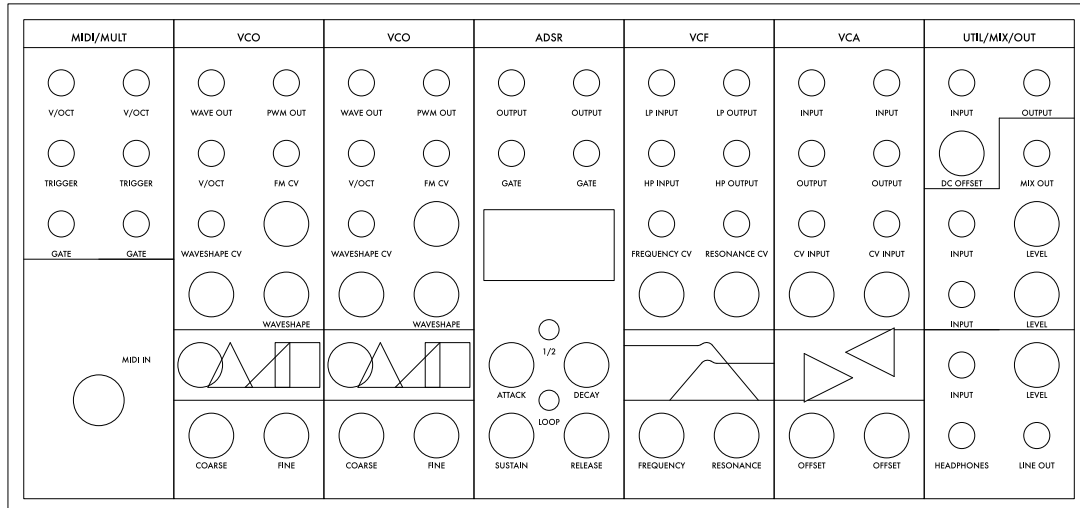




Beslagta medlen för syntes!

Kostnadseffektiv och modulär synthesizer med hjälp av digital signalbehandling

Kandidatuppsats inom datateknik

Albin Begu

Hugo Gustafsson

Felix Johannisson

Albin Johansson

Martin Jörgensen

Anton Windh

KANDIDATUPPSATS 2026

Beslagta medlen för syntes!

Kostnadseffektiv och modulär synthesizer med hjälp av
digital signalbehandling

Albin Begu
Hugo Gustafsson
Felix Johannisson
Albin Johansson
Martin Jörgensen
Anton Windh



UNIVERSITY OF
GOTHENBURG



CHALMERS
UNIVERSITY OF TECHNOLOGY

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

Beslagta medlen för syntes!

Kostnadseffektiv och modulär synthesizer med hjälp av
digital signalbehandling

Albin Begu Hugo Gustafsson Felix Johannisson Albin Johansson Martin Jörgensen
Anton Windh

© Albin Begu, Hugo Gustafsson, Felix Johannisson, Albin Johansson,
Martin Jörgensen, Anton Windh 2026.

Handledare : Erik Sintorn, Institutionen för data- och informationsteknik

Examinatorer : Patrik Jansson och Arne Linde, Institutionen för data- och informa-
tionsteknik

Rättande lärare : Arne Linde, Institutionen för data- och informationsteknik

Kandidatarbete 2026

Institutionen för data- och informationsteknik

Chalmers tekniska högskola och Göteborgs universitet

SE-412 96 Göteborg

Telefon +46 31 772 1000

Omslagsbild : En illustration av panelerna som designades för projektets syntmodu-
ler.

Typsatt i L^AT_EX

Göteborg, Sverige 2026

Beslagta medlen för syntes!

Kostnadseffektiv och modulär synthesizer med hjälp av digital signalbehandling

Albin Begu, Hugo Gustafsson, Felix Johannisson, Albin Johansson,
Martin Jørgensen, Anton Windh

Institutionen för data- och informationsteknik

Chalmers tekniska högskola och Göteborgs universitet

Sammandrag

Sedan de första modulära syntarnas utveckling har musiker drabbats av olika problem, den stora kostnaden associerad till analoga modulära syntar, bristen på insyn och interaktion med de digitala motsvarigheterna som utvecklades för att adressera den höga kostnaden. Detta projekt ämnar i att undersöka huruvida en kompromiss mellan dessa, där signalbehandlingen implementeras med digitala medel men inrespektive utgående signaler är analoga, kan vara kostnadseffektivt och troget emulera användandet av analoga syntar. Metodiken redogör för konstruktionen av ett sådant system och kan användas för att producera resultat både i form av en synt, men även i kostnaden som implementationen medför. Resultaten visar att den resulterande produkten uppfyller målen beträffande konstruktionen, och att den inte nödvändigtvis är banbrytande avseende kostnad men kan vara en prisvärd plattform för vidareutveckling. Det är möjligt att sänka kostnaden ytterligare genom val av hårdvara som är bättre lämpat för ljudbehandling vilket eliminerar behovet av kompletterande hårdvara för bättre ljudbehandling.

Abstract

Ever since the development of the first synthesizers, musicians have been plagued by two distinct problems, the large cost associated with analogue modular synthesizers and the lack of insight and interaction in their digital counterpart that was developed as a means to solve the issue of the aforementioned large cost. This project aims to examine whether a compromise between these types of synthesizers, in which the signal processing is implemented using digital means without sacrificing analogue compatibility, can faithfully emulate analogue modular synthesizers whilst retaining cost efficiency. The methodology gives an account of the construction of such a system to later produce results in terms of a finished synthesizer system and the cost that such a system entails. The results show that the resulting product is in line with the goals of construction but that it isn't necessarily groundbreaking in terms of cost but may provide a cost-effective platform for further development. It is entirely possible to reduce the cost even further with hardware that is better suited for sound processing which eliminates the need for additional hardware that is used to bridge that gap.

Nyckelord : synthesizer, synt, ljudsyntes, mikrokontroller, embedded, raspberry pi pico 2, musikinstrument, eurorack rp2350

Innehåll

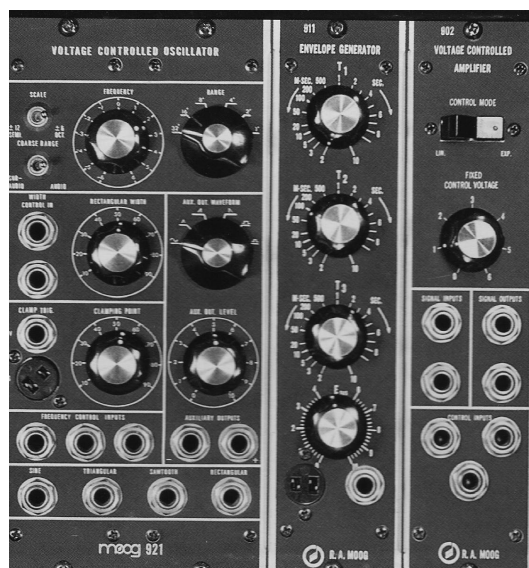
1	Introduktion	1
1.1	Syfte	2
1.2	Mål	3
1.2.1	Kostnadseffektivitet	3
1.2.2	Programmerbar funktionalitet	3
1.2.3	Kompatibilitet	4
1.2.4	Ljudkvalitet	4
2	Teori	5
2.1	Spänningsstyrd modulär synt	5
2.1.1	Styr signaler	6
2.2	Signalbehandling	6
2.2.1	Typer av signaler	6
2.2.2	Signalomvandling	7
2.2.3	Signalgenerering	7
2.2.4	Filtrering	8
2.2.5	Operationsförstärkare	9
2.3	Mikrokontroller	10
2.3.1	Raspberry Pi Pico	10
2.4	Digitala kommunikationsprotokoll	10
2.4.1	UART	11
2.4.2	MIDI	11
2.4.3	SPI	11
2.4.4	I ² S	12
2.5	Syntmoduler	12
2.5.1	MIDI-gränssnitt	12
2.5.2	VCO	12
2.5.3	VCF	14
2.5.4	VCA	14
2.5.5	ADSR	14
2.5.6	LFO	15
3	Metod	17
3.1	Hårdvara	17
3.2	Programvara	21
3.3	Gemensamma drivrutiner	21

3.3.1	Ljuddrivrutiner	21
3.3.2	Potentiometrar via multiplexer	22
3.3.3	Styrsignaler	22
3.3.4	Skärm	22
3.4	VCO	23
3.5	VCF	24
3.6	VCA	24
3.7	ADSR	25
3.8	MIDI-gränssnitt	27
3.9	LFO	27
4	Resultat	29
4.1	Evalueringsmetod	29
4.2	Konstruktion	30
4.3	Kostnadseffektivitet	31
5	Diskussion	35
5.1	Förbättringsområden	36
5.2	Slutsats	37
	Källförteckning	39
A	Kretsdiagram för en analog VCO	I
B	Kostnadstabeller	III
C	Kretsdiagram för systemets plattform	V

1

Introduktion

När ingenjören och uppfinnaren Robert Moog introducerade sin synt på 1960-talet löste han tre problem som präglade tidigare syntar, nämligen ”storlek, stabilitet och kontroll” [1, s. 208, författarnas översättning]. Genom att realisera syntens funktioner med elektronik baserad på halvledare (eng. *solid state*), gjordes systemet fysiskt hanterbart och stabilt, relativt dess föregångare. Moog lyckades åstadkomma kontroll genom att han drog nytta av sitt nätverk av bekanta inom musikindustrin, som gav synpunkter på hur instrumentet borde utformas [1]. Resultatet blev en modulerbar synt som var spänningsstyrd, bestående av ett antal diskreta moduler som på användarens initiativ kunde kopplas samman på olika sätt. Hans synt blev en succé och var ”det vanligaste instrumentet i elektroniska musikstudior under det sena 1960-talet och 1970-talet” [1, s. 208, författarnas översättning].



Figur 1.1: Moog VCO, envelope generator och VCA. [2]

Under 1970-talet påbörjades miniaturiseringen av elektroniska komponenter och till 1990-talet var de traditionella analoga syntarna ”helt efterträdade” [1, s. 272, författarnas översättning] av programmerbara, digitala syntar, tack vare kostnadseffektiviteten av integrerade kretsar och mikroprocessorer [1]. MIDI (förkortning av *Musical Instrument Digital Interface*) introducerades 1984 och gav de digitala syntarna ett gränssnitt för att styra, eller styras av, andra enheter såsom datorer [1].

År 1995 introducerade den tyske fysikern och musikern Dieter Döpfer den modulära synten A-100, vilket blev startskottet på vad som nu kallas Eurorack [3]. Eurorack är benämningen på kombinationen av A-100:s fysiska och elektriska format som under 2000-talet blev en populär formfaktor på syntmoduler, där Döpfer har konstaterat att ”det inte kunde ha förutspåtts att det skulle bli ett ’massfenomen’ med flera hundra tillverkare och långt över ett tusen olika moduler” [3, s. 352, författarnas översättning].

Sedan 2000-talet har den analoga syntens popularitet fått ett återupplivande [4]. Musiker har alltmer återgått till analoga syntar eftersom deras digitala motsvarighet, med sina förprogrammerade inställningar och sin begränsade insyn samt kontroll (s.k. ”black box-design”), har uppfattats som begränsande för musikernas kreativa utlopp. Den analoga synten uppfattas som motsatsen, där varje inställning istället har en fysisk ratt, spak eller kontakt som underlättar för musikern att uttrycka sin kreativitet. Som tidigare nämnt är dock de digitala syntarna billigare. Detta gör att tröskeln till de analoga, mindre kreativt begränsande syntarna är högre.

1.1 Syfte

Med avstamp i denna historik syftar detta projekt till att konstruera en modulär Eurorack-synt med ett analogt gränssnitt och ett digitalt innanmäte, för att dra fördel av både de analoga och digitala domänerna. På så sätt kan systemet dra nytta av den analoga syntens intuitiva och kreativa arbetsflöde samtidigt som den digitala implementationen möjliggör hög flexibilitet och kostnadseffektivitet.

Detta är tänkt att sänka den ekonomiska tröskeln till modulära syntar, genom att digitalisera signalbehandlingen utan att uppoffra den taktila och analoga upplevelsen som traditionella syntar erbjuder. Signalerna mellan moduler ska vara analoga och följa Eurorack-specifikationen för att säkerställa kompatibilitet mellan både projektets och kommersiellt tillgängliga moduler. En intern digital konstruktion möjliggör standardiserade och gemensamma kretsar för modulerna där signalbehandling överläts till mikrokontroller, vilket bör sänka kostnaden för modulerna relativt deras analoga motsvarigheter.

En av de tydligaste fördelarna med en digital modulär synt är att de individuella modulerna enkelt kan vidareutvecklas i programvara utan att fler komponenter och kretsar introduceras. Extra komponenter och utökade kretsar bidrar till en högre kostnad, och även enkla analoga funktioner kan innebära många komponenter. Se Appendix A, Figur A.1 för ett exempel på en sådan krets. En digital lösning innebär därför en lägre utvecklingskostnad för varje funktion som läggs till utan modifikation av kretsen. Digitala lösningar medför även att samma utvecklingsplattform kan återanvändas mellan olika moduler istället för att behöva utveckla olika kretsar för varje analog modul. I praktiken kan digitala syntar troget emulera analoga syntar [1], vilket resulterar i att digitala syntar utgör ett flexibelt verktyg i praktiskt bruk.

Analoga kretsar som behöver vara precisa, i synnerhet beträffande temperaturkompensering och exponentialkonvertering i VCO:er, kräver dyrare komponenter som inte alltid är lättillgängliga. Några vanligt förekommande exempel är transkonduktansförstärkare, polystyrenkondensatorer och Tempco-motstånd, där vissa av dessa komponenter dessutom befinner sig i slutet av sin livscykel [5]. Detta innebär att synt-användare tvingas välja mellan det dyrare ”analoga ljudet”, och den reducerade kostnaden som digitala syntar erbjuder men samtidigt också underminerar det kreativa arbetsflödet som står centralt för Eurorack-standarden.

Sammanfattningsvis, med bakgrund i de analoga syntarnas återupplivande och den kostnadseffektivitet som digitala syntar erbjuder [1], syftar projektet till att konstruera en hybridlösning som drar nytta av digital signalbehandling utan att uppoffra kompatibilitet eller den fysiska interaktionen som analoga syntar erbjuder. Projektet ämnar att åstadkomma detta i enlighet med egendefinierade specifikationer efter vedertagen grundläggande funktionalitet i modulernas etablerade motsvarighet.

1.2 Mål

Projektet ska resultera i en digital, modulär synt vars användande liknar det av analoga syntar. Varje modul ska erbjuda reglage och gränssnitt för användaren att konfigurera variabla parametrar respektive styra signalvägen mellan moduler.

1.2.1 Kostnadseffektivitet

Genom att i så stor utsträckning som möjligt förlita sig på mikrokontroller för signalbehandlingen samt att modulerna baseras på en gemensam hårdvaru- och mjukvaruplattform ska kostnaden för modulerna hållas nere. Denna kostnad ska med fördel för projektet kunna jämföras med en sammanställning av utvalda kommersiella moduler. Denna jämförelse ska också göras som en del av projektet. På så sätt ska en uppskattning på projektets konkurrenskraft kunna göras.

1.2.2 Programmerbar funktionalitet

Att designen baseras på digital signalbehandling och en gemensam plattform ska också resultera i att modulernas funktionalitet med enkelhet ska kunna förändras och uppdateras i programvaran. På så sätt ska ny funktionalitet och buggfixar kunna introduceras utan att behöva göra ingrepp på hårdvaran, utan endast genom att ladda upp ny programvara till hårdvaran. Detta ska också medföra att användaren med enkelhet ska kunna helt programmera om en moduls funktionalitet till en annan moduls funktionalitet. När detta görs så kan dock hårdvarugränssnittet behöva ändras, alltså kan användaren behöva ändra var ingångar, utgångar och potentiometrar leds i hårdvaran.

