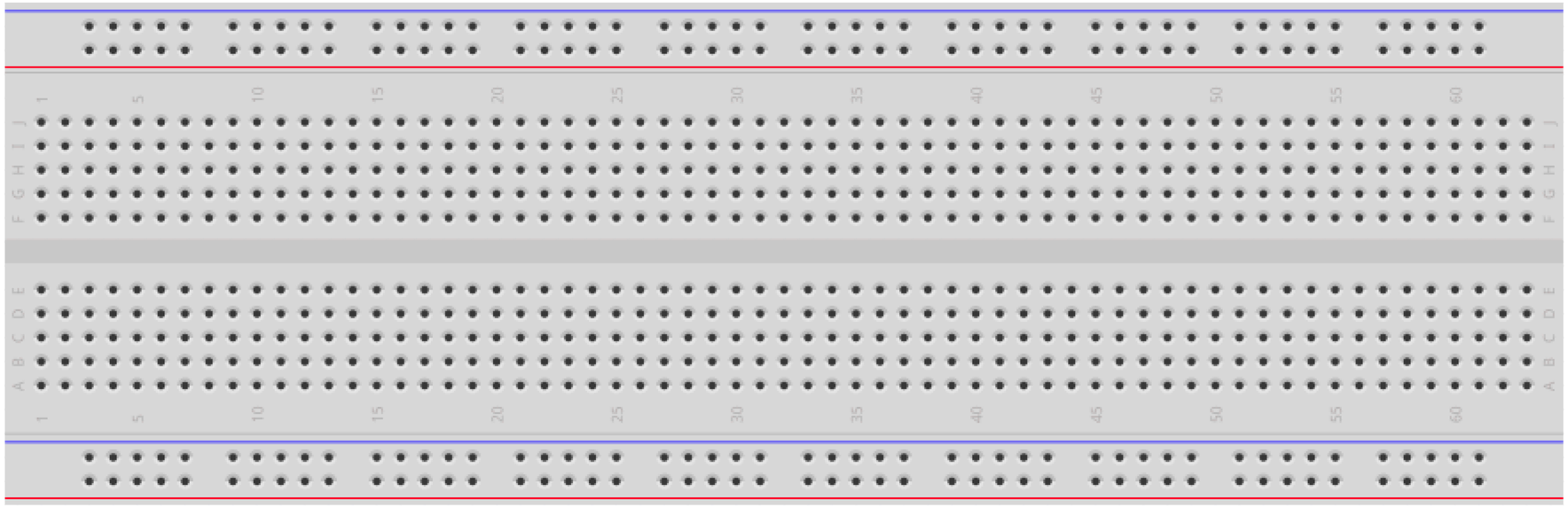
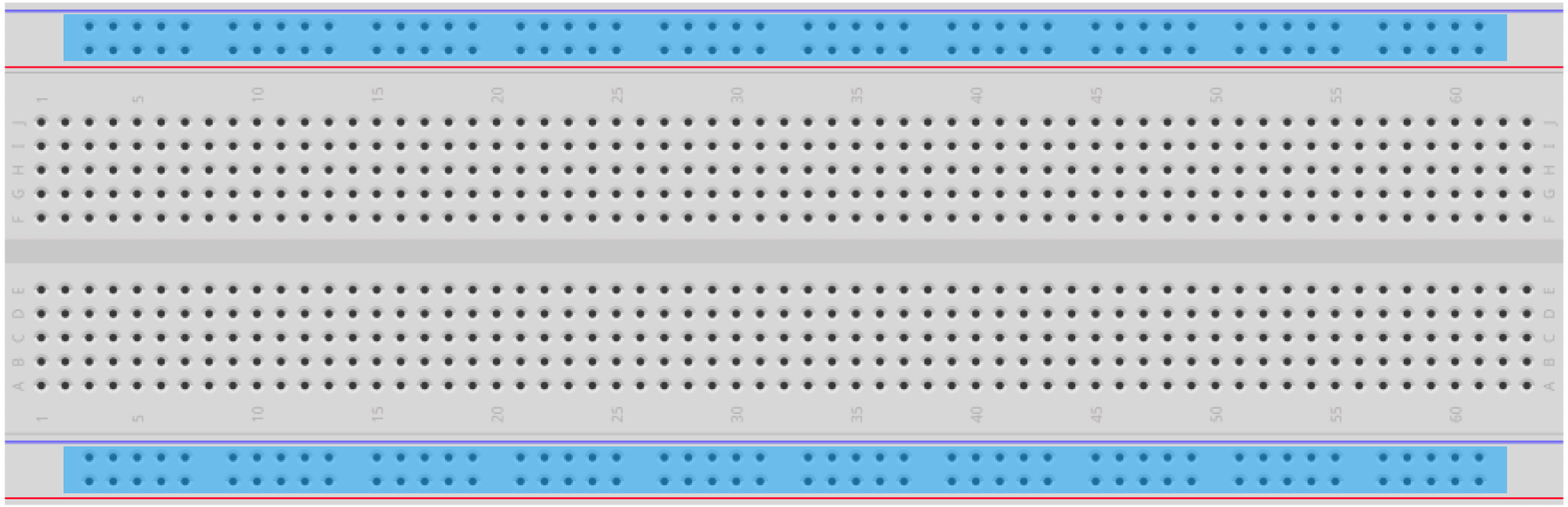
Physical Computing

