

RealTime (RT) Kernel

basierend auf : https://www.raspberrypi.com/documentation/computers/linux_kernel.html#building

Compilieren

- `sudo apt install git bc bison flex libssl-dev make libncurses5-dev #I think this is all the tools required`
- `mkdir kernel && cd kernel/`
- `git clone --depth=1 --branch rpi-6.12.y`
<https://github.com/raspberrypi/linux>
- `wget`
<https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/6.12/patch-6.12.28-rt10.patch.gz>
<https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/6.12/>
- `cd linux/`
- `zcat ../patch-6.12.28-rt10.patch.gz | patch -p1 --dry-run #check the patch fits`
- `zcat ../patch-6.12.28-rt10.patch.gz | patch -p1`
- `KERNEL=kernel8`
- `make bcm2711_defconfig`
- `make menuconfig`

- General setup --->

- Preemption Model (Preemptible Kernel (Low-Latency Desktop)) --->
 - [*] Fully Preemptible Kernel (Real-Time)
 - [] Preemption behaviour defined on boot

- Beim Exit Config Save → Yes

- `nano .config`
`CONFIG_LOCALVERSION="-DrKlipper RT"`
- `time make -j4 Image.gz modules dtbs`

```
echo $KERNEL sudo make modules_install sudo cp arch/arm64/boot/dts/broadcom/*.dtb /boot/ sudo cp arch/arm64/boot/dts/overlays/*.dtb* /boot/overlays/ sudo cp arch/arm64/boot/dts/overlays/README /boot/overlays/ sudo cp arch/arm64/boot/Image.gz /boot/$KERNEL.img
```

Links

- https://www.raspberrypi.com/documentation/computers/linux_kernel.html
- Ursprung: <https://forums.raspberrypi.com/viewtopic.php?t=344994>
- <https://blog.emlid.com/raspberry-pi-real-time-kernel/>
- ISO erstellen mit RT Kernel
<https://github.com/remusmp/rpi-rt-kernel>

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

Permanent link:
https://drklipper.de/doku.php?id=projekte:sekwai:rt_kernel&rev=1753003142

Last update: **2025/07/20 11:19**

