



FAKULTÄT ELEKTROTECHNIK UND WIRTSCHAFTSINGENIEURWESEN

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN LANDSHUT

Projektbericht im Modul Eingebettete autonome Systeme I zum Thema

Entwicklung einer Raspberry Pi basierten Handheld-Konsole mit eingebettetem GameBoy Cartridge Reader II

Autor:	Felix Weichselgartner, Matthias Lohberger
Betreuer:	Prof. Dr.-Ing. Mathias Rausch
Eingereicht:	09.07.2021
Aktualisierte Version:	05.10.2021

Hiermit erklären wir, dass wir die Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet haben.

Tiefenbach, 09.07.2021

Felix Wechselgartner, Matthias Lohberger

Acknowledgments

We want to thank the Open Source and GameBoy community,
who made this project possible.

Abstract

A popular project in the internet community is the „GameBoy Zero“. Here a Raspberry Pi (RPi) Zero [1] (with buttons, battery, power electronics, display, etc.) is integrated in the shell of an original GameBoy. This result in a RPi based handheld game console, that can emulate games from various systems. More about the history of the project can be found here [2]. The new contribution to the project is the embedded cartridge reader, which makes it possible to interact with real GameBoy cartridges. Therefore, it is not necessary to download ROMs from dubious websites and you are able to use your real GameBoy cartridges and emulate them, all in one device.

In this work, the design of a replacement GameBoy mainboard is continued. The pcb will be extended with components essential for an autonomous embedded system (e.g. LiPo-Battery, LiPo-Charger, Boost-Converter, ...). Furthermore, a replacement display pcb will be developed, which acts as a carrier board for a RPi Compute Module 4 (CM4). The CM4 used instead of the RPi Zero to get more calculating power, which enables the user to play more demanding games.

Inhaltsverzeichnis

Acknowledgments	ii
Abstract	iii
1 Motivation	1
1.1 Problemstellung	1
1.2 Zielsetzung	1
2 Grundlagen	2
2.1 GameBoy	2
2.2 Raspberry Pi Compute Module 4	2
2.3 Ausschaltverzögerung	4
2.4 Pegelwandlerschaltung	6
2.5 Analoge und digitale Masse	8
2.6 Resistiver Touchscreen	8
2.7 Inter-IC Sound (I ² S)	9
3 Implementierung	10
3.1 Hardware	10
3.1.1 Hardware-Aufbau	10
3.1.2 Cartridge-PCB	11
3.1.3 Display-PCB	16
3.2 Software	28
3.2.1 Cartridge-Reader	28
3.2.2 Display-PCB	30
4 Testen	32
5 Ergebnisse	34
Literaturverzeichnis	36
Abbildungsverzeichnis	39

Tabellenverzeichnis

41

1 Motivation

1.1 Problemstellung

Ende 2018 wurde ich auf ein populäres Elektronikprojekt namens „GameBoy Zero“ [2] aufmerksam. Im Laufe der nächsten Monate, baute ich meine eigene Version [3], womit ich schlussendlich verschiedene Systeme auf einem RPi Zero [1] emulieren konnte. Der Haken ist jedoch, sofern man nicht seine eigenen GameBoy-Spiele auslesen kann, ist man auf sogenannte ROMs aus dem Internet angewiesen. Diese beinhalten den ausgelesenen Read Only Memory (ROM)-Speicher der GameBoy-Cartridges, womit das Spiel emuliert werden kann [4]. Dies führte dazu, dass ich Mitte 2019 einen Catridge-Reader [5] baute, welcher es mir ermöglichte, meine eigenen Catridges auszulesen. Hierbei kann aber nicht nur der ROM ausgelesen werden, sondern auch der Random Access Memory (RAM), welcher den Speicherstand der Spiele enthält. Das Problem hierbei ist jedoch, dass dies zwei separate Geräte und nicht wie beim originalen GameBoy in einem Gerät vereint sind.

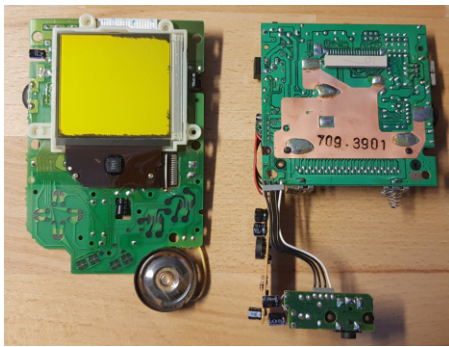
1.2 Zielsetzung

In dieser Arbeit wird das Design einer Ersatzplatine des Mainboards für den Klassischen GameBoy (DMG) fortgeführt. Die Platine wird um Komponenten erweitert, welche essenziell für ein eingebettetes autonomes System sind (z.B. LiPo-Batterie, LiPo-Ladeschaltung, Aufwärtswandler, ...). Des Weiteren wird eine Ersatzplatine für die Display-Platine des DMG entwickelt, welche u.a. als Trägerplatine für ein RPi CM4 dient. Das CM4 wird anstatt eines RPi Zero verwendet, um mehr Rechenleistung für aufwändigere Spiele zur Verfügung zu haben.

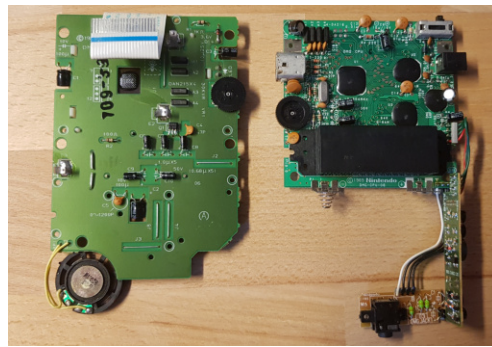
2 Grundlagen

2.1 GameBoy

Der DMG hatte zwei Platinen. Die obere Platine (jeweils links in Abbildung 2.1) enthält im Wesentlichen das Display, Audio und die Buttons. Daher wird die Ersatzplatine im Folgenden Display-PCB genannt. Die untere Platine (jeweils rechts in Abbildung 2.1) enthält die CPU und die Cartridge-Schnittstelle. Der Ersatz für diese Platine wird daher im Folgenden Cartridge-PCB genannt.



(a) Oberseiten der Platinen



(b) Unterseiten der Platinen

Abbildung 2.1: DMG Platinen

2.2 Raspberry Pi Compute Module 4

Das RPi CM4 [6] ist ein sog. „System on Module (SoM)“ welches die Kernkomponenten eines RPi auf einer kleinen aufsteckbaren Platine enthält. Sämtliche Anschlüsse für Peripheriegeräte werden auf dem sog. „Carrier-Board“ verbaut und über die Steckverbindungen leitend mit dem CM4 verbunden. Als Entwicklungsplatine wird von der Raspberry Pi Foundation das sog. „Raspberry Pi Compute Module 4 IO Board“ [7] vertrieben, welches die Spannungsversorgung, Anschlüsse für Peripheriegeräte (z. B. HDMI, PCIe, USB, GPIO-Pins, MicroSD-Karte, ...) und weitere Funktionen enthält. Auf dieses kann das CM4 aufgesteckt werden und danach ähnlich zu einem RPi 4

verwendet werden. Hierbei ist zu beachten, dass das CM4 nur lose auf dem RPi 4 basiert. Dementsprechend ist die Rechenleistung und Leistungsaufnahme vergleichbar, jedoch nicht identisch.

<https://www.jeffgeerling.com/sites/default/files/images/raspberry-pi-model-size-comparison.jpg>

Abbildung 2.2: Vergleich verschiedener RPi Modelle [8]

In Abbildung 2.2 werden verschiedene RPi Modelle gezeigt. Neben dem bisher für den Einsatz geplanten RPi Zero W, wird das CM4 gezeigt, sowie das Vorgängermodell Compute Module 3 (CM3). Des Weiteren wird noch ein RPi 4 (Model B) gezeigt, wobei auffällt, dass dieses signifikant größer ist als die anderen Modelle (sowohl in Höhe als auch in Länge und Breite). Der wesentliche Unterschied zwischen dem Compute Module und dem RPi Zero W ist die Rechenleistung und die Leistungsaufnahme. Besonders das CM4 hat wesentlich mehr Rechenleistung als der RPi Zero W, weshalb die Leistungsaufnahme beim CM4 entsprechend höher ist. Beide Modelle wären somit für den Einsatz geeignet. Hier wurde die Entscheidung getroffen ein CM4 aufgrund der hohen Rechenleistung zu verwenden. Eine detaillierte Analyse des CM4 kann in [8] gefunden werden.

Ein Vergleich der Rechenleistung verschiedener RPi Modelle kann in Abbildung 2.3 und ein Vergleich der Leistungsaufnahme in Abbildung 2.4 gefunden werden. In [9] werden mehrere RPi Modelle anhand verschiedener Parameter verglichen.

https://rpi-magazines.s3-eu-west-1.amazonaws.com/magpi/legacy-assets/2019/06/2019-06-21-15_40_11-Window.png

Abbildung 2.3: Vergleich der Rechenleistung verschiedener RPi Modelle [9]

https://rpi-magazines.s3-eu-west-1.amazonaws.com/magpi/legacy-assets/2019/06/2019-06-21-15_44_14-Window.png

Abbildung 2.4: Vergleich der Leistungsaufnahme verschiedener RPi Modelle [9]

Es fällt in Abbildung 2.4 auf, dass der RPi 4 ca. das Drei- bis Vierfache der Leistung des RPi Zero benötigt. Dementsprechend ist bei dem CM4 wesentlich geringere Akkulaufzeit zu erwarten, da der RPi den Löwenanteil der Leistungsaufnahme ausmacht. Aus Abbildung 2.3 erkennt man, dass der RPi 4 ca. die 15-fache Benchmark-Bewertung im Vergleich zum RPi Zero erhält. Mit dieser zusätzlichen Rechenleistung sollten auch rechenaufwendigere Systeme auf dem CM4 emuliert werden können.

In diesem Projekt wird das CM4 Lite verwendet. Der Unterschied ist, dass dieses keinen embedded MultiMedia Card (eMMC)-Speicher auf der Platine verbaut hat, sondern eine externe MicroSD-Karte als Speicher für das Dateisystem verwendet. Dies gibt dem Nutzer den Vorteil den Speicher auszutauschen und ohne zusätzliche Hardware mit einem anderen Gerät auf das Dateisystem zuzugreifen.

Ein weiterer Unterschied zur Vorgängerversion ist, dass, wie in Abbildung 2.5 zu sehen ist, das CM4 jetzt durch zwei High-Density Konnektoren auf der Unterseite auf zwei entsprechende Gegenstücke aufgesteckt werden kann. Jeder der Konnektoren verfügt über 100 Pins.

https://hackster.imgix.net/uploads/attachments/1204828/image_HXdePwAR7G.png

Abbildung 2.5: Rückseite des CM4 [10]

2.3 Ausschaltverzögerung

Damit der Nutzer das Gerät leicht ein- bzw. ausschalten kann befindet sich auf der Oberseite ein Schalter. Um die Elektronik zu schützen (vor allem den RPi) wird mit dem Schalter nicht direkt die Stromversorgung getrennt. Stattdessen trennt der Schalter einen Kondensator mit parallel geschaltetem Widerstand von der Batteriespannung. Das RC-Glied entlädt sich infolgedessen und unterschreitet die Schaltschwelle eines Inverters. Der Ausgang dieses Inverters ist nun eine logische 1 und wird auf einen weiteren Inverter geschaltet. Der zweite Inverter hat deshalb einen Ausgang von einer logischen 0. Dieser Ausgang wird nun auf den Enable-Pin der Lade- und Aufwärtswandlerschaltung (Adafruit PowerBoost 1000C [11]) gelegt. Um das Ausschalten des Nutzers zu erkennen, wird durch das Umlegen des Schalters ein großer Widerstand mit der Batteriespannung und einem GPIO-Pin des RPi verbunden. Der RPi kann durch das Überprüfen dieses Pins das Ausschalten erkennen und sich sicher herunterfahren. Dies ist besonders für das Dateisystem des RPi's wichtig. Die Länge der Ausschaltverzögerung kann über die Werte des RC-Glieds angepasst werden. Der Schaltplan für die Ausschaltverzögerung kann in Abbildung 2.6 gesehen werden.

