BLUE ARP

Operation Manual

Corresponds to BlueARP v2.2.2

Pattern Arpeggiator / Step Sequencer

VST/AU midiFX plug-in for Windows & OSX, 32 & 64 bit

by Oleg Mikheev aka Graywolf

http://www.graywolf2004.net

April 2016

Table of Contents

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 turn these chords into melodic phrases according to 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

Formats:	VST plugin 32-bit, VST plugin 64-bit, AU MIDI-FX (Audio Unit for Logic Pro X)
OS:	OS X (tested on 10.6.8), Windows XP and higher

Features

- Up to 64 steps per pattern;
- Up to 128 programs per bank;
- 'Pattern chains' 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 range setting for keyboard-split performances;
- Separate settings for octave and semitone per step transpose;
- Configurable color schemes;

To get the idea what can be done with BlueARP, check these videos:

https://www.youtube.com/watch?v=1KOGVuElrhY

https://www.youtube.com/watch?v=retDsYjPokA

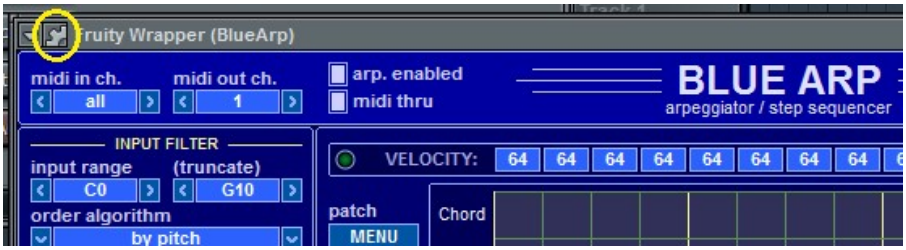
These are live performances using BlueARP with FL Studio, Ableton and many other DAWs can do the same.

Setting up BlueARP in some DAWs

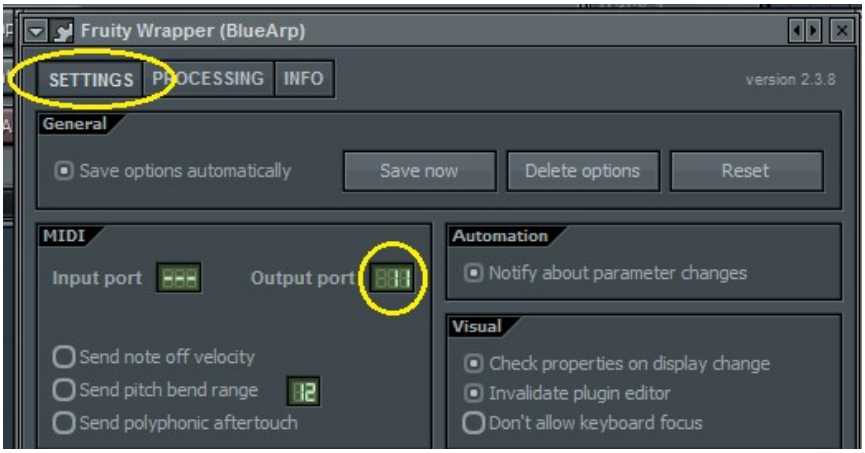
If your DAW is not present in this list, refer to other VST arpeggiator manuals like Kirnu Cream, Catanya, Nora or search for tutorials in google or youtube. For BlueARP procedure should be the same as for any other VST arpeggiator.

FL Studio (Fruity Wrapper method)

Load BlueARP, click «Wrapping settings» button:



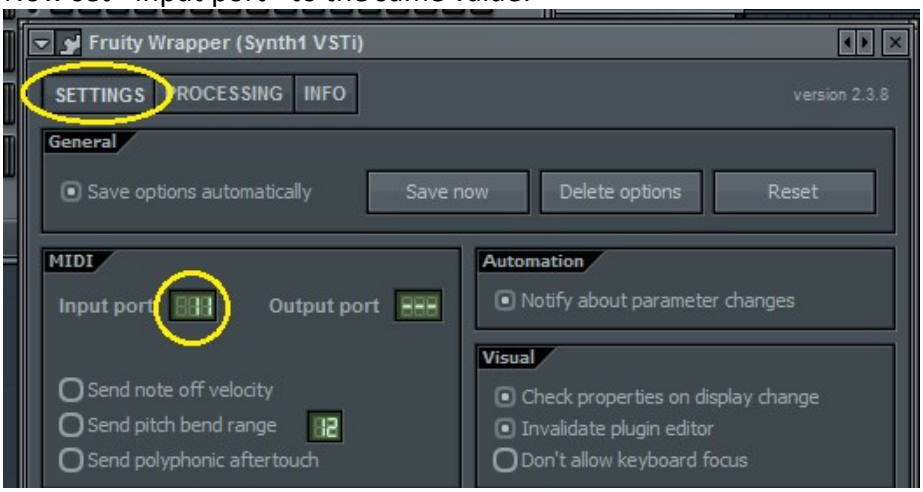
Click «SETTINGS» tab, set «Output port» to any value, not occupied by hardware MIDI devices, memorize this value:



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

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

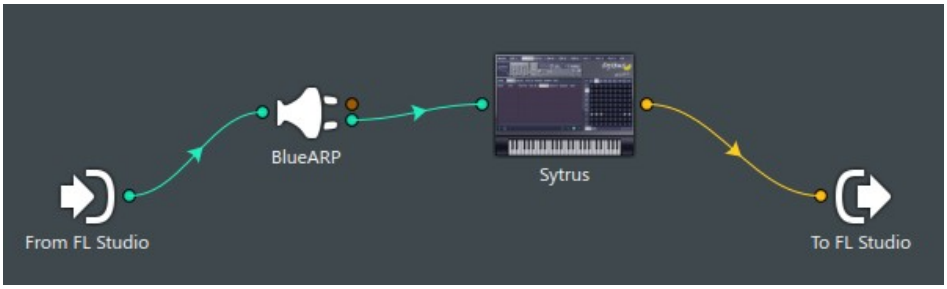
Now set «Input port» to the same value:



Now Synth1 will receive MIDI events generated by BlueARP.

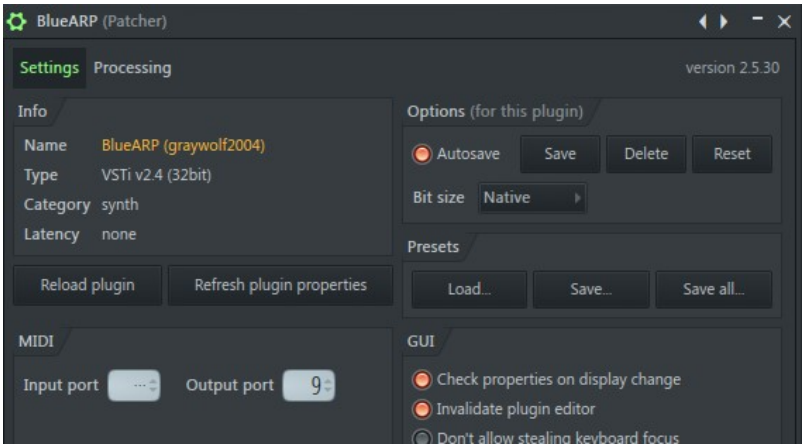
FL Studio (Patcher method)

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



Green arrows represent MIDI signal flow, yellow - audio signal.

Double cluck 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, VST synth (Synth1 VST in our case) on another.

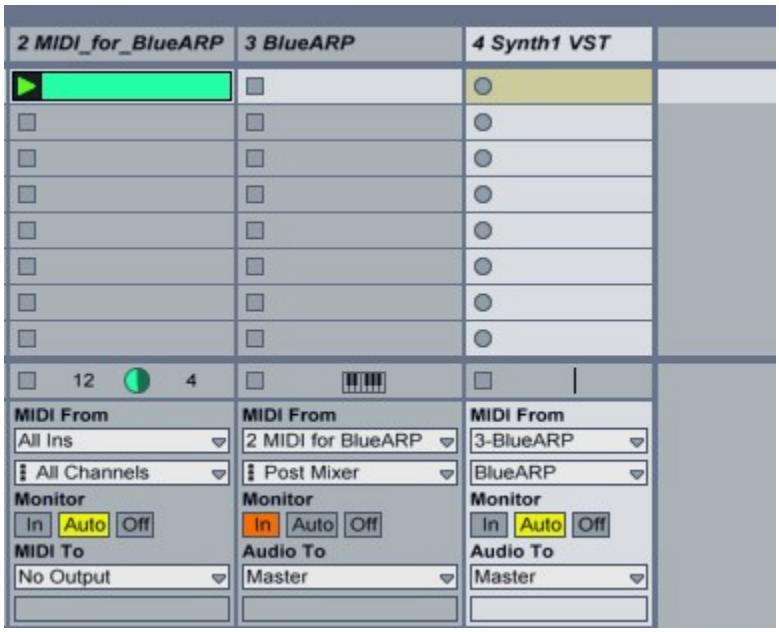
For Synth1 track, set MIDI From = BlueARP (both checkboxes).

