

# Inhaltsverzeichnis

<b>Eigene Kernel-Module für den Raspi kompilieren</b> .....	3
<i>Kurzform</i> .....	4



# Eigene Kernel-Module für den Raspi kompilieren

Wer mal so richtig in eine verschimmelt saure Zitrone beißt und sich eigene Kernel-Module für den Raspi bauen muss, weil z.B. der vxcan- Treiber nicht im Standard-Kernel enthalten ist, hier die unter Schmerzen entstandene Anleitung:

erstmal das alte python 2 installieren

```
sudo apt install python2
```

Dann den Anweisungen auf <https://github.com/RPi-Distro/rpi-source> folgen

```
sudo apt install git bc bison flex libssl-dev
sudo wget
https://raw.githubusercontent.com/RPi-Distro/rpi-source/master/rpi-source -O
/usr/local/bin/rpi-source && sudo chmod +x /usr/local/bin/rpi-source &&
/usr/local/bin/rpi-source -q --tag-update
rpi-source
```

dann noch fix das ncurses- Paket

```
sudo apt install libncurses5-dev
```

Danach ist im aktuellen Verzeichnis ein Verzeichnis linux mit all den Sourcen.

Da kann man dann reinwechseln und sich mit

```
sudo modprobe configs
zcat /proc/config.gz > .config
```

die aktuelle Config ziehen, händisch editieren

und mit

```
make oldconfig
make
```

beginnt stundenlanges Warten...

Am nächsten Morgen findet sich dann das ersehnte Kernel- Modul in ~/linux/drivers/net/can und kann dann mit einem beherzten

```
sudo insmod vxcan.ko
```

eingebunden werden

## Kurzform

Scheinbar gibt's zum Compilieren doch noch die Kurzform, ohne den ganzen Kernel machen zu müssen: Die Konfig `.config` wie oben beschrieben erzeugen und anpassen, und dann

```
cd ~/linux/drivers/net/can/  
make -C /lib/modules/$(uname -r)/build M=$(pwd) modules
```

From:

<http://koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://koehlers.de/wiki/doku.php?id=pc:raspbikernelbuild>

Last update: **2022/08/21 04:51**

