



University of Applied Sciences

HOCHSCHULE
EMDEN-LEER

Projektarbeit

RotaCon - Dokumentation

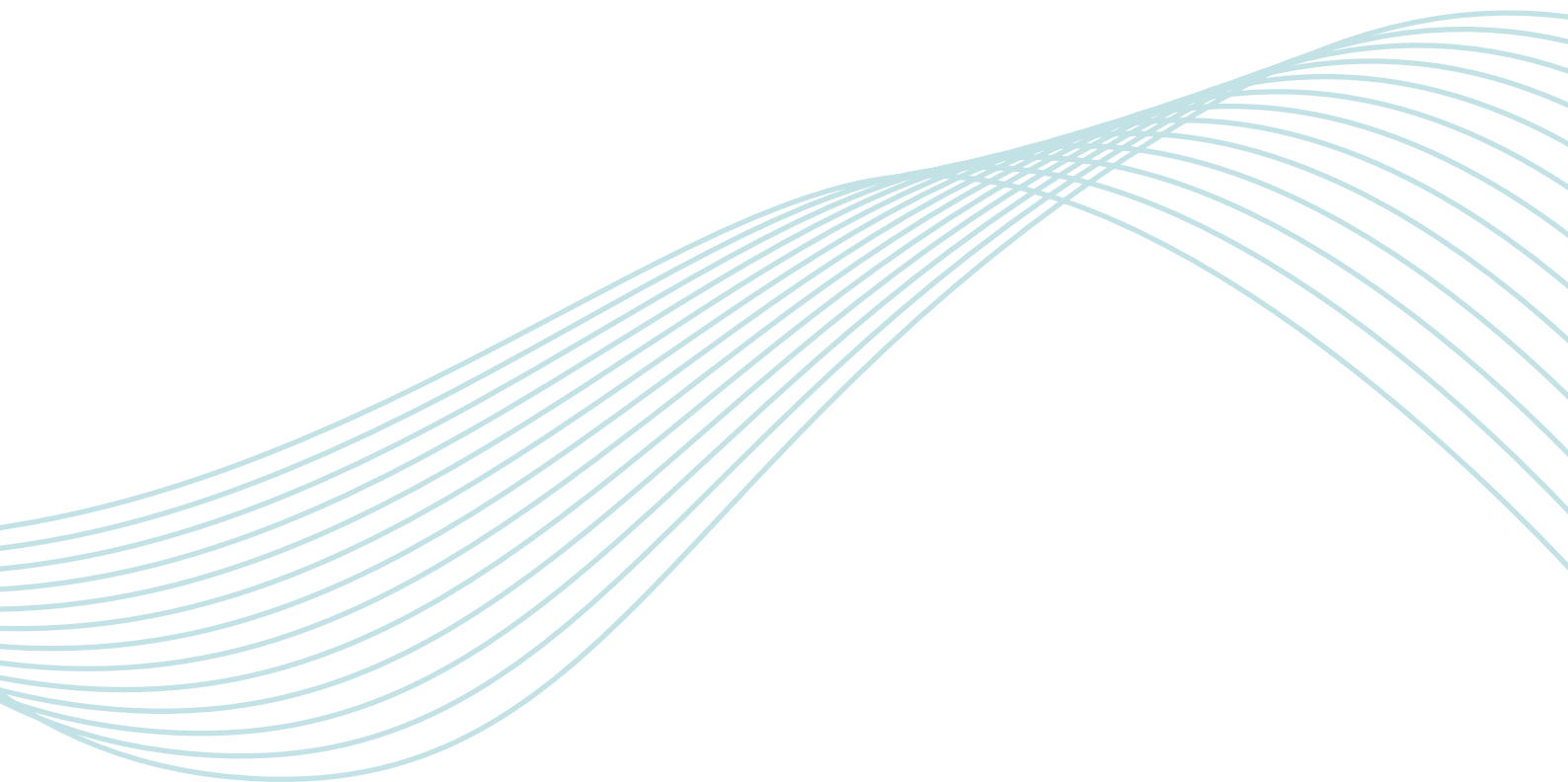
vorgelegt von:

- Tino Liebenow - Matrikelnummer 7011830

betreut durch

Dipl.-Ing. (FH) Jörg Strick

Abgabedatum: 30.01.2020



Inhaltsverzeichnis

1	Prolog	4
2	Konzeptentwurf	5
2.1	Theorie Hardware	5
2.2	Theorie Software	6
2.3	Theorie Mechanik und Design	6
3	Prototyp-Entwicklung	7
3.1	Notwendige Software	7
3.2	Verwendete Hardware	8
3.3	Realisierung der Schaltung	10
3.3.1	Motorsteuerung	10
3.3.2	IR-Empfänger	11
3.3.3	Display	12
3.3.4	Anschlagerkennung durch Strommessung	13
3.3.5	Anschlagerkennung durch Endschalter	15
3.3.6	Pinbelegungen	15
3.3.7	Gesamtschaltung der Display-Platine	16
3.3.8	Gesamtschaltung der Motor-Platine	17
3.4	Software	18
3.5	Das Gehäuse	19
3.5.1	Motorgehäuse	19
3.5.2	Displaygehäuse	20
3.5.3	Gehäuseständer	21
4	Ergebnis	22
4.1	Inbetriebnahme	22
4.2	Problemdiskussion	22
4.3	Zusammenfassung	23
4.4	Ausblick	23
5	Anhänge	24
5.1	Tabelle Teileübersicht	24
5.2	Datenblätter	24
5.3	Layouts	25
5.4	Gehäusemaße	27
5.5	Quellcode	31

Abbildungsverzeichnis

3.1	Aufbau eines unipolaren Schrittmotors.	9
3.2	Ansteuerung des Motors durch Arduino und Darlington-Transistortreiber ULN2003.	10
3.3	Infrarot-Empfänger mit Pull-Up-Widerstand und Filterkondensator.	11
3.4	D-Sub-Pinbelegung für Display und IR-Receiver(gestrichelte Linie).	12
3.5	Einsatz eines nicht-invertierenden Verstärkers zur Strommessung	13
3.6	Finale Schaltung der Display-Platine.	16
3.7	Finale Schaltung der Motor-Platine.	17
3.8	Skizze Des Motorgehäuses	19
3.9	Skizze Des Displaygehäuses	20
3.10	Gehäuseständer des Prototyps	21
5.1	Display-Platine	25
5.2	Motor-Platine	26

Tabellenverzeichnis

3.1	Pin-Belegungen	15
-----	--------------------------	----

Abkürzungsverzeichnis

A/D Analog / Digital

CAD computer-aided design

D-Sub D-Subminiatur, eine Bauform für ein Steckersystem

EEPROM electrically erasable programmable read-only memory

EQ Equalizer

HSEL Hochschule Emden Leer

IDE Integrated Development Environment

I/O Input / Output

IR Infrarot

LCD liquid crystal display

TP Tiefpass

USB Universal Serial Bus

Kapitel 1

Prolog

Hier wird Tino am Ende der Arbeit eine tolle Einleitung hinzaubern, die die Ausgangslage zu derzeitigen Aktivlautsprechern beschreibt, ähnlich der Idee. Außerdem schreibe ich in männlicher Form etc...
Außerdem reden wir immer vom AKTIV-Lautsprecher...

Überblick

Überblick über den Aufbau der Dokumentation.

Diese Dokumentation ist in vier Hauptteile gegliedert. Im Prolog wird die Ausgangssituation geschildert, sowie die Kerngedanken zum RotaCon.

Ziel

was möchte Tino heute eigentlich machen?

Ziel ist es ein Gerät zur Fernbedienbarkeit von Potentiometern, speziell an Lautsprechern, zu entwickeln. Dabei sollen weder Veränderungen am Lautsprechergehäuse, oder an der Technik vorgenommen werden. Zusätzlich ist eine Diskretisierung der Pegelwerte und deren Visualisierung via LCD-Display angedacht, sodass der Benutzer sichtbare Werte zur Orientierung der aktuellen Einstellung bekommt.

Stand der Technik

gibts sowas schon?

Fernsteuerungen für Pegeleinstellungen sind nicht neu auf dem Markt. Dabei sind zwei Varianten verbreitet, zum Einen besteht die Möglichkeit, den Potentiometer auf der Platine des Lautsprechers auszutauschen, zum Anderen die Integration eines Controllers in den Signalweg. Die erste Variante setzt somit die Öffnung des Lautsprechergehäuses sowie Manipulation der Elektronik voraus. Dies ist nicht nur für den durchschnittlichen Lautsprecherbesitzer ein schwieriger Eingriff, sondern hat ebenfalls Einfluss auf diverse Garantieansprüche.

Die andere Variante beinhaltet keine Veränderungen am Lautsprecher selbst, regelt jedoch nur die Signalpegel zum Lautsprecher hin und nicht den integrierten Verstärker. Somit muss für diese Version der Verstärker stets auf Maximum geregelt sein, dies hat je nach Verstärker Einflüsse auf Soundqualität und Stromverbrauch.

Kapitel 2

Konzeptentwurf

Das folgende Kapitel beinhaltet die theoretischen Bestandteile zur Umsetzung der Idee. Diese bilden in ihrer Gesamtheit das Konzept, nachdem der Prototyp entwickelt wird.

2.1 Theorie Hardware

Lautstärkeregler sind in den meisten Fällen Potentiometer oder Inkrementalgeber. [Wikipedia[1]] Diese besitzen in der Regel eine 6mm oder 6.35mm Achse, in D-Form oder geriffelt. Um diese Achse rotieren zu können muss also ein Motor mit einer passenden Kupplung angebracht werden. Für präzise Kontrolle und ausreichendes Drehmoment ist ein Schrittmotor geeignet. Diesen gibt es in verschiedenen Ausfertigungen: Unipolar und Bipolar. Unipolare Schrittmotoren haben vier Phasen und werden nur in einer Richtung von Strom durchflossen. Bipolare polen ihre Magnetfelder durch Umkehrung der Stromrichtung um, sie haben zwei Phasen, erreichen ein höheres Drehmoment und sind durch ihre Funktionsweise etwas aufwendiger anzusteuern. Für eine Schrittmotorsteuerung sind drei Funktionseinheiten notwendig. Diese bestehen aus Mikrocontroller, Steuerschaltung und Treiberstufen. Für dieses Projekt ist ein unipolarer Schrittmotor ausreichend, da keine hohen Drehmomente erreicht werden müssen. Dies hat den Vorteil, dass die Steuerschaltung rein auf Softwarebasis gelegt werden kann. Als Treiber soll eine Darlington-Schaltung für den Motor dienen. Die softwaretechnische Besteuerung ist durch ein Arduino mit integriertem Mikrocontroller angestrebt, dessen Spannungsversorgung von 5V idealer Weise auch für die anderen Komponenten ausreichen soll, sodass ein Betrieb des Gerätes durch USB-Spannungsversorgung ermöglicht wird. Da analoge Drehregler durch eine Minimal- und Maximalstellung begrenzt sind, ist auch der Antrieb des Motors in diesen Bereichen zu stoppen. In erster Linie sollen Endschalter den Potentiometeranschlag erkennen und software-gesteuert die Rotation einstellen. Redundant dazu wird der Stromverbrauch des Schrittmotors durch die Kombination eines nicht-invertierenden Verstärkers und dem Analog-Digital-Wandler vom Mikrocontroller überwacht. Für die Bedienung des Gerätes soll ein Infrarotempfänger mit passender Fernbedienung im Frequenzbereich verwendet werden. Da sich Lautstärkeregler meist nicht sichtbar am Lautsprecher befinden, ein IR-Empfänger jedoch eine quasioptische Verbindung voraussetzt, bieten sich zwei getrennte Gehäuseteile an, die durch ein Kabel verbunden sind. In dem sichtbaren Gehäuse befindet sich zusätzlich das Display.

2.2 Theorie Software

Softwaretechnisch sind drei Bereiche abzudecken, genannt seien die Programmierung des Mikrocontrollers an sich, das Entwerfen eines Platinenlayouts sowie der Entwurf der Gehäuse für den 3D-Druck. Bei der Programmierung des Mikrocontrollers soll beachtet werden, dass die aktuellen Werte der Motorstellung und der dazugehörigen Lautstärkeanzeige gespeichert werden, damit bei jeder Inbetriebnahme die jeweils vorherigen Werte beibehalten bleiben. Die einzige Schnittstelle zum Benutzer ist das Display, daher müssen Fehler, Warnungen und wichtige Informationen auf dem Display abgebildet werden. Alle variablen Funktionen des Gerätes sollen über die Fernbedienung steuerbar sein, also die Steuerung der Lautstärke, aber auch das Aktivieren, bzw. Deaktivieren der Displaybeleuchtung.

