



Internet of Things

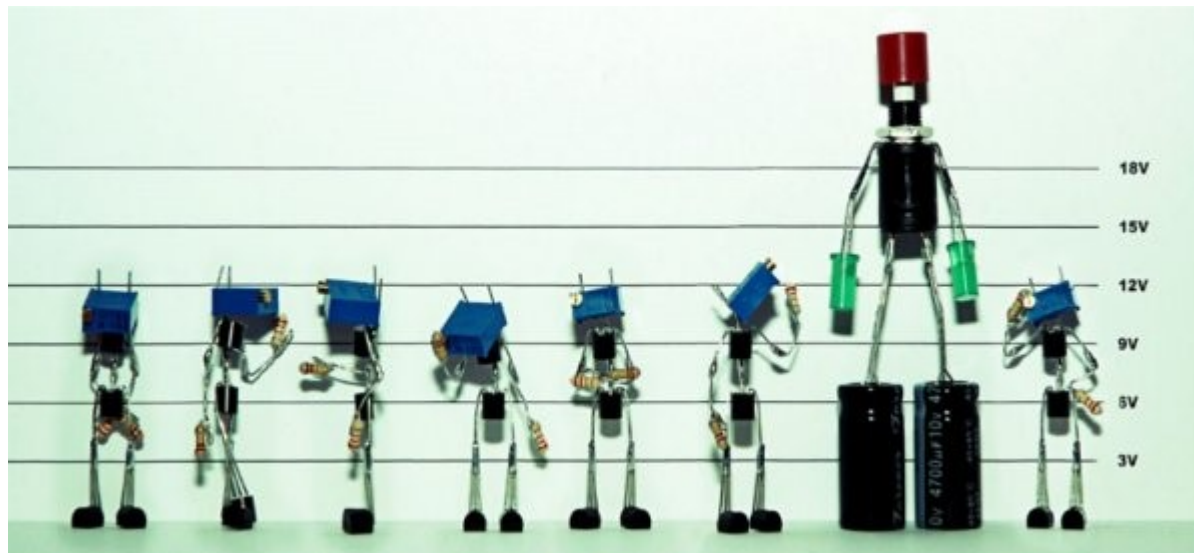
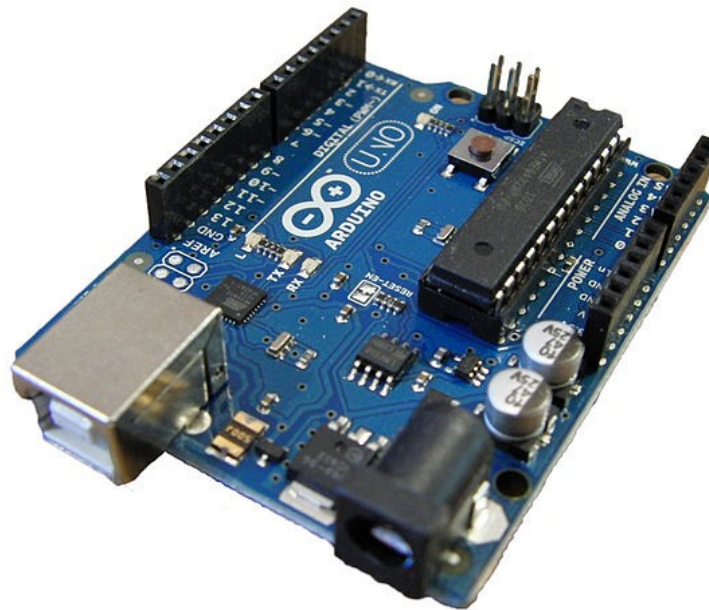


