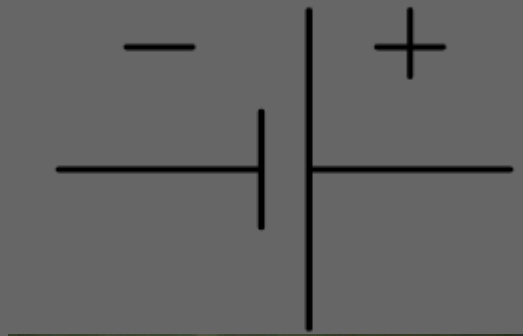
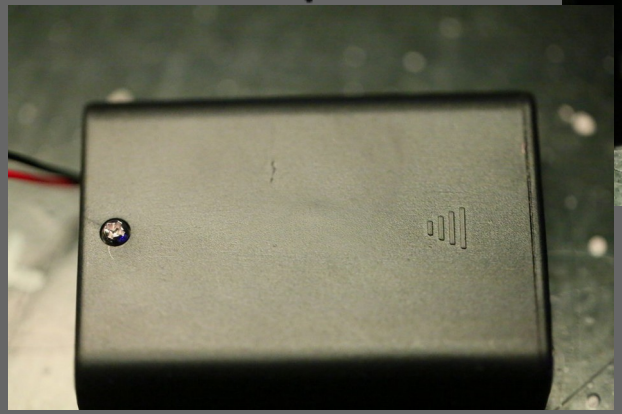
Foliensatz

- coderdojo.red/images/kyo-6/ATTiny.pdf
- erste Lötübung auch auf coderdojo.red/kyo-7/

Batterie einsetzen

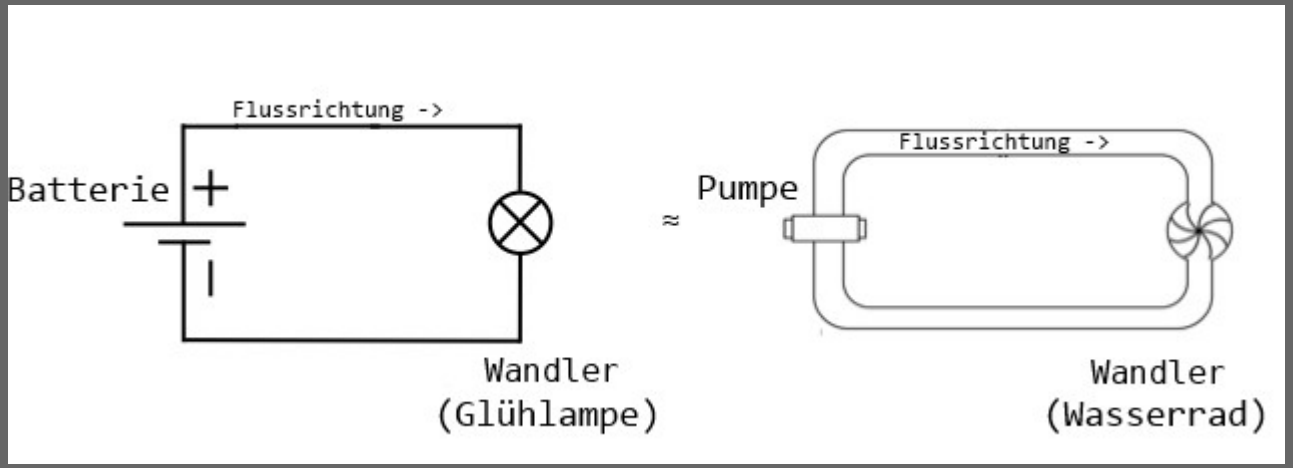


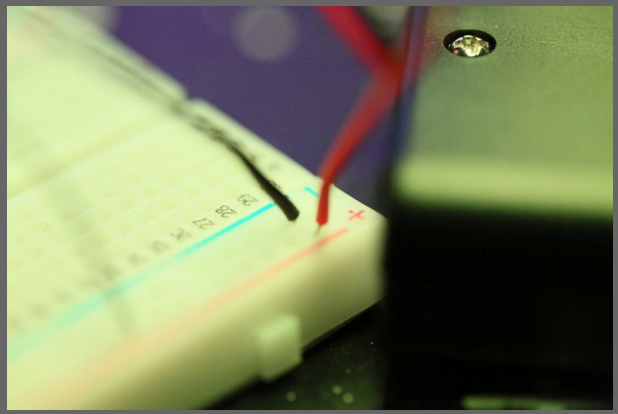


1) Schraube auf
2) aufschieben
3) Batterien gepolt einsetzen
4) schließen
5) Schraube rein

Schaltung aufbauen

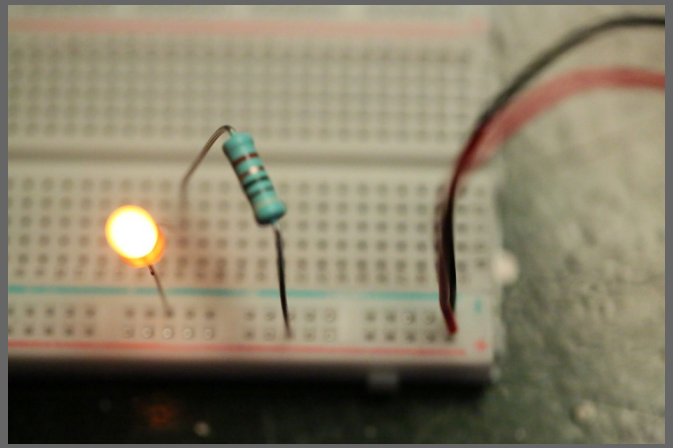
- Stromkreis aufbauen





Schaltung aufbauen

- Widerstand 220 Ω

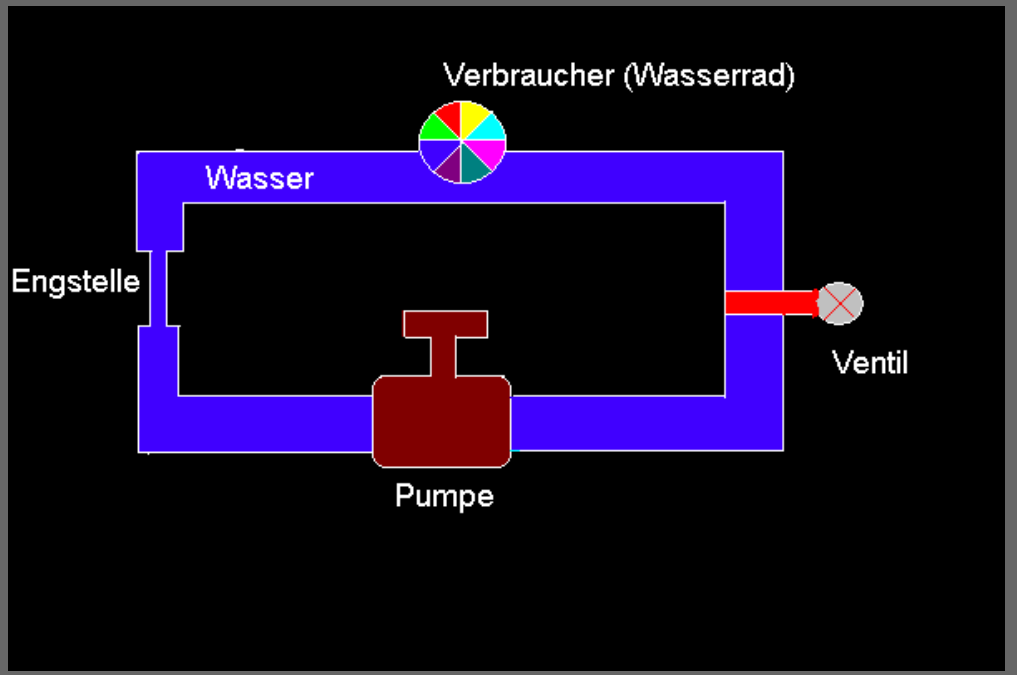


Schaltung aufbauen

- Widerstand

  → reduziert die

  Strom / Spannung



Schaltung aufbauen

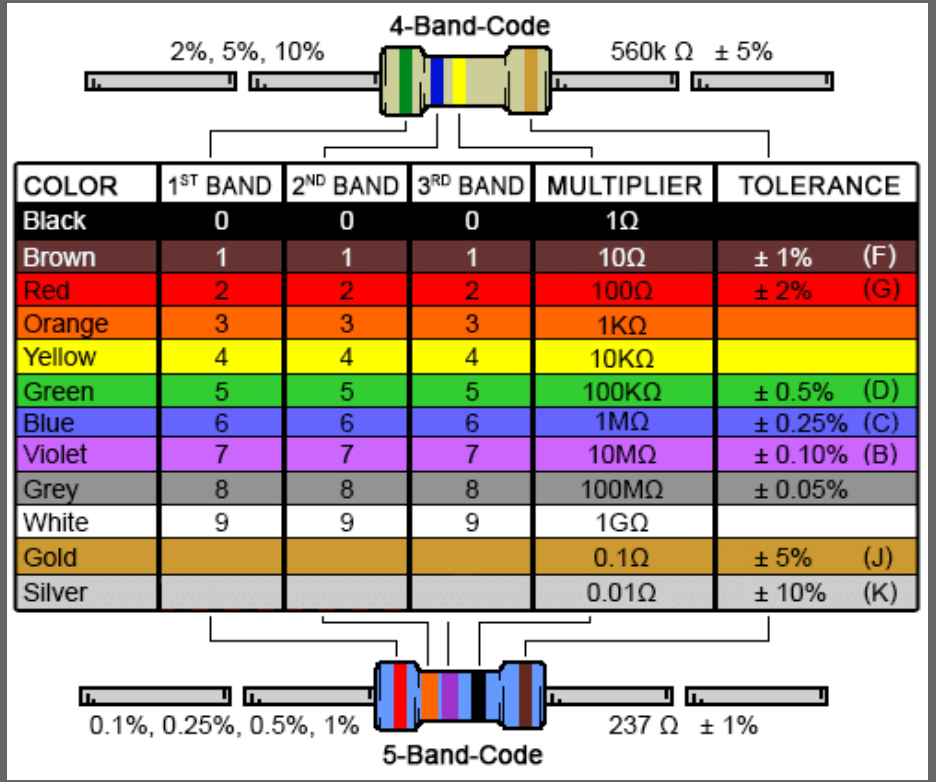
- Widerstand

  → reduziert die

  Strom /
  Spannung

Schaltung aufbauen

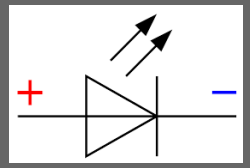
- Widerstand



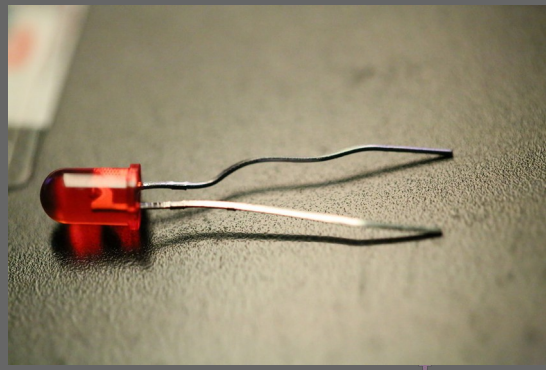
COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8	100MΩ	± 0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	± 5%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

Schaltung aufbauen

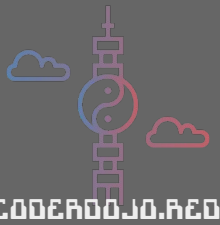
- LED



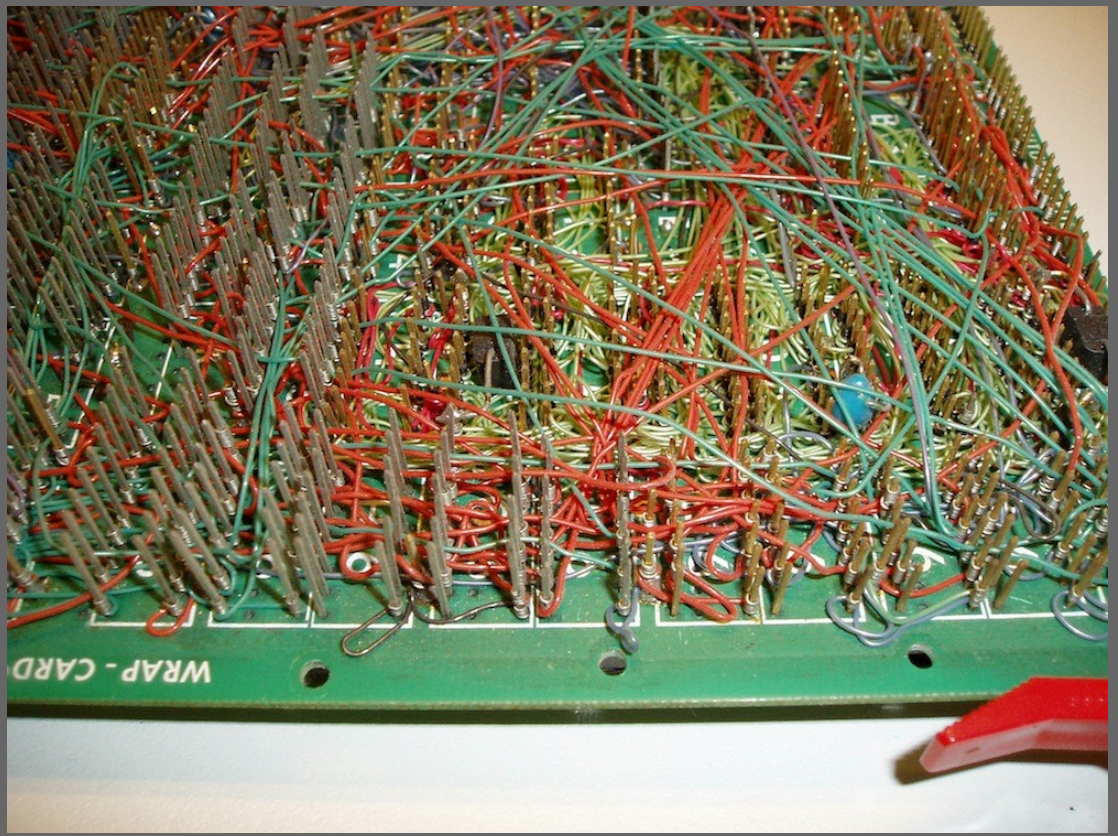
  - kurzes Bein → Kathode → – - Pol
  - langes Bein → Anode → + - Pol
  - LEDs mögen keine hohe Spannung, deshalb brauchen sie in aller Regel einen Widerstand!



