

Raspberry Pi Einführung

Adrian Wälchli

2. April 2019

1 Überblick

Der Raspberry Pi¹ ist ein Computer im Taschenformat mit minimaler Ausstattung (siehe Abb. 1). Eingebaut ist unter anderem ein 1.2 GHz Vierkernprozessor, 1 GB RAM, eine Micro-SD Karte mit vorinstalliertem *Raspbian* Betriebssystem, eine Ethernet Buchse, Wireless LAN, vier USB Anschlüsse und ein HDMI Anschluss. Sie werden im Rahmen der Vorlesung *Rechnerarchitektur* zwei Programmieraufgaben lösen, in denen Sie den Raspberry Pi und dessen Schnittstelle verwenden um auf LEDs, Tasten und andere elektronische Bauteile zuzugreifen. Zu diesem Zweck erhalten Sie ein Gesamtpacket mit folgenden Komponenten.

- Raspberry Pi
- Gehäuse
- Stromkabel
- HDMI Kabel
- SD-Karte (bereits eingesteckt)
- GPIO Extension Board mit Verbindungskabel
- Steckplatine
- Zusatzbauteile (vgl. Tabelle 1)

Im Packet enthalten ist eine von uns vorbereitete Steckplatine, welche sich über das Verbindungskabel mit den GPIO-Pins am Pi verbinden lässt.

Um das Gerät in Betrieb zu nehmen und damit programmieren zu können benötigen Sie zusätzlich Tastatur, Maus und einen Monitor mit HDMI Anschluss. Sie können Ihre eigene Peripherie oder die im ExWi Computerpool² verwenden.

2 Starten und Einrichten

Beim Raspberry Pi gibt es keinen Einschaltknopf. Das Gerät schaltet sich ein sobald die Stromverbindung über das Stromkabel hergestellt wird. Sie sollten innerhalb von 30 Sekunden zum Anmeldebildschirm gelangen. Falls dies nicht der Fall ist, überprüfen Sie die Verbindung zum Monitor. Nachdem Sie sich angemeldet haben erscheint der Raspbian Desktop (Abb. 1) über den Sie Programme und Einstellungen aufrufen können.

Zum Lösen der Aufgaben brauchen Sie lediglich die Kommandozeile, welche Sie ebenfalls über das Raspbian Hauptmenü starten können. Sie dürfen selbstverständlich auch weitere Programme installieren, wenn Sie das wünschen. Wir nehmen an, dass Sie grundlegende Kenntnisse zur Bedienung einer Linux Kommandozeile besitzen. Ist dies nicht der Fall, informieren Sie sich bitte über die Grundbefehle zur Navigation in Verzeichnissen, Erstellen und Löschen sowie Editieren von Dateien, usw. Als Texteditor empfehlen wir nano oder vim.

¹https://en.wikipedia.org/wiki/Raspberry_Pi

²Sidlerstrasse 5, Raum A93 und A94

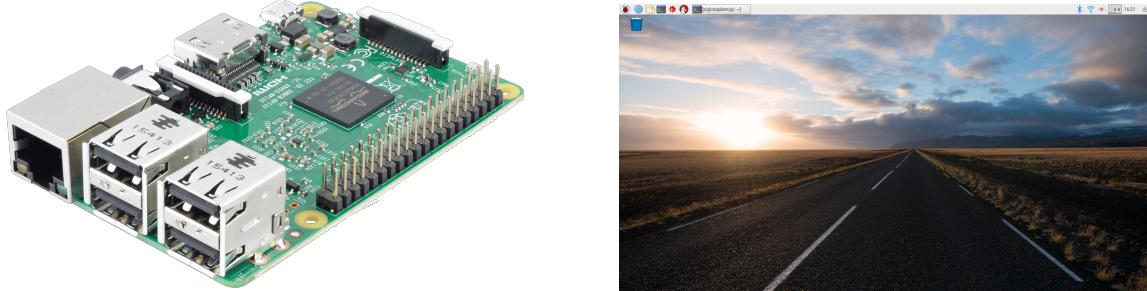


Abbildung 1: Raspberry Pi 3 Modell B und Raspbian Desktop.

