



## SV08 Toolhead Cover (cools, stays on, eddy compatible)



Will

[VIEW IN BROWSER](#)

updated 20. 4. 2025 | published 20. 4. 2025

### Summary

Unique take focused on staying on (using no-yeet technology) and cooling. Works with eddy, eddy-ng, and stock.

---

[3D Printers](#) > [Other Printer Parts & Upgrades](#)

---

Tags: [sovol](#) [sv08](#)

---

v13.1 is only available in .step. Fusion 360 keeps breaking my stl / 3mf outputs for this version, so STEPs for all.

If you need an STL or 3mf I can add one, just be aware it would need to be repaired with your slicer before printing.

Known issues:

- Case v13.1 has more clearance for the fan. A brand of AliExpress replacement fan has larger blades which needed more clearance. Thanks to Morrowshadow for the detailed comments.
- CN3D x endstop doesn't work out of the box with this shroud. I may design a custom x endstop if I have time. Also thanks to Morrowshadow for the comment.

- Two users have reported "whistling". I have not been able to replicate this issue with my prints... so I am still trying to figure out how to solve it. Case v13.1 seems to solve it for most cases.

### **Summary:**

Designed from the ground up, with a unique take on the cover – prioritizing what it is actually supposed to do – stay on your printer and cool stuff. Fan is at an angle to give much better airflow – and I added an insert to optimize splitting airflow within the fan – reducing deadspots and evening out the flow. Also this has optimized nozzle visibility.

Airflow does graze the silicone sock on the nozzle – but the sock is there for a reason – so doesn't impact performance. Any smaller on the outlets to avoid this would choke the 5020.

### **BOM:**

- 8x 4x2mm magnets (or 4x4mm)
- Stock 5020 fan with front cover removed. Just wiggle it in.
- Optional - 1x round neopixel, wiring, 4 pin JST 1.25 pitch to put into UART port.\*

\*I may add more neopixels or better wiring eventually, if anyone else ends up using this model. For now, there are 2mm holes if you want to use a small screw or bent filament to hold the wires in.

### **Printing:**

- 0.4mm nozzle
- 0.16 layer height
- 0.48 line width
- 15% infill
- PA, PET, ABS, ASA, etc
- 25% cooling for asa / abs
- Supports: Yes, organic
- Brim: recommended for ASA and ABS

Print straight up as provided. Be aware that you need very good plate adhesion for the bottom of the ducts to not fall over. I'd recommend checking the print around layer 50-75 to make sure it is working.

Use supports -- yeah it takes a fair amount, but you will be printing it once and I designed it so the supported areas can't be seen. If you want to figure out how to make it printable without supports, feel free. I tried a few orientations – this orientation seemed to give the best results. Layer lines must be horizontal to the clips, or else the clips will snap off. Do not trust OrcaSlicer's auto-orient – trust me.

Break off the bottom tabs near the back of the model, and in the upper window (if you want). The strip was just added for support. Important! make sure you get all supports out of the fan ducts. The left one takes a bit of elbow grease to break them out.

### **LED wiring and details:**

- Ground comes from left (1st) UART pin
- Signal comes from 3rd UART pin
- 5v comes from right (4th) UART pin

### **Simple white light:**

Create a file called leds.cfg.

Add this text:

```
[neopixel toolhead_leds]
```

```
pin: extra_mcu:PA10
```

```
chain_count: 1
```

```
color_order: GRBW
```

```
initial_RED: 0.2
```

```
initial_GREEN: 0.2
```

```
initial_BLUE: 0.2
```

Add [include leds.cfg] to printer.cfg.

### **More advanced using **stealthburner** macros:**

Create a file called leds.cfg.

Paste this text there:

```
[neopixel sb_leds]
```

```
pin: extra_mcu:PA10
```

```
# The pin connected to the neopixel. This parameter must be provided.
```

```
chain_count: 1
```

```
# The number of Neopixel chips that are "daisy chained" to the
```

```
# provided pin. The default is 1 (which indicates only a single
```

```
# Neopixel is connected to the pin).
```

```
color_order: GRBW
```

```
# Set the pixel order required by the LED hardware. Options are GRB,
```

```
# RGB, GRBW, or RGBW. The default is GRB.
```

```
initial_RED: 0.2
```

```
initial_GREEN: 0.2
initial_BLUE: 0.2
initial_WHITE: 0.0
# Sets the initial LED color of the Neopixel. Each value should be
# between 0.0 and 1.0. The WHITE option is only available on RGBW
# LEDs. The default for each color is 0.0
```

```
# Most configuration for the macros can be done by modifying the
variables in the _sb_vars macro
```

Keep in mind we are only using 1 neopixel vs multiples.

