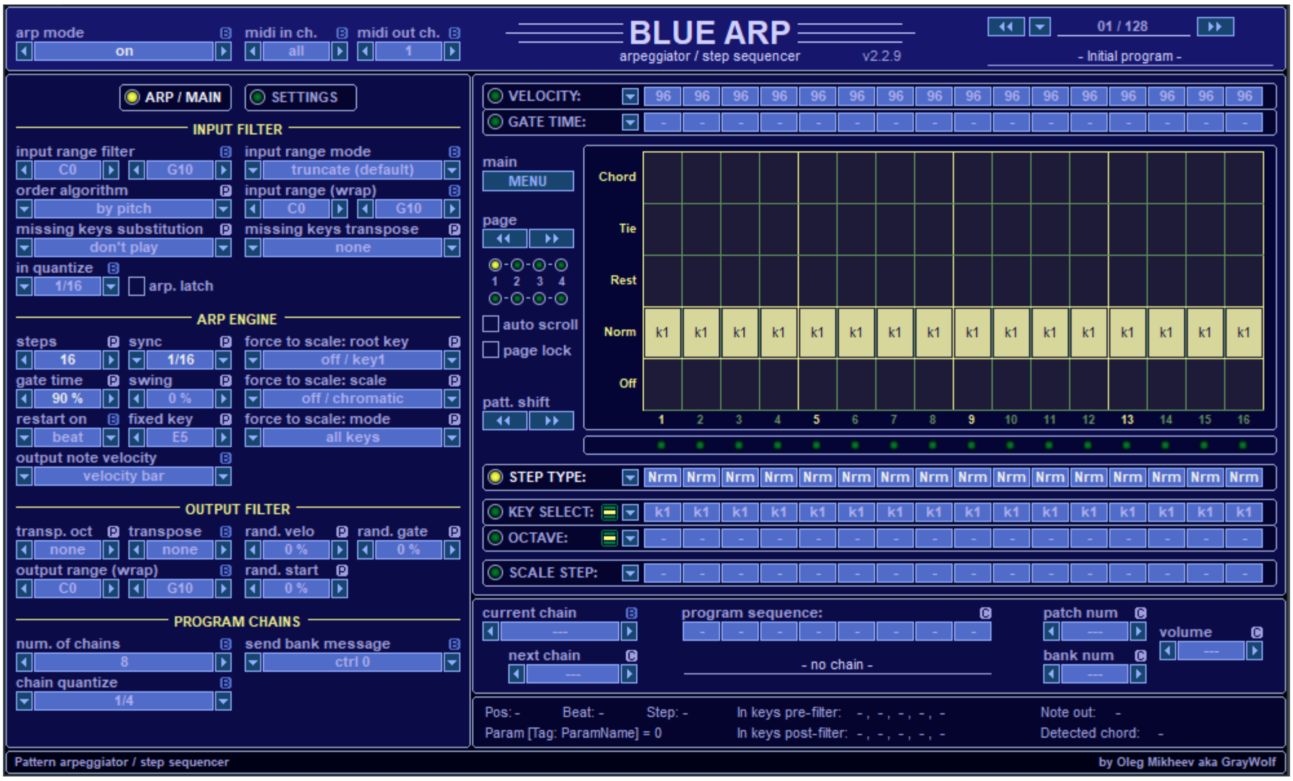
BLUE ARP

Operation Manual

Corresponds to BlueARP v2.2.9



arp mode: on
midi in ch.: all
midi out ch.: 1
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
☐ arp. latch

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
fixed key: E5
force to scale: mode: all keys
output note velocity: velocity bar

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

☐ auto scroll
☐ page lock

	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:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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: ---
next chain: ---
program sequence: - - - - - - - -
- no chain -
patch num: ---
bank num: ---
volume: ---

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

Pattern arpeggiator / step sequencer
by Oleg Mikheev aka GrayWolf

Pattern Arpeggiator / Step Sequencer

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

by Oleg Mikheev aka Graywolf

http://www.graywolf2004.net

May 2019

Table of Contents

Introduction

BlueARP is a programmable pattern arpeggiator / step sequencer, it comes as a VST or MIDI-FX plug-in for MAC OSX and Windows, both 32 and 64 bit. BlueARP is a pure MIDI plugin, it doesn't generate any sound by itself but transforms MIDI messages, so it has to be routed to either software or hardware synth in any VST/AU-enabled DAW like FL Studio, Ableton Live, Cubase, Reaper, Logic Pro, etc.

Basically you need to program some pattern in BlueARP, then play some chords and BlueARP will turn these chords into melodic phrases according to the pattern you programmed / 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 (Audio Unit for Logic Pro X)
OS: OS X (tested on 10.6.8), Windows XP and higher

Features

- Up to 64 steps per pattern;
- Up to 128 programs per bank;
- «Chains» feature to merge patterns together into longer «super-patterns», ability to switch chains on the fly;
- Comes with 128 factory patterns to start with;
- Has 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;
- Input range setting for keyboard-split performances;
- Separate settings for octave and semitone per step transpose;
- Configurable color schemes;

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

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

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

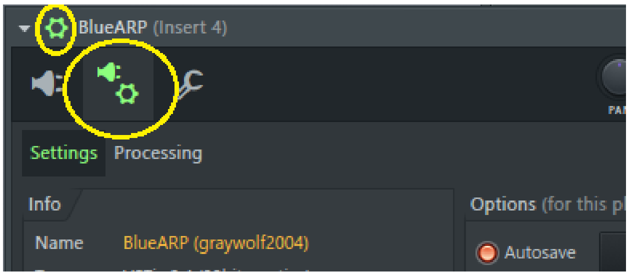
These are live performances using BlueARP with FL Studio, but the same can be done with Ableton Live and many other DAWs.

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 VST arpeggiator.

