

Vorworte

Über SunFounder

SunFounder ist ein sich auf die Open Source Community der Raspberry Pi und Arduino konzentrierendes Technologieunternehmen. Zur Popularisierung der Open Source Kultur verpflichtet, bemühen wir uns, Menschen auf der ganzen Welt den Elektronikherstellungspaß zu bringen und dass Jeder ein Hacker sein kann. Unsere Produkte umfassen Lernsets, Entwicklungsboards, Roboter, Sensormodule und Programmierwerkzeuge. Neben hochwertigen Qualitätsprodukten, bietet SunFounder auch Video-Tutorials, die hilfreich für Ihre eigenen Projekte sind. Wenn Sie Interesse an Open Source haben oder etwas cool machen möchten, nehmen Sie bitte mit uns Kontakt auf.

Über Super Kit

Dieser Super Kit eignet sich für Raspberry Pi B, Modell B +, Raspberry Pi 2 Modell B und Pi 3.

In diesem Buch zeigen wir Ihnen die Schaltungen mit realistischen Abbildungen und schematischen Diagrammen. Interesse geweckt? Dann besuchen Sie uns doch einfach auf unsere Homepage www.sunfounder.com, um betreffende Codes herunterzuladen durch Klicken auf [LEARN](#) -> [Get Tutorials](#) und [Video](#) anzusehen.

Kostenloser Support



Falls Sie **technische Fragen** haben, erstellen Sie bitte ein Thema im **FORUM** auf unserer Webseite. Darauf werden wir so schnell wie möglich antworten.



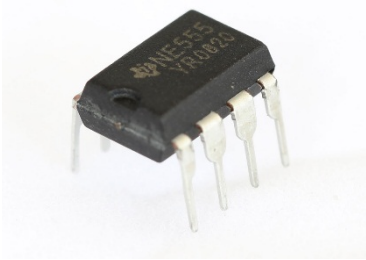
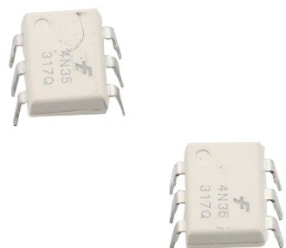
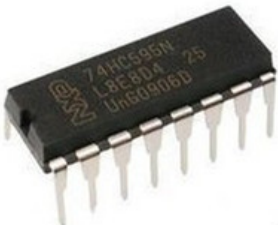



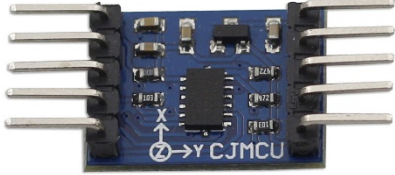
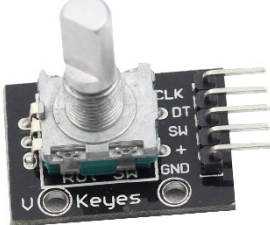






Für **nichttechnische Fragen** wie über Bestellung oder Lieferung, senden Sie bitte eine Email an service@sunfounder.com. Es ist auch zu begrüßen, Ihre Projekte im FORUM zu teilen.


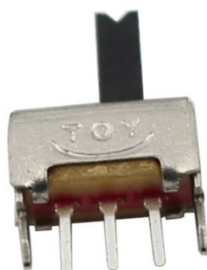
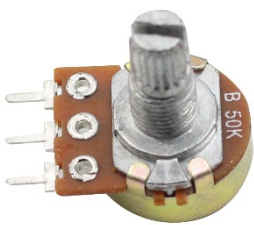
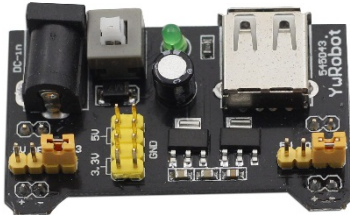
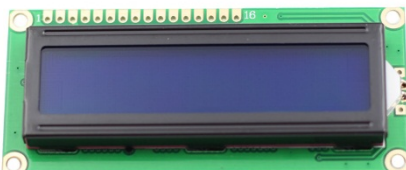
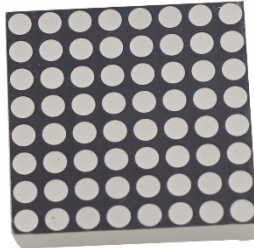
Verzeichnis

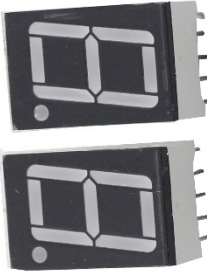





Komponentenliste	1
Hinweis	7
Nummerierung für Raspberry Pi Pins	12
Source Code bekommen	18
Lektion 1 Blinkende LED	19
Lektion 2 LED ansteuern mit Druckschalter	22
Lektion 3 Lauflicht	26
Lektion 4 Atmungslicht	29
Lektion 5 RGB LED	32
Lektion 6 Buzzer	35
Lektion 7 Gleichstrommotor antreiben	38
Lektion 8 Drehgeber	42
Lektion 9 555 Timer	45
Lektion 10 LEDs antreiben mit 74HC595	50
Lektion 11 7-Segment-Anzeige antreiben mit 74HC595	54
Lektion 12 Punktmatrix-Anzeige antreiben mit 74HC595	60
Lektion 13 LCD1602	64
Lektion 14 ADXL345	67
Weiterte Anwendung des Raspberry Pi – Auf Basis von Web eine LED ansteuern	69






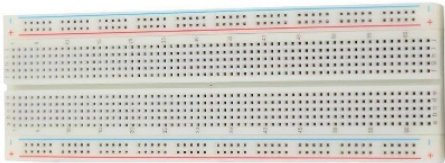
Komponentenliste

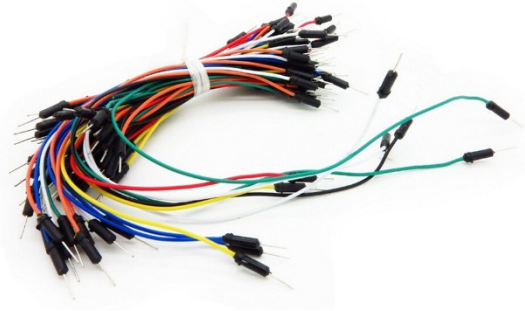



Nr.	Produktname	Menge	Bild
1	RGB LED	1	
2	Pin Header	40	
3	555 Timer	1	
4	Optokoppler (4N35)	2	
5	Schieberegister (74HC595)	2	
6	L293D	1	

7	Beschleunigungssensor (ADXL345)	1	
8	Drehgeber	1	
9	Druckschalter	5	
10	Widerstand (220Ω)	8	 (Rot, Rot, Schwarz, Schwarz, Braun)
11	Widerstand (1kΩ)	8	 (Braun, Schwarz, Schwarz, Braun, Braun)
12	Widerstand (10kΩ)	4	 (Braun, Schwarz, Schwarz, Rot, Braun)
13	Widerstand (100kΩ)	4	 (Braun, Schwarz, Schwarz, Orange, Braun)
14	Widerstand (1MΩ)	1	 (Braun, Schwarz, Grün, Gold)

15	Widerstand (5.1 MΩ)	1	 <p>(Grün, Braun, Grün, Gold)</p>
16	Schalter	1	
17	Potentiometer (50k)	1	
18	Stromversorgungsmodul	1	
19	LCD1602	1	
20	Punktmatrix-Anzeige (8*8)	1	

21	7-Segment-Anzeige	2	
22	Gleichstrommotor	1	
23	LED (rot)	16	
24	LED (weiß)	2	
25	LED (grün)	2	
26	LED (gelb)	2	

27	NPN-Transistor (S8050)	2	
28	PNP-Transistor (S8550)	2	
29	Keramikkondensator 100nF	4	
30	Keramikkondensator 10nF	4	
31	Diodengleichrichter	4	
32	Steckplatine	1	

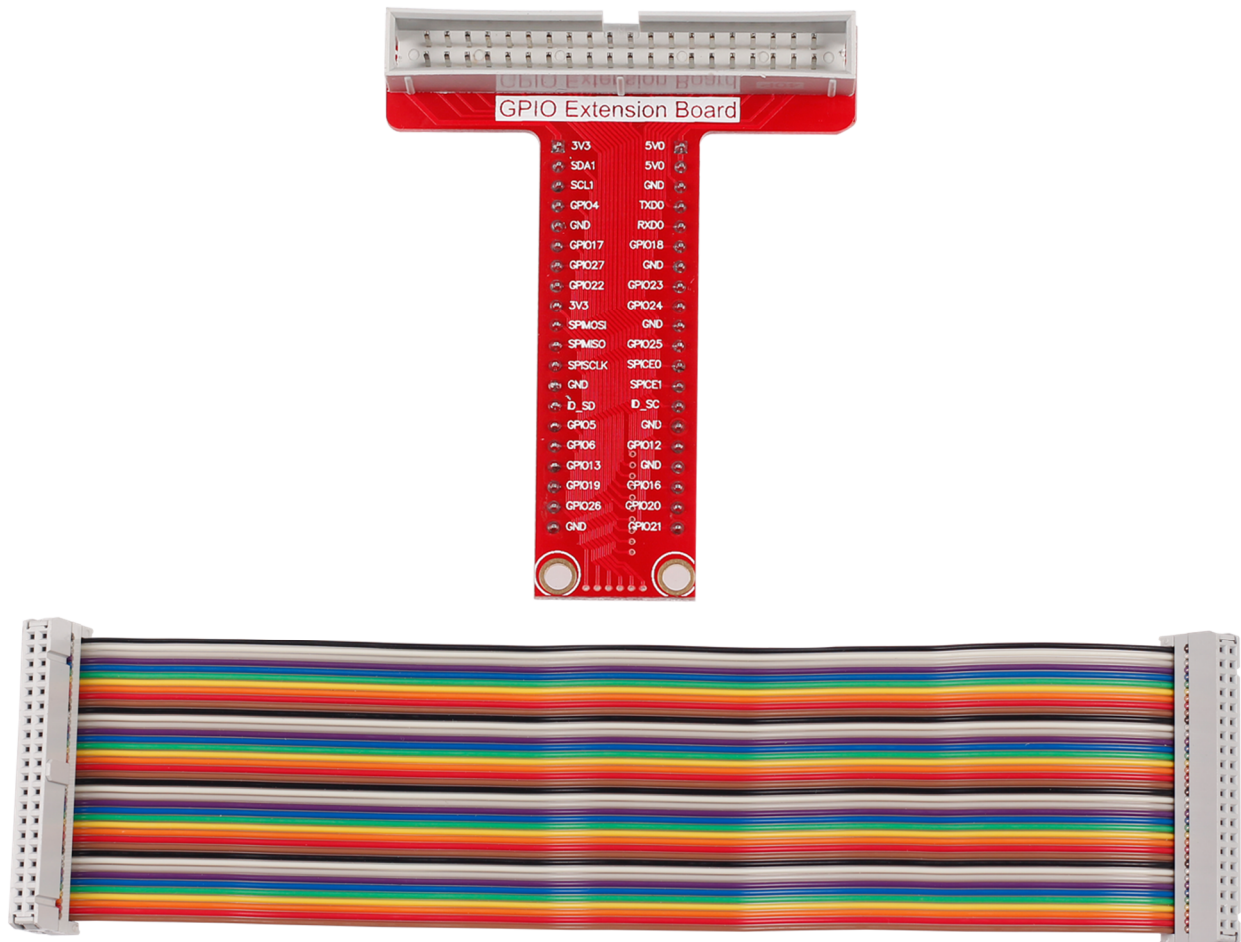
33	Male-to-Male Jumper Wire	65	
34	Female-to-Male Dupont Wire	20	
35	Buzzer(aktiv)	1	
36	Propeller	1	

Note:

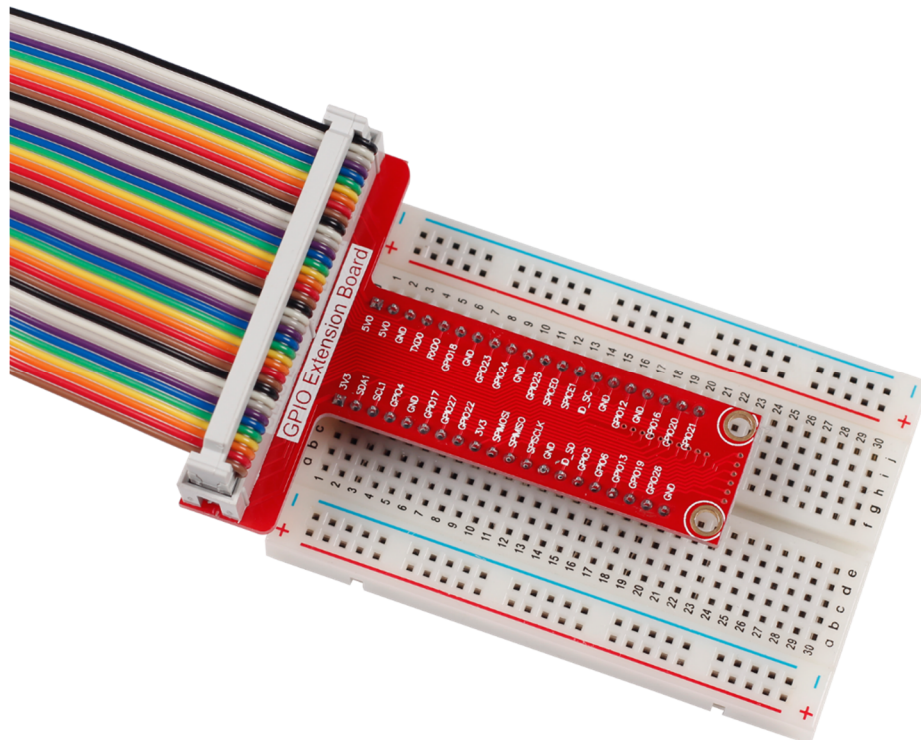
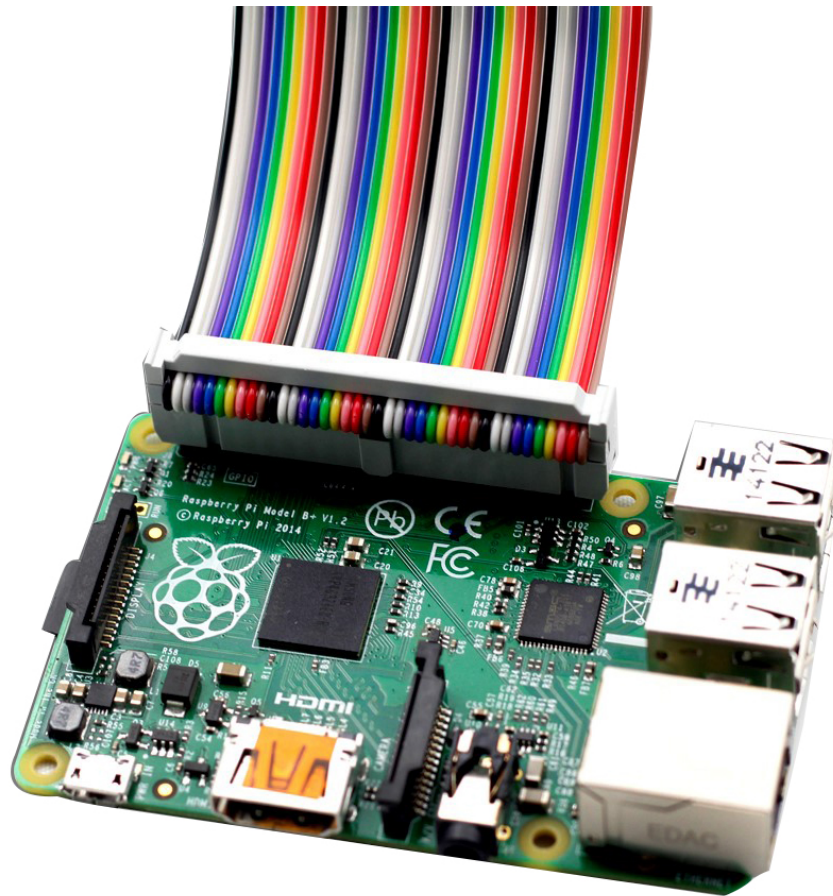
Nach dem Auspacken prüfen Sie bitte nach, ob die Menge der Komponenten richtig ist und ob sie in gutem Zustand sind oder nicht.

Hinweis

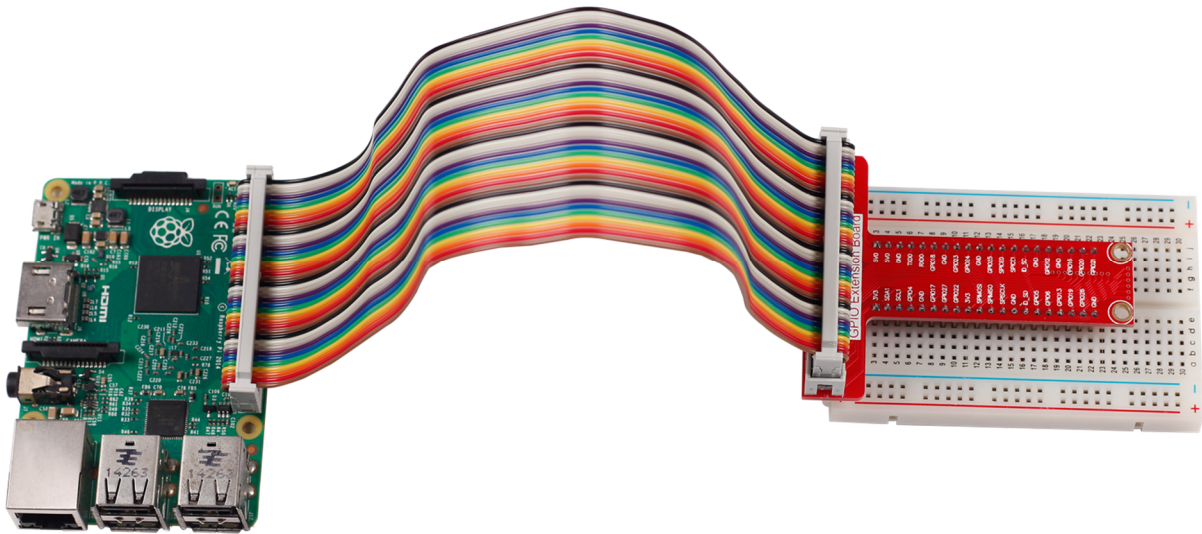
Über die GPIO Erweiterungsplatine können wir Pins des Raspberry Pi zur Steckplatine leicht herausführen, um den vom häufigen Einstecken&Ausziehen angerichteten GPIO-Schaden zu vermeiden. Das sind unsere 40-Pin GPIO Erweiterungsplatine und GPIO Kabel für Raspberry Pi Modell B+ , 2 Modell B und Pi 3.



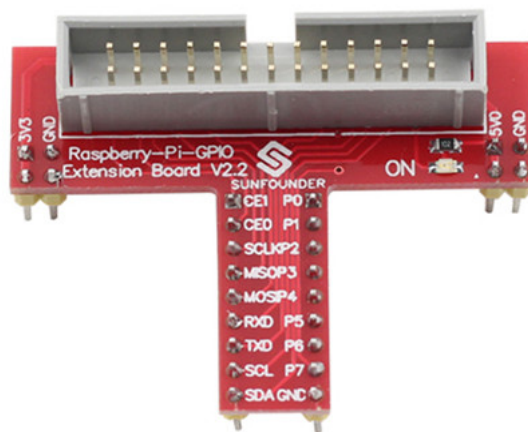
GPIO Kabel mit Raspberry Pi B+ (oder 2 Modell B und Pi 3) so verbinden:



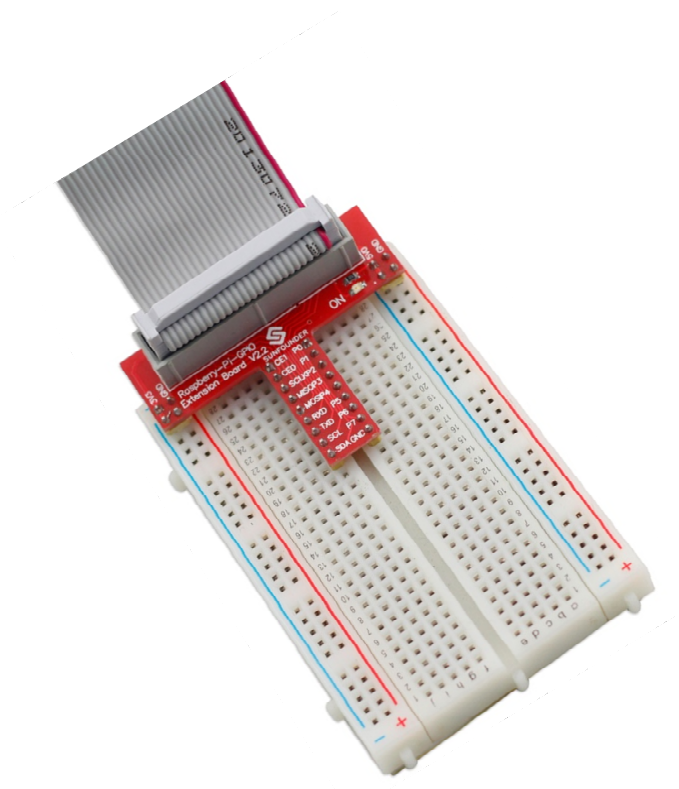
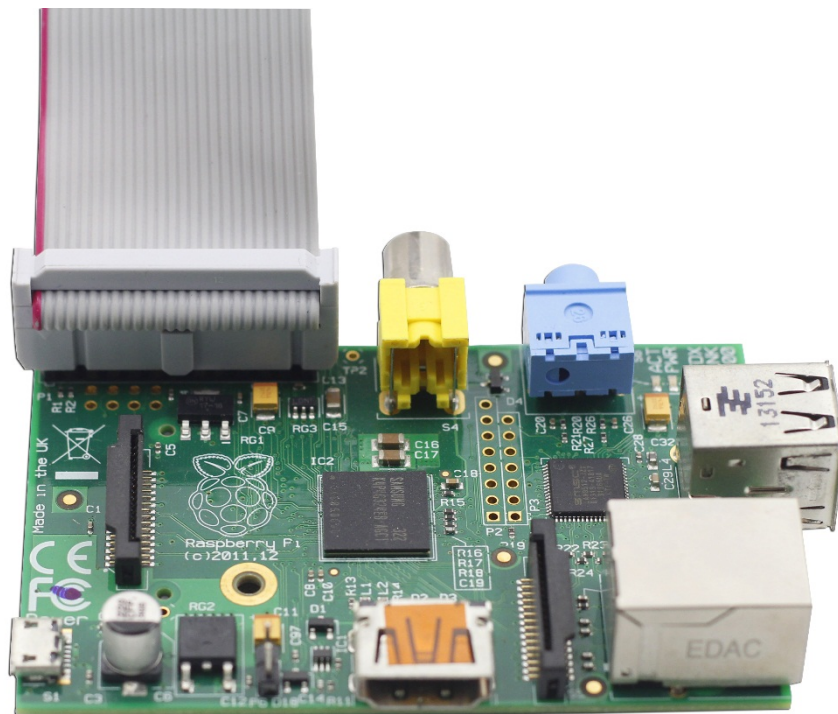
Nach Verbindung zeigt es sich wie folgt:



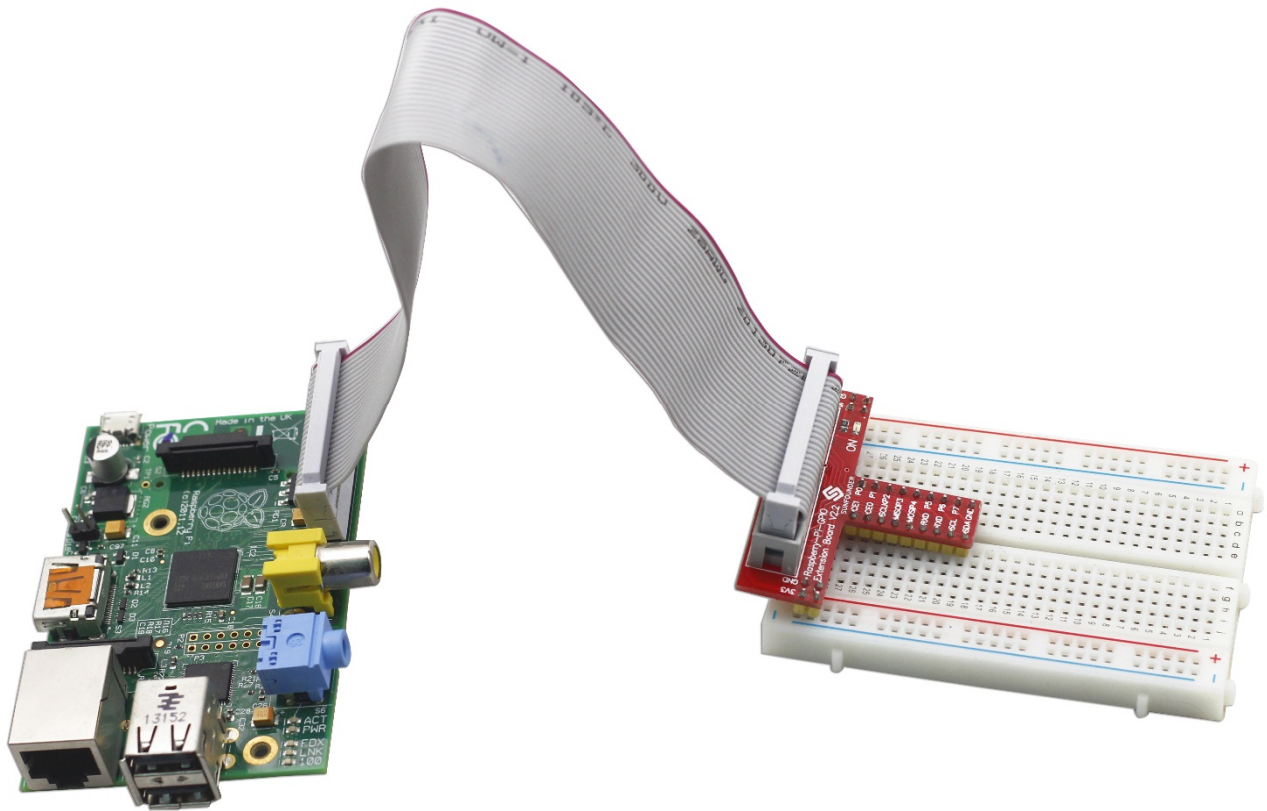
Das sind unsere 20-Pin GPIO Erweiterungsplatine und GPIO Kabel für Raspberry Pi B.



