Synthtar – User’s Manual

Version 2 – published 5/16/19

Blake Nemeth – Vermont Technical College

ELT 2720 – Electrical Projects

Contents

[Section 1 – Basic Product Information 2](#_Toc6623098)

[Introduction – 2](#_Toc6623099)

[General Features – 2](#_Toc6623100)

[Physical Specifications – 2](#_Toc6623101)

[Electrical Specifications – 2](#_Toc6623102)

[Section 2 – Operating Instructions and Troubleshooting 3](#_Toc6623103)

[Parts of the Synthtar – 3](#_Toc6623104)

[Body – 3](#_Toc6623105)

[Neck – 4](#_Toc6623106)

[Playing – 4](#_Toc6623107)

[Turning it on – 4](#_Toc6623108)

[Adjusting the Volume – 4](#_Toc6623109)

[Root Notes – 4](#_Toc6623110)

[Individual Notes – 4](#_Toc6623111)

[Minor Switches – 4](#_Toc6623112)

[Building Chords – 4](#_Toc6623113)

[Navigating the Menu – 5](#_Toc6623114)

[Quick Switches – 6](#_Toc6623115)

[Analog Adjustments – 6](#_Toc6623116)

[FAQ/Troubleshooting – 7](#_Toc6623117)

[Output sounds weird after plugging in headphones – 7](#_Toc6623118)

[Replacing switches – 7](#_Toc6623119)

[Appendix A – Overall System Schematic 7](#_Toc6623120)

[Appendix B – Electrical Layout- Need to split up smaller 8](#_Toc6623121)

[Appendix C – PCB Layouts – Still splitting into smaller boards 8](#_Toc6623122)

[Appendix D – Playing Chart – Notes (Default Tuning) 8](#_Toc6623123)

[Appendix E – Playing Chart – Chords (Default Tuning) 8](#_Toc6623124)

[Bibliography and References – 11](#_Toc6623125)

# Section 1 – Basic Product Information

## Introduction –

The Synthtar was created as a tool for musicians and non-musicians to explore music in an easy-to-digest way with a playstyle similar to popular rhythm games. It can be used as a composition tool, a MIDI controller, or as a standalone instrument.

## General Features –

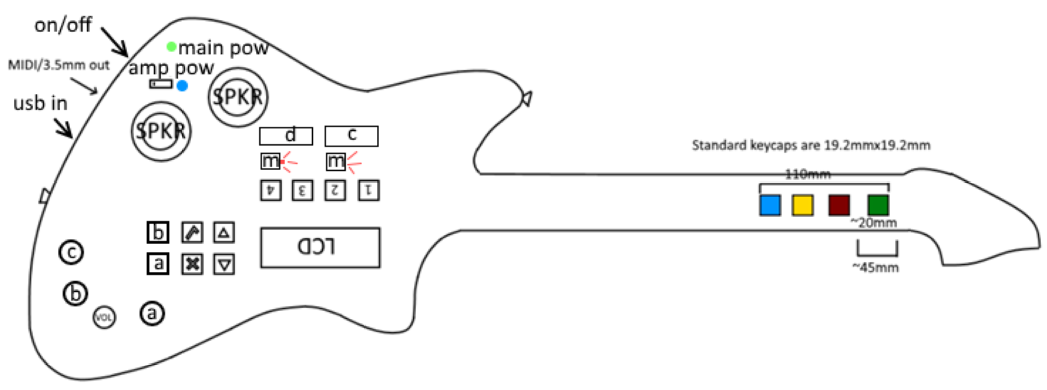
* User replaceable switches – Mechanical switches can be replaced by the user easily according to preference for tactile feel and loudness. By default, uses brown switches, which are tactile and non-clicky.
* Easily accessible chords – Unique playstyle allows for chords to be built with ease.
* Custom tunings – Don’t be limited by the world around you, make your own sound.
* Long battery life – spend more time playing and less time plugged into the wall. (V2)
* Pass-through charging – charge while you play, never miss a beat. (V2)
* Versatile – can act as a MIDI controller (via USB MIDI and standard MIDI out), MIDI synth (via USB MIDI) or a standalone instrument.

## Physical Specifications –

* Product weight: 1-2 pounds.
* Product dimensions: Approx 2.5’x11”x3”
* Charging port: Micro USB

## Electrical Specifications –

* 3000mAh Li-Ion Battery (V2)
* 30-hour battery life (V2)
* 6-hour charge time (V2)
* 5V input, maximum charge rate 500mA.
* 85mA current draw.



# Section 2 – Operating Instructions and Troubleshooting

## Parts of the Synthtar –

### Body –

#### Power Switch –

The power switch is located at the top of the device and is used to turn the device on or off. (V2)

V1 has no power switch, and is simply turned on by being plugged into power.

#### Amp Power Switch –

The power to the internal amplifier is toggled by a rocker switch near the speakers. Status is indicated by an adjacent LED.

#### Speakers + Amplifier –

There are 2 internal speakers that can be toggled by a nearby switch. Maximum output power is 3W (each).

#### Micro USB Port –

The Micro USB port is located at the bottom side of the body and is used for charging and USB MIDI communications (when plugged into a compatible computer). The Synthtar can also update its software via this port.