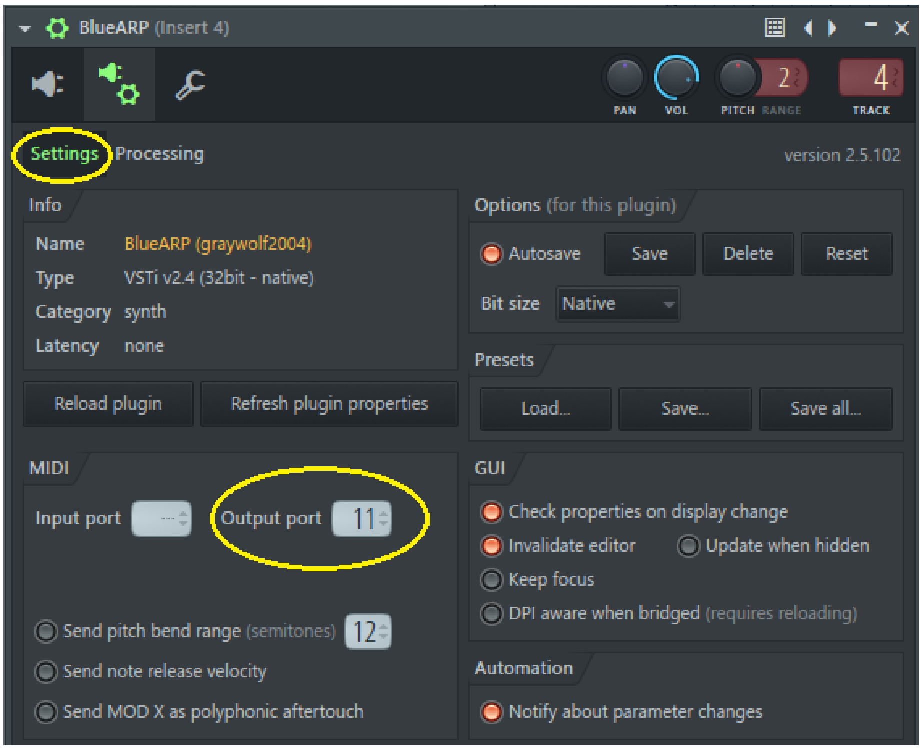
FL Studio (Fruity Wrapper method)

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):



version 2.5.102

Pitch range 2, Track 4

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

● Autosave
Bit size Native

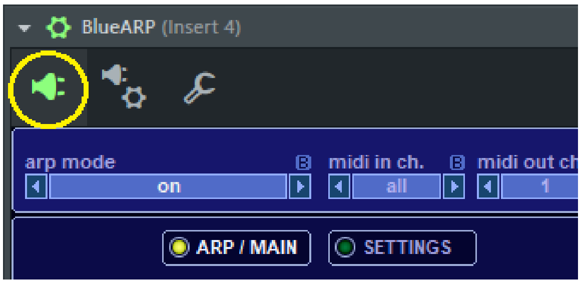
Input port ---
Output port 11

○ 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)

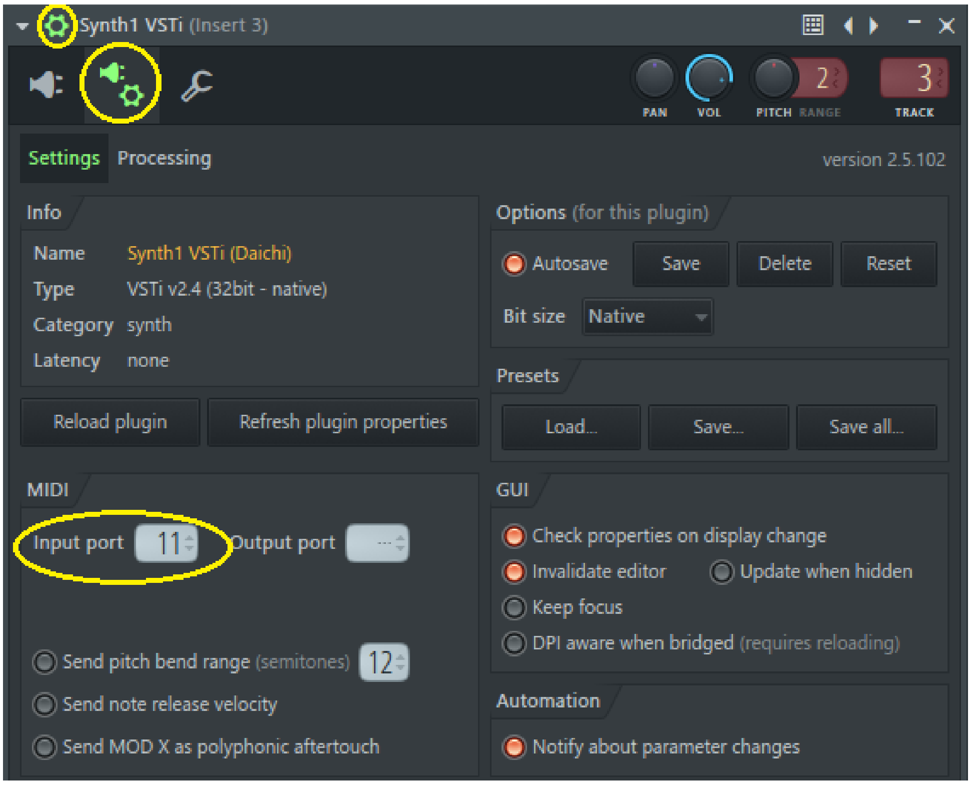
● Notify about parameter changes

Return to main plug-in 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 the same MIDI port as MIDI Input port:



PITCH RANGE: 2
TRACK: 3
version 2.5.102

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

● Autosave
Bit size: Native

Input port: 11
Output port: ---

○ 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)

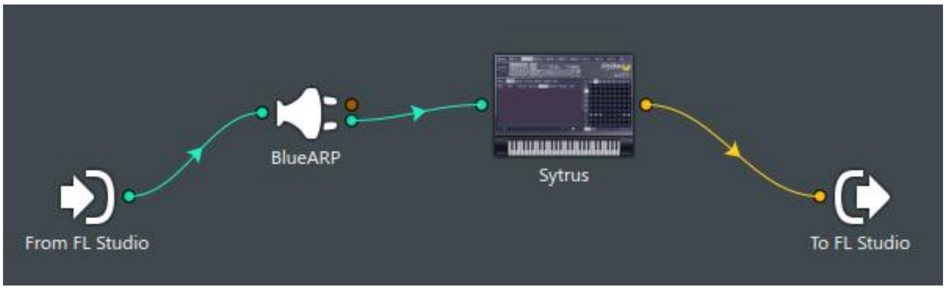
● Notify about parameter changes

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 or other routings.

