

BLUE ARP

Operation Manual

corresponds to BlueARP v2.16



Pattern Arpeggiator / Step Sequencer

VST/AU plug-in for Windows & OSX (32 & 64 bit)

by Oleg Mikheev aka GrayWolf

<http://www.graywolf2004.net>

October 2015

Table of Contents

INTRODUCTION.....	3
SETTING UP BLUEARP IN SOME DAWs	4
FL STUDIO(FRUITY WRAPPER METHOD).....	4
FL STUDIO(PATCHER METHOD)	5
ABLETON LIVE.....	6
REAPER	6
SIGNAL FLOW.....	7
INTERFACE	9
MAIN WINDOW LAYOUT	9
BLOCK (1): TOP PANEL	10
BLOCK (2): LEFT PANEL.....	11
<i>Input Filter parameters.....</i>	<i>11</i>
<i>Arp Engine parameters.....</i>	<i>14</i>
<i>Output Filter parameters.....</i>	<i>16</i>
<i>Program Chains parameters.....</i>	<i>16</i>
BLOCK (3): PATCH BROWSER	17
BLOCK (4): MAIN MENU AND PATTERN CONTROLS	17
BLOCK (5): MATRIX EDITOR.....	19
BLOCK (6): VALUE BARS	19
BLOCK (7): PROGRAM CHAINS	21
BLOCK (8): INFORMATION PANEL	21
LINKS.....	22

Introduction

BlueARP is a programmable pattern arpeggiator / step sequencer, it comes as a VST or AU (Audio Unit) plugin for MAC OSX and Windows (both 32 and 64 bit). BlueARP is a pure MIDI plugin, it doesn't generate any sound by itself but transforms MIDI messages, so it has to be routed to either software or hardware synth in any VST/AU-enabled DAW like FL Studio, Ableton Live, Cubase, Reaper, Logic Pro, etc.

Basically you need to program some pattern in BlueARP, then you play some chords and BlueARP will generate melodic phrases from this chords, according to the pattern you programmed.

BlueARP was designed for electronic music genres (like trance, house, etc.), but it also may have some unexpected applications like triggering drums, since it has swing feature.

Compatibility info

Format: VST plugin 32-bit, VST plugin 64-bit, AU (Audio Unit)
OS: OS X (tested on 10.6.8), Windows XP and higher

Features

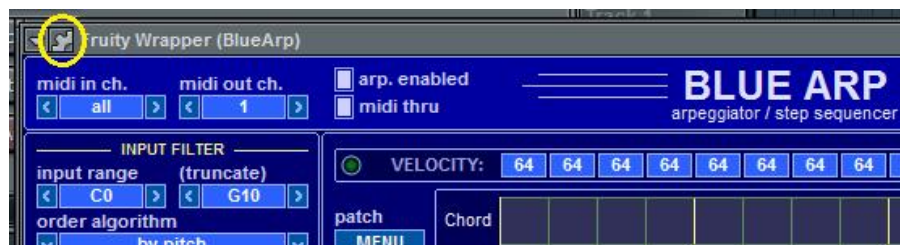
- Up to 64 steps per pattern;
- Up to 64 programs per bank;
- 'Pattern chainer' feature to switch patterns on the fly
- Comes with 64 factory patterns to start with;
- Intuitive matrix editor to program patterns quickly;
- Almost all controls are automation-enabled
- Up to 5 input keys;
- Real-time input quantization (input quantize setting, pattern restart on beat / key);
- Input range setting for keyboard-split performances;
- Separate settings for octave and semitone per step transpose;
- Configurable color schemes;

Setting up BlueARP in some DAWs

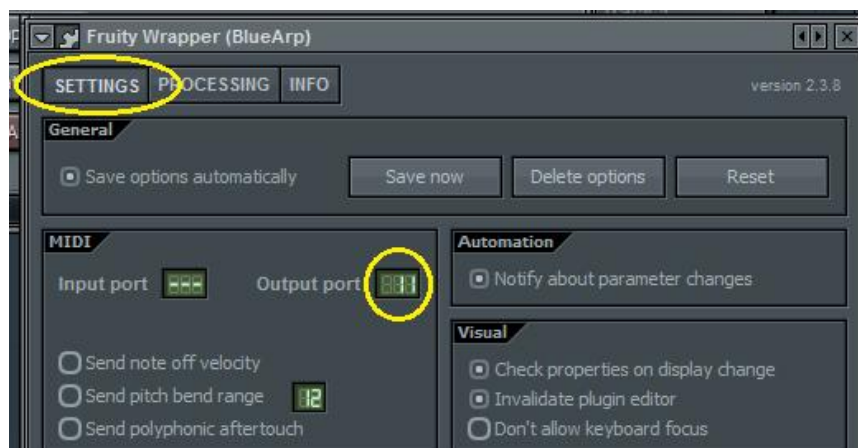
If your DAW is not present in this list, refer to other VST arpeggiator manuals like Kirnu Cream, Catanya, Nora or search for tutorials in google or youtube. For BlueARP procedure should be nearly the same as for any other VST arpeggiator.

