Tutorial: Raspberry Pi GPIO Pinbelegung und WiringPi Setup

Tutorial: Raspberry Pi - GPIO und Steuerung und Abfrage mit wiringPi

Anschlussbelegung der GPIO des Raspberry Pi und Installation und Grundlagen der Nutzung des Softwarepaketes wiringPi.

Für die Kommunikation mit anderen Modulen, Baugruppen, Sensoren und anderen Peripheriegeräten stellt uns der Raspberry Pi über die 40polige Stiftleiste des GPIO-Steckverbinders diverse Schnittstellen, Ein-/Ausgänge und Betriebsspannungen zur Verfügung.

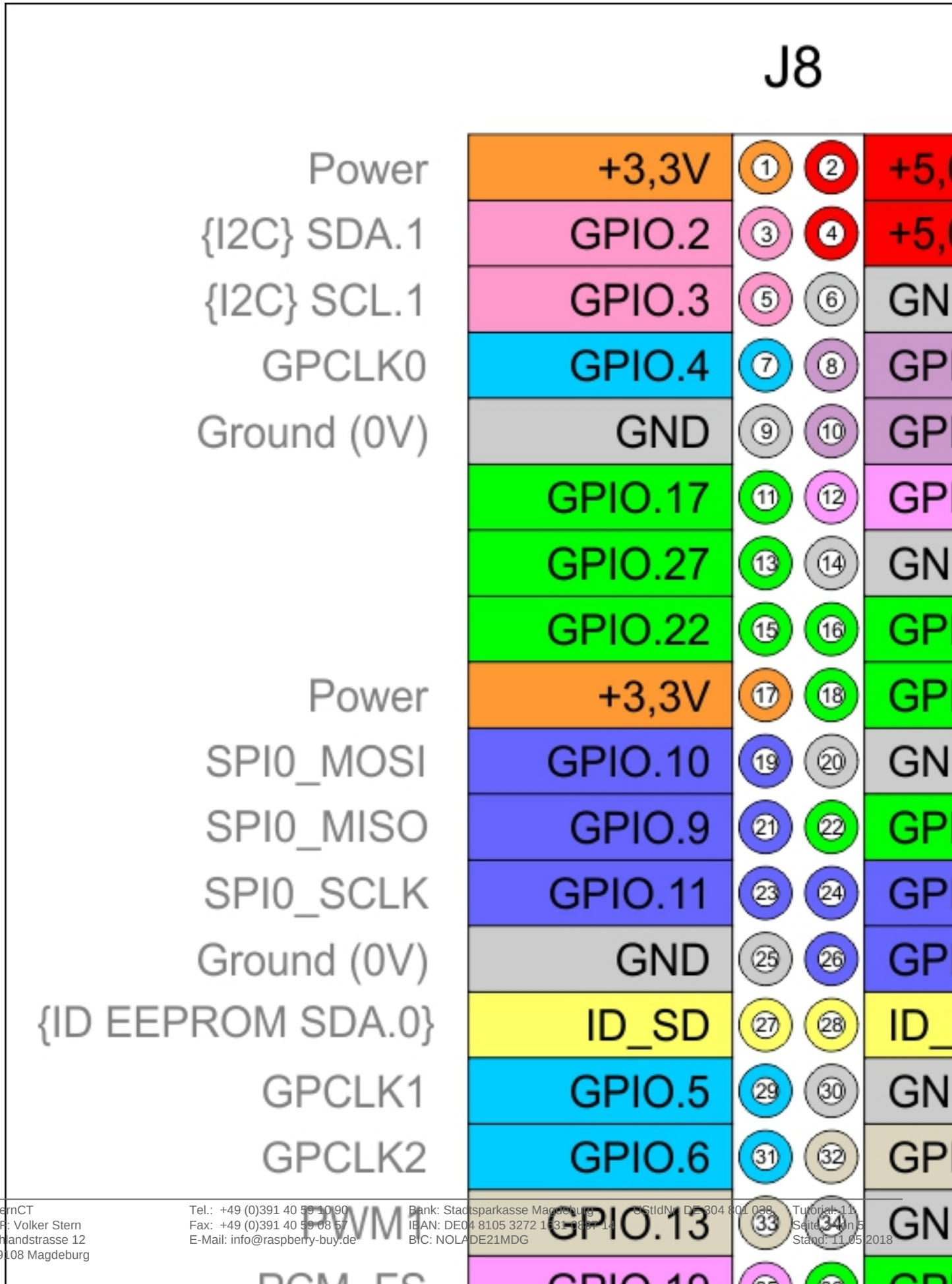
In diesem Tutorial möchte ich Euch die Pinbelegung und, falls notwendg den Setup von Wiring Pi, einem Linux-Modul zur Programmierung der GPIO, vorstellen.

