

VolcaBoy – Product Development Plan

Assignment 1 - Project Proposal

Interfaces and Interactivity - 31419 - AUT - 201820

Silvio Gregorini

ID: 77203842

8 November 2019

Contents

1	Introduction	2
2	Background and Motivation	2
2.1	Pushing the Limits Using Constraints	2
2.2	Rationale: Conscious Limitations	3
3	Project VolcaBoy	4
3.1	Concept	4
3.2	System Architecture	5
3.2.1	Hardware	5
3.2.2	DMG-001 Mods	7
3.2.3	Embedded Software	8
3.3	Production	9
4	Conclusions	10

1 Introduction

The aim of this paper is to propose the design and production of an hardware synthesizer, starting from an existing digital sound engine. In the sections that follow, it will be explained the background and the rationale behind the project, as well as the initial research undertaken. Then, it will be depicted a "blueprint" of the product: the general concept, the architecture of the system – both hardware and software – and the production plan will be covered in detail. In the last part the evaluation criteria will be set, in order to have a concrete measure of the work outcomes.

2 Background and Motivation

This project has been shaped and will be realised keeping as a pivotal point the *chiptune* subculture and its principles. As will be discussed in the next paragraphs, the product is conceived to be used by people already familiar with the environment and limitations of this musical style. Its purpose is to give users a different and more modern way to interact with a well-known set of sounds and synthesis capabilities.

2.1 Pushing the Limits Using Constraints

Chiptune As stated by Collins et al. (2014) the term *chiptune* has multiple definitions. Also known as *chip music* or *8-bit music*, it derives from the sound chips that, in the first generation of computers and gaming consoles, were used to balance the processing power of generating sound effects and music from the CPU. In its strictest meaning, chiptune is used to refer to music created entirely from the original, vintage audio chips. Nevertheless, modifications that do not alter the nature of the sound produced are allowed.

The broadest definition is more related to the aesthetics of the sound, rather than to the source generating it. The entire subculture which gravitates around the foundations and features of chip music can be called *chiptune* too. Anyway, this paper and the related project will try to stick to the strictest definition of the word, in order to create a product able to maintain the sound fidelity of the old processors.

DMG-001 and trackers In particular, the following study focuses on the Nintendo DMG-001 from 1989, known with the commercial name of *Game Boy*, which is supposedly one of the most popular tools for the production of chip music. From the official datasheet (Nintendo, 2019), this portable console runs on a custom *Sharp LR35902* 8-bit CPU, similar to the *Zilog Z80*¹ and has four

¹Popular 8-bit microprocessor widely used from the 1970s to the mid-1980s in desktop and home computers, military applications, synthesizers, arcade machines...

audio channels²:

Channel	Type	Features
1	Quadrangular ³	<ul style="list-style-type: none"> - Volume envelope - 4-mode pulse width - Frequency register from C3 upwards - Frequency envelope
2	Quadrangular	<ul style="list-style-type: none"> - Volume envelope - 4-mode pulse width - Frequency register from C3 upwards
3	Wave	<ul style="list-style-type: none"> - User-definable waveforms - Bank of 32 samples (4-bit each) - Frequency register from C2 upwards
4	Pseudo-random noise	<ul style="list-style-type: none"> - White and brown noise

Table 1: Sound channels of the Nintendo Game Boy DMG-001.

Back in the analog era (i.e. before the first DAWs were created and deployed), the tools to compose and produce music on digital computers and consoles were the *music trackers*. They can be defined as precursors of the modern music production softwares: notes, parameters, effects and other commands, in this type of vertical sequencer, are given as letters, numbers or hexadecimal digits into a fixed, time-slotted grid. Fig.1 shows an example of a tracker interface: *Little Sound Dj*, the most popular music editor for Game Boy consoles. The implementation of this tracker, along with other softwares like *nanoloop* and *mGB*, will be considered in the design of the proposed product.

2.2 Rationale: Conscious Limitations

Despite being one of the cardinal points in the *Chiptune* subculture, the idea of maintaining the limitations given by the hardware is too general and vague, and a distinction is necessary. During the composition and execution of 8-bit music, two main types of constraints can be addressed:

- **Processing capabilities** – The true and interesting challenge, i.e. to try to push the hardware CPU to its limit, creating complex sounds and

²To be precise, as stated in the 8BC Chiptune Wiki (2007), the console has a fifth – and least known – channel: it is an analogue input channel that allows external synthesis on cartridge to be mixed with the sound generated by the other channels. No cartridges are known to use this channel and its functionality, though.

³Also known as *pulse wave* or *square wave*.

