



Elektronik und technische Informatik

HWE Übungen Semester Projekt

Dokumentation zur Projektarbeit

# Leslie Speaker

Anfangsdatum: 28.02.2023

Abgabedatum: 22.06.2023



**Projektteam:**

Niko Leimegger

Pius Bickel

**Betreuungslehrer:**

Franz Lauritsch

## Inhalt

Abstract.....	3
Pflichtenheft: .....	4
Einleitung .....	4
Arbeitsziel: .....	4
Grobe Technische Einteilung .....	5
Verwendete Entwicklungsoberflächen: .....	5
Blockschaltbild: .....	5
Hardware .....	6
BLDC-Ansteuerung .....	6
Simulation: .....	7
Technische Realisierung: .....	8
Lautsprecher und Simulation des Resonanzkörpers: .....	8
Lautsprecher: .....	8
Simulation mit SubSim: .....	9
Verstärker: .....	9
Vorverstärkung .....	10
Endstufe .....	10
Spannungsversorgung: .....	11
Simulation: .....	11
PCB: .....	12
Footswitch: .....	13
Schaltung: .....	14
PCB: .....	14
Mechanik.....	15
Gehäuse .....	15
Entwicklung .....	15
Designs .....	15
Einzelne Bauteile: .....	15
Verbindungsstück .....	15
Software .....	16
Struktogramme: .....	16
Gesamter Code: .....	18
Funktionstest .....	21
Ausgabe am Arduino: .....	21
Konklusion: .....	21

## Abstract

Aktiver / Passiver Lautsprecher für Instrumente und andere Musikquellen, bei Aktivierung des Motors entsteht ein „Phaser“ Effekt.

Der Leslie benötigt eine Betriebsspannung von 12V und mindestens 1A. Oben am Gehäuse sind 3 Steuerungsmöglichkeiten angebracht.



Abbildung 1: Steuermöglichkeiten an Oberseite des Lautsprechers

Mit dem Dreiwegeschalter können Sie auswählen welche Signalquelle am Lautsprecher ausgegeben werden soll. Der Lautsprecher kann damit aktiv und passiv betrieben werden.

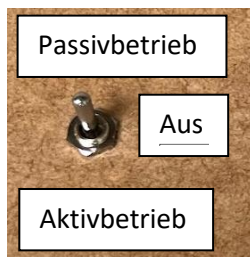


Abbildung 2: Beschrifteter Schalter

Mit dem Taster kann der Motor ausgeschaltet werden, falls ausschließlich der Verstärker verwendet werden soll, oder man keine 1A zur Verfügung hat.



Abbildung 3: Taster zur Motorsteuerung

Mit dem Drehregler kann die Lautstärke des Verstärkers geregelt werden.



Abbildung 4: Drehregler für die Lautstärke

## Pflichtenheft:

### Einleitung

Ein Leslie Speaker ist ein passiver oder aktiver Lautsprecher, welcher im Inneren Schall mit Hilfe eines drehenden Objekts („Cake“) lenkt. Somit wird ein Effekt erzeugt, der in der Audiotechnik „Phaser“ genannt wird. Leslie Speaker wurden früher bei Orgeln eingesetzt und später bei E-Gitarren. Heute gibt es Software und Digitale Schaltungen, die denselben Effekt nachahmen.

Beispiele für den Leslie gibt es in folgenden Liedern:

- Echoes – Pink Floyd
- Something – The Beatles
- Let It Loose – The Rolling Stones
- ...

Im Rahmen des HWE Übungen Unterrichts ist es Ziel, einen solchen Lautsprecher innerhalb eines Semesters zu realisieren.

### Arbeitsziel:

- a. Leslie Cabinet mit variabler Drehfrequenz
  - i. Holzgehäuse
  - ii. Styropor „Cake“
  - iii. DC-Motor Ansteuern
  - iv. Stabile Stromversorgung
  - v. Knöpfe oder Potis zum steuern
- b. Erweiterungsmöglichkeiten
  - i. Eingebauter Verstärker
  - ii. Display für Systemsteuerung
  - iii. Fußschalter

## Grobe Technische Einteilung

- c. Hardware
  - i. Stromversorgung für  $\mu$ C und Motor
- d. Software
  - i. Motor Ansteuerung mit Potentiometer und Knöpfen
- e. Mechanik
  - i. Akustisch angepasstes und Standhaftes Gehäuse Bauen

## Verwendete Entwicklungsoberflächen:

Name	Version
EAGLE	9.6.2
LTSpice XVII	17.0.35.0
Word	2304
Excel	2304
subsim.exe (by Isaac MCN)	<a href="http://users.on.net/~isaacmcn/audio/subsim/newsbsim.htm">http://users.on.net/~isaacmcn/audio/subsim/newsbsim.htm</a>
SolidWorks	2022-2023
Arduino IDE	2.1.1

## Blockschaltbild:

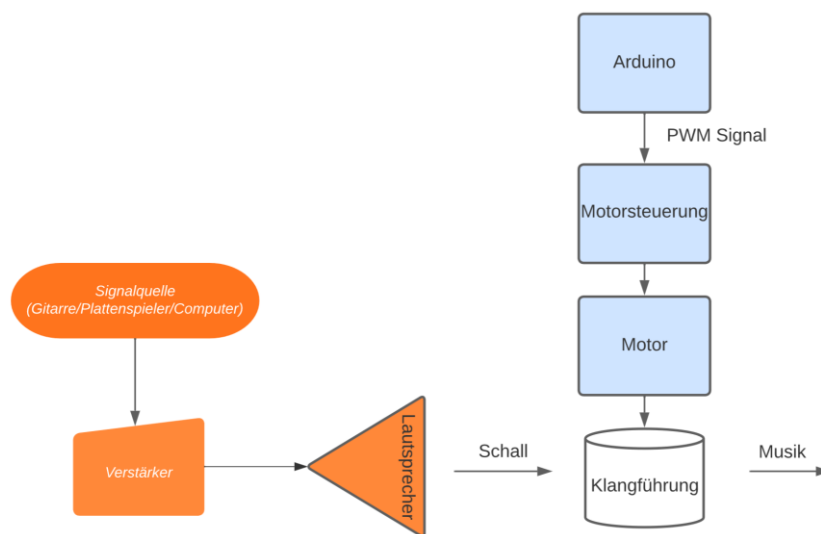


Abbildung 5: Blockschaltbild des ganzen Systems

Das Signal wird verstärkt und am Lautsprecher ausgegeben. Die Klangführung, welche durch einen Arduino mit Motorsteuerung angetrieben wird, erzeugt durch die asymmetrische Form und Drehbewegung den typischen Leslie-Sound.

## Hardware

### BLDC-Ansteuerung

Anforderungen: Steuerung, um Styroporzylinder schnell und langsam drehen zu können.

Richtwert: mindestens 60 RPM, maximal 300-400 RPM.

Für den Motor, der die Klangführung rotiert, wird ein BLDC-Motor verwendet. Die Ansteuerung erfolgt durch eine einfache Transistor Schaltung, welche mit einem Arduino Nano gesteuert wird. Der Motor benötigt mindestens einen Duty Cycle von 50%, um ohne Probleme zu laufen, deshalb wird die Betriebsspannung des Leslies für den Motor mittels eines 7805 ICs auf 5V geregelt um ihn langsamer laufen lassen zu können.

