Monophonic Hybrid Analog/Digital Synthesizer

"NSynth"

Divide and Conquer Version 2.0

Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Lei Wei

Group 27

Sebastian Carrion (EE)

Magdalena Dobinda (CpE)

Devin Vanzant (CpE)

Anna Wegener (EE)

Project Narrative

A synthesizer, sometimes referred to as a “synth” is an electronic hybrid musical instrument that generates audio signals combining piano and electronic technology. Synthesizers typically create sounds by generating waveforms through methods including subtractive synthesis, additive synthesis and frequency modulation synthesis. It’s been used in countless genres of music - hitting its main peak in 80s music and “alien noises” - and was a landmark invention for the history of music.

How does a synth work? It usually uses an oscillator, keyboard, filters and effects. The oscillator produces the initial sound while the keyboard allows the performer to control the tone and notes. The filters and effects then change the original sound of the notes.

Our team has decided to go with building a monosynth over a polysynth for a number of reasons - one of them definitely being the hardware/budgets constraints and limitations of a polysynth. A monosynth can play only one note at a time while a polysynth can play multiple - while this may seem like a disadvantage it actually means the synthesizer has much more control on that one sound. Monosynths are a natural fit for bass duties—and solos and portamento (the gliding or slurring between notes) are more common on monos than polys. The monosynth is considered “vintage” in the music industry and a very rare piece of hardware to come by. Mono synths sound “bigger” because they can be designed to be more harmonically rich, since the synth doesn’t have to accommodate multiple voices at once.

Building a monophonic analog synth is our first priority in our senior design scope, but in turn can also build a bridge for us to connect it to the software world through “MIDI messages”!

MIDI is an acronym that stands for Musical Instrument Digital Interface. It’s a way to connect devices that make and control sound — such as synthesizers, samplers, and computers — so that they can communicate with each other, using MIDI messages. “” This allows one keyboard trigger sounds on another synthesizer, and it makes it possible to record music in a form that allows for easy note editing, flexible orchestration, and song arrangement. “” MIDI supports up to 16 channels of simultaneous sound. Each channel can have its own melody, rhythm, and instrument.

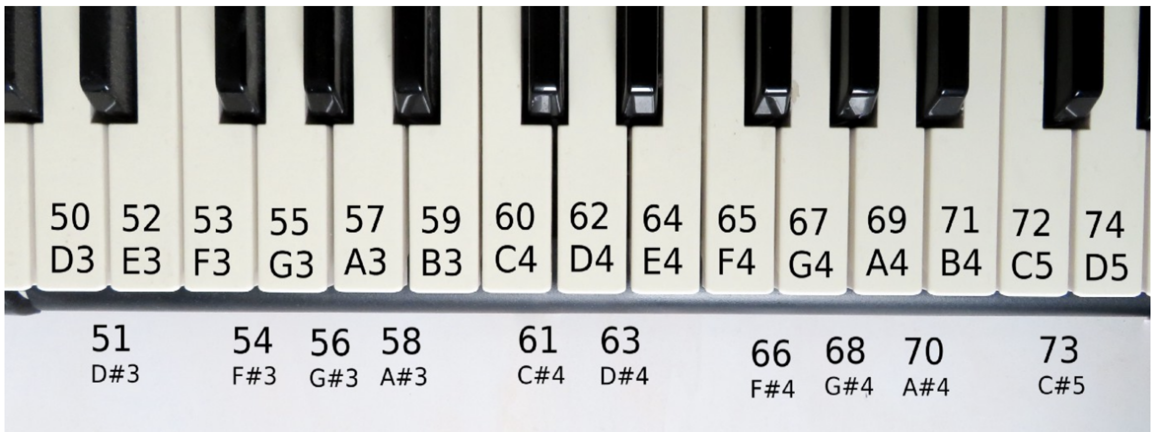


Figure 1.0 - Keys and Numerical Representations

Figure 1.1 - Formatting of MIDI Messaging

When you take a closer look as to how computers and software interprets MIDI messages, you can see that it’s much of what we as computer and electrical engineers study. You’ll find that MIDI messages are sometimes represented as binary numbers, sometimes as decimal numbers, and sometimes as hexadecimal numbers (hex, for short). Binary and hexadecimal are useful ways to represent MIDI messages because they allow us to divide the messages into meaningful groups. MIDI messages can be divided into two main categories: Channel messages and System messages. Channel messages contain the channel number. They can be further subdivided into voice and mode messages. Voice messages include Note On, Note Off, Polyphonic Key Pressure, Control

Change, Program Change, Channel Pressure/Aftertouch, and Pitch Bend. System messages are sent to the whole system rather than a particular channel. They can be subdivided into Real Time, Common, and System Exclusive messages (SysEx). For this project, we will be limiting our focus to mainly channel messages.

