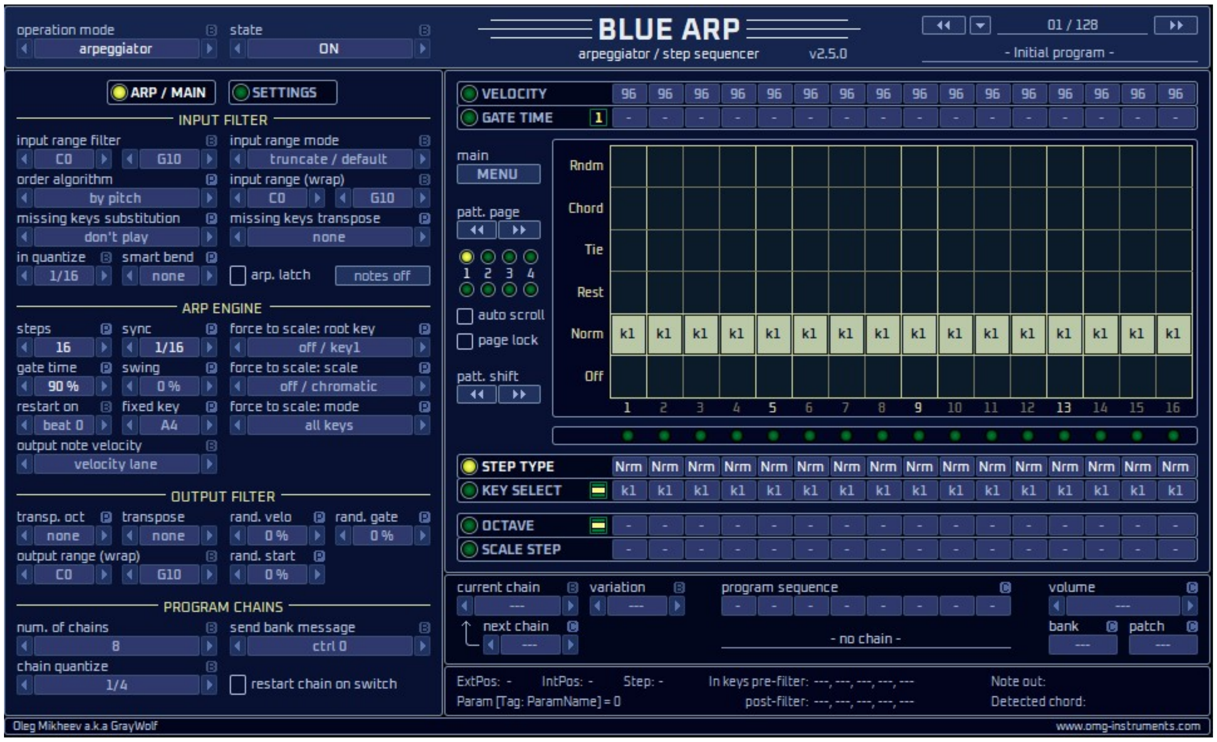
BLUE ARP

Operation Manual

Corresponds to BlueARP v2.5.0



BLUE ARP
arpeggiator / step sequencer v2.5.0
01 / 128
- Initial program -

operation mode: arpeggiator
state: ON

ARP / MAIN
SETTINGS

INPUT FILTER
input range filter: C0 – G10
input range mode: truncate / default
order algorithm: by pitch
input range (wrap): C0 – G10
missing keys substitution: don't play
missing keys transpose: none
in quantize: 1/16
smart bend: none
arp. latch
notes off

ARP ENGINE
steps: 16
sync: 1/16
force to scale: root key: off / key1
gate time: 90 %
swing: 0 %
force to scale: scale: off / chromatic
restart on: beat 0
fixed key: A4
force to scale: mode: all keys
output note velocity: velocity lane

OUTPUT FILTER
transp. oct: none
transpose: none
rand. velo: 0 %
rand. gate: 0 %
output range (wrap): C0 – G10
rand. start: 0 %

PROGRAM CHAINS
num. of chains: 8
send bank message: ctrl 0
chain quantize: 1/4
restart chain on switch

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VELOCITY	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96
GATE TIME 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Norm	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1
STEP TYPE	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm
KEY SELECT	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1
OCTAVE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SCALE STEP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

current chain: ---
variation: ---
program sequence: - no chain -
volume: ---
next chain: ---
bank: ---
patch: ---

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

Oleg Mikheev a.k.a GrayWolf
www.omg-instruments.com

Pattern Arpeggiator / Step Sequencer / Drum Sequencer

VST/AU midi-FX plug-in for Windows & OSX

by Oleg Mikheev aka Graywolf

http://www.omg-instruments.com

Feb 2023

Table of Contents

Introduction

BlueARP is a programmable pattern arpeggiator / step sequencer, it comes as a VST or MIDI-FX plug-in for Windows and OSX. 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) 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, New Wave etc.), but its usage it not limited to these genres.

From 2022, BlueARP has its hardware counterpart called BlueARP DM (Desktop Module), find out more at www.omg-instruments.com.

Compatibility info

Formats:	VST 32-bit (windows only), VST 64-bit, AU MIDI-FX 64-bit (for Logic Pro X)
OS:	OSX (10.9 or later, tested on 12.7.3), 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»
- Chains can be switched on the fly with real-time quantization;
- 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 note quantization;
- Chord recognition, you can crease chord-based patterns;
- Input range setting for keyboard-split performances;
- Separate settings for octave and semitone per step transpose;
- Configurable color schemes (skins);
- Dedicated 'drum sequencer' mode since v2.5.0
- Chord-driven chain switching since v2.5.0

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, Garage Band or Main Stage), re-scan Audio Unit plugins folder. BlueARP should appear in plugin list and it is new ready to use.

If it doesn't appear in the plugin list, try to reboot or logout/login. I got some complaints on this in 2022 and it seems like Logic now needs system reboot to see new installed MFX plugins.

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 deleting them.

To solve this problem, try the following:

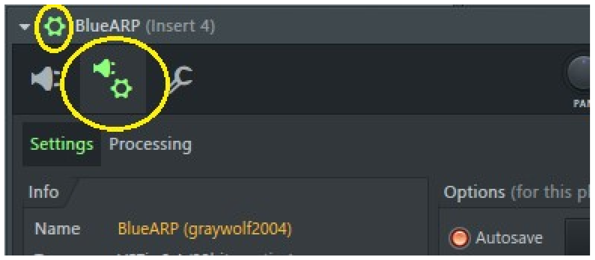
- Manually delete all "Oxanuim" fonts from your system
- Reboot
- Try to delete the folder again

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.

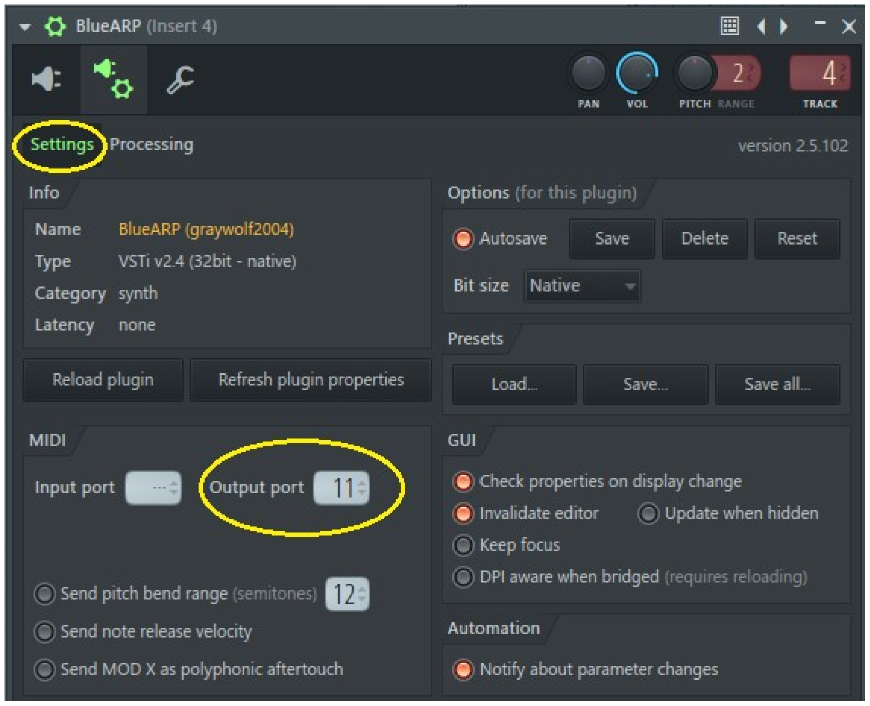
FL Studio (Fruity Wrapper method)

Load BlueARP, click the buttons as shown on picture:



Name BlueARP (graywolf2004)

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



PITCH RANGE 2
TRACK 4
version 2.5.102

Name	BlueARP (graywolf2004)
Type	VSTi v2.4 (32bit - native)
Category	synth
Latency	none

Autosave
Bit size Native

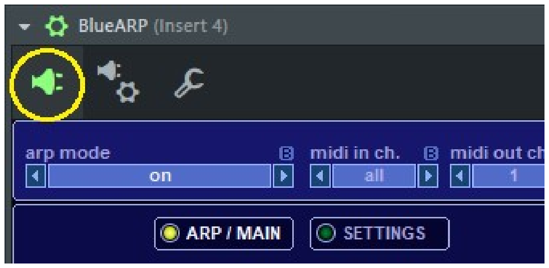
Input port ---
Output port 11

Send pitch bend range (semitones) 12
Send note release velocity
Send MOD X as polyphonic aftertouch

Check properties on display change
Invalidate editor
Update when hidden
Keep focus
DPI aware when bridged (requires reloading)

