Pi 5 Stromversorgung

Original 27W Netzteil (5.1V, 5A)

Das originale Netzteil verwendet USB-PD, um den Pi5 mit max. 5A zu versorgen. Das ist aber über der eigentlichen USB-PD Spezifikation - die geht bei 5V max. bis 3A. Danach sollte normalerweise die Spannung steigen auf 9V, um den Strom gering zu halten. Das kann der Pi5 aber nicht.

Es dürfte deshalb etwas schwierig werden, ein alternatives Netzteil für den Pi5 mit 5A zu finden. Und selbst wenn das Netzteil 5A liefern kann, wird es vom Pi5 vermutlich nicht erkannt. Denn bei USB-PD wird der maximale Strom (und im Normalfall auch die Spannung) zwischen Verbraucher (Pi5) und dem Netzteil ausgehandelt.

Mit dem Original Netzteil sollte sich der Pi 5 auch problemlos übertakten lassen (**ausreichende Kühlung vorausgesetzt!**). Wie das geht kann man bei Toms Hardware nachlesen: https://www.tomshardware.com/how-to/overclock-raspberry-pi-5

> **Hinweis**
> Übertaktet ist der Pi5 noch empfindlicher bei der Eingangsspannung. Bei 3GHz kam schon bei ~5V (PMIC ...) eine Warnmeldung und bei 4,95V hing sich der Pi auf.

Original 15W Netzteil (5.1V, 3A)

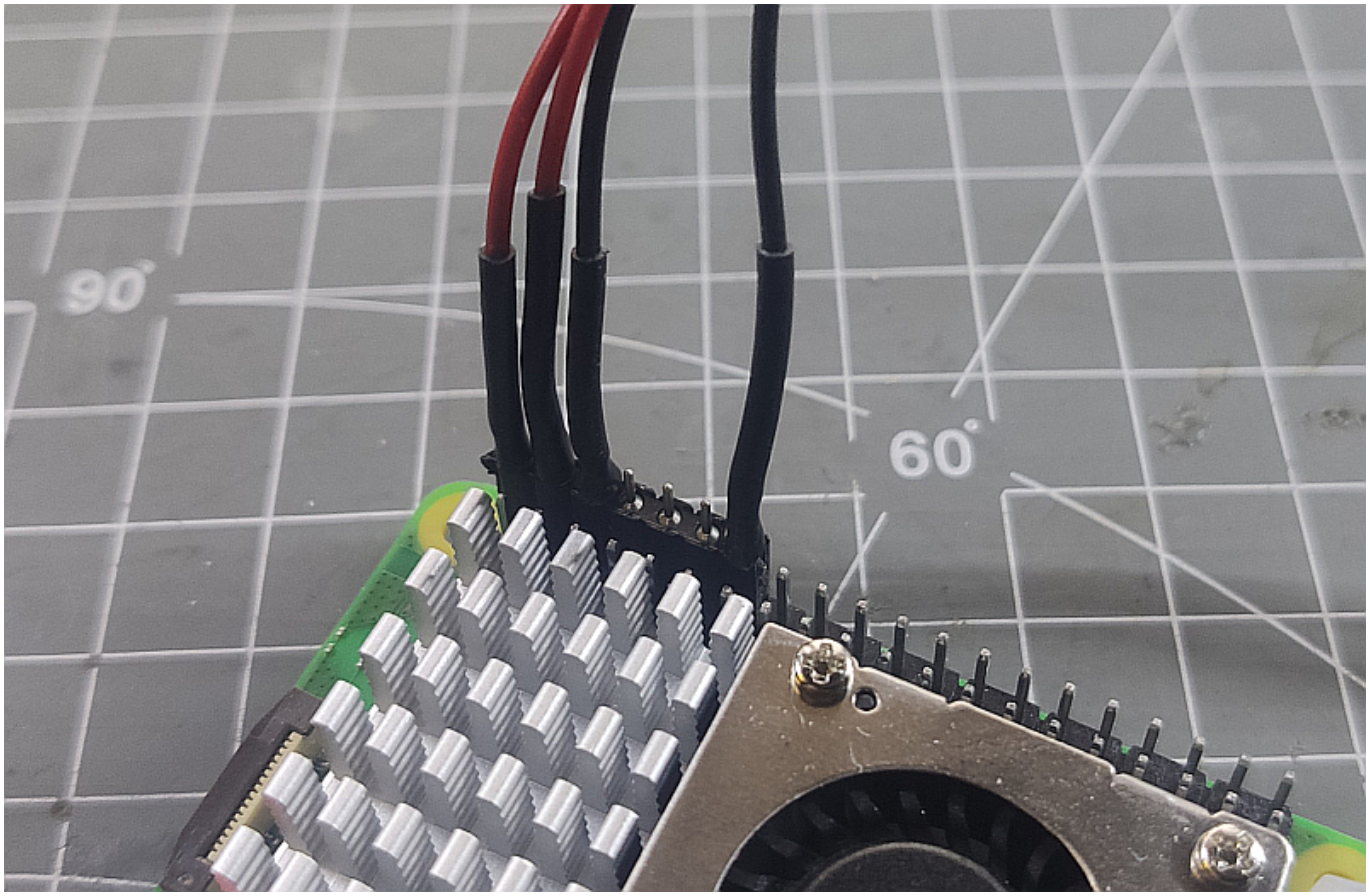
Das Netzteil (ursprünglich für den Pi4 gedacht) funktioniert auch mit einem Pi5. Da nur 3A zur Verfügung stehen, werden im Default aber nur 600mAh für die USB-Ports zur Verfügung gestellt. Diese Einschränkung kann man durch einen Eintrag in der `config.txt` umgehen - siehe hier.

