

Anders Wilson Hannigan

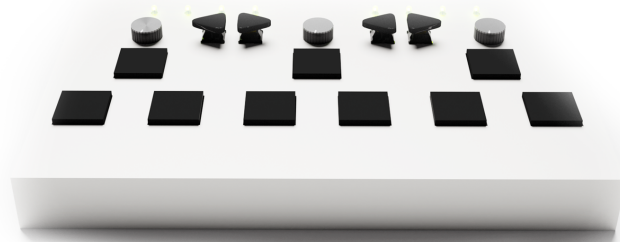
# Utvikling av et frittstående digitalt Instrument

En forsøksorientert og praksisbasert tilnærming til instrumentutvikling

Bacheloroppgave i Musikkteknologi

Veileder: Heather Frasch

Mai 2023





Anders Wilson Hannigan

# Utvikling av et frittstående digitalt Instrument

En forsøksorientert og praksisbasert tilnærming til instrumentutvikling

Bacheloroppgave i Musikkteknologi  
Veileder: Heather Frasch  
Mai 2023

Norges teknisk-naturvitenskapelige universitet  
Det humanistiske fakultet  
Institutt for musikk



Kunnskap for en bedre verden



## **Sammendrag**

Denne oppgaven tar for seg prosessen av å utvikle et frittstående instrument basert på analyse av verktøy jeg ofte utnytter i egen musikalsk praksis. På bakgrunn av denne analysen vil oppgaven redegjøre for ønskede funksjoner instrumentet skal inneha, samt undersøke hva som skal til for å oppnå dette. Det vil bli redegjort for det teoretiske grunnlaget instrumentet bygger på, inkludert undersøkelser av relevant teknologi, og relasjonen mellom teknologi og kreativt resultat. Oppgaven vil også dekke alle faser av utviklingsprosessen, fra prototyping til konstruksjon av det endelige instrumentet. Her vektlegges særdeles utviklingen av egne trykksensitive sensorer, da de utgjør en kritisk del av instrumentets unike funksjonalitet. Avslutningsvis vil det reflekteres over utfordringene som medfulgte de forskjellige løsningene gjort underveis i prosessen, og hvordan de påvirket det endelige resultatet.

## **Abstract**

This thesis addresses the process of developing a standalone instrument based on an analysis of tools I often utilize in my own musical practice. Based on this analysis, the thesis will outline the desired functions the instrument should possess and investigate what is required to achieve this. It will detail the theoretical foundation upon which the instrument is built, including investigations of relevant technology, and the relationship between technology and creative output. The thesis will also cover all stages of the development process, from prototyping to the construction of the final instrument. Particular emphasis is placed on the development of unique force-sensitive sensors, which serve as a critical part of the instrument's unique functionality. Finally, it will reflect on the challenges that accompanied the various solutions made during the process, and how they influenced the final result.

# Innhold

1. Innledning .....	3
1.1 Oppgavens grunnlag og inspirasjon .....	3
1.2 Problemstilling .....	3
2. Teori .....	4
2.1 Den paradoksale effekten av begrensninger i kreativt virke. ....	4
2.2 Maskinvare.....	4
2.2.1 Mikrokontrollere .....	4
2.2.2 FSR sensor .....	5
2.2.3 Andre metoder for brukerinput.....	6
2.3 Programvare .....	6
2.3.1 MIDI protokollen .....	6
2.3.2 Metoder for lydsyntese.....	7
3. Metode og fremgangsmåte .....	9
3.1 Analyse av egen praksis .....	9
3.2 Maskinvare.....	10
3.2.1 Utvikling av lyd-triggering systemet.....	10
3.2.2 Design.....	13
3.2.3 Komponenter og konstruksjon .....	14
3.3 Programvare .....	16
3.3.1 Håndtering av brukerinput.....	16
3.3.2 Lydgenerering .....	18
4. Resultat .....	20
5. Konklusjon.....	21
Referanseliste: .....	22
Vedlegg: .....	24

# 1. Innledning

## 1.1 Oppgavens grunnlag og inspirasjon

Oppgavens konsept stammer fra et ønske om å utfordre min nåværende musikalske praksis for å videreutvikle meg som musiker og komponist. Med bakgrunn i årelang erfaring med piano og digital komponering, har jeg utviklet en unik, hybride stil, en blanding av tradisjonelle og moderne elementer, som kombinerer det organiske med det digitale. Gjennom bruk av digitale verktøy som Digital Audio Workstations (DAWs) og sample-biblioteker, har jeg hatt muligheten til å etterligne og manipulere lydene fra et helt symfoniorkester innenfor mitt eget soverom. Ved å bruke et MIDI-keyboard har jeg også kunnet kontrollere disse instrumentene med det jeg kjenner godt fra før. Denne praksisen har gitt meg tilgang til et mangfold av lyder og instrumenter fra hele verden, noe som har vært avgjørende for å forme min egen musikalske stemme. Likevel har dette potensielt uendelige utvalget av lyder begynt å virke som en hindring i den kreative prosessen. I stedet for å inspirere, har det overveldende antallet valg ofte lammet meg, og skapt usikkerhet rundt hvor jeg skal starte. Å benytte pianoet som mitt primære kontrollinstrument for digitale lyder har utvilsomt gitt meg et solid fundament for min musikalske praksis, ettersom det er et instrument jeg er dypt fortrolig med. Imidlertid har jeg begynt å merke at dette også kan være problematisk og begrensende. Bruken av piano som det eneste grensesnittet for digitale lyder kan begrense evnen til å utforske og uttrykke et bredere spekter av musikalske ideer. Denne begrensningen ligger hovedsakelig i pianoets mekaniske natur, som ikke alltid tillater samme grad av ekspressivitet som andre instrumenter.

Dette dilemmaet har inspirert ideen om å skape et eget skreddersydd musikkinstrument. Med mål om å skape en ny musikalsk plattform som går bort fra den overveldende kompleksiteten i dagens digitale musikkproduksjon, og gir meg en mer umiddelbar, intuitiv og taktil tilnærming til musikkskapning. Ved hjelp av instrumentet skal jeg forsøke å bevege meg vekk fra de nåværende begrensningene jeg har møtt, og åpne opp nye veier for kreativ utforskning.

## 1.2 Problemstilling

Hovedproblemstillingene prosjektet skal forsøke å besvare er:

- *Hvilke muligheter og utfordringer møter jeg i min nåværende musikalske praksis?*

- *Hvilke egenskaper må instrumentet ha for å løse utfordringene, og hva skal til for å implementere disse egenskapene?*

## **2. Teori**

### **2.1 Den paradoksale effekten av begrensninger i kreativt virke.**

Kreativitet anses ofte som en fri og ubegrenset prosess, hvor nye og innovative ideer springer ut fra det tilsynelatende intet. Imidlertid tyder en økende mengde forskning på at begrensninger kan fungere som en katalysator for kreativitet, snarere enn en hindring. Denne motstridende dynamikken mellom frihet og begrensninger i den kreative prosessen fremstår som et paradoks.

En gjennomgang av 145 empiriske studier om effekten av begrensninger på kreativitet og innovasjon gjort av Harvard Business Review antydte at individer, team og organisasjoner alle drar nytte av begrensninger (Acar et al., 2019). Disse begrensningene blir kontraproduktive bare når de blir for høye og da kveler kreativiteten. Ifølge artikkelen viser studiene at færre begrensninger kan lede til at man velger den letteste utvei, i stedet for å tenke kritisk og undersøke andre mulige løsninger. Begrensninger i denne sammenheng fungerer da som en kreativ utfordring som motiverer folk til å søke etter nye løsninger.

