Reroutable Hybrid Analog/Digital Audio Synthesizer

By

Team 32
Adam Wills
Connor Jones
Ishaan Datta

October 2020
University of Illinois at Urbana-Champaign
Senior Design

Contents

1. Introduction 1
   1.1. Problem and Solution Overview 1
   1.2. Solution in Context 2
   1.3. High-Level Requirements 2

2. Design 3
   2.1. Overall Block Diagram 3
   2.2. Power Supply 4
   2.3. Software Interface 4
   2.4. Microcontroller/Control Program 7
      2.4.1. Microcontroller 7
      2.4.2. Foundational Control Program 10
   2.5. Digital-to-Analog Interface 12
      2.5.1. Post-DAC Amplifiers 12
   2.6. Routing Matrix 13
   2.7. Analog Filter/Amplifier Section 15
   2.8. Tolerance Analysis 20
   2.9. COVID-19 Contingency Plan 23

3. Cost 24
   3.1. Labor 24
   3.2. Parts 24
   3.3. Total Cost 25
   3.4. Schedule 26

4. Safety and Ethics 27
   4.1. Safety 27
   4.2. Ethics 27

Citations 29

Appendix A: Wavetable Synthesis 31
Appendix B: Thermal Compensation and Related 32
Appendix C: Synthesizer Measurements 34

1 Introduction

1.1 - Problem and Solution Overview

Our senior design project involves the implementation of a hybrid analog/digital synthesizer, primarily for use by musicians and audio researchers. Digital synthesizers use digital signal processing techniques to program sounds; they use discrete signals, signifying that a signal's value does not change continuously[14]. Conversely, analog synthesizers utilize purely electronic oscillators passing through filters and amplifiers implemented with analog integrated circuits to create a continuous signal with infinite resolution[14].

The audio market has an ever-growing demand for customizable analog synthesizers; yet, with the rise in inexpensive and accurate digital technology, digital synthesizers have attained prominence due to their easy upkeep and low costs. This has led to the development of a plethora of digital synthesizers that are available in DAWs such as Logic Pro X, FL Studio, and Pro Tools.

However, in the past 20 years the demand for analog synthesizers has steadily grown yet again, largely due to consumers, musicians especially, finding the sound quality of analog devices to surpass those of their digital counterparts[1]. Of these analog devices, the modular synthesizers are most sought after, due to their control over sound manipulation. However, they aren't always digitally-compatible, are often quite expensive, and require the use of patch cables that can easily become cluttered, as below.

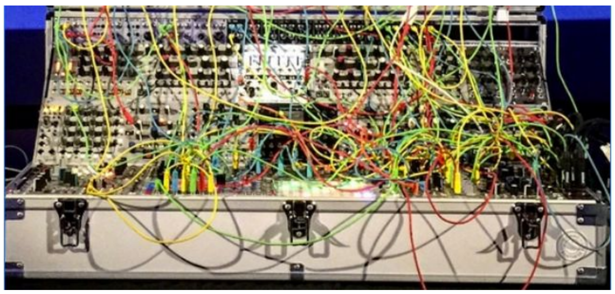


Image 1. Analog Modular Synthesizer

Our solution combines the easily adjustable digitally-generated sound with the sound manipulation components -- voltage-controlled filters and amplifiers -- of analog synthesizers, and furthermore allows both the digital inputs and analog outputs to be used anywhere in the synthesizer for any purpose. The degree of flexibility is unparalleled for the intended price-point.

The design will incorporate a software GUI to craft audio waves that are routed through an input/output matrix between units, eliminating cable-based patching. Ultimately, we will implement a hybrid modular synthesizer in the range of $300-$600, a cost-effective and highly flexible device that allows for a plethora of options within sound design and synthesis.

1.2 - Solution In Context

This synthesizer utilizes only a USB port, stereo audio outputs, and potentiometers to control the amplifier and filter levels manually. This feature is quite useful for realtime, tactile control and will allow the user to ensure parameters will never be zero when not connected to the grid. A black metal enclosure will contain these items and include a product identifier or insignia for cosmetic purposes. With greater financing, the enclosure could also contain routable external inputs that allow the user to process signals from hardware they own. An LCD screen displaying the state of the patching grid or some other useful visual data could benefit the user and would make use of the full capabilities of our microprocessor. However, the majority of the user interface will be completed in MaxMSP, as it is an easily manipulable interface that can relay the necessary signal messages between the user and the analog output.

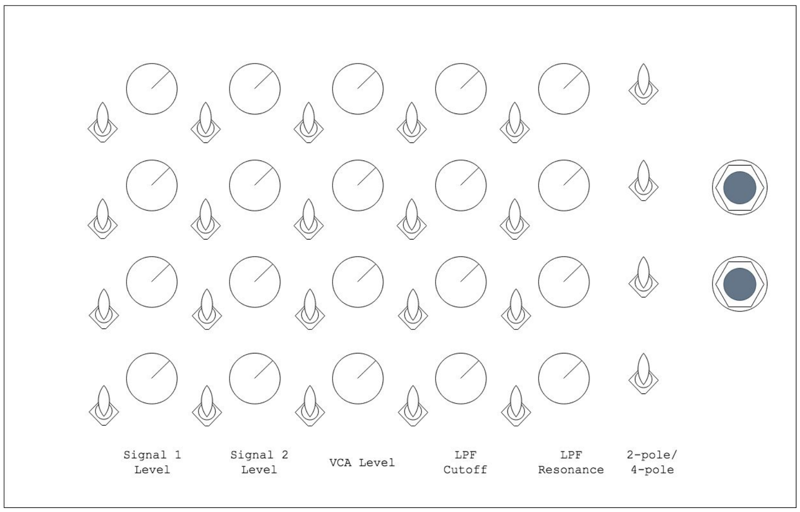


Image 2. Design Physical Layout

1.3 - High-Level Requirements

- A graphical user interface (GUI) where the consumer controls the synthesizer's audio outputs. It allows them to select, modify and control the sounds and signals which will be routed through the analog filters. Must be easy to navigate and understand.
- Programming to store user inputs and convert into wave_forms before being passed into the routing matrix. Also needed, to ensure that user information is not reset unless desired.
- A routing scheme capable of connecting any signal output, from either the DACs or the filter chips themselves, to any input (either control or signal), allowing the user to craft a huge variety of sounds from relatively few components

2 Design

2.1.1 Overall Block Diagram

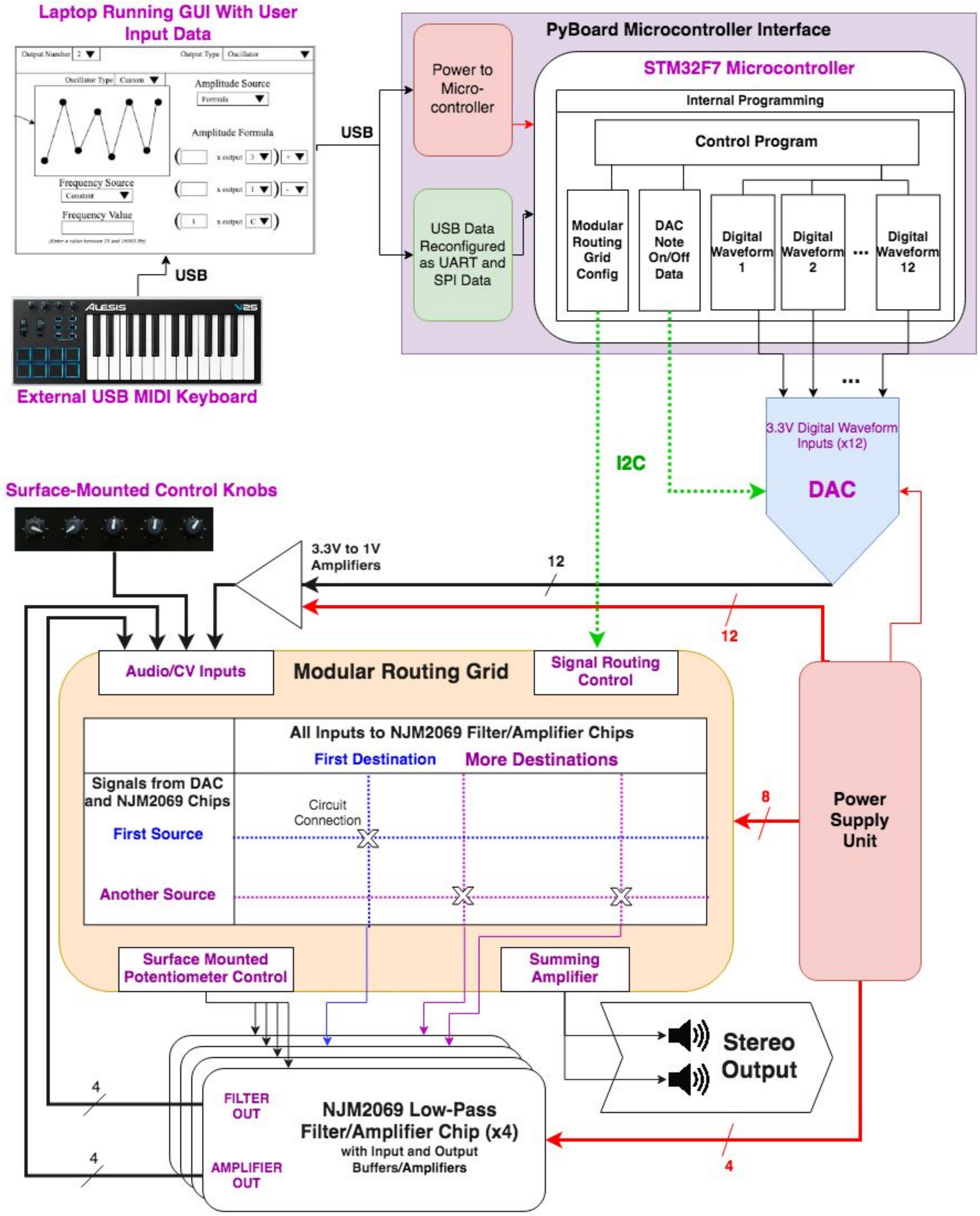


Image 3. Overall Block Diagram

2.2 Power Supply

Inputs: 36VAC 1.5A Transformer connected to 120V AC mains through 3-prong grounded power cable

Outputs: +/- 12V supply, +/- 5V supply, +/-3.3V supply (total 900mA)

