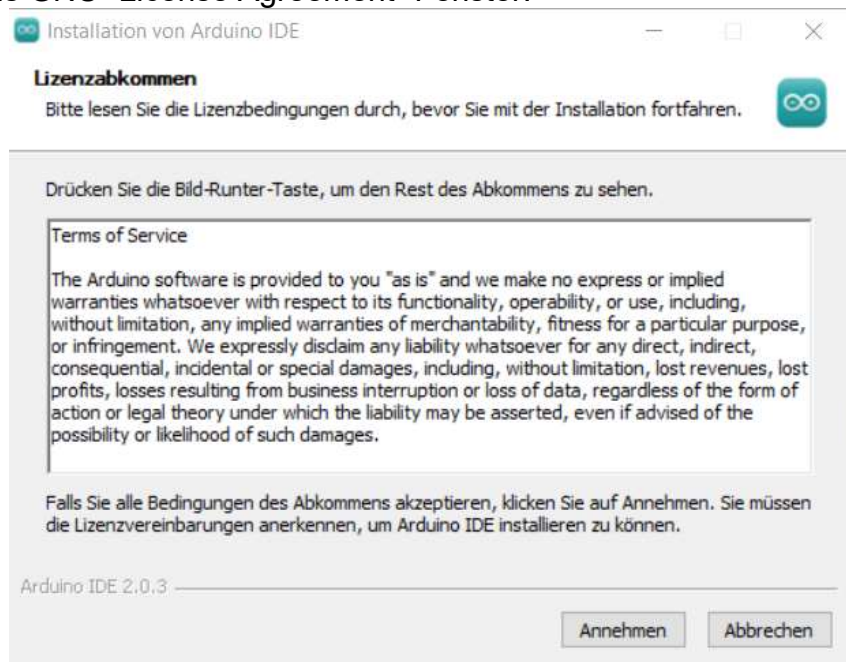


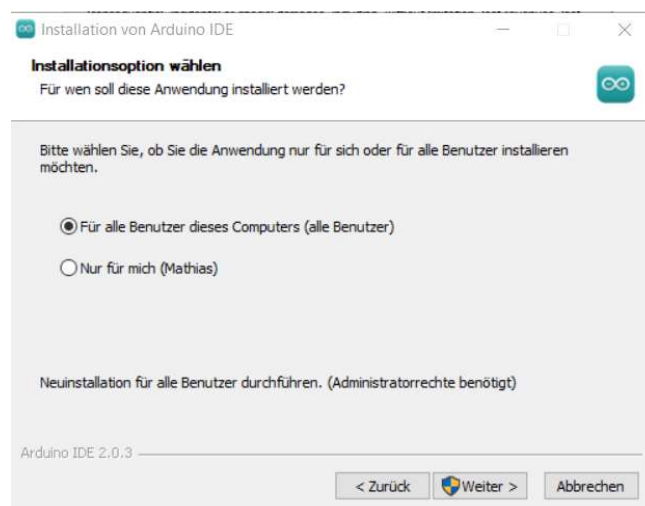
Installation der ARDUINO Entwicklungsumgebung

Im Folgenden wird die Installation der Entwicklungsumgebung und die Einstellungen, die wir für diesen Workshop verwenden wollen, gezeigt. Die Screenshots sind bei der Installation für Windows[®] aufgenommen worden. Für die anderen Betriebssysteme (IOS[®], Linux[®]) gelten sie entsprechend. Es wird hier als Beispiel die ARDUINO DIE Version 2.03 gezeigt. (Stand Januar 2023). Diese Version kann parallel zu den Versionen 1.x installiert werden, so dass zwei funktionierende Entwicklungsumgebungen auf den Rechner installiert sein können.

