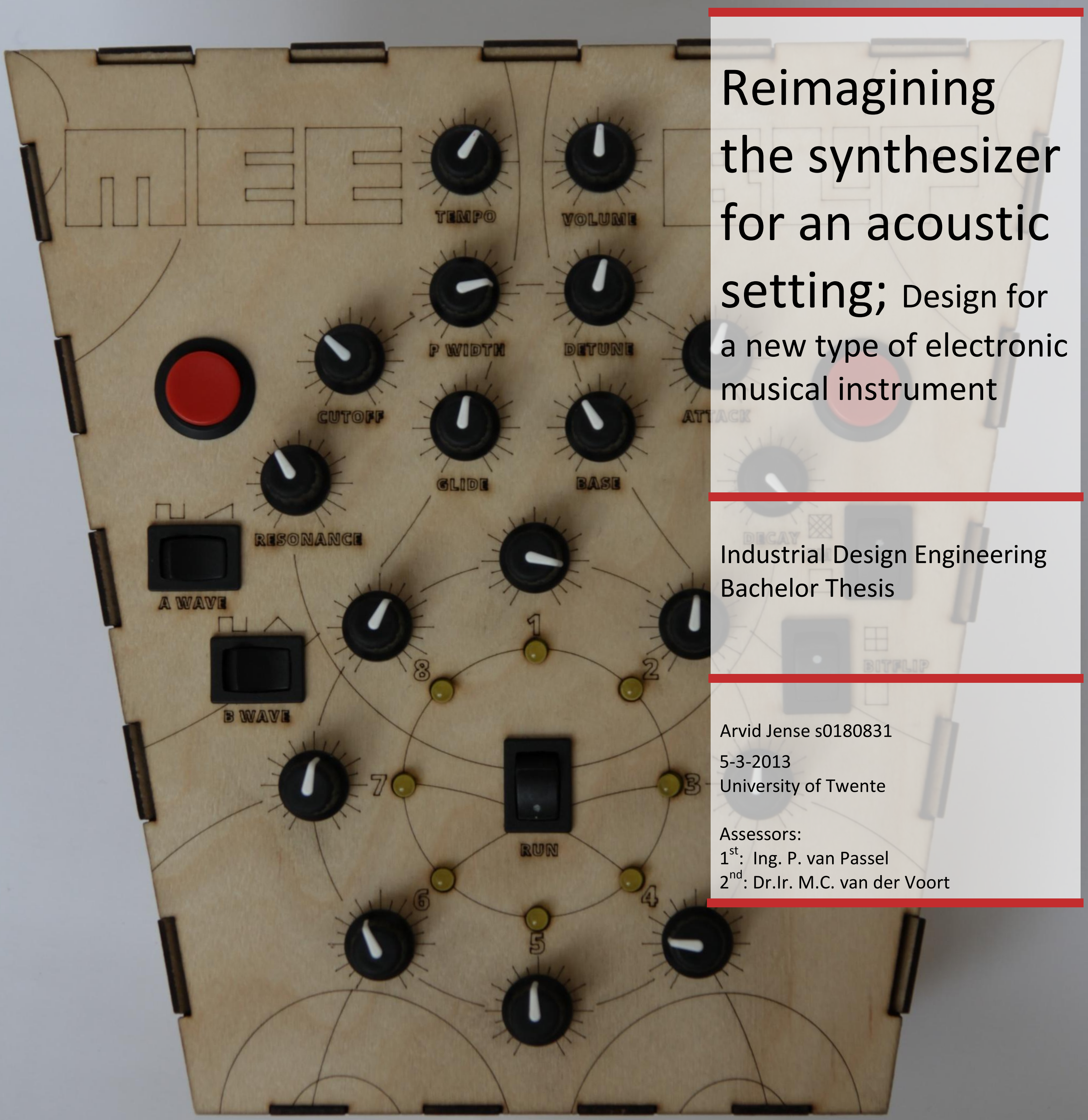


Reimagining the synthesizer for an acoustic setting; Design for a new type of electronic musical instrument

Industrial Design Engineering
Bachelor Thesis

Arvid Jense s0180831
5-3-2013
University of Twente

Assessors:
1st: Ing. P. van Passel
2nd: Dr.Ir. M.C. van der Voort

Reimagining the synthesizer for an acoustic setting;

Design for a new type of electronic musical instrument

By:

Arvid Jense

S0180831

Industrial Design Engineering
University of Twente

Presentation: 5-3-2013

Contact:

Create Digital Media/Meeblip
Betahaus, to Peter Kirn
Prinzessinnenstrasse 19-20
BERLIN, 10969
Germany

www.creadigitalmusic.com
www.meeblip.com

Examination committee:

Dr.Ir. M.C. van der Voort

Ing. P. van Passel

Peter Kirn

ABSTRACT

A design of a new type of electronic instrument was made to allow usage in a setting previously not suitable for these types of instrument. Following an open-ended design brief, an analysis of the Meeblip market, synthesizer design literature and three case-studies, a new usage scenario was chosen. The scenario describes a situation of spontaneous music creation at an outside location. A rapidly iterating design process produced a wooden, semi-computational operated synthesizer which has an integrated power supply, amplifier and speaker. Care was taken to allow for rich and musical interaction as well as making the sound quality of the instrument on a similar level as acoustic instruments.

Keywords: Synthesizer design, Electronics design, Open source, Arduino, Meeblip, Interaction design

SAMENVATTING (DUTCH)

Een ontwerp is gemaakt voor een nieuw type elektronisch muziek instrument welke gebruikt kan worden in een setting die eerder niet geschikt was voor elektronische instrumenten. Een vrij open ontwerp opdracht is via een markt-, product- en literatuuranalyse gedefinieerd in een scenario en programma van eisen. Het scenario beschrijft een situatie van spontane muziekcreatie op een openlucht locatie. Een snel itererend ontwerp proces leidde tot het ontwerp van een houten, computergestuurde synthesizer met geïntegreerde stroomvoorziening, versterker en luidspreker. Er is bijzondere aandacht besteed aan zowel het maken van een rijke muzikale interactie als wel de geluidskwaliteit op een vergelijkbaar niveau te krijgen als akoestische instrumenten.

Kernwoorden: Synthesizer ontwerp, Elektronica ontwerp, Open source, Arduino, Meeblip, Interactie ontwerp

PREFACE

Building things seems to be something in my nature, be it with Lego bricks, with wood or with music, something had to be constructed, no need for instruction manuals. The choice for Industrial Design Engineering was therefore a very natural one. During my studies, I discovered my love for electronic musical instruments. To get to more into this subject, I combined my studies with an electronic music production program. A few years later, with the eminent completing of the Bachelors program, I wanted to find a project embodying both design and music, as a demonstration of what I've learned.

After a short period of searching for companies in the musical instruments industry, I found Meeblip; a company producing a great open source synthesizer. After some mails and effort in finding the right form, I was off to Berlin for an eight week internship! During these weeks, I have finally made my first complete design, including a working prototype. Next to the project described in this document, the internship activities included support on development on the actual next generation of Meeblip synthesizers, writing articles on instrument design, like an article on the synth cube Kinektron, and helping organize an event for people using and creating novel musical instruments; MusicMakers Berlin.

Please be advised that some of the language is very technical, sometimes in design theory, sometimes in music theory and other times in electronics. The reader of this document is assumed to have enough knowledge in these fields. For the user that does not have this knowledge, simplified information will be published on CreateDigitalMusic.com early 2013.

Many thanks go out to; Peter Kirn and James Grahame of Meeblip for creating this internship and design project; to the people of my workspace Betahaus for having such an awesome workshop and creative environment; to Arie Paul van den Beukel and Pepijn van Passel of the University of Twente for the support on the industrial design side of things; to Peter Spitters for his thorough proofreading; and finally to Sophie Spitters for all her support!

CONTENTS

1. Introduction

To give a demonstration of the possibilities of the open-source synthesizer[1] Meeblip, a design is proposed for a novel type of electronic instrument meant for use in a usage scenario previously not suitable for synthesizers. Particular care is taken to design for the user and interaction. Through an extensive analysis, a concept is formed, of which a proof-of-concept is given as a fully-functional prototype. The final design shows the versatility of the Meeblip as well as a new direction for synthesizer design.

Figure 1, the design process starts by analyzing the Meeblip synthesizer both in functionality and market. To comply with the philosophy of Meeblip, an analysis is made on *Open Source Hardware* and how this impacts the design process. As to *stand on the shoulders of* giants, a literary review is done on electronic instrument design. An analysis is made of potential new users and usage scenarios, of which one will be chosen. The chosen user group is analyzed together with competing products fitting the scenario.

The functional and user needs coming from the analysis lead to a product design specification (PDS) functioning as input for design proposals for user interaction as well as appearance and interface. One of the proposals is chosen by Meeblip to develop further into a proof-of-concept. This proof-of-concept is created in the form of a fully-functional prototype and which demonstrates the functionality of the interface and the appearance of the new product, as well as being fit for user evaluation. As seen in Table 1, the design process is completed within 14 weeks.

Deliverables are defined as:

- Product design proposal, documented by renders and technical drawings as well as text;
- Proof-of-concept as a tangible musical instrument;
- Report with text and images, according to university guidelines.

This design project embodies a few different design approaches, but the main philosophy is user-centered design, but it also incorporates ideas from practice-based design (Johnston, 2011). A detailed description of the working method can be found in APPENDIX F. Prototypes were made in the workshop of Betahaus, Berlin.

[1] A musical instruments which generates sound through electronic principles

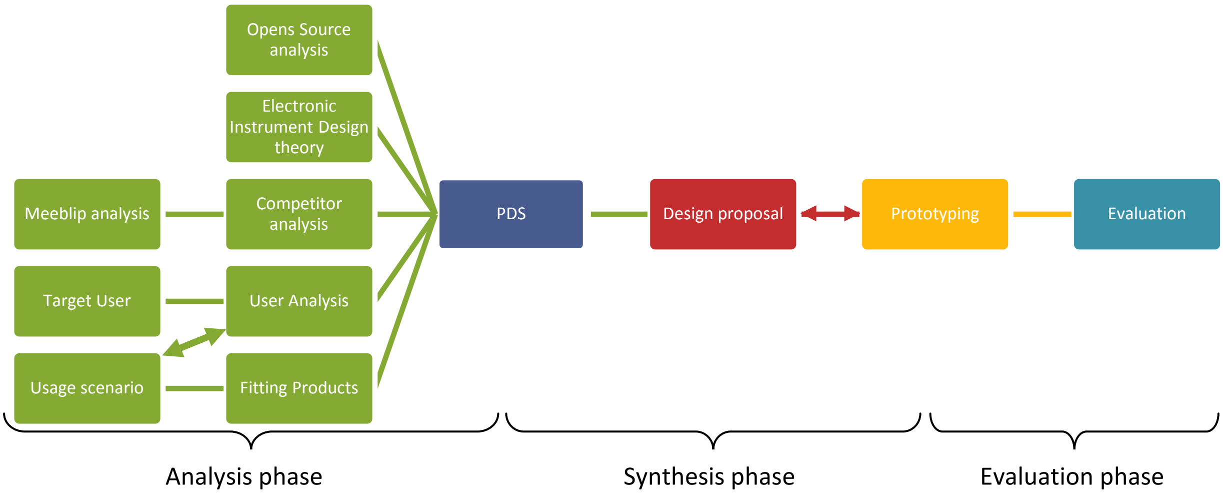


Figure 1. Design Process

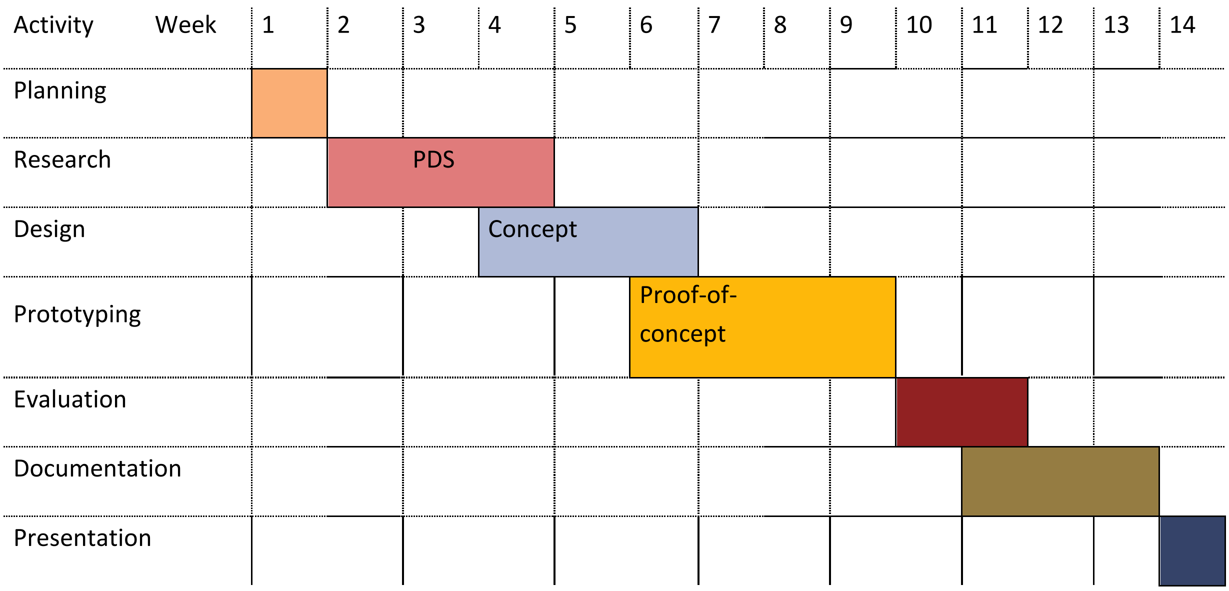


Table 1. Planning

2. DESIGN BRIEF

The purpose of this design project, as given by Meeblip, is to create a new variation of an electronic musical instrument that is based on the existing Meeblip-technology and to develop the interface (in terms of user interaction and appearance) of this new variation. This new to variation is intended to appeal to a different target audience than the existing Meeblip, while remaining interesting for the current user group.

While the Meeblip is a fully functional product receiving positive reviews, the company wants to keep developing its product to expand its market. The company would like to do this by using the existing Meeblip technology to create a new musical instrument with a different kind of interaction than the original Meeblip. This should make the product interesting for a wider range users as well as making the product more versatile to current users. While currently the Meeblip is dependent on a MIDI controller[2] or laptop for performing music, the company has stated the idea of integrating a *sequencer* with the Meeblip turn it into an independent instrument, although the implementation for this project can be defined in any way possible.

Depending on the outcome, the design can be considered for production. However making most of the little time available in this internship and allowing for maximum creative freedom, considerations on manufacturing, industry standards, legislative and marketing will not be dealt with. As to make the design fit in the current scope of Meeblip activities, a step-by-step instruction should be made, to allow current and new users to make the designed instrument themselves. This is done in the form of an *Instructable.*

2.1. OBJECTIVE

The key objective for Meeblip is to have a demonstration of relevant product variation with their 'hackable' Meeblip Micro. Serving as a marketing tool showing potential customers that the possibilities of using a Meeblip are bigger than what they now seem.

As secondary objective is for their products to have a big impact, such as key competing products like Korg MicroKORG, Korg Monotron or Lomography. It is advised to make a case study out of these products to find key competitive aspects in features and design.

2.2. TIME/SCHEDULE

There are eight weeks available in Berlin in which the research, design and prototyping should be done. Most important parts of the research project for the company are the functional prototype and documentation.

[2] An external keyboard or button/knob interface, which can control a synthesizer

3. Analysis

With the extensive possibilities the design brief allows, it would be foolish to jump headfirst into the first idea that pops into the mind. So an analysis has been made on Meeblip, definition and philosophy of open source, synthesizer design and several case studies on specific competing products. The key findings are used in the remaining chapters on concept, design and prototyping.

3.1. Meeblip

To get a good feeling of what the Meeblip actually is, a look had been taken at the product itself, the company and lastly the market including current and potential users and competitors.

3.1.1. The object

> *"MeeBlip is a hackable, affordable digital synthesizer, made for accessible sound and hands-on control. It can be someone's first synth. It can be a unique-sounding addition to your music setup, playable with MIDI hardware and software. It can be a synth you open up and modify, learning about sound creation, code, and electronics. Or it can be the basis of new projects and ideas."*[3]

Some technical features are found in Table 2. As these terms are quite technical, they will be briefly explained in chapter 3.1.2.

Monophonic, Two oscillators
LFO
ADSR
2-pole Resonant digital Low or High Pass Filter
FM
Distortion
36363Hz sampling rate
8 slot preset bank

Table 2. Meeblip features

The Meeblip is available in three different packages and is currently in its second design iteration. The first package is fully assembled which works out of the box as seen in Figure 4. The second is a kit, which includes all the parts, but needs the user to solder and assemble to form the same product as the fully assembled package. The third type of package is the Meeblip micro, seen in Figure 2, which is electronically the same as the other packages, but without the knobs and switches, and reduced in size to about twice a credit card, making it suitable for doing custom synthesizer projects.

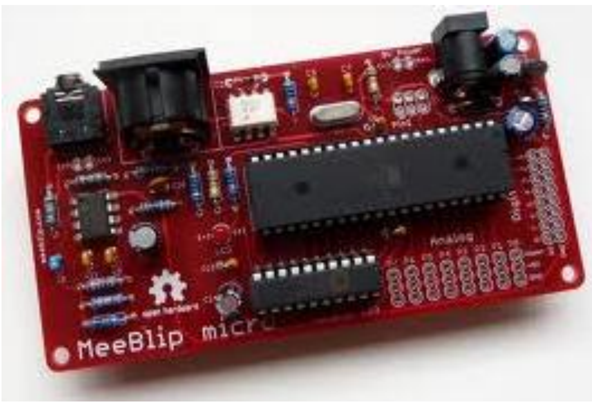


Figure 2. Meeblip micro

[3] http://www.meeblip.com

The Meeblip is hackable, this means that the company actively enables the users to modify the product to their needs. This is mostly done by providing all the design documentation under an open-source license. The most hackable part of the Meeblip is modifying the firmware. The sound-generating code and interface functionality can be adjusted by using a programmer and assembly code. There is not much hacking on the electronic components. Although the circuit board uses through-hole components which are more suitable for modifications than than the more compact surface-mount components. Figure 3 shows some of hardware modifications made by users.

FIGURE 3. USER MODIFICATIONS OF THE MEEBLIP

The aesthetics of the Meeblip are for the most part defined by the prefab encasing[4], the knobs, the interface layout and the interface graphics. There have been two version of the Meeblip, both using the same plastic encasing. The first version, shown to the left in Figure 4, has an illustration on with a cute, obscure hand drawn vibe, while the second version, shown to the right in Figure 4, has a cleaner, minimal feeling.

Next to a change in aesthetics between the two Meeblips, the second version added the possibility of storing presets, making it easier to revisit previously made sounds. For this feature, three buttons had to be added, seen at the top right of Figure 4. Other changes include improvements in the sound generation code.

FIGURE 4. THE TWO MEEBLIP ITERATIONS

---

[4] http://www.serpac.com/userprints/07S.pdf

An informal evaluation of the device found that it is a fun instrument to play with, that the knobs feel good, though a bit loose, the knobs are slightly too close to each other. Some knobs don't directly produce the desired effect, but the sonic palette is very broad and unique compared to other synthesizers.

Features that make the Meeblip unique and that should be taken in consideration for this design project are; the minimalistic, but cute aesthetic; the directness of interaction; the hackability; the low cost.

More information, such as how to order one, can be found on the Meeblip website[5].

3.1.2. Side note on synthesizer terminology

The Meeblip is described as a *digital-two-oscillator-subtractive synthesizer with a flexible filter*. To understand what this means, a bit of synthesizer theory is needed. All aforementioned terms will be broken down as to provide insight in what these mean.

A *synthesizer* is a type of electronic musical instrument that artificially generates sound and as such performs sound synthesis. Often, synthesizers do not produce the sound directly, but rely on an external amplifier and speaker.

An *oscillator* is something which vibrates. In the context of synthesizers, it is an electronic circuit which generates a waveform with a controllable frequency within the human hearing spectrum. The shape of the waveform it produces differs with different types of oscillators. Most common are simple square waves or sine waves, while others go as far as storing recorded waveforms from acoustical instruments. The most common four types of waveform are shown in Figure 5.

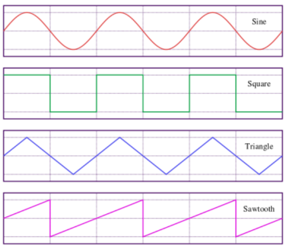


Figure 5. Four common waveforms

A *filter* is something that lets some things through, but blocks others. In electronic musical instruments a filter works by cutting of certain parts of the frequency spectrum. This causes the timbre of the sound to change depending on what type of filter is used. Typically if a sound is too harsh, a *high-cut* filter is used to cut off some of the higher frequencies, while if you would like a less full sound, a *low-cut* filter is used to remove some of the lower frequencies. More advanced filters will feature extra possibilities like resonance; boosting the frequency just before the point where you start to cut off other frequencies.

Subtractive synthesis is a type of sound synthesis that generates tones in two steps, at first it generates a wide spectrum tone from an oscillator, which will pass through a filter to alter its timbre.

In the world of synthesizers, there is a great distinction between *digital* and analog devices. It is generally felt that analog is better, but digital is more in use because it is both more flexible and affordable.

So what a *digital two oscillator subtractive synthesizer with a flexible filter* means, is that the Meeblip is a synthesizer, using two oscillators going through a filter with many controllable advanced features, to produce sound using the subtractive synthesis method, all of this done in a digital environment.

---

[5] http://meeblip.com/what-is-meeblip/

3.1.3. The company

The Meeblip is a joint project by Reflex Audio Systems (Reflex) and Createdigitalmusic.com (CDM). Reflex Audio Systems is a Canada based company, run by James Grahame. Reflex's previous endeavors include a versatile MIDI controller and a joystick controller. CDM is a Germany based company, running a blog on innovative music and visual technology. The project originated from a shared desire to make a simple and fun open-source synthesizer. Reflex handles most of the manufacturing and development, while CDM manages marketing and documentation.

The niche-market product demand is big enough according to Meeblip; CDM is currently keeping the marketing activities down as they can't keep up with demand. For a next series, the numbers are increased and distribution will be handled through Amazon.com[6] for distribution in the United States and a separate distributor is being searched for to serve the European market. Getting certified for mainstream distribution is expensive, so for the first two generations the Meeblip distribution was kept in-house.

3.1.4. The market

As detailed market analysis or economic analysis were not available on the topic of electronic instruments, so an informal analysis has been done on both the user base of the Meeblip as well as the direct competitors.

3.1.4.1. Users

An analysis has been made of 50 online user-made videos featuring the Meeblip as well as of various forum and blog posts found through Google. The videos could be categorized into four types:

- Demos: a recorded exploration of what sounds a Meeblip can sound like.
- Music: a piece of music in which the Meeblip is specifically featured.
- Unboxing/build: how a user received and constructed its Meeblip.
- Modifications: demonstrates how someone modified or hacked their Meeblip.

The forum and blog posts could be found in four categories:

- Copies of the press release
- Reviews
- Expressions of intend to buy
- Technical problems

Following the forum posts, it was found that there is a small community of about 15 active users whom participate on the Meeblip Facebook group as well as the official Meeblip forums. The most active topics they participate in have been on the modifications they make, see Figure 3. These users all seem to be both avid synthesizer users as well as being electronically adept. It is interesting to note that these have only done modifications on the casing and interface of the Meeblip, but never really adjusted the sound-generating code for tweaks or new features.

From this information a general conclusion could be made of what types of users have purchased a Meeblip as seen in Table 3. No conclusions could be made on the size of each group.

Type	Description	Unique selling point of Meeblip
Hacker	Hackers, these want to make their own synthesizer and see the Meeblip as a good starting point for that. Some work it into an entirely new instrument, while others just make a variation on the casing. Interested in versatility.	Opensource/ hackable/ moddable/ kit

[6] A big American webshop

Type	Description	Unique selling point of Meeblip
Starter	Beginning electronic musicians, these want the Meeblip as a cheap but good sounding and MIDI enabled hardware synthesizer. They are generally used to software-synthesizers, and treat the Meeblip as such, but like to use the physical controls.	Low price
Synth pro	Electronic musicians using a hardware setup, these use the Meeblip as a good value general synth in addition to their other gear. They want something unique to add to their collection/setup.	Interesting sound

TABLE 3. MEEBLIP USERS

3.1.4.2. COMPETITORS

To get an impression of what market the Meeblip is in, an analysis has been made of competitive products. Table 4 shows the products that are often mentioned in conjunction with the Meeblip or who share similar features or a similar philosophy. Aspects described are:

- Availability, where *website* means that the product is only available directly from the website of the developer and common means that it is widely distributed.
- Voices are the number note that can be played simultaneously.
- A/D indicates whether the synthesizer is Analog or Digital.
- MIDI describes whether or not the synthesizer can be controlled through the MIDI protocol.
- Prices are the most commonly found price, averaged to the nearest round number. *Kit* means that the user has to perform some assembly himself.

Unless otherwise noted, all of these synthesizers are modules, meaning they need a keyboard or other external input mechanism for controlling them.

