

Victor Samuel Burns Lykkås

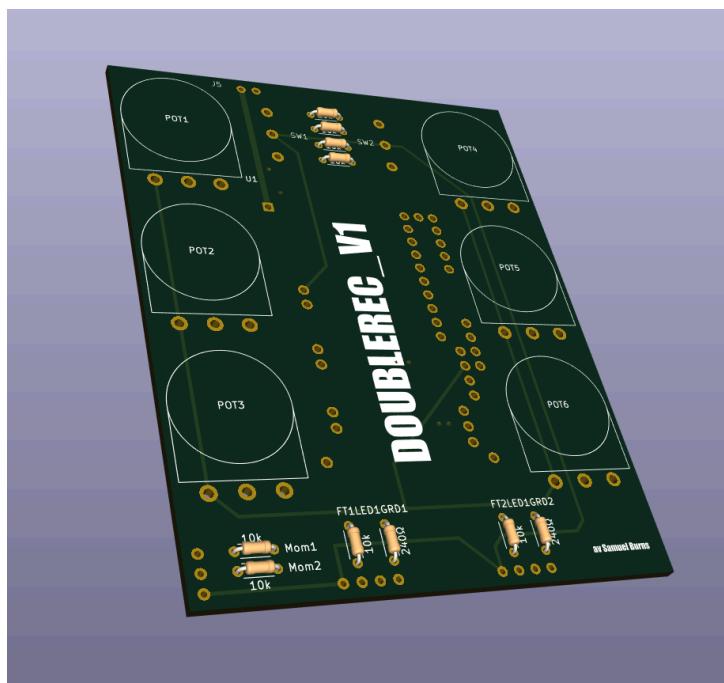
Fra konsept til skapelse

En gitarpedal som bruker granulær syntese for å inspirere låtskriving.

Bacheloroppgave i Musikkteknologi

Veileder: Daniel Buner Formo

Mai 2024



NTNU

Kunnskap for en bedre verden

Victor Samuel Burns Lykkås

Fra konsept til skapelse

En gitarpedal som bruker granulær syntese for å inspirere låtskriving.

Bacheloroppgave i Musikkteknologi
Veileder: Daniel Buner Formo
Mai 2024

Norges teknisk-naturvitenskapelige universitet
Det humanistiske fakultet
Institutt for musikk



Kunnskap for en bedre verden

Jeg vil takke min kjære Maja og mine foreldre for støtten og hjelpen dette siste året av bachelorgraden. Takk til min veileder Daniel Buner Formo for gode råd.

Abstract

This bachelor's thesis investigates the product development process of a guitar pedal with a focus on advanced digital signal processing, specifically granular synthesis, and its impact on songwriting. The thesis integrates historical and theoretical perspectives on guitar pedals, the use of microcontrollers and computers in music equipment, and techniques for digital sound processing with practical development work. The research follows a structured product development methodology that encompasses design, hardware and software development, and iterative testing. The final product, a guitar pedal utilizing granular effect processing and ambient-inspired reverb, was thoroughly tested, and used in a creative songwriting process. The results highlight the educational value of the development process, the technical challenges encountered, and the innovative potential of combining granular synthesis with traditional guitar effects to enhance musical creativity.

Sammendrag

Denne bacheloroppgaven undersøker produktutviklingsprosessen av en gitarpedal med et fokus på avansert digital signalprosessering, spesielt granulær syntese, og dens påvirkning på låtskriving. Oppgaven integrerer historiske og teoretiske perspektiver på gitarpedaler, bruken av mikrokontrollere og datamaskiner i musikkutstyr, og teknikker for digital lydprosessering med praktisk utviklingsarbeid. Forskningen følger en strukturert produktutviklingsmetodikk som omfatter design, maskinvare- og programvareutvikling, og iterativ testing. Sluttproduktet, en gitarpedal som benytter granulær effektprosessering og ambient-inspirert klang, ble grundig testet og brukt i en kreativ låtskriverprosess. Resultatene fremhever den pedagogiske verdien av utviklingsprosessen, de tekniske utfordringene som ble møtt, og det innovative potensialet ved å kombinere granulær syntese med tradisjonelle gitareffekter for å forbedre musikalsk kreativitet.

Innholdsfortegnelse

ABSTRACT	2
SAMMENDRAG	4
FIGURER	6
1. INTRODUKSJON	7
1.1 PROBLEMSTILLING	7
1.2 DISPOSISJON	7
2. TEORI	8
2.1 KORT GJENNOMGANG AV GITARPEDALENS TILBLIVELSE	8
2.2 BRUKEN AV MIKROKONTROLLERE OG DATAMASKINER I GITARPEDALER	9
2.3 CSOUND OG GRANULÆR SYNTSE.....	10
2.3.1 <i>Csound</i>	10
2.3.1 <i>Om granulær syntese</i>	11
2.4 OM BRUKEN AV GITARUTSTYR I MUSIKK OG LÅTSKRIVING	11
2.5 ARBEIDETS KONTEKST	12
3. METODE	13
3.1 SPESIFIKASJONER	13
3.2 VERKTØY	15
3.2.1 <i>Kretskort</i>	15
3.2.2 <i>Mikrokontroller</i>	15
3.2.3 <i>Kodespråk</i>	16
4. FREMGANGSMÅTE	16
4.1 FORARBEID	17
4.2 UTVIKLING AV MASKINVARE	17
4.3 UTVIKLING AV PROGRAMVARE	19
5. BRUK AV GITARPEDALEN TIL LÅTSKRIVING	22
6. KONKLUSJON OG VIDERE ARBEID	24
BIBLIOGRAFI	25
VEDLEGG	27

Figurer

Tabell 1 - Kriterier for maskinvare.....	13
Tabell 2 - Kriterier for programvare.....	14
Figur 1 - Gitarpedalens skjematiske diagram.....	17
Figur 2 - Gitarpedalens kretskortdesign.....	18
Figur 3 - Bilde av kretskortet med noen komponenter loddet fast.	18
Figur 4 - Gitarpedalens ferdige konstruksjon.....	19
Figur 5 - Implementeringen av den granulære effektprosesseringen.....	20
Figur 6 - Implementeringen av den ambiente klangeffekten.....	21
Figur 7 - Implementeringen av flangereffekten.....	21

1. Introduksjon

Gitarpedaler har lenge vært en sentral del av elektriske gitaristers verktøykasse, kjent for å gi unike soniske muligheter og utvide uttrykksregisteret til instrumentet. Fra de tidlige fuzz- og reverb-boksene til dagens avanserte digitale enheter, har utviklingen av gitarpedaler fulgt en dynamisk bane drevet av teknologiske fremskritt og kreative behov.

