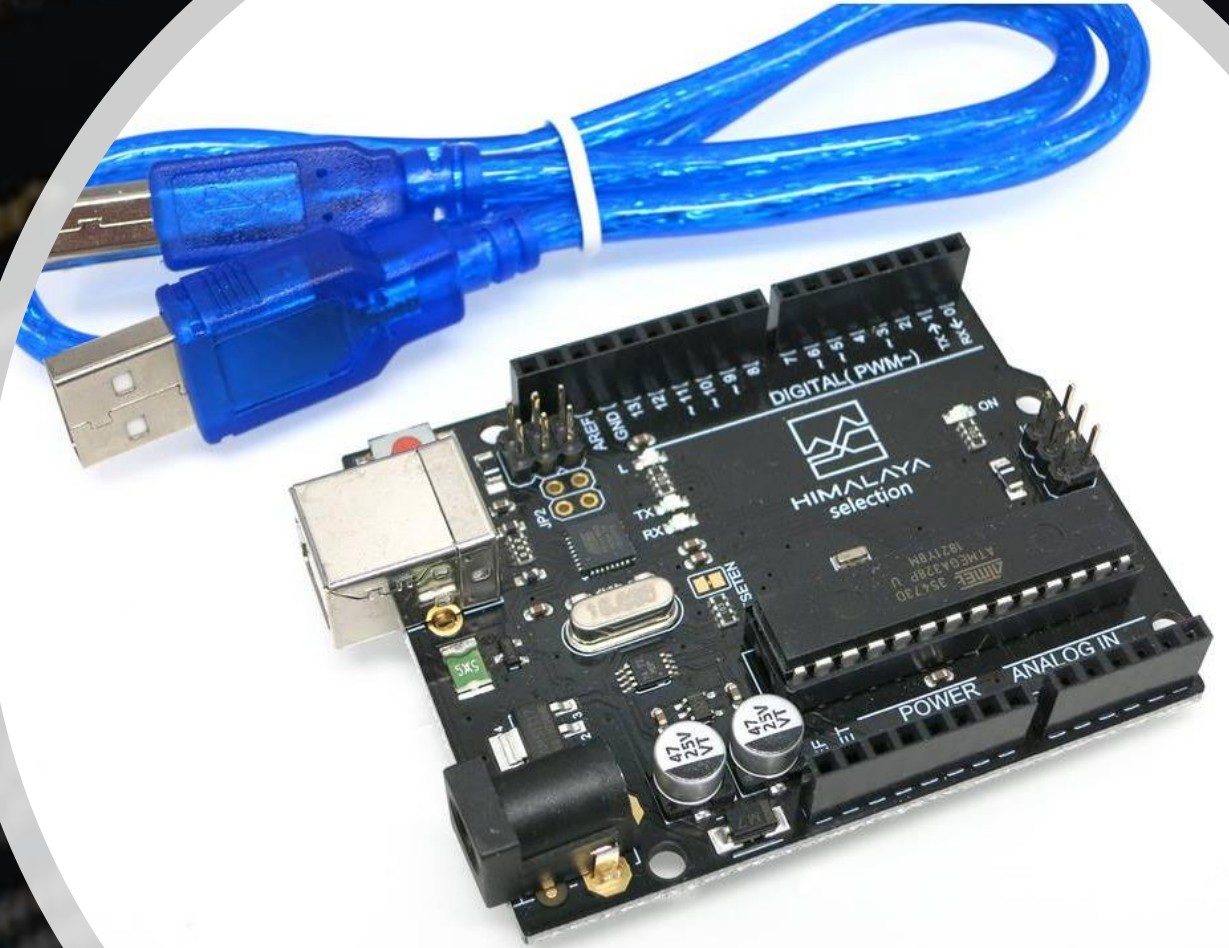
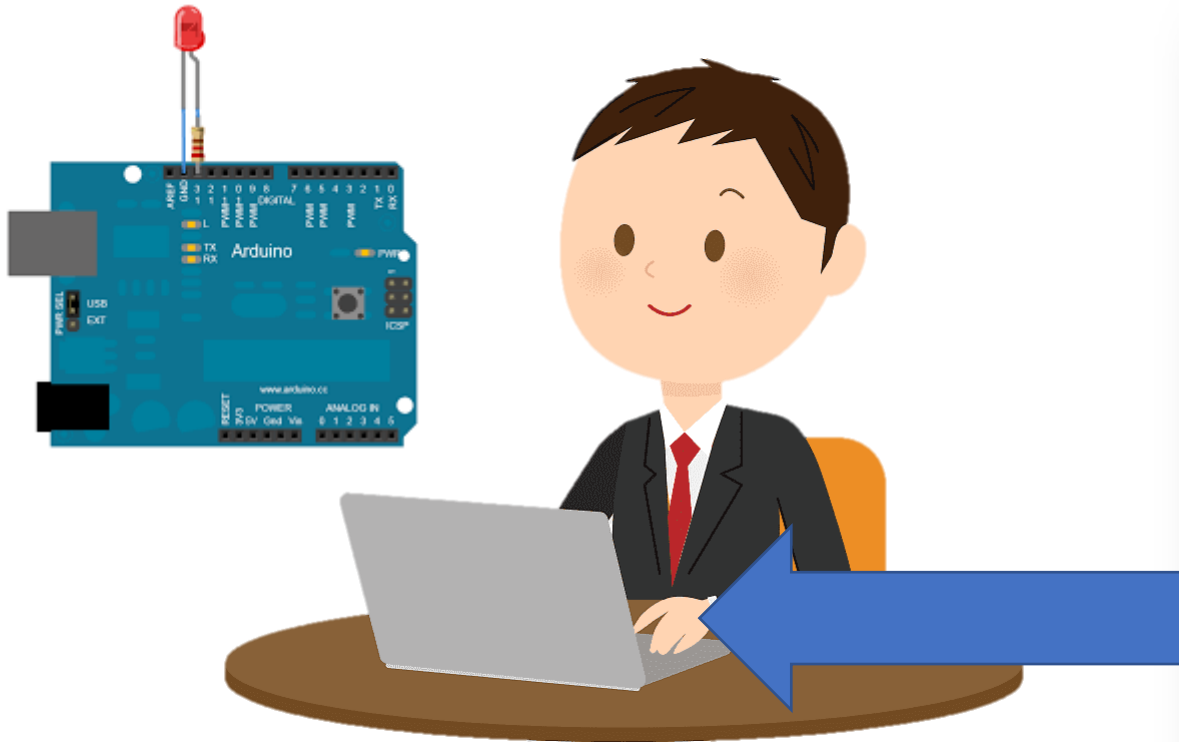


Arduino Programmierung



Programmieren, Was ist das?