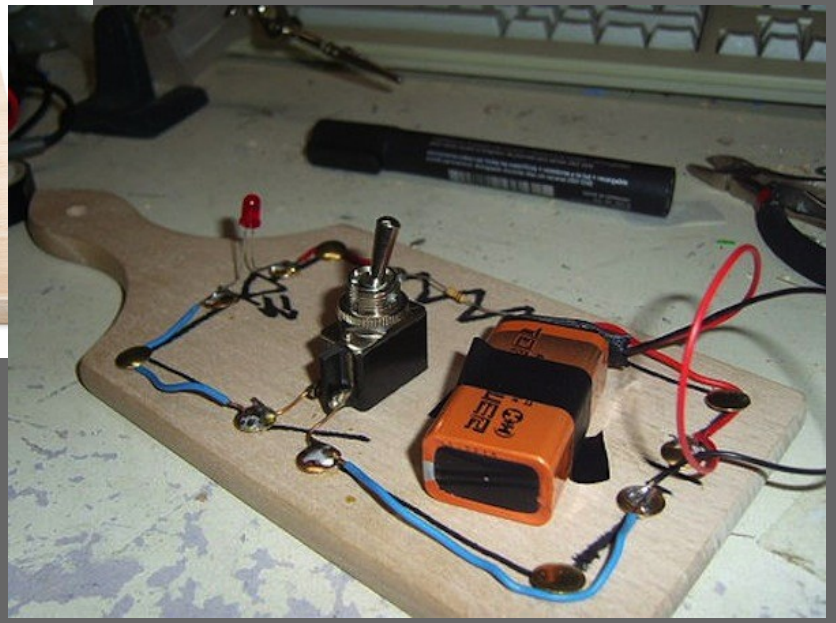
Warum Breadboard?



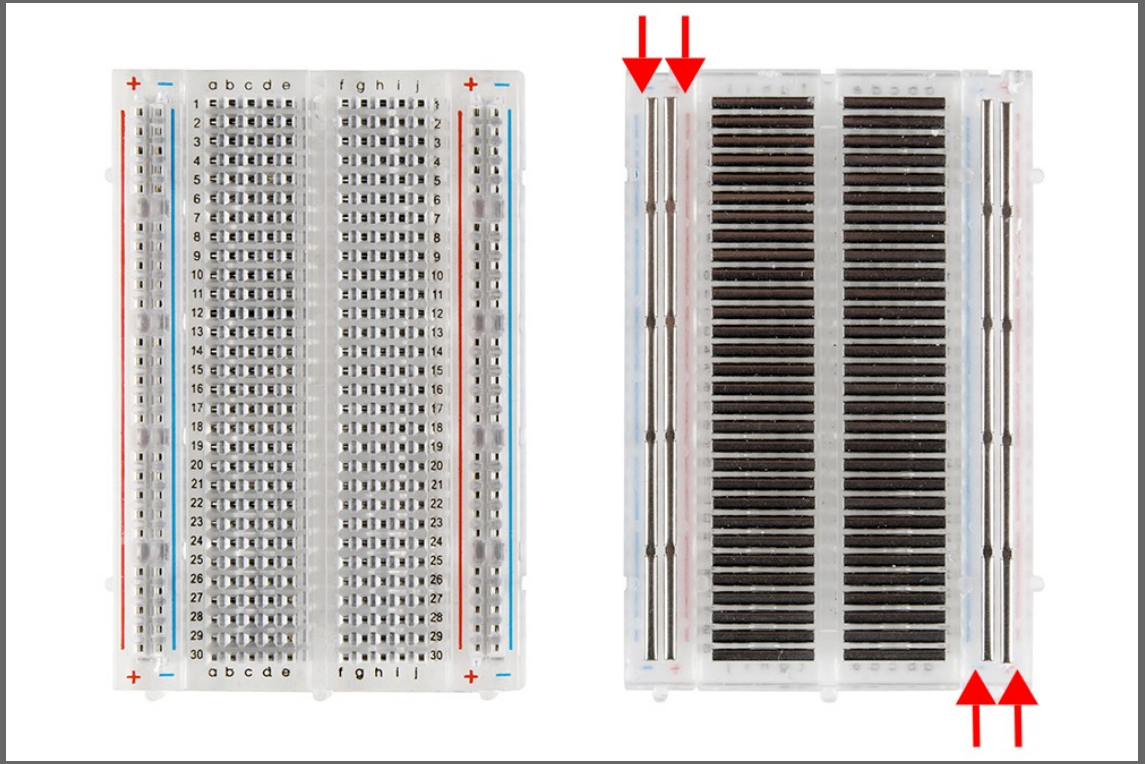
Darum Breadboard?



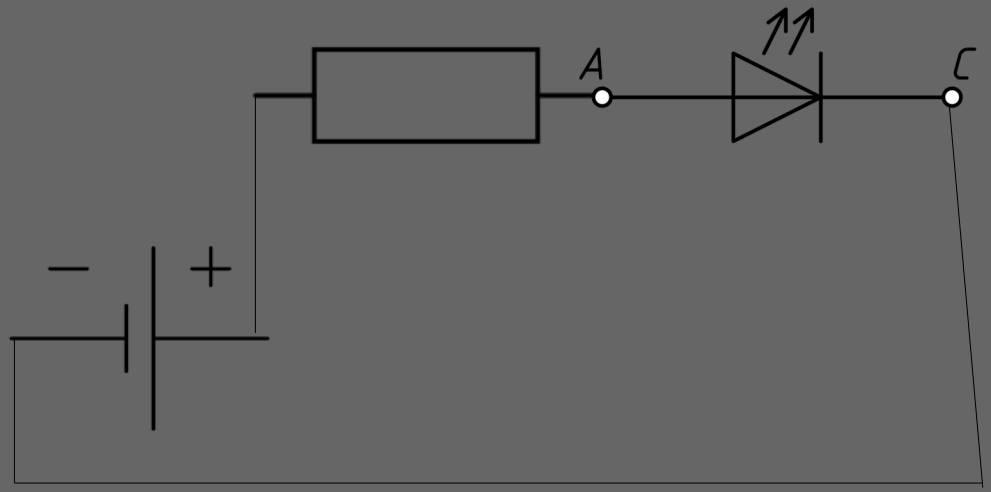
Erste Versuche der Besserung...

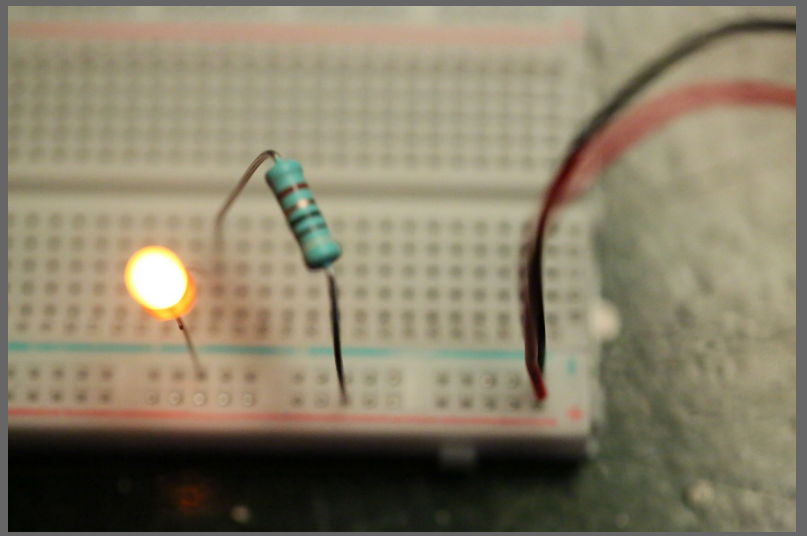


Wie funktioniert ein Breadboard?