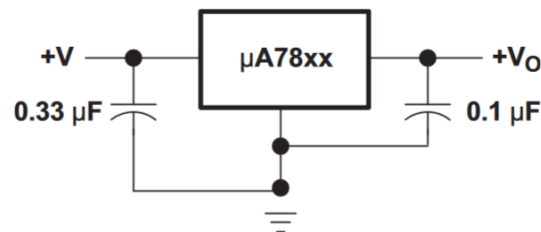


Abbildung 6: Schaltung für den 5V Spannungsregler

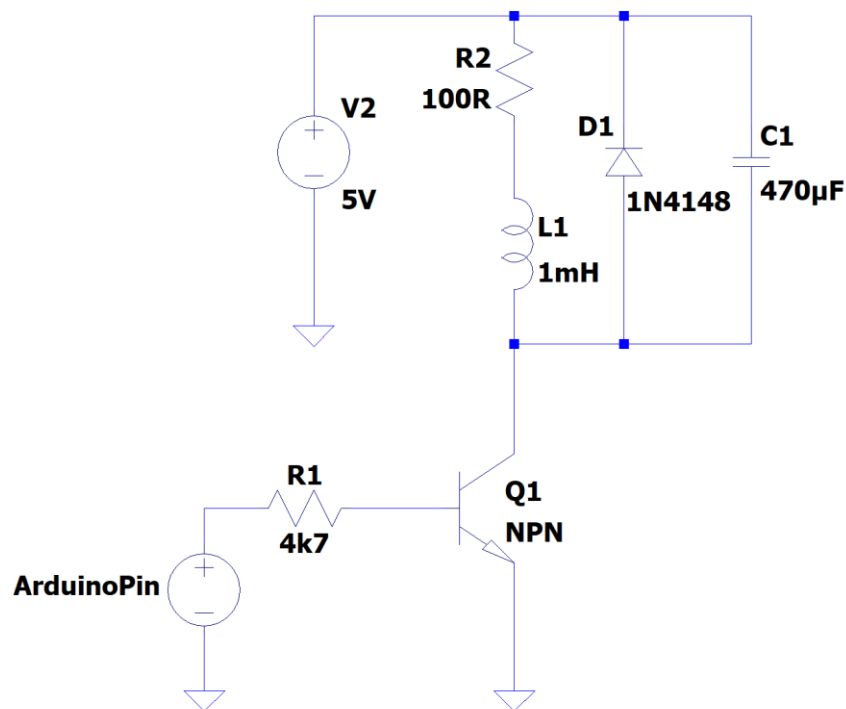


Abbildung 7: Schaltung der gesamten Motorsteuerung

## Simulation:

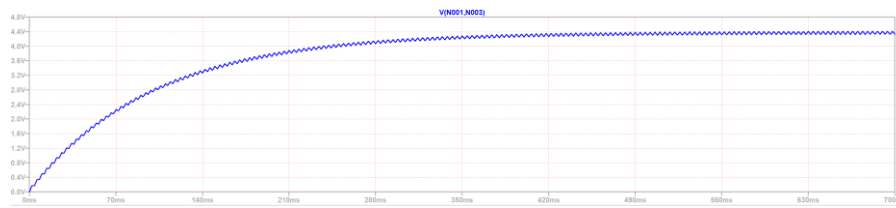


Abbildung 8: Spannung über dem Motor bei Duty Cycle 50%

Man kann erkennen, dass die Spannung allmählich steigt und dann ein maximum erreicht. Es ist noch ein leichten Rippel zu erkennen der jedoch mit einem zusätzlichen Kondensator geglättet werden könnte. In der Praxis macht dieser Rippel hier keine Probleme.

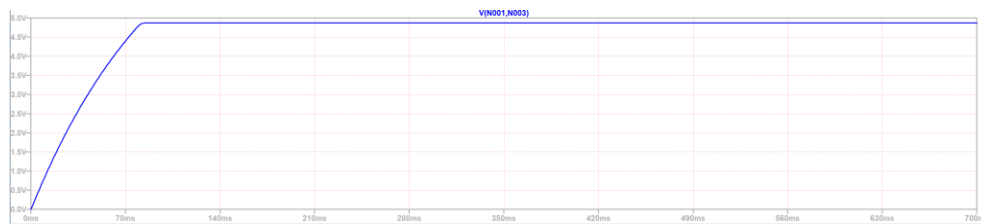


Abbildung 9: Spannung über dem Motor bei einem Duty Cycle von 100%

Bei einem Duty Cycle von 100% steigt die Spannung um einiges schneller auf das Maximum (die Betriebsspannung).

Für diese simple Schaltung lohnt es sich eine Lochrasterplatine mit folgendem Layout zu benutzen:

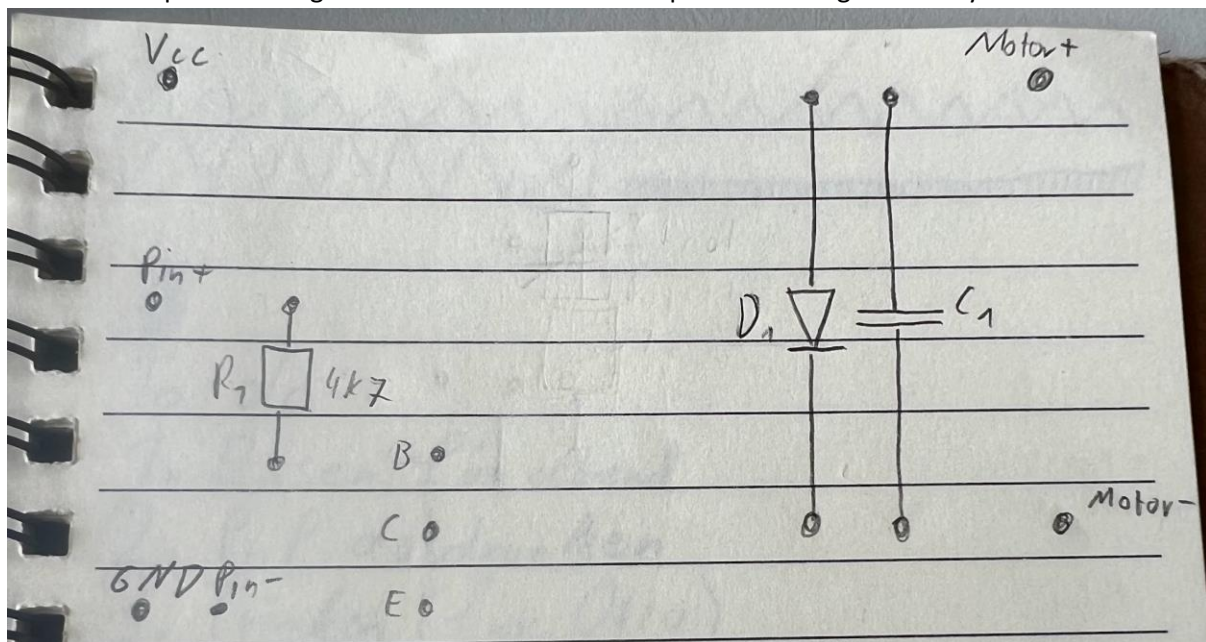


Abbildung 10: Layout der Lochrasterplatine für die Motorsteuerung



## Technische Realisierung:

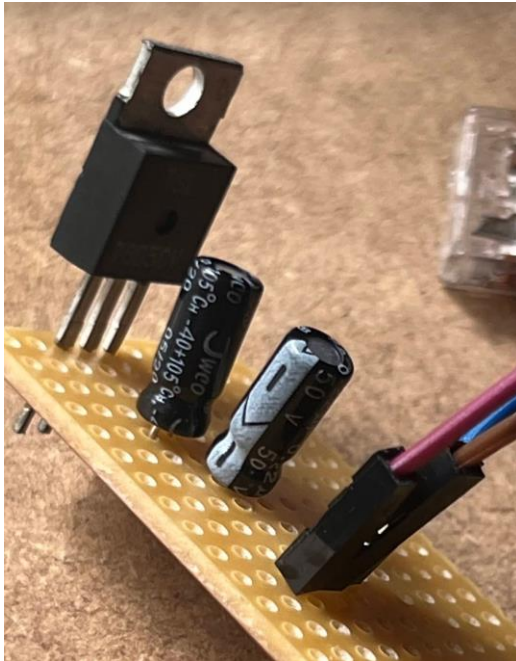
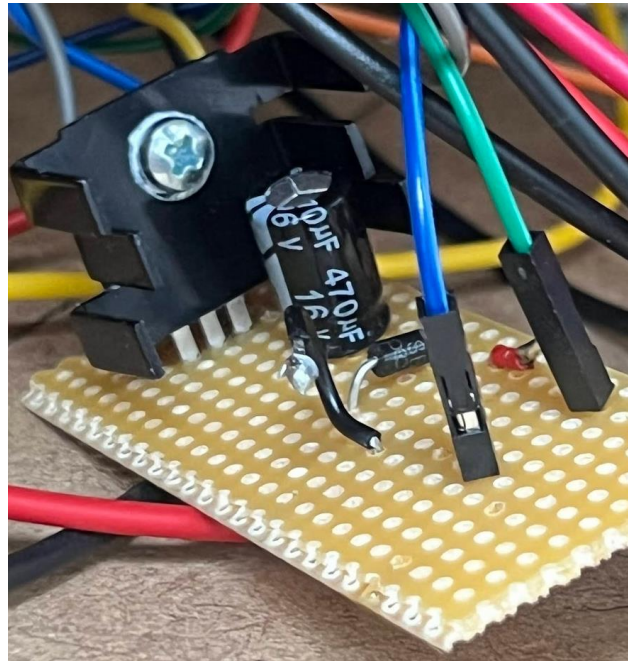
Abbildung 12: Spannungsregelung mittels  $\mu A7805$ 

Abbildung 11: Lochrasterplatine der Motorsteuerung

## Lautsprecher und Simulation des Resonanzkörpers:

Anforderungen: Niederohmig für maximale Effizienz, ausgerichtet für 20W, Vollspektrum

## Lautsprecher:

Für Gitarrenverstärker sind 8 Zoll Lautsprecher üblich. Um einen maximalen Wirkungsgrad des Verstärkers zu erzielen, ist es sinnvoll, einen niederohmigen Lautsprecher zu benutzen wie hier mit einer Impedanz von 4 Ohm.

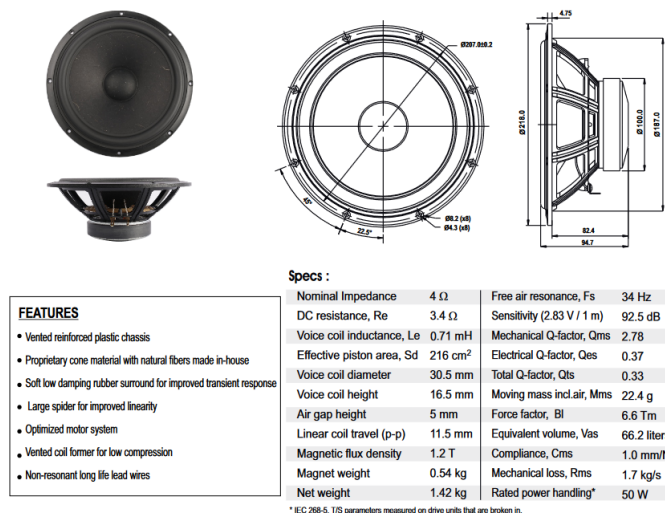


Abbildung 13: Datenblatt des Lautsprechers



### Simulation mit SubSim:

Um das Verhalten des Lautsprechers zu simulieren, kann ein Subwoofer Simulationsprogramm wie Subsim verwendet werden.

Required Data		Extra Data	
Revc (ohms)	3,4	Qts	0,33
Levc (mH)	0,71	Qes	0,37
Fs (Hz)	34	Rms (s.N/kg)	
Vas (l)	66,2	Mms (kg)	0,0224
Qms	2,78	Cms (m/N)	0,001
Sd (sq.m)	0,0216		
Bl (T.m)	6,6		
Try to Complete			

Abbildung 14: Simulationsparameter

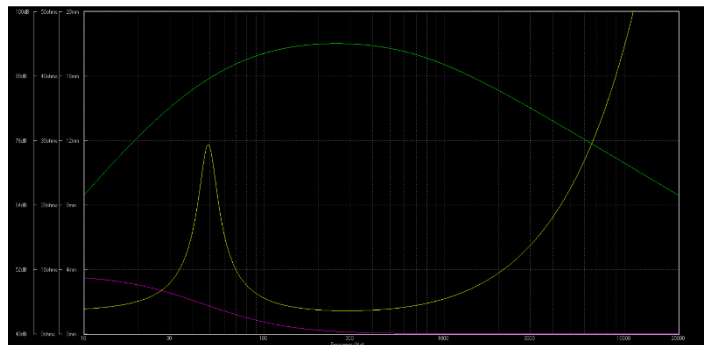


Abbildung 15: Simulationsergebnis

In Grün erkennt man das Frequenzverhalten des Lautsprechers im Leslie inklusive Resonanzkörper. In Gelb kann man die Impedanz des Lautsprechers erkennen. Man sieht, dass der Widerstand bei der Resonanzfrequenz ein peak aufzeigt und ab ~2000Hz stetig steigt. In Rosa kann man die Phase der Impedanz erkennen.

Zusammengefasst kann man feststellen, dass der gewählte Lautsprecher für Musik geeignet ist, denn er dämpft nur im sehr tiefen Bass Bereich und im Hohen Frequenzbereich

### Verstärker:

Um den Lautsprecher auch aktiv betrieben zu können muss ein Verstärker eingebaut werden. Hier lohnt sich ein Klasse AB-Verstärker, denn er bietet einen Kompromiss zwischen Klangreinheit, Effizienz und Komplexität. Der Verstärker besteht aus mehreren Stufen:

**Aux / Gitarrensinal** — Vorverstärkung — Klasse AB Endstufe — Lautsprecher

Abbildung 16: Blockschaltbild des Verstärkers

## Vorverstärkung

Das Signal (Amplitude: Gitarrensinal <50mV, Aux (~500mV-1V)) wird vorverstärkt damit die Lautstärke eingestellt werden kann. Dies wird mit einer einfachen OP-Schaltung (Nicht invertierender Verstärker) bewirkt. Um von 0.5V auf maximal 10V zu kommen, benötigen wir eine Verstärkung von 20. R6 wäre als Potentiometer verbaut

