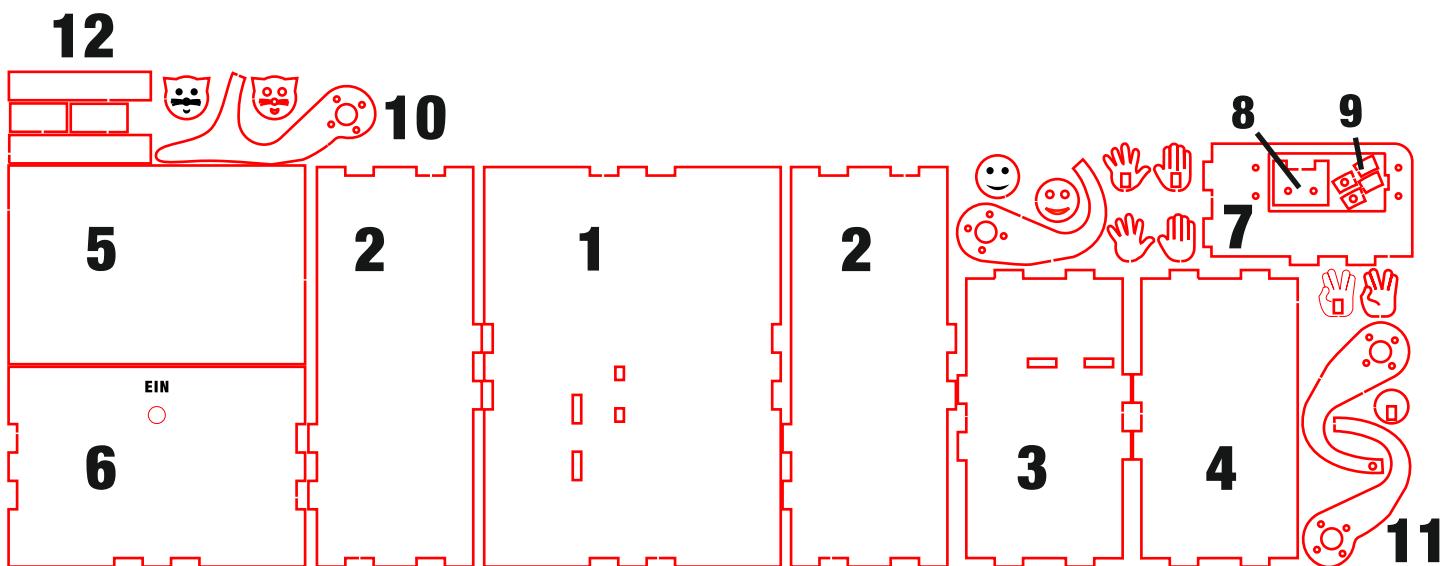


Bauanleitung „Useless Box“ v 1.2

Kai Laborenz, August 2017

Benötigte Teile



Lasercutteile:

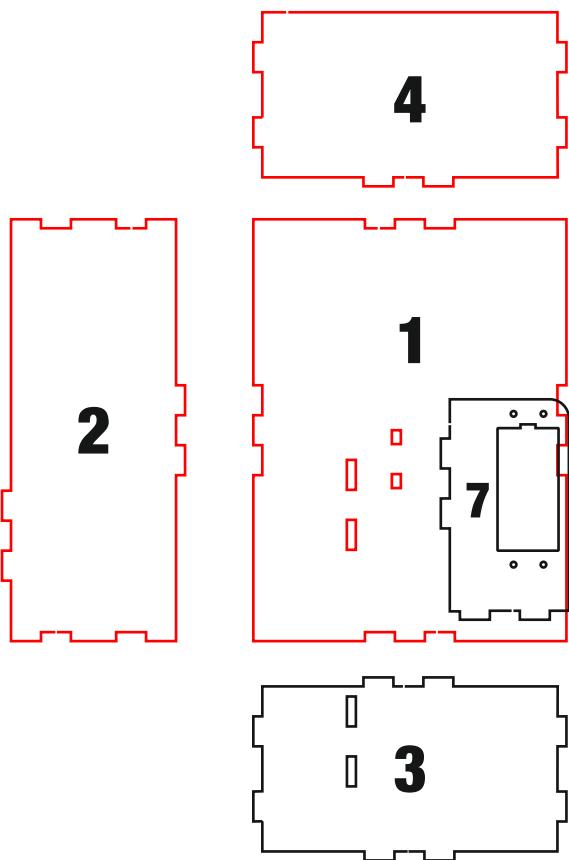
- 1 Bodenplatte
- 2 Seitenwände längs
- 3 Seitenwand quer für Servobefestigung
- 4 Seitenwand quer
- 5 lose Topplatte
- 6 feste Topplatte
- 7 Servobefestigung
- 8 Mikroschalterbefestigung
- 9 Verstärkung Hebel
- 10 Hebel für Hände/Köpfe
- 11 Hebel allein

