



# Digitalelektronik 1

## *Digitale Ausgabe*

Stefan Rothe

2015-02-17

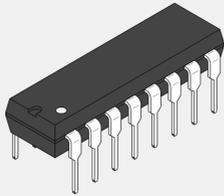


# Unsere Abstraktionsebenen

```
for (int i  
    a[i] =
```

Praktische Informatik

Programmierung in einer Hochsprache



Technische Informatik

Funktionsweise von Mikrochips



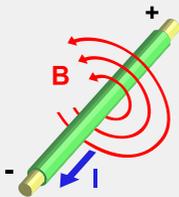
Logik / Digitalelektronik

Logikgatter als Realisierung Boolescher Funktionen



Elektronik

Stromkreise und elektronische Bauteile



Elektrodynamik

elektrische Ladungen und elektromagnetische Felder



# Stromkreis

In einem Stromkreis fließt Strom von einer **Anode** zu einer **Kathode** einer **Spannungsquelle** (z.B. einer Batterie).

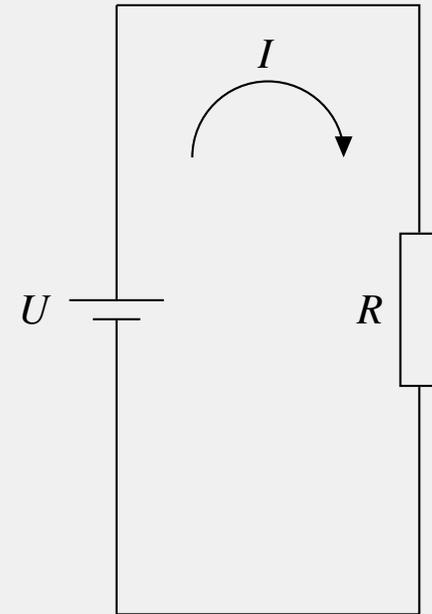
Dabei gilt das **Ohmsche Gesetz**:

$$U = R \cdot I$$

$U$  = Spannung Volt

$R$  = Widerstand Ohm

$I$  = Strom Ampere



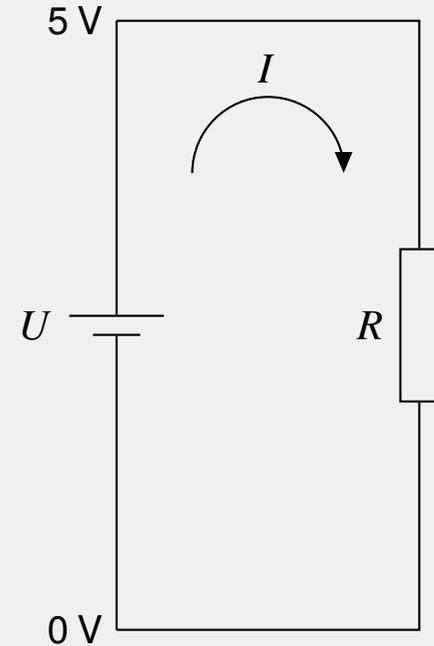


# Digitalschaltung

Eine digitale Schaltung ist ein Stromkreis, bei welchem die **Spannung konstant** und der **Strom klein** ist.

■ 5 V = **HIGH** = 1

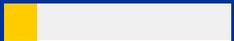
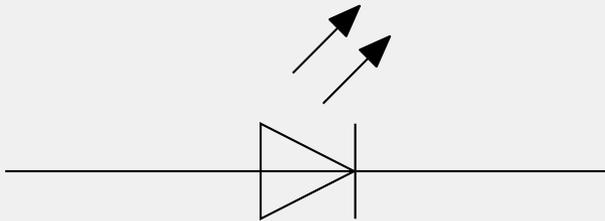
■ 0 V = **LOW** = 0





# Leuchtdiode

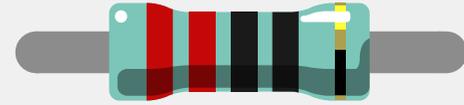
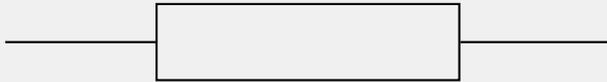
- Strom fließt nur in eine Richtung (hier von links nach rechts)
- häufige Bezeichnung: **LED** von engl. *light emitting diode*
- Erzeugt bei Strom Licht
- Typischer Strom: 20 mA
- Typische Spannung (rot): 1.6 V
- Symbol:





# Widerstand

- realisiert ohmschen Widerstand in Schaltungen
- Farbringe geben Widerstandswert in Ohm an
- Symbol:





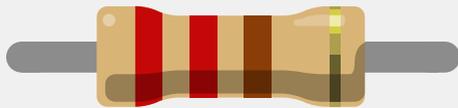
# Widerstand: Farbkodierung (4 Ringe)

Der Widerstandswert in  $\Omega$  eines Widerstands wird durch die Farbringe gekennzeichnet. Jeder Farbe ist eine Zahl zwischen -2 und 9 zugeordnet.

