

Delays		
<b>verbolay</b> (a blend of delay and reverb plus filtered feedback)	pot0 : delay pot1 : reverb pot2 : tone fdbk : feedback	pot1 goes from fully delay ccw to fully reverb cw
<b>pitch verbolay</b> (a blend of delay and reverb plus endless detuned ascending or descending feedback)	pot0 : delay pot1 : reverb pot2 : pitch fdbk : feedback	pot1 goes from fully delay ccw to fully reverb cw pitch goes down ccw and up cw
<b>particolay</b> (delay with feedback chopped up in grains)	pot0 : particle rate pot1 : delay pot2 : envelope sensitivity fdbk : feedback	when you increase envelope sensitivity, the harder you play the more the rate is going to slow down
<b>ringolay</b> (delay with feedback going through a ring modulator)	pot0 : ring mod blend pot1 : ring mod rate pot2 : delay fdbk : feedback	
<b>pitcholay</b> (delay with feedback going through a pitch shifter)	pot0 : pitch pot1 : delay pot2 : tone fdbk : feedback	
<b>aliaserolay</b> (delay with feedback going through an aliaser)	pot0 : aliaser pot1 : delay pot2 : aliaser blend fdbk : feedback	
<b>analog(-ey) mod delay</b> (analog sounding delay with filtered and modulated feedback)	pot0 : delay pot1 : mod rate pot2 : mod width fdbk : feedback	
<b>square pitcher</b> (delay with repeats going thru a pitch shifter that is modulated with a square wave LFO)	pot0 : LFO depth pot1 : LFO rate pot2 : delay time fdbk : feedback	

Loopy, samplly, freezy stuff		
<p><b>hold</b> (hold a bit of audio indefinitely, well almost)</p>	<p>pot0 : hold (function switch) pot1 : filter pot2 : momentary/latching fdbk : nothing smplr: change pitch</p>	<p>leave the pot0 below noon, it has no function in this patch pot2 below noon: function switch = momentary pot2 above noon: function switch = latching</p>
<p><b>tape loop</b> (a loop that can be slowed down or sped up, the length of the loop can be shortened to tiny bits, can feedback into itself)</p>	<p>pot0 : loop pot1 : tape speed pot2 : tape length fdbk : feedback smplr: change pitch</p>	<p><b>loop knob :</b> left half = continuously recording audio input and playing it back according to tape speed and length right half = stops recording input and loops the last recorded sample infinitely <b>tape length :</b> reducing it to the smallest settings can create glitchy artifacts, flanger-like sounds and other weird stuff, is interactive with tape speed</p>
<p><b>envolooop vibrato</b> (endless repeating delay loops are envelope triggered and run through a vibrato)</p>	<p>pot0 : sensitivity pot1 : delay time pot2 : vibrato rate fdbk : decay</p>	<p><b>sensitivity:</b> determines how much signal is added into the loop and fades a portion of the previous loop out. when set to minimum the signal is looped infinitely. you can use the function switch to control this. <b>delay time:</b> defines the length of the looped sample. On very short settings, the loop will fade out faster. <b>vibrato rate:</b> applies vibrato to the looped samples <b>decay:</b> fully CW the lamples can loop infinitely (well almost). As you dial it back the loop will start to fade out.</p>
<p><b>envolooop reverb</b> (endless repeating delay loops are envelope triggered and run through a reverb)</p>	<p>pot0 : sensitivity pot1 : delay time pot2 : reverb mix fdbk : decay</p>	<p><b>sensitivity:</b> determines how much signal is added into the loop and fades a portion of the previous loop out. when set to minimum the signal is looped infinitely. you can use the function switch to control this. <b>delay time:</b> defines the length of the looped sample. On very short settings, the loop will fade out faster. <b>reverb mix:</b> set the amount of reverb that is mixed into thwe looped signal <b>decay:</b> fully CW the lamples can loop infinitely (well almost). As you dial it back the loop will start to fade out</p>

