Solid-State Button Design – Hands-on Guide

1 Overview

The Solid-State Haptic Button is a new type of haptic button that provides a unique haptic feedback experience. The button is designed to provide customizable haptic feedback with a customizable trigger force. The button is only composed of a piezoelectric actuator. The actuator is used to provide both haptic feedback and measure applied force. Here, we will explain how to use the piezo as a sensor and a haptic actuator.

The goal here is to cover how to design a solid-state button prototype using a Boréas development kit, Haptic Studio and a custom mechanical assembly. We strongly recommend designing and debugging your prototype in this environment before integrating any custom software into your platform. The environment proposed there allows you to validate the mechanical integration and the piezo selection. This document only covers the basics, but it's an important initial step that shouldn't be overlooked. Many checks must be performed before integrating your product. This document doesn't intend to cover the calibration topics and the final product implementation.

The suggested development process is the following:

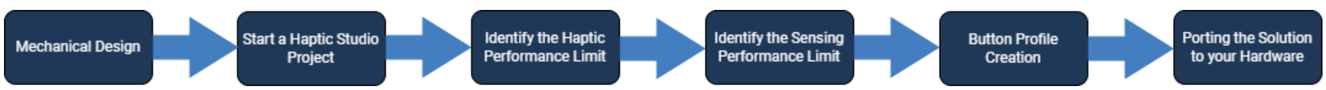


Figure 1- Suggested Development Process

The optimal development process starts with the mechanical design i.e., designing the piezo integration, selecting the piezo and sizing the hardware for driving and sensing. The next step is the application prototyping using Haptic Studio. Haptic Studio, developed by Boréas Technologies, is a tool for configuring the development kit to operate the piezo as a haptic button. It’s also a great tool for characterizing the haptic and sensing performance of your button concept; assisting designers in identifying and refining mechanical design parameters before proceeding with integration. In these phases, Haptic Studio helps to evaluate the application parameters without struggling with the software/firmware integration. Following this, the designer will be able to define its button profile within the firmware application and do the final evaluation before porting to its own platform.

2 Table of Contents

3 Piezo within Mechanical Design Considerations

First, we need to discuss the basic concepts of the piezoelectric properties. The piezo actuator is a device that converts mechanical energy into electrical energy or electrical energy into mechanical energy.

When a mechanical force is applied to the piezo actuator, it generates electric charges which can be used as sensor signals. This signal can be used to detect and derive a button press force at the Boreas IC level.

On the other hand, the piezo actuator can convert electrical energy into mechanical energy. When electrical charges are sent to the piezo by the Boreas IC, or when a difference in potential is set at the piezo boundary, this induces mechanical strains in the ceramic. The piezoelectric ceramic in the actuator will expand or contract in the direction intended by the electrode's placement and the polarization axis. This expansion and contraction will transmit forces and vibration sensation to the user.

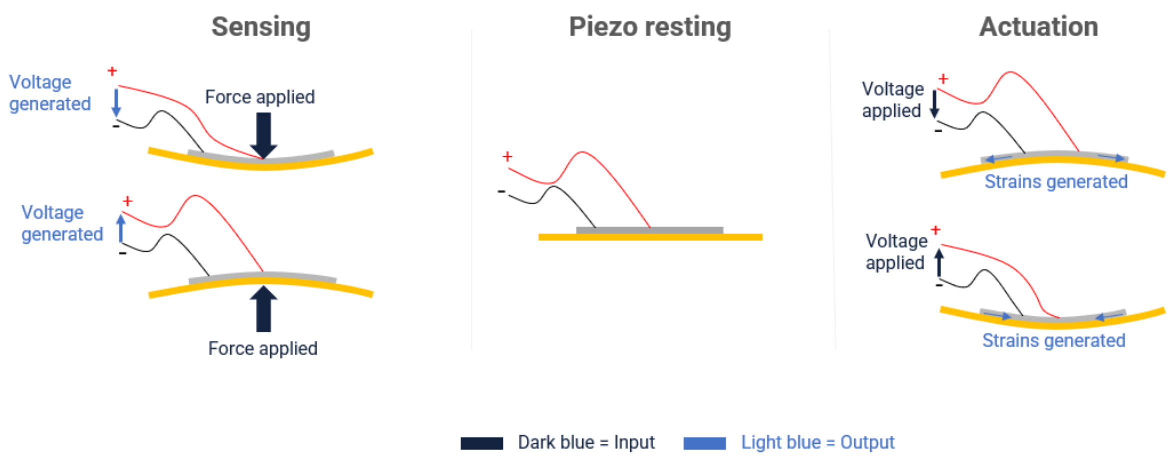


Figure 2- Piezo Position According to Different Scenarios

Sensing and actuation capabilities of the solution will be tied to the piezoelectric ceramics used and their properties (as well as the geometry and boundary conditions). For instance, applying pressure on the piezoelectric sensor during a button press will generate a positive or negative voltage. We define this property as the **sensing polarity**.

A similar property exists for the haptic feedback. The **haptic polarity** defines if a positive voltage will push or remove pressure on the finger pressing on the button.

4 Considerations for Haptic & Sensing Performances

The piezo actuators are integrated in a mechanical structure generally adapted to a targeted application, here a solid-state haptic button. When well designed, the proper mechanical integration ensures the piezo can be used at its optimal performances.

On the other hand, if some key aspects are not considered, mechanical integration can decrease the piezo performance, and at worst case suppress the haptic effect totally.

Here are the key aspects to keep in mind when designing the mechanical integration:

- Piezo mounting:
  Different types of piezo actuators are available (bender, disc, amplified/bow piezo…). The mounting points must be adapted to each actuator to allow the maximum displacement of the actuator or the targeted rigidity.

For example, if a bender piezo is used, a three-point structure is needed. The distance between the points will impact the rigidity of the actuator.

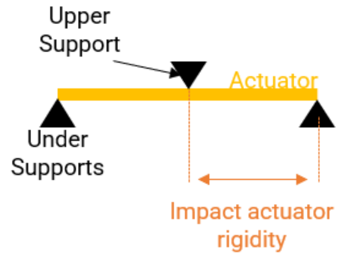


Figure 3- Piezo Three Point Support

- Piezo and button must be able to move properly:
  Haptic displacement can be anywhere between 10-100um. If the structure does not allow enough displacement to be generated by the piezo and transmitted to the button in contact with the finger, little to no haptic will be felt.
  This generally eliminate any seamless or rigid structure, as no piezo at this size using this electrical hardware is strong enough to generate enough deformation / displacement.
  A moving part is thus needed, and it must also move relatively freely without excessive friction (specific to each design).
  The piezo must be supported on one side by a surface that is as stiff and as heavy as possible. On the opposite side, the surface needs to be able to move freely along its axis. This setup directs the actuator's energy in a single direction (usually in the direction of input force, such as through a finger) and minimizes energy loss in the base of the system, where the user wouldn't feel the haptic feedback.
- The actuator must always be in contact with the button:
  To guarantee this condition, preloading the actuator is needed in most cases.
  The preload force must be adjusted to not overload the actuator. If the preload is too high, the actuator will need more energy to be able to move the button and the haptic strength will be reduced significantly.
  The preload must be lower than the blocking force.

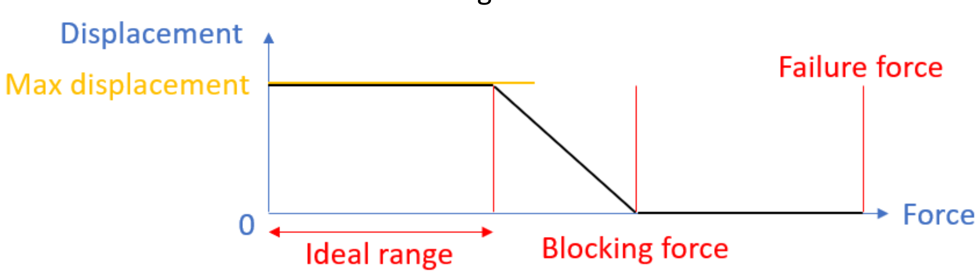


Figure 4 - Optimal Displacement in a Piezo Integration

Piezo haptic buttons are no magic. They are like conventional switches, but with less displacement but moving faster. Over-constrained or under-constrained assemblies cannot work properly.

5 Hardware Component Sizing

It is critical to ensure your piezo can be driven by the Boreas HW devkit before looking to the sensing solution i.e., the button profile.

It is important to use the adequate hardware components with the BOS1921 for your piezo and the haptic profile targeted. The hardware component mounted on the daughter card must be sized according to the piezo capacitance, driving voltage and frequency range. Otherwise, you may experience distortion and damping of the haptic.

The development kit haptic daughter card's default configuration (BOS1921) allows for operating the piezo at 300 Hz -95V to 95V sinusoidal waveform with a 100 nanofarad load. Consider frequency and dynamic amplitude as a power factor: if the amplitude decreases, the frequency can be increased. The same logic applies for the piezo load size.

If you intend to use a piezo outside of these specifications, please use the online BOM generator.

6 Start a Haptic Studio Project

Start a project with Haptic Studio by filling in the information related to your piezo. See the help embedded within Haptic Studio for the latest and greatest procedure. Help section can be found at the top of each page with the "**?**" button.

Creating a project is important so you can save your waveforms and button configurations on your PC. Choose an easy to locate folder so you can retrieve the project on your next work session. Figure 5 shows the help button and "New Project" button on Haptic Studio Home page.

