



***Synthorn* - en digital modulär synt**

Bedömer kostnadseffektiviteten av en digital modulär synt byggd med mikrokontroller

Kandidatarbete inom data- och informationsteknik

H. Andersson, F. Bakken, R. Brenander, G. Hartwig, H. Johansson, A. Skoglund

CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2026
www.chalmers.se

KANDIDATARBETE SLUTRAPPORT 2026

Synthorn - En digital modulär synt

Bedömer kostnadseffektiviteten av en digital modulär synt byggd
med mikrokontroller

H. Andersson, F. Bakken, R. Brenander, G. Hartwig, H. Johansson,
A. Skoglund



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Institutionen Data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2026

Synthorn - En digital modulär synt

Bedömer kostnadseffektiviteten av en digital modulär synt byggd med mikrokontroller

H. Andersson, F. Bakken, R. Brenander, G. Hartwig, H. Johansson, A. Skoglund

© H. Andersson, F. Bakken, R. Brenander, G. Hartwig, H. Johansson, A. Skoglund 2026.

Handledare: Erik Sintorn, Institutionen Data- och informationsteknik

Examinator: Arne Linde, Institutionen Data- och informationsteknik

Kandidatarbete slutrapport 2026

Institutionen Data- och informationsteknik

Chalmers Tekniska Högskola

SE-412 96 Göteborg

Sverige

Telefon +46 31 772 1000

Typeset in L^AT_EX

Göteborg, Sverige 2026

Sammandrag

Modulära synthesizers är dyra i jämförelse med andra instrument av liknande storlek och komplexitet. Rapporten undersöker om anledningen till detta är deras användning av analoga komponenter. Syftet är att undersöka om det är möjligt att bygga en kostnadseffektiv modulär synthesizer med mikrokontrollers.

Rapporten fastställer för- och nackdelar med att använda en Raspberry Pi Pico som primär processorenhet, samt kostnaderna för att konstruera de olika moduler som bygger upp synthesizern. Rapporten täcker utvecklingen och konstruktionen av den modulära synthesizern från design till fysisk hårdvara.

Resultaten visar att en funktionell modulär synthesizer kan byggas med mikrokontrollers till 80 % av kostnaden för kommersiella alternativ. Digital implementation möjliggör avancerade funktioner såsom komplex effektbearbetning, även om prestandakrav i realtid begränsar bearbetningstiden.

Rapporten drar slutsatsen att mikrokontrollerbaserade modulära synthesizers är tekniskt och ekonomiskt möjliga. Däremot gör begränsningar i kostnadsanalysens metodik och kvalitetsverifiering det svårt att definitivt bedöma kommersiell lönsamhet i större skala.

Nyckelord: modulär, synt, synthesizer, eurorack, mikrokontroller, digital, ljudsyntes, embedded

Abstract

Modular synthesizers are expensive in comparison to other instruments of similar size and complexity. The report will investigate if the reason for this is their use of analog components. It aims to examine if it is possible to build a cost-effective modular synthesizer with microcontrollers.

The report determines the pros and cons of using a Raspberry Pi Pico as a primary processing unit as well as the expenditures to construct the different modules that build up the synth. The report covers the development and construction of the modular synthesizer from design to physical hardware.

Results demonstrate that a functional modular synthesizer can be built using microcontrollers at 80% of the cost of a commercial alternatives. Digital implementation enables advanced features such as complex effects processing, though real time performance constraints limit the processing time.

The report concludes that microcontroller based modular synthesizers are technically and economically possible. However, limitations in the cost analysis methodology and quality verification make it difficult to definitively assess commercial viability at scale.

Keywords: modular, synthesizer, eurorack, digital, sound synthesis, digital sound processing, embedded

Erkännanden

Stort tack till vår handledare Erik Sintorn, som guidat oss under projektets gång och varit mycket hjälpsam. Vi vill även tacka Arne Linde och Lars Norén för hjälp med beställningar till projektet.

H. Andersson, F. Bakken, R. Brenander, G. Hartwig, H. Johansson, A. Skoglund

Göteborg, Maj 2026

Begrepplista

Nedan listas begrepp som använts i rapporten med tillhörande förklaringar i alfabetisk ordning:

ADSR	Förkortning för Attack, Decay, Sustain, Release - parametrar som konfigurerar ett envelopps form. På svenska: stegring, sänkning, hållnivå och avklingning.
Cutoff	Engelska för brytfrekvens. Parameter i olika filtertyper som beskriver punkten då filtret börja dämpa frekvenser.
CV-signal	Förkortning för Control Voltage - en analog styrsignal som används för att modulera parametrar i en patch.
DAW	Digital Audio Workstation (digital ljudarbetsstation) är ett program för att spela in och bearbeta ljud.
Envelopp	Beskrivning av hur ljud eller modulering ändras över tid.
ENV-signal	UART-signal som används för att skicka envelopps-inställningar.
FX	Förkortning för effects, alltså effekter på svenska. Syftar på vidare manipulering som kan läggas på en ljudsignal. Reverb och distorsion är två exempel på effekter.
LFO	Förkortning för Low Frequency Oscillator - en oscillatormodul som genererar en CV-signal inom frekvensspannet 0.1 Hz - 20 Hz .
MIDI	Förkortning för Musical Instrument Digital Interface - ett digitalt kommunikationsprotokoll som används inom musikhårdvara.
ML3	16-polig kontakt som används för anslutning till bakplan.
Monofoni	Ljudkälla som bara kan generera en ton i taget.
Parafoni	Ljudkälla som kan generera en eller flera toner och spela dem samtidigt, men som senare i signalkedjan summeras och hanteras som en monofonisk ljudkälla.
Patch	Konfiguration av sammankopplade moduler i en modulär synthesizer.
Polyfoni	Ljudkälla som kan skapa flera toner och spela dem samtidigt.
Pitchbend	Vanligt förekommande reglage på MIDI-klaviatur. Justerar ljudets tonhöjd.
Rack	Chassi som används för att montera modulära syntsystém.
VCA	Förkortning för Voltage Controlled Amplifier - en förstärkarmodul vars amplitud styrs av en CV-signal.
VCO	Förkortning för Voltage Controlled Oscillator - en oscillatormodul vars tonhöjd styrs av en CV-signal.
Velocity	Engelska för hastighet. Ett värde som beskriver hur hårt eller snabbt en tangent aktiveras på en MIDI-kontroller.

Innehåll

Ordlista	x
Figurer	xvi
Tabeller	xvii
1 Inledning	1
1.1 Syfte	1
1.2 Mål	2
1.2.1 Kravspecifikation	2
1.3 Avgränsningar	2
2 Teori	4
2.1 Ljudteori	4
2.2 Musikteori	4
2.3 Ljudsyntes	5
2.3.1 Vågformer	5
2.3.2 Envelopp	6
2.3.3 Modulering	6
2.3.4 Ljudfilter	7
2.4 Digital signalbehandling	7
2.4.1 Från analoga till digitala signaler	7
2.4.2 Signaladdition	8
2.4.3 Digitala filter	8
2.4.4 Biquad-filter	8
2.5 Projektspecifik hårdvaruteori	10
2.5.1 PCB	10
2.5.2 Raspberry Pi Pico	10
2.6 Datakommunikationsprotokoll	10
2.6.1 UART	10
2.6.2 I ² C	11
2.6.3 I ² S	12
2.6.4 MIDI	12
2.7 En modulär synthesizer	13
2.7.1 Historia	13
2.7.2 Eurorack-standard	14
2.7.3 VCO – Spänningsstyrd oscillator	14

2.7.4	VCA – Spänningsstyrd förstärkare	15
2.7.5	LFO – Lågfrekvensoscillator	15
2.7.6	VCF – Spänningsstyrt filter	15
2.7.7	Ljudmixer	16
2.7.8	ADSR	16
2.7.9	En digital modular synthesizer	16
3	Metod	17
3.1	Implementering	17
3.1.1	Systemöversikt	17
3.1.2	Strömförsörjning	18
3.1.3	VCO	19
3.1.4	LFO	20
3.1.5	VCF	20
3.1.6	Mixer	21
3.1.7	ADSR	22
3.2	Konstruktionsprocess	22
3.2.1	Konstruktion av moduler	22
3.2.2	Konstruktion av bakplanet	25
3.3	Design	26
3.3.1	Fysisk design	27
3.3.2	Design av användargränssnitt	28
3.4	Metod för validering och evaluering av produkten	29
3.4.1	Hårdvarutestning	29
3.4.2	Verifiering mot kravspecifikation	29
4	Resultat	31
4.1	Kostnadsutvärdering	31
4.2	Syntens funktioner	32
4.2.1	Miljö	32
4.2.2	Prestanda	32
4.2.3	Hårdvarugränssnitt	32
4.2.4	Bakplan	33
4.2.5	MIDI-gränssnittet	33
4.2.6	VCO	33
4.2.7	VCF	34
4.2.8	LFO	35
4.2.9	Mixer	35
4.3	Utökade mål	35
5	Diskussion	36
5.1	Kommersiell prissättning	36
5.2	Tekniska val	36
5.3	Lärodomar och förbättringsmöjligheter	41
5.3.1	Hårdvara	41
5.3.2	Mjukvara	41
5.4	Felkällor	42

5.5	Samhälleliga och etiska aspekter	42
5.5.1	Etiska aspekter med metod	42
5.5.2	Etiska aspekter med resultat	43
6	Slutsats	44
	Referenser	45
A	Kravspecifikation	I
B	Elschewan	IV
C	Kostnad	VII

Figurer

2.1	Exemepl på fyra vågformer [21], CC BY-SA 3.0.	6
2.2	ADSR-parametrar [23], CC BY-SA 3.0.	7
2.3	Digital filtreringsprocess.	9
2.4	UART-signal [38], CC BY-SA 4.0.	11
2.5	Tidsdiagram över I ² C [40], public domain.	12
2.6	Tidsdiagram över I ² S [42], CC BY 3.0.	12
3.1	Systemöversikt med exempeluppkoppling.	17
3.2	Kopplingschema över strömförsörjningskortet.	19
3.3	En prototyp-syntkedja på kopplingsdäck.	23
3.4	Överblick av prototypkort.	23
3.5	Baskort med komponenter monterade.	25
3.6	Schema över bakplanskontaktens anslutningar.	26
3.7	Initial skiss av VCF.	27
3.8	3D-printat chassi med två modulpaneler fastmonterade.	28
4.1	Oscilloskopmätningar av de olika vågorna	33
4.2	Mätningar av de olika vågformerna gjorda med oscilloskop	34
B.1	Elschema revision 1.0	V
B.2	Elschema revision 2.0	VI

Tabeller

3.1	Koefficienter för olika filtertyper i en biquad-arkitekturen.	21
4.1	Kostnaden för kommersiella syntmoduler med låg kostnad hos <i>Thomann</i> [68].	31
4.2	Priset för projektets moduler jämfört med Behringers moduler med lägst kostnad på Thomann [68].	32
C.1	Strömförsörjning, MIDI-gränssnitt och Rack.	VII
C.2	VCO	VII
C.3	VCF	VIII
C.4	LFO	VIII
C.5	Mixer	VIII

1

Inledning

En modulär synt är ett elektroniskt musikinstrument där framställande, behandling och manipulation av en ljudsignal sker med hjälp av ett valfritt antal självständiga moduler [1]. Modulerna är kopplade via en gemensam strömförsörjning och är ofta fästa i ett *rack*, ett chassi utformat för att montera och organisera modulerna. Det som skiljer färdigbyggda analoga syntar från modulära är hur modulariteten tillåter användaren att själv välja vilka moduler som ska vara en del av instrumentet. I en färdigbyggd synt är modulerna redan valda och sammankopplade. Användaren kan därför inte påverka hur de separata enheterna ska se ut, vilket leder till mindre flexibilitet.

Många upplever en skillnad mellan digitala och analoga syntar [2]. Vissa professionella musiker och ljudingenjörer menar att analoga kretsar leder till ett mer robust ljud med varmare ton och mer önskvärda imperfektioner – såsom brus, mekaniskt sus vid hårdvarustörningar samt vibrato och svängningar i tonhöjd. Om defekterna förstärks för mycket kan det leda till ett oanvändbart och omusikaliskt ljud [3]. Ett vanligt problem med äldre analoga syntar är deras bristande förmåga att hålla sig stämde då komponenter påverkas i högre grad av temperaturändringar. Digitala syntar saknar denna svaghet och är därmed betydligt stabilare [4]. Vid granskning av en digital ljudsignal, exempelvis med hjälp av ett oscilloskop, är det möjligt att observera hur kretsen delar upp signalen i diskreta steg. En digital utsignal saknar den analoga färgning som en analog synt skapar när ljudet går igenom den fysiska hårdvaran.

Gemensamt för både analoga och digitala syntar är att hårdvaran är kostsam. Enstaka moduler kan kosta tusentals kronor och även de mest rudimentära modulära syntar kräver flertalet moduler för att mata ut användbart ljud [5]. Önskas det mer komplexa moduler och effekter kan prislappen överstiga tiotusentals kronor. Att bygga en modulär synt blir snabbt kostsamt för att åstadkomma meningsfullt experimenterande med intressanta moduler.

1.1 Syfte

Syftet med detta projekt är att undersöka hur kostnadseffektivt det är att använda mikrokontroller istället för diskreta analoga komponenter för att konstruera en modulär synt. Samtidigt bör synten behålla samma upplevelse och interaktion som kommersiella modulära syntar.

1.2 Mål

Målet är att skapa en digital synt där användaren själv kan välja vilka moduler som används. Modulerna ska producera signaler som kan skickas till andra moduler eller direkt omvandlas till ljudsignaler. Modulerna ska kunna styras med hjälp av fysiska reglage eller inmatning från andra moduler. Synten ska följa standardiserade fysikaliska dimensioner hos kommersiella moduler. Det skall finnas tillräckligt många och varierande moduler för att användaren ska kunna använda synten som ett musikaliskt instrument. Dessutom ska synten vara kostnadseffektiv i förhållande till motsvarande produkter på den kommersiella marknaden.

1.2.1 Kravspecifikation

I projektets kravspecifikation, som presenteras i appendix A, definieras de krav, **K**, som tillsammans specificerar projektets mål. I kravspecifikationen finns även kriterier betecknade med **Ö**. Dessa är endast önskemål och är alltså inte obligatoriska.

1.3 Avgränsningar

En modulär synt är öppen till en stor mängd moduler och funktioner. För att möjliggöra framställandet av en fungerande synt inom projektets tidsram, där låg kostnad prioriteras, avgränsas projektet enligt följande:

Begränsad moduluppsättning

Eftersom en modulär synt kan bestå av ett obegränsat antal moduler, måste moduluppsättningen begränsas. Moduler som ingår i projektets MVP, dess *Minsta livskraftiga produkt*, innefattar följande lista:

- MIDI-gränssnitt – Möjliggör kommunikation mellan synten och en extern enhet, t.ex ett keyboard.
- VCO - *Voltage-Controlled Oscillator* – Genererar en ljudvåg i det hörbara frekvensspektrumet.
- VCF - *Voltage-Controlled Filter* – Begränsar och påverkar framträdandet av en ljudsignals frekvensinnehåll beroende på vald filtertyp.
- LFO - *Low-Frequency Oscillator* – Genererar en lågfrekvent vågform i syfte att användas som moduleringsignal.
- Mixer – Kombinerar flera ljudsignaler till en ljudsignal.
- Bakplan – Tillgodoser andra moduler med strömförsörjning och synkronisering.

- DAC – D/A-omvandlare som konverterar en digital insignal till en analog utsignal.

Efter att MVP:n har färdigställts finns möjligheten att komplettera med fler moduler till synten. Dessa moduler tillför funktioner som introducerar fler möjligheter för experimenterande och uttryck hos användare.

Ingen analog ljudsyntes

Modulerna kommer att generera, skicka och bearbeta digitala signaler med hjälp av mikrokontroller. Detta ersätter metoden med analog signalhantering som de flesta kommersiella moduler använder. Istället för att använda operationsförstärkare och analog-digital-omvandlare kommer detta simuleras i mikrokontrollernas mjukvara.

Begränsad VCO-funktionalitet

VCO:n kommer ha fyra färdiga vågformer tillgängliga: fyrkants-, sinus-, triangel- och sågtandsvåg. Alternativen begränsas till dessa färdiga vågformer för att minska komplexiteten och efterlikna analoga system.

2

Teori

I detta kapitel presenteras den underliggande teorin för implementation och konstruktion av den modulära synten. Detta inkluderar de fysikaliska principerna för ljud, musikteori, matematiken inom ljudsyntes och digital signalhantering. Hårdvaran, hur olika syntmoduler fungerar och datakommunikationsprotokoll som används samt vad en modulär synt är i detalj förklaras också.

2.1 Ljudteori

Ljud uppstår genom att partiklar vibrerar i ett medium, exempelvis luft, kring sitt jämviktsläge och propagerar genom att partiklarna överför energi till varandra. Dessa vibrerande partiklar har en rörelseenergi och det är denna energi som, när den når örat, gör att man kan höra ljud [6]. På grund av rörelsemönstret som partiklar i vibration har brukar ljud beskrivas som en våg. Hastigheten som ljudvågen vibrerar i kallas frekvens vilket mäts i hertz (Hz). Ljudets amplitud är det som beskriver storleken på svängningen i vågen och ljudets styrka mäts i decibel (dB) [7].

2.2 Musikteori

Toner är regelbundna ljudvågor som uppstår genom vibrationer i föremål [8]. Toner har en tonhöjd vilket mäts i hertz, och en tonstyrka vilket mäts i decibel. En hög tonhöjd förknippas med att tonen upplevs som ”pipig” medan en låg tonhöjd innebär att ljudet har en mer ”bullrig” eller ”dov” karaktär. Tonstyrka beskriver hur stark amplituden på tonen är. Toner som upplevs som enkla eller rena består av några enstaka frekvenser. De består då av en primär grundton tillsammans med ett fåtal högre deltoner eller övertoner. Mer komplexa toner har ett större antal övertoner. Det som avgör hur en ton låter är dess frekvenser. De frekvenser som människan kan höra ligger inom spannet 20 Hz till 20 kHz [9].

I västerländsk musikteori är det vanligt att dela upp toner i oktaver [10]. En oktav är avståndet från en ton till nästa ton med samma namn som har högre eller lägre tonhöjd. I det liksvävande tonsystemet delas oktaven in i 12 toner, där varje ton representerar en specifik tonhöjd och avståndet mellan dem är ett halvtönssteg. Tonföljden C , $C\sharp$, D , $D\sharp$, E , F , $F\sharp$, G , $G\sharp$, A , $A\sharp$ och B är en hel oktav [11]. Tillsammans kombinerar man dessa toner inom en oktav till skolor, kombinationer av toner där intervallen anses skapa en harmonisk sammanhållning. Dur- och mollskalan förekommer ofta [12]. Utöver oktaven är kvint, sekund samt dur- och mollters

särskilt viktiga intervall [13]. Man brukar utgå från A4, alltså tonen A i oktav 4, som en standard för hur instrument ska vara stämda. Stämgaflar är vanligtvis konfigurerade efter att A4 har frekvensen 440 Hz [14].