Photo by Lenny und Meriel ©

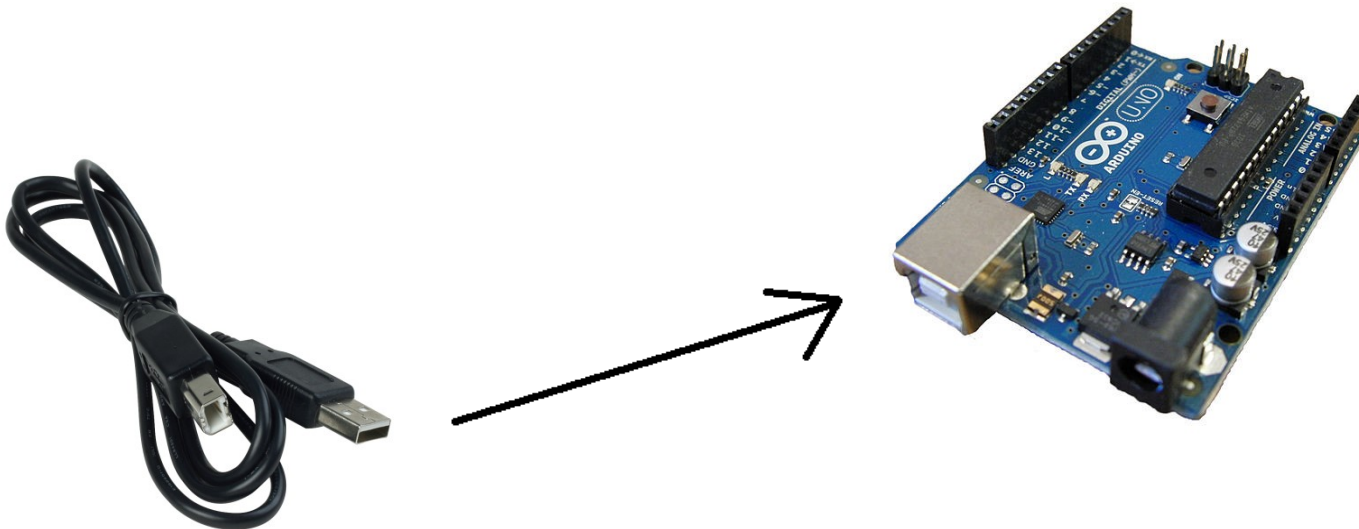


Öffnet die Box... und sucht dieses Teil:



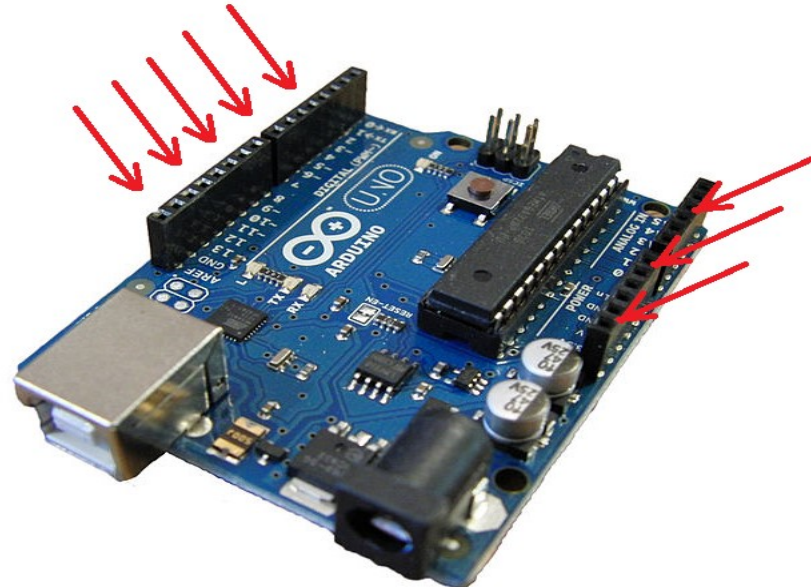


- > Man kann den Arduino anweisen, ein **Programm** auszuführen, das man zuvor auf einem Computer schreibt.
- > Um das Programm an den Arduino zu schicken, muss man ihn über ein **USB-Kabel** mit dem Computer verbinden.