Bei Widerständen mit vier Ringen gilt folgende Regel:

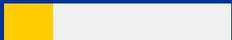
- Der **erste Ring** gibt die Zehnerstelle  $b$  an.
- Der **zweite Ring** gibt die Einerstelle  $c$  an.
- Der **dritte Ring** gibt den Exponenten  $d$  an.

Der Widerstandswert ist  $(10b + c) \cdot 10^d \Omega$ .



Für diesen Widerstand (rot, rot, braun) ergibt sich:

Farbe	Zahl
Silber	-2
Gold	-1
Schwarz	0
Braun	1
Rot	2
Orange	3
Gelb	4
Grün	5
Blau	6
Violett	7
Grau	8
Weiss	9





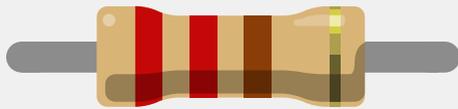
# Widerstand: Farbkodierung (4 Ringe)

Der Widerstandswert in  $\Omega$  eines Widerstands wird durch die Farbringe gekennzeichnet. Jeder Farbe ist eine Zahl zwischen -2 und 9 zugeordnet.

Bei Widerständen mit vier Ringen gilt folgende Regel:

- Der **erste Ring** gibt die Zehnerstelle  $b$  an.
- Der **zweite Ring** gibt die Einerstelle  $c$  an.
- Der **dritte Ring** gibt den Exponenten  $d$  an.

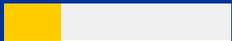
Der Widerstandswert ist  $(10b + c) \cdot 10^d \Omega$ .



Für diesen Widerstand (rot, rot, braun) ergibt sich:

$$(20 + 2) \cdot 10^1 = 220 \Omega.$$

Farbe	Zahl
Silber	-2
Gold	-1
Schwarz	0
Braun	1
Rot	2
Orange	3
Gelb	4
Grün	5
Blau	6
Violett	7
Grau	8
Weiss	9



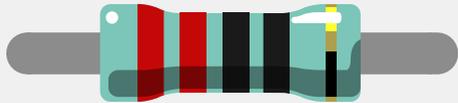


# Widerstand: Farbkodierung (5 bis 6 Ringe)

Bei Widerständen mit fünf oder sechs Ringen gilt folgende Regel:

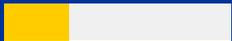
- Der **erste Ring** gibt die Hunderterstelle  $a$  an.
- Der **zweite Ring** gibt die Zehnerstelle  $b$  an.
- Der **dritte Ring** gibt die Einerstelle  $c$  an.
- Der **vierte Ring** gibt den Exponenten  $d$  an.

Der Widerstandswert ist  $(100a + 10b + c) \cdot 10^d \Omega$ .



Für diesen Widerstand (rot, rot, schwarz, schwarz) ergibt sich:

Farbe	Zahl
Silber	-2
Gold	-1
Schwarz	0
Braun	1
Rot	2
Orange	3
Gelb	4
Grün	5
Blau	6
Violett	7
Grau	8
Weiss	9



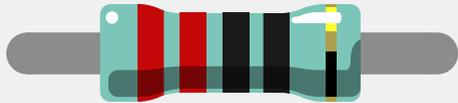


# Widerstand: Farbkodierung (5 bis 6 Ringe)

Bei Widerständen mit fünf oder sechs Ringen gilt folgende Regel:

- Der **erste Ring** gibt die Hunderterstelle  $a$  an.
- Der **zweite Ring** gibt die Zehnerstelle  $b$  an.
- Der **dritte Ring** gibt die Einerstelle  $c$  an.
- Der **vierte Ring** gibt den Exponenten  $d$  an.

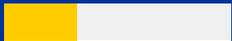
Der Widerstandswert ist  $(100a + 10b + c) \cdot 10^d \Omega$ .



Für diesen Widerstand (rot, rot, schwarz, schwarz) ergibt sich:

$$(200 + 20 + 0) \cdot 10^0 = 220 \Omega$$

Farbe	Zahl
Silber	-2
Gold	-1
Schwarz	0
Braun	1
Rot	2
Orange	3
Gelb	4
Grün	5
Blau	6
Violett	7
Grau	8
Weiss	9



