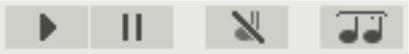
arp!0 - groovy midi arpeggiator - quick reference

notable features

- independent length for each control sequence. parameter polyrhythms!
- all real-time controls visible at once. change anything immediately.
- expandable control grids for even more immediacy!
- accent and offset sequences for cool rhythmic grooves.
- step length and voices sequences for uneven, chordal arpeggios.
- directions sequence for quick changes to input note order.
- variants for quick change arps with mouse or midi program changes.
- user definable sort transforms rearrange input notes in unique ways.
- syncs to the beat when the host is playing or recording.
- plays in time as settings change, and through variant & preset changes!
- complete midi control.
- all arp parameters saved in Reaper presets.
- real-time display of active notes, played notes, and current steps.
- pronounced "arp!-oh" (the bang is silent :^)

top left buttons



▶ play - runs arp

⏸ pause - pauses arp. keeps current notes ready to play.

♪̸ clear - clears current notes. (icon: slashed notes)

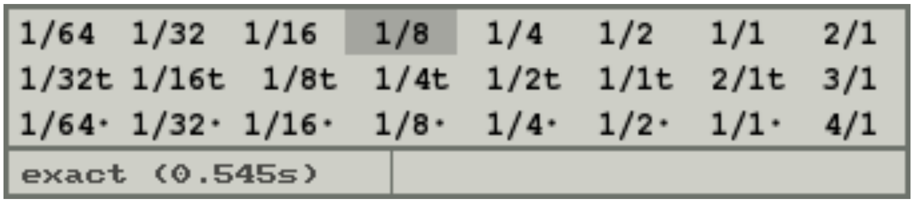
♫ hold - notes toggle on/off when enabled. (icon: 2 notes tied together)

note that while playing, the last played step is highlighted in each sequence. but while paused, the next step to be played is highlighted. so when changing from pause to play there is a brief moment when the play positions change back to the previous step before the next step occurs.

tick

tick 1/16

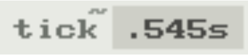
sets the time interval between arp steps according to host tempo or absolute time. tick has two modes: synchronized or absolute, which are selected on the setting pane. or right-click the "tick" label to toggle between these modes. in sync mode, the tick time is relative to the host tempo, and displayed as a fraction of a whole note. in this mode arp!0 syncs to exact beats when the host is playing or recording.



1/64	1/32	1/16	1/8	1/4	1/2	1/1	2/1
1/32t	1/16t	1/8t	1/4t	1/2t	1/1t	2/1t	3/1
1/64·	1/32·	1/16·	1/8·	1/4·	1/2·	1/1·	4/1

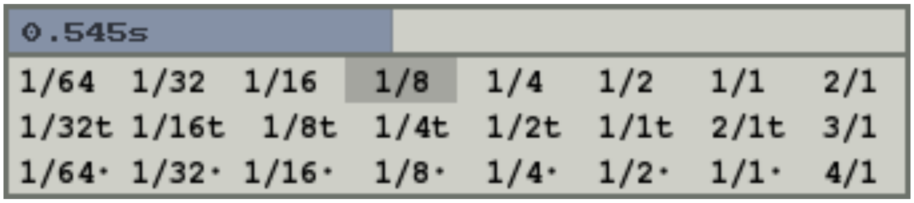
exact (0.545s)

click the sync fraction to show a dropdown menu of fractions and a scale slider. click any fraction to select it. the scale slider allows sync'd ticks that are not exact fractions. it shows a scaling factor from 0.75x to 2.0x, with a midpoint line at 1x. text at the left end of the slider shows the current scale factor, or "exact" if scaling is not in effect. it also shows the tick time in parenthesis. click in the slider to set a scale factor. right-click a fraction to disable scaling. the sync fraction on the tick button will have a "+" or "-" suffix if scaled greater or less than the exact fraction.



tick~ .545s

in absolute mode the tick time is displayed as a decimal number of seconds, and the tick label will have a "~" suffix.



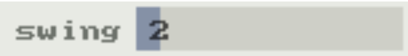
0.545s

1/64	1/32	1/16	1/8	1/4	1/2	1/1	2/1
1/32t	1/16t	1/8t	1/4t	1/2t	1/1t	2/1t	3/1
1/64·	1/32·	1/16·	1/8·	1/4·	1/2·	1/1·	4/1

click the time to show a slider and a grid of fractions. text at the left end of the slider shows the tick time in seconds. click the slider to select a tick interval. the fractions select the range of the slider relative to the current host tempo, again from 0.75x to 2.0x of the fraction of a whole note selected. click these fractions to change the range of the slider, but not change the current tick time. right-click a fraction to change the slider range and select the 1x value on the slider as a tick time.

shift-click the tick button for a high-resolution slider with a range of +/- 10% of the current value.

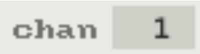
swing



swing 2

specifies extended time for certain steps, usually every other one. the value is a percent of the step length by which the following step is delayed. the swing steps value on the settings pane sets the number of steps between the extended steps.

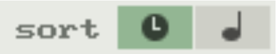
channel



chan 1

sets input & output midi channel used by arp!0. the midi bus value on the settings pane sets which Reaper bus this channel operates on. arp!0 eats all notes input on this bus and channel.