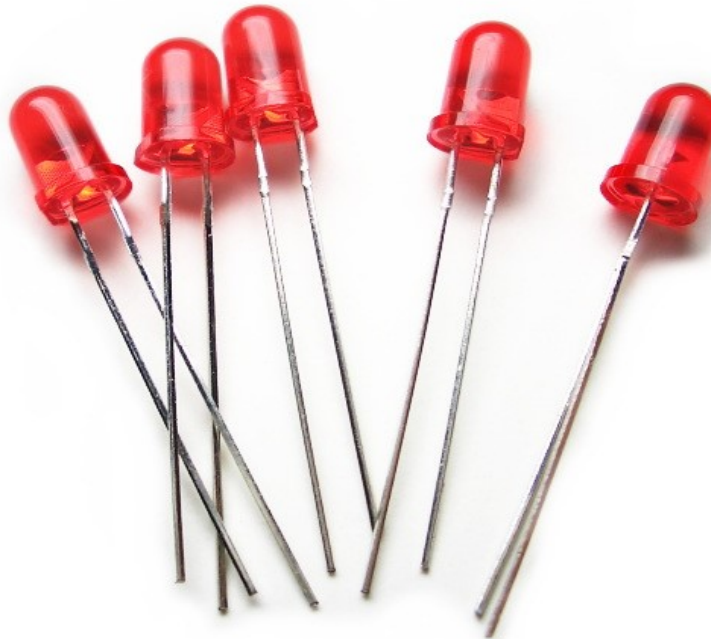


- > Über das USB-Kabel wird Arduino mit Strom versorgt. Man sagt, man legt eine **Spannung** an.
- > Arduino hat **Pins** (Löcher) an den Seiten, mit welchen er Strom empfangen oder versenden kann!



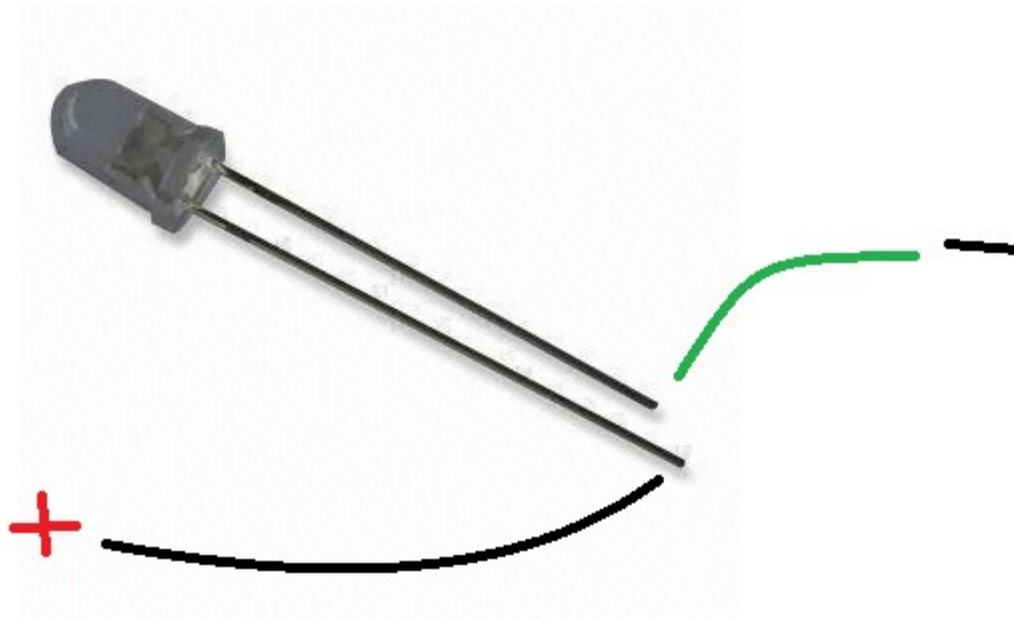


- > Eine **LED** ist ein kleines Licht, das häufig farbig ist (weiß/rot/gelb/grün).
- > Schaut Euch mal an, welche in Eurer Box sind!





