

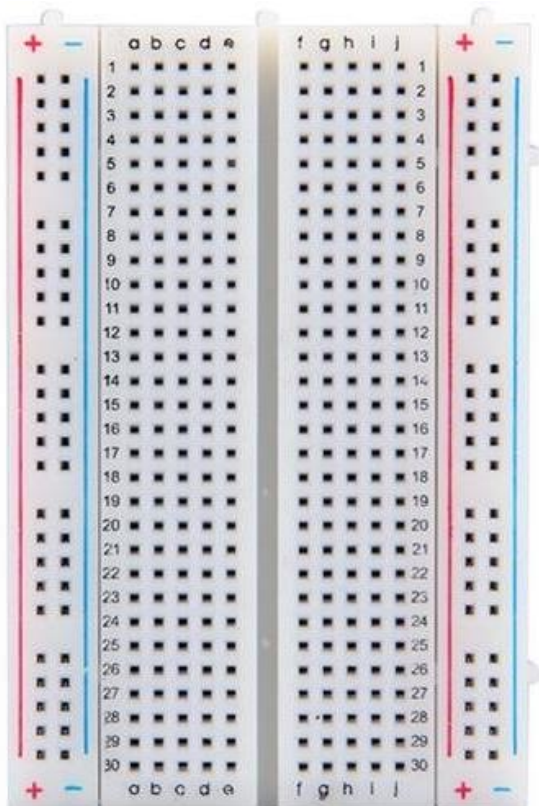
Umgang mit Breadboards

Breadboards oder Steckplatinen sind ideal für alle Elektronik-Projekte mit Arduino oder Raspberry Pi. Über Sogenannte Jumper wire (Steckbrücken) kann man Verbindungen von den GPIO-Schnittstellen zum Breadboard stecken. Dabei braucht man nichts zu löten und kann das gesteckte Projekt wieder ganz einfach demontieren.

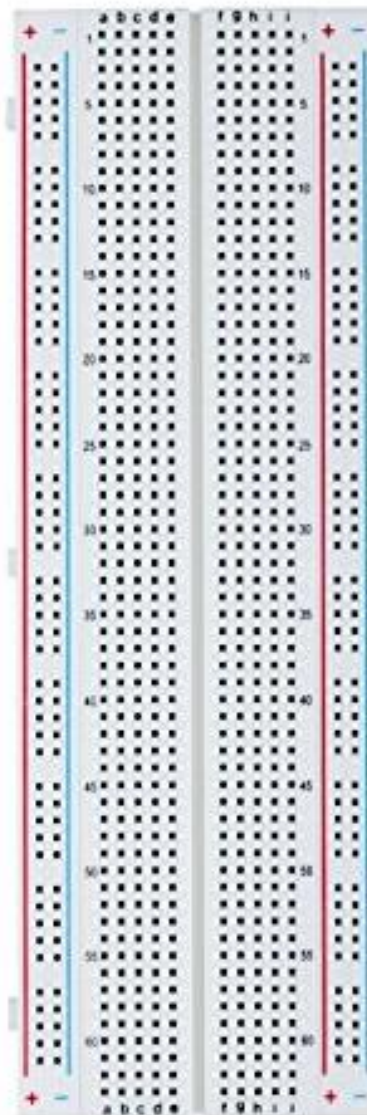
Arten:

Es gibt viele Arten von Breadboards, ideal sind solche mit 400 Kontakten, oder wer mehr Platz braucht, mit 830 Kontakten. Für Lötprojekte gibt es auch Breadboards in Form von Platinen.

