

VON ARDUINO/RASPBERRY PI UND LEUCHTENDEN KOSTÜMEN



Paola Maneggia paola.maneggia@ifi.lmu.de

Manuel Huber

TdI @LMU - 6. Juli 2018

ZIEL DES PROJEKTS

- Leuchtende Kostüme für eine Tanzchoreographie
- Feste Zeit: Aufführung am 4.-6.7.2018
- Keine weiteren Vorgaben außer:
- Die Kostüme sind nicht nur leuchtend, sondern Computer-gesteuert: Es ist ja ein P-Seminar Informatik!

STRUKTUR

- Am Anfang: Einführung in Arduino für alle
- Prototyping
- Meilensteine und agile Vorgehensweise

Menschen und Maschinen



Choreographie Gül Altner
Tanzprojekt des Oskar-Maria-Graf-Gymnasium, Neufahrn bei Freising
Musik: Game of Survival, Ruelle

Tron Dance



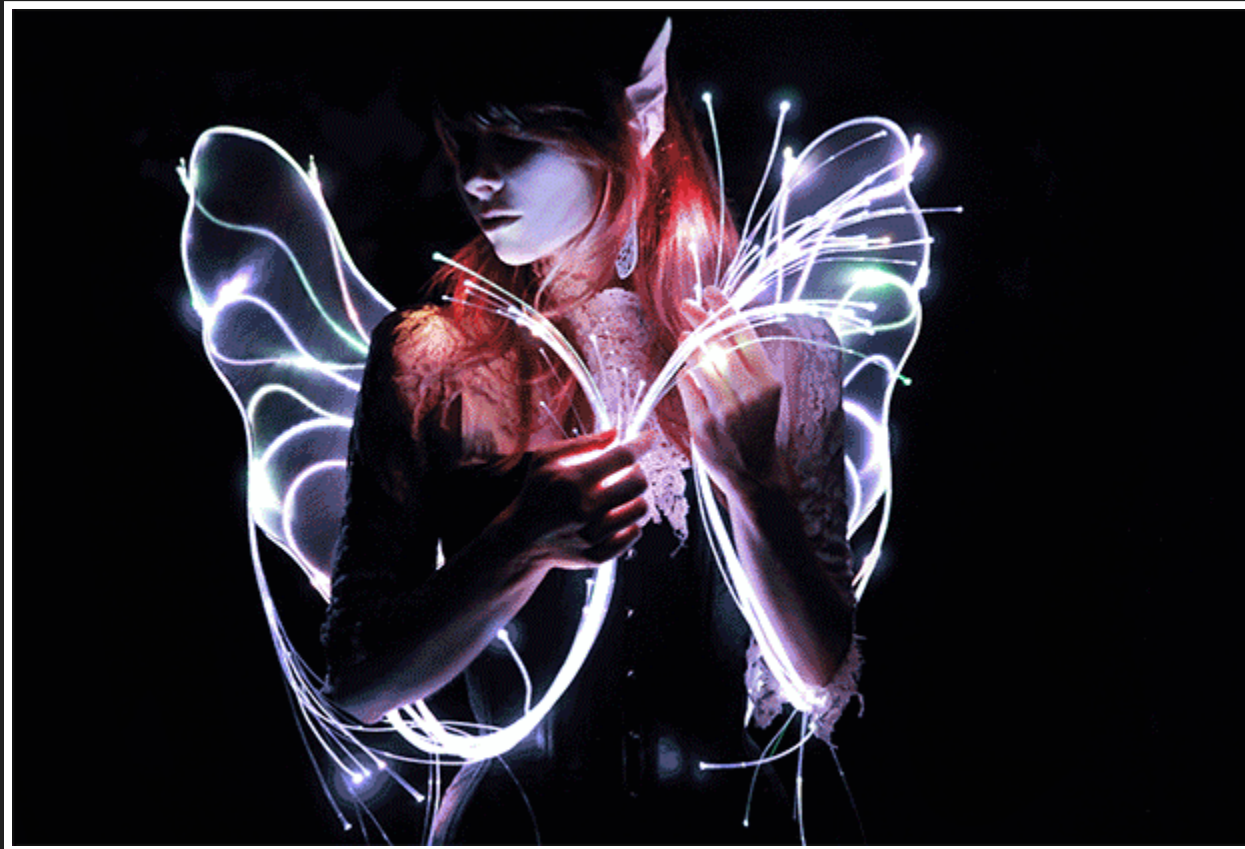
performed by Wrecking Orchestra
<https://www.youtube.com/watch?v=-Rot9uaVO8s>

Brooklyn Ballet, Nussknacker



https://www.youtube.com/watch?v=VAgtELc-_NI

Fairy Wings Tutorial, Instructables



<http://www.instructables.com/id/Fiber-Optic-Fairy-Wings/>

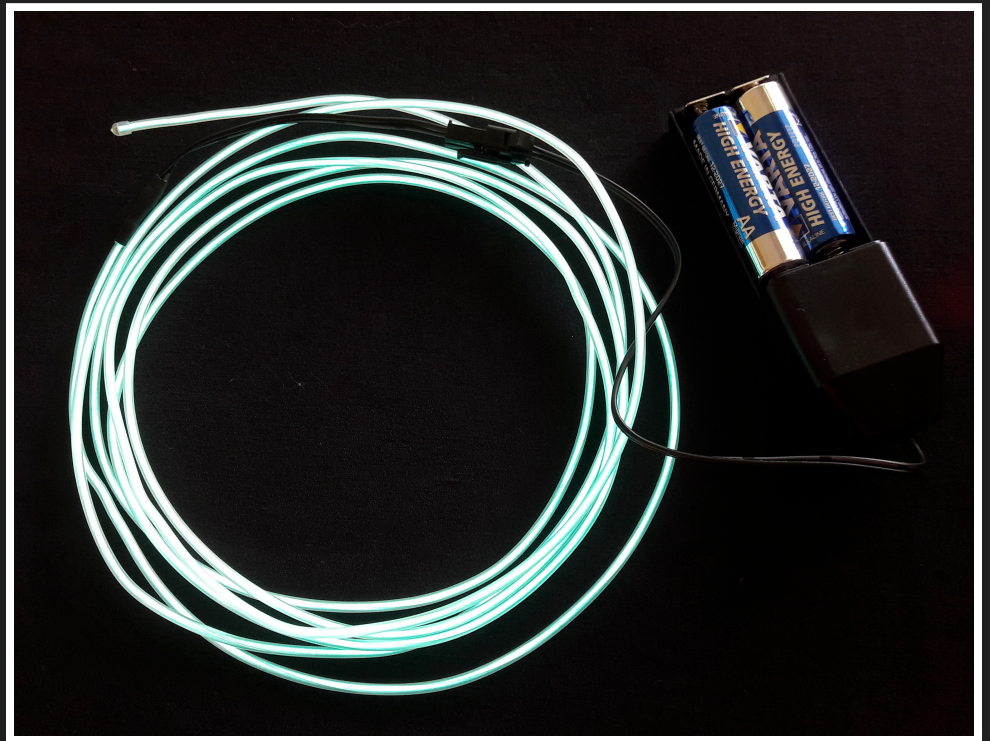
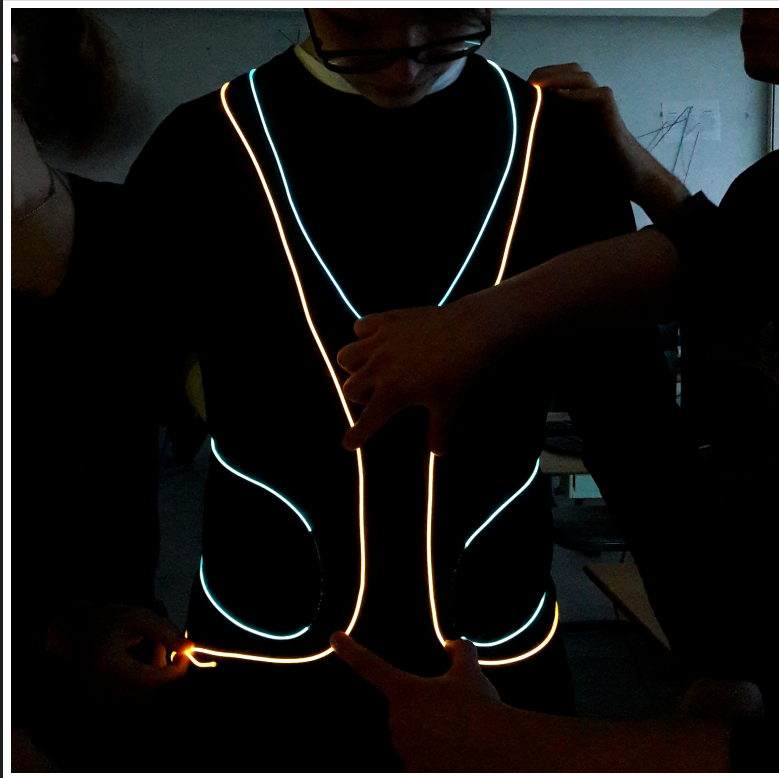
SCHWERPUNKTE