- > Jede LED hat zwei **Beine**... ein kurzes und ein langes
- > Das lange ist der PLUS Pol (+) (der positive Pol)
- > Das kurze ist der MINUS Pol (-) (der negative Pol)
- > Der Strom fließt immer von + nach -





- > Arduino erhält also seinen Strom per USB (das heißt vom Computer).
- > Man misst die „Menge“ an Strom (die **Stromstärke**), die durch den Draht fließt, in **AMPERE** (Ampere ist sozusagen der *Liter* für den elektronischen Strom).
- > Die Einheit für den „Druck“ (die **Spannung**), der den Stromfluss ermöglicht, ist **VOLT**.
- > Eine LED braucht eine **Spannung** von 1,8 V (1,8 Volt).
- > Eine LED kann einen **Strom** der **Stärke** von 15 mA (15 Milliampere, 15 Tausendstel von 1 Ampere) fließen lassen.
- > Also muss man vorsichtig sein, wenn man Strom an eine LED schickt!!!



Ein **Widerstand** sorgt dafür, dass nicht zu viel Strom fließt. Der wirkt wie ein Flaschenhals... nur für die Elektrizität...





> All dies kann in dieser Formel zusammengefasst werden:

Stromstärke [Ampere] · **Widerstand** [Ohm] = **Spannung** [Volt]

> Folglich kann man den **Widerstand** berechnen, den man braucht:

Widerstand = **Spannung** / **Stromstärke**

> Für unsere LEDs heißt das:

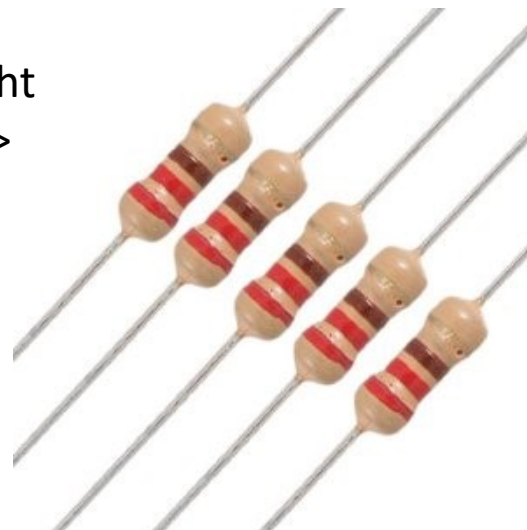
(5 – 1,8) Volt / 0,015 Ampere = 213 Ohm



Die **Widerstände** sind mit **farbigen Strichen** markiert. Diese sehen nicht nur *schick aus*, sondern **kodieren** auch die Stärke des **Widerstands** in Ohm, die sie haben.