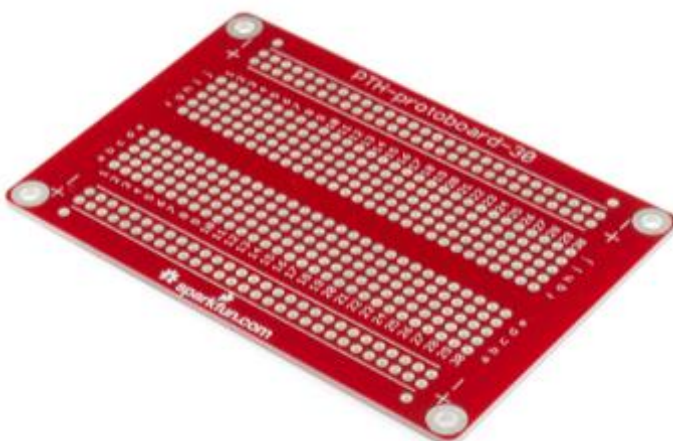
400 Steckplätze



830 Steckplätze

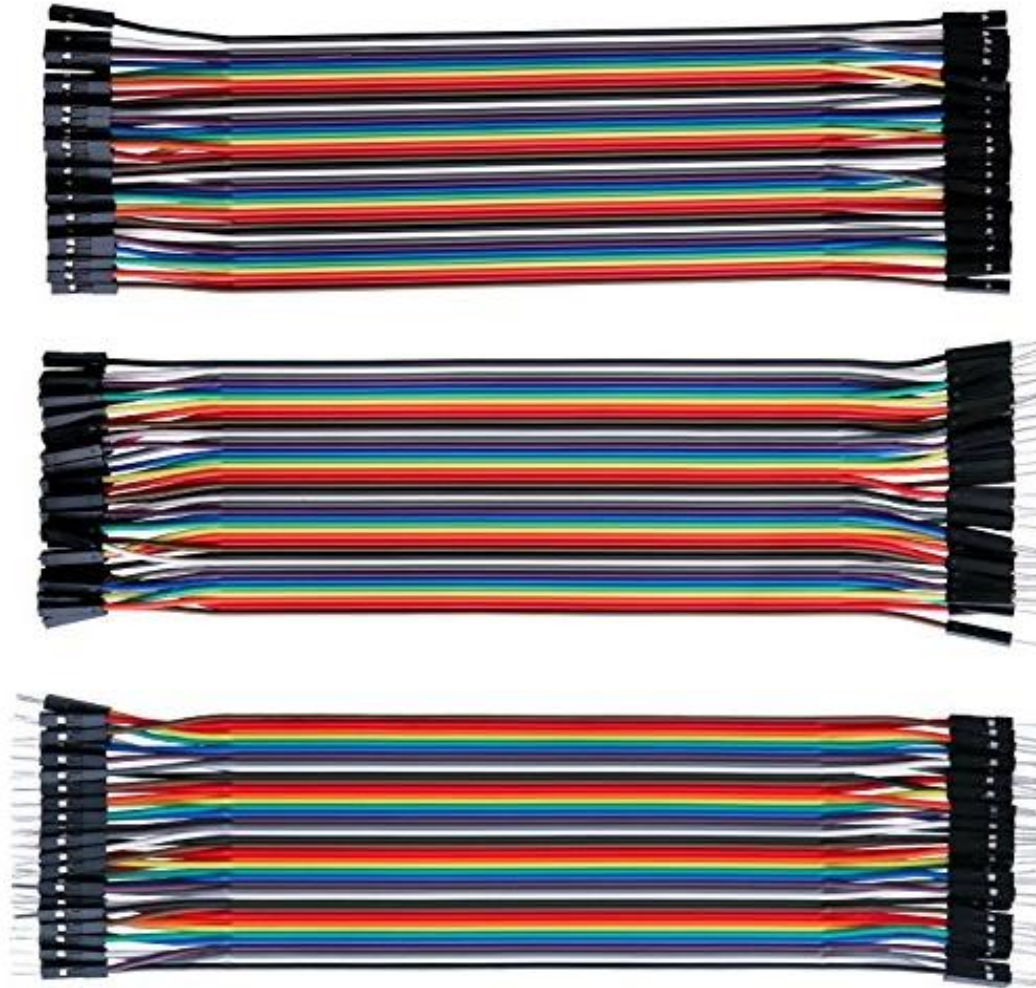


Soldering Breadboard



Kabel:

Mit Jumper Wire kann man stecken oder verlängern. Zum Stecken von den Pins (Arduino oder Raspi) zum Breadboard (Female – Male), zum Verlängern (Female – Female) und zum Überbrücken innerhalb des Breadboards (Male – Male).



