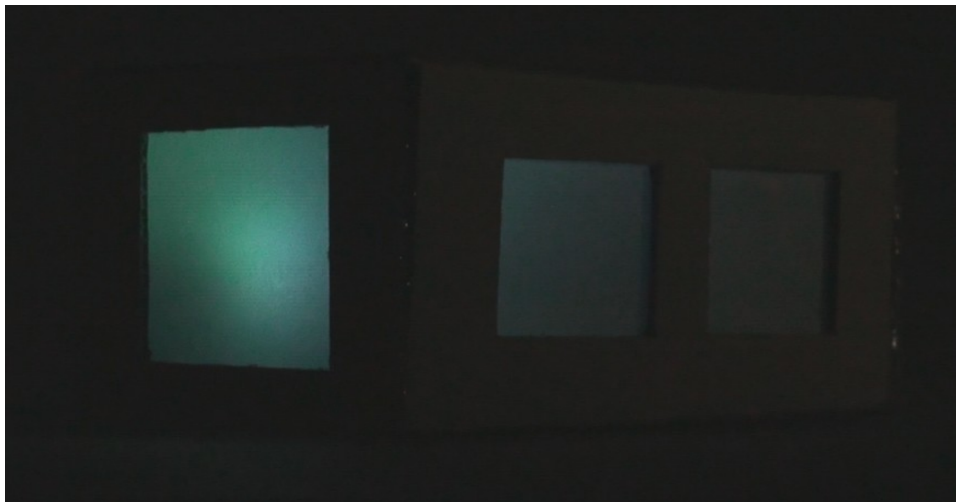
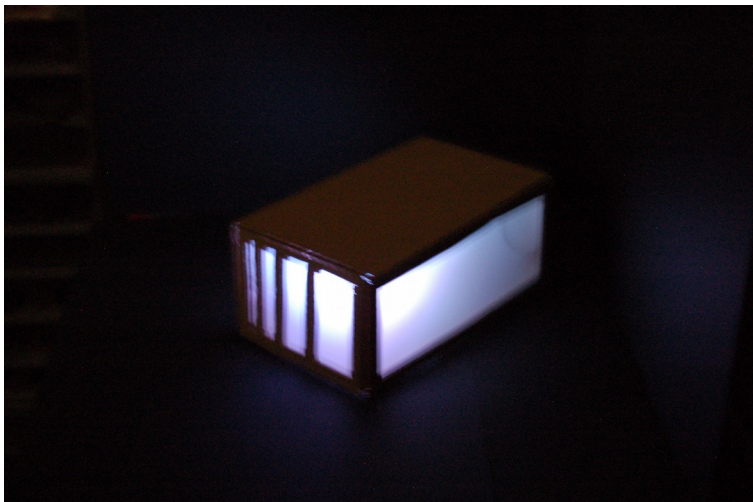
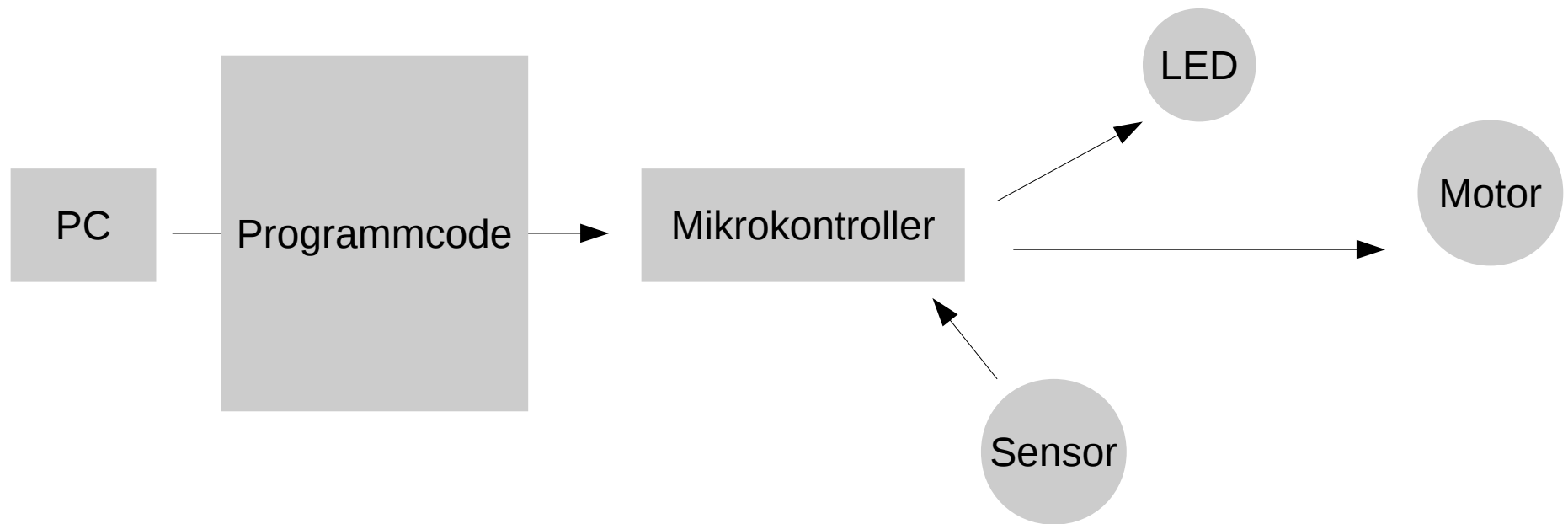


# Streifzüge ins Digitale

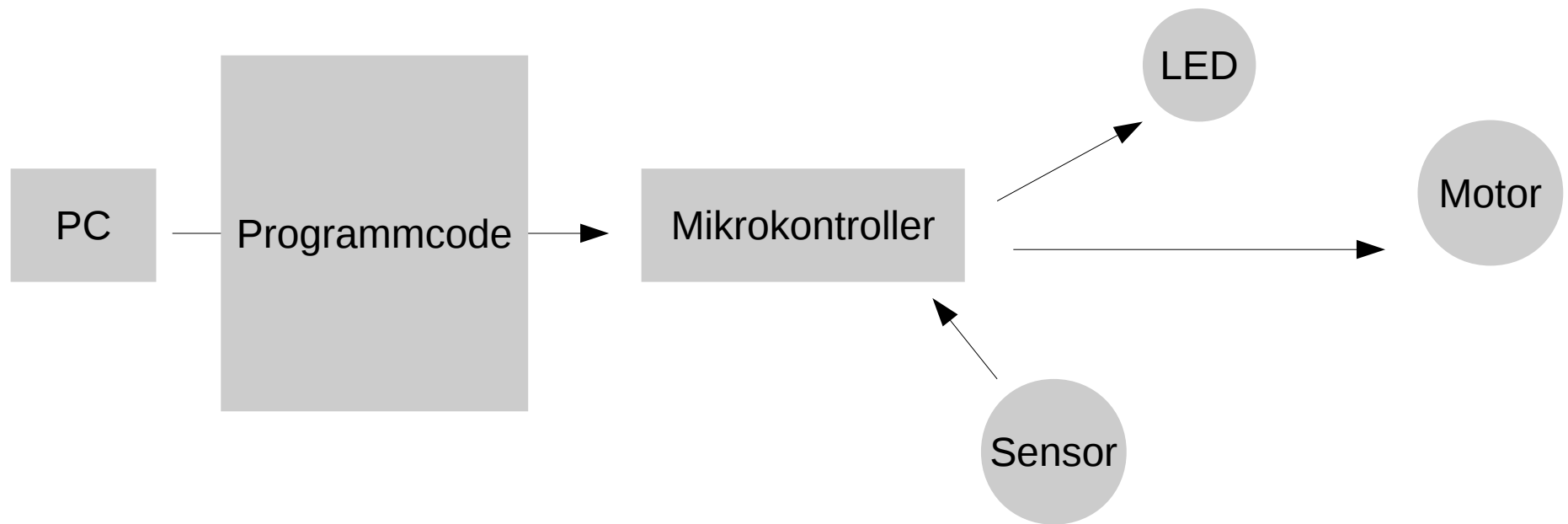
## LED Blink



# Physical Computing



# Physical Computing

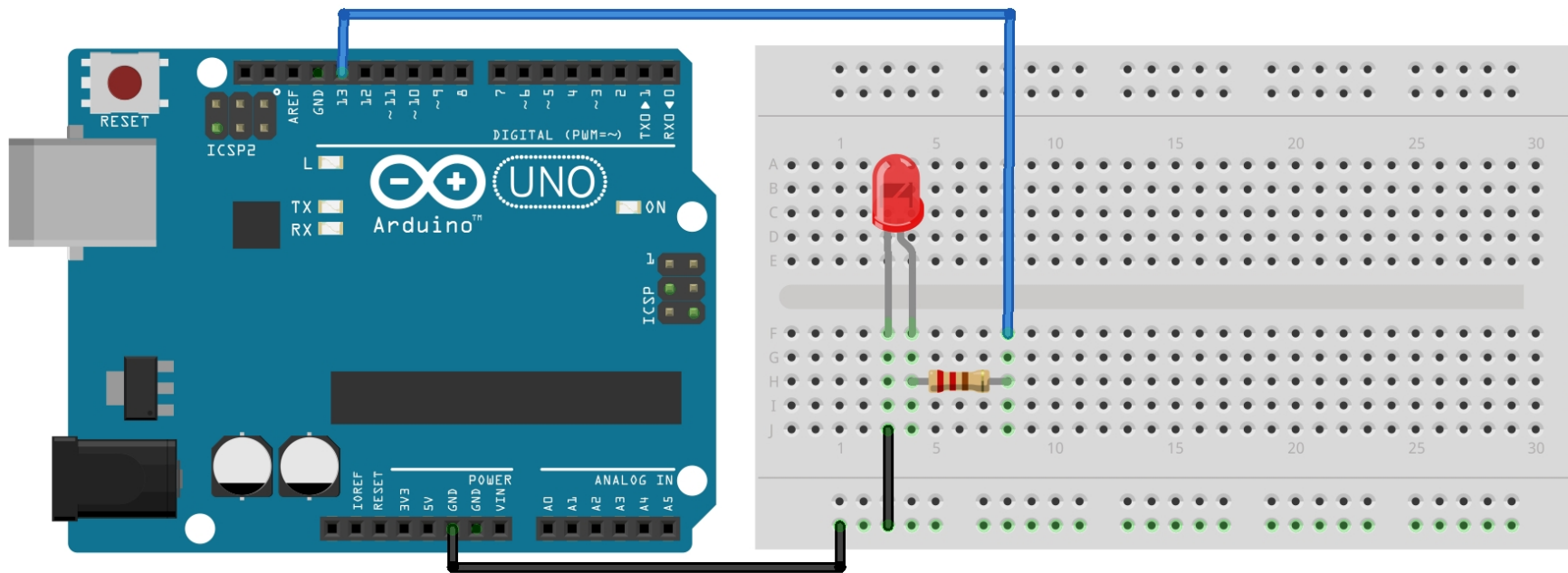


# Verwendete Komponenten

Arduino /  
Mikrokontroller

Stromimpuls (+)