Beroende på hur många röster eller toner som ett instrument kan producera samtidigt kategoriseras det annorlunda [15]. En enstämmig ljudkälla skapar endast ett ljud och kallas för *monofonisk*. Bleckblåsinstrument som trumpet är ett exempel på en monofonisk ljudkälla. Piano och gitarr däremot har möjligheten att spela flera stämmor samtidigt och kallas för *polyfoniska* instrument [16]. Detta gör att de kan bygga ackord och skapa harmonier med sig själv. Historiskt har det varit vanligast med monofoniska syntar då de varit betydligt billigare och mer portabla. Inom moderna syntar har det även uppkommit ett tredje begrepp: *parafoni* [17]. Parafoniska instrument kan ta emot flera noter samtidigt, men till skillnad från polyfoniska syntar delar stämmorna på en gemensam signalbehandling, genom exempelvis filter. Detta innebär att tonerna kan vara helt separerade i tonhöjd, men inte fullt oberoende i sin ljudformning senare i kedjan. Till skillnad från polyfoni, där varje röst har en självständig signalkedja, behandlas flera röster i en parafonisk synt ofta som en gemensam signal efter oscillatordelen, likt en monofonisk synt. Detta ger en form av begränsad polyfoni, vilket möjliggör mer komplexa ackord och spelbarhet utan att kräva lika mycket hårdvara.

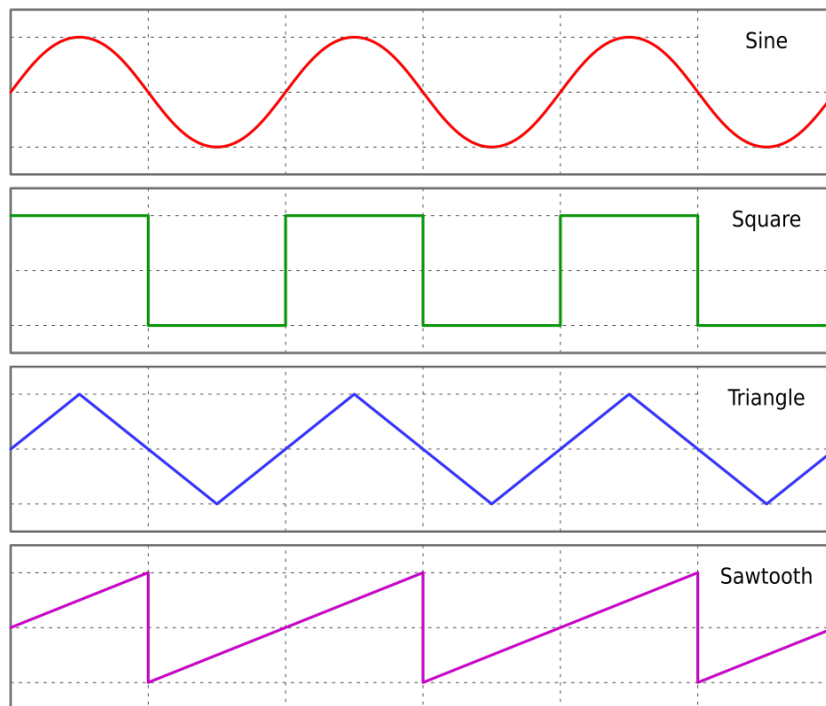
2.3 Ljudsyntes

Olika syntar utnyttjar olika typer av ljudsyntes [18]. En vanlig variant är subtraktiv ljudsyntes, som används i både digitala och analoga syntar. De primära modulerna i subtraktiv ljudsyntes är oscillator, filter och modulation. Instrumentet börjar med att oscillatorn genererar en komplex vågform som sedan formas med hjälp av filter och modulering. Den initialt komplexa tonen förvandlas till något nytt genom att frekvenser subtraheras från den. Förutom subtraktiv syntes finns även FM-syntes. FM-syntes baseras på komplexa matematiska beräkningar där olika vågformer modulerar varandra för att skapa ljud. Detta är vanligt i digitala syntar på grund av den beräkningskapacitet som krävs för realtidsbearbetning. [19].

2.3.1 Vågformer

Vågformer är sätt att visualisera hur en elektrisk signal eller ljudvåg förändras över tid [20]. Några exempel på vanliga vågformer är sinus-, triangel-, fyrkants- och såg-tandsvåg, vilka är avbildade i figur 2.1. Sinusvågen, som kan beskrivas matematiskt med sinusfunktionen, är den minst komplexa vågformen ur ett harmoniskt perspektiv. Den skapar få övertoner och har ett rent ljud med ett smalt frekvensspann. Detta kan vara användbart som grund för att sedan bygga vidare på ljudet med effekter i andra moduler. Användaren kan exempelvis koppla in en distorsionsmodul som introducerar övertoner till signalen. Triangelvågen liknar sinusvågen fast med mer branta och skarpa svängningar. Triangelvågen består av mer övertoner vilket gör att ljudet är mer annorlunda klang jämfört med sinusvågen. Fyrkantsvågen innehåller samma övertoner som triangelvågen förutom att övertonerna inte avtar eftersom

amplituden på vågen avtar drastiskt, istället för graduellt som i sinus- eller triangelvågen. Sågtandsvågen är en mer harmoniskt komplex vågform än sinusvågen. Den kan beskrivas som en linjärt ökande signal som återställs vid varje period. I motsats till sinusvågen har sågtandsvågen ett brett frekvensspann med fler övertoner. Den stora mängden frekvenser gör den användbar för att forma ljudet subtraktivt med filter och modulering.



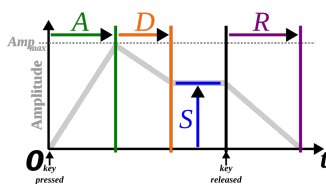
Figur 2.1: Exemepl på fyra vågformer [21], CC BY-SA 3.0.

2.3.2 Envelopp

Envelopp beskriver en kurva som kan appliceras som kontrollsignal på olika aspekter av ett ljud. Beroende på hur det appliceras kan det definiera hur ljudets klang och karaktär ändras över ett tidsspann [22]. Ett sätt att beskriva enveloppkurvan är med parametrarna ADSR, där varje bokstav är en förkortning för det engelska namnet på en av de fyra olika faserna i ljudets existens. **A** står för attack, eller stegring. **D** står för decay, eller sänkning. **S** står för sustain, eller hållnivå. Slutligen står **R** för release, eller avklingning. Hädanefter kommer de engelska termerna för ADSR användas. Detta kan ses i figur 2.2.

2.3.3 Modulering

Modulering beskriver hur moduler kan påverka olika parametrar i den modulära synten. I praktiska drag kan det beskrivas som att istället för att användaren vrider på en ratt för att ändra på en parameter, delegeras den uppgiften till en annan modul som gör detta automatiskt. Detta skapar möjlighet för förändring över tid i



Figur 2.2: ADSR-parametrar [23], CC BY-SA 3.0.

ljudet och kan skapa komplicerade och skiftande tonklanger. Vanliga moduleringskällor är *LFO* och *ADSR-envelopp* [24].

En moduleringskällas djup, det vill säga hur mycket den påverkar en viss parameter, kan vara *unipolär* eller *bipolär*. En unipolär källa modulerar enbart uppåt från parameterns nuvarande värde, medan en bipolär källa kan modulera både uppåt och nedåt. En bipolär LFO modulerad på VCO:ns tonhöjd till exempel ger ett vibrato som svänger symmetriskt runt grundtonen, medan en unipolär sådan hade istället ökat grundtonen i samma frekvens.

I en analog modulär synt sker *routing*, alltså styrning av signalflöder mellan moduler, enbart via fysiska patchkablar där varje koppling kräver en dedikerad port och ledare. I en digital synt kan istället routing implementeras mjukvarubaserat, vilket möjliggör en intern modulationsmatris där en källa kan kopplas till valfri parameter utan ytterligare hårdvara [25].

2.3.4 Ljudfilter

Även utanför ljudsyntes används filter för att kontrollera vilka frekvenser som en krets ska släppa igenom [26]. Detta kan innebära att ta bort oönskade defekter eller missljud som uppstår vid störningar. Inom ljudsyntes innebär ett ljudfilter manipulation och förändring av vilka frekvenser som ska släppas igenom. Filtret kan anta olika former beroende på vilket frekvensspann som ska påverkas.

2.4 Digital signalbehandling

Eftersom syftet med detta projekt är att ersätta en stor del av den analoga signalbehandlingen i moderna syntar med digital signalbehandling i mikrokontroller, kommer några av de metoder som används beskrivas.

2.4.1 Från analoga till digitala signaler

För att kunna behandla analoga signaler i digitala system måste signalen samplas, vilket omvandlar den analoga signalen till en diskret representation [27]. Ett digitalt system tar in en analog signal genom en analog-digital-omvandlare eller *ADC* [28]. Det finns även en motsvarande digital-analog-omvandlare eller *DAC* som används för att skicka ut en digital signal till ett analogt system, till exempel från en mobiltelefon till en högtalare. När en signal samplas är det viktigt att den samplas i en frekvens

som är dubbelt så stor som signalens högsta frekvens. Den analoga signalens högsta frekvens kallas Nyquistfrekvensen och är hälften av samplingfrekvensen.

2.4.2 Signaladdition

Olika toner har olika frekvens, vilket betyder att deras vågformer skiljer sig åt. När flera toner spelas samtidigt behöver deras ljudsignaler adderas. I digital signalbehandling sker detta genom att amplituderna för respektive signal summeras vid varje samlingspunkt [20]. Resultatet blir en ny vågform som kombinerar de två ingående signalerna. Denna metod möjliggör polyfoni, alltså förmågan för ett digitalt system att spela flera toner samtidigt.

2.4.3 Digitala filter

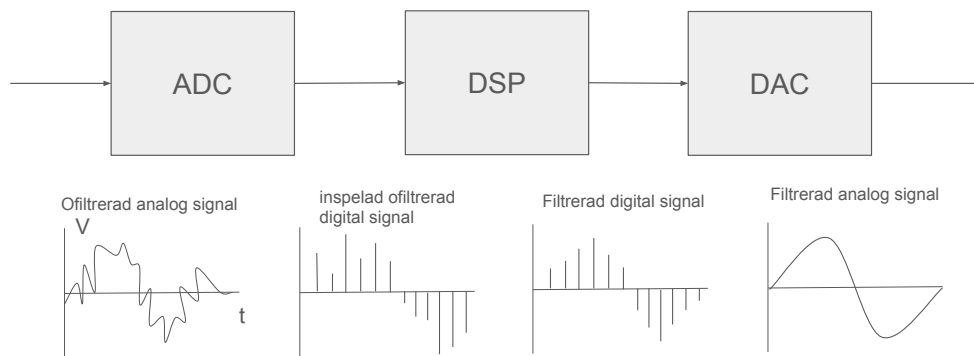
Ett filters funktion är att ta bort eller dämpa delar av en signal. I många fall behöver data separeras, exempelvis ta bort störningar och brus i ljud. Filter gör det möjligt att restaurera signaler som deformerats på något vis. Musik som spelats in med dålig utrustning kan filtreras för att bättre representera hur det lät vid inspelning [29]. Det är även vanligt att använda filter för att extrahera användbara eller mer önskvärda delar av signalen, exempelvis genom att isolera data på ett specifikt frekvensspann [30].

Det finns både analoga och digitala filter. Analoga använder elektroniska kretsar byggda från resistorer, kondensatorer och operationsförstärkare för att uppnå önskad filtreringseffekt. Detta sorts filter är snabbt, billigt och har ett brett dynamiskt omfång till kostnad av prestanda och stabilitet. I stark motsats har digitala filter så pass bra prestanda att det sällan är något man behöver ha i åtanke vid implementation [29]. Implementation sker direkt via matematiska uträkningar på samplade värden av signalen. Detta sker via digitala processorer antingen genom en dator eller ett specialiserat DSP (Digital Signal Processor) chipp, vilket möjliggör effektiv signalbehandling i realtid eller efter bearbetning. Figur 2.3 visar en simpel digital filtreringsprocess av en analog signal genom en ADC, processor och en DAC.

Två viktiga parametrar i hur filtret påverkar signalen är *cutoff* och *resonance* [31]. Cutoff, gränsvärde eller brytfrekvens, bestämmer spannet där filtret ska börja påverka. I ett lågpassfilter bestämmer cutoff hur mycket av de höga frekvenserna i insignalen som ska skäras bort. Resonance, eller resonans, syftar på ett smalt band av frekvenser som betonas i en filtertyp.

2.4.4 Biquad-filter

Ett biquad-filter syftar på ett sorts digitalt filter som används inom signalbehandling för att forma frekvensinnehållet i en signal [32]. Biquad är en förkortning för biquadratic, eller bikvadratisk, vilket refererar till att dess överföringsfunktion är förhållandet mellan två kvadratiske funktioner. Den är rekursiv och använder tidigare indata och använder de i nästa uträkning för att bilda en dynamisk återkopplingsloop. Med minnet av tidigare insignaler och utsignaler kombinerar den värdena för



Figur 2.3: Digital filtreringsprocess.

att filtrera ut ej önskvärda värden. Detta konstruerar ett *IIR*, *Infinite Impulse Response* eller oändligt impulssvar, som utgår från en liten insignal, en impuls, som genom rekursion påverkar utgången under lång tid **iir-filter**. Uträkningen för att räkna ut filterformen sker för varje sample i ingångsbufferten. Den använder sig av 5 parametrar, $a_0, a_1, a_2, b_0, b_1, b_2$ för att konfigurera formen på filtret. Genom att välja värden på dessa koefficienterna kan lowpass-, bandpass-, highpass- och notchfilter effektivt skapas.

Ekvation 2.1 visar ekvationerna i dess direkta form. Täljaren beskriver hur insignalen påverkar utsignalen. Nämnaren beskriver hur ekvationens återkoppling, dvs. hur tidigare iterationer påverkar funktionen. Ekvation 2.2 beskriver ekvationen i normaliserad form där varje värde divideras med a_0 så att den första termen i nämnaren blir 1. Detta gör det lättare att skriva om till en differensfunktion i tidsdomänen. Ekvation 2.3 är den praktiska formen av ekvationen som används i programkoden. Den tar in aktuella och tidigare insignaler samt tidigare utsignaler och kombinerar de med koefficienter.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \quad (2.1)$$

$$H(z) = \frac{\left(\frac{b_0}{a_0}\right) + \left(\frac{b_1}{a_0}\right) z^{-1} + \left(\frac{b_2}{a_0}\right) z^{-2}}{1 + \left(\frac{a_1}{a_0}\right) z^{-1} + \left(\frac{a_2}{a_0}\right) z^{-2}} \quad (2.2)$$

$$y[n] = \left(\frac{b_0}{a_0}\right) x[n] + \left(\frac{b_1}{a_0}\right) x[n-1] + \left(\frac{b_2}{a_0}\right) x[n-2] - \left(\frac{a_1}{a_0}\right) y[n-1] - \left(\frac{a_2}{a_0}\right) y[n-2] \quad (2.3)$$

2.5 Projektspecifik hårdvaruteori

Målet med projektet involverar att producera en hårdvaruprodukt, detta avsnittet förklarar de viktigaste hårdvarurelaterade tekniska detaljerna som används i projektet.

2.5.1 PCB

Printed Circuit Board, förkortat PCB, är grundläggande för komponenter inom modern elektronik. Ett PCB, eller kretskort, består av ett eller flera lager isolerande substratmaterial, vanligtvis FR4, som är ett glasförstärkt epoxilaminat på vilket ledande kopparspår läggs på för att bilda elektriska förbindelser mellan komponenterna [33]. Den huvudsakliga funktionen av ett kretskort är att ersätta behovet av fristående kablar mellan komponenter. Genom att integrera alla elektriska förbindelser i ett enda kort blir det mer pålitligt.

De ledande spåren på ett kretskort definieras under designfasen med användning av specifik mjukvara för kretskortsdesign. De resulterande designfilerna används direkt i automatiserade tillverkningsprocesser, vilket ger möjlighet till produktion av identiska kort. Kretskorten går att beställa både med komponenter förinstallerade och utan komponenter.

Från ett tillverkningsperspektiv har kretskort fördelar vad gäller kostnad och skalbarhet. När den initiala designen och tillverkningsupstarten väl är genomförd är styckpriset per kretskort lågt, eftersom hela tillverkningsprocessen kan vara helt automatiserad. Kretskortet kan även testas i fabrik innan den skickas till kunden [34].

2.5.2 Raspberry Pi Pico

Raspberry Pi Pico är en lågkostnadsmikrokontroller som den modulära synten kommer att baseras på. En anledning till att den valdes är att processorn är tvåkärnig, vilket möjliggör ökad flexibilitet i programkoden. En annan anledning är att det finns tre 12-bitars ADC-portar som kan användas för läsning av analoga signaler. Slutligen har den två PIO-block för Programmable I/O, som är ett hårdvarunära sätt att simulera olika kommunikationsprotokoll [35].

2.6 Datakommunikationsprotokoll

För att sammankoppla syntens olika moduler används datakommunikationsprotokoll till att möjliggöra kommunikation mellan modulerna.

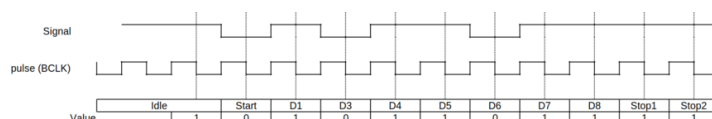
2.6.1 UART

Universal Asynchronous Receiver-Transmitter, UART, är ett av de äldsta och vanligaste seriella kommunikationsprotokollen inom inbyggda system [36]. Kommunika-

tionen bygger på en asynkron grund vilket innebär att det inte finns en gemensam klocka, utan istället en gemensam och förutbestämd överföringshastighet (baudhastighet).

Protokollet använder sig av två ledare: TX (Transmit) och RX (Receive). Sändaren skickar data på TX-tråden, som är kopplad till mottagarens RX-tråd, och vice versa. Detta gör UART till ett P2P (Peer-To-Peer), punkt till punkt protokoll. Viloläget på sändarens ledare är en logiskt hög signal, och kommunikationen inleds med en startbit som är en logiskt låg signal, vilket signalerar till mottagaren att en ny byte är på väg [37].

Den vanligaste UART-ramkonfigurationen är 8 databitar, ingen paritet och en stoppbit. Eftersom det saknas en klocksignal samplar mottagaren varje databit i mitten av dess tidsfönster, baserat på den överenskomna baudhastighet, för att minimera risken för fel.



Figur 2.4: UART-signal [38], CC BY-SA 4.0.

Baudhastighet anger antalet symboler per sekund och är en kritisk parameter som måste stämma överens på båda sidor. Vanliga värden är exempelvis 9 600 och 115 200 baud. Om baudhastigheterna skiljer sig för mycket åt kommer mottagaren att feltolka bitarna, vilket leder till korrupt data eller kommunikationsfel.

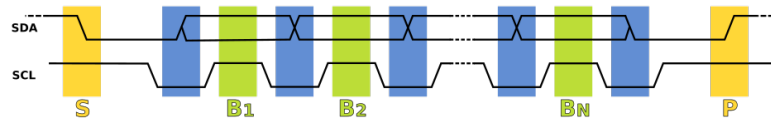
2.6.2 I²C

I²C (Inter-Integrated Circuit) är ett seriellt kommunikationsprotokoll som utvecklades på 1980-talet. Det är designat för kortdistanskommunikation mellan integrerade kretsar på samma kretskort [39].

Protokollet använder två ledare: SDA (Serial Data) som bär datainformation, och SCL (Serial Clock) som synkroniserar överföringen. Båda linjerna använder så kallad *open drain*-logik, vilket innebär att signalerna är logiskt höga som basläge, och komponenterna kan endast ändra signalerna till logiskt låga. Opåverkade signaler hålls höga genom pull-up-motstånd, vilket gör att flera enheter kan dela samma buss utan att kortsluta varandra.

Kommunikationen bygger på ett controller/responder-förhållande, där kontrollern alltid initierar kommunikation, genererar klocksignalen och anger vilken responder den vill tala med. Varje responder lyssnar och svarar endast när den adresseras, och genererar aldrig klocksignalen själv. Varje responder har en unik 7-bitarsadress, vilket tillåter upp till 128 enheter på samma buss, även om några adresser är reserverade.

