Reroutable Hybrid Analog/Digital Audio Synthesizer

Team 32 - Adam Wills, Connor Jones and Ishaan Datta
ECE 445 Project Proposal - Fall 2020
TA - Dean Biskup

1 Introduction

1.1 Objective

While digital plugins are by far the most popular form of audio synthesizer available on the market today due to factors like low cost and high portability/reliability, there is considerable demand for analog synthesizers - which were, until recently, considered obsolete by most major synthesizer manufacturers. Analog filters, used to shape the "brightness" of the output, have a characteristic sound which is prized by a subset of the synthesizer-buying market, but in general analog synthesizers are either extremely expensive, very limited in their offerings, extremely difficult to use or any combination of all three of these. Additionally, aside from analog modular synthesizers (which fit into the above "extremely expensive" and "extremely difficult to use" categories), there are virtually no analog synthesizers which allow you to "hack" the structure of the circuit to change the way signals are routed; this opens up a huge world of sonic possibilities those daring enough to try run the risk of damaging potentially expensive equipment.

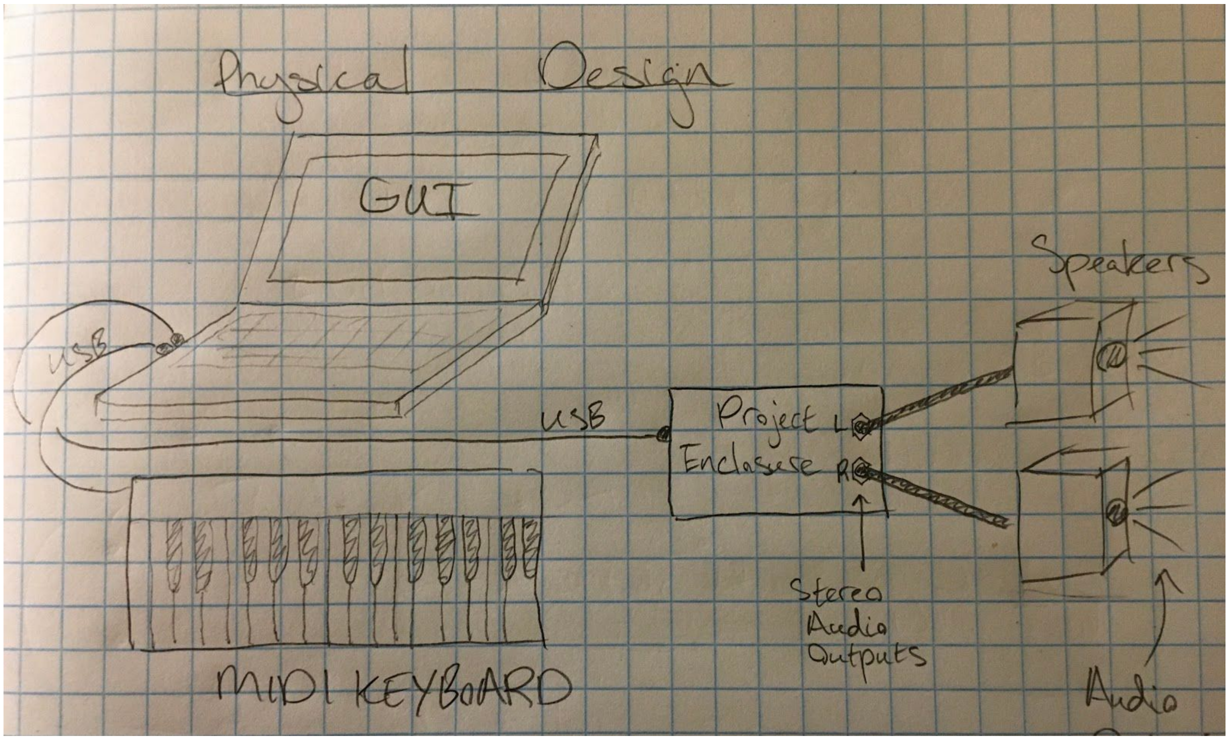
We propose a cheap solution that combines the best qualities of analog synthesizers, digital plugins and modular synthesizers (either analog or digital). Oscillators, envelopes and modulators are generated digitally with a wide variety of options and configurations; these are then converted to analog signals and the user can, through a routing matrix which allows them to set the source and destination of almost all inputs and outputs in a set of analog voltage-controlled amplifiers (VCAs) and voltage-controlled filters (VCFs), configure the synthesizer in any way they would like, allowing for the creation of extremely complex analog sounds for a relatively low price and with a high degree of precision. We will eliminate much of the cost by doing away with the keyboard and most internal circuitry aside from the combination VCF/VCA chips, as MIDI keyboards are now widely and cheaply available, and MIDI connectivity will allow it to be programmed without the use of a synthesizer as well.

