MPS

Musical Parameter Synthesizer

Version 2.0

© 2018–2024 by Björn Arlt @ Full Bucket Music
https://www.fullbucket.de/music

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

Table of Contents

Introduction

MPS – short for "Musical Parameter Synthesizer" – is a software synthesizer plug-in for Microsoft Windows (VST2/VST3/CLAP/AAX) and Apple macOS (VST2/VST3/CLAP/AU/AAX). 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
- MTS-ESP (https://oddsound.com/) dynamic micro-tuning support
- Plug-in supports Windows and macOS

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

MPS is based on the **iPlug2** framework maintained by **Oli Larkin and the iPlug2 team**. Big thanks, guys!!! Without your work it would not have been possible to create a resizable *MPS* user interface.

To resize the plug-in you just grab the yellow triangle at the bottom right of the window and drag it. You can save the current window size using the menu entry "Save Window Size" in the *Options Menu* or by right-clicking somewhere into an empty space of *MPS*'s panel.

If you have trouble with the standard version of *MPS*, please grab the (sound-wise identical) "N" version of the plug-in which is based on the original **iPlug** framework.

Changes in Version 2.0

Version 2 adds a second "ball" to the X-Y pad. The *MPS* sound generator can – but does not have to – "morph" between the two ball positions (either periodically or in a "one-shot" manner). This new feature enhance the overall sound palette.

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 loose 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 two balls 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 a ball. Play with it!

Note that the second, reddish ball will only be visible and have an effect if the *Morph Mode* (see section *Morphing*) is set to anything else but "off".
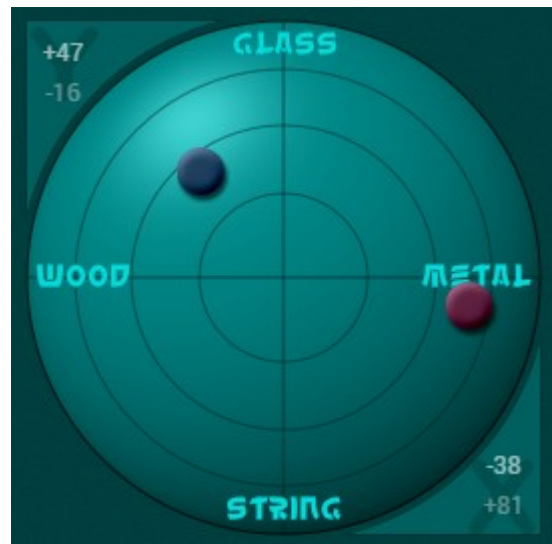
Parameters

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

The *Envelope* Parameter

This parameters 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.

Morphing

The new morphing feature of *MPS* version 2 allows to dynamically vary the mixture of the different sound generators. This is done by "morphing" between the positions of the two balls on the X-Y pad. Two parameters control the morphing.

The *Morph Mode* Parameter

There are four *Morph Modes*:

- **"off"**
  There is no morphing at all.

- **"once"**
  Once you press a key, the mixing position between the sound generators starts at the position of the reddish ball and moves to the position of the blue ball. This process will only be restarted if all keys have been released.

- **"one-shot"**
  When pressing a key, the mixing position starts at the position of the reddish ball and moves to the position of the blue ball. However, this will happen for each key (respectively voice) individually.

- **"swing"**
  The mixing position "swings" periodically between the blue and the reddish ball.

The *Morph Rate* Parameters

As you might have guessed, this parameter controls the speed or rate of the morphing process.

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 *MPS*
Save Bank	Save the 64 patches to a bank file
Init Program	Initialize the current program
Select Startup Bank	Select the bank file that should always be loaded when *MPS* 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
Default Path for Program Files	Sets the default path for program and bank files
MIDI Thru	Set globally if MIDI data sent to *MPS* should be sent through to its MIDI output (stored in configuration file)
Ignore Program Change	Set globally if MIDI Program Change data sent to the *MPS* should be ignored (stored in configuration file)
Reload Configuration	Reload the configuration file (see section *The mps.ini Configuration File*)
Save Configuration	Saves the configuration file (see section *The mps.ini Configuration File*)
Save Window Size	Stores the current window size to the configuration file so that it will be restored on the next loading of *MPS*
Check Online for Update	When connected to the Internet, this function will check if a newer version of *MPS* is available at fullbucket.de
Visit fullbucket.de	Open fullbucket.de in your standard browser

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`); the exact location of this file depends on your operating system and will be displayed when you click on "Reload" or "Save Configuration".

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
Volume	0	Master volume
Variation X	1	X-Y Ball Pad horizontal axis of the blue ball
Variation Y	2	X-Y Ball Pad vertical axis of the blue ball
Vivid	3	Vivid parameter value
Air	4	Air parameter value
Release	5	Release parameter value
Envelope	6	Envelope parameter value
Variation X2	7	X-Y Ball Pad horizontal axis of the reddish ball
Variation Y2	8	X-Y Ball Pad vertical axis of the reddish ball
Morph Mode	9	Morph mode (*off*, *once*, *one-shot*, *swing*)
Morph Rate	10	Rate or speed of the morphing effect

Frequently Asked Questions

How do I install **MPS** *(Windows VST2 32 bit version)?*

Just copy the files `mps.dll` from the ZIP archive you have downloaded to your system's or favorite DAW's VST2 plug-in folder. Your DAW should automatically register the *MPS* VST2 plug-in the next time you start it.

How do I install **MPS** *(Windows VST2 64 bit version)?*

Just copy the file `mps64.dll` from the ZIP archive you have downloaded to your system's or favorite DAW's VST2 plug-in folder. Your DAW should automatically register the *MPS* VST2 plug-in the next time you start it.

Note: You may have to remove any existing (32 bit) `mps.dll` from your VST2 plug-in folder or else your DAW may screw the versions up…

How do I install **MPS** *(Windows VST3 64 bit version)?*

Just copy the files `mps.vst3` from the ZIP archive you have downloaded to your system's or favorite DAW's VST3 plug-in folder. Your DAW should automatically register the *MPS* VST3 plug-in the next time you start it.

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

Copy the file `mps_AAX_installer.exe` from the ZIP archive you have downloaded to any of your system's folder and run it. Your AAX-enabled DAW (Pro Tools etc.) should automatically register the *MPS* AAX plug-in the next time you start it.

How do I install **MPS** *(Mac)?*

Locate the downloaded PKG package file 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*).

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.

How do I resize the **MPS** *window?*

Just grab the yellow triangle at the bottom right of the *MPS* window and drag it. You can save the current window size using the menu entry "Save Window Size" in the *Options Menu*.