

## Inhalt

1	MCP2515 CAN-Controller: .....	1
1.1	neues Image auf SD-Karte kopieren.....	1
1.2	SPI Interface aktivieren: .....	1
1.3	Update durchführen:.....	1
1.4	Overlays hinzufügen (MCP2515 -> Quarzfrequenz, Interrupt-Pin):.....	2
1.5	CAN als Netzwerk einbinden (bei jedem Neustart notwendig): .....	2
1.6	Testprogramme für CAN installieren: .....	2
1.7	can-utils: .....	3
1.7.1	candump:.....	3
1.7.2	cansend: .....	3
1.7.3	canplayer, usw.:.....	3
1.8	Nützliche Tipps für Fehlersuche: .....	3
1.8.1	Kernelmeldungen:.....	3
1.8.2	Interrupts: .....	3
1.8.3	Reservierte GPIO's anzeigen: .....	3
1.8.4	Status des CAN0: .....	3

# 1 MCP2515 CAN-Controller:

Der Treiber des MCP2515 CAN-Controller muss direkt ins Betriebssystem des Raspberry integriert werden, dieser Vorgang wird hier anhand des **Raspberry B+** mit dem Raspbian **Kernel 3.18** beschrieben.

Englischsprachige Anleitung aus der diese Anleitung abgeleitet wurde:

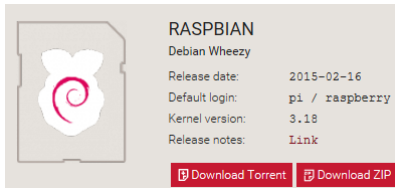
<http://skpang.co.uk/blog/archives/1165>

Die **Kernelversion 3.18.7+** hat selbst noch keinen MCP2515 Treiber integriert, kann aber ganz einfach mit einem Firmwareupdate (auf die Version 3.18.11+) um diesen Treiber erweitert werden.

In zukünftigen Kernelversionen wird dieser Treiber dann standardmäßig verfügbar sein.

## 1.1 neues Image auf SD-Karte kopieren

- Es wird ein neues Image von Raspbian benötigt  
Download über [www.raspberrypi.org](http://www.raspberrypi.org) (Kernel 3.18, Release 16.02.2015)



- Image auf Micro-SD Karte speichern und in Raspberry einlegen
- Raspberry starten und Grundeinstellungen vornehmen (Sprache, Tastatur, ...)

## 1.2 SPI Interface aktivieren:

- LXTerminal öffnen und raspi-config starten

```
cd /usr/bin
sudo ./raspi-config
```

- Unter **Advanced Options/A6 SPI** das...
  - **SPI-Interface Aktivieren** und den
  - **Autostart aktivieren**
- raspi-config mit **Finish** verlassen (Neustart JA)

## 1.3 Update durchführen:

- LXTerminal öffnen

```
sudo apt-get update
sudo apt-get upgrade
sudo reboot
```

```
sudo rpi-update
```

## 1.4 Overlays hinzufügen (MCP2515 -> Quarzfrequenz, Interrupt-Pin):

- LXTerminal öffnen

```
sudo nano /boot/config.txt
```

- Folgende Zeilen am Ende des config.txt hinzufügen

```
dtparam=spi=on
dtoverlay=mcp2515-can0-overlay, oscillator=16000000,interrupt=25
dtoverlay=spi-bcm2835-overlay
```

- mit **STRG+O** speichern, mit **ENTER** bestätigen

- nano mit **STRG+X** beenden

```
sudo reboot
```

- \*Ab kann der MCP2515 an den Raspberry angeschlossen werden. (Schaltplan beachten!)

Im Beispiel wird ein **16Mhz Quarz** verwendet und der Interrupt-Pin ist **GPIO25**

**Änderungen an der Raspberry Verdrahtung nur im ausgeschaltetem Zustand vornehmen, um Kurzschlüsse und somit eine Beschädigung des Raspberry zu vermeiden!**

Schaltpläne hierzu sind genug im Internet zu finden.

Grundsätzlich gilt:

Raspberry	MCP2515	MCP2551
3,3V	Pin 18 (VDD)	
0V	Pin 9 (VSS)	
GPIO08 (CE0)	Pin 16 (CS)	
GPIO09 (MISO)	Pin 15 (SO)	
GPIO10(MOSI)	Pin 14 (SI)	
GPIO11 (SCKL)	Pin 13 (SCK)	
GPIO25	Pin 12 (INT)	
	Pin 1(TXCAN)	Pin 1 (TXD)
	Pin 2(RXCAN)	Pin 4 (RXD) (Achtung 5V) Spannungsteiler verwenden!
		Pin 2 (VSS)
		Pin 3 (VDD) (Achtung 5V)

## 1.5 CAN als Netzwerk einbinden (bei jedem Neustart notwendig):

- LXTerminal öffnen (Bitrate definieren, im Beispiel 125kbit/s)

```
sudo /sbin/ip link set can0 up type can bitrate 125000
```

\*Wird hier eine Fehlermeldung angezeigt, so wurde der MCP2515 beim Starten nicht richtig initialisiert... Verbindungen prüfen und nochmals neustarten.

- Beim Ausführen von ...

```
ifconfig
```

... sollte jetzt can0 als Netzwerkmodul vorhanden sein.

## 1.6 Testprogramme für CAN installieren:

- LXTerminal öffnen (can-utils installieren)

```
git clone https://github.com/linux-can/can-utils.git
cd can-utils
make
sudo make install
```

## 1.7 can-utils:

### 1.7.1 candump:

Zeichnet alle Daten die über den CAN-Bus gesendet/empfangen werden auf und stellt sie im Terminal dar. Abbruch der Aufzeichnung über **STRG+C** möglich.

```
candump can0

Bsp.:
can0  001  [3]  01  02  10
can0  400  [4]  01  02  FF  FF
can0  002  [1]  01
can0  123  [2]  01  02
...

candump -tA -xe can0,0:0,#FFFFFFFF

Bsp.:
{date} can0 RX - - 123 [2] 80 00
```

### 1.7.2 cansend:

Sendet den Identifier und ein paar Datenbyte

```
cansend can0 {identifier}#{data}

Bsp.:
cansend can0 123#0102030405
cansend can0 123#01.02.03.04.05
```

### 1.7.3 canplayer, usw.:

can-utils bietet einige nützliche Tools, weitere Infos siehe [WWW](#)

## 1.8 Nützliche Tipps für Fehlersuche:

### 1.8.1 Kernelmeldungen:

Gibt die Kernelmeldungen des Ringpuffers aus

```
dmesg | egrep -I "can|mcp251|spi"
```

### 1.8.2 Interrupts:

Hier kann kontrolliert werden, welcher Interrupt für den MCP konfiguriert ist

```
cat /proc/interrupts
```

### 1.8.3 Reservierte GPIO's anzeigen:

Zeigt bereits reservierte GPIO's an

```
sudo mount -t debugfs none /sys/kernel/debug
sudo cat /sys/kernel/debug/gpio
```

### 1.8.4 Status des CAN0:

Zeigt gesetzte Bitrate, CAN Status, Anzahl gesendeter und empfangener Bytes/Pakete

```
ip -s -d link show can0
ifconfig can0 (zeigt ähnliche Infos wie vorheriger Befehl)
```