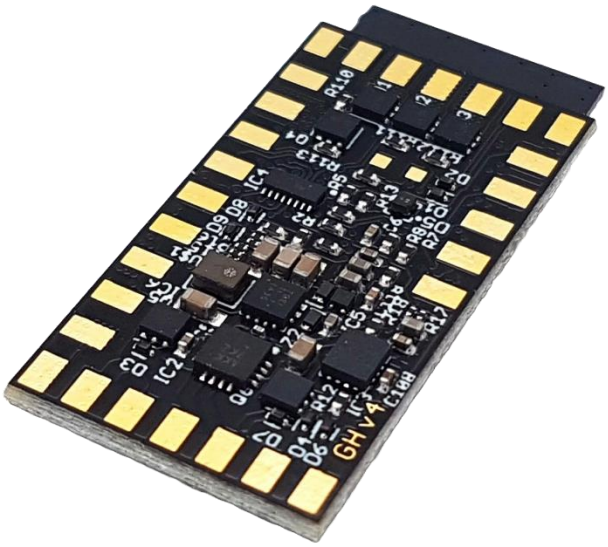
SABERTEC
GERMAN ENGINEERING POWER



Golden Harvest v4

Manual

SaberTec
Owner: Frederic Folz
Brentanoweg 3
66802 Überherrn
GERMANY

Contact: info@sabertec.net
Website: https://sabertec.net

06/10/2025

Thank you

Thank you for your unwavering passion and support! It is your enthusiasm that fuels our continuous innovation and dedication to enhancing our technology. Without your engagement, our vision would remain just that - a vision. Established in 2016, SaberTec was born out of a mission to create a soundboard that delivers the most dynamic and lifelike saber experience possible. After years of dedication, the Golden Harvest v3 marked a significant milestone in 2020. Today, we build on that foundation with the Golden Harvest v4, incorporating cutting-edge advancements and your invaluable feedback. Our commitment to community collaboration remains at the heart of our philosophy. We deeply value your ideas and suggestions and invite you to help shape the future of our products. Please share your thoughts with us at info@sabertec.net. We look forward to continuing this exciting journey together!

Safety notice

As our Golden Harvest board uses electronic parts, it is sensitive to electrostatic discharge. Please handle it with care and make sure to install it correctly. Furthermore, we recommend to use only protected Li-Ion batteries. Also make sure to check the sections "Soldering advice" and "Handling advice". We cannot be held responsible for any damage that arises from a faulty install or use of the board. Furthermore, be cautious when using the Golden Harvest v4 if you have photosensitive epilepsy or if you are photosensitive in general because the light effects can contain a series of flashing lights.

Legal information

We are not affiliated with The Walt Disney Company™ or Lucasfilm Ltd.™. All brands and trademarks mentioned are the exclusive property of their respective owners. Furthermore, we are not affiliated with Adafruit™. The trademark Neopixel™ is their exclusive property.

Content

Features of the Golden Harvest v4

The GHv4 hardware builds directly on the robust foundation of the GHv3, utilizing many of the same components. However, we have introduced select enhancements to bolster protection and introduce new functionalities. Some components are now housed in a more durable casing to enhance reliability. Building on the capabilities of the GHv3, the GHv4 introduces the exciting new features listed below.

Enhanced Power Management

- Optimized battery performance
- Features dual Batt. + and Batt. - pads that are internally connected to simplify wiring multiple connections.

Audio and Visual Upgrades

- Reinforced robustness of the audio amplifier
- Enhanced brightness for more vivid blade visuals
- Supports six accent LED channels and three data pads for Neopixel, dedicating individual pads for the main blade, side blades, and accent LEDs

Connectivity Improvements

- Increased USB and SD card speeds for faster data transfers
- Enhanced SD card reliability to reduce the probability of file corruption and extend SD card lifespan
- Wireless file transfer via WiFi:
    - Access the SD card as mass storage from a smartphone or a computer
    - Onboard antenna eliminates the need for an external one
    - Low power consumption
    - Optionally disableable
- Bluetooth control from smartphones:
    - Onboard antenna for Bluetooth, requiring no external antenna
    - Remote control capabilities allow changes to background effects, color profiles, and sound fonts
    - Minimal impact on battery runtime
    - Optionally disableable

Enhanced Diagnostics and Display

- Onboard status LED for troubleshooting (blinking patterns indicate specific error codes) and monitoring board status:
    - Minimal power usage
    - Optionally disableable
- OLED display support for showcasing static images, animations, and sound font names, fully customizable

System Enhancements

- Integrated dual-core processor for enhanced performance and multitasking capabilities
- Additional third co-processor for dedicated tasks, improving overall efficiency
- Features a real-time operating system (RTOS) for improved responsiveness
- Substantial increase in internal memory space, facilitating extensive future firmware developments

The GHv4 retains all the legacy features of the GHv3, detailed in the following sections. Many of them have been refined and improved.

Technical Specifications

- tiny dimensions of 34.5 mm x 17.9 mm x 3.7 mm / 1.36 ” x 0.7 ” x 0.15 ”
- powered by 3.7 V (single 18650 Li-ion cell for example)
- provides 10 LED channels that can be customized independently from each other:
  - 4 LED channels can deliver up to 9 A each:
    - can drive SingleCrees, DualCrees, TriCrees and QuadCrees
    - can drive Neopixel
  - 6 LED channels can deliver up to 20 mA each:
    - can drive accent, crystal chamber or switch LEDs for example
  - capable of delivering up to 36 A in sum (in practice, both battery and heat dissipation are the limiting factors here)
  - full customizability even of accent LEDs such as switch LEDs or crystal chamber LEDs using the whole range of our effect engine
- optional USB charging and file transfer feature (using our Seedling module, see below)
- microSD card slot

Installation and Use

- huge solder pads for easiest installation possible
- your hardware setup can be easily chosen on the microSD card
- full functionality accessible even if used with only one momentary button
- can also be used with two momentary button
- customizable deep sleep (hibernation mode)

Audio Engine

- maximum audio amplifier power output: 4 W
  - supported speaker power: 1 W - 6 W
  - supported speaker impedance: 4 Ohm, 8 Ohm, 16 Ohm and 32 Ohm
  - real-time dynamic range compression
  - 16 bit audio sample resolution
  - integrated speaker protection algorithms:
    - overtemperature protection
    - overcurrent protection
    - undervoltage protection
- full-blown sound effect engine:
  - unlimited smooth swing pair sounds
  - unlimited smooth swing accent sounds
  - unlimited swing sounds
  - unlimited clash sounds
  - unlimited stab sounds
  - unlimited spin sounds
  - unlimited begin lockup sounds
  - unlimited lockup sounds
  - unlimited end lockup sounds
  - unlimited begin melt sounds

- unlimited melt sounds
- unlimited end melt sounds
- unlimited begin drag sounds
- unlimited drag sounds
- unlimited end drag sounds
- unlimited blaster sounds
- unlimited force sounds
- unlimited boot sounds
- unlimited pre-on sounds
- unlimited power on sounds
- unlimited power off sounds
- unlimited post-off sounds
- unlimited hum sounds
- menu sounds
- unlimited sound fonts that contain all these sound effects
- the following sound fonts are supported without needing to rename any files: http://www.saberfont.com/Optimized-for-Plecter-CFX-Smooth-Swing-Compatible-by-Era-_c_92.html

- background music engine:
  - unlimited tracks
  - playlist feature
  - can be controlled on the fly
  - default package comprising 4 music tracks that come with the board:
    - "Fate and Fortune" by Scott Buckley
    - "Inflection" by Scott Buckley
    - "Light in Dark Places" by Scott Buckley
    - "Helios" by Scott Buckley
- default package comprising 36 sound fonts that come with the board:
  - "The Harvester" by BKSaberSounds
  - "Dark Harvest" by Blueforce Company
  - "Psy-Borg" by CrystalSoniX
  - "Son of Darkness" by Dark Path Media
  - "Shadows" by Echo Studio
  - "Daddy Issues Complete" by Fourzze FX
  - "The Classic Villain" by Fourzze FX
  - "Knighthood" by Fourzze FX
  - "Stitched" by Greyscale Fonts
  - "Mercenary" by Greyscale Fonts
  - "Awaken" by Greyscale Fonts
  - "Behemoth" by Greyscale Fonts
  - "LU-10" by JayDalorian
  - "Light Warrior" by JuanSith
  - "Dark Warrior" by JuanSith
  - "Peace" by Kyberphonic Fonts
  - "Violence" by Kyberphonic Fonts
  - "Guardian" by LDN Sabers
  - "The Eternal Prince" by LINK

- "Cryo Carbon" by LINK
- "Balance" by LordBlako Saber Fonts
- "Angelic Plazma" by Mountain Sabers
- "Dark Sith Red" by Mountain Sabers
- "Electric Essence" by Mountain Sabers
- "Jurassic" by Mountain Sabers
- "Kilovolt" by Mountain Sabers
- "Quantum Knight" by Mountain Sabers
- "The Mag-Cyl" by Mountain Sabers
- "The Prize" by Mountain Sabers
- "Torch" by Mountain Sabers
- "UnHinged" by Mountain Sabers
- "Princess Bride" by Phoenix Sabers
- "Dark Republic" by Phoenix Sabers
- "The Phantom" by Project Fonts
- "The Shadow" by Project Fonts
- "Energy Pike" by The Proplicator

Motion Detection

- both Smooth Swing and legacy motion detection supported
- Smooth Swing enhanced by Accent Swings and Accent Spins
- highly sophisticated legacy motion engine:
  - customizable sensitivity
  - ultra low motion-to-sound latency
  - detects swings
  - detects clashs
  - detects stabs
  - detects spins
- real-time measurement of orientation of the saber:
  - display of drag light and sound effects according to orientation of the saber
  - usage of pre-on sounds can be controlled by orientation of the saber
  - usage of post-off sounds can be controlled by orientation of the saber
  - responsive light effects
  - interactive menu navigation using orientation of the saber

Light Effects

- both in-hilt LEDs and Neopixel (ws2812b) are supported
- game-changing blade styles:
  - introduced as an analogue to sound fonts
  - contain all customizable light effect parameters
  - can define certain effect styles just as sound fonts define certain sound styles
  - can be changed on the fly
  - unlimited number of blade styles supported, i.e. you can have as many saber effect styles as you want, not only different colors
  - pairable with sound fonts on the fly as the Golden Harvest board remembers with which blade style a sound font was lastly used
  - default blade styles and manual available on https://sabertec.net/downloads

- in-hilt LED effects:
  - overwhelming light effect engine
  - full customizability of the effects
  - each LED channel can have completely independent light effects
  - customizability of crystal chamber effects for example
  - pulse effects:
    - conventional pulse
    - speed transition pulse, a.k.a. “HyperPulse”
    - intensity transition pulse, a.k.a. “ÜberPulse”
    - color transition effects
    - 6 different pulse shapes: sine, impulse, inverted impulse, rising sawtooth, falling sawtooth, rectangular
  - flicker effects:
    - conventional flicker
    - intensity transition, a.k.a. “PhaseFlicker”
    - 3 different flicker types: subtractive, additive and mixed
    - flicker synchronization
  - configurable fade out
  - configurable effect duration
  - independent basic effects
  - independent clash effects
  - independent stab effects
  - independent swing effects
  - independent spin effects
  - independent lockup effects
  - independent melt effects
  - independent blaster effects
  - independent force effects
  - as there are more than 600 parameters, you can literally invent your own light effects
  - assuming there are 1000 possible values for each parameter (which is a hard under-estimation), you could create 10^1800 (a one with 1800 zeros) possible blade styles – that’s more than there are atoms in the universe!
  - ingenious online configuration tool for designing, saving and sharing your light effects (will be released very soon)
  - also default parameter settings available, so you get already epic light effects even without having to change any of the parameters:
    - 10 presets for dedicated color LED
    - 120 presets for RGB LED
    - 120 presets for RGBW LED
    - 120 presets for RGBA LED
- Neopixel effects:
  - overwhelming light effect engine
  - full customizability of the effects
  - comes with three effect packages comprising more than 900 presets for Neopixel effects that can be changed on the fly:
    - basic effect package: includes 30 presets
    - default effect package: includes 140 presets

        - extended effect package: includes 924 presets
    - all Neopixel effects consist of background effects and on top effects that can be arbitrarily combined with each other
    - 6 different background effect categories:
        - Classic (e. g. Red, Green, Blue, Flicker, Pulse, …)
        - Color Flow (e. g. Rainbow, Dual Transition, Blizzard Transition, …)
        - Frozen Pulse (e. g. Spatial Pulse, Color Pulse, …)
        - Wave (e. g. Railgun, Pulsing Wave, Tiger Tail, Rattlesnake, Candy Stick, …)
        - Flame Blade (e. g. Red Flame, Green Flame, Blue Flame, …)
        - Gradient (e. g. Dual Phase, Pulsing Phase, …)
    - 7 different on top effect categories:
        - None
        - Focus Deflection
        - Thunder Storm
        - Unstable Blade
        - Kylo Blade
        - Magnetic Flare
        - Shockwave
    - configurable fade out
    - configurable effect transition:
        - configurable transition time
        - configurable transition type:
            - Basic
            - Window
            - Unstable Blade
            - Flame Blade
    - configurable effect duration
    - localized clash
    - localized lockup
    - localized drag
    - localized melt
    - localized blaster block
    - independent basic effects
    - independent clash effects
    - independent stab effects
    - independent swing effects
    - independent spin effects
    - independent lockup effects
    - independent blaster effects
    - independent force effects
- 6 independent channels to drive standard accent LEDs
- capability of defining unique light effects even for standard accent LEDs
- sequencer for standard accent LEDs:
    - up to 16 different steps
    - duration of each step customizable
    - break between steps customizable
- up to 8 independent Neopixel accent LEDs

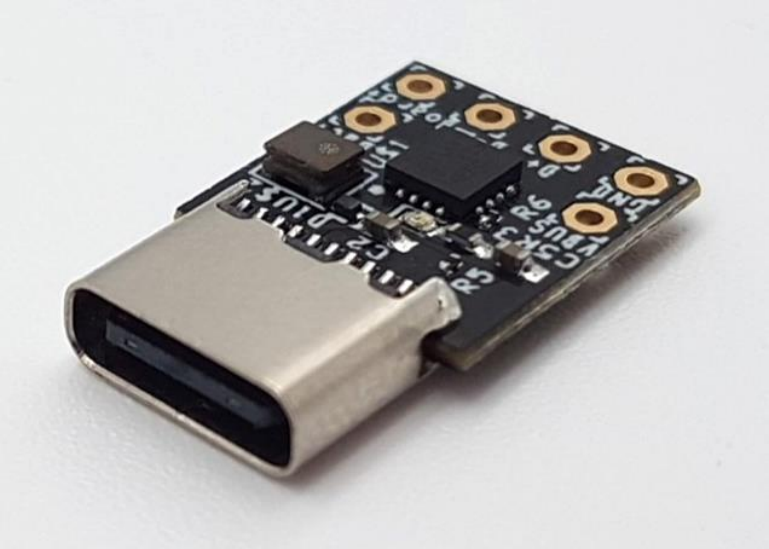
- capability of defining unique light effects even for Neopixel accent LEDs
- sequencer for Neopixel accent LEDs:
    - up to 16 different steps
    - duration of each step customizable
    - break between steps customizable

Firmware Updates

- there will be free firmware updates on a regular basis which you can simply install yourself by saving a file on your microSD card
- many more features will be introduced with firmware updates in future
- you're invited to take part in planning new features by discussing them with us or providing your own ideas in our Golden Harvest soundboard group on Facebook

USB Charging and File Transfer

- using our Seedling v2 USB-C module, the Golden Harvest can be recharged and the microSD can be accessed via USB-C
- charging with up to 2 A
- you can use any wall adapter independent from the maximum current it can supply thanks to a smart charging algorithm
- smaller than 18 mm x 11 mm (0.708" x 0.433")
- check the manual of the Seedling v2 for more details



Wiring of the Board

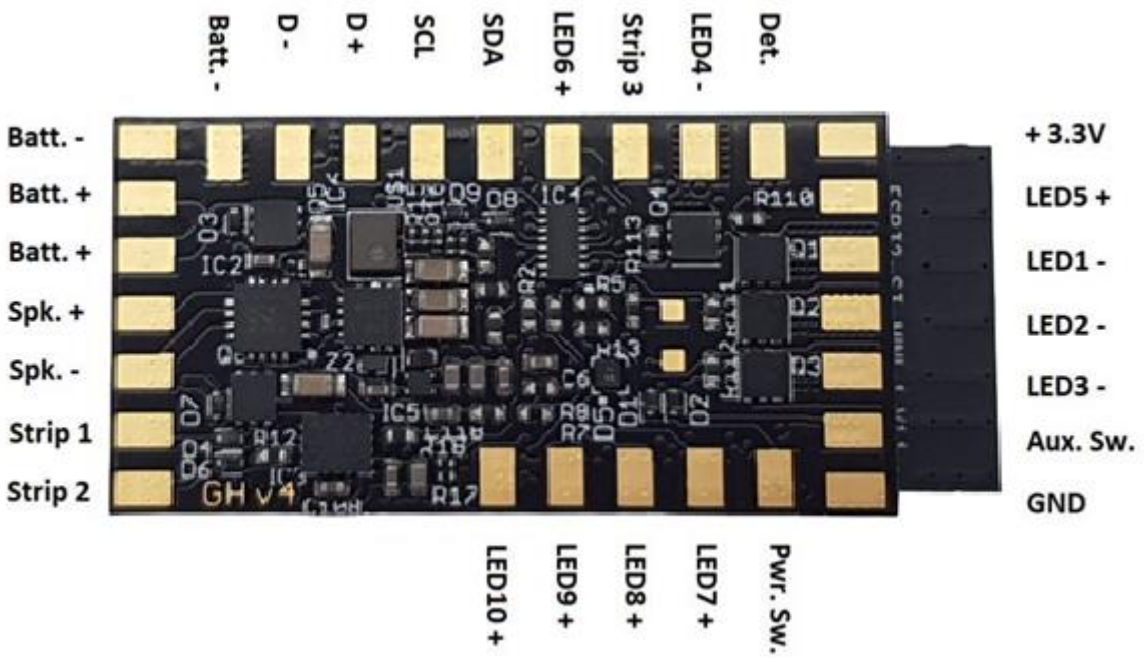
After familiarizing yourself with the technical specifications, we will now guide you through the correct installation of the Golden Harvest v4. Our board includes four high-power LED channels labeled "LED1", "LED2", "LED3", and "LED4". These should be connected to the negative terminals of your LEDs. It is crucial to use appropriate resistors when using in-hilt LEDs to avoid damage (note that we are not responsible for any damage resulting from the use of inappropriate resistors). Refer to your LEDs' datasheets to calculate the required resistance. For Neopixel setups, you must not use any resistors.

The high-power LED channels are versatile; while they can power your blade LEDs, they are also capable of driving accent LEDs. For instance, if two channels are used for the blade, the remaining can be utilized for a switch LED or crystal chamber LEDs. Each channel can theoretically support up to 450 accent LEDs, assuming each requires 20 mA.

You can create custom light effects for the accent LEDs, which may be synchronized with blade effects or set to operate independently, allowing for unique designs like independent crystal chamber effects.

For wiring the power switch, connect it to the "Pwr. Sw." pad; this switch should be a momentary type. If using a two-switch setup, add a second momentary switch to the "Aux. Sw." pad. Connect the second pin of each switch to the "GND" pad. Twisting the wires of each switch can reduce electromagnetic induction, which might otherwise cause erratic behavior. Also, use the "GND" pad located between the "Aux. Sw." and the "Pwr. Sw." pads for grounding.

To create wiring diagrams tailored to your specific setup, you can use our online tool:
https://sabertec.net/wiring-diagram-creator/



The table below lists the functions of each pad.

Pad	Function	Remark
Batt. -	Connect to negative of battery	Two identical pads that are internally connected
Batt. +	Connect to positive of battery	Two identical pads that are internally connected
Spk. -	Connect to negative of speaker	

Spk. +	Connect to positive of speaker	
Strip 1	Connect to data line of the Neopixel blade	Resistor is already included on the board
Strip 2	Connect to data line of the Neopixel accent LEDs	Resistor is already included on the board
Strip 3	Connect to data line of the Neopixel side blade	Resistor is already included on the board
D +	Connect to D + of the USB-C module	
D -	Connect to D - of the USB-C module	
SCL	Connect to SCL of an OLED display	
SDA	Connect to SDA of an OLED display	
Det.	Connect to Det. of the USB-C module	
+ 3.3V	Constant 3.3V output	Disabled while in deep sleep, can be used to power an OLED display for example
Aux. Sw.	Connect to auxiliary switch (must be momentary if used)	Second pin of the switch needs to be connected to GND
GND	Ground	
Pwr. Sw.	Connect to power switch (must be momentary)	Second pin of the switch needs to be connected to GND
LED1 -	High-power LED output (4,2 V and 9 A)	
LED2 -	High-power LED output (4,2 V and 9 A)	
LED3 -	High-power LED output (4,2 V and 9 A)	
LED4 -	High-power LED output (4,2 V and 9 A)	
LED5 +	Accent LED output (3,3V and 20 mA)	
LED6 +	Accent LED output (3,3V and 20 mA)	
LED7 +	Accent LED output (3,3V and 20 mA)	
LED8 +	Accent LED output (3,3V and 20 mA)	
LED9 +	Accent LED output (3,3V and 20 mA)	
LED10 +	Accent LED output (3,3V and 20 mA)	

Power Rating and Polarity of LED channels

All 10 LED channels of the Golden Harvest v4 are independently configurable. It features 4 high-power LED channels and 6 standard accent LED channels. The power rating and the polarity of the respective LED channels is shown below. The high-power channels can be bridged in any possible combination to deliver an even higher current, for example to drive Neopixels.

LED channel	**Polarity**	**Maximum current draw**
1	-	9 A
2	-	9 A
3	-	9 A
4	-	9 A
5	+	20 mA
6	+	20 mA
7	+	20 mA

8	+	20 mA
9	+	20 mA
10	+	20 mA

Soldering Advice

The pads on the Golden Harvest v4 are designed to be as large as possible for easy soldering. To prevent damage, keep the soldering temperature below 400°C (752°F) and avoid heating the pads for more than 1 - 2 seconds. Allow the board to cool down between soldering each pad.

Always solder while the board is not powered. Connect the battery only as the final step, or use a kill key to keep it disconnected during the soldering process. Ensure that the wires do not exert any tensile stress on the pads.

Handling Advice

In general, electronic components are sensitive to electrostatic discharge. Always ground yourself when handling the board to avoid damage.

Battery Recommendation

We recommend to use the following batteries for the respective LED setups.

In-Hilt Setup

For setups using a TriCree, most **protected** single-cell Li-ion batteries are suitable. We recommend a KeepPower 18650 protected battery, which we use in our demo sabers. For a crossguard setup with three separate TriCrees, use a KeepPower 18650 protected battery rated for at least 10A to ensure it can provide sufficient current.

Neopixel Setup

For Neopixel setups, we recommend a KeepPower 18650 protected battery rated for at least 15A.

Why Current Rating Matters

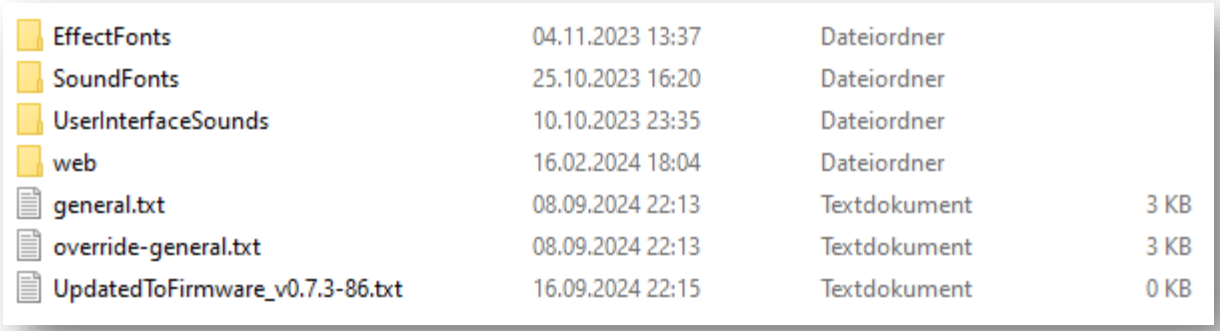
The current rating of the battery is crucial. Using a battery rated for only 3A, for example, in a Neopixel setup can cause a significant voltage drop. This not only dims the Neopixel but can also damage the battery over time due to the strain from insufficient current capacity.

Folder Structure on microSD Card

The GHv4 comes with a microSD card containing the default package. The image below shows the folder structure of the microSD card. Let's explore the contents from top to bottom:

- **EffectFonts**: This folder contains the different effect fonts, which are the light effect presets for your saber.
- **SoundFonts**: Contains all the sound fonts for your saber. The GHv4 comes preloaded with 16 sound fonts, but you can add as many as you like - there is no maximum limit.
- **UserInterfaceSounds**: Stores all user interface sounds, such as battery indicator sounds and volume control sounds.
- **web**: Contains files related to the Golden Harvest Commander, a web app that allows remote control of the board via WiFi (see the "WiFi Connectivity" section for details).
- **general.txt** and **override-general.txt** (if present): Configuration files used to customize your saber settings. These will be explained in detail in a later section.

- **UpdatedToFirmware_vX.X.X-XX.txt**: A file indicating the firmware version currently installed on the GHv4, such as "UpdatedToFirmware_v0.7.3-86.txt".



Setting Up Your Board

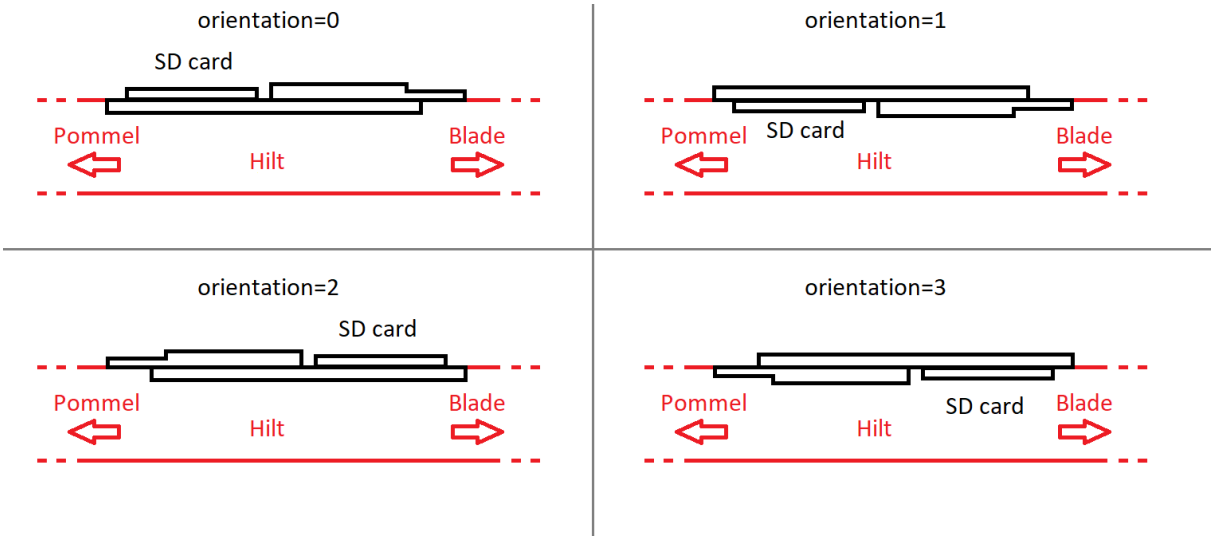
To set up the Golden Harvest v4 for your saber, follow these instructions to choose your hardware setup and configure your light effects.

Choosing Your Hardware Setup

Customize your hardware setup by editing the "general.txt" file on your microSD card. You can do this with any standard text editor or the Golden Harvest Commander, a web app that allows you to edit files via WiFi (see the " WiFi Connectivity" section for details). To configure your board, you only need to adjust three key parameters, as described below.

If an "override-general.txt" file is also present, it will take precedence over the "general.txt" values. Therefore, make sure to apply any changes to both files to ensure your configuration is correctly applied.

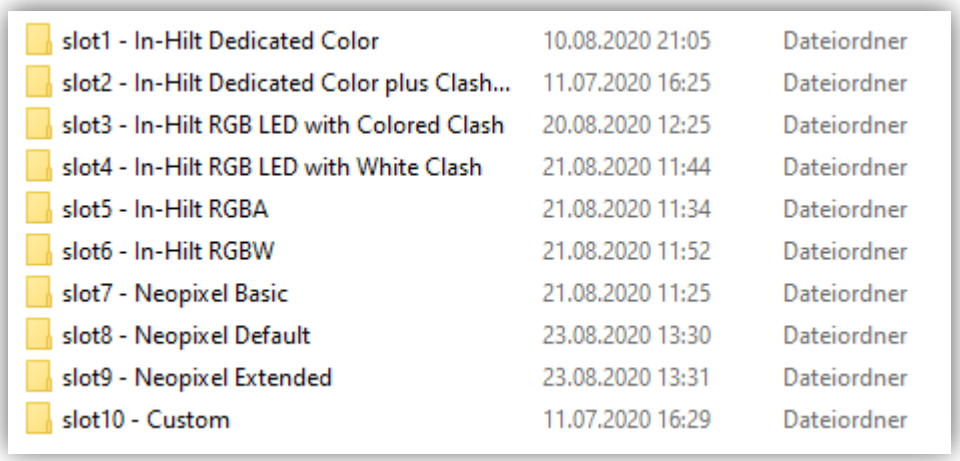
PARAMETER	DESCRIPTION
button_mode	Defines which button mode you want to use. The following modes are available: • **button_mode=4**: single momentary switch • **button_mode=2**: two momentary switches
orientation	Sets the orientation of the board in the saber for proper function of orientation-based features. The different orientations are illustrated in the picture below. • **orientation=0**: The SD card is on top, and the LED pads 1-3 point towards the blade. • **orientation=1**: The SD card is on the bottom, and the LED pads 1-3 point towards the blade. • **orientation=2**: The SD card is on top, and the LED pads 1-3 point towards the pommel. • **orientation=3**: The SD card is on the bottom, and the LED pads 1-3 point towards the pommel.
number_of_leds	Specifies the number of pixels in a Neopixel blade. If unsure, check with the blade manufacturer or use the pixel number edit mode in the Saber Editor for visual feedback. Typically, this number ranges from 110 to 135 and is internally capped at 144.



While there are many other customizable parameters in the "general.txt" file, only these three are necessary to set up your Golden Harvest v4 for your hardware. The "button_mode" and "orientation" parameters must be edited directly in the "general.txt" and "override-general.txt" files. However, the "number_of_leds" parameter can also be adjusted on the fly using the Saber Editor.

Choosing Your Light Effects

Customize your light effects by adjusting the "effect_font_slot" parameter in the "general.txt" file on your microSD card. The image below shows the values corresponding to different LED setups. For Neopixel setups, three default packages are available, each varying in the number of included presets.



You can also change the effect font slot dynamically using the Saber Editor, providing a flexible way to experiment with and adjust light effects.

Note: If you have a Neopixel saber but select an effect font slot intended for an in-hilt saber, the blade will not light up. Similarly, if you select a Neopixel effect font slot for an in-hilt saber, the in-hilt LED will remain fixed in color with no effects.

Deep Sleep

The Golden Harvest board includes a Deep Sleep mode that extends shelf life to several months. By default, the board enters Deep Sleep after 5 minutes of inactivity, but this duration can be customized

by adjusting the "deep_sleep" parameter (see the "General Settings" section for details). To wake the board from Deep Sleep, simply press the power button.

Motion Detection Engine

The Golden Harvest v4 features a highly advanced motion detection engine, supporting both Smooth Swing and Legacy Motion Detection modes. The board automatically switches to Legacy Motion Detection whenever a sound font lacks smooth swing pairs. These modes are detailed below.

Smooth Swing

The Golden Harvest v4 offers a Smooth Swing engine that generates dynamic swing sound effects in real-time, based on the saber's motion speed, creating a highly realistic experience. The board measures the saber's rotation speed and produces corresponding swing sounds. Additionally, accent swings are incorporated, with their volume modulated by the saber's rotation speed.

Legacy Motion Detection

The Legacy Motion Detection engine can detect four distinct movements: swings, clashes, stabs, and spins. Despite using a complex detection algorithm, the system requires only a minimal set of parameters for calibration. Each type of motion has a threshold parameter that determines its sensitivity; a higher threshold decreases sensitivity, requiring stronger motion to trigger the effect. For spins, there is an additional trigger duration parameter, which specifies the continuous duration that the motion must exceed the threshold to register as a spin. More details about these parameters can be found in the "General Settings" section.

General Settings

The general settings of your Golden Harvest v4 board can be found in the "general.txt" file on your microSD card. When the board boots, it loads all the parameters that are contained in this file. The parameters define the behavior of your board. You can customize these settings by adjusting the values of the respective parameters using the format "parameter=value". This can be done with a text editor or by using our configurator. In addition to the "general.txt" file, there may be an "override-general.txt" file that stores parameters that have been changed on the fly using the Saber Editor. As ist name suggests, the parameters contained in this file override the parameters in the "general.txt" file if they are contained in both files.