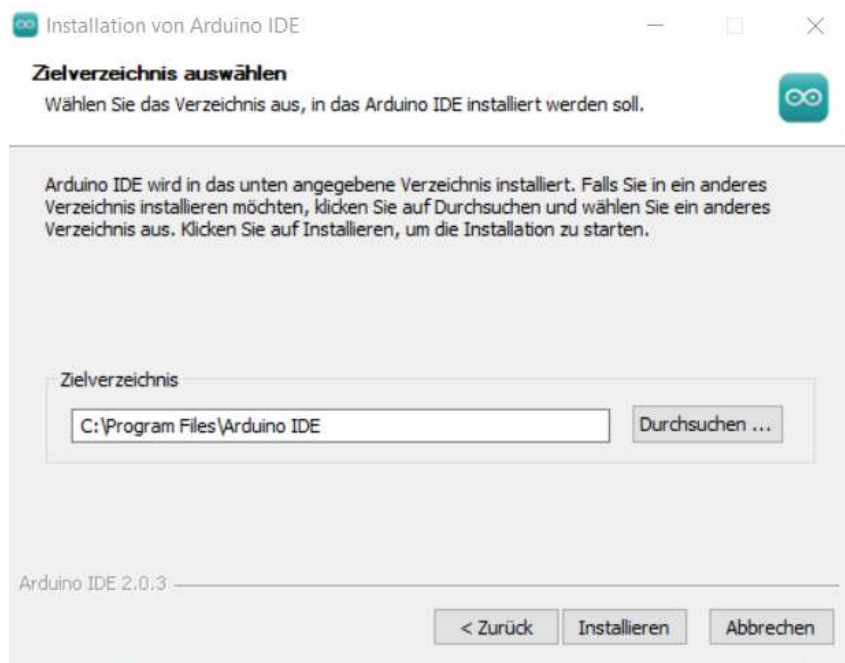
Nach dem Starten der Installationsdatei („arduino-ide_2.0.3_Windows_64bit.exe“) erscheint das GNU- License Agreement- Fenster.



Wenn das Programm genutzt werden soll, sind die GNU- Lizenzbedingungen zu akzeptieren („I Agree“ oder wie hier in der deutschen Version „Annehmen“). Es erscheint die Abfrage für welche Benutzergruppe die Software installiert werden soll. Hier sollte die Option „alle Benutzer“ gewählt werden, wenn nicht zwingende Rechte-Gründe dagegensprechen.



Nachdem Neustart des Installationsprozesses und erneutem Bestätigen der „GNU“-Lizenz erscheint die Auswahl des Installationsverzeichnisses.



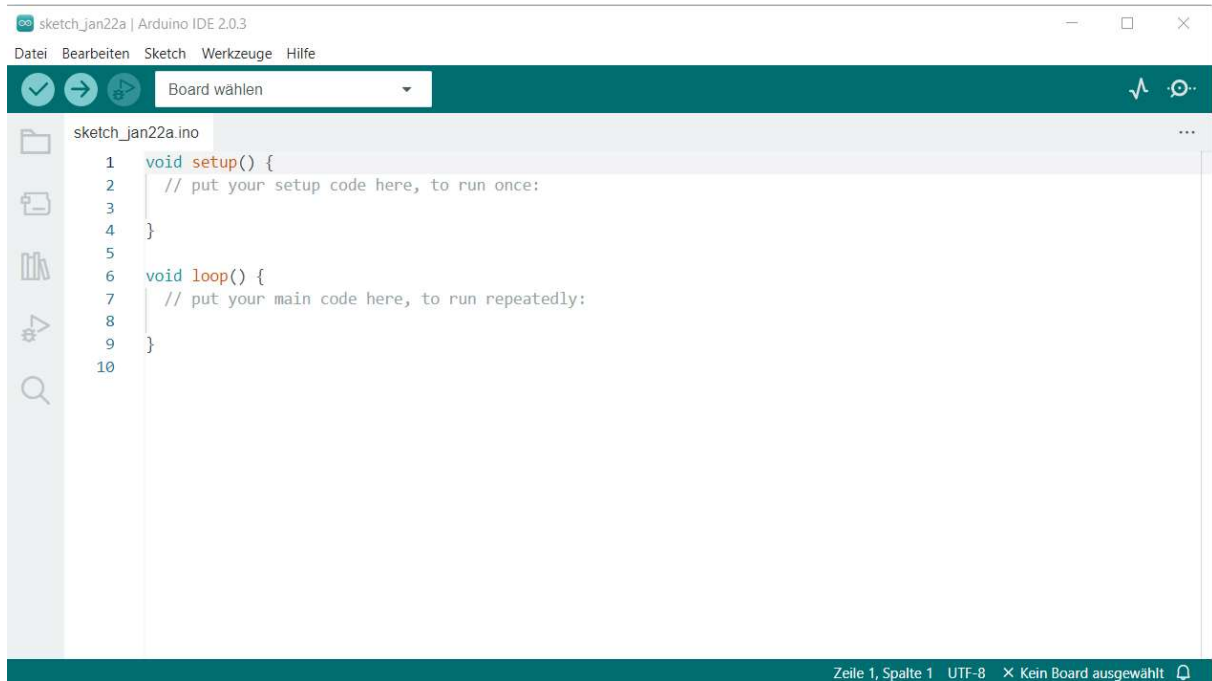
Es kann das „Default-Verzeichnis“ gewählt werden oder ein anderes Verzeichnis eurer Wahl.