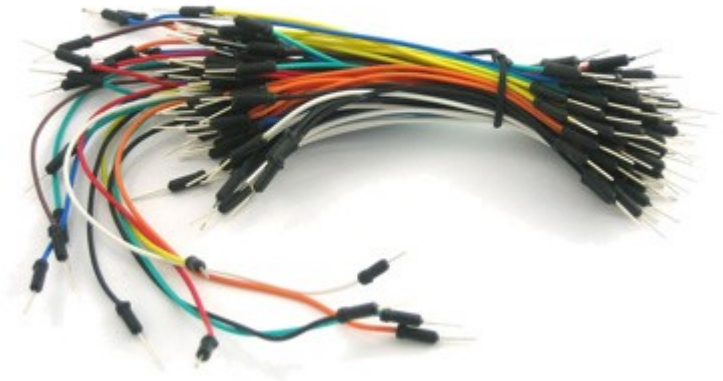
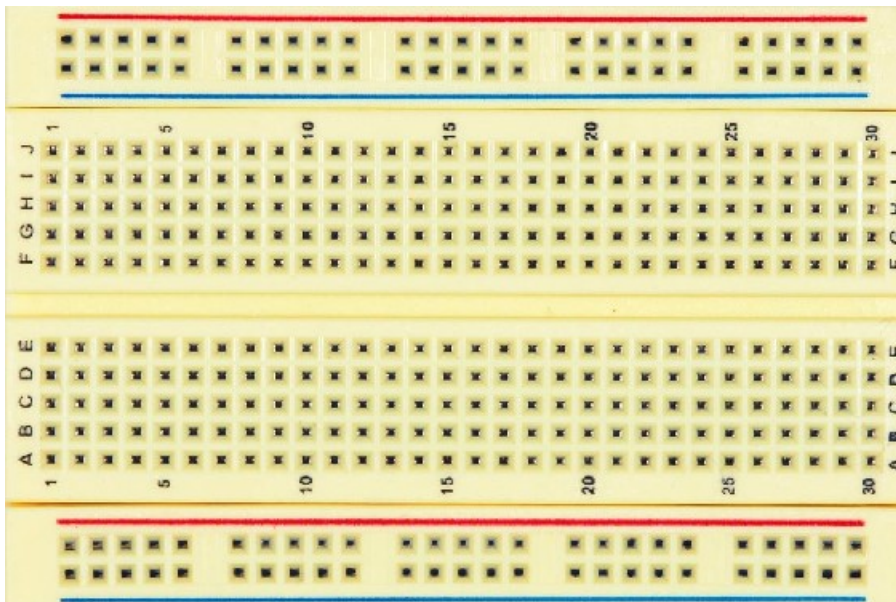
Für unsere LEDs benötigen wir 213 Ohm. Der Code **Rot-Rot-Braun** bedeutet einen **Widerstand** von 220 Ohm.

Weil es hier auf 7 Ohm mehr oder weniger nicht ankommt, werden wir diese hier verwenden ->