For BlueARP track, set Monitor = In.

There's an issue - BlueARP will pick up MIDI from clips only when Motitor 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), it will hold your MIDI clips.

Add tracks for BlueARP and VST synth (we 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.

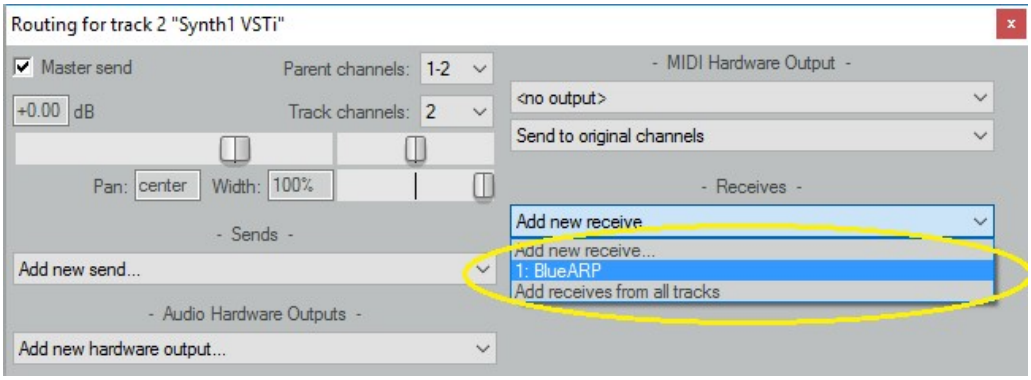
If you want to sequence external synthesizer (connected via MIDI), use “External instrument” (it's in “Live Devices” list).

REAPER

Add tracks for both BlueARP and target VST synth (Synth1 in our case).

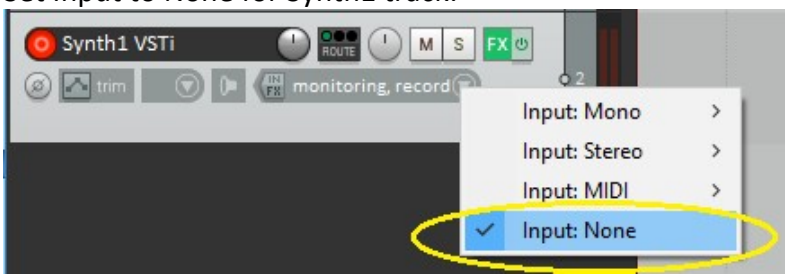
Press ROUTE button on Synth1 track:

Add new receive from BlueARP:



Now Synth1 receives notes from BlueARP, but you also need to prevent it from receiving notes directly from keyboard.

