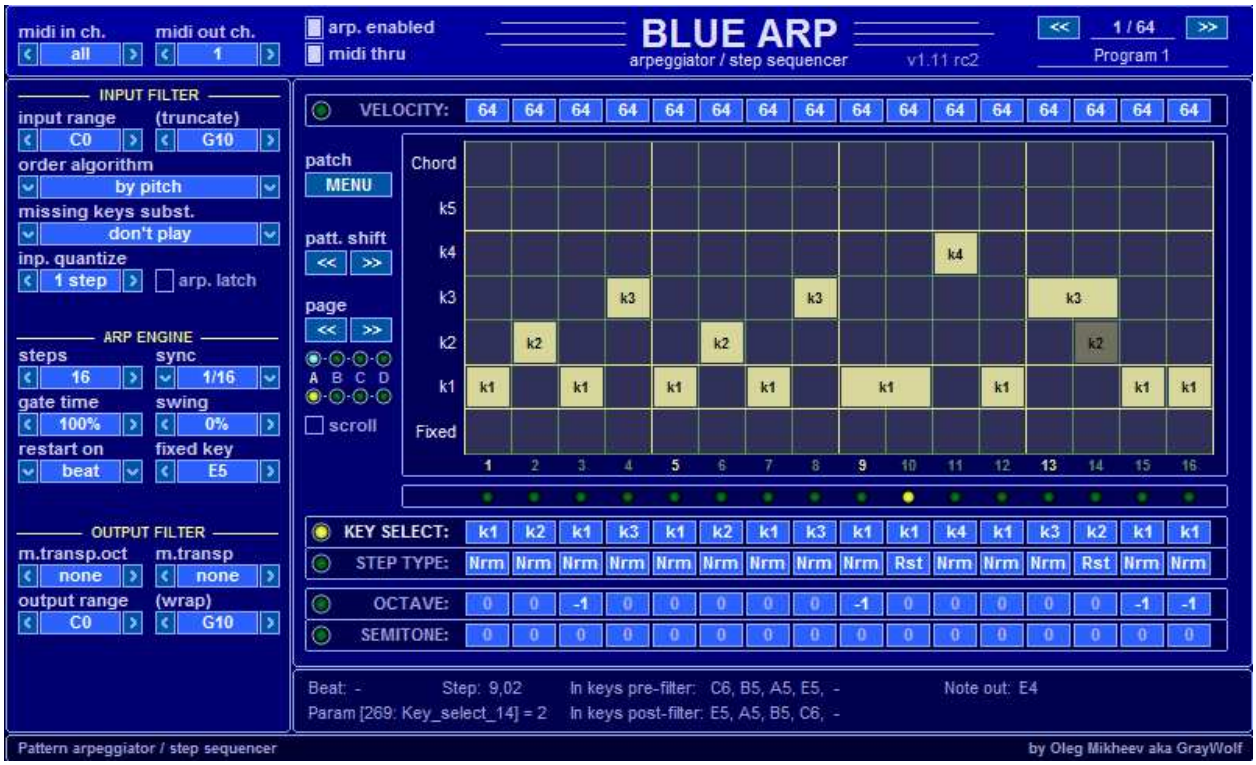
BLUE ARP

Operation Manual



Pattern Arpegiator / Step Sequencer
in a VST Plugin format

by Oleg Mikheev aka GrayWolf
http://www.graywolf2004.net

April, 2013

Table of Contents

Introduction

BlueARP is a programmable pattern arpeggiator / step sequencer, it comes as a win32 VST plugin. It's a pure MIDI plugin, so it has to be routed to either software or hardware synth (in any VST-enabled DAW like FL Studio, Ableton Live, Cubase, Reaper, etc.). Basically you need to program some pattern in BlueARP (it's quite fast with its matrix editor), 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 (it has swing feature).

Compatibility info

Format: win32 (x86) VST plugin
OS: Windows XP, Vista, Windows 7

Features

- Up to 64 steps per pattern;
- Up to 64 programs per bank;
- Intuitive matrix editor to program patterns quickly;
- 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;

Installation

You need a VST-compatible host to run this plugin (like FL Studio, Ableton Live, Cakewalk Sonar, Cubase, Nuendo, Tracktion).
To install BlueARP, simply copy BlueARP.dll file to your VstPlugins folder.
By default, it should be

C:\Program Files\Steinberg\VstPlugins

After that, you should rescan plugin directory from your DAW (refer to your DAW manual).