1.2.3 Kompatibilitet

Modulerna ska utvecklas till att följa Eurorack-specifikationen. Detta innebär att spänningsnivåerna ska finnas i intervallet $[-5, 5]$ V för ljud- och styrsignaler, och $[0, 5]$ V för digitala signaler såsom trigger-, gate- och klocksignaler [6]. Enligt den elektriska specifikationen ska spänningsnivåerna för styrsignaler ”generellt” vara mellan -2.5 V och 2.5 V. Projektet kommer istället att använda sig av spänningsnivåer mellan -5 V och 5 V för styrsignaler, för att dels kunna använda ljudsignaler som styrsignaler, men även för att minimera antalet komponenter och därmed kretsarnas storlek. Denna designavvägning är inte något som enbart förekommer inom detta projekt, utan syns även i kommersiellt tillgängliga moduler [7]. 16-poliga flatkablar med spänningarna 12 V, -12 V, 5 V och jord ska användas för strömförsörjning [6]. Modulerna ska kunna monteras i Eurorack-höljen [8].

1.2.4 Ljudkvalitet

För att upprätthålla önskvärd kvalitet på ljudet i signalkedjan mellan modulerna ska samplingsfrekvensen för de digitala ljudsignalerna inte understiga 48 kHz. Upplösningen ska vara 24 bitars bitdjup. Ljud- och styrsignalerna ska vara fria från hörbara störningar som skulle kunna uppkomma i hård- eller mjukvara. Se Avsnitt 2.2.2 för mer om varför dessa gränser valts.

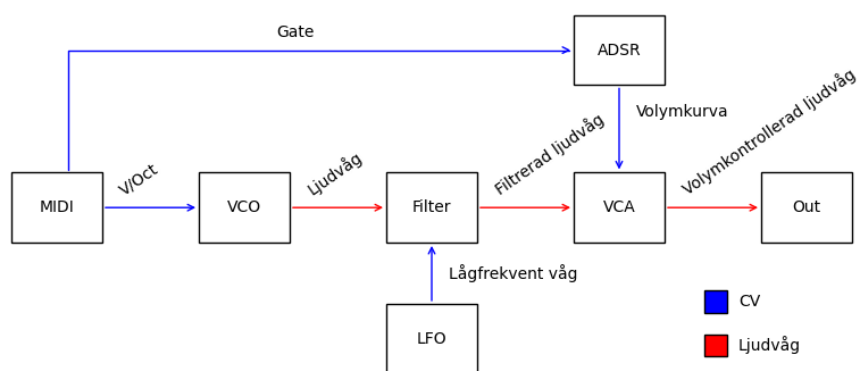
2

Teori

Detta kapitel syftar till att presentera terminologi och teori kopplat till modulära syntar implementerade som både analoga och digitala system i ett. Mer ingående förklaras relevanta typer av signalbehandling, modulernas funktioner, samt de kommunikationsprotokoll som typiskt används mellan modulerna.

2.1 Spänningsstyrd modulär synt

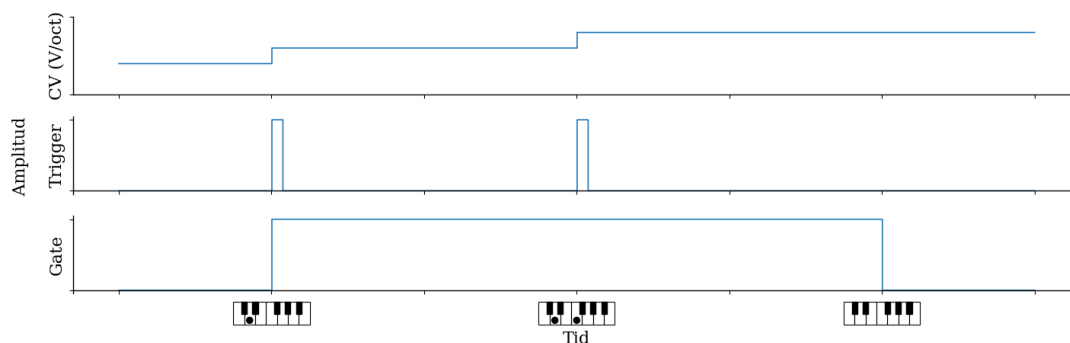
En spänningsstyrd modulär synt är en uppsättning delsystem, eller moduler, som etappvis transformerar eller producerar nya analoga signaler enligt manuellt reglerade parametrar, ingående styrspänningar, och en ingående ljudsignal (se Figur 1.1 för exempel på moduler). Beroende på hur dessa moduler är sammankopplade produceras olika slutgiltiga ljudsignaler [1]. Ett exempel på sammankoppling av moduler kan ses i Figur 2.1.



Figur 2.1: Ett diagram av hur en modulär synt skulle kunna vara uppkopplad. Lådor representerar syntmoduler och pilar representerar fysiska kopplingar mellan dem.

2.1.1 Styrsignaler

De ingående styrsignalerna utgörs i många fall av en analog styrspänning CV (fr. eng. *control voltage*), och två digitala styrsignaler, *trigger* och *gate*. Beroende på syntens elektriska specifikation antar CV ett spänningsvärde inom ett omfång av antingen $[-5, 5]$ V eller $[0, 5]$ V. Trigger är istället en digital puls som signalerar en händelse, t.ex. att en tangent trycks ner, och antar en spänning på antingen 0 V eller 5 V. Gate genereras vid en händelse, men till skillnad från trigger hålls den aktiv under hela händelsens förlopp (se Figur 2.2).



Figur 2.2: Exempel på CV, gate och trigger för en implementation av ett MIDI-gränssnitt.

2.2 Signalbehandling

Signalbehandling berör allt som har med signaler att göra från signalgenerering till manipulering av signalerna. I denna rapport behandlas flera olika typer av signaler och hur de skapas, konverteras och manipuleras för att i slutändan ge ett hörbart ljud.

2.2.1 Typer av signaler

Det finns flera olika typer av signaler men de som främst hanteras i rapporten är analoga signaler såsom ljud- och styrsignaler och digitala signaler såsom trigger, gate och seriell kommunikation. Analoga signaler varierar i spänningsvärde mellan ett minimum och ett maximum och kontinuerligt befinner sig mellan dessa. Digitala signaler brukar varierar mellan två olika lägen, ett högt spänningsvärde och ett lågt (oftast kallat *high* och *low*). Signalerna har ett par egenskaper knutet till sig som definierar dem. Avståndet en signal har från sitt nolläge kallas dess *amplitud* och ett exempel på det är ljudvolym. En signal kan även vara periodisk vilket betyder att den upprepas i tiden. Varje periodisk signal har en *frekvens*, vilket är hur ofta den upprepas och mäts i gånger per sekund eller hertz [Hz]. En signal kan innehålla ett flertal olika frekvenser och det är dessa frekvenser som människan hör som toner. Människor tenderar att kunna höra frekvenser mellan 20 Hz till 20 kHz.

2.2.2 Signalomvandling

I många fall behöver en signal omvandlas till en annan form för att kunna bearbetas. Processen av att dela upp en analog signal i diskreta värden kallas *sampling*. Detta görs av en så kallad *analog-to-digital converter* (förkortas ADC) [1]. För omvänd konvertering, från digital till analog, används en så kallad digital-to-analog converter (förkortas DAC). Hårdvara som konverterar åt båda hållen kallas ”kodek” och kommer från en sammansättning av orden *coder* och *decoder*.

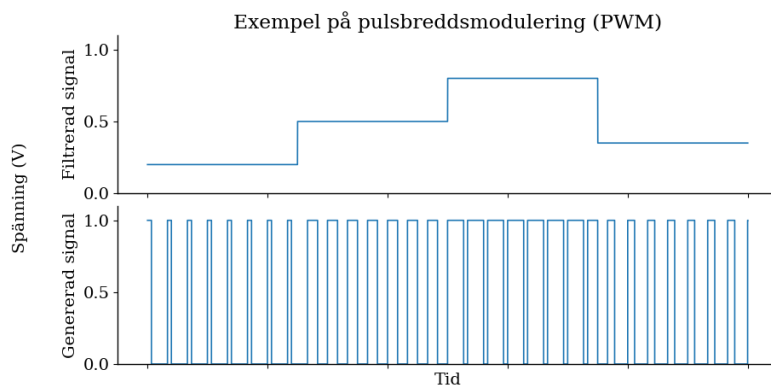
Vid omvandling av en signal från kontinuerlig (analog) till en diskret (digital) representation av signalen krävs det att avläsningsfrekvensen är minst dubbelt så hög som den högsta frekvensen i ursprungssignalen. Anledningen till detta beskrivs av Nyquist-Shannons samplingsteorem, som säger att en vågrörelse måste mätas med minst dubbla sin frekvens för att kunna återskapas pålitligt [9]. I praktiken innebär samplingsteoremet att analoga signaler behöver samplas med en samplingsfrekvens på minst 40 kHz för att kunna återskapa signalen för det mänskliga frekvensområdet. 48 kHz tenderar att användas som samplingsfrekvens för högkvalitativt ljud [10].

För varje sample som mäts av en ADC kvantiseras amplituden till ett värde. Hur många olika värden som kan representeras beror på bitdjupet, alltså hur många bitar som används för att representera amplitudintervallet. Vanliga bitdjup för ljud är 16- eller 24 bitar. Då kan amplituden variera som heltal i intervallet $[-32\,768, 32\,767]$ för 16 bitar och $[-8\,388\,608, 8\,388\,607]$ för 24 bitar.

2.2.3 Signalgenerering

För att generera styrsignaler, reglera parametrar, och driva kommunikation mellan elektroniska enheter, krävs system för signalgenerering. Tekniken som används i en signalgenerator kan skilja sig beroende på generatorns underliggande teknologi och funktionalitet. Till exempel kan en digital signalgenerator generera en analog signal med hjälp av en DAC, eller genom pulsbreddsmodulering (förkortas PWM, fr. eng. *pulse width modulation*). Den initiala digitala signalen kan i detta fall genereras genom regelbaserade metoder som beskriver hur signalen förändras över tid, eller genom exempelvis vågformstabeller (se Avsnitt 2.5.2), för att sedan göras analog genom PWM.

Pulsbreddsmodulering utnyttjar trögheten i ett digitalt elektriskt system för att mata ut en analog signal genom att alternera utsignalen mellan ett högt och lågt värde (se Figur 2.3). Det tidsmässiga förhållandet mellan det höga och låga värdet under en period kallas för pulskvoten, vilket avgör medelvärdet av den utmatade digitala signalen. I praktiken kan analoga kontinuerliga signaler approximeras genom att konstant justera pulskvoten. För att ytterligare jämma ut signalen används ofta ett enkelt lågpasfilter, vilket släpper igenom låga frekvenser (se 2.2.4). Filtringen består oftast av ett så kallat RC-filter som består av en resistor och en kondensator.



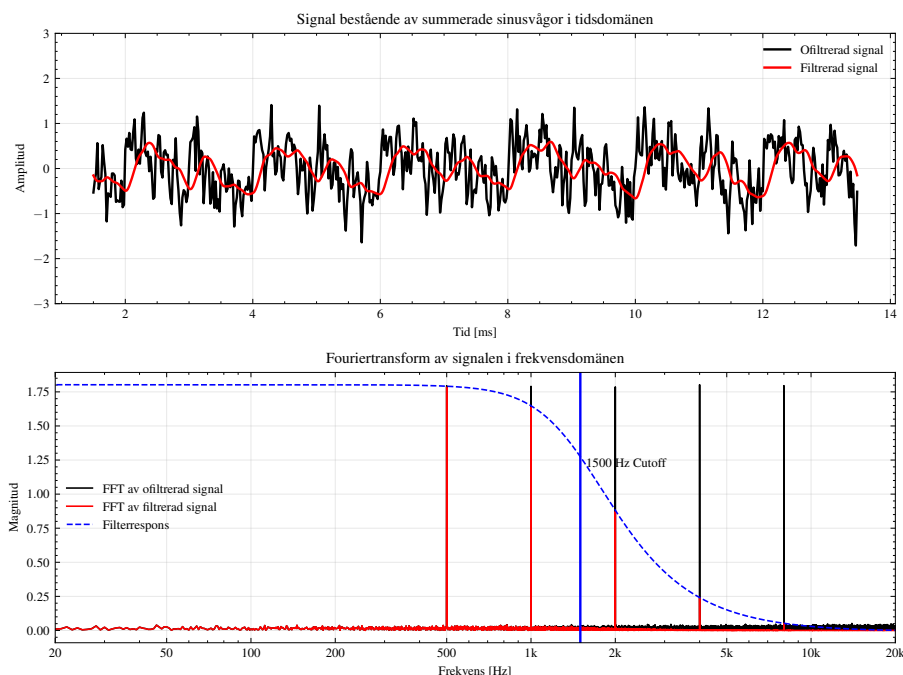
Figur 2.3: Illustration av hur en PWM-signal utformas. Den nedre grafen representerar de värden en PWM-generator matar ut för att uppnå de önskade värdena representerade i den övre grafen.

2.2.4 Filtrering

I många fall är vissa frekvenser i en signal inte önskvärda vilket kan åtgärdas med filter. Ett typiskt filter attenuerar vissa frekvenser och låter andra frekvenser passera. Ett filter som attenuerar höga frekvenser och låter lägre frekvenser passera kallas ett lågpasfilter (se Figur 2.4). Motsatsen till lågpasfilter kallas högpasfilter och en kombination av dessa kallas bandpassfilter, där ett frekvensintervall eller frekvensband släpps igenom medan resten attenueras. Den frekvens vid vilken ett filter halverar insignalen (en skillnad på 3 dB) kallas för gränshfrekvens eller cutoff-frekvens.

Ett filters karaktär kan enklast visualiseras i frekvensdomänen, där i stället för en tid/amplitud-graf (tidsdomän) har man en frekvens/amplitud-graf. För att konvertera mellan tidsdomän och frekvensdomän används Fouriertransformen. Frekvensdomänen visar tydligare hur ett filter påverkar en signal, då frekvensernas amplitud under eller över en gränshfrekvens attenueras respektive släpps igenom när filtret är aktivt.

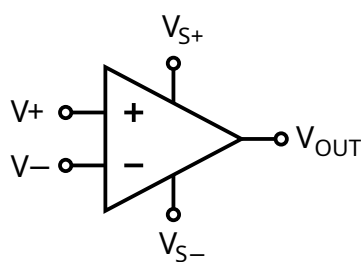
Det finns ett antal olika sätt att applicera ett filter på en signal. För applikationer där hela signalen redan finns är det vanligt att applicera ett filter i frekvensdomänen genom multiplikation. När filtrering ska ske i realtid och applikationen är känslig för fördröjningar, såsom med ljud, används bland annat IIR-filer (*infinite impulse response*) på grund av deras realtidsprestanda. Dessa filter beräknas kontinuerligt medan en signal mäts och filtreras för att omvandla ett ofiltrerat sample till ett filtrerat sample. Genom att minnas tidigare samples som bearbetats med filtret kan det räknas ut hur nästa sample ska filtreras.



Figur 2.4: Exempel på en signal i tidsdomänen och frekvensdomänen både ofiltrerad och filtrerad. Signalen filtreras av ett låpassfilter med gränshäns 1.5 kHz

2.2.5 Operationsförstärkare

Signalbehandling i elektriska kretsar innefattar ofta konvertering mellan olika spänningsnivåer, vilket kan uppnås med hjälp av operationsförstärkare, även kallade op-amp:ar (fr. eng. *operational amplifier*).



Figur 2.5: Symbolen för en op-amp, med ingångarna V_+ och V_- och spänningarna V_{s+} och V_{s-} som utgör spännet som den utmatade spänningen V_{OUT} befinner sig inom. [11]

En operationsförstärkare förstärker skillnaden mellan sina två ingångar och matar ut detta på utgången. Genom att återkoppla utgången med en av ingångarna kan operationsförstärkaren användas för att förstärka ingående signal från den andra ingången. För att kunna mata ut en signal med lika brett spänningsintervall som

operationsförstärkaren matas med används en så kallad rail-to-rail-opamp (översätts till räls-till-räls-operationsförstärkare, se Figur 2.5). Med andra ord blir en rail-to-rail-opamp som matas med 5 V och -5 V mättad vid dessa spänningar och kan därmed endast mata ut spänningar inom detta intervall.

2.3 Mikrokontroller

Den digitala signalbehandlingen, som utgör en central del av systemet, utförs av en mikrokontroller (förkortas MCU, fr. eng. *microcontroller unit*). Mikrokontrollern styrs av programmerbar mjukvara och fungerar i praktiken som en liten dator. Kommunikation mellan mikrokontrollerns periferienheter sker genom väldefinierade protokoll såsom SPI (*Serial Peripheral Interface*), I²C (*Inter-Integrated Circuit*) och UART (*Universal Asynchronous Receiver-Transmitter*). Mikrokontrollrar har ett antal pinnar med olika funktioner där vissa är relaterade till exempelvis strömförsörjning av chippet, medan andra stödjer funktioner som kan vara mer relevanta vid konstruktion av en synt.

