

I2C – Kampis Elektroecke



Hier zeige ich, wie der I2C Bus des Raspberry Pi genutzt werden kann. Der Raspberry Pi nutzt 3,3 V Pegel und deswegen dürften NIEMALS 5 V Signalen auf das Board gegeben werden, da es ansonsten zu Schäden am Prozessor kommen kann.

I2C aktivieren:

Als aller erstes muss der I2C vom Raspberry Pi aktiviert werden. Dies geschieht durch folgende Zeilen:

```
$ sudo modprobe i2c-bcm2708 baudrate="WERT"
$ sudo modprobe i2c-dev
$ sudo apt-get install i2c-tools
```

Die Zeilen

```
$ sudo modprobe i2c-bcm2708 baudrate="WERT"
$ sudo modprobe i2c-dev
```

laden die für den I2C notwendigen Module und stellen den Takt vom I2C auf den Wert

Wert

wert ein (Default ist 100 kHz).

Die durchgeführten Befehle müssen bei jedem Neustart neu ausgeführt werden und deswegen ist es empfehlenswert diese Befehle in die Datei

/etc/rc.local

/etc/rc.local zu schreiben. Mit der Zeile

```
$ sudo apt-get install i2c-tools
```

werden die notwendigen Tools installiert um auf den I2C zugreifen zu können. Anschließend kann mittels

```
$ sudo i2cdetect -y 0
```

geprüft werden ob der I2C funktioniert.

```
192.168.178.25 - PuTTY
login as: root
root@192.168.178.25's password:
Linux raspberrypi 3.2.27+ #160 PREEMPT Mon Sep 17 23:18:42 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  4 13:28:53 2012
root@raspberrypi:~# $ sudo i2cdetect -y 0
-bash: $: command not found
root@raspberrypi:~# sudo i2cdetect -y 0
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20: 20  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:~# █
```

Dadurch zeigt euch das Raspberry Pi die Adressverteilung der Devices am Bus an. Bei mir ist z. B. ein Device mit der Adresse **0x20** angeschlossen (ein PCF8574).

I2C per Konsole verwenden:

Wenn ein Device am Bus erkannt wurde, kann mit den beiden Befehlen

```
$ sudo i2cget
$ sudo i2cset
```

per I2C Bus auf das Device zugreifen werden. Wenn von dem Device gelesen werden soll, muss der Befehl

```
i2cget
```

`i2cget` verwenden und falls geschrieben werden soll

```
i2cset
```

`i2cset` verwendet werden. Ein vollständiger Befehl zum Lesen sieht z. B. folgendermaßen aus:

```
$ sudo i2cget -y 0 0x20
```

Mittels

```
i2cget -y
```

`i2cget -y` soll von einem I2C Device gelesen werden, wobei die **0** der I2C Bus ist, der für den Zugriff verwendet werden soll (das Raspberry Pi hat auch noch einen anderen I2C Bus, aber der Bus der auf den 26-pol Connector rausgeführt ist ist der Bus **0**). Der Wert **0x20** ist

die Adresse von dem I2C-Device auf das Zugegriffen werden soll. Als Rückgabewert erhält man dann auf der Konsole das gesendete Byte vom I2C-Device.

Wenn nun in das Device geschrieben werden soll, so sieht der Befehl folgendermaßen aus:

```
$ sudo i2cset -y 0 0x20 0x00
```

Mit

```
i2cset -y
```

`i2cset -y` wird ein Schreibzyklus eingeleitet, wobei die **0** ist wieder der zu verwendende Bus ist und die **0x20** ist die Deviceadresse, gefolgt von einer **0x00**, welche das zu sendende Datenbyte darstellt.

Auslesen eines EEPROMs:

Dieses Vorgehen soll beispielhaft an einem EEPROM (24C128) gezeigt werden:

```
$ i2cset -y 0x50 0x00 0x00 0x10 i  
$ i2cset -y 0x50 0x00 0x00  
$ i2cget -y 0x50
```

Die erste Zeile spricht das EEPROM mit einem Schreibbefehl an und überträgt alle Adressen und Daten, die daraufhin in das EEPROM geschrieben werden. Dabei stellt der Wert **0x50** ist die Adresse des EEPROMs dar, die zwei **0x00** sind die beiden Datenbytes für den Adresspointer im EEPROM und die **0x10** sind die Daten die in das EEPROM geschrieben werden sollen.

Der Parameter **i** gibt an, dass es sich bei dem gesendeten Werten um Integer handelt, also die Adresse und die Daten aus je zwei Bytes bestehen.. Es muss bei Übertragungen von 4 Bytes oder mehr hinzugefügt werden. Da der Pointer im EEPROM nach jedem Schreibbefehl um eins inkrementiert wird, muss dieser jetzt erst wieder auf **0x00** gesetzt werden, bevor die Zelle gelesen werden kann.

```
$ i2cset -y 0x50 0x00 0x00
```

Auch hier ist die **0x50** wieder die Deviceadresse vom EEPROM und die zwei **0x00** sind die Adresse für den Pointer.

Anschließend kann die Zelle mit dem Befehl

```
$ i2cget -y 0x50
```

ausgelesen werden. Die Konsole gibt nun den Wert **0x10** aus.

Nachfolgend noch ein paar Beispielanwendungen mit dem I2C.

- [Eine DS1307 als Uhr verwenden](#)
- [Externer ADC & CapSense](#)
- [Servo ansteuern mittels PCA9685](#)

Sämtliche Codebeispiele stehen in meinem [Raspberry Pi Repository](#) zum Download bereit.