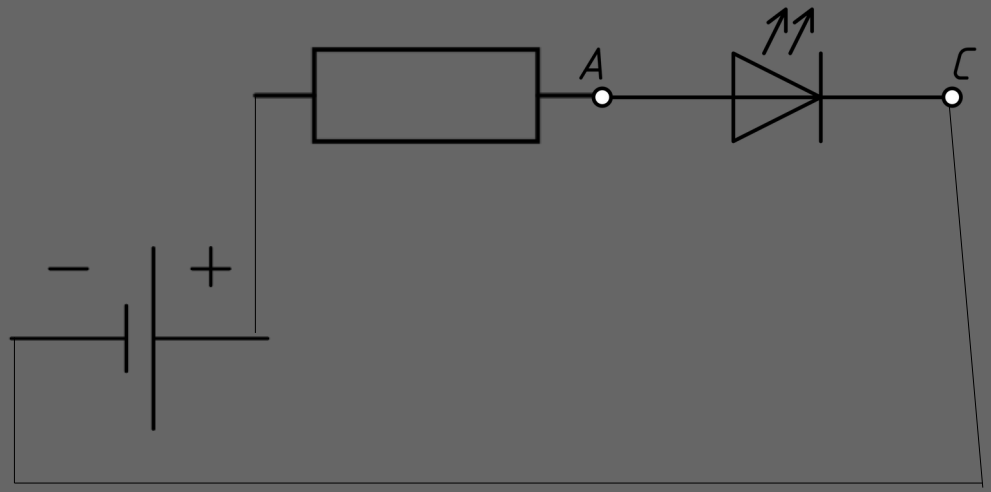
Schaltung aufbauen

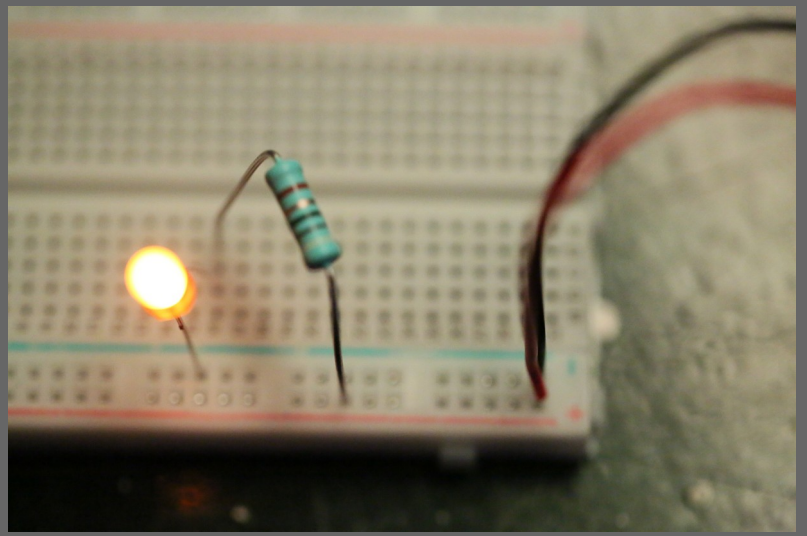




Schaltung aufbauen

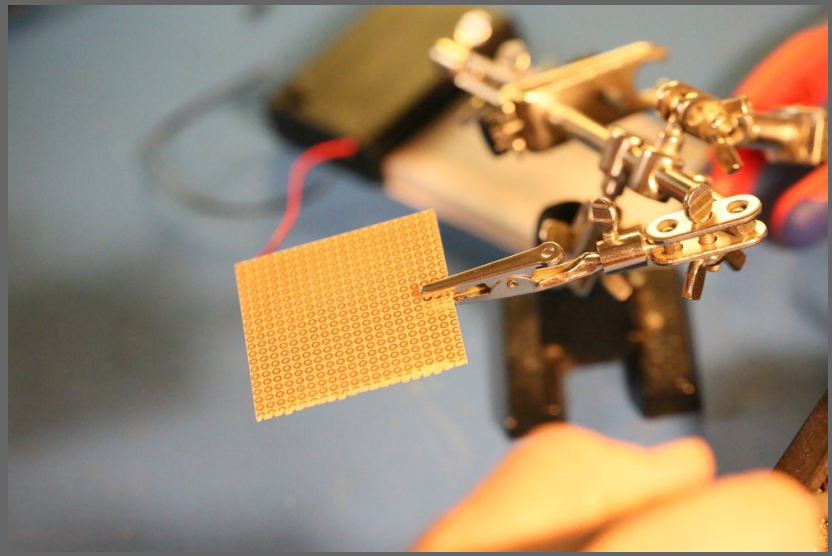
Schaltung aufbauen



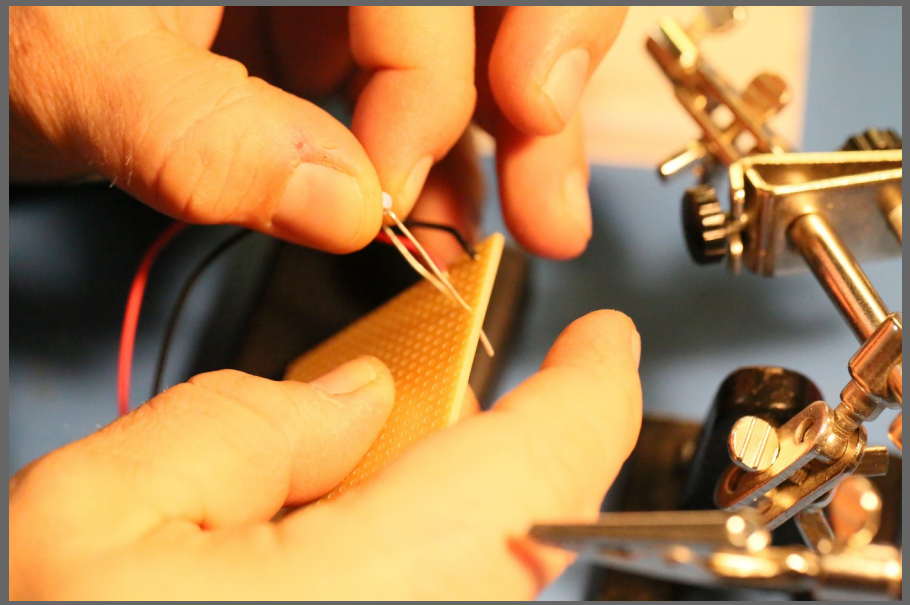


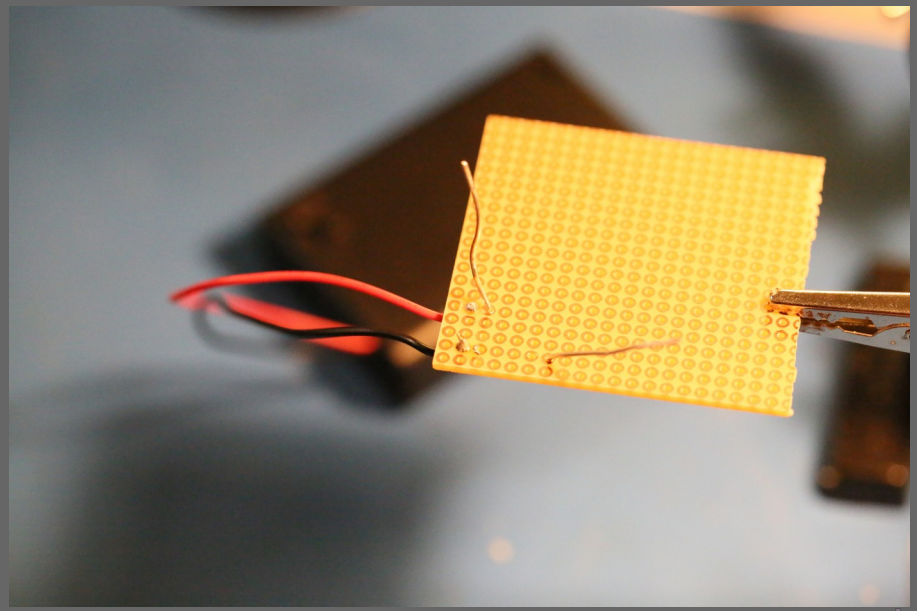
Löten

Löten

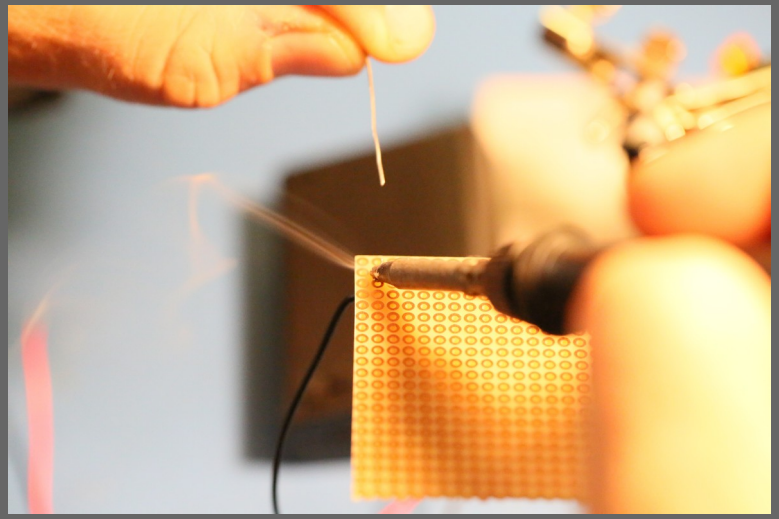


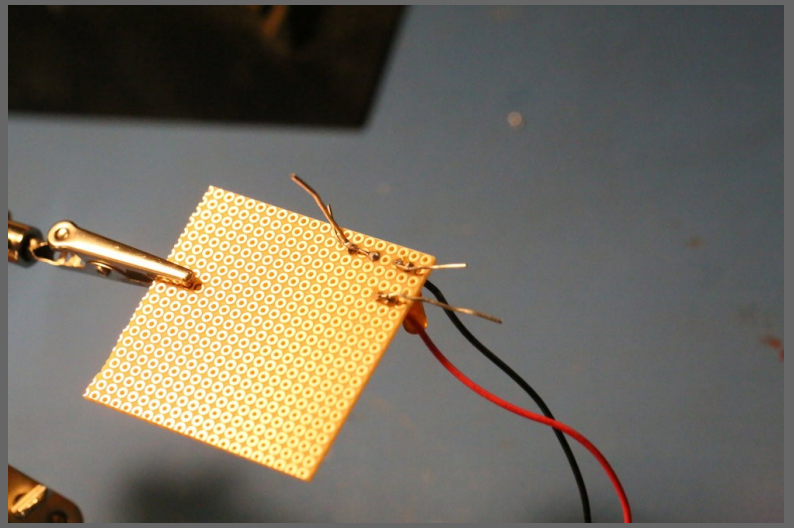
Löten



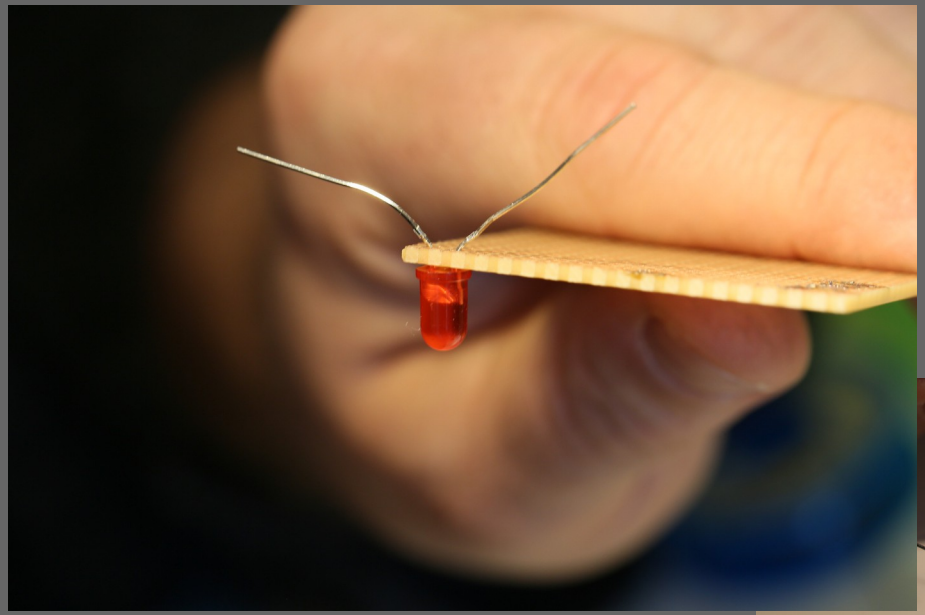


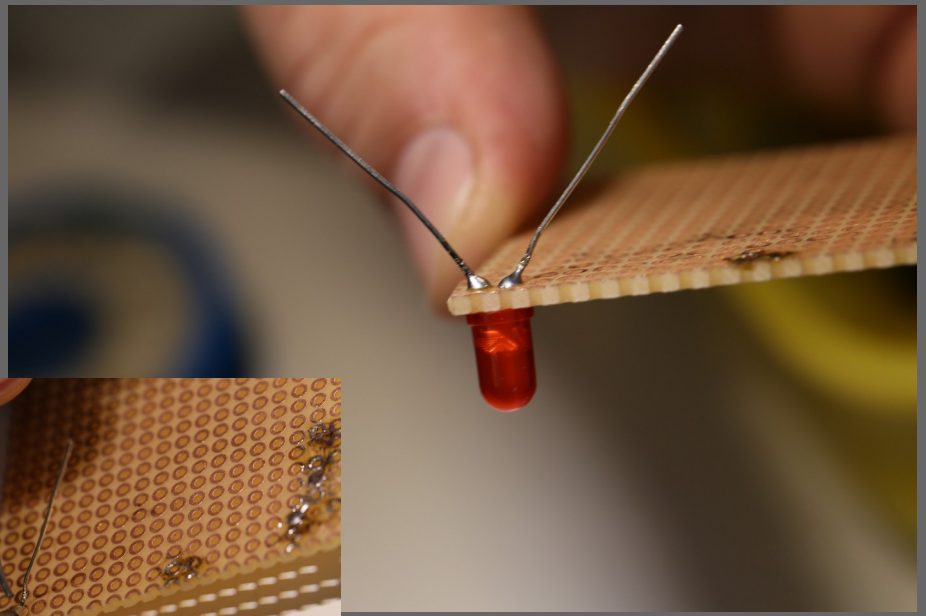
Löten

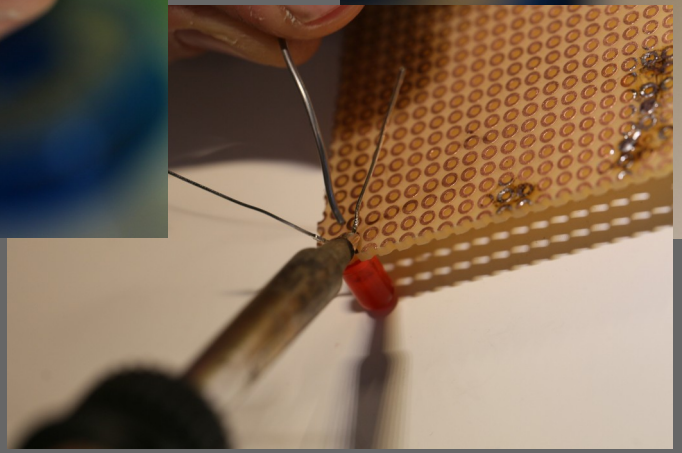




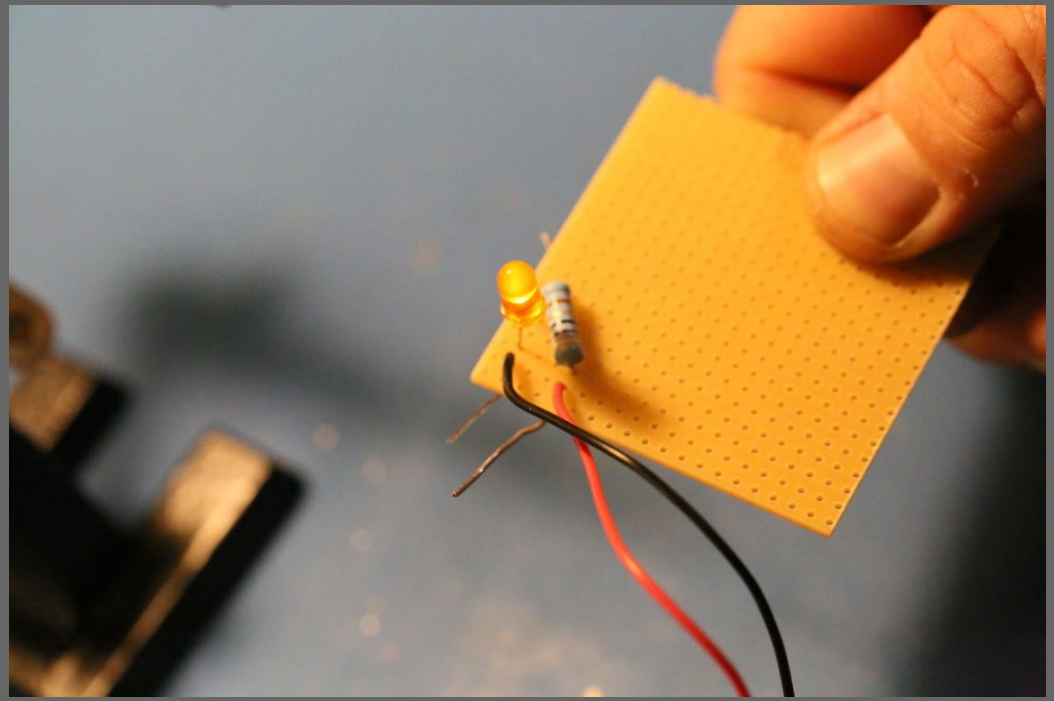
Löten







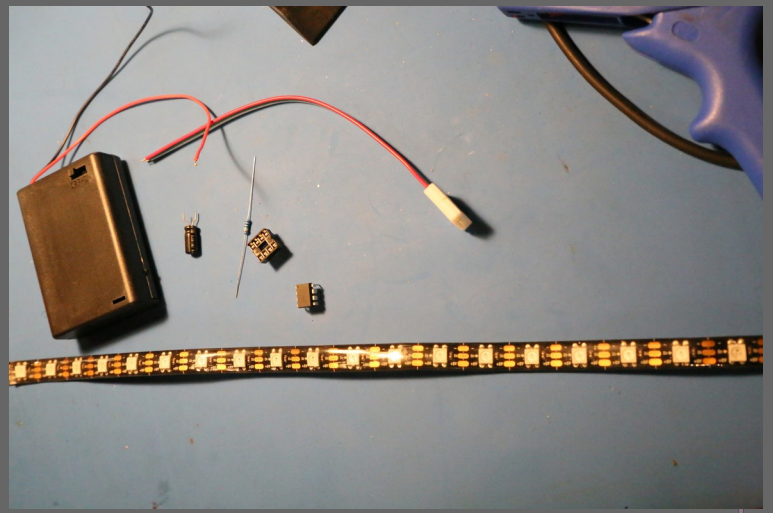
Löten



Strom an – leuchtet die LED?