IMPORTANT: BlueARP is a pure MIDI plugin, it only generates MIDI messages and doesn't produce any sound. It's designed to sequence other synths, so you need to route BlueARP's MIDI output to MIDI input of either software or hardware synthesizer. If you don't see this procedure description for you host application in the manual, refer to your host manual.

Setting up BlueARP in VST host applications

Since BlueARP is a pure MIDI plugin, you need to route it's MIDI output to either hardware of software synth. Below there's a description how to do this in several host applications.
If your application is not present in this list, you can refer to other VST arpeggiators like Kirnu or Catanya (just search for howto's in google or youtube). For BlueARP procedure should be nearly the same.
Also check this tutorial (on how to route MIDI between plugins for several DAWs):
http://www.mandelbrotdrummer.com/MidiOutVSTi.html

FL Studio

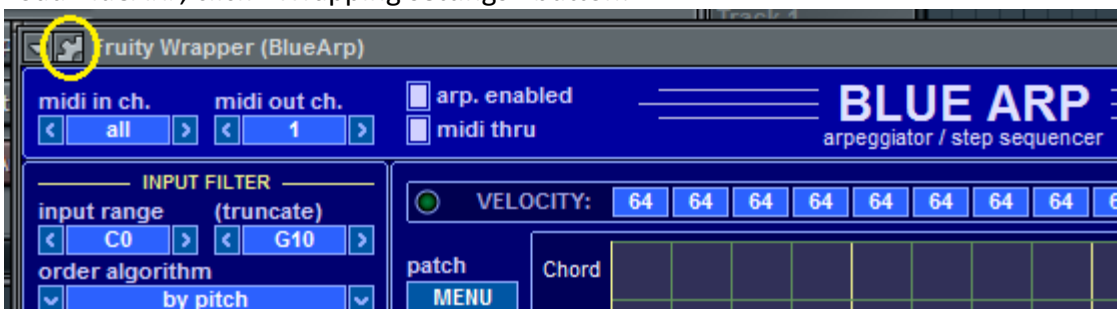
There are 2 ways to route MIDI between plugins:

- Adjusting Fruity Wrapper settings;
- Using Fruity Patcher;

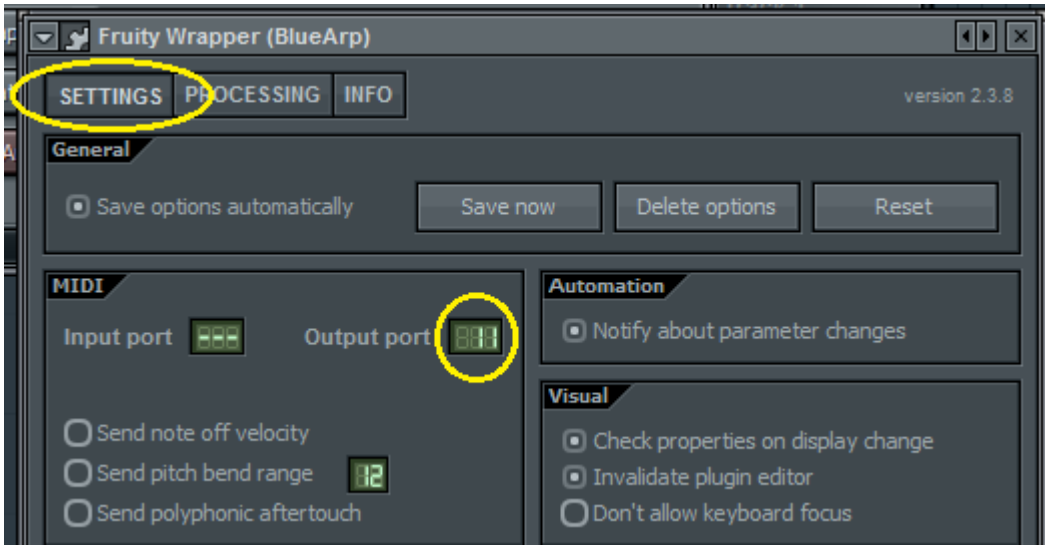
Here we describe only «Fruity Wrapper» way.

