The HoneyButtons Project

Harrison David Reiser

Abstract—An electronic generalized keyboard synthesizer is proposed, using a new experimental control scheme. The controls are mediated through software and illuminated with LEDs, allowing an endlessly reconfigurable interface.

1 Introduction

HoneyButtons is a proposed new electronic musical instrument, employing a generalized keyboard layout that dynamically transposes itself with every key played. HoneyButtons derives its name from its hexagonal arrangement of keys, resembling a honeycomb.

Generalized keyboards (also called isomorphic keyboards) arrange notes in a regular grid in which adjacent notes are separated by regular pitch intervals along each axis. [1] Such keyboards are valued for their musical symmetry: a chord or melody becomes abstracted into a 2D shape or pattern that can be replicated in any key simply by changing the starting position. [2] Chromatic button accordions have enjoyed these advantages for over a century [3], and modern-day alternatives to the traditional "seven-plus-five" piano keyboard have become available in various layouts and tunings since the advent of MIDI. [4][5][6]

Unique to HoneyButtons, however, is a novel control scheme that maps relative note intervals to absolute key positions by centering (i.e. transposing) the keyboard around each note as it is played. Along with expanding the effective range of an instrument, this "melodic relativity" further abstracts the concept of a melody into a sequence of intervals. Even when replicating the sequence by pressing keys in a specific order, the result may be in different keys, depending on which note begins at the center. This makes certain classes of repeating melodies with complex spiralling transpositions (with many of J.S. Bach's fugues as examples) elementary to perform. The concept of the relative keyboard was pioneered by Leon Gruenbaum, who designed the Samchillian keyboard around this paradigm and demonstrated its advantages in improvisational performance. [7]

The aim of this project is to make both generalized and relativistic keyboards more accessible and intuitive to the uninitiated as well as to create a general-purpose lead synthesizer suited to improvisation and compositional experimentation. As such, the project includes a research component that will analyze the experiences of musicians who use the device in order to assess the viability of such alternate control schemes.

2 Description

2.1 Interface

The prototype keyboard will have an array of 19 keys each illuminated with an arbitrary color. Key color will be used to assist the player in identifying notes and other synthesizer functions. Fig. 1 shows a typical note layout, mimicking a B-system chromatic button accordion, with half steps ($\frac{1}{12}$ octave) along the upward axis and whole steps ($\frac{2}{12}$ octave) going down and to the right. Time and money permitting, after this inital prototype is completed, another larger keyboard may be constructed or the prototype expanded to include up to 40 keys, the equivalent of two octaves.

Because the interface is entirely software-driven, the device is not tied to a particular mapping of keys to notes. The relativistic control scheme described above (henceforth called Relative Mode) is a key example: while in Relative Mode, the device will continually center the keyboard around the last note played and update the key colors appropriately, as illustrated in Fig. 2. Starting with C in the center, pressing the key just below the rightmost button will play an E note, then transpose the keyboard up by one major third such that E is now in the center. If the same button were pressed again, the next note would be a major third (or 5 half-steps) above E, which is A♭, and so on. This effectively assigns a relative pitch value to each key, instead of the usual fixed pitches. These values are represented in Fig. 3

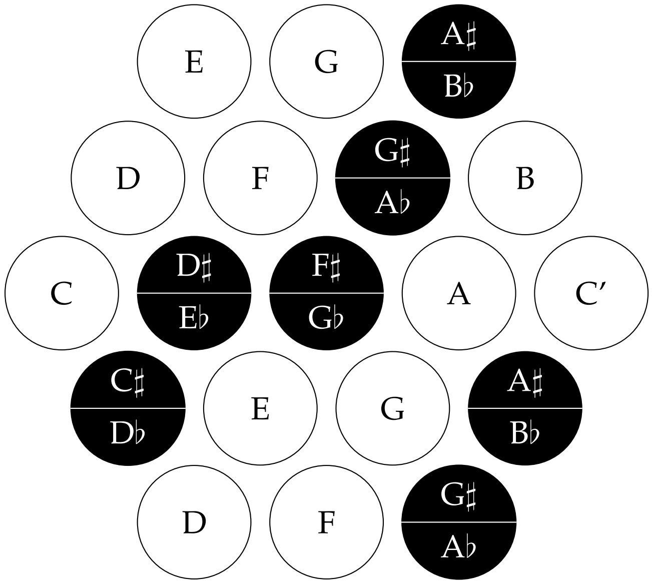

Fig. 1: bayan layout (B-system accordion)