#### Main Switches –

These 4 switches are used to trigger notes to be played, are in the center of the body, and are labeled 1 through 4. Each one represents a “string,” similar to a string instrument, so each can act independently of one another. Each string can be “tuned” to a note relative to the “root note” (see below) plus or minus 12 (one octave in each direction). By default, these are tuned to the major 7th chord of the root.

#### Minor Switches –

Minor switches are used to access chords that wouldn’t normally be accessible by shifting the tuning for a specific string, in this case the 2nd and 4th strings, because those allow access more chord variations. When pressed, they’ll toggle the “minor” mode for their corresponding string, shifting its pitch down by 1 half step (allowing for minor chords to be accessed, hence the name).

#### Menu Switches –

Menu switches are used to navigate the menu and are located adjacent to the LCD. There are 4: a check, an x, an up and a down.

#### Analog Adjustments –

There are 4 dials – 3 smalls for user-configurable settings, 1 large for volume.

#### Quick Switches –

Quick switches are used to perform user-defined actions. There are 4, labeled A, B, C and D. C and D have wider keycaps so they can be pressed with the palm.

#### Indicators –

There are 4 indicators on the body: Minor2, Minor4, Power and Speaker. Minor2 and Minor4 are tied to the minor state of body buttons 2 and 4, and are located beneath the switch that toggles that state. Power is located near the power switch, and Speaker is located near the amplifier power switch.

#### Battery –

The Synthtar runs on a 3000mAh Li-Ion battery, located inside the battery compartment, accessible from the rear.

#### Display –

A 16x2 LCD that displays various system settings and values. Located on the lower face of the body.

#### Outputs – Sound –

There are 2 audio outputs – a pair of internal speakers and a 3.5mm jack.

#### Outputs – MIDI –

There are 2 MIDI outputs – a micro USB and a 5 pin DIN.

### Neck –

#### Neck Switches –

There are 4 neck switches, numbered 1,2,4 and 8 (and color-coded green, red, yellow, blue). These are used to adjust the root note, like the frets on a guitar.

## Playing –

### Turning it on –

To turn on the Synthtar, flip the power switch to the “on” position. When the device is booted, the green power indicator will be lit.

### Adjusting the Volume –

To adjust the volume, use the large dial at the bottom of the body face. Clockwise raises volume, counterclockwise lowers volume.

### Root Notes –

The “root note” is used to determine the root of the chord, and the individual notes available at a given time. The base root note is “E” by default and is adjusted by pressing the neck buttons or transposing. The neck buttons each have a value, 1, 2, 4 and 8. The sum of the button values pressed is the number of half steps above the base your root is. E.g. the 2 key is pressed, the root becomes F# (2 half steps above E). Add the 1 key, the sum is 3, so the root becomes G.

### Individual Notes –

Individual notes can be picked out using the body switches 1,2,3,4. These correspond to the 4 string tunings to determine the note played. By default, the tunings are {0,4,7,11}, and the tunings are measured in half steps. The note played is that string’s tuning plus the root. E.g. if the root is E and the 1 and 2 keys are pressed, the notes produced are the root + 0 and the root + 4, E and G#.

### Minor Switches –

Since that standard tuning won’t allow minor chords by default, minor switches allow the 2nd and 4th string to shift down ½ step to allow for minor and diminished chords. E.g. the tuning is {0,4,7,11} and you hit the minor switch below 4, the tuning becomes {0,4,7,10}.

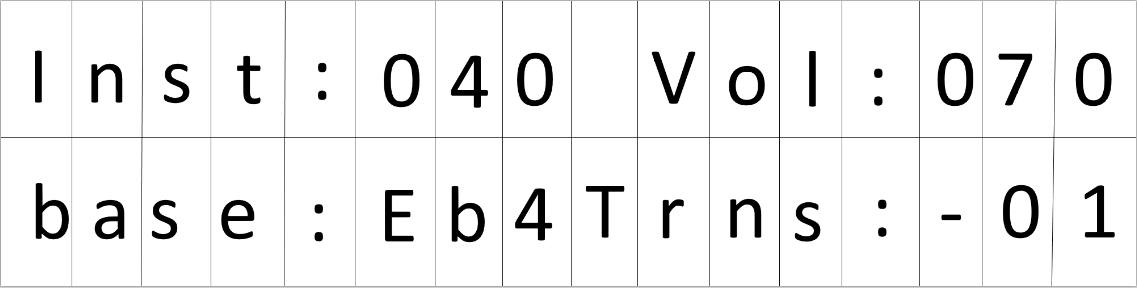
### Building Chords –

The default tunings are designed to allow access to many chords with minimal effort. For example, pressing the 1st, 2nd and 3rd keys on the body will produce the major triad of the selected root note, regardless of what that root note is. For a comprehensive chord chart, see Appendix E. 