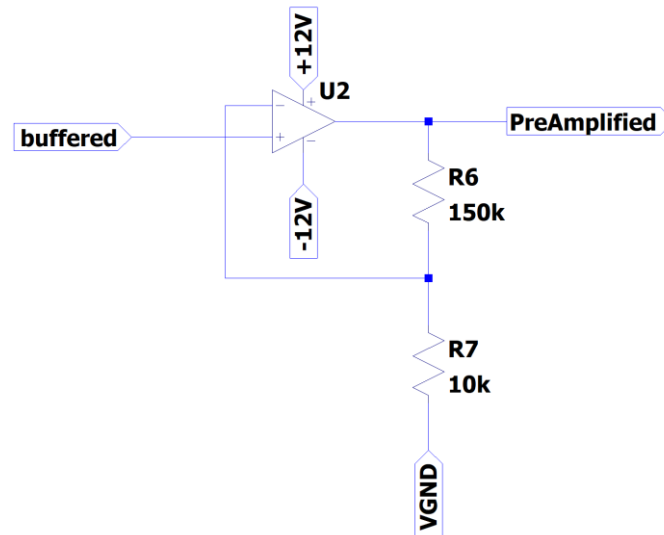


Abbildung 17: Schaltung des Vorverstärkers

## Endstufe

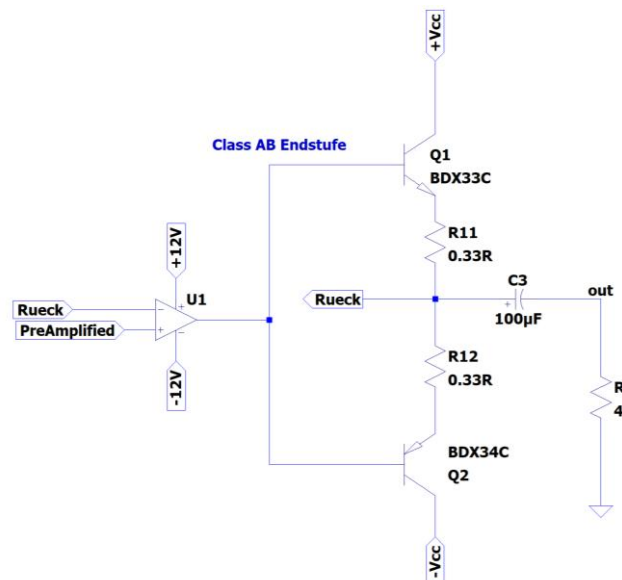


Abbildung 18: Klasse AB-Endstufe mit OP

Die Endstufe besteht aus einem Operationsverstärker und einer „Push-Pull“ Transistor Konfiguration. Der OP regelt seinen Ausgang so, dass am Ausgang ein fast unverzerrtes Signal, erzeugt wird. Der NPN-Transistor verstärkt die positive halbwelle, der PNP die negative. R11 und R12 sollen den Strom bei hohen Temperaturen stabilisieren. Optional könnte man noch Dioden ober und unter dem Ausgang des OPs dazuschalten, jedoch hat dies in unserem Fall zu Übersteuerung und unnötigen Problemen geführt. Außerdem können Basiswiderstände zur Verbesserung des

Arbeitspunktes dazugeschaltet werden. Diese haben ebenfalls zu unnötigen Problemen geführt, deshalb haben wir uns für dieses Design entschieden. Das Signal wird dann über einen Kondensator, der den DC-Anteil im Signal filtert am 4 Ohm Lautsprecher ausgegeben. Der Nachteil des AB-Verstärkers ist der hohe Ruhestrom. Er ist zwar kleiner als beim A-Verstärker jedoch liegt er beim Verbauten Transistor bei  $I_{ruhe} \approx 10mA$ .

#### Spannungsversorgung:

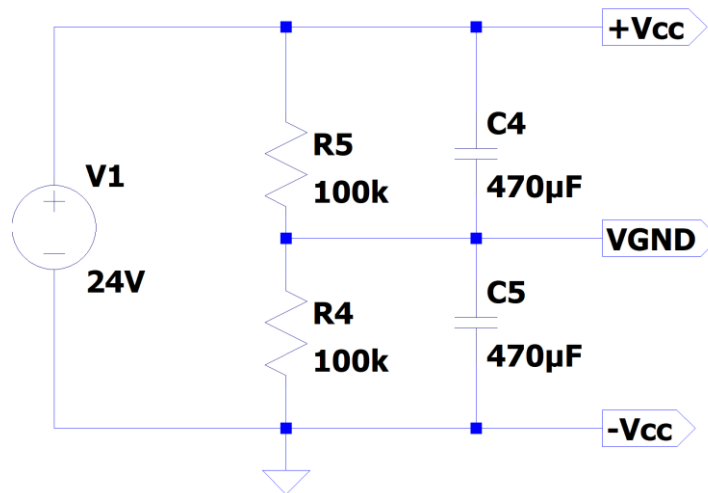


Abbildung 19: Schaltung der Spannungsversorgung

Für den OP und die Endstufe werden minus Spannungen benötigt. Diese erzeugen wir mit einem einfachen Spannungsteiler. Dieser Ansatz ist nicht ideal, für unsere Anwendung jedoch ausreichend.

Wenn man den Punkt zwischen den Widerständen als Ground für die Schaltung definiert, ist der wirkliche Ground als -Vcc anzusehen.

Die Kondensatoren dienen als einfache Glättungsfilter.

#### Simulation:

Eingangssignal: 20mV (Typisch für manche Tonabnehmer von E-Gitarren), 20kHz

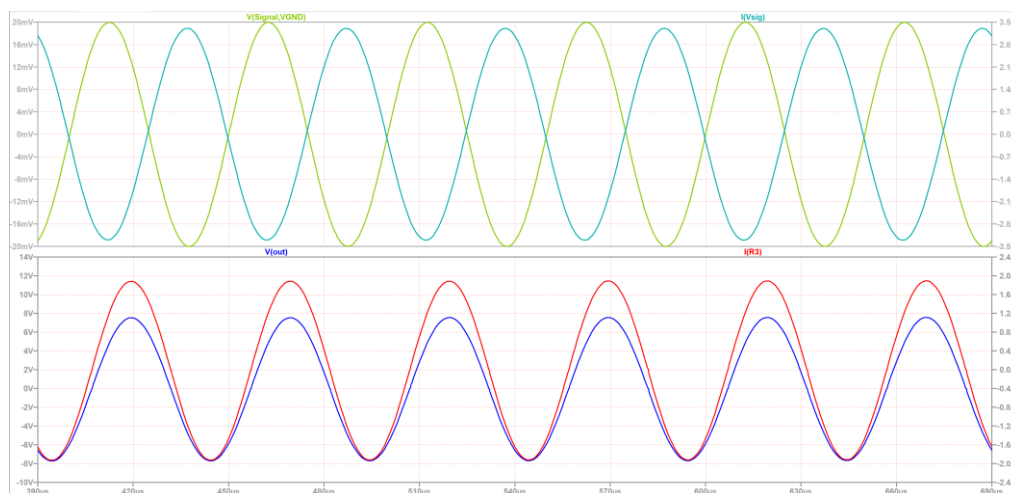


Abbildung 13: LTSpice Simulation des Verstärkers

Obwohl eine leichte Phasenverschiebung zu erkennen ist, sieht man, dass die Spannung und der Strom des Signals verstärkt wurden.