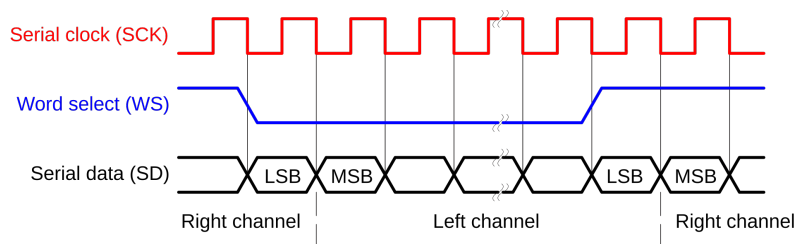
En typisk överföring börjar med att kontrollern signalerar ett starttillstånd genom att dra SDA låg medan SCL är hög. Därefter skickas responders adress följt av en bit som anger om det är en läs- eller skrivoperation. Respondern bekräftar med en ACK-signal och data börjar överföras i 8-bitars paket, där varje byte följs av en bekräftelse. Överföringen avslutas med en stop-condition där kontrollern drar SDA hög medan SCL är hög.



Figur 2.5: Tidsdiagram över I²C [40], public domain.

2.6.3 I²S

I²S eller *Inter-IC Sound* är ett seriellt kommunikationsprotokoll som skapades på 1980-talet för användning inom ljudsystem [41]. Protokollet möjliggör stereoljud genom att använda sig av två kanaler för klockor och en kanal för data, vilket kan ses i figur 2.6. Den första klocksignalen, Serial Clock (SCK), bestämmer när bitarna ska skickas. Detta görs på *rising edge*, alltså när SCK går från 0 till 1. Ett annat namn för SCK är bit clock (BCLK). Den andra klocksignalen, Word Select (WS eller WSEL), bestämmer vilken ljudkanal som bitarna ska skickas till. Det är denna klocka som möjliggör stereoljud, genom att bitarna skickas till vänstra ljudkanalen när WS är 0 och högra när WS är 1. Ett annat namn för WS är left-right clock (LRCLK). Serial Data (SD) är kanalen som innehåller ljudet, och synkroniseras till SCK och WS för att bestämma när och till vilken ljudkanal ljudet ska skickas.



Figur 2.6: Tidsdiagram över I²S [42], CC BY 3.0.

2.6.4 MIDI

MIDI 1.0 (Musical Instrument Digital Interface) är ett protokoll för digital överföring av musikdata mellan instrument och datorer. Det utvecklades 1983 som ett standardprotokoll för digitala instrument.

På hårdvarusidan använder MIDI oftast en 5-polig DIN-kontakt, där varje enhet kan ha upp till tre portar: MIDI IN för att ta emot data, MIDI OUT för att skicka data, och MIDI THRU för att vidarebefordra inkommande signaler till nästa enhet

i kedjan [43]. MIDI IN-porten använder galvanisk avskiljning, mer specifikt en optokopplare, som består av en ljuskälla och en fototransistor, för att förhindra direkt elektrisk kontakt mellan ingången och utgången. Detta skyddar mot två problem, varav det första är en så kallad *ground loop*, som uppstår när två enheter ansluts till olika punkter i elnätet och deras jordpunkter har en spänningsskillnad vilket orsakar brus eller andra störningar i ljudsignalen. Det andra problemet som optokopplaren skyddar mot är spänningsspikar, som annars skulle kunna skada ansluten utrustning.

MIDI är ett seriellt protokoll som bygger på UART som underliggande protokoll, med en baudhastighet på 31 250 baud, och meddelanden som är åtta bitar långa med två ytterligare bitar för start och stop. Protokollet överför instruktioner om hur ljud ska spelas istället för en direkt ljudvåg.

De vanligaste meddelandena är tre bytes långa [44]. I dessa meddelanden är den första byten en statusbyte, där den mest signifikanta biten (bit 7) är satt till 1 [45]. De fyra mest signifikanta bitarna anger meddelandetyp och de fyra minst signifikanta bitarna anger kanalnummer. De två följande bytetarna är databytes, där bit 7 istället är 0, vilket ger 128 möjliga värden (0–127) till informationen. Den första databyte anger vanligen vilken ton som meddelandet handlar om och den andra anger tonens styrka (velocity).

2.7 En modulär synthesizer

En modulär synthesizer, eller modulär synt, är uppbyggd av separata moduler där varje modul ansvarar för en specifik aspekt av generering eller manipulation av ljud. Till skillnad från fasta syntar där signalflödet är fördefinierat, kan en modulär synt kopplas om fritt, vilket möjliggör för stor flexibilitet i ljuddesign. Exempelvis kan användaren själv välja till eller bort moduler samt ändra signalflödet i realtid.

En specifik hopkoppling av moduler i en syntkedja kallas för en *patch*, och att konfigurera ett sådant flöde kallas för att *patcha*. Eftersom signalflödet kan kopplas fritt får varje patch en unik karaktär, vilket gör att samma uppsättning moduler kan producera ett brett spektrum av tonklanger.

2.7.1 Historia

Utvecklingen inom elektronik och massproduktion har gjort elektroniska instrument betydligt mer tillgängliga. Från syntens begynnelse i början av 60-talet till dess kulturella kulmen under 80-talets syntpop- och New-Wave genombrott [46] gick den från ett avant-garde-instrument, främst använt inom experimentell musik och akademiska kretsar, till ett etablerat och brett använt verktyg inom populärmusik [47]. Tillverkare började skifta sitt fokus mot att producera syntar som var både mer kostnadseffektiva och portabla [48]. Digitala kretsar och komponenter har blivit billigare att tillverka, vilket har bidragit till att syntar inte längre enbart används inom akademiska miljöer. Istället har de blivit ett verktyg som även amatörmusiker

kan ta del av. Idag är det enklare än någonsin för människor att börja använda och experimentera med syntar [49].

2.7.2 Eurorack-standarden

Eurorack är ett standardformat för syntmoduler som definierar fysiska dimensioner och spänningsvärden [50]. Standarden gör det möjligt att koppla ihop olika moduler från olika tillverkare. Eurorack skapar möjligheten att producera moduler som garanterat kommer passa med en mängd olika produkter. De fysiska dimensionerna inom Eurorack-standarden är en anledning till att formatet blivit utbrett. Alla moduler följer samma måttsystem och monteringspunkter. Det gör att tillverkare kan designa moduler som passar i vilket kompatibelt chassi som helst. Eurorack är en modulär arkitektur vilket möjliggör att enskilda moduler kan kommunicera effektivt med varandra. Genom standardiserade fysikaliska mått kan 3.5mm-patchkablar kopplas universellt mellan olika moduler, oavsett tillverkare eller funktion.

Genom en CV-signal, control-voltage eller styrspänning, möjliggörs kommunikation mellan de olika modulerna. De gör det möjligt för en moduleringskälla att skicka information till en parameter genom spänning. Därav bygger hela systemet på analoga (eller digitalt genererade) spänningssignaler. Det existerar inte ett tvingat regelverk för spänningsvärden vilket kan leda till variationer mellan tillverkare. Standarden för spänningskonvertering är 1V / Octave vilket menar på hur tonhöjden ändras beroende på ingående spänningssignal [51]. För varje 1 V-ökning i oscillatoren går dess tonhöjd upp en oktav. Det är detta som möjliggör kvantiserad tonhöjdshantering och gör det möjligt att precist spela noter. Spänningsvärdet för att spela tonen A4, det vill säga en frekvens på 440 Hz, är alltid detsamma och är konsekvent.

2.7.3 VCO – Spänningsstyrd oscillator

Ljudgenereringen i en synt börjar i *Voltage-Controlled Oscillator*, förkortat *VCO*. Modulen tar emot ingående signal som bestämmer på vilken frekvens tonen ska genereras, exempelvis via en MIDI-ingång eller CV-signal, och matar ut en vågform som sedan kan manipuleras av andra moduler. Vanliga typer av vågformer som en VCO kan generera är sinus-, triangel-, fyrkant- och sågtandsvåg [52]. Vanligt förekommande är också pulsvågor och brusgeneratorer som producerar brus istället för en omedelbart musikalisk ton. Brusgenerering kan vara implementerat i en VCO modul men kan också vara sin egen fristående modul [53].

En digital VCO genererar ljud genom att beräkna samplar vid en fast samplingsfrekvens. Vågformen lagras som en *wavetable*, en förberäknad array av amplitudvärden för en hel period. Oscillatoren stegar igenom tabellen med en hastighet som bestäms av en önskad frekvens, där högre frekvens innebär större steg per sample. Vid parafoni tilldelas varje röst ett eget index i arrayen som stegar framåt oberoende av övriga röster. Sampelvärden för varje röst läses ur arrayen vid respektive index och adderas numeriskt vilket ger en gemensam utsignal utan extra hårdvara. Det analoga alternativet kräver istället en komplett oscillator per röst [54].

Frekvensmodulering, förkortat *FM*, är en typ av ljudsyntes som bygger på digitala processer. FM är baserad kring två olika oscillatorer. Den ena kallas för en modulator som varierar frekvensen av en annan oscillator, kallad en *bärvåg*. Processen är icke linjär och kan skapa ett brett spektrum av ljud med få beståndsdelar [55] [56] [57] .

2.7.4 VCA – Spänningsstyrd förstärkare

En *VCA*, *Voltage-Controlled Amplifier* eller spänningsstyrd förstärkare, kontrollerar amplituden av ljudsignalen via en elektrisk spänning. Modulen styr amplituden av syntens utsignal och kan användas för att kontrollera hur volym ändras över tid. Modulen tar emot en insignal, den ton som VCO:n har skapat, samt en styrsignal för att styra amplituden.

2.7.5 LFO – Lågfrekvensoscillator

En *LFO*, *Low-Frequency Oscillator* eller Lågfrekvensoscillator, används som en moduleringskälla i en synt. Tonen som oscillatoren genererar är så låg att den inte är hörbar för människor. Den kan skapa samma sorts vågformer som en vanlig VCO, till exempel sinus, triangel och fyrkant. Signalen den skickar kan istället användas som en styrsignal för att manipulera andra parametrar i synten. Parametern förändras kontinuerligt i takt med LFO:n.

Ett användningssätt för LFO:n är att skapa en vibrato-effekt i en patch. Detta åstadkoms genom att LFO:n skickar en sinus-signal till frekvensparametern i en VCO. Tonhöjden går upp och ner i takt med styrsignalen som LFO:n genererar, vilket uppfattas som vibrato. Ett annat sätt att använda LFO:n är att koppla in den i VCA:n för att skapa en tremolo-effekt där volymen går upp och ner i takt med styrsignalen.

2.7.6 VCF – Spänningsstyrt filter

Signalen som genereras av VCO:n kan sändas till andra moduler för vidare hantering. Om användaren önskar manipulera tonen och frekvenserna i en vågform kan VCO:n kopplas till en *VCF*, *Voltage-Controlled Filter*. Ett ljudfilter tar in en ljudsignal och applicerar en frekvensbaserad förändring. En VCF kan förstärka, trimma eller dämpa frekvenser inom ett valt område. Genom att justera *cutoff*, den frekvens då filtertypen börjar påverka, och *resonans* kan användaren ändra över vilket frekvensspann filtret ska vara aktivt. En harmoniskt komplex vågform kan med hjälp av en VCF förenklas genom att valda frekvensområden filtreras bort [58]. Detta möjliggör en stor mängd ljud för användaren att skulptera utöver de som genereras direkt av VCO:n. VCF-modulen har ett antal färdiga filtertyper för användaren att konfigurera [59]. Vanliga typer är *lågpass*, *högpas* och *bandpass*.

2.7.7 Ljudmixer

Mixermodulen kombinerar flera ljudkällor till en och samma utgång, vilket är användbart för att kombinera ljud från flera oscillatorer. Varje ingång har individuellt ställbar amplitud. Det existerar en visuell komponent, exempelvis en lysdiodsramp eller skärm, som visar användaren hur stark ljudsignalen på en ingång är. Det kan vara användbart för att få inblick i hur ljudet kommer låta vid utmatning till hörlurar eller högtalare [22].

2.7.8 ADSR

ADSR-modulen är en implementering av enveloppgenerering som används för att ge ljud karaktär och förändring över tid. Liksom LFO:n är ADSR-modulen ett användbart verktyg för att forma komplexa tonklanger. Exempelvis kan ADSR vara användbart vid konstruktionen av ljud som efterliknar akustiska instrument. Ett akustiskt piano har en kort attack men beroende på hur länge tangenterna hålls ner kan ljudets release och sustain ändras. Med en ADSR kan dessa attribut återskapas. Via en styrsignal till en VCA kan ljudets volymparameter förändras så att ljudet klingar ut. Styrsignalen kan skickas till VCF-modulen för att kontrollera patchens cutoff och resonans. Med en kort decay och låg sustain kan användaren skapa staccatoliknande toner, det vill säga korta toner med hög initial intensitet som inte ringer ut.

2.7.9 En digital modulär synthesizer

Det är viktigt att notera hur en digital modulär synthesizer skiljer sig från en analog modulär synthesizer. Eftersom den digitala motsvarigheten har vuxit fram ur den analoga är begreppen något diffusa. CV, Control Voltage, i en digital synt är i själva verket digitala signaler snarare än analoga spänningar. På liknande vis är till exempel VCO (Voltage-Controlled Oscillator) och VCF (Voltage-Controlled Filter) strikt sett missvisande i ett digitalt sammanhang. Beteckningarna används dock fortsatt av tradition och för att bibehålla kompatibilitet med den etablerade terminologin.

Utöver terminologisk förskjutning medför en digital implementation funktionella skillnader jämfört med analoga system. Parafoni kan uppnås utan extra hårdvara genom parallella beräkningar av röster som summeras digitalt. FM-syntes reduceras till en aritmetisk operation per sampel, medan analog FM kräver precis och temperaturstabil hårdvara [60]. Mjukvarubaserad signalstyrning möjliggör koppling mellan moduleringskällor och parametrar utan ytterligare hårdvara.

Digitala signaler saknar dock en naturlig klangfärgning som uppstår i analoga kretsar, vilket vissa användare eftertraktar [60]. Dessutom kräver digital kommunikation mellan moduler precis klocksignalsynkronisering, vilket gör systemet känsligt för störningar i dataöverföringen.

3

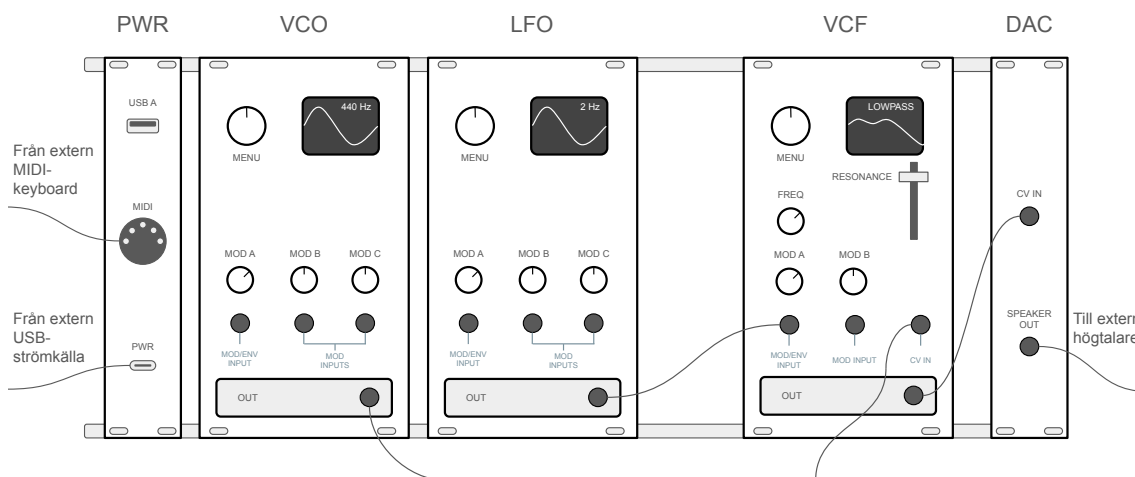
Metod

3.1 Implementering

Detta avsnitt ger insikt i hur den modulära synthen implementerats mjukvarumässigt och hårdvarumässigt. Detta görs genom en systemöversikt och detaljerad förklaring av de specifika modulerna.

3.1.1 Systemöversikt

Figur 3.1 visar hur de olika modulerna kan kopplas samman i en *patch*. På framsidan kan användaren, precis som på en kommersiell modulär synt, koppla *patchkablar* mellan de olika modulerna för att påverka ljudet. Skillnaden med denna synt är att dessa kablar överför digitala och inte analoga signaler. När det önskade ljudet har tagits fram måste därför utgångssignalen kopplas till en DAC-modul, som omvandlar den digitala signalen till en analog signal, för att sedan kunna koppla ljudet till högtalare.



Figur 3.1: Systemöversikt med exempeluppkoppling.

Alla moduler strömförsörjs från bakplanet, vilket är en 16-polig intern kontakt med ström- och dataöverföring. Ytterligare information om kontakten och vilka protokoll som används finns i avsnitt 3.1.2 och 3.2.2.

De digitala ljudsignalerna som skickas i patchkablarna motsvarar CV-signaler i en kommersiell modulär synt, och använder I²S-protokollet för överföring. Detta in-

nebär att tre ledare behövs för att skicka eller läsa ljud, men två av ledarna är klocksignaler och skickas på bakplanet. Därmed behövs endast en ledare i patchkabeln för att skicka I²S-data, och användaren får en liknande upplevelse till en kommersiell modulär synt. Hädanefter kommer dessa digitala representationer av CV-signaler benämnas synonymt med vanliga, analoga CV-signaler.

Förutom I²S används även UART i systemets patchkablar. Detta är möjligt eftersom patchkablarna är stereokablar, och består av totalt tre ledare istället för två som vanligtvis används i modulära syntar. Anledningen till att skicka UART är för att möjliggöra överföring av annan data som inte motsvarar CV-signaler. Ett exempel på detta är en ADSR-kurva som appliceras internt i en modul innan ljudet skickas ut. På detta sätt kan ADSR-kurvan appliceras oberoende på varje ton som spelas, utan att påverka tidigare toner. Hädanefter kommer dessa UART-signaler benämnas *ENV-signaler*.

Alla moduler är byggda på samma kretskortsdesign, det så kallade *baskortet*, som förklaras i ytterligare detalj i avsnitt 3.2.1. Detta innebär att alla moduler har samma interna anslutningsmöjligheter, med samma pinout-konfiguration, och en stor del av programkoden blir därmed gemensam. Två gemensamma programbibliotek har tagits fram: *synthutil* och *synthorn_i2s*.

