

# BLUE ARP

## Operation Manual

Corresponds to BlueARP v2.3.8



## Pattern Arpeggiator / Step Sequencer

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

by Oleg Mikheev aka Graywolf

<http://www.graywolf2004.net>

May 2021

# Table of Contents

<b>INTRODUCTION .....</b>	<b>3</b>
<b>HOW TO INSTALL BLUEARP.....</b>	<b>4</b>
WINDOWS VST2 VERSION .....	4
WINDOWS VST3 VERSION .....	4
MAC OSX VST2 VERSION.....	4
MAC OSX VST3 VERSION.....	4
MAC OSX MIDI-FX VERSION (FOR LOGIC PRO X).....	5
<b>HOW TO REMOVE BLUEARP .....</b>	<b>5</b>
<b>SETTING UP BLUEARP IN SOME DAWs .....</b>	<b>6</b>
FL STUDIO (FRUITY WRAPPER METHOD) .....	6
FL STUDIO (PATCHER METHOD) .....	8
ABLETON LIVE .....	9
REAPER 6.X AND LATER.....	10
REAPER OLDER VERSIONS .....	10
<b>SIGNAL FLOW.....</b>	<b>11</b>
<b>INTERFACE .....</b>	<b>13</b>
MAIN WINDOW LAYOUT .....	14
BLOCK (1): TOP PANEL .....	15
BLOCK (2): LEFT PANEL: ARP / MAIN .....	16
Input Filter .....	16
Arp Engine.....	19
Output Filter .....	21
Program Chains.....	22
BLOCK (2): LEFT PANEL: SETTINGS .....	23
BLOCK (3): PROGRAM BROWSER.....	24
BLOCK (4): MAIN MENU AND PATTERN CONTROLS .....	25
BLOCK (5): MATRIX EDITOR .....	27
BLOCK (6): VALUE LANES .....	27
BLOCK (7): PROGRAM CHAINS .....	30
BLOCK (8): INFORMATION PANEL .....	31
<b>FAQ / TROUBLESHOOTING.....</b>	<b>32</b>
INSTALLING BLUEARP .....	32
SYNC & TIMING ISSUES.....	32
RENDERING AUDIO.....	32
<b>LINKS .....</b>	<b>33</b>

# Introduction

BlueARP is a programmable pattern arpeggiator / step sequencer, it comes as a VST or MIDI-FX plug-in for Windows and MAC OSX, both 32 and 64-bit. BlueARP is a pure MIDI plugin, it doesn't generate any sound by itself but transforms MIDI messages. It has to be routed to either software or hardware synth in any VST/AU-enabled DAW (Digital Audio Workstation) software 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 transforms these chords into melodic phrases according to the pattern you programmed or selected.

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 a swing feature.

## Compatibility info

Formats: VST plugin 32-bit, VST plugin 64-bit, AU MIDI-FX 64-bit (for Logic Pro X)  
 OS: OS X (10.7 or later, tested on 10.11.6), Windows XP or higher

## Features

- Up to 64 steps per pattern;
- Up to 128 programs per bank;
- «Chains» feature to chain patterns together into longer «super-patterns»
- Ability to switch chains on the fly (for live performances);
- 128 factory patterns to start with;
- Intuitive matrix editor to program patterns quickly;
- Almost all controls can be automated;
- Up to 5 input keys in a chord;
- Real-time input quantization;
- Chord recognition;
- Input range setting for keyboard-split performances;
- Separate settings for octave and semitone per step transpose;
- Configurable color schemes (skins);

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, but the same can be done with Ableton Live and many other DAWs.

# How to install BlueARP

Before installing newer version of BlueARP, it is recommended to remove the existing version first, unless you want to use both older and newer version (refer to the next chapter «How to remove BlueARP»).

## Windows VST2 version

**Step 1.** Unzip the package, copy "BlueARP\_Win\_VST2\_vXXX" folder to your VST plugins directory.

Normally it will be:

- C:\Program Files\Steinberg\Vstplugins\ or
- C:\Program Files (x86)\Steinberg\Vstplugins\ (*for 32-bit plug-ins on Windows 64-bit*)

**Step 2.** In you DAW (Cubase, FL Studio or whatever you use), re-scan VST plugins folder (refer to the respective manual on how to do this). «BlueARP» or «BlueARP.x64» (64-bit version) should appear in plugin list and it is now ready to use.

## Windows VST3 version

**Step 1.** Unzip the package, copy "BlueARP\_Win\_VST3\_vXXX" folder to your VST3 plugins directory.

Normally it will be:

- C:\Program Files\Common Files\VST3\ or
- C:\Program Files (x86)\Common Files\VST3\ (*for 32-bit plug-ins on Windows 64-bit*)

**Step 2.** In you DAW (Cubase, FL Studio or whatever you use), re-scan VST plugins folder (refer to the respective manual on how to do this). «BlueARP» should appear in plugin list and it is now ready to use.

**PS.** Some DAW applications may combine VST2 and VST3 version of the plugin, in some cases VST3 version may have priority over VST2 version. FL Studio, for example, has a setting "Combine VST2 and VST3 versions of the plugin".

## Mac OSX VST2 version

**Step 1.** Unzip the package, copy «BlueARP\_OSX\_VST2\_vXXX» folder to your VST plugins directory.

It should be one of the following:

- Hard disk/Library/Audio/Plug-Ins/VST (*for all users*)
- Hard disk/Users/<username>/Library/Audio/Plug-Ins/VST (*for <username> only*)

**Step 2.** In you DAW (Cubase, FL Studio or whatever you use), re-scan VST plugins folder. BlueARP should appear in plugin list and it is now ready to use.

## Mac OSX VST3 version

**Step 1.** Unzip the package, copy "BlueARP\_OSX\_VST2\_vXXX" folder to your VST3 plugins directory.

It should be one of the following:

- Hard disk/Library/Audio/Plug-Ins/VST3 (*for all users*)
- Hard disk/Users/<username>/Library/Audio/Plug-Ins/VST3 (*for <username> only*)

**Step 2.** In you DAW (Cubase, FL Studio or whatever you use), re-scan VST plugins folder. BlueARP should appear in plugin list and it is now ready to use.

## Mac OSX MIDI-FX version (for Logic Pro X)

**Step 1.** Unzip the package, copy "BlueARP\_OSX\_MFX\_vXXX" folder to your Audio Units directory.

It should be one of the following:

- Hard disk/Library/Audio/Plug-Ins/Components (*for all users*)
- Hard disk/Users/<username>/Library/Audio/Plug-Ins/Components (*for <username> only*)

**Step 2.**

In you DAW (Logic Pro, GarageBand or whatever you use), re-scan Audio Unit plugins folder. BlueARP should appear in plugin list and it is new ready to use.

## How to remove BlueARP

BlueARP has no installer, so just remove "BlueARP\_Win\*" folder on Windows or "BlueARP\_OSX\*" folder on Mac (the one you copied during installation).

If you want to remove all traces of BlueARP in your system, also delete the following folder:

**Windows:** C:\Users\<user>\AppData\Roaming\BlueARP

**OSX:** C:/Users/<username>/Library/Application Support/BlueARP

This is the place where BlueARP stores its "ini" file with the settings like selected skin index, GUI scale. It is a small file, way below 1 Kbyte in size.

*The reason I had to put these settings into separate folder is because VST/AU folder with the plugin itself often doesn't grant write permission to the plugin, so it can't save the settings.*

### Troubleshooting

When you try to delete the folder, system may give an error: "Oxanium\*.ttf" files are locked by the system. This may happen because BlueARP uses these fonts for GUI rendering, they are bundled into the package. Upon loading, system locks these files and won't allow to delete them.