GPIO Kabel mit Raspberry Pi B so verbinden:



Nach Verbindung zeigt es sich wie folgt:



Nummerierung für Raspberry Pi Pins

Name	wiringPi Pin	BCM GPIO	BCM GPIO	wiringPi Pin	Name
GPIO Extention Board					
3.3V	-	-	3V3	5V0	5V
SDA	8	R1:0/R2:2	SDA1	5V0	5V
SCL	9	R1:1/R2:3	SCL1	GND	0V
GPIO7	7	4	GPIO4	TXD0	TXD
GND	-	-	GND	RXD0	RXD
GPIO0	0	17	GPIO17	GPIO18	GPIO1
GPIO2	2	R1:21/R2:27	GPIO27	GND	0V
GPIO3	3	22	GPIO22	GPIO23	GPIO4
3.3v	-	-	3V3	GPIO24	GPIO5
MOSI	12	10	SPIMOSI	GND	0V
MISO	13	9	SPIMISO	GPIO25	GPIO6
SCLK	14	11	SPISCLK	SPICE0	CE0
0V	-	-	GND	SPICE1	CE1
ID_SDA	30	0	ID_SD	ID_SC	ID_SCL
GPIO21	21	5	GPIO5	GND	0V
GPIO22	22	6	GPIO6	GPIO12	GPIO26
GPIO23	23	13	GPIO13	GND	0V
GPIO24	24	19	GPIO19	GPIO16	GPIO27
GPIO25	25	26	GPIO26	GPIO20	GPIO28
GND	-	-	GND	GPIO21	GPIO29

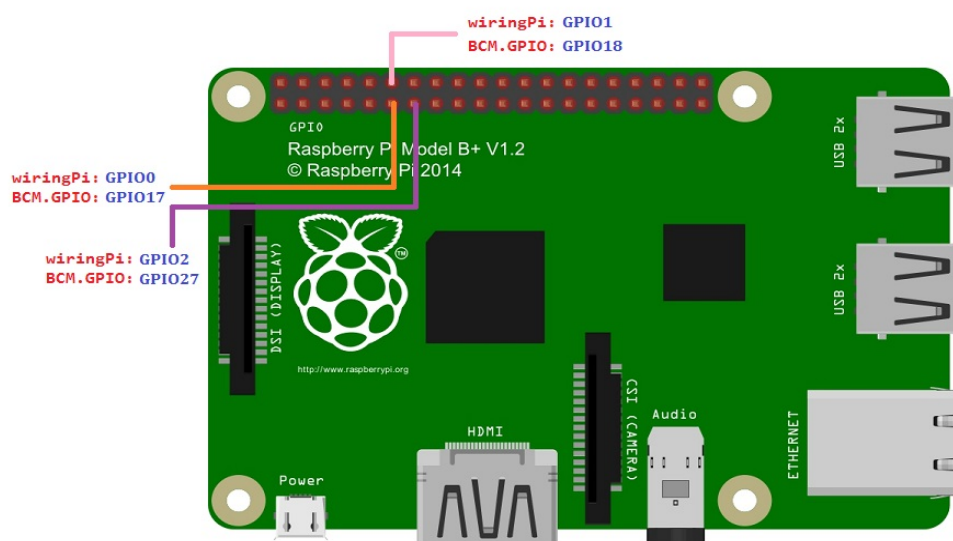
Name	wiringPi Pin	BCM GPIO	BCM GPIO	wiringPi Pin	Name
Raspberry-Pi-GPIO Extension Board V2.2					
CE1	11	7	CE1 P0	17	GPIO0
CE0	10	8	CE0 P1	18	GPIO1
SCLK	14	11	SCLK P2	21	GPIO2
MISO	13	9	MISO P3	22	GPIO3
MOSI	12	10	MOSI P4	23	GPIO4
RXD	16	15	RXD P5	24	GPIO5
TXD	15	14	TXD P6	25	GPIO6
SCL0	9	1	SCL P7	4	GPIO7
SDA0	8	0	SDA GND	0V	0V

Zur Zeit gibt es drei Methoden zur Nummerierung der Raspberry Pi Pins.

1. Nummerierung auf Basis vom physikalischen Standort der Pins.
2. Nummerierung bestimmt von der GPIO-Library wiringPi der C-Programmiersprache.
3. Nummerierung bestimmt vom BCM2835 SOC.

Wenn Sie Raspberry Pi GPIOs auf Basis wiringPi-Library in C-Sprache bedienen möchten, wählen Sie bitte die von wiringPi bestimmte Nummerierung. Aus dem Diagramm oben ist sichtbar, dass GPIO0 in wiringPi dem vom physikalischen Standort nummerierten Pin 11 entspricht, und GPIO30 dem Pin 27 entspricht.

Das Bild unten zeigt eingehend die physikalische Nummerierung von den Pin 11, 12, und 13:



Wenn Sie ein Benutzer der C-Sprache sind, installieren Sie bitte wiringPi.

Was ist wiringPi?

wiringPi ist eine Funktion der GPIO-Library und angewendet auf der Plattform Raspberry Pi. Es hält sich an GUN Lv3. Die Funktionen in wiringPi sind ähnlich wie die im wiring-System von Arduino. Sie machen die Benutzer vertraut mit Arduino und leichter wiringPi anzuwenden.

Wie wiringPi installieren?

Schritt 1: Quellcode von wiringPi bekommen

```
git clone git://git.drogon.net/wiringPi
```

Schritt 2: wiringPi installieren

```
cd wiringPi
git pull origin
./build
```


Nachdem Sie die Eingabetaste gedrückt haben, wird der Quellcode von wiringPi mit dem Skript *build* automatisch kompiliert werden und im entsprechenden Archiv von Raspberry Pi OS installiert werden.

Schritt 3: Installation prüfen

wiringPi schließt viele GPIO-Befehle ein, die Ihnen ermöglichen, verschiedene Schnittstellen auf Raspberry Pi anzusteuern. Ob die Installation erfolgt oder nicht, können Sie es nach folgenden Anweisungen prüfen.

gpio -v

```
pi@raspberrypi ~ $ gpio -v
gpio version: 2.26
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model B+, Revision: 1.2, Memory: 512MB, Maker: Sony
```

Falls die obige Message erschienen ist, erfolgt die Installation.

gpio readall

```
pi@raspberrypi ~ $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |      | 3.3v      |      |   | 1 | 2 |      |      | 5v      |      |      |
|  2  |  8  | SDA.1     | ALTO | 1 | 3 | 4 |      |      | 5V      |      |      |
|  3  |  9  | SCL.1     | ALTO | 1 | 5 | 6 |      |      | 0v      |      |      |
|  4  |  7  | GPIO. 7   | IN   | 0 | 7 | 8 | 1 | ALTO | TxD    | 15  | 14  |
|      |      | 0v        |      |   | 9 | 10 | 1 | ALTO | RxD    | 16  | 15  |
| 17  |  0  | GPIO. 0   | IN   | 0 | 11 | 12 | 0 | IN   | GPIO. 1 | 1  | 18  |
| 27  |  2  | GPIO. 2   | IN   | 0 | 13 | 14 |   |      | 0v      |      |      |
| 22  |  3  | GPIO. 3   | IN   | 0 | 15 | 16 | 0 | IN   | GPIO. 4 | 4  | 23  |
|      |      | 3.3v      |      |   | 17 | 18 | 0 | IN   | GPIO. 5 | 5  | 24  |
| 10  | 12  | MOSI      | ALTO | 0 | 19 | 20 |   |      | 0v      |      |      |
|  9  | 13  | MISO      | ALTO | 0 | 21 | 22 | 0 | IN   | GPIO. 6 | 6  | 25  |
| 11  | 14  | SCLK      | ALTO | 0 | 23 | 24 | 1 | OUT  | CE0     | 10  |  8  |
|      |      | 0v        |      |   | 25 | 26 | 1 | OUT  | CE1     | 11  |  7  |
|  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  |
| 26  | 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 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

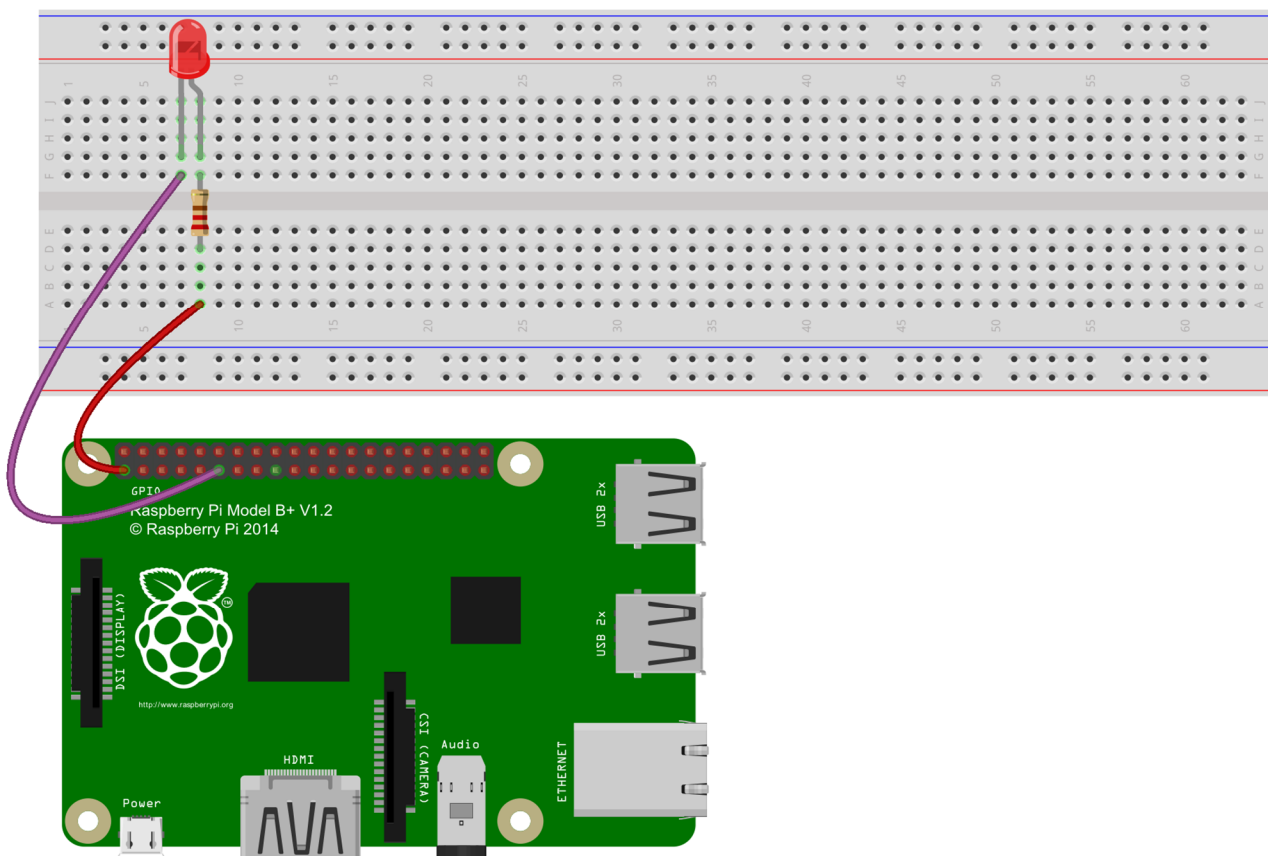
Wenn Sie ein Benutzer der Python-Sprache sind, können Sie GPIOs mit API, die von RPi.GPIO geboten ist, programmieren.

Was ist RPi.GPIO?

RPi.GPIO ist ein Modul, der GPIO-Kanäle des Raspberry Pi steuert. Das Paket bietet eine Klasse, GPIO auf Raspberry Pi anzusteuern. Für Beispiele und Dokumente, besuchen Sie bitte <http://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>

Raspbian OS image des Raspberry Pi installiert RPi.GPIO standardmäßig, deshalb können Sie es direkt anwenden.

Dann wenden Sie C- und Python-Sprache zur Programmierung an. Sie werden wiringPi- und RPi.GPIO-Modul anwenden lernen, um Raspberry Pi GPIOs anzusteuern. Nehmen wir blinkende LED zum Beispiel, Schaltung verbinden wie folgt:



fritzing

Note: Hier ist der Widerstand 220Ω. Wenn der Widerstand zu hoch ist, wird die LED zu matt sein und sogar nicht aufleuchten.

Für Benutzer der C-Sprache(auf Basis von der wiringPi-Library):

Schritt 1: Ein C-File namens *led.c* erstellen

```
touch led.c
```

Schritt 2: Nano oder andere Code-Bearbeitungswerkzeuge anwenden, um *led.c* zu öffnen, die folgenden Codes aufschreiben und speichern.

```
#include <wiringPi.h>
#include <stdio.h>

#define LedPin    0

int main(void)
{
    if(wiringPiSetup() == -1){//when the wiringPi initialization fails,
                               print message to the screen.
        printf("setup wiringPi failed !");
        return 1;
    }

    pinMode(LedPin, OUTPUT);

    while(1){
        digitalWrite(LedPin, LOW);    //led on
        printf("led on...\n");
        delay(500);
        digitalWrite(LedPin, HIGH);   //led off
        printf("...led off\n");
        delay(500);
    }
    return 0;
}
```

Schritt 3: Kompilieren

```
gcc led.c -o led -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./led
```


Jetzt sollten Sie die LED blinken gesehen haben. Die Tasten Ctrl+C drücken, um das Programm zu beenden.

Für Benutzer der Python-Sprache(auf Basis vom RPi.GPIO Python-Modul):

Schritt 1: Ein Python-File namens *led.py* erstellen

```
touch led.py
```

Schritt 2: Nano oder andere Code-Bearbeitungswerkzeuge anwenden, um *led.py* zu öffnen, die folgenden Codes aufschreiben und speichern.

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO
```

```
import time
```

```
LedPin = 11      # pin11
```

```
GPIO.setmode(GPIO.BOARD)      # Numbers GPIOs by physical location
```

```
GPIO.setup(LedPin, GPIO.OUT)  # Set LedPin's mode as output
```

```
GPIO.output(LedPin, GPIO.HIGH) # Set LedPin as high(+3.3V) to turn off  
                                the led
```

```
try:
```

```
    while True:
```

```
        print '...led on'
```

```
        GPIO.output(LedPin, GPIO.LOW)  # led on
```

```
        time.sleep(0.5)
```

```
        print 'led off...'
```

```
        GPIO.output(LedPin, GPIO.HIGH) # led off
```

```
        time.sleep(0.5)
```

```
except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the following code  
                           will be executed.
```

```
    GPIO.output(LedPin, GPIO.HIGH)      # led off
```

```
    GPIO.cleanup()                     # Release resource
```

Schritt 3: Ausführen

```
sudo python led.py
```

Jetzt sollten Sie die LED blinken gesehen haben. Die Tasten Ctrl+C drücken, um das Programm zu beenden.

Source Code bekommen

Methode 1: von Github

Directory auf `/home/pi` umstellen

```
cd /home/pi/
```

Repository klonen von github(C Code und Python Code)

```
git clone  
https://github.com/sunfounder/Sunfounder_SuperKit_C_code_for_RaspberryPi.git
```

```
git clone  
https://github.com/sunfounder/Sunfounder_SuperKit_Python_code_for_RaspberryPi  
.git
```

Methode 2: von SunFounder Website

Source Code runterladen von unserer Website. www.sunfounder.com.

Auf **LEARN->Get Tutorials** klicken.

Super Kit v2.0 for Raspberry Pi finden und klicken.

Super_Kit_v2.0_for_Raspberry_Pi.zip runterladen.

Entzippen, Files darin in den Pfad `/home/pi/` bewegen.

Lektion 1 Blinkende LED

Einführung

In dieser Lektion werden wir den Raspberry Pi programmieren lernen, um die LED blinken zu lassen. Mit einer LED können Sie nach eigenen Wünschen viele Zauber machen. Nun fangen wir an und genießen den DIY-Spaß.

Komponenten

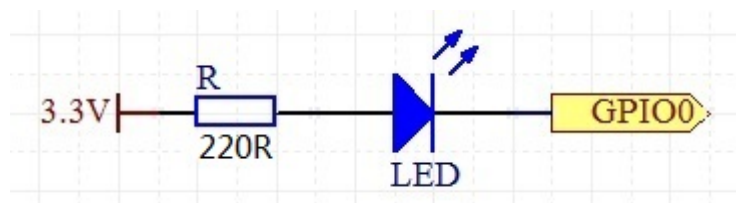
- 1* Raspberry Pi
- 1* Steckplatine
- 1* LED
- 1* Widerstand (220Ω)
- Jumper Wire

Versuchsprinzip

Halbleiter - Leuchtdioden sind ein Bauelement, das mittels PN-Übergang die elektrische Energie in die Lichtenergie umwandeln kann. Wegen verschiedener Wellenlängen können sie zur infraroten Leuchtdiode und sichtbaren Leuchtdiode kategorisiert werden. Die letztere ist üblicherweise als Licht emittierende Diode (LED) bekannt.

Wenn einer LED eine Vorwärtsspannung 2V-3V gegeben wird, blinkt die LED nur wenn der Vorwärtsstrom durch sie fließt. Es gibt gewöhnlich rote, gelbe, grüne, blaue und mit verschiedenen Spannungen farbenwechselnde LEDs. Wegen der niedrigen Betriebsspannung, des niedrigen Stromes, der Lumineszenzstabilität und der geringen Größe sind LEDs häufig benutzt.

LEDs sind auch Dioden. Daher abhängig von ihren Typen, haben sie einen Spannungsabfall, der normalerweise von 1V zu 3V wechselt. Zusätzlich strahlen LEDs normalerweise Licht aus wenn versorgt mit Strom 5mA–30mA, im allgemeinen wendet man 10mA–20mA an. Deshalb wenn eine LED angewendet ist, ist es nötig, einen Strombegrenzenden Widerstand zu verbinden, für Schutz vor Verbrennung.



Für Benutzer der Python-Sprache:

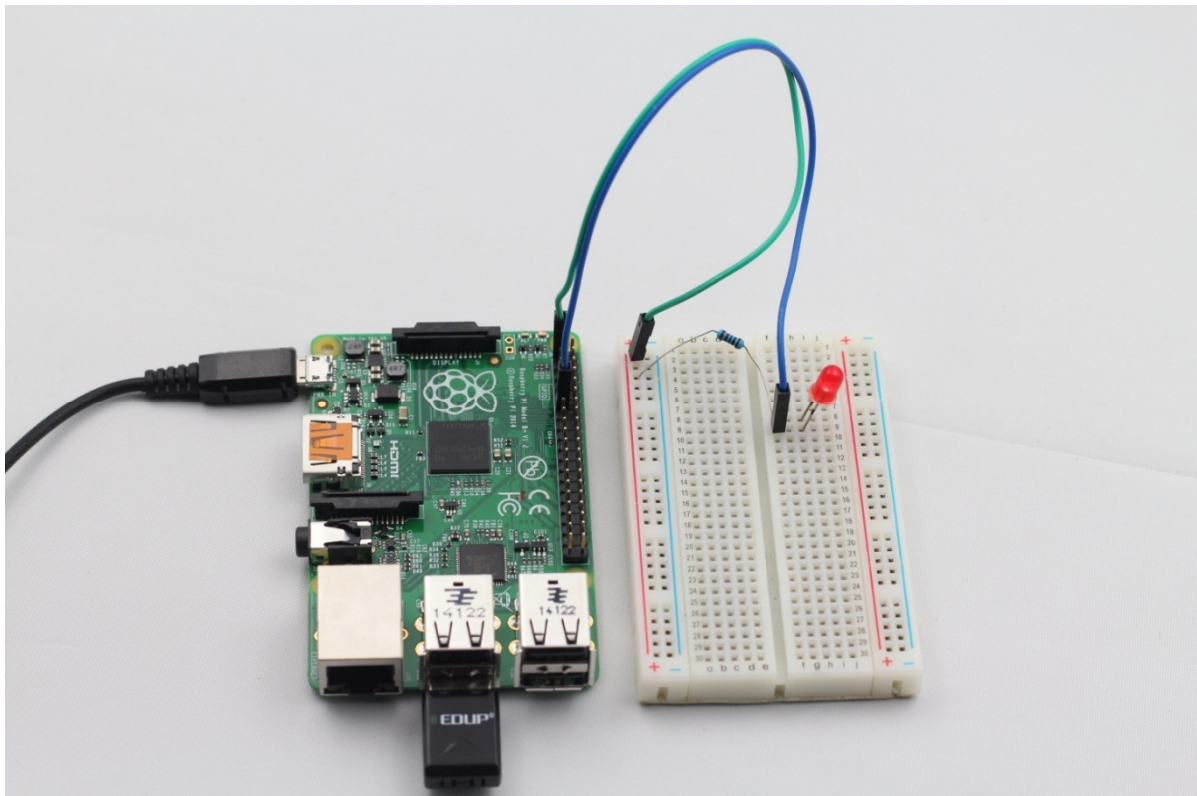
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 01_led.py
```

Jetzt können Sie die LED blinken sehen.



Erweiterung

Wenn Sie die LED schneller blinken möchten, ändern Sie gleich die Verzögerungszeit. Zum Beispiel, die Zeit auf `delay(200)` umstellen, den Code wiederkompilieren und den laufen lassen, dann werden Sie die LED schneller blinken sehen.

Tipps: Für **technische Fragen**, erstellen Sie bitte ein Thema im  **FORUM** auf www.sunfounder.com.

Für **nichttechnische Fragen**, bitte an  service@sunfounder.com mailen.

Lektion 2 LED ansteuern mit Druckschalter

Einführung

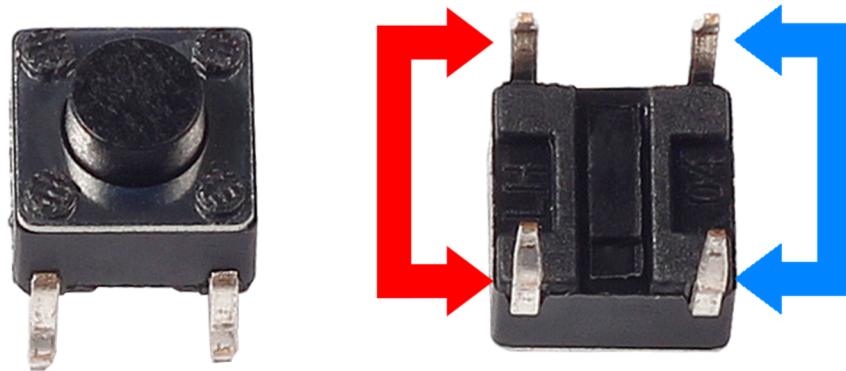
In dieser Lektion werden wir lernen, wie mit dem Druckschalter eine LED an- oder auszumachen.

Komponenten

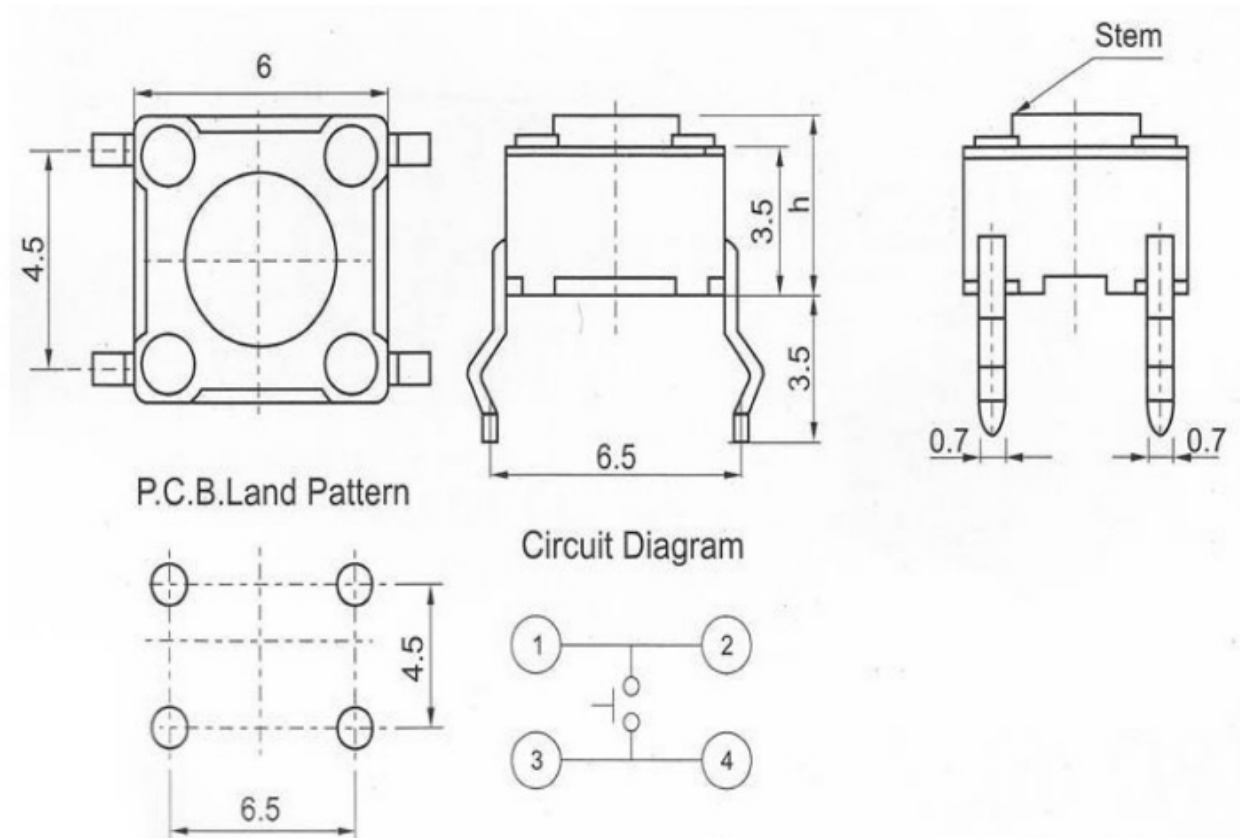
- 1* Raspberry Pi
- 1* Steckplatine
- 1* LED
- 1* Druckschalter
- 1* Widerstand (220Ω)
- Jumper Wire

Versuchsprinzip

Der Druckschalter ist eine übliche Komponente und verwendet, elektronische Geräte anzusteuern. Obwohl Druckschalter in einer Vielzahl von Größen und Formen kommen, ist der in diesem Versuch eingesetzte ein 6 mm Mini-Schalter, dargestellt wie in den folgenden Bildern. Die durch die gleichfarbigen Pfeile gezeigten Pins sind miteinander verbunden.

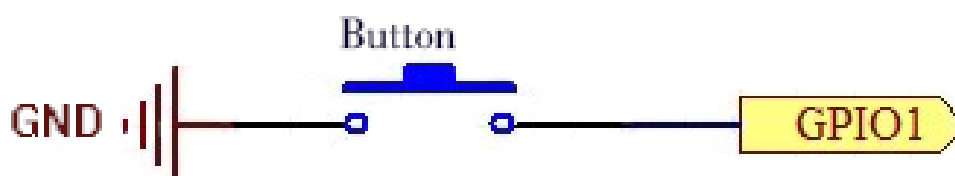


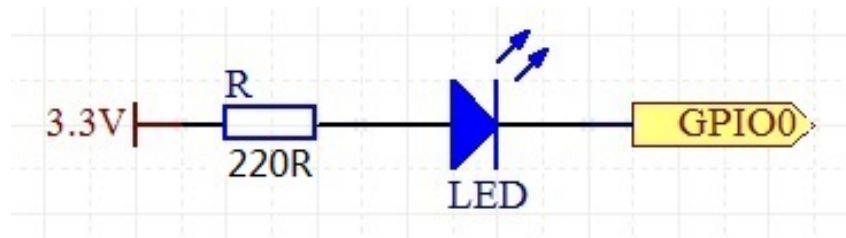
Wenn der Schalter gedrückt ist, werden die durch blaue Pfeile gezeigten Pins und die durch rote Pfeile Gezeigten miteinander verbunden werden. Dann ist der Stromkreis geschlossen wie dargestellt in den folgenden Diagrammen:



Im Allgemeinen wird der Druckschalter direkt in einer LED-Schaltung angeschlossen, um die LED an- oder auszumachen. Diese Verbindung ist relativ einfach. Aber manchmal wird die LED automatisch aufleuchten, ohne dass der Schalter gedrückt ist, was von verschiedenen Störungen verursacht ist. Um diese externen Störungen zu vermeiden, werden wir einen Pull-Down Widerstand anschließen, das heißt, einen 1K-10 k Widerstand zwischen Schalter und GND anschließen. Er kann Störungen aufwenden, wenn der Druckschalter aus ist.

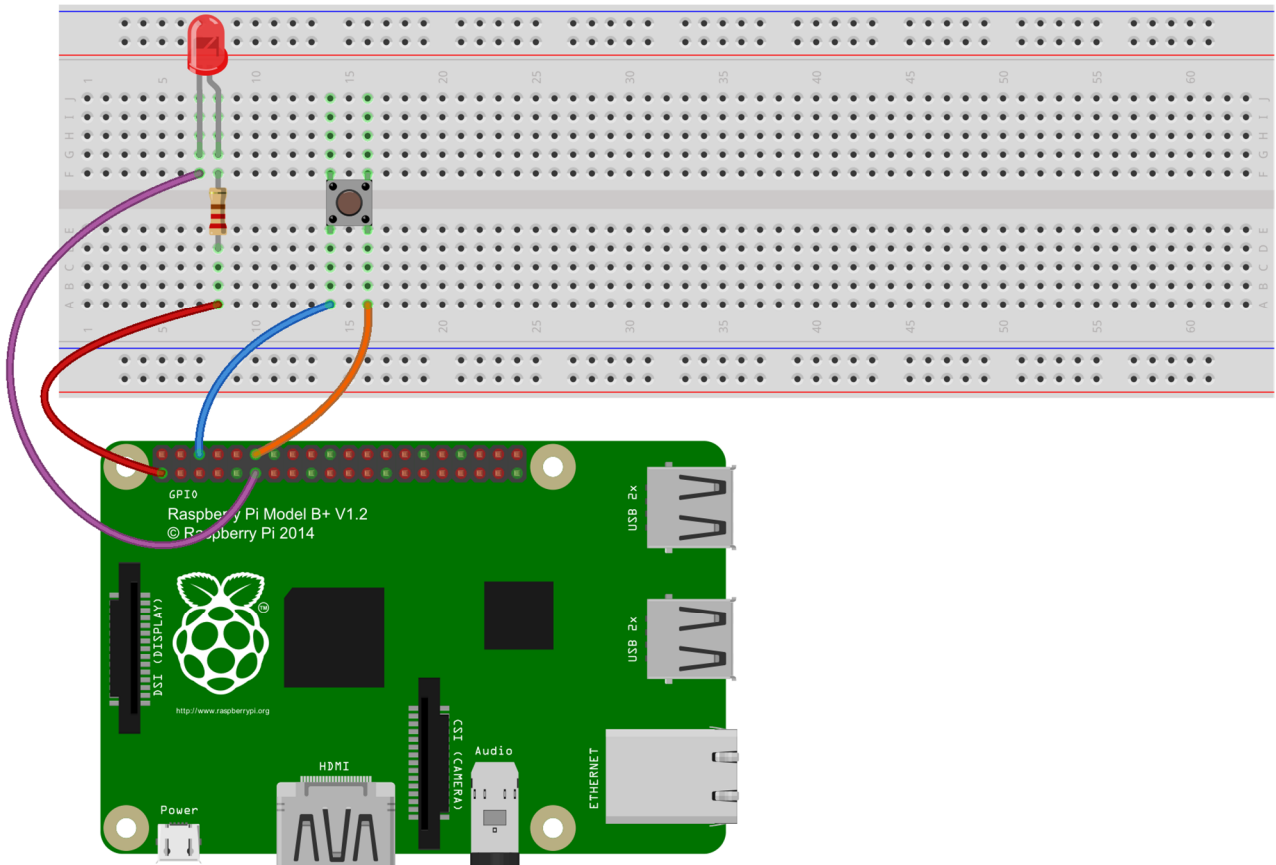
Einen Schließer-Druckschalter (normal open) als den Input des Raspberry Pi anwenden. Wenn der Schalter gedrückt wird, wird der mit dem Schalter verbundene GPIO zum niedrigen Pegel (0V) werden. Durch Programmierung können wir den Zustand von diesem GPIO überwachen. Wenn dieser GPIO zum niedrigen Pegel (0V) wird, das heißt, dass der Schalter gedrückt gewesen ist. Sie können den entsprechenden Code laufen lassen when der Schalter gedrückt wird, und dann wird die LED aufleuchten.





Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/02_BtnAndLed/
```