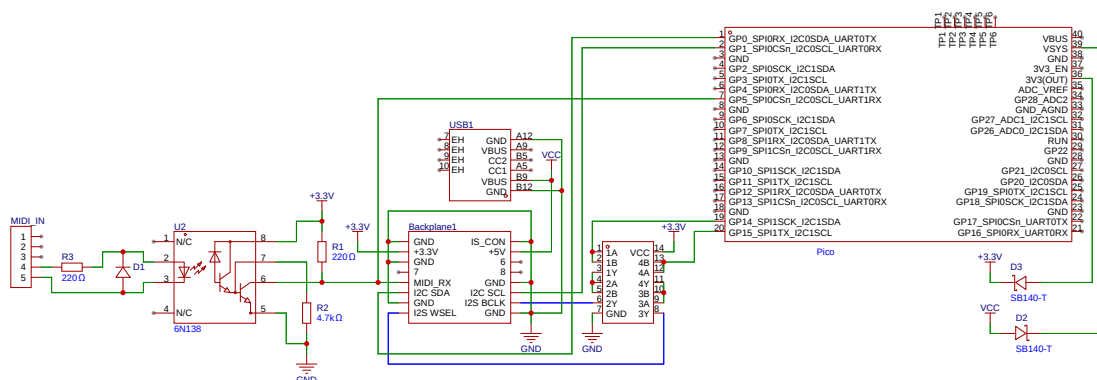
I *synthutil*-biblioteket finns gemensam funktionalitet såsom initiering av GPIO-portar, tolkning av MIDI-data och konvertering till musikaliska frekvenser, tolkning av multiplexeringångar samt funktioner för att initiera periodiska avbrott. I²S-biblioteket används för att skicka och läsa I²S-signaler med gemensam klocka från bakplanskontakten. I biblioteket finns stöd för att läsa upp till 4 ingångar samtidigt, och skriva till 1 utgång.

Eftersom I²S-protokollet skickas med hög frekvens behöver ljudprocesseringen ske på kort tid. På alla moduler används en samplingsfrekvens på 48 000 Hz, som skickas med 16-bitars upplösning på två kanaler (stereo). Detta innebär att klockfrekvensen är $48000 \times 16 \times 2 = 1\,536\,000$ Hz eller 1,536 MHz. För att underlätta processering av annan stor data, såsom att uppdatera OLED-skärmen, används därför den andra processorkärnan för mindre tidskritiska uppgifter.

3.1.2 Strömförsörjning

Strömförsörjningsmodulen, som namnet antyder, strömförsörjer alla moduler via modulernas bakplanskontakt och måste vara inkopplad för att strömförsörja systemet. Kopplingsschemat för modulen illustreras i figur 3.2. Systemets strömmatning kommer från en USB-C-port som kan kopplas till valfri strömkälla. USB-spänningen (5 V) kopplas direkt till bakplanskontakten, och en Raspberry Pi Pico genererar en 3,3 V-spänning som också är kopplad till bakplanet. Båda spänningsmatningar är kopplade via dioder för att förhindra hårdvaruproblem om systemet strömförsörjs på flera platser samtidigt, till exempel vid omprogrammering av en modul.

På modulen finns även en DIN-kontakt, som används för att läsa MIDI-meddelanden.



Figur 3.2: Kopplingschema över strömförsörjningskortet.

Även denna är kopplad till bakplanet, via en optokopplare som ger galvanisk avskiljning mellan DIN-kontakten och resten av synten. Det finns också en USB-A-kontakt för strömförsörjning av kringutrustning, som en MIDI-keyboard med USB-strömmatning.

Mikrokontrollern som används ansvarar för att generera klocksignalerna BCLK och WSEL för I²S-protokollet. Dessa går först till en schmitttrigger för att stabilisera signalen, och sedan vidare till bakplanskontakten där klockorna kan läsas av övriga moduler. Mikrokontrollern är även inkopplad till bakplanskontakten med I²C-protokollet, vilket möjliggör kommunikation mellan systemets moduler över bakplanet, till exempel för att överföra patchinställningar till andra moduler.

3.1.3 VCO

VCO-modulen bygger på ett gemensamt bibliotek, *synthutil*, som förbereder wavetable-arrayer och MIDI-stegvärden vid uppstart. Biblioteket genererar bland annat de 4 vågformerna: sinus, triangel, sågtand och fyrkant, lagrade som arrayer med 2048 amplitudvärden. För varje MIDI-nots respektive frekvens förberäknas steglängden mellan varje sample, alltså hur snabbt oscillatorn ska stega igenom wavetable-arrayen. Högre noter ger snabbare genomlöpning och därmed högre frekvens i utsignalen.

Modulen utnyttjar RP2040:ns tvåkärniga arkitektur. Kärna 0 ansvarar för realtids-ljudrendering och kärna 1 för gränssnitt och parameterstyrning. För varje aktiv MIDI-not sparas en egen position i wavetable-arrayen. För varje sampel flyttas positionen framåt med ett notspecifikt steg och ett amplitudvärde läses ur arrayen som skalas med notens MIDI-velocity. Alla aktiva noters värden summeras sen till ett gemensamt samplingsvärde som skickas ut via I^sS. Detta upprepas 48 000 gånger per sekund och möjliggör simultan generering för upp till 128 toner.

Internt har VCO:n två oberoende oscillatorer, A och B, som vardera konfigureras med egen vågform, tonhöjd och eventuell vågformsmanipulering. En vipp-potentiometer väljer vilken röst som är aktiv för redigering. Utgångssignalen styrs av 4 routinglä-

gen: enbart A, A och B parallellt, enbart B samt B modulerar A i serie. I parallellt läge summeras A:s och B:s samplar och divideras med två.

I läge B modulerar A implementeras *fasmodulering*. B:s aktuella samplevärde tolkas istället som en modulator och förskjuter A:s position i wavetable-arrayen för varje sampel. Detta möjliggör unika vågformsmoduleringar som till exempel FM.

Ytterligare har VCO:en en vågformsmodulering kallad *flip*, inverterar amplituden hos samplas upp till en konfigurerbar brytpunkt i wavetable-arrayen, liken skapar en asymetrisk vågform med förändrat övertonsinnehåll.

CV-routing möjliggör att externa signaler från två CV-ingångar samt en enveloppingång kopplas till valfri destination: tonhöjd, flip, FM-djup eller vågform. Djupet för respektive källa ställs in med dedikerad potentiometer för modulationsdjup.

3.1.4 LFO

LFO-modulen producerar en ljudvåg precis som VCO:n och gör detta på samma sätt, med funktionerna från synthutil. Den stora skillnaden från VCO:n är att ljudsignalen som LFO:n producerar har lägre frekvens än det som är hörbart för människor, mellan 0 Hz och 20 Hz.

3.1.5 VCF

Filtermodulen bygger på en implementation av ett digitalt biquad filter. Med ett redan existerande c-bibliotek [61] var det effektivt att sammanställa biquad-strukturen, och påbörja arbetet på olika filtertyper. Den färdiga funktionaliteten användes för att skapa ett eget bibliotek, *filterutil*, där funktioner för att manipulera filtret och tillhörande uträkningar definieras. Där initialiseras hur filtret ska se ut samt tillhandagges hur bearbetning av ljudsignalen ska ske. Formen på biquad-ekvationen byggs dessutom för att få till önskad filtertyp, cutoff och resonans.

Primära funktionen för att applicera filtret på insignalen är *filter_process_sample*. Den tar in ett 16-bitars värde som normaliseras till ett flyttal på spannet [-1, 1]. Detta motsvarar I²S-signalen som skickas från syntens VCO-modul. Det normaliserade värdet används sedan i funktionen *BQ_process* där biquad-filtret appliceras. Efter att processen är klar skickar modulen vidare den filtrerade signalen till nästa i kedjan.

Valet av koefficienter är gjort utefter Robert Bristow-Johnsons *Audio EQ Cookbook* [62][63], där han sammanställer ett antal implementationer av biquad-filter. Värdena är valda efter modulära prototyper som digitaliserats med hjälp av bilinjär avbildning, en process som bevarar filterstrukturens stabilitet genom att avbilda från det analoga s-planet till det digitala z-planet. Det vill säga, det går från en representation byggd på en den verkliga, kontinuerliga tidsdomänen till en baserad kring den komplexa enhetscirkeln [64].

Datastrukturen *FilterState* definieras, vilken används för att hålla värden relevanta till filtrets funktionalitet. Det innehåller variabler för filtertyp, cutoff och resonans

samt för att se om det skett en ändring i parametrar sedan tidigare bearbetning av indata. Detta gör att om det inte har skett en förändring i filtret behöver processen inte göra nya uträkningar för filtrets form vilket sparar prestanda. Tabell 3.1 visar koefficienterna för att skapa några av filtertyperna.

Tabell 3.1: Koefficienter för olika filtertyper i en biquad-arkitekturen.

Filtertyp	a_0	a_1	a_2	b_0	b_1	b_2
Lågpas	$1 + \alpha$	$-2 \cos \omega_0$	$1 - \alpha$	$\frac{1 - \cos \omega_0}{2}$	$1 - \cos \omega_0$	$\frac{1 - \cos \omega_0}{2}$
Högpas	$1 + \alpha$	$-2 \cos \omega_0$	$1 - \alpha$	$\frac{1 + \cos \omega_0}{2}$	$-(1 + \cos \omega_0)$	$\frac{1 + \cos \omega_0}{2}$
Bandpas	$1 + \alpha$	$-2 \cos \omega_0$	$1 - \alpha$	α	0	$-\alpha$

Filterutil används i programkoden som skickas till pico-enheten. I den skapas ett *filterState*-objekt som håller all inbördes information som behövs. *Filterstate*-objektet används när i initialiseringsfunktionen och sätter begynnelsevärden för filtertyp, cutoff och resonans. Via analoga potentiometrar kopplade till pico-enheten kan avläses ändringar i värdet för cutoff och resonans. Val av filtertyp sker via en meny i en digital display som är inkopplad i pico-enheten. Med displayen visualiseras filtret och hur det påverkas med ändringar i filtertyp, cutoff och resonans. En rotationsenkoder är inkopplad för att styra modulens displaymeny. Det är en elektronisk komponent som kan vridas i diskreta steg för menynavigering samt tryckas in för val.

3.1.6 Mixer

Mixermodulen viktigaste funktion är att hantera flera ljudsignaler samtidigt samt normalisera och kombinera dem till en gemensam ljudsignal. Detta implementerades genom att expandera I²S-biblioteket *synthorn_i2s* förmåga att kunna fånga flera samtidiga ljudsignaler. Eftersom en Raspberry Pi Pico har fyra tillståndsmaskiner i varje PIO-block var det möjligt för oss att köra upp till fyra instanser av signalinsamling parallellt. Modifierade av det tidigare signalinsamlingsprogrammet gjordes om till att iterera över varje aktiv tillståndsmaskin och fånga motsvarande signal. De insamlade signalerna placerades därefter i ett fält, där indexet motsvarade den kanal från vilken ljudsignalen hade sitt ursprung.

Efter mixermodulen har fångat ljudsignalerna påbörjas processen för volymhantering och normalisering. Detta görs genom att iterera över fältet med ljudsignaler och hantera varje signal individuellt. Till varje kanal finns en motsvarande potentiometer vars värde hämtas genom en ADC-ingång på PICO:n. För att systemet ska stödja upp till fyra ljudsignaler, trots att det endast finns tre ADC-ingångar, används en multiplexer. Därför utökades *syntutil*-biblioteket med förmågan att kunna läsa värden från en multiplexer beroende på kanal. Potentiometrarna är kopplade till kanal 0–3 på multiplexern, och deras värden används för att styra motsvarande index i fältet med ljudsignaler som ska modifieras. Potentiometrarna ger ett värde i intervallet 0–4095 som multipliceras med ljudsignalen vilket styr ljudsignalens amplitud. Därefter utförs en högerskiftning med 11 bitar. Detta används för att normalisera

resultatet och hålla signalnivåerna inom ett rimligt intervall. Eftersom flera signaler adderas samtidigt finns det en risk att summan överskrider maximalstorleken som en ljudsignal får ha vilket kan leda till distorsion. Därför begränsas värdet på ljudsignalens amplitud inom ett intervall på -32768 till 32767.

3.1.7 ADSR

ADSR, eller envelopp-generator, är en av syntens två moduleringskällor, tillsammans med LFO:n. Till skillnad från LFO-modulen skickar ADSR-modulen inte ut en I²S-signal för att modulera olika parametrar. Istället bygger den på en UART-implementation. Detta leder till att implementation och initialisering av moduler måste ske annorlunda. Moduler behöver veta både om det kommer en I²S-signal eller om det är en UART. Det gör att envelopp-inställningarna kan appliceras internt i modulen.

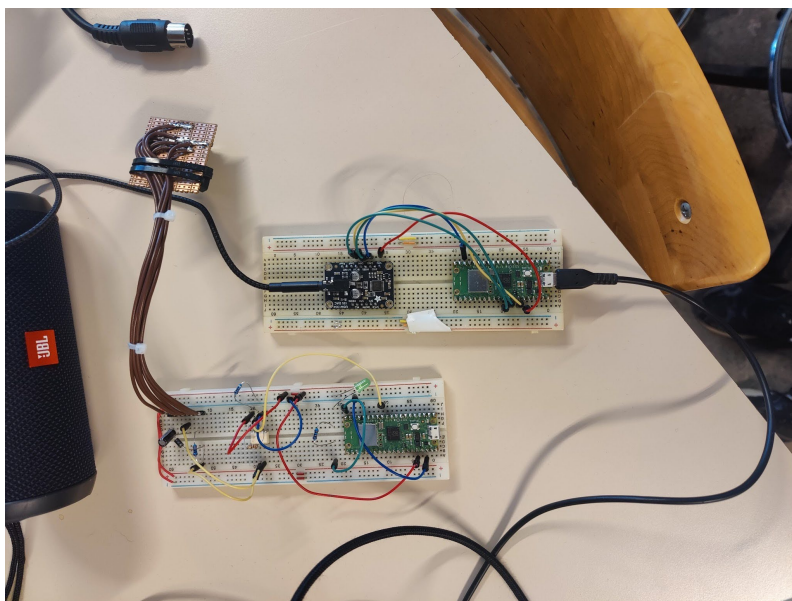
ADSR-modulen har fyra parametrar som användaren kan justera: attack, decay, sustain och release. Varje MIDI-signal (0-127) justeras och hanteras individuellt och har stöd för polyfoni. För attackparametern går amplituden från 0 till max över ett tidsspänn. Decay minskar från maxvärdet till sustain-nivån. Sustain lägger en konstant nivå som spelar så länge tangenten är nedtryckt. Release börjar på sustain-nivån och avtar till noll när tangenten släppts. Implementationen använder sig av tre arrayer. Den första innehåller information om ADSR-formen och dess aktuella position i ADSR-kurvan på nedtryckt tangent. Den andra lagrar den nuvarande amplituden för ljudet. Den tredje representerar *velocity*, det vill säga hur hårt användaren trycker ner på tangenten. Vid bearbetning av ljudsignalen multipliceras den med ADSR-kurvans amplitud och de aktiva tangenterna summeras.

3.2 Konstruktionsprocess

I detta avsnitt presenteras konstruktionsprocessen som användes under projektet, för både syntmoduler och bakplan.

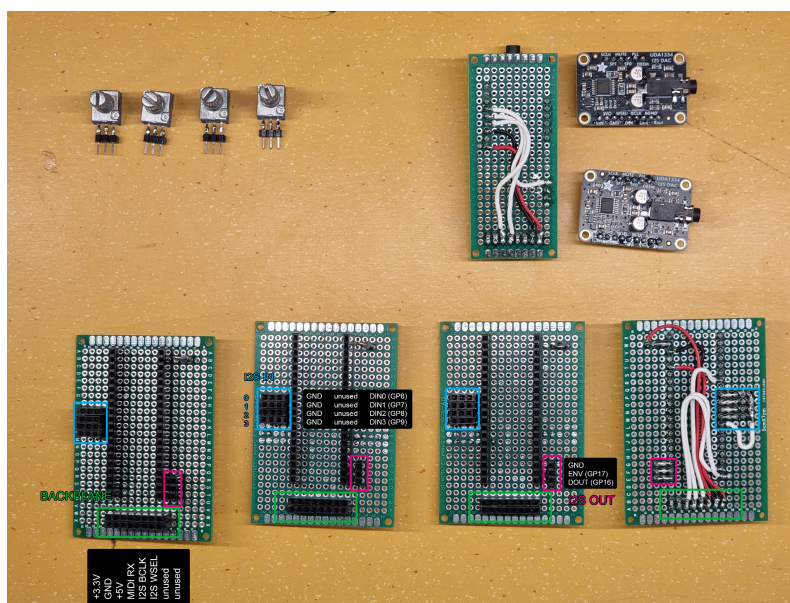
3.2.1 Konstruktion av moduler

Inledningsvis byggdes alla moduler på kopplingsdäck, då det var ett snabbt sätt att koppla samman många komponenter. Ett exempel på detta illustreras i figur 3.3. Programkoden kunde itereras kontinuerligt och det var enkelt att utföra tester med olika kopplingar. Det blev däremot uppenbart att analoga signaler var instabila, och kopplingarna som helhet var ömtåliga, speciellt när kopplingsdäcken förflyttades in och ut från förvaringsskåp.



Figur 3.3: En prototyp-syntkedja på kopplingsdäck.

Allt eftersom kopplingarna blev mer komplexa behövdes en mer robust lösning, och det var då prototypkort togs fram. Prototypkort menar på kretskort där komponenterna har manuellt sammankopplats utan förutbestämda banor och kan ses i figur 3.4. Dessa ger en något sämre flexibilitet men möjliggör betydligt bättre pålitlighet och fysisk hållbarhet. Eftersom prototypkorten skulle användas under en stor del av utvecklingsprocessen och är en mer permanent lösning, var det viktigt att testningen underlättades så mycket som möjligt. Därför utvecklades ett baskort som alla moduler utgick ifrån, och ett elschema togs fram för att underlätta tillverkningen av kortet, vilket illustreras i figur B.1 i bilaga B.



Figur 3.4: Överblick av prototypkort.

Tanken med baskortet var att ta fram en grund som är gemensam för nästan alla moduler i synten. Alla moduler utnyttjar inte all hårdvara på kortet, men testningen underlättades av att alla moduler använder samma pinout-konfiguration och att korten var utbytbara när problem uppstod. Ett baskort kunde bytas ut mot ett likadant för att avgöra om problemet var hårdvarurelaterat, och därmed finna orsaken till problemet på kortare tid.

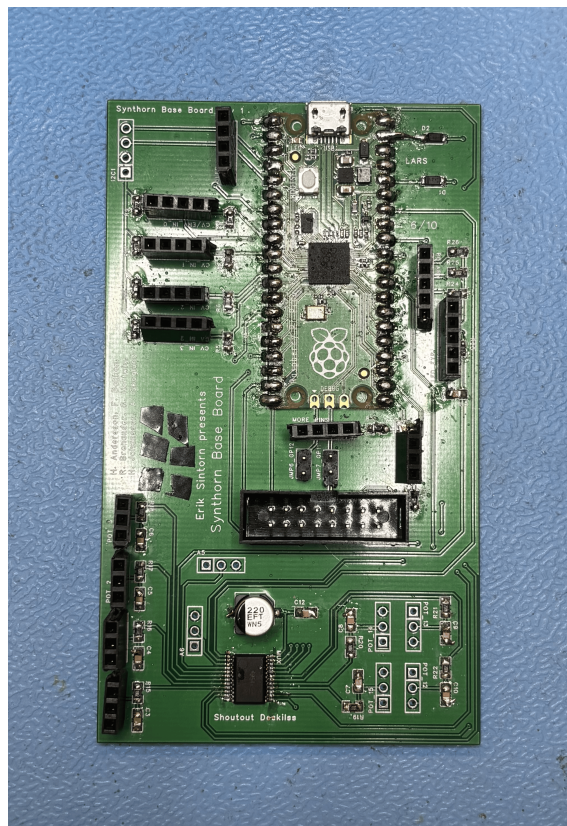
Baskortet reducerade också tiden som behövdes för att ta fram ritningar och felsöka kretskortsdesignen eftersom detta bara behövde göras en gång. Om det istället funnits olika kretskort till varje modul, och samma felsökningstid hade behövts till varje kort, hade mindre tid funnits över till annat.