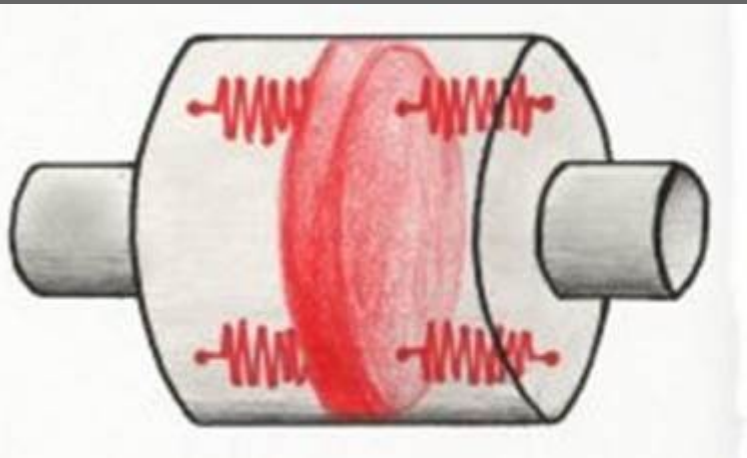
benötigte Bauteile – stückchenweise

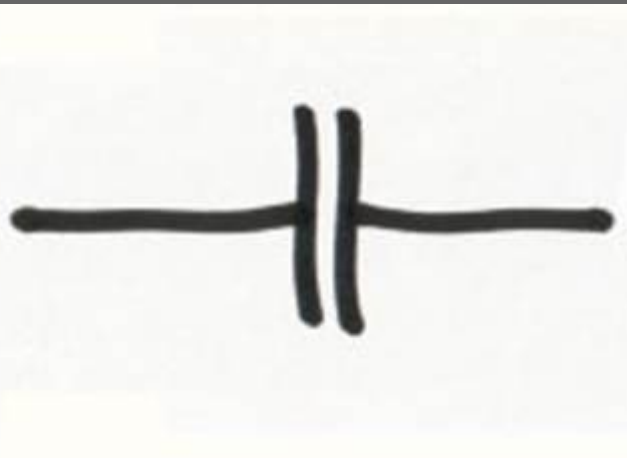
- 220 Ω Vorschalt-Widerstand
- 10 µF Kondensator
- IC-Sockelhalter mit IC
- LED-Stripe
- Stripe-Verbinder
- plus Teile von eben



Neue Bauteile – kurz erklärt

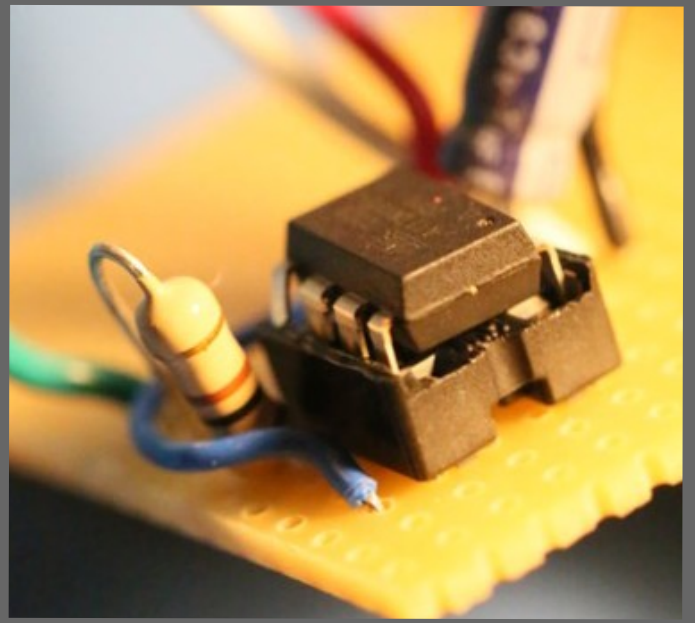
- Kondensator → F – Farad (Faraday → Käfig)
- fängt Spannungsspitzen ab (🍽 Wasserkreislauf)





Neue Bauteile – kurz erklärt

- IC → integrated circuit (integrierter Schaltkreis)
- enthalten vor allem Transistoren, aber auch Widerstände und Kondensatoren (und teilweise mehr)
- Transistoren sind vereinfacht Schalter
- ATTiny85 → Mikrocontroller, enthält noch Prozessor, also Recheneinheit und Speicher



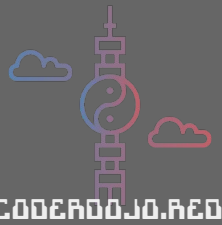
Blinklicht mit Microcontroller - vorher kurze Pause!

1. Schritt

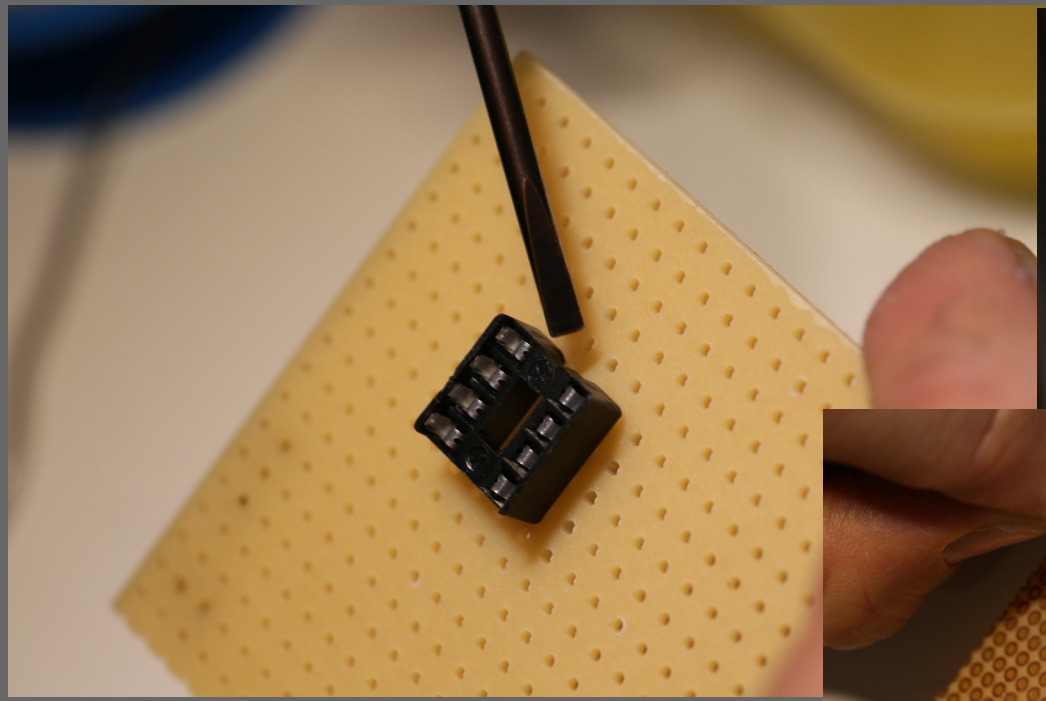
- Stromzufuhr, LED und Widerstand auslöten

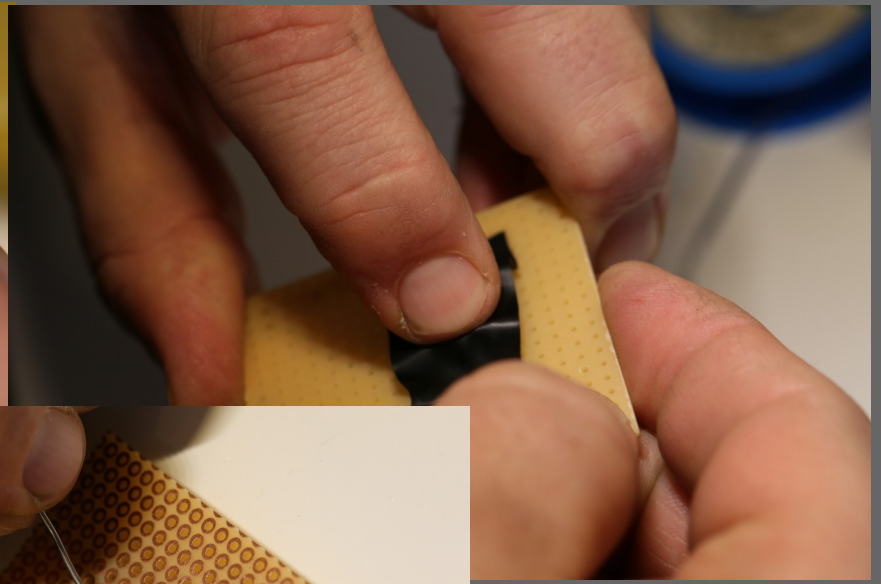
1. Schritt

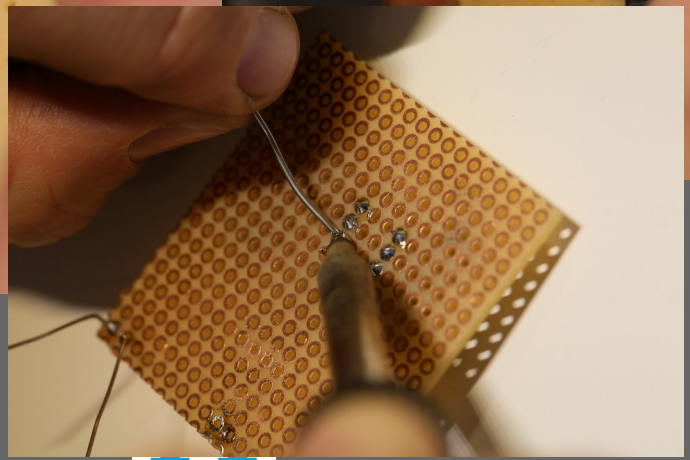
- IC-Sockelhalter anlöten
  - daran denken: Lötspitze an den Pin, dann Lötzinn drauf; nicht das Lötzinn an die Spitze halten und es “klecksen” lassen