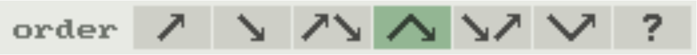
sort



sets the initial ordering of input notes.

🕒 time - notes ordered by time received
𝅗𝅥 note - notes ordered by pitch

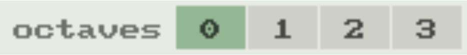
order



sets normal play order of notes based on the sort and any active sort transform.

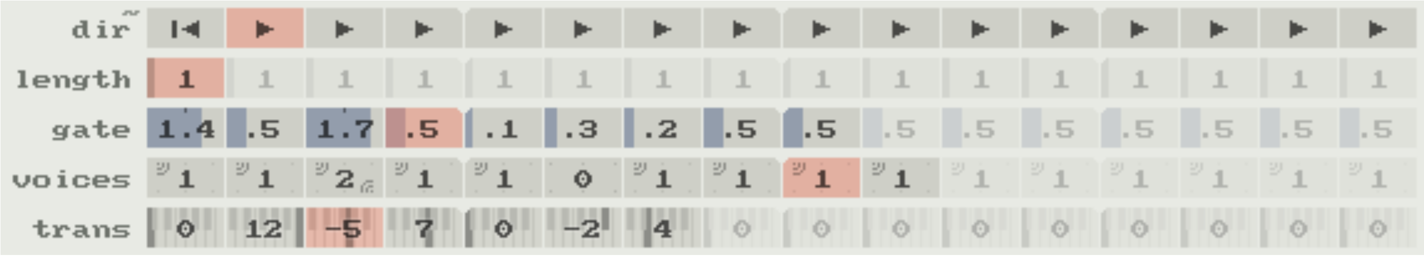
↗ up - low to high: 1,2,3,1,2,3...
↘ down - high to low: 3,2,1,3,2,1...
↗↘ up&down - up, then down. high/low notes repeat: 1,2,3,3,2,1,1,2,3,3,2,1...
∧ updown - up-down: singular high/low notes: 1,2,3,2,1,2,3,2,1...
↘↗ down&up - down, then up: 3,2,1,1,2,3,3,2,1,1,2,3...
∨ downup - down-up: 3,2,1,2,3,2,1,2,3...
? random - random note and octave

octaves



notes repeat in order for the set number of additional octaves. there are two modes for multiple octaves: replicant and heuristic, described in the settings section below. in heuristic mode, if octaves is > 0 the current octave is indicated by a period after the current octave number. control/ command-click an octaves value to set the current octave.

button grids

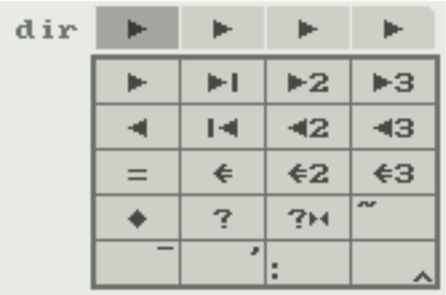


dir~	|◀	▶	▶	▶	▶	▶	▶	▶	▶	▶	▶	▶	▶	▶	▶	▶
length	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
gate	1.4	.5	1.7	.5	.1	.3	.2	.5	.5	.5	.5	.5	.5	.5	.5	.5
voices	1	1	2	1	1	0	1	1	1	1	1	1	1	1	1	1
trans	0	12	-5	7	0	-2	4	0	0	0	0	0	0	0	0	0

button grids are clickable rows of icons or values. clicking these shows a popup grid of choices, or a slider. clicking in the popup changes the value for a step. shift-clicking changes the value of the step clicked and all those to the right of it. alt/option-clicking resets the clicked step to a default value. right-clicking a step sets the sequence length of the row clicked. control/command-clicking sets the current play position. shift modifies both right and control/command clicks to change the length or play position of *all* sequences. alt/option-right-click to delete a step. control/command-right-click to insert a step. note that because arp!0 generates output ahead of the beat, changing the play position may be most useful when paused. the button grid label will have a "~" suffix if the play order for that row is random. right-click the label to toggle random play order.

button grids can be expanded to show control grids for all steps at once. this makes it easier to change the values of many steps in a particular sequence quickly. click the row label to reveal the grid. click cells or sliders in the grid to set the values of sequence steps. drag across the grid to change many values quickly. click the grid label, or anywhere left of the control grid, to hide it and show the button grid again.

dir



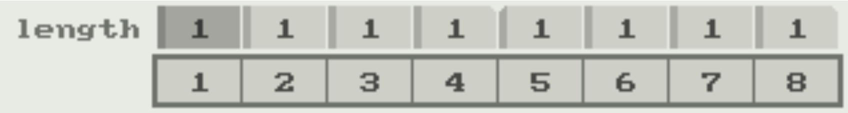
dir	▶	▶	▶	▶
	▶	▶|	▶2	▶3
	◀	|◀	◀2	◀3
	=	←	←2	←3
	◆	?	?▶◀	~
	–	,	:	^

directions which modify the play order. the most common dir types are:

- ▶ next step: plays next note in sequence based on note sort, sort transform, and order. this is the "normal" next note.
- ▶| last step: plays the last note in the input note sequence.
- ◀ previous step: plays the previous note in the input note sequence.
- |◀ first step: plays the first note in the input note sequence.
- = same step: repeats the immediately previous step.