WiSe 2019/20 – Grundlagen

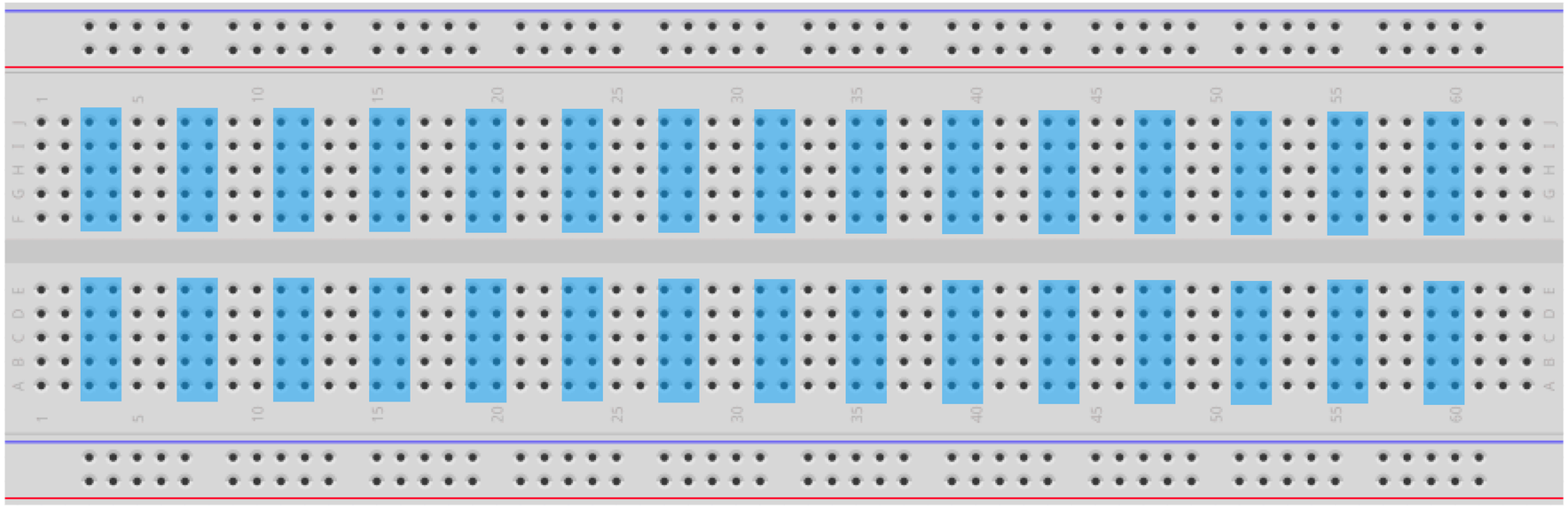
Physical Computing



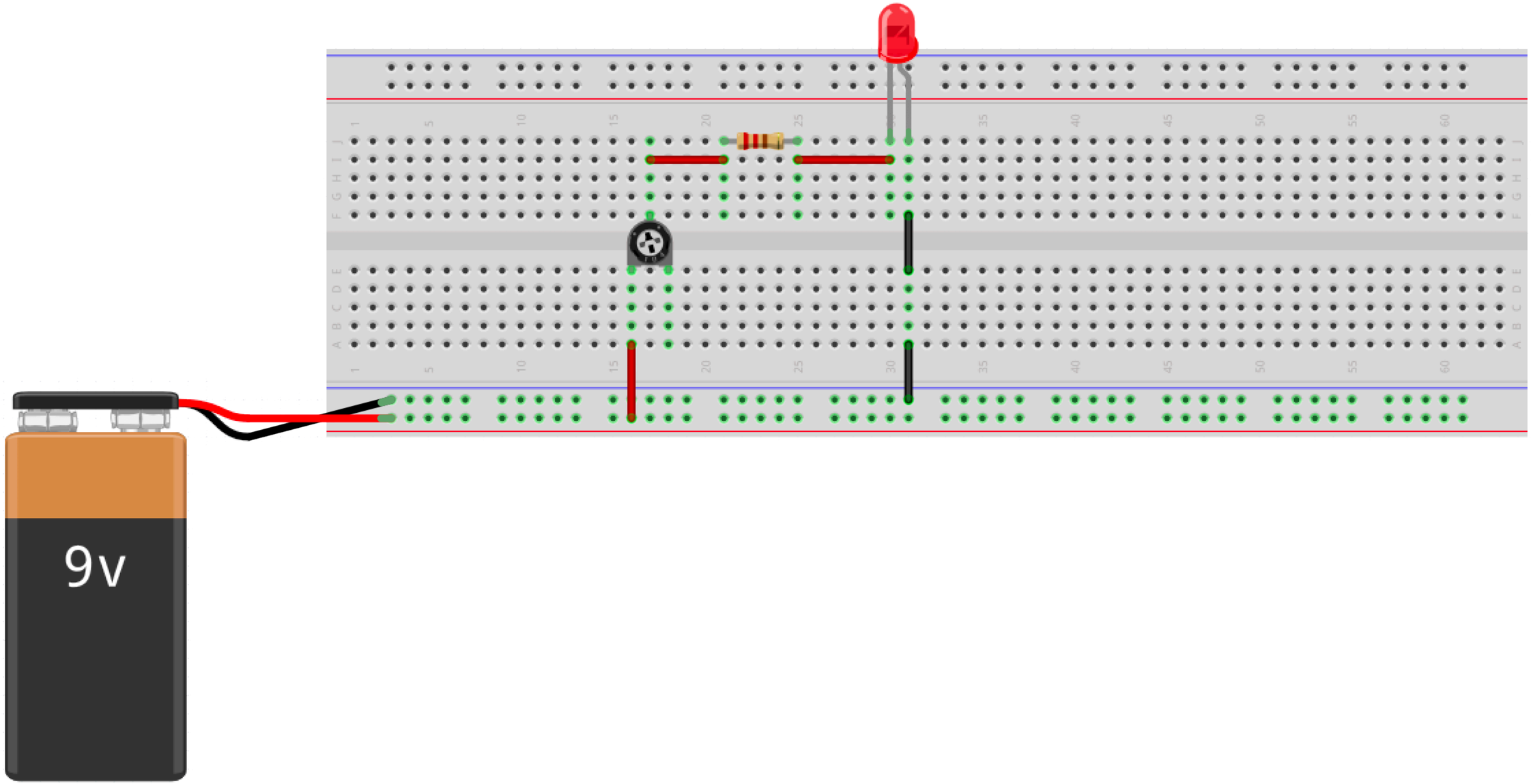
Breadboard



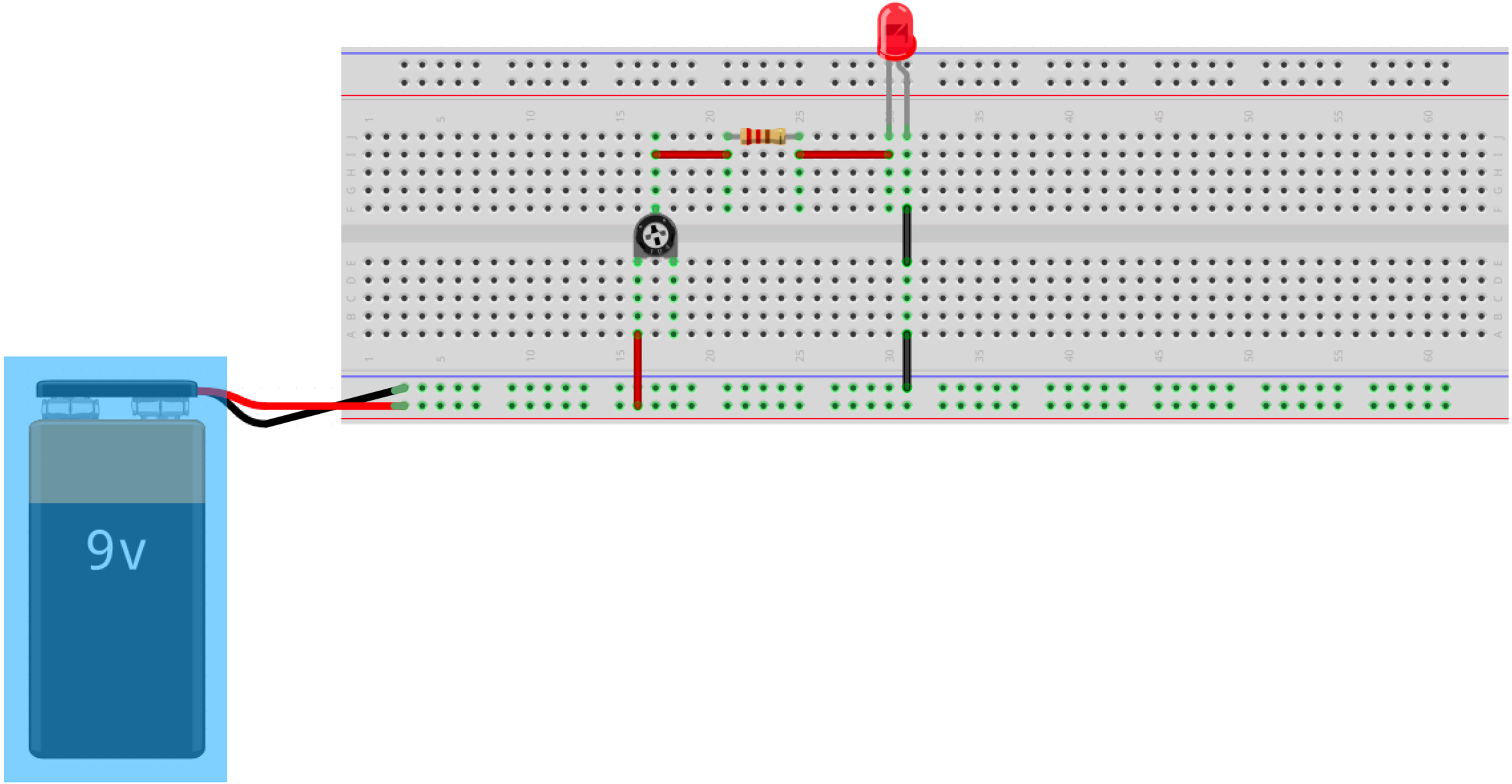
Breadboard



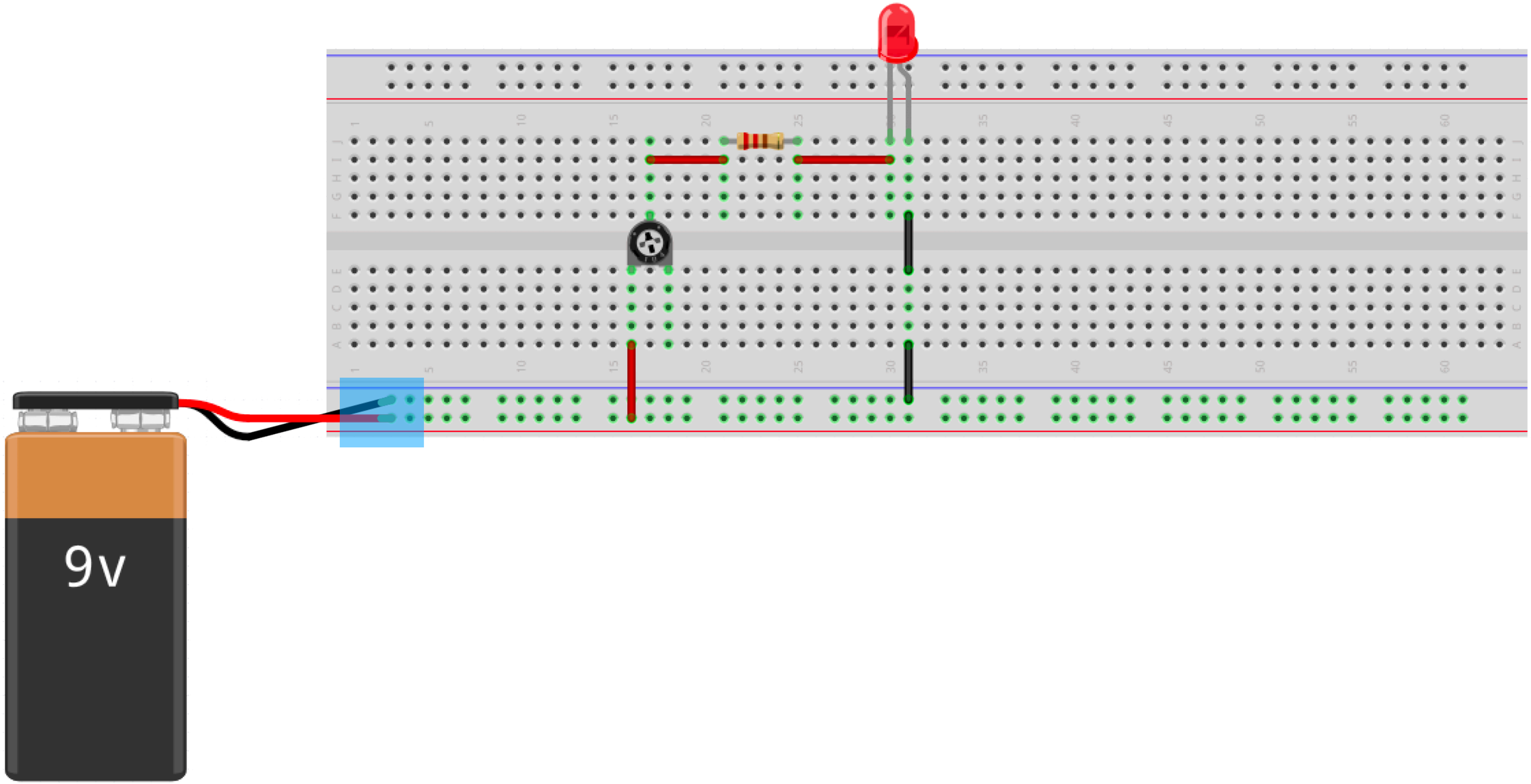
Breadboard



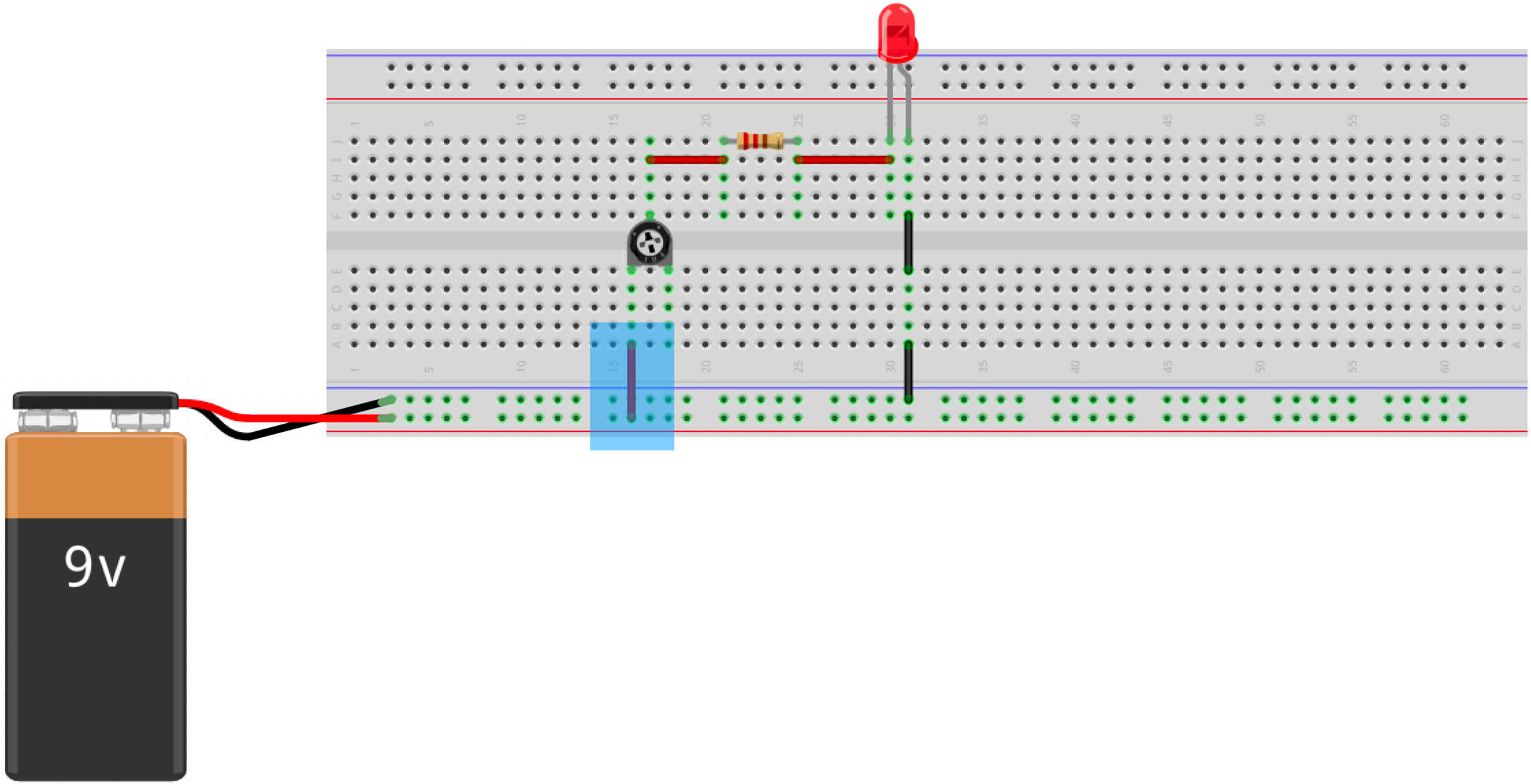
Aufgabe vom 25.10.19



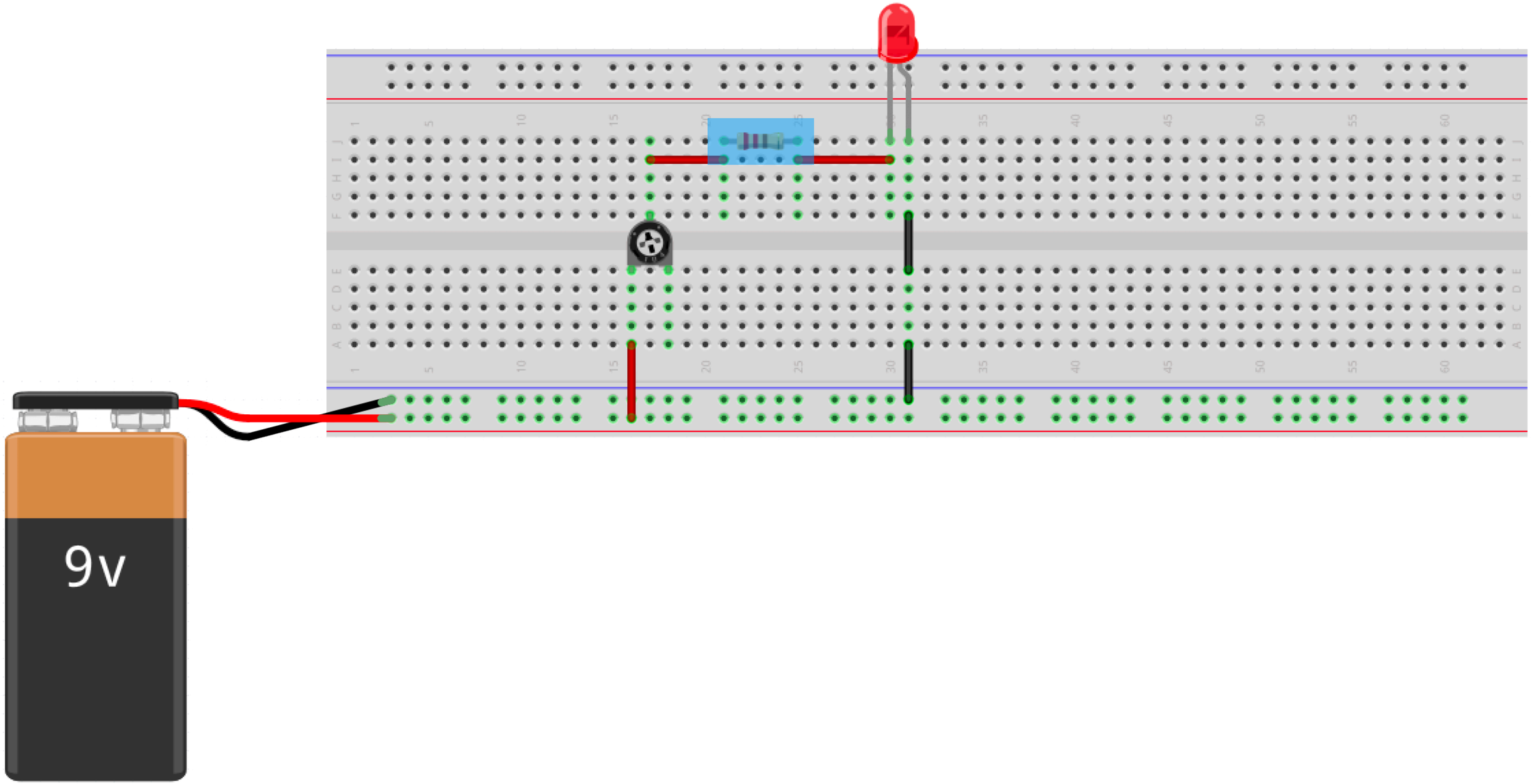
Aufgabe vom 25.10.19



Aufgabe vom 25.10.19



Aufgabe vom 25.10.19



Aufgabe vom 25.10.19

$$R = U / I$$

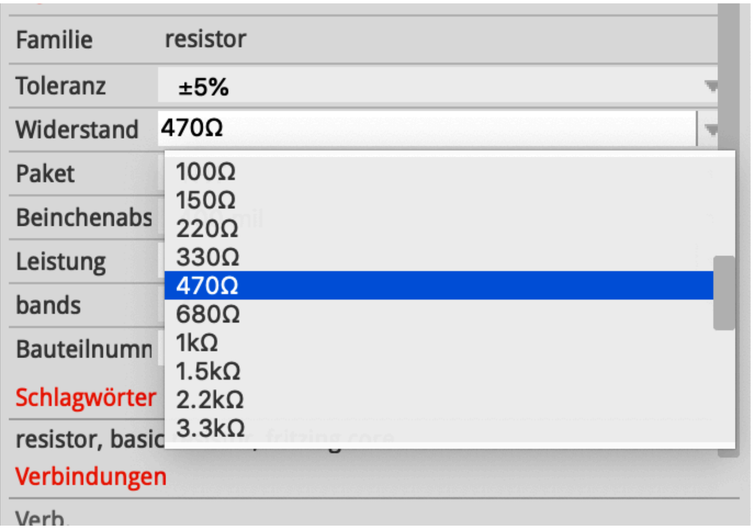
Widerstand

$$R = 9v / 0{,}020A$$

Widerstand

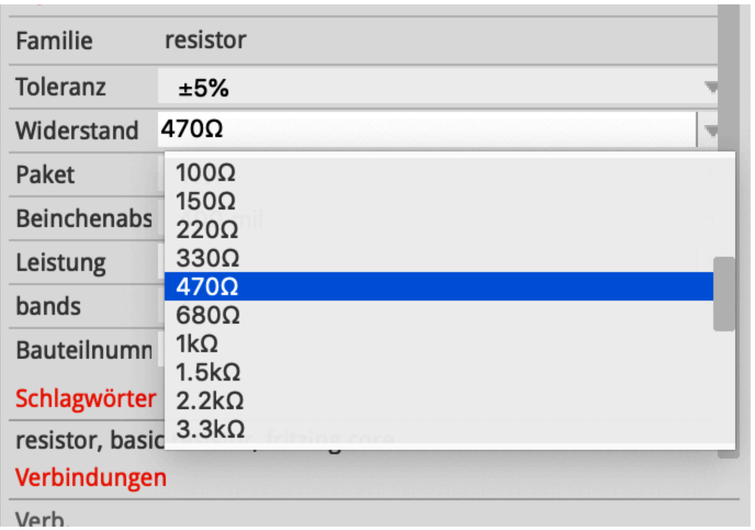
$$450\,\Omega = 9v / 0{,}020A$$

Widerstand



$$450\ \Omega = 9v\ /\ 0{,}020A$$

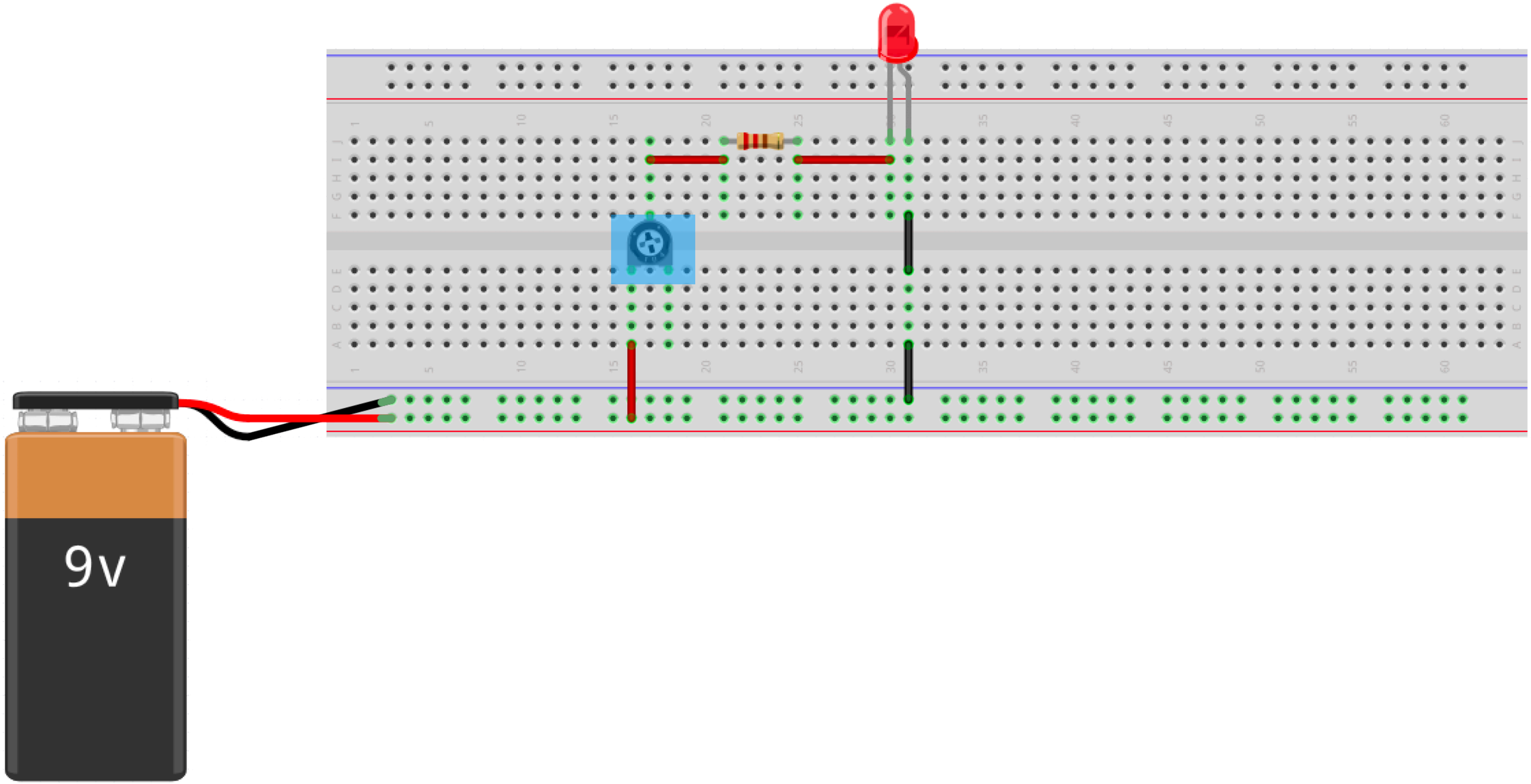
Widerstand



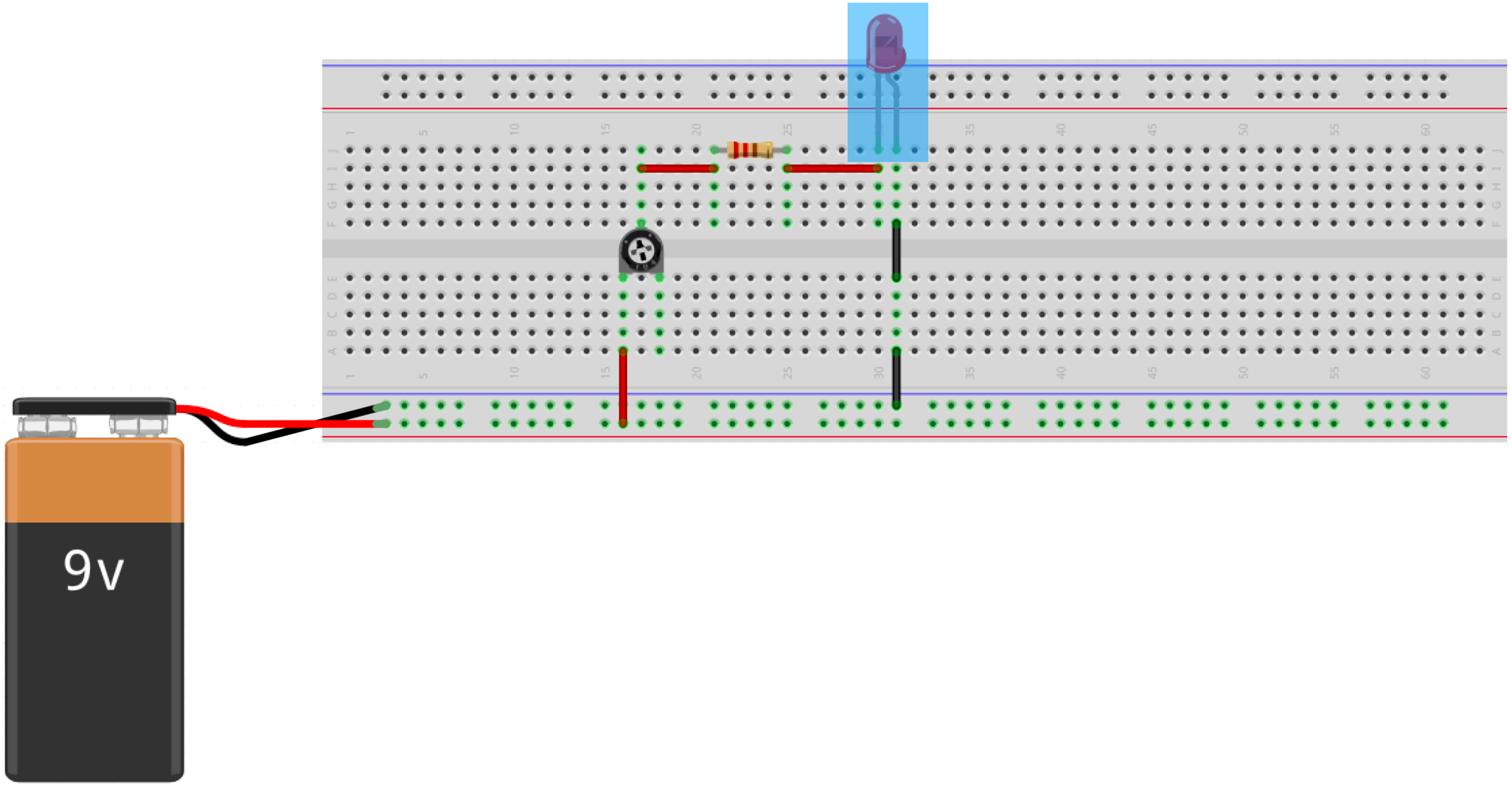
~~450 Ω~~

470 Ω = 9v / 0,020A

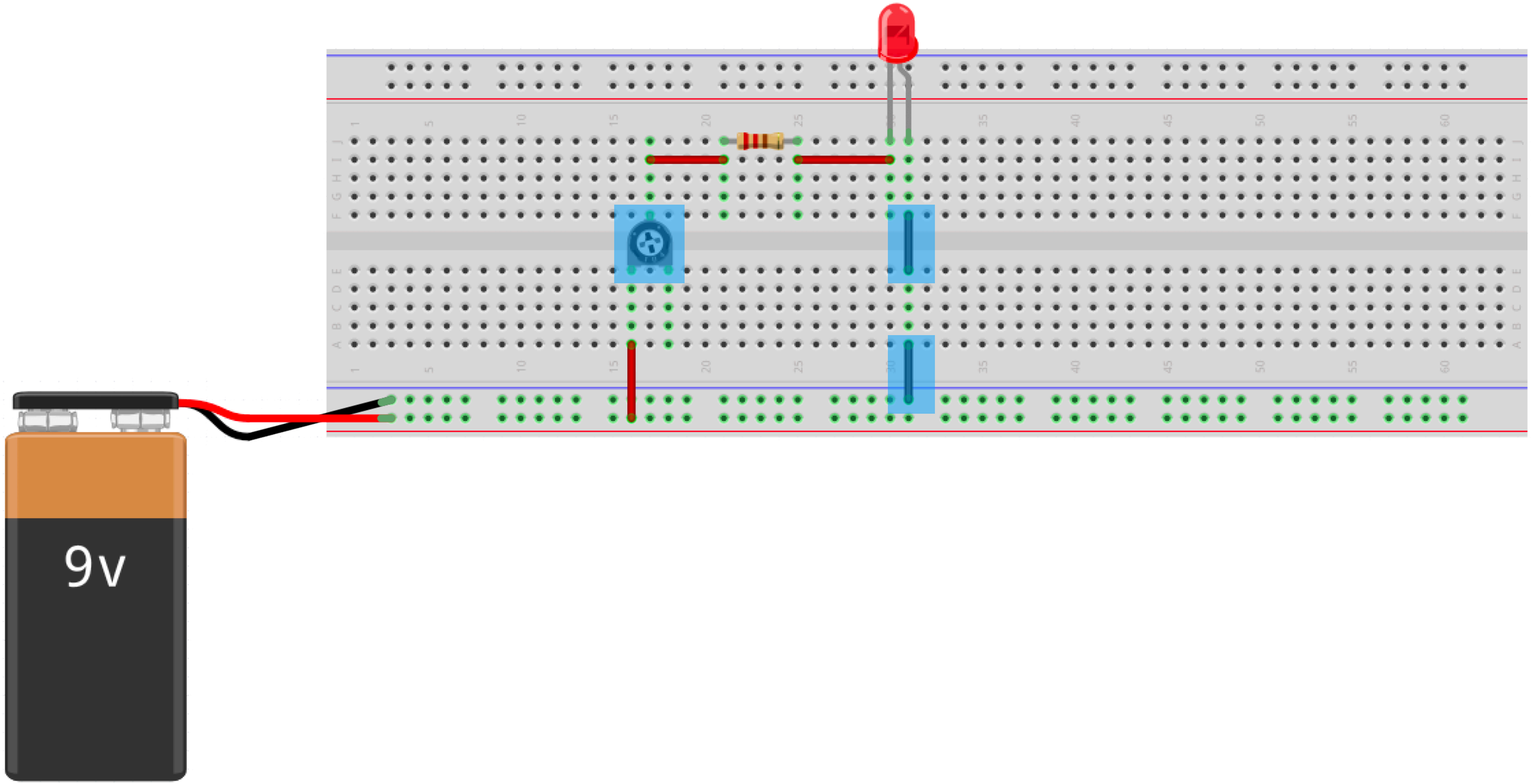
Widerstand



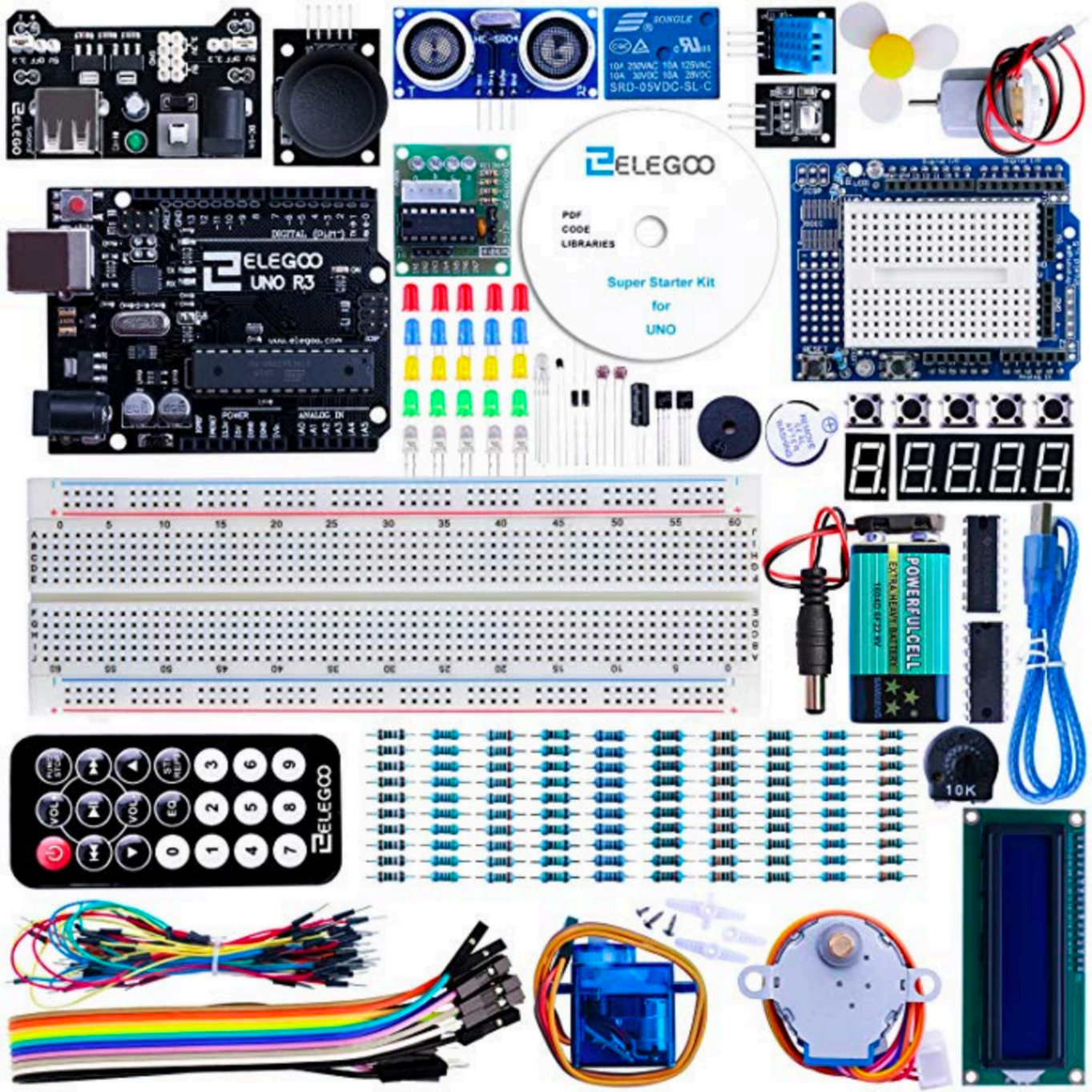
Aufgabe vom 25.10.19



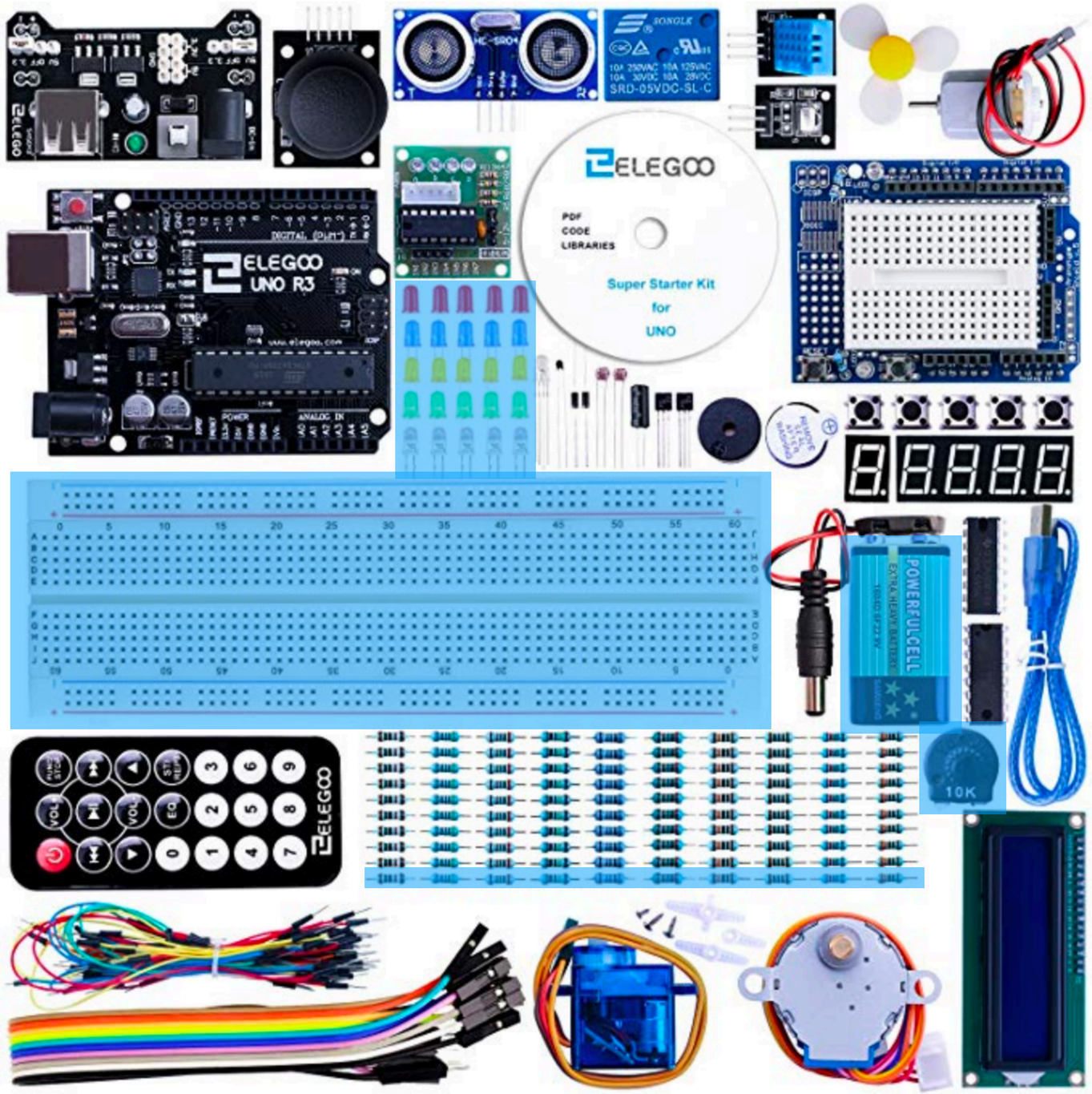
Aufgabe vom 25.10.19



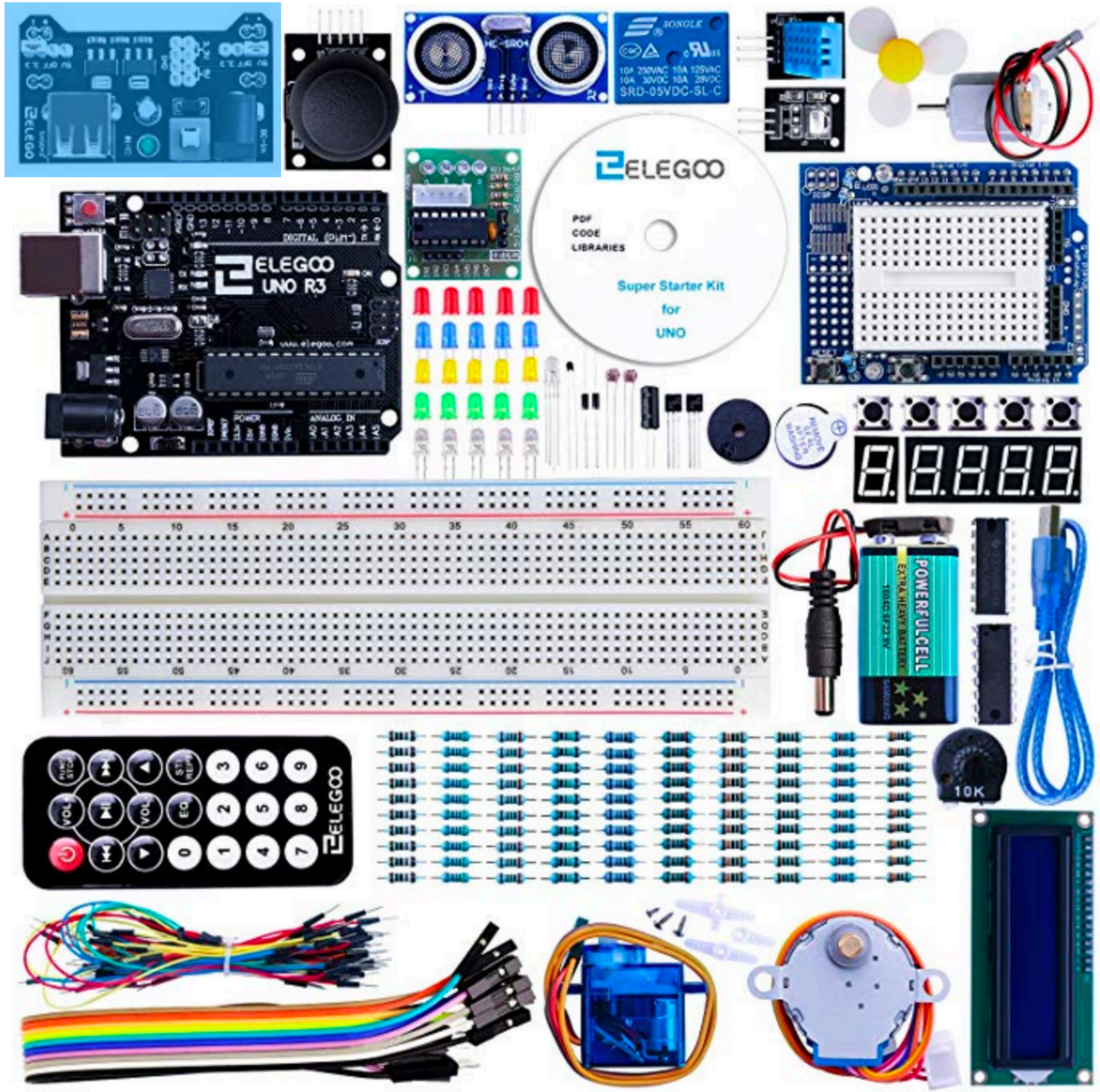
Aufgabe vom 25.10.19


ELEGOO
UNO R3
PDF
CODE
LIBRARIES
Super Starter Kit
for
UNO
HC-SR04
SRD-05VDC-SL-C
POWERFULCELL
10K

