



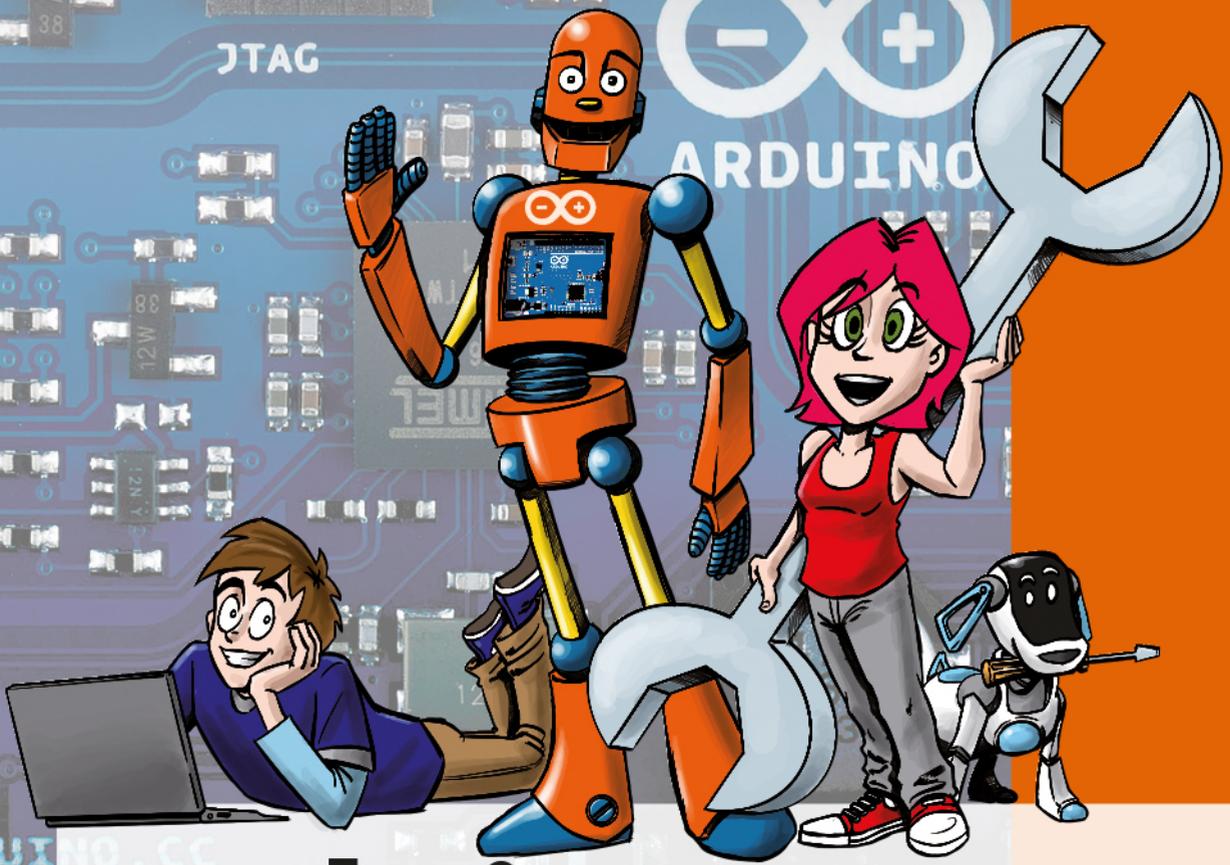
Erik  
Schernich  
2. Auflage

SDA AREF GND  
~13 ~12 ~11 ~10 ~9 ~8  
DIGITAL (PWM~)

