

Raspberry Pi als Stromquelle

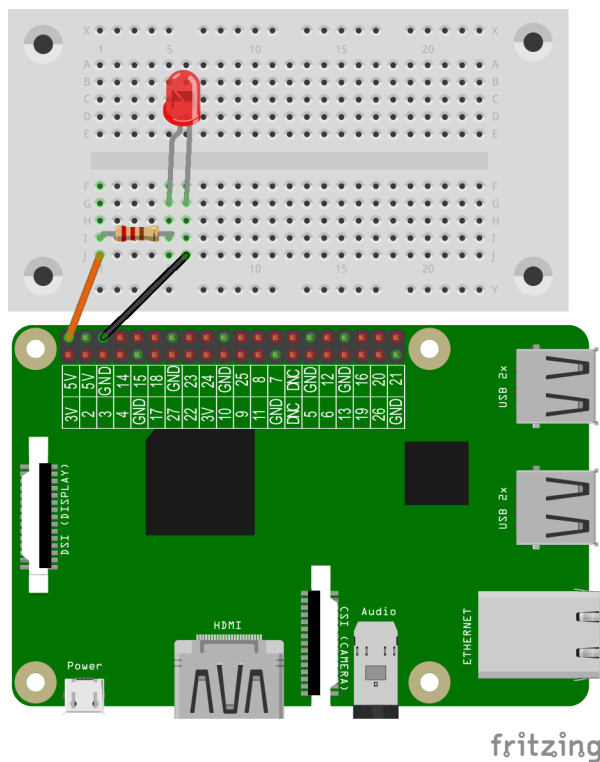
Material

- 1x LED
- 1x 220 Ohm Widerstand
- 2x Kabel
- 1x Steckbrett

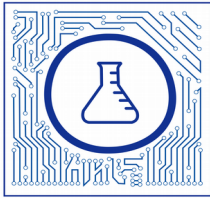
Wir nutzen den Raspberry Pi als Stromquelle. Für dieses Experiment musst du nichts programmieren.

Aufbau

Die LED wird mit dem Schutzwiderstand (am längeren Bein) am 5.5V - Pin angeschlossen. Das kurze Bein (GND) wird am GND-Pin angeschlossen. Wenn du alles richtig angeschlossen hast, leuchtet die LED.



Aufbau: Raspberry Pi als Stromquelle



LED mit einem Programm schalten

Material

- 1x LED
- 1x 220 Ohm Widerstand
- 2x Kabel
- 1x Steckbrett

Scratch: led.sb in deinem Projekt-Ordner für Scratch.

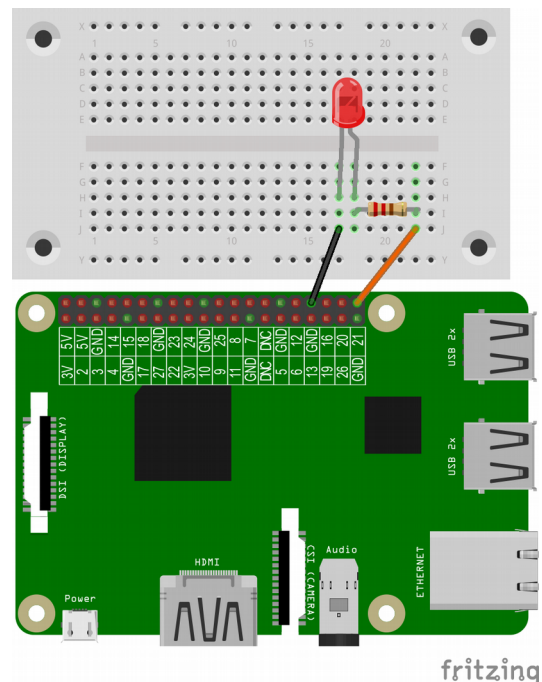
Python: led.py in deinem Projekt-Ordner für Python.