2.3.1 Raspberry Pi Pico

En tillverkare av mikrokontrollrar är Raspberry Pi. Deras mikrokontroller Pico, som är ARM-baserad, har en proprietär funktion som kallas för PIO. PIO är ett system för att köra program oberoende av mikrokontrollerns exekvering [12]. Programmen skrivs i ett assembler-liknande språk och med programmet associeras en eller flera GPIO-pinnar som programmet sedan har möjlighet att asynkront läsa från och skriva till.

PIO lämpar sig för att implementera drivrutiner för protokoll i och med att det är möjligt att med assembler-programmet sköta synkronisering, in- och utmatning på en handfull associerade GPIO-pinnar. På detta sätt avlastas processorn från att behandla detta och kan istället ägna sina resurser åt annat.

För att sedan sköta läsning och skrivning till Picons minne används DMA (*direct memory access*). Med DMA kan Picon bulkvis behandla data som ska skrivas till eller läsas från minnet eller andra periferienheter. Även här avlastas Picons processor, vilket gör att den kan ägna sig åt annat än att minnesord för minnesord behandla data [12].

2.4 Digitala kommunikationsprotokoll

Kommunikation mellan sammankopplade enheter sker enligt etablerade kommunikationsprotokoll. För integrerade kretsar finns ett antal olika protokoll med olika för- och nackdelar. De protokoll som använts i projektet implementerar seriell kommunikation vilket innebär att data skickas över tid på ett fåtal linor till skillnad från parallell kommunikation som skickar varje individuell bit som utgör datan på var sin lina.

2.4.1 UART

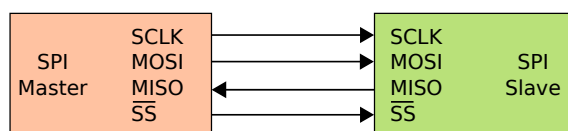
För enkla tillämpningar används UART (*Universal Asynchronous Receiver-Transmitter*). Som namnet avslöjar finns det ingen klocksignal som synkroniserar sändaren och mottagaren, utan en förutbestämd baudhastighet (hur snabbt bitar kan förändras) används för kommunikationen. UART fungerar endast för kommunikation mellan två komponenter, medan andra protokoll stödjer fler enheter.

2.4.2 MIDI

MIDI (*Musical Instrument Digital Interface*) är ett protokoll för kommunikation mellan musikproduktionsutrustning [13]. Bland de enklaste exemplen är när ett klaviatur ska meddela att en ton spelas, medan mer avancerade användningsområden, såsom extern styrning av inställningar hos en synt, också förekommer. MIDI bygger på UART-kommunikation med en baudhastighet på 31.25 kbaud, genom ett meddelandeformat som dikteras av MIDI-standarderna [13]. MIDI-meddelanden består av en statusbyte följt av 0-2 databytes. Statusbyten innehåller vilken typ av meddelande som skickas, varpå datan som meddelandet innehåller skickas i efterföljande databyte(s). Till exempel kan statusbyten innehålla typen `note_on` för att signalera att en tangent tryckts ned, varpå efterföljande databyte innehåller information om vilken ton det motsvarar.

2.4.3 SPI

För applikationer som kräver snabb och synkroniserad kommunikation mellan controller och periferienheter samt en gemensam klocksignal lämpar sig SPI (*Serial Peripheral Interface*) som kommunikationsprotokoll.



Figur 2.6: Diagram som beskriver en simpel SPI-koppling mellan en controller ("master") och ett mål ("slave"), oftast en periferienhet. [14]

Den ursprungliga specifikationen definierades av Motorola under 1980-talet [15]. Enligt den använder den mest grundläggande implementationen av SPI tre linor som buss, och en lina för att välja periferienheten för kommunikation [16]. En av linorna är en klocklina (SCLK) för att hålla all kommunikation synkron. Två datalinor, Master Out Slave In och Master In Slave Out (MOSI och MISO), möjliggör simultan dubbelriktad kommunikation, så kallat "full duplex". Dessa tre linor utgör SPI-bussen som alla periferienheter delar. Slave Select ($\overline{SS}$) väljer vilken enhet som interagerar med kontrollern över SPI-bussen. Därmed behöver kontrollern en lina för varje periferienhet som den behöver kommunicera med.

Synkroniseringen av SPI-protokollets dataöverföring kan ställas in på olika sätt. Klocksignalens polaritet bestämmer hur den utmatade kontrollspänningen motsvarar till logiska värden, medan klocksignalens fas bestämmer när data matas ut och

när den registreras i mottagande enhet. Med andra ord kan SPI-protokollet ställas in till att mata ut en ny bit från den givande enheten när klocksignalen går från logisk nolla till logisk etta, och att den mottagande enheten läser biten när klocksignalen skiftar tillbaka, eller tvärtom. Figur 2.6 visar hur dessa enheter kopplas samman.

2.4.4 I²S

Protokollet I²S (*Inter-Integrated Circuit Sound*) är designat för seriell digital överföring av två-kanaligt ljud [17]. För I²S krävs tre anslutningar: SCK (*continuous serial clock*), WS (*word select*) och SD (*serial data*). SCK och WS är klocksignaler som används för synkronisering, medan SD är signalen för själva ljuddata. WS bestämmer vilken av kanalerna som är aktiv, varpå SCK sedan matar fram ljuddata på SD bit för bit. Det blir därför WS som bestämmer samplingsfrekvens och SCK som bestämmer ordlängd. Enheten som genererar SCK och WS kallas *controller*, medan enheten som mottar på dessa kallas *target* [17].

2.5 Syntmoduler

Detta avsnitt ägnas åt att ge en överblick över några av de vanligaste syntmodulerna. Följande samling av moduler kan anses vara en minimal uppsättning för grundläggande funktionsduglighet. [1]

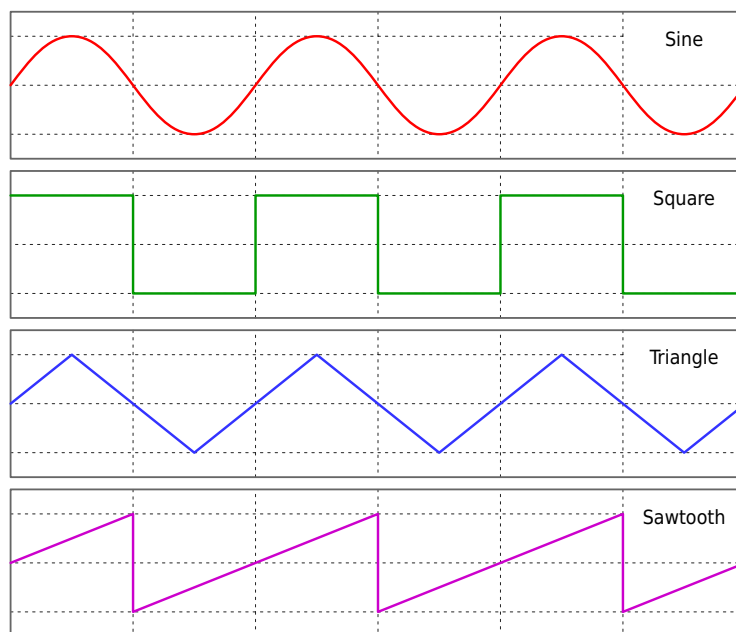
2.5.1 MIDI-gränssnitt

Ett MIDI-gränssnitt konverterar inkommande MIDI-signaler till styrsignalerna CV (1 V/oct), trigger och gate. Källan till MIDI-signaler kan exempelvis vara klaviatur, sequencers eller andra former av MIDI-styrenheter. MIDI-gränssnittets utsignaler kan användas som styrsignaler för modulerna i enlighet med vad som är beskrivet i respektive avsnitt nedan.

2.5.2 VCO

En VCO, (fr. eng. *voltage-controlled oscillator*, översätts till spänningsstyrd oscillator), genererar periodiska signaler inom det mänskligt hörbara frekvensområdet för ljud. En ingående styrspänning, som ofta produceras från ett MIDI-gränssnitt, signalerar vald frekvens på den utmatade signalen från VCO:n [1]. Styrspänningen kan exempelvis genereras av ett klaviatur (se Avsnitt 2.5.1), vilket ger möjligheten att använda synten som ett musikaliskt 12-halvtöns-instrument. Många konstruktörer av spänningsstyrda oscillatorer definierar dess funktion sådant att 1 V motsvarar en hel oktav, alltså 1 V/oktav. Varje oktav delas upp i 12 noter som motsvaras av 12 diskreta spänningsnivåer jämnt utspridda över ett spann på 1 V.

Dessutom kan funktionaliteten av en VCO inkludera syntes av olika typer av vågformer och även frekvensmodulering. Klassiska vågformer innefattar sinus-, fyrkants-, triangel- och sågtandsformer (se Figur 2.7).



Figur 2.7: Exempel på sinus- (sine), fyrkants- (square), triangel- (triangle), och sågtandsvågor (sawtooth). [18]

Frekvensmodulering innebär att en utgående signals frekvens skiftas beroende på en moduleringsparameter, ofta reglerad av en ingående analog styrsignal med en relativt lågfrekvent oscillation. Med andra ord utgör frekvensmoduleringens funktion en sorts parametrering av parametrarna som styr utsignalens frekvens i sig. I kontexten av VCO medför det möjligheten att generera signaler vars frekvenser oscillerar över tid, vilket möjliggör konstruktion av mer komplexa signaler. Exempelvis, om en ljudsignal frekvensmoduleras med en lågfrekvent oscillerande styrsignal, resulterar det i periodiska skiften i tonhöjd kring dess grundton. Samma modulering kan göras även med utsignalens amplitud, så kallad amplitudmodulering.

Beträffande fyrkantsvågor kan dess pulskvot moduleras med en styrsignal vilket kallas pulsbreddsmodulering. Pulskvoten avser det tidsmässiga förhållandet mellan högt, respektive lågt signalvärde under en period. Tekniken används också för att emulera analoga signaler med hjälp av digitala signaler, vilket beskrivs i Avsnitt 2.2.3, men i kontexten av VCO används tekniken för att förändra en utgående fyrkantsvågs karaktär.

För att digitalt generera vågformer kan s.k. *wavetable synthesis*, eller vågformstabeller, användas [1]. Vågformstabeller lagrar en period med stickprov för vardera vågform i mikrokontrollerns minne. Sedan matas stickproven för en vald vågform sekventiellt in i en DAC för att konverteras till en analog signal. Frekvensen av den genererade signalen beror på hastigheten som mikrokontrollern stegar igenom stickproven med. I praktiken görs detta genom att variera steglängden genom stickproven. Därmed förutsätts att de stickprov som finns definierade i vågformstabellen har en hög nog upplösning, med en minsta upplösning som definieras av Nyquist-Shannons samplingsteorem (se Avsnitt 2.2.2).

2.5.3 VCF

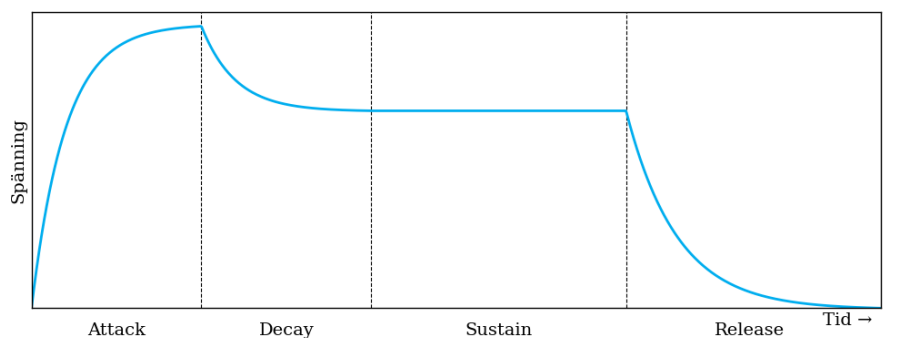
En VCF (fr. eng. *voltage-controlled filter*, översätts till spänningsstyrt filter), filtrerar signaler. Styrspänningen för denna modul styr gränshfrekvensen, eller med andra ord frekvensen varifrån högre eller lägre frekvenser attenueras, beroende på filtertyp. Filtertypen kan ändras med reglage mellan t.ex. lågpass, högpas, bandpass och bandstopp. Exempelvis filtreras frekvenser högre än brytfrekvensen bort för ett lågpass-filter, medan frekvenser lägre än brytfrekvensen för ett högpas-filter filtreras bort. Bandpass-filter konserverar ett visst omfång av frekvenser och attenuerar resten, medan bandstopp-filter gör motsatsen. Vidden och gränshfrekvens för detta omfång bestäms av ingående styrspänningar och reglage. Även filtrets resonans, alltså förstärkningen kring brytfrekvensen, kan justeras med hjälp av reglage. [1]

2.5.4 VCA

En VCA (fr. eng. *voltage-controlled amplifier*, översätts till spänningsstyrd förstärkare) används för att med hjälp av en styrspänning styra amplituden (alltså spänningen) för en ingående signal [1]. Med en styrspänning kan man alltså bestämma amplituden på ljudet som släpps igenom från VCA:ns ingång till dess utgång. Denna modul används ofta tillsammans med en ADSR-modul (se Avsnitt 2.5.5) för att styra hur ljudet utvecklas från det att en ton börjar spela tills den avslutas.

2.5.5 ADSR

ADSR-modulen (fr. eng. *attack, decay, sustain, release* och kallas även *envelope generator*, vilket översätts till konturgenerator), genererar vid inkommande gate-signal en styrsignal enligt fyra parametrar: *attack*, *decay*, *sustain* och *release* [1]. *Attack* bestämmer hur snabbt styrsignalen ska stiga till maxnivå. *Decay* reglerar hur snabbt, efter att den nått maxnivå som signalen ska sjunka till nivån som bestäms av *sustain*. Signalen hålls vid *sustain*-nivån tills dess att gate blir låg, varvid *release* bestämmer hur snabbt signalen sjunker ner till miniminivån. Se Figur 2.8 för exempelillustration. Modulen kan användas för att närmare emulera akustiska instrument som på olika sätt besitter dessa egenskaper som exempelvis fioler vars attack är beroende av hur snabbt stråken dras på strängarna.



Figur 2.8: Representation av de olika stegen i en ADSR från det att gaten aktiveras tills spänningen avtagit.

Denna styrsignal kan exempelvis användas som in-signal till VCA:n som då matar ut sin ingående ljudsignal enligt konturen av ADSR-signalen. Matas då t.ex. vågformen från VCO:n in i VCA:n, ändras den konstanta utsignalen från VCO:n till att endast matas ut när ADSR-modulen aktiveras och då i form av ADSR-parametrarna.

2.5.6 LFO

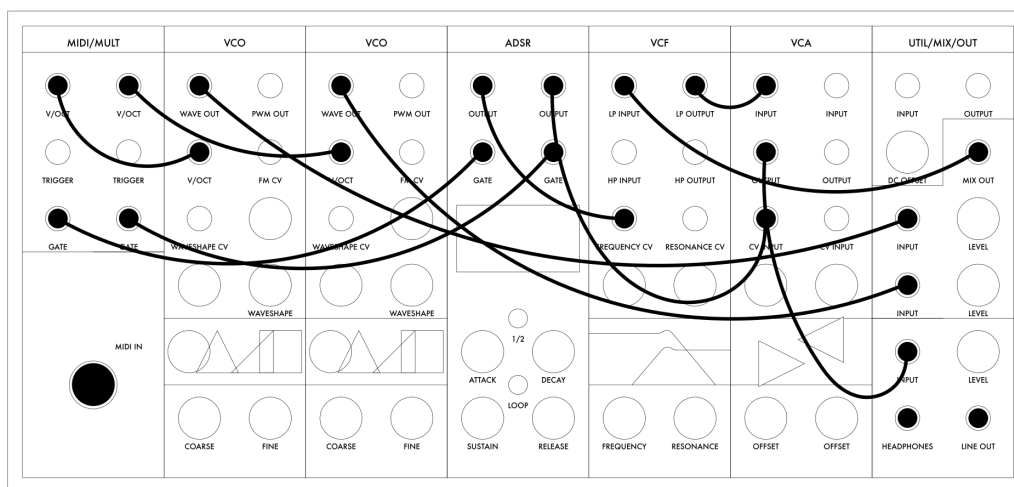
LFO (fr. eng. *low-frequency oscillator*, översätts till lågfrekvensoscillator) är, likt VCO:n, också en oscillator, men istället för att generera signaler inom det hörbara frekvensintervallet genereras signaler som har lägre frekvens. Denna signal kan i sin tur användas som styrsignal för andra moduler [1]. Exempelvis kan LFO:n modulera VCA:n för att få en tremolo-effekt, modulera VCO:n för att få en vibrato-effekt eller modulera VCF:et för att få ett periodiskt svep av filtrets gränshfrekvens. Även för LFO:n är det ofta möjligt att ändra vågformen på dess utgående signal.

