Arpeggiator

NOAH - Tactive Instrument Modeller

creamw@re© fidelity at work.

Introduction

The Arpeggiator presents you with tremendous possibilities:

- Use it to effortlessly (and interactively) generate melodic sequences in real time – with myriad control options, all of which can be adjusted at any time to yield practically endless variations – for direct use, to stimulate your own musical imagination, or to advance the frontiers of musical-acoustic research!

- Even for accomplished keyboardists, for whom keyboard technique is not an issue, the Arpeggiator is a *fabulous* labor-saving device!

- The Arpeggiator is a highly dependable tool for bringing *any* relationship (musical *or* non-musical) to a speedy end. It's therefore also a sure-fire vehicle for getting your solo career (musical *or* non-musical) off to an early start!

But seriously ...

No idea what an arpeggiator is? In simplest terms, it's a device which *captures* chords (or short sequences of notes) that you play into it, and then *scans* the captured notes, periodically sending them out one at a time, thus generating *arpeggios* from your input.

The results of this simple process can be annoying and banal, or truly esoteric. Quite a lot depends upon the sounds you apply it to and the contexts you use it in. We won't presume to prescribe these things for you. All options are open. The Arpeggiator includes enough features and flexibility to enable you reach *any* extreme!

Many features of the arpeggiator are easier to understand intuitively than by reading about them. If you can't make heads or tails of something you read here – just go ahead and try it out. You can't damage anything. Then come back here and reread the section in question and it'll probably make perfect sense.

If you're just getting acquainted with the Arpeggiator, and if you already have a basic idea of what an arpeggiator is, there's nothing at all wrong with going straight into hands-on experimentation – you can refer back to the manual to go more in-depth or for clarification of details as needed. But *do* read the manual sometime – there are features here you wouldn't want to miss out on!

Overview

Basics

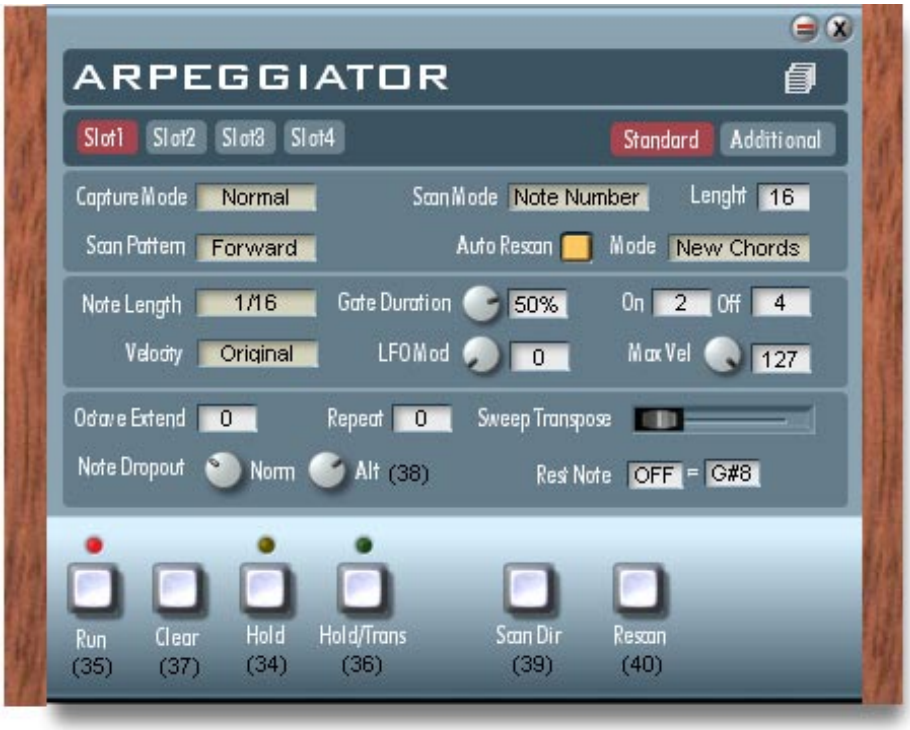
This is a MIDI arpeggiator device:

- It is driven by MIDI note events on its **MIDI** input, which it *captures* in its internal chord buffer.

- It continually *scans* this chord buffer according to the current settings and produces MIDI note events as output.

> (Note: in this text, MIDI note event and other input is referred to as coming from a MIDI keyboard, although the arpeggiator can of course be driven by *any* live or sequenced source of note events.)

The arpeggiator is monophonic. It puts out one note at a time, no overlapping notes and no chords.



-

MIDI Message Handling

The handling of received MIDI messages depends on their type. In general:

- Received **note-on messages** are captured in the chord buffer (up to sixteen at any one time).

- Received **note-off messages** may or may not cause the matching note-on messages to be removed from the chord buffer, depending upon the current settings.

- *Received* note messages are *not* echoed to the output while the arpeggiator is running. The note events which the arpeggiator sends out are primarily those which the arpeggiator itself *generates* via scanning of the captured chord.

