

Workshop

Arduino-Programmierung #1

Microcontroller, Arduino IDE, "Blink"-Sketch, LEDs

Joachim Baur

E-Mail: post@joachimbaur.de

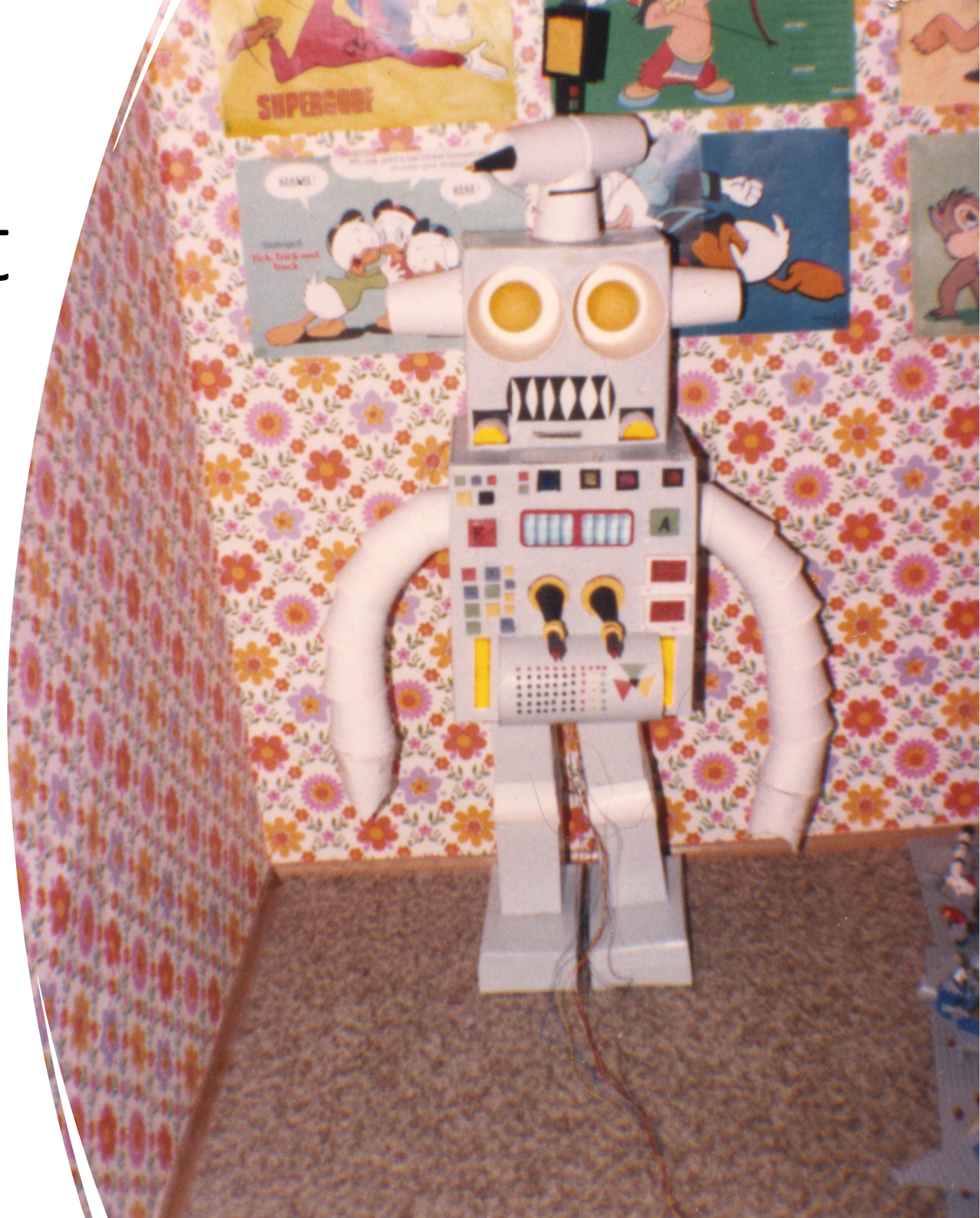
ZTL-Alias: [@joachimbaur](https://www.ztl.de/@joachimbaur)

Download für diesen Workshop: www.joachimbaur.de/WS1.zip

Plan für die Weltherrschaft

- Geboren 1969
- 90er Studium Grafik-Design
- Seither selbstständig mit Programmierung (erst CD-ROM, dann WWW)
- Arduino-Projekte seit 2015 (Löten + Code)

