SketchyTuna

Exploring A Design For Screenless Creativity

Stella, Antonio; Kampmann, Baldur; Møller, Nikolaj Villefrance; Minovski, Martin Borisov; Maunsbach, Martin Lennart Wendelboe; Overholt, Daniel; Carpenter, Vanessa

Published in:
Proceedings of the 15th Sound and Music Computing Conference (SMC2018)

DOI (link to publication from Publisher):
10.5281/zenodo.1422633

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

SKETCHYTUNA: EXPLORING A DESIGN FOR SCREENLESS CREATIVITY

Antonio Stella
Aalborg University Copenhagen
astell17@student.aau.dk

Baldur Kampmann
Aalborg University Copenhagen
bkampm17@student.aau.dk

Nikolaj Villefrance Møller
Aalborg University Copenhagen
nvma14@student.aau.dk

Martin Borisov Minovski
Aalborg University Copenhagen
mminov17@student.aau.dk

Martin Maunsbach
Aalborg University Copenhagen
mmauns17@student.aau.dk

Dan Overholt
Aalborg University Copenhagen
dano@create.aau.dk

Vanessa Carpenter
Aalborg University Copenhagen
vjc@create.aau.dk

1. ABSTRACT

In this paper we explore the idea of transforming a portable speaker into an interactive music sketch pad using low and high fidelity prototyping. We present research into the current state of the art of musical sketch pads and specify the requirements for a new concept based on pressure-sensitive conductive fabric. We developed a virtual prototype and subjected it to user testing. Results from the user test led to the design and implementation of a high fidelity prototype based on a single-board computer with an additional audio interface communicating with a custom embedded MIDI controller. A real-time, loop-based musical software platform was developed as part of the high fidelity prototype. Finally, user test results are presented and discussed.

2. INTRODUCTION

Many digital music interfaces have been made in the last decades, ranging from reconstructions of ancient instruments [1] to futuristic devices [2]. Where most musical interface designs are based on a set of musical and sonic requirements, this project is a research collaboration with a speaker manufacturer and therefore has an existing speaker design as the starting point of the concept. The aim is to enhance the speaker with musical functionality while maintaining the portable aspect.

We have chosen to focus on already skilled musicians and people with some experience on looping devices or portable synthesizers. The speaker at hand has a cylindrical form and a textile cover on the body. We opted for a detachable cover that can be played by taking it off and laying it on a flat surface. We proposed different layouts and tested them on a touchscreen before making the physical version. The final hardware layout works with our own sketch pad software that allows for looping [3], playing chords, arpeggios and more. The main sketch pad software runs on a low-latency single-board computer, the BeagleBone Black, [4] connected to the Bela audio interface cape [5]. User input is registered on the sketch pad controller containing pressure-sensitive fabric attached to a circuit board, which is connected to a Teensy microcontroller [6]. Through this report we will investigate how a portable speaker without interactive musical functionality can become a music creation tool.

3. BACKGROUND

The inspiration for this project emerged from two distinct fields of experimentation; the use of loudspeakers as instruments and the use of computers for music creation.

Making the loudspeaker musically interactive would in essence transform it into a musical instrument. However, the concepts of using loudspeakers as instruments is not new. Experiments using deliberate audio feedback began soon after the introduction of the electric guitar and amplifier [7]. Deliberate audio feedback is created by placing an electric guitar in close proximity to the amplifier's speaker. The resulting effect can be heard in songs such as "Foxy Lady" by Jimi Hendrix [8] and "I Feel Fine" by The Beatles [9].

The Acousmonium project by Francois Bayle is another example of loudspeakers being used as musical instruments [10]. For a performance of Acousmonium a series of different loudspeakers were distributed in a fashion similar to how a symphonic orchestra is arranged on a stage. Different loudspeakers presented different frequency response characteristics, which was exploited by the performer to add different "voices" to the orchestra.

Some multi-purpose digital devices are sold as musical sketch pads, in the sense that the user is able to quickly pick up the device and save fleeting musical ideas for later use. Two popular examples of such devices include the OP-1 by Teenage Engineering [11] and the Organelle by Critter and Guitari [12]. The Noise app by Roli is marketed as "A handheld music studio" [13]. Noise is a free application that allow the user to quickly record a set of

loops using pre-configured instruments. These loops can then be arranged to form a complete piece of music.