Aufbau der Boards:

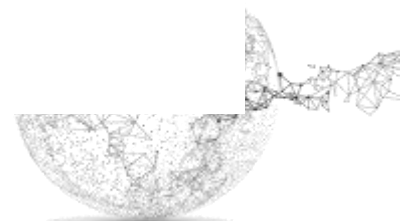
Trotz verschiedenen Ausführungen ist der grundlegende Aufbau dennoch immer identisch und unterteilt sich in folgende zwei Bereiche:

➤ Bus

An den beiden Aussenseiten der Breadboards befinden sich die sogenannten „Bus Stripes“. Dabei handelt es sich um Kontaktreihen für die Spannungsversorgung, die durchgehend vertikal miteinander verbundenen sind. Vom Bus werden die Schaltungen im Terminal-Bereich über Steckbrücken/Jumper mit Spannung versorgt.

Minus: blau / schwarz gekennzeichnet

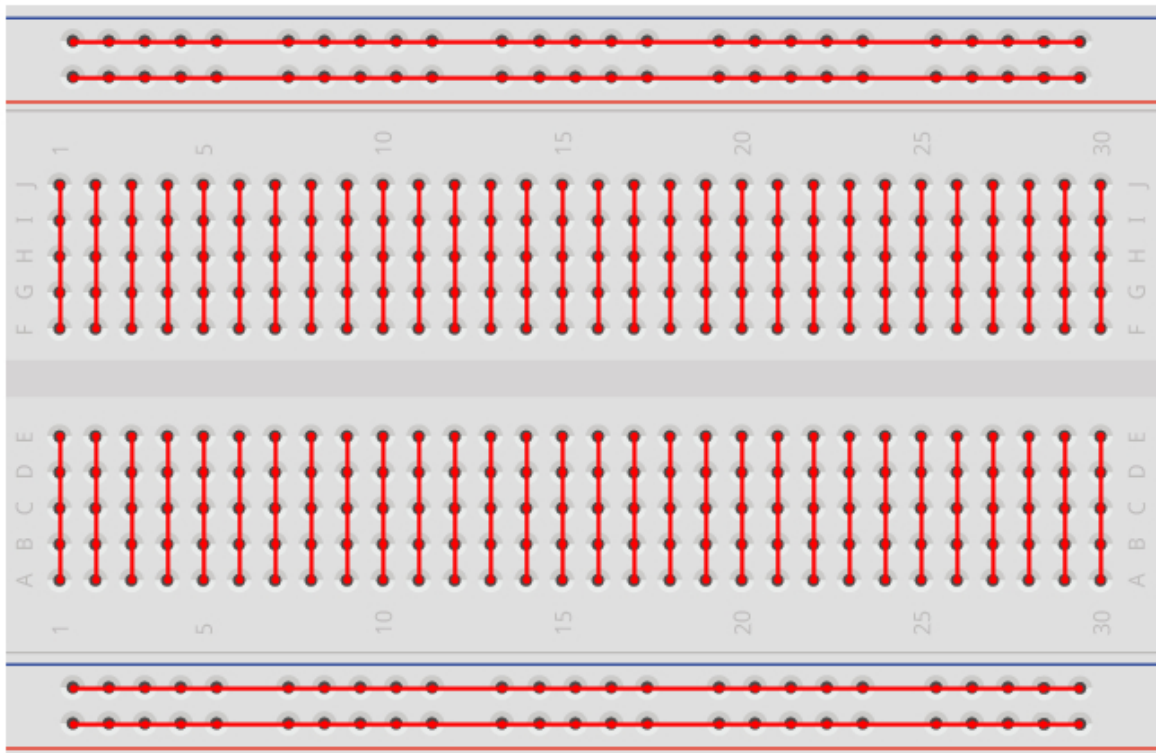
Plus: rot gekennzeichnet



➤ Terminal

Die Kontaktreihen, auf denen die eigentliche Schaltung aufgebaut wird, werden Terminal Stripes genannte. Sie sind immer in Gruppen horizontal miteinander verbunden, jeweils („a“ bis „e“) und („f“ bis „j“), und durch den Mittelsteg voneinander getrennt.

Durch die horizontale (Buchstaben) und vertikale (Zahlen) Beschriftung ergibt sich ein Koordinatensystem, welches in der Länge und Breite von Modell zu Modell variieren kann.



Bestellung:

Material bestellen kann man über Amazon oder über den chinesischen Ali Express (günstiger aber länger).