PARAMETER	DESCRIPTION
volume	Sets the maximum volume, with a range from 0 (mute) to 400 (maximum loudness). Recommended value is 100, but lower values may be needed for some speakers to prevent overdriving, which is generally indicated by a distorted sound.
button_off_time	Defines how long the power button must be pressed to turn the saber off. Time is specified in units of 6 milliseconds.
button_on_time	Defines how long the power button must be pressed to turn the saber on. Time is specified in units of 6 milliseconds.
button_short_time	In menu navigation, this parameter defines the duration a button must be pressed to be recognized as a short press. Time is specified in units of 6 milliseconds.
button_normal_time	In menu navigation, this parameter defines the duration a button must be pressed to be recognized as a normal press. Time is specified in units of 6 milliseconds.

button_long_time	In menu navigation, this parameter defines the duration a button must be pressed to be recognized as a long press. Time is specified in units of 6 milliseconds.
button_mode	Defines which switch configuration you're using. • 4: single momentary switch • 2: two momentary switches
button_reverse	Swaps the behavior of the power button and the auxiliary button when set to 1. No change when set to 0.
stab_ignition	Enables or disables saber ignition by a stabbing motion. Set to 0 to disable or 1 to enable. Customizing this parameter in the "general.txt" file only takes effect if the sound font does not contain a "settings.txt" file that includes this parameter. Otherwise, the value in the "settings.txt" will be used, allowing the stab ignition to be specific to the sound font.
swing_ignition	Enables or disables saber ignition by a swinging motion. Set to 0 to disable or 1 to enable. Customizing this parameter in the "general.txt" file only takes effect if the sound font does not contain a "settings.txt" file that includes this parameter. Otherwise, the value in the "settings.txt" will be used, allowing the swing ignition to be specific to the sound font.
swing_ignition_threshold	Sets the sensitivity for swing ignition. Lower values increase sensitivity, making swings easier to trigger, while higher values decrease sensitivity, requiring a stronger swing to ignite the saber.
twist_ignition	Enables or disables saber ignition by a twisting motion. Set to 0 to disable or 1 to enable. Customizing this parameter in the "general.txt" file only takes effect if the sound font does not contain a "settings.txt" file that includes this parameter. Otherwise, the value in the "settings.txt" will be used, allowing the twist ignition to be specific to the sound font.
twist_retraction	Enables or disables saber retraction by a twisting motion. Set to 0 to disable or 1 to enable. Customizing this parameter in the "general.txt" file only takes effect if the sound font does not contain a "settings.txt" file that includes this parameter. Otherwise, the value in the "settings.txt" will be used, allowing the twist retraction to be specific to the sound font.
twist_protection	Prevents twist retraction when the saber's movement exceeds the set value. Lower values increase sensitivity, making it easier to accidentally trigger a twist retraction during movement, while higher values decrease sensitivity, reducing the chance of accidental shutdown.
preon	Controls how preon sounds and light effects are triggered. • 0: Disabled. • 1: Play with every ignition. • 2: Play only when saber pointed downwards at ignition. • 3: Play only when saber pointed upwards at ignition. • 4: Play only with power button ignition. • 5: Play only with gesture ignition.
postoff	Controls how postoff sounds and light effects are triggered. • 0: Disabled.

	• 1: Play with every retraction. • 2: Play only when saber is pointed downwards at retraction. • 3: Play only when saber is pointed upwards at retraction. • 4: Play only with power button retraction. • 5: Play only with gesture retraction.
blaster	Controls how Blaster Blocks are triggered. • 0: Tapping the power button (one button setup) or the auxiliary button (two button setup). • 1: Swinging the saber after entering MultiBlast mode, which is activated by tapping the power button (one button setup) or the auxiliary button (two button setup). If no swing is detected within a customizable time, MultiBlast mode exits automatically. Tutorial videos: • One button setup • Two button setup
blaster_disable	Time in milliseconds after which MultiBlast mode is exited automatically if no swing is detected.
blaster_threshold	Defines the swing strength required to trigger a Blaster Block in MultiBlast mode.

lockup	Controls how lockup effects are triggered.
	• One Button Setup:
	  ○ 0: Holding the power button, releasing it, and then twisting the saber.
	  ○ 1: Holding the power button, releasing it, and then triggering a clash.
	  ○ Tutorial video
	• Two Button Setup:
	  ○ 0: Holding the auxiliary button and then tapping it.
	  ○ 1: Holding the auxiliary button, releasing it, and then triggering a clash.
	  ○ Tutorial video

force_push	Controls how force push effects are triggered.
	• One Button Setup:
	  ○ 0: Twisting the saber while holding the button.
	  ○ 1: Swinging the saber while holding the button.
	  ○ Tutorial video
	• Two Button Setup:
	  ○ 0: Tapping the power button.
	  ○ 1: Swinging the saber while holding the auxiliary button.
	  ○ Tutorial video

force_push_threshold	Sets the required swing strength to trigger a force push when the force push is activated by swinging the saber while holding the button. Lower values increase sensitivity, making it easier to trigger a force push with lighter swings, while higher values decrease sensitivity, requiring a stronger swing.

accent_menu	Defines if the accent menu is enabled or disabled. A value of 1 enables it and a value of 0 disables it.
color_menu	Defines if the color menu is enabled or disabled. A value of 1 enables it and a value of 0 disables it.
color_menu_twist_speed	Adjusts how much you need to turn the saber to change colors while in the color menu. Lower values mean you don't have to turn the saber as much to see color changes, making it easier to switch colors with small twists. Higher values require you to turn the saber more to change colors.
effect_menu	Defines if the effect menu is enabled or disabled. A value of 1 enables it and a value of 0 disables it.
effect_menu_twist_speed	Adjusts how much you need to turn the saber to change the blade style while in the effect menu. Lower values mean you don't have to turn the saber as much to change the blade style, making it easier to blade styles with small twists. Higher values require you to turn the saber more to change the blade style.

sound_menu	Controls how you navigate through sound fonts in the sound menu.
	• One Button Setup:
	  ○ 1: Navigate through sound fonts by slowly twisting the saber; twist clockwise for the next sound font and counterclockwise for the previous sound font. Tap the button while pointing the saber upwards to jump forward or downwards to jump backward through sound fonts, as defined by "sound_font_jump".
	  ○ 2: Navigate by pointing the saber upwards or downwards while twisting in any direction; orientation determines the direction of navigation. Tap the button while pointing the saber upwards to jump forward or downwards to jump backward through sound fonts, as defined by "sound_font_jump".
	  ○ 3: Tap the button to navigate to the next or previous sound font based on the saber's orientation. Quickly twist the saber while pointing upwards to jump to the next sound font or downwards to jump to the previous one, as defined by "sound_font_jump".
	  ○ Tutorial video
	• Two Button Setup:
	  ○ 1: Navigate through sound fonts by slowly twisting the saber; twist clockwise for the next font and counterclockwise for the previous font.
	  ○ 2: Navigate to the next or previous sound font by pointing the saber upwards or downwards while twisting. The twist direction does not matter.
	  ○ 3: Use the auxiliary button to navigate to the next or previous sound font depending on

	whether the saber is pointing upwards or downwards. Tap the power button while pointing the saber upwards to jump forward or downwards to jump backward by a number of fonts specified in "sound_font_jump". ○ Tutorial video
sound_font_jump	Specifies the number of sound fonts to skip when jumping forward or backward in the sound menu.
sound_menu_twist_speed	Adjusts how much you need to turn the saber to change sound fonts while in the sound menu. Lower values mean you don't have to turn the saber as much to switch sound fonts, making it easier to navigate with small twists. Higher values require you to turn the saber more to cycle through sound fonts.
volume_control	Manages the volume control feature. • 0: Disabled. • 1: Enables volume adjustment by tilting the saber. • 2: Enables volume adjustment by twisting the saber. • 3: Allows volume adjustment by both tilting and twisting the saber.
volume_control_tilt_speed	Adjusts how much you need to tilt the saber to change the volume while in volume control mode. Lower values mean you don't have to tilt the saber as much to change the volume, making it easier to change the volume with small movements. Higher values require you to tilt the saber more to change the volume.
volume_control_twist_speed	Adjusts how much you need to turn the saber to change the volume while in volume control mode. Lower values mean you don't have to turn the saber as much to change the volume, making it easier to change the volume with small twists. Higher values require you to turn the saber more to change the volume.
color_profile_edit_mode	Controls how new colors are assigned to the selected color profile while in the color profile edit mode of the Saber Editor. • 0: Disabled. • 1: Assign new colors by tilting the saber. 2: Assign new colors by twisting the saber. • 3: Assign new colors by both tilting and twisting the saber. Tutorial videos: • One button setup • Two button setup
color_profile_tilt_speed	Adjusts how much you need to tilt the saber to change the color while in the color profile edit mode of the Saber Editor. Lower values mean you don't have to tilt the saber as much to change the color, making it easier to adjust with small movements. Higher values require you to tilt the saber more to change the color.
color_profile_twist_speed	Adjusts how much you need to turn the saber to change the color while in the color profile edit mode of the Saber Editor. Lower values mean you don't have to turn the saber as much

	to change the color, making it easier to adjust with small twists. Higher values require you to turn the saber more to change the color.
pixel_number_edit_mode	A value of 1 enables the pixel number edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
pixel_number_twist_speed	Adjusts how much you need to twist the saber to change the pixel number while in the pixel number edit mode of the Saber Editor. Lower values mean you don't have to twist the saber as much to change the pixel number, making it easier to adjust with small twists. Higher values require more twisting to make changes.
ignition_speed_edit_mode	A value of 1 enables the ignition speed edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
ignition_speed_twist_speed	Adjusts how much you need to twist the saber to change the ignition speed while in the ignition speed edit mode of the Saber Editor. Lower values mean you don't have to twist the saber as much, making it easier to adjust ignition speed with small twists. Higher values require more twisting to make changes.
retraction_speed_edit_mode	A value of 1 enables the retraction speed edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
retraction_speed_twist_speed	Adjusts how much you need to twist the saber to change the retraction speed while in the retraction speed edit mode of the Saber Editor. Lower values mean you don't have to twist the saber as much, making it easier to adjust retraction speed with small twists. Higher values require more twisting to make changes.
ignition_type_edit_mode	A value of 1 enables the ignition type edit mode of the Saber Editor and a value of 0 disables it.
ignition_type_twist_speed	Adjusts how much you need to twist the saber to change the ignition type while in the ignition type edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to switch ignition types with small twists. Higher values require more twisting to make changes.
retraction_type_edit_mode	A value of 1 enables the retraction type edit mode of the Saber Editor and a value of 0 disables it.
retraction_type_twist_speed	Adjusts how much you need to twist the saber to change the retraction type while in the retraction type edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to switch retraction types with small twists. Higher values require more twisting to make changes.
battle_configuration_edit_mode	A value of 1 enables the battle configuration edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup

	• Two button setup
battle_configuration_twist_speed	Adjusts how much you need to twist the saber to change the battle configuration while in the battle configuration edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust configurations with small twists. Higher values require more twisting to make changes.
gesture_controls_edit_mode	A value of 1 enables the gesture controls edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
gesture_controls _twist_speed	Adjusts how much you need to twist the saber to turn a gesture control on or off while in the gesture controls edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to turn a gesture control on or off with small twists. Higher values require more twisting to make changes.
wifi_setup_edit_mode	A value of 1 enables the WiFi setup edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
wifi_setup_twist_speed	Adjusts how much you need to twist the saber to change the WiFi settings while in the WiFi setup edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust WiFi settings with small twists. Higher values require more twisting to make changes.
ble_setup_edit_mode	A value of 1 enables the Bluetooth setup edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
ble_setup_twist_speed	Adjusts how much you need to twist the saber to change the Bluetooth settings while in the Bluetooth setup edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust Bluetooth settings with small twists. Higher values require more twisting to make changes.
hilt_calibration_edit_mode	A value of 1 enables the hilt calibration edit mode of the Saber Editor and a value of 0 disables it. Tutorial videos: • One button setup • Two button setup
effect_font_slot_edit_mode	A value of 1 enables the effect font slot edit mode of the Saber Editor and a value of 0 disables it.
effect_font_slot_twist_speed	Adjusts how much you need to twist the saber to change the effect font slot while in the effect font slot edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust the effect font slot with small twists. Higher values require more twisting to make changes.
orientation	Sets the orientation of the board in the saber for proper function of orientation-based features. • 0: The SD card is on top, and the LED pads 1-3 point towards the blade. • 1: The SD card is on the bottom, and the LED pads 1-3 point towards the blade.

	• 2: The SD card is on top, and the LED pads 1-3 point towards the pommel. • 3: The SD card is on the bottom, and the LED pads 1-3 point towards the pommel.
swing_threshold	Sets the minimum rotation speed needed to trigger a swing when using the legacy motion detection (instead of Smooth Swing). To set the sensitivity of the accent swings when using Smooth Swing, refer to the parameter "accent_swing_threshold". Lower values increase sensitivity, making swings easier to detect, while higher values decrease sensitivity, requiring faster swings to be detected.
clash_threshold	Sets the minimum impact strength needed to trigger a clash. Lower values increase sensitivity, making clashes easier to detect, while higher values decrease sensitivity, requiring stronger impacts to be detected.
stab_threshold	Sets the minimum force required to trigger a stab. Lower values increase sensitivity, making stabs easier to detect, while higher values decrease sensitivity, requiring more forceful stabs to be detected.
spin_threshold	Sets the minimum rotation speed needed to trigger a spin when using the legacy motion detection (instead of Smooth Swing). To set the sensitivity of the accent spins when using Smooth Swing, refer to the parameter "accent_spin_threshold". Lower values increase sensitivity, making spins easier to detect, while higher values decrease sensitivity, requiring faster rotations to be detected.
spin_trigger_duration	Sets the duration you must maintain the minimum rotation speed (as defined by "spin_threshold") to trigger a spin. Lower values increase sensitivity, allowing spins to be detected with shorter rotations, while higher values decrease sensitivity, requiring longer continuous rotations.
accent_swing_threshold	Sets the minimum rotation speed needed to trigger an accent swing when using Smooth Swing. Lower values increase sensitivity, making accent swings easier to detect, while higher values decrease sensitivity, requiring faster swings to be detected. Customizing this parameter in the "general.txt" file only takes effect if the sound font does not contain a "settings.txt" file that includes this parameter. Otherwise, the value in the "settings.txt" will be used, allowing the accent swing threshold to be specific to the sound font.
accent_spin_threshold	Customizing this parameter in the "general.txt" file only takes effect if the sound font does not contain a "settings.txt" file that includes this parameter. Otherwise, the value in the "settings.txt" will be used, allowing the accent spin threshold to be specific to the sound font.
twist_threshold	Sets the minimum twist speed around the pommel-to-tip axis needed to recognize a quick twist motion. Lower values increase sensitivity, making quick twists easier to detect, while higher values decrease sensitivity, requiring faster twists to be detected. This does not apply to slow rotations, which are also referred to as twists.

swing_cooldown	Sets the duration in milliseconds after a swing is detected during which no additional swings can be triggered. Helps prevent rapid, consecutive swing detections.
clash_cooldown	Sets the duration in milliseconds after a clash is detected during which no additional clashes can be triggered. Helps prevent rapid, consecutive clash detections.
stab_cooldown	Sets the duration in milliseconds after a stab is detected during which no additional stabs can be triggered. Helps prevent rapid, consecutive stab detections.
spin_cooldown	Sets the duration in milliseconds after a spin is detected during which no additional spins can be triggered. Helps prevent rapid, consecutive spin detections.
blasterblock_protection	Sets the duration in milliseconds after a Blaster Block is triggered during which no motion can be detected. Applies only to the legacy motion detection (instead of Smooth Swing); a higher value provides greater protection against interruptions.
forcepush_protection	Sets the duration in milliseconds after a Force Push is triggered during which no motion can be detected. Applies only to the legacy motion detection (instead of Smooth Swing); a higher value provides greater protection against interruptions.
effect_font_slot	Selects the effect font slot in use, determining which blade style library is applied. Different slots correspond to different blade style libraries for both In-Hilt and Neopixel setups.
effect_fading	Defines the transition time in milliseconds for how the blade visually changes when switching between different background effects, on top effects, or color profiles.
number_of_leds	Specifies the number of pixels in a Neopixel blade. If unsure, check with the blade manufacturer or use the pixel number edit mode in the Saber Editor for visual feedback. Typically, this number ranges from 110 to 135 and is internally capped at 144.
number_of_neopixel_accents	Specifies the number of Neopixel accents you are using.
data	Specifies the GHv4 pad to which Neopixel accent LEDs are connected. Neopixel accents can always use the Strip 3 pad, but additional pads can be enabled based on the parameter value. For the additional pads, the "number_of_neopixel_accents" must be set. • 0: No additional pads are used. • 1: Data is also sent to the Strip 1 pad (used for the blade). Neopixel accent LEDs must be connected in series with the blade. • 2: Data is also sent to the Strip 2 pad. • 3: Data is sent to both Strip 1 and Strip 2 pads.

blade_detect	Controls the blade detection feature, including automatic configuration and sound effects for blade insertion or removal.
	• For connectors without illumination or with illumination (separate data line):
	  ○ 0: Blade detect feature disabled.
	  ○ 1: Enabled with automatic configuration. Plays insertion/removal sounds for Neopixel blades only, not for in-hilt adapters.

	○ 2: Enabled without automatic configuration. Plays insertion/removal sounds for Neopixel blades only, not for in-hilt adapters. • For connectors with illumination (single data line): ○ 0: Blade detect feature disabled. ○ 1: Enabled with automatic configuration. Plays insertion/removal sounds for TriCree to Neopixel adapters only, not for Neopixel blades. ○ 2: Enabled without automatic configuration. Plays insertion/removal sounds for TriCree to Neopixel adapters only, not for Neopixel blades. Check the section "Blade detect" for more details.
blade_detect_protection	Sets the duration in milliseconds for which the blade detect feature is disabled after detecting the insertion or removal of the TriCree to Neopixel adapter.
in_hilt_slot	Specifies the effect font slot to use when an in-hilt blade is detected.
pixel_slot	Specifies the effect font slot to use when a Neopixel blade is detected.
usb	Enables or disables the USB file transfer feature. Set to 0 to disable and 1 to enable. When enabled, the SD card appears as a mass storage device when the board is connected to a computer.
power_save	Sets the percentage by which the blade's brightness is reduced when power saving is activated.
ble	Enables or disables Bluetooth functionality. Set to 1 to enable Bluetooth, making the GHv4 discoverable as a Bluetooth device (e.g., by your smartphone) after booting. Set to 0 to disable it. If Bluetooth is enabled, pressing the power button will disable it by automatically setting this parameter to 0 in the "override-general.txt" file and rebooting the device.
ble_name	Defines the name under which the GHv4 appears as a Bluetooth device.
ble_pass	Sets the password for Bluetooth pairing. Must be a 6-digit number (between 0 and 9). If not specified, the default password 000000 will be used. When trying to pair with the GHv4, your phone may request a 4-digit number. Ignore this and go ahead with the 6-digit number.
ble_pairing_sounds	Both when the GHv4 is waiting for Bluetooth pairing and when it was successfully paired, the board will play a confirmation sound if this parameter is set to 1.If it is set to 0, no Bluetooth pairing sounds will be played.
wifi	Controls the WiFi functionality of the GHv4. • 0: Disabled. • 1: Enables WiFi in Mode 1 (Connect to Saved WiFi Network). • 2: Enables WiFi in Mode 2 (Connect to New WiFi Network).

	• 3: Enables WiFi in Mode 3 (Create a Hotspot). Refer to the "WiFi Connectivity" section for detailed explanations of each mode.
wifi_ssid	Specifies the SSID (network name) of the WiFi network to which the GHv4 should connect when using WiFi Mode 1. This value should match the SSID of your desired network.
wifi_pass	Specifies the password of the WiFi network to which the GHv4 should connect when using WiFi Mode 1. Must match the password for the specified network SSID.
oled	Enables or disables the OLED display, if connected. Set to 1 to enable and 0 to disable.
oled_orientation	Sets the orientation of the content on the OLED display. Set to 0 for the default orientation, or 1 to rotate the display content by 180°.
oled_screensaver	Defines the time of inactivity in seconds before the screensaver is displayed on the OLED. Note that when the GHv4 enters Deep Sleep, the OLED display will turn off regardless of this setting.
accent_standby	Determines if accent LEDs remain lit when the blade is retracted. Set to 0 to turn them off and 1 to keep them lit. Note that accent LEDs will always turn off when the board enters Deep Sleep.
accent_battery_indicator	Controls whether accent LEDs display the battery status. • 0: Accent LEDs do not display battery status; they follow the standby and ignited accent profiles. • 1: Accent LEDs display battery status when the saber is not ignited. • 2: Accent LEDs display battery status both when the saber is ignited and not ignited.
accent_blade_matching	Controls whether to match the color of an RGB switch with the blade color when using a Pixel blade. Set to 0 to disable or 1 to enable. Requires an RGB switch with separate positive terminals for the LEDs and a common cathode. Connect the red LED to LED7, the green LED to LED8, and the blue LED to LED9.
charging_indicator	Controls the charging indicator sounds. 0: No charging indicator sound. • No charging indicator sound is played. • 1: Plays a sound when charging starts using legacy detection based on voltage jumps, defined by "charging_threshold" in millivolts. Requires that the board is not in deep sleep when connecting the charger. • 2: Plays a sound when charging starts, even when the board is in deep sleep. Also, plays a sound when the battery is fully charged or charging is interrupted. Designed to work with the Seedling v2 and requires a connection between the VBUS pad of the Seedling v2 and the DET. pad of the GHv4.
charging_threshold	Sets the required voltage jump in millivolts to trigger a charging sound when using the legacy detection method of the charging indicator feature.

charging_completed	Specifies the battery voltage in millivolts above which the battery is considered fully charged. This is relevant for the charging indicator feature and determines whether a sound indicating that charging was interrupted or a sound indicating that charging was completed is played when removing the charger.
battery_low	Sets the battery voltage in millivolts at which the voltage critical indicator sound is played. After reaching this voltage, the GHv4 will automatically enter deep sleep.
deep_sleep	Sets the time in seconds after which the board enters Deep Sleep if not used while powered.

WiFi Connectivity

The GHv4 introduces seamless WiFi connectivity, enabling wireless file transfers between a computer or a smartphone and the soundboard's SD card. This feature is designed to provide users with the utmost convenience in managing their lightsaber's content without the need for cumbersome cables. Access to the SD card is facilitated through a sophisticated file manager via a browser, enhancing user interaction by allowing not just file management but also remote control of the lightsaber. Notably, the GHv4 is equipped with an onboard antenna that is fully approved for WiFi use, eliminating the need for any external antennas. This integrated design ensures a sleek and efficient setup.

The WiFi functionality has been optimized for minimal power consumption, allowing users to enjoy the benefits of wireless connectivity without compromising the battery's runtime. WiFi remains enabled after waking the board up from deep sleep, or reinserting the battery, providing a consistent connection experience across reboots. The feature can be toggled directly using the Saber Editor, via the power button, or by modifying configuration files. Furthermore, the board can enter deep sleep when WiFi is enabled but the board is not connected to our web app, the Golden Harvest Commander. This preserves battery life during idle periods. Upon connecting to a network, the board announces its IP address via audio, allowing quick access via a browser. This announcement can be interrupted by briefly pressing the power button.

The GHv4 connects to 2.4 GHz WiFi only. To access it from your phone or computer, make sure your device is connected to the same 2.4 GHz network.

WiFi Connectivity Modes

The Golden Harvest 4 soundboard enhances its versatility with three distinct WiFi connectivity modes, allowing users to choose the optimal setup for their specific needs. Each mode is designed to provide flexibility and ease of access to the board's settings and content via WiFi.

Mode 1: Connect to Saved WiFi Network

In this mode, the GHv4 automatically connects to the WiFi network whose SSID and password are stored in the general.txt file or the override-general.txt file, with the latter taking precedence if both are present. Once connected, which will be indicated by a sound, the content of the SD card can be accessed through a web browser at ghv4.local or ghv4.mshome.net. If these URLs do not work, it may be necessary to directly enter the board's IP address, which could indicate that your device does not support mDNS. You can obtain the IP address by holding the power button for 2 seconds. A voice will then read out the IP address.

Mode 2: Connect to New WiFi Network

This mode initiates a hotspot named "GHv4" from the board itself. Users can connect to this hotspot using a smartphone or computer, which will then trigger a captive portal to display available WiFi networks. Users can select a network, enter the required password, and the details will be saved to the override-general.txt file. After rebooting, the board will connect to the selected network, and access to the SD card is available via ghv4.local or ghv4.mshome.net. If these URLs do not work, it may be necessary to directly enter the board's IP address, which could indicate that your device does not support mDNS. You can obtain the IP address by holding the power button for 2 seconds. A voice will then read out the IP address.

Mode 3: Create a Hotspot

In this mode, users must first create a hotspot named "GHv4" with the password "lightsaber" from their computer or smartphone. It is important to ensure that the hotspot name is exactly "GHv4," with "GH" capitalized and "v" as a lowercase letter. The GHv4 board will search for and connect to this network. Once connected, users can manage files and settings directly through their browser by navigating to via ghv4.local or ghv4.mshome.net. If these URLs do not work, it may be necessary to directly enter the board's IP address, which could indicate that your device does not support mDNS. You can obtain the IP address by holding the power button for 2 seconds. A voice will then read out the IP address.

WiFi Feature Control

The WiFi feature can be managed on the fly using the QuickAccess feature, the Saber Editor or manually by editing the config files. The parameters in the following table control the WiFi settings. They are stored in the general.txt file and the override-general.txt file.

Parameter	Value
wifi=0	0: Disables WiFi 1: Enables Mode 1 2: Enables Mode 2 3: Enables Mode 3
wifi_ssid=ssid	Specifies the SSID for the WiFi network, settable via captive portal in Mode 2 or manually
wifi_pass=password	Specifies the WiFi network password, settable as above
wifi_setup_edit_mode=1	0: Disables access to WiFi setup in the Saber Editor 1: Enables access to WiFi setup in the Saber Editor
wifi_setup_twist_speed=100	Controls the sensitivity for rotating the saber to switch WiFi modes in the Saber Editor

These settings offer comprehensive control over the WiFi capabilities of the GHv4, making it exceptionally user-friendly and adaptable to various user preferences and technical environments.

Bluetooth Connectivity

The Golden Harvest v4 introduces robust Bluetooth connectivity, enhancing the way users can remotely control their lightsaber. This feature is specifically designed to offer seamless interaction without the physical constraints of wires. Users can effortlessly adjust settings, change sound fonts, and command their lightsaber directly from a smartphone or other Bluetooth-enabled device through the ForceSync app, a dedicated platform designed to maximize the capabilities of the GHv4. It is currently available for Android:

- Android: https://play.google.com/store/apps/details?id=com.forcesync.saber.forcesync.lite&hl=en_US

In future, it will also support iOS. When Bluetooth is enabled (see below), the board will begin advertising and can automatically connect to the ForceSync app once discovered. If your phone prompts for a PIN, make sure to enter six digits. The default PIN is 000000 unless you changed it manually. Similar to its WiFi capabilities, the GHv4 incorporates an onboard antenna approved for Bluetooth use, which means no external antennas are necessary. The Bluetooth functionality is also designed to be energy-efficient, having a minimal impact on the saber's battery life, thus allowing enthusiasts to enjoy prolonged use without performance degradation.

Bluetooth remains active even after the board wakes up from deep sleep or after reinserting the battery, ensuring reliable wireless availability across power cycles. The board is also capable of entering deep sleep while Bluetooth is enabled, provided the board is not connected to the ForceSync app, allowing it to conserve power when not in use.

Bluetooth Feature Control

The Bluetooth feature can be managed on the fly using the QuickAccess feature, the Saber Editor or manually by editing the config files. The parameters in the following table control the Bluetooth connectivity. They are stored in the config files.

Parameter	Value
ble=0	0: Disables Bluetooth 1: Enables Bluetooth
ble_name=name	Defines the name under which the GHv4 appears as a Bluetooth device.
ble_pass=password	Sets the password for Bluetooth pairing. Must be a 6-digit number (each digit between 0 and 9). If not specified, the default password 000000 will be used. When trying to pair with the GHv4, your phone may request a 4-digit number. Ignore this and go ahead with the 6-digit number.
ble_setup_edit_mode=1	0: Disables access to Bluetooth setup in the Saber Editor 1: Enables access to Bluetooth setup in the Saber Editor
ble_setup_twist_speed=100	Controls the sensitivity for rotating the saber to enable or disable Bluetooth

After the Bluetooth connection was established, the Bluetooth feature can be disabled by pressing the power button, which sets the ble parameter to 0. To re-enable Bluetooth, you can either use the Saber Editor or manually adjust the ble parameter to the desired value in the general.txt file or the override-general.txt file.

QuickAccess of Wireless Features

QuickAccess is a convenient power-button shortcut system that allows you to toggle wireless features without using the Saber Editor or changing config files. To use it, simply hold the power button for a specific duration:

- **5 seconds:** Toggle Bluetooth.
- **7.5 seconds:** Toggle WiFi (Mode 1).
- **10 seconds:** Reboot the board.

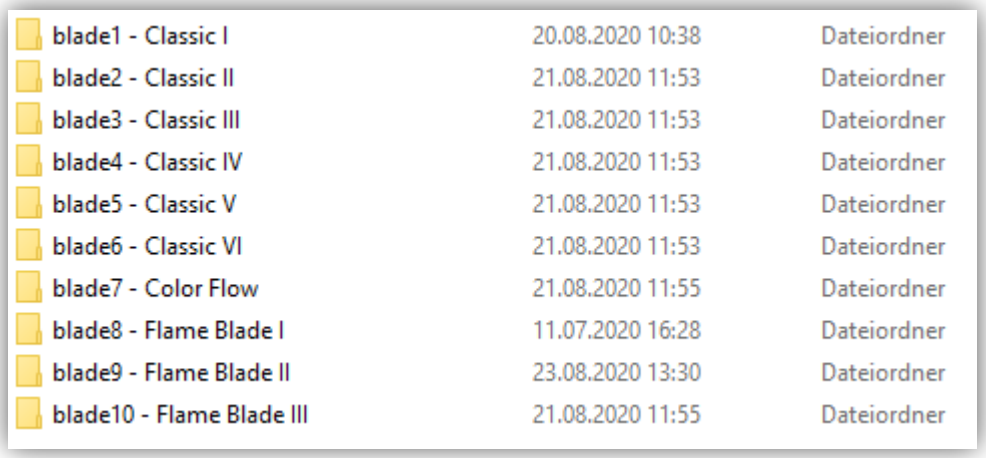
Each option is indicated by a sound.

Blade Styles

During the development of the Golden Harvest v4, we emphasized high customizability of light and sound effects. Our two main goals were to provide an immersive, realistic experience that transports you to another world, and to offer versatile configurability so you can create a unique style for your lightsaber. Whether you prefer simple, complex, or exotic effects, the extensive range of settings on the Golden Harvest v4 allows you to realize almost any effect you can imagine, and even invent new ones.

The Golden Harvest v4 supports more than 600 parameters for in-hilt setups and several thousand for Neopixel setups, which are customizable by manually editing the respective files or using our configurator. However, you don't need to change any parameters to enjoy your lightsaber - the board comes with a default set of values, ready to use out of the box.

You can store an unlimited number of blade styles on your SD card. Each style is represented by a folder labeled "blade[number]" within the "EffectFonts" folder. The numbering must start at 1 and be continuous. You may also add descriptions to each "blade[number]" folder, as long as the folder name begins with "blade[number]". An example of a possible naming convention for the blade folders is shown below.

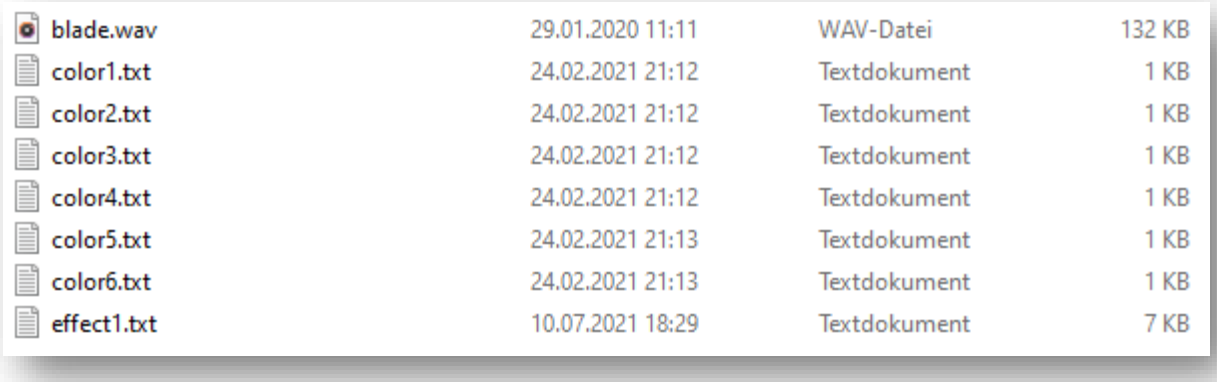


Each blade style defines specific light effects and colors for your saber. Since there are differences between in-hilt and Neopixel blade styles, their structures will be discussed separately.

