

Multi-Window Comparator documentation.

Written November 15, 2012

Last edited November 15, 2012

I. What is a Multi-Window Comparator?

- A. A "regular" window comparator is this.*
- B. A Multi-Window Comparator is this.*
- C. Inputs/Outputs/Controls*
- D. Patch ideas.*

II. Circuit Description/Schematics

- A. The Chip*
- B. Analog (CV) input.*
- C. Digital inputs*
- D. Digital (Gate) outputs*
- E. Power Supply*

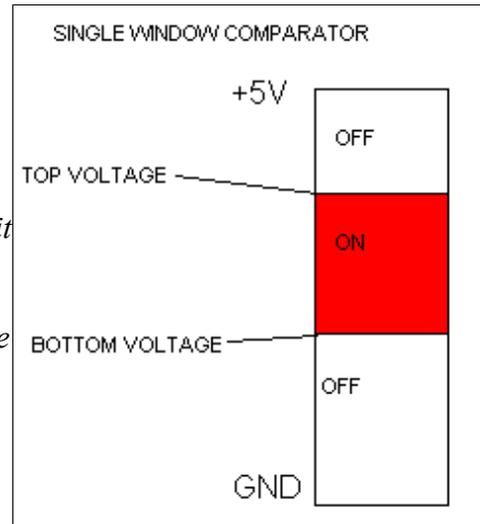
III. Construction of the Circuit.

- A. Parts List*
- B. The PCB*
- C. Wiring Diagrams*

I. What Is A Multi-Window Comparator?

A. A "regular" window comparator is this.

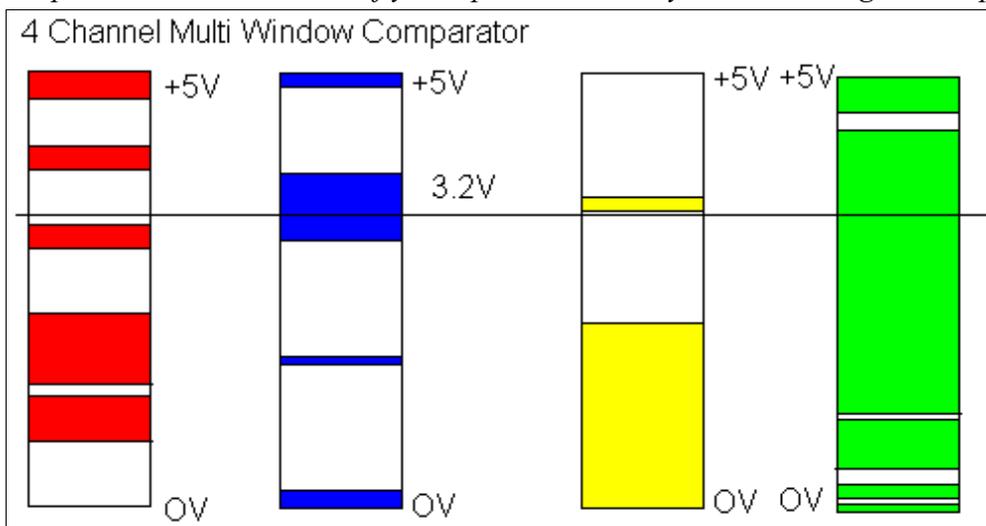
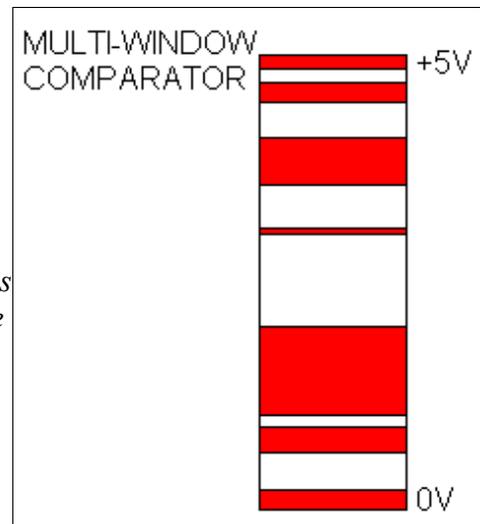
A window comparator is a circuit with three inputs and 1 output. The first two inputs are an upper voltage threshold and a lower voltage threshold. If the third input is inbetween these two thresholds, it turns the output on, and if it is not, the output turns off. There are many variations on this basic circuit as you can see from [a quick google image search](#). Usually this circuit is created with op-amps, and some signal conditioning.