I vår digitaliserte hverdag har vi rekordmange verktøy å velge mellom når vi skal skape noe kreativt. Psykologen Barry Schwartz, forfatteren av boken *The Paradox of Choice*, hevder at flere valg ofte har den paradoksale effekten at man velger mindre (Schwartz, 2006). Han hevder valgets paradoks kan lede til økt angst og stress, og selv om Schwartz analyserer det fra et forbruker- og markeds perspektiv, kan man trekke tråder til kreativt virke. I en kreativ prosess kan man påstå at et overflod av valg kan gjøre det vanskeligere å sette i gang, og videre forsinke prosessen da man bruker lengre tid på å velge verktøy i stedet for å bruke dem.

## **2.2 Maskinvare**

### **2.2.1 Mikrokontrollere**

Mikrokontrollere er integrerte kretser som kombinerer de viktigste funksjonene til en datamaskin på en enkelt chip. Disse funksjonene omfatter prosessor (CPU), minne, inngangs- og utgangsterminaler (I/O), og ofte andre komponenter som timere og konvertere mellom



digitalt og analogt signal (ADC og DAC). Mikrokontrollere er kjernen i mange innebygde systemer og elektroniske enheter, inkludert sensorer, roboter, og et bredt spekter av forbrukerelektronikk.

Kjernen til instrumentet består av to mikrokontrollere, en som håndterer brukerinnt og konverterer det til MIDI beskjeder, og en som genererer lyd med basert på MIDI. Håndtering av brukerinnt er gjort av en ItsyBitsy M0 ekspress, og lydgenereringen er gjort på en Pocketbeagle med en Bela Mini kappe. Jeg har valgt å dele prosesseringen slik for å minimere mengde prosessering og kode som må kjøres på Bela mikrokontrolleren da lydprosessering allerede er en intensiv prosess.

ItsyBitsy M0 Express er en mikrokontroller utviklet av det amerikanske selskapet Adafruit Industries i 2018. Den ble valgt grunnet kapasiteten for 12 analoge inngangsterminaler, samtidig som den har mulighet til å emulere USB enheter som vanlige MIDI-kontrollere. Dette betyr at den kan brukes med andre enheter uten å måtte ha eksterne drivere og lignende. Dette er spesielt nyttig i denne situasjonen, da den Bela baserte mikrokontrolleren har støtte for USB-MIDI inn. En annen grunn til at prosjektet baserer seg rundt denne mikrokontrolleren, er at det enkelt kan programmeres ved hjelp av det populære integrerte utviklingsmiljøet til Arduino mikrokontrollere.

Pocketbeagle og Bela Mini kappen ble valgt da lydgenerering og prosessering er en veldig krevende oppgave som de mest populære mikrokontrollerene ikke er kraftige nok til å håndtere. Bela-kappen inkluderer også en stereo-lydutgang som er essensiell for å sende lyd ut av instrumentet. Selskapet bak Bela kappene har også utviklet et eget integrert utviklingsmiljø som lar deg programmere mikrokontrolleren i programmeringsspråk laget spesifikt for lydprosessering. De har også laget en utfyllende dokumentasjon som gjør det enkelt å bli kjent med Bela systemet.

### **2.2.2 FSR sensor**

Force-Sensing Resistors (FSR) er enheter som tillater kvantitativ måling av kraft eller trykk. Disse komponentene fungerer på prinsippet om resistiv sansing, som innebærer en endring i den elektriske motstanden til et materiale som respons på mekanisk trykk (Fried, u.å.).

En typisk FSR består av tre hoveddeler: en aktiv sensing-flate, et ledende materiale, og et underlag. Den aktive sanse-flaten er vanligvis et tynt, fleksibelt materiale som har ledende polymerer spredt jevnt over. Når en kraft blir påført sanse-flaten, fører det til at de ledende

polymererne kommer nærmere sammen, noe som reduserer den totale motstanden gjennom materialet. Denne endringen i motstand kan deretter måles ved hjelp av en passende krets, som kan omforme denne motstandsverdien til en lesbar utgang, for eksempel en spennings- eller strømverdi.

Bruken av FSR sensorer er sentralt i dette prosjektet, da systemet for å spille toner baserer seg utelukkende på slike sensorer. Dette gjør det mulig å måle hvor hardt hver note skal spilles, samtidig som man kan kontinuerlig endre styrken eller andre parametere samtidig som tonen holdes. Dette omtales oftest som et Aftertouch-system. Da hver knapp har en egen FSR sensor, vil det kunne klassifiseres som et polyfonisk aftertouch system.

### **2.2.3 Andre metoder for brukerininput**

For å videre øke fleksibiliteten til instrumentet har jeg lagt til flere sensorer som lar brukeren kontrollere diverse parameter. Dette inkluderer tre potensiometer, og fire trykkbrytere. Et potensiometer er en justerbar motstand som brukes til å endre mengden av strøm i en krets, eller til å endre spenningen i en del av en krets. Det består av en resistiv bane, en knott eller skyvekontroll som beveger seg langs banen, og tre tilkoblingspunkter eller terminaler. Om man bruker alle tre terminalene vil potensiometeret fungere som en spenningsdeler, og man vil kunne bruke verdiene fra den midterste terminalen til å kontrollere diverse parametere basert på posisjonen til skruknotten.

En trykkbryter er en grunnleggende komponent i mange elektriske kretser. Denne enheten tillater eller forhindrer strømflyt basert på dens tilstand – enten åpen (av) eller lukket (på). Ved å koble disse til en mikrokontroller kan man programmere hvilke funksjoner hver knapp skal ha. Eksempler på nyttige funksjoner i prosjektets sammenheng vil være å flytte MIDI-noter opp eller ned en oktav, samt bytte mellom diverse lyder.

## **2.3 Programvare**

### **2.3.1 MIDI protokollen**

MIDI, eller Musical Instrument Digital Interface, er en standardisert protokoll designet for å forenkle kommunikasjonen mellom digitale musikkinstrumenter og datamaskiner. MIDI-data er i hovedsak en serie av meldinger som forteller en lydenhet (for eksempel en synthesizer eller en datamaskin) hva den skal gjøre (Vandenneucker, 2012). Disse meldingene kan være noter som skal spilles, instruksjoner om hvilken lyd som skal brukes, endringer i volum, og mye mer.

MIDI-beskjeder består som regel av samlinger på tre bytes, som igjen består av 8 bits hver. Disse samlingene består oftest av en status byte som sier hva slags beskjed det er, og to data bytes som sier hva verdien av beskjeden er. Figur 1 viser et eksempel på en typisk MIDI note beskjed.



