

PG-8X 2.0

USERS MANUAL



by MLVST (Martin Lueders) 2016



INTRODUCTION

The PG-8X is a virtual synthesizer, inspired by the Roland **JX-8P** with the **PG-800** programmer. The synth architecture is a standard 2-oscillator -> Filter -> VCA design. These elements can be modulated by a common LFO and one of two ENVELOPEs.

The **PG-8X** is patch compatible with the **JX-8P** and can read and write **JX-8P** Sysex data.

GETTING STARTED

Installation:

Copy the PG-8X.dll (for PC) or PG-8X.vst (for MAC) files into your standard VST folder (please, refer to your DAW documentation, where VSTs are expected to be located). No other files are necessary. After using the VST for the first time, an additional pg8x.ini file will be created. Make sure that the plugin has sufficient permissions to create and write that file.

Selecting a Preset:

The **PG-8X** implements it's own preset handling system, independent of the host. A preset can be selected either by clicking on the preset number in the display, which will open a drop-down menu, through the NEXT and PREV buttons, or via MIDI Program Change messages from a MIDI keyboard.

Loading a bank/presets:

To load a bank, click the LOAD button, which will open an operating system dependent file selector window. Navigate to the folder holding the banks or presets and select the file. For more information on how to import **JX-8P** Sysex files, please see section SYSEX.

THE SYNTH ELEMENTS

Digitally Controlled Oscillators (DCOs)



Both oscillators share controls over the RANGE (16' - 2'), the WAVE FORMS (SAW, PULSE, SQUARE and NOISE), TUNE (in semitones) and pitch modulation through the LFO or one of the envelopes. Which envelope is used and it's polarity is selected by the MODE switch, which affects both DCOs in the same way. Similarly, the DYNAMICS setting, which selects the response to the key velocity, is the same for both DCOs.

In addition, DCO-2 has a FINE-TUNE (-1 to +1 semitones) and the CROSS MODULATION switch with the modes:

- OFF: DCO-2 is independent of DCO-1
- 1: DCO-2 is hard-sync'ed to DCO-1
- 2: DCO-2 is hard-sync'ed and ring-modulated with DCO-1
- 3: DCO-2 is ring-modulated with DCO-1

MIXER



The operation of the mixer is pretty obvious. The output level of DCO 1 is controlled by a constant value DCO-1, while the level of DCO 2 can be controlled by a constant value plus (or minus) a level controlled by one of the envelopes.

Some not so obvious details are that the total level of DC02 saturates, i.e. when DCO-2 level is at 99, a further positive value for the envelope contribution does not change the total level. However, it is possible to reduce the level by using an envelope with a negative polarity (see MODE slider). As in other elements of the synth, DYNAMICS controls the velocity curve.

VCF



The filter section contains a simple 6dB/oct high pass filter (HPF) and a 24dB/oct low pass filter. The HPF does not have any dynamic controls and only 4 settings. It should be noted that '1' is the neutral setting, while '0' acts as a bass boost. '2' and '3' enable the fixed frequency high pass filter.

The low pass filter consists state variable filter, which are tuned to the same frequency and resonance. At high values of the resonance, the filter gets close to self-oscillation, but does not do so without a small input signal.

VCA



The VCA level controls the overall output of the plugin. If the sound is still too quiet or too loud, different gain levels can be set in the SETTINGS panel.

The VCA can either be controlled by the ENV-2 envelope generator, or simply by the gate signal. The velocity settings follow the same scaling curves as for the other envelope modulations.

LFO



The LFO can output three different types of signals: A SINE-like signal (actually a triangular shape), a SQUARE signal, and a random signal (also known as sample and hold).

These wave shapes only affect the modulations through the LFO-sliders in sections DCO and VCF. The LFO resulting from the modulation wheel is always of the SINE type, and the maximum mod-wheel modulation is controlled by MWHL INT.

The LFO signals are the same for all voices. A DELAY TIME can be specified, which results in a delay, starting whenever the first key is pressed after all keys have been released.

Envelope Generators



The PG-8X has two envelope generators, which emulate the (a bit sloppy) software generated envelopes of the JX-8P. KEY FOLLOW controls how the envelope times are affected by the note. In the OFF position, the envelope is independent of the key, while higher values lead to shorter times (faster envelopes) for higher notes.