B. A Multi-Window comparator is this.

A Multi-Window comparator, like the "regular" window comparator has an input voltage and an output that toggles on or off depending on the input voltage. The difference is that instead of comparing the voltage to two other voltages, instead the voltage is converted to a number and that number is looked up on a data table which says to either turn on or turn off the output at that voltage. This gives the user the chance to create lots of windows, so that multiple voltages and ranges of voltage can trigger the same output.

This project is 4 multi-window comparators which share a single input voltage, each comparator has it's own look-up table. So, if you inputted 5V, the red, blue and green outputs would all turn on. If you inputted 0v, the blue, yellow and green outputs would turn on, and if you inputted 3.2V only the blue and green outputs would turn on.



Each table is composed of 256 steps that can be turned off or on. If you were turning every other step off, this would give a maximum number of windows at 128.

C. Inputs/Outputs/Controls

INPUTS:

1. Voltage input - This is the voltage that is telling the microcontroller where to look on the data tables. This input corresponds to all four outputs.

2.External Write/Erase inputs (optional)- A trigger or gate signal can be inputted to the module to perform write/erase commands to the tables. You would need these inputs duplicated for each channel

OUTPUTS:

1.Channel Outputs - There are four of these. These output +5V whenever the input voltage is in an "on" window for that outputs look-up table.

CONTROLS:

1.Input Attenuator - This attenuates the input signal. The microcontroller only recognizes voltages between 0 and 5V, so this reduces large signals so that the full signal can be read by the microcontroller.

2.Input Offset - This adds a voltage offset of between 0 and 5v to the input signal. When using an input signal that may be negative part of the time (like an LFO) this can bring the negative parts of the signal into the proper range for the micro controller's input. This can also be used as a manual control voltage when not using an input signal.

3.Write/Erase Buttons - These should be non-latching push buttons, there is a write button and an erase button for each of the 4 channels. When the write button is engaged, the data table will be told to turn on at the current input voltage. When the erase button is engaged, the data table will be told to turn off at the current input voltage.

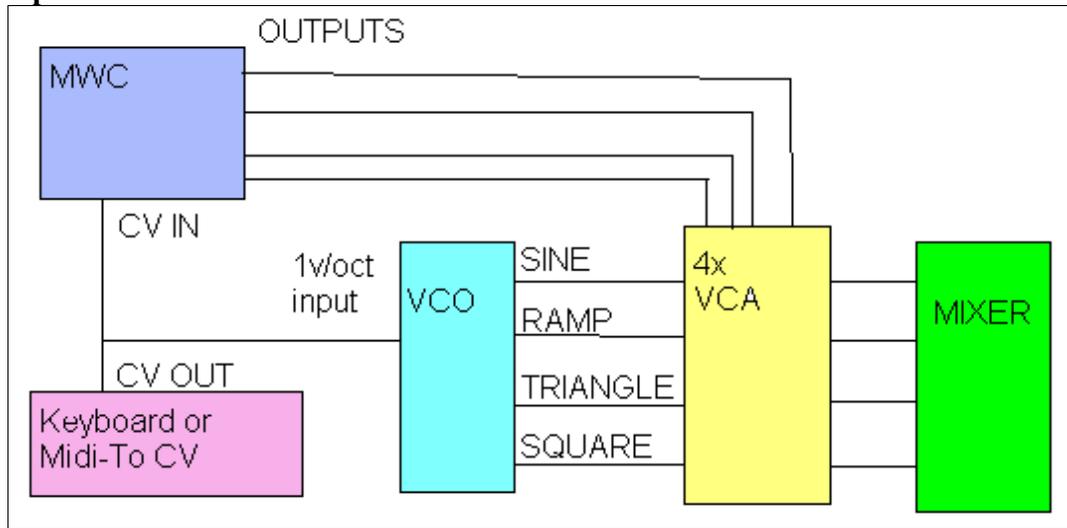
4.Bank Select switch - This should be a SPDT center off type toggle switch. There are two databanks for saving your lookup tables. When this switch is turned either up or down it tells the microcon to save or load from the up or down bank. When the switch is in center position, it disables saving or loading from EEPROM. You should leave the switch in the center position normally, so you don't accidentally save or load a table.