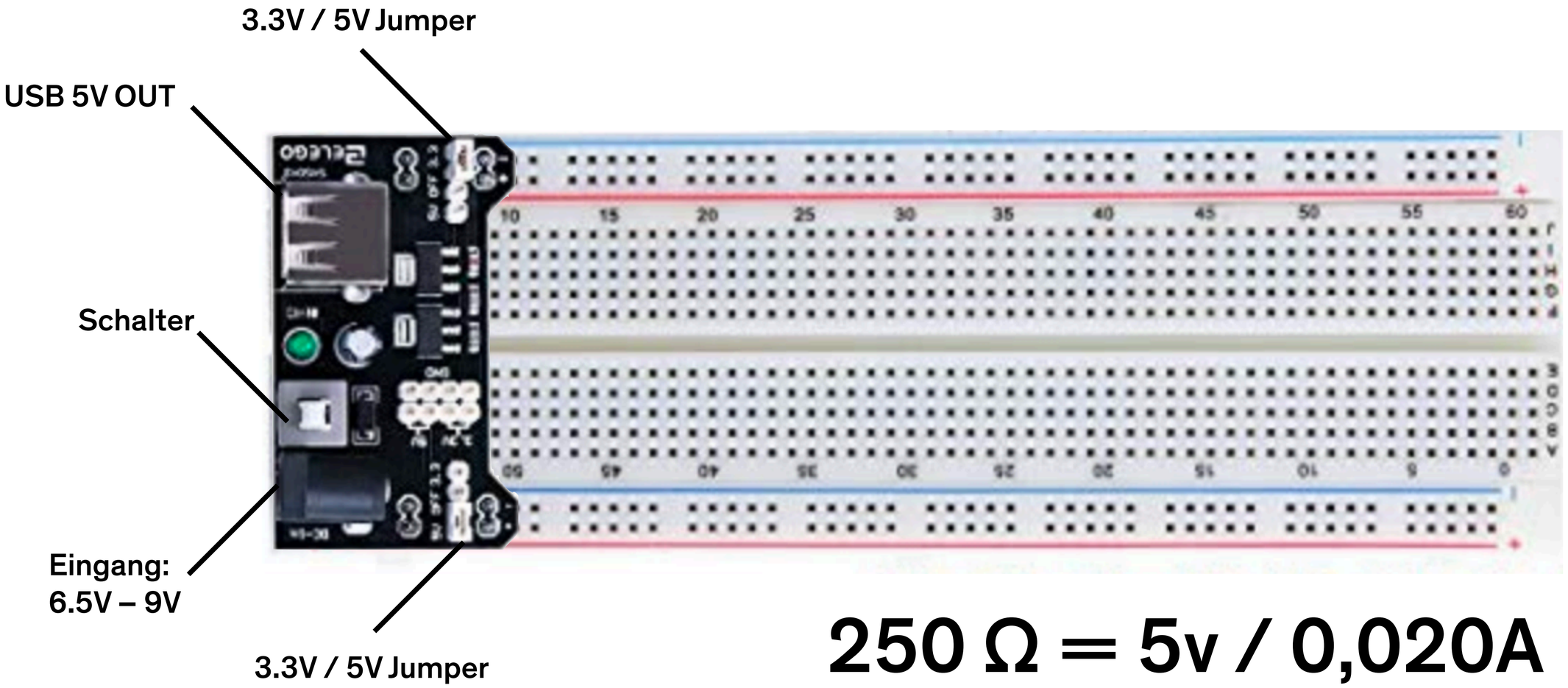
Set


ELEGOO
UNO R3
HC-SR04
SRD-05VDC-SL-C
PDF
CODE
LIBRARIES
Super Starter Kit
for
UNO
POWERFULCELL
10K

Set

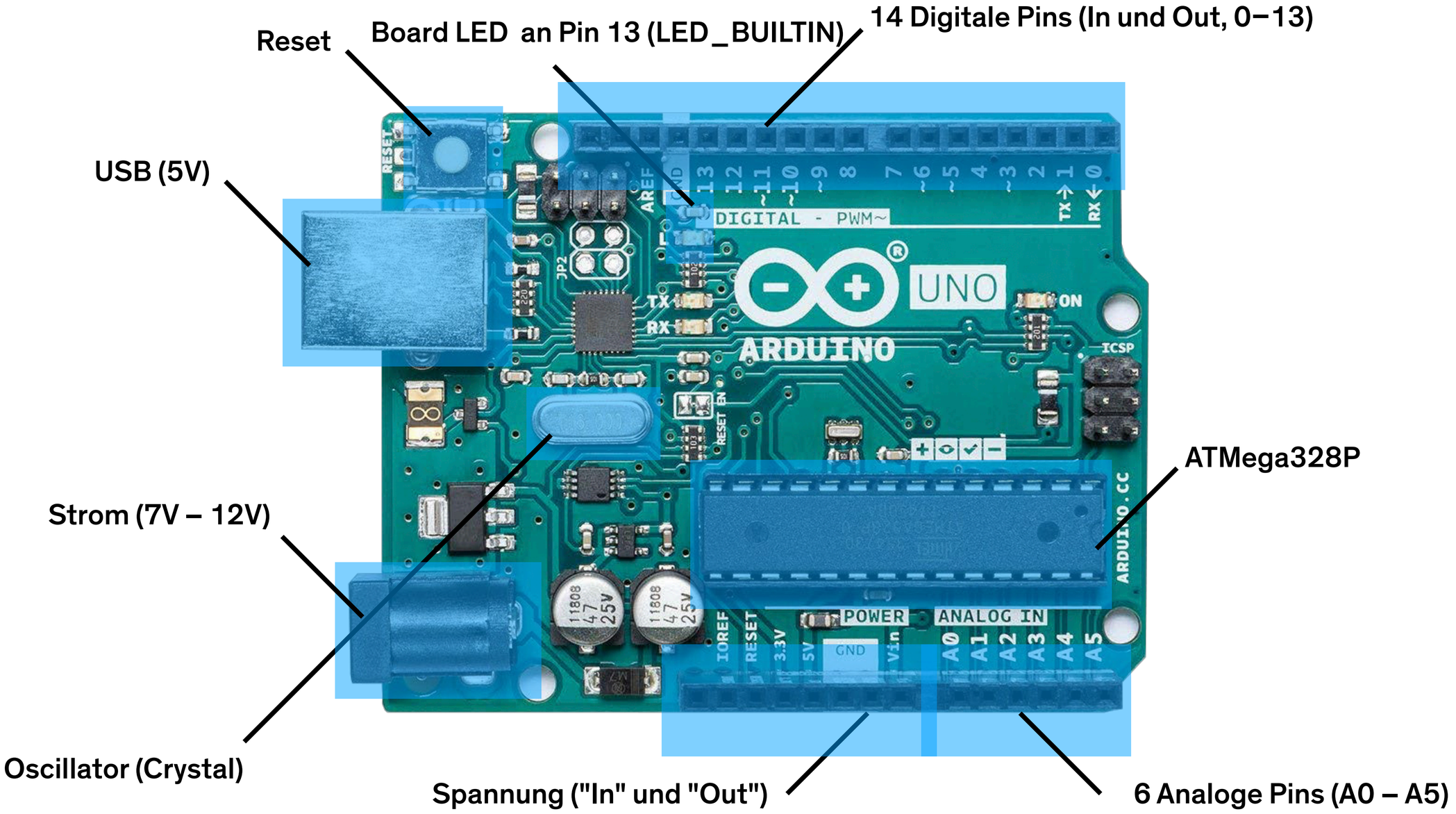

ELEGOO
UNO R3
PDF
CODE
LIBRARIES
Super Starter Kit
for
UNO
HC-SR04
SRD-05VDC-SL-C
POWERFULCELL
10K

Set

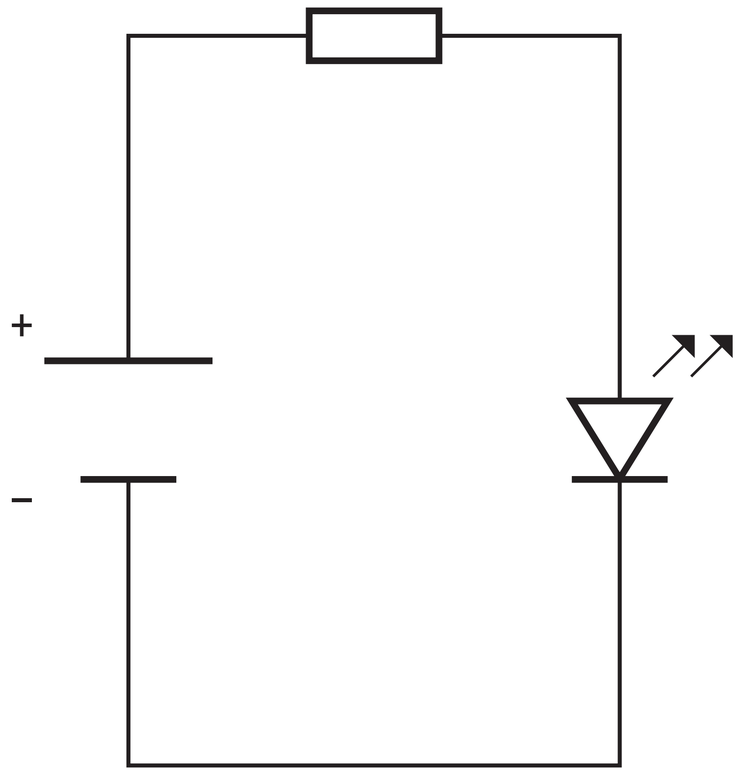


250 Ω = 5v / 0,020A
330 Ω

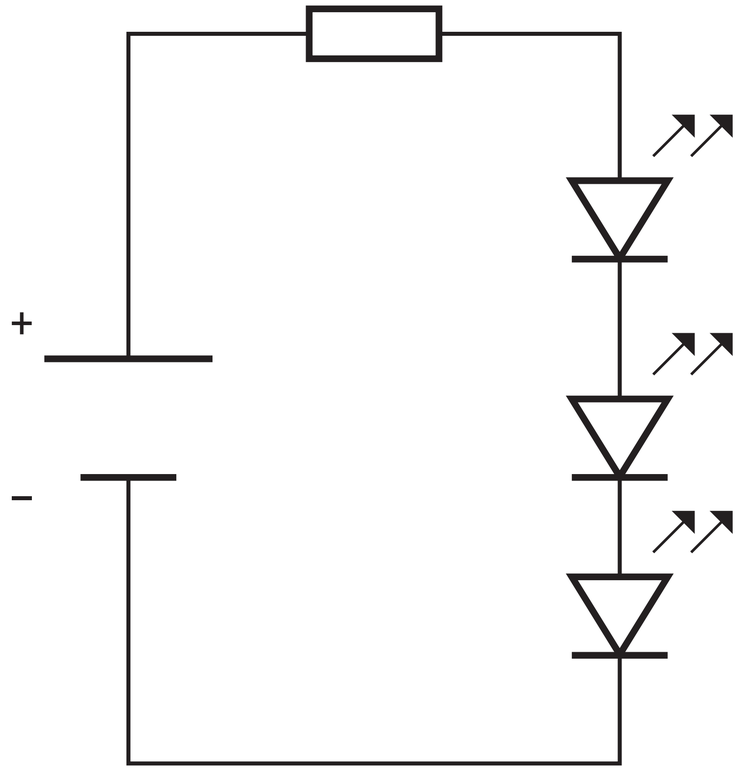
Set


Reset
Board LED an Pin 13 (LED_BUILTIN)
14 Digitale Pins (In und Out, 0–13)
USB (5V)
ATMega328P
Strom (7V – 12V)
Oscillator (Crystal)
Spannung ("In" und "Out")
6 Analoge Pins (A0 – A5)
DIGITAL - PWM~
UNO
ARDUINO
POWER
ANALOG IN