1. Leuchtmittel → Stromversorgung!
2. Steuerung (→ Stromversorgung)

LEUCHTMITTEL

1. EL Wire
2. LED-Streifen
3. Einzelne LEDs (eventuell mit Glasfasern)
4. NeoPixel (WS2812, d.h. WS2811 in 5050 LED package)

EL WIRE: ELECTROLUMINESCENT WIRE



EL WIRE

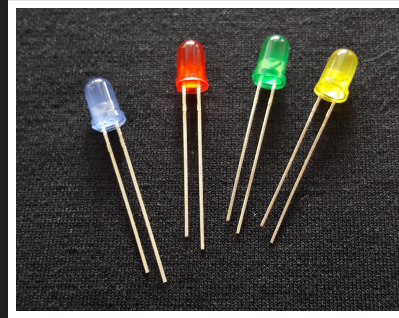
- Stromversorgung: 12V Wechselstrom
- Steuerung komplizierter
- SparkFun EL Sequencer (ca 30€)...
- ...ist ein Arduino-kompatibler Controller
- Problem: Hohe Spannung! (bis 120 V)



"KLASSISCHE" LED-STREIFEN

- Stromversorgung: 12V Gleichstrom
- Nicht individuell ansteuerbar

EINZELNE LEDS



- verwenden in der Regel 20mA
- werden nicht warm
- brauchen wenig Spannung (um 3V, je nach Farbe/Typ)
- Helligkeit in mcd (millicandels) 20mcd sehr schwach, 3000mcd hell
- Diffuse oder punktuelle Strahlung, Abstrahlwinkel 15° - 180°
- RGB haben intern drei kleine LEDs und eine gemeinsame Anode(+)

EINZELNE LEDS (CONTINUED)

- Strom fließt nur in eine Richtung
langes Beinchen: Anode (+)
- Sie brauchen die richtige Spannung, lieber weniger als mehr, sonst verbrennen sie
- Mit einem Widerstand schützen:
Zum Beispiel: LED funktioniert mit 2,2V und 20mA, wird mit 5V Batterie angeschlossen → braucht einen Widerstand von $\frac{5V-2.2V}{20mA} = 140\Omega$ (Ohmsches Gesetz)
- Sie können kaum über die Spannung gedimmt werden:
Mit zu wenig Spannung gehen sie gar nicht mehr an.
- Man verwendet stattdessen PWM (*pulse-width modulation*, Pulsweitenmodulation)

PWM - PULSE-WIDTH MODULATION

Das Problem: Die LED arbeitet mit einer vorgegebenen Spannung, aber wir möchten sie dimmen können

Die Lösung: Die Spannung bleibt konstant, aber wir schalten die LED sehr schnell immer wieder an und aus

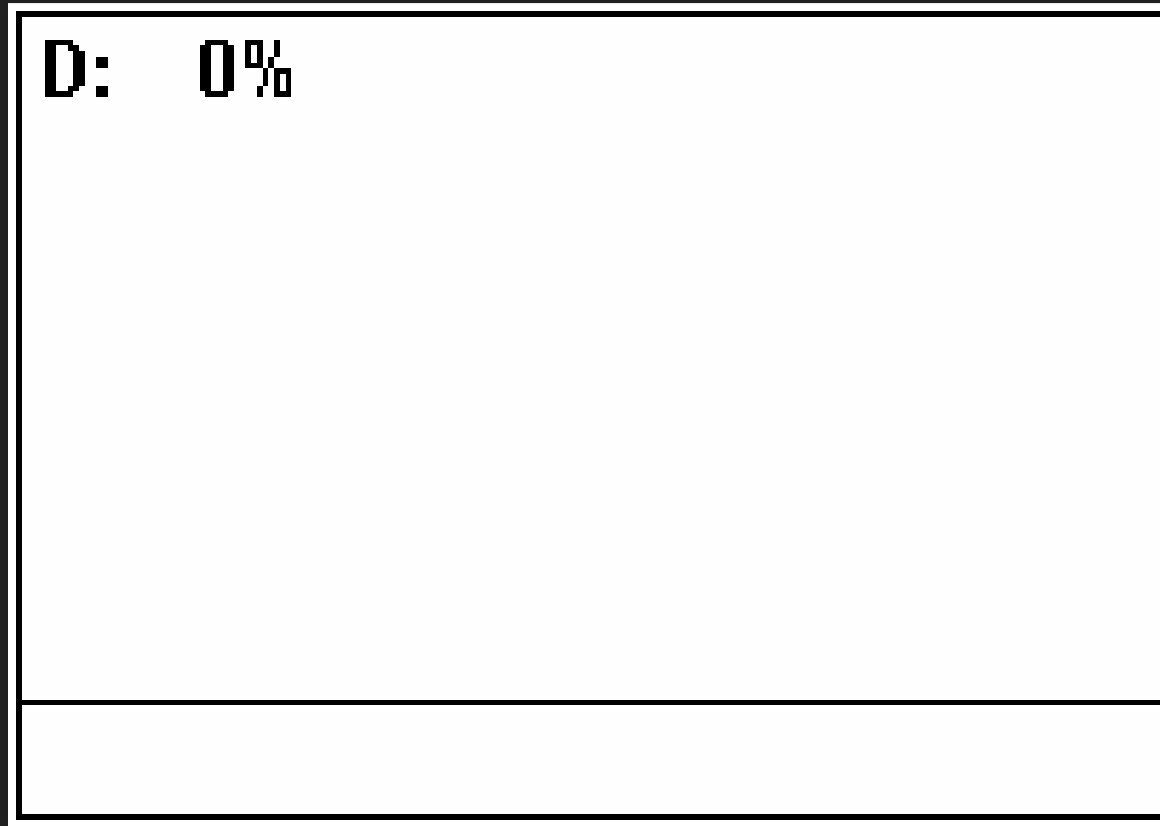
Das Auge nimmt nur weniger Licht und keine Oszillation wahr
(persistence of vision)

Graphisch kann man das Verhalten eines PWM-Pins mit einer quadratischen Welle darstellen.

