

# \*Youtube - Nevermore Controller

Hier entsteht das Installationstutorial für den Nevermore Controller. Vielen Dank an den Ersteller der Software und der englischen Anleitung: [SanaaHamel](#)

Die Anleitung zum Filtersystem wird zeitnah ergänzt:

- Aufbauanleitung Stealthmax Filtersystem: Link folgt demnächst
- Aufbauanleitung Nevermore Mini: [Youtube Anleitung](#)

## Nevermore Kontroller mit Raspberry Pi Pico

Die folgende Anleitung beinhaltet nur die Einrichtung der elektronischen Komponenten.

### Benötigte Hardware

Diese Bauteile werden für den Nevermore Controller benötigt:

#### Zwingende Komponenten:

##### 1. Über Amazon:

- 1x Pico W mit Pinleiste: <https://amzn.to/3vktXTN>
- 2x BME280: <https://amzn.to/3NHYP0E>
- 2x SGP40: derzeitig nicht bestellbar
- Jumper Wire (female to female): <https://amzn.to/3RKTQ8c>

##### 2. Über Aliexpress:

- 2x SGP40/BME240 (2in1 PCB): [https://s.click.aliexpress.com/e/\\_Dmkhbk5](https://s.click.aliexpress.com/e/_Dmkhbk5)  
oder
- 2x SGP40: [https://s.click.aliexpress.com/e/\\_DmCfMXf](https://s.click.aliexpress.com/e/_DmCfMXf)
- 2x BME240 (3xBME240 3.3V): [https://s.click.aliexpress.com/e/\\_Dndm8PX](https://s.click.aliexpress.com/e/_Dndm8PX)

#### Optionale Komponenten (abhängig vom Einsatzzweck):

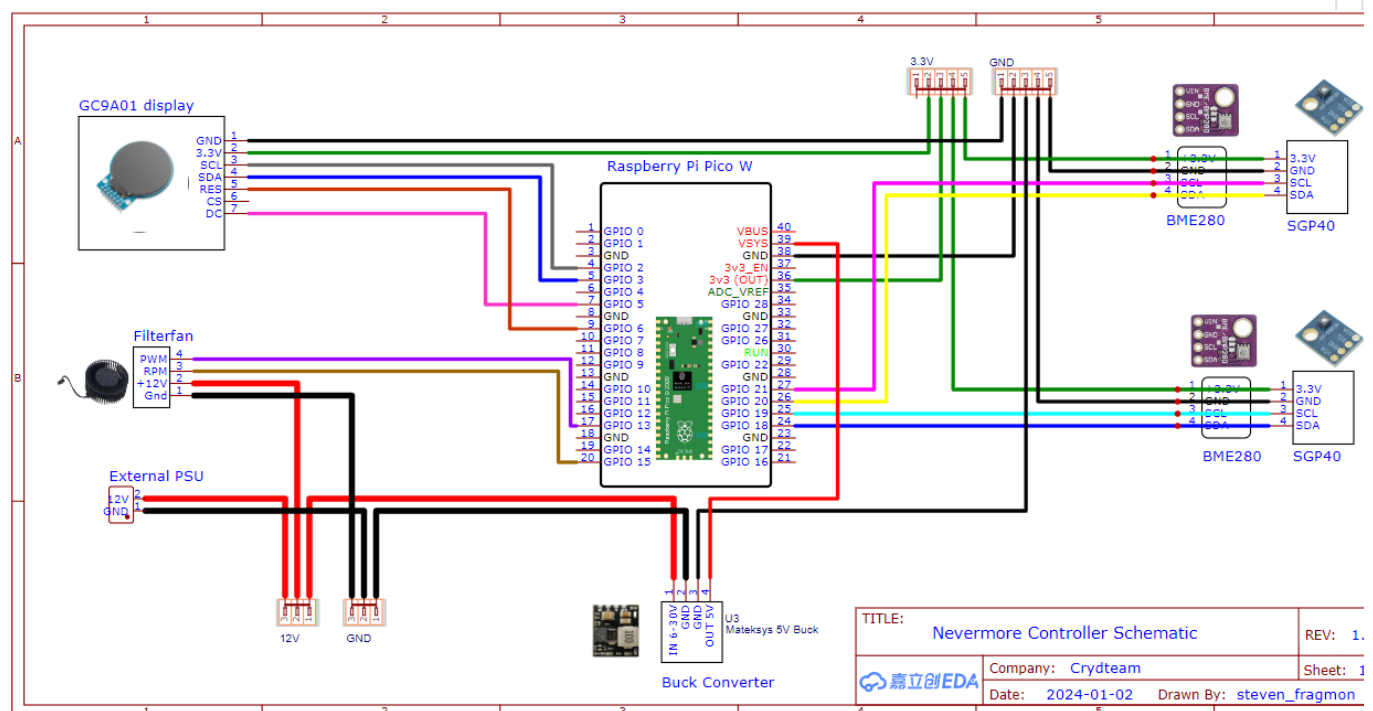
- Display: <https://amzn.to/3tDxj3O>
- Display: <https://www.aliexpress.com/item/1005006111616872.html>

