

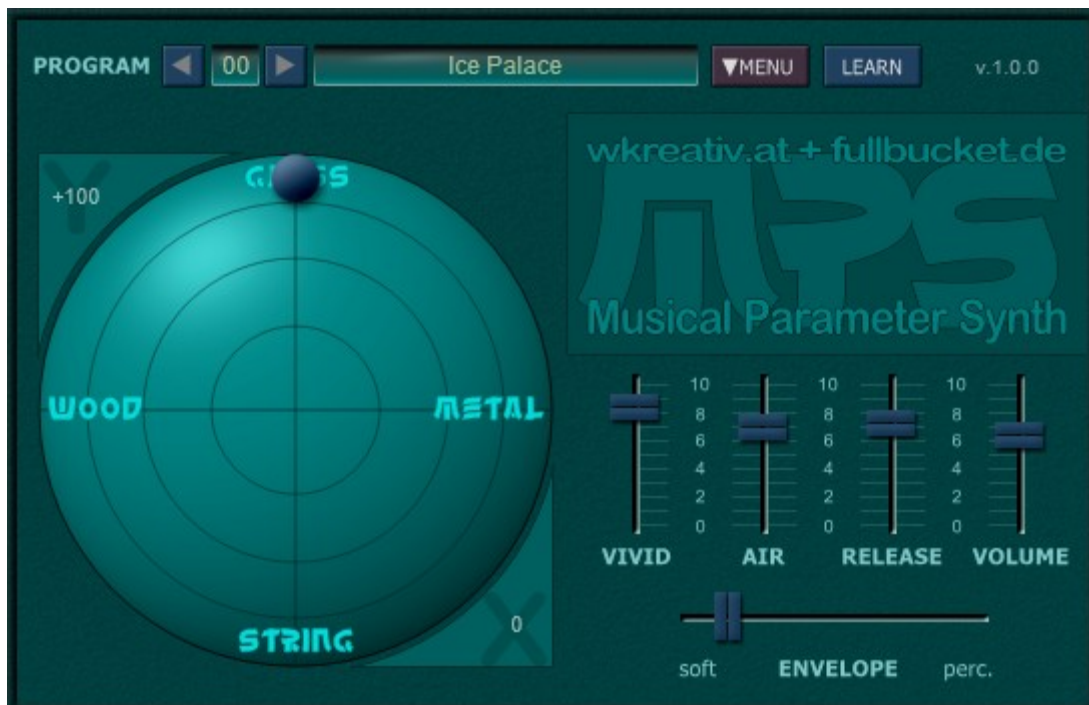
MPS

Musical Parameter Synthesizer

Version 1.0

© 2018 by Björn Arlt @ Full Bucket Music
<https://www.fullbucket.de/music>

and Wolfgang Kraus @ wkreativ.at
<http://www.wkreativ.at/>



Audio Units

VST is a trademark of Steinberg Media Technologies GmbH

Windows is a registered trademark of Microsoft Corporation

The Audio Units logo is a trademark of Apple Computer, Inc.

Table of Contents

Introduction.....	3
Background.....	3
Sound Generation.....	4
Parameters.....	4
The <i>Envelope</i> Parameter.....	4
The <i>Vivid</i> Parameter.....	4
The <i>Air</i> Parameter.....	4
The <i>Release</i> Parameter.....	4
The <i>Volume</i> Parameter.....	4
Performance Controls.....	5
Program Menu.....	5
Options Menu.....	5
MIDI Learn.....	6
The mps.ini Configuration File.....	6
MIDI Control Change Messages.....	6
Parameters.....	7
Global.....	7
Frequently Asked Questions.....	8

Introduction

MPS – short for “Musical Parameter Synthesizer” – is a software synthesizer plug-in for Microsoft Windows (VST) and Apple macOS (VST/AU). It is written in native C++ code for high performance and low CPU consumption. The main features are:

- Unconventional sound creation approach
- Fast & easy editing using just a few “musical” parameters
- Straightforward user interface
- 64 voices polyphony
- MIDI Learn – all parameters can be controlled by MIDI CC
- Double precision audio processing
- Plug-in supports Windows and macOS (32 bit and 64 bit)

The aim of *MPS* is to create sounds fast and easy using a small intuitive set of parameters without bothering about the technical details.

Background

In October 2017, I (Björn) received a mail from Wolfgang (whom I never met before). Therein Wolfgang described a novel approach for creating sounds by simple “musical” parameters like *Soft* or *Percussive*, *Wood* or *Metal*, and so forth. A hardcore synthesist knows how to translate these tags to the parameter settings of a “real” synthesizer, but this often requires changing many parameters at the same time which can get pretty tedious especially in real-time. Furthermore, the layman may not even understand what is going on.

