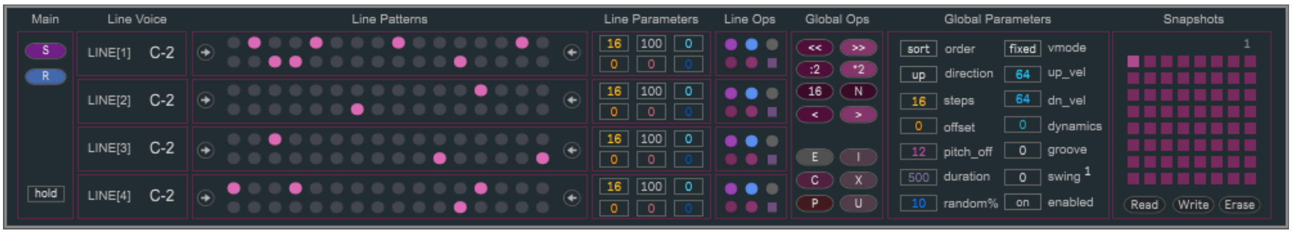
J74 ARPline

A flexible arpeggiator for Ableton Live



sort order
up direction
16 steps
0 offset
12 pitch_off
500 duration
10 random%
fixed vmode
64 up_vel
64 dn_vel
0 dynamics
0 groove
0 swing
on enabled

User Manual

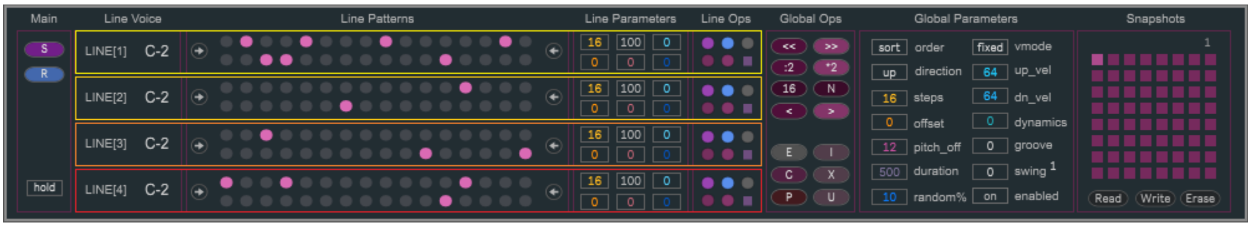
How to Install the J74 ARPline device

J74 ARPline is a flexible arpeggiator plugin for Ableton Live. It requires a Max for Live license to run (as in Ableton Live Suite) and it will run properly only in Ableton Live 9 or higher (older versions of Live, such as Live 8, are not supported).

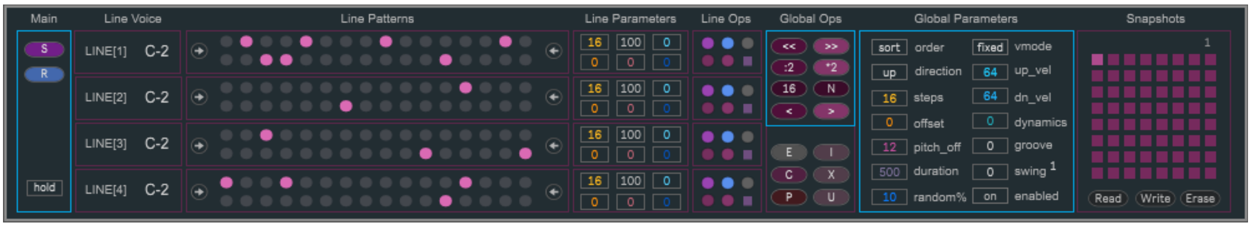
Installation of the tool set is extremely easy: unzip the download pack and copy the files to your computer hard drive, any place you like. To start it up, drop the plug-in device (.amxd) on a MIDI track in the Ableton Live liveset.