To solve this problem, try the following:

- Manually delete all "Oxanuim" fonts from your system
- Reboot
- Try to delete the folder again

## Setting up BlueARP in some DAWs

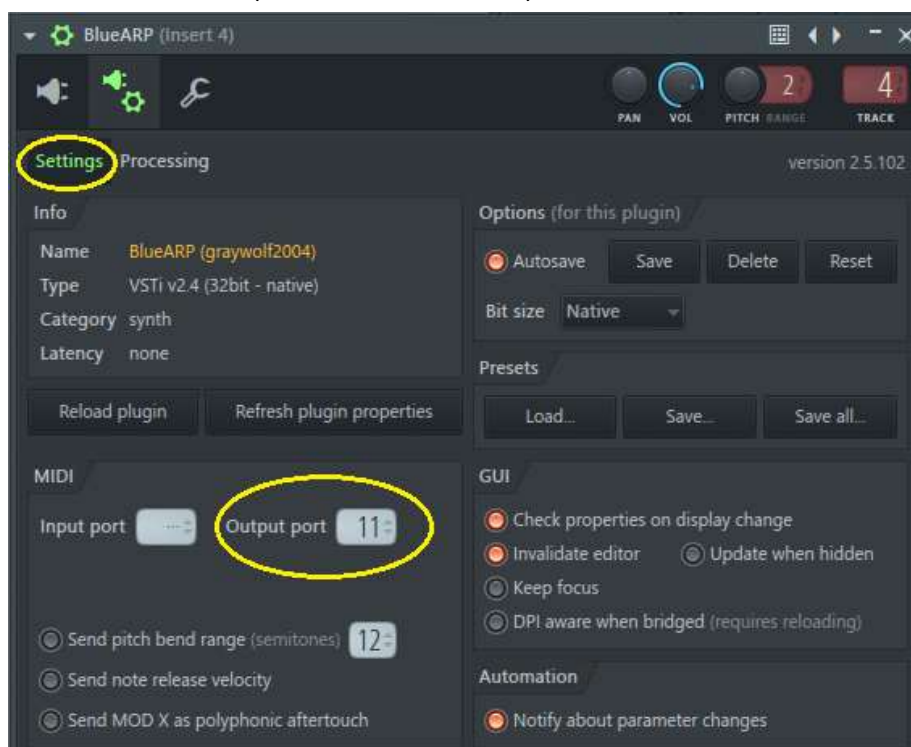
If your DAW is not present in this list, refer to other VST arpeggiator manuals like *Kirnu Cream*, *Catanya*, *Nora* or search for tutorials with keywords «**how to set up MIDI plugin in DAW ZZZ**». For BlueARP procedure should be the same as for any other MIDI plugin.

### FL Studio (Fruity Wrapper method)

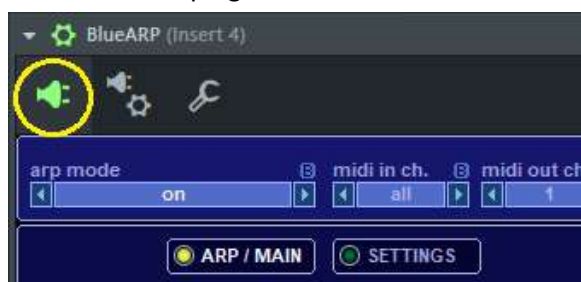
Load BlueARP, click the buttons as shown on picture:



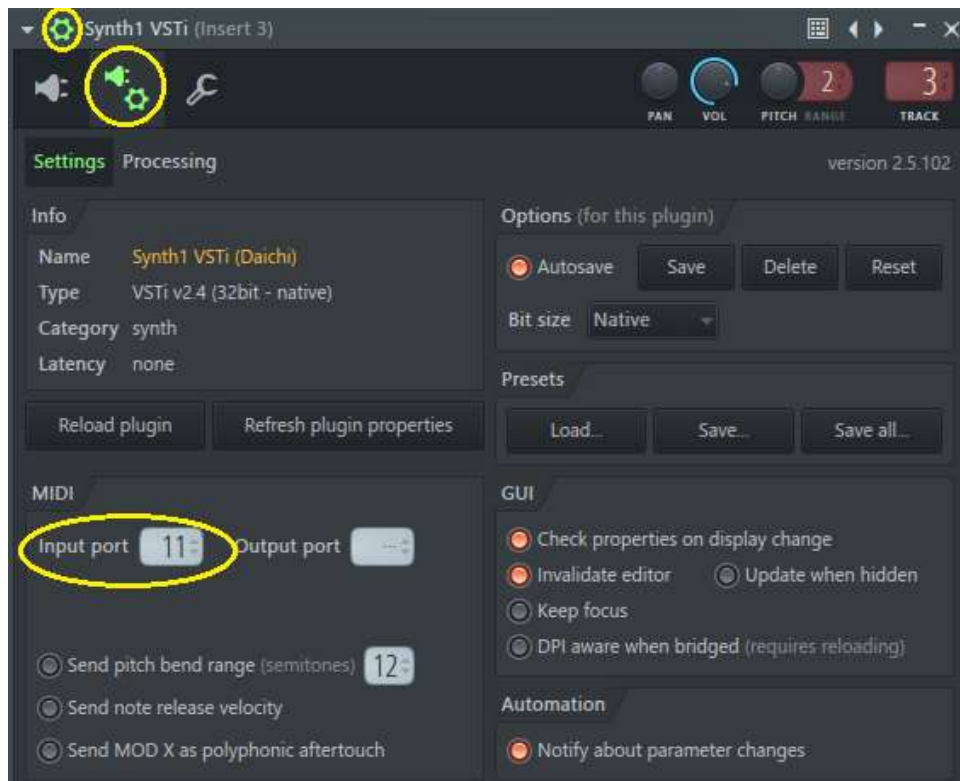
Click «SETTINGS» tab, set «Output port» to any value, not occupied by hardware MIDI devices and memorize this value (we will need it further):



Return to main plugin window:



Go to Fruity Wrapper settings of a VST synth (Synth1 in our example), set «Input port» to the value we memorized on the previous step:

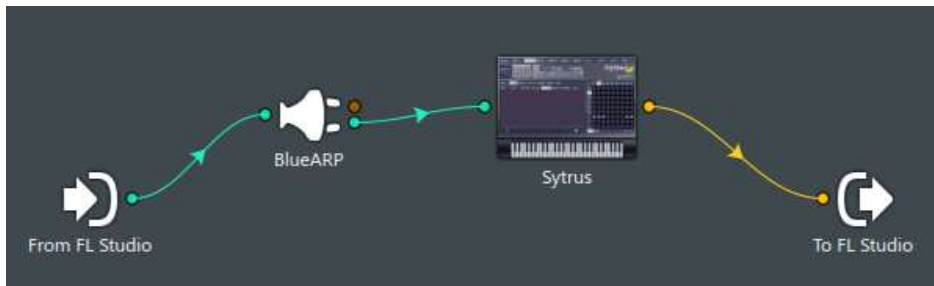


This way we tell FL Studio to route MIDI messages from BlueARP's MIDI output to Synth1's MIDI input. Just make sure this MIDI port is not occupied by hardware synths or other routings.