*Figur 1 – Oppbygningen til en typisk MIDI-beskjed*

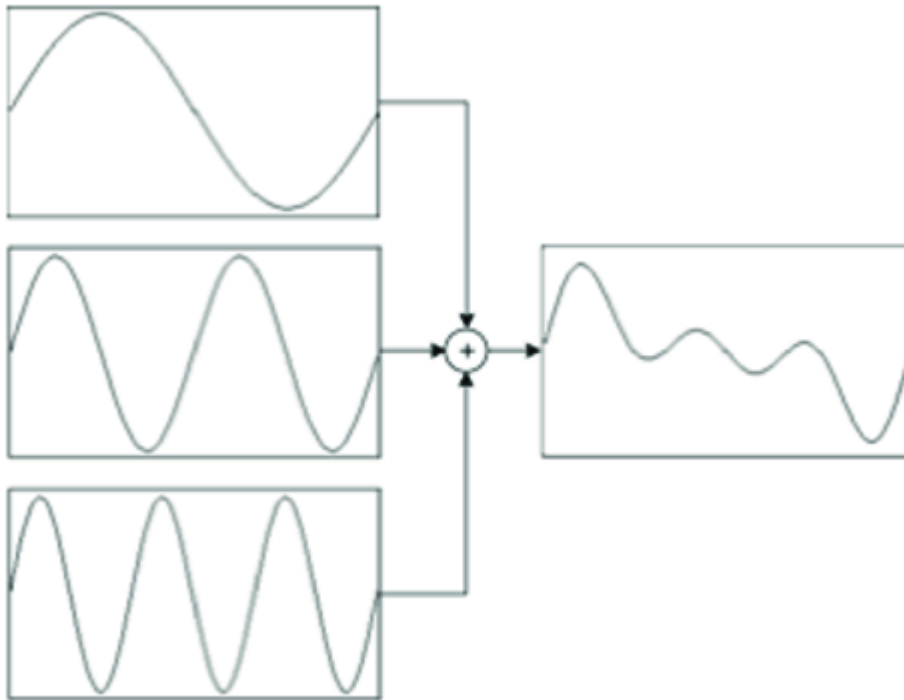
Dette prosjektet bruker en mikrokontroller for å sende MIDI-beskjeder via USB, som forenkler prosessen for å sette opp kommunikasjon mellom enhetene betraktelig.

### 2.3.2 Metoder for lydsyntese

Dette er enkle forklaringer av grunnprinsippene som står til grunne for de relevante syntesemetodene i prosjektet.

#### Additiv syntese

Additiv syntese er en tidlig teknikk for digital - og elektronisk lydsyntese, basert på Fouriers teorem som fastslår at enhver lyd kan skapes fra elementære sinusbølger (Smith, 2010). Denne metoden involverer bruk av store samlinger av sinus-oscillatorer, hver med uavhengige kontroll over amplitude- og frekvens. Til tross for sin allsidighet, har bruksområdet vært begrenset grunnet høye prosesseringskrav. Figur 2 viser et eksempel på en kompleks bølgeform oppnådd via additiv syntese (Researchgate, 2018).



Figur 2 – Et eksempel på additiv syntese

### Fysisk modellering

Fysisk modellering teknikk innen lydsyntese som forsøker å reprodusere naturlig lyd ved å simulere fysiske, akustiske fenomener (Hammer, 1997). Prinsippet bygger på anvendelsen av matematiske modeller og algoritmer som representerer de fysiske egenskapene til et musikkinstrument eller et akustisk system. Istedenfor å generere lyd ved hjelp av forhåndsinnspilte prøver eller ved å syntetisere lyden fra grunnleggende komponenter som i additiv eller subtraktiv syntese, produserer syntese via fysisk modellering lyd ved å simulere de fysiske prosessene som produserer lyd i det virkelige instrumentet. Dette kan omfatte vibrasjonene av en streng og slag mot et trommeskinn for å nevne noen. Denne formen for syntese gir en høy grad av realisme og dynamikk i lydproduksjonen, ettersom den kan ta hensyn til både den grunnleggende klangen til instrumentet og de forskjellige måtene instrumentet kan spilles på (Heintz et al., 2020). Videre tillater fysisk modellerings-syntese manipulasjon og styring av lyden i sanntid, noe som gir musikere muligheten til å uttrykke seg mer presist og med større grad av kontroll.

### Waveshaping

Waveshaping er en teknikk for lydsyntese som involverer transformasjon av en inngangsbølgeform ved hjelp av en ikke-lineær funksjon for å produsere mer komplekse og

interessante bølgeformer (Puckette 2006). Denne prosessen skaper rikere harmoniske strukturer og dermed mer sonisk varierte og komplekse lyder. Da den resulterende bølgeformen tydelig varierer basert på amplitudforskjeller, er det vanlig å inkludere en varierende amplitudekontroller. Waveshaping blir svært ofte brukt for å oppnå forvrengning, også kjent som distortion, på lydsignalet.

## **3. Metode og fremgangsmåte**

Denne delen av rapporten vil gjøre rede for metoder som er tatt i bruk, og hvilken fremgangsmåte jeg benyttet meg av for å gjennomføre prosjektet i sin helhet. Kapittelet vil først ta for seg en analyse av egen praksis for å identifisere hvilke funksjoner jeg etterlyser i et instrument, før den skal ta for seg utviklingen av instrumentet. Her ble det ansett som gunstig å dele utviklingen inn i to delkapitler, maskinvare og programvare, da arbeidet i stor grad bar preg av en todelt fremgangsmåte.

### **3.1 Analyse av egen praksis**

For å forstå min egen prosess bedre, valgte jeg en tosidig tilnærming: jeg gikk gjennom gamle sanger og reflekterte over hvordan de ble til, og jeg satte meg ned for å lage musikk med en tydelig bevissthet om handlingene mine og verktøyene jeg brukte.

I dette arbeidet kom det frem at bruk av eksterne MIDI-instrumenter er en fundamental del av min tilnærming til musikkskapning. I stedet for å skrive ned noter, finner jeg det mer intuitivt og organisk å bruke disse instrumentene. Da mange MIDI instrumenter forsøker å emulere et tradisjonelt piano, finner jeg det svært intuitivt å bruke ettersom jeg selv har bakgrunn som pianist. Det jeg anser som den viktigste funksjonen denne etterligningen gir, er muligheten til å bestemme notene sin styrke i sanntid via MIDI velocity-beskjeder. Til tross for mulighetene dette medfører, finner jeg ofte at jeg automatiserer parametere i etterkant. Dette skyldes hovedsakelig at jeg spiller MIDI piano med to hender, noe som gjør det vanskelig å automere parametere som dynamikk og styrke samtidig. Automeringen gir meg en mulighet til å justere disse parameterne i tillegg til initiell note-styrke, noe som gir et ekstra nivå av ekspressivitet og videre forsterker det organiske lydbilde jeg ofte etterlyser.

I tillegg til bruken av MIDI-piano, jobber jeg alltid i digital musikkprogramvare som åpner opp for stor grad av fleksibilitet når det gjelder produksjon av musikk. De to kjernefunksjonene jeg utnytter her er bruken av flere forskjellige digitale instrumenter for å produsere lyder med stor variasjon, og bruken av digital signalprosessering for å videreutvikle disse lydene. Videre anser jeg det som essensielt å kunne jobbe med flere lydlag som spiller av samtidig, noe som lar meg kombinere disse digitale instrumentene og lydene slik at de lager et komplekst lydbilde.

Basert på denne analysen konkluderte jeg med at jeg ønsker følgende funksjoner i instrumentet:

- Et system som måler hvor hardt du trykker for å spille av en note, og som bestemmer notens styrke eller amplitude basert på denne informasjonen
- Evnen til å modulere parametere til en note etter at den har blitt aktivert, noe som typisk blir referert til som Aftertouch i MIDI protokollen (The MIDI Association, u.å.).
- Muligheten til å velge mellom forskjellige lydgenereringsmetoder, slik at man kan produsere toner med stor variasjon i klangfarge.
- Innebygd digital effektprosessering, spesielt ønsket er klang - og forsinkelseeffekt med feedback.
- En funksjon som lar deg ha flere lydlag som spiller samtidig.