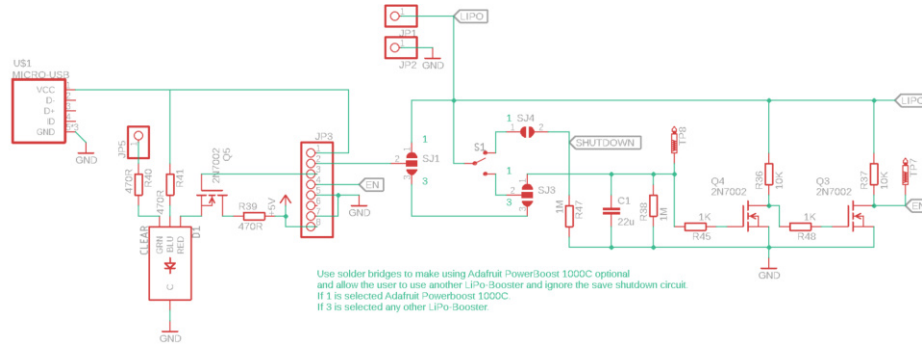


Abbildung 2.6: Schaltplan Ausschaltverzögerungsschaltung über RC-Glied

Mit der Schwellenspannung des Transistors und den einer RC-Konstanten von

$$\tau = R * C = 1 \text{ M}\Omega * 22 \mu\text{F} = 22 \text{ s} \quad (2.1)$$

ergibt sich in einer LTSpice-Simulation eine Ausschaltverzögerung von ungefähr 22s (vgl. Abbildung 2.7). Hier wird in Rot die Spannung an dem RC-Glied dargestellt, in Grün das Ausschalten des Schalters und in Blau die zweite Inverterstufe.

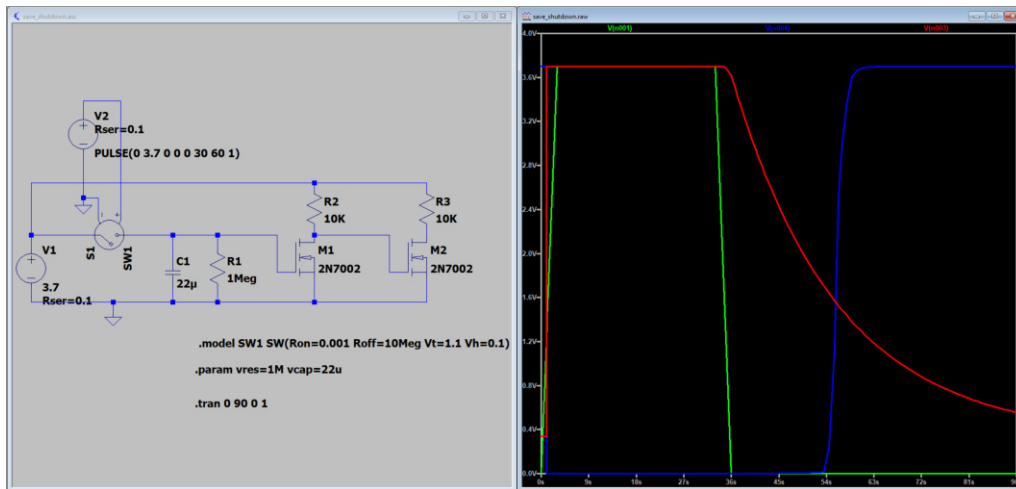


Abbildung 2.7: Simulation der Ausschaltverzögerungsschaltung über RC-Glied

An einer realen Schaltung konnte mit den oben genannten Werten eine Verzögerung von ca. 12s gemessen werden (vgl. Abbildung 2.8). Hier wird in Blau das Umlegen des Schalters dargestellt und in Gelb die Spannung des RC-Glieds. Mehr über das Testen der Ausschaltverzögerung kann in [12] nachgelesen werden.

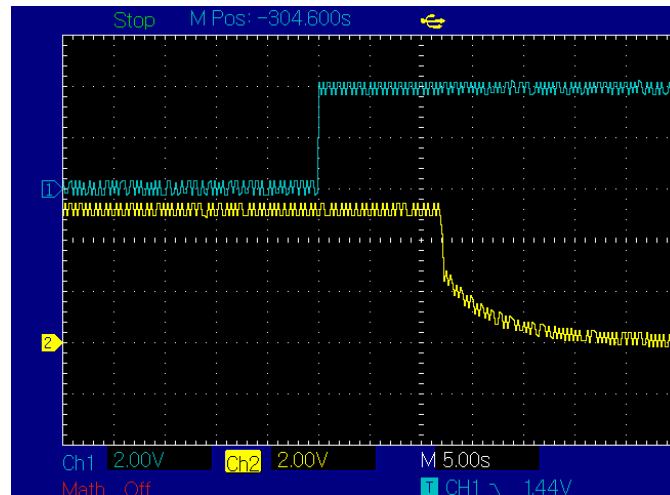
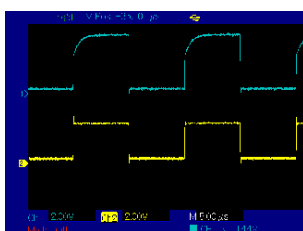


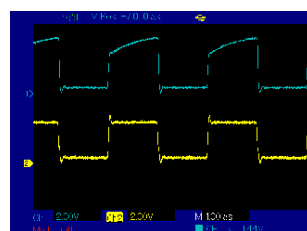
Abbildung 2.8

2.4 Pegelwandlerschaltung

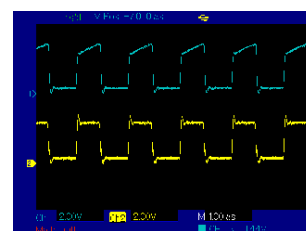
Im vergangenen Semester wurde eine bidirektionale Pegelwandlerschaltung (Pws) aus MOSFETs genutzt um die 3.3V Spannungspegel der Inter-Integrated Circuit (I²C)-Leitungen des RPi's auf 5V zu wandeln. Hiermit war eine Kommunikation mit den MCP23017-Modulen mit 100kHz ohne Probleme möglich. Die maximale I²C-Geschwindigkeit, die von den MCP23017-Modulen unterstützt wird, liegt bei 1.7 MHz. Nach dem Vermessen der Pws (siehe Abbildung 2.9, gelb Eingangsspannung, blau gewandelte Ausgangsspannung) mit unterschiedlichen Frequenzen und einer Simulation (siehe Abbildung 2.10, grün Eingangsspannung, rot gewandelte Ausgangsspannung) in LTSpice wurde klar, dass die Schaltung zu langsam für Geschwindigkeiten von 200 kHz und höher ist. Mehr über das Testen der MOSFET-Pws kann in [13] nachgelesen werden.



(a) 40 kHz



(b) 200 kHz



(c) 400 kHz

Abbildung 2.9: Geschwindigkeitsmessung der MOSFET-Pws

2.5 Analoge und digitale Masse

Auf einer Platine mit analogen und digitalen Signalen sollten die jeweiligen Schaltkreise stets voneinander getrennt werden. Dem liegt zugrunde, dass digitale Chips rauschintensiv sein können und so bei kombinierten Schaltungen durch elektromagnetische Störungen die analogen Signale beeinträchtigen. Durch die Trennung der analogen und digitalen Masse entsteht allerdings das Problem der potentialfreien Masse. Denn um ordnungsgemäß zu funktionieren, sollten sich alle analogen und digitalen Chips auf dasselbe Massepotential beziehen. Die Lösung liegt hier in der Verbindung der beiden Masseflächen. Dabei gilt es wiederum Masseschleifen zu vermeiden. Eine Möglichkeit zur Reduzierung der Wahrscheinlichkeit von Masseschleifen ist die sog. Sternpunkt-Masseverbindung. Dabei werden die unterschiedlichen Masseflächen in einem einzigen, zentralen Punkt verbunden. [15]

Auf dem Display-PCB werden insbesondere für das Audio mehrere analoge Komponenten verwendet. Hinzu kommt u.a. der PCM5100, ein Digital-Analog-Umsetzer (DAC) von Texas Instruments zur Wandlung eines digitalen I²S-Signals in analoges Stereo Audio. Dieser benötigt dafür sowohl eine analoge als auch eine digitale Masse und Spannungsversorgung. Die genaue Pinbelegung kann in Abbildung 2.12 nachgelesen werden. Während DVDD und DGND digital verbunden werden müssen, ist für AVDD, AGND, CPVDD und CPGND eine analoge Spannungsversorgung bzw. Masse vorgesehen. Weitere Informationen über den PCM5100 können unter [16] nachgelesen werden.

<https://www.ti.com/lit/gpn/pcm5100>

Abbildung 2.12: Pinbelegung des PCM5100 Audio Stereo DAC [16]

2.6 Resistiver Touchscreen

Resistive Touchscreens sind eine kostengünstige Alternative zu kapazitiven Touchscreens. Letztere werden häufig in Smartphones eingesetzt und besitzen den Vorteil, dass sie mehrere Finger gleichzeitig auf dem Touchscreen erkennen können. Resistive Touchscreens hingegen sind nur in der Lage, eine einzige Berührung zu registrieren, erlauben dafür aber außer den Fingern auch die Verwendung eines Stylus o.Ä. zur Eingabe. Die Erkennung eines Single Touch funktioniert dabei wie folgt:

Abbildung 2.13 zeigt den grundsätzlichen Aufbau eines LCD-Screens mit resistivem Touch. Der Touchscreen besteht aus zwei parallelen, leitenden Flächen, die mit einem kleinen Abstand zueinander befestigt sind. Die Elektroden der oberen Schicht (Y) sind dabei um 90° gegenüber denen der unteren Schicht (X) gedreht.

https://www.analog.com/-/media/images/analog-dialogue/en/volume-45/number-2/articles/gesture-recognition-on-resistive-touch-screens/gesture-recognition-on-resistive-touch-screens_fig_01.jpg?la=en&imgver=1

Abbildung 2.13: Aufbau (a) und Funktionsweise (b) eines resistiven Touchscreens [17]

Ein „Touch“ tritt auf, wenn sich die beiden Flächen durch Druck auf eine kleine Fläche des Touchscreens berühren. Wenn eine DC Spannung zwischen den beiden Elektroden der oberen Schicht angelegt wird und die untere Ebene floatet, bringt der „Touch“ die untere Ebene auf das Potential des Berührungspunkts. Die Touch-Koordinate in Richtung der oberen Schicht wird dann durch Messen der Spannung auf der unteren Fläche und Ermitteln des Verhältnisses des Widerstands im Touch-Punkt zum Gesamtwiderstand bestimmt. Anschließend werden die elektrischen Verbindungen getauscht und die zweite Koordinate des Touch-Punkts auf der anderen Achse kann bestimmt werden. Die mit DC Spannung versorgte Schicht wird dabei „active-layer“ genannt, die Schicht deren Spannung gemessen wird „passive-layer“. Wenn ein Single Touch erfolgt, entsteht ein Spannungsteiler im active-layer. Durch die Spannungsmessung im passive-layer wiederum kann ein Analog-Digital-Umsetzer (ADC) die Spannung proportional zum Abstand des Touch-Punkts von der negativen Elektrode einlesen. [17]

