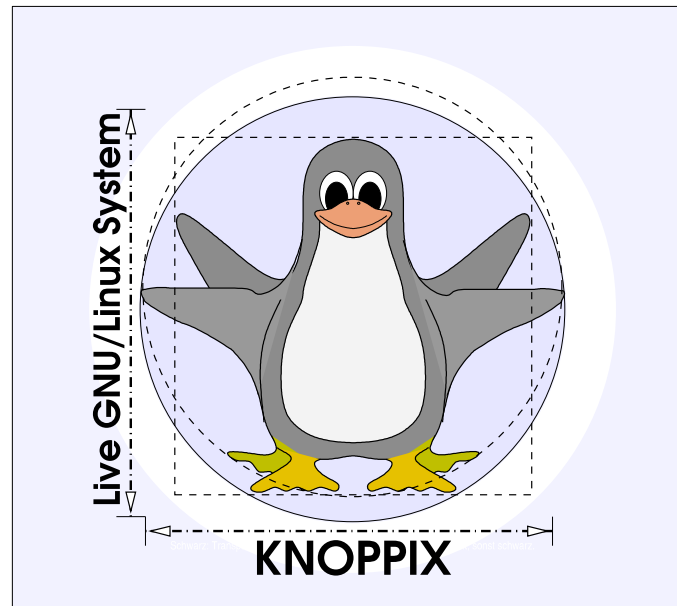


GPIO

Schnittstellen auf dem Raspberry Pi kontrollieren



Prof. Dipl.-Ing. Klaus Knopper
<knoppix@knopper.net>



Pins

Raspberry Pi besitzt 16 (relativ) frei benutzbare ¹ General Purpose Input/Output^a Pins, die sich mit Hilfe eines Kernel-Busmoduls ansteuern lassen, welches die Pins als Dateien in `/sys` abbildet.

Beispiel: LEDs ansteuern (per Shell und Webformular)

http://kampus-elektroecke.de/?page_id=1659

GPIO Pin 0 aktivieren:

```
echo 0 | sudo tee /sys/class/gpio/export
```

GPIO Pin 0 auf ¹ Ausgabe^a schalten:

```
echo "out" | sudo tee /sys/class/gpio/gpio0/direction
```

GPIO Pin 0 ¹ ein^a:

```
echo "1" | sudo tee /sys/class/gpio/gpio0/value
```



Serielle Schnittstelle vs. GPIO

Achtung: Einige Pins sind bereits durch die serielle Schnittstelle belegt, ggf. folgende Zeile in /etc/inittab auskommentieren und `sudo init q`:

```
# Spawn a getty on Raspberry Pi serial line  
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```



Pins

Raspberry Pi besitzt 16 (relativ) frei benutzbare ¹ General Purpose Input/Output^a Pins, die sich mit Hilfe eines Kernel-Busmoduls ansteuern lassen, welches die Pins als Dateien in `/sys` abbildet.

Beispiel: LEDs ansteuern (per Shell und Webformular)

http://kampus-elektroecke.de/?page_id=1659

GPIO Pin 0 aktivieren:

```
echo 0 | sudo tee /sys/class/gpio/export
```

GPIO Pin 0 auf ¹ Ausgabe^a schalten:

```
echo "out" | sudo tee /sys/class/gpio/gpio0/direction
```

GPIO Pin 0 ¹ ein^a:

```
echo "1" | sudo tee /sys/class/gpio/gpio0/value
```



Funktion	Pin		Pin	Funktion
3,3V	1		2	5V
GPI0 2 (SDA)	3		4	
GPI0 3 (SCL)	5		6	GND
GPI0 4	7		8	GPI0 14 (TXD)
GND	9		10	GPI0 15 (RXD)
GPI0 17	11		12	GPI0 18 (PCMCLK)
GPI0 27	13		14	GND
GPI0 22	15		16	GPI0 23
3,3V	17		18	GPI0 24
GPI0 10 (MOSI)	19		20	GND
GPI0 9 (MISO)	21		22	GPI0 25
GPI0 11 (SCLK)	23		24	GPI0 8
GND	25		26	GPI0 7



Serielle Schnittstelle vs. GPIO

Achtung: Einige Pins (z.B. TxD, RxD) sind bereits durch die serielle Schnittstelle belegt, ggf. folgende Zeile in /etc/inittab auskommentieren und

```
^sudo init q^:
```

```
# Spawn a getty on Raspberry Pi serial line  
# T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```



Pegelwandler

Die Signalausgabe erfolgt auf dem RasPi mit einem 3.3V „high“ Pegel, was sich nicht mit den auf vielen seriellen Schnittstellen liegenden 5V Pegel verträgt: Pegelwandler erforderlich!



GPIO skripten

PHP: s. <http://10.0.0.100/Skripte/Raspi/index.php>s

Bash: s.

<http://10.0.0.100/Skripte/Raspi/blinkenlights.sh>

Skript für GPIO Pin 4:

Erst root werden: **sudo -i**

```
echo 4 > /sys/class/gpio/export
```

```
echo out > /sys/class/gpio/gpio4/direction
```

Nun kann die LED ein- und ausgeschaltet werden:

```
echo 1 > /sys/class/gpio/gpio4/value
```

```
echo 0 > /sys/class/gpio/gpio4/value
```



Eingaben per GPIO

Im 1 in 2 -Modus können die GPIO-Pins Signale an das System übertragen, der aktuelle Wert (0, 1) wird in `/sys/class/gpio/gpioNUMMER/value` dargestellt und kann z.B. mit einem Skript ausgelesen und verarbeitet werden.

S.a.

<http://10.0.0.100/Skripte/Raspi/blinkenswitch.sh>



Dateirechte für GPIO zur Benutzung durch den Web- Server setzen

Nach Freigabe der Pins (s. Folie 2) sind die GPIO-Verzeichnisse zunächst nur für die Systemgruppe `^ gpio^` zugreifbar.

Lösung: Benutzer `^ www-data^` in die Gruppe `^ gpio^` übernehmen und apache neu starten:

```
sudo usermod -a -G gpio www-data  
sudo /etc/init.d/apache2 restart
```

Oder: GPIO-Dateien Dateirechte setzen:

```
sudo chmod -R +rwX /sys/class/gpio /sys/devices/virtual/gpio
```

