

GPIO

Die Peripherie

Problem:

In mobilen Systemen müssen Peripheriegeräte oft direkt angesprochen werden.

Beispiele:

- ▶ Fahrzeuge sollen gesteuert werden
- ▶ Motoren werden geregelt
- ▶ Umweltdaten wie Temperaturen oder Lichtwerte müssen verarbeitet werden.

Treiber?

Der **direkte** Zugriff auf Hardware ist grundsätzlich nur über den Kernel mit Hilfe von Treibern möglich.

Für komplexe Hardware ist das zweckmäßig, für einfache Komponenten wie eine LED aber nicht angemessen.

Was ist GPIO?

Eingeschränkter Zugriff auf Hardware aus dem **Userspace** ist mit General Purpose Input Output (GPIO) möglich.

Verschiedene so genannte **Pins** können

- ▶ konfiguriert werden,
- ▶ mit Hardware verbunden werden und
- ▶ mit Hardware interagieren.

GPIO ist kein Standard, sondern beschreibt grundsätzlich die Möglichkeit Hardware ‚programmatisch‘ ohne dedizierte Gerätetreiber anzusprechen.

GPIO vereinfacht

Der Zugriff, etwa aus einem C-Programm, auf die Pins ist im allgemeinen **nicht einfach**.

In diesem Kapitel sehen wir an einigen Beispielen, wie man das Pseudo-Dateisystem **sys** von Linux nutzen kann, um über GPIO *vergleichsweise bequem* auf Hardware-Komponenten zuzugreifen.

Die Dokumentation

Eine erste Schwierigkeit besteht darin herauszufinden

- ▶ welche GPIO-Pins existieren und
- ▶ unter welchem Namen sie angesprochen werden.

Diese Informationen entnimmt man im Regelfall der Dokumentation der genutzten Plattform.



Infos im Verzeichnis `sys`

In der Dokumentation zum Raspberry Pi werden die Pins etwa mit `GPIOxx` bezeichnet.

- ▶ Einige dieser Pins sind durch die verbaute Hardware vorgelegt.
- ▶ Andere Pins können frei genutzt werden.