Schritt 3: Kompilieren

```
gcc BtnAndLed.c -o BtnAndLed -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./BtnAndLed
```


Für Benutzer der Python-Sprache:

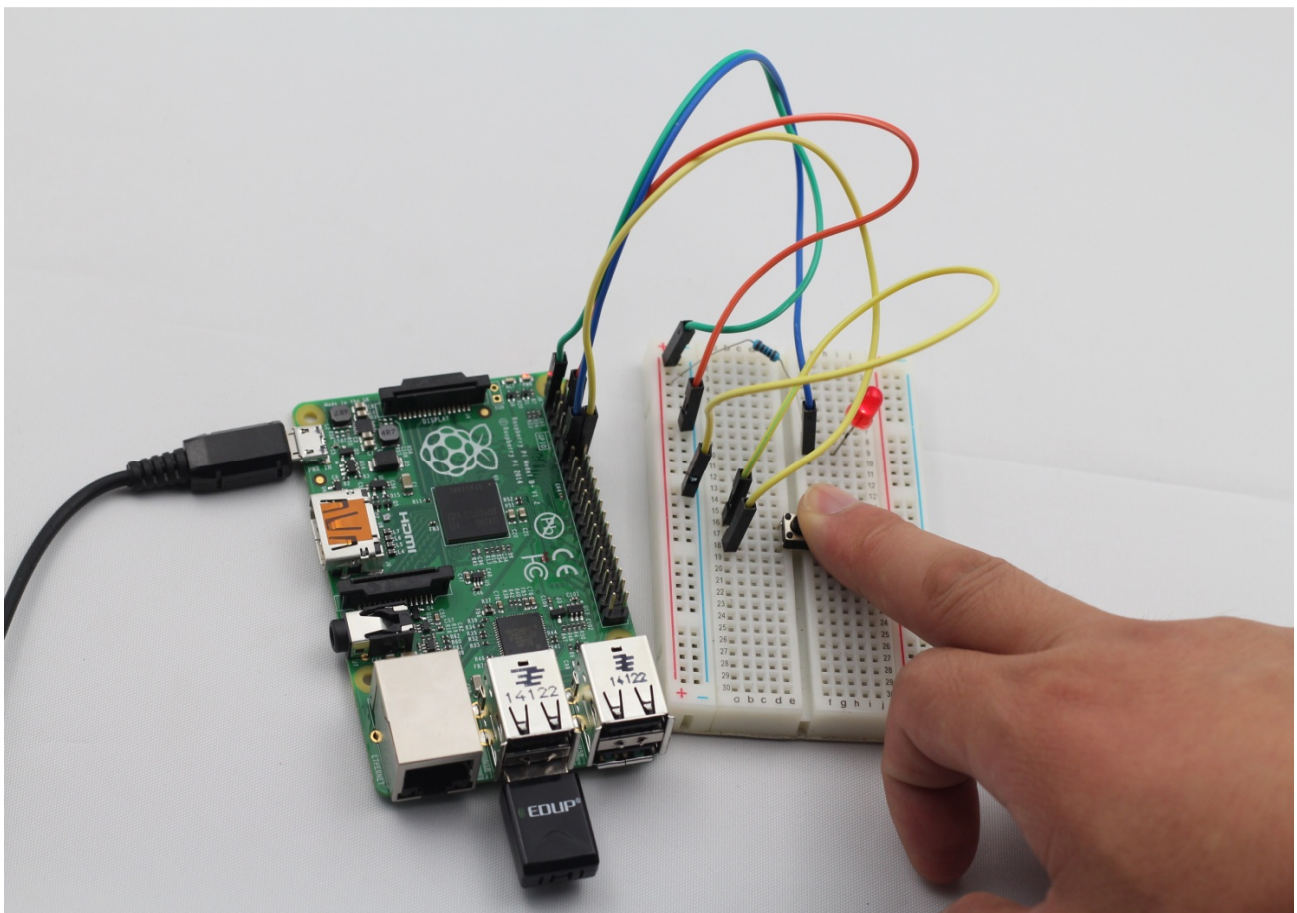
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 02_btnAndLed.py
```

Den Schalter drücken, die LED leuchtet auf; Den Schalter loslassen, die LED geht aus. Gleichzeitig wird der Zustand der LED auf dem Bildschirm gezeigt.



Lektion 3 Lauflicht

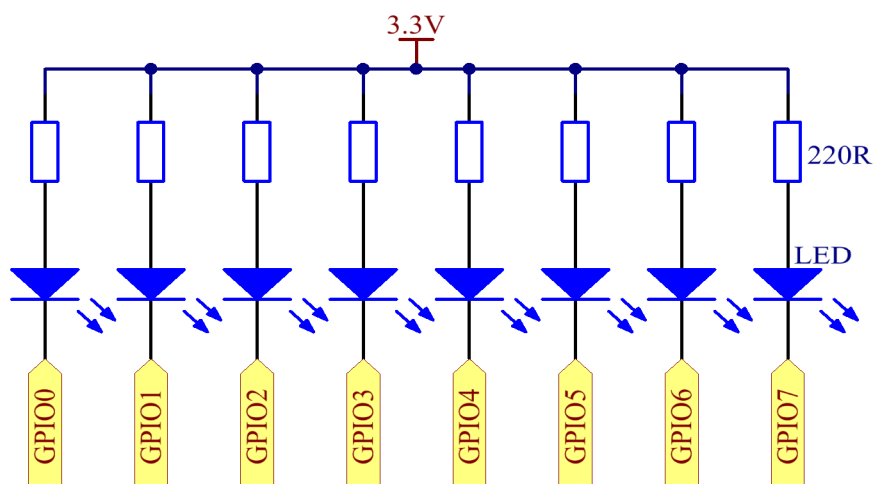
Einführung

In dieser Lektion werden wir lernen, wie nach eigenen Wünschen auf Raspberry Pi acht LEDs mit verschiedenen Effekten blinken zu lassen.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 8* LED
- 8* Widerstand (220Ω)
- Jumper Wire

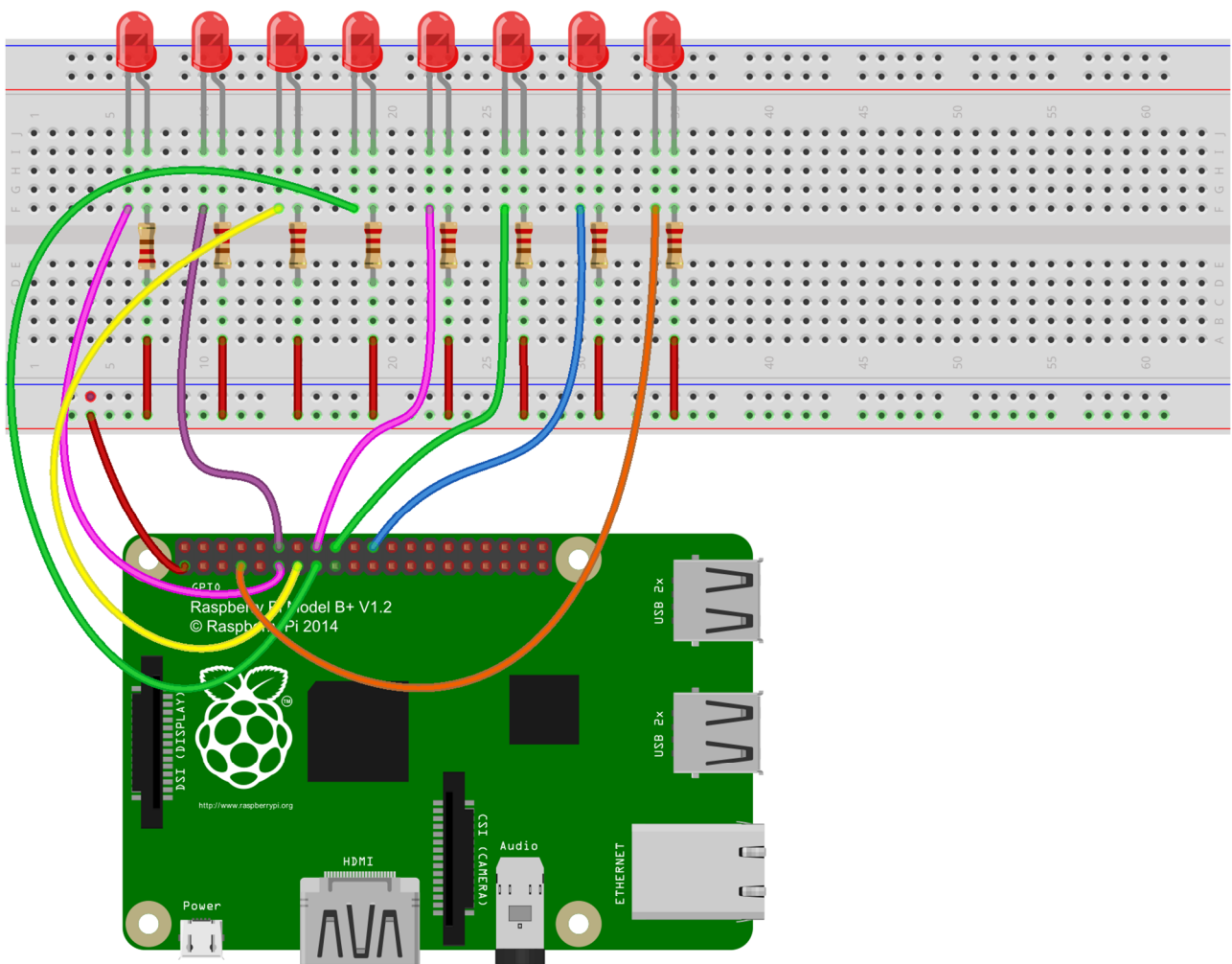
Versuchsprinzip



Durch Programmierung GPIO0 - GPIO7 nacheinander in niedrigen Pegel setzen, und dann wird LED0 - LED7 nacheinander aufleuchten. Sie können acht LEDs in verschiedenen Effekten blinken lassen, durch Steuerung von ihrer Verzögerungszeit und Aufleuchtreihe.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/03_8Led/
```

Schritt 3: Kompilieren

```
gcc 8Led.c -o 8Led -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./8Led
```

Für Benutzer der Python-Sprache:

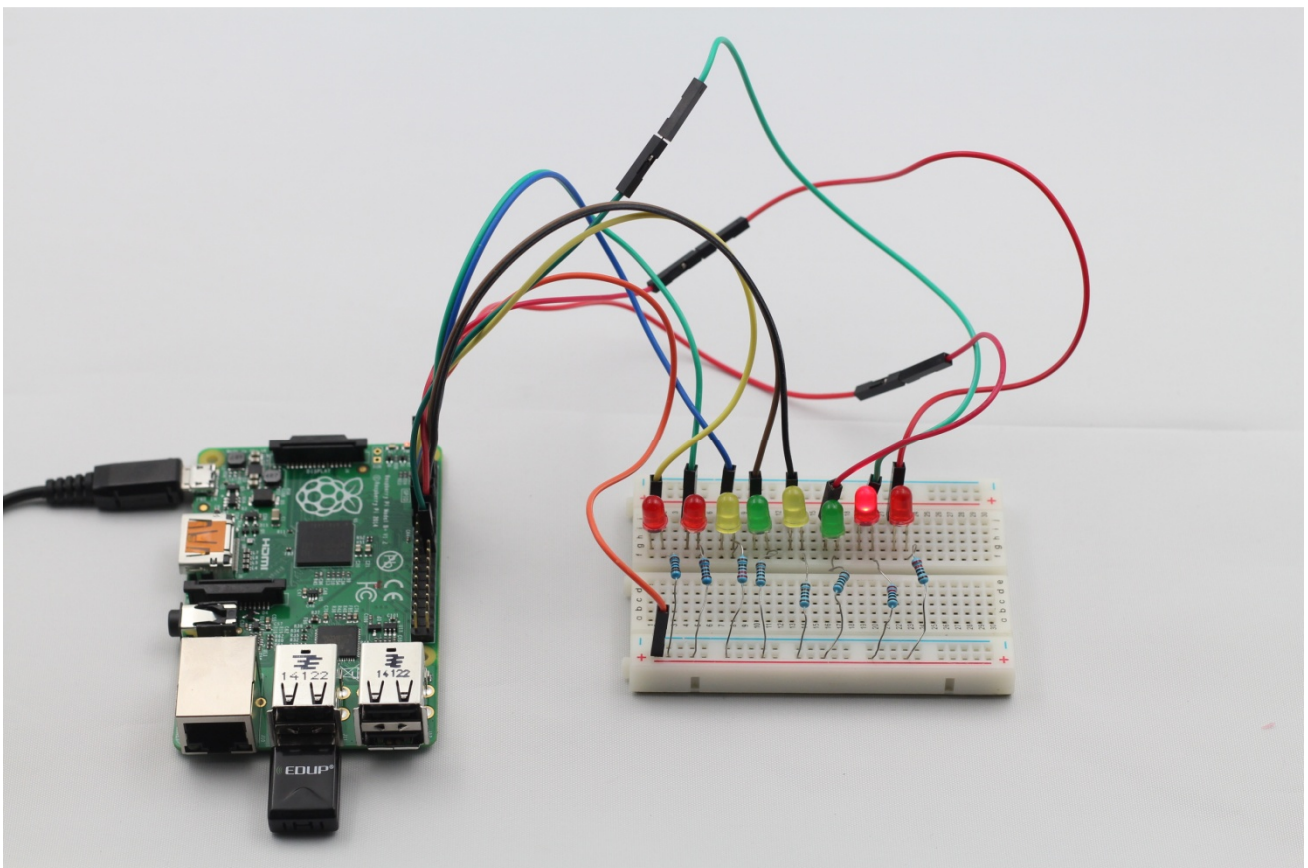
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 03_8Led.py
```

Dann sehen Sie acht LEDs nacheinander aufleuchten und abdunkeln, zyklisch, genau wie fließendes Wasser.



Erweiterung

Sie können die Blinkeffekte der LEDs auch in einem Array schreiben. Wenn Sie einen von diesen Effekten anwenden möchten, können Sie es in der `main()` -Function direkt aufrufen.

Lektion 4 Atmungslicht

Einführung

In dieser Lektion, werden wir mit PWM die Leuchtdichte einer LED allmählich zu- und abnehmen, wie die Atmung. Dazu geben wir ihm einen zauberhaften Namen -Atmungslicht.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* LED
- 1* Widerstand (220Ω)
- Jumper Wire

Versuchsprinzip

Bevor wir über PWM sprechen, werfen wir erst einen Blick auf ihre Anwendung. Die PWM ist beim Ansteuern der Motordrehzahl, des Lenkwinkels, der Lichtintensität und beim Signalausgang erfolgreich angewendet gewesen. Zum Beispiel, wenn die PWM in einer Hupe angewendet ist, wird sie einen Schall erzeugen. Nachdem wir über ihre Sonderfunktionen gewusst haben, lassen wir jetzt herausfinden, was eigentlich die PWM ist.

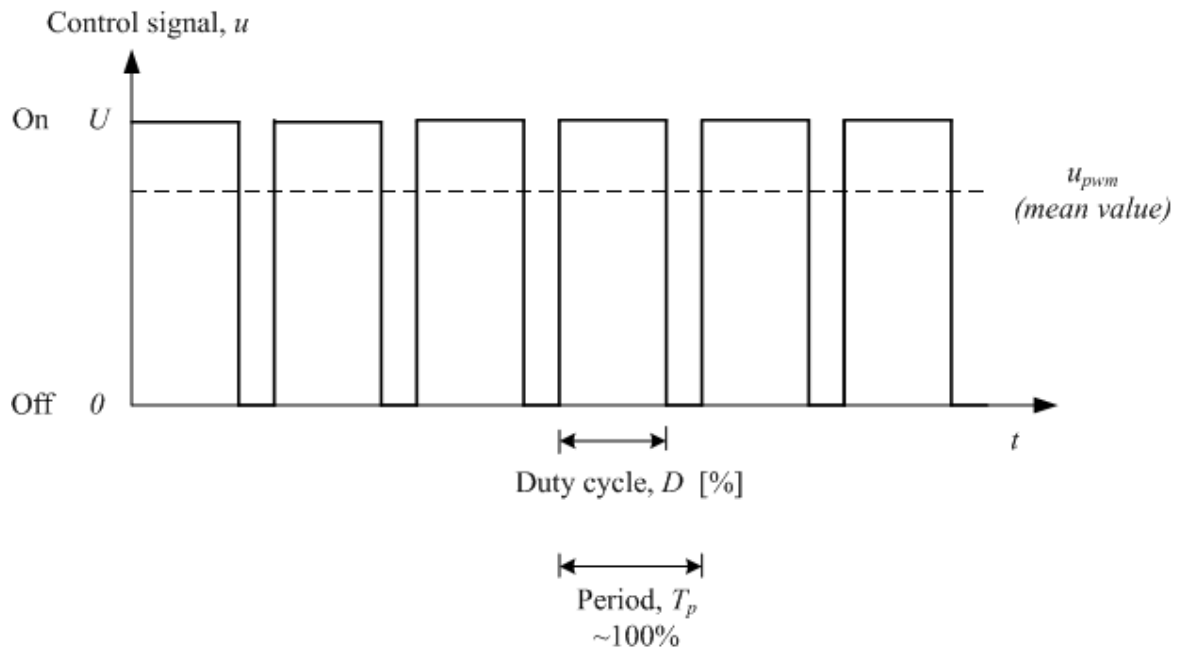
Pulsweitenmodulation, kurz PWM, ist eine Technik für Bekommen analoger Ergebnisse mit digitalen Durchschnitten. Digitale Steuerung ist angewendet, eine Rechteckwelle zu erzeugen, ein Signal schaltet zwischen EIN und AUS. Dieses EIN&AUS Schema kann Spannungen zwischen EIN (voll 5 V) und AUS (0 V) dadurch simulieren, die Portion der Zeit, die das Signal auf EIN und AUS verwendet, zu ändern. Die Dauer von EIN ist bezeichnet als Pulsweite. Um verschiedene analoge Werte zu bekommen, ändern oder modulieren wir die Pulsweite. Wenn Sie dieses EIN/AUS -Schema schnell genug wiederholen, mit einer LED zum Beispiel, ergibt sich es, als ob das Signal eine stabile Spannung zwischen 0V und 5V wäre, die die Leuchtdichte der LED steuert. (Siehe die Beschreibung der PWM auf der offiziellen Website von Arduino).

Tastgrad

Ein Tastgrad ist der Prozentsatz einer Periode, in der ein Signal aktiv ist. Eine Periode ist die Zeit, die das Signal verwendet, um einen EIN- und AUS-Zyklus zu erledigen. Als ein Formel kann der Tastgrad ausgedrückt sein wie:

$$D = \frac{T}{P} \times 100\%$$

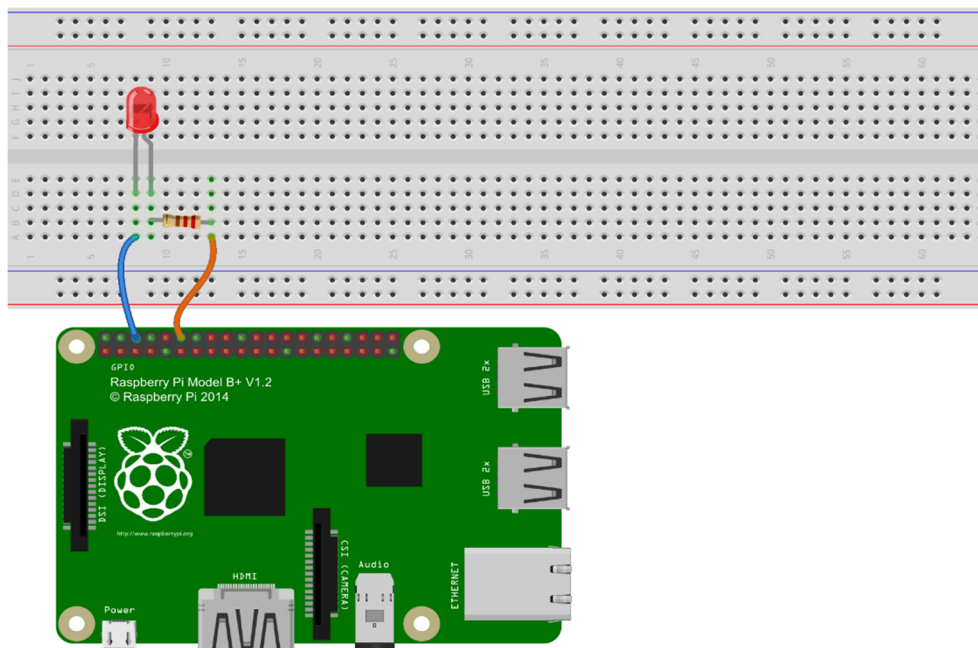
D ist der Tastgrad, T ist die Zeit, in der das Signal aktiv ist und P ist die totale Periode des Signals. Deshalb bedeutet ein 60%-Tastgrad, dass das Signal 60% der Zeit eingeschaltet ist, 40% der Zeit ausgeschaltet ist. Die EIN-Zeit für einen 60%-Tastgrad kann ein Bruchteil einer Sekunde, eines Tags, oder sogar einer Woche sein, abhängig von der Länge der Periode.



Hier wenden wir diese Technologie an, um die LED langsam aufleuchten und abdunkeln zu lassen, damit es wie der Atem aussieht.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/04_PwmLed
```

Schritt 3: Kompilieren

```
gcc PwmLed.c -o PwmLed -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./PwmLed
```

Für Benutzer der Python-Sprache:

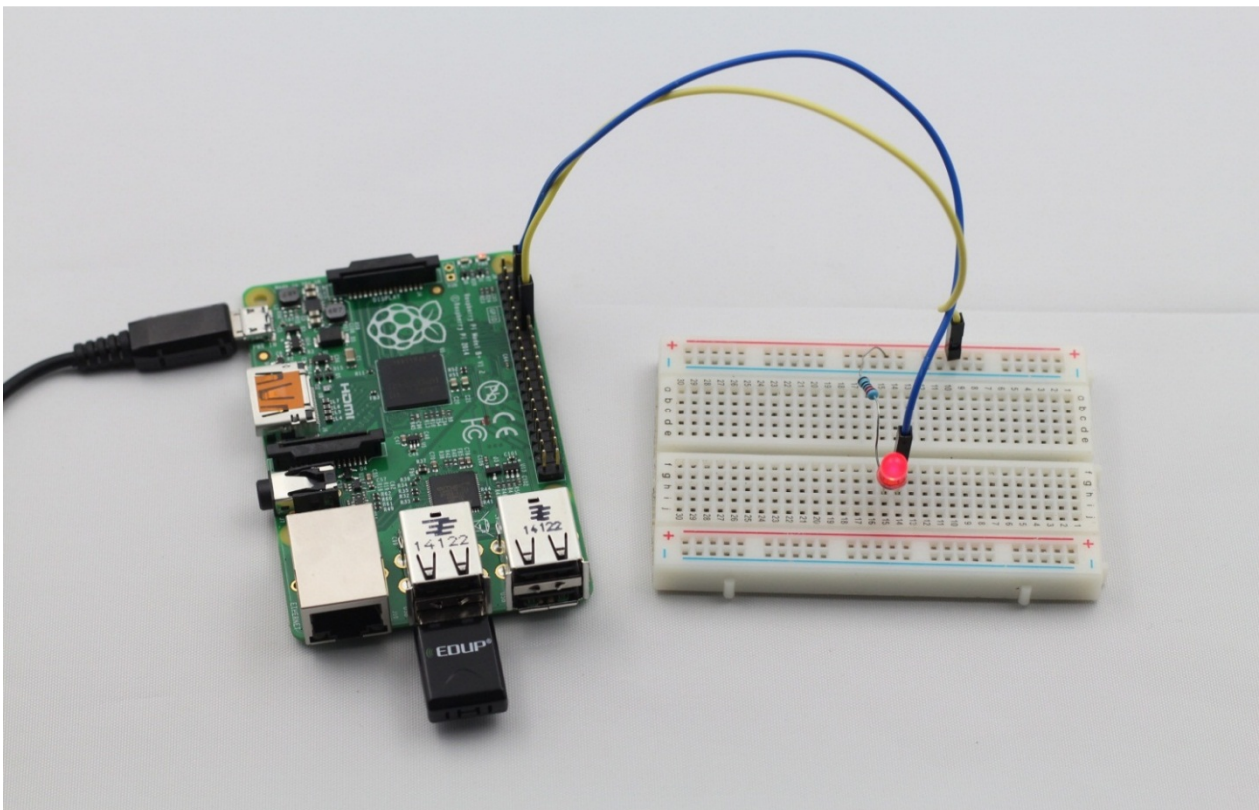
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 04_pwmLed.py
```

Eingabetaste drücken, dann werden Sie einen allmählichen Leuchtdichtenwandel der LED sehen.



Lektion 5 RGB LED

Einführung

In dieser Lektion werden wir über PWM eine RGB LED verschiedene Farben blinken lassen.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* RGB LED
- 3* Widerstand (220Ω)
- Jumper Wire

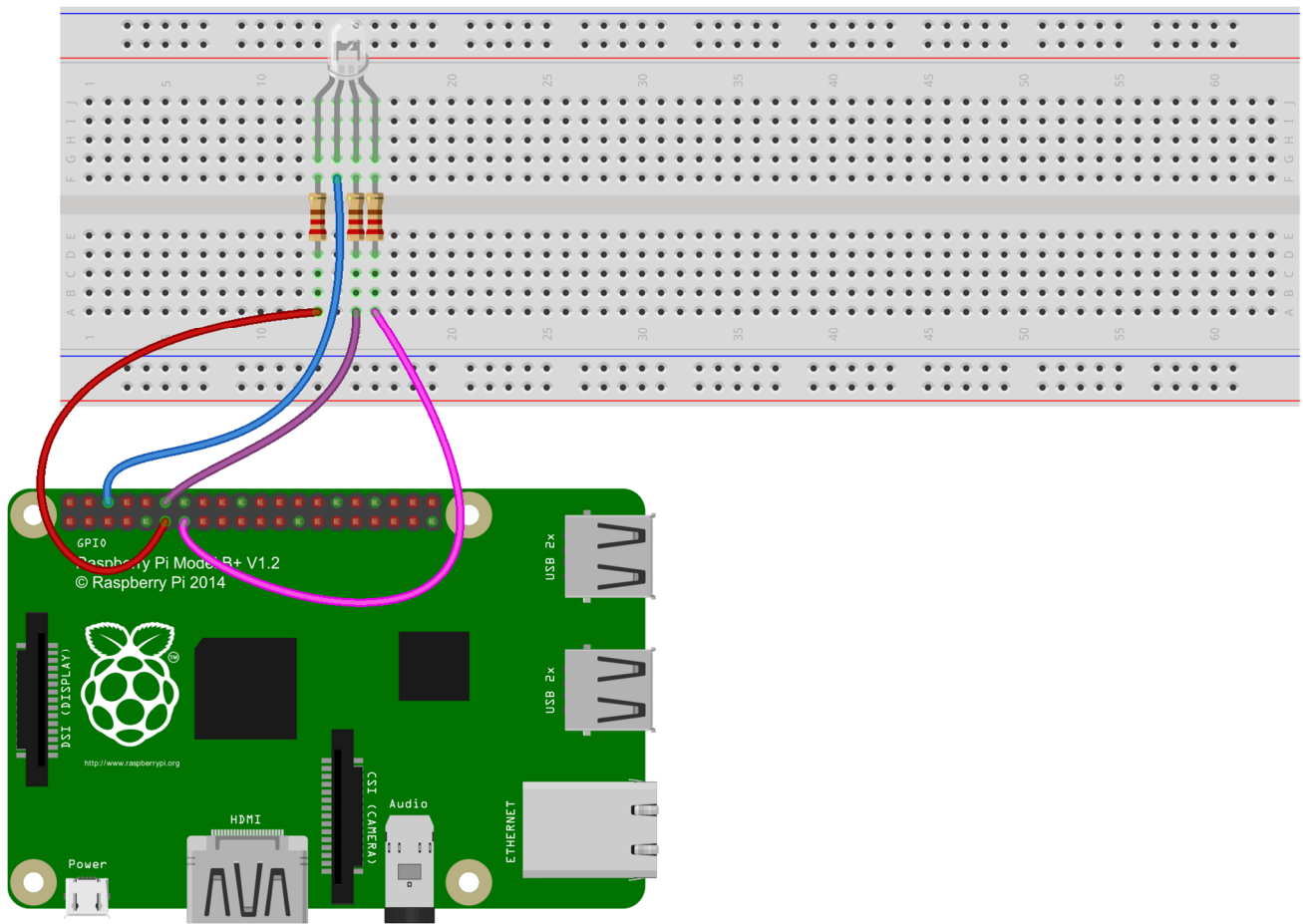
Versuchsprinzip

RGB LEDs emittieren Licht in verschiedenen Farben. In ihrem transparenten oder halbtransparenten Kunststoffgehäuse ist sie ausgestattet mit drei LEDs, eine rote, eine grüne und eine blaue, und hat vier Pins. Mit verschiedenen Leuchtdichten mischen sich die drei Primärfarben zu verschiedenen Farben. Die Leuchtdichte der LED kann mit PWM geregelt werden. Der Raspberry Pi hat nur einen Kanal für den Output der Hardware PWM, aber er braucht drei Kanäle, um die RGB LED anzusteuern, das bedeutet, es ist zu schwer, mit der Hardware PWM des Raspberry Pi die RGB LED anzusteuern. Glücklicherweise simuliert jedoch die `softPwm`-Library durch Programmierung die PWM (`softPwm`). Deshalb brauchen wir die Header-Datei `#include < softPwm.h >` einzufügen (für Benutzer der C-Sprache), und dann rufen die API auf. Der Mehrkanäleoutput der PWM ist dann einfach verwirklicht.

Man kategorisiert RGB LEDs als die „mit gemeinsamer Anode“ und die „mit gemeinsamer Kathode“. In diesem Versuch ist die letztere angewendet.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/05_RGB/
```

Schritt 3: Kompilieren

```
gcc rgb.c -o rgb -lwiringPi -lpthread
```

Schritt 4: Ausführen

```
sudo ./rgb
```

Für Benutzer der Python-Sprache:

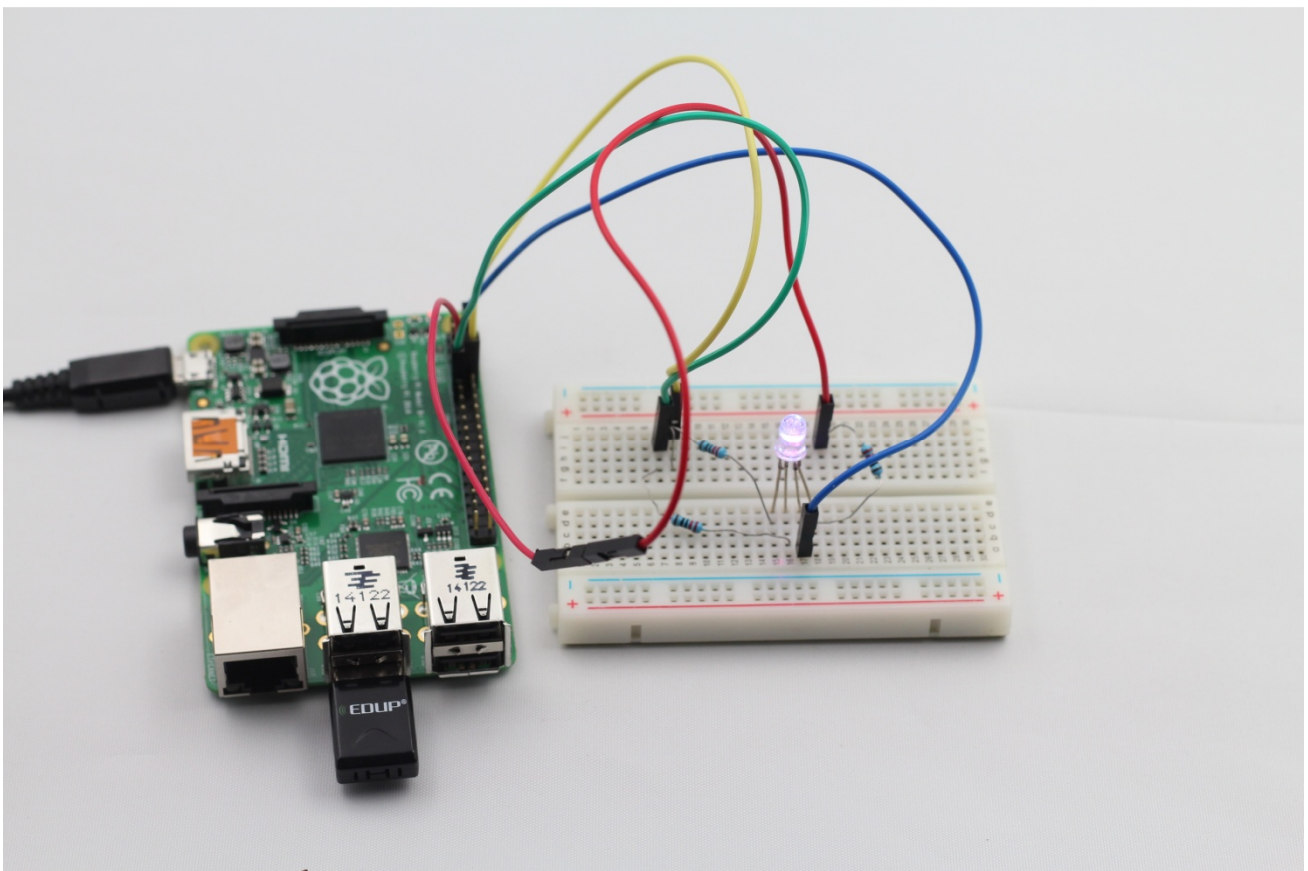
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 05_rgb.py
```

Jetzt können Sie sehen, dass die RGB LED das Licht in verschiedenen Farben emittiert.



Erweiterung

Sie können auch die Parameter der Funktion `ledColorSet()` selbst ändern, kompilieren und lassen die Codes laufen, dann sehen Sie den Farbenwechsel der RGB LED.

Lektion 6 Buzzer

Einführung

In dieser Lektion werden wir lernen, wie mit einem PNP-Transistor einen aktiven Buzzer anzutreiben, um einen Schall zu erzeugen.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* Buzzer (Aktiv)
- 1* PNP transistor (8550)
- 1* Widerstand (1K Ω)
- Jumper Wire

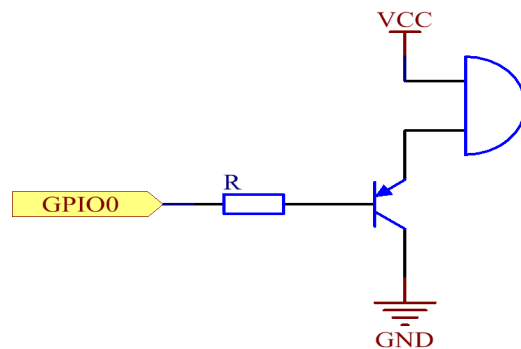
Versuchsprinzip

Elektronische Buzzer von integrierter Struktur sind mit Gleichstrom versorgt und häufig angewendet in Computern, Druckern, Fotokopiergeräten, Alarmen, elektronischen Spielzeugen, automotiven elektronischen Geräten, Telefonen, Zeitgebern und anderen elektronischen Produkten für Sprachgeräte. Buzzer können als die Aktiven und die Passiven kategorisiert werden (Siehe die Bilder unten). Die Seite mit 2 Pins nach oben umkehren, der mit einer grünen Leiterplatte ist passiv. Der andere mit schwarzem Klebstoff ist aktiv.



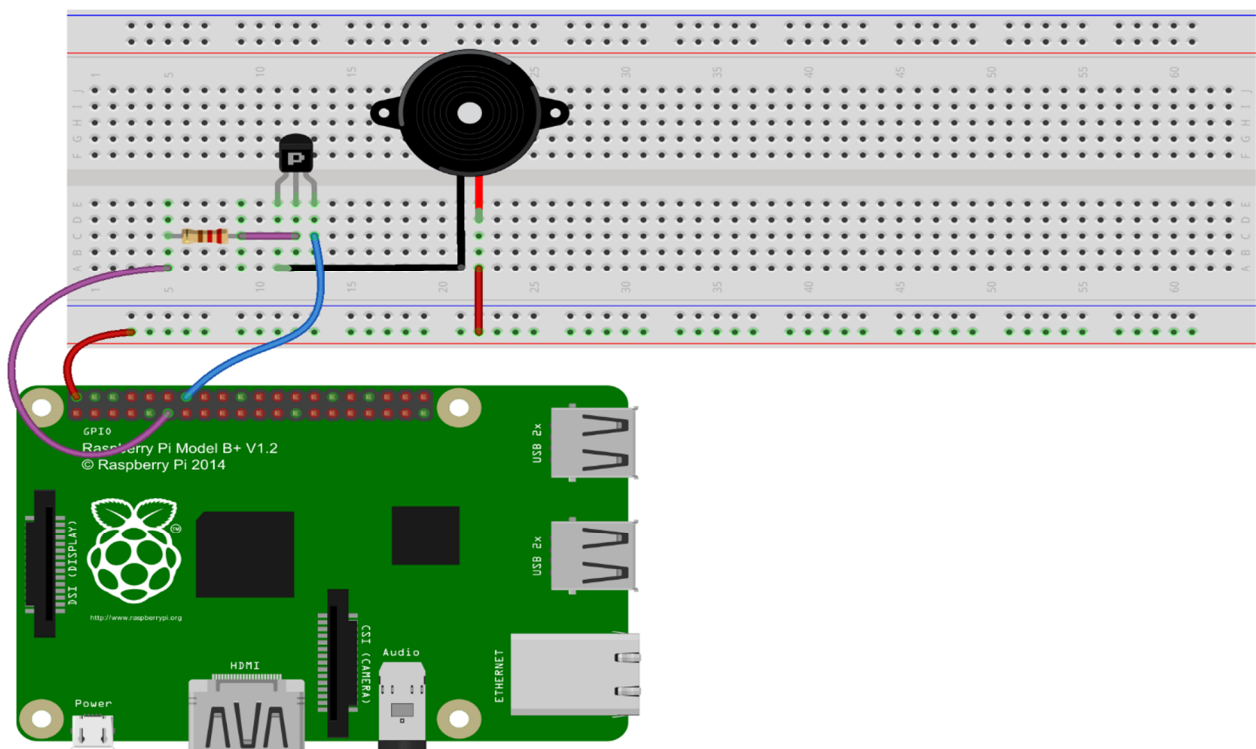
Der Unterschied: Ein aktiver Buzzer hat eine eingebaute Oszillationsquelle, deshalb wird er Schall erzeugen wenn elektrisiert. Aber ein passiver Buzzer hat keine solche Quelle, deshalb wird er keinen Schall erzeugen wenn Gleichstromsignale angewendet sind; stattdessen müssen Sie Rechteckwellen anwenden, deren Frequenz zwischen 2K und 5K ist, um ihn anzutreiben. Der aktive Buzzer ist oft teurer als der Passive wegen mehrerer eingebauten Oszillationsschaltungen.

In diesem Versuch ist ein aktiver Buzzer angewendet. Wenn der GPIO-Output des Raspberry Pi durch Programmierung mit dem niedrigen Pegel (0V) versorgt ist, wird der Transistor wegen der Stromsättigung leiten und der Buzzer einen Schall erzeugen. Aber wenn der IO des Raspberry Pi mit dem hohen Pegel versorgt ist, wird der Transistor gesperrt werden und der Buzzer keinen Schall erzeugen.



Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt: (Auf den Plus- und Minuspol des Buzzers achten)



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/06_Beep/
```

Schritt 3: Kompilieren

```
gcc beep.c -o beep -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./beep
```

Für Benutzer der Python-Sprache:

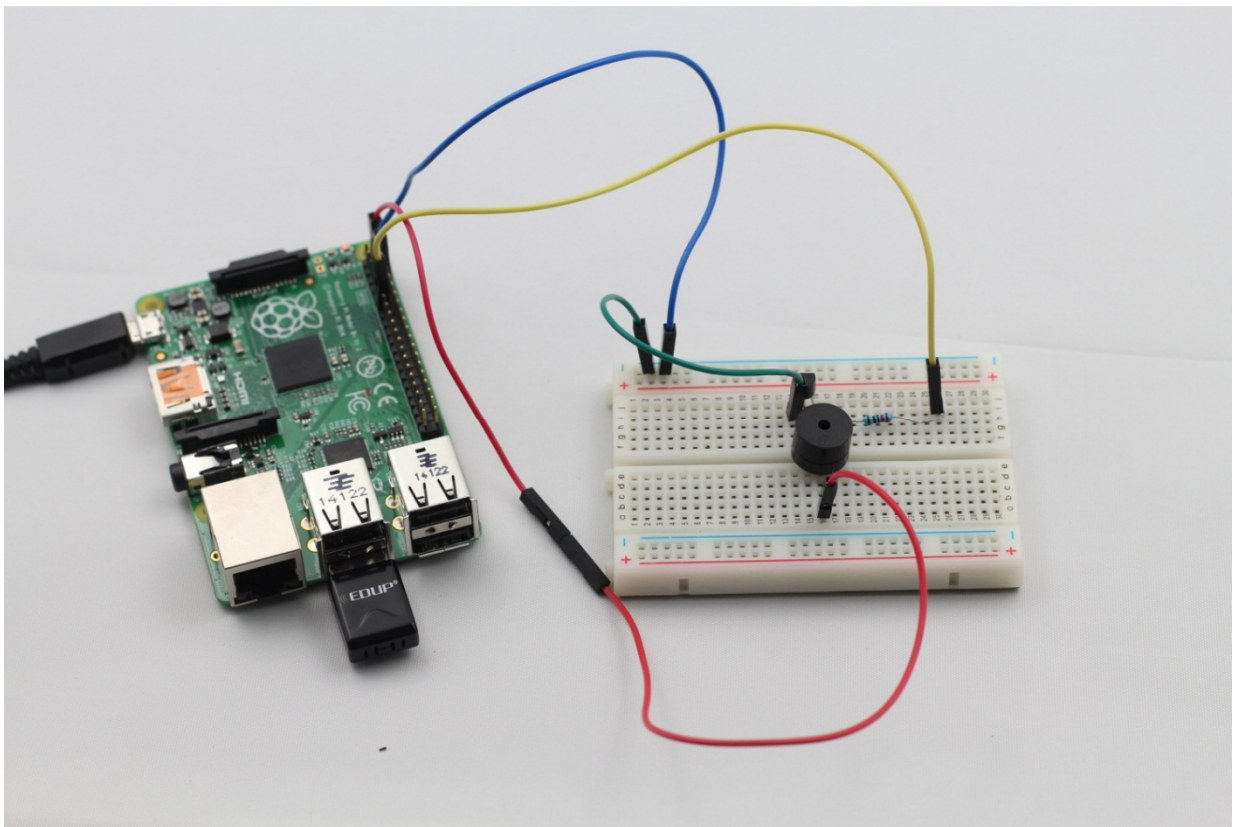
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 06_beep.py
```

Jetzt können Sie den Schall des Buzzers hören.



Erweiterung

Wenn Sie einen passiven Buzzer in der Hand haben, können Sie dadurch den aktiven Buzzer ersetzen. Jetzt können Sie mit Programmierungsgrundwissen einen Ton wie „do re mi fa so la si do“ erzeugen.

Lektion 7 Gleichstrommotor antreiben

Einführung

In dieser Lektion werden wir lernen, wie mit dem L293D einen Gleichstrommotor anzutreiben und ihn links- und rechtsgängig zu machen.

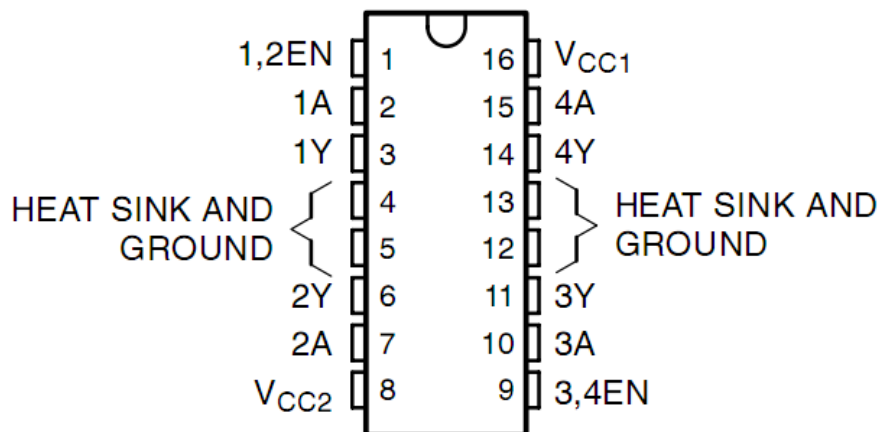
Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* L293D
- 1* Gleichstrommotor
- 1* Strommodul
- Jumper Wire

Versuchsprinzip

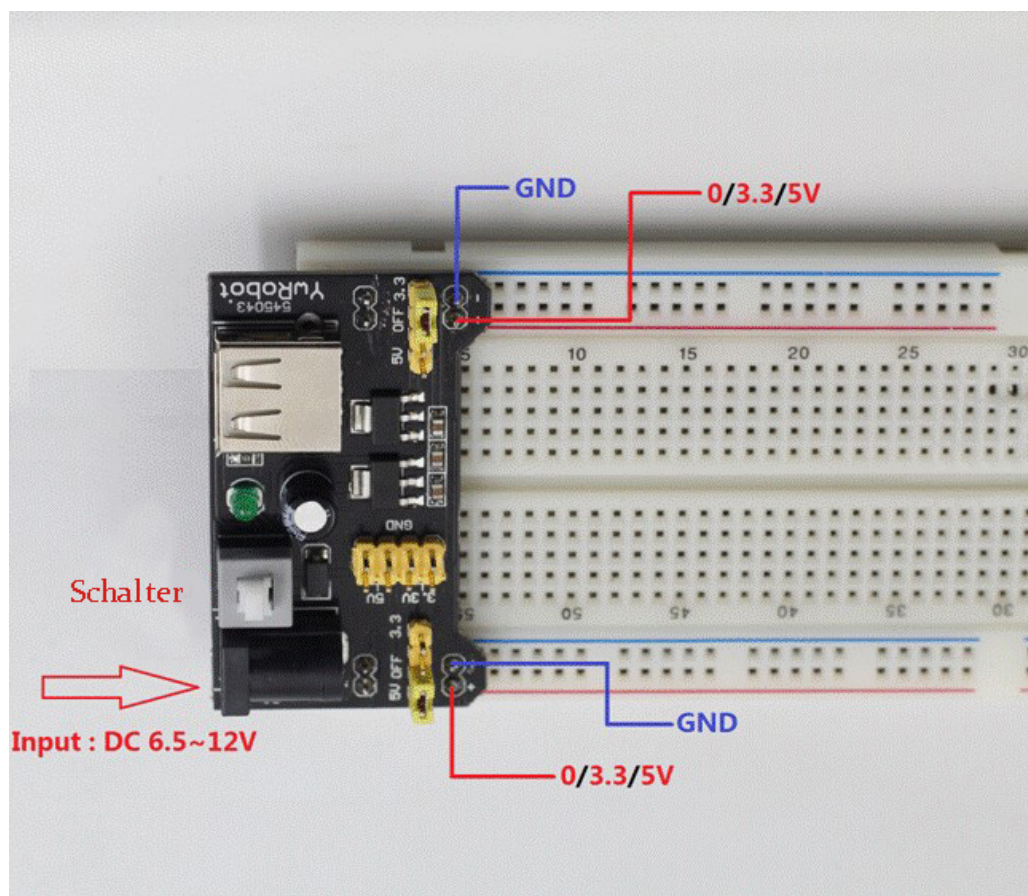
L293D

Das ist ein sehr praktischer Chip. Es kann zwei Gleichstrommotoren unabhängig ansteuern. In diesem Versuch benutzen wir einfach die Hälfte des Chips. Denn die meisten Pins auf der rechten Seite sind für einen anderen Motor, aber hier wenden wir nur einen Motor an.



L293D hat zwei Pins (Vcc1 and Vcc2) für Spannungsversorgung. Vcc2 wird verwendet, den Motor mit Spannung zu versorgen, während Vcc1 verwendet wird, den Chip mit Spannung zu versorgen. Weil hier ein kleiner DC Motor eingesetzt ist, verbinden wir die beiden Pins mit +5V. Wenn Sie einen Motor von mehr Stromverbrauch verwenden wollen, brauchen Sie Vcc2 mit einer externen Spannungsversorgung zu verbinden. Gleichzeitig soll GND von L293D mit GND von SunFounder Uno Board verbunden werden.

Zwei IOs des Raspberry Pi mit zwei Optokopplern verbinden, die den Strom isolieren, um den Raspberry Pi vor der Störung und dem Zusammenbruch zu schützen wenn der Motor läuft. Die Outputschnittstelle des Optokopplers mit dem Treiber-IC (L293D) des Motors verbinden. Wenn einer von den beiden IOs des Raspberry Pi im hohen Pegel und der andere im niedrigen Pegel ist, wird sich der Motor im Uhrzeigersinn oder gegen den Uhrzeigersinn drehen. Auch durch die PWM, können wir die Drehzahlregelung des Gleichstrommotors verwirklichen. Ein Motor ist ein induktives Gerät. Seine vier Dioden spielen eine wichtige Rolle im Freilauf und in der Stoßspannungsunterdrückung von Motorspulen, um irreversible Schäden am Treiber-IC zu verhindern.

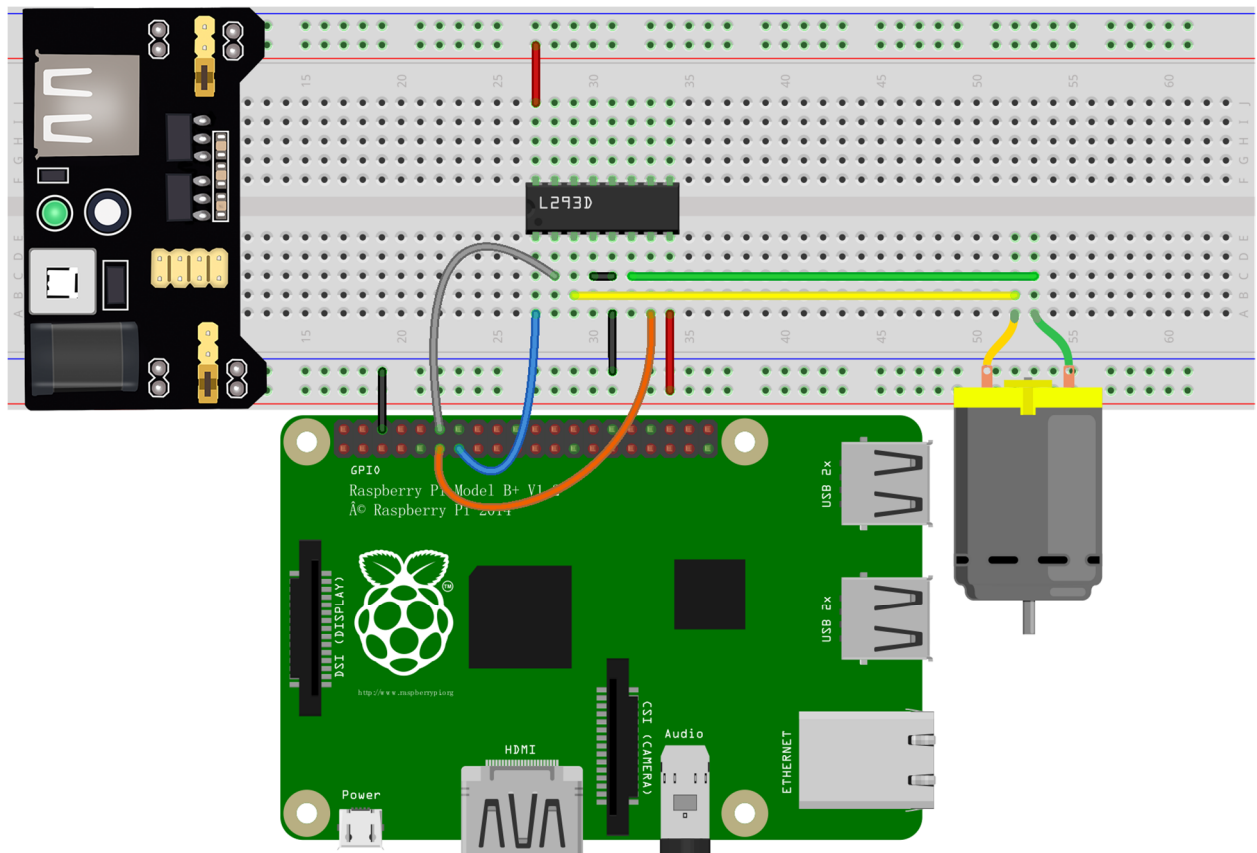


Das Modul kommt mit einer gelben Jumperkappe, die für die Auswahl zwischen dem Ausschalten und den Spannungen (3,3 V & 5 V) ist. Sie können die Ausgangsspannungen (0 / 3,3 / 5 V) dadurch umschalten, die Kappe in den exakten Pin zu setzen.

In diesem Versuch braucht es großen Strom zum Antrieb des Motors vor allem wenn er startet und stoppt. Das wird den normalen Lauf des Raspberry Pi stark stören. Dazu versorgen wir den Motor durch dieses Modul mit separatem Strom, dann kann es sicher und stetig laufen.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/07_Motor/
```