JTAG



ARDUINO



# Arduino

FÜR **KIDS**



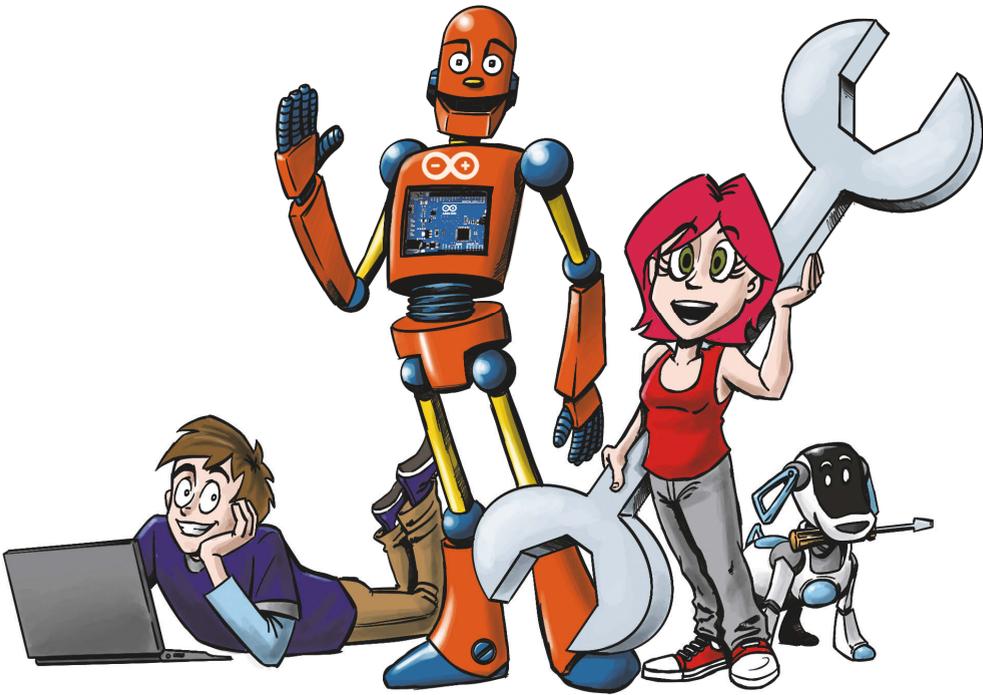
## **Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)**

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Erik Schernich



# Arduino für Kids

2. Auflage



mitp

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-95845-581-8

2. Auflage 2017

[www.mitp.de](http://www.mitp.de)

E-Mail: [mitp-verlag@sigloch.de](mailto:mitp-verlag@sigloch.de)

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2017 mitp-Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Katja Völpel

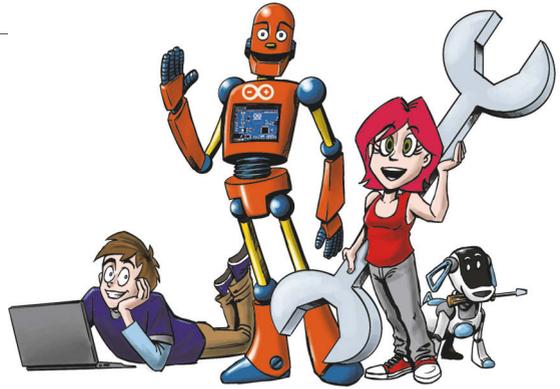
Fachkorrektur: Thomas Brühlmann

Sprachkorrektur: Petra Heubach-Erdmann

Covergestaltung: Christian Kalkert

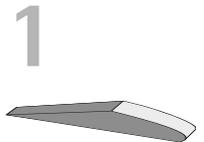
Bilder: Alle Schaltpläne wurden mit Fritzing erstellt.

Satz: Ill-satz, Husby, [www.drei-satz.de](http://www.drei-satz.de)

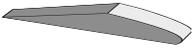


# Inhalt

<b>Einleitung</b> .....	<b>9</b>
Was ist ein Mikrochip? .....	9
Wie programmiert man? .....	10
Materialien .....	11
Der erste Anfang .....	12
Zusammenfassung .....	17
Ein paar Aufgaben .....	17
<b>Blinke, blinke, kleine LED</b> .....	<b>19</b>
Die Software installieren .....	20
Unser erstes Programm .....	22
Unser zweites Programm: Lichtspiel mit LEDs .....	26
Unser drittes Programm: Morsen .....	35
Unser viertes Programm: Der heiÙe Draht .....	38
Zusammenfassung .....	47
Ein paar Fragen ... .....	47
... und ein paar Aufgaben .....	47



2



<b>Der Arduino spricht</b> .....	<b>49</b>
Einen ersten Text senden .....	50
Zusammenfassung .....	57
Eine Frage ... ..	58
... und eine Aufgabe für heute: .....	58

3



<b>Sensoren – Schnittstellen zur Welt</b> .....	<b>59</b>
Ein Sensor, was ist das eigentlich? .....	59
LEDs leuchten lassen .....	60
Zusammenfassung .....	69
Ein paar Fragen ... ..	70
... und ein paar Aufgaben .....	70

4



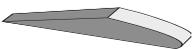
<b>Motoren – Bewegung mit dem Arduino</b> .....	<b>71</b>
DC-Motor – Spaß mit Umdrehungen .....	72
Den Motor effizient steuern .....	74
Servos .....	76
Zusammenfassung .....	81
Ein paar Fragen ... ..	82
... und ein paar Aufgaben .....	82