2.7 I²S

In den 1980er Jahren wurde I²S von Philips Semiconductors - mittlerweile NXP - entwickelt. Das serielle Protokoll sollte die Übertragung von Stereo-Audiodaten zwischen DACs, digitalen Filtern oder anderen ICs standardisieren. Die Übertragung besteht dabei aus drei Signalen, die in Abbildung 2.14 zu sehen sind. Mit dem Serial Data (SD) Signal werden die Audiodaten mit Most Significant Bit (MSB) first in Words gesendet. Das Word select (WS) Signal gibt dazu an, ob das jeweilige Word für den rechten (WS high) oder den linken (WS low) Audio Channel bestimmt ist. Das Serial Clock (SCK) Signal gibt für den gesamten Sendevorgang den Takt vor. Insgesamt ist I²S zwar begrenzt auf Audiodaten, bietet dafür aber für diese Art der Datenübertragung eine höhere Datenrate als beispielsweise Universal Asynchronous Receiver Transmitter (UART) oder I²C. [18]

https://de.wikipedia.org/wiki/I%C2%B2S#/media/Datei:I2S_Timing.svg

Abbildung 2.14: Übertragung über I²S [19]

3 Implementierung

3.1 Hardware

3.1.1 Hardware-Aufbau

Wie auch der DMG besteht die Schaltung hier aus zwei Platinen: der Cartridge-PCB und der Display-PCB. Die Funktionen der Schaltung wurden auf die zwei Platinen wie folgt aufgeteilt:

Cartridge-PCB siehe Kapitel 3.1.2	Display-PCB siehe Kapitel 3.1.3
Cartridge-Interface	CM4
Pws	Display
Lade- und Aufwärtswandlerschaltung	Audio (Verstärkung, Lautsprecher, Kopfhörerbuchse, Lautstärkerad)
Ausschaltverzögerung	
Batterieüberwachung	
MicroSD-Karte	

Tabelle 3.1: Aufteilung der Funktionen auf Platinen

Um Signale zwischen den Platinen auszutauschen, wurde ein flexible Flachbandkabel (ffc) mit dem dazu passenden flexible Leiterplatten (fpc)-Anschluss verwendet. Die 18-Pins wurden dabei wie folgt belegt:

Pin	Bezeichnung	Funktion
1	5V	Spannungsversorgung der Cartridge-PCB
2	5V	Spannungsversorgung der Cartridge-PCB
3	5V	Spannungsversorgung der Cartridge-PCB
4	5V	Spannungsversorgung der Cartridge-PCB
5	SDA	I ² C-Daten vom CM4
6	SCL	I ² C-Takt vom CM4
7	3.3V	Digitale 3.3V Spannungsversorgung vom CM4
8	CD/DAT3/CS	SD-Karte
9	DAT2/RSV	SD-Karte
10	DAT1/RSV	SD-Karte
11	DAT0/DO	SD-Karte
12	CLK/SCLK	SD-Karte
13	CMD/DI	SD-Karte
14	Shutdown	Ausschaltsignal für CM4
15	GND	Ground der Cartridge-PCB
16	GND	Ground der Cartridge-PCB
17	GND	Ground der Cartridge-PCB
18	GND	Ground der Cartridge-PCB

Tabelle 3.2: Verbindungspins der Platinen

Es ist schwer genaue Angaben zu dem maximalen Strom, der über eine Leitung des ffc übertragen werden darf, zu finden. Als Richtwert wurde hier 500mA gefunden. Wie aus Abbildung 2.4 entnommen werden kann, sollte eine Leistung von bis zu 8W abgedeckt werden. Bei einer Spannung von 5V sind vier Leitungen mit jeweils 500mA ausreichend für eine Leistung von 10W. Aus diesem Grund wurden jeweils vier Leitungen für 5V und GND beim ffc verwendet.

3.1.2 Cartridge-PCB

Im Wesentlichen wurde die Schaltung des letzten Semesters um folgende Komponenten erweitert: Ausschaltverzögerung, schnellere Pws, Lade- und Aufwärtswandlerschaltung für LiPo-Batterie, ADC zum Überwachen der Batteriespannung und einen MicroSD-Kartensteckplatz. Weiterhin befindet sich die Schaltung zum Auslesen einer Cartridge auf der Platine. Die gesamte Schaltung kann Abbildung 3.1 entnommen werden.

3 Implementierung

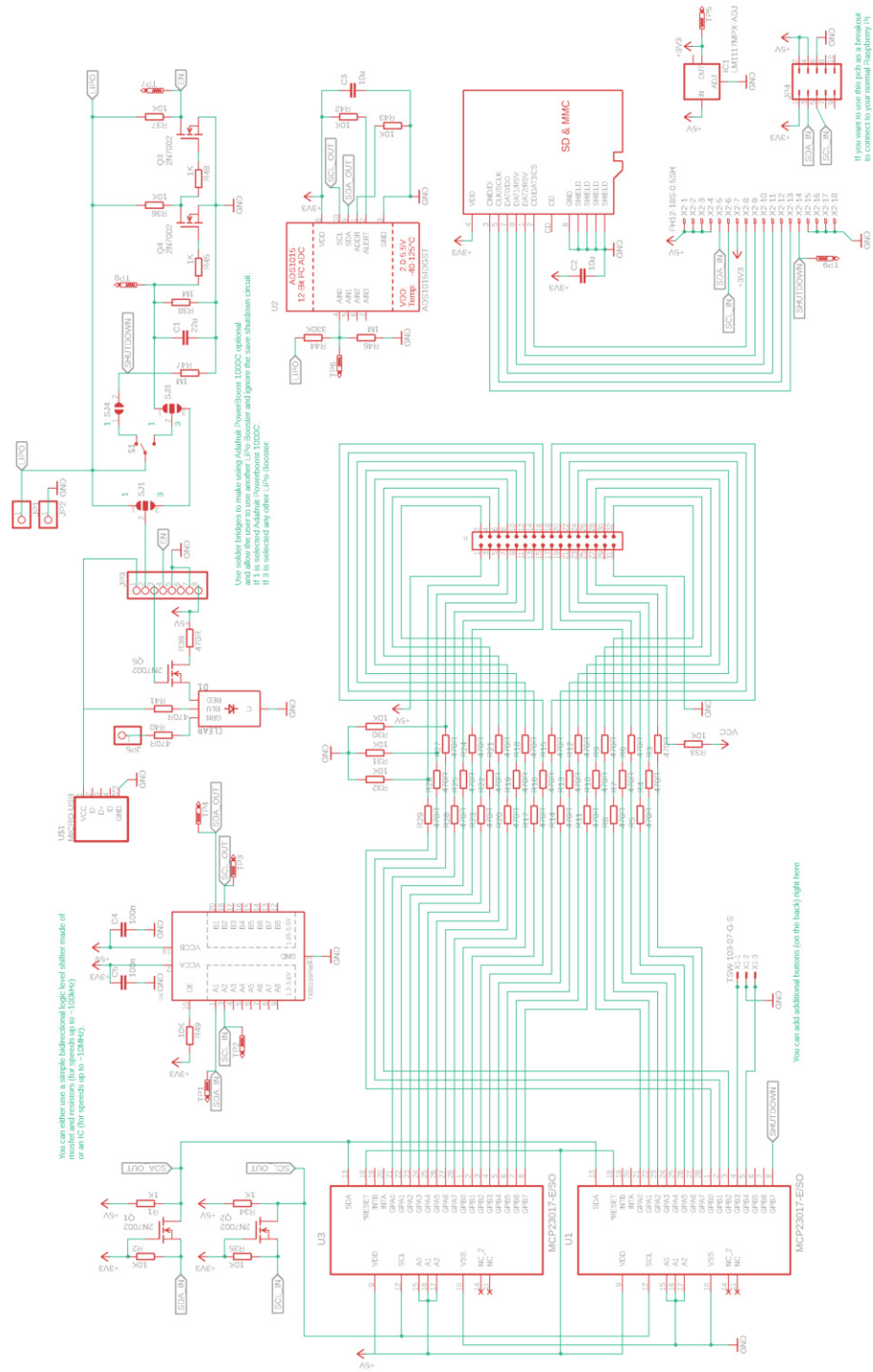
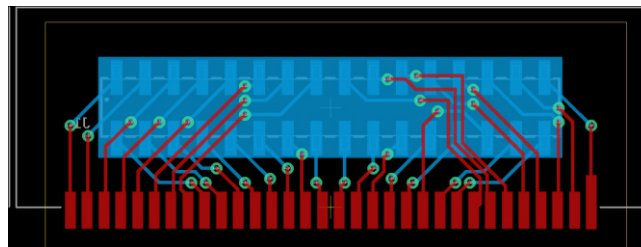


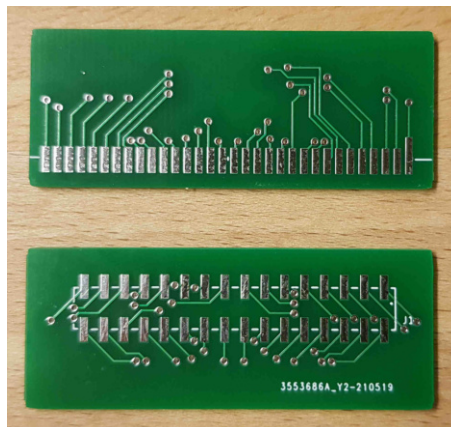
Abbildung 3.1: Schaltplan der Cartridge-PCB

Cartridge-Schnittstelle

Der einzige Unterschied der Cartridge-Schnittstelle ist, dass diese nun aus zwei Platinen besteht, die über eine männliche Stiftleiste verbunden werden. Dies liegt daran, dass im DMG ein höherer Cartridge-Adapter verwendet wurde. Da dieser nicht mehr gekauft werden kann, wurde ein niedrigerer Cartridge-Adapter (eigentlich für den NintendoDS) verwendet. Um die fehlende Höhe nun auszugleichen, befindet sich der Cartridge-Adapter auf einer eigenen Platine, welche über männliche Stiftleisten verbunden werden. Somit kann die Cartridge auf derselben Höhe wie beim DMG eingesteckt werden und das Bauteil passt in das DMG-Gehäuse. Das Layout der zweiten Platine kann in Abbildung 3.2 gesehen werden.