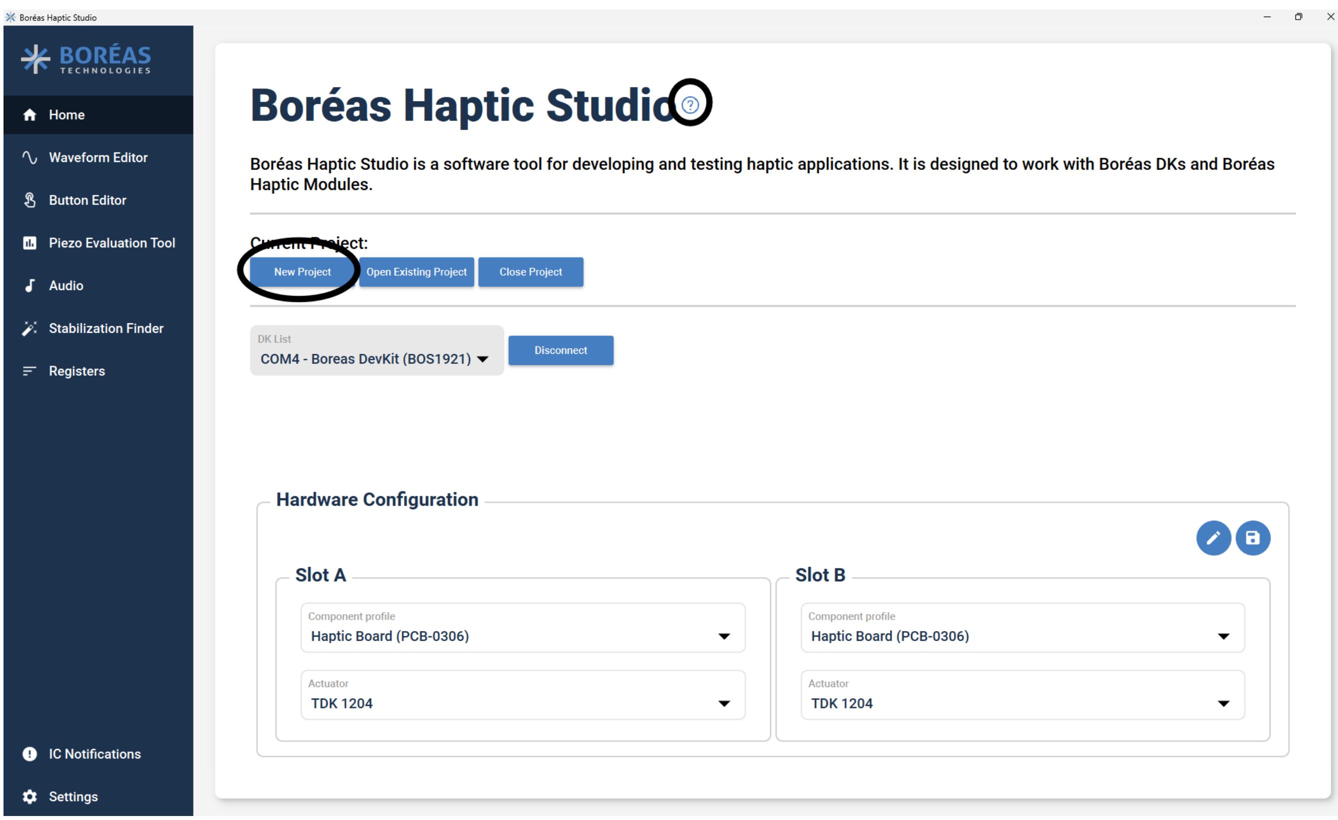


Figure 5: Haptic Studio Home Page

7 Haptic Performance Evaluation and Tuning

Before going to the button profile creation, we suggest validating the haptic actuation manually on your piezo button. The idea is to validate if the haptic feedback can be felt by the finger and is within BOS1921 capabilities.

- Create a new project in Haptic Studio.
- Go to the **Waveform Editor** page.
- Create a waveform compatible with the operating voltage of the piezo. See the piezo datasheet. Ideally, you should target the maximum voltage range available for your piezo.
- Experiment subjectively the haptic feedback with your finger by manually firing the haptic. Select the haptic feedback and click on the play button.

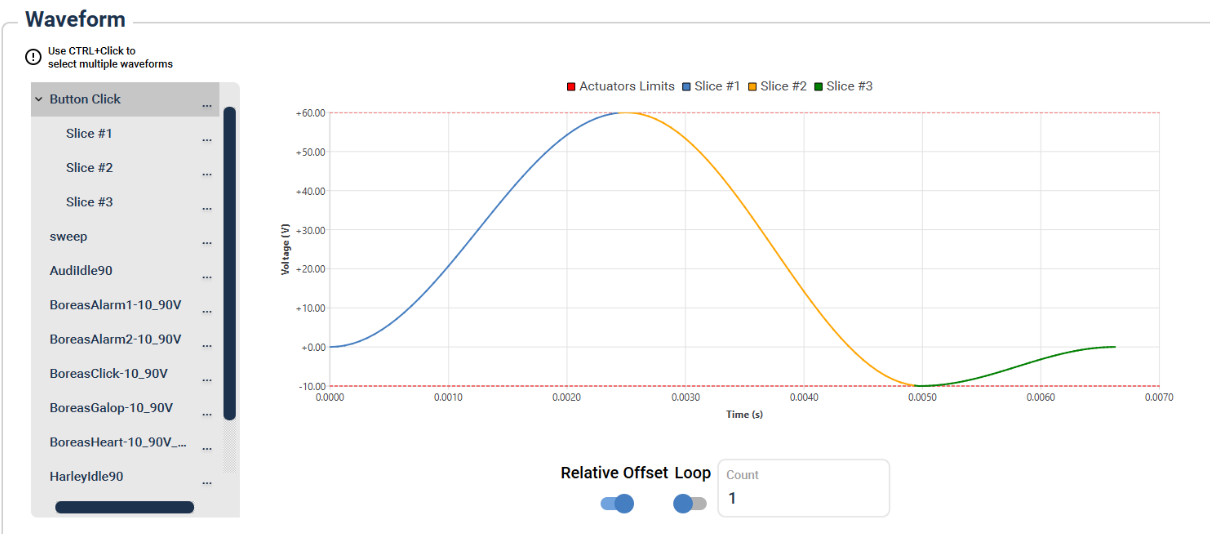


Figure 6 - Haptic Studio Waveform Editor

7.1 Haptic Waveform Design

Designing the haptic effect should start with understanding the information that needs to be transferred to the user at the UX level. The first step is to define the intention behind the haptic effect.

The objective is to replace the tactile feedback given by a mechanical switch by a unique customizable haptic effect.

Parameter	Effect on haptic	Recommendation
Amplitude	Strength/Depth	Strength should increase proportionally to the amplitude. If not, it means the actuator, or its preload conditions need to be adjusted (saturation effect).
Frequency	Sharpness/Depth	Finger optimal perception frequency is around 200Hz. Going above 350Hz is not recommended.
Number of cycles	Strength/Buzz	Use 1 or 2 cycles maximum. Above that, it would be perceived as a buzz.
Shape	Uniqueness/Noise/ efficiency	Sinus waves are the easiest and most used. It is possible to combine different functions such as Gaussian, sinus or polynomial function with an external waveform design tool.

The shape of the waveform can be separated in three parts, as shown in Figure 7.

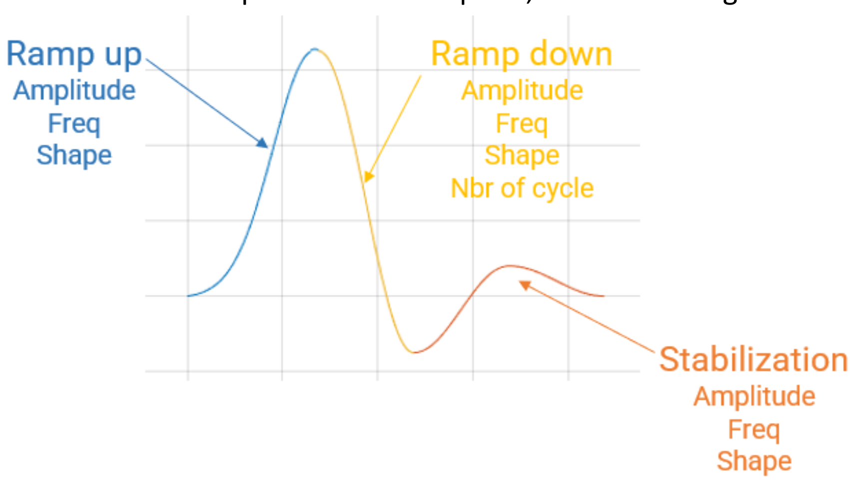


Figure 7- Haptic Feedback Dissection

The ramp up will define the maximum absolute voltage of the waveform.

The ramp down will define the peak-peak voltage of the waveform, it’s the most meaningful part of the waveform haptic effect. You can customize the amplitude, frequency and number of cycles to provide the desired haptic effect.

The stabilization is used to minimize piezo creep, it will be discussed in further detail later in this document. For now, make sure the waveform ends at 0V.

From here, you validate the following items:

- Absence of MAX_POWER warning events communicated by the development kit.
- Absence of signal distortion using an oscilloscope at the piezo terminal.

These issues above can occur even if you feel the haptic feedback with your finger. The best practice is to resolve any issues before moving forward. Otherwise, you may encounter problems later in the design process.

If you want to further optimize your waveform to reduce audible noise and electrical noise on the power rail, you can read Section 10 - IC Performance Optimization (optional).

7.2 Avoid MAX_POWER Event

Haptic Studio reports any warnings and errors in real time. See the IC notifications on the left bottom panel.

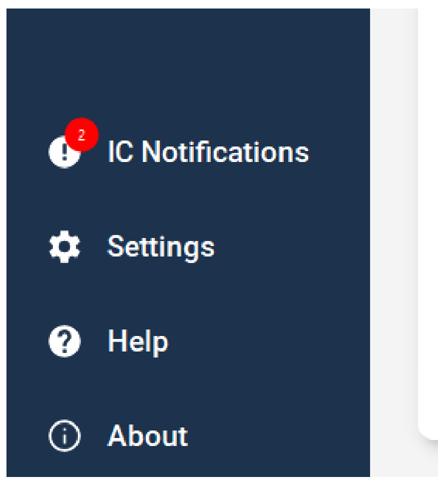


Figure 8- Haptic Studio Event Notification Icon

If any errors/warnings occur, a red circle will appear on the IC notifications buttons. You must click on it to consult the list of errors/warning reported.