Es gibt zwei Größen einer solchen quadratischen Welle:

1. der *duty cycle* (in Prozent - Deutsch: Aussteuergrad) und
2. die Frequenz (in Hz)

DER DUTY CYCLE...



Quelle: https://en.wikipedia.org/wiki/Duty_cycle

...ist das Verhältnis zwischen der Zeit, in der das Signal HIGH ist, im Vergleich zur Periode der quadratischen Welle.

DIE FREQUENZ...

...ist, wie viele Perioden der Welle in eine Sekunde passen.

Welche Frequenz man braucht, hängt von der Anwendung ab. Für das Dimmen von LEDs sind 500Hz mehr als ausreichend.

SPWM UND HARDWARE PWM

- Manche Pins haben eine PWM-Funktion in Hardware (mehr dazu in den Arduino-Abschnitt)
- PWM kann via Software *manuell* umgesetzt werden:
Widerhole 1000 mal: Licht an, warte 70ms, Licht aus, warte 30ms
- Für den Raspberry Pi gibt es **Bibliotheken**, die PWM via Software erzeugen (SPWM).

WS2812 (NEOPIXEL)

- Stromversorgung: 5V Gleichstrom
- Individuell ansteuerbar!
- Sind im Streifen parallel geschaltet; jede LED kann bis 60mA Strom ziehen
- Faustformel: 20mA per LED
- 100 LEDs brauchen 2 bis 6A
- Mit neoPixel-Library gesteuert
- Teuer

NEOPIXEL - TUTORIALS

- <https://learn.adafruit.com/adafruit-neopixel-uberguide/the-magic-of-neopixels>
- <https://www.tweaking4all.com/hardware/arduino/arduino-ws2812-led/>
- <https://www.tweaking4all.com/hardware/arduino/adruino-led-strip-effects/>

