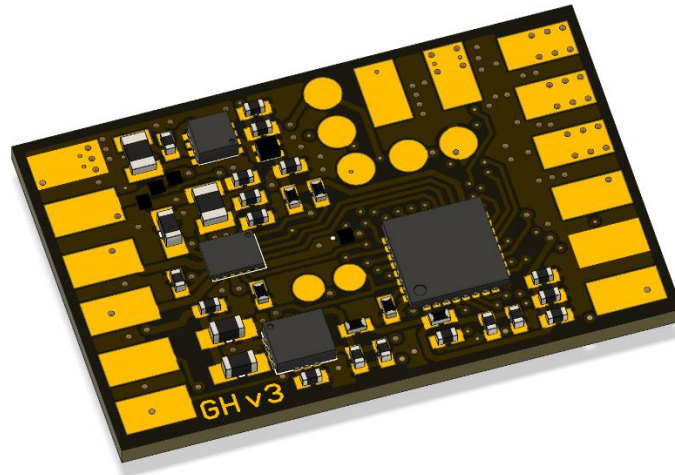# Golden Harvest v3

Manual

SaberTec
Owner: Frederic Folz
Brentanoweg 3
66802 Überherrn
GERMANY

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

07/15/2021

Thank you

Thank you for sharing your passion with us! It's only your support that enables us to further develop our technology and continue optimizing it. Without you, we couldn't realize our visions! Back in 2016, we founded SaberTec with the goal to develop a soundboard that can provide the most impressive and realistic saber experience imaginable – four years of hard work later the Golden Harvest v3 was born. Our company's fundamental philosophy is to closely include you, the community, into everything we do – may it be the development of our saber heart, the decision of its name or the further improvement of it. In this sense, we want to invite you to participate in this great project. If you have any ideas or wishes, we would be honored if you would shared them with us! Just write a mail to info@sabertec.net. We are excited to hear from you!

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 v3 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 v3

Technical Specifications

➢ tiny dimensions of 28.5 mm x 18.5 mm x 3.7 mm / 1.12 ″ x 0.73 ″ x 0.15 ″
➢ powered by 3.7 V (single 18650 Li-ion cell for example)
➢ provides 9 LED channels that can be customized independently from each other:
   o 4 LED channels can deliver up to 9 A each:
      ▪ can drive SingleCrees, DualCrees, TriCrees and QuadCrees
      ▪ can drive Neopixel
   o 5 LED channels can deliver up to 20 mA each:
      ▪ can drive accent, crystal chamber or switch LEDs for example
   o capable of delivering up to 36 A in sum (in practice, both battery and heat dissipation are the limiting factors here)
   o 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 per drag-and-drop on the microSD card
➢ full functionality accessible even if used with only one momentary switch
➢ can also be used with two momentary switches
➢ customizable deep sleep (hibernation mode)

Audio Engine

➢ maximum audio amplifier power output: 4 W
   o supported speaker power: 1 W - 6 W
   o supported speaker impedance: 4 Ohm, 8 Ohm, 16 Ohm and 32 Ohm
   o high dynamic range: 115 dB
   o real-time dynamic range compression
   o high signal-to-noise ratio: 100 dB
   o 16 bit audio sample resolution
   o integrated speaker protection algorithms:
      ▪ exceeding voice coil excursion protection
      ▪ overtemperature protection
      ▪ overcurrent protection
      ▪ undervoltage protection
      ▪ 15 kV ESD protection of speaker output
➢ full-blown sound effect engine:
   o unlimited smooth swing pair sounds
   o unlimited smooth swing accent sounds
   o unlimited swing sounds
   o unlimited clash sounds
   o unlimited stab sounds
   o unlimited spin sounds
   o unlimited begin lockup sounds

- o unlimited lockup sounds
- o unlimited end lockup sounds
- o unlimited begin melt sounds
- o unlimited melt sounds
- o unlimited end melt sounds
- o unlimited begin drag sounds
- o unlimited drag sounds
- o unlimited end drag sounds
- o unlimited blaster sounds
- o unlimited force sounds
- o unlimited boot sounds
- o unlimited pre-on sounds
- o unlimited power on sounds
- o unlimited power off sounds
- o unlimited post-off sounds
- o unlimited hum sounds
- o menu sounds
- o unlimited sound fonts that contain all these sound effects
- o 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:
  - o unlimited tracks
  - o playlist feature
  - o can be controlled on the fly
  - o 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 16 sound fonts that come with the board:
  - o "Balance" by LordBlako Saber Fonts
  - o "Psy-Borg" by CrystalSoniX
  - o "Daddy Issues Complete" by Fourzze FX
  - o "Guardian" by LDN Sabers
  - o "Cryo Carbon" by Repulse Custom Sounds
  - o "Peace" by Kyberphonic Fonts
  - o "Violence" by Kyberphonic Fonts
  - o "Son of Darkness" by Dark Path Media
  - o "Eternal Prince" by Repulse Custom Sounds
  - o "The Phantom" by Project Fonts
  - o "The Shadow" by Project Fonts
  - o "Shadows" by Echo Studio
  - o "Dark Harvest" by Blueforce Company
  - o "The Classic Villain" by Fourzze FX
  - o "Energy Pike" by The Proplicator
  - o "Knighthood" by Fourzze FX

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

- o configurable effect duration
- o independent basic effects
- o independent clash effects
- o independent stab effects
- o independent swing effects
- o independent spin effects
- o independent lockup effects
- o independent melt effects
- o independent blaster effects
- o independent force effects
- o as there are more than 600 parameters, you can literally invent your own light effects
- o 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!
- o ingenious online configuration tool for designing, saving and sharing your light effects (will be released very soon)
- o 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:
  - o overwhelming light effect engine
  - o full customizability of the effects
  - o 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
  - o all Neopixel effects consist of background effects and on top effects that can be arbitrarily combined with each other
  - o 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, …)
  - o 7 different on top effect categories:
    - ▪ None
    - ▪ Focus Deflection
    - ▪ Thunder Storm
    - ▪ Unstable Blade
    - ▪ Kylo Blade
    - ▪ Magnetic Flare

- Shockwave
  - o configurable fade out
  - o configurable effect transition:
    - configurable transition time
    - configurable transition type:
      - Basic
      - Window
      - Unstable Blade
      - Flame Blade
  - o configurable effect duration
  - o localized clash
  - o localized lockup
  - o localized drag
  - o localized melt
  - o localized blaster block
  - o independent basic effects
  - o independent clash effects
  - o independent stab effects
  - o independent swing effects
  - o independent spin effects
  - o independent lockup effects
  - o independent blaster effects
  - o independent force effects
- ➢ 5 independent channels to drive standard accent LEDs
- ➢ capability of defining unique light effects even for standard accent LEDs
- ➢ sequencer for standard accent LEDs:
  - o up to 16 different steps
  - o duration of each step customizable
  - o 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:
  - o up to 16 different steps
  - o duration of each step customizable
  - o 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 USB module, the Golden Harvest can be recharged and the microSD can be accessed via microUSB
- ➢ charging with up to 1.5 A

- ➢ you can use any wall adapter independent from the maximum current it can supply thanks to a smart charging algorithm
- ➢ easy to install as the module needs only 4 wires to be connected to the Golden Harvest
- ➢ smaller than 11 mm x 19 mm / 0.433" x 0.748"

Evolution of the Board

Since its first release, several minor hardware changes of the Golden Harvest v3 have been implemented. The different hardware iterations of the board are labelled as phase 1, 2, 3 and 4. Each of these boards is capable of using the very same firmware. In the following, we describe how you can determine which hardware iteration you own and what the differences are.

How to Recognize the Different Hardware Iterations

The phase 1 boards are green, all other boards are black. You can recognize the phase 2 boards as they have a resistor across two of the 5 circular pads on the top side of the board. The phase 3 boards have a brown rectangular part in the upper left corner of the top side of the board. The phase 4 boards are currently in production. Their look is not yet revealed. They will be the final hardware iteration.

Phase 1	Phase 2
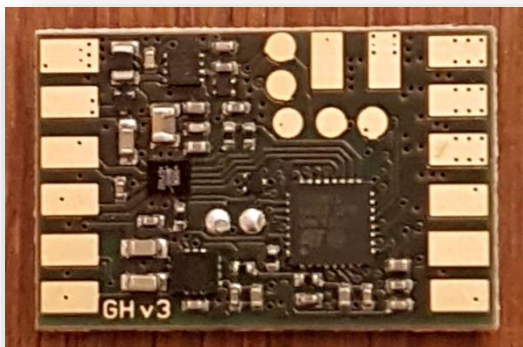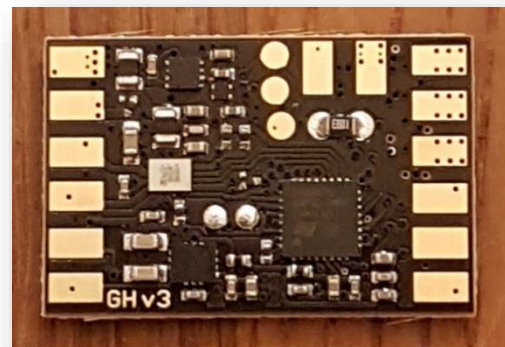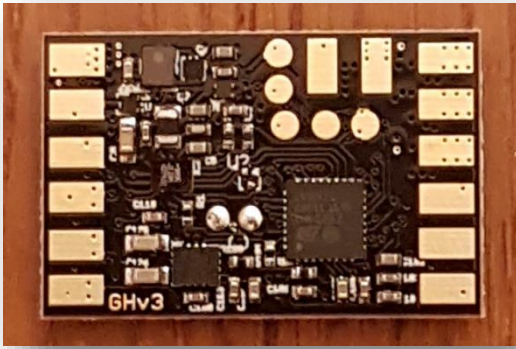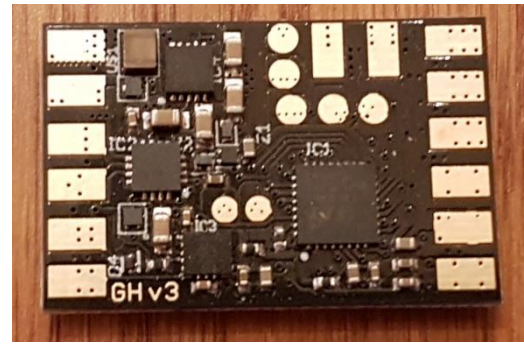
Changes introduced by phase 2:
- Decreased wait time when disconnecting and connecting the battery.
- Batt. + pad on the top side instead of on the back side.

Changes introduced by phase 3:
- Wait time when disconnecting and connecting the battery completely removed.
- Part in the upper left corner of the top side of the board replaced by a protected one to make it more robust against mistakes during install.
- Reverse polarity protection added.
- Accent LED channel 5 added.

Changes introduced by phase 4:
- Short protection for the audio amplifier added.
- Audio clarity and volume could be further enhanced.
- Protection against overheating during soldering could be further extended.
- ESD protection of the buttons was enhanced.
- ESD protection of the Pixel data line was enhanced.
- ESD protection of the power line was enhanced.
- Deep sleep shelf time was increased (current draw was reduced).
- A "PHASE4.GHV" file is automatically created on the SD card as an identification.

The protection mechanisms added to phase 4 are very powerful. However, this does not mean that you shouldn't be careful during soldering and handling.

## Wiring of the Board

After getting in touch with the technical specifications, we'll now come to how to install the Golden Harvest v3 correctly. The general wiring of our board is shown in the wiring diagrams on the following pages which are examples for different LED setups (there are much more possible than shown though). As mentioned above, our board features four different high-power LED channels. These are accessible via the pads labelled as "LED1", "LED2", "LED3" and "LED4" which need to be connected to the negative of the LEDs. Please make sure to use appropriate resistors for your in-hilt LEDs (we don't take any responsibility for possible damage due to using inappropriate resistors). To calculate the needed resistivity, have a look at the data sheets of the LEDs you want to use. The four high-power LED channels don't necessarily have to drive your blade LED as you can even drive accent LEDs with them.

If you decide to use only two LED channels for the blade for example, you can use the remaining LED channels to drive a switch LED or crystal chamber LEDs for example. As each LED channel provides up to 9 A, you could even connect up to 450 accent LEDs in parallel to each LED channel (assuming they need 20 mA each). As you'll see in one of the following sections, you can even define custom light effects for the accent LEDs. They can be synchronized to the blade effects, but you can also choose them to be completely independent of the blade effects. This enables you to design independent crystal chamber effects for example.

Another important pad is the "Pwr. Sw." pad to which you connect the power switch as its name already suggests. The power switch has to be a momentary switch. If you decide to go for a two-switch configuration, you can add a second switch to the "Aux. Sw." pad which has to be a momentary switch. The second pin of each switch has to be connected to the "GND" pad.

For each switch, it is recommended to twist its two wires around each other. That way, the impact of electromagnetic induction can be reduced. This is important as electromagnetic induction could potentially lead to random behavior. Furthermore, it is important to use the "GND" pad between the "Aux. Sw." and the "Pwr. Sw." pad to solder the ground wire of each switch.

Top side of the board   **Back side of the board**
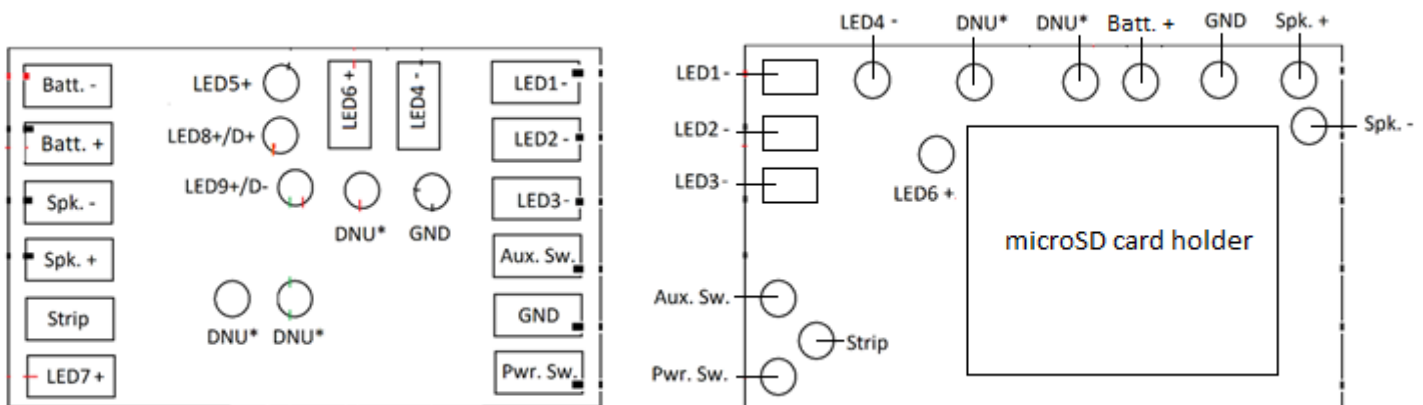
* DNU means "do not use this pad" as the corresponding pads
do not have any purpose yet. However, this might change with
future firmware updates.