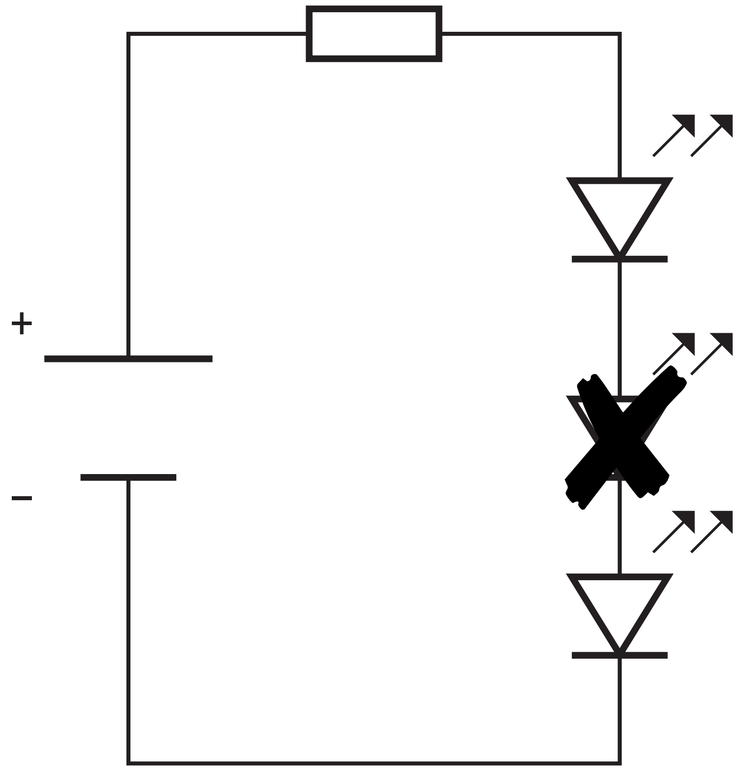
Arduino



Arduino

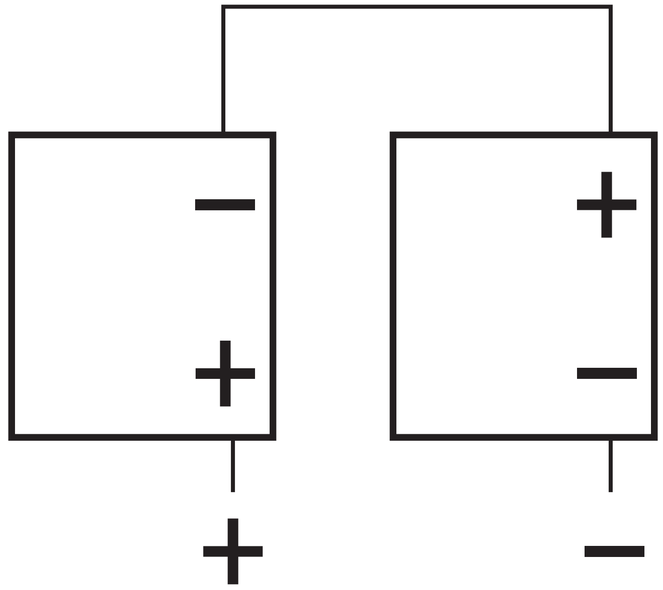


Arduino

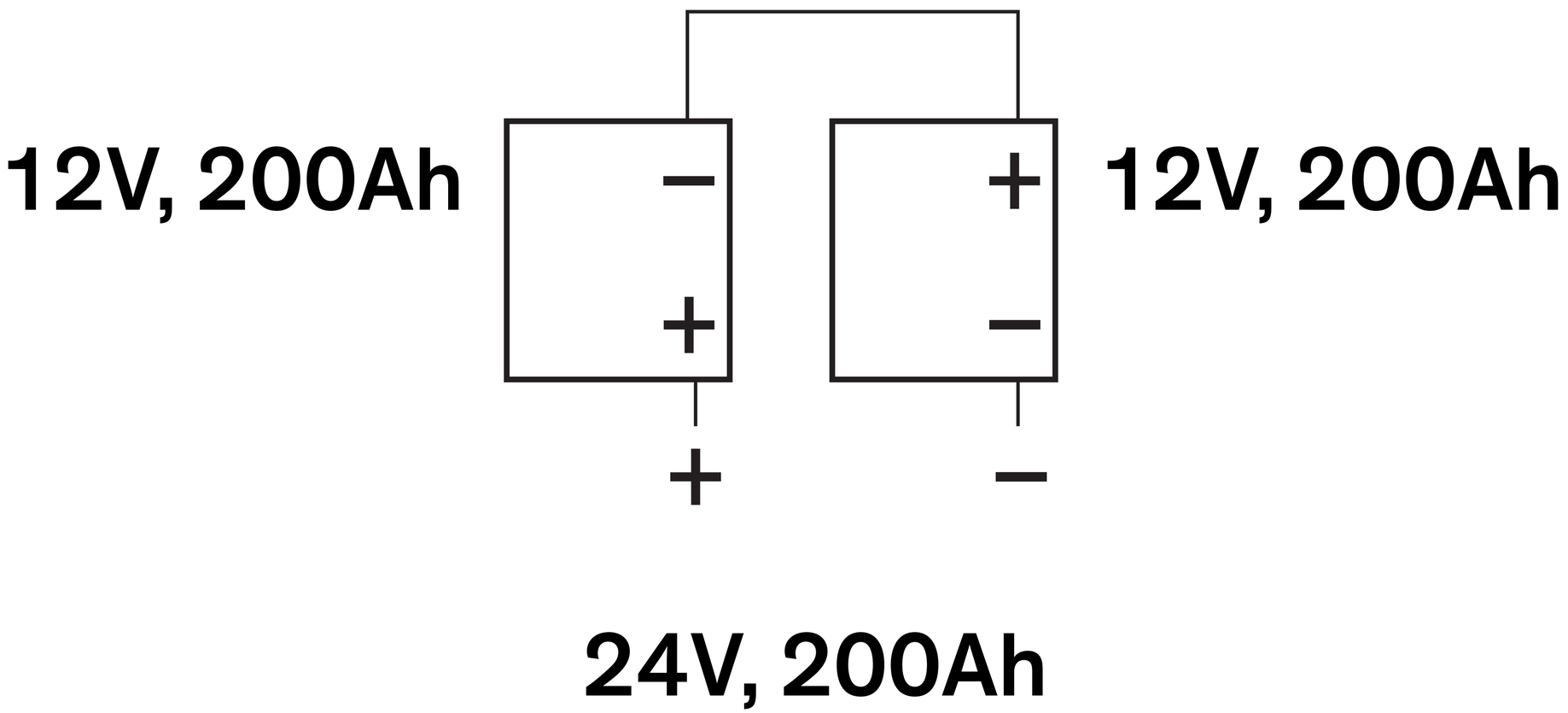


Arduino

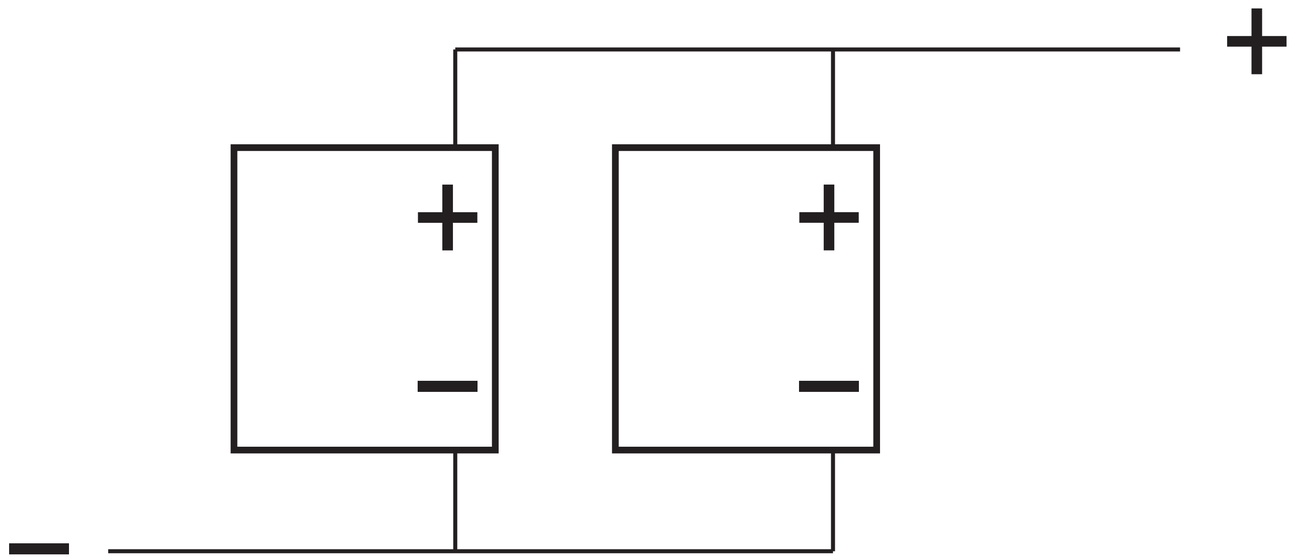
Reihen - und Parallelschaltung



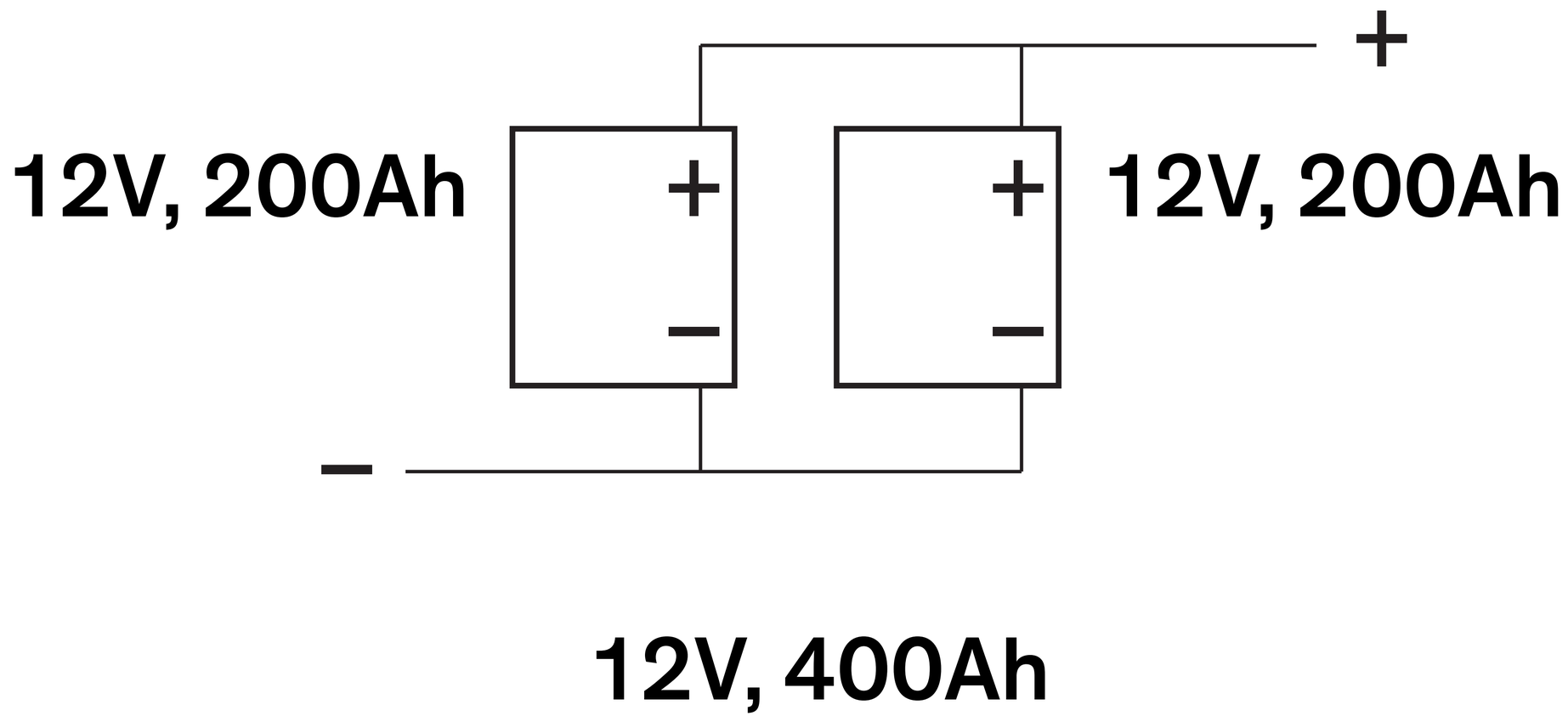
Reihenschaltung



Reihenschaltung

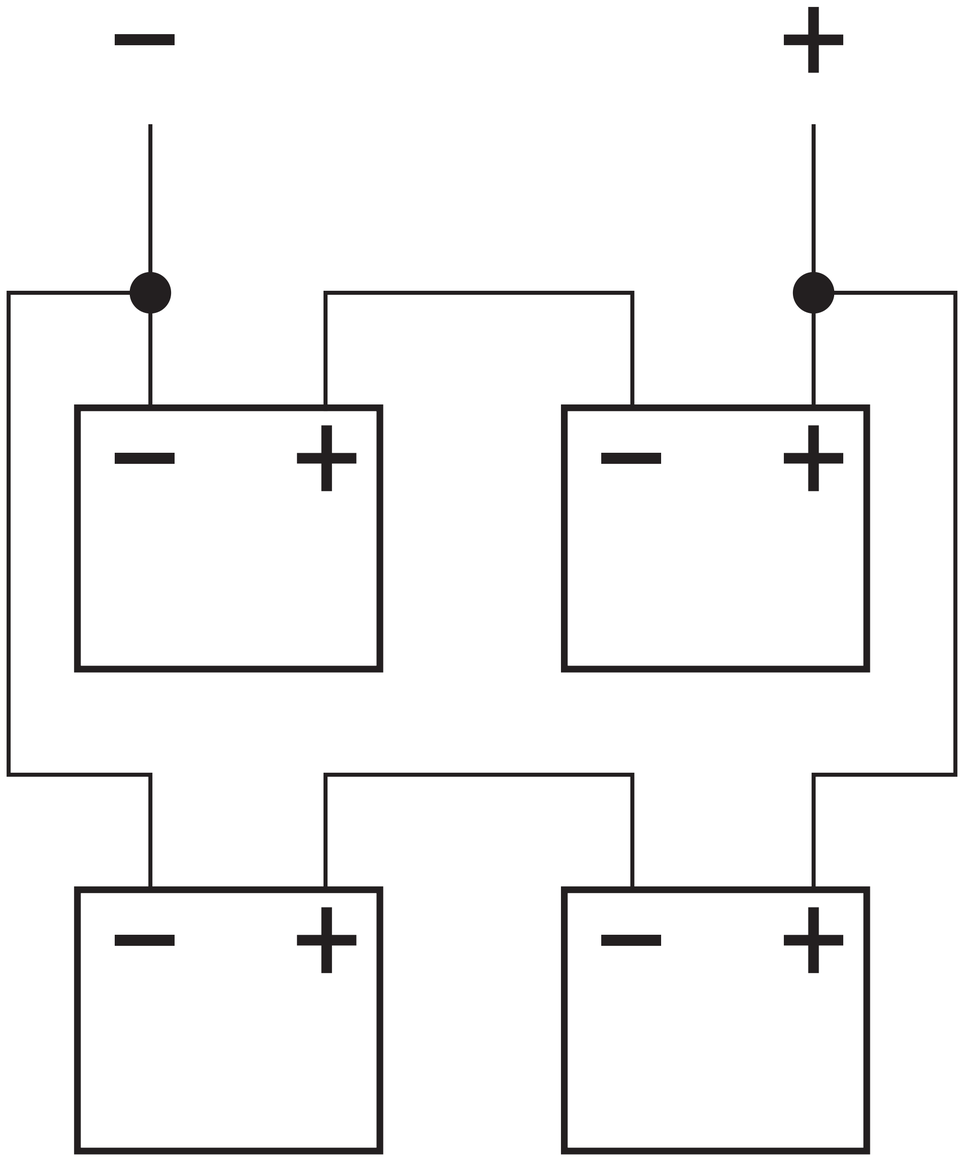


Parallelschaltung



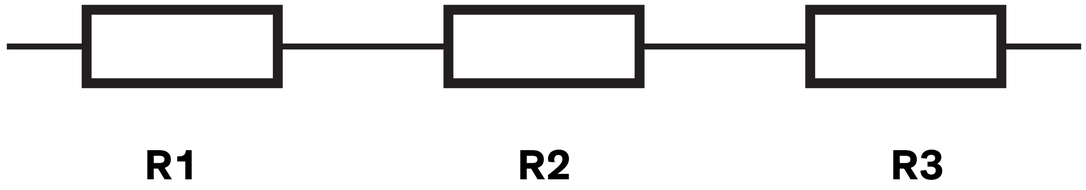
Parallelschaltung

24V, 400Ah

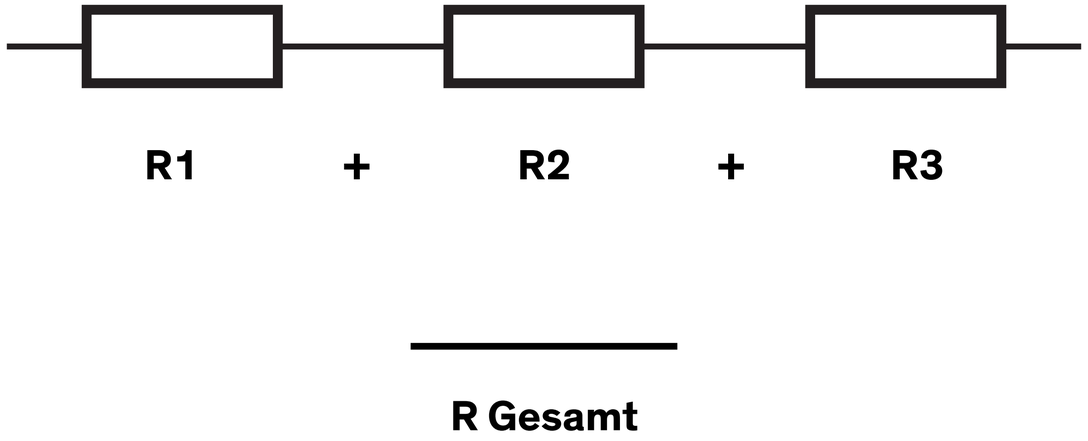


Reihen + Parallel

Widerstände in Reihe



Widerstände in Reihe

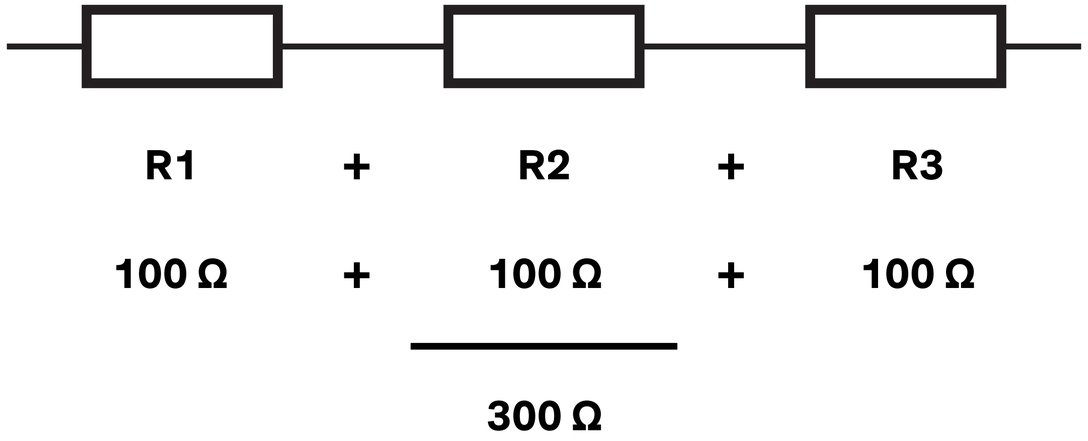


Widerstände in Reihe

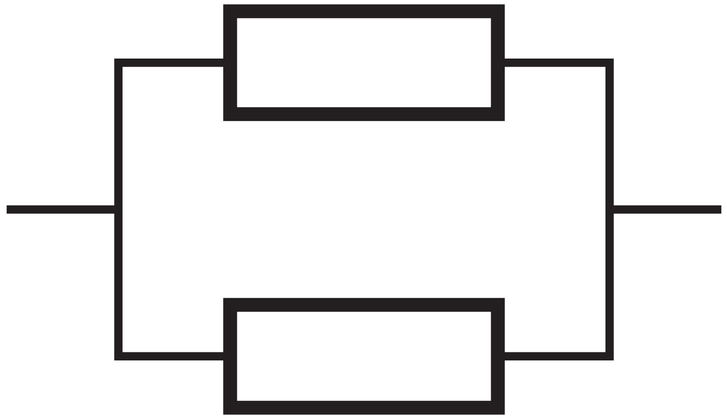
R1 + R2 + R3

100 Ω + 100 Ω + 100 Ω

Widerstände in Reihe

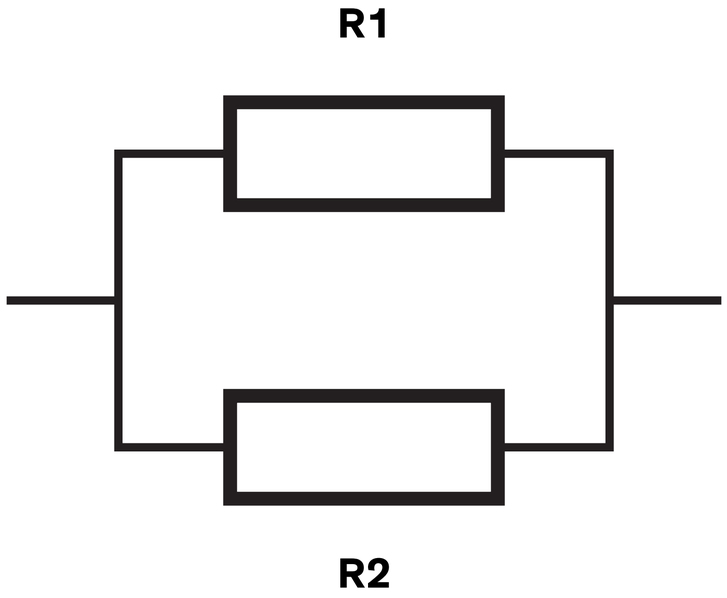


Widerstände in Reihe



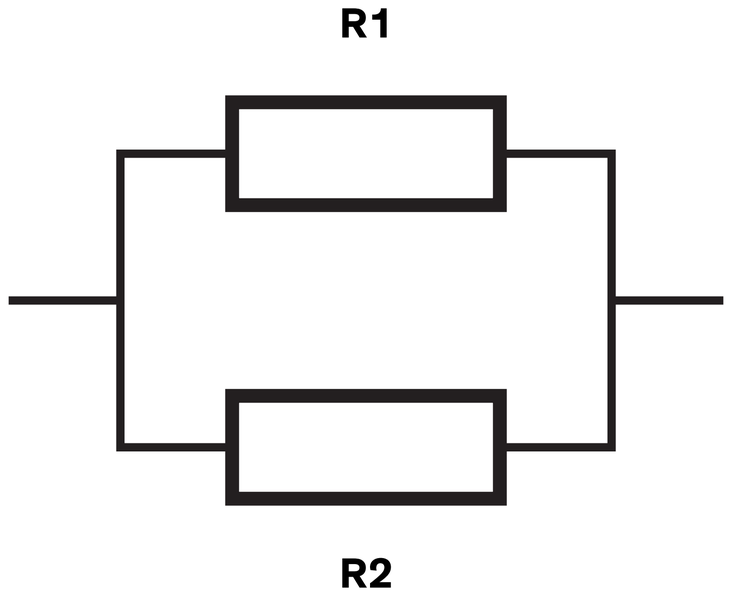
Widerstände in Parallelschaltung

$$\frac{R1 * R2}{R1 + R2} = R\ Gesamt$$



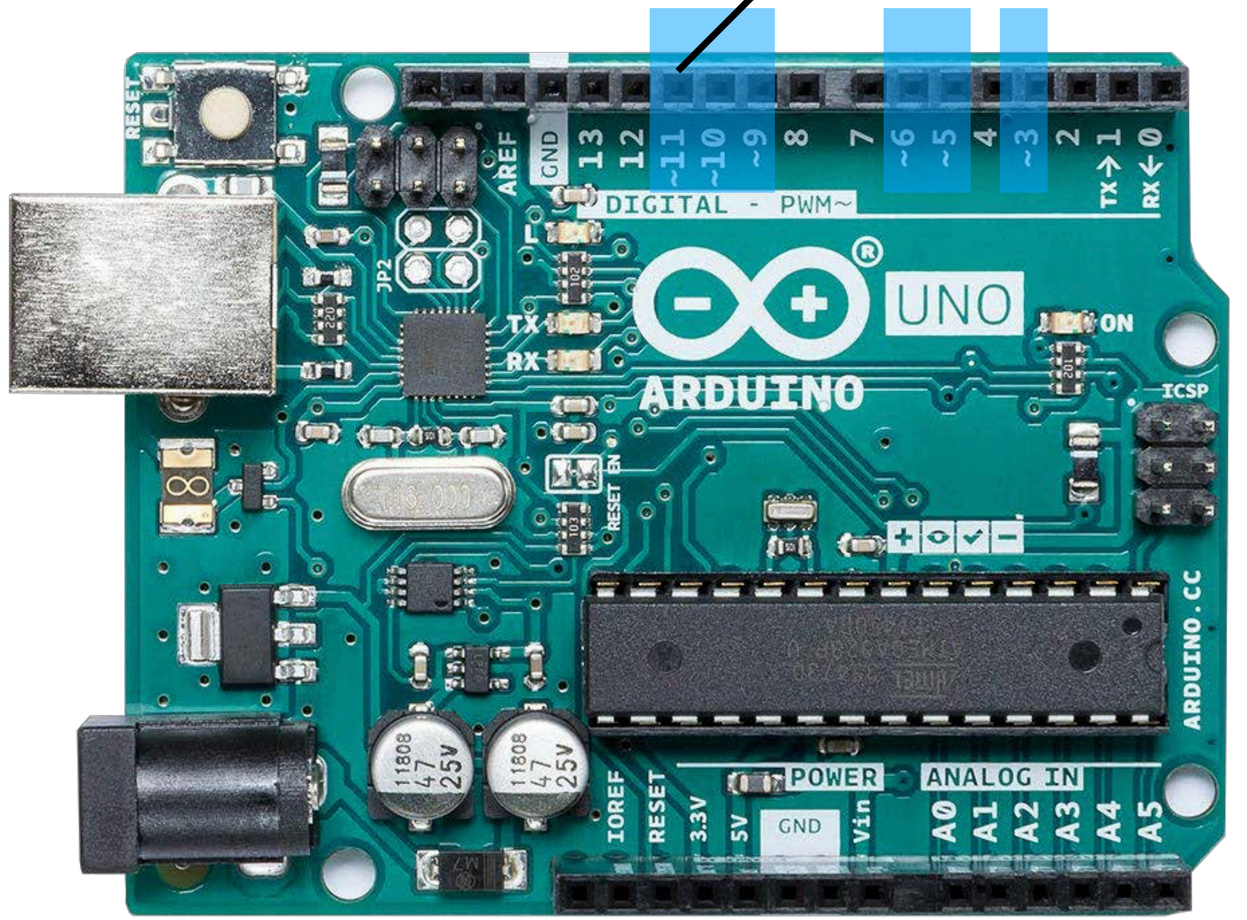
Widerstände in Parallelschaltung

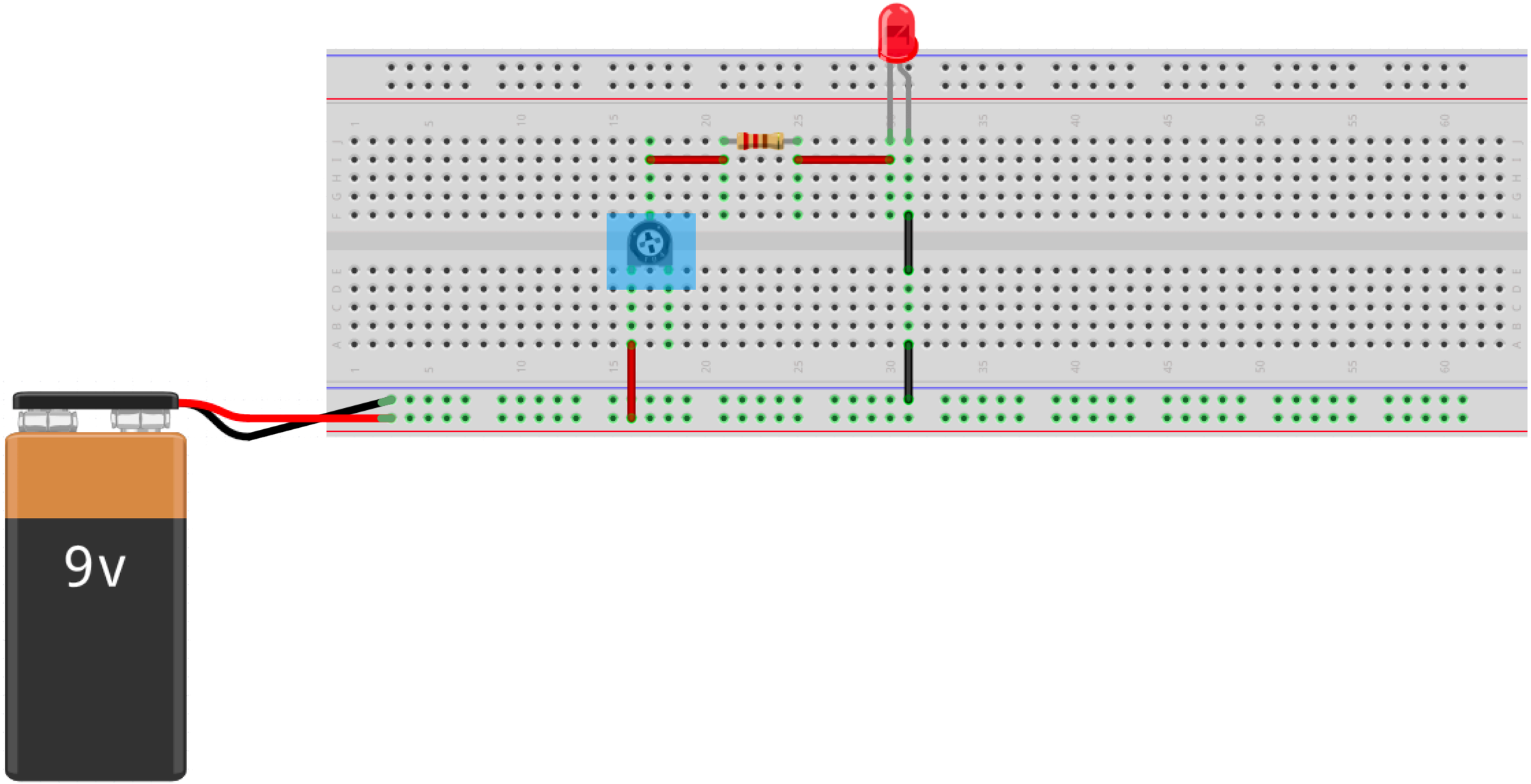
$$\frac{100\,\Omega * 100\,\Omega}{100\,\Omega + 100\,\Omega} = 50\,\Omega$$



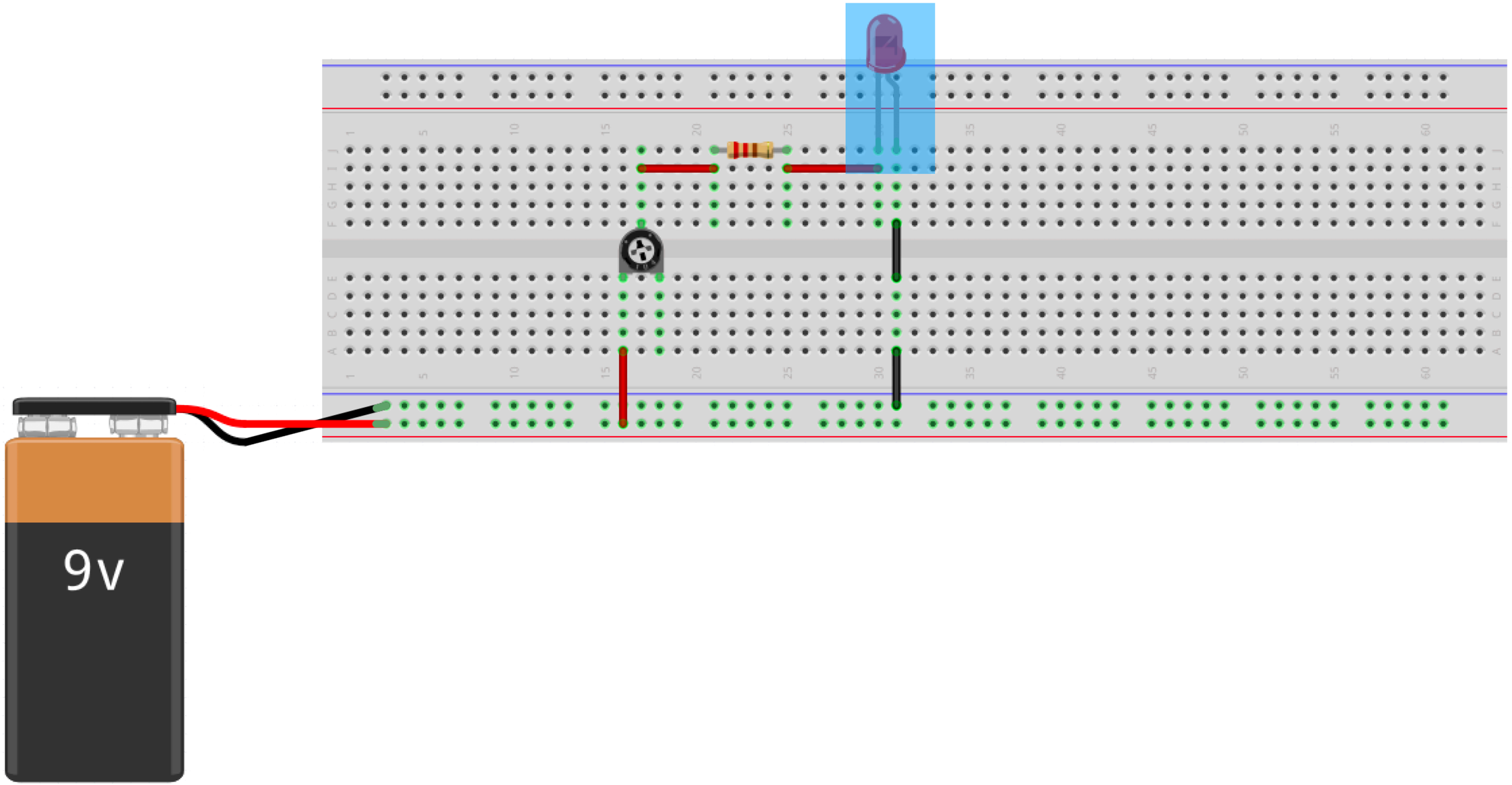
Widerstände in Parallelschaltung

PWM


PWM Pins gekennzeichnet mit ~
DIGITAL - PWM~
UNO
ARDUINO
ON
ICSP
POWER
ANALOG IN
GND
ARDUINO.CC