Raspberry Pi - GPIO Pinbelegung

Mit den 40 Pins der GPIO stellt uns der Raspberry Pi seine Schnittstellen und Ein-/Ausgänge zur Verfügung. Um zu wissen, welche der Pins wir für unsere Projekte benutzen, müssen wir natürlich auch die Belegung der einzelnen Anschlüsse kennen.

Für einfache I/O-Operationen ist es sinnvoll, zunächst die freien, nicht mit festen Funktionen belegten oder belegbaren, Anschlüsse (hier grün unterlegt) des GPIO zu verwenden.

Wie die GPIO-Schnittstelle des Raspi beschaltet ist, zeigt Euch die folgende Grafik:


J8
Power
+3,3V
1
2
+5,
{I2C} SDA.1
GPIO.2
3
4
+5,
{I2C} SCL.1
GPIO.3
5
6
GN
GPCLK0
GPIO.4
7
8
GP
Ground (0V)
GND
9
10
GP
GPIO.17
11
12
GP
GPIO.27
13
14
GN
GPIO.22
15
16
GP
Power
+3,3V
17
18
GP
SPI0_MOSI
GPIO.10
19
20
GN
SPI0_MISO
GPIO.9
21
22
GP
SPI0_SCLK
GPIO.11
23
24
GP
Ground (0V)
GND
25
26
GP
{ID EEPROM SDA.0}
ID_SD
27
28
ID_
GPCLK1
GPIO.5
29
30
GN
GPCLK2
GPIO.6
31
32
GP
GPIO.13
33
34
GN

Installation von WiringPi

Wiring Pi ist ein Linux-Modul für den Umgang mit der GPIO des Raspberry Pi. Bei den meisten Images ist Wiring Pi schon vorinstalliert. Um herauszufinden, ob Wiring Pi schon installiert ist, versuchen wir zunächst einmal, die GPIO auszulesen. Dazu geben wir folgende Zeile ein:

pi@rechnername ~ : $ gpio readall (enter)

Wenn Ihr nach dieser Eingabe eine Auflistung mit der aktuellen Belegung und den Zuständen der Ein- und Ausgänge der GPIO bekommt, ist WiringPi schon installiert und Ihr könnt den Rest dieses Schrittes überspringen.

Für alle, bei denen hier jedoch nichts passiert oder eine Fehlermeldung angezeigt wird, geht es jetzt weiter.

Wir installieren einfach das wiringpi Paket. Vorher updaten wir jedoch noch unsere Paketdatenbank und natürliche auch unsere installierten Pakete.

pi@rechnername ~ : $ sudo apt-get update (enter)
pi@rechnername ~ : $ sudo apt-get upgrade (enter)

Eventuelle Fragen, ob wir wirklich updaten wollen und wir uns sicher sind, beantworten wir mit J wie ja. Dieser Vorgang kann, je nachdem, wann Ihr das letzte Mal aktualisiert habt, unter Umständen ein Weilchen dauern.

pi@rechnername ~ : $ sudo apt-get install wiringpi (enter)

Auch hier beantworten wir wohlwollende Fragen wieder mit einem Grossen J wie ja. Nach der Installation sollten wir dann, nachdem wir noch einmal den readall-Befehl ausgeführt haben, endlich die Auflistung aller GPIO-Pins, deren Anschlussbelegung und die aktuellen Zustände der jeweiligen Anschlüsse angezeigt bekommen.