5



<b>Quellcode anderer Entwickler lesen</b> .....	<b>83</b>
Die Dokumentation .....	83
Der mysteriöse Quellcode .....	84
Zusammenfassung .....	87
Ein paar Fragen ... ..	87
... und eine Aufgabe .....	87

6



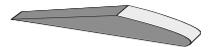
<b>LCD – Infos direkt am Arduino</b> .....	<b>89</b>
Was bedeutet eigentlich LCD? .....	90
Zusammenfassung .....	94
Ein paar Fragen ... ..	94
... und ein paar Aufgaben .....	94

<b>Der Arduino und das Multimeter</b> .....	<b>95</b>
Zum Anfang eine Anekdote .....	96
Welche Anzeige möchten wir haben? .....	96
Wir untersuchen das Poti .....	101
Durchgangsmessung .....	103
Zusammenfassung .....	104
Ein paar Fragen ... ..	105
... und ein paar Aufgaben .....	105
<b>Arduino online</b> .....	<b>107</b>
HTML – Das Tor zum Internet .....	107
»Netz, wir brauchen Netz« .....	109
Zusammenfassung .....	112
Ein paar Fragen ... ..	113
... und ein paar Aufgaben .....	113
<b>Tastatur mit dem Arduino Leonardo</b> .....	<b>115</b>
Erste Schritte mit dem Leonardo .....	116
Die erste kleine Tastatur .....	119
Ein Sicherheitstoken .....	121
Zusammenfassung .....	123
Ein paar Fragen ... ..	123
... und ein paar Aufgaben .....	124
<b>Ein Blick hinter die IDE</b> .....	<b>125</b>
C++, das Herz des Arduinos .....	126
Ein Programm auf den Arduino übertragen .....	131
AVR-Programmieren .....	133
Zusammenfassung .....	137
Ein paar Fragen ... ..	138
... und ein paar Aufgaben .....	138

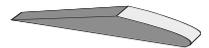
7



8



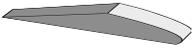
9



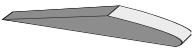
10



11



A



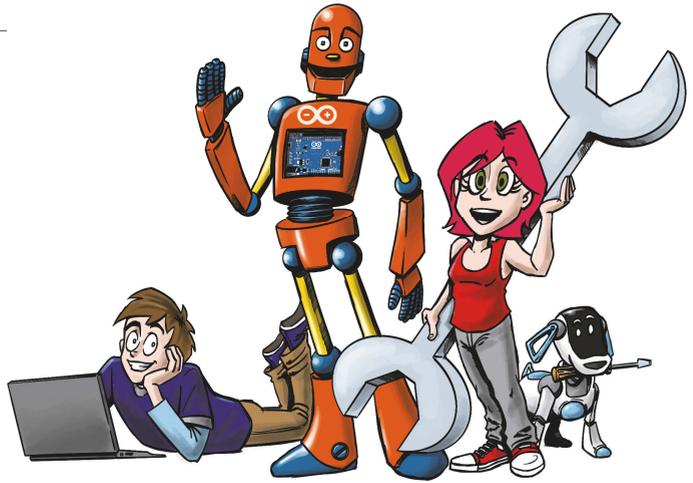
B



C



<b>Vergiss mein nicht – Nutzung des EEPROM</b> .....	<b>139</b>
Allgemeines zum EEPROM .....	139
Was könnte man mit dem EEPROM programmieren? .....	141
Projekt: Blackbox .....	141
Zusammenfassung .....	150
Ein paar Fragen ... .....	150
... und ein paar Aufgaben .....	150
<b>Das Installieren der IDE</b> .....	<b>151</b>
Installation .....	151
<b>Lösungen</b> .....	<b>153</b>
Kapitel 1 .....	153
Kapitel 2 .....	154
Kapitel 3 .....	154
Kapitel 4 .....	154
Kapitel 5 .....	155
Kapitel 6 .....	155
Kapitel 7 .....	156
Kapitel 8 .....	156
Kapitel 9 .....	157
Kapitel 10 .....	158
Kapitel 11 .....	158
<b>Die Materialliste</b> .....	<b>159</b>
Materialien .....	159
<b>Stichwortverzeichnis</b> .....	<b>161</b>



# Einleitung

Du wolltest schon immer programmieren oder mit den Bestandteilen eines Computers arbeiten? Du wirst zwar nach diesem Buch keinen Computer bauen können, aber die Einleitung erklärt einiges, was du später machen kannst:

Dies erfährst du in der Einleitung:

- ⊙ Was ist ein Mikrochip, ein Mikrocontroller und was ist ein Arduino?
- ⊙ Wie programmiert man?
- ⊙ Was brauchst du an Materialien für dieses Buch?