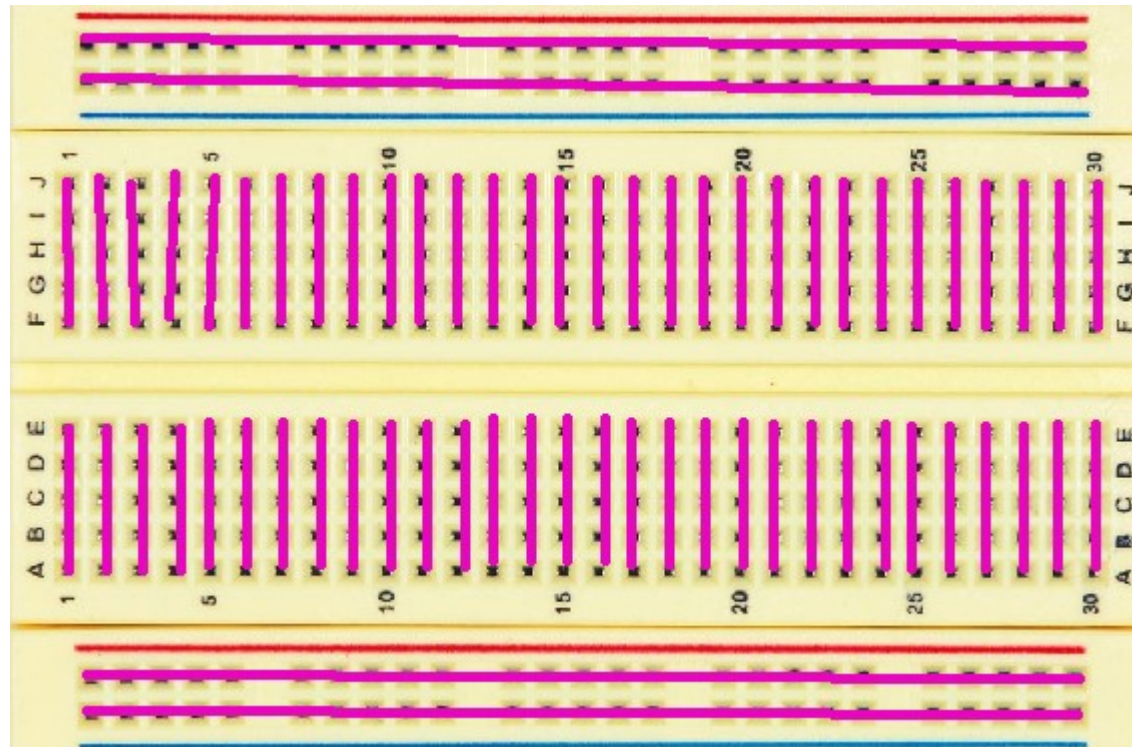


Die **Platine** und **ihre Kabel**... (Auf englisch sagt man „breadboard“, das heißt soviel wie Brotschneidebrett...)



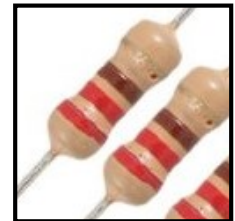
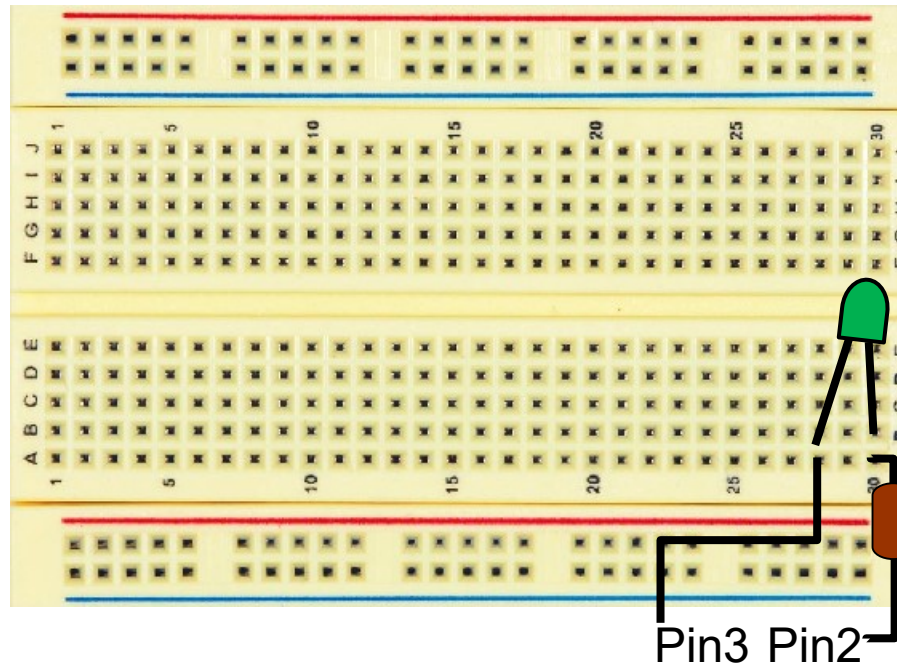


Ok, und jetzt? Wie funktioniert das nun?





Na dann los: Nun setzen wir das alles zusammen und bringen eine LED zum Leuchten!