ARPline J74.amxd device overview

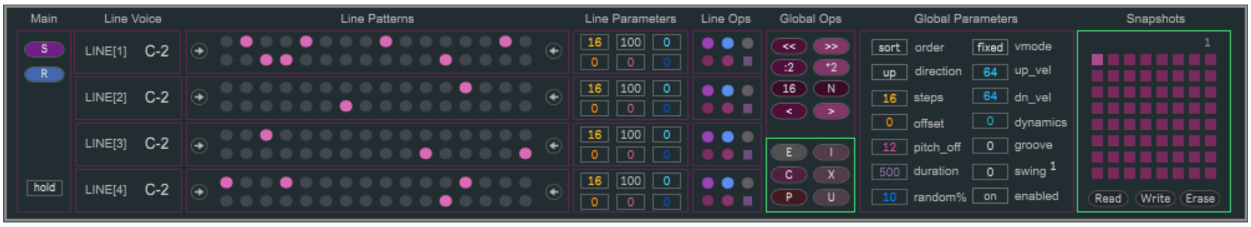
ARPline is an arpeggiator plugin processing and generating MIDI notes. With a lot of flexibility. At its heart it is composed of four independent *lines*, each line capable of the same features. A line is a programmable pattern for two note triggers and has independent controls such as number of steps, step offset, playback direction, speed, signature, trigger probability and swing/groove response.



When combined across the lines, these features allow you to create complex musical arpeggio's such as almost never-ending poly-metric and/or poly-rhythmic evolutions. On top of the line controls, the plugin offers also a set of global controls such as random arpeggio generation, pattern mixing, groove, swing and velocity dynamics.



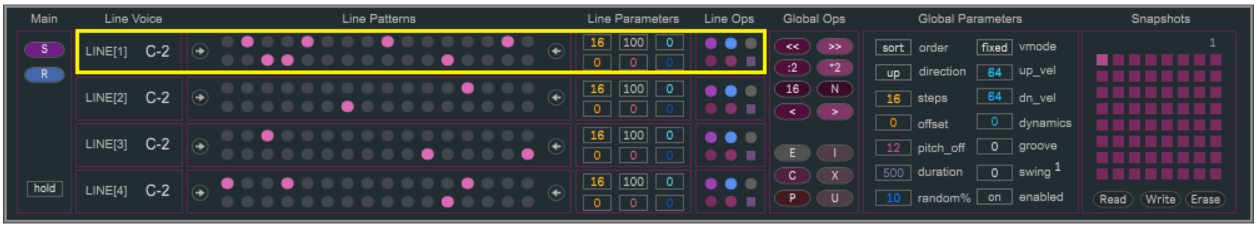
Finally the plugin, besides native support for state saving within Ableton Live (liveset and presets) also has its own capabilities for pattern banks and snapshots storage (and recall). A new pattern bank or a different snapshot can be loaded at any time, also during a performance (recall is hit-less).



Important: the device has a built in [Help] which shows you exactly where each control is located and a brief explanation of what it does. Please use this [Help] view while reviewing this user manual for better reading.

The Line Pattern Area (programming patterns) and Note triggers

The Line Pattern Area is where the editing of patterns takes place. The area presents 4 independent lines, each line having two rows. The *upper row* plays triggers for the MIDI note assigned to the line. The *lower row* plays triggers for the MIDI note given by the combination of the assigned note *and* the pitch offset given by the [**pitch_offset**] setting.



LINE[1] C-2 16 100 0 / 0 0 0
LINE[2] C-2 16 100 0 / 0 0 0
LINE[3] C-2 16 100 0 / 0 0 0
LINE[4] C-2 16 100 0 / 0 0 0

sort order
up direction
16 steps
0 offset
12 pitch_off
500 duration
10 random%
fixed vmode
64 up_vel
64 dn_vel
0 dynamics
0 groove
0 swing
on enabled

To activate a trigger, at a given spot, just click on the spot in question. You will see a purple “dot” appear. To deactivate a trigger click on an active (purple) dot, and the trigger will be removed.

Dynamic note assignments to lines

Notes are dynamically assigned to lines in typical fashion for an arpeggiator. You are expected to send MIDI notes (in input) to the device and the last active notes (up to four) will be taken for assignments. You normally send notes to the device by arming the track where the ARPline device is place and using a MIDI keyboard, MIDI plugin or a MIDI clip. To make the most of the ARPline device chords (of up to four notes) are expected in input.

The way assignment is done is defined by the following two global parameters [*Order*] and [*Direction*]:

- [**Order**] defines the assignments as either by sorted order or playing order. If {Sort} is chosen, notes will be first sorted in the direction specified by the [Direction] parameter (see next) and then assigned to lines. If [Play] is chosen notes will not be sorted and will be assigned in the receiving order.
- [**Direction**] If [Order] is set to {Sort} then direction {UP} will sort notes from low to high; if the direction is set to {DOWN} sorting is done in the opposite direction

Tip: With [Order] = {Sort} results, each time you enter a new chord, tend to be more constant. With [Order] = {Play} results are more unpredictable, as each time you enter a new chord the note input timing will possibly be different compared to what used to be the order in the previous iteration.