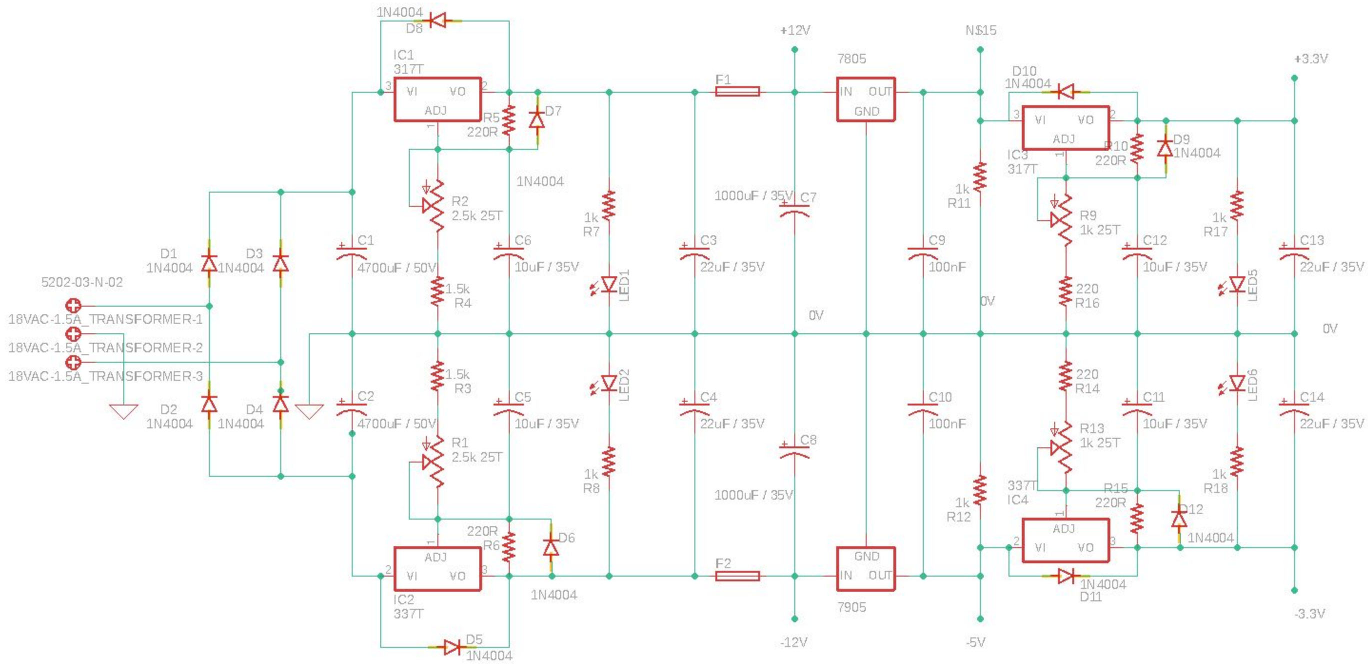


Image 4. Power Supply Diagram

The power supply consists of two adjustable LM317 positive voltage regulators, two LM337 adjustable negative voltage regulators, an LM7805 5V positive voltage regulator, and an LM7905 5V negative voltage regulator connected through a single ground bus (star ground). The CGS66 power supply by Ken Stone was referenced for the +/-12V section[11].

2.3 Software Interface

Input: User input data for each waveform generato: Output number, Output on/off, Output Type, Frequency Source, Frequency Value, Amplitude Source, Amplitude Value

Output: 30-byte message communicated serially through the USB virtual comm port (USB_VCP) representing the status of each user input selection:

INTERFACE

This serves as the access point for user interaction with the synthesizer. Twelve digital waveforms will be independently accessible through the use of generators that define the 4096 samples that make up each waveform. Each of these wave forms will then be passed into the routing grid to undergo analog processing.

The routing grid is crucial to the synthesizer's function as it allows any signal generated by the microcontroller or analog section output to connect to any analog section input. It also allows the oscillators to be used as control signals, a monumental advantage providing more unique sound design options to the consumer. Utilizing oscillators effectively can create widespread effects from slow back-and-forth modulation to frequency/amplitude modulation.

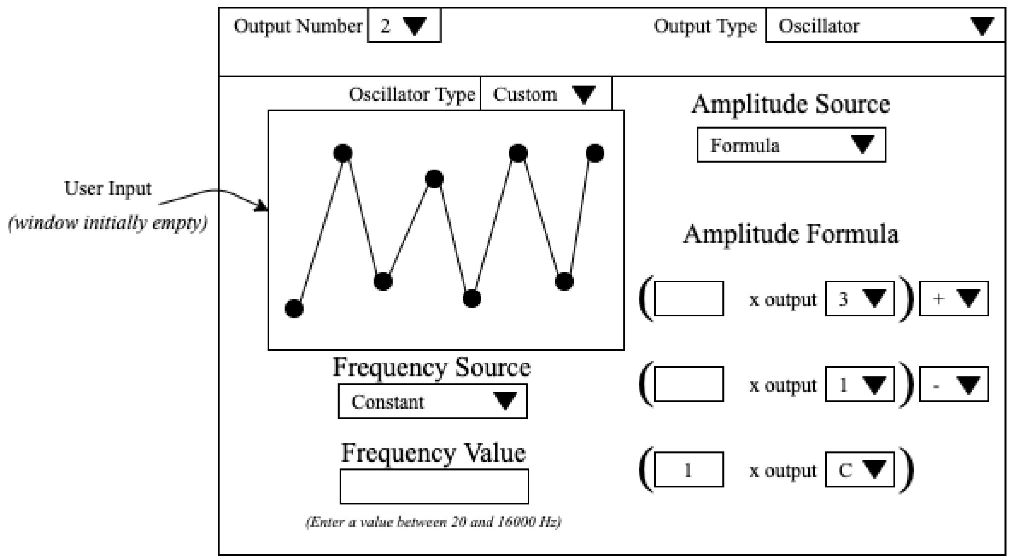


Image 5. Example Display: Oscillator Window

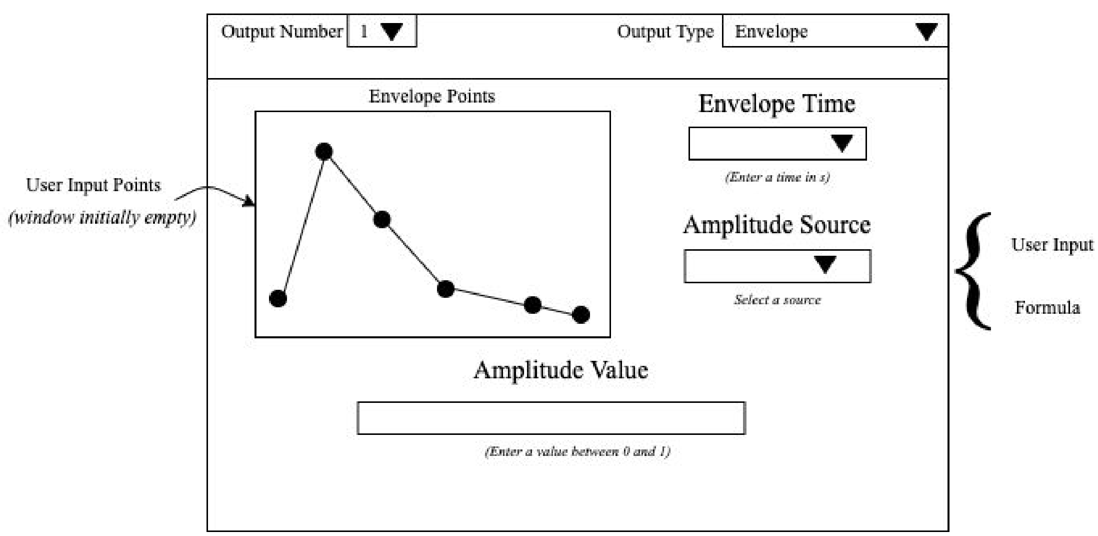


Image 6. Example Display: Envelope Window

Options	Oscillator Types	Frequency Sources	Amplitude Sources	Operations (*Formula Mode)*
	Sine	Constant	Constant	+
	Sawtooth	Keyboard - MIDI NN	Keyboard - Velocity	-
	Square	Formula	Formula	*
	Triangle			/
	Noise			

Table 1. Available Oscillator Configurations and Sources

Our synthesizer interface allows the individual manipulation of both oscillators and envelopes.

The system oscillators can be set to represent a variety of waveforms as shown in *Table 1*. Our interface also allows the user to set the waveform's frequency and amplitude. The earliest implementation of our product will incorporate constant and formulaic aspects of setting frequency and amplitude. With time, we will add in MIDI implementations as well, to allow the user to incorporate MIDI keyboards that they may already own, as they are common within the industry.

IMPLEMENTATION

We intend to implement the digital section using an STM32F767 32-bit microprocessor embedded on a D-series PyBoard. This board allows for the incorporation of microprocessor code written in microPython, a pared-down subset of the python language designed for embedded applications. This will allow us to design the computer-based user interface in Python, and generally allows for simplified real-time access to the parameters of the STM32 using the virtual comm port class, pyb.USB_VCP, which would otherwise have to be varied using the Serial Ports or USART.

Requirement	Verification
Display menus, interfaces and user input fields appear on the screen, without error and can be updated appropriately.	1. Input values into each user input field. 2. Set generators using these input values 3. Print out the generator values to ensure the values are being properly parsed by the code and stored in the designated buffer.
All menus must be togglable and feature each option allocated by the design.	1. Click on each option in the drop-down menus on the interface. 2. Ensure all features are listed within the drop down function. 3. For drop-down functions that alter the screen, select these functions and ensure

(cont'd) All menus must be togglable and feature each option allocated by the design.	that the screen makes the necessary alterations. 4. For each drop down function, set the generator. Include random sample values for other parameters. 5. Print contents of the updated generator to ensure each feature selection is linked to the proper variable in the code that gets stored.
Transmits user parameters to the microcontroller program. For all time-sensitive inputs (MIDI control of oscillator and envelope amplitude, oscillator frequency), this should occur in under 10ms. For all other inputs that don't poll user data points, this should occur in under 50ms. The applications that poll user data points (the 'custom' oscillator option and the amplitude envelope) should load those values to flash in under 2s and should be able to read that data in under 10ms after loading.	1. Configure a debug version of the user interface in python which, using the 'time' library, saves the time immediately after user input is toggled. 2. Enable a corresponding debug version of the portion of the microcontroller code which receives user input through USB. This should save the time immediately after the user input is received. 3. Systematically test each user input option 5 times. Subtract the time at which the input was toggled from the time at which it was received for each run, and average the results.

2.4 Microcontroller/Control Program

Input: 30-byte serial message indicating the status of all user selections in the user interface, MIDI note and velocity data generated by user and transmitted through GUI, Power connection through PyBoard USB interface

Output: enabled GPIO pins implementing (+/-)0-3.3V discretized versions of the desired waveforms, I2C control signals for DAC and Routing Grid sections

2.4.1 Microcontroller

The microcontroller communicates 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 read at a rate determined by the desired frequency.

The STM32F767 also provides multiple I2C controls (pins PB10 and PB11 in image 5, pins PB8 and PB9 in image 6), which will be used to interface our microcontroller with both the signal routing matrix and the digital-to-analog interface through the use of specialized ICs. Digital outputs will be programmed through the use of the GPIO pins on the STM32F7, of which there are 40. Overall, there are 80 available outputs, separated into two busses of 40.