### Navigating the Menu –

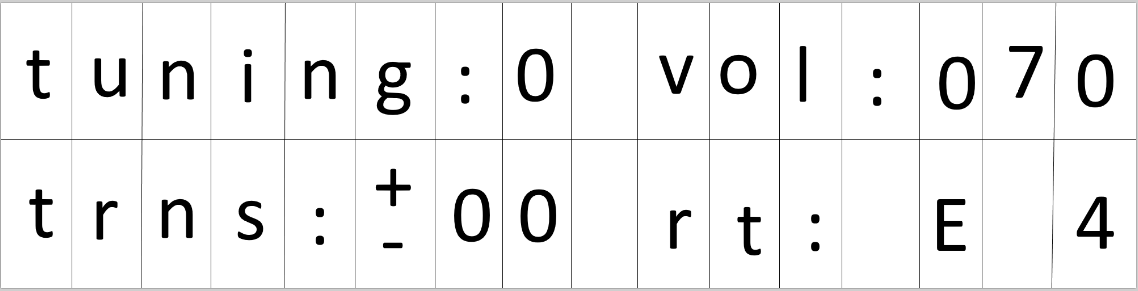
##### Menu A –

On Menu A, the instrument, volume, base note and transposition are shown. X goes to the next selection (Instrument and transposition are selectable in this menu). The up and down keys will adjust the currently selected value. The check mark key will change the screen to Menu B.



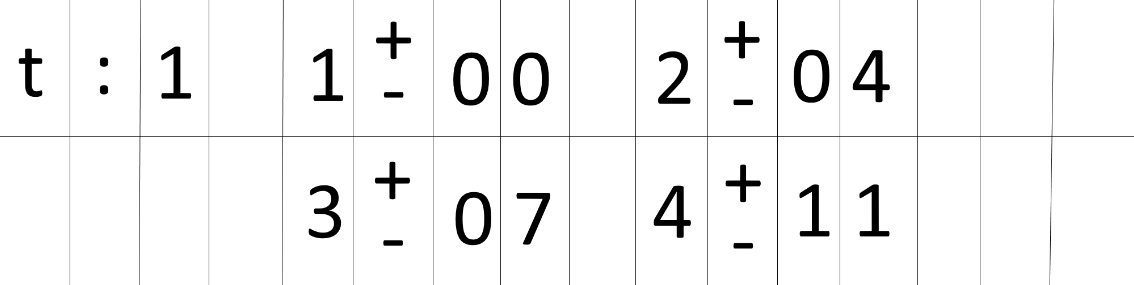
##### Menu B –

On Menu B, the tuning number, transposition, root note and volume are displayed. As in Menu A, the X will change the current selection, and the up and down keys will adjust the values. When “tuning” is selected and the check mark key is pressed, you’ll be brought to the “Tuning Adjust” menu for the current tuning slot. The check mark on transposition will return you to Menu A. Root note will change as neck buttons are pressed/released.



##### Tuning Adjust Menu –

There are 3 internal tuning slots. Each one has a pre-set value that can be reprogrammed by the user in this menu. To go to the next string, press the check key. To adjust the tuning up or down for the selected string, press the up or down key. To exit and go back to the previous menu, press X. Pressing up and down at the same time will reset the tuning to its default.



### Quick Switches –

Will be added in a future software update. These switches are user customizable to have various functions, which are listed below.

#### Quick Tuning –

Switches to a user-configurable tuning when pressed/while held (pressed/held is configurable as well).

#### Quick Transpose –

Shifts the base note up or down or sets it to a user defined value.

Shift –

Keeps the root note from changing while adjusting your fingers on the neck. Useful for tricky transitions.

### Analog Adjustments –

Will be added in a future software update. These parameters can be assigned to the 3 user-configurable analog knobs.

#### Attack –

Attack is the rising time of a note and affects how sharply or slowly it ramps to its peak before starting to fade.

#### Decay –

Decay is the falling time of the note after the attack and before the sustain.

#### Sustain –

The level during the main sequence of the sound. This one is *not* time.

#### Release –

The falling time to decay to 0 after the key is released.

#### Pan –

Adjusts the left-right of the output audio signal. Commonly seen as balance adjust.

#### Pitch Bend –

The bend of the note, like what a whammy bar on a guitar adjusts. In a future revision, this may be tied to a whammy bar.

## FAQ/Troubleshooting –

### Output sounds weird after plugging in headphones –

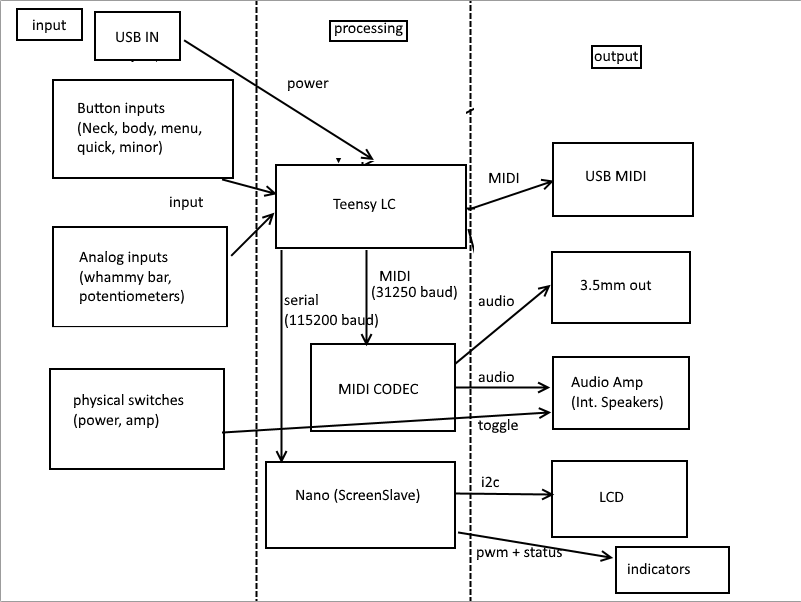
This is a known issue that occurs when plugging in headphones while the internal amplifier is active. To fix this, simply turn the device off and on again.