**Hint.** I usually reserve ports 1 – 10 for hardware MIDI devices and use numbers 11 and above for software routings.

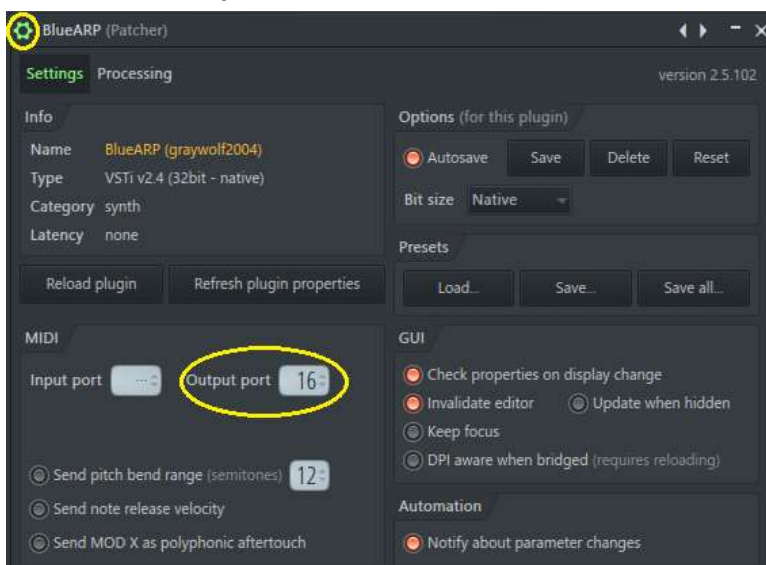
## FL Studio (Patcher method)

Add «Patcher» instrument to the track, inside Patcher add BlueARP and Fruity Generator of choice (Sytrus in our example), connect them as follows:



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

Double click BlueARP to open plugin window, go to wrapping settings and set output port to any unused number (**this is important**, otherwise it will not work).





## 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 list boxes).

For BlueARP track, set Monitor = «In».

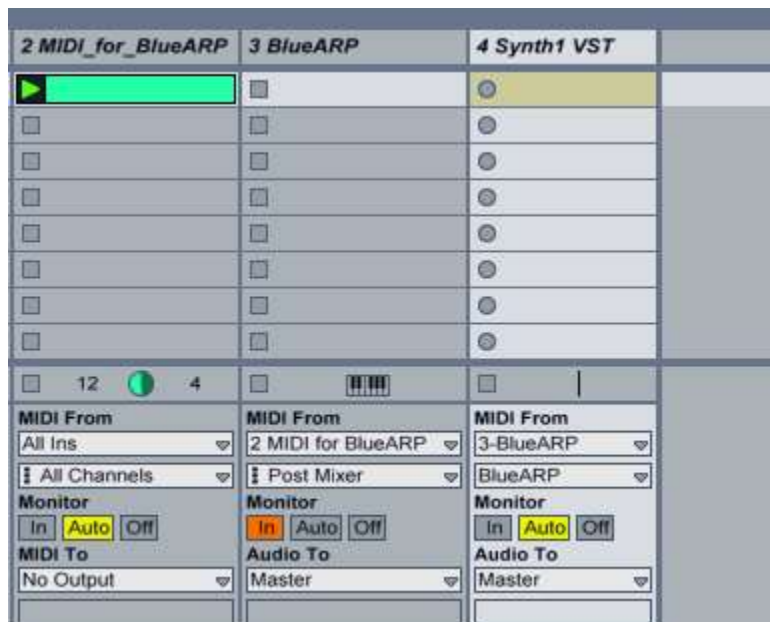
There's an issue - BlueARP will pick up MIDI from clips only when Monitor = «Auto», but it takes notes from Keyboard only with Monitor = «In». So, you have to constantly switch monitor from «In» to «Auto».

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 2 more tracks, one for BlueARP and another for a VST synth. Now we have 3 tracks in total:



For the track «*MIDI\_for\_BlueARP*», set Monitor = «Auto».

For «*BlueARP*» track, set MIDI From = «*MIDI\_for\_BlueARP*», Monitor = «In».

For «*Synth*» track, set MIDI From = «*BlueARP*» (both list boxes!), Monitor = «Auto».

Now, use «*MIDI\_for\_BlueARP*» track to record patterns and «*BlueARP*» track to play live.

If you want to drive hardware synth (connected via MIDI), use «External instrument» device (it's in «Live Devices» list) instead of a VST.

## Reaper 6.x and later

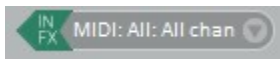
Add a track with VSTi synth of choice, click IN FX setting



Pick BlueARP from the list



Done!. IN FX button will become green



Click it to open BlueARP window.

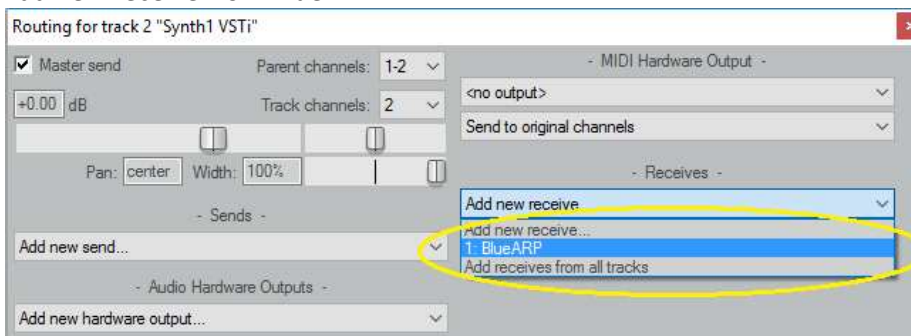
## Reaper older versions

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

Press ROUTE button on Synth1 track:



Add new receive from BlueARP:



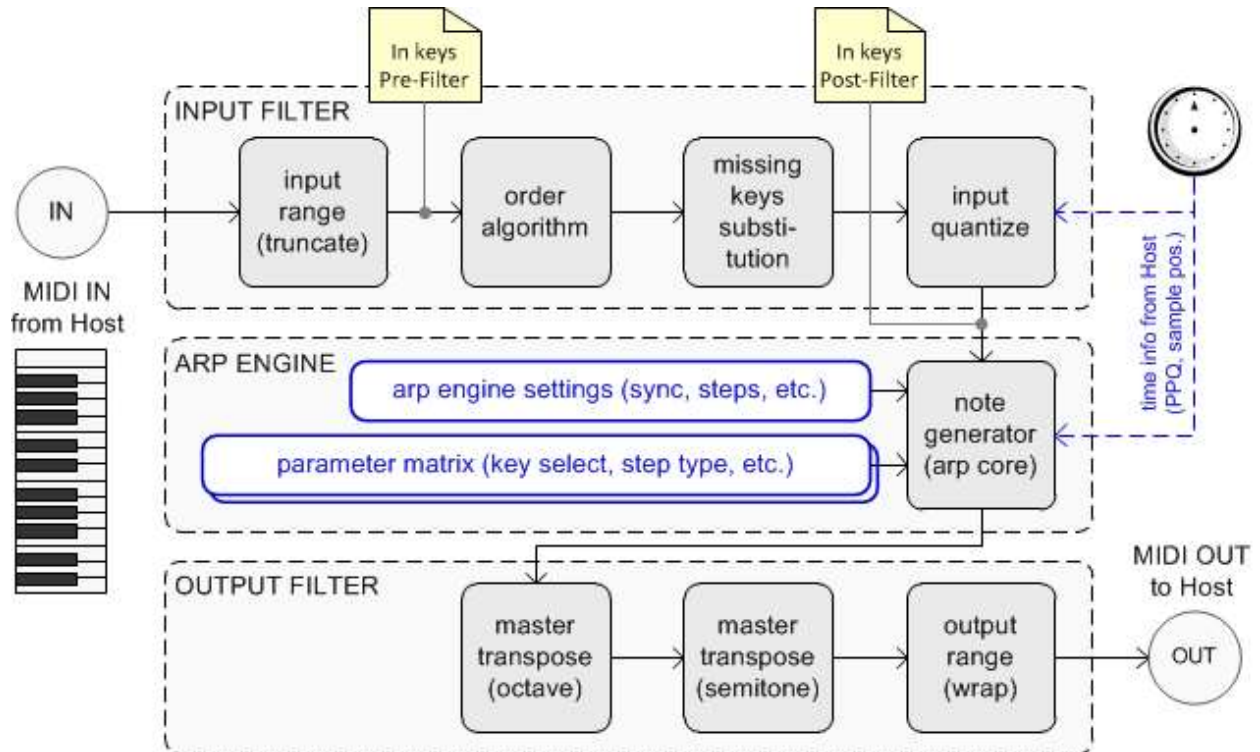
Now Synth1 receives notes from BlueARP, but you also need to prevent it from receiving notes directly from keyboard.