In-Hilt Blade Styles

File Structure

In-hilt blade styles consist of an "effect1.txt" file and multiple "color[number].txt" files, as shown in the illustration below. The "effect1.txt" file defines the saber's light effects, while the "color[number].txt" files serve as color profiles that determine the blade's colors. You can change both the blade style and the color profile on the fly, even while the saber is turned on!



blade.wav	29.01.2020 11:11	WAV-Datei	132 KB
color1.txt	24.02.2021 21:12	Textdokument	1 KB
color2.txt	24.02.2021 21:12	Textdokument	1 KB
color3.txt	24.02.2021 21:12	Textdokument	1 KB
color4.txt	24.02.2021 21:12	Textdokument	1 KB
color5.txt	24.02.2021 21:13	Textdokument	1 KB
color6.txt	24.02.2021 21:13	Textdokument	1 KB
effect1.txt	10.07.2021 18:29	Textdokument	7 KB

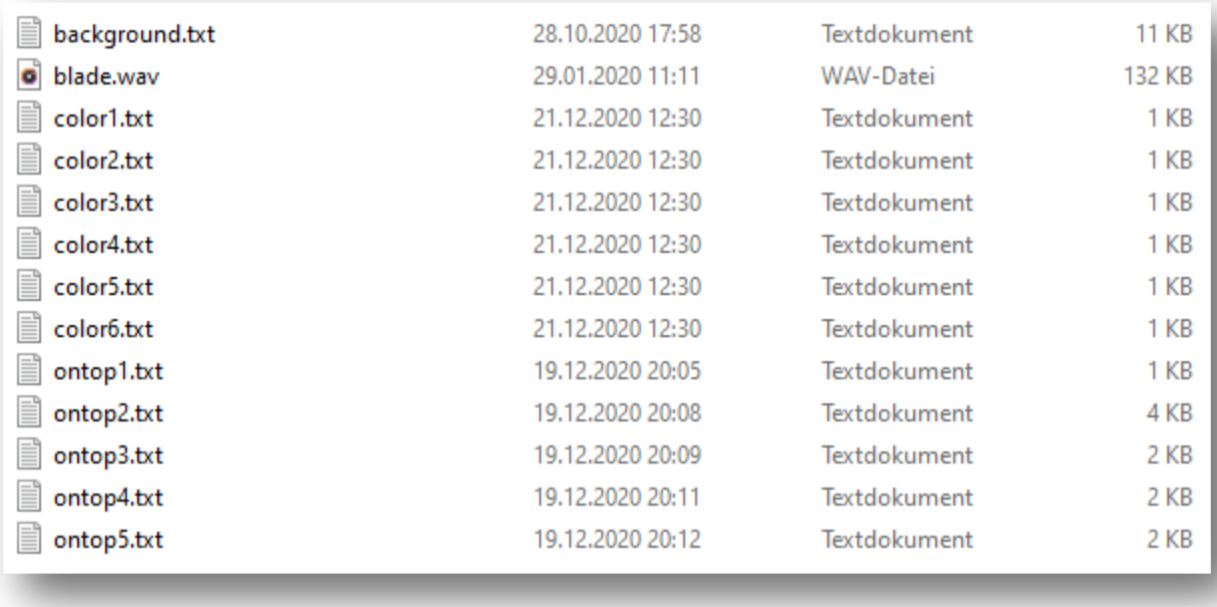
Parameters Overview

The in-hilt blade styles allow for extensive customization through various parameters. These parameters control aspects such as light effects and colors for different saber states, from ignition to force push. For detailed descriptions of each parameter, please refer to "Appendix A: In-Hilt Blade Style Parameters".

Neopixel Blade Styles

File Structure

Neopixel blade styles consist of a "background.txt" file, multiple "ontop[number].txt" files, and several "color[number].txt" files, as shown in the illustration below. The "background.txt" file defines the main background effect for the blade style, while the "ontop[number].txt" files layer additional effects on top of the background, allowing for variations within the same style. The "color[number].txt" files specify the colors the blade can display. Like with in-hilt setups, you can change the blade style, on top effects, and color profile on the fly, even while the saber is turned on!



background.txt	28.10.2020 17:58	Textdokument	11 KB
blade.wav	29.01.2020 11:11	WAV-Datei	132 KB
color1.txt	21.12.2020 12:30	Textdokument	1 KB
color2.txt	21.12.2020 12:30	Textdokument	1 KB
color3.txt	21.12.2020 12:30	Textdokument	1 KB
color4.txt	21.12.2020 12:30	Textdokument	1 KB
color5.txt	21.12.2020 12:30	Textdokument	1 KB
color6.txt	21.12.2020 12:30	Textdokument	1 KB
ontop1.txt	19.12.2020 20:05	Textdokument	1 KB
ontop2.txt	19.12.2020 20:08	Textdokument	4 KB
ontop3.txt	19.12.2020 20:09	Textdokument	2 KB
ontop4.txt	19.12.2020 20:11	Textdokument	2 KB
ontop5.txt	19.12.2020 20:12	Textdokument	2 KB

Parameters Overview

Neopixel blade styles offer extensive customization through a variety of parameters. These parameters control key aspects of the saber's light effects and colors for different states, including ignition, swing,

clash, and force push. Most parameters are specific to the chosen background effect and on top effect, determining how these effects appear and interact. In this section, we will briefly discuss a selection of parameters that are general and apply across different effects.

For a detailed explanation of all parameters, please refer to "Appendix B: Neopixel Blade Style Parameters".

Effect fading

Effect fading controls the transition of light effects back to the basic state after events like clashes. For example, when a clash is triggered, the clash effects are shown, but they must gradually transition back to the basic effects once the clash ends.

Each saber state has specific parameters to manage this fading, indicated by a prefix related to the state (e.g., “cl_” for clash). We will use the clash state as an example to explain these parameters, but the principles apply to other saber states as well.

Key Parameters:

- **cl_duration**: Defines how long clash effects are displayed, relative to the clash sound length. Ranges from 0 to 100, where 0 disables clash effects and 100 shows them for the entire duration of the clash sound. This allows you to create anything from short, sharp flashes to prolonged effects that simulate a destabilized blade.
- **cl_fade_out**: Controls the speed of the transition back to the basic effects, relative to the clash effect duration. Ranges from 0 to 100. A value of 0 means no smooth transition (abrupt change to the basic effect), while 100 provides the smoothest fade-out, blending the clash effect back into the basic effect. For example, if your blade flashes white on a clash, this parameter can smoothly transition the blade color from white to red as the clash effect fades.
- **cl_fade_out_type**: Determines the style of the fade-out. Setting it to 0 offers the most customization, such as localized clash effects. Other values provide different styles - see the "Appendix B" for details.

When cl_fade_out_type is set to 0, only a specific section of the blade displays the clash effect, while the rest shows the basic effect. This is also known as a localized clash effect. This section is defined by the following parameters:

- **cl_size**: Number of LEDs showing the clash effect.
- **cl_min_position** and **cl_max_position**: The random position range on the blade where the clash effect appears.
- **cl_smooth**: The smoothness of the transition between the clash effect section and the rest of the blade.

These parameters give you complete control over how effects are displayed and transition, allowing for unique and dynamic visual effects.

Responsive Lockup

The lockup effect can be either randomly positioned or adjusted in real-time using gestures. The responsive lockup feature allows you to control the location of the lockup by tilting, twisting, or both. You can choose which gesture(s) to use for this effect.

The parameters in the table below, which are stored in the "background.txt" file, control the responsive lockup feature.

PARAMETER	DESCRIPTION
lu_responsive	The values below determine how the responsive lockup feature behaves: • 0: Responsive lockup is disabled; the lockup position is chosen randomly. • 1: The lockup position changes by tilting the saber • 2: The lockup position changes by twisting the saber. • 3: The lockup position changes by both tilting and twisting the saber.
lu_tilt_speed	Adjusts the sensitivity of tilting the saber for lockup control. Higher values increase sensitivity (small tilts have a bigger effect), while lower values decrease sensitivity.
lu_twist_speed	Adjusts the sensitivity of twisting the saber for lockup control. Higher values increase sensitivity (small twists have a bigger effect), while lower values decrease sensitivity.

Responsive Drag

The responsive drag feature allows the size of the illuminated tip area to change by twisting the saber. The parameters in the table below, which are stored in the "background.txt" file, control the responsive drag feature:

PARAMETER	DESCRIPTION
dr_responsive	The values below determine how the responsive drag feature behaves: • 0: Responsive drag is disabled; the size of the illuminated tip area is fixed. • 1: The size of the illuminated tip area changes by twisting the saber.
dr_twist_speed	Adjusts the sensitivity of twisting the saber for drag control. Higher values increase sensitivity (smaller twists have a bigger effect), while lower values decrease sensitivity.

Responsive Melt

The responsive melt feature allows you to control the area of the tip simulating the melting effect and its color by tilting, twisting, or both. You can choose which gesture(s) to use for this effect.

The parameters in the table below, which are stored in the "background.txt" file, control the responsive melt feature.

PARAMETER	DESCRIPTION
me_responsive	The values below determine how the responsive melt feature behaves: • 0: Responsive melt is disabled; the area of the tip simulating the melting effect and its color are fixed. • 1: The melting area and its color change by tilting the saber. • 2: The melting area and its color change by twisting the saber. • 3: The melting area and its color change by both tilting and twisting the saber.
me_tilt_speed	Adjusts the sensitivity of tilting the saber for melt control. Higher values increase sensitivity (smaller tilts have a bigger effect), while lower values decrease sensitivity.
me_twist_speed	Adjusts the sensitivity of twisting the saber for melt control. Higher values increase sensitivity (smaller twists have a bigger effect), while lower values decrease sensitivity.

Responsive Blaster Block

The responsive blaster block feature allows you to control the position of the blaster block effect in real time using gestures. You can adjust the location by tilting, twisting, or both, instead of it being chosen randomly.

The parameters in the table below, which are stored in the "background.txt" file, control the responsive blaster block feature.

PARAMETER	DESCRIPTION
bb_responsive	The values below determine how the responsive blaster block feature behaves: • 0: Responsive blaster block is disabled; the position is chosen randomly. • 1: The position of the blaster block changes by tilting the saber. • 2: The position of the blaster block changes by twisting the saber. • 3: The position of the blaster block changes by both tilting and twisting the saber.
bb_tilt_speed	Adjusts the sensitivity of tilting the saber for blaster block control. Higher values increase sensitivity (smaller tilts have a bigger effect), while lower values decrease sensitivity.
bb_twist_speed	Adjusts the sensitivity of twisting the saber for blaster block control. Higher values increase sensitivity (smaller twists have a bigger effect), while lower values decrease sensitivity.

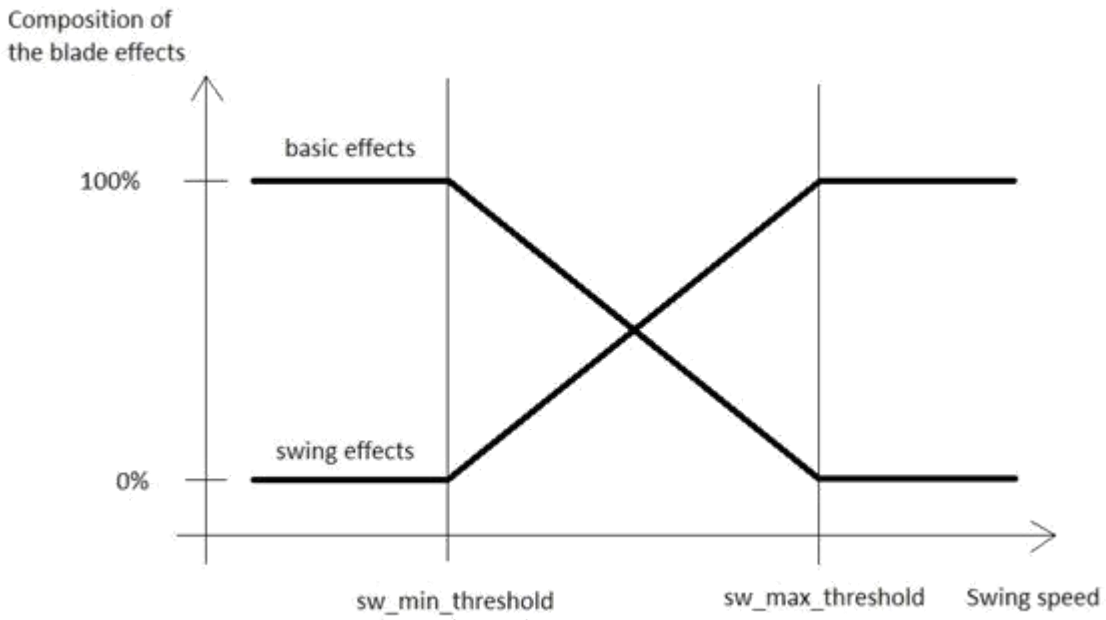
Responsive Swing Effects

The blade can dynamically change its swing color and light effects in real-time based on the swing speed. As the blade swings faster, the swing color and effects become more pronounced, blending with the basic blade color and effects. The mixing ratio shifts towards the swing effects as swing speed increases.

The parameters in the table below, which are stored in the "background.txt" file, control the responsive swing effects feature.

PARAMETER	DESCRIPTION
sw_motion_sensitive	The values below determine how the responsive swing effects feature behaves: • 0: Responsive swing color and light effects are disabled; the color and light effects of the basic state are used when swinging the saber. • 1: Responsive swing color and light effects are enabled.
sw_min_threshold	Sets the minimum swing speed at which the blending begins. Below this speed, only the basic blade color and effects are displayed.
sw_max_threshold	Sets the swing speed at which the blending reaches its maximum. Above this speed, only the swing color and light effects are displayed.

The blending process is shown in the illustration below. When the swing speed is below the value defined by "sw_min_threshold", only the basic blade color and effects are visible. As the swing speed increases, the share of the swing color and effects in the blend increases. Once the swing speed surpasses the value set by "sw_max_threshold", the blade fully transitions to displaying only the swing color and effects.



Sauron Mode

The Sauron mode creates a dramatic visual effect resembling the Eye of Sauron and can be used with the Flame Blade, Wave, and Color Flow effects. For example, with the Flame Blade effect, a flame extends from the center of the blade towards both the tip and the bottom, or vice versa. Additionally, a dark spot is located in the middle of the blade, simulating Sauron's eye. This mode is controlled by four parameters in the "background.txt" file, which are detailed in the table below.

PARAMETER	DESCRIPTION
flame_blade_sauron	Defines the type of Sauron mode used: • 0: Sauron mode disabled. • 1: Flames originate at both the bottom and tip of the blade without rescaling*. • 2: Flames originate in the middle of the blade without rescaling*. • 3: Flames originate at both the bottom and tip of the blade with rescaling*. • 4: Flames originate in the middle of the blade with rescaling*. • 5: Flames originate from the top of the blade.
flame_blade_sauron_size	Defines the size of the dark spot representing the Eye of Sauron.
flame_blade_sauron_spark	Controls the instability of the dark spot simulating the Eye of Sauron. Higher values increase the instability.
flame_blade_responsive	Defines the responsiveness of the Sauron mode: • 0-3: No responsiveness. Check the "Appendix B" for details on what these values determine. • 4: Dark spot moves by tilting the saber. • 5: Dark spot moves by twisting the saber. • 6: Dark spot moves by both tilting and twisting the saber.

* In Sauron mode, the flame effects are mapped to the upper and lower halves of the blade, with a dark spot representing the Eye of Sauron in the center. This reduces the range over which the effects are displayed to approximately half the original blade length (excluding the dark spot). Applying

rescaling means that the original blade effects are adjusted to fit this reduced range, preserving their appearance and proportions over the shorter distance. Not applying rescaling means that the original blade effects are displayed without adjustment, which may cause them to look stretched or compressed. This can result in a more intense or exaggerated visual effect.

Preon and Postoff Light Effects

The preon and postoff light effects are synchronized with the audio levels of the corresponding sounds, creating a highly immersive experience. These effects are controlled by four parameters in the "background.txt" file, which are shown in the table below.

PARAMETER	DESCRIPTION
preon_effect_intensity	Defines the intensity of the preon light effects. Higher values result in more pronounced effects.
preon_use_basic_effects	Determines the color and effect source for preon light effects: • 0: Uses custom ignition effects. • 1: Uses the same effects as in the basic state (after ignition). This setting only applies when custom ignition effects are used.
postoff_effect_intensity	Defines the intensity of the postoff light effects. Higher values result in more pronounced effects.
postoff_effect_intensity	Defines the intensity of the postoff light effects. Higher values result in more pronounced effects.

Blade Style Canvas

The Golden Harvest v4 introduces "Blade Style Canvas", that revolutionizes how the GHv3 treated blade style customization. Traditional blade styles are typically defined by a set of parameter files that detail background effects, on top effects, and color profiles. While this method offers substantial freedom in design, it naturally has limitations in terms of visual complexity and detail.

Blade Style Canvas elevates the potential for customization by allowing blade styles to be defined through .bmp files that contain every single frame of the blade's animation. This means that virtually any visual concept or design can now be realized on your lightsaber blade, from intricately detailed patterns to dynamic, animated sequences.

How Blade Style Canvas Works

File Structure

Each state of the saber - such as ignition, basic operation, clash, stab, and others - has a dedicated .bmp file that contains the animation for that specific state. The height of each .bmp file matches the number of pixels in the blade, and the width corresponds to the number of frames in the animation. In addition to that, there is a canvas.txt file containing a parameter defining the delay between two subsequent frames of the animation, which determines the animation speed.

State-Specific Animations

For every saber state, there is a primary .bmp file named [state].bmp, which displays the main animation for that state. For instance, during a clash, the clash.bmp file would be displayed.

Transition Animations

In addition to the main state files, there are also transition files named [state]_fade_out.bmp for each state except the basic state. These files are used to smoothly transition from the state effect back to the basic blade state. The clash_fade_out.bmp, for example, would be used to control how the clash

effects transition back to the normal blade appearance. The transition files serve as fade out masks and are grayscale .bmp files. The grayscale value of each pixel in these masks determines the blend between the current state's effects and the basic blade effects. A grayscale value of 0 indicates that the state effect is completely blocked, displaying only the basic effects, while a value of 255 allows the state effect to fully appear, covering the basic effects.

Special Treatment of Ignition and Retraction

Just like standard blade styles, the animation speed for ignition and retraction can be synchronized with the duration of their respective sound effects or set to a custom speed. This flexibility is controlled by the poweron and poweroff parameters in the settings.txt file of each sound font. For all other saber states, the animation speed is consistently maintained at a fixed value. This speed is determined by the frame_delay parameter found in the canvas.txt file associated with each blade style.

PARAMETER	DESCRIPTION
frame_delay	Specifies the delay between two subsequent frames in milliseconds, defining the animation speed

Note: The Blade Style Canvas feature is currently in beta testing and will not be available in the initial release of the GHv4. It is planned to be included in a future firmware update.

Blade Style Sounds and Editing

When you change the blade style, the blade.wav sound within it plays as an identification sound. Default blade.wav sounds are provided, but you can replace them with your custom sounds, as long as they meet the requirements specified in the "Sound Fonts" section.

All of the files included in a Blade Style are accessible on your microSD card and can be edited with any common text editor, or standard graphics editing tool for the .bmp files, regardless of your operating system. This allows you to manually adjust the parameters or create an animation frame by frame for a truly customized experience, which is rather for purists who enjoy hands-on control though.

Alternatively, you can use our innovative configurator to design your custom effects! After months of hard work and dedication, we are proud to offer a tool that provides a remarkable experience, letting your creativity shine. The configurator supports Neopixel setups and allows you to adjust parameters graphically via input fields, multiple-choice options, or sliders. One of its standout features is a real-time preview that shows an animation of your saber's appearance with the chosen parameters.

We recommend using the configurator on a desktop computer running Windows 10 or a more recent version for the best performance. Some laptops may not fully support the configurator, especially if they lack a dedicated graphics card.

The configurator is accessible via the link below.

SaberTec Configurator

Color menu

To access the color menu, hold the power button and then tap it. Once in the menu, you can cycle through all preset color profiles in various ways, as defined by the parameters in the table below. A gear sound indicates each switch to a new color profile.

PARAMETER	DESCRIPTION
color_menu	• 0: The color menu is disabled. • 1: Cycle through color profiles by twisting the saber. Clockwise for the next profile, counterclockwise for the previous. • 2: Cycle through color profiles by twisting the saber in any direction while pointing it up or down. Upwards orientation goes to the next profile; downwards goes to the previous. This allows cycling without twisting too far in one direction. • 3: Cycle through color profiles by tapping the power button. Pointing the saber upwards goes to the next profile; pointing it downwards goes to the previous.
color_menu_twist_speed	Adjusts the sensitivity of twisting the saber in the color menu. Lower values increase sensitivity, making it easier to switch profiles with smaller twists.

For two momentary buttons setups, you can also change the color profile on the fly by tapping the power button while holding the auxiliary button.

Tutorial videos:

- One button
- Two buttons

Effect menu

There are slight differences in navigating the effect menu depending on whether you have a one button or two buttons setup. Let's explore each configuration separately.

Two momentary buttons

To access the effect menu with two momentary buttons, hold the power button and then hold it again while pointing the saber upwards. In the effect menu, you can cycle through blade styles and on top effects. To navigate through on top effects, tap the auxiliary button while pointing the saber upwards to go to the next, and downwards to go to the previous one.

There are various ways to cycle through blade styles as defined by the parameters detailed in the table below.

PARAMETER	DESCRIPTION
effect_menu	• 0: Effect menu is disabled. • 1: Slowly twist the saber to cycle through blade styles (clockwise for next, counterclockwise for previous). • 2: Point the saber up or down while slowly twisting to change blade styles. Upwards for next, downwards for previous. Twist direction doesn't matter. • 3: Tap the power button to change blade styles. Point upwards for next, downwards for previous.

effect_menu_twist_speed	Adjusts the sensitivity of twisting the saber in the effect menu. Lower values increase sensitivity, making it easier to switch blade styles with smaller twists.

The different effect menu modes for two buttons setups are demonstrated in the following video: https://www.youtube.com/watch?v=pR5Ptf139zc

One momentary button

To access the effect menu with one button, hold the button and then hold it again while pointing the saber upwards. In the effect menu, you can navigate through blade styles and on top effects. To navigate through on top effects, tap the button while pointing the saber upwards to go to the next, and downwards to go to the previous one.

There are various ways to cycle through blade styles as defined by the parameters detailed in the table below.

PARAMETER	DESCRIPTION
effect_menu	• 0: Effect menu is disabled. • 1: Slowly twist to change blade styles (clockwise for next, counterclockwise for previous). • 2: Point the saber up or down while slowly twisting to change blade styles. Upwards for next, downwards for previous. Twist direction doesn't matter. • 3: Tap the power button to change blade styles. Point upwards for next, downwards for previous.
effect_menu_twist_speed	Adjusts the sensitivity of twisting the saber in the effect menu. Lower values increase sensitivity, making it easier to switch blade styles with smaller twists.

The different effect menu modes for one button setups are described in the following video: https://www.youtube.com/watch?v=pR5Ptf139zc

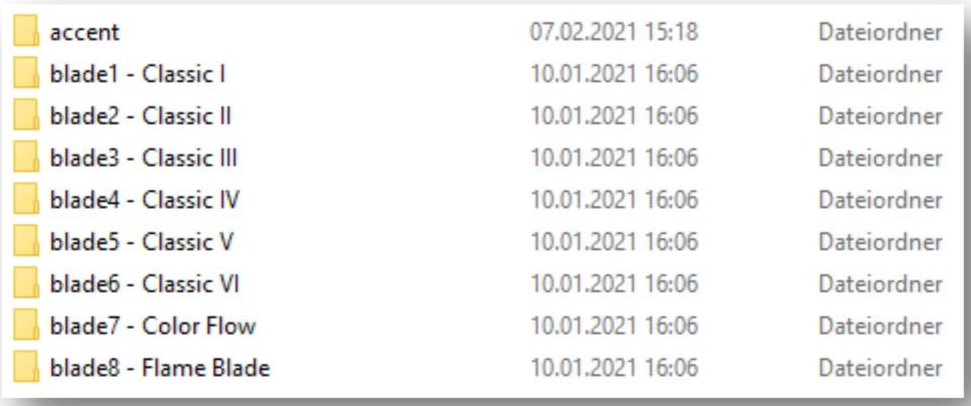
Accent Styles

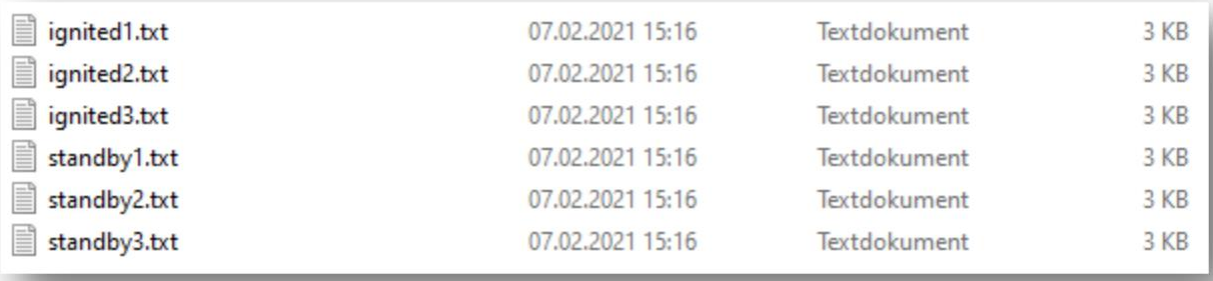
Just as blade styles determine your blade's light effects, accent styles define the light effects for your accent LEDs. This applies to both standard and Neopixel accent LEDs, so we'll simply use the term "accent LEDs".

Each slot folder on the SD card has an "accent" folder that contains a range of accent profiles, which can be selected on the fly and are categorized into two types:

- **Standby Accent Styles:** Control the behavior of the accent LEDs when the blade is not ignited. These files are named "standby[number].txt".
- **Ignited Accent Styles:** Control the behavior of the accent LEDs when the blade is ignited. These files are named "ignited[number].txt".

The screenshots below display the structure of the "accent" folder inside each slot folder, and the content it contains.





Accent Style Structure

Both standby and ignited accent styles share the same parameters and are organized into three main sections:

- **Accent LEDs 1 - 4**
- **Accent LEDs 5 - 8**
- **Sequencer Section**

The Golden Harvest v4 supports up to six dedicated standard accent LEDs (pads LED5, LED6, LED7, LED8, LED9, and LED10) and up to 32 Neopixel accent LEDs.

For **standard accent LEDs**, the mapping works as follows:

- **Accent LEDs 1 - 4** in the profiles are linked to pads LED5, LED6, LED7, and LED8.
- **Accent LEDs 5 and 6** are connected to pads LED9 and LED10.

For **Neopixel accent LEDs**, the sections work similarly:

- **Accent LEDs 1 - 4** represent the first four Neopixel accent LEDs.
- **Accent LEDs 5 - 8** represent the next four Neopixel accent LEDs.

Let's begin by exploring the parameters for accent LEDs 1 - 4, all of which use the prefix "a1_". The details for this section are illustrated in the image below.

```
#SaberTec Customizer - Accent File
#This config file contains all the parameters defining the behaviour of your accent LEDs.


#_____________________________
#
#Accent
#_____________________________


a1_color1=255,0,0
a1_color2=0,255,0
a1_color3=0,0,255
a1_color4=255,255,0

a1_led=1023,1023,1023,1023

a1_flicker_min_intensity=0,0,0,0
a1_flicker_max_intensity=0,512,512,0
a1_flicker_min_delay=1,1,1,1
a1_flicker_max_delay=1,10,1,1
a1_flicker_min_duration=1,1,1,1
a1_flicker_max_duration=1,1,1,1
a1_flicker_period=0,0,0,0
a1_flicker_phase_shift=0,0,0,0
a1_flicker_type=0,0,0,0

a1_pulse_shape=0,0,0,0
a1_pulse_min_intensity=1023,0,0,1023
a1_pulse_max_intensity=1023,0,0,1023
a1_pulse_intensity_change_period=0,0,0,0
a1_pulse_intensity_phase_shift=0,0,0,0
a1_pulse_min_period=50,0,0,20
a1_pulse_max_period=50,0,0,20
a1_pulse_phase_shift=0,0,0,0
a1_pulse_modulation_period=0,0,0,0
a1_pulse_modulation_phase_shift=0,0,0,0
```

PARAMETER	**DESCRIPTION**
a1_color1	This parameter is only active for Neopixel accent LEDs. It defines the color of the first of the up to 8 Neopixel accent LEDs. *Example:* *a1_color1=255,0,0* *In this case, the color of the first of the up to 8 Neopixel accent LEDs is red.*
a1_color2	This parameter is only active for Neopixel accent LEDs. It defines the color of the second of the up to 8 Neopixel accent LEDs. *Example:* *a1_color2=0,255,0*

	In this case, the color of the second of the up to 8 Neopixel accent LEDs is green.
a1_color3	This parameter is only active for Neopixel accent LEDs. It defines the color of the third of the up to 8 Neopixel accent LEDs. *Example:* *a1_color3=0,0,255* *In this case, the color of the third of the up to 8 Neopixel accent LEDs is blue.*
a1_color4	This parameter is only active for Neopixel accent LEDs. It defines the color of the fourth of the up to 8 Neopixel accent LEDs. *Example:* *a1_color4=255,255,255* *In this case, the color of the fourth of the up to 8 Neopixel accent LEDs is white.*
a1_led	Defines the intensity of the accent LEDs 1 - 4 both for standard accent LEDs and Neopixel accent LEDs. This parameter takes values between 0 and 1023 for each of the four different LED channels. *Example:* *a1_led=1023,1023,1023,1023* *In this case, the accent LED 1 has an intensity of 1023, the accent LED 2 has an intensity of 1023, the accent LED 3 has an intensity of 1023 and the accent LED 4 has an intensity of 1023.*

The parameters below "a1_led" describe the flicker and pulse effect each of the accent LEDs 1 - 4 displays. These parameters take four values each corresponding to the four different accent LED channels. They are the same parameters as in the blade styles for in-hilt setups. You can find a detailed description in the blade styles section.

Now, let’s come to the section for the accent LEDs 5 - 8 which is shown in the picture below. All parameters that belong to this section have the prefix "a2_".

```
a2_color5=255,255,0
a2_color6=0,0,255
a2_color7=0,255,0
a2_color8=255,0,0

a2_led=1023,1023,1023,1023

a2_flicker_min_intensity=512,512,512,0
a2_flicker_max_intensity=512,512,512,0
a2_flicker_min_delay=1,1,1,1
a2_flicker_max_delay=50,50,50,1
a2_flicker_min_duration=1,1,1,1
a2_flicker_max_duration=10,10,10,1
a2_flicker_period=0,0,0,0
a2_flicker_phase_shift=0,0,0,0
a2_flicker_type=0,0,0,0

a2_pulse_shape=0,0,0,0
a2_pulse_min_intensity=0,0,0,1023
a2_pulse_max_intensity=0,0,0,1023
a2_pulse_intensity_change_period=0,0,0,0
a2_pulse_intensity_phase_shift=0,0,0,0
a2_pulse_min_period=0,0,0,10
a2_pulse_max_period=0,0,0,50
a2_pulse_phase_shift=0,0,0,0
a2_pulse_modulation_period=0,0,0,100
a2_pulse_modulation_phase_shift=0,0,0,0
```

PARAMETER	DESCRIPTION
a2_color1	This parameter is only active for Neopixel accent LEDs. It defines the color of the fifth Neopixel accent LED. *Example:* *a2_color1=255,0,0* *In this case, the color of the fifth Neopixel accent LED is red.*
a2_color2	This parameter is only active for Neopixel accent LEDs. It defines the color of the sixth Neopixel accent LED. *Example:* *a2_color2=0,255,0* *In this case, the color of the sixth Neopixel accent LED is green.*
a2_color3	This parameter is only active for Neopixel accent LEDs. It defines the color of the seventh Neopixel accent LED. *Example:* *a2_color3=0,0,255*

	In this case, the color of the seventh Neopixel accent LED is blue.
a2_color4	This parameter is only active for Neopixel accent LEDs. It defines the color of the eigth Neopixel accent LED. *Example:* *a2_color4=255,255,255* *In this case, the color of the eigth Neopixel accent LED is white.*
a2_led	Defines the intensity of the accent LEDs 5 - 8 both for standard accent LEDs and Neopixel accent LEDs. This parameter takes values between 0 and 1023 for each of the four different LED channels. *Example:* *a2_led=1023,1023,1023,1023* *In this case, the accent LED 5 has an intensity of 1023, the accent LED 6 has an intensity of 1023, the accent LED 7 has an intensity of 1023 and the accent LED 8 has an intensity of 1023.*

The parameters below "a2_led" describe the flicker and pulse effect each of the accent LEDs 5 - 8 displays. These parameters take four values each corresponding to the four different accent LED channels. They are the same parameters as in the blade styles for in-hilt setups. You can find a detailed description in the blade styles section.