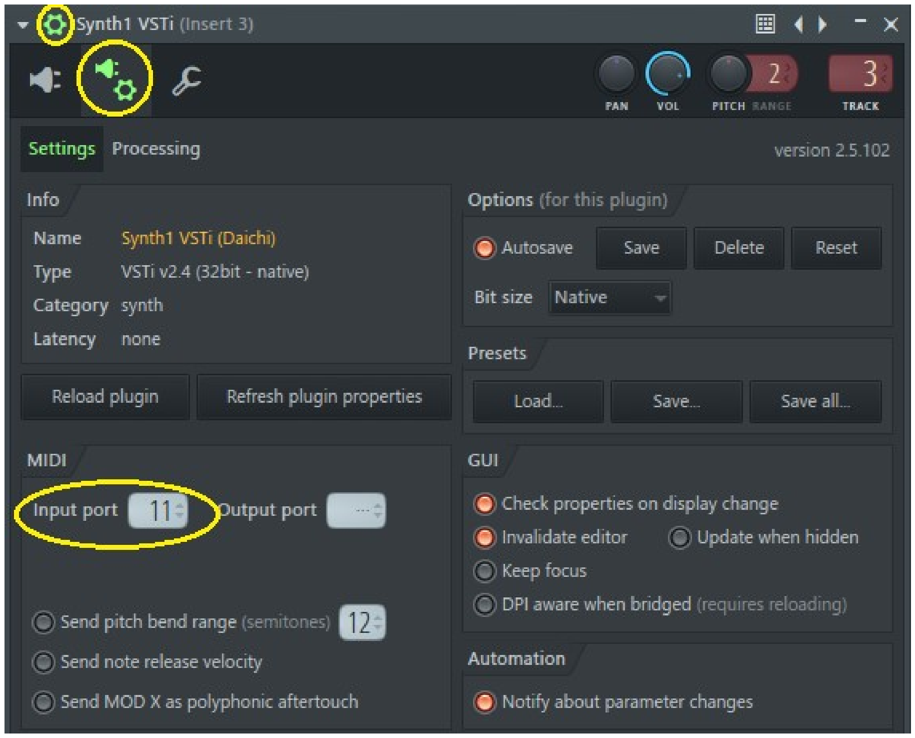
Notify about parameter changes

Return to main plugin window:



arp mode on
midi in ch. all
midi out ch 1

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:


Synth1 VSTi (Insert 3)

PAN VOL PITCH 2 RANGE 3 TRACK

Settings Processing version 2.5.102

Info

Name	Synth1 VSTi (Daichi)
Type	VSTi v2.4 (32bit - native)
Category	synth
Latency	none

Reload plugin | Refresh plugin properties

MIDI

Input port 11 Output port ---

- Send pitch bend range (semitones) 12
- Send note release velocity
- Send MOD X as polyphonic aftertouch

Options (for this plugin)

- [✓ Autosave] Save Delete Reset
- Bit size Native

Presets

Load... Save... Save all...

GUI

- [✓ Check properties on display change]
- [✓ Invalidate editor] Update when hidden
- Keep focus
- DPI aware when bridged (requires reloading)

Automation

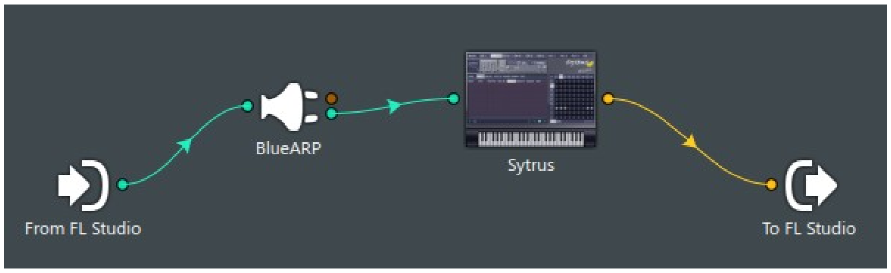
- [✓ Notify about parameter changes]

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.*

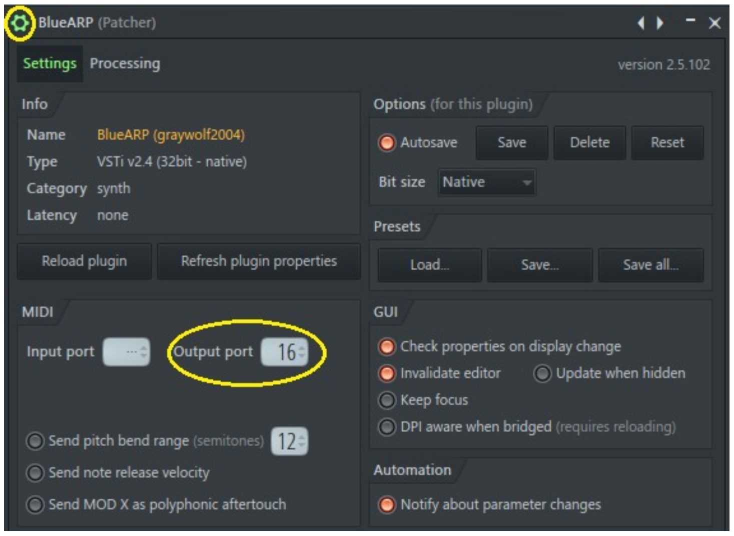
FL Studio (Patcher method)

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 cluck 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).


BlueARP (Patcher)

Settings Processing version 2.5.102

Info

Name	BlueARP (graywolf2004)
Type	VSTi v2.4 (32bit - native)
Category	synth
Latency	none

Reload plugin | Refresh plugin properties

MIDI

Input port --- Output port 16

Send pitch bend range (semitones) 12

Send note release velocity

Send MOD X as polyphonic aftertouch

Options (for this plugin)

Autosave | Save | Delete | Reset

Bit size Native

Presets

Load... | Save... | Save all...

GUI

Check properties on display change

Invalidate editor | Update when hidden

Keep focus

DPI aware when bridged (requires reloading)

Automation

Notify about parameter changes

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».

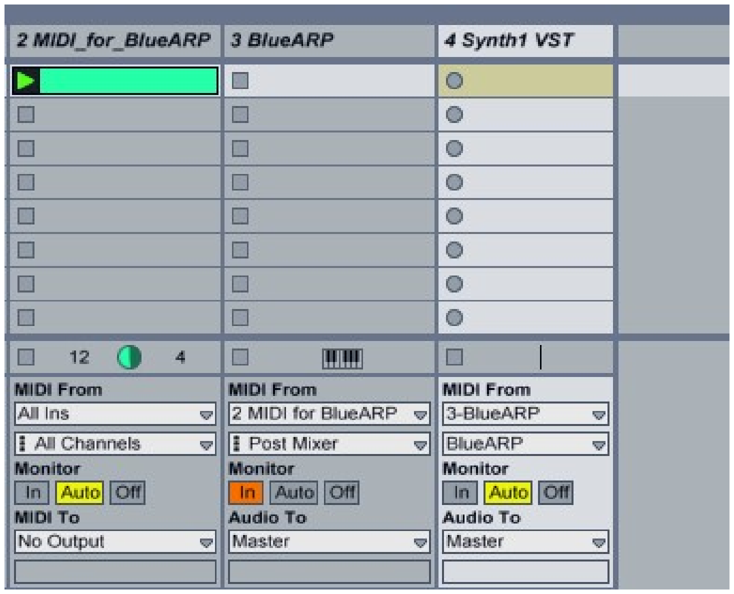
There's an issue - BlueARP will pick up MIDI from clips only when Monitor = «Auto», but it takes notes from Keyboard only with Monitor = «In». So, you have to constantly switch monitor from «In» to «Auto».

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:



2 MIDI_for_BlueARP	3 BlueARP	4 Synth1 VST
MIDI From	MIDI From	MIDI From
All Ins	2 MIDI for BlueARP	3-BlueARP
All Channels	Post Mixer	BlueARP
Monitor	Monitor	Monitor
In [✓ Auto] Off	[✓ In] Auto Off	In [✓ Auto] Off
MIDI To	Audio To	Audio To
No Output	Master	Master

For the track «*MIDI_for_BlueARP*», set Monitor = «Auto».

For «*BlueARP*» track, set MIDI From = «*MIDI_for_BlueARP*», Monitor = «In».

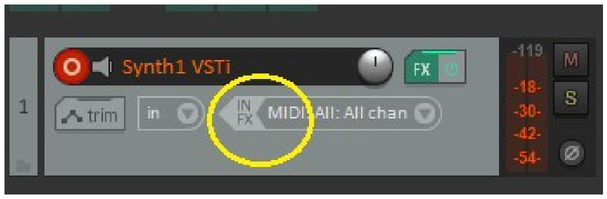
For «*Synth*» track, set MIDI From = «*BlueARP*» (both list boxes!), Monitor = «Auto».

Now, use «*MIDI_for_BlueARP*» track to record patterns and «*BlueARP*» track to play live.

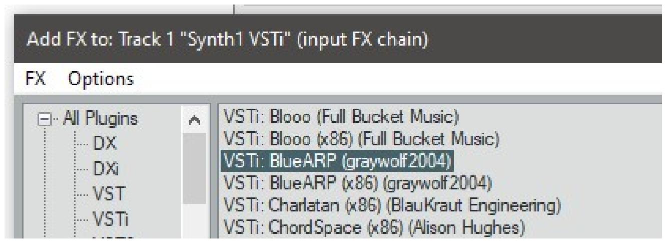
If you want to drive hardware synth (connected via MIDI), use «External instrument» device (it's in «Live Devices» list) instead of a VST.

Reaper 6.x and later

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



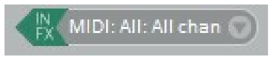
Pick BlueARP from the list


Add FX to: Track 1 "Synth1 VSTi" (input FX chain)

FX Options

VSTi: Blooo (Full Bucket Music)
VSTi: Blooo (x86) (Full Bucket Music)
VSTi: BlueARP (graywolf2004)
VSTi: BlueARP (x86) (graywolf2004)
VSTi: Charlatan (x86) (BlauKraut Engineering)
VSTi: ChordSpace (x86) (Alison Hughes)

Done!. IN FX button will become green

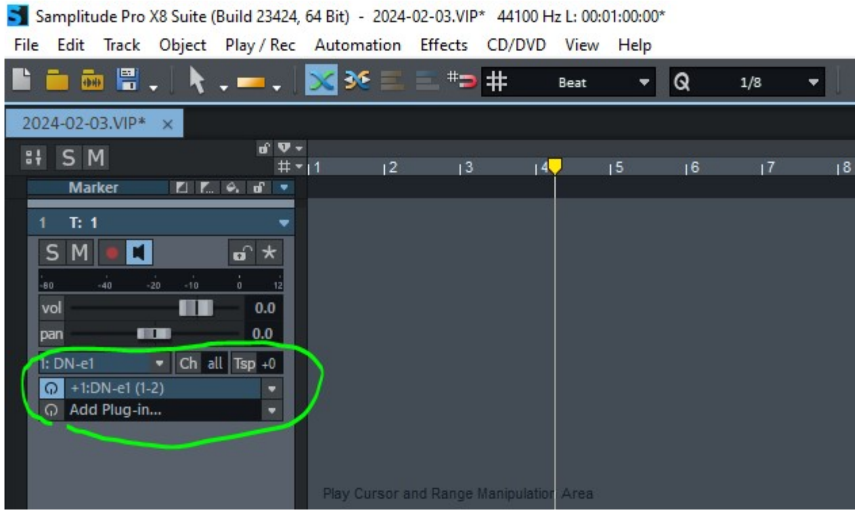


Click it to open BlueARP window.

