

Hochschule Darmstadt
Fachbereich Media
Sound and Music Production
Wintersemester 2021/2022

Prototypische Entwicklung eines Modular-Synthesizers im Eurorack- Format auf Basis von zeitgenössischen Kompositionstechniken

Abschlussarbeit zur Erlangung des akademischen Grades Bachelor of Arts (B.A.)

Abgabedatum: 31.01.2022
Vorgelegt von: Matthias Hassel
Matrikelnummer: 760976

Erstgutachter: Prof. Thorsten Greiner
Zweitgutachter: Prof. Dr. Jürgen Schwab

Abstract

Deutsch

Das Interesse an modularen Synthesizern steigt seit der Einführung des Eurorack-Formats beständig an. Die immer größer werdende Auswahl an Modulen, stellt Neueinsteiger:innen vor die schwierige Entscheidung, mit welchen Modulen das erste eigene Modular-System sinnvollerweise angefangen wird.

Ziel dieser Arbeit soll es daher sein, einen möglichst produktnahen Prototyp eines Modular-Synthesizers zu entwickeln, der als Einstiegsmodul Neueinsteiger:innen einen erfolgreichen Start in das Eurorack-Format ermöglicht. Zur Definition der grundlegenden Designrichtlinien werden die unterschiedlichen Ansätze zu Modular-Systemen von Donald Buchla und Robert Moog untersucht. Weiterhin soll der Funktionsumfang des Prototyps auf der Basis von zeitgenössischen Kompositionstechniken gewählt werden. Dazu werden unter anderem die „Modi mit begrenzter Transponierbarkeit“ von Olivier Messiaen sowie das Instrumentalstück „On the Run“ der britischen Band Pink Floyd näher betrachtet.

In der Arbeit werden die Konzeption und Realisierung des Modular-Synthesizers, sowie die Ergebnisse einer Fokusgruppenevaluation beschrieben, in der ein erster Prototyp von sechs Personen bezüglich seiner Funktionalität und seines Designs bewertet wird. Auf Basis dieser Evaluation wurde anschließend ein zweiter Prototyp gebaut, der die gesammelten Verbesserungsvorschläge der Evaluation aufgreift. Als Entwicklungsplattform für den Modular-Synthesizer wird das *Bela Board* des *Augmented Instruments Laboratory* der *Queen Mary University of London* verwendet. Die Arbeit kommt schließlich zu dem Schluss, dass der zweite Prototyp ein sinnvolles Einstiegsmodul für ein Modular-System im Eurorack-Format darstellt. Ein eingebauter Default Patch, der es Nutzer:innen ermöglicht einen vorgegebenen Signalfluss zu aktivieren erweist sich dabei als nützliche Hilfe, um Einsteiger:innen die Bedienung zu erleichtern.

English

Interest in modular synthesizers has been growing steadily since the introduction of the Eurorack format. The ever-growing selection of modules confronts newcomers with the difficult decision of choosing which modules to start their first modular system with. The aim of this thesis is to develop a prototype of a modular synthesizer that is as close to the product as possible, and that allows newcomers to the Eurorack format to get off to a successful start. To define the basic design guidelines, the different approaches to modular systems by Donald Buchla and Robert Moog will be examined. Furthermore, the functional scope of the prototype will be chosen based on contemporary composition techniques. For this purpose, among others, the "modes with limited transposability" by Olivier Messiaen as well as the instrumental piece "On the Run" by the British band Pink Floyd will be examined more closely.

The paper describes the conception and realization of the modular synthesizer, as well as the results of a focus group evaluation, in which a first prototype is evaluated by six people regarding its functionality and design. Based on this evaluation, a second prototype was subsequently built, which takes up the collected suggestions for improvement from the evaluation. The Bela Board from the Augmented Instruments Laboratory at Queen Mary University of London is used as the development platform for the modular synthesizer.

The paper finally concludes that the second prototype is a reasonable entry-level module for a modular system in Eurorack format. A built-in default patch, which allows users to activate a predefined signal flow, proves to be a very useful feature.

Inhaltsverzeichnis

<u>ABKÜRZUNGSVERZEICHNIS.....</u>	7
<u>1. EINLEITUNG.....</u>	8
1.1. FRAGESTELLUNG UND ZIELSETZUNG.....	8
1.2. ZIELGRUPPE UND RELEVANZ	9
<u>2. BEGRIFFSKLÄRUNGEN UND -ERKLÄRUNGEN.....</u>	10
2.1. MODULAR-SYSTEM	10
2.2. PATCHEN	10
2.3. SEQUENZER	10
2.4. V/OCT SKALA	10
2.5. WAVEFOLDING.....	11
<u>3. RECHERCHE</u>	12
3.1. MODULARE SYNTHESIZER	12
3.1.1. HARALD BODE	13
3.1.2. ROBERT MOOG UND DONALD BUCHLA – EASTCOAST VS. WESTCOAST.....	13
3.1.3. ANALOG / DIGITAL	16
3.2. EURORACK / DOEPFER A-100.....	17
3.2.1. BEDEUTUNG	17
3.2.2. MECHANISCHE VORGABEN	18
3.2.3. TECHNISCHE VORGABEN.....	19
3.3. MARKTÜBERSICHT	22
3.3.1. MARKANTEILE VON MODULAR-SYSTEMEN.....	22
3.3.2. EINORDNUNG DES PROTOTYP.....	23
3.3.3. VERGLEICHBARE PRODUKTE / ABGRENZUNG.....	23
<u>4. ZEITGENÖSSISCHE KOMPOSITIONSTECHNIKEN</u>	25
4.1. WENDY CARLOS – SWITCHED ON BACH.....	25

Inhaltsverzeichnis

4.2. MORTON SUBOTNICK – SILVER APPLES OF THE MOON.....	26
4.3. PINK FLOYD – ON THE RUN.....	27
4.4. OLIVIER MESSIAEN – MODI MIT BEGRENZTER TRANSPONIERBARKEIT.....	28
5. KONZEPTION	30
5.1. DESIGN	30
5.1.1. SOUNDDESIGN / FUNKTIONSUMFANG.....	30
5.1.2. MECHANISCHES KONZEPT.....	32
5.1.3. VISUELLES KONZEPT.....	32
5.2. TECHNIK	33
5.2.1. ANFORDERUNGEN	33
5.2.2. MICROCONTROLLER / EMBEDDED COMPUTING SYSTEM	34
5.2.3. BELABOARD.....	35
6. REALISIERUNG PHASE 1	38
6.1. SOFTWARE.....	38
6.1.1. PROGRAMMIERUMGEBUNG / WAHL DER PROGRAMMERSPRACHE.....	38
6.1.2. PURE DATA AUF DEM BELABOARD	39
6.1.3. INPUT/OUTPUT HANDLING.....	42
6.1.4. AUDIOSYNTHESIZE	46
6.2. HARDWARE.....	50
6.2.1. ERWEITERUNGEN ZUM BELABOARD.....	50
6.2.2. WEITERE ELEKTRONIK	51
6.2.3. FRONTPLATTE	52
7. EVALUIERUNG	53
7.1. METHODIK.....	53
7.2. DURCHFÜHRUNG	53
7.2.1. KRITERIEN ZUR GRUPPENEINTEILUNG.....	54
7.2.2. AUSWAHL DER TEILNEHMER:INNEN.....	54
7.2.3. FRAGENKATALOG	55
7.3. ERGEBNISSE.....	56
7.3.1. FUNKTIONALITÄT.....	56

Inhaltsverzeichnis

7.3.2. DESIGN.....	58
<u>8. REALISIERUNG PHASE 2</u>	<u>59</u>
8.1. HARDWAREANPASSUNGEN.....	59
8.1.1. ÜBERARBEITUNG DES DESIGNS DER FRONTPLATTE	59
8.1.2. ELEKTRONIKANPASSUNGEN.....	60
8.2. SOFTWAREANPASSUNGEN.....	62
8.2.1. ERNEUTE PROGRAMMIERUNG IN C++.....	62
8.3. NICHT REALISIERBARE ANPASSUNGEN	63
8.3.1. ERWEITERUNG DES SEQUENCERS	63
8.4. ÜBERSICHT FUNKTIONSUMFANG.....	64
<u>9. FAZIT.....</u>	<u>67</u>
9.1. KONZEPT UND REALISIERUNG	67
9.2. AUSBLICK.....	68
<u>ABBILDUNGSVERZEICHNIS</u>	<u>70</u>
<u>LITERATURVERZEICHNIS</u>	<u>71</u>
<u>EIGENSTÄNDIGKEIT UND ARCHIVIERUNG.....</u>	<u>76</u>
<u>DANKSAGUNG.....</u>	<u>77</u>
<u>DIGITALER ANHANG</u>	<u>78</u>

Abkürzungsverzeichnis

AES	Audio Engineering Society
API	Application Programming Interface
CPU	Central Processing Unit
DSP	Digital Signal Processor
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
CV	Control Voltage
IDE	Integrated Development Environment
Pd	Pure Data
VCA	Voltage Controlled Amplifier
VCF	Voltage Controlled Filter
VCO	Voltage Controlled Oscillator

1. Einleitung

1.1. Fragestellung und Zielsetzung

Seit der Einführung des Eurorack Systems 1996 von Dieter Doepfer, erleben Modular-Synthesizer eine Renaissance.¹ Die Vielfalt an erzeugbaren Klängen, die aus den häufig blinkenden Modulen hervorgehen, welche mit einem für Außenstehende oft unübersichtlichen Kabelsalat verbunden sind, erzeugt eine Faszination, die immer mehr Musikbegeisterte erreicht. Während die Auswahl an verfügbaren Modulen für Eurorack Modular-Synthesizer beständig größer wird, wächst damit aber auch die Entscheidungsvielfalt für den Kauf erster Module. Für Neueinsteiger:innen, die bisher wenig Vorerfahrungen mit Modular-Systemen sammeln konnten, kann diese Vielfalt überfordernd wirken. Verbunden mit dem hohen finanziellen Aufwand beim Start des ersten eigenen Modular-Systems stellt die Auswahl der ersten Module eine nicht einfach zu überwindende Hürde dar, die den Einstieg in die Modular-Synthesizer Welt erschwert. In dieser Arbeit wird daher die Konzeption und Realisierung eines Modular-Synthesizers im Eurorack-Format präsentiert, der eine mögliche Lösung zur Überwindung dieser Einstiegshürden darstellen könnte. Dazu wird zunächst die Entwicklung der ersten kommerziell erfolgreichen Modularsysteme von Robert Moog und Donald Buchla untersucht. Sowohl Buchla, als auch Moog haben durch ihre Modular-Systeme wichtige Prinzipien der modularen Audiosynthese definiert, auf denen auch heute noch ein großer Teil der erwerbbaren Modular-Synthesizer basieren.

Um die Klänge, die mit dem vorgestellten Modular-Synthesizer realisierbar sein sollen in einen musikalischen Kontext zu setzen, werden zeitgenössische Kompositionstechniken wichtiger Pioniere elektronischer Musik der Konzeption zu Grunde gelegt. Auf Basis des entwickelten Konzepts soll ein funktionsfähiger, physischer Prototyp entwickelt werden, der in Fokusgruppengesprächen von drei Fokusgruppen evaluiert wird. Hier wird geprüft, ob der entwickelte Prototyp als einfach zugängliches Einstiegsmodul für Neueinsteiger:innen in das Eurorack-System geeignet ist.

¹ Beat Rossmy und Alexander Wiethoff, „The Modular Backward Evolution – Why to Use Outdated Technologies“, *NIME*, 2019, 2.

1.2. Zielgruppe und Relevanz

Der Prototyp soll in erster Linie Musiker:innen ansprechen, die bisher keine praktischen Erfahrungen mit Modular-Synthesizern sammeln konnten und erst neu in die Modular-Welt einsteigen. Ergebnisse aus den Fokusgruppeninterviews bestätigen die These, dass Neueinsteiger:innen bei der Auswahl der ersten Module häufig unsicher sind, welche Module sie für einen zufriedenstellenden Start ihres Modular-Systems benötigen.² Der Prototyp bietet eine Vorauswahl von grundlegenden Komponenten der modularen Soundsynthese, die auf den recherchierten Prinzipien von Moog und Buchla, sowie auf den analysierten zeitgenössischen Kompositionstechniken beruht. Die Modular-Synthesizer Szene wird im hohen Maße von DIY-Entwickler:innen dominiert. Diese bauen eigene Module, die mit dem Eurorack-Format kompatibel sind und tragen damit erheblich zum Wachstum der Szene bei.

Für diese Gruppe an Personen ist das Produkt nicht gedacht, sie sind schon in der Modular Szene und profitieren daher wenig von der Einsteigerfreundlichkeit und dem übersichtlichen Aufbau des Prototyps. Nichtsdestotrotz besteht der Anspruch mit dem Prototyp auch für diese Zielgruppe eine qualitativ ausreichende Erweiterung des eigenen Systems zu bieten. Im [Ausblick](#) der Arbeit wird besprochen, wie bei einer Überarbeitung des Prototyps diese Gruppe möglicherweise stärker adressiert werden kann.

In der [Marktübersicht](#) wird gezeigt, dass die Nachfrage an Modulen für Eurorack Modular-Systeme steigt.³ Während bei eigenständigen Synthesizern der Einstieg durch eine Auswahl an explizit dafür vorgesehener Software- (Syntorial⁴, Ableton LearningSynths⁵) oder Hardwarelösungen (Dato Duo⁶) vereinfacht wird, gibt es im Modularbereich wenig physische Komplettlösungen, die speziell an Neueinsteiger:innen adressiert ist. Hier könnte eine Marktlücke für den in dieser Arbeit entwickelten Prototypen bestehen.

² „Expert:in1_Interview“, 12.2021, Abschn. 01:16:47.

³ Rossmay und Wiethoff, „The Modular Backward Evolution – Why to Use Outdated Technologies“, 1.

⁴ „Syntorial: The Ultimate Synthesizer Tutorial“, Syntorial, zugegriffen 18. Januar 2022, <https://www.syntorial.com/>.

⁵ „Learning Synths“, zugegriffen 18. Januar 2022, <https://learningsynths.ableton.com/>.

⁶ „Dato DUO“, zugegriffen 18. Januar 2022, <https://dato.mu/>.

2. Begriffsklärungen und -erklärungen

2.1. Modular-System

Bei einem modularen Synthesizer System, kurz Modular-System, sind die einzelnen Komponenten der analogen oder digitalen Audiosynthese als diskrete Baugruppen ausgeführt. Den Signalfluss der Klangsynthese muss dabei die bedienende Person selbständig festlegen.⁷ Eine ausführliche Beschreibung folgt im Kapitel [3.1 Modulare Synthesizer](#).

2.2. Patchen

Um bei Modular-Systemen den Signalfluss festzulegen, werden Ein- und Ausgänge einzelner Module mit Kabeln verbunden. Diesen Vorgang nennt man „patchen“. Beim Eurorack-System, welches im Abschnitt [3.2 Eurorack / Doepfer A-100](#) näher beschrieben wird, werden dabei üblicherweise 3,5mm Klinkenkabel verwendet.⁸

2.3. Sequenzer

Ein Sequenzer generiert in vorgegebenen zeitlichen Abständen Noteninformationen, die unter Anderem von Synthesizern in hörbare Klänge umgesetzt werden können. Dabei durchläuft er in Abhängigkeit von einer Tempoinformation eine feste Anzahl von Schritten einer Sequenz, welche als Modulationsquelle für unterschiedliche Parameter wie Tonhöhe oder Lautstärke genutzt werden kann. Am Ende der Sequenz beginnt der Sequenzer in der Regel wieder mit dem ersten Schritt.

2.4. V/Oct Skala

Die Voltage per Octave Skala in analogen Modular-Systemen gibt vor, dass bei einem spannungsgesteuerten Oszillatormodul, im englischen Voltage Controlled Oscillator (VCO), eine Spannungsänderung von einem Volt am Steuereingang des Moduls, eine Tonhöhenänderung von einer Oktave folgen muss.⁹ Aus dieser Vorgabe ergibt sich ein exponentieller Zusammenhang zwischen Spannung und Frequenz.

⁷ Peter Manning, *Electronic and Computer Music*, Rev. and expanded ed (Oxford ; New York: Oxford University Press, 2004), 112–13.

⁸ Rob Kam, „Eurorack“, Synth DIY Wiki, 19. Oktober 2021, <https://sdiy.info/wiki/Eurorack>.

⁹ Chris Meyer, „1 v/Oct | Learning Modular“, 16. November 2016, <https://learningmodular.com/glossary/1-voct/>.

2.5. Wavefolding

Wavefolding ist die Bezeichnung für eine spezielle Art der Verzerrung von Audiosignalen, bei der Signalspitzen, die über einen definierten Grenzwert hinausgehen, nicht abgeschnitten, sondern durch eine Reihe von Faltungen invertiert werden.¹⁰ Diese Faltungen erzeugen bei einfach harmonischen Wellenformen ein reiches Obertonspektrum.¹¹

¹⁰ Emmett Corman, „Simple Synthesis: Part 8, Wavefolding“, Blog, Keith McMillen Instruments, 17. Juni 2015, <https://www.keithmcmillen.com/blog/simple-synthesis-part-8-wavefolding/>.

¹¹ Corman.

3. Recherche

3.1. Modulare Synthesizer

Bei modularen Synthesizern sind im Gegensatz zu eigenständigen (monolithischen) Synthesizern die einzelnen Funktionen der Klangerzeugung und Klangbearbeitung (Oszillator, Filter, Hüllkurvengenerator, Verstärker, ...) als diskrete Baugruppen, sogenannte Module, realisiert.¹² Der Signalfluss der verschiedenen Audio- und Steuersignale muss dabei von der bedienenden Person durch eigene Verkabelung definiert werden.

*Kennzeichnend [...] für Modulsysteme ist die Tatsache, daß wesentliche Parameter klangerzeugender Baugruppen (z.B. VCO, NOISE) und klangformender Baugruppen (z.B. VCF, VCA) nicht nur manuell per Regler [sic!] sondern zusätzlich auch mit Hilfe von **Steuerspannungen** (engl. control voltage) eingestellt werden können.¹³*

Dieses Konzept einer modularen Bauweise analoger Synthesizer wurde 1960 von Harald Bode entwickelt und veröffentlicht.¹⁴

Ender der 1960er Jahre wurden daraufhin die ersten kommerziell erfolgreichen Modular-Systeme nahezu zeitgleich, jedoch unabhängig voneinander, von Robert Moog und Donald Buchla entwickelt.¹⁵ Weitere erwähnenswerte Pioniere sind Paul Ketoff und Tony Furse, die parallel zu Buchla und Moog Mitte der 60er eigene modulare Synthesizer entwickelten.¹⁶

Da ich mich bei der Entwicklung meines Prototyps besonders auf die Grundlagen von Donald Buchla und Robert Moog berufe, wird im Folgenden nicht näher auf die Arbeiten von Ketoff und Furse eingegangen.

¹² Manning, *Electronic and Computer Music*, 112.

¹³ Doepfer Musikelektronik GmbH, „Technische Hinweise A-100“, zugegriffen 30. Dezember 2021, https://doepfer.de/a100_man/a100t_d.htm.

¹⁴ David Dunn, „A History of Electronic Music Pioneers“, *Eigenwelt Der Apparatewelt: Pioneers of Electronic Art*, 1992, 20.

¹⁵ Joseph Akins, „An Overview of Electronic Music History“, 10. Oktober 2019, 11, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.497.7245&rep=rep1&type=pdf>.

¹⁶ Dunn, „A History of Electronic Music Pioneers“, 23.

3.1.1. Harald Bode

Zwischen Ende 1959 und 1960 entwickelte Harald Bode mit dem „Audio System Synthesizer“ den ersten bekannten Synthesizer, dem ein modulares Konzept der Audiosynthese zu Grunde liegt.¹⁷ Der Audio System Synthesizer sollte es den bedienenden Personen ermöglichen, bis dahin bekannte Elemente der Klangsynthese wie Filter, Ringmodulatoren, Hallgeneratoren, u.a., in beliebiger Anordnung zu kombinieren. Eine spannungsgesteuerte Verstärkereinheit, im englischen Voltage Controlled Amplifier (VCA), führte außerdem auch das Konzept der Steuerspannungen mit in das System ein.