SONG					PUI
	PUI	PU2	WAV	NOI	
00	---	---	---	---	128 1 2 W N
01	---	---	---	---	
02	---	---	---	---	
03	---	---	---	---	
04	---	---	---	---	
05	---	---	---	---	
06	---	---	---	---	
07	---	---	---	---	
08	---	---	---	---	
09	---	---	---	---	
0A	---	---	---	---	P S C P T T
0B	---	---	---	---	
0C	---	---	---	---	
0D	---	---	---	---	
0E	---	---	---	---	
0F	---	---	---	---	
0F	---	---	---	---	

Figure 1: *Song* Screen on the popular LSDJ tracker for Game Boy (Kotlinski, J., 2007)

tracks on a level that was considered unachievable, given the limited digital resources.

- **User interface** – Even though someone could disagree with this opinion, from a practical view, limitations in the UI can be considered nothing more than an obstacle in the production process. If we take the DMG-001 as an example, being constrained by a D-pad and four push buttons has nothing to share with the concept of taking out deep and articulated music from a 4.19 MHz CPU with 8 KB of RAM.

So, it is clear that a limitation in the UI is unnecessary and – most of the time – unwanted. For this reasons, the aim behind the proposed product is to renew the link between chiptune musicians and instruments, expanding the interactive capabilities of the Game Boy, without perverting the characteristic soundscape and the core aspects of its composition process.

The product is aimed at whoever wants to use a Nintendo Game Boy as a tool to create music rather than as a videogame console: music producers, musicians, beginners and experts will have the chance to interface more easily with the 8-bit music culture, both in composition and live performance contexts.

3 Project VolcaBoy

3.1 Concept

As previously introduced, the core idea behind the product is to create a new and modern interface to mediate between the sound engine and the consumer. The key features which can be outlined from the first design stage are the following, in order of importance:

1. **Ease of use** – The main feature of the product. A modern, straightforward interface to interact with the system. The preliminary idea is to create a sequencer-like product, trying to mimic with the hardware what trackers like LSDJ do via software.
2. **Portability** – In order to keep the essence of the Game Boy, which is the most iconic portable console, the product must be small, easy to carry, and battery powered.
3. **Compatibility** – If possible, the choice is to realise a product able to connect and share data with other digital musical instruments, to expand the boundaries of chiptune music outside its strictest definition.
4. **Aesthetics** – Although not mandatory, appearance is important. Therefore, attention must also be paid to the aesthetic part of the design, rather than focusing only on functionality.

As it could be guessed by the above list, during the initial design stage it became more and more obvious the similarity of this potential product with a type of musical tools already on the market: the *Korg Volca Series*. On Korg website (2019), the volca series is presented as follows:

“This series includes a variety of units such as synthesizers, drum machines, and bass synths that all play a specialized role in your performance or studio setup. Users can perform with multiple units using the tempo sync included on all volcas. The compact design of the Korg volca series is packed with limitless possibilities.” (Korg Inc., 2019)

Therefore, the decision was taken to use this series of synthesizers as a principal inspiration for what concerns dimensions, functionalities and overall appearance of the final product. To better understand the components and the layout of the above mentioned synthesizers, in Appendix A can be found the assembly sketch of a Korg Volca Bass.

3.2 System Architecture

In the following section the overall architecture of the product will be explained. Since the design and production process is not conventional – i.e. the product is not to be manufactured from scratch, like in a standard, professional context –, this part of the paper is separated in three main subsections, in order to follow the steps of the manufacturing process: hardware, DMG-001 modifications and software.

3.2.1 Hardware

Controls In order to translate a tracker’s GUI into a physical interface, the structure of a step sequencer turned out to be the most suitable. Hence, the

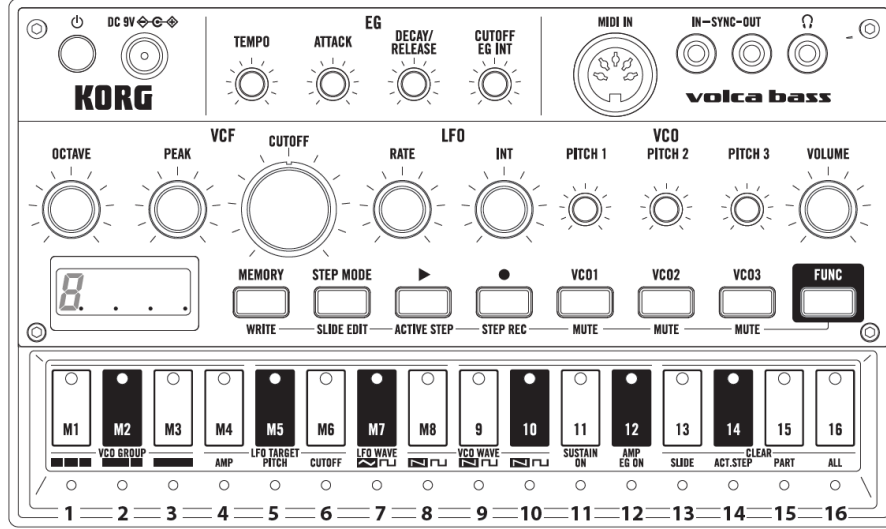


Figure 2: Front view of a *Korg Volca Bass* (Korg Inc., 2013).

product would have a grid of push buttons to map each channel and beat of the DMG-001, a digital rotary encoder to choose the note for a selected beat and an analog potentiometer to control the level of the master volume⁴.

Furthermore, considering that, in a professional environment, the entire project depicted in this paper could be associated with the prototyping stages of a commercial product the decision was made to keep the original controls of the console for debugging purposes. But, while proceeding with the preliminary blueprint, a problem related to this design emerged: sticking blindly to the step sequencer idea would mean to have a grid of 4 x 16 buttons⁵, which is an issue when considering the portability of the system.

User Interface To address the problem, it has been chosen to reduce the number of physical controls and implement a digital UI with minimal features, to balance between the *ease of use* and the *portability* of the product. Therefore, the push buttons grid will be reduced to 4 x 8 and splitted in two pairs of rows: the first will handle two channels from beat 1 to 8, the second will handle the same two channels from beat 9 to 16. Then, the channels will be grouped in *banks*:

⁴This distinction between volume and note potentiometer is made necessary by how the signals will be routed: the master volume will be hijacked directly from the console's motherboard, so it will be a continuous, analog signal. On the other hand, the note selection will be made on the Arduino embedded software, and it will be a discrete, digital signal – this topic will be expanded in 3.2.3.

⁵Four rows for the console channels and sixteen columns for the beats because, working on hexadecimal values, most of the trackers had this time subdivision.

- **Bank A** – Channel 1 and 2;
- **Bank B** – Channel 3 and 4;

Concretely, the grid is to be built following the electric schematic of a *keyboard matrix*⁶, and two additional buttons will be added to select which bank to open in the sequencer grid. Finally, a dedicated LCD screen will show which bank, parameter, note or control is being tweaked. Figure 3 shows how, potentially, the UI could appear, while in Appendix B can be found a complete preliminary design of the product, which has been modeled in SolidWorks and then exported in a standard orthogonal projection format.

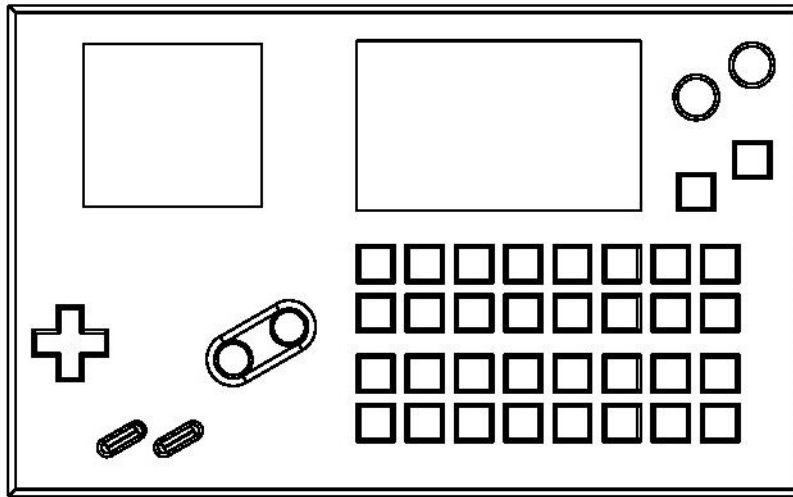


Figure 3: Front view of the first blueprint of the product.

3.2.2 DMG-001 Mods

As anticipated, an important part of this project involves the modification of the Game Boy to modernize its structure and implement additional features. Luckily, well documented work has been made by the community in previous years, so the design process in this section can be significantly reduced taking advantage of the available online resources. The only obstacle here will be to find the most convenient and efficient way to merge together the single mods. Obviously, all the sources will be properly referenced and quoted throughout the duration of this stage, but at the moment the following tutorials are planned to be used:

- **Prosound Mod** – To amplificate and clean the sound coming out from the console and to add audio sockets more suitable for a music production

⁶Refer to Trybus, M. (2013) for a general explanation of this concept.

context (e.g. 6.3 mm stereo jack, RCA connectors...). For this mod the tutorials from 8bitjunkie (2014) and Benn Venn (2018) will be followed.

- **ArduinoBoy** – The most important mod to be done, it will give the console the ability to send and receive midi messages and therefore allow the Game Boy to communicate with the controls described above. This step will require an Arduino board and some electronics, but the documentation provided by trash80 (2017) is detailed and exhaustive.
- **Screen Backlight and Bivert** – This is only an aesthetic mod, to enhance the contrast and the readability of the original screen. It is functionally useless, since all the parameters will be shown on the second screen (see 3.2.1) but, since the appearance of the finished product matters as well, it will be taken into account. Tutorials from RetroRepairs (2019) and This Does Not Compute (2016) will be used.
- **Rechargeable Battery Mod** – Another key point of the proposed product will be the portability. This modification isn’t straightforward like the others, because the console isn’t the only component that needs to be powered, so a deeper analysis is required to design a solution for the specific case.