baur 3.0
Service GmbH
EDV • IT • BEAUTY
Talstr. 4 • 76764 Rheinzabern



Prototypen für Messemodelle

FORVIA
faurecia

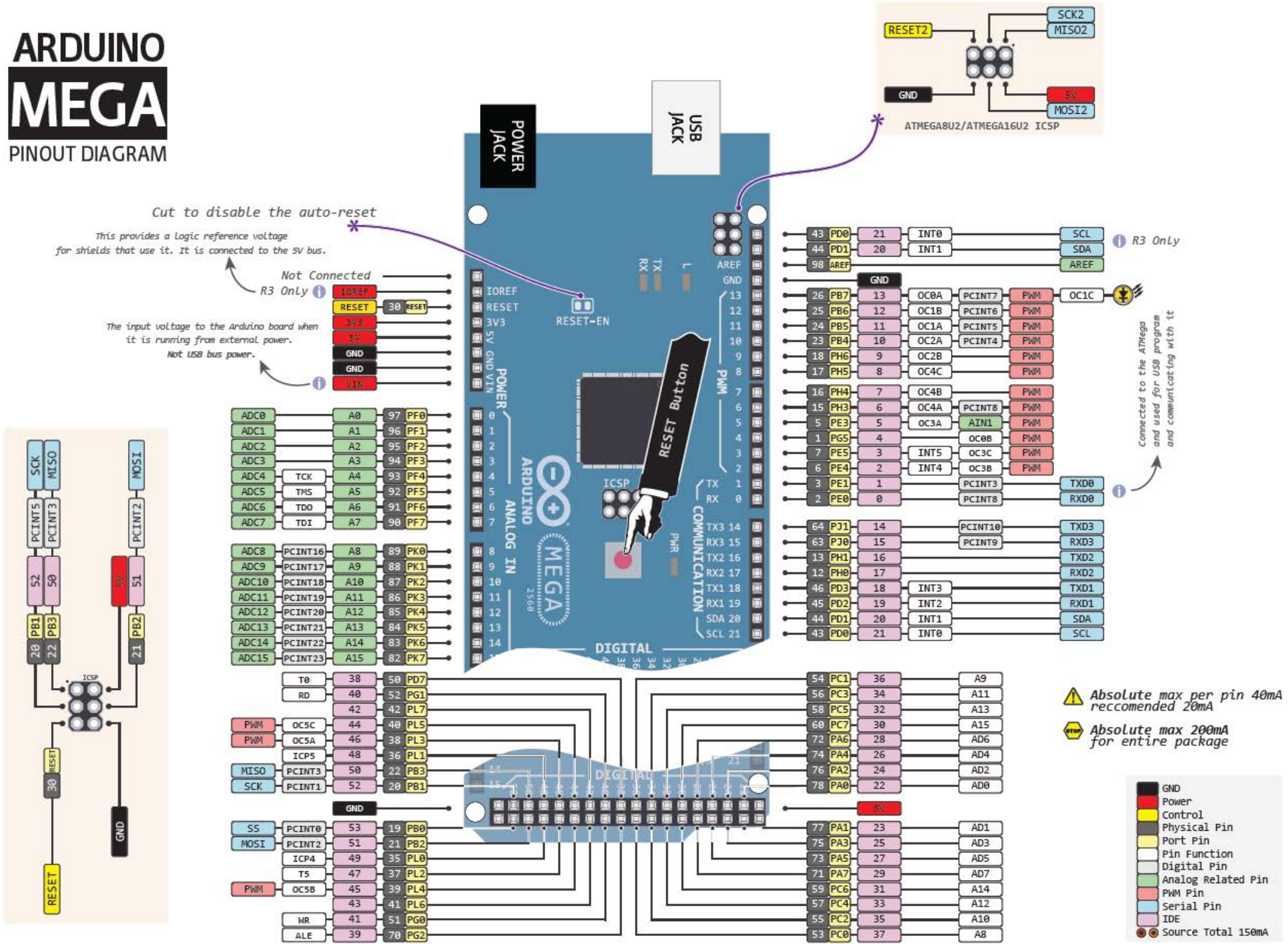


Weitere Infos: <https://baur-service-gmbh.de/prototypen.html>

[illegible]

- Klein und universell einsetzbar (GPIOs)
- Viele Größen und Varianten
- Aber: kein grafisches Betriebssystem/
Monitor/Tastatur

ARDUINO MEGA PINOUT DIAGRAM



Beispiele für Varianten

Board	CPU	Speed	Logic Level	Flash	GPIO Pins	Info im Web
Arduino Nano (2008)	Atmel (Microchip) ATmega328	16 MHz	5V	32 K	22	Arduino.cc
Arduino Mega 2560 (2010)	Atmel (Microchip) ATmega2560	16 MHz	5V	256 K	54	Arduino.cc
Adafruit Feather RP2040 (2022)	RP2040 32-bit Cortex M0+ dual core	125 MHz	3.3V	264 K	21	Adafruit.com
ESP32 (2016)	Xtensa dual-core 32-bit LX6	240 MHz	3.3V	2048 K	32	Espressif.com
Teensy 4.1 (2020)	ARM Cortex-M7	600 MHz	3.3V	7936 K	55	PJRC.com

Weitere Infos: https://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems

Beispiele für Bezugsquellen

Eckstein <https://eckstein-shop.de>
Berrybase <https://www.berrybase.de>
EXP-Tech <https://www.exp-tech.de>
Distrelec <https://www.distrelec.de>
AZ-Delivery <https://www.az-delivery.de/>

Farnell <https://de.farnell.com>
Aliexpress <https://de.aliexpress.com>
DigiKey <https://www.digikey.de/>
Mouser <https://www.mouser.de>



Arduino IDE Download & Installation

- Für Installation und Programmstart leider unbedingt Online-Verbindung nötig!
- <https://www.arduino.cc/> , Button „Software“
- Arduino IDE 2.x
- Richtige Variante (Windows/macOS) rechts anklicken
- „Just Download“
- Unter Windows werden die **Original Arduino-Treiber** mitinstalliert
- Beim Einsatz von Klonen mit anderen USB-Chips müssen ggf. **FTDI-Treiber** oder **CH340-Treiber** separat installiert werden
<https://ftdichip.com/drivers/vcp-drivers/>
<https://www.arduined.eu/ch340-windows-10-driver-download/>