Du siehst, du wirst einige Grundlagen über die Elektronik kennenlernen!

## Was ist ein Mikrochip?

Sicherlich hast du schon eine Vorstellung, wie ein Mikrochip aussieht. Wie eine kleine schwarze Fläche, die auf einer Platine, zum Beispiel dem Mainboard deines Computers, angebracht ist. Der Mikrochip, den du programmieren wirst, sieht etwas anders aus als ein Chip, den du kennen könntest. Er ist rechteckig, nicht quadratisch und besitzt wesentlich weniger Pins (die Metallbeine, die an der Seite herausragen). Außerdem sind diese Pins wesentlich größer, als du sie von »normalen« Mikrochips kennen könntest.

## Und was ist jetzt ein Mikrocontroller?

Ein Mikrocontroller ist ein Mikrochip, der alle benötigten Komponenten (sozusagen das Zubehör) bereits in sich trägt. Er sieht genauso aus wie ein Mikrochip. Du kannst es erneut mit dem Mainboard eines Computer vergleichen! Dann wäre das Mainboard der Mikrocontroller, und der Arbeitsspeicher wäre dann ein Teil des Zubehörs. Der Arbeitsspeicher ist ebenfalls in dem Mikrocontroller integriert.



Merke dir: Wenn in diesem Buch in Zukunft über einen Mikrochip geschrieben wird, ist eigentlich der Mikrocontroller gemeint! Und zur Rechtschreibung: Man kann sowohl Microcontroller mit c als auch mit k (Mikrocontroller) schreiben.

## Und was ist dann ein Arduino?

Damit du den Mikrocontroller am Anfang leichter programmieren kannst, gibt es das sogenannte Arduino-Projekt. Beim Arduino-Projekt gibt es fertige Platinen mit dem Mikrocontroller und eigener Software zum Programmieren. Die Platine wird dabei einfach als Arduino bezeichnet. In diesem Buch geht es, wie der Titel schon verrät, um die Programmierung eines Mikrochips auf der Arduino-Platine.

Das Besondere ist dabei, dass das Arduino-Projekt eine fertige Platine mit der passenden Entwicklungsumgebung für einen PC bereitstellt. Frühere Entwickler mussten sich teilweise sogar die Boards selber basteln, damit sie überhaupt lernen konnten, wie man programmiert.

## Wie programmiert man?

Programmieren kann man leider nicht durch Reden, sondern nur über Texte am Computer. Diese Texte werden zudem nicht in Deutsch verfasst, sondern die wenigen Wörter sind auch auf Englisch. Das sollte dich aber nicht abhalten, programmieren zu lernen! Du lernst programmieren mit der Programmiersprache C++, einer sehr systemnahen und systematischen, aber auch komplexen Programmiersprache. Um es am Anfang nicht viel zu schwer zu machen, haben die Arduino-Macher einen einfacheren Dialekt (= eine Variation der Programmiersprache) entwickelt. C++ basiert auf wenigen Wörtern, dafür auf vielen teilweise sehr kryptisch wirkenden Zeichen. In den Klammern steht jeweils, wie man dieses Zeichen »aussprechen« würde: zum Beispiel ++ (Inkrement), % (Modulo Division).

Wenn du erst einmal weißt, was die einzelnen Symbole bedeuten, wirst du sie fehlerfrei anwenden können. Mit folgendem Quelltext zeige ich dir, was mit C++ möglich ist. Diesen Quellcode habe ich für einen selbst gebauten Terrariencomputer entwickelt, der leider noch im Anfangsstadium ist:

```
#include "dimmen.h"
#include "kern_temperatur.h"
#include "terra_temperatur_class.h"
#include "class_cool.h"
void setup() {
  pinMode(13, OUTPUT); pinMode(12,OUTPUT);
  Serial.begin(9600); ADMUX = 0xC8; delay(10);
}

void loop() {
  Cooler cooler(13); delay(100); bool hot = kern_temp(17);
  if (hot) cooler.start();
  else{
    cooler.stop();
  }
  delay(500); }
```

Zum Programmieren brauchst du allerdings noch Software für den PC, die du unter *arduino.cc* im Internet findest. Zur genauen Installation schau bitte in den Anhang oder in das nächste Kapitel.