## PCB:

Bei Audioanwendungen ist es ratsam, eine Platine zu erstellen:

D1, D2, R3, R4 und C4 sind optional zu bestücken, jedoch müssen die Dioden D1 und D2 mit einem Draht überbrückt werden. C4 wäre dafür da einen Tiefpass zu realisieren, falls hochfrequente Störungen am Lautsprecher stören würden.

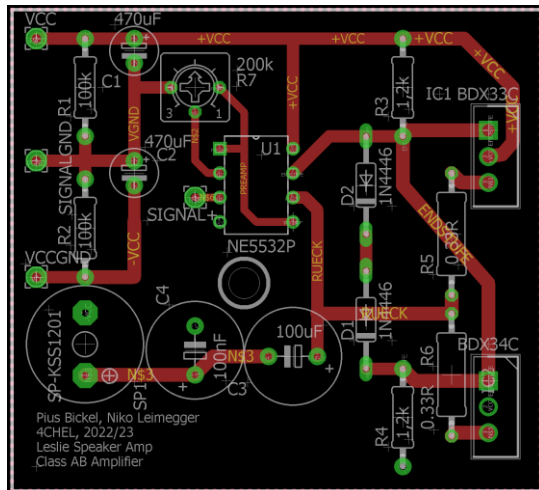


Abbildung 14: Layout der Platine

Im Nachhinein sind folgende Punkte zu verbessern:

- Widerstände R5 und R6 weiter weg von den größten Wärmeerzeugern (BDX33C, BDX34C)
- Größeres Package für R7
- Größeres Package für C1 und C2



Abbildung 20: eingebaute und bestückte Platine

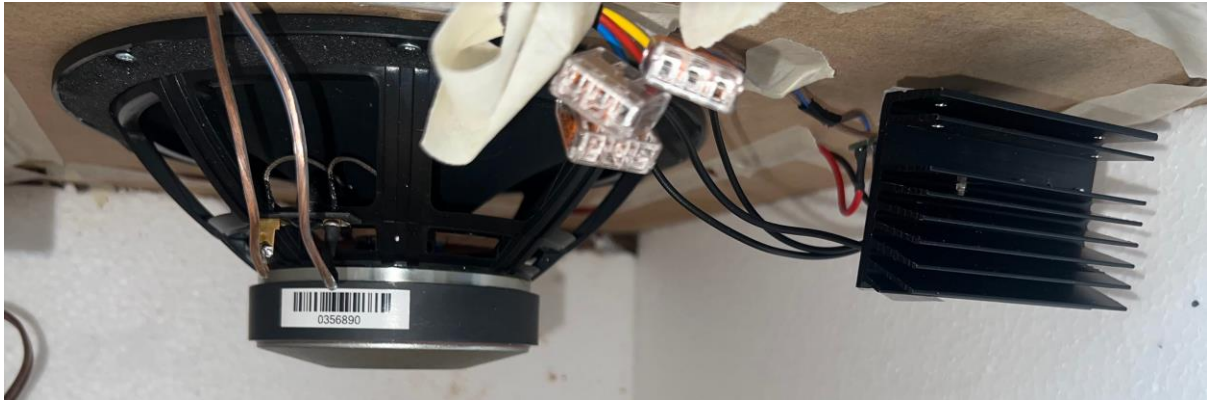


Abbildung 21: Aufbau der unteren Kammer

Footswitch:

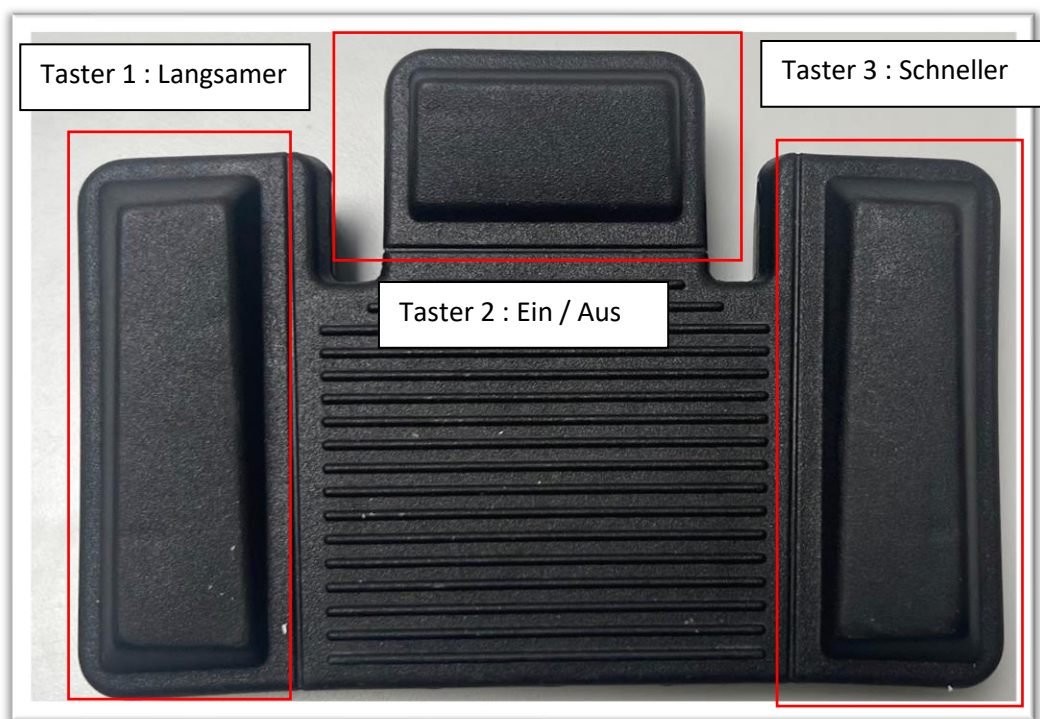


Abbildung 8 Footswitch mit Taster Beschriftung

Um den Motor auch während dem Gitarre spielen bedienbar zu gestalten, kann ein Fußschalter angebracht werden.

## Schaltung:

Um die einzelnen Switches einfach mit dem Arduino ansprechen zu können, war es hilfreich eine neue Platine zu designen, als den internen Bus abzufragen. Die Schaltung leitet lediglich einen Pin auf alle anderen und somit kann man abfragen welcher Pin z.B. an Vcc anliegt.