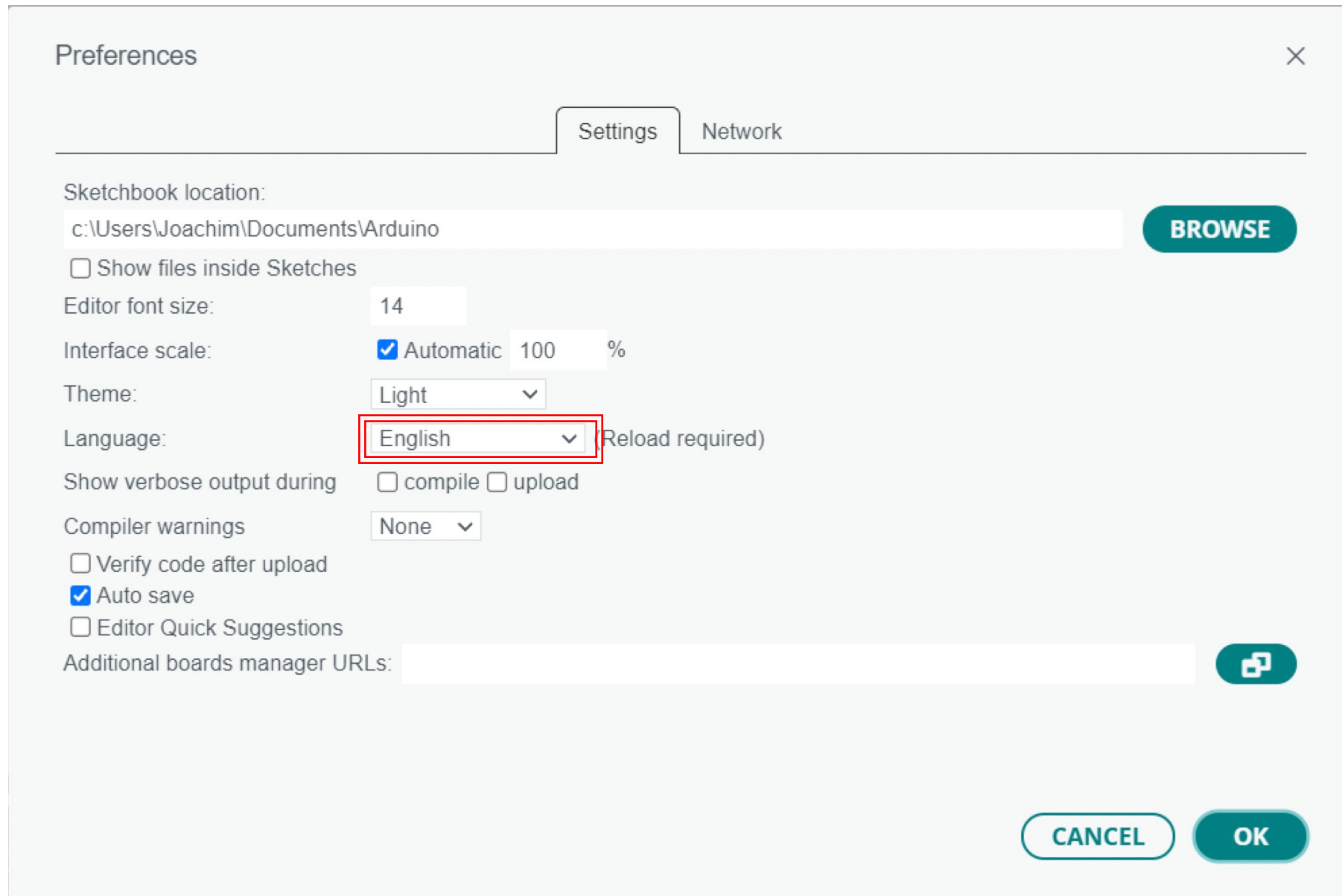
Default-Verzeichnisse der IDE

- Sketchbook-Location für Projekte und Libraries
- **Windows:** C:\Users\{username}\Documents\Arduino
- **Mac:** /Users/{username}/Documents/Arduino
- **Linux:** /home/{username}/Arduino
- Unterordner „libraries“ für Bibliotheken
- Dateiendung .ino für IDE-Codeteilen (C/C++)
- .ino-Dateien müssen in Ordnern gleichen Namens liegen ("Testprojekt.ino" also in Ordner "Testprojekt")

<https://support.arduino.cc/hc/en-us/articles/4415103213714-Find-sketches-libraries-board-cores-and-other-files-on-your-computer>

Voreinstellungen der IDE nach Installation

Aufruf über Menü (Arduino/File > Preferences) oder STRG + Komma



The screenshot shows the 'Preferences' dialog box in the Arduino IDE, with the 'Settings' tab selected. The 'Language' dropdown menu is highlighted with a red rectangle and shows 'English' selected, with a '(Reload required)' note next to it. Other settings include the sketchbook location, editor font size, interface scale, theme, and various checkboxes for output, warnings, and auto-save.