Set Input to None for Synth1 track:

## Signal flow

The picture below shows a basic data flow diagram for BlueARP. At the input BlueARP receives MIDI events from host. These are events of live pressing/releasing the keys on a MIDI keyboard or events coming from the 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 – 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 Input Filter, go to page 16).

«*In keys Post-Filter*» goes directly to the arp core.

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

ExtPos: -	IntPos: -	Step: -	In keys pre-filter: -, -, -, -, -	Note out: -
Param [Tag: ParamName] = 0			In keys post-filter: -, -, -, -, -	Detected chord: -

See Information panel description on page 31.

**Arp Engine** transforms keys coming from input filter into melodic phrases according to per-step settings in Value lanes (STEP TYPE, KEY SELECT and others). For example, «KEY SELECT» lane determines which key to take for the current step (k1 – key 1, k2 – key 2, fix – fixed key, etc.). «STEP TYPE» lane tells whether this step is a normal note (Nrm), the rest/sustaining note from the previous step (Rst) or muted (Off). Refer to page 27 for more information about Value lanes and Matrix editor.

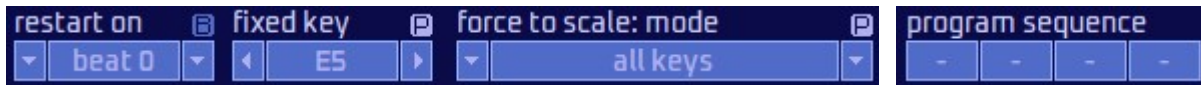
BlueARP has unique «*missing keys substitution*» feature. It works like this: when you have, for example, 4-keys pattern and play 2-key chord, by default («*missing keys substitution*» - «*don't play*») all steps with KEY SELECT = k3, k4 or K5 will be muted, cause there keys are not present at the input. If you select other options for «*missing keys substitution*», these missing keys will be substituted with the existing ones. There are several substitution algorithms, see page 16 for details.

**Output Filter** adds some post-processing to generated notes – octave / semitone transposition, wrapping notes to fit the given range. See page 20 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 was implemented with live performances in mind. See page 30 for more details.




## Interface

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



There are several ways to adjust the value:

- left-click and hold on the box, drag it up or down;
- place the pointer over the box, use mouse wheel to adjust the value;
- click   buttons to adjust the value or  button to select value from drop-down menu;

 /  /  marks next to control tell whether this particular parameter is saved with a bank **(B)**, program **(P)** or chain **(C)**. Global settings are stored in BlueARP.ini file and marked as **(G)**.

When you switch programs, **(B)** or bank-related parameters stay the same.

**(C)** or chain-related parameters are dependent on «current chain» setting (chains are described at page 14).

## Main window layout



Here are brief descriptions of GUI blocks. For more info, go to the respective chapters.

- (1) **Top panel** contains arp mode, midi in channel and midi out channel. All are bank-related (B). So, when you switch programs, these settings remain the same;
- (2) **Left panel** has 2 pages – ARP / MAIN and SETTINGS. ARP / MAIN page contains all step-independent arpeggiator settings like number of steps, synchronization, key sort order etc. Some are bank-related (B), some are program-related (P). SETTINGS page has midi filtering options, GUI settings and some other rarely changed stuff;
- (3) **Program browser** is there to select programs and to rename them;
- (4) **Main menu block** has MENU button (calls drop-down menu), page selector (for patterns longer than 16 steps), cyclic pattern shifts buttons and LEDs indicating which page is currently playing and which is being edited;
- (5) **Matrix editor** represents step-related values for the selected value lane;
- (6) **Value lanes** contain step-dependent pattern parameters. To select a lane, click on its caption. To adjust the value, drag the «value box» up and down or use mouse wheel;
- (7) **Program chains** allow you to chain several programs into one continuous sequence. «Current chain» parameter switches the chain, it can be automated;
- (8) **Info panel** - information on current position, beat, input and output keys;

## Block (1): Top panel



<b>arp mode</b>	Turning arpeggiator On or Off
values	<i>off, on, thru</i>
comments	<p><i>off</i> – BlueARP is inactive, all input keys are ignored</p> <p><i>on</i> – BlueARP is enabled, normal mode.</p> <p><i>thru</i> – BlueARP passes midi notes from input to output without arping, but some settings will still work (input range, output range, transpose, force to scale).</p> <p>In <i>thru</i> mode, you can use BlueARP as a real-time MIDI transpose tool and/or keyboard range filter.</p>
<b>midi in ch.</b>	input MIDI channel
values	<i>all, 1 .. 16</i>
comments	<i>all</i> – BlueARP will take MIDI input from all MIDI channels, <i>1 .. 16</i> – only from a given channel.
<b>midi out ch.</b>	output MIDI channel
values	<i>1 .. 16</i>
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 MIDI channels.



## Block (2): Left panel: ARP / MAIN

First if all, left panel has 2 pages – ARP/MAIN and SETTINGS.



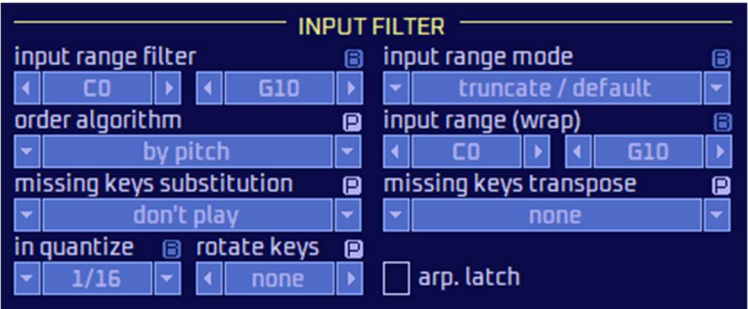
SETTINGS page contains rarely used bank-related settings, described on page 23 (you don't need to change them often, so I moved them to a separate page to save space on the left panel).

ARP/ MAIN is a primary page, it is divided into 4 blocks – «Input filter», «Arp engine», «Output filter» and «Pattern chains». Their controls are described in the following chapters.

*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-related and bank-related parameters, except the pattern itself. Program-related params have **(P)** mark, they may vary from program to program (for example, number of steps or gate time). Bank-related params have **(B)** mark, they are the same for all the programs in a given bank. For example, input range filter is bank-related, no need to set it for each program individually.