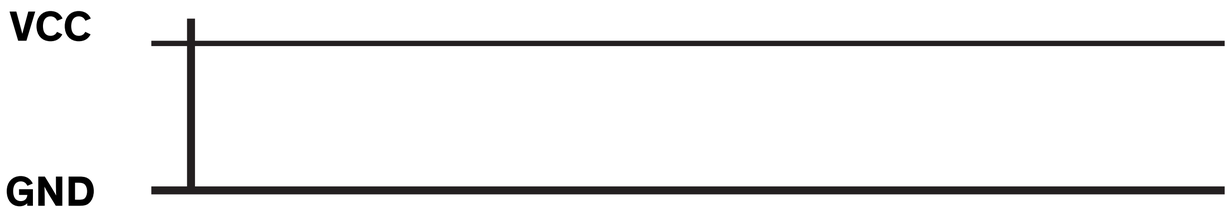
PWM



PWM

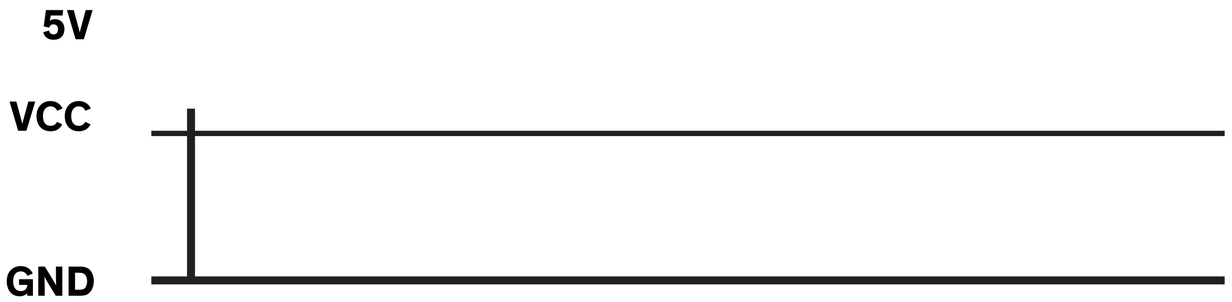
PWM

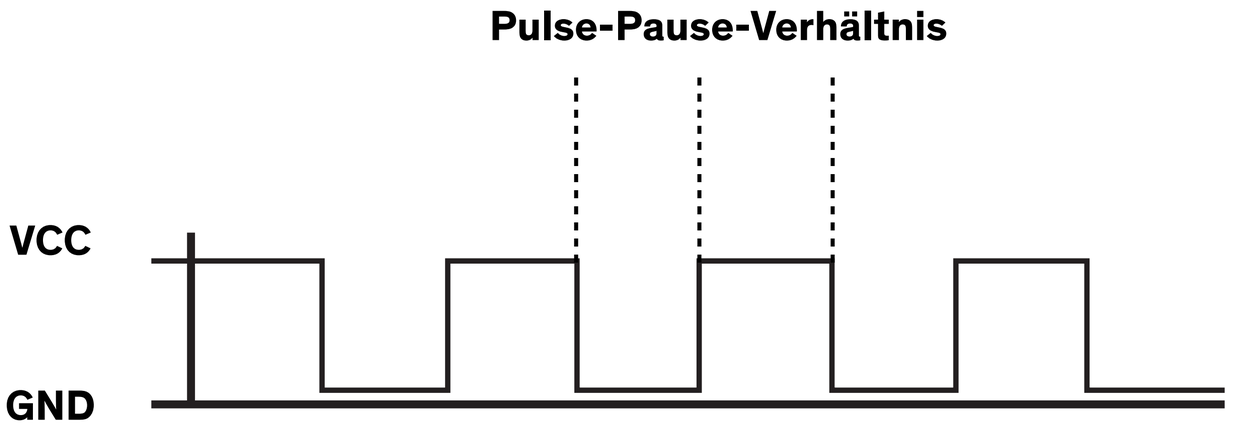
Duty Cycle 0%



PWM

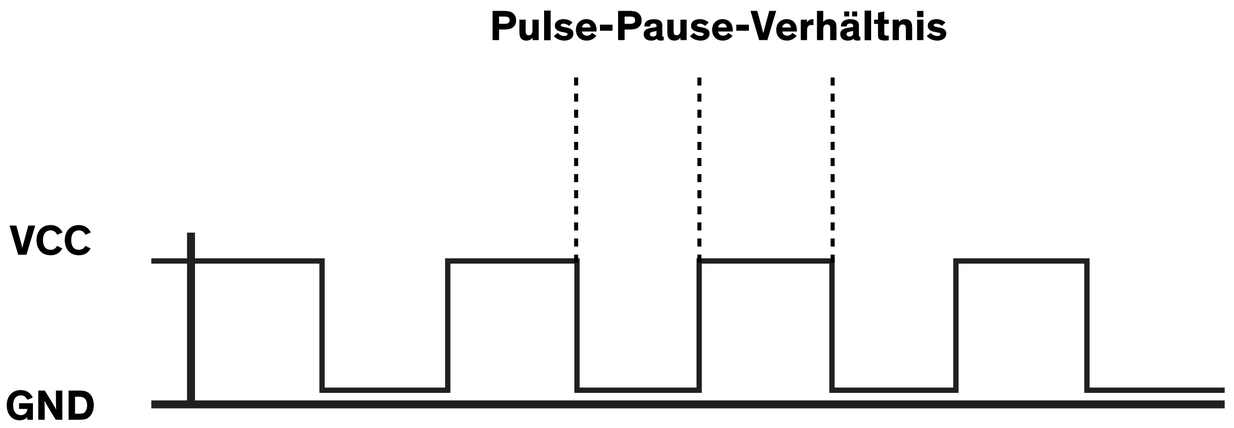
Duty Cycle 100%



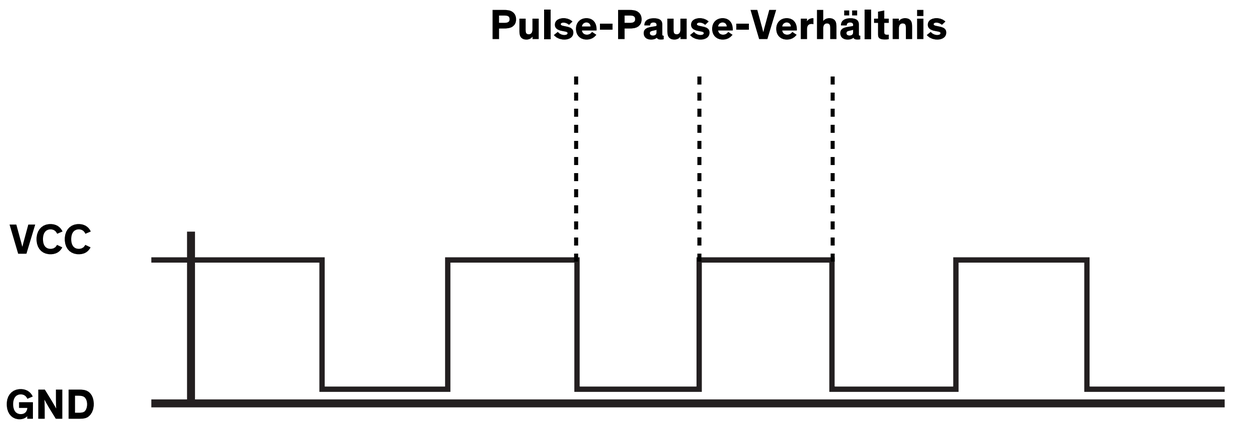


PWM

Duty Cycle 50%

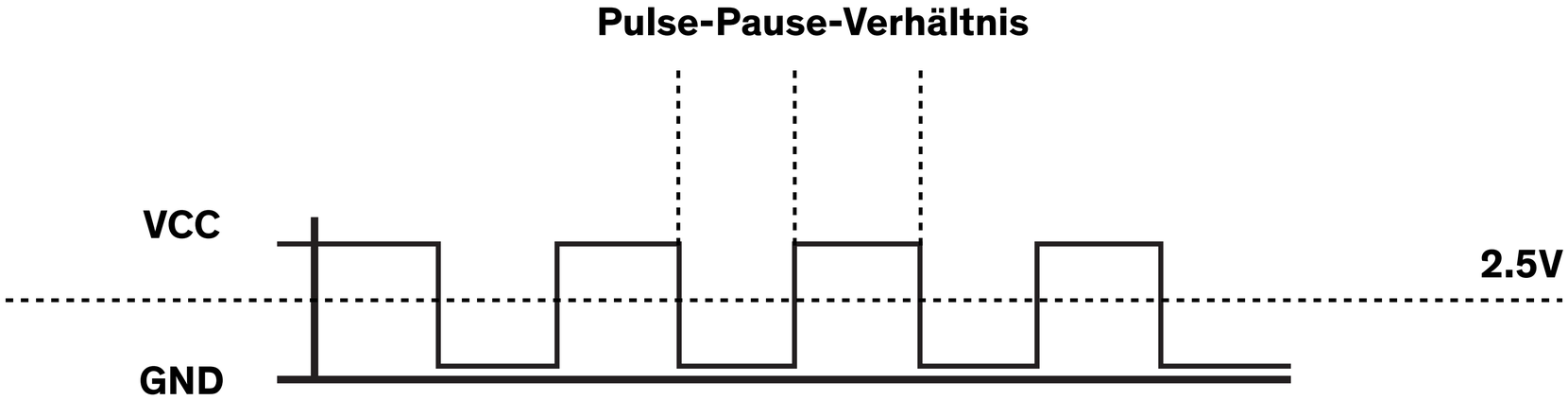


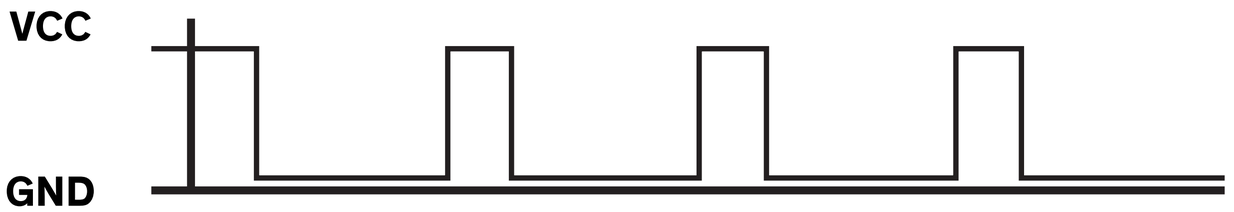
Duty Cycle 50%
50% von 5V = 2.5V



PWM

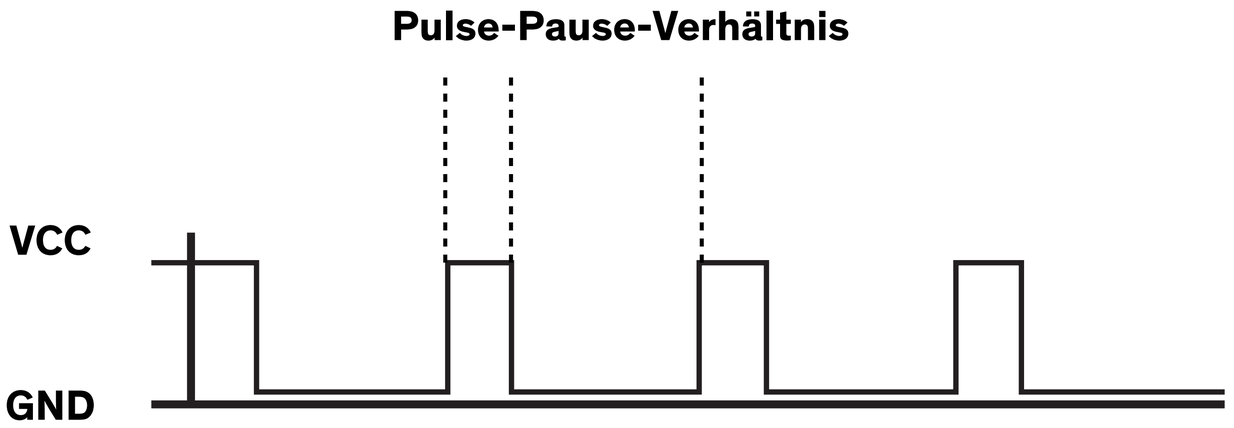
Duty Cycle 50%
50% von 5V = 2.5V





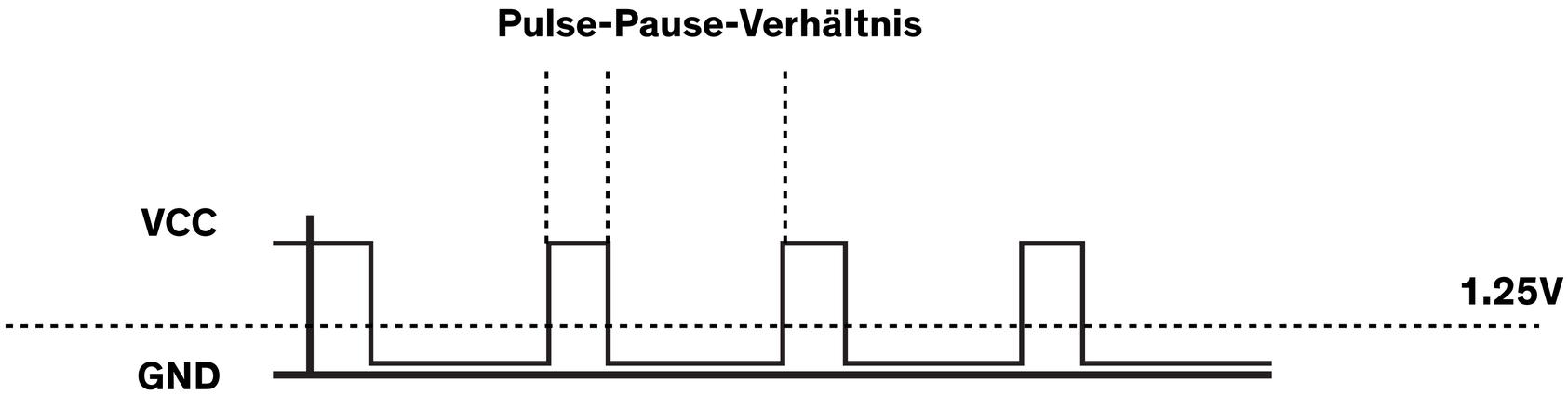
PWM

Duty Cycle 25%
= 1.25V

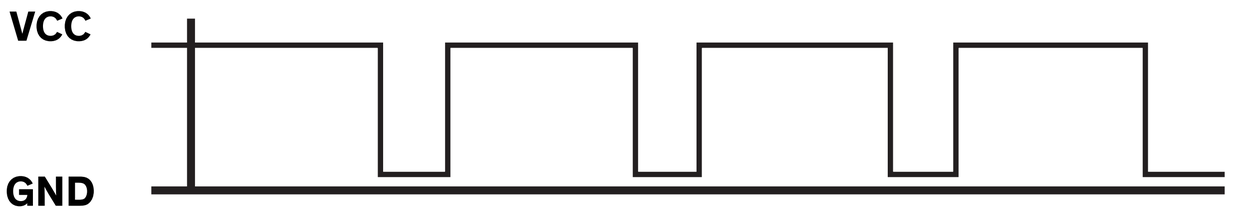


PWM

Duty Cycle 25%
= 1.25V

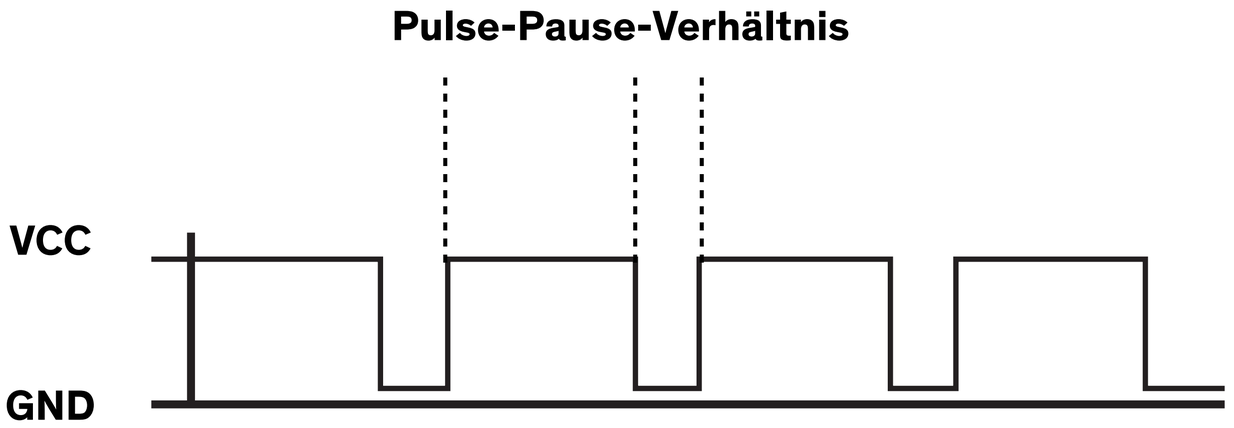


PWM

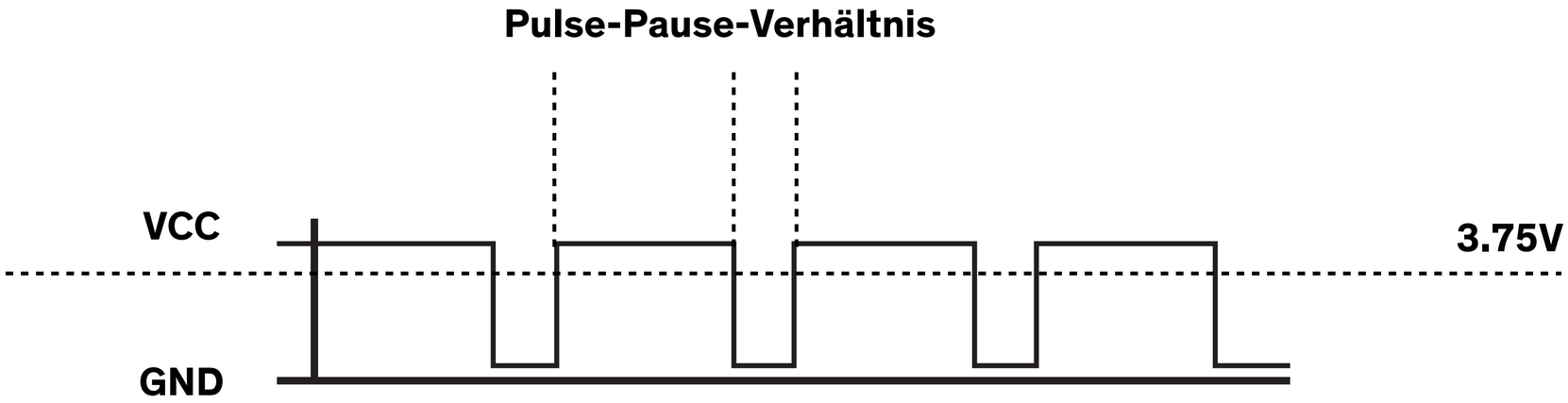


PWM

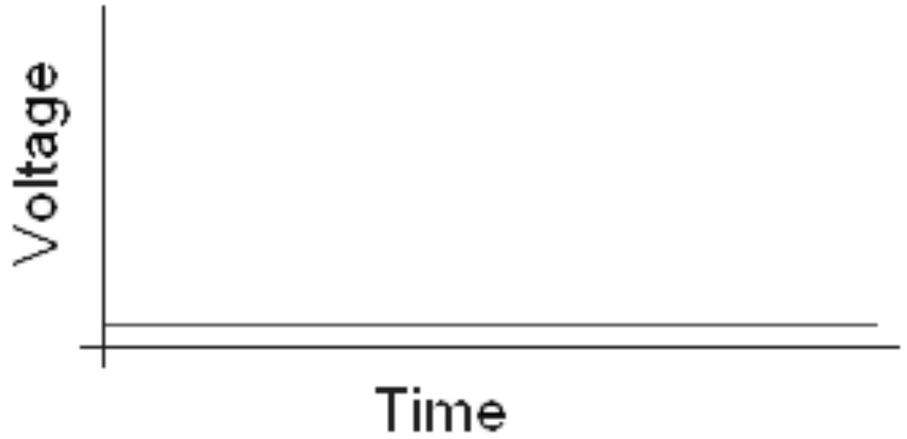
Duty Cycle 75%
= 3.75V

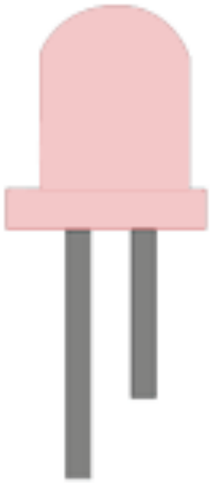


Duty Cycle 75%
= 3.75V



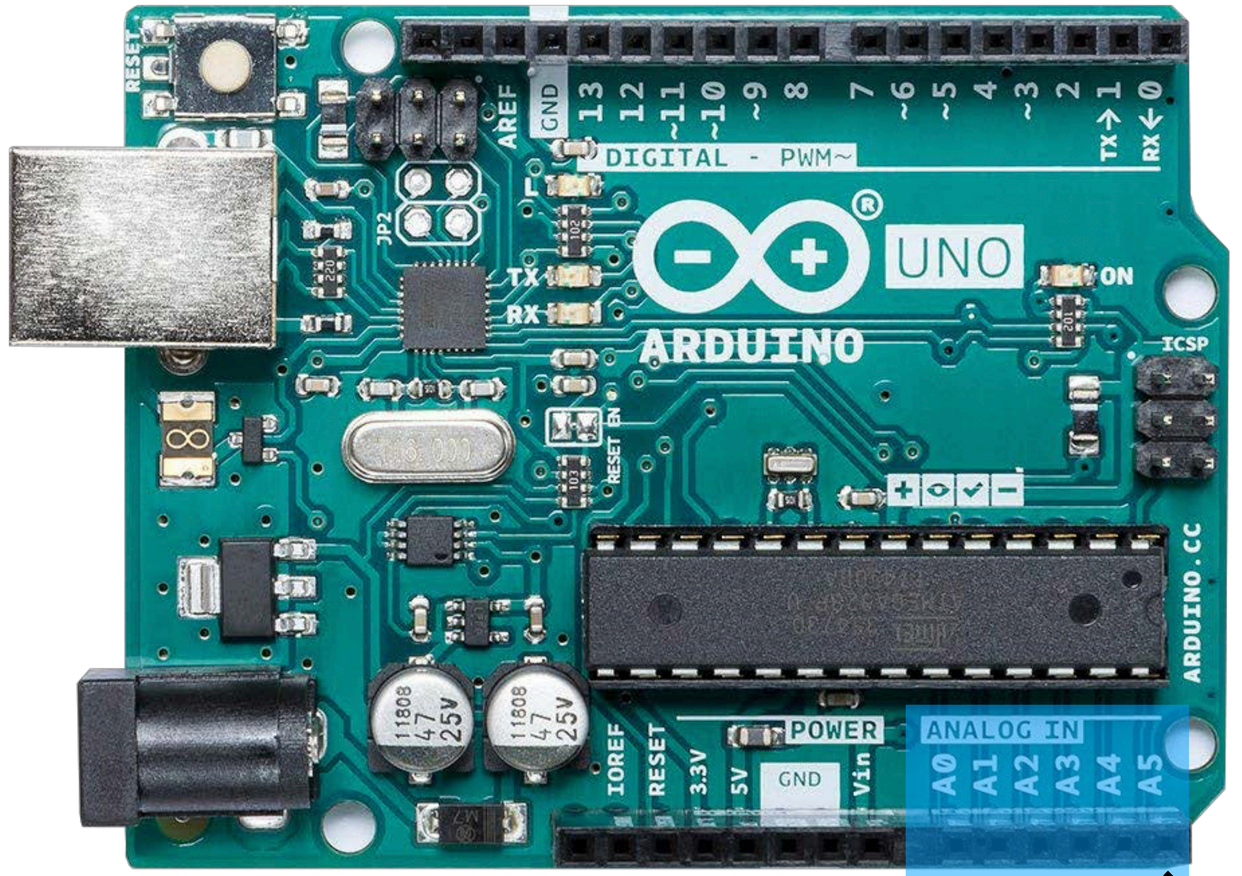
PWM





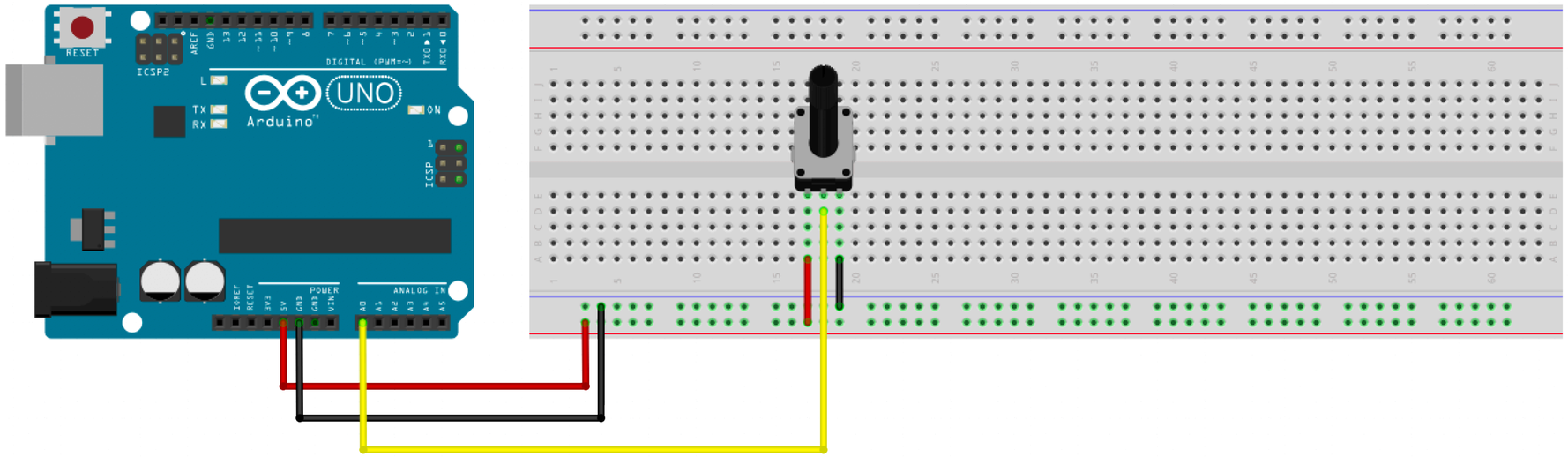
Duty Cycle: 0%

PWM



6 Analoge Pins (A0 – A5)

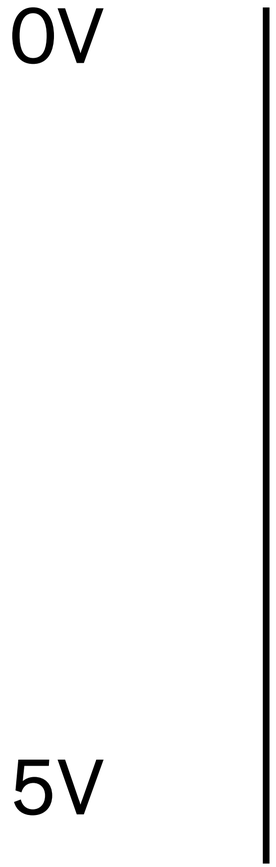
Analoge Signale lesen



Analoge Signale lesen

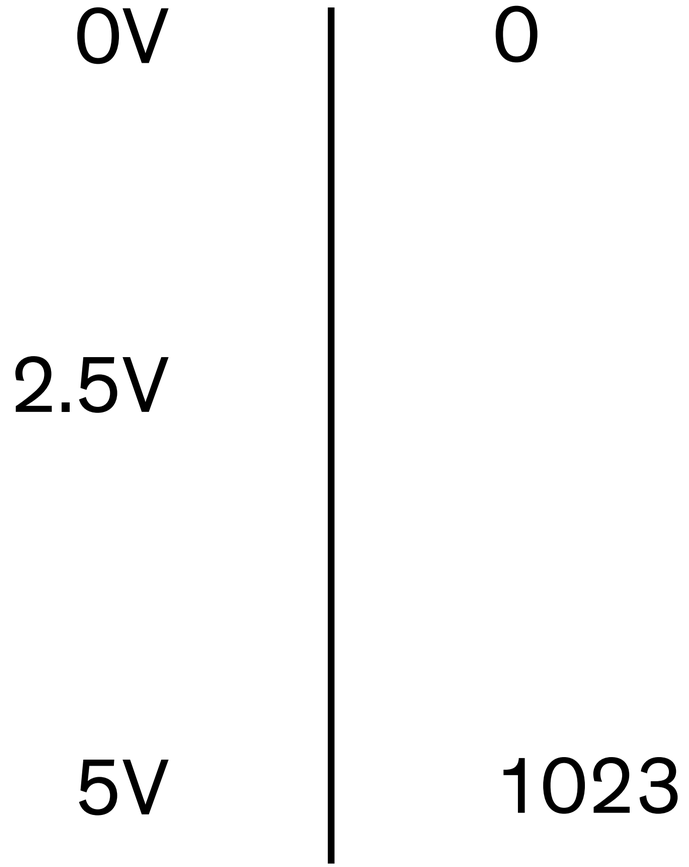
Mapping

Mapping



Mapping

Mapping



Mapping

0V	0
1.25V	255
2.5V	511
3.75V	767
5V	1023

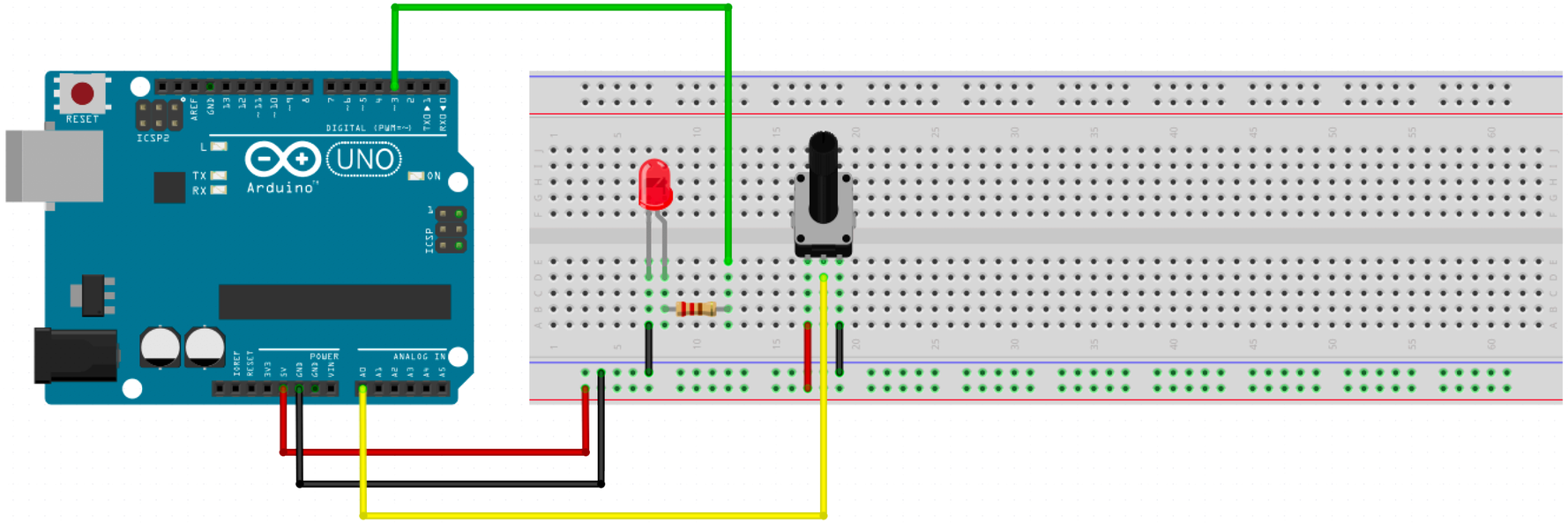
Mapping

```
map(eingabeWert, b1_untere_grenze, b1_obere_grenze, b2_untere_grenze, b2_obere_grenze);
```


Mapping

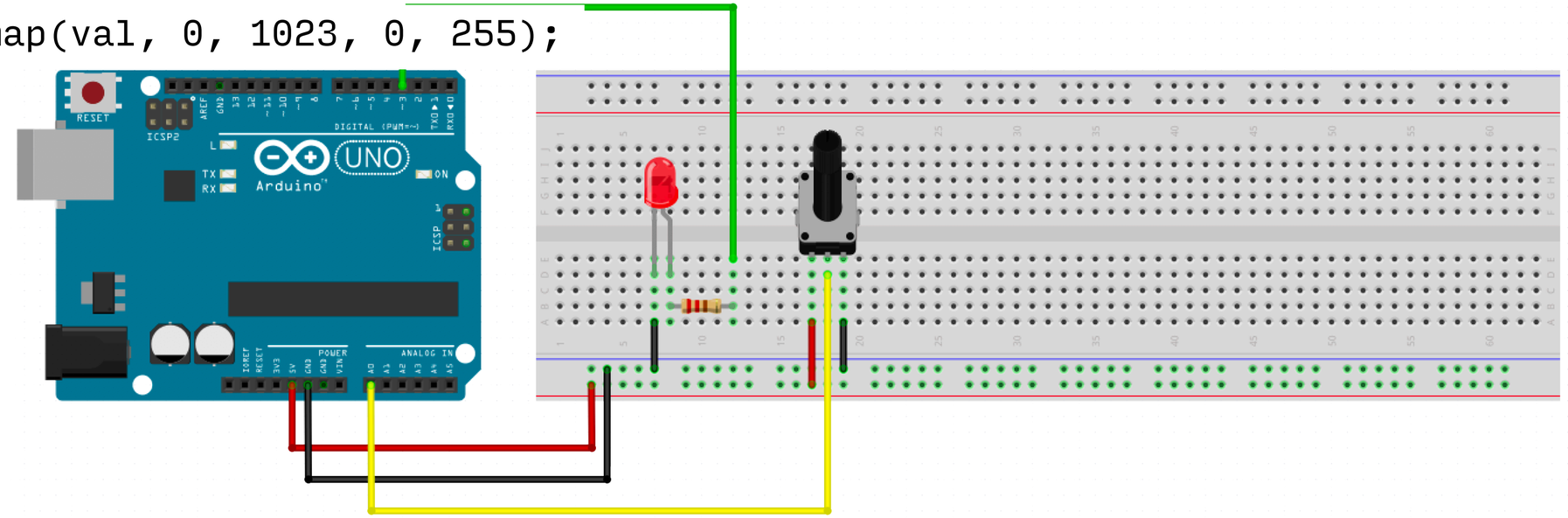
```
map(eingabeWert, b1_untere_grenze, b1_obere_grenze, b2_untere_grenze, b2_obere_grenze);

map(eingabeWert, 0, 5, 0, 1023);
```