1.1 Problemstilling

Jeg har stor interesse for gitarpedaler og de forskjellige soniske uttrykkene jeg som gitarist får tilgang til gjennom disse. I tillegg er jeg låtskriver og er alltid på utkikk etter nye unike lyder, samt kilder til inspirasjon. Jeg er også opptatt av teknologi og utvikling av teknologi, og det er i kombinasjonen av disse to interessene at denne oppgaven ble til. Jeg skal utforske en produktutviklingsprosess av en gitarpedal fra start til slutt, og bruke pedalen i låtskriving. På grunn av arbeidsmengde avgrensers jeg låtskrivingsdelen av oppgaven til en komposisjon for elgitar og vokal. Med dette som fundament har jeg kommet frem til denne problemstillingen:

Hvorfor lage en gitarpedal selv? Hva kan man lære av produktutviklingsprosessen og hvordan påvirker bruken av denne pedalen egen låtskriving?

1.2 Disposisjon

Oppgaven er strukturert som følger: Først vil jeg presentere en teoretisk gjennomgang av gitarpedalers tilblivelse, bruken av mikrokontrollere og datamaskiner i slike enheter, og teknikker for digital lydprosessering som Csound og granulær syntese. Deretter vil jeg beskrive metodikken og spesifikasjonene for utviklingen av gitarpedalen, etterfulgt av en detaljert fremgangsmåte for design, maskinvare- og programvareutvikling. Til

slutt vil jeg evaluere pedalen gjennom praktisk bruk i en låtskriverprosess, og diskutere de kreative innsiktene som har kommet ut av dette prosjektet.

2. Teori

I denne delen skal jeg gjøre rede for historie og teori som ligger til grunn for de ulike delene av oppgaven. Siden oppgaven er tverrfaglig av natur, finnes det mye forskjellig forskning, historie og vinklinger i fra ulike vitenskapelige områder. Jeg har derfor valgt å avgrense teoridelen til å fokusere på følgende:

- Kort gjennomgang av gitarpedalens tilblivelse.
- Bruken av mikrokontrollere og datamaskiner i gitarpedaler.
- Csound og granulær syntese.
- Om bruken av gitarutstyr i musikk og låtskriving.

2.1 Kort gjennomgang av gitarpedalens tilblivelse

Allerede ved elgitaren og gitarforsterkerens oppfinnelse på 1930-tallet så man at gitarister ønsket å utvikle nye soniske uttrykk for instrumentet (Almås, 2024). En av disse utviklingene var det som ble kalt «the stomp box», og som etter hvert ble det vi gjenkjenner som «gitarpedalen». Den kan spores tilbake til 1962 da musikkstudioteknikeren Glen Snoddy oppfant en effektboks med fotbryter som skulle etterligne en gitarforsterker som hadde rifter i høyttalerelementet (McCabe, 2014). Den moderne betegnelsen på slike effektbokser er gitarpedal, og Lang (2018) beskriver gitarpedalen som «[...] a device that takes an analog signal from a guitar, adds effects to the signal, and outputs the signal to an amplifier» (Lang, 2018, s. 1). Videre forklarer Lang (2018) at gitarpedaler ofte aktiveres med fotbrytere slik at gitaristen skal kunne spille samtidig som gitarpedalene kan skrues av og på (Lang, 2018, s. 1). Gitarpedalen til Glen Snoddy, som senere ble kjent under produktnavnet Maestro Fuzz-Tone, kan man argumentere for at la grunnlaget for formatet gitarpedalen har hatt tiårene siden, altså

en metallboks med fotbryter(e) og eventuelt potentiometer (knotter) for øvrig effektstyring.

Men det soniske uttrykket til elgitaren skulle utvikles videre.

2.2 Bruken av mikrokontrollere og datamaskiner i gitarpedaler

De første gitarpedalene som ble produsert hadde kretser som bestod av analog elektronikk: Transistorer, resistorer, operasjonsforsterkere (Op-amps) og kondensatorer (Lang, 2018, s. 1). Gjennom 1980-tallet så åpnet det seg en mulighet for å produsere gitarpedaler med digitale kretser, da hovedsakelig på grunn av utviklingen av billige og lett tilgjengelige mikrokontrollere (Lang, 2018, s. 1). En mikrokontroller er en integrert krets (IC) som inneholder blant annet ikke-flyktig minne (lagring), arbeidsminne (RAM) og en mikroprosessor. Disse bestanddelene jobber sammen for å kjøre programvare som løser ulike oppgaver (Larsen, 2021). I gitarpedaler brukes ofte mikrokontrollere til å behandle lyd, behandle styringen av pedalens kontrollere og til å behandle informasjon, for eksempel ved å lagre forskjellige innstillinger på pedalen (Trombley, 2017, s. 9 - 25).

I senere tid har mer kompliserte og fullstendige løsninger for digital lyd og styringsbehandling blitt allemannseie. Et eksempel på dette er Spin Semiconductor sin FV-1 IC, som ble designet for bruk i musikkindustrien, nærmere bestemt musikkteknologien (Trombley, 2017, s. 14). Et annet, og mer moderne eksempel, er Bela. Bela er ikke bare en mikrokontroller, men en hel datamaskin som kjører en egenlagd versjon av operativsystemet Linux (Bela.io, 2016 - 2024). Til forskjell fra mikrokontrollere, som ofte trenger egen og spesifikk maskinkode for å kjøre programvare, er Bela kompatibel med en rekke forskjellige kodespråk (Bela.io, 2016 - 2024). Blant annet kodespråkene C++, Supercollider og Csound, samt det visuelle kodespråket Pure Data (Bela.io, 2016 - 2024). På grunn av innovasjonene innen mikrokontrollere og små datamaskiner kan man anta at det har blitt enklere å prototype og designe nye gitarpedaler med kraftige og nye funksjoner. I tillegg åpner det digitale domenet opp for nye effekter og signalbehandlingsteknikker som er vanskelig eller praktisk umulig å emulere med en analog krets. Det er flere prosjekter som bruker Bela

maskinvare tiltenkt gitarpedalformatet. Et eksempel er *Musikverb* som er en klang-effekt som forandrer parameter og innstillinger på klangen basert på en analyse av gitarsignalet (Pereira et al., 2018, s. 356). Selv beskriver Pereira et al (2018) prosjektet slik: «MusikVerb was developed as a embedded guitar pedal system using the BELA platform and as a software standalone application in the Pure Data programming language» (Pereira et al., 2018, s. 360). Et annet eksempel på et lignende prosjekt, som startet på Raspberry PI-plattformen, men som siden har blitt tilpasset Bela, er *The COSMO Project* (Bela.io, 2019). Prosjektet er bygget rundt bruken av kodespråket Csound i utøvende musikalsk setting, og formfaktoren er basert på gitarpedalen, eller som Hofmann et al forklarer: «A classic stomp box design is used to demonstrate how Csound can be brought on stage as a stand-alone hardware effect instrument.» (Hofmann et al., 2016, s. 291).