### Replacing switches –

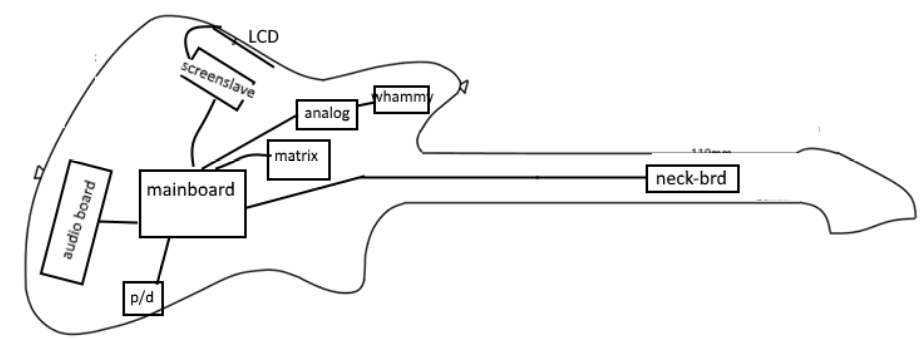
To replace a switch, first remove the keycaps on the switches in the same area and remove the underlying faceplate. Once the plate is removed, the switches will be visible; grab hold of it and pull it straight out. Put the new switch in the socket, replace the faceplate and replace the keycaps.

# Section 3 – Theory of Operation

## System diagram/process flow



## Physical Layout



## Electrical Operation:

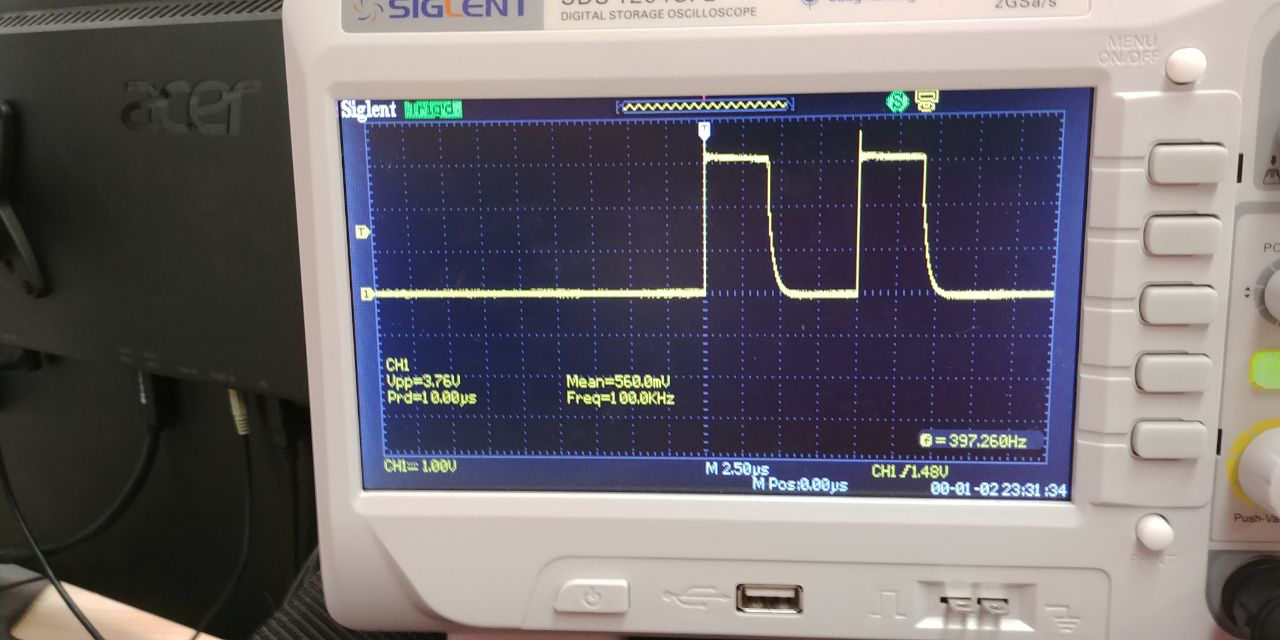
The operation of this device is mostly based in software. The only debuggable circuits are the matrix setup, the midi transmitter (V2), and the indicators.

### Input Matrix:

16 diodes that point towards the center of the board (on the body matrix). The switch connectors are reversible because the diodes are on the board. If a switch is missing, it will just register as un-pressed. If a switch isn’t working, check the wires connected to the socket. You can look at the signals on each row by putting a scope across its pulldown resistor.



*5ms matrix polling*



*Matrix polling on uS scale, looking at pulldown resistor*



*MIDI Command*

## Software Operation:

### Main MCU:

### String\_t Class

This class represents a “String” like on a guitar. As such, it can be retuned and played independently of other strings. This has getters and setters, and stores a current note, a tuning, and whether or not it’s currently playing. In addition, it has a few methods that use these values to start and stop midi notes.

### Input Matrix

Goes through the matrix by pulling one column high at a time and seeing which rows get activated. By knowing the only column with power, you can know the key that was pressed. Bit shifts them into an integer and returns it.