Configuring Fruity Wrapper

Wrapping settings for BlueARP
Load BlueARP, click «Wrapping settings» button:



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

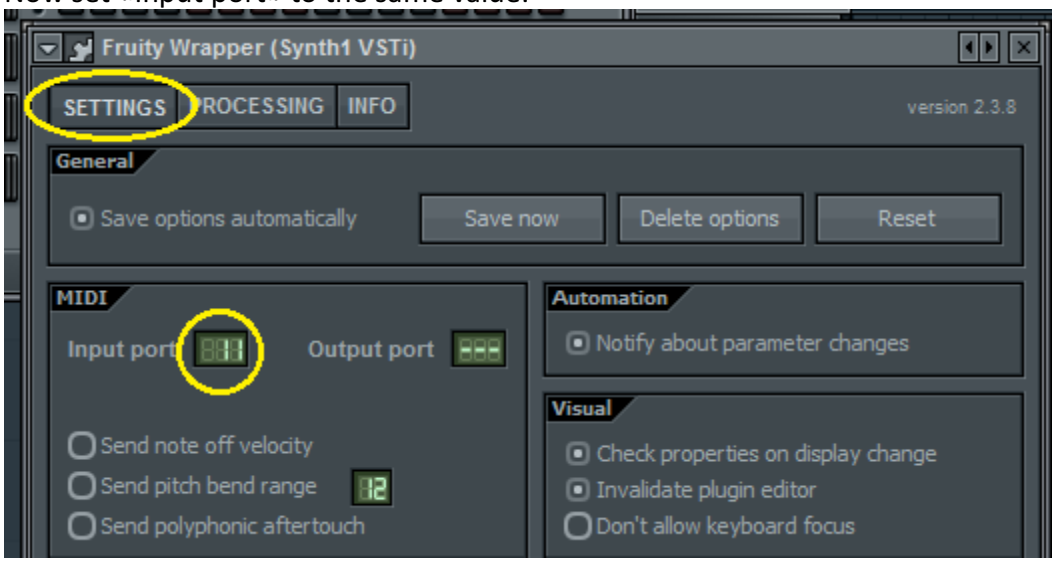


Click «Wrapping settings» button again to return to main plugin's window.

Wrapping settings for VST Synth

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

Now set «Input port» to the same value:



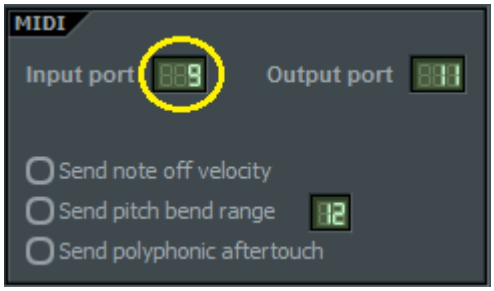
Now Synth1 will receive MIDI events generated by BlueARP.

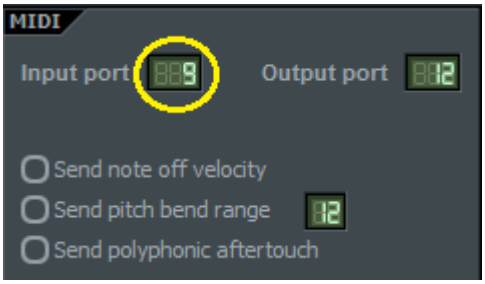
Tips for keyboard-split performance

You can control several instances of BlueARP from one keyboard, and you can either split them or layer them. To do this, you need one «master» track, which will route MIDI messages to all BlueARP instances.

Step 1. Select «MIDI Out» plugin as a master track. Set PORT to any value (not occupied by hardware), memorize it.

For each instance of BlueARP, set the same port as «Input port»:





(output ports are connected to software synths, as described above).

For keyboard-split performance, adjust «input range» settings in each instance of BlueARP (E5 is a split point in our example):

Ableton Live

Load BlueARP and VST synth you want to sequence (Synth1 in this example).

For VST Synth, set MIDI From = “BlueARP” (both combo boxes), arm trach for recordong (tiny red button in the bottom).