5.Save/Load Buttons - These should be non-latching push buttons. The save button will save the current 4 data tables to the selected memory bank when pressed. The load button will load the 4 data tables in the selected memory bank and begin using them. When an erase button is held down on a channel and the load button is pressed at the same time, it will erase all "on" windows in that channel.

D. Patch Ideas

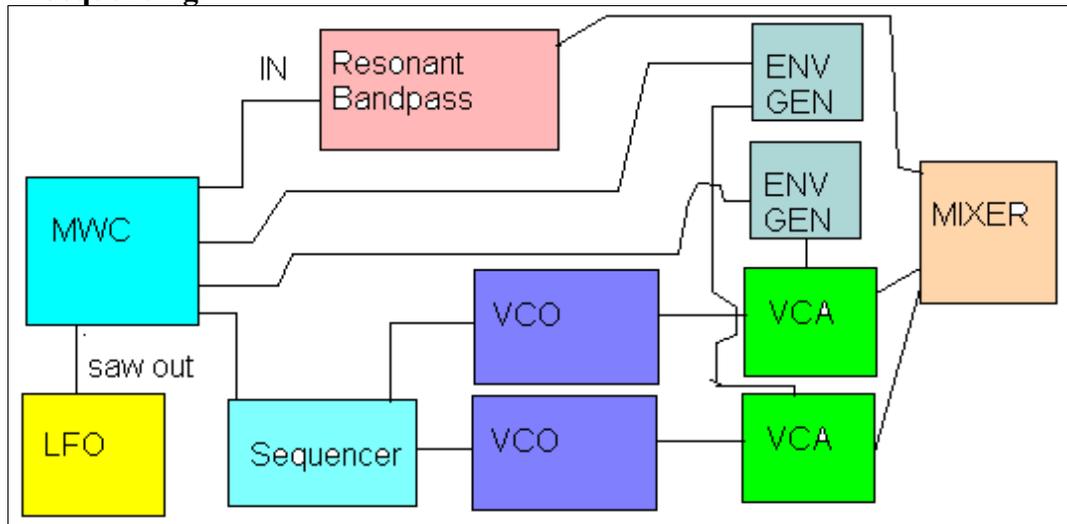
These are just a couple of simple ideas. There are plenty of uses for the MWC beyond these.

1. Pitch dependent wave forms.



This patch would switch between or mix together the outputs of a VCO depending on what control voltage is being played.

2. Rhythm Sequencing.



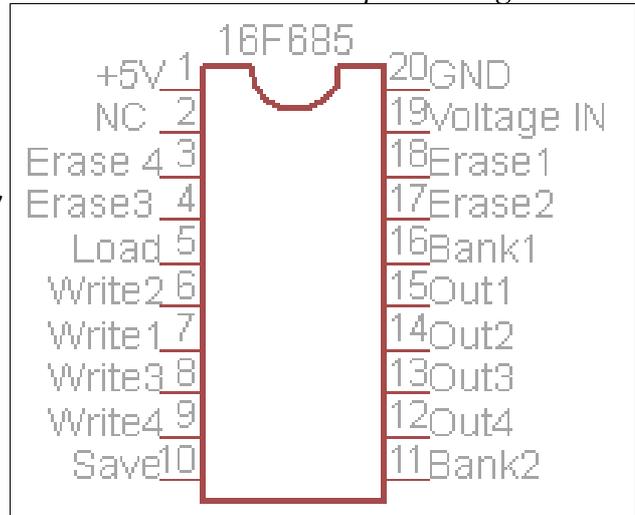
The saw output of an LFO is fed into the input of the MWC. The first output is sent to the audio input of a resonant bandpass filter set to a low frequency, this will create percussive sounds. The next two outputs are sent to the inputs of two AR generators. The final output of the MWC is being sent to the clock input of a sequencer. The sequencer's CV output is being sent to a pair of VCOs, the outputs of these VCOs are being sent into two VCAs which are being controlled by the AR generators. The VCAs and the bandpass filter's outputs are all mixed together.

