Soliton Multivibe

A Digitally Controlled Analog Vibe

Overview

I have to admit that I love a good vibe pedal. There are many excellent ones out there that do a great job getting those classic tones. However, I had the thought that it would be really cool to have the option of creating more complex movement of the phasing in a vibe pedal and decided that doing digital control would allow for some really fun flexibility.

The digital control is a primary distinguishing feature of the Soliton. An ATTiny85 produces the LFO using the input from a potentiometer. The code has been made flexible enough to allow for a wide range of LFO shapes: sine, triangle, square, rising sawtooth, falling sawtooth, or even arbitrary shapes. The LFO shape is stored in the code as a wave table that is used for creating the LFO output voltage. Digital control also allows for specifying multiple modes and including tap tempo. Further modifications are possible to make it truly unique without changing any components.

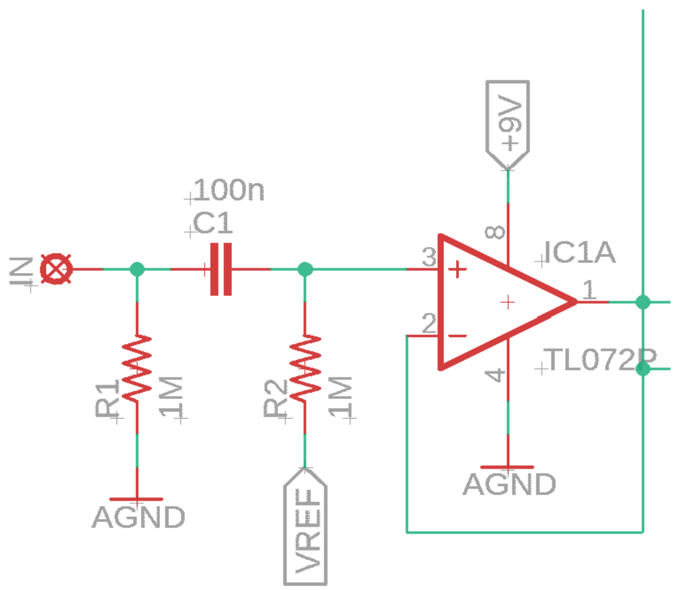
Another difference with the Soliton is that it uses two LED's instead of just one. The LFO rate and depth for each LED are independently controllable, so fascinating rhythm textures can be created unlike any other pedal out there. I did experiment with the number of LED's, but due to the rise and fall times of LFO's and LDR's, going more than two produces extremely quickly diminishing returns. Also, more than two LED's would require a larger chip that is only really feasible in surface mount. I do plenty of projects in SMD, but, when possible, I like my projects to be done through hole to make them accessible to people who can't or don't want to do SMD.

In the end, the result is a fantastic vibe that can get really trippy really fast. The controls offer a ton of flexibility and in my mind, this is another great example of digitally-controlled, analog signal path effects. Now, let's walk through how it works.

How it Works

The core of the Soliton is a four stage vibe circuit extremely similar to the EQD Depths. In fact, I originally thought of just building a clone, but I can't leave well enough alone. I also love microcontrollers for LFO's because of all the flexibility available from it. You can have waveforms that an analog LFO simply can't do very well, like a real sine wave. There are some other cool tricks I've done, so read on.