Schritt 3: Kompilieren

```
gcc motor.c -o motor -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./motor
```

Für Benutzer der Python-Sprache:

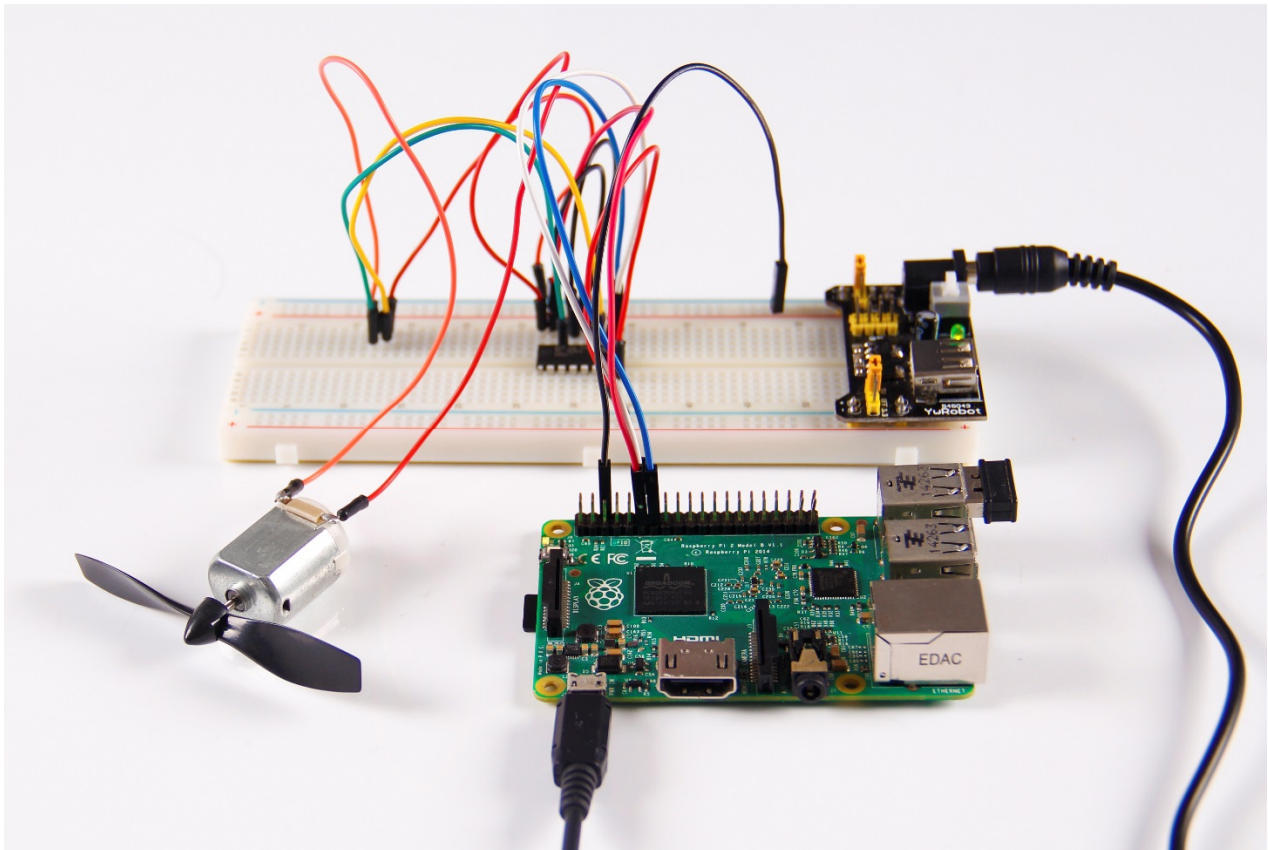
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```


Schritt 3: Ausführen

```
sudo python 07_motor.py
```

Jetzt können Sie den Motor rotieren sehen.



Erweiterung

Auf Basis von früheren Inhalten können Sie die links- und rechtsgängige Rotation des Motors auch mit Schalter oder über PWM ansteuern.

Zusammenfassung

In dieser Lektion haben Sie den Antriebsmodus des Gleichstrommotors, auch wie einen Motor mit Raspberry Pi anzutreiben, gelernt. Richten Sie bitte ein besonderes Augenmerk auf die Tatsache, dass ein Gleichstrommotor die gesamte Schaltung stark stört wenn er läuft. Deswegen brauchen Sie photoelektrische Isolation zu übernehmen und separate Stromversorgung zu liefern. Eine Freilaufdiode ist auch notwendig für den zuverlässigen und ständigen Lauf des Gesamtsystems.

Lektion 8 Drehgeber

Einführung

Der Drehgeber ist eine elektromechanische Vorrichtung, die die Winkelposition oder die Bewegung einer Achse in den analogen oder digitalen Code umwandelt. In diesem Versuch werden wir lernen, wie ihn anzuwenden.

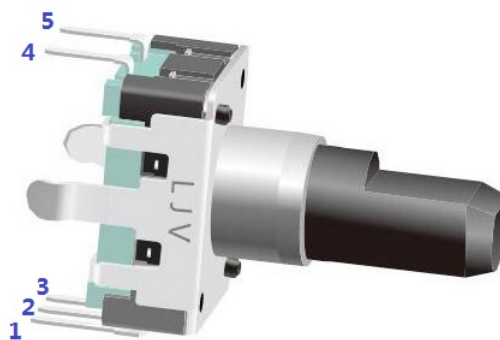
Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* Drehgeber
- Jumper Wire

Versuchsprinzip

Ein Drehgeber ist ein elektronischer Schalter mit einer Reihe von regelmäßigen Impulsen mit strenger zeitlicher Abfolge. Wenn angewendet zusammen mit IC, kann er Erhöhung, Verringerung, Umblättern und andere Funktionen, wie Menüauswahl, Akustikklangerstellung, Toaster-Temperaturregelung, Mouse-Scrolling usw. erreichen.

Es gibt zwei Typen von Drehgebern: Absolutdrehgeber und Inkrementaldrehgeber (relativ). Der Ausgang des Absolutdrehgebers gibt die aktuelle Position der Achse an, wie Winkelgeber. Der Ausgang des Inkrementaldrehgebers bietet Informationen über die Bewegung der Achse, die üblicherweise woanders zu den Informationen wie Geschwindigkeit, Entfernung und Position weiter verarbeitet sind. Hier ist die letztere eingesetzt.

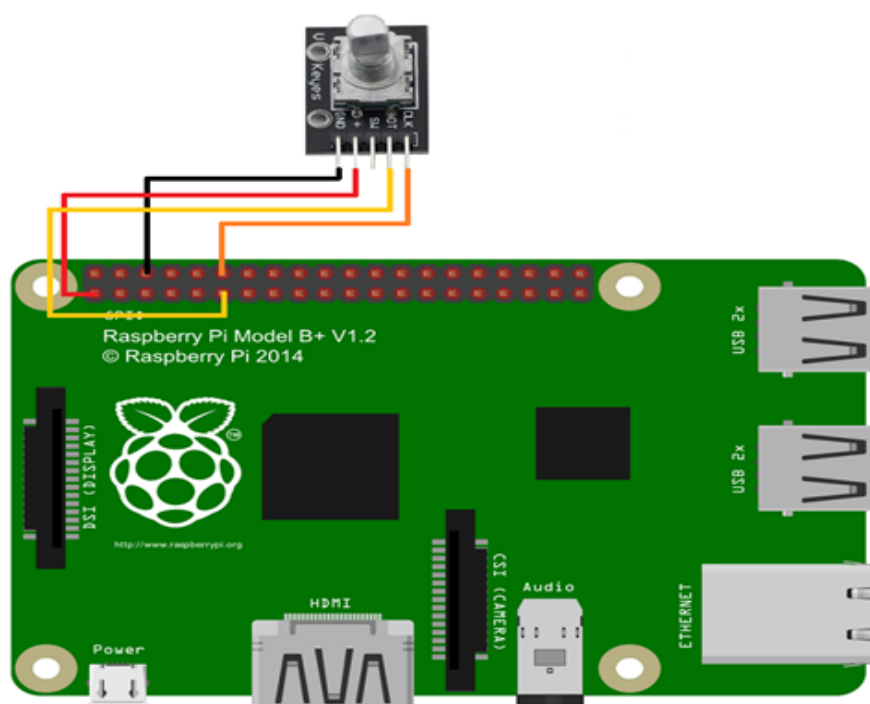


Meiste Drehgeber haben 5 Pins mit 3 Funktionen: Drehen nach links, nach rechts und Niederdrücken. Pin 4 und Pin 5 sind die Schalterverdrahtungsklemmen für Niederdrücken. Pin 2 ist im allgemeinen verbunden mit GND. Pins 1 und Pin 3 werden verbunden zuerst mit einem Pullup-Widerstand und dann mit einem Mikroprozessor. Hier sind sie verbunden mit GPIO0 und GPIO1 des Raspberry Pi. Wenn sich der Drehgeber im Uhrzeigersinn oder gegen den Uhrzeigersinn dreht, wird es im Pin 1 und Pin 3 Impulsausgänge geben.

Wenn GPIO0 und GPIO1 im hohen Pegel sind, dreht sich der Schalter im Uhrzeigersinn; Wenn GPIO0 im hohen Pegel aber GPIO1 im niedrigen Pegel ist, dreht sich der Schalter gegen den Uhrzeigersinn. Daher bei der Programmierung, brauchen Sie nur den Zustand des Pin 3 zu überprüfen, wenn Pin 1 im hohen Pegel ist, können Sie dann sehen, ob sich der Schalter im Uhrzeigersinn dreht, oder gegen den Uhrzeigersinn.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



Raspberry Pi

3.3V
GND
GPIO0
GPIO1

Drehgebermodul

'+'
GND
DT
CLK

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/08_RotaryEncoder/
```

Schritt 3: Kompilieren

```
gcc rotaryEncoder.c -o rotaryEncoder -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./rotaryEncoder
```

Für Benutzer der Python-Sprache:

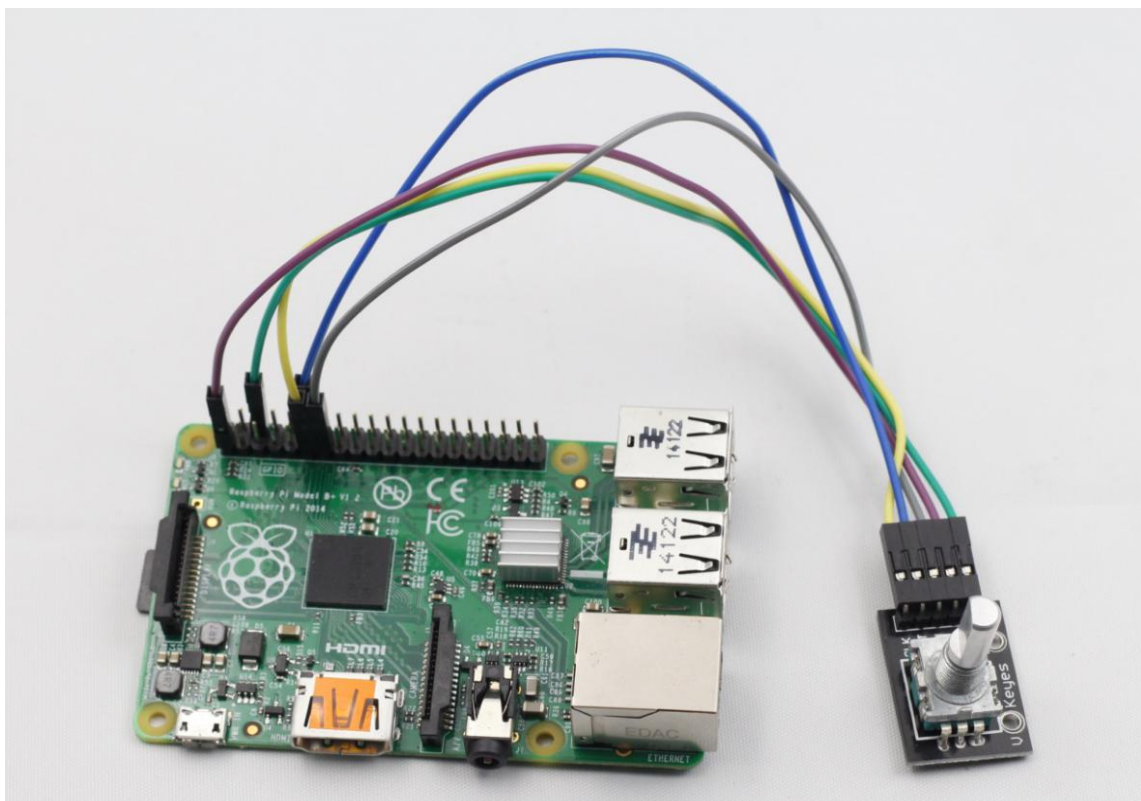
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 08_rotaryEncoder.py
```

Den Drehgeber drehen, um den Wert der Variablen im obigen Programm zu ändern. Sie werden sehen, dass der Wert auf dem Bildschirm gedruckt ist. Drehen Sie den Drehgeber im Uhrzeigersinn, wird sich der Wert erhöhen; oder gegen den Uhrzeigersinn, wird sich der Wert verringern.



Erweiterung

In diesem Versuch ist die Funktion - Niederdrücken des Drehgebers nicht erwähnt. Diese können Sie selbst versuchen. Das heißt, wenn Sie den Drehgeber drücken, wird der Wert der Variablen in Nulllage gehen.

Lektion 9 555 Timer

Einführung

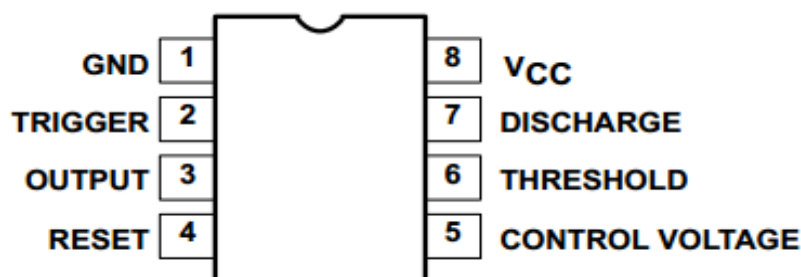
Die integrierte Schaltung NE555 ist ein aktives elektronisches Bauelement, das häufig in Zeitsteuerungen und Taktgebern, teils sogar in Schaltnetzteilen Verwendung findet. Der NE555 wurde häufig abkürzend auch nur als 555 bezeichnet. In dieser Lektion werden wir lernen, wie den 555 Timer anzuwenden.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* NE555
- 3* Widerstand (1*1K Ω , 2*10K Ω)
- 2* Kondensator (100nF)
- Jumper Wire

Versuchsprinzip

Ein 555 Timer ist eine mittelgroße IC-Vorrichtung, die die analogen und digitalen Funktionen vereint. Kostengünstig und leistungszuverlässig verbindet es die externen Widerstände mit den Kondensatoren, um die Funktionen des Multivibrators, des monostabilen Triggers, des Schmitt-Triggers und andere Schaltungen, die die Impulse erzeugen und transformieren können, zu erzielen. Als Timer wird er auch häufig angewendet bei Instrumenten, Haushaltsgeräten, elektronischer Messung, automatischer Steuerung und anderen Bereichen. Pinbelegung wie folgt:

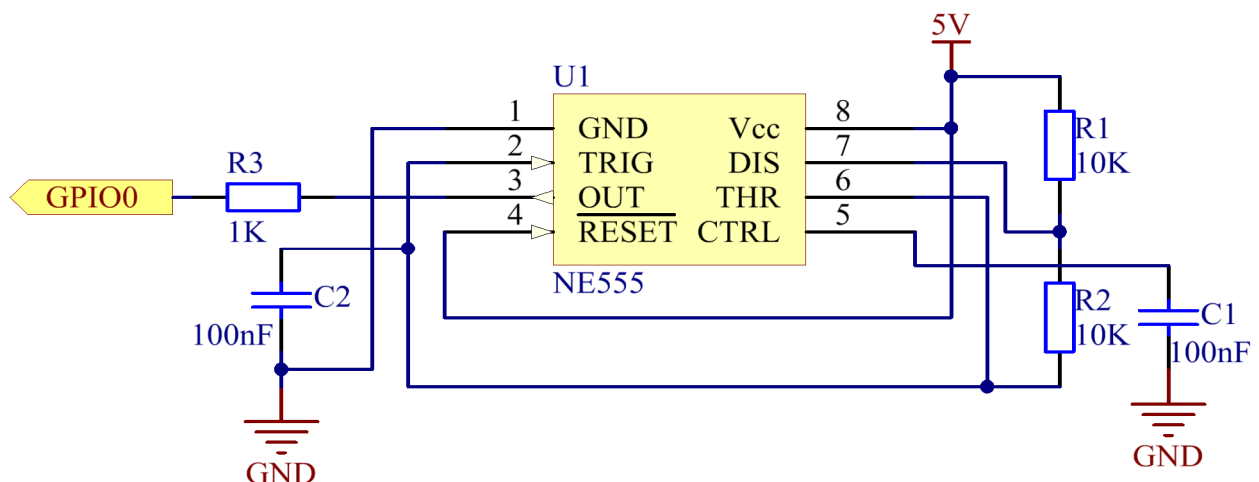


Der NE555 besitzt ein 8-poliges DIL-Gehäuse und ist somit auch ziemlich Platz sparend. Jeder Pin des Schaltkreises hat natürlich eine entsprechende Funktion. Dies sind:

- Pin 1 (**GND**): Masse (0 V).
- Pin 2 (**Trigger**): OUT steigt an und das Intervall beginnt, wenn dieser Eingang $1/3$ VCC unterschreitet.

- Pin 3 (**Ausgang**): Dieser Ausgang wird durch das IC auf VCC oder GND gezogen.
- Pin 4 (**Reset**): Ein Zeitintervall kann unterbrochen werden, wenn dieser Eingang auf GND gesetzt wird.
- Pin 5 (**Steuerspannung**): Zugriff auf den internen Spannungsteiler (üblicherweise 2/3 VCC).
- Pin 6 (**Schaltswelle**): Das Intervall endet, wenn die Spannung an THR größer als bei CTRL ist.
- Pin 7 (**Entladung**): Open-Collector-Ausgang: kann einen Kondensator zwischen den Intervallen entladen.
- Pin 8 (**VCC**): Versorgungsspannung.

Der NE555 verfügt über drei grundlegende Betriebsarten. In diesem Versuch wird er im astabilen Modus angewendet, um Rechteckwellen zu erzeugen.