II. Circuit Description/Schematics.

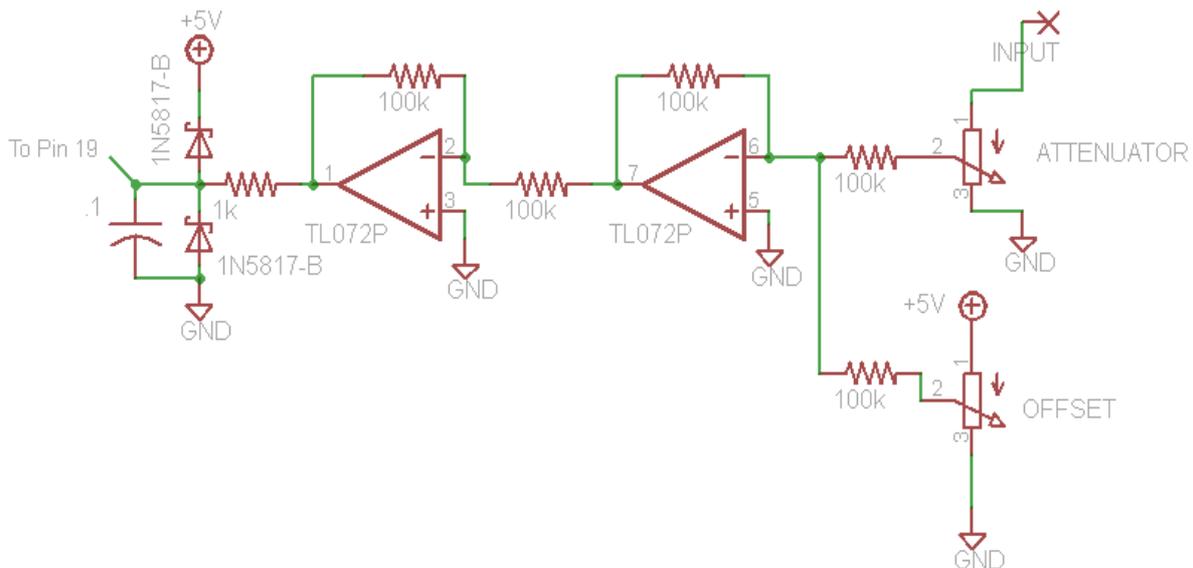
A.The Chip

The heart of this circuit is the PIC 16F685 microcontroller. Here is the pinout diagram:

Pins 1 and 20 are part of the power supply. Pin 19 is the voltage input, this is considered an "analog input," a voltage anywhere between 0 and 5V is appropriate. Pins 12-15 are the gate outputs for the four channels, these alternate between 0 and +5V. All other pins are considered "digital inputs" these should only have 0 or +5V on them.