(a) Layout



(b) Platine

Abbildung 3.2: Cartridge-PCB Erweiterung

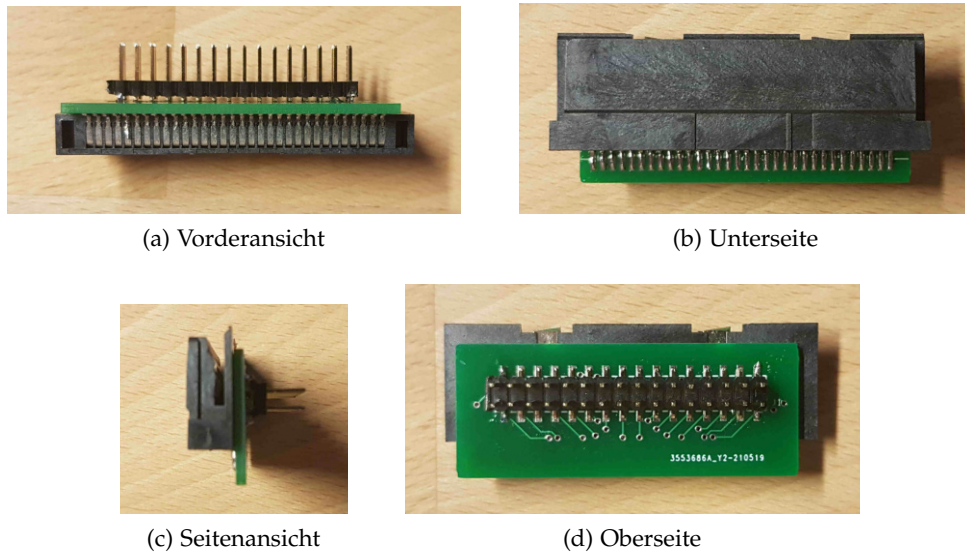


Abbildung 3.3: Bestückte Cartridge-PCB

Ausschaltverzögerung

In Kapitel 2.3 wurde bereits die Funktionsweise der Ausschaltverzögerung erklärt. Damit anstatt der geplanten Lade- und Aufwärtswandlerschaltung Adafruit PowerBoost 1000C [11] (welche den Enable-Pin für die Ausschaltverzögerung enthält) auch eine günstigere Schaltung verwendet werden kann, wurde die Ausschaltverzögerung optional entworfen. Mit Hilfe von drei Lötbrücken kann diese entweder aktiviert bzw. deaktiviert werden. So wird entweder eingestellt, dass der Schalter die Batteriespannung direkt trennt oder, wie in Kapitel 2.3 erklärt, über die Ausschaltverzögerung. Dies lohnt sich vor allem für Nutzer, welche nicht bereit sind, den hohen Preis von ca. 30,00€ für den Adafruit PowerBoost 1000C zu zahlen. Vergleichbare Platinen (welche über keinen Enable-Pin verfügen) gibt es schon ab ca. 4,00€ (vgl. Abbildung 3.4).

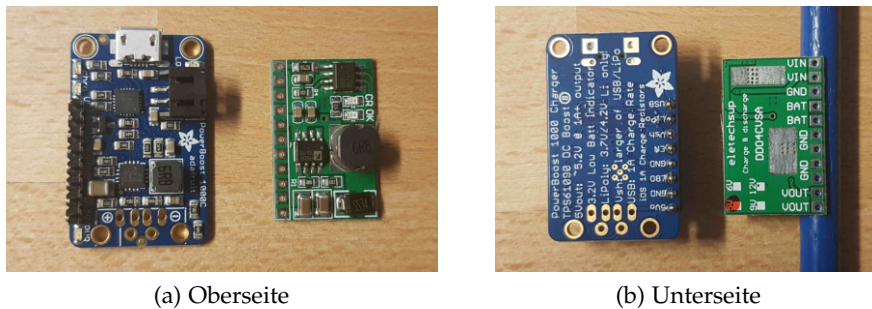


Abbildung 3.4: Adafruit PowerBoost 1000C in blau,
Günstige Lade- und Aufwärtswandlerschaltung in grün

ADC für Batterieüberwachung

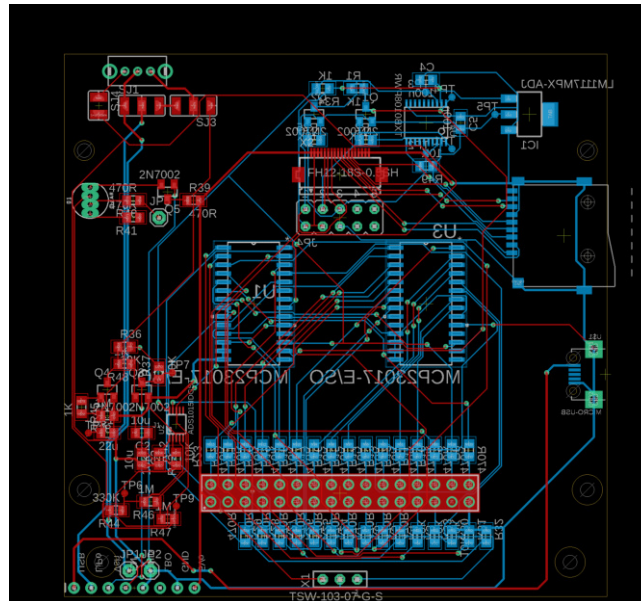
Um dem Nutzer die verbleibende Ladung des Akkus anzeigen zu können wurde ein ADC verbaut, mit welchem die Batteriespannung überwacht werden kann. Um Verbindungsleitungen zu sparen, wurde ein ADC mit I²C-Schnittstelle verwendet, da die I²C-Leitungen ohnehin für die MCP23017-ICs vorhanden sind.

Verbindungselemente (Schalter, MicroUSB, LED, MicroSD)

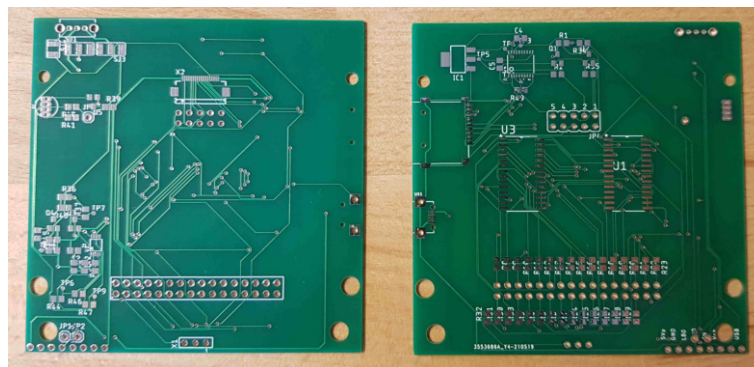
Das DMG-Gehäuse hat einige Aussparungen die für Schalter, Kontrast, Datentransfer, Lautstärkerad, Versorgungsspannung, usw. gedacht waren. Diese werden nun für andere Funktionen verwendet. Auf der Cartridge-PCB wurden hier ein Schalter, ein MicroUSB-Stecker zum Laden des Gerätes, eine MicroSD-Kartensteckplatz für das Dateisystem und eine RGB-LED zum Anzeigen des LiPo-Batteriezustandes verbaut. Da die Leitungen der MicroSD-Karte auf der Display-PCB benötigt werden, werden diese über den fpc weitergegeben (siehe Tabelle 3.2).

Platine

In Abbildung 3.5 kann das Layout der Cartridge-PCB gesehen werden.



(a) Layout



(b) Platine

Abbildung 3.5: Cartridge-PCB

3.1.3 Display-PCB

Da im Moment von RPi nur ein Footprint des CM4 für die neueste Version von KiCAD zur Verfügung gestellt wird, wurden Schaltplan und Layout des Display-PCB mit der Open Source Software in der Version 5.99 angefertigt. Die Platine basiert dabei auf dem GameBoy (GB) Projekt von Alex Hoffmann. Ein Link zu seinem GitHub Repository findet sich in [20]. Das PCB von Alex Hoffmann wurde ebenfalls in KiCAD erstellt und besaß bereits den richtigen Formfaktor mit Bohrlöchern zur Befestigung

3 Implementierung

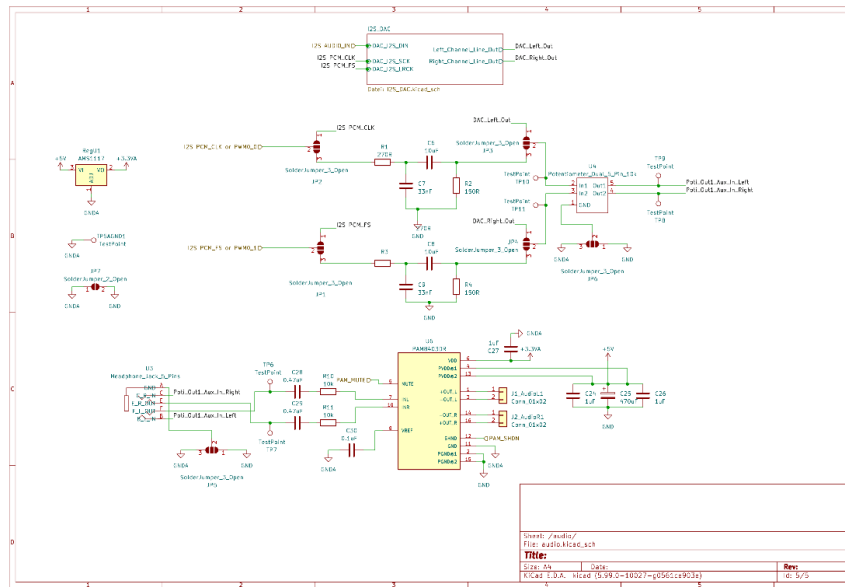


Abbildung 3.7: Display-PCB Audio

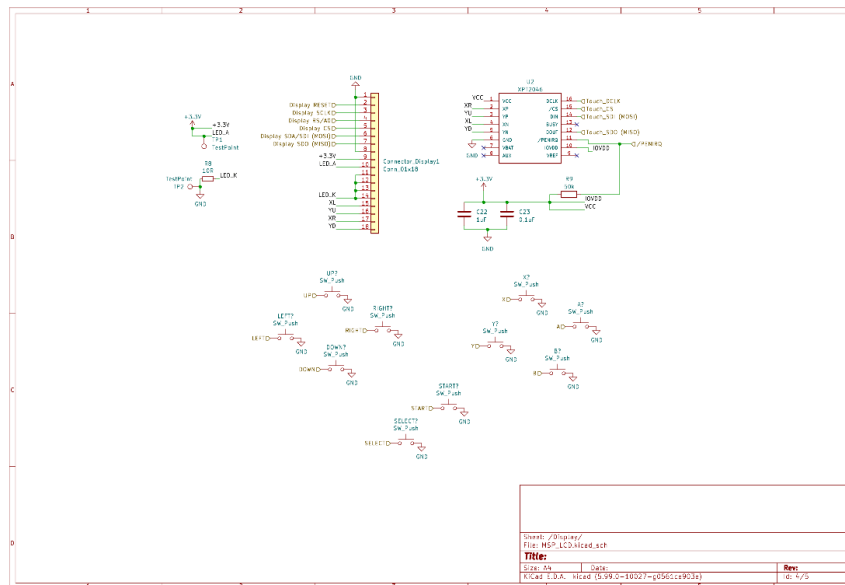


Abbildung 3.8: Display-PCB - Display mit Touchcontroller und Buttons

Vorderseite	Rückseite
Display	Sockel des CM4
Display Konnektor	Touchcontroller
Pads für Conductive Rubber Buttons	Lautstärkerad
Masseflächen	Schaltungen für I ² S und PWM Audio
	AUX-Buchse
	Konnektor zum Cartridge-PCB

Tabelle 3.3: Aufbau des Display-PCB

Wie bereits eingangs bereits erwähnt, wurden Formfaktor und Bohrlöcher von Alex Hoffmann größtenteils übernommen. Abbildung 3.10 zeigt den Grundriss des Display-PCB in gelber Farbe und die Position und Orientierung der wichtigsten Komponenten auf der Vorder- und Rückseite. Sämtliche Masseflächen und Leiterbahnen wurden dabei zur Übersichtlichkeit ausgeblendet. Die schwarzen Punkte überall auf dem Board sind die Bohrungen, über die die Platine in einem handelsüblichen GB Gehäuse angeschraubt werden kann. In wenigen Fällen, insbesondere durch die Platzierung des CM4, mussten einige Löcher entfernt werden. Dies bedeutet, dass auch im Gehäuse die entsprechenden Plastikstege nur als Auflage benutzt oder ganz herausgebrochen werden müssen.

