



Diplomarbeit Lormen

Kommunikationsgerät für taubblinde Menschen in Anwendung des Lorm-Alphabets

Jahrgang 2015/16

TGM Wien
Schule der Technik
Abendschule Elektronik
Wexstraße 19-23
1200 WIEN

BACHMEIER Alexander
EDER Michael
PLATTNER Martin

Betreuer: **Dr.DI Eckl** Gerald

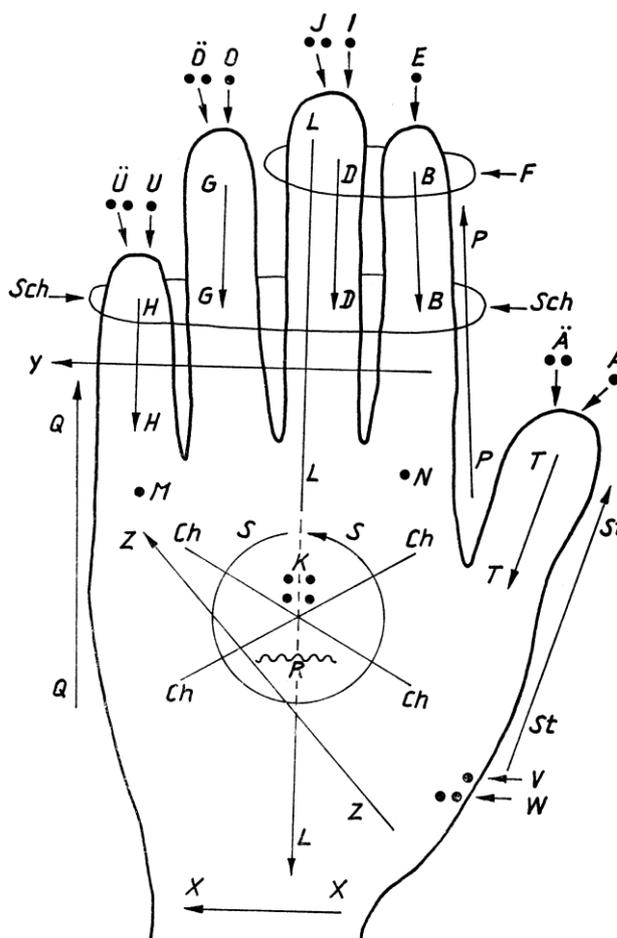


Abbildung 1: Deckblatt: Lorm-Alphabet (Taubblindendienst, 2016)

1 EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre eidesstattlich, dass ich die Arbeit selbstständig angefertigt, keine anderen Hilfsmittel als die angegebenen Hilfsmittel benutzt und alle aus ungedruckten Quellen, gedruckter Literatur oder aus dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte gemäß den Richtlinien wissenschaftlicher Arbeiten zitiert, durch Fußnoten gekennzeichnet bzw. mit genauer Quellenangabe kenntlich gemacht habe.



BACHMEIER Alexander



EDER Michael



PLATTNER Martin

Datum 16.04.2016

2 ABSTRACT (EN)

The target of this project was to develop a communication tool for deafblind people, the base is the Lorm'schen tactile alphabet. It was our wish to help deafblind people out of their isolation and so we chose this topic. The objective was to make their life easier and happier. Our invention allows deafblind people to communicate with people without that disease. Furthermore, enables the project none deafblind people who do not speak the Lorm-Alphabet to communicate with them. We took a special attention to design the application as easy and user friendly as possible. Another important aspect for us was that the system should be easily extendable and linkable. To sum it up our system includes the Lormen-hand and the Feedback-box. Due to the smart and thoughtful structure the user is able to use two different ways for the in/out-put nearly simultaneously. The input can be done via Lorm'schen tactile alphabet directly at the Lormen-hand or via the Android smartphone app. After that the input will be analyzed and sent to the Feedback-box. The received data is being processed by the Feedback-box and then shown on the display. If the with the vibration extended clove is plugged in, the clove will vibrate on the specific position according to the Lorm-Alphabet.

3 ABSTRACT (GER)

Wir haben uns das Ziel gesetzt, ein Kommunikationsgerät für taubblinde Menschen zu entwickeln, welches auf dem Lorm'schen Tastalphabet basiert. Wir haben uns für dieses Thema entschieden, da es uns ein Anliegen war, taubblinden Menschen aus ihrer Isolation zu helfen. Es sollte ihnen somit ein leichteres und glücklicheres Leben ermöglichen. Unsere Entwicklung ermöglicht taubblinden Menschen, mit jedem nicht Taubblinden zu kommunizieren. Ebenfalls dient es nicht Taubblinden, die das Lorm-Alphabet nicht beherrschen, sich Taubblinden mit zu teilen. Bei der Entwicklung und Gestaltung der Applikation legten wir großen Wert auf eine möglichst unkomplizierte und benutzerfreundliche Bedienung. Ebenfalls konzipierten wir das System so, dass es leicht vernetzbar und erweiterbar ist.

Zusammengefasst besteht unser System aus der Lormen-Hand und der Feedback-Box. Durch den geschickten und durchdachten Aufbau stehen dem Nutzer zwei Ein- und zwei Ausgabemöglichkeiten nahezu simultan zur Verfügung. Die Eingabe kann mittels Lorm'schem Tastalphabet an der Lormen-Hand, oder über eine Android Smartphone App erfolgen. Danach wird die Eingabe ausgewertet und an die Feedback-Box gesendet. Die empfangenen Daten werden nun in der Feedback-Box verarbeitet und auf das Display ausgegeben. Wenn der mit Vibratoren ausgestattete Handschuh angeschlossen ist, wird gleichzeitig auf diesem das Zeichen durch Vibrationen an den spezifischen Stellen des Lorm-Alphabets ausgegeben.

4 INHALTSVERZEICHNIS

1	Eidesstattliche Erklärung	2
2	Abstract (EN)	3
3	Abstract (GER)	3
4	Inhaltsverzeichnis	4
5	Abbildungsverzeichnis.....	7
6	Allgemeines	10
6.1	Was ist Lormen?	10
6.2	Taubblindheit	10
6.2.1	Was ist Taubblindheit?	10
6.2.2	Taubblindheit hat viele Ursachen.....	11
6.2.3	Lebenslange Hilfe nötig	11
6.2.4	Hörsehbehinderung und Taubblindheit sind Behinderungen eigener Art.....	11
6.2.5	CHARGE-Syndrom.....	12
6.2.6	USHER-Syndrom	13
6.3	Warum haben wir uns für dieses Projekt entschieden?	14
7	Pflichtenheft.....	15
7.1	Dokumenten Historie	15
7.2	Definitionen.....	15
7.2.1	Definition Lormen.....	15
7.2.2	Definition Kommunikation	15
7.3	Ziel	15
7.4	Kurzbeschreibung.....	16
7.5	Komponenten.....	16
7.6	Technische Daten der Komponenten.....	16
7.6.1	Eingabegerät.....	16
7.6.2	Ausgabegerät.....	18
7.6.3	Verbindungstechnologie	20
7.6.4	Optionale Erweiterungen	21
7.7	Termine	21
8	Eingabegerät Lormen-Hand	22
8.1	Vorstudien	23
8.2	Grundlagen	24
8.2.1	NPN Darlingtontransistor	24

8.2.2	Zweidrahttechnologie	25
8.2.3	CapacitivePin.lbr.....	25
8.2.4	CapacitiveSensor.lbr.....	26
8.3	Funktion.....	27
8.4	Schaltplan	28
8.4.1	Dimensionierung der Schaltung	31
8.5	PCB.....	35
8.6	Herstellung	36
8.6.1	Lormen - Hand	36
8.6.2	Gehäuse.....	51
8.7	Flussdiagramm	57
8.8	Programmcode.....	58
9	Eingabe Android-APP „Lormen“	74
9.1	Flussdiagramm	75
10	Ausgabegerät Feedback – Box.....	76
10.1	Vorstudien	77
10.1.1	Optische Ausgabe.....	77
10.1.2	Haptische Ausgabe	78
10.2	Grundlagen	80
10.2.1	Demultiplexer	80
10.2.2	Treiberstufe	80
10.2.3	Einstellbarer Spannungsregler	81
10.2.4	Ansteuerung mehrerer Vibratoren.....	81
10.3	Funktion.....	82
10.4	Schaltplan	83
10.5	PCB.....	87
10.6	Herstellung	88
10.7	Flussdiagramm	88
10.8	Programmcode.....	89
11	Verbindungstechnik.....	116
11.1	Radiofrequenz – Modul.....	116
11.1.1	Spezifikation	117
11.1.2	Verbindungsschnittstelle.....	117
11.2	Bluetooth – Modul	118
11.2.1	Spezifikation	119

11.2.2	Verbindungs-Schnittstelle	119
12	Projektmanagement.....	120
12.1	Zuständigkeiten der Projektmitglieder.....	120
12.2	Meilensteine.....	120
12.3	Breakeven.....	121
12.3.1	Kosten der GmbH/Monat (Fixe Kosten)	121
12.3.2	Materialkosten (Variable Kosten).....	122
12.3.3	Berechnung des Verkaufspreises	122
12.4	Timeline	123
13	Einsatzmöglichkeiten.....	131
13.1	Die Applikation als Lernsystem	131
13.1.1	Lernsystem – Mögliche Erweiterungen.....	132
13.2	Die Applikation als Sprachrohr – ein wichtiger Schritt zur Selbstständigkeit	132
13.2.1	„Schreiben“.....	132
13.2.2	„Lesen“	132
13.2.3	„Schreiben“ und „Lesen“ – Mögliche Erweiterungen	133
14	Bedienungsanleitung.....	134
14.1	Bedienung – Eingabegerät Lormen-Hand	136
14.2	Bedienung –Eingabe via Bluetooth	137
14.3	Bedienung Ausgabegerät–Feedbackbox.....	138
14.4	Bedienung der haptischen Ausgabe - Handschuh.....	139
15	Literaturverzeichnis.....	140
16	Anhang.....	141
16.1	Hardwarespezifikationen	141
16.2	Datenblätter	141
16.3	Stücklisten	142
16.3.1	Stückliste Lormen-Hand	142
16.3.2	Stückliste Feedback-Box	146
16.3.3	Stückliste Haptische Ausgabe.....	147

5 ABBILDUNGSVERZEICHNIS

Abbildung 1: Deckblatt: Lorm-Alphabet (Taubblindendienst, 2016).....	1
Abbildung 2: Eingabegerät Lormen-Hand	22
Abbildung 3: Skizze Sensorflächen – im Bezug auf das Lorm-Alphabet	23
Abbildung4: Schaltzeichen NPN Darlingtontransistor (Elektronik Kompendium, 2016)	24
Abbildung 5: Ersatzschaltbild NPN Darlington Transistor Abbildung (Wikipedia, 2016a).....	24
Abbildung 6: Zweidrahttechnologie Schaltungsaufbau Pull Up / Pull Down Widerstand (Tuelectronica, 2012).....	25
Abbildung 7: Schaltungsaufbau CapacitivePin.lbr.....	26
Abbildung 8: Schaltungsaufbau CapacitiveSensor.lbr	26
Abbildung 9: Technische Übersicht Eingabegerät: Lormen-Hand	27
Abbildung 10: Schaltplan Lormen-Hand Gesamtansicht	28
Abbildung 11: Schaltplan Lormen-Hand Detailansicht Mikrokontroller.....	29
Abbildung 12: Schaltplan Lormen-Hand Detailansicht Sensorwiderstände 1 von 2	29
Abbildung 13: Schaltplan Lormen-Hand Detailansicht Sensorwiderstände 2 von 2	29
Abbildung 14: Skizze Sensorflächen – nummeriert.....	30
Abbildung 15: Ladekurve eines Kondensators (WeTec's Technikseite, 2010).....	31
Abbildung 16: Entladekurve eines Kondensators (WeTec's Technikseite, 2010).....	32
Abbildung 17: Ersatzschaltbild Kondensator im Schaltungsaufbau mit niedriger Kapazität	33
Abbildung 18: Ersatzschaltbild Kondensator im Schaltungsaufbau mit größerer Kapazität	33
Abbildung 19: Ersatzschaltbild Kondensator im Schaltungsaufbau mit maximaler Kapazität bei Normalgebrauch des Eingabegerätes	33
Abbildung 20: Kunststoffhand mit Sensoren	34
Abbildung 21: Leiterplatte für Eingabegerät Lormen-Hand, Bauteilansicht (THT, Bauteile oben).....	35
Abbildung 22: Prototyp I.....	36
Abbildung 23: Prototyp II Handfläche	37
Abbildung 24: Prototyp II Handrücken.....	37
Abbildung 25: Prototyp II Alginatabdruck.....	38
Abbildung 26: Prototyp II Alginat verdrahtet	39
Abbildung 27: Prototyp II Spezialform.....	40
Abbildung 28: Prototyp II - Rohling Handfläche.....	41
Abbildung 29: Prototyp II - Rohling Handrücken.....	41
Abbildung 30: Prototyp II - Mechanische Bearbeitung.....	42
Abbildung 31: Prototyp II - Vergleich mit Modell.....	43
Abbildung 32: Prototyp III - incl. Gehäuse	44
Abbildung 33: Prototyp III - Alginatabdruck.....	46
Abbildung 34: Prototyp III - Sockel.....	46

Abbildung 35: Prototyp III - Epoxidharz Gießverfahren	47
Abbildung 36: Prototyp III - Rohling.....	47
Abbildung 37: Prototyp III – Nachmodellierung	48
Abbildung 38: Prototyp III - nach dem Lackieren	49
Abbildung 39: Prototyp III - fertig.....	50
Abbildung 40: Prototyp III - Gelenk	50
Abbildung 41: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Gesamtansicht	52
Abbildung 42: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Draufsicht (um 90° nach rechts gedreht).....	53
Abbildung 43: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Ansicht von rechts	54
Abbildung 44: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Ansicht von links	55
Abbildung 45: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Ansicht von vorne (um 90° nach rechts gedreht).....	56
Abbildung 46: Flussdiagramm Lormen-Hand.....	57
Abbildung 47: Startbild Android App.....	74
Abbildung 48: Tastatur Android App.....	74
Abbildung 49: Verbindung Android App.....	74
Abbildung 50: Flussdiagramm Android APP "Lormen"	75
Abbildung 51: Feedback - Box offen	76
Abbildung 52: Optische Ausgabe Display	77
Abbildung 53: Haptische Ausgabe Handschuh	78
Abbildung 54: Anordnung der Vibratoren auf der Hand	79
Abbildung 55: Feedback-Box Technische Übersicht.....	82
Abbildung 56: Schaltplan Feedback-Box – Gesamtansicht	83
Abbildung 57: Schaltplan Feedback-Box – Detailansicht Spannungsversorgung	83
Abbildung 58: Schaltplan Feedback-Box – Detailansicht Mikrocontroller, RF- Modul und Bluetooth-Modul	84
Abbildung 59: Schaltplan Feedback-Box – Detailansicht Demultiplexer.....	85
Abbildung 60: Schaltplan Feedback-Box – Detailansicht Treiberstufen und Verbindungen für Handschuh.....	86
Abbildung 61: Leiterplatte Feedback-Box, Bauteilansicht (THT, Bauteile oben).....	87
Abbildung 62: Fussdiagramm Feedback-Box	88
Abbildung 63: Radiofrequenz-Modul 2	116
Abbildung 64: Radiofrequenz-Modul 1	116
Abbildung 65: SPI Aufbau (Wikipedia, 2016b).....	117
Abbildung 66: Bluetooth-Modul	118
Abbildung 67: HC-06 Bluetooth-Modul (Reutlingen-university, 2016).....	119
Abbildung 68: Gesamtübersicht.....	134
Abbildung 69: Blockschaltbild Eingabemöglichkeiten.....	135
Abbildung 70: Blockschaltbild Ausgabemöglichkeiten.....	135

Abbildung 71: Display mit empfangenem Text	136
Abbildung 72: Eingabe mittels Android App.....	137
Abbildung 73: Feedback - Box	138
Abbildung 74: Haptische Ausgabe - Handschuh.....	139

6 ALLGEMEINES

6.1 WAS IST LORMEN?

Stellen Sie sich vor, Sie verlieren von einem Tag auf den anderen Ihr Gehör und sind nach zwei Tagen vollständig taub. Gleichzeitig lässt Ihre Sehfähigkeit ganz schnell nach - bis hin zur Blindheit. Sie werden also taubblind! Wie ertragen Sie ein solches Schicksal, wie reagieren Sie darauf? Ein Beispiel für die Bewältigung eines solchen Schicksals mag die nachstehende Kurzbiografie sein!

Heinrich Landesmann wurde 1821 in Mähren geboren und wuchs in einem wohlhabenden Elternhaus in Wien sehr behütet auf. Seine schwache Gesundheit machte seinen Eltern von Geburt an Sorgen. Dennoch konnte er zunächst eine normale Schule besuchen, später wurde er durch Privatunterricht besonders gefördert. Sein Elternhaus war offen für Künstler und Schriftsteller - Kontakte, die den jungen Heinrich stark beeinflussten.

Es zeigte sich, dass er hochmusikalisch war. Folglich wurde er von den besten Lehrern der damaligen Zeit unterrichtet; die Lehrer waren begeistert von seinem musikalischen Verständnis. In dieser für den 16jährigen Jungen so kreativen Phase, die schon seinen weiteren Lebensweg vorzeichnen schien, verlor er innerhalb von zwei Tagen sein Gehör. Er war plötzlich ohne Vorwarnung und ohne äußeren Einfluss taub geworden. Welch ein Schlag für einen jungen Menschen, dessen Fähigkeiten ausgerechnet in der Musik lagen! Dazu kam, dass er gleichzeitig erkennen musste, dass seine Sehfähigkeit deutlich nachließ.

Es blieb nicht aus, dass er in eine tiefe Resignation verfiel - ja sogar sehr pessimistisch in die Zukunft blickte. Die Resignation dauerte nicht allzu lange, und er fand die Kraft, sich seiner Fähigkeiten zu besinnen: Er wurde Schriftsteller und Philosoph, der sich in sehr vielfältiger Form äußerte - sei es in Romanform, Lyrik, Aufsätzen in der Presse, Briefen usw. In seinen Schriften prangerte er die strenge Pressezensur in Österreich an, die von Metternich veranlasst worden war. Um sich und vor allem seine Familie zu schützen, schrieb er unter dem Pseudonym Hieronymus Lorm. Mit 35 Jahren ging er eine Ehe ein, die zeitlebens sehr glücklich gewesen ist, und der drei Kinder entstammten. Seine Frau unterstützte ihn in jeder Weise. Später bekam sie sehr aktive Hilfe durch eine der Töchter, so dass Lorm in seiner Schaffenskraft kaum noch behindert schien. Zu dieser Zeit erfand er ein Tastalphabet aus Punkten und Strichen, mit dessen Hilfe seine Mitmenschen mit ihm kommunizieren konnten. Es war einfach erlernbar, so dass seine Familie, Freunde und auch Schriftsteller sich gut mit ihm verständigen konnten. Dieses Tastalphabet bezeichnet man als Lormen. (anderssehen, 2016)

6.2 TAUBBLINDHEIT

6.2.1 WAS IST TAUBBLINDHEIT?

Wenn Gehörlosigkeit und Blindheit bei einem Menschen zusammen auftreten, dann spricht man von Taubblindheit. Dabei handelt es sich um eine komplexe Sinnesbehinderung. Taubblindheit ist nicht nur dann gegeben, wenn Hör- und Sehvermögen vollständig ausfallen, sondern auch bei häufig auftretenden Kombinationen von unterschiedlich starken Hör- und Sehschädigungen. (taubblindenwerk.de B, 2016)

6.2.2 TAUBBLINDHEIT HAT VIELE URSACHEN

Wenn jemand also gleichzeitig in seiner Fähigkeit zu sehen und zu hören beeinträchtigt ist, so ist er – je nach Umfang der Beeinträchtigung – höresehbehindert bzw. taubblind. Es gibt über 70 verschiedene Ursachen für Taubblindheit. Das CHARGE-Syndrom und das USHER-Syndrom können beispielsweise eine dieser Ursachen sein. Grundsätzlich gibt es die angeborene und die erworbene Taubblindheit. Der Zeitpunkt des Eintritts der Sinnesbeeinträchtigung hat maßgeblichen Einfluss auf die Entwicklung des Menschen und auf seine Lebensperspektive. So spielt es beispielsweise eine Rolle, ob die Taubblindheit vor oder nach dem Spracherwerb eingetreten ist oder ob zuerst die Gehörlosigkeit und dann die Blindheit eingetreten ist, oder umgekehrt. Je nach dem spielen andere Hilfsmittel bei der Kommunikation und Bewältigung des Alltags eine Rolle. (taubblindenwerk.de B, 2016)

6.2.3 LEBENSLANGE HILFE NÖTIG

Hörsehbehinderte und taubblinde Menschen sind lebenslang auf Hilfe und auf vertraute Begleiter angewiesen, um ihre Isolation zu verringern und Bezüge zur personalen und sachlichen Umwelt erleben und intensivieren zu können. In Deutschland wird von schätzungsweise 10.000 hörsehbehinderten oder taubblinden Menschen ausgegangen. Es gibt etwa 1.000 Personen bis zu 20 Jahren mit angeborener Taubblindheit. (taubblindenwerk.de B, 2016)

6.2.4 HÖRSEHBEHINDERUNG UND TAUBBLINDHEIT SIND BEHINDERUNGEN EIGENER ART

Die Betroffenen sind nicht als mehrfachbehinderte hör- oder sehbeeinträchtigte Menschen einzustufen, da die Ausfälle des einen Sinnes nicht oder nur mangelhaft durch den anderen kompensiert werden. Hörsehbehinderung und Taubblindheit können mit jeder anderen Behinderung oder Erkrankung auftreten. (taubblindenwerk.de B, 2016)

CHARGE-Syndrom als Ursache von Taubblindheit und Hörsehbehinderung

Es gibt viele verschiedene Ursachen für Hörstörungen, Sehstörungen und Taubblindheit – das CHARGE-Syndrom ist eine davon. Beim CHARGE-Syndrom handelt es sich um eine seltene, genetisch bedingte Störung, die sich auf verschiedene Körperteile auswirken kann. Die Anfangsbuchstaben der häufigsten Symptome bilden die Abkürzung „CHARGE“:

C – Kolobom des Auges: Sehfehler, der Sehstörungen zur Folge haben kann

H – Herzfehler

A – Atresie der Choanen: ein- oder beidseitig blockierter oder sehr enger Nasengang

R – Retardiertes Längenwachstum und Entwicklungsverzögerung

G – Geschlechtsorgan-Anomalien

E – Ohrfehlbildungen, die Hörverlust bedingen können

Das Auftreten dieser Symptome ist bei jeder betroffenen Person unterschiedlich. Nicht alle Symptome treten bei jedem auf und auch nicht in derselben Ausprägung. Allerdings gilt das CHARGE-Syndrom heutzutage als Hauptursache für angeborene Hörseherschädigungen, die – unabhängig von den zusätzlichen Symptomen – Behinderungen ganz eigener Art darstellen. (taubblindenwerk.de A, 2016)

Das Usher-Syndrom ist eine Kombination aus Hörschädigung in Verbindung mit einer Degeneration der Netzhaut (Retinopathiapigmentosa) und wurde 1858 von Albrecht von Graefe aus Berlin, dem Begründer der modernen Augenheilkunde beschrieben. Charles Howard Usher, ein englischer Augenarzt, erkannte 1914 die rezessive Vererbung dieses Syndroms.

Das Usher-Syndrom wird in drei Typen klassifiziert:

Usher-Syndrom Typ I: angeborene Taubheit mit schweren Gleichgewichtsstörungen und der in überwiegend früher Kindheit eintretenden Augenerkrankung Retinopathiapigmentosa (RP), auch oft Retinitis pigmentosa genannt.

Usher-Syndrom Typ II: angeborene Schwerhörigkeit, die überwiegend stabil bleibt. Der Gleichgewichtssinn ist normal. Die Augenerkrankung Retinopathiapigmentosa (RP) hat einen variablen Beginn.

Usher-Syndrom Typ III: normal hörend geboren, fortschreitender Hörverlust im frühen Erwachsenenalter bis hin zur Taubheit, eine Störung des Gleichgewichtssinns ist nicht bekannt. Die Augenerkrankung RP beginnt häufig während der Pubertät. Dieser Typ des Usher-Syndroms tritt sehr selten auf.

Die Hörschädigung beruht auf einer angeborenen cochleären Innenohrschädigung. Je nach Grad der Hörschädigung wird diese häufig erst spät erkannt und dann fälschlicherweise auf eine Kinderkrankheit zurückgeführt.

Die Symptome der Augenerkrankung Retinopathiapigmentosa (RP) sind:

Gestörtes Dämmerungssehen/Nachtblindheit

Blendungsempfindlichkeit

Farbsehstörungen

Herabgesetztes Kontrastsehen

Erhebliche Gesichtsfeldeinschränkung (Tunnelblick)

Nachlassen der Sehschärfe

Der Krankheitsprozess verläuft von der Netzhautperipherie her konzentrisch nach innen bis nur noch ein röhrenartiges Gesichtsfeld besteht. Ursache ist das Absterben der Rezeptoren der Netzhaut.

Der Krankheitsverlauf ist sehr unterschiedlich, so dass man keine Prognose über den Zeitraum der Sehverschlechterung stellen kann, selbst Untersuchungen bei eineiigen Zwillingen haben sehr unterschiedliche Krankheitsverläufe gezeigt. Die Gefahr der völligen Erblindung ist gegeben, muss aber nicht zwangsläufig sein, sie ist abhängig von der Intensität des Verlaufs.

Oft geht mit der Augenerkrankung RP auch eine Linsentrübung (Grauer Star) einher, die eine zusätzliche Blendung und eine Herabsetzung der Sehschärfe verursacht. (taubblindenwerk.de C, 2016)

6.3 WARUM HABEN WIR UNS FÜR DIESES PROJEKT ENTSCHIEDEN?

Wir haben uns für dieses Projekt entschieden um auf das schwere Schicksal dieser komplexen Sinnesbehinderung aufmerksam zu machen. Da nur ca. 1400 Menschen in Österreich mit Taubblindheit leben müssen gibt es leider viel zu wenig Aufmerksamkeit für dieses Thema.

Deswegen haben wir uns zur Aufgabe gemacht mit einfachen Mitteln ein Kommunikationsgerät zu entwickeln. Welches als Lernhilfe dient um auch allein, mit Hilfe eines LCD Displays, seine Lorm Fähigkeiten zu überprüfen und zu vertiefen.

7 PFLICHTENHEFT

7.1 DOKUMENTEN HISTORIE

Version	Änderungsdatum	Änderung
V1.0	14.09.2015	Grundlage
V1.1	21.09.2015	Formatierung
V1.21	28.09.2015	Technische Daten
V1.22	05.11.2015	Technische Daten
V1.30	14.12.2015	Formatierung

Verordneter, spätester Abgabetermin ist der **24.05.2016**

7.2 DEFINITIONEN

7.2.1 DEFINITION LORMEN

Das Lormen ist ein in die Hand „geschriebenes“ Alphabet, bei dem bestimmte Zeichen an bestimmten Stellen der Hand verschiedene Einzelbuchstaben des Alphabets repräsentieren (Deutsches Taubblindenwerk GmbH, 2015)

7.2.2 DEFINITION KOMMUNIKATION

Kommunikation ist der Austausch oder die Übertragung von Informationen (Wikipedia, 2015)

7.3 ZIEL

Ziel ist die Entwicklung einer Kommunikationshilfe für taubblinde Menschen, die es ihnen erleichtert, mit sehenden Menschen in Kontakt zu treten. Sie soll jedoch ebenfalls sehenden Personen die Möglichkeit bieten, das Lorm-Alphabet für sich zu entdecken, zu lernen und üben zu können. Weiters ist es uns ein großes Anliegen, auf das Thema Lormen und Taubblindheit aufmerksam zu machen.

7.4 KURZBESCHREIBUNG

Die Lormen-Hand ermöglicht taubblinden Menschen mit anderen Personen, welche das Lorm-Alphabet nicht kennen, zu kommunizieren:

Der taubblinde Mensch schreibt mit dem/n Finger/n Lormzeichen in die Lormen-Hand.

Diese Informationen werden übersetzt und von einer Feedback-Box optisch (optional auch haptisch) ausgegeben.

7.5 KOMPONENTEN

Das System besteht aus folgenden Komponenten:

- Eingabegerät:
 - Lormen-Hand – dient zur Eingabe der Lorm-Buchstaben
- Ausgabegerät:
 - Feedback-Box – dient zur Ausgabe der übermittelten Informationen
 - (Optional) Haptisches Ausgabegerät – dient der Übertragung von Lorm-Buchstaben auf die Hand eines Benutzers.
- Verbindungen:
 - Verbindung der beiden Komponenten (Draht oder Funk)

7.6 TECHNISCHE DATEN DER KOMPONENTEN

7.6.1 EINGABEGERÄT

Die Lormen-Hand ist mit Sensoren ausgestattet, welche Berührungen detektieren.

Die Sensoren befinden sich an charakteristischen Stellen der aus Kunststoff gefertigten Handnachbildung. Die Sensoren werden zyklisch abgetastet und die daraus gewonnenen Informationen mittels Mikrocontroller, in einem Gehäuse unterhalb der Kunststoffhand, ausgewertet und an die Feedback-Box übertragen.

Die Lormen-Hand wird je nach verwendeter Verbindungstechnologie mit einem handelsüblichen Steckernetzteil versorgt, welches an der Feedback-Box oder direkt an der Lormen-Hand angeschlossen wird.

Die zu verwendenden Technologien gilt es noch durch Vorstudien zu ermitteln, um ein bestmögliches Ergebnis zu erreichen.

Optional werden benötigte Verbindungsschnittstellen zur Programmierung am Gehäuse verbaut. Ebenso wird durch Vorstudien ermittelt, ob sich noch weitere Schnittstellen zur Lormen-Hand und/oder Benutzer als sinnvoll erweisen können. Diese werden dann auch an oder in der Lormen-Hand verbaut.

7.6.1.1 MECHANISCHE EIGENSCHAFTEN

7.6.1.1.1 MATERIAL

Hand aus Epoxidharz verbunden mit Gehäuse aus Plexiglas.

7.6.1.1.2 GRÖSSE

Die Lormen-Hand ist eine 1:1 Abbildung einer menschlichen Hand auf einem Sockel (Gehäuse) welcher groß genug ist, die Bauteile unterzubringen und die Hand dabei ausreichend zu stabilisieren.

7.6.1.1.3 GEWICHT

< 4kg.

7.6.1.1.4 FARBE

Sehr vielseitig.

7.6.1.1.5 UMWELTBEDINGUNGEN

Innenbereich

7.6.1.1.6 EINSATZTEMPERATUR

+5°C - +40°C Umgebungstemperatur im Betrieb

+5°C - +40°C Lagerung

7.6.1.2 ELEKTRISCHE EIGENSCHAFTEN

7.6.1.2.1 SPANNUNGSVERSORGUNG

Die Lormen-Hand wird je nach verwendeter Verbindungstechnologie mit einem handelsüblichen Steckernetzteil versorgt, welches auf der Feedback-Box oder direkt an der Lormen-Hand angeschlossen wird.

7.6.1.2.2 ESD – FESTIGKEIT

Der Prototyp erfüllt keine ESD-Normen – ESD Prüfung lt. EN61000-4-2:2009 nicht zugelassen.

Angestrebt wird Prüfspannung 6kV.

7.6.1.2.3 EMV – AKTIV

Die Grenzwerte der aktiven EMV sind laut unten angeführten Normen zu bewerten – der Prototyp strebt Erfüllung dieser Grenzwerte an. Nachweis über die Einhaltung der jeweiligen Grenzkurven ist für den Prototyp jedoch nicht erforderlich.

Norm: EN61000-6-3:2007 + A1:2011 Störaussendung für Wohnbereich, Geschäfts- und Gewerbebereiche sowie Kleinbetriebe

Leitungsgebunden: EN55016-2-1:2004: Grenzwertkurve A

Störstrahlung: EN 55016-2-3:2006: Grenzwertkurve A

7.6.1.2.4 EMV – PASSIV

Die Hand ist nicht geschirmt. Störfeldänderungen, welche nach der Inbetriebnahme der Hand auftreten, können Auswirkungen auf das Ergebnis haben.

Die entsprechenden Grenzwerte sind in folgenden Normen definiert, werden aber vom Prototyp nicht nachweislich eingehalten:

EN 61000-4-3:2006 Störfestigkeit gegen hochfrequente elektromagnetische Felder

EN 61000-4-4:2013 Störfestigkeit gegen schnelle transiente elektrische Störgrößen (Burst)

EN 61000-4-5 Ed.3.0:2014 Störfestigkeit gegen Stoßspannungen (Surge)

EN 61000-4-6:2009 Störfestigkeit gegen leitungsgeführte Störgrößen, induziert durch hochfrequente Felder

EN 61000-4-8:1993 + A1:2001 Störfestigkeit gegen Magnetfelder mit energietechnischen Frequenzen

EN 61000-4-11:2004 Störfestigkeit gegen Spannungseinbrüche, Kurzzeitunterbrechungen und Spannungsschwankungen

EN 61000-4-20:2003 Störfestigkeit gegen TEM Felder

7.6.1.2.5 ÜBERLASTSCHUTZ

Die Lormen-Hand verfügt über keinen zusätzlichen Überlastschutz.

7.6.1.3 SCHNITTSTELLEN

Anschluss für Spannungsversorgung und Programmierschnittstelle am Gehäuse.

Optional: Radiofrequenzschnittstelle, Display , Funktionsmasse (Schukostecker mit Erdanschluss) und Kabel zur Datenübertragung.

7.6.2 AUSGABEGERÄT

Die Feedback-Box besteht aus einem Gehäuse mit Mikrocontroller. An einer praktischen und gut abzulesenden Stelle des Gehäuses befindet sich ein Display. Das Display gibt die übermittelten und verarbeiteten Informationen aus.

Das Ausgabegerät wird über ein handelsübliches Steckernetzteil mit Energie versorgt.

7.6.2.1 MECHANISCHE EIGENSCHAFTEN

7.6.2.1.1 MATERIAL

Kunststoffgehäuse

7.6.2.1.2 GRÖÖE

248 / 248 / 123 mm (L/B/H) +/- 20% Toleranz

7.6.2.1.3 GEWICHT

<2kg

7.6.2.1.4 FARBE

Mehrfärbig

7.6.2.1.5 UMWELTBEDINGUNGEN

Innenbereich

7.6.2.1.6 EINSATZTEMPERATUR

+5°C - +40°C Umgebungstemperatur im Betrieb

+5°C - +40°C Lagerung

7.6.2.2 ELEKTRISCHE EIGENSCHAFTEN

7.6.2.2.1 SPANNUNGSVERSORGUNG

Steckernetzteil – Voltcraft USPS - 1500

Eingang: 100 – 240 V/AC, 60/50 Hz 350mA

Ausgang: Stufenweise einstellbar 3-12 V/DC Out

Ausgangsstrom: 1500 mA max.

7.6.2.2.2 ESD – FESTIGKEIT

Der Prototyp erfüllt keine ESD-Normen – ESD Prüfung lt. EN61000-4-2:2009 nicht zugelassen.

Angestrebt wird Prüfspannung 6kV.

7.6.2.2.3 EMV – AKTIV

Die Grenzwerte der aktiven EMV sind laut unten angeführten Normen zu bewerten – der Prototyp strebt Erfüllung dieser Grenzwerte an. Nachweis über die Einhaltung der jeweiligen Grenzkurven ist für den Prototyp jedoch nicht erforderlich.

Norm: EN61000-6-3:2007 + A1:2011 Störaussendung für Wohnbereich, Geschäfts- und Gewerbebereiche sowie Kleinbetriebe

Leitungsgebunden: EN55016-2-1:2004: Grenzwertkurve A

Störstrahlung: EN 55016-2-3:2006: Grenzwertkurve A

7.6.2.2.4 EMV – PASSIV

Die Hand ist nicht geschirmt. Störfeldänderungen, welche nach der Inbetriebnahme der Hand auftreten, können Auswirkungen auf das Ergebnis haben.

Die entsprechenden Grenzwerte sind in folgenden Normen definiert, werden aber vom Prototyp nicht nachweislich eingehalten:

EN 61000-4-3:2006 Störfestigkeit gegen hochfrequente elektromagnetische Felder

EN 61000-4-4:2013 Störfestigkeit gegen schnelle transiente elektrische Störgrößen (Burst)

EN 61000-4-5 Ed.3.0:2014 Störfestigkeit gegen Stoßspannungen (Surge)

EN 61000-4-6:2009 Störfestigkeit gegen leitungsgeführte Störgrößen, induziert durch hochfrequente Felder

EN 61000-4-8:1993 + A1:2001 Störfestigkeit gegen Magnetfelder mit energietechnischer Frequenz

EN 61000-4-11:2004 Störfestigkeit gegen Spannungseinbrüche, Kurzzeitunterbrechungen und Spannungsschwankungen

EN 61000-4-20:2003 Störfestigkeit gegen TEM Felder

7.6.2.2.5 ÜBERLASTSCHUTZ

Die Feedbackbox verfügt über keinen zusätzlichen Überlastschutz.

7.6.2.3 SCHNITTSTELLEN

Anschluss für das Netzteil am Gehäuse.

Optional: Radiofrequenzschnittstelle, Bluetooth, Touch-Display, Anschluss/Anschlüsse für haptische/s Ausgabegerät/e und Kabel zur Datenübertragung.

7.6.3 VERBINDUNGSTECHNOLOGIE

Die Verbindung der Lormen-Hand zur Feedback-Box erfolgt mittels Funk oder Kabel. Welche der beiden Übertragungstechnologien sich für das Projekt am besten eignet, wird noch durch Vorstudien und Versuchsaufbauten ermittelt.

Falls sich eine Übertragung mittels Kabel als endgültige Lösung anbietet, wäre dadurch sowohl eine Kommunikation als auch die Energieversorgung angedacht.

Bei der Verwendung einer Funkverbindung muss die Lormen-Hand mit einem eigenen Steckernetzteil ausgestattet werden.

7.6.3.1 GALVANISCHE VERBINDUNG

7.6.3.1.1 MECHANISCHE EIGENSCHAFTEN

Handelsübliches Kabel 4x 1mm²

7.6.3.1.2 ELEKTRISCHE EIGENSCHAFTEN

Steckernetzteil – Voltcraft USPS - 1500

Eingang: 100 – 240 V/AC, 60/50 Hz 350mA

Ausgang:Stufenweise einstellbar 3-12 V/DC Out

Ausgangsstrom: 1500 mA max.

7.6.3.2 FUNKVERBINDUNG (OPTIONAL)

7.6.3.2.1 MECHANISCHE EIGENSCHAFTEN

Siehe Datenblatt: NRF24L01+PA+LNA

7.6.3.2.2 ELEKTRISCHE EIGENSCHAFTEN

Siehe Datenblatt: NRF24L01+PA+LNA

7.6.4 OPTIONALE ERWEITERUNGEN

7.6.4.1 HAPTISCHES AUSGABEBEGERÄT

Bei der haptischen Ausgabe gilt es durch Vorstudien zu ermitteln, welche Methode sich am besten dafür eignet, haptisch wahrgenommen zu werden und wie man diese am besten technisch umsetzen kann.

Es ist dabei sehr wichtig, die Abfolge der verschiedenen Zeichen des Lorm-Alphabets gut unterscheiden zu können.

7.6.4.2 MEHREMPFÄNGERBETRIEB

Eine Erweiterung der optischen Ausgabe durch haptische Signale mittels Vibrationsmotoren ist angedacht. Deren Implementierung in das System hängt von den Ergebnissen der Vorstudie ab. Es gilt hierbei zu ermitteln, ob die Anbringung von Vibrationsmotoren an einem Element (z.B.: Handschuh) die Möglichkeit bietet, Lorm-Buchstaben zu erkennen und diese voneinander zu unterscheiden. Diese Erweiterung würde eine Kommunikation zwischen zwei taubblinden Menschen, oder Menschen die das Lorm-Alphabet beherrschen, durch indirekte Berührung ermöglichen. Diese Funktion könnte sich als nützlicher Anwendungsgegenstand für immunschwache oder infektiöse Personen erweisen. Menschen, die diese Sprache erlernen möchten, könnten dies ebenfalls nutzen um das „Lesen“ des Lorm-Alphabets zu üben.

Unter anderem wären eine oder mehrere Verbindungsstellen für einen zweiten oder weitere haptische Ausgabegeräte angedacht. Diese Funktion ermöglicht es, zwei oder mehreren taubblinden Menschen zeitgleich Informationen zu übergeben. Dies könnte beispielsweise in einer betreuten Werkstatt Anklang finden, da es den Betreuern ermöglicht, zu mehreren taubblinden Klienten gleichzeitig zu „sprechen“.

7.6.4.3 ANDROID APP

Eine Erweiterung der Eingabe ist ebenfalls angedacht. Sofern es zeitlich machbar ist und sich eine haptische Ausgabe als sinnvoll erweist, werden wir eine AndroidApp programmieren, damit Menschen, die das Lorm-Alphabet nicht beherrschen, ebenfalls mit der Feedback-Box kommunizieren können.