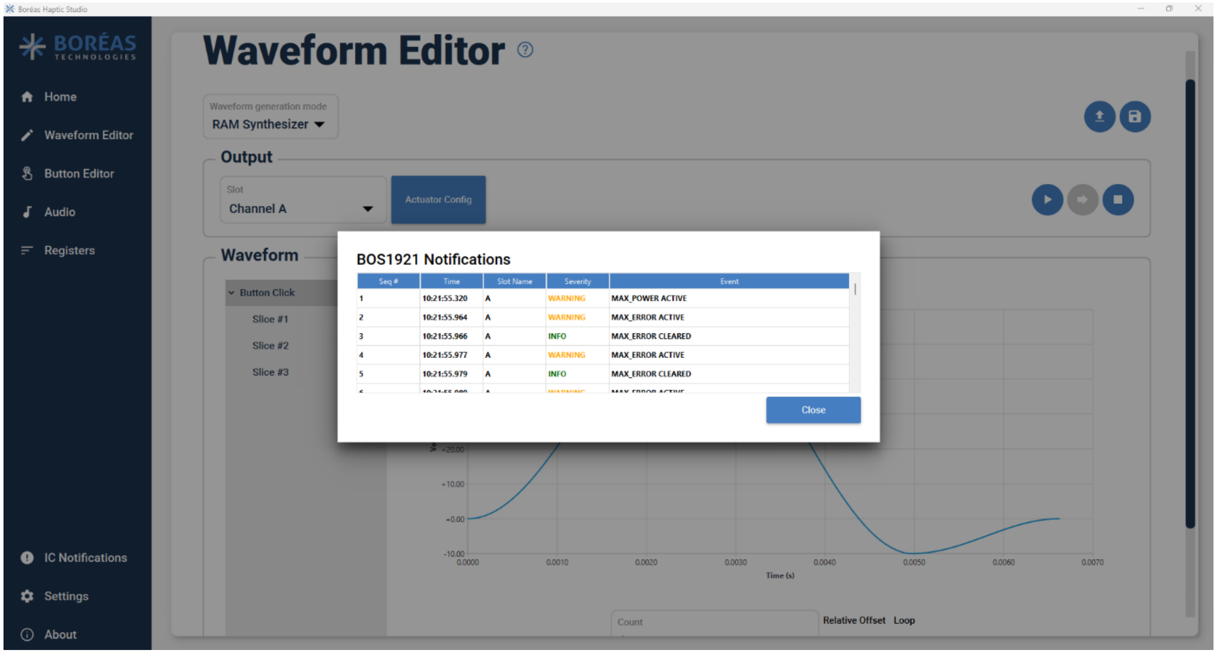


Figure 9 - Haptic Studio Event Notification Window

Encountering MAX_POWER warnings indicates that the current driver and component selection are operating at their capacity limits. From here, you should validate the bandwidth of your complete setup with the Piezo Evaluation Tool. It will help you understand the frequencies and voltages that your setup can drive without warnings.

7.3 Piezo Evaluation Tool

The Piezo Evaluation Tool iterates through voltages and frequencies to display the bandwidth of the current hardware setup.

Figure 10: Piezo Evaluation Tool Result Graph

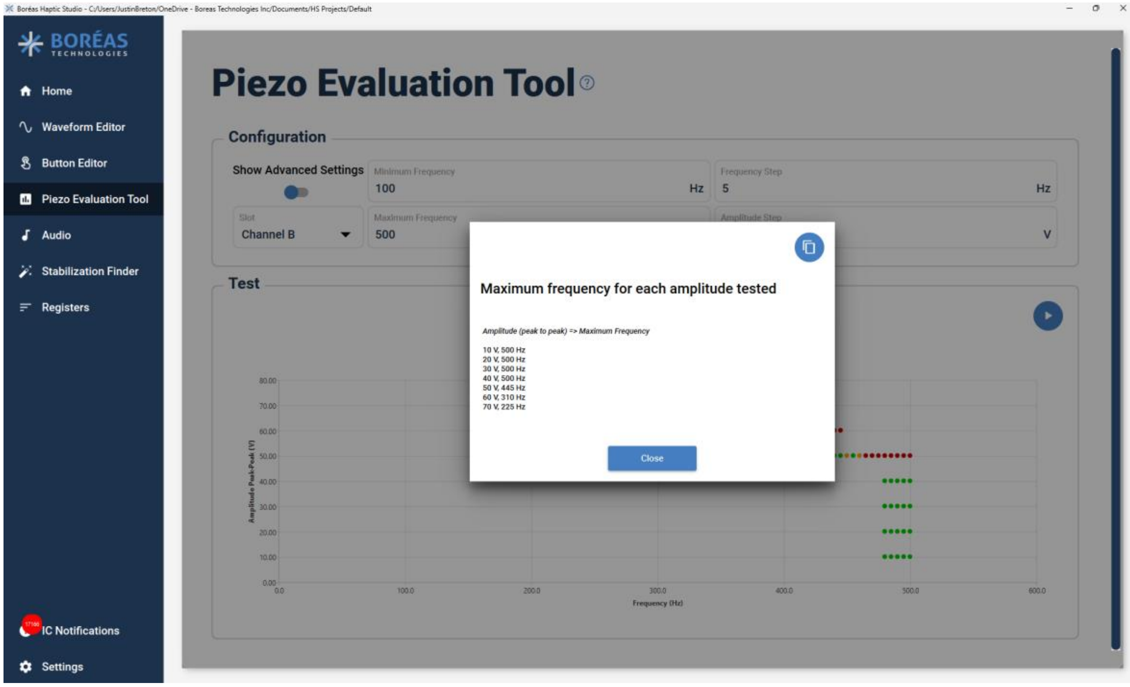


Figure 11: Piezo Evaluation Tool Result Text

With the parameters and hardware tested above, the maximum frequency for the maximum piezo voltage is 225 Hz. Higher frequency can be set with a lower voltage as shown in Figure 11. Design a waveform within your result parameters to avoid further issues.

If you are satisfied with the performance and do not encounter MAX_POWER warnings with your designed waveform, you can jump to section 0.

If you are not satisfied with the performance and want a higher frequency / voltage than the Piezo Evaluation Tool specifies, there are two options: enabling buck-boost converter to use continuous conduction mode (CCM) or changing hardware component selection.

7.4 Activate CCM mode

Enabling the continuous conduction mode (CCM) can increase the maximum power available by ~20%. However, this mode comes with requirements for the hardware component (see datasheet) and additional noise generated on the power rail. Once these requirements are met, you can turn the mode on.

In Haptic Studio, go to the settings menu and check the "Display advanced pages" toggle button. Close the settings menu and go to the newly added "Registers" menu. Change the field CCM in the PARCAP register from 0 to 1. Right click on the field and set it as a default value.

Retry the haptic feedback on your piezo. If MAX_POWER warnings are still being reported during haptic playback, consideration should be given to other options: optimizing hardware components and redesign waveforms.

7.5 Validate the Piezo Voltage Signal

There must be no significant distortion on the voltage signal sent to the piezo actuator during prototyping. Signal distortion can occur even without MAX_POWER warnings. Designers must validate this to ensure the solution is repeatable at production scale with manufacturing variability and hardware tolerance. Distortion is characterized by comparing ***desired*** waveform against ***measured*** waveforms.

To validate it, you can probe the output of the piezo with an oscilloscope. In the example below, we are using a TDK 1204 on a BOS1921-KIT. Since the piezo is bipolar, we have a probe on “OUT+” and a probe on “OUT-”. Do not connect the GND terminal of the probe to “OUT-” even if you are using a unipolar piezo since this node is biased at VDD.

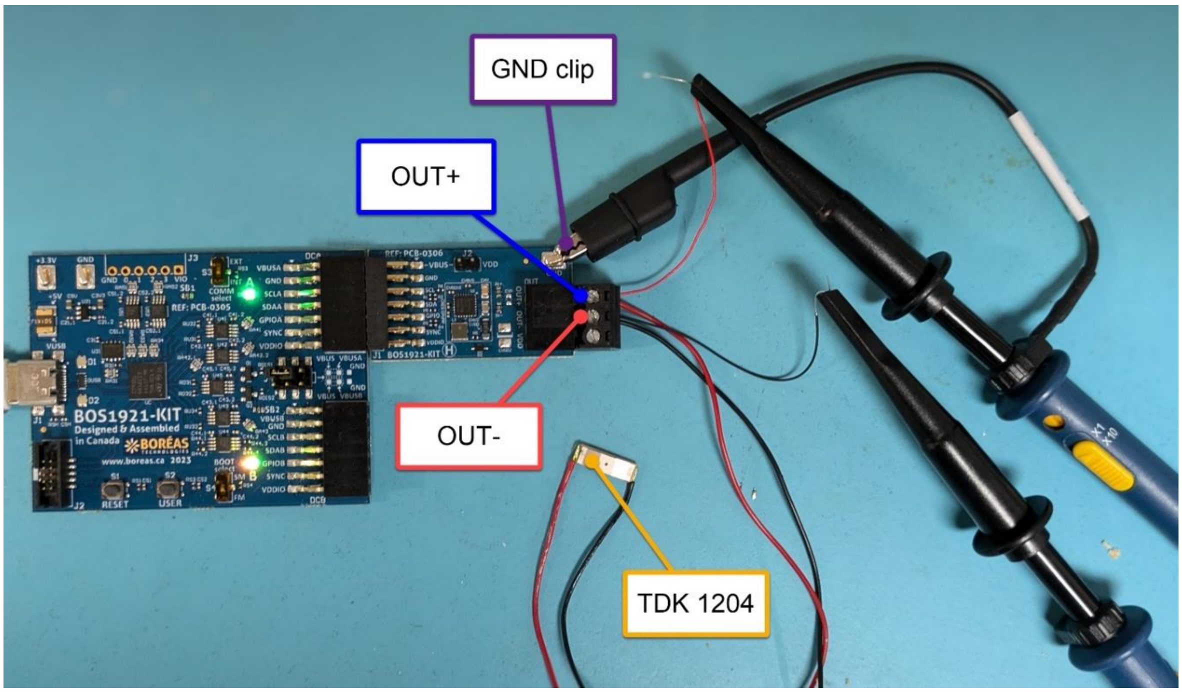


Figure 12- Measurement Setup

Next, the oscilloscope is set to see the waveform sent from Haptic Studio. In this example, we are sending two different waveforms to see what distortion is. The first wave drives the piezo at 60V/200Hz, which is in the configuration limits, whereas the second wave drives at 60V/**400Hz**, which is above the recommended limits. On the first one, we can notice little to no distortion, whereas on the second, we can clearly see distortion.