Using the [pitch_offset]

By default the [pitch_offset] is set to +12 so that triggers on the lower row of a line play notes an octave higher (which is often an interesting musical effect). This behavior can be changed to have more complex harmonic relations, such as by using fifths (+7 or -5), playing an octave lower (-12) instead or simply doubling the same pitch as the upper row (+0). Any semitone distance is allowed. Changes to the [pitch_offset] are immediately applied to the output.

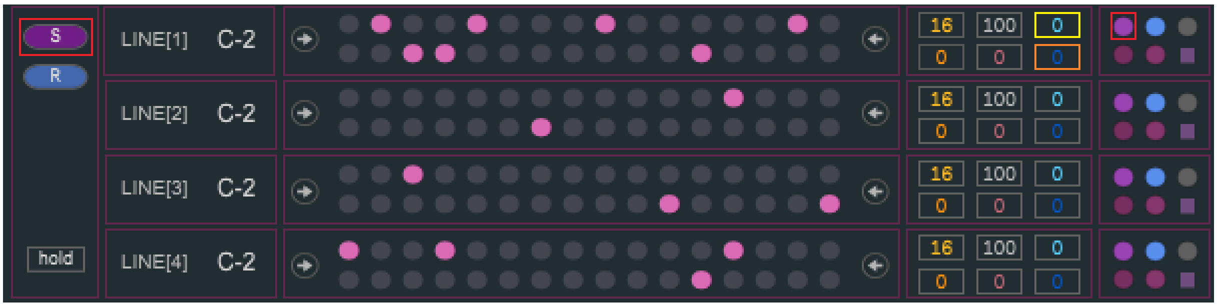
Note: Each line is *independent* and therefore it is perfectly possible to *trigger chords in output* by having active dots on the same step number on more than one line (there is no limit, you can have all 8 triggers active at the same time if you wish to). To create a more human feel when using simultaneous triggers it may be musical to add a little time fluctuation using the [Groove] a parameter ([Groove] is explained later in this user manual).

[Hold] Option

When activated the [Hold] options let's notes be hold even if the input has been released (ignoring note off MIDI messages from input) and until new notes are entered. This a classic arpeggiator feature.

Working with Line Patterns, pattern banks and slots [Line number]

Each line has its own 11 memory slots for storing and recalling patterns. Slot 0 is reserved to an empty pattern while slots 1 to 11 are freely programmable. Each memory slot is identified by a [**Line number**], which you can see highlighted in *yellow* in the picture below.



The [Line number] for each line can be selected independently from the other lines, allowing the patterns to be mixed-matched freely. This allows and extremely large set of possible combinations for the overall arpeggio pattern.

<u>Note</u>: The number of combinations is actually huge: 11*11*11*11 = 14641 unique combinations!

You can come up with different combinations of line patterns in multiple ways:

- By manually selecting the individual patterns using the [Line number] for each line
- By using the global random selector [**S**] (in the picture highlighted in red, on the left side). The global random selector randomly chooses a new [Line number] for each line
- By using the line random [**selector**] (in the picture highlighted in red, on the right side). The line random selector randomly chooses a new [Line number] for the selected line only
- You can also generate pattern lines randomly. We will look at random generation later in this user guide.

On top of slot selection, it is also possible to apply periodic variations to the line pattern combinations. Each line has its own [**Line Periodic Variation**] number (in the picture highlighted in orange) which allows you to randomly trigger a different slot (including the 0 slot) by a period of bars as specified by the value of the [Line Periodic Variation]. Value 0 means no variation, while any value greater than 0 defines the periodicity of the variation in bars. Selection automatically reverts to the pre-selected [Line number] after the variation has been played back.

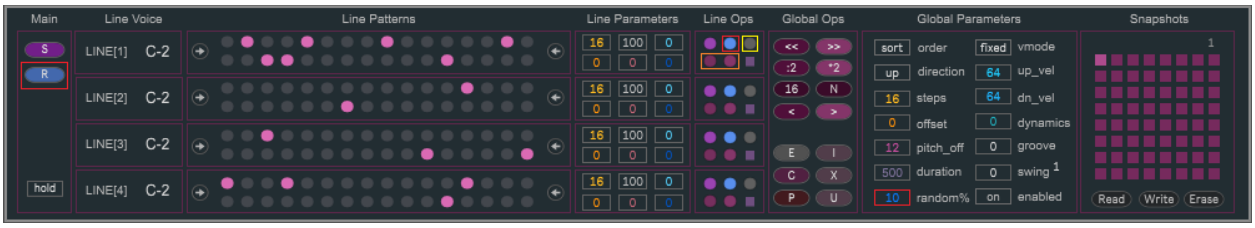
<u>Note</u>: The count of the first measure is started from the moment you modify the [Periodic Variation] number box.