Achtung!!! Für den Raspberry Pi 4 genügt es nicht, einfach die Pakete zu updaten.
Hier funktioniert der readall-Befehl zunächst einmal gar nicht, da auch nach Update und Upgrade noch die wiringpi Version 2.50 installiert ist. Auslesen der Versionsnummer funktioniert wie folgt:

pi@rechnername ~ : $ gpio -v (enter)

Die folgende Ausgabe zeigt uns die aktuell installierte Version.

gpio version: 2.50

Wir benötigen eine Version größer 2.50. Dazu müssen wir die aktuelle wiring-pi Bibliothek installieren das geht so:

pi@rechnername ~ : $ cd /tmp (enter)
pi@rechnername ~ : $ wget https://project-download.drogon.net/wiringpi-latest.deb (enter)
pi@rechnername ~ : $ sudo dpkg -i wiringpi-latest.db (enter)

Die zur Zeit der Erstellung dieses Beitrages aktuelle Version des wiring-pi liegt bei 2.52. Daher bekomme ich nach der Installation und der erneuten Abfrage die folgende Meldung:

pi@rechnername ~ : $ gpio -v (enter)
gpio version 2.52

Jetzt funktioniert auch das readall-Kommando beim Raspberry Pi 4.

WiringPi I/O-Kommandos

Für den Anfang bescheiden wir uns einmal damit, dem GPIO-Port, den wir verwenden wollen, eine Funktion (Eingang oder Ausgang) zuzuweisen.

Da es sich anbietet, einen freien (nicht mit anderen Funktionen belegten) Pin des Raspberry Pi zu verwenden, wähle ich für unser Tutorial mal den Hardware-Anschluss-Pin 11, der dem GPIO-Pin 17 zugeordnet ist.

Zunächst einmal machen wir den Pin zu einem Ausgang. Das geht so:

pi@rechnername ~ : $ gpio export 17 out (enter)

Jetzt schauen wir uns mit readall wieder an, ob der Pin nun auch ein Ausgang ist::

pi@rechnername ~ : $ gpio readall (enter)

In der Auflistung sollte der Pin11 (GPIO 17) jetzt als Ausgang angezeigt werden. Nun bietet es sich ja nahezu an, diesen neu geschaffenen Ausgang auch einmal zu schalten. Dazu setzen wir den Pin auf logisch HIGH oder 1. Auch dafür genügt wieder eine einzelne Kommandozeile:

pi@rechnername ~ : $ gpio -g write 17 1 (enter)

Das folgende Kommando liest dann den einzelnen Pin wieder aus. Natürlich könnt Ihr auch wieder die readall-Funktion benutzen.

pi@rechnername ~ : $ gpio -g read 17 (enter)
1

Die Ausgabe 1, die auf die Abfrage folgt, zeigt uns, dass das Kommando erfolgreich ausgeführt wurde.

Um aus diesem Pin einen Eingang zu machen, tippen wir einfach die folgende Zeile:

pi@rechnername ~ : $ gpio export 17 in (enter)

Das Einlesen des Zustandes des Einganges erfolgt dann auf die gleiche Art und Weise, mit der wir den Status des Ausganges abgefragt haben. Damit lassen wir es mit den I/O-Kommandos auch erst einmal gut sein.

Ansteuerung mit Python

Um die GPIO mit der Programmiersprache Python zu kontrollieren, müssen wir zuerst einmal Python und die GPIO-Bibliotheken für Python installieren. Das machen wir wie folgt:

pi@rechnername ~ $ sudo apt-get update (enter)
pi@rechnername ~ $ sudo apt-get install python-dev (enter)
pi@rechnername ~ $ sudo apt-get install python-rpi.gpio (enter)

Auch hier beantworten wir wieder alle Fragen, ob wir uns sicher sind, dass wir die Installation durchführen wollen,

mit einem überzeugten J wie ja. Damit sind wir dann für unser erstes Python-Script gerüstet.