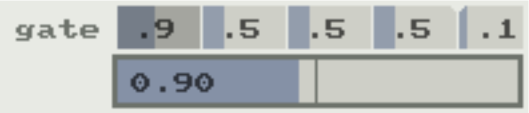
the other dir types are described in the “direction sequence step types” section below.

length



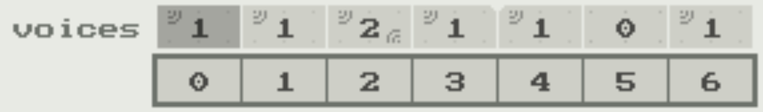
the length of an arp step as an integer multiple of the tick time.

gate



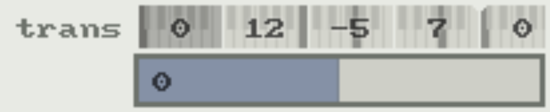
the duration of notes generated for a step as a fractional multiple of the tick time, from (almost) 0.0x to 2.0x.

voices



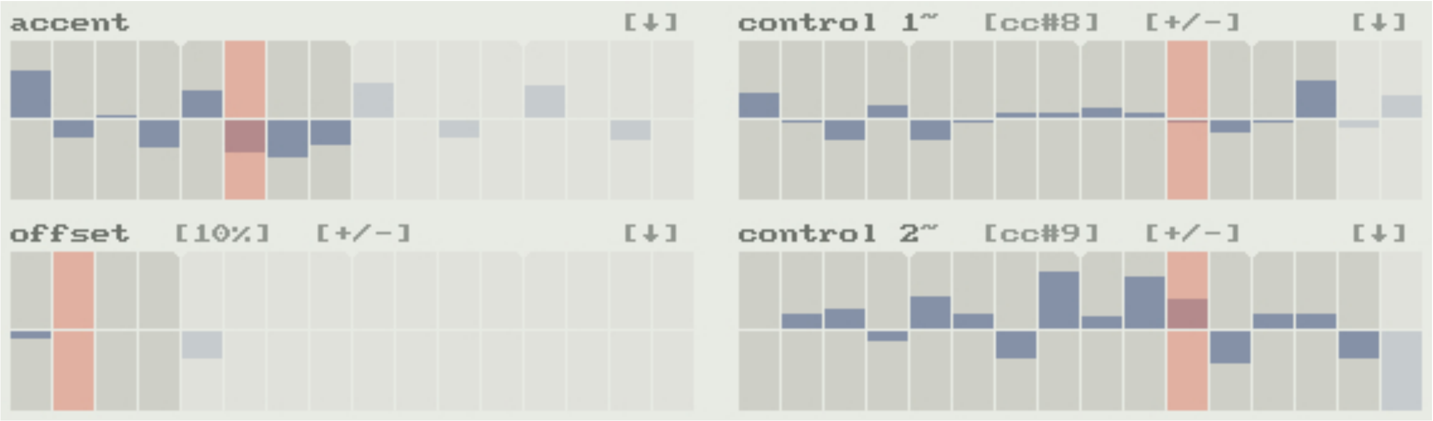
number of notes generated for a step. each voice uses another note from the currently active input notes.

trans



transposition of the notes generated for a step, in semi-tones.

slider grids



slider grids are clickable sequences of sliders. clicking and dragging in a slider changes its value. dragging across sliders will change many values quickly. shift-clicking also changes the value of all sliders to the right of the one clicked. alt/option-clicking resets the clicked slider to a default value. right-clicking a slider sets the sequence length for that grid. control/command-clicking sets the current play position for the grid. shift modifies both right and control/command clicks to change the length or play position of *all* sequences. expand [<down arrow>] and collapse [<up arrow>] clickers toggle height of slider grid rows. the button grid label will have a "~" suffix if the play order for that row is random. right-click the label to toggle random play order.

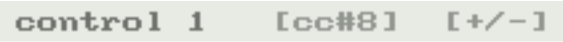
accent

modifies the velocity of the notes generated for a step. values higher than the midpoint scale the velocity from the original to maximum. values less than the midpoint scale it from the original down to 1.

offset

offsets the start time of step notes by a percent of the tick time. right of the label are indicators of the offset range and polarity. the offset range is shown as a percent of the tick time in square brackets clicking this percent value cycles the offset range between +/- 10%, 25%, 50% and 100% of the tick time. offset polarity is shown as unipolar [+++] or bipolar [+/-]. click the indicator to toggle polarity. unipolar offsets delay note start times. for bipolar offsets values higher than the midpoint delay the start time. values lower than the midpoint advance the start time ahead of the beat. this is possible because arp!0 operates ahead of the beat by the current bipolar offset range. NOTE: to maximize arp!0's responsiveness, use the minimum necessary range when using bipolar offsets. note input and parameter changes within the bipolar offset range before a step are effective one step later. input notes must be ahead of the beat by at least the offset range to avoid this delay with bipolar offsets. NOTE: for immediate arp output when host playback starts use unipolar offsets.