Mapping



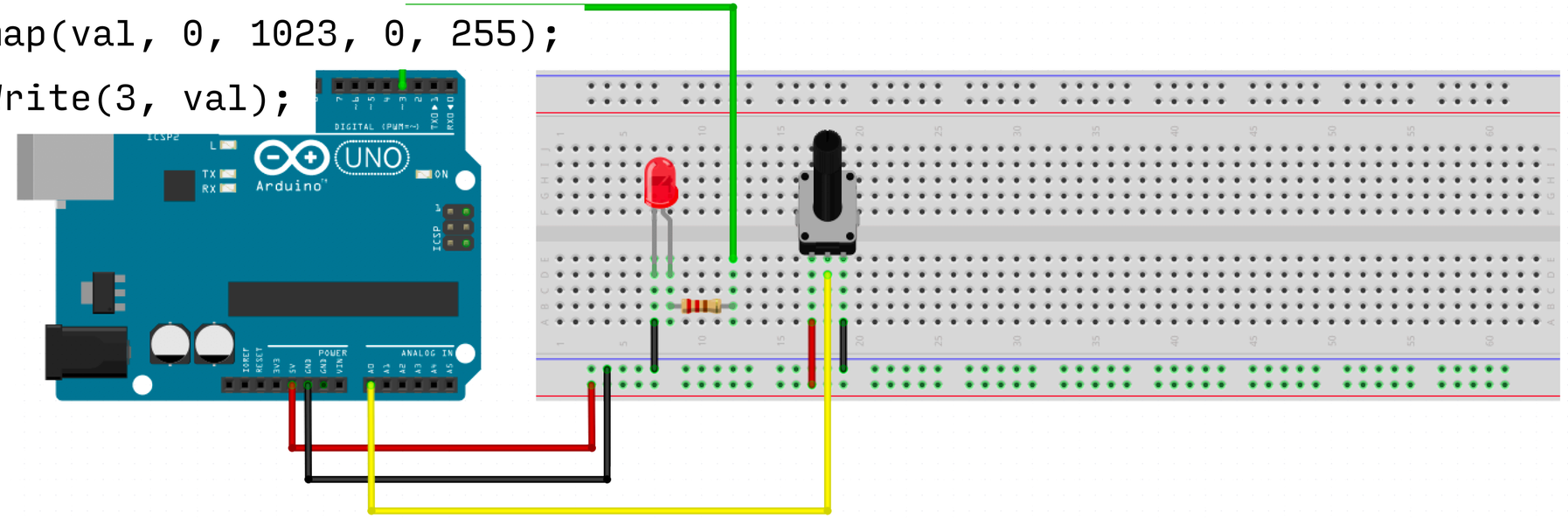
Analoge Signale lesen

```
void setup() {}

void loop() {
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
}
```



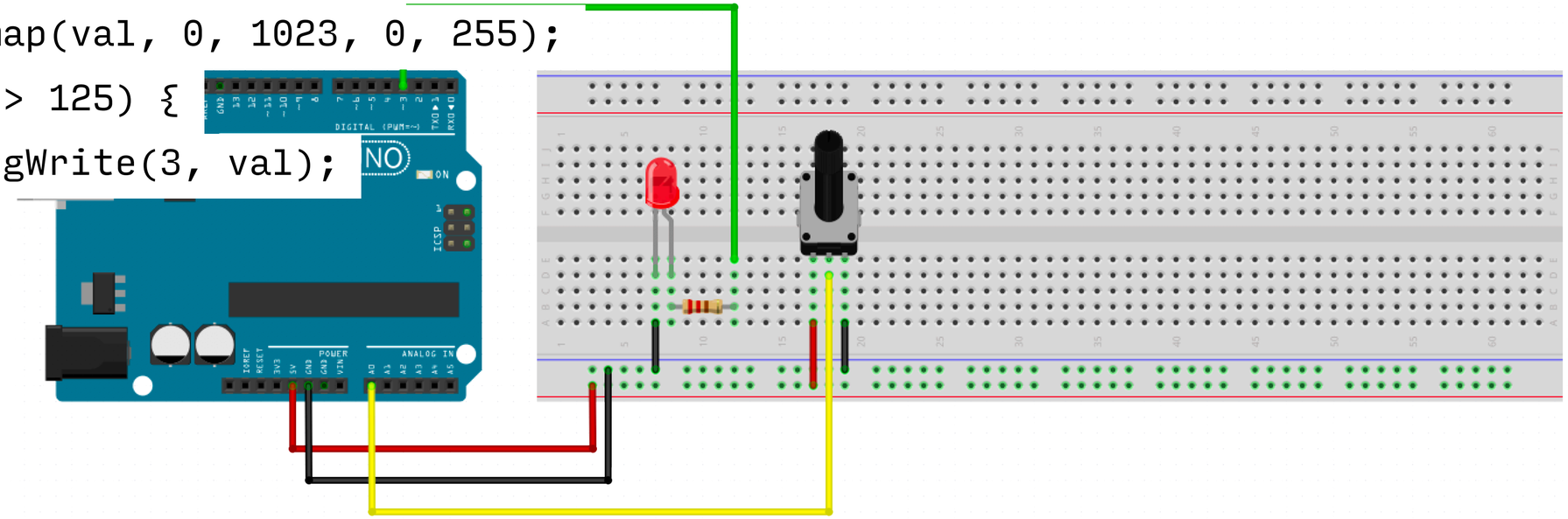
Analoge Signale lesen

```
void setup() {}

void loop() {
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(3, val);
}
```



Analoge Signale lesen

```
void setup() {}

void loop() {
  int val = analogRead(0);
  val = map(val, 0, 1023, 0, 255);
  if(val > 125) {
    analogWrite(3, val);
  }
}
```



Analoge Signale lesen

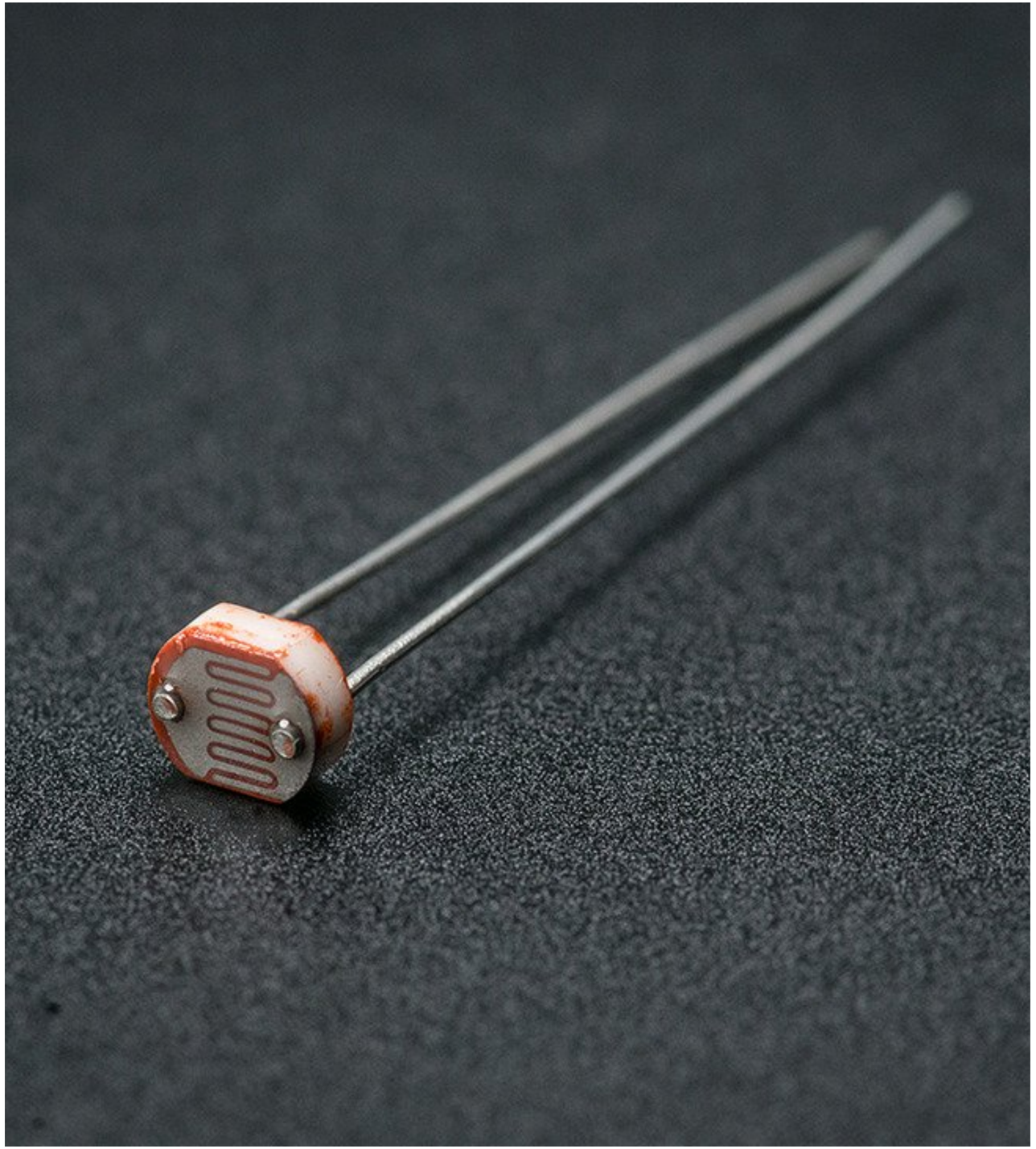
Geschwindigkeit
Beschleunigung
Orientierung
Vibration
Neigung
Position
Helligkeit

Feuchtigkeit
Töne
Temperatur
Druck
Magnetfelder
Drehung
Alkohol

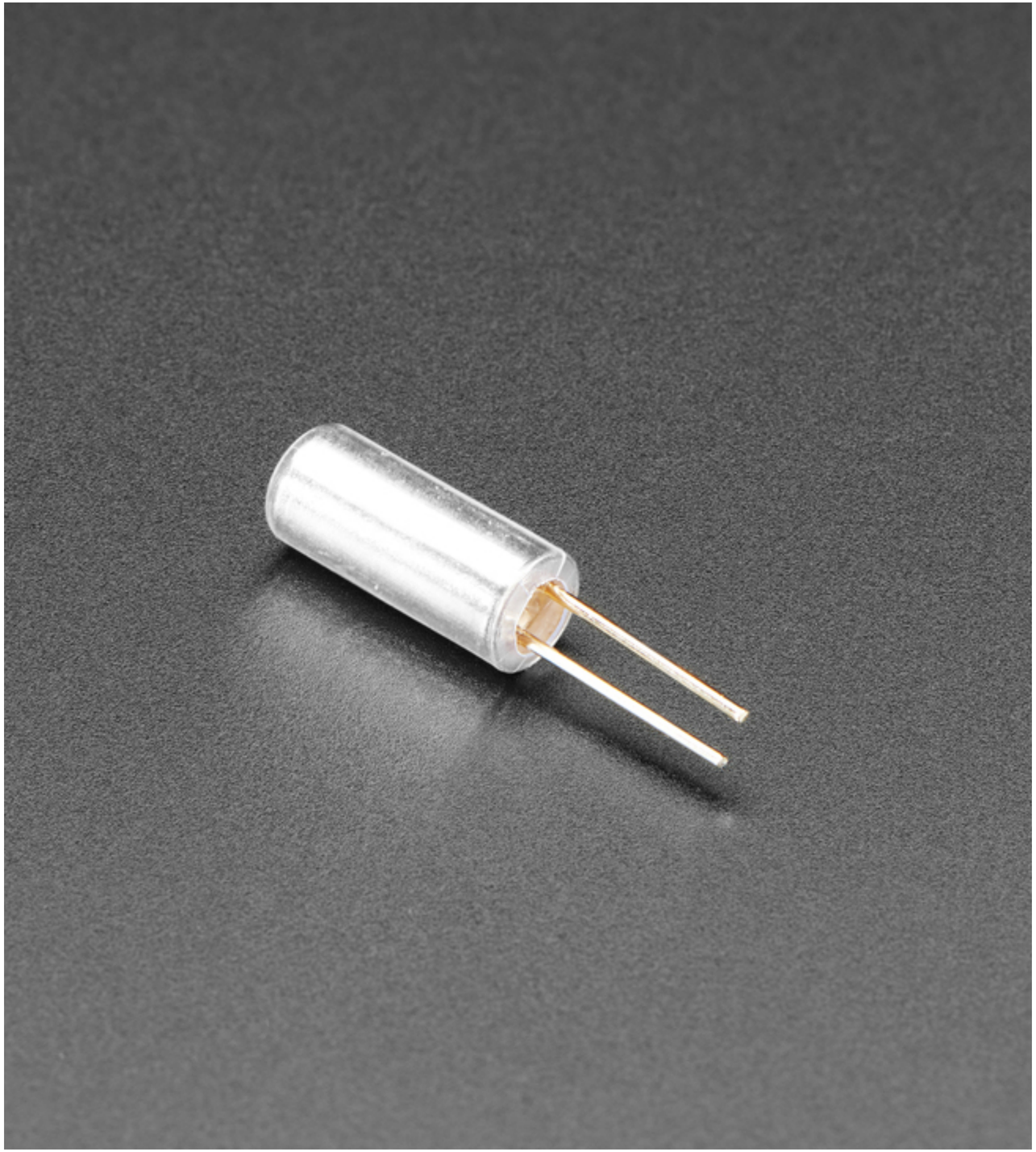
Signalstärke
Radio
WiFi
UV
Infrarot

Analoge Sensoren

Photowiderstand



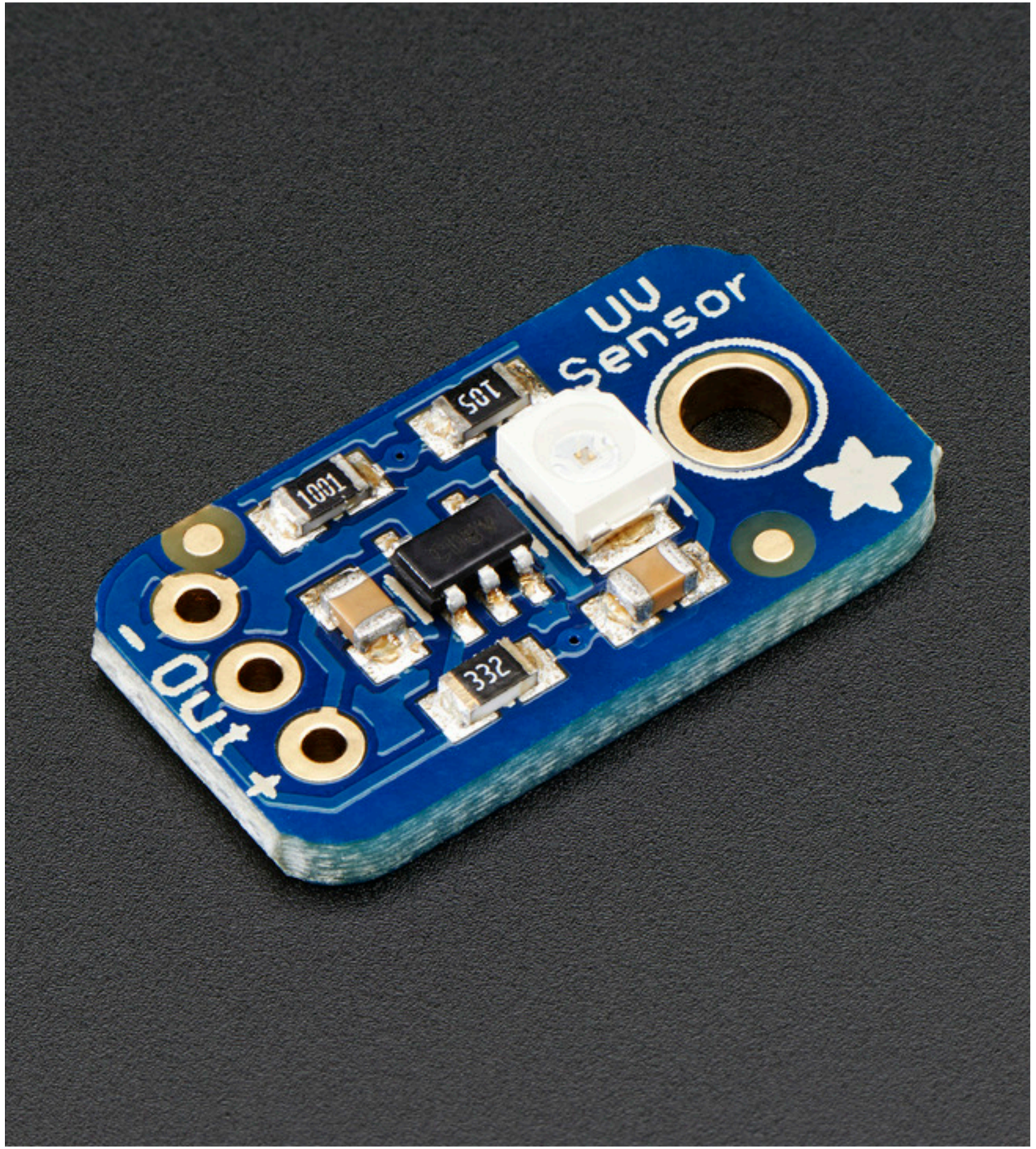
Neigungssensor



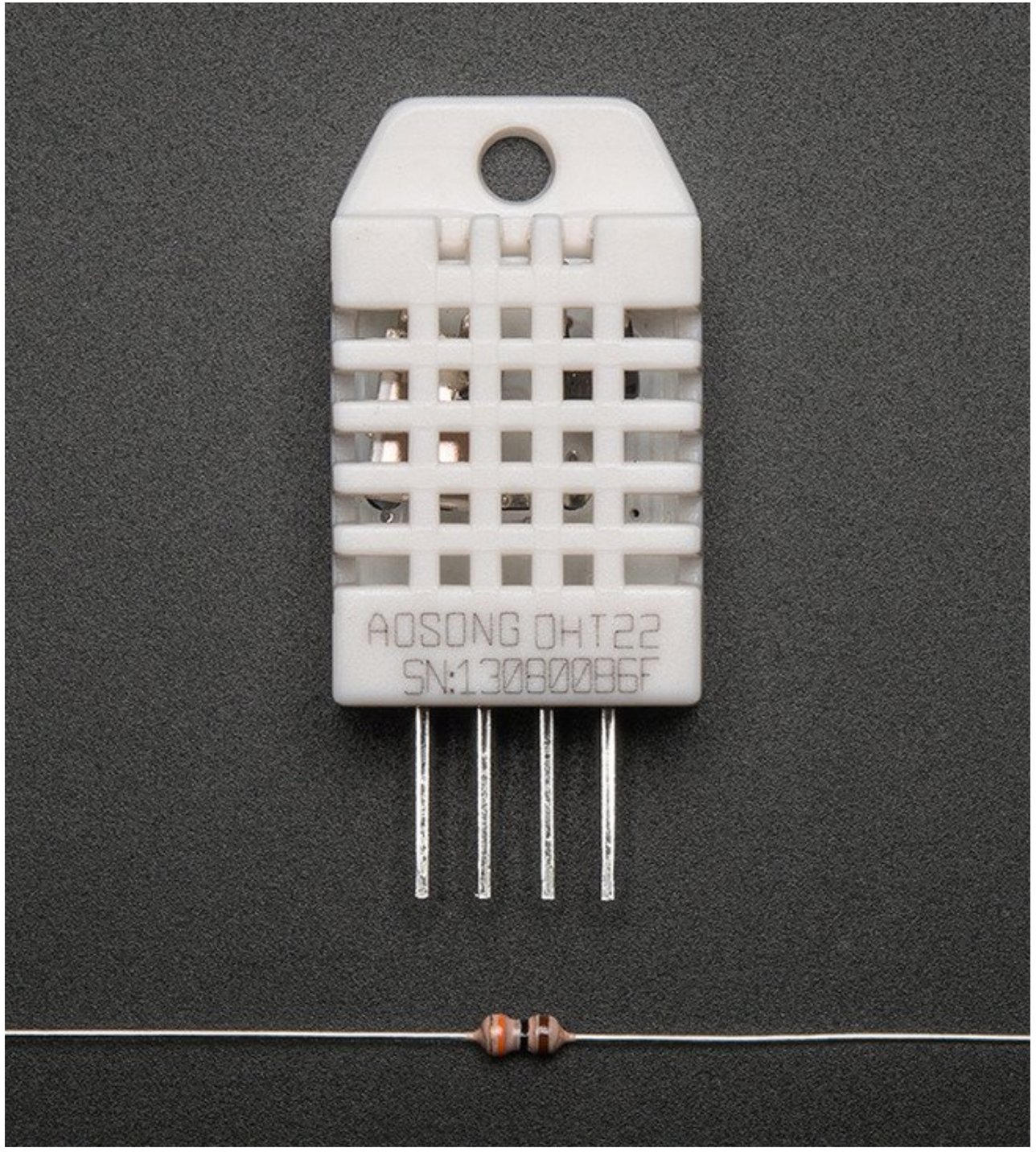
PIR Bewegungssensor

2-Achsen Joystick

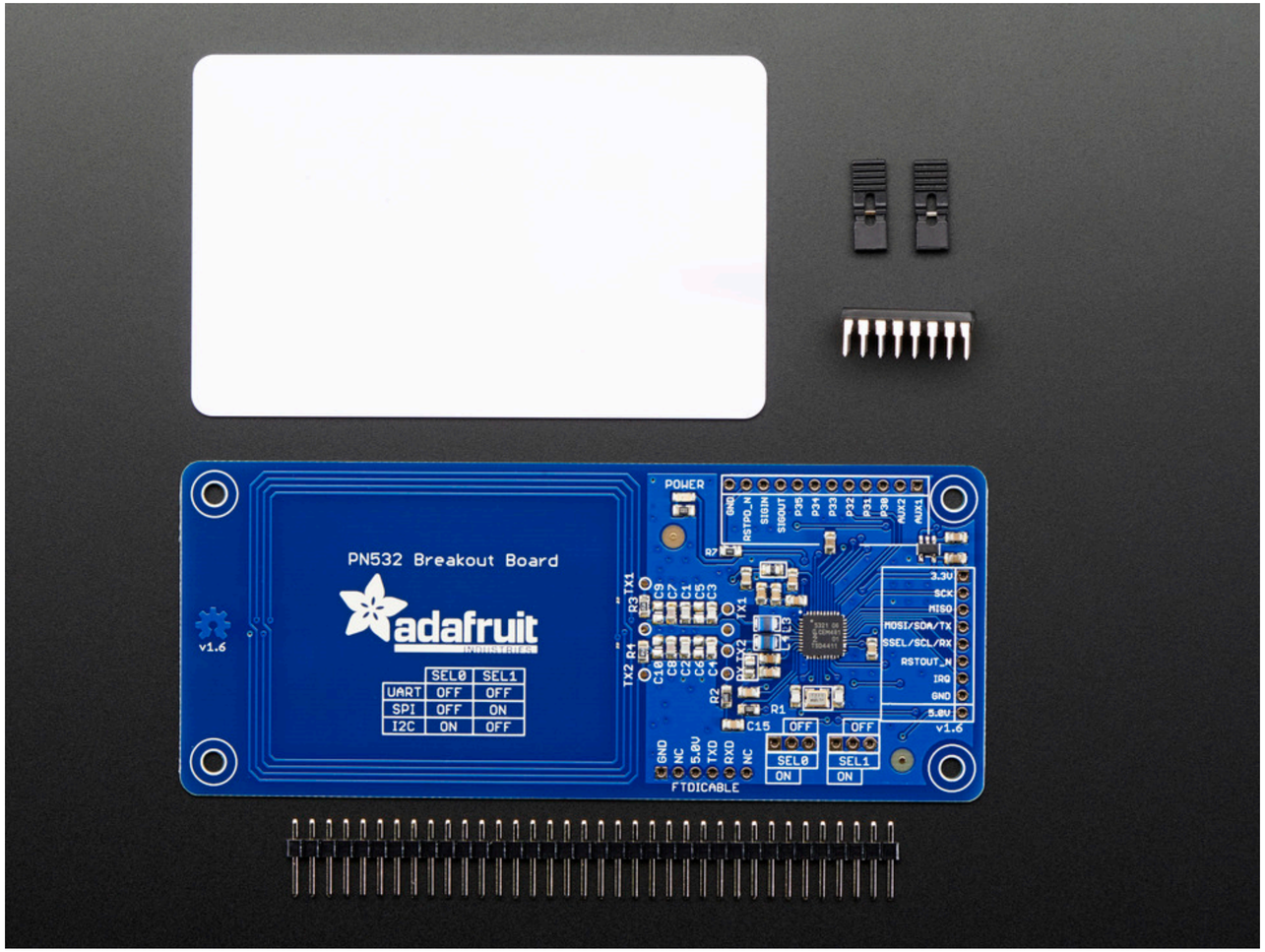
UV Sensor



Feuchtigkeitssensor



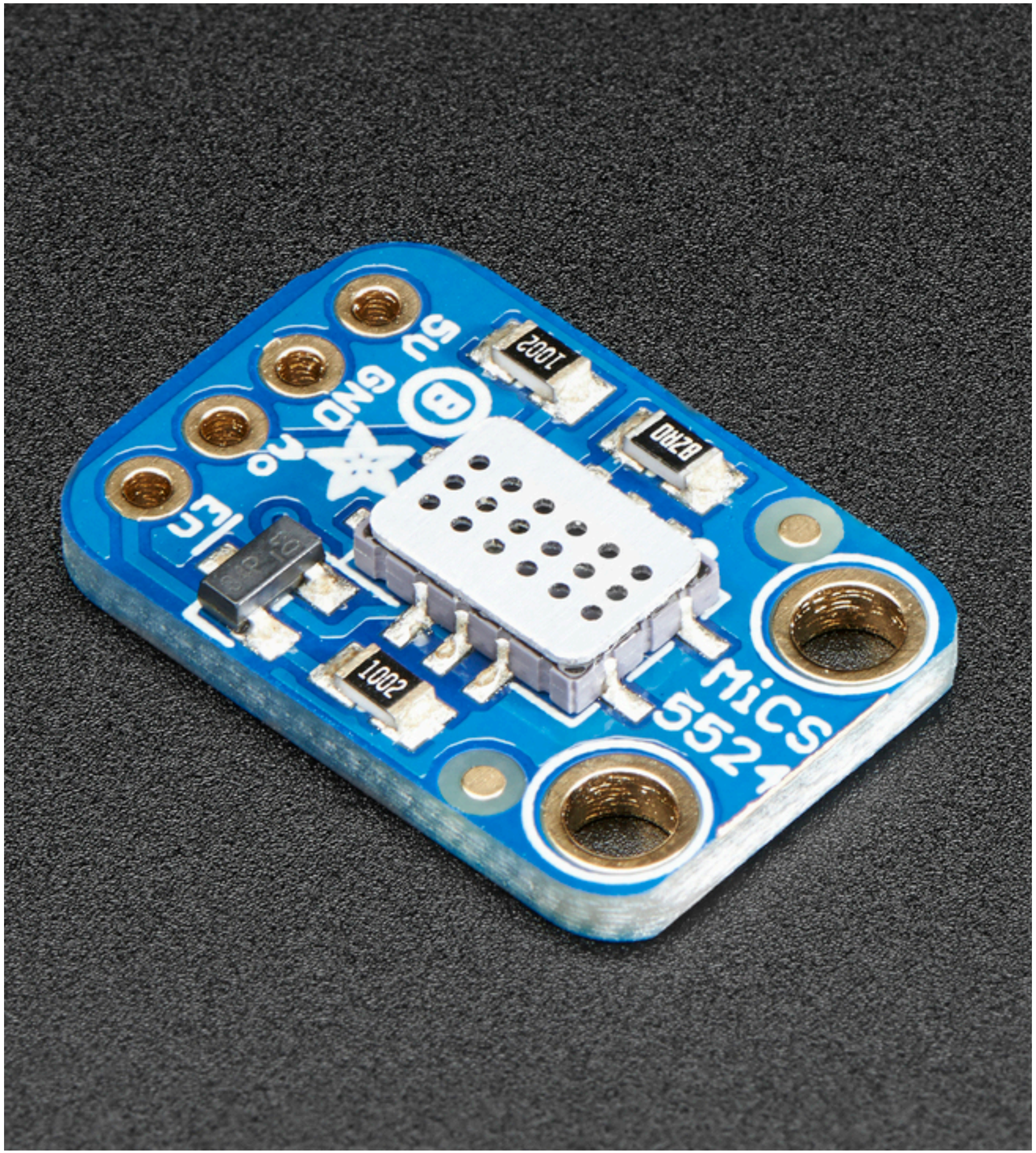
Drehgeber


PN532 Breakout Board
adafruit
v1.6
SEL0 SEL1
UART OFF OFF
SPI OFF ON
I2C ON OFF
POWER
FTDICABLE
SEL0
SEL1
ON
OFF
3.3V
SCK
MISO
MOSI/SDA/TX
SSEL/SCL/RX
RSTOUT_N
IRQ
GND
5.0V