Preferences ×

Settings Network

Sketchbook location:
c:\Users\Joachim\Documents\Arduino BROWSE

☐ Show files inside Sketches

Editor font size: 14

Interface scale: ☒ Automatic 100 %


Theme: Light ▾

Language: English ▾ (Reload required)

Show verbose output during ☐ compile ☐ upload

Compiler warnings: None ▾

☐ Verify code after upload
☒ Auto save
☐ Editor Quick Suggestions

Additional boards manager URLs: 

CANCEL OK

Voreinstellungen der IDE angepaßt

Aufruf über Menü (Arduino/Datei > Einstellungen) oder STRG + Komma

Einstellungen

Einstellungen

Netzwerk

Dateipfad des Sketchbooks:
c:\Users\Joachim\Documents\Arduino

DURCHSUCHEN

☐ Dateien im Sketch zeigen

Editor Schriftgröße: 14

Größe der Benutzeroberfläche: ☒ Automatisch 100 %

Farbdesign: Light

Editorsprache: Deutsch (Erneutes Laden erforderlich)

Compiler-Meldungen anzeigen beim ☒ Kompilieren ☒ Hochladen

Compiler-Meldungen Mehr

☒ Code nach Hochladen überprüfen

☐ Automatisch speichern

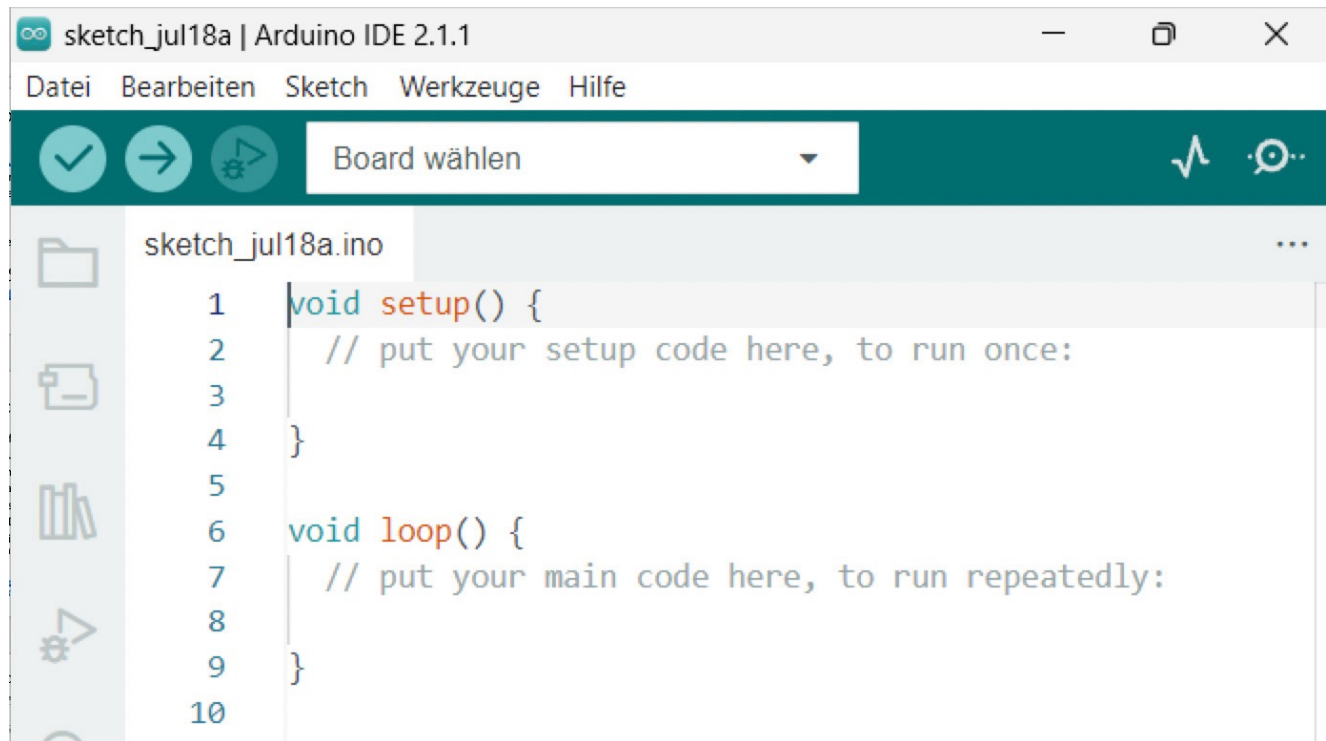
☐ Schnelle Editor Vorschläge

Zusätzliche Boardverwalter-URLs: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json,https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

ABBRECHEN

OK

Erster leerer „Sketch“



Klammern im C-Code

- **geöffnete Klammern** müssen immer auch geschlossen werden (kann auch erst mehrere Zeilen später erfolgen)
- **{ }** **Geschweifte Klammern**, umschließen einen Codeblock
- **()** **Runde Klammern**, Funktionsaufrufe bzw. Rechenpriorität
- **[]** **Eckige Klammern**, Index in einem Array