Die Platine hat grundsätzlich eine rechteckige Form, die rechte untere Ecke wurde allerdings abgerundet ausgespart, da an dieser Position im GB Gehäuse der Lautsprecher vorgesehen ist. Ansonsten gibt es auf dem Display-PCB drei überstehende Komponenten. Auf der Unterseite steht die AUX-Buchse etwas über, um die Gehäuseöffnung ausfüllen zu können. Dasselbe gilt für das Lautstärkerad etwa in der Mitte der rechten Seite. Zudem wurde das CM4 etwas überstehend platziert. Die beiden Sockel sind dabei nach wie vor auf der Platine, jedoch ragt die untere rechte Ecke des CM4 etwas über die Aussparung des Display-PCB rechts unten, um möglichst viele Bohrlöcher erhalten zu können.

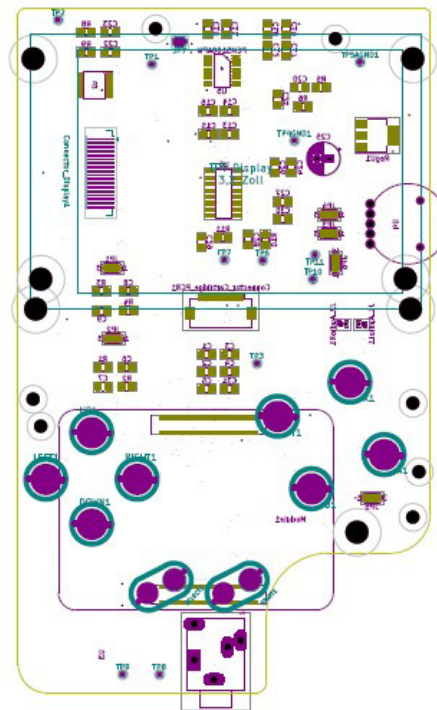


Abbildung 3.10: Formfaktor des Display-PCB

Audio

Für das Audio sind auf dem Display-PCB zwei verschiedene Varianten vorgesehen. Zunächst sollte wie in der Vorgängerversion von Felix Weichselgartner die Tonausgabe über PWM möglich sein. Als Alternative dazu sollte zudem I2S für das Audio verwendet werden können. Als Ausgänge wurden dafür sowohl eine 3,5mm Klinkenbuchse als auch ein Mono-Lautsprecher vorgesehen.

Für das PWM Audio wird das PWM0 Modul mit den folgenden General Purpose Input Output (GPIO) Pins verwendet:

CM4 Pin	Name	Funktion
GPIO 18	PWM0-0	linker Audio Channel
GPIO 19	PWM0-1	rechter Audio Channel

Tabelle 3.4: Pins für PWM Audio am CM4

Jeder der beiden Kanäle verfügt über einen Bandpass, wie er in Abbildung 3.11 zu sehen ist, um Rauschen und andere Störgeräusche aus dem Audiosignal herauszufiltern.

3 Implementierung

Der zugehörige Frequenzgang ist in Screenshot 3.12 gezeigt. Als Vorlage wurde dabei Quelle [21] verwendet.

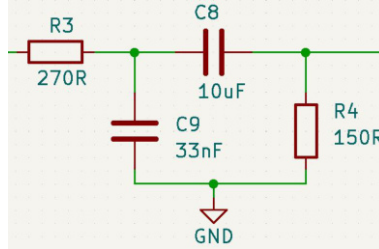


Abbildung 3.11: Bandpass-Filterschaltung für das PWM Ausgangssignal am CM4

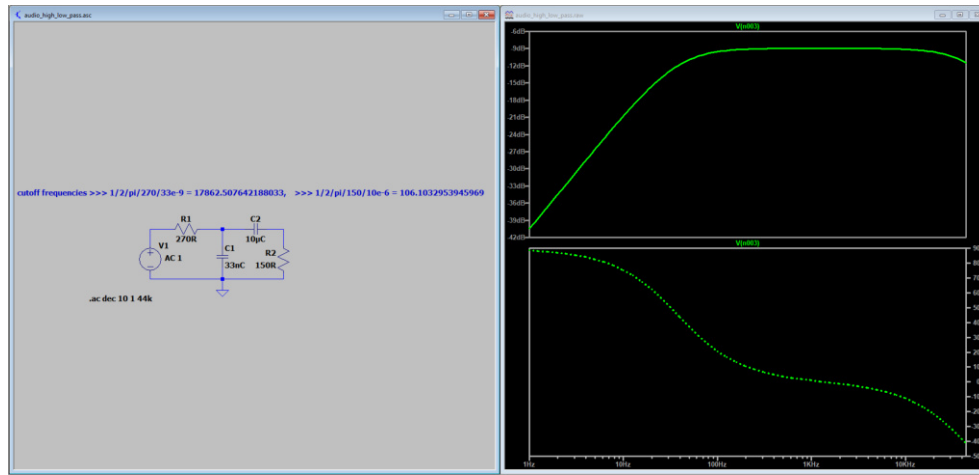


Abbildung 3.12: Frequenzgang der Bandpass-Filterschaltung aus LTSpice

Die Grenzfrequenzen des Bandpasses berechnen sich durch die Dimensionierung wie folgt:

$$f_{g,H} = \frac{1}{2\pi * R * C} = \frac{1}{2\pi * 270 \Omega * 33 \text{ nF}} \approx 18 \text{ kHz} \quad (3.1)$$

$$f_{g,L} = \frac{1}{2\pi * R * C} = \frac{1}{2\pi * 150 \Omega * 10 \mu\text{F}} \approx 100 \text{ Hz} \quad (3.2)$$

Das Audiosignal von I²S ist im Gegensatz zum PWM digital. Wie in Kapitel 2.7 bereits erklärt wurde, stellt I²S nur ein Protokoll dar. Die tatsächliche Ausgabe erfolgt am CM4 durch das Puls-Code-Modulation (PCM) Modul über die folgenden Pins:

CM4 Pin	Name	I ² S Funktion
GPIO 18	PCM-CLK	SCK
GPIO 19	PCM-FS	WS
GPIO 21	PCM-DOUT	SD

Tabelle 3.5: Pins für I²S Audio am CM4

Auffällig ist hierbei, dass die GPIO Pins 18 und 19 bei I²S und PWM verwendet werden. Dies ist der Mehrfachbelegung der GPIO Pins bei RPi geschuldet. Die einzelnen GPIO Pins können dabei über die Software, genauer gesagt über das `config.txt` File oder das Terminal für teilweise fünf verschiedene Einsatzzwecke konfiguriert werden. Auf den dazu notwendigen Code wird in Kapitel 3.2 ausführlich eingegangen. Eine detaillierte Tabelle für die alternativen Funktionen der GPIO Pins findet sich in [6]. Der Vorteil hier ist in jedem Fall, dass für die beiden Audiovarianten keine Pins verschwendet werden.

Das digitale I²S Signal ist das Eingangssignal für den PCM5100 Audio Stereo DAC. Dieser wandelt die I²S-Daten in zwei analoge Signale für den linken sowie den rechten Kanal eines Audioendgeräts. Er benötigt für die Wandlung, wie schon in Kapitel 2.5 angesprochen, sowohl eine analoge als auch eine digitale Masse und Spannungsversorgung. Aus diesem Grund wurde für den PCM5100 und einige weitere analoge Audiokomponenten mit dem AMS1117 Spannungswandler von Advanced Monolithic Systems Inc. eine eigene Versorgung mit 3,3V eingerichtet. Zusätzlich befinden sich diese Komponenten auf einer eigenen analogen Massefläche (AGND). Diese wiederum wurde nur in einem einzigen Punkt mit der normalen Massefläche für alle restlichen Bauteile verbunden. Die rudimentären Grundlagen für getrennte digitale und analoge Masse wurden bereits in Kapitel 2.5 behandelt. Für das Display-PCB ergibt sich daher folgende Aufteilung:

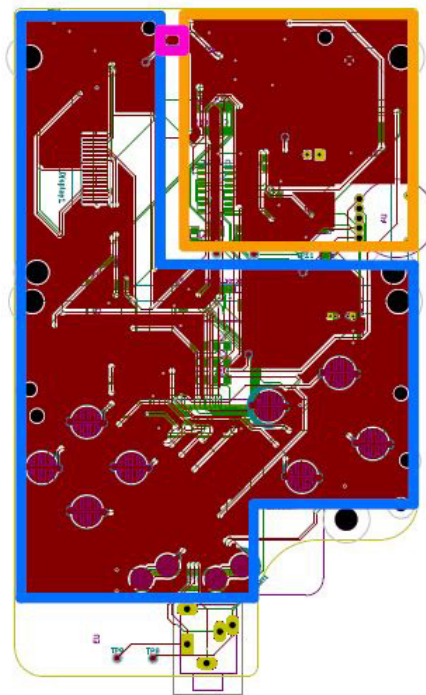


Abbildung 3.13: Masseflächen auf dem Display-PCB

Die **Massefläche der digitalen Schaltkreise** ist dabei blau umrandet, die **analoge Masse** wurde orange markiert. Nur über die violette Solder Bridge kann eine Verbindung zwischen den Masseflächen hergestellt werden.

Hardwareseitig erfolgt die Auswahl darüber, ob das Audio über I²S oder PWM verwendet werden soll, über Solder Bridges mit drei Kontakten. Zwei solche Brücken befinden sich an den Ausgängen der GPIO Pins 18 und 19 des CM4, um, wie bereits in Tabellen 3.4 und 3.5 verdeutlicht wurde, die verschiedenen Funktionen dieser Pins nutzen zu können. Falls softwareseitig das Audio über PWM konfiguriert ist, werden die Eingänge der beiden Solder Bridges mit den Ausgängen verbunden, die zu den Filtern für den rechten und linken Kanal des PWM Audio führen. Ist I²S am CM4 eingestellt, so können die Eingänge der beiden Solder Bridges mit den verbleibenden Ausgängen verbunden werden. Diese führen dann zu den Eingängen des PCM5100 Audio Stereo DAC. Für den dafür benötigten Code wird an dieser Stelle noch einmal auf das Kapitel 3.2 verwiesen.

Nach den Filtern bzw. dem Audio Stereo DAC muss das entsprechende Audio mit den Ausgängen, das heißt mit dem Potentiometer zur Lautstärkeregelung und in der Folge mit der AUX-Buchse und dem Lautsprecher verbunden werden. Hierfür sind

wieder zwei Solder Bridges mit jeweils 3 Kontakten, wie in Abbildung 3.14 gezeigt, vorgesehen. Mit diesen können wahlweise der linke und rechte Kanal des PWM Audios oder der linke und rechte Kanal des PCM5100 Audio Stereo DAC mit einem Dual-Potentiometer zur Lautstärkeregelung verbunden werden.