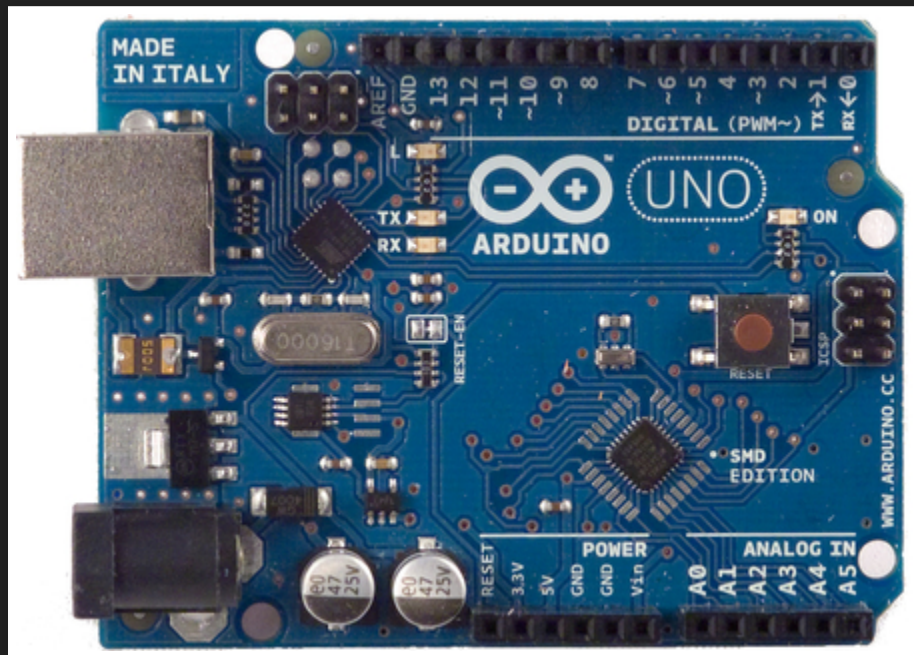


STEUERUNG

- Arduino
- Raspberry Pi

ARDUINO

Ein Mikrokontroller und eine IDE zur Programmierung



```
sketch_may25a $
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

11 Arduino/Genuino Uno auf /dev/cu.usbmodem1411

Die Sprache basiert auf C++

DOWNLOADS

<https://arduino.cc>

<https://create.arduino.cc> Online-Tool

<http://fritzing.org> Bauplan-Editor

ARDUINO FÜR...

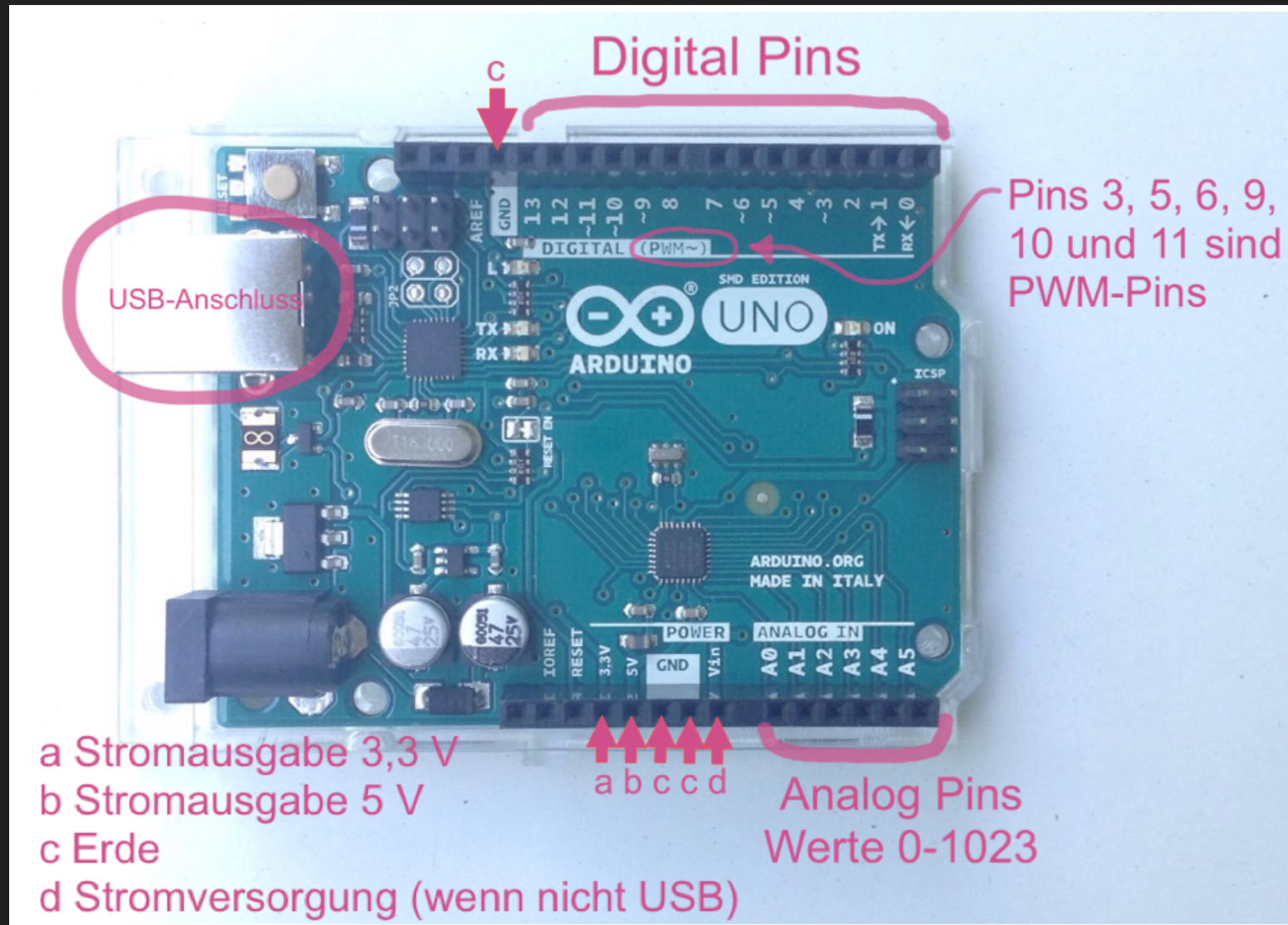
- Prototyping
- Basteln
- Kunst

MASSIMO BANZI - TED 2012

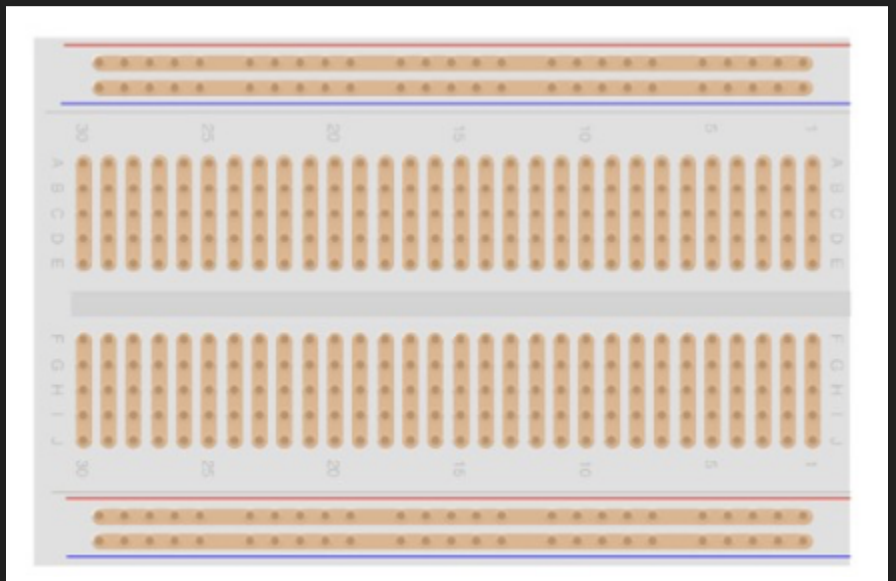
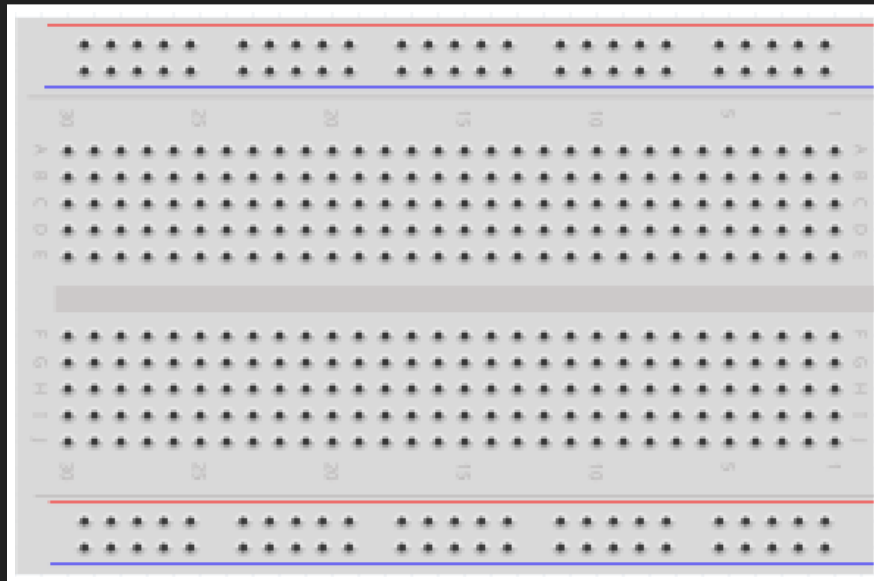
https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination



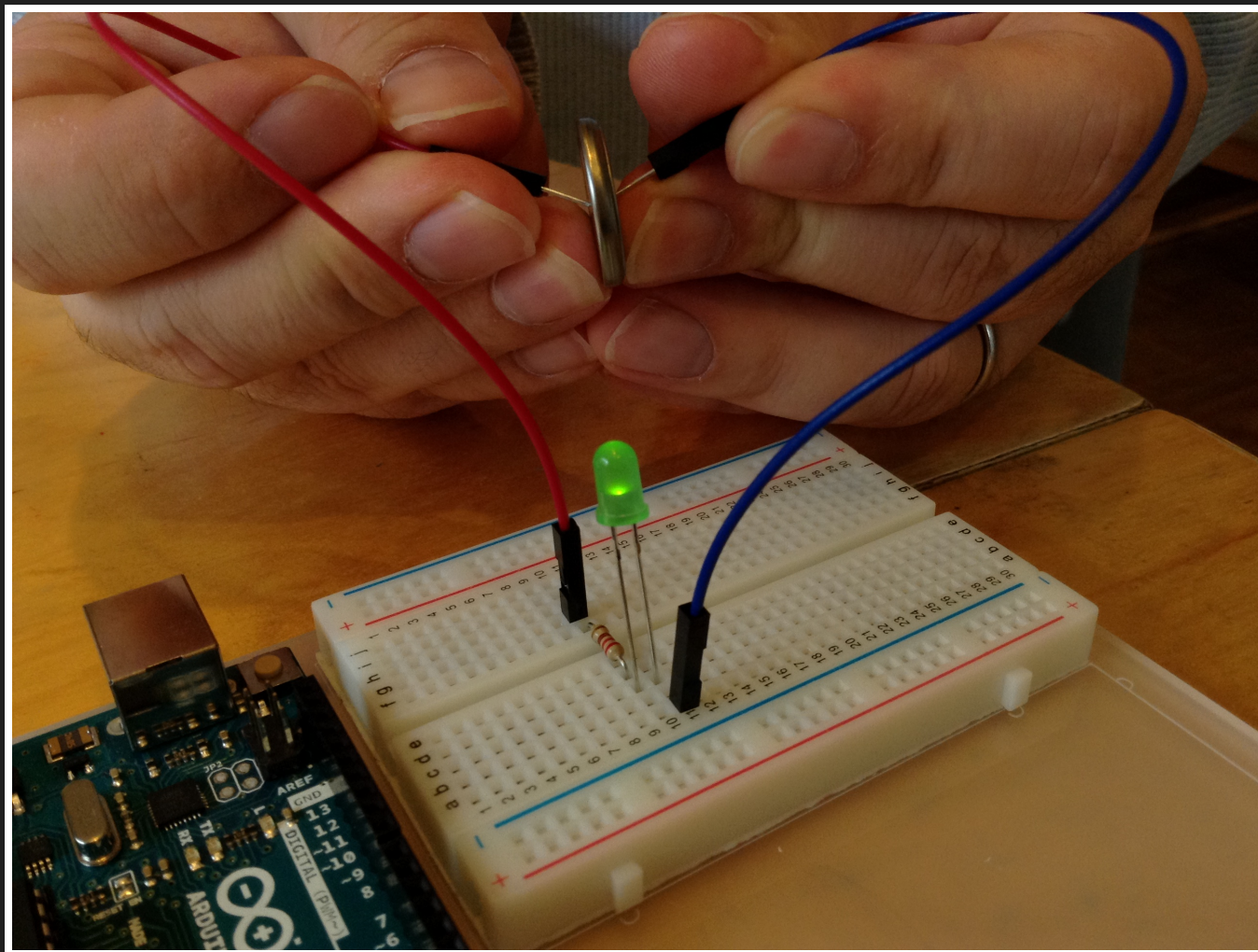
ARDUINO - DIE HARDWARE



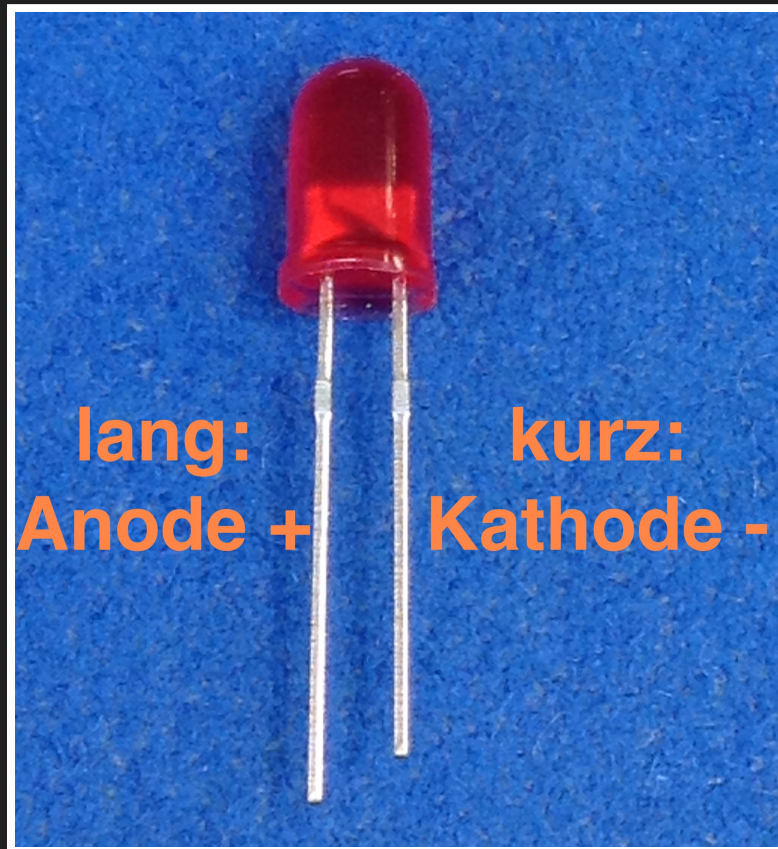
STECKBRETT



