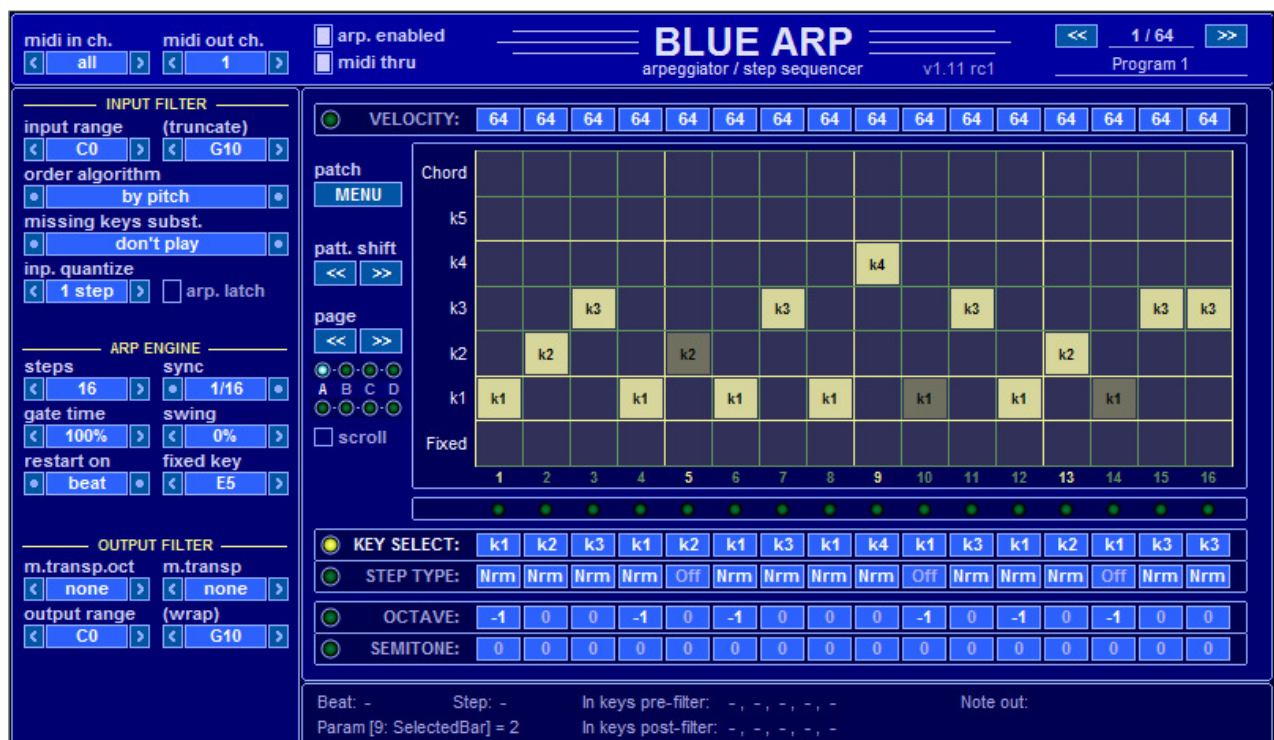


BLUE ARP

Operation Manual



Pattern Arpeggiator / Step Sequencer in a VST Plugin format

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

January, 2013

Contents

Introduction	3
Installation	4
Signal flow	5
Interface	6
Main window layout	6
Parameter description	7
Block (1): Top panel	7
Block (2): Left panel	7
Block (3): Patch browser	9
Block (4): Matrix editor	9
Block (5): Value bars	10
Block (6): Patch menu, pattern shift and page select.....	10
Block (7): Information panel	10
Links	11

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;

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

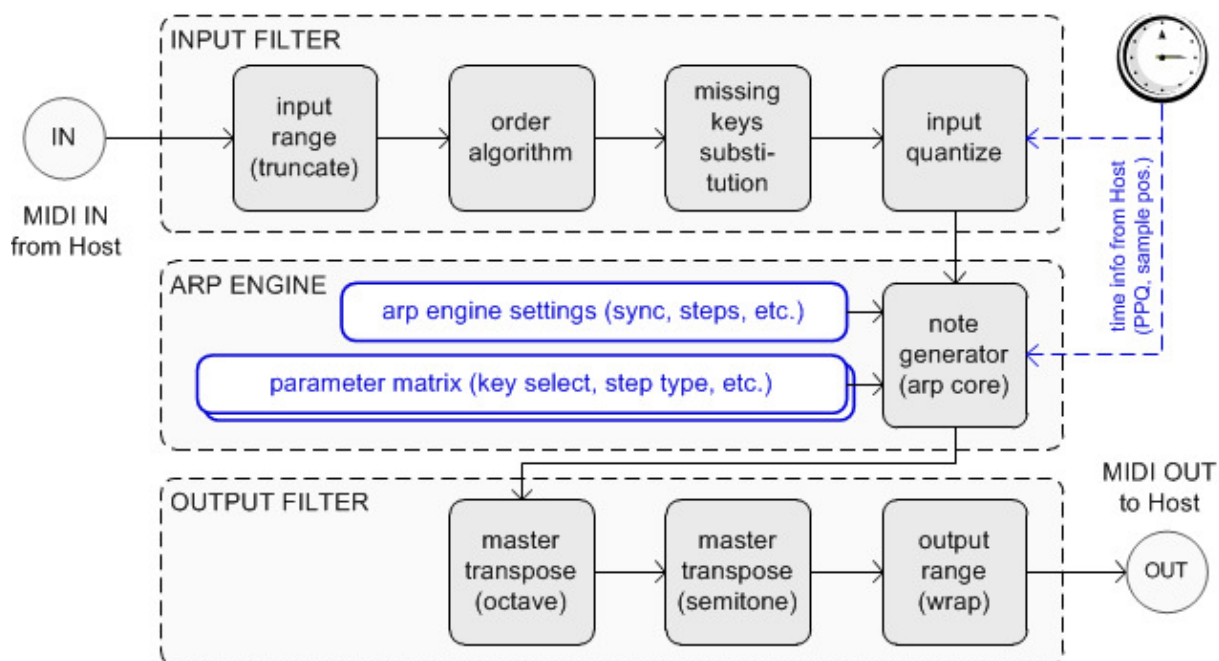
After that, you should rescan plugin directory from your DAW (refer to your DAW manual).