Harald Bode präsentierte das System im Oktober 1960 bei einer Versammlung der Audio Engineering Society (AES) als:

“A New Tool for the Exploration of Unknown Electronic Music
Instrument Performances”¹⁸

In seinem gleichnamigen Paper aus dem Journal der AES, veröffentlicht im Oktober 1961, bezeichnet Bode die Vielfalt neuer Klänge und Anwendungen seiner Module als unbegrenzt.¹⁹

3.1.2. Robert Moog und Donald Buchla – Eastcoast vs. Westcoast
1964 adaptierte der britische Elektrotechniker und Physiker Robert Moog das modulare Konzept von Bodes Audio System Synthesizer und entwickelte in New York eines der beiden ersten kommerziell erfolgreichen Modular-Systeme²⁰ im Auftrag des amerikanischen Komponisten Herbert Deutsch.²¹

¹⁷ Simon Crab, „The ‘Sound Processor’ or ‘Audio System Synthesiser’ Harald Bode, USA, 1959“, 120 Years of Electronic Music (blog), 30. Januar 2014, <https://120years.net/wordpress/the-sound-processor-harald-bode-germany/>.

¹⁸ Harald Bode, „A New Tool for the Exploration of Unknown Electronic Music Instrument Performances“, *J. Audio Eng. Soc.* 9, Nr. 4 (Oktober 1961).

¹⁹ Bode, 1.

²⁰ Harald Bode, „History of Electronic Sound Modification“, *J. Audio Eng. Soc.*, Oktober 1984, 4.

²¹ Akins, „An Overview of Electronic Music History“, 11.

Parallel dazu startete an Amerikas Westküste Donald Buchla einige Monate vor Moog mit der Entwicklung seines später als „Buchla 100 Serie“ bekannten Modular-Systems; Moogs Prototyp wurde hingegen früher veröffentlicht:

Buchla got started a few months ahead of Moog, but Moog's first prototype was finished in the summer of 1964; Buchla's appeared in the fall of 1965.²²

Beide Systeme unterscheiden sich trotz des gemeinsamen modularen Ansatzes grundsätzlich in verschiedenen Überlegungen zur Syntheseart und zum musikalischen Interface.

Moog verwendete bei seinem Modular-System die subtraktive Syntheseform und integrierte Filter und Hüllkurvengeneratoren (folgend mit dem englischen Ausdruck „Envelope“ bezeichnet) als zentrale Bestandteile.²³ Zunächst unsicher, welche Bedienungsart er für seinen Prototyp wählen sollte, entschloss sich Moog schließlich die klassische Klaviatur als Hauptinterface für bedienende Personen zu nutzen. Ausschlaggebend dafür war die Überlegung, dass zukünftigen Nutzer:innen des Synthesizers so eine bereits bekannte Bedienweise zur Verfügung stehe.²⁴ Dieser später oft als „East Coast Approach“ betitelte²⁵ Ansatz erleichterte Musiker:innen den Einstieg in Moogs Modularsystem und sorgte letztendlich für dessen großen kommerziellen Erfolg.²⁶

Donald Buchla sah in der Benutzung einer Klaviatur als musikalisches Interface eine Einschränkung der Bedienung seines Modular-Systems in klassische musikalische Strukturen:

²² Trevor Pinch und Frank Trocco, *Analog Days: The Invention and Impact of the Moog Synthesizer* (Cambridge, MA: Harvard University Press, 2002), 41.

²³ Francis Rumsey, „Modular Synths and Embedded Computing“, *J. Audio Eng. Soc.* 68, Nr. 3 (2020): 234.

²⁴ Mat Dagleish, „The Modular Synthesizer Divided: The Keyboard and Its Discontents“, 2016, 2.

²⁵ Rumsey, „Modular Synths and Embedded Computing“, 234.

²⁶ Rossmann und Wiethoff, „The Modular Backward Evolution – Why to Use Outdated Technologies“, 2.

*That is, we don't sit down at a conventional keyboard and expect to perform something other than 12-tones-per-octave music.*²⁷

Daher beschäftigte Buchla sich bei seiner Arbeit an der Buchla 100 Serie viel mit der Entwicklung neuer Interfaces.²⁸ Er rückte den Sequenzer (s. [Sequenzer](#)) als zentrales Steuerelement in den Fokus seines Systems und integrierte statt einem klassischen Keyboard 16 kapazitive Kontaktflächen als weitere Bedienmöglichkeit.²⁹

Als Syntheseform wählte Buchla im Gegensatz zu Moog die additive Synthese. Zur Beeinflussung des Obertonspektrums und der damit verbundenen Klangfarbe, nutzte er in seinem System verschiedene Arten zur Wellenformmanipulation, wie die Frequenzmodulation und Wavefolding (s. [Wavefolding](#)), statt den klassischen Filter des Systems von Robert Moog. Anstelle von Envelopes, gab es in der Buchla 100 Serie einen sogenannten „Function Generator“.³⁰

Eine detaillierte Beschreibung der Module der Buchla 100 Serie geht über den Umfang dieser Bachelorarbeit hinaus, eine Übersicht ist jedoch auf folgender Website zu finden: <https://megasynth.de/features/buchla-box/>

Ähnlich wie Robert Moog und Herbert Deutsch entwickelte auch Donald Buchla sein Modularsystem in enger Zusammenarbeit mit weiteren Komponisten:

*[...] Pauline Oliveros, Ramon Sender and Morton Subotnick of the San Francisco Tape Music Center commissioned a new modular Synthesizer from musician and engineer Donald Buchla.*³¹

Wenngleich Moogs Ansatz, die Klaviatur in den Vordergrund des Modular-Systems zu stellen, zu weitaus größerem kommerziellen Erfolg führte, beeinflusst Buchlas „West Coast“ – Ansatz bis heute nachhaltig elektronische und generative Musik.³²

²⁷ „Don Buchla Keyboard Magazine Interview - MOD WIGGLER“, zugegriffen 1. Januar 2022, <https://modwiggler.com/forum/viewtopic.php?t=235923>.

²⁸ Dunn, „A History of Electronic Music Pioneers“, 22.

²⁹ von Joker Nies, „Der Buchla-Sound: Synthesizer der besonderen Art“, KEYBOARDS, 14. März 2017, <https://www.keyboards.de/stories/der-buchla-sound-synthesizer-der-besonderen-art/>.

³⁰ Rumsey, „Modular Synths and Embedded Computing“, 235.

³¹ Manning, *Electronic and Computer Music*, 102.

³² Rossmann und Wiethoff, „The Modular Backward Evolution – Why to Use Outdated Technologies“, 2.

Erwähnenswerte Werke, die mit den beiden vorgestellten Modular-Systemen entstanden sind und wesentlich zu ihrem Erfolg beitrugen sind „Switched on Bach“ von Wendy Carlos (1968, Moog Synthesizer) und „Silver Appels of the Moon“ von Morton Subotnick (1968, Buchla Series 100).³³

3.1.3. Analog / Digital

Während die ursprünglichen Modular-Synthesizer ausschließlich mit analoger Technik realisiert worden sind, kamen mit zunehmenden Fortschritten in der Digitaltechnik ab Anfang der 70er Jahre mehr und mehr digitale Synthesizer auf den Markt.³⁴ Die analogen Schaltkreise alter Synthesizer waren ihren digitalen Nachkommen in vielerlei Hinsicht unterlegen. Ungenauigkeiten in der Stabilität von Frequenz und Amplitude, die benötigte Aufwärmzeit sensibler Bauteile wie Elektronenröhren und insbesondere die hohen Kosten, die mit dem Bau polyphoner Synthesizer mit Analogtechnik verbunden waren,³⁵ führten dazu, dass spätestens ab 1980 allgemein galt: Digitale Synthesizer werden analoge weitgehend verdrängen.³⁶

Der Vormarsch der Digitaltechnik bedeutet allerdings nicht, dass analoge Schaltungen in Modular-Systemen gänzlich verschwinden. Gerade der Aufschwung des Eurorack-Systems (s. [Eurorack / Doepfer A-100](#)) Anfang der 2000er zeigte, dass die Entwicklung analoger Module sowie analoge Module zum Eigenbau noch immer wichtige Bestandteile des elektronischen Musikmarktes sind.³⁷

Nichtsdestotrotz erwartet auch der Erfinder des Eurorack-Systems zukünftig einen weiteren Anstieg an rein digitalen Modulen.³⁸

³³ Rumsey, „Modular Synths and Embedded Computing“, 234.

³⁴ Antoine Hennion und Christophe Levaux, *Rethinking Music through Science and Technology Studies*, hg. von Antoine Hennion und Christophe Levaux, 1. Aufl. (Routledge, 2021), 174,

³⁵ Rossmy und Wiethoff, „The Modular Backward Evolution – Why to Use Outdated Technologies“, 2.

³⁶ Dalgleish, „The Modular Synthesizer Divided: The Keyboard and Its Discontents“, 10.

³⁷ Rumsey, „Modular Synths and Embedded Computing“, 234.

³⁸ DivKid, *The History and Future of Eurorack Modular Synths with Doepfer*, Interview, Thomann Synth Reactor 2019, 2019, Abschn. 26:30-27:00, <https://www.youtube.com/watch?v=CpyuqP5r9xU>.

3.2. Eurorack / Doepfer A-100

Die Doepfer Musikelektronik GmbH führte 1996 mit der Entwicklung des „A-100 Analog Modular-System“ einen neuen Standard für modulare Synthesizer-Systeme ein, welcher heute allgemein als „Eurorack“ bezeichnet wird.³⁹ Dabei ermöglichte das A-100 System durch feste mechanische und technische Vorgaben für Module, die Integration von mehreren unterschiedlichen Modulen zur Audiosynthese und -bearbeitung in einem gemeinsamen System. Nachdem das System ursprünglich mit 10 Modulen veröffentlicht wurde, gibt es nach Herstellerangaben zum Zeitpunkt dieser Bachelorarbeit 172 Module von Doepfer, von denen 38 zurzeit nicht mehr vertrieben werden.⁴⁰

3.2.1. Bedeutung

Die technischen und mechanischen Spezifikationen des „A-100 Analog Modular-System“, wurden von der Doepfer Musikelektronik GmbH als freier Standard veröffentlicht. Dadurch ist es bis heute auch anderen Herstellern und Privatpersonen möglich, eigene Module für das Eurorack zu entwickeln und diese mit den von Doepfer entwickelten Modulen des A-100 Systems frei zu kombinieren. Drei Jahre nach der Vorstellung des A-100 Grundrahmens, veröffentlichte die Firma „Analogue Systems“ mit der „RS Integrator“ Reihe, die erste Reihe von Eurorack kompatiblen Modulen, die nicht von Doepfer stammte. Durch die Verbreitung des A-100 Systems ermutigt, begannen in den frühen 2000ern einige weitere Firmen mit der Entwicklung und Produktion eigener Eurorack Module, unter ihnen „Make Noise“, „Plan B“, „Livewire“, „Tiptop Audio“, u.a.⁴¹

Auf der Seite <https://www.modulargrid.net/e/modules/home> sind zum Zeitpunkt dieser Arbeit für das Eurorack-System insgesamt 11.879 Module von 497 Herstellern gelistet.⁴²

³⁹ Wesley Groves u. a., „Beginner’s Guide to Eurorack: Case Basics, Power Supplies, and Your First Modules“, Blog, reverb.com, 6. Februar 2020, <https://reverb.com/news/beginners-guide-to-eurorack-case-basics-oscillators-filters>.

⁴⁰ Doepfer Musikelektronik GmbH, „A-100 Overview“, zugegriffen 30. Dezember 2021, <https://doepfer.de/a100.htm>.

⁴¹ Dalgleish, „The Modular Synthesizer Divided: The Keyboard and Its Discontents“, 11.

⁴² „Plan your modular synthesizer rack on ModularGrid“, zugegriffen 31. Dezember 2021, <https://www.modulargrid.net/e/modules/home>.

Damit ist das Eurorack-System das Modular-System mit den meisten verfügbaren Modulen auf dem Markt.

Der Vergleich mit dem Moog Unit System, welches mit 1.087 verfügbaren Modulen das zweitgrößte Modular-System ist, verdeutlicht die momentane Vormachtstellung des Eurorack-Systems auf diesem Markt.⁴³

3.2.2. Mechanische Vorgaben

Für den mechanischen Aufbau des A-100-Systems wird das weit verbreitete und international genormte 19-Zoll-System verwendet (DIN 41494 / IEC 297-3 / IEEE 1001.1).⁴⁴

Die Höhe und Breite der Module in einem Eurorack-System werden in definierte Höhen- und Tiefeneinheiten eingeteilt, welche im deutschsprachigen Raum mit HE (englisch: U, „Unit“) und TE (englisch: HP, „Horizontal Pitch“) abgekürzt werden.

Alle Frontplatten der Module haben eine einheitliche Höhe von 3HE und eine Breite, die immer einem Vielfachen von 1TE entspricht.

$$1 \text{ HE} = 1U = 44,45 \text{ mm} = 1,75 \text{ Zoll} = 1,75", 3\text{HE} = 3 \text{ U} = 133,4 \text{ mm}$$
$$1 \text{ TE} = 1\text{HP} = 5,08 \text{ mm} = 1/5 \text{ Zoll} = 1/5 "$$
⁴⁵

Zur Montage eines Eurorack Moduls sind auf der Frontplatte Montagelöcher für M3x6mm Schrauben vorgesehen. Die Montagelöcher sind dabei 3mm vom Mittelpunkt der Bohrung zum oberen (bzw. unteren) Rand entfernt und haben horizontal einen Abstand, der einem Vielfachen einer TE entspricht.

⁴³ „Plan your modular synthesizer rack on ModularGrid“.

⁴⁴ Doepfer Musikelektronik GmbH, „Mechanische Details A-100“, zugegriffen 30. Dezember 2021, https://doepfer.de/a100_man/a100m_d.htm.

⁴⁵ Doepfer Musikelektronik GmbH.

Recherche

Als Material für Frontplatten empfiehlt Doepfer 2mm dickes, eloxiertes Aluminium.

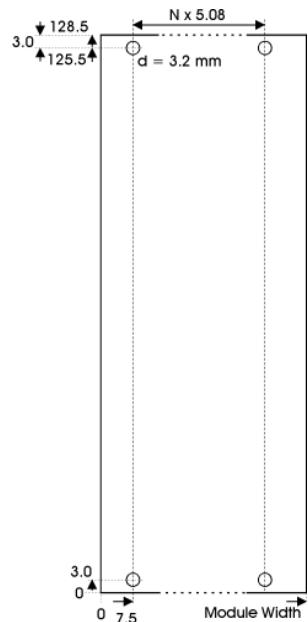
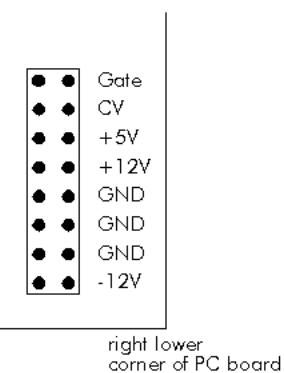


Abbildung 1: Skizze zur Lage der Montagelöcher⁴⁶

3.2.3. Technische Vorgaben

Die Eurorack Spezifikation sieht zur Spannungsversorgung der Module eine bipolare (+/-) 12V Gleichspannung vor, die im Rack über eine Busplatine an die einzelnen Module verteilt wird.⁴⁷ An diese Busplatine werden die verschiedenen Module über entweder 10-polige oder 16-polige Steckverbinder angeschlossen.

16 pin version:



10 pin version:

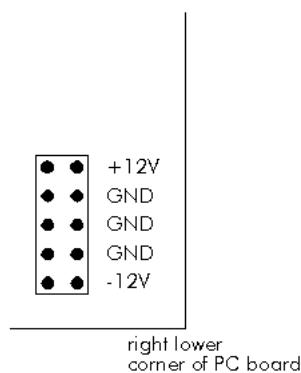


Abbildung 2: Steckverbinder Busplatine⁴⁸

⁴⁶ Doepfer Musikelektronik GmbH.

⁴⁷ Doepfer Musikelektronik GmbH, „Technische Hinweise A-100“.

⁴⁸ Doepfer Musikelektronik GmbH.

Recherche

Die Steuer- und Audiospannungen, die im Eurorack-System zwischen einzelnen Modulen gepatcht werden können, sind in drei Signalarten unterteilt:

- Audiosignale
- Steuersignale
- Trigger-/ Gate- oder Clocksignale

Für den Spannungsbereich der unterschiedlichen Signalarten gibt es in der Eurorack Spezifikation keine bindenden Vorgaben. Grundsätzlich gilt jedoch, dass bei Modulen externe Spannungen im Bereich +/- 12V die Module nicht zerstören dürfen.⁴⁹ Die genauen Signalpegel eines bestimmten Moduls sind in der Regel in der Modulbeschreibung des Herstellers angegeben.

Es folgt eine kurze Beschreibung der einzelnen Signalarten und ihre typischen Signalpegel bei Modulen des A-100 Systems:

Audiosignale sind im A-100 System von Doepfer in einem Spannungsbereich von +/- 5V (= 10Vss) definiert und stammen aus den klangerzeugenden Modulen im System.⁵⁰

Steuersignale dienen zur Einstellung der verschiedenen Parameter der klangerzeugenden bzw. klangformenden Baugruppen. Sie liegen entweder im Bereich von +/- 2,5 V (= 5Vss) bei Modulen wie einem Low Frequency Oscillator vor, oder haben einen Spannungsbereich von 0V bis 8V bei Hüllkurvengeneratoren (Envelope generators).⁵¹

Als **Trigger-/ Gate- oder Clocksignale** werden rechteckförmige Signale bezeichnet, die typischerweise einen Spannungspegel von 0/+5V haben. Im Falle eines Trigger-Signals, löst dabei der Übergang von 0V auf +5V (die positive Flanke) einen Vorgang im Modul aus, wie beispielsweise das Voranschreiten einer musikalischen Sequenz.⁵²

⁴⁹ Doepfer Musikelektronik GmbH.

⁵⁰ Doepfer Musikelektronik GmbH.

⁵¹ Doepfer Musikelektronik GmbH.

⁵² Doepfer Musikelektronik GmbH.

Recherche

So eindeutig die Definitionen der unterschiedlichen Signalarten auch sein mögen, ist in der Praxis eine klare Unterscheidung derselben häufig nicht erforderlich oder gar sinnvoll. Ein Audiosignal in Form einer Rechteckwelle kann beispielsweise auch als Trigger- oder Gate Signal für ein Sequencer Modul verwendet werden, oder als Steuerspannung die Grenzfrequenz eines Tiefpassfilters modulieren. Der fließende Übergang zwischen den Signalarten und die damit verbundene Flexibilität der Steuerungsmöglichkeiten in einem Modularsystem ist laut Doepfer die Grundlage für die Klangvielfalt und den Reiz von modularer Audio-Synthese:

Es gibt auch keine Vorschriften, wie die vorhandenen Module zu verbinden sind. Das Modulsystem ist ein offenes System, das (fast) alles erlaubt, und gerade das macht seinen Reiz und die enorme Klangvielfalt aus.⁵³

Für Oszillatormodule gilt im Eurorack die 1V/Oct Skala⁵⁴ (s. [Begriffsklärungen und -erklärungen](#)).

⁵³ Doepfer Musikelektronik GmbH, „A-100 Hauptseite/Einführung“, zugegriffen 1. Januar 2022, <https://doepfer.de/a100d.htm>.

⁵⁴ Doepfer Musikelektronik GmbH, „Technische Hinweise A-100“.

3.3. Marktübersicht

3.3.1. Marktanteile von Modular-Systemen

Der wachsende Umsatz an Modulen für Modular-Systeme⁵⁵ zeigt, dass trotz der allgemeinen Verfügbarkeit leistungsstarker Computer und digitaler Audio Workstations zur Produktion von elektronischer Musik, die Nachfrage nach Hardware Modular-Systemen steigt. In der freiwilligen Branchenstatistik der Society of Music Merchants (SOMM)⁵⁶, die zuletzt 2018 erhoben wurde, verzeichnete die Hauptwarengruppe Tasteninstrumente einen Umsatz von mehr als 170 Millionen Euro.⁵⁷ In dieser Warengruppe hat die Untergruppe Pianos, Synthesizer & Co, zu der auch Modular-Systeme zählen, einen Marktanteil von 17%. Speziell die Soundmodule konnten laut dieser Statistik 2018 eine Umsatzsteigerung von 16,9% im Vergleich zum Vorjahr erreichen.