FL Studio(Fruity Wrapper method)

Load BlueARP, click «Wrapping settings» button:



Click «SETTINGS» tab, set «Output port» to any value, not occupied by hardware MIDI devices, memorize this value:

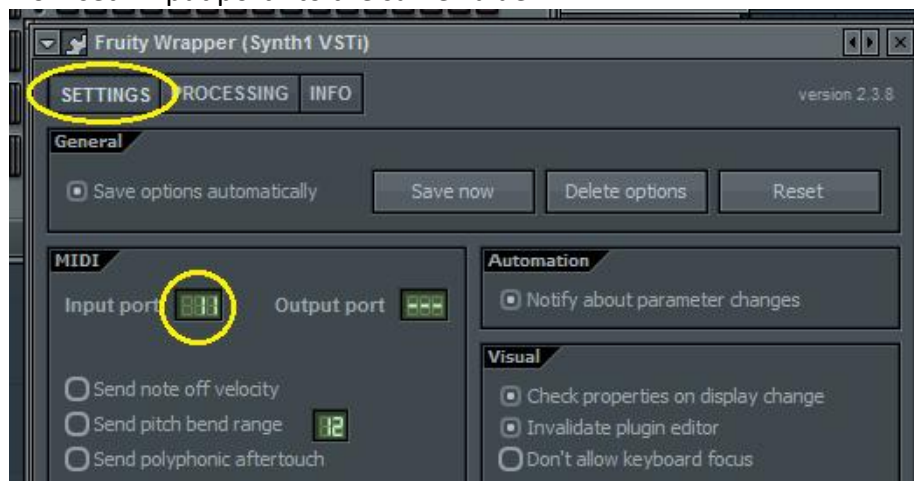


Click «Wrapping settings» button again to return to main plugin's window.

Go to Fruity Wrapper settings of VST synth (Synth1 in our example)



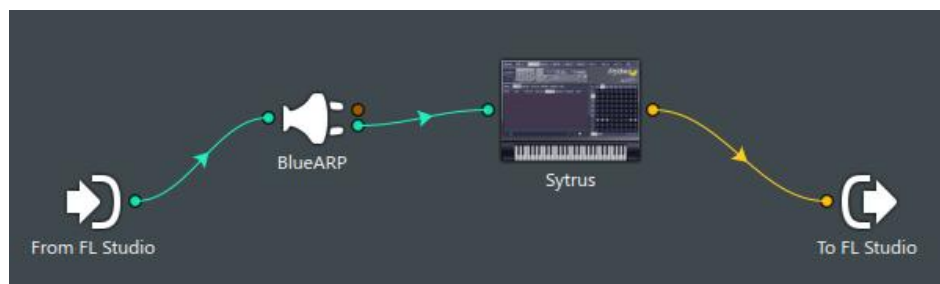
Now set «Input port» to the same value:



Now Synth1 will receive MIDI events generated by BlueARP.

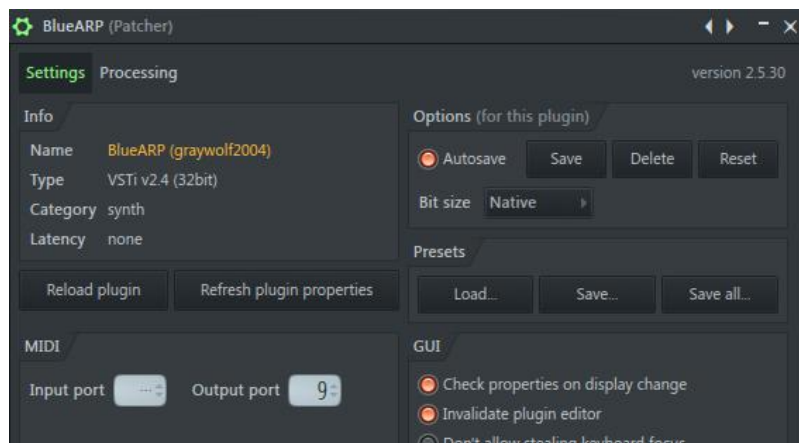
FL Studio(Patcher method)

Add "Patcher" to a track, inside patcher add BlueARP and any Fruity Generator, then connect them as follows:



(green arrows represent MIDI signal flow, yellow - audio signal).

Double click BlueARP, go to wrapping settings and set output port to any unused number.



Ableton Live

Ableton is tricky when it comes to MIDI plugins. There are 2 options.

Option 1. Load BlueARP on one track, synth you want to control (say Synth1 VST) on another. For Synth1 track, set MIDI From = BlueARP (both checkboxes). For BlueARP track, set Monitor = In. There's an issue - BlueARP will pick up MIDI from clips only when Monitor set to 'Auto', but it takes notes from Keyboard only with Monitor = In. So you have to constantly switch monitor from In to Auto. If you want to avoid it, go for Option 2.

Option 2. Create a separate track (say MIDI_for_BlueARP) which will hold your MIDI clips. Then insert BlueARP and your synth (you have 3 tracks in total)



For track "**MIDI_for_BlueARP**", Monitor should be Auto.

For "**BlueARP**" track, set MIDI From = "MIDI_for_BlueARP" (your MIDI track), Monitor = In.

For **Synth** track, set MIDI From = BlueARP (both checkboxes!), Monitor = Auto.

Now, use "MIDI_for_BlueARP" track to record patterns and "BlueARP" track to play live.

If you want to sequence external synthesizer (connected via MIDI), use "External instrument" (it's in "Live Devices" list).