Name	Description	Availability	Voices	A/D	MIDI	Price
Meeblip SE	Affordable open source subtractive monosynth.	Website	1	D	Yes	€150
Meeblip Micro	Barebones open source subtractive monosynth for hackers.	Website	1	D	Yes	€50
Atari Punk Console	Described as the simplest synthesizer possible, available in DIY kits.	Website, (Common)	1	D	No	€30 (Kit)
Chamber of Sounds	Several simple but fun sound generating devices.	Website	1	AD	No	>€250
Dave smith Mopho	Compact and affordable synthesizer.	Common	1	A	Yes	€650
Hackmeopen rockit	Open source hybrid synthesizer.	Website	1	AD	Yes	€200 (Kit)
Korg Monotron	A small synthesizer controllable by a ribbon. Made easily hackable due to Korg releasing the schematics.	Common	1	A	No	€40
MFB-Nanozwerg	A simple synthesizer, similar in concept and features to the Meeblip, but analog an closed-source.	Website	1	A	Yes	€200
MIDIBox SammichFM	Powerful Fm synthesizer sold as a DIY kit.	Website	4x6	D	Yes	€150 (Kit)
MIDIBox SammichSID	Similar to the sammichFM, but using a vintage SID chip, rather than FM.	Website	3	D	Yes	€150 (Kit)
Mutable Instruments Shruti-1	Is similar in concept to the Meeblip yet offers more features and an analog filter. Has a less direct interface.	Website	1	DA	Yes	€140 (Kit)
Nebulophone	An Arduino based synthesizer	Website	1	D	No	€50 (Kit)

Name	Description	Availability	Voices	A/D	MIDI	Price
Preenfm	An open source DIY kit for a FM synthesizer based on the Maple prototyping platform.	Discontinued	4	D	Yes	>€60 (Kit)
Sound Lab Mini-Synth	A simple, DIY, open source analog synthesizer.	Website	1	A	Yes	€200 (Kit)
WTPA	A circuit- bendable sampler kit	Website	1	D	No	€75 (Kit)
YM_MINI Synthesizer	A sound chip sound in Atari and Apple computers made controllable through MIDI.	Website	3	D	Yes	€80
Second hand commercial synthesizers	A huge range of low cost synthesizers can be found on flea markets.	Flea markets				>€20

TABLE 4. COMPETITORS

This analysis can tell us a few things. Firstly, while there are at least 15 competitors, when considering that lower priced synthesizers are either sold as a kit only or lack MIDI connectivity, the Meeblip sits at the bottom level of price. Secondly, even though there seems to be a definite market for these products, barely any is distributed through common music shops. Discussing this with Meeblip company, revealed that this is mostly a problem caused by resources; it is one thing to design and build a small series of instruments yourself, however it's a whole other game to switch bigger series. This is supported by the fact that most synthesizers are sold as kits, thereby circumventing electronic-device legislation. Thirdly, a small categorization of competitors can be made; Analog synthesizers, programmable microcontroller synthesizers and vintage chips synthesizers. Meeblip sits in the second category of programmable microcontroller synthesizers. Second-hand synthesizers are definitely seen as an alternative for Meeblip costumers, but due to the enormous range of devices and unpredictable availability, it is hard to make any conclusions on this segment.

Meeblip developer James of Reflex Audio System sees the Shruti-1 as the main Meeblip competitor and notes four main differences:

> *"[1] the MeeBlip has a panel full of knobs and switches instead of a paged OS with an LCD. Both approaches have pros and cons.*
> *[2] MeeBlip is completely open source hardware and software (the Shruti-1 hardware has non-commercial restrictions)*
> *[3] MeeBlip has a digital filter, whereas Shruti-1 allows you to pick and choose various cool analog filters [note that the Shruti's digital output is 8-bit PWM, whereas the MeeBlip is 16-bit].*
> *[4] MeeBlip's firmware is written in AVR assembly code. Shruti-1 is written in C. A number of people have complained that assembler is much harder than C, but the cold, hard truth is that a full-featured synth program is complicated and confusing in any language."*

3.2. OPEN SOURCE

As Meeblip has all their products licensed as open source, this is required for this design project as well. The aspects concerning it will be taken in consideration. Both the moral and business side of things open source are discussed, to end up with guidelines for the design process.

3.2.1. DEFINITION

Open Source is a philosophy of licensing a product, but releasing all or most of the design information. This started from software developers who wanted to share their creations, but wanted to find a middle ground between using a strict standard license, which forbids nearly any use, and making their creation public domain, which would allow the use without any credit. Through this, many collaborative projects emerged without the need for a managed group of developers.

One of the most notable licenses is Creative Commons. This license allows the maker to choose whether further use is allowed with *contribution*, m*odification*, *commercial use* or any combination of these three. Most of the licenses are geared towards digital or written material. To apply this to hardware, a group of electronic hardware developers came together to form an open source license for hardware, forming an Open Hardware definition in 2012.

Open Hardware is short for Open Source Hardware, which is a design principle stated in the "Open Source Hardware (OSHW) Statement of Principles 1.0" and the twelve criteria of the "Open Source Hardware (OSHW) Definition 1.0[7]". It is still in development, but at least 270 persons and companies already endorse this definition. These are mainly small companies dealing with specialized technology in an already open source world. The biggest Open Hardware company is Arduino, which makes the interfacing platform used in this design project. It is important to note that there aren't any mainstream distributed consumer products under the license yet.

It's a lot harder to do Open-Source Hardware then it is to do Open Source Software. Because a lot of the individual components deal with legislation such as patents, making it hard to define which part of the product is under which license. For instance, while the Meeblip is Open Hardware, the Atmel microprocessor which functions as the brain, is not, yet the logic running inside the microprocessor is Open Source again. So while you can license this physical product as Open Hardware, some parts are not. As a result, the general consensus is that all the documentation distributed by a company on an Open Source license is open, while everything else is not.

3.2.2. Considerations

An economist might see a lot of downsides to the Open Source model. For instance it is very easy for competing companies to reverse engineer you products and release it for a lower price, as all needed information is readily available. There is extra effort needed in making the documentation comprehensible for the users, not to mention the extra support need when a user does not understand something. This is countered by the "as-is" clause, but users will contact you with questions if they have than anyhow.

The reasons for using an open license are not only ideological, but in many cases it might even be profitable for a company. For instance, it might have a bigger appeal to the group of pro-Open Source users. Or it might help potential customers get a better view of what they're buying, similar to giving out free samples. Jeppesen & Frederiksen (2006) note that the open source model allows users to contribute to product innovation, which is beneficial on two levels. Firstly the user community can share creations and new features, which will increase the recognition between users and makes them enjoy the product on a higher level. Secondly the contributing users will function as external developers as well as user testers, effectively crowd sourcing development.

It should be noted that there are a lot of companies who do not use an Open Source license, but reap similar positive effects. For example Native Instruments Reaktor[8], a synthesizer programming environment, is not Open Source itself, but the company successfully encourages the sharing of content by users, with many creations shared under an Open Source license themselves.

3.2.3. Consequences

While Open Source is a choice on company level rather than design level, it is important to keep it in mind during the design process. The real implications only follow in the documentation process. First it is important that a license should be chosen to be distributed with the design. This license should be referenced in any of

[7] http://freedomdefined.org/OSHW
[8] http://www.native-instruments.com/en/products/producer/reaktor-5/

the documentation. While this isn't always the case, it is desirable that any software and hardware used for the design is Open Source itself, so that the distributed files are useable without the need for buying proprietary software. All documentation and files needed to reproduce the design should be easily available to the user. Lastly, the final product may be labeled with the Open Hardware logo, to identify it as such. Note that this would not have any implication without having other documentation openly available.

As a result; this document and all included research and illustrations, are covered by a Creative Commons Attribution ShareAlike license[9] and the hardware designs, schematics, and code are provided under the GPL v3[10], with their logos shown in Figure 6.

FIGURE 6. LOGOS FOR THE CREATIVE COMMONS AND GPL LICENSE, RESPECTIVELY

3.3. ELECTRONIC INSTRUMENT DESIGN

While a few definite form factors, as seen in Figure 7, have emerged for electronic instrument, it is not to say these are the only possibilities for their design. Some might even argue that these are responsible for a lot of the criticism on electronic music as some of the repetitiveness and lack of dynamics can be led back to the feature, or lack thereof in the instruments used for that music. It might be good to take a step back and realize we are designing *something-in-order-to-perform-music* (Armstrong, 2006).

In the following sub-chapters, literature on electronic musical instrument design will be discussed. A lot of this information stems from the yearly conference on New Interfaces for Musical Expression (NIME); a conference searching for ways to increase the level of interaction between musicians and their electronic instruments. This provides a theoretical and practical information of how to make a good electronic musical instruments.

FIGURE 7. THREE COMMON FORM FACTORS FOR SYNTHESIZERS (KEYBOARD, RACK-UNIT, TABLETOP)

3.3.1. REACTIVENESS

As Marshall (2008) and Armstrong (2006) note, electronic (and specifically digital) instruments are in a way disconnected from the user. An important cause of this disconnection is the lack of an inherent coupling between the method of sound generation and the physical interface. Where an acoustic instrument provides direct physical feedback, an electronic instrument does not. This makes the connection between an electronic instrument and the performer less direct. The model of an electronic instrument as proposed by Marshall (2008, p. 26) describes a musical instrument having three main components:

- *The physical interface*: The knobs, display and physical body.
- *The software synthesis system*: What contains the sound generation method
- *The mapping system*: What connects the two

[9] http://creativecommons.org/licenses/by-sa/3.0/
[10] http://www.gnu.org/licenses/gpl.html

The model calls for a separate feedback generating and actuator system. As this adds a lot of complexity, an alternative can be found by having the object vibrate itself through an inbuilt speaker. Effectively the speaker works as an actuator but it is not controlled as a separate entity. As a bonus this allows the user to have influence on the sound by touching or moving the device.

3.3.2. Usability and performability

Research tells us that while synthesizers are incredibly powerful and versatile machines, their usability is generally very poor (Saego, Holland, & Mulholland, 2004), causing most users to stick to the factory programmed presets and barely exploring the actual capabilities.

In an acoustic instrument, every single part of the instrument has an influence on the sound. As you cannot imagine a guitar and a flute being the same general shape, the shape is heavily dependent on the sound generation method. This is not the case with electronic instruments. Frequency modulation and subtractive synthesis are structurally different sound generation methods, but both could be fitted in the same type of microcontroller.

Acoustic instruments give physical resistance to the user, often requiring specific motor control skills to even get a steady tone. A synthesizer being not much more than a bunch of microchips and other electronic components, there is no direct way for a human to influence it except for using the physical interface, which in turn uses a mapping system to influence the actual generation of sound. This means the form factor is next to free restrictions in from. While this gives an electronic instrument designer a lot of freedom, it also burdens the designer with figuring out a way to most effectively control the electronics.

As been noted, most electronic instruments suffer from a disconnection to their users. This leads to a lack of fluency in interaction. Performers of electronic music are often not directly involved in the music that is created, only keeping an eye on their instruments to diagnose errors. It may be said that electronic music is a new paradigm of music and the inactivity of the performer is only a problem when comparing to the old paradigm of music performance. But the fact is, a lot of performers themselves are looking for involving ways to perform their music. This can be seen by the enormous popularity of using MIDI/dj controllers instead of just using a laptop, while a laptop is sonically capable of performing all tasks needed, but lacking a *tangible* interface(Ishii & Ullmer, 1997). The issue might not be in either the performer or the audience, but the instruments have unperformative affordances. Just like a single cymbal is less likely to give an interesting performance than a complete drum kit.

To give electronic instruments the right affordances for music performance, focus should be to creating enactive devices, rather than just technically more adept instruments (Armstrong, 2006). What should be aimed for could be called flow; the musician should have his entire consciousness focused on the act of performing music. The instrument should provide resistance against the user's actions and in practicing this reaction; the user develops a relationship with the instrument.

Armstrong (2006) give five key criteria for an instrument to have the right affordances to become a performative or *enactive* musical instrument:

- *Situated*: The user should be able to adapt to changes in the environment, without being fully aware of what to expect from that environment;
- *Timely*: As music is based on time constraints, the user should be able to react in time to sync up with other events;
- *Multimodal*: The user should be able to use multiple senses in unison (ie. Not having to take his attention from the rhythm to be able to program the timbre);
- *Engaging*: The user should be required to do things to keep the music going and as such captures a large portion of the attention;

- *Sense of embodiment*: The user should feel connected to the device, as being an extension to his or her body.

3.3.3. Modes of musical interaction

Complementing the previously mentioned criteria, we can use the three modes of interaction with electronic musical instruments by Johnston (2011): instrumental, ornamental and conversational. These can be used as tools to measure the level of interactivity with the instrument. These modes are stated by Johnston as follows:

> *"In instrumental mode the musician seeks a high level of detailed control over all aspects of the virtual instrument's behavior. Musicians taking an instrumental approach essentially see the virtual instrument as an extension of their acoustic instrument and want it to respond consistently so that they can trust it during performances."*

> *"In ornamental mode, musicians surrender detailed control of the generated sound and visuals and let the virtual instrument create audio-visual layers that are added to their acoustic sounds. Musicians taking an ornamental approach may not pay active attention to the behavior of the virtual instrument, instead leaving it to its own devices and expecting (or hoping) that it will do something that complements or augments their sound without requiring directed manipulation."*

> *"Conversational interaction occurs when musician approaches the virtual instrument as a musical partner. In conversational interaction the musician allows the virtual instrument to `talk back', at times directly influencing the overall direction of the music. The musical `balance of power' is in flux as responsibility for shaping musical direction continually shifts between musician and virtual instrument."*

3.3.4. Early electronic instruments

The efforts in early electronic musical instrument design clearly show an effort for making a performative interaction. Instruments like the Theremin (1928) and the electronic sackbut (1948)[11] allow for rich modes of interaction, as well as having a substantial learning curve. The interface of the electronic sackbut, shown in Figure 8, show that direct control of synthesis was one of the most important premises. In comparison, modern electronic instruments seem to focus on improved features and fine-tuned synthesis techniques. It seems as if most devices take the form factor for granted, while early examples usually have completely different forms. This suggests that inspiration should not only be taken from products in the market right now, but rather from instruments made in the more experimental stages of electronic instruments.

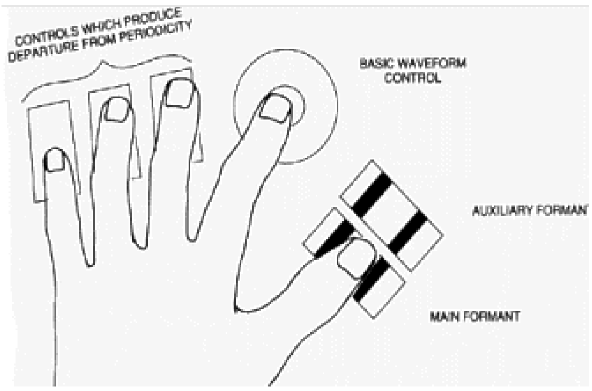


Figure 8. Timbre control of the electronic sackbut

[11] http://www.youtube.com/watch?v=69B82HrWZZU

3.3.5. Guidelines

Perry Cook (2001)(2009), a NIME regular, has been studying the design processes of people creating novel electronic musical instruments. From this research he found a lot of common pitfalls. To counter these, he has written a list of principles for designing computer music controllers. A few guidelines very useful for this design project are:

1. *Programmability is a curse*: While electronics and microprocessors allow for eternal customization, it is important to condense the devices to a single form.
3. *Copying an instrument is dumb, leveraging expert technique is smart*: Don't just replicate an instrument because it works, but try to analyze why this works and use those features.
8. *Make a piece, not an instrument or controller*: An instrument is of not much use if it is not useable in a musical context
9. *Instant music, subtlety later*: To have user making music is more important to give users access to specific features.
11. *Batteries, Die (a command, not an observation)*: Though the original principles state for no batteries to be used at all, modern batteries carry enough power to be useable. Care should be taken that performance should not hinder when they do die.
23. *Wire and document for future surgeries*: As the prototyping process was quite difficult, it was very useful to know what everything was, the next time the object opened.

3.3.6. Interface considerations

Hazel (1992) gives some key points of interest on the general design of synthesizers. While these are geared towards synthesizers in the previously mentioned form factors, there is no reason to think they wouldn't be applicable to the design of electronic musical instruments in other forms.

The instrument should function with a logical structure. Often, users get stuck at places where a developer added a feature, without this feature fitting in the general flow of the instrument. To avoid this, a synthesizer should always be functioning as a single unity.

This unity should be complemented by an easy to follow interface. The interface should help the user find the right way of operating the instrument. This is most objectively done by reducing the number of steps needed to achieve a certain goal.

Both the interface and the actual functioning should be consistent. Every time you perform a certain action, the same result should be heard. If a function has multiple functions, it should be clear which function its performing at the moment, and a single function should not move between different knobs. A consistent interface is more easy to learn because learning all inconsistencies in an interface or program costs a lot of effort. A consistent device will inspire confidence in the user, because he will be able to predict the reaction from his action.

Text should be understandable; while *four-pole* might be the type of filter, just writing *filter* next to the knob adjusting it, will make it a lot easier to understand what will happen if you turn the knob. In other words, commonly known words make cognitive processes easier. To further help with readability, Hazel (1992) suggests to use a sans-serif font, keeping enough space between elements. Text can be replaced or complemented by symbols to increase the total legibility, as symbols are often processed quicker than words.

Feedback is an area to take specific care in. Most feedback given by the synthesizer is visual, but these signals are often less effective in performance situation, where there might be little light on the instrument itself or flashing lights make readability harder.

Timing should work within the humanly noticeable range, of which Hazel (1992) gives 50ms as a maximum period between action and reaction. If nothing is heard or seen within that range, it will often distract and break the flow of the user.

3.3.7. Inspiration and iteration

While the design presented in this study is an inspirations driven design, it is likely to be the first iteration in a longer design process. Vallis, Hochenbaum, & Kapur (2010) describe this process with the Monome, a matrix of buttons in a wooden encasing, working as a versatile MIDI controller. The same article teaches some lessons on the open source model as well.

While the Monome has shown how an open customizable interface provides for “many different people” and can be “modified by users for specific needs”, the group of people actually modifying, is limited to people already experienced in customizing computer software. For live musicians, generally less inclined to modify their instruments, the tradition is to buy a customized instruments rather than doing it themselves. This makes for a level of customization at the manufacturing process, but keeping all features fixed afterwards.

An even more interesting aspect, is that what the designer may have thought is interesting in his instrument, the user might feel connected to a completely different function of the instrument. This is very apparent with the Roland TB-303, which was initially marketed as a replacement for a bass guitar player[12]. As such it was essentially a failed product which was hard to operate and which didn’t sound like a bass guitar at all. Only a handful of these instruments were sold initially. But at some point it was discovered that certain extreme settings, turned the sound into something special; a kind of bouncy, penetrating sound, perfect for house music. In this function it became so popular, that now, 30 years later, the price is almost tenfold of what it initially sold for. The weird sequencer interface of the TB-303 allowed for a new kind of expression which isn’t possible when playing a synthesizer by hand. This device is a lucky design mistake, which proves that the power of electronic instrument lies in making new types of sounds, rather than emulating existing instruments. It also shows how aiming to create your single inspiration might not work and that iterating will create more functional devices. Figure 9 shows a TB-303 with heavy aftermarket modifications.

Figure 9. TB-303 with DevilFish modification

3.3.8. Interfaces of acoustic instruments

A lot of comparisons are made between acoustic and electronic instruments, but is this valid? Some might say that electronic instrument are a whole new class of instruments and that as such, a constant comparison to

[12] http://www.synthgear.com/wp-content/uploads/2009/10/roland_tb303_advertisement.jpg

another class of instruments is more harmful than anything else. But even if you consider them to be a new class of instruments, we can learn from the old class, especially seeing they have a history longer than that of the Homo sapiens race itself.

Musical instruments have to follow a different interaction design philosophy than functional computer systems. Many acoustic instruments sound pretty bad when used by a new user. While electronic instruments, especially with presets, will sound great right away. This makes the difference in quality between a beginner and expert user much smaller and as such it diminishes the reward of learning an instrument. This causes that many common songs, use factory presets from the most popular synthesizers. Designers of electronic instruments often try to make all possible functions available to the user to make the instrument most useful. A comparison to acoustic instruments might reveal that this might actually diminishes the usefulness, because users will be overwhelmed by the functionality and just use whatever is easily accessible, often buying a new synthesizer when needing a new sound.

3.3.9. Thin line between instrument and toy

Electronic instruments do not only suffer from the disconnection between interface and sound; they are often seen a toys rather than serious instruments. This is especially so with smaller, portable instruments. Some products sit gladly in this twilight zone, while others suffer from it. There are three main reasons why instruments can be perceived as a toy.

The first reason has to do with the shape and material of a product. Many small instruments, like the Teenage Engineerin OP-1 shown in Figure 10, seem to be cute-ified versions of a synthesizer. Combining this with bright colors and plastic interface elements causes a big resemblance to toy instruments.

Figure 10. Teenage engineering OP-1 on the left, toy keyboard on the right

Another reason why portable instruments are not considered '*serious*' is that the sound created by them is not as one would expect from a serious instrument. There is often not much more than a small speaker placed in a convenient location, not only creating a limited frequency response, but creating an unpleasant frequency response.

A third reason is that small instruments often do not allow for rich interaction. Even in the few occasions that such an instrument is used in a serious performance, it is mostly used as a gimmick[13].

3.4. Case studies

As the Meeblip developers have given a few sources of inspiration, short case studies were performed on them. From the choice of these three cases can be concluded that Meeblip sees itself as becoming a wide-spread product, with the possibility to make a change in how people make music. However, by contrast, Meeblip presents itself as novel product, while the cases all exploit a sense of tradition and nostalgia. An advice on the aesthetics of the interface of new generations of Meeblip would therefore be to draw inspiration from vintage synthesizer or even inventors gear, rather than trying to be modern or minimal. The choice for studying these three products comes from suggestions by Meeblip.

[13] http://www.youtube.com/watch?v=4cHX-znop8Q

3.4.1. Lomography

Lomography is a movement and company which is focused on using '*bad'* cameras to create pictures with a characteristic atmosphere. Designed after low-cost, mass produced Soviet-era cameras, Lomography cameras purposely incorporate flaws like light leaking and distorted lenses. Two tiers of cameras are sold; beginners (€5-60) and specialty (€60-800). This allows an easy entry for users, but provides enough options to grow. Over time, the retro and nostalgic quality of the pictures as well as the '*hip'* act of doing Lomography, caused a huge following. Supporting this, a set of 10 guidelines was bundled with the cameras, including lines like: "*Lomography is not an interference in your life, but part of it*[14]." An obvious part of the success of Lomography is the fact that the manufacturers sell a feeling and a lifestyle rather than a product, which makes the cameras more likeable.

To use a similar approach for the Meeblip, two things should be done. Firstly, something in the sound or interaction style should be found that triggers a certain emotion or mood. Secondly, this *thing* should be described in a story as well as in video and music. Success comes with hitting the right note and reaching superior or viral status.

Figure 11. A Lomography camera photographed in lomography style

3.4.2. Korg microKORG

The Korg microKORG is a small synthesizer intended as a portable and veritable successor to Korgs previous synthesizers. It became a huge success due to its accessibility and low price, €450 at release. The interesting thing is that the success of the microKORG brought synthesizers to the stage and in the hands of many *non-synth-using* musicians. A few marketing factors that helped are: common availability, low price, great sound, versatility, being a brand with a heritage. The lesson that can be learned for Meeblip is that making a successful product is not only making a great product, but also incorporating smart marketing and streamlined distribution.

Figure 12. MicroKORG, used in a live setting

[14] http://www.lomography.com/about/the-ten-golden-rules

3.4.3. KORG MONOTRON

The design intent for the Korg Monotron was to make an analog synthesizer that can be used anywhere. The result is a wallet-sized instrument that is operable by touching a keyboard strip and turning small knobs. Korg succeeded in creating a product that was both attractive to synthesizer users, as well as being small and cheap enough (€40-€60) to be given as a present. While a user can't control the device like a conventional synthesizer, for instance you can't play tuned notes, musicians like the sounds that it produces. A smart move by Korg is to release the electronic schematics to the public. This caused people to make intricate modifications, which not only made the device more useful, but also generated a lot of free publicity. Again, distribution played a big role in getting these devices to the users, as their common availability allows for spontaneous purchases.

FIGURE 13. KORG MONOTRON IN USE

4. CONCEPT

This chapter will describe the usage scenario and the product design specification. This is followed by potential products fitting that description, before ending with personal concept and the choice thereof.

4.1. USAGE SCENARIO

As the design brief is very open, a specific user and scenario is chosen for the design. From scenarios described later in paragraph 4.1.1, as single scenario is chosen and described. Following this scenario, a new type of electronic instrument should be designed which allows for independent use. It could start a revolution in music making; it drags the electronic musicians out of their home studios and onto the street. Electronic music should be made anywhere, anytime. New musical genres could appear where the distinction between electronic and acoustic disappears all together.

It's mid-summer in Berlin and Brian, an electronic music producer, gets a call; his friend David and some others are jamming in the Görlitzer Park, if Brian would like to join in? He likes the idea, but as he's mostly a synthesizer player, and these synthesizers are all fixed in his studio setup, he is not very keen on taking one of these. Not to mention that he would need to bring an amplifier as well. There are a few toys though which work on batteries, like the Korg Monotron and his mini Casio keyboard, but these sound very weak and are not really taken seriously by his friend, even though they produce some nice sounds on his own records. So while it is not really his most preferred instrument, he takes his djembe and heads to the park, dreaming of some kind of acoustic synthesizer.

The first type of user is proficient with electronic instruments and is interested in using them in a new setting. He is used to all his synthesizers needing a different approach for getting the best sounds. As he has a large collection of different types of synthesizers, he knows the basics of all of them, but has some trouble getting in depth and gaining complete control of the synthesizers possibilities. While the most important thing for an electronic instrument is the quality of sound, he prefers synthesizers that feel nice to play with and that allow for a cool performance. The user is between 16 and 35 years old and is either a student or is just starting to get his career going.

A second type of user, from the design brief, is the DIY enthusiast. While he might not be so familiar with synthesizer operation or performance, he is mostly interested in building an interesting project. He has access to hand tools as well as a Fablab, where he could use specialized tools like a laser-cutter or a 3d printer.

The goal is to design an instrument that allows for the creation of electronic music in a normally acoustic environment. Key considerations are to make the instrument self contained, sound good, allow for rich interaction and to be replicable by DIY enthusiasts.

4.1.1. ALTERNATIVE USERS/SCENARIOS

Other than the previously described scenario, some other ideas were created; they are noted in Table 5 for reference. The final choice fell on the *jam-able outside synth*. This user was the best fit with what the company specified in the brief and seems to be something that has potency for being an distinctive product. Other scenarios relied too heavily on current trends in music or cannot fit in the Meeblip philosophy.