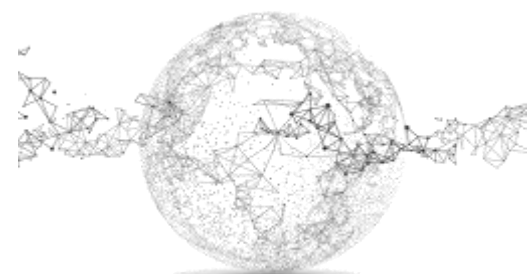
Beispiele:

Jumper Wire:

https://www.amazon.de/Elegoo-Female-Female-Male-Female-Steckbr%C3%BCcken-Drahtbr%C3%BCcken-Bunt/dp/B01EV70C78/ref=pd_bxgy_147_img_2?_encoding=UTF8&psc=1&refRID=58SCBT068KCNDCS1K5Y2

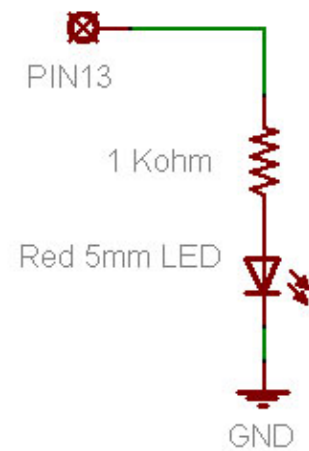
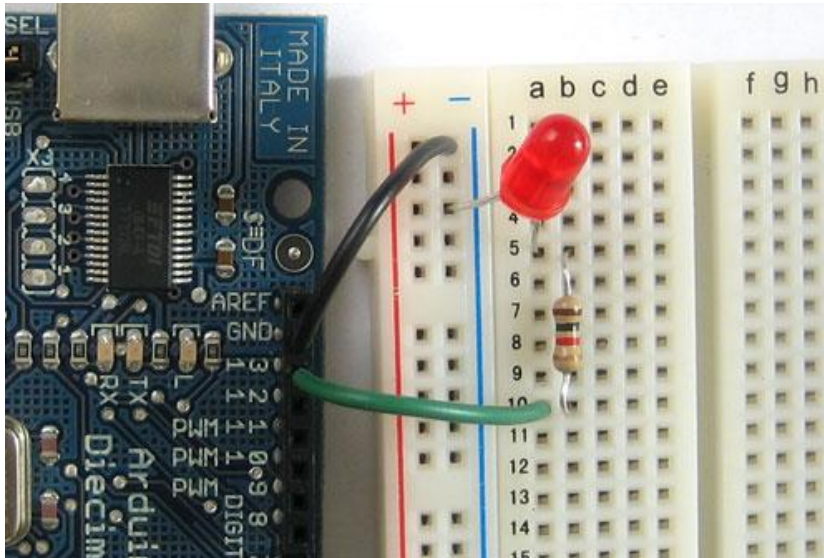
400er breadboard:

https://www.amazon.de/gp/product/B00PQC72ZS/ref=as_li_qf_sp_asin_tl?ie=UTF8&camp=1638&creative=6742&creativeASIN=B00PQC72ZS&linkCode=as2&tag=dde0b6-21



Eine Verbindung zum Arduino:

Hier sieht man eine Verbindung zum Arduino. An PIN 13 ist eine LED, die blinken soll.



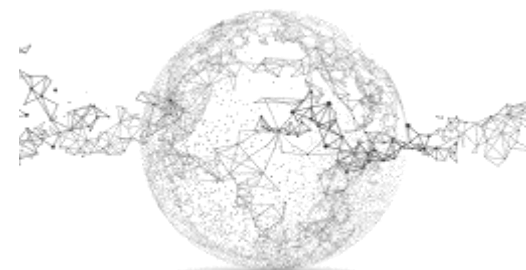
Der Strom fließt vom PIN 13 über einen Widerstand auf das lange Bein der LED (+). Das kurze Bein geht zum Minus und dann zu GND (-). Damit ist der Stromkreis geschlossen.