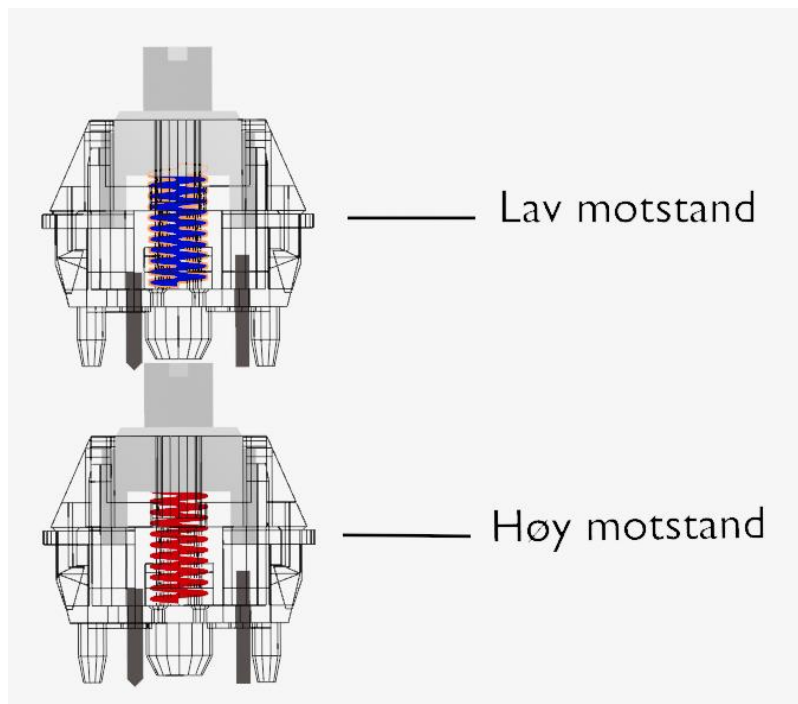
## **3.2 Maskinvare**

Dette delkapittelet skal ta for seg de fysiske aspektene ved prosessen. Dette inkluderer elektroniske komponenter, i tillegg til konstruksjon av instrumentet.

### **3.2.1 Utvikling av lyd-triggering systemet.**

Elektriske instrumenter med trykksensitivitet bruker historisk sett en tidsbasert metode for å kalkulere hvor hardt brukeren trykker på tangenten/knappen. Dette innebærer at hver note aktivering innebærer to trigger-signal som utløses på starten og slutten av brukerens trykk, og ved å kalkulere tidsforskjellen mellom disse signalene vil man få lavere tidsforskjell ved hardere trykk, og motsatt. Denne tidsmålingen kan da konverteres til en verdi som tilsvarer styrke på noten (Kovarsky, 2016). Jeg bestemte meg først for å prøve denne metoden selv. Jeg bestilte to trykkbrytere som opprinnelig er laget for å bli brukt som tastaturknapper. Grunnen

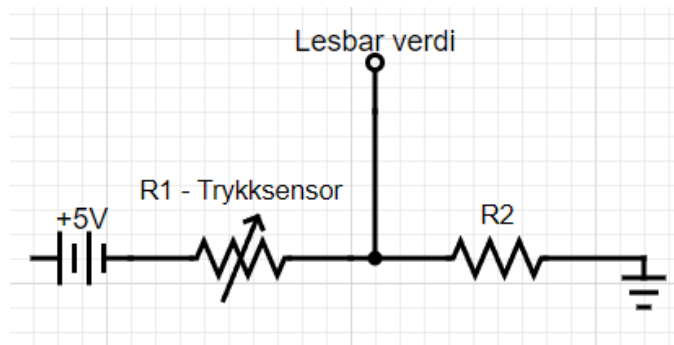
til at jeg valgte dette var at de har innebygd fjæring, og ved å velge to brytere med forskjellig styrke på fjærene hadde jeg en hypotese om at den ene ville aktiveres før den andre, og man vil kunne kalkulere tidsforskjellen. Figur 3 er en digital skisse jeg lagde av dette konseptet.



Figur 3 – En digital skisse av en tidlig bryter-basert prototype

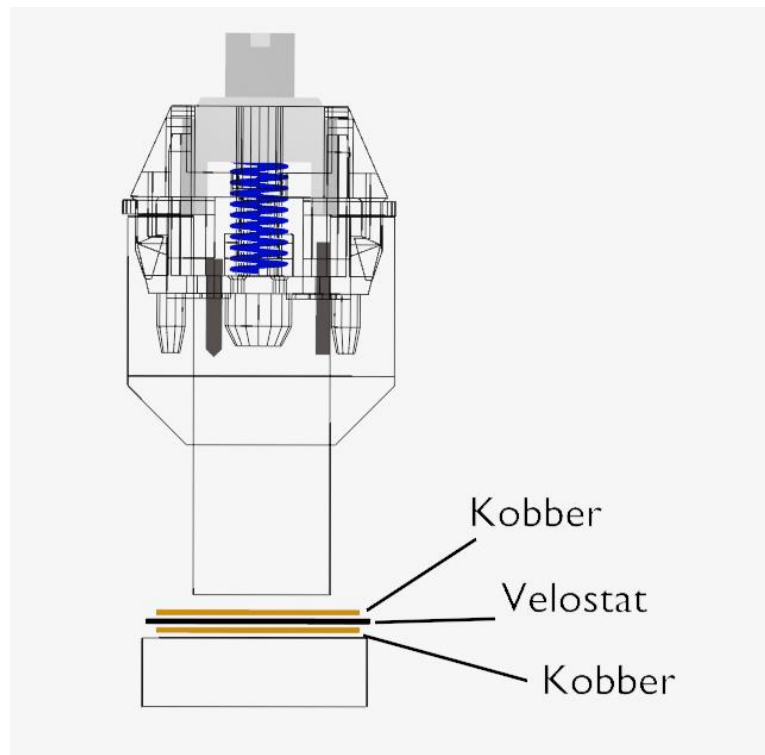
Denne fremgangsmåten ga ikke ønsket resultat, da forskjellen i motstand på fjærene ikke var signifikante nok til å gi stabile og forutsigbare avlesninger. I tillegg til dette fungerer denne metoden kun til å sette en initiell styrke på tonen, og åpner ikke for kontinuerlig justering av parametere.

Etter dette undersøkte jeg bruken av FSR, eller trykksensorer for å oppnå ønsket resultat. Problemet med å bruke ferdiglagde sensorer i dette tilfellet var prisen. På nettbutikken til elektronikksekskapet SparkFun Electronics ligger en FSR sensor på rundt 80 kroner hver, så om hver knapp skulle ha hver sin ville det fort blitt dyrt (SparkFun Electronics, 2023). Ved å undersøke hvordan en slik sensor fungerer, fant jeg ut at man kan bruke et trykk-sensitivt ledende stoff ved navn Velostat for å oppnå samme resultat. Velostat er et materiale som øker og senker elektrisk resistans basert på påført trykk (Dzedzickis et al., 2020). Ved å putte ledende materiale, for eksempel kobber, på hver sin side av dette stoffet, kan man måle denne forskjellen i motstand, og bruke det for å lese hvor mye trykk som er påført. For å lese denne variable motstanden er det nødvendig å bruke en spenningsdeler. Figur 4 viser et enkelt kretsskjema for hvordan et slikt oppsett ville sett ut.