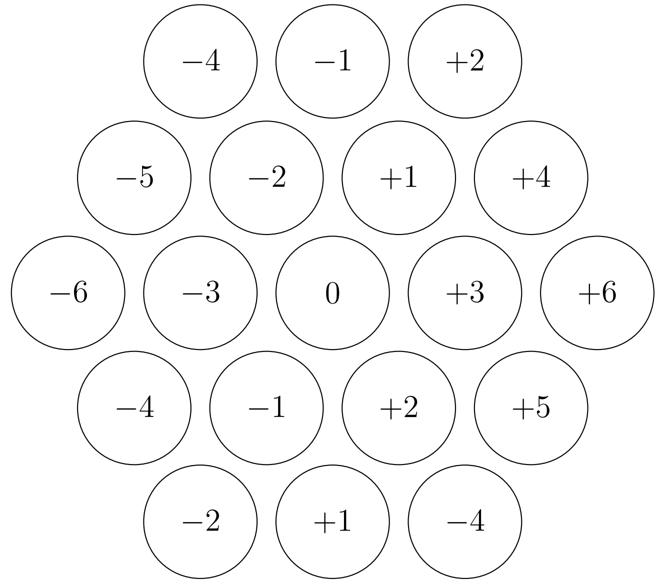

Fig. 3: relative key values for the B-system

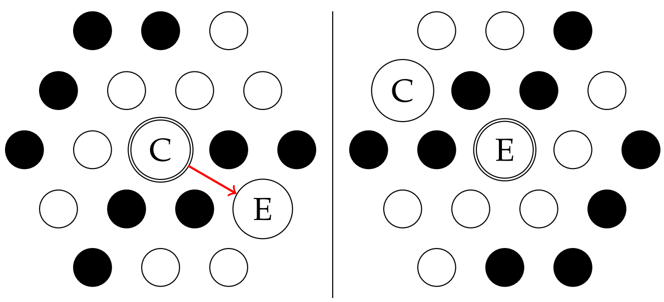

Fig. 2: demonstration of Relative Mode

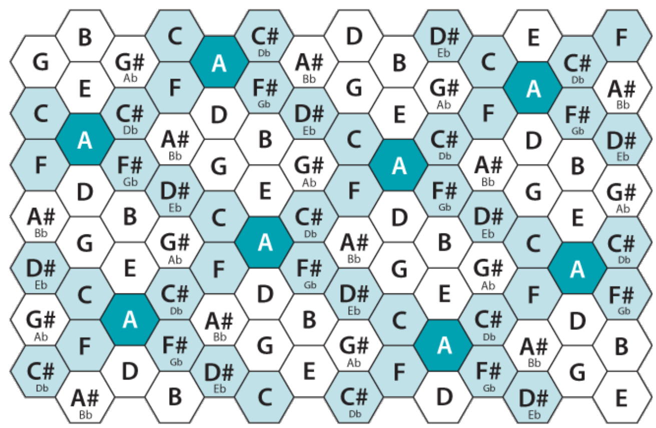

Fig. 4: Harmonic Table layout (source: http://c-thru-music.com)

by number of half-steps, using the B-system as an example.

This separation of instrument and interface also means that the keyboard can be rearranged to suit any musical style or preference. As a fundamental aspect of generalized keyboards, any layout can be represented by and generated from two distinct note intervals, corresponding to two axes along the grid. (In the case of hexagonal grids, an interval along the third axis can be found simply by adding the other two.) Countless configurations can therefore be created simply by adjusting the note intervals in either direction. For example, the fundamental intervals of the "Harmonic Table" are the major third, minor third, and perfect fifth, making a wide variety of chords and arpeggios rather simple. This layout, shown in Fig. 4, is the default mapping of the commercial AXiS and Opal MIDI controllers, and has the advantage of spanning over two octaves on a 19-button Honeybuttons at the price of only being able to reach 8 notes of each octave at a time. [5]

Alternative tunings are also possible: Fig. 5 shows how the keys might be mapped for a scale with 19 equal divisions of the octave (19EDO). As with the traditional B-system, the keys to the immediate top-right and bottom-right of a given key represent the 19EDO equivalents of half steps ($\frac{1}{19}$ octave) and whole steps ($\frac{3}{19}$ octave) respectively.