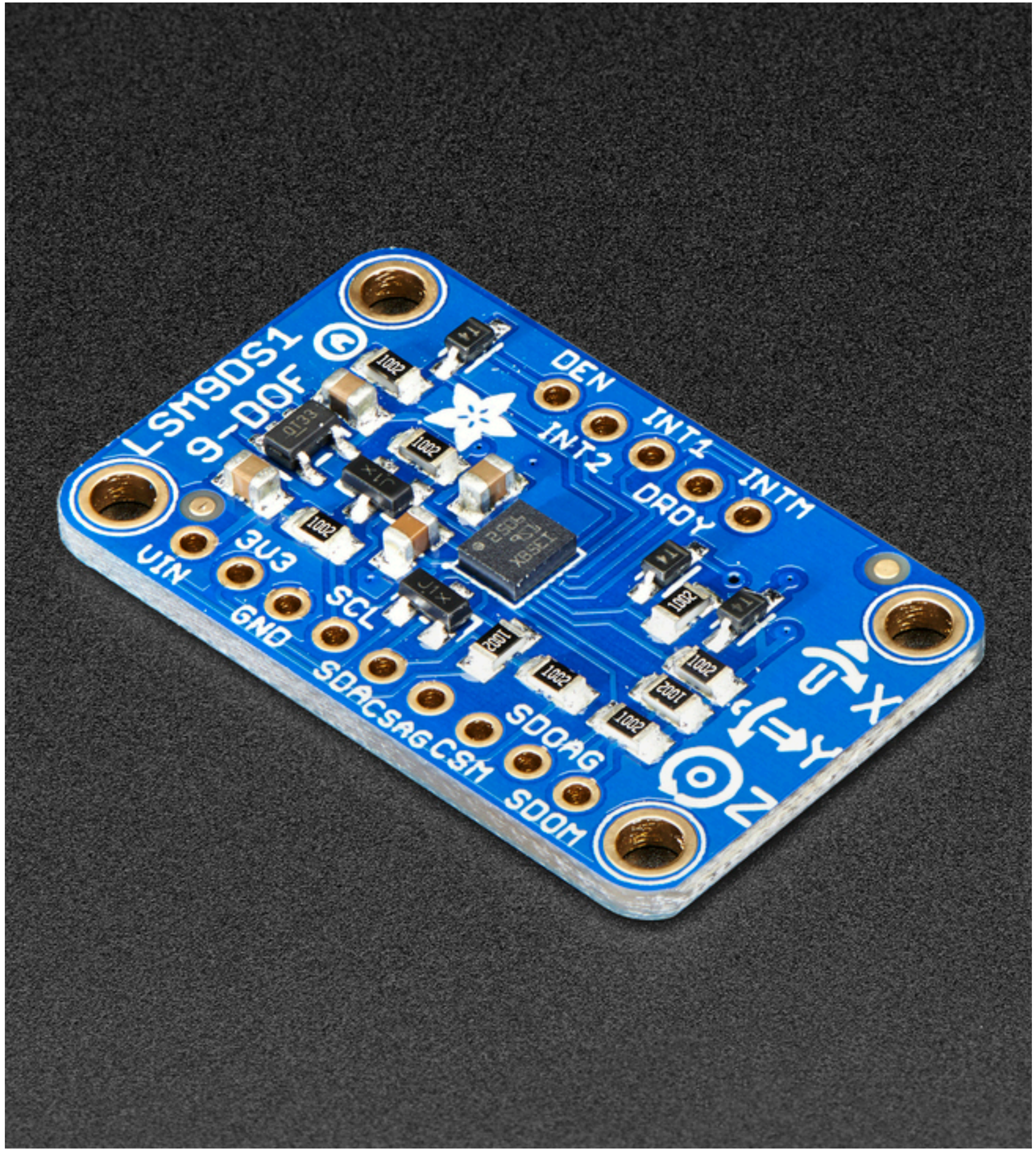
NFC / RFID

Temparatur + Feuchtigkeit

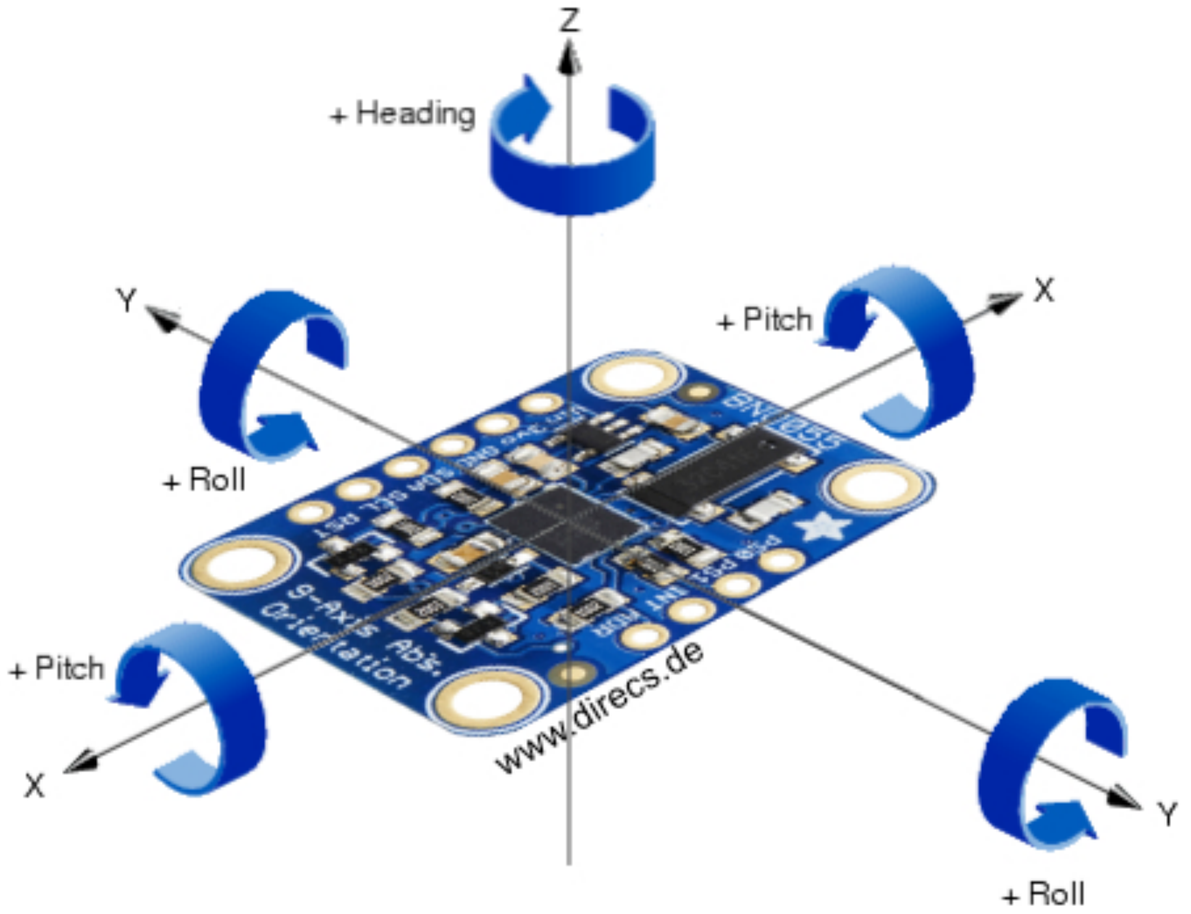
Alcohol und VOC Gase

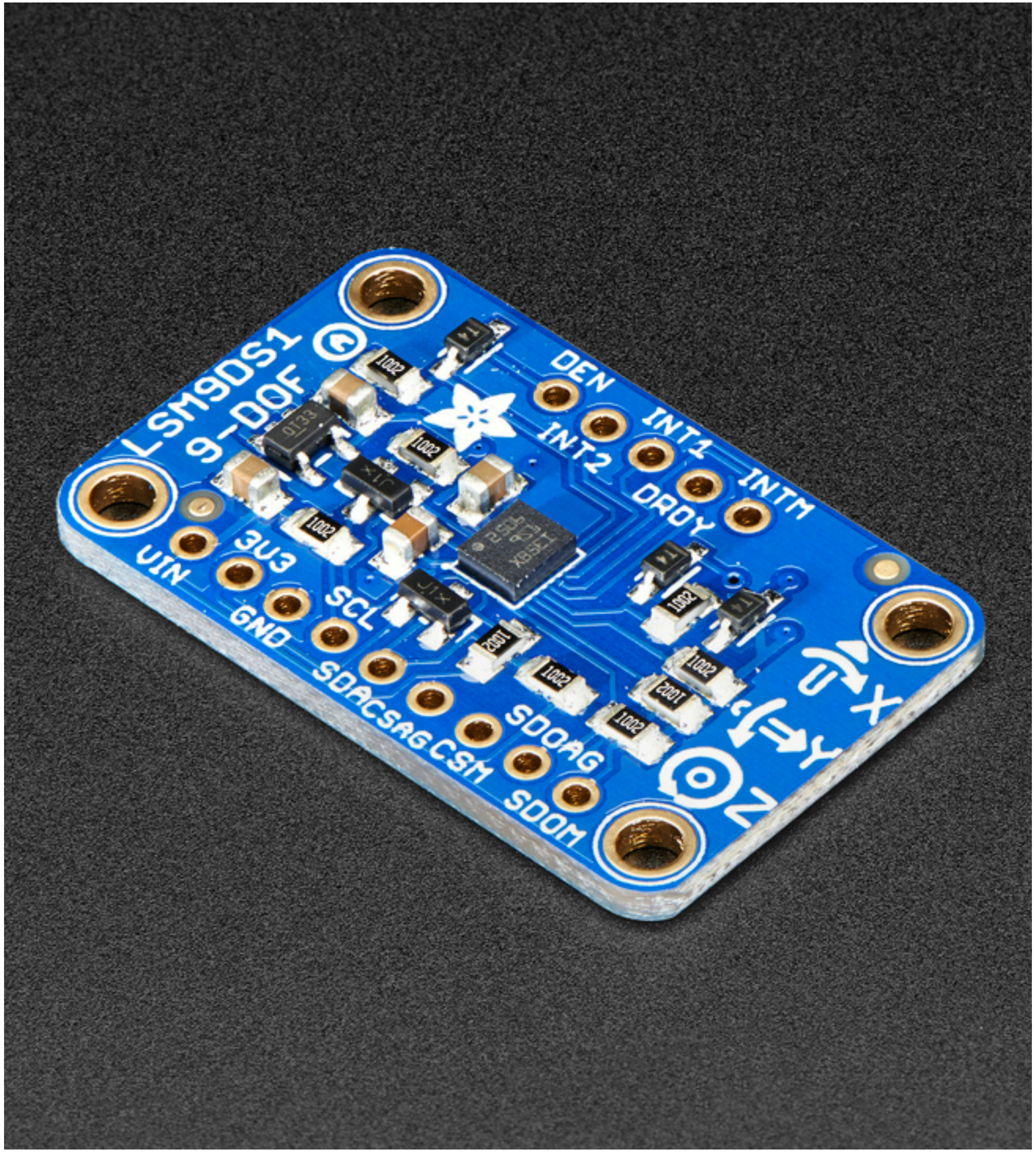


3-Achsen Gyroskop
3-Achsen Beschleunigungssensor
3-Achsen Magnetometer

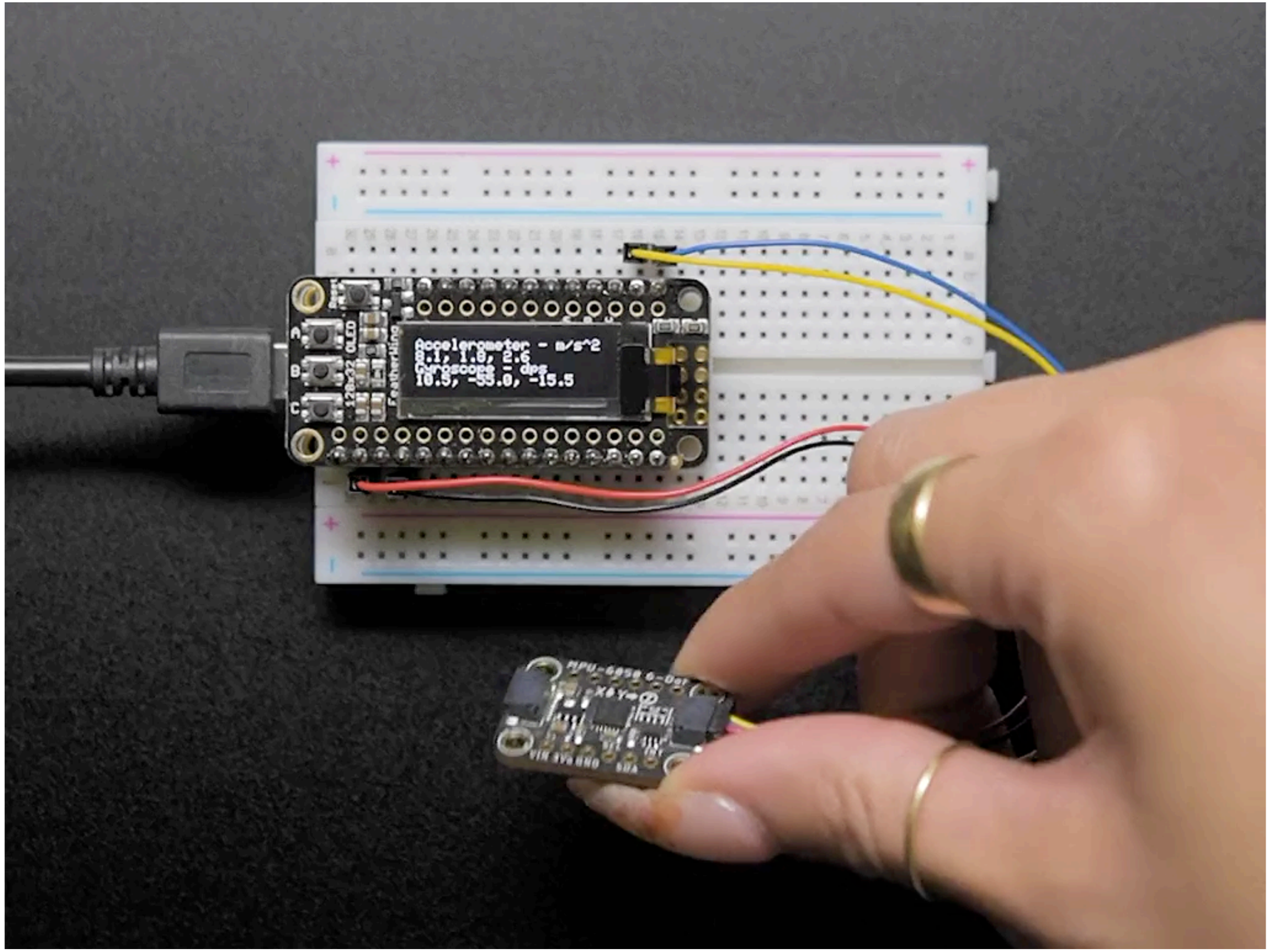


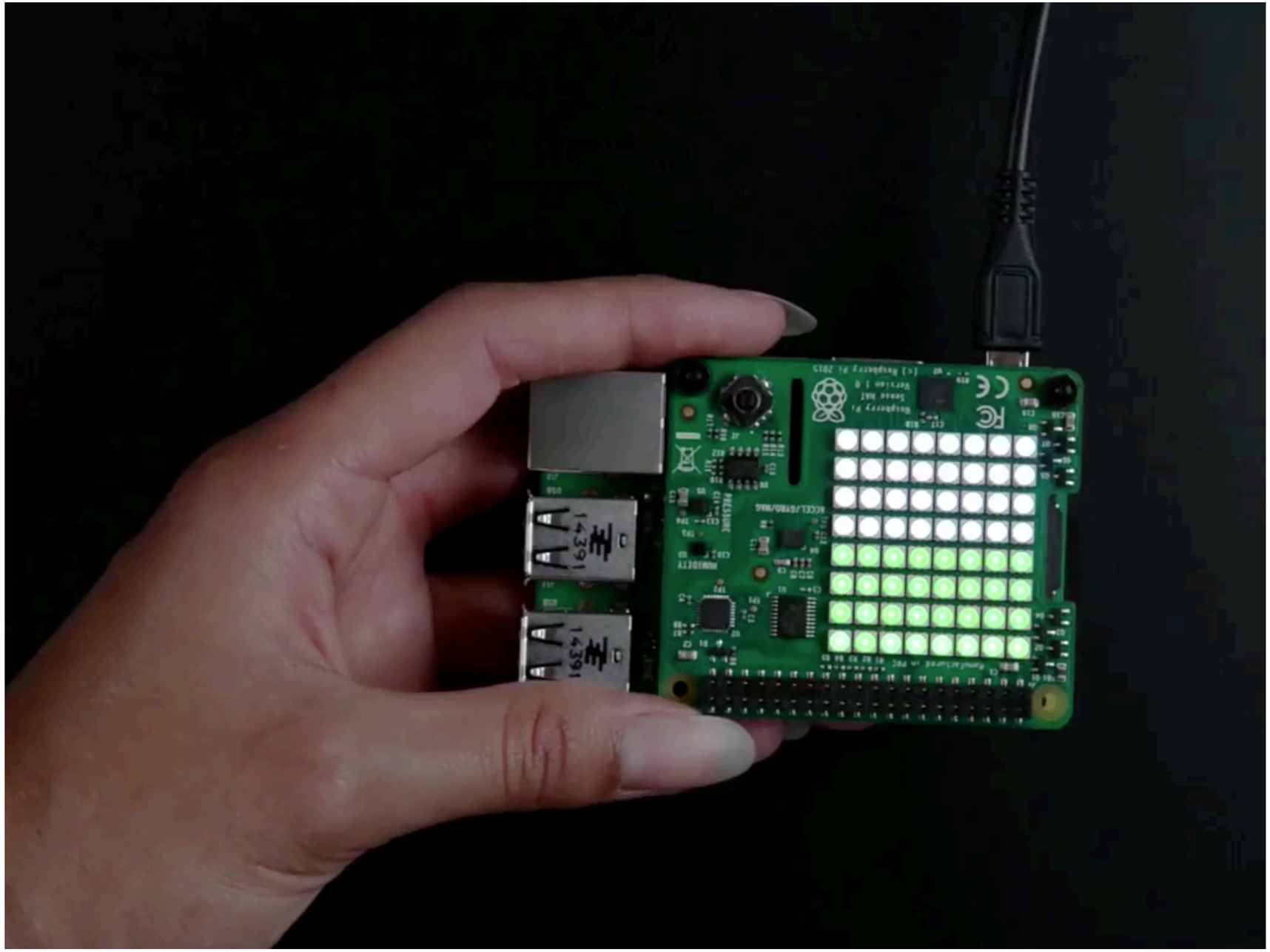
9-DOF


Z
+ Heading
Y
X
+ Pitch
+ Roll
+ Pitch
X
Y
+ Roll
www.direcs.de


LSM9DS1
9-DOF
DEN
INT1
INTM
INT2
DRDY
VIN
3V3
GND
SCL
SDA
CSAG
CSM
SDOAG
SDOM
X
Y
Z

9-DOF


Accelerometer - m/s^2
8.1, 1.8, 2.6
Gyroscope - dps
10.5, -55.0, -15.5



Display
Segment
Anzeigen
Mechanisch
Motoren
DC-Motor
Servomotor
Piezo
Buzzer

LED
Glühbirne
Magneten
Flipdots
Lautsprecher
Schrittmotor
Gase
Hitze
Matrizen

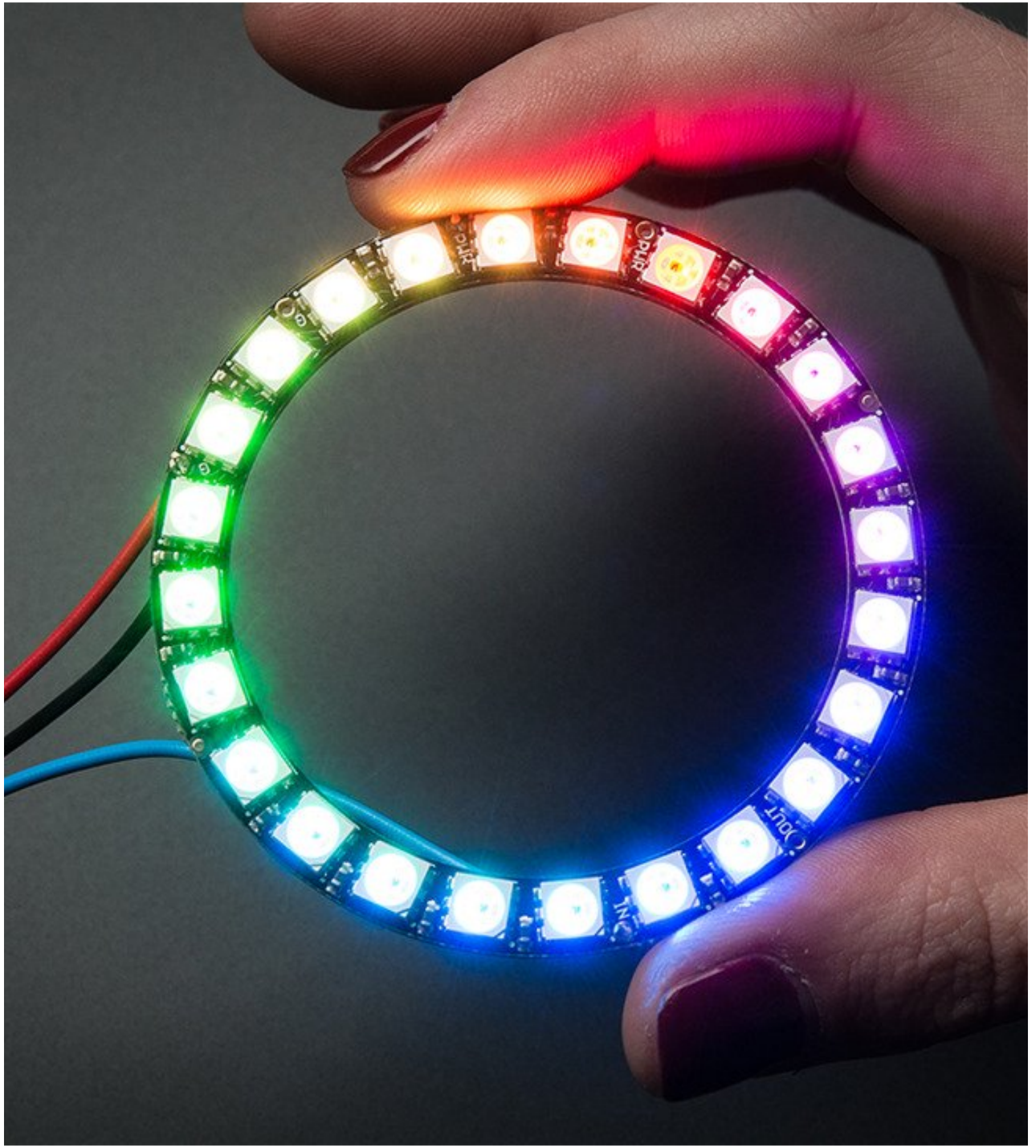
E-Ink
Thermodrucker
LCD

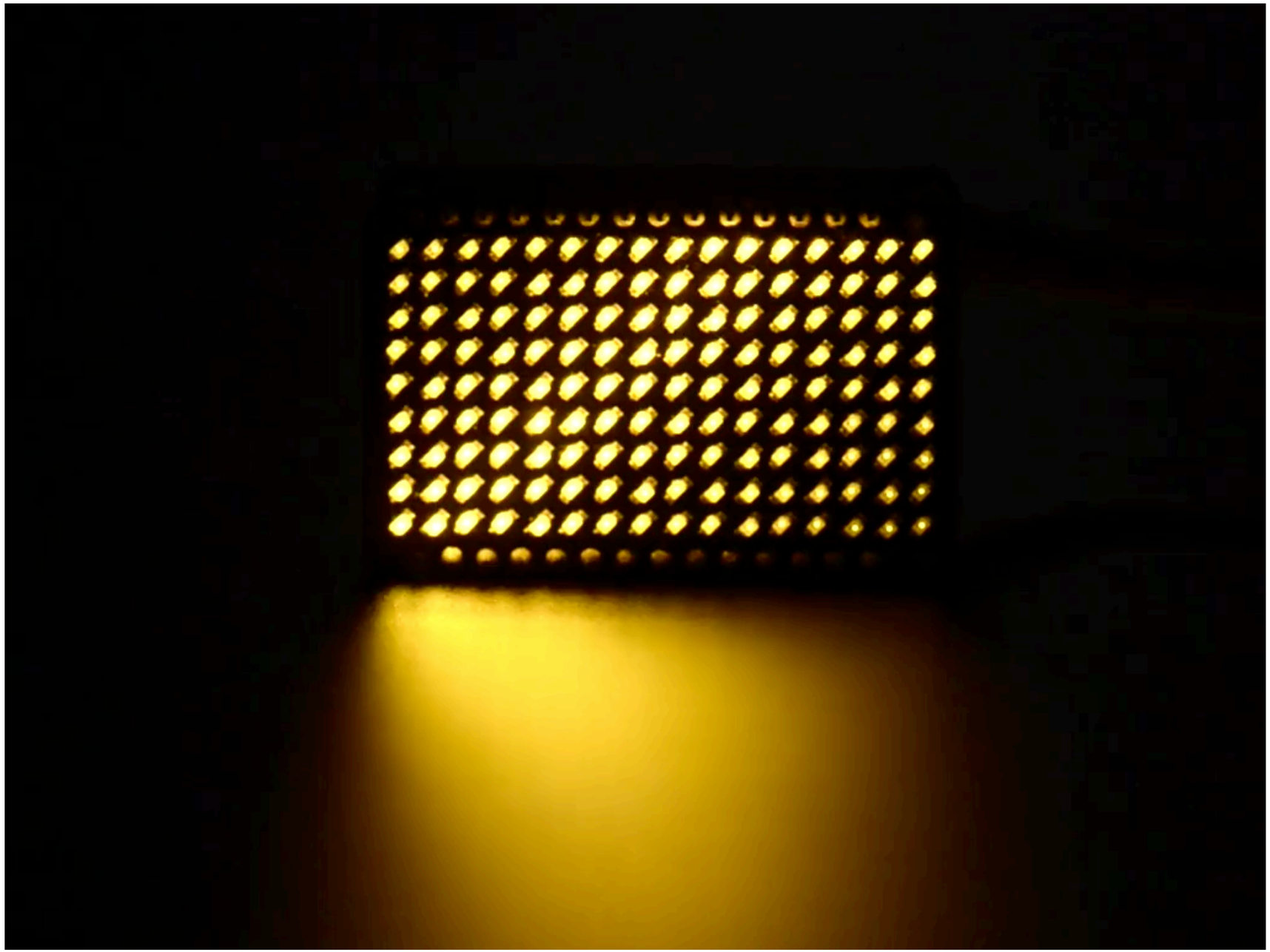
Outputs

LED

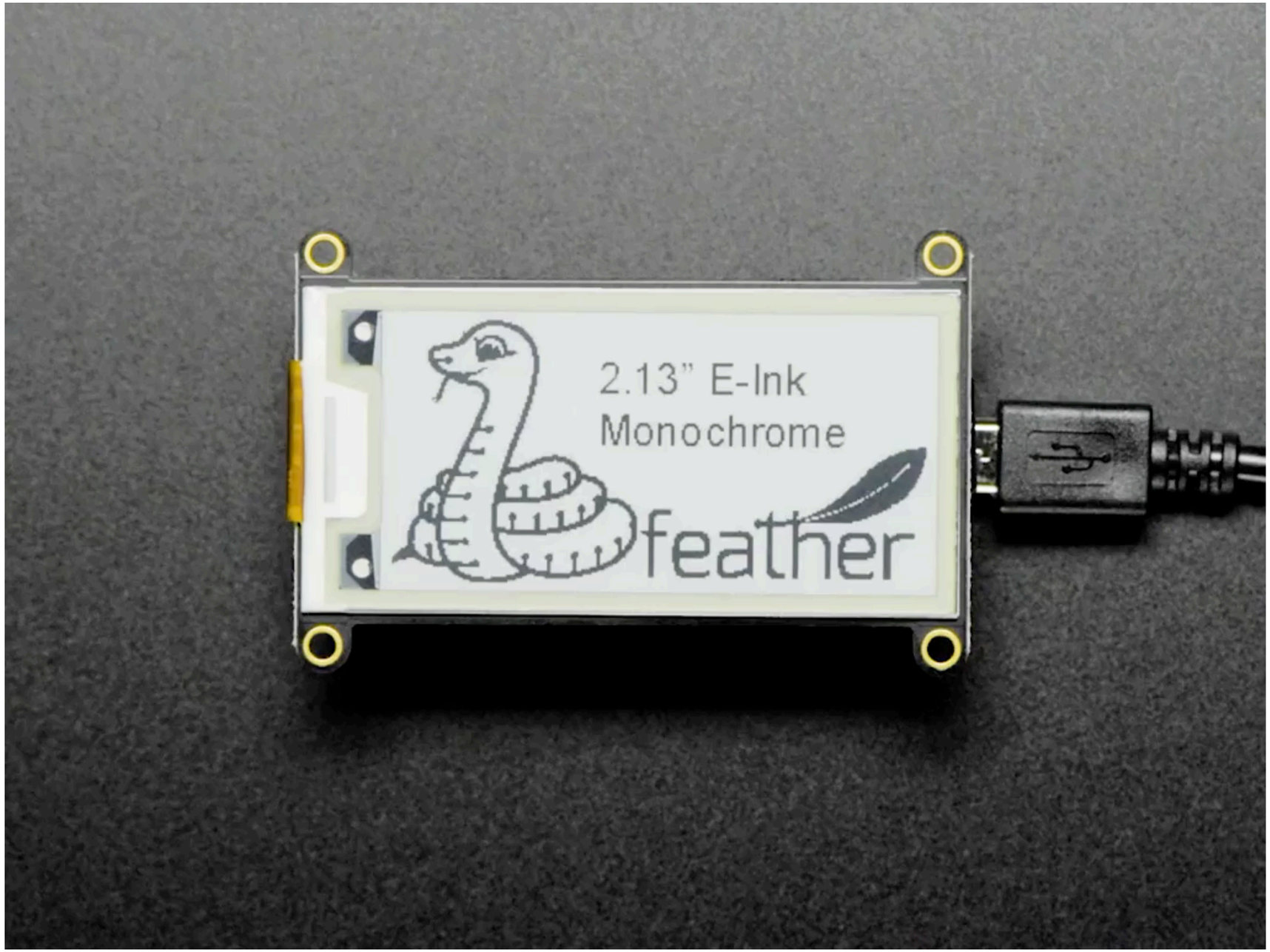
Neopixel

Neopixel

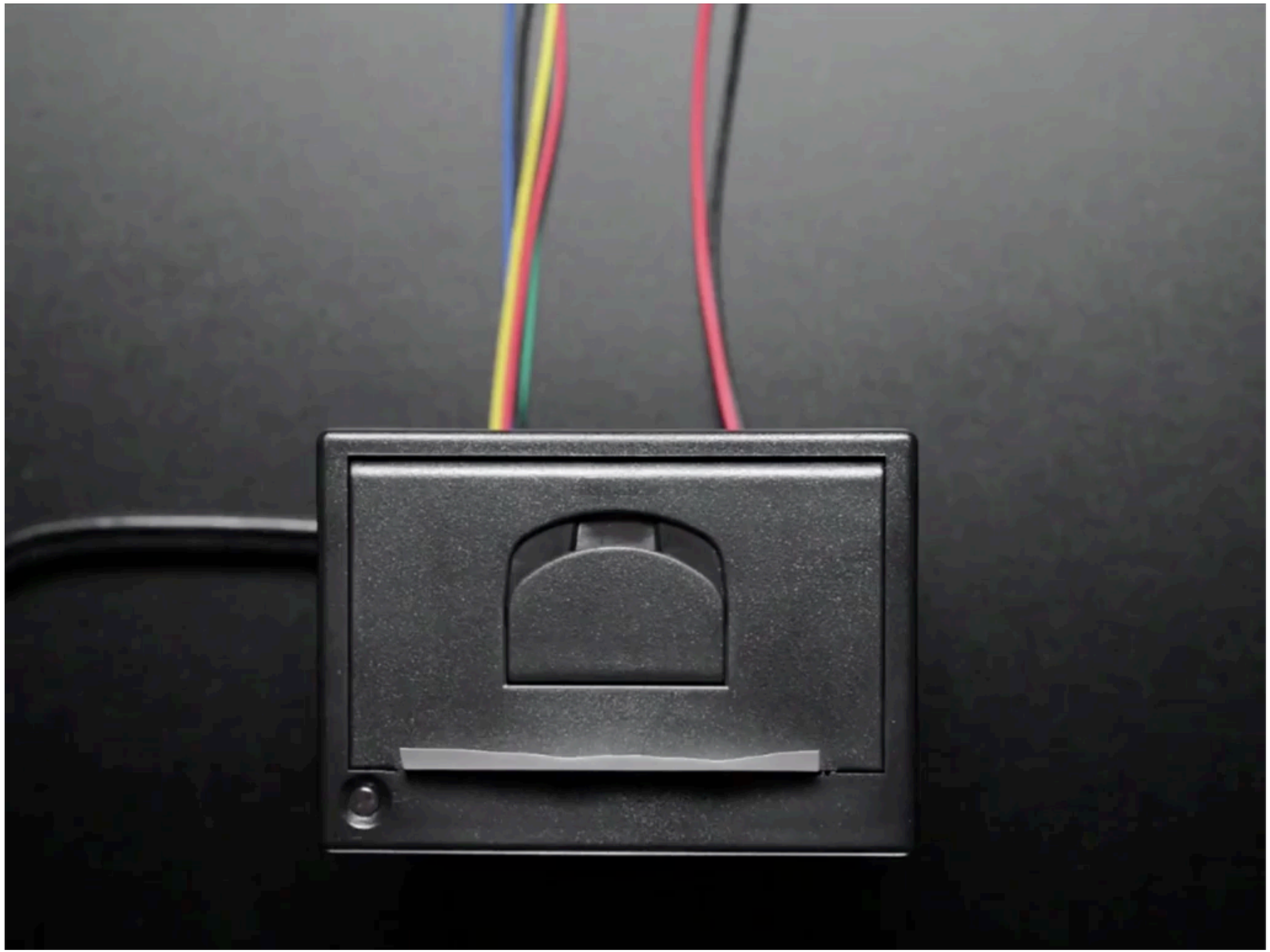




Matrizen

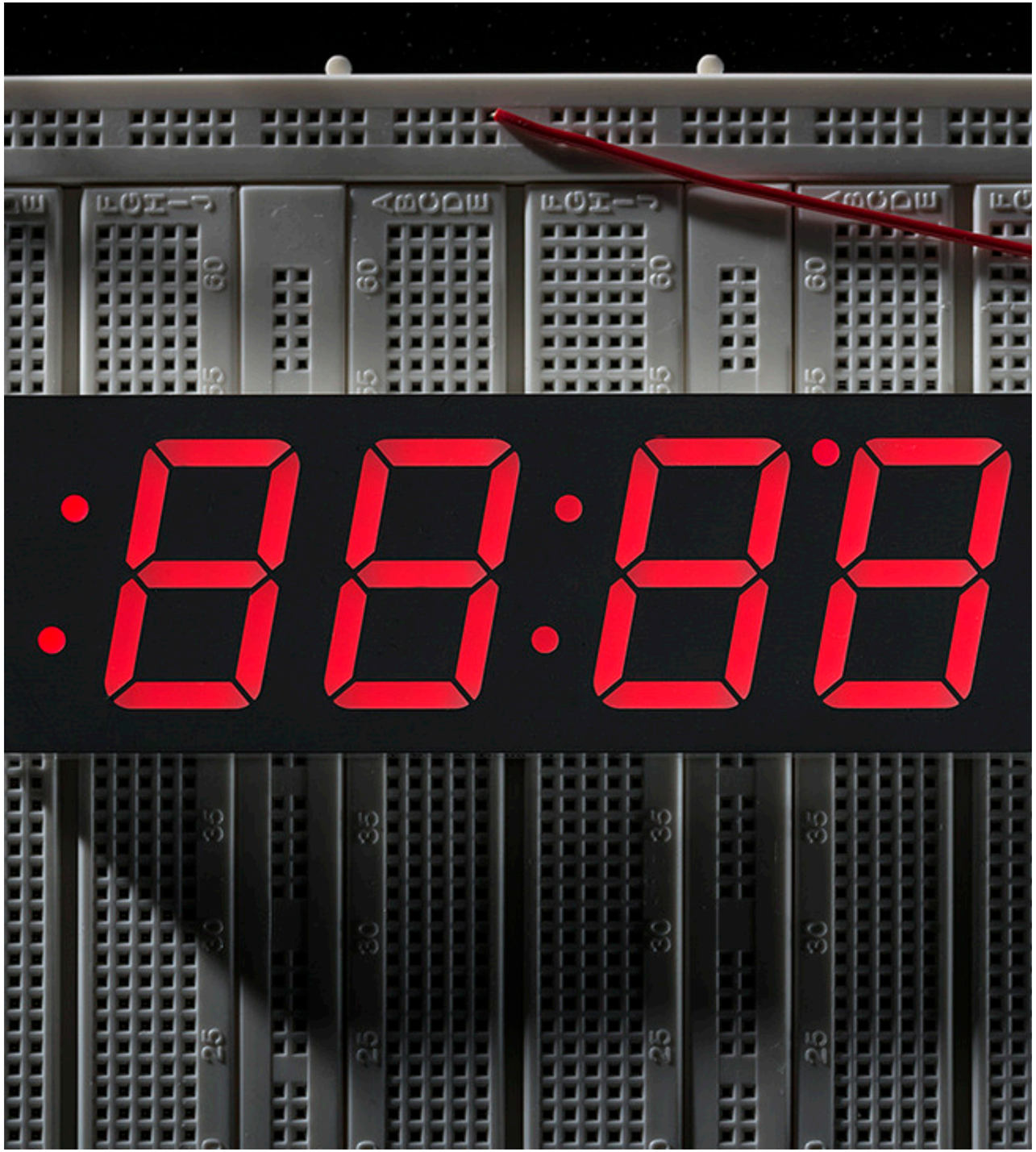


E-ink / E-Paper