På baskortet är mikrokontrollern och multiplexern alltid inkopplade, och kontakter finns för inkoppling till övriga moduler och annan hårdvara. Det finns även en kontakt för anslutning till bakplanet, vilket beskrivs i mer detalj i avsnitt 3.2.2. Varje baskort stöder anslutning till en rotationsenkoder, upp till fyra potentiometrar samt en OLED-bildskärm. Det finns även en kontakt för en “LED Expansion Board”, som kan kopplas till ett skiftregister för att styra lysdioder, vilket kommer användas på mixermodulen. För att vara beredd på framtida behov finns tre kanaler lediga på multiplexern som används för att läsa potentiometrarna, och OLED-bildskärmens I²C-buss kan utökas med fler moduler utan adressproblem, eftersom den är intern för modulen.

Kortet har 5 kontakter med beteckningen *CV* eller *CV/ENV*, varav 1 är utgång och 4 är ingångar, som senare kommer att kopplas till uttagen för patchkablar på frampanelen av modulen. Kontakterna med beteckningen *CV* skickar endast en digital CV-signal, men kontakterna med beteckningen *CV/ENV* skickar två datasignaler. Detta är möjligt eftersom patchkablarna som används är gjorda för stereoljud. En digital CV-signal skickas i den första kanalen och UART skickas i den andra, därav beteckningen *CV/ENV*. Detta innebär att kabeln som i vanliga fall endast skickar en CV-signal, nu kan användas till annan kommunikation, men fortfarande kan kopplas på samma sätt för användaren. Ett exempel på när annan kommunikation är önskvärd är ADSR-moduler som skickar envelopp-inställningar till VCO:n som applicerar ADSR internt innan ljudet skickas ut. När flera toner spelas samtidigt möjliggör detta att envelopp-inställningarna kan appliceras individuellt och få en polyfonisk signalkedja istället för en parafonisk.

Slutligen tillverkades ett kretskort, som illustreras i figur 3.5, med baskortsritningen som utgångspunkt. Elschemat uppdaterades först till en andra revision, som illustreras i figur B.2 i bilaga B. En skillnad från prototypkorten är att några av kontakterna har ändrats för att försvåra felkoppling, då de tidigare kontakterna var snarlika och opolariserade och därmed kunde kopplas i fel port. Bakplanskontakten expanderades från 8 till 16 pinnar för att möjliggöra utökad funktionalitet och täcka eventuella framtida behov. Det är denna kontakt som beskrivs i ytterligare detalj i avsnitt 3.2.2. Även multiplexern ändrades för att få 16 istället för 8 analoga ingångar, på grund av ökat behov. Det fanns även två oanvända GPIO-pinnar som via lödpunkter kan

kopplas till en separat kontakt, eller bryggas till bakplanet, beroende på framtida behov. För att minska eventuella emitterade och absorberade störningar har det nya kretskortet har korta banor vid de högfrekventa I²S-signalerna.



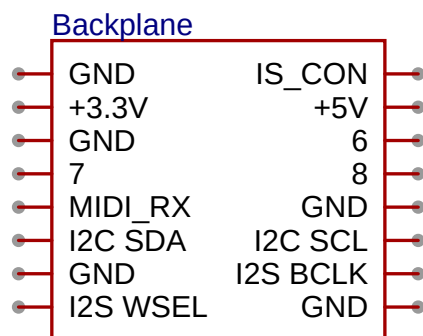
Figur 3.5: Baskort med komponenter monterade.

Förutom dessa ändringar var stora delar av elschemat likt det första schemat som användes för prototypkorten. Detta innebar att det räckte med en kretskortsritning för basmodulen, vilket minskade produktionstiden för systemet. Under tidiga utvecklingsstadier användes även simuleringsverktyget Wokwi [65] för att testa och utveckla mjukvara utan tillgång till fysisk hårdvara.

3.2.2 Konstruktion av bakplanet

För att strömförsörja modulerna skapades ett så kallat bakplan, vilket är en kontakt som sitter på baksidan av varje modul och som internt kopplas till ett gemensamt kretskort. I Eurorack-standarden, som nämndes i avsnitt 2.7.2, finns det redan en sådan anslutning, men eftersom modulerna behöver kunna kommunicera över bakplanet kunde denna anslutning inte användas. Därför togs en ny bakplanskontakt fram, under namnet *ML3*, vars pinout-konfiguration skiljer sig från den Eurorack-standarden har.

Den nya konfigurationen, som illustreras i figur 3.6, har stöd för systemets spänningar (3,3 och 5 V) samt all datakommunikation som behövs mellan modulerna. En av



Figur 3.6: Schema över bakplanskontaktens anslutningar.

de viktigaste anslutningarna är klockorna för I²S-protokollet: BCLK och WSEL. Dessa klockor finns på bakplanet för att underlätta ljudprocesseringen när ljudet behöver buffras, då alla moduler synkroniserar till samma klocka och signalerna kommer fram samtidigt. Fördelen med detta är att färre GPIO-pins används på varje mikrokontroller. Detta är speciellt viktigt på mixern, som annars hade behövt ta emot tre signaler för varje ingång (DATA, BCLK och WSEL), vilket leder till 12 GPIO-pins för att läsa 4 signaler. Mikrokontrollern måste då synkronisera till alla 8 klockor samtidigt, vilket är krävande. När modulen sedan ska generera utgångssignalen behövs ytterligare 3 GPIO-pins, och då finns det inte många GPIO-pins över till annan hårdvara.

För att generera dessa klockor finns en mikrokontroller på bakplanet som agerar som bakplans-kontroller. Det är även denna mikrokontroller som ska möjliggöra detektering och visualisering av modulerna när de har installerats. Det går även att använda denna I²C-buss för att kommunicera mellan moduler.

Ett annat kommunikationsprotokoll som skickas på bakplanet är MIDI. Denna anslutning är direkt kopplad till MIDI-ingången på framsidan av synten och eftersom den finns på bakplanet kan den läsas av alla moduler. Detta möjliggör styrning av ljudparametrar såsom vågform, cutoff-frekvens och resonans från en MIDI-keyboard som har justerbara reglage, om användaren önskar den möjligheten.

3.3 Design

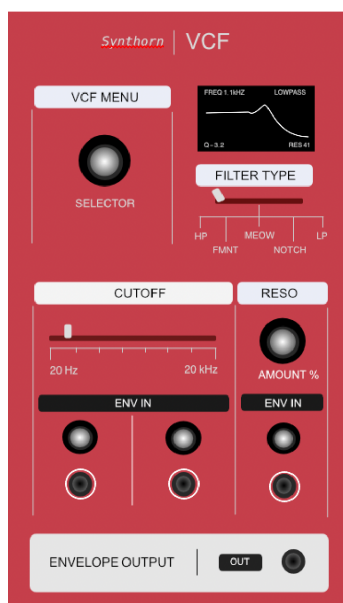
Designarbetet utgår från de krav och mål som presenterades i avsnitt 1.2 samt kravspecifikationen i bilaga A. Särskilt har fokuset lagts på att skapa ett system som är kompatibelt, användarvänligt och kostnadseffektivt, utan att kompromissa funktionaliteten.

3.3.1 Fysisk design

De fysiska aspekterna av systemet designades utifrån det centrala målet att efterlikna etablerade standarder inom modulära syntar, mer specifikt Eurorack-standarden. Detta innebär att modulerna är dimensionerade och anpassade för att vara kompatibla med Eurorack-chassin samt kunna samverka med andra Eurorack-moduler (3:e kravet i bilaga A).

För att minimera tillverkningskostnader har samtliga frontpaneler och chassikomponenter tillverkats med 3D-printing. Detta möjliggör snabb prototyputveckling och iteration av designen utan behov av extern tillverkning. Material som PLA och PETG har använts då de erbjuder en god balans mellan kostnad, hållbarhet och tillverkningsbarhet.

Frontpanelerna grovskissades i *Affinity Photo* [66] och därefter, i prototypstadiet, vidareutvecklades de med CAD i *Autodesk Fusion 360* [67]. I de initiala skisserna låg fokus på att placera reglage, kontakter och visuella element på ett sätt som gör det intuitivt att förstå signalflödet samt hur användaren kan manipulera detta.



Figur 3.7: Initial skiss av VCF.

Patchpunkter (CV/ENV) är placerade i nederkant för att minimera kabeldragning över reglage, medan interaktiva komponenter såsom potentiometrar och en rotationsenkoder är placerade i den övre delen av modulen.

Vid val av komponenter och konstruktion har kostnad varit avgörande faktor. Ytterligare har det varit väsentligt att välja komponenter som ger en “analog känsla”. Standardiserade och lättillgängliga komponenter har prioriterats, och baskortet, som nämnts i avsnitt 3.2.1, har använts som grund för samtliga moduler. Detta minskar både produktionskostnad och komplexitet samtidigt som det möjliggör snabbare utveckling och enklare felsökning.

Vidare har den fysiska konstruktionen anpassats för att vara robust nog för upprepad användning, samtidigt som det är enkelt att montera och demontera systemet.



Figur 3.8: 3D-printat chassi med två modulpaneler fastmonterade.

3.3.2 Design av användargränssnitt

Användargränssnittet har utformats med fokus på intuitiv interaktion och tydlighet för användaren. Särskild vikt har lagts på att göra systemet lätt att förstå, även för användare utan tidigare erfarenhet av modulära syntar. Placeringen av reglage och patchpunkter har utformats för att spegla signalflödet genom modulen, vilket gör det enklare för användaren att förstå hur ljudet genereras och manipuleras. Genom att separera patchpunkter och interaktiva komponenter minskas visuell komplexitet.

De mer komplexa modulerna, såsom VCO och VCF, kombinerar fysiska reglage med en OLED-bildskärm för direkt kontroll samt visuell återkoppling. En rotationsenkoder används för menynavigering, vilket möjliggör ett kompakt gränssnitt där flera

parametrar kan hanteras utan att kräva separata fysiska reglage. Detta minskar behovet av potentiometrar och andra fysiska komponenter, vilket i sin tur reducerar både komponentkostnad och kretskortsutrymme. Lösningen ligger därmed i linje med projektets mål att utveckla en kostnadseffektiv modulär synt. Designvalet möjliggör samtidigt framtida utökning av funktionalitet genom mjukvarubaserade gränssnitt.

Designen strävar efter att kombinera fördelarna med digital signalbehandling och traditionell analog interaktion. Den digitala implementationen möjliggör hög flexibilitet och komplexitet i signalhanteringen, medan det fysiska gränssnittet behåller en intuitiv och direkt användarupplevelse. Designen strävar därmed efter att balansera funktionalitet och enkelhet, samtidigt som den uppmuntrar till experimenterande, vilket är en central del av användarupplevelsen i modulära syntar.

3.4 Metod för validering och evaluering av produkten

Under konstruktionens gång har produkten validerats genom en kombination av hårdvarutestning och acceptanstestning. Acceptanströskeln för dessa tester har förhållandevis mot en marknadsredo produkt varit låg, då kvaliteten varit begränsad i prototypstadiet.

Utöver detta evaluerades produktens efterlevnad av kraven för kostnad och kvalitet när prototypen var klar.

3.4.1 Hårdvarutestning

Testning av systemets analoga utgång utförs med oscilloskop för att verifiera att ljudet reproducerar hela det hörbara frekvensområdet (20 Hz - 20 kHz) och att DAC-modulen hanterar signaler med minst 44.1 kHz samplingsfrekvens.

Testning med oscilloskop har dessutom använts vid felsökning under konstruktionens gång. Detta har varit särskilt hjälpsamt för att utreda var i en syntkedja fel i signalen uppstår.

Testning av fördröjning i systemet från MIDI-signal till genererad ton utförs genom en DAW, *digital audio workstation*, exempelvis *Ableton Live*. I programmet upptas signalen genom ett externt ljudkort och därefter undersöks fördröjningstiden. Från denna subtraheras ljudkortets och programvarans interna signalfördröjning för att få systemets fördröjning.

3.4.2 Verifiering mot kravspecifikation

För att verifiera att modulerna och systemet uppnår alla krav och eventuellt önskemål från kravspecifikationen i bilaga A, har en övergripande strategi för testning applicerats, där varje krav kopplas till en eller flera verifieringsmetoder. Kravspe-

3. Metod

cifikationen nämner endast vad systemet och modulerna ska uppnå, inte hur dessa krav och önskemål ska verifieras. Därför implementerades olika metoder för att testa under projektets gång beroende på kravet.

4

Resultat

4.1 Kostnadsutvärdering

För att bilda en uppfattning av vad låg kostnad innebär gällande kommersiella modulära syntar, har ett exempel på en modulär synt med låg kostnad sammanställts. Detta har gjorts genom att söka efter de modulerna med lägst kostnad hos musikåterförsäljaren *Thomann* [68] och sammanställt kostnaderna. Listan domineras av moduler av *Behringer*, ett företag vars marknadsmodell är att producera moduler så prisvänliga som möjligt [69].

Kostnadsutvärderingen gjordes genom en summering av alla komponentkostnader för varje modul. Vid inköpsplanering av komponenter till produkten avvägdes kostnaden mot komponentens väsentlighet.

Denna marknadsanalys, som visas i tabellen nedan, är ett försök att identifiera en av de billigaste möjliga syntarna med jämförbara funktioner som projektets synt.

Tabell 4.1: Kostnaden för kommersiella syntmoduler med låg kostnad hos *Thomann* [68].

Modul	Pris (SEK)	Namn
Rack	545	<i>Behringer Eurorack 104</i>
VCO	433	<i>Behringer 2600-VCO</i>
VCA	240	<i>Behringer 902 VCA</i>
LFO	415	<i>Behringer 150 Ring Mod/Noise/S&H/LFO</i>
Filter	314	<i>Behringer 904A VC Low Pass Filter</i>
Mixer	325	<i>Behringer CP3A-M Mixer</i>
Total	2272	

Tabellen nedan visar modulkostnaden för projektets synt. En prislista där kostnaden per modul visas i bilaga C. Kostnaden för projektets synt är 80% av kostnaden för exempelsynten från *Behringer* ($1823/2272 = 0,8$).

Tabell 4.2: Priset för projektets moduler jämfört med Behringers moduler med lägst kostnad på Thomann [68].

Modul	Egen (SEK)	Thomann (SEK)
MIDI/ Strömförsörjningskort/ Rack	183	545
VCO	471,6	433
VCA	-	240
VCF	478,7	314
LFO	344,1	415
Mixer	345,7	325
Total	1823	2272

4.2 Syntens funktioner

I detta kapitel utvärderas de tekniska aspekter som produkten uppfyller. Denna utvärdering utgår från kravspecifikation vilket presenteras i avsnitt 1.2.1 och går att hitta i sin helhet i appendix A. Utvärderingen bygger på en kombination av experimentella mätningar och objektiva observationer.

4.2.1 Miljö

Efter att systemet testats i normal inomhusmiljö utan märkbara störningar eller artefakter anses synthen vara fullt användbar i rumstemperatur.

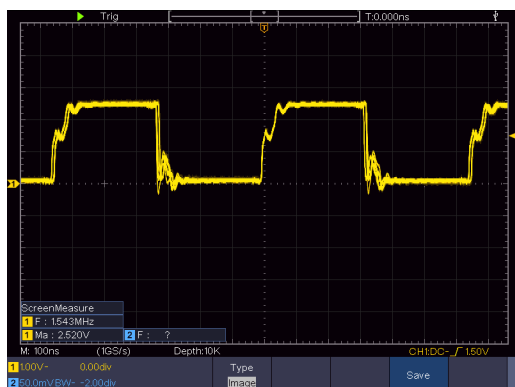
4.2.2 Prestanda

Mikrokontrollerna är programmerade att använda en samplingsfrekvens på 48 000 Hz, vilket motsvarar DVD-kvalitet. Det har verifierats genom att kontrollera klocksignalernas frekvens på oscilloskop, som illustreras i figur 4.1a. Den uppmätta frekvensen är enligt figuren 1,543 MHz, vilket beror på samplingsfrekvensen 48 000 Hz och ljudupplösningen som är 16 bitar. Dessa skickas i stereo (två ljudkanaler) vilket ger en teoretisk klockfrekvens på $48000 \cdot 16 \cdot 2 = 1\,536\,000$ Hz. Mätningen visar alltså på en felmarginal som är mindre än 0,5%.

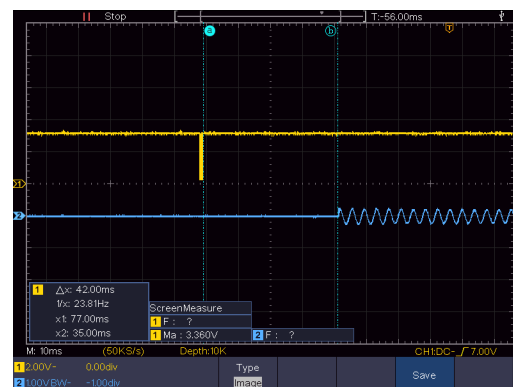
Beroende på modulens användning och behov justeras buffertstorleken. Exempelvis behöver inte mixern lika stor buffertstorlek som filtermodulen då den inte gör någon krävande ljudsignalbehandling. Detta innebär att filtret medför mer latens i systemet än mixern. Vid en buffertstorlek på 256 värden tillkommer $256/48000 \approx 0.0053$ s eller 5.3 ms. Efter att ha mätt mellan optokopplarens utgång och DAC:ens utgång uppmättes en latens på 42 ms med alla moduler inkopplade (Se figur 4.1b).

4.2.3 Hårdvarugränsnitt

Genom 3.5mm-kablar går det att skicka I²S-signaler mellan olika moduler. Genom ett extra stift på uttags-komponenterna är det möjligt att identifiera om det är en



(a) Frekvensen kan avläsas till 1.543 MHz (varje ruta är 100 ns).



(b) Mätning av latens genom hela kedjan.

Figur 4.1: Oscilloskopmätningar av de olika vågorna

kabel inkopplad eller inte. Systemet kan ta emot MIDI-signaler genom en DIN-input som sedan propageras vidare i systemet till alla moduler. För att utmata ljudsignal finns det en DAC-modul där användaren kan ansluta en 3.5mm-kabel till högtalare eller hörlurar.

4.2.4 Bakplan

Bakplanet strömförsörjer samtliga moduler i kedjan via en kabel. Via samma kabel skickas även MIDI-information och klocksignaler för I²S-protokollet.

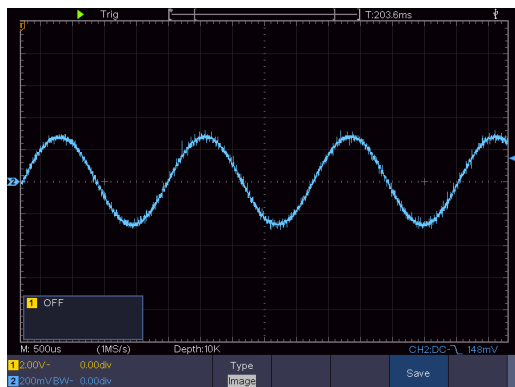
4.2.5 MIDI-gränssnittet

MIDI-gränssnittet är hårdvarumässigt gjort som en del av bakplanet, och mjukvarumässigt implementerat som en del av *synthutil*-biblioteket som körs på alla moduler som behöver läsa MIDI. Gränssnittet kan läsa och tolka hur hårt användaren aktiverat en tangent vid en given tidpunkt. Det kan även bedöma positionen på pitchbend och övriga reglage som modulation.