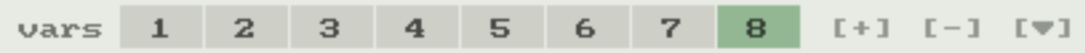
control



specifies midi controller values which are generated for each step. the midi cc number is shown in brackets to the right of the label. clicking the cc number shows a popup grid to set which midi cc is sent. pitch bend or note improbability are also options. OFF disables controller output. right of the cc number is an indicator of whether the slider grid displays unipolar [+++] or bipolar [+/-] data. clicking this indicator toggles the grid polarity.

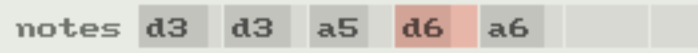
note improbability is an alternative to pitch bend for two of the four control sequences. positive improbabilities are the probability that note output for a step is suppressed, leaving a gap in arp output. negative bipolar values are the probability that a step is skipped completely. in this case the next step occurs immediately, *if* it also passes the probability test. this is so it is possible to skip multiple steps at high negative improbabilities.

vars



allows selection of one of several arp variants. right of the label are a row of buttons, one for each variant that has been created. click a button to load that variant, which includes all non-midi, non-global arp settings. right-click to load the variant and reset sequence play positions to saved starting values. shift-click to prevent changes from the previous variant from being saved before loading another. right of the variant buttons are add [+] and delete [-] text controls. click [+] to create a new copy of the current variant, up to a maximum of 12. shift-click [+] to make a copy of the current variant without saving changes to the original. click [-] to delete the current variant. click again on the red [-] to confirm the deletion. right of the add & delete controls is a [<down triangle>] text control which displays a dropdown menu. the first two menu items copy and paste current variant values between variants and presets. the "apply changes to ALL variants" item applies any changes made to the current variant to all other variants as well. *NOTE* edits to the current variant are normally saved when selecting the next variant. reselect a current variant to save any edits to it. shift-click on the current variant to abandon all edits since it was last saved. right-click on the current variant to save it and reset play positions. variants can be changed using midi controller messages. enable this on the settings pane. details below in the settings/help pane section.

notes



a view of the first 16 currently active notes, in the sort order. the note velocity is shown as a shaded bar behind the note name. notes in these first 16 which have been output in the current step are highlighted. the notes label will have a <right arrow> prefix if loading a preset also loads any notes that were active when it was saved. right-click the label to toggle loading saved preset notes.

direction sequence step types

▶ next step: plays next note in sequence based on note sort, sort transform, and order. this is the "normal" next note.

▶| last step: plays the last note in the input note sequence.

▶2 2nd next step: plays the 2nd note ahead in the input note sequence.

▶3 3rd next step: plays the 3rd note ahead in the input note sequence.

◀ previous step: plays the previous note in the input note sequence.

|◀ first step: plays the first note in the input note sequence.

◀2 2nd previous step: plays the 2nd previous note in the input note sequence.

◀3 3rd previous step: plays the 3rd previous note in the input note sequence.

= same step: repeats the immediately previous step.

← back one step in time: replays the note which was played just before the previous note. this is based on the history of notes arp!0 has played, and can be different from previous notes in the input note sequence when the normal sequence order has been changed by special steps like first/last/random/etc.

←2 back 2 steps in time: replays the 2nd note prior to the last.

←3 back 3 steps in time: replays the 3rd note prior to the last.

◆ reverse: alternates between up and down unidirectional orders. changes the current play order for bidirectional orders (up-down/updown/down-up/downup). by default, repeats the previous note in sequence, which gives the effect of the "sticky" up-down and down-up orders. add the contract meta to avoid this duplicate note and give the effect of the "bouncy" updown/downup orders.

? random step: plays a random step in the input note sequence.

?▶◀ random next or previous step: randomly does either next step or previous step.

meta directions are optional behaviors which can be added to the normal dir types. when a meta dir is selected in the dir dropdown menu it will be added or removed from whatever basic step type is already selected for that dir step. the meta dir types are:

~ twiddle meta: reserved

– subtract meta: suppresses note output for a step.

' contract meta: combines a dir step with the following dir step. contracted dir steps change the input note sequence just as if they occurred normally. but the contracted dir outputs no notes, takes no time, and is immediately followed by the next dir sequence step. contracted dir steps provide a way to specify a step relative to another step. for example:

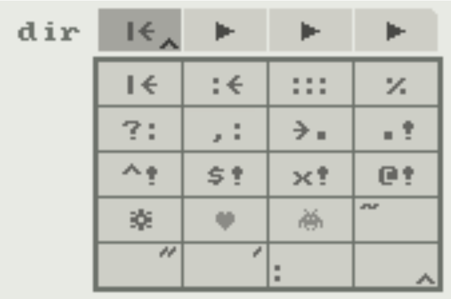
|◀' ▶ specifies the 2nd note in the input sequence: "|<'" is a contracted "first step" dir step. it selects the first note in the input sequence, but otherwise does nothing because it is contracted.   the following ">" dir step then generates the next note after that, which is the 2nd note.

▶2' ▶3 specifies the 5th note ahead in the input sequence. ">2'" selects the 2nd note ahead, but does not play it because it is contracted. ">3" plays the 3rd note after that, which is 5 notes ahead.