### Main Loop

The Teensy LC constantly polls its input matrix for changes. On the change of the neck buttons, it compares the sums of the neck keys (assigned 1,2,4,8 through bit shifts) and uses this to compute the new “String” sum. The new sum is applied to shift the notes being played by the body keys and the appropriate data is sent to the ScreenSlave. On the change of the body keys, it starts/stops the corresponding string. The minor switches change the tuning of their corresponding strings down by 1.

A test for Teensy functionality is to plug it in via USB and see if it connects as a USB device descriptor. If not, something is wrong.

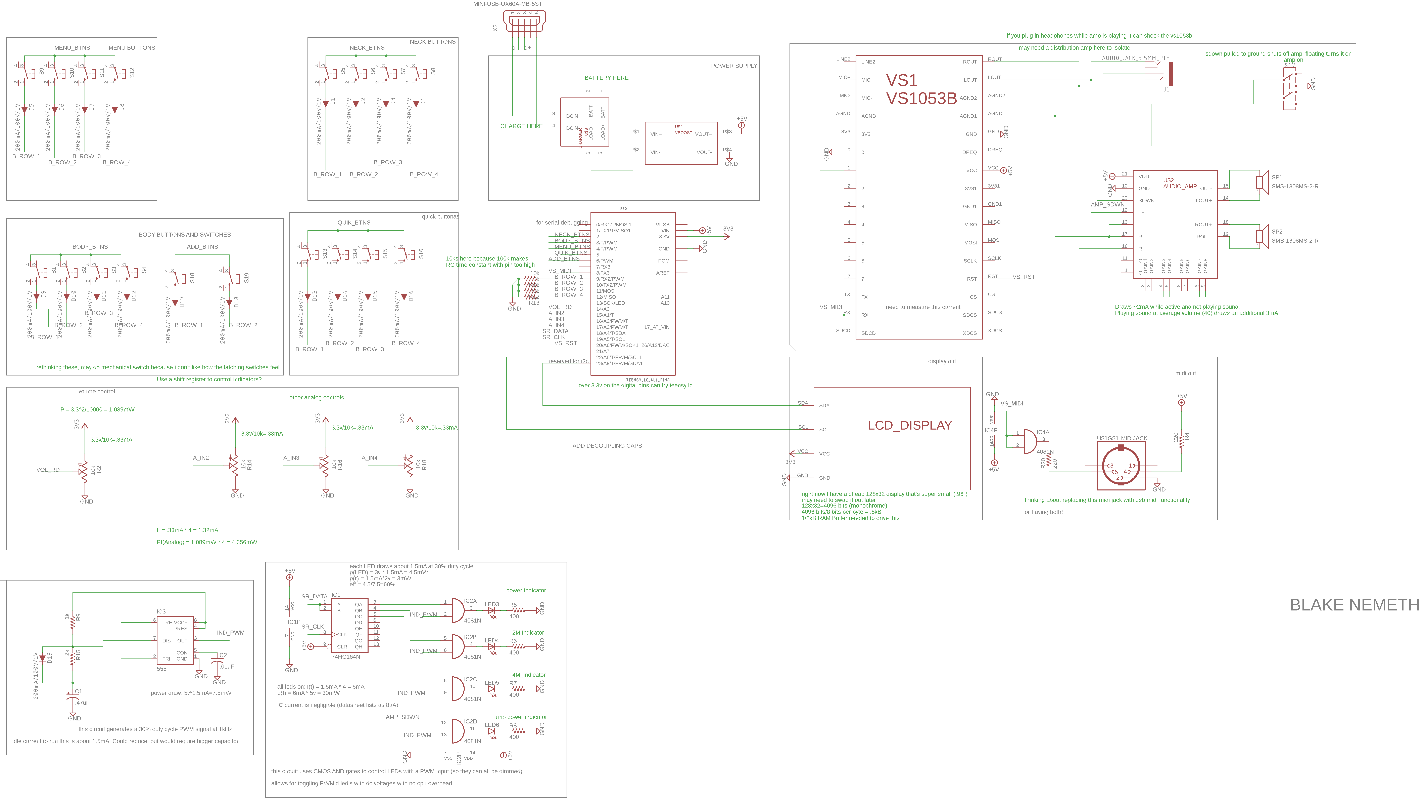
### ScreenSlave:

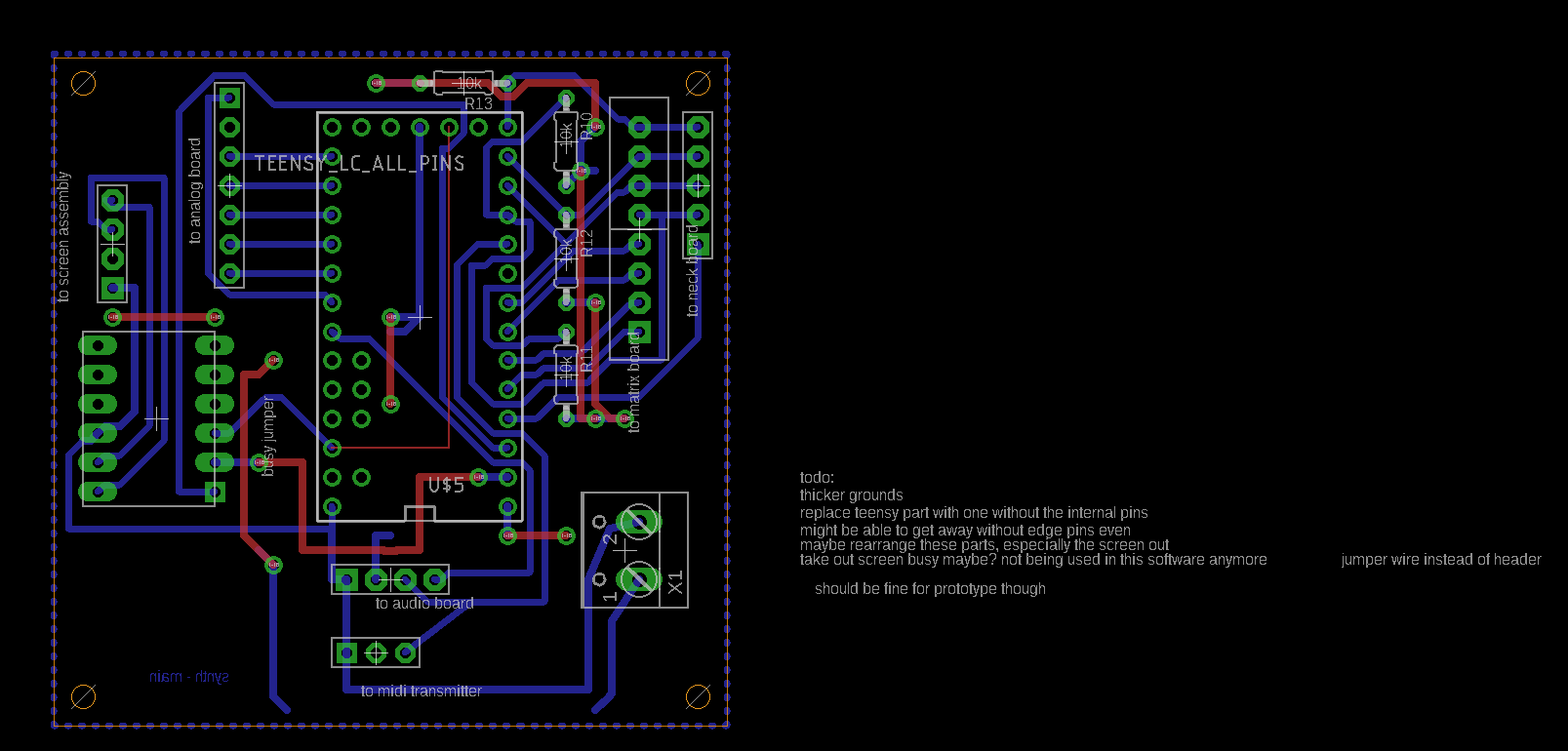
The ScreenSlave Nano constantly checks for data on its serial line, and interprets it as commands/data (depending on previous commands).

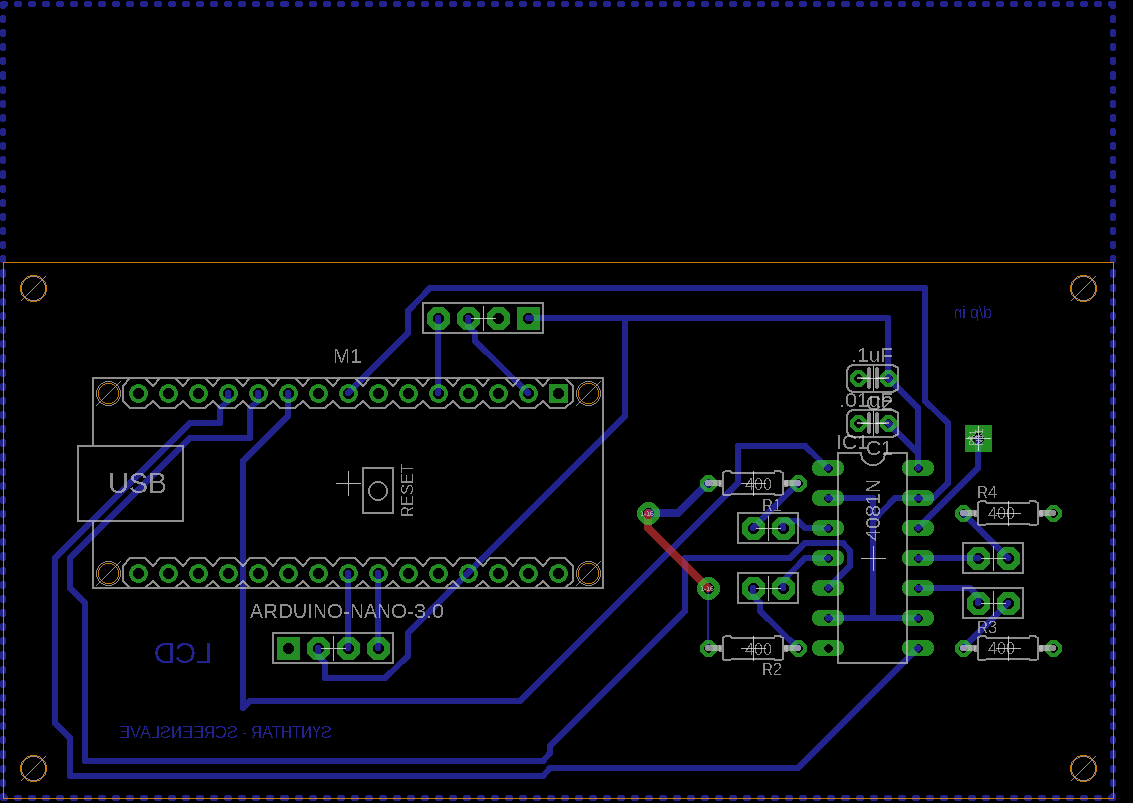
Internally, this is achieved with 4 variables and 1 array: startReceived, currentCommand, receivedBytes, desiredBytes and dataBuffer[]. Inside the main loop is a while loop that only triggers if there’s data in the serial buffer. Before receiving a start command, the screenSlave will constantly check for a start command (the “>” character), and if it receives one, it will set startReceive to a value that will let the loop progress to the next layer (254). Once it’s received a start command, the next byte is the actual command. Each command is assigned a certain number of bytes it requires to run. Every byte until that number is reached gets stored in the dataBuffer. Once the number has been reached, the command is run, and the variables are reset. For more information, check the Github.