Now, let's come to the sequencer section containing all the parameters needed in order to customize the accent LED sequencer. It is shown in the picture below. All parameters that belong to this section have the prefix "ac_".

```
ac_mirror_blade=0,0,0,0,0,0,0,0

ac_sequence1=1,0,1,0,1,0,1,0
ac_sequence2=0,1,0,1,0,1,0,1
ac_sequence3=1,0,1,0,1,0,1,0
ac_sequence4=0,1,0,1,0,1,0,1
ac_sequence5=1,0,1,0,1,0,1,0
ac_sequence6=0,1,0,1,0,1,0,1
ac_sequence7=1,0,1,0,1,0,1,0
ac_sequence8=0,1,0,1,0,1,0,1
ac_sequence9=1,0,1,0,1,0,1,0
ac_sequence10=0,1,0,1,0,1,0,1
ac_sequence11=1,0,1,0,1,0,1,0
ac_sequence12=0,1,0,1,0,1,0,1
ac_sequence13=1,0,1,0,1,0,1,0
ac_sequence14=0,1,0,1,0,1,0,1
ac_sequence15=1,0,1,0,1,0,1,0
ac_sequence16=0,1,0,1,0,1,0,1

ac_on_time1=500,250,250,250
ac_on_time2=250,100,100,100
ac_on_time3=100,100,100,100
ac_on_time4=100,100,100,100

ac_break_time1=0,0,0,0
ac_break_time2=0,0,0,0
ac_break_time3=0,0,0,0
ac_break_time4=0,0,0,0
```

PARAMETER	DESCRIPTION
ac_mirror_blade	Defines if the Neopixel accents mirror the blade. The parameter takes 8 values corresponding to the different Neopixel accents. A value of 1 enables the blade mirror feature and a value of 0 disables it.
ac_sequence1 - ac_sequence16	These parameters define the actual sequence in which the accent LEDs 1 - 8 are turned on and off. Therefore, these parameters take 8 values each. A value of 1 means that the corresponding accent LED is turned on and a value of 0 means that the corresponding accent LED is turned off. The sequences have 16 different steps and each is described by one of these 16 parameters. *Example:* *ac_sequence1=1,0,0,0,0,0,0,0* *In this case, the first of the 16 steps of the sequence turns on only accent LED 1.*

ac_on_time1 - ac_on_time4	These parameters define the duration of each of the steps defined by the parameters "ac_sequence1" to "ac_sequence16". The duration of the steps 1 - 4 of the sequence is given by the four values the parameter "ac_on_time1" takes, the duration of the steps 5 - 8 of the sequence is given by the four values the parameter "ac_on_time2" takes, the duration of the steps 9 - 12 of the sequence is given by the four values the parameter "ac_on_time3" takes and the duration of the steps 13 - 16 of the sequence is given by the four values the parameter "ac_on_time4" takes. *Example:* *ac_on_time1=500,250,250,250* *In this case, the first step of the sequence is displayed for a time of 500, the second step of the sequence is displayed for a time of 250, the third step of the sequence is displayed for a time of 250 and the fourth step of the sequence is displayed for a time of 250.*
ac_break_time1 - ac_break_time4	These parameters define the break between of each of the steps defined by the parameters "ac_sequence1" to "ac_sequence16". The break after the steps 1 - 4 of the sequence is given by the four values the parameter "ac_break_time1" takes, the break after the steps 5 - 8 of the sequence is given by the four values the parameter "ac_break_time2" takes, the break after the steps 9 - 12 of the sequence is given by the four values the parameter "ac_break_time3" takes and the break of the steps 13 - 16 of the sequence is given by the four values the parameter "ac_break_time4" takes. *Example:* *ac_break_time1=0,0,0,0* *In this case, the break after the first step of the sequence is endures for a time of 0, the break after the second step of the sequence is endures for a time of 0, the break after the fourth step of the sequence is endures for a time of 0 and the break after the fourth step of the sequence is endures for a time of 0.*

Accent Style Canvas

An alternative to the standard structure of standby and ignited accent profiles is the Accent Style Canvas feature. This feature is similar to the Blade Style Canvas and allows you to create animations for the accent LEDs frame by frame, offering even more detailed customization.

Note: Like the Blade Style Canvas feature, the Accent Style Canvas feature is currently in beta testing and will not be available in the initial release of the GHv4. It is planned to be included in a future firmware update.

Blade Mirror Feature for Neopixel Accents

It is possible to customize Neopixel accents so that they mirror the blade. The parameter "ac_mirror_blade" in the accent profiles defines this feature and is described in the following.

PARAMETER	DESCRIPTION
ac_mirror_blade	The parameter takes 8 values corresponding to first 8 Neopixel accents. A value of 1 enables the blade mirror feature and a value of 0 disables it. *Example:* *ac_mirror_blade=1,0,0,1,0,0,0,1* *In this case, the Neopixel accents 1, 4 and 8 are set to mirror the blade. The other Neopixel accents show the light effects that are defined in the accent profiles.*

Blade Matching RGB Switch LED for Neopixel

It is possible to match the color of an RGB switch LED with the blade color for Neopixel. This feature is controlled by the parameter "accent_blade_matching" in the "general.txt". In the following, this parameter is described.

PARAMETER	DESCRIPTION
accent_blade_matching	• 0: The blade matching feature is disabled. • 1: The blade matching feature is enabled. Connect the RGB switch LED to LED7, LED8 and LED9.

Accent Menu

The accent menu can be trigerred by holding the power button and then holding it again. Point the saber downwards while doing this. In the accent menu, you can cycle through the ignited accent profiles and the standby accent profiles. When entering the accent menu, a sound will be played notifying you that you can cycle through your ignited accent profiles. To go to the next ignited accent profile, tap the power button while pointing the saber upwards. To go to the previous ignited accent profile, tap the power button while pointing the saber downwards. If you want to select an ignited accent profile, hold the power button. This will save your selection and after that, you can cycle through the standby accent profiles. If you want to go to the next standby accent profile, tap the power button while pointing the saber upwards. If you want to go to the previous standby accent profile, tap the power button while pointing the saber downwards. If the parameter "accent_standby=0" is set in the "general.txt" the standby accent submenu will be skipped as the accent LEDs are turned off when the board is in standby. The accent menu can be controlled by the parameter "accent_menu". It is described in the following.

PARAMETER	DESCRIPTION
accent_menu	• 0: The accent menu is disabled. When trying to trigger the accent menu, the effect menu will be activated. • 1: The accent menu is enabled.

Accent Battery Indicator

It is possible to let the accent LEDs display the battery status. The lower the remaining charge of the battery the faster the accent LEDs pulse. If the battery is fully charged, the accent LEDs are turned on constantly. This feature is controlled by the parameter "accent_battery_indicator" in the general.txt and is described in the following.

PARAMETER	DESCRIPTION
accent_battery_indicator	• 0: The accent LEDs don't display the battery status, but behave as defined in the standby accent profile and in the ignited accent profile. • 1: The accent LEDs display the battery status when the saber is not ignited. • 2: The accent LEDs display the battery status both when the saber is not ignited and when it is ignited.

Battery Indicator

The Golden Harvest v4 features a battery indicator that can be triggered before the saber is ignited. This feature performs a measurement of the battery voltage and tells you acoustically how much charge is left. If you're using a Neopixel blade, this information is also displayed via the blade. Hereby, the currently active blade effect is used. To get a more reliable estimation of how much charge is left, it is necessary to perform a battery calibration when the battery is fully charged. This needs to be done only a single time as the board will save the calibration.

Charging indicator

The Golden Harvest v4 features two approaches for monitoring and indicating charging status: a legacy method and a new, more accurate solution. Below is an overview of both systems.

Legacy Charging Indicator

The legacy method allows for a sound to be played when a charger is connected to the Seedling v2 or to an RCP (with a 2-pin configuration, ensuring the board remains connected to the battery during charging). To activate this feature, the board should not be in deep sleep when the charger is connected. This method is controlled by the parameters described in the table at the end of the next section. They are stored in the "general.txt" file.

PARAMETER	DESCRIPTION
charging_indicator	• 0: No charging indicator sound is played. • 1: A charging indicator sound is played.
charging_threshold	Specifies the sensitivity of the detection of inserting a charger. The smaller the value the more sensitive it is.

Enhanced Charging Indicator

The Golden Harvest v4 introduces a refined approach to monitoring and indicating charging status, significantly enhancing the precision over the legay charging indicator. The latter is based on detecting any rise in voltage to trigger a charging sound, which can sometimes result in false triggers due to non-charging related voltage fluctuations. The new approach adopts a more error-safe solution for detecting the charging status of the battery using the Det. pad on the GHv4. Sounds indicating that the charging started, was interrupted or completed are played under the following conditions:

- **Charger Connection:** When the Det. pad is grounded, the GHv4 recognizes that the charger is connected and the battery is charging.
- **Charger Removal:** If the Det. pad is no longer grounded after having been grounded, the GHv4 detects the removal of the charger. At this point, the soundboard will respond differently based on the battery's charge status:
    - **Interrupted Charging:** If the battery is not yet fully charged, a sound indicating that the charging was interrupted will play.
    - **Completed Charging:** If the battery is fully charged, a sound indicating that the charging was completed will play.

This enhanced feature is specifically designed for use with the Seedling v2, with its VBUS pad having to be connected to the Det. pad on the GHv4, providing the precise and necessary information to accurately manage and indicate charging states. This system not only improves the reliability of charging status notifications but also enhances user interaction by providing clear auditory signals that reflect the actual charging activities. The charging indicator feature is controlled by the parameters described in the table below, which are stored in the "general.txt file".

PARAMETER	DESCRIPTION
charging_indicator	• 0: Disabled • 1: Enabled, using the legacy charging indicator • 2: Enabled, using the enhanced charging indicator
charging_threshold	Specifies the sensitivity of the detection of inserting a charger when the legacy charging indicator is used. The smaller the value the more sensitive it is.
charging_completed	Determines the actions of the board upon charger removal based on the battery voltage, measured in millivolts. If the battery voltage exceeds the value set for this parameter, the board will play a sound indicating that charging has been completed. Conversely, if the battery voltage is below this value, the board will play a sound signaling that the charging was interrupted. This parameter is only used by the enhanced charging indicator.

Configuring Multiple Data Pads for Neopixel Accent LEDs

The GHv4 offers versatile configuration options for connecting Neopixel accent LEDs by allowing data to be sent from multiple pads. By default, Neopixel accents can always use the Strip 3 pad, but additional pads can be enabled, which is controlled by the parameter "data" in the "general.txt" file as described in the table below.

PARAMETER	DESCRIPTION
data	• 0: Only the Strip 3 pad is used for Neopixel accents. No additional pads are activated. • 1: Data for Neopixel accent LEDs is sent to both the Strip 3 and the Strip 1 pad. In this configuration, Neopixel accent LEDs that are connected to the Strip 1 pad must be wired in series with the main blade.

	• 2: Data for Neopixel accent LEDs is sent to both the Strip 3 and the Strip 2 pad. If side blades are connected to the Strip 2 pad, the Neopixel accent LEDs must be connected in series. • 3: Data for Neopixel accent LEDs is sent to all three pads: Strip 1, Strip 2, and Strip 3.

Note: For any configuration that uses additional pads, you must set the "number_of_neopixel_accents" parameter to specify the total number of Neopixel accent LEDs.

Staggered Ignition for Neopixel Crossguard Sabers

The GHv4 allows to setup a staggered ignition for Neopixel crossguard sabers. The delay of the side blade ignition and the ignition speed of the side blades are customizable. They are controlled by the two parameters "side_blade_delay" and "side_blade_poweron" in the settings.txt of each sound font and are described in the table below.

PARAMETER	DESCRIPTION
side_blade_delay	Specifies the delay in milliseconds before the side blades ignite.
side_blade_poweron	Specifies the ignition duration for the side blades, determining the time it takes for them to fully ignite.

Blade Detect

The GHv4 automatically detects whether a Neopixel blade or in-hilt setup is in use and configures itself accordingly. This feature works with both illuminated and non-illuminated connectors, as well as TriCree to Neopixel adapters, without the need for resistors in the blade. The functionality depends on the type of connector in your hilt.

Connectors without Illumination or with Separate Data Lines

For connectors without illumination or with illumination using a separate data line, the GHv4 can play sound effects when a Neopixel blade is inserted or removed. The insertion and removal sounds, "bladeinsert.wav" and "bladeremove.wav", can be specific to each sound font. Note that blade detection is only available when the board is not in deep sleep and the blade is not ignited. If your illuminated connector has Neopixel accents, connect the data line to the Strip 3 pad of the GHv4 for this feature to work. Blade detection is controlled by the following parameters in the "general.txt" file.

PARAMETER	DESCRIPTION
blade_detect	• 0: Blade detection is disabled. • 1: Blade detection is enabled. The board automatically configures itself for Neopixel or in-hilt and plays insertion/removal sounds when a Neopixel blade is inserted or removed. No sounds are played for an in-hilt adapter. • 2: Blade detection is enabled, and insertion/removal sounds play for Neopixel blades, but the board does not automatically configure itself for Neopixel or in-hilt.
blade_detect_protection	Specifies the time in milliseconds during which blade detection is disabled after the insertion or removal of a Neopixel blade is detected.
pixel_slot	Specifies the effect font slot used when a Neopixel blade is detected.
in_hilt_slot	Specifies the effect font slot used when an in-hilt setup is detected.

Connectors with Illumination (Single Data Line)

When using illuminated connectors with a single data line shared between Neopixel accent LEDs on the connector and the blade, the GHv4 automatically configures itself to Neopixel. To support detection of a TriCree to Neopixel adapter in this setup, a 1k Ohm resistor must be placed between data and ground within the adapter. In this configuration, insertion and removal sounds "bladeinsert.wav" and "bladeremove.wav" can be played, which are specific to the sound font. Notably, these sounds are swapped: the insertion sound plays upon removing the TriCree to Neopixel adapter, and the removal sound plays when inserting it. Blade detection works only when the board is not in deep sleep and the blade is not ignited. It is controlled by the following parameters in the "general.txt" file.

PARAMETER	DESCRIPTION
blade_detect	• 0: Blade detection is disabled. • 1: Plays insertion/removal sounds and automatically configures to Neopixel. • 2: Plays insertion/removal sounds without automatic configuration to Neopixel.
blade_detect_protection	Specifies the time in milliseconds during which blade detection is disabled after the insertion or removal of a Neopixel blade is detected.
pixel_slot	Specifies the effect font slot used when a Neopixel blade is detected.
in_hilt_slot	Specifies the effect font slot used when an in-hilt setup is detected.

Volume Control

The Golden Harvest v4 includes an on-the-fly volume control feature, allowing you to adjust the saber's volume through gestures. Once in volume control mode, you can increase or decrease volume by tilting, twisting, or a combination of both. For Neopixel setups, the blade will display the current volume level using the active blade effect, providing visual feedback. The maximum volume achievable through gestures is capped by the parameter "volume" in the "general.txt" file, ensuring consistency with your maximum setting. The volume control feature is controlled by three parameters stored in the "general.txt" file, which are described in the table below.

PARAMETER	DESCRIPTION
volume_control	Manages the volume control feature. • 0: Disabled. • 1: Enables volume adjustment by tilting the saber. • 2: Enables volume adjustment by twisting the saber. • 3: Allows volume adjustment by both tilting and twisting the saber.
volume_control_tilt_speed	Adjusts how much you need to tilt the saber to change the volume while in volume control mode. Lower values mean you don't have to tilt the saber as much to change the volume, making it easier to change the volume with small movements. Higher values require you to tilt the saber more to change the volume.
volume_control_twist_speed	Adjusts how much you need to turn the saber to change the volume while in volume control mode. Lower values mean you don't have to turn the saber as much to change the volume, making it easier to change the volume with small twists. Higher values require you to turn the saber more to change the volume.

The volume control for one button setups is described in the following video:
https://www.youtube.com/watch?v=mT-p3Iq2cek

The volume control for two buttons setups is described in the following video:
https://www.youtube.com/watch?v=yJQXlptEjf0

Saber Editor

The saber editor allows you to change general settings on the fly without needing to access the SD card to change the parameters in the "general.txt". The different submenus of the saber editor to change the different settings are called edit modes. For a two momentary buttons setup, you can enter the saber editor by holding the power button before the saber is ignited and then releasing it. Point the saber downwards while doing this. For an one momentary button setup, you can enter the saber editor by twisting your saber quickly while holding the button. Point the saber downwards while doing this.

After you entered the saber editor, you can select which edit mode you want to enter. To cycle through them, tap the power button. To confirm your selection, hold the power button. When cycling through the different edit modes, you reach a "save and leave saber editor" option at some point. To save and leave saber editor, select this option by holding the power button. The following edit modes are available in the saber editor:

- **Color Profile:** Assign new colors to color profiles.
- **Pixel Number:** Adjust the number of pixels the board uses to match the number of pixels in your blade.
- **Ignition Speed:** Adjust the ignition speed individually for each sound font.
- **Retraction Speed:** Adjust the retraction speed individually for each sound font.
- **Ignition Type:** Adjust the ignition type of blade styles
- **Retraction Type:** Adjust the retraction type of blade styles
- **Battle Configuration:** Adjust how to trigger blaster block, force push or lockup.
- **Gesture Controls:** Turn on/off stab ignition, swing ignition, twist ignition or twist retraction.
- **WiFi Setup:** Choose a WiFi mode to enable
- **Bluetooth Setup:** Enable Bluetooth
- **Hilt Calibration:** Calibrate the orientation of the board
- **Effect Font Slot:** Choose the effect font slot that you want to use.

The parameters that you change in the saber editor are saved in the "override-general.txt" file or the "override-settings.txt" file. The parameters in these files will override the parameters in the "general.txt" and the "settings.txt". Thus, you can go back to default by deleting the override files. In the following, we'll describe the different edit modes.

Color Profile Edit Mode

First, let's come to the color profile edit mode which allows to override color profiles and to assign a new color to them. For example, consider you are in blade profile 1 and your first color profile "color1.txt" is defined as red. If you want to assign a new color to this color profile, you can use the color profile edit mode to do that without needing to change the file on the SD card. If you do that, the parameter for the color in the file "color1.txt" is not changed, but a new file "override-color1.txt" is created that contains the new color. When there is both a color profile and an override color profile, the board uses the latter. You can restore the default color of a color profile by stabbing the saber after entering the color profile edit mode and selecting the color profile that you want to be restored or by deleting the corresponding override color profile on the SD card.

After you entered the color profile edit mode, you first need to select the color profile that you want to modify. To cycle through the color profiles, slowly twist your saber. To confirm your selection, tap the power button. After that, you can assign a new blade color to the selected color profile by slowly twisting your saber. To confirm your selection, tap the power button. Next, you can assign a new clash color to the selected color profile. To confirm it, tap the power button. You can continue with this concept to assign new colors for all different saber states, i.e. also for stab, lockup and so on. After you chose the colors that you want to use, save the color profile by holding the power button. There are also different ways to modify the colors of a color profile after you selected it. These are described below. The color profile edit mode is available for all background effects, also for Color Flow and Flame Blade for example. It is controlled by three parameters in the "general.txt". In the following, these parameters are described:

PARAMETER	DESCRIPTION
color_profile_edit_mode	Controls how new colors are assigned to the selected color profile while in the color profile edit mode of the Saber Editor. • 0: Disabled. • 1: Assign new colors by tilting the saber. • 2: Assign new colors by twisting the saber. • 3: Assign new colors by both tilting and twisting the saber.
color_profile_tilt_speed	Adjusts how much you need to tilt the saber to change the color while in the color profile edit mode of the Saber Editor. Lower values mean you don't have to tilt the saber as much to change the color, making it easier to adjust with small movements. Higher values require you to tilt the saber more to change the color.
color_profile_twist_speed	Adjusts how much you need to turn the saber to change the color while in the color profile edit mode of the Saber Editor. Lower values mean you don't have to turn the saber as much to change the color, making it easier to adjust with small twists. Higher values require you to turn the saber more to change the color.

The color profile edit mode for one button setups is described in the following video:
https://www.youtube.com/watch?v=u2RbiqjnYG0

The color profile edit mode for two buttons setups is described in the following video:
https://www.youtube.com/watch?v=cklQdZf7-4U

Pixel Number Edit Mode

The pixel number edit mode allows to adjust the number of pixels the board uses to match the number of pixels in your blade. After you entered the pixel number edit mode, a white spot on the blade indicates the number of pixels. To move it along the blade and thus change the pixel number accordingly, slowly twist your saber until the white spot reaches the tip of the blade. To save the number of pixels, hold the power button. The number of pixels will be saved in the parameter "number_of_leds" in the "override-general.txt". The pixel number edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
pixel_number_edit_mode	A value of 1 enables the pixel number edit mode of the Saber Editor and a value of 0 disables it.
pixel_number_twist_speed	Adjusts how much you need to twist the saber to change the pixel number while in the pixel number edit mode of the Saber Editor. Lower values mean you don't have to twist the saber as much to change the pixel number, making it easier to adjust with small twists. Higher values require more twisting to make changes.

The pixel number edit mode for one button setups is described in the following video:
https://www.youtube.com/watch?v=TuR4OgrCnBE

The pixel number edit mode for two buttons setups is described in the following video:
https://www.youtube.com/watch?v=L6U7mfQN0fQ

Ignition Speed Edit Mode

The ignition speed edit mode allows to adjust the ignition speed individually for each sound font. After you entered the ignition speed edit mode, you need to choose the sound font for which you want to adjust the ignition speed by slowly twisting your saber. If you twist it clockwise, you will go to the next sound font and if you twist it counter clockwise, you will go to the previous sound font. To confirm your selection, tap the power button. After that, you can adjust the ignition speed for this sound font by slowly twisting the saber. The ignition speed is illustrated by a looping ignition of the blade. The blade color that is used for this ignition is defined by the parameter "ignition_led" in the "general.txt". To save the ignition speed, hold the power button. To synchronize the ignition speed with the duration of the poweron sound, perform a stab. The ignition speed will be saved in the parameter poweron in the "override-settings.txt" of the sound font that you selected. The ignition speed edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
ignition_speed_edit_mode	A value of 1 enables the ignition speed edit mode of the Saber Editor and a value of 0 disables it.
ignition_speed_twist_speed	Adjusts how much you need to twist the saber to change the ignition speed while in the ignition speed edit mode of the Saber Editor. Lower values mean you don't have to twist the saber as much, making it easier to adjust ignition speed with small twists. Higher values require more twisting to make changes.

The ignition speed edit mode for one button setups is described in the following video:
https://www.youtube.com/watch?v=ZkhCMXBqb8A

The ignition speed edit mode for two buttons setups is described in the following video:
https://www.youtube.com/watch?v=6Z866zeBXbU

Retraction Speed Edit Mode

The retraction speed edit mode allows to adjust the retraction speed individually for each sound font. After you entered the retraction speed edit mode, you need to choose the sound font for which you want to adjust the retraction speed by slowly twisting your saber. If you twist it clockwise, you will go

to the next sound font and if you twist it counter clockwise, you will go to the previous sound font. To confirm your selection, tap the power button. After that, you can adjust the retraction speed for this sound font by slowly twisting the saber. The retraction speed is illustrated by a looping retraction of the blade. The blade color that is used for this retraction is defined by the parameter "retraction_led" in the "general.txt". To save the retraction speed, hold the power button. To synchronize the retraction speed with the duration of the poweroff sound, perform a stab. The retraction speed will be saved in the parameter poweroff in the "override-settings.txt" of the sound font that you selected. The retraction speed edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
retraction_speed_edit_mode	A value of 1 enables the retraction speed edit mode of the Saber Editor and a value of 0 disables it.
retraction_speed_twist_speed	Adjusts how much you need to twist the saber to change the retraction speed while in the retraction speed edit mode of the Saber Editor. Lower values mean you don't have to twist the saber as much, making it easier to adjust retraction speed with small twists. Higher values require more twisting to make changes.

The retraction speed edit mode for one button setups is described in the following video:
https://www.youtube.com/watch?v=dQdf7r3HPMY

The retraction speed edit mode for two buttons setups is described in the following video:
https://www.youtube.com/watch?v=Te9knMlzUgM

Ignition Type Edit Mode

The ignition type edit mode allows to adjust the ignition type individually for each blade style. After you entered the ignition speed edit mode, you need to choose the blade style for which you want to adjust the ignition type by slowly twisting your saber. If you twist it clockwise, you will go to the next blade style and if you twist it counter clockwise, you will go to the previous blade style. To confirm your selection, tap the power button. After that, you can adjust the ignition type for this blade style by slowly twisting the saber. The ignition type is illustrated by a looping ignition of the blade. To save the ignition type, hold the power button. The ignition type will be saved in the parameter poweron in the "override-background.txt" of the blade style that you selected. The ignition type edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
ignition_type_edit_mode	A value of 1 enables the ignition type edit mode of the Saber Editor and a value of 0 disables it.
ignition_type_twist_speed	Adjusts how much you need to twist the saber to change the ignition type while in the ignition type edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to switch ignition types with small twists. Higher values require more twisting to make changes.

Retraction Type Edit Mode

The retraction type edit mode allows to adjust the retraction type individually for each blade style. After you entered the retraction speed edit mode, you need to choose the blade style for which you want to adjust the retraction type by slowly twisting your saber. If you twist it clockwise, you will go to the next blade style and if you twist it counter clockwise, you will go to the previous blade style. To confirm your selection, tap the power button. After that, you can adjust the retraction type for this blade style by slowly twisting the saber. The retraction type is illustrated by a looping retraction of the blade. To save the retraction type, hold the power button. The retraction type will be saved in the parameter poweron in the "override-background.txt" of the blade style that you selected. The retraction type edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
retraction_type_edit_mode	A value of 1 enables the retraction type edit mode of the Saber Editor and a value of 0 disables it.
retraction_type_twist_speed	Adjusts how much you need to twist the saber to change the retraction type while in the retraction type edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to switch retraction types with small twists. Higher values require more twisting to make changes.

Battle Configuration Edit Mode

The battle configuration edit mode makes it possible to customize how to trigger blaster blocks, force push and lockup. After you selected the battle configuration edit mode, tap the power button to select the action that you want to edit. After that, slowly twist your saber clockwise to enable the respective action or slowly twist your saber counter clockwise to disable it. To save the battle configuration, hold the power button. This will be saved in the parameter for the respective action in the "override-general.txt". The battle configuration edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
battle_configuration_edit_mode	A value of 1 enables the battle configuration edit mode of the Saber Editor and a value of 0 disables it.
battle_configuration_twist_speed	Adjusts how much you need to twist the saber to change the battle configuration while in the battle configuration edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust configurations with small twists. Higher values require more twisting to make changes.

Gesture Controls Edit Mode

The gesture controls edit mode makes it possible to turn on/off the stab ignition, swing ignition, twist ignition and twist retraction feature. After you selected the gesture controls edit mode, tap the power button to select the gesture ignition or retraction that you want to edit. After that, slowly twist your saber clockwise to enable the respective gesture ignition or retraction or slowly twist your saber counter clockwise to disable it. To save the gesture controls, hold the power button. This will be saved in the parameter for the respective gesture ignition or retraction in the "override-general.txt". Make sure that you delete the parameter for the respective gesture ignition or retraction in the "settings.txt"

of each sound font before you start using this edit mode. The gesture controls edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
gesture_controls_edit_mode	A value of 1 enables the gesture controls edit mode of the Saber Editor and a value of 0 disables it.
gesture_controls _twist_speed	Adjusts how much you need to twist the saber to turn a gesture control on or off while in the gesture controls edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to turn a gesture control on or off with small twists. Higher values require more twisting to make changes.

The gesture controls edit mode for one button setups is described in the following video:
https://www.youtube.com/watch?v=PcDPhTkDJ4w

The gesture controls edit mode for two buttons setups is described in the following video:
https://www.youtube.com/watch?v=Us5-4biC7eQ

WiFi Setup Edit Mode

The WiFi setup edit mode allows you to configure the WiFi mode. After selecting the WiFi setup edit mode, twist the saber slowly to cycle through the available WiFi modes. Once you reach your desired mode, hold the power button to confirm your selection. The chosen WiFi mode will be saved to the "override-general.txt" file and the board will reboot with the respective WiFi mode enabled. This feature is managed by two parameters in the "general.txt" file, described below.

PARAMETER	DESCRIPTION
wifi_setup_edit_mode	A value of 1 enables the WiFi setup edit mode of the Saber Editor and a value of 0 disables it.
wifi_setup_twist_speed	Adjusts how much you need to twist the saber to change the WiFi settings while in the WiFi setup edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust WiFi settings with small twists. Higher values require more twisting to make changes.

Bluetooth Setup Edit Mode

The Bluetooth setup edit mode allows you to turn Bluetooth on/off. After selecting the Bluetooth setup edit mode, twist the saber slowly clockwise to turn it off or counter clockwise to turn it on. Then, hold the power button to confirm your selection. It will be saved to the "override-general.txt" file and the board will reboot with the Bluetooth turned on/off. This feature is managed by two parameters in the "general.txt" file, described below.

PARAMETER	DESCRIPTION
ble_setup_edit_mode	A value of 1 enables the Bluetooth setup edit mode of the Saber Editor and a value of 0 disables it.
ble_setup_twist_speed	Adjusts how much you need to twist the saber to change the Bluetooth settings while in the Bluetooth setup edit mode of the Saber Editor. Lower values mean less twisting is needed,

	making it easier to adjust Bluetooth settings with small twists. Higher values require more twisting to make changes.

Hilt Calibration Edit Mode

The hilt calibration edit mode allows you to calibrate the orientation of the board within your saber. This calibration is particularly beneficial for curved hilts, as it enables the board to understand its specific positioning, improving the accuracy of gesture-based controls. To perform a calibration, simply follow the intuitive prompts given by the board. This process involves maneuvering the saber through various positions and orientations. The calibration values are stored in the "override-general.txt" file. This feature is managed by a parameter in the "general.txt" file, described below.

PARAMETER	DESCRIPTION
hilt_calibration_edit_mode	A value of 1 enables the hilt calibration edit mode of the Saber Editor and a value of 0 disables it.

Effect Font Slot Edit Mode

The effect font slot edit mode allows to change the effect font slot that you are using. After you entered the effect font slot edit mode, you can select the effect font slot that you want to use by slowly twisting your saber. To confirm your selection, hold the power button. This will be saved in the parameter "effect_font_slot" in the "override-general.txt". The effect font slot edit mode is controlled by two parameters in the "general.txt". In the following, these parameters are described.

PARAMETER	DESCRIPTION
effect_font_slot_edit_mode	A value of 1 enables the effect font slot edit mode of the Saber Editor and a value of 0 disables it.
effect_font_slot_twist_speed	Adjusts how much you need to twist the saber to change the effect font slot while in the effect font slot edit mode of the Saber Editor. Lower values mean less twisting is needed, making it easier to adjust the effect font slot with small twists. Higher values require more twisting to make changes.

Blaster Block

There are two different options to trigger a blaster block. The parameters in the "general.txt" file that define this feature are described in the following.

PARAMETER	DESCRIPTION
blaster	Controls how Blaster Blocks are triggered. • 0: Tapping the power button (one button setup) or the auxiliary button (two button setup). • 1: Swinging the saber after entering MultiBlast mode, which is activated by tapping the power button (one button setup) or the auxiliary button (two button setup). If no swing is detected within a customizable time, MultiBlast mode exits automatically. Tutorial videos: • One button setup

	• Two button setup
blaster_disable	Time in milliseconds after which MultiBlast mode is exited automatically if no swing is detected.
blaster_threshold	Defines the swing strength required to trigger a Blaster Block in MultiBlast mode.

Force Push

There are two different options to trigger a force push. The parameters in the "general.txt" file that define this feature are described in the following.

PARAMETER	DESCRIPTION
force_push	Controls how force push effects are triggered. • One Button Setup: ○ 0: Twisting the saber while holding the button. ○ 1: Swinging the saber while holding the button. ○ Tutorial video • Two Button Setup: ○ 0: Tapping the power button. ○ 1: Swinging the saber while holding the auxiliary button. ○ Tutorial video
force_push_threshold	Sets the required swing strength to trigger a force push when the force push is activated by swinging the saber while holding the button. Lower values increase sensitivity, making it easier to trigger a force push with lighter swings, while higher values decrease sensitivity, requiring a stronger swing.

Lockup

There are two different options to trigger lockup. The parameter "lockup" in the "general.txt" file allows to choose one of these options and is described in the following.

PARAMETER	DESCRIPTION
lockup	Controls how lockup effects are triggered. • One Button Setup: ○ 0: Holding the power button, releasing it, and then twisting the saber. ○ 1: Holding the power button, releasing it, and then triggering a clash. ○ Tutorial video • Two Button Setup: ○ 0: Holding the auxiliary button and then tapping it. ○ 1: Holding the auxiliary button, releasing it, and then triggering a clash. ○ Tutorial video

Sound Fonts

Sound fonts define the audio experience of your saber, just as blade styles define its visual effects. Each sound font consists of a collection of audio files tailored to specific saber actions - swing sounds, clash sounds, stab sounds, spin sounds, lockup sounds, melt sounds, blaster block sounds, force push sounds, boot sounds, power sounds, menu sounds, hum sounds, and more. The GHv4 supports an unlimited number of sound files for each effect. If you include multiple files for any effect, the saber randomly selects one each time the effect is triggered, enhancing variety and immersion. Sound fonts differ in style, providing atmospheres ranging from Jedi to Sith, allowing users to customize their experience.