3 Verwendung von GPIO und Steckplatine

GPIO Schnittstelle Wie in Abbildung 1 zu sehen ist, hat der Pi eine Leiste mit 40 Pins. Diese Schnittstelle nennt sich *GPIO* (general purpose input/output). Jeder Pin kann einzeln angesteuert und zu verschiedenen Zwecken verwendet werden. Mit dem Befehl

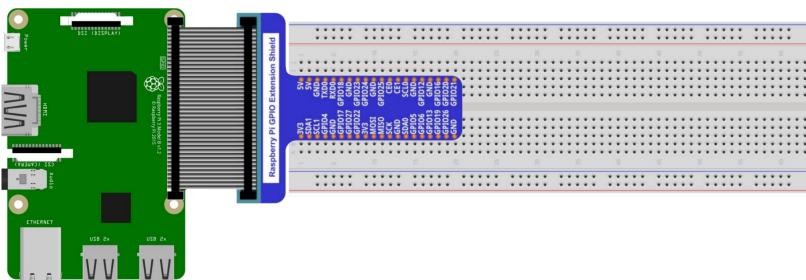
```
$ gpio readall
```

verschaffen Sie sich einen Überblick über die Benennung und den Zustand jedes Pins.

Pi 3										
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
2	8	3.3v			1	2		5v		
3	9	SDA.1	IN	1	3	4		5v		
4	7	SCL.1	IN	1	5	6		0v		
		9v			9	10	0	IN	TxD	15 14
17	0	GPIO. 0	IN	0	11	12	0	IN	RxD	16 15
27	2	GPIO. 2	IN	0	13	14		0v		18
22	3	GPIO. 3	IN	0	19	20	0	IN	GPIO. 4	4 23
		2.3V			7	18	0	IN	GPIO. 5	5 24
10	12	MOSI	IN	0	21	22	0	IN	GPIO. 6	6 25
9	13	MISO	IN	0	23	24	1	IN	CE0	10 8
11	14	SCLK	IN	0	21	20	1	IN	CE1	11 7
		3.3V			1	2	0	IN	SDA.0	31 1
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31 1
5	21	GPIO.21	IN	1	29	30		0v		
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26 12
13	23	GPIO.23	IN	0	33	34		0v		
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27 16
20	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28 20
		0v			39	40	0	IN	GPIO.29	29 21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM

Die Zwei Spalten in der Mitte repräsentieren die physikalische GPIO Schnittstelle, wobei die Pins hier von 1-40 durchnummieriert sind. Die Spalten links und rechts davon beschreiben die Einstellungen der Pins. Die Spalte **Mode** zeigt den Modus eines Pins. Für Sie relevant sind insbesondere die Modi **IN** (Eingang) und **OUT** (Ausgang). Der momentane Eingangs- oder Ausgangswert eines Pins sehen Sie in der Spalte **V**. Die beiden Spalten **Name** und **wPi** können Sie ignorieren. Dies sind im Wesentlichen nur andere Namen für dieselben Pins. In den Übungen werden wir die "BCM" Namensgebung verwenden, um die Pins zu identifizieren (erste bzw. letzte Spalte).

Steckplatine Um den Umgang mit den Pins und das Verbinden mehrerer elektronischer Komponenten zu vereinfachen verwenden wir eine Steckplatine. Verbinden Sie diese mit dem Pi wie unten abgebildet.



Diese Erweiterung erlaubt es Ihnen ganz einfach das Signal einzelner Pins abzuzweigen und zu verschiedenen Bauteilen zu leiten. Des Weiteren sind die Namen der Pins in BCM-Konvention aufgedruckt. Wie angeschrieben werden einzelne Pins für konstante Stromzufuhr (3.3V, 5V, GND) verwendet.