Bei Windows 10[®] kann u. U. eine Sicherheitsabfrage auftauchen, die wir mit „Installieren“ und „Software von ‚Arduino srl‘ immer vertrauen bestätigen. Danach können wir mit „Close“ die Installation abschließen.

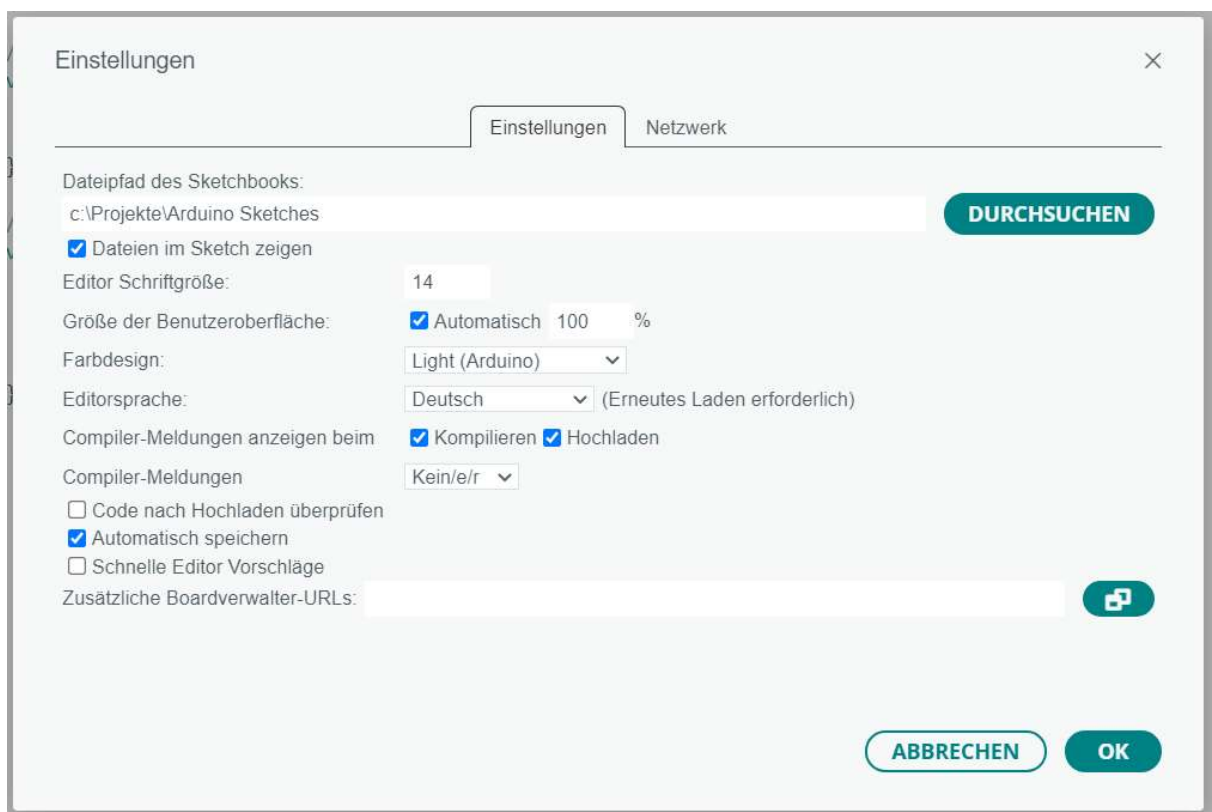


Nun ist die Installation abgeschlossen. Beim ersten Starten der Entwicklungsumgebung kann es unter Windows 10[®] u. U. zu einer „Firewall-Warnung“ kommen, denn die Entwicklungsumgebung benötigt Java.

Es startet dann die Entwicklungsumgebung.



Unter „Datei“-> „Voreinstellungen“ (oder durch Drücken der Tastenkombination <Strg>-<Komma>) wählen wir einige Voreinstellungen.