2.3 Csound og granulær syntese

2.3.1 Csound

Kodespråket Csound har sin opprinnelse fra 1986 og er skrevet av Barry Vercoe, som tok utgangspunkt i kodespråket C (Hofmann et al., 2016, s. 291). Csound har blitt en aktuell plattform for utvikling av lydrelatert maskinvare på bakgrunn av noen konkrete egenskaper. Fordi Csound er en programvare med åpen kildekode, kan man tilpasse programvaren til å fungere på mange forskjellige plattformer og på ulik maskinvare (Hofmann et al., 2016, s. 291). Csound er også bakoverkompatibel med alle tidligere versjoner, og siden videre utvikling av Csound også skal være bakoverkompatibel betyr det at all kode vil unngå å bli utdatert (Hofmann et al., 2016, s. 291). Det finnes også gode ressurser tilgjengelig for alle som ønsker å lære og bruke Csound, blant annet den interaktive manualen og opplæringsnettstedet *The Csound FLOSS Manual* (Heintz, 2023). Den brede kunnskapsdelingen og ressurstilgangen for alle interesserte har helt sikkert vært utslagsgivende faktorer for at Csound er det valgte rammeverket for *The COSMO Project* (Hofmann et al., 2016, s. 291).

2.3.1 Om granulær syntese

Som tidligere nevnt åpner bruken av digital prosessering opp nye muligheter for effekter og signalbehandling. Hovedfokus i denne bachelorrapporten er bruken av granulær syntese og granulære prosesseringsteknikker. Granulær syntese er en teknikk hvor man produserer mange små biter med lyd, og sammensetningen av disse lydbitene danner den nye lyden (Lazzarini et al, 2016, s. 337). Innholdet, lengden, periodisitet og måten lydbitene blir sammensatt, har alle en påvirkning på hvordan sluttresultatet høres ut (Lazzarini et al, 2016, s. 337). En bit med lyd blir kalt en grain og er ofte veldig korte lengder med lyd. Vanligvis brukes en sammensetning av en større mengde grains for å danne det nye lyd materialet (Lazzarini et al, 2016, s. 337). Curtis Roads (2001) beskriver prosessen slik: «A single grain serves as a building block for sound objects. By combining thousands of grains over time, we can create animated sonic atmospheres.» (Roads, 2001, s. 87). Her er det viktig å presisere at grains ikke nødvendigvis trenger å være korte lengder, og at de granulære prosesseringsteknikkene som eksisterer i Csound tillater en fleksibel tilnærming til granulær prosessering (Lazzarini et al, 2016, s. 338). Granulær syntese omhandler bruken av disse teknikkene på lyd som enten er innspilt eller som blir generert innad i teknikken, men man kan også bruke granulær prosessering på lyd i sanntid. Dette kalles granulær effektprosessering (Lazzarini et al, 2016, s. 343). Forskjellene beskrives av Lazzarini et al slik: «Technically, the creation of audio grains is the same in both cases, only the source material for grains differs» (Lazzarini et al, 2016, s. 343).

Det er liten tvil om at bruken av granulær syntese kan påvirke komposisjons- og låtskriverprosesser, da det er et unikt verktøy for komponister som lar seg inspirere av nye soniske uttrykk.

2.4 Om bruken av gitarutstyr i musikk og låtskriving

Det er tydelig at forskjellige gitarlyder- og toner, ofte fungerer som sjangermarkør og at dette er et fenomen som har historisk grunnlag (Herbst, 2020, s. 75). I en artikkel som

omhandler gitaristers forhold til eget musikkteknologisk utstyr finner Herbst (2020) at gitarister generelt er opptatt av høy kvalitet på gitar-tonen sin, samt at gitarister er opptatt av å konstruere et personlig uttrykk som gitar-tonen skal ha (Herbst, 2020, s. 90 - 91). Han fant også at ønsket om personlig uttrykk samsvarer med utøvernivå og at profesjonelle musikere ofte var mer opptatt av sin egen gitar-tone på detaljnivå (Herbst, 2020, s. 90 - 91). Herbst (2020) finner en link mellom sjangerforventning, sjanger-tradisjon og det musikkteknologiske utstyret utøvende gitarister velger å bruke (Herbst, 2020, s. 95). Det er naturlig å anta at dette også vil påvirke låtskriverprosessen, da man bruker forskjellig utstyr, også gitarpedaler, for å oppnå spesifikke gitar-toner som samsvarer med sjangerkonvensjoner, spesielt innen rock, blues og metall (Herbst, 2020, s. 95). Dette er også noe som alluderes til av Rideout (2014) som sier at låtskriving kan være at man trekker paralleller mellom musikk og konsepter som er til inspirasjon, og at låtskriveren bruker denne inspirasjonen som grunnlaget for nye ideer eller musikalsk retning (Rideout, 2014, s. 51). Med dette i bakhodet kan vi igjen ta opp bruk av lydprosesseringsteknikker som granulær syntese. Roads (2001) beskriver granulær syntese som en klynge grains som danner teksturer, og at dette tillater flere dimensjoner hvor det kan skje musikalsk utvikling (Roads, 2001, s. 349). Teksturfokuset finner vi også igjen i musikk sjangeren Ambient (Ruud, 2022).

