Delays		
stretcholay (delay whose repeats get reversed and time stretched)	pot0 : loop pot1 : stretch pot2 : grain size fdbk : feedback	pot0 : from CCW to CW it defines how much of your audio is being recorded and how much is looped. At full CW it will stop recording and only loop. The loop degrades after some time pot1 : CCW no stretching -> noon max stretching - > CW stretched grains bein to be rearranged in reverse order pot2 : defines the size of grains that are used to stretch tha sample, from CCW with larger grains and gentle slopes to CW with shorter grains and choppy slopes
reversolay (reverse repeats that can be pitch shifted up to 3 octaves)	pot0 : loop pot1 : pitch pot2 : sample size fdbk : feedback	pot0 : below noon you are recording and playing, above noon it will loop the last sample and stop recording pot1 : pitch intervals : unison, +5semitones, +7semitones, +1oct, +1oct and 5semitones, +1oct and 7semitones, +2oct, +3oct
verbolay filter (a blend of delay and reverb plus filtered feedback)	pot0 : tone pot1 : reverb pot2 : delay fdbk : feedback	pot1 goes from fully delay ccw to fully reverb cw
verbolay pitch (a blend of delay and reverb plus endless detuned ascending or descending feedback)	pot0 : pitch pot1 : reverb pot2 : delay fdbk : feedback	pot1 goes from fully delay ccw to fully reverb cw pitch goes down ccw and up cw
particolay (delay with feedback chopped up in grains)	pot0 : envelope sensivity pot1 : particle rate pot2 : delay fdbk : feedback	when you increase envelope sensitivity, the harder you play the more the rate is going to slow down
ringolay (delay with feedback going through a ring modulator)	pot0 : ring mod blend pot1 : ring mod rate pot2 : delay fdbk : feedback	
pitcholay (delay with feedback going through a pitch shifter)	pot0 : tone pot1 : pitch pot2 : delay fdbk : feedback	**pitch:** at noon it's unison, CCW is -1 octave, CW is +1 octave
aliaserolay (delay with feedback going through an aliaser)	pot0 : aliaser blend pot1 : aliaser pot2 : delay fdbk : feedback	
analog(-ey) mod delay (analog sounding delay with filtered and modulated feedback)	pot0 : mod width pot1 : mod rate pot2 : delay fdbk : feedback	
square pitcher (delay with repeats going thru a pitch shifter that is modulated with a square wave LFO)	pot0 : LFO depth/pitch pot1 : LFO rate pot2 : delay fdbk : feedback	

Loopy, samply, freezy stuff		
hold aliaser (hold a bit of audio indefinitely, well almost)	pot0 : reverb size / hold pot1 : aliaser pot2 : filter fdbk : send dry to aliaser smplrt: change pitch	pot0: up to noon it sets the reverb size and level at max it will hold the current layer of sound,
tape loop (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)	pot0 : loop pot1 : tape speed pot2 : tape length fdbk : feedback smplrt: change pitch	**pot0 :** below noon you are recording and playing, above noon it will loop the last sample and stop recording **pot2 :** reducing it to the smallest settings can create glitchy artifacts, flanger-like sounds and other weird stuff, is interactive with tape speed. Note: when pot1 is at noon, pot2 has little to no effect.
envoloop vibrato (endless repeating delay loops are envelope triggered and run through a vibrato)	pot0 : sensitivity pot1 : vibrato rate pot2 : delay fdbk : decay	**sensitivity:** 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. **delay time:** defines the length of the looped sample. On very short setttings, the loop will fade out faster. **vibrato rate:** applies vibrato to the looped samples **decay:** fully CW the lamples can loop infinitely (well almost). As you dial it back the loop will start to fade out.
envoloop reverb (endless repeating delay loops are envelope triggered and run through a reverb)	pot0 : sensitivity pot1 : reverb blend pot2 : delay fdbk : decay	**sensitivity:** determines how much signal is added into the loop while a portion of the previous loop is faded out. When set to minimum the signal is looped infinitely. **delay time:** defines the length of the looped sample. On very short setttings, the loop will fade out faster. reverb mix: the amount of reverb that is mixed into thwe looped signal **decay:** fully CW the lamples can loop infinitely (well almost). As you dial it back the loop will start to fade out.