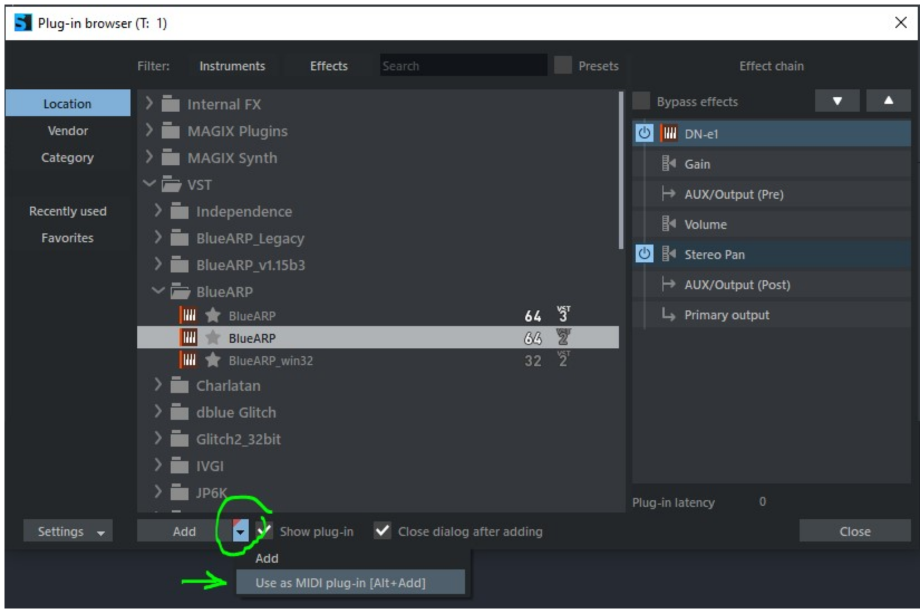
Samplitude Pro X8

Newer versions of Samplitude support MIDI plugins as a track insert (no need to add separate track for the BlueARP), this is the newer «MIDI insert» method described.

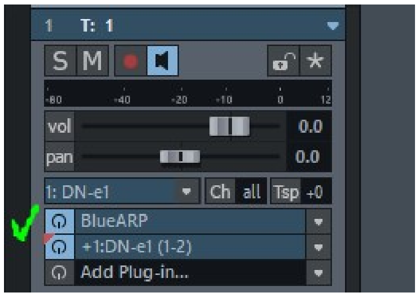
First, arm new or existing track with an instrument of choice:


Samplitude Pro X8 Suite (Build 23424, 64 Bit) - 2024-02-03.VIP* 44100 Hz L: 00:01:00:00*
File Edit Track Object Play / Rec Automation Effects CD/DVD View Help
Beat
1/8
2024-02-03.VIP*
Marker
1 T: 1
vol 0.0
pan 0.0
1: DN-e1
Ch all Tsp +0
+1:DN-e1 (1-2)
Add Plug-in...
Play Cursor and Range Manipulation Area

Next, click «Add Plug-in…», select BlueARP from the list. Instead of clicking «Add» button below, select «Use as MIDI Plug-in» from the drop-down list.



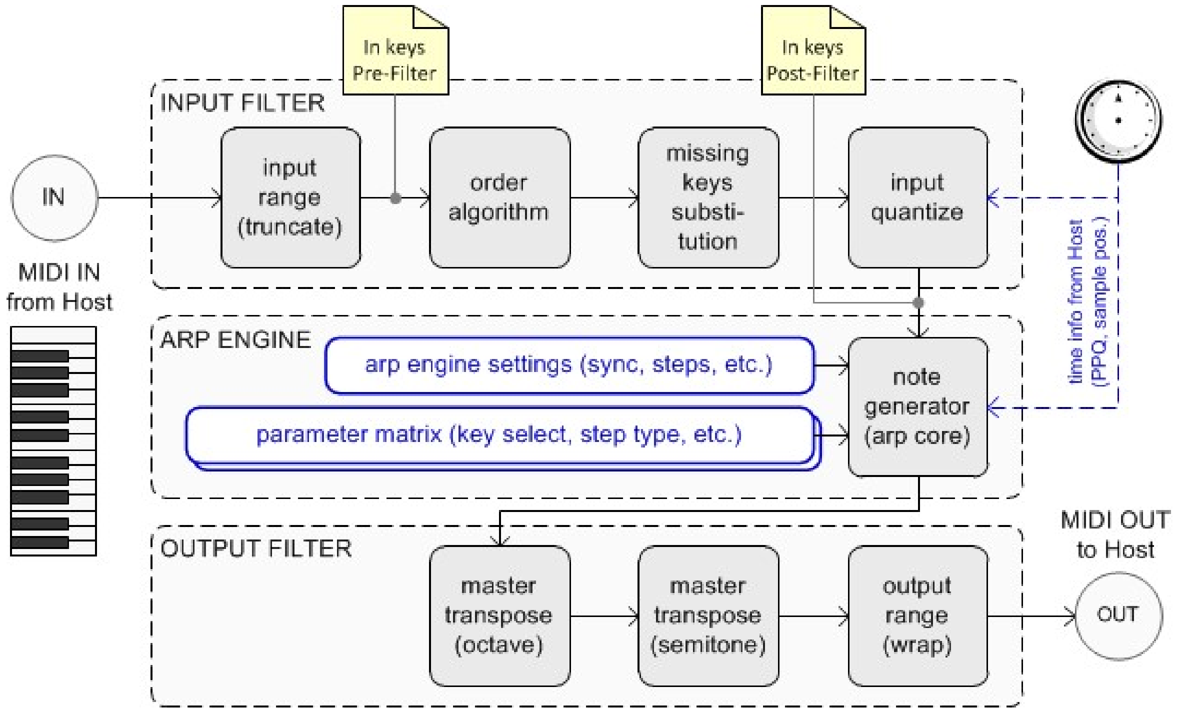
BlueARP will appear in the plugin stack above the instrument plugin:



Hint: Faster way to do the same: hold ALT while selecting and adding the plugin.

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.


In keys Pre-Filter
In keys Post-Filter
INPUT FILTER
IN
MIDI IN from Host
input range (truncate)
order algorithm
missing keys substi-tution
input quantize
time info from Host (PPQ, sample pos.)
ARP ENGINE
arp engine settings (sync, steps, etc.)
parameter matrix (key select, step type, etc.)
note generator (arp core)
OUTPUT FILTER
master transpose (octave)
master transpose (semitone)
output range (wrap)
MIDI OUT to Host
OUT

pic. 1. BlueARP processing diagram.

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

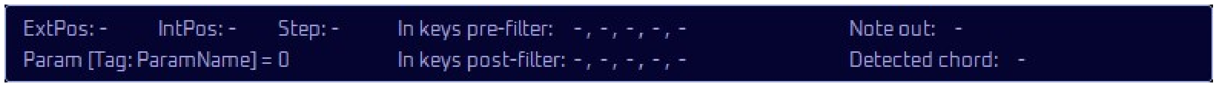
In this manual, «keys» are actually pressed notes on the keyboard, while generated «notes» come from arpeggiator output.

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 20).

«*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 37.

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 32 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 20 for details.

Output Filter adds some post-processing to generated notes – octave / semitone transposition, wrapping notes to fit the given range. See page 24 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 35 for more details.

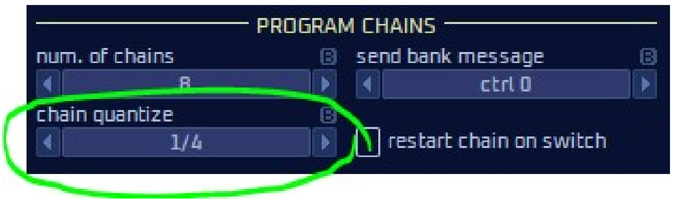
Basic Concepts

Switching patterns on the fly

You can either switch programs or assign programs to chains and switch chains. While you can switch programs on the fly during the performance, this switching will not be quantized and you may experience some early/missing notes at the output. In BlueARP, it is better to use chains instead.

Chain is a pre-programmed sequence of programs, which can be automated and changed on the fly/ Each chain can contain just 1 program or up to 8 programs, which will play one after another.

The main advantage of chains: chain switching is quantized according to this setting on the left panel:



PROGRAM CHAINS

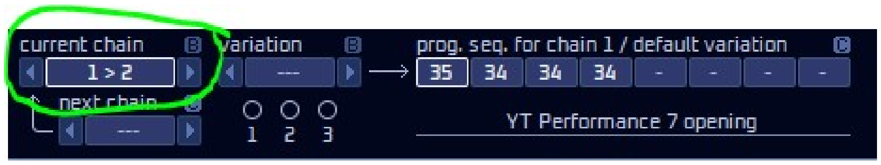
num. of chains: 8

send bank message: ctrl 0

chain quantize: 1/4

restart chain on switch

In this case, chain will actually switch not when you change «current chain» param, but at the start of the next beat (or 1/4 of a bar). While chain is about to switch but didn't switch yet, it is highlighted with a white frame and the numbers «1 > 2» say that it is switching from chain 1 to chain 2, but yet is on chain 1.