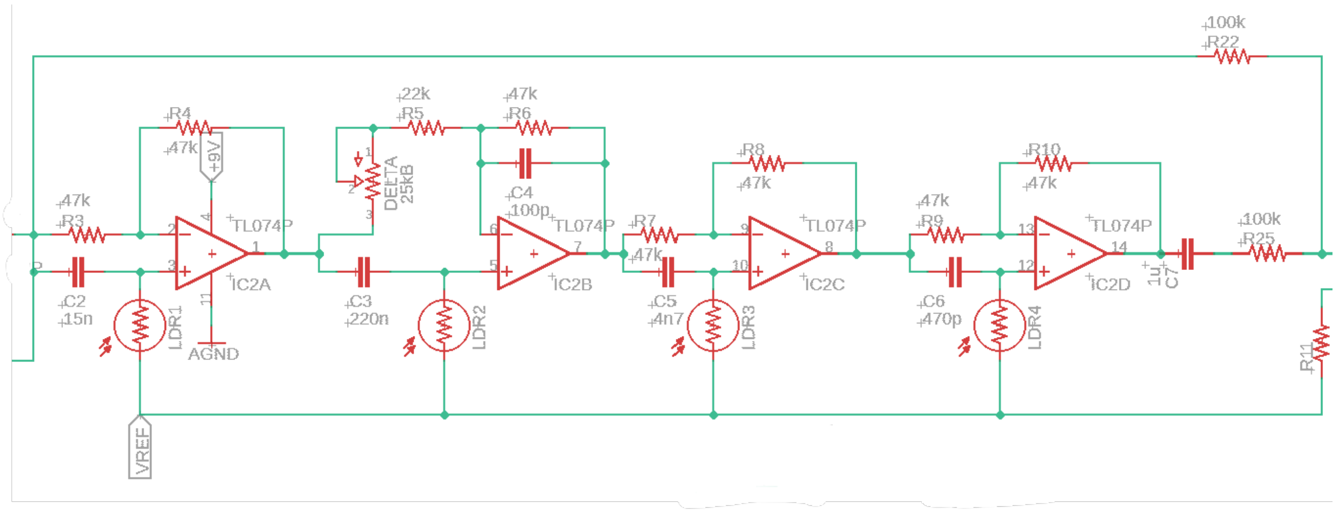
First off we have an input buffer that gives us a high input impedance and allows us to drive the delay stages with a low output impedance. This is a pretty straightforward implementation using one half of a TL072.



Soliton Input Buffer

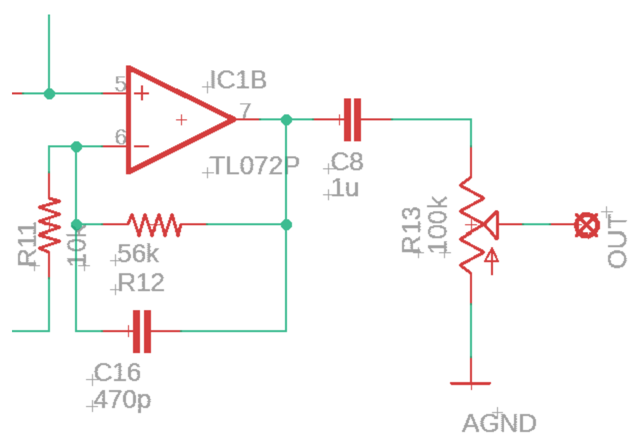
The buffered input signal is then fed into the phasing stage. This stage consists of four all pass filters with the phase relationships set by the input capacitors and the variable resistance from Vref to the inverting opamp inputs. The “Throb” control actually helps set some skew in the phasing of the second stage.

The buffered input is sent through a 100k resistor that feeds into the summing output mixer/buffer. This is rather common of any dry/wet circuit.



Soliton Phase Stages

The phased output is combined with the buffered input using an output buffer/summing amplifier. There is a small capacitor in the feedback loop to allow for some filtering. Without this, the vibe circuit can get extremely bright, even harsh. Adjusting the value of this cap will allow for the user to set the response according to preference. I liked a 470 pF cap here. Also, the output has a trimmer that is set to achieve unity gain when the pedal is engaged. I used some set values on my first board revision, but it seems tolerances can make it a little variable.



Soliton Output Buffer

The LFO architecture is what sets the Soliton apart. Using an ATTiny85, there are controls for rate, tap tempo, mode, and then the LFO outputs. The LFO's are actually using lookup tables stored in flash on the ATTiny, which allows for use of any wave shape desired. I coded up tables for sine, square, triangle, rising sawtooth, and falling sawtooth. They are included in comments in the code, just copy the table you want and put it into the entries for the wave tables.

The rate controls span a range of .25 to 5 Hz, which is a factor of 20. Because I wanted really good rates available, I found that instead of using just the analog level in with a linear or log potentiometer, a better option is to use the analog level in of a linear potentiometer with a custom taper lookup table.

This lookup table is set up to give a true logarithmic spacing of time (whereas a log taper pot is really two different linear sections). This allows for a very natural feeling distribution of available LFO rates.