PWM Fan bspw 5V Noctua: <https://amzn.to/3SkVwp4>

\*alle Links sind Affiliate Links womit du kostenfrei den Channel und den Discord unterstützt  
- jeder Cent kommt der Community zu Gute!

## Anschlussbelegung

In dieser Anordnung werden die Sensoren (BME und SGP) durch eine Pinleiste übereinandergestapelt. Der Plastikabstandshalter soll zwischen den Platinen platziert werden. Stelle sicher, dass der SGP40-Sensor nach außen gerichtet ist, um eine optimale Detektion zu gewährleisten. Der Sensor des lilafarbenen Sensors kann zwischen den Leiterplatten montiert sein.



## Verkabelung testen

Um die Verkabelung der Sensoren zu prüfen habe ich ein Micropythonskript geschrieben. Um den I2C-Bus zu überprüfen, gehe wie folgt vor:

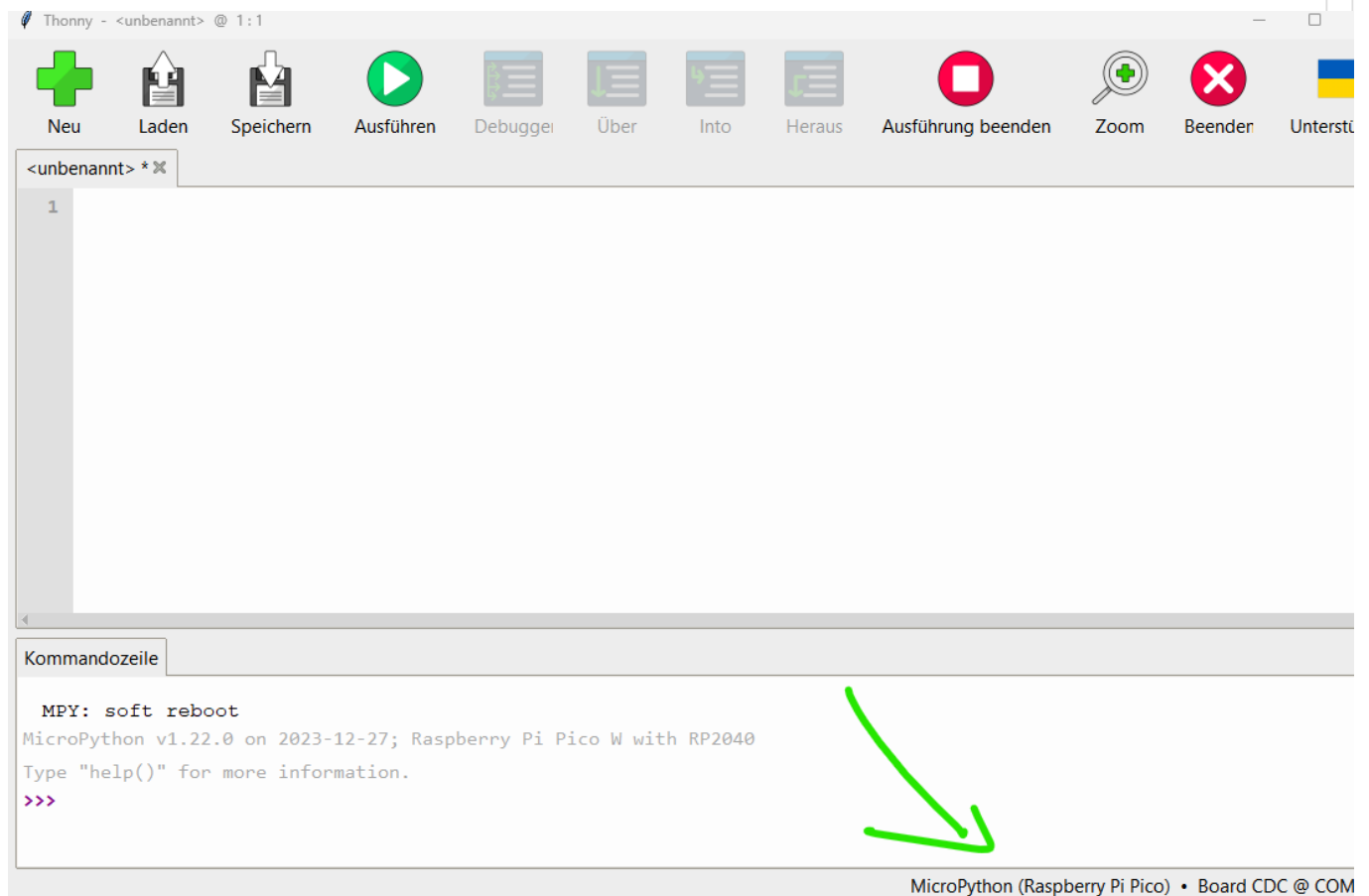
1. Lade MicroPython auf den Raspberry Pi Pico herunter und flashe es.