<u>Tip</u>: A useful approach is to have periodic variations applied on different lines each with a different period.

Pattern banks can also be imported and exported from and to *pattern bank* files. A pattern bank is the collection of the 44 patterns (11 times 4 lines), for all the lines. See the section about [Pattern banks] file management further in this user manual.

Random generation and Line Pattern modifiers

Besides manually programming the triggers on the Line Pattern Area (and pattern mixing into combinations as previously introduced), also random generation is possible. On top of that, a few controls are available for pattern manipulation.



Line Parameters (each line): 16 100 0 / 0 0 0

Global Parameters:
sort order
up direction
16 steps
0 offset
12 pitch_off
500 duration
10 random%
fixed vmode
64 up_vel
64 dn_vel
0 dynamics
0 groove
0 swing 1
on enabled

The global random generator [**R**] (left side in the picture above, highlighted in red) randomizes the arpeggio patterns for *all* lines at once. The amount of randomness is defined by the [**Random%**] setting (right side on the Global Parameters section, also in red highlight). The highest the percentage the more triggers will be placed.

Besides this global function, the following modifiers *for each line* independently are available:

- [**Random**] (blue button) random pattern generation for a single line, similar to [R] but affecting only a single line.
- [**Erase**] (gray button) erases the line pattern, affecting a single line
- [**Shift Left**] (dark-purple/wine button) shifts the line pattern to the left, affecting a single line
- [**Shift Right**] (light-purple/magenta button) shifts the line pattern to the right, affecting a single line

Tip: an interesting way to create *syncopation* is to shift a single line while leaving the other lines intact.

Working with Velocity

There are various way to have velocity added to the resulting arpeggio. The way velocity is treated depends on the following global parameters settings (all specified in the Global Parameters section):

- [**vmode**] (Velocity Mode) defines the overall strategy for note velocity. When set to "fixed" it ignores the input velocity and causes the output velocity to be defined by the [up_vel] and [dn_vel] settings; when set to "input" it carries the velocity of the input MIDI notes into the triggers output MIDI notes velocity.
- [**up_vel**] & [**dn_vel**] define the MIDI velocity for the triggers of the lines when [vmode] is set to "fixed", respectively for the upper row and lower row.
- [**Velocity Dynamics**] Adds subtle random variations to triggers output velocity. This can be applied to either modes defined by the [vmode] settings.

Other global settings

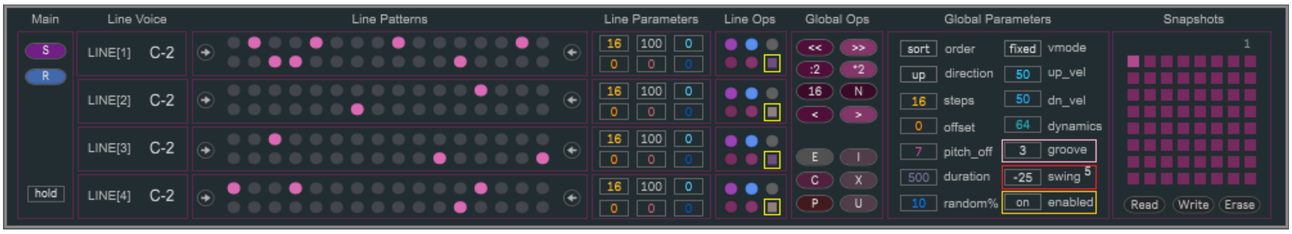
The following settings are also available in the Global Parameters section:

- [**Steps**] Active number of steps across all lines
- [**Offset**] Offsets all lines to the starting step
- [**Duration**] Defines the trigger duration

The settings concerning timing modifiers (Swing & Groove) are explained in the next section.