? ⇐' ▶ when repeated, generates random notes interleaved with sequential notes every other step. "?" selects a random note from the input sequence. "<='" selects the whatever note was output before that random note, but generates no output because it is contracted. ">" then generates the next note after that in the input sequence.

: section meta: marks a step for special dir sequence operations which are triggered by dir operation steps.

^ directory operation meta: sets a step to an alternate dir type, most of which alter the order of the dir sequence itself. when the dirop meta is selected, the dir types dropdown changes to show the alternate types available:



|⇐ first dir: seeks back to the first dir step.

:⇐ previous section: rewind the dir sequence to the closest previous step with a section meta. this is useful to restart the dir sequence from some point after the first step. it enables an initial, one time set of prefix dir steps followed by a different set of repeating dir steps.

::: multisection: counts as 3 sections for the random and sequential section dirops. useful for weighting the probability of a random section, and for repeating a sequential section. when combined with the prime meta, counts as 7 sections, or 31 when also combined with the section meta. when combined with the prime2 meta, counts as 15 sections, or 95 when also combined with the section meta. these values enable musically useful repeat counts for sequential sections.

% resortx: triggers an update for any active sort transform.

?: random section: seek forward in the dir sequence randomly to one of the following steps marked with the section meta. will not seek past a final section step. will not seek to sections inside a nested random/sequential section-final section pair.

,: sequential section: seeks to the following section steps sequentially, one each time the sequential section dirop occurs. the prime and prime2 metas allow two additional independent sets of sequential sections.

>. end section: seeks forward in the dir sequence to the step following the final section step, or back to the beginning of the dir sequence if there is no following final section step.

.! final section: terminates a set of random or sequential sections. random/sequential section dirops will not seek forward past this step, and end section seeks to the step following this one.

^! on first note: seeks to the following dir step when the input note sequence reaches the first note. note that this dir step takes effect for the next note *after* the first. if there are multiple on first note dirops they will take effect sequentially each time the input note sequence reaches the first note. this is also true for the following “on...” dirops.

on first note has a variant when combined with the prime/prime2 metas. on first note with prime seeks back to the next preceding instance of on first note with prime2. if there is no preceding instance with prime2, it wraps around to seek a following instance. these variants are also effective for the following “on...” dirops.

$! on last note: seeks to the following dir step when the input note sequence reaches the last note.

x! on passing note: seeks to the following dir step when the input note sequence passes the end of the sequence. for up and down orders, this is when the note sequence repeats. for updown/up+down/downup/down+up orders this is when the note sequence changes direction. whereas the on first and last note dirops will not trigger if the first or last note is skipped, on crossover note will always happen after the note sequence cycles or changes direction. but note that it triggers *after* the first note of the new cycle or order.

@! on host seek: seeks to the following dir step when the host seeks forward or backward.

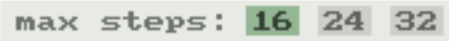
✲ reset play positions: resets the play position of *all* control sequences to their last saved positions, including the dir sequence. aka boom.

′ prime meta: selects variant behavior for some dirops.

″ prime2 meta: selects secondary variant behavior for some dirops.

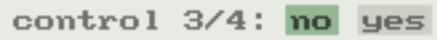
settings/help pane

click the [arp!0] logo to reveal a settings & help pane. settings are keyword lists with a button colored background following a label. click a keyword to select that setting. selected keywords are highlighted. current settings are:



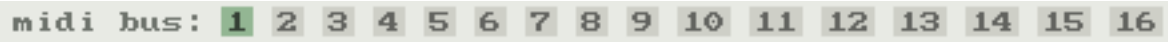
max steps: 16 24 32

select extended 24/32 step modes, or the standard 16.



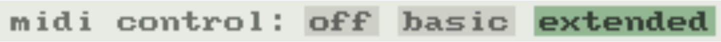
control 3/4: no yes

enable optional 3rd and 4th control sequences.



midi bus: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

select which Reaper midi bus is used for arp!0 input & output.



midi control: off basic extended

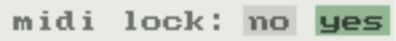
enable midi cc control. basic midi control enables midi cc's on arp!0's main midi channel as follows:

cc	effect
32	select variant
33	if nonzero, variant changes will also reset all sequence play positions to saved values. this setting persists until reset by a zero value cc.
34	if nonzero, variant changes will not save any edits made since the variant was selected. this setting persists until reset by a zero value cc.
35	if nonzero, selects the next higher variant. wraps around if the current variant is last.
36	if nonzero, selects the next lower variant. wraps around if the current variant is first.
37	if nonzero, resets all sequence play positions to saved values.
38	if nonzero, enables arp playback.
39	if nonzero, pauses arp playback.
40	if nonzero, clears any current arp notes.
41	if nonzero, toggles the note hold control.
48	sets the play position for the dir sequence.
49	sets the play position for the length sequence.
50	sets the play position for the gate sequence.
51	sets the play position for the voices sequence.
52	sets the play position for the trans sequence.
53	sets the play position for the accent sequence.
54	sets the play position for the offset sequence.