A simple test for ScreenSlave functionality is to send a sample screen. To do this, plug it into the PC via the nano’s USB port and connect via the Arduino Serial Monitor, and send “>S” followed by any 5 characters, followed by enter. The screen should display those 5 characters on the top line, and then “string test” on the bottom line.

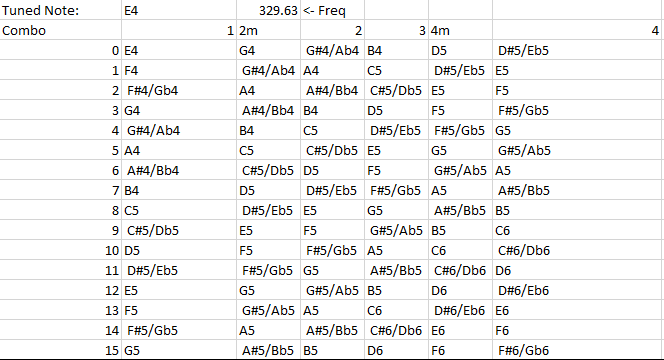
# Appendix A – Overall System Schematic



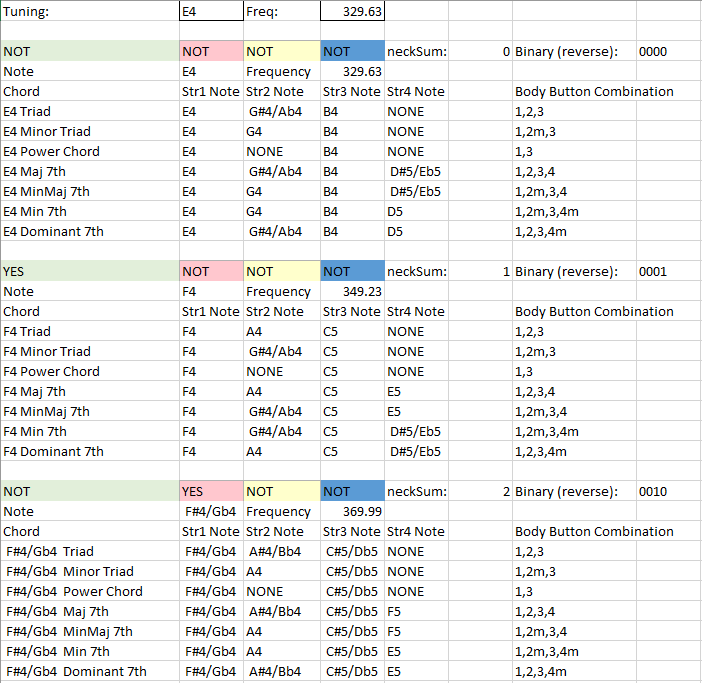
Appendix B – PCB Layout*s**Motherboard*

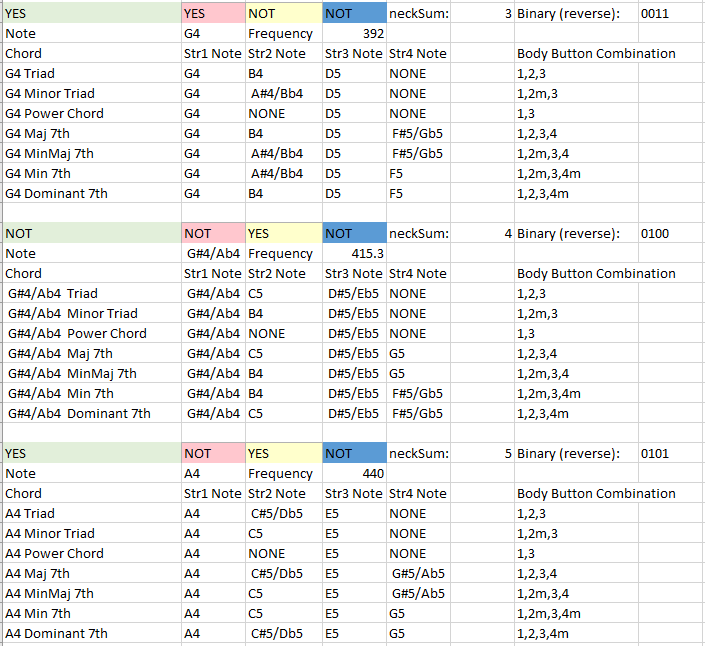
*ScreenSlave*

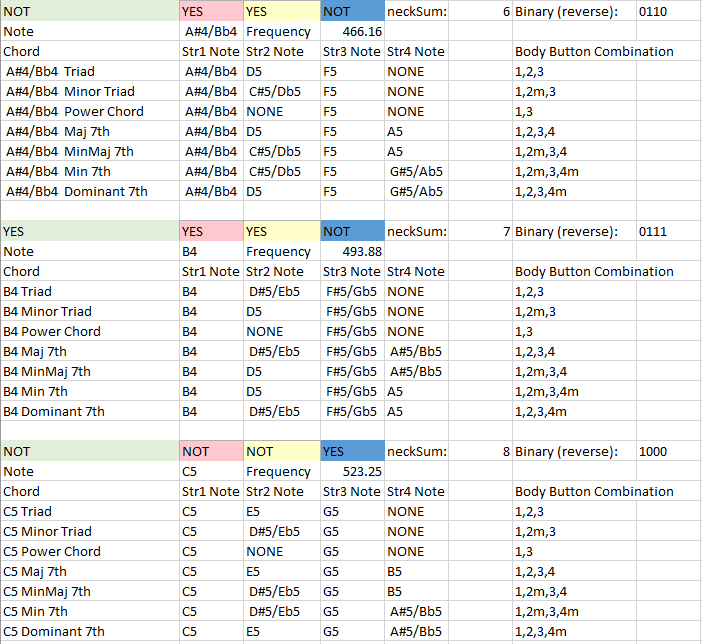
# Appendix C – Playing Chart – Notes (Default Tuning)

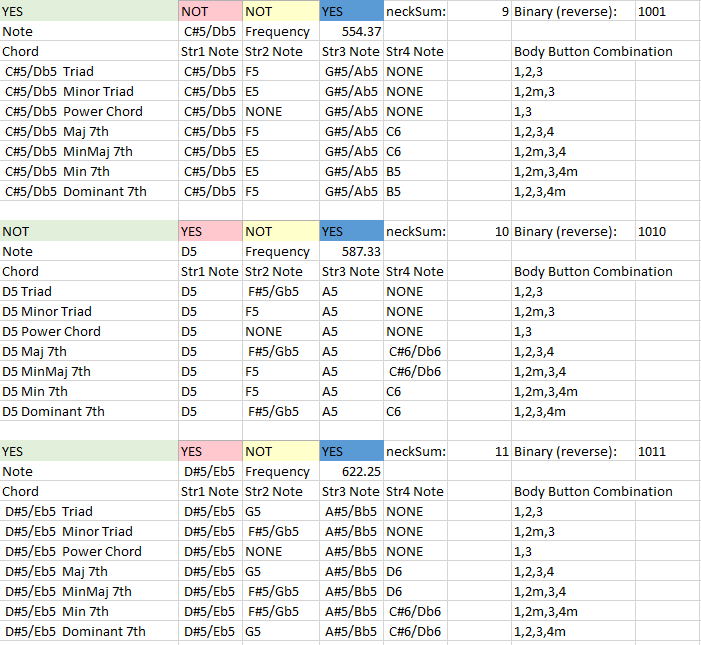


# Appendix D – Playing Chart – Chords (Default Tuning)

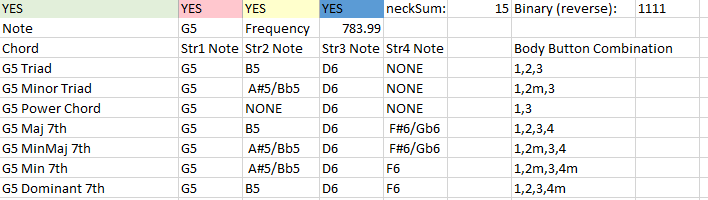












# Bibliography and References –

Teensy LC Datasheet (programming)– <https://www.pjrc.com/teensy/KL26P121M48SF4RM.pdf>

Teensy LC Datasheet (electrical specs)– <https://www.pjrc.com/teensy/KL26P64M48SF5.pdf>

VS1053B Datasheet – <http://www.adafruit.com/datasheets/vs1053.pdf>

VS1053B Midi Example – <https://github.com/adafruit/Adafruit_VS1053_Library/blob/master/examples/player_miditest/player_miditest.ino>

Cherry MX Switch Comparison – <https://www.keyboardco.com/blog/index.php/2012/12/an-introduction-to-cherry-mx-mechanical-switches/>

HD44780 LCD – Library – <https://www.arduino.cc/en/Reference/LiquidCrystal>

HD44780 LCD – Datasheet – <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

74HC164 Datasheet – <https://assets.nexperia.com/documents/data-sheet/74HC_HCT164.pdf>

Arduino Nano Datasheet – <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>

Arduino Nano Low Power Resources – <https://forum.arduino.cc/index.php?topic=418299.0>

MIDI Specifications - <https://www.midi.org/specifications-old/category/reference-tables>

MIDI Standard (Hardware) – <https://www.midi.org/specifications/item/midi-din-electrical-specification>

MIDI Standard (Software) – <https://ccrma.stanford.edu/~craig/articles/linuxmidi/misc/essenmidi.html>