MIDI Channel:

The concept of channels is central to how most MIDI messages work. A channel is an independent path over which messages travel to their destination. There are 16 channels per MIDI device. A track in your sequencer program plays one instrument over a single channel. The MIDI messages in the track find their way to the instrument over that channel.

Where do we come in?

We want to build and design a monophonic analog synthesizer that talks to a MIDI controller and in turn can transfer these signals to MIDI messages to be used with a software GUI/interface. We want to create a software that reads and manipulates all the bytes that are sent from the instrument we use, to the MIDI controller, and through to the API to implement a sequencing algorithm and interface and can take these variables and create a sort of visualizer on the GUI in real time with the music playing.

Why are we doing this?

We want to create this project as a bridge between engineering and music. This project will allow us to have full control over the type of controller we build and how it interacts with the channels, notes and software. It also will bridge together software and hardware, and using a MIDI channel web API - we can evolve the software to create niche functionality some artists have a real need for, such as a visualizer controlled by MIDI commands. Most artists don't really have control over software like that, but we can make the software easy to customize and create awesome visuals using the sequences/bpm. Overall, we want to use this as a learning experience to hone our skills that we have acquired over the years. This project will implement our knowledge of embedded systems, linear circuits, electronics, circuit design, and computer science courses as a team.

Requirement Specifications

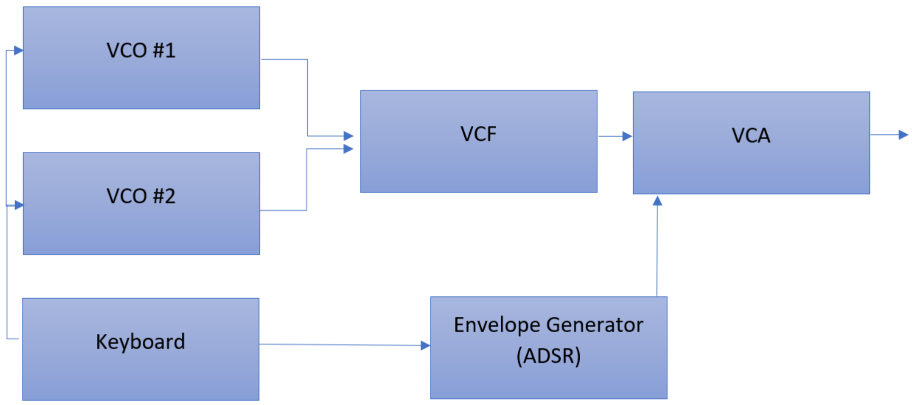
VCO (Voltage Controlled Oscillator)	Is used to create sine, triangle, square and sawtooth waveforms

VCA (Voltage Controlled Amplifier)	Used to adjust cut-off frequency through an input controlled voltage
VCF (Voltage Controlled Filter)	Used to adjust cut-off frequency through an input controlled voltage
Signal Mixer	Used to combine two oscillator waveforms
Envelope Generator	The Envelope Generator will be used to control four different stages of a sound. - Stages are composed of ADSR (Attack, Decay, Sustain, and Release)
RTDV	Real Time Digital Visualizer will be used to show the signal being applied and the output of such modulated signal
Software	Can (possibly) record samples & save them. Perhaps and audio quality option can be used such as lossless audio file format. Eg. FLAC, WAV, or AIFF. A low quality option can be incorporated as default.
MIDI Inputs	MIDI Inputs USB, Power Supply, VCF, VCO, VCA
Power Supply	Will be used as a source of controlled power to activate different components