1.2 Background

While the dominant form of audio synthesizers in use by musicians are currently purely digital audio plugins and apps, which possess the obvious advantages of relatively low cost and extremely high portability and reliability, in recent years there has been a renewed interest in the analog synthesizers of the 1970s and 1980s. This is primarily due to the perceived superiority of the auditory qualities of the analog voltage-controllable filters (VCFs) used in these designs, which have been described as “warmer” and “livelier” [1]. For much of the last decade, the sole response to this demand on the part of larger, multinational synthesizer companies like Korg and Roland was through the release (often in collaboration with smaller, more digitally-focused synthesizer plugin companies) of digital plugins and apps which emulated their most popular analog synthesizers: 3 out of 10 of the best software synthesizers on a recent list were made with the direct involvement of a large synthesizer company, and 6 out of 10 of them are based on analog synthesizers [2]. However, in the past two decades, there’s been renewed interest in analog sounds, resulting in a collector’s market for “classic” synthesizers [3] and culminating in the eventual release of analog/digital hybrid synthesizers by both Korg and Roland, two of the largest synthesizer companies in the world: the Korg minilogue in 2016 [4] and the powerful but extremely expensive Jupiter-X by Roland in 2020[5]

However, all of these synthesizers are limited to an extent by their fixed architecture. Most take the output of their oscillators (all of which were designed to use the same waveform or combination of waveforms at any given time), feed these into low-pass filters whose cutoff can be changed over time (for control of the timbre or “brightness” of the sound), and feed the result into amplifiers which can also be changed over time (to alter the dynamics of the sound, i.e. how quickly the sounds begin and the way they decay). Some offer modulation options for any of these levels, but if you want, for example, to feed the output through a string of VCAs which vary the volume at separate speeds before you filter them, or feed the output of an oscillator through a chain of multiple filters for more detailed timbre control, you were out of luck. Modular synthesizers are designed to offer this level of control and were, in fact, the first type of commercial music synthesizer to be made widely available in the late 1960s [6], but analog modular synthesizers are prohibitively expensive (thousands to tens of thousands of dollars) and difficult to use, generally requiring the user to “patch” together different modules with various functions using physical cables. Digital Modular Synthesizer plugins eliminate the cost issue, but again, the sound of the analogue synthesizer is heavily prized.