current chain: 1 > 2

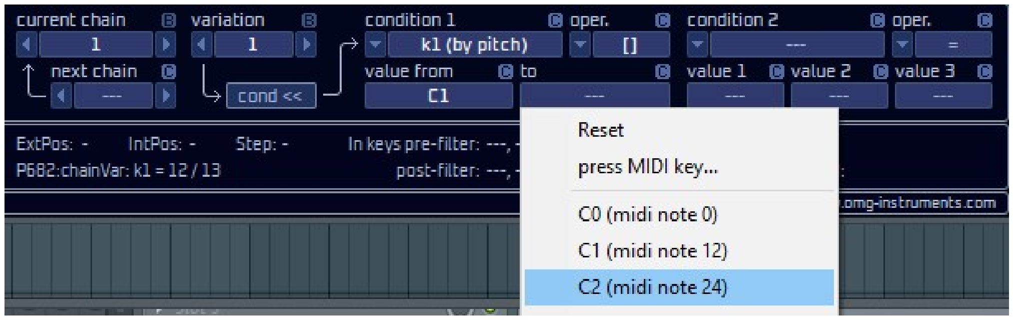
next chain: ---

variation: ---

prog. seq. for chain 1 / default variation: 35 34 34 34 - - - -

YT Performance 7 opening

Chain variations add even more flexibility to chains: you can have up to 3 variations per chain, each with its own program sequence, triggered by a certain chord, key, root note, etc. For example, this condition is input key k1 is in range C0 .. C1:



current chain: 1

next chain: ---

variation: 1

cond <<

condition 1: k1 (by pitch)

oper.: []

value from: C1

to: ---

condition 2: ---

oper.: =

value 1: ---

value 2: ---

value 3: ---

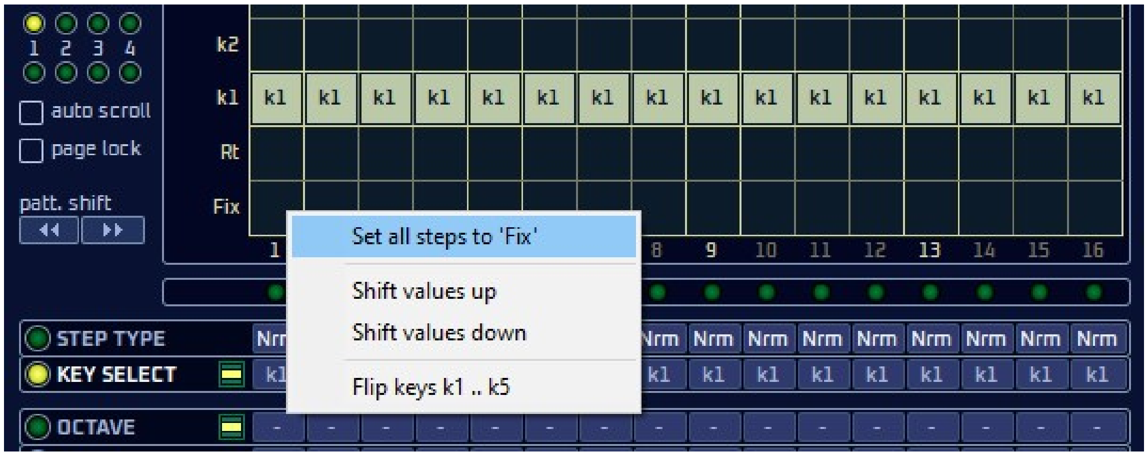
ExtPos: - IntPos: - Step: - In keys pre-filter: ---,

P682:chainVar: k1 = 12 / 13 post-filter: ---,

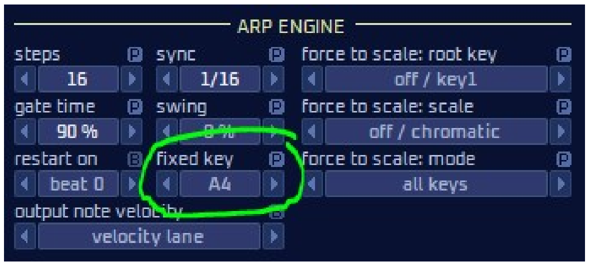
Once this condition is true, chain variation will trigger automatically (see page 35 for details on chains and chain variations).

Using BlueARP as a step sequencer

To do this, set all steps in your program to «Fixed»: select the KEY SELECT lane, then right-click on the matrix somewhere on the «Fix» value, choose «Set all steps to 'Fix'».

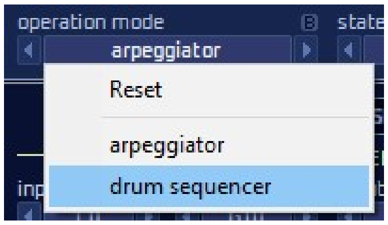


All KEY SELECT lane values will become «Fix» and now output notes will be bound to «fixed key» param in ARP ENGINE block:



Sequencing drums

To sequence drums with BlueARP, first switch it to «drum sequencer» operation mode:



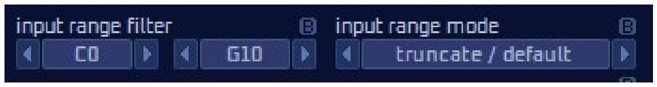
See «DRUM SEQUENCER mode» chapter on page 38 for more details.

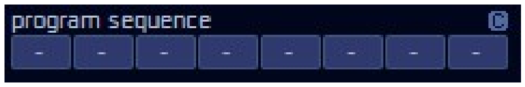
> **Note**: There's no limitation to use BlueARP for drums in «arpeggiator» mode and for melodies in «drum sequencer» mode, but still drum sequencing mode is better suited drums.

Interface

Elements and Navigation

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


input range filter B
C0 G10
input range mode B
truncate / default


program sequence C
- - - - - - - -

There are several ways to adjust the value:

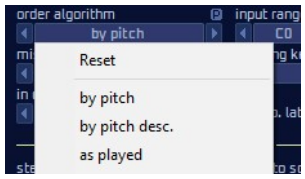
- left-click and hold on the box, drag it up or down;
- place the pointer over the box, use mouse wheel to adjust the value;
- click ◀ ▶ buttons to adjust the value;
- right-click on the box and select value from the popup menu

B / P / C 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)**.

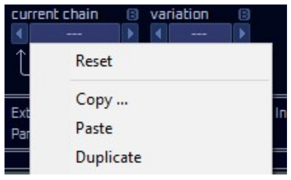
When you switch programs, (**B)** or bank-related parameters stay the same.

(C) or chain-related parameters are dependent on «current chain» setting (chains are described at page 18).

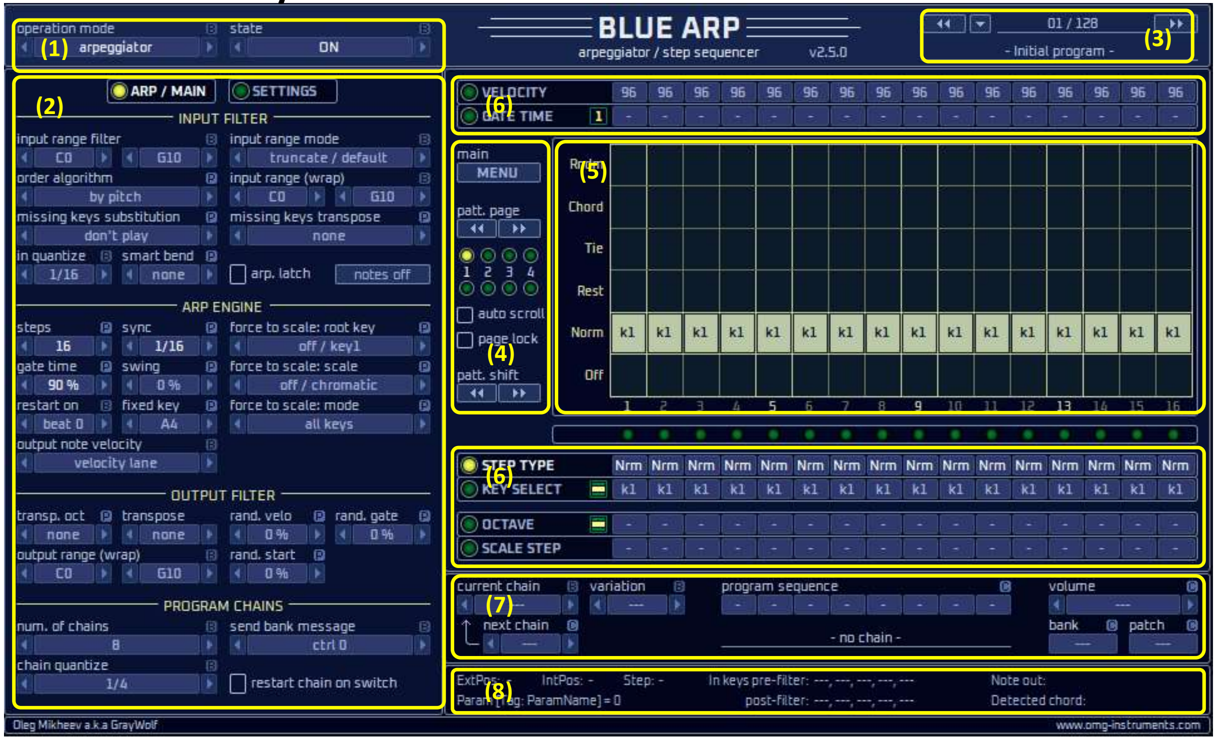
Almost all editable parameters have right-click popup menus, they mostly contain list of values to select from or actions to perform (copy, paste, etc.), for example



or



Main window layout



operation mode: arpeggiator
state: ON
01 / 128
- Initial program -

INPUT FILTER
input range filter: C0 G10
input range mode: truncate / default
order algorithm: by pitch
input range (wrap): C0 G10
missing keys substitution: don't play
missing keys transpose: none
in quantize: 1/16
smart bend: none
arp. latch
notes off

ARP ENGINE
steps: 16
sync: 1/16
force to scale: root key: off / key1
gate time: 90 %
swing: 0 %
force to scale: scale: off / chromatic
restart on: beat 0
fixed key: A4
force to scale: mode: all keys
output note velocity: velocity lane

OUTPUT FILTER
transp. oct: none
transpose: none
rand. velo: 0 %
rand. gate: 0 %
output range (wrap): C0 G10
rand. start: 0 %

PROGRAM CHAINS
num. of chains: 8
send bank message: ctrl 0
chain quantize: 1/4
restart chain on switch

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 k1 k1 k1 k1 k1 k1 k1 k1 k1 k1 k1 k1 k1 k1 k1

current chain; variation: ---; program sequence; volume: ---
next chain: ---; - no chain -; bank: ---; patch: ---

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

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;

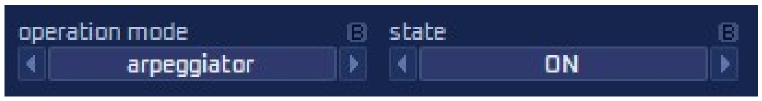
(5) **Matrix editor** represents step-related values for the selected value lane;