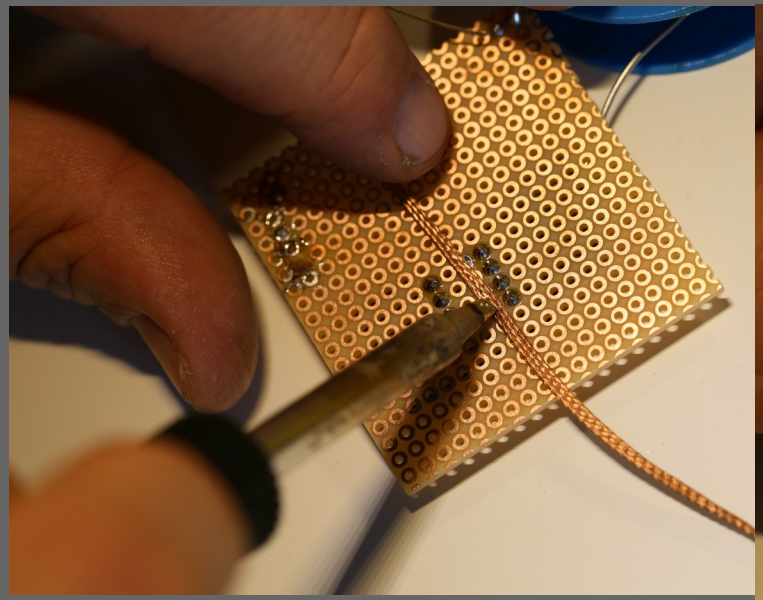
1. Schritt

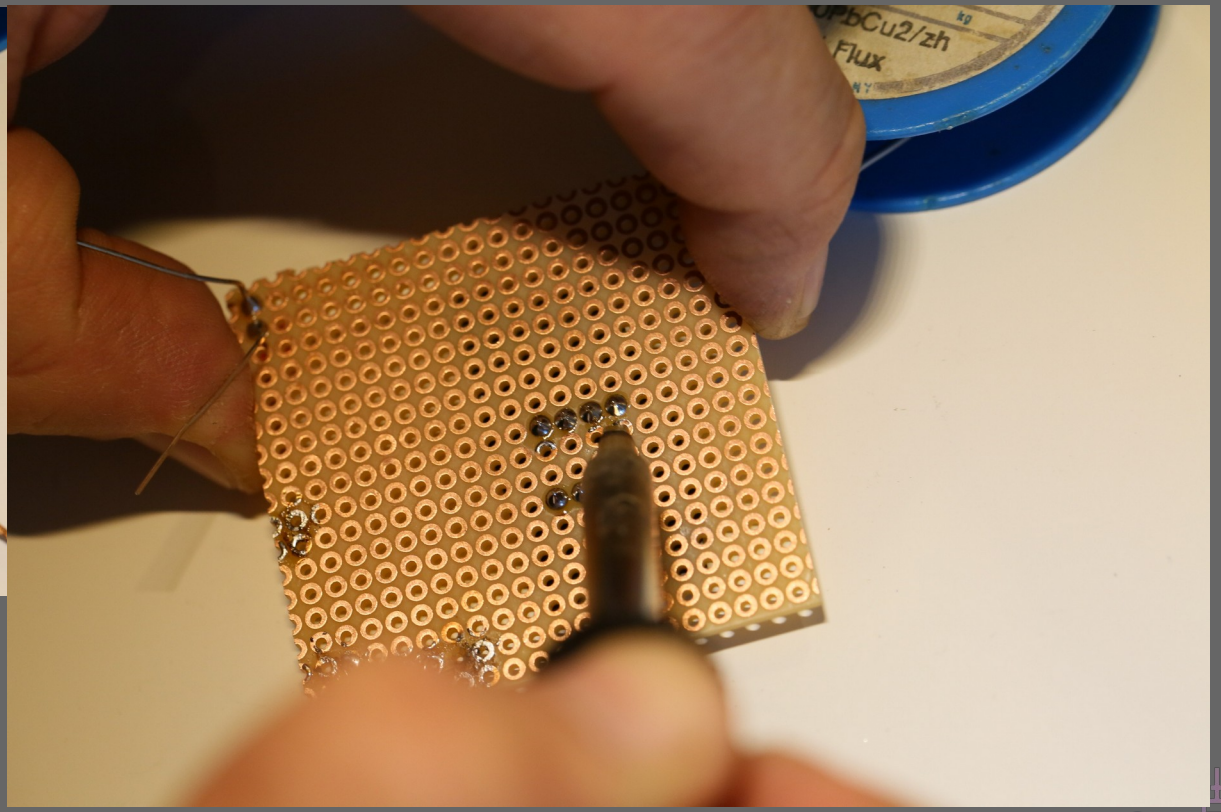






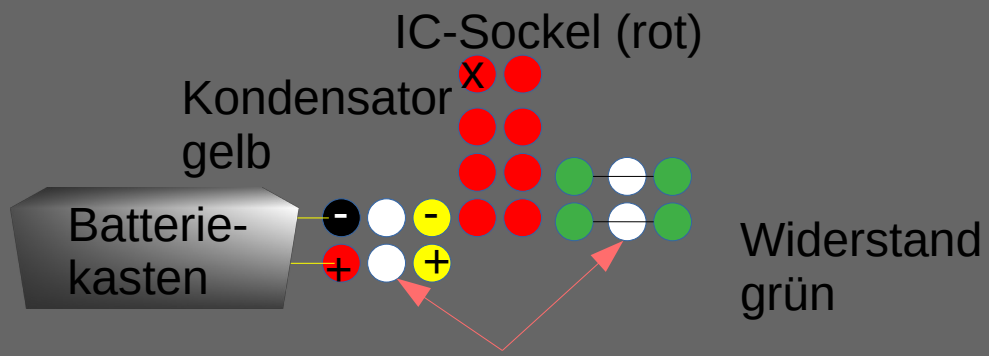
1. Schritt

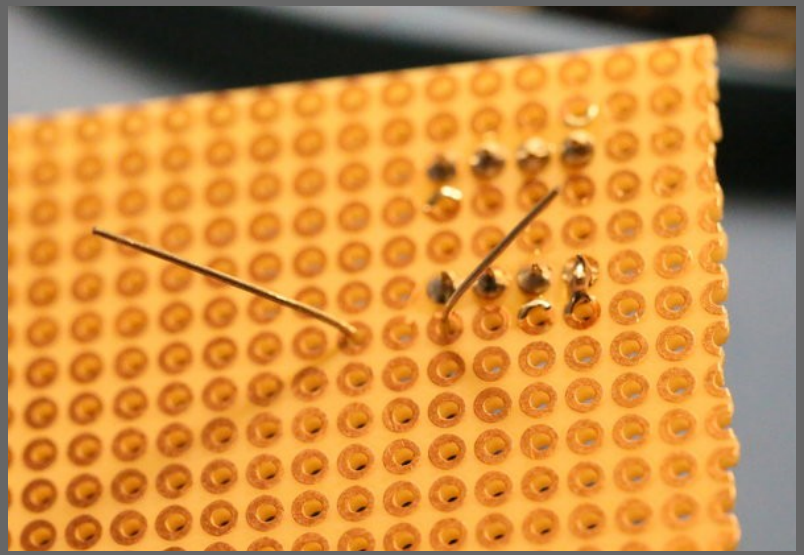




1. Schritt

- Sockel, Kondensator und Widerstände einlöten

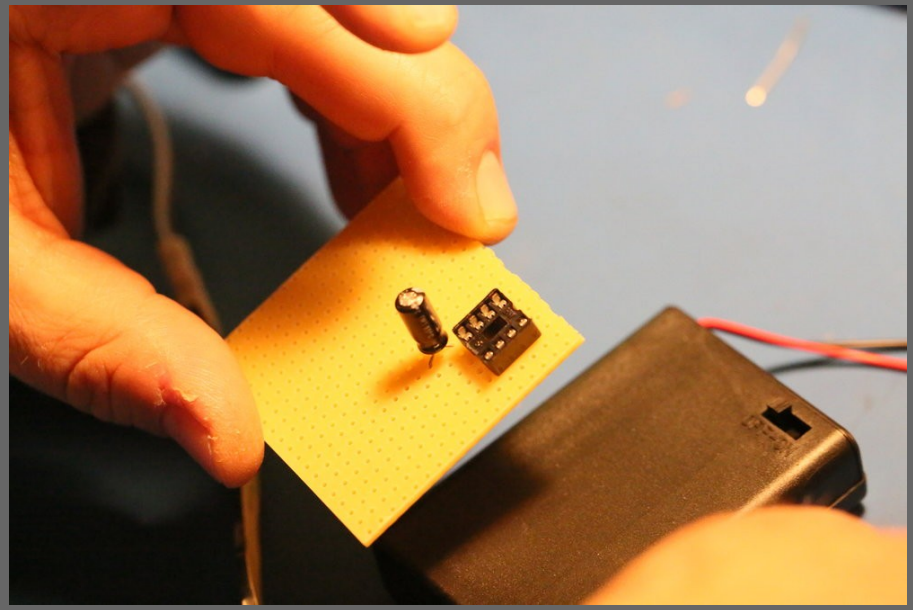


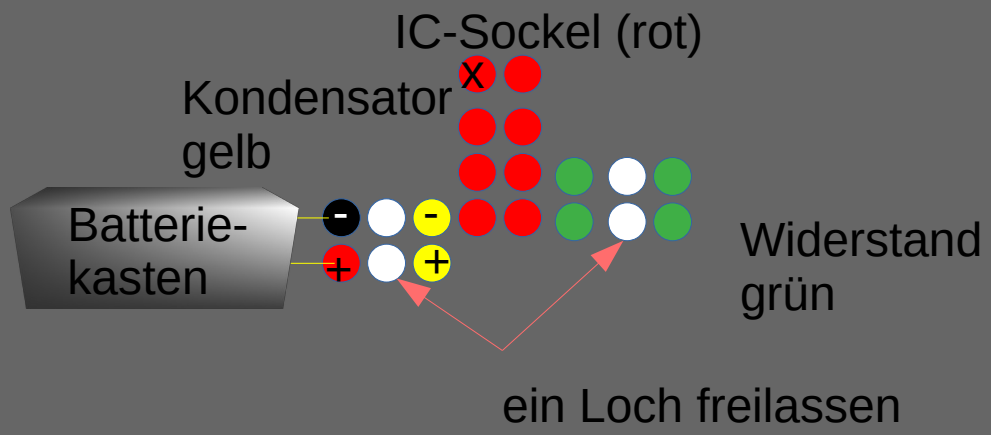


Aufsicht

2. Schritt

- Batteriekontakte einlöten

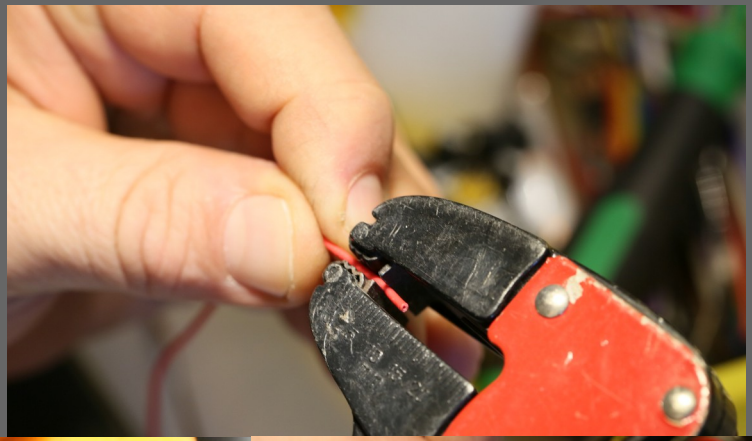


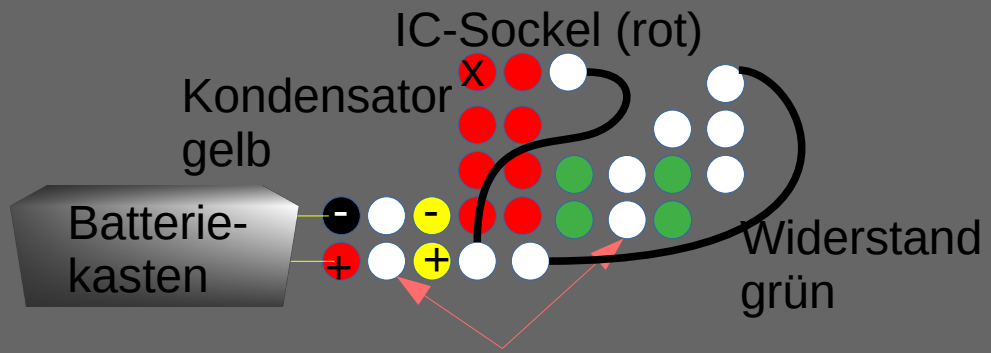