There has also been considerable academic interest in prototyping musical interfaces and controllers [14], some of which focus on non-visual interaction. In the paper by Cila et. al. [15], it is argued that visual feedback to the user in some cases may be replaced by haptic or auditory feedback. This view is also supported by Tsaknaki et. al. [16] where the design of a screenless digital notepad was explored.

4. REQUIREMENTS FOR THE SKETCH PAD

The musical sketch pad we designed can be seen as a toolbox for the musician. It is portable with a selected set of tools used to sketch out temporary ideas. It includes the basic musical functionality required to play and compose music. In addition to the internal sample playback, microphones or other audio sources can be connected to the sketch pad through the audio input.

If the user wants to play without connecting an instrument that outputs sound, the sketch pad controller or external MIDI instruments are used. The controller as designed allows the user to play chords, arpeggios and control the device as a whole. A looper is implemented so the user can build beats and melodies on top of each other. This is commonly called *live looping*, as it refers to the composition of a piece of music in real-time [17]. All the features work in tandem with the looper. Audio input, chords, arpeggios, melody and percussive beats can be looped on multiple tracks, and if mistakes are made, the track can be cleared and re-recorded.

5. DESIGN CONSIDERATIONS

5.1 Layout testing

To utilize the available size of the speaker surface, various layouts were prototyped on a touchscreen laptop running Unity [18]. Four grid-based virtual pads were made as seen in figure 1, and subsequent tests focused on 4×5 square pads being used to play chords. The ability to play chords by pressing a single pad lets the user compose quickly on the fly. This allows the user to play chords and a melody on another pad at the same time. The choice of 4×5 pads was due to hardware limitations and this led to having a single octave of 12 tones and 8 additional pads, where 2 of them are used for changing the octave. By holding down one of the remaining 6 buttons, the tone pressed changes to that chord type. A major chord was used as the default.

As the chord repertoire of skilled musicians stretches far beyond 6 types, another layout was proposed to make room for more chord types. By using roman numerals to choose the chord number in the major scale instead of the 12 semitones of an octave, the number of tone buttons decreased by a third. This gave an additional 4 pads for chord types, but limited the chords to the major scale, which is well suited for mainstream pop songs.

Each of the layouts has two variations. One with the chord types on the top seen in figures 1a and 1c and one

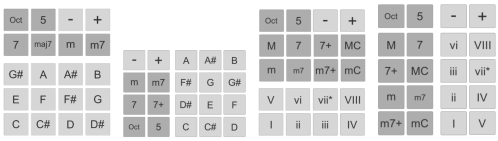

Figure 1: Chord surface layouts. Semitone layout with chord types on top (a) and side (b) and Roman numeral progression with numbers on top (c) and side (d).

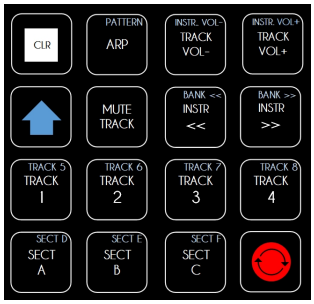

Figure 2: The control surface layout.

to the side seen in figure 1b and 1d. Users found having chord types on the side easier to play with one hand. The roman numeral layout required additional buttons to change the tonic from the default C major scale. This was solved by holding down a "key change" button on a melody grid and pressing a tone signifying the desired key.

Skilled musicians that tested the roman numeral layout noted that they were not familiar with the system which made it hard to play along to a song, if the chords were looked up on the internet or from sheet music. One non-skilled user felt it was easier to play music with the roman numeral layout, but as the target audience is skilled musicians, we diverted back to the 12 semitone octave and chose the variation with the chord types on the side.

5.2 Control surface

The looper control surface seen in figure 2 can switch *tracks*, which are audio clips stored in memory, and *sections*, which are song structure elements. For the left hand-side control surface, we used a 4×4 key grid layout. On the physical device there is a single LED above each of the keys, making it possible to reveal the state of the looper at any given time.

The two keys for managing loop tracks (REC and CLR) are located in two opposite corners to minimize the risk of accidentally pressing the CLR button and losing data while rushing to press an adjacent key. Pressing the REC key for the first time puts the looper in listening mode, clearly indicating this by putting the LED above it in a blinking state. Either playing a note or pressing the REC key for a second time will start the audio clip recording.