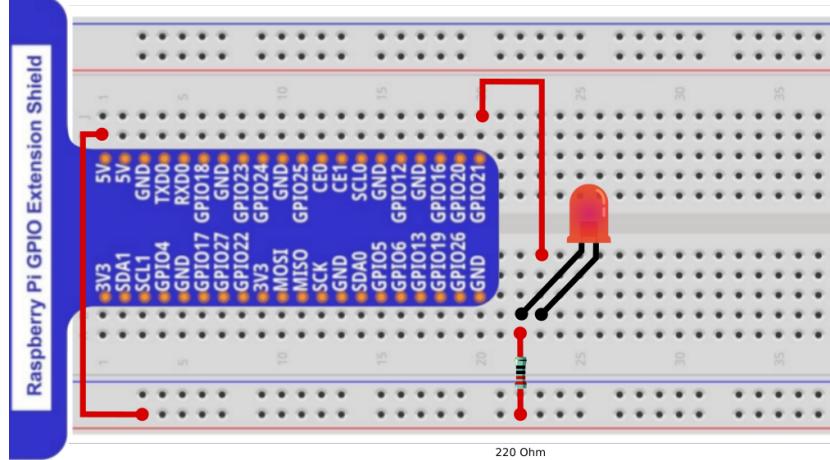


Abbildung 2: LED Schaltungsdiagramm.

Da Sie bereits eine vorbereitete Schaltplatine von uns erhalten, benötigen Sie vorerst keine weiteren Kenntnisse über die genaue Schaltung der Komponenten. Die nötigen Informationen dazu folgen in den jeweiligen Aufgabenteilen der Serien 4 und 5.

LED Test Das Kommandozeilenprogramm `gpio` ermöglicht es Ihnen, die Pins einzeln anzusteuern. Wie in Abbildung 2 dargestellt ist auf Ihrer Steckplatine eine LED über den Pin 21 geschaltet. Um die LED ein- oder auszuschalten müssen Sie den Modus des Pins zuerst auf "Ausgang" stellen. Dies geht ganz einfach mit dem Befehl

```
$ gpio -g mode 21 out
```

Nun kann die LED mit dem Befehl

```
$ gpio -g write 21 1
```

bzw. mit

```
$ gpio -g write 21 0
```

ein- und ausgeschaltet werden. Testen Sie es selbst! Mit dem zuvor beschriebenen Befehl `gpio readall` können Sie nun auch die Änderungen bei Pin 21 feststellen. Mit `man gpio` erhalten Sie einen Überblick über die verschiedenen Anwendungsmöglichkeiten vom `gpio` Tool. *Übung:* Finden Sie heraus, wie ein Eingangssignal am Pin 18 gelesen werden kann.

4 Assembler Tutorial: Blinkende LED

Als Einleitung zu den Assembleraufgaben besprechen wir hier ein einfaches Programm, welches die LED auf der Steckplatine zum blinken bringt. Dies geschieht ähnlich wie Sie es im vorherigen Test schon von Hand ausgeführt haben. Wir haben die nötigen Dateien bereits in Ihrem Heimverzeichnis hinterlegt, so dass Sie den hier besprochenen Code direkt auf Ihrem Pi ausführen können³. Navigieren Sie ins Verzeichnis mit dem Quellcode und kompilieren Sie mit `make`.

```
$ cd ~/Documents/blink
$ make
```

Danach können Sie das Programm ganz einfach ausführen.

```
$ ./blink
```

Sie sollten nun eine blinkende LED sehen. Mit der Tastenkombination `Ctrl+C` stoppen Sie das Programm. Falls die LED beim Ausführen des Codes nicht leuchtet, überprüfen Sie die Verbindungen auf der Steckplatine und stellen Sie sicher, dass diese der Abbildung 2 entsprechen.

