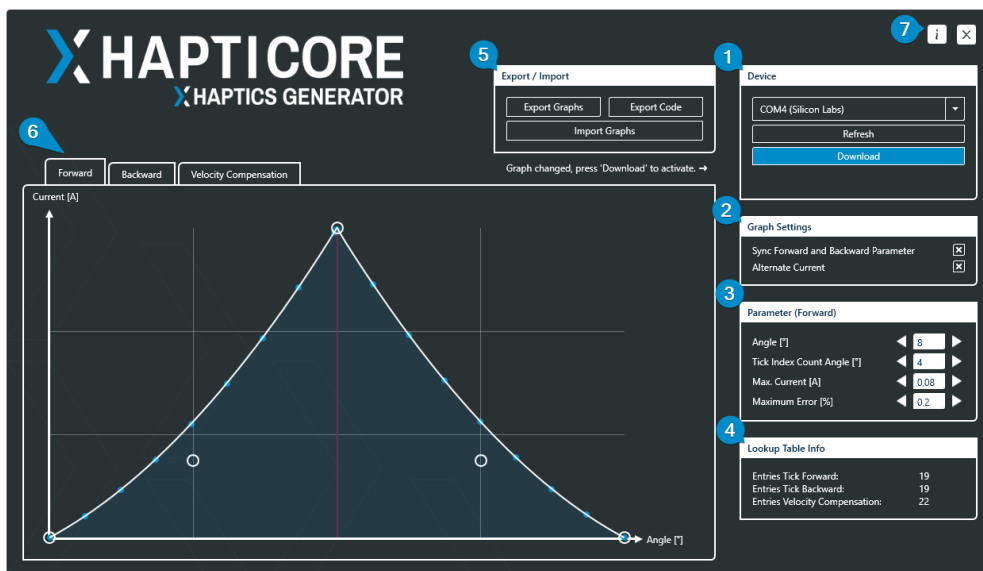


Figure 1: The Haptics Generator application window

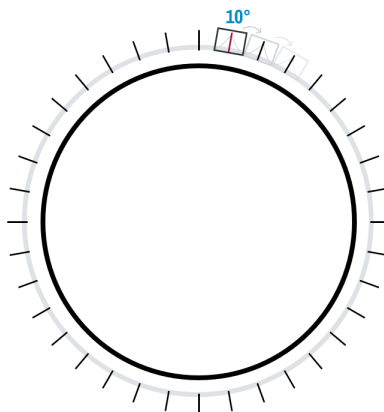


Figure 2: A sample haptic pattern based on a 360° revolution

2 Parameter Description

This section describes all available parameters. Please note that, after each change of a graph as well as of a parameter, the current configuration has to be transferred to the device. This is indicated by the *Download* button, highlighted in blue, and the appearing prompt as shown in figure 3.

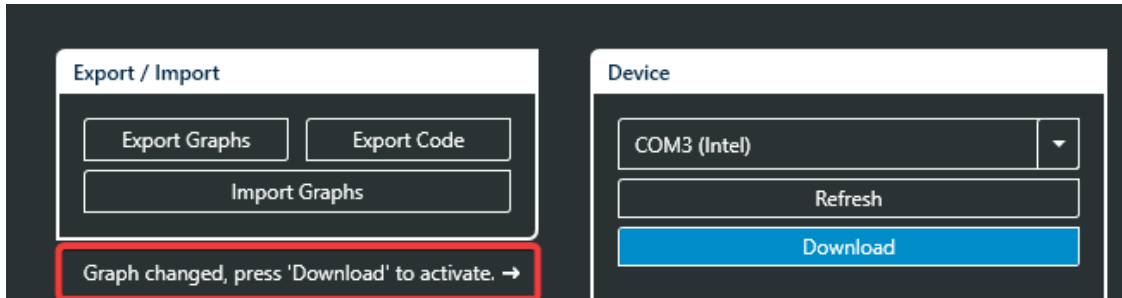


Figure 3: The download prompt, when changes are present

2.1 Device

The first drop-down displays the serial port of the selected HAPTICORE device. If the desired port isn't listed, the *Refresh* button fetches and displays all available serial devices again. Press the *Download* button to transfer and verify the current settings and graph data onto the selected device.

2.2 Graph Settings

The *Sync Forward and Backward Parameter* setting enables syncing of the graph as well as the according parameter between the forward and backward rotation direction. If the graph changes when this setting is enabled, the according setting (or graph change) is mirrored to the other direction automatically.

The *Alternate Current* setting changes the current polarity for each other repetition. Therefore, effects like residual magnetism are minimized. It is advised to keep this setting enabled in most situations. One exception might be the creation of a pattern for increased base torque, where the change in current polarity would result in a different haptic feedback instead.