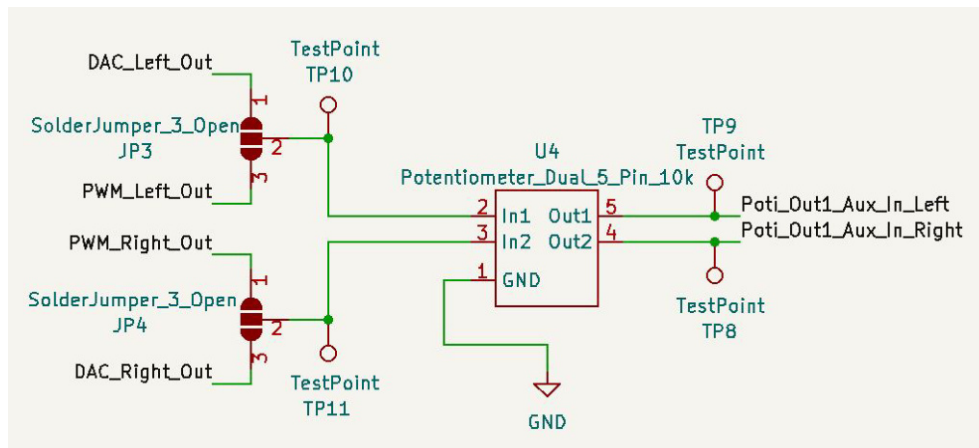


Abbildung 3.14: Solder Bridges vor dem Dual-Potentiometer

Mit einem Dual-Potentiometer lässt sich die Lautstärke zweier Audiosignale, in diesem Fall linker und rechter Kanal, mit nur einer Drehbewegung am Lautstärkerad regeln. Durch die Solder Bridges können beide Audioquellen mit dem Dual-Potentiometer verbunden und geregelt werden.

Die Ausgänge des Dual-Potentiometer führen auf eine 5-polige AUX-Buchse. Diese verfügt über zwei Pins mehr als eine gewöhnliche Klinkenbuchse. Zur Erklärung sind in Abbildung 3.15 das Symbol, die Pins auf der Unterseite der Buchse und ein 3,5mm Klinkenstecker abgebildet. Darunter die Bedeutungen der Bezeichnungen in Tabelle 3.6 festgehalten.

https://www.distrelec.de/Web/Downloads/he/et/ST-3519_en_datasheet.pdf

Abbildung 3.15: 5-polige Klinkenbuchse [22]

Pin	Funktion
A	Masse
B	linker Kanal
C	rechter Kanal
E	Öffner Kontakt rechter Kanal
F	Öffner Kontakt linker Kanal

Tabelle 3.6: Bezeichnungen der 5-poligen AUX-Buchse

Die 5-polige AUX-Buchse zeichnet sich durch zwei mechanische Öffner für den linken und den rechten Audiokanal aus. Das bedeutet, falls kein AUX-Stecker eingesteckt ist, die Pins C und E bzw. B und F verbunden sind und die Audiosignale weitergegeben werden können. Wie in Abbildung 3.16 zu sehen ist, sind die Pins E und F der 5-polige AUX-Buchse mit den Eingängen für linken und rechten Audiokanal des PAM8403 Audio Amplifiers verbunden. Dieser verstärkt die Audiosignale für einen Mono Lautsprecher. Ist jedoch ein Kopfhörer o.ä. in der Klinkebuchse eingesteckt, dann wird die Verbindung zum Amplifier und damit zum Mono Lautsprecher unterbrochen und die Musik wird nur über die Klinke ausgegeben. Wichtig zu erwähnen ist dabei, dass die Lautstärke in jedem Fall durch das Potentiometer geregelt werden kann, da dieses sich vor diesen Komponenten in der Schaltung befindet.

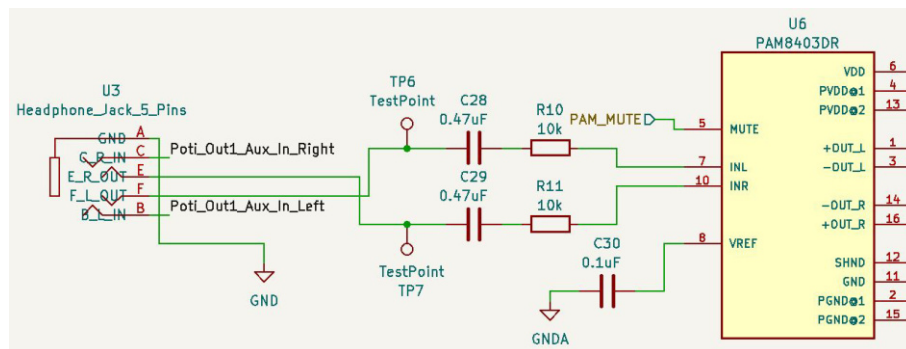


Abbildung 3.16: Verschaltung von Klinkebuchse und PAM8403 Amplifier

Ist kein Klinkestecker eingesteckt, erhält, wie bereits angesprochen, der PAM8403, ein 3W class-D Stereo Audio Amplifier von Power Analog Microelectronics Inc., die Audiosignale für rechten und linken Kanal. Von dessen Stereo-Ausgangssignal wird nur das Audio für den rechten Kanal verwendet, da aus Platzgründen nur ein Mono Lautsprecher für den GB vorgesehen ist. Trotzdem wurde auf dem Display-PCB die Möglichkeit vorgesehen, einen zweiten Lautsprecher für den linken Channel anzulöten.

Display

Als Display kommt ein 3,2 Zoll SPI TFT LCD-Display mit einer Auflösung von 240 x 320 Pixeln zum Einsatz. Dieses verfügt außerdem über einen resistiven Touchscreen. Das LCD-Display kann über ein Flachbandkabel mit 18 Pins bei einem Pitch von 0,8mm am Display-PCB angeschlossen werden. Die Anzeige wird dabei über das SPI0 Modul des CM4 angesteuert. Die Belegung der GPIO Pins am CM4 sieht dafür wie folgt aus:

CM4 Pin	Name	Funktion
GPIO 7	SPI0-CE1-N	Chip Enable (CE)1 für Touchscreen
GPIO 8	SPI0-CE0-N	CE0 für Display
GPIO 9	SPI0-MISO	Master Input Slave Output (MISO)
GPIO 10	SPI0-MOSI	Master Output Slave Input (MOSI)
GPIO 11	SPI0-SCLK	Serial Clock (SCLK)

Tabelle 3.7: Pins des SPI0 Moduls am CM4

Des Weiteren werden zwei Spannungsversorgungen für die Anzeige bzw. für die LEDs der Hintergrundbeleuchtung sowie ein normaler GPIO Pin für ein Reset-Signal benötigt. Der Touchscreen kann durch Spannungsmessung an den vier Pins X-Positiv (XP), X-Negativ (XN), Y-Positiv (YP) und Y-Negativ (YN) - in Abbildung 2.13 auch als X+, X-, Y+ und Y- bezeichnet - ausgelesen werden. Die Grundlagen hierzu wurden bereits in Kapitel 2.6 erklärt. Die Spannungsmessung erfordert allerdings analoge Eingangspins, über welche das CM4 nicht verfügt. Aus diesem Grund wurde zum Auslesen des Touchscreens der XPT2046 Touchscreen Controller von Shenzhen XPTEK Technology Ltd. hinzugefügt. Dieser besitzt genau die vier zuvor genannten analogen Eingänge und gibt ein SPI Signal an das CM4 weiter. Dazu wurde der XPT2046 ebenfalls mit dem SPI0 Modul verbunden. Wie in obiger Tabelle bereits festgehalten wurde, ist der Touchcontroller aber über den CE1 Pin von SPI0 erreichbar, während CE0 für die Anzeige des Displays vorgesehen ist.

Buttons

Auf der Vorderseite des Display-PCB sind insgesamt acht Pads für Conductive Rubber Buttons zum Spielen vorgesehen. Das Steuerkreuz links besteht aus Up, Down, Left und Right. An der Unterseite befinden sich Doppel-Pads für Start und Select. Rechts befinden sich beim GB eigentlich nur die Tasten A und B. Um aber auch andere Spiele auf dem Emulator spielen zu können, wurden auch Pads für die Buttons X und Y hinzugefügt. Für diese beiden muss jedoch das Gehäuse an der entsprechenden Stelle aufgebohrt werden.

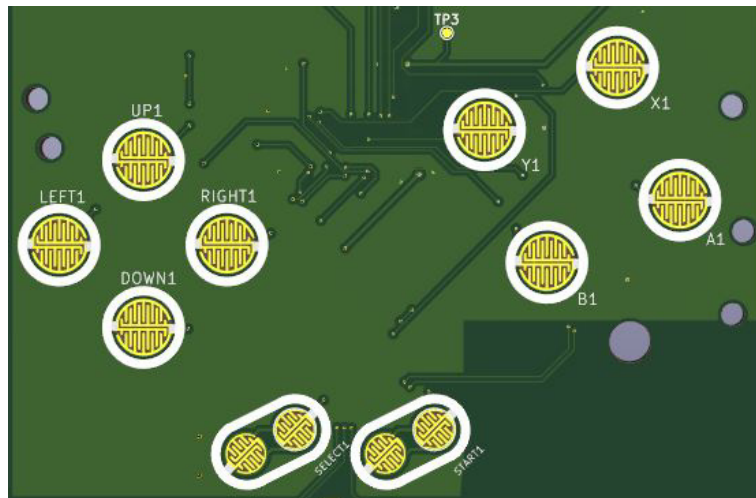


Abbildung 3.17: Buttons

3.2 Software

3.2.1 Cartridge-Reader

Die Software blieb im Wesentlichen unverändert und nur kleine Änderungen wurden vorgenommen, welche die Geschwindigkeit des Auslesens der Cartridge beschleunigen sollen.

Zum Ansteuern der MCP23017-ICs wurde im letzten Semester die WiringPi-GPIO-Bibliothek [23] für den RPi verwendet. Diese nutzt nicht alle Funktionen des MCP23017 aus, weshalb jeder Pin einzeln angesteuert (beschrieben oder gelesen) werden muss und nicht ein ganzer Port (8 Pins) auf einmal beschrieben werden kann. Dies dauert erheblich länger. Bisher konnte noch keine OpenSource-Bibliothek gefunden werden, welche diese Funktionen auf einem RPi in der Programmiersprache C oder C++ bietet. Trotzdem konnte die Geschwindigkeit signifikant verbessert werden. Wie bereits erwähnt, wird wesentlichen Anteil der Zeit wird für das Setzen der Adress-Pins benötigt. Bisher wurde hier jeder Pin jedes Mal neu gesetzt und nicht darauf geachtet ob sich die Pins überhaupt ändern. Der Code wurde nun so geändert, dass die letzte Adresse gespeichert und überprüft wird, ob sich der Pin von der letzten zur aktuellen Adresse ändert. Ändert er sich nicht wird kein Befehl an das I²C-Modul geschickt und die Zeit kann gespart werden. Dies reduziert die benötigte Zeit zum Auslesen einer Cartridge auf ca. 50%.

```
void GPIO::setAddress(Word address) {
    for (int i = 0; i < 16; i++) {
        // this reduced this time needed for cartridge reading about 50%
        // as we only need to write if the pin actually changed.
        if (((address & 0xFFFF) & (1 << i))
            != ((lastAddress & 0xFFFF) & (1 << i))) {
            digitalWrite(
                base_io_expander_address + i,
                ((address & 0xFFFF) & (1 << i)) == (1 << i)
            );
        }
    }

    lastAddress = address;
}
```

Abbildung 3.18: Setzen einer Adresse

Um die Geschwindigkeit weiter zu erhöhen kann die Geschwindigkeit des I²C-Bus erhöht werden. Da die I²C-Befehle einen Großteil der Zeit ausmachen, kann durch ein Verdoppeln der Geschwindigkeit die benötigte Zeit fast halbiert werden. Die maximale Baudrate des MCP23017 ist 1,7MHz, wohingegen bisher eine Baudrate von 100kHz verwendet wurde. Somit sollte hiermit die Zeit um erneut mehr als 90% verkürzt werden können. Um die hohe Baudrate von 1,7MHz nutzen zu können wurde, wie in Kapitel 2.4 erklärt, ein IC für die Pegelwandlung auf der Platine verbaut. Die I²C-Geschwindigkeit kann in der „/boot/config.txt“-Datei des RPi-Dateisystems mit dem folgenden Befehl verändert werden:

```
dtoverlay=i2c-arms, i2c-arms-baudrate=1700000
```

Hiermit sollte das Auslesen von einer ROM-Bank in ca. 12s möglich sein, was bisher eine Zeit von 4min benötigt hat. Durch das Ansteuern eines ganzen Ports auf einmal, mit einer geeigneten Bibliothek, sollte es möglich sein an die Zeit des Arduino Shields [24] von 7,7s anzuschließen. Mit der SPI-Version des MCP23017 (MCP23S17) wäre es denkbar, dass die Zeit sogar unterboten werden kann.