Title	User	Scenario	Functions
'Beginners' synthesizer	Musicians who have never touched an electronic instrument before, but who would like to make a start.	The user has been using music production software for a while, but is looking to make a step forward by buying a hardware synthesizers. He likes the thought of hands on control as well as the thought of better sounds.	The device should explain itself and pose as a tool to learn the basics of sound synthesizing. Should be integrated in a software environment.

Title	User	Scenario	Functions
303 clone	Musicians who are charmed by the Roland TB-303 synthesizer but can't afford one or its clones.	The sound of a TB-303 is needed, yet the real product is not available or too expensive.	Similar to Roland TB-303, with some added unique features.
Analog sequencer	People who are interested in old synthesizers and want a sequencer-controlled synthesizer, but can't afford to buy one.	While slowly building up a collection of analog synthesizers, this user is looking to sequence them. A simple digital sequencer could fit his needs for a lower price than an analog one.	A cheap device that functions and looks like the real thing. It could be quite a bit smaller to become a 'cute' version.
Generative synthesizer	Musicians making generative music, interested in a dedicated tool (as opposed to software patches).	An artist wants to build an interactive installation and is looking for a low cost solution.	Should run something similar to PD or Max. Easy to interface with other objects.
Jam-able outside synthesizer	Musicians who want electronic sounds, but doesn't want the hassle of all sorts of external gear like amplifiers and controllers.	The weather is perfect and your band members want to go to the park with their guitars and cajon to jam. Sadly, there are no power sockets in the park for your synthesizer.	It needs to be battery powered and sounds good on a sound level that can compete with an acoustic guitar and voice.
Live dubstep bass synthesizer	Live dubstep performers/musicians.	Playing live on stage with a dedicated bass sequencer/synth. The user wants something more performative than programming it on his laptop.	Programmable on the go, huge variations of sounds, sequencer on notes as well as timbre.
Serious toy synthesizer	Someone who is interested in electronic music, but never took the step.	Someone is looking for a present for a friend that has expressed interest in making music but never taken the step.	It should look appealing and fun, while harboring a serious and musical sound.

TABLE 5. PRODUCT SCENARIOS

4.2. PRODUCT DESIGN SPECIFICATION

To define the needs set in the scenario, a product design specification is made. The needs can be used as criteria for evaluation and serve as guidelines for a next iteration of design (Johnston, 2011). Musical instruments provide a complex interaction, which make it hard to set quantitative needs, therefore qualitative descriptions are used which can be tested in user-evaluation. The specifications are formed as follows:

- The design needs to contain a Meeblip Micro
- The design needs to be replicable by amateurs and licensed open source
- Materials need to be locally sourceable
- Manufacturing methods commonly available
- Documentation should be comprehensible for amateurs
- The design needs to be customizable by the builder
- The instruments needs to be loud enough to compete with acoustic guitars and hand drums
- The design needs to operate in both a low-tech setting
- The design needs to look as a serious musical instrument (as opposed by a toy)
- The design needs to sound like a serious musical instrument (as opposed by a toy)
- The design needs to look attractive to the user group
- The design needs to be of such sizes that it is usable by 95% of the population
- The text on the instrument needs to clearly legible and understandable
- The device needs to withstand at least two years of normal use in performance situations
- The user should be able to play the instrument for at least 30 minutes without fatigue

- The instrument should, over time, become blindly playable, as to allow interaction between musicians
- The instrument needs to adhere to the criteria of Armstrong (2006): *situated*; *timely*; *multimodal*; *engaging*; *sense of embodiment*.
- The instrument should allow a *conversational* mode of interaction (Johnston, 2011).
- The device need to survive a fall of one meter height
- Electrical component should be designed for a minimum of noise
- The design should not be focused on a single musical style
- The design needs to be documented open source
- The design needs to be made within eight weeks

4.3. Product Ideas

4.3.1. Existing product ideas

As there might already be instruments which fit the scenario and/or PDS, examples of these have been sought. Products are checked in columns 1, 2 and 3 for having one or more of these characteristics:

- Self-contained/portable (1)
- Jammable/enhanced expression (2)
- Regarded as a serious instrument (3)

Name	Description	Evaluation	1	2	3	Picture
Casio Mini keyboards	Meant to be portable variations of their larger keyboards. "High quality tones."	While marketed as a serious instrument, these keyboards are generally seen as kids toys or learning tools. They generally sound weak.	x			
Eigenharp	A fully customable synthesizer controller. The knobs are sensitive to pressure and vibration. A mouth controller is added for further control.	This device allows for a more musical interface for synthesizers, putting multiple parameters under one finger. But it is dependent on an external synthesizer for sound.		x	x	
Haken Continuum	A strip of touch sensitive fabric, controlling an external synthesizer.	Like the Eigenharp, this instrument allows for more dynamic musical expression, but is needs an external synthesizer as ell.		x	x	
Handheld electronic keyboard instrument (Hacker, 1991)	A patent describes an attempt to make a portable instrument, consisting a keyboard part and an electronics part.	The device as designed is unwieldy, yet could be made considerably smaller with modern technology. The fact that no realized product is on the market speaks volumes.	x		x	
Standard keyboard	Many keyboards have the option of running on batteries, allowing use without needing a power chord.	While these instruments should be able to fit the scenario, it is currently barely used as such. Part of this might be due to the sound being described as cheesy or weak. Though it's the loudest and fullest of all the instruments described in this table. This feeling might be strengthened	x	x	x	

Name	Description	Evaluation	1	2	3	Picture
		because the device aims to emulate acoustic instruments.				
Keytar	Similar to a keyboard hanged from the shoulder, meant for providing a more interesting performance.	The Keytar has gathered a cultural stigma of not being a serious instrument. The sound coming from this instrument is weak, mainly because it's meant to be amplified through external gear.	x	x		
Korg Electribe	An instrument that allows for the creation of electronic music, using a step sequencer.	This is one of the few instruments allowing the creation of complete electronic tracks. It allows for jamming in ways completely different from the other instruments. It would be interesting to see what happens if you combine this with a portable amplifier.		x	x	
Korg Monotron	A small form factor synthesizer, controllable from a small ribbon.	There is not really a way to play a tuned note, so it's hard to play together with other people. The sound is mostly described as annoying when coming from the speaker, yet it sound great on a bigger sound system.	x			
Teenage Engineering OP-1	A small but powerful synthesizer, containing several sequencing and synthesis methods. It allows for the creation of complete tracks	Even though it contains a speaker and a battery, the sound is too weak and thin to be played in collaboration with acoustic instruments. The fact that the keys are not pressure sensitive, means that there is less room for expression, which is compensated by the diverse types of synthesis and sequencing. This device is fun to use though.	x			
Yamaha Omnichord	An electronic variation of an autoharp. Is meant to be played in a similar setting.	This product seems to fit the design problem, but it has only been a specialty instrument. This mean there are some things that can be learned; The plastic casing makes it look like a toy and the sound, again, is not so pleasing.	x	x		
Stylophone	Similar to the Monotron in form factor, but older. It is operated using a stylus.	While this instrument is does allow tuned notes to be played, the sound is very harsh and unpleasing, making the general perception of an toy instrument.	x			

Name	Description	Evaluation	1	2	3	Picture
Yamaha Tenori-on	A matrix of buttons which act as a melodic sequencer.	Fun to play around with, but hard to use in interaction with other musicians. Again, the sound is not very pleasing, while the melodies themselves are.	x			
Zanzithophone	A digital saxophone, played similar to an actual saxophone.	Is designed as a serious instrument, but is regarded as a toy. The level expression is a lot lower than a real saxophone and the sound is weak.	x	x		

TABLE 6. PRODUCTS FITTING DESIGN BRIEF

NOTE: 1: SELF-CONTAINED/PORTABLE 2: JAMMABLE/ENHANCED EXPRESSION 3:REGARDED AS A SERIOUS INSTRUMENT

4.3.2. SOUND QUALITY

From an analysis on the sound quality, one main problem with the existing istruments can be found; they nearly all sound terrible. It is assumed that this is either due to the fact that they try to poorly emulate existing instruments or due to the little speaker in an acoustically unfavorable encasing. Another finding is that compact plastic products all seem have a big resemblance to toys.

To remedy these problem special attention should be given to acoustics design. This is needed to ensure a full sounding instrument. Care should also be taken in shape and material to avoid the stigma of toys. The expanding of dynamics by the Eigenharp and Continuum shows that it is desirable to have multidimensional buttons; not only pitch, but also loudness and timbre from the same button-press.

4.3.3. TAKING ELECTRONIC INSTRUMENTS OUTSIDE

As been stated in the scenario, electronic musical instruments are confined by their cables and peripherals to studios and stages. When bringing your synthesizer outside, you should at least need to take an amplifier and battery with you. As there are no batteries made specifically for this purpose, the best method is to modify some car batteries. Widely known street artist Dub FX, seen in Figure 14 which is a screenshot from a Youtube video[15], is a living proof that electronic music works outside. However at the same time, his immense setup demonstrates the big effort needed to get to that point.

How can we emancipate electronic instruments, so to bring them into the outside world without big efforts? Previously the writer has worked on two projects dealing with this question. The first project, shown on the left in Figure 15, is a DJ sound system build into a cargo-bike. It would allow a group of people to enjoy electronic music wherever they want, just taking this bike with them. To complete the party, a beer cooler is integrated in the middle. This product is mostly a social catalyst but would not directly contribute to a more performative setting for the musician.

[15] http://www.youtube.com/watch?v=kTOi1YSYxnk

FIGURE 14. STREET MUSICIAN DUBFX IN DUBLIN, 2008

The second project, shown on the right in Figure 15, is an object to enable street performance of electronic music. Where Dub FX needs to bring a cart with a big amount of gear, the amplifiers and speakers are now embedded in an object placed in the park. The three objects on the bottom right corner form a sound installation and a stable surface on which the musician's instruments can be placed.

Both of these projects try to take it for granted that electronic music is played from a static position, that the music is made by a single person and that, as such, there is no need to support collaboration. While some of the aspects of emancipation can be used from these projects, it is obvious that the product fitting the scenario, something more portable and performative is desirable.

FIGURE 15. TWO PREVIOUS OUTDOOR ELECTRONIC PROJECTS BY THE WRITER

4.3.4. ALTERNATIVE CONTROLS

The design project described in this document is by far not the first time that alternatives are sought for the common form factor of electronic musical instruments. The Eigenharp is a notable example. The possibilities are endless with this instrument, although those possibilities are kept under control and are offered in a way which invites expression. John Lambert, the developer of the Eigenharp talks about his days touring with a setup consisting of many synthesizers to the BBC[16] :

[16] http://news.bbc.co.uk/2/hi/8294355.stm

"The gigs were really good fun," says Lambert, "but the setting up was just a nightmare. Lots of stuff, equipment, wires, endless stress. (…) At the same time I would go to the folk club, take my acoustic guitar out of the box, tune it up, get a pint of beer, and play."

To overcome this problem, a more musical instrument-like controller was made. In using keys sensitive to any subtle motion of the finger and using high-grade materials, Eigenharp managed to make an electronic instrument that could feel as a part of the musician again. An interesting thing to note is that the sound that is produced is completely dependent on what kind of synthesizer is placed after the Eigenharp. However where this opens up a huge range of freedom for sounds, it makes the instrument an entity without a definite personality. Additionally, the musician still needs to rely on a sound system out of his direct control for feedback on what he is doing.

FIGURE 16. DINO SOLDO PLAYING AN EIGENHARP

A funny display of electronic instrument design is how the comic duo 'Flight of the Conchords' show their vision of how a performative electronic musical instrument should look like; just combine a computer, a guitar and some monitors, and you can rock out! It's a nice inspiration on how far you can go with morphologically combining parts and how some actual instruments seem like just that.

FIGURE 17. PERFORMATIVE DRUMCOMPUTER FROM "CAROL BROWN" - FLIGHT OF THE CONCHORDS

4.3.5. DESIGN PROPOSALS

Following previous analysis, three concepts have been made for the design project as shown in Table 7, of which one has been chosen. The chosen concept will be explained in-depth in the next chapter. A general type of instrument was concepted that would fit the scenario and of this, a number of variations were made. The general type of the concept can be stated as:

A handheld, portable instrument which is self-powered and contains a speaker. The device can play loud enough to be heard when playing together with other (acoustic) instruments. It should be directly identified as a serious device. While no knowledge would be needed to make a sound, it might take some hours of playing before real music could be made.

Variation name	Function
Bass box	This instrument is especially made for making low-frequency sounds. This device could be very useful, as no ideal acoustic bass instruments exists; a double bass is heavy and difficult to transport, while an acoustic bass guitar rarely produces enough low-frequency tones. Inspiration for the interface should both be taken from traditional bass instruments as well as bass synthesizers.
Keyboard synth	A small keyboard with integrated speaker. Where portable Casio keyboards sound like toys, this one should sound like a real instrument. It should not try to emulate traditional instruments, but rather exploit the unique sounds of the synthesizer in it. Instead of a traditional keyboard, it might use Eigenharp-like keys.
(Semi-) Computational Synth	A portable synthesizer similar to the keyboard synth, but with a computational input method. This could be a sequencer, something inspired by trackers or another computational method. The basic premise is that the computational sound generation method, the synthesizer, should be complemented by a computational input method. Care should be taken that the user should still have enough involvement with the instrument to feel embodied.
Drum box	A portable version of known drum computers. It should be programmable, yet it should feature live interaction by allowing pattern-editing on the fly and improvised fills. Drum computers have already found their way into indie and folk bands, it might be a small step to adopt them for acoustic settings as well.

TABLE 7. THREE CONCEPT VARIATIONS

In consultation with the Meeblip company, the *computational synth* was chosen. The sonic capabilities of the Meeblip would make the *drum box* hard to make. A *bass box* would require a lot of considerations on acoustics and amplifiers, as it’s hard to make small things making big sounds. Finally, the *(semi-)computational synth* was chosen over the *keyboard synth* as it seemed to fit better to the synthesizer design guidelines and the Meeblip company preferred an instrument with a sequencer.

5. Design

This chapter will describe all the steps made in the actual design. First a general description of the design will be given, followed by the more detailed considerations in interaction design, ergonomics and electronics.

5.1. General Description

The design proposal is a new type of synthesizer, meant to be played anywhere, anytime. The design is focused on performability as well as portability. While it is a synthesizer, it aims to be integrated in an acoustic-music setting rather than an electronic-music setting. This is achieved by being completely self-contained, having pleasing sonic properties and having the aesthetics reference towards acoustic instruments. The design proposal is shown in Figure 18.

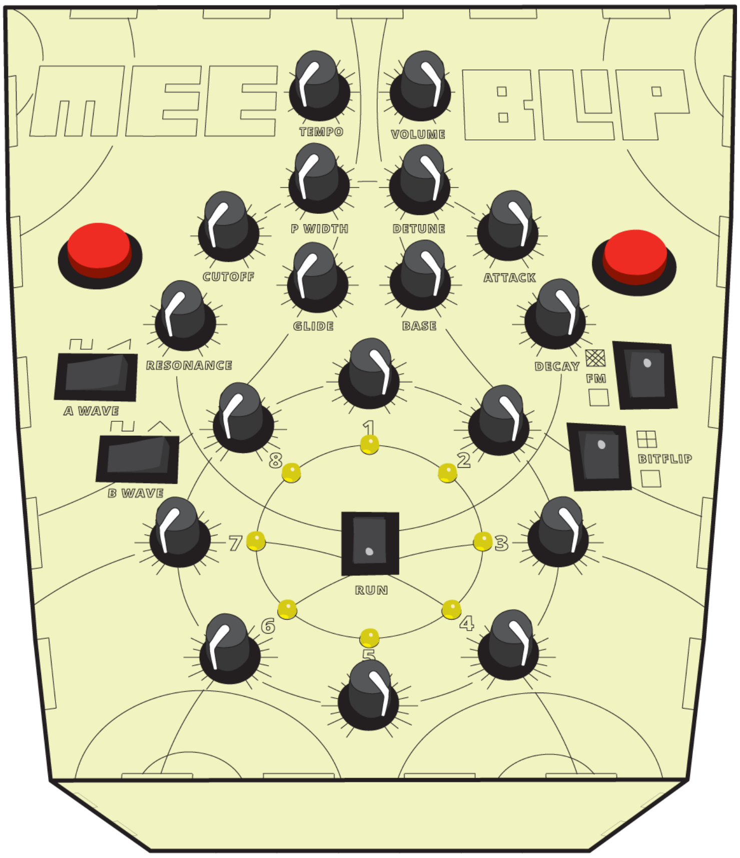


Figure 18. Illustration of the design

As instrument design is a highly iterative process (Johnston, 2011) this design should be seen as a first step in a longer process, and as such the function is to demonstrate the feasibility and to evaluate the design principles for a next iteration. This means the design has a focus on ergonomics and interface design, while considerations such as maintenance and marketing have taken a back seat. Following this, the end result is a fully functional prototype, which could serve as a basis for further development to turn it in to a marketable product. The final prototype is shown in Figure 20.

The instrument can be used as follows. The instrument should be played while sitting, as shown in Figure 19. To turn the instrument on, a power button on the back side can be flipped, the LEDs will flash to indicate the instrument is initializing. The first thing to do would be to check if the volume knob, the top-right *volume* knob, is at the desired setting. Now the instrument can be played by setting the sequencer and pressing the red dynamic buttons or using run mode. The sequencer is the ring of knobs and LEDs at the bottom of the device. The LED will indicate the note that is played at that moment, while the knob will allow you to set a relative pitch for that note. To step through all the notes the right or left button should be pressed to turn clockwise or counterclockwise respectively. By pressing the buttons with more or less force, the loudness of the note can be controlled. Alternatively, the run button can be pressed to make the sequencer run automatically, the tempo can be controlled with the top-left *tempo* knob. Now, by pressing one of the *red* buttons, you can jump to the next or previous note as well as syncing the sequencer to timing of the button press, allowing for use together with other musicians. The notes which are played are always relative to the note set by the *base* knob at the center-right. All other knobs and switches will allow the timbre of the sound to be adjusted, with the most notable being the *cutoff* knob on the center-left. A graphic explanation can be found in APPENDIX D.

FIGURE 19. USAGE POSITION

To relate the design with the literature review, we go back to the model of an electronic instrument by Marshall (2008) from chapter 3.3.1. The *physical interface* would be all that is seem in Figure 18 and Figure 20 plus the speaker on the backside. The *software synthesis system*, is the Meeblip, of which no adjustments will be made. And finally the *mapping system* consists of two parts, both the mapping system already present in the Meeblip, and a separate microcontroller dealing with the computational input.

The device is designed to comply with Armstrong's (2006) criteria from chapter 3.3.2. The instrument can be s*ituated* because all parameters can be edited on the fly, without needing additional cognitive processes other than finding the right knob, which in turn is always in the same spot. Because all functions always remain in the same location, it can be assumed that the users learn their place over time, and will be able to operate the instrument without looking at it, enabling better communication with other musicians. Due to the dynamic buttons, the designed instrument contains one of the few *timely* sequencers. It might be hard to suddenly change key or switch a chord, but this same limitation never held the blues harp back. *Multimodality* is provided though the same features that make the instrument *situated*. Additionally the fact that the instrument makes sounds itself, contrasting with other synthesizers, make that impulses coming to the different senses are correlating more closely. The instrument feature multiple modes *engagingness*; while playing the dynamic buttons attention has to be on maintaining the rhythm, and when using the *run mode*, no attention is required, but it can be directed on tweaking the sound. The s*ense of embodiment* is present in the instrument by both its direct interface and its speaker, though actual musicians will be the judge whether or not the instrument can function as extension to his or her body.

The main *value* of this instrument lies in the fact that it can be used anywhere, but still has the positive characteristics of an electronic instrument. If successful, the instrument will allow electronic instruments in an acoustic setting, without it being strange. The instrument will have an air of mystery around it, as no one knows to categorize the shape and style of interaction, which will make it more interesting for the user.

FIGURE 20. THE FINAL DESIGN

5.2. ENCASING DESIGN

The general shape of the instrument was determined from the shape ergonomically most suitable with the playing position. Figure 21 shows how this shape came to be. In addition to the 3d models, which were used for quick assessment of shape, cardboard prototypes were made to validate the ergonomic assumptions. The trapezoidal shape, shown bottom-right in Figure 21, is the final choice. The slanting of the panels allows the instrument to be placed between the legs of the user without needing force to hold it there. At the same time it provides a solid vertical balance, so pushing the buttons won't make the instrument fall.

Ergonomic data was used to determine the maximal length of the design so it would fit easily between the legs as previously seen in Figure 19. Usage position. As direct measurements were not available, a correlated combination has been made of two parameters, leading to a maximum length of 312mm (DINED, 2004).

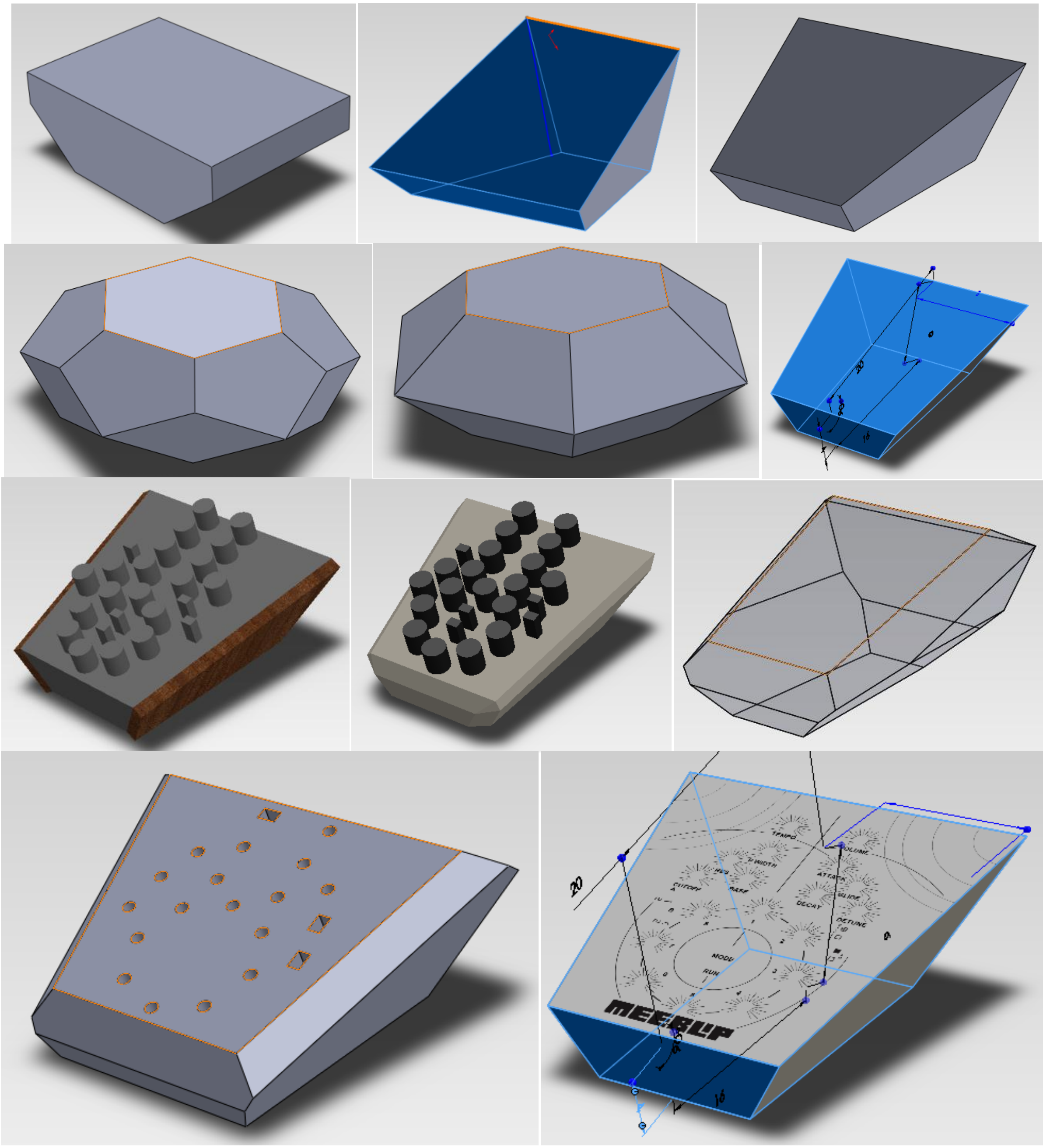


FIGURE 21. DIFFERENT ITERATIONS IN SHAPE

A few different materials were considered, including laser cut acrylic, bend metal and 3d printed ABS. As this provided the needed strength, acoustic properties and aesthetics, high-quality birch-aircraft plywood was chosen. While most traditional instruments are made from wood as well, the designed instrument is quickly associated with them.

For the assembly of the casing, it was decided to go with interlocking panels, glued together. To allow maintenance, the top panel will sit loose and is hold on by a rubber band. As the angles in which the panel sit against each other vary, the interlocking pattern is quite complex. To assure a perfect fit, 3d models were used to get the right dimensions, as seen in Figure 22. A downside to this method is that a new model would have to be made for each variation of panel thickness.