Fig. 5 also shows an example of how the colored LEDs can be used as a visual aid to the musician. In 12EDO—the standard followed by almost all modern musical instruments—

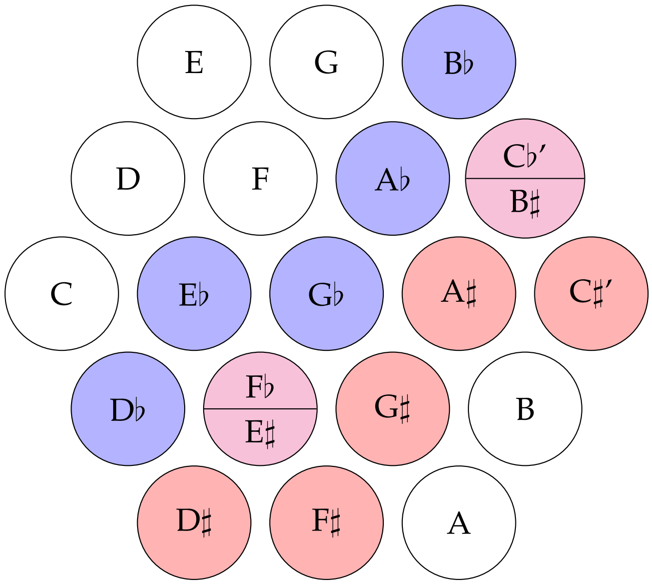

Fig. 5: a possible 19-ET mapping

sharps and flats share the same positions on the scale. 19EDO on the other hand has distinct tones for most of its sharps and flats, the equivalent of adding 7 black keys per octave on a piano. Therefore, to avoid confusion, sharps are identified with "warmer" (red) colors and flats with "cooler" (blue) colors as a sort of synaesthetic mnemonic. Similar uses of color, such as applying a gradient along the full tonal range of the instrument, can help the performer distinguish octaves and absolute pitch.

Since HoneyButtons is intended to be a largely standalone MIDI device, the LEDs and buttons will also be used to identify and control all other functions, e.g. to change the MIDI channel and voice and switch between various keyboard layouts. Additionally, an interactive tutorial mode for teaching and practicing the use of the keyboard (with or without Relative Mode) will ideally be available before the start of user trials, but is still in the speculation phase. An external pedal will be used to access these features, either as a modifier key for chorded commands or as a "home" key that accesses a central menu.

Discovering intuitive interfaces for these extra features may require much iteration and testing; in fact, because of the experimental and highly reconfigurable nature of the device, a nearly unlimited amount of time could be spent simply developing and testing the UI. Therefore, a minimal set of features has been designated as the primary project goal, with several open-ended secondary goals to be explored after the core functionality is fully tested at each milestone. (see section 3)

2.2 Hardware

The prototype keyboard consists of an array of arcade-style pushbuttons mounted with high intensity RGB LEDs. The three colors are diffused through the frosted button plastic to produce a full color spectrum, which at near-maximum output is visible even in lighted environments. The TLC5940 chip, a specialized PWM LED driver, will be used in order to provide consistent power to the LEDs. Multiple chips can be daisy-chained through its serial interface to conserve pins, similar to I^2C.

The synthesizer module will be implemented in Verilog and synthesized onto a Papilio One, an open-hardware FPGA development board centered around the Spartan-3E. Code for this synthesizer is based on a previous digital design project involving five other ECE 5710 students. [8] Its feature set is currently being expanded to include portamento (i.e. sliding pitch) and wavetable synthesis, which would enable a rudimentary form of custom synth voices by supplying recorded or artificial waveform samples. Due to the restrictions of Relative Mode, polyphony will not be supported, but additional features such as digital filters and effects are open to possible future development.