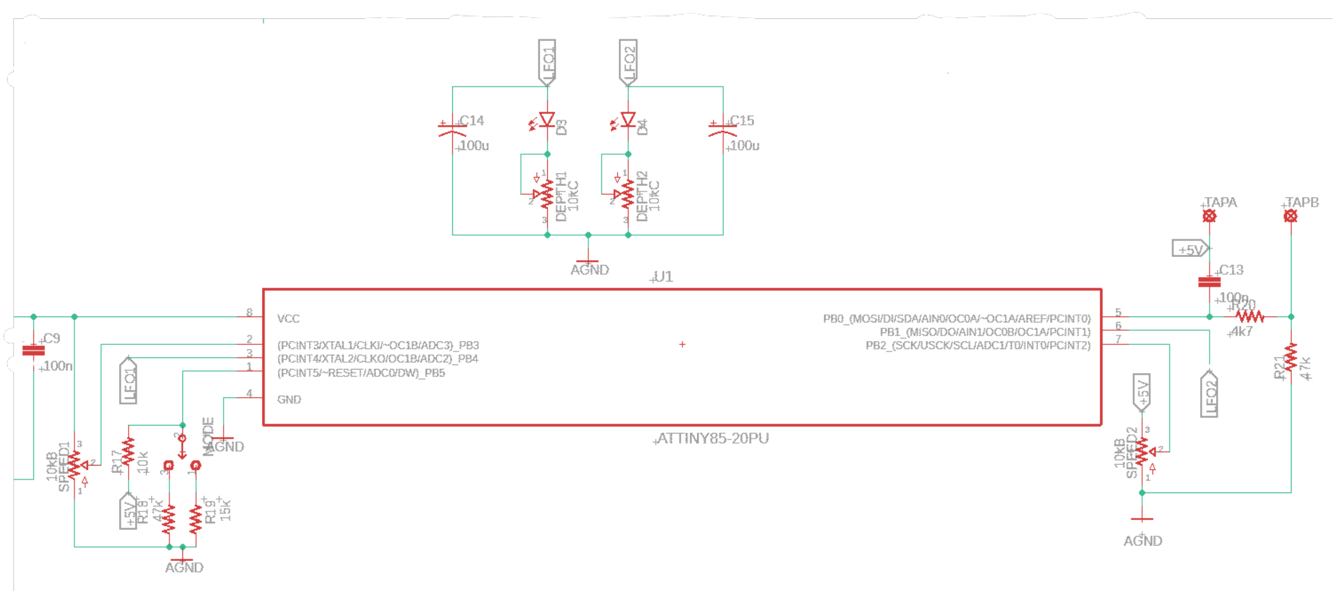
The mode switch has three settings, which in the default code correspond to:

- Dual Independent LED's (I labeled it as Σ, which is the mathematical symbol for summation)
- Single LED where the LFO is the product of LFO1 and LFO 2 (labeled as Π, the mathematical symbol for product, or multiplication)
- Dual LED's where the rate of LFO 2 is the golden ratio of LFO 1 (labeled as φ, the symbol used for the golden ratio)

Dual independent LED's is the default mode and allows for mixing of rate and amplitude to achieve cool effects. The single LED product mode results in a seemingly random LFO, depending on the settings of the two wave shapes and speeds. The golden ratio mode results in a wonderful sounding pulsing mode that is not totally predictable sounding, but is very pleasant.

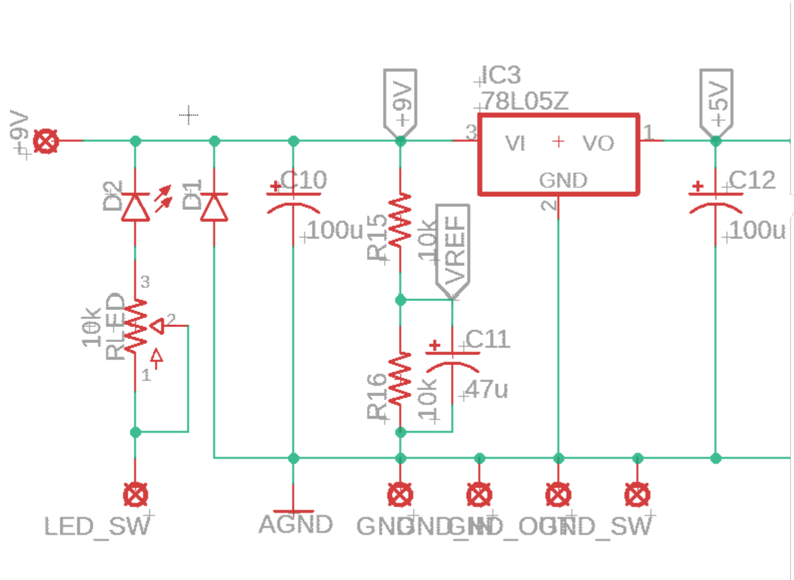
The depth controls are actually controlling the brightness of the LED's by being variable current limiting resistors. This was chosen for two reasons. First, it allowed me to use fewer microcontroller pins. Second, it allowed me to keep all the precision of the PWM output so that there weren't quantization errors. The LED/Depth Control is bypassed with a large smoothing cap to get rid of high frequency PWM noise. I tried several values here and found that 100 uF was as small as I could go while still getting nice, smooth LFO signals to the LED's.