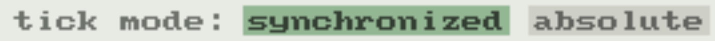
55 sets the play position for the control 1 sequence.
56 sets the play position for the control 2 sequence.
57 sets the play position for the control 3 sequence.
58 sets the play position for the control 4 sequence.

extended midi control enables midi cc’s on the 4 channels above arp!0’s main channel to change step values for the control sequences. channel main+1 controls the dir sequence with midi cc’s 24-55, the dir sequence metas with cc’s 56-87, and the length sequence with cc’s 88-119. channel main+2 controls the gate sequence with midi cc’s 24-55, the voices sequence with cc’s 56-87, and the trans sequence with cc’s 88-119. channel main+3 controls the accent sequence with midi cc’s 24-55, the offset sequence with cc’s 56-87, and the control 1 sequence with cc’s 88-119. channel main+4 controls the control 2 sequence with midi cc’s 24-55, the control 3 sequence with cc’s 56-87, and the control 4 sequence with cc’s 88-119.

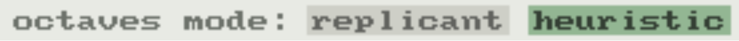
when extended midi control is enabled, arp!0 will output midi cc’s on the extended channels as edits are made to the control sequences. arp!0 will also send midi cc’s for all control sequence step values when presets are loaded or variants are changed. this output enables synchronization with midi controllers.


midi lock: no yes

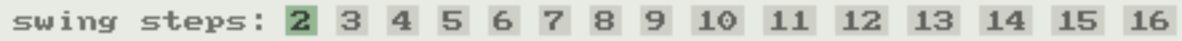
disable changes to arp!0’s midi channel, bus, and midi control settings when loading new presets. when midi lock is yes, the “chan” label on the main pane will have a “~” suffix. you can right click the “chan” label to toggle midi lock. this is a global setting which is not saved with presets. the default value, including defaults for locked midi channel, bus and control, can be changed by editing the global settings file described below in colors.


tick mode: synchronized absolute

select whether the tick interval is synchronized with the host tempo or an absolute time. this setting can also be toggled by right-clicking the “tick” label on the main pane.

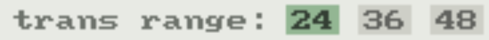

octaves mode: replicant heuristic

select how to implement additional octaves. replicant mode adds octave notes to a virtual note list after the actual input notes. these virtual notes will be shown in the active note list and behave just like additional input notes at higher octaves. this allows seamless transitions across octave boundaries for complex sort transforms, and avoids some interesting behaviors at octave boundaries when input notes are reordered by the dir sequence. heuristic octave mode uses rules to play notes at higher octaves based on when the first and last notes from the input note sequence are encountered. this means that sort transforms and the dir sequence operate on input notes without regard to octaves. this can produce different behaviors at octave boundaries compared to replicant mode. some of these differences are interesting and cool. some not so much. this setting can also be toggled by right-clicking the “octaves” label on the main pane.



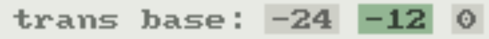
swing steps: [✓ 2] 3 4 5 6 7 8 9 10 11 12 13 14 15 16

select the number of steps across which swing is applied. arp!0 will swing every number steps on the beat.



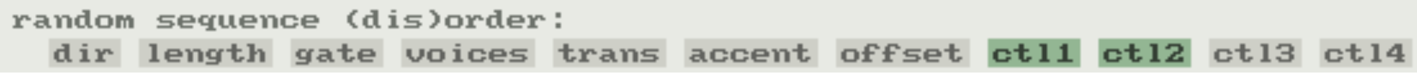
trans range: [✓ 24] 36 48

select the editable range for transposition sequence steps in semitones.



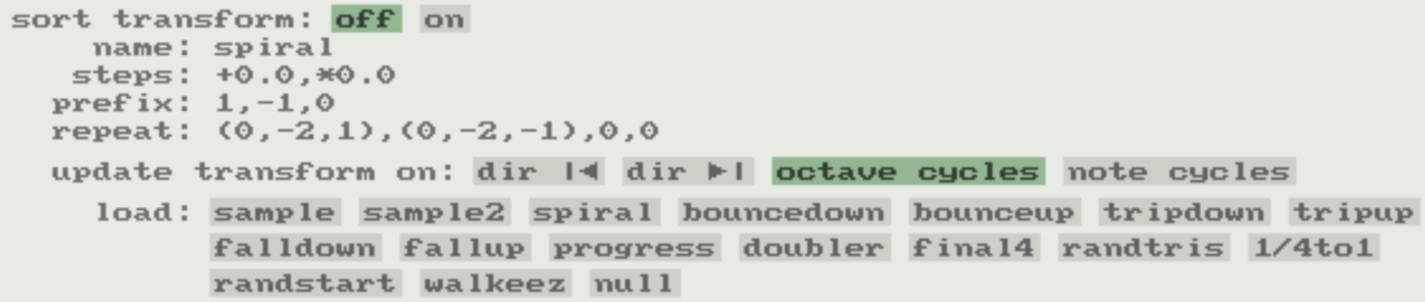
trans base: -24 [✓ -12] 0

select the lower limit for transposition sequence steps in semitones. changing the transposition base will immediately change the octave of notes generated by arp!0.