Link: <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html>

2. Lade Thonny herunter und installiere es.

Link: <https://thonny.org>

### 3. Starte Thonny und wähle deinen Raspberry Pi Pico aus.



### 4. Kopiere und füge diesen Code ein:

```
import machine

# create first i2c bus (Exhaust)
sda_exhaust = machine.Pin(18)
scl_exhaust = machine.Pin(19)
i2c_exhaust = machine.I2C(1, sda=sda_exhaust, scl=scl_exhaust, freq=400000)

# create second i2c bus (Intake)
sda_intake = machine.Pin(20)
scl_intake = machine.Pin(21)
i2c_intake = machine.I2C(0, sda=sda_intake, scl=scl_intake, freq=400000)

def scan_i2c_bus(i2c, bus_name):
    print(f'Scan i2c bus {bus_name}...')
    devices = i2c.scan()
```

```

if len(devices) == 0:
    print(f"No i2c device found on {bus_name} bus!")
else:
    print(f'i2c devices found on {bus_name} bus:', len(devices))

    for device in devices:
        print("Decimal address: ", device, " | Hexa address: ", hex(device))

# Scan I2C-Bus
scan_i2c_bus(i2c_exhaust, "Exhaust")
scan_i2c_bus(i2c_intake, "Intake")

```

5. Überprüfe die Antwort: Wenn die Meldung "Kein I2C-Gerät gefunden" erscheint, solltest du die Verkabelung überprüfen oder den Chip ersetzen. Bei Nutzung von zwei Sensorstacks müssen insgesamt vier Adresse angezeigt werden.

Thonny - <unbenannt> @ 8:25

Neu Laden Speichern Ausführen Debugger Über Into Heraus Ausführung beenden Zoom Beender Unterst

```

<unbenannt> *x
1 import machine
2
3 # Einrichten des ersten I2C-Busses (Exhaust)
4 sda_exhaust = machine.Pin(18)
5 scl_exhaust = machine.Pin(19)
6 i2c_exhaust = machine.I2C(1, sda=sda_exhaust, scl=scl_exhaust, freq=400000)
7
8 # Einrichten des zweiten I2C-Busses (Intake)
9 sda_intake = machine.Pin(20)
10 scl_intake = machine.Pin(21)
11 i2c_intake = machine.I2C(0, sda=sda_intake, scl=scl_intake, freq=400000)
12
13 def scan_i2c_bus(i2c, bus_name):
14     print(f'Scan i2c bus {bus_name}...')
15     devices = i2c.scan()
16
17     if len(devices) == 0:
18         print(f"No i2c device found on {bus_name} bus!")