³Sie finden den Code auch auf ILIAS.

Öffnen Sie nun die Quellcodedatei `blink.s` mit einem Editor Ihrer Wahl. Dies geht zum Beispiel mit

```
$ nano blink.s
```

direkt im Terminal. Sie sehen nun den in Auflistung 1 abgebildeten Code. Der Quellcode ist in verschiedene Blöcke unterteilt: `main`, `configurePin`, `blinkLoop` und `exit`. Der Einstiegspunkt ist bei `main`, wo die externe Subroutine `wiringPiSetupGpio` aufgerufen wird. Diese initialisiert unter anderem den Zugriff auf den GPIO Adressbereich.

Im nächsten Abschnitt `configurePin` wird der LED Pin auf Ausgangsmodus gesetzt, ähnlich wie Sie dies schon manuell mit dem Befehl `gpio` gemacht haben. Hierbei ist folgendes zu beachten:

- `pinMode(pin, mode)` ist eine extern aufrufbare Funktion aus der vorinstallierten Bibliothek `wiringPi`.
- Die beiden Argumente werden vor dem Aufruf in den Registern `R0` und `R1` abgelegt.
- Die Konstanten `.LED_PIN` und `.OUTPUT` sind am Ende des Quelltextes definiert.

Nach dem gleichen Prinzip werden im Abschnitt `blinkLoop` die Funktionen `digitalWrite` und `delay` aufgerufen. *Frage:* Was macht der Befehl `digitalWrite`? *Übung:* Ändern Sie den Quellcode so ab, dass die LED schneller blinkt.

Auflistung 1: blink.s

```
1 .global main
2 .func main
3
4 main:
5     // This will setup the wiringPi library.
6     // In case something goes wrong, we exit the program.
7     BL    wiringPiSetupGpio
8     CMP   R0, #-1
9     BEQ   exit
10
11 configurePin:
12     // Here we configure the pin.
13     // We use the pin number 21 as defined at the bottom of
14     // this file and set the pin to output mode.
15     LDR   R0, .LED_PIN
16     LDR   R1, .OUTPUT
17     BL    pinMode
18
19 blinkLoop:
20     // This is where the loop for blinking the LED starts
21     // Turn the LED on
22     LDR   R0, .LED_PIN
23     LDR   R1, .HIGH
24     BL    digitalWrite
25
26     // Wait 500 milliseconds
27     MOV   R0, #500
28     BL    delay
29
30     // Turn the LED off
31     LDR   R0, .LED_PIN
32     LDR   R1, .LOW
33     BL    digitalWrite
34
35     // Wait 500 milliseconds
36     MOV   R0, #500
37     BL    delay
38
39     // Repeat
40     B     blinkLoop
41
42 exit:
43     MOV   R7, #1
44     SWI   0
45
46
47 // We use GPIO pin 21 (BCM-style) to connect the LED.
48 .LED_PIN:           .word   21
49
50 // Define constants for high- and low signals on the pins
51 .HIGH:              .word   1
52 .LOW:               .word   0
53
54 // The mode of the pin can be set to input or output.
55 .OUTPUT:            .word   1
56 .INPUT:             .word   0
```

Name	Anzahl	Name	Anzahl
Raspberry Pi	1	Gehäuse für Raspberry Pi	1
Stromkabel	1	HDMI Kabel	1
SD-Karte	1	Steckplatine	1
GPIO Aufsatz für Steckplatine	1	40 Pin GPIO Verbindungsleitung	1
LED rot/grün/blau	1	Druckknopf	2
LED Leiste	1	NPN Transistor	1
Widerstand 220 Ω	9	74HC595 Schieberegister	1
Widerstand 1 kΩ	1	Aktiver Signaltongeber	1
Widerstand 10 kΩ	4	Schaltdraht M/M	26

Tabelle 1: Raspberry Pi Gesamtpacket Inhalt.