3

Metod

Detta avsnitt redogör för projektets konstruktionsdel och syftar till att ge en överblick av mjukvaru- och hårdvaruutvecklingen för projektets övergripande delar samt utvecklingen av de enskilda syntmodulerna. Projektets konstruktion delades upp i tre olika steg: hårdvara, drivrutiner och syntmoduler, där de senare steget bygger på valen som gjorts i det tidigare.

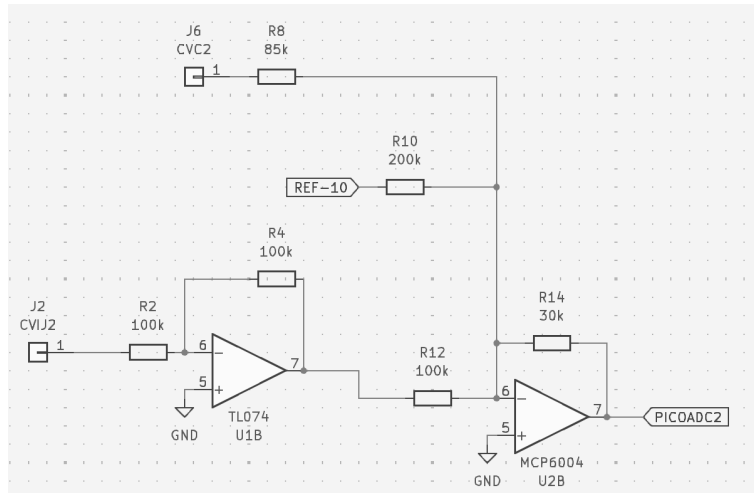
3.1 Hårdvara



Figur 3.1: Visualisering av hur en patch med liknande signalflöde som en Moog Model D kan skapas på systemet. Något som kan generera MIDI-signaler kan kopplas in, t.ex. ett klaviatur, och ljudet kommer ut ur de två utgångarna längst ner till höger.

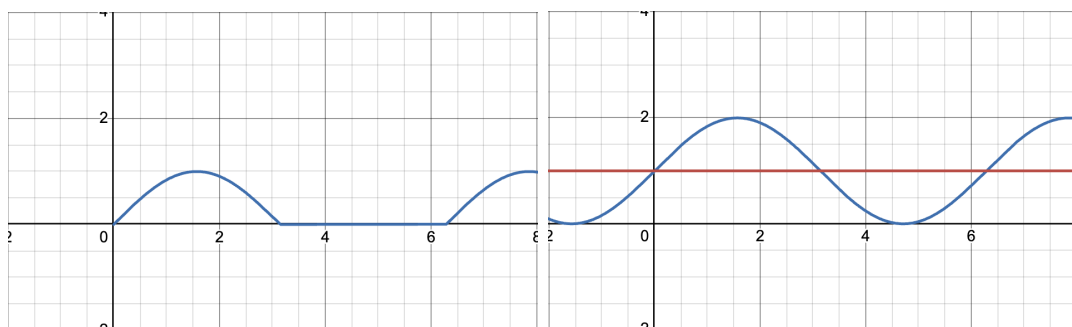
Figur 3.1 visar ett exempel på hur signalvägar kan kopplas i projektets Eurorack-synt. I hänsyn till målet att den färdiga produkten ska kunna integreras i ett Eurorack-system är konverteringar mellan spänningsnivåer med hjälp av analog hårdvara nödvändiga. I Appendix C, Figur C.1, C.2 och C.3, redovisas det kompletta kretsschemat för systemet, men detta kapitel behandlar individuella delkretsar.

Projektets mikrokontroller klarar av spänningsnivåer mellan 0 V och 3.3 V, men inte negativa spänningar vilket är vanligt förekommande inom Eurorack-system. I hänsyn till detta skapades ett analogt steg med en rail-to-rail-opamp till modulerna, se Figur 3.2, vilket innebär att förstärkarens utsignal kan vara nära matningsspänningens nivå, vilket inte är möjligt med vanliga operationsförstärkare.



Figur 3.2: Kretsschema för ingångssteg till systemet. J2 och J6 benämner eventuella ingångar som kan användas. Samma design, med minimala modifieringar användes även för ljudingången.

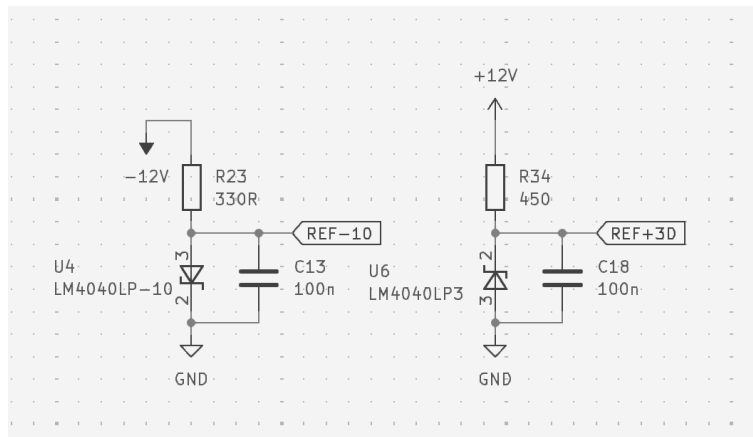
Genom att förse förstärkaren med 0 V och 3 V, begränsas insignalen till samma omfång. Ytterligare ett förstärkarsteg med förstärkningsförhållandet 1:1 lades till i hårdvarudesignen för att systemet ska vara fasriktigt och för att förse ingången med ett skydd mot felkoppling i form av hög impedans. En likspänningsnivå introducerades också mellan de två förstärkarstegen för att den totala växelspänningssignalen ska kunna nå mikrokontrollern, och inte enbart den positiva delen av perioden, se Figur 3.3.



Figur 3.3: Grafisk representation av hur en offset-spänning markerat med rött behöver introduceras för att en komplett sinusvåg ska kunna hanteras av mikrokontrollern som inte stödjer hantering av negativa spänningar. Notera att i systemet används andra amplituder än i detta exempel.

För detta projekt valdes Raspberry Pi Pico 2 som mikrokontroller för samtliga moduler på grund av dess beräkningskapacitet och stöd för hårdvarufunktioner som tillhandahåller både standardgränssnitt och alternativa sätt att implementera digital kommunikation. Mikrokontrollern tillhandahåller 26 GPIO-pinnar med hårdvarustöd för SPI, UART, ADC, PWM och PIO, vilket tillfredsställer projektets behov. Tre GPIO-pinnar anslöts till ADC:n där dessa användes för inmatning av analog ljuddata, kontrolldata och potentiometervärden till digitala värden. Mikrokontrollern möjliggör att 16 av GPIO-pinnarna används för att skicka ut PWM-signaler vilket användes för att skapa en primitivare DAC.

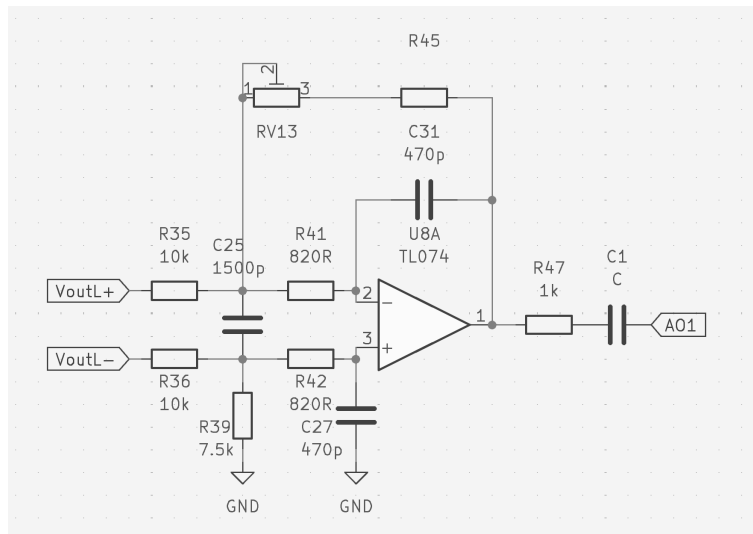
Modulerna designades även till att inkludera analoga kretsar för spänningsreferenser som används av mikrokontrollerns ADC för både likspänningsnivåer och kontroll med potentiometrar. Spänningsnivåerna 3 V och -10 V användes som referenser. Referenserna används som stabila nivåer i jämförelse med vanliga spänningsregulatorer som är mindre stabila, se Figur 3.4.



Figur 3.4: Kretsschema för spänningsreferenserna som använts i systemet.

I analoga ingångssteg som behöver hantera både potentiometrar i form av externa reglage och insignaler som kontrollspänning, användes analoga potentiometrar som spänningsdelare som sedan summeras tillsammans med en eventuell kontrollsignal innan nivåerna attenueras till den nivå som mikrokontrollern kan hantera med hjälp av rail-to-rail-opampen. Eftersom opampens matningsspänning bestämdes till 0 V och 3 V, säkerställer denna lösning att ADC:n får spänningar som den kan hantera. Eventuell signalklippning eller förlust sker analogt, och resulterar enbart i avsaknad av analog information istället för att introducera digitala störningar eller brus vilket hade inträffat om ADC:n hade behövt hantera signaler som inte förväntas.

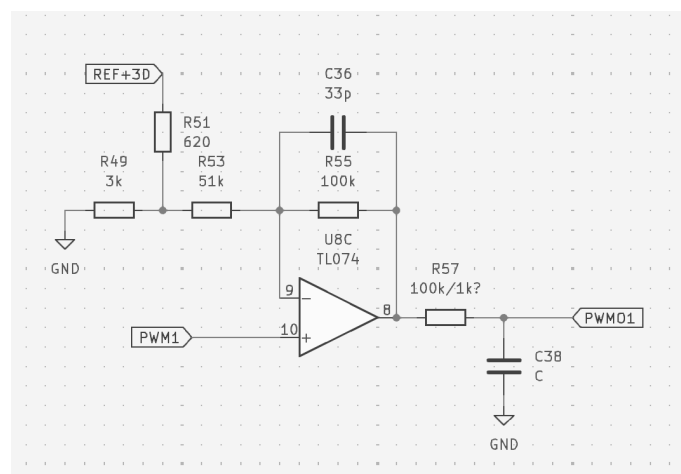
Kretsen som hanterar ljudutmatningen, se Figur 3.5, tillhandahölls från databladet för PCM3060 [19], vilket är kodeken som använts till systemet.



Figur 3.5: Kretsschema för utgångssteget för ljud i systemet.

En kodek har både ADC-ingångar och DAC-utgångar, vilket gör den optimal för tillämpningar där behov för båda funktioner krävs. Den är även specifikt anpassad för ljudsignaler och har 24-bitars upplösning för bra ljudkvalitet [19]. En trim-potentiometer användes för att göra finjusteringar för att säkerställa att nivån är mellan -5 V och 5 V som förväntat. En kondensator placerades på utgången för att ta bort eventuella likspänningsnivåer i signalen. En konvention inom Eurorack är att ha ungefär $100\text{ k}\Omega$ ingångsimpedans och $1\text{ k}\Omega$ utgångsimpedans, därför placerades motstånd på in- och utgångarna för att uppnå detta.

För hantering av kontrollsignaler användes samma krets som för ljudingången och en annan krets placerades för modulernas utgående kontrollsignaler där en likspänningsnivå adderas och förstärkning används för att konvertera en PWM-signal på intervallet $[0, 3.3]\text{ V}$ till antingen $[-5, 5]\text{ V}$, eller $[0, 5]\text{ V}$, se Figur 3.6.



Figur 3.6: Kretsschema för den primitiva DAC:en som använts för att konvertera PWM-signalerna.

Denna signal filtreras sedan för att ge en jämnare utgång. Gate- och trigger-signaler skickas till transistorbasen som används som omkopplare, vilket resulterar i en enkel hög eller låg till mikrokontrollerns GPIO.

3.2 Programvara

Modulernas programvara utvecklades i programmeringsspråket Rust med hjälp av ramverket Embassy. Embassy möjliggjorde abstraktion av den direkta interaktionen med hårdvaran vilket gör det möjligt att på sätt i enlighet med Rusts design interagera med mikrokontrollerns hårdvara. Det vill säga att gränssnitt för de inbyggda ADC:erna, GPIO, PIO, SPI och UART exponeras och kan i sin tur användas för att bygga modulerna. Ramverket bistår också med funktioner för asynkron programmering som möjliggör uppdelning av kod i *tasks* som körs oberoende av varandra. Även stöd för kommunikation mellan dessa *tasks*, t.ex. kanaler och mutexes, finns att tillgå.

3.3 Gemensamma drivrutiner

I och med att modulernas hårdvara i stor utsträckning utvecklades till att vara gemensam, innebär detta att även viss kod blir det. Detta innefattar drivrutiner för in- och utmatning för ljud respektive styrsignaler, inmatning av värden från potentiometrar genom en multiplexer, och slutligen styrning av skärmen.

3.3.1 Ljuddrivrutiner

Ljuddrivrutinernas uppgift är att säkerställa att analoga ljudsignaler som mottas av hårdvarukodekens ingångar kan läsas som digitala ljudsignaler och behandlas i programvaran samt att digitala ljudsignaler som skapas i programvaran kan matas ut som analoga ljudsignaler genom kodekens utgångar. Kodeken, i kombination med de analoga periferikretsarna, sköter själva omvandlingen mellan analoga och digitala signaler och även kommunikationen mellan kodeken och mikrokontrollern är därför digital. Denna digitala kommunikation sker med protokollet I²S.

Drivrutinerna implementerades med tre PIO-program: ett som genererar huvudklockan (denna sköter synkronisering av hårdvarukodeken och är utanför I²S-protokollet), ett som sköter inmatning och ett som sköter utmatning. PIO-programmen för huvudklockan och utmatning tillhandahölls av Embassys inbyggda bibliotek med PIO-program, medan programmet för inmatning implementerades på egen hand. Synkroniseringen för in- respektive utmatning sker på två olika sätt. Kodekens ADC konfigurerades att köras i *controller mode*, vilket innebär att ADC:n genererar WS-klocka och SCK-klocka utifrån huvudklockan, som den får som inmatning från mikrokontrollern. Det är sedan inmatningsdrivrutinens uppgift, utifrån WS- och SCK-klockorna, att vid rätt tidpunkt läsa datan från kodeken. I och med att Embassy inte hade något färdigt program för inmatning med ADC i *controller mode* behövde just detta egenimplementeras. Kodekens DAC konfigurerades istället att

köras i *target mode*, vilket innebär att kodeken endast sköter synkronisering med hjälp av externa klocksignaler. Det blir då utmatningsdrivrutinens uppgift att sköta synkroniseringen av DAC:en genom att vid rätt tidpunkter säkerställa att klockorna genereras och att utmatning till DAC:en följaktligen sker vid rätt tidpunkt.

3.3.2 Potentiometrar via multiplexer

Värden som matas in till modulerna genom potentiometer (utöver de analoga potentiometrarna vid CV-ingångarna) går via en 8-till-1-multiplexer, vilket möjliggör läsning av värden från 8 stycken potentiometrar genom en ADC-ingång. Drivrutinen för detta abstraherar bort kontrollen över multiplexern och gör det möjligt att direkt läsa värdet från en specificerad ingång på multiplexern. Drivrutinen sköter då kontroll av multiplexern (d.v.s. val av ingång) med 3 GPIO-pinnar, vilket gör det möjligt att välja mellan 8 ingångar till multiplexern, varpå ADC:en läses. I plattformen är en av RP Picons ADC:er dedikerad till att avläsa multiplexern om en sådan behöver inkorporeras. Endast potentiometrar som används som spänningsdelare har testats i projektet, men andra signaler kan vara en vidareutveckling.