7.7 TERMINE

Der Liefertermin für das fertiggestellte, funktionsfähige System laut Pflichtenheft ist der 23.5.2016.



Abbildung 2: Eingabegerät Lormen-Hand

Das Eingabegerät besteht aus einer Hand aus Epoxidharz mit kapazitiven Sensoren und einem Gehäuse, welches sowohl als Sockel für die Hand als auch als Schutz für die Schaltung dient. Die Schaltung besteht aus einem Mikrocontroller, auf den ein PCB gesteckt ist. Das Eingabegerät wird mit den Fingern bedient, indem man das Lorm-Alphabet anwendet. Der Mikrocontroller verarbeitet diese Bewegungsabläufe und übersetzt sie. Anschließend wird die Information an das Ausgabegerät gesendet. Die Planung und Herstellung des Eingabegerätes war ein kreativer und wachsender Prozess, bei dem wir unser Wissen vertiefen konnten.

8.1 VORSTUDIEN

Für die Auswahl der Hardware mussten wir einige Vorstudien leisten, weil es ein Eingabegerät in dieser Form noch nicht gab. In erster Linie war es wichtig eine einfache und kostengünstige Möglichkeit der Auswertung zu finden. Hierfür experimentierten wir unter anderem mit Darlingtontransistorschaltungen, Zweidrahttechnologien und verschiedensten Libraries für den Arduino. Nach der Entscheidung über die Sensortechnologie machten wir unzählige Skizzen, um die Positionierung und Anzahl der Sensoren zu ermitteln.

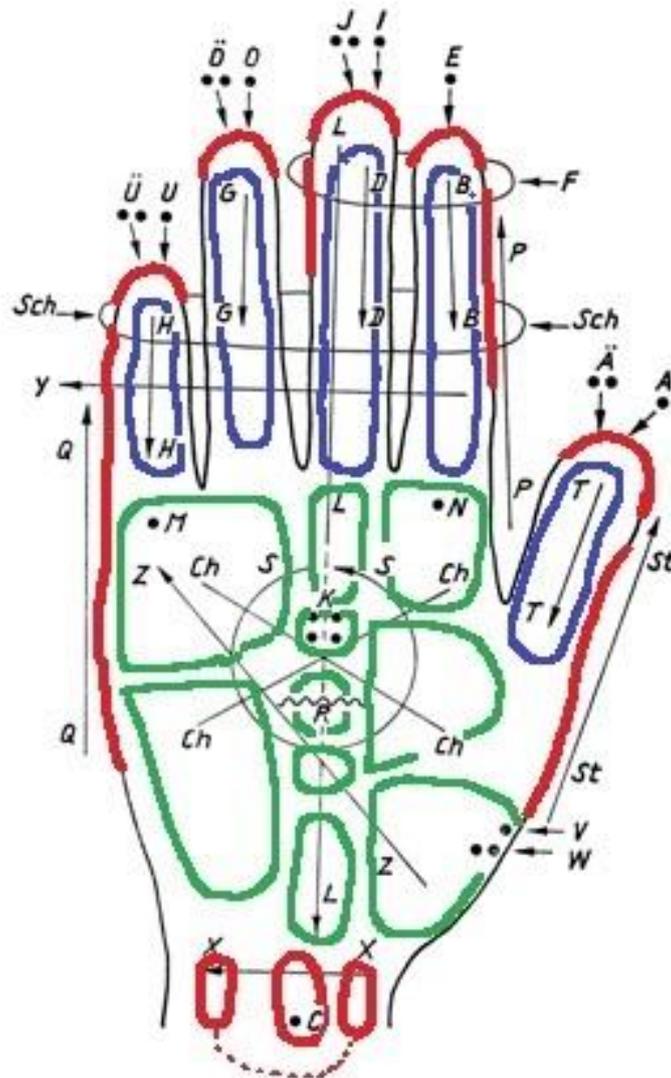


Abbildung 3: Skizze Sensorflächen – im Bezug auf das Lorm-Alphabet

Hinweis: Bei der Abbildung handelt es sich um eine Skizze der Sensorpositionierung. Die Felder am Handgelenk, welche mit Punkten verbunden sind, erhalten eine Drahtverbindung.

Bei der Auswahl der Sensoren setzten wir uns das Ziel, eine kostengünstige Lösung zu finden, die möglichst wenige Bauteile zur Umsetzung benötigt und dennoch zuverlässig Berührungen an einem oder mehreren Sensoren zeitgleich detektiert – möglichst unabhängig von der Hautbeschaffenheit.

8.2.1 NPN DARLINGTONTRANSISTOR

Ein Darlingtontransistor ist ein Bauteil, der aus zwei NPN Bipolartransistoren besteht. Die Verstärkung der beiden multipliziert sich durch den geschickten Schaltungsaufbau. Durch diese hohe Verstärkung ist es möglich, den Transistor durch Berühren der Basis leitend zu machen.

Nach einigen Versuchen haben wir uns jedoch gegen diese Technologie entschieden. Hierfür gibt es mehrere Gründe. Zum Einen gab es im Versuchsaufbau am Steckbrett bei einer gleichzeitigen Berührung zweier Kontakte nur mäßig befriedigende Ergebnisse, zweitens würden wir viele dieser Bauteile benötigen. Der dritte Grund ist der dadurch entstehende enorme Entflechtungsaufwand.

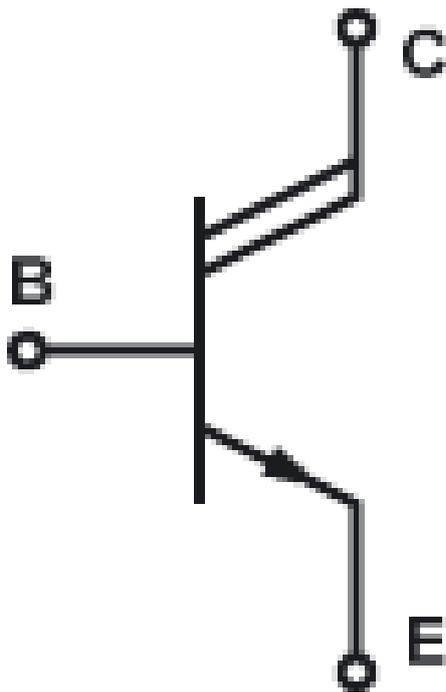


Abbildung4: Schaltzeichen NPN Darlingtontransistor (Elektronik Kompendium, 2016)

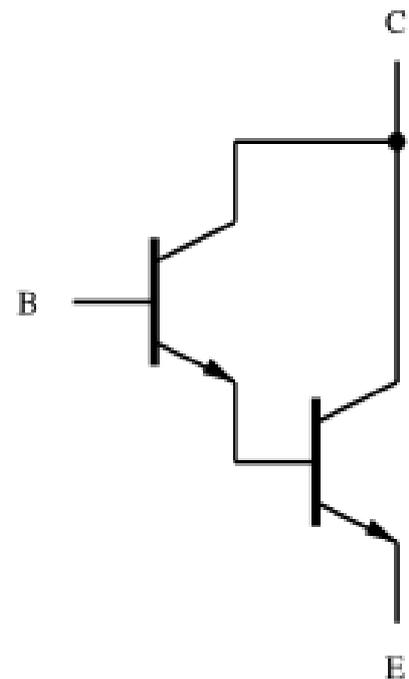


Abbildung 5: Ersatzschaltbild NPN Darlington Transistor Abbildung (Wikipedia, 2016a)

Im Versuchsaufbau am Steckbrett konnten wir erfolgreich unsere Finger durch Berühren zweier blanker Drähte als Schalter einsetzen. Das funktioniert mithilfe eines Pullup- oder eines Pulldown-Widerstandes. Diese Art der Eingabe wählten wir aufgrund der direkten Berührung der Leitungen (potenzielle Gefahr im Netzteilbetrieb) ab.

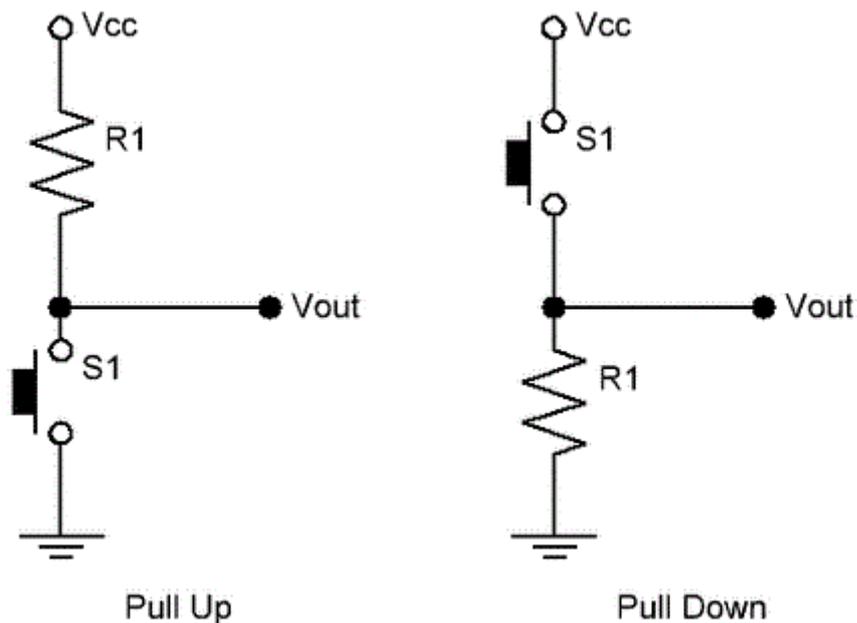


Abbildung 6: Zweidrahttechnologie Schaltungsaufbau Pull Up / Pull Down Widerstand (Tuelectronica, 2012)

8.2.3 CAPACITIVEPIN.LBR

Das ist eine vorgefertigte Library (playground.arduino, 2016) für den Arduino. Mit dieser Library kann man die Bauteile auf einen leicht zu entflechtenden und kostengünstigen Kondensator beschränken. Dieser wird in Serie zwischen den Mikrocontroller und der zu messenden Fläche verbaut. Dieser Kondensator dient als Entkopplungskondensator und hat die Aufgabe, die 50Hz Schwingungen aus der Umgebung zu dämpfen. Mit Hilfe eines Befehls wird ein digitaler Pin „High“ bzw. logisch 1 gesetzt und danach abgefragt. Das Abfragen wird bis zu 17mal wiederholt. Am Ausgang des Pins wird ein Kondensator geladen. Durch das zyklische Laden des Kondensators erhöht sich das Spannungspotenzial des Kondensators. Wenn dieses Spannungspotenzial den Logikpegel des Mikrocontrollers übersteigt, wird dies bei der Abfrage des Pins erkannt und gibt die Anzahl der benötigten „Lade-Abfrage Zyklen“ aus. Je mehr Zyklen benötigt werden, desto höher ist die Kapazität am Pin. Leider verhielt sich diese Technik in den Versuchen sehr willkürlich und unflexibel bei indirekter Berührung.

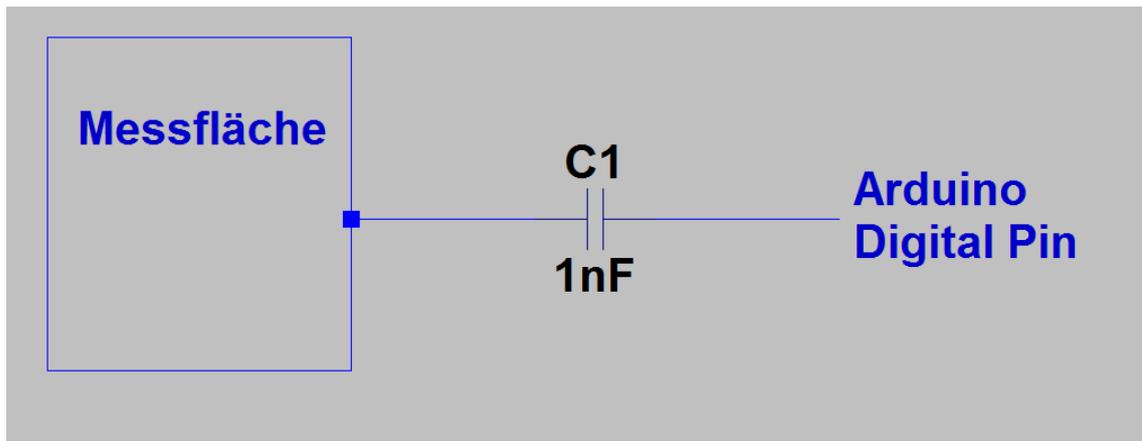


Abbildung 7: Schaltungsaufbau CapacitivePin.lbr

8.2.4 CAPACITIVESENSOR.LBR

Diese vorgefertigte Library (Badger, 2016) erfüllt unsere Kriterien. Das Messprinzip ist ähnlich wie bei der CapacitivePin.lbr. Es gibt einen „Sende Pin“ der über einen Widerstand mit einem „Empfänger Pin“ verbunden wird. Bei der Abfrage des Sensors wird der „Sende Pin“ „High“ gesetzt und der „Empfänger Pin“ fragt seinen Eingang ab. Sobald der Logikpegel am „Empfänger Pin“ überschritten wird, endet eine Messung und die benötigte Zeit wird gespeichert. Da diese Messung sehr schnell geht kann man diese mehrmals ausführen um etwaige Störungen herauszufiltern. Die Ergebnisse der Messungen werden anschließend von der Library summiert, um daraus den Mittelwert zu bilden. Dieser Wert kann gespeichert werden und dient der Zustandserkennung.

Die Messzeit lässt sich durch den Widerstand und durch die Kapazitäten am „Empfänger Pin“ beeinflussen und verhält sich daher sehr flexibel und lässt Spielraum für den weiteren Aufbau. Einziger Nachteil ist lediglich, dass die Gesamtzeit der Messung sich mit der Anzahl der Sensoren summiert und man daher die Dimensionierung der Widerstände so wählen muss, dass die Zeit der Messung möglichst kurz und dennoch aussagekräftig ist.

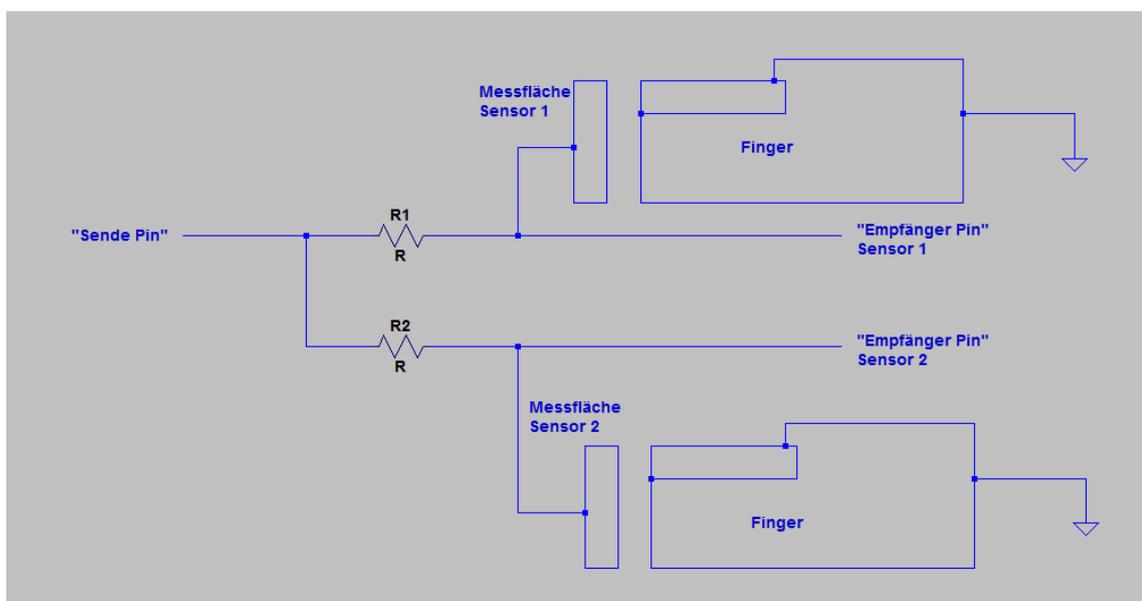


Abbildung 8: Schaltungsaufbau CapacitiveSensor.lbr

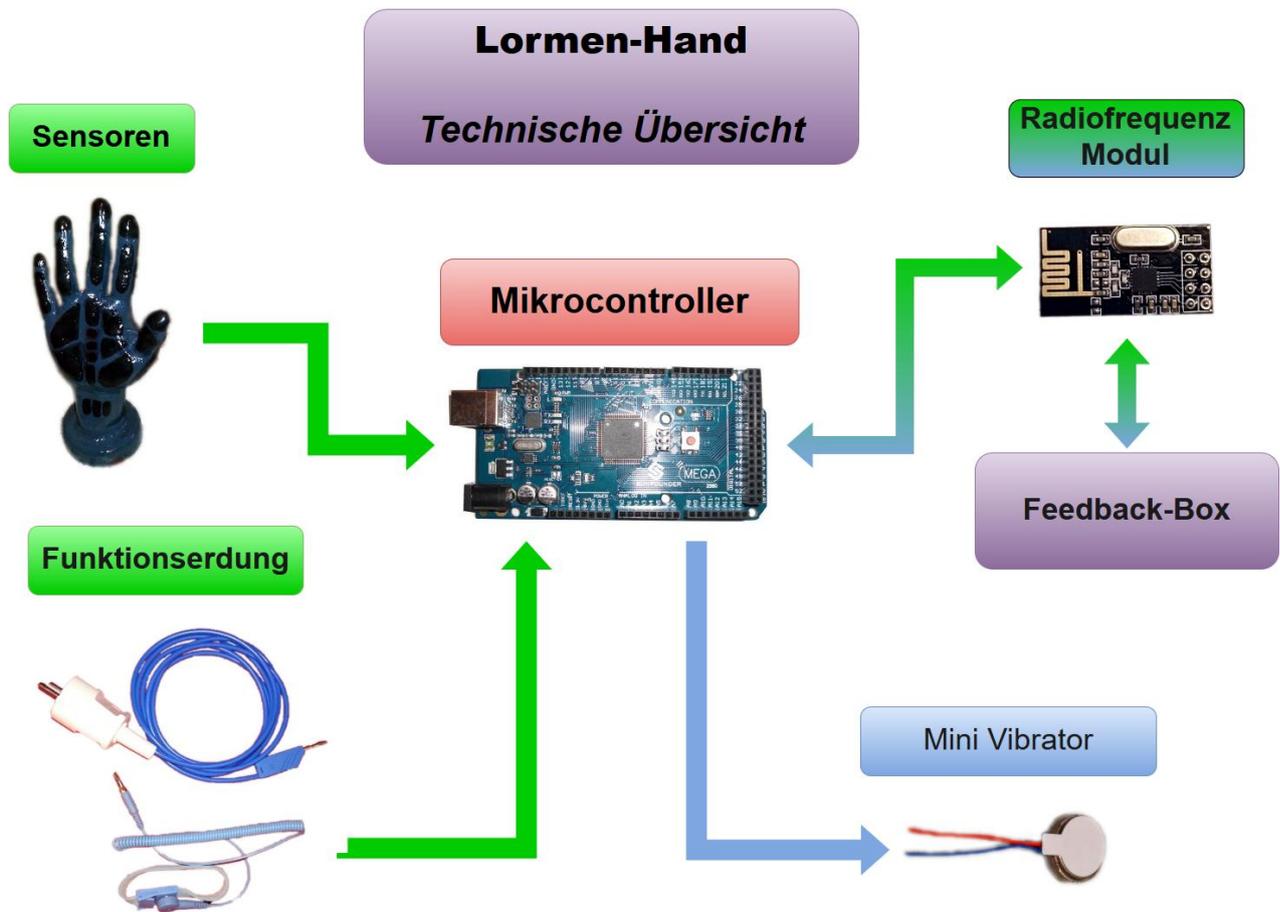


Abbildung 9: Technische Übersicht Eingabegerät: Lormen-Hand

Sobald das Eingabegerät mit Spannung versorgt wird, kalibriert der Mikrocontroller die Sensoren an der Kunststoffhand, und ist danach einsatzbereit. Die Sensoren sind auf der Handnachbildung so positioniert, dass der Mikrocontroller Lorm-Buchstaben erkennen und übersetzen kann. Durch Berühren der kapazitiven Sensoren vergrößert man die Kapazität des Sensors. Diese Kapazität wird einmal pro Messung über einen fixen Widerstand geladen. Die Ladezeit der Kapazität wird als Integere (Datentyp für ganze Zahlen) in Millisekunden ausgegeben. Übersteigt diese Zeit den vordefinierten Schwellenwert, wird dies als Berührung interpretiert und gezählt. Im Anschluss werden alle Sensoren zyklisch ausgelesen. Sobald keiner der Sensoren eine Berührung detektiert, läuft eine Zeit von 0,5 Sekunden ab, in der die Sensoren weiterhin zyklisch ausgelesen werden. Geschieht dies ereignislos, werden die Zähler der Sensoren mit einer Wahrheitstabelle verglichen. Die Wahrheitstabelle besteht aus logischen Verknüpfungen (UND, ODER). Stimmen die Zähler mit einer der logischen Verknüpfungen überein, kann die Berührung einem Buchstaben zugeordnet werden. Als Rückmeldung für die Erkennung dienen zwei kleine Vibratoren, die an der Kunststoffhand angebracht sind. Dieser Buchstabe wird anschließend mittels Funkverbindung an das Ausgabegerät (Feedback-Box) gesendet.

Wenn das Ausgabegerät mit der weiteren Verarbeitung des Satzzeichens fertig ist, sendet es den Buchstaben wieder an das Eingabegerät zurück. Stimmen gesendeter und empfangener Buchstabe überein, werden die Vibratoren zwecks Rückmeldung nochmals kurz gestartet, die Zähler wieder auf Null gesetzt, die Sensoren nochmals kalibriert und die Schleife beginnt von vorne. Im Fehlerfall vibrieren die beiden Vibratoren drei Mal in Folge um den/die Nutzer/in darauf aufmerksam zu machen.

8.4 SCHALTPLAN

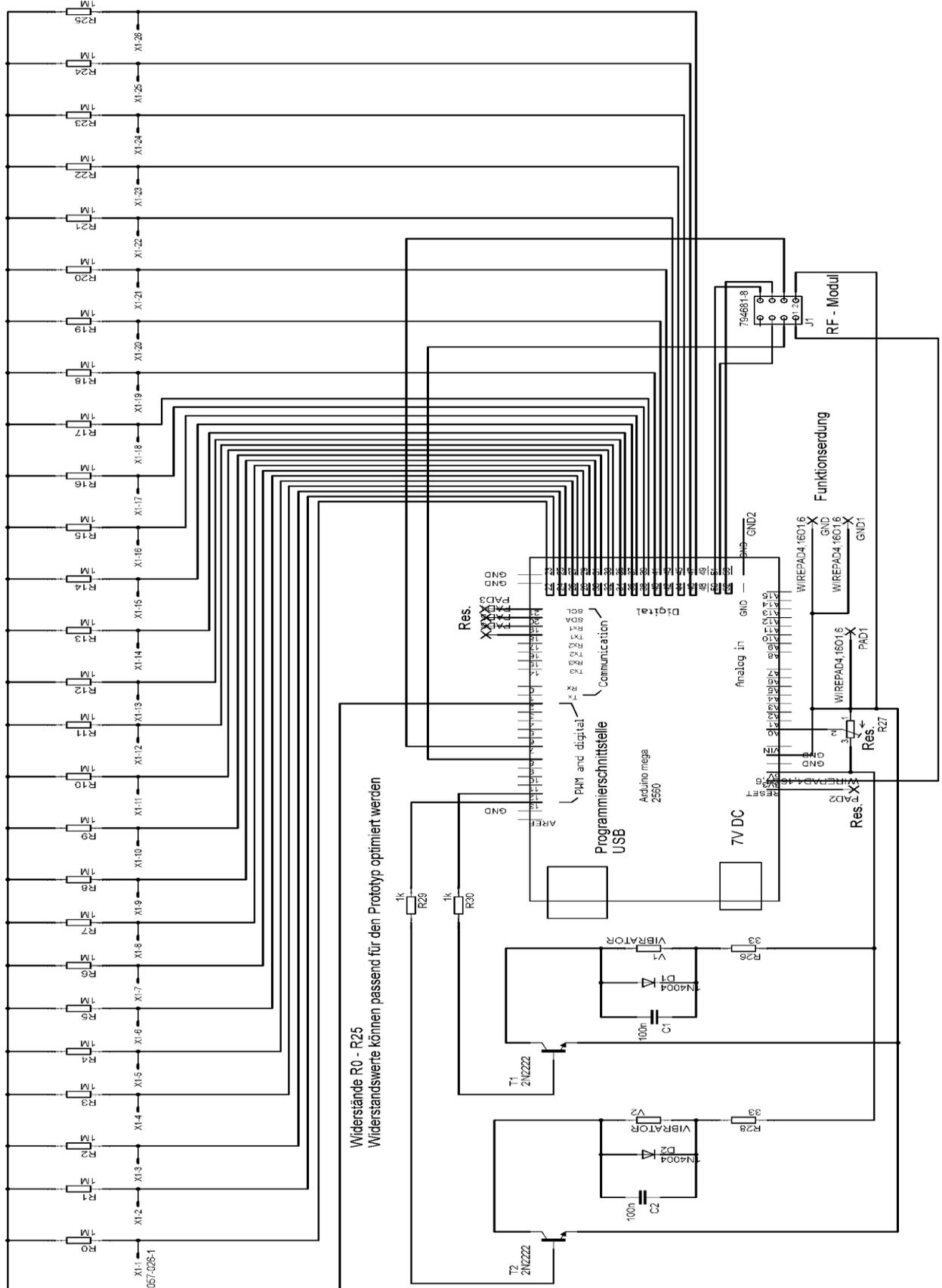


Abbildung 10: Schaltplan Lormen-Hand Gesamtansicht

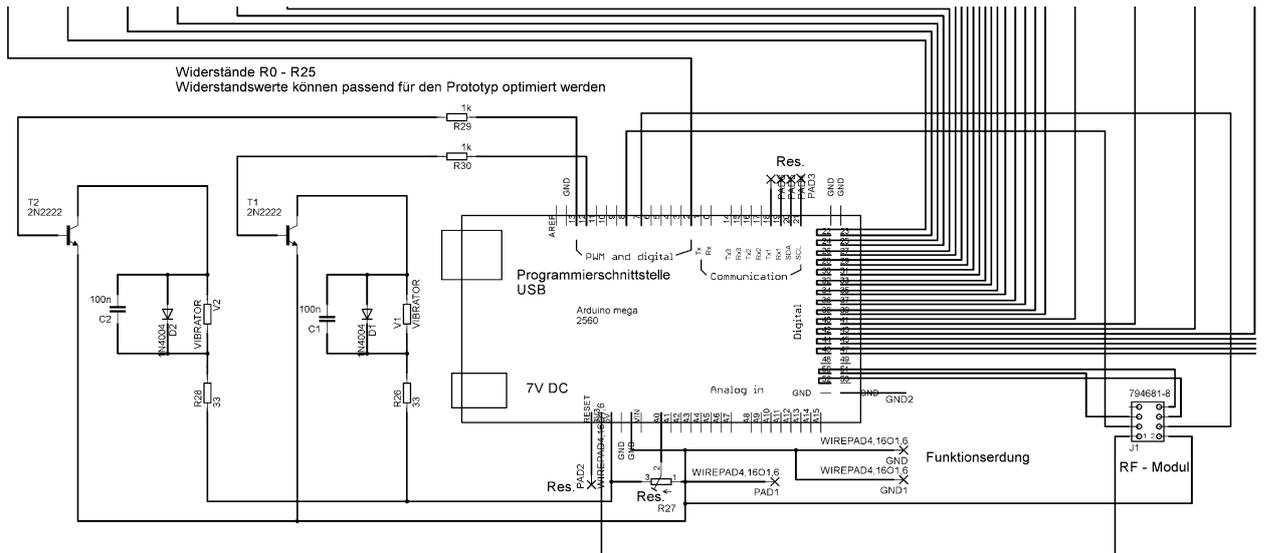


Abbildung 11: Schaltplan Lormen-Hand Detailansicht Mikrokontroller

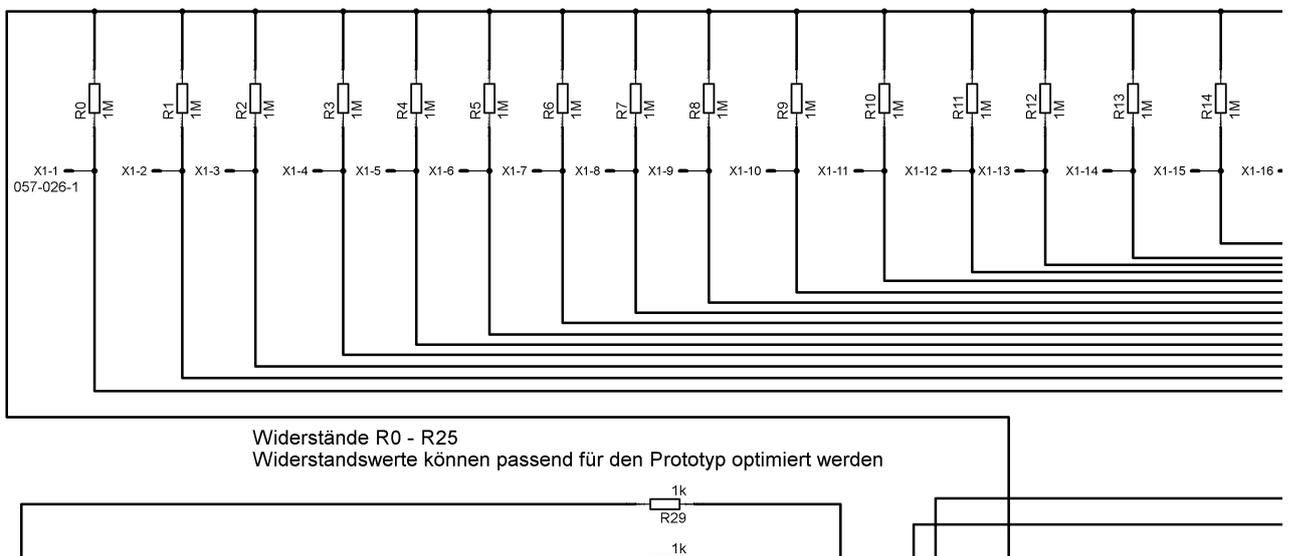


Abbildung 12: Schaltplan Lormen-Hand Detailansicht Sensorwiderstände 1 von 2

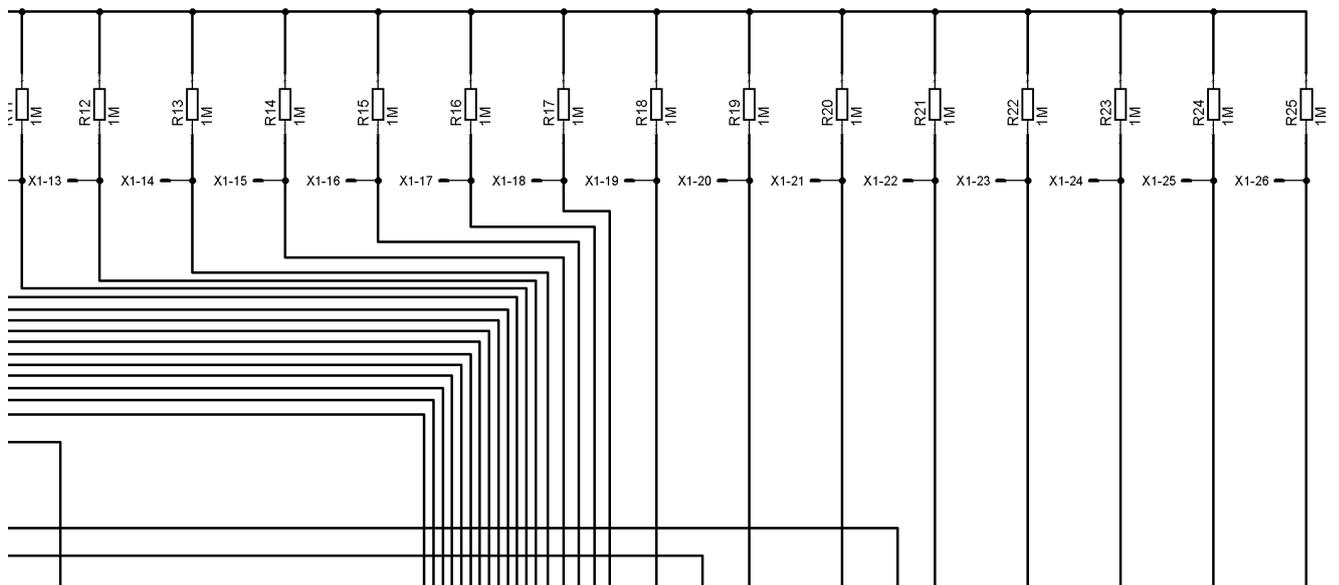


Abbildung 13: Schaltplan Lormen-Hand Detailansicht Sensorwiderstände 2 von 2



Abbildung 14: Skizze Sensorflächen – nummeriert

Der Schaltplan wurde mit Eagle erstellt, da Eagle auch die Funktion bietet mit dem Schaltplan eine Platine zu planen und zu zeichnen. Der Datensatz wird im Anhang in digitaler Form mitgeliefert. Die vorhergehende Skizze zeigt die Position der 26 Sensoren samt der Nummerierung. Die Nummern der Sensoren sind ident mit der Nummer des dafür benötigten Widerstands (Sensor0 – R0, Sensor1 – R1; etc.). Für den Programmcode wählten wir ebenfalls diese Zählweise.

8.4.1 DIMENSIONIERUNG DER SCHALTUNG

Um dieses Prinzip der Messung verstehen zu können, muss man wissen, wie ein Kondensator funktioniert und aufgebaut ist. Ein Kondensator besteht aus zwei gegenüberliegenden Platten. Zwischen den Platten ist ein Isolator (Dielektrikum). Legt man nun eine Gleichspannung an eine der Platten an, entsteht ein elektrostatisches Feld und ein Potentialunterschied zwischen den beiden Platten. Der Kondensator speichert die hinzugefügte Energie. Diese Speicherkapazität wird in Farad angegeben.

8.4.1.1 KAPAZITÄT EINES PLATTENKONDENSATORS:

$$C \left[\frac{As}{V} \right] [F] = \frac{\epsilon_0 * \epsilon_r * A}{d}$$

A ... Plattenoberfläche

d ... Plattenabstand

ϵ_0 ... Elektrische Feldkonstante / absolute Dielektrizitätskonstante $8,85 \cdot 10^{-12} \left[\frac{As}{Vm} \right]$

ϵ_r ... Dielektrikum

Dielektrikum	ϵ_r
Luft	1
Papier	2
Keramik	6

(Schnabel, 2007, S. 118)

An der Formel erkennt man, dass die Kapazität des Kondensators von der Plattengröße, dem Plattenabstand und vom Dielektrikum abhängig ist.

8.4.1.2 LADEN EINESKONDENSATORS

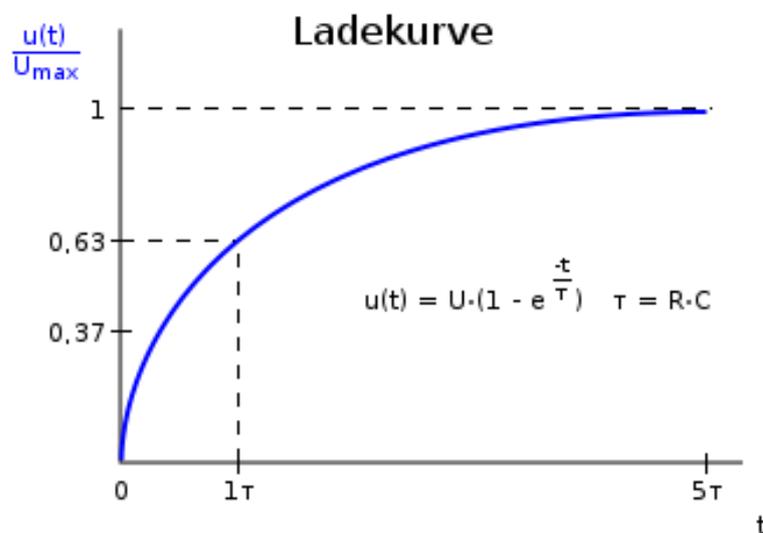


Abbildung 15: Ladekurve eines Kondensators (WeTec's Technikseite, 2010)

Die Ladekurve eines Kondensators ist eine Exponentialfunktion. Am Kurvenverlauf sieht man, dass der Spannungsaufbau am Kondensator mit fortgeschrittener Zeit abnimmt, bis die Spannung annähernd die der Quelle ist. Dies geschieht bei 5τ (tau). τ ist in dieser Funktion die Zeitkonstante. Sie ist das Produkt aus dem Widerstand und der Kapazität des Kondensators. Nach einem τ ist der Kondensator bereits zu 63% geladen.

8.4.1.3 ENTLADENEINESKONDENSATORS

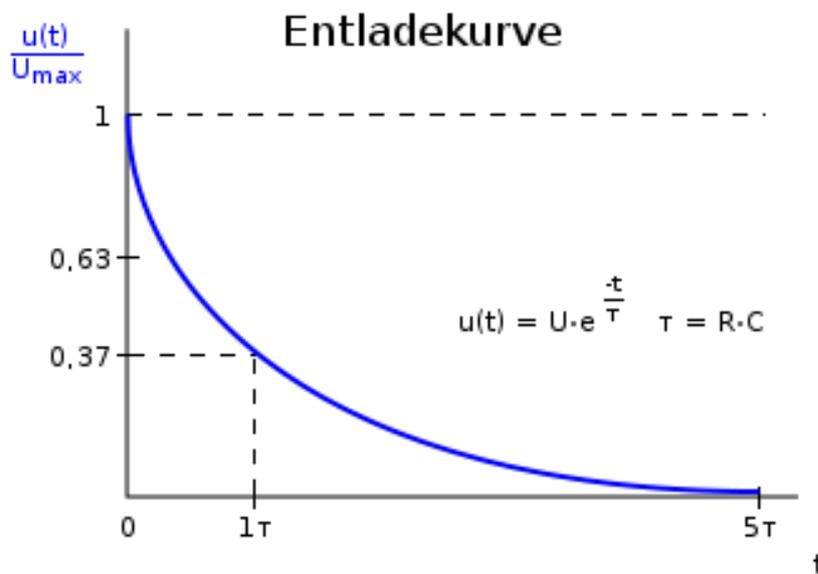


Abbildung 16: Entladekurve eines Kondensators (WeTec's Technikseite, 2010)

Ähnlich dem Laden verhält sich der Kondensator beim Entladen. Nach einem τ hat der Kondensator 63% seines Spannungspotentials verloren.

8.4.1.4 ERKENNTNIS

Die Sensoren (Graphitflächen, siehe 8.6 Herstellung) bilden eine Platte eines Kondensators. Die zweite Platte bildet die Umgebung. Hierbei ist das Dielektrikum so groß, dass die Kapazität des Sensors verschwindend gering ist. Sobald der Nutzer sich mit seinem Körper der Graphitfläche nähert, verringert er den Plattenabstand und die Kapazität des Sensors steigt. Hierbei bildet der Nutzer die zweite Platte des Kondensators.