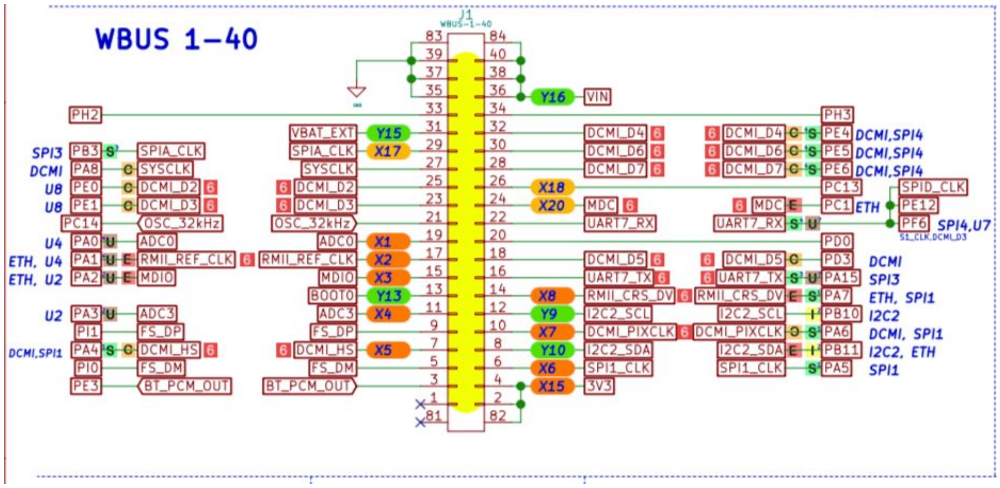


Image 7. PyBoard WBUS 1-40

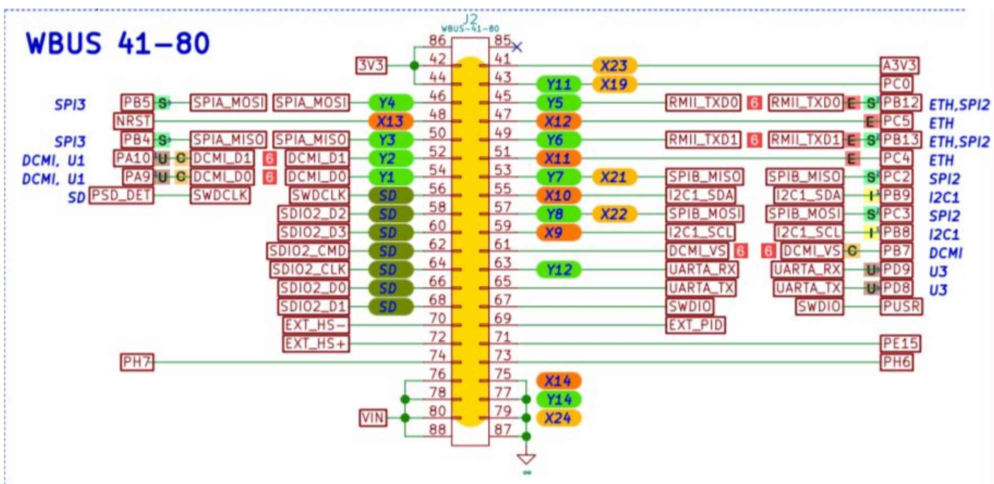


Image 8. PyBoard WBUS 41-80

The GPIO pins not occupied by a peripheral connected to the system or internal programming are available for use as outputs. As we will be using neither the internal analog-to-digital converters nor the internal digital camera media interface, all unmarked pins and pins marked as ADC or DCMI may be used for output.

Output is configurable using microPython with commands such as

```
pin = machine.pin(pin_number, machine.Pin.OUT)
pin.value(desired_value)
```

Requirement	Verification
Transmits received MIDI note and velocity data through to underlying control program in under 10ms	1. Trigger MIDI note on keyboard 2. Use python's time library to track processing delay 3. Run test 10-20 times 4. Ensure percentage of successful tests > 95%
Produces no aliasing for any oscillator waveform and all notes up to C8 - highest note on a grand piano - with frequency 4186.01 Hz	1. Display waveform with frequency >= 4186.01 on oscilloscope. 2. Rotate the horizontal scale knob, or otherwise change the horizontal scaling of the oscilloscope. If the waveform changes drastically, aliasing is present
Executing micropython code for a selected waveform and frequency results in a 24-bit discretized version of that waveform appearing at the output (+/- 5% accuracy in magnitude) for all notes up to 4186.01 Hz	1. Obtain, through a program which directly computes the desired waveform and samples it with 24-bit accuracy, a set of reference values (2-3 period's worth) indicating where output voltages should ideally fall within the 0-3.3V range. Do this for the frequency represented by A0 (F=27.5Hz), the lowest MIDI note available on traditional keyboards 2. Using the 32-bit onboard ADC of the STM32F7Nucleo, sample the output of the pyBoard as it produces the same waveform. 3. Compare the two solutions by subtracting the value of each corresponding sample from one another and taking the square of the absolute magnitude. Then sum these and take the square root to get an average estimate of the error (2-Norm) 4. Divide this value by the 2-Norm of the reference samples alone to get an estimate of the average (decimal) error. Ensure this is less than 5% (averaged over 2-3 trials) 5. Repeat for all notes up to C8 and all waveforms aside from the noise generator. This process can (and should) be automated.

2.4.2 Foundational Control Program

Wavetable Synthesis

The algorithm we will be using to produce the waveforms is referred to as wavetable synthesis. Wavetable synthesis is a means of implementing digital waveforms developed both to prevent aliasing and reduce computation time by precomputing single cycles of the waveform and storing them as arrays. These arrays are then walked through at a rate determined by the frequency using a phase oscillator, pictured and described below. A brief description of wavetable synthesis is provided in *Appendix 1*.

The STM32F7 has 2MB of onboard flash memory, and we have six waveforms we wish to implement -- sine, triangle, square, sawtooth, noise and ‘custom’. ‘Custom’ will load user-defined waves into wavetables, which may cause a delay in processing relative to the aforementioned wave forms.

Using single-precision float values (4 bytes per float) for our samples, 16 wavetables consisting of 4096 samples each will take up 262.144KB per waveform, for a total of approximately 1.57MB. This corresponds to a different wavetable every half octave. If using this amount of flash memory to store the wavetables affects performance to an extent deemed unacceptable, individual wavetables can contain smaller numbers of samples, or a smaller number of wavetables can be used.

Phase Oscillators

The mechanism by which the wavetable index is computed for a given frequency and time is known as a phase oscillator.

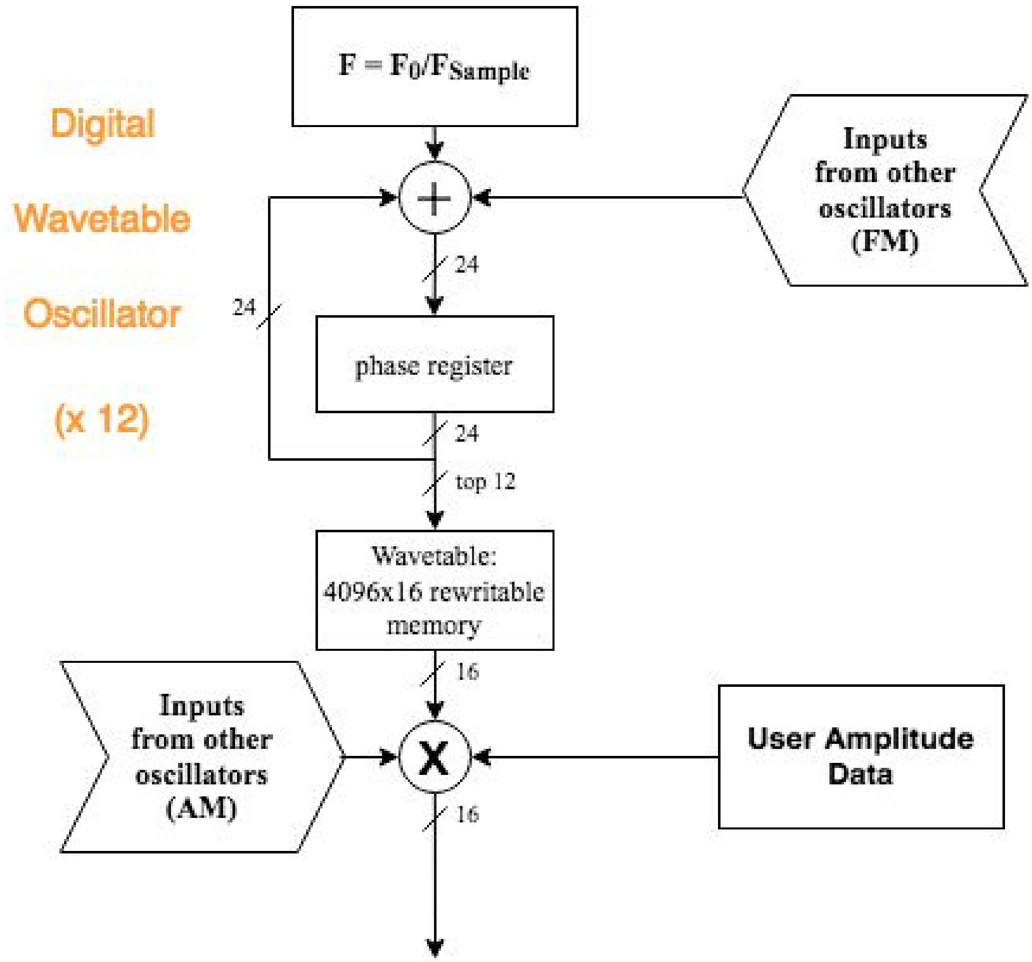


Image 9: Phase Oscillator Block Diagram

The value of F_0/F_{Sample} , where F_0 is the fundamental frequency of the waveform/note currently being used is added to the value in the phase register every clock cycle. The phase register has 24 bits for accuracy (ours may have less) but selects the top 12 bits and uses them to address one of the 4096-bit wavetables. Using the output of another oscillator as an input to the addition into the phase register implements frequency modulation, by increasing the read speed of the wavetable by the value of its output at the frequency of its oscillation, and using another oscillator's output as an input to the final amplitude multiplier implements amplitude modulation.

Requirement	Verification
The wavetable synthesis program must not alter the single-pass waveform stored in the buffer. While not altering what's stored in the buffer, it also must use the buffer and properly process the waveform at the desired frequency to produce the correct audio signal.	1. Store signal of single amplitude in the wave_form buffer. 2. Pass through the wavetable synthesis algorithm at a single frequency. 3. Ensure that the output signal is a single-tone audio output of a single-level. If fluctuations in pitch or frequency are detected, there is an error. 4. Repeat this test for; a. linearly increasing, then decreasing amplitude with single frequency; b. Linearly increasing, then decreasing frequency with single amplitude; c. And linearly increasing, then decreasing frequency and amplitude
The waveform must be able to use the output of one oscillator as an input into another.	1. Pass a simple signal through an oscillator such as a sine wave. 2. Take the output and pass it through an additional oscillator, perhaps a square wave. 3. Ensure that the two signals stack, can be measured against digital simulations of the same pairing. 4. Test should be repeated through the custom oscillator choice as well, as the majority of users will likely create a custom option themselves so as to directly manipulate the waveforms.

2.5 Digital-to-Analog Interface

Inputs: (+/-) 0-3.3 V digital inputs from GPIO pins of PyBoard, +5V & -5V Supply voltages

Outputs: (+/-)0-1 V analog inputs to analog filter/amplifier section

This interface passes the all digital signals -- waveforms, envelopes, static control -- to the DACs and handles the control signals for routing output to/from VCF/VCA chips. We will be using the TI LMP92001 Multichannel DAC, which provides 12 independent 12-bit DAC channels, and is capable of interfacing with our microcontroller through I2C.

2.5.1 Post-DAC Amplifiers

In order to ensure that outputs from the DACs won't endanger our analog VCF/VCA chips, we need to attenuate their outputs down to the 750mV peak-to-peak (max) standardized level being used for signal inputs there (see section 2.7 for more detail). This will be implemented through the use of TL072 operational amplifiers. As 3.3V is too close to the rail voltage for a +/- 5V supply to be used, the +/-12V supply will be needed for these amplifiers.

Requirement	Verification
3.3Vp-p (+/- 0.3V error) digital waveforms at the input to the DAC during operation.	1. Using an oscilloscope, ensure a digital version of the intended waveform is present at the input pin and conforms to the desired standard
1.4Vp-p output (+/- 5% error) continuous Analog Output signal at DAC output pins	1. Set to ready/inactive (through the GUI or underlying program if the GUI is not yet debugged) a simple output waveform to the GPIO connected to DAC1 2. Connect the I2C_SCL (PB10) and I2C_SDA (PB11) outputs of the pyBoard to the corresponding I2C inputs on the LMP92001 DAC 3. Ensure, through LMP92001 address bits 31:32, and by setting the I2C configurations in the underlying program to mirror the address of 31:32, that you are communicating with the proper DAC 4. Measure output at the corresponding output pin using a digital oscilloscope and ensure it conforms to the given requirement 5. Repeat for all DACs
Must properly convert any input digital signals to audio signals.	1. Pass a single-tone audio signal into the DAC. 2. Test the audio it produces digitally 3. Connect the output of the DAC to an analog speaker. 4. The audio signal produced by the analog speaker must match that digitally produced during the aforementioned test. 5. Repeat with varied audio signals.