2.3 Theorie Mechanik und Design

Die Übertragung des Drehmoments vom Motor an den Drehregler des Lautsprechers kann nur stattfinden, wenn der Motor starr gelagert ist. Daher ist das Gehäuse mit Motor und Motorsteuerung zu fixieren. Weiterhin sind Lautstärkeregler an verschiedenen Lautsprechern auch an verschiedenen Positionen, sodass dieses Gehäuse zusätzlich in seiner Position variabel positionierbar sein muss. Einige Lautsprecher mit stärkeren Verstärkern haben zusätzlich Kühlkörper an der Lautsprecherrückseite, aus diesem Grund und der Tatsache, dass eventuell weitere Regler (Phase, TP, EQ) vorhanden sind, ist ein gewisser Abstand zwischen den Gehäusen notwendig. Auch die Positionen der Endanschläge sind nicht überall gleich, das bedeutet, dass auch die Endschalter in ihrer Position variierbar sein müssen.

Kapitel 3

Prototyp-Entwicklung

Folgendes Kapitel beinhaltet die Beschreibung aller im RotaCon verwendeten Bestandteile, sowie der Software, die zum Entwickeln benutzt wurden. Es wird auf die Programmierung des Mikrocontrollers eingegangen, die elektronische Schaltung und der Entwurf der Gehäuse wird erläutert.

3.1 Notwendige Software

Arduino IDE

Diese Entwicklungsumgebung (IDE¹) stellt den Softwareteil der Arduino-Plattform dar und basiert auf Processing, einer stark typisierten Programmiersprache mit integrierter Entwicklungsumgebung. Die Programmiersprache ähnelt C++ und ist darauf ausgelegt ohne tiefgründige Programmierkenntnisse Programme schreiben zu können. Programme dieser IDE werden "Sketch" genannt. Dieser Sketch wird in der IDE kompiliert und kann via USB direkt auf den Speicher des Arduino-Boards geladen werden. Ein in die IDE integrierter serieller Monitor kann die jeweiligen Zustände der Ein- und Ausgänge überwachen und ist ein Hilfsmittel zur Fehlersuche in der Entwicklungsphase.

FreeCAD

FreeCAD ist eine parametrische Modellierungsanwendung. 3D-Objekte können mit hoher Präzision modelliert und angeordnet werden. Vorteil gegenüber einer Modellierungsanwendung wie Cinema 4D besteht in der Möglichkeit der Herstellung von Unterlagen zum designten Modell. So können beispielsweise technische Zeichnungen und Montagepläne erstellt werden. Die Anwendung ist ein Open-Source Projekt, daher wird es gemeinschaftlich weiterentwickelt und wächst in seinem Funktionsumfang. Verwendete Version: FreeCAD 0.18

Cura

Cura ist eine Slicing Software von der Firma Ultimaker. Dieses Programm ist für die Umwandlung von 3D-Objektmodellen in spezifische Druckanweisungen verantwortlich. Dazu wird ein 3D-Modell zu einem Stapel von flachen Schichten umgerechnet beziehungsweise "zerschnitten". Zusätzliche Einstellungen für den 3D-Drucker werden vorgegeben, sodass Druckablauf, Temperaturen von Druckbett und Extruder, Filament-Art, Vortriebsgeschwindigkeiten und gegebenenfalls Stützen für den Druck von Überhängen festgelegt sind.

¹IDE steht für "Integrated Development Environment" und bedeutet Entwicklungsumgebung.

Eagle

Die Software Eagle von Autodesk ist ein Platinen-CAD Programm zum entwerfen von Schaltplänen und Platinenlayouts. Dabei sind zwei Hauptschritte der Bearbeitung auszuführen. Zum Einen die Gestaltung des Schaltplans, dazu bietet Eagle durch Einbinden von Bibliotheken annähernd alle erdenklichen Schaltsymbole und Bezeichnungen die für den Entwurf der Schaltung notwendig sind. Zusätzlich können manuell Symbole erstellt werden, falls spezielle unkonventionelle oder eigens entwickelte Elemente verwendet werden. Im zweiten Schritt wird das Layout der Platine entworfen, um die Leiterbahnen und Bohrungen passend zu platzieren/ dimensionieren.

3.2 Verwendete Hardware

Arduino Nano

Arduino ist eine Open-Source Elektronik Plattform, die aus zwei Teilen besteht. Einem Hardware- und einem Softwareteil [Brühlmann [2, S.23]]. Der Hardware-Teil ist eine Mikrocontroller-Schaltung mit serieller Schnittstelle. Diese "Mikrocontroller-Boards" gibt es in verschiedenen Ausführungen und unterscheiden sich in ihrer Größe, dem verwendeten Mikrocontroller und den somit einhergehenden Eigenschaften von Ein- und Ausgängen. Die Eingänge können Daten von Sensoren einlesen, die Ausgänge dienen zur Ansteuerung von Motoren, Dioden, Relais oder anderen elektronischen Komponenten. Die Spannungsversorgung kann über die USB-Schnittstelle oder externem Netzteil erfolgen. Der im RotaCon verwendete Arduino Nano ist eine Leiterplattenversion von geringer Baugröße und nutzt als Basis den ATmega328 [Arduino [3]]. Der Arduino Nano hat insgesamt 22 Input/Output-Pins, 8 davon können als analoge Eingänge benutzt werden. Die logische Spannung des Nanos beträgt 5V, die maximale Stromstärke pro Pin 40 mA. Mit einer Taktrate von 16 MHz und einem integrierten Speicher (EEPROM) von 1KB ist der Arduino Nano in seinen Funktionen für dieses Projekt gewählt worden.

Schrittmotor (unipolar)

Der im RotaCon verwendete unipolare Schrittmotor 28BYJ-48 hat vier Phasen. Das bedeutet vier Elektromagneten steuern die Bewegung des Dauermagnetrotors. Diese vier Elektromagneten werden durch zwei Spulen realisiert, deren Wicklungen je hälftig aufgeteilt und entgegengesetzt gewickelt sind. Sie sind statisch um den Rotor angeordnet (3.1). Die einzelnen Phasen werden nur in einer Richtung von Strom durchflossen und steuern so die Polung der Magneten beziehungsweise die Rotationsrichtung des Motors. Die Versorgungsspannung beträgt 5V und der Widerstand einer Spule ist mit $50\Omega \pm 7\%$ angegeben. Je nach Einphasensteuerung oder Zweiphasensteuerung ergeben sich daher Ströme von etwa 100mA (zweiphasig), bzw. 200mA (einphasig). Bei Schrittmotoren werden im Datenblatt Schrittwinkel und Getriebeübersetzung angegeben. Der hier verwendete Motor hat einen Schrittwinkel von $5,625^\circ$ bei 64 Schritten angegeben und ein Getriebeverhältnis von $1/64$. Daraus ergibt sich, dass eine volle Umdrehung der Motorachse 4096 Schritte benötigt (64×64). Diese Winkelangabe bezieht sich auf einen Halbschrittbetrieb, daher ist im Vollschrittbetrieb der Schrittwinkel $11,25^\circ$ und eine Achsumdrehung benötigt 2048 Schritte [vergl. Tkotz et al. [4, S.641]]. Bei einem Halbschrittverfahren werden Ein- und Zweiphasensteuerung zusammen benutzt um den Motor in 45° Schritten zu bewegen [vergl. Deltron AG [5]] Die Verwendung eines unipolaren Motors hat den Vorteil, dass auf Grund der einseitigen Stromflussrichtung, die Steuerung rein durch Software erfolgen kann und daher auf H-Brücken oder ähnliche zusätzliche Schutzmaßnahmen in diesem Projekt verzichtet werden kann.

$$I = \frac{U}{R} = \frac{5V}{50\Omega} = 100mA \quad \text{bzw.} \quad \frac{5V}{25\Omega} = 200mA \quad (3.1)$$

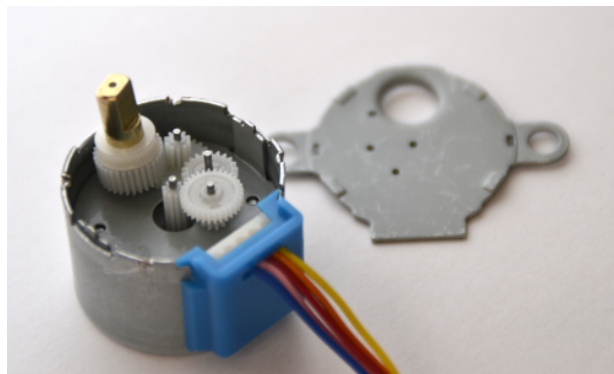
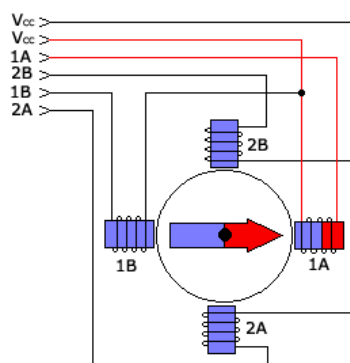


Abbildung 3.1: Aufbau eines unipolaren Schrittmotors.

3.3 Realisierung der Schaltung

3.3.1 Motorsteuerung

Eine Darlington-Schaltung ist eine Schaltung zweier Transistoren mit hoher Stromverstärkung und wird zur Schaltung beziehungsweise Steuerung größerer Lasten eingesetzt [Böhmer et al. [6, S.290]]. Dabei können die nötigen Steuerströme gering gehalten werden. Der hier verwendete ULN2003 beinhaltet mehrere Darlington-Transistoren in einem Gehäuse und verstärkt die Steuersignale des Arduinos ohne dessen Ausgänge mit zu hohen Stromstärken zu belasten. Gemäß Datenblatt ist die Kollektor-Emitter-Sättigungs-Spannung im leitenden Zustand zwischen 0,9 und 1,3 Volt bei in Kapitel 3.2 angenommenen 100mA bis 200mA. Bei der höchsten angenommenen Sättigungsspannung von 1,3 V ergibt sich ein Potentialunterschied an den jeweiligen Wicklungen von 3,7 V. Die Messung der Widerstände der einzelnen Halbspulen² des verwendeten Motors ergibt, dass jede Phase etwa 23 Ω Widerstand hat. Bei Vollschrittbetrieb des Motors, bei dem je genau eine Phase geschaltet wird, lassen sich die Ströme daher berechnen:

$$I_{Phase} = \frac{\text{Versorgungsspannung}(5V) - \text{Sättigungsspannung}}{\text{Phasenwiderstand}} \quad (3.2)$$

Stromberechnung³ je nach Sättigung:

$$I_{min} = \frac{3,7V}{23\Omega} = 160mA \quad I_{max} = \frac{4,1V}{23\Omega} = 178mA \quad (3.3)$$

Da die Spannungsversorgung des Motors je zwischen den Halbspulen anliegt (vergl.3.1) und sie somit parallel geschaltet sind, betragen Strom und Spannung bei einer Zweiphasensteuerung pro Phase nur die Hälfte. Der Gesamtstrom am Treiber jedoch entspricht dem der Einphasensteuerung.

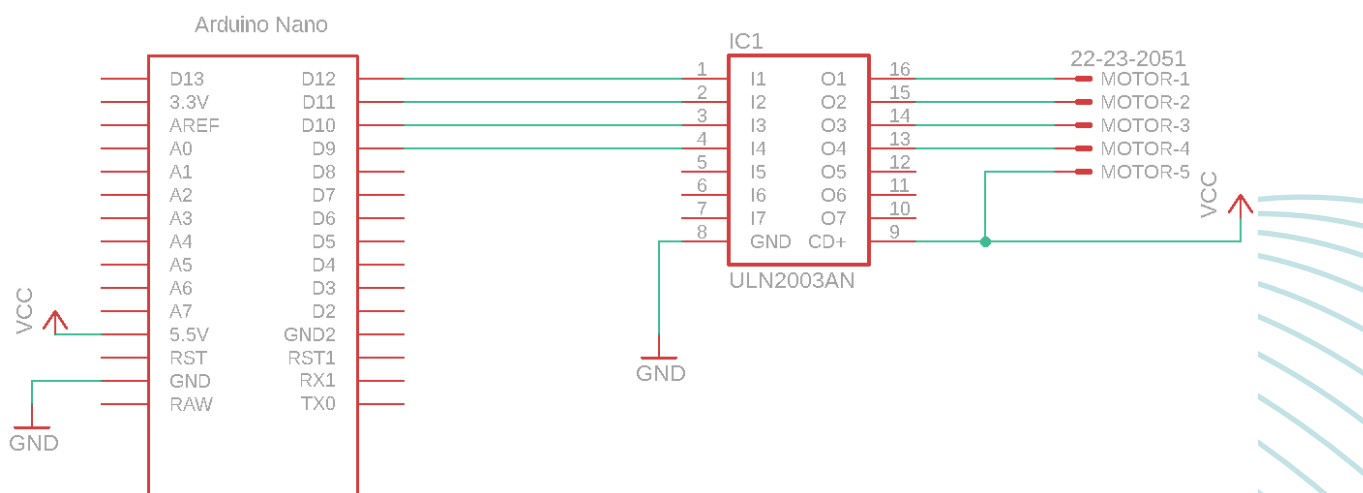


Abbildung 3.2: Ansteuerung des Motors durch Arduino und Darlington-Transistortreiber ULN2003.