The Papilio platform includes an Arduino-compatible soft core based on the AVR8 processor and supports a variety of peripheral modules known as "wings". To interface with other audio equipment, the Papilio One will be augmented with a MIDI Audio Wing, which provides a MIDI port and ⅛″ stereo audio jack. The MIDI port will allow HoneyButtons to act as a MIDI controller for external devices, while the built-in low-pass filter allows analog audio output from a digital source by first using the FPGA to perform Delta-Sigma encoding, i.e. converting digital audio samples into a high-speed 1-bit waveform. (See [9] for an example.) For the sake of testing the prototype, however,

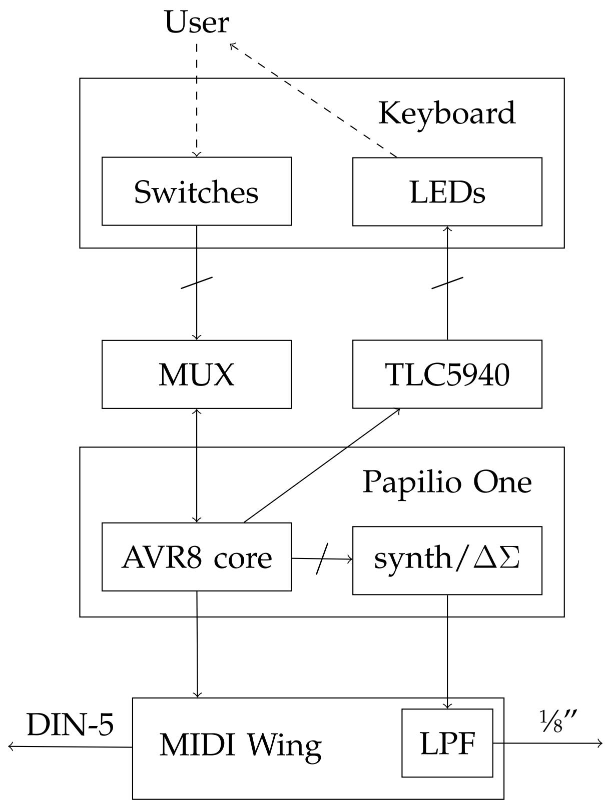

Fig. 6: block diagram of HoneyButtons

initial audio output will consist only of a single square wave sent directly to the audio port.

Software running on the emulated Arduino will also scan the keyboard matrix for button presses, send color data to the LED drivers, and send audio control signals to the synthesizer. A top-level diagram of these connections is given in Fig. 6.

Since up to 20mA will need to be supplied for the equivalent of 57 high-power LEDs (19 times one for each color component), power usage is a prime concern. Based on Sparkfun's reported forward voltage ratings, one tri-color LED can use up to 168 mW. This means that the lighted keyboard alone can draw over 3 W, which approaches the capacity of the Papilio (800 mA $\times$ 5 V = 4 W) even when using a wall adapter or battery instead of USB for power. This leads to the consideration of alternative power sources.

An estimate of the maximum power consumption of the finished instrument is given in Table 1. This does not include the MIDI Audio Wing, as an official power rating could not be found and likely varies with the impedance of the audio output device. Since unamplified headphones typically require 10–100 mW, though, and most other sound systems have their own powered amplifiers, it's safe to say that the peripheral will draw much less than a watt, making net power usage <10 W. A dual 12/5V 2A adapter of the kind used by external hard drives can comfortably supply this much power, and the Papilio One conveniently accepts auxiliary power sources.

TABLE 1: Estimated power usage

RGB LEDs	20 × 168	mW = 3.36	W
TLC5940	4 × 1.0	W = 4.00	W
HC4052	3 × 500	mW = 1.50	W
Papilio One	1 × 115	mW = 0.12	W
		8.99	W

All parts will be or are already sourced from retailers that deal in discrete units. A current bill of materials is given in Table 2 alongside vendor contacts. The list does not include basic circuit components such as wires and resistors, nor any materials consumed in prototyping such as protoboards and solder.

3 Execution

The status and estimated duration of the following tasks are summarized in Fig. 7.

3.1 Pre-project

The following tasks are already in progress and will be completed before the start of project work in Fall 2012.

1) Circuitry design to safely power and control all sensors and LEDs.
2) Drafting a complete electrical schematic that integrates and specifies pin mappings for the keyboard, FPGA, and peripherals.
3) Cataloguing and acquiring all necessary circuit components.
4) Functional specification of musical interface software.