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.5". The main text area contains the following code:

```
Blink $  
  
This example code is in the public domain.  
  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {$  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

The status bar at the bottom indicates "32" and "Arduino/Genuino Uno on COM1".

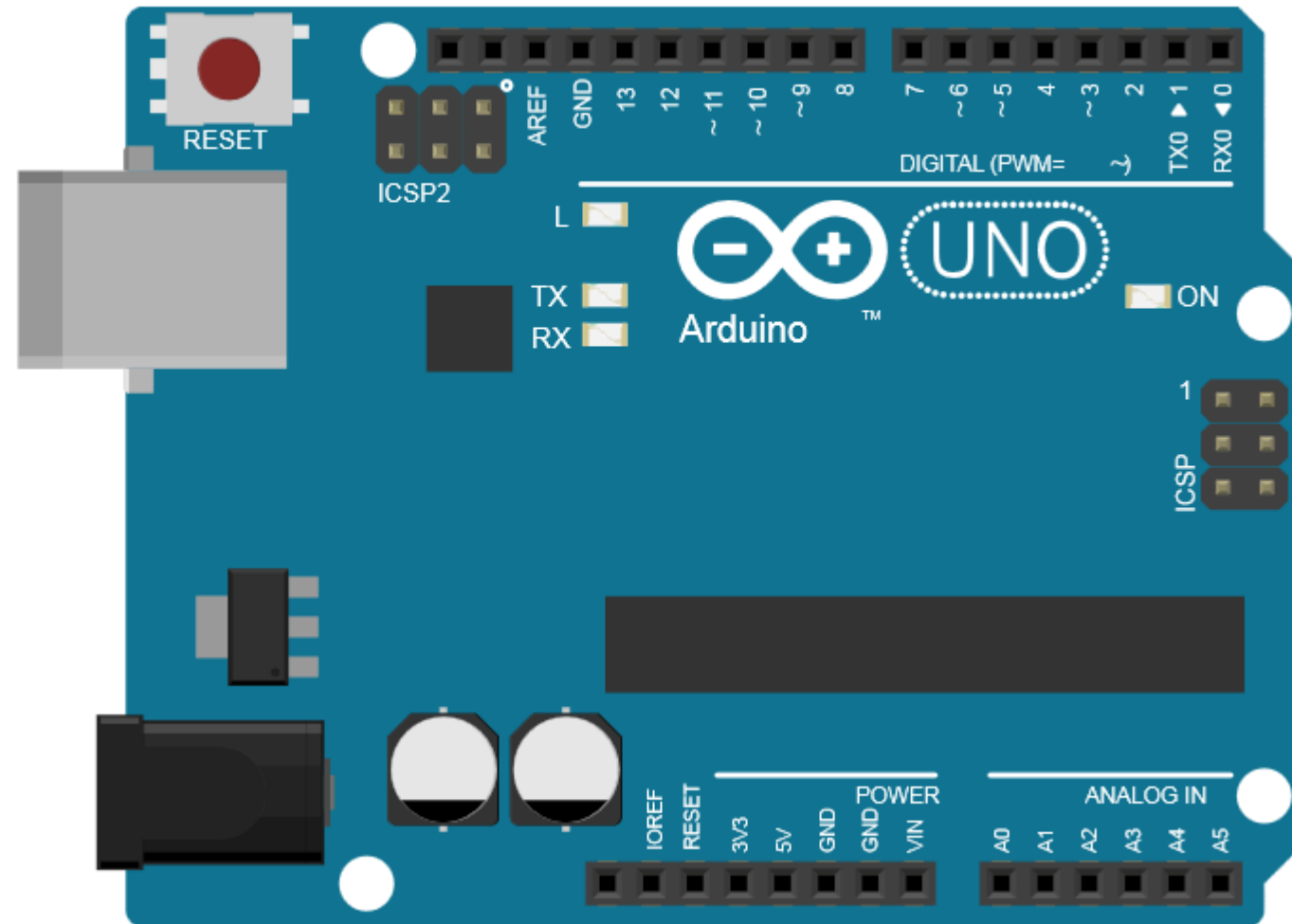
Programmieren

- Man muss genau sagen was man möchte
- **Ü: Welche Schritte braucht man zum Nuss-Nougat-Creme Brötchen**

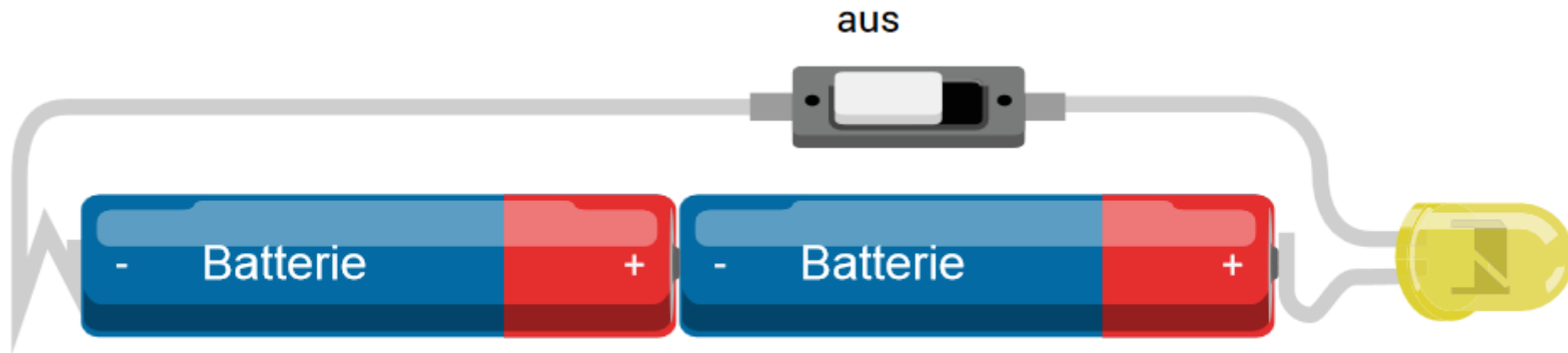
Ü: Welche Schritte braucht man zum Nuss-Nougat-Creme Brötchen

1. a) Schrank auf und ein Brötchen heraushohlen
b) Bei der Bäckerei eins kaufen
2. Messer hohlen