²Messung durch Multimeter: testo67-2

³Berechnung nur gültig für Vollschrittbetrieb des Motors und Schaltung je nur einer Phase

3.3.2 IR-Empfänger

Das verwendete Infrarotempfänger-Modul TSOP4838 beinhaltet einen Photosensor, Demodulator und Vorverstärker. Signalpulse mit einer Trägerfrequenz von 38 KHz werden empfangen, demoduliert, vorverstärkt und an den Output gegeben. Durch den Pull-Up-Widerstand wird dieses Signal weiter verstärkt und an den Arduino Nano geleitet. Der analoge Eingang A5 des Arduinos kann als digitaler Eingang betrieben werden und wird hier benutzt um das Platinenlayout übersichtlich zu halten. Zur Filterung von Störsignalen dient der RC-Tiefpassfilter.

Tastendrücke auf einer Fernbedienung in diesem Frequenzbereich werden vom Empfänger als Hexadezimalcode weitergegeben. Bei Betrieb des Arduinos und zeitgleich verbundener serieller Schnittstelle, können die demodulierten Hexadezimalcodes über den Seriellen Monitor der Arduino IDE ausgegeben werden.

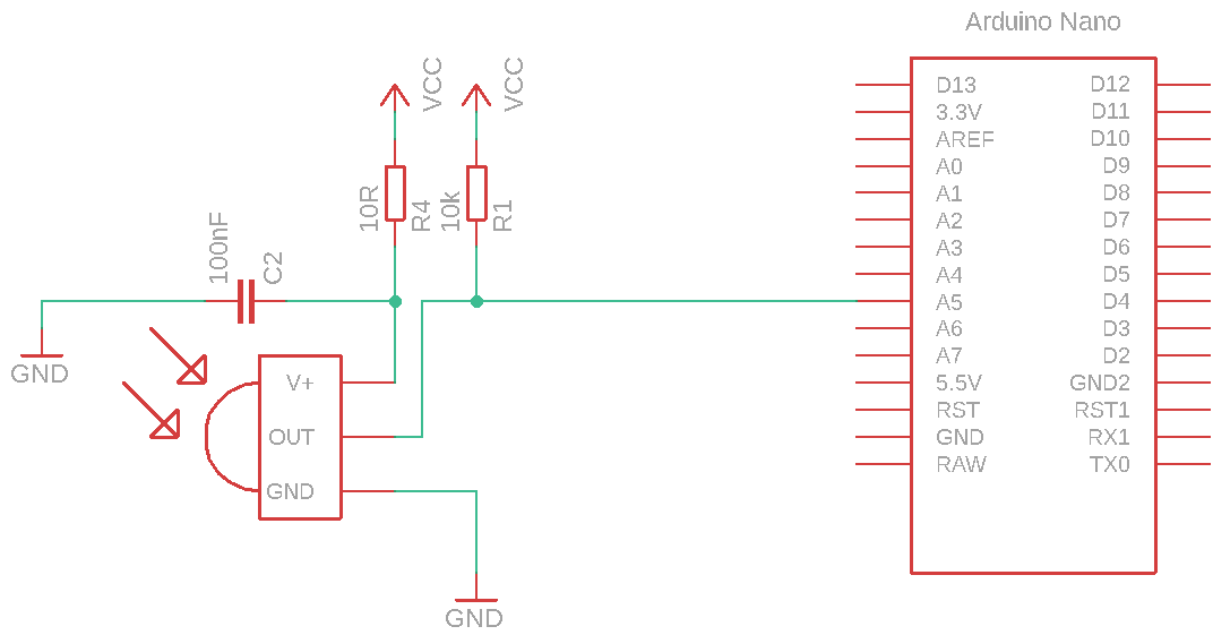


Abbildung 3.3: Infrarot-Empfänger mit Pull-Up-Widerstand und Filterkondensator.

3.3.3 Display

Das im RotaCon verwendete LCD-Display ist ein *TC1602B-01*. Wie in Kapitel 2.1 erläutert, besteht der RotaCon aus 2 Gehäusen. Neben dem IR-Empfänger befindet sich auch das Display in einem für den Nutzer sichtbaren Gehäuse, dass mit einem D-Sub-Kabel⁴ verbunden ist. Die meisten Pins sind direkt mit dem Arduino Nano verbunden, werden rein über Software gesteuert und sind in 3.3.6 beschrieben oder dem Datenblatt zu entnehmen. Display-Pin 3 setzt zusätzlich einen Spannungsteiler voraus um den Bildschirmkontrast festzulegen. Die im RotaCon benutzten Widerstandsgrößen sind durch den Einsatz eines Potentiometers ($10k\Omega$) ermittelt und ergeben die optimale Lesbarkeit bei einer Versorgungsspannung des verwendeten Display von 5V.

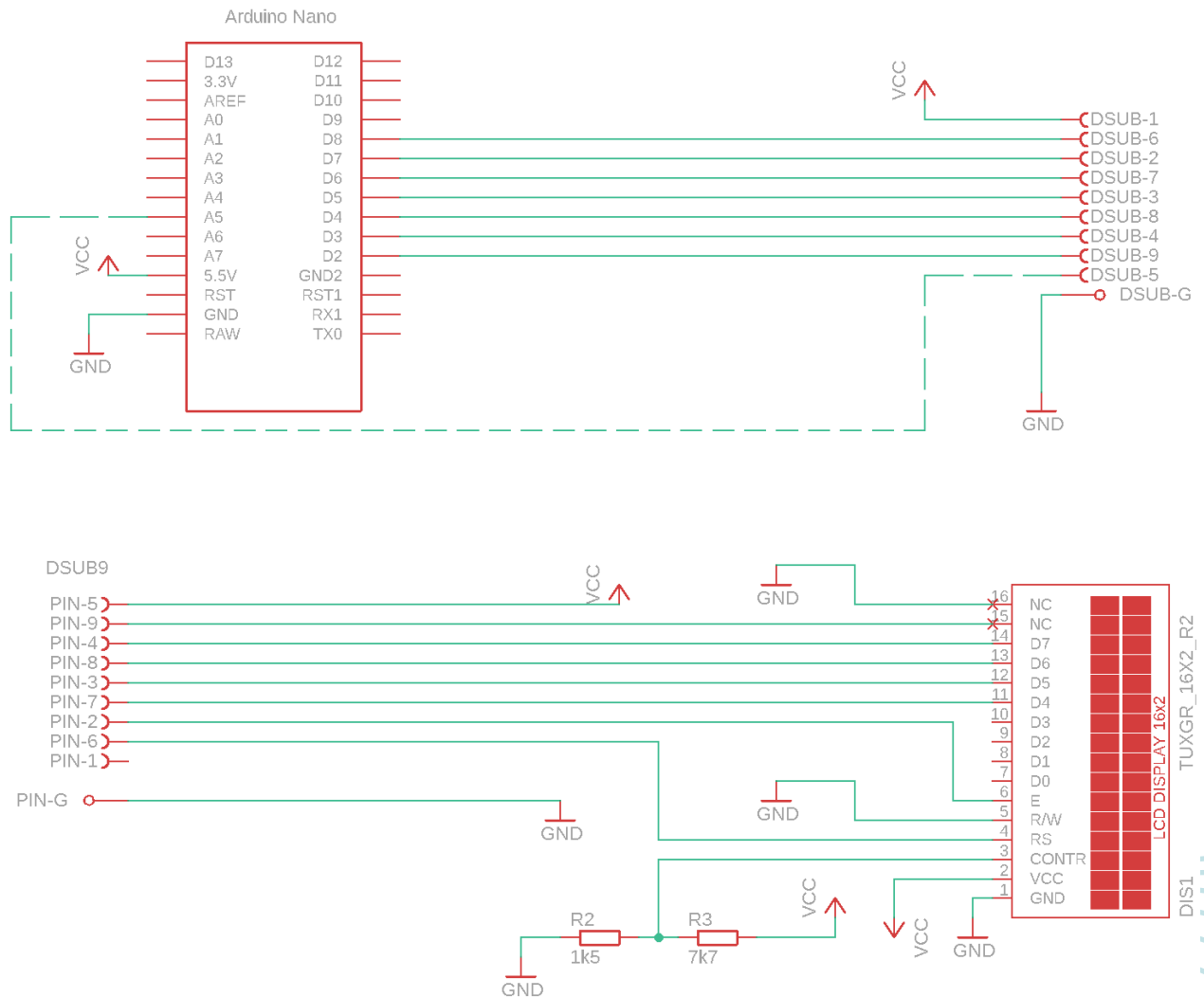


Abbildung 3.4: D-Sub-Pinbelegung für Display und IR-Receiver(gestrichelte Linie).

An dieser Stelle sei genannt, dass durch die Spiegelung der D-Sub-Schnittstelle die Pin-Nummern beachtet werden müssen. So ist beispielsweise die Versorgungsspannung bei dem D-Sub der Hauptplatine auf Pin 1, bei der Display-Platine jedoch auf Pin 5. (siehe Abbildung 3.4)

⁴D-Sub ist eine Bauform für Steckersysteme mit n Datenpolen und einer zusätzlichen Masse. Im RotaCon wird ein 9 Pol D-Sub benutzt

3.3.4 Anschlagerkennung durch Strommessung

LM324N Durch eine zusätzliche Schaltung wird der durch den Motor verbrauchte Strom gemessen. Dadurch wird untersucht, ob der Motor bei Erreichen der Maximalstellung des Potentiometers vom Normalstromverbrauch abweicht und dies softwaretechnisch zur Motorsteuerung beitragen kann. Durch einen nicht invertierenden Verstärker wird der Strom zu einer für den Arduino auswertbaren Spannung gewandelt. Die allgemeine Funktionsweise von Operationsverstärkern wird hier nicht weiter erläutert. Aufgrund der in Abschnitt 3.3.1 genannten Sättigung wird die Ausgangsspannung (U_a) der Operationsverstärkerschaltung als Ausgangsgröße auf 4V festgelegt. Diese soll durch den in den Arduino integrierten Analog-Digitalwandler ausgewertet werden können. Für einen möglichst geringen Leistungsverlust, werden 2 parallel geschaltete Strommess-Widerstände (R_1 und R_2 in Abb.3.5) eingesetzt und die Eingangsspannung (U_e) auf 0,5V festgelegt. R_3 und R_4 in der Schaltung haben einen Widerstand von $4,7\text{ k}\Omega$. Durch die angenommenen Spannungen ergibt sich ein benötigter Verstärkungsfaktor (V) von 8, welcher wiederum zur Berechnung von R_5 ausschlaggebend ist.