LED

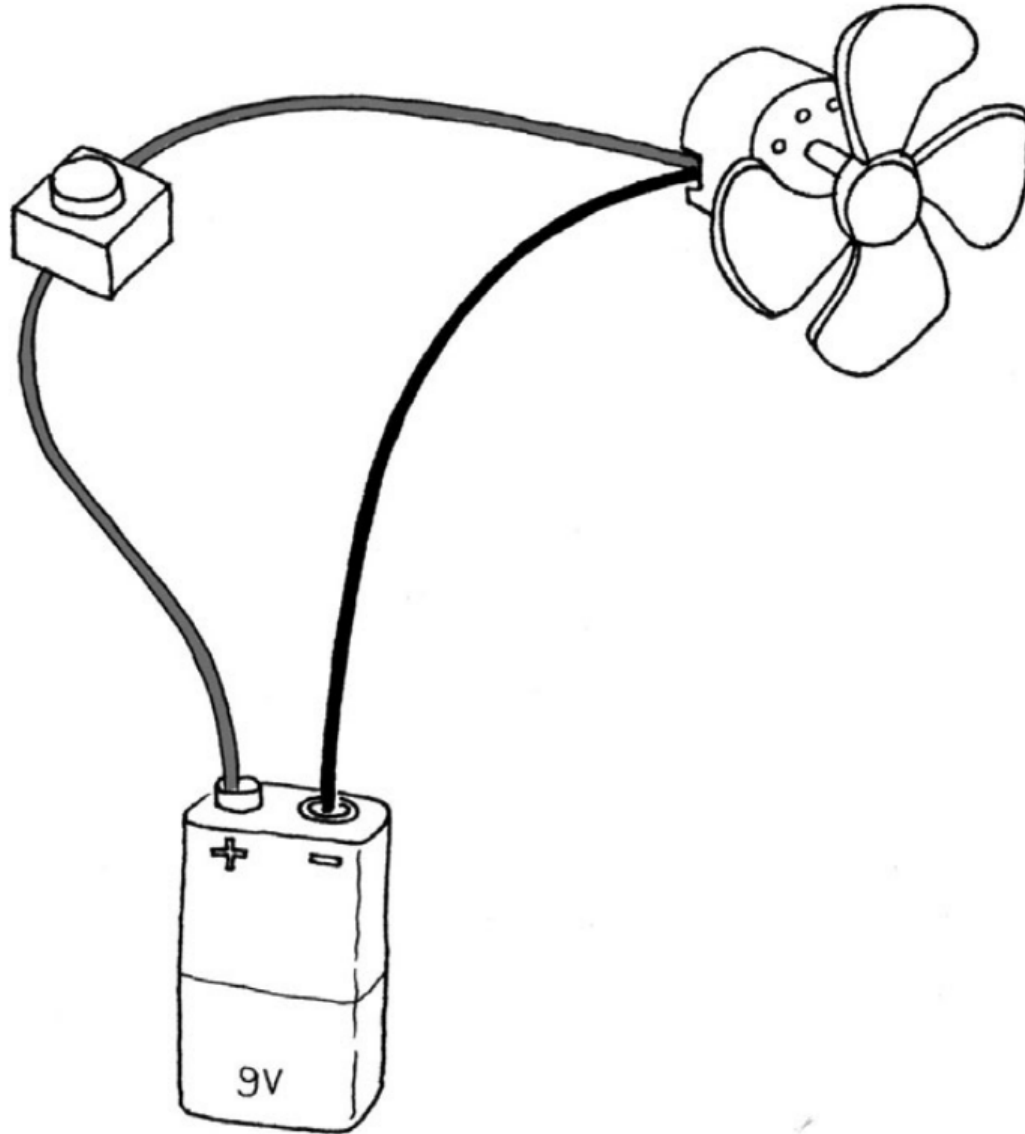


Ground (-)