Arduino Code:

```
File Edit Sketch Tools Help
[Icons]
Blink $
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 * This example code is in the public domain.
 */

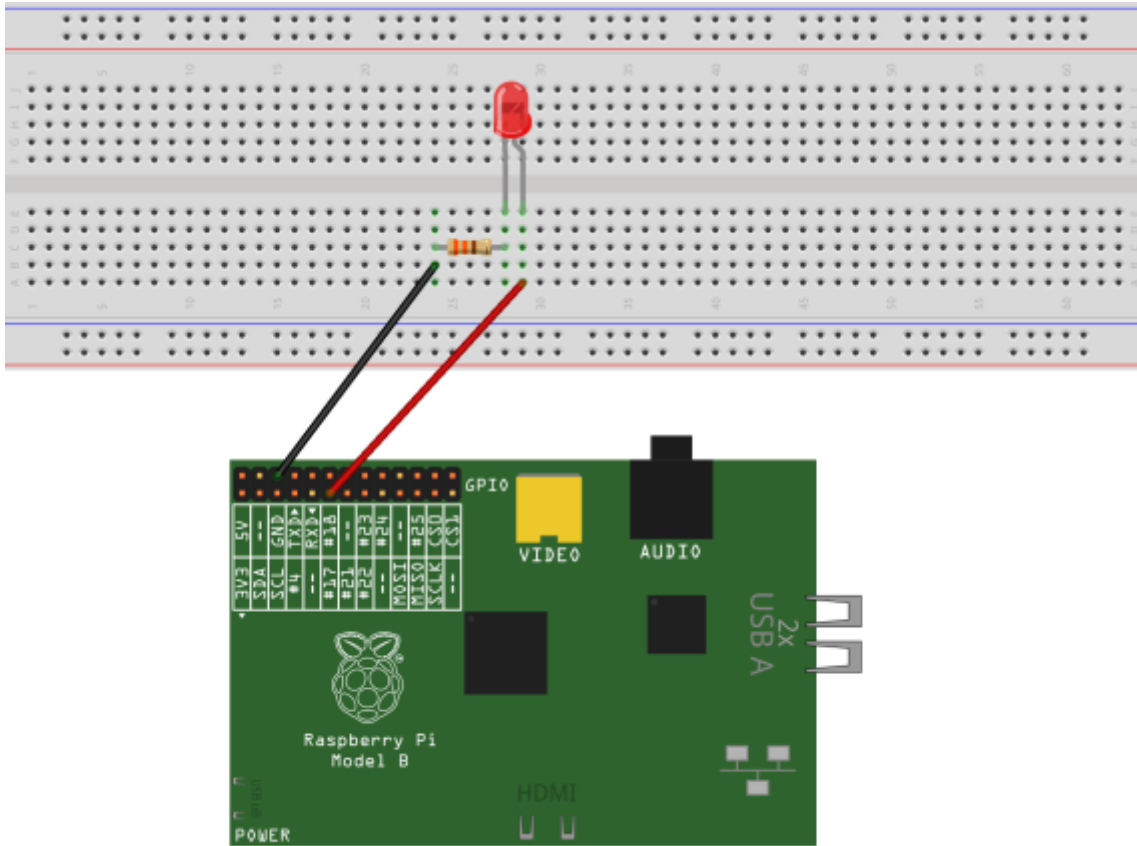
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```



Eine Verbindung zum Raspberry Pi:

Gleiches Beispiel wie beim Arduino, nun aber zu einem Raspberry Pi. Verbindung zum PIN 18. Unten zu sehen: Python Code



Python Code

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

LedPin = 11    # pin11

def setup():
    GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
    GPIO.setup(LedPin, GPIO.OUT)    # Set LedPin's mode is output
    GPIO.output(LedPin, GPIO.HIGH)  # Set LedPin high(+3.3V) to off led

def loop():
    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)

def destroy():
    GPIO.output(LedPin, GPIO.HIGH)    # led off
    GPIO.cleanup()                  # Release resource

if __name__ == '__main__':         # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:      # When 'Ctrl+C' is pressed, the child program destroy() will be executed.
        destroy()
```

Der Code für den PIN müsste hier noch von 11 auf 18 gewechselt werden.



Damit die PIN-Beschriftung einfacher ist, gibt es auch Breakout-Module, welche die Beschriftung auf dem Print haben:



https://www.amazon.de/Neuftech-T-Cobbler-Raspberry-Breadboard-Steckboard/dp/B0144HFO0A/ref=sr_1_3?s=computers&ie=UTF8&qid=1508309009&sr=1-3&keywords=breakout+raspberry+pi