- All other received *channel messages* (e.g., modwheel, pitch bend, etc.) are echoed directly to the output at all times and have no particular pre-defined effect upon the arpeggiator. (Of course, as with all Noah devices, you can assign MIDI controllers to various controls on the arpeggiator.)

Control via the MIDI Keyboard

A range of **MIDI** note numbers can optionally be dedicated to real-time control of a selected set of arpeggiator functions via the MIDI input (see **Big Buttons Group, MIDI Kbd Ctrl**). This permits these controls to be used more effectively in real time, as the response via the MIDI keyboard (in contrast to that of the corresponding graphical buttons) is virtually instantaneous. The numbers in parentheses next to various buttons on the surface indicate the MIDI note numbers currently assigned to each of these buttons or its associated performance function. These assignments can be changed via the **MIDI Kbd Ctrl** Options group on the additional page.

Timing and Synchronization

The arpeggiator's timing resolution is 24 clocks or pulses per quarter-note (PPQN). The note-on and note-off events it produces are aligned to these time increments.

Slots

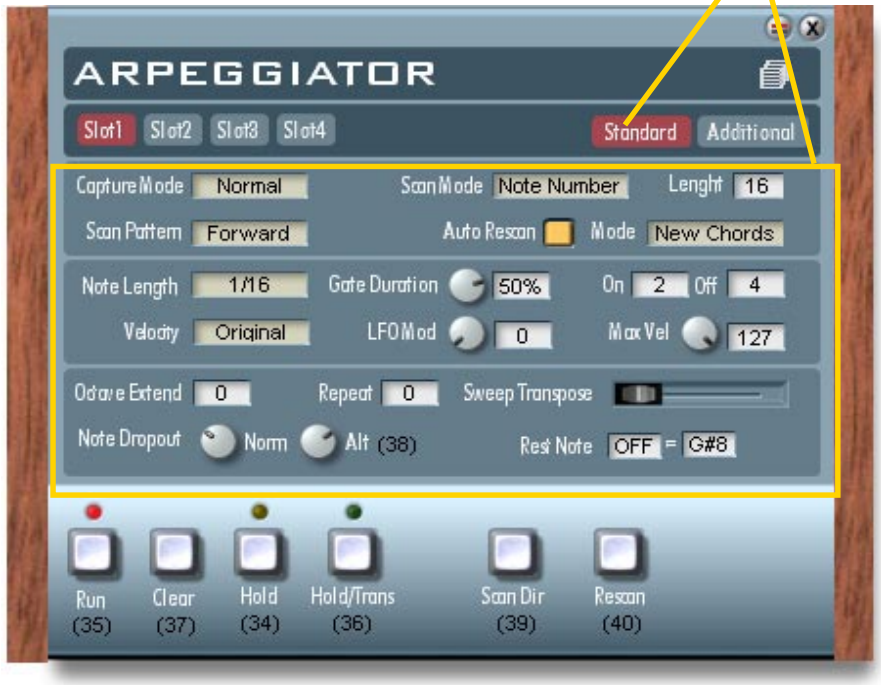
You can assign several instances of the Arpeggiator, so that each slot can have its own.

Use the switches - the number of which depends upon the operating mode (*Single, Multi*) and the hardware version (Noah, Noah EX) - to select the desired instance for display and editing.

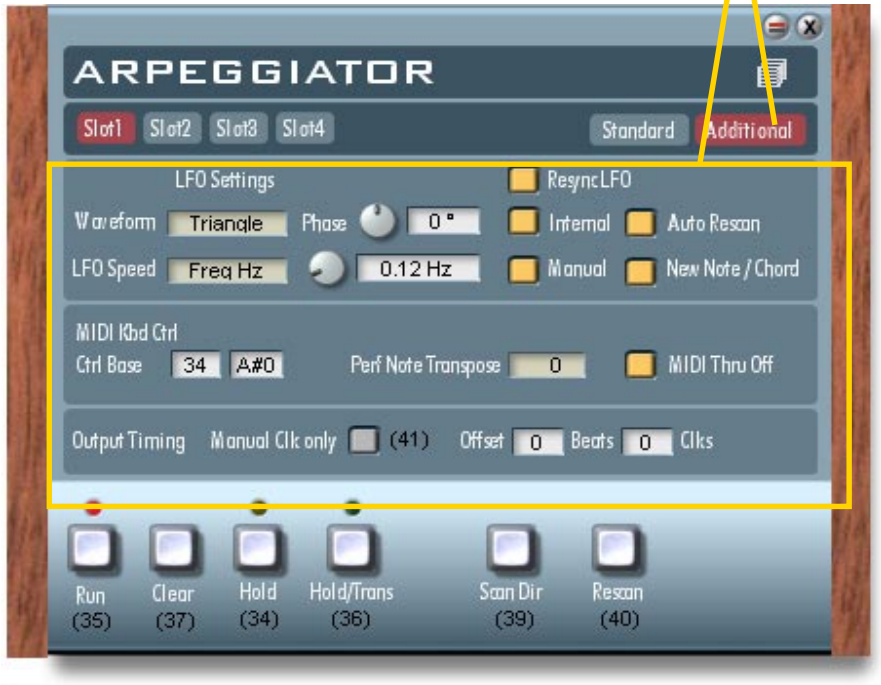
Functions

This section presents a point-by-point description of the buttons and other controls on the arpeggiator surface. The discussion here is broken down into sections which correspond to the organization of the surfaces/pages ***Standard*** and ***Additional*** itself:

Standard Page



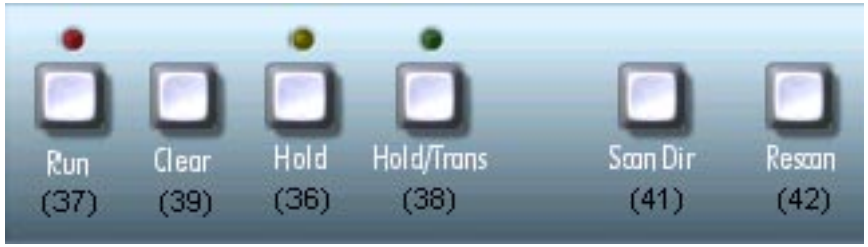
Additional Page



Big Buttons Group

This is the set of large buttons located at the bottom of the arpeggiator. These are essentially the "running" controls. Viewed as a group, they produce immediate, "big" changes. This distinguishes them from *settings* – i.e, all *other* controls on the surface – which can likewise be adjusted at any time, but whose effects are less "major" and not always immediately audible.

Each of the functions associated with one of the Big Buttons can also be actuated from the MIDI keyboard (see **MIDI Keyboard Control**). This is generally a better way to go, as the response of these functions to MIDI control is virtually instantaneous. The number in parentheses below each of these buttons indicates the MIDI note number currently assigned to it. These assignments can be changed via the **MIDI Kbd Ctrl / Ctrl base** parameter on the *Additional* page.



Run/Stop: This is the On/Off button – it alternately activates and deactivates the arpeggiator. Stopping the arpeggiator clears the chord buffer and terminates the current output note, if any. Note that when the arpeggiator is not running, it passes *all* received MIDI events directly through to its output, and the other buttons in this group are disabled.

Clear: Pressing this button "empties" the arpeggiator (i.e., clears the chord buffer), permitting a completely new chord to be captured. The arpeggiator continues to run as before.

> **Clear** also deactivates **Hold** and **Hold/Trans** if they are active (see below).

Hold: This function freezes the chord buffer, locking the current captured chord into the arpeggiator. Incoming MIDI note events are no longer captured, nor can they cause the notes which have already been captured to be removed. Instead, received MIDI note events are passed directly through to the output. This lets you "accompany" the arpeggiator live.

> Once activated, **Hold** can be deactivated only via **Clear** or **Run/Stop**.
>
> Note: **Hold** is also activated automatically whenever **Hold/Trans** (see below) is activated.

Hold/Trans: Activating **Hold/Trans** will instantly activate **Hold** (see above) if it is not already active, thus freezing the chord buffer. While **Hold/Trans** is active, the arpeggiator output can be ”live-transposed” up or down (simple semitone transpose) from the MIDI keyboard.

The transpose produced by playing any note on the keyboard is equal to the offset of this note relative to middle C (MIDI 60). While **Hold/Trans** is active, the keyboard has no effect upon the captured chord, other than transposing it as described.

However, **Hold/Trans** (unlike **Hold**) can be activated and deactivated freely. When **Hold/Trans** is deactivated, **Hold** remains active, as does the last transpose value applied under **Hold/Trans**. This allows switching between transposition and accompaniment of the frozen (and still-transposed) chord.

The transpose produced under **Hold/Trans** is cleared via **Clear** or **Run/Stop**, and is thus always zero whenever **Hold** or **Hold/Trans** is first activated.

Scan Dir:

> Note: this function is effective *only* when **Scan Pattern** is set to **Fwd-Rev**.

Hitting this button causes a reversal of the instantaneous arpeggiator scan "direction".

For example, when the arpeggiator is producing notes going up the scale/keyboard, **Scan Dir** will cause it to reverse and go down the scale instead, beginning with the next output note – without affecting the output timing in any way.

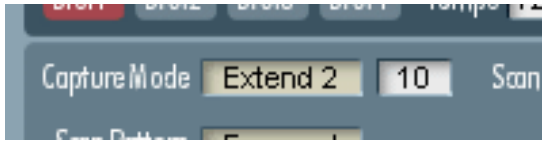
ReScan: This function "restarts" the arpeggiator scan each time it is actuated. The next arpeggiator output note is the one which should appear "first", based on the current **Scan Mode** and other settings (for example, the lowest note in the current captured chord).