2.6 Routing Matrix

Input: (+/-) 0-1V inputs from Analog-to-Digital Interface, (+/-) 0-1V inputs from Routing Matrix, +5V, +12V, +5V and -5V supply voltages from power supply

This matrix will contain several analog crosspoint switching IC's, coupled with I/O buffers specific to the given application. We will be using the ADG2128 8 x 12 switching matrix to connect each of the filter chip outputs to each of the other's inputs, all of which will be controllable through our microcontroller. This set-up allows us to control the routing of the synthesizer through the graphical interface.

For each NJM2069 filter chip, described in section 2.6, eight inputs will be used; four as signal inputs accepting an alternating waveform, and another four as control inputs which accept a variety of ranges depending on the application. The routing matrix carries an advantage in its ability to use signal inputs as control inputs. Through this quality and the use of precision full-wave rectifiers, the signal is ensured to remain above 0 volts, allowing for various modulation and synthesis options.

To route the twelve outputs of the digital-to-analog section to each of the eight inputs of the NJM2069 chips, four ADG2128 8x12 switching matrices will be used, while connections from the NJM2069 chips to other NJM2069 chips will be routed through a separate four ADG2128 matrices. All will be controlled through the use of the I2C2_SCL and I2C2_SCA pins on our PyBoard.

Requirement	Verification
Verify that ADG2128 Chips Pass Signal Through All Connection Points	Test chips individually. 1. Manually verify that test DC voltages of +/-5V are passed through each of the 96 ports when operated individually on a +/-12V power supply. 2. Manually verify that a +/- 5V sine wave passes through all ports when operated on a +/- 12V power supply.
Verify that signals on separate ADG2128 chips carrying the same buffered output to different buffered inputs have no interaction with one another, or that signal interaction occurs at a low enough level as to be negligible for the listener/user (difference in peak-to-peak voltage of less than 2% from unconnected waveform)	1. View signals on an oscilloscope individually, without the other connected. Measure peak-to-peak voltages. 2. View signals on the oscilloscope while both signals are connected. Ensure minimal or no change
Verify that signals are not degraded excessively (THD >= 5%) in routing process	1. Pass a single-amplitude, single-tone sine wave into the system 2. Measure the total energy at the output of the system at the harmonics of the input frequency, each must be measured separately[15]. 3. Using these energy readings, calculate the total harmonic distortion and ensure that the signal does not exceed 5% THD. 4. Repeat 10 times with different amplitude sine waves and ensure > 80% success rate. a. Likely to be some complications with greater amplitude sine waves as THD levels often increase proportionally with amplitude.

2.7 Analog Filter/Amplifier Section

Input: (+/-) 0-1V analog inputs from the routing matrix section, +/-5V power supply connections to NJM2069, +/-12V power supply connections to other circuitry

Outputs: (+/-) 0-1V buffered VCF output and VCA output from the analog filters

Buffered Analog Filter/Amplifier Block Diagram

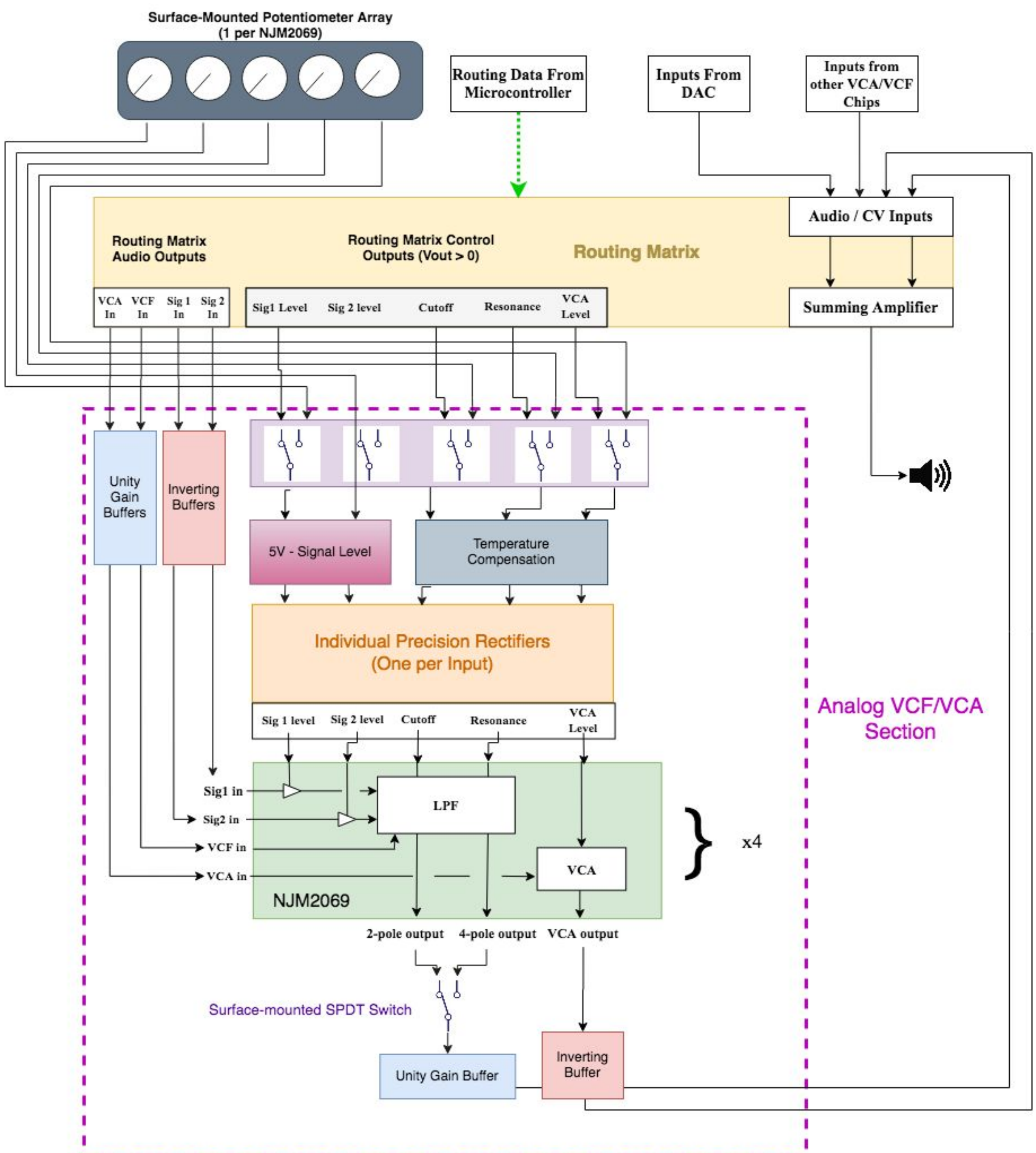


Image 10: Analog Filter/Amplifier Sub-block Diagram

The NJM2069 possesses signal inputs that are processed through analog filters and amplifiers. The chips also possess voltage-control inputs that control parameters such as VCA level and filter cutoff. Some of these inputs are inverted, for instance, -368mV represents the minimum level and 0V represents the maximum level[5].

This section integrates the filter chips with buffers and amplifiers for each input and output. The buffers and amplifiers will be implemented using TL072 op-amps. The TL072 has been selected due to its high slew rate. This characteristic of the op-amp allows it to easily sustain high audio frequencies, produce low THD of around 0.003% in most applications, and draw a low current around 1.4mA per op-amp. These characteristics lend themselves well to our project, as there are a large number of op-amps in use for buffering and amplification into and out of the NJM2069's.

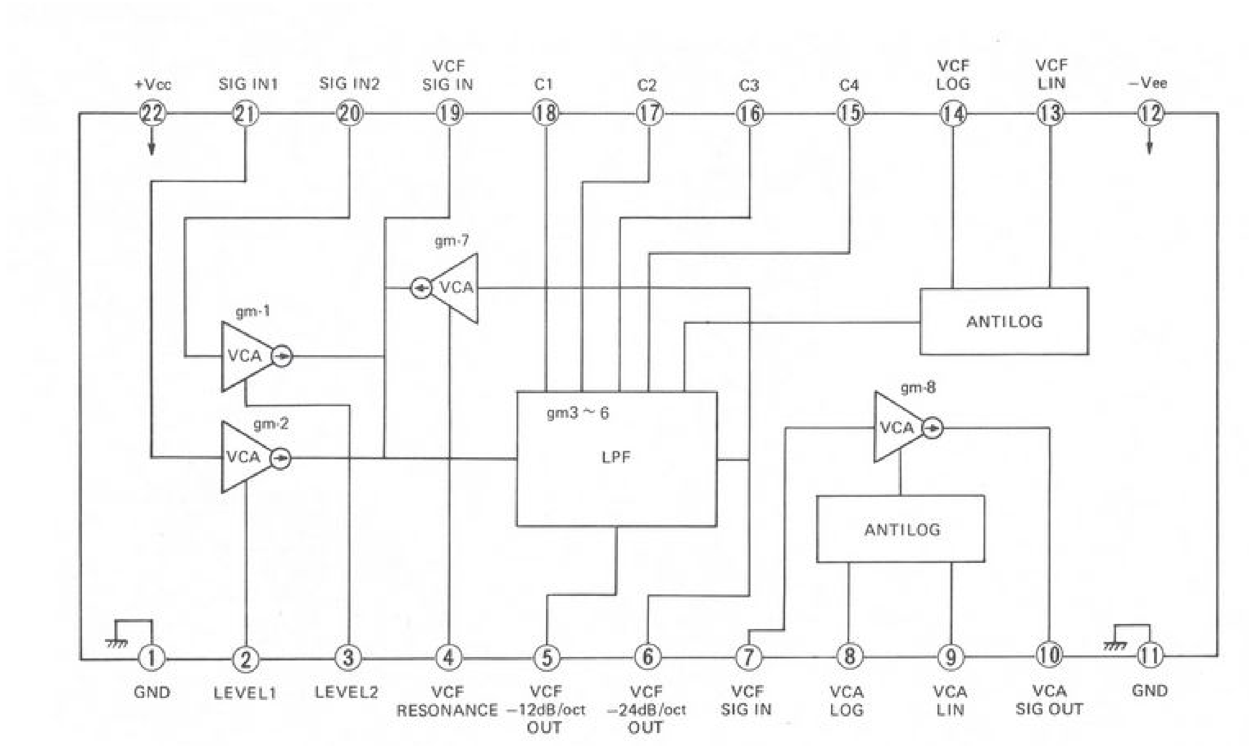


Image 11. NJM2069 Pinout and Interior Block Diagram[5]

Four NJM2069 chips will be utilized to implement the analog filter/amplifier section. A schematic illustrating their use in context is shown below, as it appears in the Korg DW-8000 service manual[4]. Due to the lack of documentation for these chips, we will be configuring our circuits from ones already known to be functional; therefore, we will use this IC roughly as implemented in the DW-8000 with the addition of a variety of buffers, amplifiers, and rectifiers on the inputs and outputs.