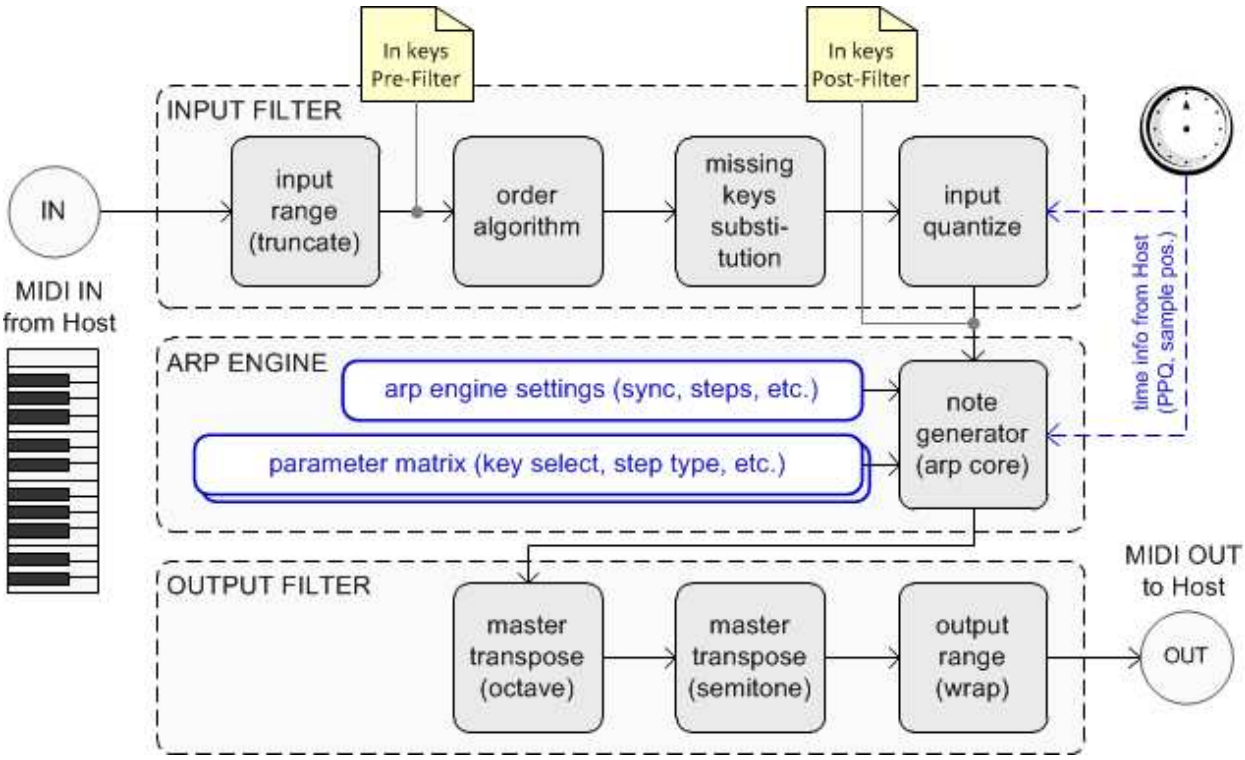
For BlueARP, set Monitor to “In” (for some reason “Auto” option doesn't work in my case).

Now BlueARP will capture MIDI inpit from your MIDI keyboard, generate MIDI arpeggio and send it to Synth1. When you're done with this track, set Monitor back to “Auto” or “Off”.

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

Signal flow

To use BlueARP to the maximum, it's necessary 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.

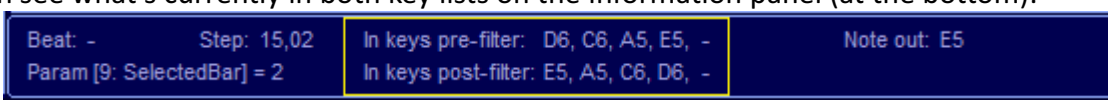


pic. 1. BlueARP simplified processing diagram.

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):



See Information panel description for more details, page 15.

Arp Engine generates pattern notes according to post-filter key list it receives from 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 simple post-processing to generated notes – octave / semitone transposition and wrapping notes to fit the given range.