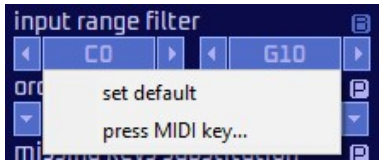
### Input Filter



**Input filter** is processing the input key list before it enters the arpeggiator «core» engine. We have «Input keys post-filter» key list at the output of this block. These keys go further into «Arp Engine» block.

<b>input range filter</b>	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 non-arpeggiated, it's controlled by «input range mode» setting.

**Hint.** Right-click value box and select «press MIDI key...» to set the value from your MIDI keyboard.



<b>input range mode</b>	modifies «input range filter» behaviour
values	truncate (default), pass thru (no arping)
comments	Sets the behavior of «input range filter» setting. In «pass thru (no arping)» mode, keys outside the range will be passed to the output non-arpeggiated.



<b>input range (wrap)</b>	range for input key «wrap-around»
values	<i>C0 .. G10 (MIDI notes 0 .. 127)</i>
comments	<p>Unlike «input range filter», 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 C3...C4. When you press keys <b>A2</b>, C3, E3, G3, <b>D4</b>, the processed keys will be <b>A3</b>, C3, E3, G3, <b>D3</b> (bold notes were wrapped into the range C3...C4).</p> <p>It's sonically useful when you play chords all over the keyboard, but want your bass line to sound right, not too low or too high.</p>
<b>order algorithm</b>	ordering (sorting) algorithm for input keys
values	<i>by pitch, by pitch desc, as played, as played desc, by velocity, by velocity desc, chord (normalized), chord (as played)</i>
comments	<p>Default setting is «by pitch» - pressed keys come into the arp engine in natural order, from left to right on the keyboard. It also means that «k1» in «KEY SELECT» lane 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.</p> <p>«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.</p> <p>«chord (as played)» behaves the same way, but inverted chord will stay inverted.</p>
<b>missing keys substitution</b>	missing keys substitution algorithm
values	<i>don't play, cyclic, first key, last key, fixed key</i>
comments	<p>When your pattern has more keys than you actually play, this setting will determine whether to mute these steps (<i>don't play</i>) or substitute missing keys with the existing ones.</p> <p>For example, you hold C5 and E5, while «KEY SELECT» lane has steps with «k1», «k2», «k3» and «k4».</p> <p>Info panel will show input keys pre-filter (before substitution) as «C5, E5, -, -, -». Key list post-filter (after substitution) will be, depending on this setting:</p> <ul style="list-style-type: none"> <li>• <i>don't play</i>      «C5, E5, -, -, -»</li> <li>• <i>cyclic</i>          «C5, E5, C5, E5, C5»</li> <li>• <i>first key</i>        «C5, E5, C5, C5, C5»</li> <li>• <i>last key</i>         «C5, E5, E5, E5, E5»</li> <li>• <i>fixed key</i>        «C5, E5, G5, G5, G5» («fixed key» = G5)</li> </ul>
<b>missing keys transpose</b>	additional transpose for substituted keys
values	<i>none, -1 octave, +1 octave</i>
comments	<p>Adds additional transposition for substituted missing keys.</p> <p>For the example above, if we set missing keys transpose to +1 octave, post-filter key list will be:</p> <ul style="list-style-type: none"> <li>• <i>don't play</i>      «C5, E5, -, -, -»</li> <li>• <i>cyclic</i>          «C5, E5, C6, E6, C6»</li> <li>• <i>first key</i>        «C5, E5, C6, C6, C6»</li> <li>• <i>last key</i>         «C5, E5, E6, E6, E6»</li> <li>• <i>fixed key</i>        «C5, E5, G6, G6, G6» («fixed key» = G5)</li> </ul>

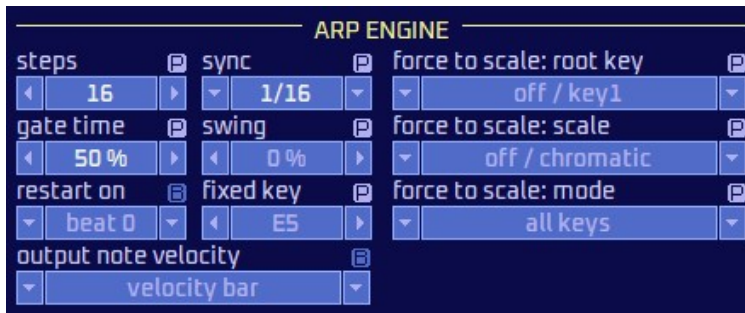
<b>in quantize</b>	input keys real-time quantization
values	<i>none, 1/16, 1/12, 1/8, 1/6, 1/4, 1/2, 1 bar, 2 bars</i>
Comments	Values are fractions of a bar (1/16 means 16th notes, 1/4 corresponds to 1 beat). For example, at value 1/4 BlueARP will capture pressed keys on the start of each beat.

**Hint.** When input quantize is on, you should press keys a little beforehand, because input keys need to be already captured when the next step/beat starts.

<b>rotate keys</b>	Rotate input keys
values	<i>-9 .. none .. 9</i>
comments	<p>When not «none», it will rotate input keys up or down before they enter the arp. For example, input keys are Cmaj chord: «C4, E4, G4».</p> <p>Output will be, depending on «rotate keys» value:</p> <ul style="list-style-type: none"> <li>• none: «C4, E4, G4»;</li> <li>• +1: «E4, G4, C5»;</li> <li>• +2: «G4, C5, E5»;</li> <li>• -1: «G3, C4, E4»;</li> <li>• -2: «E3, G3, C4»;</li> </ul> <p>In other words, rotating up one step means taking the lowest note and transposing it octave up. Rotating down means transposing an octave down the highest note.</p>

<b>arp. latch</b>	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, or to switch it off to free your hands from the keyboard to do some other stuff.

## Arp Engine

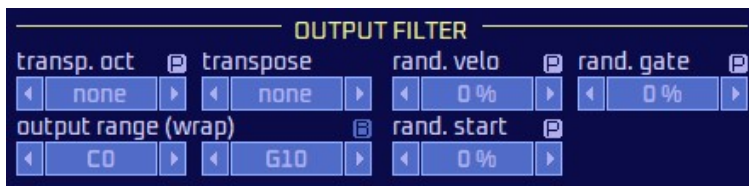


**Arp Engine** takes post-filter key list from the input filter (after fitting to range, missing keys substitution, quantize, etc.) and generates note pattern at the output, referring to MIDI clock and current song position from the Host.

<b>steps</b>	number of steps for current program
Values	0 .. 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. <b>steps = 0 and 1</b> are special modes, in this case BlueARP works as a MIDI thru (0 – simple thru, 1 – quantized thru). The purpose is to use this «MIDI thru dummy» program in chains to switch between «arpeggiated» and «midi thru» scenes.
<b>sync</b>	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».
<b>gate time</b>	note length, relative to step
values	1% .. 125%
comments	Sets generated note length as a fraction of a step length.
<b>swing</b>	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 the step length. For negative values, it will start earlier.
<b>restart on</b>	pattern restart trigger
values	beat 0, key, 1st key, play
comments	In default «beat 0» mode step number is always aligned to the song position given by host. When your song or pattern restarts in a DAW, BlueARP pattern will also restart. «play» mode is the same, but aligned to playback start position. 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.