Take for example the amplitude envelope of a sound: When we say the sound is “soft” we typically expect its envelope to start with a long attack time and to stop with a potentially long release time. On the other hand, a “percussive” sound is expected to start instantly and to decay pretty fast. With a standard ADSR generator one would have to address four parameters (Attack, Decay, Sustain, Release) while *MPS* requires just *one* parameter ranging from “soft” to “percussive” (OK, there is also an additional *Release* parameter for your convenience...).

Of course, when you reduce the number of parameters you lose part of the overall flexibility of the synthesizer. Thus, you cannot expect *MPS* to produce all the sounds that a classical synthesizer can do. However, you can achieve an extremely fast and intuitive sound creation workflow (assuming that the “parameter reduction” was done in a smart way)!

So Wolfgang created the concept and the basic UI layout of *MPS* while I myself contributed the necessary DSP/UI coding plus some experience in plug-in creation 😊. The resulting instrument is really fun to play with while not being just a cheap toy – do not get fooled by its simplistic appearance!

Sound Generation

MPS is a 64-voice polyphonic instrument that consists of five individual sound generators coming in the flavors *Wood*, *Glass*, *Metal*, *String*, and *Sine*. Their output can be mixed using the X-Y "Ball Pad" control (the *Sine* generator sits right in the middle of the pad and is not explicitly labeled).

Via dragging the ball with the mouse you can continuously change the mix balance between the five generators. The more you deviate from the center, the more harmonics will be created – which harmonics depends on the orientation of the ball. Play with it!

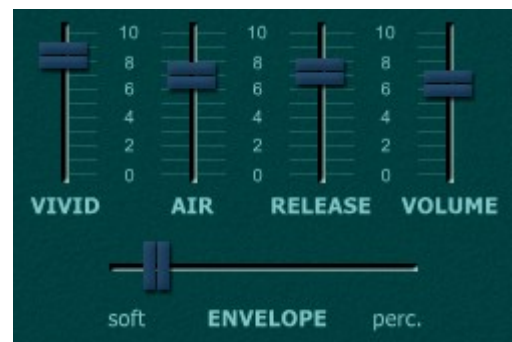


Parameters

Besides the Ball Pad control, *MPS* features five more parameters – that's it!

The *Envelope* Parameter

This parameter controls the contour of the amplitude and brilliance of the sound. "Soft" stands for smooth attack and decay while "Perc." leads to more percussive envelopes.



The *Vivid* Parameter

This parameter adds modulation (life!) to the sound. Each generator has its own kind of modulation: an Ensemble-like effect for *String*, Tremolo for *Metal*, a Phaser for *Glass*, a combination of filtered Ring modulation plus complex Tremolo for *Wood*, and a slightly detuned additional octave for *Sine*.

The *Air* Parameter

As the name says it, *Air* will bring some air to the sound. Technically it is realized by a double stereo Chorus. But please do not think in technical terms – put on your headphones and *listen*!

The *Release* Parameter

Release controls the length of the release time of the sound. Unlike the standard release parameter known from ADSR generators, the release time here cannot be longer than the decay time set via the *Envelope* parameter. This results in a more natural way of playing percussive sounds with short decay.

The *Volume* Parameter

Does exactly what you expect.

Performance Controls

Program Menu

If you know my other plug-ins then there will be no surprises: To select one of the 64 patches just click on the program number, and edit its name by clicking in the text field.

Options Menu

When clicking on the *Options* button, a context menu opens with the following options:

Copy Program	Copy current program to internal clipboard
Paste Program	Paste internal clipboard to current program
Load Program	Load a program file containing a patch to the current program
Save Program	Save the current program to a program file
Load Bank	Load a bank file containing 64 patches into <i>MPS</i>
Save Bank	Save the 64 patches to a bank file
Init Program	Initialize the current program
Reload Configuration	Reload the configuration file (see section <i>The mps.ini Configuration File</i>)
Save Configuration	Saves the configuration file (see section <i>The mps.ini Configuration File</i>)
Select Startup Bank	Select the bank file that should always be loaded when <i>MPS</i> is started
Load Startup Bank	Load the Startup bank file; can also be used to check what the current Startup bank is
Unselect Startup Bank	Unselect the current Startup bank
Check Online for Update	When connected to the Internet, this function will check if a newer version of <i>MPS</i> is available at fullbucket.de
Visit fullbucket.de	Open fullbucket.de in your standard browser

Note: If the *MPS* DLL and the `mps.ini` configuration file are located in a “protected” Windows folder (for example somewhere under “`C:\Program Files\...`”) then it may not be possible to save the configuration. In this case it is strongly recommended to use a `VstPlugins` directory that lies “outside” the protected folder.