Nun sollst du aber etwas über das Gerät lernen, das wir für das Buch benötigen, den Arduino. Der Arduino, den du vermutlich vor dir hast, ist ein Arduino Uno. Du solltest den Arduino erst einmal über USB mit dem Computer verbinden. Du kannst ihn zwar noch nicht programmieren, aber du kannst ihn schon als Stromquelle für die nächsten Schaltungen benutzen.

## Materialien

Um jetzt in der Einführung weiterzumachen, brauchst du bereits einige Materialien. Dazu gehören eine LED (eine Leuchtdiode, also ein Bauteil, das Licht abstrahlt), ein schwarzes und ein graues Kabel (oft werden von Arduino-Nutzern schwarz-rote Kabel als Farbkombination benutzt, im Buch benutzen wir wegen der schwarz-weißen Schaltpläne schwarz-grau zur besseren Erkennbarkeit), der Arduino und ein sogenanntes Breadboard.

Sämtliche benötigten Materialien sind übrigens im Anhang C am Ende des Buches aufgelistet.



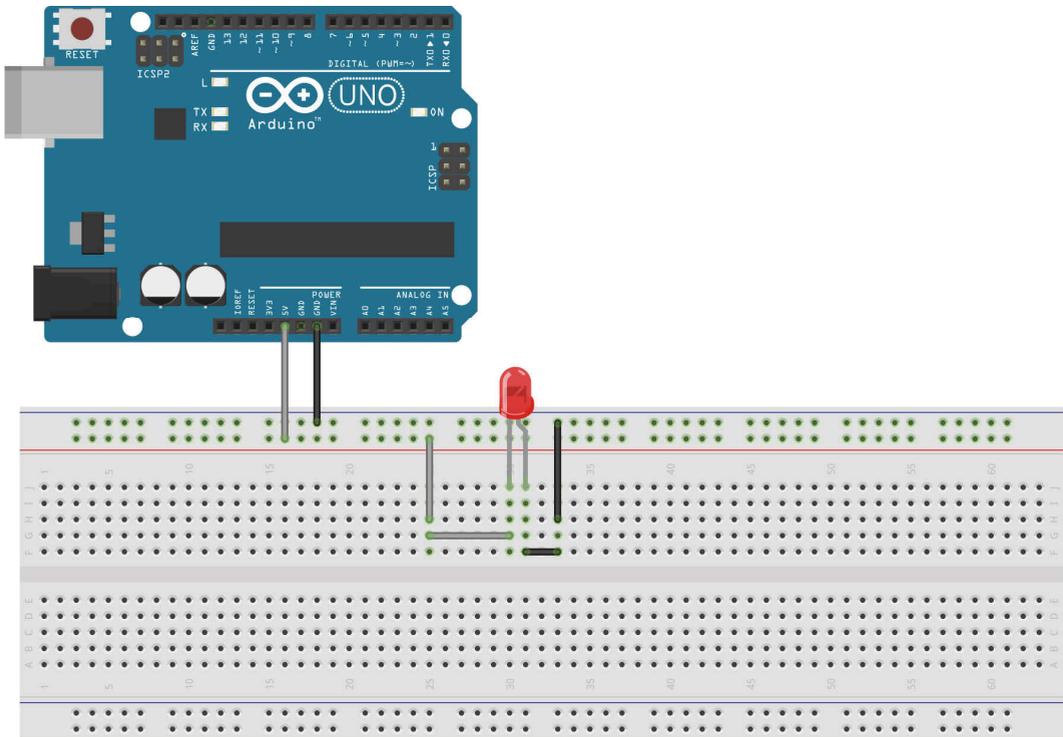
# Der erste Anfang

Nun wollen wir aber zum ersten Mal irgendetwas mit dem Arduino machen. Ich schlage vor, zuerst lassen wir einfach eine LED leuchten. Dies können wir auf zwei Arten machen: Entweder wir benutzen den Arduino als Stromquelle oder eine Batterie. Damit du dich gleich mit dem Arduino vertraut machst, baue einmal die Schaltung auf dem ersten Bild in diesem Buch nach. Dabei musst du Folgendes beachten: Das rote Kabel kommt an den Pin des Arduino mit der Beschriftung VCC, das schwarze an das Kabel mit GND, also an Plus und Minus. Und zuletzt: Das rote Kabel (der Pluspol) wird an die lange Seite der LED angeschlossen, das kurze Ende an das schwarze Kabel (Minuspol).

Verwende in diesem Beispiel nur (!) die rote LED, für andere LEDs bräuchtest du definitiv einen Widerstand (siehe weiter unten).