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

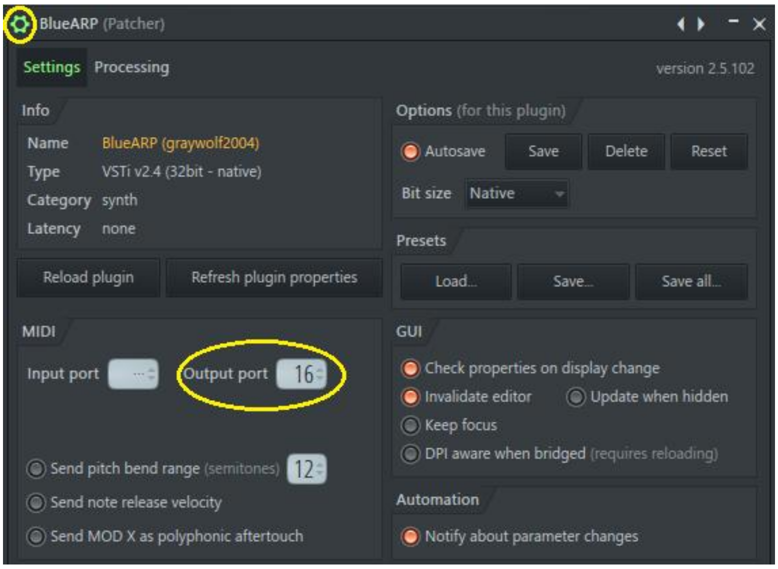
FL Studio (Patcher method)

Add «Patcher» instrument to a track, inside Patcher add BlueARP and Fruity Generator of your choice, connect them as follows:



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

Double cluck BlueARP, go to wrapping settings and set output port to any unused number.



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

● Autosave
Bit size Native

Input port ---
Output port 16

○ 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

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».

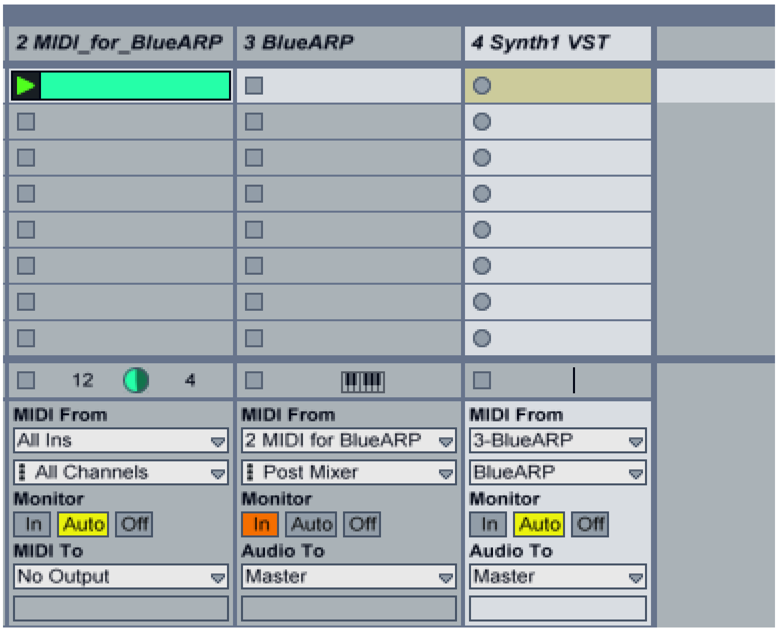
There's an issue - BlueARP will pick up MIDI from clips only when Monitor = «Auto», but it takes notes from Keyboard only with Monitor = «In». So you have to constantly switch monitor from «In» to «Auto».

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	All Ins	2 MIDI for BlueARP	3-BlueARP
	All Channels	Post Mixer	BlueARP
Monitor	Auto	In	Auto
MIDI To / Audio To	MIDI To: No Output	Audio To: Master	Audio To: 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».

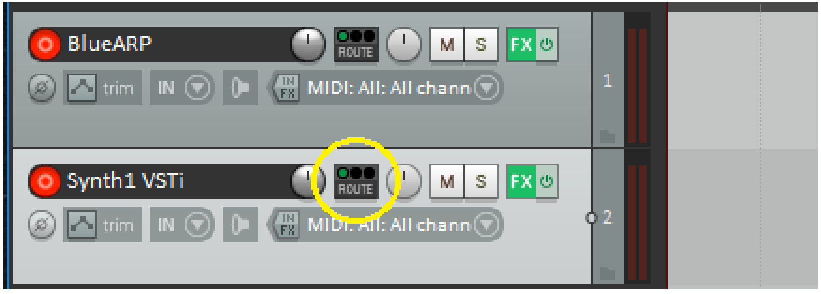
Now, use «*MIDI_for_BlueARP*» track to record patterns and «*BlueARP*» track to play live.

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

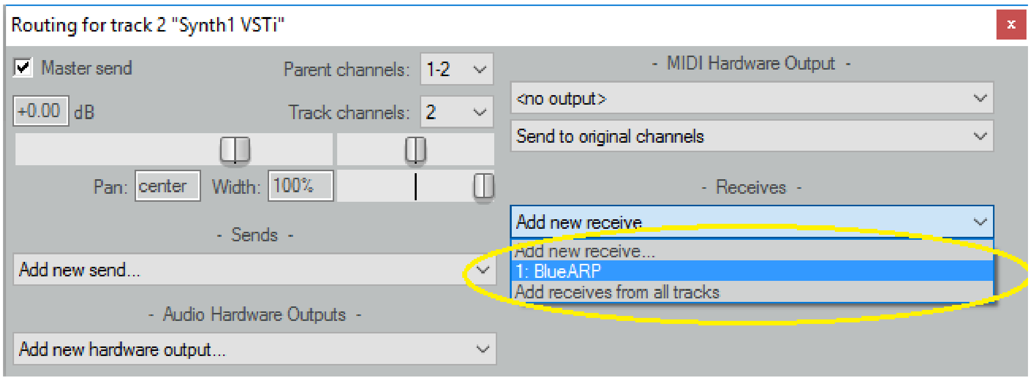
Reaper

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

Press ROUTE button on Synth1 track:



Add new receive from BlueARP:


Routing for track 2 "Synth1 VSTi"

☑ Master send

Parent channels: 1-2

+0.00 dB

Track channels: 2

Pan: center

Width: 100%

- Sends -

Add new send...

- Audio Hardware Outputs -

Add new hardware output...