Organizing Sound Fonts

Sound fonts are stored in the "SoundFonts" folder on your microSD card, each in its own folder named in the format "sound[number]", where [number] is a continuous enumeration (e.g., sound1, sound2, etc.). Optionally, you can add a description to each sound font after "sound[number]" in the name of the folder (see image below).

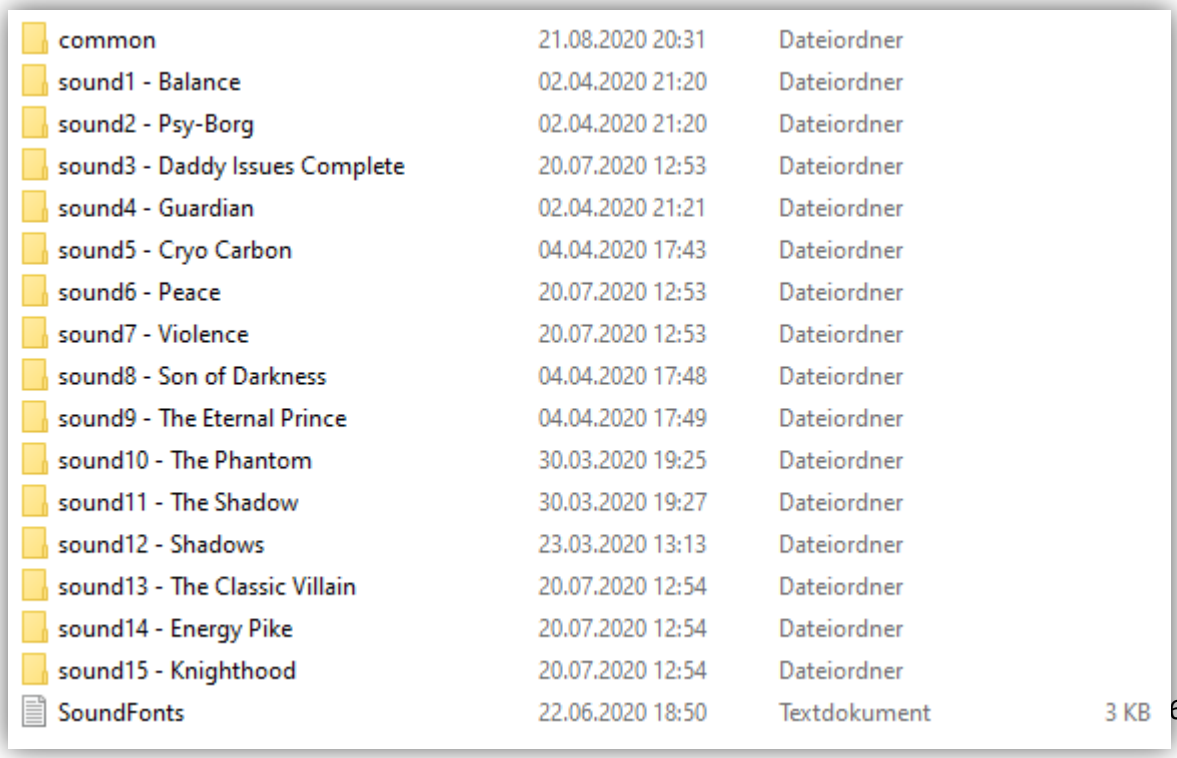
To add a new sound font:

- Create a new folder in the format above, ensuring the number is unused.
- Place the sound files in this folder, maintaining continuous folder enumeration.

Note: Ensure your microSD card is formatted to FAT32. The included microSD card is already formatted accordingly.

Finding Sound Fonts

Many high-quality sound fonts are available online, with www.saberfont.com being a key source. You can use all of the following sound fonts, even without additional renaming of the sound files: http://www.saberfont.com/Optimized-for-Plecter-CFX-Smooth-Swing-Compatible-by-Era-_c_92.html



Sound Font Settings File

Each sound font can be customized using a "settings.txt" file. The parameters contained in this file define the behavior of the Smooth Swing algorithm and allow to adjust the ignition and retraction speed of your blade. The parameters are described in the table below.

PARAMETER	DESCRIPTION
motion_theshold	Defines the minimum rotation speed of the saber required to start the smooth swing algorithm.
swing_sensitivity	Defines the rotation speed of the saber at which the smooth swing sounds reach their maximum volume.
swing_sharpness	Defines how sharp the volume of the smooth swing sounds rises with the rotation speed of the saber.
swing_max_volume	Defines the factor by which the maximum volume of the smooth swing sounds is multiplied compared to normal volume.
accent_swing_threshold	Defines the minimum rotation speed of your saber required to trigger an accent swing. The lower this value, the more sensitive your accent swing detection is.
accent_spin_threshold	Defines the minimum rotation speed of your saber required to trigger an accent spin. The lower this value, the more sensitive your accent spin detection is.
hum_max_reduction	Defines by which percentage the hum volume is reduced during a strong swing.
first_transition_deg	A full rotation of the saber covers an angle of 360°. As part of the smooth swing algorithm, the sounds "lswing[number].wav" and "hswing[number].wav" are crossfaded with each other as the saber rotates. It is chosen randomly with which sound file to begin with. In a first transition, the crossfading occurs and in a second transition, the crossfading is reversed again. This parameter defines the angle around which the saber needs to be rotated to trigger the first transition.
second_transition_deg	A full rotation of the saber covers an angle of 360°. As part of the smooth swing algorithm, the sounds "lswing[number].wav" and "hswing[number].wav" are crossfaded with each other as the saber rotates. It is chosen randomly with which sound file to begin with. In a first transition, the crossfading occurs and in a second transition, the crossfading is reversed again. This parameter defines the angle around which the saber needs to be rotated to trigger the first second.
stab_ignition	Defines if a stab ignites the saber. A value of 0 disables this feature and a value of 1 enables it. The same parameter can be found in the "general.txt" file. However, the parameter contained in the "settings.txt" will overwrite this parameter.
swing_ignition	Defines if a swing ignites the saber. A value of 0 disables this feature and a value of 1 enables it. The same parameter can be found in the "general.txt" file. However, the parameter contained in the "settings.txt" will overwrite this parameter.

twist_retraction	Defines if a twist retracts the saber. A value of 0 disables this feature and a value of 1 enables it. The same parameter can be found in the "general.txt" file. However, the parameter contained in the "settings.txt" will overwrite this parameter.
poweron	Defines the ignition duration of the blade. If a value of 0 is chosen, the ignition duration is coupled to the length of the power on sound. If a value different from 0 is chosen, then this value gives the ignition duration. A value of 100 corresponds to a fast ignition and a value of 500 to a slow igntion for example. The time is given in units of 6 ms. The same parameter can be found in the "general.txt" file. However, the parameter contained in the "settings.txt" will overwrite this parameter.
poweroff	Defines the retraction duration of the blade. If a value of 0 is chosen, the retraction duration is coupled to the length of the power on sound. If a value different from 0 is chosen, then this value gives the retraction duration. A value of 100 corresponds to a fast retraction and a value of 500 to a slow retraction for example. The time is given in units of 6 ms. The same parameter can be found in the "general.txt" file. However, the parameter contained in the "settings.txt" will overwrite this parameter.
user_interface_sounds_volume	Defines the volume of the user interface sounds relative to the total volume. A value of 1.0 corresponds to 100% of the total volume.
effect_sounds_volume	Defines the volume of the effect sounds relative to the total volume. A value of 1.0 corresponds to 100% of the total volume.
side_blade_delay	Specifies the delay in milliseconds before the side blades ignite.
side_blade_poweron	Specifies the ignition duration for the side blades, determining the time it takes for them to fully ignite.

Default Sound Font Package

The GHv4 comes pre-loaded with 36 unique sound fonts on the microSD card, each adding a distinct personality to your saber. We owe special thanks to the talented sound font designers who made this project possible. Be sure to explore more of their incredible work!

BKSaberSounds: "The Harvester"
Official Store: https://www.bksabersounds.com/
Facebook: https://www.facebook.com/BKSaberSounds
Instagram: https://www.instagram.com/bksabersounds
YouTube: https://www.youtube.com/channel/UCcA35kOrOSdrWpbxVqyCpsg

Blueforce Company: "Dark Harvest"
Official Store: https://www.blueforcecompany.com/
Facebook: https://www.facebook.com/blueforcecompany
Instagram: https://www.instagram.com/blueforcecompany
YouTube: https://www.youtube.com/@blueforcecompany

CrystalSoniX: "Psy-Borg"
Store Page: https://saberfont.com/Crystal-Sonix-_c_33.html
YouTube: https://www.youtube.com/channel/UCFq3tKhz1dFWIhM8gHCVlGA

Dark Path Media: "Son of Darkness"
Store Page: https://saberfont.com/Dark-Path-Media_bymfg_34-0-1.html
YouTube: https://www.youtube.com/c/DarkPathMedia

Echo Studios: "Shadows"
Store Page: www.saberfont.com/Echo_bymfg_7-0-1.html

Fourzze FX: "Daddy Issues Complete", "The Classic Villain", "Knighthood"
Store Page: https://saberfont.com/Fourzze-FX-Saber-Sound-Designs-_c_126.html
YouTube: https://www.youtube.com/user/sirashram3

Greyscale Fonts: "Stitched", "Mercenary", "Awaken", "Behemoth"
Official Store: https://www.greyscalefonts.com/
YouTube: https://www.youtube.com/@GreyscaleFonts
Facebook: https://www.facebook.com/greyscalefonts
Instagram: https://www.instagram.com/greyscalefonts
Reddit: https://www.reddit.com/u/gussillypants

JayDalorian: "LU-10"
Official Store: https://jaydalorian.com/
Instagram: https://www.instagram.com/jaydalorian_official

JuanSith: "Light Warrior", "Dark Warrior"
Store Page: https://saberfont.com/Juansith_c_19.html
Facebook: https://www.facebook.com/JuanSithSaberFonts

Kyberphonic Fonts: "Peace", "Violence"
Official Store: https://kyberphonic.bigcartel.com/
Instagram: https://www.instagram.com/kyberphonic
YouTube: https://www.youtube.com/c/Kyberphonic

LDN Sabers: "Guardian"
Store Page: https://saberfont.com/LDN-Sabers_c_79.html

LINK: "The Eternal Prince", "Cryo Carbon"
Store Page: https://saberfont.com/Link_c_20.html

LordBlako Saber Fonts: "Balance"
Store Page: https://saberfont.com/Lord-Blako_c_15.html

Mountain Sabers: "Angelic Plazma", "Dark Sith Red", "Electric Essence", "Jurassic", "Kilovolt", "Quantum Knight", "The Mag-Cyl", "The Prize", "Torch", "UnHinged"
Official Store: https://www.mountainsabersfonts.com/shop

Phoenix Sabers: "Princess Bride", "Dark Republic"
Store Page: https://saberfont.com/Phoenix_c_25.html
Official Store: http://phoenixsabers.com/

Project Fonts: "The Phantom", "The Shadow"
Store Page: https://saberfont.com/Project-Fonts_bymfg_29-0-1.html

The Proplicator: "Energy Pike"
Store Page: https://saberfont.com/The-Proplicator_bymfg_33-0-1.html

Guys, thank you so much for supporting our Golden Harvest project by providing these amazing sound fonts!

Also special thanks to The Rebellion Outpost who provided the default blade change sounds! Visit their site on https://www.facebook.com/TheRebellionOutpostLLC/.

Sound File Requirements

The GHv4 supports all sound fonts contained in the following category of www.saberfont.com even without additional renaming required:
http://www.saberfont.com/Optimized-for-Plecter-CFX-Smooth-Swing-Compatible-by-Era-_c_92.html

Nevertheless, we recommend to use our own naming system which is given in the following. If you want to design your own sound font, please make sure that the sound files have a sampling rate of 22,050 Hz or 44,100 Hz and a resolution of 16 bit. The following sound types are used by our Golden Harvest board:

- font[number].wav: identification sound of the font
- boot[number].wav
- preon[number].wav
- pwron[number].wav
- pstoff[number].wav
- pwroff[number].wav
- hum[number].wav
- clash[number].wav
- stab[number].wav
- swing[number].wav: accent swings
- spin[number].wav: accent spins
- hswing[number].wav
- lswing[number].wav
- bgnlock[number].wav
- lock[number].wav
- endlock[number].wav
- bgndrag[number].wav
- drag[number].wav
- enddrag[number].wav
- bgnmelt[number].wav
- melt[number].wav
- endmelt[number].wav
- blast[number].wav
- force[number].wav
- track[number].wav
- change.wav: sound played when changing blade profiles and color profiles (optional)
- save.wav: sound when saving a blade profile or color profile (optional)

Hereby, the sound files are enumerated continuously by [number]. There is an unlimited number of different files supported for each sound type which are selected randomly when triggered. The GHv4 uses default files for the menu sounds, so you don't need to add them to your sound font. If you want to add them in order to underline the overall atmosphere your sound font creates though, you would have to create the two files "change.wav" and "save.wav". The first is played when the on top effect is changed and the latter is played when the effect menu is leaved.

Background Music

The GHv4 supports custom background music, ideal for adding atmosphere during dueling. To add background music, ensure each music file is in .wav format with a 16-bit depth and a sampling rate of either 22,050 Hz or 44,100 Hz. Name each file as "track[number].wav" with [number] as an incremental track number. To assign tracks to a specific sound font, place them in the corresponding "sound[number]" folder on the microSD card. The GHv4's playlist feature supports multiple tracks per sound font, enabling continuous playback. When background music is activated (see the section "Menu Navigation" for details), the board will begin with "track1.wav", followed by "track2.wav" after the first track ends, and so on. Once the last track finishes, playback loops back to "track1.wav".

The following video describes how to trigger background music for single momentary button setups: https://www.youtube.com/watch?v=L47oo5a4OxQ

The following video describes how to trigger background music for two momentary buttons setups: https://www.youtube.com/watch?v=L47oo5a4OxQ

Common Folder

Within the "SoundFonts" folder, there's a "common" folder that acts as a fallback resource. If a sound font lacks a certain type of sound file, the GHv4 will look in the "common" folder and use any matching files it finds. However, if a sound font has even one file of a particular sound type, the GHv4 will exclusively use files from that font and ignore any additional files of that type in the "common" folder. This setup provides flexibility by allowing shared resources across sound fonts while still prioritizing specific files when available.

Sound Menu

For two buttons setups, point the saber upwards while holding and releasing the auxiliary button to enter the sound menu; for one button setups, point the saber upwards while holding and releasing the power button. There are three parameters in the "general.txt" file that control how to change sound fonts, which are decribed in the following.

PARAMETER	DESCRIPTION
sound_menu	Controls how you navigate through sound fonts in the sound menu. • One Button Setup: ○ 1: Navigate through sound fonts by slowly twisting the saber; twist clockwise for the next sound font and counterclockwise for the previous sound font. Tap the button while pointing the saber upwards to jump forward or downwards to jump backward through sound fonts, as defined by "sound_font_jump". ○ 2: Navigate by pointing the saber upwards or downwards while twisting in any direction; orientation determines the direction of

	navigation. Tap the button while pointing the saber upwards to jump forward or downwards to jump backward through sound fonts, as defined by "sound_font_jump". ○ 3: Tap the button to navigate to the next or previous sound font based on the saber's orientation. Quickly twist the saber while pointing upwards to jump to the next sound font or downwards to jump to the previous one, as defined by "sound_font_jump". ○ Tutorial video • Two Button Setup: ○ 1: Navigate through sound fonts by slowly twisting the saber; twist clockwise for the next font and counterclockwise for the previous font. ○ 2: Navigate to the next or previous sound font by pointing the saber upwards or downwards while twisting. The twist direction does not matter. ○ 3: Use the auxiliary button to navigate to the next or previous sound font depending on whether the saber is pointing upwards or downwards. Tap the power button while pointing the saber upwards to jump forward or downwards to jump backward by a number of fonts specified in "sound_font_jump". ○ Tutorial video
sound_font_jump	Specifies the number of sound fonts to skip when jumping forward or backward in the sound menu.
sound_menu_twist_speed	Adjusts how much you need to turn the saber to change sound fonts while in the sound menu. Lower values mean you don't have to turn the saber as much to switch sound fonts, making it easier to navigate with small twists. Higher values require you to turn the saber more to cycle through sound fonts.

Pairing Sound Fonts with Blade Styles, Color Profiles, On Top Effects and Accent Styles

Each sound font can be paired with a specific blade style, color profile, on top effect, standby accent style, and ignited accent style. You can set these pairings directly on the SD card, or adjust them while using the saber. Any changes you make on the fly, such as selecting a new blade style during use, will be automatically saved. The pairings are saved in the "prefs[number].txt" file, located in each effect font slot, which contains the parameters that define which blade style, color profile, etc. are assigned to the corresponding sound font "sound[number]". The image below provides an example of this file's content, showing how each sound font aligns with its designated visual settings.

```
blade_profile=1
color_profile=1
on_top_effect=1
standby_accent_profile=1
ignited_accent_profile=1
```

Preon sounds

It is possible to trigger preon sounds in different ways. The parameter "preon" defines how the preon sounds can be triggered. In the following, this parameter is described.

PARAMETER	DESCRIPTION
preon	Controls how preon sounds and light effects are triggered. • 0: Disabled. • 1: Play with every ignition. • 2: Play only when saber pointed downwards at ignition. • 3: Play only when saber pointed upwards at ignition. • 4: Play only with power button ignition. • 5: Play only with gesture ignition.

Postoff sounds

It is possible to trigger postoff sounds in different ways. The parameter "postoff" defines how the postoff sounds can be triggered. In the following, this parameter is described.

PARAMETER	DESCRIPTION
postoff	Controls how postoff sounds and light effects are triggered. • 0: Disabled. • 1: Play with every retraction. • 2: Play only when saber is pointed downwards at retraction. • 3: Play only when saber is pointed upwards at retraction. • 4: Play only with power button retraction. • 5: Play only with gesture retraction.

Firmware Version Indicator File

The GHv4 automatically creates a firmware version indicator file, which includes the currently installed firmware version in its name. This file serves as a straightforward reference for users to quickly verify the firmware version installed on their lightsaber without the need to connect to software or navigate through complex settings. Each time the firmware is updated, the board updates this file to reflect the new version, ensuring that users always have accurate and accessible information about the system's status. The presence of this file on the SD card makes it easier for users to manage their device, particularly when troubleshooting or consulting support forums. It provides a clear and immediate way to confirm that updates have been successfully applied or to ascertain the firmware version when assessing compatibility with other software or new features.

Firmware Updates

The set of features of the GHv4 is always expanding. It can be extended by firmware updates which are easy to install without requiring any coding skills. You can either install it directly on the microSD card or using the Seedling v2 if you do not want to remove the microSD card.

The GHv4 introduces an enhanced firmware update process designed to significantly increasing reliability. Similar to the GHv3, users can easily update their soundboard's firmware by simply dropping the new firmware file onto the SD card. To safeguard against potential issues such as power loss during an update - a concern that could previously lead to system failures that required to reinstall the old

firmware version manually - the GHv4 now incorporates a robust fallback safety mechanism. If the update process is interrupted or fails for any reason, the board automatically detects the issue and signals the failure with a sound. This immediate feedback allows users to recognize the problem without guessing what might have gone wrong. Furthermore, after indicating a failed update, the soundboard automatically reverts to the last working version of the firmware. This safety feature ensures that the lightsaber remains operational and stable, preventing the device from being rendered unusable due to a corrupted update. The fallback mechanism provides peace of mind, making firmware updates worry-free and more secure.

Installing Directly on the microSD Card

To install a firmware update directly on the microSD card, just do the following:

1. Disconnect your board from power (for example, by removing the battery or inserting the kill key)
2. Pull the microSD card out and insert it into your computer.
3. Download the latest firmware update from our website: https://sabertec.net/downloads/
4. Extract the .zip folder and put the .bin file on your microSD card.
5. Insert the microSD card back into your board.
6. Power your board again. The firmware update will be automatically installed, notifying you about the start and end of the process via sounds. The progress is indicated by the onboard status LED and standard accent LEDs, which pulse faster and faster as the firmware update nears completion.

Installing Using the Seedling v2

To install a firmware update using the Seedling v2, just do the following:

1. Connect your saber via USB to your computer while the saber is in idle mode. That's the state the saber is in before igniting the blade (you may have to wake it up from deep sleep first).
2. Wait until your computer detects your microSD card as mass storage device.
3. Download the latest firmware update from our website: https://sabertec.net/downloads/
4. Extract the .zip folder and put the .bin file on your microSD card.
5. Unplug the USB connection to your computer. The firmware update will be automatically installed, notifying you about the start and end of the process via sounds. The progress is indicated by the onboard status LED and standard accent LEDs, which pulse faster and faster as the firmware update nears completion.

Onboard Status LED

The Golden Harvest v4 soundboard includes an enhanced diagnostic tool with its onboard status LED, a feature designed to provide clear and intuitive indications of the board's operational status. This LED not only complements other diagnostic signals but also adds several useful indicators to help users understand the state of their device at a glance.

Indicating SD Card Issues

In line with the visual and auditory signals for SD card problems, such as "No SD card detected," the onboard status LED also participates by blinking when an SD card issue is detected. This synchronized

signaling helps ensure that users are alerted to potential problems, no matter their focus during operation.

Indication of Operation

The status LED remains constantly lit to signify that the board is powered and functioning correctly. This continuous illumination provides a quick and easy visual cue that the lightsaber is operational, enhancing user confidence during use.

Firmware Update Indicator

During a firmware update, the behavior of the onboard status LED further aids in communication with the user. It exhibits a breathing effect, slowly pulsing to indicate that the update is in progress. As the update nears completion, the pulsing gradually speeds up, giving users a visual representation of the update progress. This feature not only adds to the aesthetic experience but also keeps users informed about the status of critical operations. Additionally, the accent LEDs connected to the LED5 - 10 pads will display the same breathing effect as the onboard status LED, providing a consistent and synchronized visual indication of the firmware update status.

Onboard Status LED Control

The onboard status LED is controlled by the parameter in the following table. It is stored in the config files.

Parameter	Value
status=0	0: Disables Onboard Status LED 1: Enables Onboard Status LED

USB-C Charging and File Transfer Using the Seedling v2

The GHv4 supports USB-C file transfer using the Seedling module. It allows you to access the microSD card without needing to pull it out of your board. To activate the USB feature, set the parameter "usb=1" in the "general.txt" file. To enter the mass storage device mode, connect your Seedling v2 via USB-C to your computer when the GHv4 is in idle mode. That's the state the saber is in after booting and before igniting the blade. If the board is in deep sleep, make sure to wake it up before. When connecting the Seedling v2 to your computer for the first time, it can take several minutes until your computer recognizes the microSD card as mass storage device as it will install the driver. The USB-C file transfer is currently supported on Windows 10 and Windows 11. It may work on other platforms as well, but we cannot guarantee it.

OLED Display Support

The Golden Harvest v4 soundboard now includes support for OLED displays, significantly enhancing the user interface with high-quality visual feedback. This feature supports the popular SSD1306 model OLED display, which offers a resolution of 128 x 32 pixels, perfectly suited for crisp and clear imagery.

Static Image Display

The board can display static images that are input as .bmp files with dimensions exactly matching 128 x 32 pixels. These images are shown as static displays on the OLED screen, allowing for custom graphics like logos, icons, or text to be presented in a visually appealing way.

Animated Displays

For more dynamic visual effects, the board supports animations. Users can input .bmp files sized at 128 x (n x 32) pixels, where 'n' indicates the number of frames in the animation sequence. The OLED display cycles through each 128 x 32 pixel slice, treating it as a single frame of the animation. This creates a seamless animated sequence that continuously loops. The speed of the animation is adjustable, allowing users to customize the display rhythm to their preference.

Text Display Capabilities

The OLED display is not limited to images and animations; it can also show text information. This is particularly useful for displaying current settings such as blade style, color profile, on top effect, sound font names, and assists in navigating through the saber editor. This feature makes it easier for users to see and adjust their settings directly on the saber.

Content Rotation

An additional functionality is the ability to rotate the entire display content by 180°. This feature is useful for ensuring the display is correctly oriented regardless of how the saber is held or mounted, enhancing readability and user experience. The integration of OLED display support into the Golden Harvest v4 soundboard brings a new level of interaction and customization, allowing users to visually enhance their lightsaber experience with detailed graphics and useful information displays. The following table provides an overview of the parameters that control the OLED display features.

Parameter	Value
oled=1	0: Disables OLED 1: Enables OLED
oled_orientation=0	0: Standard orientation 1: Rotate content of the display by 180°
oled_screensaver=100	Specifies the time in seconds after which the screen saver should be displayed

Note: The OLED display support is currently in beta testing and will not be available in the initial release of the GHv4. It is planned to be included in a future firmware update.

Menu Navigation

Navigation Using Two Momentary Buttons

In this setup, the two buttons are referred to as the auxiliary button and the power button. To enable this button configuration, set the parameter "button_mode=2" in the "general.txt" file on your microSD card.

FUNCTION	TRIGGERING WITH TWO BUTTONS
Power On	Tap the power button
Change Color Profile*	Tap the power button while holding the auxiliary button
Enter Effect Menu	Hold the power button, then hold it again. Point the saber upwards while doing this.
∟ Change Blade Profile*	Slowly twist the saber
∟ Change On Top Effect*	Tap the auxiliary button
∟ Save And Leave	Hold the power button until the confirmation sound is played

Enter Accent Menu	Hold the power button, then hold it again. Point the saber downwards while doing this.
∟ Change Ignited Accent Profile*	Tap the power button
∟ Go To Ignited Accent Profile	Hold the power button
∟ Change Standby Accent Profile*	Tap the power button
∟ Save And Leave	Hold the power button until the confirmation sound is played
Enter Color Change Menu	Hold the power button, then tap it. Point the saber upwards while doing this.
∟ Cycle Through Preset Color Profiles	Slowly twist the saber
∟ Save And Leave	Hold the power button
Enter Lockup Mode	Hold the auxiliary button, then tap it. Point the saber upwards while doing this.
∟ Leave Lockup Mode	Tap the auxiliary button
Enter Drag Mode	Hold the auxiliary button, then tap it. Point the saber downwards while doing this.
∟ Leave Drag Mode	Tap the auxiliary button
Enter Melt Mode	Hold the power button, release it and then perform a stab
∟ Leave Melt Mode	Tap the power button
Blaster Block	Tap the auxiliary button
Force Push	Hold the auxiliary button while swinging the saber
Activate Power Saving	Hold the power button while performing a stab
Deactivate Power Saving	Hold the power button while performing a stab
Power Off	Hold the power button until the saber shuts down
Enter Sound Font Menu	Hold the auxiliary button before the blade is ignited. Point the saber upwards while doing this.
∟ Go To Next Sound Font*	Slowly twist the saber
∟ Jump 10 Sound Fonts Further*	Tap the power button
∟ Save And Leave Sound Font Menu	Hold the auxiliary button
Show Battery Indicator	Hold the power button while performing a stab before the saber is ignited
∟ Calibrate Battery	When your battery is fully charged, perform a stab while battery level is displayed
Enter Volume Control	Hold the auxiliary button before the blade is ignited. Point the saber downwards while doing this.
∟ Adjust Volume	Slowly twist the saber
∟ Save And Leave Volume Control	Tap the power button
Start Background Music	Hold the power button before the saber is ignited, then release it. Point the saber upwards while doing this.
∟ Skip Track	Tap the auxiliary button
∟ Stop Background Music	Hold the power button before the saber is ignited, then release it
Enter Saber Editor	Hold the power button before the saber is ignited, then release it. Point the saber downwards while doing this.
∟ Select Editor Mode	Tap the power button
∟ Color Profile	
∟ Select Color Profile	Slowly twist the saber

∟ Edit Color	Slowly twist the saber
∟ Go To Next Saber State	Tap the power button
∟ Revert Colors To Default	Perform a stab
∟ Save Color Profile	Hold the power button
∟ Confirm Selection	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ Pixel Number	
∟ Edit Pixel Number	Slowly twist the saber
∟ Save and Leave Editor Mode	Hold the power button
∟ Ignition Speed	
∟ Select Sound Font	Slowly twist the saber
∟ Edit Ignition Speed	Slowly twist the saber
∟ Save Ignition Speed	Hold the power button
∟ Confirm Selection	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ Retraction Speed	
∟ Select Sound Font	Slowly twist the saber
∟ Edit Retraction Speed	Slowly twist the saber
∟ Save Retraction Speed	Hold the power button
∟ Confirm Selection	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ Ignition Type	
∟ Select Blade Style	Slowly twist the saber
∟ Edit Ignition Type	Slowly twist the saber
∟ Save Ignition Type	Hold the power button
∟ Confirm Selection	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ Retraction Type	
∟ Select Blade Style	Slowly twist the saber
∟ Edit Retraction Type	Slowly twist the saber
∟ Save Retraction Type	Hold the power button
∟ Confirm Selection	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ Battle Configuration	
∟ Blaster Block Mode	Tap the power button
∟ Button and Swinging	Slowly twist the saber clockwise
∟ Button	Slowly twist the saber counterclockwise
∟ Force Push Mode	Tap the power button
∟ Button and Swinging	Slowly twist the saber clockwise
∟ Button	Slowly twist the saber counterclockwise
∟ Lockup Mode	Tap the power button
∟ Button and Clashing	Slowly twist the saber clockwise
∟ Button	Slowly twist the saber counterclockwise
∟ Save and Leave Editor Mode	Hold the power button
∟ Gesture Controls	
∟ Stab Ignition	Tap the power button
∟Turn stab ignition on	Slowly twist the saber clockwise
∟Turn stab ignition off	Slowly twist the saber counterclockwise
∟ Twist Ignition	Tap the power button

∟Turn twist ignition on	Slowly twist the saber clockwise
∟Turn twist ignition off	Slowly twist the saber counterclockwise
∟ Swing Ignition	Tap the power button
∟Turn swing ignition on	Slowly twist the saber clockwise
∟Turn swing ignition off	Slowly twist the saber counterclockwise
∟ Twist Retraction	Tap the power button Tap the power button
∟Turn twist retraction on	Slowly twist the saber clockwise
∟Turn twist retraction off	Slowly twist the saber counterclockwise
∟ Go To Next Gesture Control	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ WiFi Setup	
∟ Select WiFi mode	Slowly twist the saber
∟ Confirm Selection	Hold the power button
∟ Bluetooth Setup	
∟ Turn Bluetooth on	Slowly twist the saber clockwise
∟ Turn Bluetooth off	Slowly twist the saber counterclockwise
∟ Confirm Selection	Hold the power button
∟ Hilt Calibration	
∟ Effect Font Slot	
∟ Select effect font slot	Slowly twist the saber
∟ Save and Leave Editor Mode	Hold the power button
∟ Save And Leave Saber Editor	
∟ Confirm Selection	Hold the power button
Toggle WiFi	Hold the power button for at least 5 s
Toggle Bluetooth	Hold the power button for at least 7.5 s
Reset	Hold the power button for at least 10 s
Wake Up From Deep Sleep	Tap the power button

*** When changing the On Top Effect, Background Effect or Sound Font, you can go to the next one by pointing your saber upwards during selection or to the previous one by pointing your saber downwards.**

Navigation Using One Momentary Button

All features of the board can be accessed using just a single momentary button. To select this button configuration, set the parameter "button_mode=4" in the "general.txt" file on your microSD card.