Widerstand

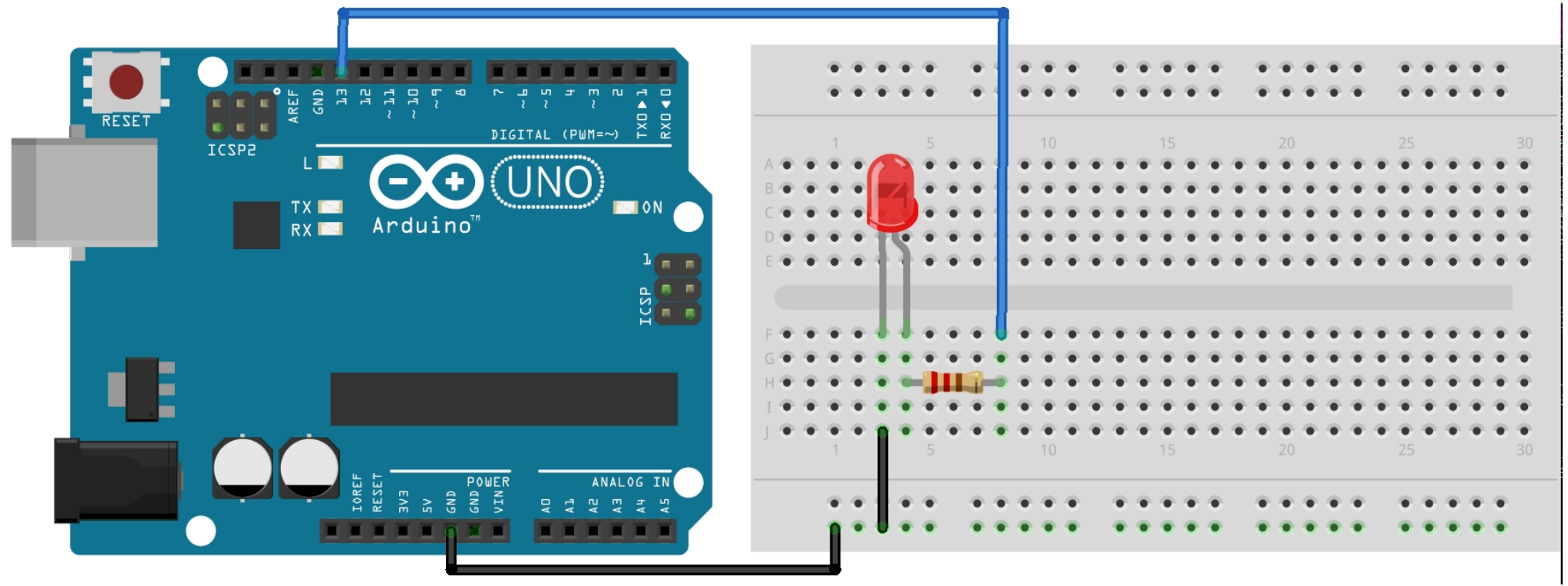
Made with  Fritzing.org

# Grundlagen Elektronik

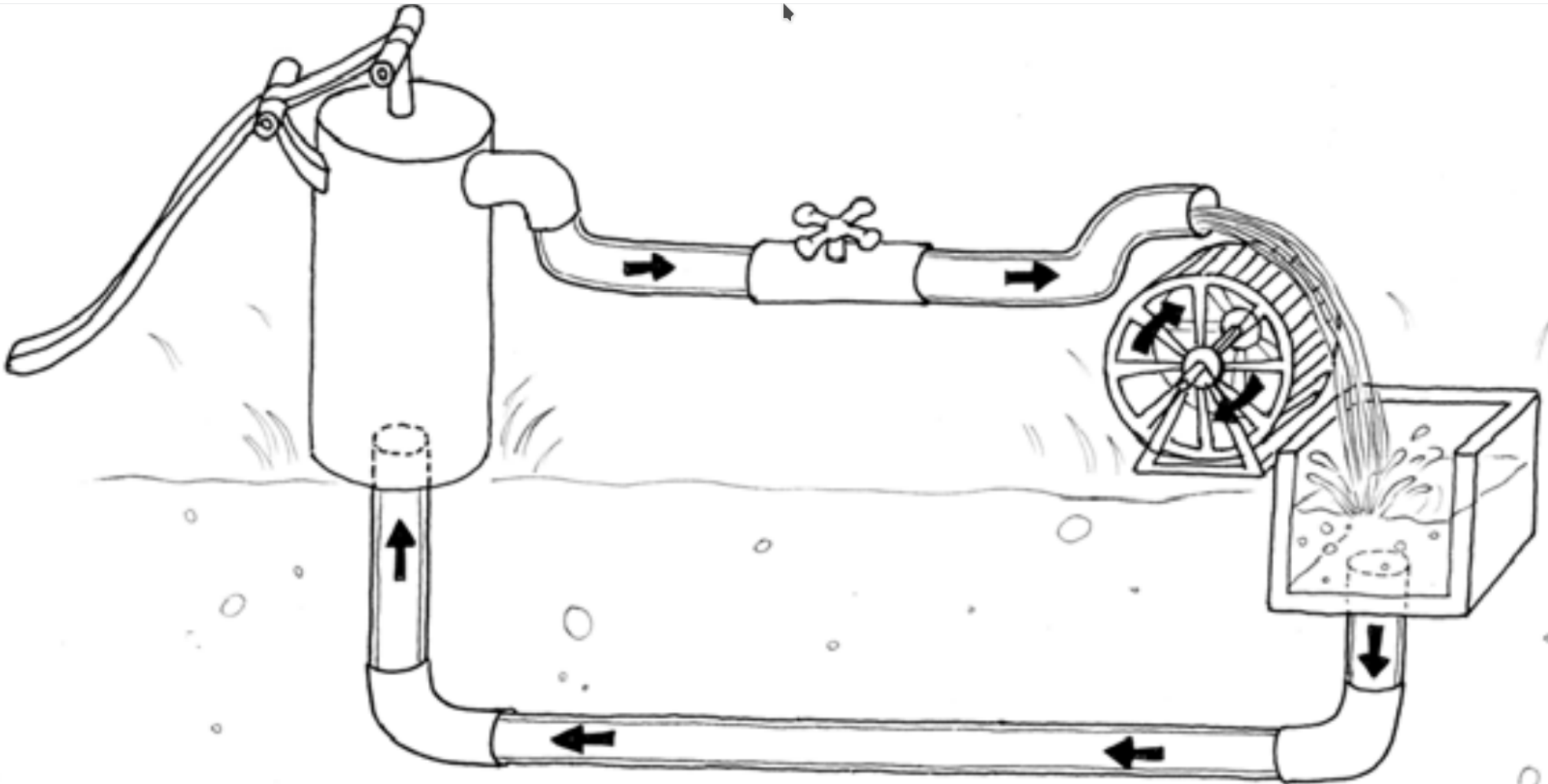


<http://www.mikolaskova.cz/nop/streifzuege>

Illustration Judy Aime Castro aus "Arduino für Einsteiger", Banzi, Shiloh



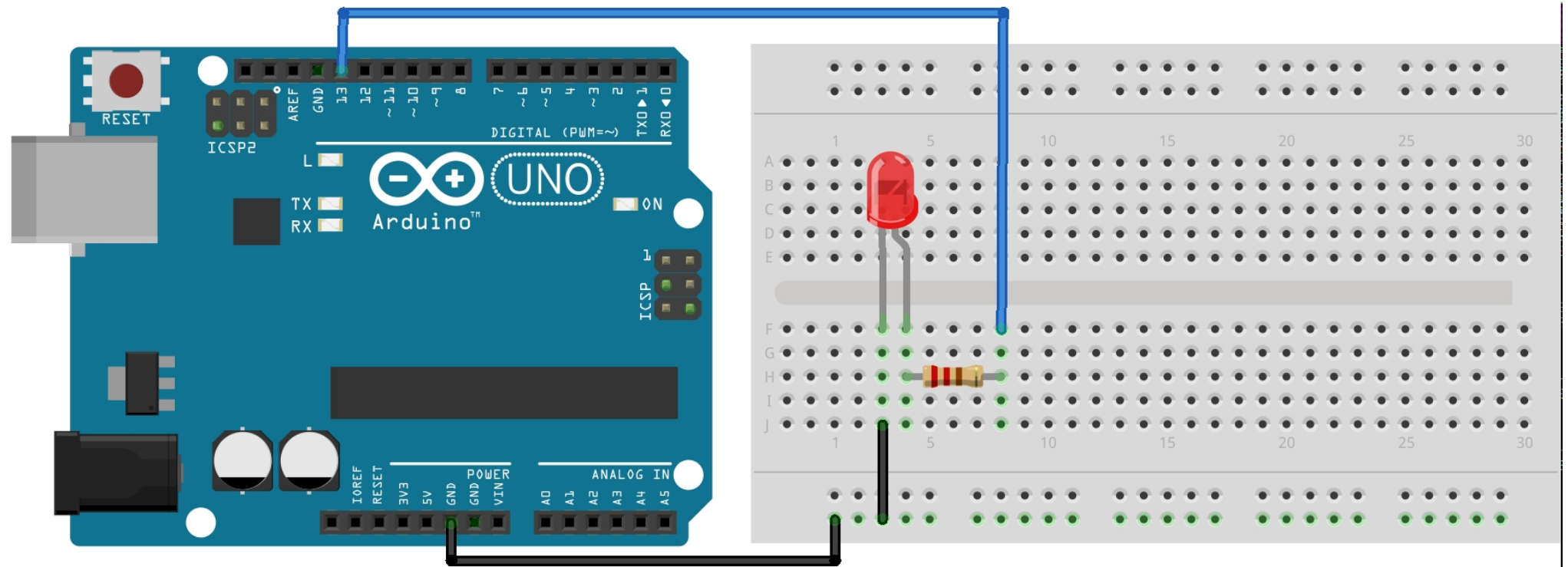
# Grundlagen Elektronik



<http://www.mikolaskova.cz/nop/streifzuege>

Illustration Judy Aime Castro aus "Arduino für Einsteiger", Banzi, Shiloh





Made with  Fritzing.org

# Über Arduino

**Wiring-Platine** von **Hernando Barragan**

Entwickelt als Master-Arbeit **am Interaction Design Institute Ivrea**

*soll auch technisch weniger Versierten den Zugang zur Programmierung und zu Mikrocontrollern erleichtern*

Betreuende Professoren: **Massimo Banzi** and Casey Reas  
(→ Processing)

2005 Massimo Banzi und **David Cuartielles** entwickeln auf Basis von Wiring das erste Arduino Board

Hernando Barragán | Interaction Design Institute Ivrea | June 2004

h.barragan@interaction-ivrea.it

# Wiring: Prototyping Physical Interaction Design

## Thesis Committee

**Massimo Banzi** . Associate Professor

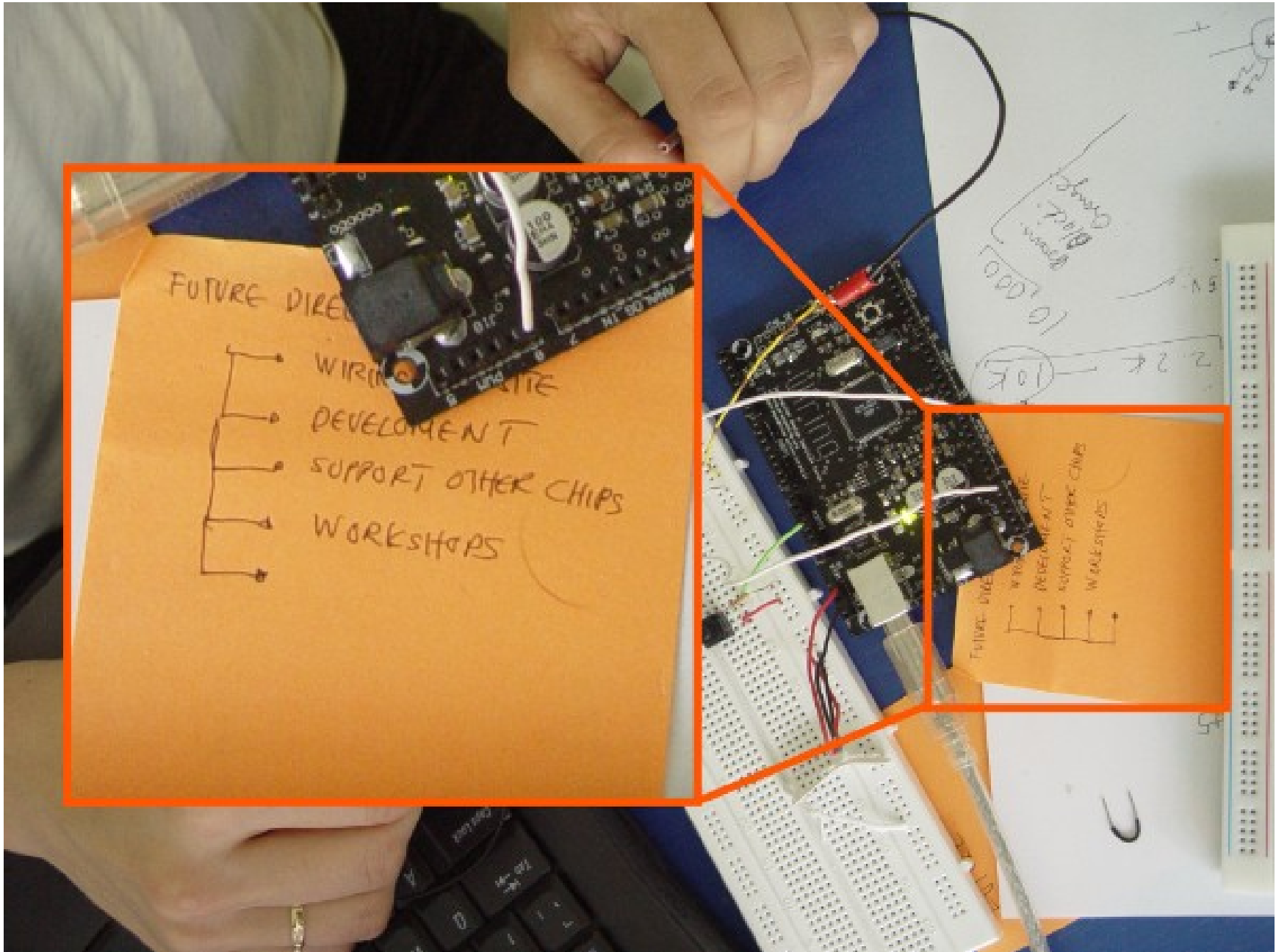
*Primary advisor*

**Casey Reas** . Visiting Assistant Professor

UCLA Design | Media Arts,

*Secondary advisor*

<http://www.mikolaskova.cz/nop/streifzuege>



# Arduino

## Bezug zur Gestaltung

entwickelt von Designern / im Design-Kontext

## Aufeinander abgestimmte Soft-und Hardware:

### Arduino Hardware – Board

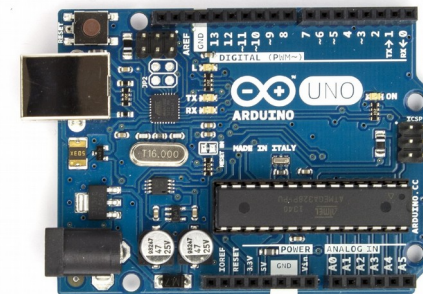
### Software - Arduino IDE

(Entwicklungsumgebung - Integrated Development Environment)