Pressing the REC key for a final, third time while its LED is constantly on (i.e. while recording) will store the

Figure 3: A mock up of the sketch pad controller on the speaker.

recorded audio clip in memory and immediately start looping it. The first recorded clip (track) determines the musical bar length. No metronome or tap tempo is required, giving users the ability to perform in any time signature.

There are three function toggle keys on our control button layout. The blue Shift key brings another dimension to the whole layout, allowing us to add functionality at the cost of an extra action. Shift functions are not as accessible as the primary functions and are therefore reserved for less time-critical, yet required operations, such as changing the instrument bank or switching over to a less frequently used instrument track or section. It acts similarly to the Shift key of mobile on-screen keyboards as it gets released after a single subsequent keystroke.

The Mute Track key will immediately mute the currently selected track. Its behaviour differs from clearing a track in that it instantly mutes it, whereas the CLR key would cut playback at the end of the current clip to allow stopping multiple tracks precisely at the end of the bar.

Finally, the ARP key toggles between chord and arpeggio modes, defining the way harmony is played on the right hand pads. The arpeggio and chord functions are described in more detail in section 7. A mockup of the layout as it would look on the speaker can be seen in figure 3.

6. SKETCH PAD CONTROLLER

In order to test the required features and research the optimal manufacturing techniques for fabric-based MIDI controllers, a prototype embedded device had to be designed and constructed. This involved a design stage as well as a hardware and software development stage.

6.1 Design

The hardware design is based on the user tests conducted using the virtual keyboard layouts developed in Unity. However, the layouts were modified slightly, so they could be manufactured with the tools available in-house at Aalborg University Copenhagen. This resulted in three separate rigid PCBs; each dedicated to a separate function for easy modification. As we can see in figure 4, the first PCB from the left contains the looper controls and LED indicators. The middle PCB is a motherboard for the microcontroller and the haptic motor controller. The rightmost PCB contains the keyboard for triggering musical notes and buttons for selecting the chord type. For an overview of the hardware connections see figure 5.

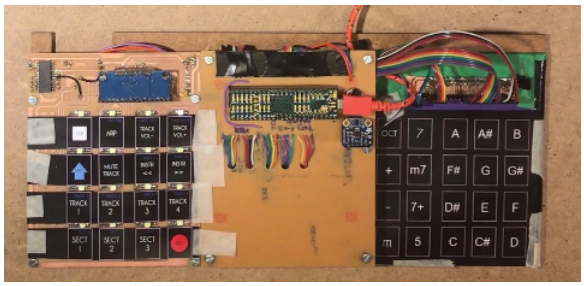

Figure 4: The sketch pad controller hardware prototype.

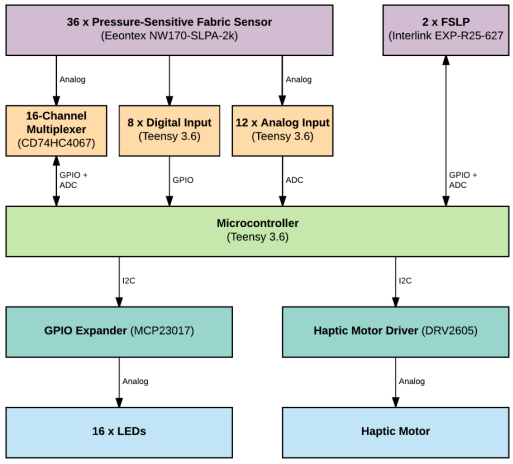

Figure 5: Hardware connection diagram.

The main difference in our design versus most other current MIDI controllers is that the final version of the controller must be physically flexible in order to be wrapped around the cylindrical speaker.

Since Bela has support for MIDI input and output via USB, it made sense to base the controller design on USB MIDI as both data and power transmission could be accomplished over a single cable. Among all of the microcontrollers available, not many have built-in support for USB MIDI making the top candidates the Teensy family of development boards by PJRC [6]. We chose to use the most recent board, the Teensy 3.6 which contains a 32 bit 180 MHz ARM Cortex-M4 processor [6] and has a large number of analog inputs.