```

Kommandozeile

```

MPY: soft reboot
Scan i2c bus Exhaust...
No i2c device found on Exhaust bus!
Scan i2c bus Intake...
i2c devices found on Intake bus: 2
Decimal address: 89 | Hexa address: 0x59
Decimal address: 118 | Hexa address: 0x76
>>>

```

MicroPython (Raspberry Pi Pico) • Board CDC @ COM

## Controllersoftware installieren

## 1. Kontrollerfirmare

Lade die [Kontrollerfirmware](#) herunter und flashe diesen auf den Raspberry Pico.

## 2. Kontrollersoftware

Lade die Software über SSH auf den Klipper-Kontroller (Raspberry Pi) und installiere diese mit folgenden Befehlen:

```
cd ~  
git clone https://github.com/SanaaHamel/nevermore-controller  
cd nevermore-controller  
./install-klipper-module.bash
```

Wenn du Mainsail OS verwendest, wird dich das Installationsskript fragen, ob du Bluetooth aktivieren möchtest. Tu dies und starte dann deinen Klipper-Host neu. (z.B. `sudo reboot`).

## 3. Bluetooth Adresse

Wenn du nur einen Nevermore-Controller in Reichweite hast, kannst du die Option `bt_address` in deiner Druckerkonfiguration weglassen und diesen Abschnitt komplett ignorieren. Wenn du jedoch mehrere Bluetooth (BT)-Geräte in Reichweite hast, die wie Kandidaten für einen Nevermore-Controller aussehen, musst du angeben, welches du verwenden möchtest. Dies geschieht, indem du ihre 'Adresse' in der Druckerkonfiguration mit `bt_address: <Adresse>` angibst.

Hinweis: Es ist möglich, aber sehr selten, dass sich die Adresse ändert, wenn ein neues UF2 auf den Pico geflasht wird. Dies wurde einmal beobachtet, nachdem das Pico SDK aktualisiert wurde.

### Methode A - Überprüfen des Klipper-Protokolls

1. In Fehler wird ausgelöst, wenn mehrere Controller in Reichweite sind. Die Fehlermeldung listet alle verfügbaren Adressen der Controller auf.

```
...  
...  
[11:06:36:535560] nevermore - multiple nevermore controllers discovered.  
specify which to use by setting `bt_address: <insert-address-here>` in your klipper  
config.
```

discovered controllers (ordered by signal strength):

address	signal strength
-----	
FA:KE:AD:RE:SS:01	-38 dBm
FA:KE:AD:RE:SS:00	-57 dBm

Config error

Traceback (most recent call last):

File "~/klipper/klippy/klippy.py", line 180, in \_connect

cb()

File "~/klipper/klippy/extras/nevermore.py", line 793, in \_handle\_connect

raise self.printer.config\_error("nevermore failed to connect - timed out")

configparser.Error: nevermore failed to connect - timed out

...

...

## Methode B: - Nur für Linux - über bluetoothctl

Stelle sicher, dass dein Nevermore-Controller eingeschaltet ist und die LED blinkt. (Zeigt an, dass er aktiv ist.)