Add this text to your macros (or to LEDs macros):

```
#####
# MACROS #
#####
```

```
# The following status macros are available (these go inside of your
macros):
```

```
#
# STATUS_READY
# STATUS_OFF
# STATUS_BUSY
# STATUS_HEATING
# STATUS_LEVELING
# STATUS_HOMING
# STATUS_CLEANING
# STATUS_MESHING
# STATUS_CALIBRATING_Z
```

```
#
# With additional macros for basic control:
```

```
#
# SET_NOZZLE_LEDS_ON
# SET_LOGO_LEDS_OFF
# SET_NOZZLE_LEDS_OFF
#
```

```
# Contributed by Voron discord users wile.e, Tetsunosuke, and etherwalker
```

```
[gcode_macro _sb_vars]
```

```
# User settings for the StealthBurner status leds. You can change the
status colors and led
```

```
# configurations for the logo and nozzle here.
```

```
variable_colors: {
'logo': { # Colors for logo states
'busy': {'r': 0.4, 'g': 0.0, 'b': 0.0, 'w': 0.0},
'cleaning': {'r': 0.0, 'g': 0.02, 'b': 0.5, 'w': 0.0},
```

```

'calibrating_z': {'r': 0.8, 'g': 0., 'b': 0.35, 'w': 0.0},
'heating': {'r': 0.3, 'g': 0.18, 'b': 0.0, 'w': 0.0},
'homing': {'r': 0.0, 'g': 0.6, 'b': 0.2, 'w': 0.0},
'leveling': {'r': 0.5, 'g': 0.1, 'b': 0.4, 'w': 0.0},
'meshing': {'r': 0.2, 'g': 1.0, 'b': 0.0, 'w': 0.0},
'off': {'r': 0.0, 'g': 0.0, 'b': 0.0, 'w': 0.0},
'printing': {'r': 1.0, 'g': 0.0, 'b': 0.0, 'w': 0.0},
'standby': {'r': 0.01, 'g': 0.01, 'b': 0.01, 'w': 0.1},
},
'nozzle': { # Colors for nozzle states
'heating': {'r': 0.8, 'g': 0.35, 'b': 0.0, 'w': 0.0},
'off': {'r': 0.0, 'g': 0.0, 'b': 0.0, 'w': 0.0},
'on': {'r': 0.8, 'g': 0.8, 'b': 0.8, 'w': 1.0},
'standby': {'r': 0.6, 'g': 0.0, 'b': 0.0, 'w': 0.0},
},
'thermal': {
'hot': {'r': 1.0, 'g': 0.0, 'b': 0.0, 'w': 0.0},
'cold': {'r': 0.3, 'g': 0.0, 'b': 0.3, 'w': 0.0}
}
}
# variable_logo_led_name: "sv08_leds"
# The name of the addressable LED chain that contains the logo LED(s)
# variable_logo_idx: "1"
# A comma-separated list of indexes LEDs in the logo
variable_nozzle_led_name: "sb_leds"
# The name of the addressable LED chain that contains the nozzle LED(s).
This will
# typically be the same LED chain as the logo.
variable_nozzle_idx: "1"
# A comma-separated list of indexes of LEDs in the nozzle
gcode:
# This section is required. Do Not Delete.

[gcode_macro _set_sb_leds]
gcode:
{% set red = params.RED|default(0)|float %}
{% set green = params.GREEN|default(0)|float %}
{% set blue = params.BLUE|default(0)|float %}
{% set white = params.WHITE|default(0)|float %}
{% set led = params.LED|string %}
{% set idx = (params.IDX|string).split(',') %}
{% set transmit_last = params.TRANSMIT|default(1) %}

{% for led_index in idx %}
{% set transmit=transmit_last if loop.last else 0 %}
set_led led={led} red={red} green={green} blue={blue} white={white}