Standard Page

Capture Mode

These settings control the way in which received MIDI notes are *captured* to (and removed from) the chord buffer. One of the four possible modes is always active. In all capture modes, the arpeggiator keeps track of the order in which captured notes arrived, in addition to recording note number (of course) and velocity for each note. Thus, the use of both **Note Number** and **Note Order** scan modes is always possible, regardless of the **Capture Mode** setting.

The capture mode setting can be changed at any time. In some cases, after doing this, you may need to use **Clear** to empty the chord buffer completely (e.g., if you change from **Auto Hold** to **Normal** after releasing all keys).



Normal: In this mode, captured notes remain in the chord buffer only for as long as the corresponding key is held down. The arpeggiator output pattern varies dynamically as keys are released. When no keys are being held down, the arpeggiator produces no output.

Auto Hold: Under **Auto Hold**, captured notes remain in the chord buffer indefinitely, even after all held keys have been released. The arpeggio continues to play as if all keys which have been played were still being held. New notes continue to be captured as long as at least one key is still being held. The first note or chord which is played following the release of all keys begins a new capture "session", at the same time clearing all previously captured notes.

Extend 1: Notes are added to the chord buffer as they are played and remain in the chord buffer indefinitely. This capture mode permits notes to be added one at a time. **Extend** mode thus makes it easy to create melodic arpeggios – a particular note can appear multiple times at different points within the arpeggio.

Note capture continues until the chord buffer is full (sixteen captured notes). The **Clear** button must be used to clear the chord buffer and silence the arpeggiator, or to capture a new chord.

> Note that **Scan Mode** (see following section) must be set to **Note Order** in order to have the arpeggiator play the notes back in the order in which they were captured.

Extend 2: Same as **Extend 1,** but new notes push old notes out of the chord buffer once the specified note capture limit ***n*** (adjustable via the adjacent text fader) has been reached.

Note capture can thus continue indefinitely – the chord buffer always contains (at most) the last *n* notes captured. Therefore, the rhythm of the arpeggiator output also remains fixed as new input notes are captured, once the limit has been reached. The note capture limit can be set to any value between 2 and 16 notes and can be adjusted while the arpeggiator is running.

Scan Mode

This control selects the basic method used to scan the captured chord and determine the next note to be played. The currently selected scan pattern (see below) produces a specific variation on the selected scan mode. The scan mode setting can be changed at any time.

Note Number: Scanning of the captured chord is done on the basis of note numbers – e.g., from lowest note to highest note.

Note Order: Scanning of the captured chord is done on the basis of the time sequence in which the notes were captured.

Scan Pattern

These controls determine the specific way in which the captured chord is scanned. These settings work in tandem with the **Scan Mode** setting. The **Scan Pattern** setting can be changed at any time.

Forward: The captured chord is scanned in order of increasing note number (**Scan Mode** set to **Note Number**) or in forward time sequence (**Scan Mode** set to **Note Order**).

Reverse: The captured chord is scanned in order of decreasing note number (**Scan Mode** set to **Note Number**) or in reverse time sequence (**Scan Mode** set to **Note Order**).

Fwd-Rev: Scanning of the captured chord scan alternates between **Forward** and **Reverse** as described above, reversing itself each time it reaches the "end" of a scan (highest/lowest or first/last note). The notes at either "end" of a scan are not repeated when the direction reverses (i.e., these notes get played only once, not twice).

Random: Chord scan follows a random pattern. The **Depth** control, which is available when Random is selected, sets the degree or range of randomness.

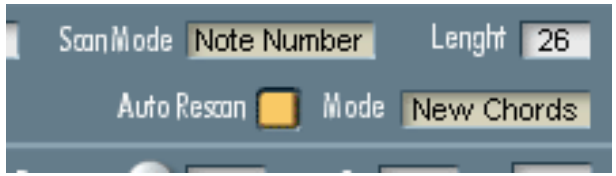
Random scan behaves differently under different **Scan Mode** settings:

- If **Scan Mode** is set to **Note Number**, then **Random** scan is basically a variation on normal **Fwd-Rev.** The scan proceeds in single steps from one note number to the next without skipping over any notes, but reverses its direction at random. With **Depth** set to minimum, this mode is in fact equivalent to **Fwd-Rev.** With **Depth** set to maximum, the scan reverses itself after almost every note (and thus tends to "stick in place", continually alternating between two notes).

- If **Scan Mode** is set to **Note Order**, then **Random** scan selects output notes randomly from among all possible output notes, taking the captured chord and all other settings (including **Octave Extend** – see **Output**) into consideration. The **Depth** control confines this selection to a specific range of "scan steps" away from the previous note in either direction. With **Depth** set to minimum, the arpeggiator "sticks" on a single note. With **Depth** set to maximum, the arpeggiator may select any note within 16 scan steps of the previous note – in effect, almost totally random.