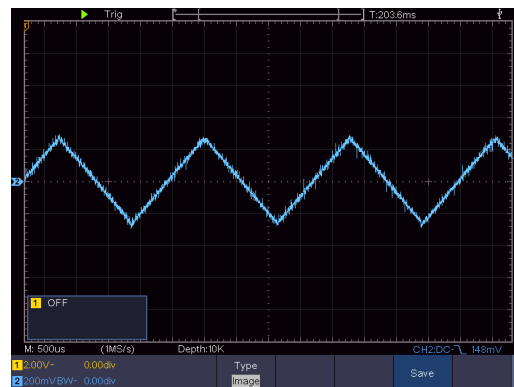
4.2.6 VCO

VCO-modulen generar toner med frekvensinnehåll mellan 20 Hz och 20 kHz samt kan generera flera toner parallellt för att uppnå parafoni. Den gör detta via bearbetning av inkommande MIDI-signaler. Det går även att justera tonläget genom tonböj-reglaget med inkommande MIDI-meddelanden. Det finns 4 olika vågformer att välja emellan: sinus, sågtand, triangel och fyrkant. Utgången från dessa illustreras i figur 4.2.

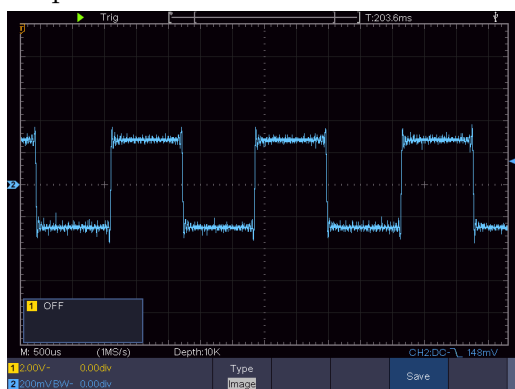
Moduler stödjer parafoni och kan generera upp till 16 simultana röster stabilt. Vid högre röstantal uppstår distorsion till följd av begränsad beräkningskapacitet hos mikrokontrollern. Vid parallellt eller seriekopplat routing-läge används två oscillatorer per röst, vilket halverar antalet simultana toner till 8. Modulen har även intern



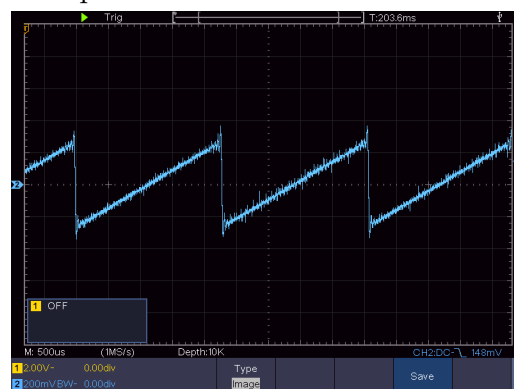
(a) Sinusvåg visualiserat på oscilloskop.



(b) Triangelvåg visualiserat på oscilloskop.



(c) Fyrkantsvåg visualiserat på oscilloskop.



(d) Sågtandsvåg visualiserat på oscilloskop.

Figur 4.2: Mätningar av de olika vågformerna gjorda med oscilloskop

VCA och kan ta emot enveloppsignaler från en ADSR modul för amplitudformning.

Via OLED-skärmen visualiseras modulens tillstånd. Skärmen i kombination med rotationsenkodern gör det möjligt att justera hur rösterna manipuleras internt, inklusive val av vågform, röstantal, routingläge samt tonhöjd. Det går exempelvis att spela flera röster unisont från samma VCO-modul. FM-modulering och flip-modulering är tillgängliga som parametrar via samma menysystem. Moduler stödjer även hantering av inkommande CV-styrsignaler för att modulera samtliga parametrar.

4.2.7 VCF

VCF-modulen uppfyller alla etablerade krav. Det går att justera frekvensen på intervallet från 20 Hz till 20 kHz, samt Q-faktorn på intervallet 0.1 till 10. Det finns tre olika typer av filter att välja mellan: lågpas, högpas och bandpas. Valet av dessa inställningar görs via en rotationsenkoder som är kopplad till en display där användaren kan se det valda filtret samt den nuvarande frekvensen och Q-faktorn. Moduler stödjer även hantering av inkommande CV-styrsignaler för att modulera cutoff och resonans.

4.2.8 LFO

LFO-modulen genererar en I²S-signal mellan 0.01 Hz och 20 Hz och kan skicka den till olika parametrar för modulering. Användaren kan justera vågformen på LFO-signalen till en av fyra olika vågformer: sinus, sågtand, triangel och fyrkant. Det går även att justera amplitud och frekvensen av den genererade vågen. Vågformen som skickas visualiseras på en display där användaren kan interagera med inställningarna med hjälp av en rotationsenkoder.

4.2.9 Mixer

Mixermodulen har upp till fyra ingångskällor som alla summeras ihop och spelar den förväntade vågen. Varje ljudkanal kan även regleras med respektive potentiometer som styr amplituden på ljudvågen. Detta verifierades genom att koppla in flera VCO-moduler i mixerns fyra ljudkanaler med olika, distinkta vågformer. Därmed bekräftades att respektive potentiometer styrde amplituden hos rätt signal.

4.3 Utökade mål

I rapportens inledning berördes möjligheten att realisera fler moduler och funktioner än det MVP:n innefattade. Nedan beskrivs de utökade mål som uppnåddes under projektet.

ADSR

ADSR-modulen skickar ut en UART-signal som moduler kan ta emot för att justera parametrar internt. Moduler skapar initialt ett envelopp som inte aktiveras förrän ADSR-modulen är inkopplad och skickar information om de olika enveloppparametrarna. Modulen stödjer endast en UART-utgång till följd av mikrokontrollerns fysiska begränsningar.

MultiFX

MultiFX syftar på en modul som innefattar ett flertal olika effekter användaren kan tillämpa på ljudsignalen. Genom ett displaybaserat menysystem går det att aktivera och justera olika funktioner och parametrar. I nuläget har två effekter implementerats: *reverb*, eller rumsklang, och *delay*, eller fördröjning. Reverb-modulen innehåller parametrar för att justera den upplevda storleken på effekten samt dess dämpning. Genom att justera dessa kan användaren skapa en modellering av olika akustiska rum som synten kan spelas i, det vill säga hur ljudets reflektioner simuleras. Delay-modulen däremot tar emot en ljudsignal, fördröjer den och spelar därefter upp den igen för att skapa en ekoeffekt. Användaren kan producera ekon och kaskader av toner genom att ändra fördröjningsintervallet samt dess feedback-parameter. I detta sammanhang beskriver feedback hur mycket av ljudsignalen som återkopplas rekursivt i fördröjningsalgoritmen.

5

Diskussion

Detta kapitel redogör de beslut och tekniska val som gjorts i projektet. Det lägger fram diskussionspunkter angående resultatet och lägger grunden för rapportens slutsats. Det går igenom lärdomar och eventuella förbättringsmöjligheter angående konstruktionen. Även felkällor som uppstått under projekten tas upp, samt etiska aspekter angående metodik och resultat.

5.1 Kommersiell prissättning

En drivande faktor i syftet av projektet är den höga prissättningen hos kommersiella modulära syntar. Syntmarknadens relativt minimala omfattning driver upp tillverkningskostnaden per enhet. Marknaden har många små aktörer, men med en relativt låg efterfrågan kan modulerna inte massproduceras. Detta innebär låga beställningsvolymen av komponenter och delar till produkterna, vilket är kostsamt för producenterna [70].

Lansering av en produkt inkluderar många kostnader utöver materialkostnaderna som design, kodning, dokumentation och mera. När produktionen sedan inte är så stor, slås dessa kostnaderna ut över få enheter, vilket bidrar till ett högre marknadspris [70].

Det finns en markant skillnad i känslan av att använda en fysisk modulär synt och digitala emuleringar i en *DAW* (Digital Audio Workstation). Det blir två helt olika upplevelser och arbetsflöde. Begränsningarna och den taktila känslan i en analog synt bidrar till att skapa den musikaliska känslan [71]. Det finns även en viss estetik som fångas i analog syntar [72]. Dessa faktorer påverkar prissättningen hos modulära syntar.

5.2 Tekniska val

I detta avsnitt förklaras och motiveras några av de tekniska val som gjorts vid planering och konstruktion av produkten.

Digital ljudsyntes

Istället för en analog synt valdes digital ljudsyntes, vilket är mindre vanligt i kommersiella syntar. Detta val gjordes för att minska kostnaden på modulerna genom att

ersätta elektriska komponenter med mikrokontroller och sedan emulera den analoga funktionaliteten med mjukvara. Detta gav upphov till flera problem under utvecklingen som inte hade påverkat oss om analoga metoder hade använts. Till exempel uppstod flera problem med att processeringen inom modulerna var för långsam vilket ledde till att ljudsignalerna ibland skickades en klockcykel försent. Det visade sig även problematiskt att hitta digitala lösningar till vissa analoga metoder som kommunikation mellan moduler och filter.

Hantera en funktion per mikrokontroller

Det är mer modulärt att använda flera mikrokontroller istället för att ha en som gör allt. Tillverkare behöver även veta hur mycket belastning en mikrokontroller klarar av att hantera samtidigt annars kan det leda till artefakter och missljud i signalen. Mycket signalhantering på en enhet är kostnadseffektivt men behöver dock inte nödvändigtvis leda till ett bättre ljud. Därav är det viktigt att om tillverkare önskar förverkliga flera olika former av ljudhantering på en enhet att koden är optimerad och utnyttjar så mycket av hårdvaran som möjligt. Flera självständigt fungerande moduler innebär att användaren kan välja vilka den vill verkligen ha istället för allt eller inget. Det blir en avvägning för utvecklaren hur mycket funktionalitet de vill implementera per modul med hänsyn till produktionskostnad och ljudkvalitet.

I flera moduler finns extra funktionalitet bortom modulens explicita funktion implementerad. VCO-modulen innehåller parametrar för att vidare justera generationen av vågformer. Genom FM och andra vågformsmanipuleringar går det att internt forma signalen i samma modul. För att detta ska ske effektivt utnyttjas båda mikrokontrollernas kärnor för att delegera arbete. I detta fall utförs ljudbearbetningen och den övriga koden på separata kärnor.

VCA-funktionalitet internt inom VCO:n

Under projektets gång identifierades möjligheten att implementera amplitudmodulering direkt i VCO:n, vilket eliminerade behovet av en separat VCA-modul. Detta besult medför en direkt kostnadsbesparing, då en hel modul kan uteslutas från systemet. I nuläget stödjer implementationen av intern VCA modulering via envelopp infång från ADSR-modulern. En utveckling från detta skulle vara att även möjliggöra amplitudmodulering via LFO, vilket skulle ge användaren ytterligare möjligheter att forma ljudet dynamiskt.

Utöver kostnadseffektivisering ger intern VCA en mer polyfonisk upplevelse. En extern VCA-modul hade applicerat amplitudmoduleringen på den redan summerade signalen från VCO:n, vilket innebär att samtliga röster följer en och samma amplitudkurva och tvingas så starta och sluta simultant. Med VCA-funktionalitet intern i VCO:n kan varje röst istället formas individuellt med egen amplitudförändring, vilket möjliggör att toner är oberoende av varandra.

Raspberry Pi Pico

Alla moduler i synten är baserade på mikrokontrollern Raspberry Pi Pico. Valet utgick främst från dess låga kostnad i förhållande till önskvärda egenskaper. Däröver stödjer Raspberry Pi Pico programmeringsspråket C, en egenskap som prioriterades på grund av dess potentiell för hög prestanda.

Motivering till programmering i C

Mjukvaran bygger på det medföljande C/C++ SDK som kommer med Raspberry Pi Pico. Valet att programmera i C grundade sig främst i behovet att uppnå hög prestanda och effektivitet när det kom till att utnyttja hårdvarans resurser. Samtidigt hade inte alla i projektgruppen samma erfarenhet av programmering för inbäddade system eller av C. Detta ledde till en period av inläring och studerande för att förstå sig på hur effektiv C-kod ska se ut. Ett alternativ till att använda C hade varit MicroPython, en implementation av Python optimerad för att köras på mikrokontroller [73].

I²S

I²S-protokollet användes i detta projekt för att det har tre kanaler vilket möjliggör funktionalitet för stereoljud. Detta innefattar separata signalvägar för vänster och höger ljudkanal. När I²S-kommunikation mellan modulerna implementerades, var det nödvändigt att implementera ett eget kodbibliotek eftersom det inte existerade ett redan tillräckligt tillfredsställande bibliotek. Att möjliggöra kommunikation mellan modulerna tog en stor del av tiden i början av projektet, vilket bromsade utvecklingen av annan funktionalitet.

Central bakplansmodul

I detta projekt används en central bakplansmodul som genererar två klocksignaler som styr kommunikationen mellan modulerna. Det observerades att när modulerna vidarebefordrade klocksignalerna till varandra i en kedja blev kommunikationen osynkroniserad. Därför gjordes detta val för att synkronisera samtliga moduler till en gemensam klocka. Det minskar också antalet GPIO-pinnar som används per modul, eftersom de endast behöver ta emot en klocksignal, vilket endast kräver en pinne per signal. Om de hade behövt skicka vidare klocksignalerna hade det krävts ytterligare två pinnar per modul. Samtidigt riskerar detta att hela synten att sluta fungera om bakplansmodulen uppvisar fel.

MIDI på bakplan

Att skicka MIDI på bakplanet valdes främst för att möjliggöra polyfoni i VCO:n. Ett alternativ hade varit en separat MIDI-modul som tolkar MIDI-meddelanden och omvandlar tonhöjden till en CV-signal. Denna konventionella CV-signal med 1 V/Oktav stödjer däremot inte flera toner samtidigt, då det inte finns något sätt att representera fler än en ton i signalen. Om två toner spelas samtidigt och deras spänningsnivåer från CV-signalen skulle summeras, blir resultatet endast en högre

tonhöjd, och det finns inget sätt att skilja på denna summerade tonhöjd och en enskild ton med samma tonhöjd. Eftersom VCO:n istället kan läsa MIDI-meddelanden direkt, kan vågformerna genereras internt för varje ton och amplituderna summeras innan ljudet skickas vidare från modulen.

En annan anledning till detta val är möjligheten för andra moduler att också kunna läsa MIDI. En ADSR-modul kan till exempel använda MIDI-meddelanden för att veta när nya toner trycks ner, och starta om sitt envelopp. Filter och effekter kan detektera när fysiska reglage justeras på en ansluten MIDI-klaviatur. Dessa justeringar kan då kopplas till interna parametrar i modulerna, och uppdateras i takt med att reglagen justeras. Till exempel kan modulationshjulet på en MIDI-klaviatur styra cutoff-frekvensen på ett lågpasfilter.

Ännu en nackdel med att ha MIDI på bakplanet är att en GPIO-pinne och ett UART-gränssnitt är reserverat för MIDI på varje modul, även i moduler som inte behöver läsa det. Detta är i sin tur anledningen till att det inte finns fler än en ENV-ingång per modul, då två ENV-ingångar annars hade varit möjligt hårdvarumässigt på vissa moduler. Ytterligare en nackdel är att signalkedjan från MIDI-kontakten till VCO:n, som annars är klart synlig i en modulär synt, blir mindre självklar.

Parafoni

En parafonisk syntkedja valdes för att behålla upplevelsen av kommersiella modulära syntar, där ljudsignaler skickas mellan moduler och processeringen appliceras på dessa ljudsignaler. Samtidigt önskades möjligheten att spela flera toner samtidigt genom en MIDI-klaviatur.

Från MIDI-ingången till VCO:ns utgång, är synten polyfonisk. All intern bearbetning sker individuellt för varje röst, exempelvis hos ADSR-enveloppet som styr amplituden för varje ton separat. Efter VCO:ns utgång summeras rösterna, och de individuella tonerna kan inte längre separeras. Från detta steg och framåt är kedjan parafonisk, och liknar den monofoniska signalväg som är vanlig hos kommersiella modulära syntar.

Parafoni medför en distinkt klangkaraktär - skild från poly- och monofoniska syntar. Skillnaden är inte omedelbart märkbar för den vanliga användaren. När tonerna i ett ackord spelas samtidigt är beteendet identiskt med en polyfonisk synt. Det blir tydligare när tonerna ringer ut var för sig, vilket färgar tonklangen på ett sätt som i sig kan betraktas som en musikalisk effekt.

En nackdel med detta är att signalbehandling efter VCO:n appliceras på den summerade signalen snarare än på enskilda röster. Detta innebär exempelvis att filter påverkar samtliga toner i ett ackord med samma parametrar, oavsett när respektive tangent aktiverades. En polyfonisk implementation hade istället kunnat tilldela varje röst ett eget filter. Parafoni är således en kompromiss: fler toner kan spelas samtidigt, men möjligheten till individuell rösthantering upphör efter VCO-steget.

Ljudkvalitet

En central aspekt under projektets gång var att minimera latens över flera moduler samtidigt som hög ljudkvalitet upprätthölls. Detta åstadkoms genom en hög samplingsfrekvens (44,1 kHz eller högre), till priset av ökad belastning på hårdvaran. Även valet av antalet samplingar per buffert hade betydelse för olika moduler. Fler samplingar innebar att en modul fick mer tid att utföra bearbetning av mjukvaran och ljudsignalen, men kunde samtidigt leda till högre latens. Om mjukvaran inte kunde optimeras i den utsträckning att den kunde köras utan märkbara artefakter, ansågs det bättre att öka buffertstorleken och acceptera något högre latens. Valet av I²S som primärt protokoll för överföring av ljuddata grundade sig i dess prestanda, då protokollet var både effektivt och stödde utmatning av högupplöst stereoljud.

3D-skrivet chassi och frontpanel

Gällande den fysiska presentationen av synten eftersöktes ett material som var lätt och snabbt att skapa prototyper med. Då det fanns tillgång till 3D-skrivare och filament användes detta primärt för att bygga frontpanelerna och chassi-komponenterna. 3D-skrifter är enkelt reproducerbara genom att delning av 3D-filer, vilket är fördelaktigt.

Överföring av ADSR-parametrar via UART

För att flera toner ska kunna tilldelas egna ADSR-kurvor, särskilt på VCO:n, måste ADSR-genereringen ske internt i modulen så att varje ton får en egen styrkurva. Parametrarna för ADSR-kurvan kan överföras via UART-meddelanden och appliceras internt i modulen. Detta möjliggör polyfoni istället för parafoni på VCO:n. VCF:n förblir parafonisk eftersom varje ton inte längre kan separeras individuellt efter att de summerats i en I²S-dataström.

Filtermodulen implementerad med biquad-filter

Det finns ett stort antal sätt att implementera ett digitalt filter. Den primära anledningen till att det valdes att använda en biquad-implementation ligger i att den är baserad på en IIR-arkitektur (Infinite Impulse Response). Detta innebär bättre prestanda då det krävs färre beräkningar för att uppnå önskad filtreringseffekt. En ytterligare fördel är att IIR-filter ofta lämpar sig väl för att efterlikna karaktären hos analoga filter.

Filter kan utformas med olika responskaraktärer, exempelvis Butterworth eller Chebyshev, vilket ger olika egenskaper i filterljudet. En bidragande faktor i valet av biquad var tillgången till ett C-bibliotek där den matematiska implementationen redan fanns tillgänglig. Därefter var det lätt att experimentera och justera parametrar för de olika filtertyperna.

5.3 Lärdomar och förbättringsmöjligheter

I det här kapitlet sammanställs de lärdomar och förbättringsmöjligheter kopplat till produkten som identifierades under projektets gång. Förhoppningen är att detta kan stödja vid potentiell vidareutveckling av produkten eller identifiering av områden för framtida studier.