- MIDI Hardware Output -

<no output>

Send to original channels

- Receives -

Add new receive

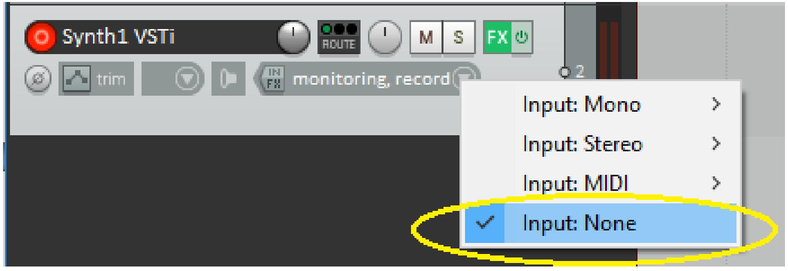
Add new receive...

1: BlueARP

Add receives from all tracks

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

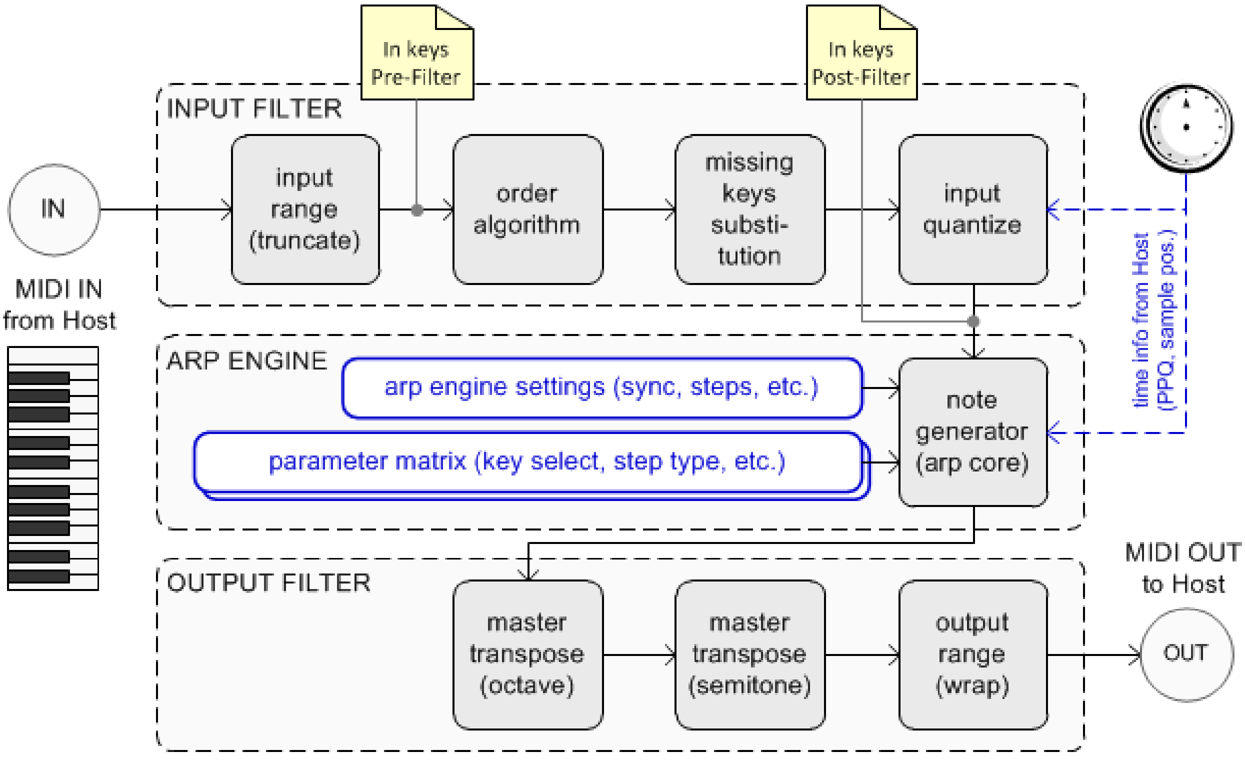
Set Input to None for Synth1 track:



Signal flow

To use BlueARP to the maximum, it's valuable to have a concept about its structure and signal processing logic.

The picture below 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».

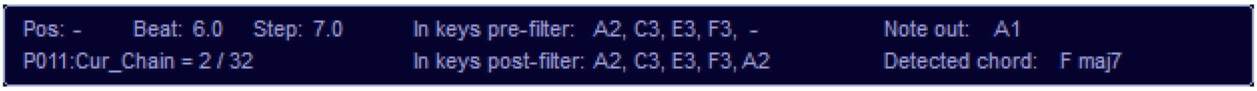
> *In this manual, «keys» are actually pressed notes on the keyboard, while generated «notes» come from arpeggiator output.*

Input Filter receives MIDI events from Host – basically key press and release events (also it may be pitch bend, aftertouch and controller messages). From key «on» and «off» events, it generates *Key List* – an ordered list of keys with corresponding velocities (velocity is how hard you pressed a key).

«*In keys Pre-Filter*» is a key list as it comes from Host (keys are ordered as they were pressed). «*In keys Post-Filter*» represents the same key list after ordering, missing keys substitution and real-time quantization (for further details on these settings, go to page 14).

«*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:



```
Pos: -      Beat: 6.0    Step: 7.0        In keys pre-filter:  A2, C3, E3, F3, -        Note out:    A1
P011:Cur_Chain = 2 / 32                   In keys post-filter: A2, C3, E3, F3, A2       Detected chord:   F maj7
```

See Information panel description for more details, page 28.

Arp Engine refers to Value bars (matrix editor), which contain pattern information for each step. For example, «key select» value bar determines which key to take from the list for the current step (k1 – key 1, k2 – key 2, fix – fixed key, etc.). «Step type» value tells whether this step is a normal note (Nrm), the rest/sustaining note from the previous step (Rst) or muted (Off). Refer to page 24 for more information about Value bars and Matrix editor.

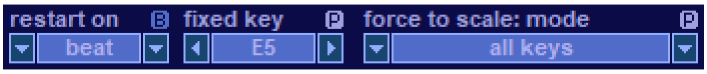
BlueARP has unique «*missing keys substitution*» feature. It works like this: when you have, for example, 4-keys pattern and play only 2-key chord you can select if you want steps for keys 3 and 4 to be silent or to be substituted with the existing keys. There are several substitution algorithms, see page 14 for details.

Output Filter adds some post-processing to generated notes – octave / semitone transposition, wraps notes to fit the given range. See page 18 for more details.

Program chains block allows you to merge several programs together to create longer patterns. You can automate current chain parameter and switch chains on the fly - it’s great for live performances. See page 27 for more details.