2.5 Arbeidets kontekst

Denne oppgaven bygger på teorien og historikken som tidligere er presentert, og har et tverrfaglig fokus hvor jeg eksperimenterer med gitarpedalformatet og avansert digital signalprosessering. Målet er å knytte gitarpedalens estetiske og praktiske valg mot låtskriving og musikkutøvelse. Ved å kombinere gitarpedalens bruksverdi med avansert lydprosessering, som granulær effektprosessering, mener jeg at man får et inspirerende verktøy for låtskriving. Dette er spesielt relevant for de med musikalske idealer som trekker gitar-tonen i en mer ambient retning. Som tidligere nevnt finnes det flere prosjekter som utforsker både Csound og Bela i kontekst av en gitarpedal. Imidlertid setter jeg slikt arbeid i en ny kontekst ved å fokusere på granulær effektprosessering, og ved å integrere gitarpedalens estetiske og praktiske valg i låtskriverkonteksten.

3. Metode

Denne oppgaven følger en produktutviklingsprosess helt frem til den praktiske bruken av sluttproduktet. Det har ført til et stort fokus ikke bare på programmeringen av lyder og selve den estetiske signalbehandlingen, men også på det funksjonelle designet av gitarpedalen. Dette har ledet til en kreativt utforskende arbeidsmetode som har bestått av fire faser: Først må gitarpedalens formål og spesifikasjoner utformes. Etter dette skal maskinvare designes og realiseres. Deretter skal programvare designes og realiseres. Til slutt skal sluttproduktet brukes i kreativ låtskriving. Denne prosessen er dynamisk, og det vil gjøres justeringer og forbedringer løpende for å sikre best resultat. Jeg skal redegjøre og reflektere rundt valgte spesifikasjoner.

3.1 Spesifikasjoner

Formålet med gitarpedalen er å gi meg tilgang til ambient-inspirert klang og granulær effektprosessering. Gitarpedalen skal ha en formfaktor som ligner på andre kommersielle gitarpedaler. Dette skal sørge for god samhandling med andre gitarpedaler, og plassering på pedalbrett eller lignende. Brukeropplevelsen skal komme så nær en kommersiell gitarpedal som mulig, noe som også har gitt den et navn: «Limerence».

For å oppnå et sluttprodukt som er av høy kvalitet, både sonisk og i brukeropplevelse, må designkravene spesifiseres. I tabellene under beskrives kriteriene for maskinvare og programvare for gitarpedalen «Limerence»:

Tabell 1 - Kriterier for maskinvare.

Kriterier: Maskinvare	Praktisk begrunnelse
Bruke boks fra Hammond Manufacturing (eller lignende).	Gitarpedalen vil ha dimensjoner som gjør den lik andre gitarpedaler. Dette vil gjøre det lettere å integrere den i allerede eksisterende oppsett. Metallbokser fra Hammond Manufacturing er billige og enkle å oppdrive.

Stereo inn/ut.	Gitarpedalen vil kunne implementeres i signalkjeder som er stereo.
Fotbryter av/på per effekt.	Gitarpedalen vil ha bedre brukeropplevelse hvis effektene kan skrues av og på uavhengig fra hverandre. Dette gjør designet med fleksibelt.
Lys av/på per effekt.	Gitarpedalen vil fortelle brukeren hvilke effekter som er av/på.
Tre potentiometer for parameterstyring per effekt.	Gir brukeren kontroll over effektene. Av erfaring gir tre parameter mulighet for en rekke forskjellige innstillinger og derav lyder.
Mulighet for tilkobling av to eksterne fotbrytere.	Gir brukeren ekstra kontroll med føttene, utenom av/på styring av effektene.
To tre-veis brytere for parameterstyring.	Gir brukeren mer kontroll av parameter som ikke trenger finjustering.
Konstruksjon som bruker kretskort.	Kretskort vil gjøre konstruksjonen av maskinvaren enklere. Konstruksjonen vil være sterk og stabil med et kretskort. Siden man kan få plass til flere koblinger på mindre området vil bruken av kretskort også hjelpe med plassproblematikk inne i Hammond-boksen.
Bruke standard 9V (senter negativ) strømtilførsel.	Gitarpedalen vil være kompatibel med vanlige strømforsyninger tiltenkt gitarpedaler.

Tabell 2 - Kriterier for programvare.

Kriterier: Programvare	Praktisk begrunnelse
Minimum to effekter.	Gitarpedalen vil ha flere soniske muligheter, og vil også gi brukeren mulighet til å utforske kombinasjonen av effektene.
Mulighet for kontroll over rekkefølgen på effektene.	De to effektene vil kunne kombineres på forskjellige måter når rekkefølgen forandrer seg. Dette gir gitarpedalen større bruksverdi og åpner opp for en fleksibel tilnærming av hele pedalen.

Høy lyd kvalitet.	Gitarpedalen må ha høy lyd kvalitet og ikke destruere gitarsignalet ut over det som er tenkt.
Minimalt med lydforsinkelse (latency).	Lydforsinkelse kan påvirke spilleopplevelsen negativt, og siden gitarpedalen skal brukes som et kreativt verktøy er det viktig at den oppleves god å bruke.

3.2 Verktøy

I denne delen skal jeg presentere hvilke verktøy som ble brukt under produktutviklingsprosessen, og kort begrunne hvorfor disse verktøyene ble valgt.

3.2.1 Kretskort

En viktig del av maskinvareutviklingen, var selvet designet av kretskortet. Til denne oppgaven ble programvaren KiCad benyttet, og det var tre særlige grunner for dette valget:

- 1) KiCad er gratis å bruke.
- 2) KiCad har et aktivt brukermiljø, med forum hvor det er lett å både stille spørsmål, diskutere, lære og få praktiske tips og svar på spørsmål.
- 3) Oppgavens veileder, Daniel Buner Formo bekreftet at KiCad, på tross av at det er Free-ware, er et hensiktsmessig verktøy å benytte til formålet.

3.2.2 Mikrokontroller

En annen viktig avgjørelse i maskinvareutviklingen, var hvilken plattform, eller mikrokontroller/datamaskin som skulle benyttes. Valget ble Bela Mini, av fire hovedårsaker:

- 1) Bela Mini har relativt høy prosessorkraft.
- 2) Den er liten nok til å kunne integreres i Hammondboxen som var foretrukket.
- 3) Den støtter Csound.

- 4) Bela Mini gir tilgang til Bela IDE, som er et enkelt og lettvent utviklingsmiljø, noe som førte til at selve programmeringen ville være smidigere å gjennomføre.