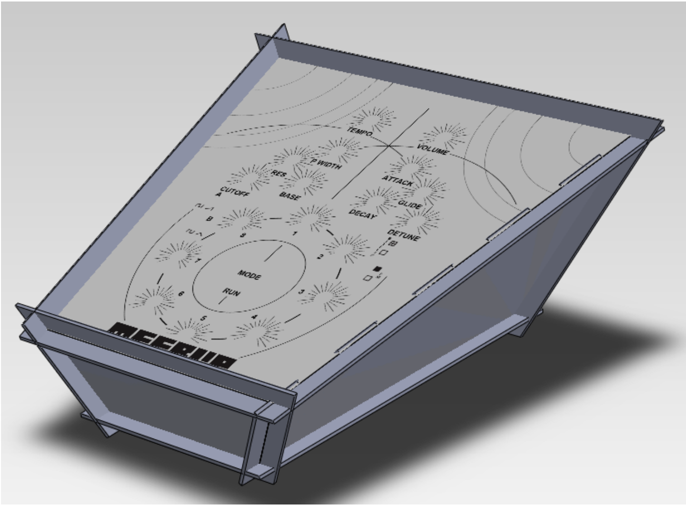


FIGURE 22. INTERSECTIONS IN 3D MODEL

5.3. INTERACTION DESIGN

One of the big questions in Human Computer Interfaces (HCI) is the balance between clarity and versatility; adding more features often leads to adding more interface elements, which increases the effort needed to understand the interface. Because clarity is so important for musical instruments, versatility will be sacrificed. Starting with the input method, the next sections will describe the decisions forming the interface.

5.3.1. COMPUTATIONAL INPUT

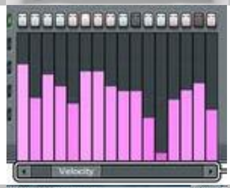
As been described in chapter 4, the instrument should be controlled by a semi-computational input to match the computational sound generation method. This means that the user does not have full direct control on the instrument, but has to set certain parameters, which will be computed into music. As there are many ways in which this can be executed, an analysis has been made of different types of possibilities, as seen in Table 8. One of them has been chosen and formed into the actual used computational input method through a series of experiments.

Computational input for a musical output is often called sequencing, as the activities often consist of arranging musical elements in a sequence, to form a piece of music. Duignan, Noble, & Biddle (2005) identify four common types of sequencer: textual language music tools, music visual programming, sample and loop triggers, and linear sequencers. Specific types within these categories have been found and are noted in Table 8. Considering the scope of this research, it has been chosen to optimize an existing computational input, rather than trying to come up with a completely new type.

A number of these input methods, like the *notational sequencer* and the *tracker*, need a graphical interface (screen) for operation, which would not fit the PDS and are thus not useable for this design. Other types, like the *real-time sequencer* and the *step-note sequencer*, have notoriously bad interfaces which do not fit with the goal of performability. Others, like the *chase-light sequencer* and the *cylinder*, have proven to possess more potencial for an interesting performance.

A big contradiction between all these methods and the scenario is the fact that they need to be programmed before a music performance, after which the *play* button can be pressed. This allows for little interaction needed. To fit the PDS, an input method is needed that can be played right away and that can be programmed on the fly during performance.

A solution is found by taking the *circular sequencer*, one of the most intuitive and performative computational inputs, and controlling the clock manually, rather than automatically. This manual control will give dynamic information at the same time, allowing for a more dynamic expression.

Name	Description	Image
Chase-light sequencer	The type of step sequencer most usual for sequencers triggering percussive sounds. A sound is selected after which the user can select the notes which should be played, represented in an array of usually 16 buttons.	
Circular sequencer	Similar to a linear sequencer but the knobs are placed in a circle.	
Cylinder	Similar in concept to the paper roll, yet here a cylinder has small pins on it, these pins hit a vibrating membrane. The device is operated by turning a crack.	
Grid sequencer	A step sequencer where all possible notes are laid out in a grid, the clock steps though the vertical lines from left to right, the selected notes will be played.	
Linear sequencer	A type of step sequencer; notes or modulation parameters are set with rotary knobs.	
Modulation sequencer	Similar to the linear sequencer, however instead of using a rotary knob, sliders are used. This allows for easier visual detection of relationship between notes.	
Notational sequencer	Music is represented like sheet music, making the music easy to understand for traditionally trained musicians.	
Paper roll/punch card	Holes are made on a long strip of paper; this is fed through a device which it reads it out. Each location of a hole corresponds with a certain note, similar to the grid sequencer.	
Piano roll	The digital version of the paper roll, which gives a huge advantage in ease of programming and editing.	

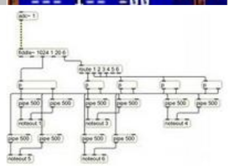
Name	Description	Image
Real-time sequencer	Often called a Composer. The musician plays in sequences of notes using an external keyboard, the device can repeat the sequences and put them in different orders, allowing for longer pieces to be made.	
Sample sequencer	A computer program that displays representations of recorded waveforms. These can be moved around in time and on different tracks to create a composition.	
Step-note sequencer	Two buttons allow you to step through all possible steps. On each step you can set a note, using the keyboard, and some additional parameters.	
Tangible interface	An arbitrary device using with a physical object of which the interaction with a musician is translated by a computer into sound. Might use sensors or visual scanning techniques.	
Textual programming	Programming languages such as Csound allow the creation of music through programming logic. This is the most abstract of all inputs, as all parameters will have to be defined by the musician himself.	
Tracker	This instrument is similar to a linear sequencer, although textual on a computer. Using the expanded capabilities of a virtual environment, the sequences can be infinitely long, allowing the creation of entire songs from one sequence.	
Visual programming	Similar to textual programming, however abstractions are used. Components are represented in boxes which can be connected to each other by drawing lines. This allows for much of the power of textual programming, although in an environment more natural to a musician.	

TABLE 8. DIFFERENT TYPES OF COMPUTATIONAL INPUT

5.3.2. EXPERIMENTS

Several simple prototypes have been made in Max for Live, which itself is *a visual programming language*. This allowed for a quick assessment of the complexity of programming and of what functions would be interesting on a sequencer. The prototypes send MIDI data to the Meeblip for actual sound and were controlled from an Akai MPK[17] to get a feel of the physical interaction. The prototypes are assessed by the designer both from the PDS as well as the allowance for making music and emotion of interaction. The code for these experiments can be found in APPENDIX A. Note that these experiments were done parallel to what is described in sub-chapter 5.3.1 on types of computational input and were part of the process of finding the right type of computational input.

5.3.2.1. SIMPLE EIGHT-STEP SEQUENCER

The first experiment was simply meant to get a step sequencer working inside the Max-for-Live programming environment. It implemented all basic controls needed for sending MIDI notes and allowed for any possible note within the MIDI definition.

---

[17] http://www.akaipro.com/mpkmini

It was found that while this type of sequencer allowed for a lot of freedom, it functioned somewhat unwieldy and did not really invite to jam. The fact that numbers are displayed underneath the knobs makes it a lot less musical, as it is hard to see the relationship between e.g., note 45 and 57, which in letter notation could be easily identified as being an octave apart.

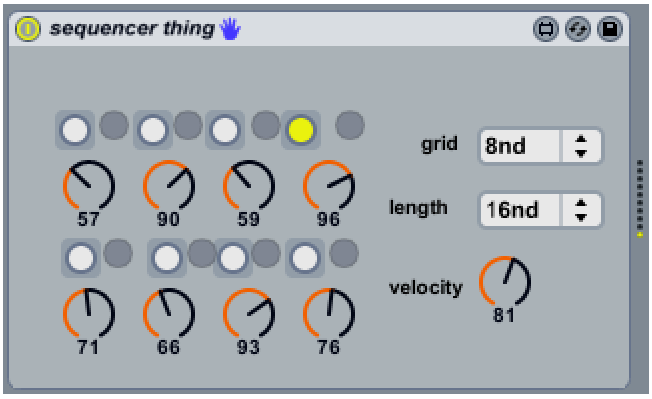


FIGURE 23. SEQUENCER THING EXPERIMENT

5.3.2.2. CIRCULAR ARPEGGIATOR

Similar to the previous experiment, however circular and instead of setting absolute notes, you set relative notes. When a note on the keyboard is played, a sequence will play relative to that note. In this experiment, the relationship between notes became more obvious and, as a result, this prototype is more fun to play with. While music is usually written from linear from left to right, this circular representation did not prove strange. In contrast; the fact that there is no definite '*home position'* was felt as positive.

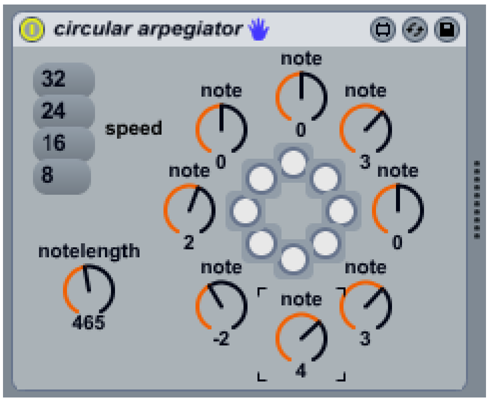


FIGURE 24. CIRCULAR ARPEGGIATOR EXPERIMENT

5.3.2.3. TIMBRE SYNTHESIZER

While the previous experiments have been sequencing the note, this experiment sequences the timbre. It does this by modulating a filter. While it generated interesting sounds, this was felt more like a sound-design tool and it did not invite for jamming. Just like the last experiment, this one needed an additional keyboard.

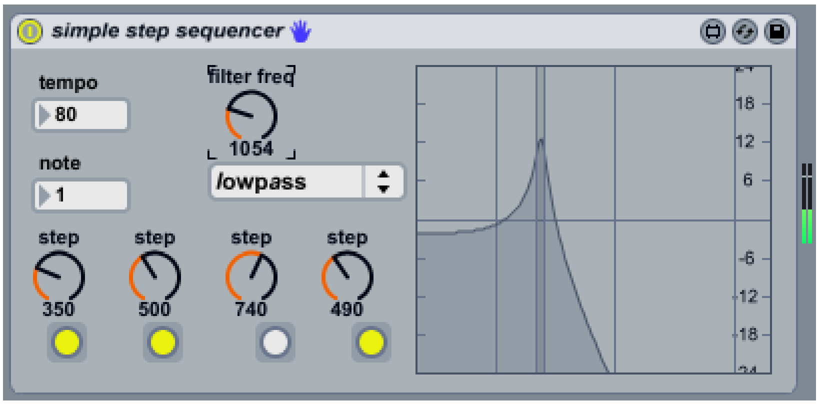


FIGURE 25. SIMPLE STEP SEQUENCER EXPERIMENT

5.3.2.4. RHYTHM SEQUENCER

This experiment sequences a drift in rhythm. Four notes triggered in regular intervals, by setting a delay time (ms) with the knobs. This way a swing rhythm or a samba rhythm could be made. Playing around with this led to some interesting results, but it was more a set-and-forget thing, than something that directly led to musical creation. Again, a keyboard was needed.

FIGURE 26. RHYTHM THING EXPERIMENT

5.3.2.5. GENERATIVE SEQUENCER

As an effort to increase the computational aspect of the sequencer, this generative-note creator was made. You input it with a rhythm from a single button, which it randomly converts to a melody as well as a different rhythm. This experiment delivered some spooky sounds, but it did seem to fit the concept as it did not allow any expression.

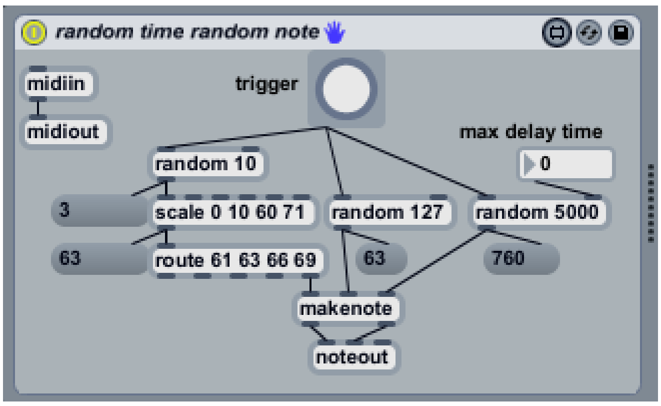


FIGURE 27. RANDOM TIME AND NOTE EXPERIMENT

5.3.2.6. CIRCULAR SEQUENCER WITH MANUAL CONTROL

In this section we return to the circular sequencer. Analyzing existing sequencers and the previous experiments, it is seen that a sequencer takes over nearly all of the usual tasks of a musician. You set the pitch up front, you dial in a volume and the rhythm is locked. What happens if we bring back the volume and rhythm to the musician, and let the sequencer just deal with the pitch of the note. Two dynamically sensitive buttons are provided; the first one lets the musician step through the sequence while sensing how hard the button is pressed and the second lets you step back to the first note of the sequence, to allow more control. As the first experiment showed that 127 choices is too much, the choice of notes is limited to an octave higher, lower and everything in between.

This prototype was allowing more expression. While the notes wouldn't really be adjusted while playing, the pre-set sequence felt as if you were playing a hang[18]. Some variations were tried with the second button stepping backwards rather than resetting. Variations in the number of selectable notes did not provide conclusive results. The fact that there is more freedom in rhythm makes this experiment feel more like an instrument than a machine.

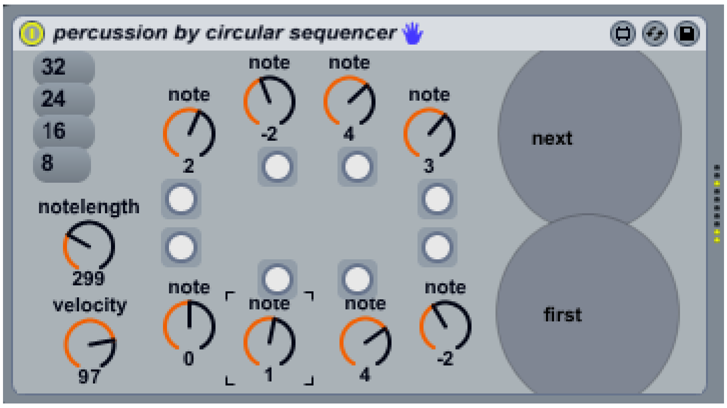


FIGURE 28. CIRCULAR SEQUENCER EXPERIMENT

5.3.2.7. HARDWARE SEQUENCER

To test the feasibility of making this type of sequencer with hardware, a proof of concept was made using an Arduino[19] and two 4051 multiplexers. It functioned similar to the first experiment but with four steps rather than eight, to limit the amount of possible problems in wiring. Through this experiment it was found that making an Arduino-based sequencer is workable and provides a very solid solution. Figure 29 shows the setup and the scheme.

[18] A type of melodic percussion instrument
[19] A prototyping platform containing a easily programmable microcontroller

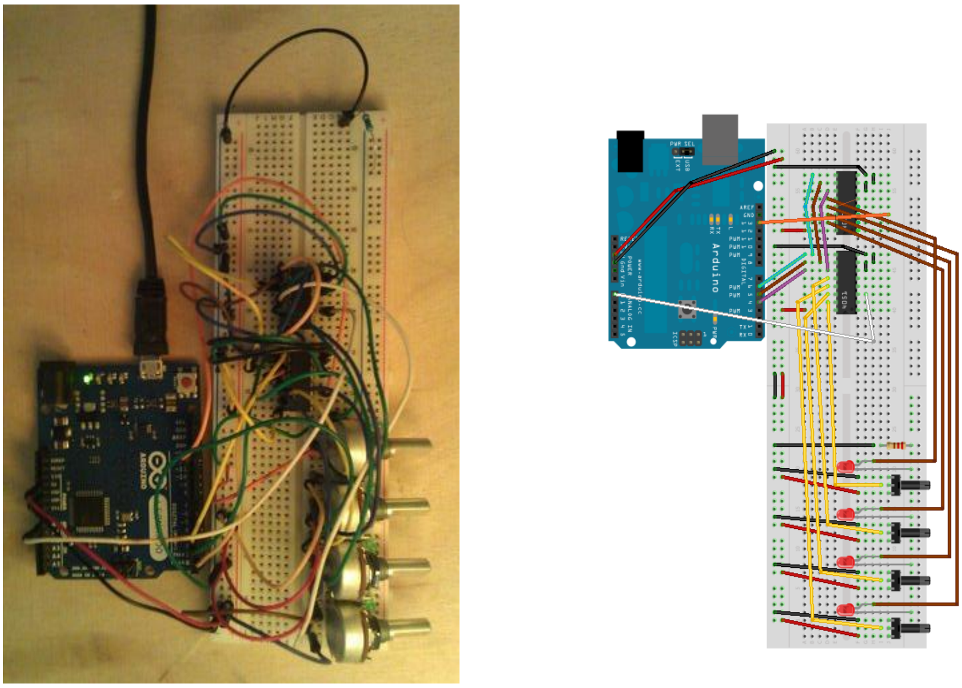


FIGURE 29. ARDUINO SEQUENCER EXPERIMENT

5.3.2.8. CONCLUSION

From all the sequencers analyzed, a manually controlled circular sequencer as described in 5.3.2.6 was chosen. In contrast to the other sequencers, it allowed for more dynamic expression as well as rhythmic variation. The circular aspect was seen as more favorable and intuitive compared to the linear representation. Figure 30 shows the final implementation of the sequencer.

FIGURE 30. A CLOSE-UP OF THE SEQUENCER

5.3.3. INTERFACE LAYOUT

With the type of computational input defined as a circular sequencer, a form had to be sought to translate this into the actual interface. To control the synthesizer, additional knobs and switches needed to find the right place on the interface. In reducing the amount of controls, ten essential knobs and four switches were defined to be placed on the interface. Using paper prototypes, many variations of possible interface layout were tried, as shown in Figure 31. Special care was taken to avoid biasing towards left or right handedness and to maximize the amount of space in between the knobs and switches.

A common critique on the Meeblip SE was that the knobs are too close together and that people got their fingers stuck because of that. Using ergonomic data, it was found that a space of at least 19 mm (DINED, 2004) between knobs should be used to allow a finger to fit in between them.

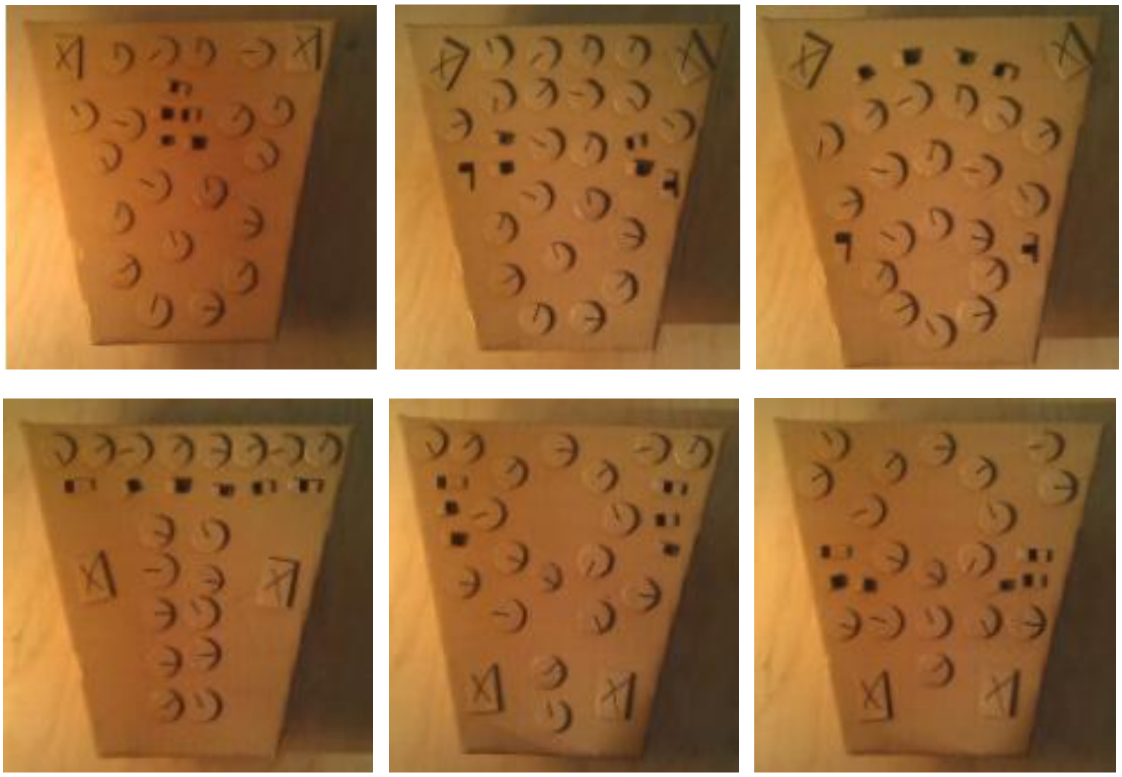


FIGURE 31. PAPER PROTOTYPING OF THE INTERFACE

With a suitable layout found for the knobs and switches, design continued on the graphical aspects of the interface. Figure 32 shows several variations, where care was taken in grouping corresponding interface elements and making the text legible. The font, logo and potentiometer markings are inspired by the interface graphics from the other Meeblips.

Colorblindness is an often occurring condition which can lead to problems in receiving the right information from interfaces. For instance, a multi-color LED, used to indicate an initializing state with red and an operational state with green, might not be seen by a colorblind person as having two different states. In this design, these kinds of problems were avoided by using a dark-on-light color scheme for the text and using single color LEDs.

The graphic will be either added to the instrument by using a decal or by graphing it into the panel. This can give some problems in legibility in low light. A solution can be backlit text, but this is a complex thing to do and the usage will generally be in environments where backlit text would not have any use.

FIGURE 32. THREE VARIATIONS OF THE INTERFACE LAYOUT

5.4. TYPE OF KNOBS

The choice of knobs is considered on the shape and material of the available types of knobs. It was not feasible to create a new type of knob specifically for this project, so a prefab knob needed to be found. As 18 knobs needed to be bought and they each cost €0.50 to €3.00 a piece, an expensive choice might take a huge chunk out of the budget. After testing quite an amount of locally available knobs, one soft knob stood out. While knobs tried to improve the ergonomics by using grooves on, they were all made out of hard plastic. The chosen knob was the only one made out of a soft plastic, which greatly enhances the feel of the instrument.

FIGURE 33. DIFFERENT TYPES OF KNOBS

The dynamic push buttons and switches went through a similar process, but due to a limited variation in the available types of buttons and switches, a choice was made for those most durable and pleasant to the touch.

FIGURE 34. DETAIL OF THE INTERFACE

5.5. ACOUSTICS DESIGN

As been concluded in the competitor analysis, nearly all competitors failed in making their instrument sound good through the inbuilt speaker. The quality of sound from a speaker system is a complex interplay between the input, speaker, the amplifier and the enclosure design. Enclosure design is the part competitors completely disregarded.

Speaker enclosure design is a complex field of work which seems more like an art than a science. The basic premise is that you can tune the enclosure to a certain note, which amplifies a wanted frequency range or supports where tone in missing. Because it's very complicated to predict what frequency range will have the most effect on the actual object, an approximation is used. Our general goal is to avoid sounding weak and 'toyish'. So we add 'warmth' and 'fullness'. The frequency chart of Minieri (2009) shows that a boost anywhere between 100 Hz and 300 Hz could lead to an improvement in these fields. To determine the resonant frequency of the instrument, it is assumed that it can be modeled as a Helmholtz resonator using the formula shown in Figure 35. Note that this disregards the acoustic properties of the material of the encasing and the electronic parts. The formula is defined: f_H is the resonant frequency (Hz), v speed of sound in a gas (~343m/s), A is the cross-sectional area of the neck (m^2), V_0 is the static volume of the cavity (calculated from SolidWorks: 0,0025 m^3), L is the length of the neck (m) (Wolfe, 2006).

$$f_H = \frac{v}{2\pi}\sqrt{\frac{A}{V_0 L}}$$

FIGURE 35. HELMHOLTZ RESONANCE

Using an Excel sheet, different variations were made and a most convenient variation was found; When a sound port is made with a hole with a 2.2cm diameter and a 1cm height, a f_H = 213 Hz can be made, which lies within the desired range, allowing for errors due to unusual shape and the effects of the electronics sitting inside the box. This is shown in Figure 36.

FIGURE 36. THE SPEAKER PORT (LEFT) AND THE SPEAKER (RIGHT)

5.6. ELECTRONIC DESIGN

As Meeblip does not possess a fully stocked electronics workshop in Berlin, a big part of the electronics design working was with whatever was available. As the writer has never done a complete electronics project before, it was a highly instructive process of putting everything together. The electronic design decisions will be discussed in order of influence. Figure 37 shows a diagram of how all electronic components are linked together.

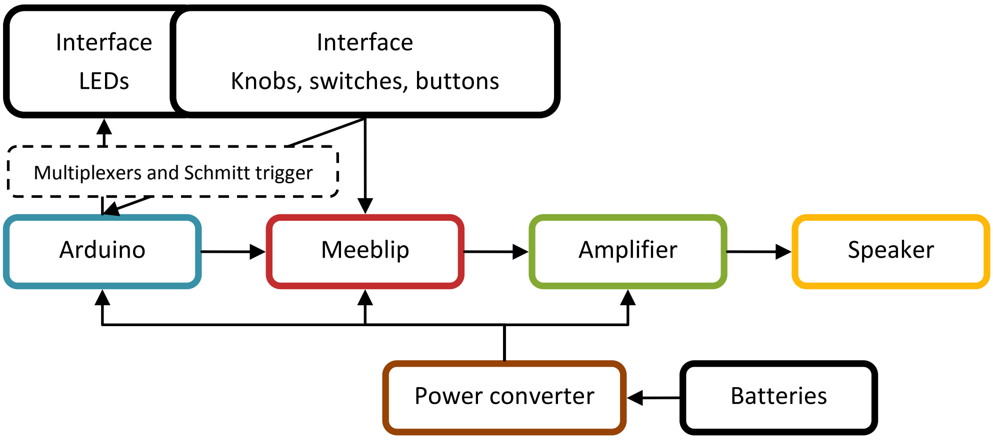


FIGURE 37. DIAGRAM OF ELECTRONIC STRUCTURE

5.6.1. POWER SUPPLY

The Arduino, Meeblip and amplifier all run on 5V internally, yet the Arduino and Meeblip both incorporate a voltage regulator to allow them to be powered on 9V, which is a more common voltage for power adapters. However rather than powering them separately on their own power supplies, it seems more logical to connect the power strips on all three devices. Arduino and Meeblip both use a low-dropout regulator[20]. While this is fine when using an external power adapter, this type of voltage regulator is inefficient compared to switching power supplies (Pietkiewicz, 1996). Downside switching regulators introduce more output noise, which is generally undesirable for an audio device. As this noise is above the frequency limit for human hearing, it should not induce practical problems.