Manipulating Timing: [Swing]

The ARPline device provides controls for manipulating timing of the steps for rhythmical [Swing] and humanized [Groove] effects. These mechanisms affect the timing of the output triggers generated by the arpeggiator and work identically to other J74 devices, making possible to jam across devices in a consistent way.



sort order
up direction
16 steps
0 offset
7 pitch_off
500 duration
10 random%
fixed vmode
50 up_vel
50 dn_vel
64 dynamics
3 groove
-25 swing 5
on enabled

The [**Swing**] control (in red on the picture above) allows you to add swing to the arpeggio. Negative % values will increase the time between certain steps (hesitation), while positive % values will decrease it (anticipation). Small [Swing] values (typically below 8) will add just a bit of a cadence. Values of +/-50, +/-33, +/-25 will shift step timing to hit rhythmic "hot spots" (such as triplets or 1/32 notes), depending on the selected [Swing Type] (the little number besides the [Swing] control). Other values will result in some unusual syncopation.

The [**Swing Type**] defines the flavor of swing to be used, with 12 variations being available:

1 mpc classic swing
2 swing variation1
3 swing variation2
4 swing skip-through
5 triplets swing
6 break/remix1
7 break/remix2
8 break/remix3
9 syncopate1
10 syncopate2
11 syncopate3
12 chaos/reverse

Tip: the swing effect is quite flexible and works identically on all J74 devices, making possible to jam with multiple devices at the same time (e.g. if you map the same controls to a single MIDI controller knob). Also keep in mind that the nature of swing is to anticipate or delay certain hits in time and that a certain amount of delay or anticipation will move you into triplets or doubling very effectively. These values are typically the integer ratio's 1:4, 1:3 1:2 which correspond to swing values +/-25, +/-33, +/-50. These are "sweet spots" for many styles. Of course a given genre (and bpm speed) will respond differently to different settings, so you probably like to explore this area yourself. It is quite fun.

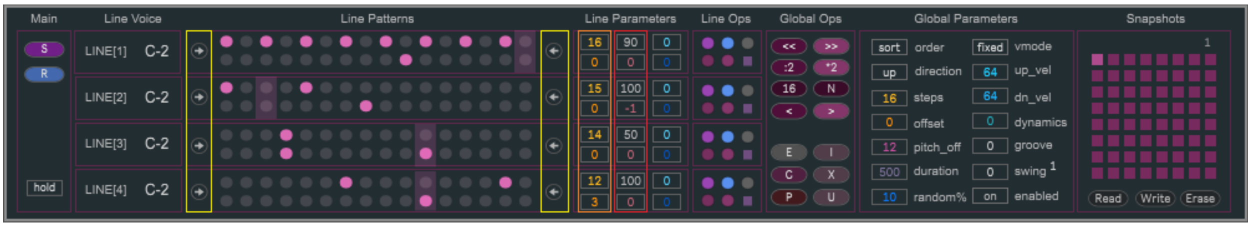
Additional controls (aimed to performance use) are the [**Global Swing Enabled**] toggle and the individual [**Line Swing Toggles**], highlighted in yellow on the picture above. The [Global Swing Enabled] toggle can be used to bypass the effect immediately, on all lines. The [Line Swing Enabled] toggle can be used to bypass the effect on a single line only.

Manipulating Timing: [Groove]

The [Groove] feature adds little random fluctuation to the trigger times, independently for each line. This gives a "human" feel to the generated arpeggio's (as it prevents simultaneous hits on different lines for instance when arpeggiating/triggering chords by having multiple lines triggering at the same step).

Creating Poly-Metric and Poly-rhythmic arpeggio's

A key aspect of the ARPline plugin is the independent treatment possible for each line in terms of steps, direction and speed. By combining these features poly-metric and poly-rhythmic arpeggio's are easy to create, resulting in almost never-ending evolutions.



sort order
up direction
16 steps
0 offset
12 pitch_off
500 duration
10 random%
fixed vmode
64 up_vel
64 dn_vel
0 dynamics
0 groove
0 swing
on enabled

Each line has the following independent controls:

- [**Line Direction**] (highlighted in yellow in the picture above). You can change the direction of play for each line independently. Changing direction of one line against the others completely changes the feel.
- [**Line Step Number**] (highlighted in orange in the picture). This defines the number of steps in the line and is a powerful tool for creating poly-metric effects.
- [**Line Step Offset**] defines the starting step in a line. This can be used to alter an arpeggio and create simple syncopation (steps coming with anticipation or hesitation). When combined with timing effects (like swing, as explained further on this manual) the effect is more emphasized.
- [**Line Probability**] Defines the probability of triggering steps (a value of 100 always triggers). This creates unpredictability within the arpeggio.
- [**Line Tempo**] Defines the speed using tempo dividers/multipliers, for each line independently. The following describes the way this parameter affects the speed of playback of a line in relation to the song tempo:

    - When set to zero (0) it conforms the line speed to the current song tempo
    - Negative even numbers (-2, -4, -8, etc.) result in integer divisions of the tempo (1:2, 1:4, 1:8 and so on);
    - Negative odd numbers (-1, -3, -5, etc.) create triplets divisions of the current tempo (3:4, 3:8, 3:16);
    - Positive even numbers (+2, +4, +8, etc.) increase (doubles) the line tempo (2, 4, 8 times and so on);
    - Positive odd numbers (+1, +3, +5, etc.) result in triplets multiples (6:4, 12:4, 12:8 and so on);

Used in combination these feature allow creation of poly-metric and poly-rhythmic effects across the lines.