Im Verzeichnis

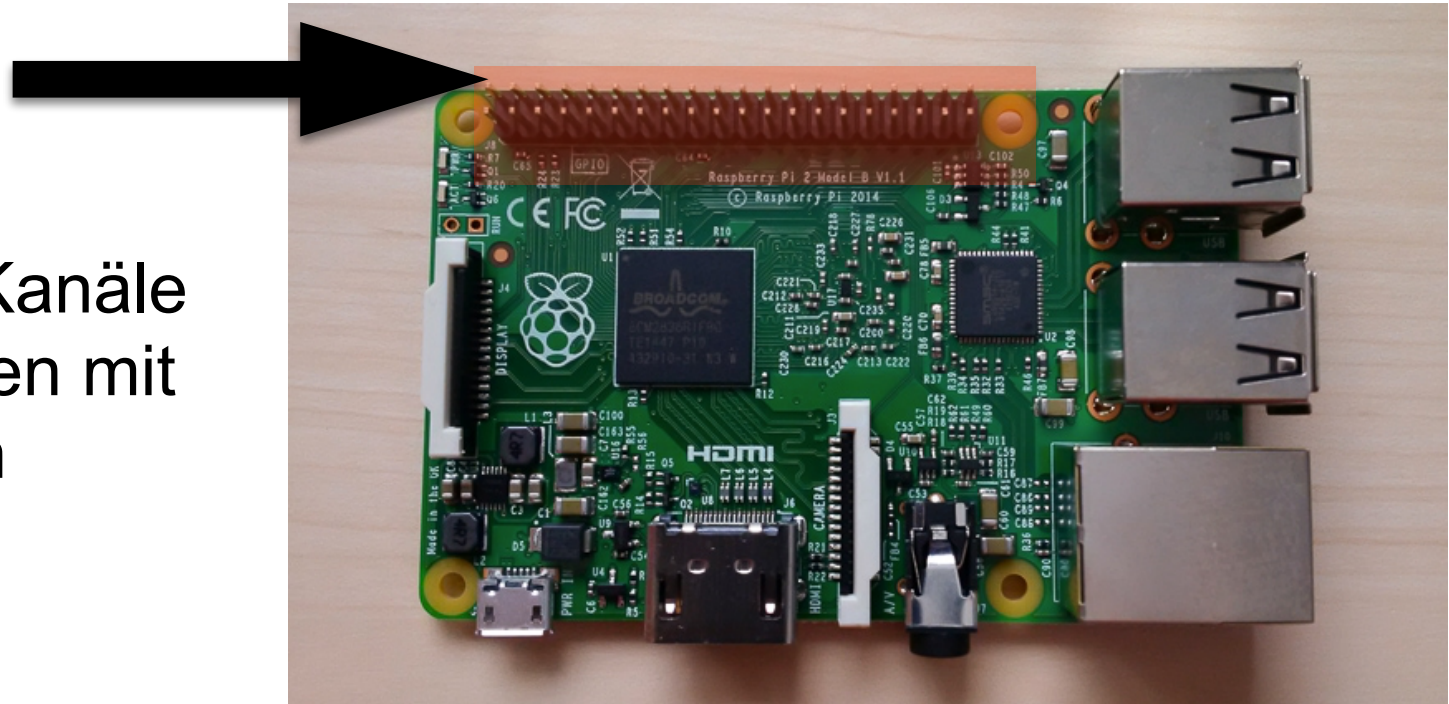
`/sys/class/gpio`

findet Verzeichnisse zu GPIO Pins:

```
export  gpiochip0  unexport
```

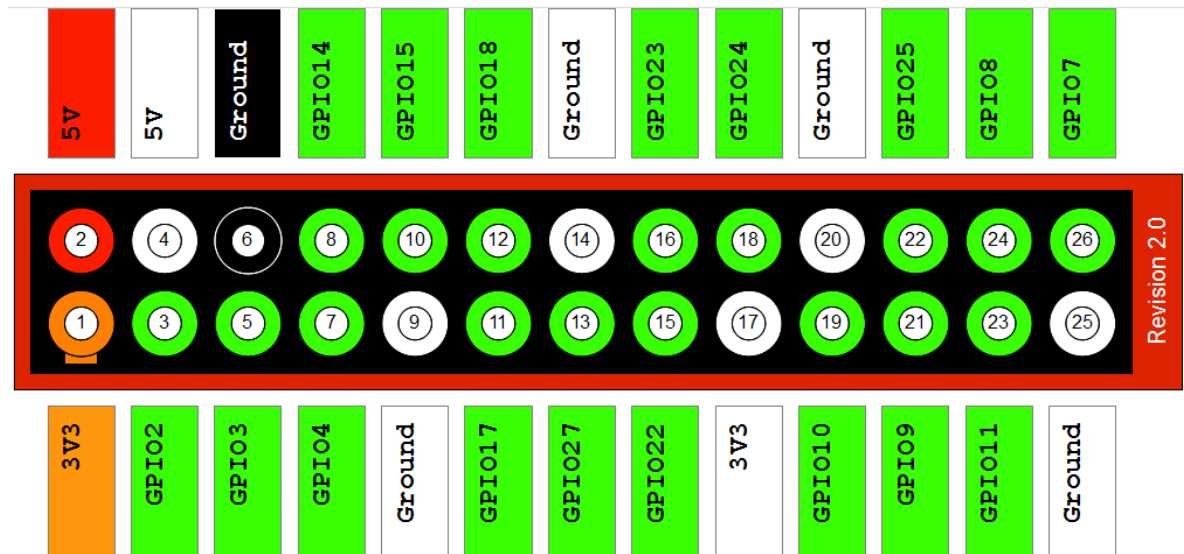
Eigene Bauteile

Einige GPIO-Kanäle korrespondieren mit der markierten Steckerleiste.



Im folgenden wird exemplarisch aufgezeigt, wie **eigene** Komponenten per GPIO angesteuert werden können.

Ein Blick auf die Steckerleiste



Die Pins sind wie folgt nummeriert (siehe Handbuch):

- ▶ Die Nummerierung beginnt mit 1 im linken unteren Pin.
- ▶ In der oberen Reihe befinden sich die geraden in der oberen Reihe die geraden Pins.
- ▶ Die grün markierten Pins können über das sys-Dateisystem angesprochen werden.

Ein Blick auf die Steckerleiste

- ▶ Pin 11 kann als **GPIO17** angesprochen werden.
- ▶ Pin 6, ist in der Tabelle als Ground (Erde/Masse) gekennzeichnet.
- ▶ Wenn wir also eine LED mit jeweils einem Bein mit den Pins 11 und 6 verbinden, sollte die Ansteuerung möglich sein.

Ein passendes Verzeichnis

Wir stecken die LED in die Pins 11 und 6 und steuern die LED über GPIO17 an:

```
cd /sys/class/gpio/  
echo 17 > export  
cd gpio17
```

Es wurde ein neues Verzeichnis `gpio17` angelegt und mit Datei belegt:

<code>active_low</code>	<code>direction</code>	<code>edge</code>	
<code>power</code>	<code>subsystem</code>	<code>uevent</code>	<code>value</code>

Die LED leuchtet

Die Anweisung

```
cat direction
```

liefert den Wert `in`. Wir benötigen GPIO17 aber für die Ausgabe:

```
echo out > direction
```

Wir schalten die LED ein:

```
echo 1 > value
```