$$V = \frac{U_a}{U_e} = \frac{4V}{0,5V} = 8$$

$$V = 1 + \frac{R_5}{R_4}$$

$$R_5 = R_4 \cdot (V - 1) = 32900\Omega \approx 33\text{ k}\Omega \quad (3.4)$$

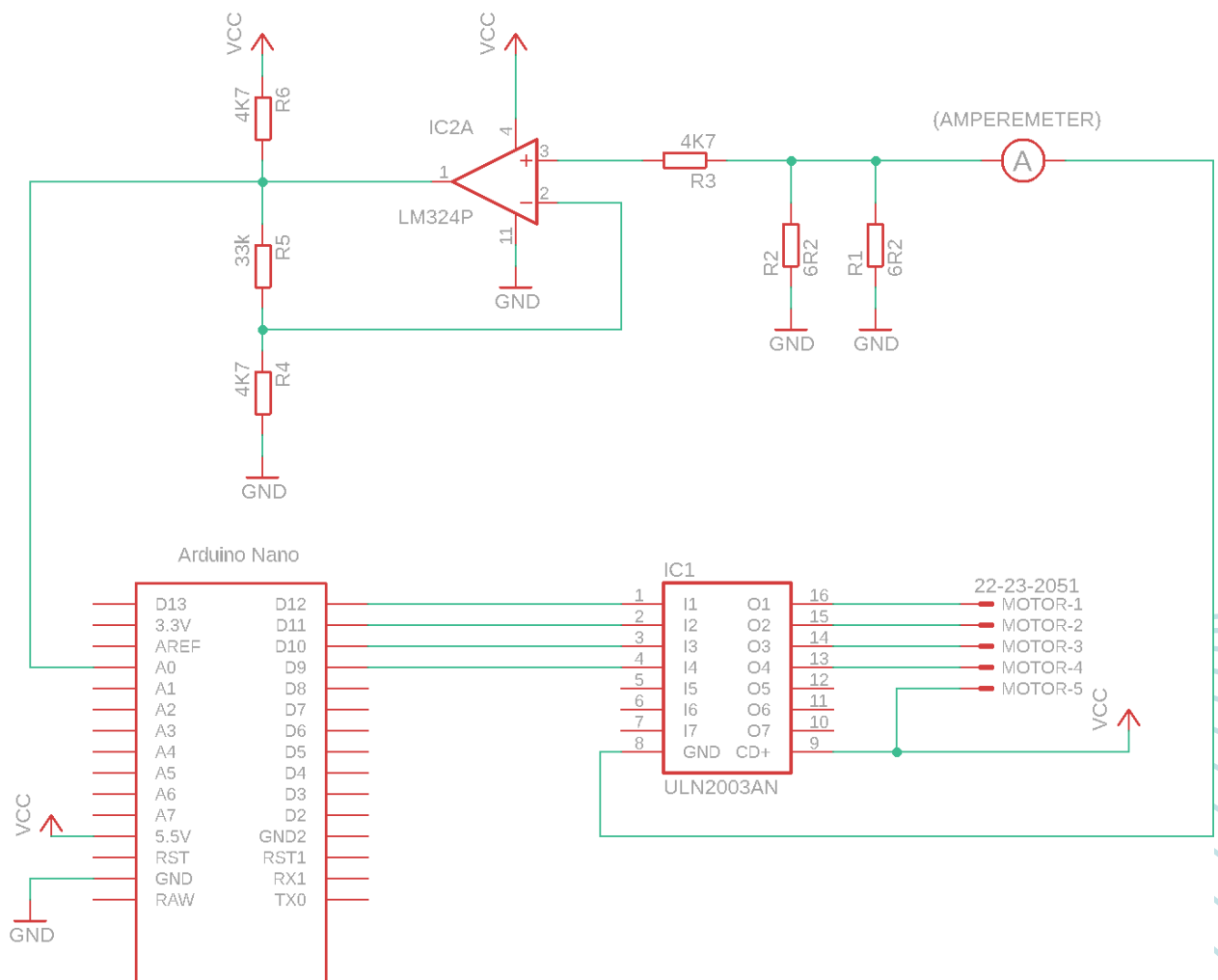


Abbildung 3.5: Einsatz eines nicht-invertierenden Verstärkers zur Strommessung

Der in Abschnitt 3.3.1 ermittelte Strom wird für diese Berechnung auf 160 mA festgelegt und teilt sich hälftig auf die beiden Messwiderstände R1 und R2 auf an denen je die Eingangsspannung U_e anliegt. Durch das ohmsche Gesetz leiten sich daher die Widerstandsgrößen her.

$$U_e = R1 \cdot \frac{I}{2} = 0,5V \qquad R1 = R2 = \frac{2 \cdot U_e}{I} = \frac{1V}{0,16A} = 6,25\Omega \equiv 6R2 \qquad (3.5)$$

Bei Schaltung einer Phase des Motors misst der Arduino mit dem integrierten A/D-Wandler 3,5V. Durch die bekannte Verstärkung⁵ errechnet sich so ein U_e von 0,441V am Widerstand R1. Daraus ergibt sich eine errechnete Stromstärke von 146 mA. Das Amperemeter⁶ gemäß Abb. 3.5 misst 144mA als Maximalwert. Somit entsprechen die im seriellen Monitor der Arduino IDE gelisteten Werte den etwa tatsächlichen Werten der Schaltung und können ausgewertet werden.

Induktivitäten benötigen eine gewisse Zeit um sich zu entladen und ein Schritt des RotaCons entspricht 64 Schaltvorgängen, die mit einer Frequenz von 500Hz (siehe Abschnitt 3.4) stattfinden. Somit wird jede Phase einmal alle 8ms geschaltet wobei 2 Phasen zu je einer Induktivität gehören. Der Strom ist daher nicht konstant und der Zeitpunkt des Messens muss möglichst nah am Schaltzeitpunkt liegen.

⁵Die in dieser Rechnung verwendete Verstärkung ist 7,94 und bezieht sich auf die tatsächlich gemessenen Werte der verbauten Widerstände

⁶Verwendetes Multimeter: testo 760-2

3.3.5 Anschlaggerkennung durch Endschalter

Redundant zur Strommessung und Auswertung besitzt der RotaCon zwei Endschalter, diese sind Mikroschalter mit Flachhebel und im "normally open" -Modus betrieben. Bei dem Prototypen sind diese Schalter durch Klebmasse befestigt und können je nach Potentiometer beliebig auf dem RotaCon-Gehäuse befestigt werden. Motorachse und Potentiometer sind durch eine flexible Wellenkupplung verbunden. An dieser Kupplung ist zusätzlich eine 25mm Schraube mit M4-Gewinde eingeschraubt, die als Hebel zum Auslösen der Schalter dient. Sobald einer der Schalter ausgelöst wird, wechselt der entsprechende Pin des Arduinos auf LOW und der Motor wird durch Software für diese Drehrichtung gestoppt.

3.3.6 Pinbelegungen

Durch die Verwendung zweier Platinen und resultierender Kabelverbindung, sind einzelne Signalpfade nur schwer nachzuvollziehen. Folgende Tabelle bietet eine Übersicht aller Pins. Die Platinen sind abgekürzt mit DB(Display-Board) und MB (Motor-Board).

Tabelle 3.1: Pin-Belegungen

Arduino	DSUB MB	DSUB DB	Leiste DB	Display	Symbol	Level	Beschreibung
-	-	-	Pin 1	Pin 1	VSS/GND	0V	Ground
-	-	-	Pin 2	Pin 2	VDD/VCC	5V	DP Versorgungsspannung
-	-	-	Pin 3	Pin 3	Vo/CONTR	-	LCD-Kontrast
D2	Pin 9	Pin 6	Pin 4	Pin 4	RS	H/L	H: Datensignal, L: Befehl
-	-	-	Pin 5	Pin 5	R/W	H/L	H: Lesen, L: Schreiben
D3	Pin 4	Pin 2	Pin 6	Pin 6	E	H/L	Enable Display
-	-	-	-	Pin 7-10	Do-D3	H/L	Display-Daten
D4	Pin 8	Pin 7	Pin 7	Pin 11	D4	H/L	Display-Daten
D5	Pin 3	Pin 3	Pin 8	Pin 12	D5	H/L	Display-Daten
D6	Pin 7	Pin 8	Pin 9	Pin 13	D6	H/L	Display-Daten
D7	Pin 2	Pin 4	Pin 10	Pin 14	D7	H/L	Display-Daten
D8	Pin 6	Pin 9	Pin 11	Pin 15	A	5V	LED-Backlight Anode
-	-	-	Pin 12	Pin 16	K	0V	LED-Backlight Kathode
D9	-	-	-	-	Orange	H/L	Motor (Spule 1a)
D10	-	-	-	-	Gelb	H/L	Motor (Spule 2a)
D11	-	-	-	-	Pink	H/L	Motor (Spule 1b)
D12	-	-	-	-	Blau	H/L	Motor (Spule 2b)
A5	Pin 5	Pin 1	-	-	-	H/L	IR-Receiver
-	Pin 1	Pin 5	-	-	VCC	5V	VCC
A0	-	-	-	-	-	0-5V	A/D Strommessung Motor
A1	-	-	-	-	-	H/L	Taster 1
A2	-	-	-	-	-	H/L	Taster 2

3.3.7 Gesamtschaltung der Display-Platine

Da das Display durch seine Platzierung im Displaygehäuse von der Platine getrennt ist, wird eine Pinleiste auf der Platine eingesetzt. Display und Platine sind durch Kabelbrücken miteinander verbunden.

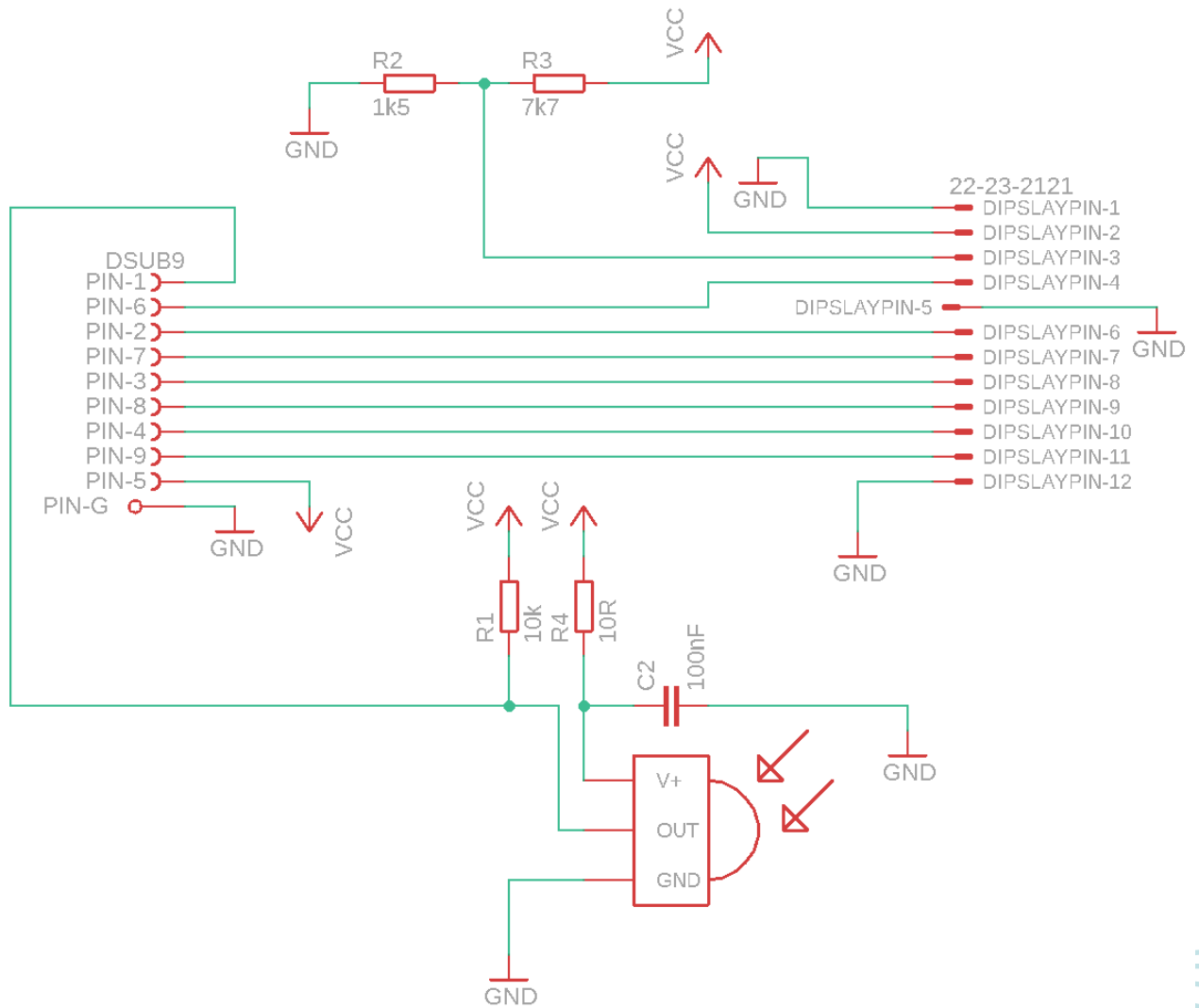


Abbildung 3.6: Finale Schaltung der Display-Platine.

3.3.8 Gesamtschaltung der Motor-Platine

Die Endschalter sind ebenfalls separat zur Hauptplatine und daher nur als Steckverbindung zu erkennen. Diverse Testpunkte sind in die Schaltung integriert. Die USB-Spannungsquelle sowie zusätzliche Klemmen für eine redundante Spannungsversorgung wurden hinzugefügt. Ein Anschluss für den Betrieb eines weiteren Ports des Arduinos um eventuelle Erweiterungen durchzuführen ist ebenfalls implementiert (A3).