Auch die wachsende Produktpalette des größten Online-Musikhändlers Thomann im Bereich „Modulare Systeme“ bestätigt die zuwachsende Nachfrage: 2016 führte Thomann insgesamt 300 Produkte von 15 Herstellern in der Kategorie „Modulare Systeme“⁵⁸; zum Zeitpunkt dieser Bachelorarbeit gibt es eine Auswahl von 1226 Produkten von 67 unterschiedlichen Herstellern.⁵⁹

In den nächsten Unterkapiteln wird der für diese Bachelorarbeit entwickelte Prototyp auf diesem Markt eingeordnet und von vergleichbaren Produkten abgegrenzt.

⁵⁵ „Studie: Tasteninstrumente gehören zu den Umsatzbringern der deutschen Musikinstrumentenbranche“, 8. Mai 2019, <https://www.musikmedia.de/musikbranche/studie-tasteninstrumente-gehoeren-zu-den-umsatzbringern-der-deutschen-musikinstrumentenbranche/>.

⁵⁶ Society of Music Merchants, „Pressemitteilung SOMM“, 2018, https://www.somm.eu/uploads/media/PM_SOMM_MI-Wirtschaft_2019_FINAL_190401_mn.pdf.

⁵⁷ „Studie“.

⁵⁸ Thomann GmbH, „Thomann, Modulare Systeme“, www.thomann.de, 14. Januar 2016, https://web.archive.org/web/20160114003953/http://www.thomann.de/de/modulare_systeme.html.

⁵⁹ Thomann GmbH, „Online-Katalog Herstellerübersicht – Musikhaus Thomann“, zugegriffen 3. Januar 2022, https://www.thomann.de/de/cat_brands.html?catKey=tasymo.

3.3.2. Einordnung des Prototyps

Der in dieser Bachelorarbeit vorgestellte Prototyp könnte prinzipiell in zwei Kategorien geführt werden. Da er in als Kombination aus unter anderem Oszillator-, Filter-, Envelope-, Sequenzer-, Mixer- und Utilitymodulen betrachtet werden kann (eine nähere Beschreibung des Funktionsumfangs folgt in [Sounddesign / Funktionsumfang](#)), könnte man ihn womöglich der Unterkategorie „Synthesizer Module“ zuordnen. Klassische Synthesizer Module sind allerdings ohne Sequenzer realisiert und bedürfen im Eurorack noch mindestens eines weiteren Moduls zur Ansteuerung der Soundsynthese. Da für diesen Prototyp kein weiteres Modul notwendig ist, wäre auch eine Einteilung in die Unterkategorie „Komplettsysteme“ möglich. Produkte in dieser Kategorie sind jedoch gegensätzlich zum Prototyp mit einem eigenen Gehäuse ausgestattet und nicht als Einbaumodul realisiert.

Da das Vergleichsprodukt, das im nächsten Unterkapitel „[Vergleichbare Produkte / Abgrenzung](#)“ vorgestellt wird, ebenfalls der Kategorie „Komplettsysteme“ zugewiesen ist, ist eine Einordnung in diese Kategorie sinnvoll.

3.3.3. Vergleichbare Produkte / Abgrenzung

In der Modular-System-Datenbank Modulargrid (www.modulargrid.net) sind aktuell vier auf dem globalen Markt verfügbare Synthesizermodule verzeichnet, die einen Sequenzer als Steuerelement integriert haben.⁶⁰ Unter diesen vier Produkten sind zwei für die Montage im Eurorack geeignet („Nerdseq“ von XOR Electronics und „Super Signal Processor“ von Percussa), die jedoch beide bei Betrachtung des Funktionsumfangs zu sehr vom vorgestellten Prototyp abweichen, um als vergleichbare Produkte gewertet zu können.

Einen sinnvolleren Vergleich mit anderen Produkten erhält man bei Betrachtung der Unterkategorie „Komplettsysteme“. Unter der Annahme, dass der Prototyp in einer Preiskategorie zwischen 300€ – 600€ einzuordnen ist, findet man beim Online Musikhändler Thomann fünf angebotene Komplettsysteme. Davon ist das „Nifty Bundle“ von cre8audio als einziges Komplettsystem mit dem Eurorack kompatibel.

⁶⁰ „Modules on ModularGrid“, zugegriffen 4. Januar 2022,

<https://www.modulargrid.net/e/modules/browser?SearchName=&SearchVendor=&SearchFunction=49&SearchSecondaryfunction=11&SearchHeight=&SearchTe=&SearchTemethod=max&SearchBuildtype=a&SearchLifecycle=&SearchSet=&SearchMarketplace=0&SearchIsmodeled=0&SearchShowothers=0&order=tag&direction=asc>.

Recherche

Das Nifty Bundle besteht aus einem Eurorack Gehäuse mit MIDI Eingang und zwei Modulen, in denen 2 Oszillatoren (davon ein Oszillator mit eingebautem Tiefpassfilter), ein LFO, und ein mit 16 Kontaktflächen programmierbarer Steuerspannungsgenerator enthalten sind.⁶¹ Das Bundle wird in Deutschland momentan für (vom ursprünglichen Marktpreis 585€ reduziert auf) 265€ angeboten.⁶²

Ähnlich zum vorgestellten Prototyp, verfolgt der Hersteller die Absicht mit seinem Komplettsystem einen einfachen Einstieg in das Eurorack-System zu ermöglichen. Hier überzeugt das Bundle vor allem durch seinen günstigen Preis und der Abgeschlossenheit des Systems: es liefert alle notwendigen Funktionen, um ohne weitere Produkte elektronische Musik kreieren zu können.⁶³

Darüber hinaus adressiert das Nifty Bundle jedoch, im Gegensatz zum Prototyp, nicht direkt Musiker:innen, die mittels einer möglichst einfachen Bedienung an die Prinzipien von Modular-Synthesizern herangeführt werden. Die einzelnen Module werden vom Hersteller als „Crazy Oscillators“, „funky Module[s]“ und „wacky sequencer“ beworben und heben sich durch eigenständige ungewöhnliche Sounds hervor.⁶⁴ Außerdem dürfe bei den Modulen keine Produkte erwartet werden die „precision-engineered to perfection“ hergestellt werden.⁶⁵ Damit adressiert das Nifty Bundle eher Personen, die bereits Vorkenntnisse mit modularen Systemen haben und auf der Suche nach besonders klingenden Modulen zu günstigen Preisen sind. Als Gegenentwurf liegt der Fokus des Prototyps mehr auf nachvollziehbarer Klangsynthese und einfachen Bedienmustern.

⁶¹ „NiftyBUNDLE - Affordable Powered Eurorack Synth Case, Modules, & Cables - Modular Mayhem for All!“, Cre8audio, zugegriffen 4. Januar 2022, <https://www.cre8audio.com/niftybundle>.

⁶² Thomann GmbH, „cre8audio NiftyBundle – Musikhaus Thomann“, zugegriffen 4. Januar 2022, https://www.thomann.de/de/cre8audio_niftybundle.htm.

⁶³ „Sound on Sound - Cre8Audio NiftyBundle Review“, Review, August 2020, <https://www.soundonsound.com/reviews/cre8audio-niftybundle>.

⁶⁴ „Cellz - Touch Controller, Sequencer, Arpeggiator & All Around Wacky Eurorack Compatible Synth Module“, Cre8audio, zugegriffen 18. Januar 2022, <https://www.cre8audio.com/cellz>.

⁶⁵ Cre8audio, „About Us“, Cre8audio, zugegriffen 18. Januar 2022, <https://www.cre8audio.com/about-us>.

4. Zeitgenössische Kompositionstechniken

Im folgenden Kapitel werden die Kompositionstechniken vier verschiedener Werke und deren Bedeutsamkeit für den Fortschritt elektronischer Musik untersucht. Dabei wird bei Morton Subotnicks „Silver Apples of the Moon“ und Wendy Carlos „Switched on Bach“ insbesondere der Einfluss auf den Erfolg der beiden im vorigen Kapitel vorgestellten Modular-Systeme (von Moog und Buchla) beleuchtet. Bei Pink Floyds „On the Run“ und Olivier Messiaens „Modi mit begrenzten Transpositions-möglichkeiten“ wird die klangliche und musikalische Struktur näher untersucht, um als kompositorische Grundlage wichtige Designrichtlinien für die Entwicklung des Prototyps zu liefern.

4.1. Wendy Carlos – Switched on Bach

Seine große Popularität verdankte der erste Moog Synthesizer vor allem dem Erfolg der Aufnahme „Switched on Bach“ von Wendy Carlos.⁶⁶ Bei der 1968 erschienenen Aufnahme, interpretiert die als Walter Carlos geborene, amerikanische Musikerin Werke von Bach mit Moogs Modular-System neu und kombinierte damit die Innovation neuartiger elektronischer Musik mit den musikalischen Strukturen klassischer Kompositionen.⁶⁷ Dieser Mix erreichte ein breites Publikum:

Bei Klassikliebhaber:innen, Pop Hörer:innen und Fans elektronischer Musik zugleich beliebt, erreichte Switched on Bach als erste Klassikplatte jemals den Platinstatus.⁶⁸ Wendy Carlos verstand den von Robert Moog entwickelten Synthesizer als Werkzeug, um bekanntes neu erklingen zu lassen:

“I thought that if I offered people a little bit of traditional music, and they could clearly hear the melody, harmony, rhythm and all the older values, they’d finally see that this was really a pretty neat new medium.”⁶⁹ (Wendy Carlos)

⁶⁶ Rumsey, „Modular Synths and Embedded Computing“, 234.

⁶⁷ Pinch und Trocco, *Analog Days*, 149.

⁶⁸ Pinch und Trocco, 132.

⁶⁹ Pinch und Trocco, 134.

Nach dem großen Erfolg von *Switched on Bach* stand Carlos im regelmäßigen Austausch mit Moog und beeinflusste maßgeblich die weitere Entwicklung seiner Synthesizer.⁷⁰

4.2. Morton Subotnick – *Silver Apples of the Moon*

Morton Subotnick war einer der drei Komponisten, die Donald Buchla 1965 im San Francisco Tape Center bei der Konzeption seines ersten Modular-Systems unterstützten (s. [Robert Moog und Donald Buchla – Eastside vs. Westside](#)). 1967 erhielt er von der Plattenfirma Nonesuch Records als erster Komponist jemals den Auftrag ein Album mit rein elektronischer Musik aufzunehmen.⁷¹

Auch wenn das infolgedessen entstandene Album „*Silver Apples of the Moon*“ nicht die gleiche Popularität wie Wendy Carlos „*Switched on Bach*“ erreichen konnte, gilt es heutzutage als Meilenstein in der Entwicklung elektronischer Musik.⁷²

Der neuartige Klang elektronischer Musik, den Subotnick mit dem Buchla Modular-System in seinem Album darbot wird in Marc Prendergasts Buch „*The Ambient Century – From Mahler to Moby*“ beschrieben:

„*Blips and Burps seem to roll around a giant aural blob, pushed forward by some manic alien hand*“⁷³

Dieses bildhaft beschriebene Hörerlebnis entsteht vermutlich durch die zentrale Rolle des Sequenzers in der Buchla 100 Serie, den Subotnick für seine Komposition verwendetet:

„*Sequencing now is something anyone can do anytime, anywhere. But this was the first time that you could do it in this way. He certainly made*

⁷⁰ Pinch und Trocco, 136.

⁷¹ „Morton Subotnick on 50 Years of *Silver Apples of the Moon*“, Fact Magazine, 13. Juli 2017, <https://www.factmag.com/2017/07/13/morton-subotnick-silver-apples-of-the-moon/>.

⁷² „Morton Subotnick on 50 Years of *Silver Apples of the Moon*“.

⁷³ Mark Prendergast, *The Ambient Century: From Mahler to Moby - the Evolution of Sound in the Electronic Age*, New ed (London, 2003), 69.

a great use of it and advised [Buchla] in that direction, I'm sure.”⁷⁴
(Morton Subotnick)

4.3. Pink Floyd – On the Run

Das 1973 veröffentlichte Album “Dark Side of the Moon” von Pink Floyd gilt als eines der einflussreichsten Alben für die weitere Entwicklung populärer Musik.⁷⁵ Es war zwischen 1973 und 1988 insgesamt 741 Wochen in den Billboard Top 100 Charts und bot für die damalige Zeit revolutionäre Synthesizer Klänge.⁷⁶

Insbesondere die Sequencer Melodie des 3. Titels der Platte – „On The Run“ – prägte den Sound elektronischer Musik und wird mitunter als „ikonisch“ für modernen Synthesizer Sound bezeichnet.⁷⁷

In der Zeit, in der die britische Band in den Abbey Road Studios an den Aufnahmen ihres achten Studioalbums arbeiteten, stand ihnen wegen einer weiteren Produktion im Tonstudio einer der ersten käuflichen Sequenzer/Synthesizer zu Verfügung: Der „Synthi AKS“ von der Firma EMS⁷⁸. Pink Floyds Bandmitglied Roger Waters spielte auf diesem Synthesizer eine Sequenz zu einem Song ein, den die Band ursprünglich bei ihren Konzerten als „Gitarrenjam“ gespielt hatten:

“We'd be playing it live that way for quite some time as a sort of guitar jam [...] I think we were... none of us were that happy with it as a piece”⁷⁹ (David Gilmour)

Aus dieser Sequenz entwickelte sich schließlich das ganze Instrumentalstück „On The Run“. Im 2003 erschienenen Dokumentarfilm „The Making of The Dark Side Of The

⁷⁴ Thom Holmes, *Electronic and Experimental Music: Pioneers in Technology and Composition*, 2. Aufl. (London: Psychology Press, 2002), 172.

⁷⁵ Russel Reising, „Speak to Me“: *The Legacy of Pink Floyd's The Dark Side of the Moon* (Ashgate Publishing Company, 2005), 13.

⁷⁶ „Gear Rundown: Pink Floyd's Dark Side Of The Moon“, *Mixdown Magazine* (blog), 2. März 2021, <https://mixdownmag.com.au/features/gear-rundown-pink-floyds-dark-side-of-the-moon/>.

⁷⁷ SYNTHHEAD, „The Iconic Sounds Of Synthesis – ‘On The Run’ From ‘Dark Side of the Moon’“, *Synthtopia* (blog), 7. Dezember 2013, <https://www.synthtopia.com/content/2013/12/07/the-iconic-sounds-of-synthesis-on-the-run-from-dark-side-of-the-moon/>.

⁷⁸ Matthew Longfellow, *The Making of The Dark Side Of The Moon*, 2003, 13:00-14:00.

⁷⁹ Longfellow, Abschn. 12:00-13:00.

Moon" sind die Originalaufnahmen zu sehen, in denen Wright den Synthie AKS bedient, sowie eine Erklärung der technischen Vorgehensweise seitens Wright:

"A series of notes, played in slowly, triggering a noise generator and oscillators and then we just sped it up"⁸⁰ (Roger Waters)

Beobachtungen aus dem Film lassen darauf schließen, dass neben den von Wright erwähnten Oszillatoren und dem Noise Generator, in der Aufnahme noch der Filter, Envelope und ein Federhall des Synthi AKS bespielt werden.⁸¹ Ein Vergleich der auf den Filmaufnahmen vorgenommenen Einstellungen am Synthie AKS und den Angaben des Herstellers in dessen Handbuch unterstützt diese Schlussfolgerung.⁸²

4.4. Olivier Messiaen – Modi mit begrenzter Transponierbarkeit

Der 1908 in Frankreich geborene Komponist Olivier Messiaen begann bereits lange vor der Entwicklung der ersten Modular-Systeme sich näher mit der Zusammensetzung einzelner Musiknoten auseinander zu setzen. Seiner Arbeit wird nachgesagt, das „Zeitalter des Klangs“ geformt zu haben.⁸³ Als gemeinsamer Lehrer und Inspirationsquelle vieler bedeutender Pioniere elektronischer Musik, wie Karl Heinz Stockhausen, Iannis Xenakis und Pierre Boulez, prägte Messiaen nachhaltig die Entwicklung elektronischer Musik.⁸⁴

Mit seinen 1944 entstandenen „Modi mit begrenzter Transpositionsmöglichkeit“, führte er eine Reihe von 7 gleichstufigen oder periodisch alternierenden Intervallketten als Skalenmaterial für distanzharmonische Läufe ein.⁸⁵ Es folgt eine Übersicht der von Messiaen postulierten Modi, als Folge von Intervallen dargestellt, wobei 1 kleine Sekund, 2 große Sekund, 3 kleine Terz, 4 große Terz bedeuten:

„1. Modus: 2 2 2 2 2 2

2. Modus: 1 2 1 2 1 2 1 2

⁸⁰ Longfellow, Abschn. 14:00-14:11.

⁸¹ Longfellow, Abschn. 12:00-16:00.

⁸² EMS, „EMS-Synthi-AKS Service Manual“, 12. September 1970.

⁸³ Prendergast, *The Ambient Century*, 38.

⁸⁴ Prendergast, 38.

⁸⁵ Zsolt Gárdonyi und Hubert Nordhoff, *Harmonik* (Möseler Verlag, 2002), 219.

3. Modus: 2 1 1 2 1 1 2 1 1

4. Modus: 1 1 3 1 1 1 3 1

5. Modus: 1 4 1 1 4 1

6. Modus: 2 2 1 1 2 2 1 1

7. Modus: 1 1 1 2 1 1 1 1 2 1⁸⁶

⁸⁶ Olivier Messiaen, *Technique de mon langage musical*, 1944, 59–61.

5. Konzeption

In diesem Abschnitt soll nun die Planungs- bzw. Vorproduktionsphase des Prototyps behandelt werden. Grundlage für die folgenden Designrichtlinien zur Mechanik und Technik sind dabei die im Kapitel [Recherche](#) vorgestellten Ergebnisse. Die Anforderungen an das Sounddesign und der sich daraus ergebende Funktionsumfang des Prototyps ergeben sich überwiegend aus dem Kapitel [Zeitgenössische Kompositionstechniken](#).

5.1. Design

5.1.1. Sounddesign / Funktionsumfang

Einen Leitfaden für die Überlegungen bezüglich des Sounddesigns, bot die Entscheidung das im Kapitel [Zeitgenössische Kompositionstechniken](#) vorgestellte Instrumentalstück „On the run“ von Pink Floyd als kompositorische Grundlage meines Prototyps zu wählen: Der fertige Prototyp soll alle Funktionalitäten liefern, um die berühmte Sequenz nachbilden zu können, die dem Song zu Grunde liegt. Die Recherche des damals verwendeten Synthi AKS zeigte, dass dazu zwei Oszillatoren, ein Tiefpassfilter, ein Envelope und ein Sequenzer notwendig sind⁸⁷ (s. [Pink Floyd – On the Run](#)). Diese Komponenten der Klangsynthese sollen daher im Funktionsumfang des Prototyps enthalten sein.

Um die modulare Denkweise in Eurorack-Systemen zu beachten, sollen diese Komponenten als einzelne Module verstanden werden und der Signalfluss nicht vorgegeben, sondern frei patchbar (s. [Patchen](#)) angelegt sein.

Wie in der [Einleitung](#) dieser Arbeit erklärt, soll der Prototyp insbesondere Neueinsteiger:innen in das Eurorack-System ansprechen, die mit der Vielfalt der Patchmöglichkeiten überfordert sein könnten. Daher soll mit einem Schalter die gesamte Modularität des Prototyps abgeschaltet werden können und ein vorgegebener Signalfluss aktiviert werden.

⁸⁷ Longfellow, *The Making of The Dark Side Of The Moon*, Abschn. 12:00-16:00.

Konzeption

Eine weitere grundlegende Entscheidung war es, die in Abschnitt [Robert Moog und Donald Buchla – Eastcoast vs. Westcoast](#) vorgestellten Ansätze zu Syntheseart und Interface zu kombinieren. Der für den Pink Floyd Sound benötigte Tiefpassfilter und Envelope legten nahe, für die Auswahl der Syntheseart und allgemeinen Funktionsgruppen des Prototyps den Ansatz von Robert Moog zu verfolgen und die subtraktive Synthese zu wählen. Im Gegenzug soll typisch für Donald Buchlas Westcoast Style ein Sequenzer mit 8 einstellbaren Schritten als Hauptinterface dienen. Der zentrale Gedanke hinter dieser Entscheidung war, dass so ein umfassenderes Bild der Prinzipien moderner Modular-Systeme abgebildet wird und der Prototyp damit einen besseren Einstieg in sowohl den von Moog begründeten „East Coast Approach“⁸⁸ als auch Donald Buchlas Westcoast Ansatz⁸⁹ bieten könnte.