Auto Rescan

When this function is enabled, the arpeggio is automatically re-processed once the number of notes as indicated by Length is reached. This is useful for imposing a specific rhythmic count or "loop length" on the output which does not depend upon the number of captured notes or other scan settings.



The auto rescan counter is reset whenever new notes or chords are played on the MIDI keyboard (see also **Mode** below) – however, *not* while **Hold** or **Hold/Trans** is active. It is also reset whenever the manual **ReScan** function is actuated (via the graphical button or the corresponding MIDI key).

> The auto rescan counter can also serve as a resynchronization source for the LFO.

Mode: This settings specifies whether the **Auto ReScan** counter should be reset whenever any new note is played on the MIDI keyboard **(New Notes**) or only when new *chords* are played (**New Chords**).

The difference between these two choices is that with **New Chords**, only the first note which is played *after all keys have been released* (e.g., the first note of a new chord) will cause the counter to be reset, whereas with **New Notes**, every note which is played will cause the counter to be reset, even if other keys are still being held down.

> For example: Using **Normal** capture mode, you can hold down the keys of a chord, and from time to time release individual keys or hold down new ones to modify the arpeggio while it plays. With **Mode**: **New Chords**, adding of new notes can be done at any time without changing the rhythm or shifting the "downbeat" of the resulting arpeggio. With **Mode**: **New Notes**, each new note you play redefines the downbeat to that point in time, cutting the current "bar" short and immediately beginning a new one – thus allowing you to redefine the dowbeat as you go, follow time-signature changes on the fly, etc.

Note that the **Mode** setting also affects keyboard-driven resynchronization of the LFO (when the option **New Note/ Chord** is activated – see **LFO Settings – ReSync LFO**).

Note Length: Controls the playback length of the notes in the arpeggio.

This setting also influences the LFO speed if the **LFO Speed – Beats/Clks** option is enabled (see **LFO Settings**).

Gate Duration : This sets the proportion of each arpeggiator beat during which the output note remains in the gate-on (sustain) state. Adjustment of this setting through its full range from minimum to maximum produces a gradual change in the arpeggiator output from staccato to legato. The number of clocks for gate-on and gate-off phases, respectively, are displayed for reference – these values cannot be adjusted directly.

Note that this control will have no effect at the minimum from *Note Length*, since in that case, the only available possibility is one clock for gate-on and one for gate-off.

Velocity: These settings provide control over the note-on velocity of arpeggiator output notes (Note-On-Velocity).

Use Orig Velocity / Replace Velocity: One of these two options is always in effect:

- When **Use Orig Velocity** is selected, output notes have the note-on velocity of the captured note from which they were generated (and the other velocity controls described below have no effect).

- When **Replace Velocity** is selected, the velocity values of the captured notes are ignored. Instead, output note velocity is determined by the following controls. It is possible to switch back and forth freely between these two modes – the actual captured notes are not affected in any way and the original velocity values are always available.

This has no influence over the captured notes whose original dynamics can always be restored.

LFO Modulation: This setting regulates modulation of output note-on velocity by the arpeggiator's built-in LFO. This setting is effective only when **Replace Velocity** is selected (see above).

LFO modulation causes the velocity values of arpeggiator output notes to vary over time. It functions by decreasing velocities relative to the **Max Velocity** value (see above). In effect, the **LFO Modulation** control sets the minimum modulated velocity.

For example:

- At the maximum **LFO Modulation** setting, velocity values vary between the **Max Velocity** setting and the absolute minimum possible MIDI velocity value of 1.

- At the "halfway" setting, by contrast, velocity values still reach **Max Velocity** at the high end of the modulation swing, but at the low end, they go only half as far down – i.e., to roughly one half of **Max Velocity**.

Max Velocity: Adjustable between 1 and 127. This setting is effective only when **Velocity:Replace** is selected (see above).

- When **LFO Modulation** (see below) is set to zero, this setting *directly* specifies the velocity of output notes, which is then constant.

- Otherwise, it sets the maximum velocity value which modulated output notes may have. Modulation by the LFO results in time-varying velocity values which are lower than this maximum.

(By the way – the **Max Velocity** control is *another* excellent candidate for assignment to a MIDI controller.)

Octave Extend

Produces cyclical upward transposition of the arpeggiator output by one or more octaves. The transpose amount is automatically "stepped" by one octave each time the arpeggiator completes a scan of the captured chord in the current scan direction. The captured chord is thus effectively extended into additional higher octaves as if the actual notes in the captured chord had been duplicated in those octaves. Setting **Octave Extend** to zero disables it.

"Stepping" of the **Octave Extend** transpose amount is always done in a manner which is consistent with the selected scan pattern. Assuming that **Octave Extend** is enabled (i.e, it is set to 1 or higher):

- With **Scan Pattern** set to **Forward**, output transpose is stepped upward by one octave following each pass through the captured chord until the until the scan in the highest octave (as specified by the **Octave Extend** setting) is complete. Transpose is then reset to 0 and the cycle repeats.

- With **Scan Pattern** set to **Reverse**, output transpose is stepped downward by one octave following each pass until a scan with a transpose of 0 is complete. Transpose is then reset to the highest octave (as specified by the **Octave Extend** setting) and the cycle repeats.

- With **Scan Pattern** set to **Fwd-Rev**, the scan direction is not reversed upon completion of a single forward scan of the captured chord, as would normally occur. Instead, output transpose is stepped upward by one octave and another forward scan is done. This repeats until the forward scan in the highest octave is complete – at this point, the scan direction reverses and the reverse scan is done, still in the highest octave. Subsequently, output

transpose is stepped downward by one octave each time a reverse scan is complete (likewise, with no scan direction reverse) until a scan with a transpose of 0 is is complete. The scan direction then switches again to forward and the entire cycle repeats.

- With **Random** scan patterns, the **Octave Extend** setting correspondingly extends (by the specified number of octaves) the set of possible output notes which the random scan can produce – again, as if the actual notes in the captured chord have been replicated in higher octaves.

Repeat

When set to values other than zero, this setting causes the arpeggiator to repeat each output note for the specified number of additional beats before proceeding to scan the captured chord for a new note. **Repeat** works with all scan modes and scan patterns.

Sweep Transpose

This function can be used to produce a dynamically variable "chord inversion" style of upward transpose. Progressively higher settings cause the currently lowest *output* note to be transposed upward by one or more full octaves so that it becomes the highest output note, in effect rolling or "sweeping" the output pattern up the keyboard one note at a time – but without altering its "chord value".

Sweep Transpose works with all scan modes and scan patterns. The control is ranged for a maximum transpose of four octaves and automatically bases itself with respect to the lowest note actually being played on the keyboard at any time. (Hint: the **Sweep Transpose** control is a *natural* for assignment to a performance controller or another MIDI Controller for "live" adjustment from the keyboard.)

Note Dropout

These controls can be left set to zero (full left) for normal use. Increasing the **Note Dropout** setting away from zero results in an increasingly probability that an output note will "drop out", or not appear, on any given beat. At the maximum setting, note output is completely suppressed. Scan timing and sequencing is otherwise not affected – these continue to function according to scan mode and scan pattern settings, **Repeats / Note**, etc., as though the dropped notes had been played normally. **Note Dropout** works with all scan modes and scan patterns.

The two **Note Dropout** controls are identical in their effect. The **Norm** control is the one whose setting is normally applied. The **Alt** control setting applies only while holding down the MIDI keyboard key assigned to the **Note Dropout Alt** function (see **MIDI Keyboard Control**).

This can be used for a variety of effects. The simplest one is to set the **Norm** control all the way to the right (100% dropout) and the **Alt** control full left. With these settings, arpeggiator output notes appear continously while the **Note Dropout SHIFT** key is held down – but *only* then.

Or, with the opposite settings, the **Note Dropout Alt**key becomes a "manual rest" or mute key, blocking arpeggiator output as long as it is held down (but not affecting its timing in any way).

More generally, the **Note Dropout Alt** key can be used to switch instantly back and forth between *any* two note dropout rates.

Rest Note

This control permits a specific MIDI input note to be designated as a "rest" note. Whenever the arpeggiator lands on this note during a scan of the captured chord, it inserts a rest – i.e., no output note is generated for this scan step. The inserted rest extends to include repeat notes, if any (see **Repeat**).

The **Rest Note** feature can be used to produce syncopated arpeggios in **Note Order** scan mode (hint: this is easiest with one of the **Extend** capture modes).

The **Rest Note** setting can be changed freely while the arpeggiator is running, with the result that rests appear at different points in the output. Setting **Rest Note** to Off (=128) disables the rest note function.

The scan mode can be switched to **Note Number** while the rest note function is active without any ill effects. This merely results in all rests (if there is more than one) occurring consecutively, since they are (of course) all generated from notes having the same note number.

Additional Page

LFO Settings

These settings control various parameters of the built-in LFO, which can be used to modulate output note-on velocity.

> Note: **Velocity** must be set to **Replace**, *and* **LFO Mod** (in the same group) must be set higher than minimum, in order for the controls in the **LFO Settings** group to have any effect.

Waveform: Five different LFO waveforms are available: Square, Sawtooth Up, Sawtooth Down, Triangle and Random.

Note that the LFO is applied to velocity in a negative direction. That is, higher (i.e., more positive) instantaneous LFO waveform values correspond to *lower* output note-on velocities. Thus, Sawtooth Up causes velocity to decrease gradually with time and then jump back up to maximum.

Phase: This setting adjusts the starting phase of the LFO – i.e., the point in its waveform to which the LFO springs when it is resynchronized.