FUNCTION	TRIGGERING WITH ONE MOMENTARY SWITCH
Power On	Tap the button
Enter Effect Menu	Hold the button, then hold it again. Point the saber upwards while doing this.
∟ Change Blade Profile*	Slowly twist the saber
∟ Change On Top Effect*	Tap the button
∟ Save And Leave	Hold the button until the confirmation sound is played
Enter Accent Menu	Hold the button, then hold it again. Point the saber downwards while doing this.
∟ Change Standby Accent Profile*	Tap the button
∟ Enter Ignited Accent Submenu	Hold the button for 1 s
∟ Change Ignited Accent Profile*	Tap the button
∟ Save And Leave	Hold the button until the confirmation sound is played

Enter Color Change Menu	Hold the button, then tap it. Point the saber upwards while doing this.
∟ Cycle Through Preset Color Profiles	Slowly twist the saber
∟ Save And Leave	Hold the button
Enter Lockup Mode	Hold the button, release it and then twist the saber. Point the saber upwards while doing this.
∟ Leave Lockup Mode	Tap the button
Enter Drag Mode	Hold the button, release it and then twist the saber. Point the saber downwards while doing this.
∟ Leave Drag Mode	Tap the button
Enter Melt Mode	Hold the power button, release it and then perform a stab
∟ Leave Melt Mode	Tap the power button
Blaster Block	Tap the button
Force Push	Hold the button while swinging the saber
Activate Power Saving	Hold the power button while performing a stab
Deactivate Power Saving	Hold the power button while performing a stab
Power Off	Hold the power button until the saber shuts down
Enter Sound Font Menu	Hold the button before the blade is ignited, then release it. Point the saber upwards while doing this.
∟ Go To Next Sound Font*	Slowly twist the saber
∟ Jump 10 Sound Fonts Further*	Tap the button
∟ Save And Leave Sound Font Menu	Hold the button
Show Battery Indicator	Hold the power button while performing a stab before the saber is ignited
∟ Calibrate Battery	When your battery is fully charged, perform a stab while battery level is displayed
Enter Volume Control	Hold the button before the blade is ignited, then release it. Point the saber downwards while doing this.
∟ Adjust Volume	Slowly twist the saber
∟ Save And Leave Volume Control	Tap the power button
Start Background Music	Hold the button while twisting the saber before the saber is ignited. Point the saber upwards while doing this.
∟ Skip Track	Quickly twist the saber
∟ Stop Background Music	Hold the button while twisting the saber before the saber is ignited
Enter Saber Editor	Hold the button while twisting the saber before the saber is ignited. Point the saber downwards while doing this.
∟ Select Editor Mode	Tap the button
∟ Color Profile	
∟ Select Color Profile	Slowly twist the saber
∟ Edit Color	Slowly twist the saber
∟ Go To Next Saber State	Tap the button
∟ Revert Colors To Default	Perform a stab
∟ Save Color Profile	Hold the button
∟ Confirm Selection	Tap the button
∟ Save and Leave Editor Mode	Hold the button
∟ Pixel Number	

∟ Edit Pixel Number	Slowly twist the saber
∟ Save and Leave Editor Mode	Hold the button
∟ Ignition Speed	
∟ Select Sound Font	Slowly twist the saber
∟ Edit Ignition Speed	Slowly twist the saber
∟ Save Ignition Speed	Hold the button
∟ Confirm Selection	Tap the button
∟ Save and Leave Editor Mode	Hold the button
∟ Retraction Speed	
∟ Select Sound Font	Slowly twist the saber
∟ Edit Retraction Speed	Slowly twist the saber
∟ Save Retraction Speed	Hold the button
∟ Confirm Selection	Tap the button
∟ Save and Leave Editor Mode	Hold the button
∟ Ignition Type	
∟ Select Blade Style	Slowly twist the saber
∟ Edit Ignition Type	Slowly twist the saber
∟ Save Ignition Type	Hold the button
∟ Confirm Selection	Tap the button
∟ Save and Leave Editor Mode	Hold the button
∟ Retraction Type	
∟ Select Blade Style	Slowly twist the saber
∟ Edit Retraction Type	Slowly twist the saber
∟ Save Retraction Type	Hold the button
∟ Confirm Selection	Tap the button
∟ Save and Leave Editor Mode	Hold the button
∟ Battle Configuration	
∟ Blaster Block Mode	
∟ Button and Swinging	Slowly twist the saber clockwise
∟ Button	Slowly twist the saber counterclockwise
∟ Force Push Mode	
∟ Button and Swinging	Slowly twist the saber clockwise
∟ Button	Slowly twist the saber counterclockwise
∟ Lockup Mode	
∟ Button and Clashing	Slowly twist the saber clockwise
∟ Button	Slowly twist the saber counterclockwise
∟ Save and Leave Editor Mode	Hold the power button
∟ Gesture Controls	
∟ Stab Ignition	
∟Turn stab ignition on	Slowly twist the saber clockwise
∟Turn stab ignition off	Slowly twist the saber counterclockwise
∟ Twist Ignition	
∟Turn twist ignition on	Slowly twist the saber clockwise
∟Turn twist ignition off	Slowly twist the saber counterclockwise
∟ Swing Ignition	
∟Turn swing ignition on	Slowly twist the saber clockwise
∟Turn swing ignition off	Slowly twist the saber counterclockwise
∟ Twist Retraction	
∟Turn twist retraction on	Slowly twist the saber clockwise

∟Turn twist retraction off	Slowly twist the saber counterclockwise
∟ Go To Next Gesture Control	Tap the power button
∟ Save and Leave Editor Mode	Hold the power button
∟ WiFi Setup	
∟ Select WiFi mode	Slowly twist the saber
∟ Confirm Selection	Hold the button
∟ Bluetooth Setup	
∟ Turn Bluetooth on	Slowly twist the saber clockwise
∟ Turn Bluetooth off	Slowly twist the saber counterclockwise
∟ Confirm Selection	Hold the button
∟ Hilt Calibration	
∟ Effect Font Slot	
∟ Select effect font slot	Slowly twist the saber
∟ Save and Leave Editor Mode	Hold the button
∟ Save And Leave Saber Editor	
∟ Confirm Selection	Hold the button
Toggle WiFi	Hold the button for at least 5 s
Toggle Bluetooth	Hold the button for at least 7.5 s
Reset	Hold the button for at least 10 s
Wake Up From Deep Sleep	Tap the button

*** When changing the On Top Effect, Background Effect or Sound Font, you can go to the next one by pointing your saber upwards during selection or to the previous one by pointing your saber downwards.**

Frequently Asked Questions

- There are no blaster or lockup effects in my Neopixel setup. How can I fix this?
  Answer: This is usually caused by the board not being setup for Neopixel light effects on your microSD card. Please follow the instructions in the "Choosing Your Light Effects" section.
- The board plays a sound indicating a missing SD card, even though the SD card is inserted. What should I do?
  Answer: Try reinserting the SD card. If the issue persists, reformat the card and reload the default files.
- The onboard LED, accent LEDs, or my blade is blinking. What does this mean?
  Answer: Blinking indicates the microSD card is not inserted properly, or files/folders are missing. Try reinserting the card. If the problem continues, reformat and reload the default files.
- There is no sound or the sound is stuttering. How can I fix this?
  Answer: Recharge your saber, as this issue often occurs when the battery is low.
- The speaker is crackling. What should I do?
  Answer: Lower the volume, either in the "general.txt" file or using the Volume Control feature on the fly.
- Some sound files are not played, or some sound fonts are not recognized. How can I fix this?
  Answer: Ensure that sound font folders and file names are numbered continuously and correctly. Refer to the manual for the correct naming conventions of each sound file.
- My computer does not recognize the microSD card as a mass storage device after connecting the Seedling v2 USB-C Module. What can I do?

Answer: Check the Device Manager for an "unknown device." Right-click on it and select "Uninstall device." Then reconnect the Seedling v2 USB-C Module. Ensure you are using Windows 10 or later.

- I can’t connect to WiFi, even though the GHv4 announces that the connection was successful.
  Answer: The GHv4 connects to 2.4 GHz WiFi only. To access it from your phone or computer, make sure your device is connected to the same 2.4 GHz network.

Appendices for Parameter Descriptions

Appendix A: In-Hilt Blade Style Parameters

This section provides detailed descriptions of the parameters of the in-hilt blade styles. While there are some differences between in-hilt and Neopixel blade styles, this appendix focuses on in-hilt blade styles.

Customizing In-Hilt Blade Style Parameters

As previously mentioned, in-hilt blade styles consist of an "effect1.txt" file and several "color[number].txt" files , which are labeled as color profiles. Both file types are organized into 10 blocks, each corresponding to one of the 10 saber states: ignition, basic, swing, clash, stab, spin, lockup, melt, blaster block, or force push. By adjusting the parameters within these blocks, you can customize the light effects and colors for each state.

This design allows for remarkable creativity - for instance, you can assign special light effects to a swing, such as a color change or a subtle flicker effect. You could simulate a blade that becomes unstable during a swing due to extra energy drawn from the diatium power cell, adding a unique and exotic feel to your saber. The possibilities for inventing custom light effects are virtually limitless! Almost all parameters follow the format "parameter=value1,value2,value3,value4", where:

- **value1** is the parameter value for LED channel 1,
- **value2** is for LED channel 2,
- **value3** is for LED channel 3, and
- **value4** is for LED channel 4.

This format allows each parameter to be set independently for each LED channel, enabling a vast range of different light effects to make your saber truly unique.

Customizing Colors

Within each saber state block, there is a parameter called "led" that defines the color by setting the relative power for each LED channel. The values range from 0 to 1023 for each channel:

- 0 means no power to that LED channel,
- 1023 means maximum power.

For example, if you set "led=0,300,600,1023", LED channel 1 receives no power, channel 2 gets a power value of 300, channel 3 receives 600, and channel 4 receives the maximum power of 1023. This allows you to create a diverse range of colors and effects tailored to your preferences. The parameter "led" of the basic state is shown below.

```
#SaberTec Customizer - Color Profile


#Basic Color
led=0,0,1023,1023
```

In other saber states, the "led" parameter includes a prefix indicating the specific state. For example, in the clash state, the parameter is called "cl_led".

Customizing Light Effects

Now, let's turn to the "effect1.txt" file, which contains all the parameters that define the light effects of the blade style. The parameters for the basic state are shown in the image below.

```
#SaberTec Customizer - Effect File
#This config file contains all the parameters defining the light effects of your saber.


#_____________________________
#
#Basic
#_____________________________


accent=0,0,0,1
delay=0,0,0,0

sync1=1,1,1,1
sync2=0,0,0,0
color_protection=0
flicker_min_intensity=0,0,0,0
flicker_max_intensity=0,0,0,0
flicker_min_delay=1,1,1,1
flicker_max_delay=5,5,5,5
flicker_min_duration=1,1,1,1
flicker_max_duration=1,1,1,1
flicker_period=0,0,0,0
flicker_phase_shift=0,0,0,0
flicker_type=0,0,0,0

pulse_type=1,1,1,1
pulse_shape=0,0,0,0
pulse_min_intensity=0,0,0,1023
pulse_max_intensity=0,0,0,1023
pulse_intensity_change_period=0,0,0,0
pulse_intensity_phase_shift=0,0,0,0
pulse_min_period=0,0,0,100
pulse_max_period=0,0,0,100
pulse_phase_shift=0,0,0,0
pulse_modulation_period=0,0,0,0
pulse_modulation_phase_shift=0,0,0,0
```

In the "effect1.txt" file, you can specify whether an LED channel drives a blade LED or an accent LED, and whether one of the LED channels should feature a delayed ignition. One of the fascinating aspects of our board is that the accent channels are fully customizable in the same way as the blade channels. You can apply the full range of light effects to accent LEDs independently of the blade LEDs if you wish!

This is particularly useful for creating unique effects in areas like a crystal chamber. With our board, you can design custom crystal chamber effects - from simple flickers or pulse effects to more advanced, exotic effects like color transitions.

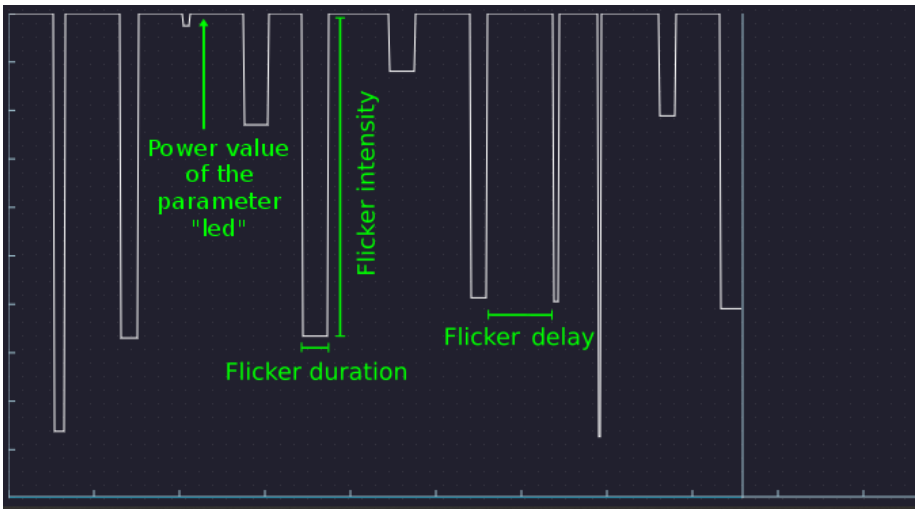
Flicker and Pulse Effects

The parameters of the basic state are structured into two parts: a flicker part and a pulse part. Let's first explore the flicker part, which controls the flicker effect. This effect produces random (or

deterministic, if desired) sudden drops in the power level defined by the "led" parameter. The flicker effect can be characterized by three key properties:

- **Flicker intensity**: Describes the depth of the power drops.
- **Flicker delay**: Defines the time between successive power drops.
- **Flicker duration**: Determines how long these power drops last.

The graphic below illustrates these three properties of the flicker effect, showing the power an LED channel receives over time (note that while "power" may not be the precise technical term, it best conveys the concept intuitively). As the graphic shows, the flicker intensity, delay, and duration can vary over time, which will be explained further in the next section.

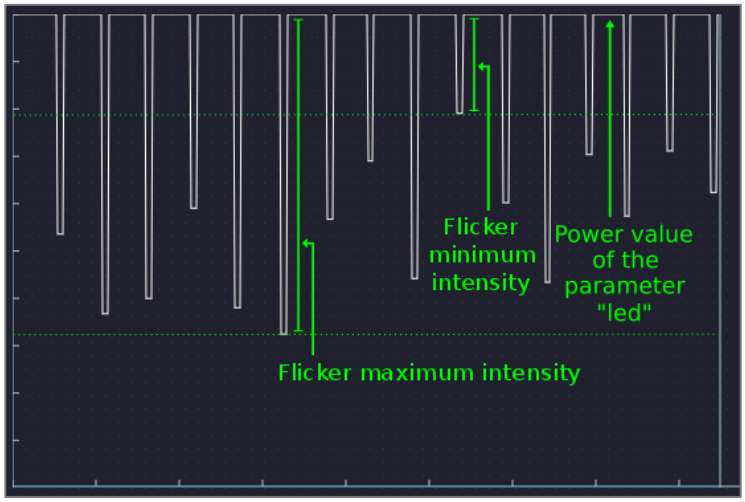


In the following, we provide an overview of the flicker part parameters, along with brief descriptions. The parameters are color-coded to indicate their intended user level:

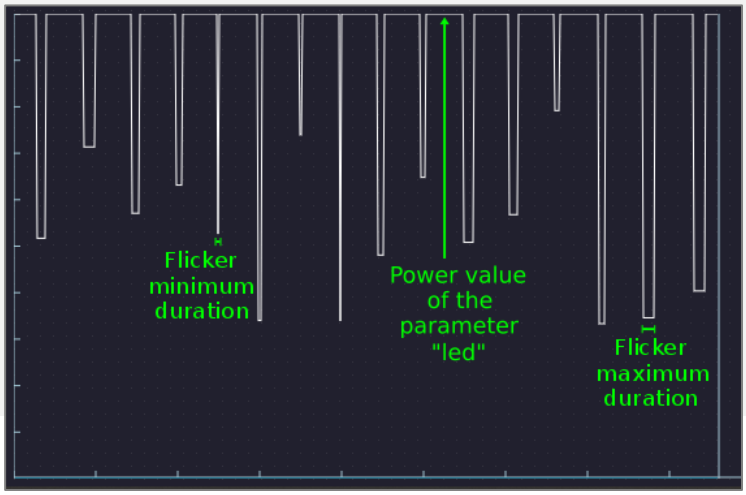
- **Black**: For users who want to make basic adjustments.
- **Green**: For advanced users looking to take full advantage of the effect engine.

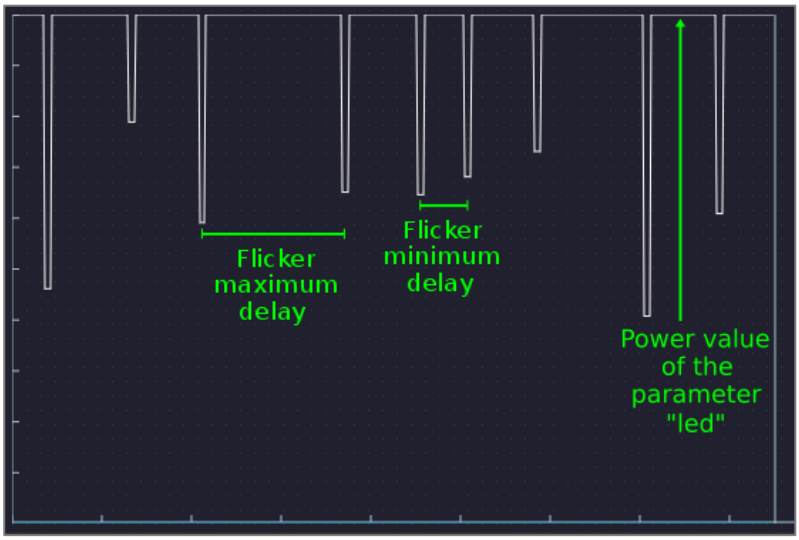
PARAMETER	DESCRIPTION
accent	Defines whether an LED channel is used for driving a blade LED or an accent LED. A value of 0 means that the respective LED channel is used for driving a blade LED whereas a value of 1 means that the respective LED channel is used for driving an accent LED. If you want to use the first three channels for the blade LED and the fourth channel for the lighting of a switch for example, you should choose "accent=0,0,0,1". The only difference between blade channels and accent channels is that the latter are already powered when the kill key is pulled, but the saber is not ignited yet.
flicker_min_intensity, flicker_max_intensity	Define the minimum and maximum flicker intensity (see description above). The actual flicker intensity varies randomly over time within the interval between these two values. The higher the flicker intensity, the stronger the flicker effect appears. Each LED channel has its own value. *Example:* *flicker_min_intensity = 200,0,600,200* *flicker_max_intensity = 700,0,600,1023* *In this case, the minimum flicker intensity of channel 1 is 200 whereas the maximum flicker intensity is 700 which means that the actual flicker intensity*

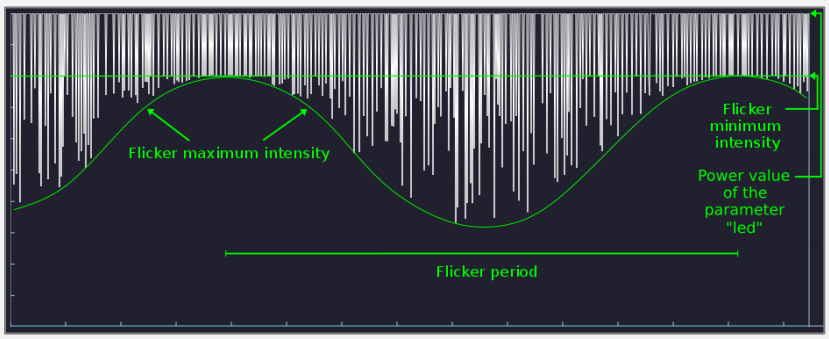
varies over time within the interval between 200 and 700. In LED channel 2, both the minimum and maximum flicker intensity is 0 which corresponds to a disabled flicker effect. The other channels are treated analogously.

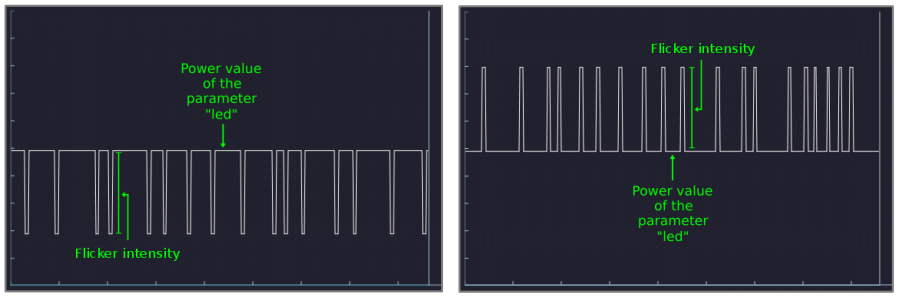
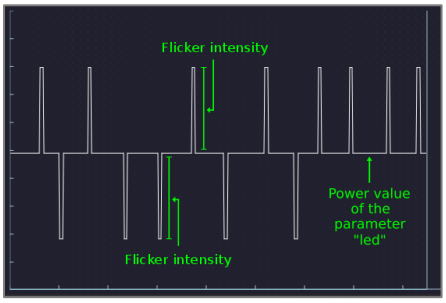


The picture above illustrates the meaning of both the "flicker_min_intensity" and the "flicker_max_intensity" parameter. It shows the change of the LED power over time using a flicker effect with the same intensity parameters as in channel 1 in the previous example. As you can see, the depth of each drop in the LED power varies randomly within the interval between the flicker minimum and maximum intensity, i.e. between 200 and 700 in this case.

flicker_min_duration, flicker_max_duration	Define the minimum and maximum flicker duration (see description above). The actual flicker duration varies randomly over time within the interval between these two values. The smaller the duration, the shorter the power drops are. Each LED channel has ist own value. *Example:* *flicker_min_duration = 1,1,5,1* *flicker_max_duration = 5,1,5,5* *In this case, the minimum flicker duration is 1 in LED channel 1 and the maximum flicker duration is 5. This means that the actual flicker duration varies randomly over time within the interval between 1 and 5. In LED channel 2, both* *the minimum and maximum flicker duration are 1. This means that every power drop has duration 1. The other channels are treated analogously.* 

	The picture above illustrates the meaning of both the "flicker_min_duration" and the "flicker_max_duration" parameter. It shows the change of the LED power over time using a flicker effect with the same duration parameters as in channel 1 in the previous example. As you can see, the duration of each drop in the LED power varies randomly within the interval between the flicker minimum and maximum intensity, i.e. between 1 and 5 in this case.
flicker_min_delay, flicker_max_delay	Define the minimum and maximum flicker delay (see description above). The actual flicker delay varies randomly within the interval between these two values. The smaller the flicker delay, the higher the speed of the flicker effect appears. Each LED channel has its own value. *Example:* *flicker_min_delay = 1,1,1,1* *flicker_max_delay = 5,1,5,10* *In this case, the minimum flicker delay is 1 in LED channel 1 and the maximum delay is 5. This means that the actual flicker delay varies randomly over time within the interval between 1 and 5. In LED channel 2, both the minimum and maximum flicker delay are 1. This means that the actual flicker delay is always 1.*  The picture above illustrates the meaning of both the "flicker_min_delay" and the "flicker_max_delay" parameter. It shows the change of the LED power over time using a flicker effect with the same delay parameters as in channel 1 in the previous example. As you can see, the delay between two successive drops in the LED power varies randomly within the interval between the minimum and maximum flicker delay, i.e. between 10 and 50 in this case. These values are usually too big for a conventional flicker effect and were only chosen that high for a better visualization.
flicker_period, flicker_phase_shift	The *maximum flicker intensity* changes smoothly over time back and forth (sine-like) between "flicker_min_intensity" and "flicker_max_intensity". Thus, the interval between the minimum and maximum flicker intensity changes over time. The parameter "flicker_period" describes how fast or slow this change occurs. The higher the value, the slower the transition happens. Ranges from 0 (no change over time at all) to 65535. When your saber is turned on, the

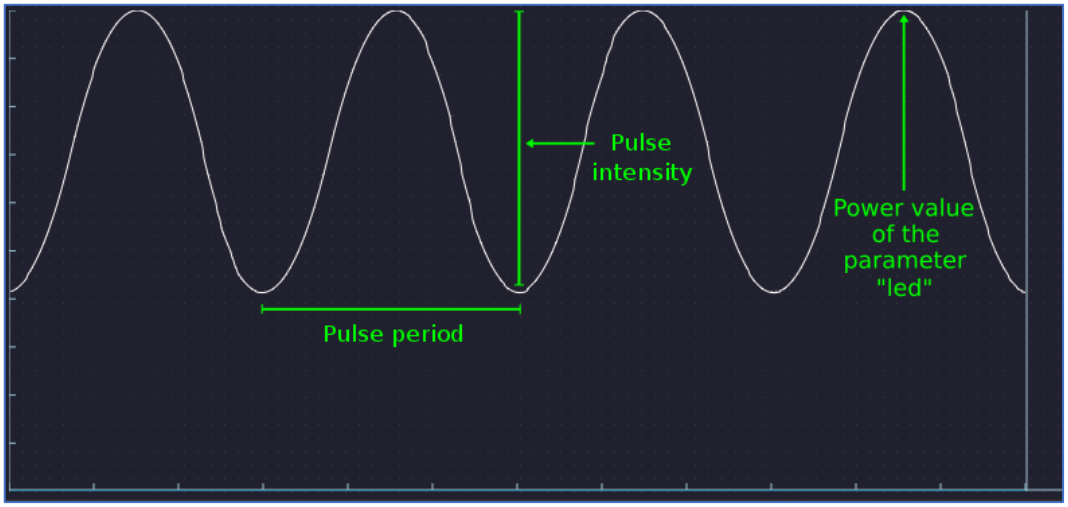
	maximum flicker intensity first has the value "flicker_max_intensity", before it decreases to the value "flicker_min_intensity". The parameter "flicker_phase_shift" can shift this behavior so that the maximum flicker intensity starts with the value "flicker_min_intensity" instead for example. Ranges from 0 (no shift) to 359 (maximum shift). We call the intensity transition effect PhaseFlicker. It can be disabled by setting "flicker_period" to 0. In this case, the maximum flicker intensity hast he value "flicker_max_intensity" and remains constant. We call the intensity transition effect PhaseFlicker. Each LED channel has ist own value. *Example:* *flicker_period = 600,0,300,1000* *flicker_phase_shift = 0,0,180,0* *In this case, the flicker period is 600 in LED channel 1 which means that the maximum flicker intensity varies over time between "flicker_min_intensity" and "flicker_maximum_intensity" with a transition duration of 600. This is especially interesting if you choose "flicker_minimum_intensity=0" as in this configuration, your saber would undergo a smooth transition between two phases: no flicker present and flicker present. This could simulate a saber that undergoes a transition between a stable and an unstable state for example. In LED channel 2, there is no smooth change in the flicker intensity over time (the flicker transition effect is disabled). Only the original random changes within the interval between "flicker_min_intensity" and "flicker_max_intensity" remain. The other channels are treated analogously.*  The picture above illustrates the meaning of the "flicker_period" parameter. It shows the change of the LED power over time using a flicker effect with the same "flicker period" parameter value as in channel 1 in the previous example. As you can see, the depth of each drop in the LED power varies randomly within the interval between the flicker minimum and maximum intensity, whereby the latter changes periodically between the parameter values "flicker_min_intensity" and "flicker_max_intensity".
sync1, sync2	As mentioned above, both the flicker duration and delay vary randomly over time between their respective minimum and maximum values defined by the corresponding parameters. This leads to the fact that even if you use the same duration and delay parameter values for different LED channels, the drops in the LED power don't necessarily occur at the same time. By using the "sync1" and "sync2" parameters, you can synchronize the LED channels such that the drops in the LED power always occur at the same time. Both parameters can be 0 or 1 for each channel. All LED channels with a value of 1 are synchronized.

	Example: *sync1=1,0,1,0* *sync2=0,1,0,1* *In this case, the LED channels 1 and 3 are synchronized as well as the LED channels 2 and 4. Now, you also see why there are two synchronization parameters: to cover all possible combinations, you need two. This can be interesting for a crystal chamber for example. By using the synchronization parameters, you could both synchronize the blade LED channels and the chamber LED channels.*
color_protection	Defines whether or not the LED channels 1, 2 and 3 are fully synchronized. By setting the "color_protection" parameter to 1, the LED channels 2 and 3 follow the same behavior as LED channel 1. This can be useful when driving an in-hilt LED for which you bridged the LED channels 1, 2 and 3 for example. A value of 0 disables the synchronization and a value of 1 enables it. This parameter has only one value for all LED channels.
flicker_type	Defines the type of the flicker effect. A flicker effect cannot only create sudden power *drops* which is called "subtractive type", but also sudden power *peaks* which is called "additive type". Furthermore, a flicker effect also can mix both types by creating sudden power drops and peaks in a random manner which is called "mixed type". A value of 0 corresponds to the subtractive type, a value of 1 to the additive type whereas a value of 2 corresponds to the mixed type. Each LED channel has its own value. *Example:* *flicker_type = 0,1,0,0* *In this case, channel 1 displays a flicker of type 0 which corresponds to a subtractive type whereas channel 2 displays a flicker of type 1 which corresponds to an additive type. An additional flicker type can be especially interesting as on top clash effect or to enlighten a crystal chamber. By using it, you could configure a "chamber flash" that creates random light bursts. The other channels are treated analogously.* subtractive type (0) additive type (1)  mixed type (2) 

The pictures above illustrate the meaning of the "flicker type" parameter. It shows the change of the LED power over time using a subtractive, additive or mixed flicker effect, respectively. As you can see, a subtractive flicker effect creates drops in the LED power whereas an additive one creates peaks and a mixed one creates both drops and peaks. For the additive and the mixed flicker effect, all the other parameters described above have the same meaning as for the subtractive flicker effect.

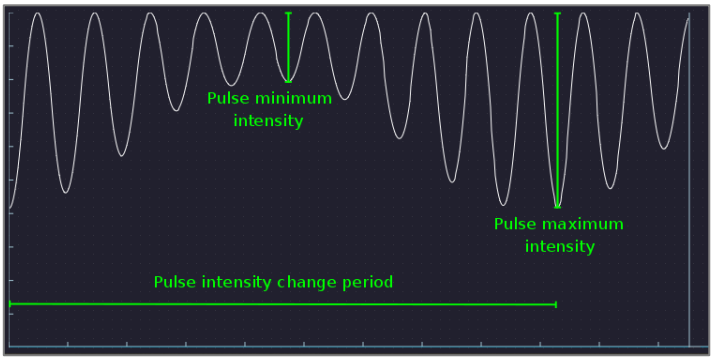
In the following, we'll turn to the pulse part of the basic light effects. It's situated below the flicker part and defines – as its name suggests – a pulse effect. In contrast to the flicker effect, the pulse effect creates periodic and smooth drops (sine-like) in the LED power that is defined by the parameter "led". It can be characterised by two basic properties: the pulse intensity and the pulse period. Hereby, the pulse intensity describes how deep the pulse effect smoothly digs into the LED power. In addition to that, the pulse period describes the duration of one drop. Thus, it is a measure for the pulse speed. The smaller the values of the pulse period, the faster the LED pulses.

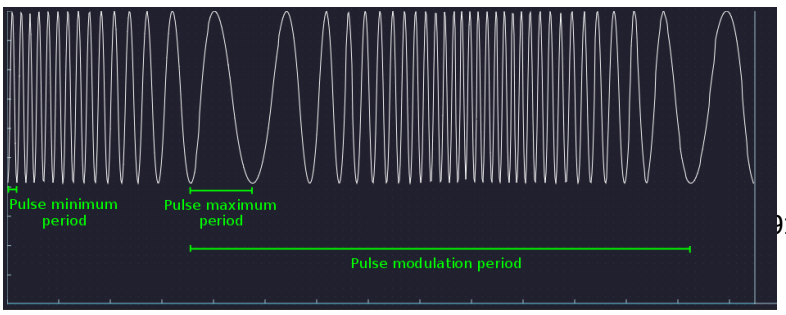
The picture below illustrates these two characteristic properties of the pulse effect graphically. It shows the power an LED gets over time from one LED channel using a pulse effect. Both pulse intensity and period can also vary over time which will become clear in the next section.

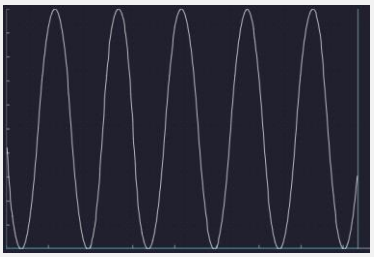
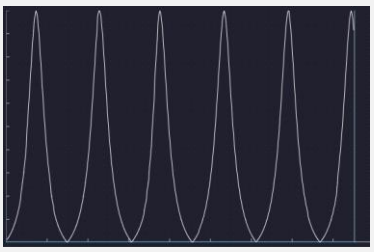
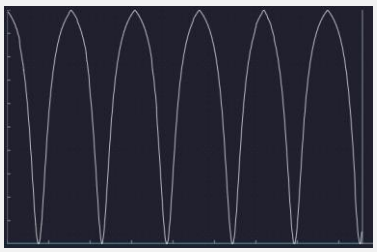
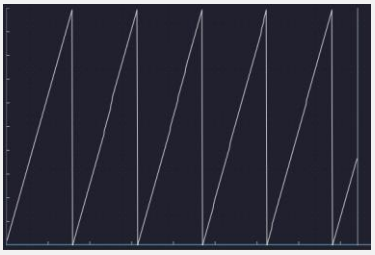
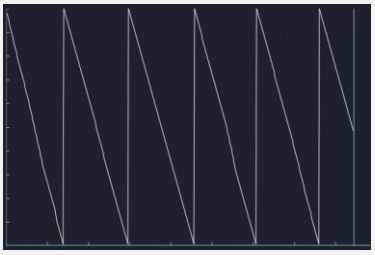


In the following, we give a list of all the parameters of the pulse part and a short description. Similarly to the flicker part, the parameters are marked by different colors. These indicate whether they're for users who only want to change a few parameters (black) or for more advanced users who want to unleash the full-blown effect engine (green).