Kommentare in C-Code

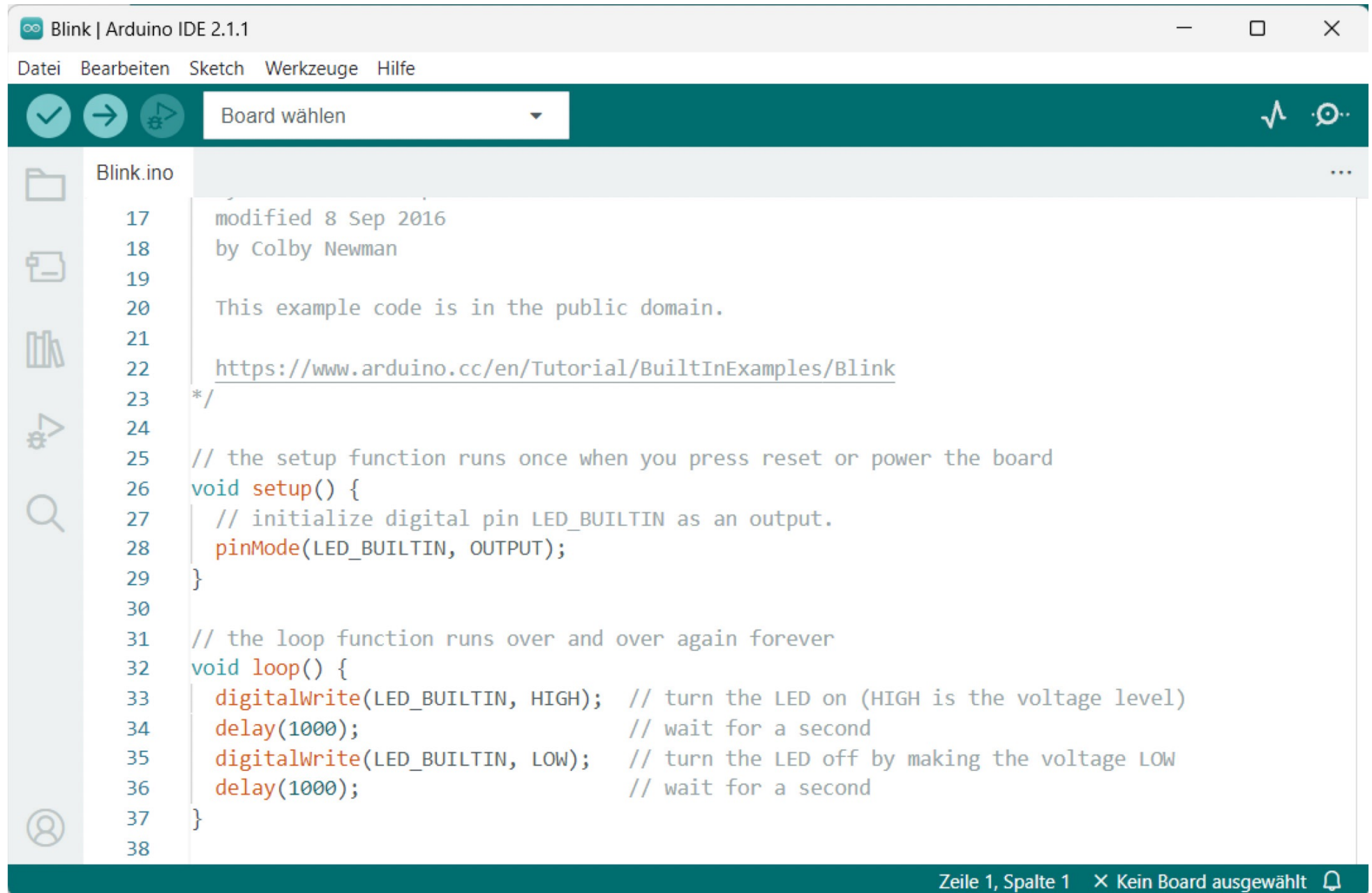
```
// Kommentar bis Zeilenende  
  
/* ab hier  
Mehrzeiliger Kommentar  
bis zu */
```

Tastatur-Eingabe der Sonderzeichen

	Mac	Windows
Eckige Klammer auf	[alt + 5	Alt Gr + 8
Eckige Klammer zu] alt + 6	Alt Gr + 9
„Pipe“ (senkr. Strich)	alt + 7	Alt Gr + >
Geschweifte Klammer auf	{ alt + 8	Alt Gr + 7
Geschweifte Klammer zu	} alt + 9	Alt Gr + 0