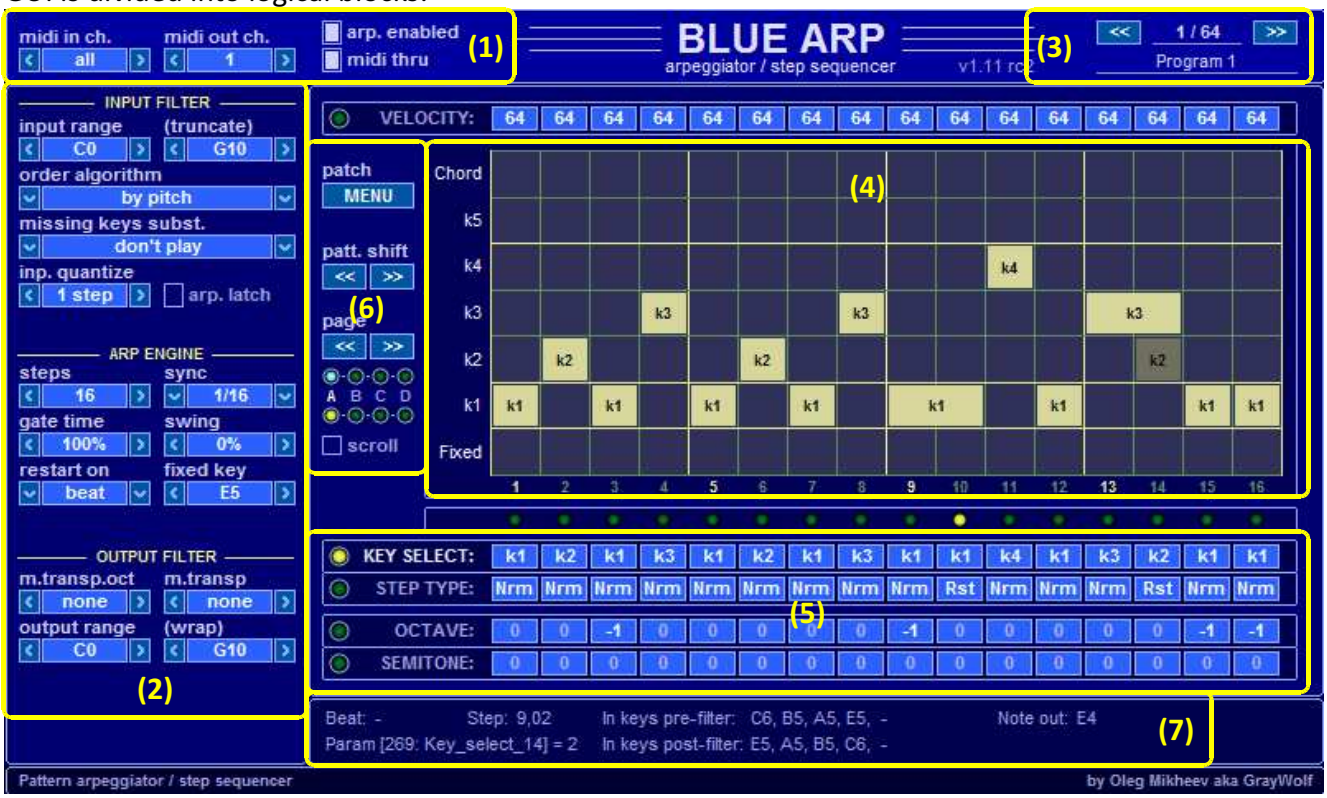
Interface

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.

Main window layout

GUI is divided into logical blocks:



(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) **Basic patch browser** allows to scroll through programs and to rename them. There are 64 programs available in a bank;

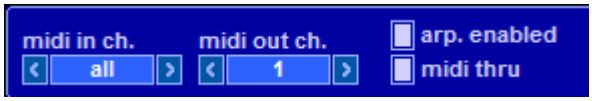
(4) **Matrix editor** is a key element of BlueARP. It represents step-related values for the selected bar (like KEY SELECT, STEP TYPE, etc.);

(5) **Value bars** represent step-dependent pattern parameters, like per-step key number, step type (normal, rest of the previous note, tie, off), octave and semitone transpose. To adjust the value, drag the «value box» up or down;

(6) **Patch menu and pattern controls** - cyclic pattern shift (left and right), page selector (for patterns longer than 16 steps);

(7) Information 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: *all, 1 .. 16*
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*
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 turned On.
values: *On/Off (checkbox)*
When set to 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)*
See «arp. enabled» setting description.

