

## LED mit einem Taster schalten

### Material

- 1x LED
- 1x 220 Ohm Widerstand
- 1x Taster
- 4x Kabel
- 1x Steckbrett

**Scratch:** `taster_led.sb` in deinem Projekt-Ordner für Scratch.

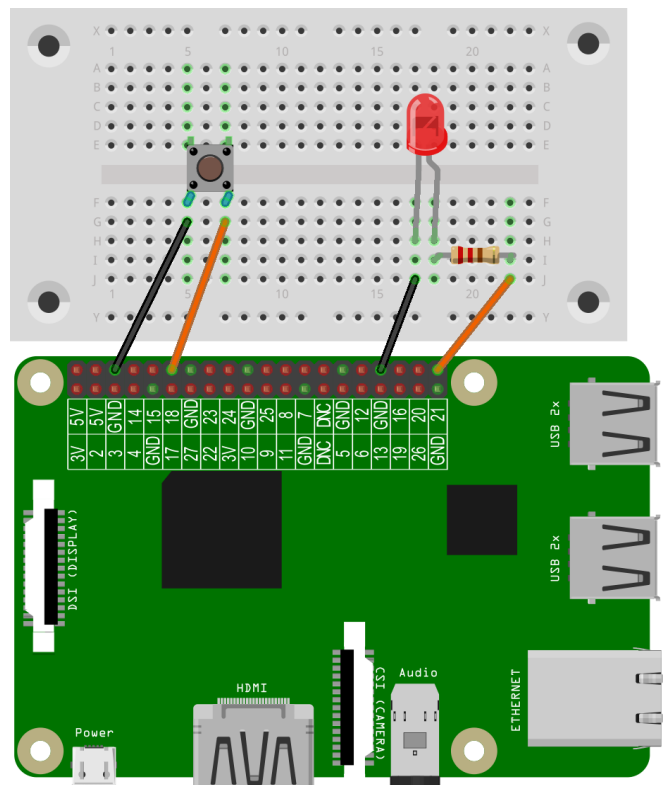
**Python:** `taster_led.py` in deinem Projekt-Ordner für Python.

Wir erweitern die Schaltung aus dem Schritt "Taster auslesen" um eine LED und einen Schutzwiderstand.

### Scratch-Programm



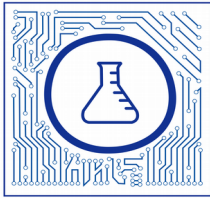
Der Block **gpioserveron** startet den notwendigen GPIO-Server, damit das Programm die GPIO-Pins nutzen kann. Mit dem Block **config21out** wird der GPIO-Pin 21 als Ausgang definiert. An diesem GPIO-Pin ist die LED angeschlossen. Mit **config18in** wird der GPIO-Pin für den Taster als Eingang definiert.



fritzing

Aufbau: LED mit einem Taster schalten

Eine fortlaufend wiederholende Schleife prüft, ob der Taster (**Wert von Sensor gpio18=0**) gedrückt wird. Wenn die Bedingung erfüllt ist, leuchtet mit **gpio21on** die LED. Sonst schaltet **gpio21off** die LED wieder aus.



### Python-Programm

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

from time import sleep

TasterPin = 18
LED_Pin = 21

GPIO.setup(TasterPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(LED_Pin, GPIO.OUT)

while True:
    if GPIO.input(TasterPin) == False:
        GPIO.output(LED_Pin, GPIO.HIGH)
    else:
        GPIO.output(LED_Pin, GPIO.LOW)
    sleep(0.2)
```

Die GPIO-Pins sind internen mit Pull-up- und Pull-down-Widerständen versehen. Das vereinfacht z. B. den Anschluss von Tastern. Wenn du den Pull-up-oder Pull-down-Widerstand an einem GPIO-Pin nutzen möchtest, dann musst im setup, wie im folgenden Beispiel, einen dritten Parameter mit angeben:

```
GPIO.setup(TasterPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

der dritter Parameter lautet: `pull_up_down=GPIO.PUD_UP` - steht für der Taster ist beim Start des Script nicht gedrückt.

Der Taster ist immer eine Eingabe, daher `GPIO.IN`.