Interface

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


restart on B
beat
fixed key P
E5
force to scale: mode P
all keys

There are several ways to adjust the value:

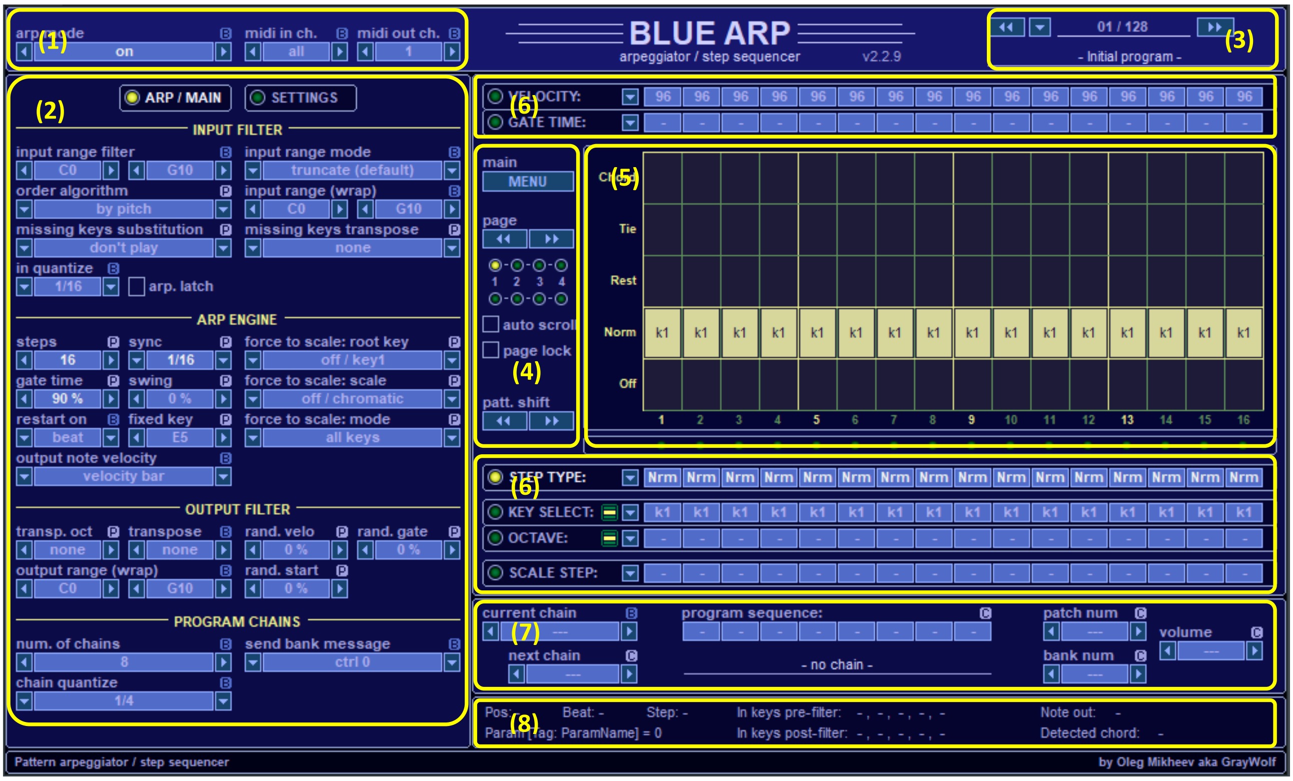
- left-click and hold on the box, drag it up or down;
- place the pointer over the box, use mouse wheel to adjust the value;
- click ◀ ▶ buttons to adjust the value or ▼ button to select value from drop-down menu;

B / P / C marks next to many controls 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 12).

Main window layout

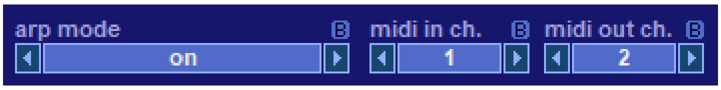


arp mode: on
midi in ch.: all
midi out ch.: 1
01 / 128 - Initial program -
input range filter: C0 – G10
input range mode: truncate (default)
order algorithm: by pitch
input range (wrap): C0 – G10
missing keys substitution: don't play
missing keys transpose: none
in quantize: 1/16
☐ arp. latch
steps: 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
fixed key: E5
force to scale: mode: all keys
output note velocity: velocity bar
transp. oct: none
transpose: none
rand. velo: 0 %
rand. gate: 0 %
output range (wrap): C0 – G10
rand. start: 0 %
num. of chains: 8
send bank message: ctrl 0
chain quantize: 1/4
VELOCITY: 96 96 96 96 96 96 96 96 96 96 96 96 96 96 96 96
GATE TIME: - - - - - - - - - - - - - - - -
☐ auto scroll
☐ page lock
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: ---
next chain: ---
program sequence: - no chain -
patch num: ---
bank num: ---
volume: ---
Pos: Beat: - Step: - In keys pre-filter: -, -, -, -, - Note out: -
Param[Tag: ParamName] = 0 In keys post-filter: -, -, -, -, - Detected chord: -

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

1. **Top panel** contains arp mode, midi in channel and midi out channel. All are bank-related (B), when you switch programs, these settings remain the same;
2. **Left panel** 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). They are saved with the current program;
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 shift buttons and LEDs indicating which page is currently playing and being edited;
5. **Matrix editor** represents step-related values for the selected value bar;
6. **Value bars** represent step-dependent pattern parameters. To select value bar, 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



arp mode on

midi in ch. 1

midi out ch. 2

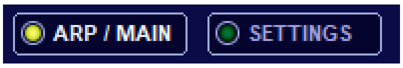
arp mode	Arpeggiator is turned On
values	*off, on, thru*
comments	*off* – BlueARP is inactive, all input keys are ignored *on* – BlueARP is enabled, normal mode. *thru* – BlueARP passes midi notes from input to output without arping, but some settings will still work (input range, output range, transpose, force to scale). In *thru* mode, you can use BlueARP as a real-time MIDI transpose tool and/or keyboard range filter.