Alle Schaltpläne kannst du dir unter [www.mitp.de/580](http://www.mitp.de/580) herunterladen. Dort sind die Kabel auch farbig – im Gegensatz zu den Schaltplänen im Buch.



Hiermit hast du gleich vier wichtige Bestandteile für eine Schaltung, die immer wieder im Buch vorkommen. Drei davon werden sogar in jeder Schaltung verwendet: der Arduino, das Breadboard, die LED und zuletzt das Kabel.

Auch wenn das letzte Bauteil eigentlich das einfachste ist, wirst du ohne dieses kaum eine Schaltung so wie im Buch beschrieben nachbauen können. Lediglich beim Lötten kann man auf Kabel verzichten, aber auch hier gibt es Ausnahmen, bei denen man ein Kabel braucht.



## Widerstände

Leider geht das obige Beispiel nur mit roten LEDs. Andere würden wegen der Stromstärke durchbrennen. Deswegen braucht man ein weiteres Bauteil, den Widerstand.

Ein *Widerstand* ist ein kleines Bauteil, das dazu dient, den Strom zu reduzieren, damit empfindlichere Bauteile nicht beschädigt werden. Hierbei ist es egal, wie herum ein Widerstand angeschlossen ist. Er sieht im Prinzip wie eine recht kleine Röhre aus, auf der mehrere farbige Streifen zu finden sind.



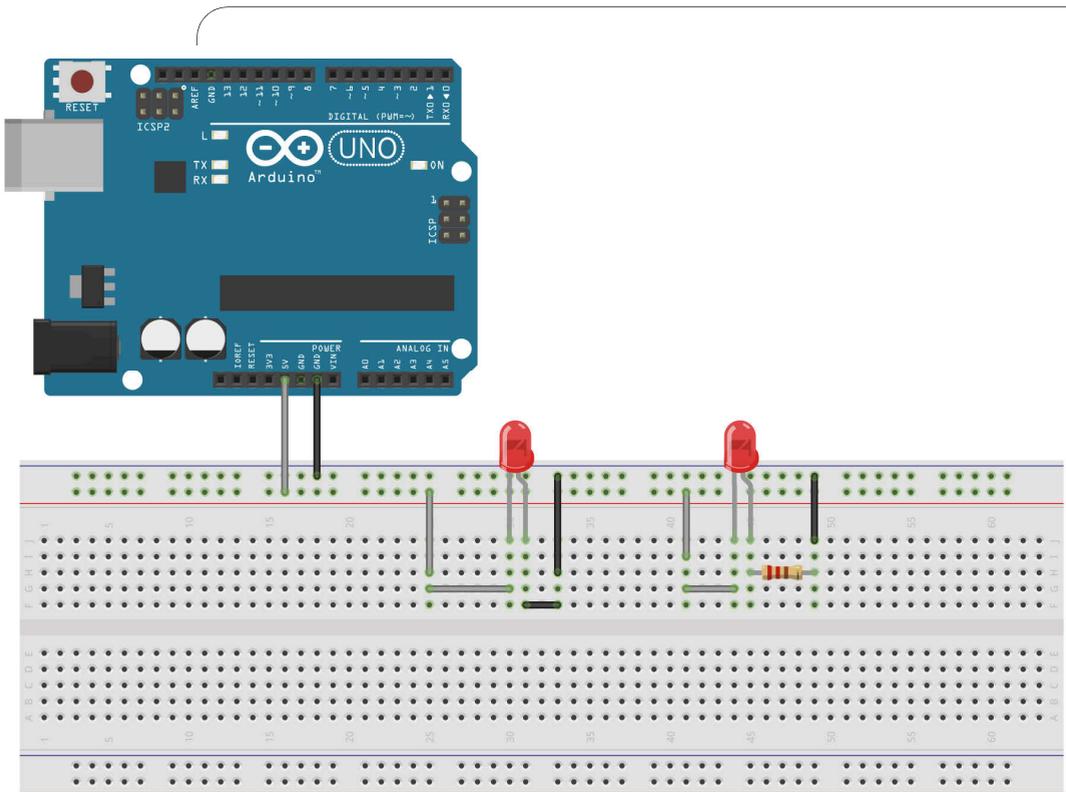
Widerstände sind relativ günstig (ungefähr 1 Cent das Stück), sodass du 100 für einen Euro bekommen kannst.

Wenn du jetzt die nächste Schaltung nachbaust, wirst du sehen, dass die LED mit Widerstand schwächer leuchtet, eben weil der Strom reduziert/geschwächt wird.