<b>fixed key</b>	Fixed key value
values	<i>C0 .. G10 (MIDI notes 0 .. 127)</i>
comments	In «KEY SELECT» lane, 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.
<b>output note velocity</b>	Sets where to take velocity for generated notes
values	<i>velocity lane, input key, lane + input key</i>
comments	«lane + input key»: BlueARP takes output note velocity from VELOCITY lane and adjusts it to input note velocity (multiplying and normalizing them)
<b>force to scale: root key</b>	root key for «force to scale» mode
values	<i>off/key1, detect from chord, C, C#, D ... Bb, B</i>
comments	Works together with «force to scale: scale» parameter. You can either set a fixed root for a selected scale or let BlueARP detect it dynamically from the chord you play. BlueARP recognizes basic chords and chord inversions, so if you press (E4, A4, C5 - Am inverted), your root key will be <b>A</b> .
<b>force to scale: scale</b>	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, minor, harmonic minor, melodic minor, pentatonic Major, pentatonic minor, pentatonic neutral, pentatonic blues"</i>
comments	If you set anything except «off/chromatic», two things will happen: <ol style="list-style-type: none"> <li>1. BlueARP will fit output notes to the given scale (either all or only semi-transposed notes, depending on «force to scale: mode» parameter);</li> <li>2. «SCALE STEP» lane will transpose notes in scale steps. Say if your scale is C Major, you pressed <b>D4</b> and scale step=+1, the output note will be <b>E4</b>.</li> </ol> <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. From minor/major chords it will derive minor/major scales, for other chords like sus2, sus4 etc., BlueARP will try to derive an altered minor/major scale which will fit the given chord (for version 2.3.8 this feature is experimental).</p>
<b>force to scale: mode</b>	how to apply semitone transposition
values	<i>all keys, semi-transposed</i>
comments	Works together with "force to scale: scale" parameter. When set to <i>semi-transposed</i> , force to scale will not be applied to the steps with SCALE STEP = "-" (zero). This way you can still play out-of-scale notes.

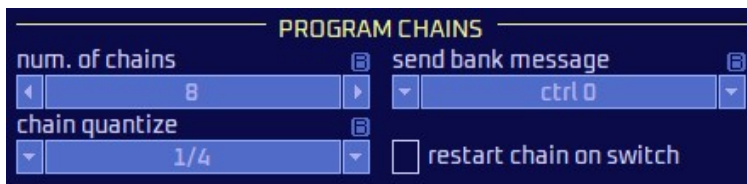
## Output Filter



**Output filter** performs some post-processing of generated notes – octave / semitone transposition, wrapping notes to fit the given range, applying randomization.

<b>transp. oct</b>	output transposition, octaves
values	-3 oct .. +3 oct
comments	It is program-related.
<b>transpose</b>	output transposition, semitones
values	-12 .. +12
comments	It is bank-related, because there's no sense to make this setting different for different programs.
<b>output range (wrap)</b>	Range for output notes (wrapping)
values	C0 .. G10 (MIDI notes 0 .. 127)
comments	Notes outside the range will be wrapped (octave-transposed up or down to fit the range). Works just like «input range (wrap)», but for output notes.
<b>rand. velo</b>	randomize output note velocity
values	0% .. 100%
comments	Add random value (positive or negative) to generated note velocity.
<b>rand. gate</b>	randomize output note gate time
values	0% .. 100%
comments	Add random value (both positive and negative) to generated note length.
<b>rand. start</b>	randomize output note start time
values	0% .. 100%
comments	Add random value (positive only) to generated note start time.

## Program Chains



Relates to "**Block (7) Program chains**" panel.

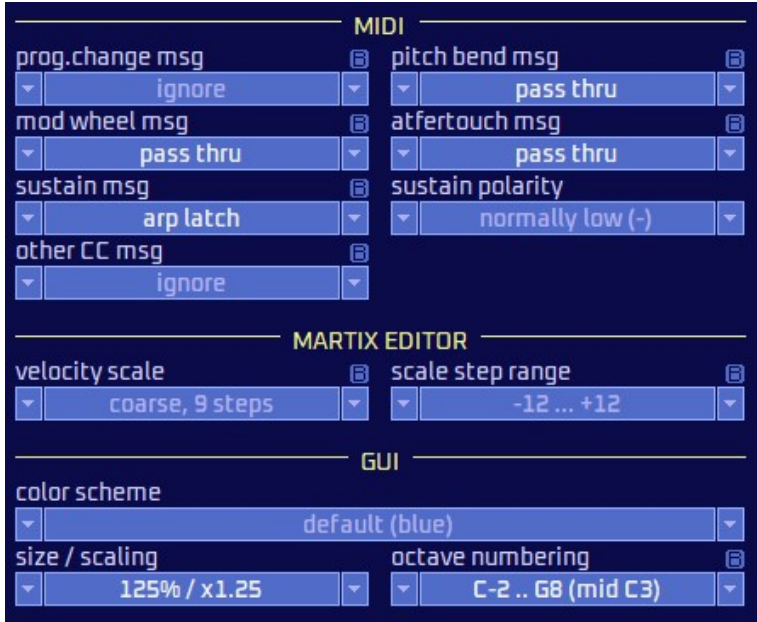
<b>num. of chains</b>	sets maximum value for «current chain» parameter
values	1 .. 16
comments	To switch chains with a midi controller, you need to automate «current chain» parameter. If you use a knob for this, setting «num. of chains» to the appropriate value will utilize full rotation range of this knob.
<b>chain quantize</b>	input quantization for chain switching
values	none, 1/16, 1/12, 1/8, 1/6, 1/4, 1/2, 1 bar, 2 bars
comments	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 is the default setting.
<b>send bank message</b>	selects bank/patch change MIDI message format
values	ctrl 0, ctrl 32, ctrl 0+32
comments	Relevant for controlling hardware synths, some VST synths will also react to this message. Sylenth1 does, for example. When you switch chains, BlueARP may send program/bank change to its MIDI output if «bank num» and «patch num» parameters are not empty. Hardware synths use different bank change message formats. If the default one doesn't work for you (synth doesn't switch banks, only patches), try other options.
<b>restart chain on switch</b>	restart chain from the beginning after chain switch
values	On, Off
comments	When checked, chain always starts from the beginning after chain switch (otherwise, in restart on «beat 0» mode, chain step is calculated from song position given by host)

## Block (2): Left panel: SETTINGS

Hit SETTINGS button on the left panel to call this page.



This page contains rarely used bank-related and global settings. They are on a separate page to save space on the main panel. You don't need to change them often.



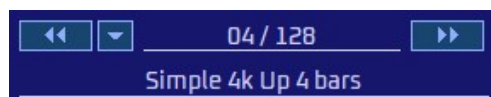
- MIDI** block contains settings for MIDI message filtering.
- MATRIX EDITOR** block defines matrix editor behavior.
- GUI** block has some GUI-related settings.

<b>prog.change msg</b>	how to respond to incoming Program Change MIDI message
values	<i>ignore, set own program, pass thru</i>
comments	«set own program»: BlueARP will set its internal program in response to Program CC message. «pass thru»: BlueARP will pass this message to its MIDI out (= to VST plugin it is connected to).
<b>pitch bend msg</b>	how to respond to incoming Pitch Bend MIDI message
values	<i>ignore, pass thru</i>
comments	«pass thru»: BlueARP will pass this message to its MIDI out (= to VST plugin it is connected to).
<b>mod wheel msg</b>	how to respond to Modulation Wheel MIDI message
values	<i>ignore, pass thru</i>
comments	«pass thru» - BlueARP will pass this message to its MIDI out.
<b>aftertouch msg</b>	how to respond to incoming aftertouch MIDI message
values	<i>ignore, pass thru</i>
comments	«pass thru»: BlueARP will pass this message to its MIDI out.
<b>sustain msg</b>	how to respond to sustain MIDI message (CC 64)
values	<i>ignore, pass thru, sustain, arp latch</i>
comments	«pass thru»: BlueARP will pass this message to its MIDI out. «sustain»: BlueARP will sustain input notes in a normal way, just like any other synth would do «arp latch»: sustain message is linked to «arp latch» parameter, with respect to «sustain polarity» value.