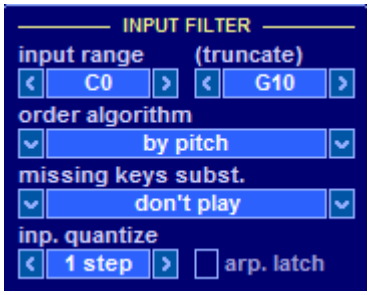
Block (2): Left panel

Left panel settings are divided into 3 blocks – «Input filter», «Arp engine» and «Output filter». It corresponds to signal flow path (see diagram on page 6).

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

In general, left panel represents all program settings, except the pattern itself.

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 range for input notes (low and high values)
values: *C0 .. G10 (MIDI notes 0 .. 127)*
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.

order algorithm range for input notes
values: *as played, as played desc, by pitch, by pitch desc, by velocity, by velocity desc*
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.

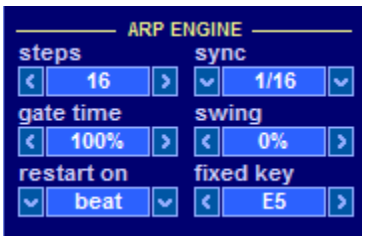
missing keys subst. missing keys substitution algorithm
values: *don't play, cyclic, first key, last key, fixed key (also +1 oct/-1 oct variations)*
When your pattern has more keys than you actually play, this setting will determine whether to ignore these steps (don't play) or substitute missing keys.
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:

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» param was set to «G5»)*

inp. quantize missing keys substitution algorithm
values: *none, 1 step .. 16 steps*
Quantization for input keys (in steps). For example, you set sync to 1/16 (1 step = 16th note). In this case inp. quantize = 4 steps means that BlueARP will capture pressed keys on the start of each beat, with inp. quantize = 16 steps – on the start of each bar and so on.

arp. latch Latch (or hold) pattern
values: *On, Off (checkbox)*
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.

Arp Engine parameters



Arp Engine takes this list from input filter and generates notes at the output, also referring to MIDI clock and current PPQ (position in a song or pattern).

steps Number of steps in pattern
values: *1 .. 64*
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 nice.

> ***Hint**: All 64 steps of a pattern are stored with 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 steps 17..32 if you change number of steps back to 32.*

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*
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).

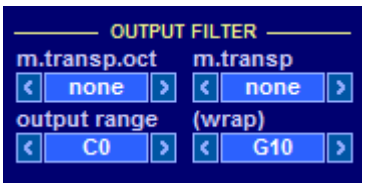
gate time Note length (relative to step)
values: *1% .. 100%*
Sets note length as a fraction of step length.

swing Swing control
values: *-50% .. 50%*
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: *beat, key*
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).

fixed key Fixed key value
values: *C0 .. G10 (MIDI notes 0 .. 127)*
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.

Output Filter parameters



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

m.transp.oct Master transpose, octaves
values: *-3 oct .. +3 oct*
Master transpose is applied to output notes

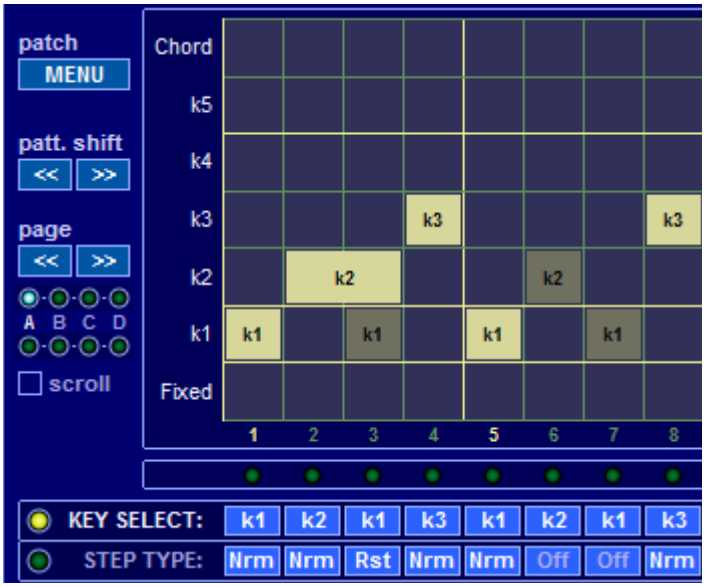
m.transp. Master transpose, semitones
values: *-12 .. +12*
Master transpose is applied to output notes