3.2.2 Display-PCB

Um beim CM4 einen Touchscreen mit SPI ansteuern und Audio über I²S bzw. PWM ausgeben zu können, müssen einige weitere Konfigurationen softwareseitig vorgenommen werden. Ausprobiert wurden diese Einstellungen auf einem RPi Model 3B+ mit Raspberry Pi OS in der Version 5.10, da zum Testzeitpunkt das Display-PCB noch nicht gefertigt wurde und daher kein Sockel zum Aufstecken des CM4 zur Verfügung stand. Da RPi über die verschiedenen Modelle ihrer Einplatinenrechner die Nummerierung, Bezeichnung, Anzahl sowie die Konfigurationsmöglichkeiten für SPI, I²S etc. der GPIO Pins beibehält, dürfte das im Folgenden erläuterte Vorgehen für das CM4 ebenso anwendbar sein.

Beim Audio muss selbstverständlich nur die Variante aktiviert werden, die über die speziellen Solder Bridges auf dem Display-PCB verbunden wurde. In jedem Fall muss dazu das config.txt File auf der SD-Karte, welche im Betriebssystem des RPi enthalten ist, bearbeitet werden. Für das Audio über PWM muss nach [25] nur folgende Zeile am Ende der Datei hinzugefügt werden:

```
dtoverlay=pwm-2chan, pin=18, func=2, pin2=13, func2=4
```

Für das Audio über I²S sind nach [26] und [27] einige Schritte mehr notwendig:

```
# 1. Deaktivieren des Default Audio durch Auskommentieren der naechsten  
# Zeile durch den "#" am Anfang:  
#dtparam=audio=on  
  
# 2. Einschalten des I2S Interface durch Hinzufuegen der folgenden Zeilen:  
#I2S DAC  
dtparam=i2s=on  
dtoverlay=hifiberry-dac  
dtdebug=1
```

Nach der Bearbeitung des config.txt Files muss der RPi mit eingesteckter SD-Karte neu gestartet werden.

Um einen Touchscreen über SPI betreiben zu können, muss das config.txt File nicht verändert werden. Stattdessen ist es notwendig, auf dem RPi - mit installiertem Betriebssystem - das Terminal zu öffnen und einige Befehle auszuführen. Für das RPi Model 3B+ mit Raspberry Pi OS wurde dabei nach der Anleitung in [28] wie folgt vorgegangen:

```
$ sudo rm -rf LCD-show
$ git clone https://github.com/goodtft/LCD-show.git
$ chmod -R 755 LCD-show
$ cd LCD-show/
$ sudo ./LCD35-show
```

In der Beschreibung des 'LCD-Show' GitHub Repositorys von goodtft (nachzulesen in [29]) wurde vermerkt, dass für das Betriebssystem RetroPi statt den Treibern für Raspberry Pi OS, die oben durch den Download des Repositorys mit dem zweiten Befehl installiert werden, Treiber aus einem anderen Repository verwendet werden müssen. Dieses wurde in [30] verlinkt. RetroPi wiederum ist ein Betriebssystem für den RPi, welches Emulatoren für verschiedenste Konsolenspiele der älteren Generation bereitstellt und daher am Ende auch in dem im Rahmen dieser Arbeit entwickelten GB zum Einsatz kommen soll.

Da das Display durch das Flachbandkabel und die Position des zugehörigen Konnektors auf dem Display-PCB eine feste Ausrichtung besitzt, kann es u.U. notwendig sein, die Anzeige des Displays zu rotieren. Dies kann mit den folgenden Befehlen bewerkstelligt werden:

```
$ cd LCD-show/
$ sudo ./rotate.sh 90
```

Zur Kalibrierung des Touchscreens wird in [28] die Installation des Programms 'xinput calibrator' mit den folgenden Befehlen empfohlen:

```
$ cd LCD-show/
$ sudo dpkg -i -B xinput-calibrator_0.7.5-1_armhf.deb
```

Das Programm kann anschließend im Menü unter 'Einstellungen -> Touchscreen kalibrieren' ausgeführt werden.

4 Testen

Im letzten Semester wurde bereits eine Platine gefertigt, mit der die Funktionalität der Cartridge-PCB überprüft werden kann. Hierbei handelte es sich um eine Vorlage [31] bei der die Pins der Cartridge mit Pads zum Anlöten von Kabeln verbunden wurden. Um das Anschließen zu beschleunigen und Fehler zu verhindern, wurde die Platine verbessert (siehe Abbildung 4.1). Auf der neuen Platine können männliche Stiftleisten eingelötet werden, womit die Platine auf einen Arduino Mega Mikrocontroller aufgesteckt werden kann (siehe Abbildung 4.2).

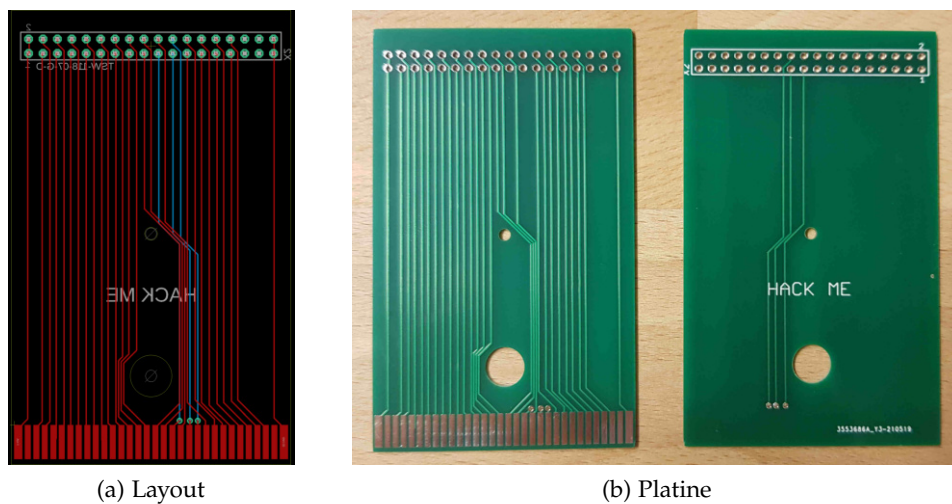
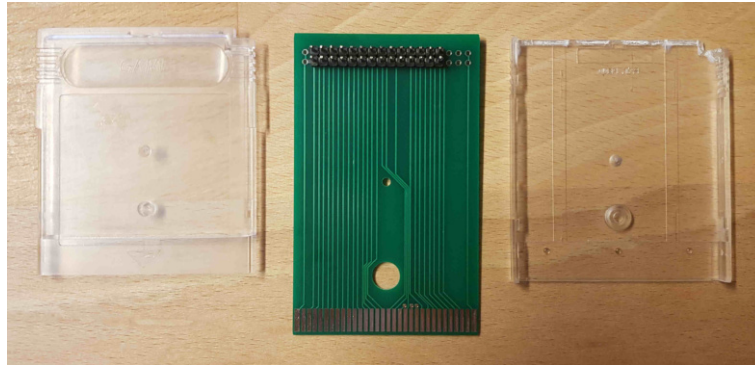


Abbildung 4.1: PCB zum Testen der Cartridge-PCB



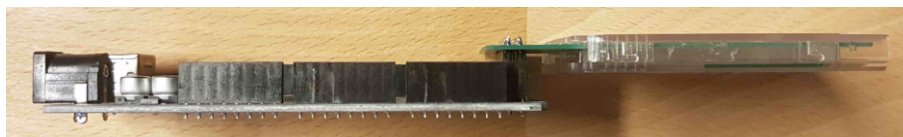
(a) Einzelteile



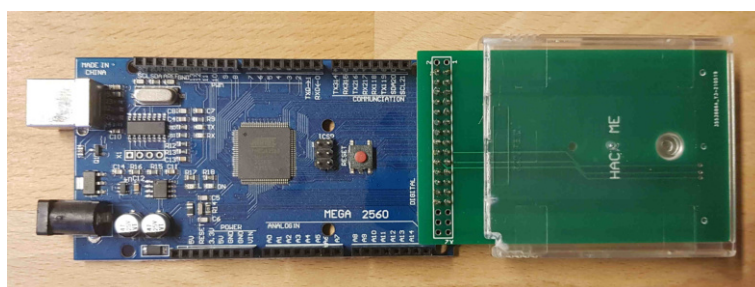
(b) Unterseite



(c) Oberseite



(d) Eingesteckt Seitenansicht



(e) Eingesteckt Draufsicht

Abbildung 4.2: Bestückte Debug Cartridge für Arduino Mega

5 Ergebnisse

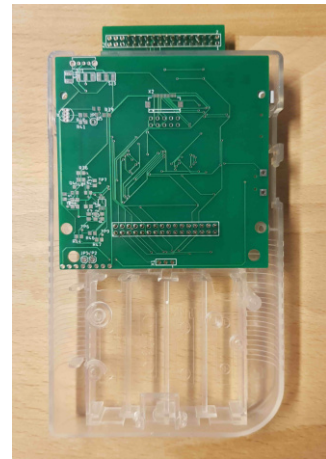
Im Laufe des Projekts wurden beide Platinen erstellt und angefertigt (vgl. Abbildung 3.3, 3.5, 5.1, 5.2). Des Weiteren wurde die Platine zum Testen der Cartridge-PCB verbessert (vgl. Abbildung 4.2). Aus zeitlich Gründen konnten nicht alle Platinen rechtzeitig bestückt und getestet werden.