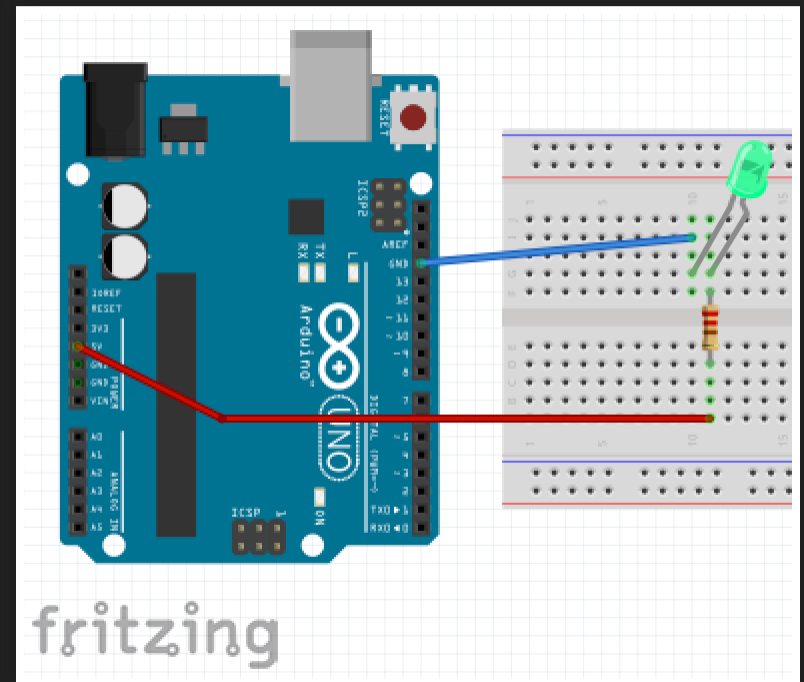
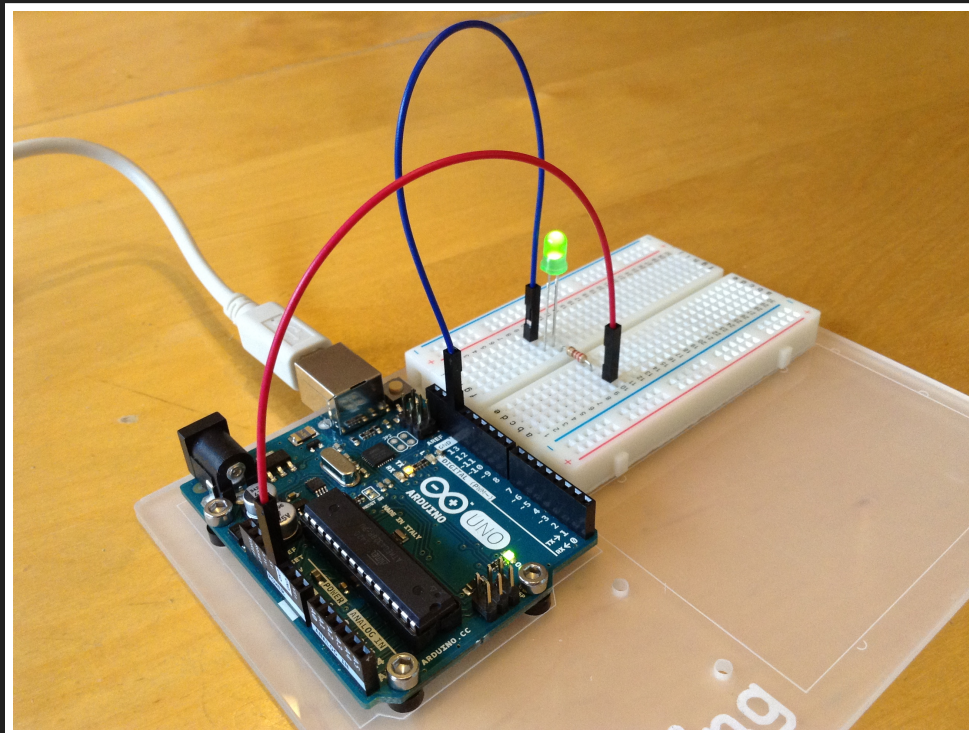
SCHALTKREIS MIT BATTERIE



EIN KLEINER SCHRITT ZURÜCK... LED UND WIDERSTAND



ARDUINO ALS BATTERIE



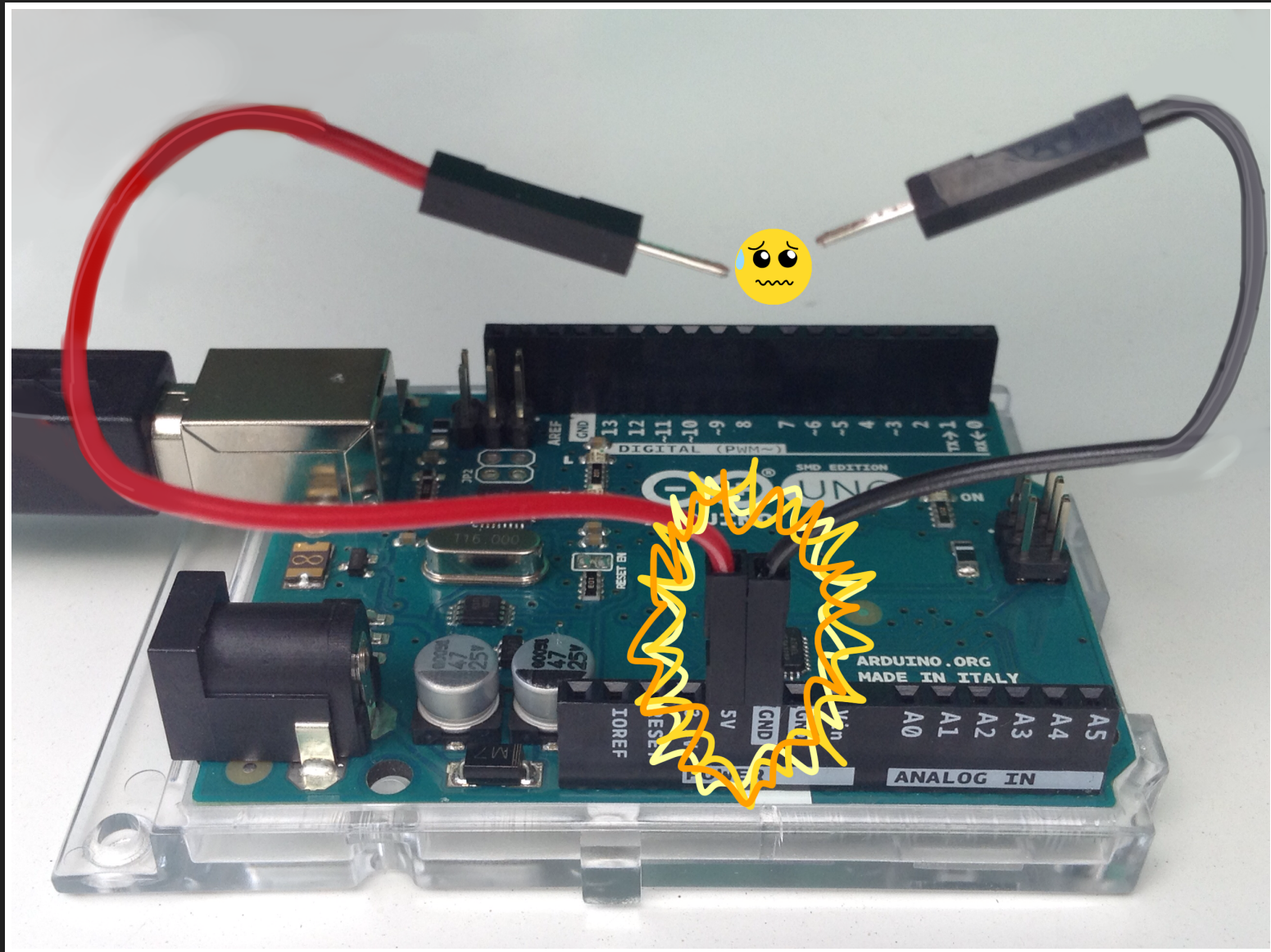
STROM - ANWENDUNGSGRENZEN

Höchstens 40 mA per Pin, empfohlen 20mA

Mehr Infos

<https://playground.arduino.cc/Main/ArduinoPinCurrentLimitation>

ACHTUNG KURZSCHLUSS!



STROMVERSORGUNG

Arduino kann mit einer externen Stromquelle von **Gleichstrom** bis 12 V verbunden werden. Dazu verwendet man entweder:

- den Pin "Vin" (bis 12V)
- den EIAJ-Steckverbinder (bis 12V)
- den 5V-Pin, bis 5V
- den USB-Stecker, bis 5V

Arduino zieht typischerweise nicht viel Strom, so dass eine PowerBank nicht richtig funktioniert (sie schaltet sich nach zwei Minuten aus)

ARDUINO - DIE SOFTWARE