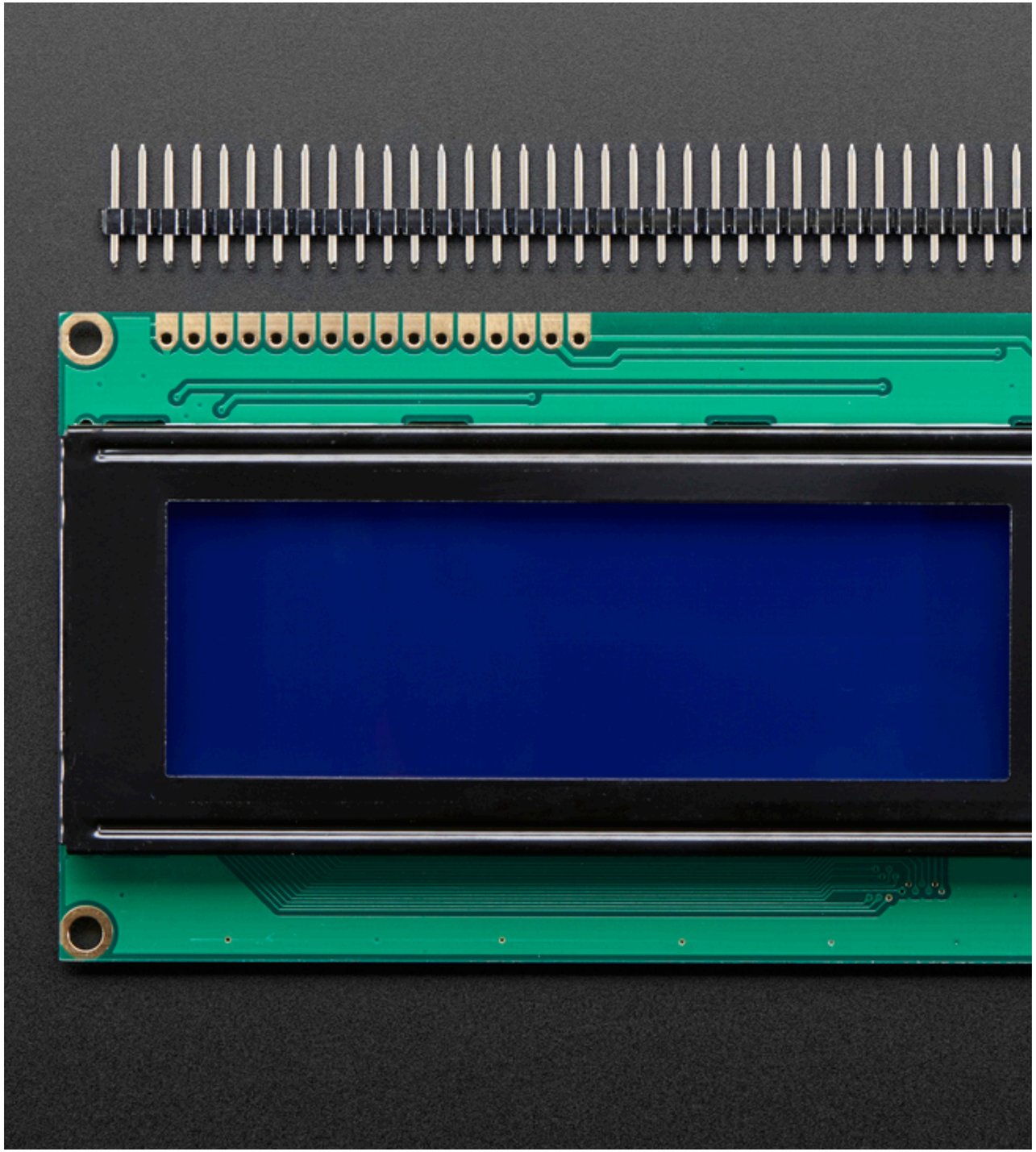


Thermodrucker

Segmentanzeige

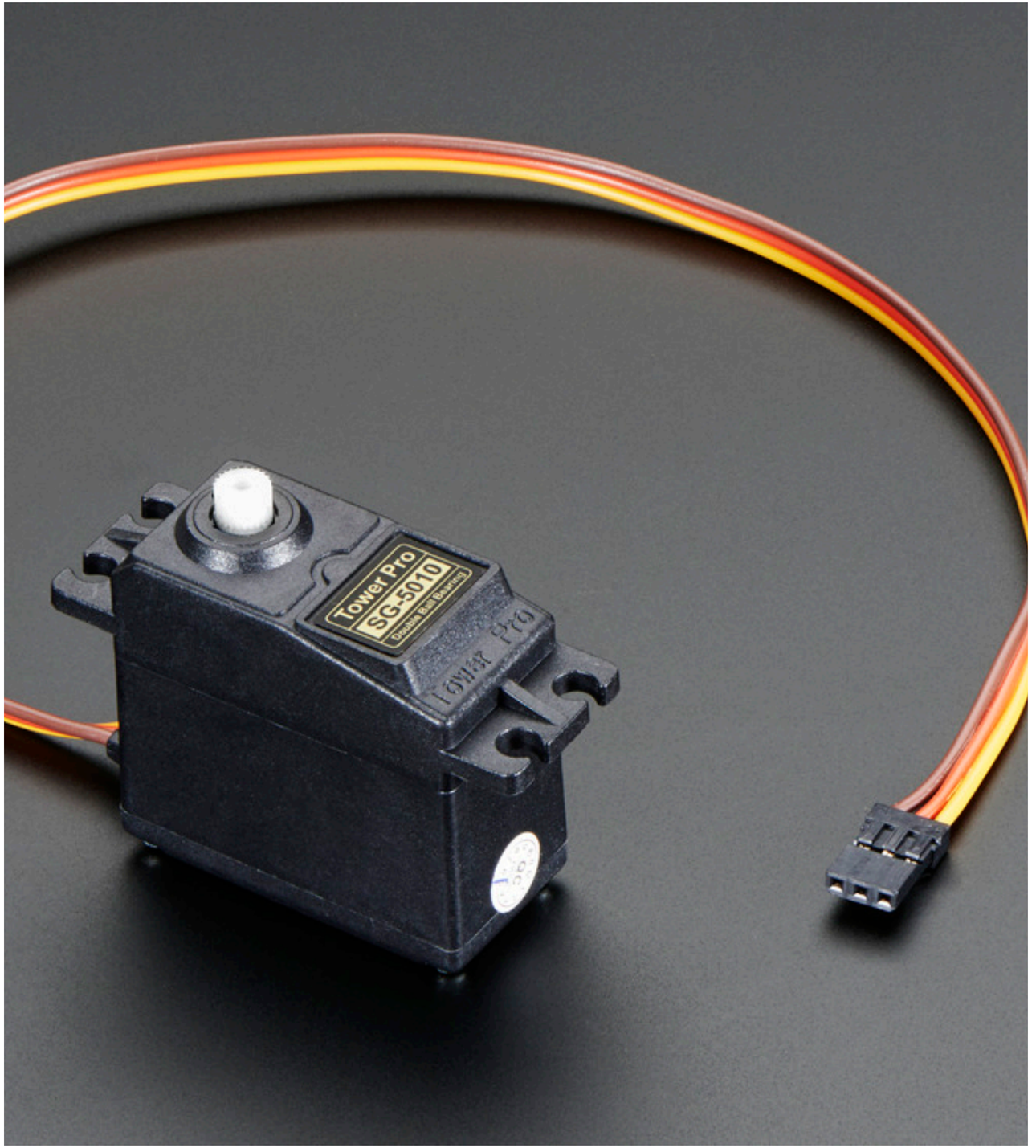


LCD



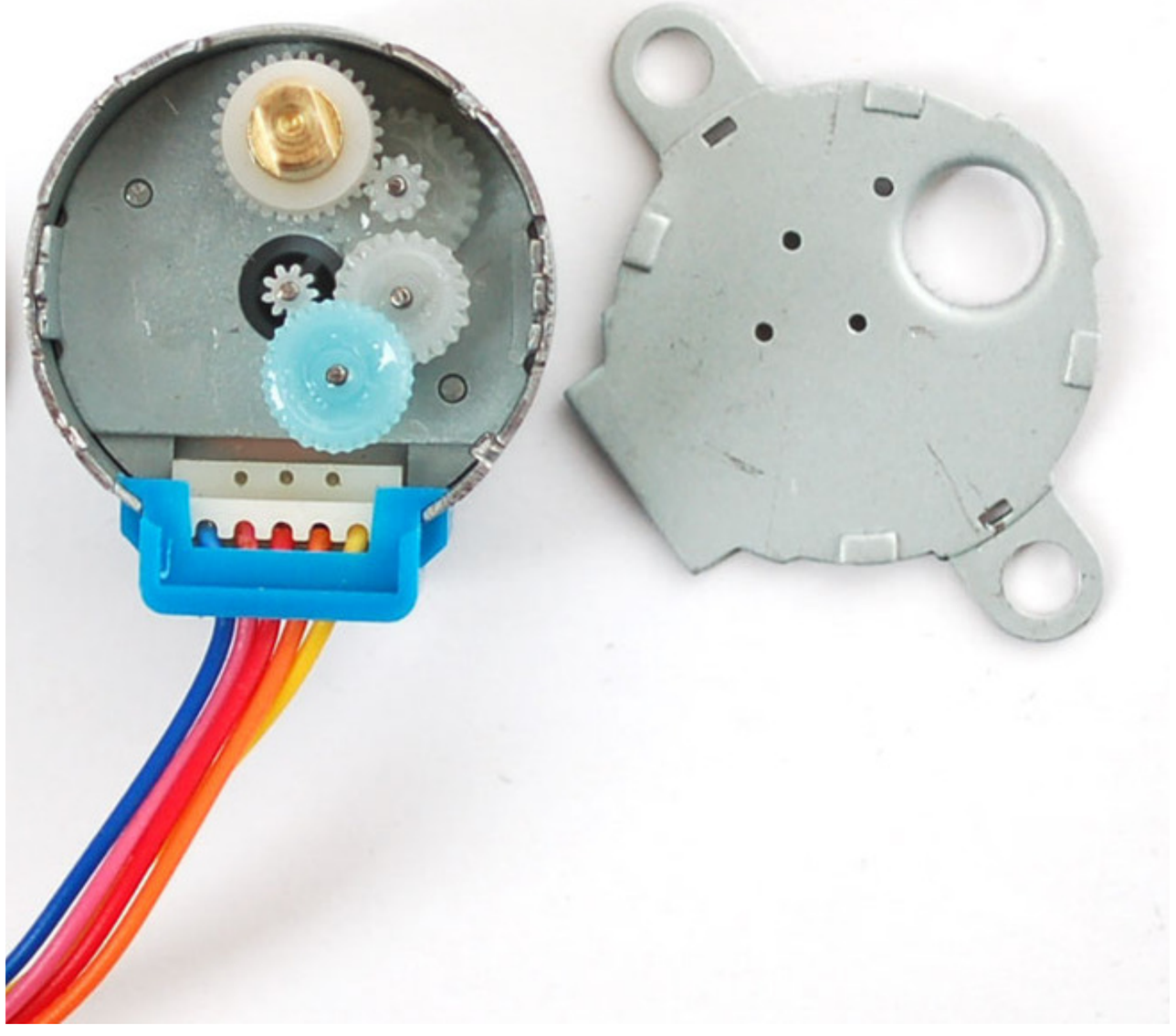
DC Motor

Servomotor



Schrittmotor

Schrittmotor



Schrittmotor

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

Arduino Code

```
int ledPin = 9;

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

void loop() {
  analogWrite(ledPin, 255);
  delay(50);
  analogWrite(ledPin, 125);
  delay(50);
  analogWrite(ledPin, 0);
  delay(50);
}
```

Arduino Code

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(0);
  Serial.println(sensorValue);
}
```

Arduino Code

Steuert eine LED mit einem Lichtsensor:

– Arduino
– Breadboard
– Jumperkabel
– Widerstände (URI)
– Photowiderstand
– PWM

Aufgabe 3 – Bis 15.11.2019

1. Plant euren Aufbau in Fritzing
2. Zeichnet den Schaltplan per Hand auf
3. Dokumentiert:
– Fotos: Echter Aufbau + Schaltplan
– Screenshots: Fritzing
– Code als *.ino Datei (Arduino Dateiformat)

Nutzt die Arduino Referenz und das Tutorium

Aufgabe 3 – Bis 15.11.2019

Fragen?