1.3 Physical Design



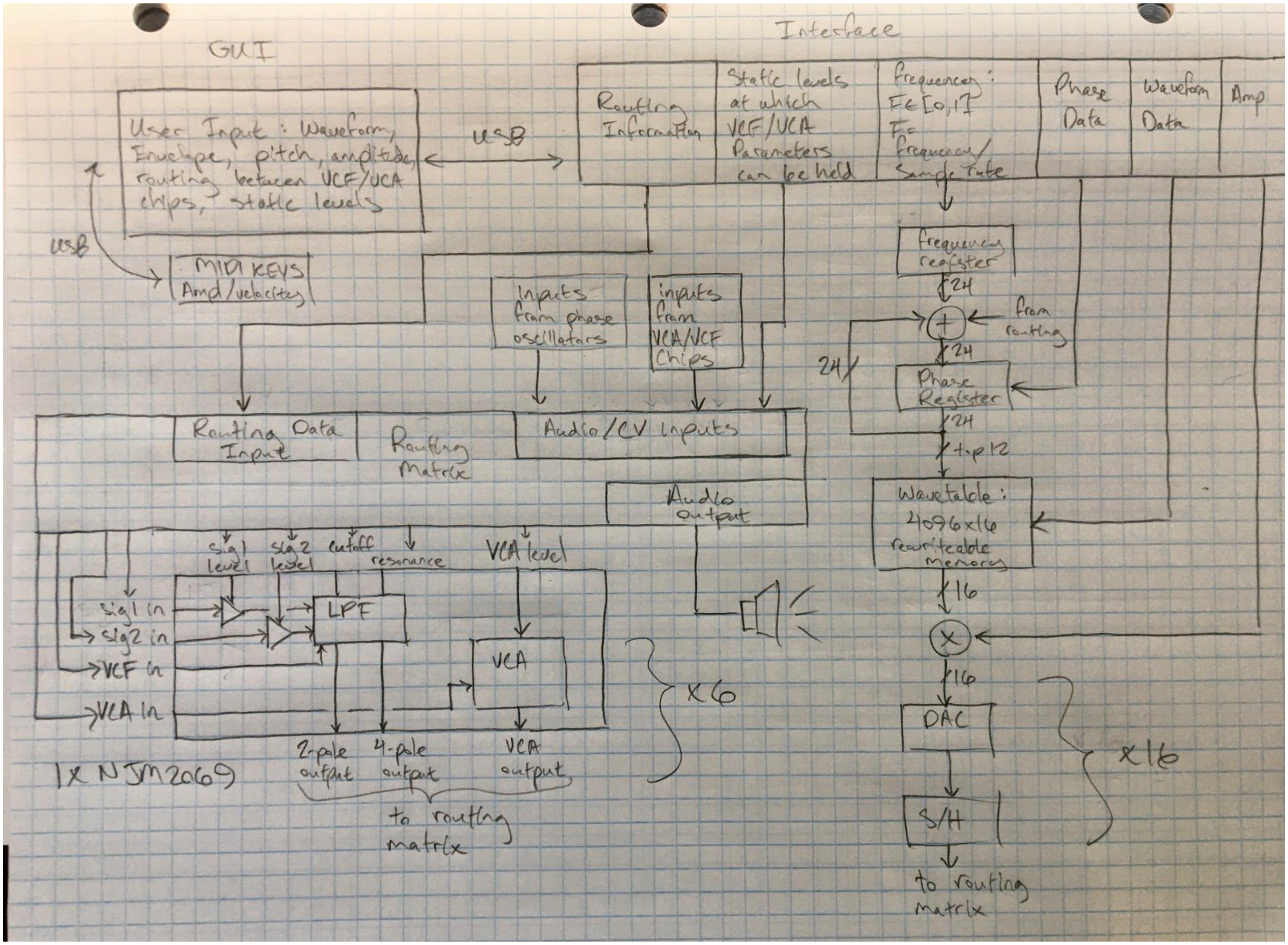
The Physical design will be small and minimal, as the only fixtures required on the box itself will be a USB port and stereo audio outputs. A black metal enclosure with just those items and perhaps an insignia is all that will be necessary.

1.4 High Level Requirements

- A graphical user interface (GUI) which allows the user to select between a variety of waveforms as well as to graphically draw envelopes. This should also be capable of allowing the user to set static, non-time-varying control voltages for various pins on the VCF/VCA chips, as well allowing for the routing of all signals to/from the VCFs and VCAs through the use of a visual grid.
- An underlying program capable of parsing the necessary information from the GUI and communicating this information to the microcontroller in order to produce the desired waveforms or envelopes
- A buffered grid which connects the outputs of the microcontroller to the inputs and outputs of each of the VCF/VCA chips, as well as the inputs and outputs of each of the VCF/VCA chips to one another, to allow for a routing scheme which can be configured through the graphical user interface and which does not damage the chips themselves.
- A set of analog VCF/VCA chips for processing the signals from the microcontroller, as well as the underlying circuitry necessary to string them together.