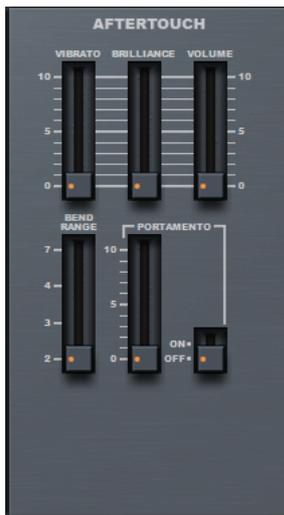
Envelopes are not reset on trigger, but will pick up at their current stage.

CHORUS



The chorus emulates a BBD based stereo chorus with two delay lines. The MODE switch controls the amount of modulation. Further controls (which are not available in the hardware) can be found in the SETTINGS panel.

PLAY CONTROLS



The standard view of the PG-8X shows the controls for Aftertouch, pitch bend range and portamento. Similar to the Super-JX and in contrast to the JX-8P, the aftertouch can be used simultaneously for vibrato, brilliance and volume. The plugin responds both to channel pressure and polyphonic aftertouch.

The PORTAMENTO switch is set to ON, the voice allocator uses a slightly different allocation scheme, which is geared towards portamento play.

KEY ASSIGNER



There are 6 different key-assignment modes, which control how the different voices are allocated, whenever a new key is pressed.

- POLY 1 (red LED): "normal polyphonic mode"

in this mode, a new key will in general trigger then next voice in a queue. When a key is released, the voice is added at the end of the queue. This ensures that sounds with long release phases can sound as long as possible. When a key is pressed which has been played recently, the voice will be retriggered, if possible. When PORTAMENTO is engaged, the voice queue is reset whenever all keys are released. This ensures that portamento behaves correctly.

- POLY 2 (yellow LED): "round robin"

every new key will trigger the next voice. Playing the same note repeatedly will trigger different voices for each note. In POLY 2 mode, the tuning of the voices is spread out evenly, according to the UNISON DETUNE parameter. This can lead to interesting effects when using sounds with long release time.

- UNISON 1 (red LED):

each note triggers two voices, leaving a polyphony of half the number of voices (see SETTINGS). The voices can be detuned using UNISON DETUNE.

- UNISON 2 (yellow LED):

similar to UNISON 1, but the second voice is played one octave below the first voice. This shift affects all parameters of the voice.

- MONO 1 (red LED):

The synth behaves like a monophonic (1 voice) synth. The key trigger mode can be set as legato, i.e. notes will not be retriggered when played overlapping, or re-trigger in the SETTINGS panel.

- MONO 2 (yellow LED):

Similar to MONO 1, but all voice play simultaneously and can be detuned against each other with UNISON DETUNE. The detuning is spread over the voices.

BASIC OPERATION

The sliders and knobs have the following functions:

- click and drag: relative motion of the slider
- SHIFT-click and drag: fine motion of the slider
- CTRL-click: immediate jump to new position
- double-click: jump to default setting (dependent on parameter)
- left mouse button click (PC) or CMD-click (MAC) opens the MIDI LEARN panel.

Knobs behave in linear fashion.

PRESET MANAGEMENT

Similar to the **JX-8P**, the **PG-8X** has a total of 128 memory slots for presets. However, in contrast to the hardware, all slots can be written by the user. Banks are organised in 4 sub banks with 32 presets, which correspond to the 32 tones per bank in the **JX-8P** (I, C, P, -P).

Loading Banks:

The LOAD button opens a file navigation window of the operating system. You can go to the directory, holding your patch data, and select any of the following files:

- *.PG8Xbank: contains a full bank in internal format
- *.PG8Xpreset: contains one patch in internal format
- *.syx: JX-8P (or Super-JX) sysex file, see SYSEX section

When loading JX-8P Sysex banks, the banks usually load into the presets 1-32, which correspond to the internal (I) memory slots of the JX-8P. In order to fill all memory slots of the PG-8X, use the ROTATE BANKS function (in a submenu which appears by clicking on the programme number). Each activation of ROTATE BANKS shifts the existing patches up by 32 positions.