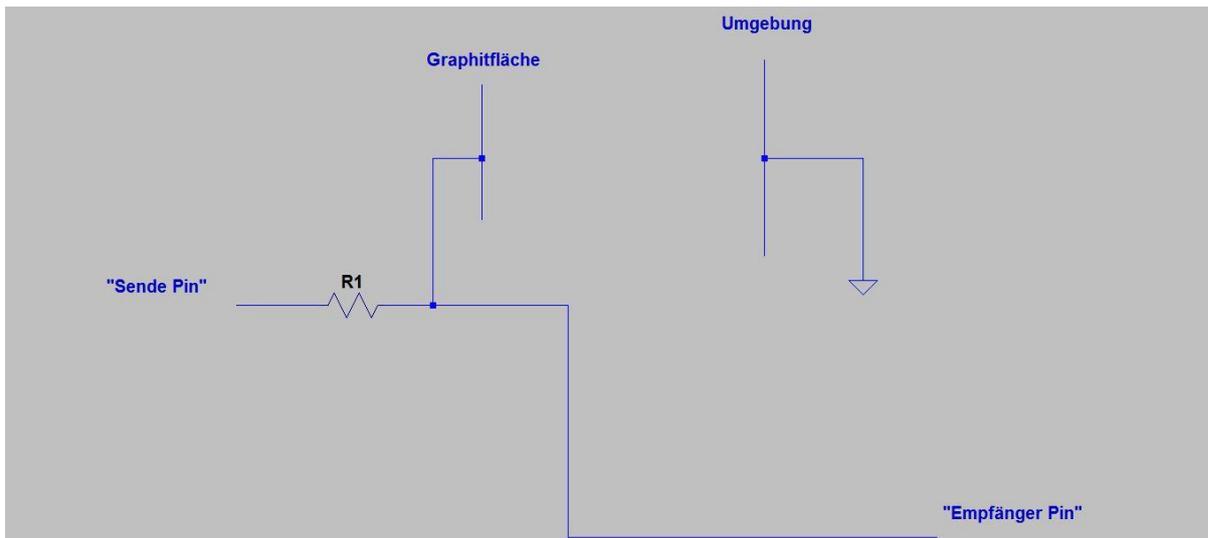


Abbildung 17: Ersatzschaltbild Kondensator im Schaltungsaufbau mit niedriger Kapazität

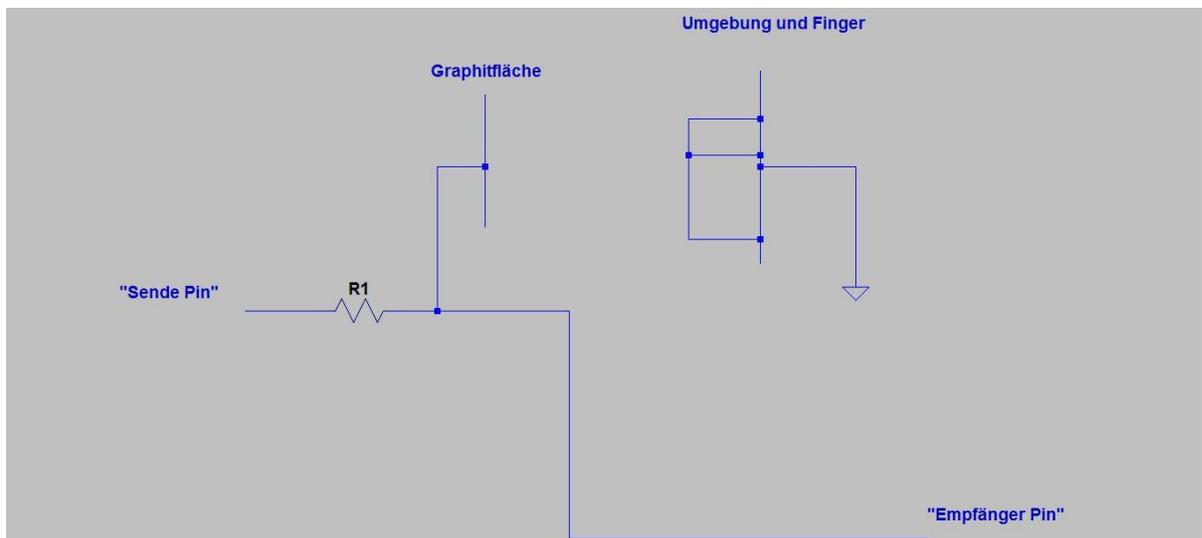


Abbildung 18: Ersatzschaltbild Kondensator im Schaltungsaufbau mit größerer Kapazität

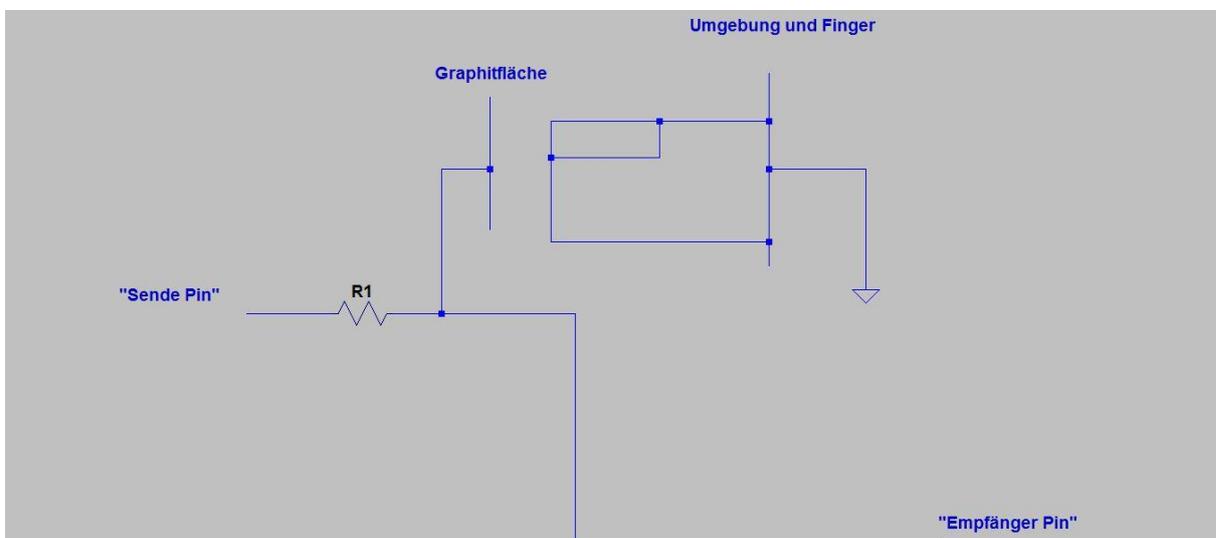


Abbildung 19: Ersatzschaltbild Kondensator im Schaltungsaufbau mit maximaler Kapazität bei Normalgebrauch des Eingabegerätes

Da wir annähernd gleiche Auslesezeiten der einzelnen Sensoren erreichen wollen, um Fehlmessungen, Übersprechen und Störsignale zu dämpfen, müssen wir die Zeitkonstante τ ($R \cdot C$) anpassen. Da unsere Sensorflächen unterschiedliche Größen haben und der Plattenabstand aufgrund des per handaufgetragenen Dielektrikums (Klarlack) variiert, ist die Kapazität aller Sensoren unterschiedlich. Um die Zeitkonstante anzugleichen, muss man demnach den Widerstand anpassen.

Je größer die Kapazität des Kondensators ist, desto kleiner ist der Widerstand zu wählen.

Da es sich bei unserem Eingabegerät um ein per Hand gefertigtes Produkt handelt, werden die Widerstände durch einen Versuchsaufbau ermittelt. Zunächst wird ein kurzer Sketch (Code für den Arduino) geschrieben. In diesem Sketch wird die Messung eines Sensors am Serial Monitor der Arduino Software ausgegeben. In den Schaltungsaufbau wird ein Sensor integriert und der Widerstand nach und nach angepasst, bis die Auslesezeit befriedigend und annähernd gleich ist.

Basierend auf der Theorie wird die Zeit mit Verkleinern des Widerstandes reduziert.



Abbildung 20: Kunststoffhand mit Sensoren

8.5 PCB

Die Platine wurde mit Eagle, basierend auf dem Schaltplan, welcher bereits vorgestellt wurde, erstellt. Der Datensatz wird im Anhang in digitaler Form mitgeliefert. Die Bemaßung der folgenden Platine versteht sich in Millimeter. Die graue rechteckige Fläche symbolisiert den Mikrocontroller (Arduino Mega). Beim Zusammenbau wird die Platine auf den Mikrocontroller gesteckt.

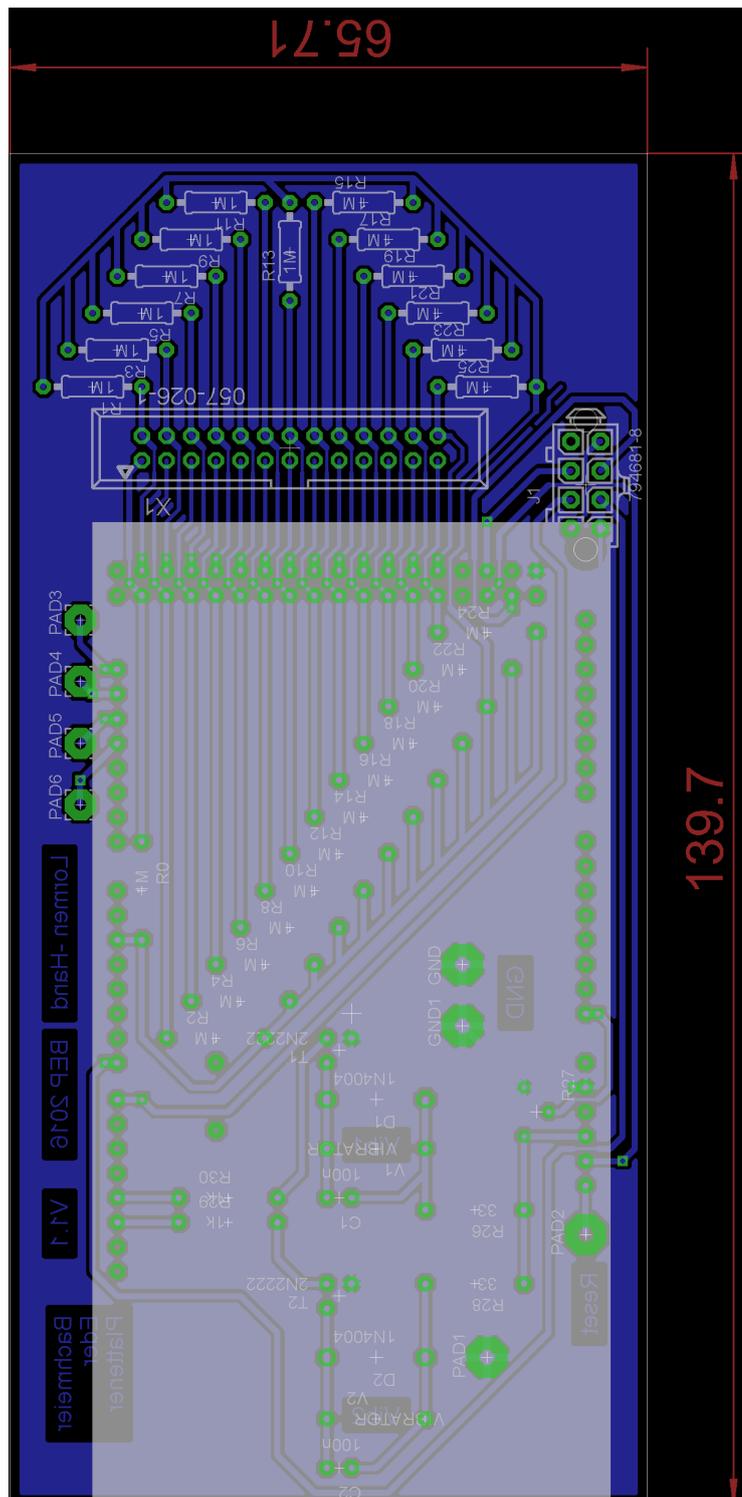


Abbildung 21: Leiterplatte für Eingabegerät Lormen-Hand, Bauteilansicht (THT, Bauteile oben)

8.6 HERSTELLUNG

Die Hardware herzustellen war eine der größten Herausforderungen. Im Zuge des laufenden Prozesses entstanden drei Prototypen. Ziel war es ein Eingabegerät zu bauen, welches unseren Vorstellungen als „lormentauglich“ entspricht und sich mit der Sensortechnologie kombinieren lässt.

Unser Ziel war es somit einen Aufbau zu konstruieren, der die folgenden Merkmale vereint:

- Kompatibel mit der Sensortechnologie
- Gut bedienbar
- Stabiler Aufbau und Stand
- Orientierungsmöglichkeit durch Ertasten
- Angenehme Haptik
- Optisch möglichst originalgetreu zu einer echten Hand
- Korrosionsbeständig (Schweiß)

Um all diese Eigenschaften in einem Konstrukt miteinander zu verbinden, experimentierten wir mit verschiedenen Herstellungsverfahren und Baustoffen.

8.6.1 LORMEN - HAND

8.6.1.1 PROTOTYP I



Abbildung 22: Prototyp I

Wir haben ein Mousepad in Form einer Hand ausgeschnitten. Als Sensoren dienen Flächen aus Alufolie, welche mittels doppelseitigen Klebebands angebracht wurden. Die galvanische Verbindung zum Mikrocontroller wurde mittels Kupferdrähten, die an metallisch-blanken Heftzwecken angebracht wurden, hergestellt.

Dieser Aufbau eignet sich leider nur bedingt für unsere Zwecke. Die Hand verrutschte bei der Eingabe, weil sie zu leicht und zu beweglich ist. Weiters war der Prototyp für blinde Nutzer schlecht bedienbar, weil der Prototyp keinen Platz hatte, um sich mit der zweiten Hand daran zu orientieren. Optische und haptische Verbesserungen wären mit diesem Aufbau auch nur schwer möglich.



Abbildung 23: Prototyp II Handfläche



Abbildung 24: Prototyp II Handrücken

Dieser Prototyp ist der zweite Versuch eine Hand mit Sensoren aus Epoxidharz herzustellen. Beim ersten Versuch gelang es uns lediglich, einen Ziegel aus Epoxidharz und Messing zu produzieren. Nachdem wir weitere Erfahrungen im Umgang mit Gießharzen und Alginat gesammelt hatten, konnten wir beim zweiten Versuch ein vorzeigbares Ergebnis erzielen.

Es handelt sich hier um eine möglichst exakte Nachbildung einer menschlichen Hand aus Epoxidharz, in die Messingbleche eingearbeitet sind. Die Messingbleche dienen als Sensorflächen, und sind mittels steifer Kupferdrähte mit dem Mikrocontroller verbunden. Wir haben uns für den Werkstoff Epoxidharz entschieden, da er stabil und wasserfest ist. Er lässt sich gießen und nach der Aushärtung gut mechanisch bearbeiten.

1. Es wurde ein zweiteiliger Handabdruck aus Alginat gemacht. Alginat ist ein Pulver aus Seetang, welches mit Wasser vermengt wird. Es härtet in etwa 1-5 Minuten aus (je nach Mischungsverhältnis, Raum- und Wassertemperatur). Im ausgehärteten Zustand ist es elastisch und vergleichbar mit festem Pudding. Meist wird es von Zahntechnikern bei Kieferabdrücken verwendet, weil es ungiftig ist und schöne und genaue Abdruckergebnisse liefert.

Um ein gutes Ergebnis bei dieser Verarbeitungstechnik zu erzielen haben wir uns eine Spezialform angefertigt. Die Spezialform besteht aus drei Teilen:

- Einen Boden mit Rahmen um den unteren Teil der Hand zu modellieren.
- Eine Einfassung, die auf den Rahmen gesteckt werden kann. Dadurch verdoppelt sich die Höhe der Kiste und ermöglicht ein passgenaues Abformen der zweiten Hälfte
- Ein Deckel der in den Rahmen passt. Dadurch lassen sich die beiden Hälften aneinander drücken. Das verhindert ein Auslaufen des Harzes und das Verrutschen der Formen beim Gießen.



Abbildung 25: Prototyp II Alginatabdruck

Hinweis: Die runden Ausschnitte haben den Zweck die beiden Negative noch genauer übereinander positionieren zu können, als es mit der Spezialform ohnehin schon möglich ist.

2. Messingbleche wurden in die Formmasse der Handfläche, in die für die Sensorpositionierung geplanten Stellen gesteckt und mit steifen Kupferdrähten verlötet. Die Kupferdrähte wurden mit Kupferösen als Abstandhalter fixiert, damit diese nur gebündelt an der geplanten Stelle aus der Form ragen.



Abbildung 26: Prototyp II Alginate verdrahtet

3. Danach wurden die zwei Abdrücke zusammengefügt und mit blau eingefärbtem Epoxidharz ausgegossen. Die Formen wurden durch eine Plastikfolie und eine extra angefertigte Spezialform aus MDF Holz (mitteldichte Holzfaserplatte) zusammen gehalten, damit das Verrutschen und Verformen der Gussform beim Ausgießen vermindert wird.



Abbildung 27: Prototyp II Spezialform

4. Nach dem Aushärten wurde der Rohling vom Alginat getrennt. Auf den ersten Blick kann man erkennen, dass die zweiteilige Abdruckform trotz Vorbeugung geringfügig verrutscht ist, da das Gießharz zu dünnflüssig für diese Gießtechnik ist. So konnte das Harz an den Übergängen der zwei Formen auslaufen.



Abbildung 28: Prototyp II - Rohling Handfläche



Abbildung 29: Prototyp II - Rohling Handrücken

5. Danach mussten wir die Messingbleche außerhalb des Gießlings und den Epoxidharzüberschuss entfernen. Hierfür verwendeten wir einen „Dremel“ (Übliche Bezeichnung für ein Multifunktionsgerät zum Schleifen, Polieren, Fräsen und Bohren. Namensherkunft ist der bekannteste Hersteller dieser Geräte.). Um den feinen Epoxid- und Messingstaub nicht zu inhalieren, verwendeten wir Staubschutzmasken und versuchten zusätzlich den Schleifstaub so gut wie möglich, mit einem handelsüblichen Staubsauger abzusaugen.



Abbildung 30: Prototyp II - Mechanische Bearbeitung

Das Endergebnis ist, trotz einiger Lufteinschlüsse, optisch durchaus vorzeigbar. Lediglich die Haptik und die Sensoren lassen Spielraum nach oben offen.

Haptisch fühlt man die Messingbleche, da diese mit dem Harz keine stufenlose Fläche bilden. Grund dafür ist die unterschiedliche Härte der beiden Grundmaterialien. Beim großflächigen Behandeln der Oberfläche wird mehr Epoxidharz (weich) als Messing (hart) abgetragen. Dadurch entsteht eine kleine Stufe an den Übergängen, welche man fühlen kann.

Die Auswertung der Sensoren ist ebenfalls ungünstig. Es zeigte sich, dass ein Einsatz von größeren Flächen von Vorteil ist. Oft werden die Berührungen nicht wahrgenommen, da die Messingbleche nur eine kleine, meist unterbrochene Fläche bilden. Der schreibenden Person wird dadurch nicht viel Freiheit geboten. Sie muss das Messing exakt berühren, um den Sensor zu stimulieren. Des Weiteren kommt es zu einer Doppelzählung durch die Unterbrechungen der Bleche. Das ungenaue Zählen bewirkt eine falsche Zuordnung der Eingabe, welche aufgrund der unterschiedlichen Fingergrößen der potentiellen Nutzer/innen im Programmcode jedoch leider nicht kompensierbar ist. Dennoch kann sich das Ergebnis zeigen lassen und ermutigte uns an der Gießharzidee festzuhalten.



Abbildung 31: Prototyp II - Vergleich mit Modell



Abbildung 32: Prototyp III - incl. Gehäuse

Dieser Prototyp entspricht all unseren Vorstellungen eines Eingabegeräts für das Lorm– Alphabet. Der Entwurf basiert auf dem Prototyp II. Es gibt jedoch einige optischen und funktionelle Veränderungen: Die Sensorflächen wurden mit einer dünnen Schicht aus Klarlack überzogen. Das bewirkt eine Trennung des Nutzers zum Stromkreis. Somit werden der Nutzer und der Mikrocontroller vor gefährlichen Überspannungen geschützt. Durch die Trennung der Sensorflächen zur Umgebungsluft sind diese vor

Korrosion geschützt. Dadurch wird das Eingabegerät wartungsfrei.

Optisch ist der Prototyp III ebenfalls besser ausgeführt. Durch das Verwenden einer einteiligen Gussform gelingt der Formabdruck ohne Auslaufen der Gießmasse. Der Arbeitsaufwand des Rohlings ist geringer und das Ergebnis hat weniger Gießfehler. Die Nachbearbeitung ist nur geringfügig aufwendiger und verändert die Struktur an der Handfläche nur wenig.

Um diese Technik anzuwenden, machten wir einige Versuche mit abdrücken unserer Finger. Zu Beginn wollten wir die Sensorflächen aus Zwei-Komponentenknete (2K Knete) auf Epoxidharzbasis mit 1-5% Aluminiumanteil formen. Leider ist der Aluminiumanteil zu gering um genügend zu leiten. Als wir versuchten der Epoxidknete Aluminium beizumengen konnte sich diese nicht mehr mit dem Rohling verbinden. Aus den dadurch resultierenden Ergebnissen konnten wir ableiten, dass die Knete Signale besser isoliert als die Kabelisolierung und sich unbehandelt gut mit dem Gießling verbindet.

Um die Sensorflächen zu bilden, experimentierten wir mit Graphitspray und erzielten fabelhafte Ergebnisse. Selbst unter einer Schicht Klarlack konnte eine Kontaktannäherung sicher detektiert werden. Einziger Wermutstropfen ist, dass der Graphitspray nicht aufgesprüht werden kann, weil dieser sehr dünnflüssig ist. Klebt man die Areale der Sensorflächen ab und behandelt diese mit dem Spray, dringt das flüssige Graphit in die Fingerabdrücke ein und verteilt sich auf der gesamten Epoxidharzform. Das kann man umgehen, indem man das Graphit mit einem Pinsel dünn und mehrschichtig aufträgt.

8.6.1.3.1 FERTIGUNG – PROTOTYP III

1. Im ersten Schritt machten wir einen einteiligen Abdruck aus Alginat in einem ausreichend großen Kübel. Wir füllten den Kübel mit Wasser auf die gewünschte Höhe, wogen ihn ab und verglichen das Gewicht mit dem Leergewicht des Kübels. Die Gewichts Differenz war Grundlage für die benötigte Menge Alginat. Nach demselben Prinzip untersuchten wir, wie viel Wasser aus dem Kübel verdrängt wird, wenn die abzudrückende Hand in die gewünschte Tiefe eintaucht. Da Wasser annähernd das spezifische Gewicht von 1 hat, konnten wir ermitteln wie viel Volumen Epoxidharz benötigt wird, um die Hand in einem Stück zu gießen. Den Sockel für die Befestigung mussten wir hinzuaddieren.

$$\begin{aligned}\text{Alginatvolumen} &= \text{Voller Kübel} - \text{leerer Kübel} \\ \text{Epoxidharzvolumen} &= \text{Handvolumen} + \text{Sockelvolumen} \\ \text{Sockel} &= \pi * r^2 * h\end{aligned}$$

Das Epoxidharzvolumen erhöhten wir um rund 15% und das Volumen der Abdruckmasse um etwa 20%, um auf Nummer sicher zu gehen.



Abbildung 33: Prototyp III - Alginatabdruck

2. Im zweiten Schritt steckten wir einen Ring in die Abformmasse. Der Ring dient als Form für den Sockel.



Abbildung 34: Prototyp III - Sockel

3. Danach wurde die Abformmasse mit Epoxidharz ausgefüllt. Wir fügten dem durchsichtigen Harz weißen Farbstoff hinzu, um Flecken am Arbeitsplatz besser sehen zu können. Dieser Schritt muss möglichst bald nach dem Herstellen der Gussform geschehen, weil die Alginatmasse aus zirka 80% Wasser besteht. Dieses Wasser dringt mit der Zeit aus der Form aus und verdunstet. Dadurch verringert sich die Größe der Form und wird unbrauchbar.



Abbildung 35: Prototyp III - Epoxidharz Gießverfahren

4. Nach zirka 48 Stunden ist der Rohling ausgehärtet und kann vom Alginat getrennt werden.



Abbildung 36: Prototyp III - Rohling

5. In den nächsten Schritten wird der Rohling mechanisch bearbeitet. Hierzu verwendeten wir eine Bohrmaschine und einen Dremel.
Zunächst zeichneten wir die Positionen der Sensorflächen mit einem Filzstift an. Dann bohrten wir 3mm Löcher in den Rohling. Die Löcher reichen von den Sensorflächen zu dem Handrücken. Bei den größeren Flächen bohrten wir zwei Löcher, um die Kontaktsicherheit zu verdoppeln, denn diese Löcher werden in einem späteren Schritt als Drahtdurchführung für einen steifen 2,5mm² Draht verwendet. Diese Drähte dienen als Verbindungen der Flachbandkabeldrähte zu den Graphitflächen.
6. Danach zeichneten wir den Kabelweg am Handrücken an. Hierbei versuchten wir die Drähte des Flachbandkabels so weit wie möglich von einander zu trennen, um ein Übersprechen zu vermeiden. Den Weg wählten wir so, dass sich die Drähte nicht kreuzen.
Entlang dieser Markierungen machten wir mit dem Dremel Platz für das Flachbandkabel.
7. Im nächsten Schritt wurde das Flachbandkabel verlegt und mittels 2K Knete auf Epoxidharzbasis fixiert. Anschließend haben wir einen steifen 2,5mm² Draht in die Bohrungen gesteckt und mit den Flachbandadern verlötet. Hierbei ist es wichtig, dass der steife Draht mindestens 0,5cm aus der Sensorfläche ragt. Danach haben wir die bearbeiteten Areale mit der Knetmasse und dem Dremel rekonstruiert, um den Originalzustand so gut wie möglich wieder herzustellen.



Abbildung 37: Prototyp III – Nachmodellierung

8. Nach Fertigstellung der Modellierung wurde der Prototyp lackiert, um ihm ein gleichmäßiges Aussehen zu verpassen. Wir haben uns für einen grauen Farbton entschieden.



Abbildung 38: Prototyp III - nach dem Lackieren

9. Nach dem Lackieren haben wir vorsichtig die überstehenden Kupferdrähte plan geschliffen und die Sensorflächen mit einem Filzstift vorgezeichnet.
10. Anschließend haben wir das Graphit mit einem Pinsel auf den vorgezeichneten Sensorflächen mehrschichtig aufgetragen. Hierfür sprühten wir den Graphitspray in einen kleinen Becher und tauchten den Pinsel darin ein.
Zum Konservieren und Trennen überzogen wir den Prototyp mit einer Schicht Klarlack. Nach 24 Stunden Trocknungszeit war der Prototyp annähernd fertig.



Abbildung 39: Prototyp III - fertig

11. Im letzten Schritt mussten wir nur noch ein Gelenk an der Unterseite anbringen. Das dient der Einstellung der Neigung. Der Nutzer kann die Hand somit seinem Empfinden nach auf die ideale Neigung justieren, um ein angenehmeres Schreiberlebnis zu haben.



Abbildung 40: Prototyp III - Gelenk

Eckdaten des Gehäuses:

- Abmaße: 248x248x123mm
- Material: durchsichtiges Plexiglas mit Gummipuffer
- Verbindung: geschraubt, magnetisch
- Besonderheiten: abnehmbare Rückwand mit Fixierung via Ferromagnete
- Schnittstellen: Netzteilanschluss, Programmierschnittstelle, Laborbuchse für Funktionserdung, Montagebohrung und Kabeldurchführung für die Kunststoffhand
- Optische Verschönerung: LED-Band an der Unterseite des Gehäuses

Das Gehäuse hat die Aufgabe die Schaltung zu schützen, die Verbindungen der Kunststoffhand zum Mikrocontroller vor direkter Berührung abzuschirmen und der Kunststoffhand einen sicheren Stand zu bieten. Wir haben uns entschieden, das Gehäuse unseres Prototyps aus durchsichtigem Plexiglas zu gestalten. Um Kosten zu sparen, haben wir das Gehäuse in Eigenleistung hergestellt. Hierfür benötigten wir eine Handkreissäge, einen Akkuschauber, Bohrer, Gewindeschneider, Senker, einen Dremel, Kleber sowie diverses Mess- und Handwerkzeug.

Wir haben das Gehäuse mit einer leicht abnehmbaren Rückwand geplant und konzipiert, da es sich um einen Prototypen handelt. Die Rückwand wird lediglich durch kleine Magnete gehalten. Das ermöglicht ein rasches Agieren bei der Fehlersuche.

Herstellung:

1. Schneiden der beiden Grundplatten und der vier Seitenteile
2. Bohren der Löcher für Verbindungen der einzelnen Elemente
3. Ein Gewinde in die Sacklöcher der Seitenteile schneiden
4. Senken der Löcher der Grundplatten an der Außenseite, passend für Senkkopfschrauben
5. Bohren eines Loches für die Befestigung des Winkels der Kunststoffhand in der Mitte der oberen Grundplatte (Deckel)
6. Schneiden eines Loches für das Flachbandkabel der Kunststoffhand
7. Befestigung der Hand am Deckel
8. Bohren und Senken eines Loches für die Laborbuchse (Funktionserdung) in den rechten Seitenteil
9. Löcher für die Befestigung der Gummipuffer an der Bodenplatte bohren und Gewinde schneiden
10. Montage der Gummipuffer
11. Verschrauben der Bodenplatte mit der vorderen und den beiden seitlichen Platten
12. Montage der Laborbuchse
13. Verkleben der Haltemagnete an der Innenseite der Rückwand
14. Montage der Winkel an den Seitenteilen
15. Bohren und Senken der Befestigungslöcher für den Mikrocontroller an der Bodenplatte
16. Reinigen der Komponenten
17. Montage des Mikrocontrollers
18. Aufstecken des PCB
19. Nachbearbeiten der Innenseite der Rückwand (notwendig, aufgrund der dicken Wandstärke der Plexiglasplatten und der kurzen Anschlüsse am Mikrocontroller)
20. Verschrauben des Deckels mit dem restlichen Gehäuse
21. Verlöten der Verbindung der Funktionserde (PCB) mit der Laborbuchse
22. Einfädeln des Flachbandkabels
23. Verbinden des Flachbandkabelsteckers mit dem PCB
24. Reinigen und Einsetzen der Rückwand
25. Anbringen des LED-Band und Fixierung durch Heißkleber

Hinweis: Die Autocad Daten sind im Anhang in digitaler Form bereitgestellt. Sämtliche Maße verstehen sich mit +/- 20% Toleranz.

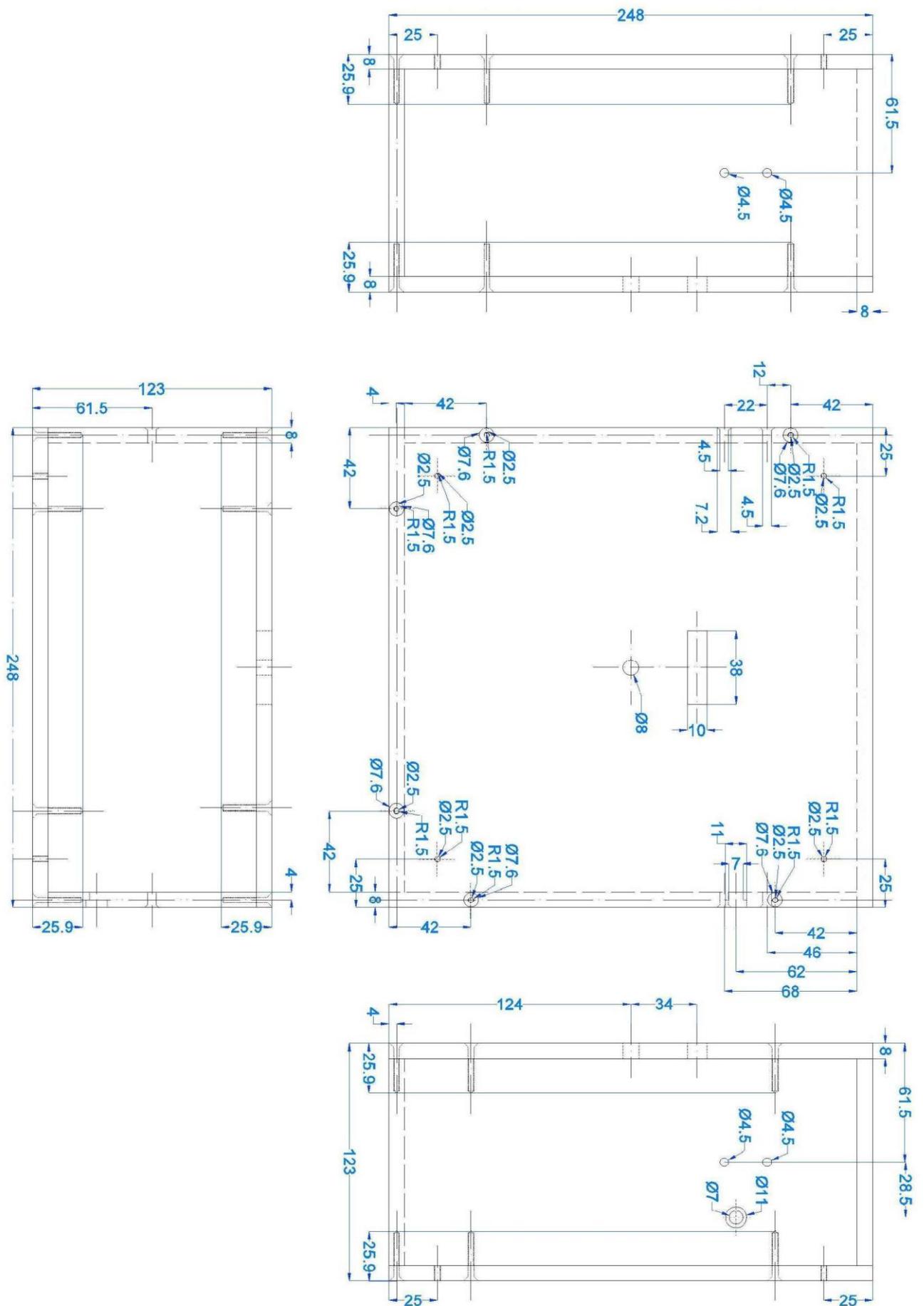


Abbildung 41: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Gesamtansicht

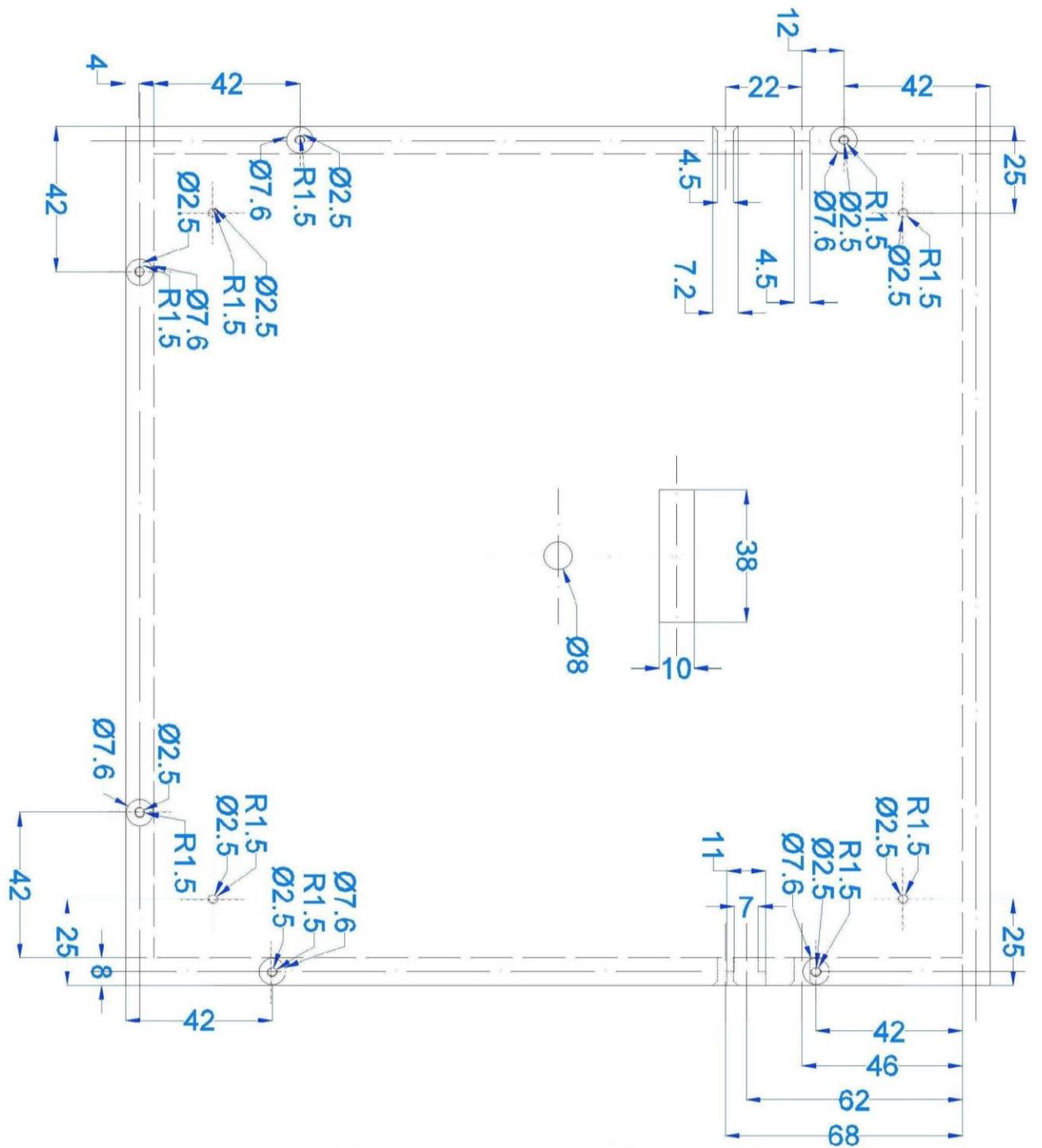


Abbildung 42: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Draufsicht (um 90° nach rechts gedreht)

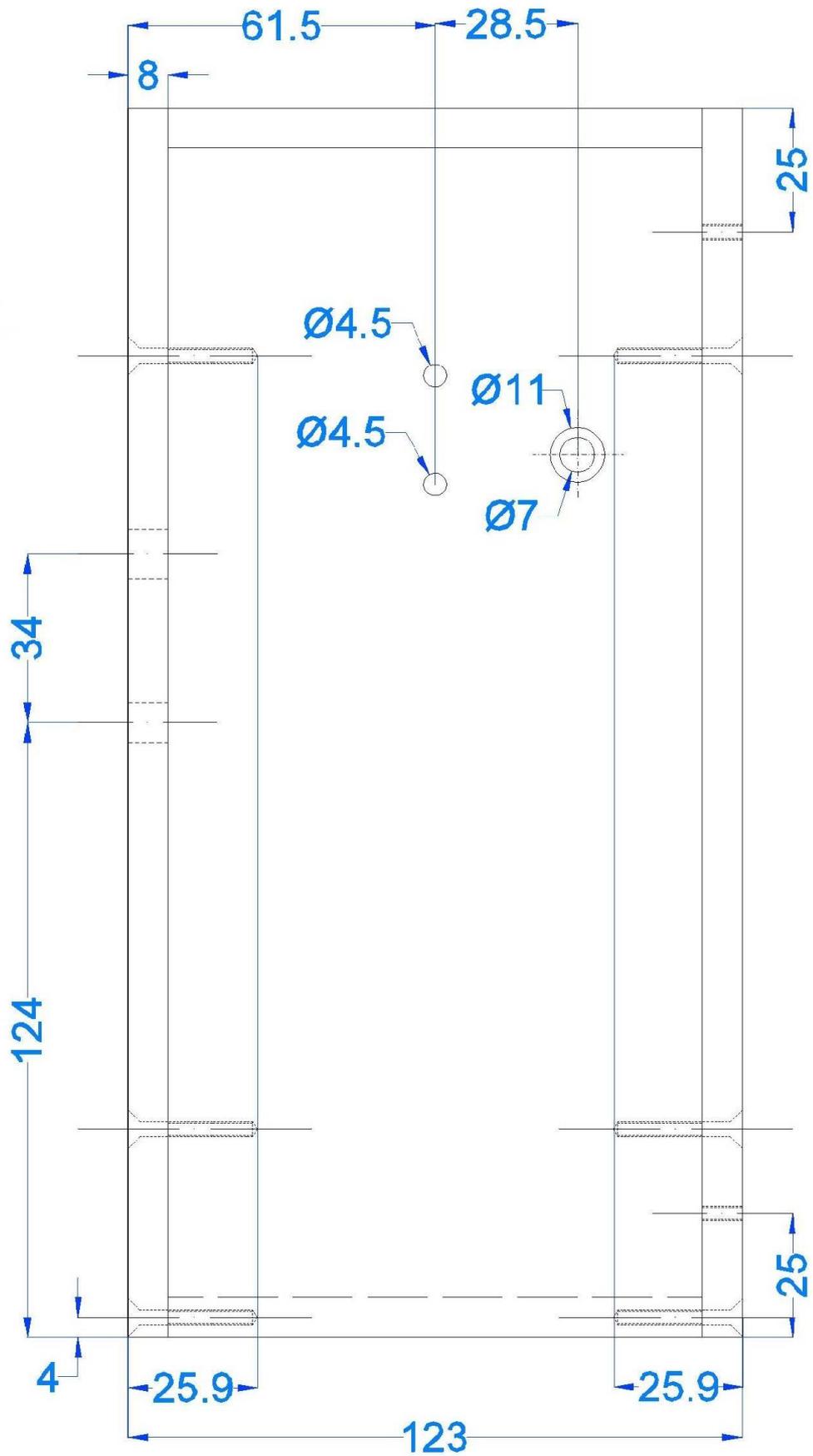


Abbildung 43: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Ansicht von rechts