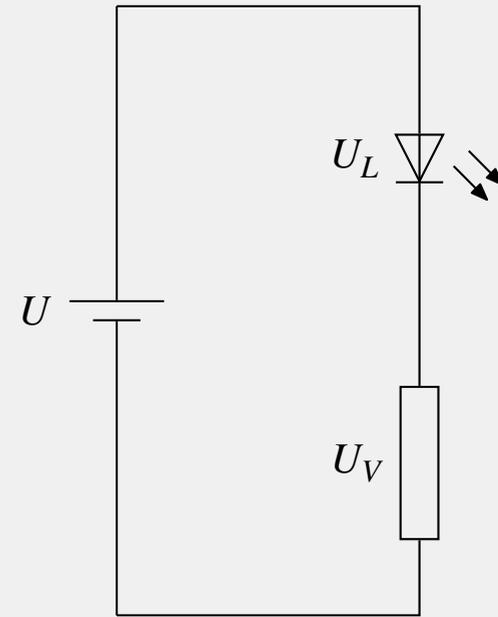


# Berechnung Vorwiderstand

Um den Strom klein zu halten, muss eine Schaltung mit einer Leuchtdiode durch einen Widerstand ergänzt werden. Der benötigte Widerstand berechnet sich so:

$$R_V = \frac{U - U_L}{I}$$

$R_V$	=	Widerstand	Ohm
$U_V$	=	Spannung über dem Widerstand	Volt
$I$	=	Strom	Ampere
$U$	=	Gesamtspannung	Volt
$U_L$	=	Spannung über der Leuchtdiode	Volt





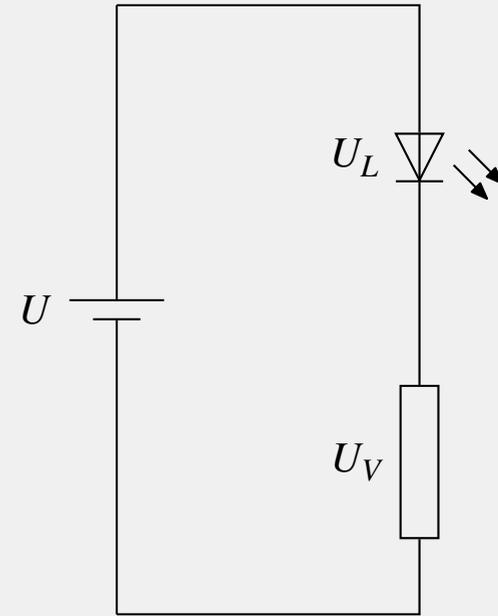
# Berechnung Vorwiderstand

Um den Strom klein zu halten, muss eine Schaltung mit einer Leuchtdiode durch einen Widerstand ergänzt werden. Der benötigte Widerstand berechnet sich so:

$$R_V = \frac{U - U_L}{I}$$

$R_V$	=	Widerstand	Ohm
$U_V$	=	Spannung über dem Widerstand	Volt
$I$	=	Strom	Ampere
$U$	=	Gesamtspannung	Volt
$U_L$	=	Spannung über der Leuchtdiode	Volt

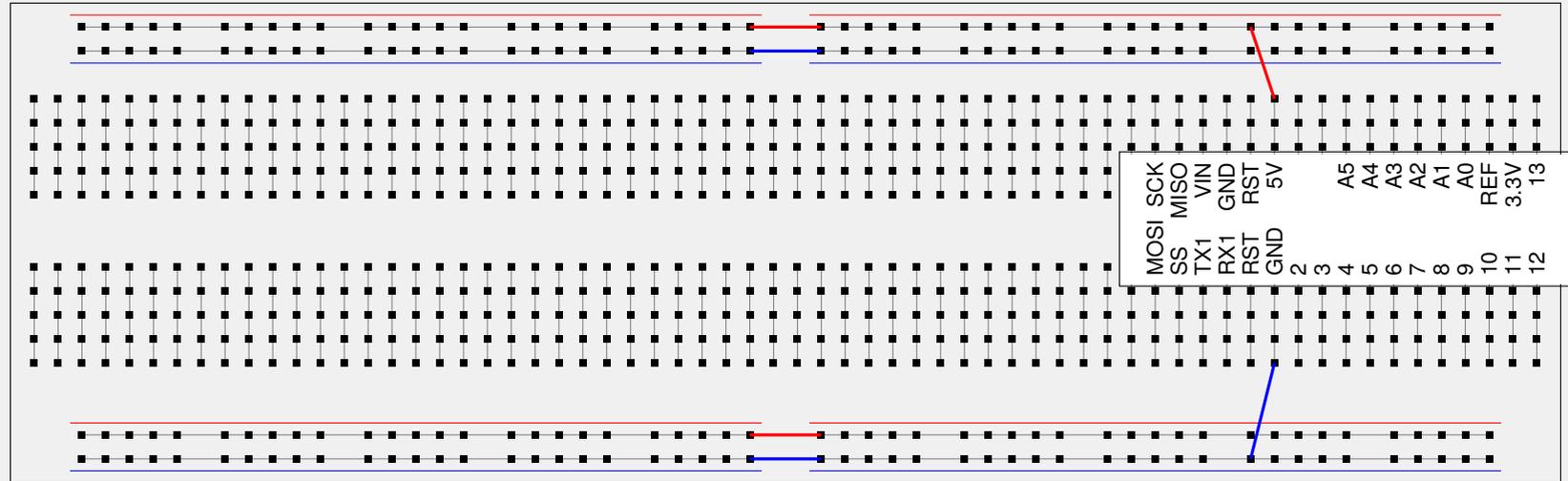
Berechnen Sie den benötigten Vorwiderstand für eine rote Leuchtdiode (20 mA, 1.6 V) bei einer 5 V-Spannungsquelle.