The position of the Batt. + pad indicated above holds only for the black boards. For the green boards, please use the Batt. + pad on the back side. The LED channel 5 is only available on phase 3 and phase 4 boards.

The table below lists the functions of each pad on the top side of the board.

Pad	Function	Remark
Batt. -	Connect to negative of battery	Internally connected to all GND pads
Batt. +	Connect to positive of battery	
Spk. -	Connect to negative of speaker	
Spk. +	Connect to positive of speaker	

Strip	Connect to data line of Neopixel	Resistor is already included on the board
D +	Connect to D + of the USB module	
D -	Connect to D - of the USB module	
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)	Only available on phase 3 and 4 boards
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)	

Power Rating and Polarity of LED channels

All 9 LED channels of the Golden Harvest v3 are independently configurable. It features 4 high-power LED channels and 5 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. The LED channel 5 is only available on phase 3 and phase 4 boards.

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

Soldering Advice

The pads of the Golden Harvest v3 were designed to be as big as possible in order to make the soldering process as easy as possible. Compared to the other pads, the pad "Batt. -" has an even higher electrical and thermal conductivity. The soldering temperature should not exceed 400° C and you should not heat the pads for longer than 1 - 2 s. Also, make sure to let the board cool down a bit before proceeding with soldering the next pad.

When soldering the board, make sure to do so while the board is not powered. That means solder the battery only in the very last step or use a kill key to keep it disconnected while soldering.

Also, it is recommended to use the rectangular pads on the top side of the board instead of the circular ones on the back side because those on the top side are stronger. When soldering to the circular pads, you need to make sure not to exert any tensile stress to the pads, neither by the wires nor by the soldering iron and not to apply excessive heat to them. This is important in order to not lift-off the pads due to their small size. Independent of that, it is also important for the rectangular pads to avoid tensile stress exerted by the wires.

Handling Advice

In general, electronics are sensitive to electrostatic discharge. Please always make sure to be grounded when touching the board.

In the following, the wiring of a recharge port and of our Seedling USB module are shown. Both wiring diagrams depict the respecitve connections independent from the actual LED setup you are using. Please also have a look at the wiring diagrams for the LED setups.

Recharge Port
(in addition to LED wiring)

The wiring of a recharge port is shown below. It is the same for any LED setup. The recharge port has to be connected such that it connects "Batt. -" and the negative of the battery if the kill key is pulled out.
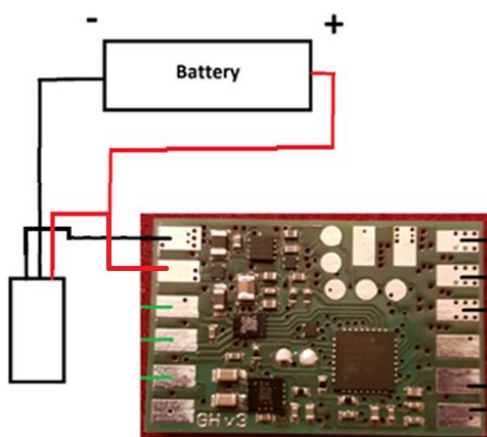
Seedling module
(in addition to LED wiring)

The wiring of the Seedling module is shown below. It is the same for any LED setup. Please make sure that the orange and green wires are of same length.
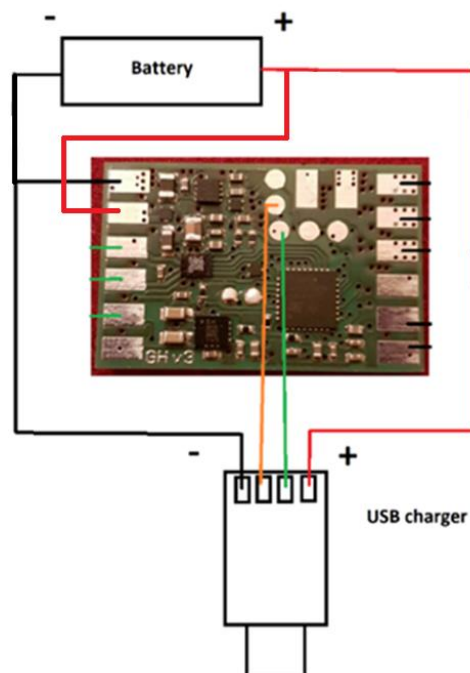
In the following, the wiring diagrams for different LED setups are shown. For in-hilt setups, the resistors can be added either to the positive or the negative of the LEDs.

Single color blade with separate clash color

The wiring of an in-hilt LED as single color blade with separate clash color is shown below.
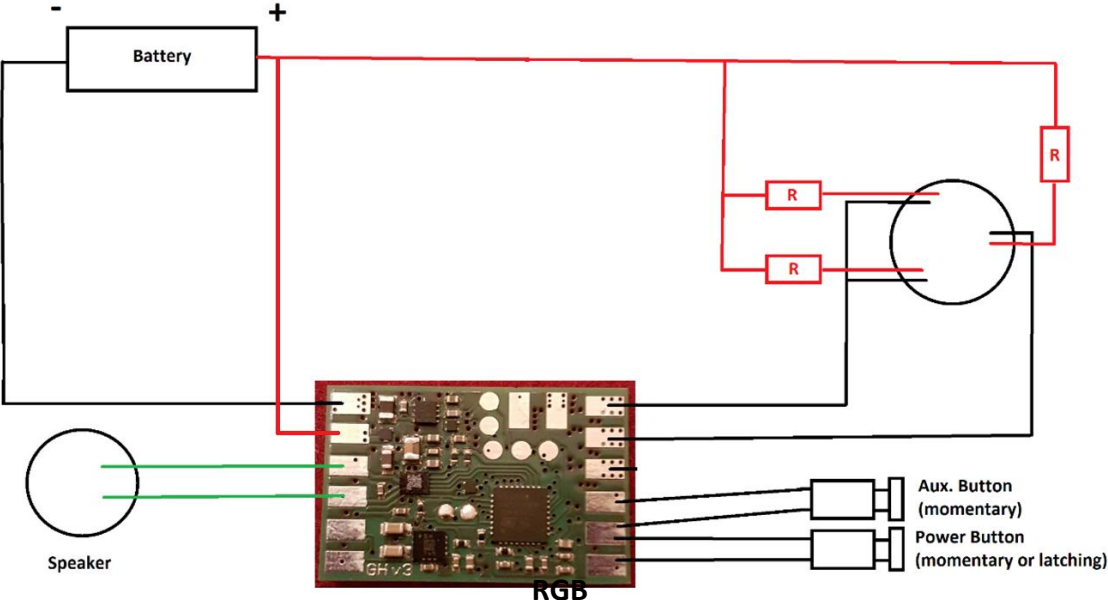
The wiring of an in-hilt RGB LED is shown below.
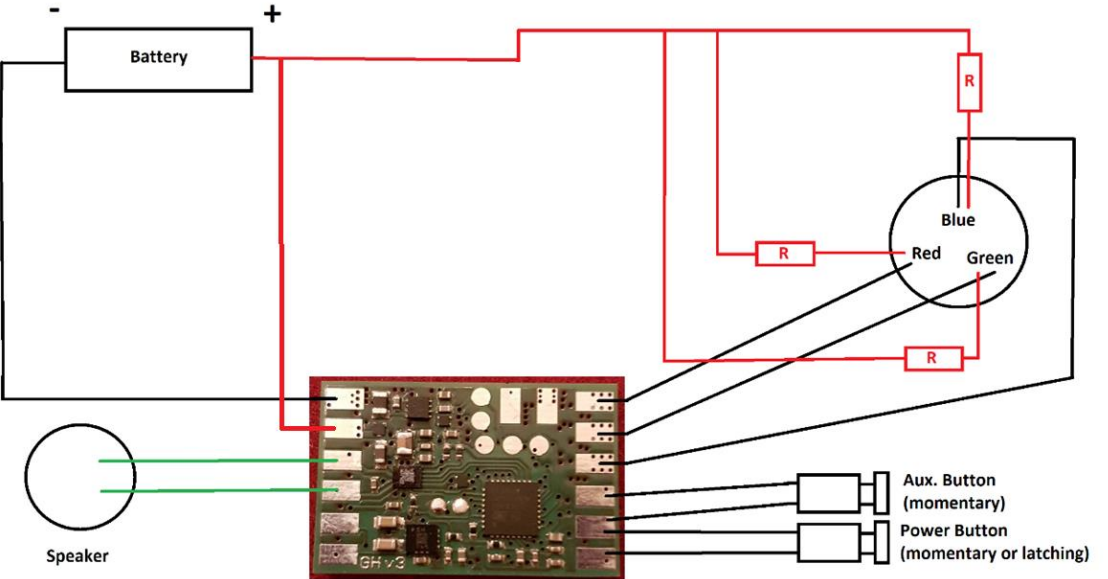
RGBW

The wiring of an in-hilt RGBW LED is shown below.
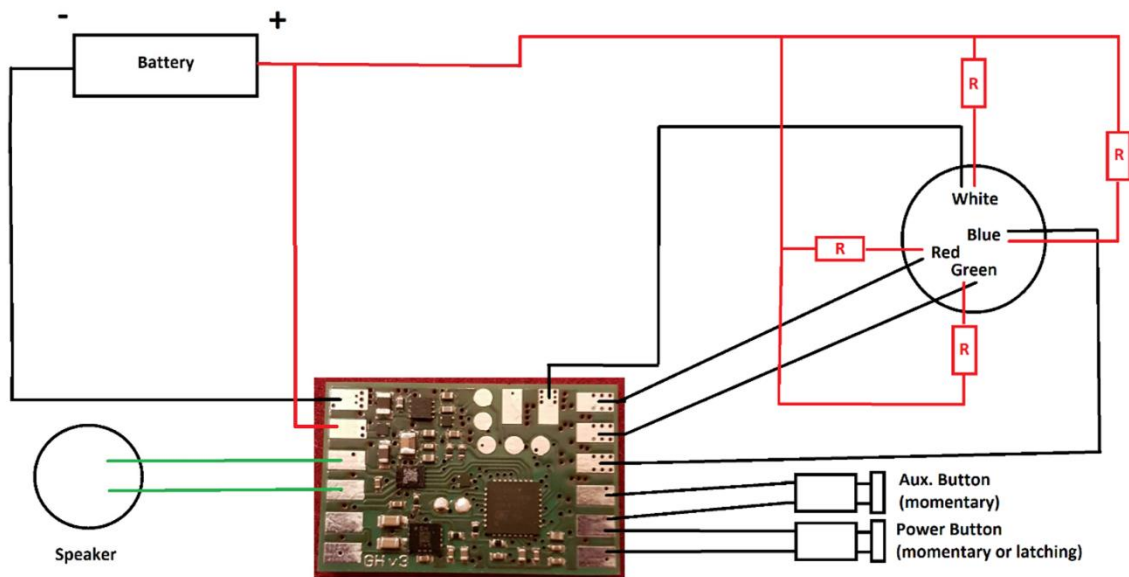
Pixel strips without Pixel accents

The wiring of a Pixel strip is shown below (commonly known as Neopixel™ which is a trademark of Adafruit™ with whom we are not affiliated). There is already a 470 Ohm data line resistor on the board so that you don't need to add one. If your blade is pre-resistored, this will not affect the functionality. To setup your board to use a Pixel strip, set the parameter "number_of_leds" in the "general.txt" to the number of Pixel LEDs in your blade. When counting the number of Pixel LEDs, please count only the Pixel LEDs on one of the two strips in your blade. The common range of numbers of Pixel LEDs is between 110 and 135. The board supports ws2812b LED strips. If you want to use standard accent LEDs with a Pixel blade, make sure to use the pads LED5 - LED9.
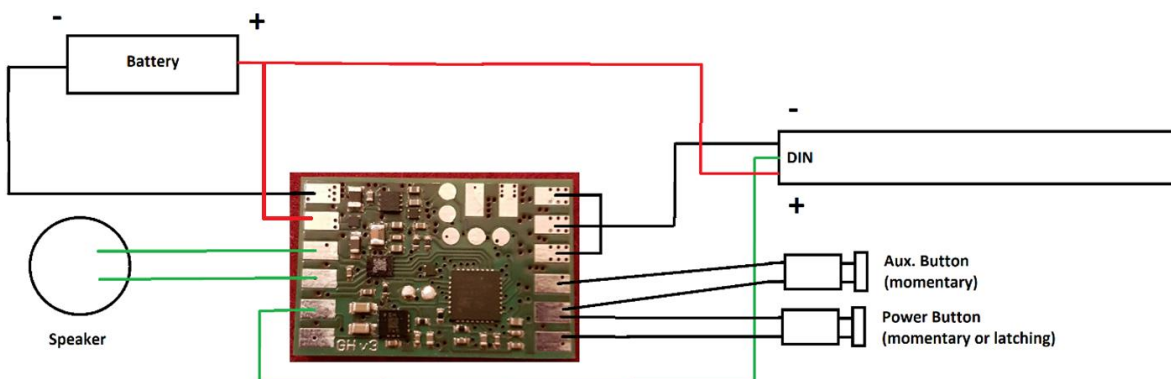
Pixel strips with Pixel accents

The wiring of a Pixel strip with Pixel accents is shown below. There is already a 470 Ohm data line resistor on the board so that you don't need to add one. If your blade is pre-resistored, this will not affect the functionality. To setup your board to use a Pixel strip, set the parameter "number_of_leds" in the "general.txt" to the number of Pixel LEDs in your blade. When counting the number of Pixel LEDs, please count only the Pixel LEDs on one of the two strips in your blade. The common range of numbers of Pixel LEDs is between 110 and 135. Beyond that, you need to set the parameter "number_of_neopixel_accents" in the "general.txt" to the number of Pixel accents you are using. The board supports ws2812b LED strips. If you want to use standard accent LEDs with a Pixel blade, make sure to use the pads LED5 - LED9.
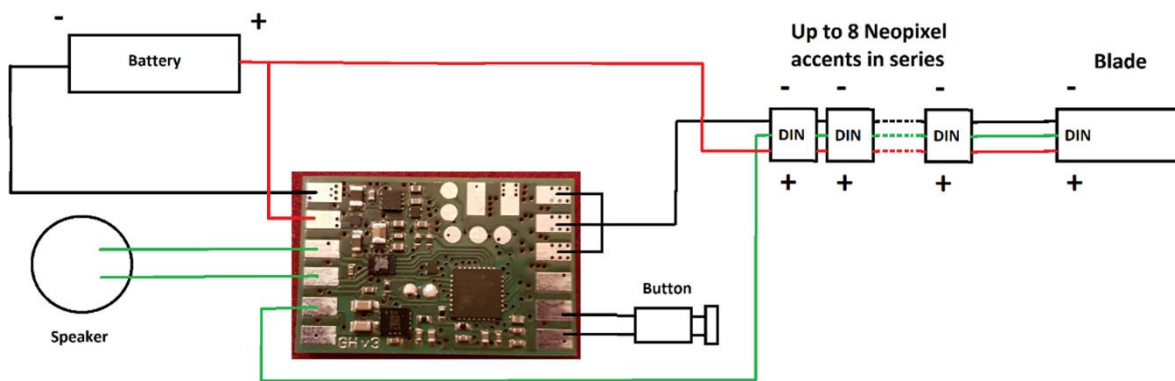
Quad Cree with fixed color

The wiring of an In-Hilt Quad Cree LED with fixed color is shown below.