Figur 4 – Et kretsdiagram av en typisk spenningsdeler-krets

Den første prototypen jeg lagde basert på dette prinsippet var en kombinasjon av den tidligere trykkbryter-baserte versjonen, og en Velostat-basert trykksensor. Figur 5 viser en digital skisse av denne versjonen

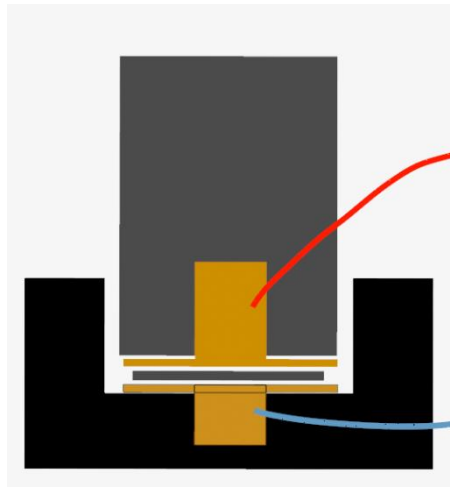


Figur 5 – En skisse av en tidlig prototype basert på en trykksensor

Dette konseptet ga ønsket resultat når det gjaldt kontrollering av parametere etter noten hadde blitt aktivert, men trykkbryteren sin innebygde fjæring ga uønsket demping som hindret en forutsigbar avlesning av den initielle trykkraften ved noteaktivering. Dette resulterte i at jeg bestemte meg for å droppe den opprinnelige ideen som innebar bruk av trykkbrytere, og heller kun forholde meg til de Velostat-baserte trykksensorene.



Den resulterende prototypen besto av en 3D-printet prisme som lå på toppen av en trykksensor, noe som betydde at all trykk fra brukeren vil gå rett på sensoren uten et dempende mellomledd. Figur 6 viser en forenklet illustrasjon av dette



*Figur 6 – En skisse av det endelige grunnkonseptet til de trykksensitive knappene*

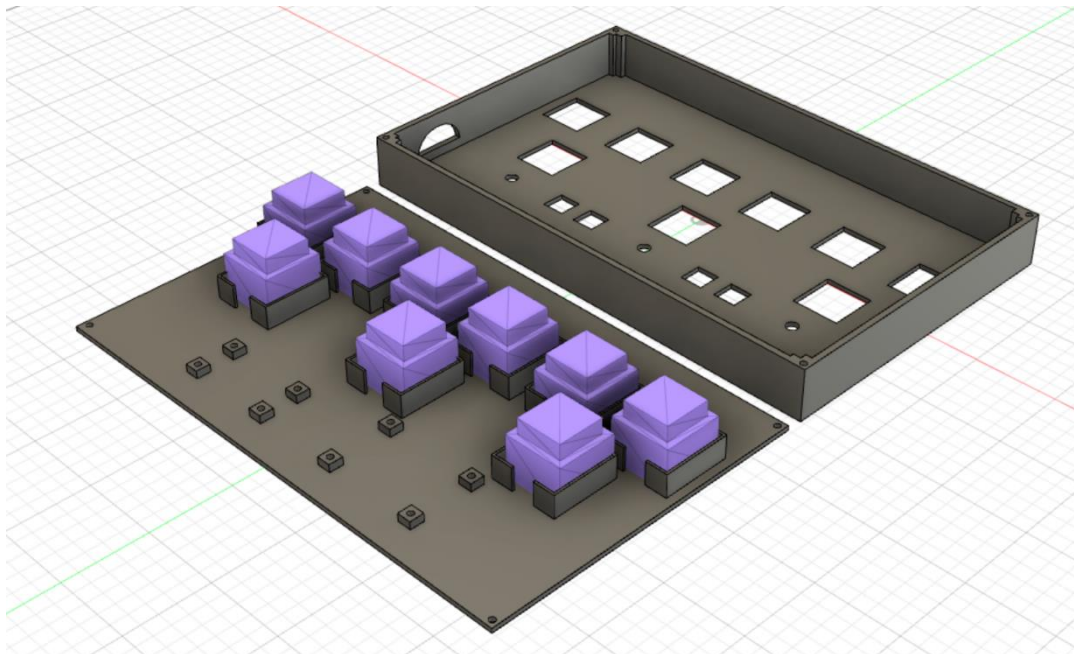
Dette endte opp med å fungere veldig bra, og ved hjelp av videre eksperimentering kom jeg frem til at to lag med Velostat mellom kobberlagene ga best resultat. Denne metoden lot meg få en intuitiv initiell trykkmåling for tonestyrke, samt kontinuerlig kontroll over valgt parameter ved å variere styrken som påføres underveis.

### **3.2.2 Design**

Etter at grunnsystemet for trykkmåling var utviklet, gikk jeg videre til å designe det endelige instrumentet. Basert på det offisielle diagrammet som viser inn og utgangene til Adafruit sin ItsyBitsy m0 Express mikrokontroller og de tidligere ønskene om funksjonalitet landet jeg på følgende komponentliste:

- 9 trykksensitive knapper til trigging av noter
- 4 trykkbrytere til valg av forskjellige moduser
- 3 skruknatter til styring av digital effektprosessering
  - *Dette i kombinasjon med de trykksensitive knappene ledet til at det var behov for 12 analoge innganger. Da jeg leste diagrammet tolket jeg det som at terminalen som brukes til ekstern analog referanse også funksjonerte som en normal analog inngang. Dette var feil, som i praksis betydde at det kun var 11 analoge innganger, og den ene skruknotten ikke kunne brukes med mikrokontrolleren. Dette ble ikke funnet sent i prosessen og det endelige designet inkluderer derfor likevel den tredje skruknotten.*
- 8 LED lys for visuell tilbakemelding til brukeren

Alle disse komponentene skulle settes i et kabinett laget ved hjelp av en 3D-printer grunnet tilgjengelighet. Designprosessen gikk videre ut på å lage små enkeltkomponenter for å teste målene som trengtes for hver komponent, før det så ble kombinert til et helhetlig design. Figur 7 viser et skjermbilde av den endelige prosjektfilen fra programvaren Autodesk Fusion 360.



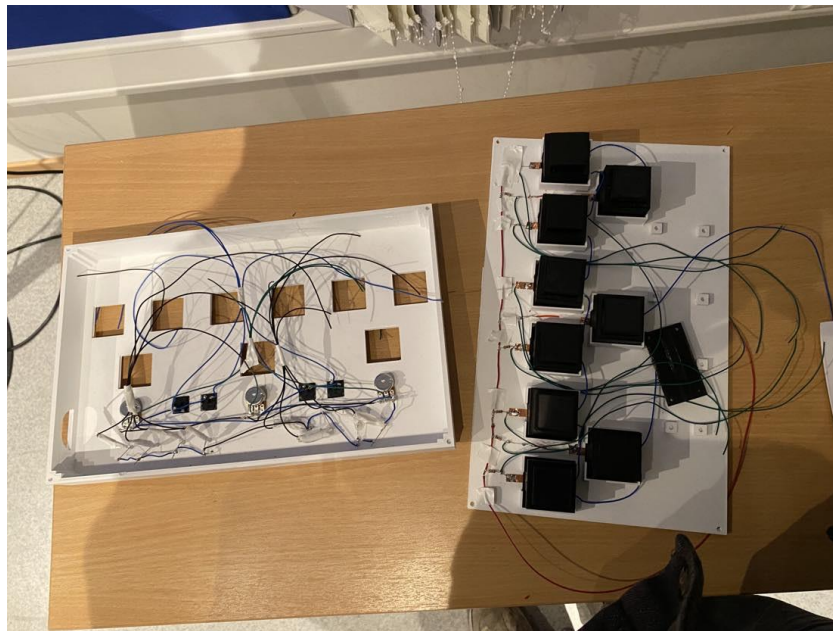
*Figur 7 – Et skjermbilde av det endelige designet*

Ettersom trykksensorene er avhengig av at knappene har fri mulighet til å bli presset nedover, kunne ikke knappene fastsettes til kabinettet. Kabinettet er av den grunn designet til å holde knappene på plass i tillegg til å åpne for fri bevegelse nedover. Dette bydde på utfordringer som vil dekkes i resultatdelen av denne oppgaven.