output range Range for output notes
values: *C0 .. G10 (MIDI notes 0 .. 127)*
Unlike input range, notes that are outside the range will be wrapped (transposed to fit the range). Say your output range is C5..C6 and generated note is D3 – it will be transposed to D5.

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 program name, click on it, type in new name and hit enter or click somewhere outside on the top panel.

Block (4): Matrix editor



Matrix editor represents value list for the selected Value bar («key select» bar on the picture above). So, you can adjust a value 2 ways – either in matrix editor or in value bar itself (see next chapter).
Just 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.

Labels for «value bricks» represent «key select» setting, to simplify pattern reading when other bars are edited. Greyed 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 3 is set to «Rest», so it will play a key from the previous step.

Block (5): Value bars

Value bars represent step-related patters parameters. Selected value bar (hit bar caption to select it) is also shown in Matrix editor.
To adjust value for a certain step, click on it and drug it up (increase) or down (decrease).
See description for each value bar below.

KEY SELECT Key selection for the given step
values: *Fixed – use fixed key from Arp Engine settings*
k1..k5 – take key №1 .. №5 from key list (post-filter)
Chord – take all keys from key list

Determines which key to take from post-filter key list for the current step.

> *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.*

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);

«*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 also 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. The same thing will happen with BlueARP generated notes when you set steps to «Tie». But it won't happen if you press them without overlapping (like press key1, release key1, press key2), the same – when you set steps to «*Nrm*».

OCTAVE Octave transposition for each step
values: *-2, -1, 0, +1, +2;*
It's convenient for bass lines, where the steps are usually transposed for the whole octave.

SEMITONE Semitone transposition for each step
values: *-12, .. +12;*
Octave and semitone transposition values are summarized like
[octave transp.]*12 + [semitone transp.].

VELOCITY Velocity value for each step
values: *0, 16, 32 .. 127;*
Default value is 64. Use it to set accent for certain steps.

Hint: Not all synths/patches respond to velocity. In most cases, velocity is linked to sound volume (amp EG amount) and filter cutoff (filter EG amount).

Block (6): Patch menu and pattern controls

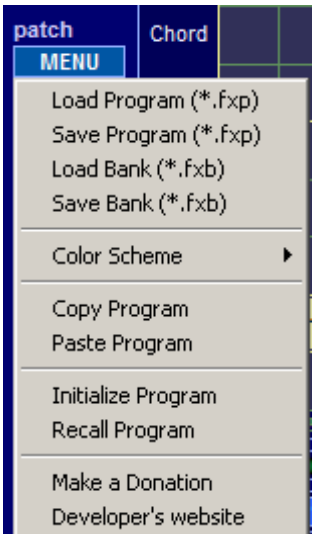
Patch menu includes Bank load/save, Program load/save and some other functions (see picture below for details).

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.

Patch menu includes:



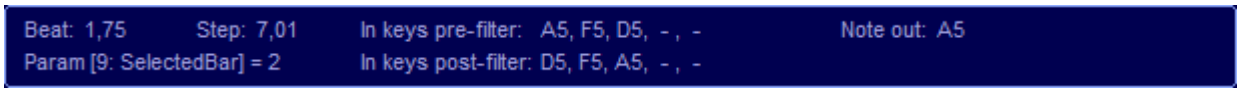
Color scheme – schemes are stored in "BlueARP.ini" file. Default blue color scheme is built in. It's possible to create your own scheme by editing ini file.

Copy \ Paste program – you can use it when you want to play with a program without losing original version (since BlueARP has no undo function)

Initialize program – reset all settings for current program.

Recall program – reverts to program state as it was on project load.

Block (7): 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.