Des Weiteren ist ein zweiter Sequenzer geplant, dessen Parameter jedoch nicht durch Regler einstellbar sind. In diesem Sequenzer sind die Werte der einzelnen Schritte in einer Reihe von Vorlagen abgespeichert und können als Steuerspannung abgegriffen werden. Moduliert man mit dieser Steuerspannung die Frequenz der beiden Oszillatoren erhält man je nach gewählter Vorlage die Melodie von „On The Run“, oder die ersten acht Töne der Intervallketten von Messiaens „Modi mit begrenzter Transponierbarkeit“ (s. [Olivier Messiaen – Modi mit begrenzter Transponierbarkeit](#)). Die Wahl der Modi von Messiaen als einzelne Sequenzer Vorlagen hat dabei folgenden Hintergrund: Messiaen begründete zwar deutlich vor der Entwicklung der ersten Modular-Systeme seine Modi mit begrenzter Transponierbarkeit, doch seine Rolle als Pionier in der Entwicklung elektronischer Musik und als Inspirationsquelle für folgende Pioniere (s.o.) schlägt eine Brücke zum technologisch-/historischen Einstieg in die Welt modularer Synthesizer. Daher könnte die Verwendung der sieben von ihm postulierten Modi einen geeigneten inhaltlichen Einstieg in diese Welt bieten.

⁸⁸ Rumsey, „Modular Synths and Embedded Computing“, 234.

⁸⁹ Rumsey, 234.

5.1.2. Mechanisches Konzept

Die Planung des mechanischen Aufbaus des Prototyps war durch die genauen Richtlinien des Eurorack-Systems (s. [Mechanische Vorgaben](#)) sehr klar vorgegeben. Alle Bedienelemente müssen auf einer gemeinsamen Frontplatte mit einer Höhe von 3HE montiert werden, damit der Prototyp in ein Gehäuse im Eurorack-Format eingebaut werden kann. Die genaue Breite der Frontplatte ergibt sich durch das visuelle Konzept, muss aber zur Einhaltung des Eurorack-Standards einem Vielfachen einer TE entsprechen. Weitere Elektronik, wie etwa integrierte Schaltkreise (ICs) soll auf einer Platine verlötet werden können, welche mit Abstandshaltern an die Rückseite der Frontplatte verschraubt wird. Auf diese Platine soll außerdem das Belaboard (s. [Belaboard](#)) aufgesteckt werden.

Alle Verbindungen zwischen den Bedienelementen und der Platine sollen steckbar sein. Außerdem werden benötigte ICs nicht direkt auf die Platine gelötet, sondern auf verlötzte Sockel aufgesteckt. Dies soll den Zusammenbau des Prototyps sowie eventuell anfallende Reparaturen erleichtern.

5.1.3. Visuelles Konzept

Die Beschriftungen und das Design der Frontplatte sollen bedienenden Personen die Funktionsweise des Prototyps visuell greifbar machen. Zusammenhänge verschiedener Bedien- und Patchmöglichkeiten müssen daher klar erkennbar sein. Die Bezeichnungen der verstellbaren Parameter sollen dabei möglichst verständlich sein, aber dennoch dem üblichen Vokabular der Modular-Szene entsprechen, damit sich am Prototyp gelernte Bedienmuster auch auf andere Module im Eurorack-Format übertragen lassen.

Alle einstellbaren Parameter sollen direkt auf der Frontplatte verfügbar sein und dürfen nicht nur durch etwaige Tastenkombinationen oder Menüführungen erreicht werden können.

Die modulare Denkweise der einzelnen Komponenten des Prototyps muss visuell abgebildet sein. Das bedeutet, dass alle Parameter einer Komponente auf der Frontplatte räumlich nah beieinander angeordnet werden. Die einzelnen Komponenten des Prototyps werden umrandet und beschriftet.

Zur besseren Verständlichkeit wird ggf. die Beschriftung einzelner Parameter mit beispielhaften Piktogrammen ergänzt.

5.2. Technik

Nachdem im vorherigen Kapitel der geplante Funktionsumfang des Prototyps definiert wurde, sollen nun die daraus folgenden technischen Anforderungen zur Realisierung der Funktionalitäten zusammengefasst werden. Anschließend werden verschiedene Möglichkeiten zur Umsetzung dieser Anforderungen besprochen. Zuletzt wird die gewählte Lösung näher betrachtet.

5.2.1. Anforderungen

Wie in Kapitel [Analog / Digital](#) beschrieben, ist in Zukunft wegen der großen Vorteile digitaler Systeme gegenüber analoger Technik ein weiteres Wachstum digitaler Eurorack Module auf dem Markt zu erwarten. Der vorgestellte Prototyp sollte daher in der technischen Umsetzung auf einem solchen digitalen System basieren. Dafür muss als Hardware ein passender Microcontroller oder ein Embedded Computing System (s. [Microcontroller / Embedded Computing](#)) gewählt werden, welcher/s folgende Anforderungen erfüllt:

- **Hohe Anzahl von In- und Outputs**

Da beim fertigen Prototyp der Signalfluss frei patchbar sein soll, werden viele Ein- und Ausgänge für Steuerspannungen benötigt. Außerdem sollen möglichst viele Parameter der Audiosynthese über Potentiometer eingestellt werden können, wofür weitere Eingänge benötigt werden.

- **Hochwertige A/D und D/A Wandler**

Für das Ausgeben und Einlesen der Steuerspannungen werden analog zu digital bzw. digital zu analog Wandler benötigt. Um den potenziell kurvigen Verlauf der Steuerspannungen dabei möglichst stufenlos ausgeben oder einlesen zu können werden daher hochwertige Wandler benötigt. Damit auch Audiosignale gepatcht werden können, sollten die Wandler in der Lage sein Audiosignale mit einer Abtastfrequenz von mindestens 44,1 kHz und einer Bittiefe von 16 Bit oder höher zu wandeln. Die Werte der Abtastfrequenz und der Bittiefe ergeben sich hierbei aus dem CD-Standard.⁹⁰

⁹⁰ Thomas Görne, *Tontechnik: Hören, Schallwandler, Impulsantwort und Faltung, digitale Signale, Mehrkanaltechnik, tontechnische Praxis; mit 33 Tabellen*, 4., aktualisierte Aufl (München: Hanser, 2015), 335.

Konzeption

- **Niedrige Kosten**

Der Prototyp richtet sich in erster Linie an Personen, die einen einfachen Einstieg in das Eurorack-System suchen und noch nicht bereit sind viel Geld zu investieren. Deshalb sollte sich das fertige Produkt möglichst im unteren Preissegment ansiedeln. Verglichen mit den restlichen Bauteilen des Prototyps (Potentiometer, Taster, etc.) wird der Microcontroller oder das gewählte Embedded Computing System das teuerste Bauteil im Gerät werden. Daher sollte bei der Auswahl auf niedrige Kosten geachtet werden, um die Gesamtkosten des Prototyps möglichst gering halten zu können.

- **Unterstützung gängiger Programmiersprachen**

Der Microcontroller sollte im Optimalfall eine Programmierung mit Pure Data oder Max/MSP zulassen. Diese beiden grafischen Programmiersprachen sind für die Programmierung von Audioanwendungen entwickelt worden⁹¹ und eignen sich demnach besonders für die Programmierung des Prototyps.

5.2.2. Microcontroller / Embedded Computing System

Wie im obigen Kapitel erläutert galt es eine Wahl zwischen einem Microcontroller oder einem Embedded Computing System als zentrale Hardware des Prototyps zu treffen. Ein Microcontroller ist ein Halbleiterchip auf dem sowohl Prozessor und Peripherie zur Ansteuerung von Ein- und Ausgängen, als auch Arbeits- und Programmspeicher verbaut sind.⁹² Embedded Computing Systeme, folgend mit der Kurzform Embedded Computer bezeichnet, bieten darüber hinaus alle Funktionalitäten eines „General-Purpose-Operating-System“ wie Linux. Für eine nähere Beschreibung eines solchen Systems sei auf „*A Linux-based Real-Time Operating System*“ von Michael Barabanov verwiesen.⁹³

Microcontroller wie der Arduino Due⁹⁴ oder Teensy 4.0⁹⁵ garantieren durch ihre Arbeitsweise exaktes Timing und ermöglichen eine einfache Anbindung von Sensoren

⁹¹ Miller Puckette, „Pure Data“, *ICMC*, 1977, 1.

⁹² A.K. Singh, *Microcontroller and embedded system* (New Age International, 2008), 3.

⁹³ Michael Barabanov, „*A Linux-Based Real-Time Operating System*“, 1. Juli 1997, 43.

⁹⁴ „Arduino Due“, Arduino Official Store, zugegriffen 8. Januar 2022, <http://store.arduino.cc/products/arduino-due>.

⁹⁵ „Teensy® 4.0“, zugegriffen 8. Januar 2022, <https://www.pjrc.com/store/teensy40.html>.

Konzeption

oder Potentiometern.⁹⁶ Das macht die Berechnung der Audiosignale in Hard Realtime (zeitliche Fristen in der Berechnung werden garantiert eingehalten) möglich. Im Vergleich zu Embedded Computern sind Microcontroller günstiger in der Anschaffung. Allerdings ist ihre Rechenleistung begrenzt und bekannte Programmierumgebungen wie Pure Data und Max/MSP werden in der Regel nicht unterstützt.

Embedded Computer wie der Raspberry Pi⁹⁷ oder BeagleBone Black⁹⁸ ermöglichen dem/der User:in im Gegenzug Audio Processing auf die gleiche Art und Weise zu behandeln, wie mit einem vollwertigen Computer. Die Nutzung von den bereits erwähnten grafischen Programmiersprachen, Netzwerkanbindung und lokale Dateiverwaltung erleichtern die Entwicklung von Prototypen im Musik-Technologie Bereich enorm.⁹⁹ Allerdings sind Embedded Computer darauf ausgelegt mehrere Prozesse gleichzeitig zu verarbeiten und garantieren dabei nur Soft Realtime (zeitliche Fristen in der Berechnung von Audiosignalen werden i.d.R. eingehalten, die Einhaltung wird jedoch nicht garantiert).¹⁰⁰

Das im folgenden Unterkapitel näher betrachtete „Belaboard“ bietet eine Kombination aus den Vorteilen beider Systeme und soll daher als Grundlage des Prototyps gewählt werden.

5.2.3. Belaboard

Das von Andrew McPherson und Victor Zappi entwickelte Belaboard, ist ein Aufsatz für den BeagleBone Black Embedded Computer.¹⁰¹ Es wurde speziell dafür entwickelt, die Vorteile von Microcontrollern bezüglich Hard Realtime Audio Processing mit den oben erwähnten Vorteilen eines Embedded Computer zu kombinieren.¹⁰² Dazu verwendet es die Linux Kernel Extension „Xenomai“, die es ermöglicht das

⁹⁶ Andrew P McPherson und Victor Zappi, „An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black“, 2015, 2.

⁹⁷ Raspberry Pi Ltd, „Buy a Raspberry Pi 4 Model B“, Raspberry Pi, zugegriffen 8. Januar 2022, <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.

⁹⁸ „BeagleBoard.org - black“, zugegriffen 8. Januar 2022, <https://beagleboard.org/black>.

⁹⁹ Romain Michon u. a., „Embedded Real-Time Audio Signal Processing With Faust“, 2020, 2.

¹⁰⁰ McPherson und Zappi, „An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black“, 2.

¹⁰¹ McPherson und Zappi, 1.

¹⁰² McPherson und Zappi, 2.

Konzeption

Audioprocessing beim BeagleBone Black mit höherer Priorität als Prozesse des Linux Kernels selbst laufen zu lassen.¹⁰³

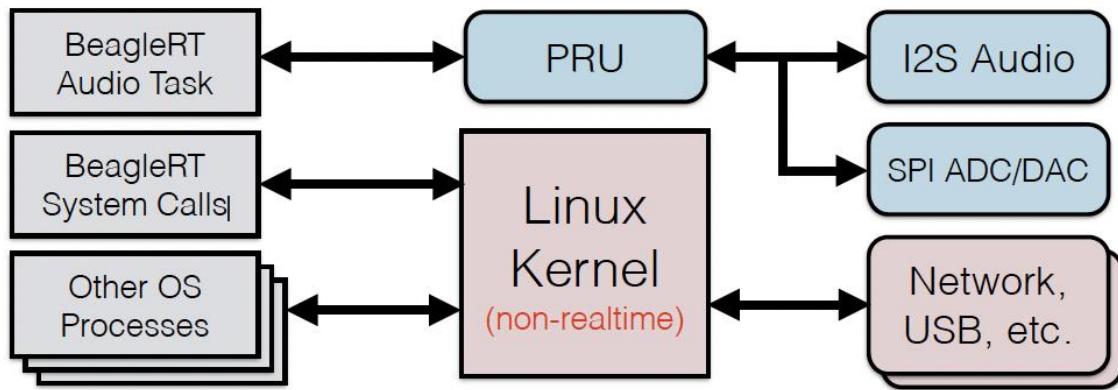


Abbildung 3: Prozessstruktur des Beagleboards¹⁰⁴

Weitere Eigenschaften des Boards fassen McPherson und Zappi in einem Paper der Audio Engineering Society von 2015 zusammen:

1. *Simultaneous stereo audio and multichannel sensor data capture*
2. *Ultra-low latency, less than 1ms round trip*
3. *High sensor bandwidth with no bottleneck between sensor and audio processing*
4. *Jitter-free synchronisation of audio and sensor data*
5. *Robust performance under load; no buffer underruns due to unrelated processes*
6. *Lightweight C-based API*
7. *Self-contained platform suitable for inclusion inside a digital musical instrument*¹⁰⁵

¹⁰³ McPherson und Zappi, 4.

¹⁰⁴ McPherson und Zappi, 4.

¹⁰⁵ McPherson und Zappi, 3.

Konzeption

Das Board unterstützt außerdem die Verwendung der Programmierumgebung Pure Data.¹⁰⁶ Somit erfüllt das Board alle im Kapitel [Anforderungen](#) angesprochenen Voraussetzungen, um als Herzstück des Prototyps verwendet zu werden. Lediglich der hohe Preis des Boards (159,⁰⁰ £ / 190,¹⁶ €) ist ein erwähnenswerter Nachteil dieser Lösung.¹⁰⁷ Hinzu kommen hohe Importkosten und eine lange Lieferzeit, da das Produkt aus dem Vereinigten Königreich importiert werden muss.

¹⁰⁶ Michon u. a., „Embedded Real-Time Audio Signal Processing With Faust“, 1.

¹⁰⁷ „Bela Starter Kit – eu.shop.bela.io“, zugegriffen 8. Januar 2022,
<https://eu.shop.bela.io/products/bela-starter-kit>.

6. Realisierung Phase 1

Die Realisierung des Prototyps teilt sich in zwei Phasen auf und orientiert sich am „User Centered Design“ Ansatz, der vorgibt die spätere Zielgruppe des Produktes frühzeitig in den Designprozess einzubinden.¹⁰⁸ Zunächst wird ein erster vollständig funktionsfähiger Prototyp entwickelt und gebaut. Anschließend werden Design und Funktionalität des Prototyps evaluiert (s. [Evaluierung](#)) und Verbesserungsvorschläge gesammelt. Auf Basis dieser Evaluation soll schließlich ein zweiter Prototyp entwickelt werden.

6.1. Software

6.1.1. Programmierumgebung / Wahl der Programmiersprache

Das Belaboard wird über eine eigene integrierte Entwicklungsumgebung (IDE) programmiert, die über den Browser erreichbar ist. Dazu muss das Board per USB mit einem Computer verbunden und im Browser die Adresse <https://bela.local> aufgerufen werden. In dieser IDE kann das Belaboard über einen integrierten Texteditor mit den Programmiersprachen C++, SuperCollider, Javascript und Csound programmiert werden.¹⁰⁹ Pure Data Patches werden im Editor visualisiert, müssen aber auf dem Computer selbst mit der Pure Data Anwendung erstellt werden.¹¹⁰ Neben dem Editor verfügt die IDE noch über ein Oszilloskop, mit dessen Hilfe die eigenen, programmierten Signale im Browser visuell dargestellt werden können.¹¹¹ Dieses Oszilloskop erwies sich bei der Realisierung des Prototyps als äußerst wichtiges Hilfsmittel zur Fehlersuche. Weiterhin ermöglicht die IDE die Programmierung einer im Browser darstellbaren graphischen Benutzeroberfläche (GUI) zur Steuerung von programmierten Parametern.¹¹² Wichtige Einstellungen des Belaboard, wie die Abtastrate der analogen und digitalen Eingänge oder die Blockgröße der Audiosamples, die pro Programmzyklus berechnet werden, können in der rechten

¹⁰⁸ Chadia Abras, Diane Maloney-Krichmar, und Jenny Preece, „User-Centered-Design“, 2004, 15.

¹⁰⁹ „Meet the IDE - The Bela Knowledge Base“, zugegriffen 10. Januar 2022, <https://learn.bela.io/the-ide/meet-the-ide/>.

¹¹⁰ „Meet the IDE - The Bela Knowledge Base“.

¹¹¹ „The oscilloscope - The Bela Knowledge Base“, zugegriffen 10. Januar 2022, <https://learn.bela.io/the-ide/the-oscilloscope>.

¹¹² „Crafting GUIs - The Bela Knowledge Base“, zugegriffen 10. Januar 2022, <https://learn.bela.io/the-ide/crafting-guis/>.

Realisierung Phase 1

Spalte der IDE angepasst werden. Hier lassen sich auch Dateien verwalten, vorprogrammierte Codebeispiele auswählen und eine interaktive Übersicht der In- und Outputs des Belaboard anzeigen.¹¹³

Für die Programmierung des Prototyps wurden zunächst die Programmiersprachen C++ für die Verarbeitung der Ein- und Ausgänge und Pure Data für die Audiosynthese verwendet. In einem zweiten Schritt wurde schließlich die gesamte Software mit der Programmiersprache C++ realisiert. In den folgenden Abschnitten wird dies näher erläutert.

6.1.2. Pure Data auf dem Belaboard

Pure Data Patches werden auf dem Belaboard mit Hilfe der „libpd“ Programmierschnittstelle (API) ausgeführt.¹¹⁴ Über diese API wird Pure Data als eine Digital Sound Processing (DSP) Library in der Programmiersprache C in die restliche Software auf dem Belaboard eingebettet.¹¹⁵ Um die Hardware des Belaboard in Pure Data nutzen zu können, wird zusätzlich zum programmierten Pure Data Patch C++ Code benötigt, der als sogenannter „Wrapper“ für die libpd Library dient.¹¹⁶ Als Wrapper bezeichnet man Programmcode, der einen anderen Teil der Software umhüllt und somit die Integration von Programmteilen in einer anderen Programmiersprache ermöglicht.¹¹⁷ Ein solcher Wrapper ist auf dem Belaboard bereitgestellt und wird automatisch in das Projekt eingebunden, sobald man einen Pure Data Patch mit dem Namen „_main.pd“ auf das Board lädt.

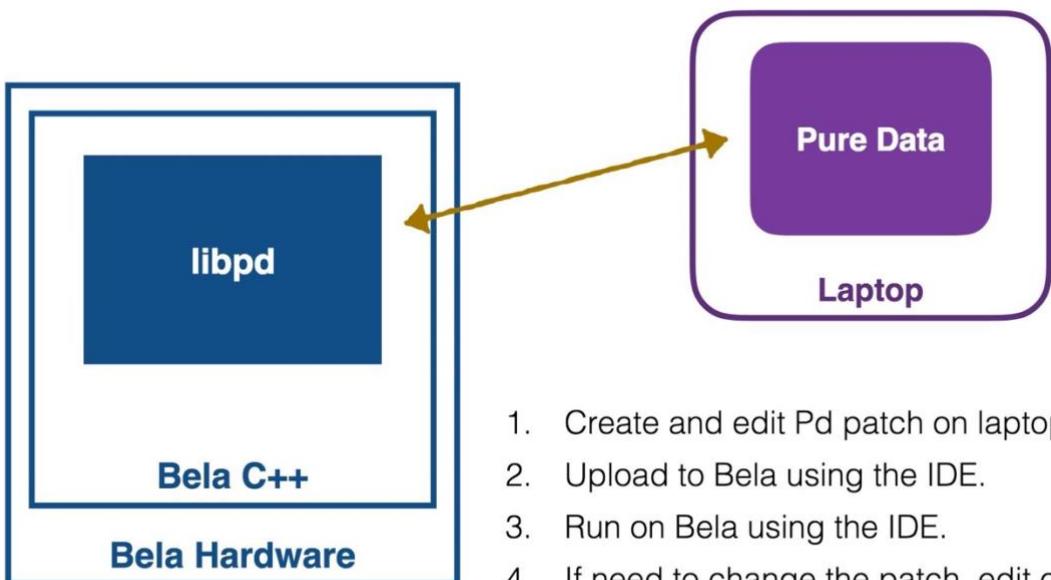
¹¹³ „Meet the IDE - The Bela Knowledge Base“.