Resistors are not depicted. They should be used to limit the LED current.

Battery Recommendation

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

In-Hilt Setup

If you're using a TriCree for example, you are fine with most **protected** single-cell Li-ion batteries. Our recommendation is to use a KeepPower 18650 protected battery as this is the one we are using in our own demo sabers. If you're using a crossguard setup with three separate TriCrees in total, we recommend to use a KeepPower 18650 protected battery that is able to provide enough current. We recommend to use a battery that is rated for 10 A.

Neopixel Setup

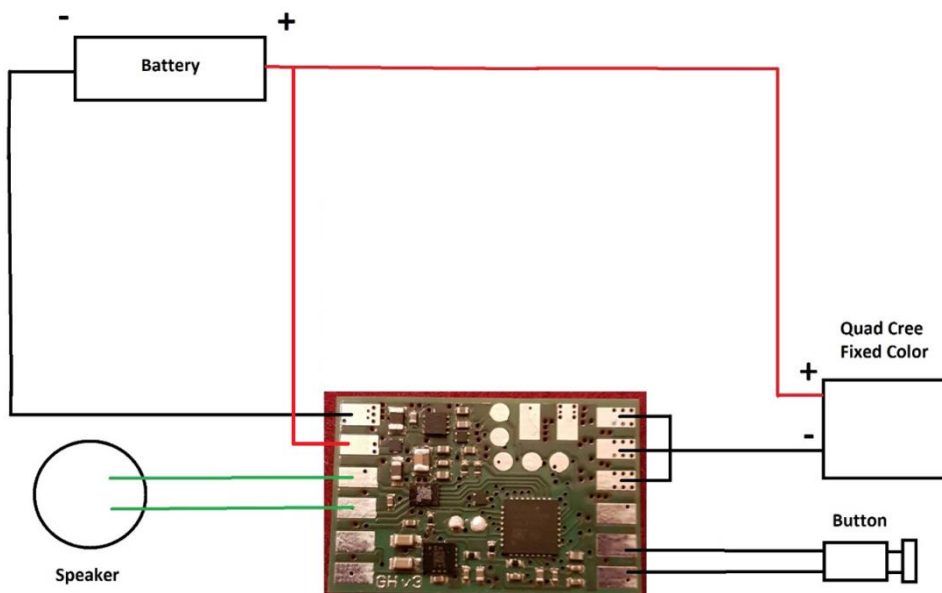If you're using Neopixel, we recommend to use a KeepPower 18650 protected battery that is rated for 15 A.

So, you see that the current rating is of great importance. Why is that? Well, if you're using a battery that is only rated for 3 A for example, but you're running Neopixel with it, the voltage of the battery will drop strongly. This low voltage will not only dim your Neopixel, but can also damage the battery in the long term.

Setting Up Your Board

To setup the Golden Harvest for your saber, please read the following instructions on how to choose your hardware setup and on how to choose your light effects.

Choosing Your Hardware Setup

You can easily customize your hardware setup by opening the "general.txt" file on your microSD card. To configure your board for your hardware setup, it is sufficient to adjust only three parameters which are described in the following.

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
usb	Defines if you want to use USB for charging and file transfer. The following values are possible: - usb=0: disable USB - usb=1: enable USB
number_of_leds	Defines the number of Pixels in your blade. Only needed to adjust if you are using Pixel strips. Please count only the number of LEDs of one of the Pixel strips. Usually, the number is between 110 and 135. The number of LEDs is internally clamped to 144.

There are even more customizable parameters in the "general.txt" of course, but to configure your Golden Harvest v3 for your hardware setup, only these three parameters need to be adjusted. You can also adjust the parameter "number_of_leds" on the fly using the Saber Editor.

Choosing Your Light Effects

You can easily customize your light effects by changing the value of the parameter "effect_font_slot" in the "general.txt" file on your microSD card. In the image below, you see which values correspond to the different LED setups. For Neopixel, there are three default packages available that differ with respect to the number of presets that they include.

You can also change the effect font slot on the fly using the Saber Editor.

Folder Structure on microSD Card

The Golden Harvest v3 comes with a microSD card containing the default package. In the picture below, you can see the folder structure of the microSD card. Let's go through it from top to bottom. The first folder is the "EffectFonts" folder which contains the different effect fonts, i. e. the different light effect presets of your saber. Similarly, there is a "SoundFonts" folder containing all the sound fonts of your saber. By default, your Golden Harvest v3 comes with 16 different sound fonts. The maximum possible number of sound fonts is not limited though. The next folder is the "UserInterfaceSounds" folder which contains all user interface sounds like the battery indicator sounds or the volume control sounds for example. In addition to these folders, there is the "general.txt" file which will be described in detail in one of the next sections. Beyond that, there may be an "UPDATE_D.DAT" file. As its name suggests, it is related to the firmware update feature of the board. This file is automatically created after a firmware update has been installed. To install a new firmware update, you just need to remove the "UPDATE_D.DAT" file and replace it by the new firmware update file "update.dat". The phase 3 and phase 4 boards don't create the "UPDATE_D.DAT" file any more so that you don't need to take care of it on these boards.

Hibernation Mode

The Golden Harvest board features also a Hibernation Mode which allows a shelf-time of several months. The board enters this mode once it wasn't used for 5 minutes while it's powered. This default value can be customized by adjusting the parameter "deep_sleep" which will be described in more detail in the "General Settings" section below. You can wake the board up from the Hibernation Mode by pressing the power button.

Motion Detection Engine

A key feature of our board is its highly sophisticated motion detection engine. The Golden Harvest v3 supports both smooth swing and legacy motion detection. The latter is automatically used whenever a sound font does not contain any smooth swing pairs. These different modes are described below.

Smooth Swing

One of the outstanding capabilities of the Golden Harvest v3 is its smooth swing engine that creates swing sound effects in dependence on the motion speed in real time giving rise to the most realistic experience possible. The Golden Harvest v3 measures the speed with which your saber rotates and generates the corresponding swing sound effects accordingly. Furthermore, accent swings are added to the experience. Their volume is modulated by the rotation speed of your saber.

Legacy Motion Detection

The legacy motion detection engine is capable of detecting four different movements: swings, clashs, stabs and spins. Although all of them are detected by a complex algorithm, we tried to reduce the set of parameters for calibrating the motion engine as far as possbile. We were able to limit it to only one sensitivity parameter for each type of motion, apart from spins which intrinsically have two parameters to specify. As it will be described in more detail in the "General Settings" section, each type of motion has a threshold parameter that belongs to it. This threshold parameter is a measure for the sensitivity as it specifies how strong a motion has to be in order to detect the respective motion type. The higher the threshold is, the smaller is the sensitivity. In addition to the threshold parameter, spins have a trigger duration parameter. It describes how long the motion has to be stronger than specified by the threshold parameter without any interruption. Virtually, this corresponds to how long you have to rotate your saber without any interruption in order to trigger a spin.

General Settings

The general settings of your Golden Harvest v3 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 the parameters in the "general.txt" using a text editor or using our online configurator which will be available very soon. 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. The picture below shows how the "general.txt" file looks when opened using a text editor.

```
#SaberTec Customizer - Hardware Setup File


#_____
#
#General Settings
#_____


volume=100

button_off_time=350
button_on_time=10

button_short_time=5
button_normal_time=125
button_long_time=200
button_mode=4

blade_double_tap=1

stab_ignition=0
swing_ignition=0
swing_ignition_threshold=5000
twist_ignition=0

twist_retraction=0
twist_protection=1500

preon=2
postoff=2

blaster=0
blaster_disable=2000
blaster_threshold=5000

lockup=0

force_push=1
force_push_threshold=5000

accent_menu=1

color_menu=1
color_menu_twist_speed=100

effect_menu=1
effect_menu_twist_speed=100

sound_menu=1
sound_font_jump=10
sound_menu_twist_speed=100

volume_control=2
volume_control_tilt_speed=100
volume_control_twist_speed=200
```

```
color_profile_edit_mode=2
color_profile_tilt_speed=100
color_profile_twist_speed=100
pixel_number_edit_mode=1
pixel_number_twist_speed=200
ignition_speed_edit_mode=1
ignition_speed_twist_speed=200
retraction_speed_edit_mode=1
retraction_speed_twist_speed=200
battle_configuration_edit_mode=1
battle_configuration_twist_speed=200
gesture_controls_edit_mode=1
gesture_controls_twist_speed=200
effect_font_slot_edit_mode=1
effect_font_slot_twist_speed=200

ignition_led=1023,0,0
retraction_led=1023,0,0

orientation=0

swing_threshold=5000
clash_threshold=3000
stab_threshold=3000
spin_threshold=15000
spin_trigger_duration=300

accent_swing_threshold=12000
accent_spin_threshold=20000

twist_threshold=20000

swing_cooldown=200
clash_cooldown=150
stab_cooldown=150
spin_cooldown=250
blasterblock_protection=10000
forcepush_protection=10000

effect_font_slot=9

effect_fading=750

number_of_leds=113
number_of_neopixel_accents=0

usb=1

power_save=50

accent_standby=1

accent_battery_indicator=0

accent_blade_matching=0

deep_sleep=300


#_____
#
#End of File
#_____
```

PARAMETER	DESCRIPTION
volume	Defines the maximum possible volume of your saber. Ranges from 0 (mute) to 400 (maximum loudness). Recommended value of 100. Please check if your speaker can handle such a high volume when increasing the volume to higher values than 100. We cannot be held responsible for any speaker damage that results from overpowering them.
button_off_time	Defines how long you have to press the power button to turn your saber off.
button_on_time	Defines how long you have to press the power button to turn your saber on.
button_short_time	In the menu navigation, a short button press triggers certain features (see section "Menu navigation"). This parameter defines how long you have to press a button to be recognized as a short press.
button_normal_time	In the menu navigation, a normal button press triggers certain features (see section "Menu navigation"). This parameter defines how long you have to press a button to be recognized as a normal press.
button_long_time	In the menu navigation, a long button press triggers certain features (see section "Menu navigation"). This parameter defines how long you have to press a button to be recognized as a long press.
button_mode	Defines which switch configuration you're using. A value of 4 corresponds to a single momentary switch and a value of 2 corresponds to two momentary switches.
blade_double_tap	See section "Effect Menu".
stab_ignition	Defines if a stab ignites the saber. A value of 0 disables this feature and a value of 1 enables it. This parameter is only used if a sound font does not contain a "settings.txt" that also allows to customize the stab ignition. Otherwise the stab ignition will be customized in the "settings.txt".
swing_ignition	Defines if a swing ignites the saber. A value of 0 disables this feature and a value of 1 enables it. This parameter is only used if a sound font does not contain a "settings.txt" that also allows to customize the swing ignition. Otherwise the swing ignition will be customized in the "settings.txt".
swing_threshold	Defines if how strong you need to swing the saber to trigger a swing ignition. The higher the value the less sensitive the swing ignition is.
twist_ignition	Defines if a twist ignites the saber. A value of 0 disables this feature and a value of 1 enables it. This parameter is only used if a sound font does not contain a "settings.txt" that also allows to customize the twist ignition. Otherwise the twist ignition will be customized in the "settings.txt".
twist_retraction	Defines if a twist retracts the saber. A value of 0 disables this feature and a value of 1 enables it. This parameter is only used if a sound font does not contain a "settings.txt"

	that also allows to customize the twist retraction. Otherwise the twist retraction will be customized in the "settings.txt".
twist_protection	The twist retraction is blocked if the movement of the saber is stronger than this parameter. This avoids that the saber accidentally turns off during movement.
preon	Defines how preon sounds and light effects are triggered. See section "Preon Sounds and Light Effects".
postoff	Defines how postoff sounds and light effects are triggered. See section "Postoff Sounds and Light Effects".
blaster	Defines how blaster blocks are triggered. See section "Blaster Block".
blaster_disable	See section "Blaster Block".
blaster_threshold	See section "Blaster Block".
lockup	Defines how lockups are triggered. See section "Lockup".
force_push	Defines how force pushs are triggered. See section "Force Push".
force_push_threshold	See section "Force Push".
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	While in the color menu, you can change the color profile by slowly twisting the saber. This parameter describes the sensitivity oft he twist motion.
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	While in the effect menu, you can change the blade profile by slowly twisting the saber. This parameter describes the sensitivity oft he twist motion.
sound_menu	Defines which sound menu mode you want to use. Depending on the mode, you can change the sound font by tapping the power button or twisting the saber while in the sound menu for example. See section "Sound Menu".
sound_font_jump	While in the sound menu, you can jump forwards or backwards by a certain number of sound fonts. This parameters describes how many sound fonts are skipped.
sound_menu_twist_speed	While in the sound menu, you can change the sound font by twisting the saber depending on the sound menu mode. This parameter defines how sensitive twisting the saber is.
volume_control	Controls the volume control feature. A value of 0 disables this feature. All other values enable it. A value of 1 allows to change the volume by tilting the saber, a value of 2 allows to change the volume by twisting the saber and a value of 3 allows to change the volume both by tilting the saber and twisting it.
volume_control_tilt_speed	Defines the sensitivity of the volume control to tilting the saber.