Open Source Software → Software kann frei genutzt werden

Open Hardware → Hardware kann nachgebaut werden

<http://www.mikolaskova.cz/nop/streifzuege>

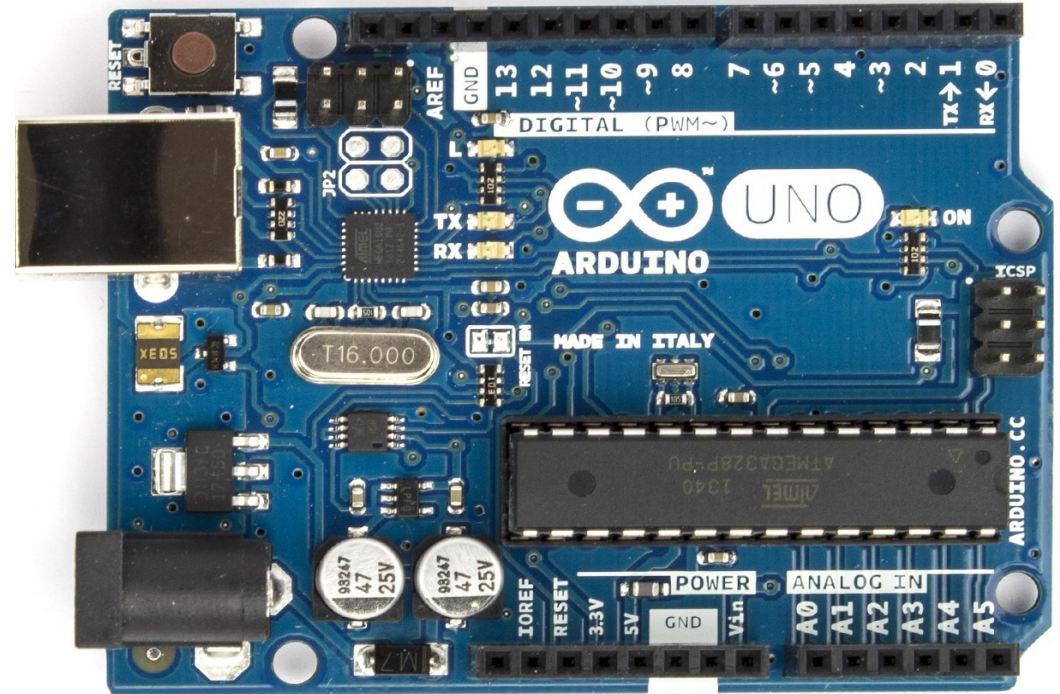
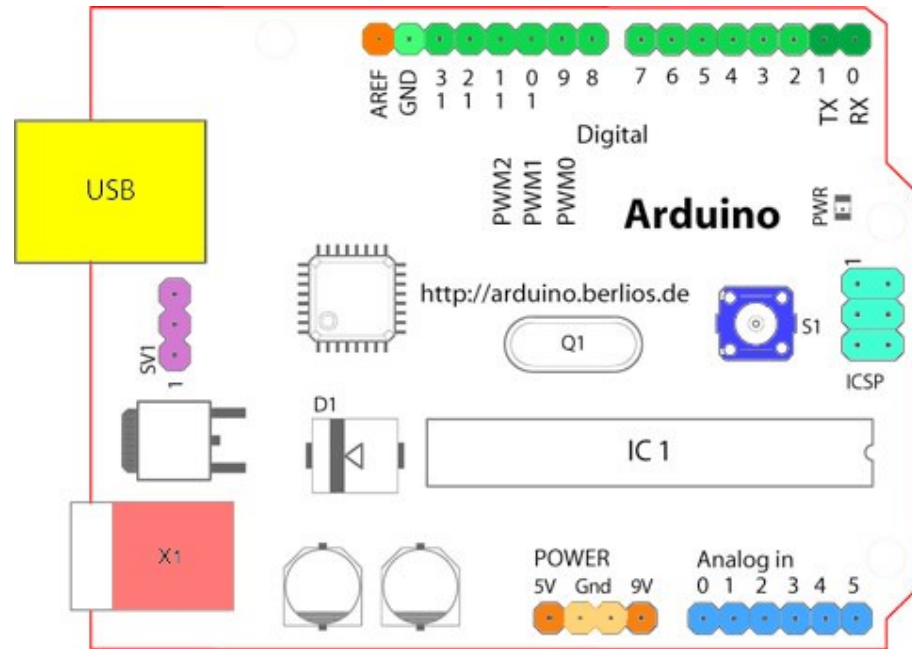


```
File Edit Sketch Tools Help
1
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 * This example code is in the public domain.
 */
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

# Arduino-Board (Hardware)

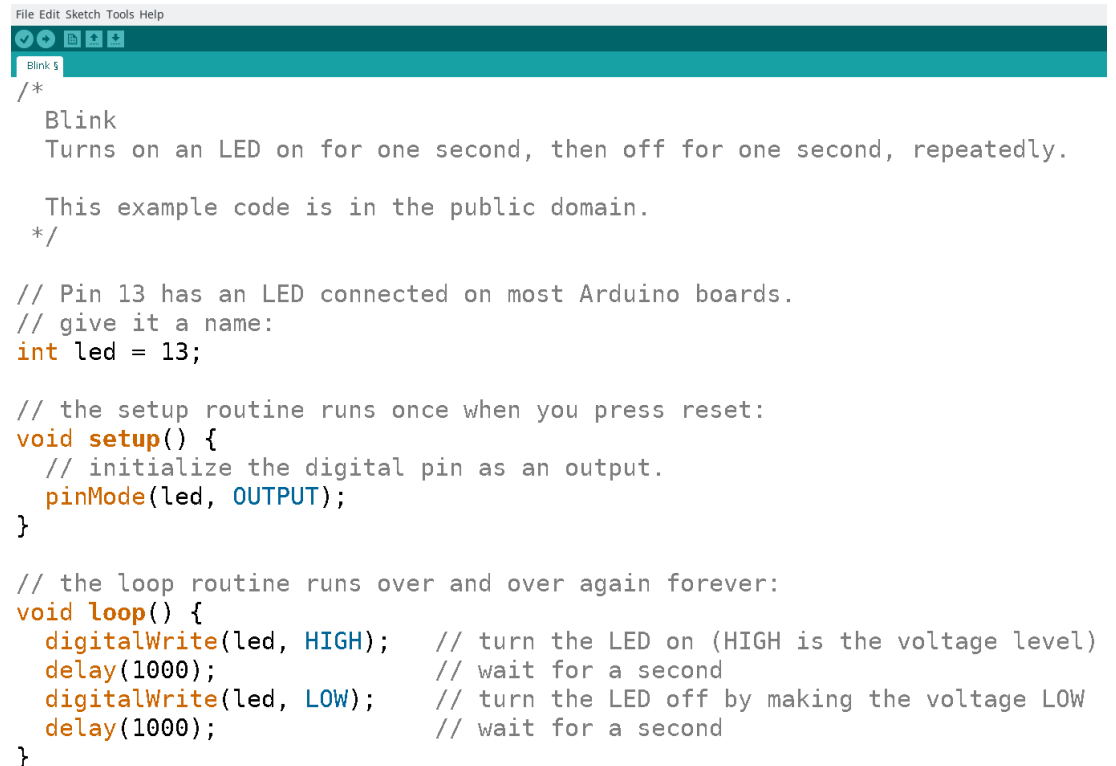


# Programmierung Arduino IDE

Integrierte Entwicklungsumgebung

Integrated Development Environment

- Syntax-Highlighting, Syntax-Prüfung,
- Kompilation
- Übertragen auf Arduino-Hardware
- ...



```
File Edit Sketch Tools Help
Blink 5
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

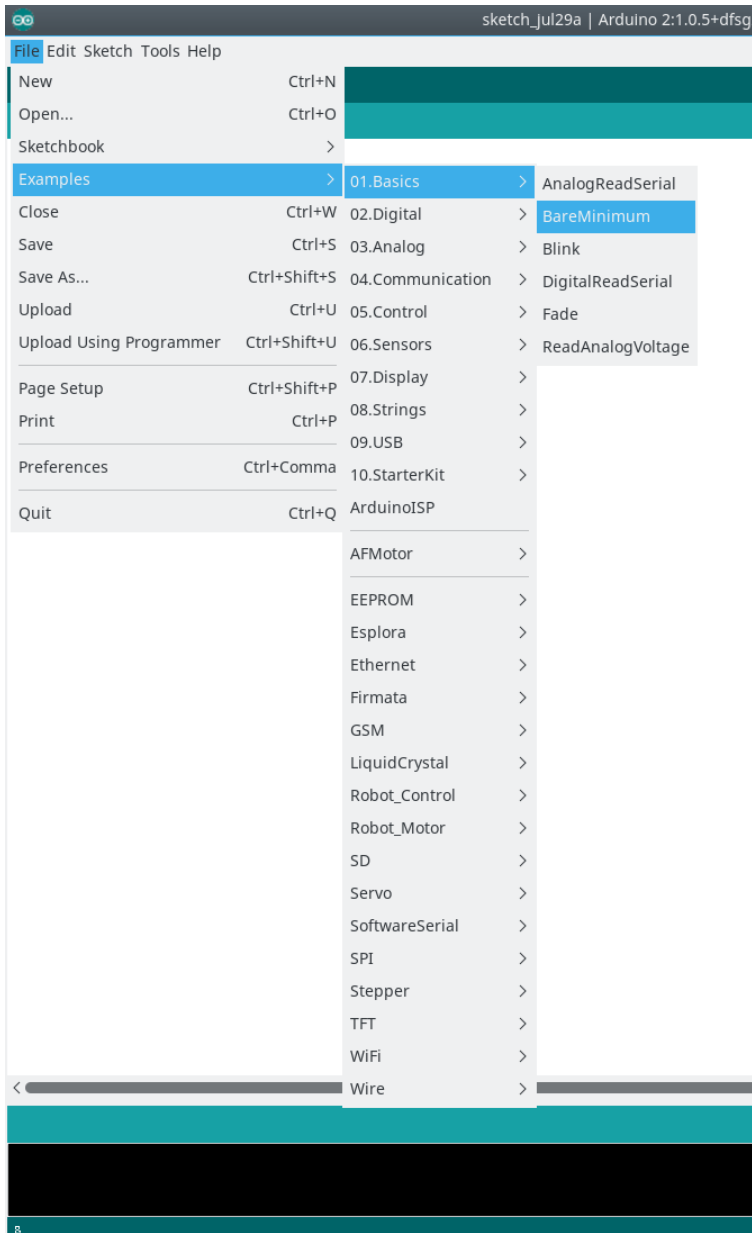
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```



# Programmierung Arduino IDE



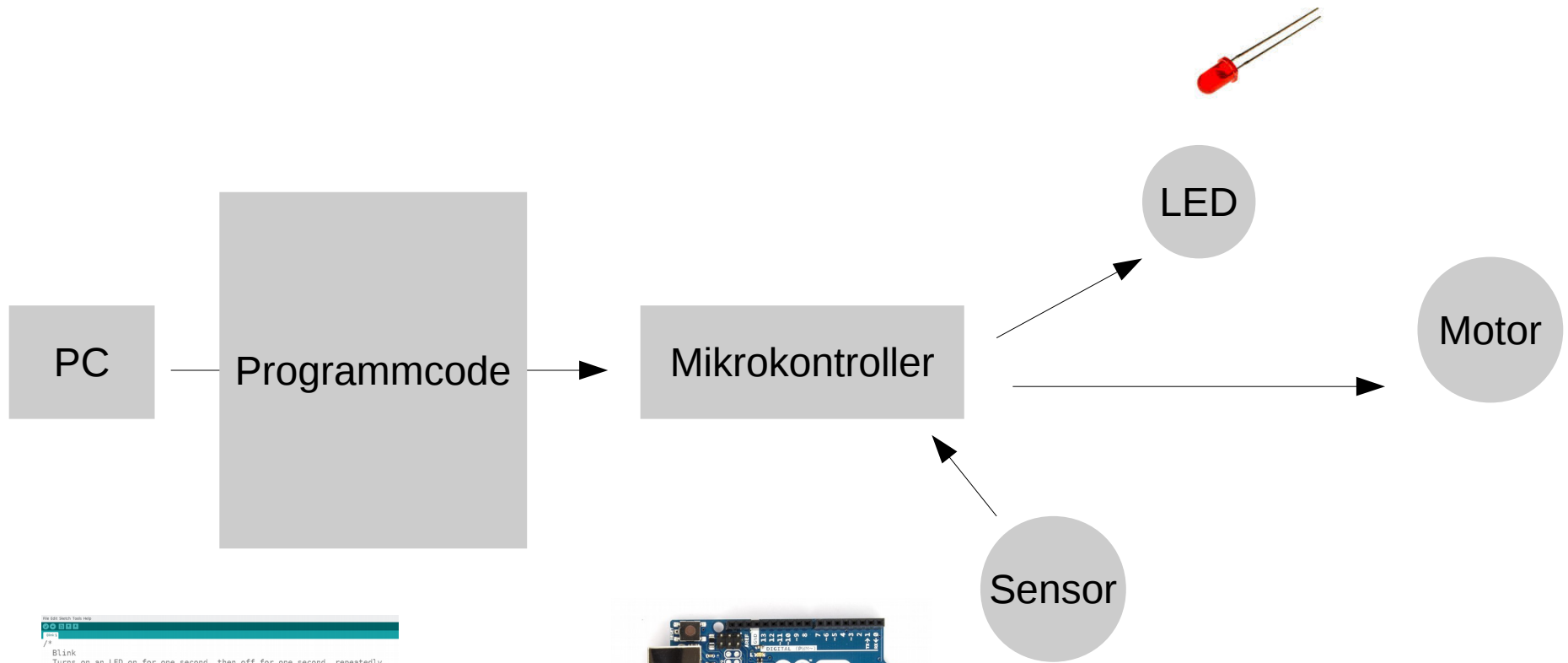
```
File Edit Sketch Tools Help
Blink 5
/*
 *
 * Blink
 * Turns on an LED on for one second, then off for one second,
 *
 * This example code is in the public domain.
 */
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

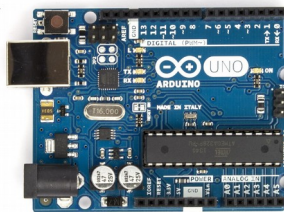
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the v
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the
  delay(1000); // wait for a second
}
```