Sonstige Materialien:

- 1x Servo (z.B. Conrad 410 Standardservo)
- 1x Kippschalter 2xUM
- 1x Mikroschalter
- 2x Batterien AA + Halter, Clip und Kabel
- oder
- 1x Kippschalter 1xUM
- 1x Widerstand 10kΩ
- 1x Arduino Nano
- 1x 9V Batterie und Clip
- 4x Batterien AA + Halter, Clip und Kabel
- Kabellitze, Kleber, Schrauben

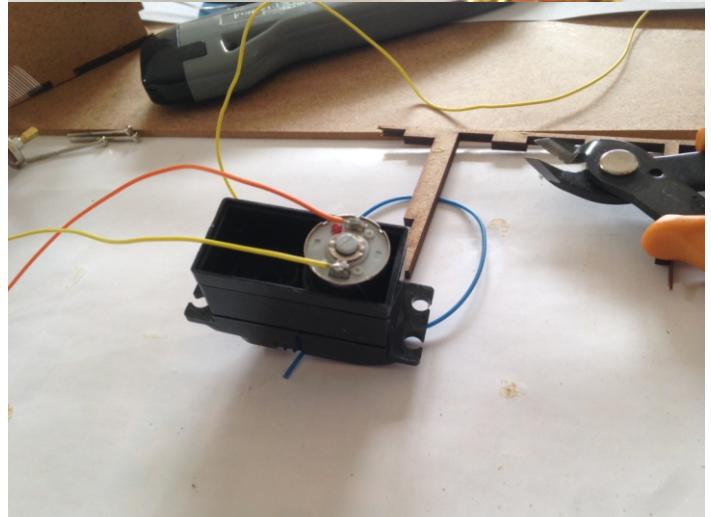
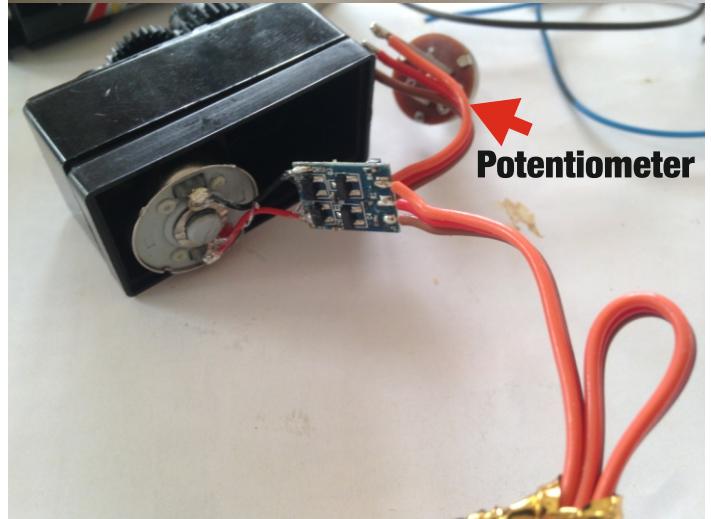
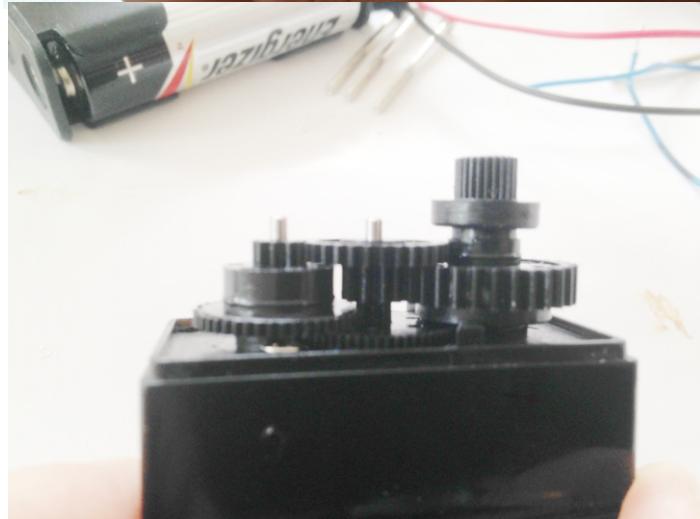
Vormontage der Box:

- Als erstes bauen wir die Box teilweise zusammen, so dass wir die Technik später noch hinzufügen können.
- Dazu leimen wir die Grundplatte (1), eine (!) Seitenwand (2), die beiden Querwände (3 und 4) und die Servohalterung (7) zusammen. Die Orientierung der Seitenwände ist wichtig.
- Zuerst die Seitenwand und die Querwand (4), dann Servohalterung (7) in die Schlitze einpassen und danach (3) verleimen.



Umbau des Servos (nur für die Nicht-Arduino-Version):

- Wir schrauben das Servo auf (Vorsicht, dass nicht alles auseinander fällt), entfernen die Elektronik (inklusive des kleinen runden Potentiometers mit den drei Kabeln).
- Wir entfernen vom dicksten Getrieberad den schwarzen Block und modellieren die fehlenden Zahnräder mit Feile und Schleifpapier nach (der schwierigste Teil des Abends) und setzen dann das Getriebe wieder zusammen.
- Dann löten wir neue Kabel an die beiden Anschlüsse des Motors, stecken das Servo wieder zusammen und testen, ob der Motor rund läuft, indem wir die beiden AA-Batterien in ihre Halterung setzen und die Kabel an die Anschlüsse halten (egal, wie herum). Wenn alles rund läuft, schrauben wir das Servo wieder zusammen.



Schalterverdrahtung (nur für die Nicht-Arduino-Version):

Der Kippschalter für die Außenmontage wird verdrahtet wie folgt:

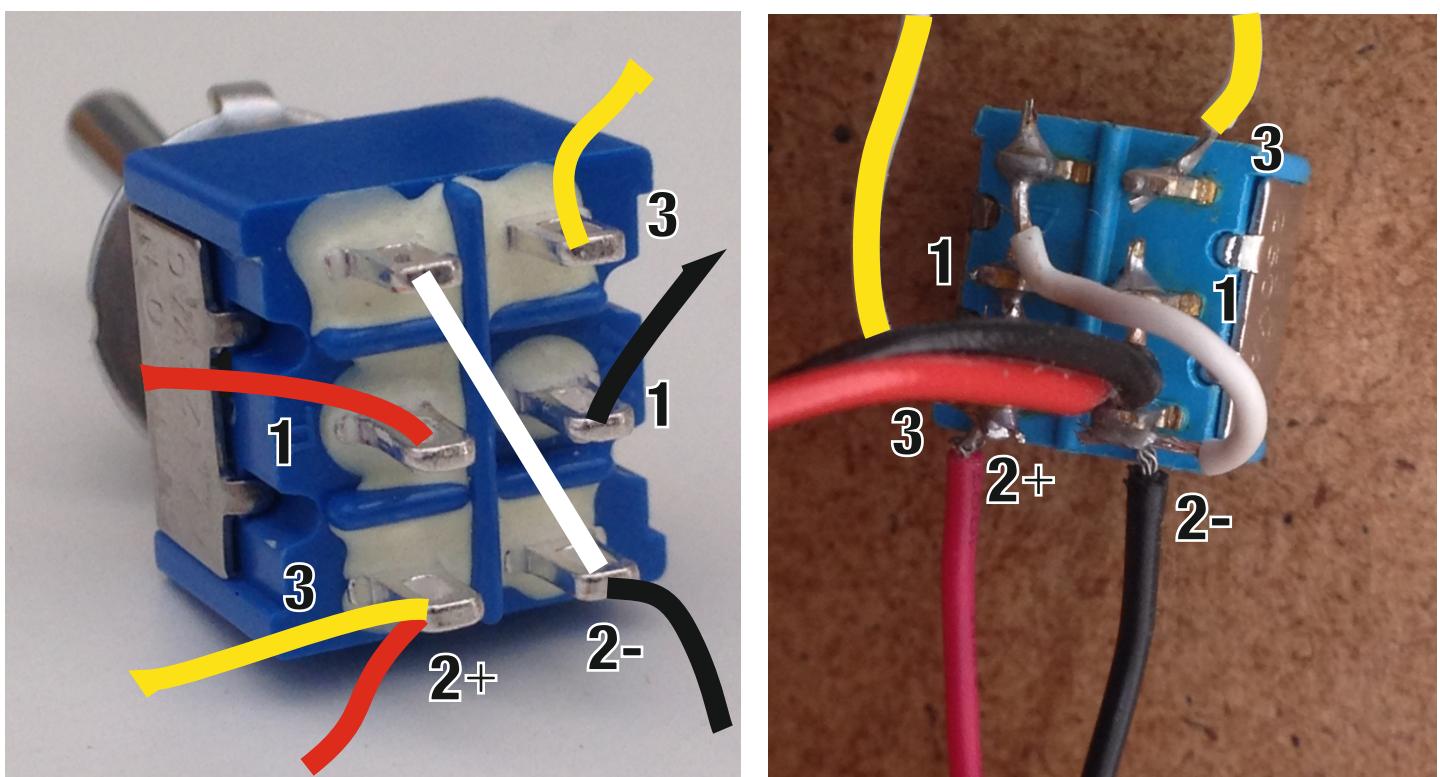
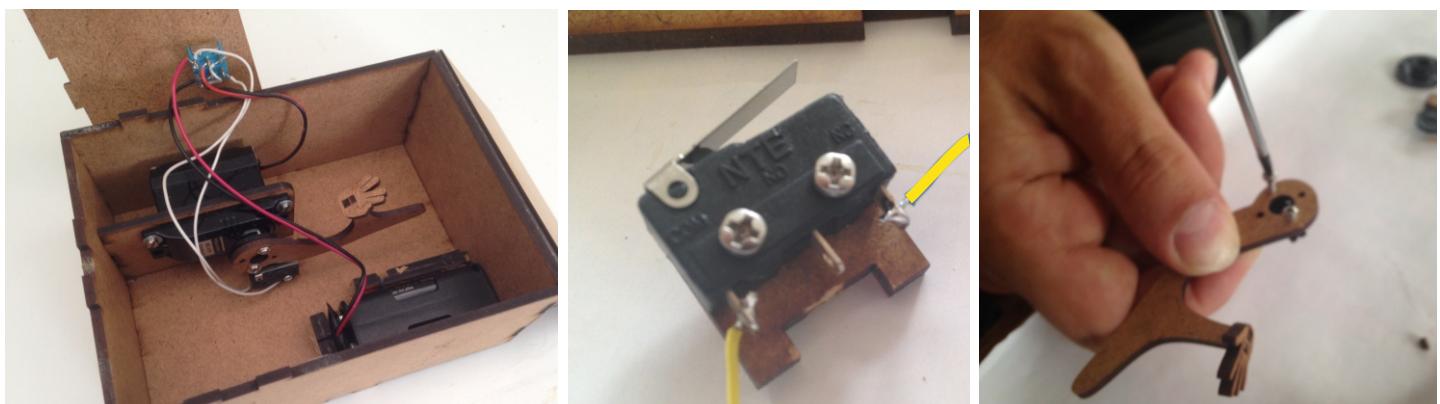
- 1 Motorkabel (rot/schwarz - erst einmal egal, wie herum)
- 2 Batterieclip (auf + (rot) und - (schwarz) achten)
- 3 Endschalter (gelb bzw. weiss, egal, wie herum, es müssen die beiden äußeren Beinchen am Endschalter verwendet werden)

Zusätzlich ein Kabel quer von Minus (unten rechts) zum leeren Pin oben links. Die komplette Verdrahtung ist auf der folgenden Seite noch einmal zu sehen.