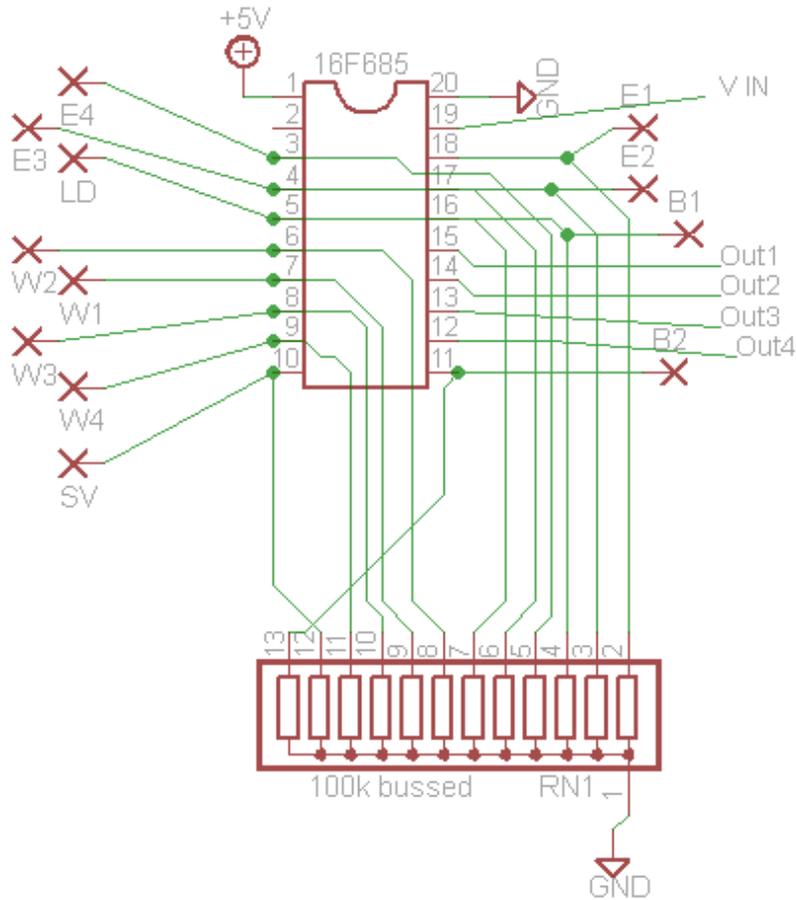


B.Analog (CV) Input

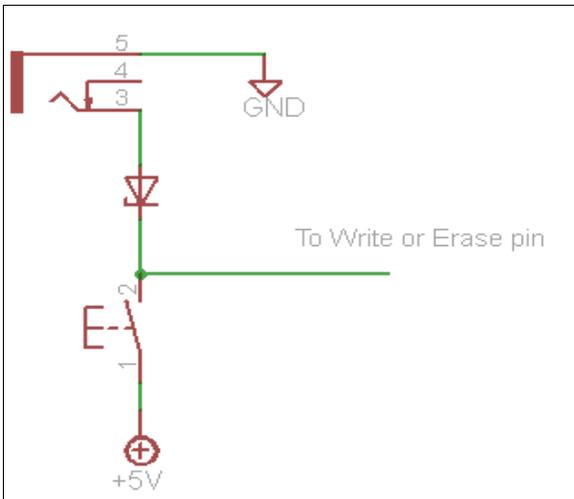


This circuit conditions an external Control Voltage so it can be read by the microcontroller. The pot in the top right corner attenuates the input signal. The pot on the bottom right is a voltage divider creating a voltage between 0 and 5v. The 100k resistors and two op amps mix these signals together. The 1k resistor, and 2 Schottky type diodes prevent negative voltages and voltages over 5V from getting to the microcontroller. The .1uF cap filters high frequency noise out of the signal.

C. Digital Inputs

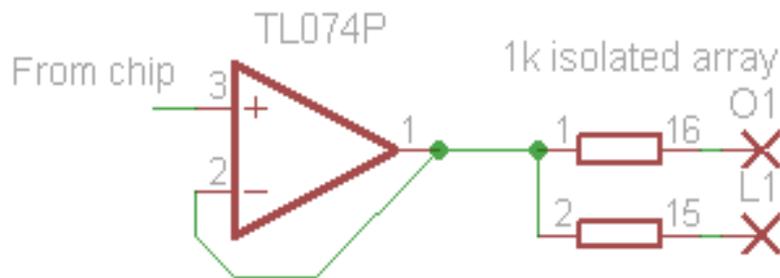


Here we see that all digital inputs are sent to a 100k 13pin bussed resistor array. These will keep all the digital input pins at 0V until a +5V signal comes from a switch or jack.