volume_control_twist_speed	Defines the sensitivity of the volume control to twisting the saber.
color_profile_edit_mode	See section "Saber Editor".
color_profile_tilt_speed	See section "Saber Editor".
color_profile_twist_speed	See section "Saber Editor".
pixel_number_edit_mode	See section "Saber Editor".
pixel_number_twist_speed	See section "Saber Editor".
ignition_speed_edit_mode	See section "Saber Editor".
ignition_speed_twist_speed	See section "Saber Editor".
retraction_speed_edit_mode	See section "Saber Editor".
retraction_speed_twist_speed	See section "Saber Editor".
gesture_controls_edit_mode	See section "Saber Editor".
gesture_controls_twist_speed	See section "Saber Editor".
effect_font_slot_edit_mode	See section "Saber Editor".
effect_font_slot_twist_speed	See section "Saber Editor".
ignition_led	See section "Saber Editor".
retraction_led	See section "Saber Editor".
orientation	Defines the orientation of the board in your saber which is required for the orientation detection based features to work properly. If the high power LED output pads point towards the tip of your saber, please set this parameter to 0. If they point towards the pommel of your saber, please set this parameter to 1.
swing_threshold	Defines the minimum rotation speed of your saber required to trigger a swing when using the legacy motion detection. The lower this value, the more sensitive your swing detection is.
clash_threshold	Defines the minimum strength with which you hit an obstacle required to trigger a clash. The lower this value, the more sensitive your clash detection is.
stab_threshold	Defines the minimum strength with which you perform a stab required to trigger a stab. The lower this value, the more sensitive your stab detection is.
spin_threshold	Defines the minimum rotation speed of your saber required to trigger a spin when using the legacy motion detection. The lower this value, the more sensitive your spin detection is.
spin_trigger_duration	Defines how long you have to rotate your saber with the minimum rotation speed defined by "spin_threshold" without any interruption to trigger a spin. The lower this value, the more sensitive your spin detection is.
accent_swing_threshold	Defines the minimum rotation speed of your saber required to trigger an accent swing when using smooth 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 when using smooth swing. The lower this value, the more sensitive your accent spin detection is.
twist_threshold	Defines the minimum twist speed around the pommel to tip axis of your saber required to recognize a twist

	motion. The twist motion is used by some button layouts described in the section "Menu navigation" instead of a double tap. The lower this value, the more sensitive your twist detection is.
swing_cooldown	Defines how long no other swing can be triggered after a swing has been detected. Can protect swings from a too fast interruption by other swings.
clash_cooldown	Defines how long no other clash can be triggered after a clash has been detected. Can protect clashs from a too fast interruption by other clashs.
stab_cooldown	Defines how long no other stab can be triggered after a stab has been detected. Can protect stabs from a too fast interruption by other stabs.
spin_cooldown	Defines how long no other spin can be triggered after a spin has been detected. Can protect spins from a too fast interruption by other spins.
blasterblock_protection	Defines how long no motion can be triggered after a blaster block has been triggered. This holds only for legacy motion detection. If you want to protect the blaster block completely, set this parameter to a high value.
forcepush_protection	Defines how long no motion can be triggered after a force push has been triggered. This holds only for legacy motion detection. If you want to protect the force push completely, set this parameter to a high value.
effect_font_slot	Defines which effect font slot you are using.
effect_fading	Defines the transition time between two different effects when you change the background effect or the on top effect.
number_of_leds	If you are using Pixel strips, set this parameter to the number of Pixel LEDs in your blade. Please count only the number of LEDs of one of the Pixel strips. Usually, the number is between 110 and 135. The number of LEDs is internally clamped to 144.
number_of_neopixel_accents	Set this parameter to the number of Pixel accents you are using.
usb	Defines whether or not you are using the USB file transfer feature of the Seedling. Set this parameter to 0 if you are not using the USB feature and set it to 1 if you are using the USB feature.
power_save	Defines the percentage by which the brightness of the blade is reduced when power saving is activated.
accent_standby	Defines if you want the accent LEDs to stay lit when the blade is retracted. A value of 0 turns them off when the blade is retracted and a value of 1 lets them stay lit when the blade is retracted. However, the accent LEDs will turn off when the board enters the Hibernation Mode in any case.
accent_battery_indicator	Defines if you want to play an indication sound when charging starts and display a charging indication on the

	accent LEDs. A value of 0 disables this feature and a value of 1 enables it.
accent_blade_matching	Defines if you want to match the color of a RGB switch with the color of the blade when using a Pixel blade. A value of 0 disables this feature and a value of 1 enables it. The red LED of the switch needs to be connected to LED7, the green LED of the switch to LED8 and the blue LED of the switch to LED9.
deep_sleep	Defines the time after which the board enters the Hibernation Mode if it's not used used while it's powered. Values are given in seconds.

Blade Styles

During the development of our Golden Harvest board we focused in particular on a high customizability of the light and sound effects. Here, especially two aspects were very important to us. On the one hand, our goal was to provide you an overwhelming, realistic experience that let's you dive into another world. On the other hand, the light and sound effects should be versatilely configurable such that you could even create your very own world by giving your lightsaber an unique style. May it be simple, complex or even exotic effects – due to the great variety of our settings, you can realize almost every effect you can imagine and literally even invent own ones. On the Golden Harvest v3, you can use more than 600 parameters for in-hilt setups and several thousands for Neopixel setups which are customizable by using our online configurator. Just unleash your creativity! But even if you're not willing to change many parameters: our Golden Harvest board can also be run without changing any parameter at all as it comes with a default set of parameter values.

There can be arbitrarily many blade styles on your SD card. They are represented by folders labelled "blade[number]" in the "EffectFonts" folder. Hereby, the enumeration needs to be continuous and start with 1. You can also add descriptions to each "blade[number]" folder. It's only important that the folder name begins with "blade[number]". An example of a possible naming of the blade folders is shown below.

Each blade style defines particular light effects and colors of your saber. As there are some differences between blade styles for in-hilt and blade styles for Neopixel, we will discuss their structure separately in the following. The structure of the blade styles for in-hilt is shown in the picture below.

Each blade style comprises an "effect1.txt" file and arbitrarily many "color[number].txt" files. The "effect1.txt" file contains all the parameters that define the light effects. The "color[number].txt" files are color profiles that define the different colors that this blade style can display. And the best thing is: you're able to change both the blade style and the color profile on the fly while the saber is turned on!

Now, let's come to the structure of the blade styles for Neopixel. It is shown in the picture below.

Each blade style comprises a "background.txt" file, arbitrarily many "ontop[number].txt" files and arbitrarily many "color[number].txt" files. The "background.txt" file contains all the parameters that define the background effect which describes the main effect category that is used for this blade style. The "ontop[number].txt" files define on top effects of this blade styles which are added on top of the background effect. Having multiple on top effects in a blade style allows to have different variations of

the background effect. The "color[number].txt" files define the different colors that this blade style can display. Similar to in-hilt, you're able to change the blade style, the on top effect and the color profile on the fly while the saber is turned on!

Whenever you change the blade style, the blade.wav sound that is contained in it is played which is an identification sound of the blade style. There are default blade.wav sounds. However, you can also replace them by your own custom ones. Just make sure that the sound files fulfill the requirements mentioned in the section "Sound Fonts".

The mentioned files are accessible on your microSD card. You can open them with all common text editors, independent of your operating system. So, one possibility of changing the light effects of your saber or creating your own light effects is to adjust the respective parameter values in the files manually. This is rather a choice of purists. The other possibility is to use our revolutionizing online configurator to design your custom effects! We're very proud of this configurator as we spent months of hard work and a lot of passion to provide you a flabergasting experience and make your eyes sparkle. In the online configurator, you can adjust all the parameteres graphically via numerical input fields, multiple-choice fields or sliders. One of the most impressive features of it is a real-time preview of the light effects you're just creating. It shows an animation of how your saber would look like using these parameter values. A more detailed description of our configurator you can find on https://sabertec.net/downloads. The support of Neopixel is currently on the way.

Our configurator will be accessible on our website soon.

Now, we come to the description of the parameters you can customize manually in the blade styles or with our configurator. There are slight differences between in-hilt blade styles and Neopixel blade styles. First, we will have a look at the in-hilt blade styles.

In-Hilt

As mentioned before, the blade styles for in-hilt comprise an "effect1.txt" file and arbitrarily many "color[number].txt" files. Both of these file types consist of 10 different blocks of similar structure. Each of these blocks corresponds to one of the 10 different states the saber can be in: ignition, basic, swing, clash, stab, spin, lockup, melt, blaster block or force push. By adjusting the parameters in a certain block, you can customize the light effects or the color of the corresponding saber state. But… wait a minute. This is genious! So, you can even assign special light effects to swings for example! A color change during the swing would be really exotic though, but you could even design a rather smooth light effect change during the swing: for example a weak flicker. So, each time you swing your saber it would flicker, simulating a blade that gets unstable because of sourcing more energy from the diatium power cell and dissipating it. You see – it's possible to literally invent your own custom light effects!

Almost all parameters have the format "parameter=value1,value2,value3,value4", whereby "value1" is the value of the parameter for LED channel 1, "value2" is the value of the parameter for LED channel 2 and so on. Thus, these parameters can be set for each LED channel independently! This creates an enormous range of different light effects that enables your saber to literally become unique. Let's first have a look at the "color[number].txt" files. In each state, there is a parameter called "led" that describes the color. This parameter defines the relative power with which the respective LED channels