Öffne ein Terminal und führe aus:

```
bluetoothctl
```

danach

```
scan on
```

aus

Es wird euch folgendes angezeigt, ist für euch aber nicht wichtig.

```
[Nevermore Controller]# scan on
Discovery started
[CHG] Controller 28:00:00:00:00:00 Discovering: yes
[NEW] Device 5A:72:5A:30:F5:90 5A-72-F5-30-F5-90
[NEW] Device 51:31:50:1A:72:1F 51-31-50-1A-72-1F
[NEW] Device 50:73:3D:36:86:EC 50-73-3D-36-86-EC
[NEW] Device 50:04:00:00:00:00 50-04-00-00-00-00
[NEW] Device 65:00:00:00:00:00 65-00-00-00-00-00
[NEW] Device 50:04:00:00:00:00 50-04-00-00-00-00
[NEW] Device 5A:72:5A:30:F5:90 5A-72-F5-30-F5-90
[NEW] Device 51:31:50:1A:72:1F 51-31-50-1A-72-1F
[CHG] Device 28:00:00:00:00:00 Discovering: no
[NEW] Device 5A:72:5A:30:F5:90 5A-72-F5-30-F5-90
[NEW] Device 44:00:00:00:00:00 44-00-00-00-00-00
```

Führe danach

devices

aus.

Es wird euch nun der Nevermore Controller angezeigt. Kopiert euch nun die Bluetooth ID.

```
[Nevermore Controller]# devices
Device 28:00:00:00:00:00 Nevermore Controller
Device 50:73:3D:36:86:EC 50-73-3D-36-86-EC
Device 51:31:50:1A:72:1F 51-31-50-1A-72-1F
Device 50:04:00:00:00:00 50-04-00-00-00-00
Device 65:00:00:00:00:00 65-00-00-00-00-00
Device 50:04:00:00:00:00 50-04-00-00-00-00
Device 5A:72:5A:30:F5:90 5A-72-F5-30-F5-90
Device 51:31:50:1A:72:1F 51-31-50-1A-72-1F
Device 44:00:00:00:00:00 44-00-00-00-00-00
Device 5A:72:5A:30:F5:90 5A-72-F5-30-F5-90
[CHG] Device 28:00:00:00:00:00 Discovering: no
```

#### 4. Anpassung der Printer.cfg

Kopiere die Bluetooth ID in deine Printer.cfg wie folgt:

```
[nevermore]
bt_address: FA:KE:AD:RE:SS:00
```

Beispiel mit auskommentierte Adresse, da nur ein Bluetoothgerät in der Nähe ist:

```
[nevermore]
#bt_address: 28:CD:C1:0F:47:55
#fan_power_coefficient: 0.5
#####
#####
```

## 5. (Option) Aktualisiere die Firmware

Wenn du eine OTA-fähige UF2 auf deinen Controller geflasht hast, kannst du ihn drahtlos aktualisieren. Der Prozess ist einfach:

```
cd ~/nevermore-controller
git pull
./tools/update_ota.py
```

Wenn du `update_ota.py` ausführst, werden fehlende Abhängigkeiten installiert. Dies kann beim ersten Mal je nach Leistungsfähigkeit des Computers eine Weile dauern.

Wenn du mehrere Controller in Reichweite hast, kannst du mit `--bt-address` angeben, welchen du aktualisieren möchtest. Z.B. `./tools/update_ota.py --bt-address XX:XX:XX:XX:XX:XX`

Siehe `./tools/update_ota.py --help` für alle Optionen.

**HINWEIS:** Der Controller startet automatisch neu, wenn er 60 Sekunden lang im Bootloader-Modus untätig bleibt.

Insgesamt solltest du eine ähnliche Ausgabe wie folgende sehen:










```
Tool environment seems up to date.
This program will attempt to update a Nevermore controller.
-----

discovering Nevermores...
connecting to XX:XX:XX:XX:XX:XX
current revision: v0.7.0
sending reboot-to-OTA command...
connecting to device...
requesting device info...
sync w/ device...
trying to update bootloader...
requesting device info...
```



current version: v0.7.0 # in this example it tried to update to the same version

Habt ihr den [Nevermore] plus die ID eingetragen sollte automatisch Nevermore in der Übersicht erscheinen.

<div>  <span>Temperatures</span> <span>PRESETS ▾</span> <span>⚙️</span> <span>▾</span> </div>				
Name	State	Current	Target	
 Extruder	off	22.9°C	0	°C ▾
 Heater Bed	off	24.5°C	0	°C ▾
 Nevermore		100 > 99 20.8 °C > 20.7 °C 995 hPa > 994 hPa 45.2 % > 46.1 % 0 RPM		
 Beacon Coil		24.2°C		
 Chamber		21.9°C		
 Mcu		34.5°C		
 RaspberryPi		64.3°C		
 Toolhead Mcu		29.5°C		

Fehlen die ersten Zahlen (hier 100>99), so wurde der SGP Sensor nicht erkannt. Fehlen Temperatur etc. dann ist der BMP Sensor nicht korrekt angeschlossen.

