

**WIR BAUEN EIN  
SMART HOME**



Deutsche bleiben skeptisch im Bezug auf ...  
stadt bromerhaven.de



Was ist ein Smart Home? Geräte, Systeme ...  
homeandamart.de



▷ Zur IFA 2017: iHaus bringt Smart Ho...  
prosoportal.de



Eir Smart-Home-System bietet Komfort und Sicher...  
hurra wir bauen.de



Smart Home: Acht Starterkits im Vergleich - con...  
sonneat.de



Smart Home | Gigaset  
gigaset.com



Smart Home – Bausparkasse Schwäbisch Hall  
schwaebisch-hall.de



Router, NAS & Co. für Smart-Home: Das müs...  
pc-magazin.de



smart home – Elektrofuchs Leipzig  
elektro-fuchs.com



Funkstandards im Smart Home (Teil 1): Woran...  
trendblog.euronics.de



Smart Home | Wohnfreude  
wohnfreude.de



Versicherungen: Vorteil für Smart-Home-Kunden - Zukun...  
zukunftskunde.de



Lohnt sich Smart Home um Heizkosten zu spar...  
mehr-talk.de



Smart Home Systeme | ELV-Elektronik  
elv.de



Smart Home – Intelligent Wohnen | Allianz  
allianz.de



Smart Home - Anlegen in Immobilien  
anlegen-in-immobilien.de



We Are All Connected: IoT And The Rise Of The G...  
verifyrecruitment.com



Smart Home: Tedo setzt bei neuen Thermosta...  
golem.de

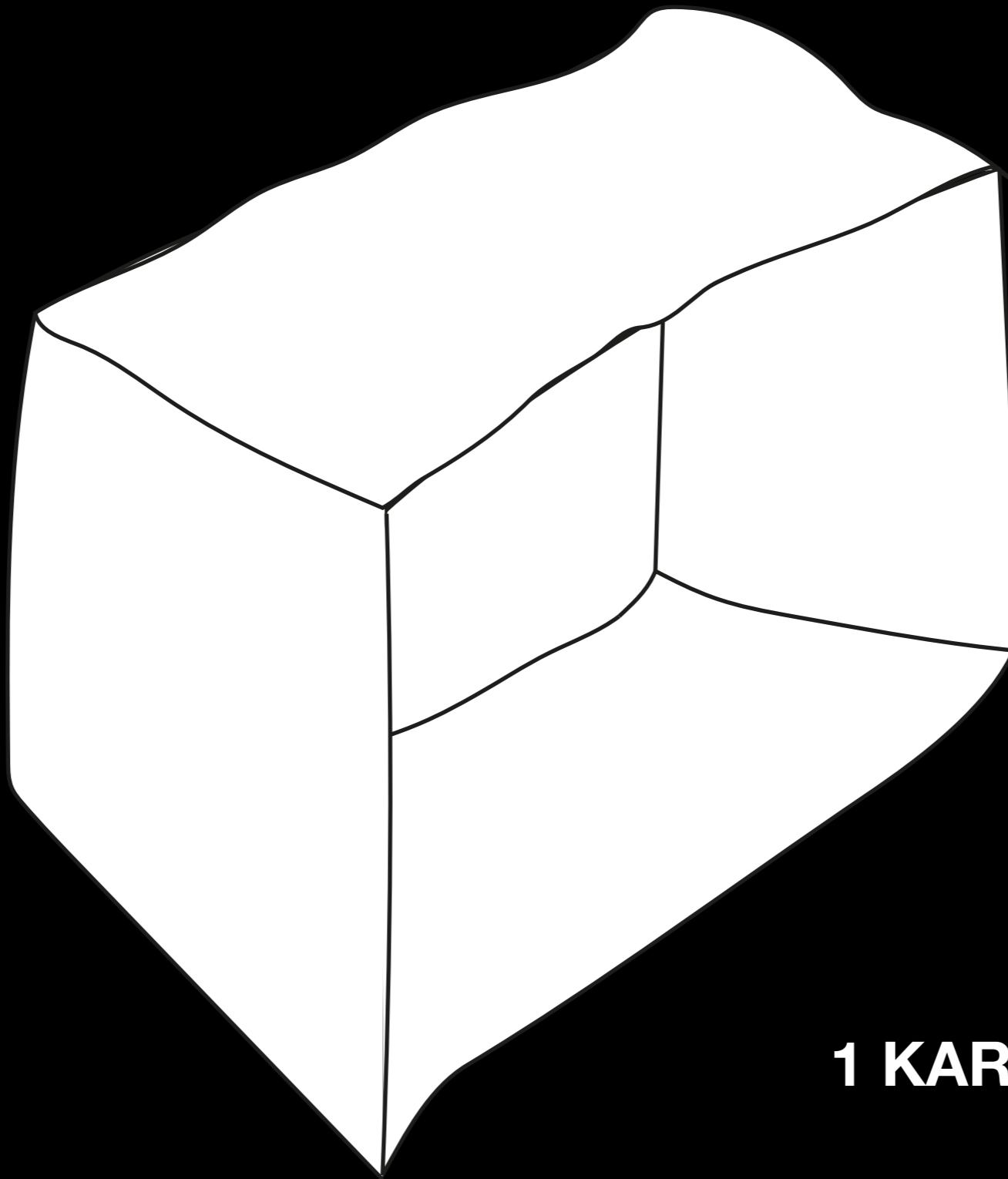


The Nest timeline: 7 years of ram...  