are driven and thus the color of the blade. Its values range from 0 to 1023 for each channel. A value of 0 means that the respective LED channel provides no power, whereas a value of 1023 means that the respective LED channel provides the maximum power. If you set "led=0,300,600,1023" for example, this means that LED channel 1 gets no power, LED channel 2 gets a power value of 300, LED channel 3 gets a power value of 600 and LED channel 4 gets a power value of 1023. The parameter "led" of the basic state is shown below.

```
#SaberTec Customizer - Color Profile


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

In the other states, this parameter has a prefix indicating the state. For example, in the clash state the parameter is called "cl_led".

Now, let's turn to the "effect1.txt" file. This file contains all the parameters that define the light effects of the blade style. The parameters of 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 define whether an LED channel drives a blade LED or an accent LED and whether one of the LED channels should display a delayed ignition (the latter is not active yet). One of the fascinating features of our board is that the accent channels are not different from the blade channels with respect to the customizability of the light effects. You can define the full range of light effects also for accent LEDs, completely independet of the blade LEDs if you want! This is a really amazing feature, especially if you want to enlighten a crystal chamber for example. With our board, you can design unique crystal chamber effects, may it be a simple flicker or pulse effect or a more advanced, exotic effect like a color transition.In the following, the parameters of the basic state are described.
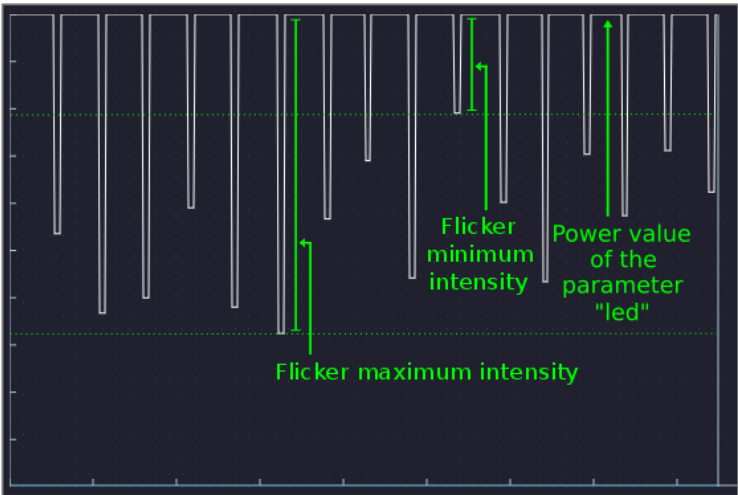
The structure consists of two parts: a flicker part and a pulse part. Let's have a look at the flicker part first. It's called the flicker part as the parameters that it contains define a flicker effect. This effect creates random (but also deterministic if you want) and sudden drops in the power level defined by the parameter "led". It can be characterised by three basic properties: the flicker intensity, the flicker delay and the flicker duration. Hereby, the flicker intensity describes how deep these drops are whereas the flicker delay describes the temporal distance between two successive drops. The flicker duration describes how long these power drops maintain.

The picture below illustrates these three characteristic properties of the flicker effect graphically. It shows the power an LED gets over time from one LED channel (the term power may not be technically correct here, but it intuitively describes it best). As you can see, the flicker intensity, duration and delay can vary over time which will become clear in the next section.
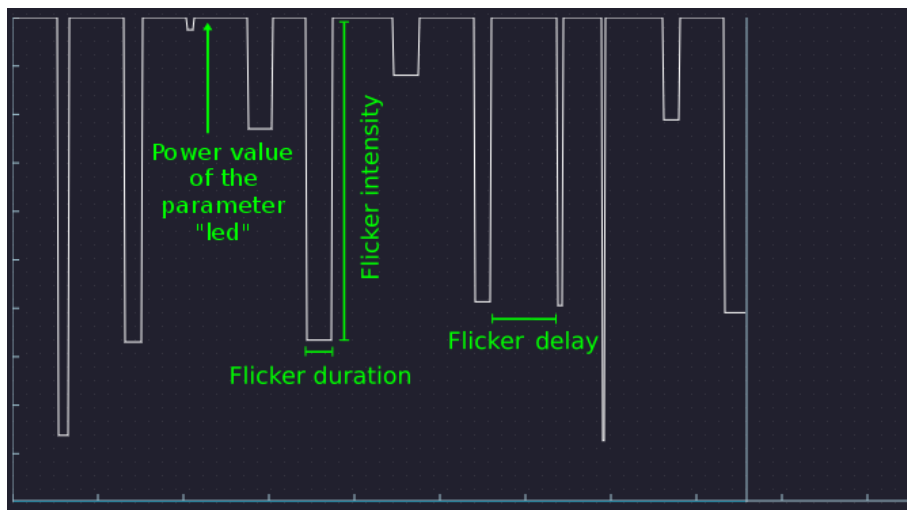
In the following, we give an overview of the parameters of the flicker part and a short description. The parameters are marked by different colors. These indicate whether they're for users who only want to change a few parameters (black) or for more advanced users who want to unleash the full-blown effect engine (green).

PARAMETER	DESCRIPTION
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

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 is a screenshot from our online configurator and 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.
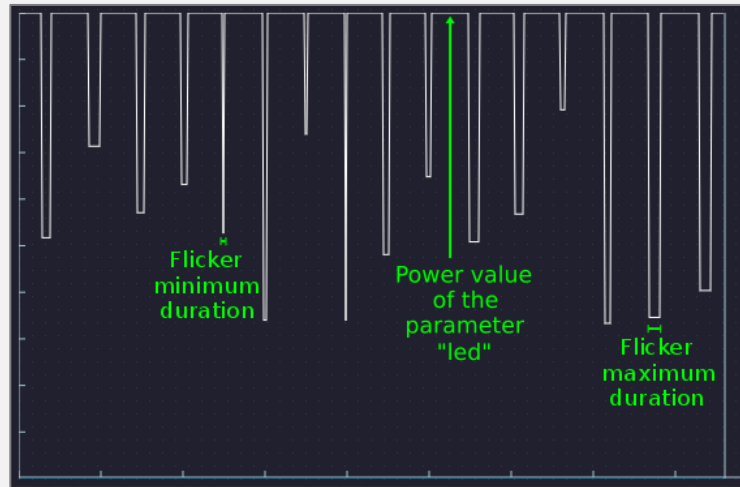
The picture above is a screenshot from our online configurator and illustrates the meaning of both the "flicker_min_duration" and the "flicker_max_duration" parameter. It shows the change of the LED power over time using a flicker effect with the same duration parameters as in channel 1 in the previous 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.
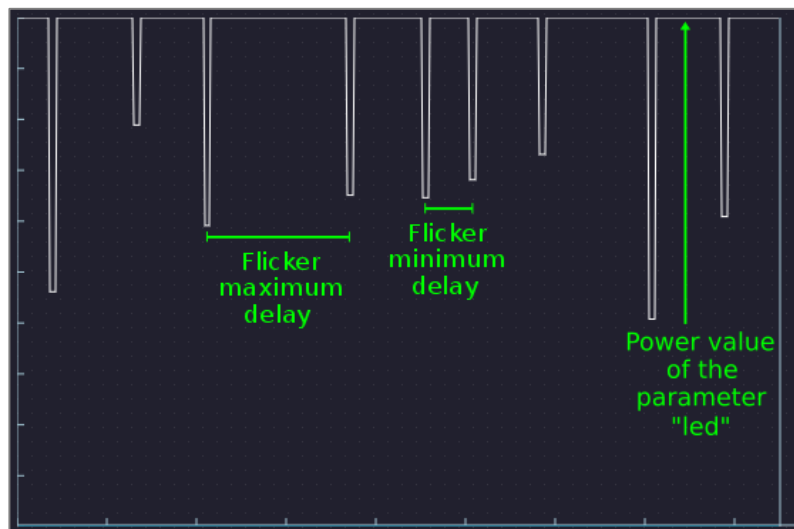
35

The picture above is a screenshot from our online configurator and 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 is a screenshot from our online configurator and 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.*
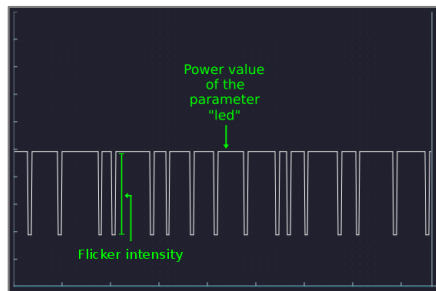
subtractive type (0)   additive type (1)
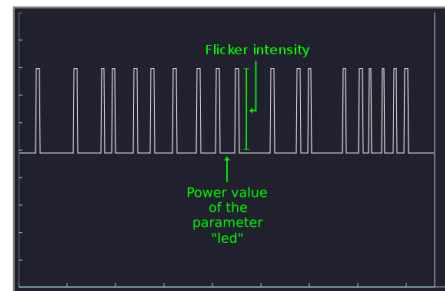
mixed type (2)
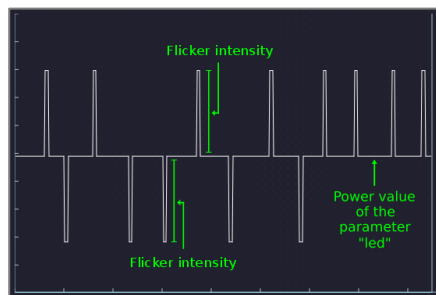
The pictures above are screenshots from our online configurator and illustrate the meaning of the "flicker type" parameter. It shows the change of the LED power over time using a subtractive, additive or mixed flicker effect, respectively. As you can see, a subtractive flicker effect creates drops in 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.
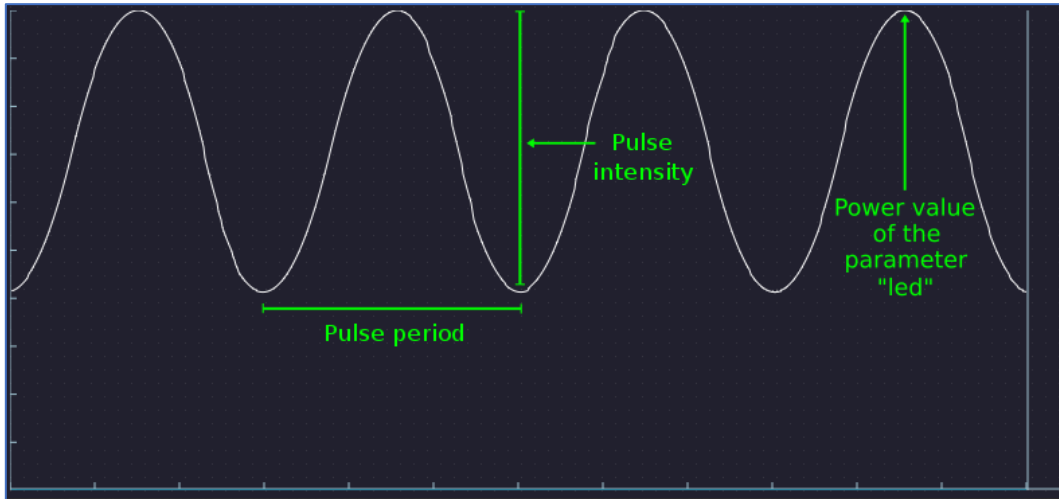
In the following, we give a list of all the parameters of the pulse part and a short description. Similarly to the flicker part, the parameters are marked by different colors. These indicate whether they're for users who only want to change a few parameters (black) or for more advanced users who want 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.
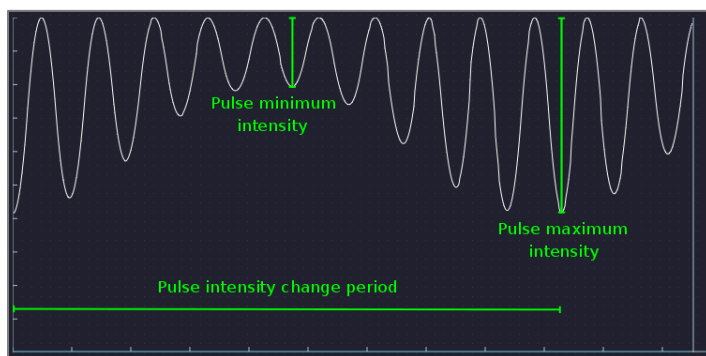
The picture above is a screenshot from our online configurator and illustrates the meaning of the "pulse_min_intensity", "pulse_max_intensity" and "pulse_intensity_change_period" parameter. It shows the change of the LED power over time using a pulse effect with the same parameter value as in channel 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 is a screenshot from our online configurator and 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.
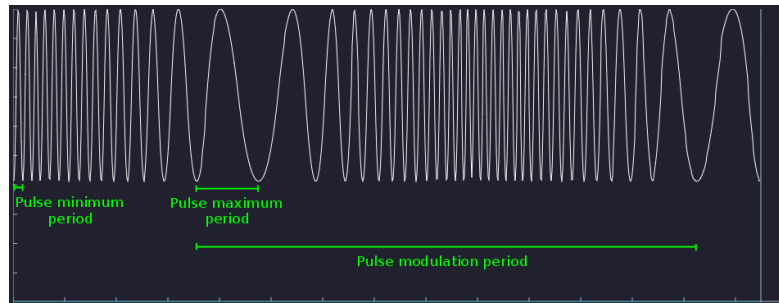
Example:
pulse_shape=0,1,0,0

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.
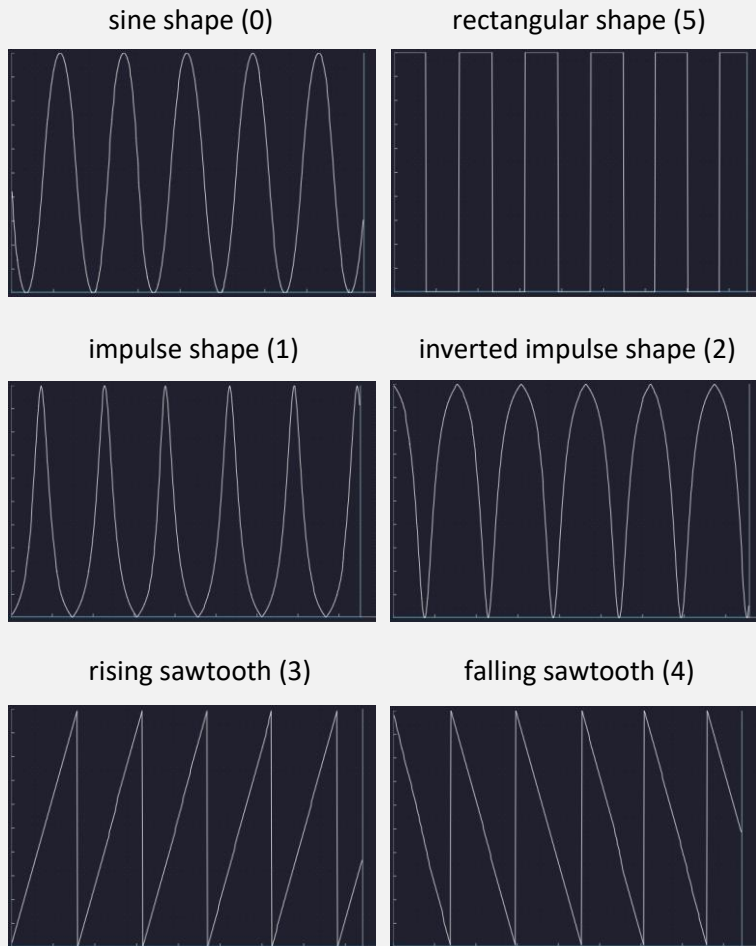
sine shape (0)   rectangular shape (5)

impulse shape (1)   inverted impulse shape (2)

rising sawtooth (3)   falling sawtooth (4)

The picture above is a screenshot from our online configurator and 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.

Neopixel

As mentioned before, the structure of the Neopixel blade styles comprises a "background.txt" file, arbitrarily many "ontop[number].txt" files and arbitrarily many "color[number].txt" files. The Neopixel light effects engine of our board uses these two different types of effects in order to provide you with the highest possible customization. You can imagine background effects and on top effects as different layers of effects. The background effect determines basic characteristics like a flicker effect or a pulse effect. The on top effect is added on top of the background effect as its name already suggests. A possible on top effect is an unstable blade effect. If you have a slightly flickering blade as background effect and add an unstable blade effect as on top effect, this will create a slightly flickering red blade with an unstable appearance. In general, on top effects modify the spatial structure of the underlying background effect. The background effect and on top effect files also consist of 10 different blocks of similar structure. Each of these blocks corresponds to one of the 10 different states the saber can be in: ignition, basic, swing, clash, stab, spin, lockup, melt, blaster block or force push. By adjusting the parameters in a certain block, you can customize the light effects of the corresponding saber state.

As the set of parameters that define the light effects is almost the same for all of the states, we consider the basic state as an example in the following. This is the state the saber is in when no motion is triggered. Le's first have a look at the "background.txt" file. Its 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
```

```
#_____
#
#Fading
#_____


cl_duration=100
cl_fade_out=60
cl_fade_out_type=1
cl_size=200
cl_min_position=0
cl_max_position=100
cl_smooth=10
cl_speed=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 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.

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.

48

- 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 and a value of 1 enables it. 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. *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*
ig_base_flash_duration	*In this case, the aggressivity of the flash at the bottom of the blade is 70.* the bottom of the blade is 70.ig_base_flash_duration The base factivated, 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.
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 value the 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*

PARAMETER	DESCRIPTION
	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 v3 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.63 *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.*

Sauron mode for the Wave effect, Color Flow effect and Flame Blade effect

There is a Sauron mode for the Wave effect, Color Flow effect and Flame Blade effect. Let us consider the Flame Blade effect as an example. When using the Sauron mode, a flame extends from the middle of the blade both to its bottom and to its tip or vice-versa. Furthermore, a dark spot is located in the middle of the blade simulating the eye of Sauron. There are four new parameters in the "background.txt" that control this feature. In the following, they are described:

- flame_blade_sauron=1: This parameter defines which type of the Sauron mode is used. A value of 0 corresponds to a disabled Sauron mode, a value of 1 corresponds to the Sauron mode in which the flames originate at the bottom of the blade and at its tip and a value of 2 corresponds to the Sauron mode in which the flames originate in the middle of the blade.

- flame_blade_sauron_size=0: This parameter defines the size of the dark spot simulating the eye of Sauron.

- flame_blade_sauron_spark=0: The dark spot simulating the eye of Sauron can be made unstable. This parameter defines how unstable it is.

- flame_blade_responsive=0: This parameter defines the responsiveness of the Sauron mode. Values of 0, 1, 2 and 3 correspond to no responsiveness. A value of 4 corresponds to moving the dark spot by tilting the saber, a value of 5 corresponds to moving the dark spot by

twisting the saber and a value of 6 corresponds to moving the dark spot both by tilting and twisting the saber.

The same parameters are available for the Wave effect and the Color Flow effect.

Responsive Lockup

For the lockup effect, you can decide on your own if the lockup location is chosen randomly or affected by gestures in real time. The responsive lockup allows for the location of the lockup to change by tilting the saber, by twisting it or by both tilting and twisting it. You can choose on your own which of both gestures should be used.

There are three parameters in the "background.txt" file that control the responsive lockup feature. In the following, these parameter are described:

- lu_responsive=0: The responsive lockup feature is disabled. The position of the lockup is chosen randomly.

- lu_responsive=1: The position of the lockup is changed by tilting the saber.

- lu_responsive=2: The position of the lockup is changed by twisting the saber.

- lu_responsive=3: The position of the lockup is changed both by tilting the saber and by twisting it.

- lu_tilt_speed=100: Defines the sensitvity of tilting the saber.

- lu_twist_speed=100: Defines the sensitvity of twisting the saber.

Responsive Drag

The responsive drag allows for the size of the illuminated area of the tip to change by twisting the saber.

There are three parameters in the "background.txt" file that control the responsive drag feature. In the following, these parameters are described:

- dr_responsive=0: The responsive drag feature is disabled.

- dr_responsive=1: The size of the illuminated area of the tip is changed by twisting the saber.

- dr_twist_speed=100: Defines the sensitvity of twisting the saber.

Responsive Melt

The responsive melt allows for the area of the tip that simulates the melting and its color to change by tilting the saber, by twisting it or by both tilting and twisting it. You can choose on your own which of both gestures should be used.

There three parameter in the "background.txt" file that control the responsive melt feature. In the following, these parameters are described:

- me_responsive=0: The responsive melt feature is disabled.

- me_responsive=1: The area of the tip that simulates the melting and its color are changed by tilting the saber.

- me_responsive=2: The area of the tip that simulates the melting and its color are changed by twisting the saber.

- me_responsive=3: The area of the tip that simulates the melting and its color are changed both by tilting the saber and by twisting it.

- me_tilt_speed=100: Defines the sensitvity of tilting the saber.

- me_twist_speed=100: Defines the sensitvity of twisting the saber.

Responsive Blaster Block

For the blaster block effect, you can decide on your own if the blaster blok location is chosen randomly or affected by gestures in real time. The responsive blaster block allows for the location of the blaster block to change by tilting the saber, by twisting it or by both tilting and twisting it. You can choose on your own which of both gestures should be used.