Der Hebel mit gewähltem Motiv wird auf das Servohorn geschraubt bzw. geklebt, dann werden Servo und Endschalter montiert und der Aufbau wird getestet. Wenn alles ok ist, die Teile (2) und (6) verkleben und die Klappe (5) mit Tesafilm befestigen. Die Batterie in einer Ecke mit den Teilen (12) gegen Verrutschen sichern. **Fertig!**

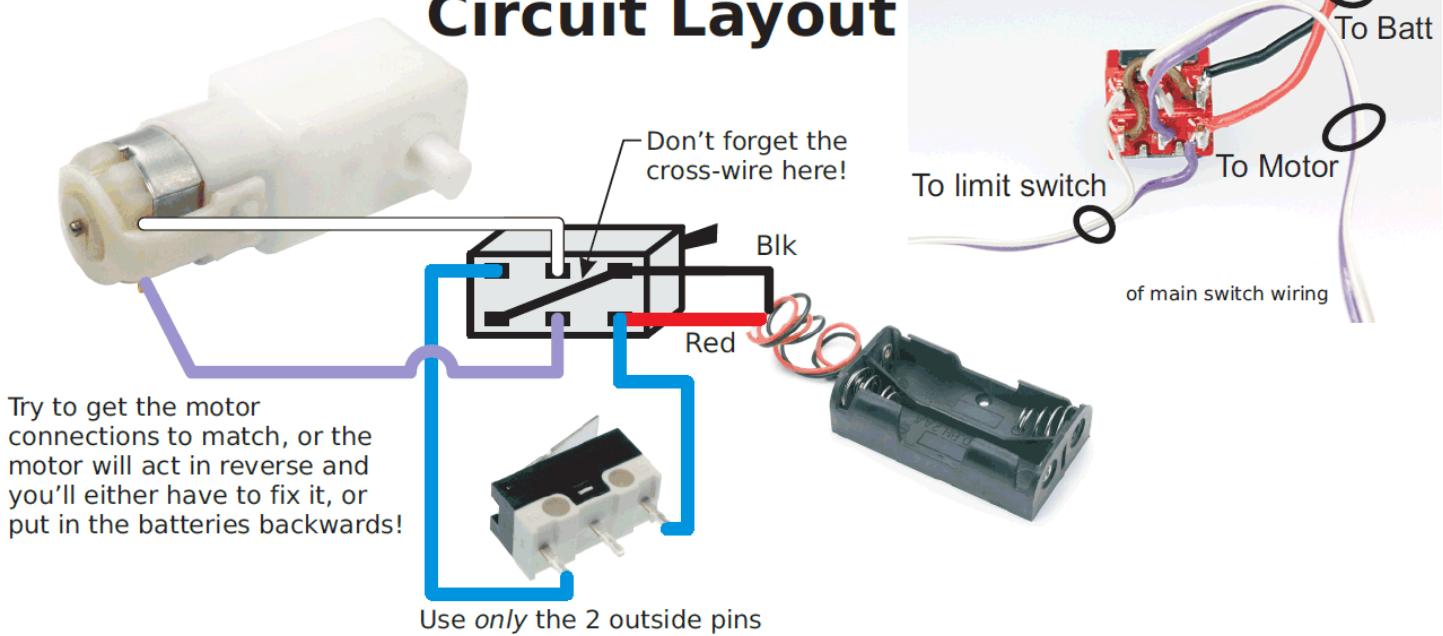
Fehlerbehebung:

Beim Anschalten der Batterie sollte der Hebel nach unten drehen, bis er an den Endschalter stößt und dann stoppen. Wenn man den Kippschalter betätigt, sollte er nach oben laufen, bis man den Schalter in die andere Position schaltet. Wenn der Hebel bei Anschließen der Batterie gleich nach oben läuft, den Kippschalter betätigen. Wenn der Hebel nach unten läuft und nicht stoppt, Batterie schnell abnehmen und die beiden Motorkabel (1) anders herum anlöten.