cnet.com



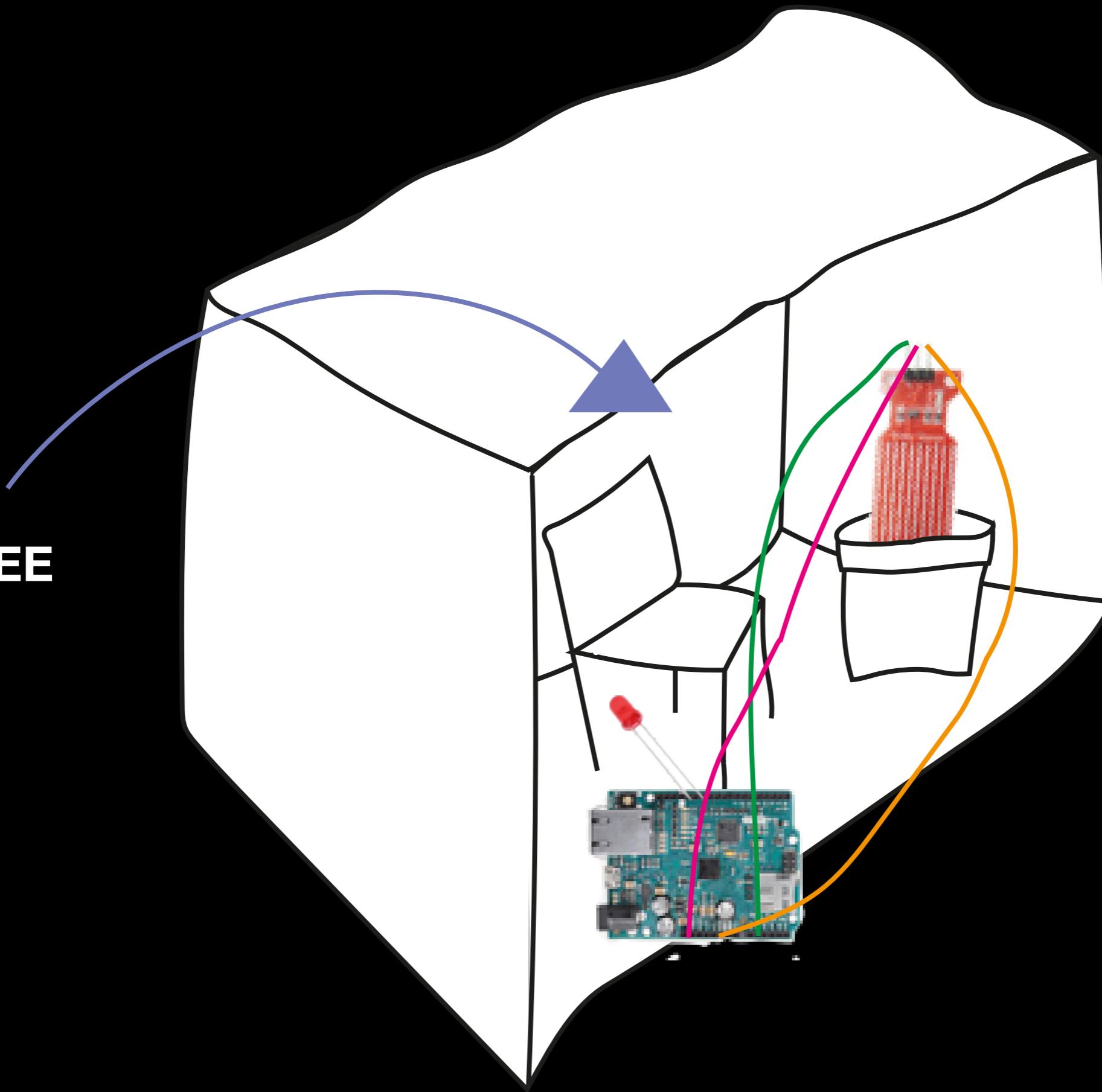


**Buster Keaton – The Electric House**  
**1922**

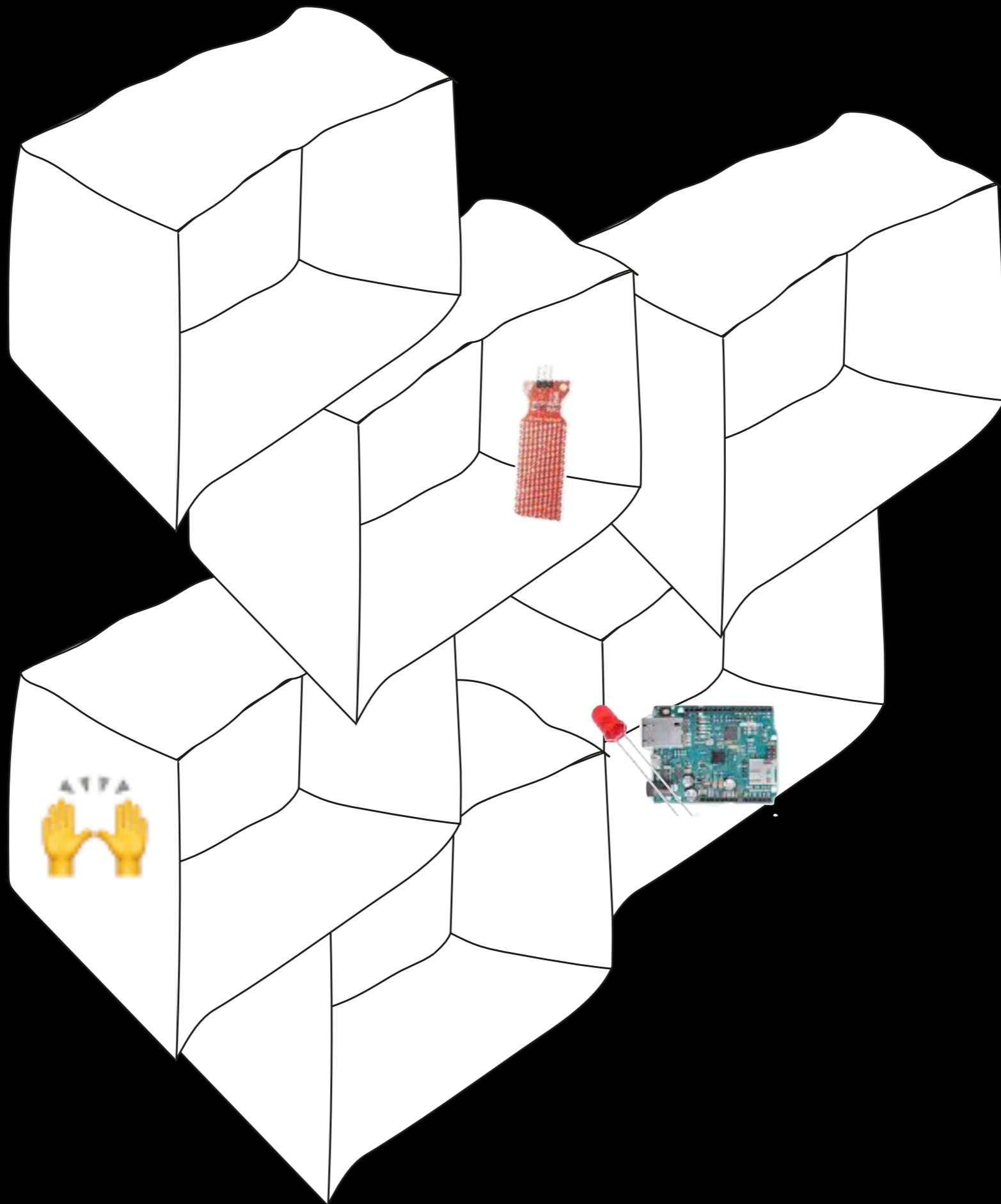


**1 KARTON / TEAM**

**IDEE**



**YEAH**





**Ana Serrano – Cartonlandia**





**11:30 – 13 Uhr**

## **Einführung Arduino**

- Generelles**
- Programmierstruktur**
- Sensorenkunde**

**13 – 14 Uhr**

## **Mittagessen**

**14 – 16 Uhr**

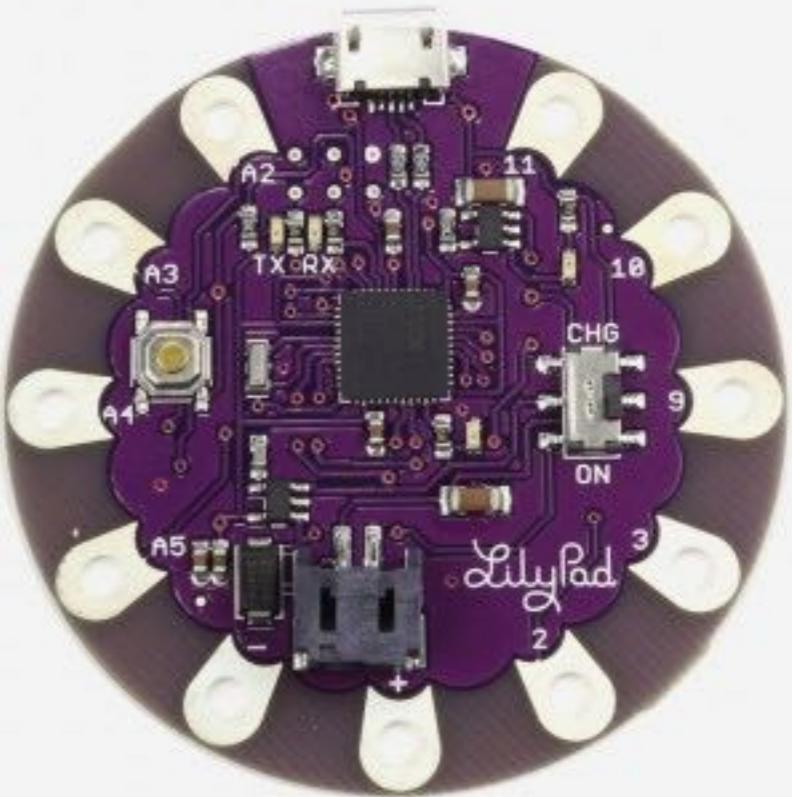
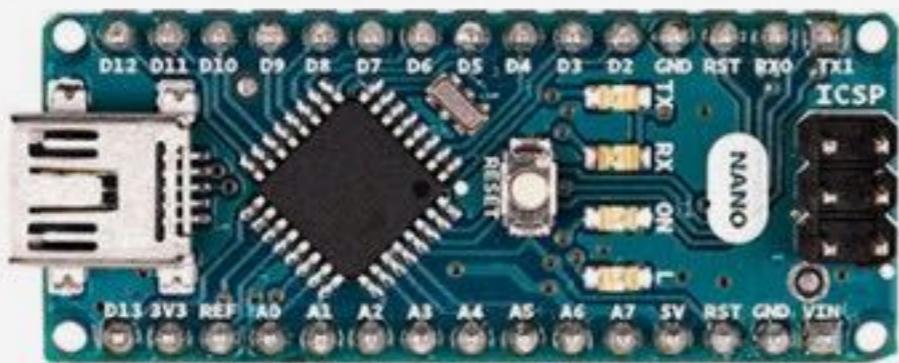
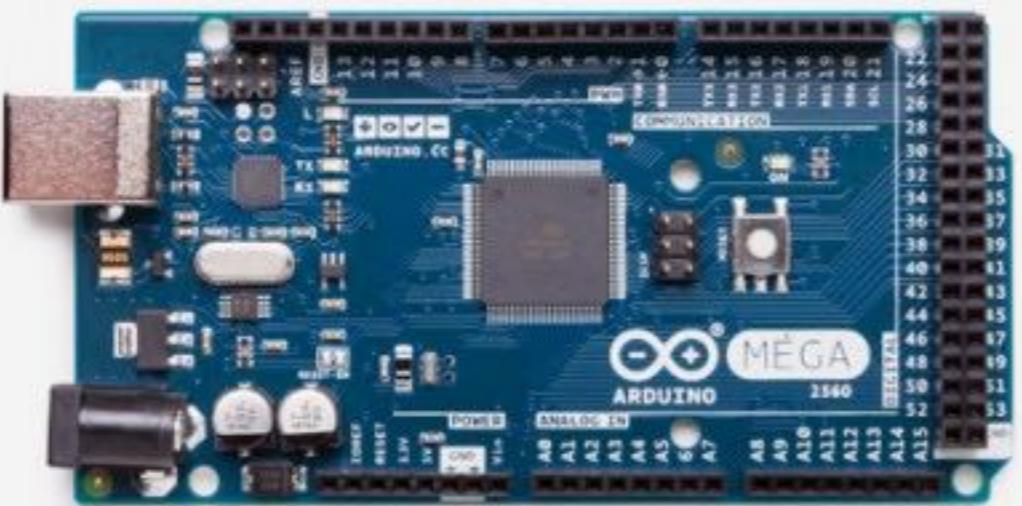
## **Teamwork: Programmieren, Stecken, Bauen**

**16 – 17 Uhr**

## **Richtfest und Präsentation**

# **WAS IST ARDUINO?**





rotary\_resistor | Arduino 1.6.9

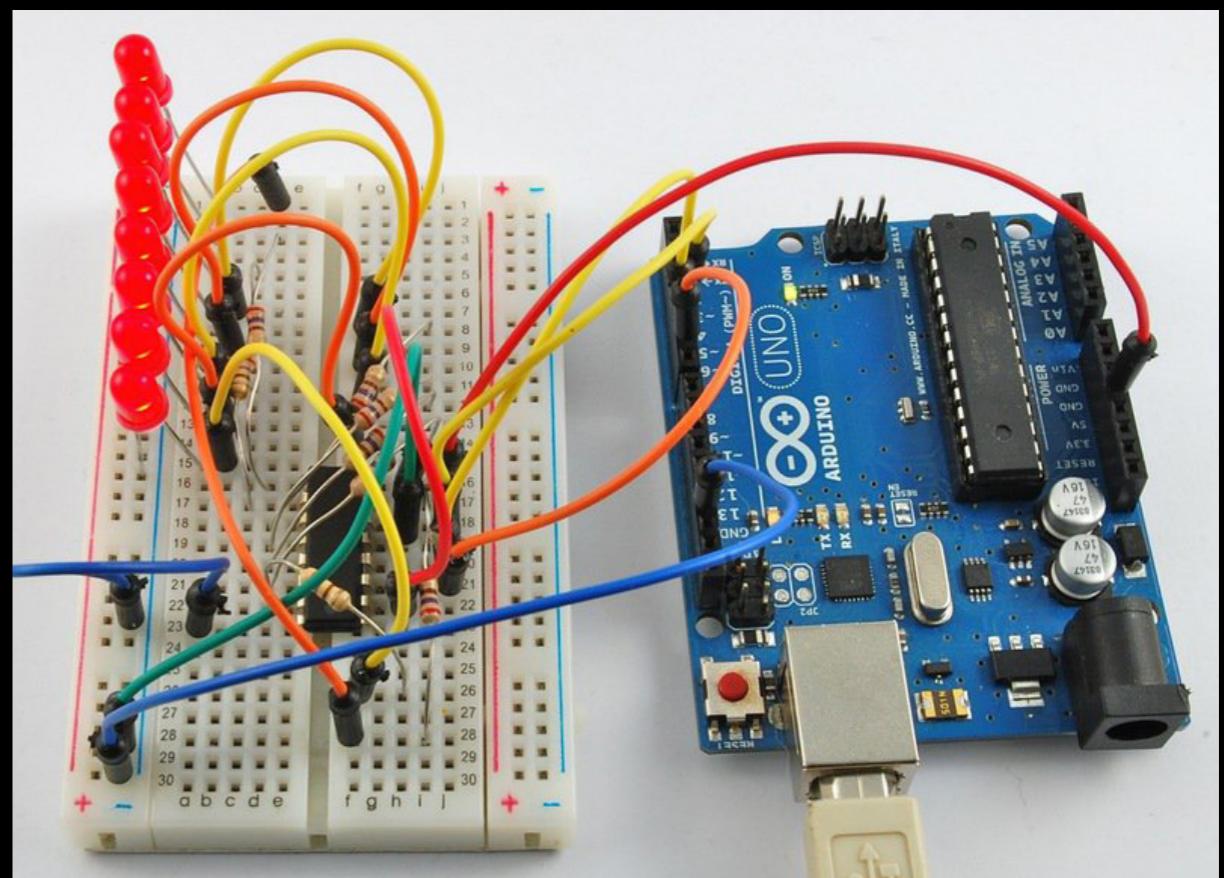
```
/*
 * code snippet: rotary resistor incl. smoothing
 */

int controlChange = 176; // MIDI Ch 1
int controllerNumber = 21;
int controllerValue = 0;

#define potiPin A0 // defines the pin A0 as an analog input
int potiValue = 0;
int prePotiValue = 0;
int potiSmooth = 0;

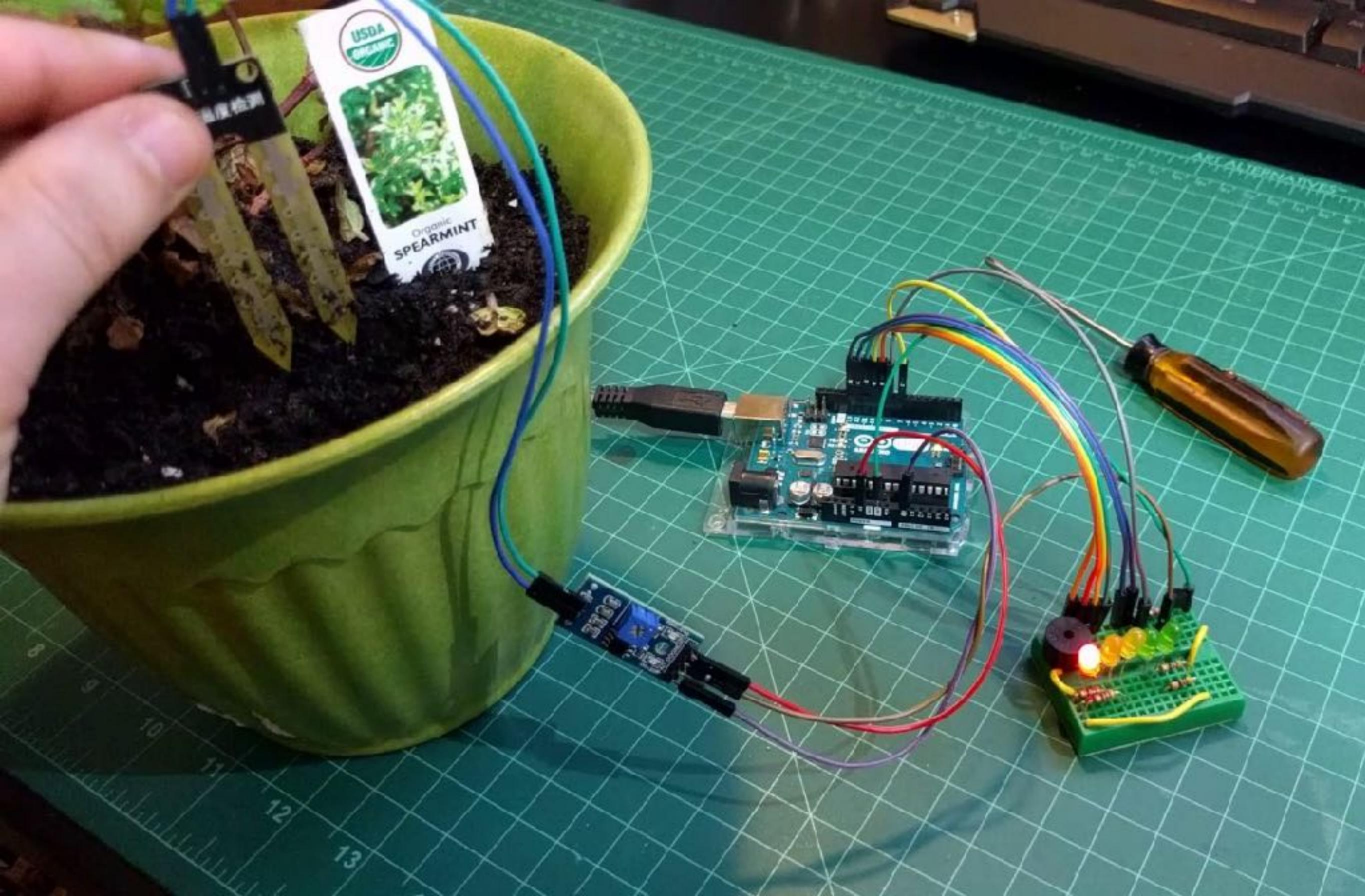
void setup() {
  Serial.begin(9600);
}

void loop() {
  potiSmooth = 0.6*potiSmooth + 0.4*analogRead(potiPin); // smoothing
  potiValue = map(potiSmooth, 5, 395, 0, 127); // maps the faderSmooth
}
```