In Appendix C is depicted a comprehensive part list for both the hardware construction and the console modifications.

3.2.3 Embedded Software

At the moment, the product is planned to contain two Arduino boards, so a necessary distinction has to be made: in this paper, for “embedded software” is meant only the code which will be written and deployed from scratch. Vice versa, the code made by third-party developers (e.g. the software running on the Arduino board that will manage the stream of MIDI messages) has been covered in 3.2.2. That said, the embedded software will run on an Arduino Nano 33 BLE and the following features are to be added:

Feature	Type
Push buttons grid mapping	Back-end
Note rotary encoder mapping	Back-end
Conversion from UI (i.e. beat/note selected) to MIDI	Back-end
Tempo	Back-end
RGB LED mapping	Front-end
GUI	Front-end

Table 2: Key features of the embedded software.

To conclude and summarise this section, in Figure 4 a schematic of the system architecture can be found. The scheme mainly follows the rules to deploy a Unified Modeling Language (UML) *Implementation Diagram* (Stevens, P. and Pooley, R., 2006).

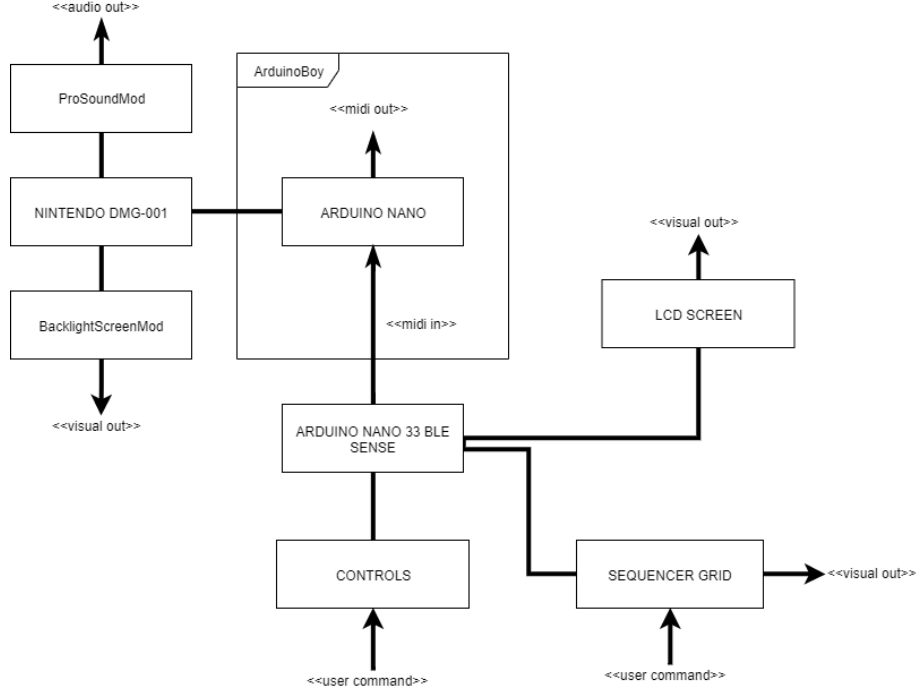


Figure 4: UML-style implementation diagram of the system.

3.3 Production

Resources Most of the literature resources that are planned to be used during the production stage are listed in the References section of this paper and in the appendices. About the production resources, a *DIY* approach has been chosen for the project, being at once the most cost effective and the most rewarding in terms of knowledge and practical experience. In addition to programming most of the software from scratch, programs like SolidWorks, PSpice, Fritzing, Simplify3D will be used to design and simulate the behavior of the system. During the production stage, tools such as soldering irons, 3D printers and dremels will be employed in the construction of the hardware, consisting of the electrical circuitry and the external enclosure.

Schedule Given the size and complexity of the project, a well organized timescale is mandatory. For the definition of the various stage, inspiration was

taken by the work of Cunningham (2018) and Nandakumar (2018). The idea is to embrace a typical electronic product development cycle, so the entire process will be splitted into four categories listed in Table 3. The detailed version of the timescale – made using the online project planning tool *TeamGantt* – can be found in Appendix D.

Stage	Time Period
Planning	30/09 – 24/11
Embedded Software	28/10 – 15/12
Hardware	11/11 – 05/01
Production	08/11 – 10/01

Table 3: Product development stages.

Health and Safety For what concerns the health and safety, in a broad process such as the one considered it is difficult to predict each potential risk and issue, since the production line is prone to changes and other manufacturing tools could be employed in the development. However, a risk assessment table has been drafted and can be found in Appendix G, with the principal potential risks at the moment.

4 Conclusions

To conclude, in this section are listed and discussed the criteria to evaluate the outcomes of the proposed project.

Minimum Viable Product First, to evaluate the final result of this production process, the Minimum Viable Product has to be defined. In this case, since the primary aim of the project is to create a new way for a chiptune producer to interact with a Game Boy, the MVP must show, at least, a different user interface provided with MIDI functionality.