```
void setup() {  
    // code: Einmal beim Starten  
  
}  
  
void loop() {  
    // code: Unendliche Schleife  
  
}
```

EINE LED BLINKEN LASSEN:

Den roten Draht von 5V auf digital-Pin 13 umstecken

```
void setup() {  
    pinMode(13, OUTPUT); // Definiert Pin 13 als Ausgabe  
}  
  
void loop() {  
    digitalWrite(13, HIGH); // 5V auf Pin 13  
    delay(500);             // Warte 500 Millisekunden  
    digitalWrite(13, LOW);  // 0V auf Pin 13  
    delay(200);             // Warte 200 Millisekunden  
}
```

Verbinden und hochladen

3) Programm zum Arduino übertragen

```
sketch_01_LED_Blink §  
  
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

2) unter dem Menüpunkt „Werkzeuge“ Mikrokontroller und Port auswählen

1) Arduino per USB-Kabel mit dem Rechner verbinden

Board: "Arduino/Genuino Uno"

Boardverwalter...

- Arduino AVR-Boards
- Arduino Yún
- ✓ Arduino/Genuino Uno
- Arduino Duemilanove or
- Arduino Nano
- Arduino/Genuino Mega
- Arduino Mega ADK
- Arduino Leonardo
- Arduino/Genuino Micro
- Arduino Esplora

ARDUINO-SPRACHE

<https://www.arduino.cc/en/Reference>

```
// Einzeiliger Kommentar
/* Mehrzeiliger Kommentar */

// Digital Pins: 1, 2, 3, ... , 12, 13 // PWM Werte 0-255 für analogWrite()
// Analog Pins: A0, A1, A2, A3, A4, A5 // Werte 0-1023 für analogRead()

pinMode(13, OUTPUT); // Definiert Pin 13 als Ausgabe
pinMode(A2, INPUT); // Definiert Pin A2 als Eingabe

digitalRead(13); // liest den (digitalen) Wert vom Pin 13
analogRead(A0); // liest den (analogen) Wert von Pin A0

digitalWrite(13, LOW); // schickt den Wert LOW (0V) auf Pin 13
digitalWrite(12, HIGH); // schickt den Wert HIGH (5V) auf Pin 12

analogWrite(10, 105); // schickt den Wert 105 auf den PWM-Pin 10

delay(1200); // Warte 1200 Millisekunden
delayMicroseconds(3000); // Warte 3000 Mikrosekunden
```

// Grundoperationen:

+ Summe

- Subtraktion

* Multiplikation

/ Division (9/2 ist 4: ganzzahlige Division zwischen ganzen Zahlen)