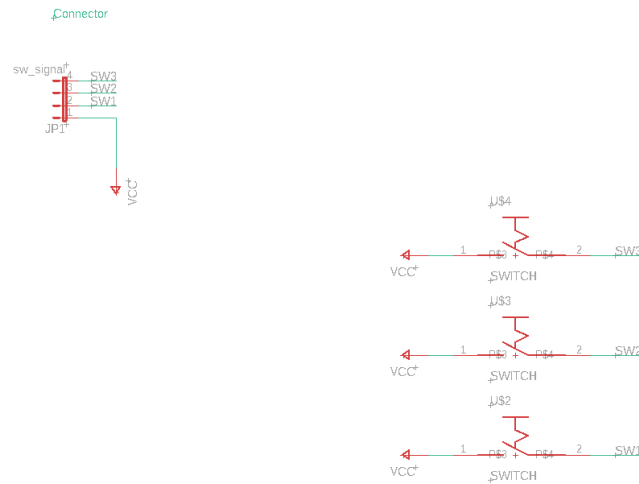


Abbildung 9 Schaltplan der Verbindungsplatine im Footswitch

## PCB:

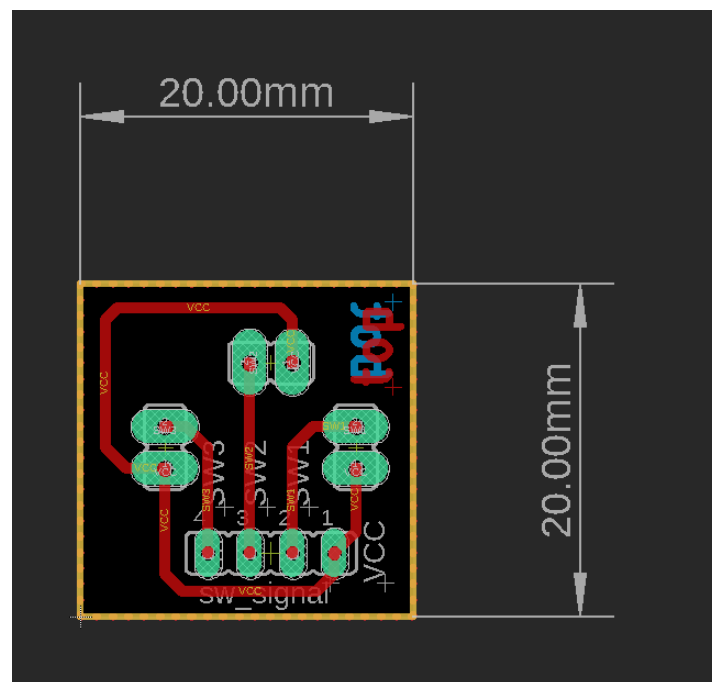


Abbildung 10 Boardplan der Verbindungsplatine

## Mechanik

### Gehäuse

Anforderungen: 3 Kammern, Stabil, akustisch fördernde Eigenschaften

### Entwicklung

Das Gehäuse besteht aus 3 Kammern, in der obersten Kammer befindet sich die Motorsteuerung, im mittleren der aus Styropor hergestellte „Cake“, und im untersten Abteil der Verstärker und Lautsprecher.

### Designs

Das Ganze Gehäuse hat Außenmaße von 800mm \* 500mm \* 500mm, und eine Wandstärke von 12mm.

Einzelne Bauteile:

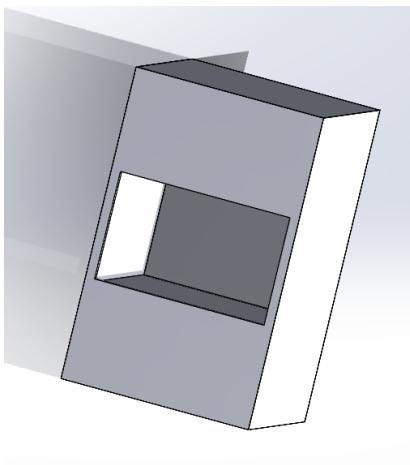


Abbildung 24: Gesamtkonstrukt

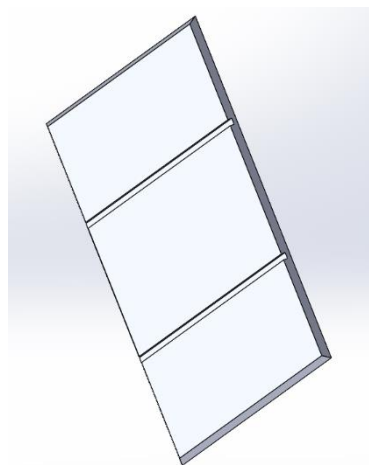


Abbildung 23: Seitenwände mit Fräsung

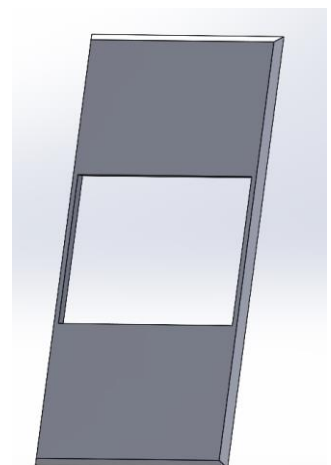


Abbildung 22: Frontplatte mit Loch für Schall

Die Ober- und Unterplatte sowie die Zwischenböden wurden nach dem Zusammenbau korrekt zugeschnitten.

### Verbindungsstück

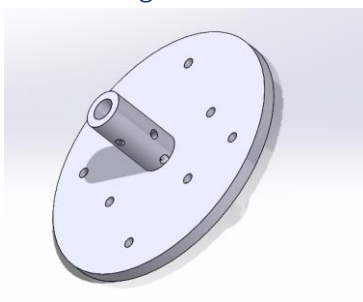


Abbildung 25: Verbindungsteil zwischen Motor und Klangführung

Nachdem Leim allein keine ausreichend starke Verbindung zwischen Metall (Rotor) und Styropor herstellen kann, wird per 3D Druck ein Adapter gefertigt, der mittels Schrauben und Leim festgemacht ist, und am Rotor ausschließlich durch Reibung festgehalten wird.

Bei einer zweiten Version wäre es sinnvoll, den Halter vom Rotor zu stützen damit die Fliehkräfte weniger Auswirkung auf die Drehbewegung haben.



### Fertiges Gehäuse



1. Das Gehäuse ohne Klangführung und Frontplatte
2. Gehäuse inklusive Klangführung und provisorischer Styropordichtung
3. Fertiges Gehäuse

Die Tonkammer für den Lautsprecher wurde mit Styropor abgedichtet, um einen besseren Klang zu realisieren.

### Software

Anforderungen: Motor PWM steuern, Taster abfragen.

Eigentlich war für dieses Projekt eine MEGACARD vorhergesehen, durch Schwierigkeiten in der permanenten Beschaffung einer MEGACARD hat sich ein Arduino Nano als ökonomischer herauskristallisiert.