Jeg vurderte også å bruke mikrokontrolleren Daisy Seed, da denne ligger i et langt lavere prissegment og er enda mindre i størrelse. Men det faktum at Daisy Seed har lavere prosessorkraft og at den ikke støtter Csound direkte, førte til at Bela Mini sine egenskaper veide tyngst.

3.2.3 Kodespråk

Csound er det siste verktøyet som inngår i denne redegjørelsen. Hoffmann et al. gav i sin artikkel «Csound Instruments On Stage» (Hofmann et al., 2016, s. 291) begrunnelser for bruk av nettopp kodespråket Csound. Disse årsakene overlapper i stor grad med begrunnelsene for å bruke Csound i arbeidet med gitarpedalen «Limerence». ¹ I tillegg må det nevnes at Csound er det kodespråket jeg personlig har mest erfaring med, og naturlig nok vil dette gjøre arbeidet enklere. Erfaringen med Pure Data, Python og C++ ville resultert i et enklere og mindre tilfredsstillende sluttprodukt.

4. Fremgangsmåte

I denne delen av oppgaven vil det faktiske og konkrete arbeidet som ble gjort for å lage gitarpedalen «Limerence» presenteres. Dette inkluderer redegjørelser for både tilnærminger og beslutninger som ble gjort under denne arbeidsprosessen. Teorien og teknikkene beskrevet i teoridelen av oppgaven utgjør det akademiske grunnlaget for valgene som ble tatt. Arbeidsprosessen vil bli presentert i kronologisk rekkefølge og fokuset på innholdet vil omhandle hvordan designkriteriene som er spesifisert, er

¹ Se oppgavens pkt. 2.3.1 «Csound»

oppfylt.² Hver del avsluttes med en refleksjon rundt endelig resultat og om kriteriene i spesifikasjonen ble møtt.

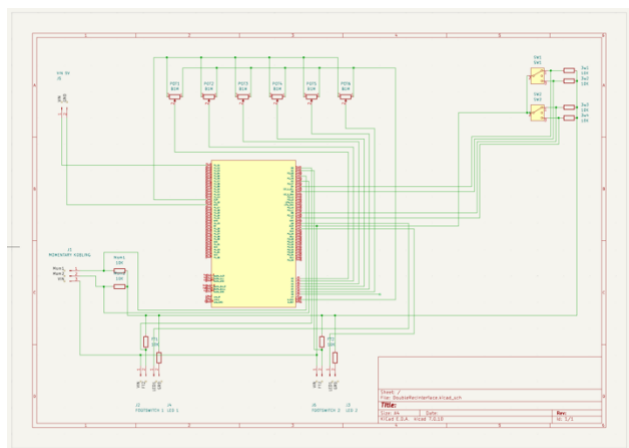
4.1 Forarbeid

Første del av arbeidet var å planlegge designkravene til gitarpedalen, fordi dette ville sette føringer for hvordan resten av arbeidet skulle gjennomføres. Under denne delen av prosessen ble det brukt mye tid på å tegne pseudokode, bestemme hvilke effekter gitarpedalen skulle ha og etter hvert bestille deler. Pseudokode er en måte å fremstille og beskrive programvare, og består ofte av uformelle beskrivelser av kode og flytdiagram som beskriver ønsket signalgang (Nätt, 2023). En stor andel av forarbeidet var også læring av nye teknikker og programvare, da i all hovedsak kretskortdesign i KiCad.

4.2 Utvikling av maskinvare

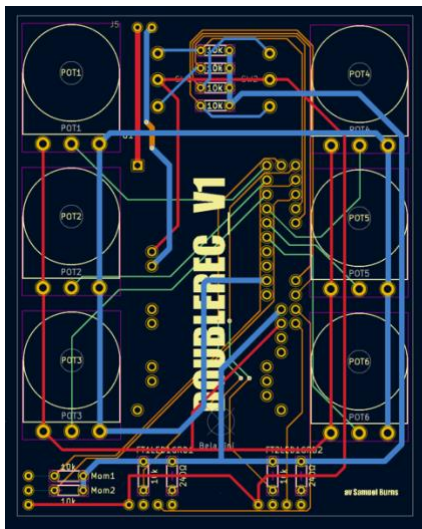
Utviklingsprosessen av maskinvare startet med kretskortdesign. Første gjøremålet var å designe et skjematisk diagram av gitarpedalen. Dette definerer alle koblinger som skal gjøres, samt redegjør for alle komponentene som er en del av disse koblingene. Spesifikasjonstabellen som var en del av forarbeidet, var viktig i denne fasen. Denne tydeliggjorde hvilke komponenter som skulle være inkludert.

Alle funksjonene og komponentene som er relatert til maskinvaren måtte designes inn i det skjematiske diagrammet.



Figur 1 - Gitarpedalens skjematiske diagram.

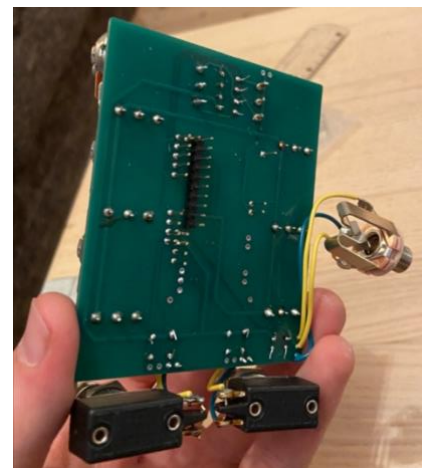
² Se oppgavens pkt. 3.1 «Spesifikasjoner»



Figur 2 - Gitarpedalens kretskortdesign.

Neste steg var å gjøre selve kretskortdesignet i KiCad. Det skjematisk diagrammet ble rammeverket for kretskortdesignprosessen. Alle komponentene som var en del av det skjematisk diagrammet, skulle nå plasseres fysisk på kretskortet, med automatisk tilrettelagte koblinger. Her måtte det tas forbehold om at koblingene faktisk fikk fysisk plass på kretskortflaten, og at det ble valgt rett type koblingspunkter slik at lodding av komponentene var fysisk mulig å gjennomføre.