Wird der Pi5 mit etwas anderem als einer SD-Karte gebootet, erscheint auch eine entsprechende Meldung.

> **Hinweis:**
> bei zu viel angeschlossener Peripherie und hoher CPU Last kann es zum Systemabsturz kommen. Entweder schaltet das Netzteil ab, oder die Spannung fällt zu stark ab.

Versorgung über Pins

Der Pi5 kann problemlos auch über die GPIO Stiftleiste versorgt werden.



Dabei sollten ein paar Dinge beachtet werden:

- **Nicht verpolungssicher und Kurzschlussgefahr!**
- Die Dupon Kontakte sind für hohe Ströme nicht ausgelegt. Deshalb 2x 5V und 2x GND belegen!
- Auch bei dieser Variante kommt der Hinweis, dass das Netzteil keine 5A liefern kann! Siehe hier
- Die Eingangsspannung überprüfen. Unterhalb von 4,85V kommt eine Low Voltage Warnung.
- Es sollte nicht unbedingt 0.14mm² Litze sein. Darf auch ruhig 0.17 oder 0.25 sein 🙂

"alte" 5V Netzteile

Jeder hat sicher noch ein Dutzend ältere 5V Ladegeräte herumliegen. Grundsätzlich kann man auch die verwenden, aber auch hier kommt die Meldung bezüglich dem erforderlichen 5A Netzteil. Es gibt zahlreiche Netzteils mit 1000mA - damit bootet der Pi oft erst gar nicht! Außerdem sollte man hier mal sein Augenmerk auf die Spannung legen. Wenn diese eher schwachen Netzteile stark belastet werden, kann die Spannung auch stark einbrechen. Zudem haben die Netzteile generell eher 5V am Ausgang und nicht 5,1V wie die neuen Raspberry Pi Netzteile!

"Fix" für ungeeignete Netzteile

- Beim Booten mittels USB-Stick, HDD, … erscheint folgende Meldung:

```
Raspberry Pi 5 - 4GB
 bootloader:  30de0ba5  2023/10/30


  board: c04170 5b2d0929 d8:3a:dd:c2:e0:f0
   boot: mode USB-MSD 4 order f41 retry 0/128 restart 0/-1
     SD: card not detected
   part: 0 mbr [0x0c:00002000 0x83:00102000 0x00:00000000 0x00:00000000]
  power: supply: Unknown 3000 mA PMIC: reset normal 0x0 usb_over_current=0
    net: down ip: 0.0.0.0 sn: 0.0.0.0 gw: 0.0.0.0
   tftp: 0.0.0.0 00:00:00:00:00:00
display: DISP0: HDMI HPD=1 EDID=ok #2 DISP1:  HPD=0 EDID=none #0

MSD INQUIRY [01:00] 3.00 000000:03
MSD [01:00] 3.00 000000:03 lun 0 block-count 60062500 block-size 512
Trying partition: 0
type: 32 lba: 8192 'mkfs.fat' ' bootfs     ' clusters 261116 (4)
Trying partition: 0
type: 32 lba: 8192 'mkfs.fat' ' bootfs     ' clusters 261116 (4)
Read config.txt bytes     1213 hnd 0x126
usb_max_current_enable default 0 max-current 3000
Read bcm2712-rpi-5-b.dtb bytes    75197 hnd 0xc9d6
***
USB boot requires high current (5 volt 5 amp) power supply.
To disable this check set usb_max_current_enable=1 in config.txt
or press the power button to temporarily enable usb_max_current_enable
and continue booting.
See https://rptl.io/rpi5-power-supply-info for more information
***
```