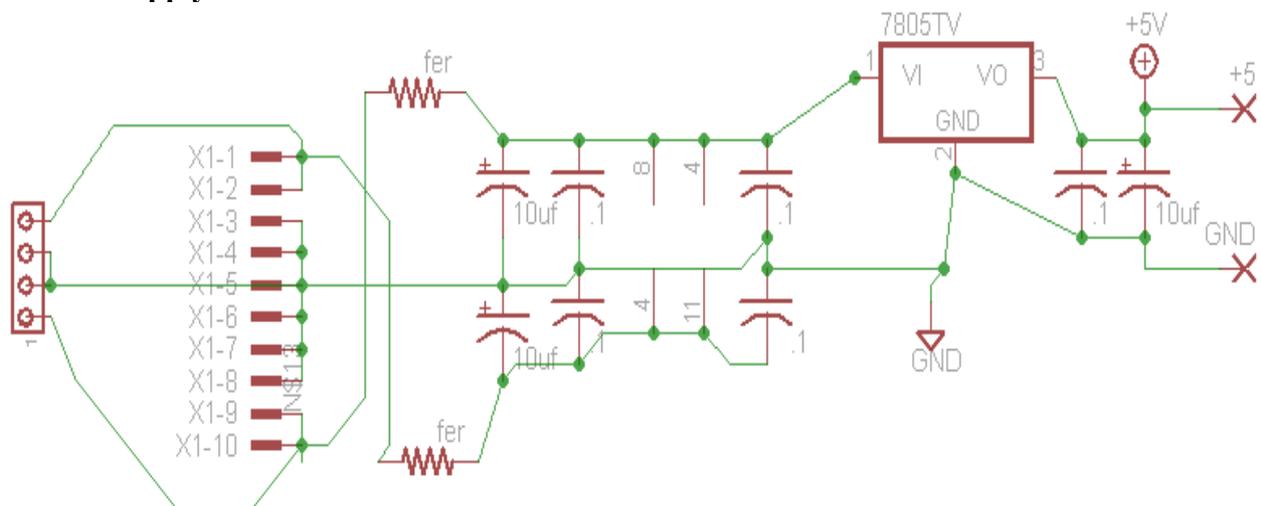
Normally all digital inputs are connected to just a momentary switch that connects them to +5V. If you want to use an external gate or trigger to cause write/erase actions, you should use a diode between the tip of the jack and the switch, like in this schematic in order to protect the output circuitry of the module providing the gate or trigger signal.

D. Digital (Gate) Outputs



This circuit is repeated for each of the 4 outputs. The output pin from the microcontroller is sent to an op-amp buffer. The output of the buffer goes through a 1k resistor (located on a DIP array, for less lead bending when building) before being connected to the output jack. There is another 1k resistor that is attached to the anode of an LED to indicate when the output is turned on.

E. Power Supply



Here we see the two power connectors. One is a 4 pin connector for MOTM synth systems, the other is a 10 pin connector for Euro Rack systems. The two resistors marked "fer" are for ferrite beads or low value resistors (10 ohms) this reduces noise in the power supply. Next we see 10uF caps filtering the positive and negative power supply rails, and we see .1uF caps providing further filtering (these caps are placed near the ICs power supply pins). The positive rail then feeds a 7805 power regulator which provides +5V, this is filtered by a 10 uF capacitor and a .1uF cap. Finally we see "+5" and "GND" wiring pads, providing access to these voltages for our off panel wiring.

III. Construction

A. Parts List.

1.Semiconductors

<i>VALUE</i>	<i>Quantity</i>	<i>NOTES</i>
16F685	1	Should come with your PCB when you order
TL072	1	Or any other dual op-amp on hand.
TL074	1	Or any other quad op-amp on hand.
7805	1	In TO-220 package
SD101C	2	Schottky Diode

2.Capacitors

<i>Value</i>	<i>Quantity</i>	<i>Notes</i>
.1 μ f	6	5mm lead spacing. Cheap ceramics should be fine
10 μ f @ 16V or higher	3	2.54mm lead spacing.

3.Resistors

<i>Value</i>	<i>Quantity</i>	<i>Notes</i>
100k 1/4W	5	5mm lead spacing. Get 3.5mm length resistors, or stand resistors up
1k 1/4W	1	" "
13 pin 100k bussed array	1	100k value non-critical. Anything from 10K to 1M should work
16pin 1k isolated array	1	Or 16 1k resistors.
B100K PC mount pot	2	Alpha RV120F-20-15F-B100K part.
10 ohm	2	or ferrite beads. 10mm lead spacing

4.Connectors

<i>Value</i>	<i>Quantity</i>	<i>Notes</i>
20 pin DIP Socket	1	
14 pin DIP Socket	1	
8 pin DIP Socket	1	
Power Connector	1	Either 10 pin or 4 pin depending on your system.