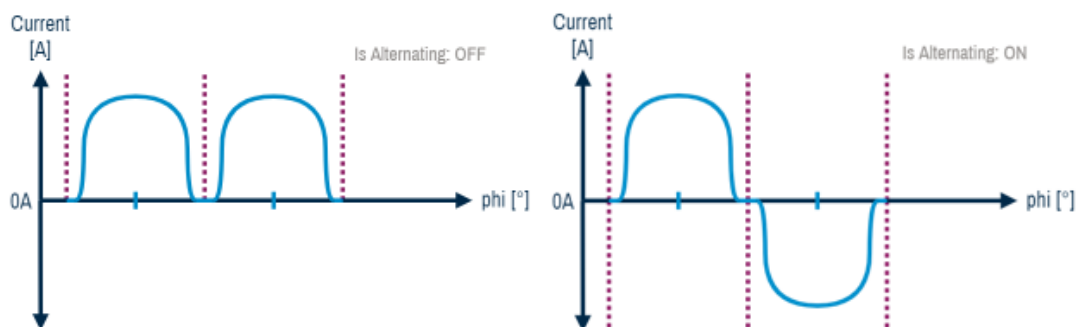


Figure 4: The difference between disabled vs. enabled alternating current

2.3 Parameter (Graph)

The parameters in this group are targeting the currently active graph. With the one exception that, if *Sync Forward and Backward Parameter* is enabled, the forward direction graph and parameters get mirrored for the backwards direction and the other way around respectively.

2.3.1 Forward / Backward

The *Angle* defines the number of repetitions for the defined haptic pattern. An angle of 10° , for example, leads to 36 repetitions over the 360° range of the HAPTICORE. This is also shown in figure 2.

The *Tick Index Count Angle* defines at which position within the haptic pattern the tick should be counted. This count is accessible through the *Tick Index* report. For more information regarding the *Tick Index* please check the *HAPTICORE Haptic Function Documentation*.

The *Maximum Current* parameter represents the current range from zero (on the Y axis of the grid) to the (maximum) current on the topmost point.

The *Maximum Error* represents a metric for the discretisation process before the data is transferred to the device. Discrete points are sampled along the curve which are transferred into a lookup table. For each angle value the current value is determined by a lookup in this table. If an angle value isn't found, linear interpolation is utilized to define the according value. If the defined maximum error is surpassed another point will be added. Therefore, a smaller error leads to more points and an increased memory consumption. Usually, the default setting leads to good haptics results at an acceptable memory footprint.

2.3.2 Velocity Compensation

Velocity Compensation is the process of multiplying the applied current (at the current point of the haptic pattern curve) by a certain factor based on the rotational velocity of the encoder.

The *Minimum Compensation Factor* setting defines the baseline (Y axis value 0) of the velocity-based compensation while the *Maximum Compensation Factor* defines the factor for the highest graph point(s). These parameters represent the range of the Y axis.

The *Maximum Velocity* represents the velocity compensation cap when the maximum compensation is reached. This parameter is used symmetrically around the Y axis, which means, that a maximum velocity of 720°/s defines a range from -720°/s to 720°/s. In this point, the graph is different from the forward / backward graph.

The *Maximum Error* parameter works the same as for the forward / backward graph settings (see section 2.3.1).

2.4 Lookup Table Info

This group lists the number of discrete points necessary for the current curve with the defined *Maximum Error* setting. This metric should help to determine how much memory is necessary to store / define this haptic pattern within the firmware and can be adjusted accordingly due to the maximum error settings.

The samples points for the current settings are visualized on the graph as small blue dots.

3 Export / Import

The *Export Graphs* and *Import Graphs* functionality exports / imports the current graphs and settings to (and from) a *.hgf file.

The *Export Code* option produces two *.c files, both containing source code to use these settings directly in firmware. In the "assignment" variant the data is assigned directly to the entries of the lookup table, while the "array" variant utilizes an initializer list. Both variants function independently, so the more suitable variant should be used for integration.

4 Graph Editing

The *Forward* and *Backward* graph editing follows some rules to ensure that the shape is valid in the Haptics Generator context (Exactly one Y value for each X value etc.). These rules are enforced automatically by the application through automatic manipulation of the constrained points. The graph consists of five control points as shown in figure 5. These points are used to form a Bézier curve. Points 1 and 5 are called *start* and *end* point respectively. Their position is fixed on the X axis. Points 2 and 4 are called left and right *control* point and point 3 is named *center* point for simplicity.