5.3.1 Hårdvara

Inledande prototyper av projektet gjordes via enkla konfigurationer på kopplingsdäck. Den stora fördelen med detta var smidigheten med att kunna testa funktionalitet i dess mest fundamentala form. Det var snabbt och enkelt att testa och felsöka tidiga implementationer av VCO:n och MIDI-gränssnittet. Detta i sin tur ledde till att potentiella felkällor och problem kunde identifieras tidigt. När synten endast bestod av ett fåtal moduler var denna metod mycket effektiv men allt eftersom antalet moduler ökade, blev störningar och bristfällig strömförsörjning ett växande problem. Till slut var sammankopplingen så instabil att det var svårt att fortsätta arbeta utan att lägga onödig tid på att navigera den stora mängden felkällor som tillkommit. Det blev svårt att arbeta då det inte gick att avgöra om missljud berodde på dålig hårdvara, kod eller hur modulerna var kopplade. Denna upplevelse ledde till ett antal prototypkort där komponenterna manuellt kopplats ihop utan förutbestämda banor. Produktionen av dessa kort gav insikt i hur det färdiga baskortet skulle se ut samt hur nödvändigt det är att alltid ha reservkomponenter. Under projektets gång gick flera komponenter sönder, vilket ledde till oförutsedda förseningar och behov av nya beställningar. Något som kan förbättra systemet är en kontakt på bakplanet som kortsluts till GND, och tillsammans med en I²C-buss som också finns på bakplanet, automatisk detektera moduler.

5.3.2 Mjukvara

Ett stort genombrott i projektet var implementationen av vårt I²S-bibliotek. Det gjordes via pico-enhetens PIO, vilket står för Programmable Input Output. Då mikrokontrollen inte har inbyggt stöd för I²S-protokollet byggdes ett eget bibliotek för att möjliggöra det. I²S-lösningen möjliggjorde att kommande moduler lätt kunde integreras med existerande kod som skickade I²S-signaler. Att detta var något som löstes tidigt ledde till snabb implementation av VCF, LFO och mixern. Återkommande problem över projektets gång har varit sample underruns, där kod inte hinner behandla alla ljudsamplingar i realtid. Detta har resulterat i oönskade artefakter och missljud. För att komma runt detta behövde koden effektiviseras. I särskilt krävande algoritmer som för filteruträkningar i VCF och MFX var det av största vikt att få de så snabba som möjligt eller annars införa fler samplingar per buffert och därmed införa mer latens på dessa moduler. En förbättring skulle vara att möjliggöra stereoljud. I den nuvarande implementationen lagrar modulerna ljudsignalerna i ett 32-bitars heltal, men använder bara den lägre halvan. Detta gör att bara den vänstra ljudkanalen får ljud när ljudsignalen skickas till den analoga utgången. Eftersom I²S redan stödjer funktionalitet för stereo, är detta en mycket

rimlig förbättring.

En annan viktig lärdom var att faktiskt dra nytta av de två processorkärnorna som finns på mikrokontrollern. Längre användes bara en processorkärna, men alltför som modulerna fick fler funktioner räckte inte tiden till. Detta observerades först på VCO-modulen, som gav upphov till ljudbortfall i några millisekunder medan rotationsenkodern stegade igenom menysystemet och displayen ritades om. Det visade sig att displayen och dess protokoll tar för lång tid att överföra ny pixelinformation samtidigt som ljudvågor genereras med en samplingsfrekvens på 48 000 Hz. Med en buffertstorlek på endast 256 samplingar behöver hela bufferten tömmas ungefär var 5:e millisekund, men tiden att uppdatera displayen kunde ibland bli över 20 millisekunder, vilket ledde till tomma buffertar och inget ljud i en kort stund.

5.4 Felkällor

Ett problem med metodologin är begränsningar i kostnadsjämförelsen som gjordes för att avgöra huruvida synten är kostnadseffektiv. Det nämnda priset på den konstruerade produkten inkluderar inte alla omkostnader som finns vid fullständig produktion, medan detta är inkluderat i priset för den kommersiella synten. En mer relevant jämförelse hade varit att analysera materialkostnaden för kommersiella modulära syntar. Denna metod har egna begränsningar, eftersom kommersiella tillverkare sannolikt erhåller volymrabatter vid komponentinköp, en förmån som inte var tillgänglig i detta projekt. Vidare har projektets komponentanskaffningsmöjligheter varit begränsade via *RS Electronics*, *Electrokit* och *Mouser Electronics*, vilket begränsat komponentutbudet och därmed möjligheten till kostnadsoptimering.

Ett annat problem med metoden är att det inte gjordes någon utvärdering på att det skulle vara samma känsla att använda den skapade produkten som en kommersiell modulärsyntar. Ett möjligt sätt att kunnat testat detta är att göra kvalitativa användartester med erfarna synt-användare.

Dessa metodlogiska begränsningar gjordes i avvägning av tid, produktionen av den modulära synten och kostnad optimering togs som prioritet över en noggrannare utvärdering.

5.5 Samhälleliga och etiska aspekter

I detta kapitlet diskuteras de etiska aspekter som identifierades med projektets genomförande och resultat, samt hur samtliga risker kan minimeras eller åtgärdades. Kapitlet belyser även kopplingen till globala mål.

5.5.1 Etiska aspekter med metod

Det största etiska problemet som identifierades under utförande var risken att utsätta projektgruppen eller personer i dess omgivning för fysisk skada. Projektet

innehöll mycket arbete med elektronik där brandrisk och utsläpp av giftiga gaser identifierades som det största riskerna. Dessa risker var som störst vid mjuklödning av komponenter till prototypkort och kretskort. Denna mjuklödning fungerar genom att använda en lödkolv som värms till över 300°C för att smälta lödtennet. Mjuklödning frigör rök vid smältning av lödtennet, vilket vid inhalering har risken att leda till astma [74]. Det finns även uppenbara risker för brand och brännskador med lödningsarbete på grund av de höga temperaturerna på lödkolven. Riskerna minimerades genom att arbete på avsatta platser för lödning vilka har fläktar för att suga undan partiklarna, arbetsplatsen var även utrustad med brandsläckare anpassade för elektroniska bränder.

Det finns även miljöaspekter att ta hänsyn till under konstruktionen. Under utvecklingen användes en iterativ process där fokus låg på att återanvända så mycket komponenter som möjligt från steg till steg i processen, för att minska svinnet av elektriska komponenter. Detta är en viktig miljöaspekt eftersom att produktionen av elektriska komponenter har en negativ påverkan på miljön. Produktionen kräver stora mängder sällsynta jordartsmetaller och släpper ut stora mängder koldioxid [75]. Även PCB-produktionen medför risk för stora utsläpp av koldioxid och skadliga ämnen [76]. Det är inte möjligt att undvika utsläppen helt, men det kan vara värt att se över valet av leverantör.

Den slutliga miljöaspekten som togs i åtanke var att användningen av AI vid kodning minimerades, eftersom AI:s stora energibehov ställer stora krav på miljön. Detta syns tydligt i att AI i jämförelse med en vanlig Google-sökning använder tio gånger mer energi [77].

5.5.2 Etiska aspekter med resultat

Den huvudsakliga etiska aspekten med resultatet är ökad jämställdhet genom att åtgärda de ekonomiska och sociala aspekterna av problemet [78]. Denna etiska aspekten skulle ske genom att rapporten driver forskningen om eller skapandet av en kommersiell modulär synt av låg kostnad. Detta hade i sin tur kunnat ökat jämställdheten genom att ge individer av lägre ekonomisk ställning tillgång till modulära syntar.

6

Slutsats

Syftet med detta projekt var att undersöka hur kostnadseffektivt det är att använda mikrokontroller vid konstruktion av en modulär synt av kommersiell kvalitet. Utifrån resultatet kan slutsatsen dras att det under projektet framgångsrikt konstruerats ett komplett modulärt syntsystem till ett kostnadseffektivt pris. Synten fullföljer projektets MVP och kraven på ljudkvalitet och latens.

Moduler baserade på mikrokontroller möjliggjorde utvecklingen av avancerade moduler. Samtidigt introducerades prestandamässiga utmaningar, då det uppstod svårigheter med att hinna exekvera all programlogik inom de givna klockcyklerna.

Vid framtida forskning är det berikande att fokusera på en mer robust marknadsanalys för att ta reda på konkurrenskraften av att använda mikrokontrollers i konstruktionen av kommersiella modulära syntar. Det hade även varit givande att fokusera att göra fler användartester för att ta reda på om projektet har en jämförbar känsla som en kommersiell synt.

Referenser

- [1] K. Kaiser. "Modular Synthesis — A Beginner's Guide", Hosa Technology, hämtad 28 april 2026. URL: <https://hosatech.com/press-release/modular-synthesis-a-beginners-guide/>.
- [2] Future Music. "Do analogue synthesizers really sound better?", hämtad 18 maj 2026. URL: <https://www.musicradar.com/news/whats-so-special-about-analogue-synthesizers>.
- [3] T. Pinch och K. Bijsterveld, "'Should One Applaud?': Breaches and Boundaries in the Reception of New Technology in Music", *Technology and Culture*, årg. 3, s. 548–559, 2003, Hämtad: 2026-01-29.
- [4] S. McGuire och Z. Matějů, *The Art of Digital Orchestration*. Taylor & Francis, 2020, s. 100–101, Hämtad: 2026-04-14, ISBN: 9781000286991. URL: <https://books.google.se/books?id=S90NEAAQBAJ>.
- [5] J. Vanhanen, "Rewired – The Resurgence of Analog Sound Synthesis", *Mustekala*, dec. 2016, Hämtad: 2026-04-21. URL: <https://mustekala.info/teemanumerot/art-technology-science-philosophy-4-2016-vol-65/rewired-the-resurgence-of-analog-sound-synthesis/>.
- [6] Khan Academy, *Sound properties: amplitude, frequency, wavelength*, <https://www.khanacademy.org/science/physics/mechanical-waves-and-sound/sound-topic/v/sound-properties-amplitude-period-frequency-wavelength>, Hämtad: 2026-04-16, 2014.
- [7] Khan Academy, *Decibel scale*, <https://www.khanacademy.org/science/physics/mechanical-waves-and-sound/sound-topic/v/decibel-scale>, Hämtad: 2026-04-16, 2014.
- [8] Nationalencyklopedin, *Ton*, <https://www.ne.se/uppslagsverk/encyklopedi/enkel/ton>, NE Nationalencyklopedin AB, hämtad 2026-04-21, 2026.
- [9] Nationalencyklopedin, *Ljud*, <https://www.ne.se/uppslagsverk/encyklopedi/enkel/ljud>, NE Nationalencyklopedin AB, hämtad 2026-04-21, 2026.
- [10] Nationalencyklopedin, *Oktav*, <https://www.ne.se/uppslagsverk/encyklopedi/enkel/oktav>, NE Nationalencyklopedin AB, hämtad 2026-04-21, 2026.
- [11] Nationalencyklopedin. "Liksvävande temperatur", NE Nationalencyklopedin AB, hämtad 28 april 2026. URL: <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/liksv%C3%A4vande-temperatur>.

- [12] Nationalencyklopedin. “Skala”, NE Nationalencyklopedin AB, hämtad 28 april 2026. URL: <https://www-ne-se.eu1.proxy.openathens.net/uppslagsverk/encyklopedi/l%C3%A5ng/skala>.
- [13] Nationalencyklopedin. “Intervall”, NE Nationalencyklopedin AB, hämtad 28 april 2026. URL: <https://www-ne-se.eu1.proxy.openathens.net/uppslagsverk/encyklopedi/l%C3%A5ng/intervall>.
- [14] Nationalencyklopedin. “Stämgafl”, NE Nationalencyklopedin AB, hämtad 28 april 2026. URL: <https://www-ne-se.eu1.proxy.openathens.net/uppslagsverk/encyklopedi/l%C3%A5ng/st%C3%A4mgafl>.
- [15] T. Connaghan. “Polyphonic vs. Monophonic: What’s the Difference?”, eMastered, hämtad 24 april 2026. URL: <https://emastered.com/sv/blog/polyphonic-vs-monophonic>.
- [16] Nationalencyklopedin. “Polyfoni”. Hämtad: 2026-04-24, NE Nationalencyklopedin AB. URL: <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/polyfoni>.
- [17] P. Adrien. “Monophonic, Polyphonic, Paraphonic Synthesizers: What’s the Difference?”, hämtad 17 maj 2026. URL: <https://www.projethomestudio.fr/en/synthes-mono-poly-paraphonic/>.
- [18] T. Rosati och T. Hsu, *Play with Sound: Manual for Electronic Musicians and Other Sound Explorers*. Taylor & Francis, 2025, s. 170–171, Hämtad: 2026-04-14, ISBN: 9781040194041. URL: <https://books.google.se/books?id=FlKNEQAAQBAJ>.
- [19] T. Horgan, *How Synthesizers Work - A Simple Guide*. Tony Horgan, 2024, s. 7–16, Hämtad: 2026-04-17. URL: https://books.google.se/books?id=_PScEAAAQBAJ.
- [20] J. Comeau, *Waveforms*, <https://pudding.cool/2018/02/waveforms/>, Publicerad på The Pudding, Hämtad: 2026-04-23, 2018.
- [21] Omegatron. “Waveforms”, hämtad 14 april 2026. URL: <https://commons.wikimedia.org/wiki/File:Waveforms.svg>.
- [22] M. Vail, *The Synthesizer: A Comprehensive Guide to Understanding, Programming, Playing, and Recording the Ultimate Electronic Music Instrument*. Oxford University Press, 2013, Hämtad: 2026-04-17, ISBN: 9780195394894. URL: <https://research.ebsco.com/linkprocessor/plink?id=bdb3189f-3530-35ac-a3aa-e51d982c7e3c>.
- [23] Abdull. “ADSR Parameter”, hämtad 14 april 2026. URL: https://commons.wikimedia.org/wiki/File:ADSR_parameter.svg.
- [24] Audient. “A Beginner’s Guide to Modulation”, Audient, hämtad 18 maj 2026. URL: <https://audient.com/tutorial/a-beginners-guide-to-modulation/>.
- [25] G. Reid, “Modulation”, *Sound On Sound*, febr. 2000. hämtad 18 maj 2026. URL: <https://www.soundonsound.com/techniques/modulation>.
- [26] Nationalencyklopedin, *Filter*, <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/filter>, NE Nationalencyklopedin AB, hämtad 2026-04-23, 2026.

-
- [27] M. OpenCourseWare, *Signals and Systems Lecture 21: Sampling*, https://ocw.mit.edu/courses/6-003-signals-and-systems-fall-2011/12e6e5d7567fca2e993ef8563fef5a60_MIT6_003F11_lec21.pdf, Föreläsningsanteckningar, Hämtad: 2026-04-24, 2011.
- [28] Nationalencyklopedin, *Analog-digitalomvandlare*, <https://www.ne.se/uppslagsverk/encyklopedi/lång/analog-digitalomvandlare>, NE Nationalencyklopedin AB, hämtad 2026-04-21, 2026.
- [29] S. W. Smith, *Chapter 14: Introduction to Digital Filters*, https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_Ch14.pdf, From The Scientist and Engineer's Guide to Digital Signal Processing, Hämtad: 2026-04-10, 1999.
- [30] University of California, Davis, *Introduction to Digital Filters*, [Online]. Available: https://123.physics.ucdavis.edu/week5_files/filters/digital_filter.pdf, Hämtad: 2026-03-27.
- [31] Image-Line Software, *Cutoff and Resonance Filter*, Hämtad: 2026-04-27, u.å. URL: https://www.image-line.com/fl-studio-learning/fl-studio-beta-online-manual/html/glossary_cutres.htm.
- [32] D. Samal, *The impact of biquad filters on the Digital world: From audio to wearables*. Hämtad: 21-04-2026. URL: <https://www.einfochips.com/blog/the-impact-of-biquad-filters-on-the-digital-world-from-audio-to-wearables/>.
- [33] SFUptownMaker. "PCB Basics", hämtad 29 april 2026. URL: <https://learn.sparkfun.com/tutorials/pcb-basics/all>.
- [34] Highleap Electronic. "PCB Manufacturing Process Flow – Ultimate Guide Is Here", hämtad 29 april 2026. URL: <https://hilelectronic.com/pcb-manufacturing/>.
- [35] Raspberry Pi Ltd, *RP2040 Datasheet*, 2024. hämtad 23 april 2026. URL: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>.
- [36] R. M. Guglielmi. "UART 101: the blueprint of Serial Communications". Updaterad på: April 4, 2024, hämtad 26 mars 2026. URL: <https://www.playembedded.org/blog/uart-101-the-blueprint-of-serial-communications/>.
- [37] E. Peña och M. G. Legaspi, "UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter", *Analog Dialogue*, 2020, Hämtad: 2026-04-22.
- [38] EidenNor. "UART-signal", hämtad 28 april 2026. URL: <https://commons.wikimedia.org/wiki/File:UART-signal.png>.
- [39] J. Wu, "A Basic Guide to I2C", 2022, Hämtad: 2026-03-26. URL: <https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=17773439719>.
- [40] Mfloryan. "I2C data transfer", hämtad 28 april 2026. URL: https://commons.wikimedia.org/wiki/File:I2C_data_transfer.svg.

- [41] R. Keim. "Introduction to the I2S Interface", hämtad 26 mars 2026. URL: <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2s-interface/>.
- [42] wdwd. "I2S Timing", hämtad 26 mars 2026. URL: https://commons.wikimedia.org/wiki/File:I2S_Timing.svg.
- [43] J. Byron. "MIDI Tutorial: Hardware and Electronic Implementation". Hämtad: 2026-04-20. URL: <https://learn.sparkfun.com/tutorials/midi-tutorial/hardware--electronic-implementation>.
- [44] J. Byron. "MIDI Tutorial: Messages", hämtad 20 april 2026. URL: <https://learn.sparkfun.com/tutorials/midi-tutorial/messages>.
- [45] T. M. Association. "About MIDI-Part 3:MIDI Messages", hämtad 20 april 2026. URL: <https://midi.org/about-midi-part-3midi-messages>.
- [46] Gear4music. "The 11 Synths That Defined the '80s", hämtad 28 april 2026. URL: <https://www.gear4music.com/blog/80s-synths/>.
- [47] J. Kovarsky. "History of the Synthesizer, Part 1", hämtad 28 april 2026. URL: <https://hub.yamaha.com/keyboards/synthesizers/history-of-the-synthesizer-part-1/>.
- [48] T. Holmes, *Electronic and Experimental Music: Technology, Music, and Culture*, 6. utg. New York: Routledge, 2015, s. 277, Hämtad: 2026-04-28. URL: https://www.google.se/books/edition/Electronic_and_Experimental_Music/0cycCwAAQBAJ.
- [49] T. Pinch och F. Trocco, "The Social Construction of the Early Electronic Music Synthesizer", *Icon*, årg. 4, s. 4, 1998, Hämtad: 2026-01-29. URL: <https://www.jstor.org/stable/23785956>.
- [50] Thomann. "What's a Eurorack?", hämtad 23 april 2026. URL: https://www.thomann.se/onlineexpert_page_eurorack_modular_synthesizers_what_s_a_eurorack.html.
- [51] A. Delaro. "Understanding Voltage in Eurorack", hämtad 29 april 2026. URL: <https://www.rakt.info/post/understanding-voltage-in-eurorack>.
- [52] T. Horgan, *How Synthesizers Work - A Simple Guide*. Tony Horgan, 2022, s. 20, Hämtad: 2026-04-14. URL: https://books.google.se/books?id=_PScEAAAQBAJ.
- [53] T. Horgan, *How Synthesizers Work - A Simple Guide*. Tony Horgan, 2022, s. 27, Hämtad: 2026-04-14. URL: https://books.google.se/books?id=_PScEAAAQBAJ.
- [54] J. Wilczek. "Wavetable Synthesis Algorithm Explained", hämtad 17 maj 2026. URL: <https://thewolfsound.com/sound-synthesis/wavetable-synthesis-algorithm/>.
- [55] Nationalencyklopedin, *frekvensmodulering*, <https://www.ne.se/uppslagsverk/encyklopedi/adult/frekvensmodulering>, NE Nationalencyklopedin AB, hämtad 2026-05-18, u.å.