Tip: "Good" values for this setting depend upon the selected waveform (see the list above). Setting it to zero causes the LFO to resync to a "zero-crossing" point, which is not always the most interesting starting point:

> - For **Saw Up** and **Saw Down** waveforms, a setting of 180° or -180° restarts the waveform at one "end", so that the modulation "ramps" up or down starting from one extreme of its range.
>
> - Similarly, **Triangle** waveforms are resynchronized to positive and negative "peaks" with phase settings of 90° and -90°, respectively.
>
> - By contrast, the **Square** waveform has only two values (+Max and –Max). Therefore, it doesn't "ramp" at all, but instead (as applied to note-on velocity) produces simple loud/soft rhythmic accenting – the **Phase** setting merely affects the "positioning" of this rhythm.
>
> - Finally, the **Phase** setting has no effect upon the Random waveform, which maintains a single constant (but random) value over the duration of each LFO waveform cycle.

LFO Speed: Two methods of LFO speed control are available:

Beats/Clks: Activating this option permits the LFO speed to be specified in terms of a number of whole arpeggiator beats and individual clocks (fractional beats). Under **Beats/Clks**, the LFO waveform goes through a complete cycle over precisely the specified amount of "rhythmical" time.

With this option, LFO speed is based upon current tempo and beat and adjusts itself accordingly if either the beat length (**Note Length**) or the tempo is changed.

Freq HZ: This permits LFO frequency to be specified directly (in Hz). If this option is enabled, the Frequency control will be available. With this option, LFO speed is completely unrelated to beat settings or tempo and is unaffected by changes to either of these.

ReSync LFO: The LFO can be restarted or *resynchronized* by various internal and external events to help produce controlled (or out-of-control) rhythmic effects. Upon resync, the LFO waveform jumps to the point determined by the **Phase** setting (above).

There are four resync sources which can be enabled singly or in any combination. A switch **Resync LFO** permits LFO resync enable/disable without the need to enable/disable the individual sources.

Internal: This indicates an "internal" scan restart. That is: in the course of normal scanning, the arpeggiator has once again returned to the "start" of its output pattern – e.g., the lowest output note. The timing of this event depends entirely upon the captured chord and all relevant scan settings. Correspondingly, it varies dynamically as the chord and/or settings are changed.

It should be mentioned that the arpeggiator, as a device which responds dynamically to its input and its settings, does not actually "know" in advance when an internal scan restart will occur, but merely detects this *when* it occurs. This means that LFO resync in response to internal scan restart occurs just barely too late to have an effect upon the "first" or scan restart note sent out by the arpeggiator – instead, the resynchronized LFO is applied starting with the *subsequent* output note. This is not an issue with the other LFO resync sources.

Manual: This indicates a scan restart triggered via the **ReScan** button or the corresponding MIDI key. When this source is enabled, the LFO is resynchronized whenever either of these occurs.

Auto ReScan: This indicates a scan restart triggered by the **Auto ReScan** beat counter (see **Scan Pattern – Auto ReScan**). When this source is enabled, the LFO is resynchronized periodically, according to the number of beats specified for **Auto ReScan**.

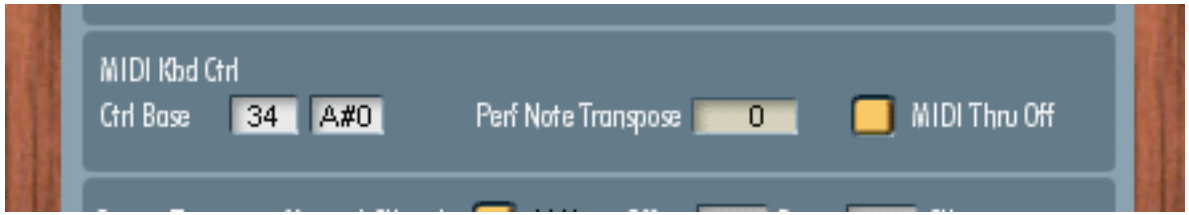
Note that the **Auto ReScan** beat counter can be used to trigger LFO resync *regardless* of whether **Auto ReScan** is itself currently enabled. Likewise, the options affecting the restarting of this counter (see **Auto ReScan – ReSync Ar Upon**:) also remain in effect when **Auto ReScan** is disabled, and thus affect the resynchronization of the LFO under this option.

New Note/Chord: When this source is enabled, the LFO is resynchronized in response to activity on the MIDI keyboard – i.e., whenever any new note is played on the keyboard, or *only* when new *chords* are played (as determined by the **Auto ReScan/Mode**).

However, this source is not effective while **Hold** or **Hold/Trans** is active, but is automatically disabled for the duration. Thus, live accompaniment or transposition of a "held" arpeggio via the MIDI keyboard is possible without upsetting the rhythm of the LFO modulation.

MIDI Keyboard Control

This group of settings provides control over the mapping of arpeggiator functions to the MIDI keyboard and the echoing of MIDI events from the arpeggiator input to its output.



MIDI Kbd Ctrl

A set of eight "performance" controls is assigned to a range of eight contiguous MIDI note numbers, referred to as the **MIDI Control Zone**.