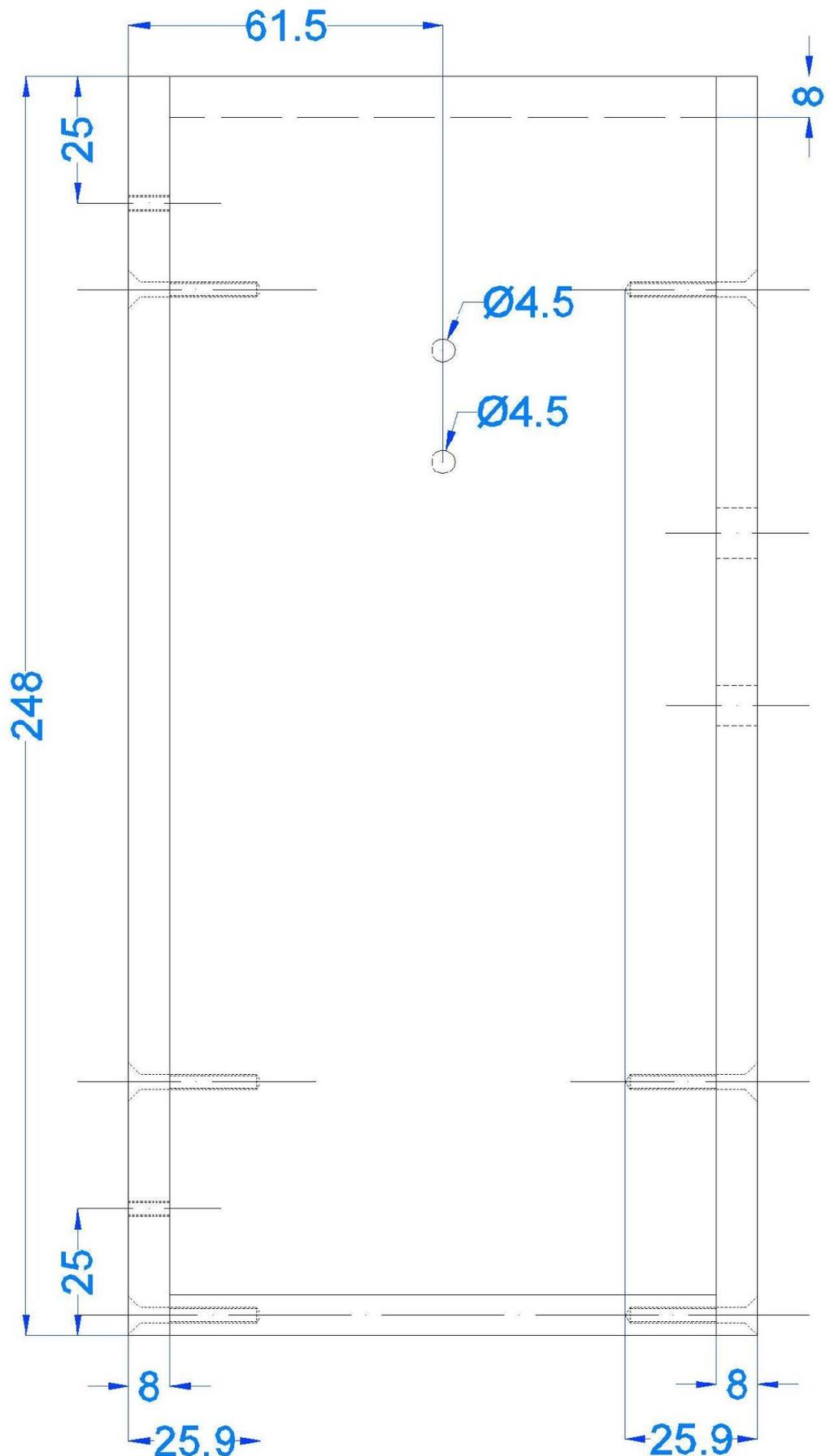


Abbildung 44: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Ansicht von links

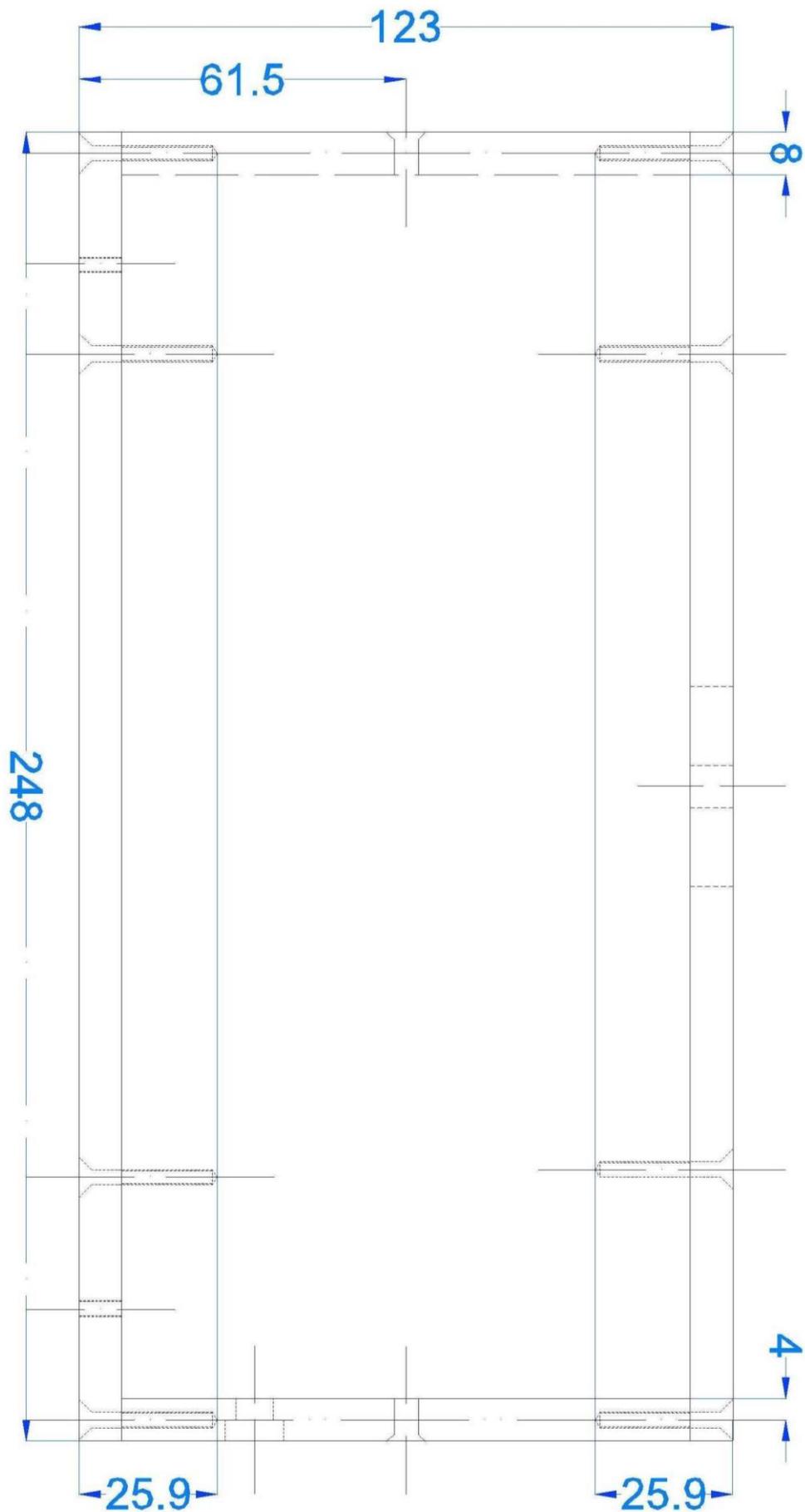


Abbildung 45: Gehäuse Eingabegerät Lormen-Hand: Maßplan - Detailausschnitt Ansicht von vorne (um 90° nach rechts gedreht)

8.7 FLUSSDIAGRAMM

Bei dem Flussdiagramm handelt es sich, aufgrund der Komplexität des Programmcodes, um eine vereinfachte Darstellung. Es dient lediglich dazu, die Funktionsweise der Applikation zu veranschaulichen. Würde man einen exakten Funktionsablauf inklusive der Wahrheitstabellen für die Buchstabenerkennung und der Debugging Einstellungen (Einstellungen um Fehler im Code oder am Aufbau zu finden) einfügen, wäre das Flussdiagramm einige Seiten lang und zu unübersichtlich.

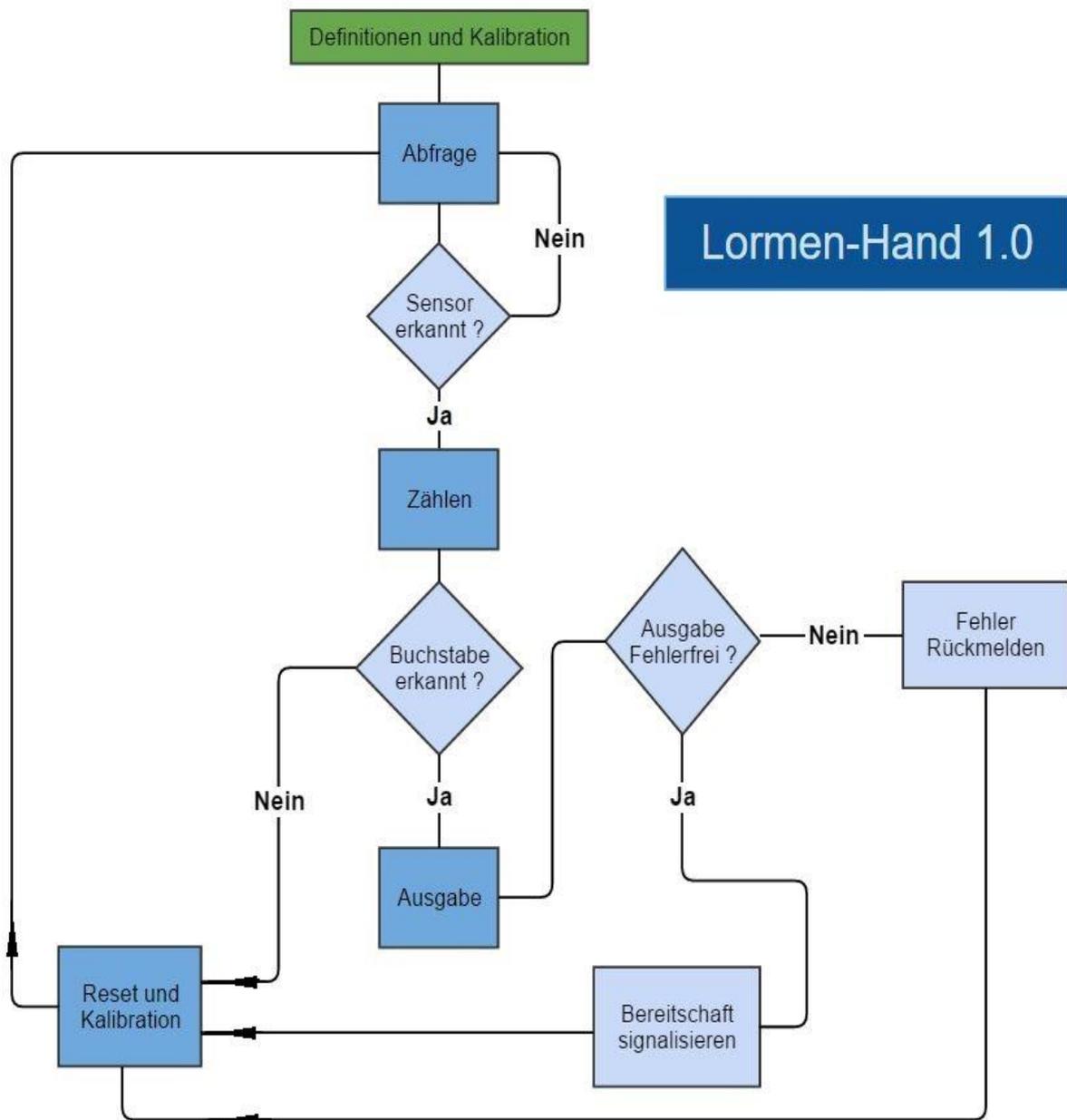


Abbildung 46: Flussdiagramm Lormen-Hand


```

// Definitionen für Übermittlungsfehlerüberprüfung
// Zeiten müssen mit dem Code der Feedbackbox zusammenpassen
inthandvibedelay= 1200; // Summe aus Zeit Vibrator - Vibration und Pause
intnextletterdelay=200;// Zeitabstand zwischen der haptischen Ausgabe der
Buchstaben
intdisplayrefresh=8000;// Zeit die das Display benötigt, um das Bild zu
erneuern
intleerzeichendelay=0; //Spezielle Zeitabfolge bei der haptischen Ausgabe -
Zeit muss in Ausgabe()- if Leerzeichen geändert werden

//Definitionen für Messung und Auswertung
#defineCS(Y)CapacitiveSensor(2,Y) // Generiert die Sensoren - (SendePin,
EmpfangsPin)
//#define COMMON_PIN 2 // Gemeinsamer Pin (Sende) für Sensoren
#definemessungen100 // Anzahl der Messungen /Sensor /Messzyklus mit
Genauigkeit steigt die Zykluszeit
#definesenscount 26 // Anzahl der Sensoren für Schleifen
intinput[senscount]; // Definiere Input[]- speichert die Sensorzustände
intinputsum=0; // Für erste Messung - bildet Quersumme input[]
intstat[senscount]; // Speichert Zustand des Sensors in Abfrage() 1=
betätigt 0= unbetätigt
intcount[senscount]; // Zähler für Ausgabe - zählt wie oft ein Sensor in
dem Eingabezyklus gedrückt wurde
intschwellwert[senscount];// Schwellwerte für Sensoren
inthandmitte=0; // Summe der Sensoren 14,18,19
intfingerspitzen=0; // Summe der Sensoren an den Fingerspitzen
1,3,7,20,25
intfingerflaechen=0; // Summe der Sensoren an den Fingerflächen
0,4,9,11,24
inthandflaechen=0; // Summe der Sensoren an der Handfläche exkl.
Handmitte 6,10,12,13,15,21,23
inthandgelenk=0; // Summe der Sensoren am Handgelenk 16,17
intgesamtflaechen=0; // Summe aller Sensoren 0-senscount
inthandvibcount=0; // Summe der aktuell benötigten Vibratoren der
haptischen Ausgabe

//Anschlüsse der Sensoren
CapacitiveSensorkeys[]={CS(22),CS(23),CS(24),CS(25),CS(26),CS(27),CS(28),CS(29)
,CS(30),CS(31),CS(32),
CS(33),CS(34),CS(35),CS(36),CS(37),CS(38),CS(39),CS(40),CS(41),CS(42),CS(43),CS
(44),CS(45),CS(46),CS(47)};

//Definitionen für Fehlersuche und Programmeinstellungen- zur Aktivierung 1
setzen.
intdebugcount=0; //Zeigt Zähler der Sensoren am Ende der Eingabe
intdebugloop=0; //Zeigt Position im Loop an
intdebugflaechen=0; //Zeigt Zähler der Flächen
intbuchstabenauswertung=1; //Unterdrückt Buchstabenzuordnung bei Fehlersuche
intbuchstabenanzeige=1; //Gibt zusätzlich Buchstaben über den Serial
Monitor aus
intrueckmeldung=1; //Aktiviert die Vibratoren
intleer=0; //Definiert wann gesendet wird (Leerzeichen oder
nach jedem Buchstaben) - 0 = zeichenweise

//Definitionen der Vibratoren für Rückmeldung
constintvib1= 12;
constintvib2= 13;

voidsetup(){

```

```

//Deklaration der Vibratoren als Ausgang
pinMode(vib1, OUTPUT);
pinMode(vib2, OUTPUT);

Serial.begin(9600); //Kommunikation mit dem SerialMonitor mit einer BAUD9600

//Definitionen für RF-Modul
radio.begin(); //RF-Modul Aktiviert
radio.setRetries(150,80); //Setze (Zeit in der gesendet wird / wie oft
wiederholt wird)
radio.openWritingPipe(txAddr); //Definieren des Sendepin
radio.openReadingPipe(0,rxAddr); //Empfängeradresse
radio.stopListening(); //Deaktivieren des Abhörens

//Feineinstellung der Sensorschwellen
schwellwert[0]=450;
schwellwert[1]=400;
schwellwert[2]=300;
schwellwert[3]=230;
schwellwert[4]=300;
schwellwert[5]=300;
schwellwert[6]=300;
schwellwert[7]=250;
schwellwert[8]=500;
schwellwert[9]=300;
schwellwert[10]=200;
schwellwert[11]=400;
schwellwert[12]=300;
schwellwert[13]=200;
schwellwert[14]=200;
schwellwert[15]=500;
schwellwert[16]=250;
schwellwert[17]=300;
schwellwert[18]=200;
schwellwert[19]=100;
schwellwert[20]=400;
schwellwert[21]=400;
schwellwert[22]=500;
schwellwert[23]=500;
schwellwert[24]=400;
schwellwert[25]=500;
}

```

```

voidloop() {

    while(inputsum==0) {
        abfrage();

    }

    zaehlen();

    if(buchstabenauswertung>=1) {
        ausgabe();
    }

    if(debugcount>=1) {
        zaehlerauswaertung();
    }

    reset();
}

//-----Abfrage-----Abfrage-----Abfrage-----Abfrage-----Abfrage-----
//Dient zum Abfragen der Eingänge
voidabfrage() {

    // Abfrage aller Eingänge und Speichern/Überschreiben der Zustände (stat[i])

    for(inti=0;i<senscount;++i) {
        input[i]=keys[i].capacitiveSensor(messungen);

        if(input[i]>=schwellwert[i]) {
            stat[i]=1;
            inputsum++;//setzt inputsum +1, um bei Nichtbetätigen in Abfrage zu
bleiben

        }

        if (debugloop >= 1 ) {
            Serial.println("Ausgelesen");
        }
    }
}

//-----Abfrage-----Abfrage-----Abfrage-----Abfrage-----Abfrage-----
//-----Zählen-----Zählen-----Zählen-----Zählen-----Zählen-----
// Wichtigster Programmteil
// Dient zum "Aufnehmen" der Schreibhandlung
voidzaehlen() {

    T1=millis(); //Speichert Serverzeit
    T2=T1; //T2=T1

    while(T1-T2<=T3) {
        //Abfrage
        abfrage();
    }
}

```

```

// Zählt wie oft die Kontakte betätigt wurden
// Erkennt, ob eine "Taste" gehalten wird
// und resetet den Timer für die Eingabe

//Schleife für jeden Eingang
for(int i=0;i<senscount;++i){

    switch(stat[i]){
        //wenn stat[x] 1 dann
        case 1:
            //Solange gedrückt wird
            while(input[i]>=schwellwert[i]){

                //Neuabfrage, um Loslassen des Kontaktes und Änderungen der
                //anderen Pins wahrzunehmen
                abfrage();

            }

            //Dazuzählen und Eingang[x] reseten
            count[i]++;
            stat[i]=0;

            //Zeit der Letzten Eingabe aktualisieren
            T2=millis();

            break;}
    }

//Aktualisieren der Serverzeit
T1=millis();}

// Zählen der vordefinierten Flächen
handmitte=(count[19]+count[18]+count[14]);
fingerspitzen=(count[1]+count[3]+count[7]+count[20]+count[25]);
fingerflaechen=(count[0]+count[4]+count[9]+count[11]+count[24]);
handflaechen=(count[16]+count[12]+count[13]+count[6]+count[10]+
               count[15]+count[17]+count[21]+count[23]);
handgelenk=(count[16]+count[17]);

for(int i=0;i<senscount;++i){
    gesamtflaechen+=count[i];
}

if(debugflaechen>=1){
    Serial.println("");
    Serial.print("Handmitte:");
    Serial.print("\t");
    Serial.println(handmitte);
    Serial.print("Fingerspitzen:");
    Serial.print("\t");
    Serial.println(fingerspitzen);
    Serial.print("Fingerflächen:");
    Serial.print("\t");
    Serial.println(fingerflaechen);
    Serial.print("Handfläche:");
    Serial.print("\t");
    Serial.println(handflaechen);
    Serial.print("Gesamtfläche:");
    Serial.print("\t");
    Serial.println(gesamtflaechen);
    Serial.print("Handgelenk:");
    Serial.print("\t");
    Serial.println(handgelenk);
}

```

```

    }
        if(debugloop>=1) {
            Serial.println("Gezählt");
        }
    }

//-----Zählen-----Zählen-----Zählen-----Zählen-----Zählen-----

//-----Ausgabe-----Ausgabe-----Ausgabe-----Ausgabe-----Ausgabe-----

//Dient der Auswertung und Zuordnung der Buchstaben
voidausgabe() {

    //Buchstabe
    //If(Zählerbedingung für Zuordnung)
    // If(Ausgabe im Serial erwünscht)
    // Ausgabe
    //Schreibe Buchstaben in Array texttx an der Stelle k
    //Stelle k um eins nach rechts setzen für den nächsten Buchstaben
    //handvibcount = handvibcount + benötigte Vibratoren der haptischen Ausgabe
    //Starte void vibrieren()

//Umschaltung Einzelzeichen senden - wortweise senden
if(count[25]>=1&&count[17]>=1&&count[1]>=1) {

    switch(leer){

        case0:
            leer=1;

            Serial.println("Wort senden");
            multivibration(2,700);

            reset();

            break;

        case1:
            leer=0;

            Serial.println("Einzelzeichen senden");
            multivibration(2,200);

            reset();

            break;

    }
}

//A
if(count[1]==1&&fingerflaechen==0&&count[0]==0&&count[2]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("A");
    }
    texttx[k]='A';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//B

```

```

if(count[4]>=1&&count[22]==0&&fingerflaechen<=1&& count[5]==0){
    if(buchstabenauswertung==1){
        Serial.println("B");
    }
    texttx[k]='B';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//C
if((count[16]==1||count[17]==1)&&handgelenk<=2&&count[9]==0){
    if(buchstabenauswertung==1){
        Serial.println("C");
    }
    texttx[k]='C';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//D
if(count[9]>=1&&count[22]==0&&fingerflaechen<=1&& count[5]==0&&
count[8]==0&&count[15]==0){
    if(buchstabenauswertung==1){
        Serial.println("D");
    }
    texttx[k]='D';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//E
if(count[3]>=1&&count[5]==0&&count[4]==0&&count[5]==0 && count[8]==0){
    if(buchstabenauswertung==1){
        Serial.println("E");
    }
    texttx[k]='E';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//F
if((count[8]>=1 || count[7]>=1 ) &&
(count[3]>=1 || count[5]>=1)&&
count[22]==0){

    if(buchstabenauswertung==1){
        Serial.println("F");
    }
    texttx[k]='F';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

```

```

//G
if(count[11]>=1&&fingerflaechen<=1&&count[23]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("G");
    }
    texttx[k]='G';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//H
if(count[24]>=1&&count[22]==0&&fingerflaechen<=1&& count[5]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("H");
    }
    texttx[k]='H';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//I
if(count[7]==1&&fingerflaechen==0&&count[8]==0&&count[5]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("I");
    }
    texttx[k]='I';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//J
if(count[7]>1&&fingerflaechen==0&&count[8]==0&&count[5]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("J");
    }
    texttx[k]='J';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//K
if(count[6]>=1&&(count[19]>=1||count[10])&&count[21]>=1&&handflaeche<=3) {
    if(buchstabenauswertung==1) {
        Serial.println("K");
    }
    texttx[k]='K';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

```

```

//L
if(count[9]>=1  && (count[15]  >=1  /*|| handmitte  >=1 */)&&
    count[5]==0&&handgelenk==0) {

    if(buchstabenauswertung==1) {
        Serial.println("L");
    }
    texttx[k]='L';
    k=k+1;
    handvibcount+=5;
    vibrieren();
}

//M
if(count[21]>=1  &&handflaeche==1&&fingerflaechen==0&&count[6]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("M");
    }
    texttx[k]='M';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//N
if(count[6]>=1  &&handflaeche==1&&fingerflaechen==0  ) {
    if(buchstabenauswertung==1) {
        Serial.println("N");
    }
    texttx[k]='N';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//O
if(count[20]==1&&fingerflaechen==0&&count[11]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("O");
    }
    texttx[k]='O';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//P
if(count[5]>=1&& count[8]==0&&count[22]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("P");
    }
    texttx[k]='P';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

```

```

//Q
if(count[22]>=1 &&count[5]==0){
    if(buchstabenauswertung==1){
        Serial.println("Q");
    }
    texttx[k]='Q';
    k=k+1;
    handvibcount+=3;
    vibrieren();
}

//R
if((handmitte>=1||count[15]>=1) &&
    (count[13]>=1||count[12]>=1)&&
    count[21]==0&& count[23]>=1&&
    count[6]==0&&count[9]==0){

    if(buchstabenauswertung==1){
        Serial.println("R");
    }
    texttx[k]='R';
    k=k+1;
    handvibcount+=3;
    vibrieren();
}

//S
if(handflaeche>=6){
    if(buchstabenauswertung==1){
        Serial.println("S");
    }
    texttx[k]='S';
    k=k+1;
    handvibcount+=8;
    vibrieren();
}

//T
if(count[0]>=1&&count[2]==0){
    if(buchstabenauswertung==1){
        Serial.println("T");
    }
    texttx[k]='T';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//U
if(count[25]==1&&fingerflaechen==0&&count[22]==0){
    if(buchstabenauswertung==1){
        Serial.println("U");
    }
    texttx[k]='U';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

```

```

//V
if(count[12]==1&&count[21]==0&&count[9]==0&&count[23]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("V");
    }
    texttx[k]='V';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

//W
if(count[12]>1&&count[21]==0&&count[9]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("W");
    }
    texttx[k]='W';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//X
if(count[16]>=2||count[17]>=2) {
    if(buchstabenauswertung==1) {
        Serial.println("X");
    }
    texttx[k]='X';
    k=k+1;
    handvibcount+=3;
    vibrieren();
}

//Y
if(fingerflaechen>2) {
    if(buchstabenauswertung==1) {
        Serial.println("Y");
    }
    texttx[k]='Y';
    k=k+1;
    handvibcount+=4;
    vibrieren();
}

//Z
if(count[12]==1&&(count[15]>=1||handmitte>=1)&&
    count[21]>=1 &&count[6]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("Z");
    }
    texttx[k]='Z';
    k=k+1;
    handvibcount+=4;
    vibrieren();
}

```

```

//Ä
if(count[1]>1&&fingerflaechen==0&&count[0]==0&&count[2]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("AE");
    }
    texttx[k]='1';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//Ö
if(count[20]>1&&fingerflaechen==0&&count[11]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("OE");
    }
    texttx[k]='2';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//Ü
if(count[25]>1&&fingerflaechen==0&&count[22]==0) {
    if(buchstabenauswertung==1) {
        Serial.println("UE");
    }
    texttx[k]='3';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//CH
if(count[21]>=1&&count[12]>=1&&count[6]>=1&&count[23]>=1&&
    handmitte>=1&&count[15]==0&&handflaeche<6) {

    if(buchstabenauswertung==1) {
        Serial.println("CH");
    }
    texttx[k]='4';
    k=k+1;
    handvibcount+=6;
    vibrieren();
}

//SCH
if((count[22]>=1||count[25]>=1)&&
    (count[5]>=1||count[3]>=1)&&
    count[8]==0) {

    if(buchstabenauswertung==1) {
        Serial.println("SCH");
    }
    texttx[k]='5';
    k=k+1;
    handvibcount+=1;
    vibrieren();
}

```

```

//ST
if(count[2]>=1){
    if(buchstabenauswertung==1){
        Serial.println("ST");
    }
    texttx[k]='6';
    k=k+1;
    handvibcount+=2;
    vibrieren();
}

//" " - Leerzeichen
if(count[11]>=1&&count[23]>=1&&count[21]>=1){
    if(buchstabenauswertung==1){
        Serial.println("");
        Serial.println("Leerzeichen");
    }
    texttx[k]=' ';
    k=k+1;
    leerzeichendelay=3000;
    vibrieren();

    if(leer==1){
        senden(); //Sendet an RF-Empfangsmodul
    }
}

//Ausgabe Position im Loop - Debugging only
if(debugloop>=1) {
    Serial.println("Ausgegeben");
}

}

//-----Ausgabe-----Ausgabe-----Ausgabe-----Ausgabe-----Ausgabe-----
//-----Vibrieren-----Vibrieren-----Vibrieren-----Vibrieren-----Vibrieren-----

voidvibrieren(){

if(rueckmeldung>=1){
digitalWrite(vib1,HIGH);
digitalWrite(vib2,HIGH);
delay(vibration);
digitalWrite(vib1,LOW);
digitalWrite(vib2,LOW);
}

//Ausgabe nach jeder erfolgreichen Eingabe
//Wenn leer = 1 wird nur nach jedem Leerzeichen die Information an das RF-
Modul gesendet
//Routine Senden wird direkt in void ausgabe() aufgerufen
if(leer==0){
    senden();
}
}

//-----Vibrieren-----Vibrieren-----Vibrieren-----Vibrieren-----Vibrieren-----

```

```

//-----Reset-----Reset-----Reset-----Reset-----Reset-----
//Dient zum Reseten der Counter
voidreset(){

    //Reset counter nach Ausgabe eines Buchstaben
    for(inti=0;i<senscount;++i){
        count[i]=0;
        input[i]=0;
    }

    handmitte=0;
    fingerspitzen=0;
    fingerflaechen=0;
    handflaechen=0;
    handgelenk=0;
    gesamtflaechen=0;
    inputsum=0;
    handvibcount=0;
    T4=0;
    leerzeichendelay=0;

    //Automatische Rekalibration
    if(debugloop>=1){
        Serial.println("kalibrieren");
    }
    voidreset_CS_AutoCal();

    if(debugloop>=1) {
        Serial.println("Reset");
    }
}

//-----Reset-----Reset-----Reset-----Reset-----Reset-----
//-----Senden-----Senden-----Senden-----Senden-----Senden-----
voidsenden(){

    if(debugloop>=1) {
        Serial.print("Senden - Beginn");
    }

    radio.write(&texttx,sizeof(texttx)); //Senden (texttx-Array, mit der
größe von texttx)
    empfangen();
    //Reset Array
    for(inti=0;i<sizeof(texttx)-1;i++)
    {
        texttx[i]={0};
    }
    //Reset Arraypositionszähler
    k=0;
    delay(10);

    if(debugloop>=1) {
        Serial.println("Senden - Abgeschlossen");
    }
}

//-----Senden-----Senden-----Senden-----Senden-----Senden-----

```

```

//-----Empfangen-----Empfangen-----Empfangen-----Empfangen-----Empfangen
void empfangen() {

    if(debugloop>=1) {
        Serial.println("Empfangen");
    }

    int p=0;          //Fehler Variable

    radio.startListening(); //Empfangen aktivieren
    T1=millis();      //Aktualisiert Serverzeit

    //T4 = Wartezeit (3S + Display überschreiben +(benötigte Vibratorenanzahl *
    Vibrationszeit) + (Buchstaben*Buchstabendelay)+ Zeit für Leerzeichen

    T4=T1+3000+displayrefresh+(handvibcount*handvibedelay)+(k*nextletterdelay)+1
    eerzeichendelay;

    while(T1<T4) {

        if(radio.available()){

            radio.read(&textrx,sizeof(textrx));
            radio.stopListening();
            Serial.print("Empfangen:");
            Serial.println(textrx);

            for(int e=0;e<sizeof(texttx)-1;e++){

                if(texttx[e]!=textrx[e]){
                    p=p+1;
                }
            }

            if(p!=0){

                if(debugloop>=1) {
                    Serial.println("Fehler - Übertragungsinhalt");
                }

                multivibration(3,700);
                break;
            }

            if(p==0){

                if(debugloop>=1) {
                    Serial.println("Feedbackbox fertig");
                }

                multivibration(1,700);
                break;
            }
        }

        T1=millis(); //Aktualisiere Zeit
    }
}

```

```

        if (T1>=T4) {

        if(debugloop>=1)  {
        Serial.println("Fehler - Übertragungszeit");
        }

        multivibration(3,200);

        radio.stopListening();
    }

}
//-----Empfangen-----Empfangen-----Empfangen-----Empfangen-----Empfangen-----
//-----Mulivibration-----Mulivibration-----Mulivibration-----Mulivibration-----

voidmultivibration(intanzahl,intdauer) {

    for(inti=0;i<anzahl;++i) {
        digitalWrite(vib1,HIGH);
        digitalWrite(vib2,HIGH);
        delay(dauer);
        digitalWrite(vib1,LOW);
        digitalWrite(vib2,LOW);

        if(i<anzahl-1) {
            delay(dauer);
        }

    }
    k=0;
}

//-----Mulivibration-----Mulivibration-----Mulivibration-----Mulivibration-----
//-----Zählerauswertung-----Zählerauswertung-----Zählerauswertung-----

voidzaehlerauswaertung() {

    for(inti=0;i<senscount;++i) {

        if(count[i]>0) {
            Serial.println("");
            Serial.print("Sensor:");
            Serial.print(i);
            Serial.print("\t");
            Serial.print(count[i]);
        }

    }
    Serial.println("");
}

//-----Zählerauswertung-----Zählerauswertung-----Zählerauswertung-----

```

Hinweis: Dieser Code findet sich im Anhang in digitaler Form wieder.

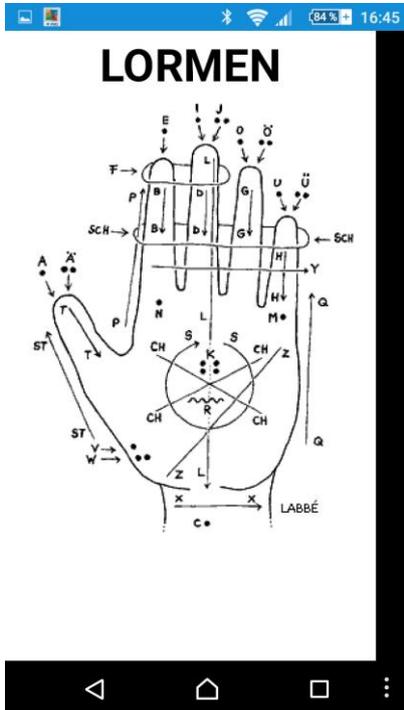


Abbildung 47: Startbild Android App



Abbildung 49: Verbindung Android App



Abbildung 48: Tastatur Android App

Wir haben die Android-App „Lormen“ entwickelt, um eine Eingabemöglichkeit für nicht taubblinde Menschen zu schaffen. Mit der App kann man nun durch die wortweise Eingabe Nachrichten an Taubblinde schicken. Diese werden über die haptische Ausgabe (Handschuh mit Vibratoren) für den taubblinden Menschen ausgegeben. Das Layout der App wurde so simpel wie möglich gestaltet, damit eine einfache und intuitive Nutzung gewährleistet werden kann. Die Bedienung der App wird in Kapitel [14.2 -Bedienung –Eingabe via Bluetooth](#) noch genau beschrieben. Im Anhang ist das Programm zudem in digitaler Form beigelegt.

9.1 FLUSSDIAGRAMM

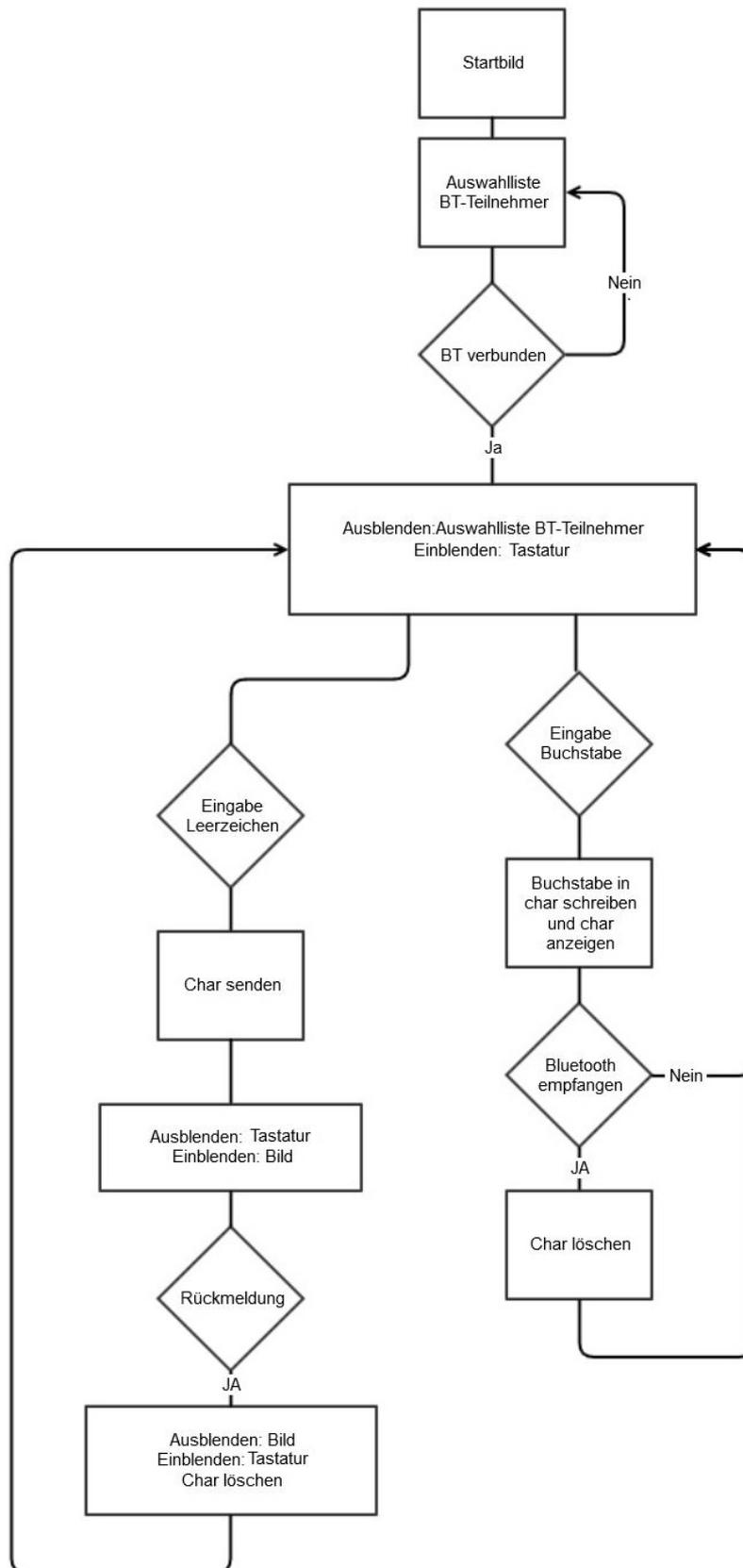


Abbildung 50: Flussdiagramm Android APP "Lormen"

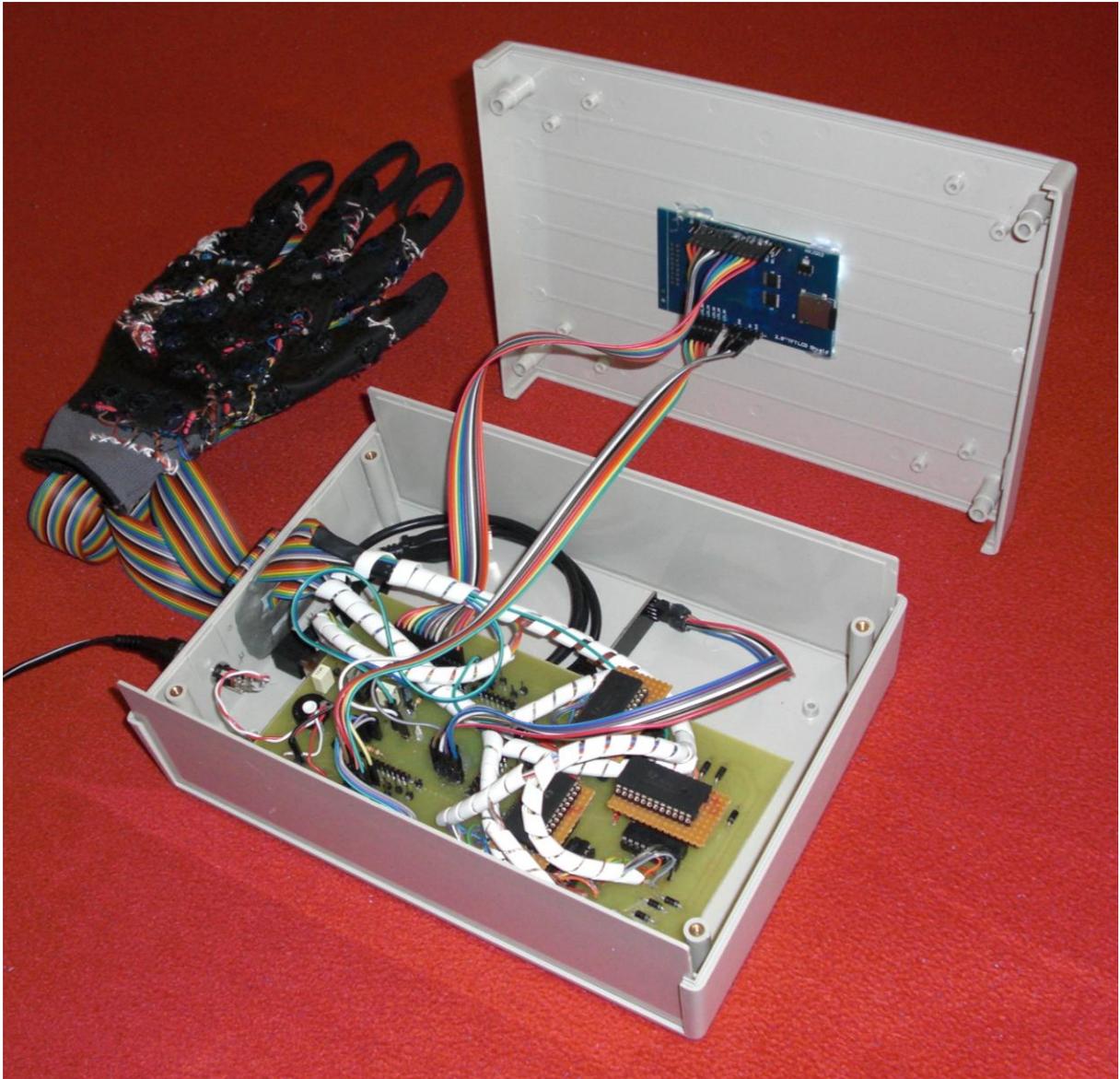


Abbildung 51: Feedback - Box offen

Die Feedback-Box besteht aus folgenden Komponenten:

Die Hauptkomponente ist der Mikrocontroller, der unser Gehirn der Box darstellt. Hier werden alle Informationen empfangen, gesendet, verarbeitet und ausgegeben.