REAPER

On an instrument track, click "Input FX" (expand your track if this button doesn't appear)

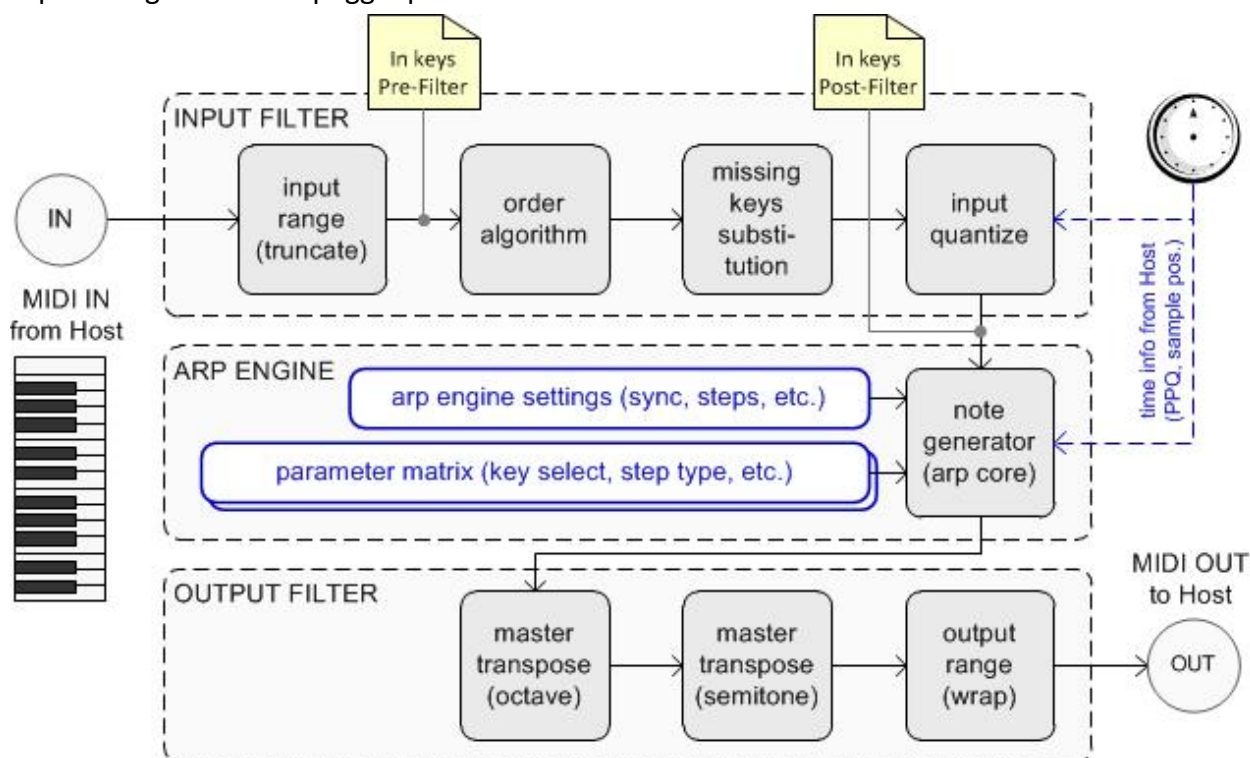


Plugin selector will open, select "BlueARP", that's it. Input FX button will become green.

Signal flow

To use BlueARP to the maximum, it's valuable to have a concept about its structure and signal processing logic.

The picture below is a simplified data flow diagram. At the input we have MIDI notes coming from host. In other words, events of pressing or releasing keys on MIDI keyboard, or events coming from MIDI track. At the output we have the same type of events (MIDI notes), but here they represent generated arpeggio pattern.



pic. 1. BlueARP processing diagram.

Main blocks are «Input Filter», «Arp Engine» and «Output Filter».

In this manual, «keys» are actually pressed notes on the keyboard, while generated «notes» come from arpeggiator output.

Input Filter receives MIDI events from Host – basically it's key press and release events (also it may be pitch bend, aftertouch and controller messages). From these events, we generate *Key List* – an ordered list of keys with corresponding velocities (velocity represents how hard did you press a key).

«*In keys Pre-Filter*» is a key list as it comes from Host (keys are ordered as they were pressed). «*In keys Post-Filter*» represents the same key list after ordering, missing keys substitution and real-time quantization (for further details on these settings, go to page 10).

You can see what's currently in both key lists on the Information panel (at the bottom):

Pos: -	Beat: 6.0	Step: 7.0	In keys pre-filter: A2, C3, E3, F3, -	Note out: A1
P011:Cur_Chain = 2 / 32			In keys post-filter: A2, C3, E3, F3, A2	Detected chord: F maj7

See Information panel description for more details, page 21.

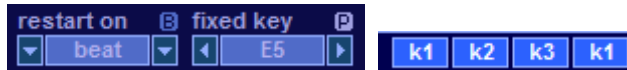
Arp Engine generates pattern notes according to post-filter key list, which comes out of "Input Filter block", also referring to Value bars (matrix editor), which contain pattern information for each step. For example, «key select» setting determines which key to take from the list for the current step (k1 – key 1, k2 – key 2, fix – fixed key, etc.). «Step type» value tells whether the step is a normal note (Nrm), the rest of the previous step's note (Rst) or doesn't generate any note (Off). Refer to page 13 for more information about Value bars and Matrix editor.






BlueARP has «*missing keys substitution*» feature you won't find in other arp's. It means when you have say 4-keys pattern and play only 2 key chord you can select if you want steps for keys 3 and 4 to be silent or to be substituted with existing keys (there are several substitution algorithms, see page 10 for details).

Output Filter adds some post-processing to generated notes – octave / semitone transposition and wrapping notes to fit the given range.