% Modulo (Rest der ganzzahligen Division)

// Vergleichsoperatoren:

< <= kleiner, kleiner oder gleich,

> >= größer, größer oder gleich

== ist gleich

!= ist ungleich

Logische Operatoren

&& Und

|| Oder

! Nicht

```
// Datentypen
int // ganze Zahl, 16-Bit
byte // ganze Zahl, 8-Bit
long // ganze Zahl, 32-Bit
unsigned long // ganze Zahl, nur positiv, 32-Bit
float // Gleitkommazahlen
String // Zeichenkette
char // Buchstabe
boolean // Wahrheitswert

// Deklaration einer Variable mit <Datentyp> <Bezeichner>;
int x;
float y;
boolean z;

// Wertzuweisung
x = 10;
y = 0.42;
z = true;
```

// Definition von Funktionen/Methoden

```
void meine_funktion_1() {  
    // tue was  
}
```

Ohne Rückgabe

Ohne Eingabeparametern

```
int meine_funktion_2(int ein1, int ein2) {  
    // tue was  
    return ein1 + ein2;  
}
```

Rückgabe

zwei Eingabeparameter

KONTROLLSTRUKTUREN: IF

```
if (Bedingung) {  
    // tue was  
}
```

```
if (Bedingung_1) {  
    // tue A  
} else if (Bedingung_2) {  
    // tue B  
} else {  
    // tue C  
}
```


KONTROLLSTRUKTUREN: FOR

```
.  
.  
for (Anfangswert; Bedingung; Schrittweite) {  
    // tue was  
}
```

```
.  
.  
for (int x = 5; x < 100; x = x + 10) {  
    // tue was  
}
```

AUSGABE AUF DEN SERIELLEN MONITOR

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    for (int x = 0; x < 10; x++) {  
        Serial.println(x);  
    }  
}
```

KONTROLLSTRUKTUREN: WHILE

```
while (Bedingung) {  
    // tue was  
}
```

ARRAY

```
int a[5]; // Array von ganzen Zahlen der Länge 5
boolean b[100]; // Array von Wahrheitswerten der Länge 100

a[0] // erstes Element von Array a
b[99] // letztes Element von Array b

// serielle Ausgabe aller Elemente von a:
for(int i = 0; i < 5; i++) {
    Serial.println(a[i]);
}
```

ÜBUNGEN LEDES MIT ARDUINO

1. Zwei LEDs abwechselnd blinken lassen.
2. Drei LEDs, eine der drei geht zufallsgesteuert an: Der Befehl `random(3) ;` erzeugt zufällig 0, 1 oder 2.
3. Eine LED mit einem PWM-Pin dimmen: Die digitalen Pins mit einem "~" (PWM-Pins, "pulse-width modulation") können mit dem Befehl `analogWrite(pin, wert)` Werte zwischen 0 und 255 ausgeben (duty cycle). Mit einer for- oder while-Schleife.
4. Mit `millis() ;` und dem Modulo-Operator eine Reihe von LEDs "laufen" lassen.

EXPERIMENTIEREN SIE MIT EINEM RGB-LED



and more.de

5mm LED RGB 4-Pin Klar (Kathode)

Größe: Höhe 9,0mm / Ø 5mm

Rot / Grün / Blau

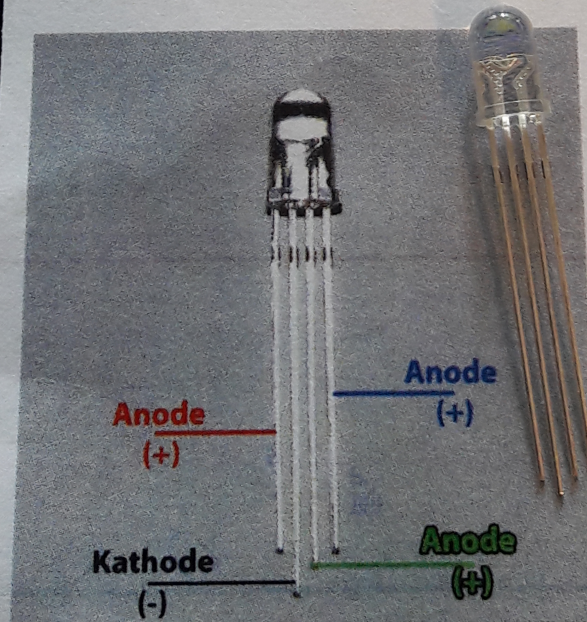
Spannung: Typ. 2,00 / 3,30 / 3,20 Volt

Lichtleistung: 1500 / 4000 / 3000 mcd

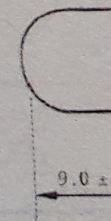
Lichtwellenlänge: 630 / 520 / 465 nm

Strom: 3 x 20 mA

Abstrahlwinkel: 40°



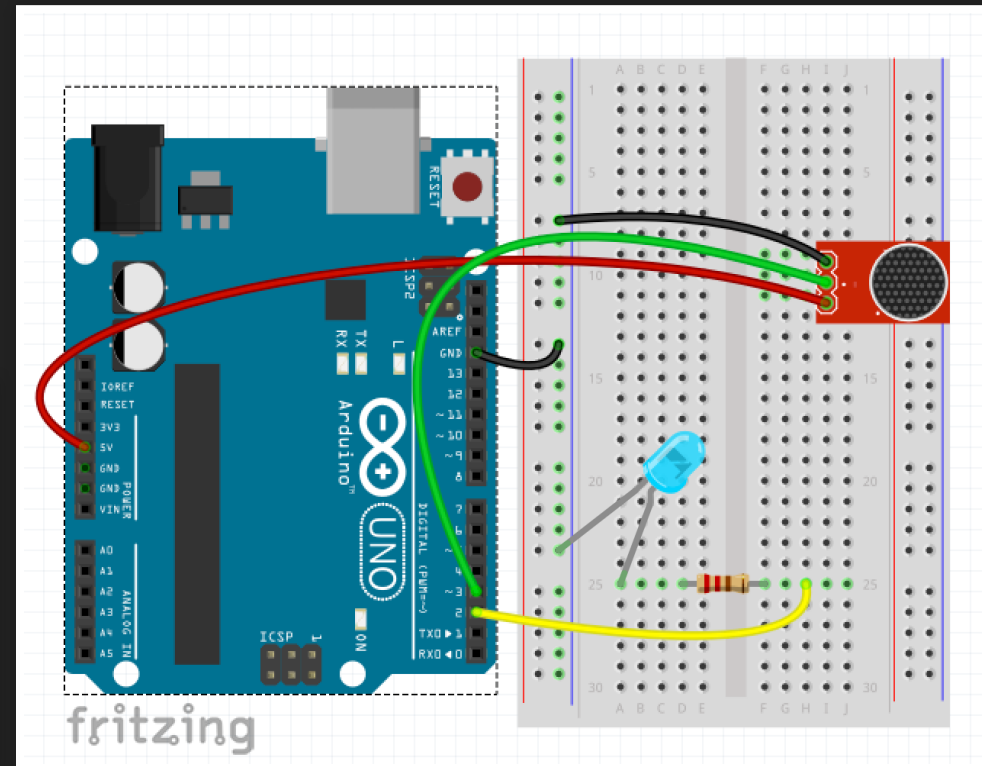
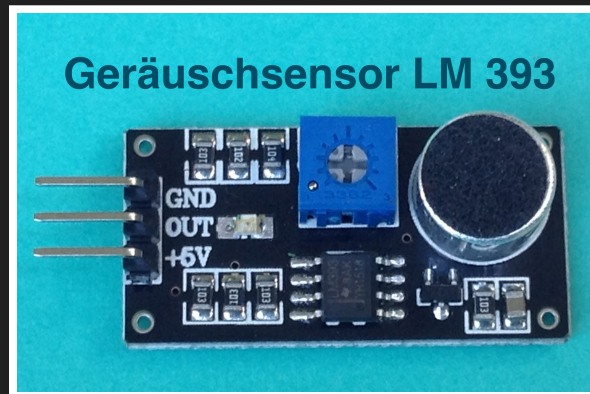
Package



ARDUINO - EINGABE

- Ein Pin (digital oder analog) kann als Eingabe verwendet werden um Signalen aus Sensoren zu empfangen
- `pinMode(13, INPUT);` schaltet den digitalen Pin 13 auf Eingabe und `pinMode(A0, INPUT);` den analogen Pin A0.
- `digitalRead(pin);` liest eine digitale Eingabe: Die möglichen Werte sind 0 (keine Spannung) oder 1 (5V Spannung).
- `analogRead(pin);` liest eine analoge Eingabe: Die möglichen Werte sind zwischen 0 und 1023 (inklusive). Der Wert 0 entspricht 0V, der Wert 1023 entspricht 5V.

BEISPIEL: DER DIGITALEN GERÄUSCHSENSOR



CODE - MIT SERIELLER ÜBERTRAGUNG!

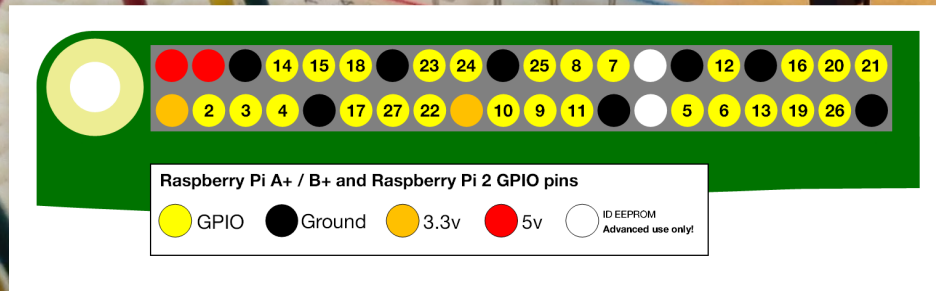
```
void setup() {  
  pinMode(2, OUTPUT); // Ausgabe LED  
  pinMode(3, INPUT); // Eingabe Geräuschsensor  
  Serial.begin(9600); // Serielle Übertragung  
}  
  