¹¹⁴ „Combining Pure Data and C++ - The Bela Knowledge Base“, zugegriffen 10. Januar 2022, <https://learn.bela.io/tutorials/pure-data/advanced/custom-render/>.

¹¹⁵ Peter Brinkmann u. a., „Embedding Pure Data with Libpd“, 2011, 1.

¹¹⁶ „Combining Pure Data and C++ - The Bela Knowledge Base“.

¹¹⁷ Harry M Sneed, „Encapsulation of Legacy Software: A Technique for Reusing Legacy Software Components“, *Annals of Software Engineering* 9 (2000): 295–97.



1. Create and edit Pd patch on laptop.
2. Upload to Bela using the IDE.
3. Run on Bela using the IDE.
4. If need to change the patch, edit on laptop and repeat process.

Abbildung 4: Pure Data Integration beim Belaboard¹¹⁸

Dieser Wrapper ermöglicht es die digitalen Inputs und des Belabards in Pure Data mit Reciever Objekten einzulesen und mit Sender Objekten die digitalen Outputs des Boards anzusteuern. Analoge Ein- und Ausgänge werden mit den Objekten „adc~“ und „dac~“ angesteuert, wobei adc~ 1 und 2 den beiden Audioeingängen und dac~ 1 und 2 dem Stereo-Audioausgang entsprechen.¹¹⁹ Mehr Details zur Verwendung von Pure Data folgen im Kapitel [Audiosynthese](#).

Es ist möglich, den von Bela bereitgestellten Standard Wrapper Code zu modifizieren, um einzelne Programmabschnitte in C++ realisieren zu können. Dies ist insbesondere dann sinnvoll, wenn exaktes Timing beim Einlesen und der Ausgabe der digitalen Pins sehr wichtig ist, oder über die In- und Outputs des Belabards andere Bauteile mit zeitkritischen Kommunikationsprotokollen wie I2C verbunden werden sollen.¹²⁰ Da der Prototyp mit sehr vielen Ein- und Ausgängen arbeitet, und die zeitliche Abfolge der Ausgabe von Steuerspannungen und dem Einlesen der Inputs von Bedeutsamkeit ist, wird die Steuerung der In- und Outputs in einem solchen modifizierten Wrapper

¹¹⁸ „Pure Data - The Bela Knowledge Base“, zugegriffen 10. Januar 2022, <https://learn.bela.io/using-bela/languages/pure-data/>.

¹¹⁹ „Pure Data - The Bela Knowledge Base“.

¹²⁰ „Combining Pure Data and C++ - The Bela Knowledge Base“.

Realisierung Phase 1

realisiert. Es folgt eine kurze Übersicht über die wichtigsten drei Programmabschnitte des Wrappers:

setup():

In der Setup Funktion des Belabards wird das Übertragen von Daten zwischen dem C++ Code und dem Pure Data Patch vorbereitet, indem hier die benötigten Receiver und Sender angelegt werden. Außerdem werden die digitalen General Purpose Inputs/Outputs (GPIOs) zu Ein- oder Ausgängen definiert und das bereits erwähnte Oszilloskop aktiviert. Die Setup Funktion wird beim Start des Belabards einmalig ausgeführt.

render():

In der Render Funktion werden die analogen und digitalen Eingänge eingelesen und die Werte verarbeitet. Der Audiobuffer wird vorbereitet und zusammen mit den eingelesenen Inputwerten an Pure Data geschickt. Nach dem Audioprocessing in Pure Data wird der Output Buffer von Pure Data in den Bela Output Buffer übertragen, die Steuerspannungen und weitere Outputsignale wie die Ansteuerung der LEDs werden ausgegeben und ausgewählte Signale werden an das integrierte Oszilloskop zur Anzeige weitergeleitet.¹²¹

Die Render Funktion wird für jeden neuen Sampleblock aufgerufen. Die Größe des zu verarbeitenden Blocks an Samples kann im „Settings“ Tab der Bela IDE eingestellt werden.¹²²

Beim Prototyp wurde die Blocksize auf 64 Samples festgelegt. Durch eine erneute Programmierung des Prototyps mit reinem C++ Code, konnte die Blocksize auf 16 Samples reduziert werden, ohne die CPU des Bela Boards zu überlasten (s. [Erneute Programmierung in C++](#)).

¹²¹ „Combining Pure Data and C++ - The Bela Knowledge Base“.

¹²² „C++ - The Bela Knowledge Base“, zugegriffen 10. Januar 2022, <https://learn.bela.io/using-bela/languages/c-plus-plus/>.

cleanup():

Zum Schluss des Programms gibt die Cleanup Funktion den im Setup belegten Programmspeicher auf dem Belaboard wieder frei. Sie wird einmalig beim Beenden des Programms ausgeführt.¹²³

Eine vollständige Erklärung des C++ Codes würde den Umfang dieser Bachelorarbeit überschreiten, daher wird sich im nächsten Kapitel auf die selbst implementierten Code-Abschnitte zum Input/Output Handling konzentriert. Der vollständige Code befindet sich im digitalen Anhang dieser Arbeit.

6.1.3. Input/Output Handling

Bei den Input- und Outputsignalen des Prototyps muss grundsätzlich zwischen digitalen Signalen und analogen Signalen unterschieden werden. Digitale Signale können nur zwei Zustände annehmen: *0* oder *1*, auch als *Low* und *High* oder *false* und *true* bezeichnet; 0V oder +3,3V. Spannungen über +3,3V an digitalen Eingängen können das Belaboard zerstören.

Analoge Signale können einen fließenden Spannungsverlauf haben. Die analogen Ausgänge können dabei Spannungen zwischen 0V und +5V erzeugen, die Eingänge können allerdings nur einen Spannungsbereich von 0V bis +4.096V erfassen. Spannungen in diesem Bereich werden als ein float Wert zwischen 0.0 und 1.0 ausgegeben. Bei Eingangsspannungen zwischen +4.096V und +5.0V an den analogen Pins wird das Belaboard zwar nicht beschädigt, das Signal wird jedoch unabhängig von der Höhe der Spannung als 1.0 bemessen.

Durch den unterschiedlichen Spannungsbereich der analogen Ein- und Ausgänge ergibt sich folgender wichtiger Zusammenhang: Eine „1.0“ bei einem analogen Eingang entspricht einer Eingangsspannung von +4.096V (oder höher). Eine „1.0“ an einem analogen Ausgang wird als +5V ausgegeben. Möchte man eine analoge Spannung an Pin X ausgeben, die an Pin Y als „1.0“ eingelesen werden soll muss man also in den analogen Ausgang folgenden Wert schreiben: $1 * (4.096 / 5.0)$

Sowohl analoge als auch digitale Signale werden beim Belaboard wie Audiosignale abgetastet (gesampled). Sie werden automatisch mit einer konstanten Abtastfrequenz

¹²³ „C++ - The Bela Knowledge Base“.

Realisierung Phase 1

(Samplerate) in einen Buffer geschrieben, sodass man wie bei Audiosignalen auf die analogen und digitalen Signale als Werte über einen zeitlichen Verlauf zugreifen kann. Bei den analogen Ein- und Ausgängen beträgt die Samplerate 22.05kHz. Durch das Multiplexer Capelet verringert sich die reelle Samplerate für jeden der 64 analogen Eingänge jedoch auf 2.75kHz (mehr dazu in [Erweiterungen zum Belaboard](#)). Die Samplerate der digitalen Ein- und Ausgänge entspricht der Audio-Samplerate von 44.1kHz.

Die drei Taster am Prototyp (Waveform A, Waveform B und Sequencer 2 Preset), sowie die beiden Gate Eingänge beim Envelope und Sequenzer Modul sind an digitale Eingänge des Belabards angeschlossen. Die beiden Gate Eingänge werden direkt in Pure Data über ein Reciever Objekt ausgelesen: `r bela_digitallIn16`

Die Taster müssen beim Einlesen digital entprellt werden, da sonst ein einmaliges Betätigen des Tasters im Schaltmoment als mehrfaches Wechseln von „HIGH“ zu „LOW“ registriert wird. Dazu wurde das Auslesen der Taster in Form von sogenannten „State Machines“ implementiert. Eine State Machine ist ein mathematisches Rechenmodell mit einer definierten Anzahl an Zuständen mit festen Übergangskriterien.

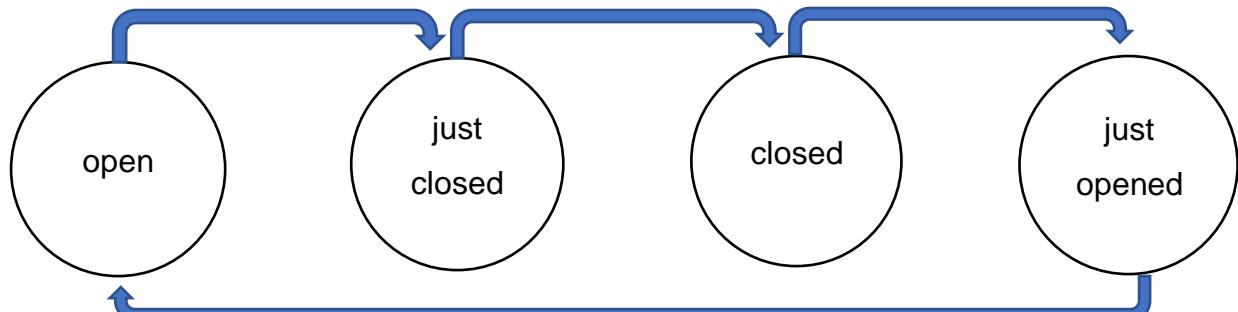


Abbildung 5: State Machine

Im Fall von Tastern gibt es vier Zustände: Open, just closed, closed und just opened. Wenn ein Taster gedrückt wird, liest der digitale Eingang an dem der Taster angeschlossen ist eine 0 aus und die State Machine geht vom Zustand „kStateOpen“ in den Zustand „kStateJustClosed“ über.

Realisierung Phase 1

```
buttonState[i] = digitalRead(context, n, kButtonPin[i]);  
if(gDebounceState[i] == kStateOpen)  
{      // Button is not pressed, could be pressed anytime  
      // Input: look for switch closure  
      if (buttonState[i] == 0)  
      {  
          gDebounceState[i] = kStateJustClosed;  
          gDebounceCounter[i] = 0;  
          ...  
      }  
}  
}
```

*Hier wird die gewünschte Auswirkung des Tasterdrucks eingefügt (zum Beispiel die Auswahl einer neuen Wellenform).

In diesem Zustand bleibt die State Machine nun eine vorgegebene Zeit und ignoriert das Prellen des Tasters. Erst nachdem das vorgegebene Zeitintervall (gDebounceInterval) überschritten wird, geht die State Machine in den Zustand „kStateClosed“ über.

```
else if(gDebounceState[i] == kStateJustClosed)  
{      // Button was just pressed, wait for debounce  
      // Input: run counter, wait for timeout  
      if(++gDebounceCounter[i] >= gDebounceInterval)  
      {  
          gDebounceState[i] = kStateClosed;  
      }  
}
```

Hier wird der digitale Eingang wieder ausgelesen und bei einem Wechsel von 0 zu 1 (Taster wurde losgelassen) geht die State Maschine in den letzten Zustand „kStateJustOpen“ über.

```
else if(gDebounceState[i] == kStateClosed)  
{      // Button is pressed, could be released anytime  
      // Input: look for switch opening  
      if (buttonState[i] != 0)  
      {  
          gDebounceState[i] = kStateJustOpen;  
          gDebounceCounter[i] = 0;  
      }  
}
```

Realisierung Phase 1

Nun wird erneut das Prellen des Tasters ignoriert und das „gDebounceInterval“ abgewartet, bevor die State Machine wieder in den ersten Zustand „kStateOpen“ wechselt.

```
else if(gDebounceState[i] == kStateJustOpen)
{
    // Button was just released, wait for debounce
    // Input: run counter, wait for timeout
    if(++gDebounceCounter[i] >= gDebounceInterval)
    {
        gDebounceState[i] = kStateOpen;
    }
}
```

Die Potentiometer am Prototyp, sowie alle Eingänge für Steuerspannungen, werden über die analogen Eingänge des Multiplexer Capelets (s. [Erweiterungen zum Belaboard](#)) mit der Funktion *multiplexerAnalogRead()* eingelesen und in das Array „inputVal“ geschrieben.

```
for(unsigned int input = 0; input < 3; input++)
{
    for(unsigned int muxChannel = 0; muxChannel < context->multiplexerChannels; muxChannel++)
    {
        inputVal[muxChannel + (8 * input)] = multiplexerAnalogRead(context, input, muxChannel);
    }
}
```

Da die Potentiometer an +3,3V angeschlossen sind, kann mit ihnen nur eine Spannung zwischen 0V und +3,3V eingestellt werden. Wie oben erwähnt interpretieren die analogen Eingänge Spannungen zwischen 0V und +4,096V als float Werte zwischen 0.0 und 1.0. Eine Eingangsspannung von +3.3V entspricht somit dem float Wert $3,3/4,096 \approx 0,805$.

Um mit den Potentiometern Werte zwischen 0.0 und 1.0 zu erreichen, müssen die Eingangswerte folglich mit dem Kehrbruch $4,096/3,3$ multipliziert werden. Anschließend wird mit zwei if-Bedingungen sichergestellt, dass die beiden Enden des Wertebereichs nicht durch etwaige Messungenauigkeiten überschritten werden können.

Realisierung Phase 1

```
for(unsigned int n = 0; n < 22; n++)
{
    potVal[n] = inputVal[n] * (4.096 / 3.3);
    if (potVal[n] < 0.00004)
    {
        potVal[n] = 0.0;
    }
    if (potVal[n] > 1.0)
    {
        potVal[n] = 1.0;
    }
}
```

Nachdem alle Eingangswerte erfolgreich eingelesen und in einen Wertebereich zwischen 0.0 und 1.0 skaliert wurden, werden sie mit der Funktion *libpd_float(RecieverName, Wert)* der libpd Schnittstelle übergeben.

```
libpd_float("Waveform1", Waveform1);
...
libpd_float("potVal21", potVal[21]);
```

Mit der Funktion *libpd_process_sys()* werden nun die neuen Eingangsdaten im Pure Data Patch verarbeitet und der nächste Block von Audiosamples sowie auszugebende Steuerspannungen berechnet und ausgegeben.

6.1.4. Audiosynthese

Im ersten Programmierdurchlauf wurde die Audiosynthese mit Pure Data realisiert. Bei der graphischen Programmiersprache erfolgt die Programmierung durch das Verbinden von Objekten, wodurch der angestrebte Signalablauf des Modular-Synthesizers gedanklich einfach auf den Programmablauf zu übertragen ist.

Die einzelnen Komponenten des Synthesizers sind im Pure Data Patch auf eigene Subpatches aufgeteilt. Subpatches sind Objekte, bei denen es sich um separate Patches handelt, die in die Objektboxen eingekapselt sind. Solche Subpatches können über **Inlets** und **Outlets** mit den Objekten im übergeordneten Patch verbunden werden.¹²⁴

¹²⁴ Miller Puckette, „Encapsulation in Pd“, 30. Dezember 2006,
<http://msp.ucsd.edu/techniques/v0.11/book-html/node67.html>.

Realisierung Phase 1

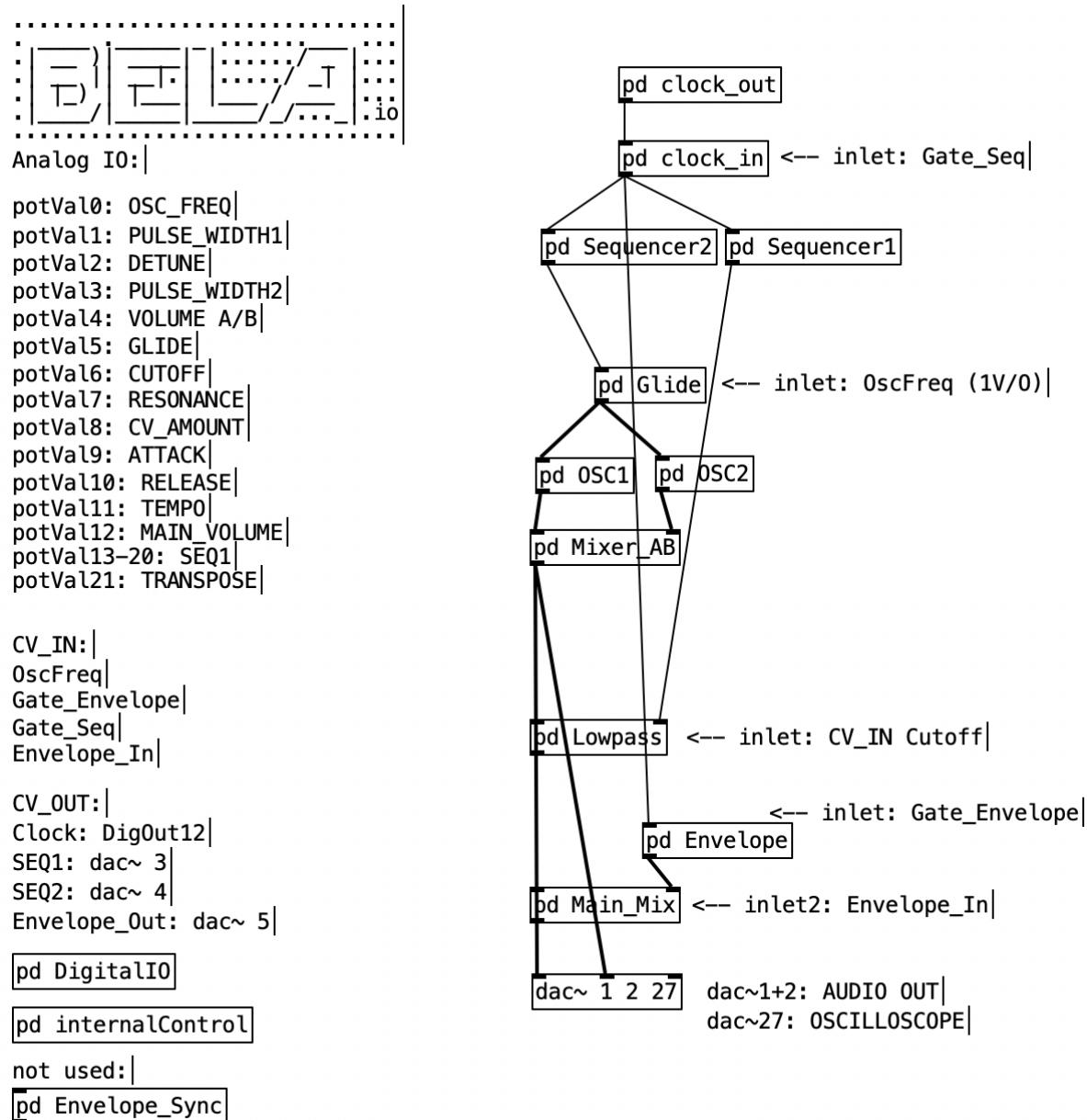


Abbildung 6: Pure Data Patch, Prototyp 1

Clock Out

Im Subpatch `Clock Out` ist das „Clock Modul“ des Prototyps realisiert. Ein `metro` Objekt erzeugt dabei in regelmäßigen Abständen ein „Bang“. Als „Bang“ wird in Pure Data ein Nachrichtentyp bezeichnet, der mit einem Mausklick oder der Betätigung eines Tasters verglichen werden kann.¹²⁵ Diese Nachricht betätigt dabei immer einen Schalter, der abwechselnd eine 1 und eine 0 darstellt und auf dem Clock Ausgang des Prototyps somit abwechselnd ein High und ein Low Signal erzeugt – das Taktsignal. Die

¹²⁵ Johannes Kreidler, „Programming Electronic Music in Pd“, zugegriffen 12. Januar 2022, <http://pd-tutorial.com/english/ch02s02.html>.