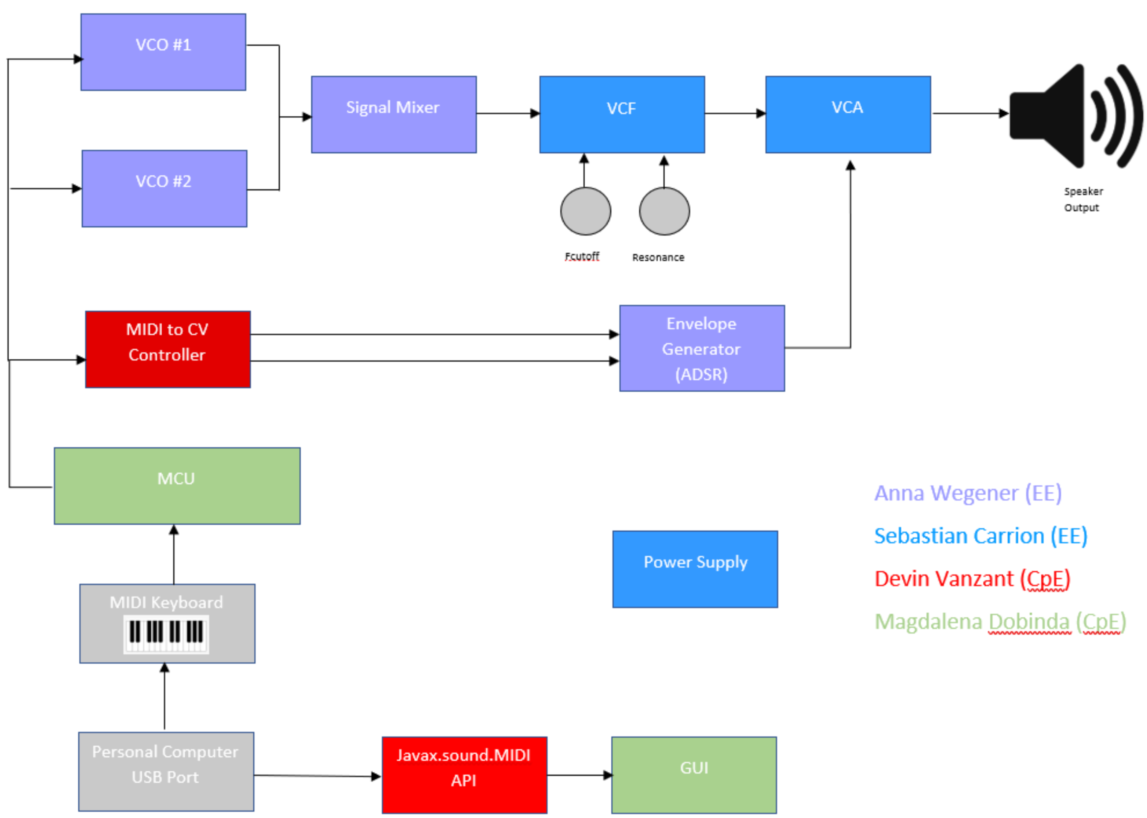
Table 1.0 - Requirements Specifications

Block Diagrams

High Level Sound Processing Block Diagram



System Level Block Diagram



Project Financing

Description	Name	Manufacturer	Quantity	Price (Est)
Hardware	VCO		3	$8.00
Hardware	VCA		1	$10.00
Hardware	VCF		1	$11.65
Hardware	Power Supply		1	$70.00
Hardware	MIDI Controller		1	$0.00
Hardware	MCV4 MIDI to CV Gate Interface		1	$166.00
Hardware	Bread Board		2	$0.00
Travel Expense	Shipping		-	N/A
Hardware	Custom PCB		1-5	N/A
Hardware	Raspberry Pi		1	$25.00
Hardware	AR envelope generator		1	$6.31
Hardware	Enclosure		1	N/A
	Miscellaneous			$100.00
	Total			$396.96

Table 2.0 - Project Financing

Initial Project Milestone Tracker

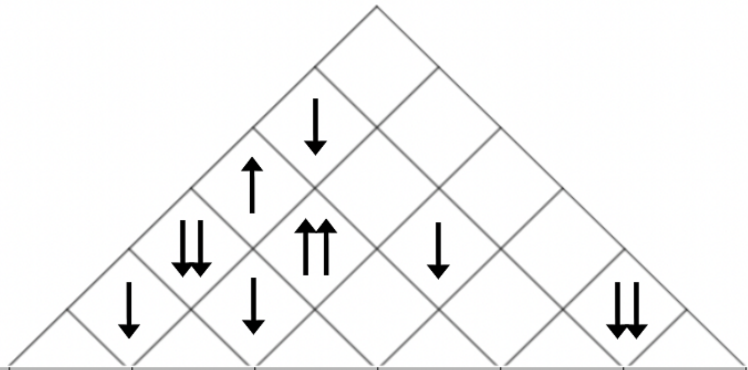
Number	Tasks	Start	End	Status	Assignee
Senior Design I					
1	Brainstorm	08/22/2022	08/26/2022	COMPLETE	ALL
2	Project Selection + Role Assignments	08/30/2022	09/20/2022	COMPLETE	ALL
3	Initial Document - Divide and Conquer Version 1.0	09/06/2022	09/16/2022	COMPLETE	ALL
3a	High level overview	09/06/2022	09/16/2022	COMPLETE	ALL
3b	Narrative description	09/06/2022	09/16/2022	COMPLETE	ALL
3c	Requirement specifications	09/11/2022	09/16/2022	COMPLETE	ALL
3d	Block diagrams	09/11/2022	09/16/2022	COMPLETE	ALL
3e	Project budget and financing	09/15/2022	09/16/2022	COMPLETE	ALL
4	Presentation with Dr. Wei/Dr. Richie	09/20/2022	09/20/2022	COMPLETE	ALL
5	Divide and Conquer Version 2.0	09/20/2022	09/30/2022	IN PROGRESS	ALL
6	Schematics / Bread Board	10/03/2022	N/A	TODO	Hardware