3.3.3 Styrsignaler

För inmatning av styrsignaler stöder drivrutinen läsning från de övriga två av mikrokontrollernas inbyggda ADC-kanaler samtidigt. Drivrutinerna möjliggör läsning av styrsignalerna genom DMA, vilket resulterar i att en samplingsfrekvens på 48 kHz kunde användas. Med hänsyn till en begränsning i mikrokontrollern, att ADC:erna inte kunde läsas med både DMA och icke-DMA samtidigt, kunde inte heller multiplexern (i och med dess timing-drivrutin som inte fungerar med DMA) och DMA användas samtidigt. Detta resulterar i att ADC:erna inte kan läsas med en samplingsfrekvens på 48 kHz om modulen också behöver potentiometrar via multiplexern. Detta åtgärdades genom att vid användning av multiplexern istället låta läsning ske genom att välja att läsa övriga ADC:er i samband med att ljudbuffern behandlas, förutsatt att tiden som ägnas åt ADC-läsningar inte överstiger ljudbufferperioden.

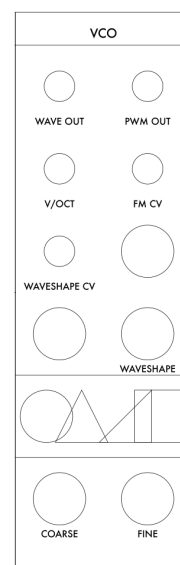
PWM användes för utgående styrsignaler. Den gemensamma koden för detta är minimal eftersom Embassy redan har inbyggt stöd för att använda mikrokontrollernas pinnar för PWM-utgångar. Uppdateringsfrekvensen sattes till 9155.27 Hz mellan samtliga moduler för att stämma överens med RC-filtret i den analoga utmatningskretsen.

3.3.4 Skärm

Kod som driver OLED-skärmar genom SPI utvecklades för att erbjuda visuell återkoppling. Embassys abstraktionslager och bibliotek för kommunikation med skärmens SH1106-drivchip abstraherades mycket av koden bort, och de inställningar som konfigurerades är relaterade till skärmens och mikrokontrollernas hårdvara. Till exempel förutsätter SPI-protokollet att kommunikation sker med en lämplig klockfrekvens, vilket begränsas av skärmens drivchip och mikrokontrollernas processor. Utvecklaren tillhandahåller ett gränssnitt som gör det möjligt att rita till skärmen.

3.4 VCO

VCO:ns hårdvarugränssnitt använder sig av två styrsignalsingångar med tillhörande potentiometrar, en ljudingång och två ljudutgångar. Den ena styrsignalen används för att styra VCO:ns frekvens och tolkas enligt enheten V/oct, d.v.s. att en förändring med en volt ändrar frekvensen med ett oktav (som innehåller 12 stycken linjärt fördelade halvtoner). Potentiometern som är kopplad till denna styrsignalsingång står därmed basfrekvensen som VCO:n genererar. Den andra styrsignalen bestämmer VCO:ns vågform och pulsbreddsmodulering. Ljudingången används för frekvensmodulering och möjliggör på så sätt modulering av oscillatorns frekvens med en signal i det hörbara frekvensintervallet. De två ljudutgångarna matar ut den resulterande vågen, där den ena utgången ger vågformen där styrsignalen styr vågform och den andra ger vågformen när styrsignalen styr pulsbreddsmodulering för en fyrkantsvåg.



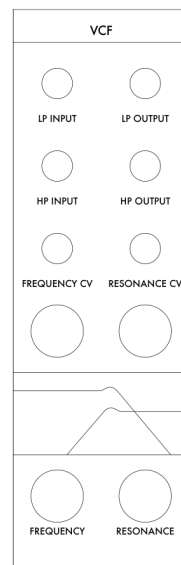
Figur 3.7: Illustration över VCO:ns frontpanel

Mjukvaran för VCO:n utvecklades med en s.k. *wavetable oscillator* som utgångspunkt. Varje vågform består av 64 stycken samples som genereras och lagras programmässigt vid uppstart och interpoleras linjärt under exekvering för att jämna ut signalen. Det är möjligt att i realtid blanda vågformerna parvis mellan sinus till triangel, triangel till såg och såg till fyrkant via styrsignal. Figur 3.7 visar modulens panel med markerade in- respektive utgångar.

3.5 VCF

Filtret har stöd för två simultana ingångar och utgångar, ett par för lågpasfiltrering, och ett par för högpasfiltrering. Den gemensamma gränshfrekvensen, samt resonansnivån kan ställas med potentiometrar, men även styras med kontrolls signaler. Två enkla volymkontroller är implementerade för att kunna bestämma nivån på de inkommande kontrolls signalerna.

Filtrets mjukvara använder ett bibliotek för att enklare skapa och hantera bipolära IIR-filer. För detta uppdateras filtrets koefficienter när ett av invärdena överstiger en viss gräns. Gränshfrekvensens insignal behöver sedan konverteras till en exponentiell skala från den linjära mätningen av ADC:n. Sedan exekveras filterberäkningen per sample som kommer in på ingången och skickas vidare till utgången. Figur 3.8 visar modulens panel med markerade in- respektive utgångar.

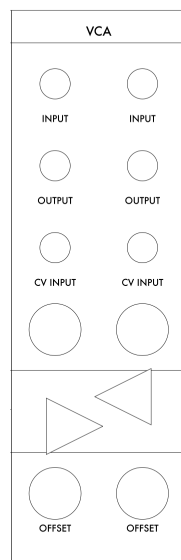


Figur 3.8: Illustration över VCF:ens frontpanel

3.6 VCA

VCA:n utvecklades till att styra en ljudsignals amplitud genom användandet av en kontrolls signal som förs in i mikrokontrollens ADC-ingång. Den ingående ljudsignalen förs in och ut via den externa kodekens ADC respektive DAC. VCA-modulen implementerar två förstärkare på samma mikrokontroll. Det möjliggörs av att det finns två uppsättningar av de tidigare nämnda in- och utgångar vilka kan användas parallellt. Figur 3.9 visar modulens panel med markerade in- respektive utgångar. En potentiometer installerades för att möjliggöra manuell justering av insignalens amplitud analogt utan att använda någon av kanalerna på mikrokontrollens ADC.

Mjukvaran använder de gemensamma drivrutinerna i stor utsträckning och är överlag egenimplementerad. Implementationen använder en linjär övergång från låg till hög volym, men det hade kunnat ändras i mjukvara till exempelvis exponentiella eller logaritmiska kurvor.



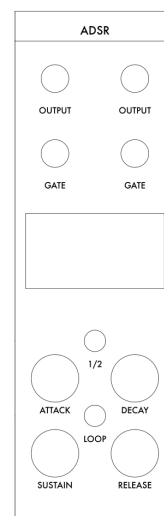
Figur 3.9: Illustration över VCA:ns frontpanel

3.7 ADSR

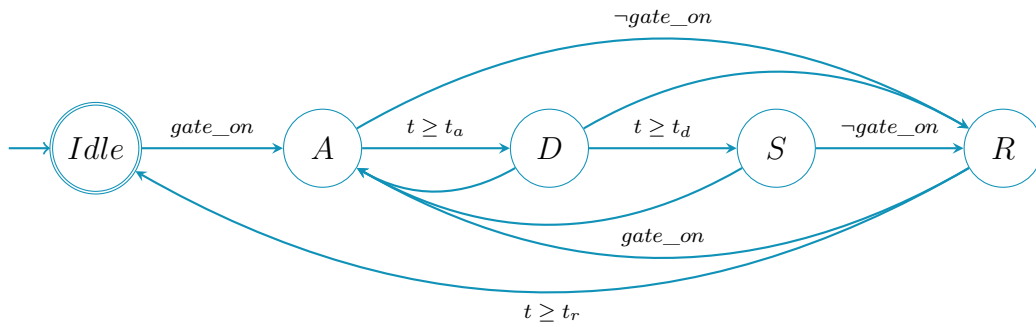
ADSR-modulen utvecklades med dess huvudsakliga insignaler, *trigger* och *gate*, i åtanke som i kombination med dess justerbara parametrar *attack*, *decay*, *sustain* och *release*, påverkar utsignalerna. Utsignalerna konfigurerades till att vara PWM-signaler där de variabla pulskvoterna motsvarar spänningsnivåer mellan $[0, 5]$ V. Figur 3.10 visar modulens panel med markerade in- respektive utgångar.

Målen för modulen implicerade att en utgående spänningsnivå varierar beroende på tid och konturens parametrar vilket kunde åstadkommas i mjukvara genom att implementera en ändlig automat där utspänningen uppdateras i varje tillstånd. Modulens main-loop körs iterativt i diskreta tidsintervall om $500\ \mu\text{s}$ där detta tidsintervall valdes i hänsyn till behovet av pålitliga, diskreta uppdateringar med marginal för långsammare exekvering efter att en utvärdering fastställde att main-loopens maximala exekveringstid är ungefär $450\ \mu\text{s}$. Eftersom övergången mellan tidsintervaller är delvis beroende av hur länge programmet befunnit sig i ett visst tillstånd, innehåller varje tillstånd information om hur länge det tillståndet varit aktivt. Spänningsnivåerna beräknas genom interpolering och är beroende av den nuvarande och maximala spänningen i tillståndet, det nuvarande tidsintervallet och slutligen den maximala tiden för tillståndet. Ett gränssnitt implementerades i mjukvaran för att i modulens källkod välja huruvida den resulterande konturen är exponentiell eller linjär.

Varje uppdatering innebär även en undersökning om huruvida villkoren för övergång till ett annat tillstånd uppfylls, om så är fallet, ställs spänningsnivåer in följaktligen och därefter uppdateras programmet till att, i nästa uppdatering, befinna sig i nästa tillstånd. Figur 3.11 visar en graf som redogör för den ändliga automaten och de villkorliga övergångarna.



Figur 3.10: Illustration över ADSR:ets frontpanel



Figur 3.11: Grafisk representation av den ändliga automaten som visar ADSR-modulens övergångar. Kanterna visar övergångarna mellan tillstånden och villkoren som måste uppfyllas för detta. t representerar tiden spenderat i ett givet tillstånd och t_a, t_d, t_s, t_r anger maxtiden för hur länge ett tillstånd varar.

Modulen stöder en dubbel uppsättning av konturgeneratorer i en modul där två konturer genereras samtidigt beroende på respektive konturgenerators gate-ingång. En vippomkopplare installerades på modulen för att styra vilken konturgenerators parametrar påverkas vid användande av modulens reglage. Ytterligare installerades en brytare som påverkar huruvida konturgenereringen för en vald konturgenerator är beroende av gate-ingången eller om konturen genereras periodvis oberoende av ingången. Det sistnämnda kan användas som en LFO och benämns som *loopande ADSR*. Ytterligare implementerades looping-funktion på modulen för att låta modulen producera konturen kontinuerligt oberoende av gate-signalen. Looping-funktionen kan aktiveras av användare genom en taktill omkopplare.

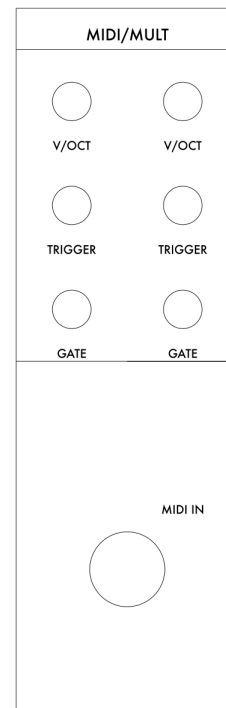
Modulens mjukvara utvecklades till att läsa värdena från potentiometrarna som representerar inställningarna för modulens justerbara parametrarna. Dessa påverkar hur den resulterande konturen för den utgående spänningen ser ut och hur länge varje tillstånd varar. Potentiometrarnas värden som läses från ADC:n användes för att spegla intervallet $[1, 10\,000]$ ms och kan ställas in med en upplösning av 1 ms för inställning av attack, decay och release. Parameterinställning av sustain konfigurerades till spegla ett intervall på $[0, 100]\%$ med en upplösning av 0.5 procentenheter.

Visualisering av den utgångssignalen är anmärkande för ADSR-modulen och implementeras ofta i kommersiella digitala syntar. En skärm lades till modulen för att visualisera den resulterande konturen på en tidsberoende grafik med hjälp av den ändliga automatens parametrar och dess representativa datastruktur. Drivrutinerna för skärmen vidareutvecklades till att visa den omedelbara positionen i konturen i en given tidpunkt.

3.8 MIDI-gränssnitt

För att upprätta ett gränssnitt mellan modulen och MIDI-kontroller krävdes en skyddande periferikrets som specificeras av MIDI-standarden [13]. Den skyddande kretsen realiserades på ett experimentkort för att ta indata från en MIDI-kontroller genom en DIN-5-port och därefter skicka den resulterande datan till ett UART-gränssnitt på mikrokontrollern. MIDI-standarden specificerar även en baudhastighet på 31.25 kbaud vilket MIDI-mottagaren konfigurerades till.

MIDI-standarden erbjuder även kommunikationsprotokoll som specificerar hur meddelandena ska se ut och tolkas [13] och därför implementerades en parser i mjukvara för att erbjuda stöd för alla MIDI-meddelanden. Med möjligheten att tolka MIDI-meddelanden realiserades den utgående signalen för *gate* som GPIO där spänningsnivån sätts högt när MIDI-meddelandet `note_on` mottas och spänningsnivån sätts lågt när `note_off` mottas. Därefter sätts pitch-signalen vid mottagandet av ett `note_on` meddelande genom en utgående PWM-signal där den resulterande spänningsnivån på intervallet $[-5, 5]$ V motsvarar en av 121 unika toner som kan genereras av en MIDI-kontroller, vilket motsvarar tonerna C_{-1} till C_9 . Läsningen av inkommande meddelanden konfigurerades att ske genom asynkron exekvering vilket möjliggör tolkning av MIDI-meddelandena i realtid. Figur ?? visar modulens panel med markerade in- respektive utgångar.



Figur 3.12: Illustration över MIDI-gränssnittets frontpanel

3.9 LFO

För LFO-funktionalitet, alltså generering av vågformer lägre än den undre gränsen för det hörbara frekvensområdet, utvecklades ingen egen modul. Dess funktioner kan istället åstadkommas med två andra moduler: VCO- och ADSR-modulerna. VCO:n klarar även av att generera frekvenser under den lägre gränsen för det hörbara frekvensområdet, vilket resulterar i att den kan användas som en LFO. Därtill kan ADSR-modulen användas som LFO i och med dess loop-funktionalitet, där den kan generera periodiska ADSR-signaler.

4

Resultat

Detta avsnitt redogör för de resultat som projektet producerat. I samband med detta evalueras de mot de mål som fastställdes för projektet och som beskrivs i kapitel 1.2. Kostnadseffektivitet särskiljs från de övriga målen eftersom dessa främst behandlar konstruktionen av synten.

4.1 Evalueringsmetod

Kostnadseffektiviteten av den resulterande modulära synten utvärderas i förhållande till priset för motsvarande moduler med öppen hårdvara. Med antagandet att modulerna som ingår i datauppsättningen har samma produktvärde som projektets moduler kan en lämplig jämförelse göras. Analysen utgår ifrån komponentkostnaderna för de delar i respektive modul som hanterar signalbehandlingen. Delarna som samtliga moduler har gemensamt, såsom reglage, kretskort och exempelvis potentiometrar, tas inte med i beräkningen i syftet att strikt undersöka kostnadseffektiviteten av signalbehandlingen i sig. Ytterligare kostnadsdata samlas in om färdiga syntmoduler som konsumenter kan köpa från musikåterförsäljaren Thomann.

Priskalkylen för både projektets moduler och de kommersiellt tillgängliga moduler med öppen hårdvara förenklas ytterligare genom att fastställa ett pris på 1 SEK/styck för passiva komponenter såsom resistorer, kondensatorer och dioder eftersom skillnaderna i pris mellan komponenternas variationer är försumbart. Övriga komponentpriser tillhandahålls av tre elektronikförsäljare som en hobbyist med liknande förutsättningar skulle rimligtvis kunna beställa från och inkluderar även moms. I allra första hand tillhandahålls priserna från försäljaren Electrokit, och beroende på tillgänglighet, även RS-Online och Mauser.