Wollt ihr eure Sensoren, Lüfter etc individuell einrichten, dann schaut in diese [Dokumentation](#).

## Troubleshooting

Falls es nicht geklappt hat, dann schaue in die [FAQ](#) von Sanaa oder Frage auf unserem Discord nach.

## Nevermore Mini - Display + Sensor

### Benötigte Hardware

Diese Bauteile werden für den Nevermore Controller benötigt:

Zusammenbau	Anzahl
Magnets 6x3mm	8x
m3x6mm BHCS	4x
m3x6 SHCS	6x
M3 Einschmelzmuttern	10x
<b>Montage</b>	
1515/2020 Extrusion: m3x8 SHCS	2x
3030 Extrusion: m4x8mm SHCS	2x
Bambulabs Magnets 6x3mm	1x
<b>Elektronik</b>	
Arctic S4028-15K 12V 4028 Lüfter	1x
AWG 28 Kabel	20cm x Anzahl Kabel
Waveshare RP2040 1,28" Touch Display	1x
SGP40 VOC sensor	1x
BME280 Temp/Humidity Sensor	1x
USB C Stecker zum löten	1x
Wago221	2x
<b>Filter</b>	
HEPA: Nevermore 190x45mm HEPA Filter	1x

Aktivkohle: 4mm Pellet

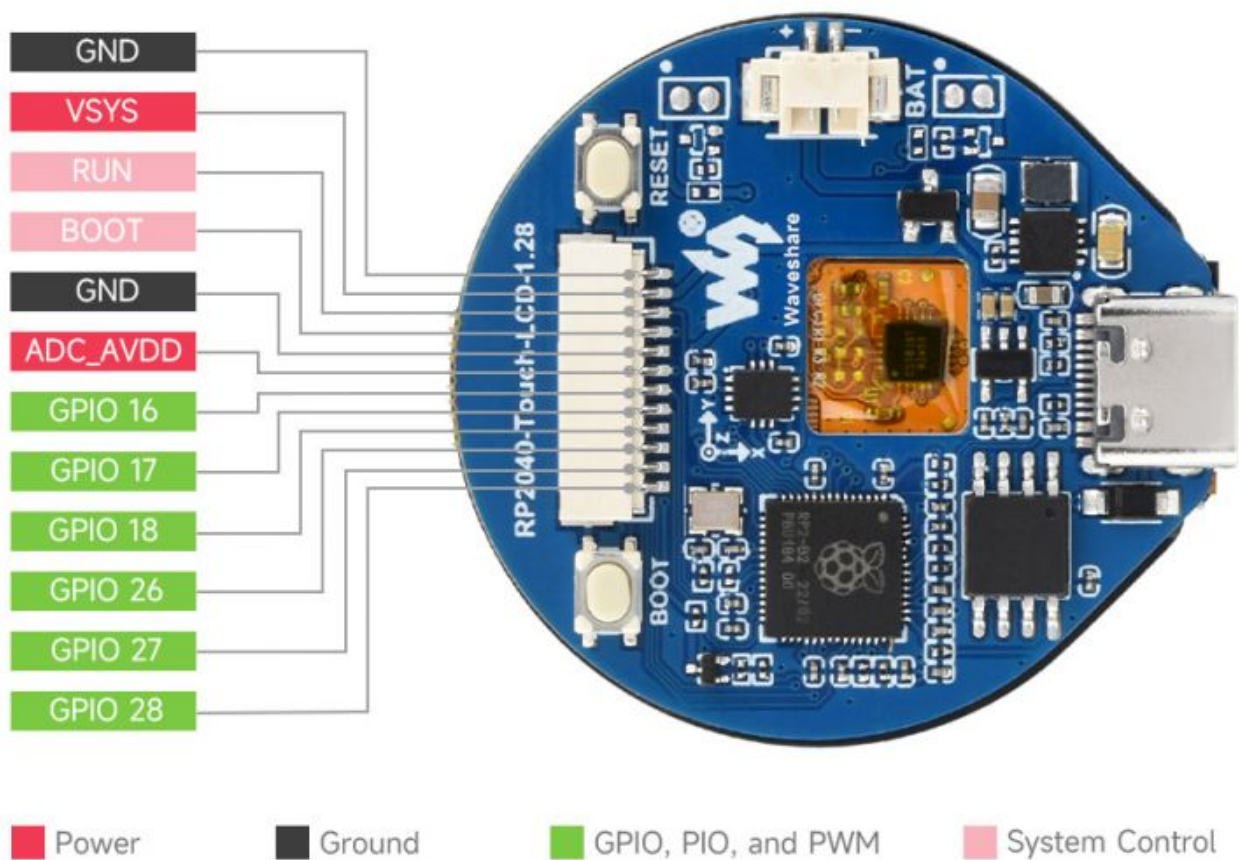
150g

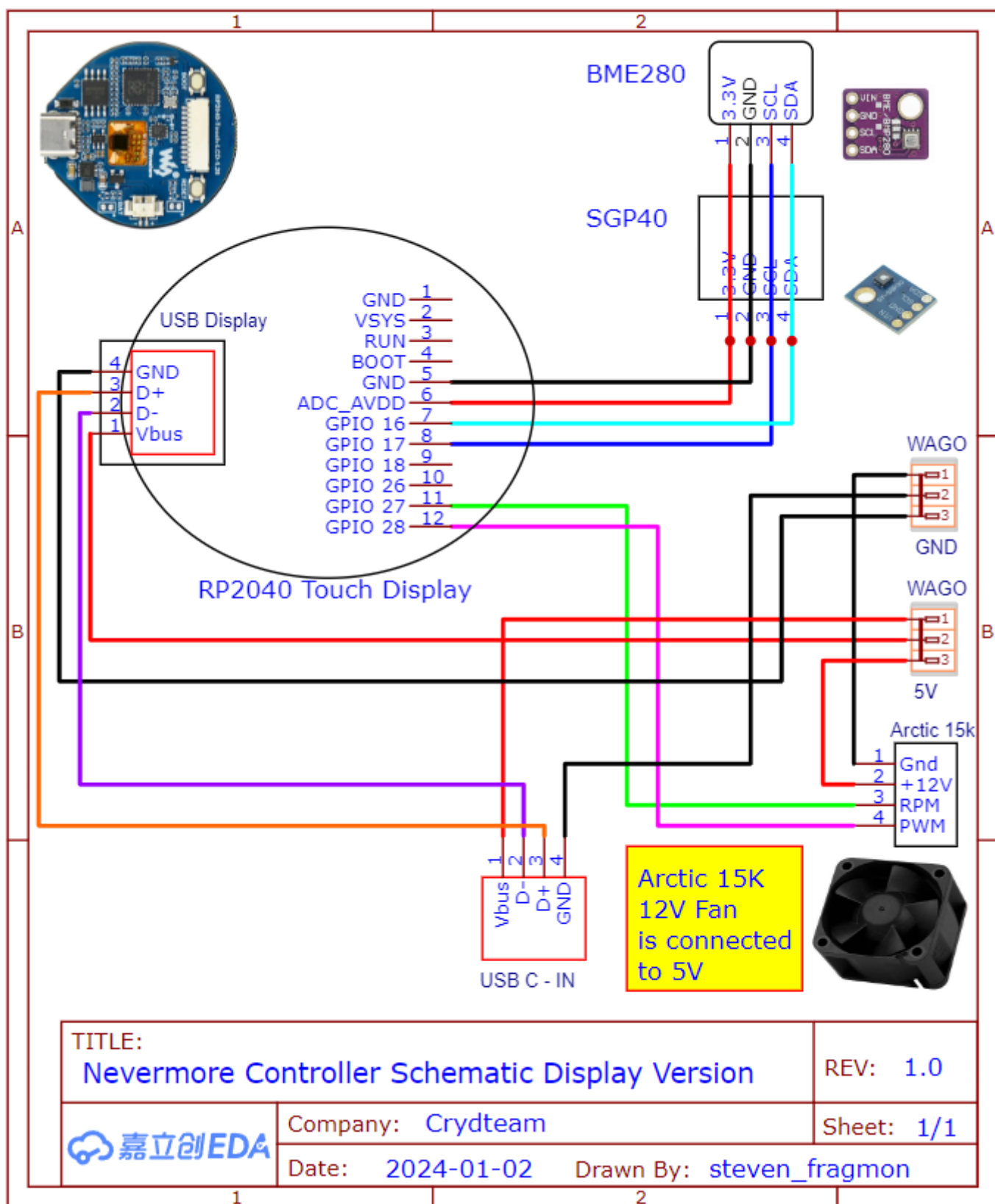
### Shoplink:

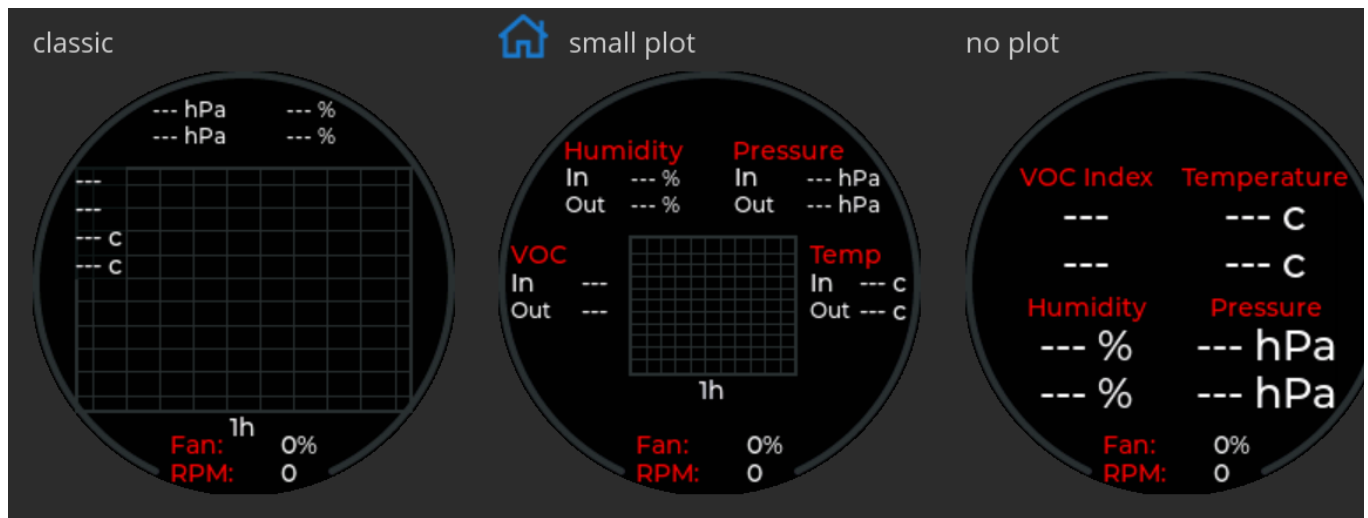
<https://cryd.de/unsere-empfehlungen/3d-druck/nevermore-mini/>

\*alle Links sind Affiliate Links womit du kostenfrei den Channel und den Discord unterstützt  
- jeder Cent kommt der Community zu Gute!

### Schaltplan







Small plot: [nevermore-controller-waveshare-1.28-v0.10-small-plot.uf2](#)

no plot: [nevermore-controller-waveshare-1.28-v0.10-no-plot.uf2](#)

100% Fan when power on: [nevermore-controller\\_waveshare-rp2040-touch-lcd-1.28\\_100%\\_fan.uf2](#)

Revision #32

Created 2 January 2024 12:25:26 by Steven

Updated 13 November 2024 14:56:55 by Cryd