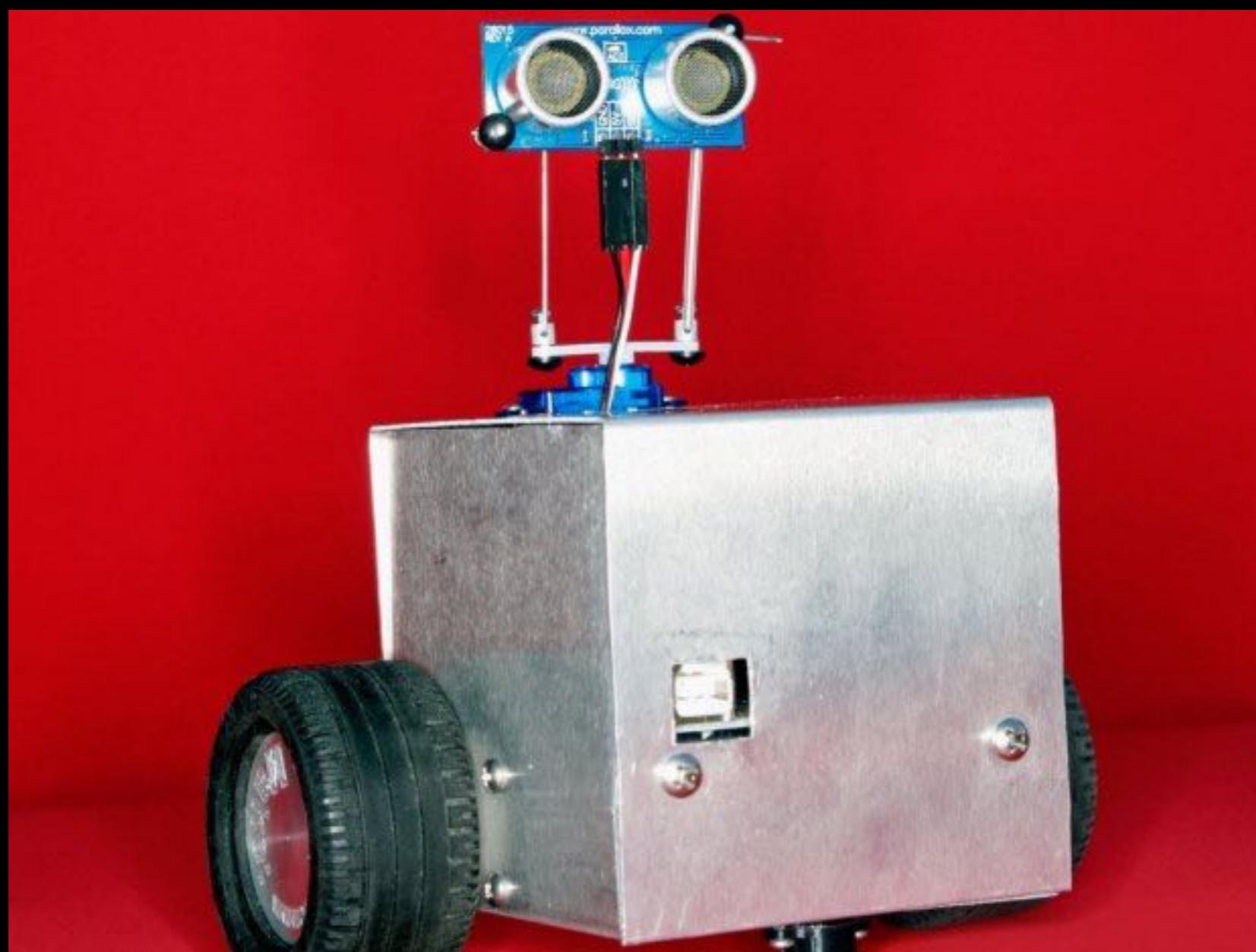
Arduino/Genuino Uno auf COM1

**WAS KANN ICH DAMIT MACHEN?**



Zachary Nall – Arduino Soil Moisture Detector





**WIE KANN ICH DAS MACHEN?**

**DOWNLOAD NOW**

**ARDUINO.CC**



**FRITZING.ORG**

**cmd ⌘ + C**

**cmd ⌘ + V**

---

**STRG + C**

**STRG + V**

Arduino Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch\_oct29d | Arduino 1.8.8

sketch\_oct29d.ino

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

A T

7944 - 07.08.2019 10:28:25 0.0000000000000000

# Arduino Programming Cheat Sheet

Primary source: Arduino Language Reference  
<http://arduino.cc/en/Reference/>

## Structure & Flow

### Basic Program Structure

```
void setup() {
    // Runs once when sketch starts
}

void loop() {
    // Runs repeatedly
}
```

### Control Structures

```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
for (int i = 0; i < 10; i++) { ... }
break; // Exit a loop immediately
continue; // Go to next iteration
switch (var) {
    case 1:
        ...
        break;
    case 2:
        ...
        break;
    default:
        ...
}
return x; // x must match return type
return; // For void return type
```

### Function Definitions

```
<ret. type> <name>(<params>) { ... }
e.g. int double(int x) {return x*2;}
```

## Operators

### General Operators

- = assignment
- + add - subtract
- \* multiply / divide
- % modulo
- == equal to != not equal to
- < less than > greater than
- <= less than or equal to
- >= greater than or equal to
- && and || or
- ! not

### Compound Operators

- ++ increment
- decrement
- += compound addition
- = compound subtraction
- \*= compound multiplication
- /= compound division
- &= compound bitwise and
- |= compound bitwise or

### Bitwise Operators

- & bitwise and | bitwise or
- ^ bitwise xor ~ bitwise not
- << shift left >> shift right

### Pointer Access

- & reference: get a pointer
- \* dereference: follow a pointer

## Variables, Arrays, and Data

### Data Types

boolean	true   false
char	-128 - 127, 'a'-'\$' etc.
unsigned char	0 - 255
byte	0 - 255
int	-32768 - 32767
unsigned int	0 - 65535
word	0 - 65535
long	-2147483648 - 2147483647
unsigned long	0 - 4294967295
float	-3.4028E+38 - 3.4028E+38
double	currently same as float
void	i.e., no return value

### Strings

```
char str1[8] =
{'A','r','d','u','i','n','o','\0'};
// Includes \0 null termination
char str2[8] =
{'A','r','d','u','i','n','o'};
// Compiler adds null termination
char str3[] = "Arduino";
char str4[8] = "Arduino";
```

### Numeric Constants

123	decimal
0b01111011	binary
0173	octal - base 8
0x7B	hexadecimal - base 16
123U	Force unsigned
123L	Force long
123UL	Force unsigned long
123.0	Force Floating point
1.23E5	1.23*10^5 = 1230000

### Qualifiers

static	persists between calls
volatile	in RAM (nice for ISR)
const	read-only
PROGMEM	in flash

### Arrays

```
int myPins[] = {2, 4, 8, 3, 6};
int myInts[6]; // Array of 6 ints
myInts[0] = 42; // Assigning first
                // index of myInts
myInts[6] = 12; // FRROR! Indexes
                // are 0 through 5
```

## Built-in Functions

### Pin Input/Output

Digital I/O - pins 0-13 A0-A5
pinMode(pin,
[INPUT, OUTPUT, INPUT_PULLUP])
int digitalRead(pin)
digitalWrite(pin, [HIGH, LOW])

### Analog In

int analogRead(pin)
analogReference([DEFAULT, INTERNAL, EXTERNAL])

### PWM Out

analogWrite(pin, value)
-------------------------

### Advanced I/O

tone(pin, freq_Hz)
tone(pin, freq_Hz, duration_ms)
noTone(pin)
shiftOut(dataPin, clockPin,             [MSBFIRST, LSBFIRST], value)
unsigned long pulseIn(pin,             [HIGH, LOW])

### Time

unsigned long millis()
// Overflows at 50 days
unsigned long micros()
// Overflows at 70 minutes
delay(msac)
delayMicroseconds(usec)

### Math

min(x, y) max(x, y) abs(x)
sin(rad) cos(rad) tan(rad)
sqrt(x) pow(base, exponent)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)

### Random Numbers

randomSeed(seed) // long or int
long random(max) // 0 to max-1
long random(min, max)

### Bits and Bytes

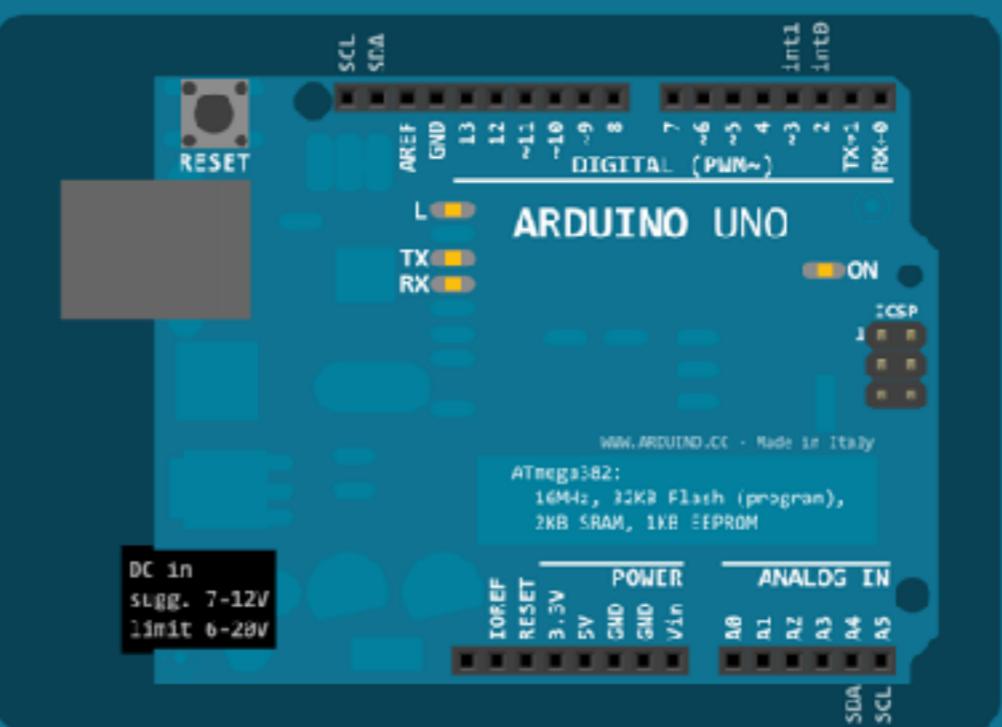
lowByte(x) highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn) // bitn: 0=LSB 7=MSB

### Type Conversions

char(val) byte(val)
int(val) word(val)
long(val) float(val)

### External Interrupts

attachInterrupt(interrupt, func,             [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupt)
interrupts()
noInterrupts()



## Libraries

**Serial** - comm. with PC or via RX/TX  
begin(long speed) // Up to 115200

end()  
int available() // #bytes available

int read() // -1 if none available  
int peek() // Read w/o removing

flush()  
print(data) println(data)

write(byte) write(char \* string)  
write(data, size)

SerialEvent() // Called if data ready

**SoftwareSerial.h** - comm. on any pin  
SoftwareSerial(rxPin, txPin)

begin(long speed) // Up to 115200

listen() // Only 1 can listen  
isListening() // at a time.

read, peek, print, println, write  
// Equivalent to Serial library

**EEPROM.h** - access non-volatile memory

byte read(addr)  
write(addr, byte)  
EEPROM[index] // Access as array

**Servo.h** - control servo motors

attach(pin, [min\_us, max\_us])  
write(angle) // 0 to 180  
writeMicroseconds(us) // 1000-2000; 1500 is midpoint  
int read() // 0 to 180  
bool attached()  
detach()

**Wire.h** - I<sup>2</sup>C communication

begin() // Join a master  
begin(addr) // Join a slave @ addr  
requestFrom(address, count)  
beginTransmission(addr) // Step 1  
send(byte) // Step 2  
send(char \* string)  
send(byte \* data, size)  
endTransmission() // Step 3  
int available() // #bytes available  
byte receive() // Get next byte  
onReceive(handler)  
onRequest(handler)

by Mark Liffiton

Adapted from:  
- Original: Gavin Smith  
- SVG version: Frederic Dufour  
- Arduino board drawing: Fritzing.org