Realisierung Phase 1

Geschwindigkeit des `metro` Objekts wird mit dem Tempopotentiometer in der Einheit Beats per Minute (bpm) eingestellt.

Clock In, Sequencer 1 + 2

Die Subpatches `Clock In`, `Sequencer1` und `Sequencer2` entsprechen dem Sequencer Modul des Prototyps. Im `Clock In` Subpatch wird das eingehende Taktsignal genutzt um einen Zähler wiederholt von 0 -7 zählen zu lassen. Dieser Zähler lässt die beiden Sequencer in den jeweiligen Subpatches durchlaufen. In `Sequencer1` werden dabei die Werte der einzelnen Schritte durch die Potentiometer 1-8 bestimmt. In `Sequencer2` sind die Vorlage-Sequenzen abgespeichert. Der Taster „Sequencer 2 Preset“ bestimmt die Auswahl der Vorlage. Am Ende des `Sequencer2` Subpatches müssen die Werte vor der Ausgabe als Steuerspannung umgerechnet werden, um gemäß der V/Oct Skala (s. [V/Oct Skala](#)) bei Oszillatoren die passende Melodie zu erzeugen.

OSC 1 + 2, Mixer A/B, Glide

Die Funktionalität des Oszillatoren Moduls des Prototyps ist in den Subpatches `Glide`, `osc1`, `osc2` und `Mixer A/B` realisiert. In `osc1` und `osc2` wird die jeweils ausgewählte Wellenform der beiden Oszillatoren erzeugt. Grundlage sind dabei das `osc~` Objekt, welches eine bipolare Sinuswelle erzeugt und das `phasor~` Objekt, das eine unipolare Sägezahn Welle ausgibt. Die Dreieckwelle wird über eine Abwandlung der Sägezahnwelle realisiert, die Pulswelle entsteht aus der bearbeiteten Sinuswelle. Im Subpatch `Mixer A/B` kann mit dem Volume A/B Potentiometer zwischen den beiden Oszillatoren hin und her geblendet werden.

Mit welcher Frequenz die beiden Oszillatoren schwingen sollen, wird im `Glide` Subpatch bestimmt. Dazu wird zunächst die abgetastete Eingangsspannung des V/Oct Eingangs in die eine Frequenz umgerechnet. Dabei stellt das `clip~` Objekt sicher, dass die Frequenz nicht unter 20Hz und über 20kHz eingestellt werden kann. Mit dem Glide Potentiometer kann schließlich eine Übergangszeit eingestellt werden, in der die Grundfrequenz bei einem Eingangssignal im V/Oct Eingang zur nächsten Frequenz ansteigt oder sinkt (0 – 80ms). Dieser Übergang wird durch das `line~` Objekt umgesetzt.

Lowpass

Im `Lowpass` Subpatch befindet sich der Programmabschnitt für das Lowpass-Filter Modul des Prototyps. Das zentrale Objekt dieses Patches ist das `bob~` Objekt. Dieses Objekt ist eine Simulation eines analogen, resonanten Moog-Filter. Mit den Cutoff- und Resonance-Potentiometern können hier die beiden gleichnamigen Parameter des Filters eingestellt werden. Das Potentiometer CV Amount bestimmt, wie stark eine Steuerspannung am Cutoff CV Eingang des Filtermoduls die Grenzfrequenz des Filters moduliert.

Envelope

Der `Envelope` Subpatch erzeugt die Hüllkurve, die aus dem CV Out des Envelope Moduls am Prototyp abgegriffen werden kann. Das `vline~` Objekt generiert dafür einen linear steigenden und fallenden Zahlenverlauf, womit eine gleichermaßen lineare Steuerspannung am entsprechenden analogen Ausgang des Belabards erzeugt wird. Die Attack- und Releasezeit des Envelopes können mit den entsprechenden Potentiometern eingestellt werden. Erkennt das Belaboard am Gate Eingang des Envelopes eine positive Flanke, wird die Attackphase der Hüllkurve eingeleitet; ist diese abgeschlossen folgt direkt die Releasephase.

Main Mix

Das Mixer Modul des Prototyps ist im `Main Mix` Subpatch programmiert. Hier wird das eingehende Audiosignal mit der abgetasteten Steuerspannung des CV In des Mixers multipliziert. Dadurch wird die Amplitude des Signals mit dieser Steuerspannung moduliert. Eine Multiplikation mit dem eingelesenen Wert des Volume Potentiometers bietet zum Schluss noch die Möglichkeit die Gesamtlautstärke des Prototyps einzustellen.

In den meisten Subpatches befindet sich an erster Stelle nach den `inlet` Objekten der Subpatch `switcher`. Dieser Subpatch ermöglicht das Ausschalten der Modularität des Prototyps mit einem Schalter auf der Frontplatte. Ist dieser Schalter in der unteren Stellung (Default Patch Off), werden einprogrammierte Verbindungen zwischen den

Realisierung Phase 1

einzelnen Modulen aktiv. Alle Steuerspannungen an den verschiedenen Eingangsbuchsen an der Frontplatte werden in dieser Stellung ignoriert. Der einprogrammierte Standardpatch hat folgenden Signalfluss:

Der Output des **Clock Out** Patches wird über den **Clock In** Subpatch auf die beiden Sequencer geleitet, sowie auf den Gate Input des **Envelope** Patches. **Sequencer2** moduliert die Frequenz der beiden Oszillatoren über den **Glide** Subpatch. Die Grenzfrequenz des Filters wird durch **Sequencer1** moduliert. Schließlich wird das Audiosignal im **Main Mix** mit der Hüllkurve des **Envelope** bearbeitet.

6.2. Hardware

6.2.1. Erweiterungen zum Belaboard

Zur Bedienung des Prototyps müssen insgesamt einundzwanzig Potentiometer sowie fünf Eingänge für Steuerspannungen (dreimal analog, zweimal digital) ausgelesen werden. Ausgegeben werden insgesamt vier Steuerspannungen. Mit acht analogen Eingängen und weiteren acht analogen Ausgängen verfügt das Belaboard zwar über genügend Ausgänge, jedoch nicht über eine ausreichende Anzahl an analogen Eingängen.¹²⁶

Zur Erweiterung der analogen Eingänge wurde daher ein „Multiplexer Capelet“ genutzt, welches mechanisch und elektrisch mit dem Belaboard kompatibel ist und auf dieses aufgesteckt werden kann.¹²⁷ Das Multiplexer Capelet ermöglicht es, jeden der acht analogen Eingänge bis zu achtmal aufzuteilen, wodurch insgesamt 64 analoge Signale mit dem Belaboard abgetastet werden. Durch diese Erweiterung verringert sich die Abtastfrequenz der Eingänge um das Achtfache zur Standardabtastfrequenz von 22,05kHz auf 2,756kHz, was zur fehlerfreien Abtastung der Steuerspannungen immer noch ausreichend ist.

Die achtfache Aufteilung jedes analogen Eingangs wird beim Prototyp für die Eingänge 4 - 7 umgangen, indem eine Jumperkappe zwei dafür vorgesehene Pins des

¹²⁶ „Bela hardware - The Bela Knowledge Base“, zugegriffen 11. Januar 2022, <https://learn.bela.io/using-bela/about-bela/bela-hardware/#analog-io>.

¹²⁷ „Multiplexer Capelet – eu.shop.bela.io“, zugegriffen 12. Januar 2022, <https://eu.shop.bela.io/collections/extras/products/multiplexer-capelet>.

Realisierung Phase 1

Multiplexercaplets miteinander verbindet. Dadurch erhält man vier analoge Eingänge mit der Abtastfrequenz von 22,05kHz und weitere $4 \times 8 = 32$ Eingänge mit der niedrigeren Abtastfrequenz von 2,756kHz.

Der Audio Expander von Bela – ein weiterer Aufsatz für das Belaboard – ermöglicht es, die vier analogen Eingänge mit höherer Abtastfrequenz als weitere Audioeingänge zu nutzen. Außerdem werden über den Audio Expander die noch verfügbaren analogen Ausgänge in Audioausgänge umgewandelt. Eine detaillierte Beschreibung der Unterschiede zwischen Audioausgängen und analogen Ausgängen des Belabards finden sich auf der Website von Bela: <https://blog.bela.io/analog-vs-audio-on-bela-explained/>.

Grundsätzlich reicht die Qualität der zusätzlichen Audio In-/Outputs aus, um auch den Signalfluss der Audiosignale des Prototyps frei wählbar zu realisieren, sodass beispielsweise der Audioausgang des Oszillatormoduls direkt in den Mixer, statt in das Filtermodul gepatcht werden kann. Dies erfordert jedoch einige Anpassungen in der Elektronik des Prototyps, da Audiosignale beim Belaboard andere Spannungsspegele haben, als gewöhnliche analoge Signale. Daher wurde bei der Realisierung des ersten und zweiten Prototyps die Möglichkeit Audiosignale zu patchen vorerst nicht umgesetzt und auf die Verwendung des Audio Expanders verzichtet.

In der Software wurde dieses Feature während der Realisierung jedoch bereits vorbereitet. Die entsprechenden Ein- und Ausgangsbuchsen zum Patchen der Audiosignale wurden auch bereits in die Frontplatte integriert (s. [Ausblick](#)).

6.2.2. Weitere Elektronik

Die im Prototyp verbauten Taster sind alle mit jeweils einem Pulldown Widerstand mit Masse verbunden. Dadurch liegt bei Nicht-Betätigung des Tasters ein definiertes 0V Signal am digitalen Eingang des Belabards an.

Die 3,5mm Klinkenbuchse für den Steuerspannungseingang am Mixer Modul („CV In“) ist über einen Pullup Widerstand mit +3,3V verbunden. Dadurch wird erreicht, dass die Lautstärke des Audiosignals im Mixermodul ausschließlich über das Volume-Potentiometer eingestellt wird, wenn kein CV-Signal im CV In anliegt. Ohne diesen Pullup Widerstand würde die Lautstärke dauerhaft auf „0“ (kein hörbarer Output) bleiben, wenn der CV Input des Mixers nicht belegt ist.

6.2.3. Frontplatte

Das Design der Frontplatte wurde mit der CAD Software „Corel Draw“ der Firma „Autodesk“ erstellt. In der Software können Bohrdaten und Druckdaten auf verschiedenen Ebenen als Vektorgrafiken angelegt und exportiert werden.

Für den ersten Prototyp wurde als Material für die Frontplatte eine einfache Pressspanplatte genutzt, bei der die Bohrlöcher zur Montage der verschiedenen Bauteile per Hand aufgebohrt wurden. Ein Ausdruck des in Corel Draw erstellten Designs wurde auf die Frontplatte aufgeklebt.

7. Evaluierung

Dieses Kapitel beschreibt die Evaluierung des ersten Prototyps zur Sammlung wichtiger Richtlinien für die Umsetzung eines zweiten Prototyps. Hierbei werden die gewählte Methodik und die Durchführung der Evaluierung erläutert.

7.1. Methodik

Um sicherzustellen, dass das Konzept der ersten Realisierung das angestrebte Ziel erreicht, einen Modular-Synthesizer zu bauen, der einen möglichst einfachen Einstieg in das Eurorack System bietet, galt es den ersten Prototyp hinsichtlich seiner Funktionalität und seines Designs zu bewerten. Für diese Bewertung wurde eine qualitative Evaluierungsmethodik in Form von Fokusgruppengesprächen gewählt. Dabei wird zum Informationsgewinn in einem moderierten Diskursverfahren eine geringe Anzahl an Personen zu einem Gespräch angeregt.¹²⁸ Unter den Bedingungen der Corona Pandemie war ein klassisches Gruppeninterview leider nicht durchführbar. Daher wurden die einzelnen Personen in fokussierten Einzelinterviews befragt. Dies hat sich im Übrigen als durchaus vorteilhaft erwiesen, da sich so die Teilnehmer:innen der Evaluierung die Bedienung des Prototyps nicht im Dialog erarbeitet haben, sondern selbstständig unter Realbedingungen erschließen mussten.

7.2. Durchführung

Zur Vorbereitung der Interviews wurden Kriterien festgelegt, nach denen die bei der Evaluierung teilnehmenden Personen in drei verschiedene Fokusgruppen eingeteilt werden können: *Expert:innen*, *Neueinsteiger:innen* und *Interessierte*. Anschließend wurden insgesamt sechs Personen an unterschiedlichen Terminen (zwei Personen pro festgelegter Fokusgruppe) zur Evaluierung eingeladen. Dabei hatten die Personen zunächst die Möglichkeit den Prototyp beliebig lange zu testen. Direkt im Anschluss erfolgten die fokussierten Interviews, welche in transkribierter Form im digitalen Anhang der Bachelorarbeit zu finden sind. Zu Beginn des Interviews wurden die Teilnehmer:innen gebeten, ihre Vorerfahrung mit Modular-Synthesizern kurz zu skizzieren, damit unter Umständen eine Fehleinteilung in eine unpassende

¹²⁸ Marlen Schulz, Birgit Mack, und Ortwin Renn, Hrsg., *Fokusgruppen in der empirischen Sozialwissenschaft* (Wiesbaden: VS Verlag für Sozialwissenschaften, 2012), 9, <https://doi.org/10.1007/978-3-531-19397-7>.

Evaluierung

Fokusgruppe angepasst werden kann. Dies war jedoch bei keiner der Interviews notwendig. Alle Teilnehmer:innen sind anonymisiert und werden daher im weiteren Verlauf dieser Arbeit mit der jeweiligen Fokusgruppenbezeichnung genannt.

7.2.1. Kriterien zur Gruppeneinteilung

Expert:innen

Zur Fokusgruppe *Expert:innen* zählen Personen mit guter Kenntnis in der Bedienung modularer Synthesizer. Idealerweise sind sie im Besitz weiterer Eurorack Module, mit denen die Kompatibilität des vorgestellten Synthesizers mit weiteren Modulen getestet werden kann.

Neueinsteiger:innen

Personen in der Fokusgruppe *Neueinsteiger:innen* haben fundiertes musikalisches Grundwissen und sind geübt in der Bedienung herkömmlicher Synthesizer. Sie haben keine praktische Erfahrung mit Modular-Systemen. Allerdings ist ein grundsätzliches Interesse an modularen Systemen vorhanden. Diese Fokusgruppe ist die primäre Zielgruppe des späteren Produktes.

Interessierte

Zu den *Interessierten* zählen Musiker:innen mit grundsätzlichem Interesse an modernen Synthesizern ohne nennenswerte praktische Erfahrung mit deren Bedienung. Ziel dieser Fokusgruppe ist hauptsächlich die intuitive Bedienbarkeit des Prototyps zu überprüfen.

7.2.2. Auswahl der Teilnehmer:innen

Bei der Auswahl der Teilnehmer:innen in den Fokusgruppen gab es mehrere Rahmenbedingungen, die zu beachten waren. Um die Personen vor der Evaluierung der jeweils für sie passenden Fokusgruppe zuteilen zu können, muss die vorhandene Vorerfahrung in der Bedienung von Modular-Synthesizern vorher richtig eingeordnet werden können. Daher kamen nur dem Verfasser dieser Arbeit bekannte Personen als Teilnehmende in Frage. Wichtig war hierbei jedoch, dass alle an der Evaluierung beteiligten Personen nicht vorher in Prozesse der Konzeptions- oder Realisierungsphase des Prototyps involviert waren, damit die Ergebnisse der Bewertung nicht durch Eigenbeteiligung im Entstehungsprozess verfälscht werden

Evaluierung

könnten. Zuletzt müssen die Personen in der Lage sein, unter den Bedingungen der Corona Pandemie an einem Präsenztermin im Raum Hessen / Rheinland-Pfalz teilzunehmen.

Unter Beachtung dieser Kriterien wurden schließlich für jede Fokusgruppe jeweils zwei Teilnehmer:innen gewählt.

7.2.3. Fragenkatalog

Ein Fragenkatalog diente bei den Interviews als Leitfaden für die Gesprächsführung. Da es ein Ziel von fokussierten Interviews ist „[...] Befragten die Chance zu geben, auch nicht antizipierte Gesichtspunkte zur Geltung zu bringen“¹²⁹ wurde dieser Fragenkatalog flexibel eingesetzt und muss als Orientierungsgrundlage der Gespräche verstanden werden. Es folgt eine Übersicht über die in den Interviews angesprochenen Fragen, sortiert nach übergeordneten Themengebieten:

Bedienung:

1. Wie übersichtlich/sinnvoll findest du die Anordnung der Bedienmöglichkeiten des Prototyps?
2. Konntest du den Prototyp intuitiv zufriedenstellend bedienen?
3. Falls Frage 2 mit „Nein“ beantwortet wurde: welche Probleme hattest du bei der Bedienung?
4. Konntest du am Prototyp eigene Patches kreieren?

Sound:

5. Wie bewertest du die Soundqualität des Prototyps?
6. Gibt es einzelne Funktionen des Synthesizers, die das Audiosignal qualitativ unzureichend bearbeiten?

¹²⁹ Christel Hopf, *Qualitative Forschung. Ein Handbuch*, 9. Aufl., 2012, 354.

Evaluierung

Features:

7. Wie bewertest du den Funktionsumfang des Prototyps? (Sollte der Funktionsumfang des Prototyps noch erweitert werden / Sollte der Funktionsumfang des Prototyps zur besseren Übersicht reduziert werden?)
8. Welche Funktionen fehlen dem Prototyp?
9. Gibt es überflüssige Bedienmöglichkeiten?

Fazit / Verbesserungen:

10. Hast du Verbesserungsvorschläge?
11. Hattest du Spaß beim Bedienen des Synthesizers?
12. Könntest du dir vorstellen den fertigen Synthesizer als Einstieg für ein Eurorack-System zu nutzen?

7.3. Ergebnisse

7.3.1. Funktionalität

Der erste Prototyp wurde bezüglich seines Funktionsumfangs von allen Teilnehmer:innen im Wesentlichen positiv bewertet. *Neueinsteiger:in 2* hob dabei insbesondere hervor, dass sich der Prototyp gut als Einstiegsmodul für ein Eurorack-System eignet:

„Also ich finde es cool, dass man so einen Grundstock an Modulen hat, mit denen man sich reinfinden kann. So kann man halt irgendwie lernen so grundsätzlich damit umzugehen und Spaß dran zu haben. Und genau, ich finde das cool als Startgerät.“¹³⁰

Diese Einschätzung wird von *Expert:in 1* unterstützt. Dabei bestätigt *Expert:in 1* auch die Annahme, dass Neueinsteiger:innen beim Einstieg in ein Modular-System von der Auswahl der ersten Module überfordert sein könnten:

„Weil man vor allem ja eigentlich bei Modular-Synthesizern immer das große Problem hat, dass man super viele Module braucht, wo man gar nicht weiß, was die machen. [...] Zum Beispiel brauchst du eine Art Stromversorgung, du brauchst einen Amp, du brauchst einen

¹³⁰ „Neueinsteiger:in2_ Interview“, 12.2021, Abschn. 01:26:50.

Evaluierung

*Mixer, du brauchst... alles Mögliche, eine Clock und so... Und das wird ja dadurch quasi so ein bisschen schon mal dir abgenommen. Weil du erstmal was hast, was klingt und was Spaß macht.*¹³¹

Expert:in 2 ergänzt die Bewertung des Prototyps als gutes Einsteigermodell durch den Hinweis, dass dieser zu einem günstigen Preis verfügbar sein muss:

*„Wenn das halt irgendwie für einen guten Preis auch irgendwie verkauft wird, dann ist es halt sau das gute Einsteigerding, also es ist super gut.“*¹³²

Die Möglichkeit mit einem Schalter einen vorgegebenen Signalfluss (im Folgenden als Default Patch bezeichnet) zu aktivieren und die Modularität des Prototyps zu umgehen wurde besonders in der Gruppe *Interessierte* als hilfreich empfunden. Auf die Frage, ob der Default Patch die Bedienung des Prototyps erleichtert hätte antwortete *Interessierte:r 1*:

*„Auf jeden Fall. Ich glaube sonst würde ich irgendwann ausrasten, weil dann kein Ton mehr kommt und ich denken würde ich habe alles kaputt gemacht. Also fand das schon gut.“*¹³³

Interessierte:r 2 fügt dem noch hinzu:

*[...] Vor allem, weil ich jetzt auch so wie es [der Default Patch] gepatcht war, total glücklich war damit.*¹³⁴

Die einzigen mehrfachgenannten Verbesserungsvorschläge zur Funktionalität des Prototyps bezogen sich auf die limitierte Bedienweise des Sequencers. Sowohl *Interessierte:r 2* als auch *Expert:in 1* äußerten im Interview den Wunsch, einzelne Schritte im Sequencer ausstellen zu können oder bewusste Pausen zu setzen.¹³⁵¹³⁶

¹³¹ „Expert:in1_Interview“, 12.2021, Abschn. 01:16:47.