-
- [56] T. Rosati och T. Hsu, *Play with Sound: Manual for Electronic Musicians and Other Sound Explorers*. Taylor & Francis, 2025, s. 174–178, Hämtad: 2026-04-14, ISBN: 9781040193945. URL: <https://books.google.se/books?id=HCMzEQAAQBAJ>.
- [57] J. Wilczek. “FM Synthesis Explained For Audio Programmers: Art and Science”, hämtad 17 maj 2026. URL: <https://thewolfsound.com/fm-synthesis-explained-for-audio-programmers-art-and-science/>.
- [58] T. Horgan, *How Synthesizers Work - A Simple Guide*. Tony Horgan, 2022, s. 31, Hämtad: 2026-04-14. URL: https://books.google.se/books?id=_PScEAAAQBAJ.
- [59] Songstuff. “Standard Filter Types”, hämtad 27 mars 2026. URL: https://www.songstuff.com/recording/article/standard_filter_types/.
- [60] Oberheim. “Analog Synths and Expected Variations”, hämtad 17 maj 2026. URL: <https://support.oberheim.com/hc/en-gb/articles/20508911001874-Analog-Synths-and-Expected-Variations>.
- [61] M. Pennock, *Biquad: Small C Library for Implementing Second Order IIR Filters*, Hämtad: 2026-04-21, 2025. URL: <https://github.com/MattPennock/Biquad>.
- [62] Web Audio Working Group, *Cookbook formulae for audio equalizer biquad filter coefficients*, <https://webaudio.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html>, Hämtad: 2026-04-27, 2021.
- [63] R. Bristow-Johnson och W. A. W. Group, *Cookbook formulae for audio equalizer biquad filter coefficients*, <https://webaudio.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html>, Hämtad: 2026-04-27, 2021.
- [64] S. W. Smith, *The z-Transform*, https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_ch33.pdf, Chapter 33, Hämtad: 2026-04-27, 1999.
- [65] Wokwi. “wokwi.com”, hämtad 18 maj 2026. URL: <https://wokwi.com/>.
- [66] Serif (Europe) Ltd, *Affinity Photo*, version 2, Hämtad: 2026-04-23, 2023. URL: <https://affinity.serif.com/photo>.
- [67] Autodesk Inc., *Autodesk Fusion 360*, Hämtad: 2026-04-23, 2024. URL: <https://www.autodesk.com/products/fusion-360>.
- [68] Thomann. “thomann.se”, hämtad 19 mars 2026. URL: <https://www.thomann.se>.
- [69] R. Roberts, “My goal is to make synthesizers for \$9.90 so we can make them accessible to kids in poor countries: Behringer’s synths are cheap – but it wants to go further”, *Musictech.com*, 2025. URL: <https://musictech.com/news/gear/uli-behringer-synths-more-accessible>.
- [70] A. Douglas. “Are modular synths too expensive? and what is "too expensive" anyway?”, hämtad 17 maj 2026. URL: <https://www.attackmagazine.com/features/long-read/are-modular-synths-too-expensive-and-what-is-too-expensive-anyway/>.

- [71] J. Dobos. “Analog vs. Digital Synthesizers – My Take on the Old Debate”, hämtad 17 maj 2026. URL: <https://www.juliusdobos.com/the-philosophy-behind/analog-vs-digital-synthesizers-debate/>.
- [72] C. T. Andreas Kitzmann. “The Modular Journey: Uncovering analogue aesthetics in digital landscapes”, hämtad 17 maj 2026. URL: <https://www.cambridge.org/core/journals/organised-sound/article/modular-journey-uncovering-analogue-aesthetics-in-digital-landscapes/69C90BCA230B7C28CFB20323A7DC0BBD>.
- [73] M. Venkataramanan, “Micro Python: More Powerful Than Arduino, Simpler Than the Raspberry Pi”, *WIRED*, dec. 2013, Hämtad: 2026-05-14. URL: <https://www.wired.com/story/micro-python/>.
- [74] B. Sjögren, “Hälsoeffekter av gaser och partiklar bildade vid svetsning”, 2013, Accessed: 2026-05-16. URL: <https://www.av.se/globalassets/filer/publikationer/kunskapssammanställningar/halsoeffekter-av-gaser-och-partiklar-vid-svetsning-kunskapssammanställningar-rap-2013-5.pdf>.
- [75] Natur och Miljö. “Varför reparera elektronik?”, hämtad 16 maj 2026. URL: <https://www.naturochmiljo.fi/lar-dig-mer/repareramera/elektronik/>.
- [76] Adrian. “Green PCB Production Impact on the Environment: Eco Conscious PCB Production”, hämtad 16 maj 2026. URL: <https://www.allpcb.com/allelectrohub/green-pcb-production-impact-on-the-environment-eco-conscious-pcb-production>.
- [77] L. Towler. “Search Engines vs AI: energy consumption compared”, hämtad 16 maj 2026. URL: <https://kanoppi.co/search-engines-vs-ai-energy-consumption-compared/>.
- [78] UNDP Sverige. “Minskad ojämlikhet”, hämtad 16 maj 2026. URL: <https://globalamalen.se/om-globala-malen/mal-10-minskad-ojamlikhet/>.

A

Kravspekifikation

Kriterier		K/Ö	Vikt
1.	Miljö		
a	Alla moduler ska kunna förvaras och användas i rumstemperatur (15° C till 30° C) med normal luftfuktighet	K	
2.	Prestanda		
a	Samplingsfrekvensen ska vara minst 44 100 Hz (CD-kvalitet)	K	
b	Propageringsfördröjningen ska inte överstiga 11 ms per modul (512 samples vid 48 kHz)	K	
c	Propageringsfördröjningen ska vara lägre än 3 ms per modul (128 samples vid 48 kHz)	Ö	3
3.	Hårdvarugränsnitt		
a	Användaren ska kunna använda 3-poliga patchkablar med 3,5 mm-kontakter för att sammankoppla moduler	K	
b	Systemet ska kunna strömförsörjas från en USB-C-port	K	
c	Systemet ska ha en DIN-kontakt för anslutning till MIDI-utrustning	K	
d	Utgångssignalen ska kunna kopplas till högtalare eller hörlurar från en 3-polig 3,5 mm-kontakt	K	
e	Modulerna ska kunna monteras i ett rack med Eurorack-dimensioner	K	
4.	Bakplan		
a	Bakplanet ska kunna strömförsörja samtliga moduler	K	
b	Bakplanet ska kunna förse samtliga moduler med data från MIDI-ingången	K	
c	Bakplanet ska kunna förse samtliga moduler med klocksignaler för I ² S-protokollet	K	
d	En lysdiod ska indikera när MIDI-meddelanden tas emot	Ö	3
e	Bakplanet ska kunna möjliggöra kommunikation mellan moduler	Ö	3
f	Bakplanet ska kunna avgöra om en modul är ansluten i en given plats	Ö	4

Fortsätter på nästa sida

Fortsättning från föregående sida

Kriterier		K/Ö	Vikt
g	Bakplanet ska kunna avgöra vilken modul som är ansluten i en given plats	Ö	4
h	Bakplanet ska kunna visualisera den nuvarande uppkopplingen på en display eller hemsida	Ö	5
i	Bakplanet ska kunna förse samtliga moduler med en temporeferens	Ö	5
j	Bakplanet ska ge samtliga moduler tillgång till USB-portens dataanslutningar	Ö	5
5. MIDI-gränssnitt			
a	Gränssnittet ska kunna bedöma notnummer	K	
b	Gränssnittet ska kunna bedöma tryckkänslighet	K	
c	Gränssnittet ska kunna bedöma position på "pitch bend"-reglaget	Ö	2
d	Gränssnittet ska kunna bedöma position på övriga reglage	Ö	4
6. VCO			
a	Modulen ska kunna generera frekvenser mellan 20 och 20 000 Hz	K	
b	Frekvensen ska kunna justeras genom MIDI-signaler eller manuellt via reglage	K	
c	Vågformen ska kunna ändras mellan sinus, sågtand, triangel och fyrkant	K	
d	Modulen ska kunna generera flera frekvenser samtidigt (polyfoni) när MIDI-ingången används	K	
e	Den valda vågformen ska kunna visas på en display	K	
f	Vågformen ska kunna moduleras med inkommande styrsignaler	K	
g	Modulen ska kunna generera flera voices samtidigt (unison) när MIDI-ingången används	Ö	
7. VCF			
a	Frekvensen ska kunna justeras mellan 20 Hz och 20 000 Hz	K	
b	Resonansen (Q-faktorn) ska kunna justeras mellan 0,1 och 10	K	
c	Filtertypen ska kunna väljas från en lista (lågpass, bandpass, högpass, osv)	K	
d	Cutoff och resonans ska kunna moduleras med hjälp av en inkommande styrsignal	K	
f	Det ska finnas en display där användaren kan se den nuvarande filtertypen samt dess frekvens- och resonansinställningar	Ö	4
8. LFO			
a	Frekvensen ska kunna justeras mellan 0,01 Hz och 20 Hz	K	
b	Vågformen ska kunna ändras mellan sinus, sågtand, triangel och fyrkant	K	

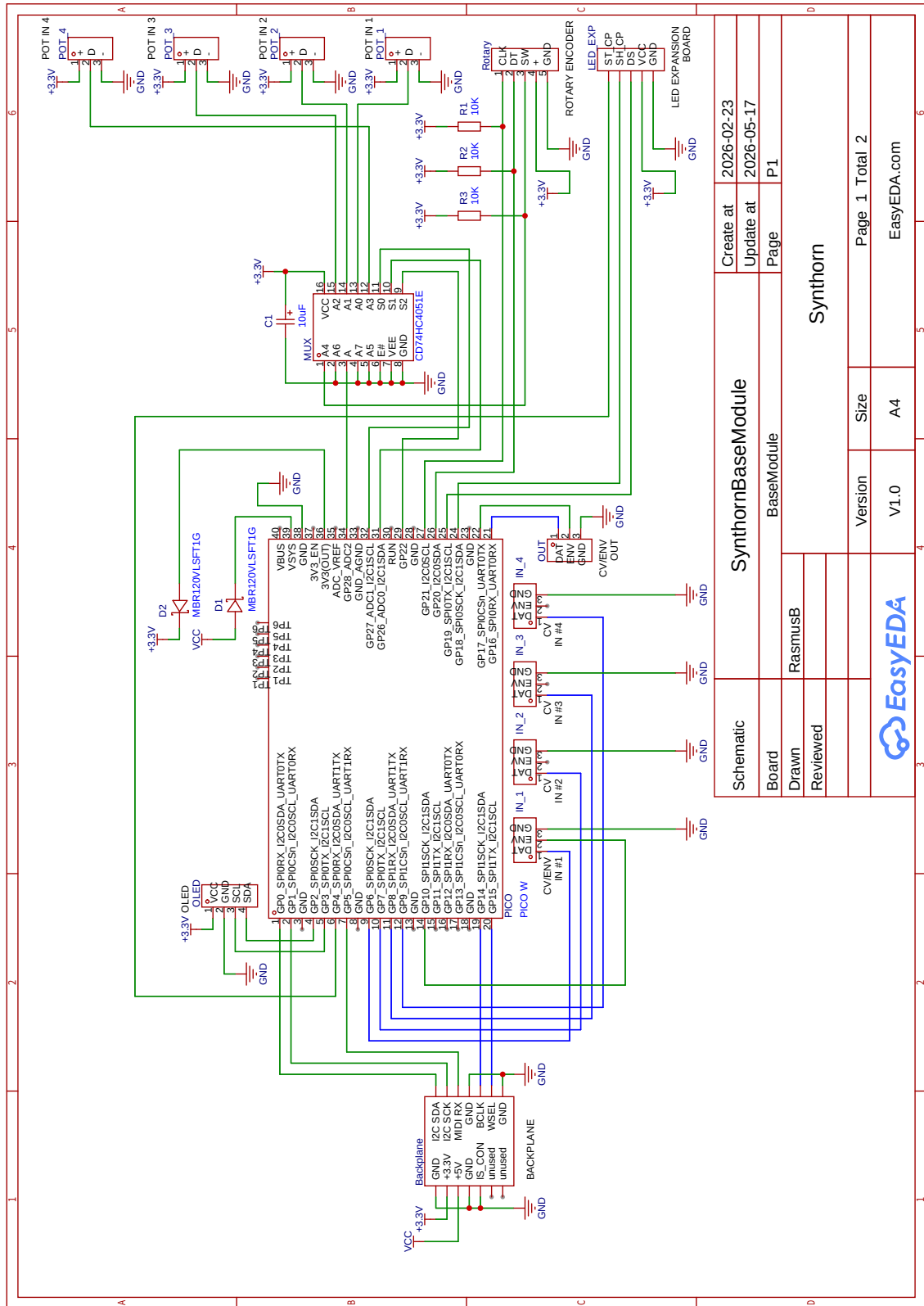
Fortsätter på nästa sida

Fortsättning från föregående sida

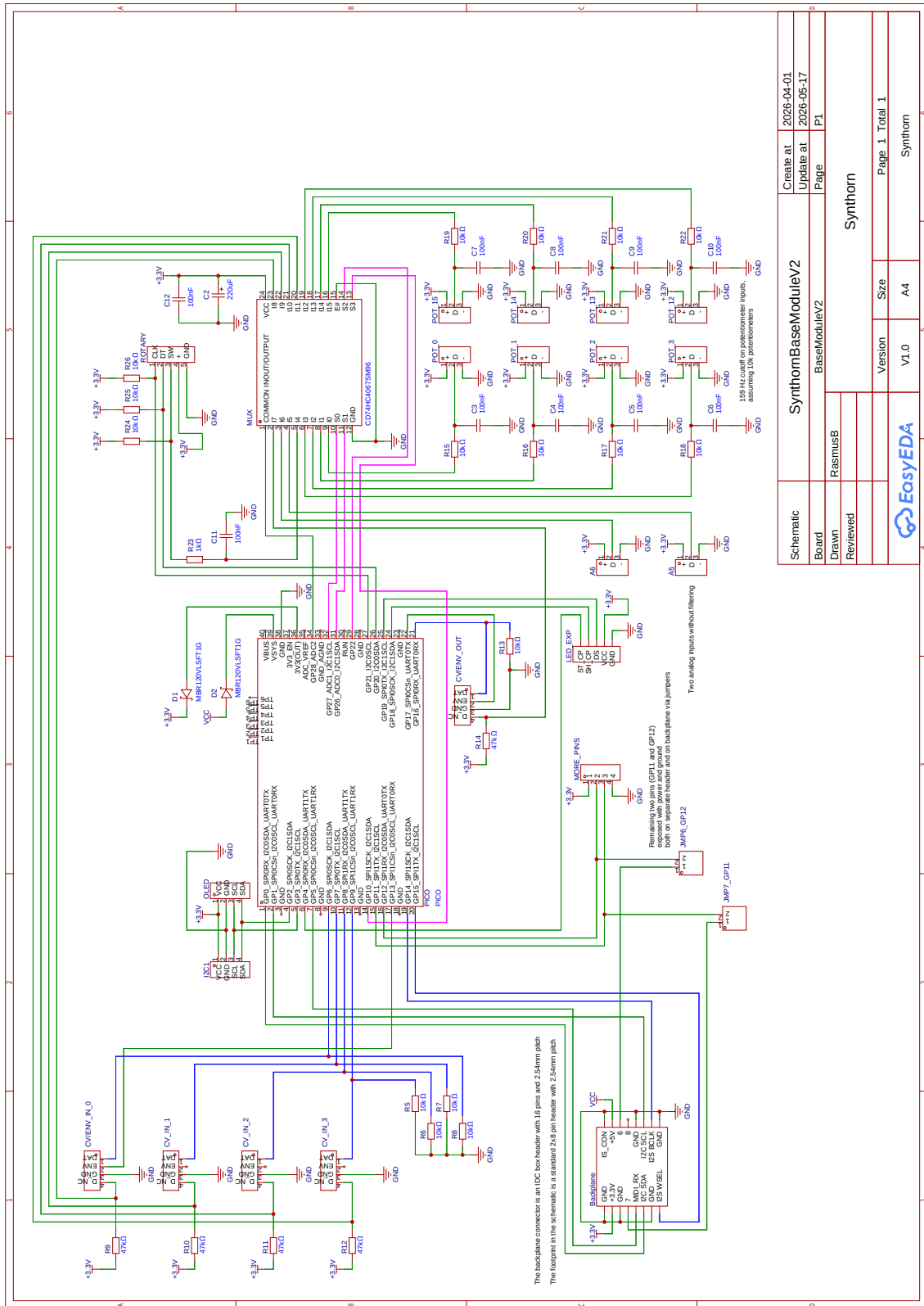
Kriterier		K/Ö	Vikt
c	Amplituden ska kunna ändras	K	
d	Frekvensen ska kunna justeras från tempot	Ö	4
9.	Mixer		
a	Modulen ska kunna läsa upp till 4 insignaler samtidigt	K	
b	Volymen för varje ingång ska kunna justeras med reglage	K	
c	Lysdioder ska indikera volymnivån för ingångssignalerna	Ö	4
d	Panorering för varje ingång ska kunna justeras med reglage	Ö	5

B

Elscheman



Figur B.1: Elschema revision 1.0



Figur B.2: Elschema revision 2.0

C

Kostnad

Tabell C.1: Strömförsörjning, MIDI-gränssnitt och Rack.

Komponent	Antal	Styck pris (SEK)	Kostnad (SEK)
Raspberry Pi Pico	1	50,0	50,0
USB A-uttag	1	5,20	5,20
USB C-uttag	1	19,0	19,0
MIDI-uttag	1	60,0	60,0
Multiplexer	1	10	10
Optokopplare	1	25	25
3D-skrivarmaterial (PLA)	-	-	14
Summa (SEK)			183

Tabell C.2: VCO

Komponent	Antal	Styck pris (SEK)	Kostnad (SEK)
Raspberry Pi Pico	1	50	50
Display	1	187	187
Rotationsenkoder	1	29	29
Potentiometer	3	32,49	97,48
Stereouttag	4	17	68
Vippbrytare	2	12,7	25,39
PCB	1	8	8
3D-skrivarmaterial (PLA)	-	-	6,7
Summa (SEK)			471,6

Tabell C.3: VCF

Komponent	Antal	Styck pris (SEK)	Kostnad (SEK)
Raspberry Pi Pico	1	50,0	50,0
Display	1	187	187
Rotationsenkoder	1	29	29
Potentiometer	4	24,4	97,48
Skjutpotentiometer	1	32,48	32,48
Stereouttag	4	17	68
PCB	1	8	8
3D-skrivarmaterial (PLA)	-	-	6,7
Summa (SEK)			478,7

Tabell C.4: LFO

Komponent	Antal	Styck pris (SEK)	Kostnad (SEK)
Raspberry Pi Pico	1	50,0	50,0
Display	1	187	187
Potentiometer	2	12,2	24,4
Stereouttag	4	17,0	68
PCB	1	8	8
3D-skrivarmaterial (PLA)	-	-	6,7
Summa (SEK)			344,1

Tabell C.5: Mixer

Komponent	Antal	Styck pris (SEK)	Kostnad (SEK)
Raspberry PI Pico	1	50,0	50,0
Skjutpotentiometer	4	49	196
Stereouttag	5	17,0	85
PCB	1	8	8
3D-skrivarmaterial (PLA)	-	-	6,7
Summa (SEK)			345,7



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY