

Workshop 4: Raspberry Pi

Ziele

In diesem Workshop nehmen Sie einen Raspberry Pi selbst in Betrieb, schließen Sensoren und Aktoren an und werden diese mit Hilfe der Programmiersprache Scratch programmieren können.

Je nach Ihrem Wissensstand können Sie einzelne Aufgaben überspringen bzw. schnell bearbeiten und zur nächsten Aufgabe übergehen! Die angegebenen Zeiten sind nur Anhaltswerte!

Einführung Raspberry Pi

Der **Raspberry Pi** ist ein Einplatinencomputer, der von der britischen *Raspberry Pi Foundation* entwickelt wurde. Die Grundfläche der Platine entspricht etwa den Abmessungen einer Kreditkarte. Der Raspberry Pi kam Anfang 2012 auf den Markt; bis September 2016 wurden mehr als zehn Millionen Geräte verkauft. Er wird als Revival der früheren Heimcomputer (C64, Apple][e) zum Programmieren und Experimentieren angesehen. Der im Vergleich zu üblichen Personal Computern (PC) sehr einfach aufgebaute Rechner wurde mit dem Ziel entwickelt, jungen Menschen den Erwerb von Programmier- und Hardwarekenntnissen zu erleichtern. Entsprechend niedrig wurde der Verkaufspreis angesetzt, der je nach Modell etwa 10 bis 70€ beträgt.

Mittlerweile existieren umfangreiche Softwaresammlungen für verschiedenste Anwendungsbereiche (Mediacenter, Internetfirewall, Steuerung elektronischer Schaltungen). Als Betriebssystem kommen vor allem angepasste Linux-Versionen mit grafischer Benutzeroberfläche zum Einsatz (am weitesten verbreitet ist „Raspbian“, das vom das Linux-Derivat Debian abstammt); für das neueste Modell existiert auch Windows 10 in einer speziellen Version ohne grafische Benutzeroberfläche.

Aufgabe 1: Einführung in den Raspberry Pi (5min)

a) Komponenten

Überprüfen Sie die Anschlüsse des Raspberry Pi:

- *USB-Tastatur, USB-Maus*
- *HDMI-Kabel*
- *Stromkabel (Mini-USB)*
- *Massenspeicher (microSD-Karte)*
- *weitere Anschlussmöglichkeiten (z.B. LAN)*
- *WLAN-Modul (integriert)*

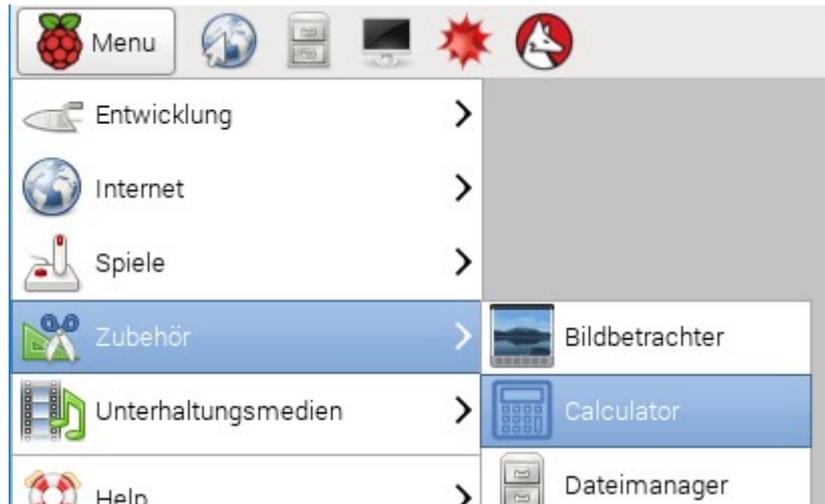
b) Betriebssystem und Bedienungsoberfläche

Starten Sie den Taschenrechner und berechnen Sie die Fakultät von 7 (7!).

Hinweis:

Menü → Zubehör → Calculator

Taschenrechner →
Notationsmodus →
Formeleingabe



Anmerkungen:

Es können auf dem Raspberry Pi die „üblichen“ Programme installiert und genutzt werden:

- LibreOffice
- Mozilla Firefox und Thunderbird
- Gimp
- Java
- zusätzlich „Mathematica“ in einer kostenlosen Version

Die Bedienung der Oberfläche ist ähnlich bekannter Systeme, enthält zusätzlich eine umfangreiche Kommandozeile („shell“).

Optional: Finden und starten Sie weitere Programme.

Aufgabe 2: Einführung in die Programmierumgebung Scratch (10min)

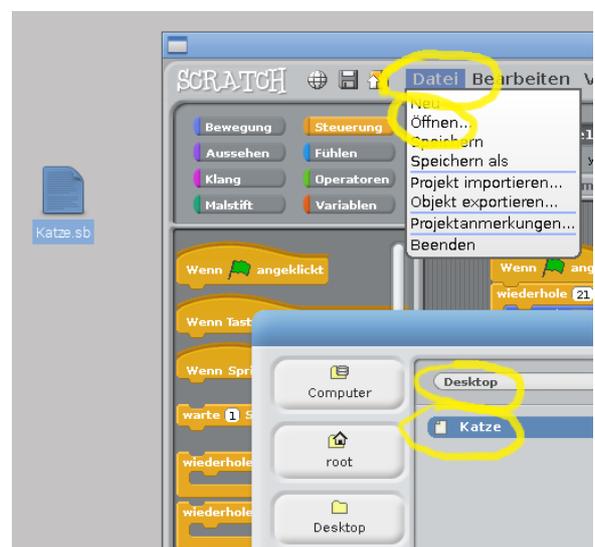
Scratch ist eine erstmals 2007 veröffentlichte visuelle Programmiersprache für Kinder und Jugendliche. Ihr Ziel ist es, Neueinsteiger – besonders Kinder und Jugendliche – mit den Grundkonzepten der Computerprogrammierung vertraut zu machen.

a) Scratch starten

Starten Sie das Programm „ScratchGPIO 7“.
Öffnen Sie nun das vorbereitete Scratch-Programm „Katze“ im Pfad /home/pi.

Anmerkungen:

Die Version „ScratchGPIO 7“ wird benötigt, um auf die GPIOs zugreifen zu können. Scratch ohne diese Erweiterung (via Menü) kann die GPIOs nicht nutzen.



b) Programm „Katze“ ausführen

*Klicken Sie auf die grüne Flagge und beobachten Sie die Aktionen auf dem Bildschirm.
Beschreiben Sie die Aktionen in Bezug auf den Reiter „Skripte“.*

c) Programm „Katze“ verändern.

Erweitern Sie das Programm durch „Drag‘n‘Drop“ so, dass die Katze am Ende der Zählschleife noch 100 Schritte „geht“ und nach unten „schaut“. Starten Sie Ihr Programm und überprüfen Sie den Erfolg.

Optional: Erweitern Sie das Programm beliebig.

Aufgabe 3: Elektronische Schaltungen am RaspberryPi mittels Scratch programmieren (10+10min)

Der Raspberry Pi besitzt neben seinen „üblichen“ Anschlüssen noch eine zusätzliche Steckerleiste. Mittels dieser Steckkontakte können elektronische Schaltungen angeschlossen und gesteuert werden. Hierbei ist aber stets von einem gewissen Grundverständnis des Bedieners auszugehen, da keinerlei Sicherungsschaltungen existieren und fehlerhafte Schaltungen den Raspberry Pi zerstören können!

Diese zusätzlichen Anschlüsse sind nummeriert und werden „GPIO“ (General Purpose Input Output – allgemeine Ein-/ Ausgabe) genannt.