Sowohl das Radiofrequenz-Modul, als auch das Bluetooth-Modul sind im Gehäuse versteckt.

Das PCB (Print Circuit Board) ist mit dem Mikrocontroller verbunden und dient als Träger für eine Schaltung mit mehreren Demultiplexern und Treiberstufen. Zudem dient es dazu, alle elektrischen Komponenten der Feedback-Box mit dem Mikrocontroller zu verbinden. Des Weiteren wird auf dem PCB auch die Spannung der Vibratoren geregelt.

Die Spannungsversorgung ist mittels einer Buchse, die sich am Gehäuse befindet, an das PCB angeschlossen.

Flachbandkabel verbinden das PCB mit den am Gehäuse befindlichen Buchsen für den Handschuh.

Das Display befindet sich an der Oberseite des Gehäuses.

10.1 VORSTUDIEN

Um eine optimale Ausgabe zu ermöglichen, wurden verschiedene Möglichkeiten evaluiert. Für die praktische Umsetzung wurde zuerst ein zweizeiliges LC-Display mit jeweils 16 Zeichen pro Zeile herangezogen. Dieses wurde jedoch den Ansprüchen der Ausgabe nicht gerecht, da man keine längeren Sätze (mehr als 32 Zeichen) auf einmal ausgeben konnte.

Die weitere Wahl fiel auf ein 2,8“ TFT-LCD-Shield, da dieses eine Anpassung der Schriftgröße erlaubt und somit auch mehrzeilige Texte problemlos darstellen kann.

Des Weiteren mussten wir uns überlegen wie wir die haptische Ausgabe gestalten wollen. Nach längerer Suche haben wir Minivibratoren in einem passenden Format gefunden um sie an einem Handschuh zu befestigen.

10.1.1 OPTISCHE AUSGABE



Abbildung 52: Optische Ausgabe Display

Als optische Ausgabe dient ein 2,8“ TFT-LCD. Das LCD wird über den Mikrocontroller angesteuert und ermöglicht uns sowohl einen Text auf das Display auszugeben als auch ein Hintergrundbild zu laden.

Das Bild wird Zeilen und Spaltenweise in das LCD geladen. Die Bilder sind auf einer SD-Karte gespeichert und werden von dieser in das Display geladen.



Abbildung 53: Haptische Ausgabe Handschuh

Als haptische Ausgabe dient ein handelsüblicher Arbeitshandschuh an dem Minivibratoren angenäht wurden. Die Vibratoren sind so am Handschuh platziert, dass sie durch ihre Vibration oder die Abfolge mehrerer Vibrationen und auch verschiedener Vibratoren, ein Schema im Lorm-Alphabet ergeben.

Die Vibratoren sind mittels zweier Flachbandkabel zum Mikrocontroller verbunden. Die Kabel sind an den Vibrator-Kontakten angelötet und auf der Gegenseite auf einen Stecker gekrimpt. Der Stecker lässt sich an die Feedback-Box anschließen. Um den Halt der Kabel am Handschuh zu gewährleisten, sind die Kabel ebenfalls am Handschuh vernäht und haben eine eigene Zugentlastung.

Um den Handschuh optimal ausziehen zu können, sind an den Fingerspitzen Gummibandschlaufen befestigt. Diese dienen auch dem Schutz der Lötstellen und der Verkabelung.

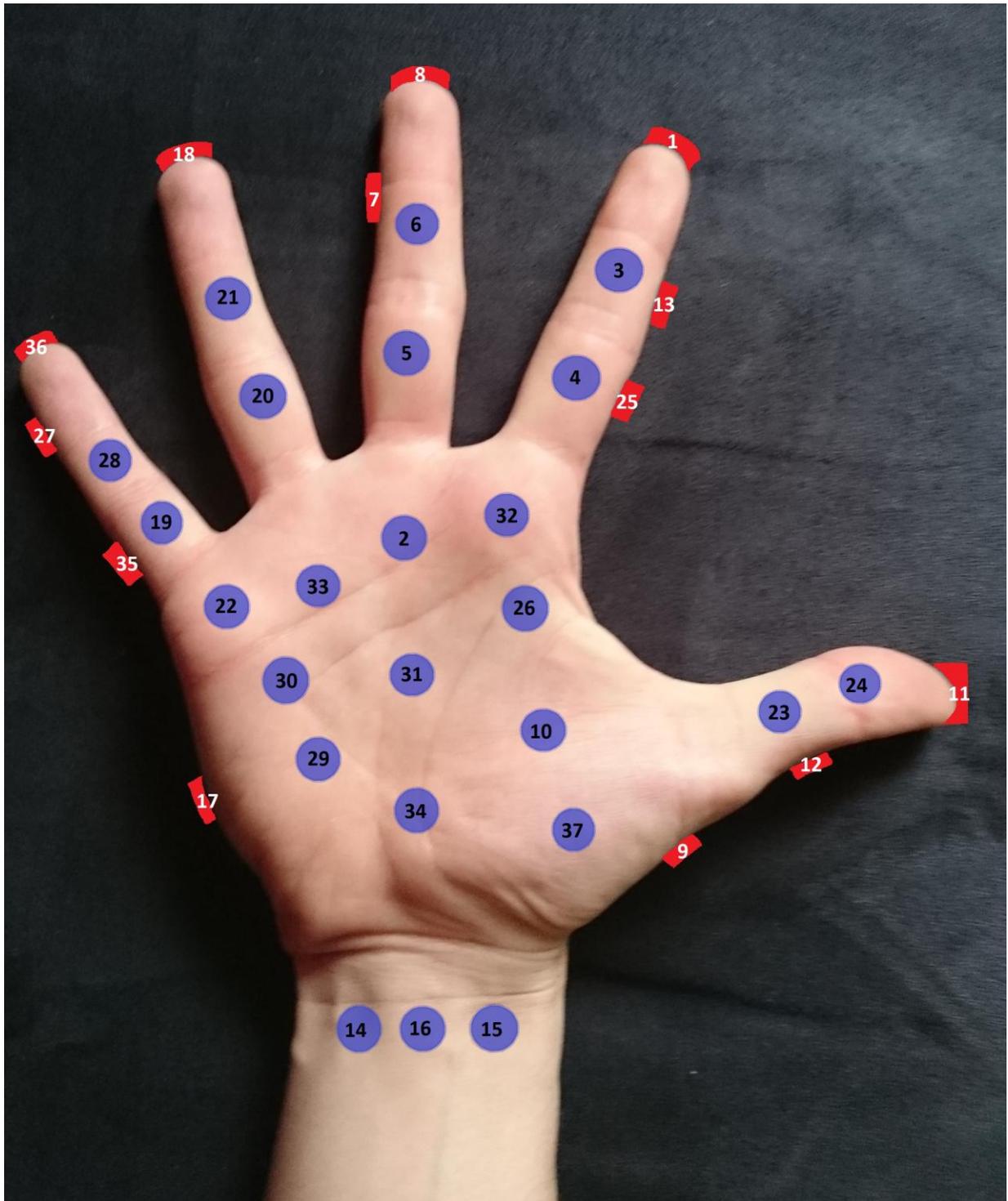


Abbildung 54: Anordnung der Vibratoren auf der Hand

Die Ansteuerung des Buchstaben K erfolgt im Programm mit der Bezeichnung Vibrator38[] und steuert Vibrator 2,26,31 und 33 zeitgleich an.

Die Ansteuerung des Zeichens Sch erfolgt im Programm mit der Bezeichnung Vibrator39[] und steuert Vibrator 13, 25, 27 und 35 zeitgleich an.

Die Ansteuerung des Buchstaben F erfolgt im Programm mit der Bezeichnung Vibrator40[] und steuert Vibrator 7 und 13 zeitgleich an.

Technische Details (adafruit, 2015)

Dimension: 10mm Durchmesser, 2.7mm Dick

Spannung: 2V - 5V

Spannung[V]	Stromaufnahme[mA]
5	100
4	80
3	60
2	40

Umdrehungen: 11000 bei 5V

Gewicht: 0.9 Gramm

10.2 GRUNDLAGEN

Wir haben eine Lösung gesucht, mit der wir es schaffen konnten, auf möglichst günstige und effiziente Weise unsere Ausgänge zu erweitern. Hinzu kam außerdem die Notwendigkeit einer einstellbaren Spannungsregelung um die Vibratoren zu regulieren.

10.2.1 DEMULTIPLEXER

Mit der Hilfe von drei Demultiplexern haben wir es geschafft, durch 8 Ausgänge des Mikrocontrollers 48 mögliche Vibratoren anzusteuern. Je nachdem welches Binärsignal wir nun mit dem Mikrocontroller auf die Ausgänge legten, konnten wir nun einen der 48 erweiterten Ausgänge ansprechen. Leider war aber die Stromabgabe des Demultiplexers alleine nicht ausreichend, um unsere Vibratoren zu versorgen. Was noch dazu kam ist, dass es uns mit diesen Ausgängen nicht möglich war, mehrere Vibratoren gleichzeitig anzusteuern.

10.2.2 TREIBERSTUFE

Mit Hilfe der Treiberstufe war es uns möglich, die benötigte Leistung für die Vibratoren zur Verfügung zu stellen. Im finalen Prototyp versorgen wir die Vibratoren mit einer einstellbaren Gleichspannung und schalten sie über die Treiberstufen gegen Masse.

10.2.3 EINSTELLBARER SPANNUNGSREGLER

Der einstellbare Spannungsregler dient dazu, die Vibratoren so einstellen zu können, dass eine optimale, haptische Wahrnehmung gewährleistet wird. Die Vibrator-Intensität hängt von der zur Verfügung gestellten Spannung maßgeblich ab. Die Stromaufnahme hängt ebenfalls von dem Spannungspegel ab.

10.2.4 ANSTEUERUNG MEHERER VIBRATOREN

Wir bekommen also ein 8 Bit-Signal, dieses schaltet nun nur einen der Demultiplexer-Ausgänge durch. Dann steuert ein Ausgang über die Treiberstufe auch nur einen Vibrator an. Um mehrere Vibratoren ansteuern zu können, haben wir eine Schaltung entworfen, mit der wir mit einem Ausgang mehrere Vibratoren ansteuern konnten.

Wir haben dafür mehre Vibratoren auf einen Ausgang der Treiberstufe angeschlossen. Um jetzt zu verhindern, dass bei dem Ansteuern eines einzelnen Vibrators die anderen mitgeschaltet werden, mussten wir Dioden mit in die Schaltung einbauen.

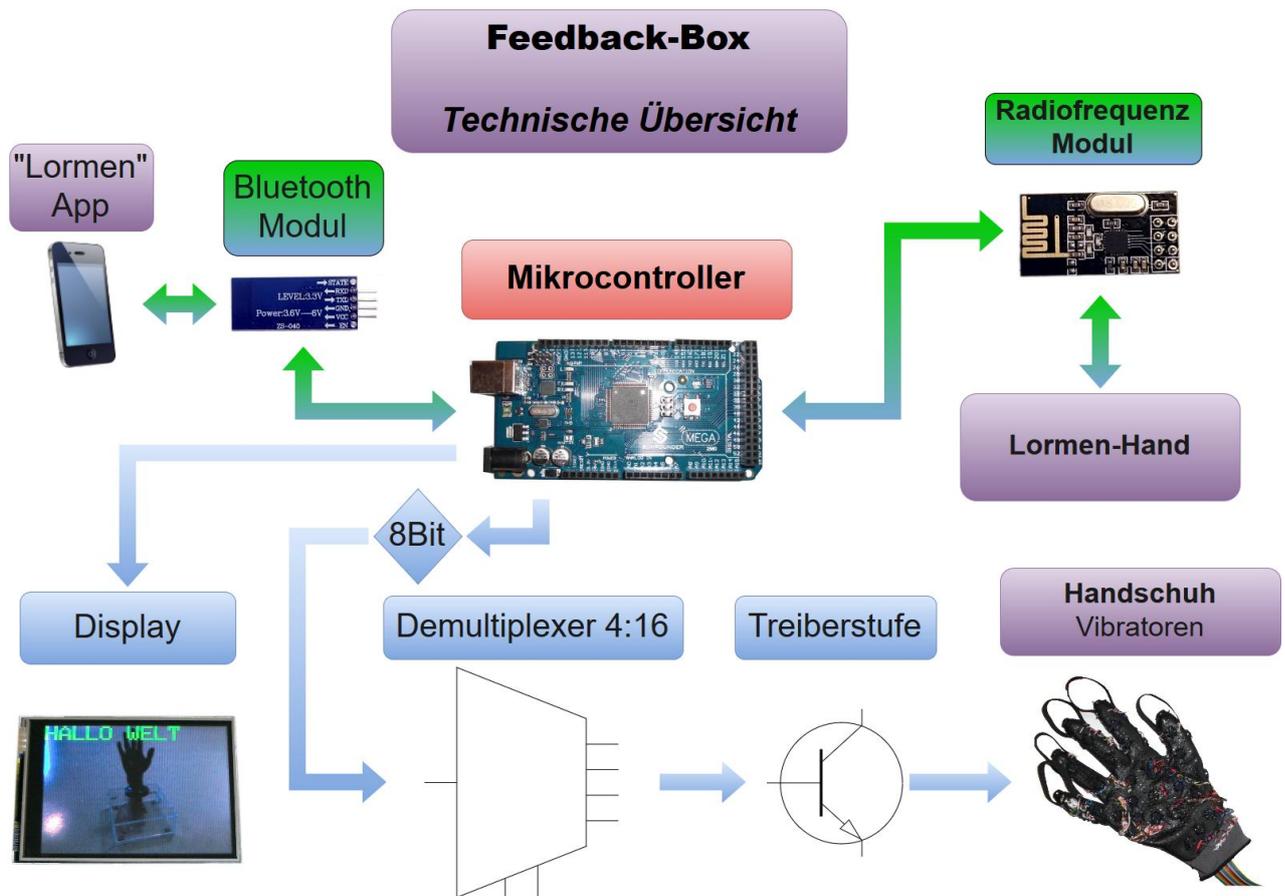


Abbildung 55: Feedback-Box Technische Übersicht

Nach dem Anschließen der Feedback-Box an ihre Spannungsversorgung beginnt sich das Display zu initialisieren. Sofern die SD-Karte gefunden werden kann, wird auch das Startbild in das Display geladen.

Sobald wir einen String von der Feedback-Box oder von der Lormen-App erhalten, beginnt die Verarbeitung der erhaltenen Informationen. Das Display wird nun auf den Teilnehmer angepasst, der gesendet hat. Der erhaltene String wird nun an jeder Stelle durchlaufen und ausgewertet. Je nachdem welches Zeichen er enthält, wird nun eine passende Abfolge ausgeführt.

Wird nun ein Buchstabe erkannt, prüft das Programm des Mikrocontrollers um welchen Buchstaben es sich handelt. In weiterer Folge wird für das Zeichen ein passendes 8-Bit-Signal erstellt und mit Hilfe der Ausgänge des Mikrocontrollers an die Demultiplexer geschickt. Die Demultiplexer schalten nun einen der erweiterten Ausgänge durch welcher wiederum einen passenden Eingang der Treiberstufen ansteuert.

Mit der Treiberstufe wird der gewünschte Vibrator gegen GND geschaltet und beginnt zu vibrieren.

Für Buchstaben die eine Abfolge mehrerer Vibratoren benötigen, wird diese Abfolge auch im Programm ermittelt und umgesetzt.

Der Wechsel zwischen Lormen-Hand und Handy-App erfolgt automatisch. Hierfür wird die Information von dem Teilnehmer ausgewertet der gesendet hat. Die Ausgabe der Lormen-Hand wird dabei bevorzugt. Sollten also Lormen-Hand und Handy-App gleichzeitig zur Eingabe verwendet werden, wird nur die Information von der Lormen-Hand angezeigt, während der Eingegebene Text auf der Handy-App wieder zurückgesetzt wird.

10.4 SCHALTPLAN

Der Schaltplan wird in digitaler Form im Anhang zur Verfügung gestellt. Er wurde mit Eagle erstellt und ist zugleich auch die Vorlage für die Platine.

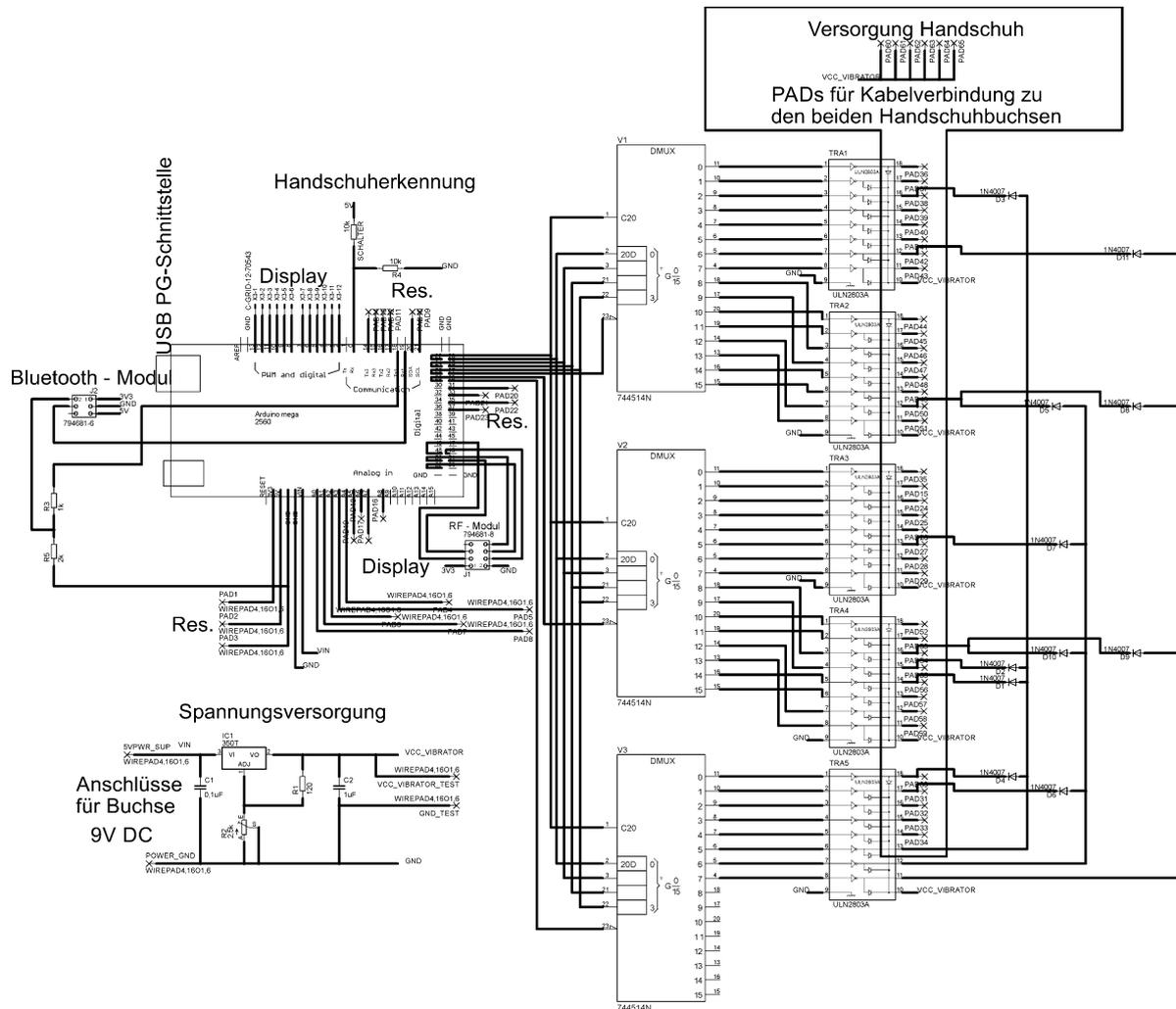


Abbildung 56: Schaltplan Feedback-Box – Gesamtansicht

Spannungsversorgung

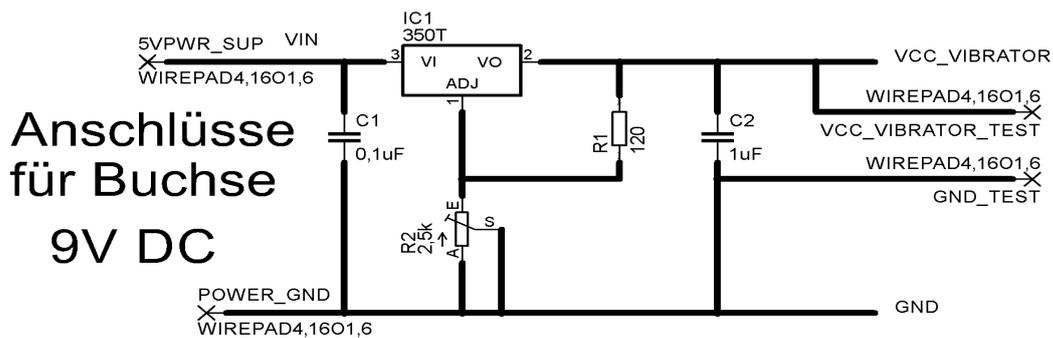


Abbildung 57: Schaltplan Feedback-Box – Detailansicht Spannungsversorgung

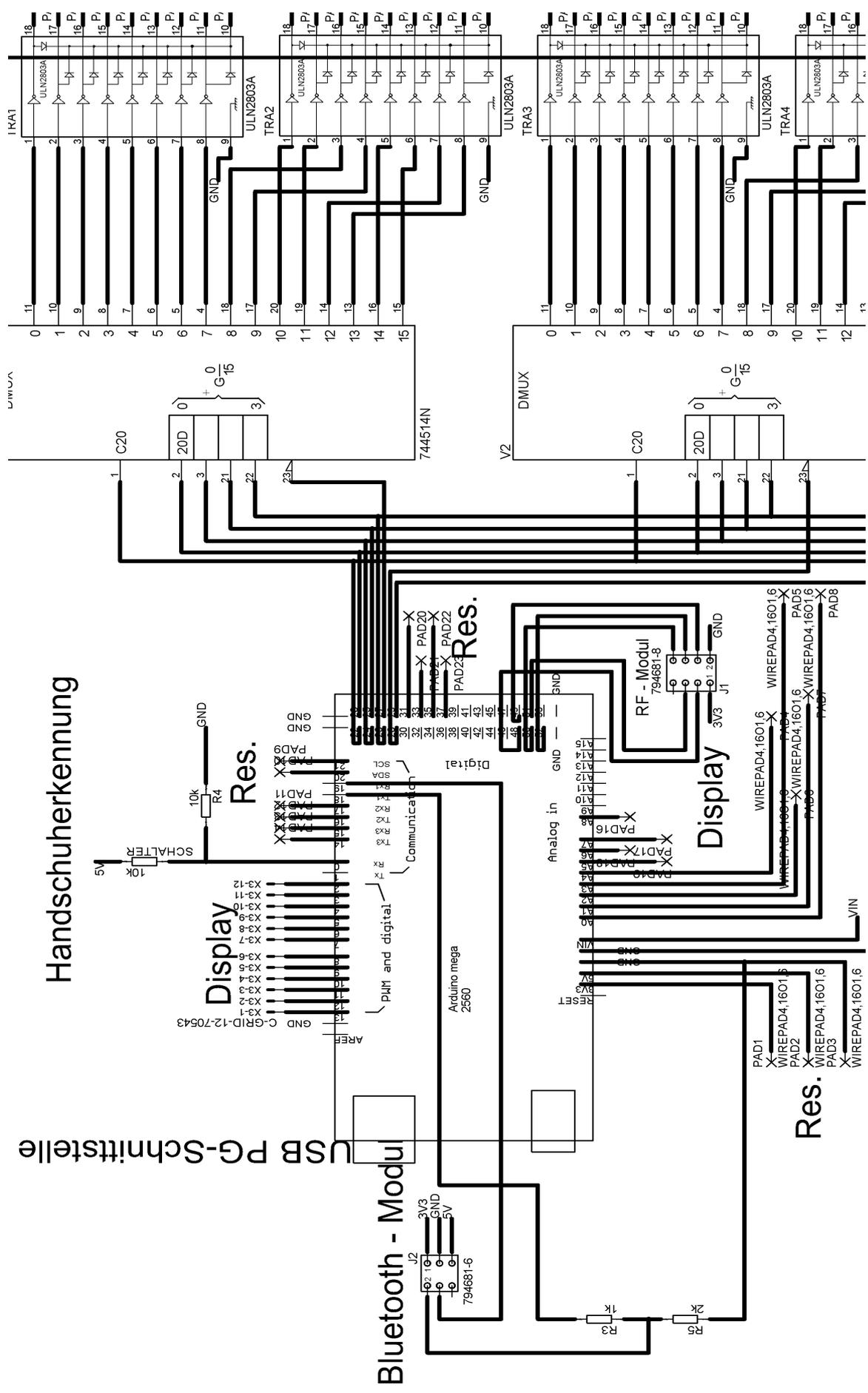


Abbildung 58: Schaltplan Feedback-Box – Detailansicht Mikrocontroller, RF- Modul und Bluetooth-Modul

erkennung

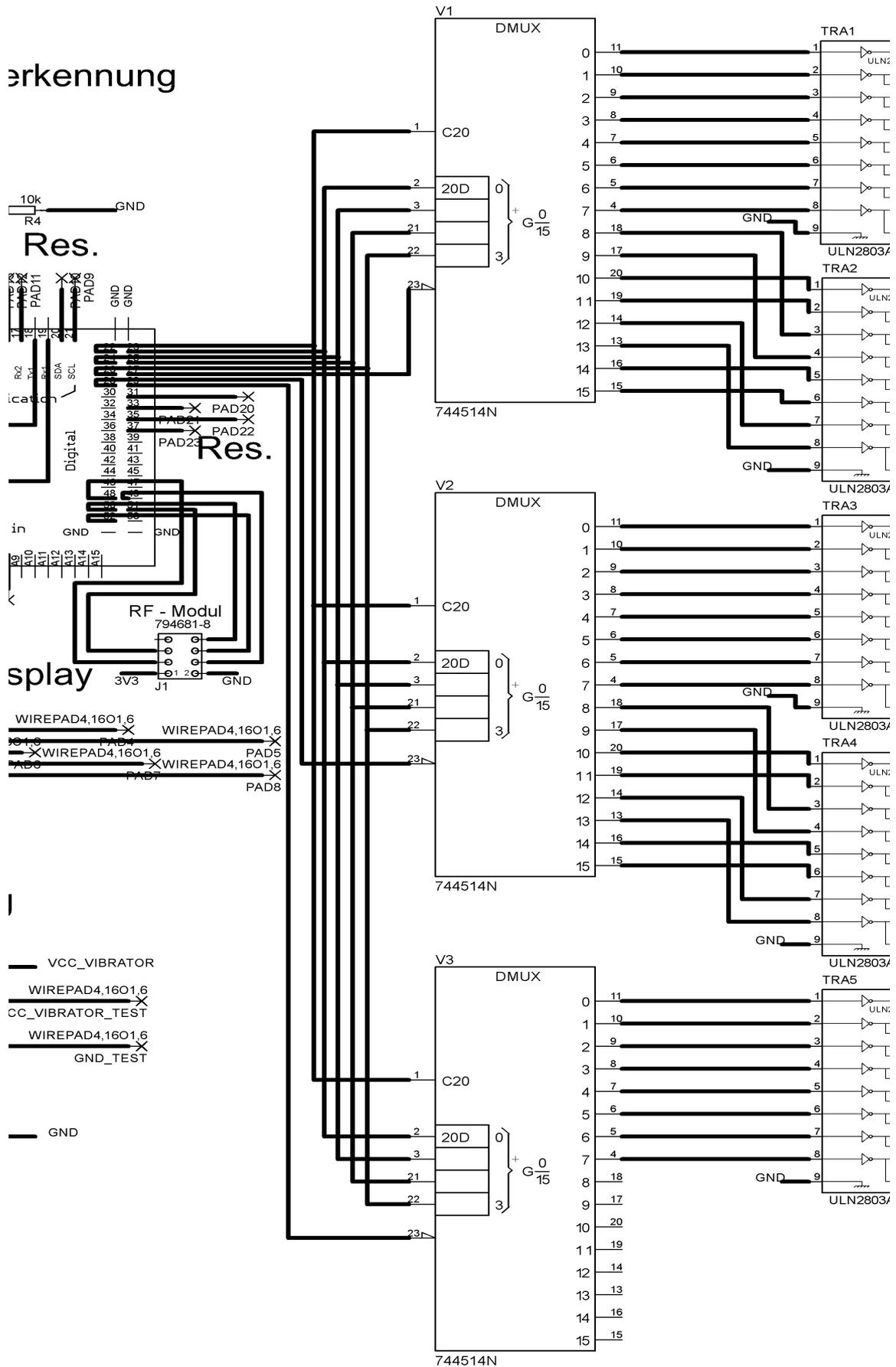


Abbildung 59: Schaltplan Feedback-Box – Detailansicht Demultiplexer

Versorgung Handschuh

PADs für Kabelverbindung zu den beiden Handschuhbuchsen

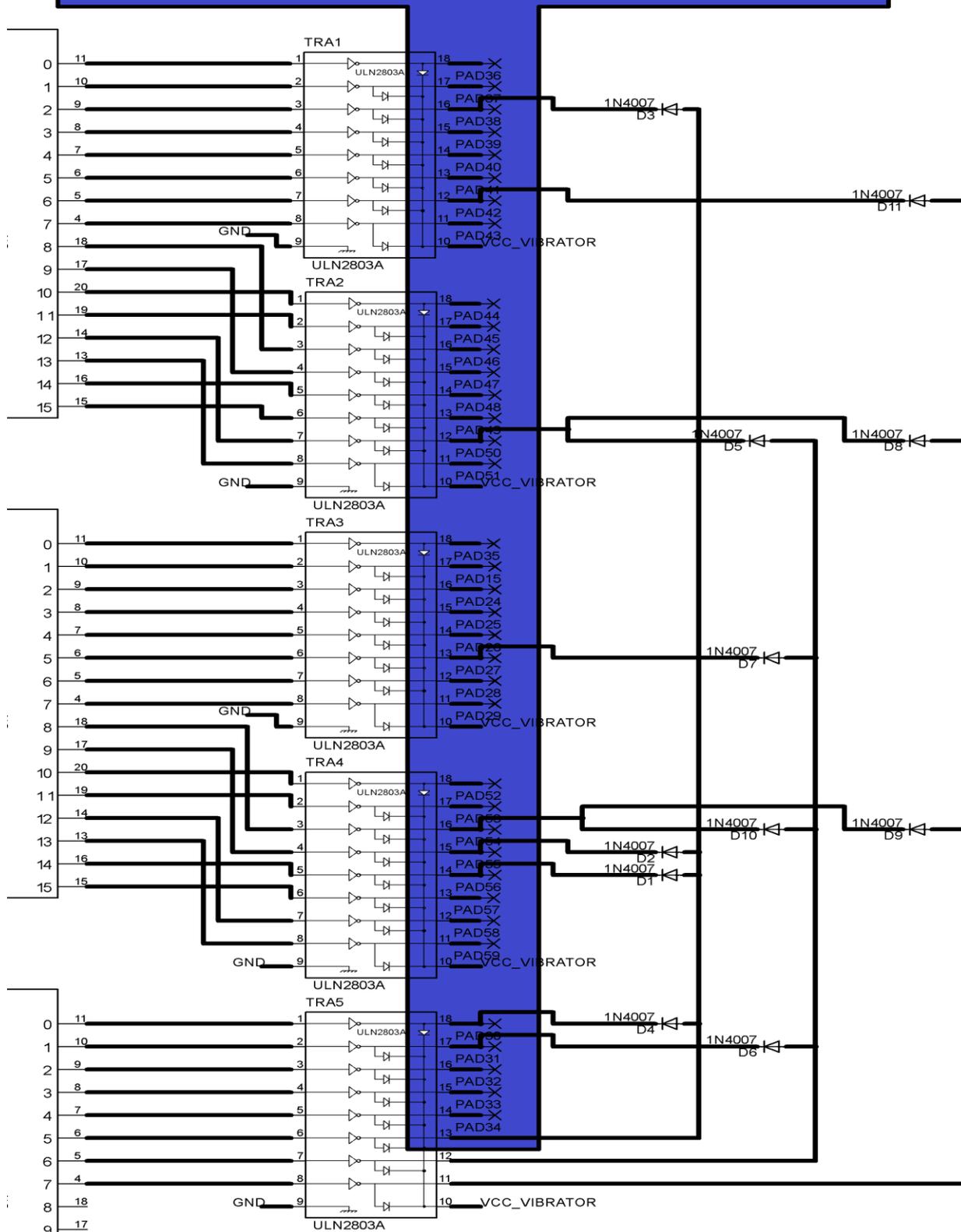


Abbildung 60: Schaltplan Feedback-Box – Detailansicht Treiberstufen und Verbindungen für Handschuh

10.5 PCB

Die Platine wurde mit Eagle erstellt. Digitalisiert findet sich der PCB auch im Anhang. Die Bemaßung der Platine ist in Millimeter angegeben. Die graue Fläche stellt die Kontur des Mikrocontrollers dar. Die Platine wird in die Buchsen des Mikrocontrollers gesteckt.

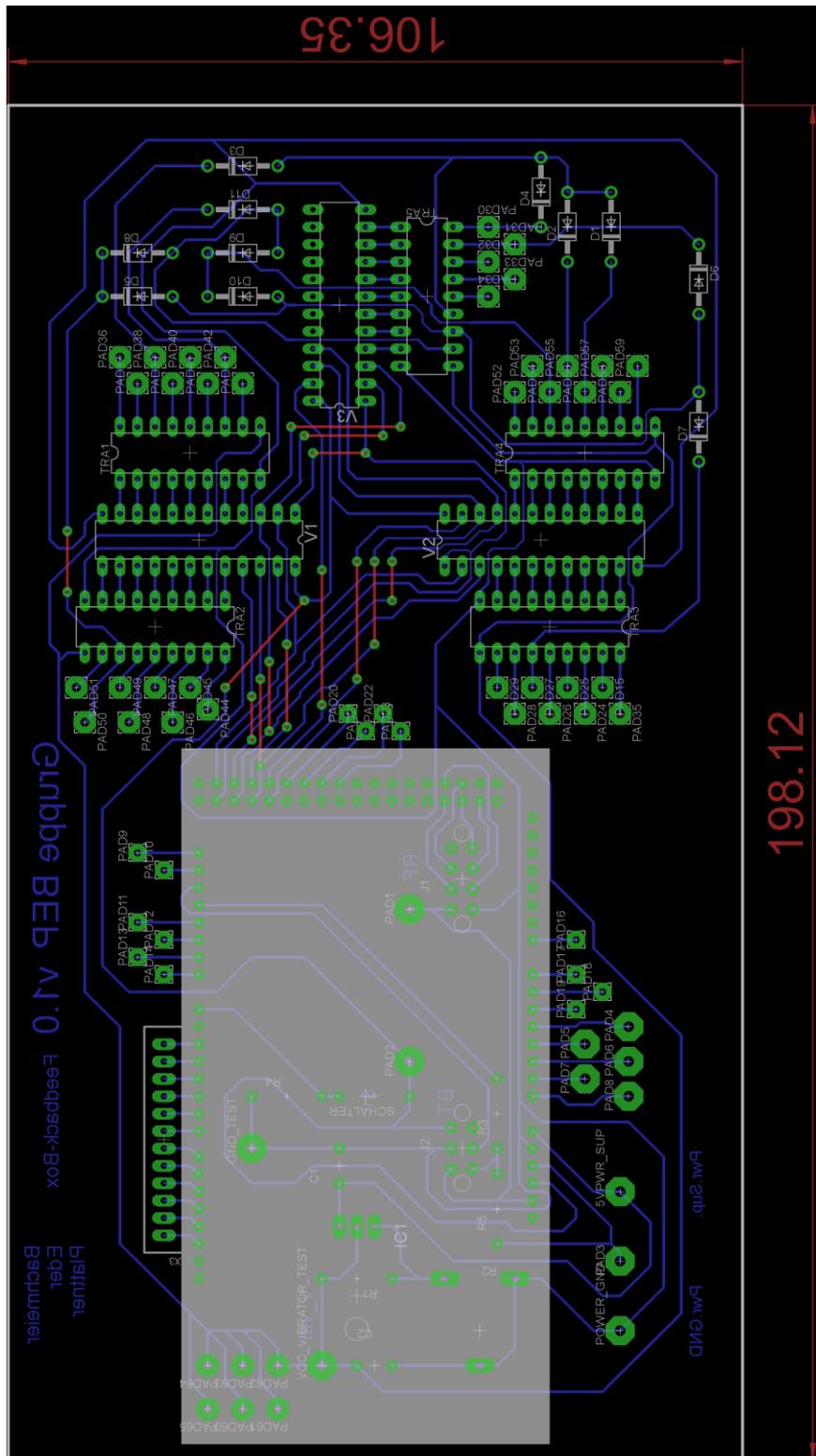


Abbildung 61: Leiterplatte Feedback-Box, Bauteilansicht (THT, Bauteile oben)

10.6 HERSTELLUNG

Bei der Herstellung mussten wir etliche Versuchsaufbauten machen, um die beste Funktionsweise und den besten Aufbau zu ermitteln.

Nachdem wir unsere Versuchsaufbauten vollständig hatten und mit der Funktion zufrieden waren, haben wir eine PCB zu entwerfen die alle Versuche in sich vereint.

Aufgrund der Größe der Printplatte haben wir uns dann ein passendes Gehäuse ausgewählt und alle benötigten Komponenten in dieses Gehäuse eingebaut.

10.7 FLUSSDIAGRAMM

Bei diesem Flussdiagramm handelt es sich um eine sehr vereinfachte schematische Darstellung, um einen Grundüberblick vermitteln zu können. Aufgrund des Umfangs des Programmes und der Komplexität lässt sich eine 1:1 Darstellung hier nicht sinnvoll umsetzen. Es sind hier grob die wichtigsten Funktionen abgebildet.