Beispiel-Code „Blink“

Arduino IDE Menü: „Datei“ -> „Beispiele“ -> „01.Basics“ -> „Blink“



```
Blink | Arduino IDE 2.1.1
Datei Bearbeiten Sketch Werkzeuge Hilfe

✓ → ⏏ Board wählen

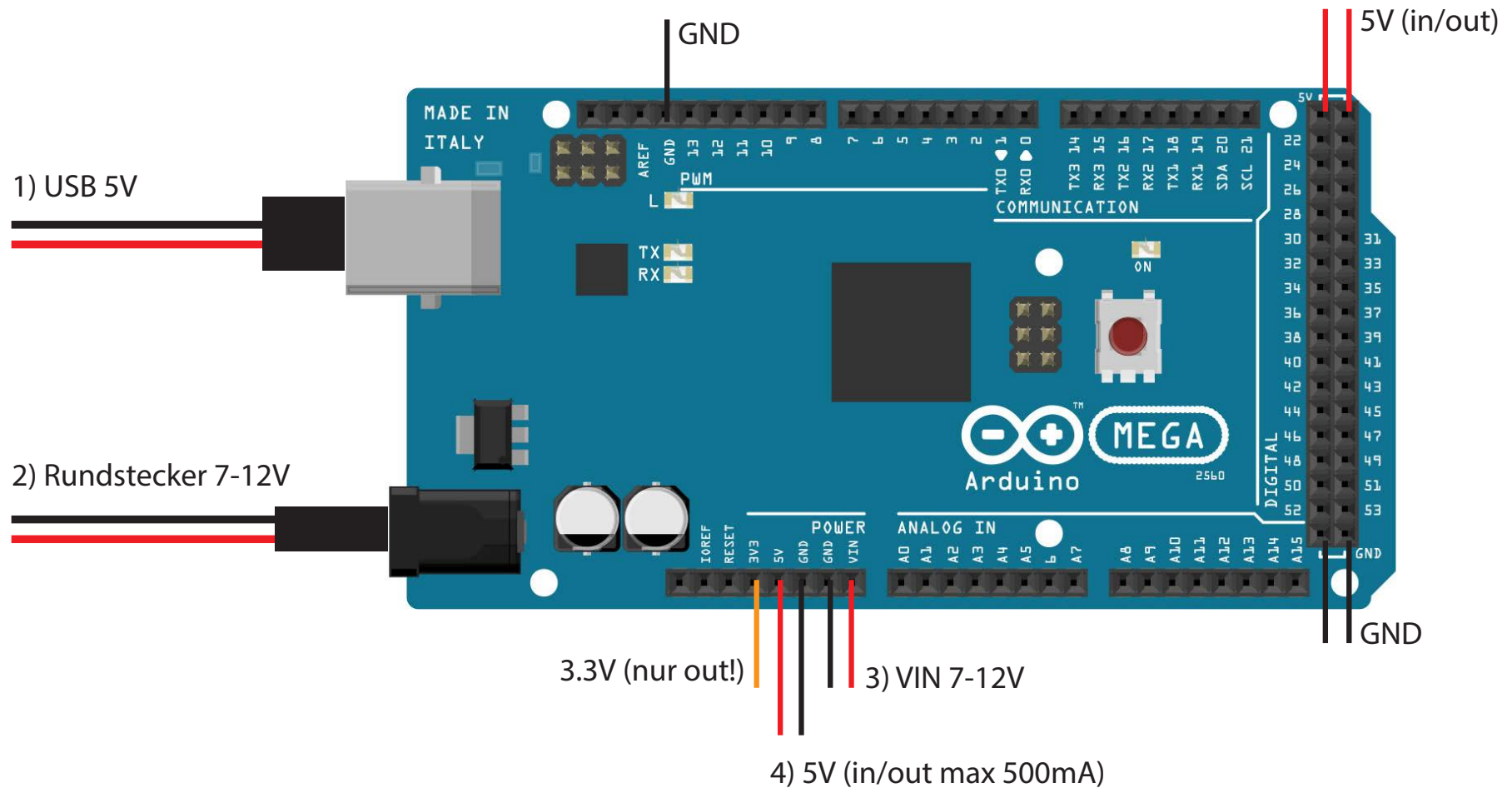
Blink.ino
17 modified 8 Sep 2016
18 by Colby Newman
19
20 This example code is in the public domain.
21
22 https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36   delay(1000); // wait for a second
37 }
38

Zeile 1, Spalte 1 × Kein Board ausgewählt
```


void	reserviertes „keyword“, bedeutet „leer“/“nichts“, Funktion gibt nichts zurück
setup()	spezielle Funktion, wird einmal beim Start des Arduinos ausgeführt
pinMode()	vordefinierte Funktion, legt die Funktionsweise eines GPIO-Pins fest
LED_BUILTIN	vordefinierte Konstante, hat beim Mega den Wert 13 pinMode(LED_BUILTIN, OUTPUT) ist also dasselbe wie pinMode(13, OUTPUT)
OUTPUT	vordefinierte Konstante, der LED-Pin soll als Ausgang verwendet werden
loop()	spezielle Funktion, wird nach setup() in einer Endlos-Schleife ausgeführt
HIGH / LOW	vordefinierte Konstanten (1 und 0), entspricht Spannungsspiegel +5V / 0V
digitalWrite()	vordefinierte Funktion, setzt den Ausgangspegel eines Pins digitalWrite(LED_BUILTIN, HIGH); ist also dasselbe wie digitalWrite(13, 1);
delay(n)	vordefinierte Funktion, hält die Ausführung des Codes um n Millisekunden an (1 Millisekunde = 1/1000 Sekunde)

- Alle Code-Zeilen werden mit einem Strichpunkt abgeschlossen, sonst Fehler beim Kompilieren
- Code-Blöcke { } werden **nicht** mit einem Strichpunkt abgeschlossen

Spannungsversorgung (5-12V)

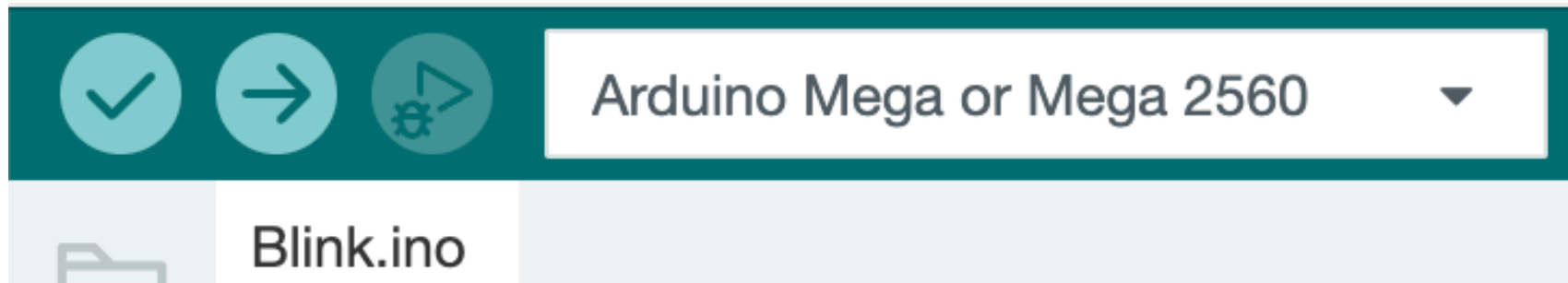


Beim Arduino MEGA beträgt die Spannung an den Input/Output-Pins 5V!
Vorsicht beim Verbinden von Sensoren u.a., die mit 3.3V arbeiten

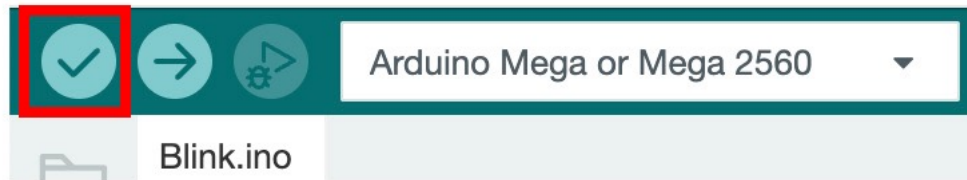
NIE gleichzeitig mehrere Spannungsversorgungen anschließen!