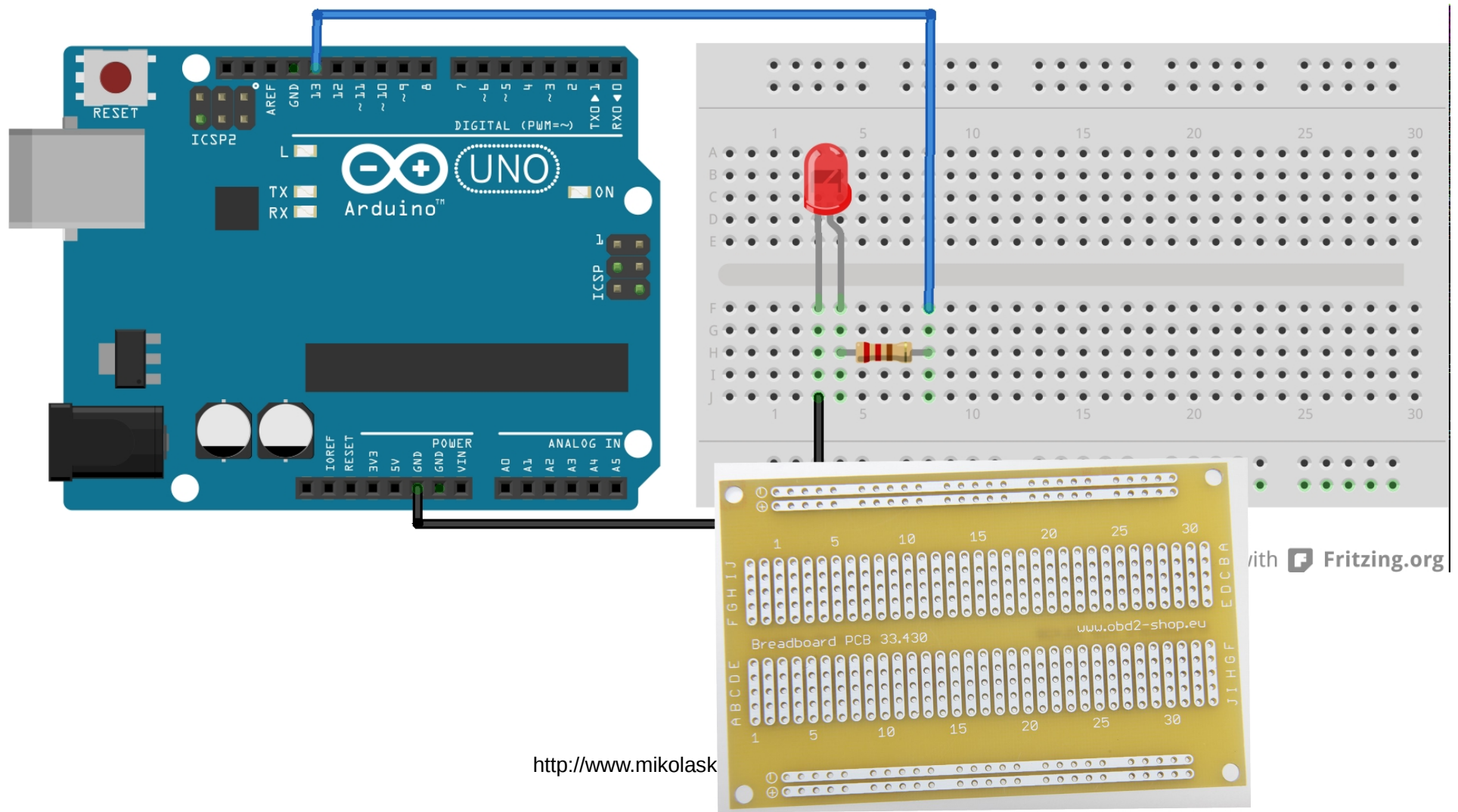
# Physical Computing



```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

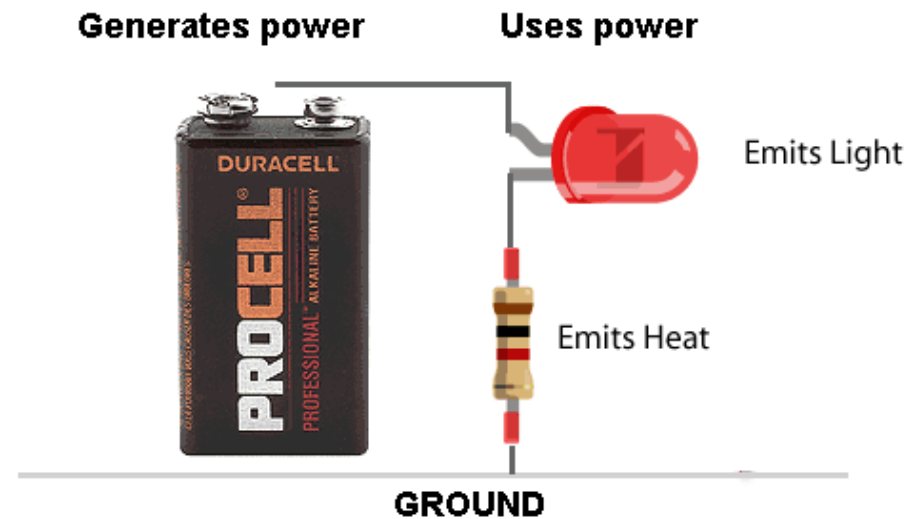
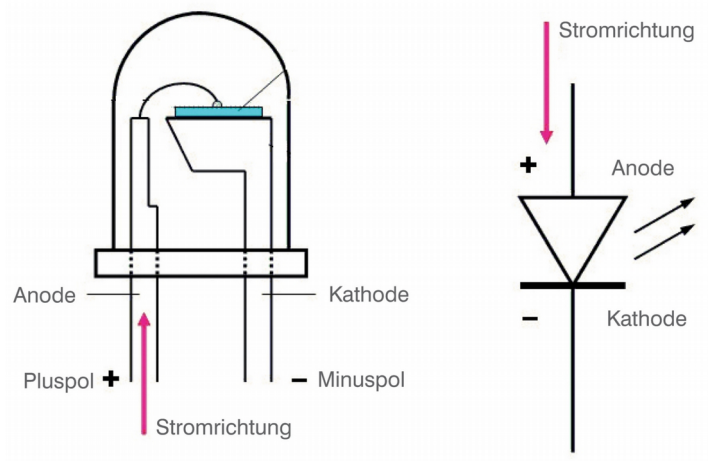


# Weitere allgemeine Komponenten



<http://www.mikolask>

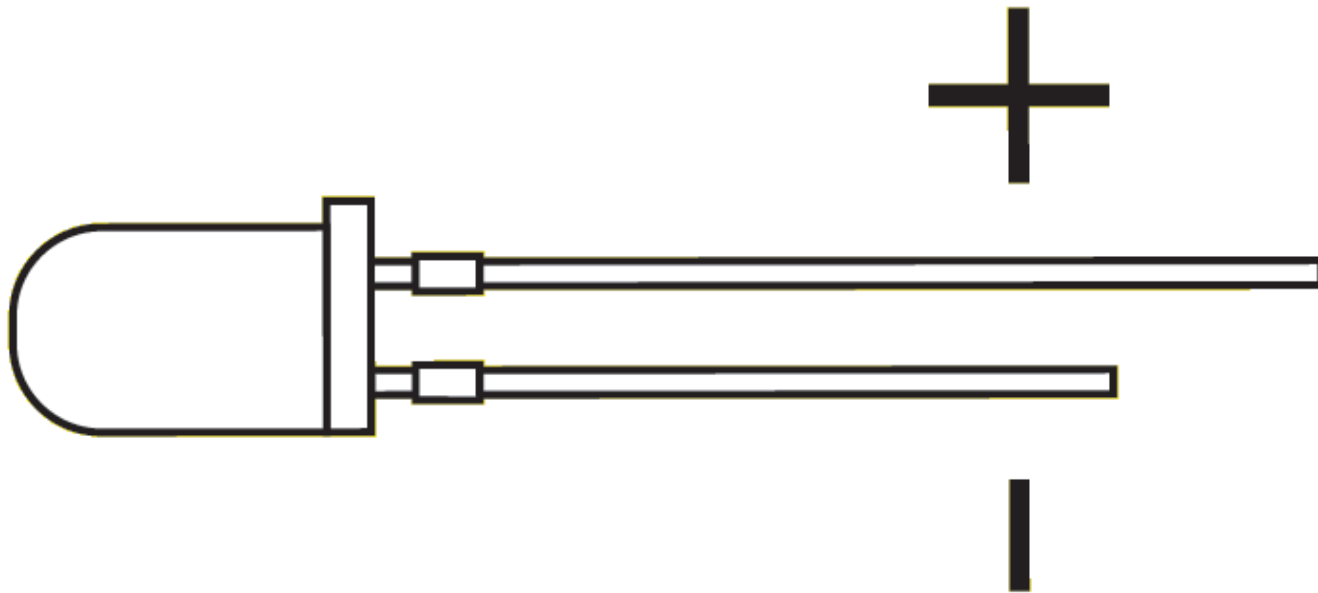
# Weitere Komponenten: LED



<http://www.mikolaskova.cz/nop/streifzuege>

<https://learn.adafruit.com/all-about-leds/overview>

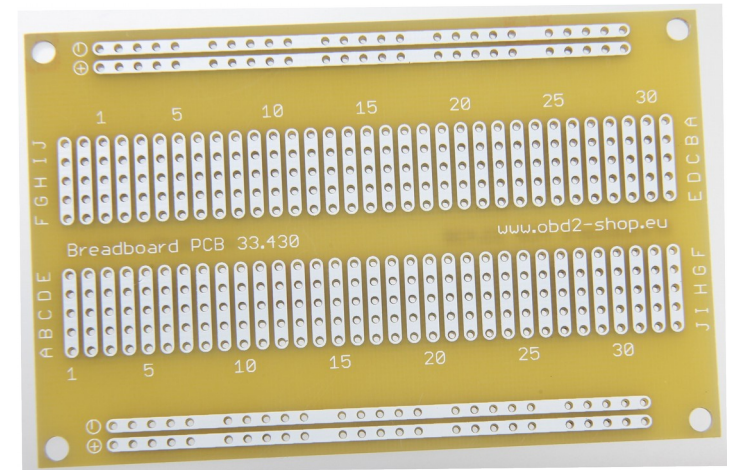
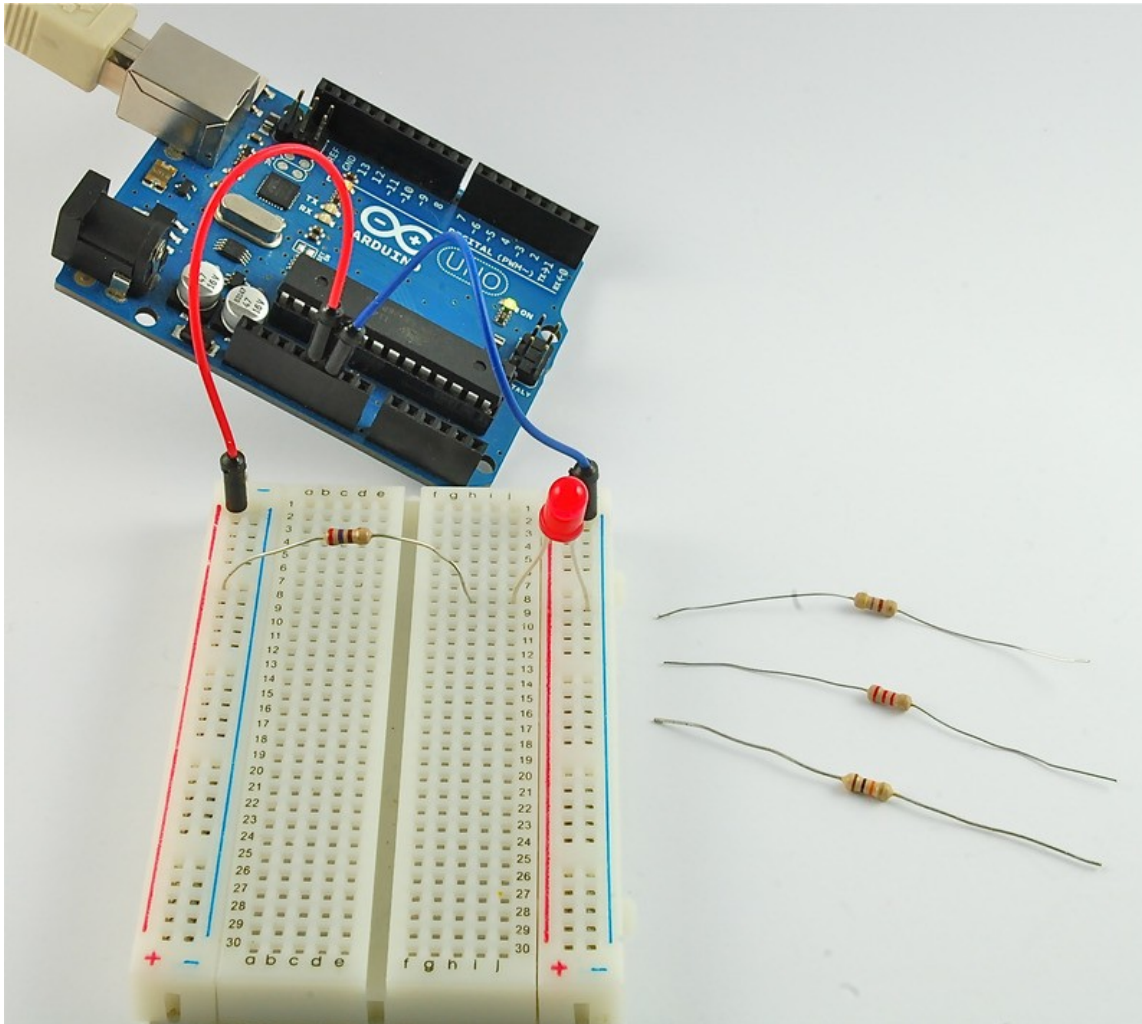