- Meldung kann mit On/Off Taste übersprungen werden (was automatisch usb_max_current_enable=1 setzt)
- USB Geräte werden auf 600mA limitiert (mit dem Original-Netzteil sind es 1.6A)
- Übergehen kann man das mit einem Eintrag in der `/boot/firmware/config.txt`
  - `sudo nano /boot/firmware/config.txt`
  - folgenden Eintrag ergänzen: `usb_max_current_enable=1`

Spannungen & Ströme

Die folgenden Werte sind experimentell ermittelt und dienen auch nur als grober Anhaltspunkt!

Spannungen

Verhalten	vcgencmd (EXT5V_V)	Netzteil
Warnschwelle Unterspannung	~4,85V	5,01V
Minimale Spannung	~4,5V	4.59V

Strom

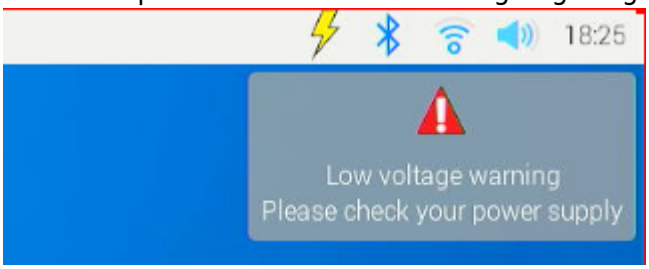
Modus	Bootmedium	Strom	Hinweis
Booten	SD	~800 mA	
	USB	~1000 mA	
Idle Desktop	SD	~600 mA	
	USB	~750 mA	

Modus	Bootmedium	Strom	Hinweis
Stress Test (4 Cores)	SD	~1250 mA	
	USB	~1600 mA	
	USB	~2000 mA	Übertaktet mit `arm_freq=3000` `gpu_freq=1000`

> **Hinweis**
> Stromverbrauch bei der Verwendung von USB Geräten kann je nach Gerät variieren!

Unterspannung erkennen

- Im Desktop Betrieb wird eine Meldung angezeigt:



- In der Konsole kann man sich die Meldungen über `dmesg -HW` ansehen.

```
pi@Pi5Test:~ $ dmesg -HW
[Dec29 18:27] hwmon hwmon3: Undervoltage detected!
[  +8.063959] hwmon hwmon3: Voltage normalised
```

Stresstest druchführen

- `sudo apt install stress`
- `stress -c 4 -m 4`

Check Voltage

Der Pi5 hat einen PMIC (Power Monitor IC) verbaut, den man mit dem Tool `vcgencmd` abfragen kann.

- Alle Daten abfragen
  `vcgencmd pmic_read_adc`

```
pi@Pi5Test:~ $ vcgencmd pmic_read_adc
 3V7_WL_SW_A current(0)=0.09076149A
   3V3_SYS_A current(1)=0.06538731A
   1V8_SYS_A current(2)=0.18249890A
  DDR_VDD2_A current(3)=0.02147046A
  DDR_VDDQ_A current(4)=0.00000000A
   1V1_SYS_A current(5)=0.24691030A
    0V8_SW_A current(6)=0.33474400A
```

```
  VDD_CORE_A current(7)=0.71440000A
   3V3_DAC_A current(17)=0.00000000A
   3V3_ADC_A current(18)=0.00030525A
   0V8_AON_A current(16)=0.00482295A
      HDMI_A current(22)=0.01452990A
 3V7_WL_SW_V volt(8)=3.70441600V
   3V3_SYS_V volt(9)=3.31838500V
   1V8_SYS_V volt(10)=1.79535800V
  DDR_VDD2_V volt(11)=1.10805800V
  DDR_VDDQ_V volt(12)=0.59890050V
   1V1_SYS_V volt(13)=1.10439400V
    0V8_SW_V volt(14)=0.80146440V
  VDD_CORE_V volt(15)=0.72017020V
   3V3_DAC_V volt(20)=3.30952000V
   3V3_ADC_V volt(21)=3.31043600V
   0V8_AON_V volt(19)=0.79794790V
      HDMI_V volt(23)=5.06922000V
     EXT5V_V volt(24)=5.07056000V
      BATT_V volt(25)=0.00000000V
```