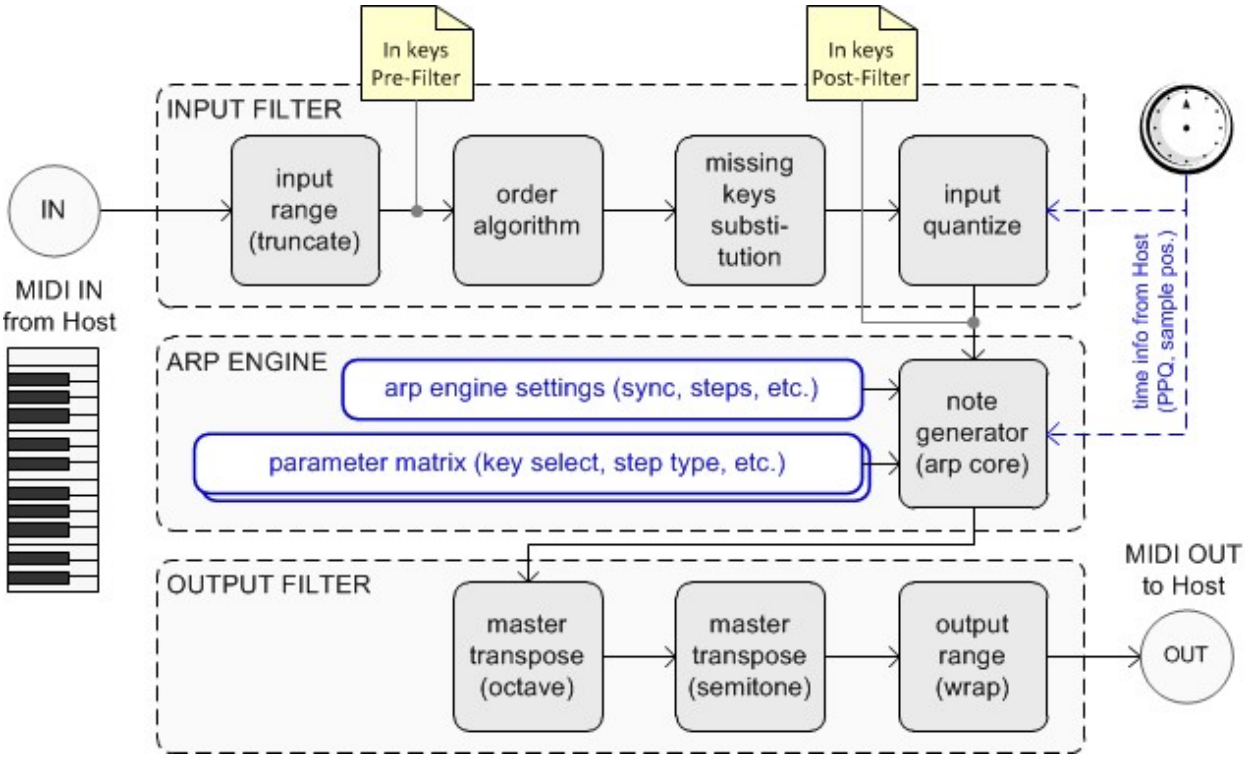
Set Input to None for Synth1 track:



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 basic data flow diagram for BlueARP. At the input BlueARP receives MIDI notes coming from host. These 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), generated by arpeggiator engine and further transposed by output filter.



pic. 1. BlueARP processing diagram.

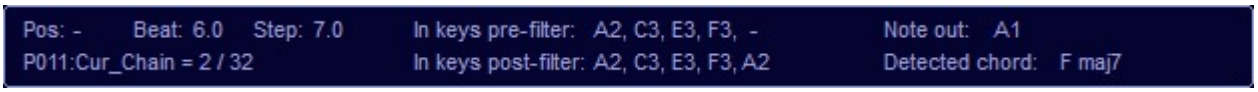
Main blocks are «Input Filter», «Arp Engine» and «Output Filter».

> *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 key press and release events (also it may be pitch bend, aftertouch and controller messages). From key on and off events, it generates *Key List* – an ordered list of keys with corresponding velocities (velocity is how hard you pressed 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 12).

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



See Information panel description for more details, page 23.

Post-filter key list goes directly to Arp Engine.

Arp Engine refers to Value bars (matrix editor), which contain pattern information for each step. For example, «key select» value bar 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 muted (Off). Refer to page 20 for more information about Value bars and Matrix editor.

BlueARP has unique «*missing keys substitution*» feature. It means when you have for example 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 12 for details).

Output Filter adds some post-processing to generated notes – octave / semitone transposition and wrapping notes to fit the given range. See page 17 for more details.

Program chains block allows you to merge several programs together to create longer patterns. You can automate current chain parameter and switch chains on the fly - it's great for live performances. See page 23 for more details.