¹³² „Expert:in2_Interview“, 12.2021, Abschn. 01:17:09.

¹³³ „Interessierte:r1_Interview“, 12.2021, Abschn. 01:30:25.

¹³⁴ „Interessierte:r2_Interview“, 12.2021, Abschn. 01:52:38.

¹³⁵ „Interessierte:r2_Interview“, 12.2021, Abschn. 01:49:39.

¹³⁶ „Expert:in1_Interview“, 12.2021, Abschn. 01:18:49.

7.3.2. Design

Klare Richtlinien für die Realisierung des zweiten Prototyps ergaben die Fragen, die auf die Übersichtlichkeit des Designs abzielten und die verständliche Anordnung der Bedienmöglichkeiten am Prototyp prüften. Wenngleich hier einige Verbesserungsvorschläge gesammelt werden konnten, wurde die Übersichtlichkeit und einfache Bedienung des Prototyps von *Neueinsteiger:in 2* insgesamt als gut bewertet:

„[...] von der Bedienbarkeit finde ich's, oder von der Anordnung finde ich es gut so.“¹³⁷

Positiv bewertet wurde hier auch die visuelle Abgrenzung zwischen den gedachten Modulen. *Neueinsteiger:in 1*:

„Also übersichtlich finde ich halt schon, dass es halt nicht sehr viel Schnick Schnack ist, sondern dass du halt so klare Kästen hast, macht es [die Bedienung] auf jeden Fall ein bisschen leichter.“¹³⁸

Die gesammelten Verbesserungsvorschläge werden im Kapitel [Realisierung Phase 2](#) beschrieben, um Zusammenhänge zwischen vorgenommenen Anpassungen und der Fokusgruppenevaluation möglichst nachvollziehbar darzustellen.

¹³⁷ „Neueinsteiger:in2_Interview“, 12.2021, Abschn. 01:22:03.

¹³⁸ „Neueinsteiger:in1_Interview“, 12.2021, Abschn. 01:18:36.

8. Realisierung Phase 2

In diesem Kapitel sollen nun die vorgenommenen Anpassungen in der zweiten Realisierungsphase beschrieben werden. Bei den Hardwareanpassungen wurden die in den Fokusgruppengesprächen gesammelten Verbesserungsvorschläge soweit es möglich war beachtet. Die Softwareanpassungen dienten dabei lediglich zur Optimierung des Codes, nicht zur Erweiterung des Funktionsumfangs, da dieser bei der Evaluierung als ausreichend bewertet wurde (s. [Funktionalität](#)).

8.1. Hardwareanpassungen

8.1.1. Überarbeitung des Designs der Frontplatte

Bei der Evaluierung des ersten Prototyps konnte festgestellt werden, dass der größte Verbesserungsbedarf für die zweite Realisierungsphase im Design der Frontplatte bestand. Personen aus den beiden Fokusgruppen *Neueinsteiger:innen* und *Interessierte* konnten den ersten Prototyp bereits zufriedenstellend bedienen (s. [Ergebnisse](#)), in beiden Gruppen wurden jedoch Anpassungen am Design der Frontplatte vorgeschlagen, die zur besseren Verständlichkeit der einzelnen Parameter des Prototyps beitragen könnten. *Interessierte:r 1* antwortete auf die Frage, ob alle Funktionen und Parameter des Prototyps nach der Testphase verstanden wurden wie folgt:

„Also was die Regler machen schon, glaube ich. Aber nicht wie die interagieren. Ja, also wie die sich gegenseitig beeinflussen.“¹³⁹

Beim Überarbeiten des Designs wurde daher versucht die Vernetzung einzelner Parameter visuell auf der Frontplatte abzubilden. Dafür wurde das bildliche Narrativ einer „Klangfabrik“ gewählt. Die Verbindungen zwischen den Steuerspannungseingängen und den Parametern, die durch die Steuerspannungen beeinflusst werden sind dabei auf der Frontplatte über abgebildete Fließbänder und Zahnräder verbunden. Die Abbildungen entsprechen dabei dem Signalfuss des aktivierbaren Default Patches.

Außerdem wurde der von *Expert:in 1* geäußerte Vorschlag übernommen, die einzelnen Parameter des Prototyps mit Piktogrammen zu verdeutlichen.¹⁴⁰

¹³⁹ „Interessierte:r1_Interview“, 12.2021 Abschn. 01:29:41.

¹⁴⁰ „Expert:in1_Interview“, 12.2021, Abschn. 01:10:45.

Realisierung Phase 2

Sowohl *Neueinsteiger:in 1*, als auch *Neueinsteiger:in 2* hatten bei der Evaluierung des ersten Prototyps vorgeschlagen, die einzelnen Module des Prototyps zu benennen¹⁴¹ und Abkürzungen bei der Parameterbezeichnung zu vermeiden.¹⁴² Dies wurde beim angepassten Design beachtet.

Zuletzt wurden Ein- und Ausgänge am Prototyp visuell deutlich voneinander abgegrenzt. Der zweite Prototyp ist zwar gegen ein versehentliches Verbinden zweier Ausgänge abgesichert (s. [Elektronikanpassungen](#)), *Neueinsteiger:in 2* vermerkte aber im Interview, dass die Unklarheit über die Trennung von Ein- und Ausgängen ein hemmender Faktor beim Patchen war:

„Tatsächlich war ich manchmal nicht ganz sicher, was jetzt Ausgang und was Eingang ist so von den [...] Signalen her. Theoretisch ist es ja nicht so schlimm, aber trotzdem war ich da ein bisschen gehemmt, quasi zu stecken, wie es mir gerade beliebt.“¹⁴³

Beim zweiten Prototyp wurde eine 2mm dicke Frontplatte aus Aluminium gewählt. Die Bohrlöcher wurden mit einer CNC Fräse ausgefräst. Anschließend wurde die Platte mit einem Laserdrucker bedruckt.

8.1.2. Elektronikanpassungen

Wie im Abschnitt [Input/Output Handling](#) bereits erwähnt, dürfen beim Belaboard nur Spannungen zwischen 0V und +3,3V bei digitalen Eingängen und zwischen 0V und +5V bei analogen Eingängen angelegt werden. Spannungen unterhalb oder oberhalb dieser Bereiche können das Belaboard zerstören und müssen unbedingt vermieden werden.¹⁴⁴ In Kapitel [Eurorack / Doepfer A-100](#) wird beschrieben, dass im Eurorack System üblicherweise auch mit Steuerspannungen gearbeitet wird, die außerhalb der Toleranzen des Belabards liegen. Daher müssen die Eingänge gegen zu hohe oder niedrige Spannung elektronisch geschützt werden. Außerdem muss verhindert werden, dass das versehentliche Kurzschließen von zwei Steuerspannungs-Ausgängen des Prototyps mit einem Patch Kabel Schäden am Belaboard verursacht.

¹⁴¹ „Neueinsteiger:in2_Interview“, 12.2021, Abschn. 01:22:03.

¹⁴² „Neueinsteiger:in1_Interview“, 12.2021, Abschn. 01:21:34.

¹⁴³ „Neueinsteiger:in2_Interview“, 12.2021, Abschn. 01:22:41.

¹⁴⁴ „Bela hardware - The Bela Knowledge Base“.

Realisierung Phase 2

Diese beiden Schutzfunktionen wurden durch eine Schaltung aus zwei Schottky Dioden und einem Widerstand gelöst.

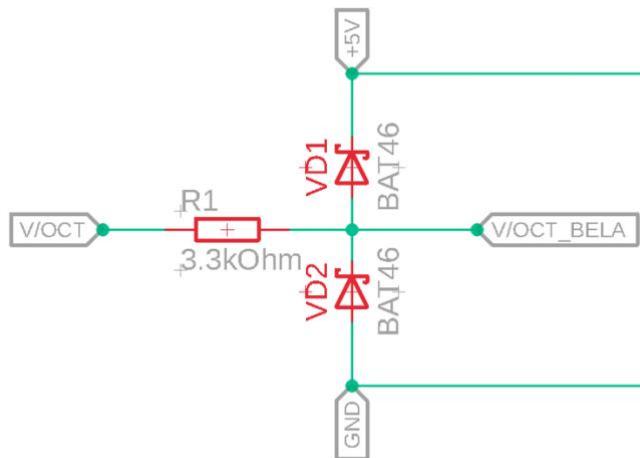


Abbildung 7: Schutzschaltung für In-/Outputs des Belabboards (erstellt in „Eagle“)

Spannungen über +5V werden hier über die Diode D1 abgeleitet, Spannungen unter 0V über die Diode D2. Der Widerstand R1 begrenzt bei einem Kurzschluss zweier Ausgänge den Strom auf maximal $5V/3,3k\Omega = 1,5mA$ und verhindert eine zu hohe Belastung des Belabboards. Jeder für Steuerspannungen verwendete Ein- und Ausgang des Belabboards ist über eine solche Schaltung mit der jeweiligen Buchse an der Frontplatte verbunden.

Die Ansteuerung der LEDs übernehmen 3 Integrierte Schaltkreise des Typs „74HC 4017“. Diese ICs sind sogenannte Dekadenzähler mit 10 Outputs, die bei einer positiven Flanke am Clock Input des ICs nacheinander auf HIGH geschaltet werden. Über einen Reset-Input können die ICs zurückgesetzt werden, sodass wieder der erste Output auf HIGH gesetzt wird. Die LEDs des Sequencer Moduls, die vier LEDs zur Anzeige der Waveform A und die vier LEDs zur Anzeige der Waveform B sind jeweils an ein solches IC angeschlossen. Bei der Sequencer Schaltung wird das Signal am Gate Input des Moduls als Clock Signal der Schaltung genutzt. Der Reseteingang des ICs ist mit dem neunten Output des ICs verbunden.

Die Clock Inputs der beiden Waveform LED-Schaltungen werden über die beiden Taster Waveform A und Waveform B angesteuert. Bei beiden ICs sind die Reseteingänge jeweils mit dem fünften Output des ICs verbunden. Die restliche Beschaltung kann dem Datenblatt des „74HC 4017“ ICs (im digitalen Anhang) entnommen werden.

Realisierung Phase 2

Die 7 Segmentanzeige, welche die ausgewählte Sequencer 2 Vorlage anzeigt, wird über das IC „74HC 4543“ gesteuert. Das Belaboard übermittelt dem IC über vier Inputpins die Zahl der ausgewählten Vorlage in binärer Schreibweise. Das IC übersetzt die Zahl in die passende Kombination von HIGH und LOW an den Outputs, sodass die einzelnen Segmente der 7 Segmentanzeige die Zahl der ausgewählten Vorlage anzeigen. Im Datenblatt des ICs wird die weitere Beschaltung des Bauteils beschrieben. Das Datenblatt befindet sich im digitalen Anhang der Arbeit.

Alle LEDs der 7 Segmentanzeige sowie alle weiteren am Prototyp verbauten LEDs sind mit jeweils einem 75Ω Vorwiderstand in Reihe geschaltet.

Für die Elektronik wurde mit der Software *Eagle* von Autodesk ein Platinen Layout erstellt und anschließend eine zweischichtige Platine geätzt. Das Platinen Layout sowie der vollständige Schaltplan des Prototyps sind dieser Bachelorarbeit angehängt (digitaler Anhang).

8.2. Softwareanpassungen

8.2.1. Erneute Programmierung in C++

Die Programmierung der Audiosynthese in Pure Data eignet sich zwar hervorragend zur schnellen prototypischen Entwicklung, führt aber auf dem Belaboard zu einer hohen CPU-Auslastung.

The messaging architecture and the audio engine of Pd was not designed to be fast and computationally efficient which can lead performance penalties on platforms with limited computational power.¹⁴⁵

Durch eine Erhöhung der Blocksize der zu verarbeitenden Audiosamples pro Programmzyklus auf 64 Samples kann die CPU-Auslastung verringert werden, jedoch steigt mit einer größeren Blocksize die Gesamtlatenz des Systems. Zur Optimierung der Software des Prototyps für eventuelle Anpassungen und Erweiterungen in der Zukunft (s. [Ausblick](#)) wurde daher in einem zweiten Programmierdurchlauf die

¹⁴⁵ Giulio Moro u. a., „Making High-Performance Embedded Instruments with Bela and Pure Data“, 2016, 3.

Realisierung Phase 2

Software vollständig mit C++ Code realisiert. Einen wichtigen Leitfaden boten hierbei Materialien des Mastermoduls "Musik- und Audioprogrammierung", welches von Dr. Andrew McPherson an der Fakultät für Elektronik und Informatik der Queen Mary University of London unterrichtet wird.¹⁴⁶

Prinzipiell ist die Programmstruktur des C++ Codes vergleichbar mit der programmierten Audiosynthese in Pure Data. Die einzelnen Komponenten des Prototyps wurden soweit es möglich war in dedizierten Programmabschnitten programmiert. Im Code sind diese Abschnitte durch Kommentarzeilen markiert.

Der gesamte C++ Code ist im digitalen Anhang dieser Arbeit einzusehen.

8.3. Nicht realisierbare Anpassungen

8.3.1. Erweiterung des Sequencers

Die in der Evaluierung gewünschte Erweiterung des Sequencers um eine Stopfunktion konnte in der Kürze der Bearbeitungszeit nicht mit in die Realisierung des zweiten Prototyps aufgenommen werden. Mit dieser Erweiterung wäre eine Anpassung in der Hardware verbunden, die mit der jetzigen Bauform des Modular Synthesizers nicht realisierbar ist. In Überlegungen für eine nächste Version des Prototyps wurde der Vorschlag aufgenommen (s. [Ausblick](#)).

¹⁴⁶ „Course Introduction - The Bela Knowledge Base“, zugegriffen 11. Januar 2022, <https://learn.bela.io/tutorials/c-plus-plus-for-real-time-audio-programming/course-introduction/>.

Realisierung Phase 2

8.4. Übersicht Funktionsumfang

Es folgt eine Übersicht über alle Module des Prototyps und ihre einstellbaren Parameter sowie Ein- und Ausgänge der Steuerspannungen (CV). Hierbei muss erwähnt werden, dass die Einteilung in Module lediglich gedanklichen Ursprungs ist. Der Prototyp stellt selbst ein einziges Modul als Ganzes dar.

Oszillator A + B

Parameter	Funktion
Osc Freq	Grundfrequenz der beiden Oszillatoren A + B, (55 – 1055 Hz)
Waveform A	Wellenform A (Sinus-, Dreieck-, Sägezahn- oder Pulswelle)
Waveform B	Wellenform B (Sinus-, Dreieck-, Sägezahn- oder Pulswelle)
Pulse Width A	Pulswerte von Oszillator A (Bei Wellenform Pulswelle)
Pulse Width B	Pulswerte von Oszillator B (Bei Wellenform Pulswelle)
Volume A/B	Blendet zwischen Oszillator A und B. Bei Mittelstellung sind beide Oszillatoren gleich laut
Detune	Verstimmt beide Oszillatoren zueinander um wenige Hertz im Abstand zur Grundfrequenz
Glide	Übergangszeit, in der die Grundfrequenz bei einem Eingangssignal im V/Oct Eingang (s.u.) zur nächsten Frequenz ansteigt oder sinkt (0 – 80ms)

CV In/Out	Funktion
V/Oct (Input)	Voltage per Octave Eingang. Moduliert die Grundfrequenz der Oszillatoren A+B. 0V entsprechen 55Hz + eingestellter Osc Freq (s.o.). Jede 1V Steigerung entspricht einer Verdopplung der Frequenz.

Filter:

Parameter	Funktion
Cutoff	Grenzfrequenz des Tiefpassfilters (80 -15000Hz)
Resonance	Resonanz des Tiefpassfilters
CV Amount	Bestimmt, wie stark die Grenzfrequenz des Tiefpassfilters vom Signal im CV In des Filters (s.u.) beeinflusst wird.

Realisierung Phase 2

CV In/Out	Funktion
CV In (Input)	Eingehende Steuerspannung moduliert die Grenzfrequenz des Tiefpassfilters

Envelope

Parameter	Funktion
Attack	Einschwingzeit der Hüllkurve (0 - 100ms)
Release	Ausklingzeit der Hüllkurve (0.1 - 300ms)

CV In/Out	Funktion
Gate (Input)	Steigende Flanke am Eingang startet Hüllkurve (Attack-Time) Fallende Flanke am Eingang beendet Hüllkurve (Release-Time)
CV Out (Output)	Gibt den Verlauf der Hüllkurve als Steuerspannung aus

Sequencer 1+2

Parameter	Funktion
1-8	Spannungswerte der Schritte 1-8 von Sequenz 1
Seq 2 Preset	Auswahl der Vorlage für Sequenz 2

CV In/Out	Funktion
Gate (Input)	Steigende Flanke des Eingangssignal lässt Sequenz 1 und 2 einen Schritt voranschreiten
Seq1 Out (Output)	Gibt den Verlauf von Sequenz 1 als Steuerspannung aus
Seq2 Out (Output)	Gibt den Verlauf von Sequenz 2 als Steuerspannung aus

Realisierung Phase 2

Clock

Parameter	Funktion
Tempo	Tempo (50 - 230 Bpm)

CV In/Out	Funktion
Clock (Output)	Ausgabe des Metronoms. Bei jedem Tick wird der Ausgang kurz auf 1 (3,3V) und anschließend wieder auf 0 (0V) geschaltet.

Mixer

Parameter	Funktion
Volume	Ausgangslautstärke

CV In/Out	Funktion
CV In (Input)	Moduliert die Lautstärke

9. Fazit

9.1. Konzept und Realisierung

Für ein umfassendes Urteil wäre eine weitere Testphase mit dem zweiten Prototyp notwendig. Allerdings lässt sich nach den Ergebnissen der Fokusgruppenevaluierung feststellen, dass der Ansatz einen Prototyp zu entwickeln, der Neueinsteiger:innen den Einstieg in Modular-Systeme erleichtert, grundsätzlich gelungen ist.¹⁴⁷ Der Prototyp erfüllt die aus den Kompositionsgrundlagen hergeleiteten Anforderungen zum Funktionsumfang (s. [Sounddesign / Funktionsumfang](#)) und schafft es dabei, eine für Neueinsteiger:innen verständliche, übersichtliche Bedienung anzubieten. *Neueinsteiger:in 1* fasst dies folgendermaßen zusammen:

„Das ist ja irgendwie auch schön, dass du nicht nach 10 Minuten das Gefühl hast, du hast alles gemacht, was mit so einem Gerät funktioniert. Und trotzdem waren es nicht so viel Knöpfe, dass ich mich hier hab erschlagen gefühlt.“¹⁴⁸

Diese Einschätzung deckt sich auch mit eigenen Beobachtungen während der Testphase vor den fokussierten Einzelinterviews. Bereits nach kurzer Zeit waren hier die Personen aus allen Fokusgruppen in der Lage, den Prototyp für sie zufriedenstellend zu bedienen. Der einstellbare Default Patch erweist sich dabei als sinnvoll und wurde von allen Interviewpartner:innen positiv hervorgehoben. Eine interessante Beobachtung war hier, dass der als Einstiegshilfe gedachte Schalter zum Umstellen auf den Default Patch auch in der Fokusgruppe Expert:innen genutzt wurde, obwohl Personen in dieser Gruppe bereits Erfahrung mit dem Patchen bei Modular-Systemen mitbrachten. Im Interview erklärte *Expert:in 2*, in einem größeren Modular-System mit mehreren Modulen sei es angenehm, einzelne Komponenten auch mal nicht mit Patchkabeln verbinden zu müssen.¹⁴⁹

Innerhalb der Bearbeitungszeit der Bachelorarbeit war es nicht möglich die Kompatibilität des Prototyps mit weiteren Modulen im Eurorack-Format zu überprüfen und gegebenenfalls anzupassen. Bei der Entwicklung des Prototyps lag hierbei die höchste Priorität zunächst beim Schutz gegen zu hohe/niedrige Eingangsspannungen.

¹⁴⁷ „Neueinsteiger:in2_Interview“, 12.2021, Abschn. 01:25:25.