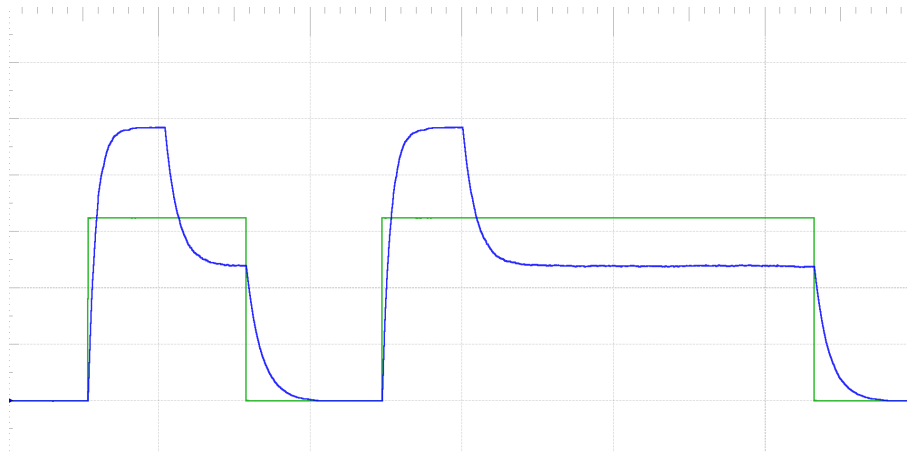
Evalueringen av resterande mål såsom programmerbar funktionalitet 1.2.2, kompatibilitet 1.2.3 och slutligen ljudkvalitet 1.2.4, beror huvudsakligen på resultatet av konstruktionen. Dessa mål evalueras på olika sätt där resultatet för programmerbar funktionalitet påverkas av huruvida det finns ett gränssnitt för att ändra på modulernas källkod, kompatibilitet mäts genom att undersöka huruvida modulerna följer Eurorack-standarderna, och ljudkvalitet evalueras genom att med olika medel undersöka huruvida modulerna ger upphov till hörbara störningar.

De enskilda modulernas specifika implementation är inte väldefinierad och varierar mellan implementationer, det finns därför inget fast beteende att verifiera korrekthet mot. Ljud- och styrsignaler är däremot väldefinierade och därför kan korrekthet, som är beroende av den hantering som implementeras gemensamt för modulerna, undersökas. Av dessa anledningar kan evalueringen av ljudkvalitet avgränsas till att undersöka huruvida, fördröjning uppstår vid signalhantering, signaler mottas och sänds med en samplingsfrekvens på 48 kHz och att fördröjningar inte introduceras mellan moduler.

4.2 Konstruktion

Mjukvarans anpassningsbarhet har uppnåtts till den grad där varje modul bygger på en gemensam plattform där diskreta periferikrestar kan kopplas in efter funktion. Detta innebär att det finns stöd för implementering av vidare funktionalitet, förutsatt att mikrokontrollerns och periferikretsarnas resurser räcker till. Dessutom innehåller mikrokontrollern mikro-USB-gränssnitt som tillhandahåller ett lätt sätt att omprogrammera mikrokontrollern.

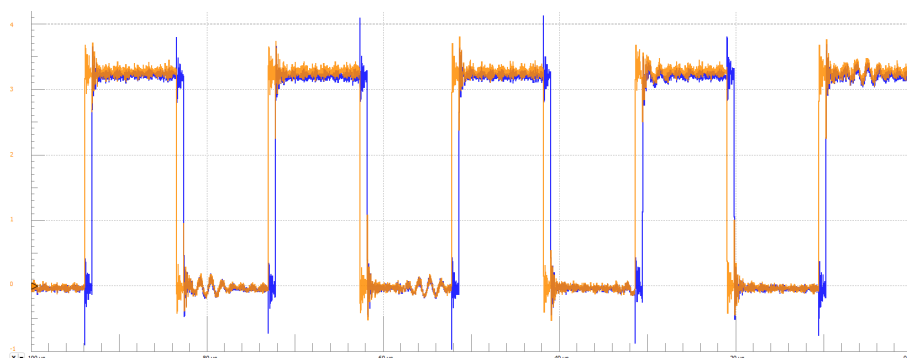
Gränssnittet mellan syntmodulerna designades med kompatibilitet i åtanke från början. De uppfyller de Eurorack-specifikationer som krävs för att kunna användas med kommersiellt tillgängliga syntar. Figur 4.1 visar att ADSR-signalen genereras som förväntat och mätningarna som gjordes visade att spänningen inte överstiger 5 V vilket är i enlighet med standarden för styrsignaler. Figuren visar även hur konturen genereras i realtid i förhållande till gate-signalen. Sammanfattningsvis uppnår den färdiga konstruktionen kompatibilitetsmålet.



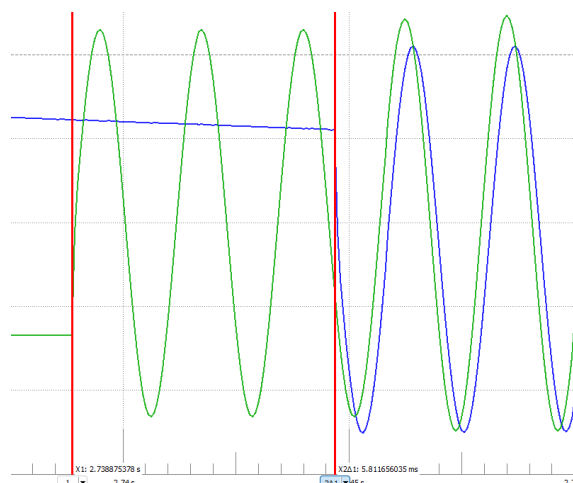
Figur 4.1: Mätning av ADSR-utgången representerat i blått, i förhållande till gate-signalen i grönt.

Den utgående ljudsignalens upplösning uppmättes till att ha en samplingsfrekvens på 48 kHz, denna bedömning gjordes efter en mätning av kodekens *word select*-klocka visade en frekvens på 48 kHz. Figur 4.2 visar denna mätning. Detta innebär att varje ord i I²S-meddelandena behandlas med den samplingsfrekvens som stipulerades i målen. Ytterligare gjordes mätningar för att testa huruvida hörbara artefakter

produceras vid ljudbehandling. Figur 4.3 visar mätningar som gjordes på VCA:ns utgång i förhållande till dess ingång. Mätningarna visade en fördröjning av ungefär 5.8ms. Val av kodek och elektroniska komponenter med adekvata egenskaper förebyggde auditiva fel vid konvertering och spridning av ljudsignalerna, såväl som i de styrsignaler som används för att avgöra hur ljudsignalerna ska behandlas.



Figur 4.2: Mätning av klocksignalerna för kodekens DAC respektive ADC. Signalen är periodisk och har en frekvensen 48 kHz



Figur 4.3: Figuren visar hur en signal börjar att passera VCA-modulen med en fördröjning på ungefär 5.8ms. Den gröna signalen motsvara in-signalen och den blå representerar ut-signalen från VCA:n.

4.3 Kostnadseffektivitet

Tabellerna B.1 för VCO, B.2 för VCF, B.3 för VCA, samt B.4 för ADSR, i Appendix B visar sammanställningen av komponentkostnaderna för modulerna med öppen hårdvara. Dessa jämförs med kostnaden för projektets motsvarande moduler som redovisas i B.5 i Appendix B. Ytterligare visar Tabell 4.1 vad färdigmonterade jämförbara moduler kostar att köpa, inklusive moms, från återförsäljaren Thomann. Figur 4.4 visar hur projektets moduler prismässigt förhåller sig till komponentkostnaderna för syntmoduler med öppen hårdvara samt den billigaste modulen från Thomann i respektive kategori.

Som den grafiska representationen i Figur 4.4 visar, befinner sig VCO:n under den översta kvartilen i jämförelse med andra moduler med öppen hårdvara. Dessutom är skillnaden mellan VCO:n och den kommersiellt tillgängliga modulen signifikant, där projektets VCO är 224 SEK billigare. VCF:n och ADSR-modulen är dyrare än samtliga moduler med öppen hårdvara men billigare än alternativen från Thomann. ADSR:n är synnerligen dyrare än modulerna med öppen hårdvara där skillnaden mellan den dyraste modulen från urvalet är 186 SEK. VCA:n är dyrare än de färdigmonterade alternativen och modulerna med öppen hårdvara. Priset för LFO:n är detsamma som för VCO:n eftersom en egen modul för detta inte designades i hänsyn till att VCO:n uppfyller den funktionaliteten. Figuren visar inte heller hur LFO:n och MIDI-modulen förhåller sig till moduler med öppen hårdvara, detta är i hänsyn till begränsningen av urval för liknande moduler med öppen hårdvara. LFO:n och MIDI-modulen jämförs därför endast med deras billigaste motpart från Thomann, där projektets LFO är 199 SEK billigare, och MIDI-modulen är 347.5 SEK billigare än deras motsvarighet från Thomann.

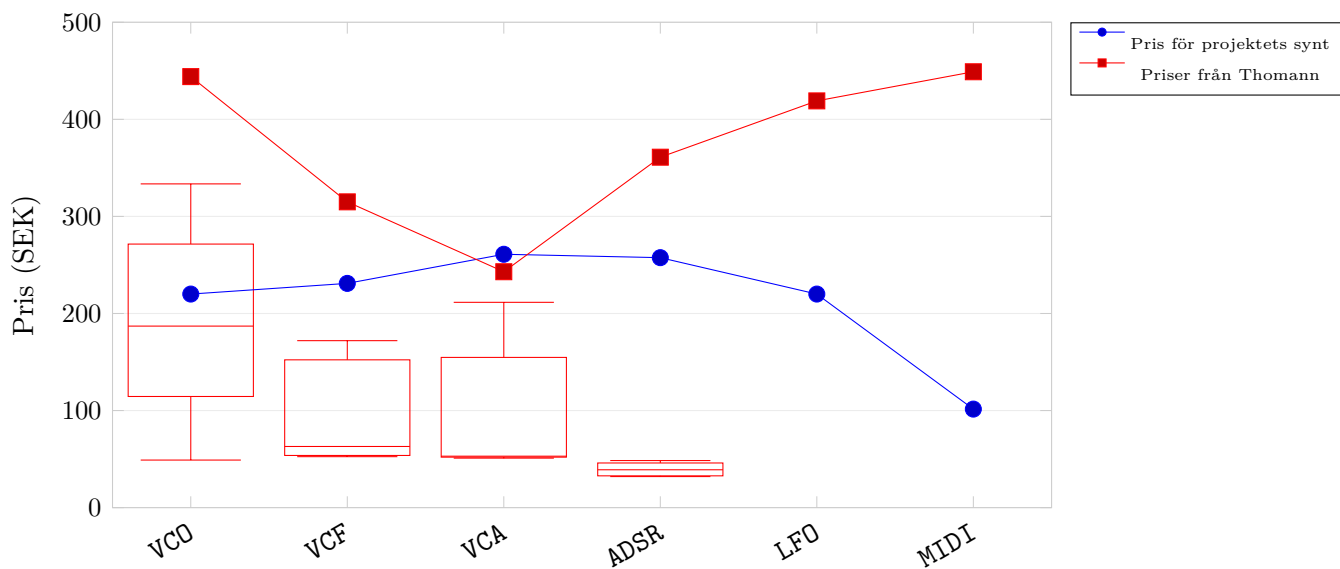
Ett gemensamt kretskort designades med kostnadseffektivitet i åtanke vilket kan användas till alla moduler. Tanken var att på så vis både dra ner på den hypotetiska utvecklingskostnaden och framtida tillverkningskostnad. Alla modulers hårdvarugränssnitt är dock inte gemensamt, vilket innebär att man vid montering kan välja till exempel hur många in- och utgång för styr- och ljudsignaler som ska monteras. På så vis dras tillverkningskostnaden ner, i och med att man bara behöver montera komponenterna som just den aktuella syntmodulen kräver.

En annan kostnadsfördel med att ha ett gemensamt kretskort är att eventuella framtida certifieringar som hade krävts inför kommersiell försäljning (t.ex CE och RoHS) endast skulle behöva göras för ett kretskort, istället för att behöva göras för en handfull kretskort om de skulle skilja sig åt mellan modulerna.

Mikrokontrollern valdes med kostnadseffektivitet i åtanke och stödjer endast tre ADC:er där en av dessa används av multiplexern som är nödvändig för att stödja fler potentiometrar. Övriga ADC:er användes i samband med ingångar av styrsignaler vilket i sin tur kräver kringkomponenter för omvandlingen, vilket utgör en signifikant del av kostnaden. Vissa moduler använder samtliga tillgängliga ingångar och därför skulle en eventuell utökning med befintlig plattform inte vara möjlig.

För att underlätta design av kretskort, både i form av reducerad tid och reducerad designkomplexitet, så integrerades hela Raspberry Pi Pico 2-kortet i kretskortet. Pico 2 bygger på mikrokontrollerchippet RP2350, som också finns kommersiellt tillgängligt i sig, och kommer med alla kringkomponenter som krävs för att använda hela MCU:ns funktionalitet. I Tabell B.5 blir det tydligt att en betydande del av kostnaden är just Pico 2:an.

Kodeken utgör också en signifikant del av kostnaden vilket tydliggörs i Tabell B.5. Utifrån projektets mål valdes en kodek som hade stöd för 48kHz och 24 bitars upplösning. Utöver att komponenten i sig står för en del av kostnaden så kräver den även kringkomponenter, som i sin tur också bidrar till ökad kostnad.



Figur 4.4: Grafisk representation av prisjämförelser där priserna för modulerna med öppen hårdvara representeras som låddiagram. Färdiga produkter på marknaden och projektets moduler representeras som punkter.

Modul	Produktnamn	Pris (SEK)
VCO	Behringer 2600-VCO	444
VCF	Behringer 904A VC Low Pass Filter	315
VCA	Behringer 902 VCA	243
ADSR	Behringer Dual Envelope Generator 1003	361
LFO	Behringer 150 Ring Mod/Noise/S&H/LF	419
MIDI	Behringer CM1A	449

Tabell 4.1: Kostnaden, inklusive moms, för den billigaste jämförbara modulen från Thomann under tiden för projektet.

5

Diskussion

Resultaten som presenteras i 4.3 avseende kostnadseffektivitet visar på att det är möjligt att konstruera modulära syntar för en lägre kostnad än det som konstruerats i projektet. En nyanserad diskussion om huruvida den färdiga konstruktionen är kostnadseffektiv bör däremot även innehålla en jämförelse av funktionalitet och möjligheten för vidareutveckling. De mest minimala analoga motsvarigheterna kan realiseras med jämförelsevis få komponenter och låga kostnader men det reflekterar inte nödvändigtvis vad den kan göra. Detta exemplifieras bäst av den stora skillnaden i pris mellan ADSR-modulerna där den skillnaden kan reduceras genom att exempelvis ta bort skärmen eftersom det inte är strikt nödvändigt. Faktumet att projektets synt tillhandahåller en smidig mjukvaruplattform att lägga till ytterligare funktionalitet i en eventuell vidareutveckling utan att behöva göra större ändringar i övriga modulen introducerar en faktor som skulle göra den kostnadseffektiv i viss bemärkelse.

I förhållande till huruvida det är mer kostnadseffektivt att bygga projektets synt än att köpa en färdig beror på ett antal tolkningar. De färdigtillverkade syntarna är begränsade avseende vidareutveckling, vilket inte stämmer för projektets synt vars programvara kan vidareutvecklas. Ytterligare behandlar priskalkylen för projektets synt endast kostnaden för signallogiken vilket innebär att priskalkylen inte reflekterar den slutgiltiga kostnaden för den färdiga produkten, inklusive montering, och troligtvis blir den slutliga kostnaden för en likvärdig produkt högre. Likt jämförelsen av att konstruera syntar med öppen hårdvara, blir analysen begränsad av att de kommersiellt tillgängliga modulerna inte nödvändigtvis stödjer funktionalitet som projektets motsvarighet gör, och vice versa.

Båda analyser av kostnadseffektivitet bör innehålla beroende på hur relevant vidareutveckling är och även en uppskattning av eventuella kostnader för vidareutvecklingen av etablerade modul, förutsatt att det är möjligt.

Det tillgängliga urvalet och evalueringsmetoden brister i bemärkelsen att vissa moduler innehåller specialiserade och okonventionella komponenter som kan vara svårtillgängliga för konsumenter både i avseende till pris och till tillgänglighet. Dessa komponenter försvårar jämförelser eftersom deras priser tenderar att vara volatila. Detta går inte nödvändigtvis emot syftet med studien eftersom otillgänglighet av komponenter nämns som en del av syftet i avsnitt 1.1.