3. Brötchen aufschneiden
4. Nuss-Nougat Creme hohlen
5. Messer in Nuss-Nougat Creme tauchen
6. Brötchen schmieren
7. Alles wieder Aufräumen
8. Messer waschen

Arduino

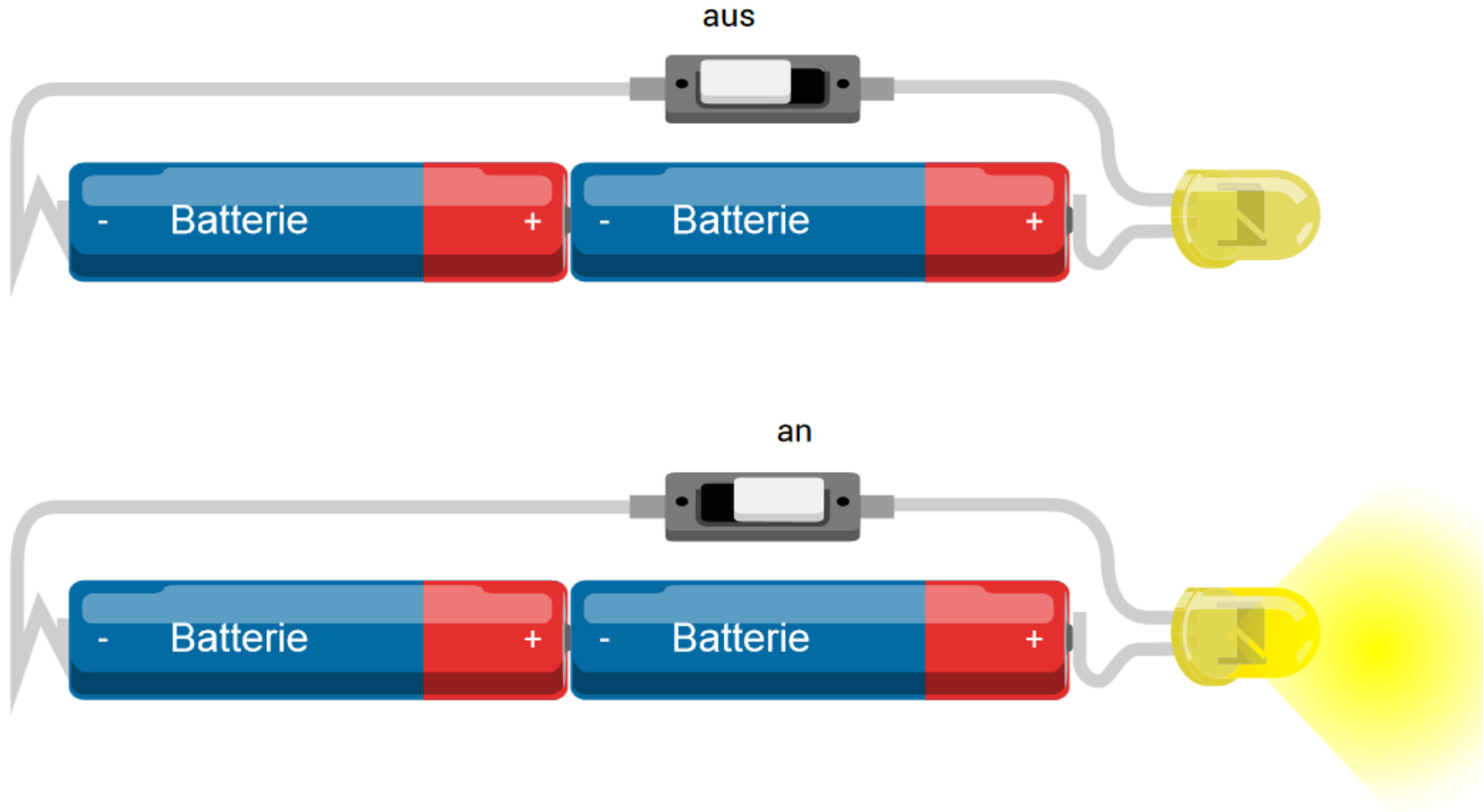


Wie funktioniert der Arduino?

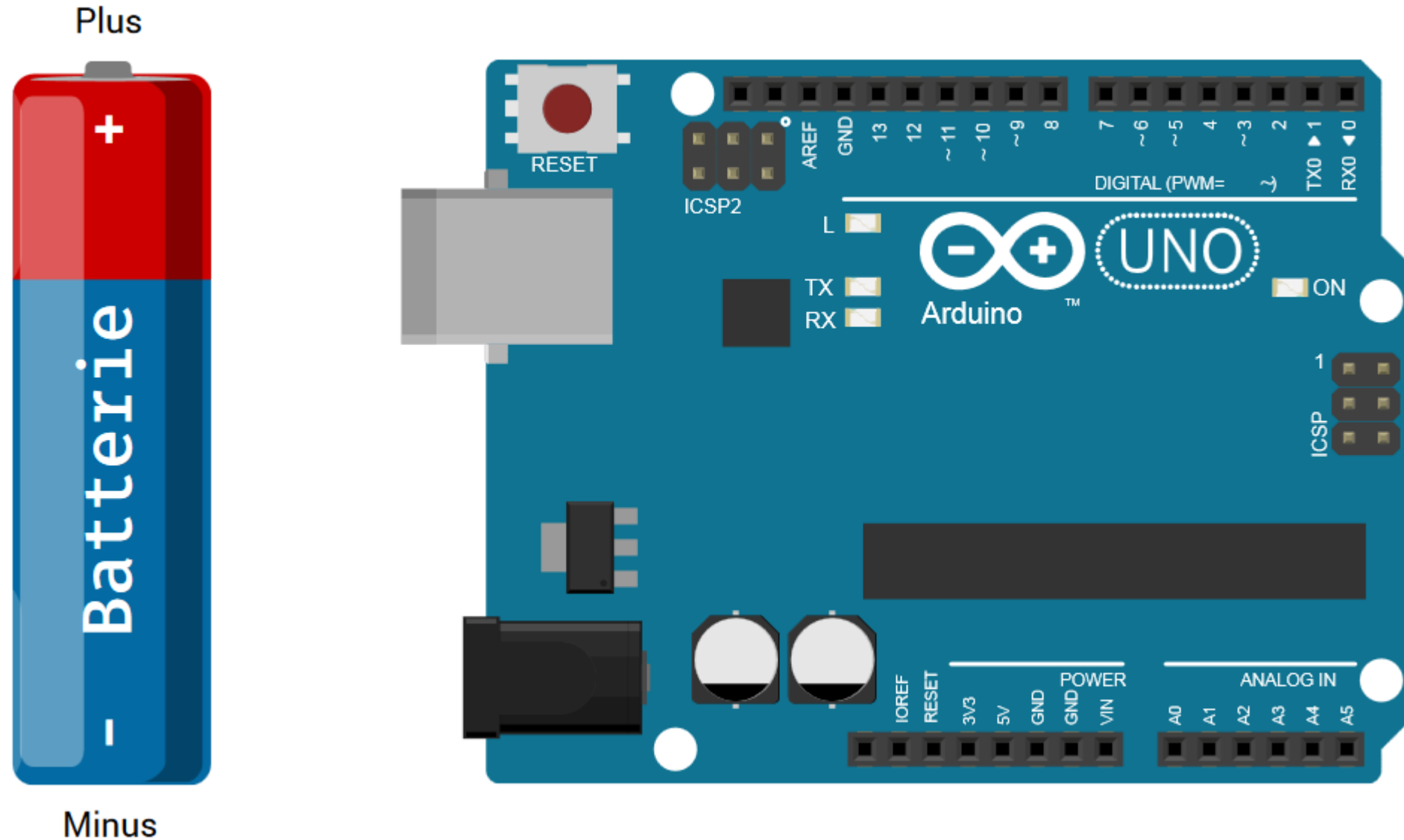


Wie funktioniert der Arduino?

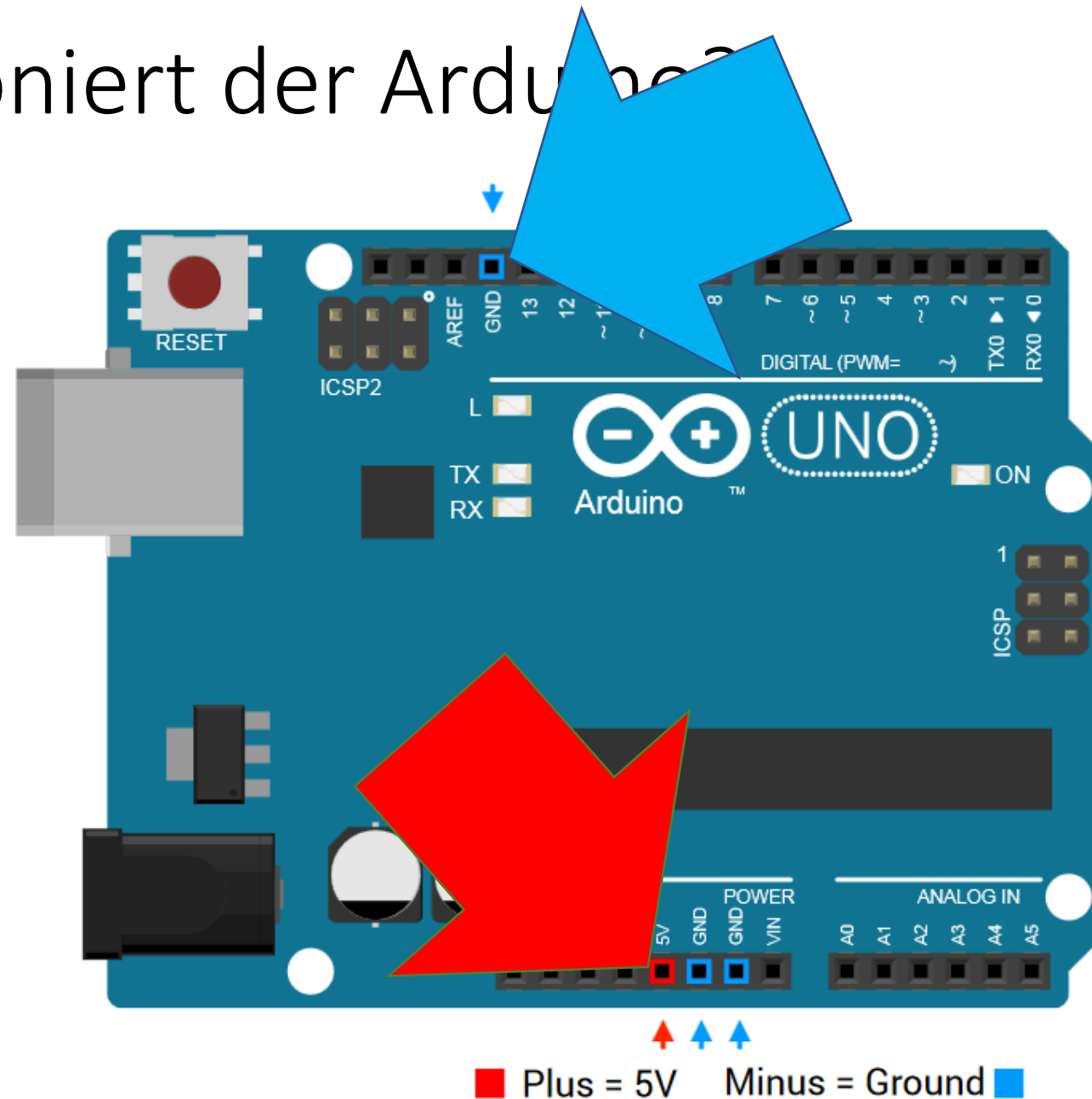
Schaltet man sie ein, leuchtet die Taschenlampe.



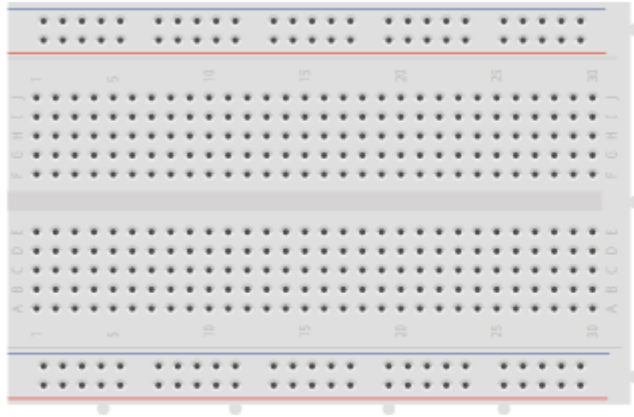
Wie funktioniert der Arduino?



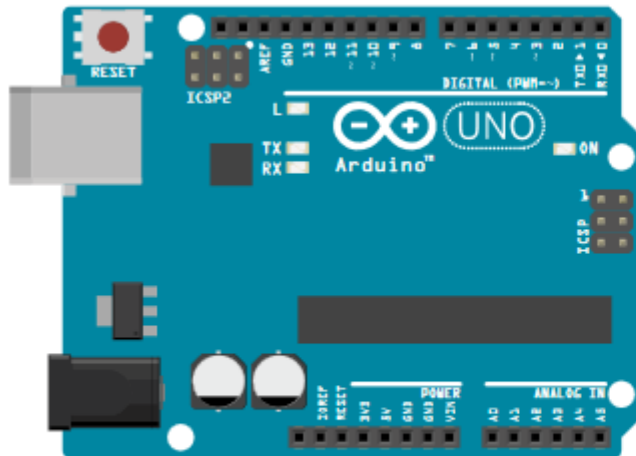
Wie funktioniert der Arduino?



Wie funktioniert der Arduino?



Breadboard



Arduino

Taster



Widerstand 220 Ω (Ohm)



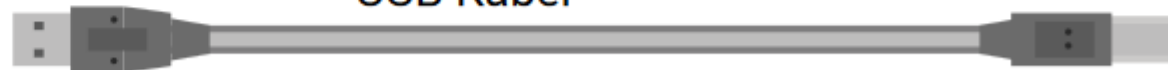
LED



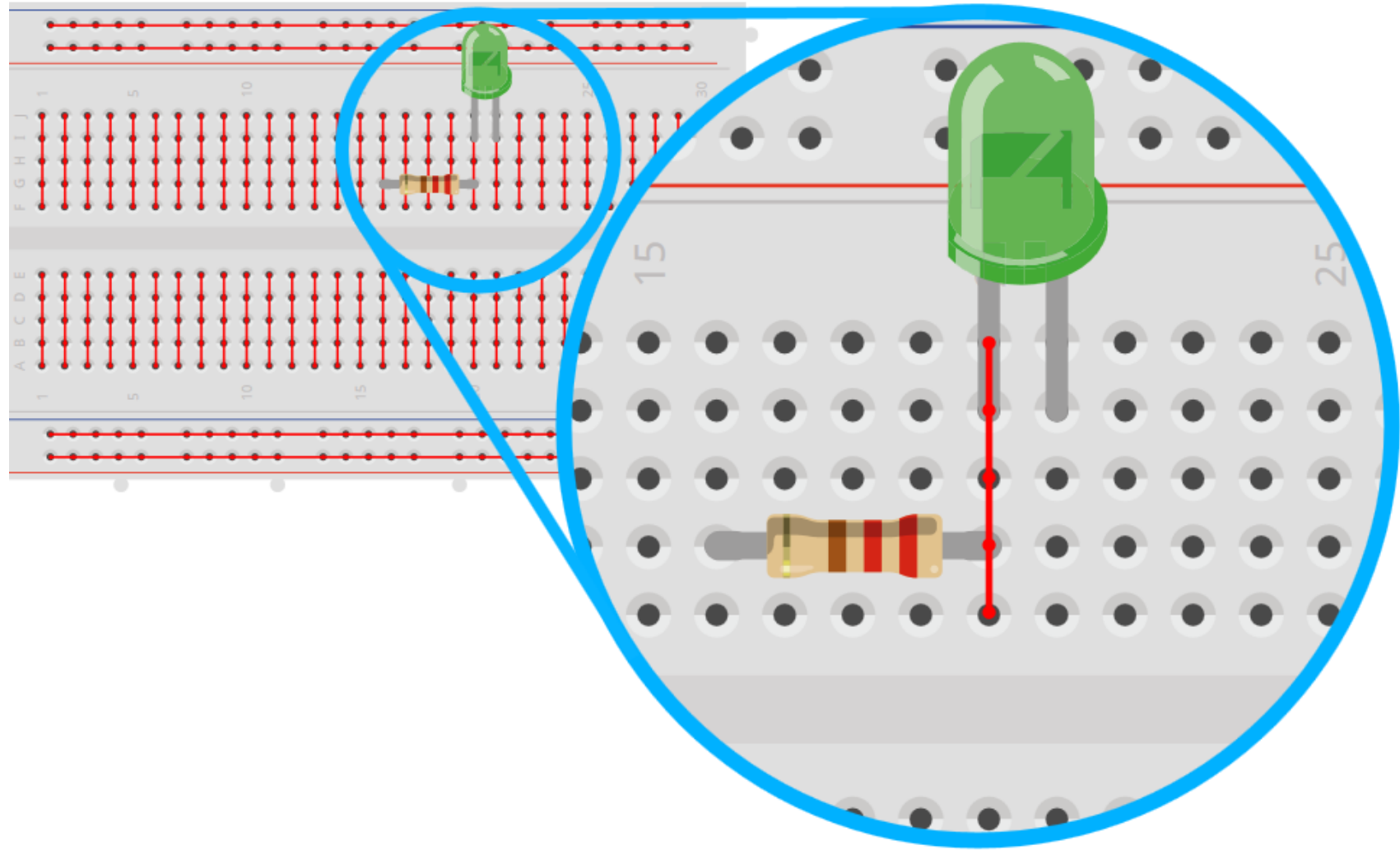