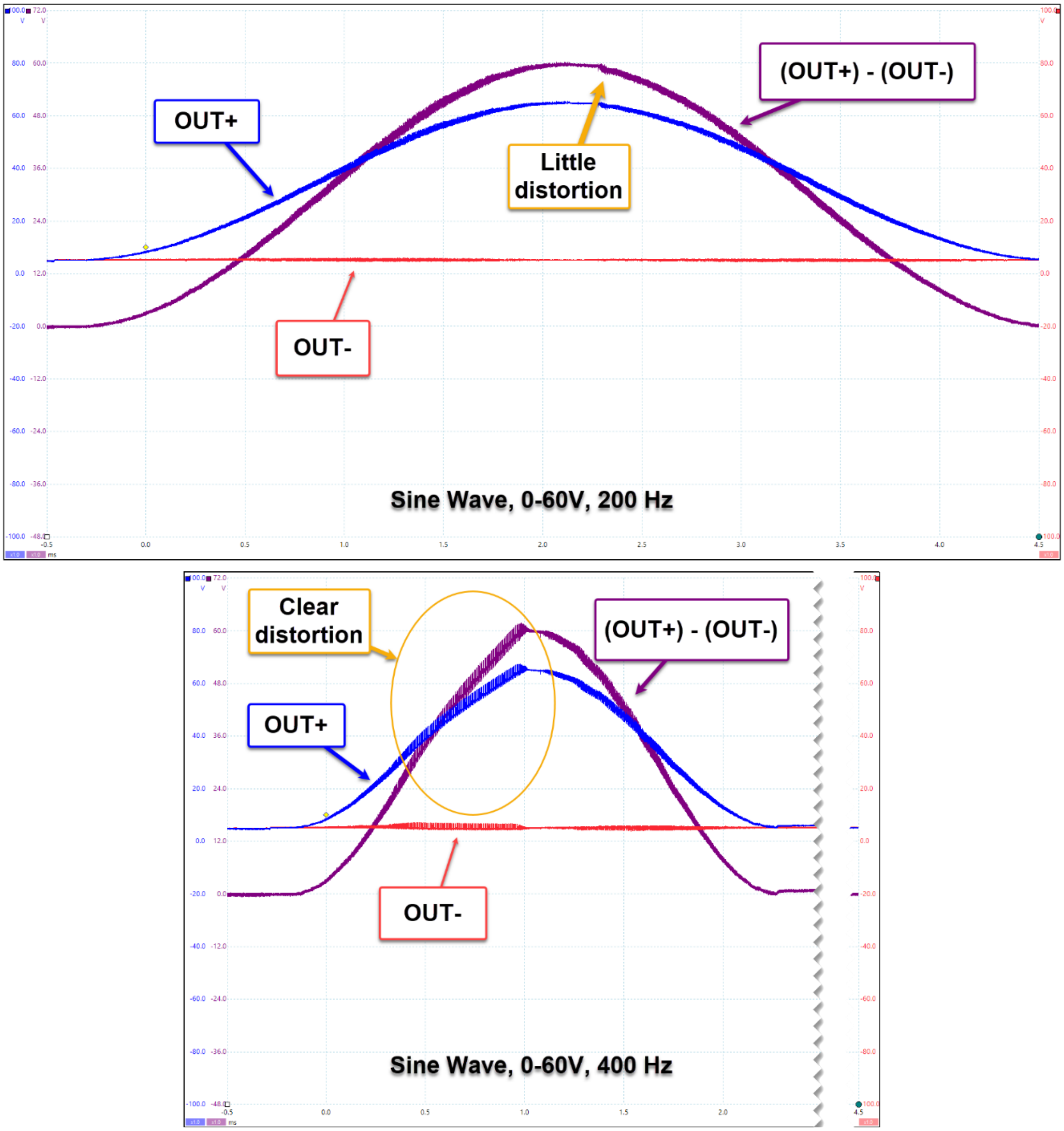


Figure 13- Two Distortion Cases

Just like encountering MAX_POWER warnings, if signal distortion occurs, you can:

- Enable buck-boost converter to use continuous conduction mode (CCM)
- Optimize driver PI controller and register value (Section 10)
- Adjust the component selection

7.6 Waveform Stabilization

Once the waveform is designed and no warnings appear, there is an important step left to implement it in a button application. The waveform may provide good haptic feedback performance, but it needs to be stabilized for the sensing capabilities of the IC to work properly. A stabilized waveform means that the piezo voltage is stable at 0V after the end of the haptic feedback. Naturally, a piezoelectric device will introduce a creep if not stabilized, this phenomenon will harm the sensing performance.

To simplify waveform stabilization process, Haptic Studio features a Stabilization Finder tool that will find the right stabilization that needs to be added at the end of your designed waveform.

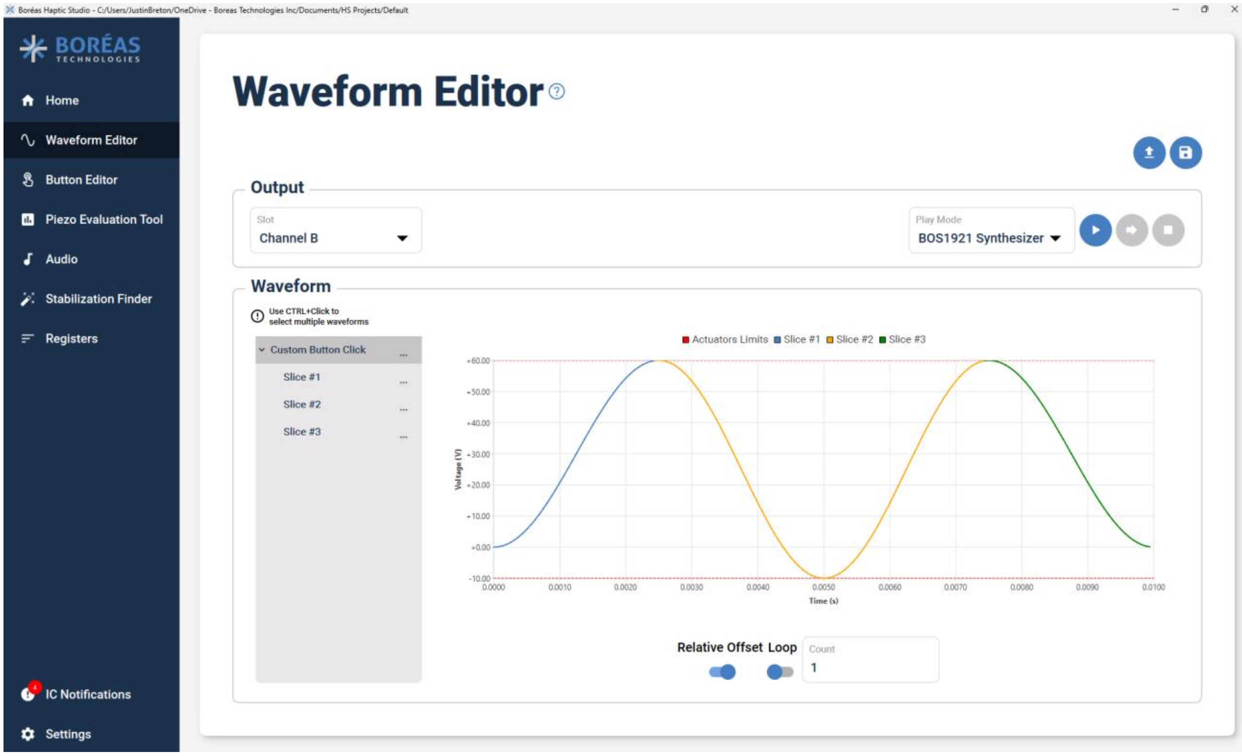


Figure 14: Initial Un-stabilized Waveform

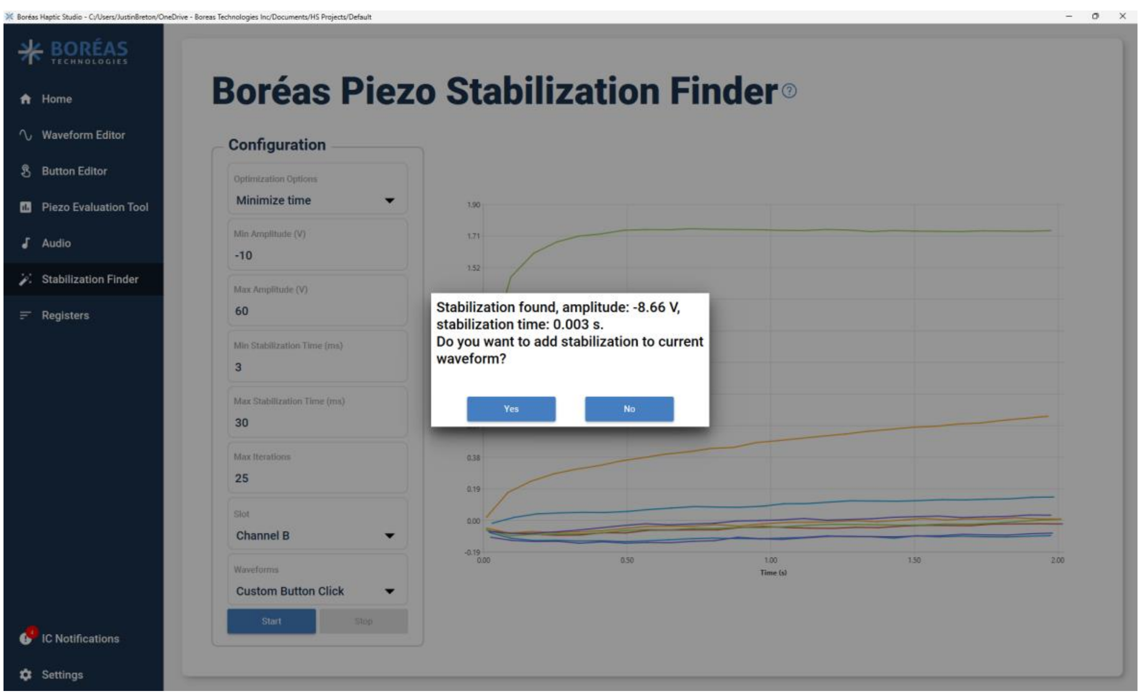


Figure 15: Stabilization Finder Tool

Once the tool has found a suitable stabilization for your waveform, it will add it to the waveform list in the waveform editor page.