Verdrahtung für die Nicht-Arduino-Version:

Circuit Layout

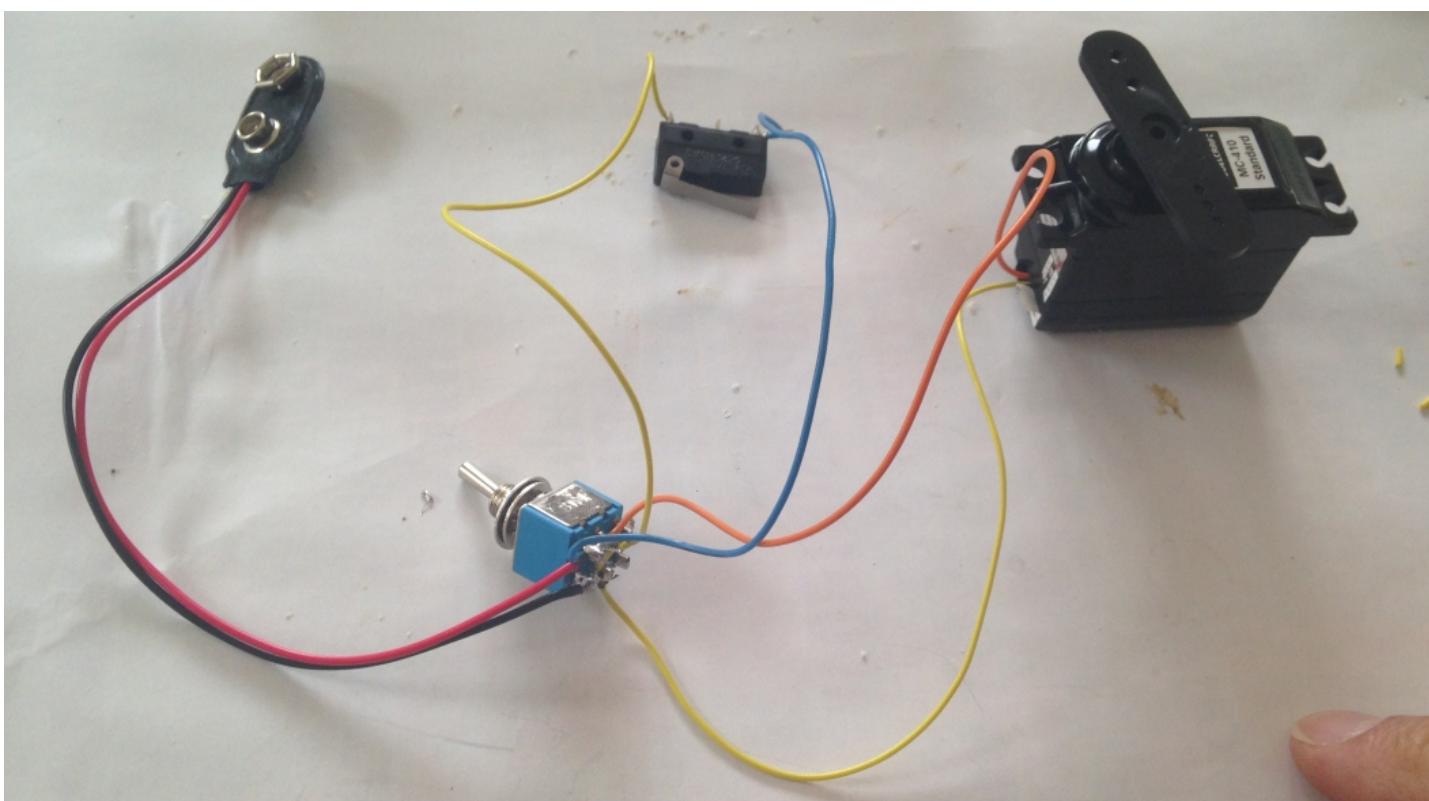


Wie funktioniert die Schaltung?

Der Kippschalter schaltet die beiden mittleren Anschlüsse je nach Stellung nach rechts oder links. Der Endschalter verbindet die beiden äußeren Anschlüsse, wenn er nicht betätigt ist.

In Ruhestellung ("AUS") sind die Motoranschlüsse nach rechts geschaltet (weiss an rot) und der Motor dreht sich so lange, bis er den Hebel gegen den Endschalter drückt und dieser den Strom unterbricht.