5.Off Board

<i>Value</i>	<i>Quantity</i>	<i>Notes</i>
Jacks (1/4" or 3.5mm or banana)	5 or 13	Quantity depends on whether wanting to use external triggers for write/erase.
Pushbutton switches	10	Momentary SPSTs.
SPDT Toggle	1	Center off
LEDs	4	

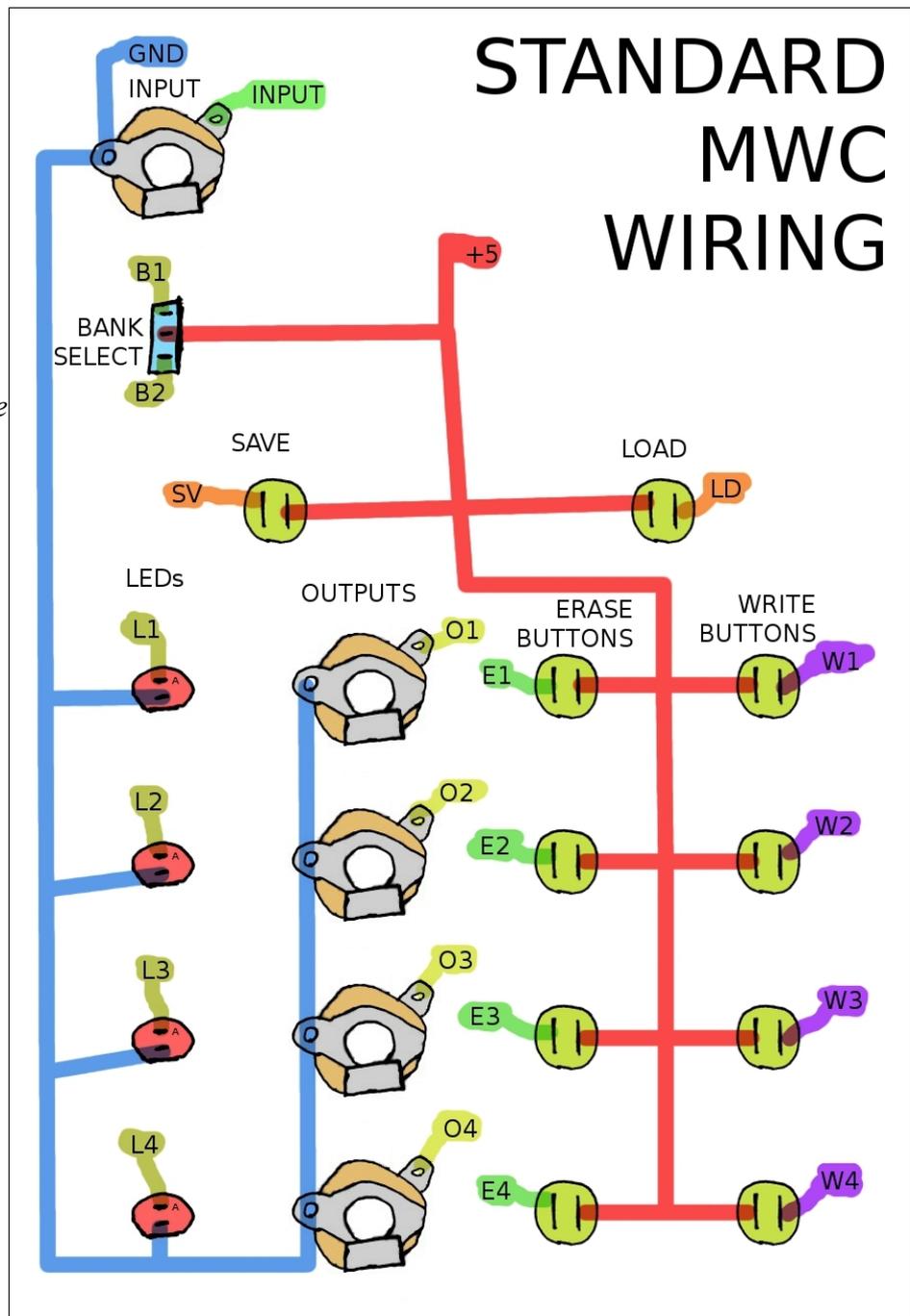
Wiring Option #1.

This is how I wired my initial prototype. The colored text indicates which wiring pad on the PCB that wire should be connected to.

Because the Pots are PC mounted, I did not include them in this wiring diagram.

For connecting the +5V supply to the pushbuttons, I suggest using cut off resistor leads left over from stuffing your PCB.

You can also solder the cathodes of your PCBs directly to the sleeve pins of your output jacks and thus use the output jack to stabilize the LED instead of buying bezels or snap-in LEDs.



Wiring Option #2

This includes the optional input jacks for write/erase jacks.