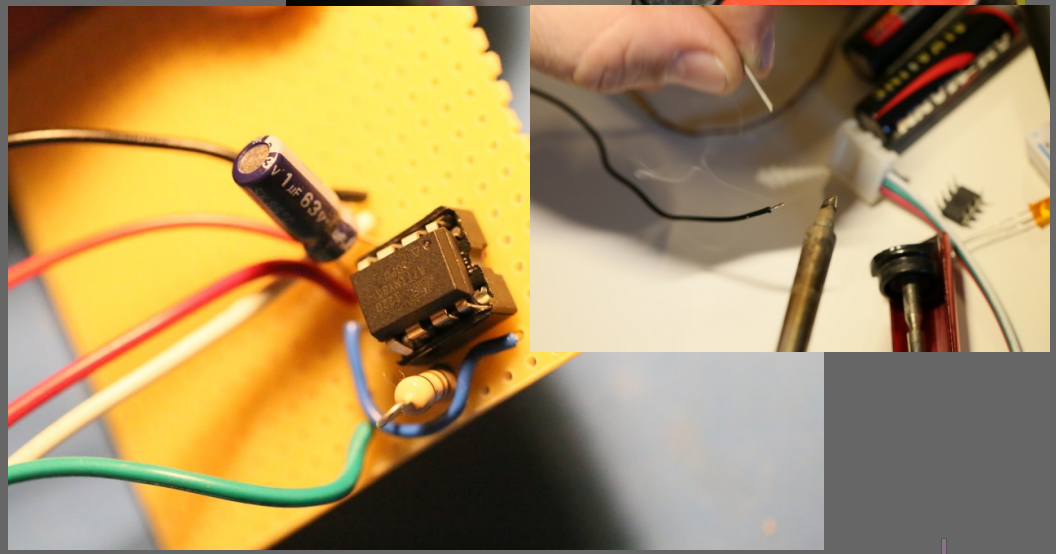
Aufsicht

3. Schritt

- Drahtbrücke löten



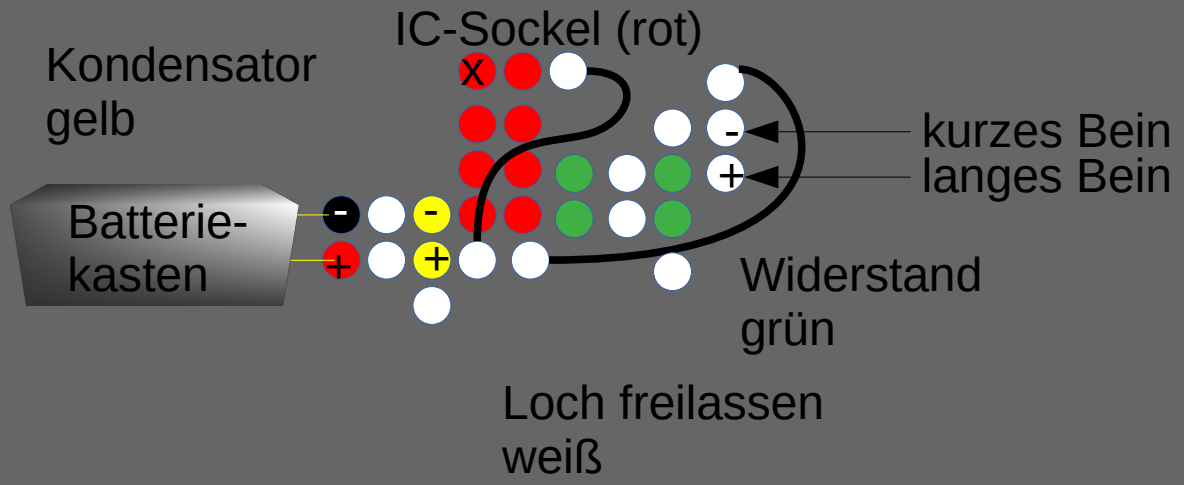


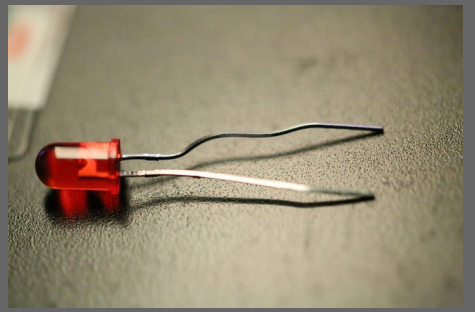


Aufsicht

4. Schritt

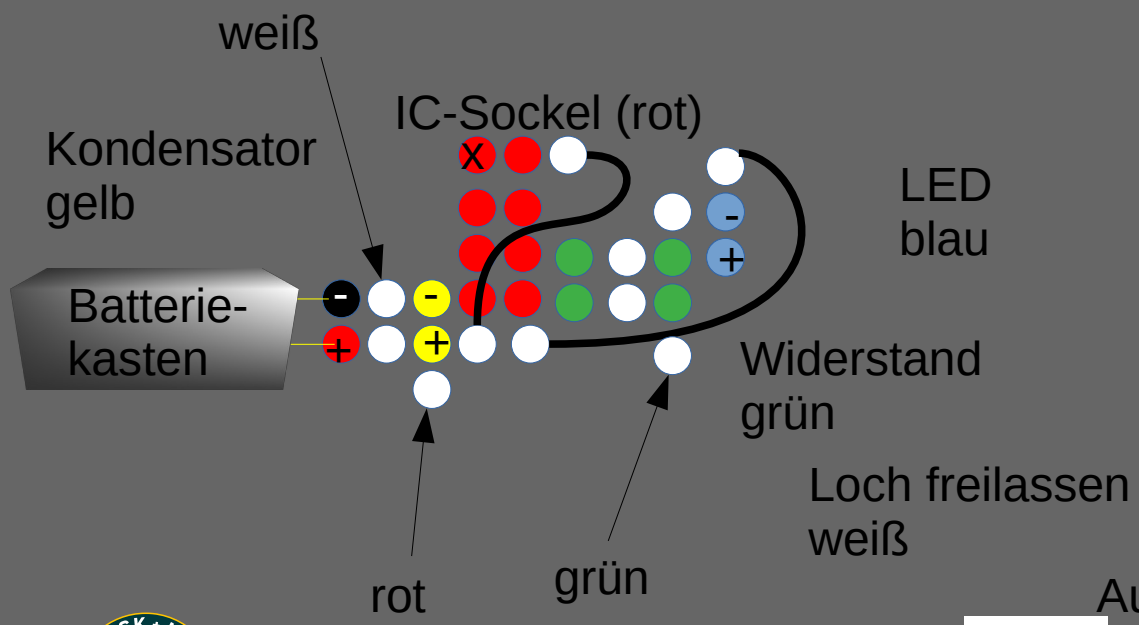
- LED einlöten – volle Konzentration bei der Polung!

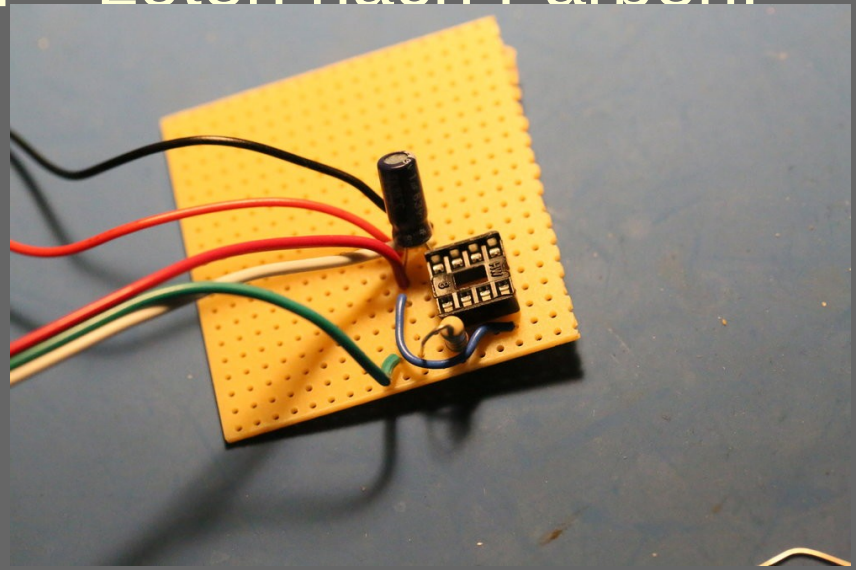




5. Schritt

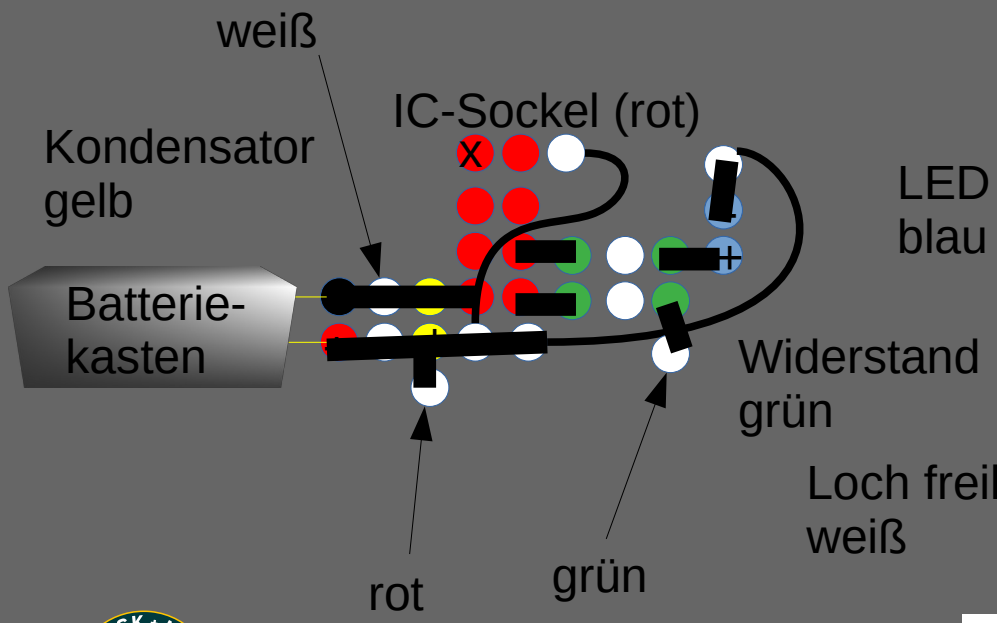
- LED-Stripe-Connector einlöten – Löten nach Farben!