Evaluation Criteria From this point, the product will be evaluated according to the key features listed in Section 3.1. For example, on a scale from 1 to 5, the outcome could be assessed as follow:

- **1/5** – The product fails to satisfy the features defined in the MVP, a non-working prototype has been made and no documentation has been created;
- **2/5** – The product satisfies the MVP definition, although the prototype works only under specific conditions and the documentation is vague and inconsistent;

- **3/5** – The product adds functionalities to the MVP definition, the prototype works in most testing conditions and basic documentation has been provided;
- **4/5** – The product shows most of the functionalities blueprinted during the design stage, the prototype is fully functional and basic documentation has been provided;
- **5/5** – The product is exactly as described during the design stage, the prototype is fully functional and aesthetically appealing, exhaustive and detailed documentation has been provided.

Finally, in Appendix F there is a 3D render that simulates how the product could potentially appear if the design and production process proceed as planned.

References

- BennVennElectronics (2018) *GBC Amp Mod* [Online video], 27 February 2018. Available from: <https://www.youtube.com/watch?v=iyG_c3MIVwI> [Accessed 24 October 2019].
- Collins, K. and Kapralos, B. and Tessler, H. and Paul, J. L. (2014) *The Oxford Handbook of Interactive Audio*. Oxford: Oxford University Press.
- Cunningham, G. (2018) Product Development Stages *CUNNINGHAM CONSULTING - Hardware and Software Engineering*, 26 August 2018 [Online]. Available from: <<https://www.cunninghamconsulting.net/2018/08/26/product-development/>> [Accessed 1 November 2019].
- Korg Inc. (2013) *KORG Analogue Bass Machine volca bass (volca-bass) - SERVICE MANUAL* [Online]. Available from: <http://www.ksadhu.niezba.org/sajty/korg_volca_bass_sm.pdf> [Accessed 1 November 2019].
- Korg Inc. (2019) *Volca Series Lineup* [Product page] [Online]. Available from: <<https://www.korg-volca.com/en/>> [Accessed 5 November 2019].
- Kotlinski, J. (2007) *Little Sound Dj v3.7.4 - Operating Manual* [Online]. Available from: <https://www.littlesounddj.com/lsd/latest/documentation/LSDj_3_7_4.pdf> [Accessed 20 September 2019].
- Marquez, I. (2014) Playing new music with old games: The chiptune subculture. *G— A— M— E Games as Art, Media, Entertainment* [Online], 1 (3), pp. 67-79. Available from: <http://www.gamejournal.it/wp-content/uploads/2014/04/GAME_3_Subcultures_Journal_Marquez.pdf> [Accessed 20 October 2019].
- Nandakumar, M. (2018) *Using UML - Software Engineering with Objects and Components*. 2nd ed. Edinburgh: Pearson Education Limited.
- Nintendo (2019) *Game Boy, Game Boy Color, Game Boy Pocket Technical Data* [Support page] [Online]. Available from: <<https://www.nintendo.co.uk/Support/Game-Boy-Pocket-Color/Product-information/Technical-data/Technical-data-619585.html>> [Accessed 20 October 2019].
- RetroRepairs (2019) *Game Boy DMG - Backlight and Bivert Mod - Fixing eBay Junk* [Online video], 15 April 2019. Available from: <<https://www.youtube.com/watch?v=-nZD3q4i-Cc&t=1026s>> [Accessed 6 November 2019].
- Stevens, P. and Pooley, R. (2006) *Lean Product Management: Successful products from fuzzy business ideas*. Birmingham: Packt Publishing.
- TEDx Talks (2013) *Chiptune: Pushing the Limits Using Constraints: Dan Behrens(Danimal Cannon) at TEDxBuffalo* [Online video], 15 November 2013. Available from: <https://www.youtube.com/watch?v=_7k25pwNbj8> [Accessed 27 October 2019].

This Does Not Compute (2016) *Backlight and Bivert Your Original Game Boy DMG!* [Online video], 5 January 2016. Available from: <<https://www.youtube.com/watch?v=xS8NQ3dds2k>> [Accessed 7 November 2019].

trash80 (2017) *Official ArduinoBoy Repository for serial MIDI communication to the Nintendo Gameboy*, 11 June 2017 [Online]. Available from: <<https://github.com/trash80/Arduinoboy>> [Accessed 13 October 2019].

Trybus, M. (2013) HOW TO MAKE A KEYBOARD - THE MATRIX *komar's techblog*, 2 September 2013 [Online]. Available from: <<http://blog.komar.be/how-to-make-a-keyboard-the-matrix/>> [Accessed 7 November 2019].

8BC Chiptune Wiki (2007) *Game Boy* [Online]. Available from: <https://web.archive.org/web/20080221140919/http://8bitcollective.com/wiki/index.php?title=Game_Boy#Sound> [Accessed 5 October 2019].