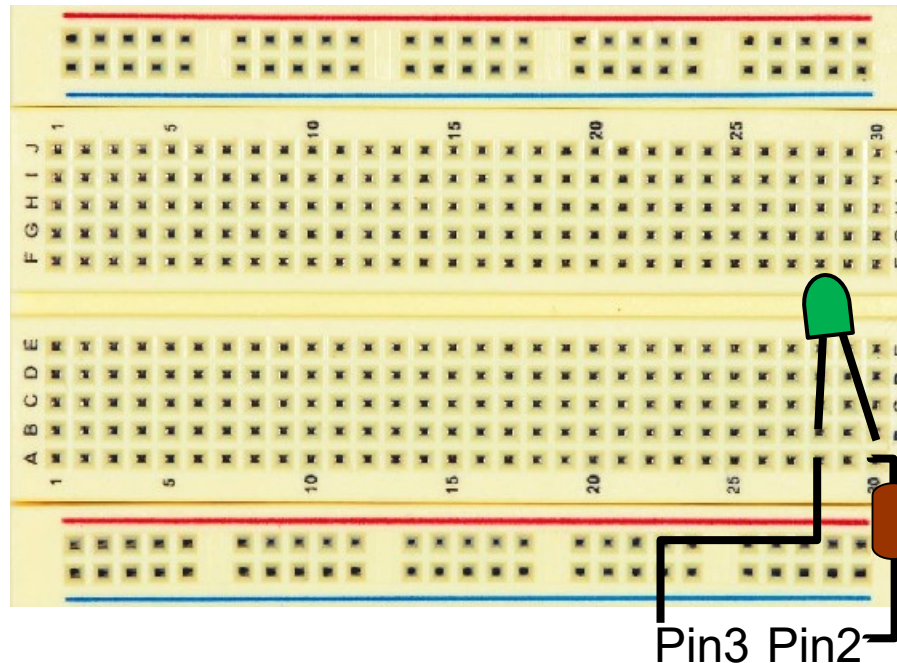


Nun schließen wir den Arduino mit dem USB-Kabel an den Computer an und laden die Software „00-eineLedLeuchten“ aus dem Ordner „workshop“ auf dem Desktop.

Nach dem Übertragen des Programms sollte die LED nun leuchten.

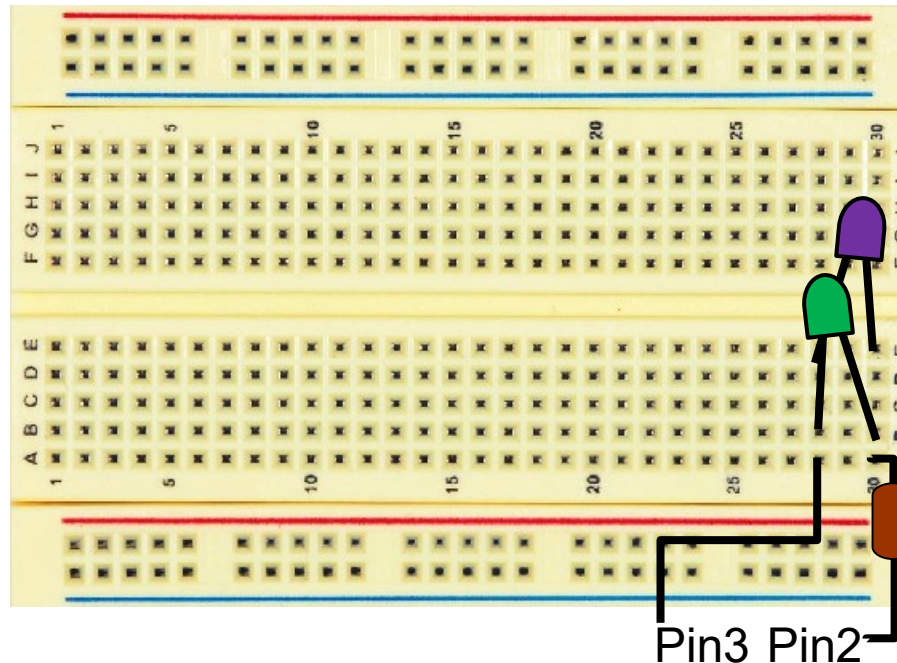


Was passiert, wenn wir die LED umdrehen? Probiert es doch mal aus!