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

Block (2): Left panel: ARP / MAIN

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



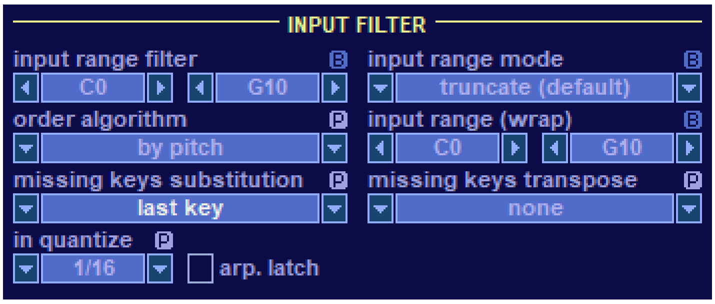
SETTINGS page contains rarely used bank-related settings, described on page 21 (you don't need to change them often, so I moved them to a separate page to save space on the left panel).

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 range filter B: C0 G10

input range mode B: truncate (default)

order algorithm P: by pitch

input range (wrap) B: C0 G10

missing keys substitution P: last key

missing keys transpose P: none

in quantize P: 1/16

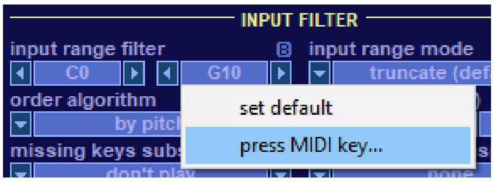
☐ arp. latch

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.

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	range for filtering out input notes
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» bar will be the lowest key. Sometimes it’s not the best way to order pressed keys. For example, if you play 1-key bass line, it’s better to set order algorithm to «as played, desc». In this case «k1» will always be the last pressed key. «chord (normalized)» can be explained by example. You press C4+E4, Cmaj chord is detected. Ordered list will be C4+E4+G4 (complete Cmaj chord). If you play inverted Cmaj – G3+C4+E4, output will be the same, because chord is normalized. «chord (as played)» behaves the same way, 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 the keys C5 and E5, while «KEY SELECT» value bar 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)*

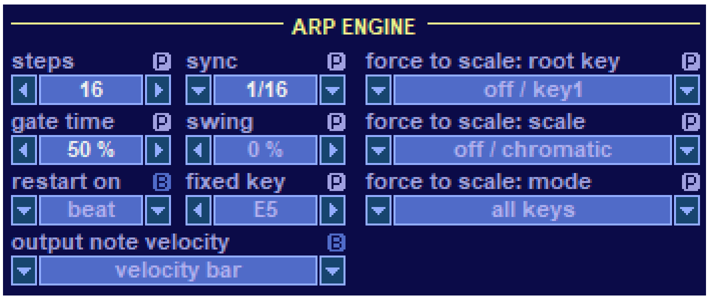
missing keys transpose	additional transpose for substituted keys
values	*none, -1 octave, +1 octave*
comments	Specifies additional transposition for substituted missing keys. For the example above, if we set missing keys transpose to +1 octave, post-filter key list will be: • *don't play* *«C5, E5, -, -, -»* • *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.

arp. latch	Latch (or hold) pattern
values	*On, Off (checkbox)*
comments	When checked, BlueARP will continue to play pattern for the last pressed chord even after all input keys are released, until another key is pressed. For live performances it may be useful to assign "arp.latch" to sustain pedal, or to switch it or to free your hands from the keyboard to do some other stuff.

Arp Engine



steps: 16
sync: 1/16
force to scale: root key: off / key1
gate time: 50 %
swing: 0 %
force to scale: scale: off / chromatic
restart on: beat
fixed key: E5
force to scale: mode: all keys
output note velocity: velocity bar

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 is a special mode, in this case BlueARP works as a MIDI thru. It works the same as arp mode = 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, key, 1st key*
comments	In default «beat» 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. With «key» setting, BlueARP will restart pattern each time new key/chord is pressed, after all previous keys were released. In "1st key" mode pattern will start with the first key/chord pressed and will keep going until you restart playback in a DAW.

fixed key	Fixed key value
values	*C0 .. G10 (MIDI notes 0 .. 127)*
comments	In «KEY SELECT» bar, you can set any step to «Fixed», it tells BlueARP to ignore input keys and take «fixed key» value. Set all steps to «Fixed» to use BlueARP as a step sequencer.

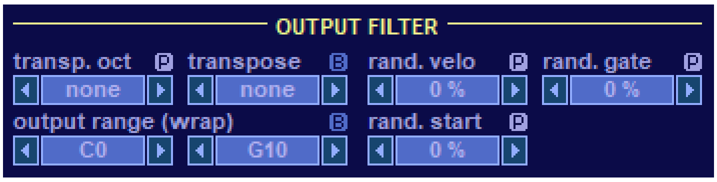
output note velocity	Sets where to take velocity for generated notes
values	*"velocity bar", "input key", "bar + input key"*
comments	"bar + input key": BlueARP takes output note velocity from VELOCITY bar and adjusts it to input note velocity (multiplying and normalizing them)

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 fill fit all output notes to the given scale; 2. "SCALE STEP" value bar will transpose notes in scale steps. Say if your scale is C Major, you pressed **D4** and scale step=+1, the output note will be **E4**. With "off/chromatic" selected, "SCALE STEP" will work as a semitone transposition. With "detect from chord" selected, BlueARP will derive scale from a chord you play. It's not very smart, but at least it will give "Major/minor" scales for Major/minor chords.

force to scale: mode	tow 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).

Output Filter



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

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	Is is program-related.

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

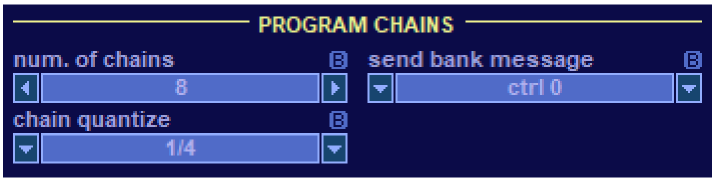
output range (wrap)	Range for output notes (wrapping)
values	*C0 .. G10 (MIDI notes 0 .. 127)*
comments	Notes outside the range will be wrapped (transposed up or down an octave to fit the range). Works just like «input range (wrap)», but for output notes.

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



num. of chains: 8

send bank message: ctrl 0

chain quantize: 1/4

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

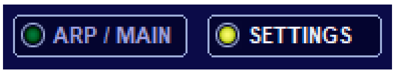
num. of chains	sets maximum value for "current chain" parameter
values	*1 .. 16*
comments	To switch chains with a midi controller, you need to automate "current chain" parameter. If you use a knob for this, setting "num.chains" to the appropriate value will utilize full rotation range of this knob.

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's the default setting.