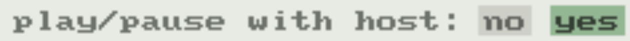
random sequence (dis)order:
dir length gate voices trans accent offset [✓ ctl1] [✓ ctl2] ctl3 ctl4

select sequences which will have random playback order. random order sequences are indicated by a "~" char after the sequence label on the main pane. right-clicking those labels also toggles random order.

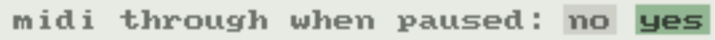


sort transform: [✓ off] on
name: spiral
steps: +0.0,*0.0
prefix: 1,-1,0
repeat: (0,-2,1),(0,-2,-1),0,0
update transform on: dir |◀ dir ▶| [✓ octave cycles] note cycles
load: sample sample2 spiral bouncedown bounceup tripdown tripup falldown fallup progress doubler final4 randtris 1/4to1 randstart walkeez null

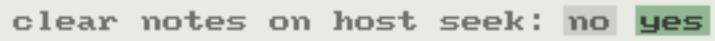
sort transforms rearrange current input notes according to specifications loaded from text files. select off or on to enable any currently loaded sort transform. the name and specification of the current transform are shown next. the "update transform on:" section shows events which will reapply the active sort transform to current input notes if selected here. this is useful to update the reordered notes at intervals when a (semi-)random sort transform is active. in the "load:" section, the names of available sort transforms are listed. click a sort transform name to load and activate it. the transforms are loaded from text files. these transform specification text files are located in the bangzero_data folder inside the Data folder in Reaper's resource folder. the files are arpbangzero_sortx.txt for user transforms and arpbangzero_defsortx.txt for the default transforms included with arp!0. see the sample transforms in arpbangzero_sortx.txt for details on the transform specification. if a sort transform is active the sort label on the main arp!0 pane will have a "~" suffix. right-click the sort label to toggle the last selected sort transform on and off. right click either sort button for a dropdown menu. the first menu item is the name of the currently loaded sort transform, if any, prefixed by "~". click this name to toggle the transform on and off. after that are items which copy and paste the current sort transform between variants and presets. last are a series of menu items which will load sort transforms directly from the main pane. select the "load:" menu item or click the "load:" label on the settings pane to reload the transforms from their source specification files. note: the current sort transform is saved and restored with presets. loading a sort transform only reads a transform specification into the current preset. sort transforms in presets do not change when the transform specification they were originally loaded from is modified. sort transform specification files are an independent, modifiable source for sort transforms in presets.


play/pause with host: no [✓ yes]

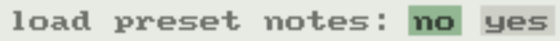
set whether arp!0 will automatically play and pause when host playback stars and stops. this is a global setting and is not saved with presets. to change the default value edit the global settings file described below in colors.


midi through when paused: no [✓ yes]

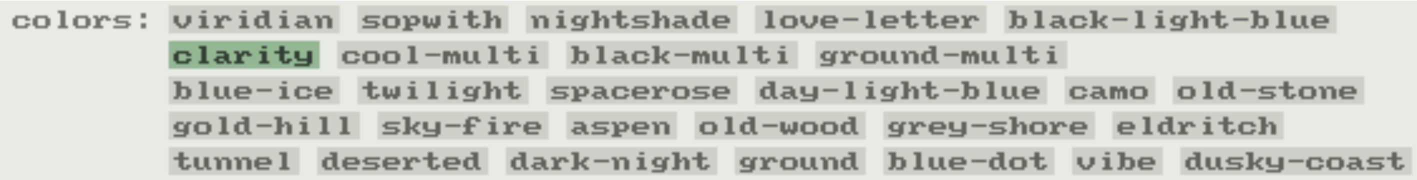
set whether arp!0 will pass input midi notes through when paused. this is another global setting.


clear notes on host seek: no [✓ yes]

set whether arp!0 will clear active notes when the host playback position changes significantly. another global setting.


load preset notes: [✓ no] yes

select yes to enable restoring active notes with presets. notes label will have a leading right arrow character if note restore is enabled. yet another global setting.


colors: viridian sopwith nightshade love-letter black-light-blue
[✓ clarity] cool-multi black-multi ground-multi
blue-ice twilight spacerose day-light-blue camo old-stone
gold-hill sky-fire aspen old-wood grey-shore eldritch
tunnel deserted dark-night ground blue-dot vibe dusky-coast

select different color schemes. schemes after the first 5 are defined in a global settings file and can be tweaked to taste. the settings file is in Data/arpbangero_files/arpbangzero_settings.txt in the Reaper resources folder. click the "colors:" label to reload the global settings file. additional default color schemes are defined in Data/arpbangzero_files/arpbangzero_defcolors.txt. these color schemes are listed starting on a new line after those in the global settings file.

saved sequence playback positions

arp!0 saves initial playback positions for all control sequences. these saved positions are restored when the host begins playback, when a variant is selected with right-click, when the restore play position midi cc is received, or when the 'boom' dirop occurs. by default the saved play position for all sequences is the first step. this can be changed by control/command-clicking to set a sequence play position while arp!0 is paused. when paused, saved play positions are indicated by a small cutout at the bottom left corner of a step. saved play positions are saved and restored with presets. to quickly set play positions for all sequences to the saved locations right-click on the current variant button.