8bitjunkie (2014) *How to: Original Game Boy DMG Internal Prosound Mod* [Online video], 10 October 2014. Available from: <<https://www.youtube.com/watch?v=FG1Y0quuHGw>> [Accessed 29 October 2019].

Appendix A

14

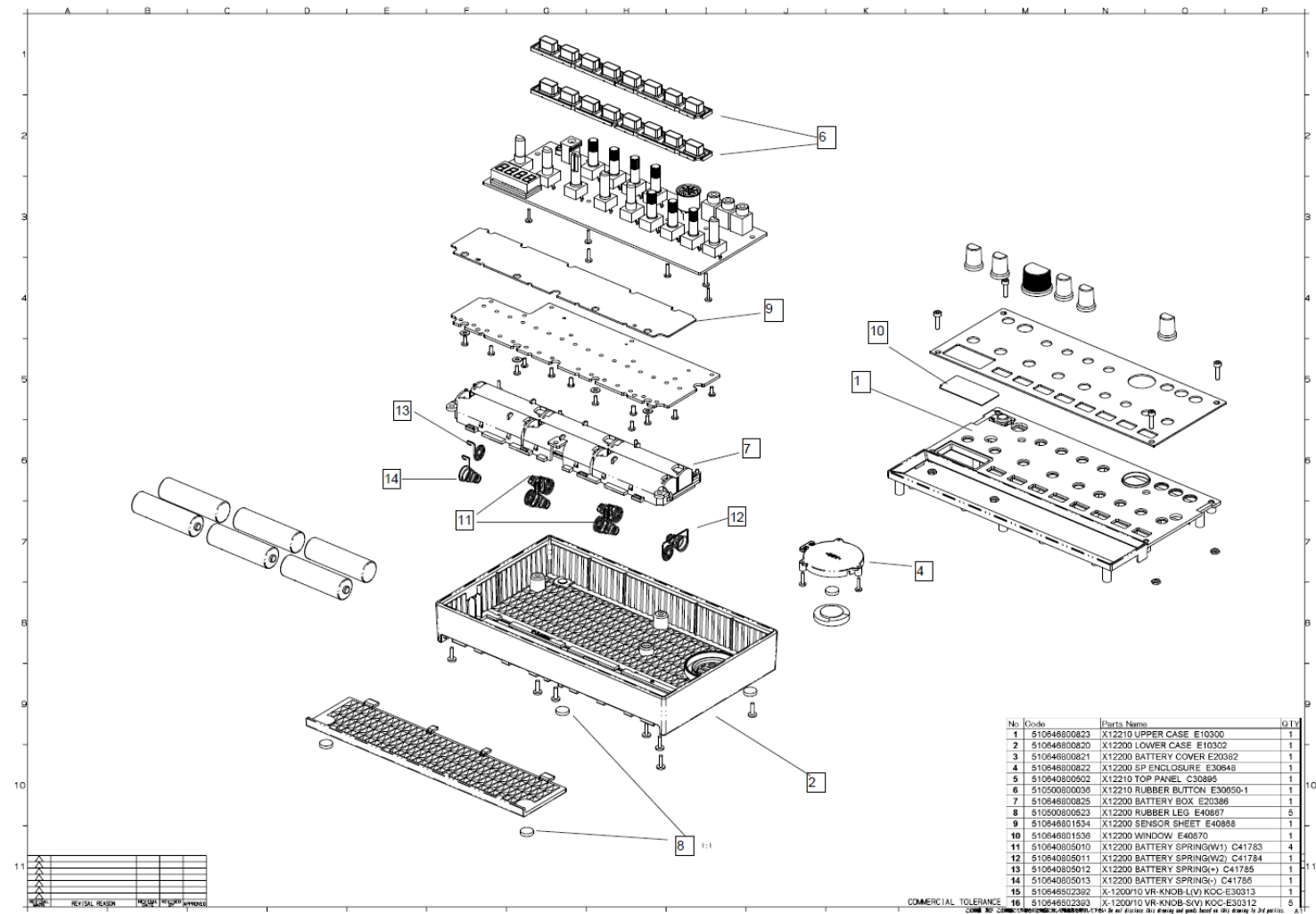


Figure 5: Korg Volca Bass assembly sketch (Korg Inc., 2013).

