

# PARALOGY

**Paraphonic Synthesizer**  
Version 1.0

© 2025 by Björn Arlt @ Full Bucket Music  
[www.fullbucket.de/music](http://www.fullbucket.de/music)



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.  
AAX is a trademarks of Avid Technology, Inc.

# Table of Contents

|   |    |
|---|----|
| Introduction.....                               | 3  |
| Acknowledgments.....                            | 3  |
| The Italian Job.....                            | 4  |
| Disclaimer.....                                 | 4  |
| <i>Paralogy</i> Basics.....                     | 5  |
| Oscillator Section.....                         | 6  |
| Oscillator Glide.....                           | 7  |
| Waveform Generation.....                        | 7  |
| <i>Organ</i> Section.....                       | 8  |
| <i>String Ensemble</i> Section.....             | 8  |
| <i>Synthesizer</i> Section.....                 | 8  |
| Waveform Selection Section.....                 | 8  |
| Envelope Generator Section.....                 | 8  |
| Voltage Controlled Filter (VCF).....            | 9  |
| Voltage Controlled Amplifier.....               | 9  |
| Volume/Pan Section.....                         | 9  |
| Low Frequency Oscillator (LFO).....             | 10 |
| Modulation Section.....                         | 10 |
| Wheels Section.....                             | 10 |
| Phaser And Delay.....                           | 10 |
| Tweak Section.....                              | 11 |
| Program Section.....                            | 12 |
| Options Menu.....                               | 12 |
| The <i>paralogy.ini</i> Configuration File..... | 13 |
| MIDI Control Change Messages.....               | 13 |
| MIDI Learn.....                                 | 13 |
| Parameters.....                                 | 14 |
| General.....                                    | 14 |
| <i>Organ</i> Section.....                       | 14 |
| Oscillator/Glide Section.....                   | 14 |
| <i>Synthesizer</i> Section.....                 | 15 |
| LFO Section.....                                | 15 |
| <i>String Ensemble</i> Section.....             | 15 |
| Phaser Section.....                             | 16 |
| Delay Section.....                              | 16 |
| Tweak Section.....                              | 16 |
| Frequently Asked Questions.....                 | 17 |

## Introduction

*Paralogy* is a software instrument plug-in for Microsoft Windows and Apple macOS simulating the paraphonic Crumar *Trilogy* and *Stratus* keyboards from the early 1980s. It is written in native C++ code for high performance even on “lighter” systems. The main features are:

- Close simulation of the original hardware
- Fully “polyphonic” paraphonic madness
- *Organ, Synthesizer and String* sections
- Additional Phaser and Delay effects
- All parameters can be controlled by MIDI controllers
- Plug-in supports Windows and macOS (32 bit and 64 bit)

*Paralogy* 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 *Paralogy* 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 *Paralogy*’s panel.

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

## Acknowledgments

- **Oli Larkin** and the **iPlug2** team.
- **kraftraum** (<https://soundcloud.com/kraftraum>) did the Beta testing.

## The Italian Job

In 1981, the *Trilogy* and its little brother *Stratus*<sup>1</sup> were Crumar's answer to the still new and extremely popular polyphonic synthesizers coming from Japan and the US. Obviously, the idea of having a separate oscillator, filter and amplifier section *per voice* was striking. But it has one problem: You usually need a little computer or micro controller to schedule notes to voices.

Having a lot of expertise with string machines and multi-keyboards, the Italian company decided to combine a dual top octave generator/frequency divider design with a set of six voices. Each voice contains a voltage-controlled lowpass filter (VCF), a voltage-controlled amplifier (VCA) and an envelope generator (EG). All these components are realized with the famous CEM chips, which are used in countless instruments from Oberheim, Sequential Circuits and many other manufacturers, too.

Now *Trilogy* and *Stratus* do not use a computer for voice scheduling. Instead, the tone generators are hard-wired to the voices in a very unusual way: All the C and F# notes are sent to voice 1, the C# and G notes to voice 2, the D and G# notes to voice 3, etc. Consequently, when you play a C1 and then an F#3, the C1 will get re-triggered by the F#3. This is a very special kind of paraphony, and I cannot think of any other instrument that is like it!

But obviously this was not enough, so Crumar also added a (pretty limited) *Organ* section and (in case of the *Trilogy*) a *String Ensemble* section. Furthermore, the *Trilogy* also has seven presets for the *Synthesizer* section plus another "preset" which refers to the manual setting. However, the most clever marketing trick was giving *Trilogy* and *Stratus* a similar design like the Oberheim OB-X! You eat with your eyes...  
☺

The *Paralogy* plugin closely emulates all those quirky features. Thus, this manual will also explain how the original hardware works. You will see, there is paraphony all over the place...

