

# USB Device an einen Container durchreichen

## Informationen sammeln

```
(root@n1-pve-cluster marko) > lsusb
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 0403:6001 Future Technology Devices International,
Ltd FT232 Serial (UART) IC
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Mich interessiert

```
Future Technology Devices International, Ltd FT232 Serial (UART) IC
```

Daher nehme ich als Major Nummer 003 und als Minor Nummer die 002

Nun schauen wir uns den dazugehörigen Verzeichnisbaum an

```
(root@n1-pve-cluster marko) > ls -l /dev/bus/usb/003/002
crw-rw-r-- 1 root root 189, 257 Feb 24 09:14 /dev/bus/usb/003/002
```

Für mich relevant sind die zwei Zahlen 189 und 257. Diese benötigen wir für unsere Container Konfiguration.

## Konfiguration des Containers anpassen

Nun bearbeiten wir das Container Configfile. Dafür liste ich mir alle Container von der Node auf und schaue mir die Container ID meines Containers an welcher das USB Device bekommen soll. (p-fhem02)

```
(root@n1-pve-cluster marko) > pct list
VMID      Status      Lock      Name
201       running      
202       stopped      
203       running      
208       running      
213       running      
p-fhem02
p-printserver
p-psql01
p-gitserver
p-nextcloud
```

Das passende Konfigfile editiere ich nun

```
vim /etc/pve/lxc/201.conf
```

Und passen die entsprechenden 2 Zahlen an

```
lxc.cgroup.devices.allow: c 189:257 rwm
lxc.mount.entry: /dev/SIGNALduino dev/SIGNALduino none
bind,optional,create=file
```

## Dank udev Rule neues Device mit korrekter Gruppenzuordnung

Und nun zum wesentlichen. Wie man sehen kann habe ich in meiner Konfig /dev/SIGNALduino durchgereicht. Dies musste ich machen damit die tatsächliche benötigte Gruppe im Container auch die Rechte bekommt.

Damit mir ein solches Device auch erstellt wird **/dev/SIGNALduino** muss eine entsprechende udev Rule auf dem Host angelegt werden.

Dafür benötigen wir etwas Vorarbeit. Wir müssen unser Device besser Identifizieren.

```
(root@n1-pve-cluster marko) > udevadm info -a -n /dev/ttyUSB0 | grep
'{idProduct}' | head -n1
```

```
(root@n1-pve-cluster marko) > udevadm info -a -n /dev/ttyUSB0 | grep
'{idVendor}' | head -n1
```

Mit diesen Infos erstellen wir unsere Rule

```
(root@n1-pve-cluster marko) > vim /etc/udev/rules.d/98-RFXtrx-CUL.rules
```

```
SUBSYSTEMS=="usb", KERNEL=="ttyUSB*", ATTRS{idProduct}=="6001",
ATTRS{idVendor}=="0403", SYMLINK+="RFXCOM_RFXtrx433_A1XEABQ0", MODE="0666",
GROUP="dialout"
```

Zum Abschluss müssen die udev Rules neu eingelesen werden

```
(root@n1-pve-cluster marko) > udevadm control --reload-rules && udevadm
trigger
```

Und zu guter letzt den Container neu starten

```
(root@n1-pve-cluster marko) > pct stop <cid>
(root@n1-pve-cluster marko) > pct start <cid>
```

From:  
<https://www.cooltux.net/> - TuxNet DokuWiki

Permanent link:  
[https://www.cooltux.net/doku.php?id=it-wiki:proxmox:usb\\_passthrough](https://www.cooltux.net/doku.php?id=it-wiki:proxmox:usb_passthrough)

Last update: **2020/03/05 08:19**