(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



operation mode: arpeggiator

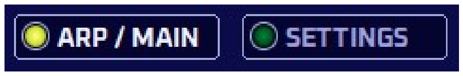
state: ON

operation mode	Switching between ARP and DRUM SEQUENCER mode
values	*arpeggiator, drum sequencer*
comments	*arpeggiator – default mode* *drum sequencer – some of the functions and GUI elements change to sequence drums, see page 38 for details*

state	arp state
values	*OFF, ON, THRU, 2k auto*, 3k auto**
comments	*OFF – switched off, ignores incoming MIDI;* *ON – ARP is active;* *THRU – pass thru for MIDI noted with respect to input range filter;* *2k auto on/off – ON when 2 or more keys pressed, OFF otherwise;* *2k auto on/thru – ON when 2 or more keys pressed, THRU otherwise;* *3k auto on/off – ON when 3 or more keys pressed, OFF otherwise;* *3k auto on/thru – ON when 3 or more keys pressed, THRU otherwise;*

Block (2): Left panel: ARP / MAIN

Left panel has 2 pages – ARP/MAIN and SETTINGS, click there buttons to switch:



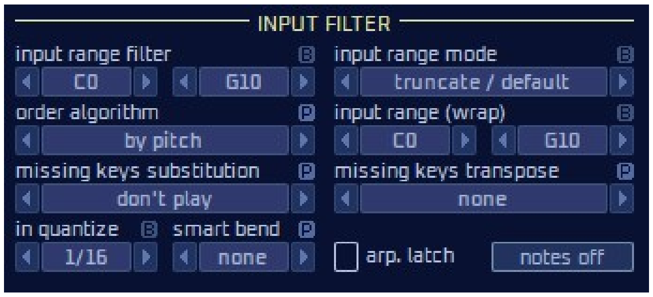
SETTINGS page contains rarely used bank-related settings, described on page 27 (you don't need to change them often, so they were moved them to a separate page to save main GUI space).

ARP/ MAIN is a primary page, it is divided into 4 blocks – «INPUT FILTER», «ARP ENGINE», «OUTPUT FILTER» and «PROGRAM CHAINS». Their controls are described in the following chapters.

In this manual, we use the word «keys» to represent what's pressed on the MIDI keyboard, while «notes» is what comes from the 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 and it won't change as you switch programs.

INPUT FILTER

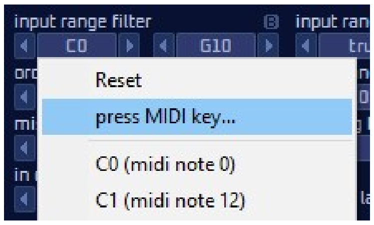


INPUT FILTER processes the input key list before it enters the arpeggiator «core» engine. We have «Input keys post-filter» key list at the output of this block.

These keys go further into «Arp Engine» block.

input range filter	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.



input range mode	modifies «input range filter» behavior
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.

input range (wrap)	range for input key «wrap-around»
values	*C0 .. G10 (MIDI notes 0 .. 127)*
comments	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 **A2**, C3, E3, G3, **D4**, the processed keys will be **A3**, C3, E3, G3, **D3** (bold notes were wrapped into the range C3...C4). 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.

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 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. «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, but inverted chord will stay inverted.

missing keys substitution	missing keys substitution algorithm
values	*don't play, cyclic, first key, last key, fixed key*
comments	When your pattern has more keys than you actually play, this setting will determine whether to mute these steps (*don't play*) or substitute missing keys with the existing ones. For example, you hold C5 and E5, while «KEY SELECT» lane has steps with «k1», «k2», «k3» and «k4». 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: • *don't play* *«C5, E5, -, -, -»* • *cyclic* *«C5, E5, C5, E5, C5»* • *first key* *«C5, E5, C5, C5, C5»* • *last key* *«C5, E5, E5, E5, E5»* • *fixed key* *«C5, E5, G5, G5, G5» («fixed key» = G5)*

missing keys transpose	additional transpose for substituted keys
values	*none, -1 octave, +1 octave*
comments	Adds additional transposition for substituted missing keys. For the example above, if we set missing keys transpose to +1 octave, post-filter key list will be: • *don't play* *«C5, E5, -, -, -»* • *cyclic* *«C5, E5, C6, E6, C6»* • *first key* *«C5, E5, C6, C6, C6»* • *last key* *«C5, E5, E6, E6, E6»* • *fixed key* *«C5, E5, G6, G6, G6» («fixed key» = G5)*

in quantize	input keys real-time quantization
values	*none, 1/16, 1/12, 1/8, 1/6, 1/4, 1/2, 1 bar, 2 bars*
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.

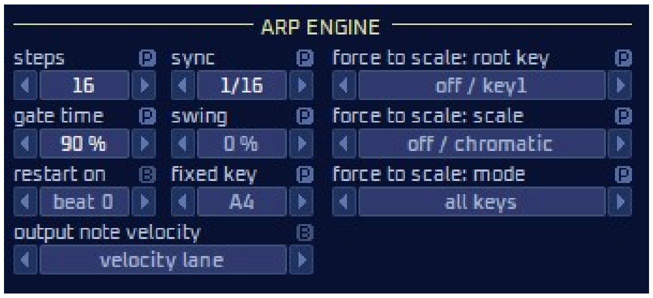
Hint. When input quantize is on, you should press keys a little beforehand, because input keys need to be already captured when the next step/beat starts.

smart bend	transpose up or down for solo leads (experimental)
values	*-9 .. none .. 9*
comments	This experimental feature will transpose the output by speps of the selected scale, with respect to quantization setting. It was designed to mimic guitar shredding for leads.

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 off to free your hands from the keyboard to do some other stuff.

notes off	All notes off (button)
Values	-
comments	Works like ‘PANIC’ button in DAWs, will clear the output note buffer and send ‘All notes OFF’ midi message.

ARP ENGINE


ARP ENGINE

steps: 16	sync: 1/16	force to scale: root key: off / key1
gate time: 90 %	swing: 0 %	force to scale: scale: off / chromatic
restart on: beat 0	fixed key: A4	force to scale: mode: all keys
output note velocity: velocity lane		

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.

steps	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. ***steps = 0 and 1*** 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.

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% .. 125%*
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 the step length. For negative values, it will start earlier.

restart on	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.

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

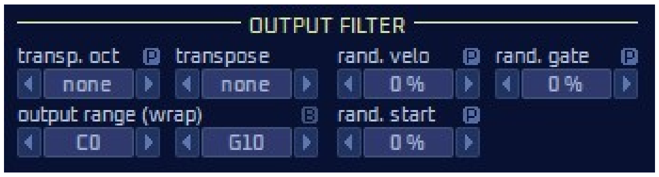
output note velocity	Sets where to take velocity for generated notes
values	*velocity lane, input key, lane + input key*
comments	«lane + input key»: BlueARP takes output note velocity from VELOCITY lane and adjusts it to input note velocity (multiplying and normalizing them)

force to scale: root key	root key for «force to scale» mode
values	*off/key1, detect from chord, C, C#, D ... Bb, B*
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 **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 will fit output notes to the given scale (either all or only semi-transposed notes, depending on «force to scale: mode» parameter); 2. «SCALE STEP» lane 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. 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).

force to scale: mode	how to apply semitone transposition
values	*all keys, semi-transposed*
comments	Works together with "force to scale: scale" parameter. When set to *semi-transposed*, 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.

transp. oct	output transposition, octaves
values	*-3 oct .. +3 oct*
comments	It is program-related.

transpose	output transposition, semitones
values	*-12 .. +12*
comments	It is bank-related, because there's no sense to make this setting different for different programs.

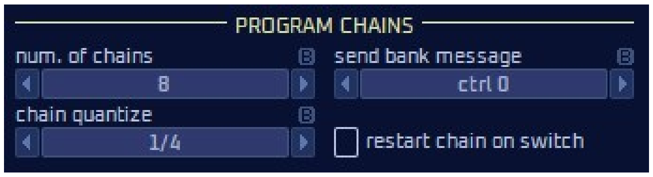
output range (wrap)	Range for output notes (wrapping)
values	*C0 .. G10 (MIDI notes 0 .. 127)*
comments	Notes outside the range will be wrapped (octave-transposed up or down to fit the range). Works just like «input range (wrap)», but for the output notes.

rand. velo	randomize output note velocity
values	*0% .. 100%*
comments	Add random value (positive or negative) to each output note velocity.

rand. gate	randomize output note gate time
values	*0% .. 100%*
comments	Add random value (both positive and negative) to generated note length.

rand. start	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.

num. of 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, setting «num. of chains» to the appropriate value will utilize full rotation range of this knob.

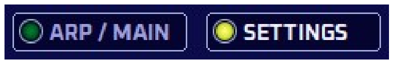
chain quantize	input quantization for chain switching
values	*none, 1/16, 1/12, 1/8, 1/6, 1/4, 1/2, 1 bar, 2 bars*
comments	When you switch chains, for better transition it should be done strictly at the start of a new beat. Chain quantize = 1/4 does exactly that and it is the default setting.

send bank message	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.

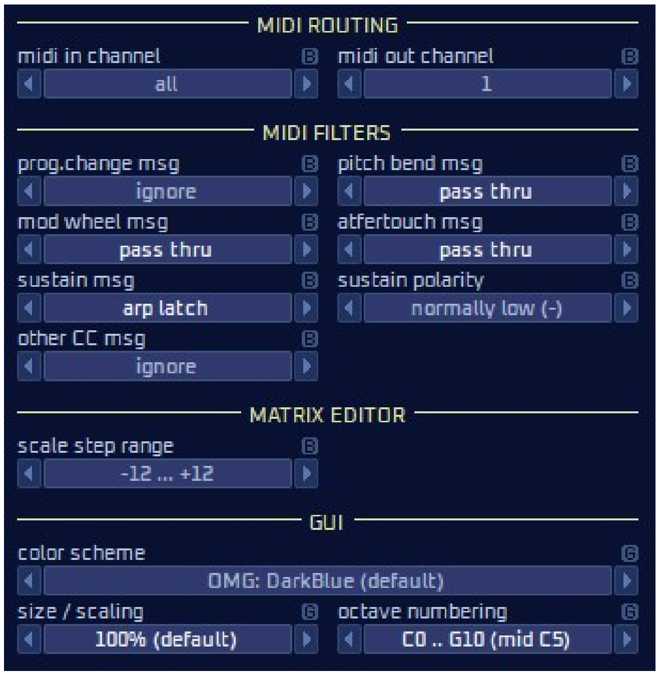
restart chain on switch	restart chain from the beginning after chain switch
values	*On, Off*
comments	When checked, chain always starts from the beginning after chain switch (otherwise, in restart on «beat 0» mode, chain step is calculated from song position given by host)

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 were moved to a separate page to save GUI space on the main panel. You don't need to change them often.