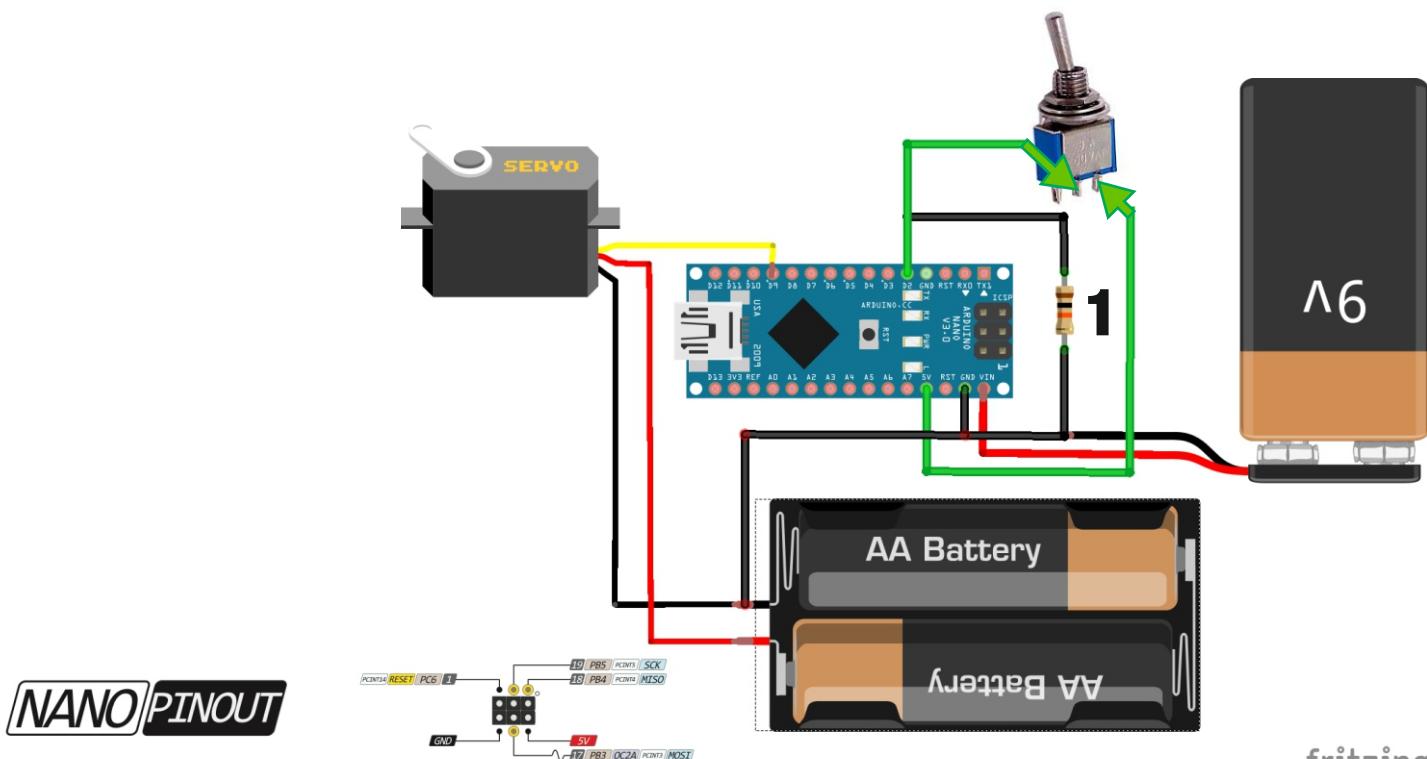
Wenn der Kippschalter dann in die andere Richtung gekippt wird ("EIN"), werden die Motoranschlüsse direkt mit der Batterie verbunden (und zwar diesmal weiss an schwarz) und der Motor dreht in die andere Richtung, bis der Hebel den Kippschalter wieder auf „AUS“ kippt und der Motor wieder anders herum angeschlossen wird (der Endschalter ist ja inzwischen nicht mehr gedrückt und der Motor kann wieder drehen). Der Hebel bewegt sich zurück und betätigt wieder den Endschalter und der Motor stoppt.