```

```
index={led_index} transmit={transmit}
{% endfor %}
```

```
[gcode_macro _set_sb_leds_by_name]
```

```
gcode:
```

```
{% set leds_name = params.LEDS %}
{% set color_name = params.COLOR %}
{% set color = printer["gcode_macro_sb_vars"].colors[leds_name]
[color_name] %}
{% set led = printer["gcode_macro_sb_vars"][leds_name + "_led_name"]
%}
{% set idx = printer["gcode_macro_sb_vars"][leds_name + "_idx"] %}
{% set transmit = params.TRANSMIT|default(1) %}
```

```
_set_sb_leds led={led} red={color.r} green={color.g} blue={color.b}
white={color.w} idx="{idx}" transmit={transmit}
```

```
#[gcode_macro _set_logo_leds]
```

```
#gcode:
```

```
# {% set red = params.RED|default(0)|float %}
# {% set green = params.GREEN|default(0)|float %}
# {% set blue = params.BLUE|default(0)|float %}
# {% set white = params.WHITE|default(0)|float %}
# {% set led = printer["gcode_macro_sb_vars"].logo_led_name %}
# {% set idx = printer["gcode_macro_sb_vars"].logo_idx %}
# {% set transmit=params.TRANSMIT|default(1) %}
```

```
# _set_sb_leds led={led} red={red} green={green} blue={blue}
white={white} idx="{idx}" transmit={transmit}
```

```
[gcode_macro _set_nozzle_leds]
```

```
gcode:
```

```
{% set red = params.RED|default(0)|float %}
{% set green = params.GREEN|default(0)|float %}
{% set blue = params.BLUE|default(0)|float %}
{% set white = params.WHITE|default(0)|float %}
{% set led = printer["gcode_macro_sb_vars"].nozzle_led_name %}
{% set idx = printer["gcode_macro_sb_vars"].nozzle_idx %}
{% set transmit=params.TRANSMIT|default(1) %}
```

```
_set_sb_leds led={led} red={red} green={green} blue={blue}
white={white} idx="{idx}" transmit={transmit}
```

```
#[gcode_macro set_logo_leds_off]
```

```
#gcode:
```

```
# {% set transmit=params.TRANSMIT|default(1) %}
# _set_logo_leds red=0 blue=0 green=0 white=0 transmit={transmit}
```

```
[gcode_macro set_nozzle_leds_on]
gcode:
{% set transmit=params.TRANSMIT|default(1) %}
_set_sb_leds_by_name leds="nozzle" color="on" transmit={transmit}
```

```
[gcode_macro set_nozzle_leds_off]
gcode:
{% set transmit=params.TRANSMIT|default(1) %}
_set_sb_leds_by_name leds="nozzle" color="off" transmit={transmit}
```

```
[gcode_macro status_off]
gcode:
# set_logo_leds_off transmit=0
set_nozzle_leds_off
```

```
[gcode_macro status_ready]
gcode:
# _set_sb_leds_by_name leds="logo" color="standby" transmit=0
_set_sb_leds_by_name leds="nozzle" color="standby" transmit=1
```

```
[gcode_macro status_busy]
gcode:
# _set_sb_leds_by_name leds="logo" color="busy" transmit=0
set_nozzle_leds_on
```

```
[gcode_macro status_heating]
gcode:
# _set_sb_leds_by_name leds="logo" color="heating" transmit=0
_set_sb_leds_by_name leds="nozzle" color="heating" transmit=1
```

```
[gcode_macro status_leveling]
gcode:
# _set_sb_leds_by_name leds="logo" color="leveling" transmit=0
set_nozzle_leds_on
```

```
[gcode_macro status_homing]
gcode:
# _set_sb_leds_by_name leds="logo" color="homing" transmit=0
set_nozzle_leds_on
```

```
[gcode_macro status_cleaning]
gcode:
# _set_sb_leds_by_name leds="logo" color="cleaning" transmit=0
set_nozzle_leds_on
```

```
[gcode_macro status_meshing]
gcode:
```

```
# _set_sb_leds_by_name leds="logo" color="meshing" transmit=0
set_nozzle_leds_on

[gcode_macro status_calibrating_z]
gcode:
# _set_sb_leds_by_name leds="logo" color="calibrating_z" transmit=0
set_nozzle_leds_on

[gcode_macro status_printing]
gcode:
# _set_sb_leds_by_name leds="logo" color="printing" transmit=0
set_nozzle_leds_on
```

## Model files



**case-v131-clearance-and-whistling.step**

## License ©



This work is licensed under a  
**Creative Commons (4.0 International License)**

**Attribution—Noncommercial—Share Alike**

- 
- ✗ | Sharing without ATTRIBUTION
  - ✓ | Remix Culture allowed
  - ✗ | Commercial Use
  - ✗ | Free Cultural Works
  - ✗ | Meets Open Definition