PARAMETER	DESCRIPTION
pulse_min_intensity, pulse_max_intensity	Define the minimum and maximum pulse intensity. The actual pulse intensity changes over time periodically within the interval between "pulse_min_intensity" and "pulse_max_intensity" (see picture below). We call this intensity change effect *"ÜberPulse"*. It can be disabled by setting "pulse_min_intensity" and "pulse_max_intensity" to the same value. By doing so, one obtains a conventional pulse effect with a constant pulse intensity. Each LED channel has its own value. *Example:*

	pulse_min_intensity=200,300,0,0 *pulse_max_intensity=600,300,1023,0* *In this case, LED channel 2 shows a pulse effect with a minimum intensity of 200 and a maximum intensity of 600. Thus, drops in the LED power are periodically created with a depth that changes over time repetitively back and forth within the interval between 200 and 600. By that, you can create a pulse effect that starts almost invisible and gets stronger and stronger over time for example. LED channel 1 shows a pulse effect with an intensity of 300 which means that the pulse effect periodically digs into the LED power down to a depth of 300. The other LED channels are treated analogously.*
pulse_intensity_change_period, pulse_intensity_phase_shift	The parameter "pulse_intensity_change_period" defines how long the periodic change of the pulse intensity from the minimum to the maximum value and back again takes (see picture below). Thus, it is a measure for how fast the transition between a pulse with intensity "pulse_min_intensity" and one with intensity "pulse_max_intensity" occurs. The smaller the value of "pulse_intensity_change_period", the faster the transition occurs. If you turn your saber on and if you have set a pulse effect with an intensity transition, the pulse will begin with its maximum intensity, before changing to its minimum intensity (see picture below). You can change this behavior by adjusting the parameter "pulse_intensity_phase_shift". By changing its value, you can shift the transition such that the pulse begins with its minimum intensity instead of its maximum intensity for example. Each LED channel has its own value. *Example:* *pulse_intensity_change_period=600,0,1000,0* *pulse_intensity_phase_shift=0,0,180,0* *In this case, LED channel 1 shows a pulse intensity change with a period of 600. In LED channel 2, the pulse intensity change is disabled as a value of 0 means that the pulse intensity doesn't change over time. Its constant value is "pulse_maximum_intensity". The other LED channels are treated analogously.*  The picture above illustrates the meaning of the "pulse_min_intensity", "pulse_max_intensity" and "pulse_intensity_change_period" parameter. It shows the change of the LED power over time using a pulse effect with the same parameter value as in channel 1 in the

	previous two examples. As you can see, the actual pulse intensity, i.e the depth of each drop in the LED power varies smoothly within the interval between the minimum and maximum pulse intensity.
pulse_min_period, pulse_max_period	Define the minimum and maximum period of the pulse effect. The actual period of the pulse effect changes periodically within the interval between these two values. As mentioned above, the pulse period is a measure for the pulse speed. The smaller the pulse period, the higher the pulse speed. Thus, you can create a pulse whose speed changes periodically over time. We call this speed transition effect HyperPulse. You can disable it by setting both parameters "pulse_min_period" and "pulse_max_period" to the same value. This creates a conventional pulse effect. Each LED channel has its own value. *Example:* *pulse_min_period=100,100,1000,1000* *pulse_max_period=100,1000,1000,1000* *In this case, the pulse period is constant at a value of 100 in LED channel 1 as both minimum and maximum period have the same value. LED channel 2 shows a pulse with a period that changes back and forth within the interval between 100 and 1000. Thus, you get a pulse that periodically decreases and increases its speed. This can create an amazing look as it simulates a blade that undergoes a transition between a calm and an aggressive phase. The other LED channels are treated analogously.*
pulse_modulation_period, pulse_modulation_phase_shift	The parameter "pulse_modulation_period" defines how long the periodic change between the minimum and maximum period, i.e. the maximum and minimum speed takes. Thus, it is a measure for how fast this speed transition occurs. The smaller the value of "pulse_modulation_period", the faster the transition occurs. If you turn your saber on and if you have set a pulse effect with speed transition, it starts with its minimum period, before continously changing to its maximum period (see picture below). Each LED channel has its own value. *Example:* *pulse_modulation_period=1000,10000,0,0* *pulse_modulation_phase_shift=0,180,0,0* *In this case, the actual pulse period changes within the interval between "pulse_minimum_period" and "pulse_maximum_period" with a transition duration of 1000 in LED channel 1. LED channel 2 shows a pulse with a speed transition duration of 10000 and a phase shift of 180. This means that the pulse starts with ist maximum period instead of its minimum period. The other LED channels are treated analogously.* 

	The picture above illustrates the meaning of the "pulse_min_period", "pulse_max_period" and the "pulse_modulation_period" parameter. It shows the change of the LED power over time using a pulse effect with the same parameter values as in channel 1 in the previous two examples. As you can see, the pulse speed varies smoothly within the interval between the minimum and maximum pulse speed.
pulse_shape	Defines the shape of the periodic, smooth drops into the LED power that the pulse effect creates. A value of 0 corresponds to a sine shape (which was used in the pictures above), a value of 1 to an impulse shape, a value of 2 to an inverted impulse shape, a value of 3 to a rising sawtooth shape, a value of 4 to a falling sawtooth shape and a value of 5 to a rectangular shape (see the picture below). Each LED channel has its own value. *Example:* *pulse_shape=0,1,0,0* *In this case, the pulse effect in LED channel 1 creates sine shaped periodic, smooth drops in the LED power. LED channel 2 shows a pulse effect that creates impulse shaped periodic, smooth drops in the LED power. The other LED channels are treated analogously.* sine shape (0)  rectangular shape (5)  impulse shape (1)  inverted impulse shape (2)  rising sawtooth (3)  falling sawtooth (4) 

	The picture above illustrates the meaning of the "pulse_shape" parameter. It shows the change of the LED power over time using a pulse effect with sine, impulse, inverted impulse, rising sawtooth, falling sawtooth or rectangular shape, respectively. For the pulse shapes different than sine, all the other parameters described above have the same meaning as for the sine shaped pulse effect.

So far, these are all parameters that define the light effects of the basic state. As mentioned above, the other states, namely ignition, clash, stab, swing, spin, lockup, melt, blaster block and force push have similar parameters, except from the "accent" and "delay" parameters. They need to be set only in the basic state. Therefore, the other saber states do not contain them. For all of the additional states, there are two more parameters each in comparison to the basic state: effect duration and fade out. Let's have a look at the clash state for example. All the clash parameters are shown below.

```
#_____________________________
#
#Clash
#_____________________________


cl_sync1=1,1,1,1
cl_sync2=0,0,0,0
cl_color_protection=0
cl_flicker_min_intensity=512,512,512,0
cl_flicker_max_intensity=512,512,512,0
cl_flicker_min_delay=1,1,1,1
cl_flicker_max_delay=1,1,1,1
cl_flicker_min_duration=1,1,1,1
cl_flicker_max_duration=1,1,1,1
cl_flicker_period=0,0,0,0
cl_flicker_phase_shift=0,0,0,0
cl_flicker_type=0,0,0,0

cl_pulse_type=1,1,1,1
cl_pulse_shape=0,0,0,0
cl_pulse_min_intensity=0,0,0,1023
cl_pulse_max_intensity=0,0,0,1023
cl_pulse_intensity_change_period=0,0,0,0
cl_pulse_intensity_phase_shift=0,0,0,0
cl_pulse_min_period=0,0,0,100
cl_pulse_max_period=0,0,0,100
cl_pulse_phase_shift=0,0,0,0
cl_pulse_modulation_period=0,0,0,0
cl_pulse_modulation_phase_shift=0,0,0,0

cl_duration=100
cl_fade_out=60
```

As you can see, you have the same degree of customizability just as in the basic state, whereby the parameters here have the additional prefix "cl_" which indicates that they belong to the clash state. In addition to these settings you also have the parameters "cl_duration" and "cl_fade_out". The parameter "cl_duration" describes how long the clash light effects are shown, given relatively to the length of the clash sounds. It ranges from 0 to 100, which corresponds to a duration of 0 % of the clash sounds (disabled clash light effects) or a duration of 100 % of the clash sounds (light effects are shown until the end of the clash sounds is reached), respectively. This is an amazingly powerful customization parameter. On the one hand, you could create light effects that are very short like an epic, sudden flash when a clash is triggered for example. On the other hand, you could create light effects that maintain very long which could simulate a blade that is strongly affected by the clash and takes a long time to stabilize again. Here, you have again the full freedom in designing your very own, unique light effects as you could also choose values of 70 % for example or any other value within the respective range. The parameter "cl_fade_out" forms a congenial feature together with the clash duration. When a clash is triggered and the clash light effects occur, they start to smoothly fade out to the basic light effects. This is done by mixing continuously more and more of the basic effects to the clash effects. The parameter "cl_fade_out" describes how fast this transition between the clash and basic effects occurs. It is given relatively to the duration of the clash light effects and ranges from 0 to 100. A value of 0 corresponds to a duration of 0 % of the clash light effects duration (disabled fade out, the clash effects are immediately interrupted by the basic effects when they end). A value of 100 corresponds to a duration of 100 % of the clash light effects duration (smoothest transition possible). Using the fade out parameter, you can create flabbergasting effects. If you have a red blade with a white LED flashing on clash and if you set a fade out for example, the blade color smoothly changes from white over white-red to red again as the blade stabilizes after the clash.

The other states, namely, ignition, stab, swing, spin, lockup, melt, blaster block and force push also have a duration and fade out parameter with the very same functionality as in the clash state each. The only thing that is different between these states is the prefix of the parameters.

Appendix B: Neopixel Blade Style Parameters

As previously mentioned, Neopixel blade styles consist of a "background.txt" file, multiple "ontop[number].txt" files, and several "color[number].txt" files, which are referred to as color profiles. The Neopixel light effects engine of the Golden Harvest v4 uses these two effect types to offer unparalleled customization.

You can think of background effects and on top effects as different layers. The background effect defines basic characteristics, such as flicker or pulse, while the on top effect adds additional layers, like an unstable blade effect. For example, combining a slightly flickering red blade as the background with an unstable blade effect as the on top layer will create a flickering blade with an unstable appearance.

In general, on top effects alter the spatial structure of the underlying background effect. Both background and on top effect files contain 10 blocks of parameters, each corresponding to one of the 10 saber states: ignition, basic, swing, clash, stab, spin, lockup, melt, blaster block, or force push. By adjusting the parameters in each block, you can fully customize the light effects for the corresponding saber state.

Since the parameter set is similar for all states, we will use the basic state as an example. This is the state when no motion is triggered. Let's begin with the "background.txt" file, whose structure is shown below.

```
#_____________________________
#
#Basic Background Effect
#_____________________________


############
#Classic
############
bg=0

led=1023,0,0,1023

sync1=1,1,1,1
sync2=0,0,0,0
color_protection=0
flicker_min_intensity=0,0,0,0
flicker_max_intensity=256,256,256,0
flicker_min_delay=1,1,1,1
flicker_max_delay=5,5,5,5
flicker_min_duration=1,1,1,1
flicker_max_duration=1,1,1,1
flicker_period=0,0,0,0
flicker_phase_shift=0,0,0,0
flicker_type=0,0,0,0

pulse_shape=0,0,0,0
pulse_min_intensity=0,0,0,1023
pulse_max_intensity=0,0,0,1023
pulse_intensity_change_period=0,0,0,0
pulse_intensity_phase_shift=0,0,0,0
pulse_min_period=0,0,0,250
pulse_max_period=0,0,0,250
pulse_phase_shift=0,0,0,0
pulse_modulation_period=0,0,0,0
pulse_modulation_phase_shift=0,0,0,0
```

In this file, you can find the parameters that describe the background effect. Comments are indicated by hashtags and are ignored by the board when reading out the parameters contained in the file. Below the headline "Background Effect" there is a headline describing which background effect was chosen in the respective state. In this example, the background effect "Classic" was chosen which corresponds

to the light effects of the in-hilt setup. However, this headline is only a guide to the eye for the user. The board recognizes that this background effect is selected by reading out the parameter "bg=0".

Similarly to the in-hilt setup, the other states, namely ignition, clash, stab, swing, spin, lockup, melt, blaster block and force push have exactly the same parameters. Thus, they can be customized in the same manner as the basic state. For all of these additional states, there are even more parameters in comparison to the basic state. These parameters customize the fading of the light effects of these states back to the light effects of the basic state. Let's have a look at the clash state for example. All the clash parameters are shown below.

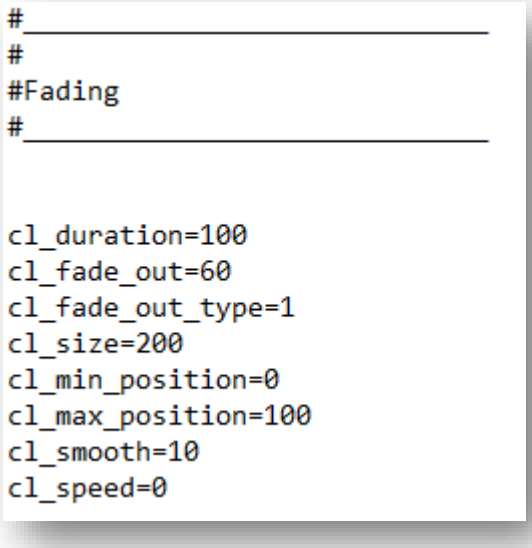
```
#_____________________________
#
#Clash Background Effect
#_____________________________


############
#Classic
############
cl_bg=0

cl_led=1023,0,1023,1023

cl_sync1=1,1,1,1
cl_sync2=0,0,0,0
cl_color_protection=0
cl_flicker_min_intensity=512,512,512,0
cl_flicker_max_intensity=512,512,512,0
cl_flicker_min_delay=1,1,1,1
cl_flicker_max_delay=1,1,1,1
cl_flicker_min_duration=1,1,1,1
cl_flicker_max_duration=1,1,1,1
cl_flicker_period=0,0,0,0
cl_flicker_phase_shift=0,0,0,0
cl_flicker_type=0,0,0,0

cl_pulse_shape=0,0,0,0
cl_pulse_min_intensity=0,0,0,1023
cl_pulse_max_intensity=0,0,0,1023
cl_pulse_intensity_change_period=0,0,0,0
cl_pulse_intensity_phase_shift=0,0,0,0
cl_pulse_min_period=0,0,0,250
cl_pulse_max_period=0,0,0,250
cl_pulse_phase_shift=0,0,0,0
cl_pulse_modulation_period=0,0,0,0
cl_pulse_modulation_phase_shift=0,0,0,0
```



As you can see, you have the same degree of customizability just as in the basic state, whereby the parameters here have the additional prefix "cl_" which indicates that they belong to the clash state. In addition to these settings you also have the parameters "cl_duration" and "cl_fade_out" among other parameters. These parameters are summarized into a "Fading" section. The parameter "cl_duration" describes how long the clash light effects are shown, given relatively to the length of the clash sounds. It ranges from 0 to 100, which corresponds to a duration of 0 % of the clash sounds (disabled clash light effects) or a duration of 100 % of the clash sounds (light effects are shown until the end of the clash sounds is reached), respectively. This is an amazingly powerful customization parameter. On the one hand, you could create light effects that are very short like an epic, sudden flash when a clash is triggered for example. On the other hand, you could create light effects that maintain very long which could simulate a blade that is strongly affected by the clash and takes a long time to stabilize again. Here, you have again the full freedom in designing your very own, unique light effects as you could also choose values of 70 % for example or any other value within the respective range. The parameter "cl_fade_out" forms a congenial feature together with the clash duration. When a clash is trigge red and the clash light effects occur, they start to smoothly fade out to the basic light effects. This is done by mixing continuously more and more of the basic effects to the clash effects. The parameter "cl_fade_out" describes how fast this transition between the clash and basic effects occurs. It is given relatively to the duration of the clash light effects and ranges from 0 to 100. A value of 0 corresponds to a duration of 0 % of the clash light effects duration (disabled fade out, the clash effects a re immediately interrupted by the basic effects when they end). A value of 100 corresponds to a duration of 100 % of the clash light effects duration (smoothest transition possible). Using the fade out parameter, you can create flabbergasting effects. If you have a red blade with a white LED flashing on clash and if you set a fade out for example, the blade color smoothly changes from white over white-red to red again as the blade stabilizes after the clash.

Let's now consider the next parameter of the "Fading" section. The parameter "cl_fade_out_type" specifies the style of the fading. In the following, the meaning of the different values of this parameters is given:

- cl_fade_out_type=0: During the fading, the clash light effects are localized. That means that only a customizable section of the blade displays the clash light effects

and lets the basic light effects shine through more and more. This section can be customized by the parameters "cl_size", "cl_min_position", "cl_max_position" and "cl_smooth". Hereby, the parameter "cl_size" defines the size of the section of the blade that displays the clash light effects. The value are given as number of LEDs. The position of the section of the blade that displays the clash light effects varies randomly between "cl_min_position" and "cl_max_position". The smoothness of the transition between the section that displays the clash light effects and the rest of the blade is given by "cl_smooth".

- cl_fade_out_type=1: During the fading, the clash light effects dissolve and let the basic light effects shine through more and more. This is the unstable fading setting 1.

- cl_fade_out_type=2: During the fading, the clash light effects dissolve and let the basic light effects shine through more and more. This is the unstable fading setting 2.

- cl_fade_out_type=3: During the fading, the clash light effects dissolve and let the basic light effects shine through more and more. This is the unstable fading setting 3.

- cl_fade_out_type=4: During the fading, the clash light effects dissolve and let the basic light effects shine through more and more. This is the unstable fading setting 4.

- cl_fade_out_type=5: During the fading, the clash light effects dissolve and let the basic light effects shine through more and more. This is the unstable fading setting 5.

- cl_fade_out_type=6: During the fading, the clash light effects are localized and their position moves periodically from the bottom of the blade to the tip and let the basic light effects shine through more and more. This is the running fading setting 1.

- cl_fade_out_type=7: During the fading, the clash light effects are localized and their position moves periodically from the bottom of the blade to the tip and let the basic light effects shine through more and more. This is the running fading setting 2.

- cl_fade_out_type=8: During the fading, the section of the blade that displays the clash light effects behaves like a flame blade and lets the basic light effects shine through more and more. This is the flame blade fading setting 1.

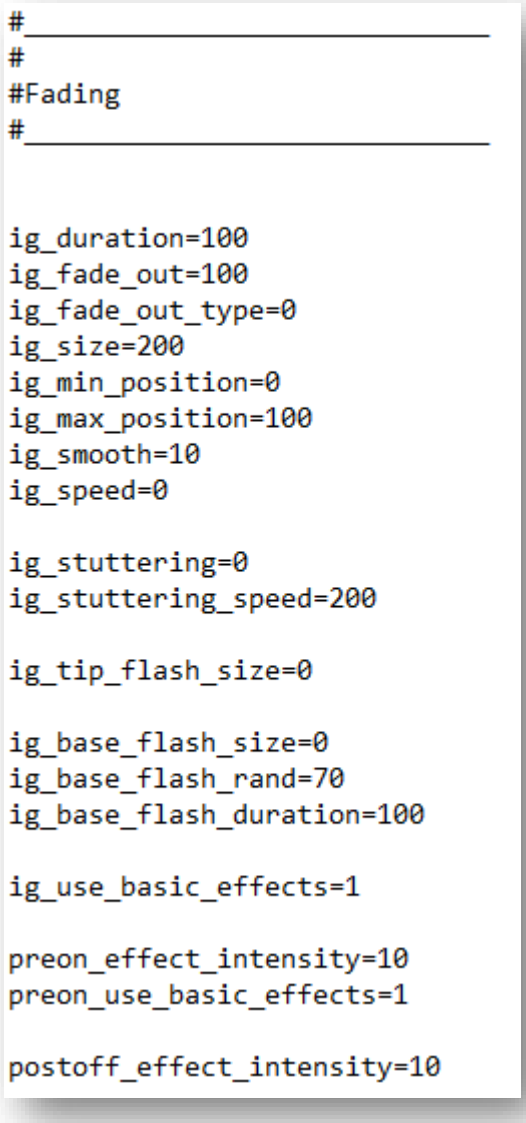
- cl_fade_out_type=9: During the fading, the section of the blade that displays the clash light effects behaves like a flame blade and lets the basic light effects shine through more and more. This is the flame blade fading setting 2.

- cl_fade_out_type=10: During the fading, the clash light effects are displayed along the whole blade and let the basic light effects shine through more and more.

- cl_fade_out_type=11: During the fading, the section of the blade that displays the clash light effects moves up and down the blade and lets the basic light effects shine through more and more.

There is also the parameter "cl_speed" which needs to be used together with "cl_fade_out_type=0" and a localized impact. For a value of the parameter "cl_speed" larger than 0, the impact splits into two parts which start to move both towards the bottom and towards the tip of the blade at the speed defined by this parameter.

So far, we discussed all parameters of the "Fading" section, except from some parameters that are only available for the ignition state as they allow to customize the ignition effect. In the image below, the fading parameters of the ignition state are shown. In addition to the parameters we already know from the other states, there are a series of parameters that we will describe in the following.



```
#_____________________________
#
#Fading
#_____________________________


ig_duration=100
ig_fade_out=100
ig_fade_out_type=0
ig_size=200
ig_min_position=0
ig_max_position=100
ig_smooth=10
ig_speed=0

ig_stuttering=0
ig_stuttering_speed=200

ig_tip_flash_size=0

ig_base_flash_size=0
ig_base_flash_rand=70
ig_base_flash_duration=100

ig_use_basic_effects=1

preon_effect_intensity=10
preon_use_basic_effects=1

postoff_effect_intensity=10
```

PARAMETER	DESCRIPTION
ig_stuttering	Defines whether or not the stuttering ignition effect should be used. A value of 0 disables this ignition effect, a value of 1 enables it using mode 1 and a value of 2 enables it using mode 2. If it is enabled, additional ignition effects that may be set in the "Fading" section are ignored. When this effect is activated, it simulates a blade that assembles itself from pieces during ignition. *Example:* *ig_stuttering=1* *In this case, the stuttering effect is enabled.*
ig_stuttering_speed	Defines the speed at which the blade assembles itself from pieces during ignition if the stuttering effect is enabled using mode 1. *Example:* *ig_stuttering_speed=200* *In this case, the speed of the stuttering effect is 200.*
ig_tip_flash_size	The tip flash effect is an ignition effect. When this effect is activated, it creates a white tip that moves from the bottom of the blade to its top when it is ignited. The color can be customized, too. This parameter defines the size of the dot that moves from the bottom of the blade to its top when it is ignited. The tip flash effect can be deactivated by choosing a value of 0. *Example:* *ig_tip_flash_size=1* *In this case, the size of the dot that moves from the bottom of the blade to its top is 1.*
ig_tip_flash_color	The tip flash effect is an ignition effect. When this effect is activated, it creates a white tip that moves from the bottom of the blade to its top when it is ignited. The color can be customized, too. This parameter defines the color of the dot that moves from the bottom of the blade to its top when it is ignited. *Example:* *ig_tip_flash_color=255,255,255* *In this case, the color of the dot that moves from the bottom of the blade to its top is white.*
ig_base_flash_size	The base flash effect is an ignition effect. When this effect is activated, it creates a white flash at the bottom of the blade when it is ignited that simulates an explosive ignition. The color can be customized, too. This parameter defines the size of the flash at the bottom of the blade when it is ignited. The bottom flash effect can be deactivated by choosing a value of 0. *Example:* *ig_base_flash_size=50* *In this case, the size of the flash at the bottom of the blade is 50.*
ig_base_flash_rand	The base flash effect is an ignition effect. When this effect is activated, it creates a white flash at the bottom of the blade when it is ignited that simulates an explosive ignition. The color can be customized, too. This

	parameter defines the aggressivity of the flash at the bottom of the blade when it is ignited. *Example:* *ig_base_flash_rand=70* *In this case, the aggressivity of the flash at the bottom of the blade is 70.*
ig_base_flash_duration	The base flash effect is an ignition effect. When this effect is activated, it creates a white flash at the bottom of the blade when it is ignited that simulates an explosive ignition. The color can be customized, too. This parameter defines the duration of the flash at the bottom of the blade when it is ignited. The value is given as percentage of the ignition duration. *Example:* *ig_base_flash_duration=50* *In this case, the flash at the bottom of the blade is displayed for the first 50% of the duration of the ignition.*
ig_base_flash_color	The base flash effect is an ignition effect. When this effect is activated, it creates white flash at the bottom of the blade when it is ignited that simulates an explosive ignition. The color can be customized, too. This parameter defines the color of the flash at the bottom of the blade when it is ignited. *Example:* *ig_base_flash_color=255,255,255* *In this case, the color of the flash at the bottom of the blade is white.*
ig_use_basic_effects	Defines whether you want to use custom ignition effects or the basic light effects should be displayed during ignition. A value of 0 means that the custom ignition effects are used and a value of 1 means that the basic light effects are used. *Example:* *ig_use_basic_effects=0* *In this case, the custom ignition effects are used.*
ig_center	The center effect is an ignition effect that modifies how the blade is extended. A value of 0 creates a blade that ignites from the bottom to the tip this effect, a value of 1 creates a blade that ignites from the center of the blade and extends outwards without rescaling it, a value of 2 creates a blade that ignites from outwards and extends towards the center of the blade without rescaling it, a value of 3 creates a blade that ignites from the center of the blade and extends outwards with rescaling it, a value of 4 creates a blade that ignites from outwards and extends towards the center of the blade with rescaling it and a value of 5 creates a blade that ignites from the top to the bottom. *Example:* *ig_center=1* *In this case, the blade ignites from the center oft he blade and extends outwards without rescaling it.*

preon_effect_intensity	Defines the intensity of the preon light effects. The higher the value the stronger the preon light effect will be. A value of 0 disables the preon light effects.
preon_use_basic_effects	Defines if you want to use the basic light effects as the preon light effects or the ignition light effects. A value of 0 means that the ignition light effects are used and a value of 1 means that the basic light effects are used.
postoff_effect_intensity	Defines the intensity of the postoff light effects. The higher the valuethe stronger the postoff light effect will be. A value of 0 disables the preon light effects.

Now, let's come to the file containing the parameters of the on top effect. Similarly to the background effect file, a headline indicates which on top effect was chosen. In this example, the on top effect "Kylo Blade" was chosen which creates a variation of an unstable blade. However, this headline is only a guide to the eye for the user. The board recognizes that this on top effect is selected by reading out the parameter "ot=2".

```
#_____________________________
#
#Basic On Top Effect
#_____________________________


############
#Kylo Blade
############
ot=2

kylo_blade_strength=75
kylo_blade_type=0
kylo_blade_range=10
kylo_blade_density=100
kylo_blade_smoothness=100
kylo_blade_dissipation=100
```

In the following, we will describe all the different background effects and on top effects. Let's first consider the background effects. There are 6 different effect categories: Classic, Color Flow, Frozen Pulse, Wave, Flame Blade and Gradient. All of them will be described in detail in the next sections.

Classic

The background effect Classic basically extends the light effects of the In-Hilt RGB setup to Neopixel. Therefore, you can find an extensive description of the parameters in the "In-Hilt" section. All the parameters of the background effect Classic are of the form "parameter=value1,value2,value3,value4" with each value representing a different color. Value 1 stands for the red channel, value 2 stands for the green channel and value 3 stands for the blue channel. Currently, value 4 does not represent any color channel.

Color Flow

The background effect Color Flow generates a smooth sequence of colors that enlighten the blade emerging from its lower end and moving ahead to its tip with a configurable speed. Also the color sequence and the width of each color section as well as the smoothness of the transition between the different color sections can be customized. The parameters of the background effect Color Flow are listed below.

PARAMETER	DESCRIPTION
color_flow_color1	Defines the first color of the color sequence in RGB code. When your saber is ignited and a Color Flow effect is active, this is the first color that will emerge from the lower end of your blade and move to its tip. *Example:* *color_flow_color1=255,0,0* *In this case, the first color of the color sequence is red which is given by the RGB value (255, 0, 0).*
color_flow_color2	Defines the second color of the color sequence in RGB code. When your saber is ignited and a Color Flow effect is active, this is the second color that will emerge from the lower end of your blade and move to its tip. *Example:* *color_flow_color2=0,255,0* *In this case, the first color of the color sequence is green which is given by the RGB value (0, 255, 0).*
color_flow_color3	Defines the third color of the color sequence in RGB code. When your saber is ignited and a Color Flow effect is active, this is the third color that will emerge from the lower end of your blade and move to its tip. *Example:* *color_flow_color3=0,0,255* *In this case, the first color of the color sequence is blue which is given by the RGB value (0, 0, 255).*
color_flow_color4	Defines the fourth color of the color sequence in RGB code. When your saber is ignited and a Color Flow effect is active, this is the fourth color that will emerge from the lower end of your blade and move to its tip. *Example:* *color_flow_color5=255,255,0* *In this case, the fourth color of the color sequence is yellow which is given by the RGB value (255, 255, 0).*
color_flow_color5	Defines the fifth color of the color sequence in RGB code. When your saber is ignited and a Color Flow effect is active, this is the fifth color that will emerge from the lower end of your blade and move to its tip. *Example:* *color_flow_color5=0,255,255* *In this case, the fifth color of the color sequence is cyan which is given by the RGB value (0, 255, 255).*
color_flow_color6	Defines the sixth color of the color sequence in RGB code. When your saber is ignited and a Color Flow effect is active, this is the sixth color that will emerge from the lower end of your blade and move to its tip.

	Example: *color_flow_color6=255,0,255* *In this case, the sixth color of the color sequence is magenta which is given by the RGB value (255, 0, 255).*
color_flow_range	Defines the length of each color section of the six different color sections of the sequence. This parameter takes six different values. The first value defines the length of the first color section, the second value defines the length of the second color section and so on. *Example:* *color_flow_range=10,20,30,10,10,10* *In this case, the first color of the sequence is displayed over a length of 10, the second color over a length of 20, the third color over a length of 30 and the fourth, fifth and sixth color over a length of 10.*
color_flow_fading	Defines how smooth the transition between two successive color sections of the color sequence is. This parameter takes six different values. The first value describes the smoothness of the transition from the first to the second color section of the color sequence, the second value describes the smoothness of the transition from the second to the third color section of the color sequence and so on. *Example:* *color_flow_fading=10,30,10,10,10,10* *In this case, the transition from the first color section to the second color section takes place over a length of 10, the transition from the second color section to the third color section takes place over a length of 30 and so on.*
color_flow_speed	Defines the speed with which the sequence of color sections moves from the lower end of the blade to its tip. *Example:* *color_flow_speed=100* *In this case, the sequence of color sections moves from the lower end of the blade to its tip with a speed of 100.*
color_flow_responsive	Defines if the effect reacts to gestures in real-time by changing the smoothness of the transitions of the color sections. A value of 0 disables this feature. All other values enable it. A value of 1 lets the effect react to tilting the saber, a value of 2 lets it to react to twisting the saber and a value of 3 lets it react to both tilting the saber and twisting it. *Example:* *color_flow_responsive=1* *In this case, the smoothness of the transitions between the color sections can be changed by tilting the saber.*
color_flow_sauron	See the section "Sauron mode".
color_flow_sauron_size	See the section "Sauron mode".
color_flow_sauron_spark	See the section "Sauron mode".

Frozen Pulse

The background effect Frozen Pulse creates a pulsating spatial pattern on your blade that can be periodically shifted over time. The parameters of the background effect Frozen Pulse are shown below.

PARAMETER	DESCRIPTION
frzn_pls_color	All the parameters of the Frozen Pulse effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines the color of each channel which can be assigned to red, green or blue. The effect channels can be customized independently from each other. If one of the values of this parameter is set to 0, this assigns the corresponding effect channel to red. If it is set to 1, this assigns the corresponding effect channel to blue. If it is set to 2, this assigns the corresponding effect channel to green. *Example:* *frzn_pls_color=0,1,2* *In this case, the first effect channel is assigned to red, the second is assigned to blue and the third is assigned to green.*
frzn_pls_shape	Defines the spatial structure of the pattern on your blade. A value of 0 creates a sine pattern, a value of 1 creates an impulse pattern, a value of 2 creates an inverted impulse pattern, a value of 3 creates a rising sawtooth pattern, a value of 4 creates a falling sawtooth pattern, a value of 5 creates a triangle pattern and a value of 6 creates a rectangle pattern. *Example:* *frzn_pls_shape=0,1,5* *In this case, the first effect channel displays a sine pattern, the second displays an impulse pattern and the third displays a rectangle pattern.*
frzn_pls_int	Defines the maximum intensity of each effect channel. The intensity can take values between 0 and 255. *Example:* *frzn_pls_int=255,100,100* *In this case, the first effect channel has a maximum intensity of 255, the second has a maximum intensity of 100 and the third has a maximum value of 100.*
frzn_pls_contrast	Defines the contrast of each effect channel. The higher the contrast, the stronger the spatial pattern is visible. The contrast can take values between 0 and 100. *Example:* *frzn_pls_contrast=30,100,100* *In this case, the first effect channel has a contrast of 30, the second has a contrast of 100 and the third has a contrast of 100.*
frzn_pls_period	Defines the period with which the spatial pattern of each effect channel pulsates.