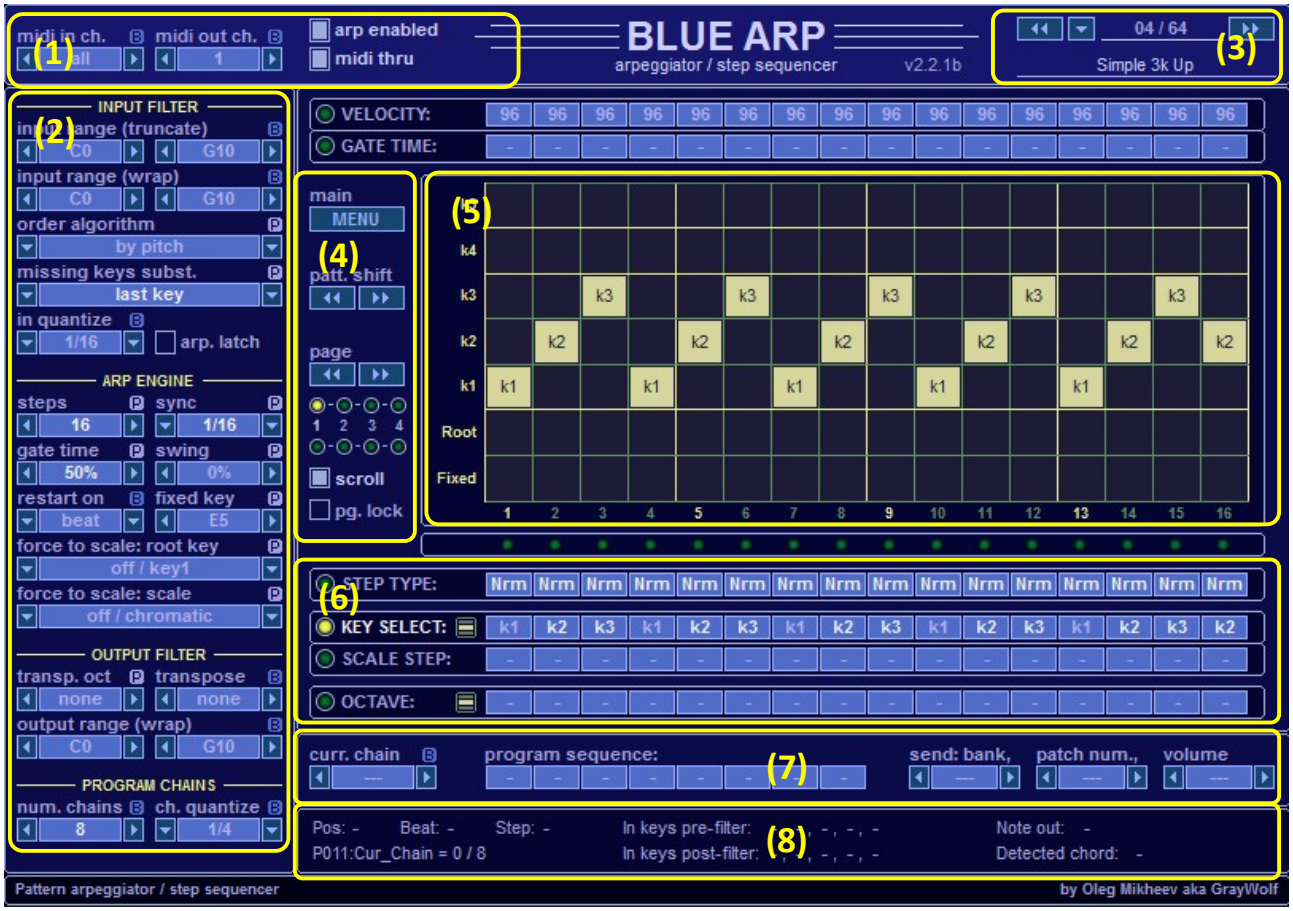
Interface

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.

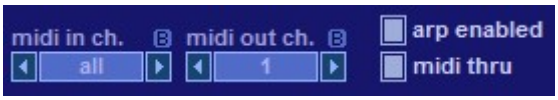
B or P marks tells whether this setting is saved with a bank **(B)** or program **(P)**. Global settings, stored in BlueARP.ini file, are marked **(G)**.

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;
(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** - information on current position, beat, input and output keys;

Block (1): Top panel



midi in ch	input MIDI channel
values	*all, 1 .. 16*
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	*1 .. 16*
comments	Default setting is 1, because soft synths usually don't care about MIDI channel. You may need it if you have multi-timbral hardware synth connected to BlueARP or several hardware synths chained on one MIDI output port, separated by channels

arp enabled	Arpeggiator is turned On
values	*On/Off (checkbox)*
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	*On/Off (checkbox)*
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 program-related «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 and have **(P)**-mark, some are bank-related and have **(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. *BlueARP can also pass outside-the-range notes 'as is' instead of ignoring them, it's controlled by this setting:* *MENU >> Settings >> «input range (trunc.) mode»*

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 will fit them into the given range by applying up or down octave transposition. Assume your set this range to A2...A3. When you press keys **F2**, A2, E2, A3, **E3**, the processed keys will be **F3**, A2, E2, A3, **E2** (bold notes were wrapped into the range A2...A3). It's sonically useful when you play chords all over the keyboard but want to your bass line to sound right, not too low or too high.