MIDI Learn

Every parameter of *MPS* can be controlled by one MIDI controller. If you want to change the assignment of MIDI controller (CC; *MIDI Control Change*) to *MPS* parameter the *MIDI Learn* function comes in quite handy: Just click the *LEARN* button on the *MPS*'s control panel (caption turns red) and wiggle both the MIDI controller and the parameter you want to assign (you can abort *LEARN* by clicking the red button again). To save the controller assignments use "Save Configuration" in the *Options* menu (see previous section).

If you want to unlearn the assignment, right-click the *LEARN* button (the label now reads "UNLEARN") and activate it. Now wiggle the MIDI controller or the parameter that you want to unlearn.

The mps.ini Configuration File

MPS is able to read some settings from a configuration file (`mps.ini`) located in the same directory as the *MPS* VST DLL (`mps.dll` or `mps64.dll`) or Mac VST/AU (`MPS.component` or `MPS.vst`) itself. After you have edited this INI file in a text editor, you have to reload it using the *Reload Configuration* command from the *Options* menu (see section *Performance Controls*).

MIDI Control Change Messages

All parameters of *MPS* can be controlled by MIDI controllers, or more precise: Each MIDI controller (except *Modulation Wheel* and *Sustain Pedal*) can control one of the *MPS*'s parameters. The mapping is defined in the `mps.ini` for example like this:

```
[MIDI Control]
CC7  = 0 # Volume
CC16 = 1 # Variation X
CC17 = 2 # Variation Y
...
```

The syntax is straight forward:

```
CC<controller number> = <parameter ID>
```

Given the above example, controller 16 directly controls the *Variation X* parameter, controller 17 the *Variation Y* etc. As you can see, comments are introduced by the Pound sign (#); they are here just for description purposes and completely optional.

The *parameter ID* of one of the *MPS*'s parameters is given in the section *Parameters* below. Note that the *controller number* can run from 0 to 119, with the exception of 1 (*Modulation Wheel*) and 64 (*Sustain Pedal*); the latter two are simply ignored.

Of course, instead of editing the controller/parameter assignments in `mps.ini` with a text editor it is much easier to use the *MIDI Learn* function and save the configuration (see sections *Options Menu* and *MIDI Learn* above).

Parameters

Global

parameter	ID	description
<i>Volume</i>	0	Master volume
<i>Variation X</i>	1	X-Y Ball Pad horizontal axis
<i>Variation Y</i>	2	X-Y Ball Pad vertical axis
<i>Vivid</i>	3	Vivid parameter value
<i>Air</i>	4	Air parameter value
<i>Release</i>	5	Release parameter value
<i>Envelope</i>	6	Envelope parameter value

Frequently Asked Questions

How do I install MPS (Windows 32 bit version)?

Just copy the files `mps.dll` and `mps.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST plug-in folder. Your DAW should automatically register the *MPS* VST plug-in the next time you start it.

How do I install MPS (Windows 64 bit version)?

Just copy the file `mps64.dll` and `mps.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST plug-in folder. Your DAW should automatically register the *MPS* VST plug-in the next time you start it.

Note: You may have to remove any existing (32 bit) `mps.dll` from your VST plug-in folder or else your DAW may screw the versions up...

How do I install MPS (Mac VST/AU universal 32/64 bit)?

Locate the downloaded PKG package file `mps_1_0_0_mac.pkg` in Finder (!) and do a right- or control-click on it. In the context menu, click on "Open". You will be asked if you really want to install the package because it comes from an "unidentified developer" (me ☺). Click "OK" and follow the installation instructions.

What is the plug-in ID of MPS?

The ID is `M P S 1` .

Can I save the MIDI controller/parameter assignments?

Yes, by using "Save Configuration" in the *Options* menu (see section *Options Menu*).

Why doesn't the configuration file (mps.ini) get saved?

Under Windows, you may have placed the *MPS* files `mps.dll/mps64.dll` and `mps.ini` into a "protected" folder somewhere under "C:\Program Files\..."? In this case you should use a `VstPlugins` directory that lies "outside" the protected folder (also see section *Options Menu*).

How do I know if a new version of MPS is available?

When connected to the Internet, open the *Options* menu (see section *Options Menu*) by clicking the disk icon and select the entry "Check Online for Updates". If a new version of *MPS* is available on fullbucket.de the respective information will be shown in a message box.