Another consideration is the choice of battery. As the instrument would be used spontaneously, there is always the chance of a battery going dead while jamming. To soften this problem, commonly available batteries should be used. This way, in the case of battery failure, the instrument won't have to be plugged into a specific charger. The batteries could just be replaced. When using a 9V linear regulator, you would need either a 9v battery (€2-€10 apiece), or six 1.5V AA batteries(€0.25-€1.00 apiece), while research (Pietkiewicz, 1996) shows that using a step-up regulator with two 1.5V AA batteries show an increase of 58% operating time. So for a portable application like this, where battery life is very important, using two AA batteries with a step-regulator is the way to go. Rather than using the already present voltage regulators, a step-up module[21] is used.

5.6.2. MICROCONTROLLER

The Meeblip contains a microcontroller programmed in assembly. This is a programming language not known to the writer and too complicated to learn in the scope of this project. Additionally, the Meeblip might not have the processing power to perform additional tasks. So an additional microcontroller is used.

The Arduino prototyping platform provides the most interesting solution, although there are other possible options for microcontrollers. This platform provides a board including power supply and USB connectivity. However the definite reason for using Arduino is the programming language used for programming them. The Arduino programming language is made to be understandable by humans, as opposed to the usual assembly language, which is meant for the microprocessor. This saves a lot of time in prototyping while it doesn't diminishes any possibilities. Additionally, Arduino fits in the whole concept, as its Open Source Hardware itself.

---

[20] NCP1117ST50T3G and LM78L05 for the Arduino and Meeblip respectively
[21] http://www.prodctodc.com/dc-to-dc-15v-to-5152v-stepup-module-power-converter-for-mp3-mp4-phone-p-66.html#.UNhbB-TAfmc

5.6.3. Wiring

In wiring electronic prototypes, there are several non-exclusive ways to assemble all the electronic components as shown in order of complexity in Figure 38. *Breadboards* makes use of a board containing interconnected metal clamps. This has the advantage that it doesn't need soldering to connect the electronic parts. It is reusable and allows for quick iterations of prototyping. In *wire-to-wire*, you connect the wires directly to each other according to your scheme, while this is the most intuitive way, it often lead to problems caused by wires shorting each other. *Protoboards* are regularly perforated boards with conductive points or strips on them. These help overcome some of the problems of *wire-to-wire* as well as providing a stable underground for all the parts. As the previous three wiring types tend to get overcrowded and chaotic, it is often advisable to make a board with the wiring embedded in them; this is called a *printed circuit board* or *PCB*. These can be made either with low cost and low precision DIY techniques, or a custom made *PCB* can be ordered from a professional service, offering increased precision and multiple layers. Making a *PCB* is time consuming and relatively expensive. Additionally, it is hard to correct mistakes, making them most useful at the end stage of prototyping.

FIGURE 38. BREADBOARD, WIRE-TO-WIRE, PROTOBOARD, DIY PCB, PCB

As the wiring proved to be too complicated, as seen in Figure 40, to make on a breadboard, it was attempted to make use of *wire-to-wire* for prototype 1. After that proved unreliable, a *protoboard* was used for prototype 2. The first prototype was made without having the complete electronic plan ready, as components like the Schmitt Trigger (see chapter 5.6.6 Bouncing buttons) were not yet integrated. While this provided a proof-of-concept, problems were quick to arise. While adding new components, old connection broke or shorted, causing more repair time than actually creating new feature. While a PCB would have been preferred for the second prototype, the cost, lead time and most importantly the learning curve for proper PCB design proved too big, so the choice fell on a *protoboard*.

An interesting property of the Arduino was used to make the design more compact. An Arduino's often use so called *shields*. These are circuit boards with pins that can be put on top of an Arduino. Mimicking this, the protoboard has protruding pins, as seen on the left in Figure 39, on which the Arduino can be mounted.

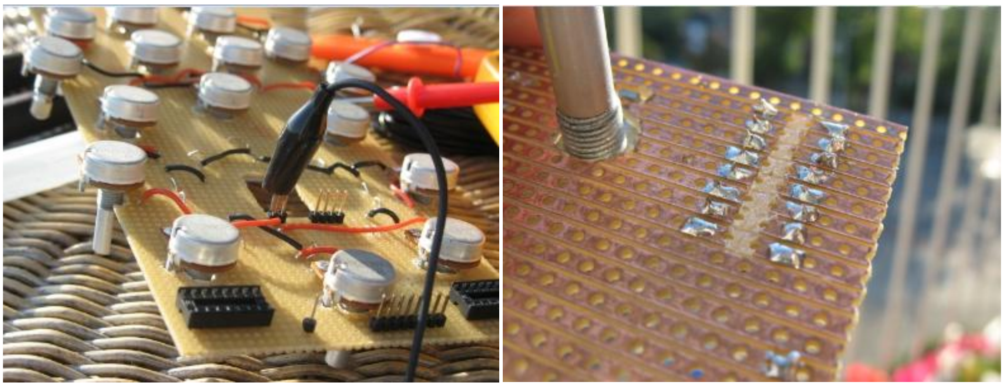


FIGURE 39. BACK AND FRONT SIDE OF THE STRIPBOARD

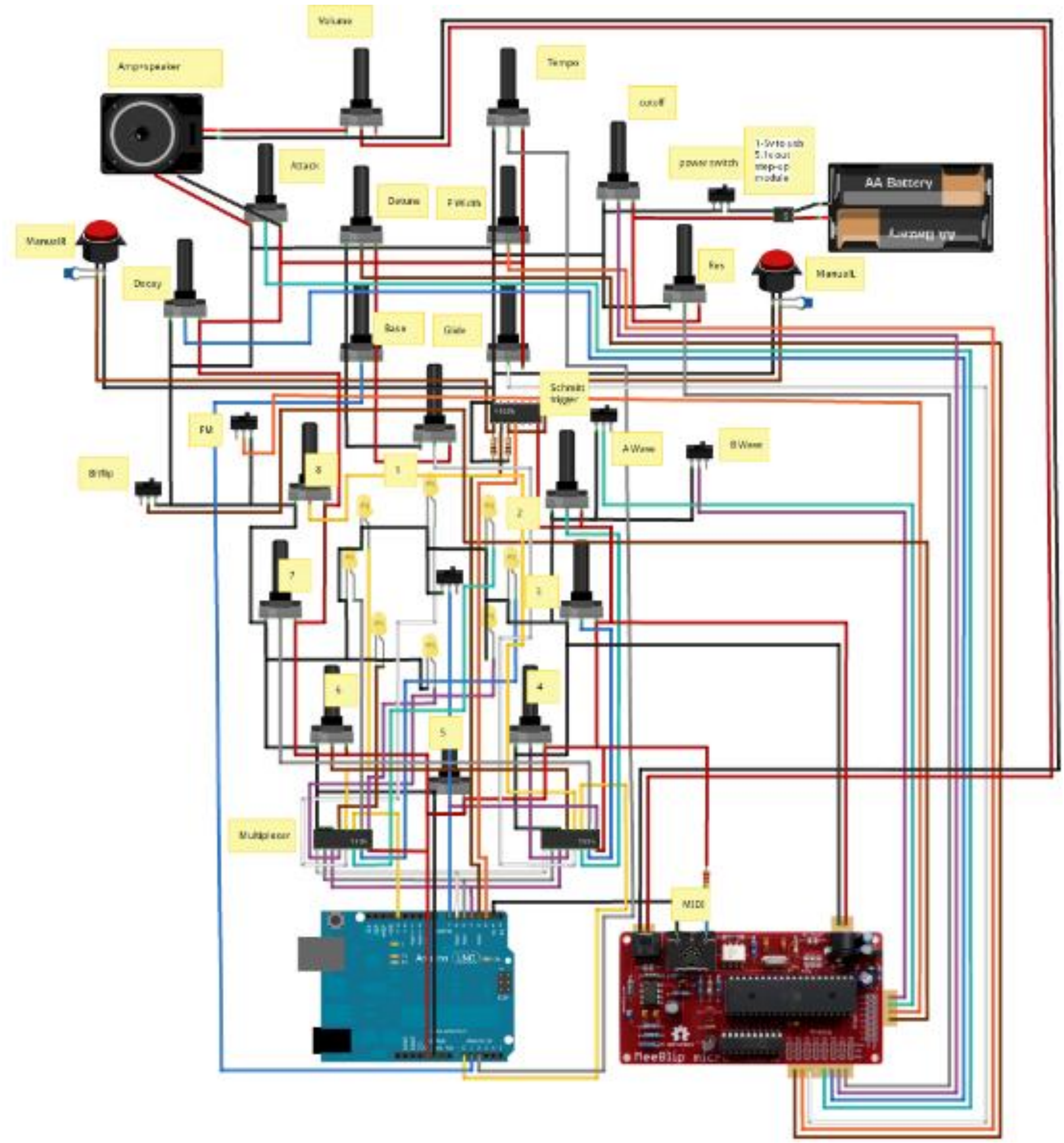


FIGURE 40. THE FINAL WIRING

5.6.4. AMPLIFIER/SPEAKER

As been said before, the quality of sound from a speaker system is a complex interplay between the input, speaker, the amplifier and the enclosure design. To find the right type of speaker and amplifier could be a whole research topic in itself. Therefore it was chosen to reap the fruits of mass production.

USB powered speakers are available at a very low cost and it can be assumed the design team has spent effort to find the right speaker-amplifier interplay. Another advantage is that, just like the Arduino and Meeblip, it is powered on 5V, diminishing power conversion problems. Considering space constraints, only the smallest type of speakers could be used, with a maximum diameter of 6 cm. From a comparison of available types, it became apparent that nearly all low-cost Chinese manufactured USB speakers used the same type of aluminum diaphragm, suggesting these speakers were all the same. (With an increase in price correlating with improved aesthetics.) Following these considerations a €10 no-brand USB speaker was purchased and disassembled for use. As it was a stereo speaker, only one of the channels was used.

FIGURE 41. THE USB SPEAKER USED FOR THE PROTOTYPE

5.6.5. DYNAMIC BUTTONS

The sequencer is controlled by two buttons that determine the timing as well as the dynamics. Sadly, no suitable prefab buttons were found and a custom solution had to be found.

A musical note is commonly expressed in three dimensions, pitch, rhythm, and dynamic. The first is controlled by setting the sequencer, the second can be controlled by the rhythm one plays on the knobs, and the dynamic or loudness of a note should be controlled by how hard the user pushes the buttons.

There are several ways to make a button dynamically sensitive. Velocity sensitivity is the way the keys of MIDI keyboards and synthesizers work. It works by measuring the time between when the button is touched and when the button reaches its lowest position. The smaller the time, the faster the button is pressed and bigger the expression.

Pressure sensitivity is a method used in some drum controllers[22,23] and is usually associated with *pad* style buttons. The most common way to do this is to use a semi-conductive material, which changes resistance based on the pressure it receives. This material is sandwiched between two layers of conductive material. When the sandwich is pressed, the resistivity changes, which can be measured by a microcontroller, which in turn determines how hard the button is pressed.

Capacitive sensing works by the phenomenon that if a finger touches a circuit, it becomes part of it. So by measuring the change of capacitance when a conductive surface is touched, a button press can be determined. Using more accurate measurements, the amount of force used when touching can be calculated as well, though this is a complex thing to get working on an Arduino.

A last method is the use of a vibration sensor like a piezoelectric element to sense how an instrument is touched. Though sensing a state change is possible, it is hard to measure actual dynamic expression. Additionally, making two separate buttons is hard, because the sensors measure all vibration, not just that near them.

---

[22] http://www.akaipro.com/mpc
[23] http://www.native-instruments.com/#/products/producer/maschine/

Various tests were done to find the right method, however none were found. A modified arcade button did function like a velocity sensor, but did not prove accurate enough for music use. Figure 42 shows some of this setup. Commonly available pressure sensors were tried, but these weren't accurate enough either to allow musical expression. Small tests were done with capacitive sensing and piezoelectric elements, but these proved even less useful than two previous methods.

As the dynamic sensing of the buttons did not produce any satisfactory results, it was chosen to disregard this feature for the prototype and just use momentary switches; in this case arcade buttons. This way, at least the timing could be tightly controlled.

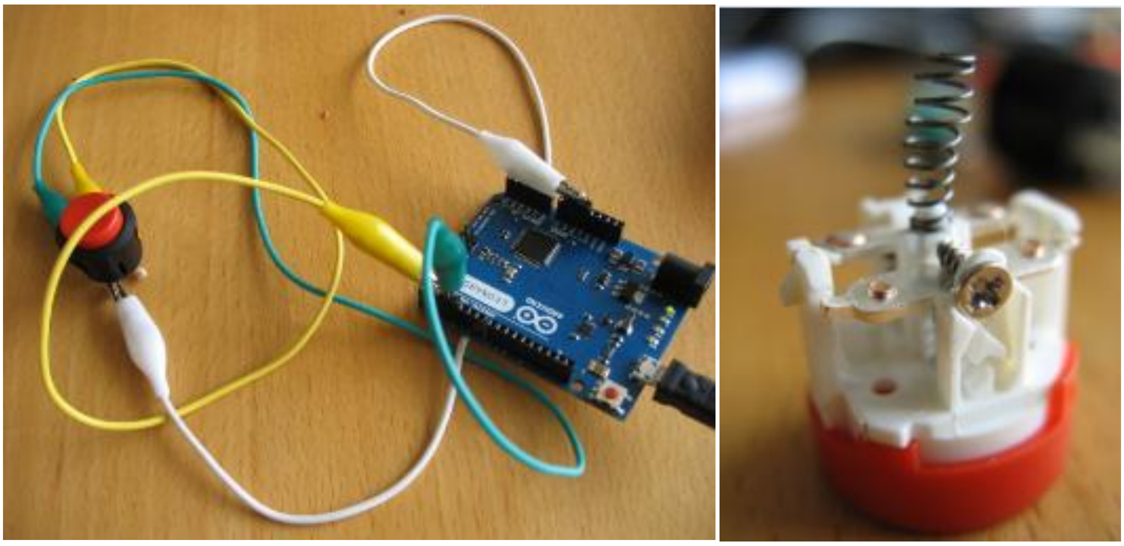


FIGURE 42. ATTEMPT MADE AT VELOCITY SENSITIVE ARCADE BUTTON

5.6.6. BOUNCING BUTTONS

A commonly occurring problem with using momentary switches to control digital logic is bounce. This is the effect that a physical button does not close immediately when pressed, but bounces for a small moment of time. This bouncing is similar to how a ball bounces when dropped to the ground, although it happens only for a small time; between 1 and 10ms. 10ms is just the desired latency between action and reaction for a musical instrument. 30 ms between haptic and visual are acceptable and 60ms might not even induce errors in playing(Marshall & Wanderley, 2006). While smaller latency is better, consistency is even more important.

There are ways to keep a button from causing trouble by the bouncing. The first is to make a button that does not bounce. The second is using an electronic component to avoid or reduce the bounce and the last way is to filter the bounce in the software, both called debouncing.

As the software solution introduces extra latency and the hardware solution was said to be the most robust, the second option was chosen. This solution, as proposed by *all electric kitchen*[24], makes use of an inverting Schmitt trigger to keep the signal either high or low, thereby eliminating bounce.

5.6.7. MULTIPLEXERS

As the Arduino does not have enough IO pins to read the sequencer as well as making the lights blink, multiplexers were used. These chips allow a single signal to be send or read from either other point. The common, reliable and inexpensive 4051 IC chips were used.

---

[24] http://www.all-electric.com/schematic/debounce.htm

5.6.8. Delay

Initially, a delay or reverb unit was intended to be added to the prototype as to give it a fuller sound. Through the scheme of the Korg Monotron Delay[25] it was discovered that the PT2399 echo audio processor IC would allow for a very inexpensive (<€4) solution. A prototype was built on a breadboard, see Figure 43. While this functioned as intended, the process of building it was quite complex and time consuming and transferring it to a circuit board would impose additional problems. As the delay/reverb is not essential for the design, merely a bonus, it was decided to disregard it for the prototype.

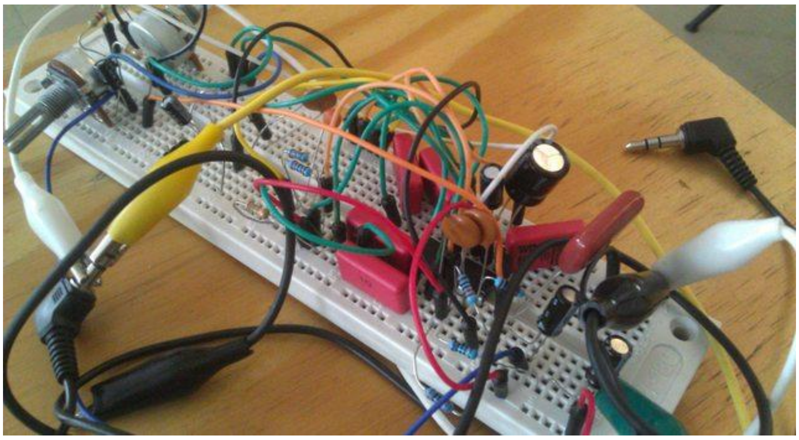


Figure 43. Breadboard prototype of delay

5.7. Firmware design

One of the main advantages of using an Arduino, is that it is programmable using the Arduino programming language, which is a lot more comprehensible than alternatives like C and assembly. Another advantage is that it can be both programmed and powered from an USB connection, allowing quick firmware iterations.

Starting at the hardware sequencer described at chapter 5.3.2.7, the firmware was gradually expanded to slowly form its final iteration. A major consideration was whether to use timer interrupts or timer checks.

While the full code is printed in APPENDIX A, Figure 44 shows a schematic representation of it. Black arrows are electronic signals, green arrows are digital information and blue arrows are function calls. When the instruments starts up, the *Setup()* function is called, which initializes all functions needed for the operation of the interface as well as using *MIDIcc()* to make the Meeblip take on the desired initial settings. *AttachInterrupt()* is a function that is used for the dynamic buttons. It causes that when a dynamic button is pressed, all sequencer processes are stopped and reset to fit with the timing of the button press. The *Buttoncheck Loop()* and *Readnot()* handle the way the sequencer works and the notes are read from the knobs. Finally, the note information is converted by *Noteon()* to MIDI data and sent to the output of the Arduino and into the Meeblip.

[25] http://www.korg-datastorage.jp/Manual/monotron%20DELAY_sch.pdf

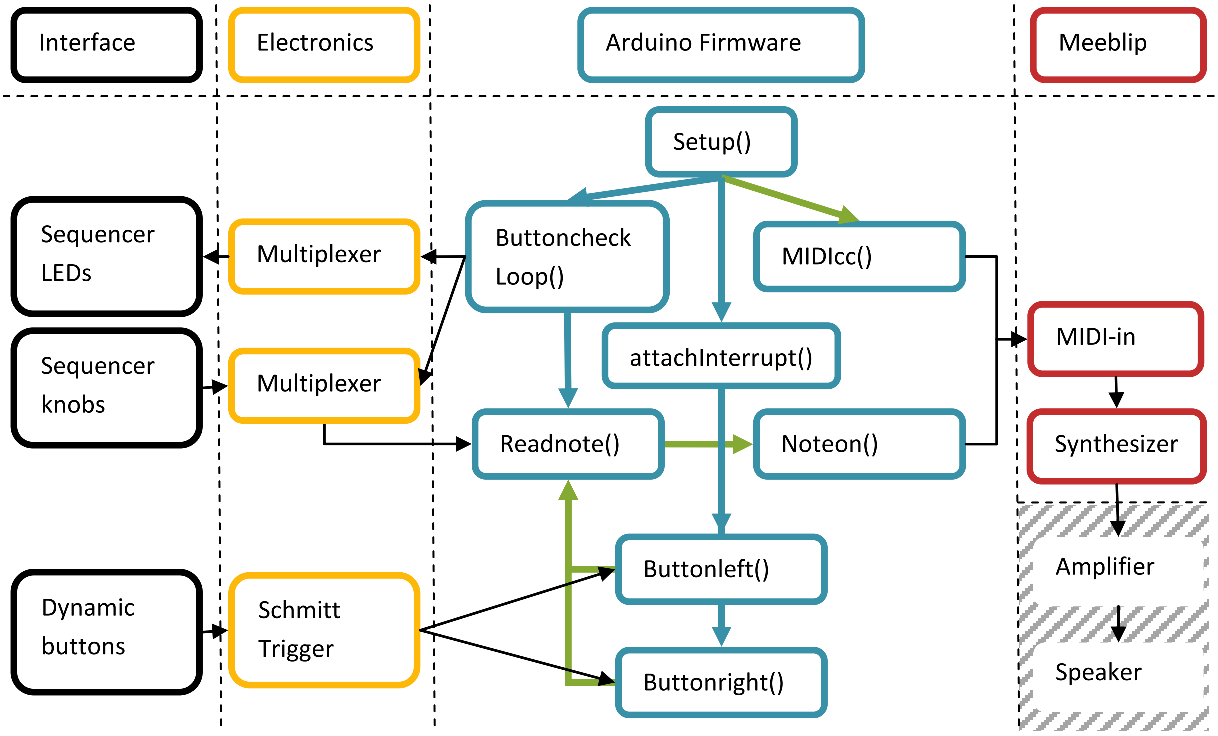


FIGURE 44. DIAGRAM OF THE ELECTRONICS AND FIRMWARE ARCHITECTURE

5.8. OTHER CONSIDERATIONS

It is important for designers to think about the environmental impact their product makes. While the effects of a product on the environment more obviously present themselves in mass produced products, it is best to start working towards an eco-friendly product right from the start. While nearly none of the materials used are recyclable, they are very much reusable. The Arduino can be pulled from the circuit board, the ICs from their sockets, the amplifier and Meeblip can be removed without hassle. This makes it easy to replace components when they break as well as allow the creation of a new instrument with the same parts.

While a real marketing plan and in particular a market penetration strategy is outside the scope of this design project, some thoughts were gathered on how to go about doing it. The most interesting way would be to take inspiration from Lomography (see chapter 3.4.1) and make viral videos of a group of people jamming, with this instrument as a centerpiece. Before getting to that point, there should be the capacity to make a few hundred devices, so a small part of the demand could be satisfied. A good marketing approach would be to sell the first batch of instruments to known or at least proficient musicians for a reduced price, with the request that they make an online video with it.

To keep things simple, the device is hold together by a rubber band as seen in Figure 45. While this provides for easy maintenance, it is not durable. A next iteration should have a hinge or other system which would allow for opening when needed, but keeps a high structural integrity when closed.

A similar note is on portability. The device is small and light enough to fit in most bags. The trapezoidal shape gives is excellent properties on bending. The knobs and plywood is strong enough to withstand some general misuse, though the plywood would be hard to repair if damaged. While outside the scope of this project, it would be suggestible to make a custom bag that would protect the instrument from travel damage.

The final prototype has cost about €200 in total. This price could go down a bit if a larger series would be made, so PCBs and parts could be ordered in bulk. With added overhead costs of 500%, this could mean the product would be sold for about €800 if it were handmade. This is a reasonable price for a custom made instrument which has proven its use. However for something that is made for a novel usage scenario, it may be too expensive. So if the instrument is to be sold, it is recommended that some research is done into making it more cost efficient.

FIGURE 45. HOW A RUBBER BAND HOLD THE DEVICE TOGETHER

6. PROTOTYPE

As a proof-of-concept as well as part of the practice-based design project, two prototypes were made. The manufacturing of these prototypes were done parallel to, although for the end of the design phases. This way, the impact of certain considerations could quickly be tested. The creation of the final prototype has been described in an Instructable in APPENDIX E, this document allows users to reproduce the instrument.

6.1. VERSION 1

As the instrument was supposed to be easily reproducible, it was attempted to create the first prototype using the common tools. As material, a nicely grained piece of 6mm plywood was chosen, for its stiffness, lightness and cost. Measurements were taken from the 3d model as to make the prototype in the planned dimensions. After the casing was finished, the interface was drawn on the top panel and holes were drilled for the potentiometers, LEDs and switches. While fitting the switches, it became apparent something was wrong; the potentiometers have to be fastened by a small nut on top of the panel, however the thread this nut screws on only protruded 4mm. To allow the potentiometers to be mounted anyway, a countersink was used to remove material from the top panel, and allow the nut to be screwed on. Due to the inexperienced handwork, the final casing did not meet the standards and might even be considered sloppy. For a next prototype, it was decided, a computer-controlled manufacturing method should be used, be it laser-cutting or CNC milling. To test whether or not the designed interface would work as intended, it was decided to make this prototype functional.

FIGURE 46. PROTOTYPE 1 ASSEMBLED

After the casing was manufactured, a start was made with wiring. Keeping Cook's (2001) 20th principle (*Wire and document for future surgeries*) in mind, it was attempted to make the wiring as flexible as possible. This was done by using: a breadboard on the Arduino, where the multiplexer would be mounted; header[26] connections on the Meeblip; and power supply and wire-to-wire connections on the backside of the interface panel. The wires coming from the potentiometers and switches, could then be plugged into the headers and breadboard. While this method initially seemed very orderly due to color coding of the wires, the complexity quickly rose. When everything was wired up, it became apparent that this type of wiring is very fragile, often, when opening or closing the casing, several wires would either lose their connection, or short to other cables. While the software could be written in this prototype, it became very tedious to track faults and repair them. The cable mess can be seen in Figure 47.

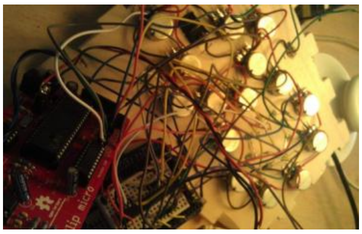


FIGURE 47. THE CABLE MESS OF PROTOTYPE 1

---

[26] A type of connector, made to clamp a wire

Even though Cook (2009) says "*Build a (new) copy, don't trash the original,*" it was decided to remove the components for reuse, as keeping a not functional device, is barely useful for later reference. A few recordings were made to have a reference anyhow. Figure 46 shows prototype 1 in its final state.