Software geschrieben in der Arduino IDE

### Struktogramme:

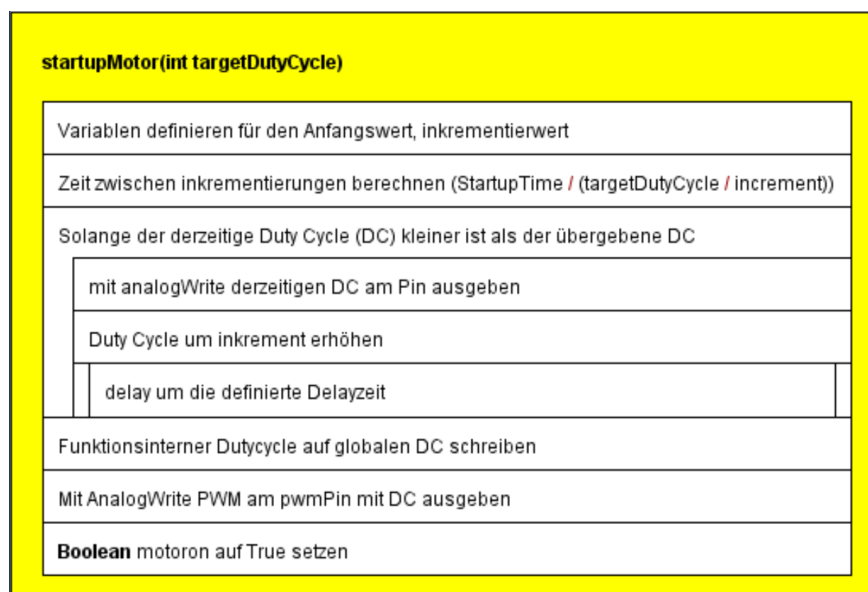


Abbildung 12 Struktogramm der Motor-Startup-Routine

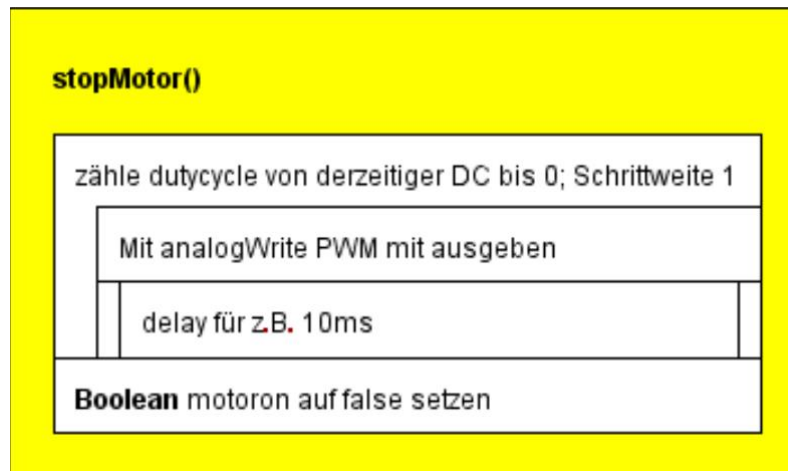


Abbildung 13 Struktogramm der Motor-Stopp-Routine

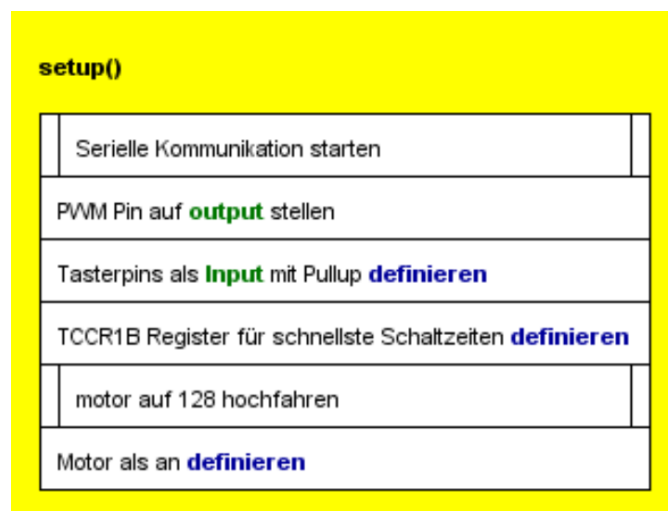


Abbildung 26: Struktogramm für die Setup Funktion

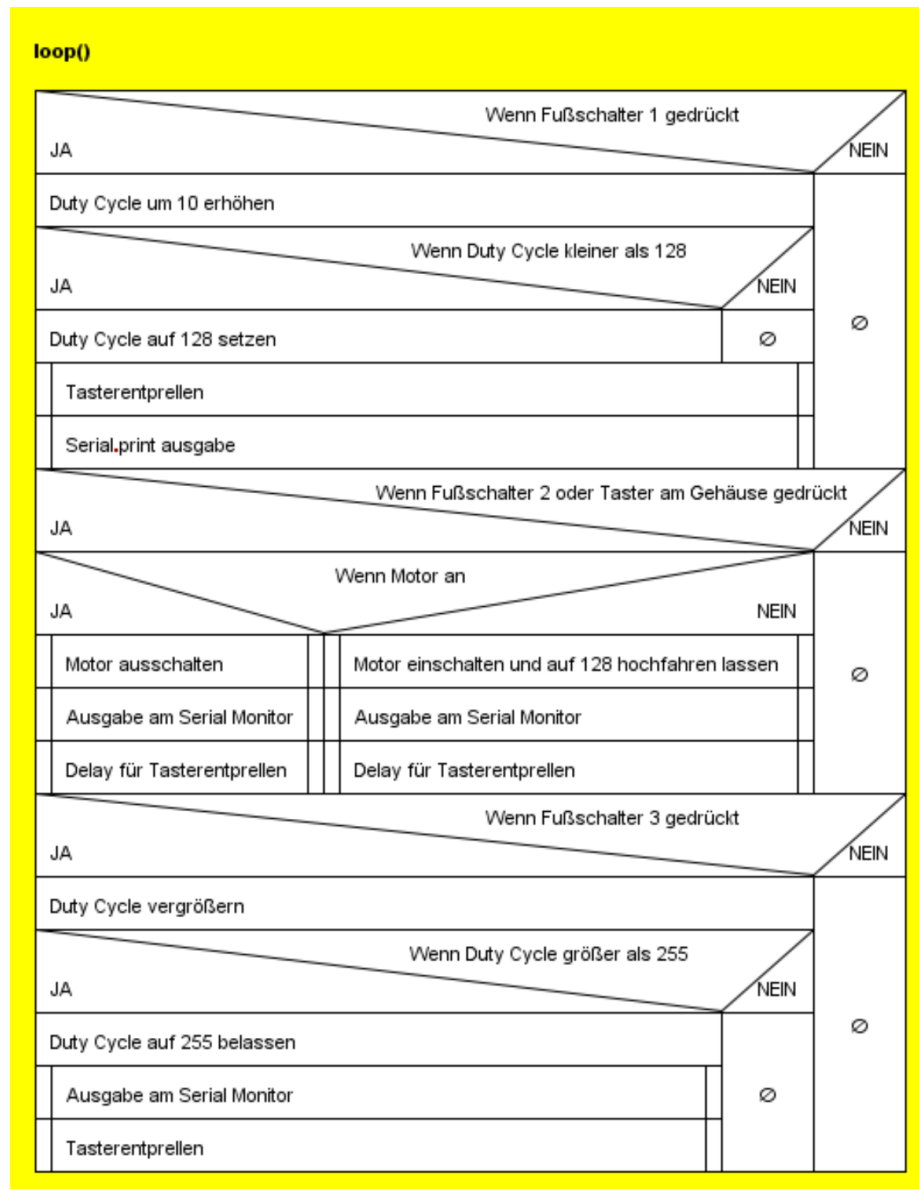


Abbildung 27: Struktogramm des Main programms

Gesamter Code:

```
// PWM1 fuer Leslie Speaker
```

```

const int pwmPin = 5; // Define the PWM pin
const int startupTime = 5000; // Zeit für das Hochfahren in Millisekunden
const int Foot1 = 5; // Fußschalter Taster 1
const int Foot2 = 2; // Fußschalter Taster 2
const int Foot3 = 3; // Fußschalter Taster 3
const int Switch = 7; // Schalter um den Motor ein und aus zu stellen
volatile bool motoron; // Boolean variable um Status des Motors festzustellen
volatile int duty = 128; // Standard DutyCycle (50%)