MIDI ROUTING

midi in channel: all
midi out channel: 1

MIDI FILTERS

prog.change msg: ignore
pitch bend msg: pass thru
mod wheel msg: pass thru
aftertouch msg: pass thru
sustain msg: arp latch
sustain polarity: normally low (-)
other CC msg: ignore

MATRIX EDITOR

scale step range: -12 ... +12

GUI

color scheme: OMG: DarkBlue (default)
size / scaling: 100% (default)
octave numbering: C0 .. G10 (mid C5)

MIDI ROUTING block has MIDI input and output channel, moved here to save GUI space.

MIDI FILTERS block contains settings for MIDI message filtering(*).

MATRIX EDITOR block adjusts matrix editor behavior.

GUI block has some GUI-related settings.

(*) Some of these settings are not supported in VST3 version, see «VST3 compatibility» in «FAQ / Troubleshooting» section on page 40.

midi in channel	input MIDI channel
values	*all, 1 .. 16*
comments	*all* – BlueARP will take MIDI input from all MIDI channels, *1 .. 16* – only from a given channel.

midi out channel	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 MIDI channels.

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

pitch bend msg	how to respond to incoming Pitch Bend MIDI message
values	*ignore, pass thru*
comments	«pass thru»: BlueARP will pass this message to its MIDI out (= to VST plugin it is connected to).

mod wheel msg	how to respond to Modulation Wheel MIDI message
values	*ignore, pass thru*
comments	«pass thru» - BlueARP will pass this message to its MIDI out.

aftertouch msg	how to respond to incoming aftertouch MIDI message
values	*ignore, pass thru*
comments	«pass thru»: BlueARP will pass this message to its MIDI out.

sustain msg	how to respond to sustain MIDI message (CC 64)
values	*ignore, pass thru, sustain, arp latch*
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.

sustain polarity	sustain pedal polarity for «sustain msg» setting
values	*normally low (-), normally high (+)*
comments	Normally low (-) means that in released state it should be value 0.

other CC msg	sets how to respond to incoming CC MIDI messages
values	*ignore, pass thru*
comments	The same as other MIDI filters, but applies to all other CC messages not mentioned before. *This setting is not working in VST3 version due to VST3 limitations*

velocity scale	sets velocity accuracy for VELOCITY lane
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 value span for SCALE STEP lane
values	*«-12...+12», «0...+12», «-7...+7», «0...+7»*
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.

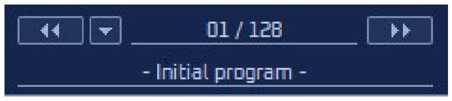
color scheme	sets skin / color theme
values	*default (blue) and others*
comments	Color schemes are stored in *.ini files in \skins sub-directory. On windows it is in plugin directory, on Mac – inside the bundle). Selected color scheme index is stored in BlueARP.ini file in user directory: **Windows**: C:\Users\<user>\AppData\Roaming\BlueARP **OSX**: c:/Users/<username>/Library/Application Support/BlueARP

Hint. 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.

size / scaling	sets GUI size
values	*100%, 125%, 150%, 200%*
comments	Adjusts GUI size.

octave numbering	sets one of note naming conventions
values	*«C-2 .. G8 (mid C3)», «C-1 .. G9 (mid C4)», «C0 .. G10 (mid C5)»*
comments	It tells BlueARP how to display notes or which key is the middle - C3, C4 or C5. It doesn't affects the functioning, just the way note names are displayed.

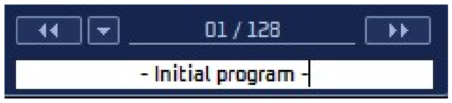
Block (3): PROGRAM BROWSER



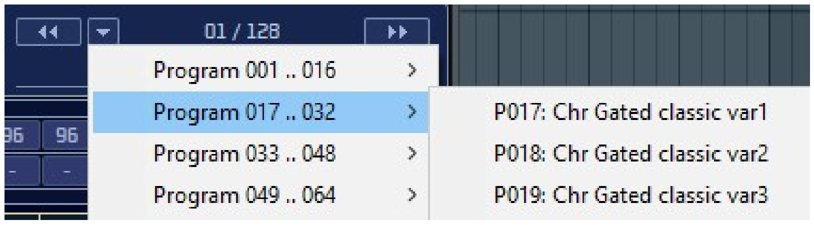
Use buttons to navigate through the programs in a current bank. Alternatively, click on the program number and drag up or down to change it (just like with the other value boxes).

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

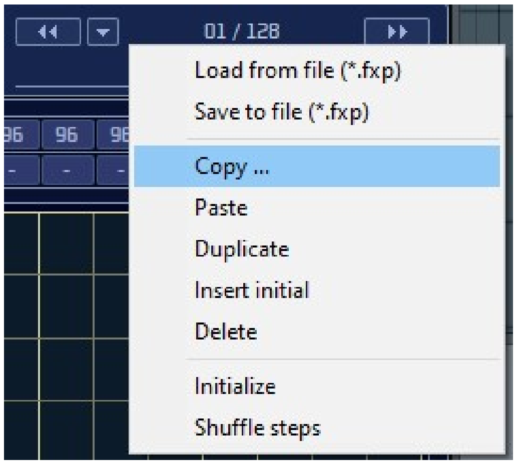
To change program name, click on it, type in new name and hit enter or click somewhere outside this area.



Click drop-down button to select a program from the list:


01 / 128
Program 001 .. 016
Program 017 .. 032
Program 033 .. 048
Program 049 .. 064
P017: Chr Gated classic var1
P018: Chr Gated classic var2
P019: Chr Gated classic var3

Right click on program number label to access actions menu:

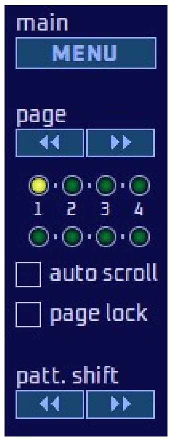


Here you can quickly copy, paste and duplicate programs, i.e. when you are experimenting and don't want to lose the original pattern.

It duplicates the «Program» submenu from the main menu, accessible via MENU button.

Note: when you duplicate, delete of insert the program, all the rest of the programs are shifted left of right and their numbers are changed, but BlueARP takes care of that and adjusts program numbers in chains, so your chains won't be spoiled.

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 you pattern is longer than 16 steps, so it doesn't fit single screen. There are 2 small LED lanes underneath, upper one shows the 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 to prevent pages from switching while you edit).

Pattern shift buttons perform cyclic step shifting (pattern rotation). It's useful when your pattern doesn't match the beat and you try 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 (this is why it is also called pattern rotation).

Main menu includes the following items:

Bank	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

Program	load, save and copy/paste programs
Load from file (*.fxp)	Load program from file, current program will be overwriten
Save to file (*.fxp)	Save current program to file
Copy …	Memorize the current program for «Paste» operation.
Paste	Paste program at a current location («Copy …» should be done before). Paste overwrites the target program.
Duplicate	Duplicate program at the current location: inserts empty slot after the current program and copies it there.
Insert initial	Insert initial program at the current location. Current program and all the rest will be shifted right to make space for the new program, the last (128th) 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

Chain	copy/paste chains
Copy …	Memorize current chain as a source for copy/paste operation.
Paste	Paste chain at the current location («Copy …» should be done before). Paste overwrites target chain.
Duplicate	Duplicate the current chain: insert empty chain after this one and then copy current chain to the next one.
Insert initial	Insert initial chain at the current location, shifting the rest to the right.
Delete	Delete current chain.
Initialize	Clear current chain data
Initialize all chains	Clear all chains data

Chain variation	copy/paste chain variations
Copy …	Memorize current chain var. as a source for copy/paste operation.
Paste	Paste chain var. at the current location («Copy …» should be done before). Paste overwrites target chain variation.
Duplicate	Duplicate the current chain variation: insert empty chain variation after this one and then copy current chain var. onto the next one.
Insert initial	Insert initial chain var. at the current slot, shifting the rest to the right.
Delete	Delete current chain variation.
Initialize	Clear current chain variation data
Initialize all chains	Clear all chain variations for this chain

Page	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.

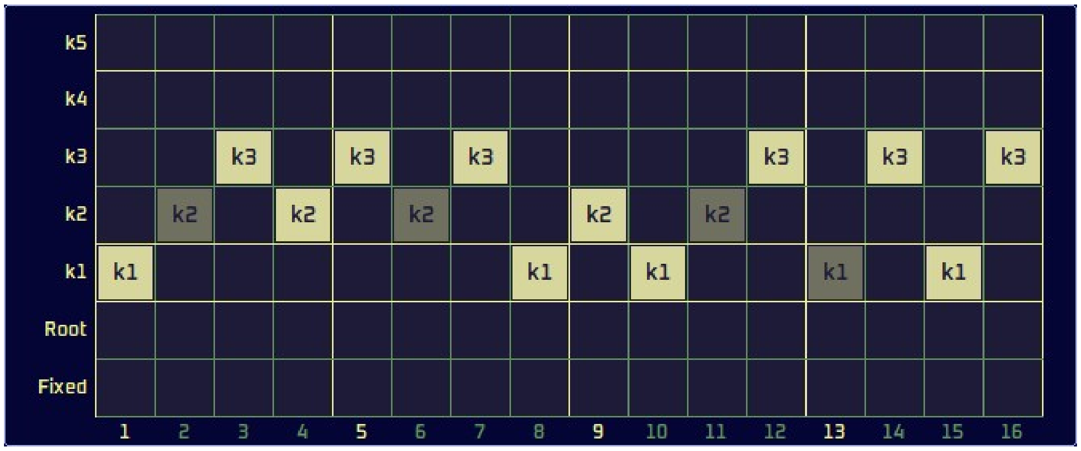
Debug info	various information
Open BlueARP.ini location	Opens BlueARP.ini file location. BlueARP.ini holds global settings like GUI scale, GUI skin index, octave numbering.
pGraphics->GetGUIAPI()	Show selected GUI API
pGraphics->PluginPath()	Show path to the plugin file
pGraphics->HostPath()	Show path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

Open Manual (pdf)	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

Make a donation	link to support BlueARP development via PayPal donation

Developer’s website	link to developer’s website with the latest updates for BlueARP

Block (5): MATRIX EDITOR



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

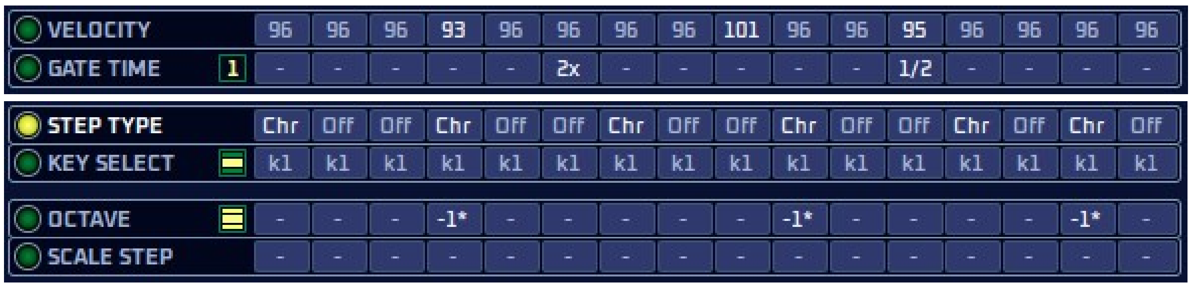
You can adjust step-related values either in a matrix editor or on a value lane itself.