# Steckplatine (*Breadboard*)

Die Steckplatine wird so gedreht, dass der Mikrocontroller **rechts** liegt.



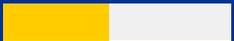
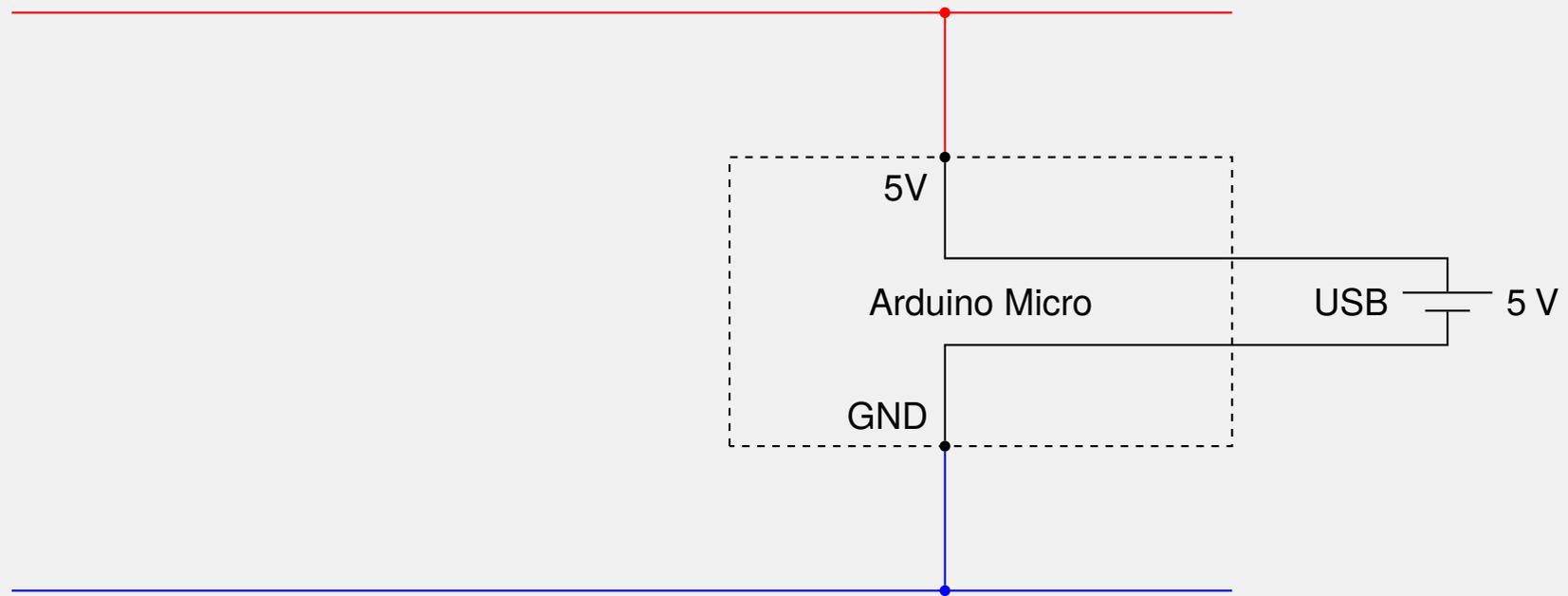
- An der **obersten** Reihe (rot) liegt eine Spannung von 5 V an.
- An der **untersten** Reihe (blau) liegt 0 V an (Masse).
- Die zweitoberste und zweitunterste Reihe sind **nicht belegt**.





# Steckplatine: Spannungsversorgung

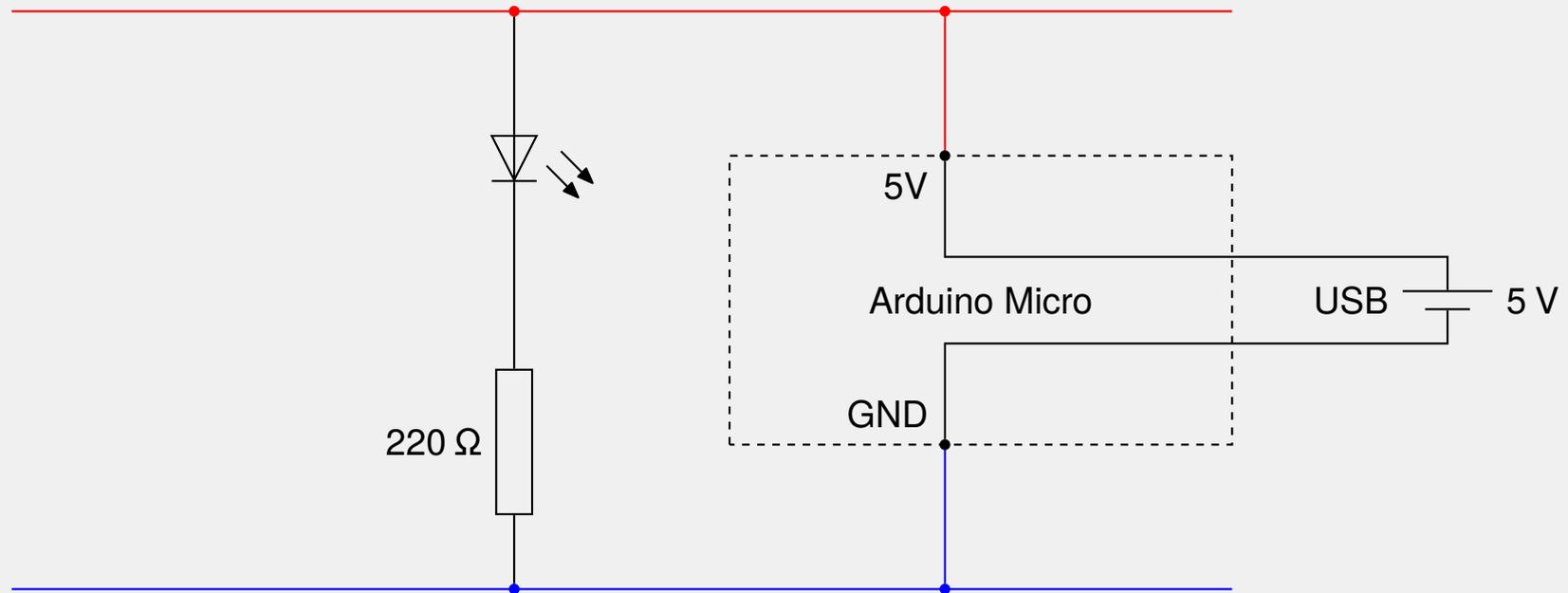
Die Spannungsversorgung erfolgt über den USB-Anschluss des Mikrocontrollers.





# Aufgabe 1: Leuchtdiode

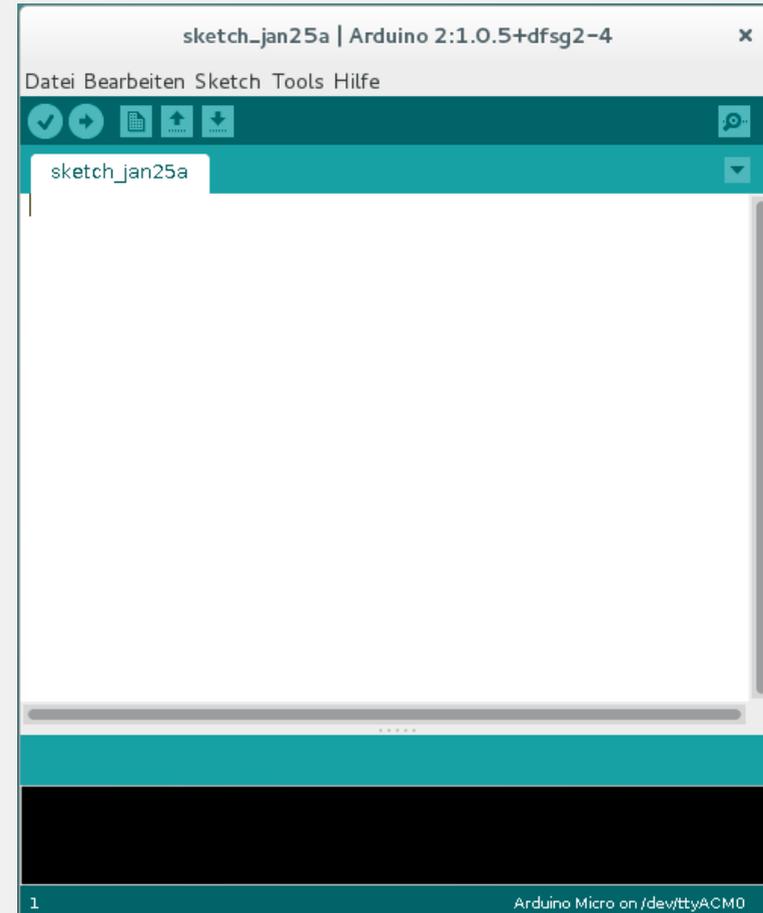
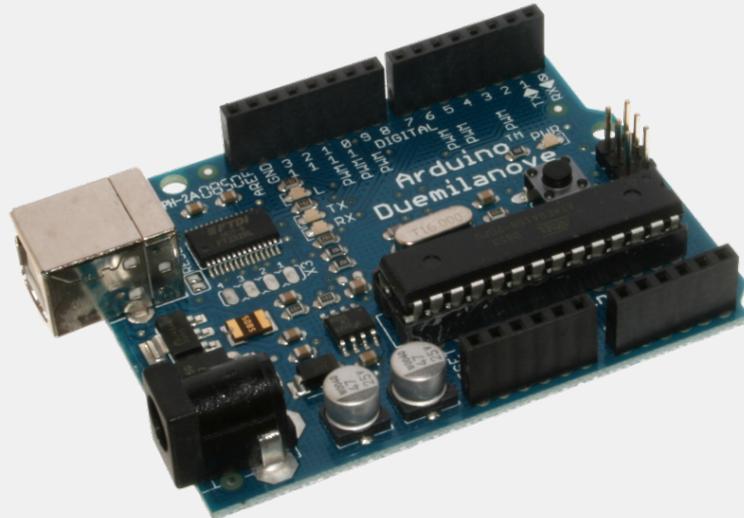
Bringen Sie eine Leuchtdiode zum Leuchten, indem Sie auf der Steckplatine den untenstehenden Schaltkreis realisieren.





# Arduino-Plattform

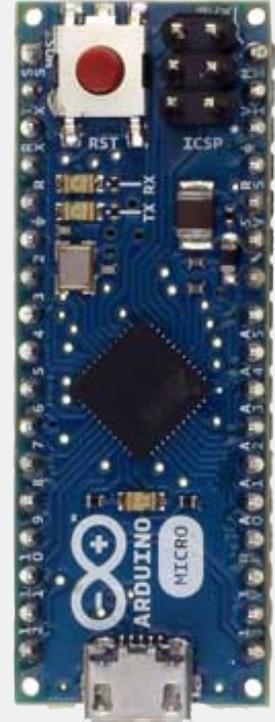
- Open Source-Mikrocontroller-Plattform
- einfach anzuwendende Hardware und Software
- Programmierbar in C/C++
- USB-Anschluss





# Arduino Micro

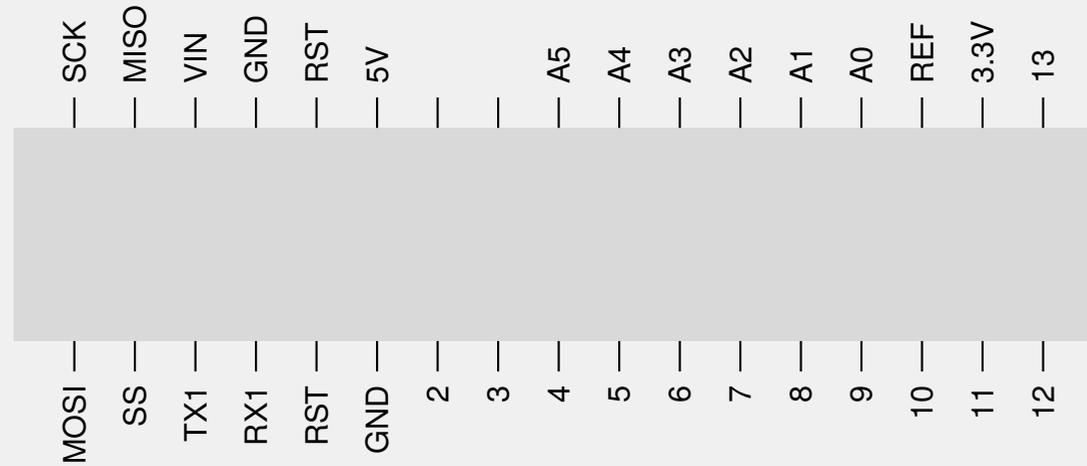
- Atmel AVR 8-Bit-Mikrocontroller (ATmega32u4)
- 14 digitale Ein-/Ausgänge
- davon unterstützen acht Pulsweitenmodulation (PWM)
- sechs analoge Eingänge (sechs weitere auf digitalen Pins)
- SPI-Schnittstelle
- USB-Anschluss





