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 - Problem and Solution Overview

We plan to build an analog/digital hybrid synthesizer for musicians. This market is currently looking for highly customizable analog synthesizers with the conveniences of digital control due to the changing nature of their industry. The recent resurgence of analog synthesizers is attributed to many musicians perceiving sound quality of analog devices to surpass those of their digital counterparts [1].

Of these analog devices, the modular synthesizers are the most sought after, due to their control over sound manipulation. However, these synthesizers aren't digitally-compatible, make up the most expensive end of the market, and require the use of patch cables which can easily become cluttered.

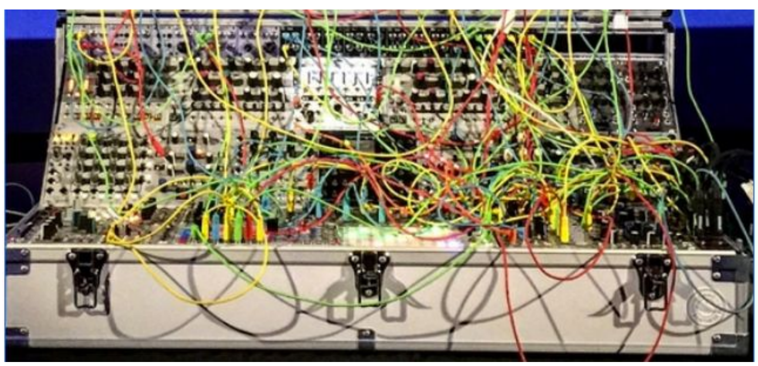


Image 1. Analog Modular Synthesizer

Our solution is to design a product that takes digitally-generated sound and routes it through an array of analog voltage-controlled filters and amplifiers. Our 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-$500 that will offer similar features of more expensive modular synthesizers and offer digital compatibility. The proposed synthesizer combines desirable qualities of analog, digital and modular synthesis into one cost-effective and highly flexible device, allowing for a plethora of sound design and synthesis options.

1.2 - Solution In Context

We offer a relatively conservative design for our synthesizer, utilizing only a USB port and stereo audio outputs, and potentiometers with enable/disable switches for setting the control levels of filters and amplifiers manually. This feature could be quite useful for real time, tactile control, and will allow the user to ensure parameters aren't zero whenever they aren't connected by 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 pathing grid or some other useful visual data could benefit the user and would make use of the full capabilities of our microprocessor.

// why did i delete the diagram showing the context

// this is the physical design

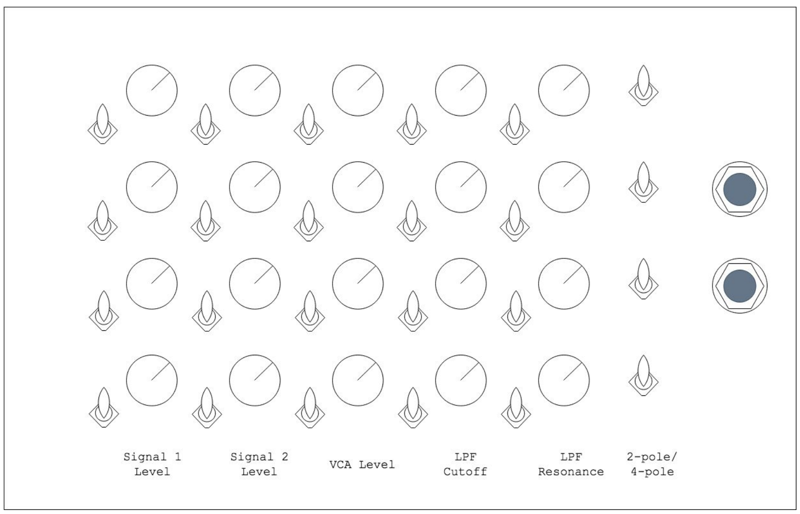


Image 2. Design Physical Layout

1.3 - High-Level Requirements

- An easily decipherable graphical user interface (GUI) where the consumer controls the synthesizer's audio outputs. It allows them to select, modify and control the sounds being produced by the synthesizer with the convenience of a computer program.
- An integrated digital interface (microcontroller) with a dedicated power supply which takes user input from the graphical interface/controller and uses the input to perform low-level functions in order to produce digital/discretized versions of the desired waveform.
- A set of (or a single appropriately full-featured) digital-to-analog converter, which takes the discretized waveforms produced by the microprocessor and converts them into an analog output.

- A set of analog filter and amplifier chips through, appropriately buffered to prevent accidental destruction or damage.
- 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) on any of the filter and vca chips, aside from connections from a given chip's outputs to its own inputs (to prevent damage).

2 Design

2.1.1 Overall Block Diagram

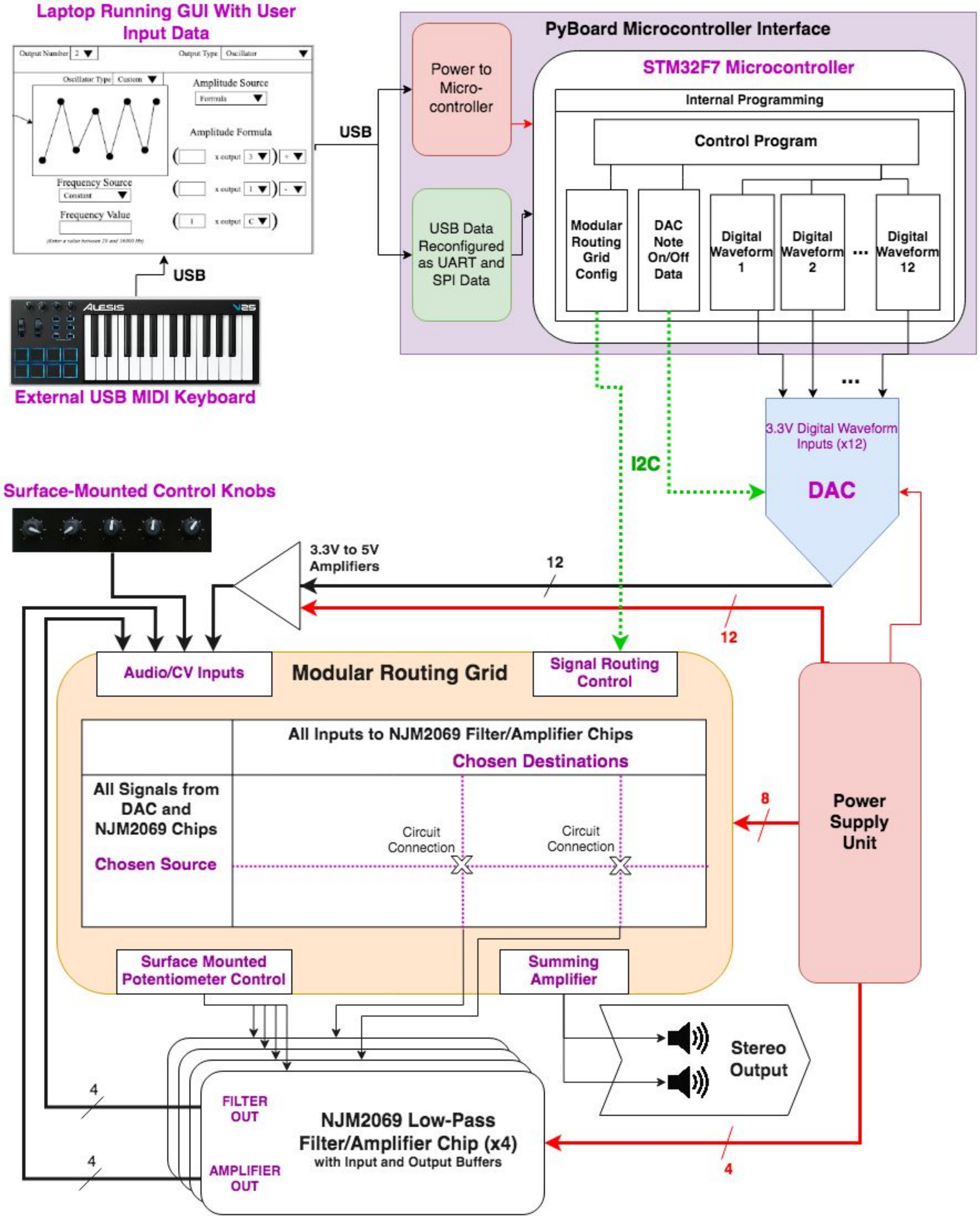
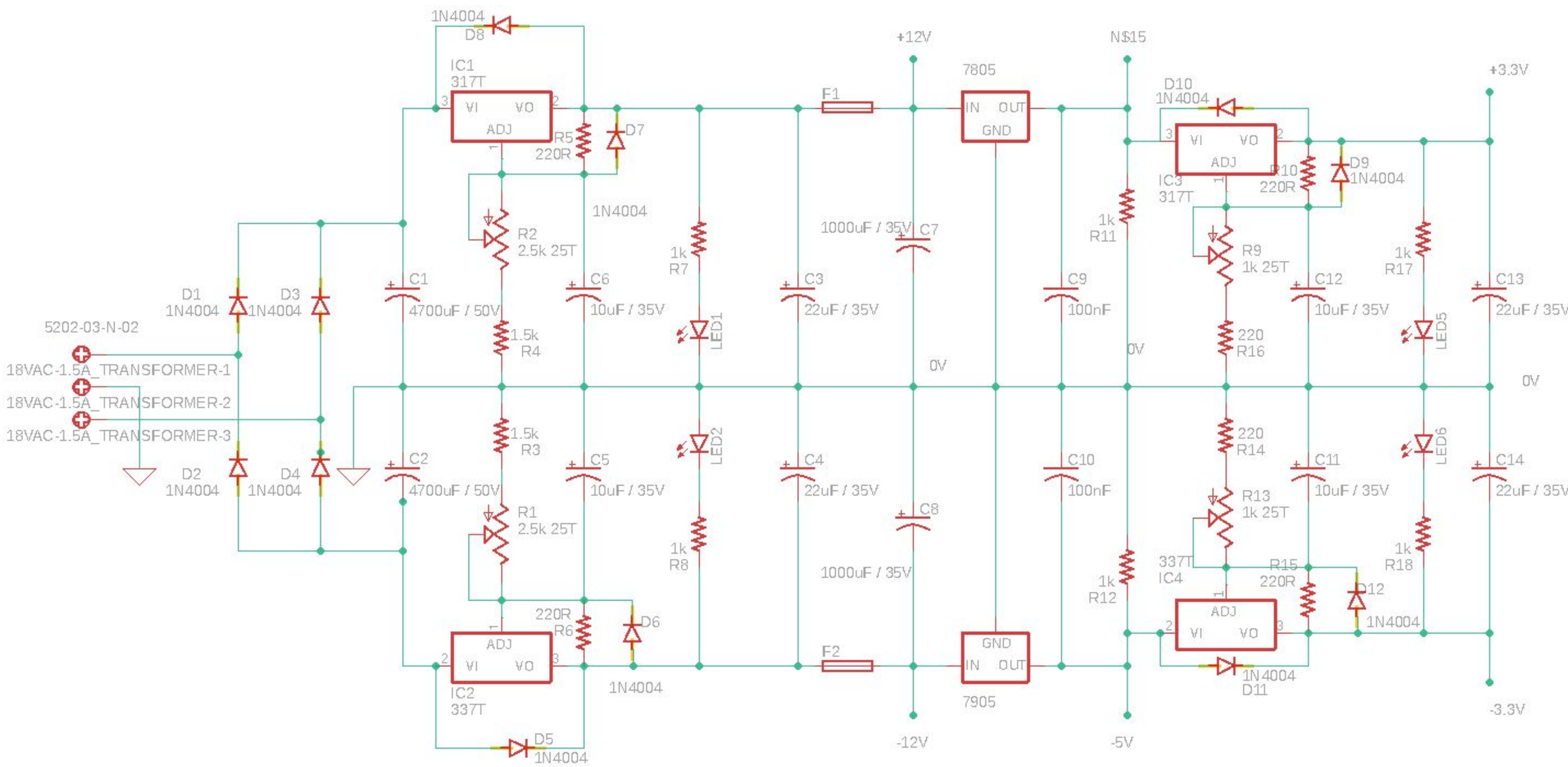


Image 3. Overall Block Diagram

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)



The power supply consists of 2 adjustable LM317 positive voltage regulators, 2 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].

Image 4. Power Supply Diagram

2.3 Software Interface

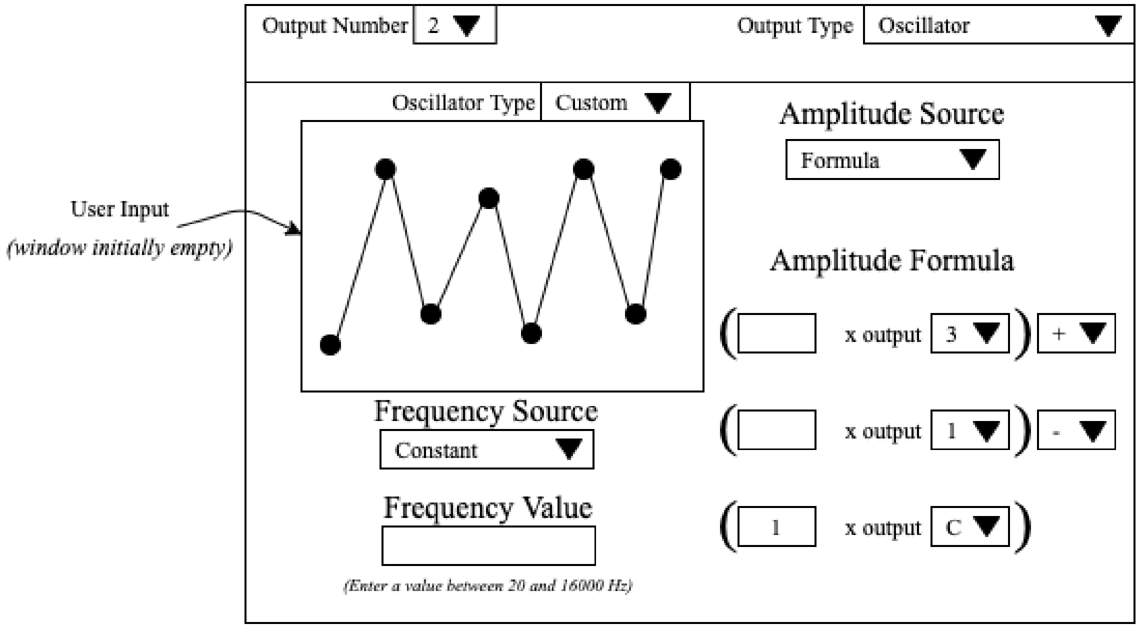
Input: User input data for each waveform generator in use: Output number, Output on/off, Output Type, Frequency Source, Frequency Value, Amplitude Source, Amplitude Value

Output: 12-bit arrays communicated serially through the USB virtual comm port (USB_VCP) representing the status of each user input selection:

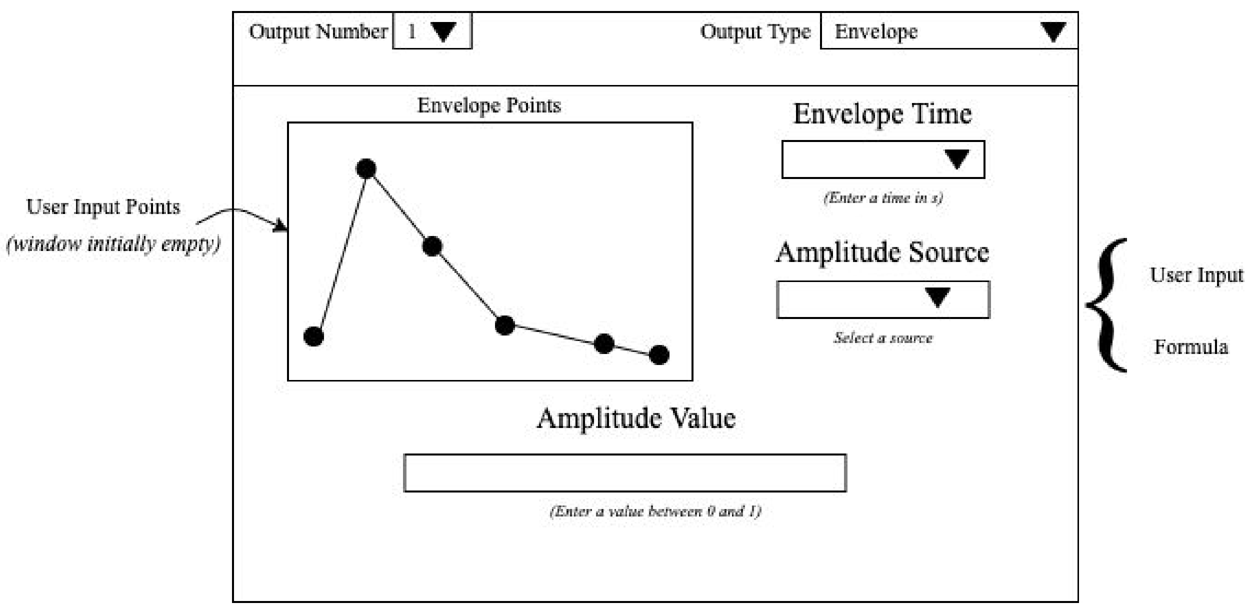
INTERFACE
This serves as the access point for user interaction with the synthesizer. Twelve digital waveforms will be independently accessible, and can serve as oscillators or controls. Oscillators produce audible notes and sounds to be routed through the analog section, whereas controls are used to create envelopes to modulate aspects of the analog section over time, such as filter resonance or cutoff. Two other windows (not pictured) will be included wherein control voltages can be set at a fixed level for the analog section and the routing grid can be visualized.

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 which is a monumental advantage. Utilizing oscillators effectively can create widespread effects from slow back-and-forth modulation to frequency/amplitude modulation.

DISPLAY EXAMPLE: Oscillator Window



DISPLAY EXAMPLE: Envelope Window



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

Table 1. Oscillator Configurations and Sources

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

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. Each will limit the input range from -*1* to *1* -- most negative to most positive microcontroller output.

The envelope, however, is only designed to represent one type. Our interface also only allows the user to set the maximum amplitude. All envelopes will have an input range from *0* to *1* -- no output voltage to full output voltage, respectively.

Amplitude and frequency can be controlled in 3 ways; as a constant, using the MIDI Note Number and velocity (respectively) of an input keyboard, or using a formula, combining previously defined digital waveforms into an algebraic expression.

IMPLEMENTATION

We intend to implement the digital section using an STM32F767 32-bit microprocessor embedded on a D-series PyBoard, which will allow for microprocessor code to be written in microPython (an extremely 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.

The graphical interface will be implemented using one of the standard GUI peripheral libraries available in python: Tkinter, wxPython or PyQT are likely candidates.

Requirement	Verification
Display appears and works as designed	1.) Make sure each of the menus, interfaces and user input fields appear in each of the windows as designed 2.) Make sure all menus are togglable and feature the items they are intended to feature 3.) Check to see if values can be entered in each of the user input boxes 4.) Test each of the grid points in the routing grid to ensure they are togglable
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.
Must be easily operable	1) Utilize different students/test subjects (if possible) to test our user interface and return feedback

2.4 Microcontroller/Control Program

Input: 12-bit arrays 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. A pinout Diagram of our PyBoard is shown below. There are 80 available outputs, separated into two busses of 40.

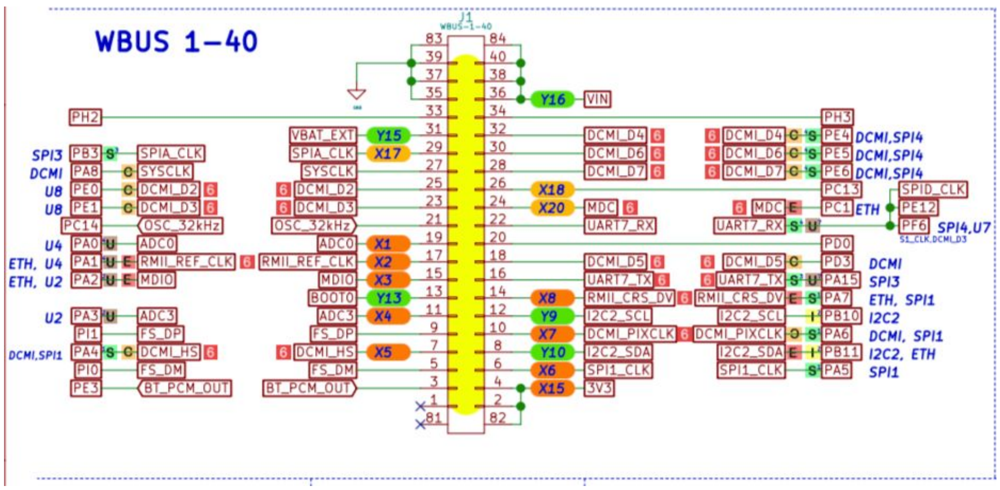


Image 5. PyBoard WBUS 1-40

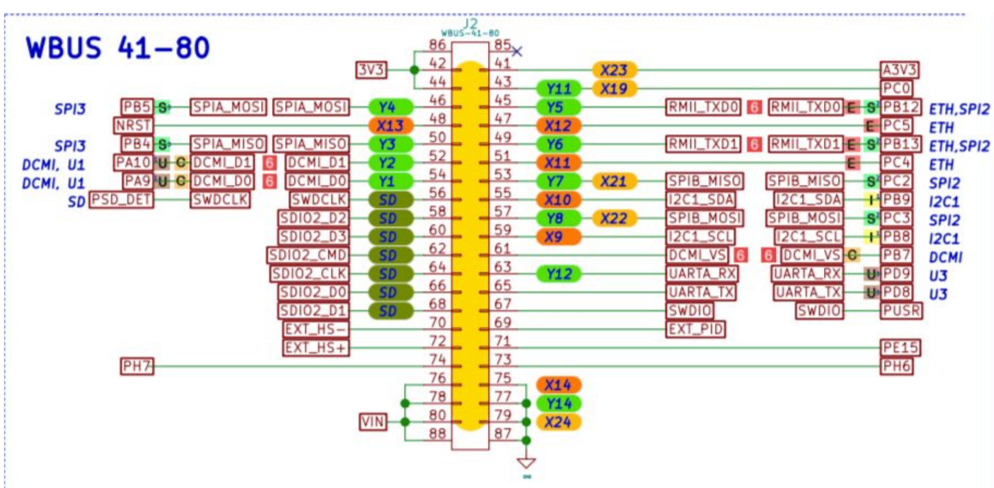


Image 6. PyBoard WBUS 41-80

The GPIO pins which are not occupied by a peripheral in use by the system or our 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. As we have yet to receive the PyBoard, we haven't specifically demarcated which pins will be used for which outputs as of yet.

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 Ensure processing delay < 10ms
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
Python Board integration	1. Call a program using the analog input, determine if the program is processing and outputting information appropriately. 2. Output through the board should mirror output in system tests.

2.4.2 Underlying Control Program

Wavetable Synthesis

The algorithm we intend to use to produce the offered 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 5 waveforms we wish to implement (sine, triangle, square, sawtooth and 'custom', which will have to load user-defined waves into wavetables and will therefore take slightly longer to be initially available as analog output). 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 1.31MB. 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.

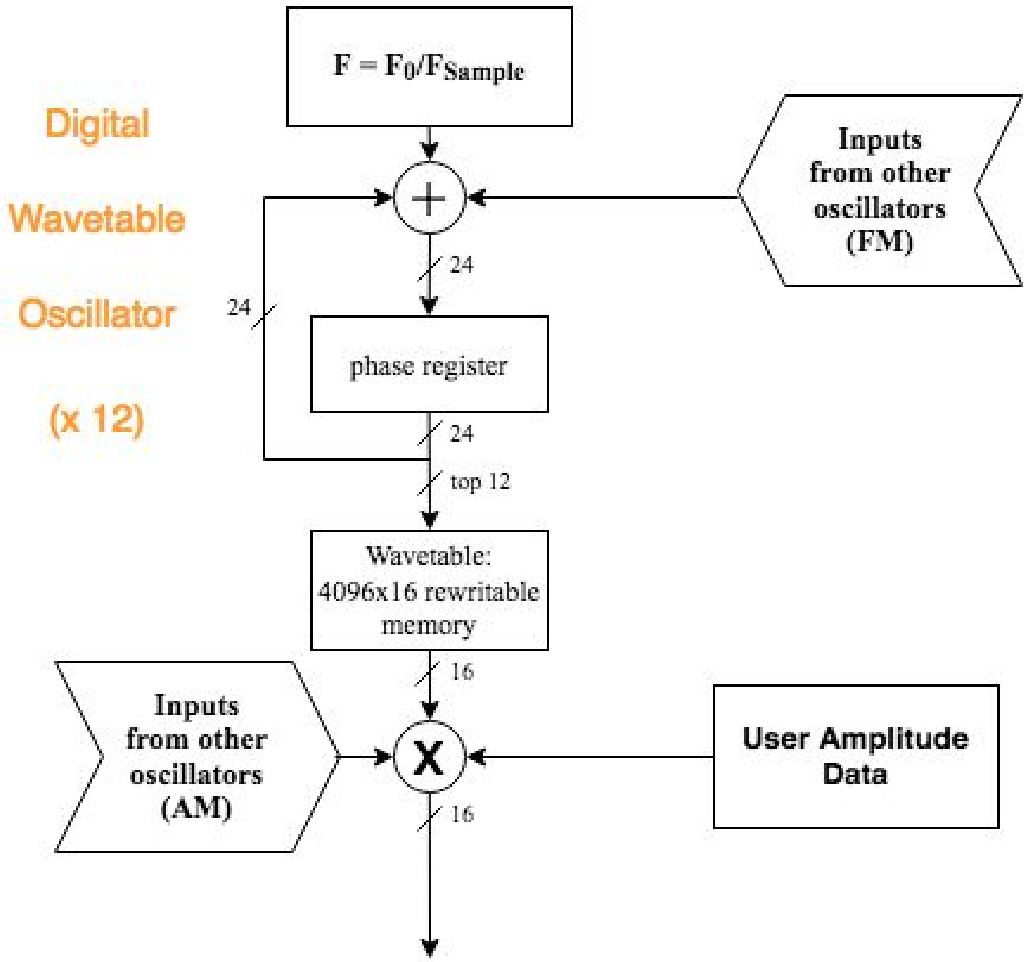


Image 7: Phase Oscillator Block Diagram

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.

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 will pass the waveforms/envelopes/static control signals to the DACs and handle 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.

Requirement	Verification
0-3.3V (+/- 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
0-3.3V (+/- 0.3V error) continuous Analog Output signalOutput at DAC output pins	Provide input, ensure output changes with differing user input
Interface DAC with microcontroller through I2C	1. Ensure device address is set through pins 31:32 2. Ensure corresponding device address is in use by microprocessor to interface with the DAC (set in programming) 3. Connect I2C SCL and SDA outputs from microcontroller (pins P88 and P89) to SCL and SDA pins on DAC

2.6 Routing Matrix

Input: (+/-) 0-1V inputs from Analog-to-Digital Interface, (+/-) 0-1V inputs from Routing Matrix, +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, all 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
Proper routing	Patch different signal combinations through, check for output of corresponding signal combinations
Verify that signals are not degraded excessively (THD >= 5%) in routing process	

2.7 Analog Filter/Amplifier Section

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

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

Buffered Analog Filter/Amplifier Block Diagram

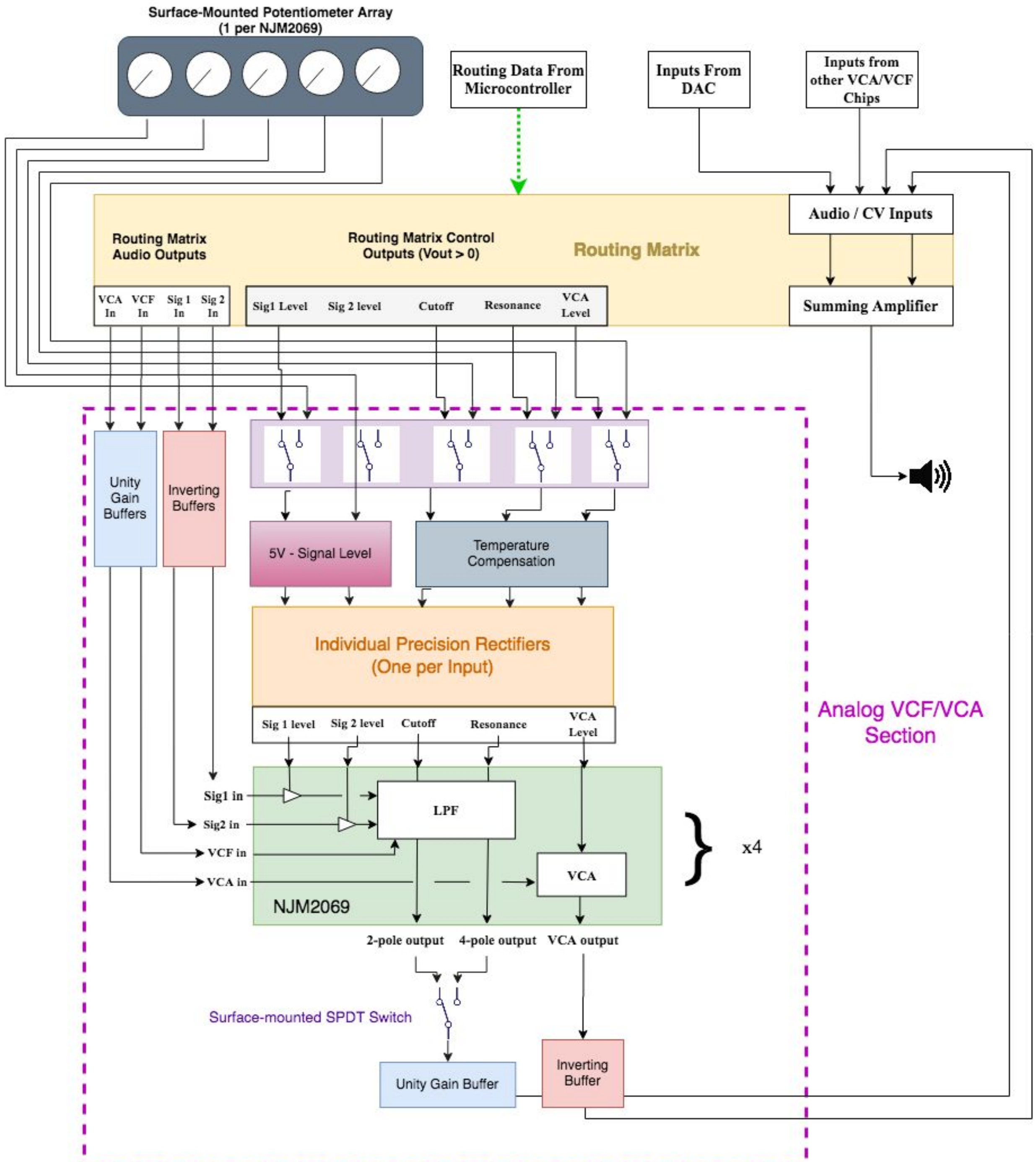


Image 8: Analog Filter/Amplifier Sub-block Diagram

This section consists of the NJM2069 analog filter/amplifier chips, as well as buffers for each input and output implemented using NE5532 operational amplifiers. The TL072 has been selected because it has a high slew rate allowing it to attain high audio frequencies with relative ease, low total harmonic distortion of around 0.003% in most applications, and a low current draw of around 1.4mA per op-amp. All of these features make it well suited to our purposes, especially given the high number of op-amps we will be using for buffering purposes.

The NJM2069 possesses signal inputs that are processed through the 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, such that 5V represents the minimum level and 0V represents the maximum level[5]. Given this, the rather complicated-looking buffer scheme shown above must be adopted.

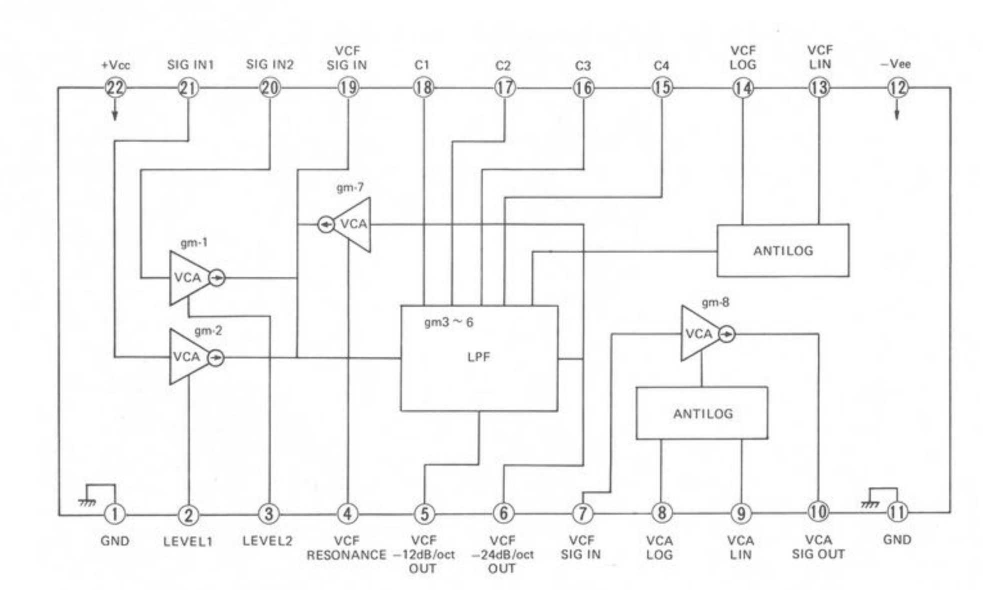


Image 7. NJM2069 Pinout and Interior Block Diagram[5]

Note: Pin 7 is mislabeled as VCF SIG IN; it is actually VCA SIG IN

Name	**Pin**	**Type**	**Inverting? (Inv.)**	**Voltage Range** (Input clips past +/-5V)	**Buffer Type**
Signal 1 In	21	Sig. Output	Yes	Up to +/-12V	Inverting Unity Gain
Signal 2 In	20	Sig. Output	Yes	Up to +/-12V	Inverting Unity Gain
VCF In	19	Sig. Output	No	Up to +/-12V	Unity Gain
VCA In	7	Sig. Output	No	Up to +/-12V	Unity Gain
Sig 1 Level	2	CTRL Input	Yes	0V to 5V	Inverting Full-Wave R
VCF Cutoff (VCF Log)	14	CTRL Input	No	0V to 5V	Full Wave Rectifier
VCF Resonance	4	CTRL Input	No	0V to 5V	Full Wave Rectifier

Name	Pin	Type	Inverting? (Inv.)	Voltage Range (Input clips past +/-5V)	Buffer Type
VCA Level (VCA Lin)	9	CTRL Input	Yes	0V to 5V	Full Wave Rectifier
VCA Out	10	Sig. Output	Yes	+/- 5V	Inverting Buffer
VCF Out (-24dB Out)	6	Sig. Output	No	+/-5V	Unity Gain Buffer

Table 2. Summary of NJM2069 I/O, their modes of action, and the buffer type

Main Filter Block

We will use four NJM2069 chips to implement the analog filter/amplifier section. A schematic illustrating their use in context is shown below, taken directly from the Korg DW-8000 service manual. 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 the inputs and outputs in different ways.

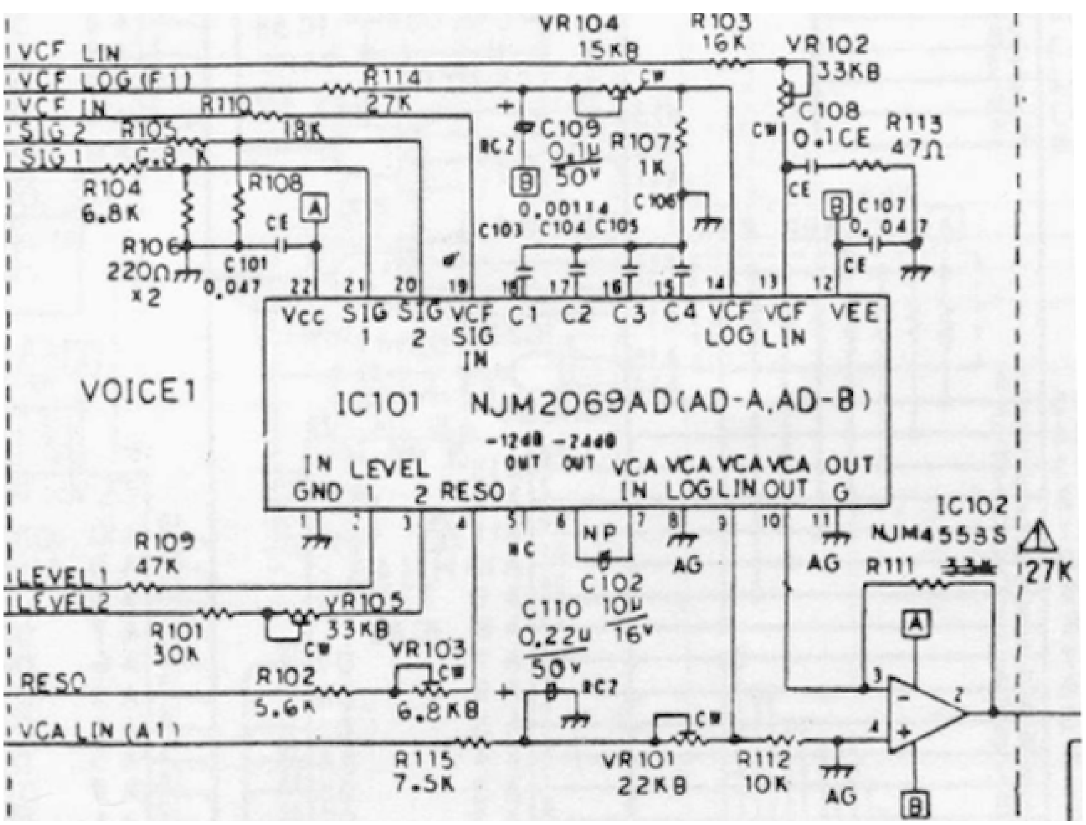


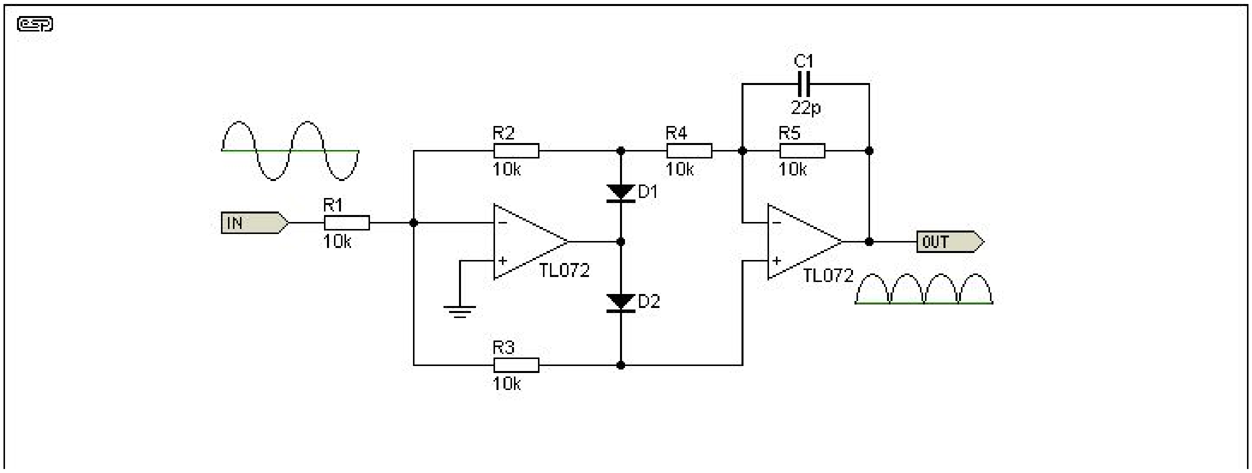
Image 8. Filter Schematic Using the NJM2069 taken from DW-8000 Manual[4]

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

Measurements Taken at Full Circuit Inputs/Outputs

Name	Pin	Test Location	Min	Max	$Vp\text{-}p_{typical}$	$Vp\text{-}p_{max}$	Notes
Signal 1 In	21	R104	-104mV	+130mV	208mV	224mV	Min Vp-p : 140mV
VCF In	19	R110	-288mV	+284mV	208mV	572mV	
VCA In	7	Pin 7	-384mV	+384mV	472mV	768mV	
Sig 1 Level	2	R109	0mV	+336mV	N/A	N/A	Sig 2 Level Similar
VCF Log	14	R114	380mV	-364mV	N/A	N/A	Inverted AC Control
Resonance	4	R102	0mV	+316mV	N/A	N/A	
VCA Lin	9	R115	-368mV	0mV	N/A	N/A	Inverted DC Control
VCF Lin	13	R103	N/A	+208mV	N/A	N/A	Stable Value
VCA Out	10	IC102 Pin 2	-208mV	176mV	224mV	384mV	
-12dB Out	6	Pin 6	-488mV	+424mV	480mV	912mV	
-24dB Out	5	Pin 5	-392mV	+376mV		768mV	

Op-amps at the inputs and outputs of each of the controls and signals will serve to amplify or reduce the outputs to a homogeneous/circuit-appropriate level, so all signals can be routed in and out of the device. To better correspond with control voltages, and using the bipolar VCF Log control voltage as a reference, output signals will be converted from typical levels of ~500mV peak-to-peak (VCF out) and ~250mV peak-to-peak (VCA output) to approximately 750mV peak-to-peak. Control inputs will be buffered with unity gain buffers and a precision rectification circuit (shown below) at the Resonance, Sig1 Level, and VCA Lin inputs, with the VCA Lin using inverting inputs. Signal 1, Signal 2 and VCF input levels will be reduced from 750mV (assumed) input to 200mV to correspond with measured input levels, while input levels to the VCA will be unchanged.



Precision Full Wave Rectifier Capable of Gain[12].

Requirement	Verification
Output levels standardized to 750mVp-p (+/-500mV)	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) +/66% and with proper filtering if necessary	Build based on schematics; ensure circuit output matches calculated output.

2.8 Risk 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-1[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^o to 150^{o}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 it's 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 it's 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 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^{o}C to 120 mA at 105^{o}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].

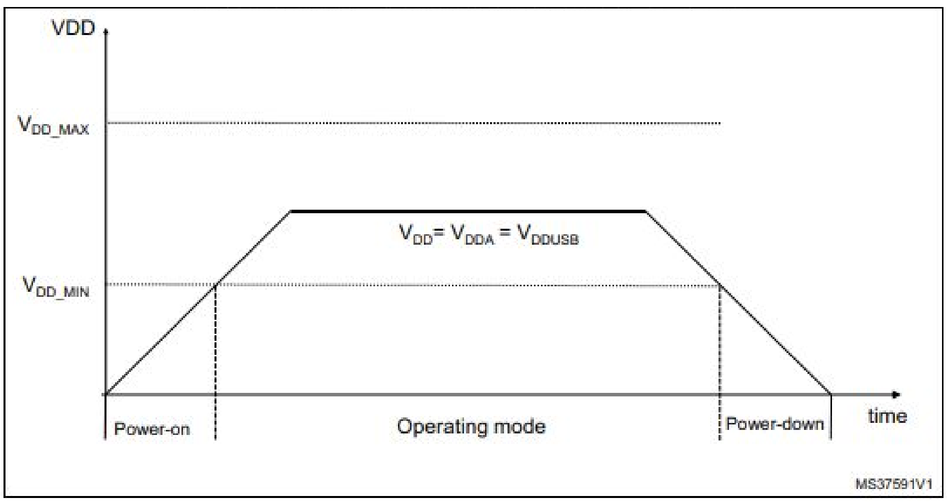


Image 11. $V_{DD,}$ USB connected to V_{DD} Power Supply[9]

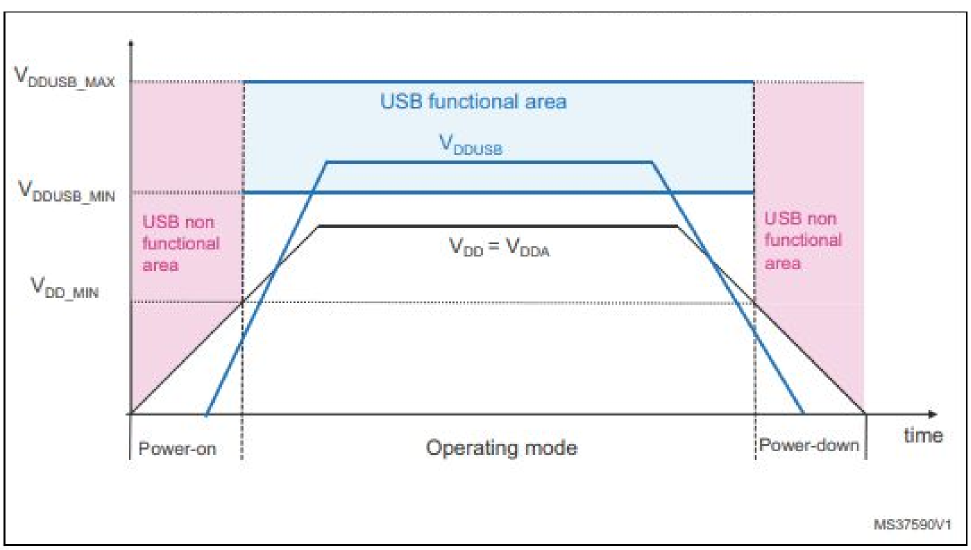


Image 12. $V_{DD,}$ USB connected to External Power Supply[9]

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 regulations, all hardware designs and constructions would be completed off campus. One team member has a digital oscilloscope, signal generator, multimeter, and all other tools necessary to test the results of the project

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

Team Total: $40,500

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: 10
    - 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

Part Total: $342.42

3.3 Total Cost

$40,842.42

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 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, and hand sanitizer at minimum.

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.

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

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.

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. This is described in further detail in appendix 2.

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.

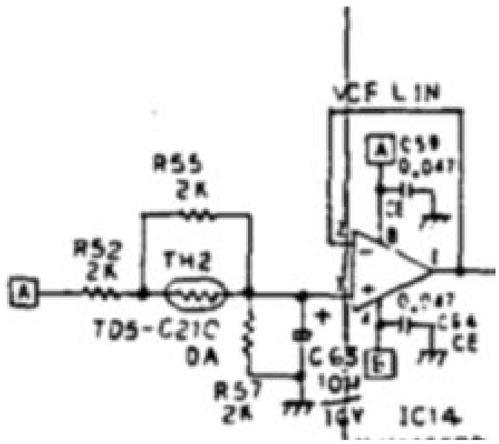


Image 10: Thermal Regulation for VCA control voltage, from DW-8000 Service Manual[4]

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

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).

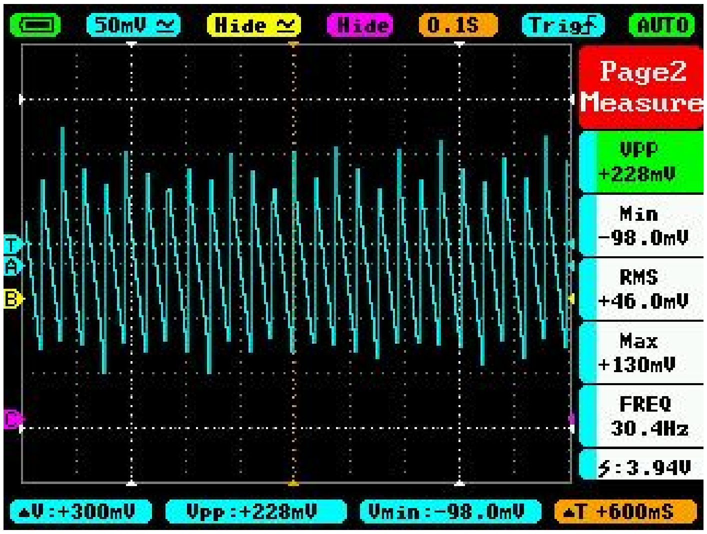
First, thermistors were desoldered from the DW-8000 board. Three thermistors are used, labeled TH1, TH2, and TH3. The resistance of the thermistors were tested in the lab, where the temperature readout on the thermostat was 73.5 degrees fahrenheit, corresponding to 23.0555 degrees celsius or 296.20555 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.

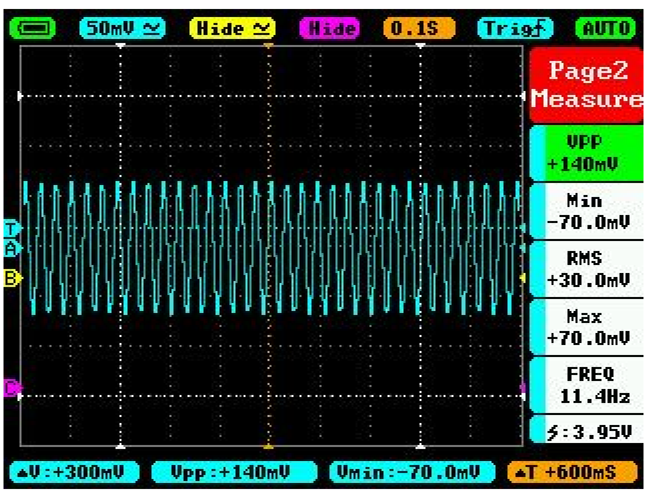
Appendix 3

Synthesizer Measurements

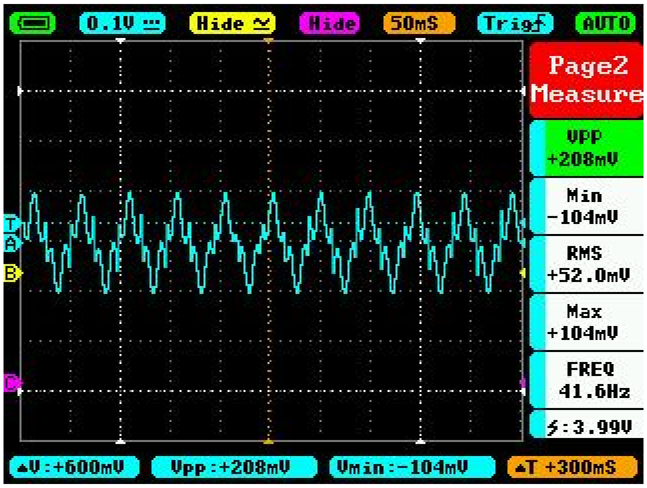
Measurements were taken using a portable digital oscilloscope
Saw Sig2 In:



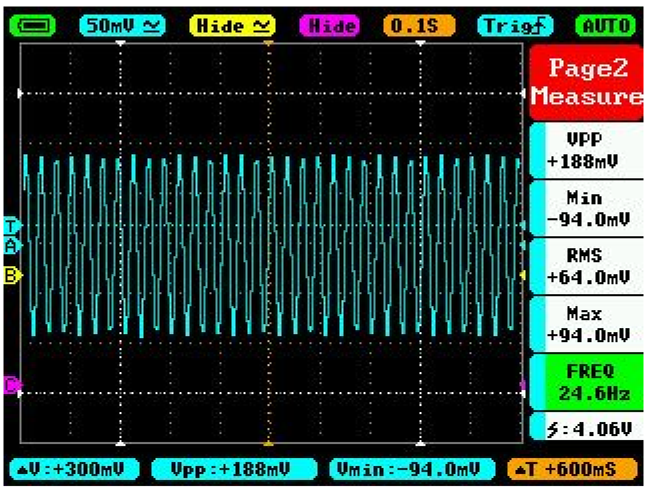
Sin Sig1 In:



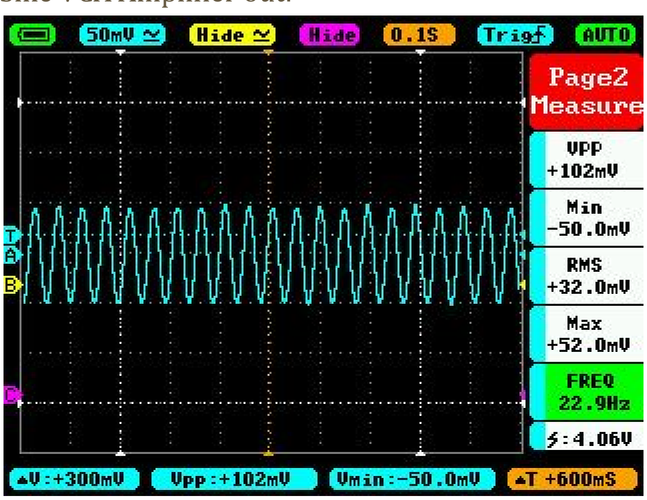
Square-ish Sig1 In:



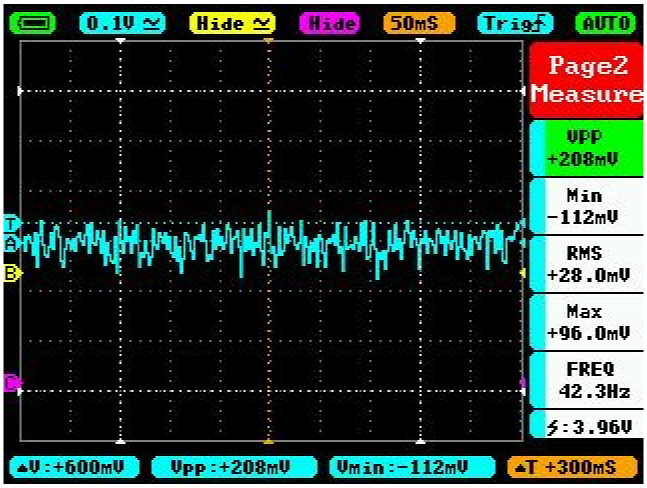
Sine -24dB out:



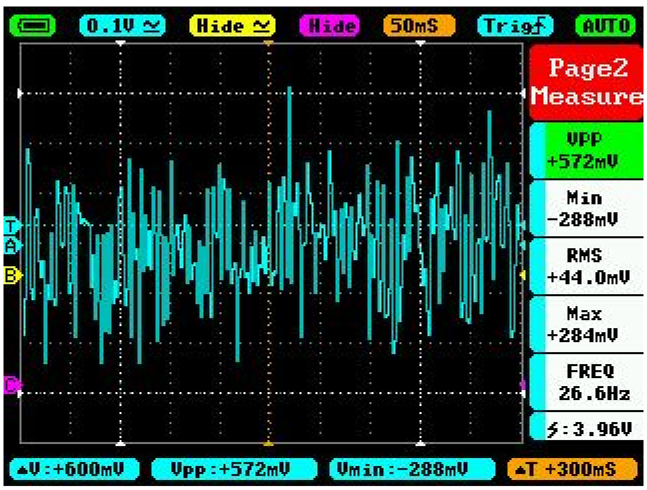
Sine VCA Amplifier out:



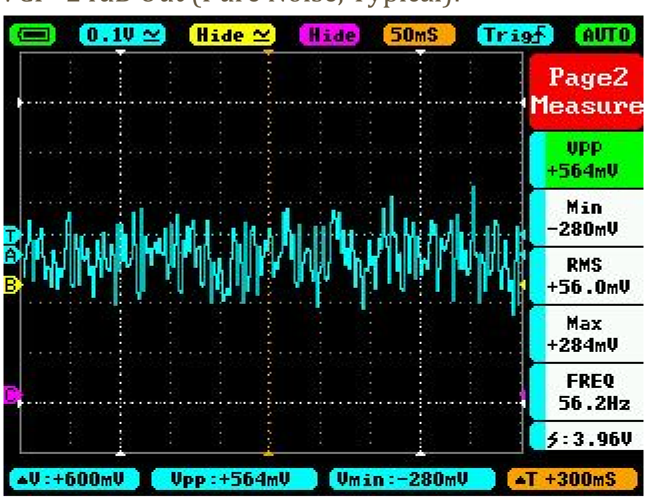
VCF In (Pure Noise, Typical):



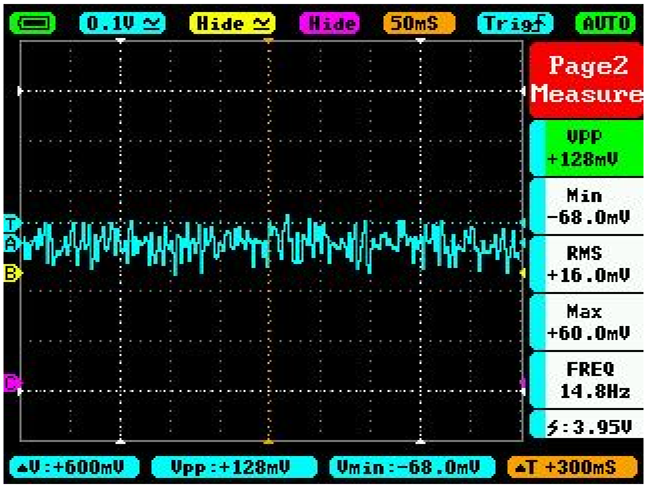
VCF In (Pure Noise, Max):



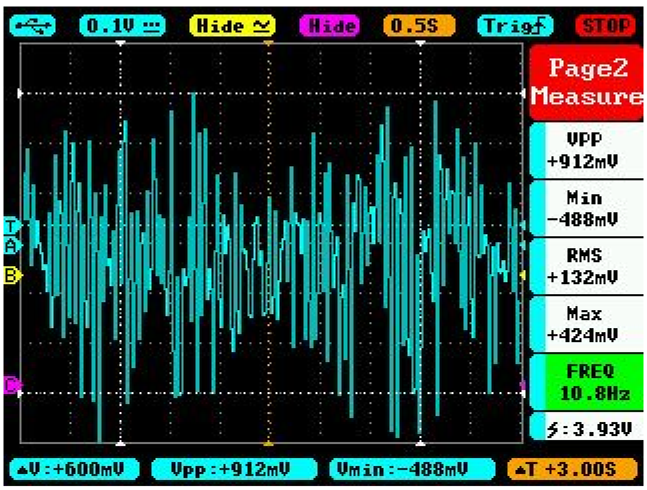
VCF -24dB Out (Pure Noise, Typical):



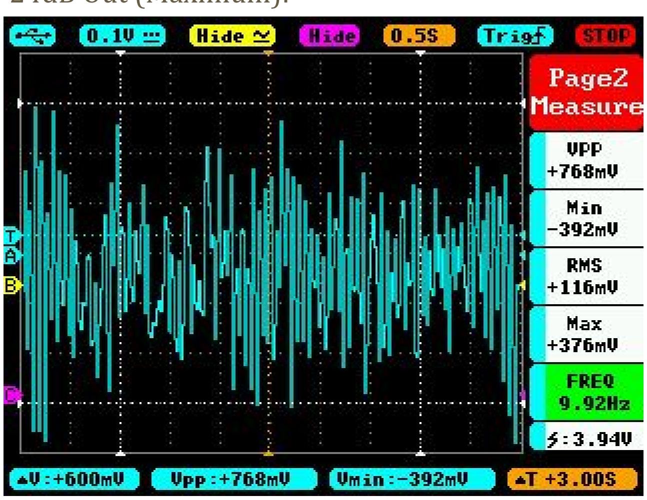
VCA Amplifier Out (Pure Noise, Typical):



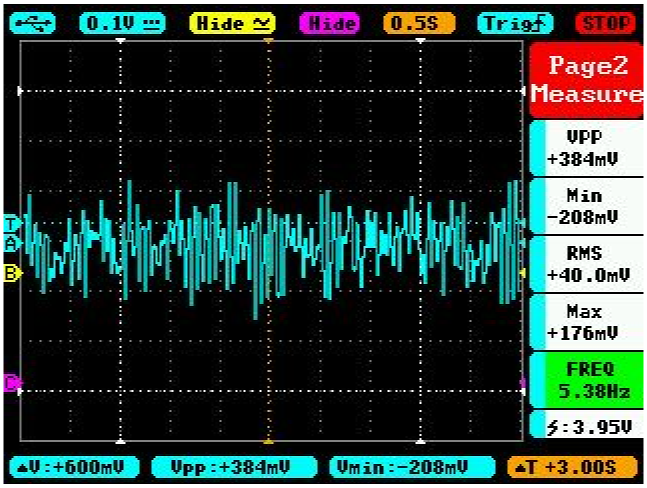
-12dB Out (Maximum):



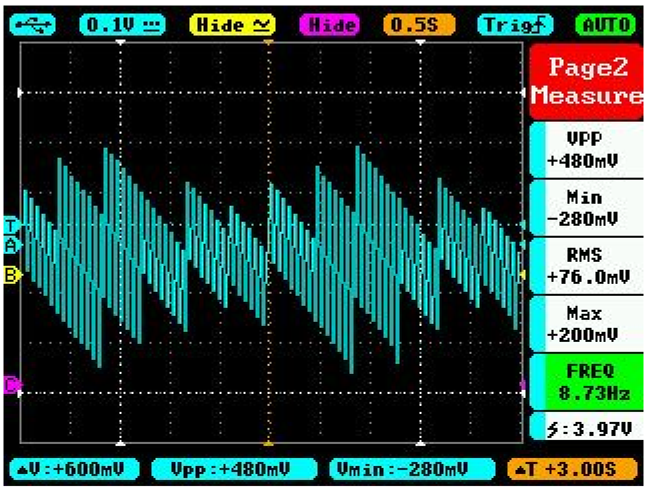
-24dB Out (Maximum):



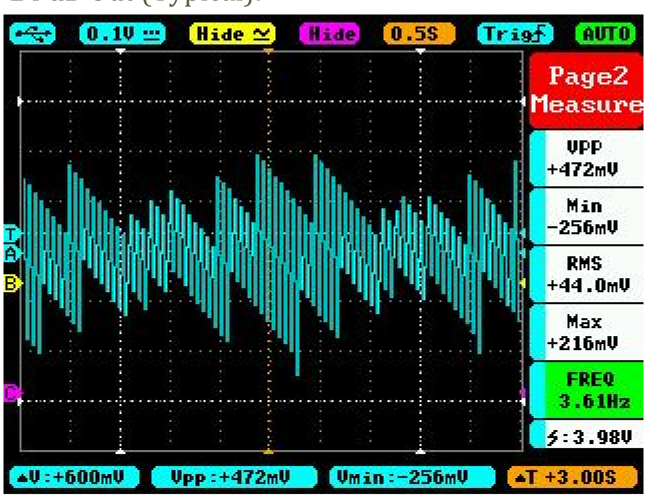
VCA Amplifier Out (Maximum):



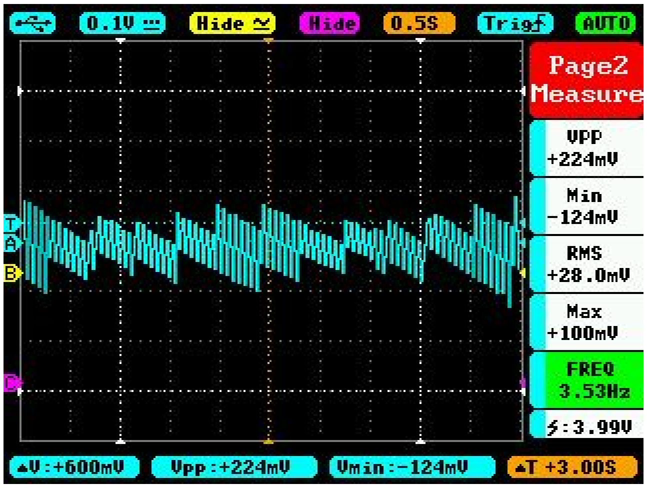
-12dB Out (Typical):



-24 dB Out (Typical):



VCA Amplifier Out (Typical):


0.1V
Hide
Hide
0.5S
Trig
AUTO
Page2
Measure
VPP
+224mV
Min
-124mV
RMS
+28.0mV
Max
+100mV
FREQ
3.53Hz
:3.99V
V:+600mV
Vpp:+224mV
Vmin:-124mV
T +3.00S