- Nur die Eingangsspannung ermitteln
  `vcgencmd pmic_read_adc | grep EXT5V_V | cut -d"=" -f2`

```
pi@Pi5Test:~ $ vcgencmd pmic_read_adc | grep EXT5V_V | cut -d"=" -f2
5.06922000V
```

- Die Eingangsspannung alle 2 Sekunden auslesen (Abbruch mit STRG + C)
  `watch -n 2 'vcgencmd pmic_read_adc | grep EXT5V_V | cut -d"=" -f2'`

```
Every 2.0s: vcgencmd pmic_read_adc | grep EXT5V_V | cut -d"=" -f2
Pi5Test: Fri Dec 29 18:09:42 2023

5.07860000V
```

V Watch

- `sudo apt install python3-virtualenv python3-tk -y`
- `virtualenv grapher`
- `cd grapher/`
- `source bin/activate`
- `pip3 install matplotlib`
- `nano grapher.py`

```
import matplotlib
import matplotlib.pyplot as plt
from collections import deque
from datetime import datetime
import subprocess
import time
```

```
# Funktion zum Ausführen des Linux-Programms und Lesen der Ausgabe
def run_linux_program():
    command = '/usr/bin/vcgencmd measure_temp | cut -d"=" -f2 | cut -
d"\'" -f1'
    result = subprocess.run(command, capture_output=True, text=True,
shell=True)
    return float(result.stdout.strip())  # Hier anpassen, abhängig von
der Art der Ausgabe

# Initialisierung des Diagramms
matplotlib.use('TkAgg')
print(f"Interactive mode: {matplotlib.is_interactive()}")
print(f"matplotlib backend: {matplotlib.rcParams['backend']}")
plt.ion()  # Interaktiver Modus für kontinuierliche Aktualisierung
fig, ax = plt.subplots()
values = deque(maxlen=100)

# Datumsformat für die X-Achse
date_format = '%H:%M:%S:%f'

# Hauptschleife
print(run_linux_program())

plt.show()

try:
    while True:
        # Zeitstempel für die aktuelle Zeit
        current_time = datetime.now().strftime(date_format)

        # Wert vom Linux-Programm erhalten
        new_value = run_linux_program()

        # Werte in der Deque aktualisieren
        #values.append((current_time, new_value))
        values.append(new_value)

        # Diagramm aktualisieren
        ax.clear()
        ax.plot(values)
#       ax.plot(*zip(*values))  # Entpackt die Werte in zwei separate
Listen
        ax.set_title('Oszilloskop')
        ax.set_xlabel('Zeit')
        ax.set_ylabel('Wert')
#        ax.xaxis.set_major_formatter(plt.FuncFormatter(lambda x, _:
datetime.fromtimestamp(x).strftime(date_format)))
```

```
        # Nur 5 Ticks auf der X-Achse anzeigen
#        x_ticks = ax.get_xticks()
#        if x_ticks > 0:
#            x_tick_labels = ax.get_xticklabels()
#            ax.set_xticks(x_ticks[::len(x_ticks)/5])
#            ax.set_xticklabels(x_tick_labels[::len(x_ticks)/5])

#        plt.draw()
#        plt.pause(1.0)
        print("read")
        time.sleep(1.0)
except KeyboardInterrupt:
    pass
finally:
    plt.ioff()
    plt.show()
```

Der An / Aus Taster

- Einmal drücken für Einschalten
- bei der Netzteilwarnung kann man durch einmal drücken den Pi booten (was gleichzeitig usb_max_current_enable=1 setzt)
- Wenn der Pi läuft, kann man durch gedrückt halten (ca. 2-3 Sekunden) den Pi direkt ausschalten
- Wenn der Pi läuft, kann man durch 1x drücken den Pi herunter fahren. Im GUI Betrieb muss das durch ein zweites Drücken bestätigt werden.

Links

- https://www.raspberrypi.com/documentation/computers/raspberry-pi-5.html#powering-raspberry-pi-5
- https://www.raspberrypi.com/documentation/computers/raspberry-pi-5.html#usb-boot-and-power-supplies
- USB-PD - USB Power Delivery https://www.elektronik-kompendium.de/sites/com/1809251.htm
- https://vilros.com/pages/raspberry-pi-5-pinout

From:
https://drklipper.de/ - **Dr. Klipper Wiki**

Permanent link:
https://drklipper.de/doku.php?id=sbcs:raspberry_pi:pi_5_stromversorgung&rev=1703871083

Last update: **2023/12/29 18:31**