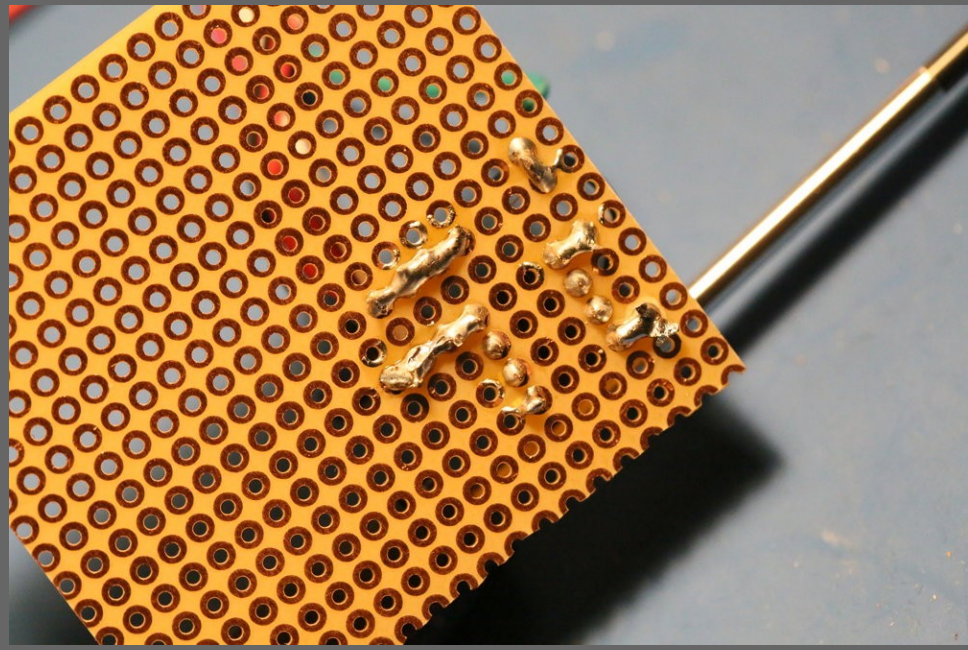




6. Schritt

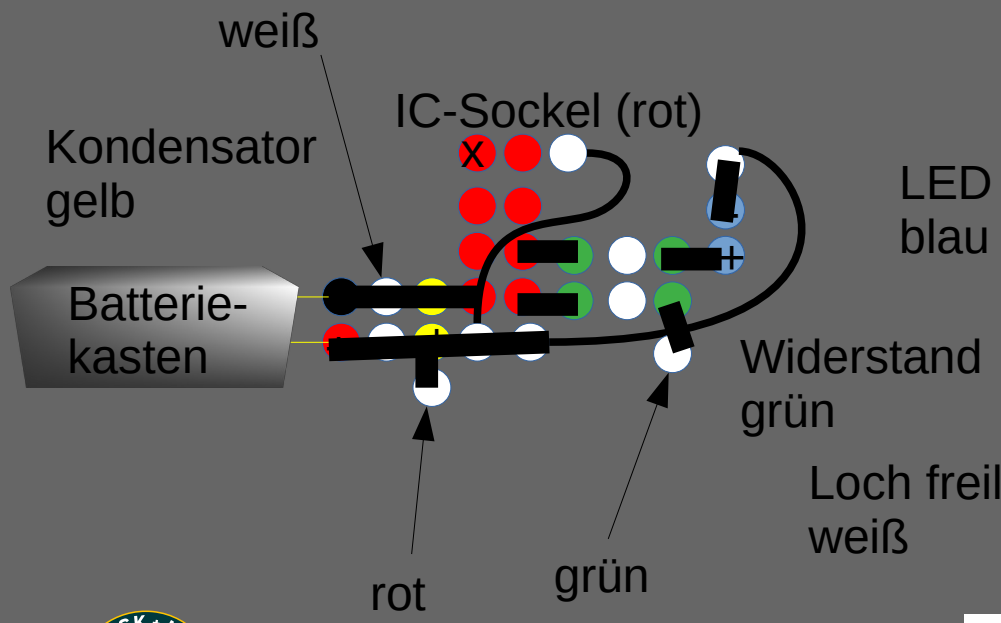
- Brücken löten

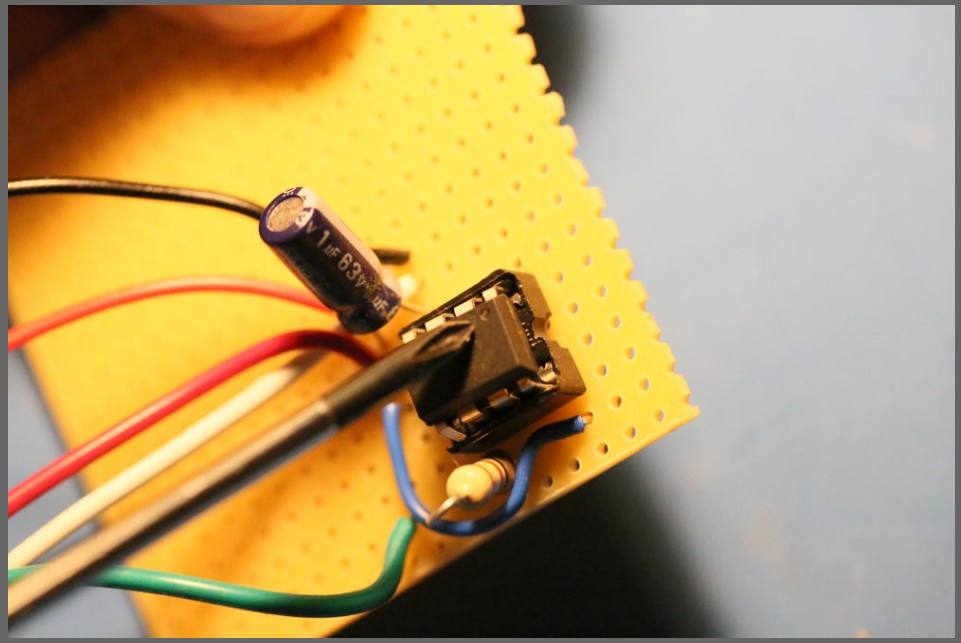




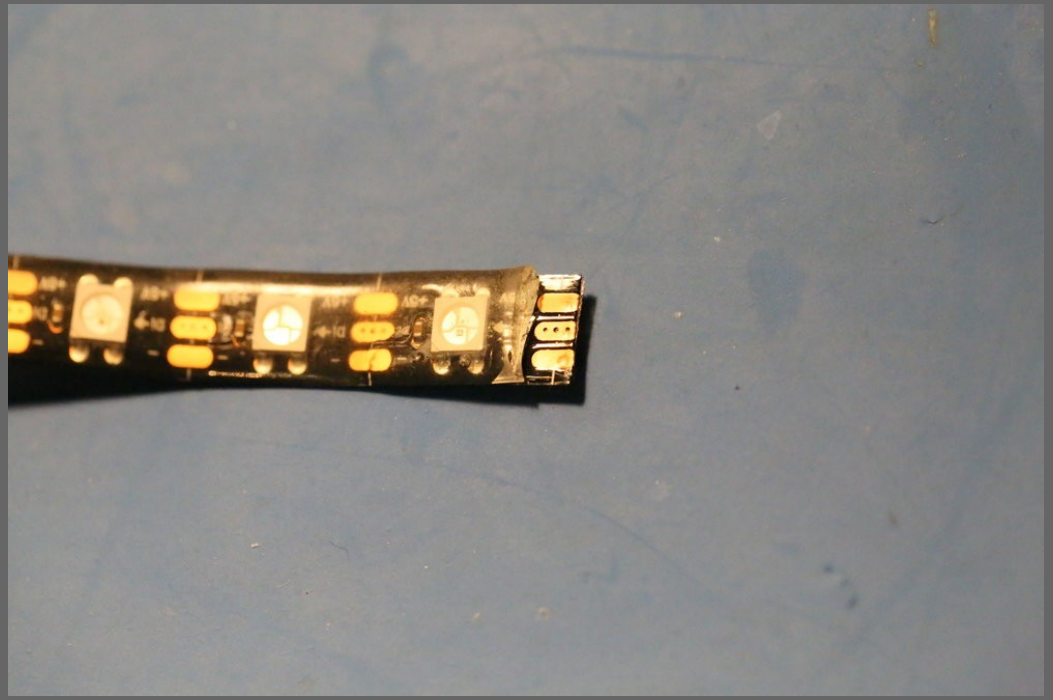
7. Schritt

- ATTiny85 einsetzen





LED-Stripe verbinden



- Stripe in Klemme legen
- Klemme schließen

Fruststop

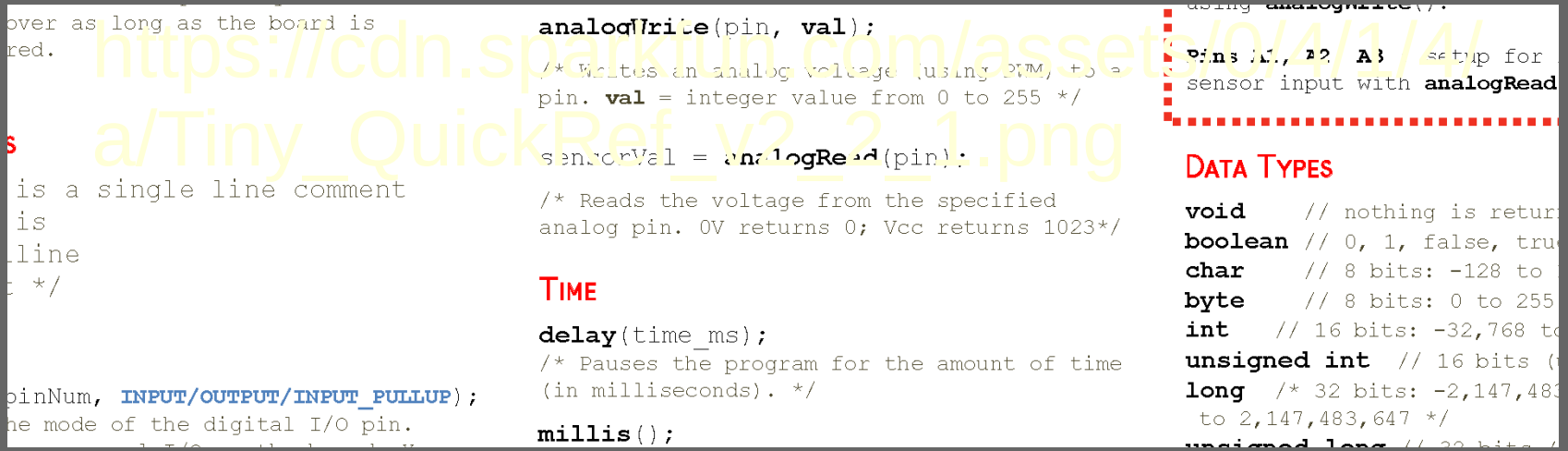
- Solange Du keinen programmierten IC hast, kann es nicht laufen!
- Zum Testen nimmst Du Dir am besten einen vorbereiteten IC.
  Danach beginnst Du mit der Programmierung.

Pause

auf zum Code

Tiny-Referenz aufrufen

https://cdn.sparkfun.com/assets/0/4/1/4/a/Tiny_QuickRef_v2_2_1.png



```
over as long as the board is
red.

s
 is a single line comment
 is
line
 */

pinNum, INPUT/OUTPUT/INPUT_PULLUP);
he mode of the digital I/O pin.
```

```
analogWrite(pin, val);
/* Writes an analog voltage (using PWM) to a
pin. val = integer value from 0 to 255 */

sensorVal = analogRead(pin);

/* Reads the voltage from the specified
analog pin. 0V returns 0; Vcc returns 1023*/
```

Time

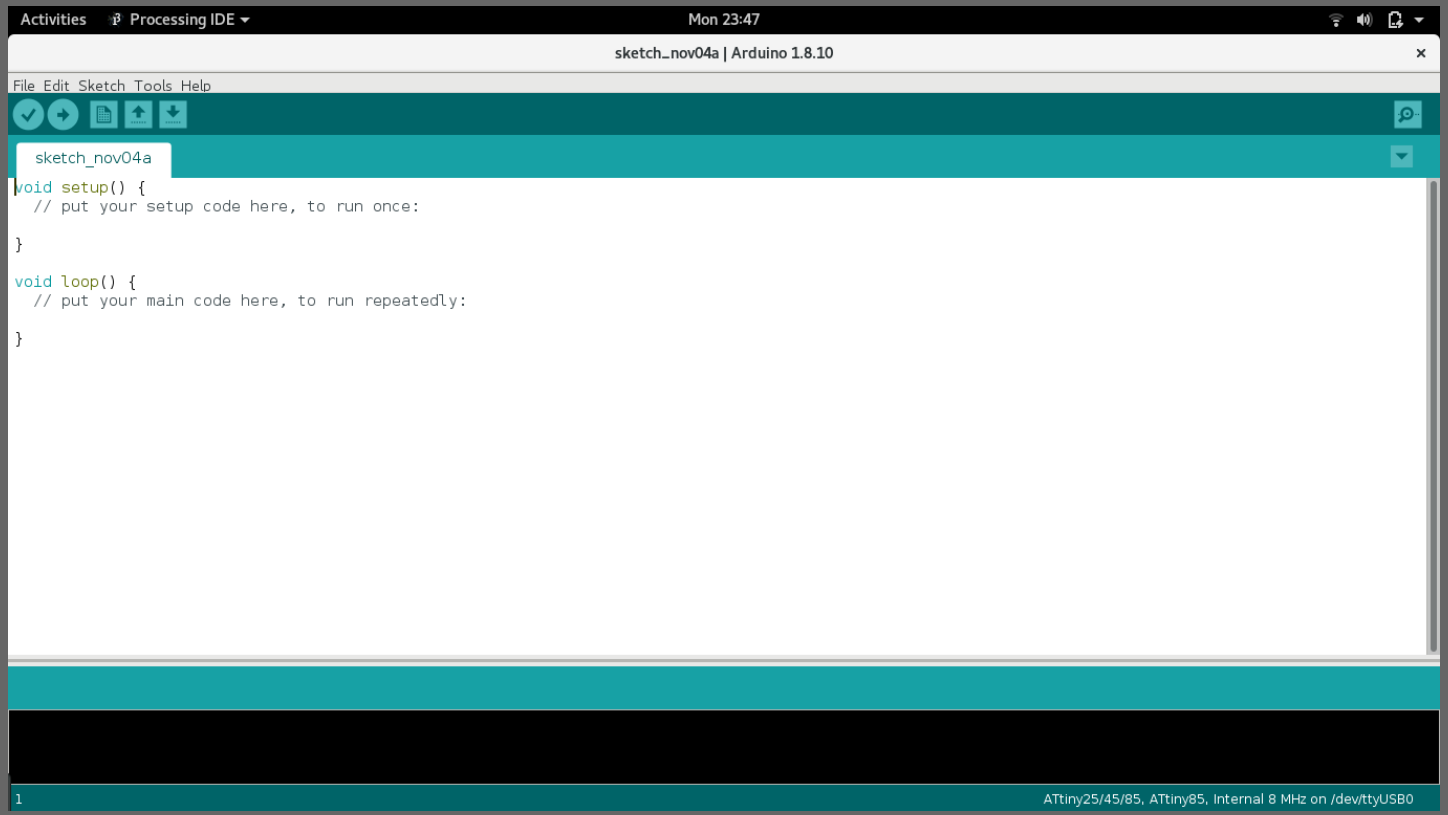
```
delay(time_ms);
/* Pauses the program for the amount of time
(in milliseconds). */

millis();
```

```
using analogWrite().
Pins A1, A2, A3 setup for
sensor input with analogRead
```

Data Types

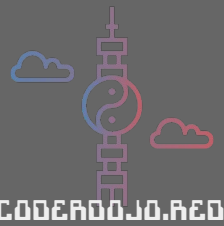
```
void     // nothing is retur
boolean  // 0, 1, false, tru
char     // 8 bits: -128 to
byte     // 8 bits: 0 to 255
int    // 16 bits: -32,768 t
unsigned int  // 16 bits (
long  /* 32 bits: -2,147,483
 to 2,147,483,647 */
```

Arduino-IDE aufrufen



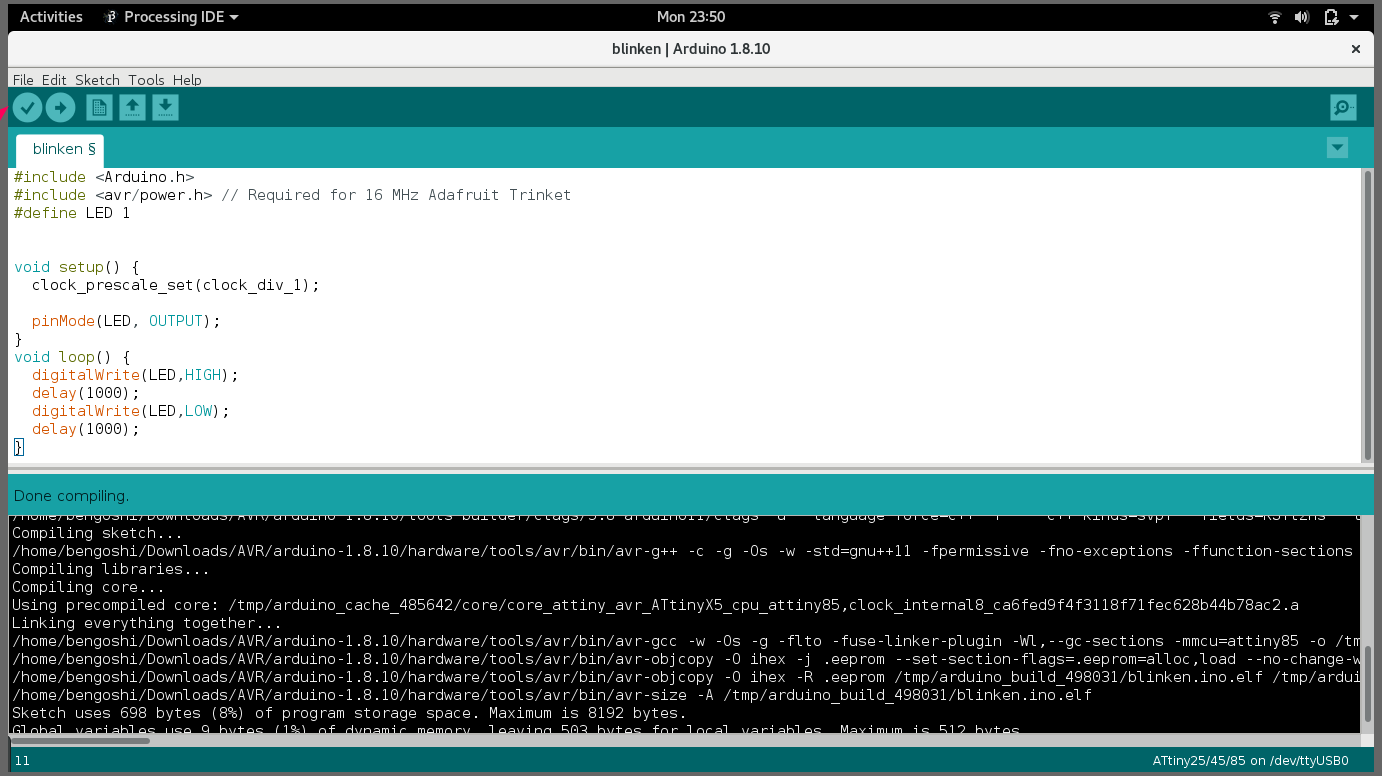
Code

```
#include <Arduino.h>
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#define LED 1
void setup() {
  clock_prescale_set(clock_div_1);
    pinMode(LED, OUTPUT);
}
void loop() {
  digitalWrite(LED,HIGH);
  delay(1000);
  digitalWrite(LED,LOW);
  delay(1000);
}
```