Prisuppskattningarna i rapporten är baserat på styckpriser av elektroniska komponenter från endast några få återförsäljare av elektroniska komponenter som även tar emot småskaliga beställningar. I verkligheten påverkas dessa priser kraftigt av inköpsvolym och vilken återförsäljare som används vilket innebär att prisuppskattningarna sannolikt är överskattade. Eftersom denna överskattning gäller för både projektets synt och modulerna med öppen hårdvara, förblir den jämförelsen relativt giltig. Av dessa anledningar är det rimligt att anta att en kostnadsanalys av projektets synt som även beaktar kostnadseffektivitet i förhållande till massproduktion skulle vara avsevärt lägre.

Resultaten visar att konstruktionen är förenlig med målen avseende programmerbar funktionalitet, kompatibilitet och ljudkvalitet. Systemet är responsivt och producerar högupplösta signaler utan några hörbara artefakter. En fördröjning på ungefär 5.8 ms uppmättes för ljudsignaler vilket kan introducera hörbara fördröjningar om man har längre signalkedjor som kräver ljudbearbetning men som i sig inte är hörbart. Spänningarna som mättes upp ligger inom de specificerade intervallen för att säkerställa kompatibilitet.

5.1 Förbättringsområden

Under projektets gång, framförallt under konstruktionsdelen och resultatsammanställning, identifierades ett par förslag på förbättringar. Generellt skulle de flesta innebära att en avvägning mellan funktion och kostnad skulle behöva göras.

En förbättring där så inte är fallet är hur integrationen av mikrokontrollern gjorts. Som tidigare nämnts i Avsnitt 4.3 beslutades det av designeffektivitetsskäl att i kretskortsdesignen ha med hela Raspberry Pi Pico 2-kortet. En framtida revision av kretskortet hade gjort det möjligt att istället använda enbart chippet som Pico 2:an också använder: RP2350. Detta arbete skulle innefatta design av kringkretsar för att kunna använda RP2350, men komponentpriserna hade i slutändan kunnat bli billigare än Pico 2-kortet.

Vid eventuell vidareutveckling av själva syntmodulerna skulle det kunna hända att kretskortets maximala kapacitet vad gäller ingångar nås. Som tidigare nämnt är antalet styrsignalsingångar direkt begränsat av antalet ADC:er som mikrokontrollern har. Om en annan mikrokontroller än Pico 2 skulle användas så hade det varit möjligt att mångdubbla antalet styrsignalsingångar. Detta hade dock i sin tur lett till fler komponenter och därmed högre kostnad.

Antalet in- och utgångar för ljudsignaler begränsas istället av kodeken, som i sig är en betydande del av kostnaden. Det hade varit fördelaktigt att undersöka vidareutveckling av in- och utgångar för ljudsignaler ur ett kostnadseffektivitetsperspektiv. Detta projekt undersökte inte hur mycket upplösningen av ljudsignalerna spelar roll i sammanhanget av en modulär synt. Om en sådan undersökning hade gjorts och också fastställt att en lägre upplösning på ljudsignaler hade varit godkänd så hade man kunnat designa kretskortet runt billigare komponenter än den valda kodeken.

Till exempel hade mikrokontrollernas inbyggda ADC:er då även kunnat användas för insignaler och billigare DAC:ar hade kunnat användas för utsignaler. Detta hade också kunna gett större flexibilitet kring antalet in- och utgångar, vilket i sin tur också hade varit positivt för kostnadseffektivitetens skull.

5.2 Slutsats

Projektet har visat på en lyckad konstruktion av en modulär synt som är kompatibel med andra analoga moduler i enlighet med kompatibilitetsmålet som specificeras i avsnitt 1.2.3. Syntens ljudkvalitet tillfredsställer målsättningen från avsnitt 1.2.4. Den färdiga produkten är implementerad med hjälp av digital signalbehandling vilket har lett till att den är programmerbar och tillfredsställer därför målen enligt avsnitt 1.2.2.

Beträffande kostnadseffektivitet har projektet visat att det finns möjlighet att skapa moduler till en lägre kostnad, men att denna jämförelse inte behandlar produktvärde. Det finns även utrymme för framtida optimeringar avseende kostnadseffektivitet. Modulernas funktionalitet och egenskaper är aspekter som påverkar deras slutgiltiga marknadsvärde och konkurrenskraft vilket skulle göra framtida undersökningar mer nyanserade.

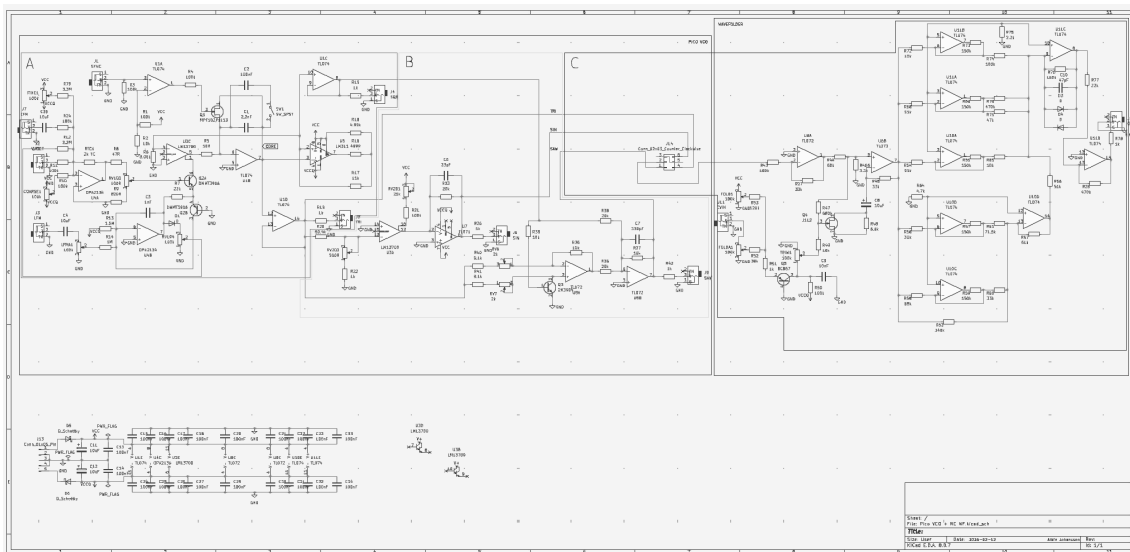
Källförteckning

- [1] T. Holmes, *Electronic and Experimental Music: Technology, Music, and Culture*, 3. utg. Routledge, 2008, ISBN: 0203929594.
- [2] G. Joly, "Moog 921", Fotografi, Licenserat enligt CC BY-SA 2.0. Inga ändringar har gjorts. hämtad 27 april 2026. URL: <https://www.flickr.com/photos/24987280@N00/4359425319/>
- [3] K. Bjørn och C. Meyer, *Patch and Tweak - Exploring modular synthesis*, 3. utg. Bjooks, 2018, ISBN: 9788799999514.
- [4] A. Nelson, A. Callen och M. Tripsas, "If I Could Turn Back Time: Occupational Dynamics, Technology Trajectories, and the Reemergence of the Analog Music Synthesizer.," *Administrative Science Quarterly*, årg. 68, nr 2, 2023. DOI: 10.1177/00018392231163178
- [5] *NT73 Temperature Compensation Thick Film Resistors*, Koa Speer Electronics Inc., 5 april 2023. hämtad 12 febr. 2026. URL: <https://www.koaspeer.com/products/thermal-sensors-inductors/negative-tempco-thermistors/nt73/>
- [6] Doepfer, utg. "Technical Details A-100," hämtad 27 april 2026. URL: https://doepfer.de/a100_man/a100t_e.htm
- [7] O. Gillet, *clouds_v30*, 28 aug. 2017. hämtad 27 april 2026. URL: https://bgr360.github.io/assets/pdf/mutable_instruments/clouds_v30.pdf
- [8] Doepfer, utg. "Construction Details A-100," hämtad 27 april 2026. URL: https://www.doepfer.de/a100_man/a100m_e.htm
- [9] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, årg. 27, nr 3, s. 379–423, 1948.
- [10] *AES Recommended Practice for Professional Digital Audio — Preferred Sampling Frequencies for Applications Employing Pulse-Code Modulation*, AES5-2018, Revision of AES5-2008, Audio Engineering Society, 2018. URL: <https://www.elecenghub.com/NewSamples/AES/159886771/AES-5-2018-1.pdf>
- [11] Omegatron, "Op-amp pinouts", Illustration, Licenserat enligt CC BY-SA 3.0. Denna vectorbild skapades med gnuplot. Inga ändringar har gjorts. hämtad 18 maj 2026. URL: <https://commons.wikimedia.org/w/index.php?curid=983276>
- [12] *RP2350 Datasheet, A microcontroller by Raspberry Pi*, Raspberry Pi Ltd, 29 juli 2025. hämtad 27 april 2026. URL: <https://pip-assets.raspberrypi.com/categories/1214-rp2350/documents/RP-008373-DS-2-rp2350-datasheet.pdf>

- [13] *MIDI 1.0 Detailed Specification*, The MIDI Manufacturers Association, febr. 1996. hämtad 27 april 2026. URL: <https://drive.google.com/file/d/1ewRrvMEFRPlKon6nfSCxqnTMEu70sz0c/view>
- [14] C. M. Burnett, "A single master and a single slave on a Serial Peripheral Interface (SPI) bus", Illustration, Licenserat enligt CC BY-SA 3.0. Denna W3C-ospecificerade vectorbilden skapades med Inkscape. Inga ändringar har gjorts. hämtad 13 maj 2026. URL: <https://commons.wikimedia.org/w/index.php?curid=1476502>
- [15] *8-Bit Microprocessor & Peripheral Data*, Motorola Inc., 1983.
- [16] Corelis, utg. "SPI Tutorial," hämtad 12 maj 2026. URL: <https://www.corelis.com/education/tutorials/spi-tutorial/>
- [17] *UM11732, IIS bus specification*, NXP B.V., 17 febr. 2022. hämtad 28 april 2026. URL: <https://www.nxp.com/docs/en/user-manual/UM11732.pdf>
- [18] Omegatron, "Waveforms", Illustration, Licenserat enligt CC BY-SA 3.0. Denna vectorbild skapades med gnuplot. Inga ändringar har gjorts. hämtad 16 maj 2026. URL: <https://commons.wikimedia.org/wiki/File:Waveforms.svg>
- [19] *24-BIT, 96/192-kHz ASYNCHRONOUS STEREO AUDIO CODEC*, Texas Instruments, mars 2008. hämtad 27 april 2026. URL: <https://www.ti.com/lit/ds/symlink/pcm3060.pdf>
- [20] *VCO_MANUAL*, Erica Synths. hämtad 29 april 2026. URL: https://www.ericasyths.lv/service/file/download/product_id/825/file_id/431/
- [21] *Braids v50*, Oliver Gillet, 28 aug. 2017. hämtad 29 april 2026. URL: https://bgr360.github.io/assets/pdf/mutable_instruments/braids_v50.pdf
- [22] *Ultrafade Schematics*, Stroh. hämtad 29 april 2026. URL: <http://www.loudestwarning.co.uk/wp-content/uploads/2015/02/UltraFadeVCO.pdf>
- [23] *Dannysound EN129 Thru Zero Oscillator*, Dannysound, 7 okt. 2018. hämtad 29 april 2026. URL: https://www.thonk.co.uk/wp-content/uploads/Documents/dannysound/EN129_ThruZero/v2%20September%202020%20onwards/v2%20schematics/
- [24] E. Bergman, *Synthesizer Build part-40: WAVETABLE OSCILLATOR (VCO) By Electric Druid*. 12 mars 2021. hämtad 29 april 2026. URL: <https://www.eddybergman.com/2021/03/synthesizer-build-part-40-wavetable.html>
- [25] *VCF_MANUAL*, Erica Synths, 10 juni 2022. hämtad 16 maj 2026. URL: https://www.ericasyths.lv/service/file/download/product_id/823/file_id/450/
- [26] O. Gillet, *Ripples v40*, 28 aug. 2017. hämtad 16 maj 2026. URL: https://bgr360.github.io/assets/pdf/mutable_instruments/ripples_v40.pdf
- [27] *KS-20 VCF*, Kassutronics, 26 april 2020. hämtad 16 maj 2026. URL: https://github.com/kassu/kassutronics/blob/master/documentation/KS-20%20VCF/Kassutronics_KS-20_Build_Docs_1.1C.pdf
- [28] *MS20v2.5*, AI Synthesis. hämtad 16 maj 2026. URL: <https://aisynthesis.com/wp-content/uploads/2020/05/MS20v2.5.pdf>
- [29] filip, *Filther Build Guide*, 14 juli 2021. hämtad 16 maj 2026. URL: <https://www.cctv.fm/post/filther-build-guide>

- [30] *VCA_MANUAL*, Erica Synths, 27 jan. 2022. hämtad 16 maj 2026. URL: https://www.ericasynts.lv/service/file/download/product_id/821/file_id/433/
- [31] O. Gillet, *veils_v40*, 28 aug. 2017. hämtad 16 maj 2026. URL: https://bgr360.github.io/assets/pdf/mutable_instruments/veils_v40.pdf
- [32] *Dual VCA*, Kassutronics, 12 jan. 2019. hämtad 16 maj 2026. URL: https://github.com/kassu/kassutronics/blob/master/documentation/Dual%20VCA/DualVCADocs_1.1.pdf
- [33] *DualVCA*, Stroh, 27 nov. 2015. hämtad 16 maj 2026. URL: <https://www.modwiggler.com/forum/download/file.php?id=19718&sid=612008ad01fbe66dc74781391b>
- [34] *1181 DUAL VCA*, LOOK MUM NO COMPUTER. hämtad 16 maj 2026. URL: <https://www.lookmumnocomputer.com/1181-dual-vca>
- [35] *EG_MANUAL*, Erica Synths, 7 mars 2022. hämtad 16 maj 2026. URL: https://www.ericasynts.lv/service/file/download/product_id/794/file_id/439/
- [36] *adsr*, AI Synthesis, 23 sept. 2016. hämtad 16 maj 2026. URL: <https://aisynthesis.com/wp-content/uploads/2026/02/adsr.pdf>
- [37] *ASR Envelope*, Kassutronics, 10 dec. 2020. hämtad 16 maj 2026. URL: https://github.com/kassu/kassutronics/blob/master/documentation/ASR%20Envelope/Kassutronics_ASR_docs_1.1B.pdf
- [38] E. Bergman, *Synthesizer Build part-63: FASTEST ADSR IN THE WEST by Rene Schmitz*. 18 dec. 2024. hämtad 16 maj 2026. URL: <https://www.eddybergman.com/2024/12/ADSR%5C%20Rene%20Schmitz.html>
- [39] *Building a Eurorack Envelope Generator*, N8Synth. hämtad 16 maj 2026. URL: <https://www.n8synth.co.uk/diy-eurorack/envelope-generator/>

A



Figur A.1: Figuren illustrerar delkretsarna som utgör en fullständig VCO. Krets A utgör konstruktionen av en grundläggande oscillator med triangel- och fyrkantsvåg. Krets B utgör påbyggnaden som möjliggör såg- och sinusvågor. Krets C avser implementationen av wavefolder-funktionalitet.