Steckkabel (lang)



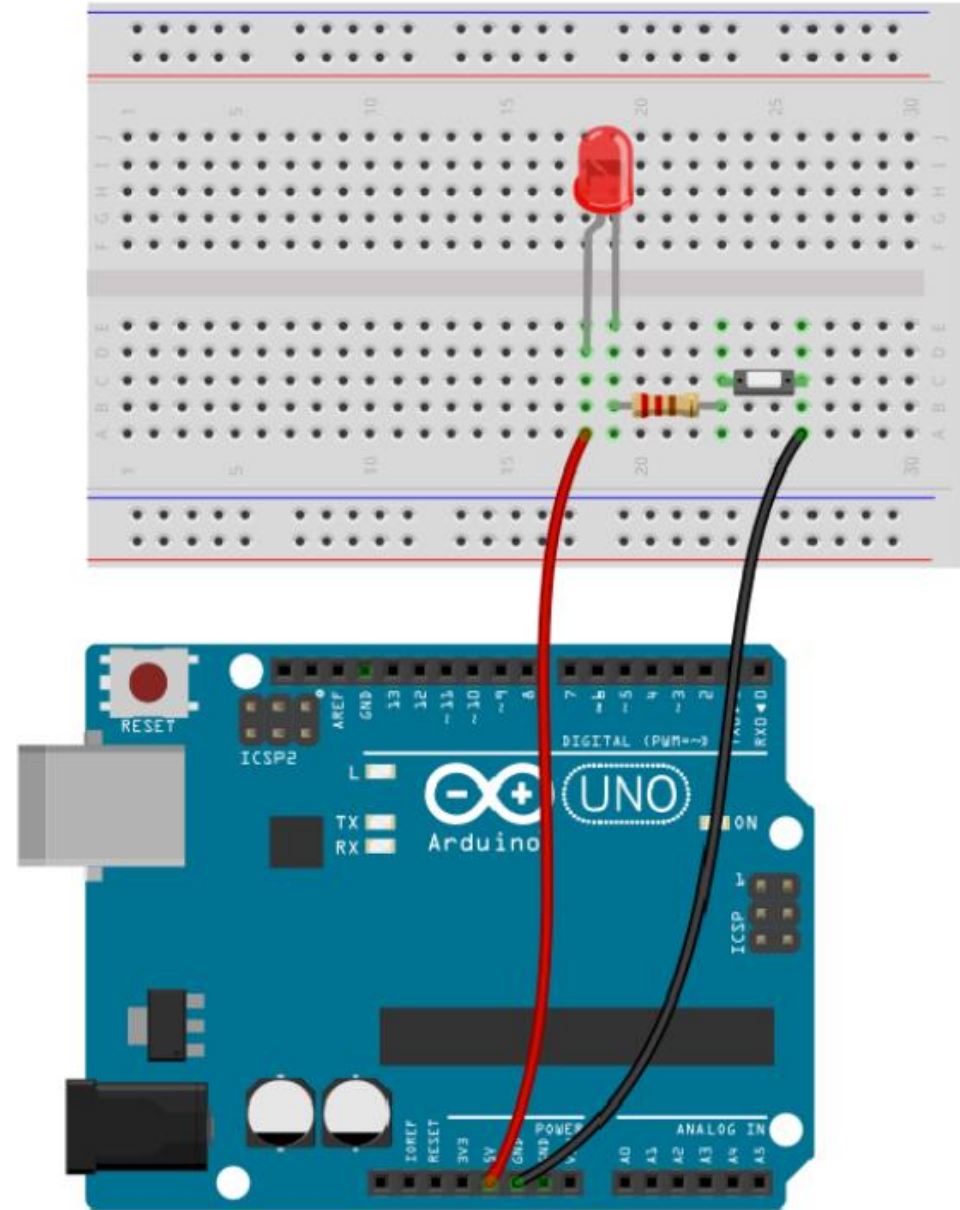
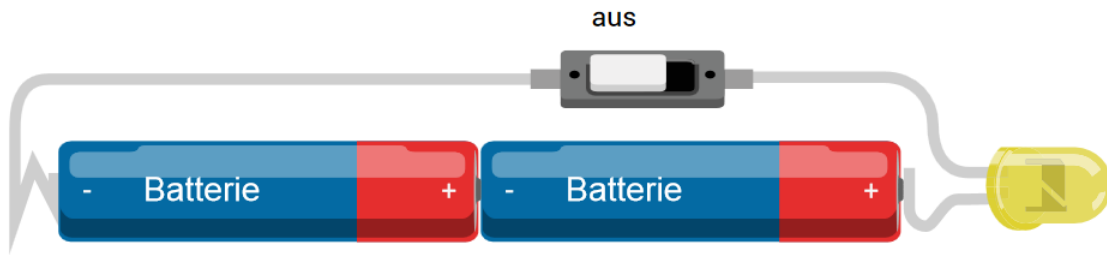
USB Kabel



Breadboard

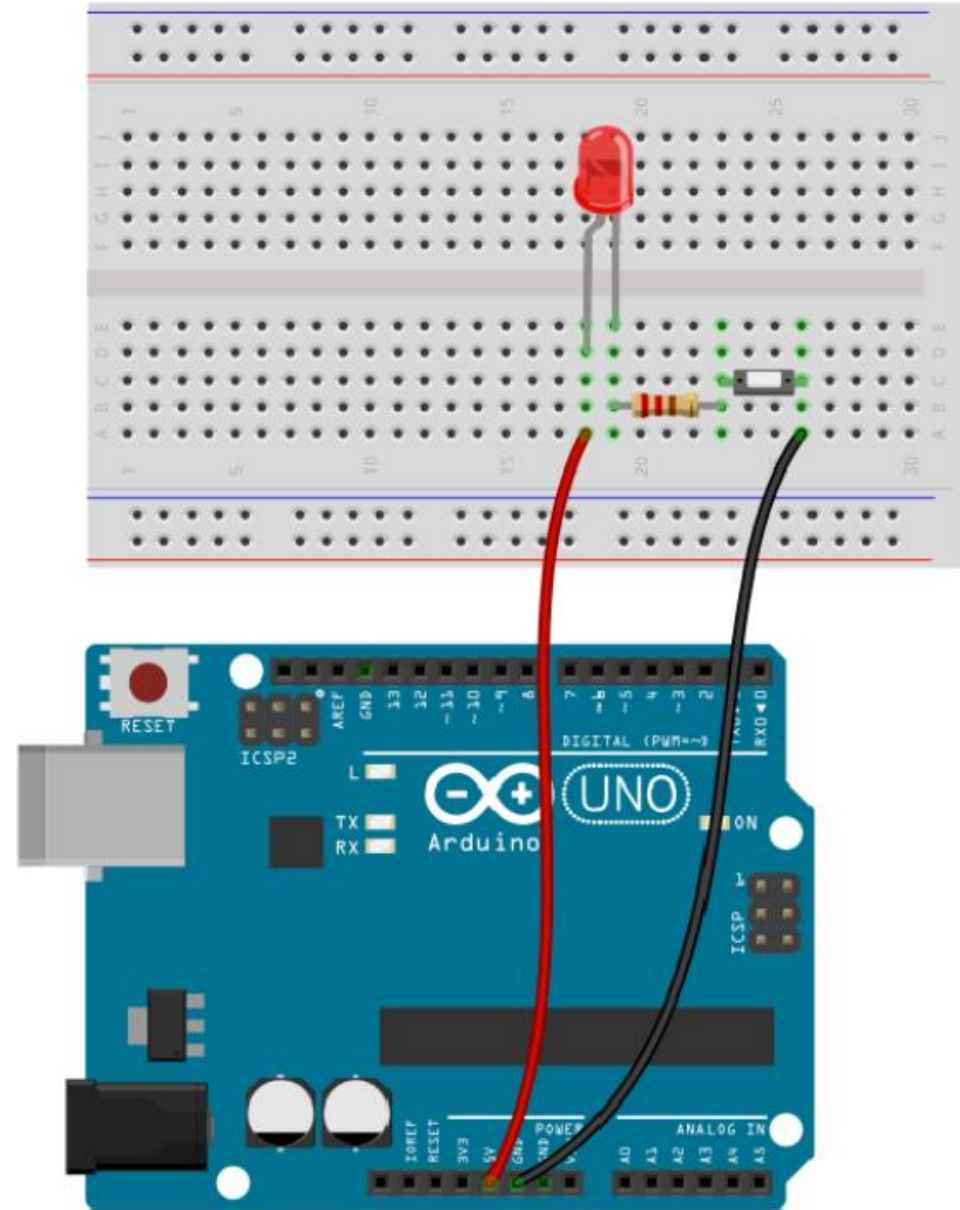
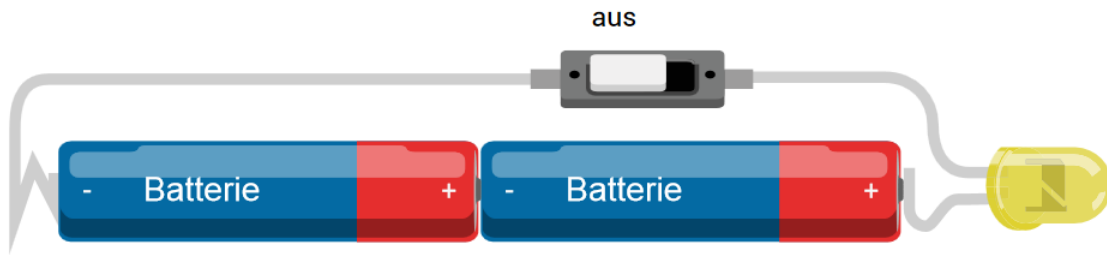


Wie funktioniert der Arduino?

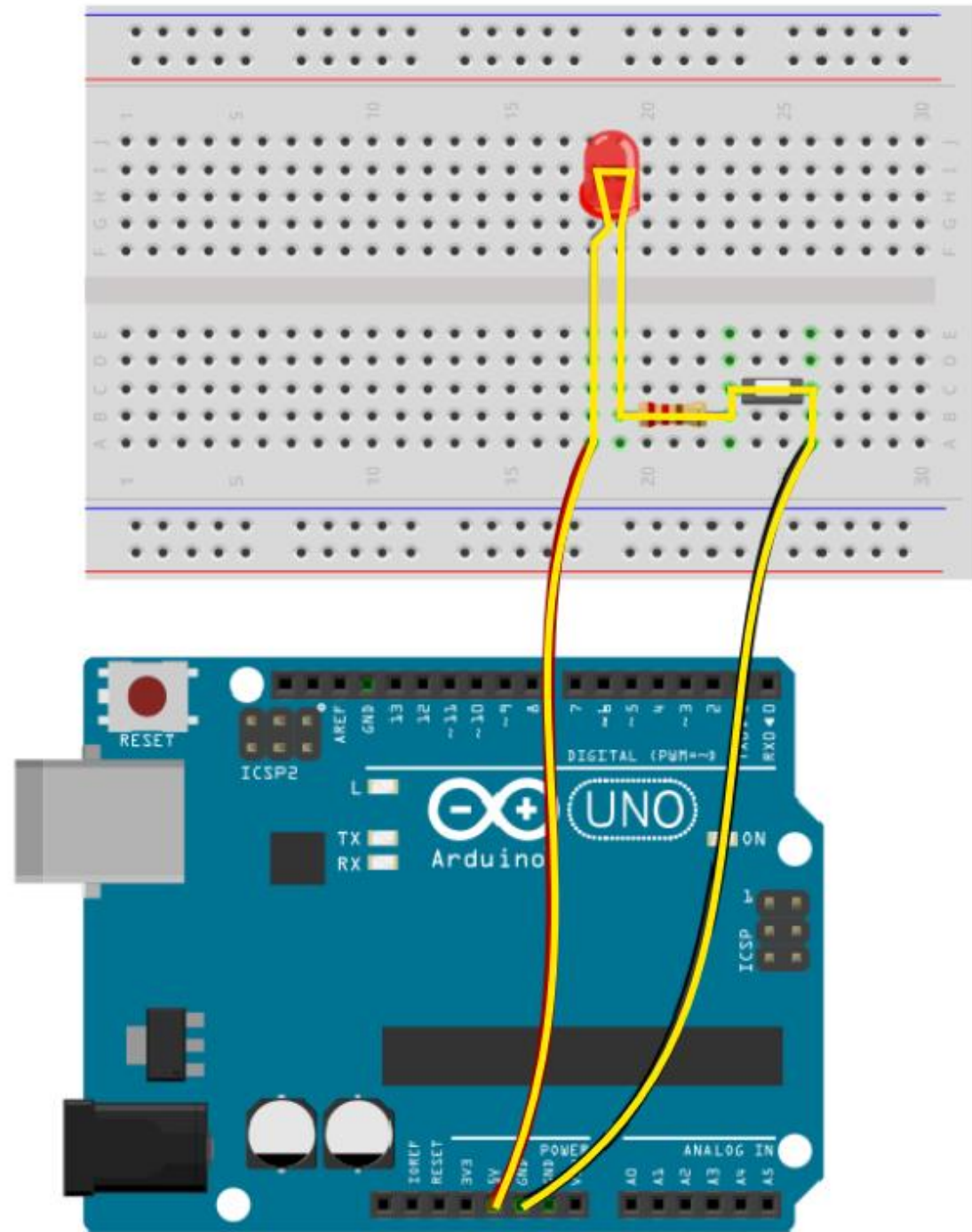


Wie funktioniert der Arduino?

F: Wo fließt der Strom lang?



Wie funktioniert der Arduino?



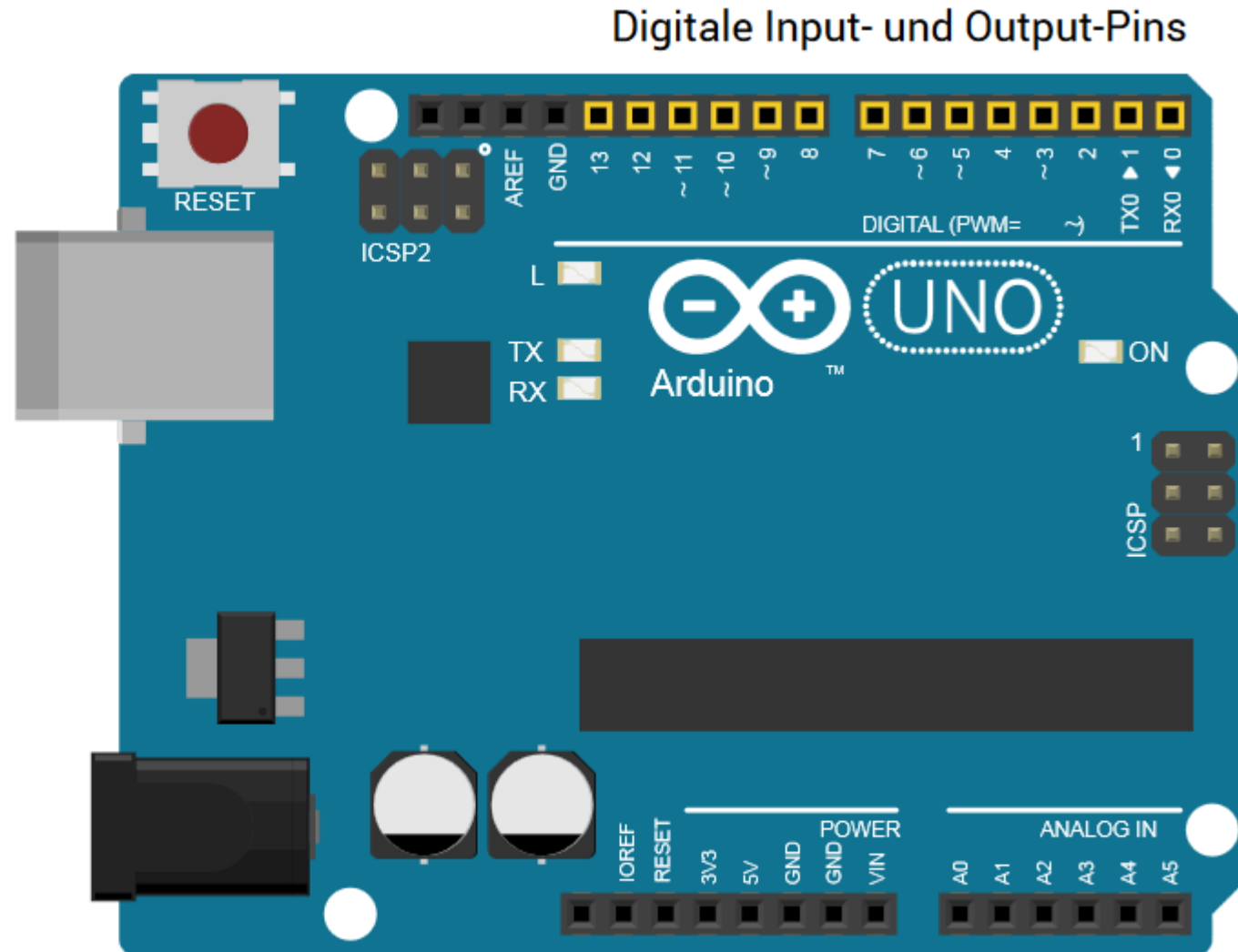
Wie funktioniert der Arduino?

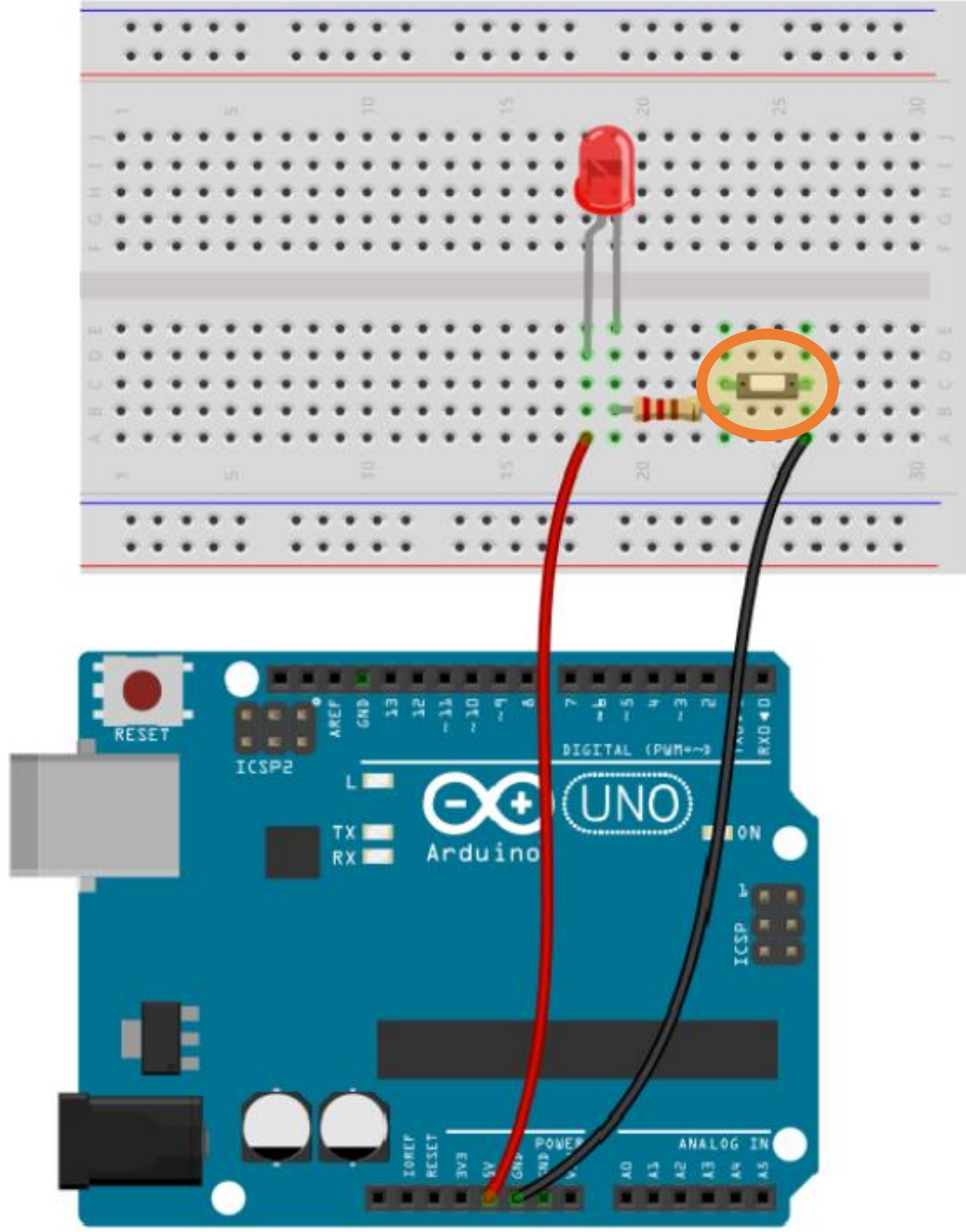
Inputs

- Schalter
- Taster
- Kontakte

Outputs

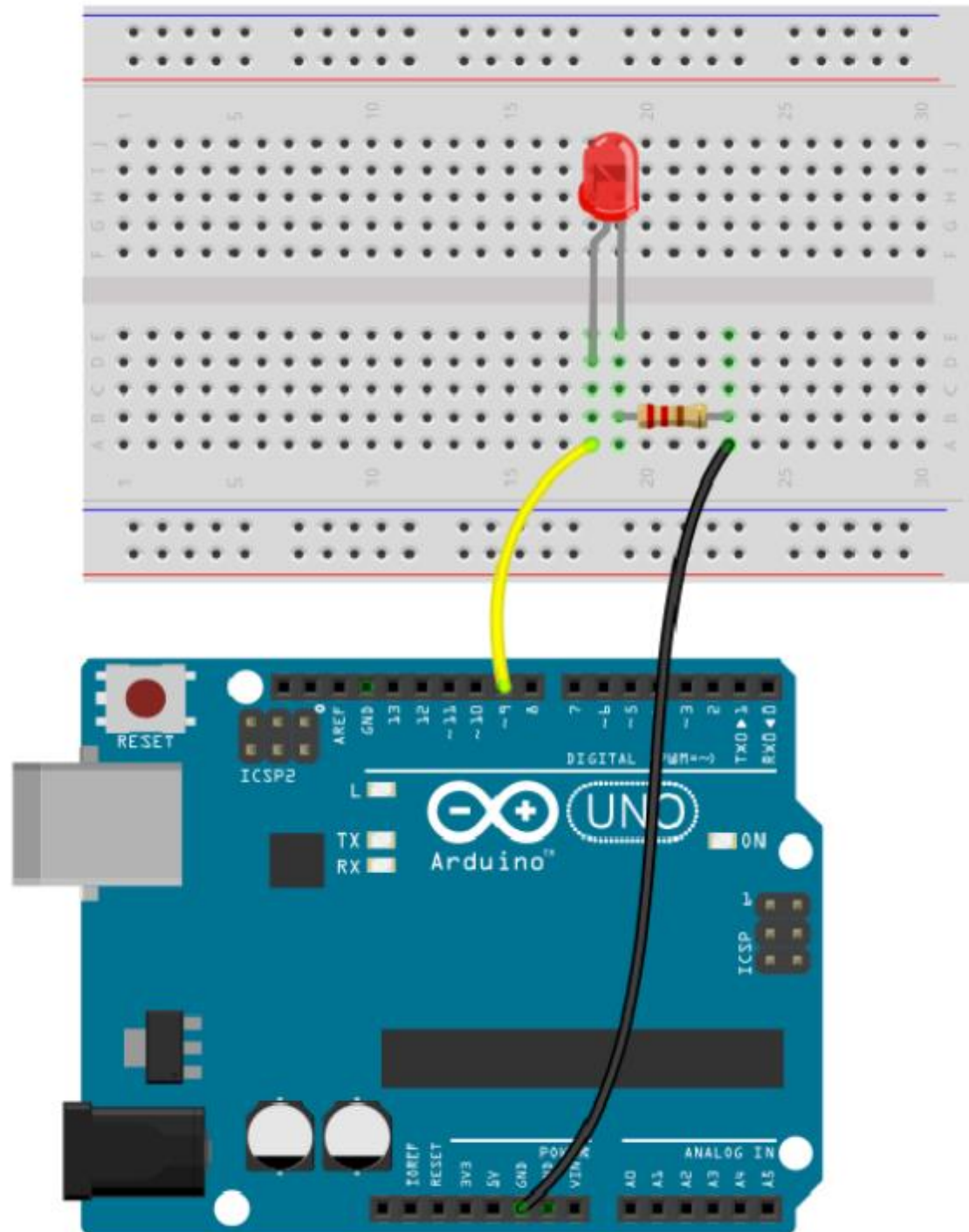
- LEDs
- Motoren
- Elektromagnete
- Lautsprecher





Wie funktioniert der Arduino?

Arduino wird zum „Schalter“





Programmierung

Struktur eines Programms

```
void setup(){
```

```
}
```

```
void loop(){
```

```
}
```


bei Programmstart

ständige Wiederholung

Programmierung

- Befehle:

`Serial.println ("Hello World") ;`

 **Semikolon!!**

Name des Befehls Argument(e)

- Kommentare:

```
// Dies ist ein Kommentar
```

```
/*  
 * Dies auch  
 */
```

Programmierung – wichtige Befehle

- `pinMode(13, OUTPUT)` / `pinMode(14, INPUT)`
- `delay(1000)` -> in Milisekunden (1000ms = 1s)
- `digitalWrite(13, HIGH)` / `digitalWrite(13, LOW)`
- `digitalRead(14)`

Ü: LED blinken

Das zweite Programm

- „Serieller Monitor“
 - Kommunikation mit Arduino während Programmablauf
- Textausgabe am Computer

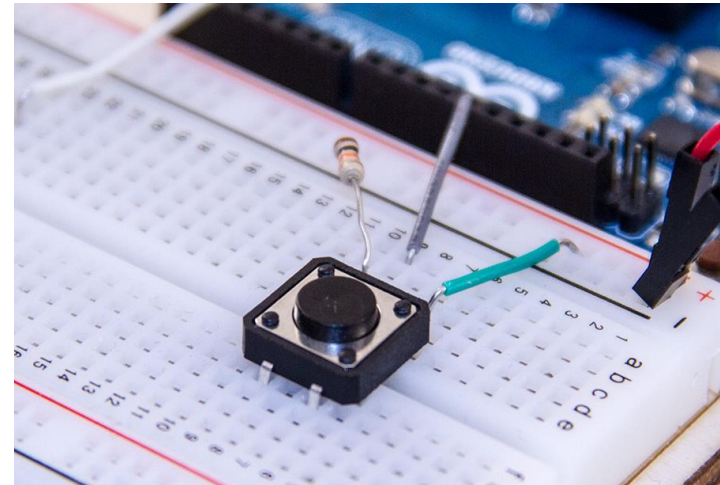
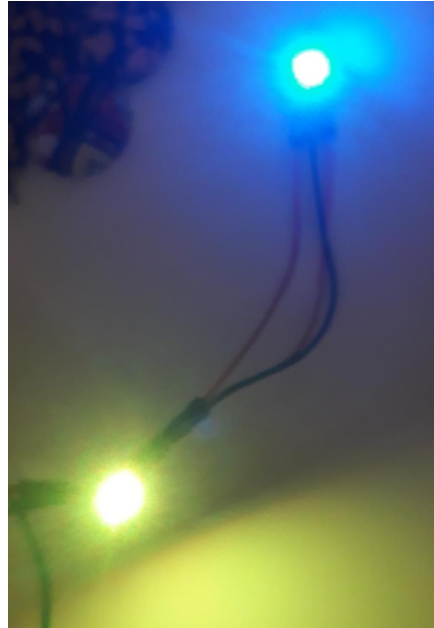
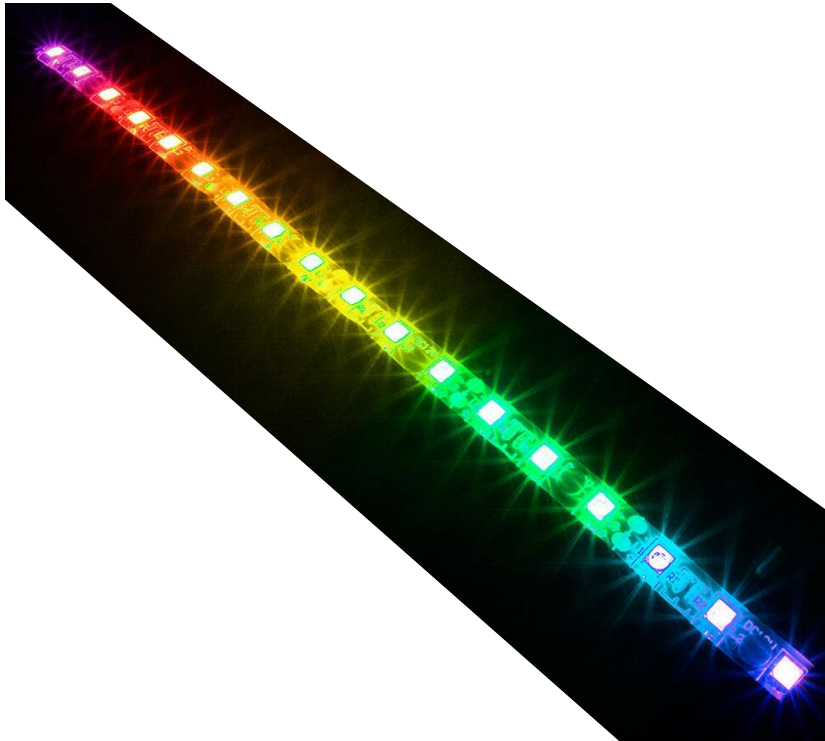
Ü: „Hello World Program“



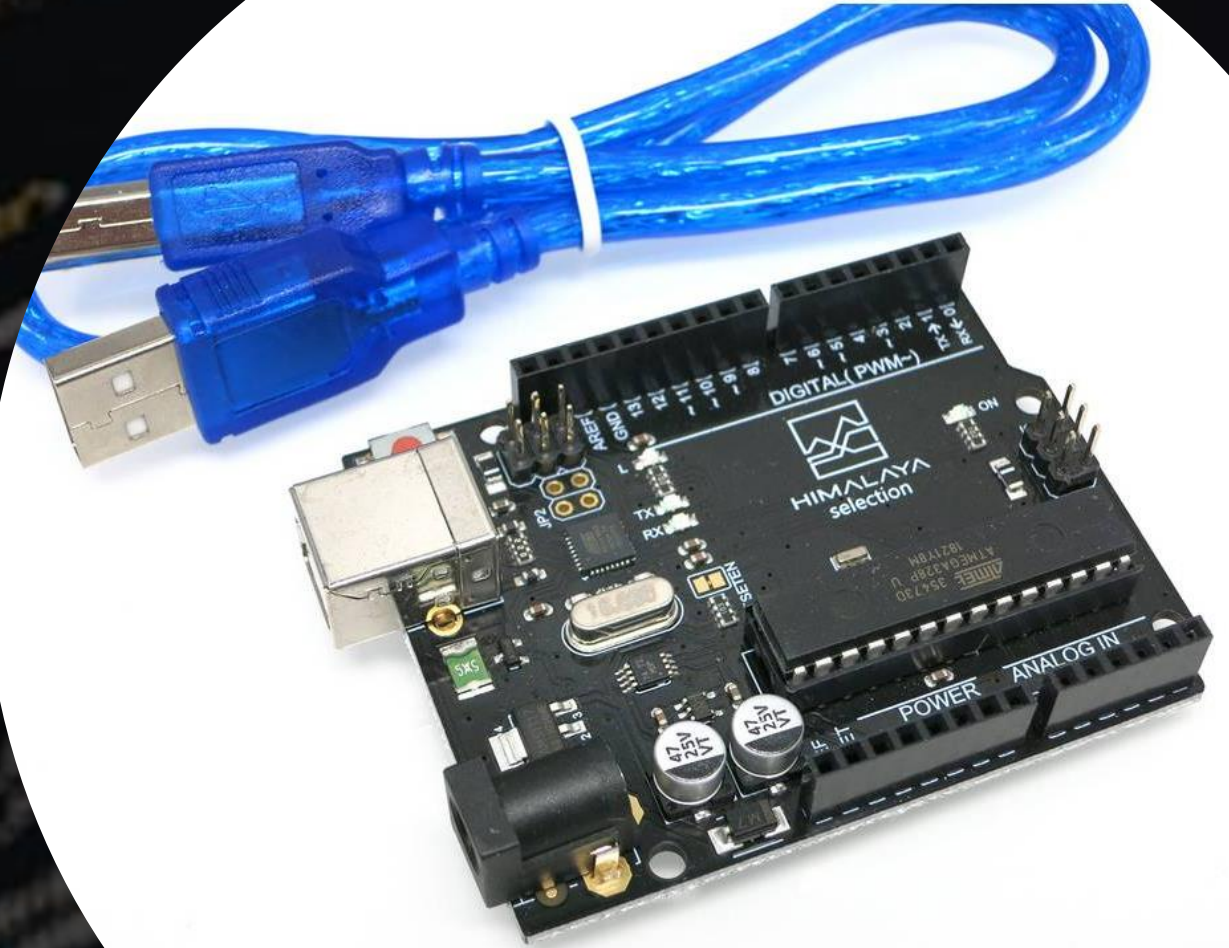
```
Hello_World | Arduino 1.8.13
Datei Bearbeiten Sketch Werkzeuge Hilfe

Hello_World
1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   Serial.println("Hello World");
8   delay(1000);
9
10 }
```

Bauteile

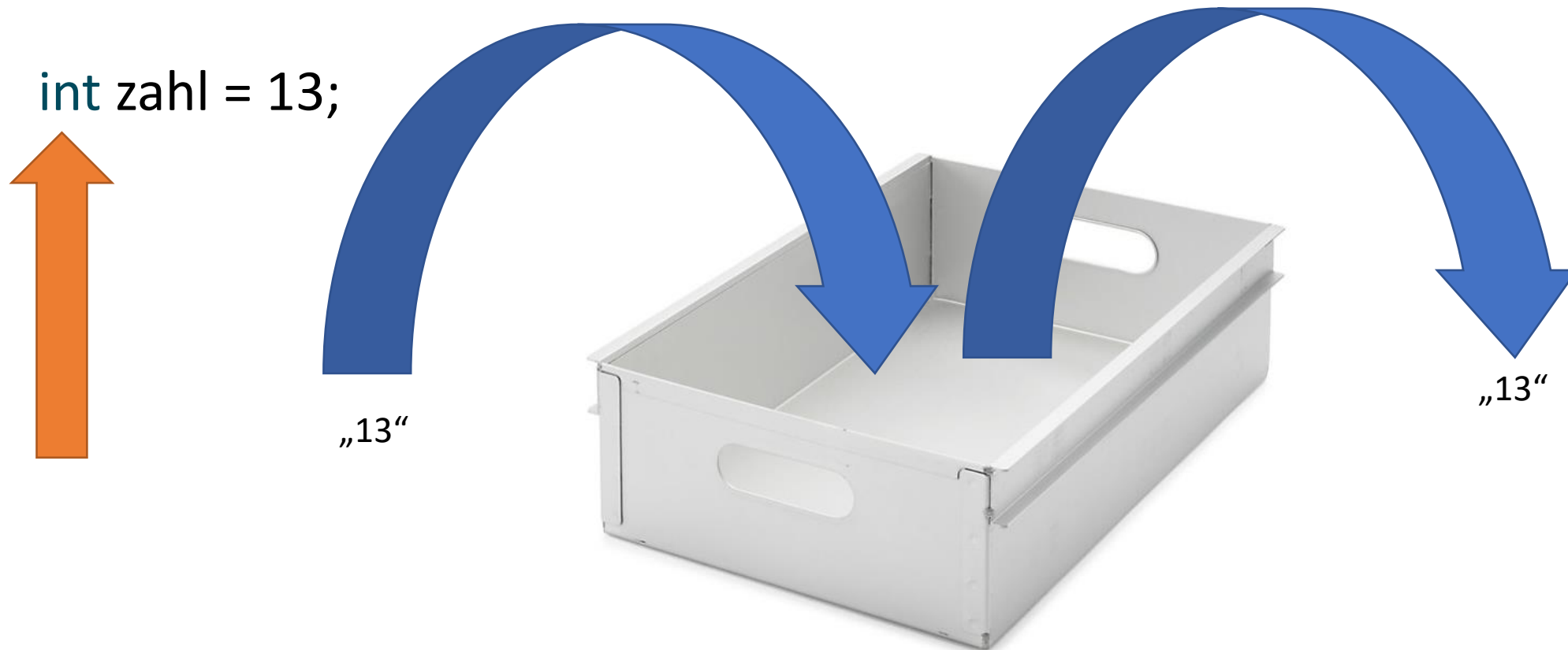


Eure Programmierideen



Programmieren - Variablen

- Dinge „abspeichern“



Programmieren - Variablen

- Variablentypen
- 13 -> Zahl
 - byte
 - **int**
 - long
 - float
- „Hello World“ -> Text
 - String

Programmieren - Variablen

- Wahr/Falsch -> Wahrheitswerte
 - boolean

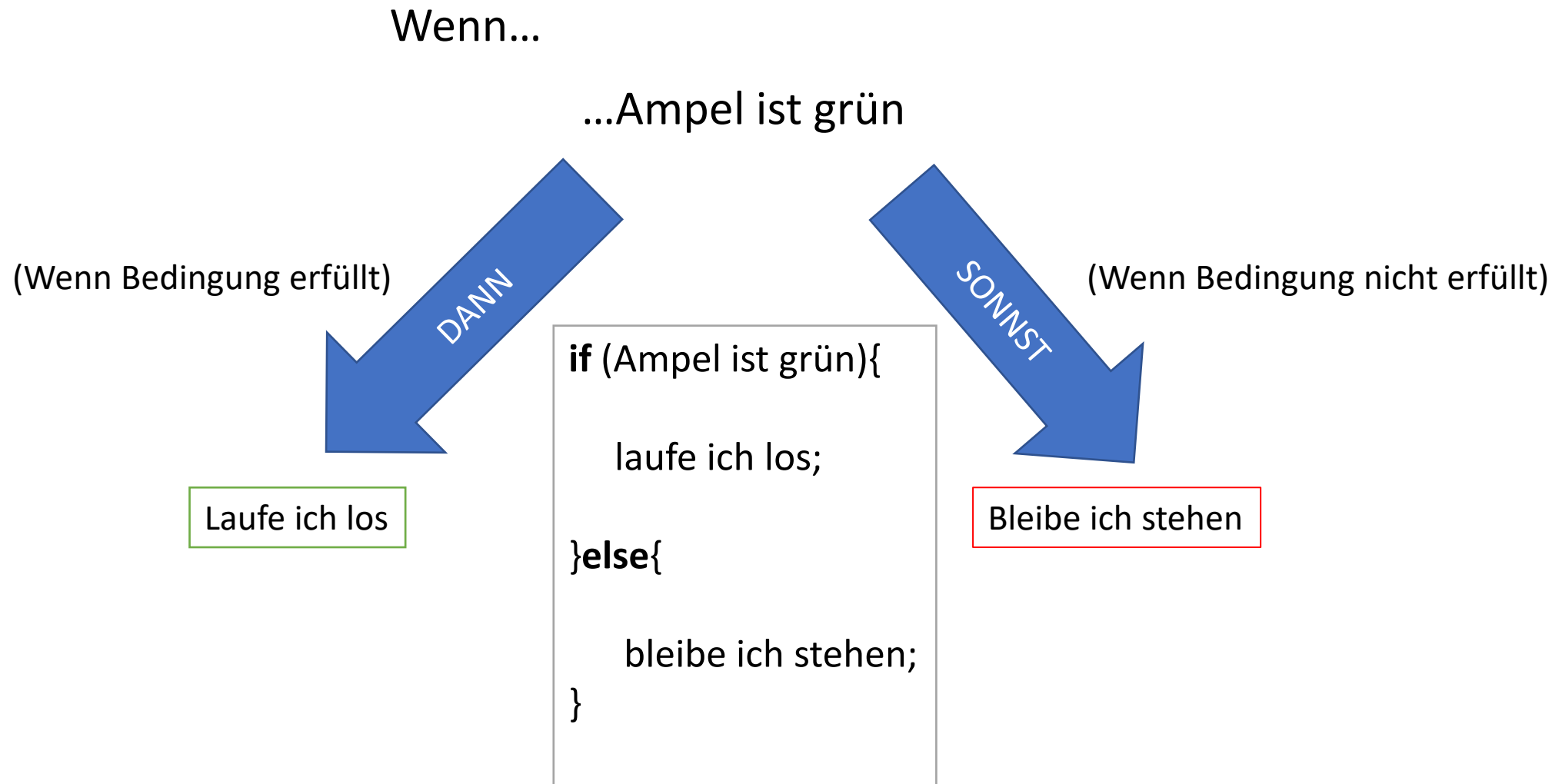
```
1  
2 boolean Variable;  
3  
4 void setup() {  
5     Variable = true;  
6 }
```

Ü: Variablen in Led Blink

Programmieren - Bedingung

- Wenn... , dann
- Knüpfen des Programmablaufs an eine Bedingung

Programmieren - Bedingung



Programmieren - Bedingung

Ü: Blinken an Bedingung knüpfen

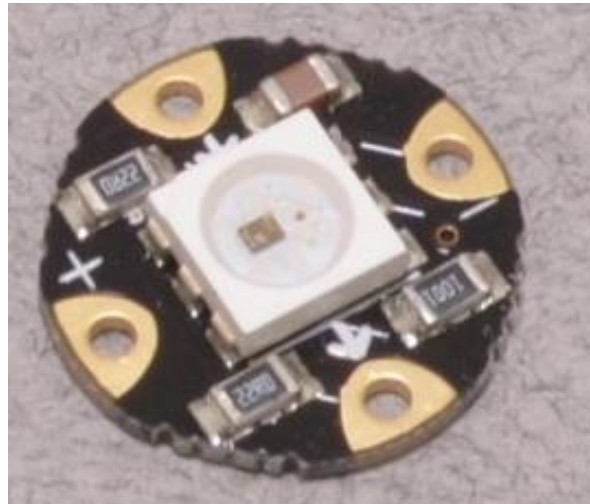
Wahrheitsvergleiche:

- Gleichheit: ==
- Größer/Kleiner: < >
- Größer gleich / Kleiner gleich: <= >=

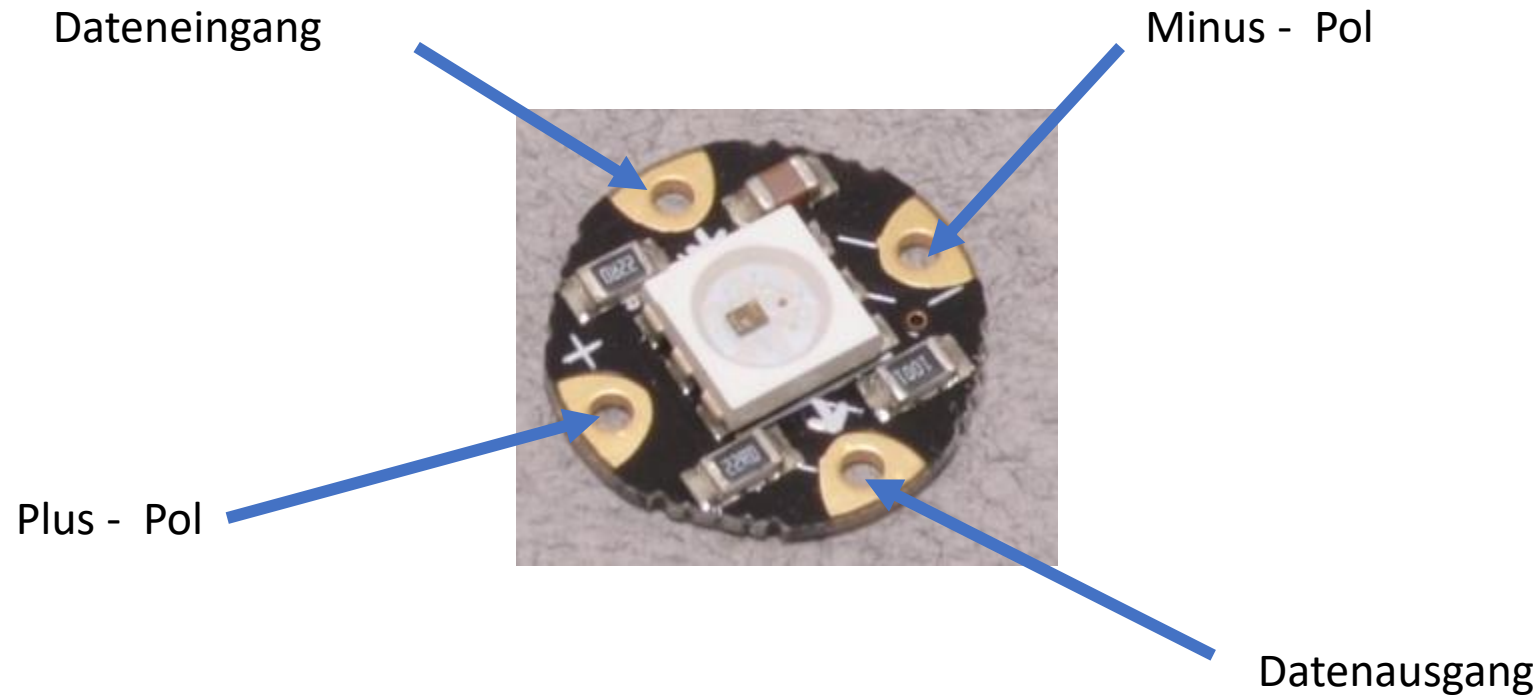
```
if (Ampel == grün){  
    laufe ich los;  
}else{  
    bleibe ich stehen;  
}
```

Programmieren - LED

- Einzelne LEDs anschließen -> Programmierung wie vorher
- „Neopixel“ ??



Programmieren - NeoPixel



Programmieren - NeoPixel

- Mit Bibliothek

```
1 #include <Adafruit_NeoPixel.h>
2
3
```

Ü: Bibliothek einbinden

Programmieren - NeoPixel

1. „LED“ Objekt erstellen

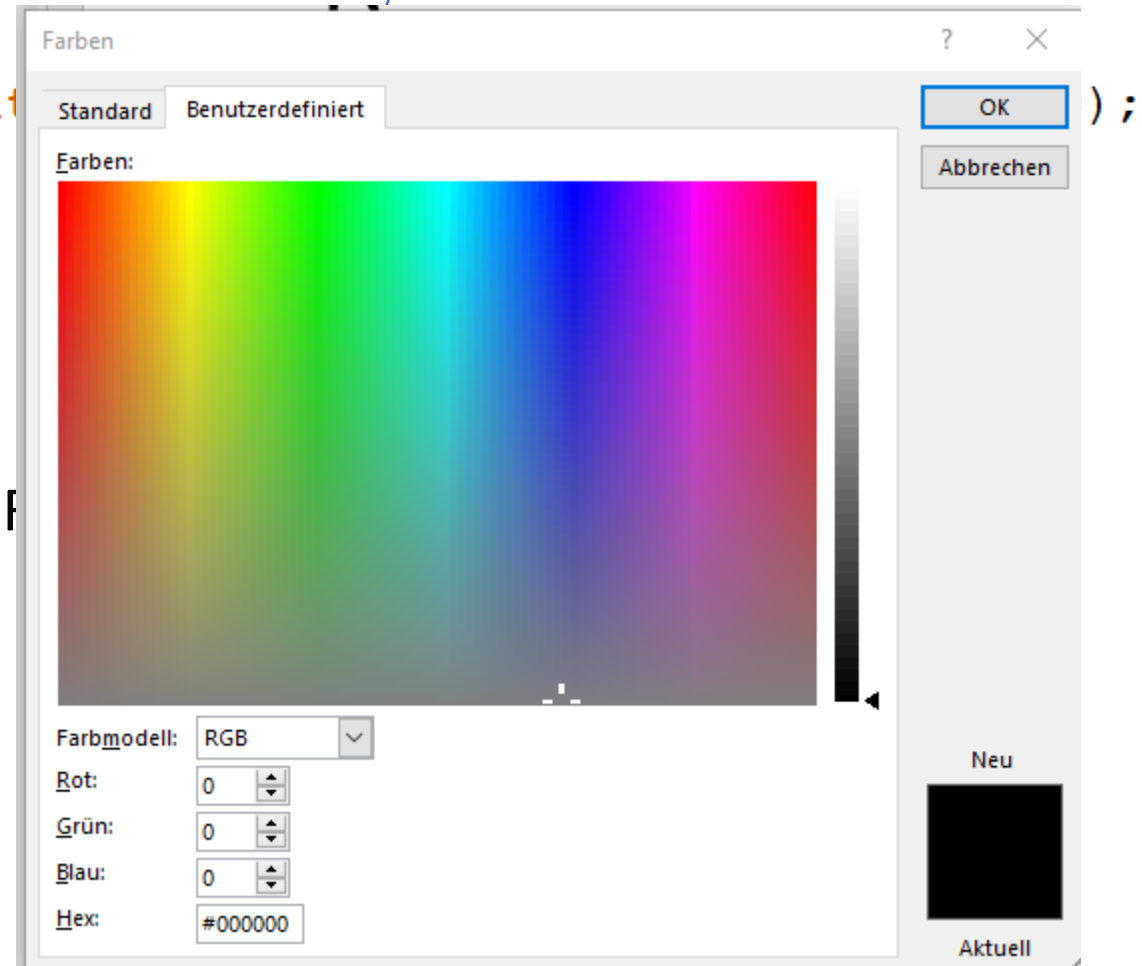
```
Adafruit_NeoPixel led = Adafruit_NeoPixel(1, <pin>, <led_type>);  
led.begin();
```

2. Farbe erstellen

```
long colorVar1 = 0;  
colorVar1 = led.Color(0, 0, 0);
```

0-255

Anzahl LEDs

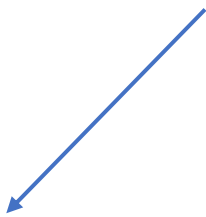


Programmieren - NeoPixel

3. Farbe LED zuordnen

```
led.setPixelColor(0, colorVar1);
```

```
colorVar1 = led.Color(0,0,0);
```

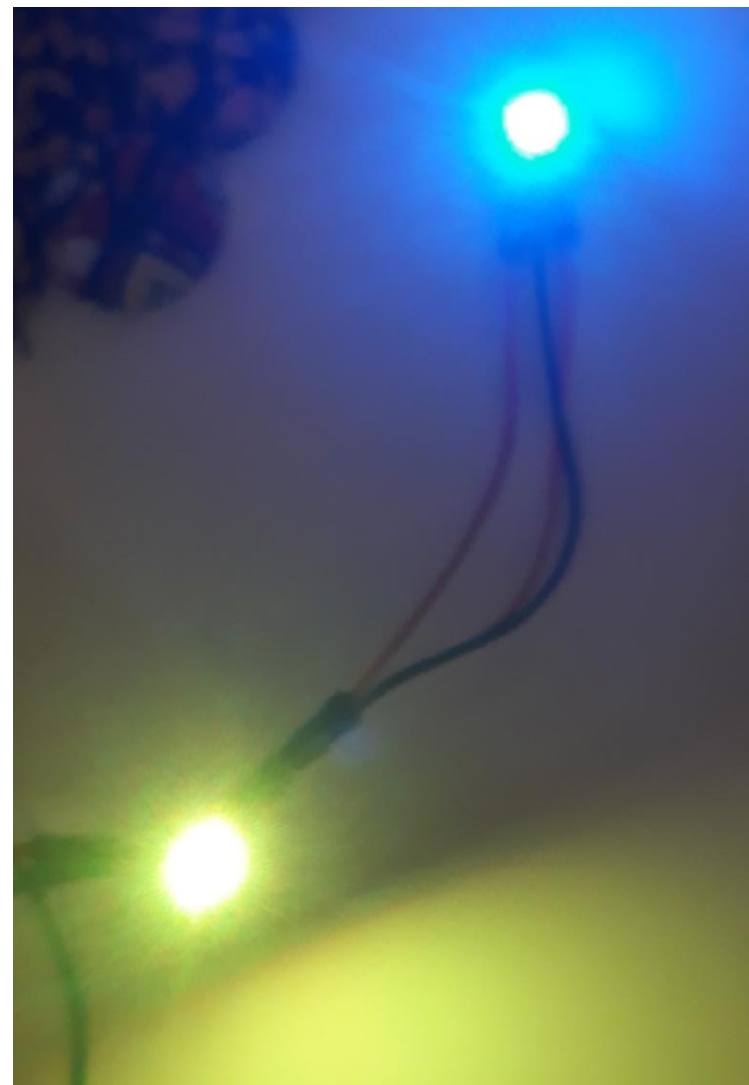


Welche LED? Bei „0“ mit Zählen beginnen!!!

4. Farben anzeigen

```
led.show();
```

```
1 #include <Adafruit_NeoPixel.h>
2
3 Adafruit_NeoPixel led = Adafruit_NeoPixel(2,2, NEO_GRB + NEO_KHZ800);
4 long colorVar1 = 0;
5 long colorVar2 = 0;
6
7 void setup() {
8   led.begin();
9 }
10
11 void loop() {
12
13   colorVar1 = led.Color(120,230,0);
14   colorVar2 = led.Color(0,100,255);
15
16   led.setPixelColor(0,colorVar1);
17   led.setPixelColor(1,colorVar2);
18   led.show();
19
20 }
```



Programmieren - NeoPixel

Ü: LED Blinken lassen

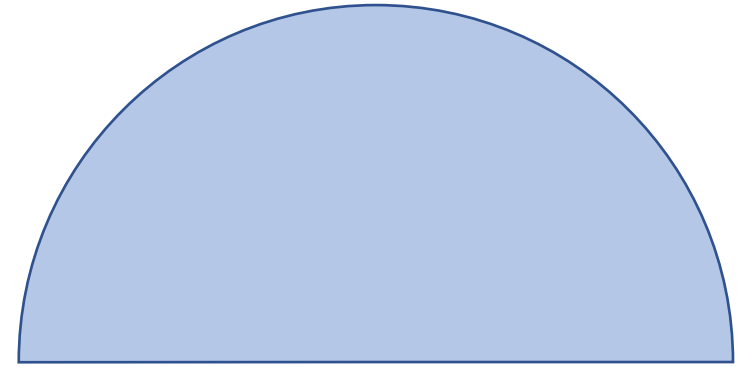
```
1 #include <Adafruit_NeoPixel.h>
2
3 Adafruit_NeoPixel led = Adafruit_NeoPixel(2, 2, NEO_GRB + NEO_KHZ800);
4 long colorVar1 = 0;
5 long colorVar2 = 0;
6
7 void setup() {
8     led.begin();
9 }
10
11 void loop() {
12
13     colorVar1 = led.Color(120, 230, 0);
14     colorVar2 = led.Color(0, 100, 255);
15
16     led.setPixelColor(0, colorVar1);
17     led.setPixelColor(1, colorVar2);
18     led.show();
19
20 }
```

Anzahl LEDs

Daten Pin am Arduino

Ü: Neo-Pixel Blinken lassen

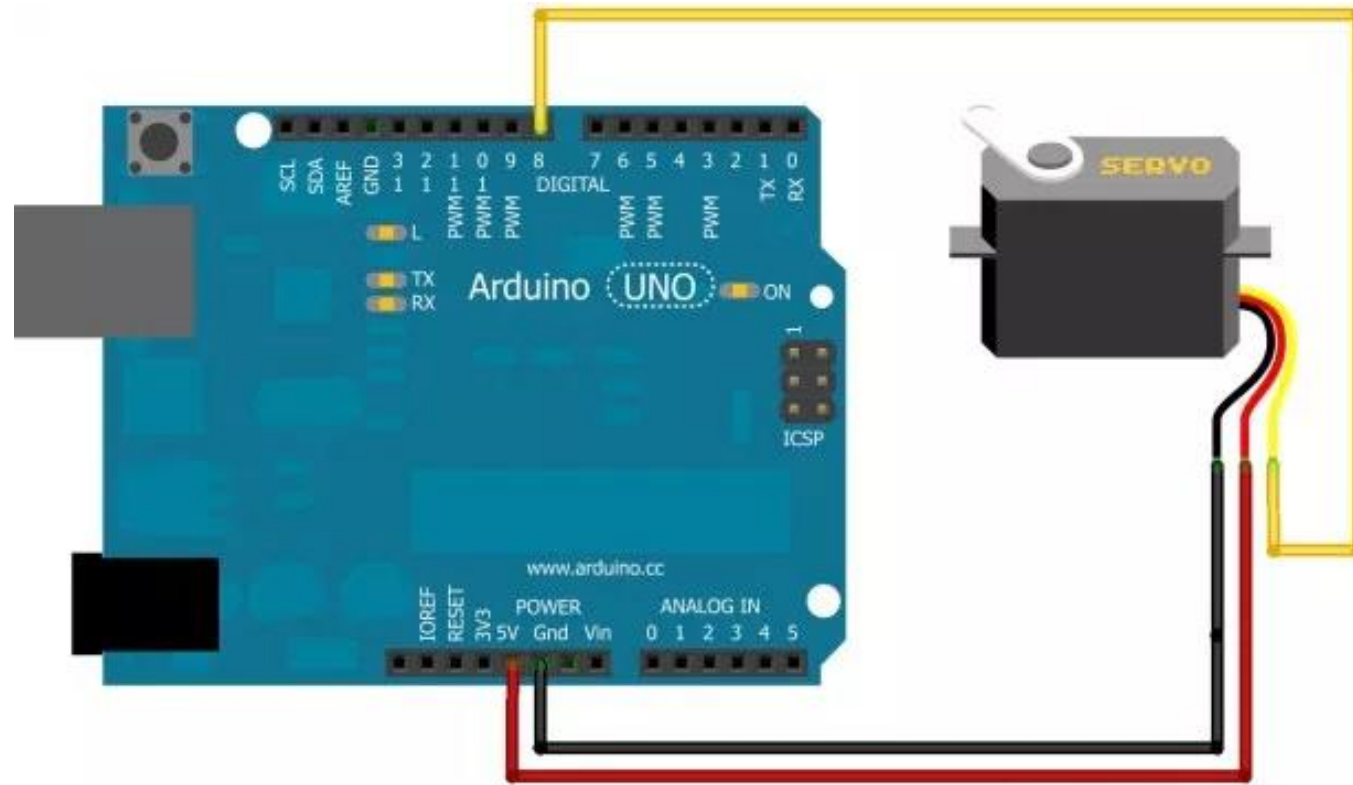
Programmieren - Motor



180° Bewegung

-> Grad genau steuerbar

Programmieren - Motor



Programmieren - Motor

- 1. Bibliothek einbinden

```
6 | #include <Servo.h>
```

Ü: Bibliothek einbinden

- 2. Servo Objekt erstellen

```
Servo servo;
```

- 3. Setup Servo

```
servo.attach(3);
```

Daten Pin am Arduino

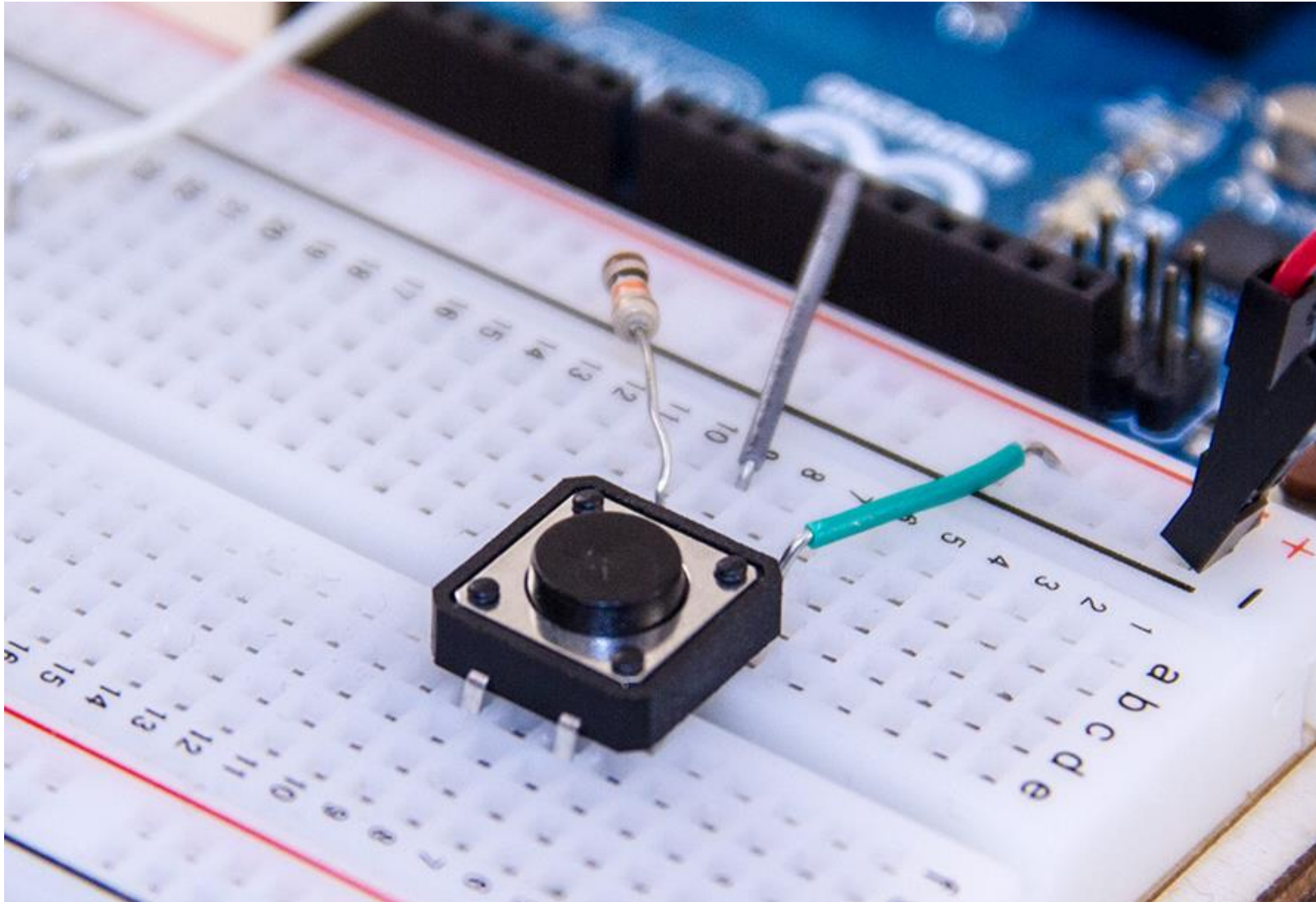


- 4. Gradzahl einstellen

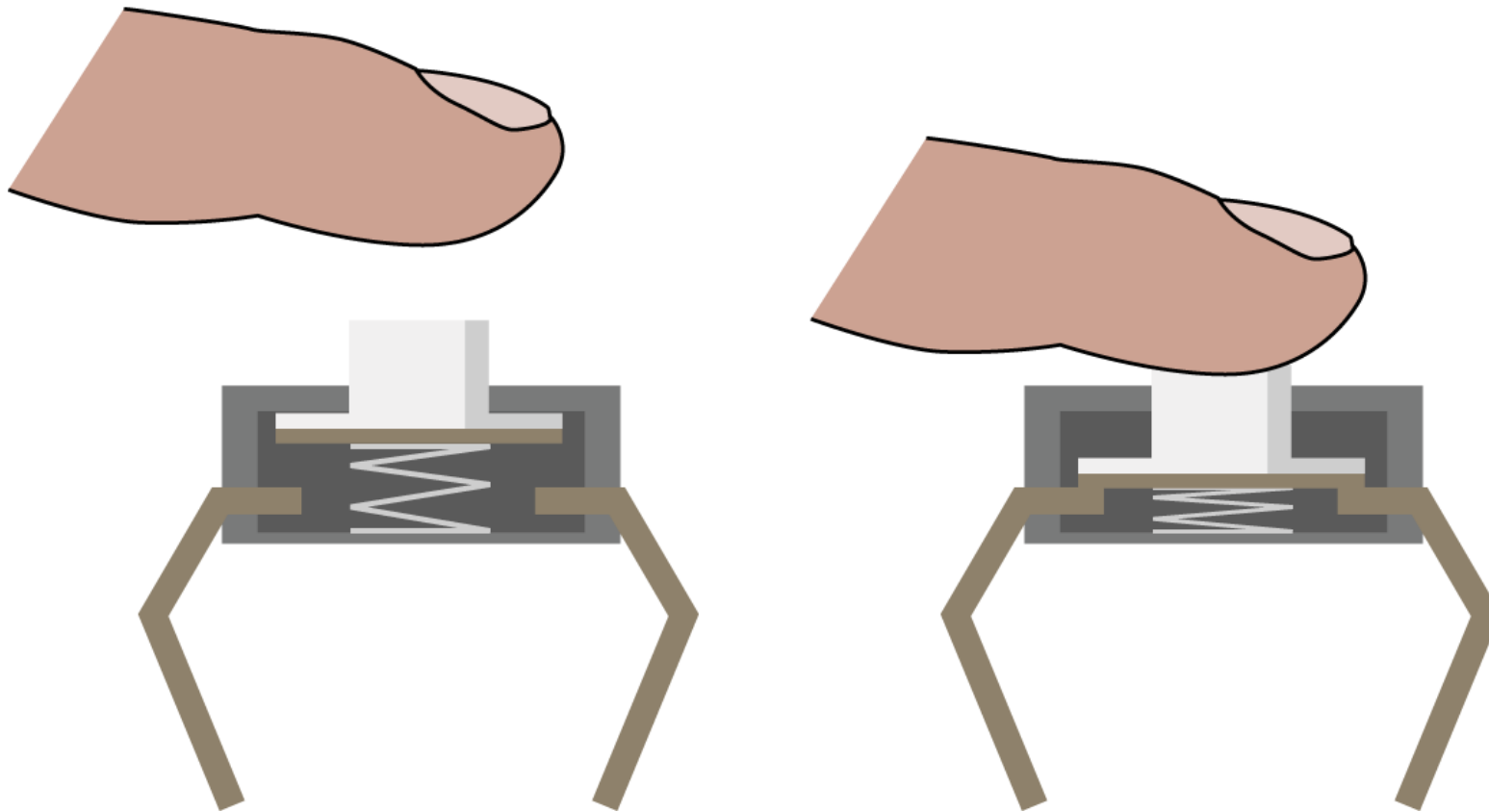
```
servo.write(90);
```

```
1 #include <Servo.h>
2 Servo servo;
3 void setup() {
4     // put your setup code here, to run once:
5     servo.attach(3);
6 }
7
8 void loop() {
9     servo.write(0); //Position 1 ansteuern mit dem Winkel 0°
10
11     delay(3000); //Das Programm stoppt für 3 Sekunden
12
13     servo.write(90); //Position 2 ansteuern mit dem Winkel 90°
14
15     delay(3000); //Das Programm stoppt für 3 Sekunden
16
17
18 }
```

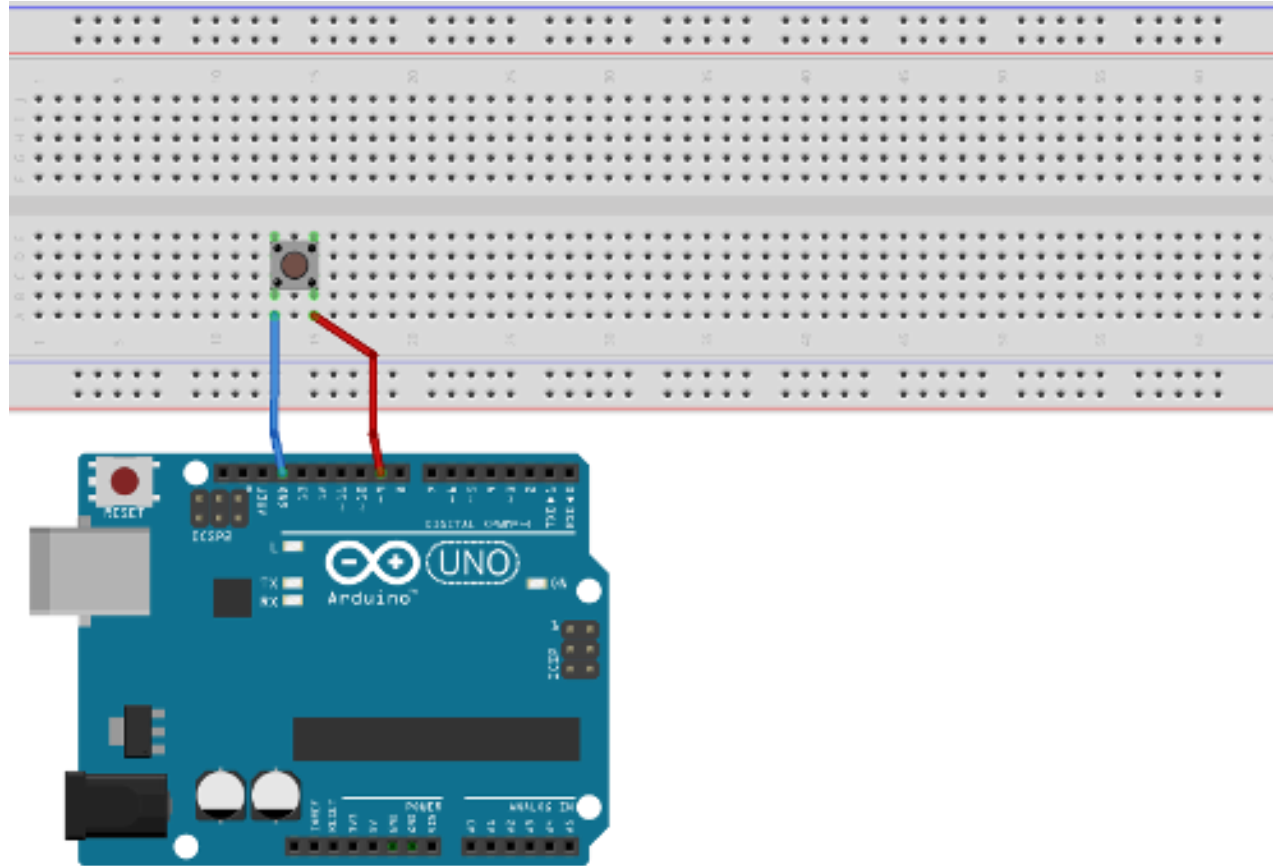
Programmieren - Taster



Programmieren - Taster



Programmieren - Taster



fritzing

Programmieren - Taster

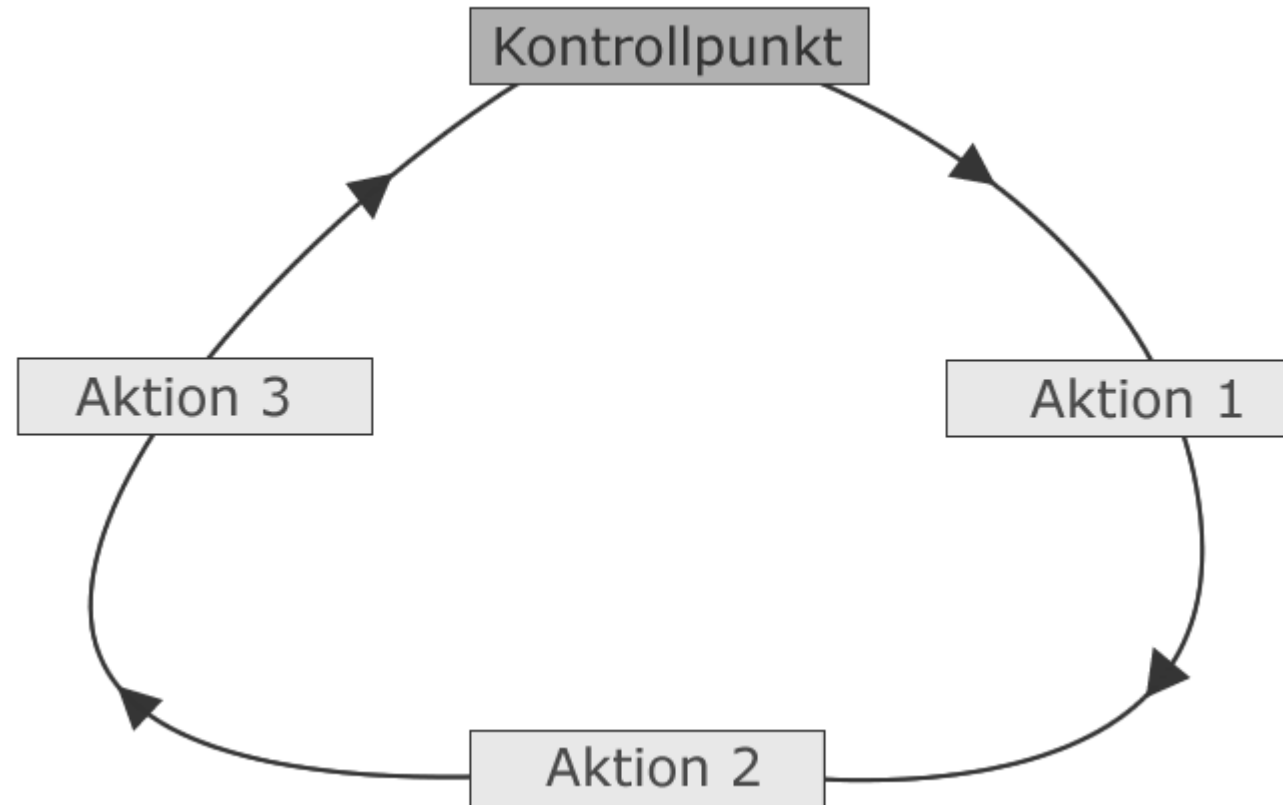
- Eingang auslesen

```
pinMode(12, INPUT);
```

```
taster = digitalRead(12);
```

```
1 boolean toggle;  
2 int tasterPin = 12;  
3  
4 void setup() {  
5     pinMode(tasterPin, INPUT);  
6     Serial.begin(9600);  
7  
8 }  
9  
10 void loop() {  
11     toggle = digitalRead(tasterPin);  
12  
13     if(toggle == HIGH) {  
14         Serial.println("An");  
15     }else{  
16         Serial.println("Aus");  
17     }  
18  
19     delay(100);  
20
```

Programmieren - Schleifen



Programmieren - Schleifen

```
for (int i=0; i<10; i++) {  
    // Anweisungen  
}
```

1.	int i = 0;
2.	while (i<=10) {
3.	// Anweisungen
4.	}

Programmieren

```
pinMode(13, OUTPUT) / pinMode(14, INPUT)
```

```
delay(1000)
```

```
digitalWrite(13, HIGH) / digitalWrite(13, LOW)
```

```
digitalRead(14)
```

Besseres Delay

```
1 *
2 Blink without Delay
3 const int ledPin = LED_BUILTIN; // the number of the LED pin
4 int ledState; // ledState used to set the LED
5
6 unsigned long previousMillis = 0; // will store last time LED was updated
7 const long interval = 1000; // interval at which to blink (milliseconds)
8
9 void setup() {
10 // set the digital pin as output:
11 pinMode(ledPin, OUTPUT);
12 }
13
14 void loop() {
15 unsigned long currentMillis = millis();
16
17 if (currentMillis - previousMillis >= interval) {
18 // save the last time you blinked the LED
19 previousMillis = currentMillis;
20
21 // if the LED is off turn it on and vice-versa:
22 if (ledState == LOW) {
23 ledState = HIGH;
24 } else {
25 ledState = LOW;
26 }
27
28 // set the LED with the ledState of the variable:
29 digitalWrite(ledPin, ledState);
30 }
31 }
32
```