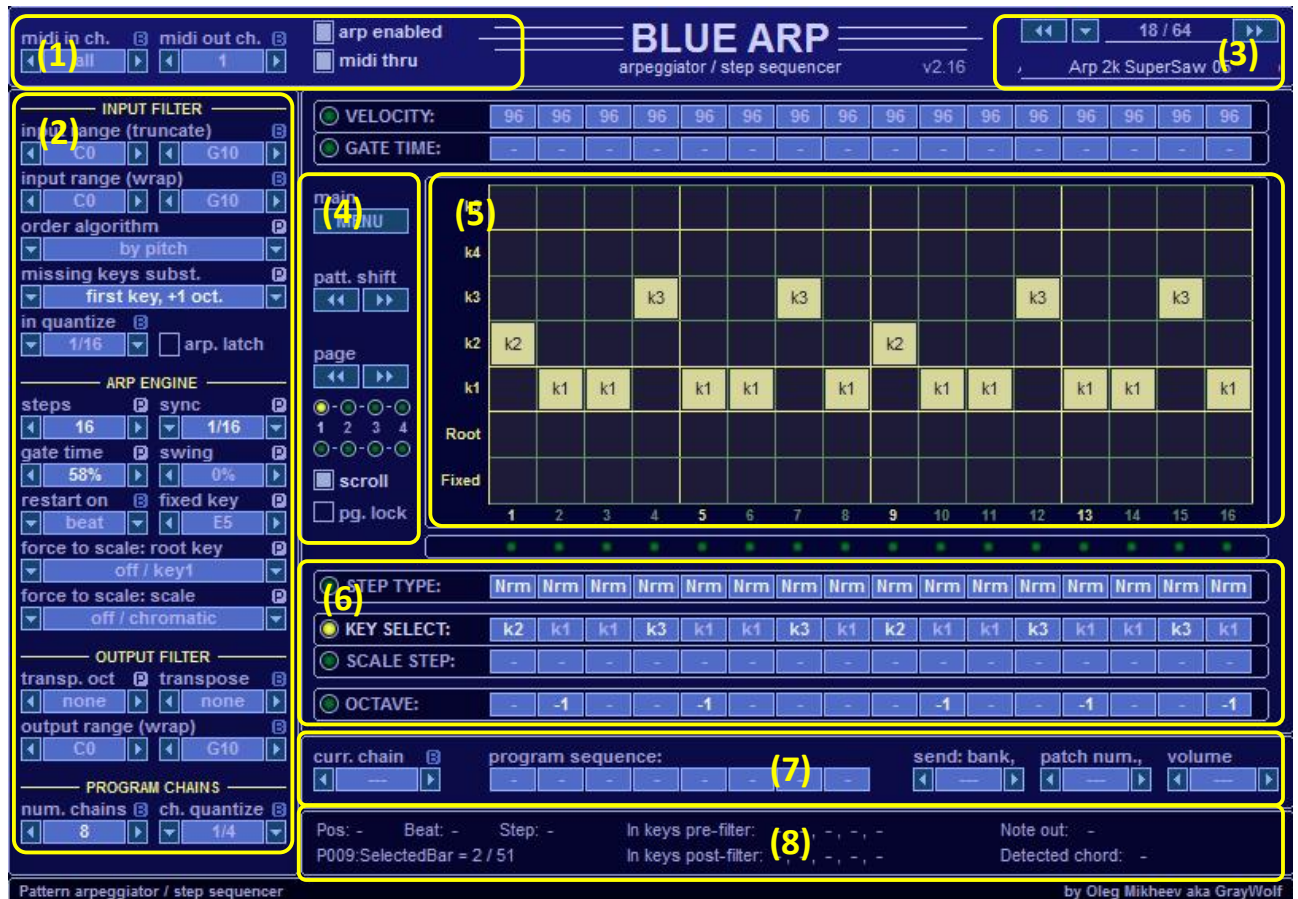
Interface

The main GUI element is a «value box», either surrounded by buttons or not:

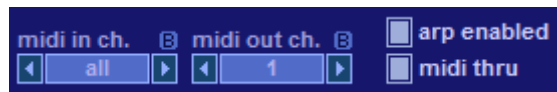


To adjust the value, drag it up or down. For controls with buttons, arrow buttons   adjust the value, «down arrow» buttons  show the value list menu.  or  mark tells whether this setting is saved with a bank (B) or program (P).

Main window layout



- (1) **Top panel global settings** (MIDI In channel, MIDI Out channel, etc.), they are saved for all programs in the current bank. When you switch programs, these settings remain the same;
- (2) **Left panel settings** represent all the step-independent arpeggiator settings like number of steps, synchronization, key sort order etc. They are saved with the current program;
- (3) **Patch browser** allows to scroll through programs and to rename them. There are 64 programs available in a bank;
- (4) **Main menu and pattern controls** - cyclic pattern shift (left and right), page selector (for patterns longer than 16 steps);
- (5) **Matrix editor** is a key element of BlueARP. It represents step-related values for the selected bar (like KEY SELECT, STEP TYPE, etc.);
- (6) **Value bars** represent step-dependent pattern parameters. To adjust the value, drag the «value box» up or down;
- (7) **Program chains** - allows to chain several programs (patterns) into a continuous sequence;
- (8) **Info panel**. Most important things here are «In keys pre-filter» and «In keys post-filter» lists;

Block (1): Top panel

Top panel contains global settings (MIDI In channel, MIDI Out channel, etc.), they are saved for all programs in the current bank. It means after you

changed say midi channel, it will stay the same when you switch to another program.