## Disclaimer

As usual, I do not argue whether *Paralogy* sounds exactly like the *Trilogy/Stratus*. I don't have the original hardware and modeled everything from the schematics plus videos and sound examples on the Internet.

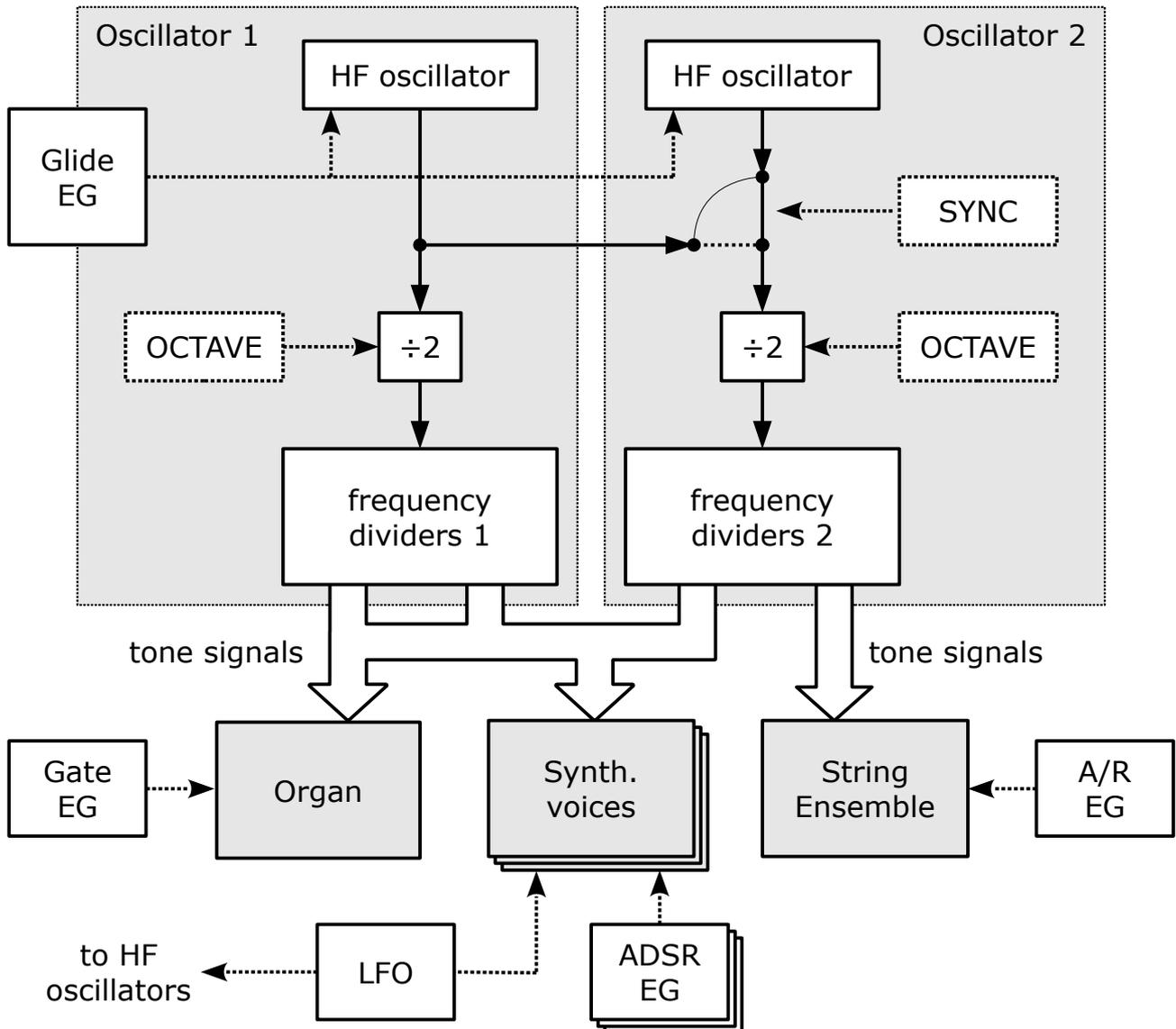
The *Paralogy* adds a Phaser and a Delay effect to the instrument, and I deliberately adopted the *Envelope Generator* section of the *Stratus* (not the *Trilogy's* ugly slider-based thingy). There are also a few tweak parameters including master volume and master tune (!) but no velocity sensitivity. Instead of the *Trilogy's* preset section, the plugin comes with my decent but not too spectacular program management system. Some hate it, some won't care.