(a) Layout



(b) Platine



(c) Platine

Abbildung 5.1: Cartridge-PCB mit eingestecktem Cartridge-PCB-Tester

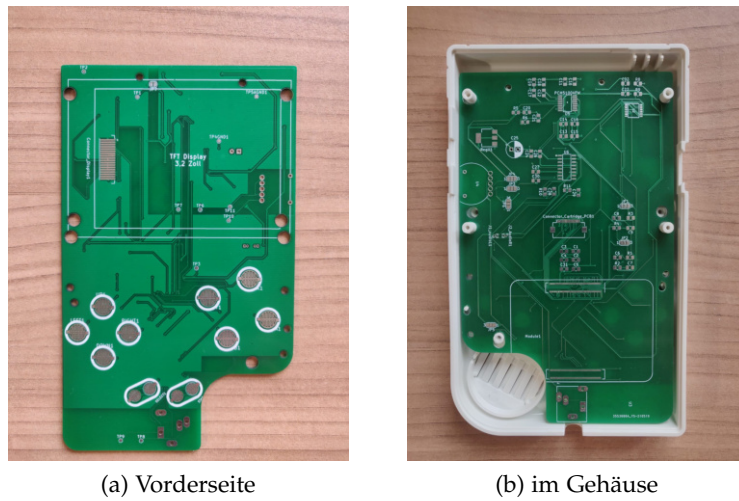


Abbildung 5.2: Display-PCB

Der weitere Verlauf des Projekts kann in [32] verfolgt werden. Hier stehen alle Projektdateien als Open-Source Projekt unter dem Namen PiHeld zur Verfügung.

Literaturverzeichnis

- [1] "Raspberry Pi Zero W," <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>, Zugriff: 17.12.2020.
- [2] "ZeroBoy: Geschichte," <https://www.zeroboy.eu/geschichte/>, Zugriff: 05.12.2020.
- [3] F. Weichselgartner, "GameBoy Zero," https://felixweichselgartner.github.io/2019/01/29/GameBoy_Zero.html, Zugriff: 12.12.2020.
- [4] —, "GitHub: GameBoy-Classic-Emulator," <https://github.com/FelixWeichselgartner/GameBoy-Classic-Emulator>, Zugriff: 07.12.2020.
- [5] —, "GB Cartridge Reader Writer," https://felixweichselgartner.github.io/2019/09/01/GB_Cartridge_Reader_Writer.html, September 2019, Zugriff: 07.12.2020.
- [6] "Raspberry Pi Compute Module 4," <https://datasheets.raspberrypi.org/cm4/cm4-datasheet.pdf>, Zugriff: 07.05.2021.
- [7] "Raspberry Pi Compute Module IO Board," <https://datasheets.raspberrypi.org/cm4io/cm4io-datasheet.pdf>, Zugriff: 07.05.2021.
- [8] J. Geerling, "The Raspberry Pi Compute Module 4 Review," <https://www.jeffgeerling.com/blog/2020/raspberry-pi-compute-module-4-review>, Zugriff: 08.05.2021.
- [9] R. Zwetsloot, "Raspberry Pi 4 specs and benchmarks," <https://magpi.raspberrypi.org/articles/raspberry-pi-4-specs-benchmarks>, Zugriff: 07.05.2021.
- [10] G. Halfacree, "Hands-On with the Raspberry Pi Compute Module 4 and Compute Module 4 Lite," <https://www.hackster.io/news/hands-on-with-the-raspberry-pi-compute-module-4-and-compute-module-4-lite-827bb3293ee9>, November 2020, Zugriff: 01.07.2021.
- [11] "Adafruit Powerboost 1000C - Overview," <https://learn.adafruit.com/adafruit-powerboost-1000c-load-share-usb-charge-boost>, Zugriff: 22.05.2021.

- [12] F. Weichselgartner, "GitHub: Adafruit_PowerBoost_SaveShutdown," https://github.com/FelixWeichselgartner/Adafruit_PowerBoost_SaveShutdown, Zugriff: 29.04.2021.
- [13] —, "GitHub: RPi-I2C-LevelShifter," <https://github.com/FelixWeichselgartner/RPi-I2C-LevelShifter>, Zugriff: 29.04.2021.
- [14] *TXB0108 8-Bit Bidirectional Voltage-Level Translator with Auto-Direction Sensing and ± 15 -kV ESD Protection*, Texas Instruments Inc., August 2020, <https://www.ti.com/lit/ds/symlink/txb0108.pdf>.
- [15] Z. Peterson, "Wie Sie einen Sternpunkt für analoge und digitale Masseverbindungen verwenden," <https://resources.altium.com/de/p/how-to-use-a-star-point-for-analog-ground-digital-ground-connection>, Dezember 2020, Zugriff: 14.06.2021.
- [16] *Audio Stereo DAC with 32-bit, 384kHz PCM Interface*, Texas Instruments Inc., September 2012, <https://www.ti.com/lit/gpn/pcm5100>.
- [17] J. Calpe, "AD7879 Controller Enables Gesture Recognition on Resistive Touch Screens," <https://www.analog.com/en/analog-dialogue/articles/gesture-recognition-on-resistive-touch-screens.html#>, Juni 2011, Zugriff: 14.06.2021.
- [18] R. Keim, "Introduction to the I2S Interface," <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2s-interface/>, März 2020, Zugriff: 14.06.2021.
- [19] wdwd, "I²S," https://de.wikipedia.org/wiki/I%C2%B2S#/media/Datei:I2S_Timing.svg, Juni 2020, Zugriff: 14.06.2021.
- [20] A. Hoffman, "Alex Hoffman Gameboy DMG Raspberry Pi Board," <https://github.com/alxhoff/gameboy-dmg-pcb>, April 2021, Zugriff: 06.08.2021.
- [21] J. List, "Behind the Pin: How the Raspberry Pi gets its Audio," <https://hackaday.com/2018/07/13/behind-the-pin-how-the-raspberry-pi-gets-its-audio/>, Juli 2018, Zugriff: 14.06.2021.
- [22] *ST-3519 - 3,5mm Stereo Jack*, November 2006, https://www.distrelec.de/Web/Downloads/he/et/ST-3519_en_datasheet.pdf.
- [23] "Wiring Pi - GPIO Interface library for the Raspberry Pi: I2C: MCP23008 & MCP23017," <http://wiringpi.com/extensions/i2c-mcp23008-mcp23017/>, Zugriff: 05.12.2020.

- [24] F. Weichselgartner, "GitHub: GB-Cartridge-Reader-Writer," <https://github.com/FelixWeichselgartner/GB-Cartridge-Reader-Writer>, Zugriff: 07.12.2020.
- [25] —, "GameBoy Zero," https://felixweichselgartner.github.io/2019/01/29/GameBoy_Zero.html, Januar 2019, Zugriff: 20.06.2021.
- [26] Mirko, "Carrier board for Raspberry Pi Compute Module 4," <https://hackaday.io/project/177626-mirkopc-cm4-carrier-board>, April 2021, Zugriff: 20.06.2021.
- [27] —, "MirkoPC Manual - Yet another Raspberry Pi Carrier Board," <https://cdn.hackaday.io/files/1776267599550784/MirkoPC%20user%20manual%20v1.0%202021-04-14.pdf>, April 2021, Zugriff: 20.06.2021.
- [28] "3.5inch RPi Display," http://www.lcdwiki.com/3.5inch_RPi_Display, April 2021, Zugriff: 14.06.2021.
- [29] gooftft, "LCD-show," <https://github.com/gooftft/LCD-show>, Juni 2021, Zugriff: 14.06.2021.
- [30] lcdwiki, "LCD-show-retropi," <https://github.com/lcdwiki/LCD-show-retropie>, Dezember 2021, Zugriff: 14.06.2021.
- [31] "Xyl2k - Homebrew Gameboy Color Cartridge," <https://github.com/Xyl2k/Gameboy-Color-Cartridge>, Zugriff: 19.12.2020.
- [32] F. Weichselgartner, "GitHub: PiHeld," <https://github.com/FelixWeichselgartner/PiHeld>, Zugriff: 13.06.2021.

Abbildungsverzeichnis

2.1	DMG Platinen	2
2.2	Vergleich verschiedener RPi Modelle [8]	3
2.3	Vergleich der Rechenleistung verschiedener RPi Modelle [9]	3
2.4	Vergleich der Leistungsaufnahme verschiedener RPi Modelle [9]	3
2.5	Rückseite des CM4 [10]	4
2.6	Schaltplan Ausschaltverzögerungsschaltung über RC-Glied	5
2.7	Simulation der Ausschaltverzögerungsschaltung über RC-Glied	5
2.8	Messung der Ausschaltverzögerungsschaltung über RC-Glied	6
2.9	Geschwindigkeitsmessung der MOSFET-Pws	6
2.10	Simulation der Maximalgeschwindigkeit der MOSFET-Pws	7
2.11	Bidirektionale Pws mit einem MOSFET und TXB0108	7
2.12	Pinbelegung des PCM5100 Audio Stereo DAC [16]	8
2.13	Aufbau (a) und Funktionsweise (b) eines resistiven Touchscreens [17]	9
2.14	Übertragung über I ² S [19]	9
3.1	Schaltplan der Cartridge-PCB	12
3.2	Cartridge-PCB Erweiterung	13
3.3	Bestückte Cartridge-PCB	14
3.4	Vergleich Lade- und Aufwärtswandlerschaltung	15
3.5	Cartridge-PCB	16
3.6	Überblick über den Schaltplan des Display-PCB - Hauptseite	17
3.7	Display-PCB Audio	18
3.8	Display-PCB - Display mit Touchcontroller und Buttons	18
3.9	Display-PCB - Audio mit I ² S	19
3.10	Formfaktor des Display-PCB	21
3.11	Bandpass-Filterschaltung für das PWM Ausgangssignal am CM4	22
3.12	Frequenzgang der Bandpass-Filterschaltung aus LTspice	22
3.13	Masseflächen auf dem Display-PCB	24
3.14	Solder Bridges vor dem Dual-Potentiometer	25
3.15	5-polige Klinkenbuchse [22]	25
3.16	Verschaltung von Klinkenbuchse und PAM8403 Amplifier	26
3.17	Buttons	28

3.18	Setzen einer Adresse	29
4.1	PCB zum Testen der Cartridge-PCB	32
4.2	Bestückte Debug Cartridge für Arduino Mega	33
5.1	Cartridge-PCB	34
5.2	Display-PCB	35

Tabellenverzeichnis

3.1	Aufteilung der Funktionen auf Platinen	10
3.2	Verbindungspins der Platinen	11
3.3	Aufbau des Display-PCB	20
3.4	Pins für PWM Audio am CM4	21
3.5	Pins für I²S Audio am CM4	23
3.6	Bezeichnungen der 5-poligen AUX-Buchse	26
3.7	Pins des SPI0 Moduls am CM4	27

Abkürzungsverzeichnis

DMG Klassischen GameBoy

GB GameBoy

ROM Read Only Memory

RAM Random Access Memory

IC Integrated Circuit

I²C Inter-Integrated Circuit

SPI Serial Peripheral Interface

I²S Inter-IC Sound

PWM Pulsweitenmodulation

PCM Puls-Code-Modulation

UART Universal Asynchronous Receiver Transmitter

MSB Most Significant Bit

WS Word select

SD Serial Data

SCK Serial Clock

GPIO General Purpose Input Output

AGND analogen Massefläche

SCLK Serial Clock

MOSI Master Output Slave Input

MISO Master Input Slave Output

CE Chip Enable

RPi Raspberry Pi

CM3 Compute Module 3

CM4 Compute Module 4

eMMC embedded MultiMedia Card

ffc flexible Flachbandkabel

fpc flexible Leiterplatten

ADC Analog-Digital-Umsetzer

DAC Digital-Analog-Umsetzer

Pws Pegelwandlerschaltung

XP X-Positiv

XN X-Negativ

YP Y-Positiv

YN Y-Negativ