On the matrix editor, there are several ways to change values:

- Click the cell to set the single value;
- Drag the mouse from left to right set the steps to a certain value, as you drag;
- Right click on a matrix sell and select «Set all steps to ***», where *** is the desired value, it will set all steps in a program, including those on other pages;

Grayed-out bricks mean that this particular setting doesn't affect the 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



		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VELOCITY		96	96	96	93	96	96	96	96	101	96	96	95	96	96	96	96
GATE TIME	1	-	-	-	-	-	2x	-	-	-	-	-	1/2	-	-	-	-
STEP TYPE		Chr	Off	Off	Chr	Off	Off	Chr	Off	Off	Chr	Off	Off	Chr	Off	Chr	Off
KEY SELECT		k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1	k1
OCTAVE		-	-	-	-1*	-	-	-	-	-	-1*	-	-	-	-	-1*	-
SCALE STEP		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

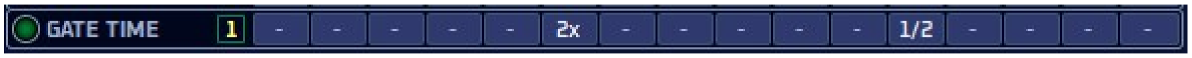
Value lanes contain step-related pattern parameters. Selected value lane is also shown in the MATRIX EDITOR. To select it, click the lane caption.

To adjust the value for a certain step:

- Click on it and drag up or down;
- Put the mouse over the box and use mouse wheel to adjust the value;
- Right-click on the box and select the value from the popup menu;

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 in matrix editor.

«GATE TIME» can be switched to «CHANNEL» or channel per step mode via 1 indicator:



GATE TIME	1	-	-	-	-	-	2x	-	-	-	-	-	1/2	-	-	-	-

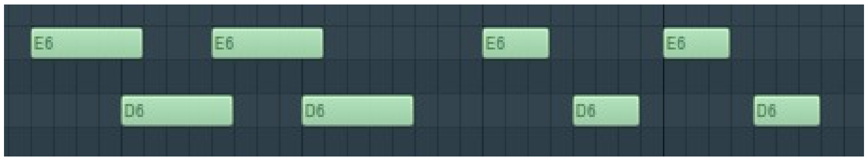
See descriptions for each value lane below.

VELOCITY	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.

GATE TIME (mode 1)	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.

CHANNEL (mode 2)	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 comples textures/arpeggios with different sounds for various 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* *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

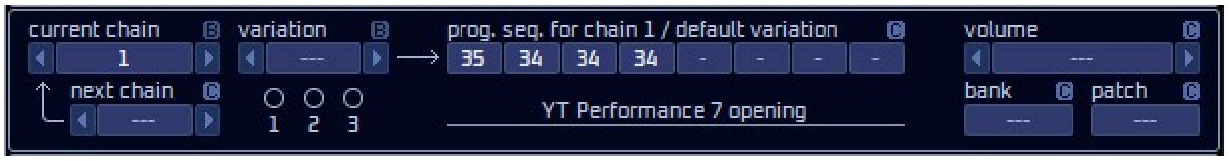
KEY SELECT	Input key selection for the given step
values	*Fixed – use fixed key from Arp Engine settings* *Root – root key from detected chord, key1 if no chord detected* *k1..k5 – take keys №1..5 from key list (post-filter)*
comments	Tells which key to take from «post-filter key list» for the current step. Yellow label next to KEY SELECT caption (or) toggles between monophonic and polyphonic mode. In **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» and use BlueARP as a step sequencer, or set some steps to «fixed» to create 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», this will work as a semitone transposition. Otherwise, it 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

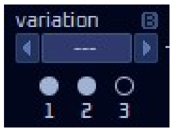


Program chains deliver the possibility to chain several programs (patterns) together into a longer «super-pattern». It was implemented with live performances in mind. White chains should be switched manually, chain variations are kind of «sub-chains» that are triggered automatically, according to pre-programmed conditions like the chords you play, keys you press, etc.

All controls to the right of the «variation» box are driven by chain + variation: when either chain or variation switches, it will bright up another program sequence.

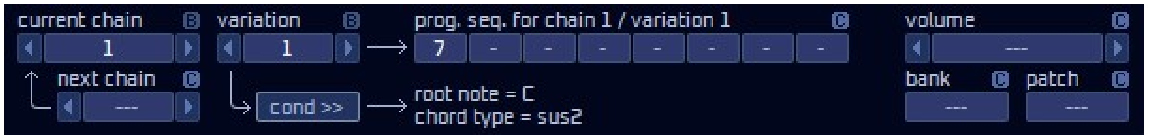
Chain variation «---» is the default variation or the chain itself. You may not use variations and use just chains; in this case you don't need to change chain variation box.

Dots and numbers below chain variation box show if any of the variations is used (non-empty):

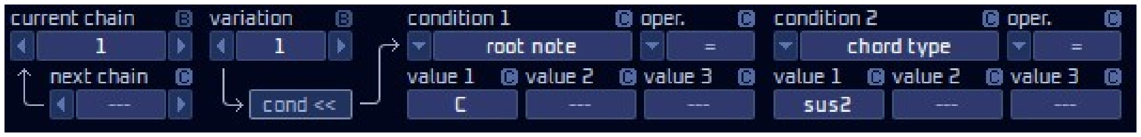


Click the dot to jump to that variation (the same as adjusting «variation» box).

When you jump to the variation, you can program a different program sequence for that variation"

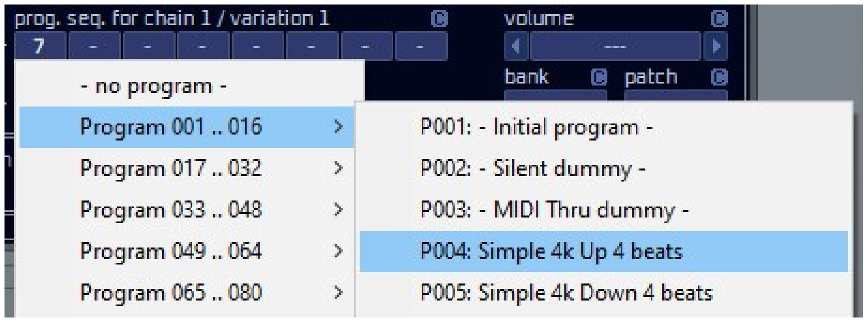


Press «cond >>» button to change trigger condition for this variation. In this example, variation 1 of chain 1 will be triggered when «Csus2» chord is detected:



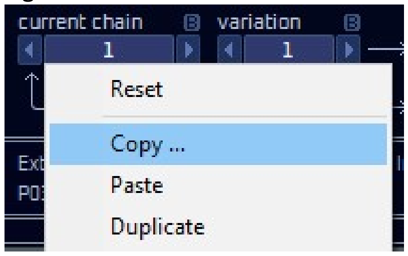
Program sequence lane holds numbers of chained programs for a current chain + variation.

Right-click program sequence slot to select program for a particular chain/variation step:



current chain	current chain
values	*---, 1, 2, … number of chains (up to 16)*
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 (1st step of the program sequence).
Right-click the box to access actions menu to copy/paste/duplicate chains, etc.:



variation	current chain variation
values	*---, 1, 2, 3*
comments	Adjust it to select chain variation for editing. During live performance, it is not meant to be automated / changed manually; rather you should define conditions on which it will trigger (like the chord or a certain key). When the conditions overlap for 2 or more variations within a chain, the first possible chain variation will trigger. Once no conditions are met, the default variation (or the chain itself) will trigger.

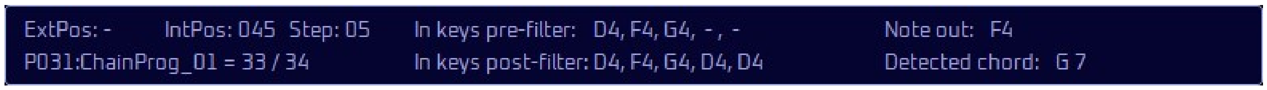
Hint. Right-click the box to access actions menu to copy/paste/duplicate chains, etc.:
(popup menu is the same way for chains and chain variations)

next chain	next chain auto-switch
values	*---, caller, caller-1, caller+1, chain 1, … chain 16*
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;

patch num, bank num	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).

volume	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): INFO 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;
- **Detected chord**: shows root key + chord type

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» – corresponding midi CC value. May be useful for automation with external controllers: for example, for checkboxes internal values will normally be 1 for On and 0 for Off, while the corresponding MIDI CC values will be 64 for On and 0 for Off. The be more precise, MIDI CC value range 0..63 will be interpreted as Off and 64..127 – as On.