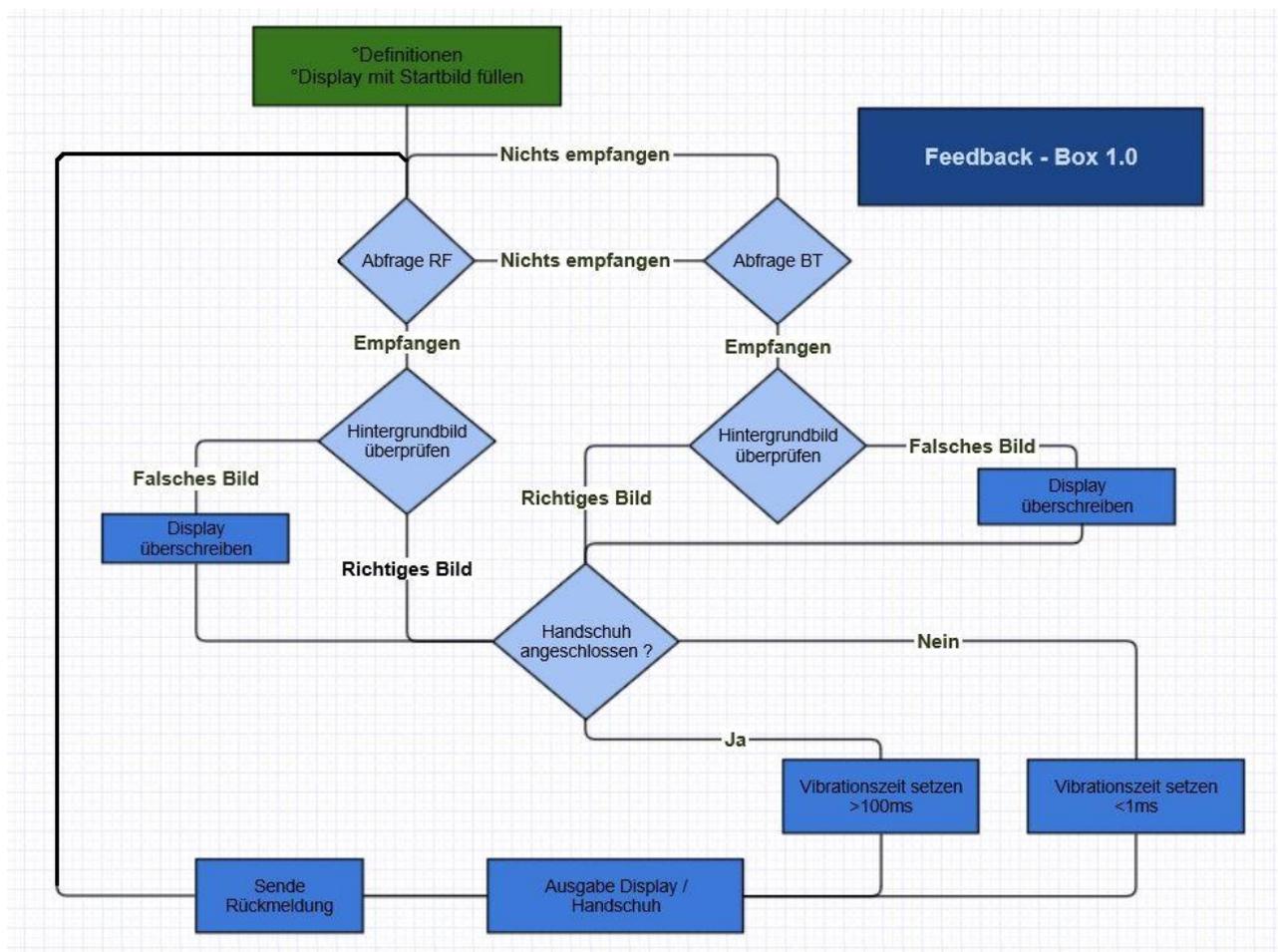


Abbildung 62: Flussdiagramm Feedback-Box

10.8 PROGRAMMCODE

```
[code]
/*
FEEDBACK-BOX
* Autor:
* Michael Eder
* Co-Autor:
* Alexander Bachmeier
*/
//Lade RF-Modul Libraries
#include <nRF24L01.h>
#include <RF24.h>
#include <SPI.h>
//Lade Display Libraries
#include <Adafruit_GFX.h> // Core graphics library
#include <Adafruit_TFTLCD.h> // Hardware-specific library

//Lade SD-Card Library
#include <SD.h>

//BT-Modul Definitionen
int tx=19;
int rx=18;
char inSerial[32]={0};

//RF-Modul Definitionen
RF24 radio(49,48); //setzen von (RX/TX) für RF
char textrx[32]={0}; //Text-Array das empfangen wird name[Länge] = {jede Stelle = 0}
char texttx[32]={0}; //Text-Array das gesendet wird name[Länge] = {jede Stelle = 0}
const byte rxAddr[6]="00001"; //Empfangs-Adresse des RF-Modul empfänger[länge]=adresse
const byte txAddr[6]="00002"; //Sende-Adresse des RF-Modul

// Anschluß Display
#define LCD_CS A3 // Chip Select - Analog 3
#define LCD_CD A2 // Command/Data - Analog 2
#define LCD_WR A1 // LCD Write - Analog 1
#define LCD_RD A0 // LCD Read - Analog 0
#define LCD_RESET A4 // LCD Reset - Analog 4 - Alternativ auf Reset Arduino

// Definiere Farbcode für Display
#define BLACK 0x0000 /* 0, 0, 0 */
#define Navy 0x000F /* 0, 0, 128 */
#define DarkGreen 0x03E0 /* 0, 128, 0 */
#define DarkCyan 0x03EF /* 0, 128, 128 */
#define Maroon 0x7800 /* 128, 0, 0 */
#define Purple 0x780F /* 128, 0, 128 */
#define Olive 0x7BE0 /* 128, 128, 0 */
#define LightGrey 0xC618 /* 192, 192, 192 */
#define DarkGrey 0x7BEF /* 128, 128, 128 */
#define BLUE 0x001F /* 0, 0, 255 */
#define RED 0xF800 /* 255, 0, 0 */
#define GREEN 0x07E0 /* 0, 255, 0 */
#define CYAN 0x07FF /* 0, 255, 255 */
#define MAGENTA 0xF81F /* 255, 0, 255 */
#define YELLOW 0xFFE0 /* 255, 255, 0 */
#define WHITE 0xFFFF /* 255, 255, 255 */
#define Orange 0xFD20 /* 255, 165, 0 */
#define GreenYellow 0xAFE5 /* 173, 255, 47 */
```

```

#define Pink                0xF81F

#if defined __AVR_ATmega2560__
#define SD_SCK 13
#define SD_MISO 12
#define SD_MOSI 11
#endif
#define SD_CS 10           //

//Laden des Schaltungsaufbaus in die Library
Adafruit_FTTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

//Globale Variablen für Output Setzung der Vibratoren über die Demultiplexer
int startPort =23;//Demultiplexer Ausgangsanfang
int endPort =29;//Demultiplexer Ausgangsende
int vibdelay =0;//Millisekunden für Vibrationsverzögerung zwischen den
Vibrationen
int vibdelay1=0;//Millisekunden für Vibrationsverzögerung zwischen den
Buchstaben
int k=0;//Zählervariable löscht Display wenn voll
int q=0;//Zählervariable für Display reset bei 2maligen den Vibrationen beim
Leerzeichen
int p=0;//Wechsel zwischen RF-Farbe und BT-Farbe am Display
int Tlleer=0;//Millisekunden für Vibrationsverzögerung zwischen Leerzeichen
int leerzeichenwait=0;

//Vibratoren Dezimaldefinition und Enable
//{enable Demux1,enable Demux2,enable Demux3,binär Code 4 Bit)   -
2^3,2^2,2^1,2^0
//{41,42,43,44,45,46,47}
char Vibrator[]={'1','1','1','0','0','0','0'};//Vibrator für Pinset,
Vibratorenzuordnung wird in dieses Array geschrieben
//Demultiplexer1
char Vibrator1[]={'1','1','0','0','0','0','0'};
char Vibrator2[]={'1','1','0','0','0','0','1'};
char Vibrator3[]={'1','1','0','0','0','1','0'};
char Vibrator4[]={'1','1','0','0','0','1','1'};
char Vibrator5[]={'1','1','0','0','1','0','0'};
char Vibrator6[]={'1','1','0','0','1','0','1'};
char Vibrator7[]={'1','1','0','0','1','1','0'};
char Vibrator8[]={'1','1','0','0','1','1','1'};
char Vibrator9[]={'1','1','0','1','0','0','0'};
char Vibrator10[]={'1','1','0','1','0','0','1'};
char Vibrator11[]={'1','1','0','1','0','1','0'};
char Vibrator12[]={'1','1','0','1','0','1','1'};
char Vibrator13[]={'1','1','0','1','1','0','0'};
char Vibrator14[]={'1','1','0','1','1','0','1'};
char Vibrator15[]={'1','1','0','1','1','1','0'};
char Vibrator16[]={'1','1','0','1','1','1','1'};
//Demultiplexer2
char Vibrator17[]={'1','0','1','0','0','0','0'};
char Vibrator18[]={'1','0','1','0','0','0','1'};
char Vibrator19[]={'1','0','1','0','0','1','0'};
char Vibrator20[]={'1','0','1','0','0','1','1'};
char Vibrator21[]={'1','0','1','0','1','0','0'};
char Vibrator22[]={'1','0','1','0','1','0','1'};
char Vibrator23[]={'1','0','1','0','1','1','0'};
char Vibrator24[]={'1','0','1','0','1','1','1'};
char Vibrator25[]={'1','0','1','1','0','0','0'};
char Vibrator26[]={'1','0','1','1','0','0','1'};
char Vibrator27[]={'1','0','1','1','0','1','0'};
char Vibrator28[]={'1','0','1','1','0','1','1'};
char Vibrator29[]={'1','0','1','1','1','0','0'};

```

```

char Vibrator30[]={ '1', '0', '1', '1', '1', '0', '1' };
char Vibrator31[]={ '1', '0', '1', '1', '1', '1', '0' };
char Vibrator32[]={ '1', '0', '1', '1', '1', '1', '1' };
//Demultiplexer3
char Vibrator33[]={ '0', '1', '1', '0', '0', '0', '0' };
char Vibrator34[]={ '0', '1', '1', '0', '0', '0', '1' };
char Vibrator35[]={ '0', '1', '1', '0', '0', '1', '0' };
char Vibrator36[]={ '0', '1', '1', '0', '0', '1', '1' };
char Vibrator37[]={ '0', '1', '1', '0', '1', '0', '0' };
char Vibrator38[]={ '0', '1', '1', '0', '1', '0', '1' };
char Vibrator39[]={ '0', '1', '1', '0', '1', '1', '0' };
char Vibrator40[]={ '0', '1', '1', '0', '1', '1', '1' };
char Vibrator41[]={ '0', '1', '1', '1', '0', '0', '0' };
char Vibrator42[]={ '0', '1', '1', '1', '0', '0', '1' };
char Vibrator43[]={ '0', '1', '1', '1', '0', '1', '0' };
char Vibrator44[]={ '0', '1', '1', '1', '0', '1', '1' };
char Vibrator45[]={ '0', '1', '1', '1', '1', '0', '0' };
char Vibrator46[]={ '0', '1', '1', '1', '1', '0', '1' };
char Vibrator47[]={ '0', '1', '1', '1', '1', '1', '0' };
char Vibrator48[]={ '0', '1', '1', '1', '1', '1', '1' };
//Vibrartor Reset
char Vibrator000[]={ '1', '1', '1', '0', '0', '0', '0' }; //Vibrator für Zero() zum
nicht ansteuern der Vibratoren

//setup wird einmal beim Start des Mikrokontrollers durchlaufen
void setup()
{
//BT-Modul Initialisierung
  Serial1.begin(9600); //Aktiviere BT-Modul
  pinMode(19, INPUT_PULLUP); //fix Serial1
  pinMode(tx, OUTPUT); //setze tx-Pin als Ausgang
  pinMode(rx, INPUT); //setze rx-Pin als Eingang

//Aktiviere Serial Monitor
  Serial.begin(9600);

//Pin für Demultiplexer LE
  pinMode(22, OUTPUT);
  digitalWrite(22, HIGH);

//Display Initialisierung
  tft.reset(); //Display neustart
  uint16_t identfier = tft.readID(); //Displaygröße ermitteln
  tft.begin(identfier); //Aktiviere Display setzen
  tft.setRotation(1); //Orientierung des Displays
  tft.fillScreen(0); //fülle Display komplett schwarz
  tft.setCursor(0,0); //setze Cursor auf Position
  tft.setTextColor(WHITE); //Schriftfarbe
  tft.setTextSize(2); //Schriftgröße

//Initialisierung der SD-Karte für Bild-Ausgabe am Display
tft.println(F("Initializing SD card..."));

//Prüfe SD-Card
  #if defined __AVR_ATmega2560__

  if(!SD.begin(SD_CS, SD_MOSI, SD_MISO, SD_SCK )) {
    tft.println(F("failed!"));
    return;
  }
  #else

  if(!SD.begin(SD_CS)) {

```

```

    tft.println(F("failed!"));
return;
}
#endif

    bmpDraw("start.bmp",0,0);//Lade start.bmp in Display
tft.setTextSize(4);//Schriftgröße
    tft.setTextColor(GREEN);//Schriftfarbe
    tft.setCursor(0,0);//setze Cursor auf Position

//RF-Modul Initialisierung
    radio.begin();//Aktiviere RF Modul
    radio.setRetries(15,15);//sendeversuche( 150 mal, für 150ms * 80)
    radio.openReadingPipe(1, rxAddr);//Empfängeradresse
    radio.openWritingPipe(txAddr);//Definieren des Sende-Pin

    Zero();//Setze Vibratoren auf 0
}

//loop wird nach dem setup permanent wiederholt
void loop()
{
//Aufruf der void RFmodul
    RFmodul();
}

//Anfang der BTmodul-Funktion
void BTmodul()
{

if(Serial1.available()>0&& radio.available()!=true)//Wird aktiv wenn BT-Signal
kommt und RF-Signal nicht
{
if(p!=2)//Wenn zuvor nicht BT aktiv war
{
    bmpDraw("Handy.bmp",0,0);//Lade Lorm.bmp in Display
tft.setCursor(0,0);//setze Cursor auf Position
}

    p=2;//Setze um zu erkennen, dass BT-Modul aktiv ist
String inSerial=Serial1.readString();//Liest den kommenden String von BT-Modul

for(int j =0; j <31; j++)//Schreibt den BT-String in unser textrx-Array
{
    textrx[j]=inSerial[j];
}

    tft.setTextColor(Orange);//Schriftfarbe
    Vibration();//Ausgabe wird abgearbeitet
    senden();//Rückmeldung senden an RF-Modul/BT-Modul
    radio.startListening();//Empfangen RF-Modul aktivieren
}
}

//void dient dem Einlesen des RF Signals und startet die Void Vibration
void RFmodul()
{
    radio.startListening();//Empfangen RF-Modul aktivieren

while(radio.available()!=true)//warten auf Rf-Modul Daten
{
    BTmodul();//Wenn kein RF-Signal verfügbar ist dann Warte auf BT-Signal
}
}

```

```

if(p==0)//p ist 0 nach dem Setup
{
    bmpDraw("Lorm.bmp",0,0);//Lade Lorm.bmp in Display
}

if(p==2)//Wenn zuvor das BT-Modul aktiv war dann lade Lorm.bmp
{
    bmpDraw("Lorm.bmp",0,0);//Lade Lorm.bmp in Display
    tft.setCursor(0,0);//setze Cursor auf Position
}
    p=1;//Setze um zu erkennen das RF-Modul aktiv ist

    tft.setTextColor(GREEN);//Schriftfarbe
    radio.read(&textrx,sizeof(textrx));//empfangen des CharArrays (name ,
größe)
    radio.stopListening();//Empfangen RF-Modul deaktivieren
    Vibration();//Ausgabe wird abgearbeitet
    senden();//Rückmeldung senden an RF-Modul/BT-Modul
    radio.startListening();//Empfangen RF-Modul aktivieren
}

//Beginn der void DisplayAusgaben
void DisplayAusgaben(int j)//die void bekommt j übergeben um zu wissen an
welcher Stelle des text-Arrays wir uns befinden
{

    k=k+1;// k ist unser Zeichenzähler und wird jeweils nach einem Zeichen
erhöht
    delay(vibdelay1);// Verzögerungszeit vor der Ausgabe am Display
if(k>=91)// wenn das Display voll geschrieben ist wird es neu geladen
{
    tft.setCursor(0,0);//Position für Zeichen wird auf x,y = 0,0 gesetzt

if(p==1)//Wenn das RF-Modul zuvor aktiv war
{
    tft.setTextColor(GREEN);//Schriftfarbe: grün
    tft.setTextSize(4);//Schriftgröße 4
    bmpDraw("Lorm.bmp",0,0);//Bild: Lorm.bmp
}

if(p==2)//Wenn das BT-Modul zuvor aktiv war
{
    tft.setTextColor(Orange);//Schriftfarbe: orange
    tft.setTextSize(4);//Schriftgröße 4
    bmpDraw("Handy.bmp",0,0);//Bild: Handy.bmp
}

    k=0;//Zeichenzähler wieder auf 0
}

if(q<2)//wenn weniger als 2 Leerzeichen
{
    tft.print(textrx[j]);//Ausgabe am Display des Zeichens von textrx an der
Stelle j
}

    Serial.print(textrx[j]);//Ausgabe am Serialmonitor des Zeichens von textrx
an der Stelle j
    q=0;//setze Variable für Leerzeichen zählen auf 0
}

//Beginn der void Vibration

```

```

void Vibration()
{
for(int j =0; j <sizeof(textrx)-1; j++)//Lässt jede Stelle der textrx-Array
durchlaufen
{
    pinMode(14, INPUT);//Input für Handschuh angeschlossen

if(digitalRead(14)==HIGH)//Wenn der Handschh angeschlossen ist werden die
Zeiten so gesetzt das der Ablauf mit den Vibrationen optimal läuft
{
//setzen der Zeiten
    vibdelay=600;
    vibdelay1=400;
    Tlleer=40;
    leerzeichenwait=2000;
}

if(digitalRead(14)!=HIGH)//Wenn der Handschh nicht angeschlossen ist werden die
Zeiten möglichst kurz gesetzt
{
//setzen der Zeiten
    vibdelay=10;
    vibdelay1=10;
    Tlleer=1;
    leerzeichenwait=10;
}

//Anfang des Switch
switch(textrx[j])//Im Switch wird je nach dem welches Zeichen im textrx-Array
ist passende Ausgaben gemacht
{

case 'A'://Wenn das Zeichen im textrx[j]=A ist
    DisplayAusgaben(j);//Aufruf der Ausgabe mit übergabe der Stelle j im
textrx-Array

for(int i =0; i <sizeof(Vibrator11); i++)//schreibt in das Vibrator[]-Array das
Vibrator11[]-Array um den richtigen Vibrator anzusteuern
{
    Vibrator[i]=Vibrator11[i];
}

    PinSet();//setzt die Ausgänge um Vibrator11 anzusteuern
    delay(vibdelay);//Zeit wie lange der Vibrator vibriert
    Zero();//setzt das Vibrator[]-Array auf Vibrator000 und dann die
Ausgänge, um keinen Vibrator mehr anzusteuern
break;//bricht den Switch ab

case 'E'://Wenn das Zeichen im textrx[j]=E ist
    DisplayAusgaben(j);//Aufruf der Ausgabe mit übergabe der Stelle j im
textrx-Array

for(int i =0; i <sizeof(Vibrator1); i++)//schreibt in das Vibrator[]-Array das
Vibrator1[]-Array um den richtigen Vibratoranzusteuern
{
    Vibrator[i]=Vibrator1[i];
}

    PinSet();//setzt die Ausgänge um Vibrator1 anzusteuern
    delay(vibdelay);//Zeit wie lange der Vibrator vibriert
    Zero();//setzt das Vibrator[]-Array auf Vibrator000 und dann die
Ausgänge, um keinen Vibrator mehr anzusteuern
break;//bricht den Switch ab
}
}

```

```

case 'I':
DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator8); i++)
{
    Vibrator[i]=Vibrator8[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'O':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator18); i++)
{
    Vibrator[i]=Vibrator18[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'U':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator36); i++)
{
    Vibrator[i]=Vibrator36[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case '1'://Wenn das Zeichen im textrx[j]=1 für "AE" ist
//Serial1.println("AE");
    q=0;//Variable für doppeltes Leerzeichen wird zurückgesetzt
    k=k+2;//erhöht die Variable zum Zeichenzählen um 2
if(k>=91)//wenn das Display voll geschrieben ist wird das Display neu geladen
{
    delay(1000);//Verzögerung um die Zeichen am Display noch lesen zu
können bevor das Display neu geladen wird
    tft.setCursor(0,0);//setzten den Zeichencursor auf x,y = 0,0

if(p==1)//Wenn das RF-Modul zuvor aktiv war
{
    tft.setTextColor(GREEN);//Schriftfarbe: grün
    tft.setTextSize(4);//Schriftgröße 4
    bmpDraw("Lorm.bmp",0,0);//Bild: Lorm.bmp
}

if(p==2)//Wenn das BT-Modul zuvor aktiv war
{
    tft.setTextColor(Orange);//Schriftfarbe: orange
    tft.setTextSize(4);//Schriftgröße 4
    bmpDraw("Handy.bmp",0,0);//Bild: Handy.bmp
}

    k=0;//Zeichenzähler wieder auf 0

```

```

}
    tft.print("AE");//Ausgabe am Display "AE"
    Serial.print(textrx[j]);//Ausgabe am Serialmonitor textrx-Array an der
Stelle j

for(int i =0; i <sizeof(Vibrator11); i++)//schreibt in das Vibrator[]-Array das
Vibrator11[]-Array um den richtigen Vibratoranzusteuern
{
    Vibrator[i]=Vibrator11[i];
}

    PinSet();//setzt die Ausgänge um Vibrator11 anzusteuern
    delay(vibdelay);//Zeit wie lange der Vibrator vibriert
    Zero();//setzt das Vibrator[]-Array auf Vibrator000 und dann die
Ausgänge, um keinen Vibrator mehr anzusteuern

for(int i =0; i <sizeof(Vibrator11); i++)//schreibt in das Vibrator[]-Array das
Vibrator11[]-Array um den richtigen Vibratoranzusteuern
{
    Vibrator[i]=Vibrator11[i];
}

    PinSet();//setzt die Ausgänge um Vibrator11 anzusteuern
    delay(vibdelay);//Zeit wie lange der Vibrator vibriert
    Zero();//setzt das Vibrator[]-Array auf Vibrator000 und dann die
Ausgänge, um keinen Vibrator mehr anzusteuern
break;//bricht den Switch ab

case '2':
//Serial1.println("OE");
    q=0;
    k=k+2;

if(k>=91)
{
    delay(1000);
    tft.setCursor(0,0);

if(p==1)
{
    tft.setTextColor(GREEN); tft.setTextSize(4);
    bmpDraw("Lorm.bmp",0,0);
}

if(p==2)
{
    tft.setTextColor(Orange); tft.setTextSize(4);
    bmpDraw("Handy.bmp",0,0);
}

    k=0;
}

tft.print("OE");
Serial.print(textrx[j]);

for(int i =0; i <sizeof(Vibrator18); i++)
{
    Vibrator[i]=Vibrator18[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

```

```

for(int i =0; i <sizeof(Vibrator18); i++)
{
    Vibrator[i]=Vibrator18[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case '3':
//Serial1.println("UE");
    q=0;
    k=k+2;

if(k>=91)
{
    delay(1000);
    tft.setCursor(0,0);

if(p==1)
{
    tft.setTextColor(GREEN);    tft.setTextSize(4);
    bmpDraw("Lorm.bmp",0,0);
}

if(p==2)
{
    tft.setTextColor(Orange);    tft.setTextSize(4);
    bmpDraw("Lorm.bmp",0,0);
}

    k=0;
}

tft.print("UE");
Serial.print(textrx[j]);

for(int i =0; i <sizeof(Vibrator36); i++)
{
    Vibrator[i]=Vibrator36[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator36); i++)
{
    Vibrator[i]=Vibrator36[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'J':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator8); i++)
{
    Vibrator[i]=Vibrator8[i];
}

```

```

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator8); i++)
    {
        Vibrator[i]=Vibrator8[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();
break;

    case 'B':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator3); i++)
    {
        Vibrator[i]=Vibrator3[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator4); i++)
    {
        Vibrator[i]=Vibrator4[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();
break;

    case 'D':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator6); i++)
    {
        Vibrator[i]=Vibrator6[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator5); i++)
    {
        Vibrator[i]=Vibrator5[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();
break;

    case 'G':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator21); i++)
    {
        Vibrator[i]=Vibrator21[i];
    }

```

```

}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator20); i++)
{
    Vibrator[i]=Vibrator20[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'H':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator28); i++)
{
    Vibrator[i]=Vibrator28[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator19); i++)
{
    Vibrator[i]=Vibrator19[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'T':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator24); i++)
{
    Vibrator[i]=Vibrator24[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator23); i++)
{
    Vibrator[i]=Vibrator23[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'F':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator40); i++)

```

```

{
    Vibrator[i]=Vibrator40[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'P':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator25); i++)
{
    Vibrator[i]=Vibrator25[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator13); i++)
{
    Vibrator[i]=Vibrator13[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'K':
    DisplayAusgaben(j);
for(int i =0; i <sizeof(Vibrator38); i++)

{
    Vibrator[i]=Vibrator38[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'L':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator6); i++)
{
    Vibrator[i]=Vibrator6[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator5); i++)
{
    Vibrator[i]=Vibrator5[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

```

```

for(int i =0; i <sizeof(Vibrator2); i++)
{
    Vibrator[i]=Vibrator2[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator31); i++)
{
    Vibrator[i]=Vibrator31[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator34); i++)
{
    Vibrator[i]=Vibrator34[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'M':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator22); i++)
{
    Vibrator[i]=Vibrator22[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'N':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator32); i++)
{
    Vibrator[i]=Vibrator32[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'R':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator29); i++)
{
    Vibrator[i]=Vibrator29[i];
}

    PinSet();

```

```

        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator31); i++)
    {
        Vibrator[i]=Vibrator31[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator10); i++)
    {
        Vibrator[i]=Vibrator10[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    break;

    case 'S':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator2); i++)
    {
        Vibrator[i]=Vibrator2[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator33); i++)
    {
        Vibrator[i]=Vibrator33[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator30); i++)
    {
        Vibrator[i]=Vibrator30[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator29); i++)
    {
        Vibrator[i]=Vibrator29[i];
    }

        PinSet();
        delay(vibdelay);
        Zero();

    for(int i =0; i <sizeof(Vibrator34); i++)
    {

```

```

        Vibrator[i]=Vibrator34[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    for(int i =0; i <sizeof(Vibrator10); i++)
    {
        Vibrator[i]=Vibrator10[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    for(int i =0; i <sizeof(Vibrator26); i++)
    {
        Vibrator[i]=Vibrator26[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    for(int i =0; i <sizeof(Vibrator2); i++)
    {
        Vibrator[i]=Vibrator2[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    break;

    case 'Z':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator34); i++)
    {
        Vibrator[i]=Vibrator34[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    for(int i =0; i <sizeof(Vibrator29); i++)
    {
        Vibrator[i]=Vibrator29[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    for(int i =0; i <sizeof(Vibrator30); i++)
    {
        Vibrator[i]=Vibrator30[i];
    }

    PinSet();
    delay(vibdelay);

```

```

        Zero();

    for(int i =0; i <sizeof(Vibrator22); i++)
    {
        Vibrator[i]=Vibrator22[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    break;

    case 'V':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator37); i++)
    {
        Vibrator[i]=Vibrator37[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    break;

    case 'W':
        DisplayAusgaben(j);

    for(int i =0; i <sizeof(Vibrator37); i++)
    {
        Vibrator[i]=Vibrator37[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    for(int i =0; i <sizeof(Vibrator37); i++)
    {
        Vibrator[i]=Vibrator37[i];
    }

    PinSet();
    delay(vibdelay);
    Zero();

    break;

    case '4':
        //Serial1.println("CH");
        q=0;
        k=k+2;

    if(k>=91)
    {
        delay(1000);
        tft.setCursor(0,0);

    if(p==1)
    {
        tft.setTextColor(GREEN);    tft.setTextSize(4);
        bmpDraw("Lorm.bmp",0,0);
    }
}

```

```

if(p==2)
{
    tft.setTextColor(Orange);  tft.setTextSize(4);
    bmpDraw("Handy.bmp",0,0);
}

    k=0;
}

tft.print("CH");
Serial.print(textrx[j]);

for(int i =0; i <sizeof(Vibrator26); i++)
{
    Vibrator[i]=Vibrator26[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator31); i++)
{
    Vibrator[i]=Vibrator31[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator29); i++)
{
    Vibrator[i]=Vibrator29[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator33); i++)
{
    Vibrator[i]=Vibrator33[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator31); i++)
{
    Vibrator[i]=Vibrator31[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator10); i++)
{
    Vibrator[i]=Vibrator10[i];
}

    PinSet();
    delay(vibdelay);

```

```

        Zero();
break;

case '5':
//Serial1.println("SCH");
    q=0;
    k=k+3;

if(k>=91)
{
    delay(1000);
    tft.setCursor(0,0);

if(p==1)
{
    tft.setTextColor(GREEN);    tft.setTextSize(4);
    bmpDraw("Lorm.bmp",0,0);
}

if(p==2)
{
    tft.setTextColor(Orange);    tft.setTextSize(4);
    bmpDraw("Handy.bmp",0,0);
}

    k=0;
}

    tft.print("SCH");
    Serial.print(textrx[j]);

for(int i =0; i <sizeof(Vibrator39); i++)
{
    Vibrator[i]=Vibrator39[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case '6':
//Serial1.println("ST");
    q=0;
    k=k+2;

if(k>=91)
{
    delay(1000);
    tft.setCursor(0,0);

if(p==1)
{
    tft.setTextColor(GREEN);    tft.setTextSize(4);
    bmpDraw("Lorm.bmp",0,0);
}

if(p==2)
{
    tft.setTextColor(Orange);    tft.setTextSize(4);
    bmpDraw("Handy.bmp",0,0);
}

    k=0;
}

```

```

}

    tft.print("ST");
    Serial.print(textrx[j]);

for(int i =0; i <sizeof(Vibrator9); i++)
{
    Vibrator[i]=Vibrator9[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator12); i++)
{
    Vibrator[i]=Vibrator12[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'C':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator16); i++)
{
    Vibrator[i]=Vibrator16[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'X':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator14); i++)
{
    Vibrator[i]=Vibrator14[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator16); i++)
{
    Vibrator[i]=Vibrator16[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator15); i++)
{
    Vibrator[i]=Vibrator15[i];
}

    PinSet();

```

```

        delay(vibdelay);
        Zero();
break;

case 'Q':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator17); i++)
{
    Vibrator[i]=Vibrator17[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator35); i++)
{
    Vibrator[i]=Vibrator35[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator27); i++)
{
    Vibrator[i]=Vibrator27[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case 'Y':
    DisplayAusgaben(j);

for(int i =0; i <sizeof(Vibrator4); i++)
{
    Vibrator[i]=Vibrator4[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator5); i++)
{
    Vibrator[i]=Vibrator5[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

for(int i =0; i <sizeof(Vibrator20); i++)
{
    Vibrator[i]=Vibrator20[i];
}

    PinSet();
    delay(vibdelay);
    Zero();

```

```

for(int i =0; i <sizeof(Vibrator19); i++)
{
    Vibrator[i]=Vibrator19[i];
}

    PinSet();
    delay(vibdelay);
    Zero();
break;

case ' ':
q=q+1;//Zähler erhöhen für doppeltes Leerzeichen

if(q>1)//Variable größer als eins dann Display neu laden
{
    tft.setCursor(0,0);//Zeichencursor auf x,y = 0,0

if(p==1)//Wenn das RF-Modul zuvor aktiv war
{
    tft.setTextColor(GREEN);//Schriftfarbe: grün
    tft.setTextSize(4);//Schriftgröße 4
bmpDraw("Lorm.bmp",0,0);//Bild: Lorm.bmp
}

if(p==2)//Wenn das BT-Modul zuvor aktiv war
{
    tft.setTextColor(Orange);//Schriftfarbe: orange
    tft.setTextSize(4);//Schriftgröße 4
bmpDraw("Handy.bmp",0,0);//Bild: Handy.bmp
}

    k=0;//Zeichenzähler wieder auf 0
}

    delay(vibdelay1);//Verzögerung um Leerzeichen deutlicher zu erkennen
    DisplayAusgaben(j);//Aufruf void DisplayAusgaben mit Übergabe von j für
die Position im textrx-Array
    q=q+1;//Zähler erhöhen für doppeltes Leerzeichen
long T4=0;//Zeit für die while-Schleife
long T5=0;//Zeit für die while-Schleife
T4=millis();//Aktualisiert Serverzeit

    T5=T4+leerzeichenwait;//Anfangszeit setzen

//In der while-Schleife werden mehrere Vibratoren in kurzer zeit hintereinander
gesetzt bis die zeit abgelaufen ist
while(T4<T5)//wenn leerzeichenwait vergangen ist hört die schleife auf
{
for(int i =0; i <sizeof(Vibrator38); i++)
{
    Vibrator[i]=Vibrator38[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator22); i++)
{
    Vibrator[i]=Vibrator22[i];
}

    PinSet();
    delay(T1leer);

```

```

for(int i =0; i <sizeof(Vibrator30); i++)
{
    Vibrator[i]=Vibrator30[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator29); i++)
{
    Vibrator[i]=Vibrator29[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator2); i++)
{
    Vibrator[i]=Vibrator2[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator31); i++)
{
    Vibrator[i]=Vibrator31[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator34); i++)
{
    Vibrator[i]=Vibrator34[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator32); i++)
{
    Vibrator[i]=Vibrator32[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator26); i++)
{
    Vibrator[i]=Vibrator26[i];
}

    PinSet();
    delay(T1leer);

for(int i =0; i <sizeof(Vibrator10); i++)
{
    Vibrator[i]=Vibrator10[i];
}

    PinSet();
    delay(T1leer);

```

```

for(int i =0; i <sizeof(Vibrator37); i++)
{
    Vibrator[i]=Vibrator37[i];
}

    PinSet();
    delay(Tl1eer);

for(int i =0; i <sizeof(Vibrator38); i++)
{
    Vibrator[i]=Vibrator38[i];
}

    PinSet();
delay(Tl1eer);

    T4=millis();//aktualisiere Zeit
}

//schreibt in das Vibrator[]-Array das Vibrator000[]-Array um keinen Vibrator
anzusteuern
for(int i =0; i <sizeof(Vibrator000); i++)
{
    Vibrator[i]=Vibrator000[i];
}
    PinSet();//Setzt die Ausgänge, um keinen Vibrator anzusteuern
delay(vibdelay1);
break;
} //ende switch

} //ende for

}

//Anfang der PinSet-Funktion
void PinSet()
{
//Variablen für die Funktion PinSet
int counter =0;//Zähler für die for-Schleife
int from=endPort;//Definition des letzten Ausgang
int to=startPort;//Definition des ersten Ausgang

//Schleife zum setzen der Ausgänge
for(from; from >= to; from--)//Setzt die Ausgänge anfangend mit dem letzten
{
    pinMode(from, OUTPUT);//Definition Ausgang

if(Vibrator[counter]==Vibrator000[1])//Wenn die stelle 1 ist setze den Ausgang
HIGH
{
    digitalWrite(from, HIGH);
}

if(Vibrator[counter]==Vibrator000[6])//Wenn die stelle 0 ist setze den Ausgang
LOW
{
    digitalWrite(from, LOW);
}

    counter++; //Erhöht den Zähler bei jedem durchlaufen um 1
}
}

```

```

//Anfang der Zero-Funktion
void Zero()
{
//schreibt in das Vibrator[]-Array das Vibrator000[]-Array um keinen Vibrator
anzusteuern
for(int i =0; i <sizeof(Vibrator000); i++)
{
    Vibrator[i]=Vibrator000[i];
}
    PinSet();//Setzt die Asugänge, um keinen Vibrator anzusteuern
    delay(vibdelay);//Verzögerung nach und zwischen den Vibrationen
}

//Anfang der senden-Funktion
void senden()
{
//RF-Modul in Sendemodus setzen
    radio.stopListening ();

//textrx-Array in das texttx-Array schreiben
for(int j =0; j <sizeof(textrx)-1; j++)
{
    texttx[j]=textrx[j];
}
//Ausgabe auf Serialmonitor das wir senden
Serial.print('\n');
    Serial.print(texttx);
Serial.print('\n');
    Serial.print("Senden");
    Serial.print('\n');
    delay(30);

//Wenn Bluetooth aktive auf Handy schicken mittels Serial1
if(p==2)
{
    Serial1.println(texttx);//Senden des (texttx-Array, mit der Größe von
texttx)
}

//Wenn Radiofrequenz aktiv dann zur Lormen-Hand schicken
if(p==1)
{
    radio.write(&texttx,sizeof(texttx));//Senden des (texttx-Array, mit der
Größe von texttx)
}

//resetet texttx-Array
for(int i =0; i <sizeof(texttx)-1; i++)
{
    texttx[i]={0};
}
}

//SD-Card laden der Abgespeicherten Bilder in das Display
#define BUFFPIXEL 20
//void bmpDraw ist eine vor gegebene Funktion von https://learn.adafruit.com/
wird benötigt um Display mit Bildern zu laden
void bmpDraw(char*filename,int x,int y){
    File    bmpFile;
int        bmpWidth, bmpHeight;// W+H in pixels
    uint8_t bmpDepth;// Bit depth (currently must be 24)

```

```

    uint32_t bmpImageoffset;// Start of image data in file
    uint32_t rowSize;// Not always = bmpWidth; may have padding
    uint8_t  sdbuffer[3*BUFFPIXEL];// pixel in buffer (R+G+B per pixel)
    uint16_t lcdbuffer[BUFFPIXEL];// pixel out buffer (16-bit per pixel)
    uint8_t  buffidx =sizeof(sdbuffer);// Current position in sdbuffer
    boolean  goodBmp =false;// Set to true on valid header parse
    boolean  flip    =true;// BMP is stored bottom-to-top
    int      w, h, row, col;
    uint8_t  r, g, b;
    uint32_t pos =0, startTime = millis();
    uint8_t  lcdidx =0;
    boolean  first =true;

    if((x >= tft.width())|| (y >= tft.height()))return;

#ifdef DEBUG
    tft.fillScreen(0);
    tft.setCursor(0,0);
    tft.print(F("Loading image "));
    tft.print(filename);
    tft.println('\n');
#endif
// Open requested file on SD card
if((bmpFile = SD.open(filename))==NULL){
#ifdef DEBUG
    tft.println(F("File not found"));
#endif
return;
}

// Parse BMP header
if(read16(bmpFile)==0x4D42){
// BMP signature
#ifdef DEBUG
    tft.println(F("File size: ")); tft.println(read32(bmpFile));
#else
    read32(bmpFile);
#endif
(void)read32(bmpFile);// Read & ignore creator bytes
    bmpImageoffset = read32(bmpFile);// Start of image data
#ifdef DEBUG
    tft.print(F("Image Offset: ")); tft.println(bmpImageoffset, DEC);
// Read DIB header
    tft.print(F("Header size: ")); tft.println(read32(bmpFile));
#else
    read32(bmpFile);
#endif

    bmpWidth  = read32(bmpFile);
    bmpHeight = read32(bmpFile);
    if(read16(bmpFile)==1){// # planes -- must be '1'
        bmpDepth = read16(bmpFile);// bits per pixel
#ifdef DEBUG
        Serial.print(F("Bit Depth: ")); Serial.println(bmpDepth);
#endif
        if((bmpDepth ==24)&&(read32(bmpFile)==0)){// 0 = uncompressed

            goodBmp =true;// Supported BMP format -- proceed!
#ifdef DEBUG
            tft.print(F("Image size: "));
            tft.print(bmpWidth);
            tft.print('x');
            tft.println(bmpHeight);
            delay(3000);

```

```

        #endif
// BMP rows are padded (if needed) to 4-byte boundary
        rowSize = (bmpWidth *3+3)&~3;

// If bmpHeight is negative, image is in top-down order.
// This is not canon but has been observed in the wild.
if(bmpHeight <0){
        bmpHeight =-bmpHeight;
        flip      =false;
}

// Crop area to be loaded
        w = bmpWidth;
        h = bmpHeight;
if((x+w-1)>= tft.width()) w = tft.width()- x;
if((y+h-1)>= tft.height()) h = tft.height()- y;

// Set TFT address window to clipped image bounds
        tft.setAddrWindow(x, y, x+w-1, y+h-1);

for(row=0; row<h; row++){// For each scanline...
// Seek to start of scan line. It might seem labor-
// intensive to be doing this on every line, but this
// method covers a lot of gritty details like cropping
// and scanline padding. Also, the seek only takes
// place if the file position actually needs to change
// (avoids a lot of cluster math in SD library).
if(flip)// Bitmap is stored bottom-to-top order (normal BMP)
        pos = bmpImageoffset + (bmpHeight -1- row)* rowSize;
else// Bitmap is stored top-to-bottom
        pos = bmpImageoffset + row * rowSize;
if(bmpFile.position() != pos){// Need seek?
        bmpFile.seek(pos);
        buffidx =sizeof(sdbuffer);// Force buffer reload
}

for(col=0; col<w; col++){// For each column...
// Time to read more pixel data?
if(buffidx >=sizeof(sdbuffer)){// Indeed
// Push LCD buffer to the display first
if(lcdidx >0){
        tft.pushColors(lcdbuffer, lcdidx, first);
        lcdidx =0;
        first =false;
}

        bmpFile.read(sdbuffer, sizeof(sdbuffer));
        buffidx =0;// Set index to beginning
}

// Convert pixel from BMP to TFT format
        b = sdbuffer[buffidx++];
        g = sdbuffer[buffidx++];
        r = sdbuffer[buffidx++];
        lcdbuffer[lcdidx++] = tft.color565(r,g,b);
}// end pixel
}// end scanline
// Write any remaining data to LCD
if(lcdidx >0){
        tft.pushColors(lcdbuffer, lcdidx, first);
}

}

}

}

```

```
    bmpFile.close();
if(!goodBmp) tft.println(F("BMP format not recognized.));
}

// These read 16- and 32-bit types from the SD card file.
// BMP data is stored little-endian, Arduino is little-endian too.
// May need to reverse subscript order if porting elsewhere.

uint16_t read16(File f) {
    uint16_t result;
    ((uint8_t *)&result)[0]= f.read();// LSB
    ((uint8_t *)&result)[1]= f.read();// MSB
    return result;
}

uint32_t read32(File f) {
    uint32_t result;
    ((uint8_t *)&result)[0]= f.read();// LSB
    ((uint8_t *)&result)[1]= f.read();
    ((uint8_t *)&result)[2]= f.read();
    ((uint8_t *)&result)[3]= f.read();// MSB
    return result;
}

[/code]
```

11 VERBINDUNGSTECHNIK

Bei der Verbindungstechnik hatten wir zwei Herausforderungen zu meistern. Wir wollten, dass die Feedback-Box und die Lormen-Hand getrennt in einem Raum aufgestellt werden können und wenn möglich ohne eine direkte Drahtverbindung. Zudem wollten wir auch noch zusätzlich mit der Feedback-Box eine Verbindung zu einem Handy ermöglichen.