B

Kostnadstabeller

Erica VCO [20]	Analog	MI Braids [21]	Digital	Stroh Ultrafade [22]	Analog	Tobinski EN129 [23]	Analog	ED VCDO [24]	Digital
Resistorer	25	Resistorer	36	Resistorer	77	Resistorer	71	Resistorer	52
Kondensatorer	10	Kondensatorer	31	Kondensatorer	33	Kondensatorer	25	Kondensatorer	28
NPN/PNP	2	NPN/PNP	5	NPN/PNP	3	NPN/PNP	6	NPN/PNP	1
Dioder	2	Dioder	2	Dioder	3	Dioder	7	Dioder	10
TL074	5	STM32	80	LM13700 x 2	20	J113	3.5	TL072 x 6	9
CD40106	5	MCP6004	9	TL074 x 2	10	78L09	2.5	VCDO	70
		TL074	5	LT1013 x 2	40	79L09	2.5	7805	4
		TL072	1.5	TL071 x 3	9	TL072 x 6	9	7905	6
		74HC595 x 2	9	TL072 x 2	3	CD4013	8		
		MCP3204	15	LM311	3	CA3140	32		
		DAC8551	55	J113	3.5	CD4070	5.5		
		LM1117	10	DMMT3906	5	TL074	5		
		LM4040B25	10			CA3080 x 2	Otillgänglig		
		LM4040B10	10						
		Display	50	Kräver Tempco-motstånd		Kräver Tempco-motstånd			
		Kristallosc.	5			LM13700 som alt. för CA3080 10 SEK			
Tot	49	Tot	333.5	Tot	209.5	Tot	187	Tot	180

Tabell B.1: Komponentkostnader VCO:er med öppen hårdvara (SEK)

Erica VCF [25]	MI Ripples [26]	Kassutronics F. [27]	AI MS20 [28]	CCTV Filther [29]
Resistorer 27	Resistorer 44	Resistorer 24	Resistorer 26	Resistorer 31
Kondensatorer 15	Kondensatorer 28	Kondensatorer 10	Kondensatorer 10	Kondensatorer 10
Dioder 11	NPN/PNP 2	NPN/PNP 2	NPN/PNP 2	Dioder 5
TL074 x 2 10	Dioder 3	Dioder 2	Dioder 2	CEM3320 ~ 80
	TL074 x 2 10	LM13700 10	LM13700 10	TL074 5
	TL072 x 2 3	TL072 x 2 3	TL074 5	TL072 1.5
	LM13700 10	NE5532 1.5		
	SSI2164 62			CEM3320 är ett ovanligt chip
	LM4040 10			Pris taget från Banzaimusic
				med AS3320, som är ersättning
Tot 63	Tot 172	Tot 52.5	Tot 55	Tot 132.5

Tabell B.2: Komponentkostnader VCF:er med öppen hårdvara (SEK)

Erica VCA [30]		MI Veils / 2 [31]		Kassutr. Dual [32]		Stroh Dual [33]		LMNC Dual [34]	
Resistorer	34	Resistorer	34	Resistorer	31	Resistorer	20	Resistorer	30
Kondensatorer	8	Kondensatorer	22	Kondensatorer	17	Kondensatorer	16	Kondensatorer	7
NPN/PNP	4	Dioder	4	Dioder	4	NPN/PNP	2	NPN/PNP	2
Dioder	2	LM358	1.5	AS3360	~ 40	Dioder	2	Dioder	4
TL072 x 2	3	SSI2164	62	TL072 x 4	6	LM13700 x 2	10	TL074 x 2	10
		OPA1654 x 2	68	<i>AS3360 är ett ovanligt chip</i>		TL072 x 2	3		
		LM4040 x 2	20	<i>Pris taget från Erica Synths</i>					
Tot	51	Tot	211.5	Tot	98	Tot	53	Tot	53

Tabell B.3: Komponentkostnader VCA:er med öppen hårdvara (SEK)

B. Kostnadstabeller

Erica ASR [35]		AI Loop ADSR [36]		Kassutr. ASR [37]		Rene S. ADSR [38]		N8Synth ADSR [39]	
Resistorer	18	Resistorer	22	Resistorer	15	Resistorer	16	Resistorer	9
Kondensatorer	13	Kondensatorer	9	Kondensatorer	7	Kondensatorer	7	Kondensatorer	10
NPN/PNP	1	NPN/PNP	1	NPN/PNP	1	NPN/PNP	3	NPN/PNP	3
Dioder	7	Dioder	10	Dioder	6	Dioder	3	Dioder	5
TL072 x 3	4.5	TL074	5	TL072 x 2	3	ICM7555	5	ICM7555	5
		TL072	1.5			TL074	5	TL072	1.5
Tot	43.5	Tot	48.5	Tot	32	Tot	39	Tot	33.5

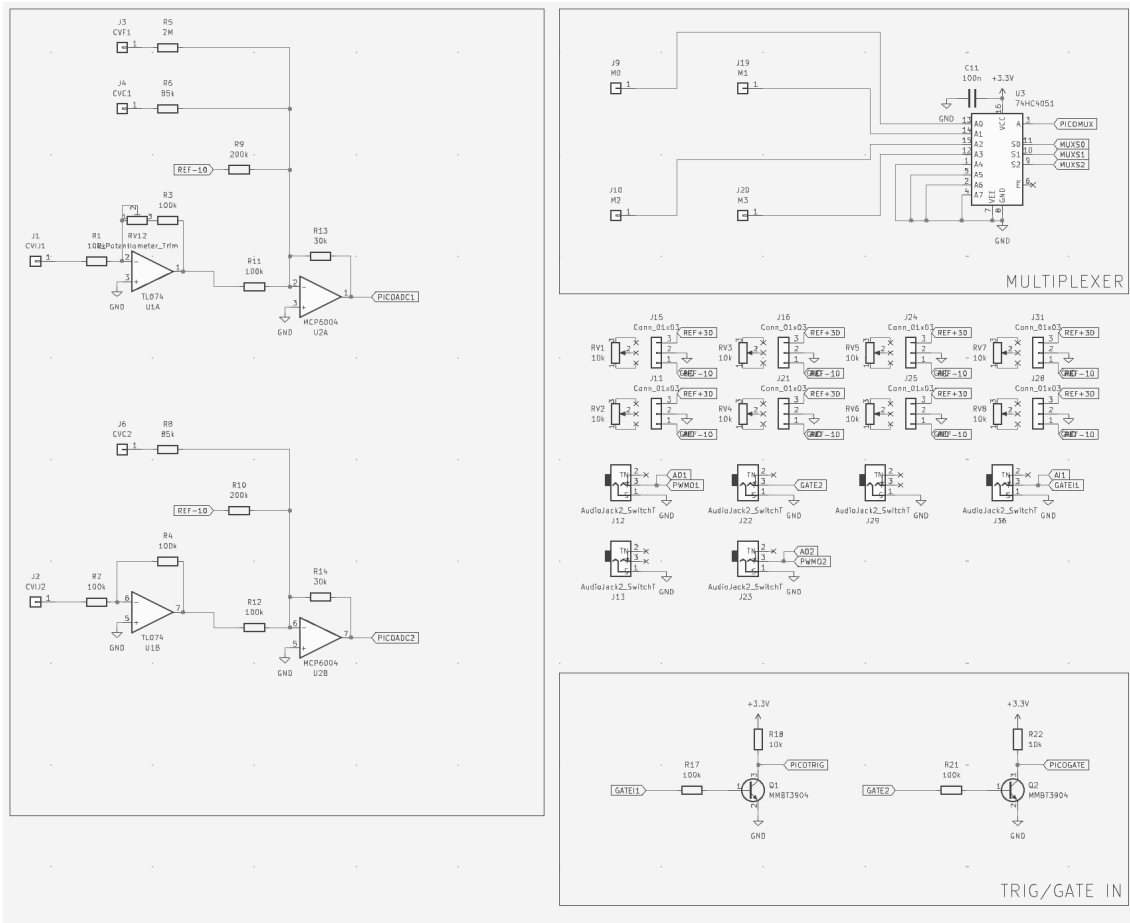
Tabell B.4: Komponentkostnader ADSR:er med öppen hårdvara (SEK)

VCO		VCF		VCA		ADSR		MIDI	
Resistorer	30	Resistorer	41	Resistorer	31	Resistorer	18	Resistorer	5
Kondensatorer	27	Kondensatorer	27	Kondensatorer	23	Kondensatorer	23	Kondensatorer	12
Dioder	3	Dioder	3	Dioder	2	NPN/PNP	2	Dioder	3
TL074 x 2	10	TL074 x 2	10	TL074 x 2	10	Dioder	2	TL074	5
MCP6004	9	MCP6004	9	MCP6004	9	TL074	5	6N138	8.5
RP Pico	68	RP Pico	68	RP Pico	68	RP Pico	68	RP Pico	68
PCM3060	53	PCM3060	53	PCM3060	53	LM4040 x 2	20		
74HC4051	4.5	LM4040 x 2	20	LM4040 x 2	20	74HC4051	4.5		
						1.3 OLED-skärm	115		
Tot	204.5	Tot	231	Tot	216	Tot	257.5	Tot	101.5

Tabell B.5: Komponentkostnader för modulerna som skapats i projektet (SEK)

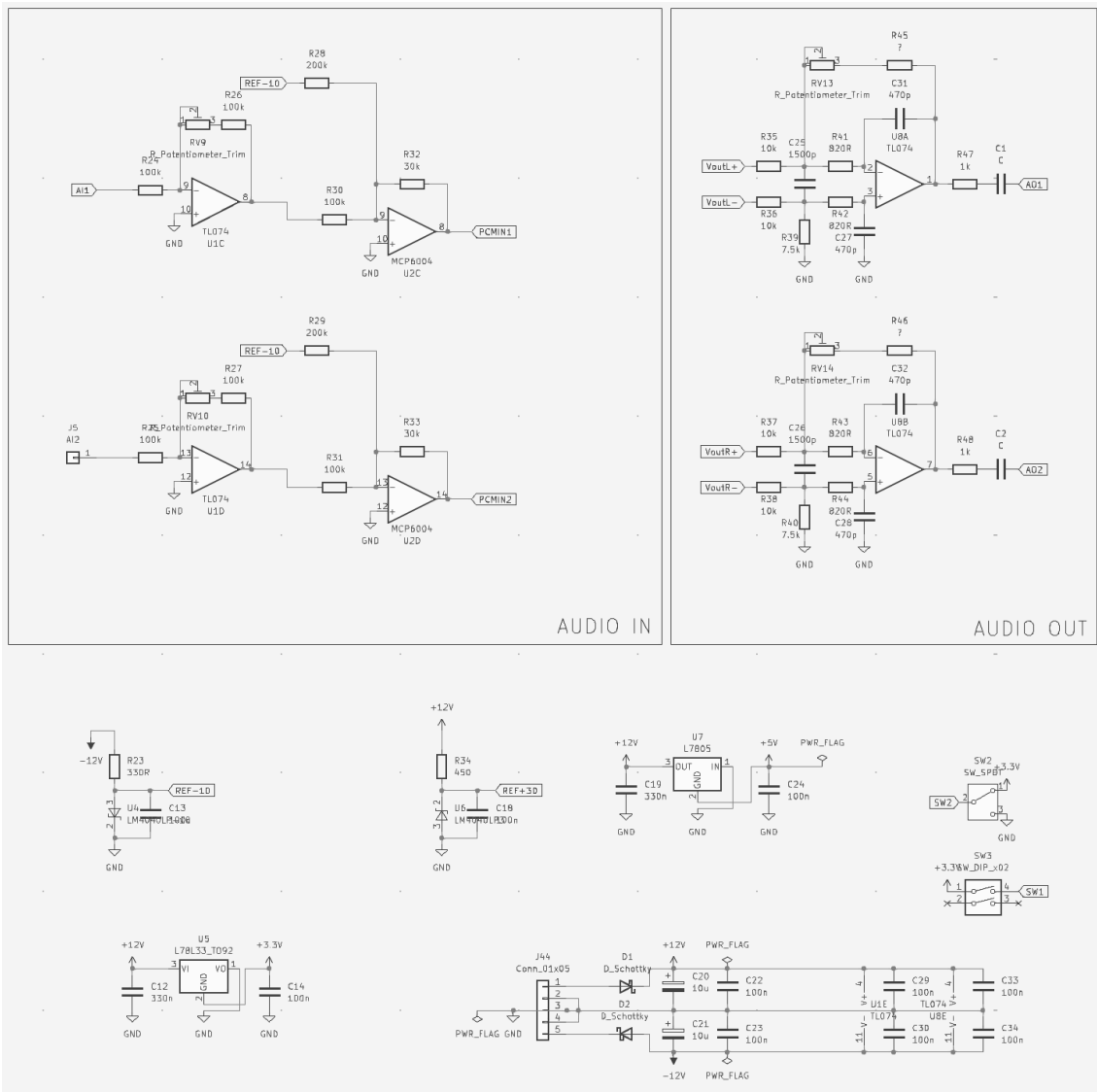
C

Kretsdiagram för systemets plattform

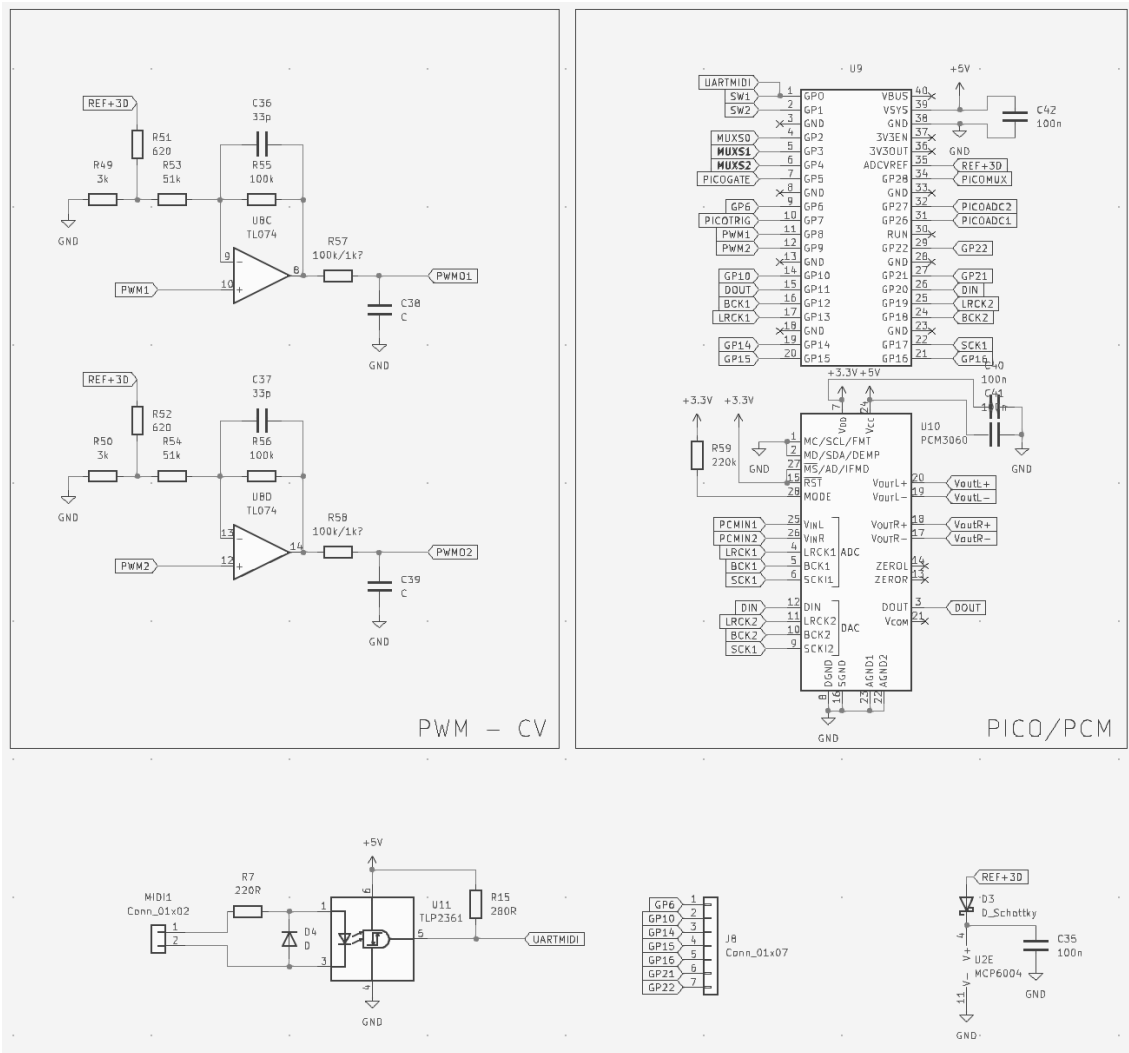


Figur C.1: Kretsdiagram för hantering av inputs av kontrollsignaler, samt multiplexer inklusive potentiometrar.

C. Kretsdiagram för systemets plattform



Figur C.2: Kretsdiagram för ljud in- och utmatning, samtliga spänningsregulatorer och referenser, och slutligen strömförsörjning och omkopplare.



Figur C.3: Kretsdiagram för utmatning av kontrollsignaler, MIDI-in och pinfördelning för både RP Pico och PCM3060.