# Weitere Komponenten: LED



<http://www.mikolaskova.cz/nop/streifzuege>

<https://learn.adafruit.com/all-about-leds/overview>

# Steckplatine/Breadboard - Layout



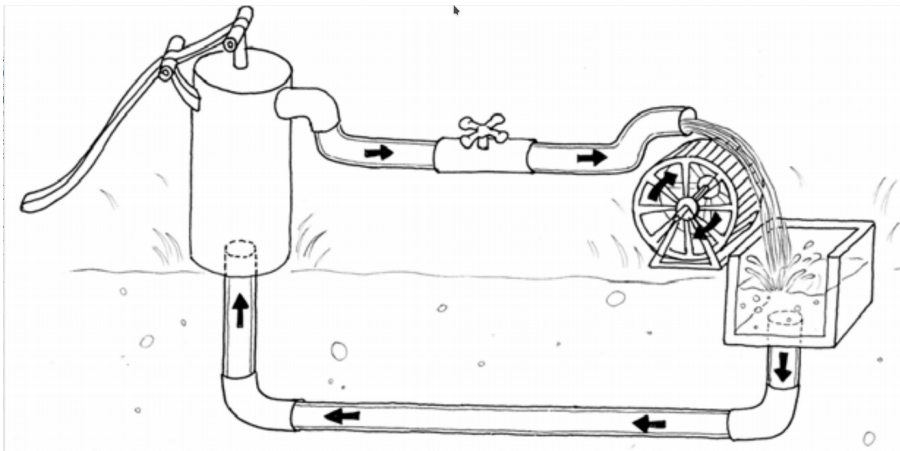
<http://www.mikolaskova.cz/nop/streifzuege>

<https://learn.adafruit.com/breadboards-for-beginners/introduction>

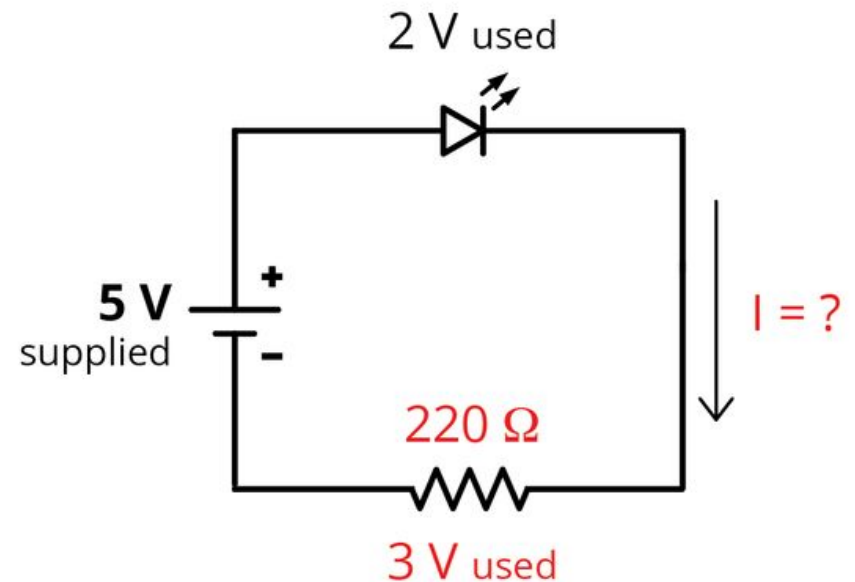
# Weitere Komponenten: Widerstand

270 Ohm

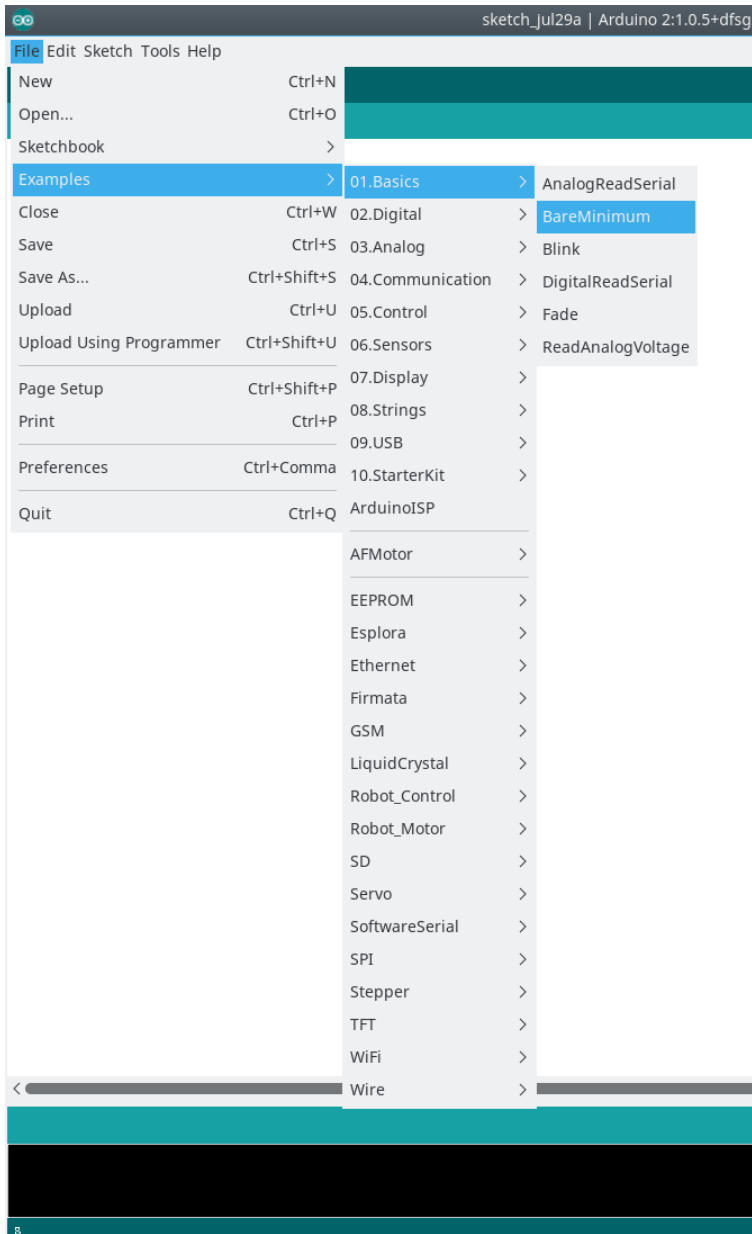
( → Ohmsches Gesetz: Spannung=Widerstand\*Strom )



<http://www.mikolaskova.cz/>



# Blink-Code