Zunächst legten wir die Verbindung für die Kommunikation mit dem Handy fest. Dabei entschlossen wir uns für eine Bluetooth-Verbindung. Für die Kommunikation mit der Lormen-Hand, wollten wir eine andere Schnittstelle nutzen, um unser technisches Wissen zu erweitern und um eine gleichzeitige Verbindung von Lormen Hand, Handy und Feedback-Box zu ermöglichen. Nach einigen Feldversuchen entschlossen wir uns, ebenfalls eine Funkverbindung zu nutzen. Als geeignete Variante hatten sich Radiofrequenz-Module erwiesen. Mit diesen ist eine ausreichende Reichweite und auch eine bidirektionale Kommunikation möglich.

Dadurch sind wir nun nicht mehr auf eine Verkabelung zwischen den Mikrocontrollern angewiesen, brauchen jedoch auch für Ausgabe und Eingabe jeweils eine eigene Spannungsversorgung.

11.1 RADIOFREQUENZ – MODUL



Abbildung 64: Radiofrequenz-Modul 1

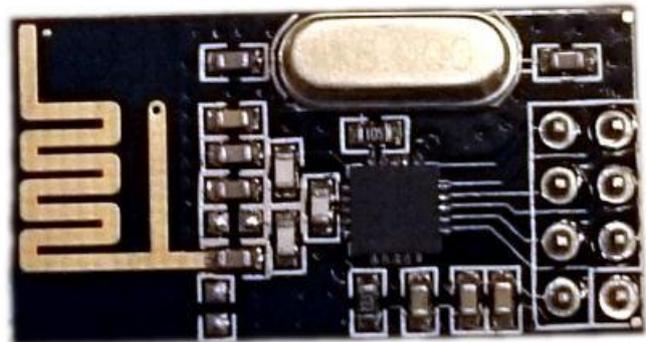


Abbildung 63: Radiofrequenz-Modul 2

Bei dem Radiofrequenz-Modul handelt es sich um ein NRF24101-Modul. Mit diesem Modul ist es uns möglich sowohl Daten zu senden als auch zu empfangen und das auch mit mehr als nur zwei Teilnehmern. In eigenen Versuchen haben wir festgestellt, dass die Übermittlung auch in größeren Klassenräumen gut funktioniert. Wir verwenden das Radiofrequenz-Modul um die Lormen-Hand mit der Feedback-Box zu vernetzen.

11.1.1 SPEZIFIKATION

Übersicht (itead, 2015)

Dieses Modul nutzt ein 2.4GHz Sende- und Empfangsgerät von Nordic Semiconductor

Eigenschaften:

Spannung:3-3.6V (Empfohlen 3.3V)

Maximale Ausgangsleistung: +20dBm

Abstrahlungsleistung (Spitze): 115mA

Empfangsleistung (Spitze): 45mA

Stromsparmmodus: 4.2µA

Zusätzliche Informationen

Modell Typ Kabellos

Gewicht 6 Gramm

Modell IM120606002

Abmessungen 2.9 x 1.5 x 1.2cm

Spannung Digital 3.3V

Netzteil Extern 3.3V

Version1.0

11.1.2 VERBINDUNGSSCHNITTSTELLE

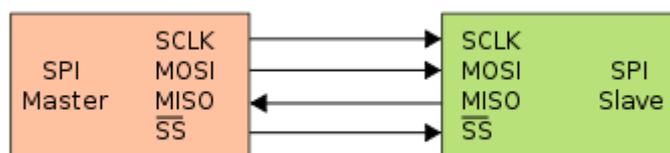


Abbildung 65: SPI Aufbau (Wikipedia, 2016b)

Die Radiofrequenz-Module verwenden zur Kommunikation mit dem Mikrocontroller das SPI (Serial Peripheral Interface) Protokoll. Das SPI Protokoll ermöglicht den Aufbau von Schaltungen nachdem Master-Slave-Prinzip. (Wikipedia, 2016b)



Abbildung 66: Bluetooth-Modul

Bluetooth ist eine der meist verbreiteten Arten Daten zwischen Handy und anderen Geräten zu übertragen. Es ermöglicht uns eine Peer to Peer Verbindung aufzubauen und Daten vom Handy zum Mikrocontroller zu schicken und umgekehrt. Das Bluetooth-Modul wird in unserem Fall benötigt, um die Android-App „Lormen“ mit der Feedback-Box zu vernetzen.

11.2.1 SPEZIFIKATION

(Reutlingen-university, 2016)

PIN1	UART_TXD , TTL/CMOS level, UART Data output
PIN2	UART_RXD, TTL/COMS level, s UART Data input
PIN11	RESET
PIN12	VCC, voltage supply for logic, the st andard voltage is 3.3V, and can work at 3.0-4.2V
PIN13	GND
PIN22	GND
PIN24	LED

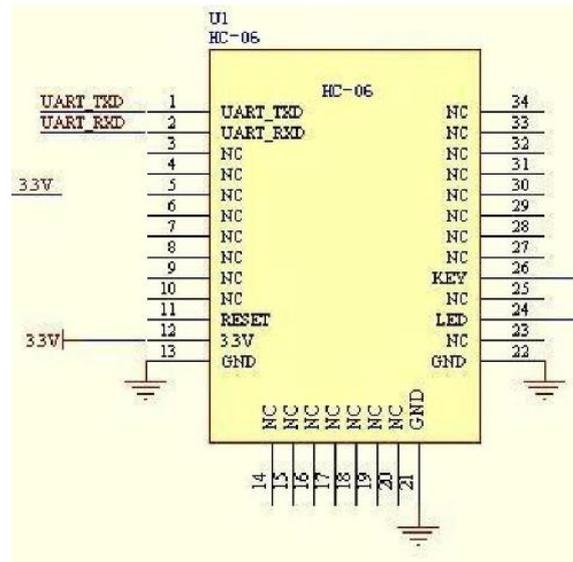


Abbildung 67: HC-06 Bluetooth-Modul (Reutlingen-university, 2016)

11.2.2 VERBINDUNGS-SCHNITTSTELLE

Die Radiofrequenz-Module verwenden zur Kommunikation mit dem Mikrocontroller die UART-Schnittstelle (Universal Asynchroun Receiver Transmitter). Die Schnittstelle kann zum Senden und Empfangen von Daten genutzt werden. Es ist eine der Standardchnittstellen an Computern und Mikrocontrollern. (mikrocontroller.net, 2016)

12 PROJEKTMANAGEMENT

12.1 ZUSTÄNDIGKEITEN DER PROJEKTMITGLIEDER

Alexander Bachmeier:

Vorstudien, Entwicklung, Hardware, Software, Mechanischer Aufbau, Dokumentation, **Koordination**

Michael Eder:

Vorstudien, Entwicklung, Hardware, Software, Mechanischer Aufbau, Dokumentation, **Projektmanagement**

Martin Plattner:

Vorstudien, Entwicklung, Hardware, Software, Mechanischer Aufbau, Dokumentation, **Endausfertigung der Dokumentation**

12.2 MEILENSTEINE

1 Lormen-Hand

- 1.1 Vorstudien zur Ermittlung der zu verwendenden Technologien
- 1.2 Kunststoffhand fertigen
- 1.3 Code für Mikrocontroller erstellen
- 1.4 Aufbau und Funktionstest mittels Serial Monitor und Breadboard

2 Feedback-Box

- 2.1 Vorstudien zur Ermittlung der zu verwendenden Technologien
- 2.2 Code für Mikrocontroller erstellen
- 2.3 Verknüpfung mit Lorm-Hand herstellen sowie Aufbau und Funktionstest mittels Breadboard

3 Haptisches Ausgabegerät

- 3.1 Vorstudien mit Mini Vibratoren
- 3.2 Bei erfolgreicher Vorstudie (optional)
 - 3.2.1 Namensgebung
 - 3.2.2 Entwicklung und Bau des Ausgabegeräts
 - 3.2.3 Integration in das bestehende System
 - 3.2.4 Funktionstest

- 4 (Optional)
 - 4.1 zusätzliches Eingabesystem via Bluetooth konzipieren und implementieren
 - 4.2 Funktionstest des bestehenden Systems
- 5 PCB
 - 5.1 PCB Design und Entflechtung
 - 5.2 PCB Fertigung
 - 5.3 PCB Bestückung
- 6 Mechanischer Aufbau
 - 6.1 Systemkomponenten verbinden
 - 6.2 Letzter Funktionstest
- 7 Dokumentation

12.3 BREAK EVEN

Um den Break Even Point unseres Kommunikationsgerätes dazustellen haben wir die fiktive BEP GmbH gegründet. Hierbei handelt es sich um eine GmbH welche zu gleichen Teilen zwischen Hr. Bachmeier Alexander, Hr. Eder Michael und Hr. Plattner Martin aufgeteilt ist. Anbei finden sie eine Auflistung aller relevanten Kosten.

12.3.1 KOSTEN DER GMBH/MONAT (FIXE KOSTEN)

Aufwand	Betrag/Monat	Anzahl	Summe
Lohn	3.000 €	3	9.000 €
Lohnnebenkosten	925 €	3	2.775 €
Miete 60qm ²	700 €	1	700 €
Strom	120 €	1	120 €
Wasser/Heizung	90 €	1	90 €
Berufshaftpflichtversicherung	161 €	1	161 €
Steuerberater	300 €	1	300 €
Leasing für Dienstwagen	350 €	1	350 €
Treibstoff	250 €	1	250 €
Autoversicherung	150 €	1	150 €
Kontoführung	20 €	1	20 €
Registrierkasse	22 €	1	22 €
Telefon & Internet	100 €	1	100 €
Gesamt:			14.038,00 €

12.3.2 MATERIALKOSTEN (VARIABLE KOSTEN)

Materialkosten	Preis
Feedback-Box	188,97 €
Lormen-Hand	317,11 €
Haptische-Ausgabe	98,98 €
Gesamt:	604,06 €

12.3.3 BERECHNUNG DES VERKAUFSPREISES

Verkaufspreis		
Arbeitszeit und Herstellungskosten: (Annahme 5 Stk/Monat)		
Fixkosten/Anzahl der produzierten Einheiten		2.807,60 €
Material		604,06 €
Kalkulation Selbstkosten		3.411,66 €
& Gewinn in Prozent	10%	341,17 €
Barverkaufspreis		3.752,83 €
& Skonto in %	10%	375,28 €
Zielverkaufspreis		4.128,11 €
& Rabatt in Prozent	20%	825,62 €
Zwischensumme		4.953,73 €
Forschung & Entwicklung	20%	990,75 €
Listenverkaufspreis Netto		5.944,48 €
MwSt.	20%	1.188,90 €
	Verkaufspreis	7.133,37 €

Berechnung des Deckungsbeitrags:

$$\text{Listenverkaufspreis} - \text{Variable Kosten} = \text{Deckungsbeitrag}$$

$$5944,48\text{€} - 604,06\text{€} = 5340,42\text{€}$$

Berechnung des Gewinnschwellenmenge:

$$\frac{\text{Fixe Kosten}}{\text{Deckungsbeitrag pro Stück}} = \text{Gewinnschwellenmenge}$$

$$\frac{14038\text{€}}{5340,42\text{€}} = 2,63 \text{ Stk.}$$

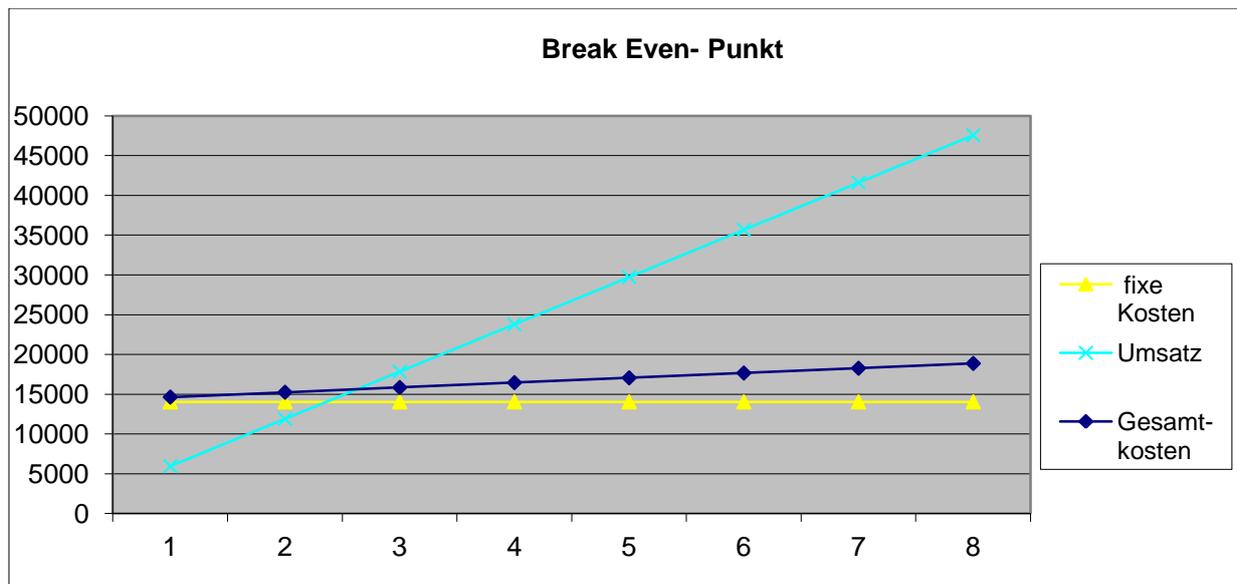
Berechnung des Gewinnschwellenumsatzes:

$$\frac{\text{Fixe Kosten}}{\text{Deckungsbetrag pro Stück}} * \text{Listenverkaufspreis} = \text{Gewinnschwellenumsatz}$$

$$\frac{14038\text{€}}{5340,42\text{€}} * 5944,48\text{€} = 15625,85\text{€}$$

Bei einem Absatz von 2,63 Stk. im Monat (\Rightarrow 15625,85€) wird die Vollkostendeckung erreicht.

Der Break Even Point grafisch dargestellt:



Diese Break Even Analyse macht deutlich das es sich sehrwohl lohnen würde eine GmbH im Bereich Biomedizin zu eröffnen. Die Gewinnmargen auf Produkte in diesem Bereich sind sehr hoch. Leider hat man aber nur ein sehr eingeschränktes Kundenfeld was die positiven Aussichten wieder ein wenig trübt.

Wenn man mit der Firma klein anfängt, die Fixkosten niedrig hält und eventuell in der ersten Zeit auf ein eigenes Büro verzichtet und stattdessen das Produkt in den eigenen vier Wänden herstellt könnte es durchaus ein angenehmes finanzielles Zubrot jedes Monat sein.

12.4 TIMELINE

Anbei eine Auflistung der geleisteten Stunden jedes Diplomprojekt Teilnehmers.

Datum	Name	Thema	Dauer in Std.
Freitag, 4. September 2015	Bachmeier Alexander	Thema Überlegen für Diplomarbeit	2
Freitag, 4. September 2015	Eder Michael	Thema Überlegen für Diplomarbeit	2
Freitag, 4. September 2015	Plattner Martin	Thema Überlegen für Diplomarbeit	2
Samstag, 5. September 2015	Bachmeier Alexander	Vorstudie: Wie kann man eine "Lormenhand" realisieren?	2,5
Samstag, 5. September 2015	Eder Michael	Vorstudie: Wie kann man eine "Lormenhand" realisieren?	2,5
Samstag, 5. September 2015	Plattner Martin	Vorstudie: Wie kann man eine "Lormenhand" realisieren?	2,5
Sonntag, 6. September 2015	Bachmeier Alexander	Prototyp aus Mauspad und Alufolie erstellt	3

Sonntag, 6. September 2015	Eder Michael	Prototyp für gut befunden und die Idee weitergedacht	2
Sonntag, 6. September 2015	Plattner Martin	Grober Umfang des Projektes wurde festgelegt	2
Montag, 7. September 2015	Bachmeier Alexander	Ideen gesammelt was das Projekt alles umfasst	1,5
Montag, 7. September 2015	Eder Michael	Ideen gesammelt was das Projekt alles umfasst	1,5
Montag, 7. September 2015	Plattner Martin	Projekt wurde in Stichworten niedergeschrieben	1,5
Donnerstag, 10. September 2015	Bachmeier Alexander	Entscheidung für Betreuer	1
Donnerstag, 10. September 2015	Eder Michael	Entscheidung für Betreuer	1
Donnerstag, 10. September 2015	Plattner Martin	Entscheidung für Betreuer	1
Freitag, 11. September 2015	Bachmeier Alexander	Technische Machbarkeit wurde besprochen	2
Freitag, 11. September 2015	Eder Michael	Eventuelles Zubehör wurde besprochen	2
Freitag, 11. September 2015	Plattner Martin	Funktionen wurden besprochen	2
Samstag, 12. September 2015	Plattner Martin	Sämtliche besprochenen Fakten über das Projekt wurden in ganze Sätze gefasst um sich auf ein Ziel konzentrieren zu können	3
Montag, 14. September 2015	Bachmeier Alexander	Mail verfasst und Hr. Eckl gefragt ob er unser Betreuer sein möchte	0,75
Montag, 14. September 2015	Eder Michael	Mail verfasst und Hr. Eckl gefragt ob er unser Betreuer sein möchte	0,75
Montag, 14. September 2015	Plattner Martin	Mail verfasst und Hr. Eckl gefragt ob er unser Betreuer sein möchte	0,75
Montag, 21. September 2015	Bachmeier Alexander	Hr. Eckl das Projekt erklärt	1
Montag, 21. September 2015	Eder Michael	Hr. Eckl das Projekt erklärt	1
Montag, 21. September 2015	Plattner Martin	Hr. Eckl das Projekt erklärt	1
Donnerstag, 24. September 2015	Bachmeier Alexander	Hat sich überlegt wie man eine Hand nachbauen kann.	3
Donnerstag, 24. September 2015	Eder Michael	Hat sich Gedanken über die möglichen Ausgabegeräte gemacht	3
Donnerstag, 24. September 2015	Plattner Martin	Grundgerüst des Diplomarbeits erstellt.	3
Freitag, 25. September 2015	Bachmeier Alexander	1 Prototyp auf einem Mauspad wurde erstellt	5
Freitag, 25. September 2015	Eder Michael	1 Prototyp auf einem Mauspad wurde erstellt	5
Freitag, 25. September 2015	Plattner Martin	1 Prototyp auf einem Mauspad wurde erstellt	5
Samstag, 26. September 2015	Bachmeier Alexander	1 Prototyp wurde für ausgiebig getestet	2
Samstag, 26. September 2015	Eder Michael	1 Prototyp wurde für ausgiebig getestet	2

Samstag, 26. September 2015	Plattner Martin	1 Prototyp wurde für ausgiebig getestet	2
Sonntag, 27. September 2015	Bachmeier Alexander	Nächster Prototyp wurde überlegt	4
Sonntag, 27. September 2015	Eder Michael	Nächster Prototyp wurde überlegt	4
Sonntag, 27. September 2015	Plattner Martin	Nächster Prototyp wurde überlegt	4
Montag, 28. September 2015	Bachmeier Alexander	Informationen gesammelt über Epoxidharz	4
Montag, 28. September 2015	Eder Michael	Form zum gießen überlegt	4
Montag, 28. September 2015	Plattner Martin	Material für Negativabdruck überlegt	4
Donnerstag, 1. Oktober 2015	Bachmeier Alexander	Rücksprache mit Hr. Eckl gehalten bzgl. der Verwendung von Epoxidharz	4
Donnerstag, 1. Oktober 2015	Eder Michael	Rücksprache mit Hr. Eckl gehalten bzgl. der Verwendung von Epoxidharz	4
Donnerstag, 1. Oktober 2015	Plattner Martin	Rücksprache mit Hr. Eckl gehalten bzgl. der Verwendung von Epoxidharz	4
Freitag, 2. Oktober 2015	Bachmeier Alexander	Benötigte Materialien gekauft	2
Freitag, 2. Oktober 2015	Eder Michael	Benötigte Materialien gekauft	2
Freitag, 2. Oktober 2015	Plattner Martin	Benötigte Materialien gekauft	2
Samstag, 3. Oktober 2015	Bachmeier Alexander	2ter Prototyp erstellt	6
Samstag, 3. Oktober 2015	Eder Michael	2ter Prototyp erstellt	6
Samstag, 3. Oktober 2015	Plattner Martin	2ter Prototyp erstellt	6
Sonntag, 4. Oktober 2015	Bachmeier Alexander	2ter Prototyp wurde fertiggestellt	3
Sonntag, 4. Oktober 2015	Eder Michael	2ter Prototyp wurde fertiggestellt	3
Sonntag, 4. Oktober 2015	Plattner Martin	2ter Prototyp wurde fertiggestellt	3
Montag, 5. Oktober 2015	Bachmeier Alexander	2ter Prototyp wurde getestet	4
Montag, 5. Oktober 2015	Eder Michael	2ter Prototyp wurde getestet	4
Montag, 5. Oktober 2015	Plattner Martin	2ter Prototyp wurde getestet	4
Donnerstag, 8. Oktober 2015	Bachmeier Alexander	Prototyp wurde grundsätzlich für gut befunden ist aber verbesserungswürdig	3
Donnerstag, 8. Oktober 2015	Eder Michael	Prototyp wurde grundsätzlich für gut befunden ist aber verbesserungswürdig	3
Donnerstag, 8. Oktober 2015	Plattner Martin	Prototyp wurde grundsätzlich für gut befunden ist aber verbesserungswürdig	3
Freitag, 9. Oktober 2015	Bachmeier Alexander	Code für Arduino wurde entworfen	6
Freitag, 9. Oktober 2015	Eder Michael	Code für Arduino wurde entworfen	6
Freitag, 9. Oktober 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4

Samstag, 10. Oktober 2015	Bachmeier Alexander	Erste Tests mit dem Prototypen und dem Arduino	5
Samstag, 10. Oktober 2015	Eder Michael	Erste Tests mit dem Prototypen und dem Arduino	5
Samstag, 10. Oktober 2015	Plattner Martin	Erste Tests mit dem Prototypen und dem Arduino	5
Sonntag, 11. Oktober 2015	Bachmeier Alexander	Tests weitgehend erfolgreich, aber Prototyp wurde für verbesserungswürdig befunden	3,25
Sonntag, 11. Oktober 2015	Eder Michael	Tests weitgehend erfolgreich, aber Prototyp wurde für verbesserungswürdig befunden	3,25
Sonntag, 11. Oktober 2015	Plattner Martin	Tests weitgehend erfolgreich, aber Prototyp wurde für verbesserungswürdig befunden	3,25
Montag, 12. Oktober 2015	Bachmeier Alexander	Code wurde ergänzt	4
Montag, 12. Oktober 2015	Eder Michael	Code wurde ergänzt	4
Montag, 12. Oktober 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4
Donnerstag, 15. Oktober 2015	Bachmeier Alexander	Code wurde ergänzt	4
Donnerstag, 15. Oktober 2015	Eder Michael	Code wurde ergänzt	4
Donnerstag, 15. Oktober 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4
Freitag, 16. Oktober 2015	Bachmeier Alexander	Material für neuen Prototyp wurde gekauft	1,5
Freitag, 16. Oktober 2015	Eder Michael	Material für neuen Prototyp wurde gekauft	1,5
Freitag, 16. Oktober 2015	Plattner Martin	Material für neuen Prototyp wurde gekauft	1,5
Samstag, 17. Oktober 2015	Bachmeier Alexander	3ter Prototyp wurde gegossen	6
Samstag, 17. Oktober 2015	Eder Michael	3ter Prototyp wurde gegossen	6
Samstag, 17. Oktober 2015	Plattner Martin	3ter Prototyp wurde gegossen	6
Sonntag, 18. Oktober 2015	Bachmeier Alexander	Verkabelung und Sensorflächen für Prototyp wurde überlegt	3
Sonntag, 18. Oktober 2015	Eder Michael	Verkabelung und Sensorflächen für Prototyp wurde überlegt	3
Sonntag, 18. Oktober 2015	Plattner Martin	Verkabelung und Sensorflächen für Prototyp wurde überlegt	3
Montag, 19. Oktober 2015	Bachmeier Alexander	3ter Prototyp wurde verkabelt und Sensorflächen wurde erstellt	5
Montag, 19. Oktober 2015	Eder Michael	3ter Prototyp wurde verkabelt und Sensorflächen wurde erstellt	5
Montag, 19. Oktober 2015	Plattner Martin	3ter Prototyp wurde verkabelt und Sensorflächen wurde erstellt	5

Donnerstag, 22. Oktober 2015	Bachmeier Alexander	3ter Prototyp wurde ausgiebig getestet - Widerstände für Sensorflächen wurden bestimmt	4
Donnerstag, 22. Oktober 2015	Eder Michael	3ter Prototyp wurde ausgiebig getestet - Widerstände für Sensorflächen wurden bestimmt	4
Donnerstag, 22. Oktober 2015	Plattner Martin	3ter Prototyp wurde ausgiebig getestet - Widerstände für Sensorflächen wurden bestimmt	4
Freitag, 23. Oktober 2015	Bachmeier Alexander	Print für Lormen Hand wurde ausgearbeitet in Eagle	7
Freitag, 23. Oktober 2015	Eder Michael	Print für Lormen Hand wurde ausgearbeitet in Eagle	7
Freitag, 23. Oktober 2015	Plattner Martin	Print für Lormen Hand wurde ausgearbeitet in Eagle	7
Samstag, 24. Oktober 2015	Bachmeier Alexander	Print wurde verbessert	1,5
Samstag, 24. Oktober 2015	Eder Michael	Print wurde verbessert	1,5
Samstag, 24. Oktober 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	0,75
Montag, 26. Oktober 2015	Bachmeier Alexander	Print wurde in der Schule geätzt und alle Löcher wurde gebohrt	2
Montag, 26. Oktober 2015	Eder Michael	Print wurde in der Schule geätzt und alle Löcher wurde gebohrt	2
Montag, 26. Oktober 2015	Plattner Martin	Print wurde in der Schule geätzt und alle Löcher wurde gebohrt	2
Donnerstag, 29. Oktober 2015	Bachmeier Alexander	Bauteile wurden bestellt	1,75
Donnerstag, 29. Oktober 2015	Eder Michael	Bauteile wurden bestellt	1,75
Donnerstag, 29. Oktober 2015	Plattner Martin	Bauteile wurden bestellt	1,75
Donnerstag, 5. November 2015	Bachmeier Alexander	Diplomarbeit wurde weitergeschrieben	2,25
Donnerstag, 5. November 2015	Eder Michael	Platine wurde gelötet	2,25
Donnerstag, 5. November 2015	Plattner Martin	Platine wurde gelötet	2,25
Freitag, 6. November 2015	Bachmeier Alexander	Platine wurde mit Arduino verbunden und mit Lormen-Hand getestet	3
Freitag, 6. November 2015	Eder Michael	Platine wurde mit Arduino verbunden und mit Lormen-Hand getestet	3
Freitag, 6. November 2015	Plattner Martin	Platine wurde mit Arduino verbunden und mit Lormen-Hand getestet	3
Donnerstag, 8. Oktober 2015	Bachmeier Alexander	Arduino Code wurde angepasst	4
Donnerstag, 8. Oktober 2015	Eder Michael	Arduino Code wurde angepasst	4
Donnerstag, 8. Oktober 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4
Freitag, 9. Oktober 2015	Bachmeier Alexander	Arten der Ausgabe überlegt und definiert	4,5

Freitag, 9. Oktober 2015	Eder Michael	Arten der Ausgabe überlegt und definiert	4,5
Freitag, 9. Oktober 2015	Plattner Martin	Arten der Ausgabe überlegt und definiert	4,5
Samstag, 10. Oktober 2015	Bachmeier Alexander	Benötigtes Material für Feedbackbox bestellt	1
Samstag, 10. Oktober 2015	Eder Michael	Benötigtes Material für Feedbackbox bestellt	1
Samstag, 10. Oktober 2015	Plattner Martin	Benötigtes Material für Feedbackbox bestellt	1
Freitag, 16. Oktober 2015	Bachmeier Alexander	Platine für Feedbackbox in Eagle erstellt	3
Freitag, 16. Oktober 2015	Eder Michael	Platine für Feedbackbox in Eagle erstellt	3
Freitag, 16. Oktober 2015	Plattner Martin	Platine für Feedbackbox in Eagle erstellt	3
Montag, 19. Oktober 2015	Bachmeier Alexander	Platine für Feedbackbox geätzt und gebohrt	4
Montag, 19. Oktober 2015	Eder Michael	Platine für Feedbackbox geätzt und gebohrt	4
Montag, 19. Oktober 2015	Plattner Martin	Platine für Feedbackbox geätzt und gebohrt	4
Donnerstag, 22. Oktober 2015	Bachmeier Alexander	Platine wurde bestückt und gelötet	4
Donnerstag, 22. Oktober 2015	Eder Michael	Platine wurde bestückt und gelötet	4
Donnerstag, 22. Oktober 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4
Donnerstag, 29. Oktober 2015	Bachmeier Alexander	Platine und Arduino wurden in Feedback Box eingebaut	2
Donnerstag, 29. Oktober 2015	Eder Michael	Platine und Arduino wurden in Feedback Box eingebaut	2
Donnerstag, 29. Oktober 2015	Plattner Martin	Platine und Arduino wurden in Feedback Box eingebaut	2
Montag, 2. November 2015	Bachmeier Alexander	Funktionen wurden getestet und Code angepasst	3
Montag, 2. November 2015	Eder Michael	Funktionen wurden getestet und Code angepasst	3
Montag, 2. November 2015	Plattner Martin	Funktionen wurden getestet und Code angepasst	3
Donnerstag, 5. November 2015	Bachmeier Alexander	Übertragung zwischen Lormen-Hand und Feedbackbox überlegt	1,5
Donnerstag, 5. November 2015	Eder Michael	Übertragung zwischen Lormen-Hand und Feedbackbox überlegt	1,5
Donnerstag, 5. November 2015	Plattner Martin	Übertragung zwischen Lormen-Hand und Feedbackbox überlegt	1,5
Freitag, 6. November 2015	Bachmeier Alexander	Hardware für Übertragung bestellt	1
Freitag, 6. November 2015	Eder Michael	Hardware für Übertragung bestellt	1
Freitag, 6. November 2015	Plattner Martin	Hardware für Übertragung bestellt	1

Freitag, 13. November 2015	Bachmeier Alexander	Funk Modul provisorisch verkabelt und getestet	2
Freitag, 13. November 2015	Eder Michael	Funk Modul provisorisch verkabelt und getestet	2
Freitag, 13. November 2015	Plattner Martin	Funk Modul provisorisch verkabelt und getestet	2
Samstag, 14. November 2015	Bachmeier Alexander	Code für Funk Modul verbessert	3
Samstag, 14. November 2015	Eder Michael	Code für Funk Modul verbessert	3
Samstag, 14. November 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	3
Sonntag, 15. November 2015	Bachmeier Alexander	Überlegung ein Smartphone in das Projekt einzubinden um	1
Sonntag, 15. November 2015	Eder Michael	Überlegung ein Smartphone in das Projekt einzubinden um	1
Sonntag, 15. November 2015	Plattner Martin	Überlegung ein Smartphone in das Projekt einzubinden um	1
Donnerstag, 19. November 2015	Bachmeier Alexander	Hardware für Smartphone Übertragung bestellen	1
Donnerstag, 19. November 2015	Eder Michael	Diplomarbeit wurde weitergeschrieben	
Donnerstag, 19. November 2015	Plattner Martin	Hardware für Smartphone Übertragung bestellen	1
Donnerstag, 26. November 2015	Bachmeier Alexander	Bluetooth Modul wird provisorisch mit der Platine in der Feedback Box verbunden	3
Donnerstag, 26. November 2015	Eder Michael	Bluetooth Modul wird provisorisch mit der Platine in der Feedback Box verbunden	4
Donnerstag, 26. November 2015	Plattner Martin	Bluetooth Modul wird provisorisch mit der Platine in der Feedback Box verbunden	4
Samstag, 28. November 2015	Bachmeier Alexander	Arduino App für Eingabe normaler Schrift	5
Samstag, 28. November 2015	Eder Michael	Arduino Code wurde angepasst	6
Samstag, 28. November 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4,75
Samstag, 12. Dezember 2015	Bachmeier Alexander	Lormen-Hand wurde in der Diplomarbeit beschrieben	4
Samstag, 12. Dezember 2015	Eder Michael	Feedback Box Funktion wurde in der Diplomarbeit beschrieben	4
Samstag, 12. Dezember 2015	Plattner Martin	Stückliste wurde geschrieben	4
Freitag, 18. Dezember 2015	Bachmeier Alexander	Gehäuse für Lormen-Hand wurde entworfen	4
Freitag, 18. Dezember 2015	Eder Michael	Gehäuse für Feedback-Box wurde entworfen	4
Freitag, 18. Dezember 2015	Plattner Martin	Diplomarbeit wurde weitergeschrieben	4

Sonntag, 20. Dezember 2015	Bachmeier Alexander	Material für beide Gehäuse wurde bestellt	1
Sonntag, 20. Dezember 2015	Eder Michael	Material für beide Gehäuse wurde bestellt	1
Sonntag, 20. Dezember 2015	Plattner Martin	Material für beide Gehäuse wurde bestellt	1
Samstag, 9. Januar 2016	Bachmeier Alexander	Gehäuse Lormenhand wurde gebaut	3,5
Samstag, 9. Januar 2016	Eder Michael	Gehäuse Lormenhand wurde gebaut	3,5
Samstag, 9. Januar 2016	Plattner Martin	Gehäuse Lormenhand wurde gebaut	3,5
Samstag, 16. Januar 2016	Bachmeier Alexander	Gehäuse für Feedback-Box wurde gebaut	3,5
Samstag, 16. Januar 2016	Eder Michael	Gehäuse für Feedback-Box wurde gebaut	3,5
Samstag, 16. Januar 2016	Plattner Martin	Gehäuse für Feedback-Box wurde gebaut	3,5
Samstag, 30. Januar 2016	Bachmeier Alexander	Finaler Test der kompletten Diplomarbeit	5
Samstag, 30. Januar 2016	Eder Michael	Finaler Test der kompletten Diplomarbeit	5
Samstag, 30. Januar 2016	Plattner Martin	Finaler Test der kompletten Diplomarbeit	5
Donnerstag, 19. Mai 2016	Bachmeier Alexander	Gemeinsame Kontrolle der finalen Diplomarbeitsdokumentation	3
Donnerstag, 19. Mai 2016	Eder Michael	Gemeinsame Kontrolle der finalen Diplomarbeitsdokumentation	3
Donnerstag, 19. Mai 2016	Plattner Martin	Gemeinsame Kontrolle der finalen Diplomarbeitsdokumentation	3

Summe der geleisteten Stunden pro Diplomarbeitteilnehmer

Name	Geleistete Stunden	
Bachmeier Alexander	180,00	Stunden
Eder Michael	180,00	Stunden
Plattner Martin	180,00	Stunden

13 EINSATZMÖGLICHKEITEN

Unsere Applikation kann sehr vielfältig eingesetzt werden. So besteht unser System aus zwei Eingabe- und zwei Ausgabeschnittstellen, um das „Schreiben“ und das „Lesen“ sowohl Personen mit, als auch jenen ohne Sehbehinderung zu ermöglichen. Zunächst werden die beiden Eingabeschnittstellen beschrieben: Das „Schreiben“ kann einerseits an der Lormen-Hand mittels Lorm-Alphabet erfolgen. Hierfür streicht oder tippt der Nutzer/die Nutzerin mit seinem/ihrem Finger über bzw. auf die Kunststoffhand, wie er/sie es beim konventionellen Lormen – auf der Hand seines/ihres jeweiligen Gegenübers – tun würde. Nutzer/Nutzerinnen, die das Lorm-Alphabet nicht beherrschen, können mit ihrem Bluetooth-fähigen Smartphone ihre Eingabe über eine App tätigen. Die Eingabe, unabhängig von der Eingabemethode (Kunststoffhand oder App), wird dann von dem System verarbeitet und ausgegeben.

Im Folgenden werden die beiden Ausgabeschnittstellen behandelt:

Bei der Ausgabe haben wir uns für ein handelsübliches Display und eine weitere, haptische Ausgabe in Form eines modifizierten Handschuhs entschieden. Die Ausgaben erfolgen parallel zueinander und müssen somit nicht vor Benutzung vorkonfiguriert werden. Am Display werden die Buchstaben bzw. Wörter ausgegeben und ermöglicht es so sehenden Personen die Eingaben zu lesen. Für die haptische Ausgabe zieht der Nutzer/die Nutzerin einen Handschuh an, an dem einige Vibratoren angebracht sind. Erfolgt eine Eingabe die zum Ausgabegerät gesendet wird, so werden diese Vibratoren gezielt in einer zeitlichen Abfolge angesteuert. Bei der Positionierung und Ansteuerung der Vibratoren haben wir uns an das Lorm-Alphabet gehalten.

Die Kombination zweier so unterschiedlicher Methoden bewirkt, dass das System träge wird, weil es nicht möglich ist die Informationen ohne Verluste simultan zu übersetzen. Würde man beispielsweise einen Lorm-Buchstaben mit der Lormen-Hand schreiben, wäre es möglich, diesen zeitgleich und ungefiltert an die haptische Ausgabe zu senden. Hier würde jedoch der Mikrocontroller, bei einer schlampigen Eingabe, einen falschen oder gar keinen Buchstaben ermitteln. Dadurch wären die Informationen der Ausgabe an der optischen und an der haptischen Schnittstelle unterschiedlich. Um dies zu unterdrücken nehmen wir es in Kauf, dass die Ausgabe der Eingabe folgt und es dadurch zu Wartezeiten aufgrund der zeitlichen Abfolge der haptischen Ausgabe kommt. Je nach Einsatzgebiet ist es möglich, unsere Applikation an die gegebenen Umstände anzupassen, um eine bessere Performance zu erzielen.

In den folgenden Kapiteln werden verschiedene Einsatz- und Erweiterungsmöglichkeiten der eben beschriebenen Applikation vorgestellt.

13.1 DIE APPLIKATION ALS LERNSYSTEM

Mit der Applikation ist es möglich das Lormen zu erlernen. Der Nutzer/die Nutzerin kann auf der Lormen-Hand schreiben und die Eingabe auf dem Display der Feedback-Box überprüfen. So kann er/sie das Lorm-Alphabet spielerisch lernen und dabei den Wissensstand selbstständig überprüfen. Weiters ist es möglich, die Hand des Nutzers/der Nutzerin mit Hilfe der haptischen Ausgabe auf Buchstaben des Lorm-Alphabets zu sensibilisieren und das „Lesen“ zu üben.

13.1.1 LERNSYSTEM – MÖGLICHE ERWEITERUNGEN

Es wäre möglich, die Lormen-Hand mit einem separaten Display auszustatten. Dadurch könnte die Lormen-Hand autonom als Lernsystem eingesetzt werden. Weiters könnte man Lernspiele in den Programmcode einfügen, bei dem der Nutzer zufällig ausgewählte Wörter aus einer Datenbank nachschreiben kann. In Kombination mit einem Punktesystem, welches Fehler und/oder Zeit wertet, könnte man das Lernen noch spannender gestalten.

13.2 DIE APPLIKATION ALS SPRACHROHR – EIN WICHTIGER SCHRITT ZUR SELBSTSTÄNDIGKEIT

Mit der Applikation ist es taubblinden Personen möglich, sich auch ohne direkte Berührung einer weiteren Person bzw. weiterer Personen mitzuteilen.

13.2.1 „SCHREIBEN“

Die taubblinde Person kann mittels der Lormen-Hand Informationen an die Feedback-Box senden, die die Informationen bzw. den Text am Display ausgibt. Das könnte bei der Betreuung der taubblinden Personen zum Einsatz kommen, um die Qualifikationsansprüche an die Betreuer/Betreuerinnen zu senken. Somit könnte beispielsweise auch ein Zivildienstler einen Teil der Betreuung übernehmen, ohne zuvor das Lormen zu erlernen. Weiter könnte man diese Art der Kommunikation auch in Arztpraxen, Spitälern und Ämtern einsetzen. Das bietet taubblinden Personen die Möglichkeit, ihrem Gegenüber selbst ihr Anliegen zu schildern.

13.2.2 „LESEN“

Informationen, die zur Feedback-Box gesendet werden, werden nicht nur über ein Display ausgegeben. Parallel dazu werden Vibratoren, die sich auf einem Handschuh befinden, angesteuert. Diese Vibratoren haben die Aufgabe, die von der Feedback-Box empfangenen Buchstaben auszugeben. Dies geschieht durch gezieltes, zeitgesteuertes Ansteuern der einzelnen Vibratoren am Handschuh. Die Intensität der Vibratoren kann über ein Potentiometer auf der Platine individuell eingestellt werden. Somit kann eine taubblinde Person einer zweiten taubblinden Person ohne direkten Körperkontakt Nachrichten übermitteln. Einsetzen könnte man diese Applikation z.B., wenn eine der beiden Personen ein schwaches Immunsystem, oder eine ansteckende Erkrankung hat.