6.2. VERSION 2

The first decision of the second prototype was the choice on which manufacturing technique to use. The workshop the prototypes were made in had a simple 2d CNC wood router available. However after a few tests, both the software and hardware did not prove to be reliable enough. An alternative was sought in laser-cutting. While there are a lot of laser-cutting facilities in Berlin, none of them could cut through the 6mm plywood that was used in prototype 1. An alternative material had to be found that was under 2mm. Most types of 2mm plywood or thinner proved to be quite flimsy; a suitable material was found with high quality birch aircraft plywood. This was four times more expensive than the previously used plywood, but it not only possessed just the right amount of stiffness, while being thin enough, it also featured a more beautiful grain. Due to this grain it was chosen not to finish it. The plates were cut as seen in Figure 48 and assembled using high strength wood glue.

FIGURE 48. LASER CUT PARTS, READY FOR ASSEMBLY

The wiring of the second prototype was done using a *protoboard* as described in chapter 5. While this made wiring a lot more straightforward, it still wasn't perfect. A lot of time was spent rechecking connections and fixing them. Though through making this prototype, the wiring scheme was proven successful and a PCB could be made relatively easy.

FIGURE 49. PROTOTYPE 2

7. Evaluation

The prototype was evaluated using two methods, the first was an user evaluation and the second an informal evaluation by the designer and the Meeblip company. As the work was done as described in this document should be considered as a first iteration in the iterative design process, the analysis is meant to lead to recommendations towards a next prototype, rather than as an assessment of quality or marketability.

As seen in the analysis in the third chapter, the evaluation of the interfaces of musical instruments is more ambiguous than those of functional devices. This is especially well illustrated in the example of the TB-303, seen at chapter 3.3.7, where the actual usage of the device is completely different from what the designers had intended, and what they would not have evaluated. This means the evaluation has been executed in an open-ended manner, loosely based on the guidelines by Stone et al. (2005).

The best evaluation would have been to give the prototype to an experienced musician, to have him or her make a song with it. Sadly, due to problems with the prototype and time constraints, this was outside the scope of the document.

7.1. Approach

While extensive user testing on various user-groups would have been desirable, the huge time constraints meant only a condensed variation of this could be performed. It was chosen to limit both the user observation and the designers evaluation to a single session. As resources did not allow a second observer or video documentation, it was chosen to record only qualitative data. Notes would be made on both the remarks by the users as well as the aspects described below.

The location chosen for the user evaluation was the Music Makers event[27] in Berlin. This event gathered a number of expert musicians, DIY enthusiasts as well as developers of electronic musical instruments (domain experts). The event was set up as an informal exposition where other novel instruments were presented as well. Visitors with an interest in the device were asked to play around with it, and to think-aloud, telling what he or she was trying to achieve. While the user was playing the instrument, instructions would be given on the features, allowing a gradual exploration of the device. This was chosen over using cognitive walkthrough methods, which might have caused the user to feel tested. As there were no resources for a second evaluator, all had to be done by the author himself, functioning in the roles of observer, note-taker as well as facilitator. The actual number of test subjects was eight.

The informal evaluation by the designer and Meeblip company has been done at the workshop where the prototype has been built. The instrument was examined and played, which was cross-referenced with the design brief and PDS.

Other than noting imminent qualitative descriptions, special attention has been given to the following items:

- Usability requirements of Quesenbery (2003): effectiveness, efficiency, engagingness, error tolerance, ease of learning;
- Affordances of Armstrong (2006): Situated; Multimodal; Engaging; Sense of embodiment;
- Modes of interaction by Johnston (2011): Instrumental mode; Ornamental mode; Conversational mode;
- Level of seriousness: Quality of sound; Resemblance to a toy.

[27] http://musicmake.rs/berlin-2012/

7.2. Analysis

The two evaluations led to a list of remarks, which was distilled into practical recommendations, listed in chapter 9.

1) User evaluation:

- The chromatic note choice led to unmusical sequences. It would be desirable to have a choice which scale you can use (chromatic, pentatonic, etc.).
- A problem during testing was the fact that a firmware bug on the Meeblip caused the volume to fluctuate as well as making the envelope knobs unresponsive.
- The left dynamic button was thought to make the sequencer turn counter-clockwise when held while as the sequencer is running (this should be implemented in a next prototype).
- Users expressed joy on the aesthetic quality of the interface.
- Users expressed joy in interacting with the device, while most started interacting with a feeling of confusion. This quickly turned over to interest, suggesting that the instrument is easy to learn.
- Users found the concept interesting and would like to try using it in a setting like the usage scenario.
- The prototype seems sensitive, which might even increase the value of the product.
- The users did not all directly feel the right playing position of the instrument; some put it on top of their lap, instead of inside it, while others tried to play it like a kalimba as seen in the image below.It could be positive to improve the ergonomics such that the device is pleasant to use in both positions.
- When it was explained that the dynamic buttons should have been sensitive to the dynamic expression of the user, the users generally agree that this would greatly increase the usefulness of the instrument.
- Users did not directly understand what the function was of each button or that there was a distinction between the sequencer and the synthesizer interface. It might be helpful to make the knobs distinctive, although when the user was explained what the function was of each part, no big mistakes were made.
- Due to the delayed response of setting a note and hearing it, interaction did not always seem effective. A great suggestion was that when the sequencer wasn't running and, at the same time, one of the sequencer knobs would be turned, that the change in note could be heard.
- The efficiency of the interface was hard to test, as its normal for musical instruments to have a learning curve. This could be further assessed in personal sessions where a user would be instructed to make a song.
- The instrument defiantly seemed to engage the users, as they often stopped communication with the evaluator in favor of playing with the instrument.
- As no user got to the point of making actual music, no definite conclusion could be made on the affordances of Armstrong (2006) nor could an analysis of the mode of interaction (Johnston, 2011) be established.
- In response to the question whether the instrument was a toy, gadget or actual serious instrument, the consensus pointed toward being a serious instrument.

Figure 50. Playing position of a kalimba

2) Informal evaluation:

- The dynamic buttons were sometimes still bouncy; when pressed hard it would step twice in the sequencer.
- The fact that the instrument has to be fastened with a rubber band is not ideal, preferably some kind of hinge should be made.
- Sound seems loud enough, sometimes even too loud.
- The unfinished wooden casing is sensitive to dirt; it should be treated.
- The wood used in the second prototype is very beautiful, it instantly makes the instrument look more valuable.
- The prototype fits into most types of bags and is light enough to carry around.
- Battery time is about three hours, which seems too short for general use.
- The batteries are not easily accessible, making changing them quite a hassle.
- The size and angles of the instrument seem well chosen, though the sharp corners might give problems.
- The missing touch sensitivity of the dynamic buttons is a real drawback to the prototype.
- As all functions are directly under a knob or button, the instrument is really error tolerant; even if an error is made, reversing the action leading to that error fixes it.
- While it was not possible to assess the qualitative specifications from the PDS, all other seem to have been met.
- The markings around the knobs do not correspond with the actual effect. This could be improved in further prototyping. In particular, this is desirable at the sequencer, base and tempo knobs.

7.3. PDS EVALUATIONS

Reflecting back on the initial conditions, the final product is checked with the PDS. A √ indicates an achieved specification, while a ○ indicates a specification which has not been met.

- ✓ The design needs to contain a Meeblip Micro
- ✓ The design needs to be replicable by amateurs
- ✓ Materials need to be locally source-able
- ✓ Manufacturing methods commonly available
- o Documentation should be comprehensible for amateurs
  - o *Is yet to be done*
- ✓ The design needs to be customizable by the builder
- ✓ The instruments needs to be loud enough to compete with acoustic guitars and hand drums
- ✓ The design needs to operate in both a low-tech setting
- ✓ The design needs to look as a serious musical instrument (as opposed to a toy)
- ✓ The design needs to sound like a serious musical instrument (as opposed to a toy)
- ✓ The design needs to look attractive to the user group
- ✓ The design needs to be of such sizes that it is usable by 95% of the population
- ✓ The text on the instrument needs to clearly legible and understandable
- o The device needs to withstand at least two years of normal use in performance situations
  - o *Is yet to be tested*
- o The user should be able to play the instrument for at least 30 minutes without fatigue
  - o *Is yet to be tested*
- o The instrument should, over time, become blindly playable, as to allow interaction between musicians
  - o *Is yet to be tested*
- ✓ The instrument needs to adhere to the criteria of Armstrong (2006): *Situated*; *Timely*; *Multimodal*; *Engaging*; *Sense of embodiment*.

- ✓ The instrument should allow a *conversational* mode of interaction (Johnston, 2011).
    - o *Is yet to be tested*
- ✓ The device need to survive a fall of one meter height
    - o *Is yet to be tested*
- ✓ Electrical components should be designed for a minimum of noise
- ✓ The design should not be focused on a single musical style
- ✓ The design needs to be documented in open source
- ✓ The design needs to be made within eight weeks

From the items that can be verified from the PDS without extensive (user) testing, the proposed design appears to fit the initial product design specification. Time will have to learn whether the instrument, or even the scenario, is interesting enough. However, it seems fair to state that the proposed design meets the requirements set at the start of the project.

8. CONCLUSIONS

Through an extensive analysis and an instructive design process, a highly satisfactory design has been made. A working prototype has been manufactured together with instructions for fellow researchers. While resources and time constraints prevented the prototype to be tested in the actual usage scenario, an evaluation was made with domain-experts.

Starting from an open ended design brief (chapter 2), the mere constraints of containing a Meeblip and being interesting for the current Meeblip user group, an analysis had been made of all aspects concerning the Meeblip, general synthesizer design, open source and three case studies that inspired the Meeblip.

Following the design brief, chapter 3 contains an extensive analysis. The three available packages of the Meeblip were described in chapter 3.1, as well as the difference between the two design iterations. An informal evaluation has been performed, indicating a few points that seem to define the Meeblip. The company has been profiled together with some remarks on the process of designing the Meeblip. An analysis of online videos and forum posts led to a categorization of Meeblip users in three groups; hackers, starters and synth pros. Following this, a selection of competing products has been analyzed, leading to an impression of the Meeblip's market.

As the Meeblip is an open source product, this design project is performed under an open source license as well. Details surrounding this subject are described in chapter 3.2. To get a sense of the field of design for electronic musical instruments, an analysis has been made on existing literature as described in chapter 3. These lead to general design guidelines as well as an impression on how to improve on current design form factors. Three cases have been studied in chapter 3.4 to get a view on how Meeblip places itself in a larger spectrum.

The forming of the concept is lined out in chapter 4. Following the analysis, a more specific user and scenario have been chosen as described in chapter 4.1. The goal is to design an instrument that allows for the creation of electronic music in a normally acoustic environment. Key considerations are to make the instrument self-contained, sound good, allow for rich interaction and to be replicable by DIY enthusiasts. The requirements are formed into a product design specification in chapter 4.2. Both existing and conceptualized products are found that fit the PDS on different levels (chapter 4.3), before finding a satisfactory concept, ready for further development.

Chapter 5 is the documentation of the design process. Firstly, the concept is detailed, after which the considerations in Encasing design Interaction design, Type of knobs, Encasing design, Acoustics design, Electronic design, and Other considerations are described in chapters 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8 respectively.

The creation of the prototype is described in chapter 6, starting from the planning and going through two different iterations, before ending up with a fully functional prototype. This prototype and design is evaluated in chapter 7 through an informal evaluation as well as a user evaluation. A list of remarks has been distilled into concise points of interest, which flowed through into the recommendations in chapter 9.

The designer, company and users felt that a very interesting and novel design has been produced in a very limited amount of time. As instrument design is an iterative process, this report is not the end of the project, but a review of the first iteration.

The future will learn if there is an actual need for the concept presented, but even if not, this project has been an inspiration for Meeblip users, to bring their devices to another level.

9. RECOMMENDATIONS

As been said before, the prototype that came from this research project is just a first iteration of an ongoing design process. To support producing a new iteration and allowing peer researchers to continue my work, it is open source after all, a list of recommendations has been made.

The working method, produced at the start of this design project, prescribed clear phases of research, design, prototyping and evaluation. In practice, these cannot be separated to easily; while prototyping, you'll want to make changes in the fundamentals of the design and while designing, you'll want to make parts of the prototype to check for feasibility. It is advisable, for further design projects, that a planning would not be made in separated design phases, but in deliverables, working up to an end product.

This design project is entirely done by the author. While the distance between Germany and Canada is big, it would've been beneficial to have more feedback from Reflex Audio in the form of internet conferences. Another thing that was attempted, but not found, was getting information on speaker enclosure design.

The design presented, is positive for the Meeblip community, as it tries to make a leap in synthesizer design itself. Hopefully it will inspire other users to do wild projects. While publicity is yet to come, it will be interesting to see what aspects of this design catch the attention of the public. Will it be the novel use scenario, or rather the technology inside the design?

For the next iteration of this design, the focus will most likely be in making a more durable prototype. While the current prototype functions, it is fragile. Another important aspect is the dynamic buttons, more experiments should be done to make these function. As the electronics are working as needed, the creation of a PCB for the next prototype is relatively straightforward and very advisable. Adding the amplifier and mounting points for both the Arduino and Meeblip would be advisable. Very important is finding a way to open and close the device without a chance of damaging the electronics; this should also remove the need for a rubber band. Other issues are fine-tuning the functioning of certain knobs and buttons, and tweaking the acoustics. For further iterations, it will be interesting to experiment with different usage positions. To make the device completely self-sufficient, experiments could be done with solar power or hand-cranked dynamos. The addition of a delay or reverb should still be considered.

There is the possibility to make the instrument syncable to other electronic musical instrument through MIDI clock. This would make the instrument more interesting to users and would add relatively little complexity.

While an evaluation with domain experts led to a wealth of information, testing should also be continued in the actual usage scenario. More prototypes could be made and handed out to at least three different musicians, let them play with the instrument for a few weeks and see what music they come up with and what they think of the device.

As the presented design is not the only product that would fit the scenario, different fitting products could be made. For instance a similar object could be made, but instead of a sequencer, it would be controlled by a touch-sensitive surface. Perhaps a percussion-sound synthesizer could be made, complementing the device.

To finish the recommendations, it should be noted that this design process has been focused purely on producing a novel design and getting a proof-of-concept. To make the most out of this design, predominantly marketing, financial and manufacturing aspects will have to be considered, before releasing it to the public at large. As of now, the plans will be put on the internet for those interested and hopefully they will start making further iterations.

10. Table of Abbreviations

A/D	Analog/Digital
CDM	Create Digital Music (.com)
DIY	Do It Yourself
HCI	Human Computer Interfaces
IC	Integrated Circuit
IO	Input/Output
LED	Light Emitting Diode
MIDI	Musical Instrument Digital Interface
ms	Milliseconds
PCB	Printed Circuit Board
PDS	Product Design Specification
USB	Universal Serial Bus

11. Tables

12. FIGURES

13. References

Armstrong, N. (2006). *An Enactive Approach to Digital Musical Instrument Design.* Princeton University.

Cook, P. (2001). Principles for Designing Computer Music Controllers. *NIME-01.* Seattle, USA.

Cook, P. (2009). Re-Designing Principles for Computer Music Controllers: a Case Study of SqueezeVox Maggie. *NIME-09.* Pittsburgh, Pennsylvania, USA.

DINED. (2004). Opgeroepen op 1 7, 2013, van P95 Dutch mixed 20-30 years, 46. forefinger breadth: http://dined.io.tudelft.nl/dined/

DINED. (2004). Opgeroepen op 1 7, 2013, van P5 Dutch mixed 20-30 years, 33. buttock-knee depth minus 30. abdominal depth with correlation 0.2: http://dined.io.tudelft.nl/dined/

Duignan, M., Noble, J., & Biddle, R. (2005). A TAXONOMY OF SEQUENCER USER-INTERFACES. *ICMC.* Wellington, New Zealand: Victoria University of Wellington.

Hacker, R. (1991). *Patentnr. 5,065,661.* United States of America.

Hazel, P. (1992). SYNTHESISERS; Interface Design & Development.

Ishii, H., & Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *CHI '97.*

Jeppesen, L. B., & Frederiksen, L. (2006). Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer-Controlled Music Instruments. *OrganizationScience* , pp. 45–63.

Johnston, A. (2011). Beyond Evaluation: Linking Practice and Theory in New Musical Interface Design. *NIME-2011.* Oslo, Norway.

Marshall, M. T. (2008, April). Physical Interface Design for Digital Instruments. Montreal, QC, Canada: McGill University.

Marshall, M. T., & Wanderley, M. M. (2006). Vibrotactile feedback in digital musical instruments. *NIME* , 1-5.

Minieri, N. (2009, December 7). *Hi-Res EQ Chart*. Opgeroepen op 1 7, 2013, van BBD: http://www.beantownboogiedown.com/beantown-boogie-down-beat-bo/2009/12/7/beat-box-hi-res-eq-chart-for-instruments-chords-and-more.html

Pietkiewicz, S. (1996, Februari 7). *2 AA Cells Replace 9V Battery, Extend Operating Life.* Opgeroepen op September 14, 2012, van Linear Technology: http://cds.linear.com/docs/Design%20Note/dn63.pdf

Quesenbery, W. (2003). *The Five Dimensions of Usability.* Mahwah, NJ: Lawrence Erlbaum Associates.

Saego, A., Holland, S., & Mulholland, P. (2004). *A CRITICAL ANALYSIS OF SYNTHESIZER USER INTERFACES FOR TIMBRE.* Milton Keynes: The Open University, Faculty of Mathematics and Computing.

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User Interface Design and Evaluation.* San Fransisco: Morgan Kaufmann.

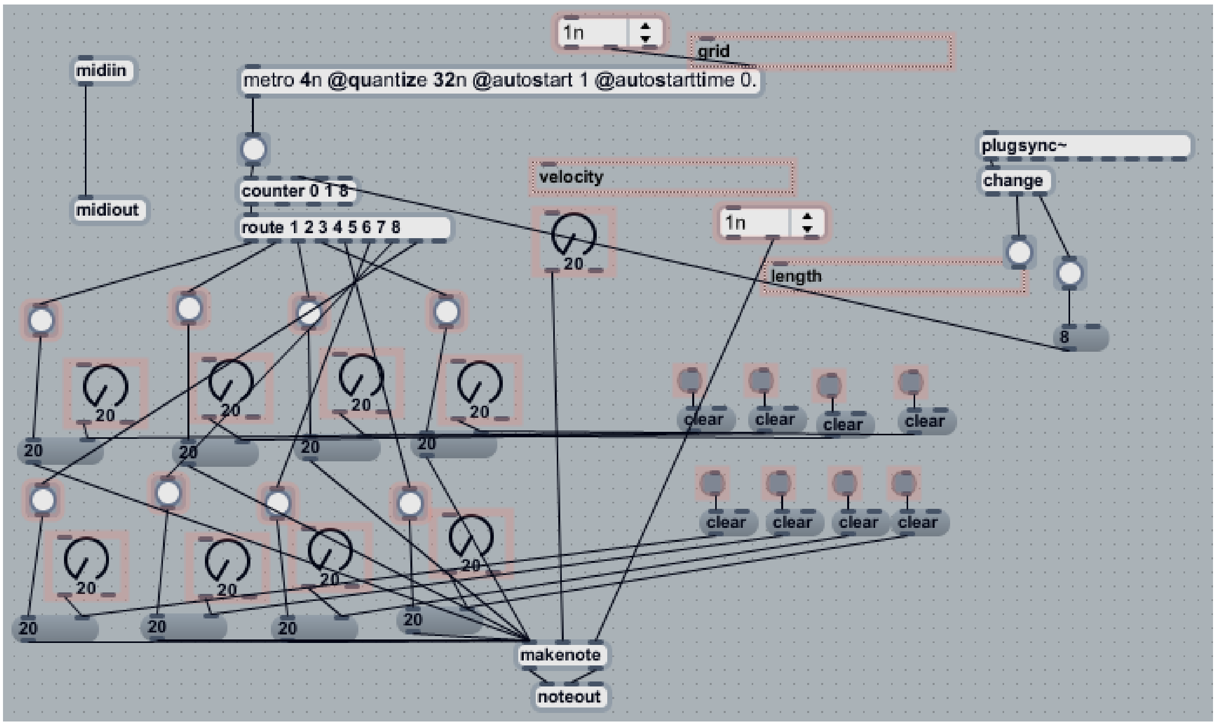
Vallis, O., Hochenbaum, J., & Kapur, A. (2010). A Shift Towards Iterative and Open-Source Design for Musical Interfaces. *Conference on New Interfaces for Musical Expression (NIME 2010)*, (pp. 1-6). Sydney, Australia.

Wolfe, J. (2006). *Helmholtz Resonance* . Opgeroepen op 1 7, 2013, van UNSW Music Acoustics: http://www.phys.unsw.edu.au/jw/Helmholtz.html