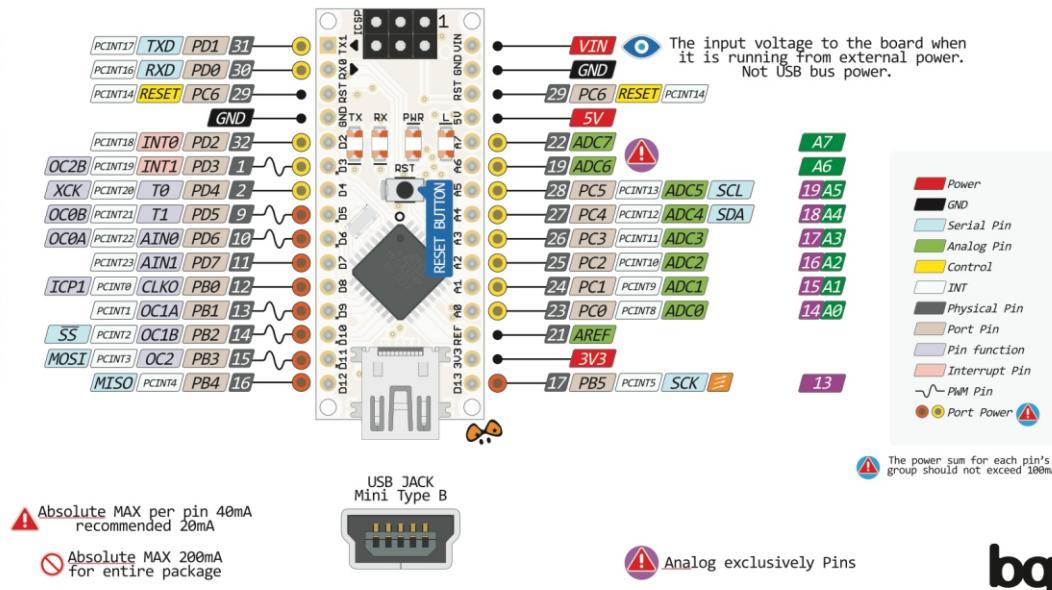
Verdrahtung für Arduino-Version:

- Der Schalter wird mit Pin D2 des Arduino und mit 5V verbunden.
- Der 10k-Widerstand (1) wird ebenfalls mit Pin D2 und GND ("Minus" / Masse) verbunden.
- Der Batterieclip 9V wird mit GND und Vin (Eingangsspannung) verbunden.
- Der Servo wird verbunden mit der Batteriebox, GND (braunes Kabel) und D9 (oranges Kabel).

Danach übertragen wir das Programm über die Arduino-IDE per Mini-USB auf den Arduino. Die konkreten Servopositionen (Winkelangaben, im Code markiert) müssen evtl. angepasst werden. Dazu erst einmal zum Testen nicht den Hebel (10) auf das Servo strecken, sondern ein Stück Malerkrepp als Fahne. Wenn der Kippschalter „aus“ ist, muss der Hebel horizontal oder leicht nach unten im Gehäuse verschwinden. Wenn der Kippschalter „ein“ ist (in Richtung Klappe (5) zeigt), ist die Servoposition eher nach oben ausgerichtet. Wenn es mit der Fahne gut aussieht, den Hebel aufstecken, bei noch nicht verklebtem Deckel (6) testen und Position ggf. so lange anpassen bis der Hebel den Schalter betätigt und dann wieder im Gehäuse verschwindet.



für später...



Code für die Arduino-Version:

```
// Useless Box v1.1
//

// include library to operate servos
#include <Servo.h>

// const and variables definitions
const int buttonPin = 2;
int buttonState = 0;

Servo myservo;          // create servo object to control a servo
                        // a maximum of eight servo objects can be created
int pos;                // variable to store the servo position
long timeDelay;         // variable to store random delay times for servo operation
int counter = 0;          // counter for counting the switch uses
int counterSwitch = 0;

void setup()
{
    pinMode(buttonPin, INPUT);
    myservo.attach(9);      // attaches servo on pin 9 to the servo object

    Serial.begin(9600); // initialize serial communication at 9600 bits per second:
}

void loop()
{
    timeDelay = 0;
    buttonState = digitalRead(buttonPin);

    if (buttonState == HIGH) { // not switched

        if (counterSwitch == 0) {
            counter = counter + 1; // advance counter
            counterSwitch = 1;
            Serial.println(counter);
        }

        for(pos = myservo.read(); pos >=20; pos -= 1) {
            myservo.write(pos);
        }
    } else { // switched

        // if (counter % 2 == 0) { timeDelay = 50; } // every second round goes slower

        for(pos = myservo.read(); pos <=90; pos += 1) {
            myservo.write(pos);
            delay(timeDelay);
        }
        counterSwitch = 0;
    }
} // end main loop
```

Ruheposition für Servo (im Gehäuse)

Servoposition zum Schalter betätigen