<b>sustain polarity</b>	sustain pedal polarity for «sustain msg» setting
values	<i>normally low (-), normally high (+)</i>
comments	Normally low (-) means that in released state it should be value 0.
<b>other CC msg</b>	sets how to respond to incoming CC MIDI messages
values	<i>ignore, pass thru</i>
comments	The same as other MIDI filters, but applies to all other CC messages not mentioned before.
<b>velocity scale</b>	sets velocity accuracy for VELOCITY lane
values	<i>«coarse, 9 steps», «fine, 128 steps»</i>
comments	Select "fine, 128 steps" if you want to make fine velocity adjustments, otherwise it will go like 16, 32, 48, etc.
<b>scale step range</b>	sets value span for SCALE STEP lane
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 «0...+12», «-7...+7» or «0...+7» for easier adjustment.
<b>color scheme</b>	sets skin / color theme
values	<i>default (blue) and others</i>
comments	Color schemes are stored in *.ini files in \skins sub-directory. On windows it is in plugin directory, on Mac – inside the bundle). Selected color scheme index is stored in BlueARP.ini file in user directory: <b>Windows:</b> C:\Users\<user>\AppData\Roaming\BlueARP <b>OSX:</b> c:/Users/<username>/Library/Application Support/BlueARP
<p><b>Hint.</b> When you load BlueARP for the first time, it will create this directory and BlueARP.ini inside it. Ini file it was placed here, cause plugin directory doesn't usually grant write permission to the plugin.</p>	
<b>size / scaling</b>	sets GUI size
values	<i>100%, 125%, 150%</i>
comments	Adjusts GUI size.
<b>octave numbering</b>	sets one of note naming conventions
values	<i>«C-2 .. G8 (mid C3)», «C-1 .. G9 (mid C4)», «C0 .. G10 (mid C5)»</i>
comments	It tells BlueARP how to display notes or which key is the middle - C3, C4 or C5.

## Block (3): Program browser



Use buttons to navigate through the programs in a current bank.

Bank contains 128 programs, so you can configure up to 128 arpeggiator patterns, they will be all saved with your project file.

To change program name, click on it, type in new name and hit enter or click somewhere outside this area.



## Block (4): Main menu and pattern controls



**MENU** button calls drop-down menu with Bank load/save, Program load/save and some other functions.

**page** buttons are necessary when your pattern is longer than 16 steps, so it doesn't fit single screen. There are 2 small LED lanes underneath, upper one shows selected page (page being edited), lower one – page being played.

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

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

**Pattern shift** buttons perform cyclic step shifting (rotation of the pattern). 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 pattern right, the last step won't disappear but will «jump» to the beginning.

**Main menu** includes the following items:

<b>Bank</b>	bank contains entire BlueARP state, except global (G) settings
Load from file (*.fxb)	Load bank from file, current state will be overwritten
Save to file (*.fxb)	Save bank to file
Initialize	Initialize all programs in a current bank

<b>Program</b>	load, save and copy/paste programs
Load from file (*.fxp)	Load program from file, current program will be overwritten
Save to file (*.fxp)	Save current program to file
Copy ...	Memorize current program as a source for copy/paste operation.
Cut ...	Memorize current program as a source for cut/paste operation. On paste, source program will be deleted
Paste	Paste program at a current location ("Copy ..." or "Cut ..." should be done before). Paste overwrites target program.
Insert initial	Insert initial program at the current location. Current and all the following programs will be shifted to the right to make space for the new program, the last (128 <sup>th</sup> ) program will be lost.
Delete	Delete current program. The remaining programs will be shifted to the left to fill the gap.
Initialize	Initialize the current program
Shuffle steps	Randomly shuffle steps in the current program

<b>Chain</b>	copy/paste chains
Copy ...	Memorize current chain page as a source for copy/paste operation.
Cut ...	Memorize current chain as a source for cut/paste operation. On paste, source chain will be deleted
Paste	Paste chain at a current location ("Copy ..." or "Cut ..." should be done before). Paste overwrites target chain.
Insert initial	Insert initial chain at the current location.
Delete	Delete current chain.
Initialize	Clear current chain data
Initialize all chains	Clear all chains data

<b>Page</b>	copy/paste chains
Copy ...	Memorize current pattern page as a source for copy/paste operation.
Paste	Paste pattern page at a current location ("Copy ..." should be done before). Paste overwrites the target chain.
Initialize	Initialize all steps for the pattern page.

<b>Debug info</b>	various information
Open BlueARP.ini location	Opens BlueARP.ini file location. BlueARP.ini holds global settings like GUI scale, GUI skin index, octave numbering.
pGraphics->GetGUIAPI()	Show selected GUI API
pGraphics->PluginPath()	Show path to the plugin file
pGraphics->HostPath()	Show path to host application
ArpEngine->PatchVer_loaded	Show currently loaded bank format index

<b>Open Manual (pdf)</b>	available versions of the manual
English (EN)	Opens "BlueARP_Manual_vNNN_EN.pdf" file, where NNN stands for version. File should be located in the same folder as the plugin, it is included into the .zip installation package.
Other languages	Opens the manual in another language by a direct web link

<b>Make a donation</b>	link to support BlueARP development via PayPal donation
------------------------	---

<b>Developer's website</b>	link to developer's website with the latest updates for BlueARP
----------------------------	---

## Block (5): Matrix editor

k5																
k4																
k3			k3		k3		k3					k3		k3		k3
k2		k2		k2		k2			k2		k2					
k1	k1						k1		k1			k1		k1		
Root																
Fixed																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Matrix editor allows you to adjust current value lane values in a more friendly graphic way.

So, you can adjust a step-related value 2 ways – either in a matrix editor or on a value lane itself.

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.



Greyed-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 lanes

VELOCITY	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96
GATE TIME	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
STEP TYPE	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm
KEY SELECT	k1	k2	k3	k4	k1	k2	k3	k4	k1	k2	k3	k4	k1	k2	k3	k4
OCTAVE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCALE STEP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Value lanes contain step-related pattern parameters. Selected value lane is also shown in Matrix editor (to select it, click lane caption).

To adjust value for a certain step, click on it and drag up or down to change the value. Or, use mouse wheel to do the same thing.

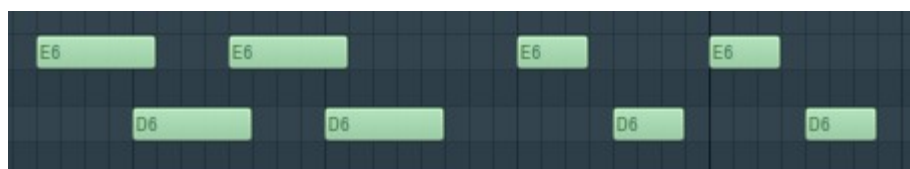
Yellow indicator (  or  ) next to «OCTAVE» and «KEY SELECT» labels switch the lane between **monophonic** and **polyphonic** mode. In polyphonic mode, you can set several values at once (either keys or octaves) on Matrix editor.