Der wichtigste Punkt in den Voreinstellungen ist der Pfad, der in „Dateipfad für Sketchbooks“ eingetragen ist. Als Voreinstellung ist hier ein Pfad in User-Bereich von Windows eingetragen. Da dieser schwer zu finden ist, sollte der geändert werden. Bibliotheken für z. B. Unterstützung besonderer Bauteile werden auch dort gespeichert.

Zu erwähnen gibt noch die „Symbolleiste“ am linken Rand. Hier kann man seine bisherigen Projekte (Sketchbooks) mit dem ARDUINO verwalten. Ebenso können hier neue Boards eingebunden (die bekanntesten sind schon per default eingebunden) werden, die Bibliotheken verwaltet werden, Einstellungen zum Debuggen des Codes vorgenommen werden und das geladene Sketchbook durchsucht werden.



Wie die Debugereinstellungen genutzt werden, sehen wir später noch.

Wie ist ein ARDUINO-Sketch aufgebaut

Ein ARDUINO-Programm besteht aus mindestens den folgenden vier Abschnitten.

```
1  /*
2   * Fade
3   *
4   * This example shows how to fade an LED on pin 9 using the analogWrite()
5   * function.
6   *
7   * The analogWrite() function uses PWM, so if you want to change the pin you're
8   * using, be sure to use another PWM capable pin. On most Arduino, the PWM pins
9   * are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.
10  *
11  * This example code is in the public domain.
12  *
13  * http://www.arduino.cc/en/Tutorial/Fade
14  */
15
16 int led = 9;           // the PWM pin the LED is attached to
17 int brightness = 0;   // how bright the LED is
18 int fadeAmount = 5;   // how many points to fade the LED by
19
20 // the setup routine runs once when you press reset:
21 void setup() {
22   // declare pin 9 to be an output:
23   pinMode(led, OUTPUT);
24 }
25
26 // the loop routine runs over and over again forever:
27 void loop() {
28   // set the brightness of pin 9:
29   analogWrite(led, brightness);
30
31   // change the brightness for next time through the loop:
32   brightness = brightness + fadeAmount;
33
34   // reverse the direction of the fading at the ends of the fade:
35   if (brightness <= 0 || brightness >= 255) {
36     fadeAmount = -fadeAmount;
37   }
38   // wait for 30 milliseconds to see the dimming effect
39   delay(30);
40 }
```

Abbildung 15: Der ARDUINO-Sketch

Der erste Abschnitt (rosa Rahmen) ist der Programmierklärungsbereich. Hier sollte ein Kommentar, der die Funktionsweise des Programms beschreibt, eingefügt werden. Bei „kleineren“ Projekten entfällt er meistens oder wird nur bei Veröffentlichung hinzugefügt.

Der zweite Abschnitt (roter Rahmen) beschreibt den Deklarationsteil. Innerhalb dieses Teils werden globale Einstellungen für das Programm vorgenommen. Z. B. Pin-Zuordnungen oder Programm-Konstanten oder aber auch die Anweisungen zum Einbinden von Bibliotheken.

Der dritte Abschnitt (grüner Rahmen) wird mit dem Schlüsselwort „`void Setup()`“ eingeleitet. Es ist ein Unterprogramm, das nur einmal im Programm, nämlich beim Start nach einem RESET des Programms, ausgeführt wird. Darin werden in der Regel Variablen-Initialisierungen und die Datenrichtungseinstellungen der Pins vorgenommen.

Der vierte Abschnitt (violetter Rahmen) ist das Hauptprogramm des Sketches. Er wird mit dem Schlüsselwort „`void Loop()`“ eingeleitet. Loop deshalb, weil ein ARDUINO-Programm niemals aufhört und daher am Ende der Anweisungen immer wieder neu (ohne das Setup-Programm zu durchlaufen) gestartet wird.

Einige in den Projekten benutzte Bauteile



Bild 16: Beispiele für Widerstände als Bauteil Quelle: www.clipartpanda.com

Der Widerstand ist das zentrale Bauteil in der Elektronik. Wie der Name sagt, setzt er der Spannung einen „Widerstand“ entgegen. Fließt ein Strom durch den Widerstand „fällt“ an ihm eine Spannung ab. Oder anders herum gesagt: Legt man an den Widerstand eine Spannung an, fließt durch den Widerstand ein Strom, der von dem Wert des Widerstandes abhängig ist. Nach dem sog. „ohm’schen Gesetz“ gilt:

$$U = R \times I$$

Hierbei ist „ U “ die Spannung, „ R “ der Widerstand und „ I “ der durch den Widerstand fließende Strom.