Ein wichtiger Tipp: Du musst im Buch immer lesen, welchen Widerstand man einfügen soll, denn die Abbildungen enthalten nur einen allgemeinen Platzhalter für den Widerstand.



Im unteren Beispiel wird ein 130-Ohm-Widerstand gefordert.



fritzing

In diesem Beispiel ist es so, dass die linke LED heller leuchtet als die rechte. Es ist übrigens egal, auf welcher Seite man den Widerstand anschließt. Jetzt aber die entscheidende Frage: Wie bestimmt man den Wert eines Widerstandes, denn es gibt ja verschiedene Stärken?

Wenn du genau hinsiehst, hat jeder Widerstand einen Farbcode, also mehrere verschiedenfarbige Ringe. Je nach Farbe hat das Bauteil eine andere Eigenschaft. Die Stärke des Widerstands wird übrigens in der Einheit Ohm gemessen. Mit der folgenden Tabelle kannst du den Wert eines Widerstands bestimmen.

Ein Tipp: Der goldene oder silberne Ring ist immer rechts, wenn du den Widerstand richtig hältst.



Farbe	1. Ring	2. Ring	Multiplikator	Toleranz
Silber			0.01	10%
Gold			0.1	5%
Schwarz	-	0	1	
Braun	1	1	10	
Rot	2	2	100	
Orange	3	3	1K = 1000	
Gelb	4	4	10K	
Grün	5	5	100K	
Blau	6	6	1M	
Violett	7	7	10M	
Grau	8	8	100M	
Weiß	9	9	1G	

Mit dieser Tabelle kannst du Widerstandswerte nachschlagen. Überall, wo ein »K« steht, musst du dir drei Nullen denken. Denn K steht für Kilo, also Tausend (1K entspricht 1.000, 10K entsprechen 10.000). M(ega) und G(iga) stehen für 1.000.000 bzw. 1.000.000.000.

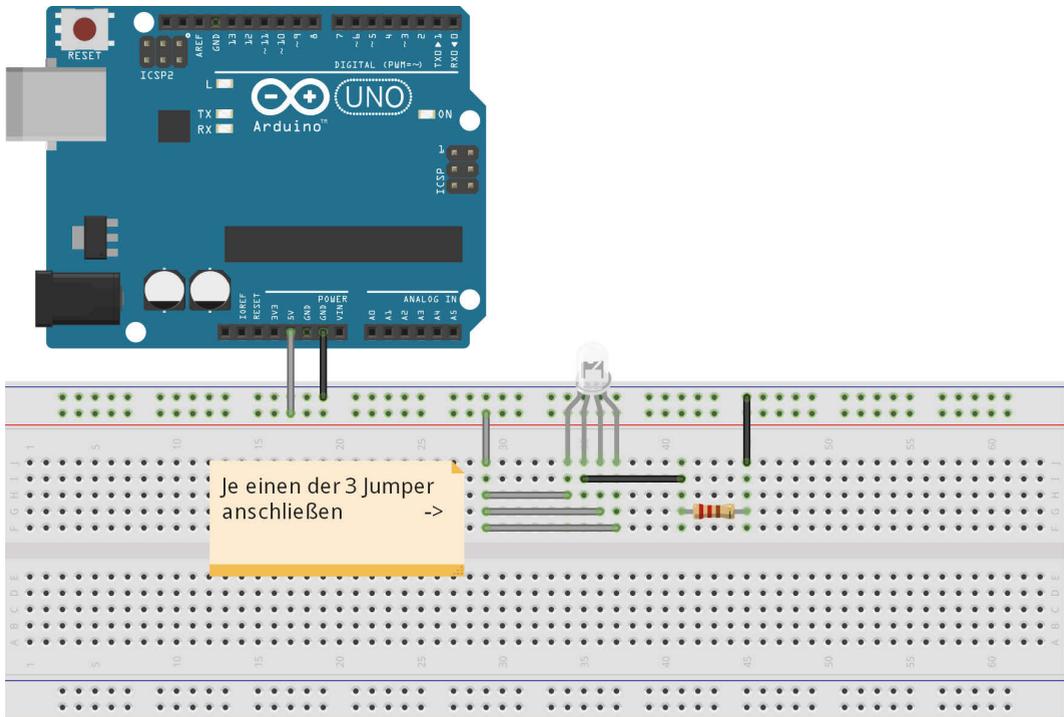
Die Toleranz steht dafür, wie sehr der Wert abweichen darf. Dieser Ring ist meistens goldenfarbig.