order algorithm	ordering (sorting) algorithm for input keys
values	*by pitch, by pitch desc, as played, as played desc, by velocity, by velocity desc, chord (normalized), chord (as played)*
comments	Default setting is «by pitch» - pressed keys come into arp engine in natural order, from left to right on the keyboard. It also means that «k1» in «KEY SELECT» bar will 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. «chord (normalized)» can be explained by example. You press C4+E4, Cmaj chord is detected. Ordered list will be C4+E4+G4 (complete Cmaj chord). If you play inverted Cmaj – G3+C4+E4, output will be the same, because chord is normalized. «chord (as played)» behaves the same way, except inverted chord will stay inverted.

missing keys subst.	missing keys substitution algorithm
values	*don’t play, cyclic, first key, last key, fixed key, (+1 oct/-1 oct variations)*
comments	When your pattern (KEY SELECT bar) 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. For example, you hold keys C5 and E5, while your pattern has keys «k1», «k2», «k3» and «k4». Info panel will show input key list pre-filter (before substitution) as «C5, E5, -, -, -». Key list psat-filter (after substitution) will be: *cyclic* *«C5, E5, C5, E5, C5»* *cyclic, +1 oct* *«C5, E5, C6, E6, C6»* *first key* *«C5, E5, C5, C5, C5»*

	first key, -1 oct	*«C5, E5, C4, C4, C4»*
	last key	*«C5, E5, E5, E5, E5»*
	last key, +1 oct	*«C5, E5, E6, E6, E6»*
	fixed key	*«C5, E5, G5, G5, G5» («fixed key» = G5)*

in quantize	input keys quantization
values	*none, 1/16, 1/12, 1/8, 1/6, 1/4, 1/2, 1 bar, 2 bars*
Comments	Real-time 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	*On, Off (checkbox)*
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, or to switch it or to free hands for knob adjustments.

Arp Engine parameters

Arp Engine takes note list from the input filter (after fitting to range, missing keys substitution, 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. ***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.*

sync	Step length (as a fraction of a bar)
values	*1/64, 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.

gate time	Note length (relative to step)
values	*1% .. 100%*
comments	Sets generated note length as a fraction of a step length.

swing	Swing control
values	*-50% .. 50%*
comments	Sets relative time shift for even steps as a fraction of a 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. For negative values, it will start earlier.

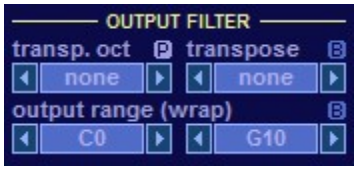
restart on	Pattern restart trigger
values	*beat, key, 1st key*
comments	In default «beat» mode step number is always aligned to PPQ (or song position) given by host. When your song or pattern restarts in a DAW, BlueARP pattern will also restart. With «key» setting, BlueARP will restart pattern each time new key/chord is pressed, after all previous keys were released. In "1st key" mode pattern will start with the first key/chord pressed and will keep going until you restart playback in a DAW.

fixed key	Fixed key value
values	*C0 .. G10 (MIDI notes 0 .. 127)*
comments	In «KEY SELECT» bar, you can set any step to «Fixed», it tells BlueARP to ignore input keys and take «fixed key» value. Set all steps to «Fixed» 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	*off/key1, detect from chord, C, C#, D ... Bb, B*
comments	You can either set a fixed root for a scale you want to fit notes in, or let BlueARP detect it dynamically from the chord you play. BlueARP recognizes chord inversions, so if you press (E4, A4, C5 - Am inverted), your root key will be **A**.

force to scale: scale	Sets scale key for "force to scale mode". Works together with "force to scale: root key" parameter
values	*"off/chromatic, detect from chord, Major, minor, harmonic minor, melodic minor, pentatonic Major, pentatonic minor, pentatonic neutral, pentatonic blues"*
comments	If you set anything except "off/chromatic", two things will happen: 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**. With "off/chromatic" selected, "SCALE STEP" will work as a semitone transposition. 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.