Saving Banks:

The SAVE button first opens a sub-menu to select what to save:

- PG8Xpreset: current preset in native format
- PG8Xbank: whole bank in native format
- SYSEXpreset: current preset as TONE data JX-8P sysex .
- SYSEXbank:
 - 1-32, 33-64, 65-96, 97-128: 32 preset SYSEX files, which can be loaded by a JX-8P without cartridge

- 1-64, 65-128: 64 preset SYSEX files, which can be loaded by a JX-8P with memory cartridge
- 1-128: full bank (e.g. for the old pg-8x version)

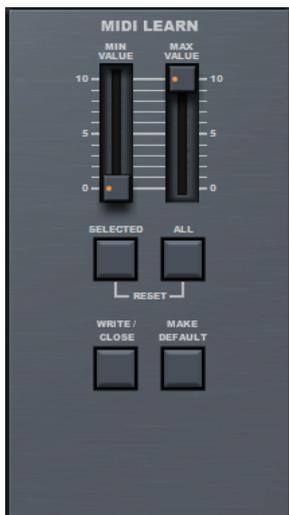
After selecting the format, the usual file navigator will open.

MIDI



The MIDI panel allows to set the MIDI channel and contains switched to enable or disable the sending of SYSEX messages on parameter changes or program changes.

MIDI LEARN



All controls of the PG-8X can be assigned to MIDI messages. To start the learn process, right-click on a slider or knob on the GUI. This changes the view to the MIDI LEARN panel, which displays the name of the control and the name of the MIDI message to be learned. To assign a specific MIDI message, simply move the controller on your MIDI hardware. Minimum and Maximum values of the parameter can be set using the corresponding sliders.

To store the Midi Learn data, click the WRITE/CLOSE button. This will also reset the panel to the normal view. It is recommended to store each parameter, before learning the next one.

Midi Learn data is stored per patch, but each bank also has a default Midi Learn map, which is active if no parameters have been assigned for a particular patch.

The default map can be created from a patch with the MAKE DEFAULT button.

SETTINGS



The SETTINGS panel contains controls over parameters, which are hard-wired in the hardware.

MASTERTUNE: can be set between

UNISON DETUNE: detunes the different voices.

- POLY 1, MONO 1: no effect
- UNISON 1/2: the two voices playing together are detunes against each other.
- POLY 2, MONO 2: all voices are evenly detuned

OCTAVE STRETCH: determines a slight deviation from the plain doubling of frequency with each octave. This gives a fuller sound and prevents phase locking.

FILTER NOISE: a small amount of white noise can be added to the signal before it is fed into the filter. This gives a slightly fuller sound when filter resonance is used.

MONO TRIGGER: Set the trigger mode for MONO 1/2. Possible modes are LEGato and RETrigger.

LFO SYNC: this switch determines whether the LFO is synched to the host tempo. If the sync is on, LFO speeds are displayed in fractions of the tempo.

CHORUS SETUP



The Chorus is a dual delay line chorus, where both delay lines are modulated out with opposite phase by a single LFO with triangular wave form. In the hardware, the BBD (bucket brigade delay) chorus is quite noisy, which for some gives character, for others is simply annoying. In the PG-8X, the SPEED, DEPTH and the level of the NOISE can be adjusted.

THE PG-8X.ini FILE

The PG-8X.ini file contains data, which are loaded when a new instance of **PG-8X** is into a project. In general, these information will be overwritten by the restored state of the plugin, when an existing project is reloaded.

#PG8X configuration file

```
DEFAULT_NUM_VOICES = 12 (6, 8, 10 or 12)
GLOBAL_GAIN = LOW (LOW, NORMAL, HIGH)
IMMEDIATE_WRITE = FALSE (TRUE or FALSE)
LOCK = FALSE (TRUE or FALSE)
MIDI_THRU = FALSE (TRUE or FALSE)
POLY_ON_GUI = FALSE (FALSE, TRUE or NEW)
EXTENDED_PB = FALSE (TRUE or FALSE)
PRESETDIR = (path to the preset directory)
DEFAULT_BANK = (name of the *PG8Xbank to load at startup)
SEND_PARAM_SYSEX = FALSE (TRUE or FALSE)
SEND_PROG_SYSEX = FALSE (TRUE or FALSE)
```

CREDITS:

JX-8P, **SUPER-JX** and **PG-800** are trademarks of ROLAND.



is trademark of Steinberg.

The WDL-OL, which was used for this plugin, was developed by Oliver Larkin.

Special thanks to ENV1 for the GUI!!