For å minimere støy i lyd- og kontrollsignalene måtte koblingene plasseres på forskjellige lag. Det er vanlig å skille mellom et strømførende lag og et lag som er jordet. Fordi det var mange komponenter som skulle kobles, valgte jeg å designe et kretskort med fire lag. Et lag til strømførende koblinger, to lag som holdes av til kontrollsignaler og et som holdes av til jording. Den siste delen av denne prosessen var å eksportere det ferdige designet og sende det til produksjon. Da dette ble gjort, var det også på tide å sende bestilling på resten av komponentene som skulle benyttes i gitarpedalen.



Figur 3 - Bilde av kretskortet med noen komponenter loddet fast.

Konstruksjonen av maskinvaren tok utgangspunkt i kretskortet. De forskjellige komponentene ble så loddet fast på kretskortet. Jeg designet koblingspunkter som bestod av hull som går gjennom hele kretskortet. Komponentene er laget med ben som kan tres gjennom disse hullene, og vil lage en kobling når de loddet fast. Til slutt drilllet jeg hull i Hammond-boksen som samsvarte med komponentenes plassering på kretskortet. Dette tillot meg å tre komponentene for kontrollstyring, Jack-pluggene for lyd og strømpluggen gjennom boksen, og derav feste hele konstruksjonen fast til boksen.

For å oppfylle kriteriene definert i spesifikasjonen, ble kretskortdesign og komponentvalg gjort på bakgrunn av disse. Antall fotbrytere, potentiometer, brytere og lys, var alle gjort rede for i kretskortdesignet. Dimensjonene til kretskortet ble valgt på bakgrunn av Hammond-boksen sine dimensjoner, og jeg kjøpte eget kretskort for strømhåndteringen slik at gitarpedalen blir kompatibel med standard gitarpedalstrømforsyninger.



Figur 4 - Gitarpedalens ferdige konstruksjon.

4.3 Utvikling av programvare

Utviklingsprosessen av programvaren bar preg av iterasjoner og utforskning. Lydrelatert kodearbeid krever mye testing, da koden ikke nødvendigvis høres musikalsk og tilfredsstillende ut med en gang. Dette arbeidet startet med å prototype pedalens effektkode. Hver effekt ble designet separat, før de til slutt ble kombinert. Logikk for styring av gitarpedalens parameter og brukerfunksjon var det siste som ble gjort.

Programvaren består av 3 effekter:

- 1) Ambient klang
- 2) Granulær effektprosessering
- 3) Flanger effekt

Det naturlige startpunktet ble å prototype effektkombinasjonene, og teste disse med elgitar. Dette gav større innsikt i den soniske estetikken til effektene, og resulterte i endring av opprinnelig plan. I prototypen var det integrert en egen vring-effekt, som en del av både klangeffekten og den granulære effekten. Dette resulterte i et rotete og ufokusert lydbilde. Dessuten manglet prototypen ekte granulær effektprosessering. Prototypen hadde emulert en lignende effekt med flere instanser av loopereffekter. En loopereffekt gjør opptak av signalet og spiller opptaket tilbake til brukeren. Det viste seg

å være en dårligere løsning enn først forventet. Jeg valgte å fjerne vring-effekten og jeg ønsket å gjøre den granulære effekt prosesseringen ordentlig.

Etter at prototypefasen var over, startet arbeidet med den granulære effektprosesseringen. Denne effekten trenger et kontinuerlig opptak av gitarsignalet. Et slikt opptak kalles gjerne en buffer. Koden som genererer grains henter lydutsnittene

```
;Granulaer - FX2: Granular effektprosesseringen.
kphraseLen = kPot4      ;Potentiometer 4 kontrollerer parameteret "fraselengde" i Granulaereffekten.
kfeed1     = kPot5      ;Potentiometer 4 kontrollerer parameteret "tilbakekobling" i Granulaereffekten.
kmiks1     = kPot3      ;Potentiometer 4 kontrollerer parameteret "miks" i Granulaereffekten.
ktempo     = kTapTempo  ;Tap tempo sendes fra tap tempo -koden.

iftable1   = 1          ;Opptaks-Buffer 1
iftable2   = 2          ;Opptaks-Buffer 2
iftableLen = giRecLen   ;Bufferlengde

;Bruker Granulaereffekten, lyd sendes inn, blir prosessert, og sendes ut.
aGranulaerL Granulaer aGranulaerInL, kphraseLen, kfeed1, kmiks1, ktempo, iftable1, iftableLen, gkstartFollow
aGranulaerR Granulaer aGranulaerInR, kphraseLen, kfeed1, kmiks1, ktempo, iftable2, iftableLen, gkstartFollow

;Ekstra mix kontroll på effekten - Dette er egentlig bare en volumkontroll på den granulære effektprosesseringen.
aGranulaerL = aGranulaerL * kmiks1
aGranulaerR = aGranulaerR * kmiks1
```

Figur 5 - Implementeringen av den granulære effektprosesseringen.

sine fra denne bufferen. Koden som lagrer lyd til bufferen og leser lyd fra bufferen er basert på kodeeksempel fra *Csound: A Sound and Music Computing System*, spesifikt fra kapittelet om granulær syntese (Lazzarini et al, 2016, s. 343).

For å produsere grains ble opkoden *granule* valgt, fordi den var enkel å implementere og hadde alle funksjonene som trengtes. En opkode er en spesifikk kommando innad i Csound som gjør en spesifikk oppgave. Opkoden *granule* gjør granulær syntese og spytter ut en lydkilde som er det ferdigprosesserte lydsignalet. Da koden for granulær effektprosessering fungerte, skiftet arbeidet fokus til det estetiske. Dette var iterativt arbeid som krevde testing og finjustering for å tilpasse den granulære effekten til gitarsignalet, samt finne innstillinger som hadde god estetisk kvalitet. Målestokken for god estetisk kvalitet omhandlet hvordan effekten responderte til gitarsignalet, om frekvensene i det granulære lydsignalet samspilte med gitarsignalet og om kontrollstyringen på potentiometer ga brukeren nok kontroll på tonaliteten til effekten.


```

;Reverb - FX3: Ambient klangeffekt, med mulighet for selvoscillering.

kmix3 = kPot1 ;Potentiometer 1 kontrollerer parameteret "miks" i klangeffekten.
kfco = 12000 ;Lowpass-filter som kutter ved 12000 Hz.
israte = sr ;Sample-raten er den samme som Bela opererer på.
ipitch = .5 ;Dybdeverdi på tilfeldig variasjon på delaylinjene innad i effekten.

;Hvis den eksterne fotbryteren holdes nede skal verdien overkjøre potentiometer 2 sin verdi.
;Dette tillater fotkontroll over selvoscilleringen.

if kmom1 == 0 && kmom2 != 0 then
| kfbvl = 1.05 ;Verdi over 1 og effekten selvoscillerer
else
| kfbvl = kPot2 ;Potentiometer 2 kontrollerer parameteret "tilbakekoblingsnivå" i klangeffekten.
endif

;Bruker klangeffekten, lyd sendes inn, blir prosessert, og sendes ut.
a1, a2 reverb aRevInL, aRevInR, kfbvl, kfco, israte, ipitch

;Ekstra mix kontroll på effekten - Dette er egentlig bare en volumkontroll på klangeffekten.
aRevL = a1 * kmix3
aRevR = a2 * kmix3

