

Der GPIO und Python

Mit Hilfe des GPIO lassen sich Elemente durch ein Python-Programm außerhalb des Computers steuern, aber auch von außen kann auf das Programm Einfluss genommen werden. Dazu stehen alle programmierbaren Pins als Eingang und als Ausgang zur Verfügung. Man muss nur im Programm festlegen, welche dieser beiden Eigenschaften sie jeweils einnehmen sollen. Über sie kann jeweils nur True oder False gesendet bzw. empfangen werden.

1 Einstellungen

Um mit dem GPIO arbeiten zu können, müssen mit `import RPi.GPIO as GPIO` die entsprechenden Bibliotheken eingebunden werden. Dann kann man mit `GPIO.setmode(GPIO.BOARD)` die Zählweise der Pins setzen, so dass man sie von 1 bis 26 herunter zählen kann. Durch `GPIO.setup(11, GPIO.OUT)` oder `GPIO.setup(12, GPIO.IN)` sollte anschließend festgelegt werden, ob ein Pin als Ausgang oder Eingang genutzt wird.

2 Ausgänge steuern

Wurden die entsprechenden Einstellungen getroffen, so kann man an entsprechenden Stellen im Programm mit `GPIO.output(11, False)` den entsprechenden Pin ausschalten. In diesem Fall ist es Pin 11. Ein komplettes Programm, was zwischen zwei LEDs hin und her schaltet sähe dann so aus:

```
import RPi.GPIO as GPIO
import time

#Setzen der Nummernzaehlung der Pins. Alternativ GPIO.BCM
GPIO.setmode(GPIO.BOARD)

#Setzen der Pins auf Ausgabe
GPIO.setup(11, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)

#Beide Pins ausschalten
GPIO.output(11, False)
GPIO.output(13, False)

#Und ewiges Wechseln...
while 1:
    GPIO.output(11, True)

    #Fuer 1/5 Sekunde warten
    time.sleep(0.2)

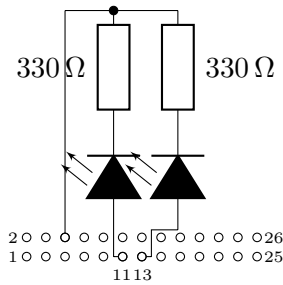
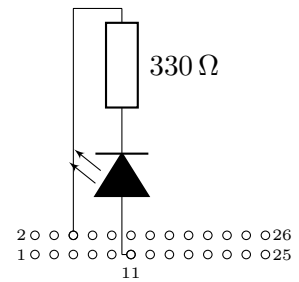
    GPIO.output(11, False)
    GPIO.output(13, True)

    time.sleep(0.2)
    GPIO.output(13, False)
```



3 Anschluss einer LED

Beim Anschluss einer LED an den GPIO muss beachtet werden, dass diese einen Vorwiderstand benötigen, damit sie nicht durchbrennen. Außerdem ist die Anschlussrichtung bei der LED zu beachten. Das längere der beiden Beine muss auf der Seite des geschalteten Ports liegen, während das kürzere mit der Masse (GND) verbunden wird. In der rechten Schaltskizze ist die LED so verbunden, dass sie über den Port 11 geschaltet wird.



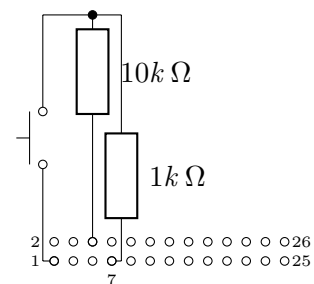
Sollen mehrere LEDs beschaltet werden, bekommt jede LED einen eigenen Vorwiderstand. Es gibt aber eine gemeinsame Verbindung zur Masse. Eine solche Schaltung ist in der linken Schaltskizze aufgezeichnet worden. Bei dieser lassen sich die zwei LEDs über die Ports 11 und 13 schalten.

4 Eingänge

Analog zum Belegen der Ausgänge lässt sich auch in Python mit `GPIO.input(11)` abfragen, ob ein Eingang belegt ist oder nicht. Diese Methode liefert einen entsprechenden Booleschen Wert zurück.