Board und Port auswählen

- Den Arduino Mega mit dem USB-Kabel anstecken
- Einen Augenblick warten, dann in der IDE im Menü „Werkzeuge“ -> „Board“ -> „Arduino AVR Boards“ -> „Arduino Mega oder Mega 2560“ auswählen
- Im Menü „Werkzeuge“ -> „Port“ den seriellen Port wählen (unter Windows z.B. „COM8“, unter Mac z.B. “/dev/cu.usbmodem....“)
- Bei mehreren Ports den Mega ab- und wieder anstecken zur Kontrolle, welcher Port verschwindet/wieder erscheint

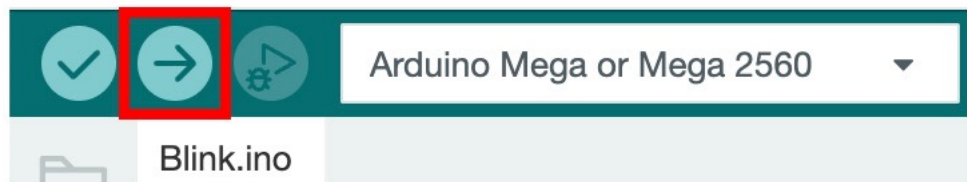


Code kompilieren und flashen



Ein Klick auf den „Haken“-Button kompiliert den Code
(eine Binärdatei, die der Arduino ausführen kann, wird erstellt)

Im Ausgabefenster erscheinen die Details in weißer Schrift. Rote Zeilen sind Warnungen und Fehler

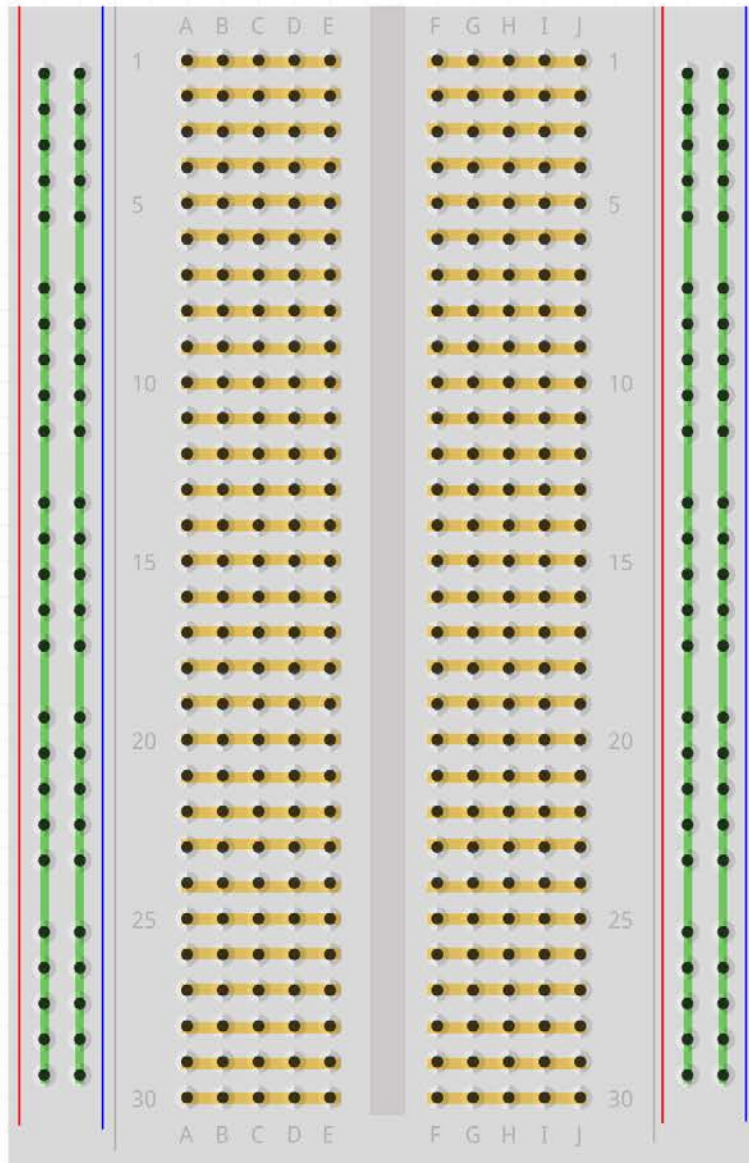


Ein Klick auf den „Pfeil nach rechts“-Button flasht den Code
(die Binärdatei wird an den Arduino übertragen, dort dauerhaft gespeichert und anschließend auf dem Arduino ausgeführt)

