DigiLiner HARDWARE-ARCHITEKTUR & SYSTEMOPTIMIERUNGS-KARTE

Raspberry Pi 40-Pin GPIO Pinout & Linux-Spickzettel

40-Pin Header, vcgencmd, Watchdog, OverlayFS, ZRAM & PCIe Gen 3 Leitfaden (2026)

1. 40-Pin GPIO Farb-Pinout & Stromversorgungs-Map

40-Pin Header

3.3V	Pin 1 (3V3 PWR)	Pin 2 (5V0 PWR)	5V0
I2C	Pin 3 (GPIO 2 / SDA1)	Pin 4 (5V0 PWR)	5V0
I2C	Pin 5 (GPIO 3 / SCL1)	Pin 6 (GND Masse)	GND
GP	Pin 7 (GPIO 4 / GPCLK0)	Pin 8 (GPIO 14 / TXD0)	UART
GND	Pin 9 (GND Masse)	Pin 10 (GPIO 15 / RXD0)	UART
GP	Pin 11 (GPIO 17)	Pin 12 (GPIO 18 / PWM0)	PWM
GP	Pin 13 (GPIO 27)	Pin 14 (GND Masse)	GND
GP	Pin 15 (GPIO 22)	Pin 16 (GPIO 23)	GP
3.3V	Pin 17 (3V3 PWR)	Pin 18 (GPIO 24)	GP
SPI	Pin 19 (GPIO 10 / MOSI)	Pin 20 (GND Masse)	GND
SPI	Pin 21 (GPIO 9 / MISO)	Pin 22 (GPIO 25)	GP
SPI	Pin 23 (GPIO 11 / SCLK)	Pin 24 (GPIO 8 / CE0)	SPI
GND	Pin 25 (GND Masse)	Pin 26 (GPIO 7 / CE1)	SPI
ID	Pin 27 (ID_SD / EEPROM)	Pin 28 (ID_SC / EEPROM)	ID
GP	Pin 29 (GPIO 5)	Pin 30 (GND Masse)	GND
GP	Pin 31 (GPIO 6)	Pin 32 (GPIO 12 / PWM0)	PWM
PWM	Pin 33 (GPIO 13 / PWM1)	Pin 34 (GND Masse)	GND
PCM	Pin 35 (GPIO 19 / PCM_FS)	Pin 36 (GPIO 16)	GP
GP	Pin 37 (GPIO 26)	Pin 38 (GPIO 20 / PCM_DIN)	PCM
GND	Pin 39 (GND Masse)	Pin 40 (GPIO 21 / PCM_DOUT)	PCM

WICHTIGE WARNUNG: GPIO-Pins arbeiten mit **3,3V LVTTL**! 5V-Signale zerstören den Broadcom-SoC dauerhaft. Pegelwandler verwenden.

2. Wichtige vcgencmd Diagnose-Befehle

Diagnostic

Befehl	Beschreibung	Erwarteter Wert / Ausgabe
`vcgencmd measure_temp`	Kerntemperatur	`temp=48.2'C` (Max 80°C)
`vcgencmd measure_volts`	SoC-Kernspannung	`volt=0.8500V` (<4.63V)
`vcgencmd measure_clock arm`	Aktiver CPU-Takt	`frequency(48)=2400000000`
`vcgencmd get_mem arm/gpu`	RAM-Aufteilung	`arm=3968M gpu=128M`
`vcgencmd get_throttled`	Spannung & Temperatur	`throttled=0x0` (Einwandfrei)

```
# get_throttled Bitmasken-Analyse:
0x1: Aktive Unterspannung (<4.63V!) | 0x2: Aktive CPU-Drosselung
0x4: Aktives thermisches Throttling (>80°C) | 0x50000: Unterspannung in der Vergangenheit
```

3. OverlayFS Schreibgeschütztes Root-Dateisystem

Zero Corruption

```
# 1. Overlay-Root-Dateisystem via raspi-config aktivieren:
sudo raspi-config
# -> Performance Options -> Overlay File System -> Enable -> Yes
# -> Boot partition read-only -> Yes -> Reboot
mount | grep "overlay on /" # (Keine SD-Karten-Korruption bei Stromausfall!)
```

4. ZRAM In-Memory-Komprimierung (Kein Disk-Swap)

RAM Boost

```
# 1. SD-Karten-schädigenden dphys-swapfile deaktivieren:
sudo dphys-swapfile swapoff && sudo systemctl disable dphys-swapfile
# 2. zram-tools installieren und /etc/default/zramswap bearbeiten:
sudo apt update && sudo apt install -y zram-tools
PERCENTAGE=60 # 60% des RAMs als komprimierter Swap
ALGO=lz4 # Schneller lz4-Algorithmus (Kein CPU-Engpass)
sudo systemctl restart zramswap && zramctl
```

5. Hardware-Watchdog-Timer (Absturzsicheres System)

Broadcom Hardware

```
# 1. In /boot/firmware/config.txt eintragen:
dtparam=watchdog=on
# 2. In /etc/systemd/system.conf aktivieren:
RuntimeWatchdogSec=14s   RebootWatchdogSec=10min
# 3. Test: Hardware erzwingt nach 15s Kaltstart bei Kernel-Panik!
echo 1 | sudo tee /proc/sys/kernel/sysrq && echo c | sudo tee /proc/sysrq-trigger
```

6. USB OTG Gadget-Modus (Ethernet über USB)

Headless

```
# Pi Zero / Zero 2W / Pi 4 (Headless-SSH über ein einzelnes USB-Kabel):
1. boot/config.txt: dtoverlay=dwc2
2. boot/cmdline.txt: modules-load=dwc2,g_ether
3. Leere 'ssh'- und 'userconf.txt'-Datei in boot anlegen.
ssh pi@raspberrypi.local # (Direkter Terminal-Zugriff via USB!)
```

7. Raspberry Pi 5 PCIe Gen 3 & NVMe SSD Boot

870+ MB/s

```
# 1. In /boot/firmware/config.txt eintragen:
dtparam=pciex1
dtparam=pciex1_gen=3 # PCIe Gen 3.0 erzwingen (870+ MB/s)
# 2. EEPROM-Boot-Reihenfolge anpassen: (6=NVMe, 1=SD-Karte)
sudo rpi-eeprom-config --edit # BOOT_ORDER=0xf416
```

8. GPIO Elektrische Sicherheit & Grenzwerte

Electrical Limits

Parameter	Sicherer Grenzwert	Beschreibung & Vorsichtsmaßnahme
Eingangsspannung	`4.85V - 5.25V`	Unter 4.63V droht Taktkürzung oder Reset.
Einzelpin-Strom	`16 mA (Max)`	Empfohlener Betriebsstrom liegt bei 3 - 8 mA.
Gesamt-GPIO-Budget	`50 mA (Total)`	Relais und Motoren über externe Transistoren ansteuern.
Interne Widerstände	`~50 kΩ`	Softwareseitiger Pull-up / Pull-down-Wert.