There are three parameter in the "background.txt" file that control the responsive blaster block feature. In the following, these parameters are described:

- bb_responsive=0: The responsive blaster block feature is disabled. The position of the blaster block is chosen randomly.

- bb_responsive=1: The position of the blaster block is changed by tilting the saber.

- bb_responsive=2: The position of the blaster block is changed by twisting the saber.

- bb_responsive=3: The position of the blaster block is changed both by tilting the saber and by twisting it.

- bb_tilt_speed=100: Defines the sensitvity of tilting the saber.

- bb_twist_speed=100: Defines the sensitvity of twisting the saber.

Responsive swing effects for Neopixel

It is possible to customize the swing color and the swing light effects for Neopixel so that they respond to the swing speed in real-time. The faster the swinging of the blade, the more pronounced the swing color and the swing light effects are. The basic blade color and the basic blade light effects are blended with the swing color and the swing light effects, whereby the mixing ration moves towards the swing color and the swing light effects as the swing speed increases. The parameters that define the responsive swing effects are described in the following:

- sw_motion_sensitive=0: If this parameter is set to 0, the responsive swing color and swing light effects are disabled. If it is set to 1, they are enabled.

- sw_min_threshold=1000: Minimum swing speed at which the basic blade color and the basic blade light effects start to blend with the swing color and the swing light effects. Below this swing speed, the blade only displays the basic blade color and the basic blade light effects.

- sw_max_threshold=15000: Swing speed at which the blending of the basic blade color and the basic blade light effects with the swing color and the swing light effects ends. Above this swing speed, the blade only displays the swing color and the swing light effects.

The blending mechanism is illustrated in the sketch below. As we can see, the blade only displays the basic blade color and the basic blade light effects below the swing speed given by sw_min_threshold. The higher the swing speed, the higher the share of the swing color and the swing light effects in the mixing. When the swing speed increases above the swing speed given by sw_max_threshold, the blade only displays the swing color and the swing light effects.
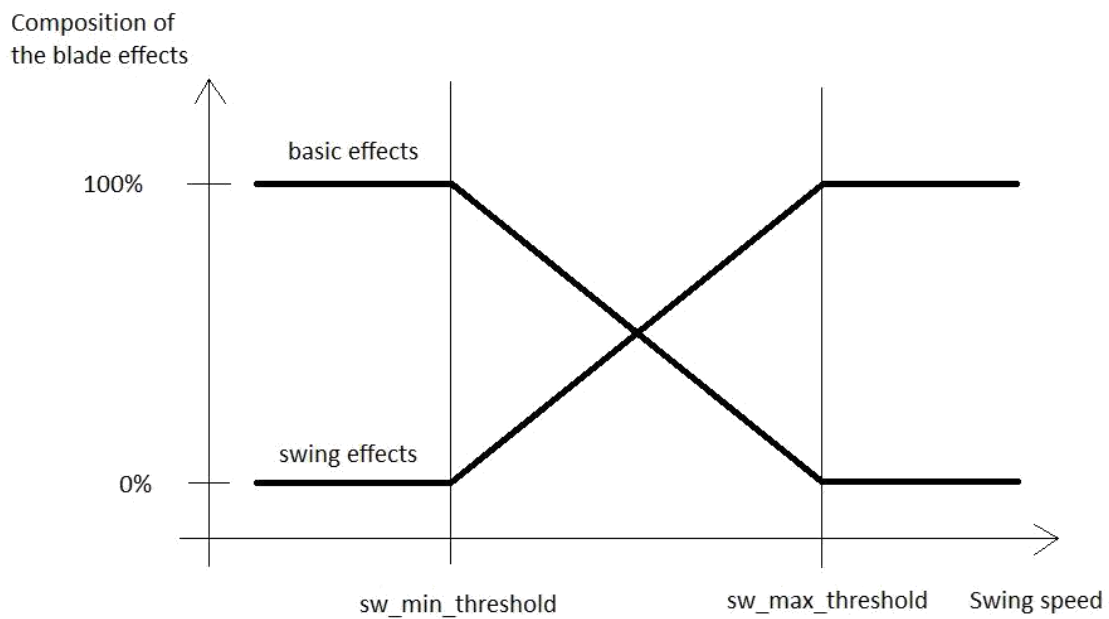
These parameters are located in the "background.txt" file in each blade profile. Furthermore, there are also parameters defining the style of the blending which are the same as the parameters defining the fade out style of the clash effects, stab effects and so on. Have a look at the manual for more details on these parameters.

Preon and postoff light effects for Neopixel

There are preon and postoff light effects available. They are responsive to the actual audio level of the preon and postoff sounds and thus match the sounds amazingly well. There are four new parameters in the "background.txt" that define the preon and postoff light effects. In the following, these parameters are described:

- preon_effect_intensity=10: Defines the intensity of the preon light effects.

- preon_use_basic_effects=1: If it is set to 0, the preon light effects will use the same color and the same effects as during the ignition. This matters only when using custom ignition effects. If it is set to 1, the preon

light effects will use the same effects as when the saber is in the basic state, i.e. after the ignition.

- postoff_effect_intensity=10: Defines the intensity of the postoff light effects.

Color menu

The color menu can be trigerred by holding the power button and then tapping it. In the color menu, you can cycle through all preset color profiles by slowly twisting the saber. Hereby, twisting clockwise goes to the next color profile and twisting counter clockwise goes to the previous color profile. Each time you get to the next color profile, a gear sound indicates this. There are also alternative ways to cycle through the different color profiles in the color menu. These are defined by the parameter "color_menu". The parameter "color_menu_twist_speed" describes the sensitivity of twisting the saber. Both parameters are described in the following:

- color_menu=0: The color menu is disabled.

- color_menu=1: When in the color menu, you can cycle through the different color profiles by slowly twisting the saber. Hereby, twisting clockwise goes to the next color profile and twisting counter clockwise goes to the previous color profile.

- color_menu=2: When in the color menu, you can cycle through the different color profiles by slowly twisting the saber. If you point the saber upwards while doing this, you go to the next color profile. If you point the saber downwards while doing this, you go to the previous color profile. Hereby, it does not matter if you twist the saber clockwise or counter clockwise. Both directions lead to the next or previous color profile depending on the orientation of the saber. This makes it possible to cycle through the color profiles without having to twist the saber too far into one direction. If you want to cycle through the color profiles, you could twist the saber clockwise first and then counter clockwise for example.

- color_menu=3: When in the color menu, you can cycle through the different color profiles by tapping the power button. If you point the saber upwards while doing this, you go to the next color profile. If you point the saber downwards while doing this, you go to the previous color profile.

- color_menu_twist_speed=100: Defines the sensitvity of twisting the saber.

For two momentary buttons setups, the color profile can also be changed on the fly. It is possible to change it by tapping the power button while holding the auxiliary button.

The different color menu modes for one button setups are described in the following video: https://www.youtube.com/watch?v=76x9M9PZ4VA

The different color menu modes for two buttons setups are described in the following video: https://www.youtube.com/watch?v=aH1z84HTAJM

Effect menu

As there are small differences between setups with two momentary buttons and setups with one momentary button, let's consider them separately in the following.

Two momentary buttons

Let's first come to the two momentary buttons setup. The effect menu can be trigerred by holding the power button and then holding it again. Point the saber upwards while doing this. In the effect menu, you can cycle through the blade profiles and the on top effects. To go to the next blade profile, tap the power button while pointing the saber upwards. To go to the previous blade profile, tap the power button while pointing the saber downwards. If you want to go to the next on top effect, tap the auxiliary button while pointing the saber upwards. If you want to go to the previous on top effect, tap the auxiliary button while pointing the saber downwards. It is also possible to cycle through the blade profiles by slowly twisting the saber. The different ways to cycle through the blade profiles are defined by the parameter "effect_menu". The parameter "effect_menu_twist_speed" describes the sensitivity of twisting the saber. Both parameters are described in the following:

- effect_menu=0: The effect menu is disabled.

- effect_menu=1: When in the effect menu, you can cycle through the different blade profiles by slowly twisting the saber. Hereby, twisting clockwise goes to the next blade profile and twisting counter clockwise goes to the previous blade profile.

- effect_menu=2: When in the effect menu, you can cycle through the different blade profiles by slowly twisting the saber. If you point the saber upwards while doing this, you go to the next blade profile. If you point the saber downwards while doing this, you go to the previous blade profile. Hereby, it does not matter if you twist the saber clockwise or counter clockwise. Both directions lead to the next or previous blade profile depending on the orientation of the saber. This makes it possible to cycle through the blade profiles without having to twist the saber too far into one direction. If you want to cycle through the blade profiles, you could twist the saber clockwise first and then counter clockwise for example.

- effect_menu=3: When in the effect menu, you can change the blade profile by tapping the power button. If you point the saber upwards while doing this, you go to the next blade profile. If you point the saber downwards while doing this, you go to the previous blade profile.

- effect_menu_twist_speed=100: Defines the sensitvity of twisting the saber.

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

One momentary button

Now, let's consider the one momentary button setup. The effect menu can be trigerred by holding the button and then holding it again. Point the saber upwards while doing this. In the effect menu, you can cycle through the blade profiles and the on top effects. To go to the next blade profile, twist your saber quickly while pointing it upwards. To go to the previous blade profile, twist your saber quickly while pointing it downwards. If you want to go to the next on top effect, tap the button while pointing the saber upwards. If you want to go to the previous on top effect, tap the button while pointing the saber downwards. It is also possible to cycle through the blade profiles by slowly twisting the saber. The different ways to cycle through the blade profiles are defined by the parameter "effect_menu". The parameter "effect_menu_twist_speed" describes the sensitivity of twisting the saber. Both parameters are described in the following:

- effect_menu=0: The effect menu is disabled.

- effect_menu=1: When in the effect menu, you can cycle through the different blade profiles by slowly twisting the saber. Hereby, twisting clockwise goes to the next blade profile and twisting counter clockwise goes to the previous blade profile.

- effect_menu=2: When in the effect menu, you can cycle through the different blade profiles by slowly twisting the saber. If you point the saber upwards while doing this, you go to the next blade profile. If you point the saber downwards while doing this, you go to the previous blade profile. Hereby, it does not matter if you twist the saber clockwise or counter clockwise. Both directions lead to the next or previous blade profile depending on the orientation of the saber. This makes it possible to cycle through the blade profiles without having to twist the saber too far into one direction. If you want to cycle through the blade profiles, you could twist the saber clockwise first and then counter clockwise for example.

- effect_menu=3: When in the effect menu, you can change the blade profile by quickly twisting the saber. If you point the saber upwards while doing this, you go to the next blade profile. If you point the saber downwards while doing this, you go to the previous blade profile.

- effect_menu_twist_speed=100: Defines the sensitvity of twisting the saber.

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

Accent Profiles

Similar to blade styles that define the light effects displayed by your blade, we introduced so-called accent profiles that contain all the parameters describing the light effects of your accent LEDs. As both standard and Neopixel accent LEDs are treated in the same way, we will use the general term accent LEDs in the following. There is an accent folder inside each slot folder that contains arbitrarily many accent profiles that can be changed on the fly. That way, you can customize the behavior of your accent LEDs without needing to change anything on the SD card. There are two different types of accent profiles: standby accent profiles and ignited accent profiles. The former define the behavior of your accent LEDs while the blade is not ignited. The latter define the behavior of your accent LEDs while the blade is ignited. The standby accent profiles are labelled "standby[number].txt" and the ignited accent profiles are labelled "ignited[number].txt". The screenshots below show the new accent folder inside each slot folder and the content of the new accent folder.

The standby accent profiles and the ignited accent profiles contain the same parameters and consist of three sections: the section for the accent LEDs 1 - 4, the section for the accent LEDs 5 - 8 and the sequencer section. When reading the following descriptions, please keep in mind that the Golden Harvest v3 currently supports five dedicated standard accent LEDs (driven by the pads LED5, LED6, LED7, LED8 and LED9) and even 8 Neopixel accent LEDs. For standard accent LEDs, the accent LEDs 1 - 4 in the accent fonts are mapped to the pads LED5, LED6, LED7 and LED8 and the accent LED 5 in the accent profiles is mapped to the pad LED9 of the Golden Harvest v3. Let's first have a look at the section for the accent LEDs 1 - 4 which is shown in the picture below. All parameters that belong to this section have the prefix "a1_".