¹⁴⁸ „Neueinsteiger:in1_Interview“, 12.2021, Abschn. 01:26:46.

¹⁴⁹ „Expert:in2_Interview“, 12.2021, Abschn. 01:19:54.

Fazit

So kann der Prototyp zwar ohne weitere Sicherheitsmaßnahmen beachten zu müssen mit allen weiteren Eurorack Modulen verbunden werden, es werden jedoch nur Steuerspannungen zwischen 0V und 5V sinnvoll verwertet. Das schränkt den einfachen Einsatz des Prototyps in einem größeren System ein. Für vollständige Modularität fehlt zudem die Möglichkeit, Audiosignale zu patchen. Diese Funktion ist softwareseitig bereits vorbereitet worden, für eine Umsetzung muss allerdings noch eine Hardwarelösung entwickelt werden.

In der [Marktübersicht](#) lässt sich bei Modular-Systemen eine Marktlücke für Module, die explizit an Neueinsteiger:innen adressiert sind, erkennen, welche der Prototyp füllt. Durch einen stärkeren Fokus auf die marktwirtschaftlichen Aspekte hätte bei der Recherche diese Einschätzung womöglich besser beurteilt werden können. Bei der Entwicklung des Prototyps wurde jedoch bewusst ein Schwerpunkt auf die bestmögliche Umsetzung der Konzeption gesetzt, ohne dabei wirtschaftliche Überlegungen zu sehr in diese einfließen zu lassen.

Insgesamt hat der Prototyp die Erwartungen vor der Realisierung übertroffen. Es ist gelungen einen produktnahen Prototyp zu entwickeln, der klangliche Vielfalt ermöglicht, verständlich zu bedienen ist und Spaß an Modular-Synthesizern vermittelt.

9.2. Ausblick

Bei einer Weiterentwicklung des vorgestellten Prototyps sollte der Fokus zunächst auf der Kompatibilität mit weiteren Eurorack Modulen liegen. Durch die klare Einordnung des Prototyps als Einstiegsmodul ist es wichtig, eine Erweiterung des Modular-Systems durch weitere Module von beliebigen Herstellern so einfach wie möglich zu gestalten. Daher sollte in einem nächsten Entwicklungsschritt der Prototyp so erweitert werden, dass er die üblichen Spannungsbereiche der A-100 Serie (s. [Eurorack / Doepfer A-100](#)) auslesen und ausgeben kann.

Für diese und weitere kommende Anpassungen wäre es außerdem sinnvoll, eine größere Platine zu entwerfen, auf der auch alle In-/Outputs und Bedienmöglichkeiten direkt Platz finden.

So ließe sich dann der Sequenzer auch um die in den Fokusgruppengesprächen erwähnten Funktionen erweitern.

Um den Lerneffekt in der Bedienung des Prototyps zu vergrößern, könnten weitere Default Patches eingefügt werden. Dadurch würde der Unterschied zwischen verschiedenen Signalwegen bei der Audiosynthese durch das Wechseln der Patches

Fazit

schnell erlebbar werden und Nutzer:innen würden eventuell mehr zum Kreieren eigener Patches angeregt werden. Eine mögliche Visualisierungshilfe wäre hierbei die Bereitstellung von Schablonen für jeden weiteren Default Patch, die auf der Frontplatte des Prototyps aufgesetzt den Signalfluss skizzieren.

Da die DIY-Szene auf dem Eurorack Markt eine wichtige Rolle spielt, wäre es außerdem wirtschaftlich sinnvoll, zu versuchen, diese in einer überarbeiteten Version des Prototyps stärker zu adressieren. Eine Öffnung der Softwareschnittstelle über eine USB-Buchse auf der Frontplatte wäre hier eine mögliche Lösung. Diese würde es Nutzer:innen ermöglichen, den Prototyp nach eigenem Belieben umzuprogrammieren. Zuletzt muss wegen der hohen Kosten in Erwägung gezogen werden, eine Alternative zum Belaboard zu suchen. Durch die erneute Programmierung in C++ ist eine Übertragung des Programms auf ein anderes System prinzipiell mit einem überschaubaren Aufwand zu bewerkstelligen.

Abbildungsverzeichnis

Abbildung 1: Skizze zur Lage der Montagelöcher	19
Abbildung 2: Steckverbinder Busplatine	19
Abbildung 3: Prozessstruktur des Belabards	36
Abbildung 4: Pure Data Integration beim Belaboard	40
Abbildung 5: State Machine.....	43
Abbildung 6: Pure Data Patch, Prototyp 1	47
Abbildung 7: Schutzschaltung für In-/Outputs des Belabards (erstellt in „Eagle“)....	61

Literaturverzeichnis

Abras, Chadia, Diane Maloney-Krichmar, und Jenny Preece. „User-Centered-Design“, 2004, 15.

Akins, Joseph. „An Overview of Electronic Music History“, 10. Oktober 2019.

Arduino Official Store. „Arduino Due“. Zugegriffen 8. Januar 2022.
<http://store.arduino.cc/products/arduino-due>.

Barabanov, Michael. „A Linux-Based Real-Time Operating System“, 1. Juli 1997, 43.

„BeagleBoard.org - black“. Zugegriffen 8. Januar 2022. <https://beagleboard.org/black>.

„Bela hardware - The Bela Knowledge Base“. Zugegriffen 11. Januar 2022.
<https://learn.bela.io/using-bela/about-bela/bela-hardware/#analog-io>.

„Bela Starter Kit – eu.shop.bela.io“. Zugegriffen 8. Januar 2022.
<https://eu.shop.bela.io/products/bela-starter-kit>.

Bode, Harald. „A New Tool for the Exploration of Unknown Electronic Music Instrument Performances“. *J. Audio Eng. Soc.* 9, Nr. 4 (Oktober 1961).

Bode, Harald. „History of Electronic Sound Modification“. *J. Audio Eng. Soc.*, Oktober 1984, 10.

Brinkmann, Peter, Chris McCormick, Peter Kirn, Martin Roth, Richard Lawler, und Hans-Christoph Steiner. „Embedding Pure Data with Libpd“, 2011, 8.

„C++ - The Bela Knowledge Base“. Zugegriffen 10. Januar 2022.
<https://learn.bela.io/using-bela/languages/c-plus-plus/>.

Cre8audio. „Cellz - Touch Controller, Sequencer, Arpeggiator & All Around Wacky Eurorack Compatible Synth Module“. Zugegriffen 18. Januar 2022.
<https://www.cre8audio.com/cellz>.

„Combining Pure Data and C++ - The Bela Knowledge Base“. Zugegriffen 10. Januar 2022. <https://learn.bela.io/tutorials/pure-data/advanced/custom-render/>.

Corman, Emmett. „Simple Synthesis: Part 8, Wavefolding“. Blog. Keith McMillen Instruments, 17. Juni 2015. <https://www.keithmcmillen.com/blog/simple-synthesis-part-8-wavefolding/>.

„Course Introduction - The Bela Knowledge Base“. Zugegriffen 11. Januar 2022.
<https://learn.bela.io/tutorials/c-plus-plus-for-real-time-audio-programming/course-introduction/>.

Crab, Simon. „The ‘Sound Processor’ or ‘Audio System Synthesiser’ Harald Bode, USA, 1959“. *120 Years of Electronic Music* (blog), 30. Januar 2014.
<https://120years.net/wordpress/the-sound-processor-harald-bode-germany/>.

Literaturverzeichnis

„Crafting GUIs - The Bela Knowledge Base“. Zugegriffen 10. Januar 2022.
<https://learn.bela.io/the-ide/crafting-guis/>.

Cre8audio. „About Us“. Cre8audio. Zugegriffen 18. Januar 2022.
<https://www.cre8audio.com/about-us>.

Dalgleish, Mat. „The Modular Synthesizer Divided: The Keyboard and Its Discontents“, 2016, 15.

„Dato DUO“. Zugegriffen 18. Januar 2022. <https://dato.mu/>.

DivKid. *The History and Future of Eurorack Modular Synths with Doepfer*. Interview. Thomann Synth Reactor 2019, 2019.
<https://www.youtube.com/watch?v=CpyuqP5r9xU>.

Doepfer Musikelektronik GmbH. „A-100 Hauptseite/Einführung“. Zugegriffen 1. Januar 2022. <https://doepfer.de/a100d.htm>.

Doepfer Musikelektronik GmbH. „A-100 Overview“. Zugegriffen 30. Dezember 2021.
<https://doepfer.de/a100.htm>.

Doepfer Musikelektronik GmbH. „Mechanische Details A-100“. Zugegriffen 30. Dezember 2021. https://doepfer.de/a100_man/a100m_d.htm.

Doepfer Musikelektronik GmbH. „Technische Hinweise A-100“. Zugegriffen 30. Dezember 2021. https://doepfer.de/a100_man/a100t_d.htm.

„Don Buchla Keyboard Magazine Interview - MOD WIGGLER“. Zugegriffen 1. Januar 2022. <https://modwiggler.com/forum/viewtopic.php?t=235923>.

Dunn, David. „A History of Electronic Music Pioneers“. *Eigenwelt Der Apparatewelt: Pioneers of Electronic Art*, 1992, 46.

EMS. „EMS-Synthi-AKS Service Manual“, 12. September 1970.

„Expert:in1_Interview“, 2022

„Expert:in2_Interview“, 2022

Gárdonyi, Zsolt, und Hubert Nordhoff. *Harmonik*. Möseler Verlag, 2002.

Mixdown Magazine. „Gear Rundown: Pink Floyd's Dark Side Of The Moon“, 2. März 2021. <https://mixdownmag.com.au/features/gear-rundown-pink-floyds-dark-side-of-the-moon/>.

Görne, Thomas. *Tontechnik: Hören, Schallwandler, Impulsantwort und Faltung, digitale Signale, Mehrkanaltechnik, tontechnische Praxis; mit 33 Tabellen*. 4., Aktualisierte Aufl. München: Hanser, 2015.

Literaturverzeichnis

Groves, Wesley, Tony Rivas, Theron Humiston, Grant Winship, und Joel Handley. „Beginner’s Guide to Eurorack: Case Basics, Power Supplies, and Your First Modules“. Blog. reverb.com, 6. Februar 2020. <https://reverb.com/news/beginners-guide-to-eurorack-case-basics-oscillators-filters>.

Hennion, Antoine, und Christophe Levaux. *Rethinking Music through Science and Technology Studies*. Herausgegeben von Antoine Hennion und Christophe Levaux. 1. Aufl. Routledge, 2021. <https://doi.org/10.4324/9780429268830>.

Holmes, Thom. *Electronic and Experimental Music: Pioneers in Technology and Composition*. 2. Aufl. London: Psychology Press, 2002.

Hopf, Christel. *Qualitative Forschung. Ein Handbuch*. 9. Aufl., 2012.

„Interessierte:r1_Interview“, 12.2021

„Interessierte:r2_Interview“, 12.2021

Kam, Rob. „Eurorack“. Synth DIY Wiki, 19. Oktober 2021. <https://sdiy.info/wiki/Eurorack>.

Kreidler, Johannes. „Programming Electronic Music in Pd“. Zugegriffen 12. Januar 2022. <http://pd-tutorial.com/english/ch02s02.html>.

„Learning Synths“. Zugegriffen 18. Januar 2022. <https://learningsynths.ableton.com/>.

Longfellow, Matthew. *The Making of The Dark Side Of The Moon*, 2003.

Ltd, Raspberry Pi. „Buy a Raspberry Pi 4 Model B“. Raspberry Pi. Zugegriffen 8. Januar 2022. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.

Manning, Peter. *Electronic and Computer Music*. Rev. and Expanded ed. Oxford ; New York: Oxford University Press, 2004.

McPherson, Andrew P, und Victor Zappi. „An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black“, 2015, 7.

„Meet the IDE - The Bela Knowledge Base“. Zugegriffen 10. Januar 2022. <https://learn.bela.io/the-ide/meet-the-ide/>.

Messiaen, Olivier. *Technique de mon langage musical*, 1944.

Meyer, Chris. „1 v/Oct | Learning Modular“, 16. November 2016. <https://learningmodular.com/glossary/1-voct/>.

Michon, Romain, Yann Orlarey, Stéphane Letz, Dominique Fober, und Dirk Roosenburg. „Embedded Real-Time Audio Signal Processing With Faust“, 2020, 9.

Literaturverzeichnis

„Modules on ModularGrid“. Zugegriffen 4. Januar 2022.
<https://www.modulargrid.net/e/modules/browser?SearchName=&SearchVendor=&SearchFunction=49&SearchSecondaryfunction=11&SearchHeight=&SearchText=&SearchMethod=max&SearchBuildtype=a&SearchLifecycle=&SearchSet=&SearchMarketplace=0&SearchIsmodeled=0&SearchShowothers=0&order=tag&direction=asc>.

Moro, Giulio, Astrid Bin, Robert H Jack, Christian Heinrichs, und Andrew P McPherson. „Making High-Performance Embedded Instruments with Bela and Pure Data“, 2016, 5.

Fact Magazine. „Morton Subotnick on 50 Years of Silver Apples of the Moon“, 13. Juli 2017. <https://www.factmag.com/2017/07/13/morton-subotnick-silver-apples-of-the-moon/>.

„Multiplexer Capelet – eu.shop.bela.io“. Zugegriffen 12. Januar 2022.
<https://eu.shop.bela.io/collections/extras/products/multiplexer-capelet>.

„Neueinsteiger:in1_Interview“, 12.2021

„Neueinsteiger:in2_Interview“, 12.2021

Nies, von Joker. „Der Buchla-Sound: Synthesizer der besonderen Art“. KEYBOARDS, 14. März 2017. <https://www.keyboards.de/stories/der-buchla-sound-synthesizer-der-besonderen-art/>.

Cre8audio. „NiftyBUNDLE - Affordable Powered Eurorack Synth Case, Modules, & Cables - Modular Mayhem for All!“ Zugegriffen 4. Januar 2022.
<https://www.cre8audio.com/niftybundle>.

Pinch, Trevor, und Frank Trocco. *Analog Days: The Invention and Impact of the Moog Synthesizer*. Cambridge, MA: Harvard University Press, 2002.

„Plan your modular synthesizer rack on ModularGrid“. Zugegriffen 31. Dezember 2021. <https://www.modulargrid.net/e/modules/home>.

Prendergast, Mark. *The Ambient Century: From Mahler to Moby - the Evolution of Sound in the Electronic Age*. New ed. London, 2003.

Puckette, Miller. „Encapsulation in Pd“, 30. Dezember 2006.
<http://msp.ucsd.edu/techniques/v0.11/book-html/node67.html>.

Puckette, Miller. „Pure Data“. *ICMC*, 1977, 4.

„Pure Data - The Bela Knowledge Base“. Zugegriffen 10. Januar 2022.
<https://learn.bela.io/using-bela/languages/pure-data/>.

Reising, Russel. „Speak to Me“: *The Legacy of Pink Floyd's The Dark Side of the Moon*. Ashgate Publishing Company, 2005.

Rossmay, Beat, und Alexander Wiethoff. „The Modular Backward Evolution – Why to Use Outdated Technologies“. *NIME Conference*, 2019, 6.

Literaturverzeichnis

Rumsey, Francis. „Modular Synths and Embedded Computing“. *J. Audio Eng. Soc.* 68, Nr. 3 (2020): 4.

Schulz, Marlen, Birgit Mack, und Ortwin Renn, Hrsg. *Fokusgruppen in der empirischen Sozialwissenschaft*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2012. <https://doi.org/10.1007/978-3-531-19397-7>.

Singh, A.K. *Microcontroller and embedded system*. New Age International, 2008.

Sneed, Harry M. „Encapsulation of Legacy Software: A Technique for Reusing Legacy Software Components“. *Annals of Software Engineering* 9 (2000): 293–313.

Society of Mucsic Merchants. „Pressemitteilung SOMM“, 2018. https://www.somm.eu/uploads/media/PM_SOMM_MI-Wirtschaft_2019_FINAL_190401_mn.pdf.

„Sound on Sound - Cre8Audio NiftyBundle Review“. Review, August 2020. <https://www.soundonsound.com/reviews/cre8audio-niftybundle>.

„Studie: Tasteninstrumente gehören zu den Umsatzbringern der deutschen Musikinstrumentenbranche“, 8. Mai 2019. <https://www.musikmedia.de/musikbranche/studie-tasteninstrumente-gehoeren-zu-den-umsatzbringern-der-deutschen-musikinstrumentenbranche/>.

SYNTHHEAD. „The Iconic Sounds Of Synthesis – ‘On The Run’ From ‘Dark Side of the Moon’“. *Synthtopia* (blog), 7. Dezember 2013. <https://www.synthtopia.com/content/2013/12/07/the-iconic-sounds-of-synthesis-on-the-run-from-dark-side-of-the-moon/>.

Syntorial. „Syntorial: The Ultimate Synthesizer Tutorial“. Zugegriffen 18. Januar 2022. <https://www.syntorial.com/>.

„Teensy® 4.0“. Zugegriffen 8. Januar 2022. <https://www.pjrc.com/store/teensy40.html>.

„The oscilloscope - The Bela Knowledge Base“. Zugegriffen 10. Januar 2022. <https://learn.bela.io/the-ide/the-oscilloscope>.

Thomann GmbH. „cre8audio NiftyBundle – Musikhaus Thomann“. Zugegriffen 4. Januar 2022. https://www.thomann.de/de/cre8audio_niftybundle.htm.

Thomann GmbH. „Online-Katalog Herstellerübersicht – Musikhaus Thomann“. Zugegriffen 3. Januar 2022. https://www.thomann.de/de/cat_brands.html?catKey=tasymo.

Thomann GmbH. „Thomann, Modulare Systeme“. www.thomann.de, 14. Januar 2016. https://web.archive.org/web/20160114003953/http://www.thomann.de/de/modulare_systeme.html.

Eigenständigkeit und Archivierung

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig erstellt und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Soweit ich auf fremde Materialien, Texte oder Gedankengänge zurückgegriffen habe, enthalten meine Ausführungen vollständige und eindeutige Verweise auf die Urheber und Quellen.

Alle weiteren Inhalte der vorgelegten Arbeit stammen von mir im urheberrechtlichen Sinn, soweit keine Verweise und Zitate erfolgen.

Mir ist bekannt, dass ein Täuschungsversuch vorliegt, wenn die vorstehende Erklärung sich als unrichtig erweist.

Darmstadt, 28.01.2022

M. Hessel

Ort, Datum Unterschrift

Erklärung zur Archivierung

Bitte zutreffendes ankreuzen:

Mit der Archivierung der gedruckten Abschlussarbeit in der Bibliothek

bin ich einverstanden.

bin ich nicht einverstanden.

Begründung:

- Die Arbeit ist gesperrt, da sie in einem Betrieb durchgeführt wurde und ihr Inhalt ausdrücklich durch diesen gesperrt ist. (Vgl. ABPO § 18 (9))
- Persönliche Gründe

Darmstadt, 28.01.2022

M. Hessel

Ort, Datum Unterschrift

Danksagung

Professor Thorsten Greiner

Professor Dr. Jürgen Schwab

Jonathan Hassel

Dorothée Hassel

Hans-Georg Hassel

Matthias Piechazcek

Marietta Osterlow

Max Werz

Manuel Schalk

Daniel Kroos

Maté Sipos

Justin Niestroj

Digitaler Anhang

Fokusgruppenevaluierung

Interessierte.r1_Interview.pdf
Interessierte.r2_Interview.pdf
Neueinsteiger.in1_Interview.pdf
Neueinsteiger.in2_Interview.pdf
Expert.in1_Interview.pdf
Expert.in2_Interview.pdf

Quellcode

1.Prototyp_Code (C++ Wrapper & Pure Data Patch)
2.Prototyp_C++Code.cpp

Schaltung & Platinenlayout

Eagle Files
2.Prototyp_Schaltplan.png
2.Prototyp_Plattenlayout.png
Datenblatt_4543.pdf
Datenblatt_4017.pdf

Prototyp 1

1.Prototyp_vorne.png

Prototyp 2

2.Prototyp_vorne1.png
2.Prototyp_vorne2.png
2.Prototyp_Platine_unverkabelt.png
2.Prototyp_Platine_verkabelt.png

Design

2.Prototyp_Design.png
2.Prototyp_CNC.cdr
2.Prototyp_Druck.cdr
1.Prototyp_Design.svg

Video

Videodemonstration.mp4

Literatur

Literaturquellen (Zotero Bibliothek)