void loop() {  
  int sensorValue = digitalRead(3); // 0 wenn Geräusch, 1 wenn kein Geräusch  
  Serial.println(sensorValue); // Serielle Ausgabe (z.B auf den seriellen Monitor)  
  digitalWrite(2, !sensorValue); // LED an falls Geräusch  
}
```

BEISPIEL MIT ANALOGEM GERÄUSCHSENSOR

- Ein analoger Geräuschsensor, z.B. TE136.
- Pin "A0" mit A0 auf Arduino verbinden
- Pin "G" mit Erde
- Pin "+" mit 5V
- (Pin "D0" ist für eine digitale Ausgabe)

RASPBERRY PI

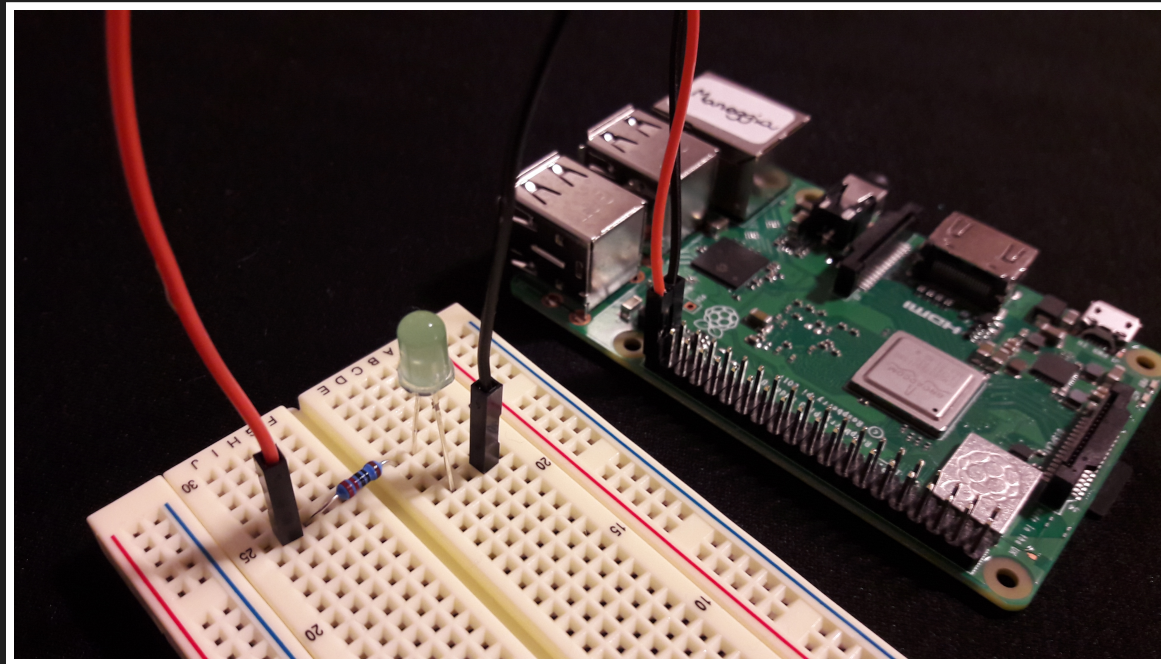
- Nummerierung der Pins



<https://www.raspberrypi.org/documentation/usage/gpio/>

- Stromversorgung: 5V Gleichstrom via USB-mikro

EINE LED MIT EINEM PYTHON-SCRIPT STEUERN



<http://www.instructables.com/id/Raspberry-Pi-LED-Blink/>

- Ein Terminal öffnen und mit `mkdir bin` einen Ordner "bin" erzeugen
- in dem Ordner bin/ mit dem Editor "nano" (oder "vi") eine Datei `gpio.py` erzeugen
 - `cd bin`
 - `nano gpio.py`
- in dieser Datei die Befehle wie im Tutorial eingeben (eventuell mit Pin 21 statt Pin 18)
- mit Strg+X die Datei speichern
- mit `python gpio.py` den script ausführen

EVENTUELL: DIE DATEI AUSFÜHRBAR MACHEN

- Die sha-bang-Zeile `#!/usr/bin/python` am Anfang der Datei ergänzen (mit nano die Datei so bearbeiten)
- mit `chmod u+x gpio.py` die Datei ausführbar machen
- Mit `gpio.py` die Datei ausführen

Four small, colorful light-emitting diodes (LEDs) are arranged in a slightly curved line across the frame. From left to right, they are blue, red, green, and yellow. Each LED has a translucent, dome-shaped top and two long, thin, gold-colored pins extending downwards. The background is a dark, textured surface, possibly black fabric or a similar material. The word "DANKE!" is superimposed in the center of the image in a bold, white, sans-serif font.

DANKE!