```

Figur 6 - Implementeringen av den ambiente klangeffekten.

passer bra til gitar, da frekvensene i lydsignalet samhandler godt, og den tillater lang klangtid, noe man kan argumentere passer bra hvis man har et ambient klangideal. Videre jobbet jeg med en flanger-effekt. Dette er en effekt som oppnås ved å doble lydsignalet, samt å tidsforskyve og faseforskyve doblingen. Inkluderingen av denne effekten ble mer en bonusfunksjon. Jobben til flanger-effekten er å skape større stereofølelse ved bruk av to gitarforsterkere.

Gitarpedalens maskinvare skal styre parameter i programvaren, for at dette skal fungere slik man kanskje forventer må programvaren inneholde logikkode. For enkelthets skyld defineres logikkode i denne oppgaven som: All kode som ikke

omhandler eller påvirker lydsignalene. De funksjonene som ble implementert er fotbryterne som skrur effektene av eller på, potentiometer som forandrer parameter på effektene, og brytere som blant annet forandrer rekkefølge på effektene i pedalens signalgang. Utviklingen av logikkoden var en iterativ prosess hvor testing av funksjonene var essensielt for å garantere at funksjonene opererte på en stabil og hensiktsmessig måte.

```

;Flanger - FX1: Random tapewarble / flanger.

;Treveis-bryteren bestemmer tidsforsinkelsen til Flangereffekten.

if ktg2p1 != 0 && ktg2p2 == 0 then
| atime = 0.005
elseif ktg2p1 == 0 && ktg2p2 == 0 then
| atime = 0.02
elseif ktg2p2 != 0 && ktg2p1 == 0 then
| atime = 0.04
endif

kmix = kPot0 ;Potentiometer 0 kontrollerer mix på Flangereffekten.

;Bruker Flangereffekten, lyd sendes inn, blir prosessert, og sendes ut.
aFlangerL Flanger aL, kmix, atime
aFlangerR Flanger aR, kmix, atime

```

Figur 7 - Implementeringen av flangereffekten.

For å oppfylle kravene i spesifikasjonen ble det implementert to hovedeffekter. Begge disse effektene har valgbar rekkefølge med hverandre, og det ble brukt mye tid på å finjustere parameter slik at lyd kvaliteten var høy. I tillegg ble testing av pedalen utført etter at alle funksjonene var ferdigstilte slik at Bela sine innstillinger som omhandler lydforsinkelse og prosessorkraft ble justert slik at det er minimalt med lydforsinkelse.

5. Bruk av gitarpedalen til låtskriving

Det finnes flere måter å anvende en gitarpedal til låtskriving og kreativ musikkutøvelse. Eksempel på noen av disse metodene er:

- **Idéskaping og lek:** Gitarpedaler, spesielt de med unike effekter (slik som granulær syntese), kan bidra til å skape nye idéer. Den teksturbaserte effekten kan produsere uventede lyder, som igjen kan bli grunnlaget for nye gitarriff, melodier eller akkordprogresjoner. Det å bruke tid på hva jeg best kan beskrive som «lek», i betydningen å bare spille, prøve ut og «leke seg» uten et klart mål vil erfaringsvis ofte lede frem til spennende innskytelser som senere benyttes i låtskriving.
- **Arrangement:** Muligheten til å manipulere gitarpedalens lydmateriale og innstillinger i sanntid, kan sette føringer for hvordan en låt skal arrangeres. Et eksempel på dette vil være å bare bruke klangeffekten i pedalen på versene, og skru på den granulære effekten på refrengene. Måten disse effektene tar plass i lydbildet kan påvirke bruken av andre instrumenter, eller sette føringer for hvordan vokalen skal utformes. Dette fungerer også i motsatt rekkefølge, for eksempel ved at vokalen setter føringer for hvor effektene i gitarpedalen skal brukes.
- **Fremføring:** Gitarpedaler er en viktig del av en gitarist sitt soniske uttrykk, og spiller en instrumental rolle i fremføring av musikk. Bruken av gitarpedaler i en framførings-setting kan også åpne opp for å gjenskape avanserte

studioproduksjonsteknikker. I en låtskriverprosess kan man med bruken av gitarpedaler sikre kontinuitet på lydbildet gjennom en hel musikkproduksjonsprosess.

I dette prosjektet ble den ferdig utviklede gitarpedalen brukt mye i kreativ låtskriving. Baktanken med denne låtskriverprosessen var å utforske de praktiske og estetiske fordelene ved å implementere en gitarpedal i låtskriving. Prosessen bar preg av en forståelse for de forskjellige anvendelsesmetodene beskrevet over. Dette er en oversikt over låtskrivingsarbeidet:

- 1) **Konsept og utforskning:** Låtskrivingen startet med eksperimentering med gitarpedalens granulære effekt prosessering. Under denne utforskningen var jeg nøye med å teste forskjellige innstillinger og lyder, samt forskjellige tilnærminger på gitaren. De forskjellige teksturene i effekten la grunnlaget for arbeidet videre.
- 2) **Utvikling av tema og motiv:** Under utforskningen gitarpedalens effekter kom jeg fram til noen akkorder og gitarløp som inspirerte meg. Det var i dette stadiet noe av sangteksten også ble skrevet. I tillegg ble utforskning av klangeffekten også gjort på dette stadiet. Effektbruken resulterte i eteriske og stemningsfulle partier i sangen.
- 3) **Finsliping og slutføring:** Det siste stadiet av låtskrivingprosessen handlet om å bestemme konkret hvordan effektene og arrangementet skulle være. Låtskriving er et kreativt utforskende og iterativt arbeid. Finsliping av de forskjellige lydene, akkordprogresjon, arrangement og tekst bidro til å øke den overordnede kvaliteten av låten.
- 4) **Innspilling:** Som en del av prosjektet gjorde jeg en demoinnspilling av låten. Dette var for å konkret gi et innblikk i hvordan effekten høres ut og hvordan den kan utnyttes kreativt. For å gi et tydelig bilde på denne prosessen er det gjort minimalt av opptak og etter-prosessering av de innspilte lydsporene. Jeg har prøvd å emulere en «Live»-tilnærming til innspillingen.