Arduino Datei Bearbeiten Sketch Werkzeuge Hilfe

87 % (7) So, 21:43 Q



sketch\_oct29d.g

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

a T

794 von 1000 100 % 2470ms 0.0ms 0.3s

# **DIE ELEKTRISCHEN GRUNDGRÖSSEN**

$$U = R * I$$

**SPANNUNG**

**U**

**VOLT**

**WIDERSTAND**

**R**

**OHM**

**STROMSTÄRKE**

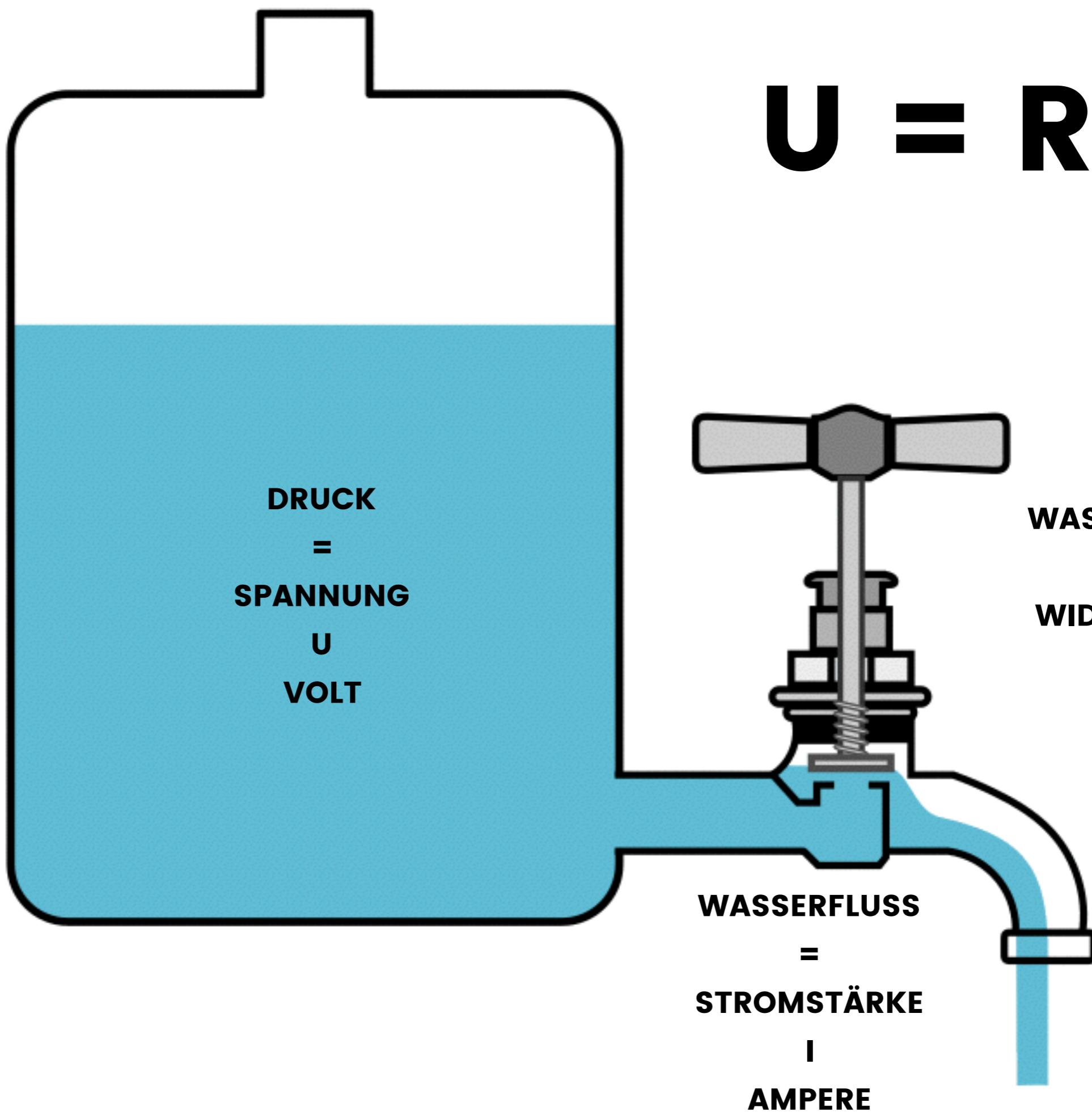
**I**

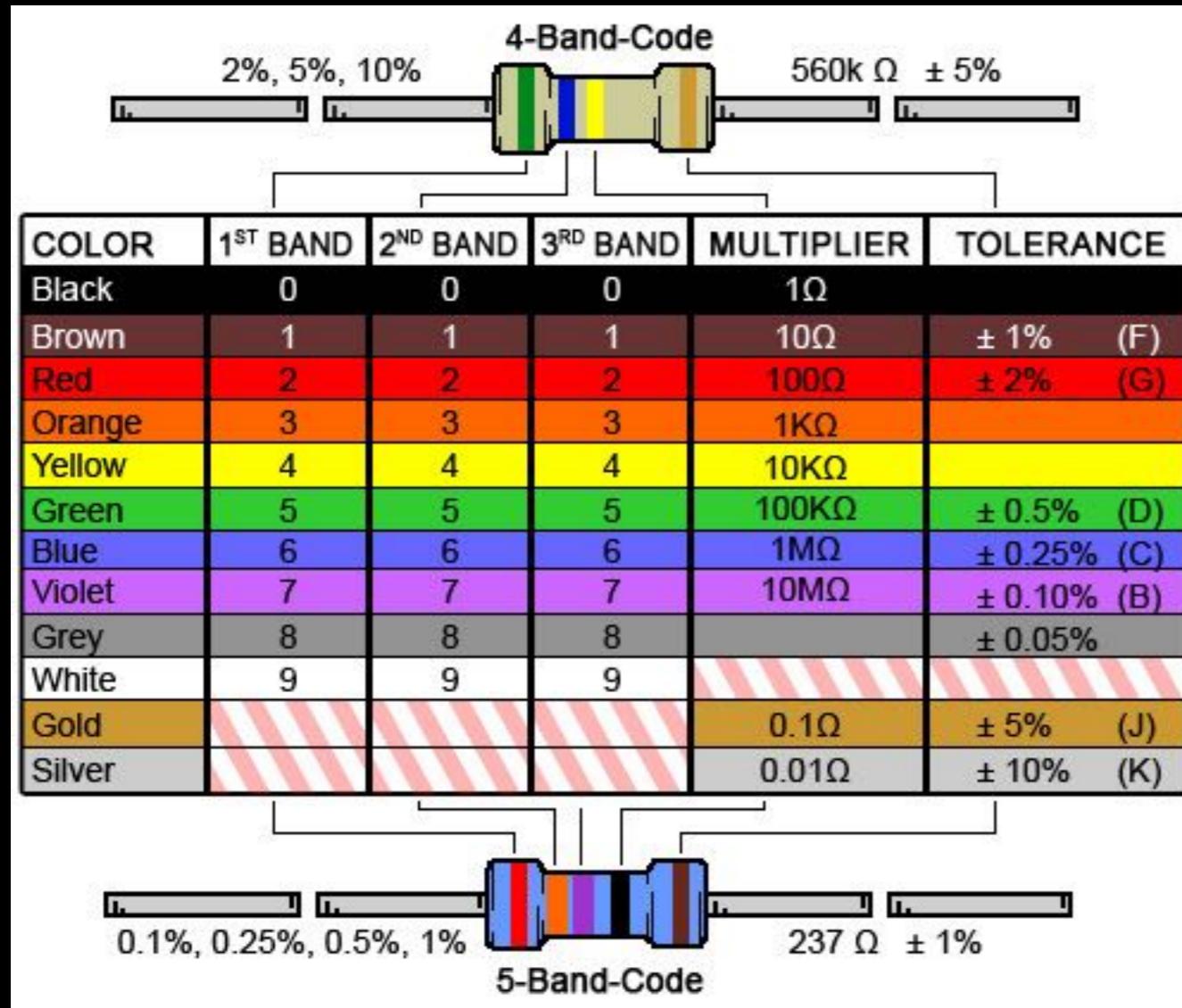
**AMPERE**

$$I = U/R$$

$$R = U/I$$

$$U = R * I$$





# ODER

# ONLINE-TOOL

# ODER



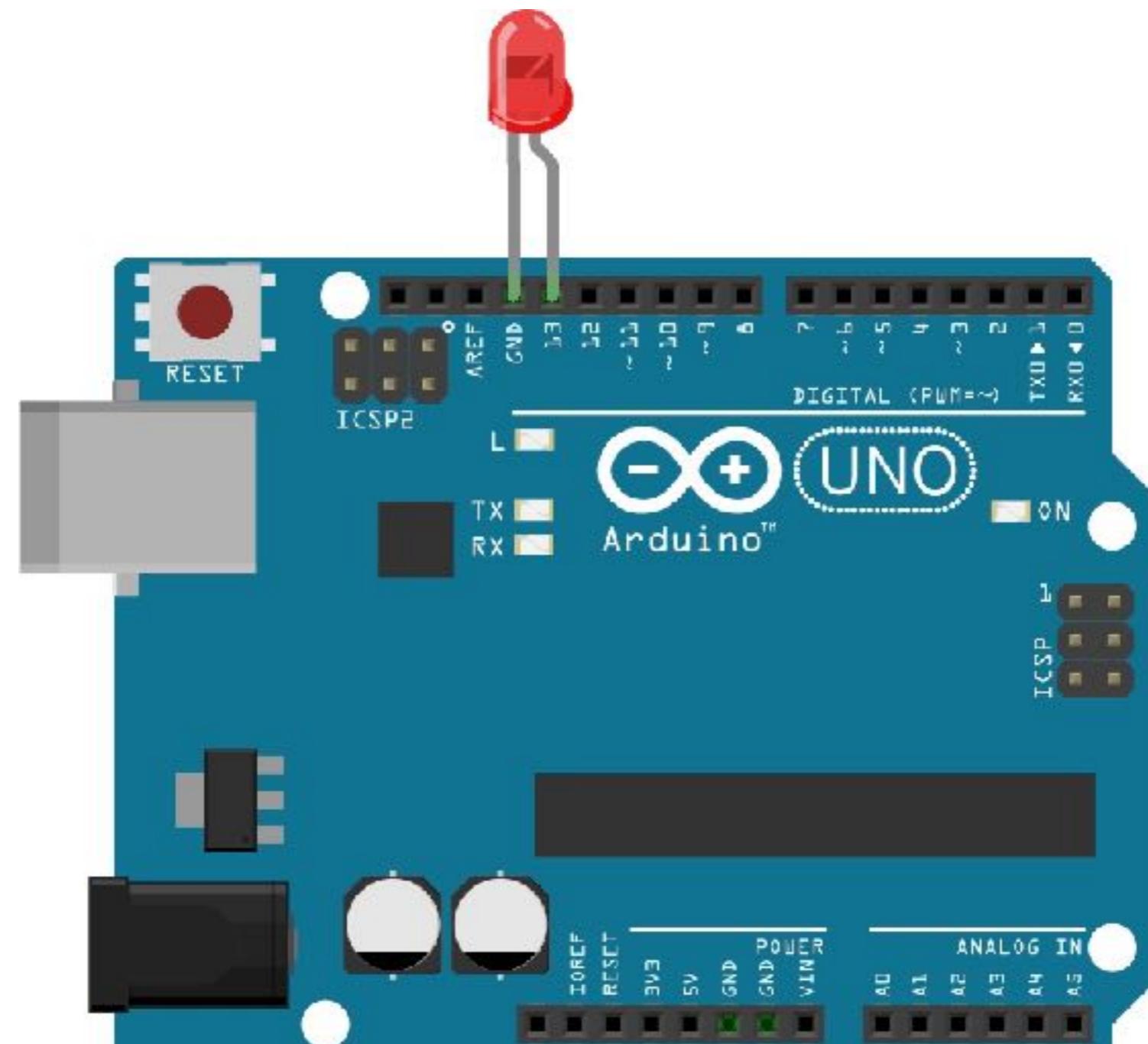




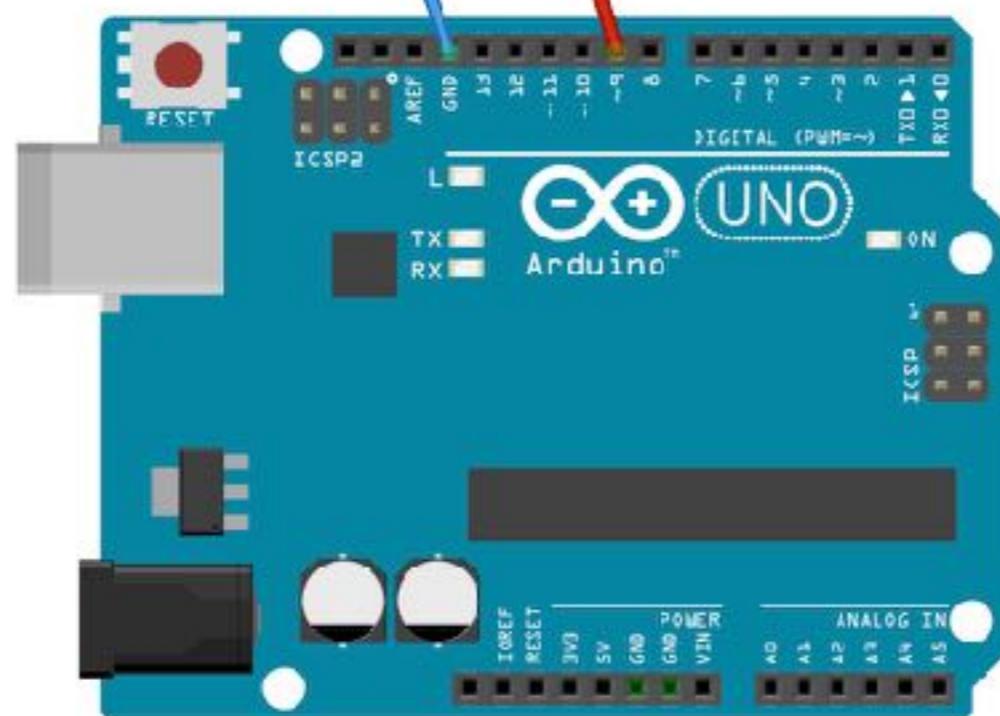
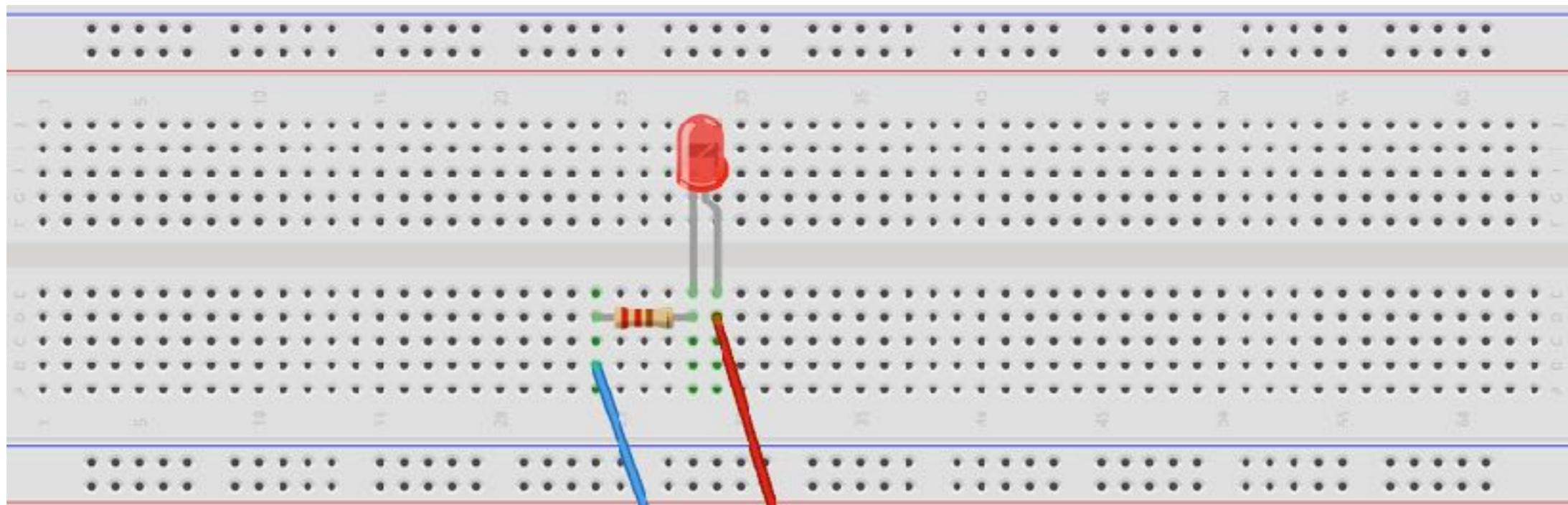
-

+

# BLINK

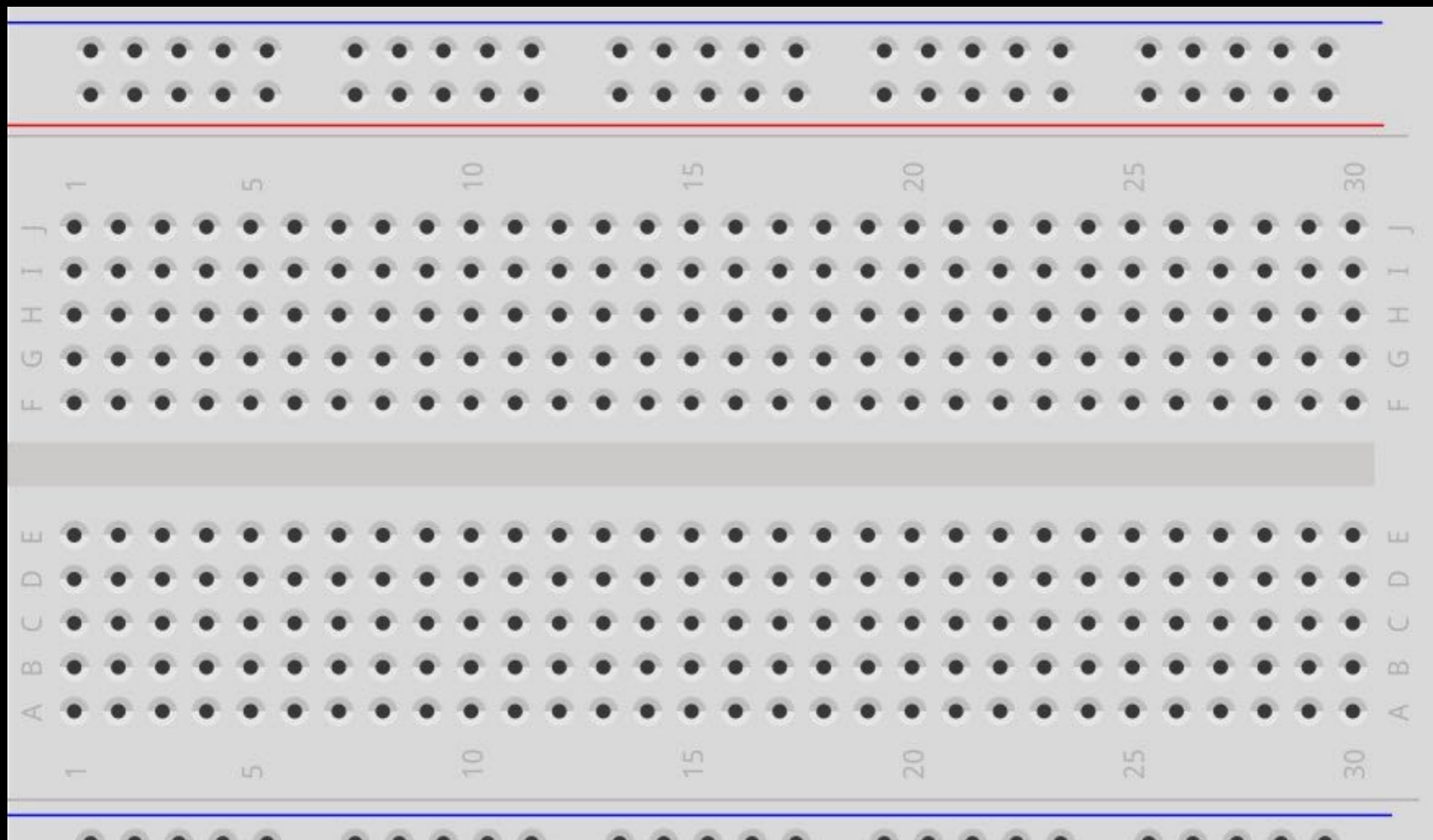


fritzing

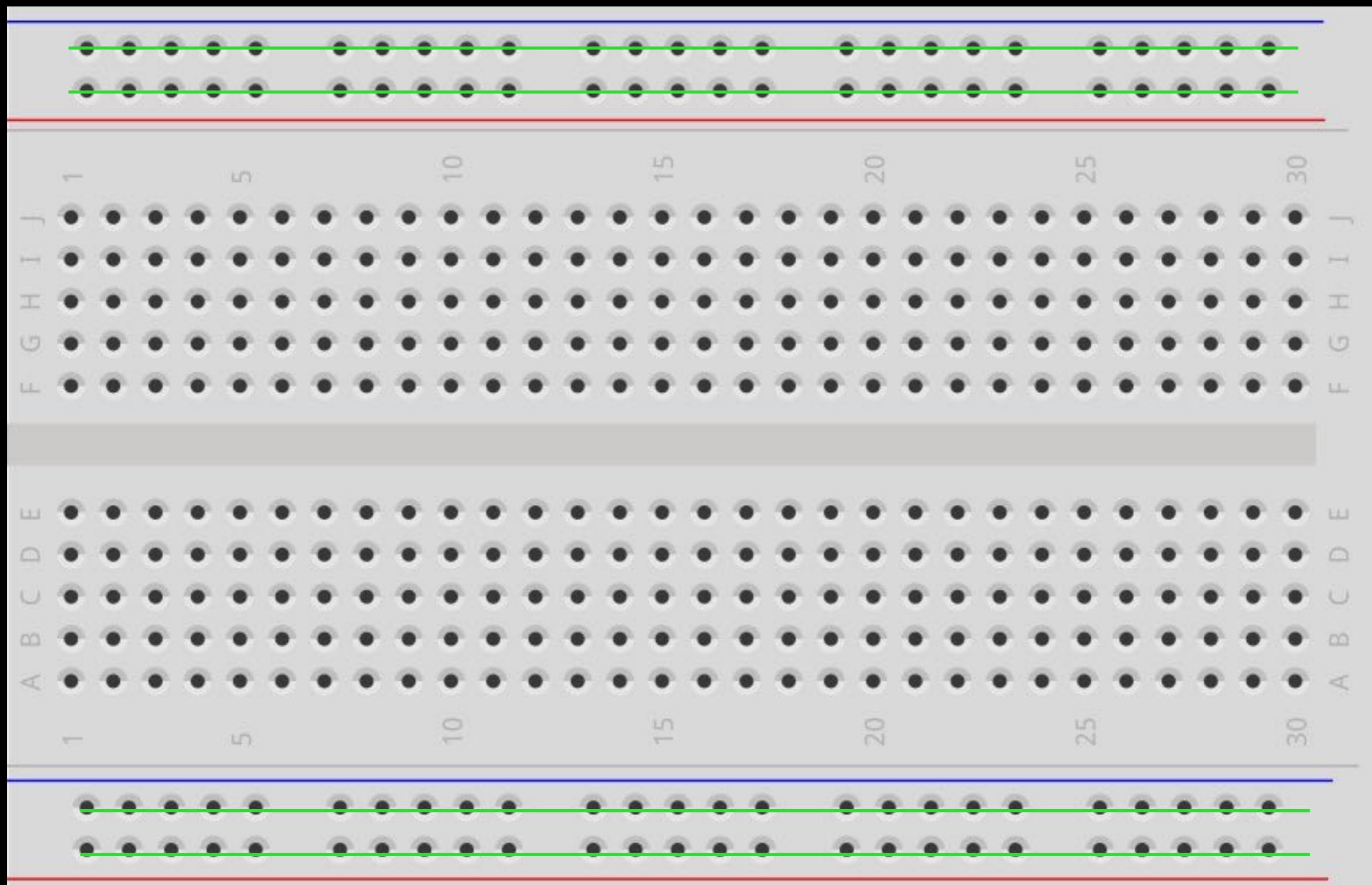


FADE

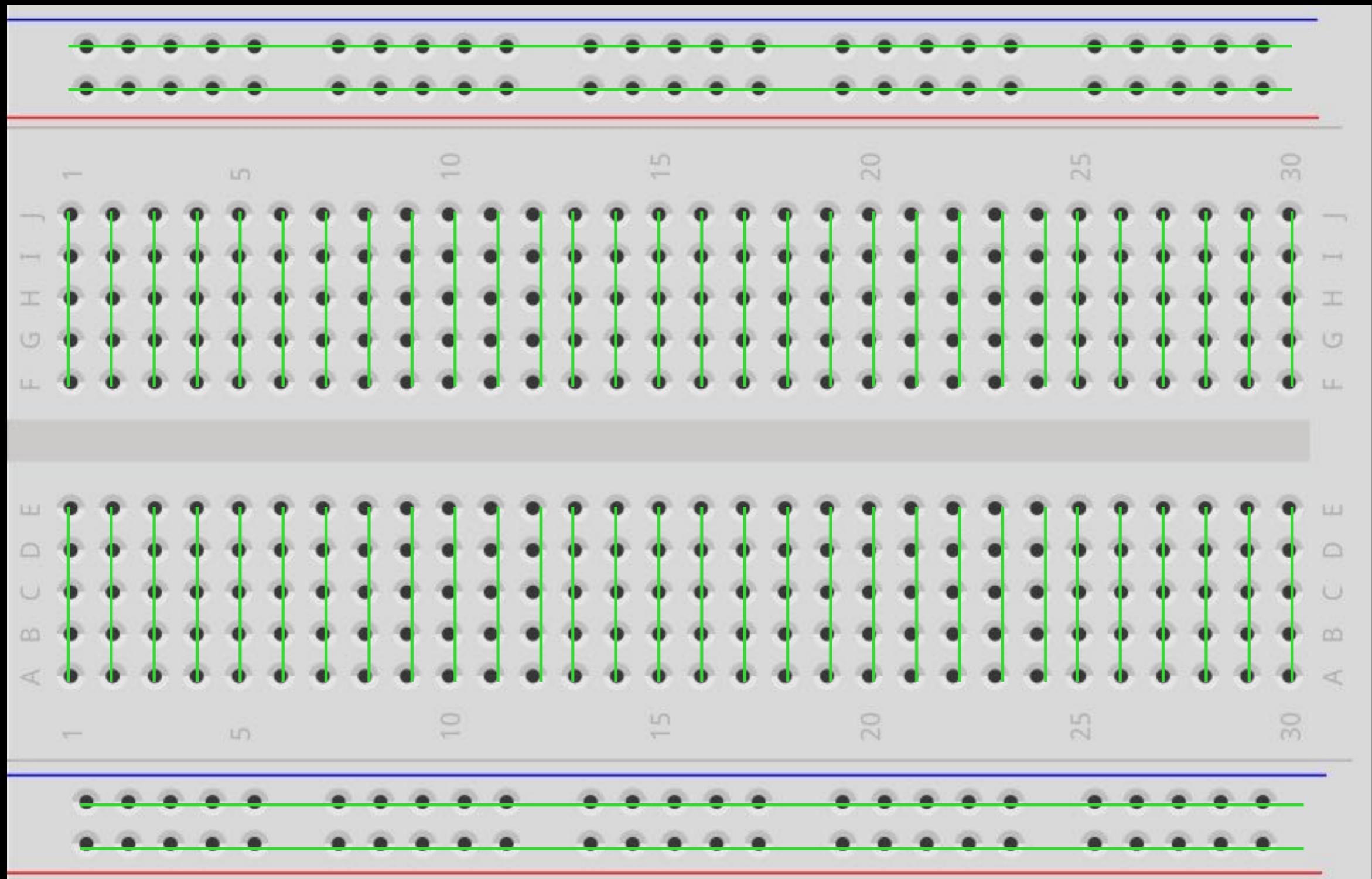
fritzing



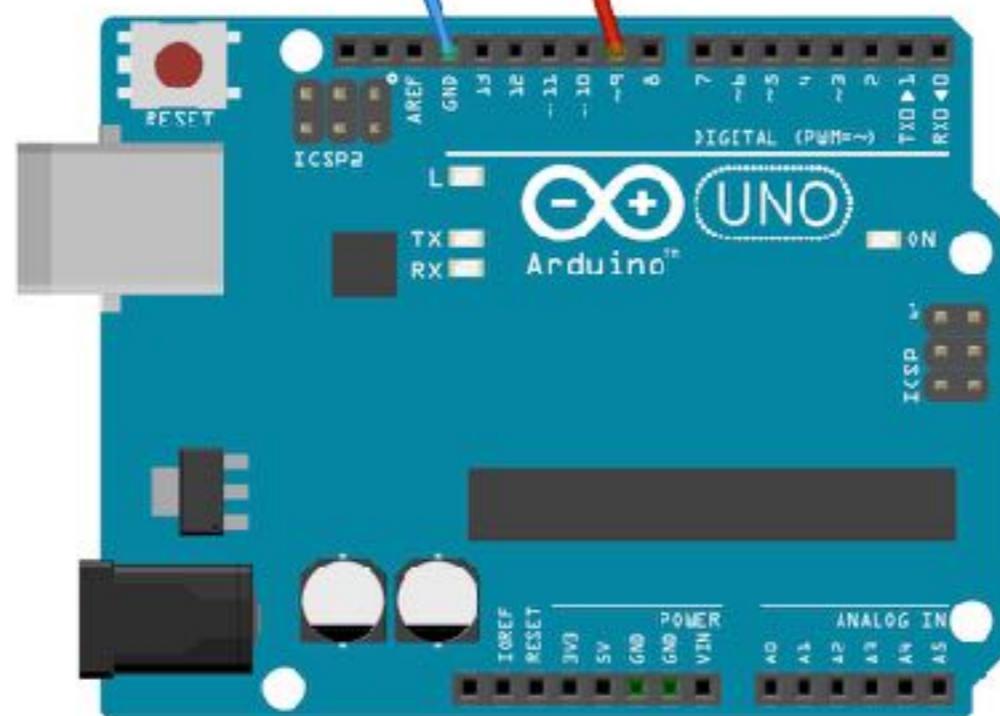
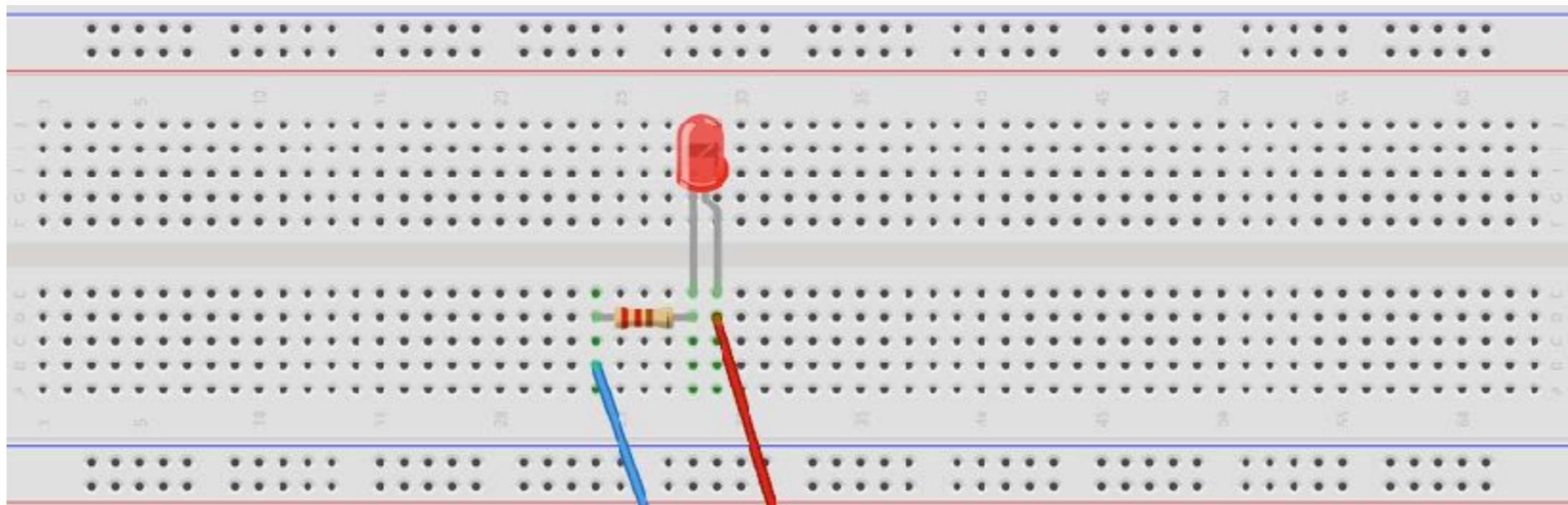
fritzing



fritzing



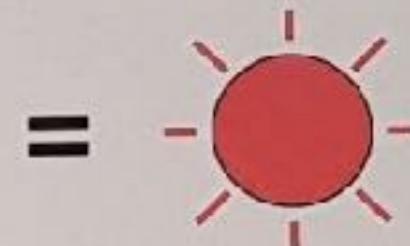
fritzing



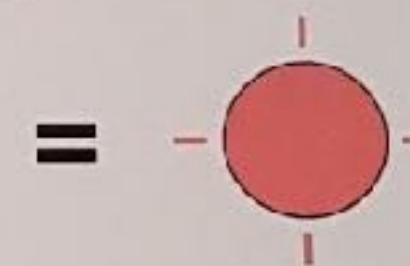
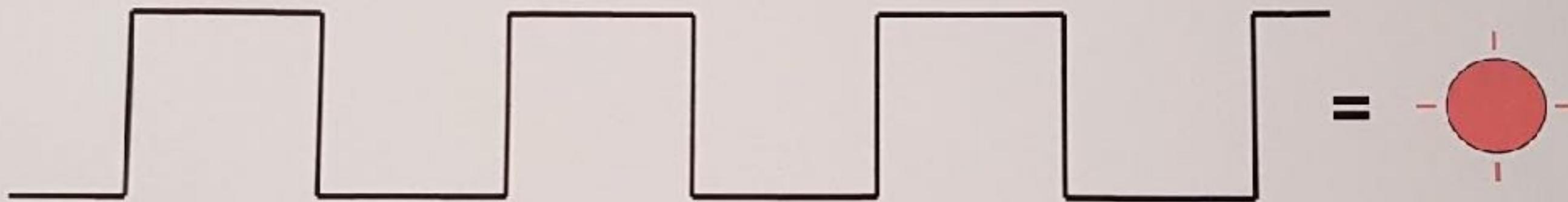
FADE

fritzing

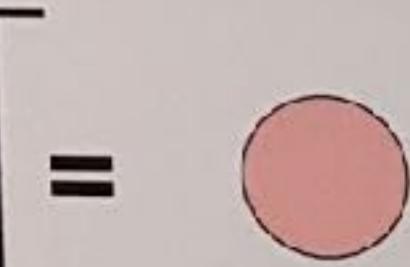
90% duty cycle



50% duty cycle

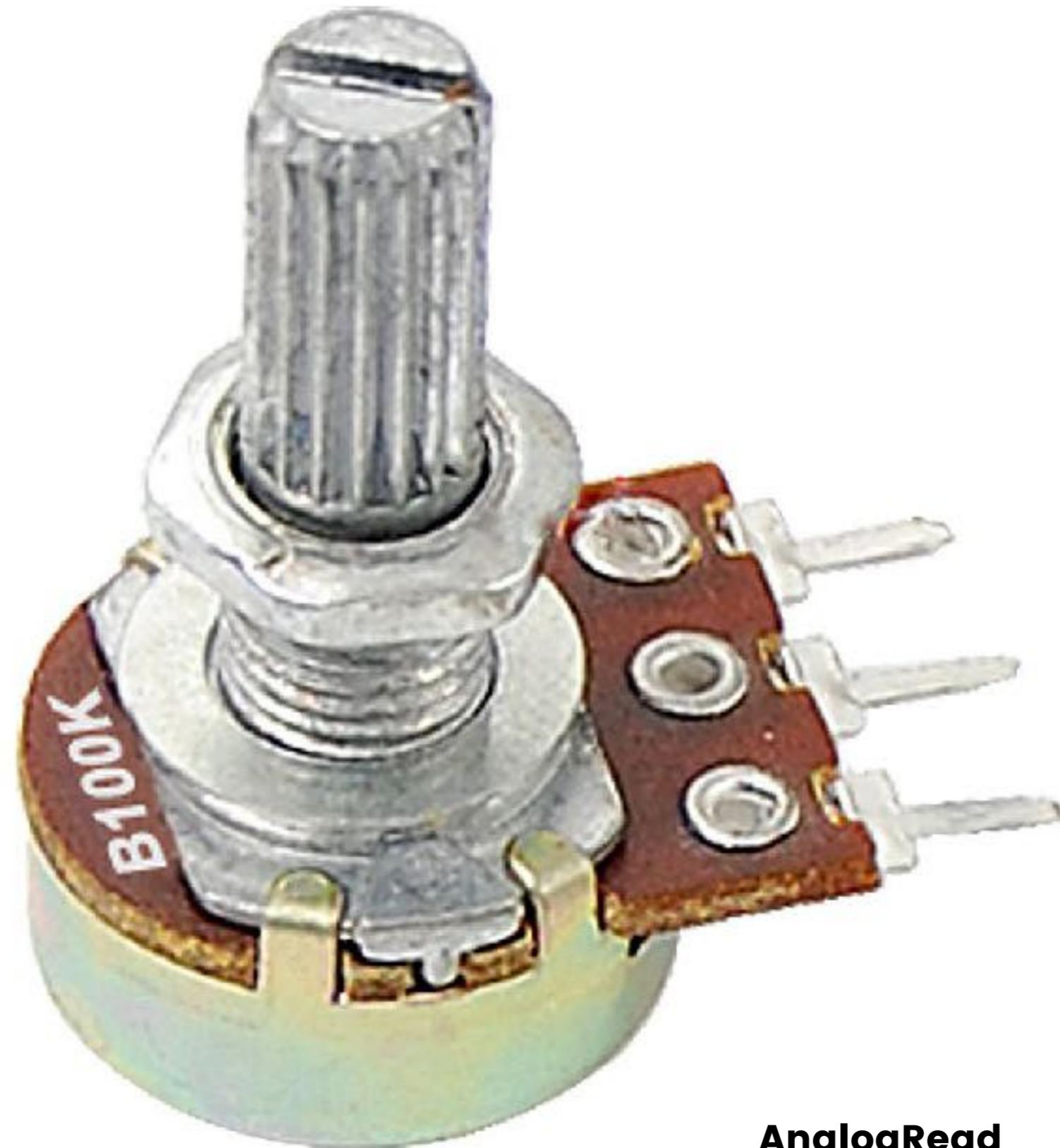


10% duty cycle



PWS

# POTI

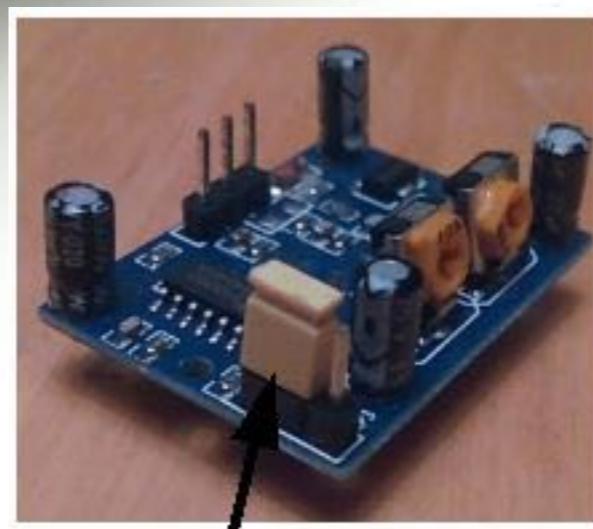


**AnalogRead**

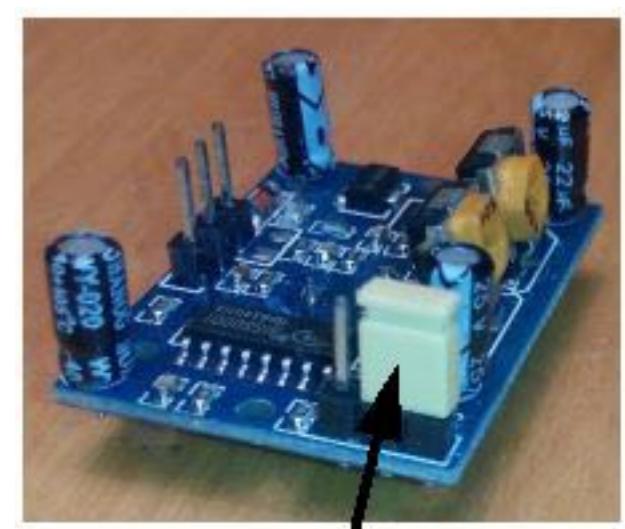
**AnalogInOutSerial**

# **INPUTS**

# BEWEGUNGSMELDER

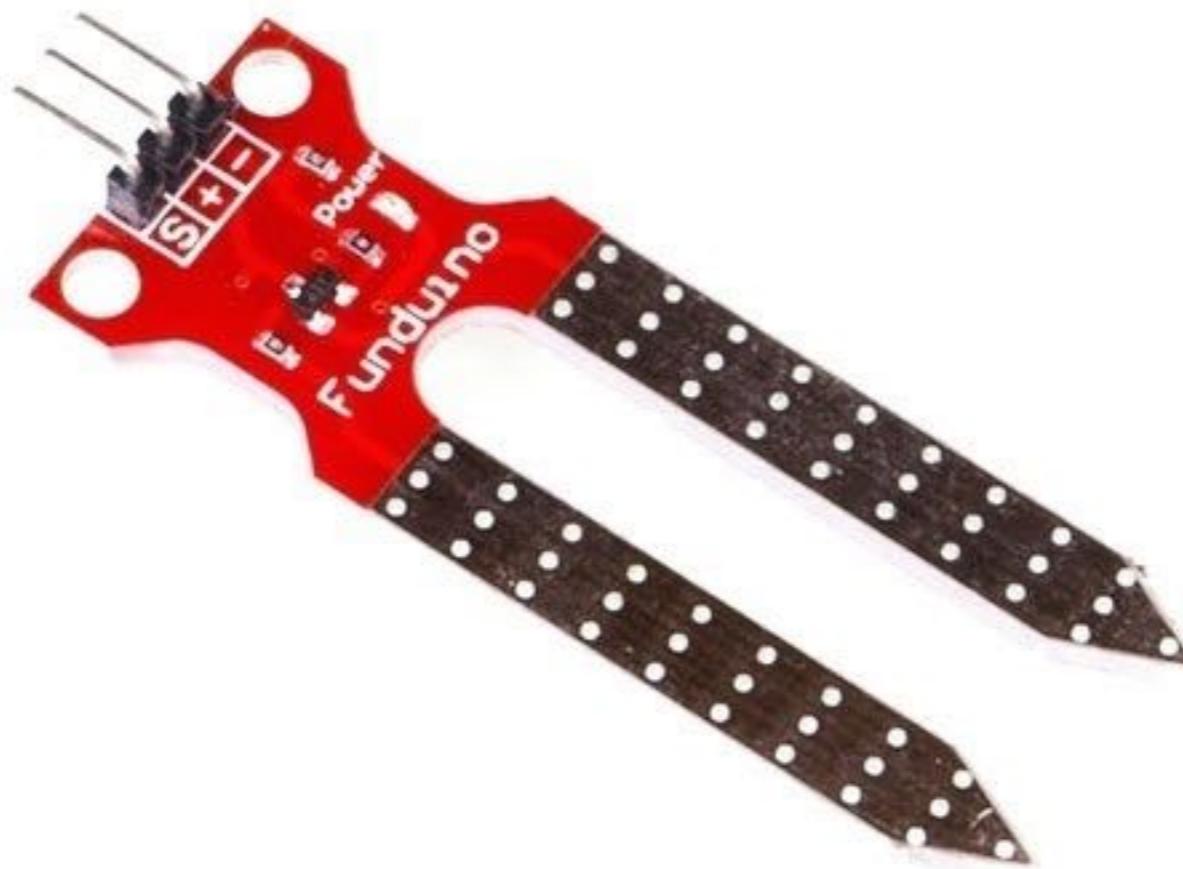


Jumper in Pos. 1

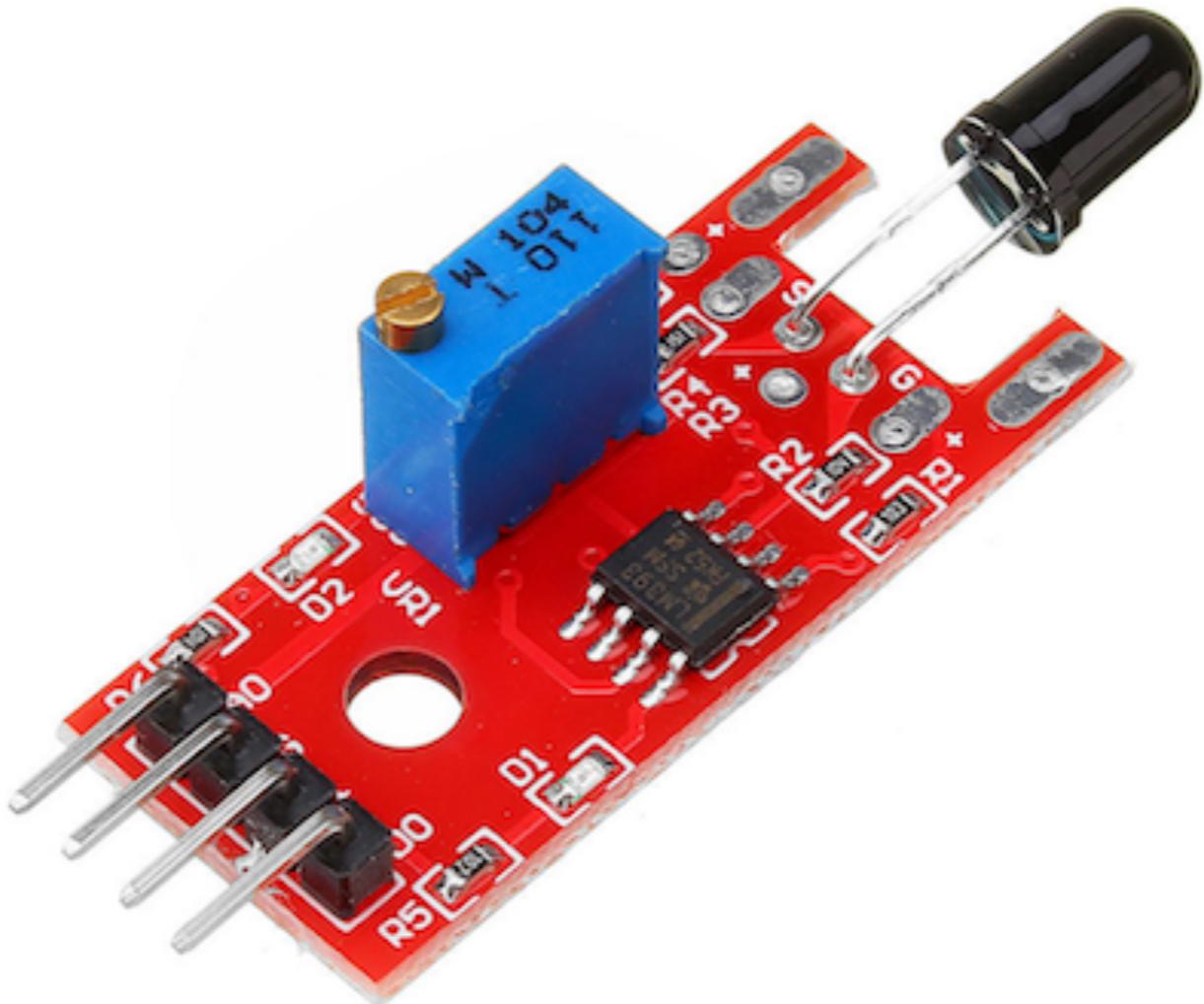


Jumper in Pos. 2

# FEUCHTIGKEITSSENSOR



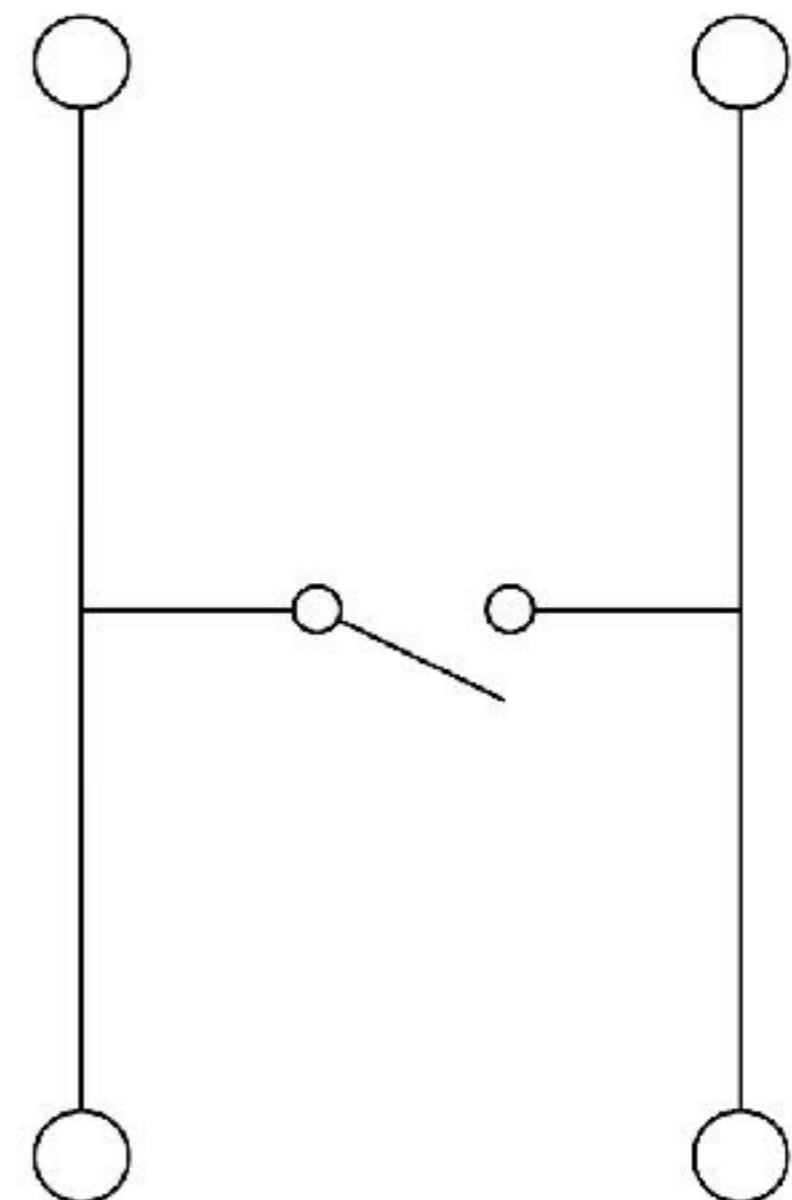