5 Anschluss eines Schalters oder Tasters

Für den Anschluss eines Schalters oder Tasters an den GPIO werden immer drei Anschlüsse benötigt, damit sichergestellt ist, dass der Eingang entweder mit der Masse (GND) oder einer Spannung verbunden ist. Dabei ist zu beachten, dass hier maximal der 3,3 V Anschluss genutzt wird. Bei der nebenstehenden Schaltskizze ist bei geöffnetem Schalter eine direkte Verbindung zwischen dem Pin 7 und der Masse vorhanden. Wird der Schalter gedrückt, so liegt durch den geringeren Widerstand am Schalter eine Spannung von 3,3 V an.



6 GPIO Belegung

Um eine einfache Übersicht über die Ports des GPIO und ihre Belegung zu erhalten, kann man die untere linke Übersicht für den Raspberry Pi Modell B oder Modell A genutzt werden. Schneidet man sie aus und versieht sie an den passenden Stellen mit Löchern, so kann sie auf den GPIO gesteckt werden. Beim Modell B+, A+ und 2B stimmen die ersten 26 Pins in



ihrer Belegung mit dem Modell A und B überein. **Wichtig:** Bei diesem Vorgang sollte der Raspberry Pi nicht mit der Stromversorgung verbunden sein, da es sonst zu Beschädigungen kommen kann.

Bei der rechten Übersicht sind die Seiten vertauscht. Sie wird dann benötigt, wenn man z. B. mit einem alten Festplattenkabel die Anschlüsse des GPIO direkt an ein Steckbrett legt. Durch die Kabelführung kommt es zu dieser Vertauschung.

Möchte man die Pins in Python mit den Nummern ansprechen, die zusätzlich angegeben sind, so ist `GPIO.BCM` bei `GPIO.setmode` anzugeben.

Raspberry Pi

```

3.3V 1 0 0 2 5V
SDA 2 3 0 0 4 5V
SCL 3 5 0 0 6 GND
4 7 0 0 8 14 TXD
GND 9 0 0 10 15 RXD
17 11 0 0 12 18
27 13 0 0 14 GND
22 15 0 0 16 23
3.3V 17 0 0 18 24
MOSI 10 19 0 0 20 GND
MISO 9 21 0 0 22 25
SCKL 11 23 0 0 24 8
GND 25 0 0 26 7

```

Board

```

5V 2 0 0 1 3.3V
5V 4 0 0 3 2 SDA
GND 6 0 0 5 3 SCL
TXD 14 8 0 0 7 4
RXD 15 10 0 0 9 GND
18 12 0 0 11 17
GND 14 0 0 13 27
23 16 0 0 15 22
24 18 0 0 17 3.3V
GND 20 0 0 19 10 MOSI
25 22 0 0 21 9 MISO
8 24 0 0 23 11 SCKL
7 26 0 0 25 GND

```

Raspberry Pi

```

3.3V 1 0 0 2 5V
SDA 2 3 0 0 4 5V
SCL 3 5 0 0 6 GND
4 7 0 0 8 14 TXD
GND 9 0 0 10 15 RXD
17 11 0 0 12 18
27 13 0 0 14 GND
22 15 0 0 16 23
3.3V 17 0 0 18 24
MOSI 10 19 0 0 20 GND
MISO 9 21 0 0 22 25
SCKL 11 23 0 0 24 8
GND 25 0 0 26 7
ID_SD 27 0 0 28 ID_SC
5 29 0 0 30 GND
6 31 0 0 32 12
13 33 0 0 34 GND
19 35 0 0 36 16
26 37 0 0 38 20
GND 39 0 0 40 21

```

Board

```

5V 2 0 0 1 3.3V
5V 4 0 0 3 2 SDA
GND 6 0 0 5 3 SCL
TXD 14 8 0 0 7 4
RXD 15 10 0 0 9 GND
18 12 0 0 11 17
GND 14 0 0 13 27
23 16 0 0 15 22
24 18 0 0 17 3.3V
GND 20 0 0 19 10 MOSI
25 22 0 0 21 9 MISO
8 24 0 0 23 11 SCKL
7 26 0 0 25 GND
ID_SC 28 0 0 27 ID_SD
GND 30 0 0 29 5
12 32 0 0 31 6
GND 34 0 0 33 13
16 36 0 0 35 19
20 38 0 0 37 26
21 40 0 0 39 GND

```