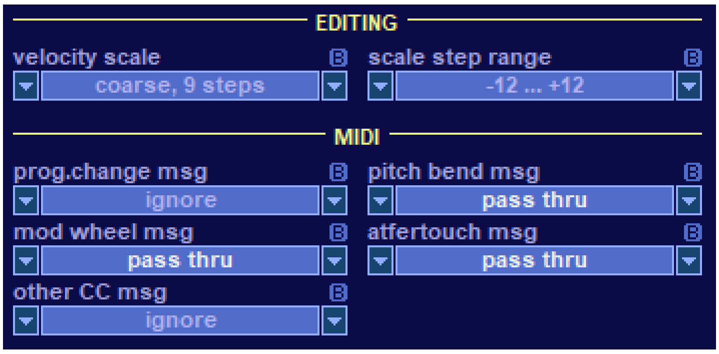
send bank message	selects bank/patch change MIDI message format
values	*ctrl 0, ctrl 32, ctrl 0+32*
comments	Relevant for controlling hardware synths, some VST synths will also react to this message. Sylenth1 does, for example. When you switch chains, BlueARP may send program/bank change to its MIDI 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.

Block (2): Left panel: SETTINGS

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



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


EDITING

velocity scale
coarse, 9 steps

scale step range
-12 ... +12

MIDI

prog.change msg
ignore

pitch bend msg
pass thru

mod wheel msg
pass thru

aftertouch msg
pass thru

other CC msg
ignore

EDITING block defines matrix editor behavior.

MIDI block contains settings for MIDI message filtering.

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

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

prog.change msg	Sets 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's connected to).

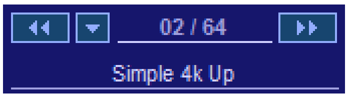
pitch bend msg	Sets how to respond to incoming Pitch Bend MIDI message
values	*"ignore", "pass thru"*
comments	"pass thru" - BlueARP will pass this message to its MIDI out (= to VST plugin it's connected to).

mod wheel msg	Sets how to respond to Modultaion Wheel MIDI message
values	*"ignore", "pass thru"*
comments	"pass thru" - BlueARP will pass this message to its MIDI out (= to VST plugin it's connected to).

aftertouch msg	Sets how to respond to incoming aftertouch MIDI message
values	*"ignore", "pass thru"*
comments	"pass thru" - BlueARP will pass this message to its MIDI out (= to VST plugin it's connected to).

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.

Block (3): Program browser

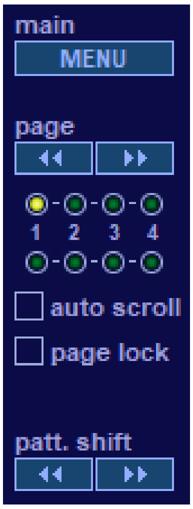


Use buttons to navigate through the programs in a current bank.

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



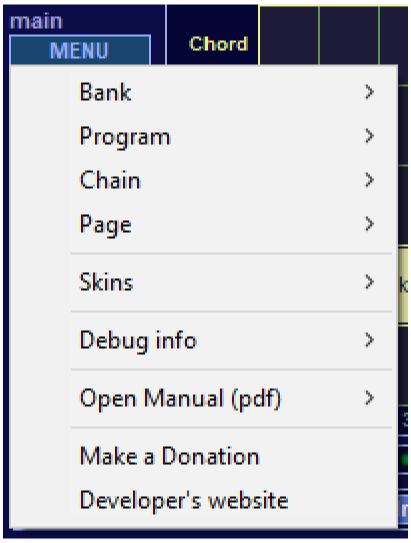
Main menu button calls drop-down menu with Bank load/save, Program load/save and some other functions.

page buttons are necessary when you pattern is longer than 16 steps, so it doesn't fit single screen. There are 2 small LED bars 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 1 step shifting. It's useful, when your pattern doesn't match the beat and you want to align it. The shift is cyclic, so when you shift the patter right, the last step won't disappear but will «jump» to the beginning.



Main menu includes:

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

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

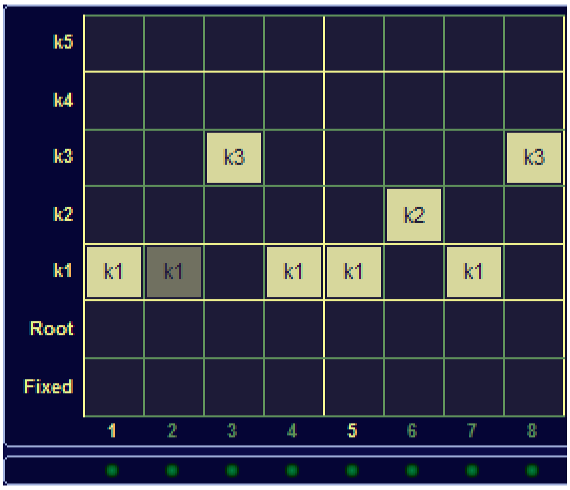
Chain – copy/paste and initialization for chains.

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

Skins – select color theme.

Skins are stored as *.ini files in "skins" directory. Default blue color scheme is built in. It's possible to create your own schemes by editing *.ini file.

Block (5): Matrix editor



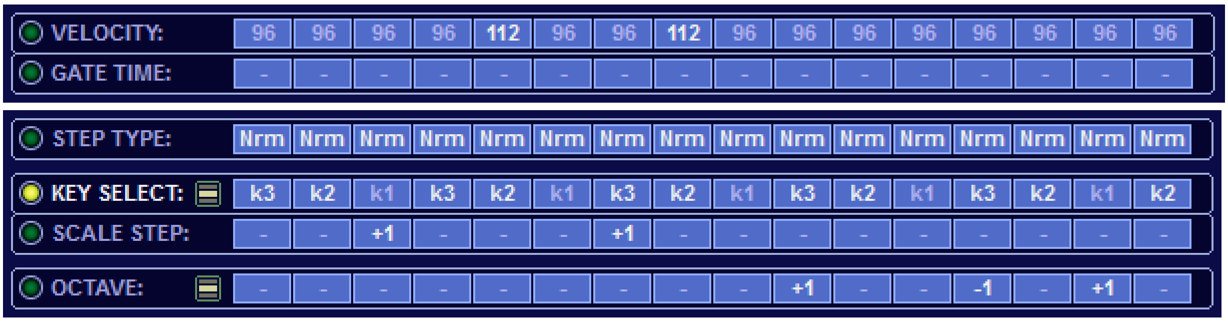
Matrix editor allows you to adjust current "value bar" values in a more friendly graphic way.