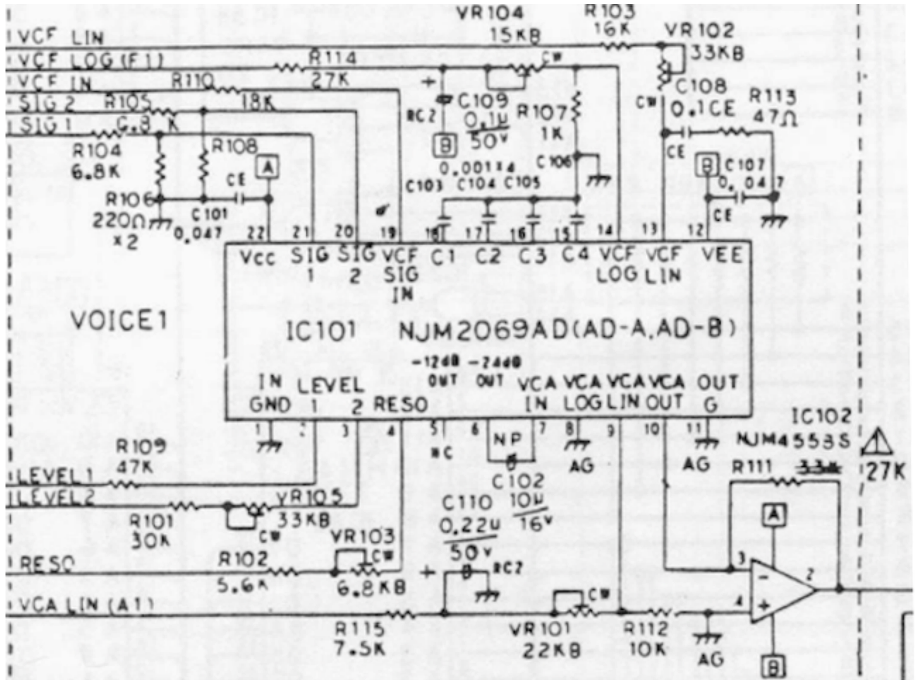


Image 12. Filter Schematic Using the NJM2069 taken from DW-8000 Manual[4]

As the NJM2069 circuit shown above is essentially a "black box", measurements were taken to determine the appropriate levels of signal/control inputs and signal outputs. More information from these measurements can be found in *Appendix 3*. Measurements were taken at the resistors, capacitors, or pins where signals would be input/output if using the circuit as shown above.

Op-amps at the inputs of each of the controls and signals will serve to amplify or attenuate the inputs from a homogeneous/circuit-appropriate level to the level appropriate for the given input, so all signals can be routed in and out of the device. This will also be done at the outputs.

To better correspond with control voltages, output signals will be converted from typical levels of ~500mV peak-to-peak VCF out, denoted as '-12dB Out' and '-24dB Out' in the table) and ~225mV peak-to-peak (VCA Out) to approximately 500mV peak-to-peak. Each I/O pin, corresponding buffer configuration and active NJM2069 circuit measurements are summarized in the table below.

Signal Inputs	Pin	Inverting?	Vpp_{max}	$Vpp_{typical}$	Buffer/Amplifier Type
Signal 1 In	21	Yes	572mV	~225mV	Inverting Amplifier with A_V = 0.5
Signal 2 In	20	Yes	572mV	~225mV	Inverting Amplifier with A_V = 0.5
VCF In	19	No	572mV	~225mV	Non-inverting Buffer
VCA In	7	No	912mV	~500mV	Unity Gain Buffer
VCA Out	10	Yes	384mV	~225mV	Inverting Amplifier with A_V = 2
-24dB Out	6	No	768mV	~500mV	Unity Gain Buffer
-12dB Out	5	No	912mV	~500mV	Unity Gain Buffer
Control Inputs	**Pin**	**Inverting?**	**Off-Voltage**	**On-Voltage**	**Buffer/Amplifier Type**
Sig 1 Level	2	No	0V	336mV	Precision Rectifier, Gain = 0.66
VCF Log	14	Yes	380mV	-364mV	Inverting amplifier, Gain = 1.33
VCF Resonance	4	No	0	316mV	Precision Rectifier, Gain = 0.66
VCA Lin	9	Yes	-368mV	0V	Precision Rectifier Buffer into Inverting amplifier, Gain = 0.66

Table 3. Summary of NJM2069 I/O Properties and the Buffers or Amplifiers to be used

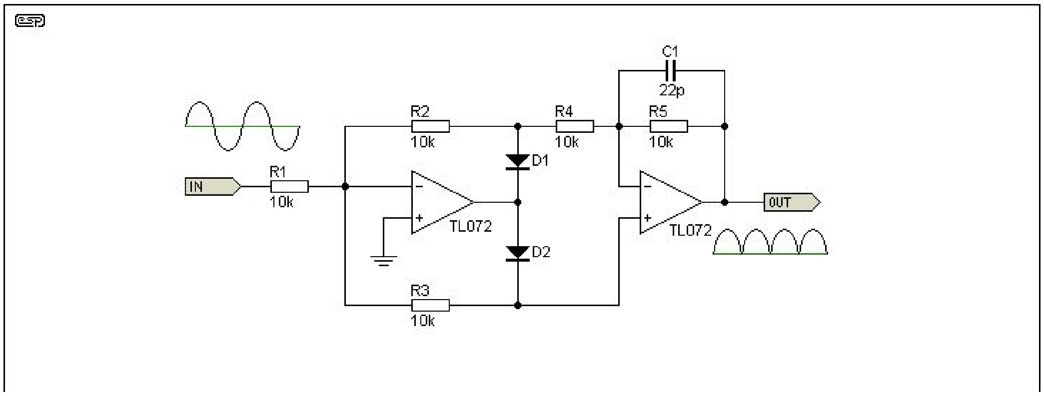


Image 13. Precision Full Wave Rectifier Capable of Gain[12].

As stated in the caption, the above circuit will be utilized for full wave rectification as it has a relatively low input impedance. However, we will replace all 10k resistors with resistors of larger impedance --, e.g., 100kOhm - 1mOhm resistors -- to achieve a buffering effect.

To obtain the total current draw, we need to tabulate the number of op-amps used in the above buffering configurations.

- Inverting amplifiers can be realized with a single op-amp, and can be made buffering with appropriate resistor choices. There are 3 of these, with another used to invert the rectified signal into VCA Lin, for a total of 4 op-amps
- Non-inverting amplifiers with $A_V < 1$ must be realized by stringing together an inverting amplifier with the desired gain and an inverting buffer with a gain of 1. There are 2 of these, or another 4 op-amps.
- The precision rectifier buffer can be constructed from two op-amps; 3 of these results in additional 6 op-amps.
- Non-inverting amplifiers with $A_V > 1$ can be constructed with a single op-amp; There are 2 of these.

This makes for a total of 16 op-amps per NJM2069 section, or 64 op-amps in total. The current draw of a TL072 is quite low at ~1.4mA per op-amp[13], leading to a total current draw of 89.6mA.

Requirement	Verification
Output levels standardized to 750mVp-p (+/- 10%)	1. Test outputs of VCF (-24dB and -12dB) and VCA with various combinations of inputs 2. If any exceed the designated values, adjust the level of amplification.
All inputs receive voltages corresponding to their typical values (as seen in table above) and with proper filtering if necessary.	1. Build circuit based on schematics. 2. Utilize an oscilloscope to test measurements at each pin. 3. The voltages received at each input should be within +/- 66% of their typical value.

2.8 Tolerance Analysis

Our project focuses on enhancing products currently available in the industry and making them more accessible to wider audiences. Due to the nature of the device, it's imperative that all filters function appropriately and that every signal is properly routed through the matrix.

The NJM2069 chips are a critical component to this section as they handle analog filtering. The chips are nearly 40 years old and require specific precautions to ensure no voltage exceeds the proper voltage rating for its associated input. The chip itself has an absolute rating voltage of +/- 12 V (dual-rail), over voltage is not supported for chip powering and will result in the chip being fried[5]. However, the audio inputs do support over voltage, and will be clipped above +/- 5 V[5].

As there has been no official documentation for these chips published by their original manufacturer, Korg, we are relying on secondhand information from an individual who has produced similar voltage-controllable filter modules and from schematics depicting synthesizers using chips at 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 +/- 5 V to +/- 12 V (with a +12 V/-5 V configuration appearing in the schematic for the DSS-[7]), the aforementioned individual claims that at +/- 12 V a maximum of +/-5 V can be presented to the control voltage inputs without destroying the chip[5]. Hence, we intend to amplify the signal outputs of the DAC to an absolute maximum of +/- 4.7V from their original +/- 3.3V, to allow for some degree of variation.

The TL072 operational amplifier carries a low typical harmonic distortion percentage, 0.003%, so slight changes in the operating mechanics of this component won't heavily affect our project's ability to function. In addition, this component has a wide range for temperature functionality, -65°C to 150°C, so proximity to other components, device usage time and location of operation should not have much of an effect on its use[6]. Most musicians prefer to have equipment that's portable and able to work under a variety of conditions, which makes this component particularly compatible for this product. Although its optimal operating voltage is between 5 and 15 V, the component has a maximum voltage range from -0.3 V to 36 V[6]. With the previous component in mind, we'll likely keep the system voltage in the 5 to 15 V range to prevent other chips from frying, which will also protect the TL072.

The ADG2128 matrix is central to ensuring that the signals are properly routed through the device. We must ensure that we monitor this device, as exposure to absolute maximum rating conditions for extended periods of time could affect device reliability[7]. One area of concern lies in the voltage ratings for the analog and digital inputs. For the analog inputs, there's a wide voltage operating range, -7.3 V to 15.3 V that shouldn't create much issue with the project as voltage monitoring for other devices should minimize operating time at maximum rating conditions, ensuring device reliability. For the digital inputs, however, there is a much smaller allowable input range, dependent on V_L from -0.3 V to 7.3 V (Max V_L = +7 V).

Optimal V_L occurs at 5V, so we can avoid having any device malfunction by powering the matrix with the same monitored power source as that for the NJM2069 dual rails. There is also a cap on the digital input's circuit flow at 30 mA, as opposed to its optimal flow rate near 0.4 mA. As the input requires very little circuit flow and there is a relatively large amount of gain needed to reach maximum circuit flow, it's unlikely that this will be an issue throughout our project design. We will however, test input circuit flow using an oscilloscope prior to connecting to the matrix to ensure that the device will not be damaged.

The TI LMP92001 Multichannel DAC is crucial to this section as well, in its ability to connect with the microcontroller embedded on our PyBoard. Specifically, it provides the analog interface between the microcontroller and a given analog system[8]. Similar to the ADG2128 matrix, this chip has a relatively small powering input range, from -0.3 V to 6.0 V. These similarities present the solution in that we can power this chip using the same power source as the ADG2128 matrix and the aforementioned NJM2069 chips. However, there are stated specifications that should be noted about current flow. Each pin should have at most 5 mA of current pass through at any point.Similar to the routing matrix, current flows will be measured via an oscilloscope before passing them through the device to prevent damage or reduced operability. There is also a maximum current setting for those that pass through VDD or GND. This amount varies by about 40 mA depending on the temperature -- 78 mA at 125°C to 120 mA at 105°C[8]. The device contains an on-board analog temperature sensor that monitors the device's internal temperature[8]. The output will be readback when we first start activating the device to record where our chip's temperature tends to fall during operation so we could monitor the powering current appropriately.

Arguably, the most important component to the function of our product concerns the STM32F767 32-bit microprocessor that we have embedded on the PyBoard. The V_{DD} USB powers the USB port that we're using to interconnect our devices. It's imperative that how the device is powered is constantly checked, because there are consequences for improper usage that could damage the board. The port can either be directly linked with the V_{DD} port on the board, which ensures that both signals will rise and fall at the same time. However, when the port is connected to an external power supply, the V_{DD} USB supply must be "the last supply to be provided and the first to disappear"[9].

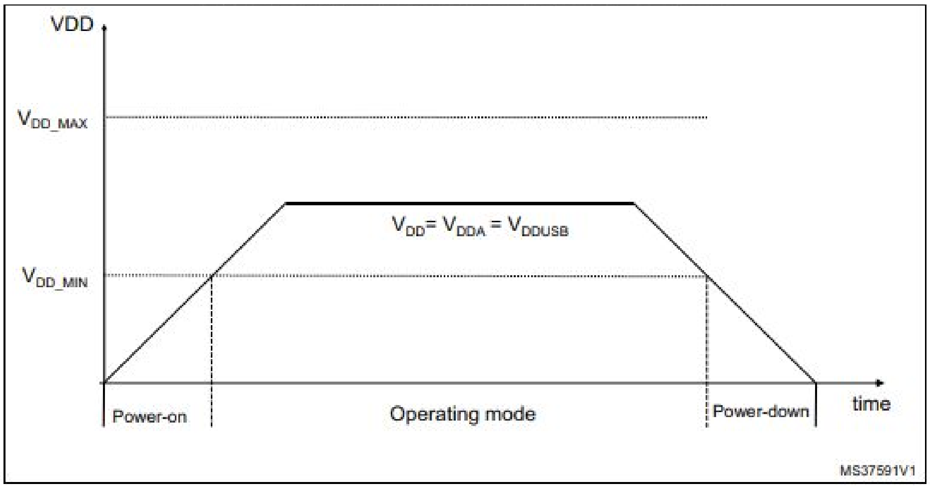


Image 14. $V_{DD,}$ USB connected to V_{DD} Power Supply[9]

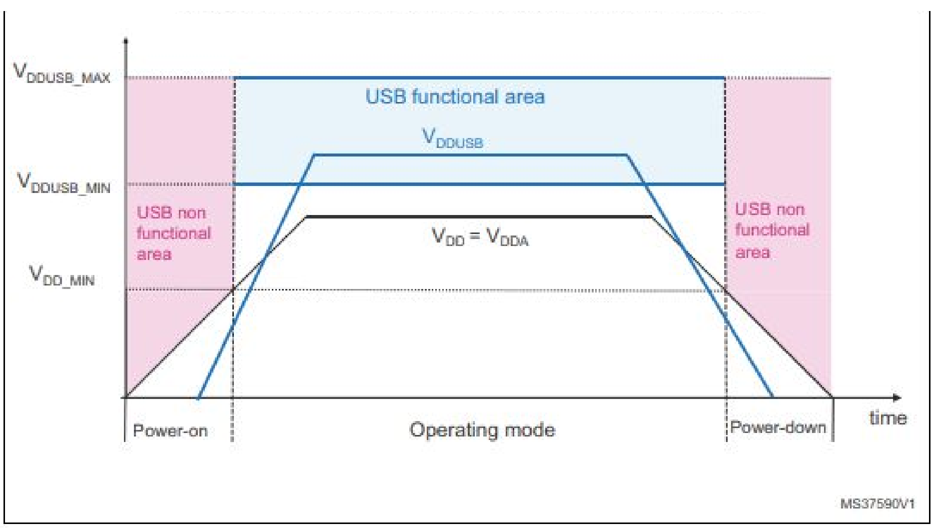


Image 15. $V_{DD,}$ USB connected to External Power Supply[9]

Throughout the analysis, we've found that the majority of the components have problems arising with voltage input and output. In a project that depends on signal flow, this makes ample sense and by keeping these factors in mind throughout development we will be able to avoid damaging our equipment. Temperature related factors shouldn't cause much of an issue throughout the majority of our components, but should be kept in mind during the initial development factors when we first find how much voltage and current flow is being passed from our signal into our components.

The analysis has also guided our plans, specifically with regard to how we will be rigging our power supply. By discovering the differences in USB capabilities from powering the USB through an external power supply or in conjunction with the main V_{DD}, we've decided to use the latter rig. By connecting the two ports, we can ensure that there is no lapse in USB performance.

2.9 COVID-19 Contingency Plan

Currently, our project has only required one member to be present in the lab. In the event that the school building is closed due to changes in COVID-19 regulations, all hardware designs and constructions would be completed off campus. Our COVID-19 contingency apparatus consists of an EZ Digital FG-7002C 2MHz Sweep Function Generator, a SainSmart DSO Note II Portable Digital Oscilloscope and, for the remainder of the semester, an Analog Devices M1K ADALM1000 board with accompanying ALICE software, capable of a variety of measurement and testing procedures. At least one instance of all essential mixed-signal SMD chips (several ADG2128 switchpoint matrices and the ADAU1962A DAC, as well as the pyBoard itself) have been adapted to DIP breakout boards in the event that further testing (under, admittedly, less than ideal conditions) is needed and access to the lab is further restricted or cut altogether. A power supply with separate digital and analog +/-5V rails, as well as analog +/-12V rails, is available in the form of the DW-8000 power supply, which I've made switchable and outfitted with terminal binding posts for use in testing other projects.

3 Cost and Schedule

3.1 Labor

Based on average ECE salary

- BS Computer Engineering Average Salary: $84.25k
- BS Electrical Engineering Average Salary: $67k

30 hours per week over 6 weeks = 180 hours

Per Partner:
($30/hour) x 2.5 x 180 hours = $13,500

3.2 Parts

- NJM2069 analog filter/amplifier chips
  - Description: Composed of a 24/12db lowpass filter, a two input voltage controlled mixer, and voltage controlled amplifier.
  - Manufacturer: Korg
  - Quantity: 4
  - Cost: ~$89.95/4
- TL072 JFET-input operational amplifiers
  - Description: A high speed amplifier incorporating well matched, high voltage JFET and bipolar transistors in a monolithic integrated circuit. The device features high slew rates, low input bias and offset current, and low offset voltage temperature coefficient.
  - Manufacturer: Texas Instruments
  - Quantity: 32
  - Cost: $0.70/DIP
- D-Series Pyboard
  - Description: A compact, powerful and low power microcontroller module that runs Micropython. It's embedded with a STM32F767 device for audio integration
  - Manufacturer: Python
  - Quantity: 1
  - Cost: $105.23/Board
- ADG2128 8 x 12 Unbuffered Analog Switch Array
  - Description: An I2C Compatible analog crosspoint switch arranged as an 8 x 12 array
  - Manufacturer: Analog Devices
  - Quantity: 8
  - Cost: $13.78/IC
- TI LMP92001 Multichannel DAC
  - Description: A complete analog monitoring and control circuit which includes a sixteen channel 12-bit Analog to Digital Converter (ADC), twelve 12-bit Digital to

Analog Converters (DACs), an internal reference, an internal temp sensor, an 8-bit GPIO port, and an I2C-compatible interface
    - Manufacturer: Texas Instruments
    - Quantity: 1
    - Cost: ~$15 each
- Miscellaneous
    - Assorted resistors, capacitors, ICs, crystals, sockets (Digikey; est.)
    - Manufacturer: Texas Instruments
    - Quantity: Varied
    - Cost: ~$15

3.3 Total Cost

	Name	Indiv. Cost ($)	Number	Total Cost ($)
Labor	Design Team	13,500	3	40,500
Part	NJM2069 analog filter/amplifier chips	22.49	4	89.95
	TL072 JFET-input operational amplifiers	0.70	32	22.4
	D-Series Pyboard	105.23	1	105.23
	ADG2128 8 x 12 Unbuffered Analog Switch Array	13.78	8	110.24
	TI LMP92001 Multichannel DAC	15	1	15
	Miscellaneous	15	~	15
Projected Project Cost				**$40,857.42**

3.4 Schedule

Week	Adam	Connor	Ishaan
10/5	Buy parts	Buy parts	Buy parts
10/12	Construct Enclosure	Develop waveform functions	Research GUI design
10/19	Construct hardware circuits Test A/D Connection	Complete Python Coding	Implement GUI
10/26	Finalize hardware components	Fine-tune/Debug Code	Test GUI Interface
11/2	Test Full System		
11/9	Refine Prototype		
11/16	Prepare for Demo		

4 Safety & Ethics

4.1 Safety

Safety considerations for this project essentially follow the basic precautions for working with any given ECE-related hardware. 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.

Potential safety hazards may arise during soldering due to the nature of the equipment in use, as would be in the creation of our metal enclosure. Through the school we have been trained on how to properly solder, and the enclosure we'll be designing may require the use of some heavier machinery, but our prior experience in the lab will advise us on how to complete what's necessary without sustaining injury. If something were to go wrong, there are a series of procedures to be followed in the lab that will ensure that we are properly treated as soon as possible.

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, and hand sanitizer at minimum. It should also be noted that we are required to be COVID-19 tested by the school; ensuring our safety, as well as the safety of our other classmates.

The other members will be following COVID-19 precautions as well, but will not be interacting with other members of the student body in the ECEB for this project as the aforementioned team member. The work is divided between hardware and software with two of our team members being predominantly software-oriented and off campus. However, in the case that we do meet to discuss initial project ideas and to specify our design, those attending will be sure to get tested as per the school's COVID-19 policy prior to meeting.

4.2 Ethics

We could think of very few ethical considerations for the project, largely due to the nature of the product. The synthesizer is a specialized tool for audio creation, a design that doesn't carry immediate implications towards an individual's privacy or security.

It should be noted that in the event of mass production, NJM2069 chips are no longer produced, which could cause potential issues. In light of this, it must be noted that not many other chips are on the market, aside from the recently re-released Roland 80017A VCF / VCA JUNO-106 Voice Chip Filter IC[16]. These chips were originally released in 1984 and were eventually discontinued. However, recent drives in demand for analog products, led Roland to release clones of the original 80017A chips and are now widely available. However, consumers who have used synthesizers with this chip sooner or later experience "dead voices," or loss of sounds, due to the fact that the 80017A chips were "poorly manufactured and [aged] badly for internal heat with resin." Newer versions of this chip are likely to be of better quality than their original iteration in 1984, however, the unreliability of the chips isn't something we would want in our future. It's also worth considering that the NJM2069A may be put back into production as the 80017A due to the rise in demand for these powerful audio chips.

This does bring up some potential ethical issues with regard to the environment. When companies decide to start producing any component or device, it requires factory space, machines and either human or artificial labor. In a world where factories contribute heavily to the changing of the climate, the repercussions of producing such a component must be considered. Even though producing the component won't take up as much factory space as more widely in-demand products,like plastic water bottles, the production still contributes to the emissions that get released into our atmosphere.

Ethical considerations related to COVID-19 (apart from the above listed precautions) include making sure that we were getting regularly tested, particularly before in-person meetings with teammates or accessing the lab. The nature of our work division lended itself towards following COVID-19 procedures as we did not have to meet in person to work on the project. However, in the case that one of us must go into the lab to use the oscilloscope or to solder parts of the project, we are required to be COVID-19 tested by the school; ensuring our safety, as well as the safety of our other classmates.

5 Citations

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

[2] Dave Hylands, Moving Data to Host Using USB, MicroPython.org, Feb. 28, 2017. Accessed on: Sept. 28, 2020. [Online Forum Comment]. Available: https://forum.micropython.org/viewtopic.php?t=3092#p18251

[3] Vesa Valimaki, '*Alias-Suppressed Oscillators Based on Differentiated Polynomial Waveforms',* IEEE, Jun. 30, 2009. Accessed on: Sept. 26, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/5153306

[4] '*Korg DW-8000 Service Manual',* manualslib. Accessed on: Sept. 28, 2020. [Online]. Available: https://www.manualslib.com/manual/997976/Korg-Dw-8000.html

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

[6] Texas Instruments, 'TL07xx Low-Noise JFET-Input Operational Amplifiers [Datasheet]", ti. Accessed on: Sept. 30, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/tl072.pdf?HQS=TI-null-null-mousermode-df-pf-null-wwe&ts=1601664240059&ref_url=https%253A%252F%252Fwww.mouser.com%252F

[7] Analog Devices, 'I^2C® CMOS 8 × 12 Unbuffered Analog Switch Array With Dual/Single Supplies [Datasheet]", analog. Accessed on: Sept. 30, 2020. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADG2128.pdf

[8] Texas Instruments, 'LMP92001 Analog System Monitor and Controller [Datasheet]", ti. Accessed on: Sept. 30, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/lmp92001.pdf?ts=1601670263007&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLMP92001

[9] STMicroelectronics, 'STM32F765xx STM32F767xx STM32F768Ax STM32F769xx [Datasheet]", ti.

Accessed on: Sept. 30, 2020. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f765zi.pdf

[10] Phil Burk, "Bandlimited Oscillators Using Wave Table Synthesis," in *Audio Anecdotes: Tools, Tips*

and Techniques for Digital Audio, vol. 2. R Barzel and K. Greenbaum, Eds. Wellesley, MA: A. K. Peters, 2004, ch. 7, pp. 37–52.

[11] Ken Stone, “Power Supply for Music Synthesizers”, CGS Synthesizers. Accessed on: Oct. 2, 2020. [Online]. Available: http://www.synthpanel.com/modules/cgs66_psu.html

[12] Rod Elliot, “Precision Rectifiers”, ESP. Accessed on Oct. 4, 2020. [Online]. Available: https://sound-au.com/appnotes/an001.htm

[13] Douglas Self, “Op-amps in Small-Signal Audio Design”, EETimes, Aug. 24, 2011. Accessed on: Oct. 4, 2020. [Online]. Available: https://www.eetimes.com/op-amps-in-small-signal-audio-design-part-4-selecting-the-righ t -op-amp-jfet-input-types-reviewed/

[14] Sweetwater, “Digital Synthesizer”, Sweetwater, Nov. 16, 2004. Accessed on: Oct. 20, 2020. [Online]. Available:https://www.sweetwater.com/insync/digital-synthesizer/

[15] Dave Berners,, “Total Harmonic Distortion”, Universal Audio, n.d. Accessed on: Oct. 20, 2020. [Online]. Available: https://www.uaudio.com/blog/total-harmonic-distortion/#:~:text=In%20order%20to%20perform%20a,is%20the%20value%20of%20THD

[16] n.a.,, “Roland 80017A”, Polynomial.com, n.d. Accessed on: Oct. 30, 2020. [Online]. Available: https://www.polynominal.com/roland-80017a/

Appendix 1

Wavetable Synthesis

Wavetable synthesis is a means of implementing digital waveforms developed both to prevent aliasing and reduce computation time by precomputing the waveforms in the form of arrays.

The arrays contain anywhere from 64 to 8192 samples (depending on how stringent your space requirements are, among other considerations) of each waveform computed using Fourier series, as the primary "geometric" waveforms used in most analog synthesizers (sine, triangle, square and sawtooth waves) are all periodic and can therefore be decomposed into Fourier series.

The arrays are used to prevent aliasing by ensuring the notes you play never contain harmonics above the Nyquist rate, or $F_S/2$, where F_S is your sampling frequency. Generally, the array corresponding to the highest notes contains just a single sine, as only the fundamental harmonic is below the sampling frequency in that register. Implementations vary with regard to how often you introduce a new wavetable[10].

Appendix 2

Thermal Compensation and Thermistor Measurement Procedure

One of the weaknesses of analog audio-processing equipment is its sensitivity to temperature. Several of the control voltage inputs to the filter and amplifier are sensitive to temperature. In the DW-8000, specific compensation circuits involving thermistors are used to prevent fluctuations in, for example, the precise cutoff frequency of the low pass filter when the control voltage is supplied with a constant input.

While 3 thermistors are used in the original DW-8000 to thermally compensate each of the 8 NJM2069 chips simultaneously, this is because they are all designed to operate at the same frequency and amplitude levels at all times. As we will be allowing the user to change whatever parameters they wish on any chip in our synthesizer, we would require 3 thermistors for each of the 4 NJM chips, a total of 12, for the same level of thermal compensation. However, as the resonance and frequency are designed to be modulated fairly heavily over the course of the use of our synthesizer, we've opted only to include the circuit which provides the reference voltage for the low-pass filter's frequency cutoff.

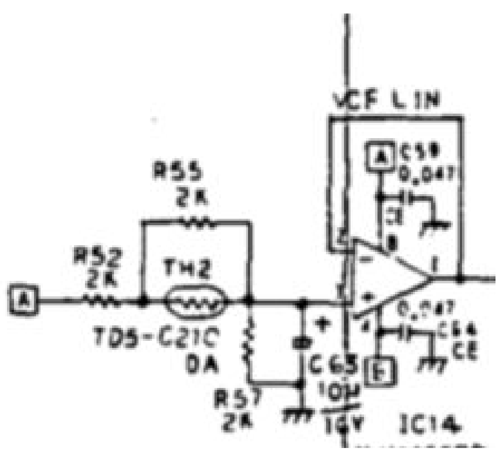


Image 16: Thermal Regulation for VCA control voltage, from DW-8000 Service Manual[4]

(Note: The line appearing through the middle of the figure is not a wire but a page-break)

In keeping with Korg's apparent level of secrecy surrounding the operation of their once-proprietary NJM2069 chips, none of the values for the thermistors were given on the datasheets. As I own a functional DW-8000, a test procedure was implemented to determine a rough approximation of their value.

Thermistor Measurement

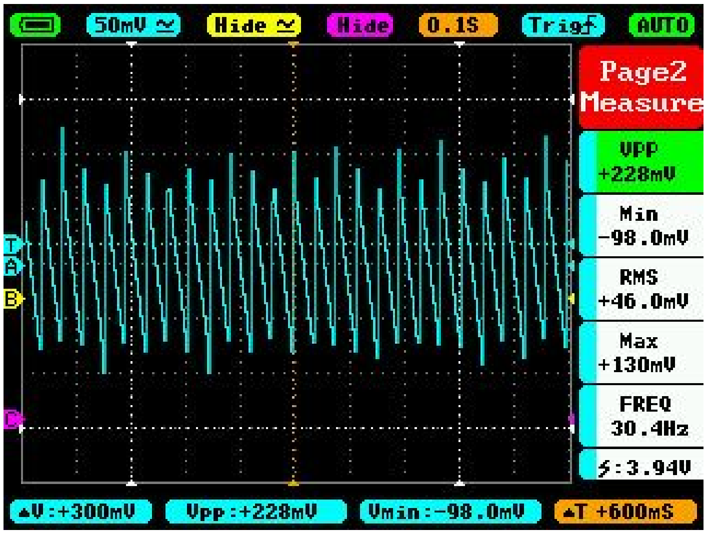
Thermistors are generally characterized by both their resistance at 25 degrees Celsius and their B value, which is a measure of how much resistance fluctuates with temperature. B is determined for industry-standardized temperature values, using 25 degrees celsius or 296.15 Kelvin for T_0 and 50 degrees celsius or 323.15 Kelvin for T in the equation $B = (1/(\frac{1}{T} - \frac{1}{T_0}))ln(\frac{R}{R_0})$. I used 23.05556 Celsius as an approximate T_0 and then froze the thermistors into ice cubes to get their values at 0 Celsius (I don't own an accurate thermometer and have no way of reliably regulating temperature, so these B values are at best approximations, at worst misleading).

First, thermistors were desoldered from the DW-8000 board. Three thermistors are used, labeled TH1, TH2, and TH3. The resistances of the thermistors were tested in the lab, where the temperature readout on the thermostat was 73.5 degrees fahrenheit, corresponding to 23.0555 degrees celsius or 296.20555 Kelvin. TH1 and TH3 were measured as having roughly the same resistance at this temperature; TH1 was measured at 5.162 kOhms, while TH3 was measured at 5.180 kOhms. It was determined that the thermistors were of the NTC (negative temperature coefficient) type, as their resistance fell when I handled them. As 23.0555 Celsius is slightly below the industry reference standard T_0, I assume these are NTC 5kOhm thermistors. A similar procedure was used to identify TH2 as a 1 kOhm NTC thermistor.

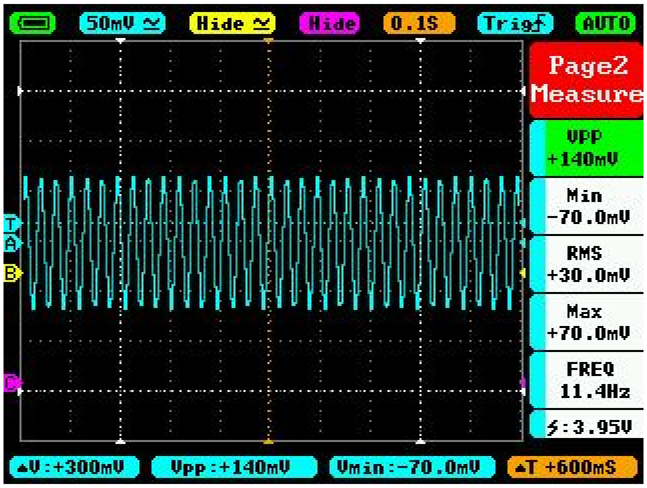
Using their values at 0 Celsius in the above equation, I arrived at B values of approximately 5600 for TH1 and TH3 and 4200 for TH2. Some parts were found precisely fitting these values; the 1kOhm thermistor was close to $22 dollars. Close approximations were found for as little as $0.26.

Appendix 3
Synthesizer Measurements

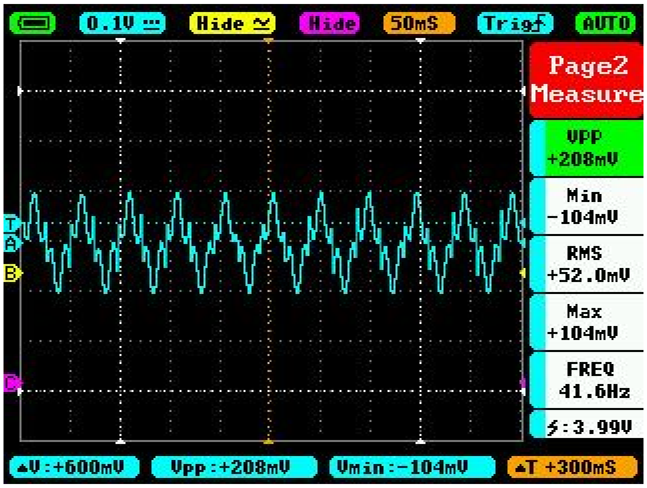
Measurements were taken using the DSO Note II portable digital oscilloscope
Saw Sig2 In:



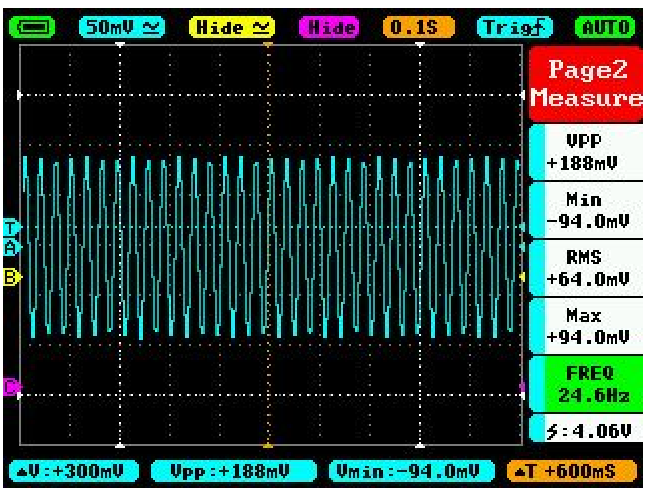
Sin Sig1 In:



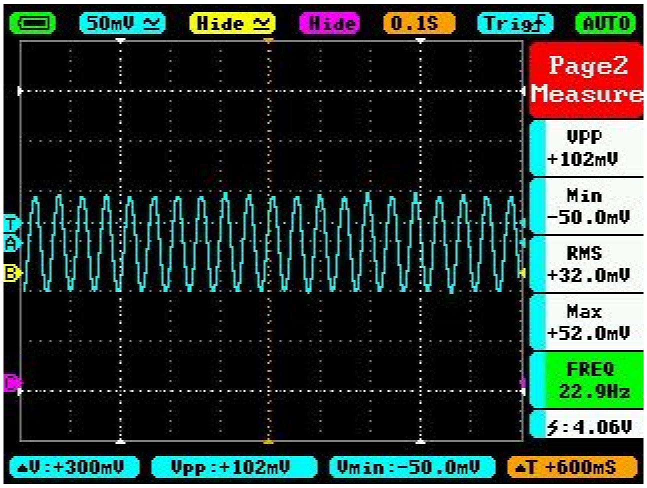
Square-ish Sig1 In:



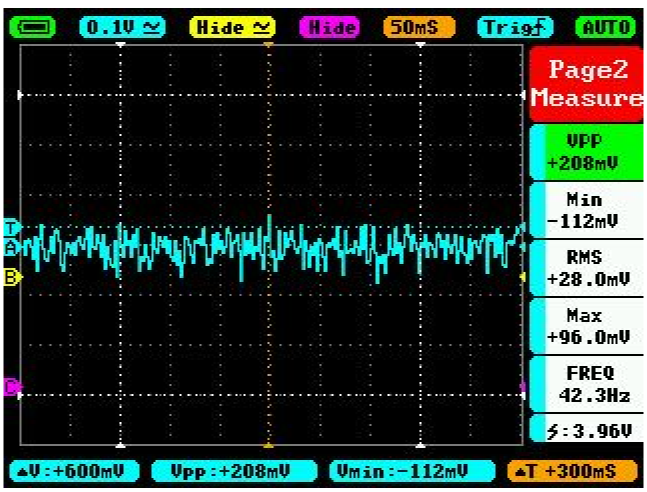
Sine -24dB out:



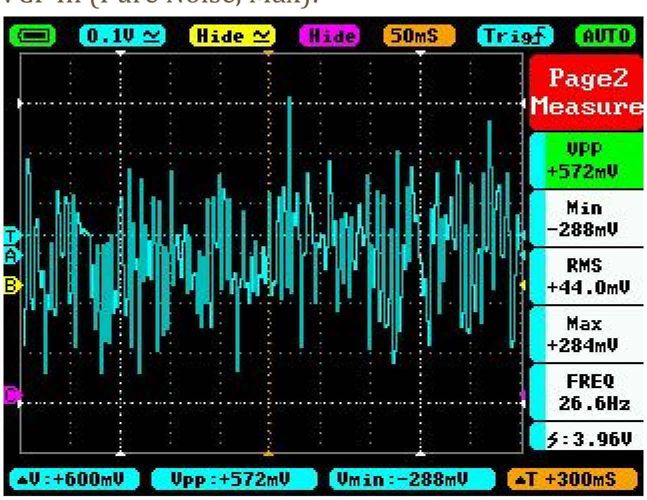
Sine VCA Amplifier out:



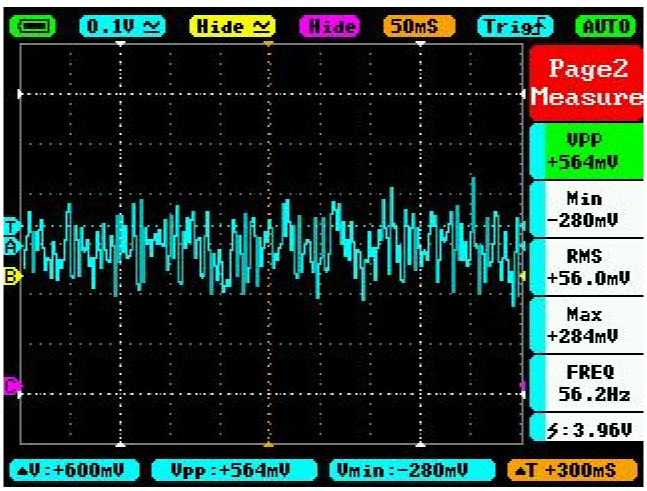
VCF In (Pure Noise, Typical):



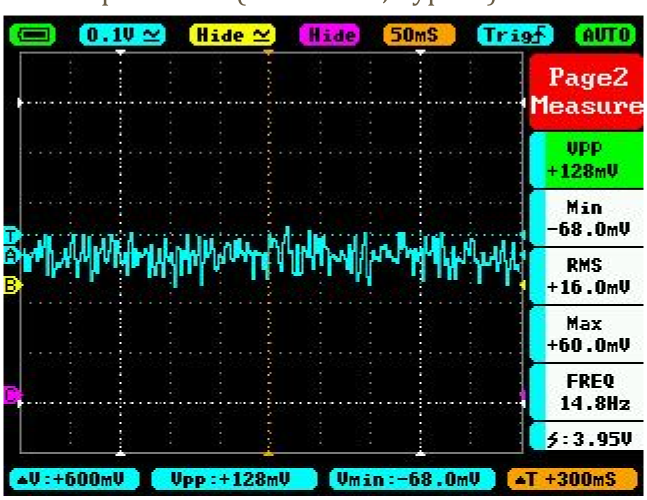
VCF In (Pure Noise, Max):



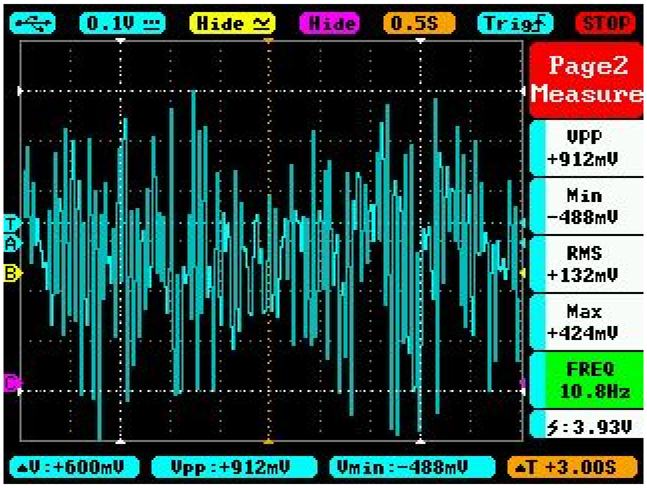
VCF -24dB Out (Pure Noise, Typical):



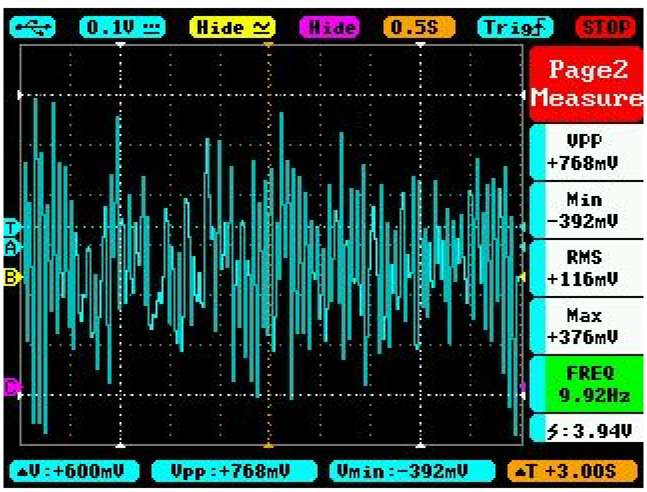
VCA Amplifier Out (Pure Noise, Typical):



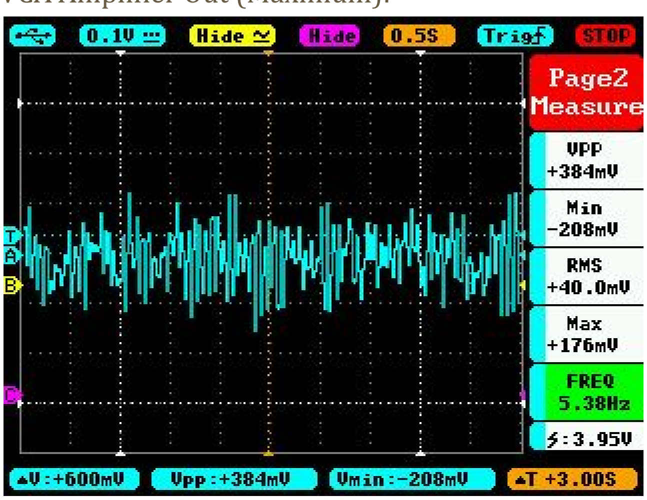
-12dB Out (Maximum):



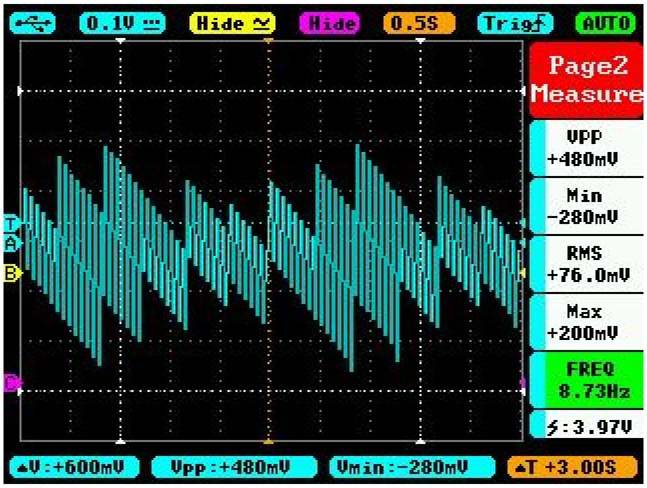
-24dB Out (Maximum):



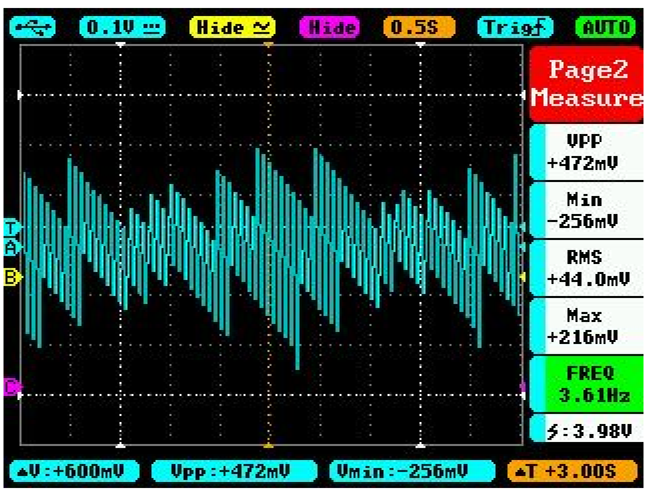
VCA Amplifier Out (Maximum):



-12dB Out (Typical):



-24 dB Out (Typical):



VCA Amplifier Out (Typical):