Code testen

hier klicken



Code testen

Activities Processing IDE Mon 23:52

blinken | Arduino 1.8.10

File Edit Sketch Tools Help

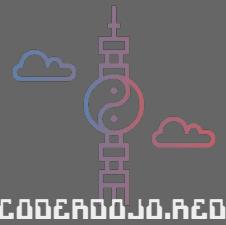
blinken

```
#include <Arduino.h>
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#define LED 1


void setup() {
  clock_prescale_set(clock_div_1);

  pinMode(LED, OUTPUT)
}
void loop() {
  digitalWrite(LED,HIGH);
  delay(1000);
  digitalWrite(LED,LOW);
  delay(1000);
}
```

expected ';' before '}' token Copy error messages

```
Generating function prototypes...
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-g++ -c -g -Os -w -std=gnu++11 -fpermissive -fno-exceptions -ffunction-sections
/home/bengoshi/Downloads/AVR/arduino-1.8.10/tools-builder/ctags/5.8-arduino11/ctags -u --language-force=c++ -f - --c++-kinds=svpf --fields=KSTtzns --l
Compiling sketch...
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-g++ -c -g -Os -w -std=gnu++11 -fpermissive -fno-exceptions -ffunction-sections
/home/bengoshi/Arduino/blinken/blinken.ino: In function 'void setup()':
blinken:10:1: error: expected ';' before '}' token
 }
 ^
exit status 1
expected ';' before '}' token
```

10 ATtiny25/45/85 on /dev/ttyUSB0



Code testen

Activities Processing IDE Mon 23:58

blinken | Arduino 1.8.10

File Edit Sketch Tools Help

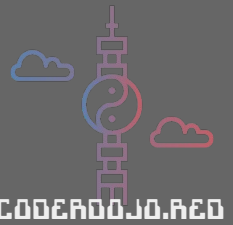
blinken

```
#include <Arduino.h>
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#define LED 1


void setup() {
  clock_prescale_set(clock_div_1);

  pinMode(LED, OUTPUT);
}
void loop() {
  digitalWrite(LED,HIGH);
  delay(1000);
  digitalWrite(LED,LOW);
  delay(1000);
}
```

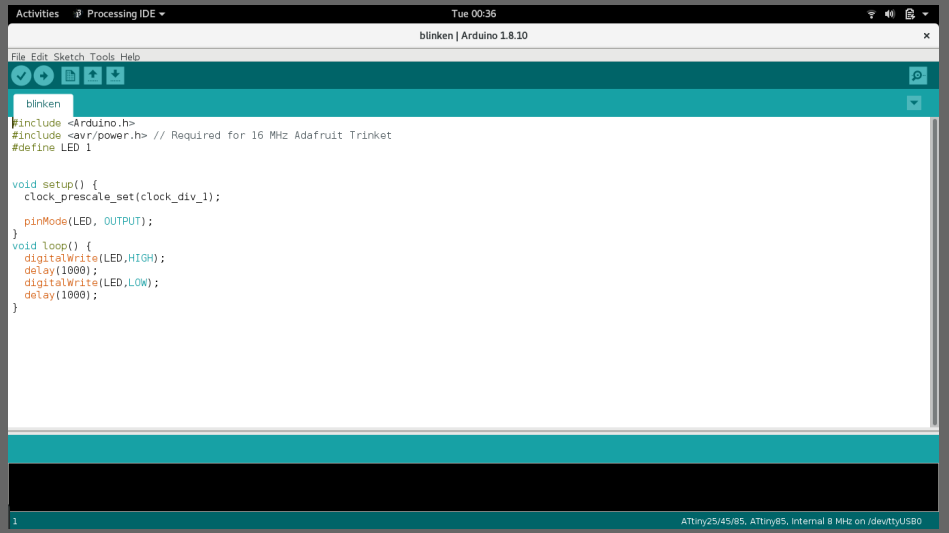
Done compiling.

```
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-g++ -c -g -Os -w -std=gnu++11 -fpermissive -fno-exceptions -ffunction-sections
Compiling libraries...
Compiling core...
Using precompiled core: /tmp/arduino_cache_485642/core/core_attiny_avr_ATtinyX5_cpu_attiny85,clock_internal8_ca6fed9f4f3118f71fec628b44b78ac2.a
Linking everything together...
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-gcc -w -Os -g -flto -fuse-linker-plugin -Wl,--gc-sections -mmcu=attiny85 -o /tm
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-objcopy -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-change-w
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-objcopy -O ihex -R .eeprom /tmp/arduino_build_498031/blinken.ino.elf /tmp/ardui
/home/bengoshi/Downloads/AVR/arduino-1.8.10/hardware/tools/avr/bin/avr-size -A /tmp/arduino_build_498031/blinken.ino.elf
Sketch uses 698 bytes (8%) of program storage space. Maximum is 8192 bytes.
Global variables use 9 bytes (1%) of dynamic memory, leaving 503 bytes for local variables. Maximum is 512 bytes.
```

9 ATtiny25/45/85 on /dev/ttyUSB0



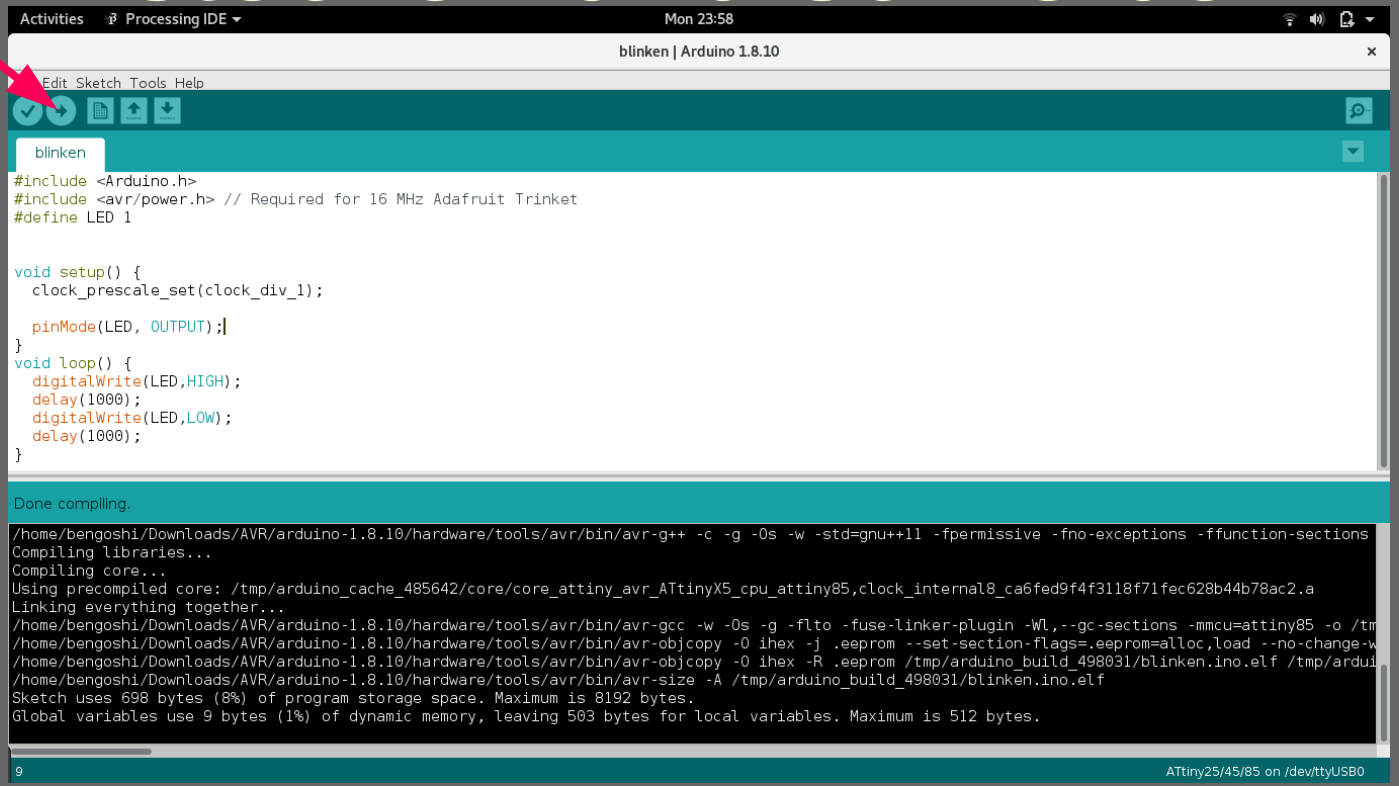
Einstellungen



- Tools →
- Board:
  ATTiny25/45/85
- Prozessor:
  ATTiny85
- Clock:
  Internal 8MHz
- Port → testen
- Programmer:

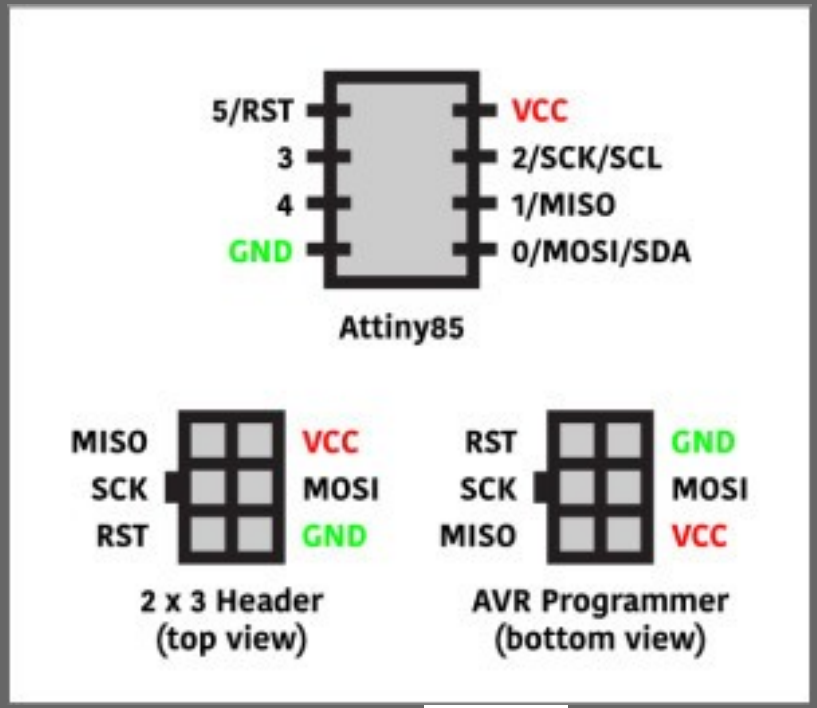
  USBasp

ATTiny in den Programmer stecken und schreiben



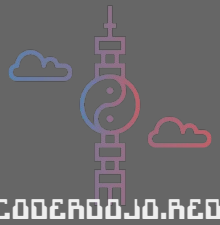
Er bedankt sich, wenn alles richtig ist!

Löt-Freak?
Hinweis für den eigenen Programmer

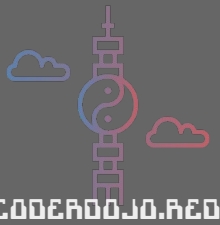


ATTiny einsetzen

Batterie einschalten



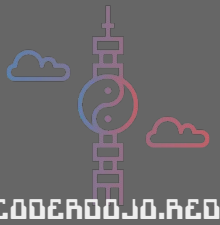
Blinkt die LED? - wenn ja, dann



Mit Code spielen

„großen“ Code implementieren

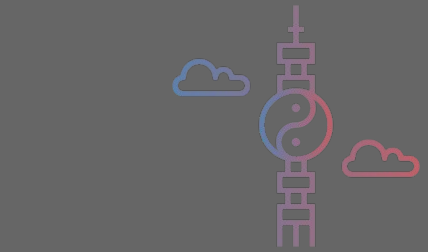
- siehe https://github.com/dermicha/tinyPixelFun/

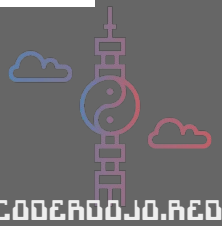


Und jetzt?

- wöchentliches Online-CoderDojo
  → coderdojo.red

- Jugend hackt
  - offenes Lab oder Workshops einmal im Monat dienstags
  - Workshops einmal im Monat samstags






HACK + MAKESPACE
XHAIN