# Arduino Micro: Pinbelegung

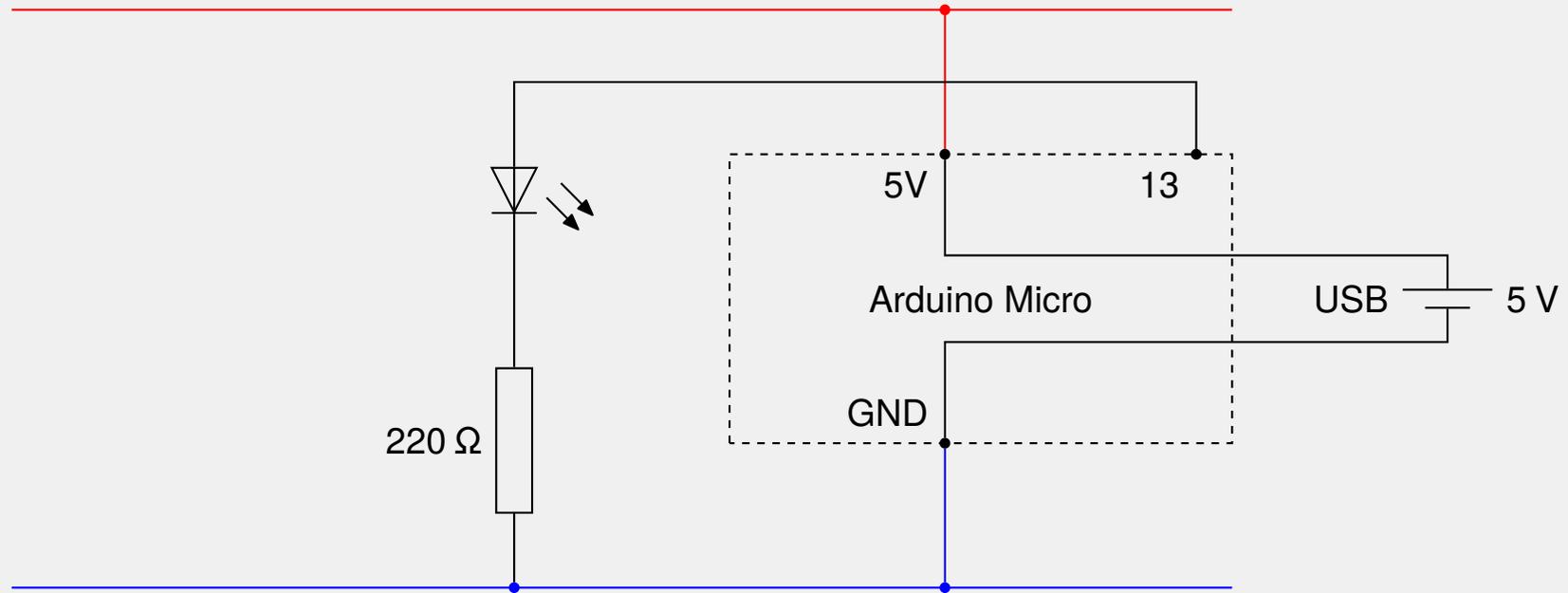
- Digitale Ein-/Ausgänge: TX1, RX1, 2 bis 13
- Analoge Eingänge: A0 bis A5, REF
- SPI-Schnittstelle: MOSI, SCK, MISO
- Spannungsversorgung: VIN, 5V, 3.3V, GND





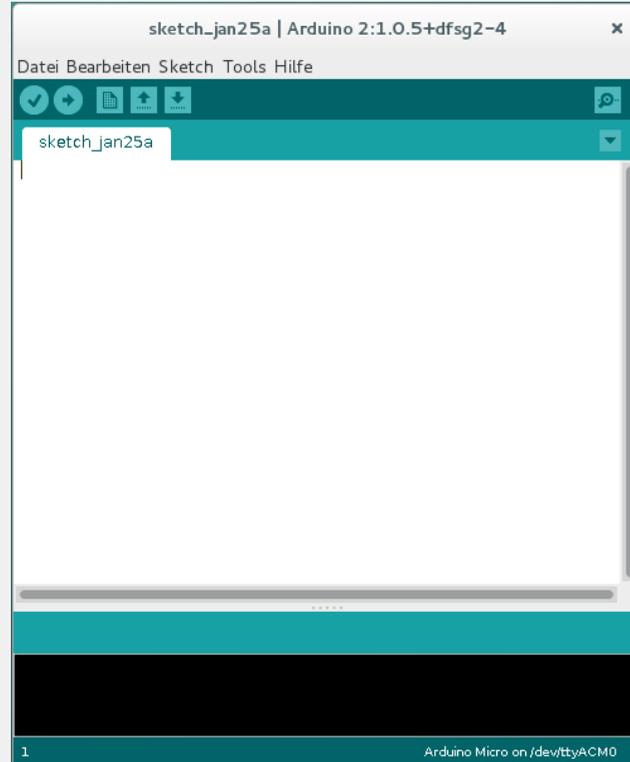
# Blinkende Leuchtdiode - Schaltung

Lassen Sie die Leuchtdiode blinken.