```
#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 of the up to 8 Neopixel accent LEDs. *Example:* *a2_color1=255,0,0* *In this case, the color of the fifth of the up to 8 Neopixel accent LEDs is red.*
a2_color2	This parameter is only active for Neopixel accent LEDs. It defines the color of the sixth of the up to 8 Neopixel accent LEDs. *Example:* *a2_color2=0,255,0* *In this case, the color of the sixth of the up to 8 Neopixel accent LEDs is green.*

a2_color3	This parameter is only active for Neopixel accent LEDs. It defines the color of the seventh of the up to 8 Neopixel accent LEDs. *Example:* *a2_color3=0,0,255* *In this case, the color of the seventh of the up to 8 Neopixel accent LEDs is blue.*
a2_color4	This parameter is only active for Neopixel accent LEDs. It defines the color of the eigth of the up to 8 Neopixel accent LEDs. *Example:* *a2_color4=255,255,255* *In this case, the color of the eigth of the up to 8 Neopixel accent LEDs 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.

Blade mirror feature for Neopixel accents

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

- ac_mirror_blade=1,0,0,1,0,0,0,1: 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. In this example, 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:

   - accent_blade_matching=0: The blade matching feature is disabled.

   - accent_blade_matching=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:

   - accent_menu=0: The accent menu is disabled. When trying to trigger the accent menu, the effect menu will be activated.

   - accent_menu=1: The accent menu is enabled.

Battery Indicator

The Golden Harvest v3 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

It is possible to display a charging indicator via the accent LEDs connected to LED5 - LED9 or via the Pixel accent LEDs. When the battery charge is below 80 %, the LEDs pulse slowly while charging, when it is above 80 % they pulse fast and when it is fully charged they are turned on constantly. Furthermore, a charging indicator sound plays when charging starts and when it finishes. This feature

is only available in combination with the Seedling or with a recharge port that does not disconnect the battery from the board while charging. This feature is still in beta testing. It is controlled by the parameter "accent_battery_indicator" in the "general.txt". In the following, this parameter is described:

- accent_battery_indicator=0: The charging indicator feature is disabled.

- accent_battery_indicator=1: The charging indicator feature is enabled.

Volume Control

The Golden Harvest v3 features a volume control on the fly. After entering the volume control mode, this feature allows to change the volume of the saber by gestures. You can customize the volume control to change volume by tilting the saber, by twisting it or by both tilting and twisting it. If you're using a Neopixel blade, the volume is also displayed via the blade. Hereby, the currently active blade effect is used. When in volume control mode, the maximum possible volume is defined by the parameter "volume" in the "general.txt".

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.
- 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.
- 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 new "override-general.txt" file or the new "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:

- color_profile_edit_mode=0: The color profile edit mode is disabled.

- color_profile_edit_mode=1: When in the color profile edit mode, you can assign a new color to the selected color profile by tilting the saber.

- color_profile_edit_mode=2: When in the color profile edit mode, you can assign a new color to the selected color profile by twisting the saber.

- color_profile_edit_mode=3: When in the color profile edit mode, you can assign a new color to the selected color profile both by tilting the saber and by twisting it.

- color_profile_tilt_speed=100: Defines the sensitvity of tilting the saber.

- color_profile_twist_speed=100: Defines the sensitvity of twisting the saber.

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=ckIQdZf7-4U

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:

- pixel_number_edit_mode=0: A value of 1 enables the pixel number edit mode and a value of 0 disables it.

- pixel_number_twist_speed=100: Defines the sensitvity of twisting the saber.

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:

- ignition_speed_edit_mode=0: A value of 1 enables the ignition speed edit mode and a value of 0 disables it.

- ignition_speed_twist_speed=100: Defines the sensitvity of twisting the saber.

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:

- retraction_speed_edit_mode=0: A value of 1 enables the retraction speed edit mode and a value of 0 disables it.

- retraction_speed_twist_speed=100: Defines the sensitvity of twisting the saber.

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

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. 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:

- battle_configuration_edit_mode=0: A value of 1 enables the battle configuration edit mode and a value of 0 disables it.

- battle_configuration_twist_speed=100: Defines the sensitvity of twisting the saber.

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. 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:

- gesture_controls_edit_mode=0: A value of 1 enables the gesture controls edit mode and a value of 0 disables it.

- gesture_controls_twist_speed=100: Defines the sensitvity of twisting the saber.

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

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:

- effect_font_slot_edit_mode=0: A value of 1 enables the effect_font_slot edit mode and a value of 0 disables it.

- effect_font_slot_twist_speed=100: Defines the sensitvity of twisting the saber.

Blaster block for one momentary button setups

For single momentary button setups, there are two different options to trigger a blaster block available. It is possible to trigger a blaster block by tapping the power button or by swinging the saber after entering the MultiBlast mode by tapping the power button. The new parameters that define this feature are described in the following:

- blaster=0: A blaster block is triggered by tapping the power button.

- blaster=1: A blaster block is triggered by swinging the saber after entering the MultiBlast by tapping the power button. If no swing is performed for a customizable time, the MultiBlast mode is left automatically.

- blaster_disable=2000: This parameter defines the time after which the MultiBlast mode is left automatically if no swing is performed. It is given in milliseconds.

- blaster_threshold=5000: This parameter defines how strong you need to swing the saber to trigger a blaster block in MultiBlast mode.

The different blaster block modes for one momentary button setups are described in the following video: https://www.youtube.com/watch?v=76x9M9PZ4VA

Blaster block for two momentary buttons setups

For two momentary buttons setups, there are two different options to trigger a blaster block available. It is possible to trigger a blaster block by tapping the auxiliary button or by swinging the saber after entering the MultiBlast mode by tapping the auxiliary button. The new parameters that define this feature are described in the following:

- blaster=0: A blaster block is triggered by tapping the auxiliary button.

- blaster=1: A blaster block is triggered by swinging the saber after entering the MultiBlast by tapping the auxiliary button. If no swing is performed for a customizable time, the MultiBlast mode is left automatically.

- blaster_disable=2000: This parameter defines the time after which the MultiBlast mode is left automatically if no swing is performed. It is given in milliseconds.
- blaster_threshold=5000: This parameter defines how strong you need to swing the saber to trigger a blaster block in MultiBlast mode.

The different blaster block modes for two momentary buttons setups are described in the following video: https://www.youtube.com/watch?v=aH1z84HTAJM

Force push for one momentary button setups

For single momentary button setups, there are two different options to trigger a force push available. It is possible to trigger a force push by twisting the saber while holding the button or by swinging the saber while holding the button. The parameter "force_push" allows to choose one of these options and is described in the following:

- force_push=0: A force push is triggered by twisting the saber while holding the button.

- force_push=1: A force push is triggered by swinging the saber while holding the button.

- force_push_threshold=1: This parameter defines how strong you have to swing the saber to trigger a force push if you have chosen to trigger a force push by swinging the saber while holding the button.

The different force push modes for one momentary button setups are described in the following video:

https://www.youtube.com/watch?v=DYc7hDa5_ow

Force push for two momentary buttons setups

For two momentary buttons setups, there are two different options to trigger a force push available. It is possible to trigger a force push by tapping the power button or by swinging the saber while holding the auxiliary button. The parameter "force_push" allows to choose one of these options and is described in the following:

- force_push=0: A force push is triggered by tapping the power button.

- force_push=1: A force push is triggered by swinging the saber while holding the auxiliary button.

- force_push_threshold=1: This parameter defines how strong you have to swing the saber to trigger a force push if you have chosen to trigger a force push by swinging the saber while holding the auxiliary button.

The different blaster block modes for two momentary buttons setups are described in the following video: https://www.youtube.com/watch?v=7DdZ6q7yHLs

Lockup for one momentary button setups

For single momentary button setups, there are two different options to trigger lockup available. It is possible to trigger lockup by holding the power button, releasing it and then triggering a clash or by holding the power button, releasing it and then twisting the saber. The parameter "lockup" allows to choose one of these options and is described in the following:

- lockup=0: Lockup is triggered by holding the power button, releasing it and then twisting the saber.

- lockup=1: Lockup is triggered by holding the power button, releasing it and then triggering a clash.

The different lockup modes for one momentary button setups are described in the following video: https://www.youtube.com/watch?v=7LlKZZ1D7gY

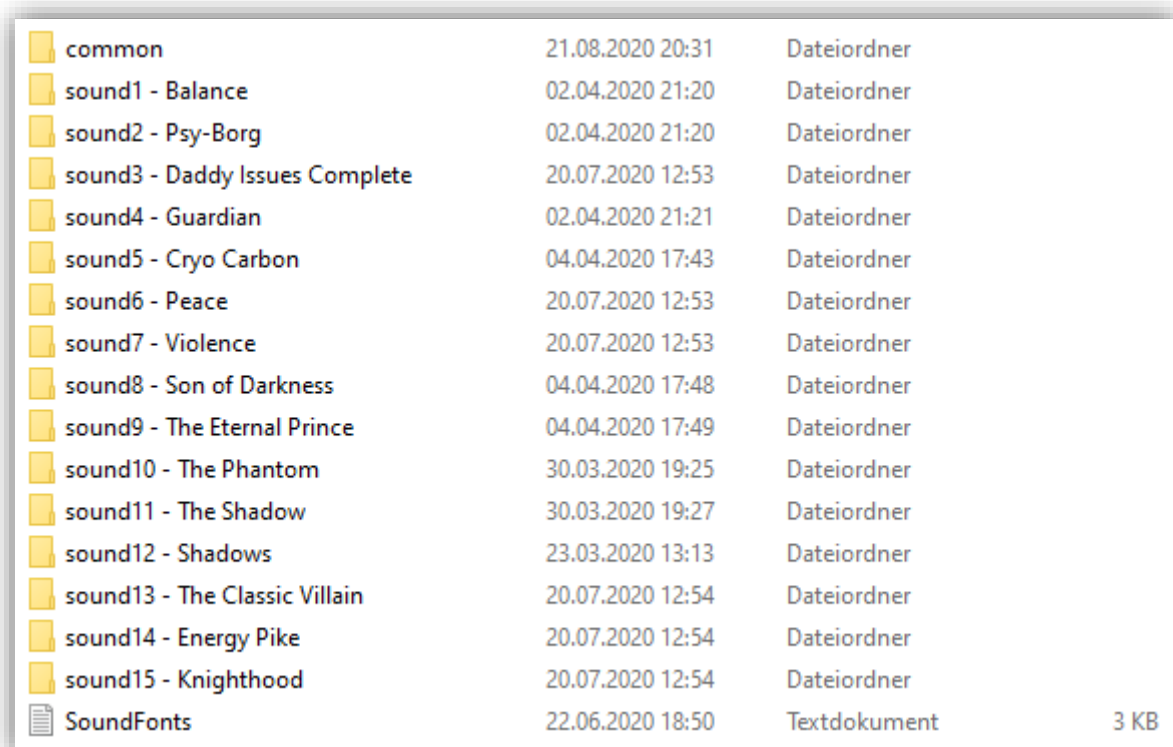## Lockup for two momentary buttons setups

For two momentary buttons setups, there are two different options to trigger lockup available. It is possible to trigger lockup by holding the auxiliary button and then tapping it or by holding

the auxiliary button, releasing it and then triggering a clash. The parameter "lockup" allows to choose one of these options and is described in the following:

- lockup=0: Lockup is triggered by holding the auxiliary button and then tapping it.

- lockup=1: Lockup is triggered by holding the auxiliary button, releasing it and then triggering a clash.

The different lockup modes for two momentary buttons setups are described in the following video: https://www.youtube.com/watch?v=P8-haIr4yI0

Sound Fonts

Just as blade styles define the light effects of your saber, sound fonts define its sound effects. Each sound font contains a set of sound files for all effects our board supports. Thus, they contain 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 many more. Our board supports up to 99 different sound files of each of these sound effects. In case you have more than one sound file for a sound effect, it's chosen randomly which sound file is played when the respective effect is triggered. So, you see there is a huge variety of different sound files that are summarized to a sound font. Sound fonts usually differ with respect to their style and the atmosphere they create. Whereas some sound fonts could give you a Jedi-feeling, others may contain Sith-like sounds. The sound fonts are saved in the folder "SoundFonts" on the microSD card of your saber, each of them as an own folder of the format "sound[number]", whereby [number] enumerates them. It's only important that the folder names begin with "sound[number]". You can assign a description to each folder that can be added after "sound[number]" in the name of the folder (see image below). If you want to add a sound font to your saber, just create a new folder of the format mentioned above with a number that isn't in use yet and put the sound files in there. Hereby, it is important that the enumeration of the sound folders is continuous. In order to ensure that the sound fonts work properly, please make sure that your microSD card is formatted in the FAT32 file system. The microSD card that comes with your Golden Harvest is already formatted in this file system. There are many sound fonts available online. Make sure to visit www.saberfont.com, which is the biggest source for sound fonts. 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

common	21.08.2020 20:31	Dateiordner
sound1 - Balance	02.04.2020 21:20	Dateiordner
sound2 - Psy-Borg	02.04.2020 21:20	Dateiordner
sound3 - Daddy Issues Complete	20.07.2020 12:53	Dateiordner
sound4 - Guardian	02.04.2020 21:21	Dateiordner
sound5 - Cryo Carbon	04.04.2020 17:43	Dateiordner
sound6 - Peace	20.07.2020 12:53	Dateiordner
sound7 - Violence	20.07.2020 12:53	Dateiordner
sound8 - Son of Darkness	04.04.2020 17:48	Dateiordner
sound9 - The Eternal Prince	04.04.2020 17:49	Dateiordner
sound10 - The Phantom	30.03.2020 19:25	Dateiordner
sound11 - The Shadow	30.03.2020 19:27	Dateiordner
sound12 - Shadows	23.03.2020 13:13	Dateiordner
sound13 - The Classic Villain	20.07.2020 12:54	Dateiordner
sound14 - Energy Pike	20.07.2020 12:54	Dateiordner
sound15 - Knighthood	20.07.2020 12:54	Dateiordner
SoundFonts	22.06.2020 18:50	Textdokument	3 KB

Sound Font Settings File

All sound fonts 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 shown in the picture below.

```
#SaberTec Customizer - Smooth Swing Settings File

#_____
#
#Smooth Swing Settings
#_____

motion_threshold=20
swing_sensitivity=450
swing_sharpness=1.75
swing_max_volume=3
accent_swing_threshold=12000
accent_spin_threshold=20000
hum_max_reduction=75
first_transition_deg=160
second_transition_deg=45

tremolo_probability=20
tremolo_min_frequency=0.01
tremolo_max_frequency=0.1
tremolo_sharpness=0.9

stab_ignition=0
swing_ignition=0

twist_retraction=0

poweron=0
poweroff=0
```

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.

99

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.
tremolo_probability	To enhance the smooth swing experience, the Golden Harvest v3 applies real-time audio effects to the smooth swing sounds. For a higher variability, they are applied with a certain probability to each smooth swing sound. This parameter defines the probability with which a tremolo effect is added to the smooth swing sounds.
tremolo_min_frequency	To enhance the smooth swing experience, the Golden Harvest v3 applies real-time audio effects to the smooth swing sounds. For a higher variability, they are applied with a certain probability to each smooth swing sound. The tremolo effect modulates the volume of the smooth swing sounds with a certain frequency. This modulation frequency increases with the rotation speed of the saber from "tremolo_min_frequency" to "tremolo_max_frequency". This parameter defines the minimum modulation frequency of the tremolo effect.
tremolo_max_frequency	To enhance the smooth swing experience, the Golden Harvest v3 applies real-time audio effects to the smooth swing sounds. For a higher variability, they are applied with a certain probability to each smooth swing sound. The tremolo effect modulates the volume of the smooth swing sounds with a certain frequency. This modulation frequency

	increases with the rotation speed of the saber from "tremolo_min_frequency" to "tremolo_max_frequency". This parameter defines the maximum modulation frequency of the tremolo effect.
tremolo_sharpness	To enhance the smooth swing experience, the Golden Harvest v3 applies real-time audio effects to the smooth swing sounds. For a higher variability, they are applied with a certain probability to each smooth swing sound. The tremolo effect modulates the volume of the smooth swing sounds with a certain frequency. This modulation frequency increases with the rotation speed of the saber from "tremolo_min_frequency" to "tremolo_max_frequency". This parameter defines how sharp the modulation frequency of the tremolo effect increases with the rotation speed of the saber.
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 5 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 5 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.