Glitchy, noisy, random stuff		
<b>glitcholay</b> (sample size randomly changing plus lots of interdependent randomness, can feedback on itself)	pot0 : rate 1 pot1 : sample size pot2 : randomness fdbk : feedback	pot0 controls how often the signal will begin to loop. pot1 defines the length of the samples being looped. pot2 will gradually introduce randomness to the rate and sample size as you turn CW.
<b>pitch-glitcholay</b> (similar to glitcholay but you can octave up the feedback)	pot0 : feedback pot1 : rate pot2 : octave blend fdbk : nothing	pot0 feeds the signal back to itself in random bursts pot1 defines how often the sample size changes pot2 adds an octave up to the feedback I like to leave the param_1 mode on either pot 1 or exp 1 and occasionally step on the tap switch for some extreme glitchiness :-)
<b>pitch step glider</b> (sample&hold modulated pitch shifter with discreet steps or glissando, can feedback on itself)	pot0 : glide pot1 : rate pot2 : pitch depth fdbk : feedback	<b>glide:</b> CCW will be clean random steps, as you turn CW the transitions from one step to the other will gradually glide more, until no more steps are audible but a wobbly random LFO
<b>pitch square lfo</b> (pitch shifter modulated by a square LFO whose rate can be randomized, can feedback on itself)	pot0 : randomness pot1 : rate pot2 : pitch depth fdbk : feedback	
Modulation		
<b>flanger barberpole TZF ring</b> (an endlessly descending or ascending flanger, capable of going through zero, feedback running through a ring modulator)	pot0 : rate/direction pot1 : through-zeroneess pot2 : ringmod mix/rate fdbk : feedback	<b>pot0</b> sets the direction of the barberpole sweep and its rate. Right in the middle there is no sweep. As you turn CCW the sweep gets faster and downward. As you turn CW the sweep gets faster and upward. <b>pot1</b> flanges closer to zero as you turn CW <b>pot2</b> introduces ringmodulation (to the feedback only) careful on the amount of feedback on some settings :-) It's set to work best with wet only, but there i no harm in adding dry if you like it
<b>granpa's vinyl</b> (add vinyl crackle, noise, pitch warp and frequency bandwidth loss to your sound)	pot0 : rate pot1 : pitch warp pot2 : filter fdbk : noise & crackle	
<b>dynamic ringulator</b> (ring mod with blendable sample&hold and envelope controlled rate)	pot0 : s&h blend pot1 : ring mod rate pot2 : s&h rate fdbk : envelope sensitivity	
<b>dynamic vibrato</b> (vibrato whose rate can be envelope controlled, also does a convincing rotary speaker thing if blended with dry signal)	pot0 : randomness pot1 : rate pot2 : depth fdbk : envelope sensitivity	<b>randomness</b> goes from sine modulation ccw to random steps cw <b>envelope sensitivity</b> sets how much the guitar signal will affect the rate
<b>dynamic phaser</b> (phaser whose rate can be envelope controlled)	pot0 : depth pot1 : rate pot2 : regeneration fdbk : envelope sensitivity	<b>envelope sensitivity</b> sets how much the guitar signal will affect the rate It's set to work best with wet only, but there i no harm in adding wet if you like it

Filters		
<b>envelope filter and optional bit crusher</b> (vocal sounding envelope controlled filter, running through a bit crusher)	pot0 : bit crusher blend pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	<b>filter range:</b> CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds
<b>envelope filter aliased</b> (vocal sounding envelope controlled filter running through an aliaser)	pot0 : aliasing pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	<b>filter range:</b> CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds
<b>envelope filter ring modulated</b> (vocal sounding envelope controlled filter, running through a ring modulator)	pot0 : ring mod rate pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	<b>filter range:</b> CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds <b>envelope sensitivity</b> controls the filter frequency but also the ring mod rate
<b>lfo filter</b> (vocal sounding sine LFO controlled filter)	pot0 : rate pot1 : filter range pot2 : lfo depth/filter resonance fdbk : envelope sensitivity	<b>filter range:</b> CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds <b>envelope sensitivity:</b> allows to envelope control the rate
<b>step filter</b> (vocal sounding sample&hold controlled filter)	pot0 : rate pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	<b>filter range:</b> CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds <b>envelope sensitivity:</b> allows to envelope control the rate
<b>step filter bit crushed</b> (vocal sounding sample&hold controlled filter, running through a bit crusher)	pot0 : rate pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	<b>filter range:</b> CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds <b>envelope sensitivity:</b> allows to envelope control the rate
Other		
<b>tannhauser gate</b> (bitcrusher, pitch-shifter, filter, reverb)	pot0 : amount of pitch shifting pot1 : filter/pitch control pot2 : amount of filter sweeping fdbk : reverb	this patch was made with expression use in mind. Pot1 (exp) controls pitch and filter simultaneously. pot0 defines how much pot1 controls the pitch (0 to -1oct) pot2 controls how much pot1 affects the filter.
<b>hounds</b> (A synth-ey patch inspired by Kate Bush's running up that hill. A pitch shifter the glides up to the played note with a user definable glissando)	pot0 : glissando pot1 : filter pot2 : vibrato fdbk : envelope sensitivity	
<b>drones</b> (2 individual oscillator drones)	pot0 : drone 1 frequency pot1 : drone 2 frequency pot2 : mix of both drones fdbk : FM modulation	When you turn up the FM modulation (fdbk) the frequency of drone 2 will be used to FM modulate drone 1