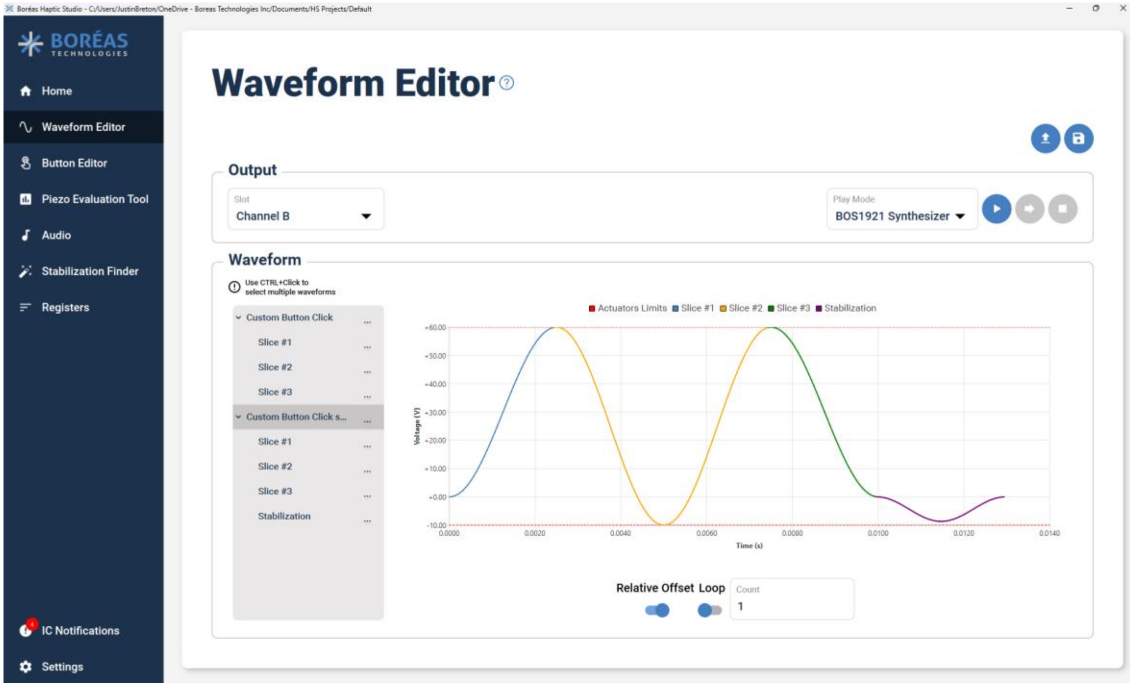


Figure 16: Stabilized Waveform

The haptic feedback modifications from the original waveform should be unperceivable by the user.

8 Button Profile

The following sections will guide you through the Button Editor page and how to create a button profile for your application.

The button profile groups the sensing and haptic concepts altogether. Essentially, the button profiles define the sensing configuration, and the haptic feedback assigned for each button state transition. These transitions are illustrated below. There are two types of button profile: single-level and multi-level. As you can see below, the single-level button refers to a traditional mechanical button with two states: released (level #0) and pressed (level #1).

The multi-level button is composed of three states: released (level #0), mid-press (level #1) and fully pressed (level #2). This type of state machine is customizable by deactivating some transitions. Make sure the button can return to level #0.

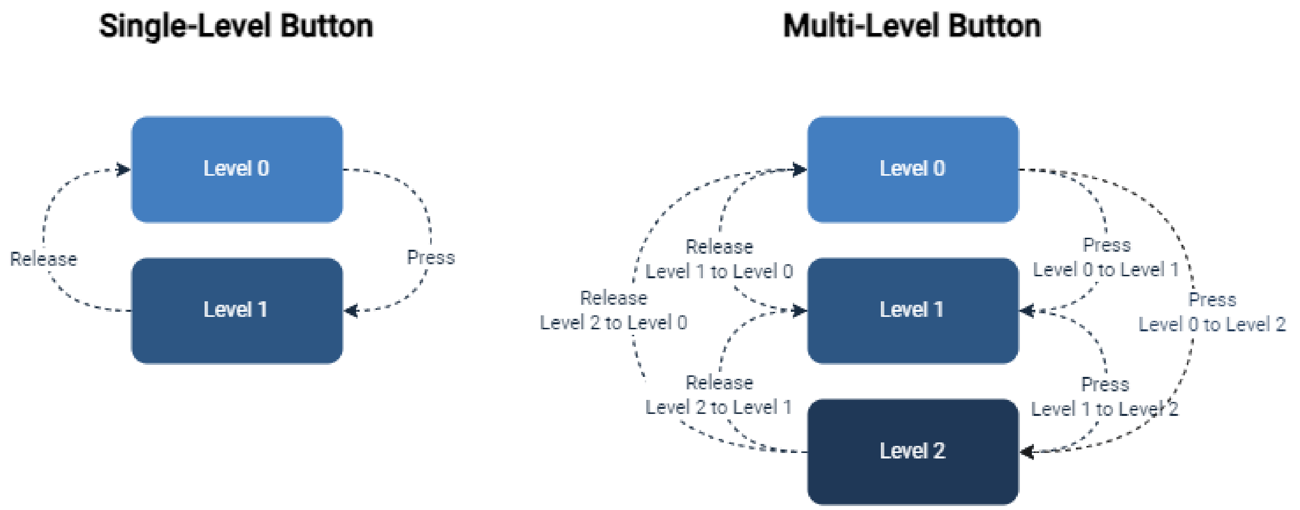


Figure 17- State Machine Implementation per Button Type

8.1 Button Transition Parameter Definitions

The following sequence diagram describes how the button transition has been implemented and how the button profile is linked to this logic. Regardless of the button type, the same logic applies when transitioning between levels: the state machine expects a threshold to be reached. In some use case, an additional debounce time can be added. Once the two conditions are reached, we fire the haptic.

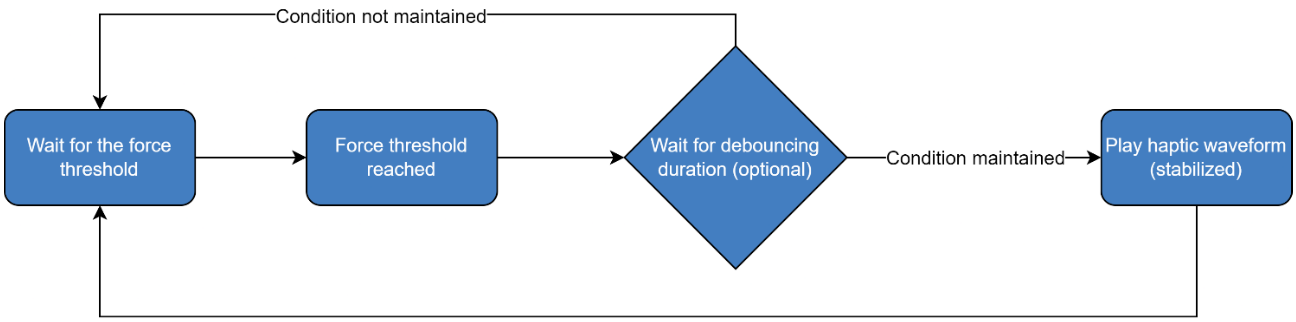


Figure 18- Level Transition Sequence Diagram

The button profile is defined by the parameters below. Haptic Studio uses them in the button editor tab.

- **Threshold** is the sensing force required to fire the haptic.
- **Debouncing** is an additional criterion to hold the threshold condition for an elapsed time before firing the haptic feedback.
- **Waveform** is played when the threshold and debounce time conditions are met.
  Make sure the waveform is correctly stabilized, using the Stabilization Finder tool in Haptic Studio

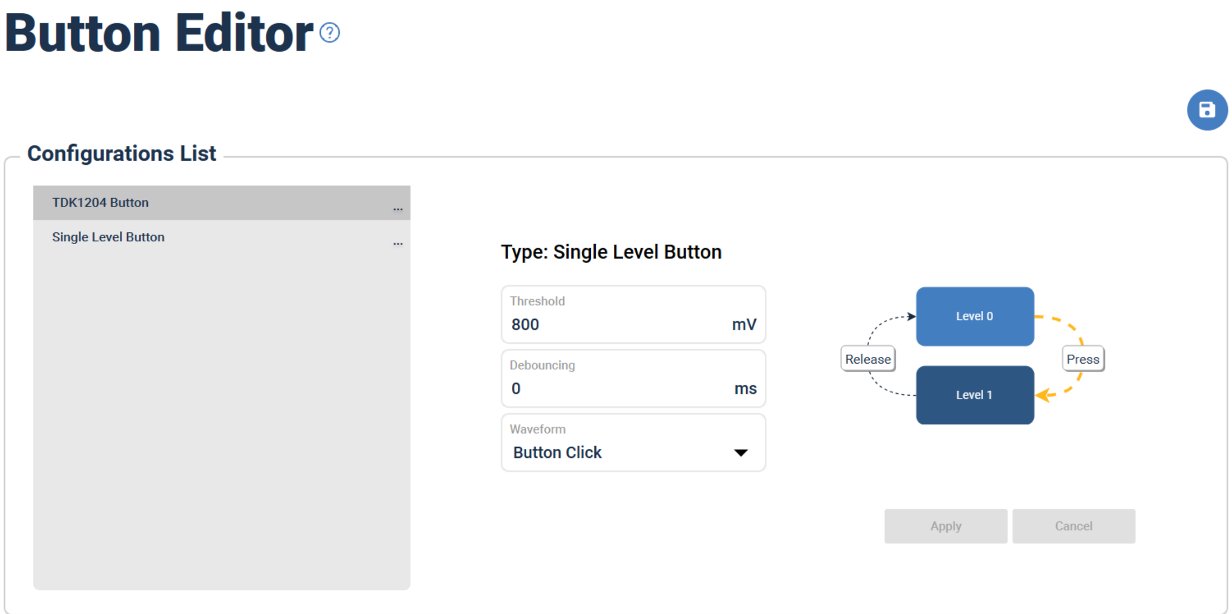


Figure 19: Button Editor

8.2 Button Profiles Guide

Now that the fundamental concepts of button configuration have been covered, a working button profile may be crafted using the waveform created in a previous section.

First, the operating sensing range of the piezo sensor must be determined if it is not already specified in its datasheet. This refers to the relation between the applied force and the generated voltage. To do this, you can use the Real-Time Button Sensor page, at the bottom of the button editor page. This allows you to see, in real time, the voltage corresponding to the applied force.