2 Design

2.1 Overall Block Diagram



The waveform data is sampled 4096 times and sent to flash memory in the STM32F7 microcontroller. Other relevant information like the amplitude, frequency, phase data, routing info, etc are sent to the STM32F7 for processing. Routing information is sent to the routing matrix, while frequency, amplitude and phase information are used to determine the rate at which the samples in flash memory are walked through, thus determining frequency. Routing matrix signals are sent to the VCF and VCA chips as dictated by the routing config, and the end result is sent to audio outputs

2.3 Block Descriptions

2.3.1 Graphical User Interface

This serves as the access point for the user in their interaction with the synthesizer.
Sources will have two possible configurations: Oscillator or Envelope.
1) In the first configuration, a source oscillator can be selected from a pulldown menu of traditional oscillator waveforms (Sinusoid, Triangle, Sawtooth, Pulse, Noise), and the frequency source of the oscillator can be selected:

- MIDI keyboard input
- Fixed/user configurable frequency (if, for instance, a user wanted to use the oscillator output at a sub-audio frequency for modulation/control purposes but wished to change this frequency over time using a MIDI CC message controlled via a knob)
- (ideally and if permitted by time/implementation) frequency input from another/other selectable oscillator(s), allowing for FM synthesis before the signal ever becomes analog.

2) In the 2nd configuration, a window will be available where the user can create an envelope by adding points (using a mouse or touch screen) to create a piecewise linear curve and select a designated time period over which the envelope unfolds. This will have two modes:

1) A one-shot mode, which will be used for applications such as providing ADSR-like envelopes to control the volume/level of a voice or other control.

2) An oscillator mode, where the bottom and top of the window will represent the (positive and negative) peak voltage of the output. In this configuration, the points the user has drawn will be cycled like an oscillator and the selectable time period will be replaced by a selectable frequency. In this way, the user can employ hand-drawn waveforms to drive the DACs.

In both of these configurations the amplitude will also be controllable, and will hopefully be controllable through use of a formula as well, though again this is a detail not necessary for functionality.

There will also be a screen for the rerouting grid where the routing is controlled through a patching matrix. The user can click on a point between an input (rows, listed on the left) and an output (columns, listed on the top) to connect them to one another. The matrix organizational scheme prevents the user from routing inputs to inputs or outputs to outputs.

In addition, a screen which allows the user to set the control voltages for the VCAs to constant, non-time-varying levels will be provided. This is used in the event that the user wishes to pass input through one of the chips without varying some of the levels

2.3.2 Control Program

This is the underlying program which will communicate with the STM32F7 chips to produce the waveforms, envelopes, routing configuration and static control levels selected in the GUI. It will implement a wavetable oscillator, where values are written to temporary storage in flash representing each waveform, and values are read through at a rate determined by the desired frequency.

The layout of this implementation is available in the block diagram. A frequency register sets the slope of the phase which is actually the speed of the table walkthrough, while an accumulator periodically updates this value

As we have not fully decided on a programming implementation, this is subject to some change, and details which may depend on the degree to which we are able to program the STM32F7 are

2.3.3 Microcontroller/Digital-to-Analog Interface

This will pass the waveforms/envelopes/static control signals to the DACs and will also handle the control signals for routing output to/from VCF/VCA chips.

Currently, the idea is that the waveforms the user has selected, drawn, or otherwise engineered will be sampled and sent to temporary storage using flash memory. These samples will be walked through at a rate determined by the desired frequency. The clock frequency needed to achieve this can be calculated by taking the number of samples per waveform (4096 in the case of this example) multiplied by the desired frequency of the output; As the highest note producible on a MIDI keyboard is 12543.854 Hz, this could be achieved with a clock frequency of $(4096 \frac{samples}{cycle}) * 12543.854 \frac{cycles}{second} \approx 51.4\ MHz$ which is easily achievable by most modern standards. In a worst case scenario, the highest pitch which might reasonably need to be synthesized for audio purposes (assuming a human audience) is 20kHz, resulting in a clock frequency of 81.92 MHz, again well within the realm of feasibility. The flash memory and clock will be provided by the microcontroller. Currently the plan is to use the STM32F7 series, although the number of these which will be necessary for our purposes remains to be determined.