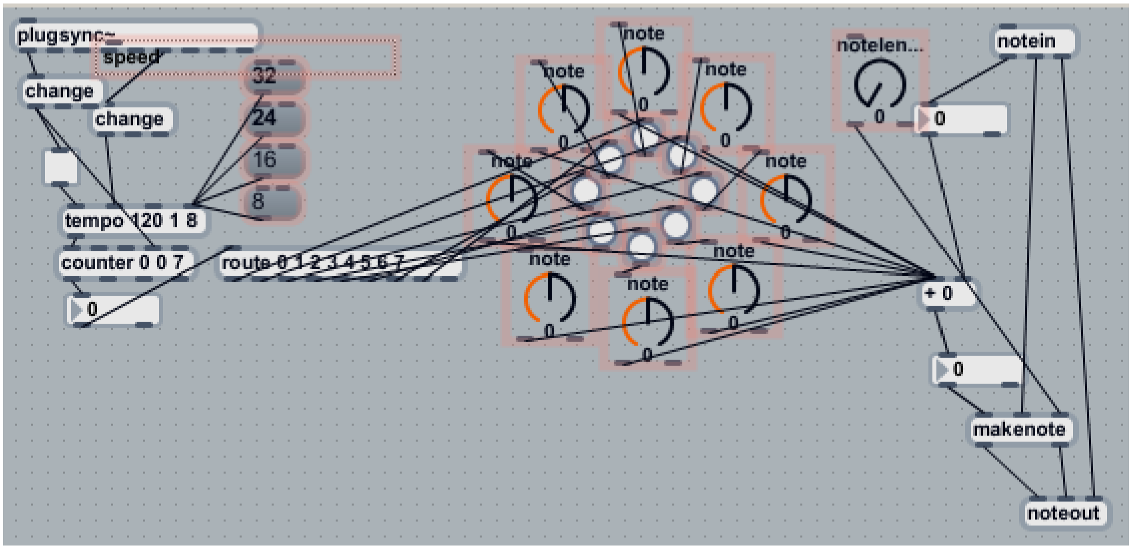
APPENDIX A. CODE

1. MAX EXPERIMENTS

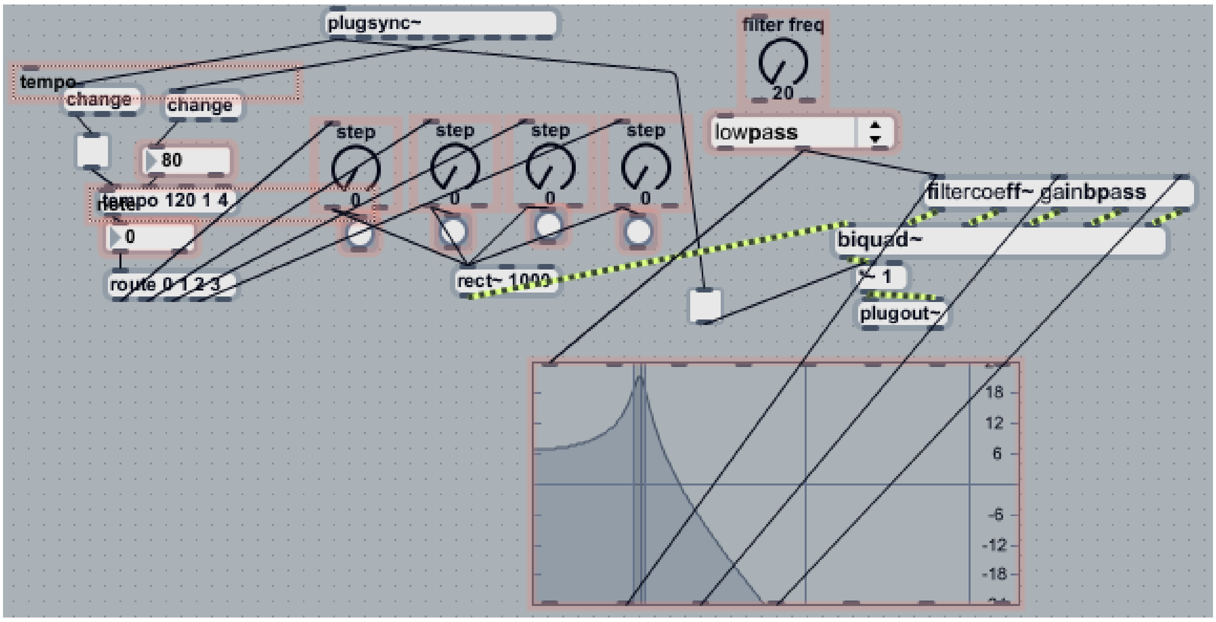
1.1. SIMPLE EIGHT STEP SEQUENCER



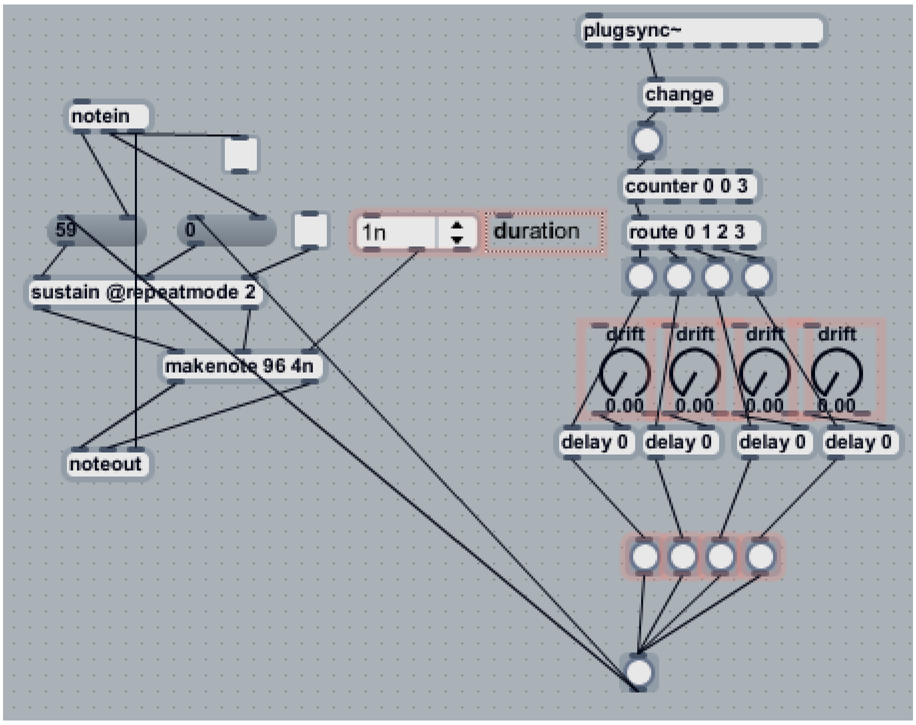
1.2. CIRCULAR ARPEGGIATOR



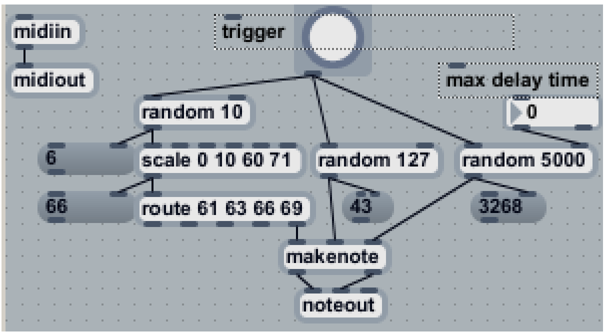
1.3. Timbre synthesizer



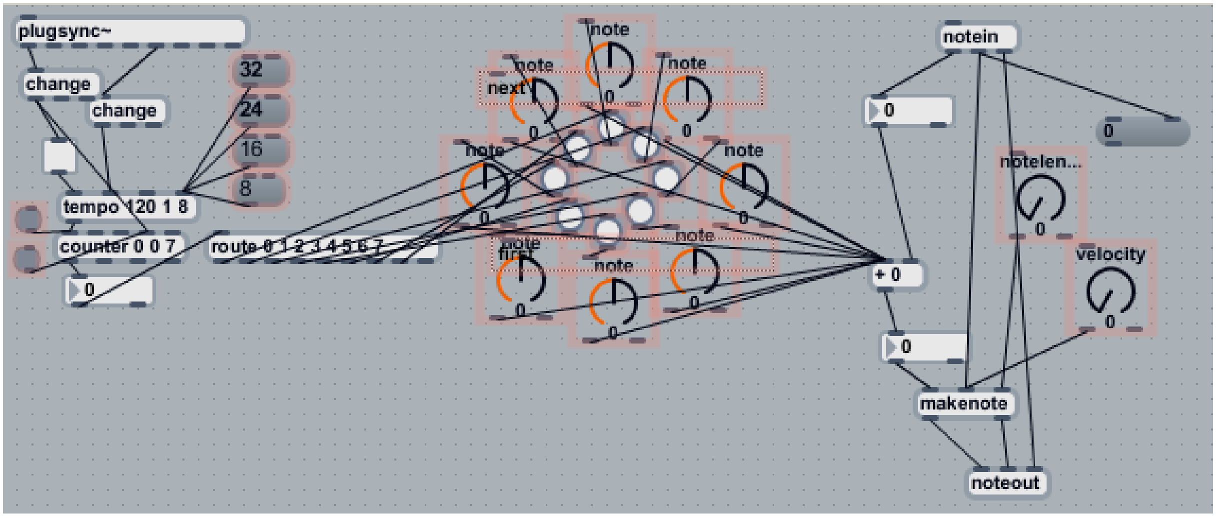
1.4. Rhythm sequencer



1.5. GENERATIVE SEQUENCER



1.6. CIRCULAR SEQUENCER WITH MANUAL CONTROL



2. Arduino code

Meeblip_Project0_3 | Arduino 1.0.1

File Edit Sketch Tools Help

Verify

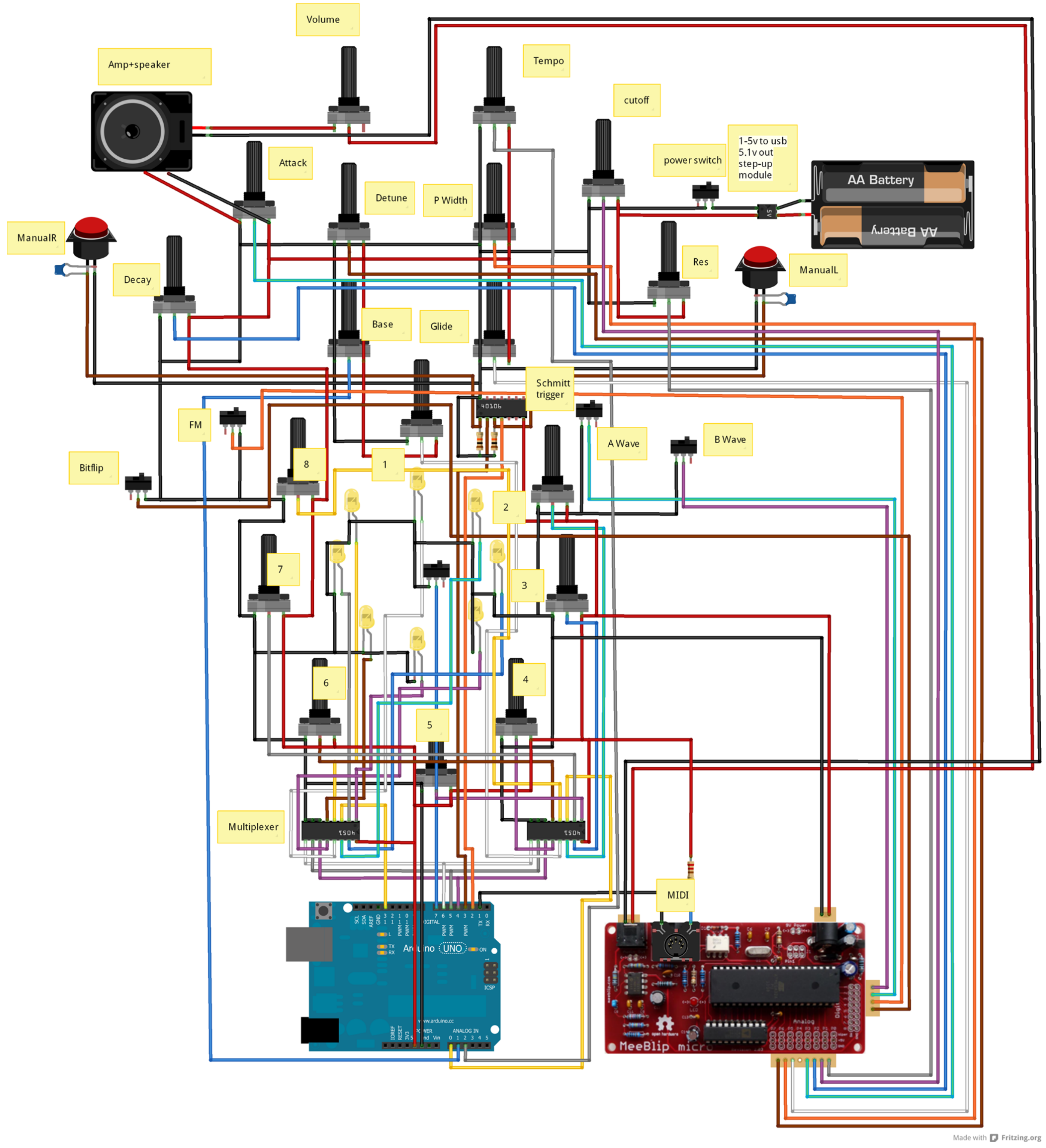
Meeblip_Project0_3 §

```
//Code by Arvid Jense 2012   http://www.arvidjense.com
//Licenced as GPLv3 http://www.gnu.org/licenses/gpl.html


//Initialising variables


//Multiplexer
int r0 = 0;      //value of select pin at the 4051 (s0)
int r1 = 0;      //value of select pin at the 4051 (s1)
int r2 = 0;      //value of select pin at the 4051 (s2)

//Sequencer
int count = 0;   //which y pin we are selecting
int note = 0;
int seqValue = 0;
int tempoKnob = 500;
int base = 0;
int baseKnob = 0;

unsigned long Timer1 = 0;
int waitTime = 200;

//On what pin the LED is connected
#define LEDPIN 13


void setup(){

  //Serial.begin(9600); //This is needed for debugging on the computer
  Serial1.begin(31250); //This initialises the Serial protocol to send and recieve MIDI

  //For the multiplexers
  pinMode(4, OUTPUT);    // s0
  pinMode(5, OUTPUT);    // s1
  pinMode(6, OUTPUT);    // s2

  pinMode(LEDPIN,OUTPUT);
  pinMode(7, INPUT_PULLUP);

  attachInterrupt(1, buttonLeft, RISING);
  attachInterrupt(0, buttonRight, RISING);

  digitalWrite(LEDPIN,HIGH); //to make sure a led is lit, even though run is turned off

  //These MIDI messages make sure the Meeblip is set as desired when the instrument is
  //turned on. The codes can be found in the Meeblip documentation
  midiCC(51,0);  //LFO depth 0
  midiCC(52,64); //filter envelope medium
  midiCC(52,64); //filter decay medium
  midiCC(52,64); //filter attack medium
  midiCC(64,0);  //knob shift off
  midiCC(70,0);  //LFO off
  midiCC(74,127);//OSC B on
  midiCC(76,0);  //env sust off
  midiCC(78,127);//PWM on
}

void loop () {
  if (digitalRead(7)==0){ //if the run button is on, play the sequencer
    tempoKnob = analogRead(A2);
    waitTime = map(tempoKnob, 0, 1023, 500, 50); //A range of 16th at 30 to 300 BPM
    if (millis() - Timer1 > waitTime){
      count++;
      Timer1 = millis();
      readNote(count);
    }
  }
}
```

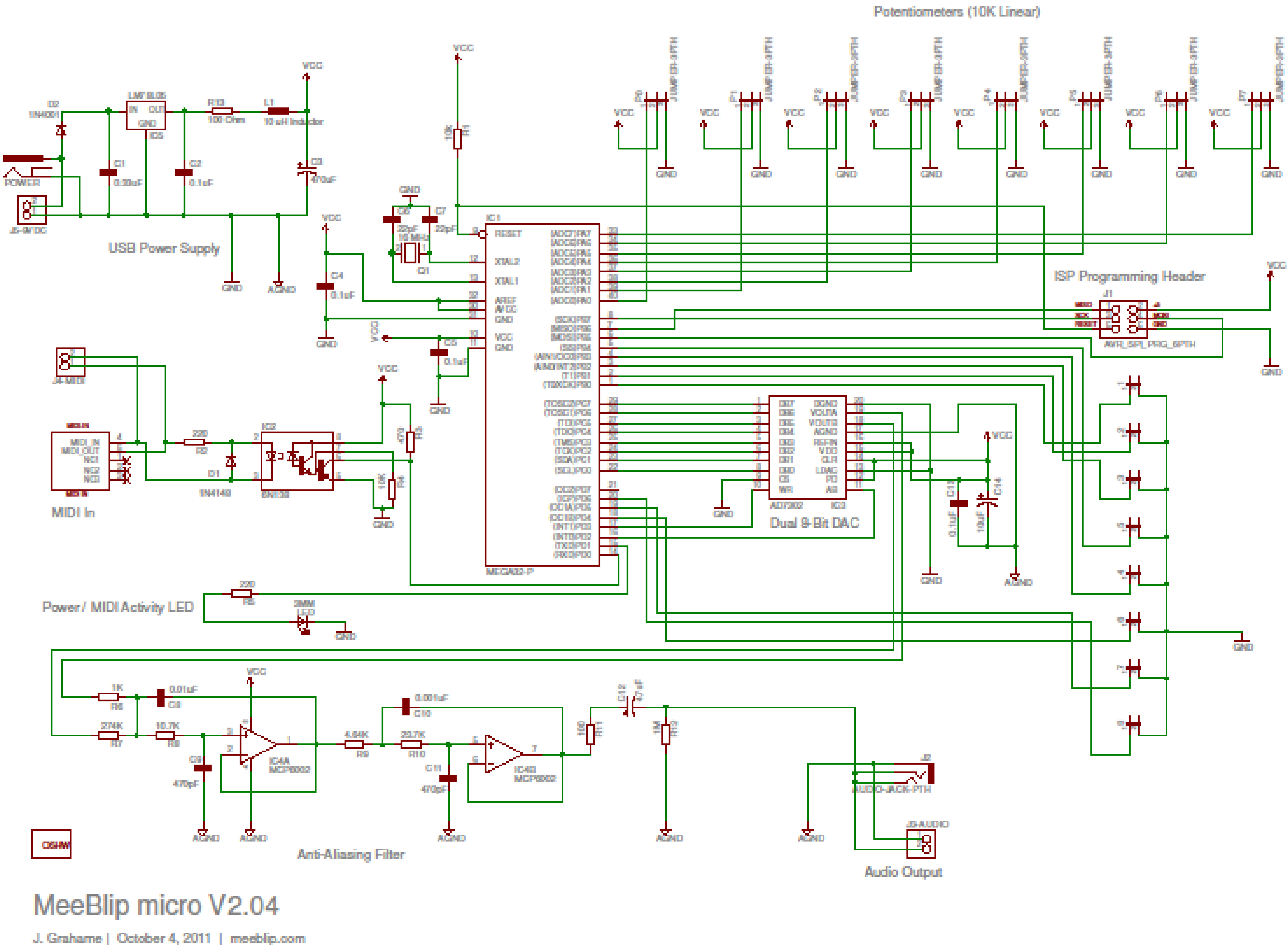
```
//What happens when the dynamic buttons are pressed
void buttonLeft(){
  count--;
  readNote(count);
  Timer1 = millis();
}
void buttonRight(){
  count++;
  readNote(count);
  Timer1 = millis();
}

//What reads the knobs
void readNote(int cnt){
  if (count > 7  && count < 0){
    count = 0 ;
  }
  digitalWrite(LEDPIN,LOW); //turn the led off, might be arbitrary

  //noteOn(note,0x00); //turns the note off to avoid hanging notes,
  //but interfers with glide?

  //code to set the multiplexer to the desired mode

  // select the right bits for the multiplexer
  r0 = bitRead(cnt,0);
  r1 = bitRead(cnt,1);
  r2 = bitRead(cnt,2);
  digitalWrite(4, r0);
  digitalWrite(5, r1);
  digitalWrite(6, r2);

  //reading the value from the knobs and mapping is to a midi compatible range
  seqValue = analogRead(A0);
  baseKnob = analogRead(A1);
  base = map(baseKnob, 0 ,1023, 48, 84);
  note = map(seqValue, 1, 1023, -12, 12); //meeblip range is limited from 36 to 96
  note = note + base;

  //is the knob is set all the way to the left, no note will be played and the
  //LED will be more dim
  if (seqValue > 1) {
    noteOn(note,0x45);
    digitalWrite(LEDPIN,HIGH);
  }
  else {
    analogWrite(LEDPIN,10);
  }

  //These makes the right binary code for midi messages
}
void noteOn(int pitch, int velocity) {
  Serial1.write(0x90);
  Serial1.write(pitch);
  Serial1.write(velocity);
}

void midiCC(int cc, int value) {
  Serial1.write(0xB0);
  Serial1.write(cc);
  Serial1.write(value);
}
```

APPENDIX B. SCHEMES

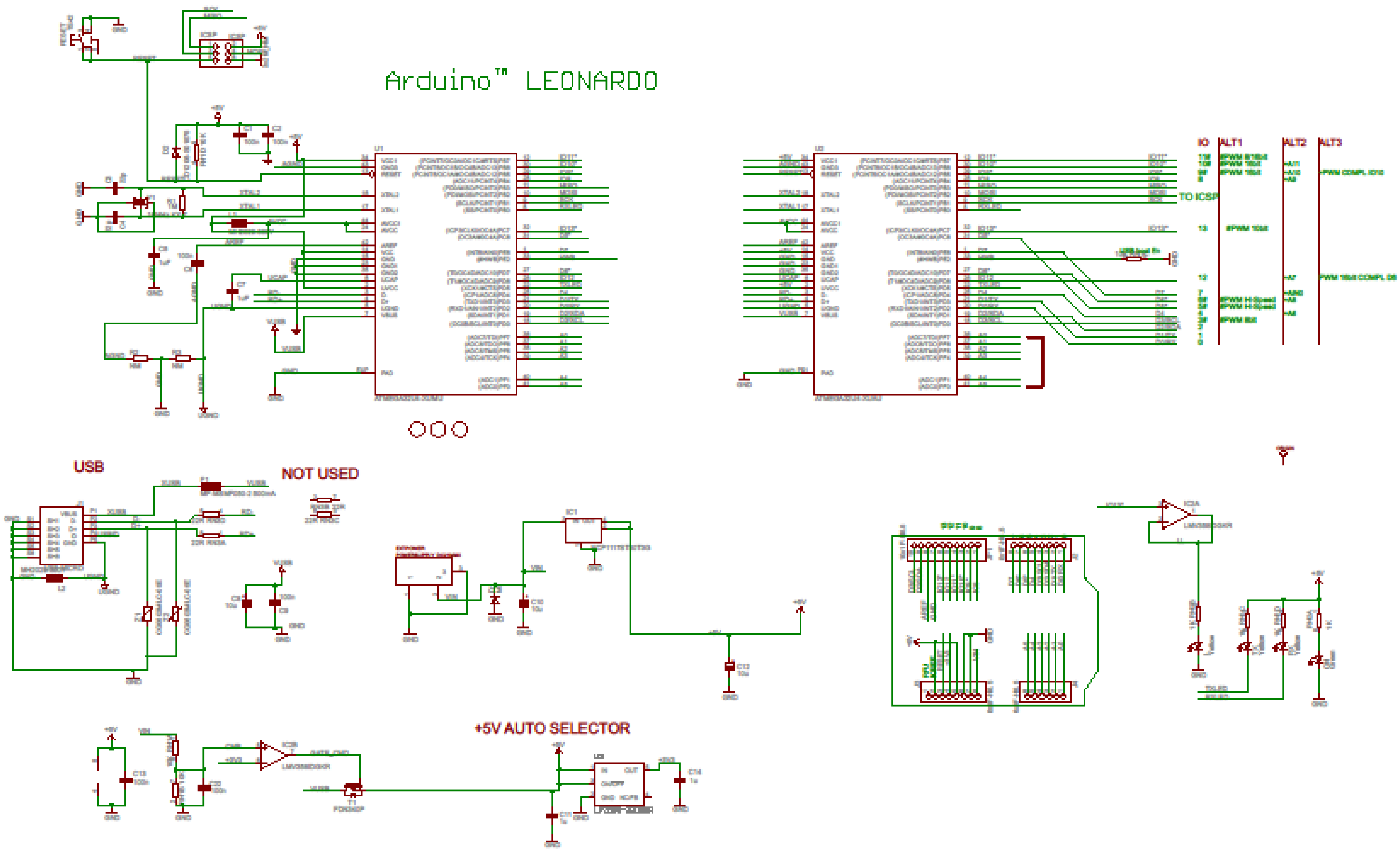
1. THE DESIGNED INSTRUMENT



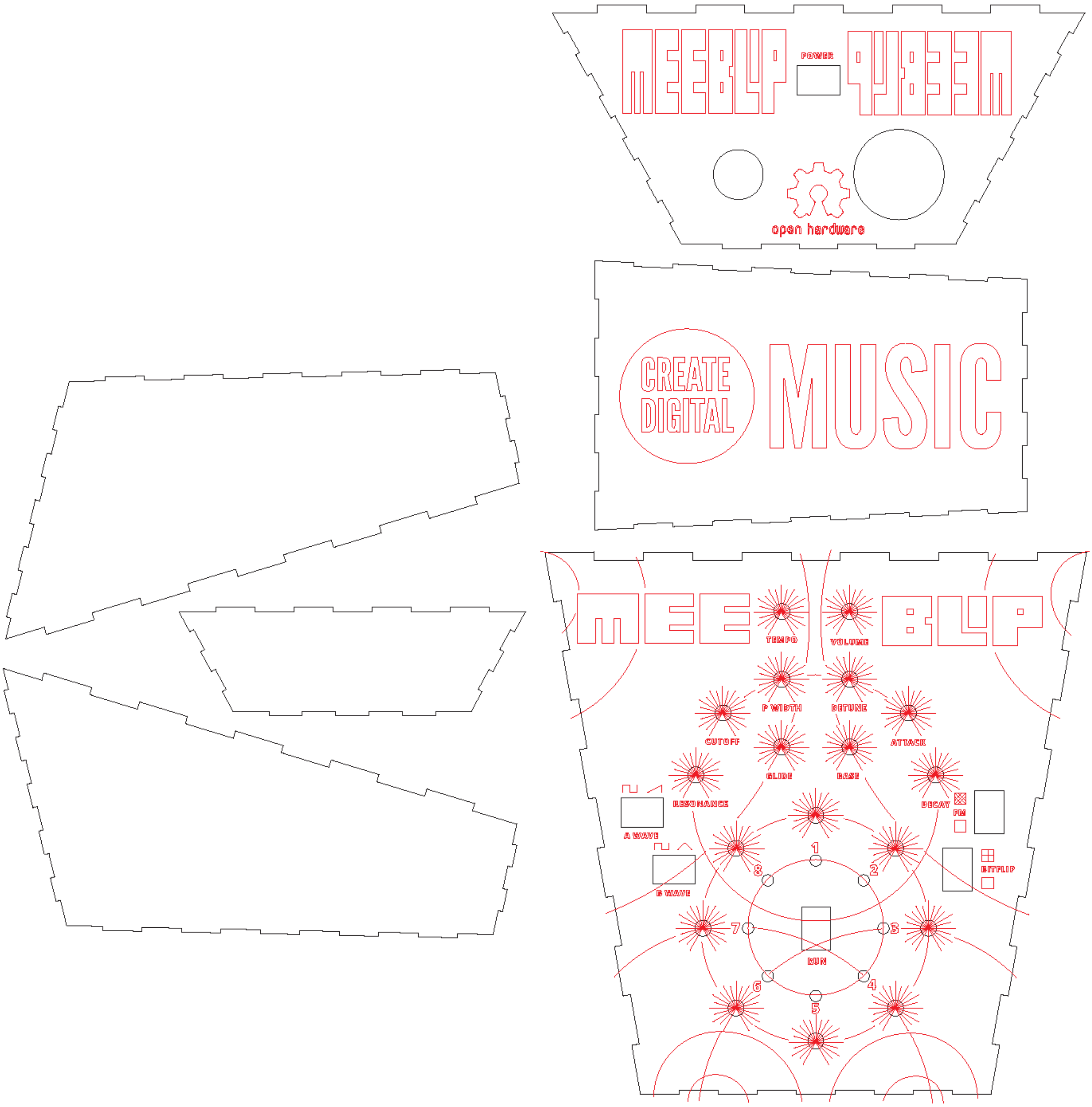
2. MEEBLIP


Potentiometers (10K Linear)
USB Power Supply
ISP Programming Header
MIDI In
Dual 8-Bit DAC
Power / MIDI Activity LED
Anti-Aliasing Filter
Audio Output
MeeBlip micro V2.04
J. Grahame | October 4, 2011 | meeblip.com

3. Arduino Leonardo



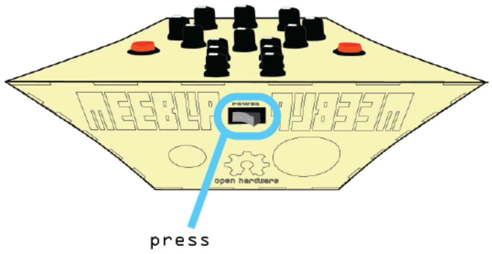
APPENDIX C. LASER CUT FILE



APPENDIX D. MANUAL

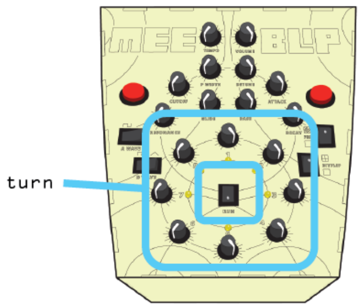
1) Aquire Device

2) Turn it on

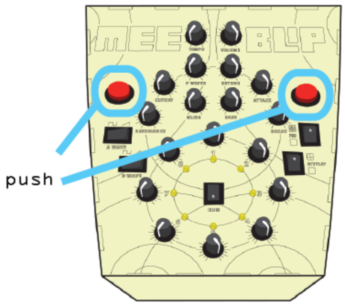


3) Place in lap

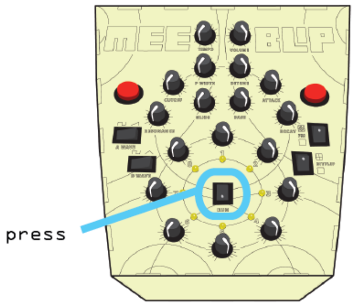
4) Set notes



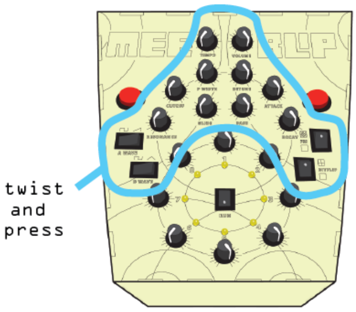
5a) Play manually using the red buttons



5b) Play automatically using the run switch



6) Use these knobs and switches to change the tone

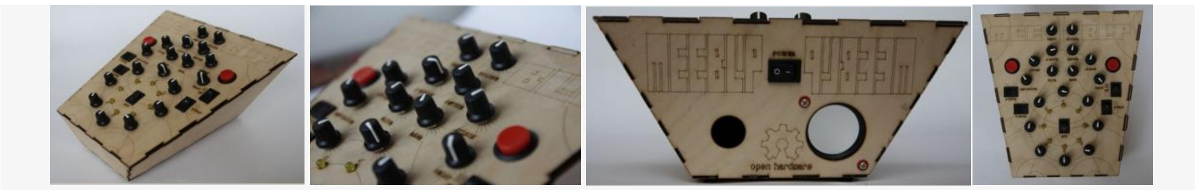


7) Have fun!