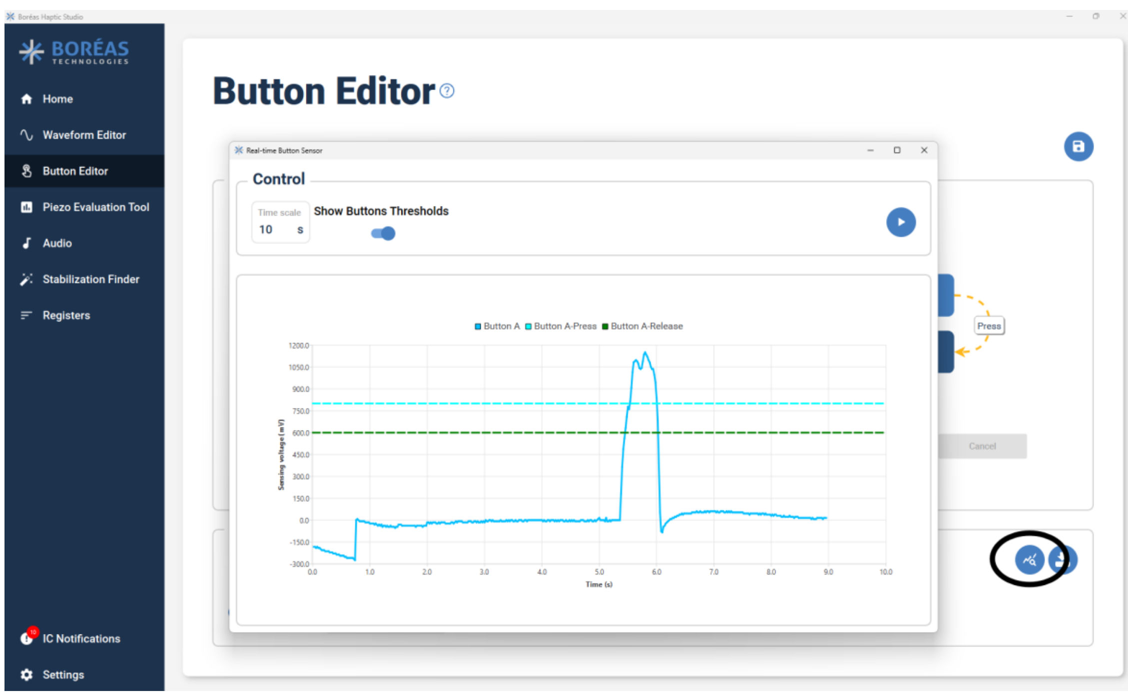


Figure 20: Real-Time Button Sensing

8.2.1 Single Level Button Profile

To create a custom button profile, follow the steps below:

1. In the button editor, right-click on the profile list and click on **Add**.
2. Select the button configuration type. Let's start with "Single Level", we will cover "Multiple Level" later. This prompt will create a generic button profile that you need to adjust for your design.
3. Select the press transition by clicking on the state machine illustration. Set the threshold around 10% of the total displacement from your measurement. Typically, no debounce period is required for a single-level button, so keep it to 0.
4. Select the release transition and set the threshold 0.75x the press threshold. The goal is to trigger the release close to the press force threshold without exceeding it, to avoid detection issues.
5. Apply the button profile to the development kit and experiment with it. Open the Real-Time Button Sensor to display the sensing force relative to the threshold set.

From here, you can begin to adjust the button according to your specification i.e., the stiffness and the button feedback strength.

When selecting the sensing force threshold, the user needs to know it can influence the haptic feedback strength and its perception:

- If the force threshold is too high, it could block the actuator displacement and thus reduce the strength of the haptic effect.
- If the force threshold is too low, the contact between the finger and the button could be too small and the haptic effect could be perceived weaker than it is.

The following procedure is a highly empirical approach. However, this allows the designer to validate some assumptions on the mechanical and electrical domains. If you are satisfied, the next step is to integrate it into your platform. A reference code firmware is available for porting the button application on your own MCU.

8.2.2 Multi-Level Button Profile

The multi-level button resembles the single level button but with additional transitions. See Figure 17 for a comparison between Single Level and Multilevel transitions.

All transitions can be used together if set up correctly. Key points for designing a multi-level button are:

- Transition 1→2 threshold must always be greater than transition 0→1.
- If using transition 0→2 simultaneously with 0→1 and 1→2, you need to put a debouncing value on the 0→1 transition. The debouncing value will impact how sensitive the 0→2 transition will be.
    - The same rule applies for the release, but the debouncing will be on the 2→1 transition.
- Make sure you set up each transition with a stabilized waveform. You need to select each transition one by one.
- Use the Real-Time Button Sensor to visualize the current force and thresholds.

9 Porting the Solution to your MCU/Firmware

If you consider your mechanical design operational, you can start to port it on your platform. Boréas offers a software development kit (SDK) for implementing your software button. The SDK runs out of the box on our development kit hardware. A guide is also included to compile for a different target MCU.

In the process, you must port the button profile defined in the Haptic Studio project. The documentation within the SDK explains.

10 IC Performance Optimization

Optimizing the registers of the IC for your application is important to achieve optimal performance of your system. While default parameters will lead to a functional solution in many scenarios, suboptimal performance may lead to higher electrical noise, higher acoustic noise, higher-power consumption or the system stopping working properly in some corner use cases.

The optimization of the parameters is done in three steps.

1. For all parameters in the datasheet with a formula that includes a component value, make sure to compute the recommended value of the parameter for your application based on your BOM. For the BOS1921 these parameters are:

Register Address	Register Name	Parameter name
0x01	ION_BL	I_ON_SCALE
0x02	DEADTIME	DHS
0x03	KP	KP*
0x06	PARCAP	PARCAP
0x07	SUP_RISE	VDD
0x07	SUP_RISE	TI_RISE

   *The default value of KP is generally a good starting point and assumes the maximum capacitive load of the IC (100 nF for the BOS1921). This value can be scaled proportionally to the load in your system. For instance, if your system has a 10 nF load, you will start the initial KP at $KP_{default}$ * 10nf/100nf = 1/10 $KP_{default}$

2. The second phase of optimization requires that you play sinusoidal continuous waveforms and observe the shape of the waveform with an oscilloscope (refer to section 0 on how to take measurements). It is recommended to perform this step with two types of waveforms. Each oscilloscope capture must have a scale that enables you to see the full amplitude of the waveform and approximately at least one complete cycle and preferably between 3 and 5 cycles.
   a. Maximum output power waveforms, with the lowest input voltage. In this case, you will play a waveform that has the maximum combination of frequency and output voltage while setting the input voltage to the minimum. For instance, if your supply is a Li-Ion battery with a nominal voltage of 3.6V and the haptic function needs to be supported down to 3.2V, you would set the supply at 3.2V for this test. Make sure your waveform doesn`t trigger a MAX_POWER warning as it is not possible to optimize the parameters from a waveform that requires more power than the IC can provide.

b. Minimum output power with the highest supply voltage. In this case, you will play a waveform that is expected to require the least amount of power. The lowest combination of voltage and frequency while setting the input voltage to the highest level. For instance, if your supply is a Li-Ion battery with a nominal voltage of 3.6V and maximum voltage of 4.2V, you would use a 4.2V supply for this test.

3. This final step explains how to adjust key parameters based on the waveform shape and the optimal order for adjustments to minimize time. It is recommended to use the maximum output power waveform to proceed with the adjustment and then validate if the parameters found also work well with the low output power waveform.
   a. Adjusting KP: KP defines the gain of the controller at low output voltage (near 0V).
      i. If KP is too high, you will see an oscillation in the output waveform near 0 V. Most often, the symptoms will be more apparent on the voltage decrease. For bipolar waveforms, the zero-crossing point will also show a step.
      ii. If KP is too low, it is more difficult to see it directly. On a bipolar waveform, if KP is too low, the transition near 0 will have a more pronounced flat section near 0V. Even if your application has a unipolar waveform, it may be a good idea to make the adjustment with a slightly bipolar waveform (ex: -10 V to + 90V).
      iii. Adjust the value of KP to obtain the best low-voltage behaviour possible. It is better to optimize KP with the maximum output power waveform and then validate that it also works well with the lowest output power.

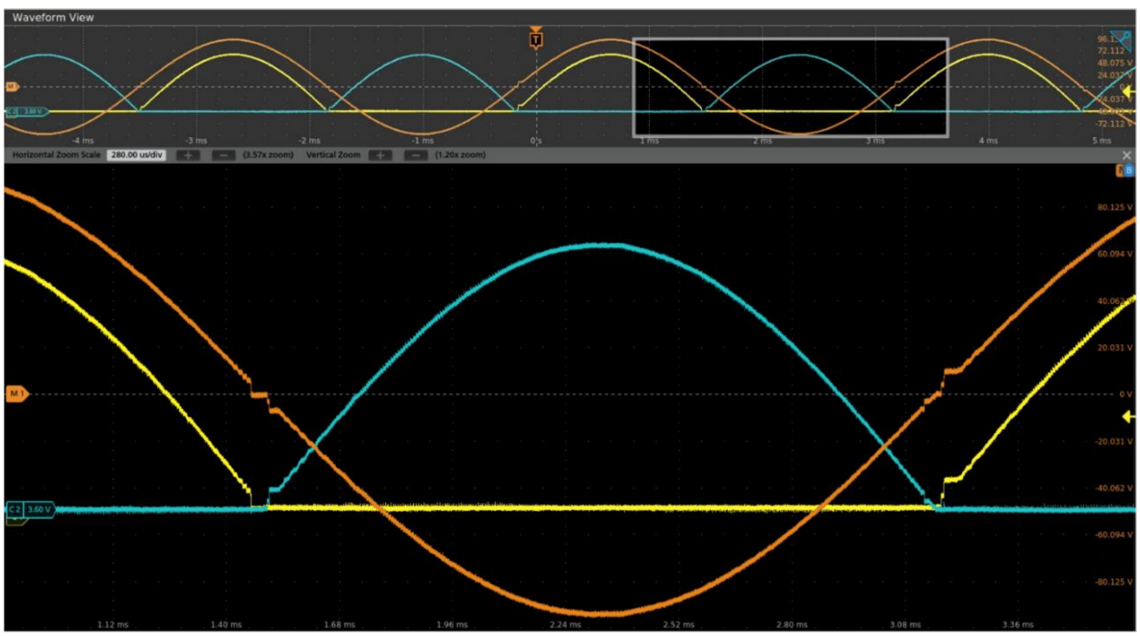


Figure 21: Example of a waveform with KP too high zoomed on zero crossing transition. We can see an oscillation before and after the zero-crossing as well as a step like function at 0V. Yellow: OUT+, Blue: OUT-, Orange: OUT – OUT-