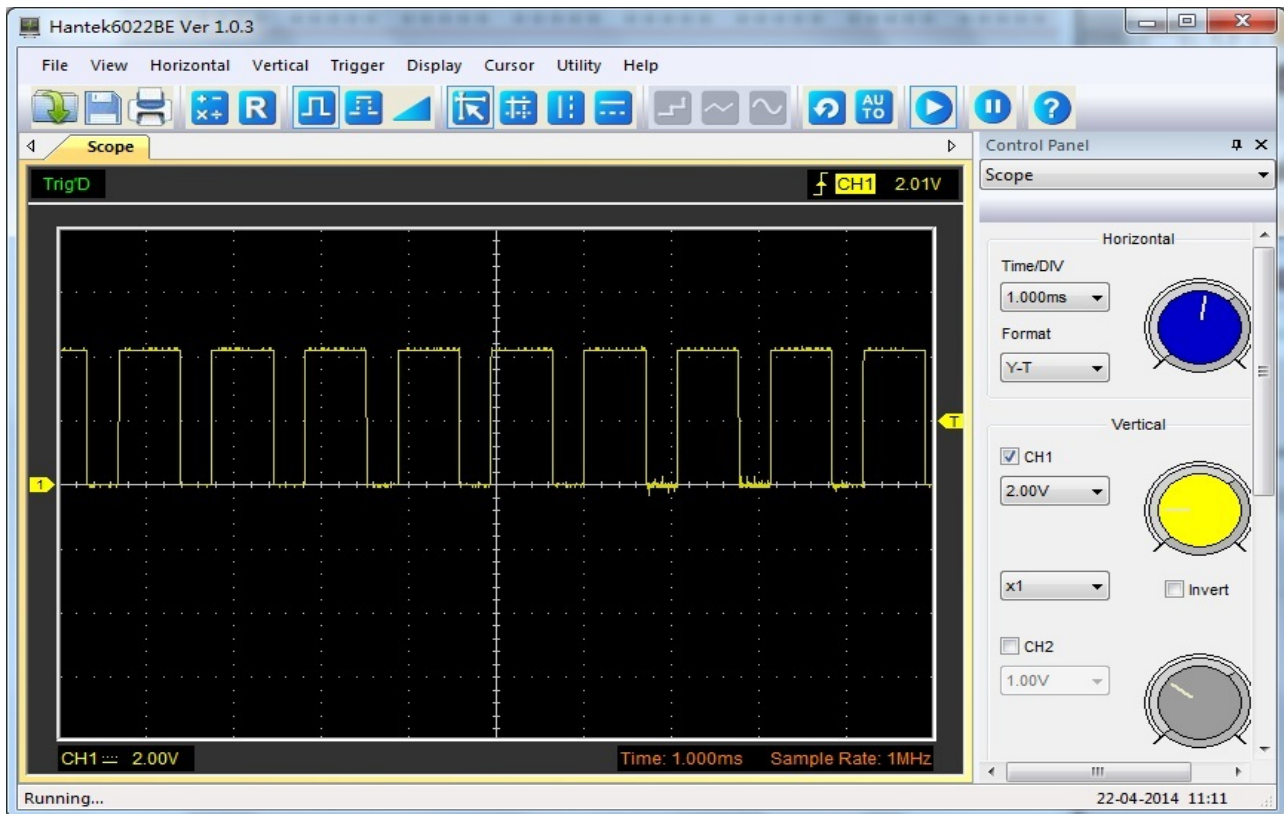
Im astabilen Modus lässt sich die Frequenz der Ausgangswellenform des 555 Timers bestimmen von R_1 , R_2 und C_2 :

$$f = \frac{1}{\ln 2 * C_2 * (R_1 + 2R_2)}$$

In der obigen Schaltung, $R_1=R_2=10K\Omega=10^4\Omega$; $C_2=100nF=10^{-7}F$, so können wir die Frequenz bekommen:

$$f = \frac{1}{\ln 2 * 10^{-7} * (10^4 + 2 * 10^4)} \approx 481Hz$$

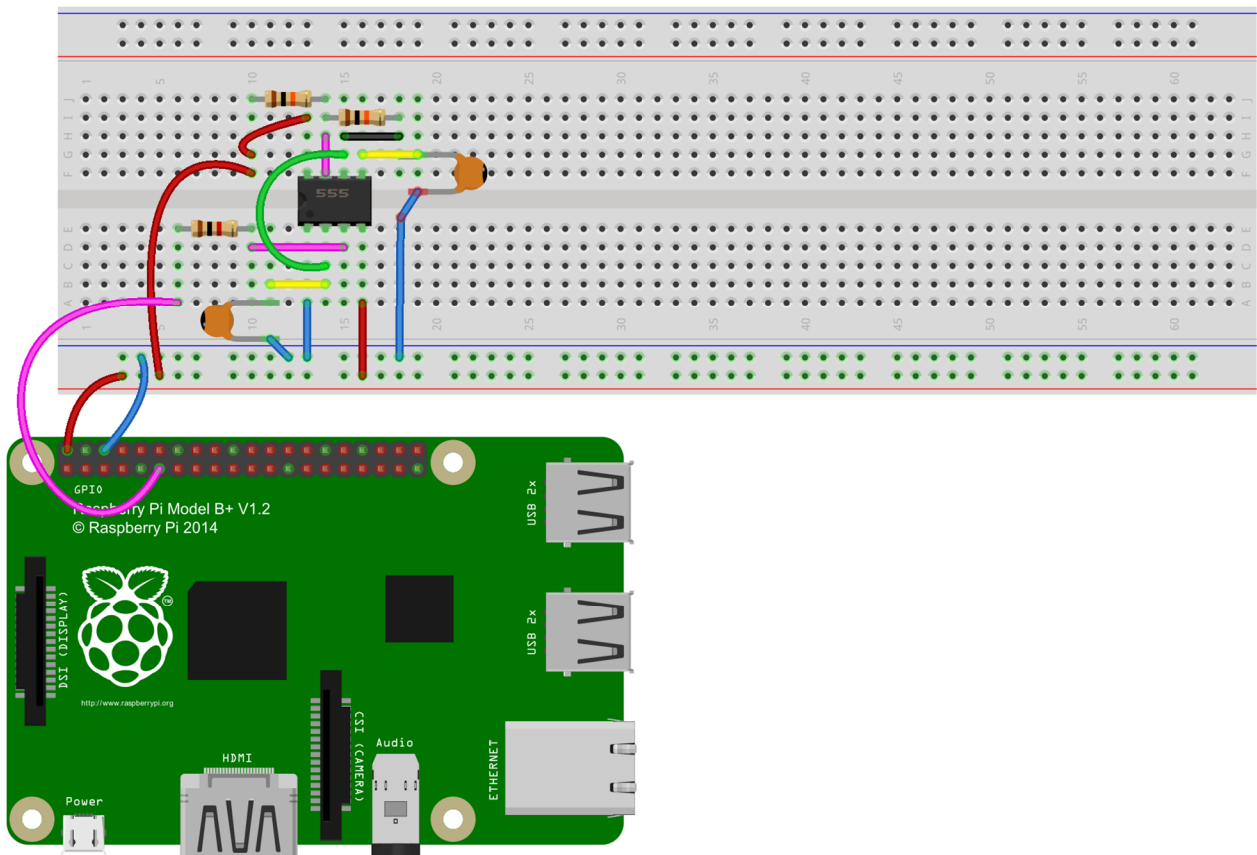
Um die Frequenz der Ausgangswellenform zu beobachten, nach der Schaltungsverbindung wenden wir dem schematischen Diagramm gemäß einen Oszilloskop an. Wir sehen, dass sie im Einklang mit dem oben berechneten Ergebnis steht.



Der Ausgang-Pin (z.B. Pin 3) des 555 Timers mit GPIO0 des Raspberry Pi verbinden, durch Programmierung den GPIO0 als den Interruptionsmodus mit der positiven Flanke Konfigurieren, und dann mit der Interruption die vom 555 erzeugten Rechteckwellenimpulse ermitteln. Die Leistung der Unterbrechungsroutine(Interrupt Service Routine, kurz ISR) ist, einer Variable 1 zu addieren.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/09_Timer555/
```

Schritt 3: Kompilieren

```
gcc timer555.c -o timer555 -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./timer555
```

Für Benutzer der Python-Sprache:

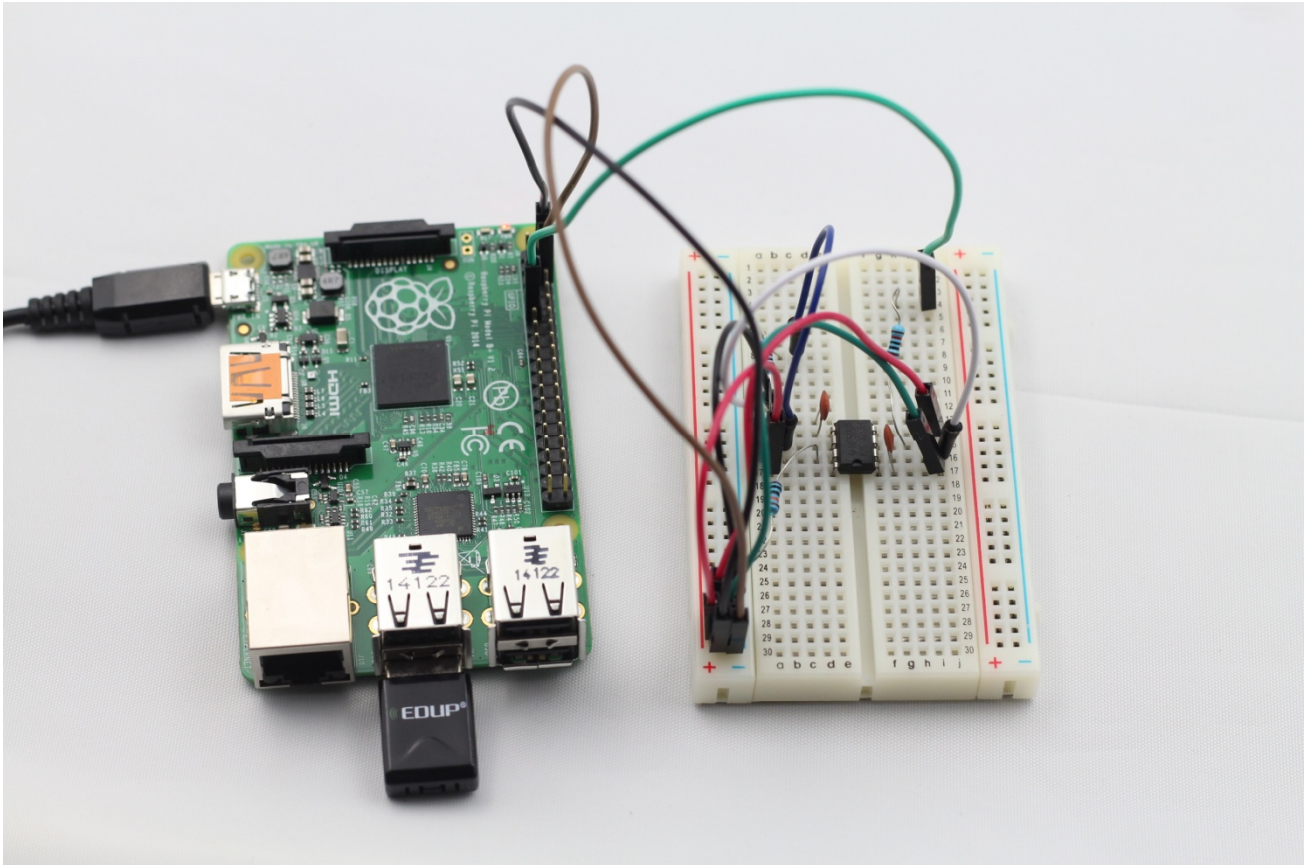
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 09_timer555.py
```

Jetzt können Sie sehen, dass die Daten auf dem Display gedruckt sind, die die vom 555 erzeugten Rechteckwellen sind. Das Programm zählt Impulse durch Interruption.



Erweiterung

Die oben genannte 555-Timer-Schaltung gibt Rechteckwellen mit konstanter Frequenz und Tastgrad aus. Sie können diese Schaltung einfach ändern, um die Frequenz und den Tastgrad zu regeln. Deshalb können Sie das Atmungslicht, beschrieben in Lektion 4, ohne Raspberry Pi machen.

Lektion 10 LEDs antreiben mit 74HC595

Einführung

Der 74HC595 ist ein sehr beliebter und weit verbreiteter Schieberegister-Baustein. Man kann Schieberegister beispielsweise zur Port-Erweiterung für Ausgänge verwenden, um einen seriellen Datenstream in einen Parallelen umzusetzen oder zum Multiplexen. In dieser Lektion werden wir lernen, wie mit dem 74HC595 acht LEDs regelmäßig blinken zu lassen.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* 74HC595
- 8* LED
- 8* Widerstand (220Ω)
- Jumper Wire

Versuchsprinzip

Der 74HC595 ist ein 8-bit 3-state Seriell-in/Parallel-out Schieberegister mit Latch. Das Latch braucht man, damit beim seriellen reinschieben der Bits diese nicht nacheinander an den Ausgängen erscheinen, sondern erst wenn das ganze Datenwort reingeschoben ist. Der 74HC595 ist sogar kaskadierbar, d.h. man kann theoretisch beliebig viele Ausgänge realisieren, ohne weitere Portleitungen zu benötigen.

Prinzipiell besteht ein Schieberegister aus zwei Einheiten. Dem Schieberegister und dem Ausgangsregister. Im Schieberegister werden die Daten sozusagen zunächst abgelegt. Nach einem entsprechenden Signalwechsel werden sie später an das Ausgaberegister übergeben. Alle im Ausgaberegister befindlichen Daten können dann ebenfalls bei Bedarf, sprich dem entsprechenden Signalwechsel, an die Ausgangspins weitergegeben werden. Pinbelegung wie folgt:

1	Q1	VCC	16
2	Q2	Q0	15
3	Q3	<u>DS</u>	14
4	Q4	<u>CE</u>	13
5	Q5	STcp	12
6	Q6	<u>SHcp</u>	11
7	Q7	MR	10
8	GND	Q7'	9

Q0-Q7: 8-Bit Paralleldatenansgangspin, in der Lage 8 LEDs oder 8 Pins einer 7-Segment-Anzeige direkt anzusteuern.

Q7': Das tolle an Schieberegistern ist, dass man mehrere davon hintereinander, also in Reihe, schalten kann. Ist das Schieberegister voll werden die Daten die sonst sozusagen hinten wieder herausfallen würden in das nächste Schieberegister übergeben.

MR: Reset Pin, wird der Inhalt des Schieberegisters zurückgesetzt. Aktiv beim niedrigen Pegel. Hier ist direkt mit 5V verbunden.

SH_CP: Zeitabfolgeeingang des Schieberegisters. Mit einer positiven Flanke, bewegt sich die Daten im Schieberegister sukzessive ein Bit, das heißt, die Daten in Q1 gehen in Q2, und so weiter. Mit einer negativen Flanke bleiben die Daten im Schieberegister unverändert.

ST_CP: Zeitabfolgeeingang des Speicherregisters. Mit einer positiven Flanke, bewegt sich die Daten im Schieberegister in das Speicherregister.

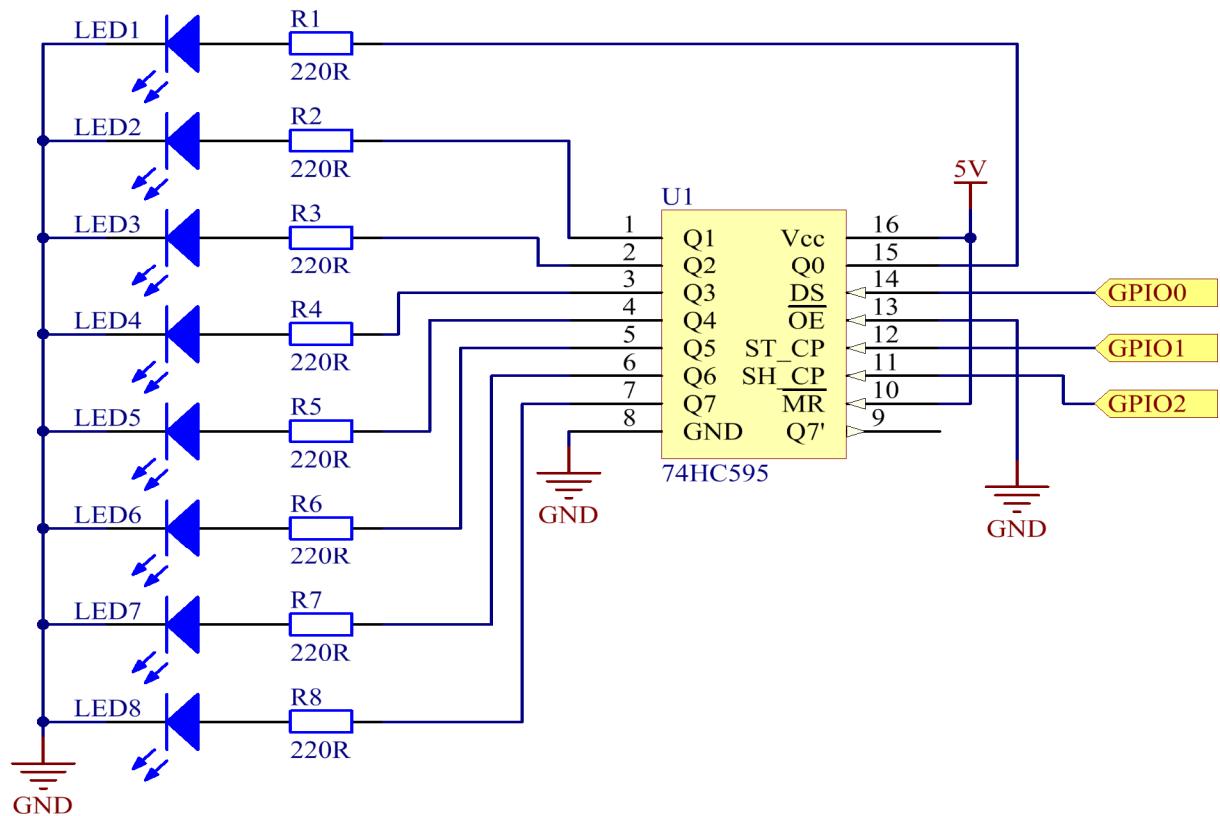
OE: Output Enable Pin, Schaltet die Ausgangspins Q0 bis Q7 ein oder aus. Aktiv beim niedrigen Pegel, mit GND verbunden.

DS: Hier stehen die Daten an, die in das Schieberegister übernommen werden sollen.

VCC: Spannungsversorgung.

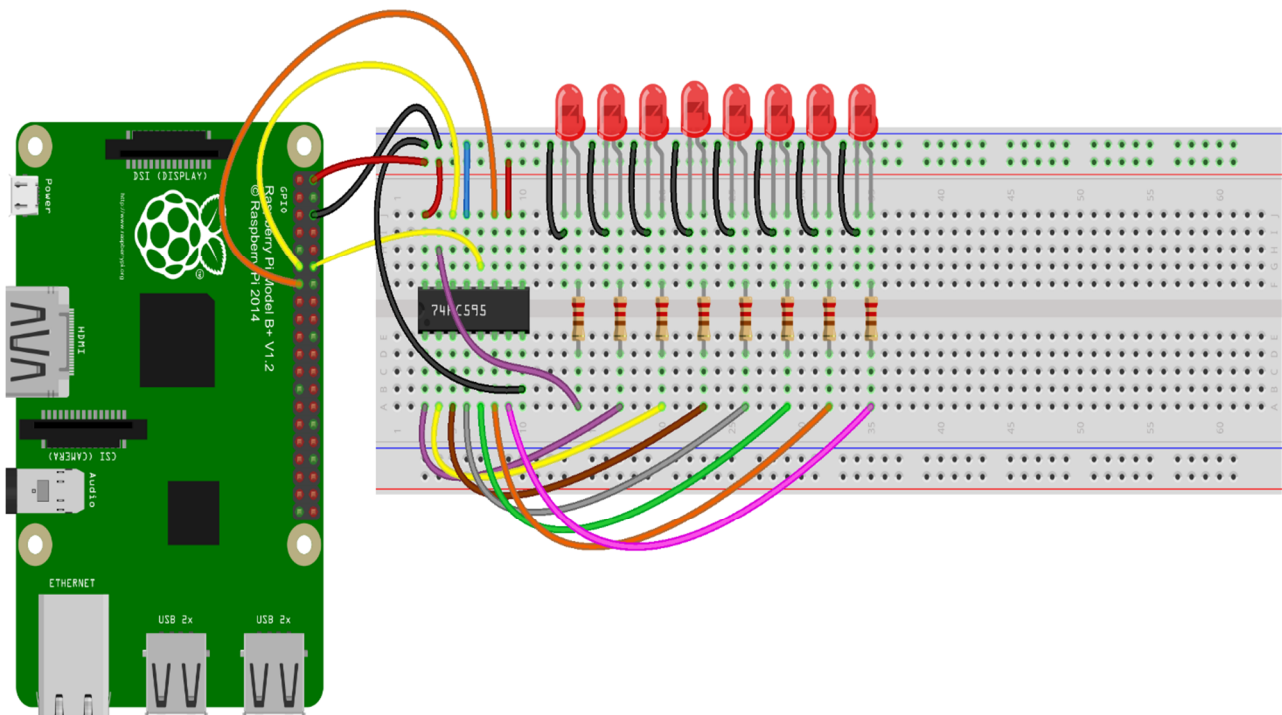
GND: Masse.

Verbinden wir ST_CP mit GPIO1 des Raspberry Pi, SH_CP mit GPIO2, und DS mit GPIO0. Wenn SH_CP (der Takteingang des Schieberegisters) mit einer positiven Flanke ist, geben wir die Daten im DS-Pin ins Schieberegister ein. Und wenn ST_CP (der Takteingang des Speichers) mit einer positiven Flanke ist, geben wir die Daten ins Speicherregister ein. Dann können wir die Zustände von SH_CP und ST_CP durch GPIO ansteuern und serielle Eingangsdaten in parallele Ausgangsdaten transformieren, um GPIOs des Raspberry Pi zu sparen.



Versuchsv erfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/10_74HC595_LED/
```

Schritt 3: Kompilieren

```
gcc 74HC595_LED.c -o 74HC595_LED -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./74HC595_LED
```

Für Benutzer der Python-Sprache:

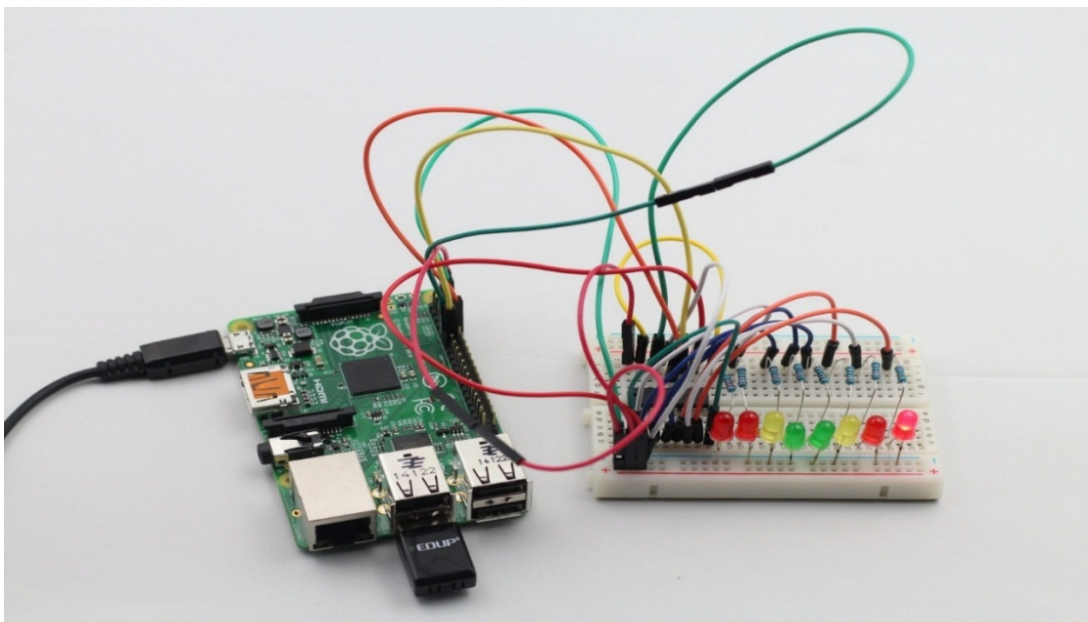
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 10_74HC595_LED.py
```

Jetzt können Sie sehen, dass acht LEDs regelmäßig blinken wie folgt:



Erweiterung

In diesem Versuch werden drei Raspberry Pi GPIOs eingesetzt, um separat 8 LEDs auf Basis von 74HC595 anzusteuern. In der Tat hat 74HC595 eine andere leistungsstarke Funktion - Kaskade. Mit der Kaskade, können Sie einen Mikroprozessor anwenden, um mehr Peripheriegeräte anzusteuern.

Lektion 11 7-Segment-Anzeige antreiben mit 74HC595

Einführung

In dieser Lektion werden wir lernen, wie mit 74HC595 eine 7-Segment-Anzeige zur Darstellung der dezimalen Ziffern 0 bis 9 anzutreiben.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* 74HC595
- 1* 7-Segment-Anzeige
- 1* Widerstand (1K Ω)
- Jumper Wire

Versuchsprinzip

Die 7-Segment-Anzeige besteht aus sieben LEDs in Form von Rechteck. Jede der sieben LEDs ist ein Segment genannt, weil wenn beleuchtet das Segment einen Teil einer numerischen Ziffer (Dezimal und Hexadezimal) formt, um angezeigt zu werden. Der achte LED wird manchmal in der gleichen Bestückung verwendet, womit eine Dezimalstelle (DP) angezeigt werden kann. Wenn zwei oder mehr 7-Segment-Anzeigen miteinander verbunden sind, lassen sich die Zahlen größer als zehn anzeigen.



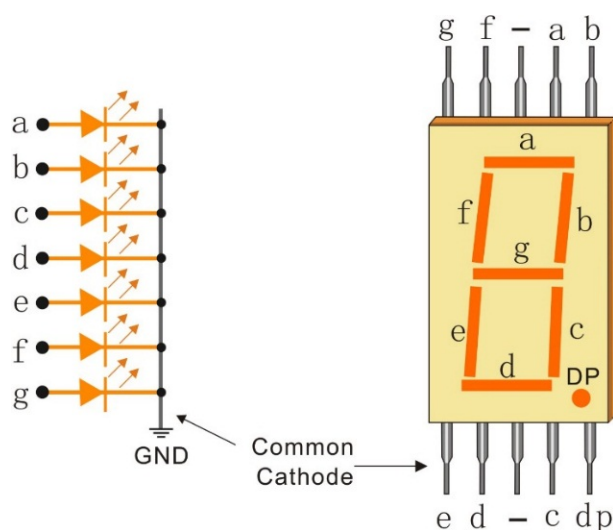
Jeder der sieben LEDs im Display ist ein Positionsegment gegeben, einer seiner Anschlusspins wird gerade aus dem rechtwinkligen Kunststoffgehäuse gebracht. Diese LED-Pins werden von **a** bis **g** gekennzeichnet, die jede einzelne LED darstellt. Die anderen LED-Pins sind miteinander verbunden und verdrahtet, um einen gemeinsamen Pin zu bilden.

Durch Vorwärtsspannung der entsprechenden Pins von den LED-Segmenten in einer bestimmten Reihenfolge werden dazu einige Segmente hell und die anderen dunkel, was erlaubt, dass das gewünschte Zeichenmuster der Ziffern auf dem Display erzeugt werden kann. Das lässt uns dann jeden der zehn Dezimalziffern 0 bis 9 auf der gleichen 7-Segment-Anzeige zeigen können.

Der gemeinsame Pin der Anzeige ist in der Regel verwendet, um den Typ der 7-Segment-Anzeigen zu identifizieren. Da jede LED zwei Verbindungspins hat, die „Anode“ und die „Kathode“, gibt es damit zwei Arten von 7-Segment-Anzeigen: die mit gemeinsamer Kathode (CC) und die mit gemeinsamer Anode (CA).

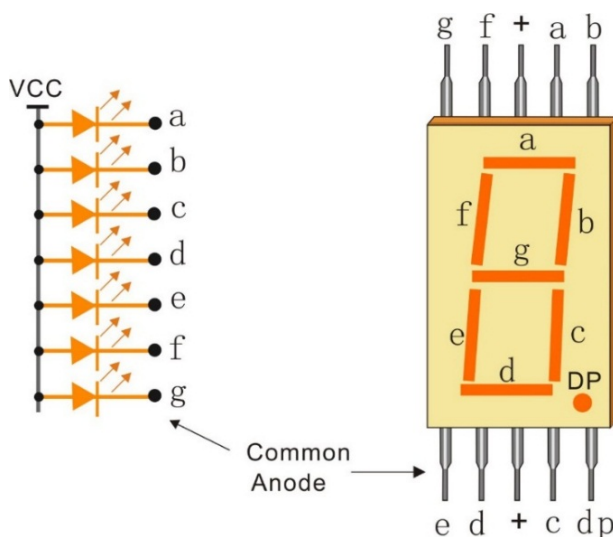
7-Segment-Anzeige mit gemeinsamer Kathode

Im Display mit gemeinsamer Kathode sind alle Kathodenanschlüsse der LED-Segmente zusammen mit Logik "0" oder Masse verbunden. Die einzelnen Segmente werden dadurch beleuchtet, ein Signal "HIGH" oder Logik "1" anzuwenden und einen strombegrenzenden Widerstand an die Anode des einzelnen Segments (a-g) anzuschließen.