Und nun die zweite LED (das lange Bein auf der anderen Seite):





Benutze das gleiche Programm wie gerade nochmal – was passiert?
Vertausche im Code überall die Zahlen 2 und 3 – was ändert sich?

Nun laden wir die Software „2LedsAbwechselndLeuchten“ aus dem Ordner „workshop“ auf dem Desktop auf den Arduino.

Nach dem Übertragen des Programms sollten beide LEDs im Wechsel blinken.

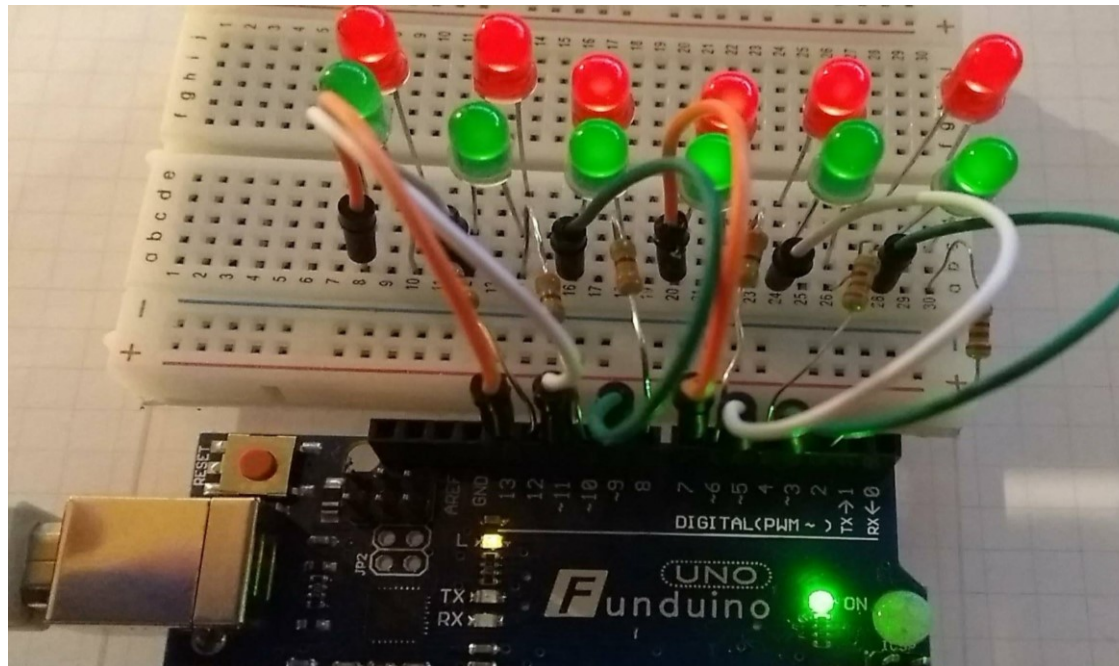
In Zeile 12 und Zeile 19 steht „delay(200)“.

Was passiert, wenn du hier eine andere Zahl einsetzt?

Können wir beide LEDs gleichzeitig zum Leuchten bringen?



Wir brauchen mehr LEDs: Die Schaltung von gerade brauchen wir insgesamt 6 mal. Die erste könnt ihr einfach so stehen lassen und die nächste kommt genauso daneben, verbunden mit Pin 4 und 5. Und dann geht es immer so weiter bis 12 und 13





Nun laden wir die Datei „Framework12Leds“ aus dem Ordner „workshop“ auf dem Desktop auf den Arduino.

Das ist ganz schön viel Quelltext, aber um den müsst ihr euch kaum kümmern.

An einigen Stellen stehen Kommentare (das ist alles, was hinter // in einer Zeile steht, und im Editor grau ist).

Die erklären euch, was ihr machen könnt. Die Schaltung bleibt immer die gleiche.