

## Downloading the new firmware update

To install the new firmware update, make sure to perform the following steps:

1. Remove the SD card from your board and insert it into your computer.
2. Open the general.txt and the settings.txt with a text editor and add the new parameters described below. It does not matter where you insert them, but make sure that each parameter has its own line and that you don't insert any space between the parameter name and the "=" sign and between the "=" sign and the number.
3. Download the new UserInterfaceSounds folder and put it on the SD card after unzipping it:  
<https://sabertec.net/wp-content/uploads/2022/02/UserInterfaceSounds.zip>
4. Download the new default blade insert and blade remove sounds and add them to the "common" folder in the "SoundFonts" folder:  
<https://sabertec.net/wp-content/uploads/2022/01/sound.zip>
5. Download the update.dat file and put it on the SD card:  
[https://sabertec.net/wp-content/uploads/2022/05/APP\\_25\\_05-17\\_29.zip](https://sabertec.net/wp-content/uploads/2022/05/APP_25_05-17_29.zip)
6. The firmware update will be automatically installed after rebooting the board.

## New parameters of the general.txt

Before installing the new firmware update, add the following parameters to your general.txt:

```
ignition_type_edit_mode=1
ignition_type_twist_speed=200
retraction_type_edit_mode=1
retraction_type_twist_speed=200
charging_indicator=0
charging_threshold=25
accent_battery_indicator=0
data=0
blade_detect=0
blade_detect_protection=0
pixel_slot=9
in_hilt_slot=3
```

Customize the values according to the description of the respective feature.

## New parameters of the background.txt

Before installing the new firmware update, add the following parameters to your background.txt in each blade folder:

```
ig_center=0
re_center=0
```

```
re_stuttering=0  
re_stuttering_speed=200
```

Customize the values according to the description of the respective feature.

## New parameters of the settings.txt

Before installing the new firmware update, add the following parameters to your settings.txt in each sound font:

```
side_blade_delay=100  
side_blade_poweron=100
```

Customize the values according to the description of the respective feature.

## New features introduced by firmware v3.34

### Charging indicator

It is now possible to play a charging indicator sound when inserting the charger into the Seedling or into the RCP. If you're using a RCP, it must have only two pins so that it doesn't disconnect the board from the battery during charging. To play the sound, the board must not be in deep sleep when inserting the charger of course. This feature is controlled by the new parameters "charging\_indicator" and "charging\_threshold" in the general.txt and is described in the following:

- charging\_indicator=0: No charging indicator sound is played.
- charging\_indicator=1: A charging indicator sound is played.
- charging\_threshold=25: This parameter defines the sensitivity of the detection of inserting a charger. The smaller the value the more sensitive it is.

### Accent battery indicator

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

- accent\_battery\_indicator=0: The accent LEDs don't display the battery status, but behave as defined in the standby accent profile and in the ignited accent profile.
- accent\_battery\_indicator=1: The accent LEDs display the battery status when the saber is not ignited.

- `accent_battery_indicator=2`: The accent LEDs display the battery status both when the saber is not ignited and when it is ignited.

## Second data pad for Neopixel

A second data pad for Neopixel is now supported. It is the LED7+ pad. When connecting Neopixel to this pad make sure to add a 470 Ohm resistor in series to the data line. This feature allows to support also crossguard sabers with delayed ignition of the side blades now. It is controlled by a new parameter "data" in the general.txt and is described in the following:

- `data=0`: Only the main data pad is used to drive Neopixel. If you're using Neopixel accent LEDs, these are also driven by this pad. The LED7+ pad can be used to drive a standard accent LED.
- `data=1`: The LED7+ pad can also be used to drive Neopixel. It can be used to drive the side blades of a crossguard saber for example. The Neopixel accent LEDs are driven by the main data pad. If you're using Neopixel accent LEDs, you need to connect them in series with the main blade.
- `data=2`: The LED7+ pad can also be used to drive Neopixel. It can be used to drive the side blades of a crossguard saber for example. The Neopixel accent LEDs are driven by the second data pad. If you're also using side blades for crossguard sabers make sure to connect the Neopixel accent LEDs in series with them.

## Staggered ignition for Neopixel crossguard sabers

The new second data pad for Neopixel allows to setup a staggered ignition for Neopixel crossguard sabers now. The delay of the side blade ignition and the ignition speed of the side blades are customizable. They are controlled by the two new parameters "side\_blade\_delay" and "side\_blade\_poweron" in the settings.txt of each sound font and are described in the following:

- `side_blade_delay=100`: This parameter defines the delay in milliseconds after which the side blades ignite.
- `side_blade_poweron=100`: This parameter defines the ignition duration of the side blades, i.e. how much time it takes until they are fully ignited.

## Blade detect

It is now possible to detect whether a Neopixel blade is inserted or in-hilt is used and the board automatically configures itself accordingly. This does not require any resistors in the blade and works also for TriCree to Neopixel adapters. This feature depends on whether the Neopixel connector in your hilt is illuminated or not. In the following, we will describe how this feature works in detail depending on what kind of connector you are using.

### Connectors without illumination or connectors with illumination (separate data line)

When using connectors without illumination or connectors with illumination that have a separate data line, the board can play an insertion sound when a Neopixel blade is inserted and a removal sound when a Neopixel blade is removed. The sounds are "bladeinsert.wav" and "bladeremove.wav" and can be sound font specific. The board must not be in deep sleep and the saber must not be ignited for the blade detect to work. If your connector is illuminated, connect the data line of the Neopixel accent LEDs to the second data pad of the board. This feature is controlled by the three new parameters "blade\_detect", "pixel\_slot" and "in\_hilt\_slot" in the general.txt and is described in the following:

- `blade_detect=0`: The blade detect feature is disabled.