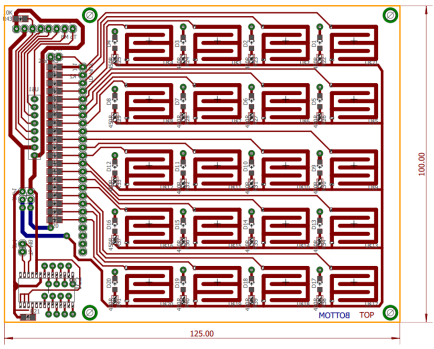

Figure 6: Sketch pad control surface PCB layout.

6.2 Hardware

Due to the complexity of the interconnections between the components, the cover prototype hardware required a custom PCB layout in order to provide us with a stable platform that would stand up to being interacted with as a musical interface.

Autodesk Eagle [19] was used to create the schematics and PCB layouts for three separate single-sided PCBs, an example of which can be seen in 6. Due to the prohibitive cost of manufacturing flexible PCBs, a Bantam Tools PCB milling machine [20] was used to manufacture the prototype boards.

The sensors used for the controller buttons are based on Eeontex NW170-SLPA-2k [21], which is a pressure-sensitive conductive fabric. When pressure is applied to the fabric its electrical resistance decreases, which is a condition that we can measure and use as a control input value.

In order to utilize the Eeontex fabric as a sensor, a technique commonly used to construct membrane switches was employed. By creating PCB traces that are interleaved but not touching, an area is created where electrical contact can be made when a membrane switch is pressed or in our case, when a pressure-sensitive fabric is pressed.

Since the Eeontex fabric changes electrical resistance when compressed, the most practical way to measure the applied pressure is to consider the fabric one half of a voltage divider and connect it to the development boards 3.3V power supply. By using a fixed resistor with a value of 3.3kΩ connected to ground for the other half of the divider, the varying resistance of the fabric can be measured as a voltage using an analog input on the Teensy development board. The output voltage seen at the ADC input can be calculated using the voltage divider formula in equation 1.

$$V_{out} = V_{in} \times \frac{R2}{R1 + R2} \tag{1}$$

Instead of finding the output voltage, it is more useful to find the resistance value maximum and minimum when the fabric is not pressed and with the maximum expected finger pressure applied. By measuring the output voltage in these two cases, we can rearrange equation 1 to calculate the value of R1. Measurements obtained with a standard laboratory voltmeter indicate an output voltage of approximately 0.2V when not pressed and around 3V with maximum pressure. This gives us range of resistance values for a 1.2 × 1.2 cm square of Eeontex material spanning from approximately 50kΩ to 0.33kΩ.

For this version of the prototype, the pressure-sensitive fabric is directly attached to the PCB using double-sided tape with a cutout in the middle to allow for electrical contact. This method is very simple and performs sufficiently for prototype testing, but after multiple presses in the same area the fabric has a tendency to get stuck and cause false triggering.

The other main active components in the design are the DRV2605 haptic motor driver, the MCP23017 I2C GPIO expander and the CD74HC4067 multiplexer. The DRV2605 and the haptic motor are located on the motherboard, and generate haptic feedback when a button is pressed. Since the fabric has limited tactile feedback, we anticipated that synthetic tactile feedback when a button press is registered would provide the user with a more natural interaction with the device. Although the effect with the current motor is very subtle, it provides a useful response when the user interacts with the interface.

The MCP23017 GPIO expander is used to expand the system with 16 additional digital pins that are connected to LEDs on the looper control board. Each LED can therefore be controlled individually without giving up pins on the microcontroller. The expander is controlled via I2C, which allows it to be placed close to the LEDs, since it only needs two data wires back to the Teensy development board.

Finally, the CD74HC4067 is used to switch the signal from the 16 buttons on the looper control board and feed them into a single analog input on the Teensy. In this way, the pressure values of multiple buttons can be measured using only one analog input. There is of course a trade-off in this situation as the sampling rate for these buttons will be divided by 16, i.e. the amount of sensors that must be measured. However, as we shall see in the following chapter, the latency caused by the multiplexer switching is negligible.

6.3 Software

The software for the user interface is based on a superloop architecture [22], which after the initial setup procedure has completed, keeps iterating indefinitely as can be seen in figure 7.

Within the superloop, the software follows a predefined path starting from the top to the bottom testing for a specific condition for each input type. If a button press is registered, a corresponding action is performed, sending out a MIDI message and triggering haptic feedback. If a valid MIDI input from another device is registered, in our case from the Bela, although it could be from any device with USB MIDI host capabilities, the LED indicators are set accordingly. If no input is registered, the program keeps looping through the code indefinitely. In order to simplify the communication between the controller and the Bela, each button and LED indicator was mapped to a specific MIDI message. Table 1 shows the MIDI mapping within the controller.

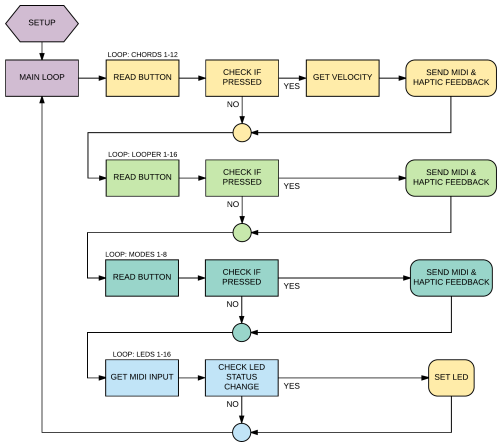

Figure 7: Sketch pad controller software flowchart.

CONTROLS	NOTE	VELOCITY	CHANNEL	I/O
Looper 1-16	0-15	ON: 127; OFF: 0	1	Output
Mode 1-8	20-27	ON: 127; OFF: 0	1	Output
Chord 1-12	30-41	ON: 127; OFF: 0	1	Output
LED 1-16	0-15	ON: 127; OFF: 0; BLINK: 64	ANY	Input

Table 1: MIDI implementation chart for sketch pad controller and software.

The same mapping strategy was employed in the sketch pad software in order enable communication between the sketch pad controller and software.

The software relies on a range of classes to communicate with the various input and output devices in the system (see system diagram in hardware section). This keeps the software architecture flexible in order to accommodate future hardware changes. The connection between the classes can be seen in figure 8.

To generate velocity values for the MIDI Note On messages, the software attempts to find the average velocity of the pressure value curve. The result is converted to a value between 0 and 127 to fit with the velocity value used in a standard MIDI Note On message. The average velocity can be determined using equation 2:

$$AverageVelocity = \frac{Y2 - Y1}{X2 - X1} \quad (2)$$

Since we are comparing subsequent samples with a fixed

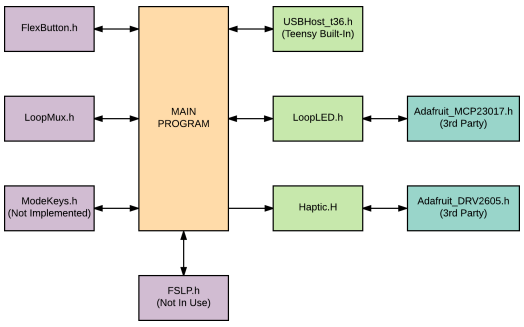

Figure 8: Sketch pad controller class diagram

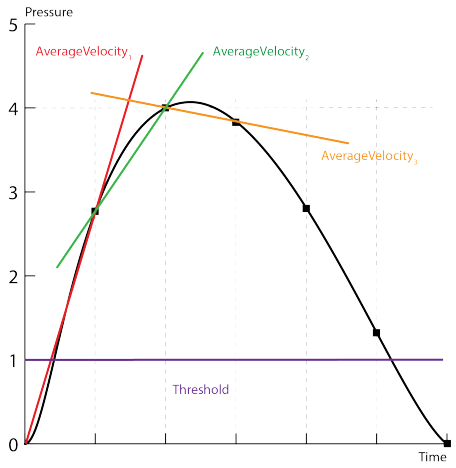

Figure 9: Arbitrary average velocities at sequential sample points.

time interval, we know that the difference between X2 and X1 is always 1 in an arbitrary time scale based on the chosen sampling rate and we can therefore simplify as seen in equation 3:

$$AverageVelocity = \frac{Y2 - Y1}{1} = Y2 - Y1 \quad (3)$$

As we can see in figure 9, the average velocity is shown for 3 different samples. The values for $AverageVelocity_2$ is used to calculate a velocity value for the MIDI Note On message, since it is the first measurement to have a pressure value higher than the threshold. All subsequent samples are then ignored until the pressure value has fallen below the threshold; after which the process is be repeated. While the pressure is above the threshold, the measurements can be used to calculate a value for MIDI Aftertouch messages, which in turn can be mapped to modulate the sound timbre or other relevant controls.

We use a 58 kHz sample rate with a 4 sample average resulting in a sample every 17.24μs measured on a Teensy LC. Since we are sampling a total of 36 buttons on 13 analog inputs (using a 16 input multiplexer for the looper control buttons) and 8 digital inputs, each buttons should be sampled once every 36 * 17.24μs, equal to 620.64μs or 0.62ms. Adding to this, switching inputs on the multiplexer will take around 0.71μs * 4 pins * = 2.84μs for each input using the normal digital output functions on the Teensy [6]. This amounts to 45.44μs or 0.04544ms in addition to the previous 0.62ms giving us a final value of 0.67ms between samples of each input.

This is not taking into account other processes such as USB MIDI processing, setting LED indicators and such.

7. SKETCH PAD SOFTWARE

The sketchpad looper (figure 10) was written in C++ 11.

7.1 Looping audio tracks

The process of recording the first loop is shown in figure 11. Any subsequently recorded tracks are added to the

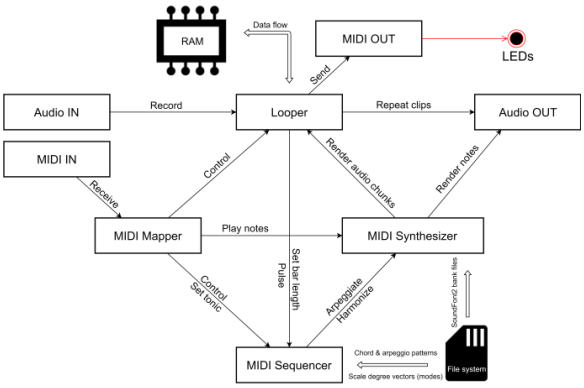

Figure 10: Looper system architecture.

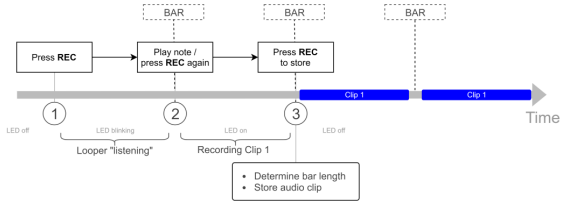

Figure 11: Looper timing: defining the bar length.

mix taking into account two important factors. One is the amount of time that has passed since the start of the current bar, which will be used to shift the new clip in time so that it will be played back in the same relation to the bar line as the original performance. The latter is also referred to as the track offset. If a new track is much longer than the bar length, repeating the clip on each bar would produce unwanted notes, especially in cases where long phrases containing complex chord progressions are involved in the performance. The second factor is the amount of bars that phrases span across, that is determined by rounding the ratio between the amount of time of the new track and the original bar length as depicted in figure 12. The latter would reset only when all tracks from all sections are cleared from memory, effectively deactivating the looper. Manual clearing of the selected track is done by pressing the CLR button.

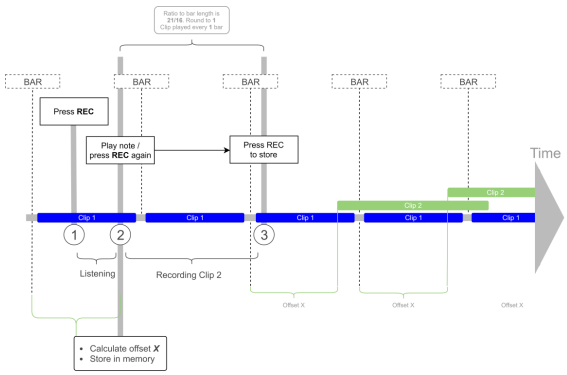

Figure 12: Looper timing: recording a second track.

7.2 Sequencer and MIDI synthesizer

The implementation was largely made possible because of the compatibility between Bela and a single-file library providing an open-source SoundFont 2 synthesizer based on the open Sforzando (SFZ) [23] file format.

Any MIDI device connected to the BeagleBoard is linked to the synthesizer making it possible to hear real recordings of instruments with dynamics control when physically playing notes on the MIDI device.

The mapping of MIDI data being transferred is established in the MIDI Mapper module. Most MIDI piano key devices are inherently supported by it and pass the Note On and Off messages over to the Synthesizer module, but any complex MIDI control surfaces would need to be manually re-mapped in order to enable users to use them for controlling the looper or changing the instrument. The MIDI Mapper is also responsible for interpreting MIDI note-on messages coming from the control surface in which the required action is signified by the pitch value of the message, as seen in table 1 in section 6.

Even though the Sequencer module is a relatively recent addition, it provides a basic technique for scheduling function calls relative in time to the looper cycle. Since the bar size is measured in samples, finding the sample length of a quarter note is done by dividing the window length by four.

One of the requirements for this project is the chord customization pad implying that chords would need to be constructed in one of the many ways possible. Four basic patterns are included in the current setup - major and minor triads with the median moved an octave up, their inversions and a neutral chord composed by two fifth intervals in 2 adjacent octaves. Changing between these is done by pressing the Shift key followed by the ARP key.

Pressing the ARP key without Shift toggled puts the sequencer in arpeggiator mode, breaking up chords in a way that will fit the timing of the looper. The sequencer, like any standard USB MIDI device connected to the Bela, provides the synthesizer module with instructions to play notes in the form of MIDI messages.

8. EVALUATION AND DISCUSSION

The prototype was evaluated with three users. The results of the evaluation were used to determine how the user experience of the prototype could be improved. The user evaluation methodology was modeled after the one used by Wu & Bryan-Kinns [24]. The testing procedure consisted of a three session setup designed to make it possible for the user to explore prototypes and provide feedback of their experience. The four sessions are: 1. introduction, 2. interaction with prototype A, 3. a short interview.

The participants all had some experience in creating electronic music; one of them was very experienced. All three participants regularly use loop features while playing or for inspiration when composing. Furthermore, they all use Ableton Live frequently, a loop based music creation software [25].

The overall impression was that the participants were interested in interacting with the prototype in order to create loop based music. Occasionally technical problems impeded the user experience.

All users found the looping feature intuitive to use and appreciated the functionality of having separate sections. Using an early version of the sketchpad controller firmware, the primary problem mentioned unanimously was the latency of the buttons. The availability of a large array of sounds was generally found stimulating. However, the order of the sounds made it difficult to navigate the available sounds. Navigating the sounds also became an issue when the participants wanted to delete a specific loop, but could not locate the instrument the sound was recorded with. Some users expressed an interest in being able to play single notes instead of chords in order to develop riffs or perform a musical solo along to recorded loops. Octave button was never utilized and the major/minor switch rarely, the arpeggiator button functionality was found non-intuitive. The pressure sensitivity provided by the chord surface was not taken advantage of because participants were more concerned about the music being on time than on the musical expressivity.

Although the sketch pad controller showed potential, a marketable product would require additional development efforts and further user testing. Also, substituting the currently used Bela with the recently announced Bela Mini would make it possible to achieve similar performance in a considerably smaller footprint and should be investigated further [26].

9. CONCLUSION

In this paper have we presented a musical sketch pad prototype. The project started as an exploration of how a portable loudspeaker could be transformed from a passive to an interactive object that enables useful musical expression and creation. Early in the initial idea generation phase we decided to transform the speaker into a musical sketchpad on which the user is able to quickly sketch musical ideas. Initial user testing was conducted using virtual and low-fidelity prototypes of the layout designs. Based on the results, a fully functional high-fidelity prototype containing hardware and software was designed and evaluated.

The sketch pad software was programmed to provide a feature set determined from the initial requirement specification and user evaluation. A final round of user evaluation was conducted using the high-fidelity prototype to evaluate the functionality. Based on the evaluation, we concluded that the prototype has potential for further development.

Acknowledgments

We would like to thank Libratone for their support. They Kindly gave us hardware to tear-up and provided helpful feedback at the early stages of the project.