Die Belegung des aktuellen Modells entnehmen Sie dieser Darstellung, wobei der Raspberry Pi in der Darstellung so ausgerichtet ist, dass die USB-Anschlüsse „unten“ liegen:

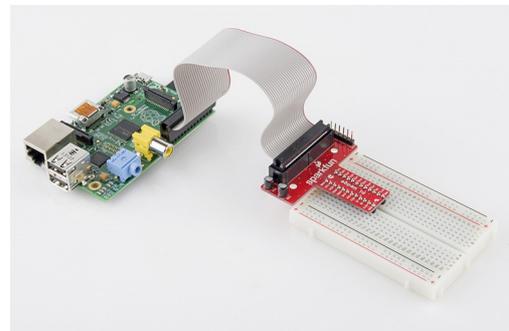
Die GPIOs können auch als Eingaben genutzt werden und so z.B. Sensoren (Schalter, Ultraschallsensoren, lichtempfindliche Widerstände) erfassen.

Hinweis:

Für umfangreichere Aufgaben werden üblicherweise sogenannte „Breadboards“ genutzt. Mit ihnen ist es möglich, die Anschlüsse des Raspberry Pi einfacher zu verbinden (ohne „fliegende Verkabelung“).

Die jeweils links und rechts liegenden Anschlussmöglichkeiten entsprechen den Pins der Steckerleiste am Raspberry Pi.

Weitere Hinweise dazu finden sich im Internet (z.B. <https://www.datenreise.de/raspberry-pi-wie-verwendet-man-ein-breadboard-steckplatine-anleitung/>)



Raspberry Pi 3 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

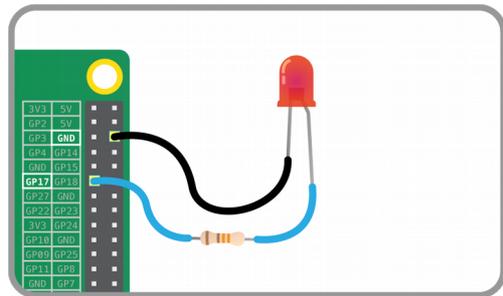
Rev. 2
29/02/2016

www.element14.com/RaspberryPi

Teil1 (Ausgaben steuern)

Schließen Sie die bereit liegende LED so an, dass ihr „kurzes“ Bein an „GND“ (hier Pin 6) angeschlossen ist und zwischen dem Langem Bein und „GPIO17“ (Pin 11) der Widerstand geschaltet ist. Nutzen Sie dazu die vorhandenen Steckkabel.

Bei Unsicherheiten wenden Sie sich an den Referenten!



Anmerkung:

Der Raspberry Pi liefert an seinen GPIOs eine Spannung von 3,3V (und akzeptiert als Input ebenfalls max. 3,3V); die vorhandenen LEDs benötigen eine niedrigere Spannung, so dass ein Vorwiderstand als Spannungsteiler in den Stromkreis eingefügt werden muss.

Öffnen Sie das Scratch-Programm „LED“. Führen Sie es aus. Beschreiben Sie die erfolgenden Aktionen und vergleichen Sie mit dem vorliegenden Programmcode.



Anmerkung:

Der Programmcode öffnet zu Beginn die allgemeine Kommunikation mit den GPIO-Ports („sende *gpioserveron* an alle“). Danach wird der GPIO mit der Nummer 17 als Ausgang definiert („sende *config17out* an alle“).

Die verbleibenden Sende-Befehle schalten den GPIO-Port an bzw. aus.

Optional: Verändern Sie den Code so, dass die LED in Abhängigkeit eines einzugebenden Wertes an bleibt.

Teil2 (Input auslesen)

Lassen Sie die LED angeschlossen und schließen Sie den bereit liegenden Taster an. Ein Anschluss an GPIO4, der andere an einen Ground (GND, z.B. Pin 39). Der Taster schließt bzw. öffnet jeweils die beiden untereinander liegenden Anschlüsse.