«GATE TIME» can be switched to «CHANNEL» or channel per step mode via  indicator:

CHANNEL	2	-	2	-	-	1	-	-	-	2	-	-	-	4	-	-	-
---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



See descriptions for each value lane below.

<b>VELOCITY</b>	Velocity value for each step
values	0, 16, 32 .. 127
comments	Default value is 96. Use it to set velocity 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.
<b>GATE TIME (mode 1)</b>	Gate time multiplier for each step
Values	1/16, 1/8, 1/4, 1/2, -, 2, 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.
<b>CHANNEL (mode 2)</b>	Modify output MIDI channel for each step
Values	-, 1 .. 16
comments	When not "-", output midi channel will be changed to the specified value for a step. Use this with multi-timbral synths to create complex textures/arpeggios with different sounds for various steps.
<b>STEP TYPE</b>	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 Rnd – Random, picks up random key from input key list
comments	«Rst» step means that this step continues the note from the previous step. You may chain several «Rst» steps together to make longer notes. «Tie» option may be tricky and not self-describing. Its 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 (press key1, press key2, release key1), sound pitch will glide between the notes, but not when you press them with gaps (see picture below). When you configure the synth this way, «Tie» steps will create glides.





«Tie» steps

«Nrm» steps

<b>KEY SELECT</b>	Input key selection for the given step
values	<i>Fixed</i> – use fixed key from Arp Engine settings <i>Root</i> – root key from detected chord, key1 if no chord detected <i>k1..k5</i> – take keys №1..5 from key list (post-filter)
comments	Tells 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 <b>monophonic</b> mode you can only select one key for a step or all keys at once with STEP TYPE = Chord. In <b>polyphonic</b> 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» and use BlueARP as a step sequencer, or set some steps to «fixed» to create variations.

<b>SCALE STEP</b>	Semitone/Scale step transposition for each step
values	-12 .. +12
comments	Depends on «force to scale: scale» parameter. When the latter is «off/chromatic», this will work as a semitone transposition. Otherwise, it will transpose output note with respect to the selected scale.

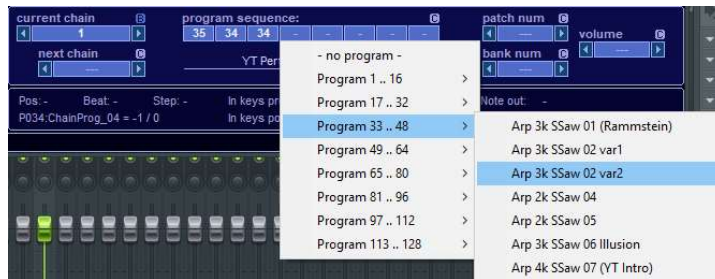
<b>OCTAVE</b>	Octave transposition for each step
values	-3, -2, -1, 0, +1, +2, +3
comments	It's convenient for bass lines, where the steps are usually transposed for the whole octaves. Yellow label next to OCTAVE caption (  or  ) toggles between monophonic and polyphonic mode. In <b>monophonic</b> mode all keys for a given step are transposed by octaves. In <b>polyphonic</b> 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 chains deliver the possibility to chain several programs (patterns) together into a longer «super-pattern». It was implemented mostly with live performances in mind.

**Program sequence** lane holds numbers of chained programs for a current chain. Right-click program sequence slot to select program for a particular chain step:



Program sequence is linked to particular chain (current chain parameter), switching current chain calls up another program sequence. Chain-related or **(C)** – parameters only make sense with current chain is selected (value is not «- -»).

<b>current chain</b>	set current chain
values	---, 1, 2, ... <i>number of chains (up to 16)</i>
comments	Current chain parameter can be automated; its maximum value is set by "num. chains" parameter on the left panel.

**Hint.** Pay attention to «restart chain on switch» setting on the left panel. When On, switched chain will always start from the beginning (1<sup>st</sup> step of the program sequence).

<b>next chain</b>	next chain auto-switch
values	---, <i>caller, caller-1, caller+1, chain 1, ... chain 16</i>
comments	Allows you to automatically jump to another chain after current chain plays once. The options include: «caller» - switch back to the chain it was invoked from; «caller-1», «caller+1» - the same, but with the shift to the «caller» chain; «chain 1» ... «chain 16» - switch to particular chain after this chain ends;

<b>patch num, bank num</b>	send bank\program change on chain switch
values	---, 0 ... 127
comments	If specified, BlueARP will send program\bank change midi message to the connected synth each time current chain is changed (with respect to chain quantize).

<b>volume</b>	send volume change when chain switches
values	---, 0 ... 127
comments	As previous, BlueARP will send volume change MIDI message to the connected synth each time current chain is changed.

## Block (8): Information panel

ExtPos: -	IntPos: 045	Step: 05	In keys pre-filter: D4, F4, G4, -, -	Note out: F4
P031:ChainProg_01 = 33 / 34	In keys post-filter: D4, F4, G4, D4, D4			Detected chord: G 7

Shows current beat, step and some other information:

- **ExtPos:** song position, reported by host. For *restart on = beat 0*, it is used as a reference for step position;
- **IntPos:** internal song position (with respect to looping);
- **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.

**Hint.** Lower left label «P031: ChainProg\_01 = 33 / 34» gives information about last changed parameter and associated value. First number (33) represents internal value, second number (34) – midi CC value.

# FAQ / Troubleshooting

## Installing BlueARP

*«Unrecognized Developer» error message when trying to run BlueARP on OSX (reported for OSX Catalina, probably the same with earlier versions)*

By default, OSX disables to run apps from developers not registered in apple store. But you can manually change that.

Go to System Settings -> Security & Privacy, change «Apps downloaded from» setting to «Anywhere». Alternatively, if you see message «BlueARP was blocked because...» message, you can press «Open anyway» to create exception from BlueARP only (which is better security-wise).

## Sync & Timing issues

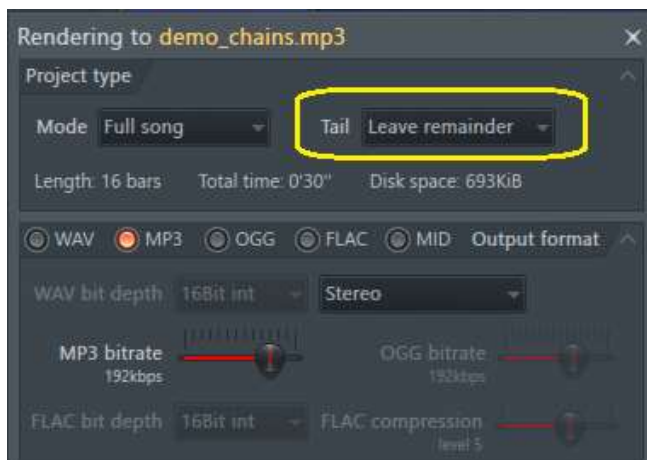
*Output note timing is not perfect, like they are delayed by random values.*

Check audio settings in your DAW. Your audio buffer size should be 256 samples or less, 128 is recommended. 256 samples will give maximum inaccuracy of 5ms at 48kHz ( $5 / 48000 \approx 0.005s$ )

## Rendering audio

*When trying to render a project in FL Studio, only 1<sup>st</sup> note comes out of BlueARP, others are missing.*

In rendering settings (you get there automatically, when you call Export -> mp3 or whatever) change «Tail» option to «Leave remainder»



*solution provided by Saif Sameer*



## 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@gmail.com](mailto:graywolf2004@gmail.com)



Oleg Mikheev aka Graywolf, © 2012-2021