Das Schaltzeichen für einen Widerstand ist folgendes. Der Wert des Widerstandes steht unmittelbar über dem Schaltzeichen, darunter oder auch direkt im Schaltsymbol, je nach dem wie es der Platz im Schaltplan zulässt.



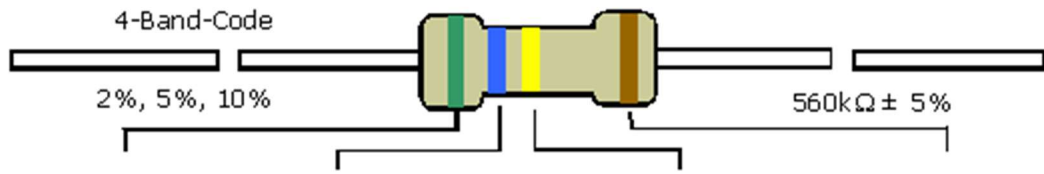
Bild 17: Schaltsymbol Widerstand Quelle: DIN

In US-amerikanischen Schaltbildern ist auch folgendes Symbol üblich:



Bild 18: US- Schaltsymbol Widerstand Quelle: FCC

Da meist der Platz auf einem Widerstand nicht ausreicht, um den Wert aufzudrucken, wird der Wert durch Farbringe kodiert.



COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	±0.5% (D)
Blue	6	6	6	1MΩ	±0.25% (C)
Violet	7	7	7	10MΩ	±0.10% (B)
Grey	8	8	8		±0.05%
White	9	9	9		
Gold				0.1	± 5% (J)
Silver				0.01	± 10% (K)



Quelle: <http://www.sengpielaudio.com/Farbcodewiderstaende13.htm>



Bild 20: LEDs Quelle: www.hobby-bastelecke.de

Eine Leuchtdiode (kurz LED genannt) ist eine Diode, die aufleuchtet, wenn sie in Durchlassrichtung betrieben wird. Früher in den 80er Jahren gab es nur rote, gelbe und grüne Leuchtdioden. Die blauen waren damals sehr teuer (über 10 DM für eine Diode!). Heute sind dagegen auch die blauen Leuchtdioden für wenig Geld zu haben und es gibt daneben auch weiße sowie superhelle Leuchtdioden.

Genau wie normale Dioden haben Leuchtdioden zwei Anschlüsse: Die **Anode** und die **Kathode**. Sie leuchten nur, wenn der Strom von der Anode zur Kathode fließt. Wird die Diode in Sperrrichtung betrieben, leuchtet sie nicht da kein Strom fließen kann. Genau wie normale Dioden haben Leuchtdioden zwei Anschlüsse: Die **Anode** und die **Kathode**. Sie leuchten nur, wenn der Strom von der Anode zur Kathode fließt. Wird die Diode in Sperrrichtung betrieben, leuchtet sie nicht da kein Strom fließen kann. Allerdings haben die meisten LEDs nur eine **geringe maximale Sperrspannung**. Daher sollte man eine Verpolung vermeiden, da die LED so zerstört werden kann. Aus dem gleichem Grund sind LEDs auch gegen elektrostatische Aufladungen empfindlich!

Leider sind die Anschlussbezeichnungen nicht einheitlich. Oft ist das Anschlussbein der Kathode etwas kürzer oder das Gehäuse auf Seite der Kathode etwas abgeflacht. LEDs haben eine höhere Durchlassspannung als normale Dioden. Sie liegt je nach Typ der Leuchtdiode zwischen 1,5 bis 4,0 Volt.

Rote LEDs: 1,6 Volt
Gelbe LEDs: 2,0 Volt
Grüne LEDs: 2,2 Volt
Blaue LEDs: 3,0 bis 4,0 Volt
Weiße LEDs: 3,0 bis 4,0 Volt