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

Grayed-out bricks mean that this particular setting doesn't affect generated pattern. On the picture above, steps 2 is set to Off, so «key select» value for this step doesn't make any difference.

Block (6): Value bars



VELOCITY:	96	96	96	96	112	96	96	112	96	96	96	96	96	96	96	96
GATE TIME:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
STEP TYPE:	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm	Nrm
KEY SELECT: ▤	k3	k2	k1	k3	k2	k1	k3	k2	k1	k3	k2	k1	k3	k2	k1	k2
SCALE STEP:	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-	-	-
OCTAVE: ▤	-	-	-	-	-	-	-	-	-	+1	-	-	-1	-	+1	-

Value bars represent step-related pattern parameters. Selected value bar is also shown in Matrix editor.

To adjust value for a certain step, click on it and:

- drag it up or down to increase / decrease the value;
- use mouse wheel to do the same thing;

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

See descriptions for each value bar below.

VELOCITY	Velocity value for each step
values	*0, 16, 32 .. 127;*
comments	Default value is 96. Use it to set 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	Gate time multiplier for each step
values	*1/16; 1/8; 1/4; 1/2; -; 2x; 4x; 8x; 16x;*
comments	Multiplies gate time by a given value. "-" means no change (default value). For example, with gate = 60% and GATE TIME for a step = "2x" note length for this step will be 60% * 2 = 120% or 1.2 steps.

STEP TYPE	Several options for output note generation
values	*Off – this step doesn't generate any note* *Nrm – Normal(default) – generates a note;* *Rst – this step will play the Rest of the previous step;* *Tie – this note will overlap with the previous one (for glides);* *Chr – Chord, or triggering all notes at once*
comments	«*Rst*» step simply means that this step continues to play the note from the previous step. You may chain several «*Rst*» steps together to make longer notes. «*Tie*» option may be tricky and not self-describing. It's main purpose is to create «glides» between notes. But it requires configuring synth properly – set it to monophonic mode, with legato and portamento on. In this case, when you press keys with overlapping (like press key1, press key2, release key1), sound pitch will glide between the notes, but not when you press them with gaps (like press key1, release key 1, press key2, release key2). When you configure the synth this way, «Tie» steps should create glides between the notes.

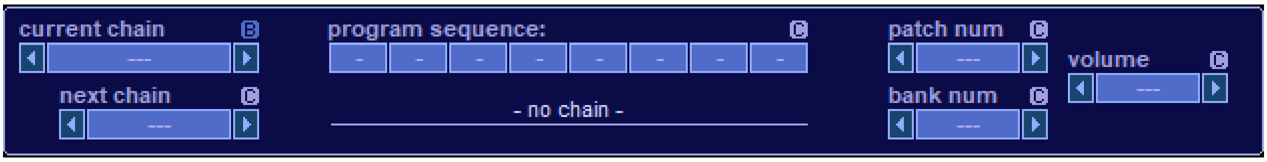
KEY SELECT	Input key selection for the given step
values	*Fixed* – use fixed key from Arp Engine settings *Root* – root key from detected chord, key1 if no chord detected *k1..k5* – take keys №1..5 from key list (post-filter)
comments	Determines which key to take from «post-filter key list» for the current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

Hint. *Fixed key doesn't depend on pressed keys, so you can set all steps to «fixed» 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.

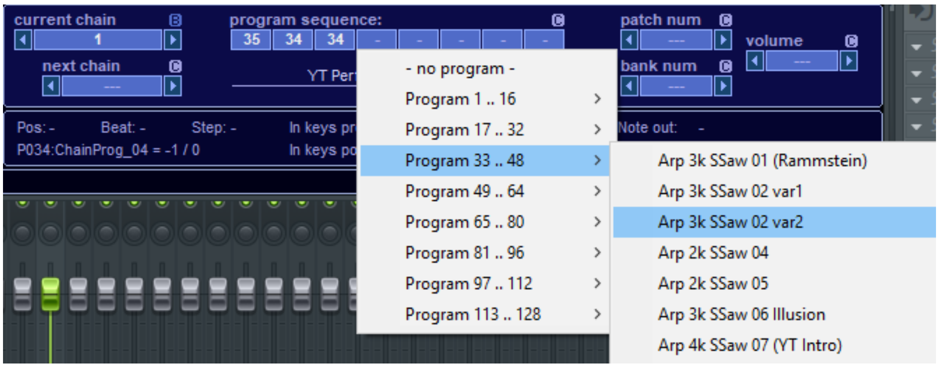
OCTAVE	Octave transposition for each step
values	*-3, -2, -1, 0, +1, +2; +3*
comments	It's convenient for bass lines, where the steps are usually transposed for the whole octaves. Yellow label next to "OCTAVE" caption (▤ or ☰) toggles between monophonic and polyphonic mode. In **monophonic** mode all 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 stack several programs (patterns) together into a longer "super-pattern". It’s done mostly with live performances in mind.

Program sequence bar 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 related 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 not «- - -».

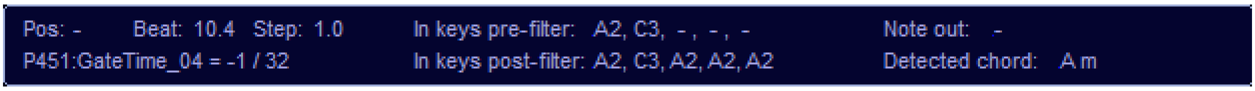
current chain	set current chain
values	*---, 1, 2, … number of chains (up to 16)*
comments	Current chan parameter can be automated; its maximum value is set by "num. chains" parameter on the left panel.

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): Information panel



Pos: - Beat: 10.4 Step: 1.0
P451:GateTime_04 = -1 / 32
In keys pre-filter: A2, C3, - , - , -
In keys post-filter: A2, C3, A2, A2, A2
Note out: -
Detected chord: A m

Shows current beat, step and some other information.

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

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

Note out – generated notes.

Links

Developer's website:
http://www.graywolf2004.net/

BlueARP discussion thread at KVR Audio forums (latest updates, news):
http://www.kvraudio.com/forum/viewtopic.php?p=5080757

Video demonstrations and tutorials are available on developer's YouTube channel:
http://www.youtube.com/user/graywolf2004ru?feature=watch

Please write bug reports and suggestions to KVR audio thread or email me at
graywolf2004@mail.ru
graywolf2004@gmail.com