3.2 Prototype

The prototype will be developed in three stages, after each of which a discrete portion

TABLE 2: Bill of materials

Vendor	Part no.	Description	Price
Gadget Factory	18	Papilio One 250k	1 @ $49.99 = $49.99
support@gadgetfactory.net	58	MIDI Audio Wing	1 @ $14.99 = $14.99
Sparkfun Electronics	COM-10818	RGB diffused common anode LED (25-pack)	1 @ $19.95 = $19.95
customerservice@sparkfun.com	COM-09907	74HC4052 Serial/Analog Mux/Demux	3 @ $ 0.95 = $ 2.85
	COM-10136	TLC5940 PWM Driver	5 @ $ 5.95 = $29.75
GroovyGameGear.com	238	Electric ICE 2 lightable horizontal pushbutton	20 @ $ 2.75
support@groovygamegear.com	—	+ True-Leaf Pro switch	+ $ 1.49 = $84.80
	210	Red female quick disconnect .187″ (25-pack)	1 @ $ 2.49 = $ 2.49
Amazon.com	B000MGG6SC	110V AC to 12/5V DC 4pin Molex 2A adapter	1 @ $12.94 = $12.94
orders@amazon.com	B000BSHG8K	internal drive 4pin extension cable 12″	1 @ $ 2.05 = $ 2.05
	B001331MRI	anodized aluminum hexagon cake pan 8x3″	1 @ $16.99 = $16.99
	B0002JENOS	Wüsthof 2036 bamboo cutting board	1 @ $ 9.99 = $ 9.99
			$246.79

of its functionality will be able to be demonstrated. After all three milestones are met, the device will be (minimally) feature-complete.

3.2.1 *Keyboard*

Deadline: August 1st

1) Assembly and testing of LED power circuitry to adjust white balance.
2) Development of TLC5940 control software and color-mixing library.
3) Testing of color mixing/fading with a single LED.
4) Connection and testing of a daisy chain of TLC5940 chips.
5) Development of keyboard matrix scanning software.
6) Connection and testing of a single LED illuminated switch.
7) Connection and testing of a single row of switches mounted with LEDs.
8) Connection and testing of complete keyboard matrix with multiplexers and diodes.

At the start of fall semester I plan to host a demo of the keyboard responding to input with visual feedback.

3.2.2 *Audio*

Deadline: October 6th

1) Design and development of MIDI control software, temporarily mapping static note values to every key.
2) Testing control of other MIDI devices.
3) Design and development of digitally controlled oscillator in Verilog.
4) Testing audio output and frequency/volume control.
5) Development and testing of Delta-Sigma encoder.
6) Design and development of additional synthesizer effects as time allows.

At this point the device should act as a rudimentary (possibly harsh sounding) electronic musical instrument with a range of 19 notes.

3.2.3 *User Interaction*

Deadline: December 31st

1) Development and testing of keyboard generalization software with programmable intervals.
2) Development and testing of software to enable Relative Mode.
3) Construction of a suitable housing for all components and peripherals.
4) Design and development of extra functions/menus/modes as time allows.

By the end of 2012, the HoneyButtons prototype with all planned functionality will be completed and demonstrated.

3.3 Research

After the prototyping phase I plan to host trials of the device with both musicians and non-musicians, recording their experiences and opinions along with their age and musical background. The written thesis will focus on analyzing the results of these experiments as well as document the development process.

All remaining time will be devoted to iterative development of new feature ideas that may arise during testing.