# FLAMMENSENSOR



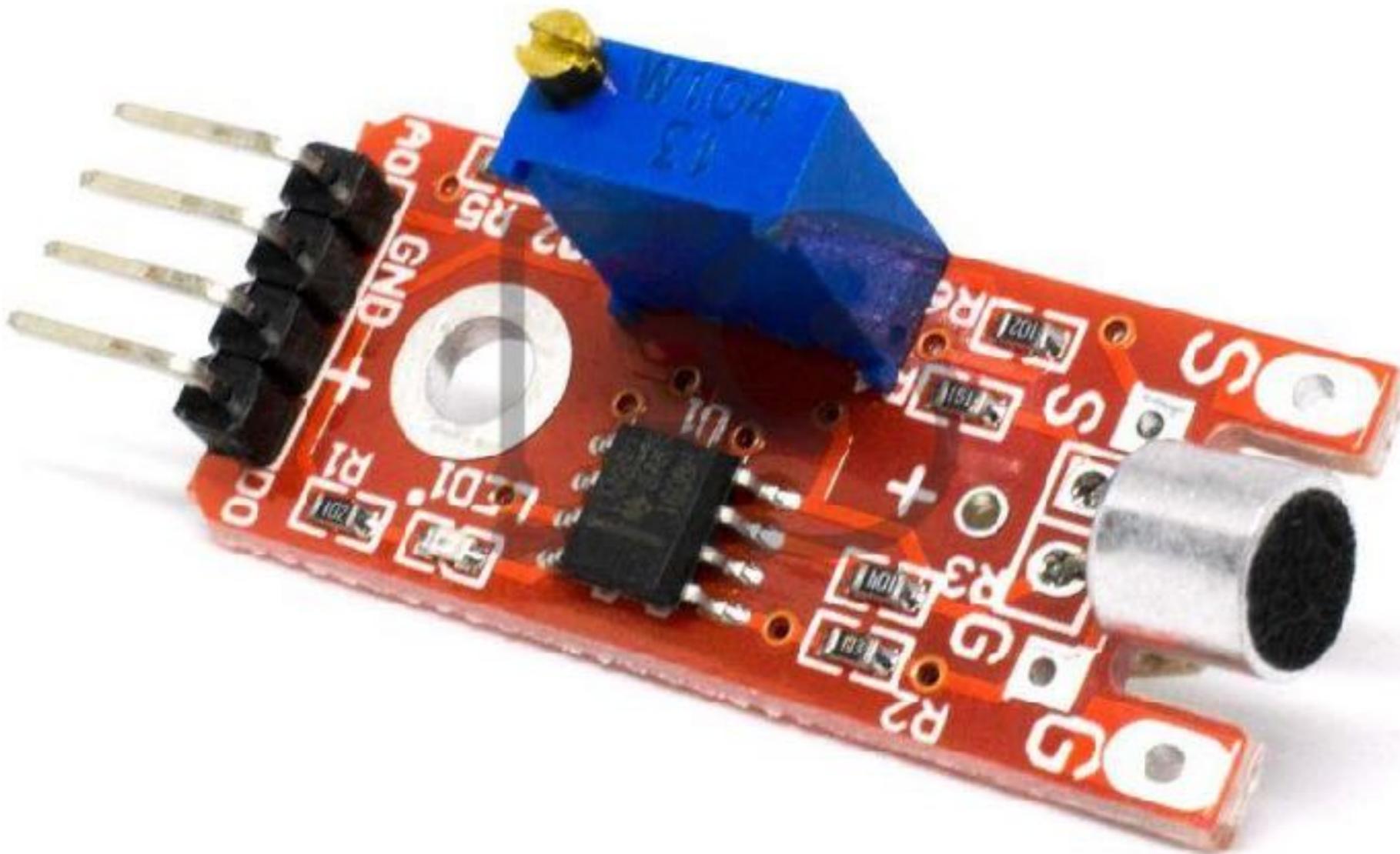
# LICHTSENSOR



# TASTER



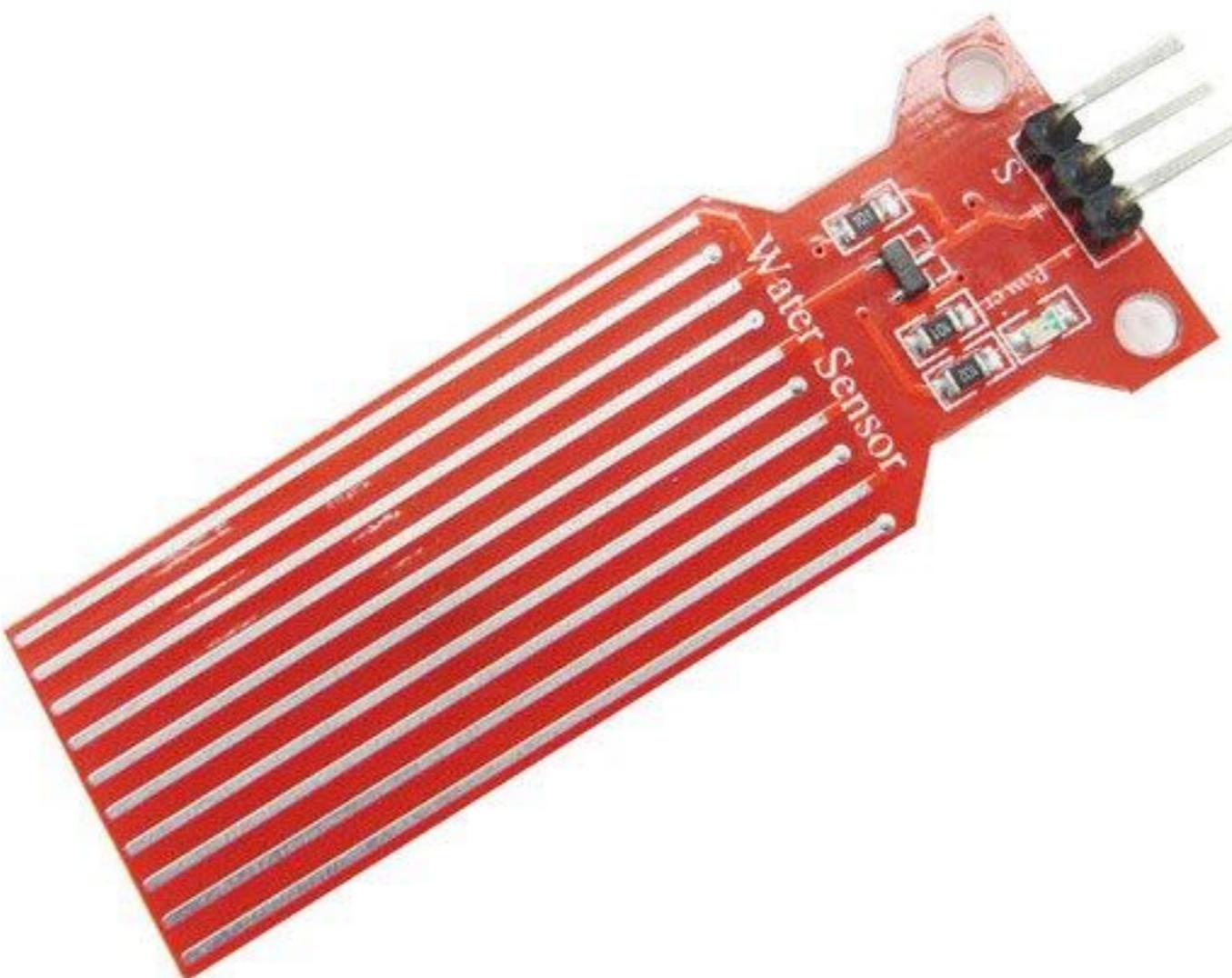
# SOUNDSENSOR



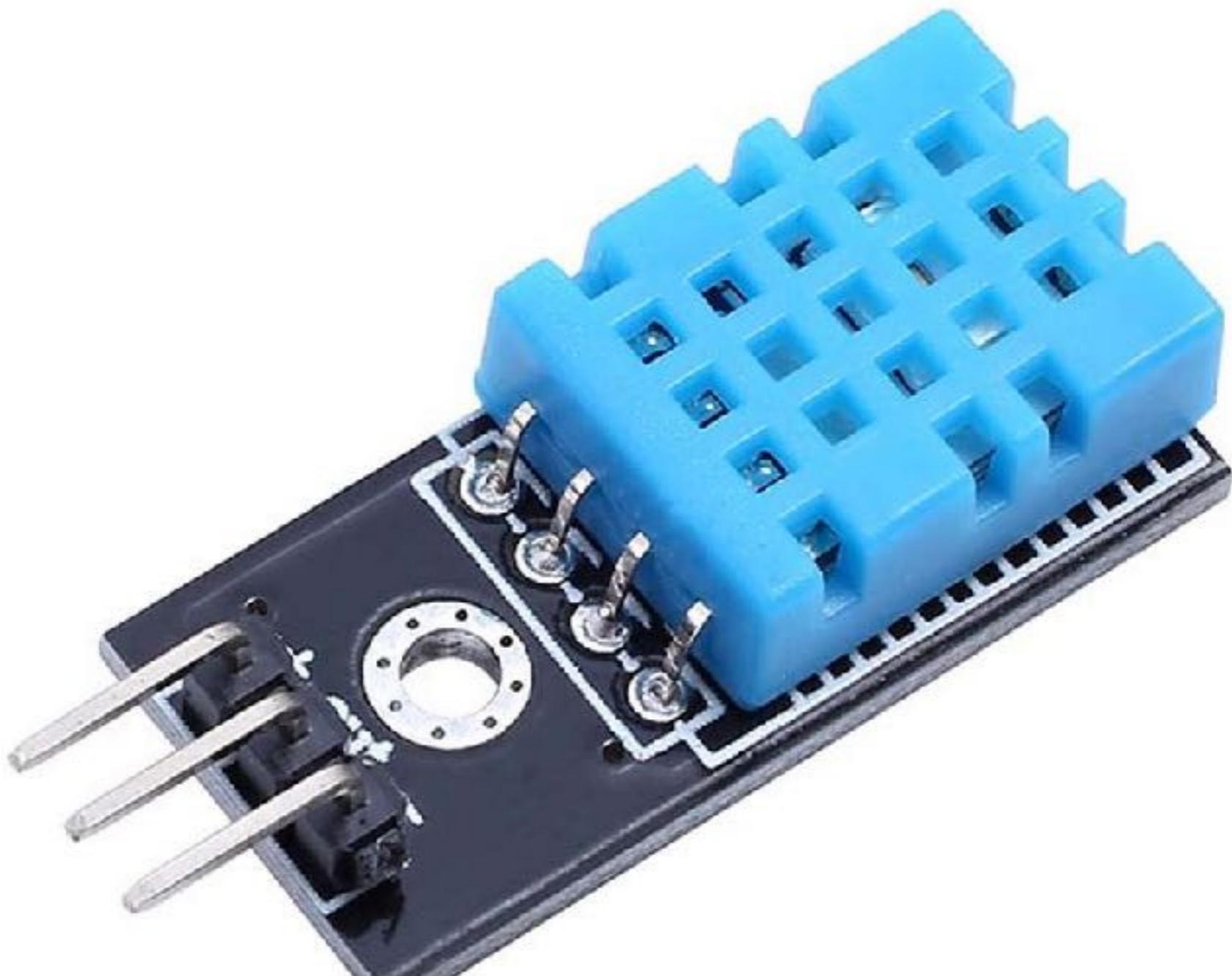
# TILT



# WASSERSTAND

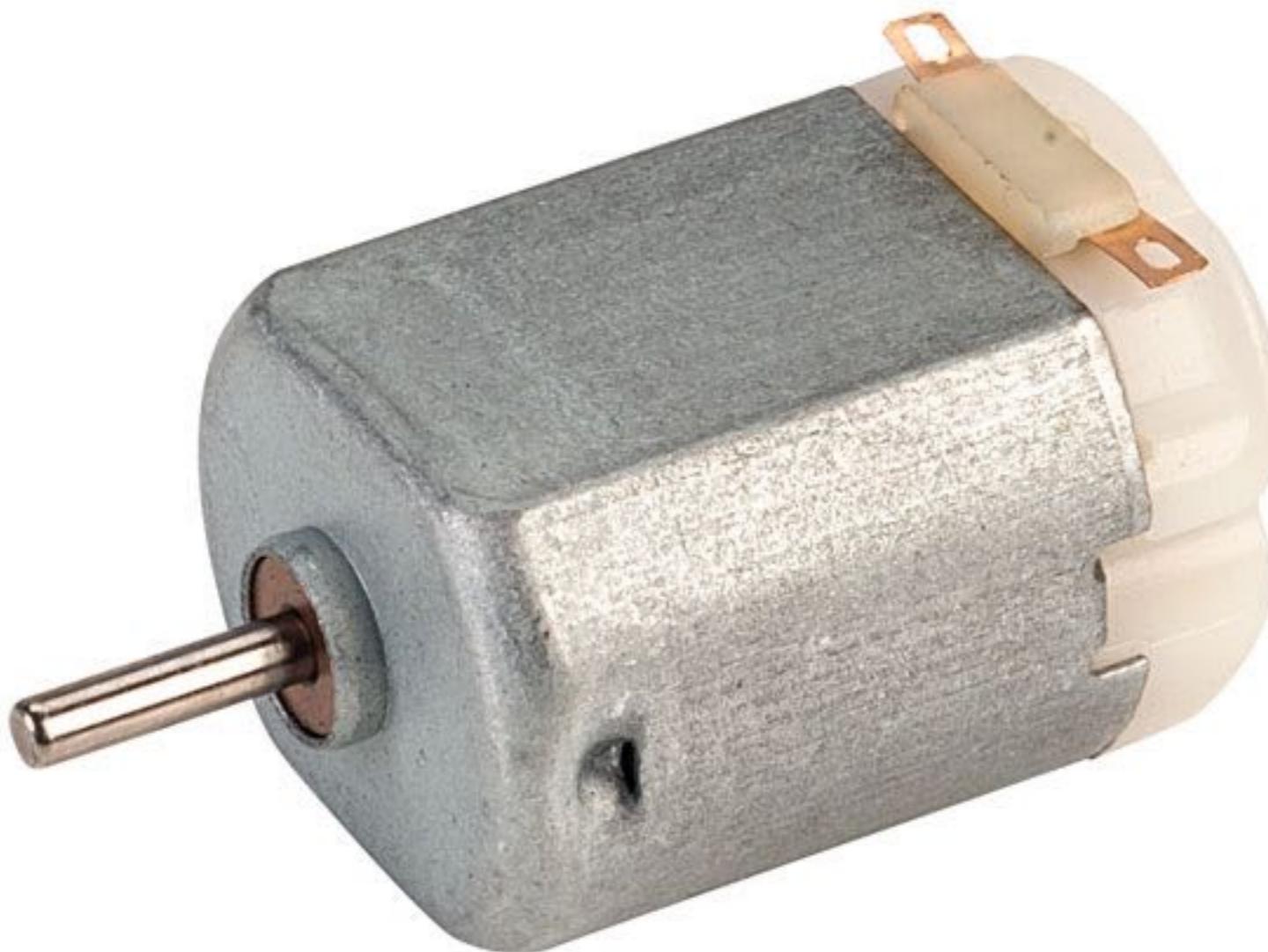


# **TEMPERATUR- UND LUFTFEUCHTIGKEIT**



# **OUTPUTS**

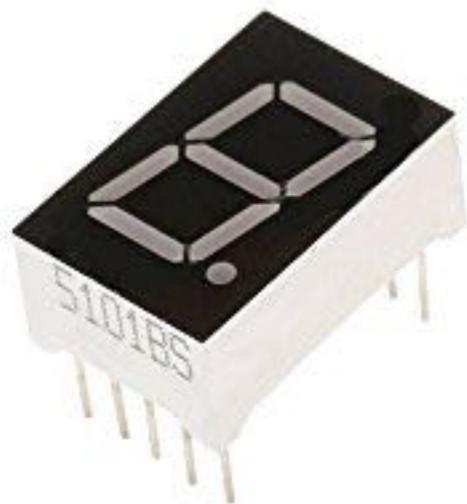
# DC-MOTOR



# SERVOMOTOR



# DIGITALANZEIGE



# BUZZER



# NÜTZLICHE LINKS

## BASIC TUTORIALS

<https://www.arduino.cc/en/Tutorial/HomePage>

<https://www.youtube.com/watch?v=ig6Zl0JJSgQ&t=796s>

## ARDUINO LANGUAGE REFERENCE

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

## PROJEKTE, HACKS, NERDSTUFF, TUTORIALS

<http://www.instructables.com/howto/ARDUINO/>

<https://www.adafruit.com/>

<https://learn.sparkfun.com/tutorials/tags/arduino>

<https://diyhacking.com/diy-projects/arduino-projects/>

<https://www.hackster.io/arduino/projects>

[youtube.com](https://www.youtube.com/)

...