This output will be sent to DACs, rendering them analog waveforms capable of interfacing with the VCF/VCA chips. 16 DACs can be implemented using two PCM1690 chips by TI for a total cost of $4.14, although a single STM32F7 chip contains two DACs and these may be used depending on the number of these chips deemed necessary for the project.

Static control signals will likely be implemented as 6-8 bit binary values which will be multiplied by a small voltage and sent to the corresponding point in the control grid. For instance, if we wanted to set the cutoff frequency of one VCF to half of its full range and were using a 6-bit control value, we would set the grid to route a constant value to the pin in question, and set that value in the GUI to half (this would likely be visually represented as some kind of slider or knob). This value would be sent through our microcontroller to a circuit which multiplies it by a reference voltage equal to 1/64th of the voltage corresponding to the maximum cutoff of the VCF ($V_{VCF, Max Cutoff}$ = 7.65 volts on a +12V/-5V supply for the chips selected, making $V_{VCF, Ref}$ = 0.1195V, approximately).

2.3.4 Routing Matrix

This will (in all likelihood) consist of a network of muxes controlled by our microprocessor(s). Buffers will be used at the inputs and outputs of each of the VCF/VCA chips to avoid crosstalk between the various signals. This should also allow for a single audio output or control output from the microcontroller to be routed to multiple inputs on the VCF/VCA chips for advantages such as using a single envelope to control all available VCAs for a polysynth-type configuration, etc).

2.3.5 Analog Filter/Amplifier Chips

We'll be using obsolete synthesizer chips here as the technology which was available for these in 1985 simply isn't made anymore. This is sufficient for a working prototype, while mass production would require other considerations. There are several "synth-on-a-chip"-style "music voicing systems" to choose from, which include utilities such as a VCF and several VCAs on a single

IC. Among these, the standouts are the CEM3389, the SSM2045, and the NJM2069. The latter happens to be the chip we will use for this project, both because the filter and main VCA are independently accessible on the same chip and because I happen to possess 10 of them, with an additional 8 available if we need them enough to ruin a rather nice synthesizer. Each NJM2069 Boasts:

- 4 signal inputs: 2 inputs to the VCF whose input levels can be controlled by another voltage source (one VCA per voice is included on the chip to facilitate this) as well as a direct input to the VCF and a direct input to the main (separate/independent) VCA
- 3 outputs: one producing a 2-pole VCF and one for a 4-pole VCF (could be reduced to one output and made switchable) as well as a VCA output
- 5 CV inputs which allow for control of the parameters of the VCAs and VCF:
- VCF Resonance, VCF Cutoff, Signal 1 Level, Signal 2 Level and VCA Level (technically there are two more, but these will be used in the temperature stabilization circuits) [3]

We estimate that we will use between 4 and 8 of these chips in the final product; this will depend on the feasibility of the routing system as well as other considerations (they're likely to be the most expensive part of the circuit, and we have a limited number of them given that the budget probably won't allow for the purchase of many replacements). Each of these will require temperature stabilization through the use of 2 thermistors per chip.

2.4 Risk Analysis

The safety of the NJM2069 chips is integral to the success of the project, and as these chips are verging on 40 years old, precautions need to be taken to ensure no voltages exceeding the proper voltage rating for each input appears there. As the processes taking place at the inputs will be largely unknown (as they will be configured by the user), a system needs to be put in place to properly regulate this.