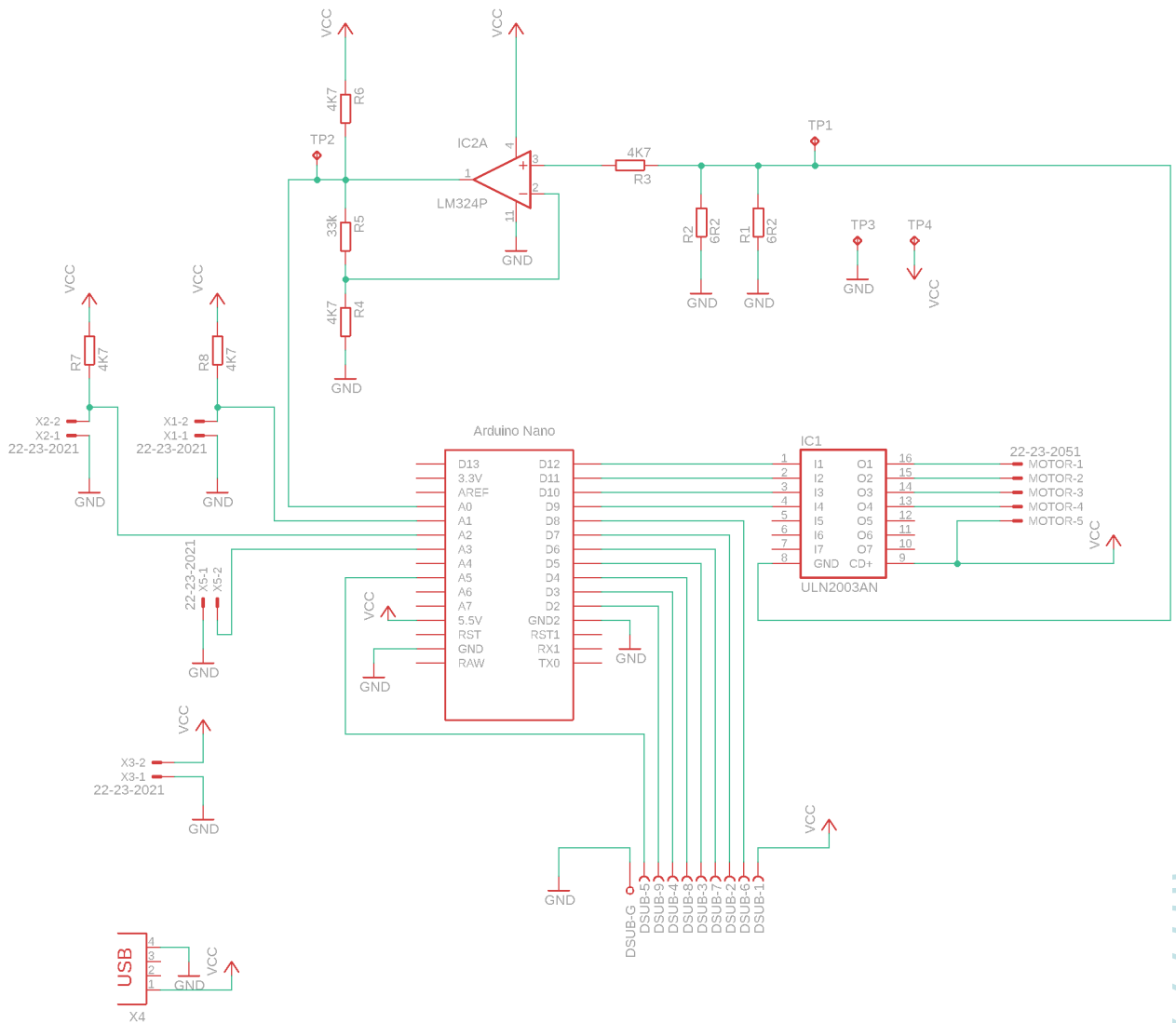


Abbildung 3.7: Finale Schaltung der Motor-Platine.

3.4 Software

Die Schaltung des Motors ist einfach gehalten. Die vier Phasen-Ports sind über eine Liste angesteuert in der jeweils ein Wert auf HIGH und der Rest auf LOW gesetzt ist. Je nach Rotationsrichtung des Motors muss das HIGH-Bit nun innerhalb der Liste verschoben werden. Da es bei dem Einphasenbetrieb im Vollschrittverfahren nur vier Varianten dieser Liste gibt, wurde hier eine 4x4 Matrix implementiert deren Zeilen je einer dieser Listen entsprechen:

```
int stepMatrix[4][4] = {  
    {HIGH, LOW, LOW, LOW},  
    {LOW, HIGH, LOW, LOW},  
    {LOW, LOW, HIGH, LOW},  
    {LOW, LOW, LOW, HIGH},  
};
```

Eine volle Umdrehung der Motorachse benötigt 2048 dieser "Zeilensprünge" (siehe 3.2). Als Anhalt für den maximalen Drehwinkel eines Potentiometers wird 320° angenommen. Dies entspricht etwa 1820 Motorschritten. Für die angestrebten 28 bis 30 Lautstärkestufen des RotaCons bedeutet das, dass je 64 Motorschritte für eine Lautstärkestufe passieren müssen. Damit Rotor und Getriebe sich in Bewegung setzen können, ist bewusst ein Delay zwischen den einzelnen Schritten implementiert.

Unmittelbar nach jedem Schritt wird die am A/D-Wandler liegende Spannung abgefragt, die den Motor-Stromverbrauch repräsentiert. Der tiefste Wert nach 64 Schritten wird übergeben und dient zur Feststellung ob der Motor den Endanschlag des Potentiometers erreicht hat. Sind die Werte überhalb eines Schwellwertes, ist dies ein Anzeichen, dass der Rotor starr zwischen den schaltenden Elektromagneten steht. Der Schwellwert wurde über Tests im seriellen Monitor ermittelt und liegt bei 2,8V.

Wird somit ein Anschlag erkannt, oder einer der beiden Taster betätigt, ist die Motorsteuerung für eine Rotation in diese Richtung gesperrt und auf dem Display wird eine Warnung ausgegeben.

Für die Displaysteuerung und Datenauswertung des Infrarotempfängers werden entsprechende Bibliotheken benutzt. Eine "Switch-Case-Anweisung" koordiniert die Eingaben der Fernbedienung. So kann jeder Hexadezimalwert vom IR-Empfänger einer Methode zugewiesen werden. Hält der Benutzer eine beliebige Taste der Fernbedienung gedrückt, so sendet sie den Wert "0xFFFFFFFF". Dieser Wert steht für die Wiederholung des zuletzt gesendeten Hex-Codes, daher ist dieser Wert gesondert in der Software betrachtet. (siehe Anhang 5.5) Die aktuelle Motorposition, sowie Lautstärke wird im internen EEPROM-Speicher abgelegt. Sollte der RotaCon vom Strom getrennt werden, bleiben diese Werte somit unverändert.

Die komplette Software ist in dem GitHub-Projekt einsehbar, daher werden weitere Details hier nicht weiter erläutert um den Umfang dieser Dokumentation übersichtlich zu halten.

3.5 Das Gehäuse

Dieser Abschnitt bezieht sich auf Feststellungen aus Kapitel 2.3. Alle Angaben beziehen sich auf die in diesem Projekt verwendeten Materialien. So sind die Maße der Platinen und deren Bohrungen zum Beispiel ausschlaggebend für den Gehäuseentwurf.

3.5.1 Motorgehäuse

Für den Prototypen wurde ein Gehäuse entworfen, dass weder verklebt noch vielseitig verschraubt werden muss. Somit kann zu Entwicklungszwecken ein häufiges unkompliziertes Öffnen stattfinden. Exakte Maße sind den detaillierteren technischen Zeichnungen im Anhang zu entnehmen. Ausmaße des Gehäuses ergeben sich durch Platinen- und Motorgröße, da der Motor komplett im Gehäuse untergebracht ist. Dieses Gehäuse besteht aus zwei einzeln gefertigten Teilen, die mit einem 3D-Drucker produziert werden. 3D-Modell und notwendige Dateien zum Druck sind im GitHub-Projekt abgelegt, variieren jedoch je nach verwendetem Filament und Drucker.

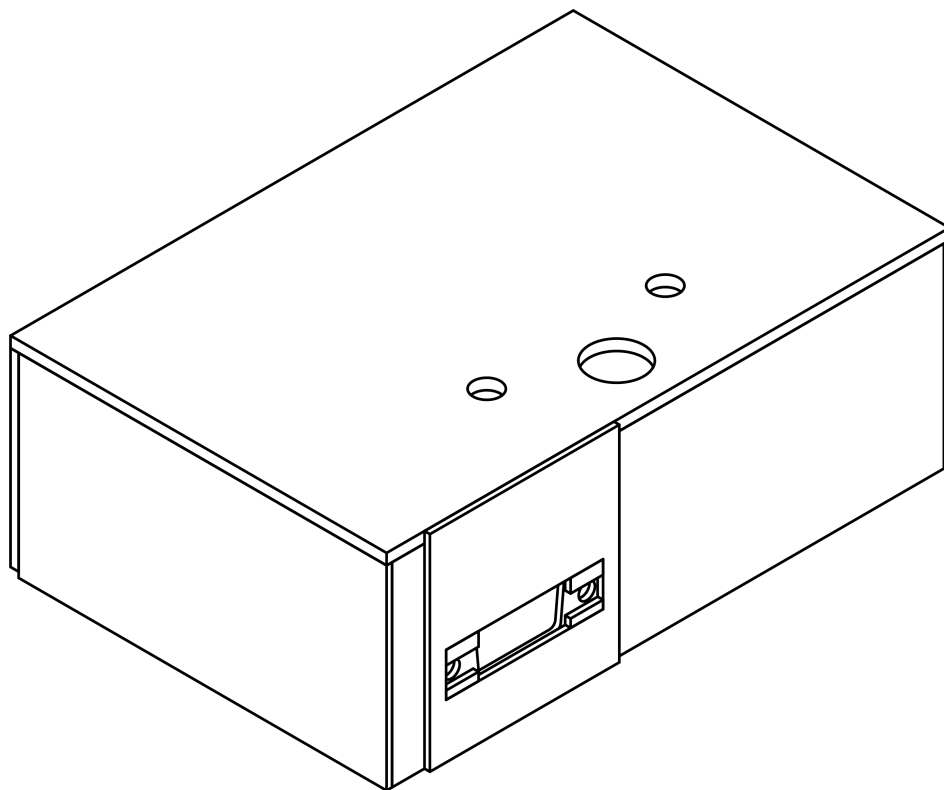


Abbildung 3.8: Skizze Des Motorgehäuses

3.5.2 Displaygehäuse

Auch bei diesem Gehäuse wurde darauf geachtet, dass es möglichst einfach geöffnet werden kann. Höhe und Breite sind durch die Displaymaße beeinflusst. Die Öffnung für den IR-Empfänger ist mit möglichst dunkler Folie verdeckt, da Licht der Displaybeleuchtung sonst nach außen scheint. Display und Platine sind in diesem Gehäuse auf Plastikstifte gesteckt und halten ohne Einsatz von Heißkleber oder Schrauben. Durch das 40cm lange D-Sub-Kabel kann dieses Gehäuse frei auf der Oberseite des Lautsprechers platziert werden. Zum Einen verbinden keilförmige Haken als Klemmverschluss die beiden Gehäuseteile miteinander, zum Anderen sorgt die D-Sub-Steckverbindung für den Zusammenhalt.

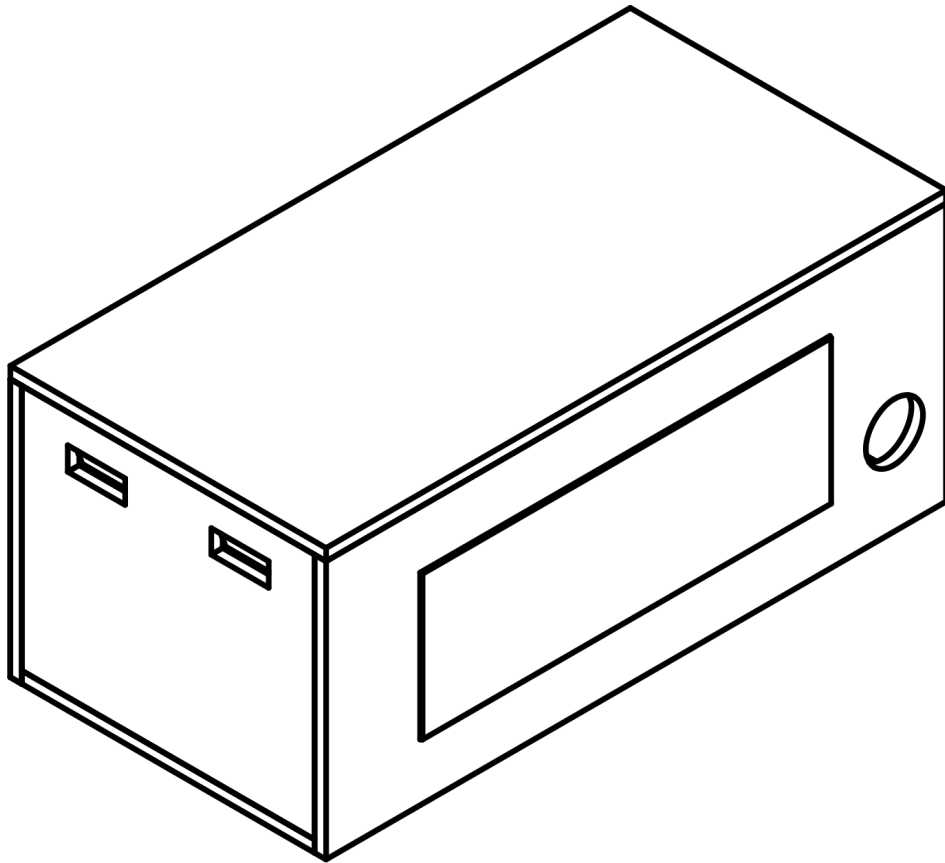


Abbildung 3.9: Skizze Des Displaygehäuses

3.5.3 Gehäuseständer

Das Gehäusestativ ist provisorisch und aus entwicklungszeitlichen Gründen nur für den Prototypen gestaltet. Eine Gewindestange ist in eine Einschraubmutter des lackierten Holzfußes eingeschraubt und durch ein an das Motorgehäuse geklebtes Winkelblech gesteckt. Durch Einsatz zweier Flügelmutter und Unterlegscheiben lässt sich das Gehäuse somit sehr fein in der Höhe verstellen. Dieses Winkelblech kann bei zukünftigen Versionen zum Beispiel durch Laschen am Motorgehäuse direkt ersetzt werden. Angenehmer für den Benutzer wäre ein Gewinde innerhalb dieser Laschen und die gewindelose Lagerung der Gewindestange im Fuß, so könnte die Gehäusehöhe durch Rotation der Gewindestange eingestellt werden (z.B. via Motor oder Kurbel).

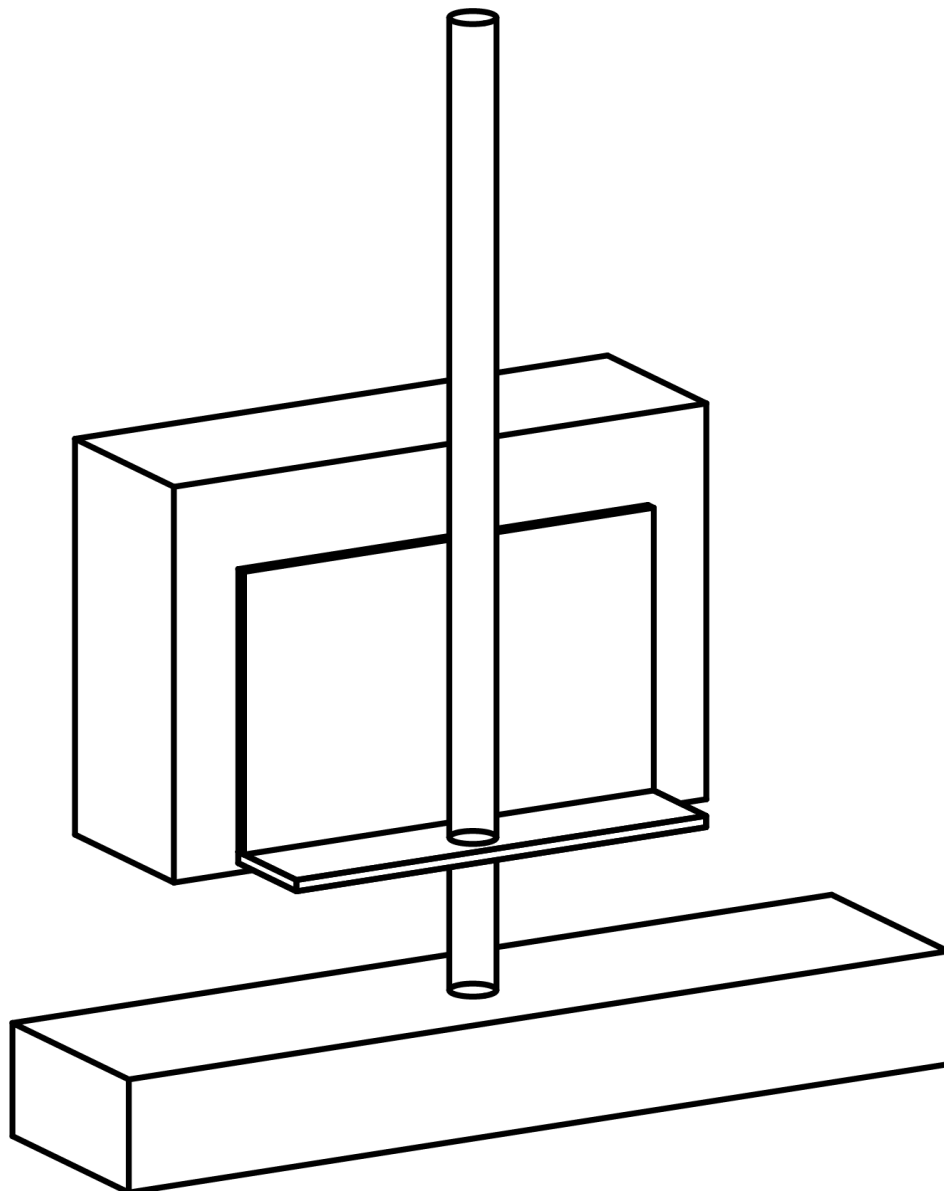


Abbildung 3.10: Gehäuseständer des Prototyps

Kapitel 4

Ergebnis

In diesem Kapitel wird die Inbetriebnahme des Prototypen geschildert. Probleme, die bei diesem Projekt aufgetreten sind werden dargestellt und Lösungsansätze erwähnt. Nach einer kurzen Zusammenfassung des RotaCon-Projekts folgt ein kurzer Abschnitt über mögliche Erweiterungen durch Folgeprojekte.

4.1 Inbetriebnahme

1. Abnehmen der Potentiometerkappe am Lautsprecher
2. Einstellen der Höhe des Motorgehäuses durch Flügelmuttern
3. Festschrauben der Wellenkupplung (Innensechskant 2mm)
4. Potentiometer mit Wellenkupplung auf Minimalwert stellen
5. Anbringen des ersten Tasters durch Klebmasse sodass Hebel gedrückt
6. Stellung auf Maximalwert und letzten Schritt wiederholen
7. Verbinden beider RotaCon-Gehäuse mit D-Sub-Kabel
8. Anschließen der Spannungsversorgung
9. Regeln des Potentiometers per Fernbedienung auf Minimalstellung
10. Auslösen des Minimalwert-Tasters setzt "Volume" auf o
11. Der RotaCon ist betriebsbereit

4.2 Problem Diskussion

- 1 Das D-Sub-Verbindungskabel ist sperrig und schränkt die Platzierung des Display-Gehäuses ein. Durch den Einsatz eines I^2C -Datenbuses können die Pinbelegungen und somit auch die Anzahl der Leitungen im Kabel so reduziert werden, dass ein flexibleres Micro-USB-Kabel verwendet werden kann.
- 2 Die Platzierung der Endschalter mit Klebmasse ist unzuverlässig aber für den ersten Prototypen ausreichend. Eine in das Gehäuse integrierte Ringscheibe auf der die beiden Endschalter verschoben und befestigt werden können ist eine zweckmäßigere Variante. Darauf wurde jedoch aus Kosten- und Aufwandsgründen bisher verzichtet.

3 Die Strommessung ist ein guter Ansatz, jedoch noch nicht ausreichend um sichere Aussagen über die Stellung des Potentiometers treffen zu können. Je nach Menge der Eingangssignale von der Fernbedienung oder der Leuchtgängigkeit des Potentiometers variieren die gemessenen Werte. Weitere Tests durch Variation des Motorbetriebes (Softwaretechnisch) und damit verbundener Änderung der Messwiderstände können Abhilfe schaffen, wurden aus zeittechnischen Gründen in diesem Projekt jedoch nicht durchgeführt.

4

4.3 Zusammenfassung

4.4 Ausblick

Kapitel 5

Anhänge

5.1 Tabelle Teileübersicht

5.2 Datenblätter

Eine Tabelle mit der Auflistung aller Einzelteile mit anschließenden wichtigen Bereichen der Datenblätter. Die gesamten Datenblätter werden nicht eingefügt, jedoch ein Link der zu einem derzeitigen Datenblatt im Internet führt.

5.3 Layouts

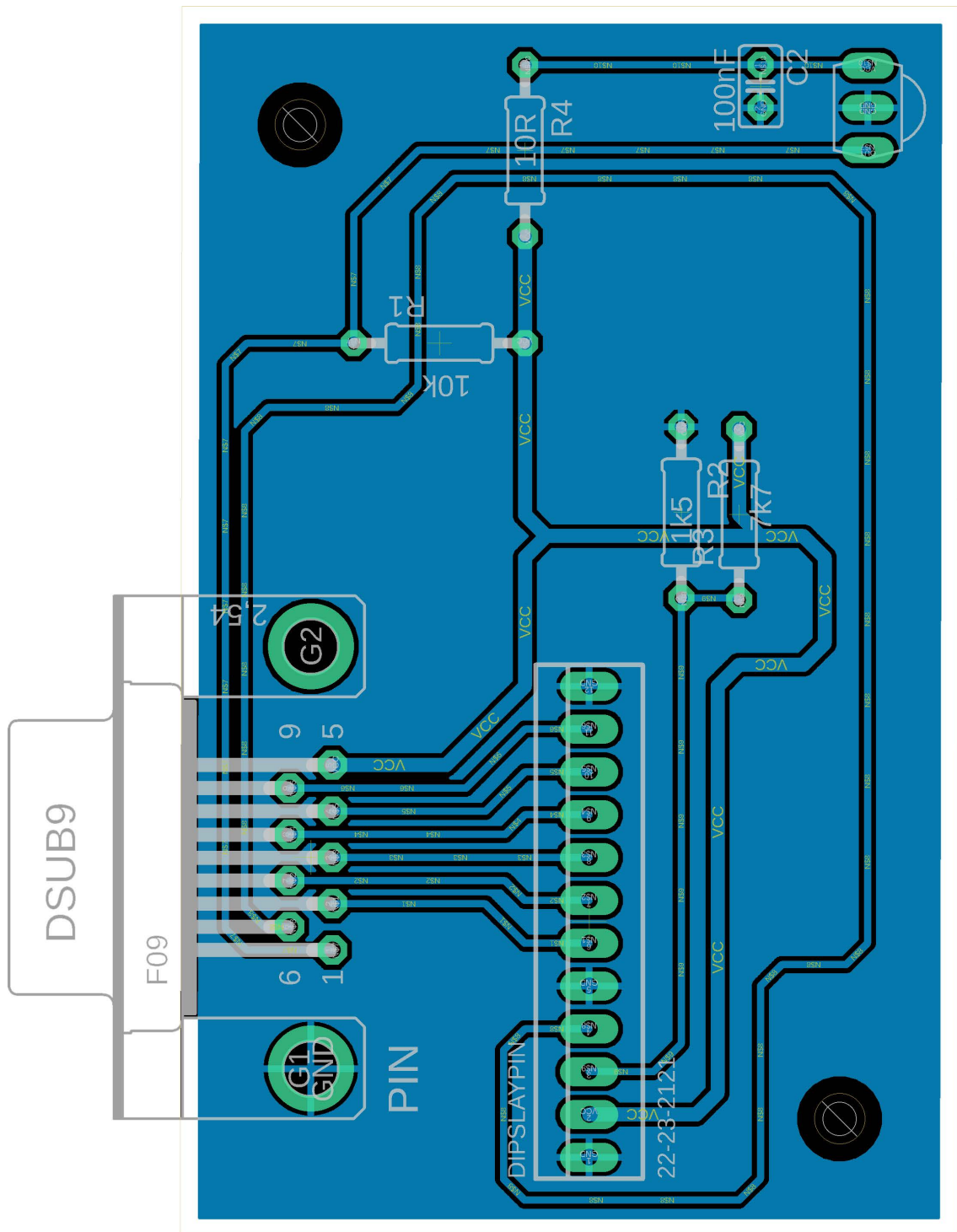


Abbildung 5.1: Display-Platine

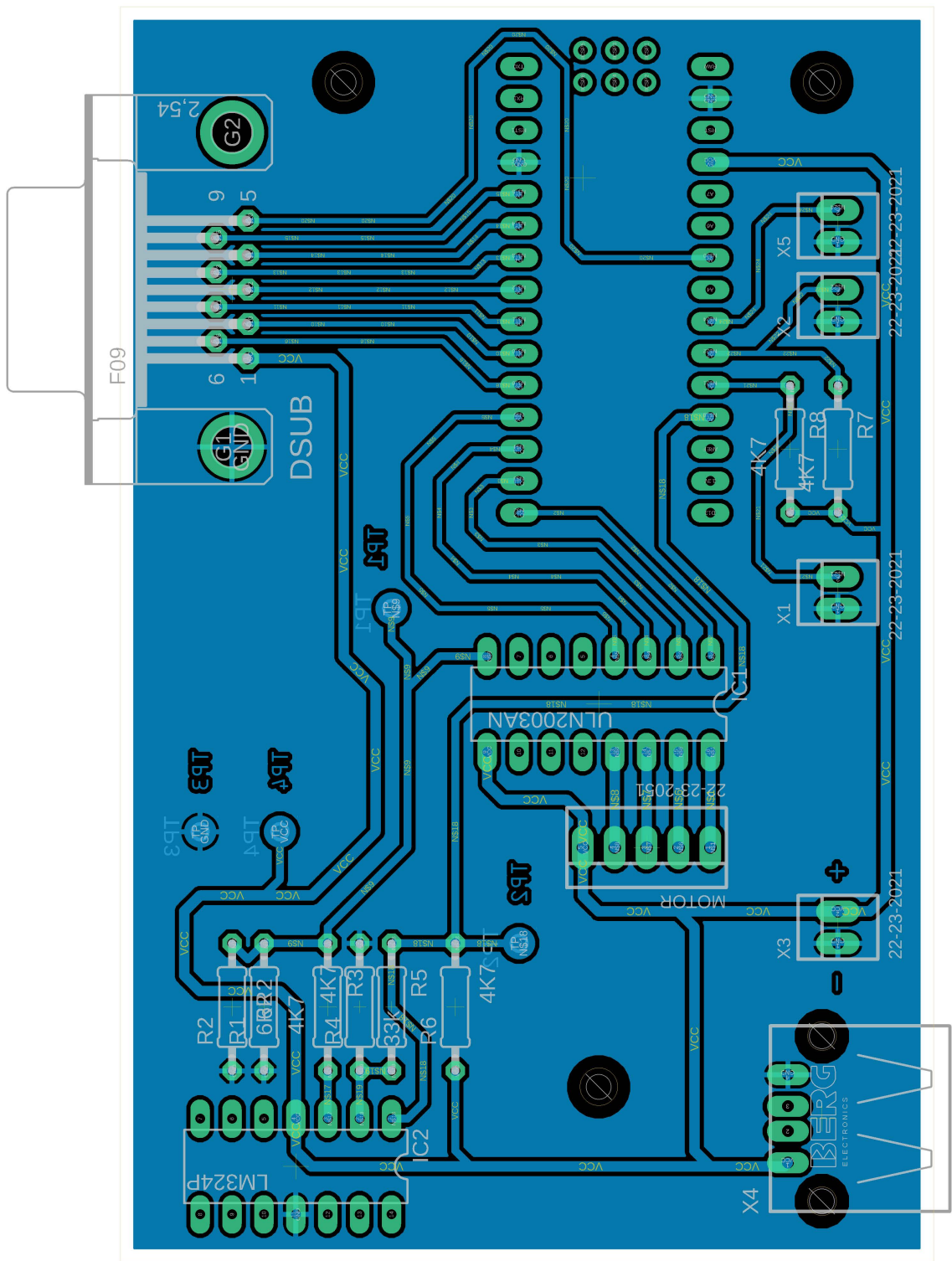
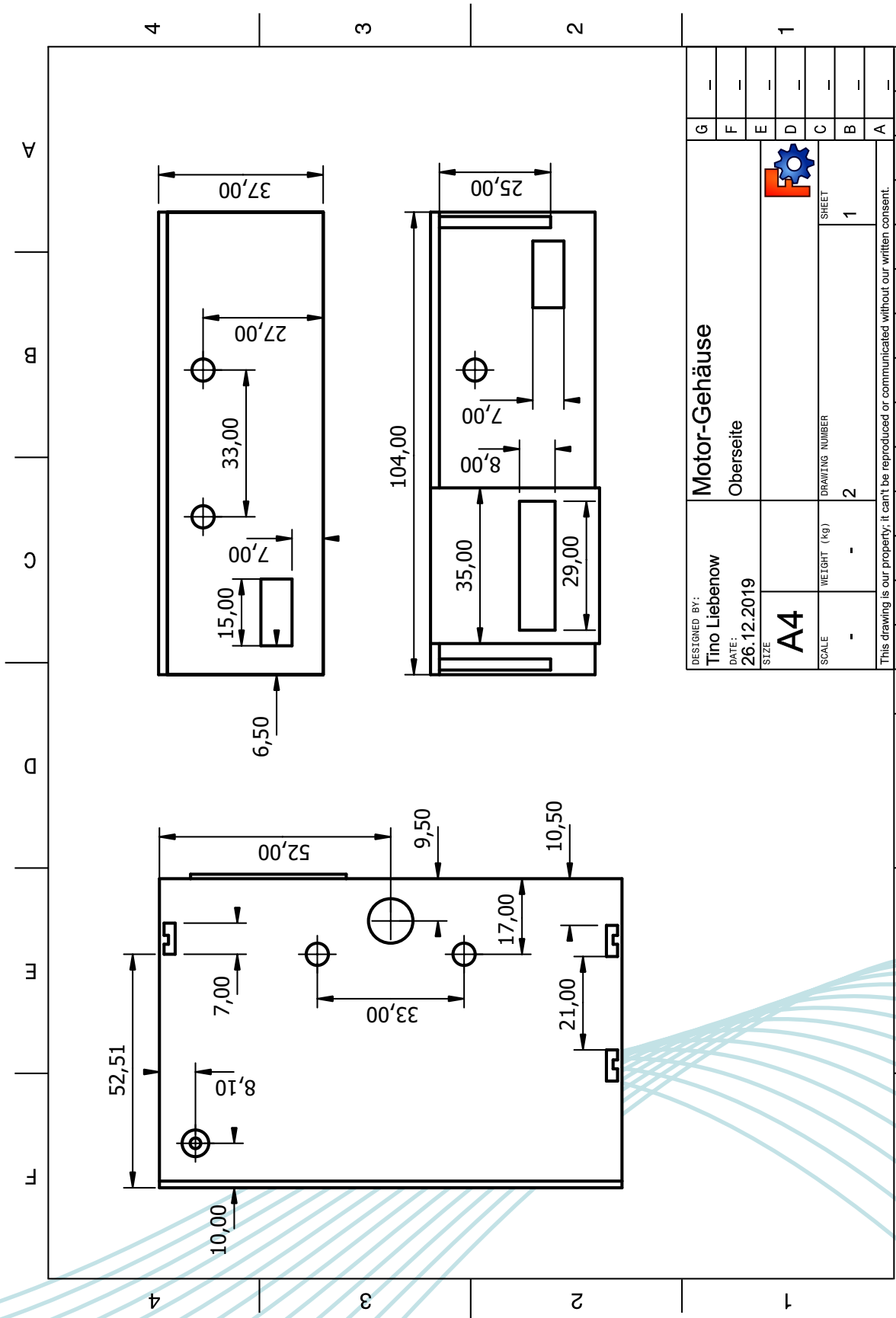