Output Filter parameters



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

transp. oct	Output transposition, octaves
values	*-3 oct .. +3 oct*
comments	

transpose	Output transposition, semitones
values	*-12 .. +12*
comments	

output range (wrap)	Range for output notes (wrapping)
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 highest value for "current chain" parameter.

num.chains	Sets maximum value for "current chain" parameter
values	*1 .. 16*
comments	To switch chains with a midi controller, you need to automate "current chain" parameter. If you use a knob for this, set "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 transition it should be done strictly at the start of a new beat. Chain quantize = 1/4 does exactly that and it's the default setting.

Block (3): Patch browser

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

Bank contains 128 programs, so you can configure up to 128 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).

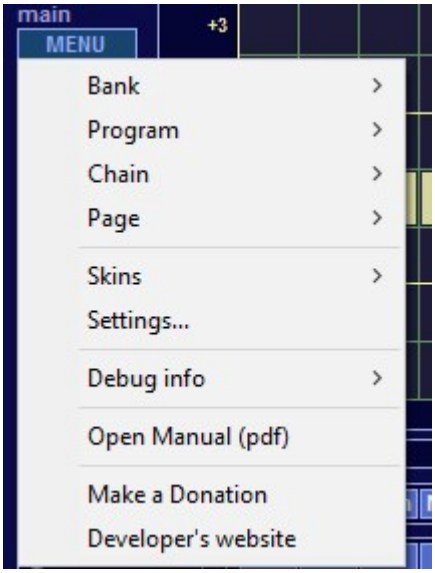
Pattern shift buttons perform cyclic 1 step shifting. It's useful, when your pattern doesn't match the beat and you want to align it. The shift is cyclic, so when you shift the patter right, the last step won't disappear but will «jump» to the beginning.

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

scroll checkbox - when checked, matrix will always show the page actually playing.

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

Main menu includes:



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

For Program, you can also copy\paste programs within a bank.

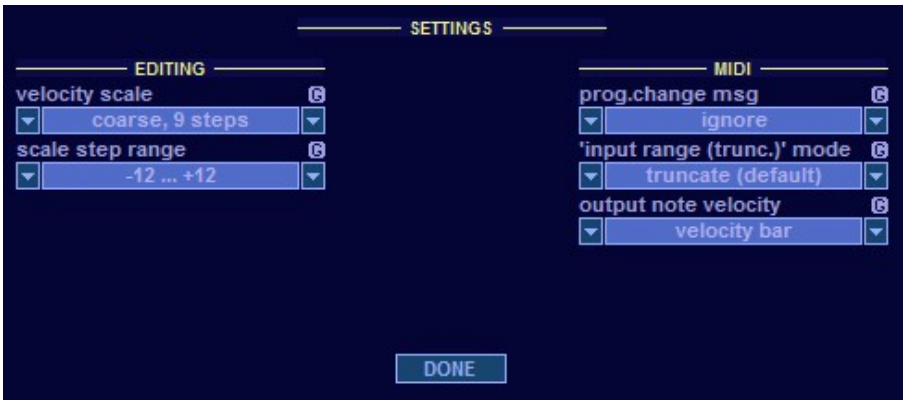
Chain – copy/paste and initialization for chains.

Page – copy and paste pages (makes sense for long patterns)

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.

Settings – brings up global settings page (see below)

"Settings..." screen



velocity scale	Sets velocity accuracy for "VELOCITY" value bar
values	*"coarse, 9 steps", "fine, 128 steps"*
comments	Select "fine, 128 steps" if you want to make fine velocity adjustments, otherwise it will go like 16, 32, 48, etc.

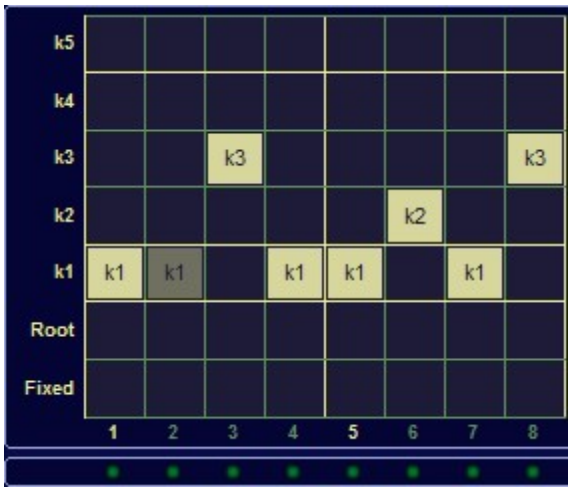
scale step range	Sets the range for "SCALE STEP" value bar
values	*"-12...+12", "0...+12", "-7...+7", "0...+7"*
comments	Default value is "-12...+12". For touch-screens it may be better to set "-7...+7" or "0...+7" for better accuracy.