- blade\_detect=1: The blade detect feature is enabled. The board plays an insertion sound when a Neopixel blade is inserted and a removal sound when a Neopixel is removed. No sound is played when inserting or removing an in-hilt adapter. The board automatically configures itself for Neopixel or in-hilt.
- blade\_detect=2: The blade detect feature is enabled. The board plays an insertion sound when a Neopixel blade is inserted and a removal sound when a Neopixel is removed. No sound is played when inserting or removing an in-hilt adapter. The board does not automatically configure itself for Neopixel or in-hilt.
- blade\_detect\_protection=1000: This parameter defines the time in ms during which the blade detect feature is disabled after the insertion or removal of a Neopixel blade was detected.
- pixel\_slot=9: This parameter defines which effect font slot should be used when a Neopixel blade is detected.
- in\_hilt\_slot=3: This parameter defines which effect font slot should be used when in-hilt is detected.

#### Connectors with illumination (single data line)

When using connectors with illumination that have a single data line for the Neopixel accent LEDs on the connector and the blade, the board automatically configures itself to Neopixel. If you want to use a TriCree to Neopixel adapter, you need to have a 1k Ohm resistor between data and ground in the adapter in order for it to be detectable by the board. It can play an insertion sound when the TriCree to Neopixel adapter is inserted and a removal sound when the TriCree to Neopixel adapter is removed. The sounds are "bladeinsert.wav" and "bladeremove.wav" and can be sound font specific. Note that for this setup these sounds are swapped, i.e. the insertion sound is played when removing the TriCree to Neopixel adapter and the removal sound is played when inserting the TriCree to Neopixel adapter. The board must not be in deep sleep and the saber must not be ignited for the detection of the TriCree to Neopixel adapter to work. This feature is controlled by the three new parameters "blade\_detect", "pixel\_slot" and "in\_hilt\_slot" in the general.txt and is described in the following:

- blade\_detect=0: The blade detect feature is disabled.
- blade\_detect=1: The blade detect feature is enabled. The board plays an insertion sound when the TriCree to Neopixel adapter is inserted and a removal sound when the TriCree to Neopixel adapter is removed. No sound is played when inserting or removing a Neopixel blade. The board automatically configures itself for Neopixel or in-hilt.
- blade\_detect=2: The blade detect feature is enabled. The board plays an insertion sound when the TriCree to Neopixel adapter is inserted and a removal sound when the TriCree to Neopixel adapter is removed. No sound is played when inserting or removing a Neopixel blade. The board does not automatically configure itself for Neopixel or in-hilt.
- blade\_detect\_protection=1000: This parameter defines the time in ms during which the blade detect feature is disabled after the insertion or removal of the TriCree to Neopixel adapter was detected.

- pixel\_slot=9: This parameter defines which effect font slot should be used when a Neopixel blade is detected.
- in\_hilt\_slot=3: This parameter defines which effect font slot should be used when in-hilt is detected.

### New ignition effect

There is a new ignition effect that modifies how the blade is extended. This feature can be used in combination with the stuttering effect, tip flash and base flash to create unique ignition effects. It is controlled by the new parameter "ig\_center" in the "background.txt" and is described in the following:

- ig\_center=0: The blade ignites from the bottom to the top.
- ig\_center=1: The blade ignites from the center of the blade and extends outwards without rescaling it.
- ig\_center=2: The blade ignites from outwards and extends towards the center of the blade without rescaling it.
- ig\_center=3: The blade ignites from the center of the blade and extends outwards with rescaling it.
- ig\_center=4: The blade ignites from outwards and extends towards the center of the blade with rescaling it.
- ig\_center=5: The blade ignites from the top to the bottom.

### New mode of the stuttering effect

A new mode of the stuttering effect was introduced. This effect is controlled by the parameter "ig\_stuttering" in the "background.txt" and is described in the following:

- ig\_stuttering=0: The stuttering effect is disabled.
- ig\_stuttering=1: The stuttering effect is enabled using mode 1. The stuttering effect can be customized using the parameter ig\_stuttering\_speed.
- ig\_stuttering=2: The stuttering effect is enabled using mode 2.

### New edit mode in the Saber Editor

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

- ignition\_type\_edit\_mode=0: A value of 1 enables the ignition type edit mode and a value of 0 disables it.
- ignition\_type\_twist\_speed=100: Defines the sensitivity of twisting the saber.

### Sauron mode for the Classic effect and Gradient effect

The Sauron mode was extended to the Classic effect and Gradient effect.

### Extended Sauron mode

The Sauron mode was extended. There are more values available for the parameter "flame\_blade\_sauron" in the "background.txt" now. In the following, they are described for the example of using the flame blade effect:

- 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 without rescaling it, a value of 2 corresponds to the Sauron mode in which the flames originate in the middle of the blade without rescaling it, a value of 3 corresponds to the Sauron mode in which the flames originate at the bottom of the blade and at its tip with rescaling it, a value of 4 corresponds to the Sauron mode in which the flames originate in the middle of the blade with rescaling it and a value of 5 corresponds to the Sauron mode in which the flames originate from the top of the blade.

### Smooth Swing enhancement

The firmware was optimized for further increased SD card performance which makes the transitions between different Smooth Swing pairs as smooth as possible.

### SD card error indicator

When the SD card is missing, corrupted or bad, the blade blinks white when using Pixel blades. The brightness of this blinking was reduced so that the current draw is also reduced. This makes sure that the charging is not slowed down if the SD card error code occurs at low voltages.