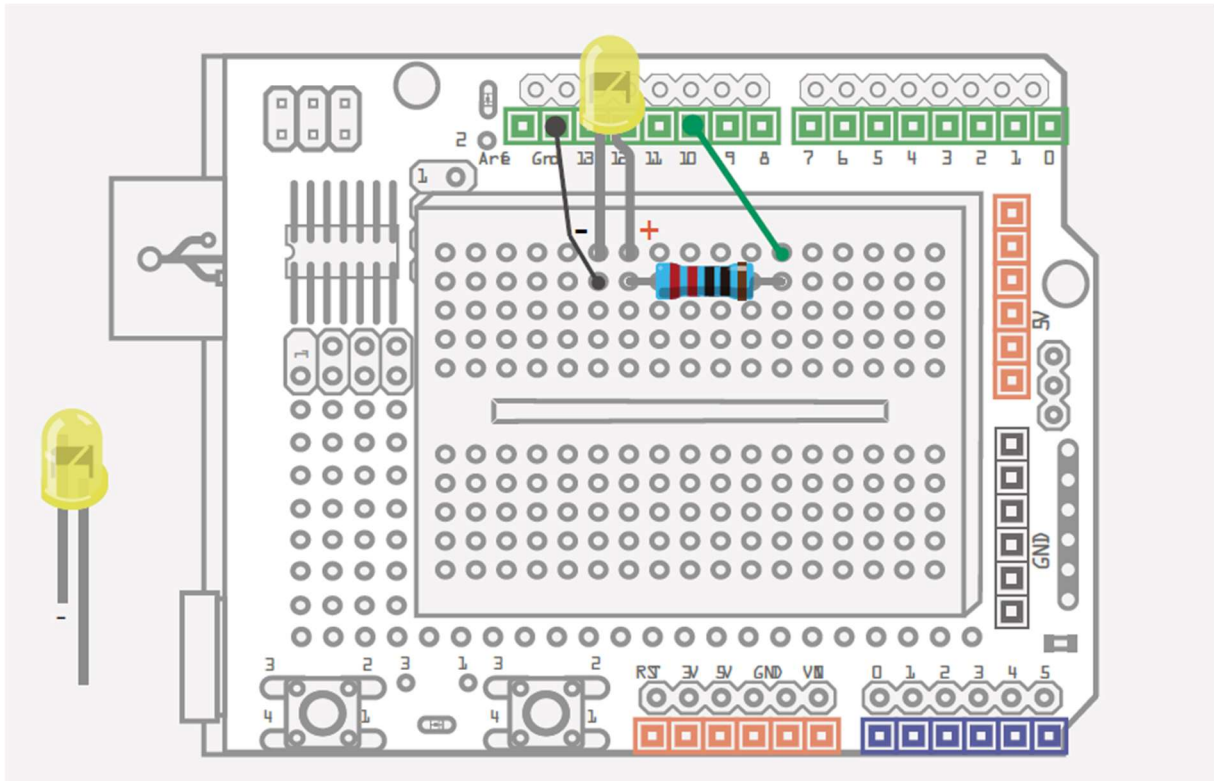
Der höchstzulässige Strom durch eine LED, der **Durchlassstrom**, liegt bei älteren LEDs bei etwa 50 mA. Wenn man eine LED aber an dieser Belastungsgrenze betreibt, ist ihre Lebensdauer nicht sonderlich hoch! Die meisten LEDs erreichen bei

einem Durchlassstrom von etwa 15 bis 20 mA bereits eine hohe Leuchtkraft, daher haben sich diese Werte als Standardwerte eingebürgert. Die Begrenzung des Stromes erfolgt durch einen Vorwiderstand. Daher zuerst ein wichtiger Merksatz:

LEDs immer mit einem Vorwiderstand anschließen da die LED ansonsten sofort zerstört wird!

Heute gibt es auch **superhelle LEDs**, die eine besonders hohe Leuchtkraft haben und immer mehr Einzug halten. Sie werden z. B. in kleinen Taschenlampen verwendet. Auch für sie gilt der erwähnte Durchlassstrom von 15 bis 20 mA. Ihre Vorteile gegenüber normaler Glühlampen sind klar: Einen deutlich geringeren Stromverbrauch und eine erheblich höhere Lebensdauer! Diese superhellen LEDs haben aber meist eine höhere Durchlassspannung als die normalen Typen.

Projekt 1: Blinkende LED



Die an PIN 10 angeschlossene LED soll im halbsekündlichen Abstand leuchten und wieder ausgehen.

Erweiterung: Die LED soll nicht rhythmisch blinken, sondern zweimal kurz blitzen und dann eine Sekunde nicht leuchten.

Fragen: Was ist der Nachteil der Funktion „delay()“?
Was könnte gegen Nachteil unternommen werden?