midi in ch	input MIDI channel
values	<i>all, 1 .. 16</i>
comments	When set to 1 .. 16, BlueARP will take input notes only from specified MIDI channel. You may need this if you have several MIDI keyboards connected and you want to control several instances of BlueARP with different keyboards.
midi out ch	output MIDI channel
values	<i>1 .. 16</i>
comments	Default setting is 1, cause soft synths usually don't care about MIDI channel. You may need it if you have multithimbral hardware synth connected to BlueARP or several hardware synths chained on one MIDI output port
arp enabled	Arpeggiator is turned On
values	<i>On/Off (checkbox)</i>
comments	When Off, BlueARP behavior depends on «midi thru» setting. When «midi thru» is On, it will pass input notes to the output unchanged (but input and output range will work anyway). When MIDI thru is Off, BlueARP will be completely off. You may want to automate «arp. enabled» setting to switch certain arps on and off during the performance.
midi thru	Pass notes thru when arp is disabled.
values	<i>On/Off (checkbox)</i>
comments	Set this to On and "arp enabled" to Off if you want to use BlueARP as a midi key range filter, for example in a keyboard-split performance

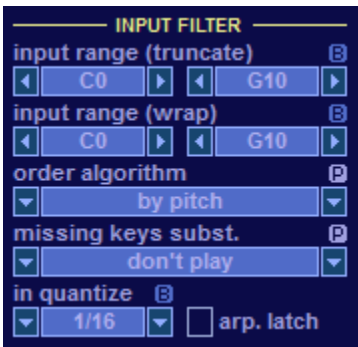
Block (2): Left panel

Left panel settings are divided into 4 blocks – «Input filter», «Arp engine», «Output filter» and «Pattern chains». It corresponds to signal flow path (see diagram on page 6).

In this manual, «keys» are actually what's pressed on the keyboard, while generated «notes» come from arpeggiator output.

In general, left panel represents all program settings, except the pattern itself. Some of them are program-related (P-mark), some are bank-related (B-mark).

Input Filter parameters



Input filter is responsible for operations with the key list before it enters the arpeggiator «core» engine. Current key list before and after the filter is shown on the Information panel (see page 15)

input range (truncate)	range for filtering out input notes
values	C0 .. G10 (MIDI notes 0 .. 127)
comments	Change it if you want this instance of BlueARP to react to MIDI keys only within a given range. All notes outside this range will be ignored. You will need this if you want to create keyboard-split performance with several instances of BlueARP.

Hint. Right-click value box and select "press MIDI key..." to set a value from your MIDI keyboard



input range (wrap)	range for wrapping input notes
values	C0 .. G10 (MIDI notes 0 .. 127)
comments	Unlike "input range (truncate)", this one won't ignore notes outside the range, but fill fit it to the range applying up or down octave transposition. Say your set this range to A2...A3. When you press keys F2 , A2, E2, A3, E3 , the output of this "note wrapper" will be F3 , A2, E2, A3, E2 (bold notes were wrapped)

into a range A2...A3).

It's sonically useful when you play chords all over the keyboard but want to your bass line sounding right, not too low or too high.

order algorithm	ordering (sorting) algorithm for input keys														
values	<i>by pitch, by pitch desc, as played, as played desc, by velocity, by velocity desc</i>														
comments	Default setting is «by pitch», which means that pressed keys come into arp engine in natural order (from left to right on the keyboard). It also means «k1» in «KEY SELECT» bar will also be the lowest key. Sometimes it's not the best way to order pressed keys. For example, if you play 1-key bass line, it's better to set order algorithm to «as played, desc». In this case «k1» will always be the last pressed key.														
missing keys subst.	missing keys substitution algorithm														
values	<i>don't play, cyclic, first key, last key, fixed key, (+1 oct/-1 oct variations)</i>														
comments	<p>When your pattern has more keys than you actually play, this setting will determine whether to ignore these steps (don't play) or substitute missing keys with existing ones.</p> <p>For example, you hold keys C5 and E5, while your pattern has keys «k1», «k2», «k3» and «k4». Order algorithm is set to «by pitch», so input key list before substitution is «C5, E5, -, -, -». Here's input key list after substitution for several settings:</p> <table> <tr> <td><i>cyclic</i></td><td>«C5, E5, C5, E5, C5»</td></tr> <tr> <td><i>cyclic, +1 oct</i></td><td>«C5, E5, C6, E6, C6»</td></tr> <tr> <td><i>first key</i></td><td>«C5, E5, C5, C5, C5»</td></tr> <tr> <td><i>first key, -1 oct</i></td><td>«C5, E5, C4, C4, C4»</td></tr> <tr> <td><i>last key</i></td><td>«C5, E5, E5, E5, E5»</td></tr> <tr> <td><i>last key, +1 oct</i></td><td>«C5, E5, E6, E6, E6»</td></tr> <tr> <td><i>fixed key</i></td><td>«C5, E5, G5, G5, G5» («fixed key» = G5)</td></tr> </table>	<i>cyclic</i>	«C5, E5, C5, E5, C5»	<i>cyclic, +1 oct</i>	«C5, E5, C6, E6, C6»	<i>first key</i>	«C5, E5, C5, C5, C5»	<i>first key, -1 oct</i>	«C5, E5, C4, C4, C4»	<i>last key</i>	«C5, E5, E5, E5, E5»	<i>last key, +1 oct</i>	«C5, E5, E6, E6, E6»	<i>fixed key</i>	«C5, E5, G5, G5, G5» («fixed key» = G5)
<i>cyclic</i>	«C5, E5, C5, E5, C5»														
<i>cyclic, +1 oct</i>	«C5, E5, C6, E6, C6»														
<i>first key</i>	«C5, E5, C5, C5, C5»														
<i>first key, -1 oct</i>	«C5, E5, C4, C4, C4»														
<i>last key</i>	«C5, E5, E5, E5, E5»														
<i>last key, +1 oct</i>	«C5, E5, E6, E6, E6»														
<i>fixed key</i>	«C5, E5, G5, G5, G5» («fixed key» = G5)														

in quantize	input keys quantization
values	<i>none, 1/16, 1/12, 1/8, 1/6, 1/4, 1/2, 1 bar, 2 bars</i>
Comments	Quantization for input keys, as a fractions of a bar (1/16 means 16th notes, 1/4 corresponds to 1 beat). For example, you set sync to 1/4 - in this case BlueARP will capture pressed keys on the start of each beat.