integration with host

arp!0 saves and restores all settings in Reaper presets, including sequence play positions, color scheme and optionally current notes. use "Save preset as default..." to save your favorite settings.

when host begins playback, arp!0 restores sequence play positions from the last time they were changed manually or restored from a preset. saved play positions are lost when Reaper quits & restarts. save & restore them from a preset if necessary.

tips and tricks

- while paused, right click the play button to play the single next step. while playing and not sync'd to a playing host, right click the play button to immediately play the next step.
- right-clicking in dropdowns usually changes the setting but leaves the dropdown visible and active. this is particularly useful in the tick and dir dropdowns where multiple clicks may be necessary to select a desired value.
- alt/option-click the [arp!0] logo to reload the global settings file, including user color schemes. control/command-click the logo to toggle these global options: play pause with host, midi through when paused, and clear notes on host sync.
- right-click when setting a swing value to set the swing to twice the value selected. this makes swing values of ~25-50% possible.
- alt+control/option+command-click when enabling the contract meta dir to set an experimental additive contract mode. this adds notes from the contracted step to the following step. this is similar to a voices step >1, but can combine notes from different, non-sequential places in the input note sequence. also, each added note is transposed independently.
- the heart and space invader dirops implement an experimental dir subsequence feature. the invader dirop temporarily replaces the current dir sequence with a oneshot dir sequence from next variant. the heart dirop in a dir subsequence immediately resumes the invoking dir sequence. the invader dirop with prime or prime2 metas takes the dir subsequence from the 2nd or 3rd next variants.
- alt/option-click many labels to show/hide semi-relevant debug info courtesy of IX. right-click the [arp!0] logo to hide debug info.

installation

the arp!0 zip file will unzip into something like this:

```
  midi_arpbangzero_v000/
    midi_arpbangzero
    README.txt
    arpbangzero_documentation.pdf
    arpbangzero_changes.txt
    gpl.txt
    bangzero_files/
      <lots o' bits>
    bangzero_data/
      arpbangzero_settings.txt
      arpbangzero_defcolors.txt
      arpbangzero_sortx.txt
      arpbangzero_defsortx.txt
```

after unzipping, select "Show REAPER resource path in explorer/finder..." from the Reaper Options menu to open the Reaper resources folder. in that folder there will be an Effects subfolder for Jesusonic fx plugins, and a Data folder. copy the midi_arpbangzero Js script file and the bangzero_files folder into the Effects folder. for new installations, copy the bangzero_data folder into the Data folder.
after installation you should have something like this:

```
  <Reaper resources folder>/
    Effects/
      midi_arpbangzero
      bangzero_files/
        <lots o' bits>
    Data/
      bangzero_data/
        arpbangzero_settings.txt
        arpbangzero_defcolors.txt
        arpbangzero_sortx.txt
        arpbangzero_defsortx.txt
```

when updating, replace midi_arpbangzero and bangzero_files in the Effects folder. if there are significant changes to the global settings or sort transforms i will note this at the top of this README.txt file. in that case, if you have customized the global settings or sort transforms, you will need to merge your changes with the files in the bangzero_data folder. this includes the arpbangzero_settings.txt global settings file and the arpbangzero_sortx.txt sort transforms file.

credits

first and formost, highest praise and thanks to Justin Frankel and the Cockos team for Jesusonic, the audio/midi hacker's paradise. it is *so* cool to be able to create & customize fx plugins that integrate with the Reaper ecosystem. heya! hey!

arp!0 was largely inspired by the arpeggiator in Urs Heckmann's amazing Zebra2 software synthesizer, <http://www.u-he.com/cms/zebra>, and by the many cool arp presets available for Zebra. in addition to the powerful step and voices options, the way the entire Zebra arp is visible and accessible makes it exceptionally *useable*. i wanted this useability with other softsynths, and also the option to record and postprocess arp output.

i have wanted the groovy accent and offset features for a long time. seeing these realized in Arto Vaarala's Kirnu arp was inspiring. Kirnu has many unique capabilities of its own: <http://www.artovaarala.com/index.html>.

the arp!0 user interface mostly grew out of the natural possibilities of the Jesusonic graphics api. but a major inspiration for the clean, flat look are Sean Costello's beautiful interfaces for his awesome ValhallaDSP fx: <http://www.valhalladsp.com/>.

a quick heyah! to sound design craftsman Michael Cavallo, who's marvelously   mutable mc Percaluptus Zebra preset was good company as i developed   arp!0. Michael's Zebra soundbanks are great, and his arp patches are   especially luminous: <http://www.monomo-sounddesign.com/>.

big thanks to all the marvelous beta testers in the Reaper scripting forum! arp!0 would be a pale, buggy shadow of itself without their help. so many great ideas! so little time. :^)

finally, all praise and honor to Baba O'Riley and all the rhythmic synth pioneers. we stand on the shoulders of giants in this teenage wasteland.

feedback welcome: bang at forum.cockos.com - email: bang at bangzero dot org

___