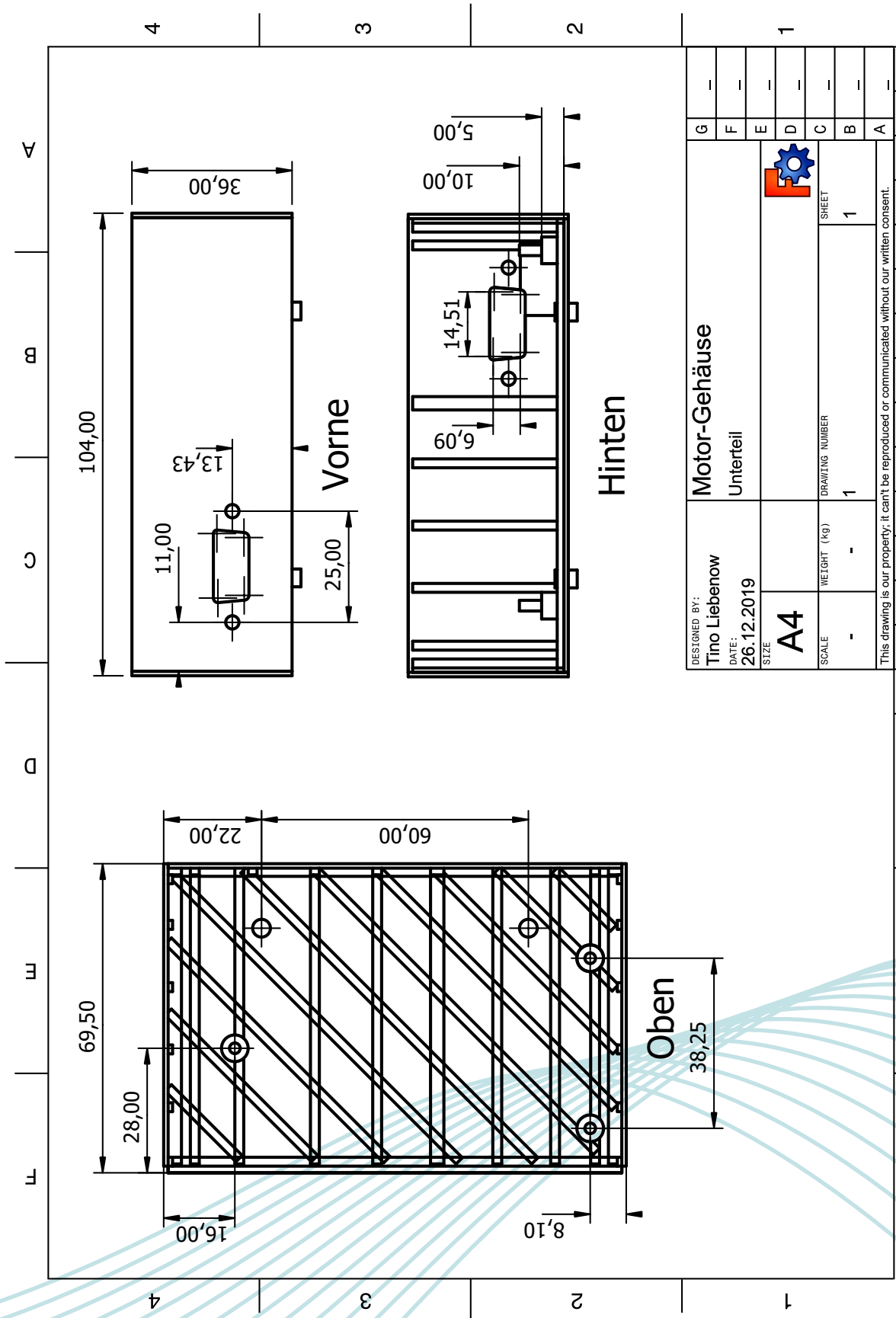



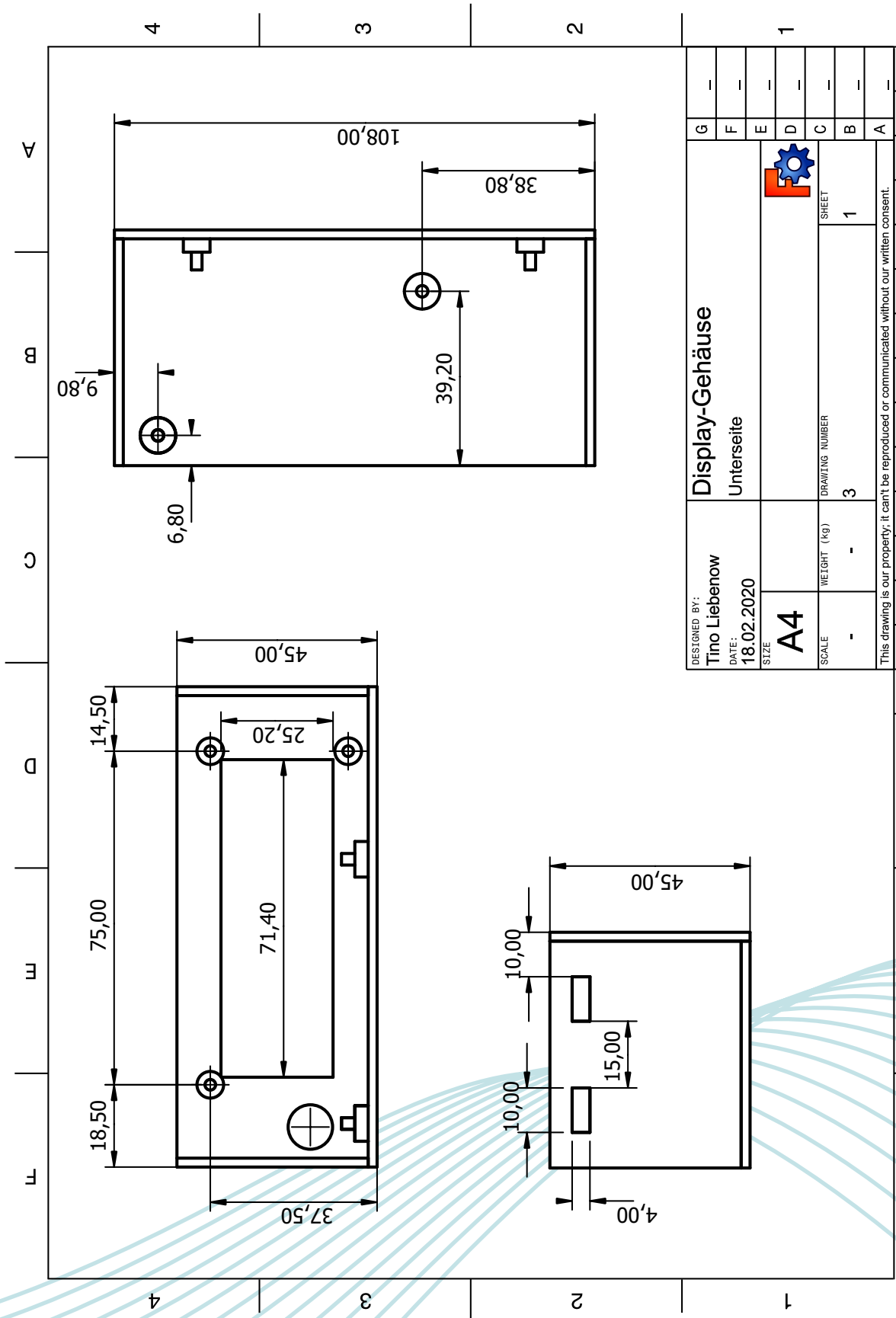
Abbildung 5.2: Motor-Platine




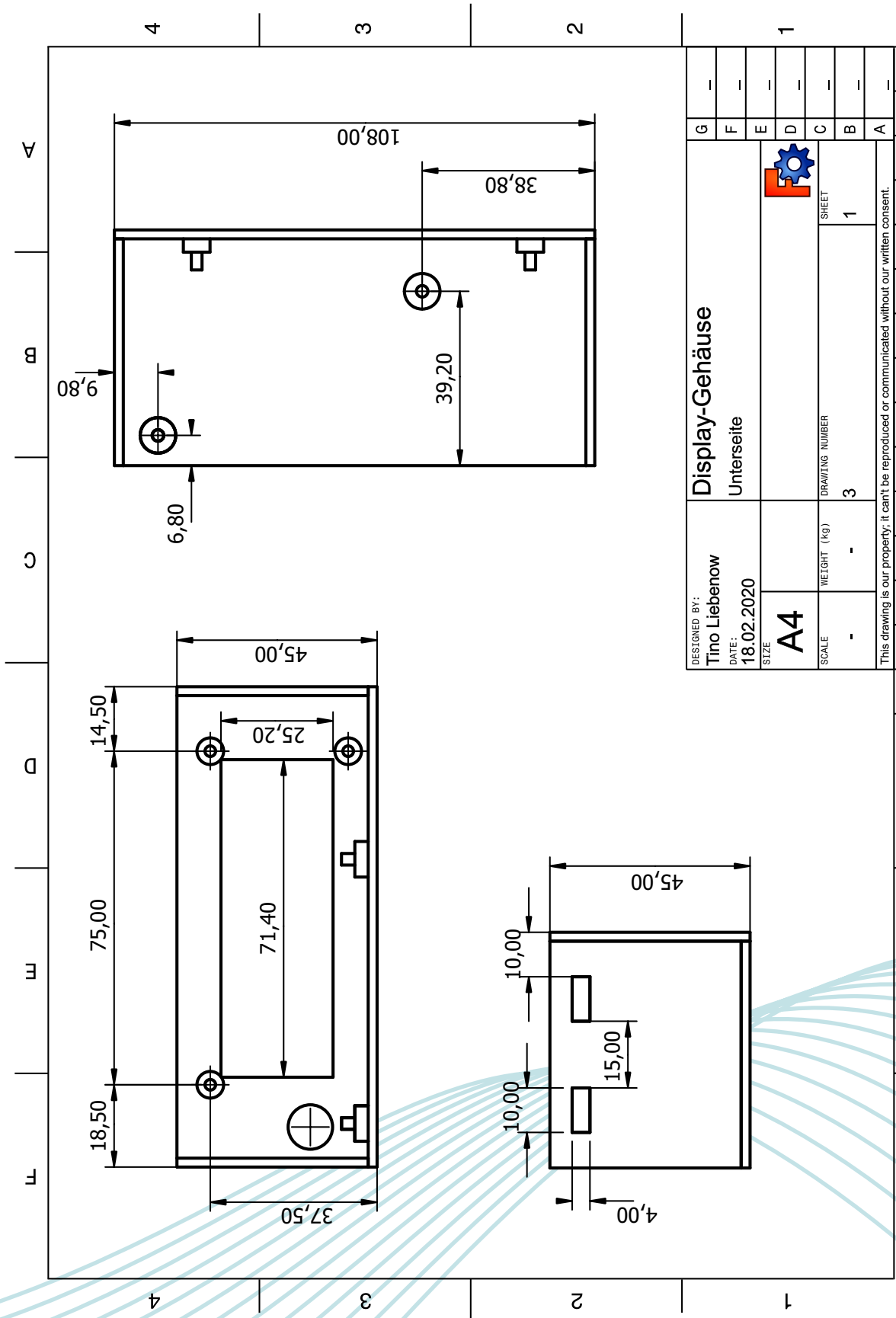
DESIGNED BY: Tino Liebenow		Motor-Gehäuse		G		—		
DATE: 26.12.2019		Oberseite		F		—		
SIZE A4			SHEET 1		E		—	
					D		—	
SCALE	—		DRAWING NUMBER 2	C		—		
	—			B		—		
This drawing is our property, it can't be reproduced or communicated without our written consent.			A		—			




DESIGNED BY: Tino Liebenow DATE: 26.12.2019	Motor-Gehäuse		G	-
	Unterteil		F	-
		SHEET 1	E	-
			D	-
			C	-
			B	-
SIZE A4	WEIGHT (kg) -	DRAWING NUMBER 1	A	-
This drawing is our property; it can't be reproduced or communicated without our written consent.				



DESIGNED BY: Tino Liebenow		Display-Gehäuse		G		—		
DATE: 18.02.2020		Unterseite		F		—		
SIZE A4			SHEET 1		E		—	
					D		—	
SCALE	WEIGHT (kg)		DRAWING NUMBER		C		—	
-	-		3		B		—	
This drawing is our property; it can't be reproduced or communicated without our written consent.			A		—			



DESIGNED BY: Tino Liebenow		Display-Gehäuse				G	—	
DATE: 18.02.2020		Unterseite				F	—	
SIZE A4							E	—
							D	—
SCALE	WEIGHT (kg)		DRAWING NUMBER		SHEET	C	—	
-	-		3	1	B	—		
This drawing is our property; it can't be reproduced or communicated without our written consent.				A	—			

5.5 Quellcode

Ausschnitte des Quellcodes zur Motorsteuerung:

```
int arrayDefault[4] = {LOW, LOW, LOW, LOW};
int outArray[4];
int stepMatrix[4][4] = {
    {HIGH, LOW, LOW, LOW},
    {LOW, HIGH, LOW, LOW},
    {LOW, LOW, HIGH, LOW},
    {LOW, LOW, LOW, HIGH},
};

void writeStep(int outArray[4]) {
    digitalWrite(9, outArray[0]);
    digitalWrite(10, outArray[1]);
    digitalWrite(11, outArray[2]);
    digitalWrite(12, outArray[3]);
}

void moveOneStep(bool dir) { //dir = Richtung des Steps
    if (currentStep >= 0 || currentStep <= 3) {
        writeStep(stepMatrix[currentStep]);
    } else {
        writeStep(arrayDefault);
    }
    (dir == true) ? (currentStep++) : (currentStep--);
    if (currentStep > 3) {
        currentStep = 0;
    } else if (currentStep < 0) {
        currentStep = 3;
    }
    EEPROM.put(1, currentStep);
}

void moveSteps(bool dir, int steps) {
    float lowestVoltage = 5.0;
    for (int i = 0; i < steps; i++) {
        moveOneStep(dir);
        float u=currentVoltage();
        if (u < lowestVoltage) {
            lowestVoltage = u;
        }
        delay(2);
    }
    lcd.clear();
    uaMin=lowestVoltage;
}
```


Auschnitte der Messung und Auswertung der IR-Signale:

```
void check() {
    if (uaMin > 2.8) {
        impact = false;
    } else {
        Serial.println("ANSCHLAG");
        impact = true;
    }
}

float currentVoltage() {
    return analogRead(messPinUa) * 5.0 / 1024;
}

void motorControl(unsigned long buttonID) {
    switch (buttonID) {
        //Remote pressed "+"
        case 0xFF02FD:
            if (!maxHit) {
                moveSteps(true, 1 * 64);
                Serial.println(uaMin);
                check();
                if (!impact) {
                    volume++;
                } else {
                    lcd.print("STOP");
                    delay(2000);
                    lcd.clear();
                }
            } else {
                lcd.print("MAX");
            }
            EEPROM.put(0, volume);
            lastButton = buttonID;
            break;

        //Remote hold button
        case 0xFFFFFFFF:
            motorControl(lastButton);
            break;
    }
}
```

Literatur

- [1] o.A. *Lautstärkereger*. 2020. URL: <https://de.wikipedia.org/wiki/Lautst%C3%A4rkereger> (besucht am 12.02.2020) (siehe S. 5).
- [2] Thomas Brühlmann. *Arduino Praxiseinstieg. Behandelt Arduino 1.0*. mitp, 2012. ISBN: 978-3-8266-8342-8 (siehe S. 8).
- [3] o.A. *Arduino Nano*. o.J. URL: <https://store.arduino.cc/arduino-nano> (besucht am 02.02.2020) (siehe S. 8).
- [4] Tkotz, Klaus Burgmaier, Monika Eichler, Walter Kober, Karsten Bastian, Peter Bumiller, Horst Käppel, Thomas Spielvogel, Otto Schwarz, Jürgen Manderla, Jürgen Ziegler, Klaus Klee, Werner Winter, Ulrich. *Fachkunde Elektrotechnik*. Europa-Fachbuchreihe für elektrotechnische Berufe. Europa Lehrmittel, 2009. ISBN: 978-3-8085-3188-4 (siehe S. 9).
- [5] o.A. *Schrittmotor kurz erklärt*. o.J. URL: https://wiki.ntb.ch/infoportal/_media/hardware/sysp/bauteile/schrittmotor_kurz_erklaert_d.pdf (besucht am 02.02.2020) (siehe S. 9).
- [6] Erwin Böhmer, Dietmar Ehrhardt, Wolfgang Oberschelp. *Elemente der angewandten Elektronik. Kompendium für Ausbildung und Beruf*. Vieweg+Teubner, 2010. ISBN: 978-3-8348-0543-0 (siehe S. 10).