Hint. When input quantize is on, you should press keys a little beforehand, since BlueARP needs to capture input keys before next step/beat starts.

arp. latch	Latch (or hold) pattern
values	<i>On, Off (checkbox)</i>
comments	When checked, BlueARP will continue to play pattern for the last pressed chord even after all input keys are released, until another key is pressed. For live performances it may be useful to assign "arp.latch" to sustain pedal.

Arp Engine parameters

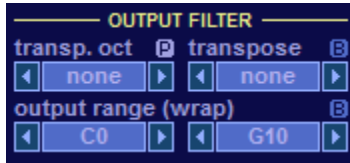


Arp Engine takes note list from the input filter (after fit-to-range, quantize, etc.) and generates note pattern at the output, referring to MIDI clock and current PPQ (position in a song or pattern).

steps	Number of steps for current program
values	1 .. 64
comments	Default value is 16. You may also experiment with irregular values like 15 or 17, it will make the pattern sound less predictable which is sometimes sonically useful. <i>Hint: All 64 steps of a pattern are stored within a program. So if you decrease number of steps say from 32 to 16, then save a program and reload it, you won't lose information for these extra "hidden" steps.</i>
sync	Step length (as a fraction of a bar)
values	1/48, 1/32, 1/24, 1/16, 1/12, 1/8, 1/6, 1/4, 3/64, 3/32, 3/16, 3/8
comments	Default value is 1/16, it means 1 step = 16th note. 1/12 is 8th triplets or 16th dotted. 3/n values may give some nice results (try 3/16 with all steps set to «Chord» to get some deadmau5 style stuff).
gate time	Note length (relative to step)
values	1% .. 100%
comments	Sets note length as a fraction of step length.
swing	Swing control
values	-50% .. 50%
comments	Sets relative time shift for even steps as a fraction of step length (assuming step numbers start from 1). For example, swing = 33% means that each even step will be delayed for 33% of step length (with negative values, it will start earlier).

restart on	Pattern restart trigger
values	<i>beat, key, 1st key</i>
comments	<p>Default value is «beat», it means that step number is always aligned to PPQ (song position) from host. When your song or pattern restarts in DAW, BlueARP pattern will also restart. With «key» setting, BlueARP will restart pattern each time new key/chord is pressed (after all keys were released).</p> <p>With "1st key" pattern will start with first chord pressed and will keep going until you restart playback.</p>
fixed key	Fixed key value
values	<i>C0 .. G10 (MIDI notes 0 .. 127)</i>
comments	In «KEY SELECT» bar, you can set any step to «Fixed», in this case it won't depend on pressed keys. You can also set all steps to «Fixed» if you want to use BlueARP as a step sequencer.
force to scale: root key	Sets root key for "force to scale mode". Works together with "force to scale: scale" parameter
values	<i>"off/key1, detect from chord, C, C#, D ... Bb, B"</i>
comments	<p>You can either set a fixed root for a scale you want to fit notes in, or detect it dynamically from the chord you play. BlueARP recognizes chord inversions, so if you have selected "detect from chord" and pressed (E4, A4, C5 - Am inverted), your root key will be A.</p>
force to scale: scale	Sets scale key for "force to scale mode". Works together with "force to scale: root key" parameter
values	<i>"off/chromatic, detect from chord, Major, Majpr pentatonic, minor, minor harmonic, minor melodic, minor pentatonic"</i>
comments	<p>If you set anything except "off/chromatic", two things will happen:</p> <ol style="list-style-type: none"> 1. BlueARP fill fit all output notes to the given scale; 2. "SCALE STEP" value bar will transpose notes in scale steps. Say if your scale is C Major, you pressed D4 and scale step=+1, the output note will be E4. <p>With "off/chromatic" selected, "SCALE STEP" will work as a semitone transposition.</p> <p>With "detect from chord" selected, BlueARP will derive scale from a chord you play. It's not very smart, but at least it will give "Major/minor" scales for Major/minor chords.</p>

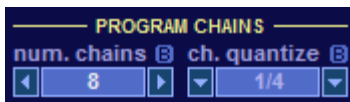
Output Filter parameters



Output filter adds additional processing to generated notes – octave / semitone transposition and wrapping notes to fit a given range.

transp. oct	Output transpose, octaves
values	-3 oct .. +3 oct
comments	Transpose is applied to output notes
transpose	Output transpose, semitones
values	-12 .. +12
comments	Transpose is applied to output notes
output range (wrap)	Range for output notes wrapping (2 values - min and max)
values	C0 .. G10 (MIDI notes 0 .. 127)
comments	Notes outside the range will be wrapped (transposed up or down an octave to fit the range). Say your output range is C5..C6 and generated note is D3 – it will be transposed to D5.

Program Chains parameters



Relates to "**Block (7) Program chains**" panel. "num.chains" sets maximum number of chains, 16 is maximum.

num.chains	Sets maximum value for "current chain" parameter
values	1 .. 16
comments	To switch chains with midi controller, you need to automate "current chain" parameter. If you use a knob for this, setting "num.chains" to the right value will utilize full rotation range of this knob.
ch. quantize	Input quantization for chain switch event
values	1/48, 1/32, 1/24, 1/16, 1/12, 1/8, 1/6, 1/4, 3/64, 3/32, 3/16, 3/8
comments	Input quantization for chain switching. When you switch chains, for better sound it should be done strictly and the start of new beat. Chain quantize = 1/4 does exactly that, it's the default setting. You may also quantize it to the whole bar.