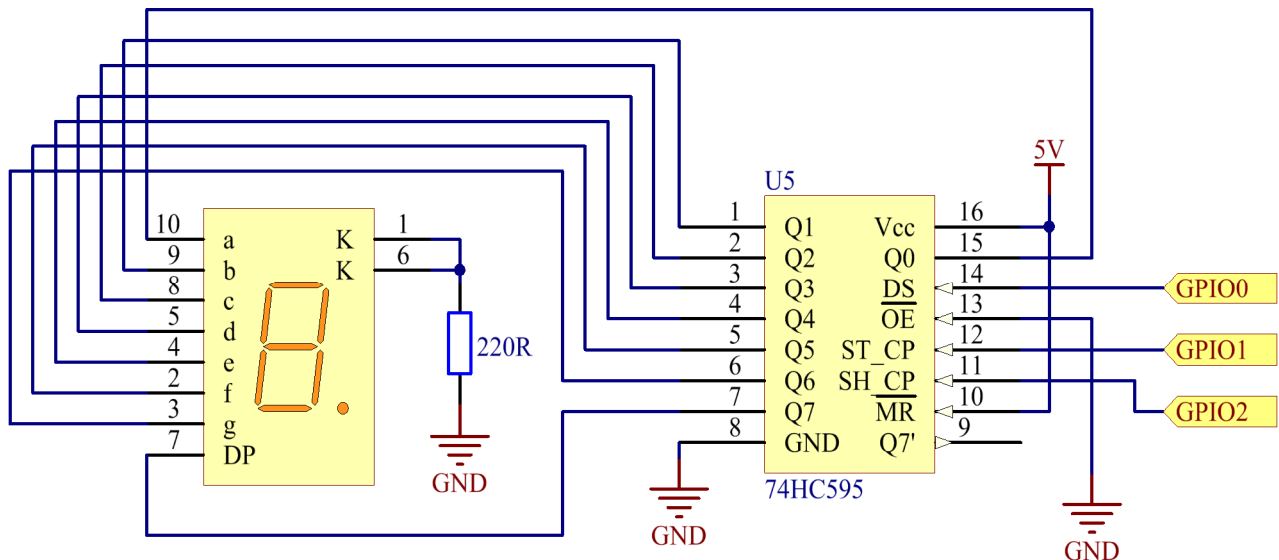


7-Segment-Anzeige mit gemeinsamer Anode

Im Display mit gemeinsamer Anode sind alle Anodenanschlüsse der LED-Segmente zusammen mit Logik "1" verbunden. Die einzelnen Segmente werden dadurch beleuchtet, ein Signal "LOW" "GND" oder Logik "0" anzuwenden und einen strombegrenzenden Widerstand an die Kathode des einzelnen Segments (a-g) anzuschließen.



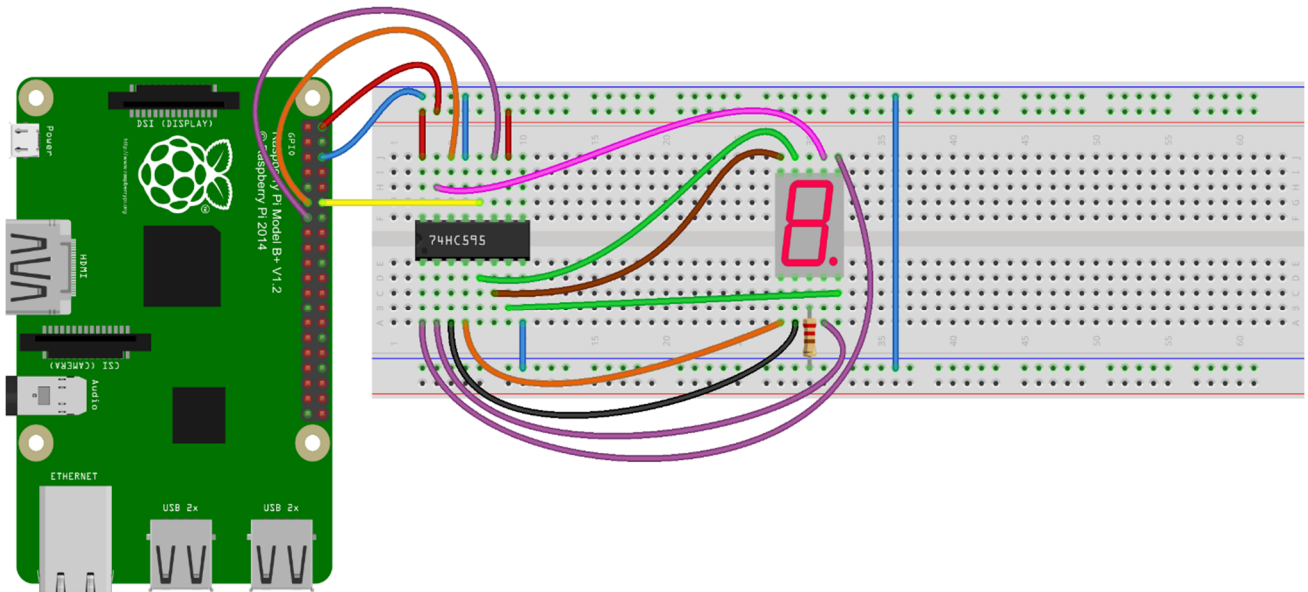
In diesem Versuch wird eine 7-Segment-Anzeige mit gemeinsamer Kathode verwendet, die alle Kathoden der LEDs miteinander verbindet, um eine Elektrode gemeinsamer Kathode (COM) zu bilden. Es soll mit der Erde verbunden werden. Wenn die Anode einer LED in einem bestimmten Segment beim hohen Pegel ist, wird das entsprechende Segment aufleuchten; wenn die beim niedrigen Pegel ist, wird aber das Segment nicht aufleuchten.



Verbinden wir Pin ST_CP des 74HC595 mit GPIO1 des Raspberry Pi, SH_CP mit GPIO2, DS mit GPIO0, parallele Ausgangsschnittstellen mit 8 Segmenten der LED-Segment-Anzeige. Wenn SH_CP (der Takteingang des Schieberegisters) mit einer positiven Flanke ist, geben wir die Daten im DS-Pin ins Schieberegister ein. Und wenn ST_CP (der Takteingang des Speichers) mit einer positiven Flanke ist, geben wir die Daten ins Speicherregister ein. Dann können Sie die Zustände von SH_CP und ST_CP durch GPIO ansteuern und serielle Eingangsdaten in parallele Ausgangsdaten transformieren, um GPIOs zu sparen und die LED-Segment-Anzeige anzutreiben.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/11_Segment/
```

Schritt 3: Kompilieren

```
gcc segment1.c -o segment1 -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./segment1
```

Für Benutzer der Python-Sprache:

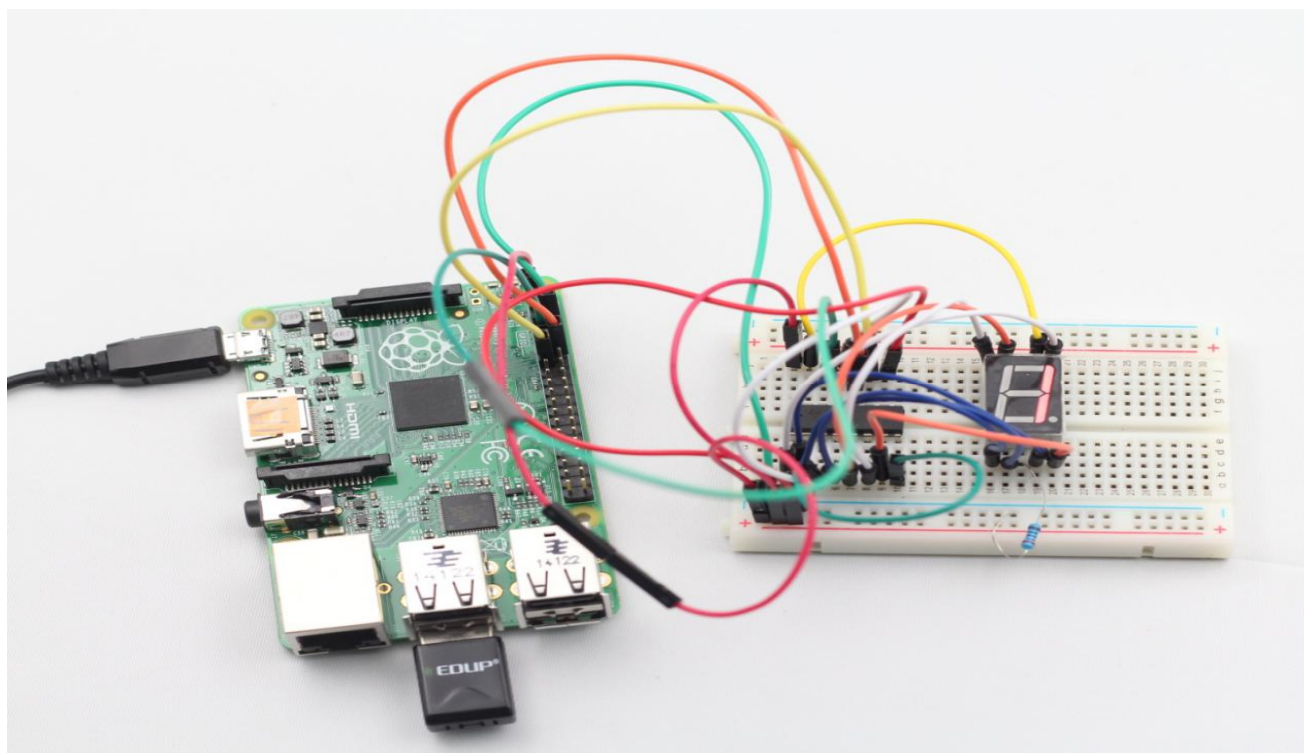
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

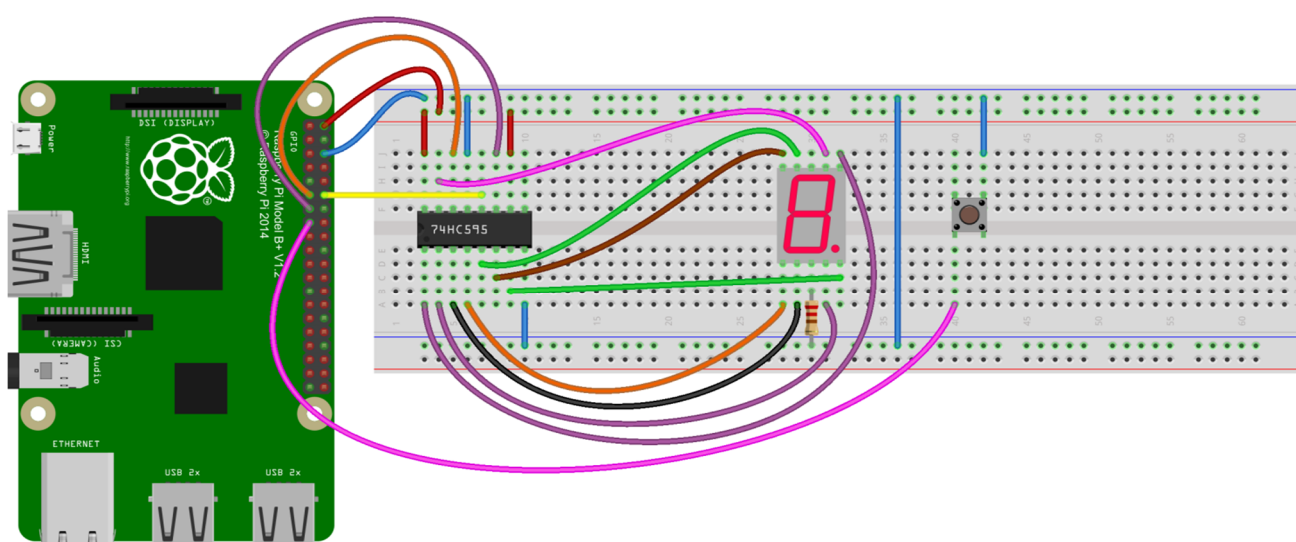
```
sudo python 11_segment.py
```

Jetzt können Sie sehen, dass sich 0 bis 9 und A bis F auf der 7-Segment-Anzeige zeigt. Die praktische Wirkung ist wie unten dargestellt:



Erweiterung

Auf Basis dieses Versuches können Sie auch die Hardware und die Software etwas modifizieren, um einen Würfel zu machen. Für Hardware, fügen Sie der originalen Platine einen Schalter hinzu. Dann zeigt sich die modifizierte Schaltung wie folgt:



fritzing

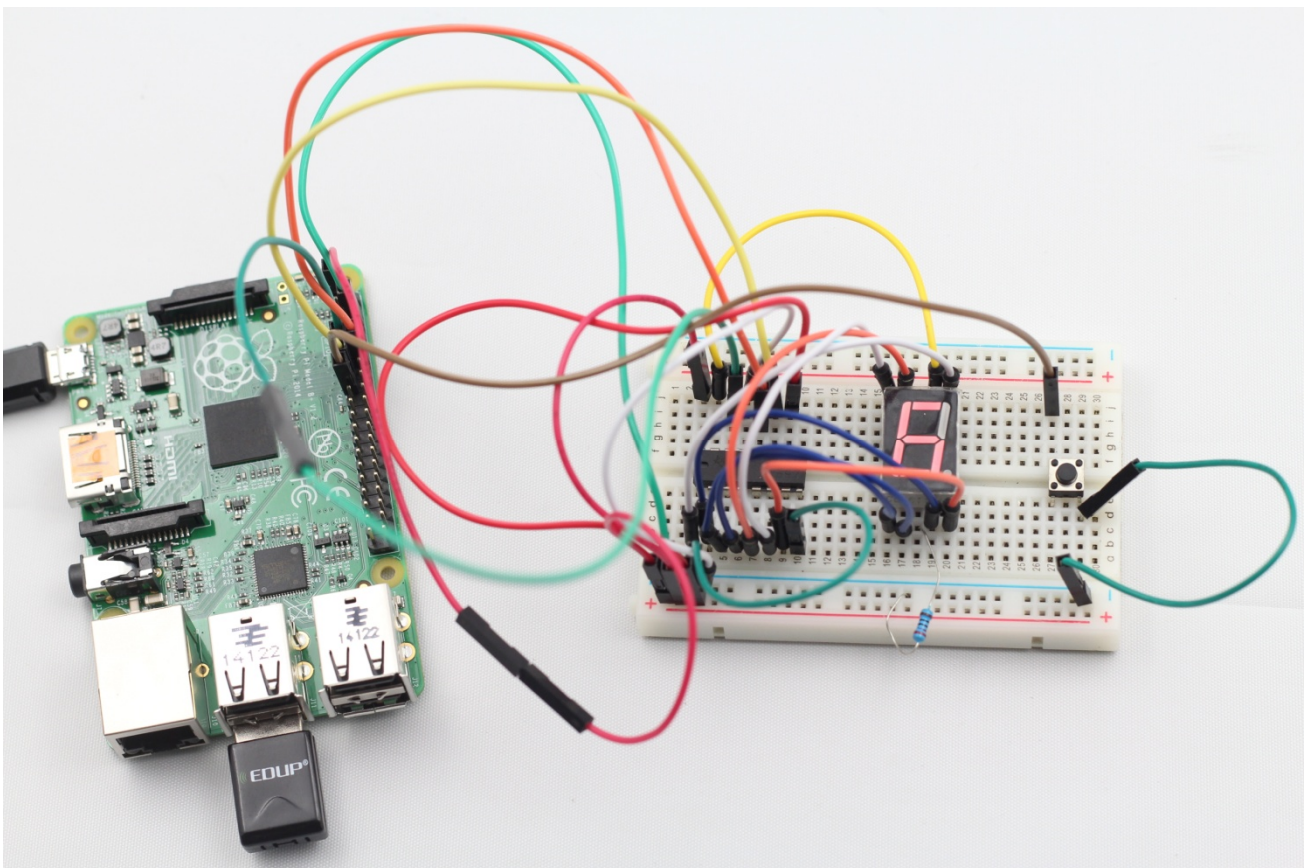
Als nächstes ins directory 11_Segment gehen, und *dice.c* kompilieren

```
gcc dice.c -o dice -lwiringPi
```

Ausführen

```
sudo ./dice
```

Sie können sehen, dass die Ziffern zwischen 0 und 6 auf der Segment-Anzeige schnell blinken. Den Schalter auf der Steckplatine drücken, die Segment-Anzeige wird irgendeine Ziffer zwischen 0 und 6 für 2 Sekunden statisch zeigen und dann zyklisch irgendeine Ziffer zwischen 0 und 6 schnell wieder zeigen.



Lektion 12 Punktmatrix-Anzeige antreiben mit 74HC595

Einführung

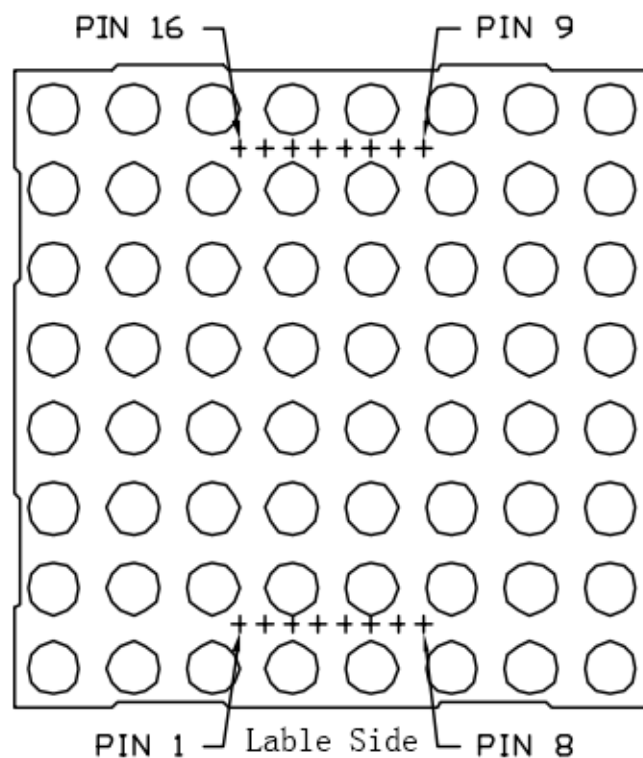
Mit dem Niederspannungsscanning, haben LED-Punktmatrix-Anzeigen viele Vorteile wie Strom einsparung, lange Lebensdauer, hohe Helligkeit, großen Blickwinkel, wasserdicht, kostengünstig, usw. LED-Punktmatrix-Anzeigen können den Bedarf an verschiedenen Anwendungen decken und haben somit eine breite Entwicklungsperspektive. In dieser Lektion werden wir lernen, wie mit 74HC595 eine LED-Punktmatrix anzutreiben.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 2* 74HC595
- 1* Punktmatrix-Anzeige
- Jumper Wire

Versuchsprinzip

Die Außenansicht einer Punktmatrix ist wie folgt:



Definition der Pins:

Zeilen- und Spaltennummerierung zunächst definieren (nur für die Punktmatrix mit dem Schluss „BS“)

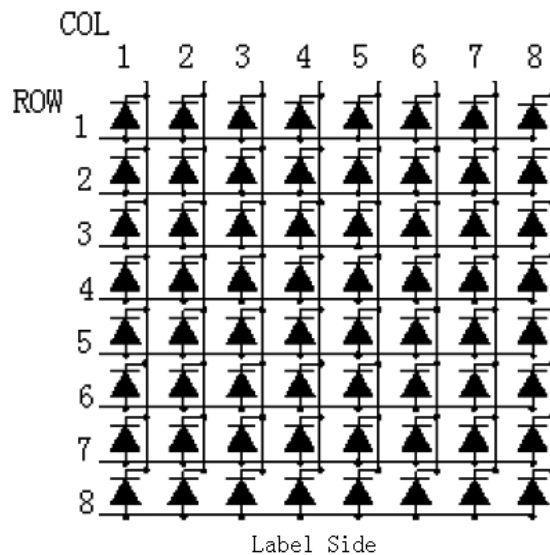


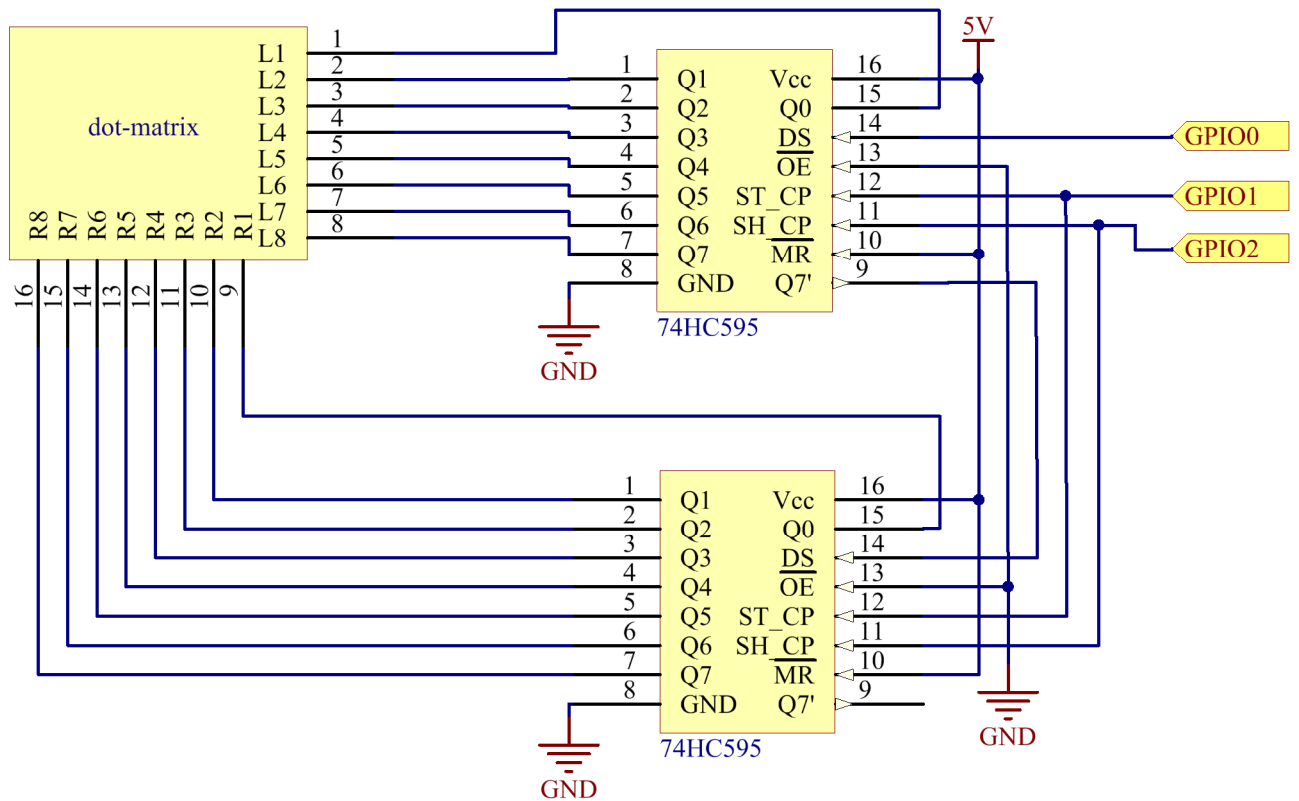
Tabelle für die Pin-Nummer in Zeilen und Spalten:

Spalten	1	2	3	4	5	6	7	8
Pin Nr.	13	3	4	10	6	11	15	16
Zeilen	1	2	3	4	5	6	7	8
Pin Nr.	9	14	8	12	1	7	2	5

Das Anzeigenprinzip der 8 * 8 Punktmatrix:

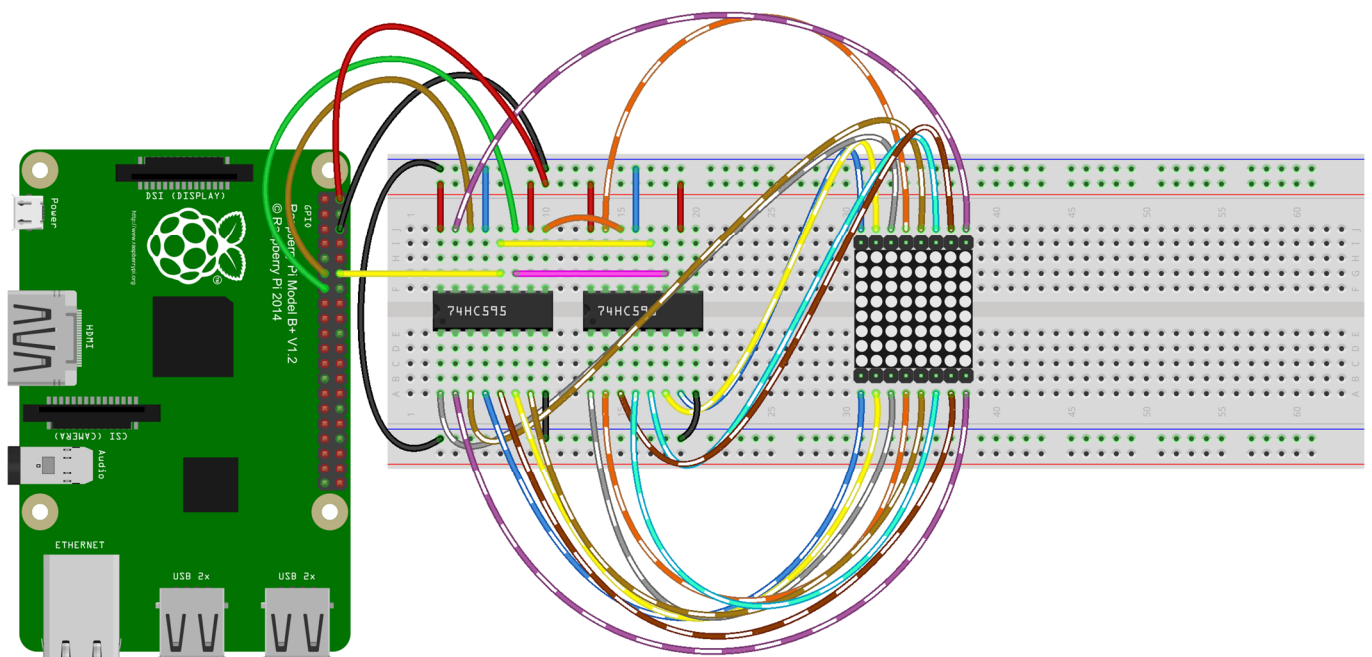
Die 8 * 8 Punktmatrix besteht aus vierundsechzig LEDs und jede LED ist an den Kreuzungspunkt einer Zeile und einer Spalte gesetzt. Wenn der elektrische Pegel einer bestimmten Zeile hoch ist und der einer bestimmten Spalte niedrig ist, wird die entsprechende LED am Kreuzungspunkt aufleuchten; wenn wir die LED am ersten Punkt anmachen wollen, lassen wir die Zeile 1 beim hohen Pegel und die Spalte 1 beim niedrigen Pegel. Wenn wir die LEDs auf der Zeile 1 anmachen wollen, lassen wir die Zeile 1 beim hohen Pegel und die Spalte (1, 2, 3, 4, 5, 6, 7, 8) beim niedrigen Pegel. Wenn wir die LEDs auf der Spalte 1 anmachen wollen, lassen wir die Spalte 1 beim niedrigen Pegel und die Zeile (1, 2, 3, 4, 5, 6, 7, 8) beim hohen Pegel.

Die Funktionen vom 74HC595 wurden zuvor erläutert. Ein Chip ist verwendet, um die Zeilen der Punktmatrix anzusteuern, und der andere ist verwendet, um die Spalten anzusteuern.



Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/12_DotMatrix/
```

Schritt 3: Kompilieren

```
gcc dotMatrix.c -o dotMatrix -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./dotMatrix
```

Für Benutzer der Python-Sprache:

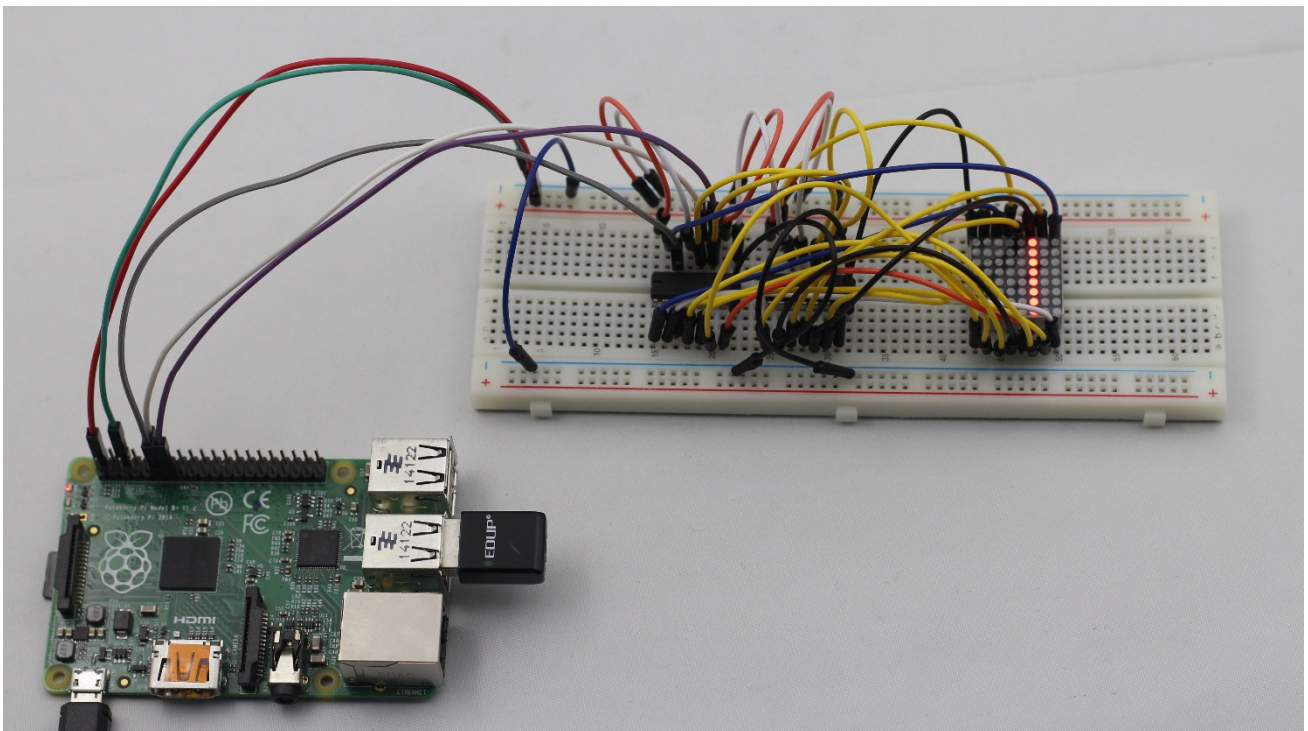
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 12_DotMatrix.py
```

Jetzt können Sie sehen, dass die LEDs aufleuchten.



Lektion 13 LCD1602

Einführung

In dieser Lektion werden wir lernen, wie mit dem LCD1602 die Zeichenketten anzuzeigen. Vor dem Versuch brauchen Sie die bereitgestellten Stiftleisten auf das LCD1602 aufzulöten.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* LCD1602
- 1* Potentiometer
- Jumper Wire

Versuchsprinzip

Das LCD1602, oder Zeichentyp LCD1602, ist ein Punktmatrix-LCD-Modul, speziell verwendet bei der Anzeige von Buchstaben, Zahlen, Symbolen und so weiter. Es ist zusammengesetzt von vielen $16 * 2$ Punktmatrizen, und jede davon besteht aus dem Zeichen-Bit $5 * 7$ oder $5 * 11$. Jedes Zeichen-Bit kann ein Zeichen anzeigen. Es gibt einen Punktraum zwischen jedem benachbarten Zeichen-Bit. Auch gibt es einen Punktraum zwischen jeder Zeile. Der Punktraum dient als Zeichenraum oder Zeilenraum; deshalb kann das LCD1602 Grafiken nicht sehr gut anzeigen.

Das LCD1602 verwendet die Standard-Schnittstelle von 16-Pin, unter denen sind:

Pin 1 (GND): Mit Masse verbunden;

Pin 2 (Vcc): Mit positiver Stromversorgung 5V verbunden;

Pin 3 (Vo): Angewendet für die Kontrastregelung vom LCD1602; der Pegel ist am niedrigsten, wenn es mit positiver Stromversorgung verbunden ist; und am höchsten, wenn mit der Erde verbunden (Sie können ein 10K-Potentiometer verbinden, um seinen Kontrast bei der Verwendung des LCD1602 zu regeln);

Pin 4 (RS): Für Registerauswahl; es wählt das Datenregister wenn mit High-Pegel (1) versorgt, und wählt das Befehlsregister wenn mit Low-Pegel (0) versorgt;

Pin 5 (R/W): Signale lesen/schreiben; es liest Signale wenn mit High-Pegel (1) versorgt, und es schreibt Signale, wenn mit Low-Pegel (0) versorgt. In diesem Versuch brauchen Sie nur Daten an LCD1602 zu schreiben, deshalb verbinden Sie diesen Pin mit der Erde;

Pin 6 (E): Ein Freigabe-Pin; wenn mit Low-Pegel versorgt, wird das LCD-Modul relevante Anweisungen ausführen;

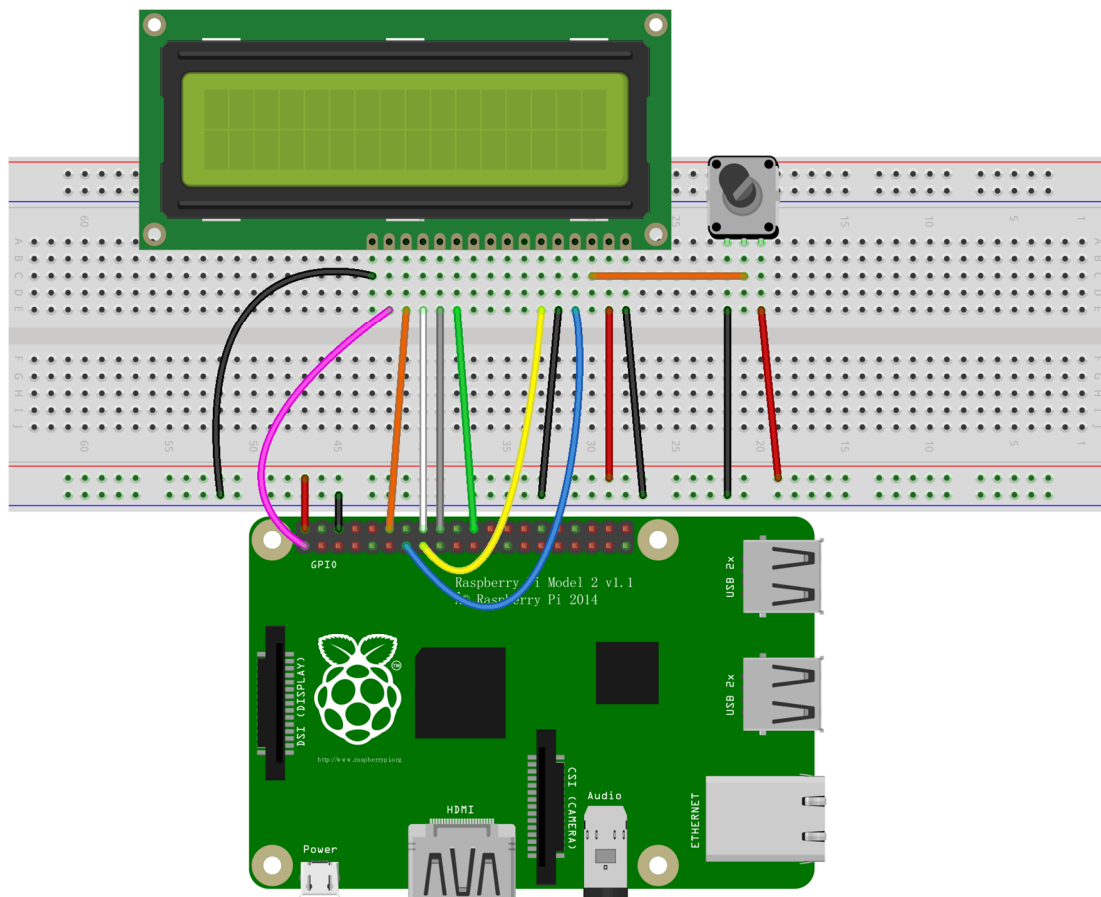
Pin 7 (D0-D7): Pins, die die Daten lesen und schreiben;

A und K: Die Stromquelle der LCD-Hintergrundbeleuchtung

Das LCD1602 hat zwei Betriebsmodi: 4-Bit und 8-Bit. Wenn die IOs des Mikroprozessors(MCU) ungenügend sind, können Sie den 4-Bit-Modus wählen, daunter sind nur die Pins D4 ~ D7 angewendet. Nach der Schaltungsverbindung, können Sie das LCD1602 durch Raspberry Pi bedienen.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:



fritzing

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/13_LCD1602/
```

Schritt 3: Kompilieren

```
gcc lcd1602_2.c -o lcd1602_2 -lwiringPiDev -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./lcd1602_2
```

Für Benutzer der Python-Sprache:

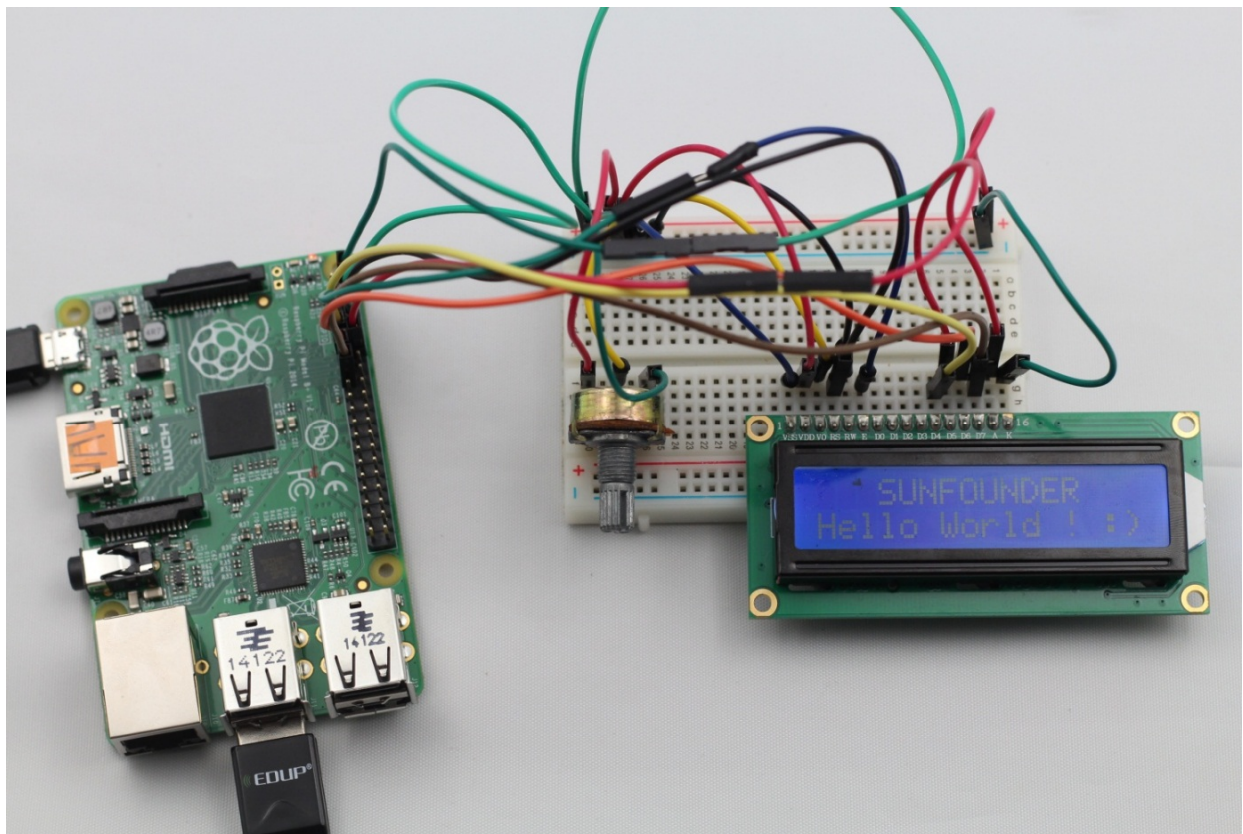
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python 13_lcd1602.py
```

Sie können sehen, dass zwei Zeilen von Zeichen auf dem LCD1602 angezeigt werden. Die realistische Wirkung ist wie folgt:



Erweiterung

In diesem Versuch wurde das LCD1602 im 4-Bit-Modus angetrieben. Sie können die Programmierung selbst ändern, um es im 8-Bit-Modus anzutreiben.

Lektion 14 ADXL345

Einführung

In dieser Lektion werden wir lernen, wie den Beschleunigungssensor ADXL345 anzuwenden.

Komponenten

- 1* Raspberry Pi
- 1* Steckplatine
- 1* ADXL345 Modul
- Jumper Wire

Versuchsprinzip

Der ADXL345 ist ein digitaler Beschleunigungsmesser mit sehr geringem Stromverbrauch und einer einstellbaren Ausgangsdatenrate von 0,1 Hz bis 3,2 kHz. Er bietet die Messbereiche $\pm 2/4/6/16g$ an, die Auflösung beträgt 13 bit. Der ADXL345 misst dynamische Beschleunigungen von Bewegungen oder Erschütterungen und ist mit seinen 10.000 g Stoßfestigkeit gut für Anwendungen mit hohen Beschleunigungen z. B. in PC-Festplatten geeignet.

Der neue Beschleunigungsmesser hat eine Auflösung von 4-Milli-g/LSB (niedrigstwertige Bitposition) über alle g-Bereiche, Bewegungserkennung bei Einzel- und Doppel-Tippen, Aktivität und Inaktivität, Freifall-Erkennung und programmierbare Schwellenwerte. Er bietet auch I2C, eine digitale 3-polige und eine 4-polige SPI-Schnittstelle und einen Spannungsbereich von 1,8 V bis 3,6 V. Bei einer typischen Versorgungsspannung von 2,5 V nimmt der ADXL345 einen Strom zwischen 25 bis 130 μA auf. Der Bewegungssensor ADXL345 enthält einen On-Chip-FIFO-(First-in/First-Out)-Speicherblock, der bis zu 32 Abtastsätze speichert. In diesem Versuch wird die I²C digitale Schnittstelle verwendet.

Versuchsverfahren

Schritt 1: Schaltung verbinden wie folgt:

ADXL345 Modul	Raspberry Pi
GND	GND
3.3V	3.3V
SCL0	SCL
SDA0	SDA
CS	3.3V
SDO	GND

Die I²C-Schnittstelle ist im Programm unten verwendet. Bevor Sie das Programm ausführen, stellen Sie bitte sicher, dass das Treibermodul I²C des Raspberry Pi normal geladen hat.

Für Benutzer der C-Sprache:

Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_C_code_for_RaspberryPi/14_ADXL345/
```

Schritt 3: Kompilieren

```
gcc adxl345.c -o adxl345 -lwiringPi
```

Schritt 4: Ausführen

```
sudo ./adxl345
```

Für Benutzer der Python-Sprache:

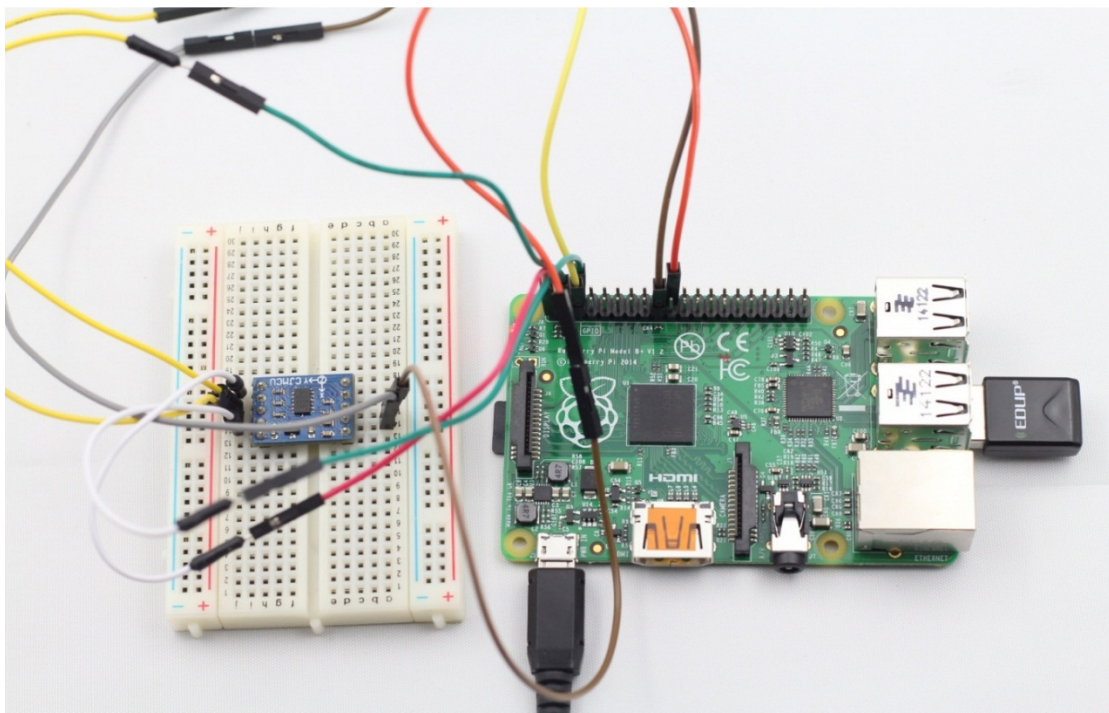
Schritt 2: Directory umstellen

```
cd /home/pi/Sunfounder_SuperKit_Python_code_for_RaspberryPi/
```

Schritt 3: Ausführen

```
sudo python Adafruit_ADXL345.py
```

Nun drehen Sie den Beschleunigungssensor. Sie können sehen, dass sich die Werte, die auf dem Bildschirm gedruckt werden, ändern.



Weiterte Anwendung des Raspberry Pi – Auf Basis von Web eine LED ansteuern

Einführung

Der Grund, warum Raspberry Pi ist so erstaunlich, ist seine Netzwerkfunktion und Unterstützung der zahlreichen Web-Anwendungen von Open Source. Damit können Sie Ihr eigenes System vom Internet der Dinge bequem aufbauen. In dieser Lektion werden Sie auf Basis von der Web mit Raspberry Pi eine LED ansteuern. Sie können Ihr Handy, Tablet-PC oder Computer verwenden, solange es im gleichen lokalen Netzwerk (LAN) mit Ihrem Raspberry Pi ist.

Komponenten

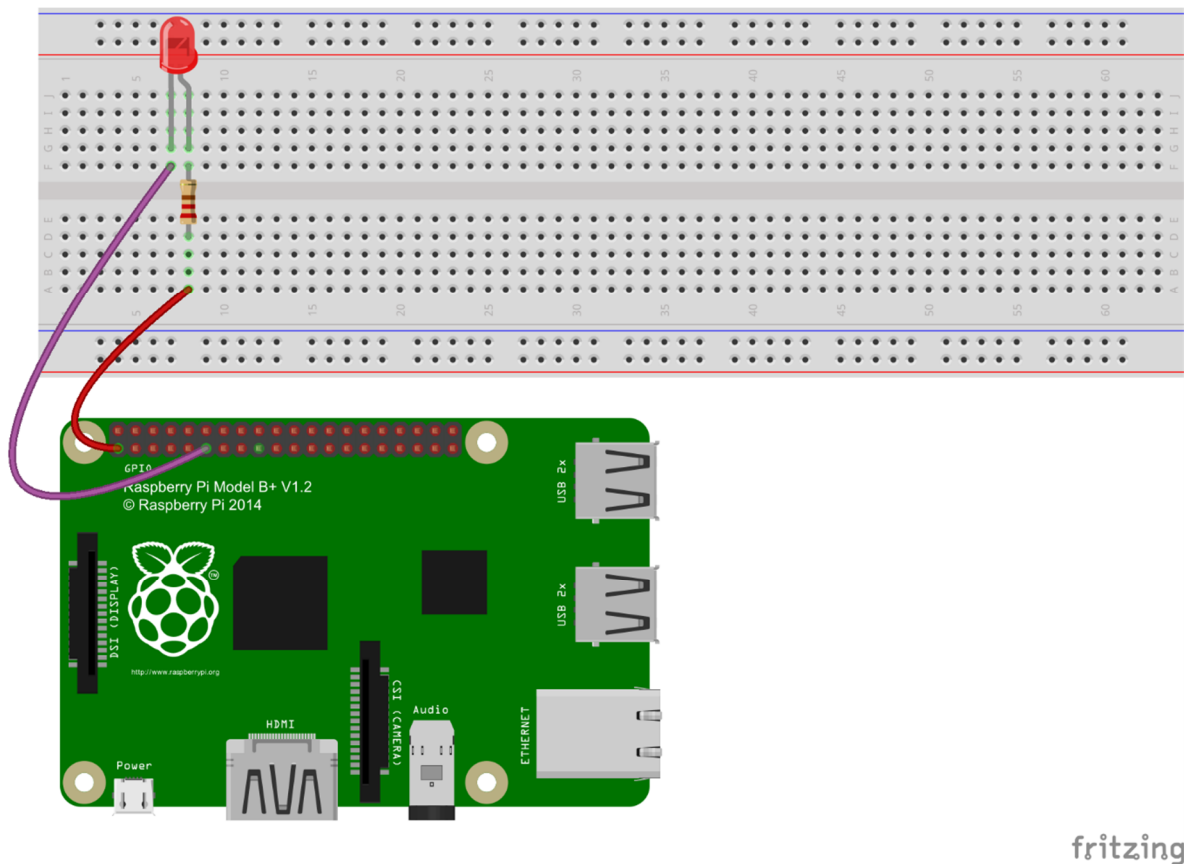
- 1* Raspberry Pi
- 1* Steckplatine
- 1* LED
- 1* Widerstand (220Ω)
- Jumper Wire

Versuchsprinzip

WebIOPi ist ein voll integrierter Rahmen vom Internet der Dinge für den Raspberry Pi. Es ist ein leichtes Web-Server-Programm auf der Raspberry Pi Plattform, die den Nutzern eine Lösung für den Fernzugriff zu Raspberry Pi bietet. Das heißt, Sie können aus der Ferne Ihren Raspberry Pi bedienen und die mit ihm verbundene Hardware durch einen Browser oder eine APP ansteuern.

Versuchsverfahren

1. Schaltung verbinden wie folgt:



- ## 2. WebIOPi installieren

```
wget http://webiopi.googlecode.com/files/WebIOPi-0.5.3.tar.gz
```

```
tar xvzf WebIOPi-0.5.3.tar.gz
```

```
cd WebIOPi-0.5.3
```

```
sudo ./setup.sh
```

- ### 3. Das Passwort von *WebIOPi* setzen oder ändern

```
sudo webiopi-passwd
```

Benutzername: webiopi

Das Standardpasswort ist raspberry.

- #### 4. WebIOPi starten

```
sudo python -m webiopi 8000
```

Sie können die Schnittstelle selbst ändern. Die Standardschnittstelle ist 8000.

5. `WebIOpi` im Backend-System ausführen, oder es wird beendet werden, wenn Sie `Ctrl + C` drücken

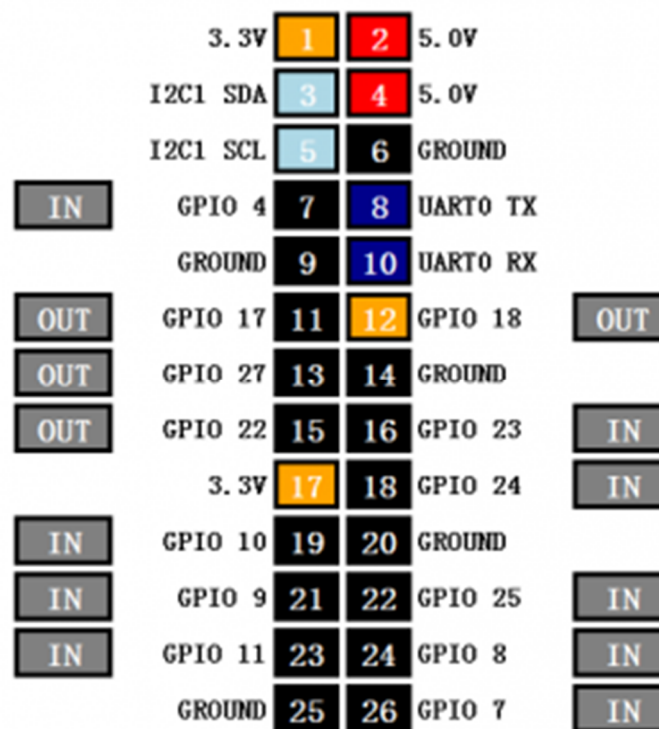
```
sudo /etc/init.d/webiopi start
and
sudo /etc/init.d/webiopi stop
```

- ## 6. WebIOPi setzen um mit dem System zu starten

```
sudo update-rc.d webiopi defaults
```

- ## 7. Die Schnittstelle der IP-Adresse-Zugriffsverwaltung öffnen

<http://192.168.1.106:8000/webiopi/>



Steuerverfahren:

- 1) Klicken Sie auf IN / OUT, um GPIOs zwischen dem Eingangs- und Ausgangsmodus zu wechseln.
- 2) Im Ausgangsmodus klicken Sie auf die Innenzahlen, um die Ausgänge zwischen hoch und niedrig zu wechseln. Nach Verbindung der LED mit GPIO17, klicken Sie auf Block 11, und werden Sie sehen, dass sich der Zustand der LED geändert hat.
- 3) Im Eingangsmodus repräsentieren die Innenzahlen die Eingangszustände des GPIOs

Zusammenfassung

In dieser Lektion haben Sie gelernt, wie über Raspberry Pi das Internet der Dinge für WebIOPi-Plattform aufzustellen und wie auf der Plattform Ihr Gerät aus der Ferne anzusteuern.

Copyright

Die vorliegenden Inhalte in dieser Anleitung sind urheberrechtlich geschützt. Alle Rechte sind vorbehalten. Mit Ausnahme der angebotenen Downloads zur Verwendung in der Originalfassung verstößt die Verwendung der Texte, Code und Abbildungen, auch auszugsweise, ohne die vorherige schriftliche Zustimmung von SunFounder gegen die Bestimmungen des Urheberrechts und ist damit rechtswidrig. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.