Scratch-Programm

In Scratch brauchst du für das Programmieren keinen Programmcode einzugeben. Die benötigten Blöcke ziehst du einfach per Drag-and-drop an die passende Stelle. Im linken Teil des Scratch-Fensters findest du, nach Themen geordnet, die verfügbaren Blöcke. Den Aufbau und die Funktionsweise von Scratch haben wir in der Einleitung beschrieben.

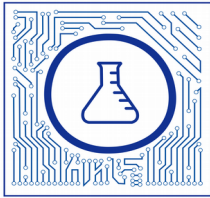


Scratch-Programm: LED mit einem Programm schalten



Aufbau: LED mit einem Programm schalten

Der Block **gpioserveron** startet den notwendigen GPIO-Server, damit das Programm die GPIO-Pins nutzen kann. Mit dem Block **config21out** wird der GPIO-Pin 21 als Ausgang definiert. An diesem GPIO-Pin ist die LED angeschlossen. Jeder GPIO-Pin kann entweder Ausgang oder Eingang sein. Eine fortlaufend wiederholende Schleife schaltet mit **gpio21on** die LED ein. Wartet zwei Sekunden und schaltet mit **gpio21off** die LED wieder aus. Wartet zwei Sekunden und die Schleife beginnt wieder vorne. Die Schleife wird endlos wiederholt bis der Benutzer das Programm anhält.



Python-Programm

```
import RPi.GPIO as GPIO #Import der notwendigen Bibliotheken
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False) #Warnungen abschalten
from time import sleep
LED_Pin = 21 #LED ist am GPIO-Pin 21 angeschlossen
GPIO.setup(LED_Pin, GPIO.OUT) #Pin21 wird als Ausgabe definiert
while True: #Schleife (Wiederholung)
    GPIO.output(LED_Pin, GPIO.HIGH) #LED ein schalten
    sleep(2) #2 sek warten
    GPIO.output(LED_Pin, GPIO.LOW) #LED aus schalten
    sleep(2) #2 sek warten
    #nun wird die LED nach 2 Sekunden wieder eingeschaltet...
```

Module und „import“-Funktion

Zusatzfunktionen sind in Form von Modulen in Python umgesetzt. Diese Module müssen vor ihrer Verwendung importiert werden. Der Befehl: **from time import sleep** importiert nur die Funktion **sleep** aus dem **time** Modul.

GPIO-Zugriff mit RPi.GPIO

Beim Import des GPIO-Moduls ist es ratsam, mit **as GPIO** ein Kürzel zu definieren.

So musst du nicht nicht beim jedem Zugriff auf die Funktion **RPi.GPIO** eingefügt werden muss.

```
import RPi.GPIO as GPIO
GPIO.setmode ( GPIO.BCM )
```

Mit dem Befehl **GPIO.setwarnings (False)** werden unschöne Warnungen ausgeblendet.

Für jeden verwendeten GPIO-Pin musst du mitteilen, wie du den GPIO-Pin im Skript nutzen möchtest. Dazu gibst du mit **setup** an, ob du den GPIO-Pin zur Ein- oder zur Ausgabe verwenden möchtest. In diesem Beispiel nutzen wir eine LED als Ausgabe:

```
GPIO.setup(LED_Pin, GPIO.OUT)
```

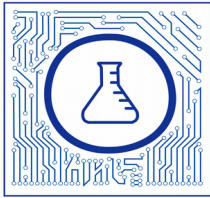
GPIO-Pin einschalten bzw. ausschalten

```
GPIO.output(LED_Pin, GPIO.HIGH) #LED ein schalten
GPIO.output(LED_Pin, GPIO.LOW) #LED aus schalten
```

Schleifen (while)

Schleifen können mit **while** formuliert werden. Die eingerückten Anweisungen werden dann so lange ausgeführt, wie die Bedingung erfüllt ist.

Hier: **while True:** Solange das Skript ausgeführt wird, ist die Bedingung erfüllt. Die eingerückten Anweisungen werden so lange ausgeführt, bis du **Strg + C** zum Abbruch drückst.