	Example: *frzn_pls_period=100,100,100* *In this case, the first effect channel has a period of 100, the second has a period of 100 and the third has a period of 100.*
frzn_pls_offset	Defines the offset of the spatial pattern of each effect channel. The offset can take values between 0 and 359. *Example:* *frzn_pls_offset=0,90,90* *In this case, the first effect channel has an offset of 0, the second has an offset of 90 and the third has an offset of 90.*
frzn_pls_min_dens	The spatial pattern of the Frozen Pulse effect consists of brighter and darker spots on the blade. The inverse distance of these spots is referenced as density in the following. The density of the pattern can spatially change across the blade. This parameter defines the minimum density of the spatial pattern of each effect channel. *Example:* *frzn_pls_min_dens=20,20,100* *In this case, the first effect channel has a minimum density of 20, the second has a minimum density of 20 and the third has a minimum density of 100.*
frzn_pls_max_dens	The spatial pattern of the Frozen Pulse effect consists of brighter and darker spots on the blade. The inverse distance of these spots is referenced as density in the following. The density of the pattern can spatially change across the blade. This parameter defines the maximum density of the spatial pattern of each effect channel. *Example:* *frzn_pls_max_dens=20,20,100* *In this case, the first effect channel has a maximum density of 20, the second has a maximum density of 20 and the third has a maximum density of 100.*
frzn_pls_dens_change_range	Defines the length of the section of the blade along which the density of each effect channel varies between its minimum value given by "frzn_pls_min_dens" and its maximum value given by "frzn_pls_max_dens". *Example:* *frzn_pls_dens_change_range=1000,2000,5000* *In this case, the first effect channel has a density change range of 1000, the second has a density change range of 2000 and the third has a density change range of 5000.*
frzn_pls_int_change_range	Currently not active.
frzn_pls_spd	Defines the speed with which the pulsating spatial pattern of each effect channel moves forward and backward along the blade. *Example:*

	frzn_pls_spd=50,100,100 *In this case, the first effect channel has a speed of 50, the second has a speed of 100 and the third has a speed of 100.*
frzn_pls_sauron	See the section "Sauron mode".
frzn_pls_sauron_size	See the section "Sauron mode".
frzn_pls_sauron_spark	See the section "Sauron mode".

Wave

The background effect Wave generates waves that emerge from the lower end of your blade and propagate to its tip. The parameters of the background effect Wave are shown below.

PARAMETER	DESCRIPTION
wave_led	Defines the background intensity of each effect channel. This is the intensity that would be displayed when no wave would propagate along the blade. The wave pattern digs into this background intensity. *Example:* *wave_led=255,0,0* *In this case, the background intensity of the first effect channel is 255, the background intensity of the second effect channel is 0 and the background intensity of the third effect channel is 0.*
wave_color	All the parameters of the Wave effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines the color of each channel which can be assigned to red, green or blue. The effect channels can be customized independently from each other. If one of the values of this parameter is set to 0, this assigns the corresponding effect channel to red. If it is set to 1, this assigns the corresponding effect channel to blue. If it is set to 2, this assigns the corresponding effect channel to green. *Example:* *wave_color=0,1,2* *In this case, the first effect channel is assigned to red, the second is assigned to blue and the third is assigned to green.*
wave_shape	Defines the spatial structure of the wave. A value of 0 creates a sine wave, a value of 1 creates an impulse wave, a value of 2 creates an inverted impulse wave, a value of 3 creates a rising sawtooth wave, a value of 4 creates a falling sawtooth wave, a value of 5 creates a triangle wave and a value of 6 creates a rectangle wave. *Example:* *wave_shape=0,1,5* *In this case, the first effect channel displays a sine wave, the second displays an impulse wave and the third displays a rectangle wave.*

wave_min_intensity	Defines the minimum depth of each effect channel by which the wave digs into the background intensity. *Example:* *wave_min_intensity=255,0,0* *In this case, the minimum depth of the first effect channel by which the wave digs into the background intensity is 255, the minimum depth of the second effect channel by which the wave digs into the background intensity is 0 and the minimum depth of the third effect channel by which the wave digs into the background intensity is 0.*
wave_max_intensity	Defines the maximum depth of each effect channel by which the wave digs into the background intensity. *Example:* *wave_max_intensity=255,0,0* *In this case, the maximum depth of the first effect channel by which the wave digs into the background intensity is 255, the maximum depth of the second effect channel by which the wave digs into the background intensity is 0 and the maximum depth of the third effect channel by which the wave digs into the background intensity is 0.*
wave_int_change_period	The wave pattern changes its intensity periodically between "wave_min_intensity" and "wave_max_intensity". This parameter defines the period of this change of each effect channel. *Example:* *wave_int_change_period=100,100,100* *In this case, the intensity change period of the first effect channel is 100, the intensity change period of the second effect channel is 100 and the intensity change period of the third effect channel is 100.*
wave_int_phase_shift	The wave pattern changes its intensity periodically between "wave_min_intensity" and "wave_max_intensity". This parameter defines at which intensity the wave of each effect channel starts when you ignite your saber. This parameter takes values between 0 and 359. *Example:* *wave_int_phase_shift=0,120,120* *In this case, the intensity phase shift of the first effect channel is 0, the intensity phase shift of the second effect channel is 120 and the intensity phase shift of the third effect channel is 120.*
wave_min_period	Defines the minimum period of each effect channel at which the waves are generated at the lower end of your blade. Higher values correspond to smaller speeds of generating the waves. *Example:* *wave_min_period=200,300,300*

	In this case, the minimum period of the first effect channel is 200, the minimum period of the second effect channel is 300 and the minimum period of the third effect channel is 300.
wave_max_period	Defines the maximum period of each effect channel at which the waves are generated at the lower end of your blade. Higher values correspond to smaller speeds of generating the waves. *Example:* *wave_max_period=200,300,300* *In this case, the maximum period of the first effect channel is 200, the maximum period of the second effect channel is 300 and the maximum period of the third effect channel is 300.*
wave_modulation_period	The period of each effect channel at which the waves are generated at the lower end of your blade alternates between "*wave_min_period"* and "*wave_max_period" over time.* This parameter defines the time needed for a transition between "*wave_min_period "* and "*wave_max_period" to take place*. *Example:* *wave_modulation_period=1000,2000,2000* *In this case, the modulation period of the first effect channel is 1000, the modulation period of the second effect channel is 2000 and the modulation period of the third effect channel is 2000.*
wave_modulation_phase_shift	The period of each effect channel at which the waves are generated at the lower end of your blade alternates between "*wave_min_period"* and "*wave_max_period" over time.* This parameter defines at which period the wave starts when igniting your saber. This parameter takes values between 0 and 359. *Example:* *wave_modulation_phase_shift=0,270,0* *In this case, the modulation phase shift of the first effect channel is 0, the modulation phase shift of the second effect channel is 270 and the modulation phase shift of the third effect channel is 0.*
wave_phase_shift	Defines the starting point of the wave when igniting your saber. This parameter takes values between 0 and 359. *Example:* *wave_phase_shift=0,120,120* *In this case, the phase shift of the first effect channel is 0, the phase shift of the second effect channel is 120 and the phase shift of the third effect channel is 120.*
wave_spd	Defines the speed at which the wave propagates from the lower end of your blade to its tip. *Example:* *wave_spd=100,200,100*

	In this case, the speed of the wave of the first effect channel is 100, the speed of the wave of the second effect channel is 200 and the speed of the wave of the third effect channel is 100.
wave_responsive	Defines if the effect reacts to gestures in real-time by changing the speed at which the waves are generated. A value of 0 disables this feature. All other values enable it. A value of 1 lets the effect react to tilting the saber, a value of 2 lets it to react to twisting the saber and a value of 3 lets it react to both tilting the saber and twisting it. *Example:* *wave_responsive=1* *In this case, the speed at which the waves are generated can be changed by tilting the saber.*
wave_sauron	See the section "Sauron mode".
wave_sauron_size	See the section "Sauron mode".
wave_sauron_spark	See the section "Sauron mode".

Flame Blade

The background effect Flame Blade lets your blade look like a flame that evolves over time as new sparks of the fire ignite at the lower end of your blade. The underlying algorithm that creates the flame effect simulates the behavior of a real fire which makes the flame effect as authentic as possible. The color gradient of the flame is determined by four different colors characterising the color of the hottest part of the flame, the higher medium temperature part of it, the lower medium part of it and the coolest part of it. The parameters of the background effect Flame Blade are shown below.

PARAMETER	DESCRIPTION
flame_blade_type	Defines the type of the flame. A value of 0 creates a realistic flame and a value of 1 creates a stylized flame.
flame_blade_speed	Defines the speed of the flame.
flame_blade_poweroff	Defines whether the flame retracts downwards or upwards. A value of 0 corresponds to the flame retracting downwards and a value of 1 corresponds to the flame retracting upwards.
flame_blade_color1	Defines the color of the coolest part of the flame. *Example:* *flame_blade_color1=0,0,0* *In this case, the coolest part of the flame is of black color which is given by the RGB value (0, 0, 0).*
flame_blade_color2	Defines the color of the lower medium temperature part of the flame. *Example:* *flame_blade_color2=255,0,0* *In this case, the lower medium temperature part of the flame is of red color which is given by the RGB value (255, 0, 0).*
flame_blade_color3	Defines the color of the higher medium temperature part of the flame. *Example:*

	flame_blade_color2=255,255,0 *In this case, the higher medium temperature part of the flame is of yellow color which is given by the RGB value (255, 255, 0).*
flame_blade_color4	Defines the color of the hottest part of the flame. *Example:* *flame_blade_color2=255,255,255* *In this case, the hottest part of the flame is of red color which is given by the RGB value (255, 255, 255).*
flame_blade_cooling	Defines how strongly the flame is cooled. Higher values lead to shorter flames. *Example:* *flame_blade_cooling=40* *In this case, the cooling of the flame is 40.*
flame_blade_fueling	Defines how strongly the flame is fueled. Higher values lead to more roaring flames. *Example:* *flame_blade_fueling=50* *In this case, the fueling of the flame is 50.*
flame_responsive	Defines if the effect reacts to gestures in real-time by changing the aggressivity of the flame. A value of 0 disables this feature. All other values enable it. A value of 1 lets the effect react to tilting the saber, a value of 2 lets it to react to twisting the saber and a value of 3 lets it react to both tilting the saber and twisting it. *Example:* *flame_blade_responsive=1* *In this case, the aggressivity of the flame can be changed by tilting the saber.*
flame_blade_sauron	See the section "Sauron mode".
flame_blade_sauron_size	See the section "Sauron mode".
flame_blade_sauron_spark	See the section "Sauron mode".

Gradient

The background effect Gradient creates a color gradient across the blade. This color gradient can change periodically over time. The parameters of the background effect Gradient are shown below.

PARAMETER	DESCRIPTION
gradient_number_of_colors	Defines the number of different colors between which the gradient is created. Only allowed value is 2 at the moment. *Example:* *gradient_number_of_colors=2* *In this case, the number of different colors between which the gradient is created is 2.*

gradient_speed	Defines the speed with which the color gradient alternates between the lower end of your blade and its tip. *Example:* *gradient_speed=100* *In this case, the speed with which the color gradient alternates between the lower end of your blade and its tip is 100.*"
gradient_color1	Defines the first of the two colors between which a color gradient is created. *Example:* *gradient_color1=255,0,0* *In this case, the first of the two colors between which a color gradient is created is red which is given by the RGB value (255, 0, 0).*
gradient_color2	Defines the second of the two colors between which a color gradient is created. *Example:* *gradient_color1=0,0,255* *In this case, the second of the two colors between which a color gradient is created is blue which is given by the RGB value (0, 0, 255).*

Now, let's have a look at the on top effects of the Golden Harvest v4 for Neopixel setup. There are 6 different categories: Focus Deflection, Thunderstorm, Kylo Blade, Unstable Blade, Magnetic Flare and Shockwave.

Focus Deflection

The on top effect Focus Deflection creates a focus point on your blade that moves forward and backward along the blade.

PARAMETER	DESCRIPTION
focus_defl_color	All the parameters of the Focus Deflection effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines the color of each channel which can be assigned to red, green or blue. The effect channels can be customized independently from each other. If one of the values of this parameter is set to 0, this assigns the corresponding effect channel to red. If it is set to 1, this assigns the corresponding effect channel to blue. If it is set to 2, this assigns the corresponding effect channel to green. *Example:* *focus_defl_color=0,1,2* *In this case, the first effect channel is assigned to red, the second is assigned to blue and the third is assigned to green.*

focus_defl_type	Defines the appearance of the focus point that moves forward and backward along the blade. This parameter takes values between 0 and 3 corresponding to four different types of appearance. *Example:* *focus_defl_type=0,0,1* *In this case, the first effect channel displays a focus point of type 0, the second displays a focus point of type 0 and the third displays a focus point of type 1.*
focus_defl_shape	Defines the type of movement of the focus point along the blade. A value of 0 creates a sine movement, a value of 1 creates an impulse movement, a value of 2 creates an inverted impulse movement, a value of 3 creates a rising sawtooth movement, a value of 4 creates a falling sawtooth movement, a value of 5 creates a triangle movement and a value of 6 creates a rectangle movement. *Example:* *focus_defl_shape=2,4,5* *In this case, the first effect channel displays an inverted impulse movement of the focus point, the second displays a falling sawtooth movement of the focus point and the third displays a triangle movement of the focus point.*
focus_defl_int	Defines the intensity of the focus point. This parameter takes values between 0 and 255. *Example:* *focus_defl_int=255,100,100* *In this case, the first effect channel displays a focus point with intensity of 255, the second displays a focus point with intensity of 100 and the third displays a focus point with intensity of 100.*
focus_defl_size	Defines the size of the focus point. *Example:* *focus_defl_size=1,1,5* *In this case, the first effect channel displays a focus point of size 1, the second displays a focus point of size 1 and the third displays a focus point of size 5.*
focus_defl_position_offset	Defines the offset of the starting point of the focus point when you ignite your blade. This parameter takes values between 0 and 359. *Example:* *focus_defl_position_offset=0,180,90*

	In this case, the first effect channel displays a focus point with an offset of the starting point of 0, the second displays a focus point with an offset of the starting point of 180 and the third displays a focus point with an offset of the starting point of 90.
focus_defl_smooth	Defines the smoothness of the focus point.114 *Example:* *focus_defl_smooth=10,10,20* *In this case, the first effect channel displays a focus point with smoothness of 10, the second displays a focus point with smoothness of 10 and the third displays a focus point with smoothness of 20.*
focus_defl_spark_int	Defines the intensity of the sparkling of the focus point. This parameter takes values between 0 and 255. *Example:* *focus_defl_spark_int=255,100,100* *In this case, the first effect channel displays a focus point with sparkling intensity of 255, the second displays a focus point with sparkling intensity of 100 and the third displays a focus point with sparkling intensity of 100.*
focus_defl_spark_delay	Defines the delay between successive intensity drops of the focus point. Higher values lead to a lower speed of the sparkling. *Example:* *focus_defl_spark_delay=1,1,1* *In this case, the first effect channel displays a focus point with sparkling delay of 1, the second displays a focus point with sparkling delay of 1 and the third displays a focus point with sparkling delay of 1.*
focus_defl_min_range	Defines the minimum range in which the focus points moves forward and backward along the blade. Takes values between 0 and 100. The values are given as percentage of the whole blade length. *Example:* *focus_defl_min_range=50,90,100* *In this case, the first effect channel displays a focus point with minimum range of 50, the second displays a focus point with minimum range of 90 and the third displays a focus point with minimum range of 100.*
focus_defl_max_range	Defines the maximum range in which the focus points moves forward and backward along the blade. Takes values between 0 and 100. The values are given as percentage of the whole blade length.

	Example: *focus_defl_max_range=50,90,100* *In this case, the first effect channel displays a focus point with maximum range of 50, the second displays a focus point with maximum range of 90 and the third displays a focus point with maximum range of 100.*
focus_defl_range_change_period	The movement of the focus point along the blade is restricted to the range of the effect. The range changes over time between "focus_defl_min_range" and "focus_defl_max_range". This parameter defines the period with which the range changes between "focus_defl_min_range" and "focus_defl_max_range". *Example:* *focus_defl_range_change_period=1000,5000,10000* *In this case, the first effect channel displays a focus point whose range of movement changes with a period of 1000, the second displays a focus point whose range of movement changes with a period of 5000 and the third displays a focus point whose range of movement changes with a period of 10000.*
focus_defl_min_speed	Defines the minimum speed at which the focus points moves forward and backward along the blade. *Example:* *focus_defl_min_speed=100,50,50* *In this case, the first effect channel displays a focus point that moves at a minimum speed of 100, the second displays a focus point that moves at a minimum speed of 50 and the third displays a focus point that moves at a minimum speed of 50.*
focus_defl_max_speed	Defines the maximum speed at which the focus points moves forward and backward along the blade. *Example:* *focus_defl_max_speed=100,50,50* *In this case, the first effect channel displays a focus point that moves at a maximum speed of 100, the second displays a focus point that moves at a maximum speed of 50 and the third displays a focus point that moves at a maximum speed of 50.*
focus_defl_speed_change_period	The speed at which the focus point moves forward and backward along the blade changes over time between "focus_defl_min_speed" and "focus_defl_max_speed". This parameter defines the period with which the speed of the focus points changes over time.

	Example: *focus_defl_speed_change_period=1000,2000,2000* *In this case, the first effect channel displays a focus point whose speed changes with a period of 1000, the second displays a focus point whose speed changes with a period of 2000 and the third displays a focus point whose speed changes with a period of 2000.*

Thunder Storm

The on top effect Thunder Storm generates random lightnings that can appear in groups and form lightning swarms.

PARAMETER	DESCRIPTION
thnd_strm_color1	All the parameters of the Thunder Storm effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines the color of the first effect channel. The effect channels can be customized independently from each other. *Example:* *thnd_strm_color1=255,255,255* *In this case, the first effect channel displays white lightnings as white is given by the RGB code (255, 255, 255).*
thnd_strm_color2	All the parameters of the Thunder Storm effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines the color of the second effect channel. The effect channels can be customized independently from each other. *Example:* *thnd_strm_color1=255,0,0* *In this case, the first effect channel displays red lightnings as red is given by the RGB code (255, 0, 0).*
thnd_strm_color3	All the parameters of the Thunder Storm effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines the color of the third effect channel. The effect channels can be customized independently from each other. *Example:* *thnd_strm_color1=0,0,255* *In this case, the first effect channel displays blue lightnings as is given by the RGB code (0, 0, 255).*

thnd_strm_min_intensity	Defines the minimum intensity of the lightnings of each effect channel. This parameter takes values between 0 and 100. *Example:* *thnd_strm_min_intensity=100,50,50* *In this case, the first effect channel displays lightnings with a minimum intensity of 100, the second effect channel displays lightnings with a minimum intensity of 50 and the third effect channel displays lightnings with a minimum intensity of 50.*
thnd_strm_max_intensity	Defines the maximum intensity of the lightnings of each effect channel. This parameter takes values between 0 and 100. *Example:* *thnd_strm_max_intensity=100,50,50* *In this case, the first effect channel displays lightnings with a maximum intensity of 100, the second effect channel displays lightnings with a maximum intensity of 50 and the third effect channel displays lightnings with a maximum intensity of 50.*
thnd_strm_swarm_min_dur	Defines the minimum duration of the lightnings within a swarm of each effect channel. *Example:* *thnd_strm_swarm_min_dur=1,1,1* *In this case, the first effect channel displays lightnings with a minimum duration of 1 within a swarm, the second effect channel displays lightnings with a minimum duration of 1 within a swarm and the third effect channel displays lightnings with a minimum duration of 1 within a swarm.*
thnd_strm_swarm_max_dur	Defines the maximum duration of the lightnings within a swarm of each effect channel. *Example:* *thnd_strm_swarm_min_dur=1,1,1* *In this case, the first effect channel displays lightnings with a maximum duration of 1 within a swarm, the second effect channel displays lightnings with a maximum duration of 1 within a swarm and the third effect channel displays lightnings with a maximum duration of 1 within a swarm.*
thnd_strm_swarm_min_del	Defines the minimum delay between successive lightnings within a swarm of each effect channel. *Example:* *thnd_strm_swarm_min_del=1,10,50* *In this case, the first effect channel displays lightnings with a minimum delay of 1 between successive lightnings within a*

	swarm, the second effect channel displays lightnings with a minimum delay of 10 between successive lightnings within a swarm and the third effect channel displays lightnings with a minimum delay of 50 between successive lightnings within a swarm.
thnd_strm_swarm_max_del	Defines the maximum delay between successive lightnings within a swarm of each effect channel. *Example:* *thnd_strm_swarm_max_del=1,10,50* *In this case, the first effect channel displays lightnings with a maximum delay of 1 between successive lightnings within a swarm, the second effect channel displays lightnings with a maximum delay of 10 between successive lightnings within a swarm and the third effect channel displays lightnings with a maximum delay of 50 between successive lightnings within a swarm.*
thnd_strm_swarm_min_number	Defines the minimum number of lightnings within a swarm of each effect channel. *Example:* *thnd_strm_swarm_min_number=1,10,10* *In this case, the first effect channel displays lightnings with a minimum number of 1 lightning within a swarm, the second effect channel displays lightnings with a minimum number of 10 lightnings within a swarm and the third effect channel displays lightnings with a minimum number of 10 lightnings within a swarm.*
thnd_strm_swarm_max_number	Defines the maximum number of lightnings within a swarm of each effect channel. *Example:* *thnd_strm_swarm_max_number=1,10,10* *In this case, the first effect channel displays lightnings with a maximum number of 1 lightning within a swarm, the second effect channel displays lightnings with a maximum number of 10 lightnings within a swarm and the third effect channel displays lightnings with a maximum number of 10 lightnings within a swarm.*
thnd_strm_light_min_del	Defines the minimum delay between successive swarms of lightnings of each effect channel. *Example:* *thnd_strm_light_min_del=1,10,50* *In this case, the first effect channel displays swarms of lightnings with a minimum delay of 1 between successive swarms, the second effect channel displays swarms of*

	lightnings with a minimum delay of 10 between successive swarms and the third effect channel displays swarms of lightnings with a minimum delay of 50 between successive swarms.
thnd_strm_light_max_del	Defines the maximum delay between successive swarms of lightnings of each effect channel. *Example:* *thnd_strm_light_min_del=1,10,50* *In this case, the first effect channel displays swarms of lightnings with a maximum delay of 1 between successive swarms, the second effect channel displays swarms of lightnings with a maximum delay of 10 between successive swarms and the third effect channel displays swarms of lightnings with a maximum delay of 50 between successive swarms.*

Kylo Blade

The on top effect Kylo Blade gives your blade a grained structure that alters the underlying background effect.

PARAMETER	DESCRIPTION
kylo_blade_strength	Defines how strong the structure of the background effect is altered. This parameter takes values between 0 and 100. *Example:* *kylo_blade_strength=100* *In this case, the structure of the background effect is maximally altered.*
kylo_blade_type	There are three different variations of the effect. This parameter defines which type is active. This parameter takes values between 0 and 2. *Example:* *kylo_blade_type=0* *In this case, the type 0 is active.*
kylo_blade_range	Defines how coarse or fine the grained structure is. *Example:* *kylo_blade_range=10* *In this case, a range of 10 is chosen.*
kylo_blade_density	Defines the density of the grained structure. *Example:* *kylo_blade_density=250*

	In this case, a density of 250 is chosen.
kylo_blade_smoothness	Defines the smoothness of the grained structure. *Example:* *kylo_blade_smoothness=100* *In this case, a smoothness of 100 is chosen.*
kylo_blade_dissipation	Defines the stability of the grained structure. Higher values lead to increased instability. *Example:* *kylo_blade_dissipation=100* *In this case, a dissipation of 100 is chosen.*

Unstable Blade

The on top effect Unstable Blade gives your blade an unstable structure that alters the underlying background effect by a spatially distributed, random intensity drops.

PARAMETER	DESCRIPTION
unst_blade_color	All the parameters of the Unstable Blade effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines which color component of the background effect is affected by the corresponding each channel. It can be assigned to red, green or blue. The effect channels can be customized independently from each other. If one of the values of this parameter is set to 0, the corresponding effect channel affects the red color component of the background effect. If it is set to 1, the corresponding effect channel affects the blue color component of the background effect. If it is set to 2, the corresponding effect channel affects the green color component of the background effect. *Example:* *unst_blade_color=0,1,2* *In this case, the first effect channel affects the red color component of the background effect, the second affects the blue color component of the background effect and the third affects the green color component of the background effect.*
unst_blade_int	Defines how strong the corresponding color component of the background effect is altered. This parameter takes values between 0 and 255. *Example:* *unst_blade_int=255,0,0* *In this case, the color component of the background effect that is affected by the first effect channel is maximally altered, the color component of the background effect that is affected by the second effect channel is not*

	altered and the color component of the background effect that is affected by the third effect channel is not altered.
unst_blade_contrast	Defines the contrast of the spatial structure applied to the color component of the background effect that is affected by the corresponding effect channel. The higher the contrast, the stronger the spatial pattern is visible. The contrast can take values between 0 and 100. *Example:* *unst_blade_contrast=30,100,100* *In this case, the spatial distribution of the color component of the background effect that is affected by the first effect channel has a contrast of 30, the spatial distribution of the color component of the background effect that is affected by the second effect channel has a contrast of 100 and the spatial distribution of the color component of the background effect that is affected by the third effect channel has a contrast of 100.*
unst_blade_type	There are three different variations of the effect. This parameter defines which type of each effect channel is active. This parameter takes values between 0 and 2. *Example:* *unst_blade_type=0,0,2* *In this case, the first effect channel is of type 0, the second effect channel is of type 0 and the third effect channel is of type 2.*
unst_blade_min_range	Defines the minimum range of the spatially distributed, random intensity drops of each effect channel. *Example:* *unst_blade_min_range=1,2,5* *In this case, the intensity drops of the first effect channel have a minimum range of 1, the intensity drops of the second effect channel have a minimum range of 2 and the intensity drops of the third effect channel have a minimum range of 5.*
unst_blade_max_range	Defines the maximum range of the spatially distributed, random intensity drops of each effect channel. *Example:* *unst_blade_max_range=1,2,5* *In this case, the intensity drops of the first effect channel have a maximum range of 1, the intensity drops of the second effect channel have a maximum range of 2 and the intensity drops of the third effect channel have a maximum range of 5.*
unst_blade_density	Defines the density of the spatially distributed, random intensity deops of each effect channel. *Example:* *unst_blade_density=100,500,1000*

	In this case, the density of the intensity drops of the first effect channel is 100, the density of the intensity drops of the second effect channel is 500 and the density of the intensity drops of the third effect channel is 1000.
unst_blade_smooth	Defines the smoothness of the spatially distributed, random intensity drops of each effect channel. *Example:* *unst_blade_smooth=10, 20, 50* *In this case, the intensity drops of the first effect channel have a smoothness of 10, the intensity drops of the second effect channel have a smoothness of 20 and the intensity drops of the third effect channel have a smoothness of 50.*
unst_blade_cooling	Defines the cooling, i.e. how fast the blade recovers from the spatially distributed, random intensity drops of each effect channel. *Example:* *unst_blade_cooling=100, 100, 100* *In this case, the cooling of the first effect channel is 100, the cooling of the second effect channel is 100 and the cooling of the third effect channel is 100.*

Magnetic Flare

The on top effect Magnetic Flare gives your blade a regular, pulsating spatial pattern that alters the underlying background effect.

PARAMETER	DESCRIPTION
mag_flare_strength	Defines how strong the underlying background effect is altered. This parameter takes values between 0 and 100. *Example:* *mag_flare_strength=50* *In this case, the strength with which the background effect is altered is 50.*
mag_flare_period	Defines the period with which the regular spatial pattern pulsates over time. *Example:* *mag_flare_period=100* *In this case, the period with which the regular spatial pattern pulsates over time is 100.*
mag_flare_offset	Defines the spatial offset of the regular, pulsating spatial pattern. This parameter takes values between 0 and 359. *Example:* *mag_flare_offset=0* *In this case, the spatial offset of the regular, pulsating spatial pattern is 0.*

mag_flare_dens	Defines the density of the regular, pulsating spatial pattern. *Example:* *mag_flare_dens=100* *In this case, the regular, pulsating spatial pattern has a density of 100.*
mag_flare_spd	Defines the speed at which the regular, pulsating spatial pattern moves from the lower end of your blade to its tip. *Example:* *mag_flare_spd=10* *In this case, the speed at which the regular, pulsating spatial pattern moves from the lower end of your blade to its tip is 10.*

Shockwave

The on top effect Shockwave gives your blade a fluid structure by creating spatial waves on the underlying background effect.

PARAMETER	DESCRIPTION
shockwave_color	All the parameters of the Shockwave effect take three values. The first values of all parameters belong to the same effect channel as do the second and third values. This parameter defines which color component of the background effect is altered by the spatial waves of the corresponding each channel. It can be assigned to red, green or blue. The effect channels can be customized independently from each other. If one of the values of this parameter is set to 0, the corresponding effect channel affects the red color component of the background effect. If it is set to 1, the corresponding effect channel affects the blue color component of the background effect. If it is set to 2, the corresponding effect channel affects the green color component of the background effect. *Example:* *shockwave_color=0,1,2* *In this case, the first effect channel affects the red color component of the background effect, the second affects the blue color component of the background effect and the third affects the green color component of the background effect.*
shockwave_int	The spatial waves are generated by creating sudden intensity drops similar to rain drops creating waves. This parameter defines the intensity of these intensity drops of each effect channel. *Example:*

	shockwave_int=10,10,10 *In this case, the intensity of the intensity drops that generate the the spatial waves of the first effect channel is 10, the intensity of the intensity drops that generate the spatial waves of the second effect channel is 10 and the intensity of the intensity drops that generate the spatial waves of the third effect channel is 10.*
shockwave_contrast	Defines the contrast of the spatial waves of each effect channel that are created on the corresponding color component of the background effect. *Example:* *shockwave_contrast=75,75,75* *In this case, the contrast of the spatial waves of the first effect channel is 75, the contrast of the spatial waves of the second effect channel is 75 and the contrast of the spatial waves of the third effect channel is 75.*
shockwave_sync	The three different effect channels can be synchronized with respect to which color component of the background effect they affect. This parameter defines if an effect channel should affect the same color component of the background effect as the first effect channel. *Example:* *shockwave_sync=1,1,1* *In this case, the three different effect channels affect the same color component of the background effect as the first effect channel.*
shockwave_type	Defines the shape of the intensity drops that generate the spatial waves of each effect channel. This parameter can take values between 0 and 2. *Example:* *shockwave_type=0,1,2* *In this case, the intensity drops that generate the spatial waves of the first effect channel are of type 0, the intensity that generate the spatial waves of the second effect channel are of type 1 and the intensity that generate the spatial waves of the third effect channel are of type 2.*
shockwave_min_size	Defines the minimum size of the intensity drops that generate the spatial waves of each effect channel. *Example:* *shockwave_min_size=10,10,10* *In this case, the minimum size of the intensity drops that generate the spatial waves of the first effect channel is 10, the minimum size of the intensity drops that generate the*

	spatial waves of the second effect channel is 10 and the minimum size of the intensity drops that generate the spatial waves of the third effect channel is 10.
shockwave_max_size	Defines the maximum size of the intensity drops that generate the spatial waves of each effect channel. *Example:* *shockwave_max_size=10,10,10* *In this case, the maximum size of the intensity drops that generate the spatial waves of the first effect channel is 10, the maximum size of the intensity drops that generate the spatial waves of the second effect channel is 10 and the maximum size of the intensity drops that generate the spatial waves of the third effect channel is 10.*
shockwave_min_pos	Defines the minimum position of the intensity drops that generate the spatial waves of each effect channel. This parameter takes values between 0 and 100. The values are given as percentage of the whole blade length. *Example:* *shockwave_min_pos=0,0,0* *In this case, the minimum position of the intensity drops that generate the spatial waves of the first effect channel is 0, the minimum position of the intensity drops that generate the spatial waves of the second effect channel is 0 and the minimum position of the intensity drops that generate the spatial waves of the third effect channel is 0.*
shockwave_max_pos	Defines the maximum position of the intensity drops that generate the spatial waves of each effect channel. This parameter takes values between 0 and 100. The values are given as percentage of the whole blade length. *Example:* *shockwave_max_pos=100,100,100* *In this case, the maximum position of the intensity drops that generate the spatial waves of the first effect channel is 100, the maximum position of the intensity drops that generate the spatial waves of the second effect channel is 100 and the maximum position of the intensity drops that generate the spatial waves of the third effect channel is 100.*
shockwave_maintenance	Defines the how long the spatial waves of each effect channel are maintained. This parameter takes values between 0 and 100. *Example:* *shockwave_maintenance=99,99,99*

	In this case, the spatial waves of the first effect channel have a maintenance of 99, the spatial waves of the second effect channel have a maintenance of 99 and the spatial waves of the third effect channel have a maintenance of 99.
shockwave_speed	Defines the speed of the spatial waves of each effect channel. *Example:* *shockwave_speed=50,50,50* *In this case, the spatial waves of the first effect channel have a speed of 50, the spatial waves of the second effect channel have a speed of 50 and the spatial waves of the third effect channel have a speed of 50.*
shockwave_min_del	Defines the minimum delay between successive intensity drops that generate the spatial waves of each effect channel. *Example:* *shockwave_min_del=100,200,500* *In this case, the minimum delay between successive intensity drops that generate the spatial waves of the first effect channel is 100, the minimum delay between successive intensity drops that generate the spatial waves of the second effect channel is 200 and the minimum delay between successive intensity drops that generate the spatial waves of the third effect channel is 500.*
shockwave_max_del	Defines the maximum delay between successive intensity drops that generate the spatial waves of each effect channel. *Example:* *shockwave_max_del=100,200,500* *In this case, the maximum delay between successive intensity drops that generate the spatial waves of the first effect channel is 100, the maximum delay between successive intensity drops that generate the spatial waves of the second effect channel is 200 and the maximum delay between successive intensity drops that generate the spatial waves of the third effect channel is 500.*