### **3.2.3 Komponenter og konstruksjon**

Konstruksjonen av instrumentet var en tidkrevende prosess som introduserte flere utfordringer. Jeg startet ved å sette sammen trykksensorene. Dette ble gjort ved å feste kobberteip på bunnplaten av kabinettet og bunnen av hver knapp. Deretter ble det kuttet to firkanter av Velostat til hver knapp, som ble satt mellom kobberteipen og knappen. Ledninger ble da loddet på kobberteipen, slik at strømmen flyter fra knappene til kobberteipene via lagene med Velostat, før det så går til hver sin spenningsdeler (se figur 4). Etter at trykksensorene var montert, festet jeg tre 10k ohms potensiometer, og fire trykkbrytere til kabinettet. Da alle komponentene som skulle brukes til brukerinput var loddet på og testet, festet jeg til slutt åtte

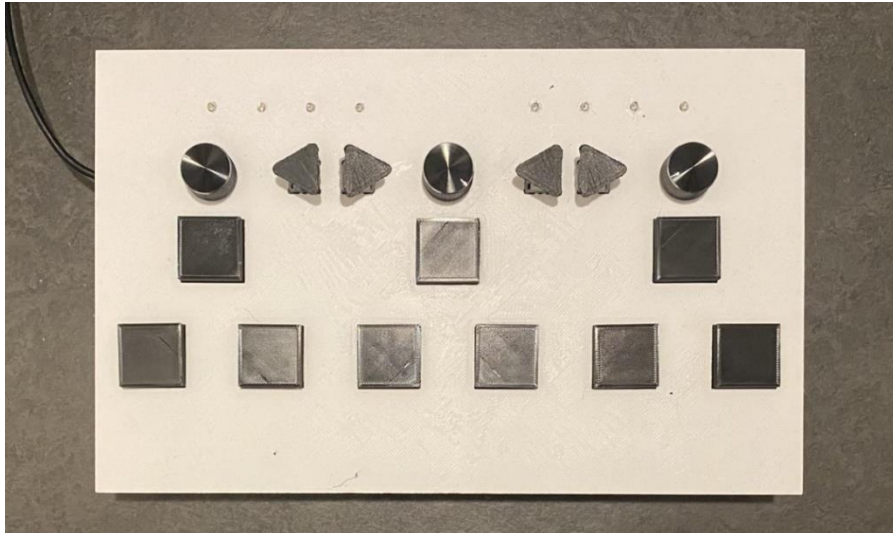
LED lys som skulle brukes som visuell tilbakemelding, slik at brukeren vet hvilken modus som er aktiv. Figur 8 viser et bilde tatt underveis i prosessen.



*Figur 8 – Et bilde fra monteringsprosessen*

Mengden ledninger i kombinasjon med at knappene ikke kunne settes fast i kabinettet gjorde at det ofte ble fysiske forstyrrelser og uforutsigbare resultater til trykksensoren. En annen utfordring var plassen ledningene opptok. Designet på kabinettet ble utformet digitalt og da Bela mikrokontrolleren i teorien hadde fått plass, gjorde ledningene det vanskelig å feste den på en sikker måte. Resultatet ble da at Bela-kontrolleren måtte festes på utsiden, noe som ikke var gunstig. Disse problemene kunne blitt løst ved hjelp av et printet kretskort, men grunnet tid og tilgjengelighet var ikke dette en mulighet.

Figur 9 er et bilde tatt av det ferdig konstruerte instrumentet.



Figur 9 – Utseende til det ferdig-konstruerte instrumentet

### 3.3 Programvare

Dette delkapittelet skal ta for seg delene av prosessen som baserer seg på utvikling av skreddersydd programvare til instrumentet. Dette var en todelt prosess som besto av brukerinntut-prosessering, og lydgenerering.

#### 3.3.1 Håndtering av brukerinntut

Elektronisk støy vil alltid være til stede grunnet elektronikkens natur. Av den grunn vil det være nødvendig å prosessere og filtrere signalene som kommer fra ulike sensorer, slik at de gir ønsket resultat. Jeg har av laget et eget prosesseringssystem til hver form for brukerinntut instrumentet bruker. Se vedlegg 1 for den endelige kildekoden. ItsyBitsy M0 Express mikrokontrolleren kan programmeres i språket C++, og jeg har brukt Arduino-infrastrukturen for å jobbe med dette.

#### Analoge avlesninger

Den første metoden jeg brukte for filtrering av analoge signaler fra trykksensorene og potensiometrene var noe som ofte kalles for et *moving average* filter (Gaberman, 2020). Dette baserer seg på å kontinuerlig regne ut gjennomsnittet av en satt mengde med avlesninger, slik at støy og enkelte avvik ikke har stor effekt. Figur 10 viser et pseudokode eksempel på hvordan dette kan implementeres. For potensiometrene fungerte dette godt, spesielt da det ble kombinert med en minsteterskel for hvor mye verdien skal endres før det registreres. Det fungerte derimot ikke like godt for trykksensorene, da jeg etter videre eksperimentering opplevde suboptimale

resultater når det gjaldt den initielle styrkeverdien som ofte ble lavere enn forventet. For å motvirke dette implementerte jeg et system som baserer seg på det samme prinsippet, som var å samle en rekke avlesninger og bruke det som utgangspunkt, men jeg tok kun hensyn til den høyeste verdien i samlingen i stedet for å regne ut gjennomsnittet. Dette resulterte i en mer responsiv avlesning av initiell styrke. Videre skaleres de analoge signalene fra 0 - 1053, til 0 - 127 før det sendes som MIDI ut, noe som også har en filtrerende funksjon.

```
1 function movingAvarage()
2   mengdeTall = 10
3   for (i < mengdeTall; i++)
4     analogAvlesning = analogRead(A0)
5     totalVerdi += analogAvlesning
6   Endfor
7   snittVerdi = totalVerdi/mengdeTall
8 endfunction
9
0
```