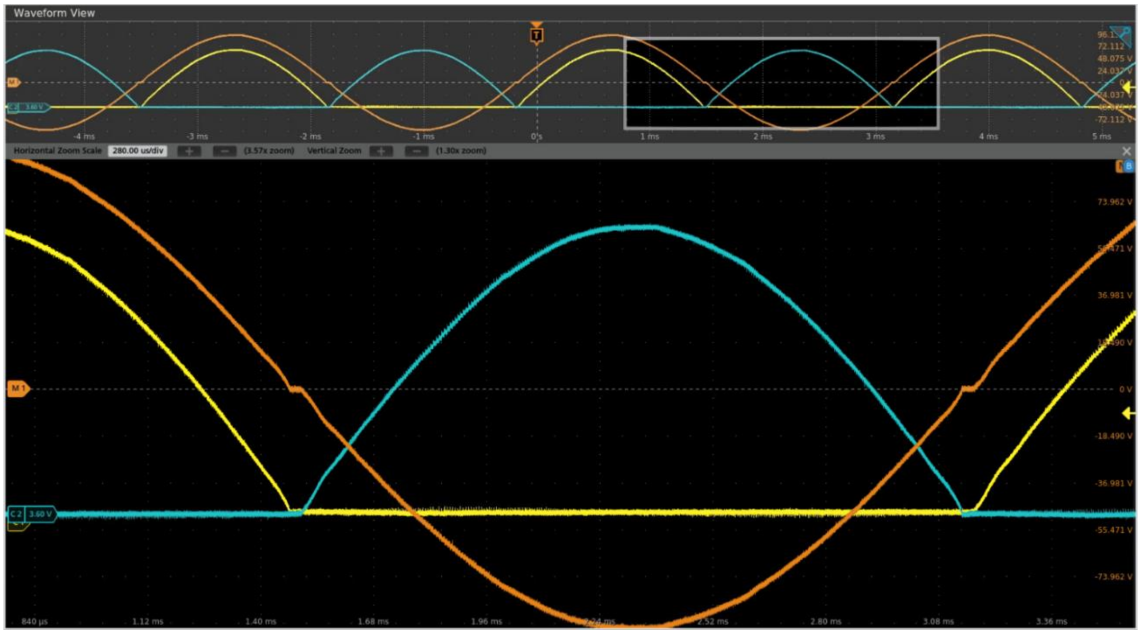


Figure 22: : Example of a waveform with KP value too low zoomed on zero-crossing transition. The waveform displays a shape that is similar to a flat section between the positive and negative value of the waveform. Yellow: OUT+, Blue: OUT-, Orange: OUT – OUT-

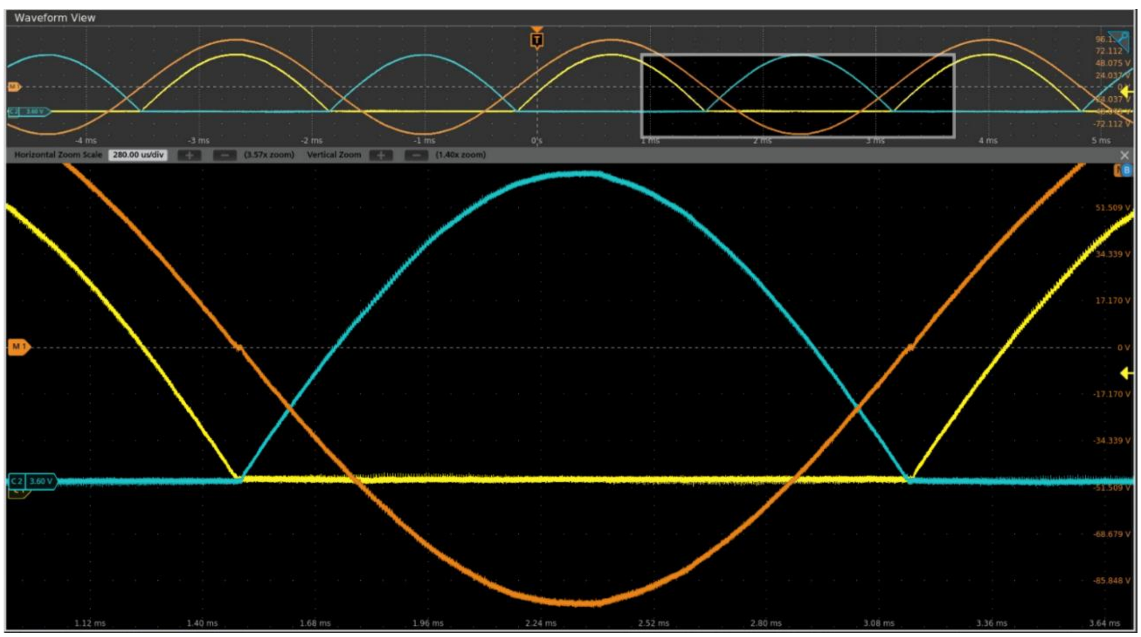


Figure 23: : Example of a waveform with proper KP value zoomed on zero crossing. Even when crossing zero, the waveform displays no sign of instability and not flat section. Yellow: OUT+, Blue: OUT-, Orange: OUT – OUT-

b. Adjusting KPA: KPA defines the gain at high output voltage. Its impact will typically be more pronounced above 50% of the output voltage.
    i. If KPA is too high, you will see an oscillation in the output waveform at high voltage. KPA needs to be reduced until this oscillation completely disappears. Note that this instability will depend significantly on the BOM and waveform. It

may never appear in some systems or show only in specific discontinuous waveforms (square waves, triangles, etc.).

ii. If KPA is too low, the main symptom is generally a flat waveform near the top of the waveform. In this scenario, KPA needs to be increased to minimize this discontinuity. Note that PARCAP may also be partially responsible for this discontinuity, so it may not be possible to remove it completely only with KPA.

iii. Adjust the value of KPA to obtain the best high-voltage behaviour. (Show a waveform with a perfect behaviour.)

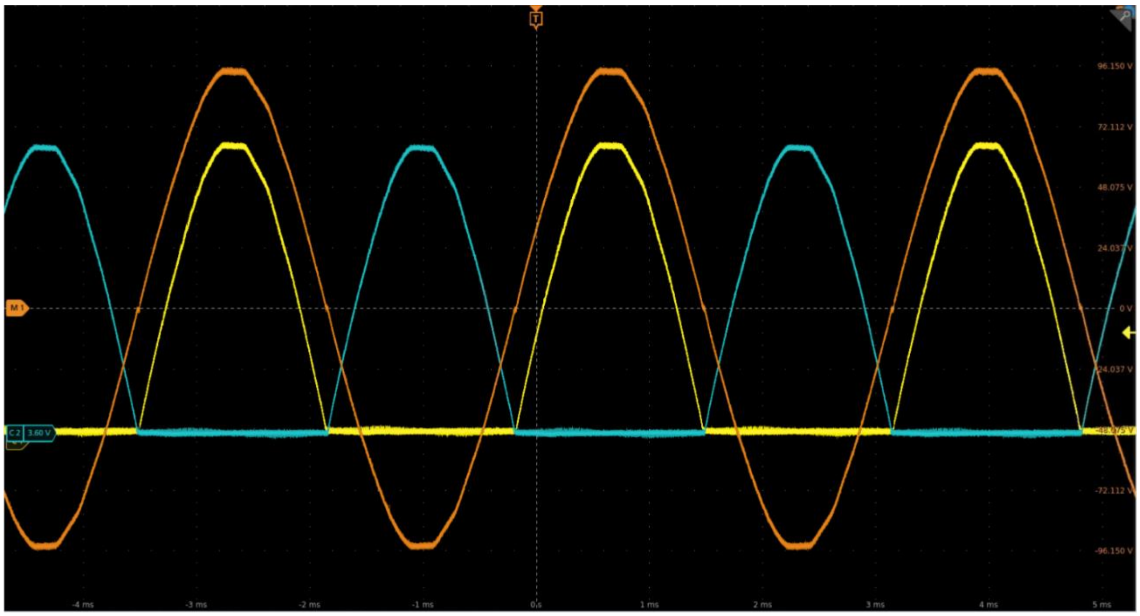


Figure 24: Example of a waveform with KPA too low

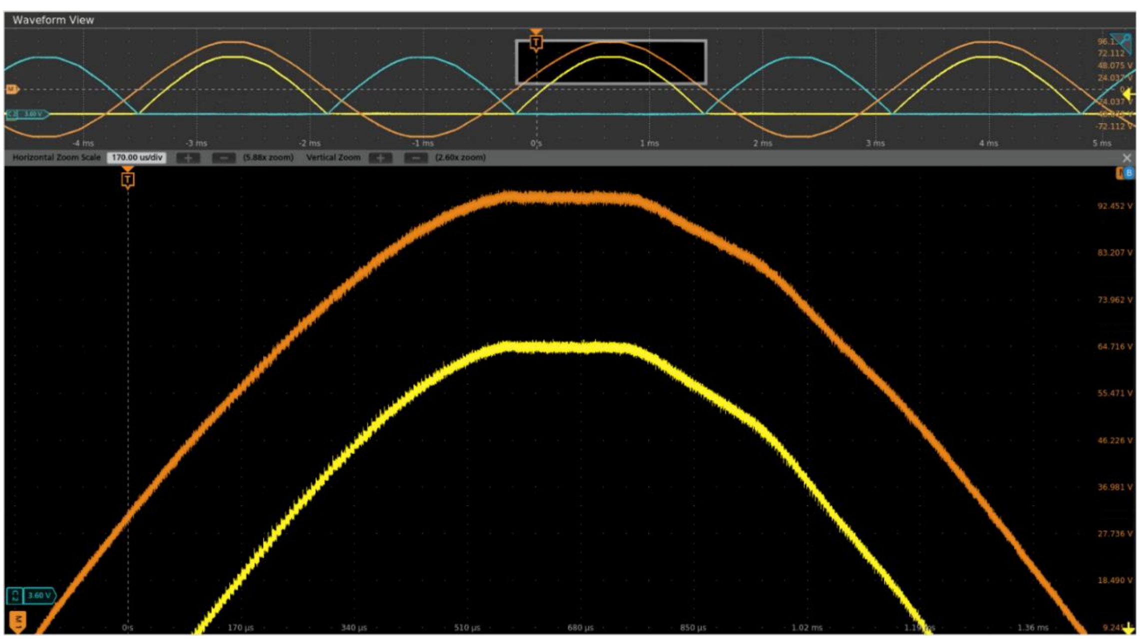


Figure 25: Example of a waveform with KPA too low zoom on the top portion of the waveform. We can see the top of the waveform is flat.

c. Adjusting PARCAP: PARCAP informs the controller of the parasitic capacitance on the SW node. Its value is important for the waveform shape and to minimize power consumption of the system.
   i. If PARCAP is too high, you will see near the top of the waveform, just after it started to come down something that looks like a notch pointing up.
   ii. If PARCAP is too low, another type of discontinuity will be seen like a capacitor discharging also near the top of the waveform.