Default Sound Font Package

Our Golden Harvest v3 comes with 16 default sound fonts that are already saved on the microSD card. The default sound fonts are listed below (in chronological order). Thanks to all of the sound font designers! Without you, this project wouldn't have been possible! Please make sure to explore more of their great work.

1. "Balance" by LordBlako Saber Fonts:
Check out more of the genious sound fonts by LordBlako Saber Fonts on http://www.saberfont.com/Lord-Blako_c_15.html

2. "Psy-Borg" by CrystalSoniX:
Check out more of the amazing work of CrystalSoniX on http://www.saberfont.com/Crystal-Sonix-_c_33.html.

3. "Daddy Issues Complete" by Fourzze FX:
Check out more of the mind-blowing fonts by Fourzze FX on http://repulsecustomsounds.com.

4. "Guardian" by LDN Sabers:
Check out more of the epic fonts by LDN Sabers on http://www.saberfont.com/LDN-Sabers_c_79.html.

5. "Cryo Carbon" by LINK:
Check out more of the awesome fonts by LINK on http://repulsecustomsounds.com.

6. "Peace" by Kyberphonic Fonts:
Check out more of the mind-blowing fonts by Kyberphonic Fonts on https://kyberphonic.bigcartel.com/, https://www.instagram.com/kyberphonic/ and http://www.youtube.com/c/Kyberphonic.

7. "Violence" Kyberphonic Fonts:
Check out more of the amazing fonts by Kyberphonic Fonts on https://kyberphonic.bigcartel.com/, https://www.instagram.com/kyberphonic/ and http://www.youtube.com/c/Kyberphonic.

8. "Son of Darkness" by Dark Path Media:
Check out more of the epic fonts by Dark Path Media on http://www.saberfont.com/Dark-Path-Media_bymfg_34-0-1.html.

9. "The Eternal Prince" by LINK:
Check out more of the awesome fonts by LINK on http://repulsecustomsounds.com.

10. "The Phantom" by Project Fonts:
Check out more of the mind-blowing fonts by Project Fonts on http://www.saberfont.com/Project-Fonts_bymfg_29-0-1.html.

11. "The Shadow" by Project Fonts:
Check out more of the epic fonts by Project Fonts on http://www.saberfont.com/Project-Fonts_bymfg_29-0-1.html.

12. "Shadows" by Echo Studios:
Check out more of the amazing fonts by Echo Studios on http://www.saberfont.com/Echo_bymfg_7-0-1.html.

13. "Dark Harvest" by Blueforce Company:

Check out more of the epic fonts by Blueforce Company on https://www.blueforcecompany.com/ and https://www.youtube.com/channel/UCVSMLABlRYnb7uovd9MOBfw.

15. "The Classic Villain" by Fourzze FX:

Check out more of the mind-blowing fonts by Fourzze FX on http://repulsecustomsounds.com.

15. "Energy Pike" by The Proplicator:

Check out more of the amazing fonts by The Proplicator on http://www.saberfont.com/The-Proplicator_bymfg_33-0-1.html.

16. "Knighthood" by Fourzze FX:

Check out more of the mind-blowing fonts by Fourzze FX on http://repulsecustomsounds.com.

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

Our board 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.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 are up to 99 different files possible for each sound type which are selected randomly when triggered. Our Golden Harvest board 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 Golden Harvest v3 allows you to use your own background music that plays during dueling for example. To add background music to your microSD card, please make sure that the music file you want the Golden Harvest v3 to play is a wave file with 16 bit and 22,050 Hz or 44,100 Hz sampling rate. Furthermore, rename the music file to "track[number].wav", whereby [number] enumerates your track files. To add a track file to your microSD card, please choose a sound font to which you want it to be assigned to and just add the track file to the corresponding "sound[number]" folder. You can have arbitrarily many track files for each sound font as the Golden Harvest v3 has a playlist feature. If you activate the background music as it is described in the section "Menu Navigation", the board will start playing the "track1.wav" file. After the board finished playing this file, it will play the "track2.wav" file and so on. After the board finished playing the last track will, it will play the "track1.wav" again.

The triggering of background music for single momentary button setups is described in the following video: https://www.youtube.com/watch?v=L47oo5a4OxQ

The triggering of background music for two momentary buttons setups is described in the following video: https://www.youtube.com/watch?v=L47oo5a4OxQ

Common Folder

The folder "common" is contained in the "SoundFonts" folder. If a sound font does not contain specific sound files like music tracks for example, the board will check if the missing sound files are present in this new folder and will use them. That way, you could add music tracks to the "common" folder for example and they would be used for all sound fonts except from those that feature own music tracks.

Sound menu

As there are small differences between setups with two momentary buttons and setups with one momentary button, let's consider them separately in the following.

Two momentary buttons

Let's first come to the two momentary buttons setup. The sound menu allows to cycle through the sound fonts quickly and even to jump by a predefined number of sound fonts. To enter the sound menu, hold the auxiliary button and then release it. Point the saber upwards while doing this. A sound file will play notifying you that you entered the sound menu. At the same time, the background music of the sound menu will start to play. To go to the next sound font, tap the auxiliary

button while pointing the saber upwards. To go to the previous sound font, tap the auxiliary button while pointing the saber downwards. If you want to jump forward by a number of sound fonts that is given by the parameter "sound_font_jump", tap the power button while pointing the saber upwards. If you want to jump backwards, tap the power button while pointing the saber downwards. There are also alternative ways to go to the next or previous sound font. The possible options are described by the parameter "sound_menu" and are described in the following:

- sound_menu=1: When in the sound menu, you need to slowly twist the saber to go to the next or the previous sound font. You can go to the next sound font by twisting the saber clockwise and to the previous sound font by twisting the saber counter clockwise.

- sound_menu=2: When in the sound menu, you need to slowly twist the saber to go to the next or the previous sound font. If you point the saber upwards while doing this, you go to the next sound font. If you point the saber downwards while doing this, you go to the previous sound font. Hereby, it does not matter if you twist the saber clockwise or counter clockwise. Both directions lead to the next or previous sound font depending on the orientation of the saber. This makes it possible to cycle through the sound fonts without having to twist the saber too far into one direction. If you want to cycle through the sound fonts, you could twist the saber clockwise first and then counter clockwise for example.

- sound_menu=3: When in the sound menu, you need to tap the auxiliary button to go to the next or the previous sound font depending on the orientation of the saber.

The different sound menu modes for two momentary buttons setups is described in the following video: https://www.youtube.com/watch?v=7yDvKBBNZPM

One momentary button

Now, let's consider the one momentary button setup. The sound menu that allows to cycle through the sound fonts quickly and even to jump by a predefined number of sound fonts. To enter the sound menu, hold the button and then release it. Point the saber upwards while doing this. A sound file will play notifying you that you entered the sound menu. At the same time, the background music of the sound menu will start to play. To go to the next sound font, tap the button while pointing the saber upwards. To go to the previous sound font, tap the button while pointing the saber downwards. If you want to jump forward by a number of sound fonts that is given by the parameter "sound_font_jump", twist your saber quickly while pointing it upwards. If you want to jump backwards, twist your saber quickly while pointing it downwards. There are also alternative ways to go to the next or previous sound font and to jump forward or backward by a number of sound fonts. The possible options are described by the parameter "sound_menu" and are described in the following:

- sound_menu=1: When in the sound menu, you need to slowly twist the saber to go to the next or the previous sound font. You can go to the next sound font by twisting the saber clockwise and to the previous sound font by twisting the

saber counter clockwise. To jump forward or backward by a number of sound fonts, you need to tap the button.

- sound_menu=2: When in the sound menu, you need to slowly twist the saber to go to the next or the previous sound font. If you point the saber upwards while doing this, you go to the next sound font. If you point the saber downwards while doing this, you go to the previous sound font. Hereby, it does not matter if you twist the saber clockwise or counter clockwise. Both directions lead to the next or previous sound font depending on the orientation of the saber. This makes it possible to cycle through the sound fonts without having to twist the saber too far into one direction. If you want to cycle through the sound fonts, you could twist the saber clockwise first and then counter clockwise for example. To jump forward or backward by a number of sound fonts, you need to tap the button.

- sound_menu=3: When in the sound menu, you need to tap the button to go the next or previous sound font depending on the orientation of the saber. To jump forward or backward by a number of sound fonts, you need to quickly twist the saber.

The different sound menu modes for one momentary button setups is described in the following video: https://www.youtube.com/watch?v=NRzsD-MJ7vI

Pairing of sound fonts with blade profiles, color profiles, on top effects and accent profiles

You can pair each sound font with a certain blade profile, color profile, on top effect, standby accent profile and ignited accent profile directly on the SD card. Alternatively, you can still do the pairing on the fly of course. A new file called "prefs[number].txt" was introduced. It is located in each effect font slot and contains parameters that define which blade profile, etc. is assigned to the sound font "sound[number]". The screenshot below shows the content of the file.

```
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:

- preon=0: The preon sounds are disabled.

- preon=1: The preon sounds will be played with every ignition independent of how it was triggered.

- preon=2: The preon sounds will be played only when you ignite the saber while pointing it downwards. When igniting the saber while pointing it upwards, the preon sounds will not be played.

- preon=3: The preon sounds will be played only when you ignite the saber while pointing it upwards. When igniting the saber while pointing it downwards, the preon sounds will not be played.

- preon=4: The preon sounds will be played only when you ignite the saber with tapping the power button. If you ignite the saber with gestures, the preon sounds will not be played.

- preon=5: The preon sounds will be played only when you ignite the saber with gestures. If you ignite the saber with tapping the power button, the preon sounds will not be played.

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:

- postoff=0: The postoff sounds are disabled.

- postoff=1: The postoff sounds will be played with every ignition independent of how it was triggered.

- postoff=2: The postoff sounds will be played only when you ignite the saber while pointing it downwards. When igniting the saber while pointing it upwards, the postoff sounds will not be played.

- postoff=3: The postoff sounds will be played only when you ignite the saber while pointing it upwards. When igniting the saber while pointing it downwards, the postoff sounds will not be played.

- postoff=4: The postoff sounds will be played only when you ignite the saber with tapping the power button. If you ignite the saber with gestures, the postoff sounds will not be played.

- postoff=5: The postoff sounds will be played only when you ignite the saber with gestures. If you ignite the saber with tapping the power button, the postoff sounds will not be played.

Firmware Updates

The set of features of the Golden Harvest v3 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 Module if you do not want to remove the microSD card.

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. Delete the file "UPDATE_D.DAT" from your microSD card (if this file is present).
4. Download the latest firmware update from our website: https://sabertec.net/downloads/
5. Extract the .zip folder and put the "update.dat" file on your microSD card.
6. Insert the microSD card back into your board.
7. Wait for at least 2 minutes before powering your board again (by inserting the battery or pulling the kill key). After powering it again, it takes approximately 10 - 20 s before the update is finished. If the boot sound is played after this time, the update was successful.

Installing Using the Seedling Module

To install a firmware update using the Seedling Module, 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. Delete the file "UPDATE_D.DAT" from your microSD card (if this file is present).
4. Download the latest firmware update from our website: https://sabertec.net/downloads/
5. Extract the .zip folder and put the "update.dat" file on your microSD card.
6. Unplug the USB connection to your computer. Now, it takes approximately 10 - 20 s before the update is finished. If the boot sound is played after this time, the update was successful.

USB Charging and File Transfer Using the Seedling Module

The Golden Harvest v3 supports USB 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, just choose the parameter "usb=1" in the "general.txt" file. Please follow the wiring diagram for the Seedling module and make sure that you do not add any accent LEDs to the D + and D - pads that are necessary for USB file transfer. To enter the mass storage device mode, connect your Seedling Module via USB to your computer when your Golden Harvest v3 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 Module 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 file transfer is currently supported on Windows 7 and Windows 10.

When charging your battery using the Seedling Module, make sure to turn your blade LEDs off before charging and never leave the charging unsupervised. Please do not try to connect your saber to USB while the blade LEDs are turned on. Also make sure to use the batteries we recommend in the section "Battery Recommendation". We cannot be help responsible for any damage inflicted by the batteries.

If you have discharged your battery completely, make sure to wait for 5 minutes before you charge your saber using the Seedling Module. You know that the battery is fully depleted by the fact that you cannot ignite your saber any more and the switch LED is turned off. We recommend not to discharge your battery completely to increase its lifetime. In general, make sure to turn your blade LEDs off

before and during charging and never leave the saber unsupervised during charging. We cannot be held responsible for any damage inflicted by the batteries.

Menu Navigation

Navigation Using Two Momentary Buttons

The two buttons are reffered to as auxiliary and power button according to the wiring examples above. This button mode can be selected by setting 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
∟ 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
∟ 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
Wake Up From Deep Sleep	Tap the power button
Reset	Hold the power button for 10 s

* 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 Only One Momentary Button

As mentioned above, all of the features of our board are accessible even with using only one momentary button. This button mode can be selected by setting 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
∟ 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
∟ 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
Wake Up From Deep Sleep	Tap the button
Reset	Hold the button for 10 s

* 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

- My blade is blinking white and does not respond to any button taps. How can I fix that?
  Answer: The white blinking indicates that your microSD card is not inserted properly or some files or folders are missing on it.
- There are no blaster and lockup effects in Neopixel setup for example. How can I fix that?
  Answer: The most probable reason for this is that you forgot to activate the Neopixel light effects on your microSD card. Please follow the steps described in the section "Choosing Your Light Effects"
- There is no sound or stuttering sound. How can I fix that?
  Answer: Please recharge your saber.
- Some sound files are not played or some sound fonts are not recognized. How can I fix that?
  Answer: Please check that your sound font folders and the sound file names are enumerated continuously and check the naming of the sound files.
- The Seedling Module is connected to my computer, but even after 10 minutes it does not recognize the microSD card as mass storage device. How can I fix that?
  Answer: Please check the device manager, look for an "unknown device", click on it with the right mouse button and uninstall the "unknown device". After that, you can try to reconnect the Seedling Module to your computer. Also make sure that you are using a computer with Windows 7 or Windows 10.