Tap tempo is implemented the same was as in the EchoWreck. The switch pulls the input pin high, which triggers an interrupt to start the tap tempo process. This allows for more accurate tap tempo, since it doesn't have to wait every polling period. It also makes other processes faster, since it's not consuming cycles just polling the switch.



Soliton LFO and Digital

The final block is the power section. This is very basic, with 9V coming in and a 4.5V reference voltage for the opamps and 5V for the ATTiny85. Note that decoupling caps are local to each IC.



Soliton Power Section

BOM

The BOM below is the list of parts I used for mine along with quantities. One of my goals was to make a cool vibe that didn't use esoteric parts. Note that the LDR's used are the ones that Tayda carries. SmallBear also carries them, as do some other suppliers.

Part	Qty.	Notes
4k7 Resistor	1	
10k Resistor	4	
15k Resistor	1	
22k Resistor	1	
47k Resistor	9	
56k Resistor	1	
100k Resistor	2	
1M Resistor	2	
100pF Capacitor	1	
470pF Capacitor	2	
4.7nF Capacitor	1	
15nF Capacitor	1	
100nF Capacitor	3	

220nF Capacitor	1	
1uF Film Capacitor	2	
47uF Electrolytic Capacitor	1	
100uF Electrolytic Capacitor	4	
B10k Potentiometer	2	16mm PCB Mount
B25k Potentiometer	1	16mm PCB Mount
C10k Potentiometer	2	16mm PCB mount
SPDT On/Off/On Toggle	1	
1N4148/1N5817/1N4001	1	Voltage polarity protection
KE-10720 LDR	4	
Red LED	2	Needs to be red for best performance, Pads spaced for 3mm LED
LED	1	Status LED
78L05	1	
TL072	1	
TL074	1	
ATTiny85	1	
Enclosure	1	125B
1/4” input jack	2	
DC power jack	1	
3PDT footswitch	1	
SPST Momentary (NO) footswitch	1	

Schematic

The schematic for this project is a little too big to be usable in this size. A PDF of the full schematic is included in the project build files.

Build Notes

Here are some things I noted from building the Soliton rev 1.0 PCB. Please read this section to make sure you don't go through excessive frustration.

Enclosure Size/Drilling

This project is made for a 125B with top mount jacks. It fits relatively easily, though there isn't a ton of wiggle room for drilling for the footswitches. Just be careful and you'll be fine. Drilling for the power and in/out jacks requires some precision, but there are a few mm to play with.

Jacks

There shouldn't be any real restriction here. Use whatever you use for a 125B with top mount jacks. I use the box jacks for most of mine.

Changing Wave Tables

The wave tables in the code are labeled as waveTable1 and waveTable2 for LFO1 and LFO2, respectively. There is also a section where the different wave shapes are commented out. If you want to change the wave shape for a particular wave table, just copy and paste the table in. DO NOT just rename the table, as you will soon forget what it is and get rid of it at some point and you will hate your life. Ask me how I know...

Changing Modes

You can change the behavior of the mode switch to do something else if you like. For example, it could be tied to which wave shape to use for a wave table, or a rate multiplier. Just code in the behavior you want in the checkMode() function.

Future Plans

So I have plans for a deluxe version of the Soliton, which will be in stereo and use rotary switches for each LFO's wave shape. There are a couple other things I'm investigating for it, so stay tuned.

In Closing

This is a fun vibe. If you want it to be a straight up single vibe, it can be. If you want to get trippy with the LFO's and the LED interactions, go for it. Whatever you do, have fun with it!