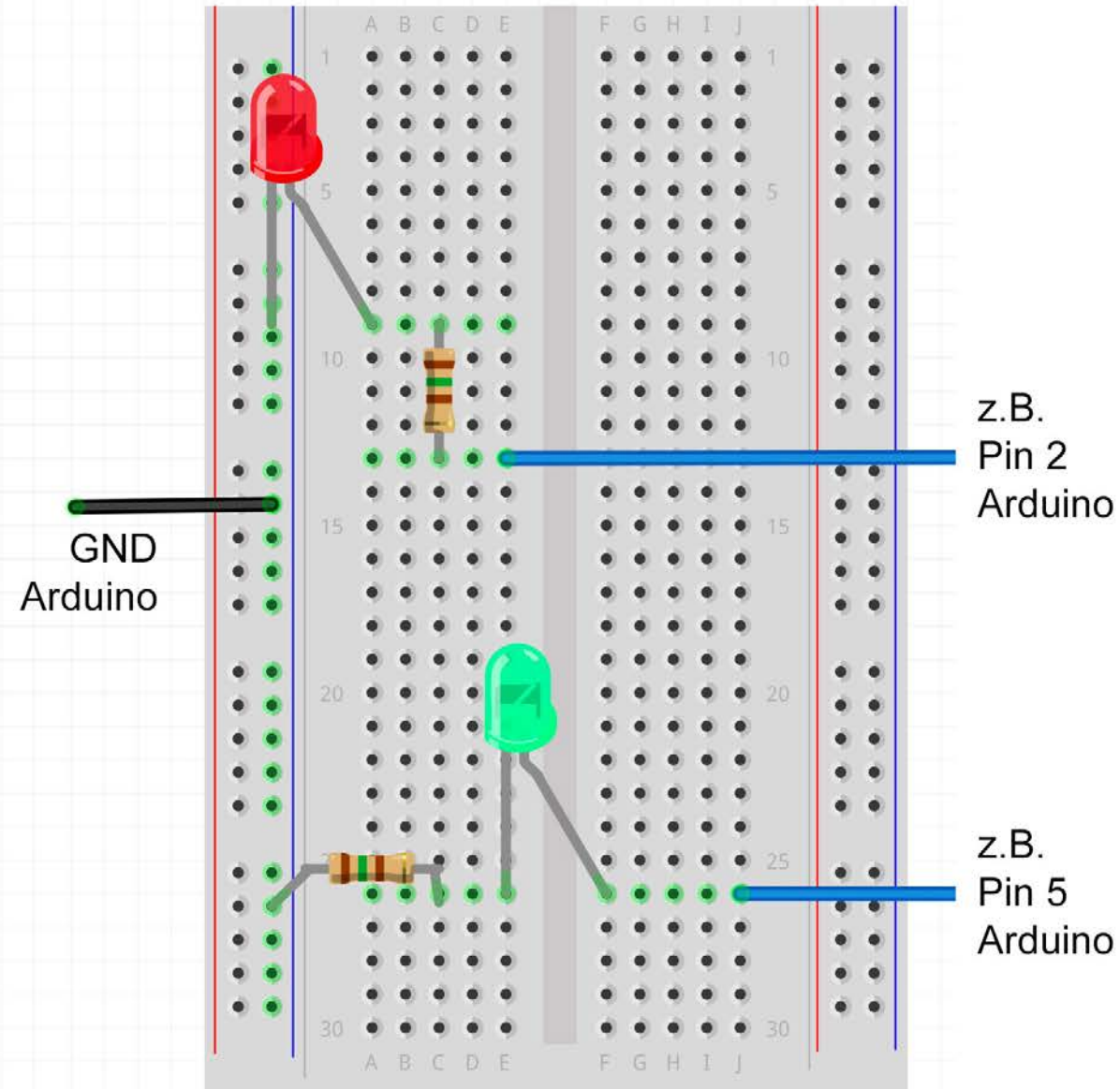
Im Ausgabefenster wird der Flash-Vorgang in (unglücklicherweise) roter Schrift ausgegeben

Steckbrett ("Breadboard")

Interne Verbindungen der Steckfedern

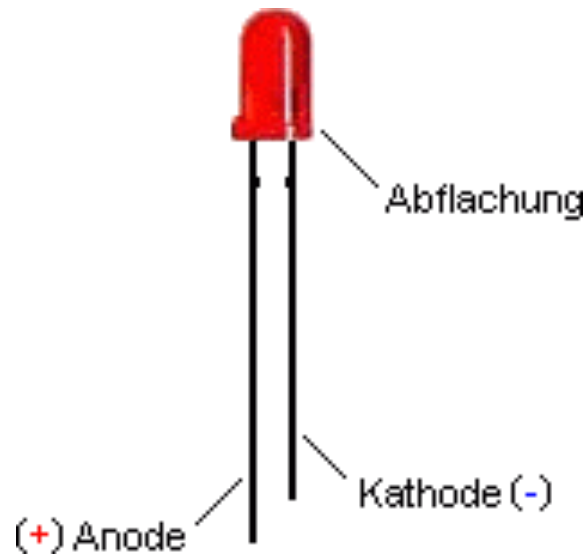


Beispiele für mögliche Schaltungen



Blinken externer LEDs

LEDs haben eine Polung



Anschließen der LED(s)
an den Arduino Mega
(hier z.B. an Pin 2 und 5
mit 150 Ohm Widerstand)

Aufgabe: 2-5 LEDs anschließen
Blinken/Leuchten lassen
Ampel, Blaulicht, Lauflicht etc



Widerstand 150 Ohm