prog.change msg	Sets how to respond to incoming Program Change MIDI message
values	*"ignore", "set program", "pass to synth"*
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).

input range (trunc.) mode	Sets how to treat keys outside the range "*input range (truncate)*"
values	*"truncate (default)", "pass thru (no arping)"*
comments	"truncate" - keys outside the input 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	*"velocity bar", "input key", "bar + input key"*
comments	"bar + input key": BlueARP takes output note velocity from VELOCITY bar and adjusts it to input note velocity (multiplying and normalizing them)

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 2 is set to Off, so «key select» value for this step doesn't make any difference.

Block (6): Value bars

Value bars represent step-related pattern 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;
- use mouse wheel to do the same thing;

Yellow label (or) next to «OCTAVE» and «KEY SELECT» labels sets the bar to **monophonic** and **polyphonic** mode. In polyphonic mode, you can set several values at once (either keys or octaves).

See descriptions for each value bar below.

VELOCITY	Velocity value for each step
values	*0, 16, 32 .. 127;*
comments	Default value is 96. Use it to set accent for certain steps. VELOCITY values will be ignored, if you set "output note velocity" = "input key" in MENU >> Settings. By default velocity has harsh scale (0, 16, 32 …), but you can switch it to **fine increment** in MENU >> Settings >> velocity scale.

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, with gate = 60% and GATE TIME for a step = "2x" note length for this step will be 60% * 2 = 120% or 1.2 steps.

STEP TYPE	Several options for output note generation
values	*Off – this step doesn't generate any note* *Nrm – Normal(default) – generates a note;* *Rst – this step will play the Rest of the previous step;* *Tie – this note will overlap with the previous one (for glides);* *Chr – Chord, or triggering all notes at once*
comments	«*Rst*» step simply means that this step continues to play the note from the previous step. You may chain several «*Rst*» steps together. «*Tie*» option may be tricky and not self-describing. It's main purpose is to create «glides» between notes. But it requires configuring 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 key2). When you configure the synth this way, «Tie» steps should create glides between notes.

KEY SELECT	Input key selection for the given step
values	*Fixed* – use fixed key from Arp Engine settings *Root* – root key from detected chord, key1 if no chord detected *k1..k5* – take keys №1..5 from key list (post-filter)
comments	Determines which key to take from «post-filter key list» for the current step. Yellow label next to "KEY SELECT" caption (or) toggles between monophonic and polyphonic mode. In **monophonic** mode you can only select one key for a step or all keys at once with STEP TYPE = Chord. In **polyphonic** mode you can select several keys at once, like k1+k2 or k1+k3.

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 variations*.

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. Otherwise, will transpose output note with respect to the selected scale.

OCTAVE	Octave transposition for each step
values	*-3, -2, -1, 0, +1, +2; +3*
comments	It's convenient for bass lines, where the steps are usually transposed for the whole octaves. Yellow label next to "OCTAVE" caption (or) toggles between monophonic and polyphonic mode. In **monophonic** mode all keys for a given step are transposed by octaves. In **polyphonic** mode only key 1 is transposed. So if you have STEP TYPE = Chord, OCTAVE = -1; 0 and press F4 + A4, output notes will be F3 + F4 + A4. (key1 = F4 is copied down an octave, but not key2 = A4)

Block (7): Program chains

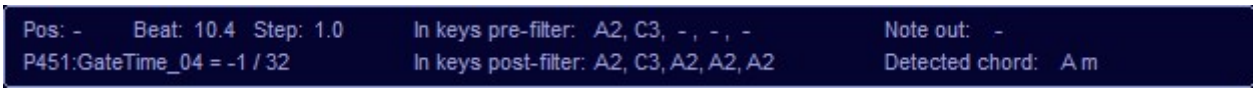
Program chainer is a tool to stack several programs (patterns) together into a longer "super-pattern".

Program sequence bar holds numbers of chained programs for a current chain.

Current chain parameter can be automated; its maximum value is set by "num. chains" parameter.

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.

In keys pre-filter - input keys, as they are pressed.

In keys post-filter - input keys after «input filter» - truncated and wrapped to fit the given range, ordered, with missing keys substituted, quantized. This is what goes into the BlueARP «core» engine.

Note out – generated notes.

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