Poly-metric arpeggio's can be achieved by having lines running on different [Line Step Number]. This create a beating effect which brings a much longer periodicity to the arpeggio than a single bar of 16 steps.

Poly-rhythmic arpeggio's can be achieved by having lines running at different [Line Tempo] speeds. This create a syncopated effect having steps at unusual times. Keep in mind that *odd* speeds work on *triplets* divisions or multiples so it may be musical to set the [Line Step Number] of the line *to a multiple of 3* (such as 3, 6, 9, 12, 15), instead of a multiple of 2 steps (like the default 16).

Pattern Banks: [Update], [Copy/Paste] and [Export/Import]

You can save and recall pattern banks from and to files. A pattern bank is the collection of all the 44 patterns (11 times 4), for all lines. Here the "rules" which apply to successfully make and recall changes to ARPline Pattern Banks:

- *By default the patterns in a bank are read-only* and changes you make while editing are not saved: if you make changes to a slot and then move to another slot, those changes will be lost. Same result if you reload the Live set, *even if the Liveset was saved* (only the last view will be saved).
- *Commit changes while editing:* The [**U**] or [**Update**] button will commit the current configuration in the buffer memory of the computer. This will not save the bank file, though. In other words, if you make changes *and* press [U], you can move to another slot and later come back to the same slot to find the changes you committed still there. The [U] action alone, though, will *not save the changes in the Live set or to a Pattern Bank file.*
- *Save a Pattern Bank file*: you can save changes to the bank by using the [**E**] or [**Export**] button. If you export the Pattern Bank as a file (eventually overriding another file or creating a new one) and *then* save the Live set, a future reload of the Live set will recall the saved *Pattern Bank file* (and therefore your changes will be associated with the liveset).
- *You can load a Pattern Bank* using the [**I**] or [**Import**] button. This will load the patterns in the slots.
- *You can copy a Pattern (all lines) in memory* using the [**C**] or [**Copy**] button.
- *You can paste a Pattern from memory to all lines* using the [**P**] or [**Paste**] button.
- *You can erase a Pattern (on all lines)* using the [**E**] or [**Erase**] button.

Pattern Bank File names with BPM embedded

There is a handy little feature associated with Pattern Banks you should also be aware of: if you save a Pattern Bank with a name including the part "- #<bpm_value>bpm" in the name of the Pattern Bank file (with <bpm_value> an integer between 20 and 480 - what the tempo speed of your liveset should be), you can save the BPM and return to it automatically when the Pattern Bank is recalled ([I] button). An example of a valid name with BPM embedded: "Arpeggio-003-#105bpm.txt". Opening this Pattern Bank will set Ableton Live's tempo to 105 BPM.

Snapshots

Besides storage for pattern banks the device also provides its own snapshot save/recall/storage functionality. This way you can make a global snapshot of *all* the settings and recall it at any time, even during a performance.

How it operates:

- **SHIFT + CLICK** on a slot creates a snapshot (or replaces an existing one)
- **CLICK** on a slot recalls a stored snapshot
- **SHIFT + {ALT|OPTION} + CLICK** on a slot deletes a stored snapshot slot
- Snapshots banks can be exported using the [**Write**] operation.
- Snapshots banks can be imported back using the [**Read**] operation.
- All Snapshots from the current bank can be erased using the [**Erase**] operation.
- On the top/right corner you can also find a MIDI mappable [**Snapshot Number**] for recalling snapshots

Note: MIDI program change messages can be used to recall snapshots. This is a powerful way to add automation to snapshot recalling. A useful approach for this is to add the program number directly on a MIDI clip in the same track (see Launch section of MIDI clips) so that if the clip is triggered the selected snapshot is recalled.

Important: the file extension for snapshot files MUST be ".maxpresets" in order for the files to be readable.

Questions?

If you still have questions or issues, please send an email to: info [at] fabriziopoce.com