The following rules are applied:

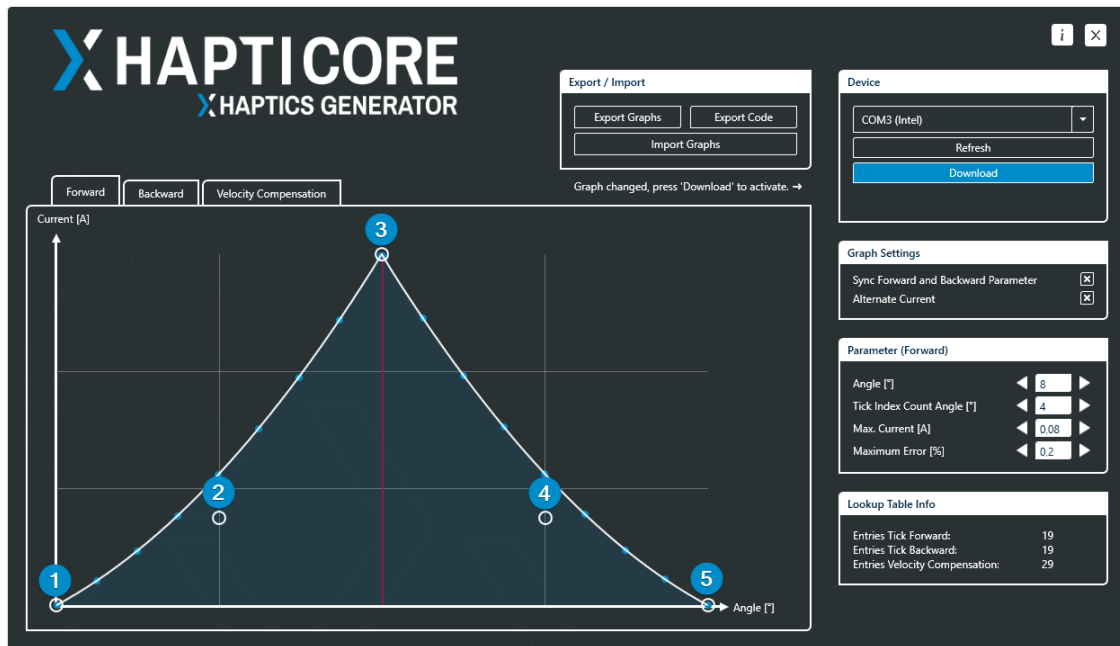


Figure 5: These five control points are used to define the graph.

- The points (as their respective numbers) have to be ordered ascendingly along the horizontal axis (start point to the left of left control point etc.).
- Start and end point are not movable on the X axis.
- The vertical positions of the elements are constrained in a way that center point is always higher than the control points left and right. These two have to be positioned (in terms of Y value) above the start and end point.

Note that the third rule **doesn't** apply for the *Velocity Compensation* graph, since the need for velocity compensation increases with the rotation velocity.

5 Other Applications

This section describes the interaction of the *Haptics Generator* with other HAPTICORE applications.

5.1 HAPTICORE Customer Test App

Within the *HAPTICORE Customer Test App*, the Haptics Generator is integrated as an independent haptic function, which can be combined with other haptic feedback functions like the *Barrier* or *Tick* function (see figure 6).

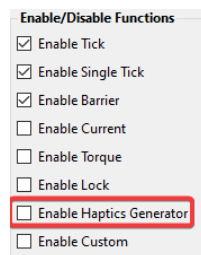


Figure 6: The Haptics Generator mode is listed as its own haptic mode in the Customer Test App.

Overall, the Haptics Generator is integrated as it's own haptic mode, which allows combining it with the "classic" haptic modes. However, to apply the Haptics Generator settings, the generated data needs to be transferred first, since, by default, it disables all other haptic functions. In a second step, other haptic functions can be added.

Note that the Haptics Generator needs to use the virtual serial port to transfer data, so the *HAPTICORE Customer Test App* has to be disconnected when the Haptics Generator *Download* button is pressed.

Please note, that this tool's scope is to define haptic patterns through a curve. General settings, like *Idle Detection* or encoder filter settings etc. require the use of the HAPTICORE Customer Test App.

6 Troubleshooting

- If the correct serial port isn't listed in the Device section, please ensure, that the according driver is installed. To install the appropriate driver, please follow the instructions of the *HAPTICORE Quick Start Guide*.

You still didn't find the answer to your question?

If you need additional assistance or have recommendations for changes, please don't hesitate to get in touch with us:

support@xeeltech.com
+43 5552 93081-0
www.xeeltech.com

Or visit www.xeeltech.com/hapticore-support where you can find helpful FAQs and video tutorials.

Here you will also find our other HAPTICORE software solutions, allowing you to create even more personalized haptic feedback modes.


by EELTECH