Beispielrechnung:

Wenn wir nun den Widerstand 130 Ohm brauchen, hat er folgende Farben: (1. Ring = Braun, 2. Ring = Orange und 3. Ring = Braun), denn 13 (1. und 2. Ring) + 10 (Multiplikator) gleich (=) 130 (Ohm).

## Eine Schaltung mit einer mehrfarbigen LED

Baue einmal die nächste Schaltung nach.



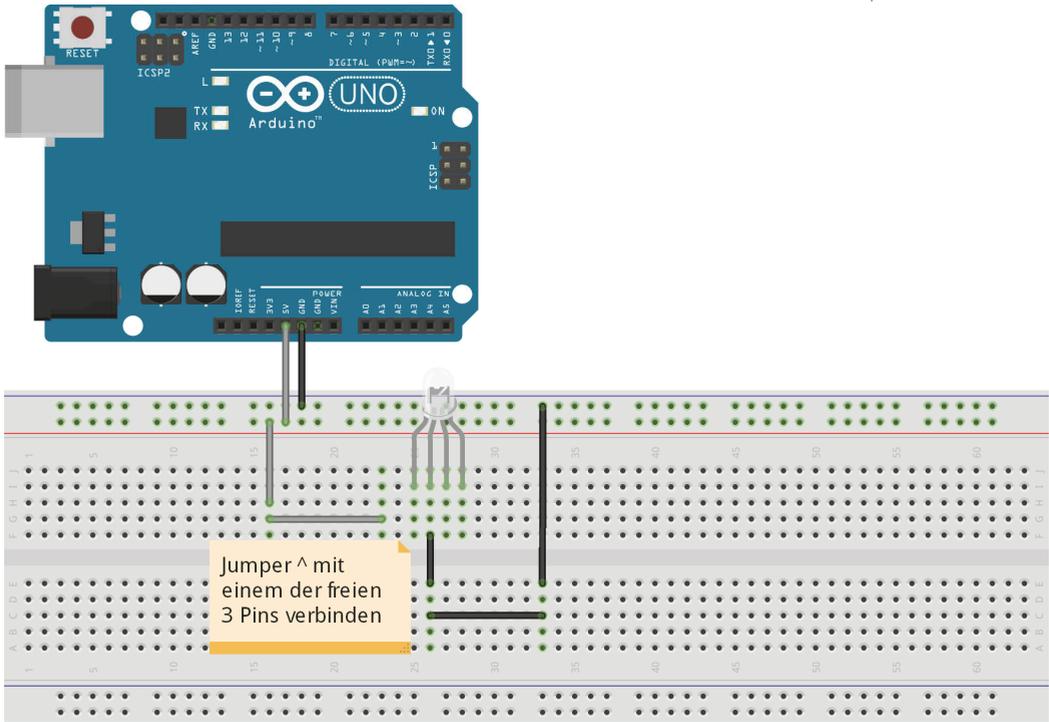
fritzing

Dies ist eine Schaltung mit einer mehrfarbigen LED. Das heißt, dass eine LED in drei Farben leuchten kann (in der Regel Rot, Grün und Blau). In der obigen Schaltung brauchst du wieder einen 130-Ohm-Widerstand. Wenn du nun das orangefarbene Kabel nimmst und an einen der freien (LED-)Pins hältst, leuchtet die spezielle LED in einer der drei Farben.

Dabei gibt es einen gemeinsamen Pluspol und für jede Farbe der LED einen Minuspol. Um dieses Kapitel gut abzuschließen, baue einfach die nächste Schaltung nach und sieh, was passiert. Das neue Bauteil ist ein Fotowiderstand, den ich aber erst in einem späteren Kapitel beschreibe.

Statt einer Batterie kannst du auch den Arduino benutzen (VCC->Rot, GND->Schwarz), du musst dann aber eventuell den Widerstand einbauen. Bei 3 V kann man ihn auch teilweise auslassen.





fritzing

## Zusammenfassung

Du weißt nun, ...

- ◇ wie man LEDs anschließt
- ◇ was ein Widerstand ist
- ◇ wie man mehrfarbige LEDs anschließt

Und du weißt ...

- ◇ Wie man eine LED zerstören könnte

## Ein paar Aufgaben

1. Schalte drei LEDs in eine Reihe (+ 3 \* 330-Ohm-Widerstände).
2. Ersetze den Arduino als Stromquelle durch eine Zungenbatterie und beobachte, wie sich die Lichtstärke der LEDs verändert.