As these particular chips were devised by Korg and no datasheet is publicly available, we are relying on secondhand information from an individual who produced several hundred voltage-controllable filter modules for use in modular synthesizers with regards to the maximum voltages, as well as schematics from the synthesizers in which the chips were used in their official capacity. While the chips can be run, according to synthesizer schematics for the Korg Poly-800, DW-8000 and DSS-1 synthesizers, at a variety of rail voltages from +/- 5V to +/- 12V (with a +12V/-5V configuration appearing in the schematic for the DSS-1[7]), the aforementioned individual claims that at +/- 12V a maximum of +/-5V can be presented to the control voltage inputs without destroying the chip [8]

Hence, we intend to run the signal outputs for the microcontroller on a +/- 5V supply. While efforts will be presented to the user to keep all waveforms which might enter the control inputs between 0 and 1 (corresponding to 0V and a Vmax slightly below 5V yet to be determined), some processes we've proposed might lead to larger output magnitudes, which will be clipped by the microcontroller.

3 Safety and Ethics

3.1 Safety

Safety considerations for this project are essentially the basic precautions while working with the given hardware/setup. The highest voltages we will be working with are going to be +/-12V (for the power rails). The PCB for the project will be enclosed in an aluminum case, thus preventing the user from accessing even this minimally dangerous voltage.

As one team member will be using lab equipment in ECEB over the course of the design and testing process (signal generators and oscilloscopes), all precautions against COVID-19 will be taken by this member, including the use of disposable gloves, a mask with a filter, hand sanitizer, etc.

3.2 Ethics

We could think of very few ethical considerations for the project. As the chips were once manufactured for Korg on a proprietary basis, there may be some ethical considerations to make there. However, when a company representative was emailed by one team member to ask for a datasheet for the NJM2069, the response was that there was no record of any such datasheet existing. Hence, it's probably safe to assume that the patent has expired. In any event, as this is only a proof of concept and, as mentioned in the corresponding section, other measures would have to be taken in the event of mass production given that NJM2069 chips are no longer produced, we think the point is somewhat moot.

References

[1] Sam Beattle, '*What Makes Synthesizers so Expensive?*', Andertons Music Co., Dec. 12, 2019. Accessed on: Sept. 12, 2020 [Online]. Available: https://blog.andertons.co.uk/learn/what-makes-synths-so-expensive

[2] James Nugent, '*The 10 Best Synth VST Plugins 2020*', Consordini, Apr. 14, 2020. Accessed On: Sept. 15, 2020. [Online]. Available: https://consordini.com/best-synth-vst-plugins/

[3] Sound on Sound, '*The Analogue Revival*' , Mar., 2014. Accessed on: Sept. 16, 2020 [Online]. Available: https://www.soundonsound.com/reviews/analogue-revival

[4] Scott Wilson , 'Korg Minilogue Affordable Analogue Polysynth Announced', FactMag, Jan. 14, 2016. Accessed on: Sept 16, 2020. [Online]. Available: www.factmag.com/2016/01/14/korg-minilogue-polyphonic-analog-synth-announced/

[5] Rebecca Eaddy, '*Roland Debuts All-New Jupyter-X Synthesizer Lineup*', Roland Press Memo, Sept. 5, 2019. Accessed on: Sept 16, 2020. [Online]. Available: https://www.roland.com/us/company/press_releases/2019/ROLAND-DEBUTS-ALL-NEW-JUPITER-X-SYNTHESIZER-LINEUP/

[6] Jeff Harder, '*Early History of Electronic Synthesizers*', howstuffworks. Accessed on: Sept. 16, 2020. [Online]. Available: https://electronics.howstuffworks.com/gadgets/audio-music/synthesizer3.htm

[7] '*Korg DSS-1 Service Manual*', manualslib. Accessed on: Sept. 16, 2020. [Online]. Available: https://www.manualslib.com/manual/1245904/Korg-Dss-1.html

[8] user 'antichambre', '*Midibox 2069*', midibox. Accessed on: Sept. 16, 2020. [Online]. Available: http://www.midibox.org/dokuwiki/doku.php?id=mb_2069