```
File Edit Sketch Tools Help
Blink 5
/*
  Blink
  Turns on an LED on for one second, then off for one second...

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making
  delay(1000);             // wait for a second
}
```

→ *Menu Datei → Beispiele → Basics → Blink*



# Blink-Code

```
File Edit Sketch Tools Help
Blink.g
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

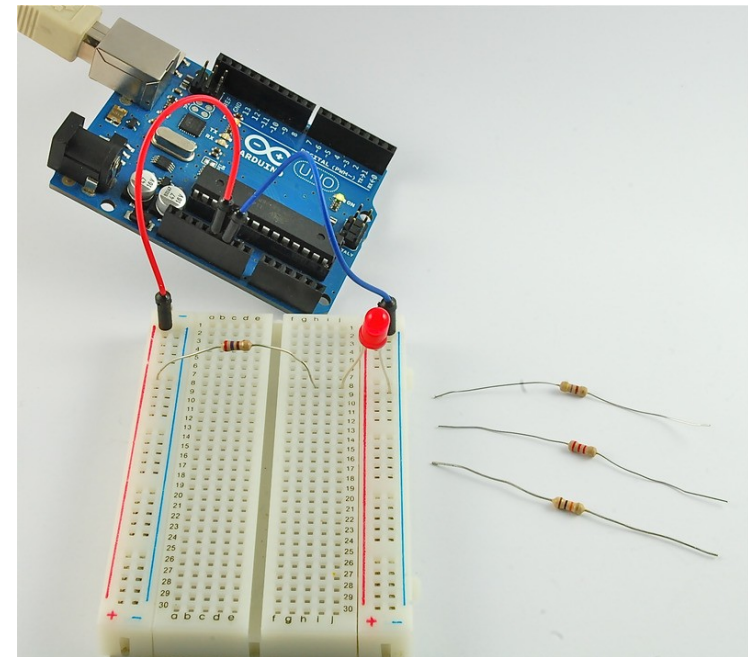
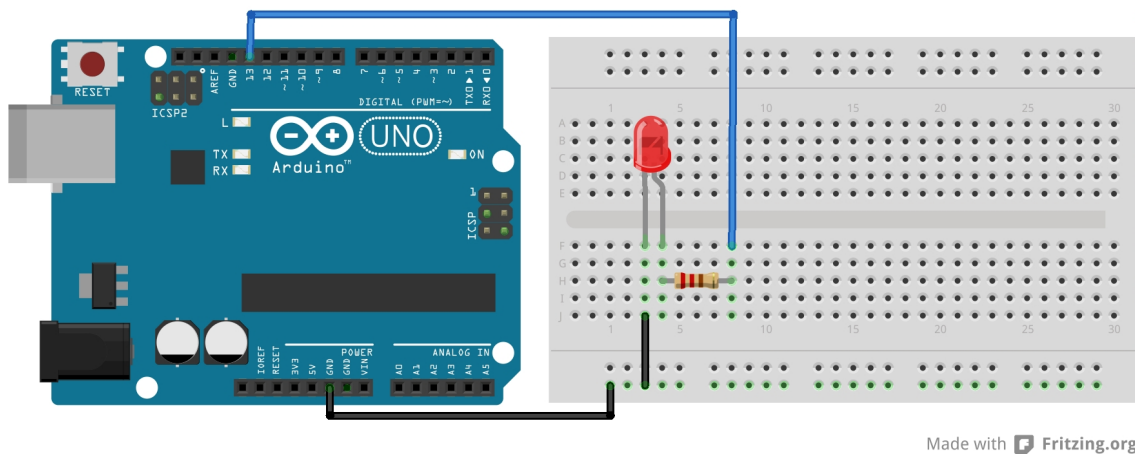
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

→ *Menu Datei* → *Beispiele* → *Basics* → *Blink*



→ selber machen

- Physisch: **Arduino- LED - Widerstand** aufbauen
  - Arduino-IDE: Blinkbeispiel öffnen und auf Arduino laden
- *Menu Datei* → *Beispiele* → *Basics* → *Blink*



# Licht an- und ausschalten

DigitalWrite( ) - HIGH oder LOW

```
digitalWrite(led1,HIGH)
```

```
Delay(1000);
```

```
digitalWrite(led1,LOW)
```

```
delay(1000);
```

# Lichtmenge steuern

**analogWrite( )**

**Werte zwischen 0 und 255**

**Nur auf Pins 3, 5, 6, 9, 10, und 11 möglich!**

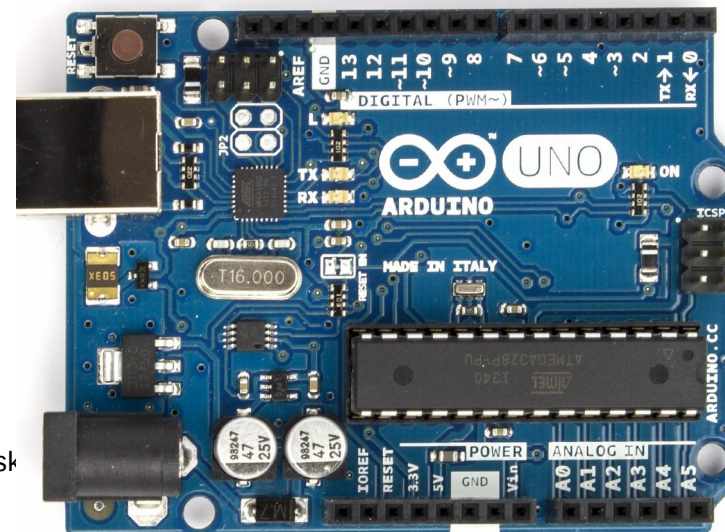
```
analogWrite(led, 255);
```

```
delay(1000);
```

```
analogWrite(led, 0);
```

```
delay(1000);
```

<http://www.mikolask>



# → selber machen

- Pin ändern für AnalogWrite()

- analogWrite(led, 255)
- analogWrite(led,0)

```
void loop() {  
  
    analogWrite(led, 255);  
    delay(10000);  
    analogWrite(led, 0);  
    delay(1000);  
  
}
```