# Arduino IDE





# Arduino IDE: Auswahl des Boards

The screenshot shows the Arduino IDE interface with the 'Tools' menu open. The 'Board' option is selected, and a list of available boards is displayed. The boards listed are:

- Arduino Uno
- Arduino Duemilanove w/ ATmega328
- Arduino Diecimila or Duemilanove w/ ATmega168
- Arduino Nano w/ ATmega328
- Arduino Nano w/ ATmega168
- Arduino Mega 2560 or Mega ADK
- Arduino Mega (ATmega1280)
- Arduino Leonardo
- Arduino Esplora
- Arduino Micro
- Arduino Mini w/ ATmega328

The 'Arduino Micro' option is highlighted with a blue background and a mouse cursor is pointing at it.



# Blinkende Leuchtdiode - Programm 1

- `setup()` wird beim Einschalten und beim Reset ausgeführt.
- `loop()` wird nach `setup()` immer zyklisch aufgerufen.

```
void setup() {  
  
}  
  
void loop() {  
  
}
```





# Blinkende Leuchtdiode - Programm 2

```
const Typ Name = Wert;
```

- Mit **const** werden Konstanten definiert
- Namen von Konstanten in Grossbuchstaben
- Alle Werte als Konstanten definieren

```
const int DELAY_MS = 1000;  
const int PIN = 13;  
  
void setup() {  
  
}  
  
void loop() {  
  
}
```



# Blinkende Leuchtdiode - Programm 3

```
pinMode(Pin, OUTPUT);
```

- `pinMode()` legt Modus eines Pins fest
- `OUTPUT` ist eine vordefinierte Konstante
- Wird in `setup()` aufgerufen

```
const int DELAY_MS = 1000;  
const int PIN = 13;  
  
void setup() {  
    pinMode(PIN, OUTPUT);  
}  
  
void loop() {  
  
}
```



# Blinkende Leuchtdiode - Programm 4

```
digitalWrite(Pin, HIGH);  
digitalWrite(Pin, LOW);
```

- **digitalWrite()** schaltet Spannung auf Pin um
- **HIGH** und **LOW** sind vordefinierte Konstanten
- **HIGH** bedeutet 5 V
- **LOW** bedeutet 0 V

```
const int DELAY_MS = 1000;  
const int PIN = 13;  
  
void setup() {  
    pinMode(PIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(PIN, HIGH);  
    digitalWrite(PIN, LOW);  
}
```



# Blinkende Leuchtdiode - Programm 5

```
delay(Millisekunden);
```

- **delay()** wartet Anzahl Millisekunden

```
const int DELAY_MS = 1000;
const int PIN = 13;

void setup() {
    pinMode(PIN, OUTPUT);
}

void loop() {
    digitalWrite(PIN, HIGH);
    delay(DELAY_MS);
    digitalWrite(PIN, LOW);
    delay(DELAY_MS);
}
```



# Arrays

Wie in Java können in C/C++ *Arrays* definiert werden:

```
const int VALUE[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

**ACHTUNG:** In C/C++ stehen die eckigen Klammern hinter dem Namen der Variable, nicht hinter dem Typ wie in Java.

In C/C++ wird die Länge eines *Arrays* mit Hilfe des `sizeof`-Operators ermittelt.

```
const int VALUE_COUNT = sizeof(VALUE) / sizeof(int);
```



# Arrays und Schleifen

Wie in Java kann eine **while**-Schleife verwendet werden, um alle Elemente eines Arrays zu durchlaufen.

So werden beispielsweise mehrere Pins initialisiert, die in einem Array definiert sind:

```
const int PIN[] = { 2, 3, 4, 5 };
const int PIN_COUNT = sizeof(PIN) / sizeof(int);

void setup() {
  int i = 0;
  while (i < PIN_COUNT) {
    pinMode(PIN[i], OUTPUT);
    i = i + 1;
  }
}
```



## Aufgabe 2: Lauflicht

Erstellen Sie ein Lauflicht aus mindestens vier Leuchtdioden.

**Zusatzaufgabe a:** Ändern Sie das Lauflicht so, dass es sich hin und her bewegt.

**Zusatzaufgabe b:** Ändern Sie die Hin- und Herbewegung eine harmonische Schwingung darstellt (d.h. die Abhängigkeit zwischen dem Abstand der leuchtenden LED von der Mitte und der Wartezeit ist eine Sinusfunktion).

### Hinweise:

- Für alle Leuchtdioden kann der gleiche Vorwiderstand verwendet werden.
- Verwenden Sie die **Referenz Digitalelektronik**
- Viele lauffähige Beispielprogramme sind im Menü **Datei / Beispiele** verfügbar.
- Über den Menüpunkt **Hilfe / Referenz** kann die Webseite mit der gesamten Arduino-Befehlsreferenz geöffnet werden.