APPENDIX E. INSTRUCTABLE

This is a reduced version, see full version at: http://www.instructables.com/id/Designing-and-building-an-acoustic-synthesizer/?ALLSTEPS

Designing and building an 'acoustic' synthesizer by ArvidJense



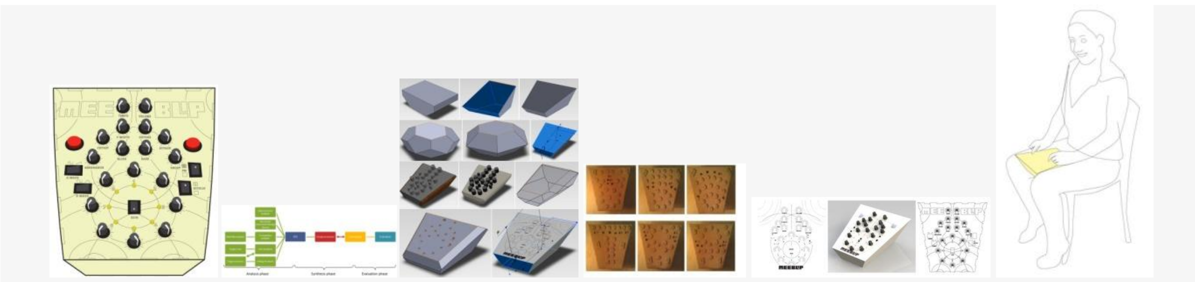
For my internship at Create Digital Music/Meeblip I've created quite a fun little synthesizer. This thing is meant to be more flexible than other synthesizer by making it completely independent; it is battery powered, has its own amplifier/speaker and is controlled by a manually operated sequencer. Inspiration for this form comes from my frustrations with most synthesizers: that I can't just take them out and jam with friends!

This instructable shows how I made my prototype and describes how you could make your own. It won't go into super-deep detail, so if you want any assistance, please don't hesitate to contact me!

More details can be found in the report I wrote here.

This document and all included research and illustrations, are covered by a Creative Commons Attribution-ShareAlike 3.0 Unported License. The hardware designs, schematics and code are provided under GPL v3. This means that you can rebuild and improve on this project as you wish, just let me an Meeblip know!

STEP 1: CONCEPT AND DESIGN



So, synthesizers are pretty cool instruments, making the sounds for the most of the music I listen to right now. Yet they are quite limited in where and how they can be used. For instance, I can play my guitar at home, but I could take it to the park or to a friend for a quick jam-session. Not so simple for a synth: I'd have to plug it out of my studio setup, take a midi keyboard or laptop and just hope the place where I'm going has the right connections to plug in again.

Another point where I feel synthesizers are performing worse than traditional instruments, is in their connection to the user. Where I can feel the vibrations of my guitar resonating through my hands and body, the sound of a synthesizer comes from a speaker a few meters away from me. Similar on the control interface; A guitar is quite clear in the fact that if you pluck a string,a sound comes out. But for a synthesizer you will have to learn the function of dozens of knobs which often have several layers of functionality.

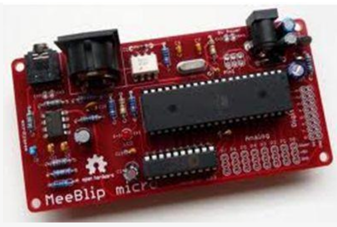
On the other hand the palette of different sounds coming from a synthesizer is bigger than that of most traditional instruments. Also the possibilities for programming the notes allow for things not possible on traditional instruments. So why choose one or the other? I wanted to create a synthesizer which has the flexibility and direct connection of a traditional instrument.

As I've done this project for Meeblip, one thing was certain, I would use a Meeblip Micro as the sonic centerpiece. Other than that, everything was still open. Would it become a drumbox or rather a guitar-like synth? After looking at a lot of different synthesizers, reading up on synthesizer design literature and sketching lot of variations, I came up with something.

After making a lot of prototypes of all the individual pieces (interface, electronics, sequencer etc) I finally found the 'final form' which is made in the next steps. All functions are directly accessible through the knobs and switches on the interface The internal speaker allows the instrument itself to vibrate, which, especially when you place the instrument in your lap, help you feel what your playing. The instrument is controlled through a circular sequencers, in which you set the notes form a range of -12 to +12 semitones from a center note set by the base knob.

I've yet to create a decent video showing all the function, but as soon as that is present, I'll put it in here!

Step 2: Materials, tools and budget



Things to buy:

Price (total) **Item**

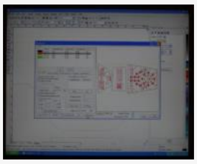
€ 50,00 1 x Meeblip Micro
€ 20,00 1 x Arduino Leonardo NOTE: MOST OTHER ARDUINOS COULD DO, BUT WOULD NEED SOME MODIFICATIONS IN THE CODE
€ 10,00 1 x Cheap USB speakers NOTE: THESE ARE ALL REALLY SIMILAR, I GOT MINE AT SATURN
€ 15,00 1 x 250x500x2mm model airplane beech plywood NOTE: MY LASER-CUTTING FILES WILL ONLY WORK FOR 2MM WOOD, BUT THEY COULD BE MODIFIED IF NEEDED
€ 30,00 1 x 30minutes of laser-cutting NOTE: THIS COULD BE FREE IF YOU HAVE A FABLAB NEAR YOU OR IT COULD BE HUGELY MORE EXPENSIVE IF YOU HAVE TO RELY ON AN ONLINE LASER-CUTTING SERVICE
€ 18,00 18 x Potentiometers (10K)
€ 18,00 18 x Knobs
€ 3,90 6 x Rocker switches
€ 4,00 2 x Arcade buttons modified
€ 0,40 8 x LEDs (5mm yellow)
€ 10,00 1 x Protoboard
€ 0,30 3 x IC mounts
€ 0,10 2 x Resistor (10k)
€ 0,10 2 x Capacitor (10nF)
€ 0,50 1 x Headers male (about 30 bits)
€ 0,50 1 x Headers female (about 30 bits)
€ 2,00 2 x Multiplexers (4051)
€ 0,50 1 x Hex inverting Schmitt trigger (40106)
€ 1,00 1 x 2aa battery holder
€ 3,00 1 x USB step up converter
€ 1,00 1 x Rubber band (X shape)
€ 4,00 1 x Assorted Wires
€ 2,00 4 x Rubber feet
€ 8,00 2 x AA batteries (rechargable)
€ 202,30 Total

NOTE: THESE PRICES ARE APPROXIMATIONS, YOU MIGHT BE ABLE TO GET SOME PARTS CHEAPER OR MORE EXPENSIVE, BUT THE BALLPARK COST IS AROUND €200,-

Tools and consumables:
Glue
Solder
Electrical tape
Drill bits
Soldering iron
10mm wrench
File

Sanding paper
Clear stain

Step 3: Encasing


First step is to do the laser-cut enclosure. As I couldn't find any laser-cutting facilities with a powerful machine, I didn't have much choice in wood; it had to be 2mm or thinner. Most types of wood are quite weak and flexible at that thickness, so I had quite a search finding the right wood. I finally settled to a beech plywood for model airplanes of 'F1' quality, whatever that means.

The cutting files can be found in **Meeblip lasercut 2mm hairlines part1 [Converted].pdf** and **Meeblip lasercut 2mm hairlines part2 [Converted].pdf**.

Before cutting the files, you should do some tests on the wood to determine the right intensity and speed. This is very important on the lettering, as a line too thin will be illegible from a distance, while too thick of a line will make the lines flow over. I used the same settings for the lines and letters, but for a next version I would increase the intensity on the lines, to make them a bit more pronounced. The pdf files show the cut lines in red, while the engraved lines are black

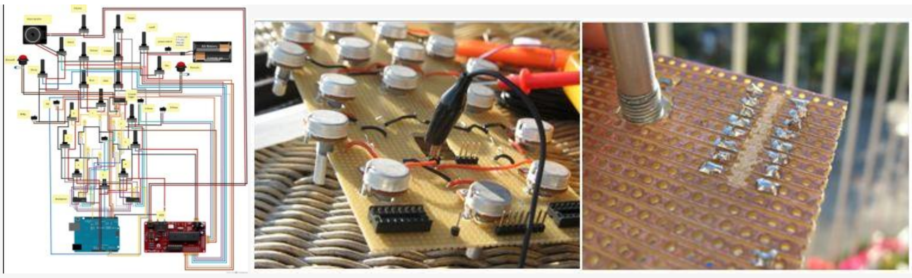
After everything is cut and engraved, it is time for assembly. Just put all the pieces, except for the front panel, together. Put some high strength (wood) glue in the seams. (Be careful that you don't smudge the wood or leave large blobs of glue hanging on the outside.) Tie this together with some rubber bands and leave it overnight.

With the encasing glued, its time for some finishing. First, be sure there is no grease on the wood, if there is, remove it with any degreaser. Now, use some fine grit (>P150) to sand the wooden panels with the grain. Clean it again after this is finished and let it dry. Now you can put any number of layers of light stain on, I did only one.

Meeblip lasercut 2mm hairlines part2 [Converted].pdf216 KB

Meeblip lasercut 2mm hairlines part1 [Converted].pdf333 KB

Step 4: Electronics



Getting the electronics right in this project is by fare the hardest task in this project. If you can make a PCB, go for it, as it will save a lot of time fixing minor errors. Otherwise, use a stripboard, as you won't need any special tools. You can follow the wiring as shown in the Fritzing file found in **ACOUSTIC MEEBLIP SCHEME.ZIP**.

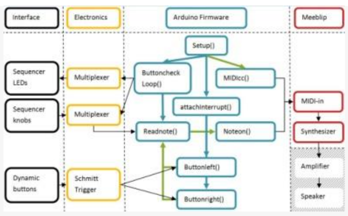
You should be very careful to do everything right the first time, as fixing mistakes is really time consuming. Often when I try to fix one broken wire, the process makes some other wires snap, causing me to spend an hour on what should've been a five minute repair.

Things to note here are how I use two AA batteries with an step-up converter instead of a 9V battery. This is an advantage on multiple levels, AA batteries are cheaper and more common than 9V block and they also provide a a longer operating time on the right voltage.

Another things is the use of the Inverting Schmitt Trigger. This, in combination with the resistor and capacitor, causes the Manual buttons to avoid bouncing and causing a note to be triggered double. (This is complemented by a pieces of debouncing code on the Arduino.)

Acoustic Meeblip Scheme.zip1 MB

STEP 5: SOFTWARE



Now all the wiring is done, we can go on with the code. The scheme shows how all the parts interact with each other. What we can see is that the Arduino does all the sequencing and note generation, while all the actual sound synthesis is left to the Meeblip.

The Arduino code is not very complex. It has two modes of operation, automatic and manual. The automatic mode works when the 'run' switch is flipped. Now it will walk through all the sequencer steps on an interval set by the 'tempo' knob. On each step it will set the multiplexers to the right knob and LED, of which the value will be translated in a MIDI note, which in turn is send to the

Meeblip through Serial1. (Note: on other Arduinos than Leonardo, this might just be changed to Serial). The manual mode works similar, but works instantly (interrupts the processor), with the right button walking clockwise and the left button walking counter-clockwise.

The arduino code can be found in **Meeblip_Project0_4.zip**

Some code was changed on the Meeblip as well. Most notably the bindings of the knobs, but it also checks the state of all the knobs on start-up and removes the use of midi CC. This causes that the value a knob is set to, is always the setting which you can hear. Using Arduino ISP and avrdude, I uploaded the firmware with the following command:
avrdude -P COM5 -b 19200 -c avrisp -p m32 -B 5 -U flash:w:meeblip-micro.hex -U lfuse:w:0xBF:m -U hfuse:w:0xD9:m -U eeprom:w:meeblip.eep
(More info on this can be found here)

Both micro.hex and micro.eep can be found in **Acoustic meeblip micro firmware.zip**

Meeblip_Project0_4.zip 2 KB

Acoustic meeblip micro firmware.zip 12 KB

Step 6: Conclusion




And there you have it, an 'acoustic' synthesizer!

As I feel like this is just a first iteration on a completely new type of instrument, I've made a little to-do list. Things I would like to do for a next prototype:
 * Use a PCB istead of a stripboard
 * Easy battery access
 * Open/Close system
 * Make the push buttons dynamic sensitive (velocity or pressure)
 * Add audio+midi out and midi (sync) in
 * Higher quality amp+speaker
 * Better ledgible interface lettering
 * Battery level indication
 * Integrate the Meeblip and Arduino on the PCB
 * Make it solar/hand powered
 * Make it waterproof

APPENDIX F. WORKING METHOD

WORKING METHOD (PVA) BACHELOR ASSIGNMENT: CREATE DIGITAL MEDIA/MEEBLIP – ARVID JENSE S0180831

This is the Working Method for the Internship Arvid Jense is taking at Create Digital Media in Berlin between 9/7 and 1/9. All word in *italic* are further explained in the definitions section.

TARGET (GEPLAND DOEL)

4. ACTOR ANALYSIS

4.1.1. CREATE DIGITAL MUSIC

Create Digital Media (CDM) is a young creative company focused on people who make music and live visuals with technology. It sells targeted *open source* musical instruments and produces advertiser-supported online content for an international audience (daily blogs with news and features, community, video, and media). The product line is currently represented by different variations of the *MeeBlip* synthesizer.

Operated by Peter Kirn since 2004, CDM in 2012 reincorporated with Berlin as its capital. It contracts with writers and media sales in San Francisco and partners with Reflex Audio of Calgary, Canada for hardware design and manufacturing.

Openness is a unique part of CDM's business model. All CDM content is released under a Open Source and *Open Hardware* license.

In launching their internship program, they provide experience with this emerging business and an open set of contributions available to the broader communities of artists and open-model businesses internationally. Peter Kirn will function as the company supervisor.

4.1.2. ARVID, STUDENT

Arvid is finishing his Industrial Design Engineering bachelor program. He needs to show competence as an Industrial Designer at Bachelor's level by fulfilling the assignment and showing the following abilities:

- Applying self-directly expertise of Industrial Design Engineering, and if required deepen this expertise;
- Providing (with before mentioned expertise) a surplus value for the company;
- Working systematically and being able to communicate appropriately and effectively;
- Possessing sufficient reflective abilities to function as an Industrial Designer.

Can provide CDM with fresh ideas and can use his diverse IDE competences to supplement CDM.

4.1.3. Universiteit Twente (UT), Industrial Design Engineering

UT needs to see Arvid apply his learned knowledge and provides two persons to guide Arvid.
Bachelor Assignment Coordinator: ir. Arie Paul van den Beukel.
Bachelor Assignment Tutor: ??? During the assignment, a university tutor will guide the student. This guidance focuses on the approach and progress. The tutor will also monitor if the assignment succeeds in providing the appropriate opportunities for a student to prove his or her competences. Furthermore, the tutor, and if necessary also other university professors, will be available to provide course related counsel and information.

5. Project scope (Projectkader)

CDM produces the *Meeblip* Synthesizer. While the Meeblip is a fully functional product receiving positive reviews, CDM wants to keep developing their product to expand their market. They would like to do this by using the existing Meeblip technology for a new musical instrument with a different kind of interaction than the original Meeblip. While currently the Meeblip is dependent on *a midi controller* or laptop for performing music, they hope to turn it into an independent instrument by integrating a *sequencer* with the Meeblip.

6. Objective (Doelstelling)

The objective of this assignment is to create a new variation of a music instrument that is based on the existing 'Meeblip-technology' from CDM and to develop the interface (in terms of user interaction and the instrument's appearance) of this new variation. CDM is interested in doing this by integrating a sequencer within the product. This new to be created variation is intended to appeal to a different target audience than the existing Meeblip.

The design process starts by analyzing the the Meeblip synthesizer both in functionality and user experience. To comply with company policy, an analysis will have to be made on *Open Hardware* and how this impacts the design process. An analysis will be made of potential futures users and use scenarios, of which one will be chosen. Chosen target audience will be analyzed together with competing products for the scenario.

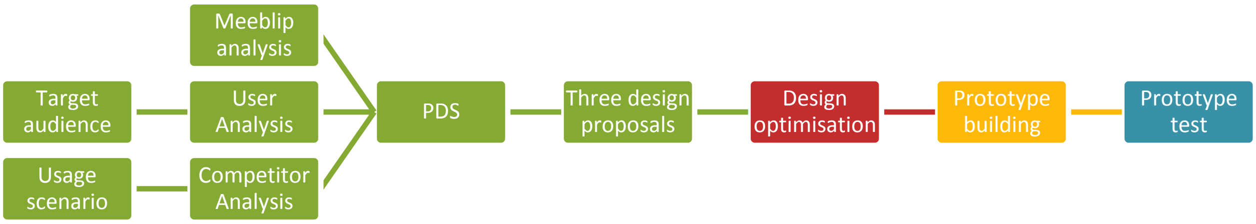
The needs of previous research will be formed into a product design specification (PDS) that will function as input for three design proposals of both the user interaction as well as the appearance and interface. One of these may be chosen by CDM to develop further into a proof of concept. This should demonstrate functionality of the interface, using the simple *Arduino* platform, as well as demonstrating the appearance of the new product. This assignment will be completed within 14 weeks. As a lot of the considerations dealt with are in specialized musical/technical language, special care will be taken to write the report into layman's terms.

To limit the scope of the assignment, manufacturing and marketing considerations will not be dealt with.

Deliverables:

- Product design proposal, documented by renders and technical drawings as well as text.
- Proof of concept as a tangible musical instrument.
- Report with text and images, according to university guidelines.

7. Research Model



8. Research questions (Vraagstelling)

Explorative research

1. What is the *'Meeblip technology'*?
    a) What are the features of the 'Meeblip technology'?
    b) What is the design philosophy behind the 'Meeblip technology'?
    c) What have been the design considerations on the current Meeblip?
    d) Who are the current Meeblip users?
    e) What do they (dis)like about the current Meeblip?
    f) What features do they miss on the current Meeblip?
2. What will be the target audience for the design proposals?
    a) Who could be potential future Meeblip users?
    b) What is holding these potential users from buying a Meeblip right now?
    c) What are usage scenarios for this target audience?

Why choose this one?

3. What can be learned from competing products and literature dealing with this scenario?
    a) What scenarios are there?
    b) Which one is chosen…
    c) What products are currently used in the usage scenario?
    d) What design features do these products employ to encourage usage in this scenario?
    e) What literature can be found that deals with usage in this scenario?
4. What are the concluding specifications which need to be fulfilled in the new design?
    a) What are the specific needs and wishes for the target audience?
    b) What are the specific needs and wishes for the usage scenario?
    c) What are the specific needs and wishes for CDM?

Design Proposal (specify like in objective)

5. What design proposal fits the PDS best?
    a) What solutions can be found within the PDS?
    b) What are the features of this design?
    c) What is the appearance of this design?

Design optimization

6. How can this design be optimized?
    a) How can the interaction between the instrument and its users be optimized?
    b) How can the appearance be more appealing to the target audience?

c) How can the design be optimized to allow the building of a prototype?

Prototyping

7. What does a prototype need to comply with for testing whether it complies with the PDS or not?
   a) What does the testing scenario look like?
   b) How can the electronic features be prototyped?
   c) How can the casing be prototyped?

Evaluation

8. How does the prototype comply with the PDS?
   a) How does the prototype perform in its usage scenario?
   b) What does the user think about the new design on terms of function?
   c) What does the user think about the new design on terms of appearance?

9. DEFINITIONS (BEGRIPSBEPALING)

9.1.1. MEEBLIP

From http://meeblip.com/: "MeeBlip is a hackable, affordable digital synthesizer, made for accessible sound and hands-on control. It can be someone's first synth. It can be a unique-sounding addition to your music setup, playable with *MIDI hardware* and software. It can be a synth you open up and modify, learning about sound creation, code, and electronics. Or it can be the basis of new projects and ideas."
Meant as the synthesizer for everyone. While it should be a functional product right out of the box, it accommodates users to modify it to their needs. Being one of the first Open Hardware synthesizer, the Meeblip could either be considered a niche product for diy (do it yourself) synthesizer enthusiasts, or the forerunner of a new generation of electronic instruments, using a new 'information age' business model.

9.1.2. MEEBLIP TECHNOLOGY

The microcontroller used for controlling and generating sound in the Meeblip. CDM has produced a barebones version of the Meeblip, just the microcontroller on a PCB, the Meeblip Micro, meant to allow user to create their own synth based on the Meeblip.

9.1.3. SEQUENCER

From http://recordingworkshopuk.com/music-sequencer-tutorial: "A music sequencer is an electronic device that generates data used to control musical devices such as sound modules, effects units and processors. Traditionally these were 'boxes' containing sliding faders and LEDs that flashed in sequence i.e. one after another. These were used to activate a corresponding fader and each fader sent out a voltage to control for example a voltage controlled oscillator or VCO, the result created a melody or base line which was commonly used in electronic music during the early 1970's right through until the 1980's when computers became powerful enough to take over the role."

9.1.4. MIDI CONTROLLER

Most electronic musical instruments can communicate by connecting them with MIDI cables; this means you can hear one instrument, while playing on another. Cost and space considerations over time have caused most synthesizers to be sold in two version, one with an included keyboard, which you could play independently, and another version which is just a box with knobs, which is dependant on external devices like a keyboard, called a midi-controller, to make sounds.

9.1.5. ARDUINO

From http://arduino.cc/: “Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).”

Open Source

From Wikipedia: “In production and development, open source is a philosophy or pragmatic methodology that promotes free redistribution and access to an end product's design and implementation details. […]The open-source model includes the concept of concurrent yet different agendas and differing approaches in production, in contrast with more centralized models of development such as those typically used in commercial software companies. A main principle and practice of open-source software development is peer production by bartering and collaboration, with the end-product, source-material, "blueprints", and documentation available at no cost to the public.”

9.1.6. Open Hardware

From http://freedomdefined.org/OSHW: “Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to it. Ideally, open source hardware uses readily-available components and materials, standard processes, open infrastructure, unrestricted content, and open-source design tools to maximize the ability of individuals to make and use hardware. Open source hardware gives people the freedom to control their technology while sharing knowledge and encouraging commerce through the open exchange of designs.”

9.1.7. Product design specification (PDS)

From http://www.bath.ac.uk/idmrc/themes/projects/delores/co-design-website/dpg/pds/pds1.htm: “A product design specification (PDS) is a document which sets out fully and in detail exactly what will be required of a product, before it is designed. […]”

This document will include all needs and wishes for both the user, company and regulations. This is similar to a dutch *Plan van Eisen (PvE).*

Method (geplande aanpak)

10. Project strategy (Projectstrategie) Materials (Onderzoeksmateriaal)

Question	**Strategy**	**Material**	**Source**
1a	Content analysis	documents	Meeblip documentation
	Interview	person	Meeblip developers
	Experiment	product	Testing current Meeblip
1b	Interview	person	Meeblip developers
1c	Interview	person	Meeblip developers
1d,e,f	Interview	person	Meeblip developers
	Observation	media	Internet user reviews, forums, product reviews
2a,b,c	Content analysis	documents	Open Hardware documentation
	Interview	person	Meeblip developers
3a,b	Interview	person	Meeblip developers
	Observation	media	Internet forums

3c	Brainstorm	person	Intern and Meeblip developers
4a	Observation	media	Internet video’s
	Interview	person	Target audience member
4b	Content analysis	product	Earlier found products
4c	Observation	documents	Online articles and essays
5a	Interview	person	Target audience member
	Observation	media	Internet user reviews, forums
5b	Brainstorm	person	Intern and Meeblip developers
5c	Interview	person	Meeblip developers
6a	Brainstorm	person	Intern
6b,c	Observation	product	Design proposals
7a	Brainstorm	person	Intern
7b	Interview	person	
7c	Brainstorm	person	Intern
8a,b,c	Brainstorm	person	Intern and Meeblip developers
9a,b,c	Case study	person	Multiple target audience members

11. PLANNING

Week	Time period	Activities	Deliverables
-1		ITO	
0		Solicitations, communication, writing working method	Working Method
1	9/7-13/7	Research	Chosen target audience and usage scenario
2	16/7-20/7	Research, setting up PDS and concepting	PDS
3	23/7-27/7	Concept development and design	Concept chosen
4	30/7- 3/8	Design optimization	Renders and technical drawings
5	6/8-10/8	Design optimization , prototyping	First working proof of concept
6	13/8-17/8	Prototyping	Casing
7	20/8-24/8	Debugging prototype, setting up user test	Functional prototype
8	27/8-31/8	User tests, gathering data for documentation and	User test results

		report	
9		Writing documentation and report	Documentation
10		Writing report	
11		Finishing report	Report
12		Presentation	

Grey weeks are done on location, white weeks are done in Enschede

Bottlenecks:

Bottleneck	**Solution**
The right person an interview can't be found	Search for alternative sources of information on the internet or in literature
Interview doesn't produce the needed information	Find alternative person to interview
Prototyping tools available are not sufficient to demonstrate the concept	Use an alternative method to evaluate use, like a software simulation