The assigned controls include all buttons in the **Big Buttons** group (**Run/Stop, Clear, Hold, Hold/Trans, Scan Dir** and **ReScan**). This permits these controls to be used more effectively in real time, as the response via the MIDI keyboard (in contrast to that of the corresponding graphical buttons) is virtually instantaneous.

The MIDI Control Zone also contains two additional functions: **Manual Clocking** and **Note Dropout Alt**.

The MIDI Control Zone includes the **Ctrl Base** key and the next seven keys above it (see table following). MIDI notes within this range are used exclusively for control of arpeggiator functions and are *not* captured in the arpeggiator chord buffer, nor are they echoed to the arpeggiator MIDI output. All notes above and below this range are handled normally.

The MIDI Control Zone can be positioned wherever desired on the MIDI keyboard via the **Ctrl Base** setting (see below).

The mapping of functions to keys *within* the MIDI Control Zone is fixed, as shown in the following table. The layout is "optimized" for use with the base key set to **C** – typically at the lower end of the keyboard, to permit control with the left hand while playing with the right hand:

Key Position *	Assigned Function
0	Hold
1	Run/Stop
2	Hold/Trans
3	Clear
4	Note Dropout Alt
5	Scan Dir
6	ReScan
7	Manual Clocking

(*relative to **Ctrl Base** key)

The numbers in parentheses next to various buttons on the arpeggiator surface indicate the MIDI note numbers currently assigned to each of these buttons or its associated performance function in the MIDI Control Zone. These values are updated whenever the **Ctrl Base** setting is changed.

Ctrl Base: This setting permits the MIDI Control Zone to be positioned as desired on the MIDI keyboard – or to be moved completely out of the way, if it is not wanted.

Perf Note Transpose: To compensate for the loss of nearly an octave of the MIDI keyboard when the MIDI Control Zone is mapped into the active keyboard range, it is possible to "pre-transpose" incoming note events up or down by a full octave to permit a different range of notes to be used with the arpeggiator, if desired.

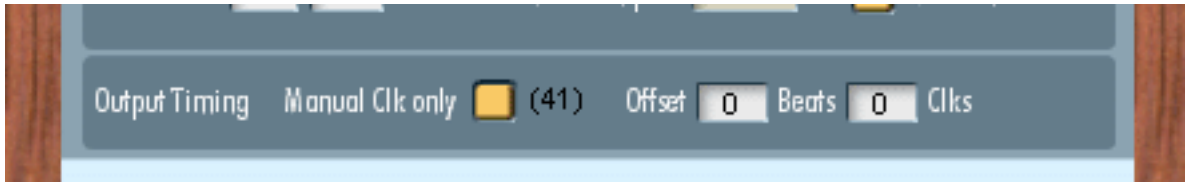
> For example, with the MIDI Control Zone positioned at the left end of the keyboard, setting **Perf Note Transpose** to **-12** restores access to the notes in the lowest octave of the keyboard for performance use. This is accompanied, of course, by a corresponding "sacrifice" of note range at the high end of the keyboard.

MIDI Thru Off: Normally, all MIDI messages other than note events are echoed directly to the arpeggiator output – as are note events as well, when the arpeggiator **Hold** function is active, or when the arpeggiator is not running.

In some situations, however, you may not want this. The **MIDI Thru Off** switch is provided for these situations – activating it suppresses *all* echoing of MIDI events from input to output.

Output Timing

These settings provide additional "special-purpose" control over the timing of output notes generated by the arpeggiator. They can be changed at any time.



Manual Clk Only: This option can normally be left switched off. Activating it disables the arpeggiator's free-running tempo clock (whether internal or synchronized to external MIDI), freezing the arpeggiator in time. Arpeggiator clocking (i.e., stepping) is now manual only, via the MIDI key assigned to the **Manual Clocking** function (as indicated in parentheses at far right).

Offset: This is an output delay setting which can be left set to zero for normal use. Non-zero values cause the arpeggiator output to drop "behind" the beat by the specified number of beats and clocks (see also **Clks / Beat**), by delaying the start of each scan accordingly.

The **Offset** function can be useful when two (or more) synchronized arpeggiators are being used together, connected to the same MIDI note source. One arpeggiator is used with **Offset** set to 0, the other with a non-zero **Offset** setting. The second arpeggiator produces output whose timing is delayed with respect to that of the first. This can be merely a delayed version of the output from the first arpeggiator – if all other settings on both arpeggiators are identical – or it can be radically different.

Setting of the output delay to fractional-beat values can be done via the **Clks** field. Among other possibilities, this permits an arpeggiator which is synchronized to a MIDI clock stream from a sequencer to drop its notes "between" or somewhat behind the beat instead of directly on it.

> Note that the specified offset is applied anew on every scan restart, regardless of whether the restart is triggered manually (**ReScan**), via the **Auto ReScan** beat counter, or in response to new note input from the keyboard – or if it is an "internal" restart detected and signalled by the arpeggiator each time it works its way back around to the "first" output note.

Index