Figur 10 – Pseudokode for implementering av filteret

### Digitale avlesninger

Digitale innganger er også utsatt for elektronisk støy, og har behov for etterprosessering slik at det kan bli brukt i praksis. En av de vanligste metodene for dette er en teknikk som heter debouncing. Debouncing gjort ved hjelp av programvare består oftest av å implementere et system som tar tiden på en avlesning, og kun tillater en viss mengde avlesninger i et tidsrom. Figur 11 viser et pseudokode eksempel på hvordan dette kan implementeres. I tillegg til dette sjekkes det ofte flere ganger om resultatet vedvarer, slik at man vet at det er stabilt. I min kode har jeg implementert dette ved å sette en tidsterskel på 10 millisekunder mellom hver aktivering. Dette blir videre kombinert med et system som kun sender MIDI signal én gang under aktivering, i stedet for at den kontinuerlig sender signaler når knappen blir holdt. Knappene sender da MIDI beskjeder ut som tolkes i koden til den lydgenerende mikrokontrolleren. For å videre øke fleksibiliteten til instrumentet, sjekkes det også om brukeren har trykket to knapper samtidig, som da resulterer i en annen MIDI-beskjed som kan tolkes videre av Bela mikrokontrolleren.

```

1 tidsTerskel = 10
2 tidTrykket = 0
3
4 Function debounce()
5     if(currentTime()-tidTrykket > tidsTerskel ) then
6         //når aktivert
7         sendBeskjed()
8         tidTrykket = currentTime()
9     Endif
10 endfunction

```

Figur 11 – Pseudokode for implementering av et debounce system

### Visuell tilbakemelding ved hjelp av LED lys

Den siste delen av koden til mikrokontrolleren som håndterer brukerinput er å gi visuell tilbakemelding når en knapp er trykket. Tanken her var at knappene styrer hvilken modus du er i, og det vil av den grunn være veldig nyttig for brukeren å ha en visuell indikasjon på dette slik at det ikke er rom for usikkerhet under en eventuell opptreden. I koden ble dette implementert ved å lage en matrise med oversikt over alle lysene, og dersom modusen endres, vil mengden aktive lys i matrisen økes eller senkes.

### 3.3.2 Lydgenerering

MIDI-signalene sendes via USB fra ItsyBitsy M0 Express mikrokontrolleren videre til Bela systemet. Alt av lydprodusering er programmert i programmeringsspråket Csound som opprinnelig ble utviklet av Barry L. Vercoe i 1985 (Csound, u.å.). Språket gjør det enklere å utvikle egne lydsystemer ved hjelp av de innebygde funksjonene som kalles for opcodes. Bela mikrokontrolleren sin infrastruktur lar deg direkte laste opp Csound kode som kjører rett på mikrokontrolleren.

Etter å ha mottatt MIDI-beskjeder, sorteres de ved hjelp av midiin opcoden til Csound (Maldonado, 1998). Denne lar meg sortere beskjeder basert på innholdet til MIDI-beskjeden, som kanalnummer og statusnummer. På den måten kan jeg enkelt videresende notebeskjeder til diverse instrumenter basert på hvilken modus som er aktiv, i tillegg til å gi individuelle funksjoner til de forskjellige trykkbryterne og potensiometrene.

Endelig kildekode til Bela mikrokontrolleren ligger vedlagt som vedlegg nr. 2.

### De fire modusene

Csound dokumentet til instrumentet tar inn MIDI-signalene og produserer lyder basert på hvilken modus som er valgt. Den første modusen er en myk pad laget ved hjelp av additiv syntese, som deretter går gjennom en waveshaping algoritme som forvrenger lyden. Dette er oppnådd ved bruk av distort opcode utviklet av Barry L. Vercoe (Vercoe, u.å.) . Grad av forvrenging kontrolleres direkte av trykkstyrken som blir påført hver enkelt knapp.

Den andre modusen er en emulering av en tibetansk syngbolle ved hjelp av additiv syntese. Relasjonen mellom frekvensene og amplitudene til de forskjellige overtonene er hentet fra en artikkel skrevet av Guy Beider som viser til forskjellige frekvensanalyser av syngboller (Beider, 2022). Dette inkluderte opprinnelig en lavfrekvent oscillator som styrte amplituden av hver enkelt overtone for å videre øke troverdigheten, men etter testing i praksis ble dette fort for mange oscillatorer til at prosessoren kunne håndtere det. Amplituden til instrumentet styres av trykkstyrken som blir påført hver enkelt knapp.

Den tredje modusen er et melodisk perkusjonsinstrument som tar utgangspunkt i vibes opcode som igjen baserer seg på arbeidet til Perry Cook (Ffitch, u.å.). Opcode baserer seg på prinsippene bak fysisk modellering, og er av den grunn avhengig av en impulsrespons. Her har jeg valgt å bruke samme impulsrespons som brukes i dokumentasjonen til opcode, da den var lett tilgjengelig og ga ønsket resultat. Denne filen er vedlagt som vedlegg nr. 3. Videre går lyden gjennom et lavpassfilter for å forme resultatet etter egne preferanser. Amplituden til hver note settes basert på den initielle verdien fra trykksensoren, og kan ikke varieres underveis samtidig som noten holdes.

Den siste modusen er et sample-basert instrument, opprinnelig utviklet med atonale perkusjonslyder i tankene. I henhold til Bela sin dokumentasjon på bruk av Csound på deres mikrokontrollere, blir alle lydfilene lastet inn til minne når programmet starter opp i stedet for at man leser direkte fra filen hver gang et sample skal trigges (Bela, u.å.). Hver knapp er tildelt hver sin lydfile, og i likhet med modus nummer 3 settes amplituden ved notens start. Lydfilene er hentet fra en gratis samling med samples fra [angelicvibes.com](http://angelicvibes.com) og ligger vedlagt som vedlegg nr. 4 (Wolf, u.å.).

## **Digitale effekter**

Etter en av de fire modusene har produsert lyd, blir det videresendt til effektseksjonen av dokumentet. Dette starter med en enkel stereo forsinkelseeffekt som har forskjellige forsinkelsestider på hver kanal. Dette er gjort for å forsterke oppfattelsen av et bredt lydbilde. Amplituden til effekten kan endres i sanntid gjennom en av de tre potensiometrene. Forsinkelsestiden kan ikke endres i sanntid, da det produserte uønskede artefakter i lyden.

Videre i signalkjeden kommer etterklngen. Denne effekten baserer seg rundt reverbsc opcoden, som igjen baserer seg på arbeidet til Sean Costello (Varga, 2005). Ettersom denne opcoden gjør mye av arbeidet, var det ikke mye utvikling som skulle til før den var funksjonell. I likhet med forsinkelse effekten kan man styre amplituden til effekten gjennom en av de tre potensiometrene på instrumentet.

### **Looper funksjon**

For å øke fleksibiliteten til instrumentet, samt tilfredsstille ønsket om muligheten for flere lydlag som spiller samtidig, har forsøkt å legge til en enkel opptaks- og avspillingsfunksjon som emulerer en tradisjonell looper. Den er relativt rudimentær, men funksjonell. Ved å trykke inn to av knappene til høyre på instrumentet vil man starte et opptak som varer i maksimalt 30 sekunder. Om enten opptaket når 30 sekunder eller om man trykker på de to knappene igjen, vil opptaket stoppe. Deretter vil det man tok opp automatisk spilles av. Denne avspillingen repeterer seg med en gang slutten av opptaket blir nådd. De to knappene til venstre på instrumentet kan brukes til å pause og starte avspillingen på nytt etter behov.

## **4. Resultat**

Resultatet av prosjektet er et instrument som håndterer alt av brukerininput og lydprosessering på en frittstående måte. Med frittstående menes det at instrumentet i motsetning til et typisk MIDI-instrument ikke er avhengig av andre enheter for å generere lyd. Instrumentet har funksjoner som kalkulerer styrken på notene basert på hvor hardt brukeren trykker, samt mulighet for kontinuerlig endring av enkelte noter sine parametere underveis i notens spilletid. Instrumentet kan ved hjelp av trykkbrytere enkelt bytte mellom forskjellige moduser som har hver sin fremgangsmåte når det gjelder produksjon av lyd. Videre kan man velge å sende lyden gjennom en forsinkelses effekt, og/eller en etterklangseffekt. Til slutt har man også muligheten for å ta opp og spille av det man spiller underveis, noe som kan bidra til å skape et mer komplekst lydbilde med flere lag.