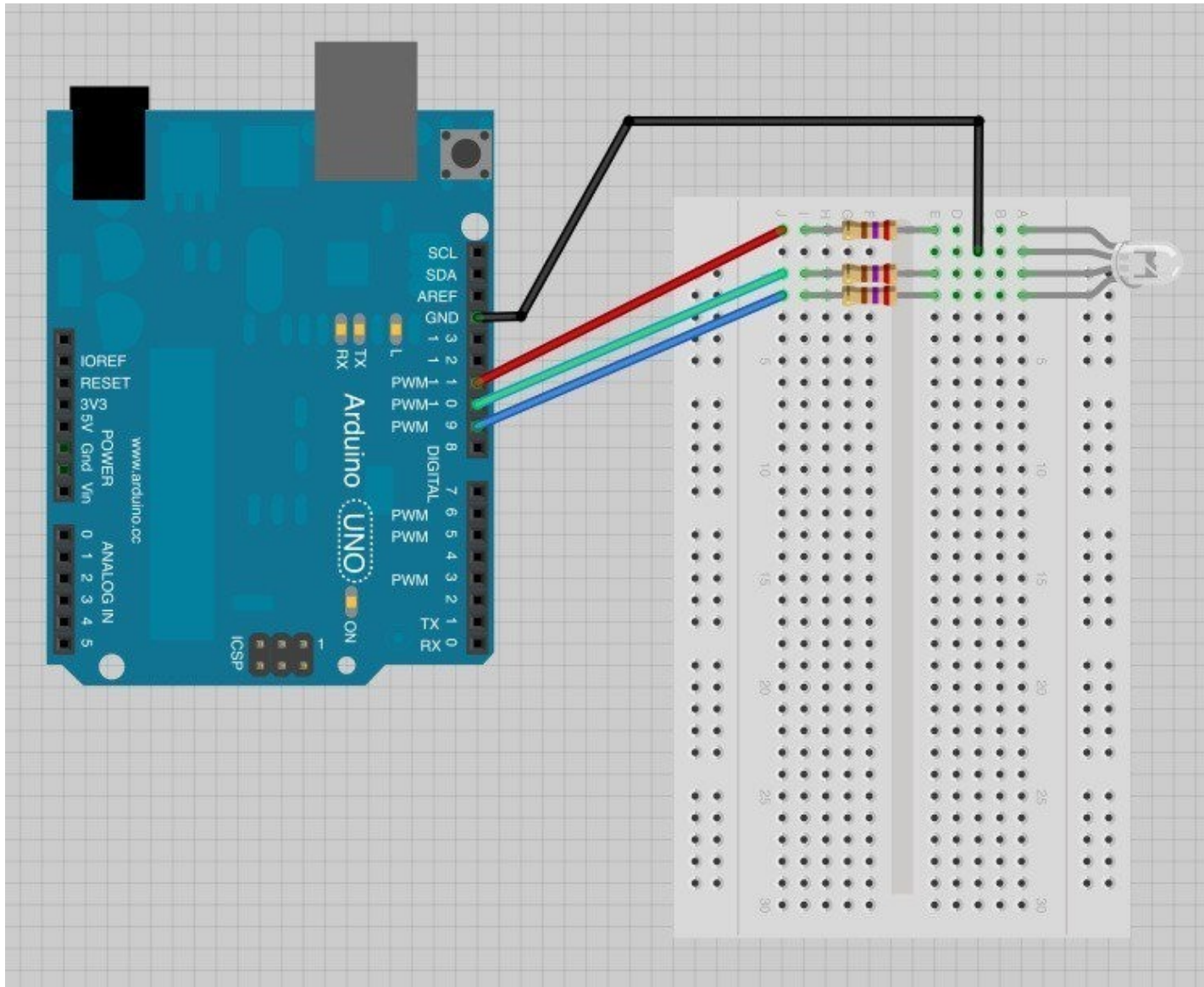
# RGB-LEDS

Drei Lampen in einer LED

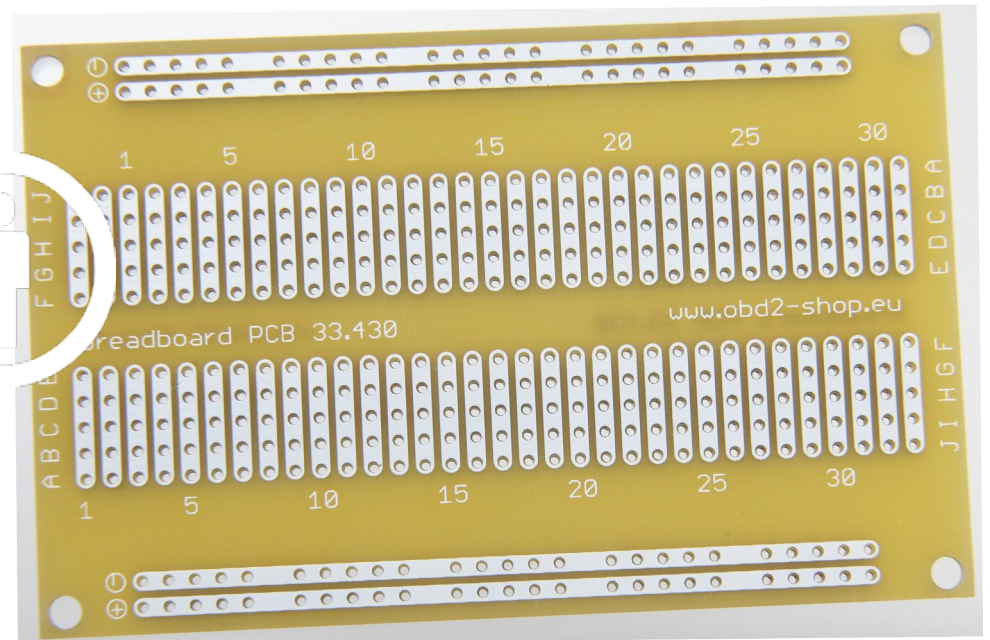
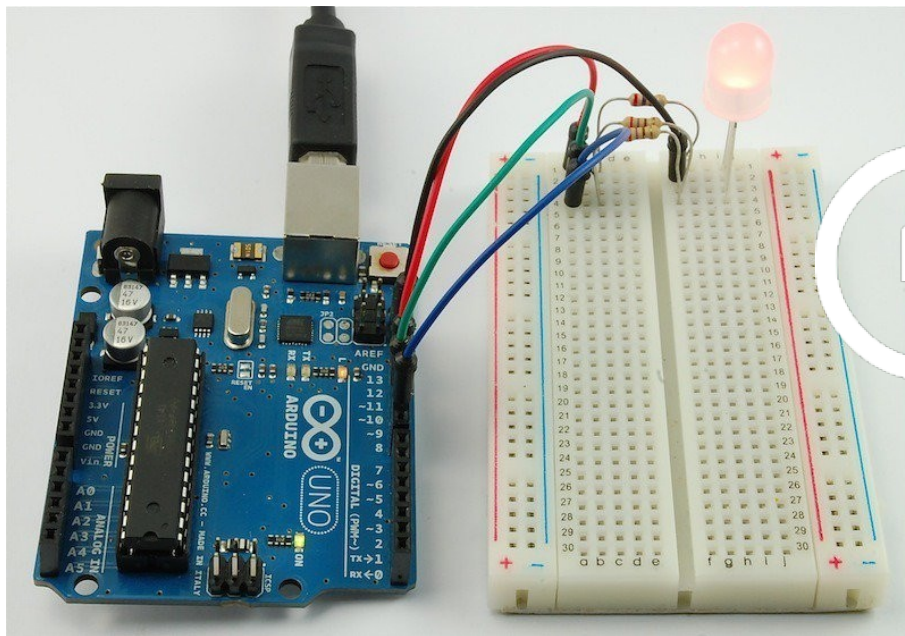
Common Cathode: langes Bein → -

Common Anode: langes Bein → +









<http://www.mikolaskova.cz/nop/streifzuege>

<https://learn.adafruit.com/breadboards-for-beginners/introduction>

```
sketch_oct03b | Arduino 2:1.0.5+dfsg2-4.1
File Edit Sketch Tools Help
sketch_oct03b
int redPin = 11;
int greenPin = 10;
int bluePin = 9;

//uncomment this line if using a Common Anode LED
//#define COMMON_ANODE

void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop()
{
  setColor(255, 0, 0); // red
  delay(1000);
  setColor(0, 255, 0); // green
  delay(1000);
  setColor(0, 0, 255); // blue
  delay(1000);
  setColor(255, 255, 0); // yellow
  delay(1000);
  setColor(80, 0, 80); // purple
  delay(1000);
  setColor(0, 255, 255); // aqua
  delay(1000);
}

void setColor(int red, int green, int blue)
{
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}

Done compiling.
Binary sketch size: 1,432 bytes (of a 32,256 byte maximum)
```

# Code für RGB-LED

Statt jeden Pin einzeln zu setzen → auslagern

eigene Funktion **setColor(...)**

```
setColor(255, 0, 0);
```

...

```
void setColor(int red, int  
green, int blue)
```

```
{  
  analogWrite(redPin, red);  
  analogWrite(greenPin, green);  
  analogWrite(bluePin, blue);  
}
```

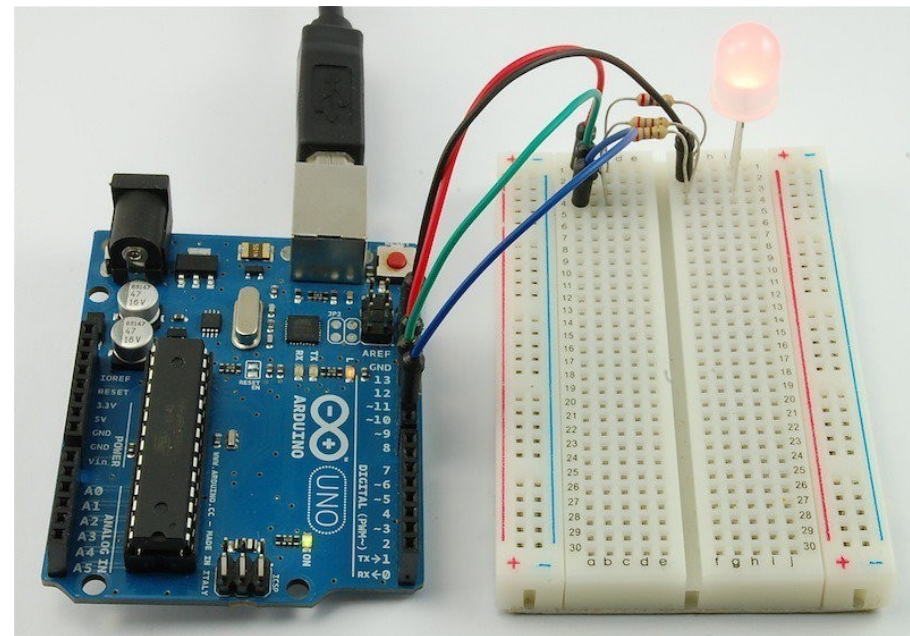
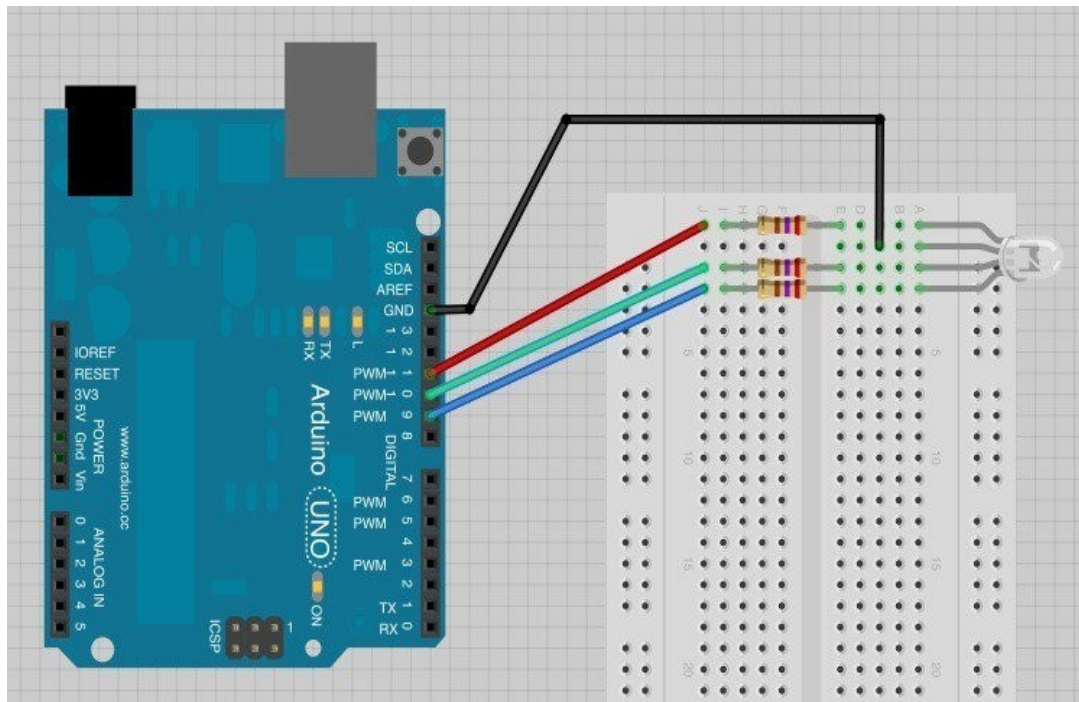


→ selber machen

**Physisch- Hardware:** Arduino- LED-Widerstände aufbauen (Alternative: RGB-LED-Modul)

**Software:** Arduino-IDE: RGB-Blink Beispiel öffnen → <http://www.mikolaskova.cz/nop/streifzuege>

Prüfen ob Code mit Farben übereinstimmt → allenfalls Pins anpassen



<http://www.mikolaskova.cz/nop/streifzuege>

# Programmierung

# Kontrollstrukturen

z.B. for-Schleife: **4 mal wiederholen:**

So lange  $i$  kleiner als 4 ( $i < 4$ ) ist, wird das Codestück wiederholt und  $i$  am Ende um 1 erhöht ( $i=i+1$ ).

```
for ( int i= 1;  i < 4;  i=i+1 ) {  
    setColor(255,0,0);  
    delay(250);  
    setColor(0,0,0);  
    delay(250);  
}
```

# Zufallsfunktion

Bereich abstecken. In welchem Bereich liegt der Zufall?

z.B. Zufall im **Timing**

```
int dauer=random(500,1000);
```

Eine zufällige Zahl zwischen 500 und 1000 wird der Variablen “dauer” zugewiesen und kann so der delay-Funktion übergeben werde

```
delay(dauer);
```

# Zufallsfunktion – random( )

In welchem Bereich soll der Wert liegen?

z.B. zufälliger **Farbwert**

```
int gruenwert=random(0,255);
```

Eine zufällige Zahl zwischen 0 und 255 wird der Variablen “grün” zugewiesen und kann so der setColor-Funktion übergeben werden

```
setColor(255,gruenwert,0);
```