10. REFERENCES

[1] A. Bellia, "The virtual reconstruction of an ancient musical instrument: The aulos of selinus," *Proceedings of the International Congress Digital-Heritage 2015*, pp. 55–58, Autumn 2015. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7413833

[2] R. Beschizza, "The thummer: A musican instrument for the 21st century?" 2007. [Online]. Available: https://www.wired.com/2007/01/the_thummer_a_m/

[3] G. Vigliensoni and M. M. Wanderley, "Soundcatcher: Explorations in audio-looping and time-freezing using an open-air gestural controller," *ICMC*, 2010. [Online]. Available: https://www.researchgate.net/profile/Marcelo_Wanderley/publication/228852385_Soundcatcher_Explorations_In_Audio-Looping_And_Time-Freezing_Using_An_Open-Air_Gestural_Controller/links/00b495231c43a0e91e000000.pdf

[4] BeagleBoard, "Beaglebone black," 2017. [Online]. Available: https://beagleboard.org/black

[5] Bela, "What is bela?" 2017. [Online]. Available: https://bela.io

[6] pjrc, "Teensy usb development board," 2017. [Online]. Available: https://www.pjrc.com/teensy/

[7] J. J. Anselmi, "Ride the feedback: A brief history of guitar distortion," Feb. 2017. [Online]. Available: https://noisey.vice.com/en_us/article/wn7ja9/ride-the-feedback-a-brief-history-of-guitar-distortion

[8] H. Shapiro and C. Glebbeek, *Jimi Hendrix: Electric Gypsy*, 1st ed. St. Martin's Griffin, August 1995, pp. 525–529.

[9] B. Miles, *Paul McCartney: Many Years From Now*, reprint ed. Holt Paperbacks, October 1998, p. 172.

[10] C. R. Sandra Desantos and F. Bayle, "Acousmatic morphology: An interview with franois bayle," *Computer Music Journal*, vol. 21, no. 3, pp. 11–19, Autumn 1997. [Online]. Available: http://www.jstor.org/stable/pdf/3681010.pdf

[11] T. Engineering, "the portable wonder synthesizer," 2017. [Online]. Available: https://www.teenageengineering.com/products/op-1

[12] Critter and Guitari, "Organelle," 2017. [Online]. Available: https://www.critterandguitari.com/products/organelle

[13] Roli, "Noise: Music made mobile," 2017. [Online]. Available: https://roli.com/products/software/noise

[14] A. R. Jensenius and M. J. Lyons, *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*. Springer International Publishing, 2017.

[15] E. G. Nazli Cila, Iskander Smit and B. Krse, "Products as agents: Metaphors for designing the products of the iot age," *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*, pp. 448–459, May 2017. [Online]. Available: https://dl.acm.org/citation.cfm?doid=3025453.3025797

[16] V. Tsaknaki and Y. Fernaeus, "Expanding on wabi-sabi as a design resource in hci," *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*, pp. 5970–5983, May 2016. [Online]. Available: https://dl.acm.org/citation.cfm?id=2858459

[17] M. H. Florent Berthaut, Myriam Desainte-Catherine, "Drile: An immersive environment for hierarchical live-looping," *New Interface for Musical Expression*, pp. 192–198, June 2010. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00530071/document

[18] Unity, "Unity 2017.3," 2017. [Online]. Available: https://unity3d.com/

[19] Autodesk, "Eagle features," 2017. [Online]. Available: https://www.autodesk.com/products/eagle/overview

[20] B. Tools, "Bantam tools desktop pcb milling machine," 2017. [Online]. Available: https://www.bantamtools.com/pages/products

[21] Eeonyx, "Ntex non-stretchy sensor," 2017. [Online]. Available: http://eeonyx.com/products/ntex-20k-nonwoven-sensor/

[22] R. Oshana, *Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications (Expert Guide)*, 1st ed. Newnes, 2013, pp. 24–25.

[23] sforzando, "Main page," 2015. [Online]. Available: http://www.sfzformat.com/index.php?title=Main_Page/

[24] Y. Wu and N. Bryan-Kinns, "Supporting non-musicians? creative engagement with musical interfaces." *In Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, pp. 275–286, 2017.

[25] Ableton, "What's new in ableton live 10," 2017. [Online]. Available: https://www.ableton.com/en/live/

[26] Bela, "Introducing the new bela mini," 2018. [Online]. Available: http://blog.bela.io/2018/02/22/bela-mini-launch/