Block (3): Patch browser



Left and right buttons switch to previous or next program in a current bank.

Bank contains 64 programs, so you can configure up to 64 arpeggiator patterns (and it will be saved with your project file). To change current program name, click on it, type in new name and hit enter or click somewhere outside on the top panel.

Block (4): Main menu and pattern controls



Main menu includes Bank load/save, Program load/save and some other functions (see picture below for details).

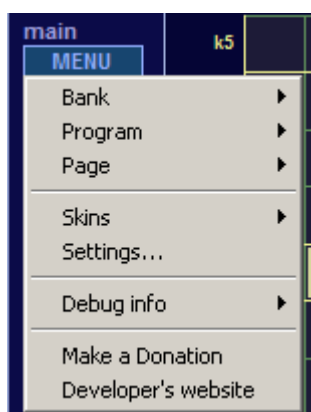
Pattern shift buttons perform cyclic 1 step shifting. It's useful, when your pattern doesn't match the beat and you want to align it. Since the shift is cyclic, you can shift it right 16 times for 16-step pattern and it will make no change.

Page select buttons are necessary when you pattern is longer than 16 steps and doesn't fit single screen (16 steps). Check «scroll» if you want BlueARP to automatically switch pages while pattern is playing. There are 2 small LED bars underneath, first one shows selected page, second one – page being played.

scroll checkbox - when checked, matrix will always show the page actually playing (for patterns longer than 16 steps).

pg. lock checkbox - when checked, current page will cycle over and over until unchecked (useful for programming long patterns).

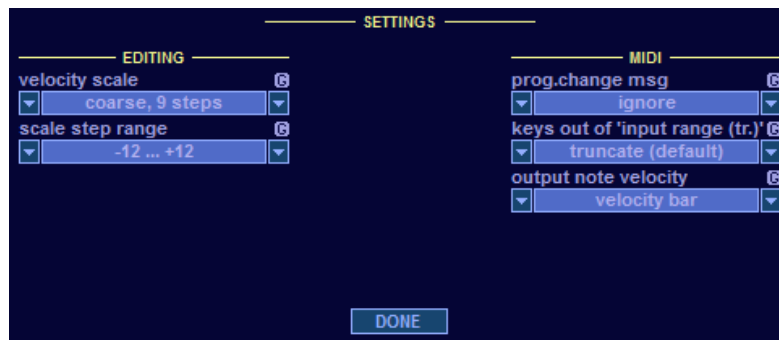
Main menu includes:



Skins – select color theme. skins are stored as *.ini files in “skins” directory. Default blue color scheme is built in. It's possible to create your own scheme by editing ini file.

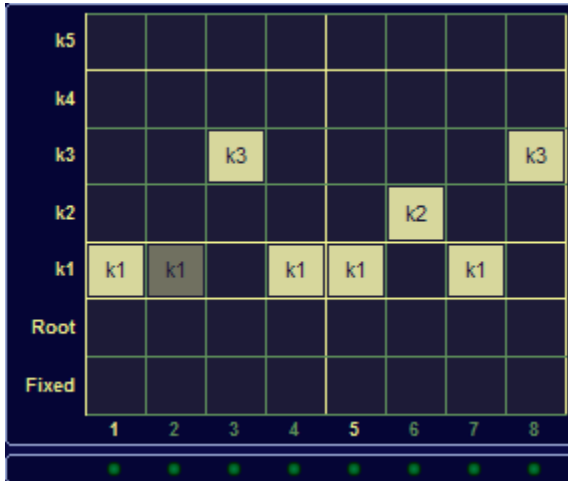
Bank, Program - load, save, initialize bank or program.

Settings – global settings (see below)

"Settings..." screen

velocity scale	Sets velocity accuracy for "VELOCITY" value bar
values	<i>"coarse, 9 steps", "fine, 128 steps"</i>
comments	Select "fine, 128 steps" if you want to do fine velocity adjustments, otherwise it will go like 16, 32, 48, etc.
scale step range	Sets range (min and max) for "SCALE STEP" value bar
values	<i>"-12...+12", "0...+12", "-7...+7", "0...+7"</i>
comments	Default value is "-12...+12". For touch-screens it may be better to set "-7...+7" or "0...+7" for better accuracy.
prog.change msg	Sets how to respond to incoming Program Change MIDI message
values	<i>"ignore", "set program", "pass to synth"</i>
comments	"Set program" - BlueARP will set its internal program in response to Program CC message. "Pass to synth" - BlueARP will do nothing, but will pass this message to its MIDI out (= to VST plugin it's connected to).
keys out of input range (tr)	Sets how to treat keys outside the range <i>"input range (truncate)"</i>
values	<i>"truncate (default)", "pass thru (no arping)"</i>
comments	"truncate" - keys outside the range will be ignored, "pass thru" - keys will be passed as-is, non-arpeggiated.
output note velocity	Sets where to take velocity for generated notes
values	<i>"velocity bar", "input key"</i>
comments	