```

```
void setup() {
    Serial.begin(9600); // Serielle Kommunikation starten um Tasterabfragen zu
    // überprüfen
    pinMode(pwmPin, OUTPUT); // PWM Pin als Output

    pinMode(Foot1, INPUT_PULLUP); // Pins für den Fußschalter auf Input
    //inklusive Pullup Widerstand
    pinMode(Foot2, INPUT_PULLUP);
    pinMode(Foot3, INPUT_PULLUP);
    pinMode(Switch, INPUT_PULLUP);

    TCCR1B = TCCR1B & B11111000 | B00000001; // Register für schnellere
    //Schaltgeschwindigkeiten.

    startupMotor(duty); // Motor auf Standard DutyCycle (DC) hochfahren
    motoron = true;      // Motor als an definieren
}

void loop() {

    // Wenn Taster 1 gedrückt wurde, DC um 10 verkleinern außer der DC ist schon
    //bei 50% + Ausgabe an den Serial Monitor
    if (digitalRead(Foot1) == LOW){ // Tasterabfrage
        duty = duty - 10; // DC verkleinern
        if (duty <= 128) {duty = 128;}; // Mindest DC: 50%
        // Ausgabe am Serial Monitor:
        Serial.print("Langsamer (duty: ");
        Serial.print(duty);
        Serial.println(")");
        delay(150); // Tasterprellen vermeiden
    }

    // Wenn Taster am Gehäuse oder am Fußschalter in der Mitte gedrückt: Motor an
    //bzw. Ausschalten
    if (digitalRead(Foot2) == LOW || digitalRead(Switch) == LOW){ //
    //Tasterabfrage
        if(motoron){ // Motor ausschalten + Ausgabe Serial Monitor
            motoron = false;
            Serial.println("Aus");
            stopMotor();
            delay(150);
        }else{ // Motor einschalten wenn er aus war
            motoron = true;
            Serial.println("Ein");
            startupMotor(200);
            delay(150);
        }
    }
}
```

```
// Wenn Taster 3 am Fußschalter gedrückt, DC um 10 erhöhen
if (digitalRead(Foot3) == LOW){ // Tasterabfrage
    duty = duty + 10; // DC erhöhen
    if(duty >= 255){duty = 255;} // Höchster DC: 255 = 100%

    // Ausgabe
    Serial.print("Schneller (dutyCycle: ");
    Serial.print(duty);
    Serial.println(")");
    delay(150);
}

// PWM ausgeben
analogWrite(pwmPin, duty);
}

void startupMotor(int targetDutyCycle) {
    int currentDutyCycle = 0; // Aktueller Duty Cycle
    int increment = 5; // Inkrement zur Erhöhung des Duty Cycles
    int incrementDelay = startupTime / (targetDutyCycle / increment); //
    Verzögerung zwischen den Inkrementen

    // Hochfahren des Motors
    while (currentDutyCycle <= targetDutyCycle) {
        analogWrite(pwmPin, currentDutyCycle); // Duty Cycle auf den Motor-Pin
        schreiben
        currentDutyCycle += increment; // Duty Cycle erhöhen
        delay(incrementDelay); // Verzögerung zwischen den Inkrementen
    }
    duty = targetDutyCycle; // derzeitiger DC umschreiben auf normalen DC
    analogWrite(pwmPin, duty);
    motoron = true;
}

void stopMotor(){
    // Motor allmählich stoppen
    for (int speed = duty; speed >= 0; speed--) {
        analogWrite(pwmPin, speed); // Geschwindigkeit setzen
        delay(10); // Kleine Verzögerung, um die Veränderung wahrnehmbar zu
        machen
        duty = speed;
    }
    motoron = false;
}
```

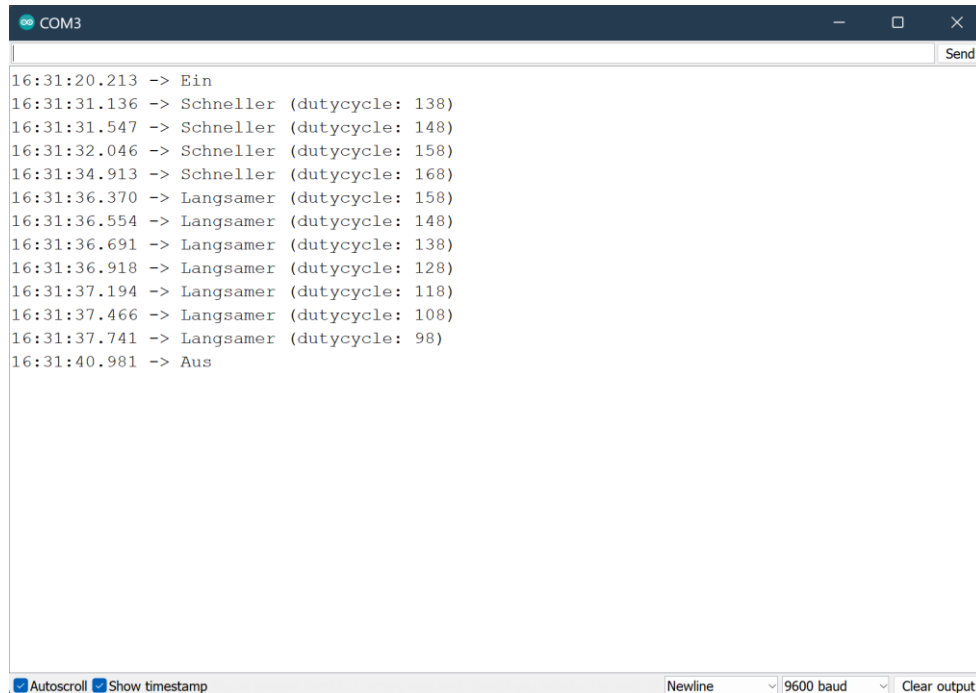
## Funktionstest

Ausgabe am Arduino:

Langsamer : Taster 1 links

Ein/Aus : Taster 2 in der Mitte

Schneller: Taster 3 rechts



```
COM3
16:31:20.213 -> Ein
16:31:31.136 -> Schneller (duty cycle: 138)
16:31:31.547 -> Schneller (duty cycle: 148)
16:31:32.046 -> Schneller (duty cycle: 158)
16:31:34.913 -> Schneller (duty cycle: 168)
16:31:36.370 -> Langsamer (duty cycle: 158)
16:31:36.554 -> Langsamer (duty cycle: 148)
16:31:36.691 -> Langsamer (duty cycle: 138)
16:31:36.918 -> Langsamer (duty cycle: 128)
16:31:37.194 -> Langsamer (duty cycle: 118)
16:31:37.466 -> Langsamer (duty cycle: 108)
16:31:37.741 -> Langsamer (duty cycle: 98)
16:31:40.981 -> Aus
```

Autoscroll Show timestamp Newline 9600 baud Clear output

Abbildung 28: UART-Ausgabe des Arduino während der Geschwindigkeitsanpassung

## Konklusion:

Es wurde ein Leslie Speaker realisiert, der Passiv sowie Aktiv betrieben werden kann. Der Verstärker im Lautsprecher hat eine maximale Ausgangsleistung von 20W. Der Motor des Lautsprechers kann mit einem externen Fußschalter gesteuert werden. Im Projekt wurden Elektrotechnische-, Softwaretechnische- und Mechanische Elemente verbaut und entwickelt.

Erweiterungsmöglichkeiten wären zum Beispiel ein Bluetooth Modul, um Wireless Musik abspielen zu können, Verzierung des Gehäuses, Rollen auf der Unterseite des Gehäuses, neuer Adapter zwischen Rotor und Klangführung für stabilere Anwendung.