Öffnen Sie das Programm „Taster“ und führen Sie es aus. Vergleichen Sie den Programmcode mit den Aktionen.



Anmerkung:

Scratch nummeriert die Anschlüsse z.T. leider nicht konsistent, so dass in manchen Fällen „gpio4“ genutzt werden muss. In anderen Fällen die Nummer des Pins (hier: 7). Ggf. sind beide Varianten zu testen. Die Pin-Nummern können der GPIO-Übersicht entnommen werden.

Programmieren Sie eine Steuerung der LED, so dass die LED nach dem Tastendruck für 5 Sekunden an ist und danach wieder aus.

Anmerkung:

Scratch unterstützt keine Interrupt-gesteuerten Abfragen, so dass in einer Schleife die Stellung des Tasters im Sekundentakt abgefragt werden muss.

Zusatzaufgabe: Vertiefung der GPIO Anschlüsse des RasPi (verbleibende Zeit)

a) Scratch

Programmieren Sie in Scratch eine Steuerung so, dass die LED nach drei Tastendrücken in Abhängigkeit eines vorher eingegebenen Wertes an schaltet und danach wieder aus geht.

b) Python

Öffnen Sie eine Kommandozeile („Terminal“) und führen Sie mit dem Befehl „sudo python LED.py“ ein Python-Programm aus. Beschreiben Sie die Funktionsweise.

Sehen Sie sich den Programmcode mittels „less LED.py“ an und vergleichen Sie den Code mit dem Scratch-Programm „LED“.

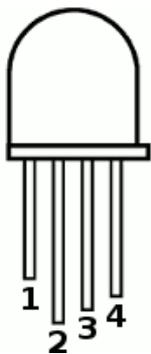
Optional: Schließen Sie die RGB-LED an und erstellen Sie mittels Python ein Programm, das die verschiedenen LED-Farben in Abhängigkeit eines Eingabewertes schaltet. Beachten Sie hierbei auch wieder die Länge der LED-Beine (das Längste ist GND) sowie den nötigen Vorwiderstand.

Anmerkung:

Nutzen Sie als Vorbild die Schaltung für die rote LED und vergessen Sie den Vorwiderstand nicht!

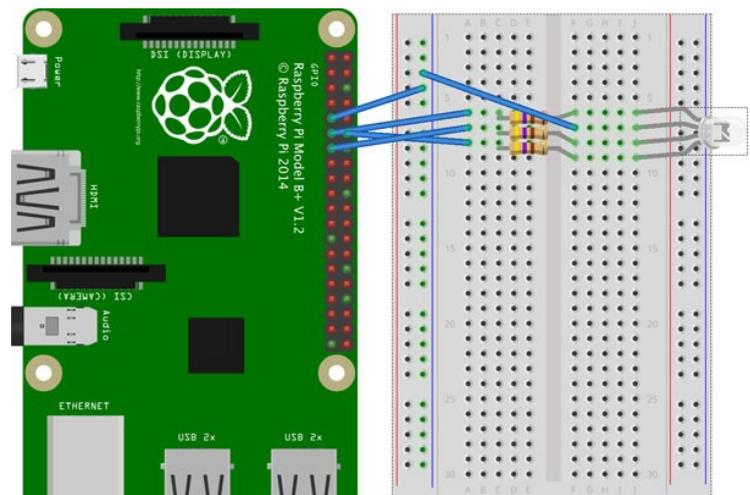
Hinweise:

Führen Sie die Schaltung gem. der Abbildung durch und beachten Sie dabei die Pin-Nummerierung.



RGB LED

- 1: Green (+)**
- 2: Ground (-)**
- 3: Blue (+)**
- 4: Red (+)**



Abschließende Hinweise:

Rund um den Raspberry Pi existiert im Internet eine große Gemeinde. Beispiele für Schaltungen finden sich zu Hauf! Nutzen Sie diese Quellen für eigene Ideen.