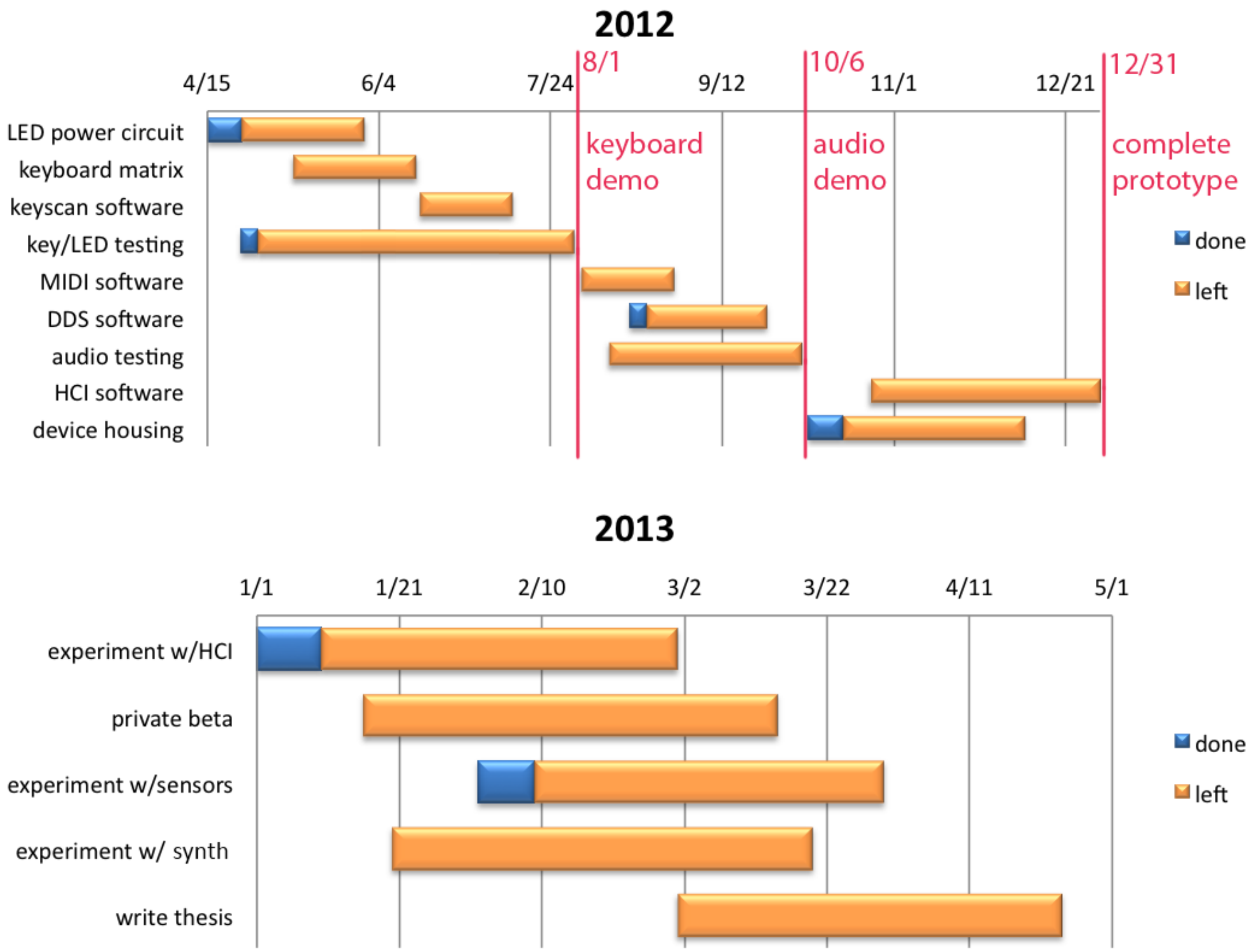

Fig. 7: Gantt chart with milestones

References

[1] R. H. M. Bosanquet, "The generalised keyboard," in *An elementary treatise on musical intervals and temperament*. London: Macmillan and Co., 1876, ch. 4, pp. 19–23. [Online]. Available: http://books.google.com/books?id=CiwDAAAAQAAJ

[2] A. Milne, W. A. Sethares, and J. Plamondon, "Isomorphic controllers and dynamic tuning: Invariant fingering over a tuning continuum," *Computer Music Journal*, vol. 31, no. 4, pp. 15–32, 2007. [Online]. Available: http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2007.31.4.15

[3] H. Doktorski, "The classical bayan," in *A Short History of the Free-Reed Instruments in Classical Music*. The Classical Free-Reed, Inc., 1998. [Online]. Available: http://www.ksanti.net/free-reed/history/classic2.html

[4] Microzone u-648. Starr Labs. [Online]. Available: http://www.starrlabs.com/products/keyboards/microzone-u-648

[5] The axis-64 pro midi controller. C-Thru Music. [Online]. Available: http://www.c-thru-music.com/cgi/?page=prod_axis-64

[6] Tonal plexus keyboard layout. Hπ Instruments. [Online]. Available: http://www.h-pi.com/TPX28keyboard.html

[7] L. Gruenbaum. About samchillian. [Online]. Available: http://www.samchillian.com/aboutsam.html

[8] S. Jenkins *et al.*, "Cmos midi synthesizer," University of Utah, Tech. Rep., 2010. [Online]. Available: http://www.eng.utah.edu/~reiser/ECE5710final.pdf

[9] J. Logue, *Virtex™ Synthesizable Delta-Sigma DAC*, Xilinx, 1999, version 1.1. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp154.pdf