Appendix B

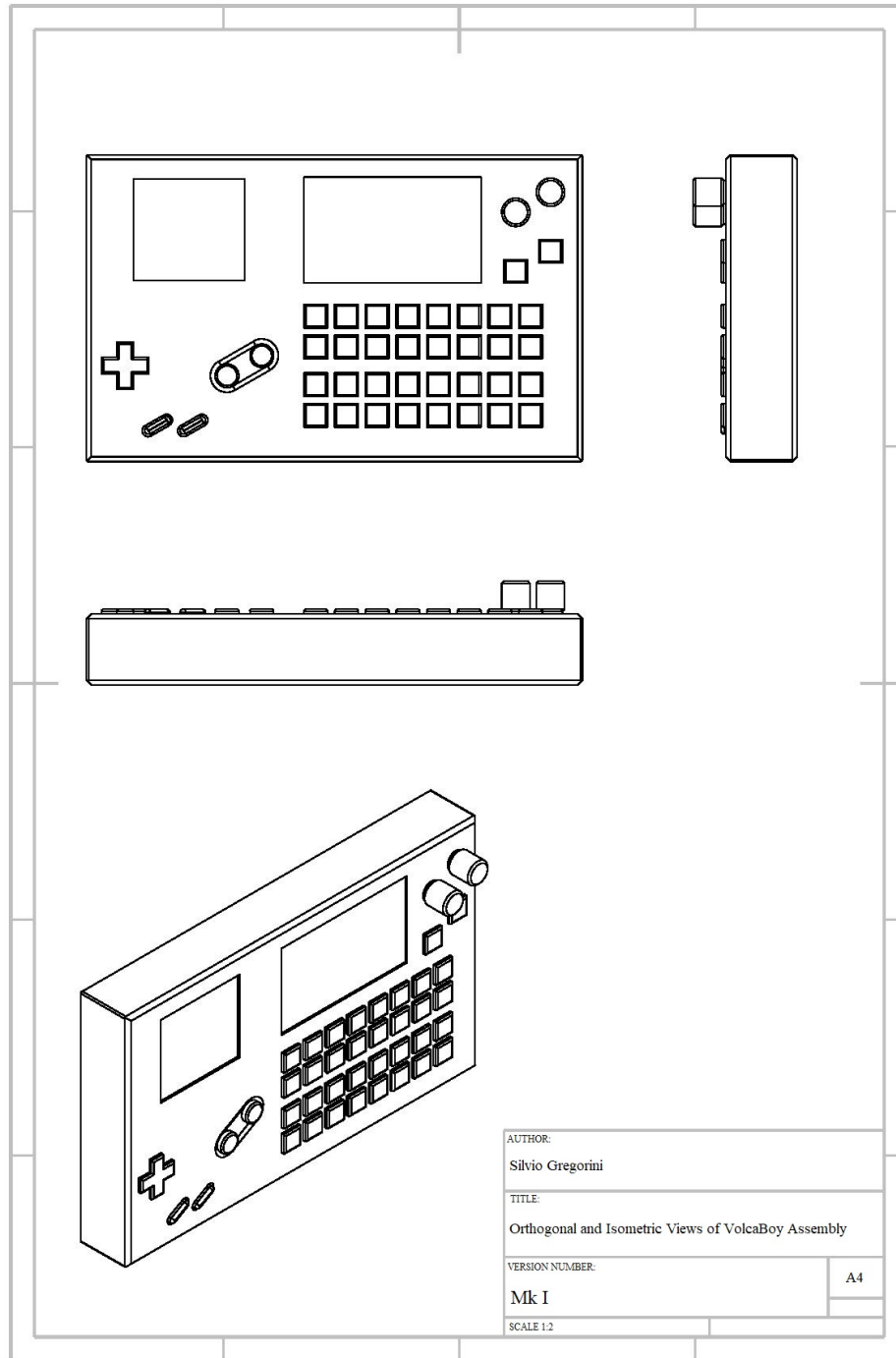


Figure 6: Orthogonal projectijon and isometric view of a preliminary version of the product, modeled and exported SolidWorks.

Appendix C

Part	Qt.	Link
Arduino Nano 33 BLE Sense	1	https://bit.ly/2Q0qRxJ
2.42" OLED display	1	https://amzn.to/2NPwWKy
Side-lit LED strips	1	https://www.sparkfun.com/products/14731
Push button	34	https://amzn.to/2WYsJrX
EC11 digital rotary encoder	1	https://amzn.to/2pSIuEM
Single sided proto board	1	https://bit.ly/2Q1aSj0
1N4148 diodes	34	https://bit.ly/34E9x5q

Table 4: Part list for the construction of the grid and the interface.

Mod	Part	Qt.	Link
ProSound	PAM8403 digital amplifier	1	https://bit.ly/32v7qiQ
	Potentiometer	1	https://bit.ly/32tcyEp
ArduinoBoy	Arduino Nano	1	https://bit.ly/32xrrWa
	6NI38 optocoupler chip	1	https://bit.ly/2NwNeJn
	1N914 diode	1	https://bit.ly/2WUUXnA
	220 Ω resistor	2	https://bit.ly/2K2EbgZ
	270 Ω resistor	1	https://bit.ly/2pL285Q
	2 $k\Omega$ resistor	7	https://bit.ly/2NvEt2n
	Push button	1	https://amzn.to/2WYsJrX
	5 pin DIN socket	2	https://bit.ly/2oZYhBj
	DMG-04 game link cable	1	https://bit.ly/2Nt72g0
	LED	6	https://bit.ly/33vGVLK
Backlight Screen	Screen backlight	1	https://bit.ly/2qyCHEA
	Bivert chip	1	https://bit.ly/2qyCHEA
Rechargeable Battery	2000mah 2S 15 25C Lipo	1	https://bit.ly/32teqwV
	2-3S AC Compact Charger	1	https://bit.ly/36Ge21r

Table 5: Part list for the console modification.