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 your host application in the manual, refer to your host manual.

Signal flow

To use BlueARP to the maximum, it's necessary to have a concept about it's 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.



Also BlueARP receives timing information from host (clock, current song position, transport status). Timing information is necessary for real-time input quantize.

(to be extended) ???

Interface

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

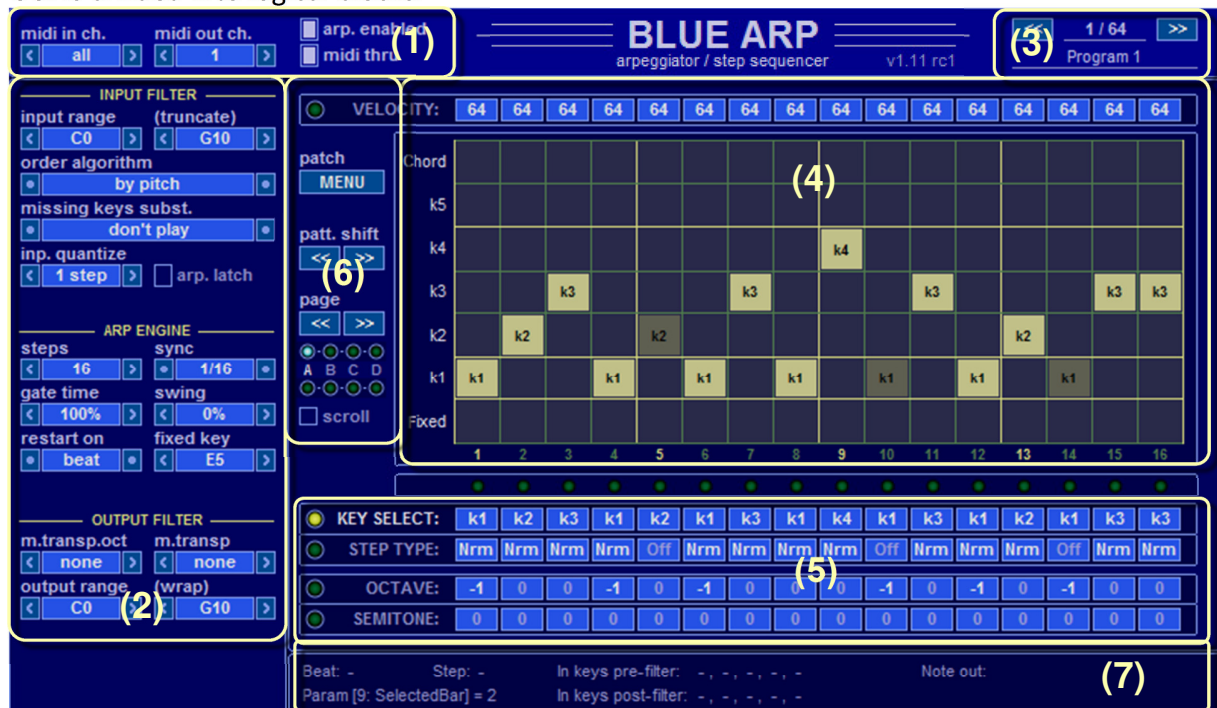


To adjust the value, drag it up or down.

For controls with buttons, arrow buttons ◀ ▶ adjust the value, «dot» 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, 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;

Parameter description

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.

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

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

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:

<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» 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, 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.

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.

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's 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.

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

Bank contains 64 programs, so you can configure up to 64 arpeggiator patterns.

Left and right buttons switch to previous or next program.

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 is quite intuitive by itself, so only some tips:

- drag the mouse from left to right to quickly set all the steps to certain value;
- you can also adjust step-related values in value bars (5);

Block (5): Value bars

Hit bar caption to make it active. Matrix editor (4) shows active bar. To adjust any value for a certain step, click on it and drag it up (increase) or down (decrease).

Block (6): Patch menu, pattern shift and page select

Patch menu includes Bank load/save, Program load/save and Program initialize functions.

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

Page select buttons are necessary when your pattern is longer than 16 steps and doesn't fit single screen (16 steps maximum). Check «scroll» flag if you want BlueARP to automatically switch pages while pattern is playing.

There are 2 small LED bars underneath, 1st one shows selected page, second one – page being played.

Block (7): Information panel

Shows current beat, step and some other information.

Links

Developer website:

<http://www.graywolf2004.net>

BlueARP discussion thread at KVR Audio forums:

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

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

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

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

graywolf2004@gmail.com

graywolf2004@mail.ru

Oleg Mikheev aka graywolf.