Es wäre leicht möglich mehrere Handschuhe zugleich anzusteuern. Hierfür müsste lediglich die Leiterplatine erweitert werden. Die Ausführung der Erweiterung obliegt jedoch den speziellen Kundenansprüchen. Man könnte die Platine für eine fixe Anzahl von haptischen Ausgabegeräten erweitern. Innovativer und angepasster wäre es jedoch, wenn man die Erweiterungen modular baut und einfach auf dem Mainboard übereinander steckt. Hierfür müsste man in der Planung nur berücksichtigen, wie viele haptische Ausgabegeräte erwünscht sind, um die maximale Leistung der Spannungsversorgung anzupassen. Die Erweiterung mehrerer haptischer Ausgabegeräte bietet die Möglichkeit, mehreren taubblinden Menschen zeitgleich Informationen zu übermitteln. Diese Art der Kommunikation ist derzeit noch einzigartig und ohne technische Hilfsmittel nicht machbar. Die Einsatzmöglichkeiten sind enorm und geben taubblinden Menschen ein Stück Selbstständigkeit zurück. Es wäre beispielsweise möglich, mehrere taubblinde Personen zeitgleich zu schulen oder zu informieren. So kann eine Person mehreren Personen unter anderem die aktuellen Tagesgeschehen aus der Zeitung übermitteln. Außerdem bietet es auch die Möglichkeit, mehrere taubblinde Menschen in geförderten Arbeitsstätten gleichzeitig über die geplanten Tätigkeiten zu informieren. Weiters wäre es möglich, die haptische Ausgabe mit weiteren Schnittstellen zu erweitern. Es könnte beispielsweise auch Feedback-Boxen geben, die mit einer Speichermedienschnittstelle, einem GSM (Telefonnetz)- oder Wifi (Internet)-Modul ausgestattet sind, um Informationen bzw. Texte auch anderweitig zu übertragen oder zu empfangen. So könnte man die Speichermedienschnittstelle nutzen, um ganze Bücher in das System zu laden. Dieses System könnte dann – vergleichbar mit einem Hörbuch – das Buch vortragen. Das GSM- oder Wifi-Shield könnte man nutzen, um Informationen über das Handynetz oder das Internet über weitere Strecken zu transportieren.

Weiters wäre es möglich, ganze Kommunikationsnetzwerke, bestehend aus mehreren Applikationen, zu bauen. So könnte beispielsweise ein Lehrer/ eine Lehrerin mehrere Klassen in einem Raum unterrichten. Durch gezielte Adressierung der Empfänger/innen könnte dies geschehen, ohne Schüler/innen der anderen Klassen zu stören. Die Schüler/innen könnten im Gegenzug auch Fragen direkt an den Lehrer/ die Lehrerin oder an die Klasse senden.

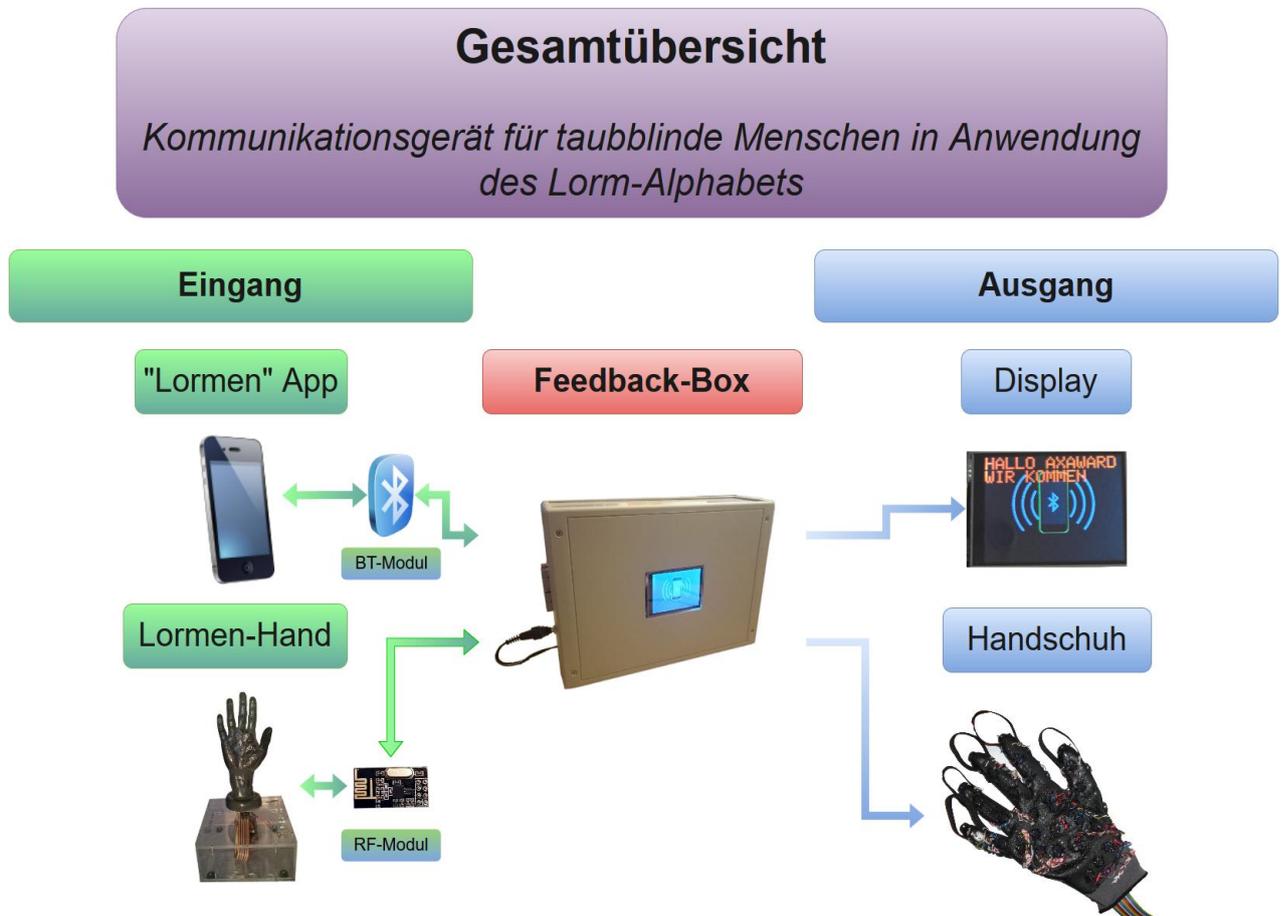


Abbildung 68: Gesamtübersicht

Anhand oben stehender Grafik lässt sich sehr schön erkennen, wie die einzelnen Komponenten mit einander zusammen spielen.

Die eingezeichneten Pfeile spiegeln den Informationsfluss wider.

Verbindungen werden mittels Bluetooth oder Radiofrequenz aufgebaut und Informationen werden in beide Richtungen gesendet. Dies geschieht sowohl von der Feedback-Box weg, als auch zur Feedback-Box hin.

Der Handschuh ist mittels Flachbandkabel an der Feedback-Box angeschlossen.

Das Display ist direkt im Gehäuse verbaut und verkabelt.

In den folgenden Kapiteln wird der genaue Einsatz der Applikation schrittweise beschrieben. Bei den Funktionskontrollen, welche wir mit Familienmitgliedern durchgeführt haben, erwies sich die Anleitung als hilfreich und ausreichend.



Abbildung 69: Blockschaltbild Eingabemöglichkeiten

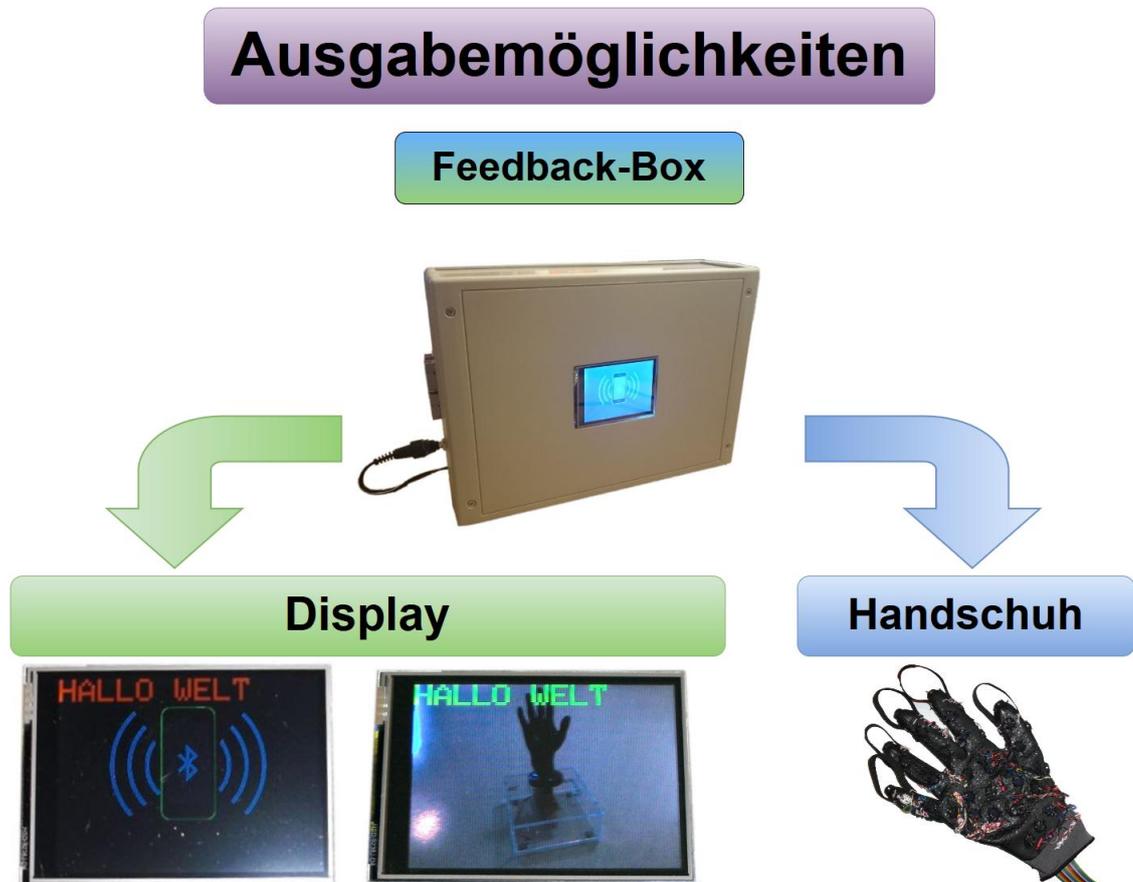


Abbildung 70: Blockschaltbild Ausgabemöglichkeiten

14.1 BEDIENUNG – EINGABEGERÄT LORMEN-HAND

Zunächst muss eine Verbindung für die Funktionserde hergestellt werden. Hierfür wird das mitgelieferte blaue Kabel sowohl mit einer Steckdose als auch mit der Laborbuchse rechts am Gehäuse verbunden. Sollte keine freie Steckdose oder keine fehlerfreie Erdung vorhanden sein, kann optional auch das Erdungsband –an Stelle des Erdungskabels mit Schukostecker - mit dem Körper des/der Nutzers/in und der Laborbuchse verbunden werden. Hierfür wird das Erdungsband am Handgelenk wie eine Uhr getragen. Dann kann das Netzteil an der Rückseite des Gehäuses angeschlossen werden. Nach etwa 5 Sekunden Wartezeit ist das Eingabegerät einsetzbar.

Zu Beginn ist es so eingestellt, dass die Buchstaben zeichenweise an das Ausgabegerät gesendet werden. Je nach Belieben kann das Senden auch wortweise eingestellt werden. Hierfür müssen die Sensoren an den Fingerspitzen (kleiner Finger und Zeigefinger – wie bei Eingabe „U“ und „E“) und der mittlere Sensor an dem Handgelenk (wie bei Eingabe „C“) gleichzeitig betätigt werden. Die Umstellung wird durch zweimaliges Vibrieren bestätigt (lang = wortweise, kurz = zeichenweise).

Geschrieben wird mit den Fingerflächen mit mäßigem Druck. Wird ein Buchstabe erkannt, so vibriert das Eingabegerät einmalig. Je nach oben beschriebener Einstellung werden einzelne Buchstaben automatisch und sofort (zeichenweise), oder nach einem Leerzeichen (wortweise) gesendet. Sobald das Ausgabegerät mit der Ausgabe fertig ist, meldet es das an das Eingabegerät zurück. Bei erfolgreicher Übertragung vibriert das Eingabegerät ein weiteres Mal, um dem/der Nutzer/in dies rückzumelden. Danach kann der/die Nutzer/in das Eingabegerät weiter benutzen. Sollte die Übertragung nicht fehlerfrei funktioniert haben, wird dies durch dreimaliges Vibrieren signalisiert. Mögliche Ursachen hierfür könnten Störsignale im Bereich der Funkmodule, ein abgeschaltetes Ausgabegerät, oder die gegenwärtige Verarbeitung einer vorhergehenden Eingabe sein.

Wir haben uns bei der Eingabe sehr genau an das Lorm-Alphabet gehalten. Jedoch haben wir zwei Unterschiede zum herkömmlichen Lormen eingebaut, sowohl um die Bedienung zu erleichtern als auch um die Ausgabegenauigkeit zu erhöhen. Hiervon sind das „K“ und das „Leerzeichen“ betroffen.

Beim „K“ wird anstelle des Tippens mit vier Fingern an der Handfläche nur ein gerades Wischen im oberen Teil der Handfläche (etwa Höhe „N“, von links nach rechts oder umgekehrt) benötigt.

Das Leerzeichen, das ein neues Wort signalisiert, wird anstelle von Wischen über die Handfläche mit einer senkrechten Wischbewegung von der Ringfingerfläche bis zum unteren Bereich der Handfläche eingegeben. Bei einer wortweisen Ausgabe wird mit dem Leerzeichen das Senden gestartet.



Abbildung 71: Display mit empfangenem Text

14.2 BEDIENUNG –EINGABE VIA BLUETOOTH

Zuerst muss der/die Nutzer/in die Bluetooth-Schnittstelle am Smartphone aktivieren und die App starten. Diese App ist nicht im öffentlichen App-Store verfügbar. Sie wird lediglich über das Entwicklerteam der Applikation (Bachmeier, Eder, Plattner) auf dem Smartphone installiert. Zu Beginn muss sich der/die Nutzer/in mit dem Bluetooth-Modul verbinden (Passwort: 1234). Geschieht dies erfolgreich, so werden die gekoppelten Geräte ausgeblendet und eine Tastatur erscheint. Nun kann der/die Nutzer/in Eingaben tätigen und diese mittels Leerzeichen an das Ausgabegerät übertragen. Solange das Ausgabegerät mit der Ausgabe beschäftigt ist, erscheint ein Wartebildschirm. Es kann keine weitere Eingabe erfolgen, solange das Ausgabegerät mit der Verarbeitung beschäftigt ist.

Der zu sendende Text wird beim Schreiben angezeigt. Sollte das Ausgabegerät mit der Verarbeitung von Informationen von der Lormen-Hand beschäftigt sein, wird der angezeigte Text automatisch gelöscht, um dies dem/der Nutzer/in zu signalisieren.

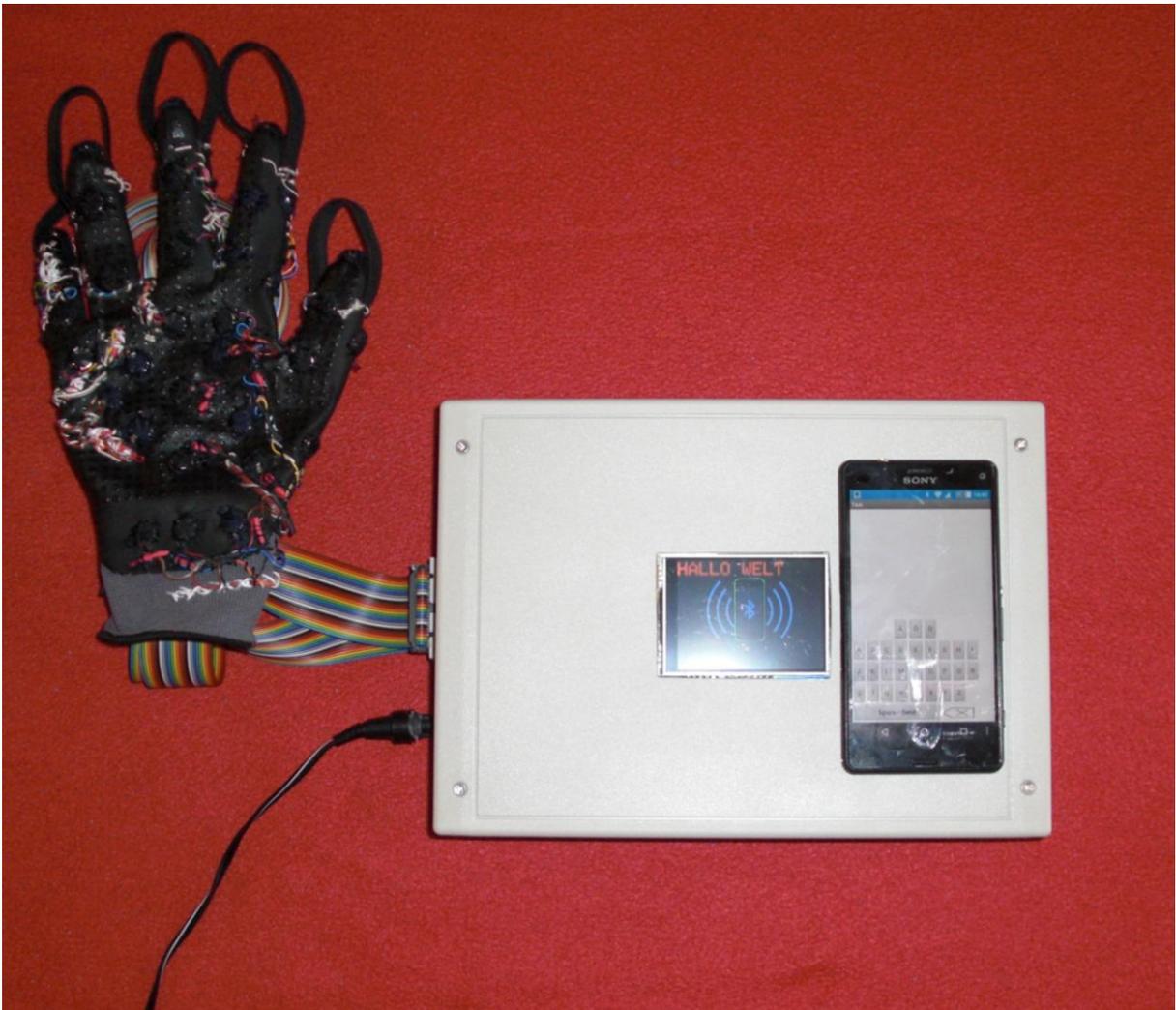


Abbildung 72: Eingabe mittels Android App

14.3 BEDIENUNG AUSGABEGERÄT–FEEDBACKBOX

- Platzieren Sie die Feedback-Box so, dass Sie vom Display angenehm ablesen können.
- Nun schließen Sie das Netzteil an eine 230V AC Steckdose an.
- Nehmen Sie den schmalen runden Stecker und schließen Sie ihn an der Feedback-Box an.
- Jetzt warten Sie bis das Display „Betriebsbereit“ anzeigt
 - Dieser Vorgang kann einige Sekunden dauern
- Optional lässt sich nun der Handschuh zur haptischen Ausgabe anstecken.
 - Ist der Handschuh angesteckt, so dauert die Ausgabe am Display länger, da die Ausgabe immer simultan zwischen Display und Handschuh stattfindet.
 - Beim Anstecken des Handschuhs ist auf die Beschriftung zu achten. Siehe 12.1.4
- Zum Abschalten einfach den Netzstecker wieder abziehen.

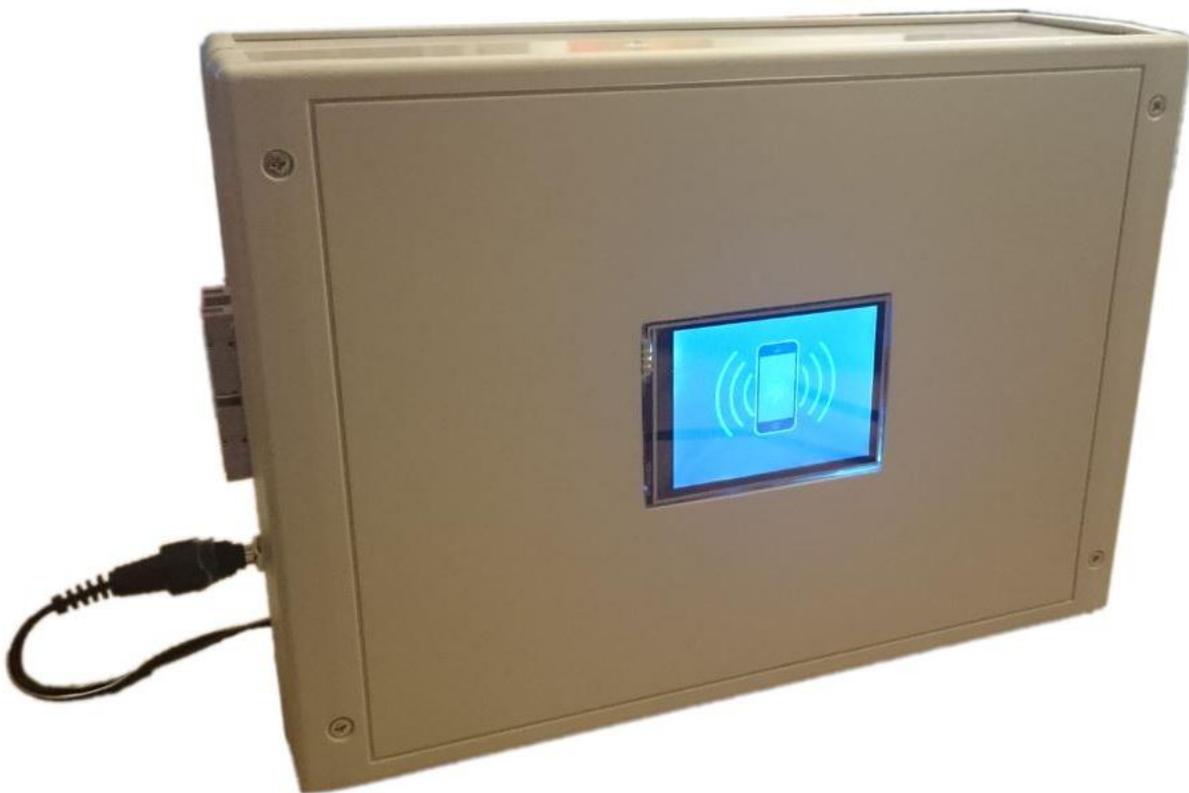


Abbildung 73: Feedback - Box

14.4 BEDIENUNG DER HAPTISCHEN AUSGABE - HANDSCHUH

Anlegen:

- Den Handschuh mittels der beiden gekennzeichneten Stecker an die dazugehörige Buchsen der Feedback-Box anschließen.
- Den Handschuh anziehen. Dabei ist zu beachten, dass nicht an der Verkabelung gezogen wird sondern am Stoff des Handschuhs.

Ablegen:

- Beim Ausziehen des Handschuhs sind die angebrachten Schlaufen an den Fingerspitzen zu verwenden. Man legt die Finger der freien Hand in die Schlaufen und zieht den Handschuh gerade von der Hand ab.
- Zuletzt entfernt man die beiden Stecker wieder von der Feedback-Box.



Abbildung 74: Haptische Ausgabe - Handschuh

15 LITERATURVERZEICHNIS

- adafruit. (2015). *adafruit: Vibrating Mini Motor Disc*. Retrieved Dezember 12, 2015, from <https://www.adafruit.com/product/1201>
- anderssehen. (2016). *anderssehen.at: WER ODER WAS IST LORM/EN?* Retrieved Mai 19, 2016, from <http://www.anderssehen.at/portrait/lormen.shtml>
- Badger, P. (2016). *playground.arduino: Capacitive Sensing Library*. Retrieved Mai 15, 2016, from <http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense>
- Deutsches Taubblindenwerk GmbH. (2015). *Deutsches Taubblindenwerk GmbH: Lormen*. Retrieved Mai 17, 2016, from <http://www.taubblindenwerk.de/lormen.html>
- Elektronik Kompendium. (2016). *Elektronik Kompendium*. Retrieved Mai 5, 2016, from Darlington-Schaltung / Darlington-Transistor: <http://www.elektronik-kompendium.de/sites/slt/0411221.htm>
- itead. (2015). *itead: nRF24L01 Module*. Retrieved Dezember 17, 2015, from <https://www.itead.cc/nrf24l01-module.html>
- mikrocontroller.net. (2016). *mikrocontroller.net: UART*. Retrieved Mai 14, 2016, from <http://www.mikrocontroller.net/articles/UART>
- playground.arduino. (2016). *playground.arduino: Native Capacitive Sensors without additional Hardware*. Retrieved Mai 9, 2016, from <http://playground.arduino.cc/Code/CapacitiveSensor>
- Reutlingen-university. (2016). *Reutlingen-university: HC Serial Bluetooth Products* . Retrieved Mai 14, 2016, from http://www.tec.reutlingen-university.de/uploads/media/DatenblattHC-05_BT-Modul.pdf
- Schnabel, P. (2007). *Elektronik-Fibel* (Auflage: 4. voll. überarbeitete Auflage (Juni 2007) ed.). Ludwigsburg: Books on Demand.
- Taubblindendienst. (2016). *Taubblindendienst: Verständigung durch Lormen - Das Handalphabet nach Hieronymus Lorm*. Retrieved Mai 24, 2016, from <http://www.taubblindendienst.de/index.php?menuid=50>
- taubblindenwerk.de A. (2016). *taubblindenwerk.de: Charge*. Retrieved Mai 19, 2016, from <http://www.taubblindenwerk.de/charge-syndrom.html>
- taubblindenwerk.de B. (2016). *taubblindenwerk.de: taubblindheit*. Retrieved Mai 19, 2016, from <http://www.taubblindenwerk.de/was-ist-taubblindheit.html>
- taubblindenwerk.de C. (2016). *taubblindenwerk.de USHER*. Retrieved Mai 19, 2016, from <http://www.taubblindenwerk.de/usher-syndrom.html>
- Tuelectronica. (2012). *Tuelectronica*. Retrieved Mai 5, 2016, from <http://www.tuelectronica.es/tutoriales/electronica/resistencia-pull-up-y-pull-down.html>
- WeTec's Technikseite. (2010). *WeTec's Technikseite: Kondensator*. Retrieved Mai 28, 2016, from <http://home.arcor.de/wetec/rechner/cclad.htm>
- Wikipedia. (2016a, März 30). *Wikipedia: Darlington-Schaltung*. Retrieved Mai 8, 2016, from <https://de.wikipedia.org/wiki/Darlington-Schaltung>
- Wikipedia. (2015). *Wikipedia: Kommunikation*. Retrieved September 20, 2015, from <https://de.wikipedia.org/wiki/Kommunikation>
- Wikipedia. (2016b). *Wikipedia: Serial Peripheral Interface*. Retrieved Mai 14, 2016, from https://de.wikipedia.org/wiki/Serial_Peripheral_Interface#

16 ANHANG

16.1 HARDWARESPEZIFIKATIONEN

Die Hardwarespezifikationen sind unter [7.6 Technische Daten der Komponenten](#), sowie auch in digitaler Form

16.2 DATENBLÄTTER

Finden sich in Digitaler Form. Die Namen der Datenblätter sind Ident mit der Position in der Stückliste.

16.3 STÜCKLISTEN

Anbei sind die Stücklisten unseres Projektes aufgelistet.

16.3.1 STÜCKLISTE LORMEN-HAND

Pos Nr.	Anzahl	Bezeichnung	Daten	Definition	Gebinde	Hersteller/Lieferant	Bestell Nr.	Preis	Preis gesamt
10	1	Epoxydharz & Härter	TOOLCRAFT Epoxydharz 1000 g. TOOLCRAFT Härter L 40 min 400 g.		1	Conrad	1230973	31,99 €	31,99 €
20	1	Grafitspray	Graphit 33 - 200ml Dose		1	Conrad	813826	14,99 €	14,99 €
25	1	Lackspray Grey	ACRYL-Schutzlack RAL 7035 Licht-Grau 400ml		1	Conrad	892993	8,99 €	8,99 €
30	1	Klarlackspray	Klarlack-Spray Nigrin 400ml		1	Conrad	893411	5,99 €	5,99 €
35	1	Aderleitung 2,5mm ²	Anschlussleitung H05RR-F 3 G 2.5 mm ² Schwarz LappKabel 1600209 Meterware		1	Conrad	600694	3,79 €	3,79 €
40	1	Flachbandkabel	26 x 0.08 mm ²		1	Conrad	744317	3,79 €	3,79 €
45	1	Knetmasse	QuikSteel 16402EU		1	Conrad	479065	8,99 €	8,99 €
50	2	Diode	1n4007	D1- 11	1	Conrad	162272	0,08 €	0,16 €
55	1	Gelenk	Winkelankerplatte Hekatron AFS 55 für Türhaftmagneten (Magnet)		1	HaBeFa	6500102	33,90 €	33,90 €
60	1	6-Kantschraube	Sechskantschrauben M8 25 mm Außensechskant DIN 933 Stahl verzinkt 100 St. SWG		100	Conrad	1202623	15,99 €	0,16 €
65	1	Microcontroller	Arduino Mega 2560		1	Conrad	191790	49,99 €	49,99 €
70	3	Stiftleiste Lang	36 Pin		1	Conrad	741133	1,09 €	3,27 €
75	1	Stiftleiste Kurz	40 Pin		1	Conrad	741146	0,49 €	0,49 €

80	12	Schraube Kreuz M4x15mm	Gewindeschrauben- Sortiment M4 25 mm Kreuzschlitz Philips DIN 7985 Stahl verzinkt 100 St. SWG		100	Conrad	1202535	5,49 €	0,66 €
85	4	Beilagscheibe für M4	Kotflügelscheibe Innen-Durchmesser: 4.3 mm M4 Stahl verzinkt 100 St. SWG		100	Conrad	1204381	4,49 €	0,18 €
90	1	RF Transreceiver	4.3 mm M4 Stahl verzinkt 100 St. SWG	J1	1	Amazon	B0108BT5Y2	4,99 €	4,99 €
95	1	Jumperkabelset	Bunt		120	Amazon	B013B0CF30	7,99 €	0,07 €
100	4	Gummifüße	Gummifuss 38 x 20 8er Set		8	Conrad	304443	7,99 €	4,00 €
105	4	Haltemagnet	Permanent-Magnet Rechteckig N35 Grenztemperatur (max.): 80 °C W20L12T3-M4		2	Conrad	1218984	3,59 €	7,18 €
110	1	Superkleber	UHU Alleskleber Super Strong & Safe Minix 3x1 g 44305		1	Conrad	478770	7,49 €	7,49 €
115	1	Heißkleber	Heißklebesticks Ø 7 mm 100 mm Transparent 12 St. 59 g		12	Conrad	438014	2,99 €	0,25 €
120	1	Leiterplatte	200mm x 150mm		1	Conrad	523850	5,29 €	5,29 €
125	1	Lötzinn	Sn60Pb39Cu1		1	Conrad	812838	2,49 €	2,49 €
130	4	Schraube Kreuz M3x30mm	Linsenschrauben M3 3 mm Kreuzschlitz Philips DIN 7985 Stahl galvanisch verzinkt 200 St. TOOLCRAFT 145898		200	Conrad	145898	21,99 €	0,44 €
135	25	Widerstand 1M Ohm	Kohleschicht- Widerstand 1 MΩ axial bedrahtet 0207 0.25 W Yageo CFR-25JT-52-1M0 1 St.		1	Conrad	1417664	0,08 €	2,00 €
140	6	Plexiglas 8mm	Acrylglascheibe (L x B) 100 mm x 200 mm Materialstärke 3 mm Transparent 1 St.		1	Conrad	530816	3,79 €	22,74 €
145	2	Widerstand 1k Ohm	Kohleschicht- Widerstand 1 kΩ axial bedrahtet 0207 0.25 W Yageo		1	Conrad	1417699	0,08 €	0,16 €

			CFR-25JT-52-1K0 1 St.					
150	2	Widerstand 33 Ohm	Kohleschicht-Widerstand 33 Ω axial bedrahtet 0207 0.25 W Yageo CFR-25JT-52-33R 1 St.	1	Conrad	1417709	0,08 €	0,16 €
155	2	Kondensator 100nF	Keramik-Kondensator radial bedrahtet 100 nF 50 V/DC 10 % Holystone RDCX104K050DKA 1 St.	1	Conrad	531855	0,11 €	0,22 €
160	2	Vibrator	Vibrating Mini Motor Disc	1	Adafruit	1201	1,76 €	3,52 €
165	2	NPN Transistor 2N2222	Transistor Central Semiconductor corp. 2N2222A NPN Gehäuseart TO-18 I(C) 800 mA Emitter-Sperrspannung U(CEO) 40 V	1	Conrad	1359807	2,69 €	5,38 €
170	1	6-Kant Mutter M8	Sechskantmuttern M8 DIN 934 Stahl verzinkt 1 St.	1	Conrad	888723	0,12 €	0,12 €
175	1	Laborbuchse	Laborbuchse Buchse, Einbau vertikal Stift- \varnothing : 4 mm Grau SKS Hirschmann BIL 20 1 St.	1	Conrad	734571	1,09 €	1,09 €
180	1	Laborstrippe 2m	Messleitung [Lamellenstecker 4 mm - Lamellenstecker 4 mm] 2 m Schwarz SKS Hirschmann CO MLN 200/1	1	Conrad	122606	8,19 €	8,19 €
185	1	Erdungsarmband	ESD-Handgelenkband Schwarz Bernstein 9-342-1 Druckknopf 3 mm	1	Conrad	1425583	11,99 €	11,99 €
190	1	Netzteil	USPS-1500	1	Conrad	518372	23,99 €	23,99 €

195	1	Gummimatte 3mm	Gummimatte Hazet 160-012 (L x B) 45.5 cm x 28.5 cm	1	Conrad	1287833	14,99 €	14,99 €
200	12	Schraube Kreuz M3x20mm	Senkschrauben M3 20 mm Torx DIN 965 Edelstahl A2 1 St.	1	Conrad	889789	0,18 €	2,16 €
205	2	Winkel	Stuhlwinkel (L x B) 25 mm x 25 mm	1	Conrad	889631	0,26 €	0,52 €
210	1	Stiftleiste WS26G	Stiftleiste WS26G Gesamtpolzahl 26 Anzahl Reihen 2 econ connect 1 St.	1	Conrad	1303915	0,59 €	0,59 €
215	1	Pfostenverbinder	Buchsenleiste Rastermaß: 2.54 mm Polzahl Gesamt: 26 ASSMANN WSW 1 St.	1	Conrad	745376	0,96 €	0,96 €
220	8	Beilagscheibe für M3	Unterlegscheiben Innen-Durchmesser: 3.2 mm M3 DIN 9021 Stahl verzinkt 100 St. TOOLCRAFT 3,2 D9021:A2K 194723	100	Conrad	521800	1,99 €	0,16 €
225	1	LED Leiste Blau	LED-Lichtleiste mit Stecker 12 V 28.5 cm Blau Rolux	1	Conrad	289532	17,99 €	17,99 €
230	12	Mutter für M3	Sechskantmuttern M3 DIN 439 Stahl verzinkt 100 St. TOOLCRAFT 827814	100	Conrad	827814	5,49 €	0,66 €

Summe: 317,12 €

16.3.2 STÜCKLISTE FEEDBACK-BOX

Pos Nr.	Anzahl	Bezeichnung	Daten	Definition PCP	Gebinde Anzahl	Hersteller/ Lieferant	Bestell Nr.	Preis	Preis gesamt
10	1	Gehäuse	250x180x75 Grau		1	Conrad	534443	19,99 €	19,99 €
15	1	Einstellbarer Spannungsregler	3A, 1,2 - 33V	IC1	1	Conrad	176052	0,54 €	0,54 €
20	1	Potenziometer	2,5kΩ, 0,25W	R2	1	Conrad	431346	0,48 €	0,48 €
25	1	Kondensator	1μF	C2	1	Conrad	38108	0,50 €	0,50 €
30	1	Kondensator	0,1μF	C1	1	Conrad	458211	0,12 €	0,12 €
40	1	Widerstand	1kΩ, 0,25W	R3	1	Conrad	1417707	0,08 €	0,08 €
45	1	Widerstand	2kΩ, 0,6W	R5	1	Conrad	1417622	0,12 €	0,12 €
50	11	Diode	1n4007	D1- 11	1	Conrad	162272	0,08 €	0,88 €
55	3	Demultiplexer	CD74H4514E	V1- V3	1	Distrelec	666831	2,66 €	7,98 €
60	5	Darlington Transistor Stufen	ULN2803A	TRA 1- TRA 5	1	Conrad	171824	0,44 €	2,20 €
65	1	Microcontroller	Arduino Mega 2560		1	Conrad	191790	49,99 €	49,99 €
70	3	Stiftleiste Lang	36 Pin		1	Conrad	741133	1,09 €	3,27 €
75	2	Stiftleiste Kurz	40 Pin		1	Conrad	741146	0,49 €	0,98 €
80	1	Steckverbinder Buchse	5,5mm x 2,5mm		1	Conrad	072336	1,39 €	1,39 €
85	1	Bluetooth Modul	HC-06	J2	1	Amazon/ Hi-elec	B00IY7K41 G	7,75 €	7,75 €
90	1	RF Transreceiver	nRF24L01	J1	1	Amazon	B0108BT5 Y2	4,99 €	4,99 €
95	1	Jumperkabelset	Bunt		12 0	Amazon	B013B0CF 30	7,99 €	7,99 €
100	1	Display	2.8" TFT Touch		1	Amazon	B00YXOCT SU	12,60 €	12,60 €
105	1	Flachbandkabel	26 x 0.08 mm ²		1	Conrad	744317	3,79 €	3,79 €
110	2	Flachbandkabel Stecker	Stiftleiste WS26SK		1	Conrad	1303905	3,29 €	6,58 €
115	1	Knickschutz	Spiralschlauch Weiß		5	Conrad	1091096	14,99 €	14,99 €
120	1	Leiterplatte	200mm x 150mm		1	Conrad	523850	5,29 €	5,29 €
125	1	Lötzinn	Sn60Pb39Cu1		1	Conrad	812838	2,49 €	2,49 €
130	1	Knetmasse	QuikSteel 16402EU		1	Conrad	479065	8,99 €	8,99 €
135	1	Netzteil	USPS-1500		1	Conrad	518372	23,99 €	23,99 €

Summe: 187,97 €

16.3.3 STÜCKLISTE HAPTISCHE AUSGABE

Pos Nr.	Anzahl	Bezeichnung	Daten	Definition PCP	Gebinde anzahl	Hersteller/ Lieferant	Bestell Nr.	Preis	Preis gesamt
10	37	Vibrator	Vibrating Mini Motor Disc		1	Adafruit	1201	1,76 €	65,12 €
20	1	Flachbandkabel	3m		1	Conrad	744317	3,79	3,79 €
30	1	Heißkleber	Heißklebesticks Ø 7 mm 100 mm Transparent 12 St. 59 g		12	Conrad	438014	2,99 €	0,25 €
40	1	Zwirn	er Set Sternzwirn weiß, beige, braun, blau Sternchenzwirn, Nähgarn, Zwirn, auf SB Karte, 2070		6	Amazon	2070	2,46 €	0,41 €
50	1	Zwirn	er Set Sternzwirn weiß, beige, braun, blau Sternchenzwirn, Nähgarn, Zwirn, auf SB Karte, 2070		6	Amazon	2070	2,46 €	0,41 €
60	3	Schrumpfschlauch h 1m	Schrumpfschlauch, dünnwandig Ø vor/nach Schrumpfung: 1.5 mm/0.6 mm Schrumpfrate 2 : 1 Schwarz		1	Conrad	1225410	0,17 €	0,51 €
70	1	Handschuh	worky 1151 Feinstrickhandschuh MICRO black 100% Nylon mit PU-Beschichtung Größe 10		2	Conrad	1285314	1,99 €	1,00 €
80	1	Gummimatte 3mm	Gummimatte Hazet 160-012 (L x B) 45.5 cm x 28.5 cm		1	Conrad	1287833	14,99 €	14,99 €
90	3	Gummiband	Gummiband, 3 m, 5 mm breit, schwarz		1	Stoffe Hemmers	25404	2,70 €	8,10 €

100	2	Pfostenverbinder	Buchsenleiste Rastermaß: 2.54 mm Polzahl Gesamt: 26 ASSMANN WSW 1 St.		1	Conrad	745376	0,96 €	1,92 €
								Summe:	<u>98,98 €</u>