Appendix D

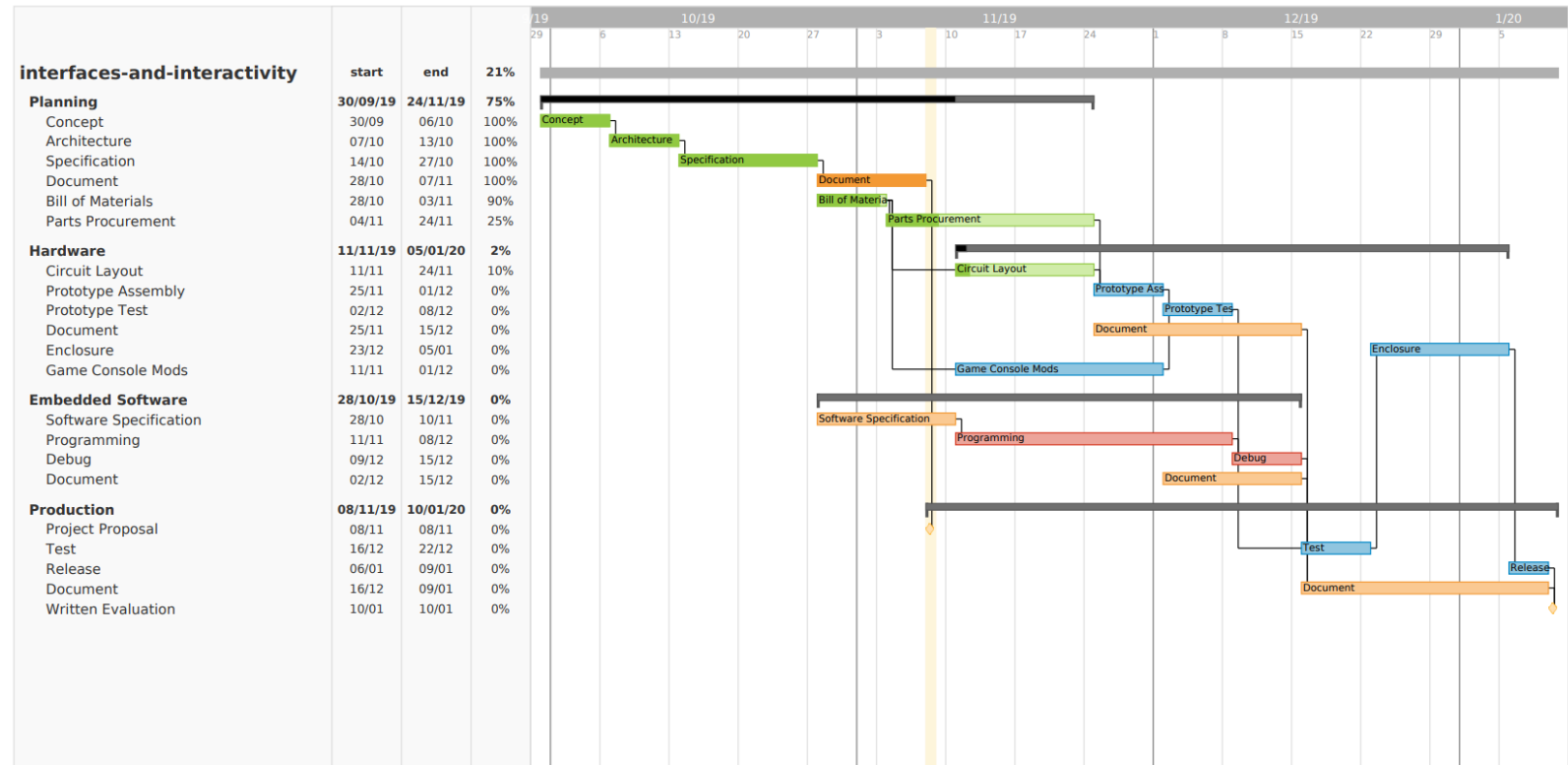


Figure 7: Gantt chart showing the product development stages.

Appendix E

Hazards	Risk	Preventative Measures	Remedial Measures
Nuisance noise/vibration	Low	- Wear ear protection	- Turn off the operating machine - In case of injury, seek medical attention
Hot components	Medium	- Check the 3D printer temperature settings - Wear protection while soldering	- Seek medical attention
Entanglement in moving machinery	Medium	- Step away when the 3D printer is moving	- In case of injury, seek medical attention
Risk of electric shock	Medium	- Turn off the main power before touching the electronics - Wear insulated gloves	- Turn off main power - In case of injury, seek medical attention

Table 6: Risk assessment table.

Appendix F



Figure 8: 3D render made with SolidWorks and Photoview 360 to show the concept look of the final product.