Jeg vil foreløpig klassifisere instrumentet som en funksjonerende prototype heller enn et ferdig produkt, da det er flere aspekter som ikke fungerer som ønsket. Bruken av 3D printer var essensiell for å utvikle tidlige prototyper innen kort tid, men da det er en prosess som kan medføre unøyaktigheter er det visse kompromisser som må inngås. Den mest fremtredende årsaken til feil ved instrumentet er trykksensor-knappene som ikke sitter fast. Grunnet 3D-printerens natur er det viktig å beregne mer plass enn nødvendig for å gjøre rom for unøyaktigheter under printing. Resultatet av dette ble at knappene fikk mer rom for bevegelse enn det som ville vært foretrukket. Dette leder til forskjellige avlesninger basert på hvor knappen befinner seg, noe som gjør det svært vanskelig å kalibrere sensitiviteten. For å motvirke dette ble jeg nødt til å gjøre knappene mindre sensitive for å unngå falske lesinger, noe som igjen gjør instrumentet mindre intuitivt å bruke enn ønsket.

3D printeren satte også begrensninger på instrumentets størrelse. Jeg skulle helst sett at instrumentet var større slik at Bela mikrokontrolleren hadde fått plass på innsiden, men det nåværende designet holdt seg så vidt innenfor den øvrige grensen på størrelse.

## **5. Konklusjon**

Med bakgrunn i et ønske om å utfordre etablerte vaner i min egen kreative praksis, har denne oppgaven utforsket prosessen bak å utvikle et eget instrument. Etter analyse av egen praksis kom jeg frem til ønsker for funksjonene instrumentet skulle inneha, noe resten av prosessen forsøkte å oppfylle. Jeg har tatt i bruk diverse komponenter som mikrokontrollere, potensiometer, trykkbrytere, og fremst av alt, egenlagde trykksensorer. Disse trykksensorene krevde mye eksperimentering før jeg fant en tilfredsstillende løsning. Denne løsningen ble likevel satt på prøve da instrumentet i sin helhet skulle konstrueres, og som et resultat av det har instrumentet fortsatt flere mangler i sin nåværende tilstand. Til tross for dens begrensninger, representerer dette prosjektet et betydelig skritt mot realiseringen av et unikt, individuelt tilpasset musikkinstrument.

## Referanseliste:

Acar, O. A., Tarakci, M., & van Knippenberg, D. (2019, 22 november). Why constraints are good for innovation. Harvard Business Review. <https://hbr.org/2019/11/why-constraints-are-good-for-innovation>

Beider, G. (31. august 2022). The pulse of a singing bowl. Bells of Bliss. <https://bellsofbliss.com/blogs/good-to-know/the-pulse-of-singing-bowls-brainwaves-and-brain-entrainment>

Bela. Csound. Hentet 19. februar 2023 fra <https://learn.bela.io/using-bela/languages/csound/#good-programming-practices-for-csound-on-bela>

Csound. About Csound. Hentet 13. mai 2023 fra <https://csound.com/>

Dzedzickis, A., Sutiny, E., Bucinskas, V., Samukaite-Bubniene, U., Jakstys, B., Ramanavicius, A., & Morkvenaite-Vilkonciene, I. (2020). Polyethylene-Carbon Composite (Velostat®) Based Tactile Sensor. Polymers, 12(12), 2905. <https://doi.org/10.3390/polym12122905>

Ffitch, J. vibes. Hentet 20. mars 2023 fra <http://www.csounds.com/manual/html/vibes.html>

Fried, L. Force Sensitive Resistor (FSR). Adafruit Industries. Hentet 10. februar 2023 fra <https://learn.adafruit.com/force-sensitive-resistor-fsr>

Gaberman, B. (31. mars 2020). How to Clean Up Noisy Sensor Data With a Moving Average Filter. Maker Pro. <https://maker.pro/arduino/tutorial/how-to-clean-up-noisy-sensor-data-with-a-moving-average-filter>

Hammer, Ø. (21. februar 1997). Fysisk modellering. Notam <https://archive.notam02.no/arkiv/kurs/DIGLYD/node22.html>

Heintz, J., McCurdy, I., & Neukom, M. (2020). Physical modelling. The Csound Manual.  
<https://flossmanual.csound.com/sound-synthesis/physical-modelling>

Kovarsky, J. (26. mars 2016). Key Feel and Response of Keyboards. Keyboard For Dummies  
<https://www.dummies.com/article/academics-the-arts/music/instruments/piano/key-feel-and-response-of-keyboards-155364/>

Maldonado, G. (1998). midiin. <http://www.csounds.com/manual/html/midiin.html>

Puckette, M. (30. desember 2006). Waveshaping. University of California San Diego.  
<http://msp.ucsd.edu/techniques/v0.11/book-html/node78.html>

Researchgate. (2018) Synthesis and analysis of sounds developed from the Bose-Einstein condensate: Theory and experimental results. Researchgate.net.  
[https://www.researchgate.net/figure/An-example-of-the-basic-principle-of-the-additive-synthesis-process-3-simple-tones\\_fig5\\_327494789](https://www.researchgate.net/figure/An-example-of-the-basic-principle-of-the-additive-synthesis-process-3-simple-tones_fig5_327494789)

Schwartz, B. (2006). More isn't always better. Harvard Business Review.  
<https://hbr.org/2006/06/more-isnt-always-better>

Smith, J.O. (2010). Additive Synthesis. Stanford University.  
[https://ccrma.stanford.edu/~jos/pasp/Additive\\_Synthesis.html](https://ccrma.stanford.edu/~jos/pasp/Additive_Synthesis.html)

SparkFun Electronics (2023). Force Sensitive Resistor 0.5". Hentet 3. Mai fra  
<https://www.sparkfun.com/products/9375>

The MIDI Association. Summary of MIDI 1.0 messages. MIDI.org. Hentet 23. april 2023 fra  
<https://www.midi.org/specifications-old/item/table-1-summary-of-midi-message>

Vandenneucker, D. (2012). MIDI Tutorial. Carnegie Mellon University.  
<https://www.cs.cmu.edu/~music/cmsip/readings/MIDI%20tutorial%20for%20programmers.html>

Varga, J. (2005). reverb. <http://www.csounds.com/manual/html/reverb.html>

Vercoe, B.V. distort. Hentet 17. mars 2023 fra  
<http://www.csounds.com/manual/html/distort.html>

Wolf, Z. Free Old School Sample Pack. AngelicVibes. Hentet 10. mai 2023 fra  
<https://www.angelicvibes.com/free-old-school-sample-pack/>

## **Vedlegg:**

- Vedlegg nr. 1:  
*1\_MIDI\_Kildekode.ino* - Kildekode i Arduino sitt .ino filformat
  
- Vedlegg nr. 2:  
*2\_Csound\_Kildekode.csd* - Kildekode i Csound sitt. csd filformat
  
- Vedlegg nr. 3:  
*3\_marmstk.wav* - Impulsrespons-fil brukt i forbindelse med vibes opcoden til Csound
  
- Vedlegg nr. 4:  
*4\_Samples.zip* - En komprimert mappe med lydfilene brukt i sampleinsturmentet