Ampel

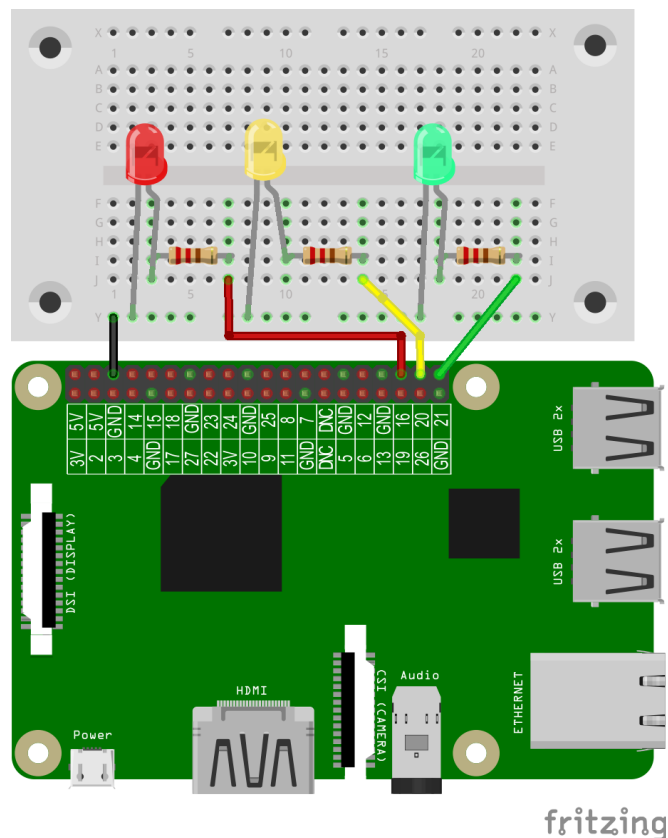
Material

- 1x LED rot
- 1x LED gelb
- 1x LED grün
- 3x 220 Ohm Widerstand
- 4x Kabel
- 1x Steckbrett

Scratch: [ampel.sb](#) in deinem Projekt-Ordner für Scratch.

Python: [ampel.py](#) in deinem Projekt-Ordner für Python.

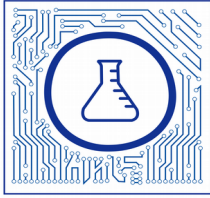
Scratch-Programm



Aufbau: Ampel

Der Block **gpioserveron** startet den notwendigen GPIO-Server, damit das Programm die GPIO-Pins nutzen kann. Mit den Blöcken **config16out** **config20out** **config21out** werden die GPIO-Pins 16, 20 und 21 als

Ausgang definiert. Am GPIO-Pin 16 ist die rote LED, am GPIO-Pin 20 ist die gelbe LED und am GPIO-Pin 21 die grüne LED, angeschlossen. Eine fortlaufend wiederholende Schleife schaltet nun die Ampel. Die Schleife wird endlos wiederholt bis der Benutzer das Programm anhält. **Frage:** Kannst du im Programm ablesen, wie genau die Ampel geschaltet wird? Nutze dabei dein Wissen zur Steuerung von LEDs aus dem letzten Arbeitsblatt.



Python-Programm

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False) #keine unnötigen Warnungen

from time import sleep

rot = 16
gelb = 20
gruen = 21

GPIO.setup(rot, GPIO.OUT)
GPIO.setup(gelb, GPIO.OUT)
GPIO.setup(gruen, GPIO.OUT)

while True:

    GPIO.output(rot, GPIO.HIGH) #rot an
    sleep(2) #2 sek warten

    GPIO.output(gelb, GPIO.HIGH) #gelb an
    sleep(1.5) #1,5 sek warten

    GPIO.output(gruen, GPIO.HIGH) #gruen an
    GPIO.output(rot, GPIO.LOW) #rot aus
    GPIO.output(gelb, GPIO.LOW) #gelb aus
    sleep(2) #2 sek warten

    GPIO.output(gruen, GPIO.LOW) #gruen aus
    GPIO.output(gelb, GPIO.HIGH) #gelb an
    sleep(2) #2 sek warten
    GPIO.output(gelb, GPIO.LOW) #gelb aus
```