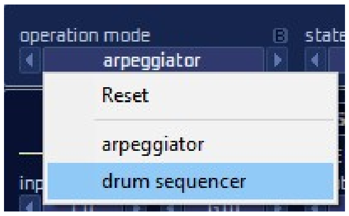
DRUM SEQUENCER mode

Overview

Since version 2.5.0, BlueARP has «drum sequencer» operation mode alongside the default «arpeggiator» mode. It was designed mostly for the BlueARP DM (hardware BlueARP counterpart), but may be useful with the plugin as well, cause drum sequences:

- Can be chained and automated the same way, using program chains and variations;
- Have some probability and randomization functions that can't be easily done in a DAW piano roll;

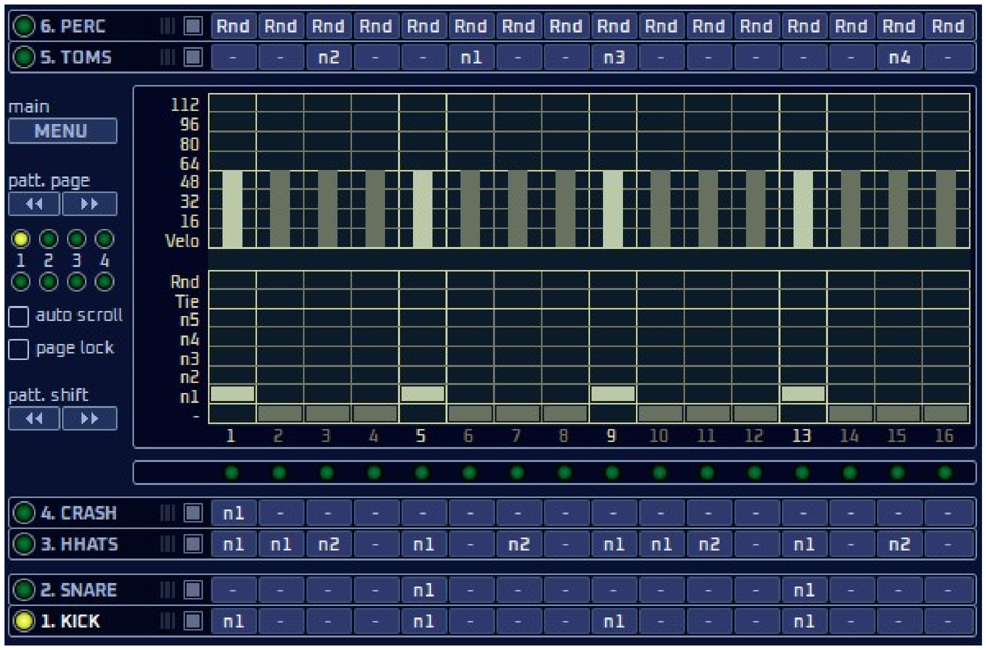
To switch to the drum mode, change «operation mode» one the left top:



Important note: Current bank with all its programs and chains will be initialized for the drum mode; there will be no warning dialog, so make sure to save your bank if needed.

Controls

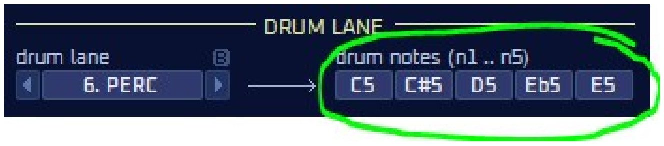
In «drum sequencer mode», some of the GUI elements will change to match this new functionality. First, value lanes will change their names and function:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
6. PERC	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd	Rnd
5. TOMS	-	-	n2	-	-	n1	-	-	n3	-	-	-	-	-	n4	-
4. CRASH	n1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3. HHATS	n1	n1	n2	-	n1	-	n2	-	n1	n1	n2	-	n1	-	n2	-
2. SNARE	-	-	-	-	n1	-	-	-	-	-	-	-	n1	-	-	-
1. KICK	n1	-	-	-	n1	-	-	-	n1	-	-	-	n1	-	-	-

Each of these lanes now can play a selected drum note per step, with varying velocity. Lane names like «KICK», «SHARE», «HHATS» are fixed and can't be changed, but it doesn't force you to use that particular sounds for those lanes.

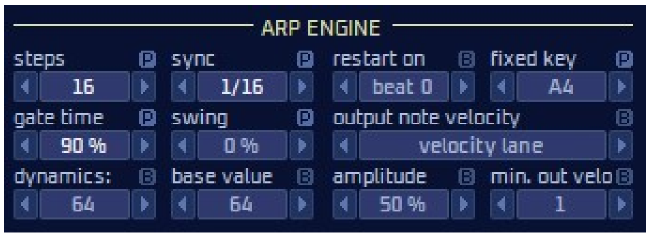
Values «n1..n5» are pre-programmed MIDI notes for each lane; you can change them on the left panel:



Value «-» will mute the step, «Tie» will extend the note from the previous step, «Rnd» will pick the random note from the «n1..n5» list.

On the left panel, INPUT FILTER block will shrink, some setting not relevant for drums will disappear. The remaining settings will work the same way as in the «arpeggiator» mode.

APR ENGINE block:

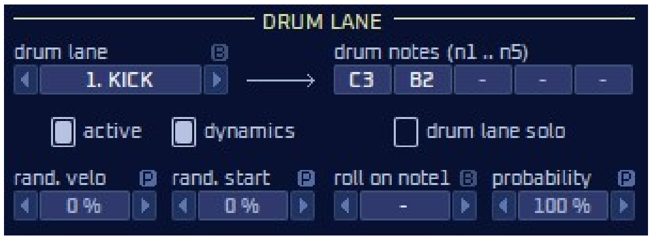


ARP ENGINE
steps 16 | sync 1/16 | restart on beat 0 | fixed key A4
gate time 90 % | swing 0 % | output note velocity velocity lane
dynamics: 64 | base value 64 | amplitude 50 % | min. out velo 1

«force to scale*» params will disappear, they are not relevant for drum sequencing.

A set of params for dynamic velocity control will appear - «**dynamics**» param and boxes to the right (see description below).

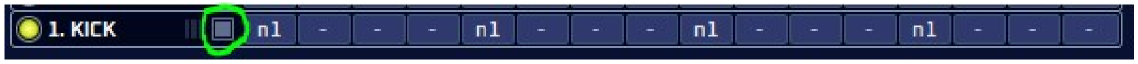
New **DRUM LANE** block will appear:



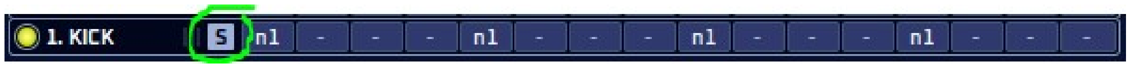
DRUM LANE
drum lane 1. KICK ⟶ drum notes (n1 .. n5) C3 B2 - - -
active | dynamics | drum lane solo
rand. velo 0 % | rand. start 0 % | roll on note1 - | probability 100 %

The «**drum lane**» parameter is linked to one of the 6 drum lanes, once you change it here, it will change the selected value bar and vice versa.

Uncheck «**active**» checkbox to mute the drum lane, you can also do it by clicking this box on the lane itself:



Check «drum lane solo» to solo the selected drum lane. On the lane itself, «S»-mark will appear, indicating that this lane is soloed.



When «**dynamics**» checkbox is checked, output note values for this lane will react to «dynamics» param value and its associated controls:

- «base value» sets the neutral value for «dynamics» param, greater dynamics values will amplify the velocity, lower values will attenuate it;
- «amplitude» param defines how much will output note velocity be amplified or attenuated by the «dynamics» param;
- «min. out velo» sets minimum value for the output velocity, it won't go any lower despite of all the other values;

If you switch to another lane, it will become the soloed one, unless you uncheck «drum lane solo».

Other options in DRUM LANE block include:

- «rand. velo» - add random value to output note velocity;
- «rand. start» - add random positive offset to output note start time;
- «roll on note 1» - trigger drum roll for drum note n1;
- «probability» - sets the trigger chance for any note on this lane;

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

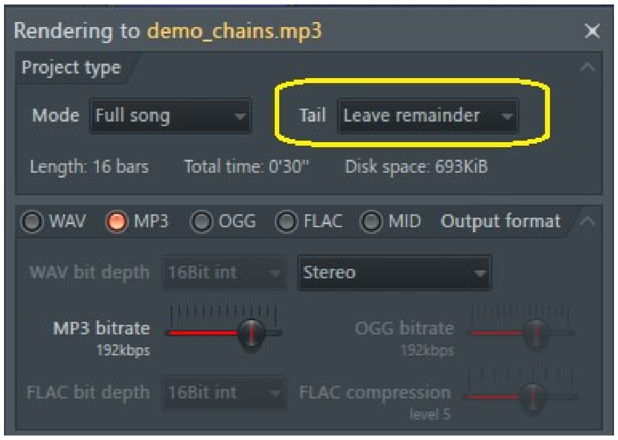
Output note timing is not perfect, like they are delayed by random values.

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 (256 / 48000 ≈ 0.005s)

Rendering audio in FL Studio

When trying to render a project in FL Studio, only 1st 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»


Rendering to demo_chains.mp3

Project type

Mode: Full song

Tail: Leave remainder

Length: 16 bars   Total time: 0'30"   Disk space: 693KiB

WAV [✓ MP3] OGG FLAC MID   Output format

WAV bit depth: 16Bit int   Stereo

MP3 bitrate: 192kbps   OGG bitrate: 192kbps

FLAC bit depth: 16Bit int   FLAC compression: level 5

solution provided by Saif Sameer

VST3 compatibility

«MIDI FILTERS» settings like pass thru for pitch bend and mod wheel do not work in VST3 version of BlueARP, while in VST2 version they work fine.

VST3 standard has some artificial limitations, preventing developers to create full-featured MIDI plugins. There are some «Legacy» extensions in VST3 added at some point, but still normally MIDI CC reception is not supported by VST3. If your DAW has full VST3 support (in particular, if it calls getMidiControllerAssignment() to ask plugin about supported controller messages), then almost all settings in «MIDI FILTERS» block should work. Except one - «other CC msg», it will be grayed out in VST3 version, because it is too tricky to implement (will require adding 100+ dummy params to the plugin to handle MIDI CC messages).

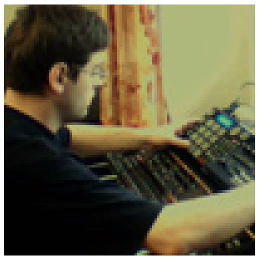
If this functionality is critical for you, use VST2.4 version of BlueARP instead.

Links

Developer's website:
http://www.omg-instruments.com/
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



Oleg Mikheev aka Graywolf, © 2012-2024