	Initial Design				
7	Software environment setup	10/03/2022	N/A	TODO	Software
8	Halfway Milestone (60pgs)	TBD	11/04/2022	TODO	ALL
9	Parts list - draft / final budget overview	TBD	N/A	TODO	ALL
10	Parts - Ordered	TBD	N/A	TODO	ALL
11	Software API Testing	TBD	N/A	TODO	Software
12	Second to last Milestone (100pgs)	TBD	11/18/2022	TODO	ALL
13	Final Milestone (120pgs)	TBD	12/06/2022	TODO	ALL
Senior Design II					
14	Begin creation of PCB and prototyping	~January 2023	~January 2023	TODO	Hardware
15	Initial interfacing of software to hardware	~January 2023	~January 2023	TODO	Software
16	Begin drafting final document	~January 2023	~January 2023	TODO	ALL
17	Heavy testing phase - software bug fixing	~February 2023	~March 2023	TODO	Software
18	Heavy testing phase - hardware alterations	~February 2023	~March 2023	TODO	Hardware
19	Finalize prototype	~February 2023	~March 2023	TODO	Hardware
20	Revise final document	~February 2023	~April 2023	TODO	ALL
21	Deliver final project presentation	~May 2023	~May 2023	TODO	ALL

Table 3.0 - Milestone Tracker

House of Quality

Legend	
●	Strong Relationship
○	Medium Relationship
▲	Weak Relationship
↑↑	Strong Positive Correlation
↑	Positive Correlation
↓	Negative Correlation
↓↓	Strong Negative Correlation



	Importance	Power Consumption	Computational Power	Cost per Unit	Audio Processing	Weight	Dimensions
Sound Quality	5	○	●	○	●	▲	▲
Size	4	○	▲	▲	▲	○	●
Latency	4	○	●	○	●	▲	▲
Cost Per Unit	1	▲	○	●	●	▲	▲
Features	3	●	●	●	○	○	○
Portability	5	▲	○	○	○	○	●

Table 4.0 - House of Quality

Project Goals

Basic Goals

Hardware: The hardware of this device should be able to generate analog waveforms, which are produced by the VCO. Such waveforms should be able to be modulated with the VCF, VCA, and envelope generator. The bare minimum goal of this project is to be able to output an audio signal that can be modulated by these components and heard by the user.

Software: The analog and digital components will interface with a MIDI to CV controller allowing for 1 V/Octave tracking. The synthesizer will be MIDI compatible through a USB and will be able to be interfaced to a computer.

Overall: The baseline goal of the final product is that the device will be able to generate audio waveforms that can be transcribed back into audible sounds for the user to perceive as well as letting the user be able to display said waveforms within the computer application through a GUI.

Advanced Goals

Hardware: The hardware will allow the user to see and hear the effects of their chosen setting parameters reflected on the signal's waveform.

Software: GUI interface / App Interface will actively read MIDI input from the protocol and show the current waveforms, frequency, pitch, etc. being played. Interface should be able to show the type of waveforms being produced by the synthesizer.

Overall: As we develop the device more, our goal is to enable the user to have more parameters to customize and in turn produce a wider variety of audio waveforms. These parameters will be tracked within the application on the GUI so that the user may see both the parameters affecting the generation of the waveform and the waveform itself.

Stretch Goals

Hardware: The hardware would include a module that would enable the user to connect to the synthesizer via Bluetooth. The synthesizer will be able to recreate previously generated sounds through recalling previous user parameters.

Software: Real time visualization of notes/pitches/etc being played with ability to digitally manipulate. Visualization would include waveforms as well as giving the user artistic expression through customizable visuals that are following the pitch, frequency and beat.

Overall: The long term stretch goals would be real time visualization and playback of previously generated sounds.

Conclusion

This project will allow us to have full control over the type of monosynth we build and how it interacts with the channels, notes and software.This will also allow us to experience using our own acquired software and hardware knowledge by applying concepts learning in linear circuits, electronics, differential equations, embedded systems, and computer science in a real world application. It is a great way to understand the connection music has in the engineering world and its roots, as well as how we can push the limits of what we already know we can manipulate and control regarding sound waves and transmission between channels. Overall, we want to use this as a learning experience to hone our skills that we have acquired over the years.

Citations

https://www.sweetwater.com/insync/synthesizer-basics-explained-with-audio-samples/

https://reverb.com/news/monosynth-vs-polysynth-which-should-i-buy

https://ask.audio/articles/create-your-own-midi-generated-realtime-visuals-with-openframeworks

https://www.youtube.com/watch?v=_g6DgWcF5Vs&ab_channel=FloydSteinberg

https://kulak.medium.com/web-midi-api-sending-notes-from-javascript-to-your-synth-1dfee9c57645