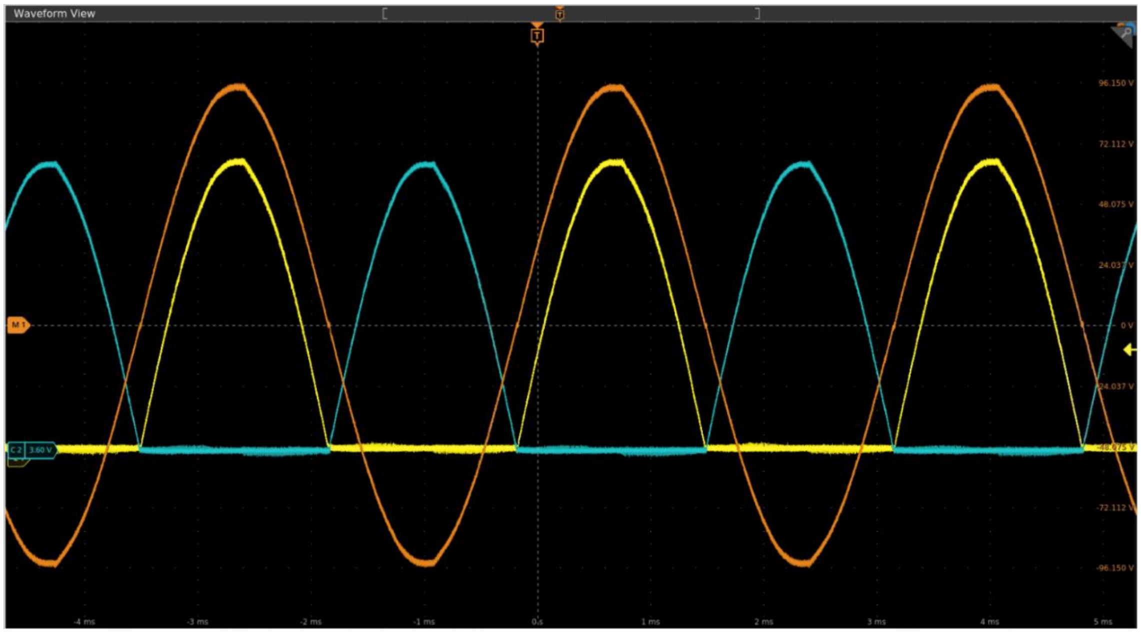


Figure 26: Example of a waveform with PARCAP too high

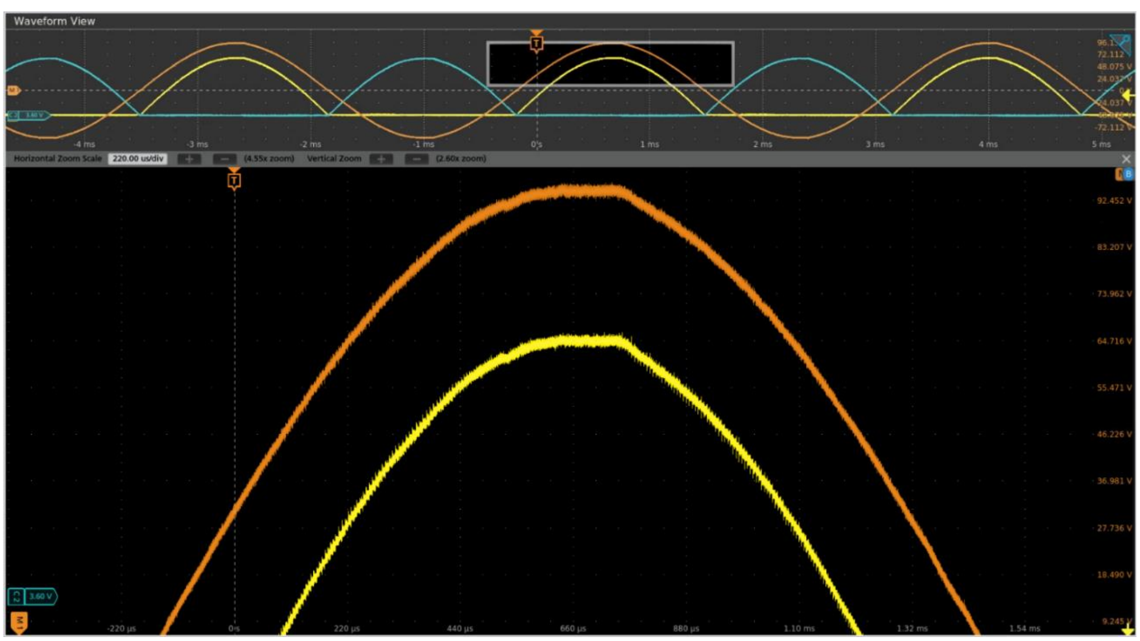


Figure 27: Example of a waveform with PARCAP too high zoomed on the top portion of the waveform. We can see the flatness in the top that creates a notch at the transition between the top and descending signal.

d. After adjusting these three parameters, it is possible that you still see small oscillations in the waveforms. To improve it, you can adjust empirically the following parameters
   i. FSWMAX: Decreasing FSWMAX will typically decrease any oscillation left but may lead to a slightly coarser waveform at low output power.
   ii. KI_BASE: Increasing KI_BASE will also help reduce oscillations. However, if KI_BASE greater than 4 is needed, you likely have another issue in your system. Recommended values are typically 2 or 3.

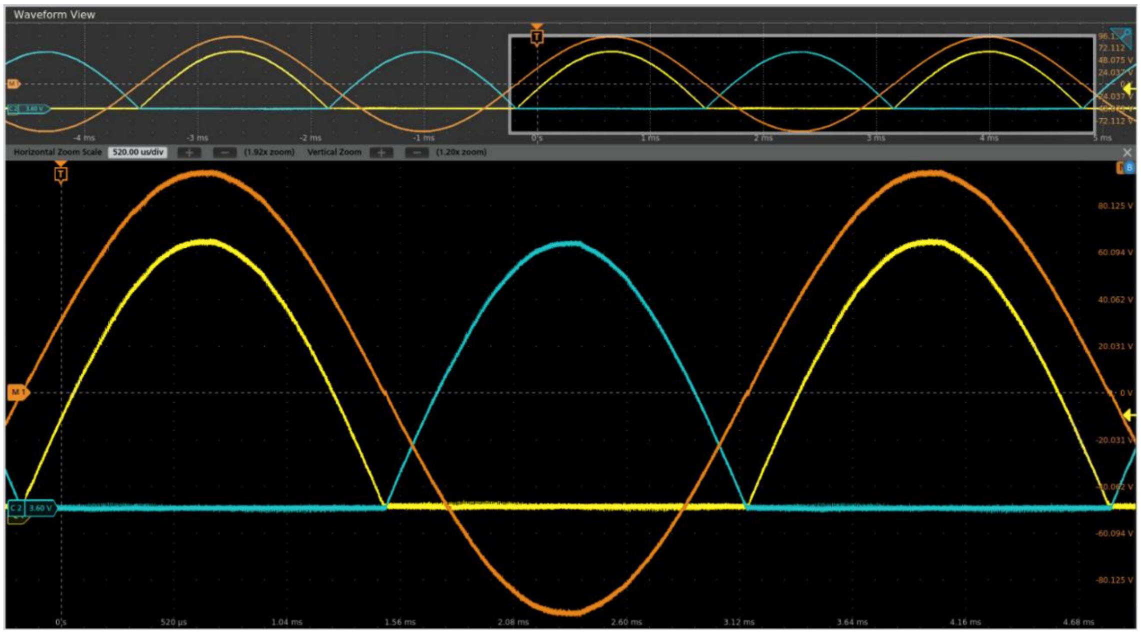


Figure 28: Optimized waveform. Crossings near zero are clean, waveform shows no sign of instability, top section of the waveform tracks well the desired sine waveform. Yellow: OUT+, Blue: OUT-, Orange: OUT – OUT-

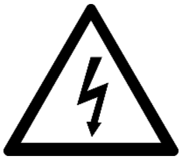
11 Related Products

	PRODUCT NAME	Description
1	BOS-HapticStudio	Desktop Application for Solid State Button and Haptic only prototype design

12 Document History

ISSUE	DATE	DOCUMENT NUMBER	CHANGES
1	November 2024	BT012AAN01.01	Original document

13 Notice and Warning



Danger High Voltage!

Electric shock possible when connecting board to live wire. Board should be handled with care by a professional. For safety, use of isolated test equipment with overvoltage and/or overcurrent protection is highly recommended.

ESD Caution

This product uses semiconductors that can be damaged by electrostatic discharge (ESD). When handling, care must be taken so that the devices are not damaged. Damage due to inappropriate handling is not covered by the warranty.

The following precautions must be taken:

- Do not open the protective conductive packaging until you have read the following and are at an approved anti-static workstation.
- Use a conductive wrist strap attached to a good earth ground.
- If working on a prototyping board, use a soldering iron or station that is marked as ESD-safe.
- Always disconnect the microcontroller from the prototyping board when it is being worked on.
- Always discharge yourself by touching a grounded bare metal surface or approved anti-static mat before picking up an ESD - sensitive electronic component.
- Use an approved anti-static mat to cover your work surface.

Oscilloscope measurements:

Both sides of the actuator (OUTx and VDD) are active outputs. When measuring these signals using an oscilloscope, use a separate probe on each output. Never connect the ground of a probe to one of the actuator terminals. Doing so might damage the BOS1921-KIT and/or your oscilloscope. For more information, please consult the *Probing BOS1901 with an Oscilloscope* application note available for download on [Boréas website.]