6. Konklusjon og videre arbeid

Å lage en gitarpedal selv gir en unik mulighet til å skreddersy både lyd og funksjonalitet etter egne preferanser. Kommersielle gitarpedaler kan tilfredsstillende et bredt spekter av effekter, men det er likevel ofte at konkrete ønsker eller behov ikke kan tilfredsstillende av løsningene som er tilgjengelige. Når man designer en gitarpedal, kan man tilpasse funksjonalitet og designe unike effekter. En slik pedal kan bli hjørnesteinen i en gitarist sitt soniske uttrykk. Det er i tillegg flere læringsutbytter ved å gjennomføre en produktutviklingsprosess. Man tilegner seg teknisk kunnskap både innen utvikling av maskinvare og programvare. Det er også viktig med god problemhåndtering og tidsstyring i en slik prosess. I forhold til låtskriving så vil en egenlaget gitarpedal bidra til å gi inspirasjon, nye idéer og en enda dypere kontroll over det musikalske uttrykket.

I framtiden er det muligheter for både mer finsliping av gitarpedalen og mer akademisk utforskning av møtet mellom låtskriver og musikkteknologisk verktøy. Gitarpedalen kan utvikles for å inneholde flere effekter, eller andre unike effekter. Det er også mulig å teste andre mikrokontrollere eller datamaskiner. I forhold låtskriving og musikkteknologisk verktøy, hadde det vært spennende å gjøre mer konkret forskning rundt låtskriveres bruk av gitarpedaler og lignende verktøy.

Denne oppgaven har utforsket en produktutviklingsprosess og sluttresultatet har blitt brukt i låtskriving. Bakgrunnsteori med hovedvekt i *Csound: A Sound and Music Computing System* (Lazzarini et al, 2016), har vært en hovedpilar i utførelsen av arbeidet. I tillegg har den konkrete utviklingen av gitarpedalen bygget videre på arbeid som: *The COSMO Project* (Hofmann et al, 2016), *Design of a Programmable Four-preset Guitar Pedal* (Trombley, 2017), og *Musikverb: A Harmonically Adaptive Audio Reverberation* (Pereira et al, 2018).

Bibliografi

Almås, K. (2024, 14. Mars). El-gitar. *Store Norske Leksikon*. <https://snl.no/el-gitar>

McCabe, A. (2014, 13. Desember). 'Voices Within The Music': A Brief History Of Guitar Effects. *NPR Music*. <https://www.npr.org/2014/12/13/370361269/voices-within-the-music-a-brief-history-of-guitar-effects>

Lang, I. (2018). *Digital Guitar Effects Pedal* [Senior Project, California Polytechnic State University]. Cal Poly. <https://digitalcommons.calpoly.edu/eesp/434/>

Larsen, B. B. (2021, 4. Juni). Mikrokontroller. *Store Norske Leksikon*. <https://snl.no/mikrokontroller>

Trombley, M. (2017). *Design of a Programmable Four-preset Guitar Pedal* [Master thesis, Wright State University]. OhioLINK. https://etd.ohiolink.edu/acprod/odb_etd/etd/r/1501/10?clear=10&p10_accession_number=wright1515591271810386

Bela.io (2016 - 2024). *Bela Knowledge Base*. <https://learn.bela.io>

Pereira, J. P. C., Bernardes, G. & Penha, R. (2018, 4. – 8. September). *Musikverb: A Harmonically Adaptive Audio Reverberation* [Paperpresentasjon]. DAFx-18, Aveiro, Portugal. https://dafx.de/paper-archive/2018/papers/DAFx2018_paper_36.pdf

Bela.io (2019, 3. Mai). Touching opcodes: Learning Csound through Bela. *The Bela Blog*. <https://blog.bela.io/learn-csound-with-bela/>

Hofmann, A., Wærstad, B. I. & Koch, K. E. (2016, 11. – 15. Juli). *Csound Instruments On Stage* [Paperpresentasjon]. NIME 2016, Brisbane, Australia. https://www.researchgate.net/publication/308899957_Csound_Instruments_On_Stage

Heintz, J. (2023, Mars). *The Csound Floss Manual*.
<https://flossmanual.csound.com/introduction/preface>

Lazzarini, V., Yi, S., Ffitch, J., Heintz, J., Brandtsegg, Ø. & McCurdy, I. (2016).
Csound: A Sound and Music Computing System. Springer.

Roads, C. (2001). *Microsound*. The MIT Press.

Herbst, J.-P. (2020). Empirical Explorations of Guitar Players' Attitudes Towards Their Equipment and the Role of Distortion in Rock Music. *Current Musicology*, (105).
<https://doi.org/10.7916/cm.v0i105.5404>

Rideout, R. (2014). *Creativity and Songwriting* [Master thesis, University of Huddersfield]. University of Huddersfield ePrints Repository.
<https://eprints.hud.ac.uk/id/eprint/25896/1/Creativity%20and%20Songwriting%20FINISHED%20-%20final%20thesis.pdf>

Ruud, E. (2022, 21. November). Bakgrunnsmusikk. *Store Norske Leksikon*.
<https://snl.no/bakgrunnsmusikk>

Nätt, T., H. (2023). pseudokode. *Store Norske Leksikon*.
<https://snl.no/pseudokode>

Vedlegg

- i. vedlegg1_LimerencePedal_v4 – Kildekoden lastet ned direkte fra Bela IDE.
- ii. vedlegg2_main.csd – Kopi av kodens hovedfil.
- iii. vedlegg3_Granulaer – Kopi av den egenskrevne opkoden som gjør granulær effektprosessering.
- iv. vedlegg4_Flanger – Kopi av den egenskrevne opkoden som er en flangereffekt.
- v. vedlegg5_hvaOmDemo – Demoinnspilling av min egenskrevne låt: *Hva om*.