Glitchy, noisy, random stuff		
glitcholay (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.
pitch-glitcholay (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 :-)
pitch step glider (sample&hold modulated pitch shifter with discreet steps or glissando, can feedback on itself)	pot0 : glide pot1 : rate pot2 : pitch depth fdbk : feedback	**glide:** 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
pitch square lfo (pitch shifter modulated by a square LFO whose rate can be randomized, can feedback on itself)	pot0 : pitch depth pot1 : rate pot2 : randomness fdbk : feedback	
Modulation		
flanger (pretty wide range, from earpiercing accelerated jet plane sounds to super boing rubber sounds)	pot0 : lfo rate pot1 : range (delay length) pot2 : lfo depth fdbk : feedback	range : CCW gives you the shortest delay times and more airplane-like conventional flanger sounds. As you increase the delay, you start to hear very short repeats. It's set to work with wet only, but you can add dry if you like it
flanger barberpole TZF ring (an endlessly descending or ascending flanger, capable of going through zero, feedback running through a ring modulator)	pot0 : rate/direction pot1 : through-zeroness pot2 : ringmod mix/rate fdbk : feedback	**pot0** 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. **pot1** flanges closer to zero as you turn CW **pot2** introduces ringmodulation (to the feedback only) careful on the amount of feedback on some settings :-) It's set to work with wet only, but you can add dry if you like it
granpa's vinyl (add vinyl crackle, noise, pitch warp and frequency bandwidth loss to your sound)	pot0 : rate pot1 : pitch warp pot2 : filter fdbk : noise & crackle	
dynamic ringulator (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	pot0: as you turn up the blend the s&H will affect the ring mod frequency more fdbk: envelope will affect ring mod frequency and s&h rate
dynamic vibrato (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	**randomness** goes from sine modulation ccw to random steps cw **envelope sensitivity** sets how much the guitar signal will affect the rate
dynamic phaser (phaser whose rate can be envelope controlled)	pot0 : regen pot1 : rate pot2 : depth fdbk : envelope sensitivity	**envelope sensitivity** 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		
envelope filter and optional bit crusher (vocal sounding envelope controlled filter, running through a bit crusher)	pot0 : bit crusher blend pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	**filter range:** CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds
envelope filter aliased (vocal sounding envelope controlled filter running through an aliaser)	pot0 : aliasing pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	**filter range:** CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds
envelope filter ring modulated (vocal sounding envelope controlled filter, running through a ring modulator)	pot0 : ring mod rate pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	**filter range:** CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds **envelope sensitivity** controls the filter frequency but also the ring mod rate
lfo filter (vocal sounding sine LFO controlled filter)	pot0 : rate pot1 : filter range pot2 : lfo depth/filter resonance fdbk : envelope sensitivity	**filter range:** CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds **envelope sensitivty:** allows to envelope control the rate
step filter (vocal sounding sample&hold controlled filter)	pot0 : rate pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	**filter range:** CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds **envelope sensitivty:** allows to envelope control the rate
step filter bit crushed (vocal sounding sample&hold controlled filter, running through a bit crusher)	pot0 : rate pot1 : filter range pot2 : filter resonance fdbk : envelope sensitivity	**filter range:** CCW is a low pass. CW is a high pass. In the middle when you mix both together you get some more vocal sounds **envelope sensitivty:** allows to envelope control the rate
Other		
tannhauser gate (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.
hounds (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 duration pot1 : filter pot2 : vibrato fdbk : envelope sensitivity	**enveloppe sensitivity :** defines how well the audio input triggers the enveloppe follower that controls the pitch glissando and filter sweep.
drones (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