Block (5): Matrix editor



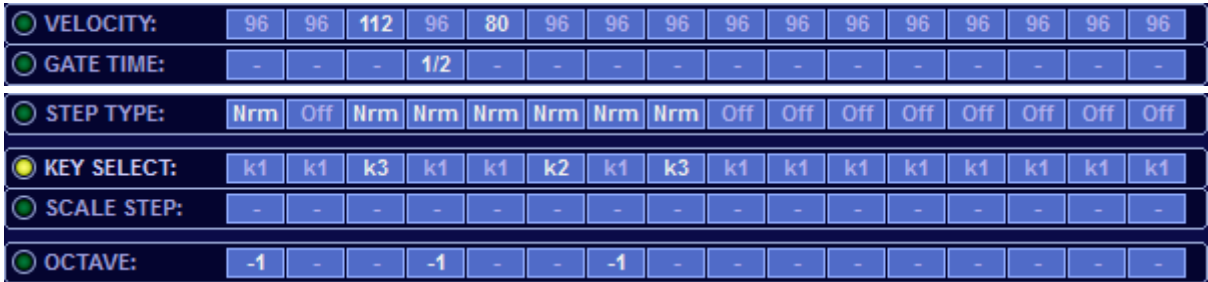
Matrix editor allows to edit selected "value bar" values in a more-friendly graphic way.

So, you can adjust a step-related value 2 ways – either in matrix editor or in value bar itself (see next chapter).

Click matrix cell to set the value. You can also drag the mouse from left to right to quickly set all the steps to a certain value.

Grayed out bricks mean that this particular setting doesn't affect generated pattern. On the picture above, steps 6 and 7 are set to Off, so «key select» value doesn't make any difference. Step 2 is set to «Off», so it will not generate any note.

Block (6): Value bars



Value bars represent step-related patters parameters. Selected value bar is also shown in Matrix editor.

To adjust value for a certain step, click on it and drag it up or down to increase / decrease the value. See description for each value bar below.

VELOCITY	Velocity value for each step
values	0, 16, 32 .. 127;
comments	Default value is 64. Use it to set accent for certain steps. VELOCITY values will be ignored, if you set "output note velocity" = "input key" in MENU >> Settings.
GATE TIME	Gate time multiplier for each step
values	1/16;1/8;1/4;1/2;-;2x;4x;8x;16x;
comments	Multiplies gate time by a given value. "-" means no change (default value). For example gate = 60%, GATE TIME for a step is "2x". It means note length for this step will be 60% * 2 = 120% or 1,2 steps.

STEP TYPE	Several options for output note generation
values	<i>Off</i> – this step doesn't generate any note <i>Nrm</i> – Normal(default) – generates a note; <i>Rst</i> – this step will play the Rest of the previous step; <i>Tie</i> – this note will overlap with the previous one (for glides);
comments	<p>«Rst» step simply means that this step continues to play the note from the previous step. You may chain several «Rst» steps together.</p> <p>«Tie» option may be tricky and not self-describing. It's main purpose is to create «glides» between notes. But it requires to configure synth properly – set it to monophonic mode, with legato and portamento on. In this case, when you press keys with overlapping (like press key1, press key2, release key1), sound pitch will glide between the notes, but not when you press and release them (like press key1, release key 1, press key2, release key1). When you configure the synth this way, «Tie» steps should create glides between notes.</p>
KEY SELECT	Input key selection for the given step
values	<i>Fixed</i> – use fixed key from Arp Engine settings <i>k1..k5</i> – take key №1 .. №5 from key list (post-filter) <i>Chord</i> – take all keys from key list
comments	<p>Determines which key to take from post-filter key list for the current step.</p> <p><i>Hint: Fixed key doesn't depend on pressed keys, so you can set all steps to fixed to use BlueARP as a step sequencer, or set some steps to fixed to create some interesting variations.</i></p>
SCALE STEP	Semitone/Scale step transposition for each step
values	-12, .. +12;
comments	Depends on "force to scale: scale" parameter. When the latter is "off/chromatic", will work as a semitone transposition (in versions up to 2.11). Otherwise, will transpose with respect to the selected scale.
OCTAVE	Octave transposition for each step
values	-2, -1, 0, +1, +2;
comments	It's convenient for bass lines, where the steps are usually transposed for the whole octave.

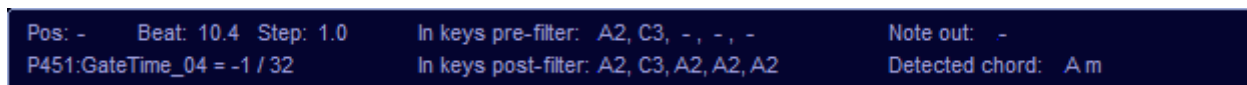
Block (7): Program chains



Program chainer is a tool to stack several programs (patterns) together into a longer "super-pattern". **Program sequence** holds numbers of chained programs for a given chain (**current chain** param).

You can also send **bank\patch** change and **volume** MIDI messages when you switch a chain. It's designed for live performance - you switch a chain and program on your synth is switched automatically.

Block (8): Information panel



Shows current beat, step and some other information.

Most important are «**In keys pre-filter**» and «**In keys post-filter**» - they represent Key List before and after input filter (ordering, missing key substitution), see diagram on page 6 for more details.

Links

Developer's website:

<http://www.graywolf2004.net/>

BlueARP discussion thread at KVR Audio forums (latest updates, news):

<http://www.kvraudio.com/forum/viewtopic.php?p=5080757>

Video demonstrations and tutorials are available on developer's YouTube channel:

<http://www.youtube.com/user/graywolf2004ru?feature=watch>

Please write bug reports and suggestions to KVR audio thread or email me at

graywolf2004@mail.ru

graywolf2004@gmail.com



Oleg Mikheev aka Graywolf.