---

1 The *Stratus* is almost identical to the *Trilogy* but misses the String section and the synthesizer presets.

## Paralogy Basics

*Paralogy* is fully "para-polyphonic" – you can press all 128 MIDI keys at once. The instrument consists of three main sections: *Organ*, *Synthesizer* and *String Ensemble*. Additionally, there is a Phaser and a Delay effect not available on the original hardware. However, to understand what happens where and why, here is a nice diagram which will be explained in detail below.



**Block diagram of *Paralogy*'s components**

## Oscillator Section

The basic principle is as follows: All tone signals are generated by dividing down the clock signal of a high-frequency (HF) oscillator. This is well known from electronic organs and has some consequences:

- There is a tone signal for each note and each octave.
- The tone signals are square waves.
- Tone signals of the same key but different octaves are phase-locked.

The correct wording for such a circuitry would be "Tone Generator", but on the *Trilogy/Stratus* it is called "Oscillator", and there are *two* such oscillators (which can be controlled individually).



Oscillator 1 can be tuned by  $\pm 1$  semitone and Oscillator 2 by  $\pm 7$  semitones. The "Beating" that results from oscillator detuning is visualized via the PHASE LED.

The OCTAVE switch divides the oscillator clock by two and thus lowers the whole oscillator signal by one octave. When OCTAVE MOD is on, both OCTAVE switches are toggled by the LFO (see below) – a funny chip-tune effect.

SYNC is something special and has almost nothing in common with the (Hard) Sync effect known from other synthesizers. When activated, the divider circuitry of Oscillator 2 will be fed with the clock signal of Oscillator 1. Thus, the frequency of both oscillators is "in sync" i.e. exactly the same, with the exception that you still can put Oscillator 2 one octave down (via OCTAVE). SYNC is useful to avoid "Beating" but it has the nasty downside effect that the synchronized oscillators can randomly get out of phase which may result in a "thin" sound – or not<sup>2</sup>. That is why I added the SYNC PHASE tweak (see section *Tweak Section*).

---

2 Try it yourself: Hold a key and active/deactivate SYNC several times without releasing the key. The sound will vary in an unholy manner.

## Oscillator Glide

This section controls a “decay-only” envelope generator (realized by a CEM 3310 chip) that can modulate (“auto-bend”) the pitch of the oscillators.

AMOUNT sets the amount of the effect, SPEED the time until the final pitch is reached, and DIRECTION the bend mode:

- **A**: Bend both oscillators upwards.
- **B**: Bend Oscillator 2 only upwards.
- **C**: Bend Oscillator 2 only downwards.
- **D**: Bend both oscillators downwards.



Note that modes B and C will have no effect when SYNC is on! Why? Because Oscillator 2 is clocked by Oscillator 1 then. No “Hello Again”<sup>3</sup>... ☹

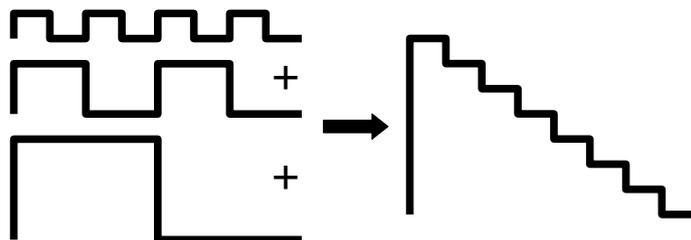
ENABLE is our first encounter with *Paralogy's* paraphony. Since there is only one Glide envelope for all voices (better: tone signals) you have to choose when the effect shall be triggered:

- **MONO**: Only for the first key pressed – Glide will not re-trigger until all keys have been released.
- **MULTI**: For each key pressed – this will re-Glide already pressed keys.

This functionality is more commonly known as “Single/Multiple Triggering”.

## Waveform Generation

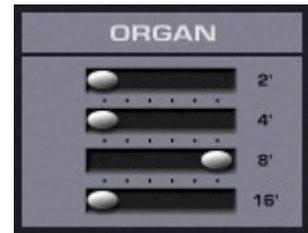
As stated above, the oscillators *per se* only generate square wave signals; sawtooth signals are approximated by adding the square tone signals of different octaves (16', 8', 4', 2') in a cunning manner. This results in the infamous “staircase” waveforms which are well-known from many other instruments of that era, too.



Interestingly, the “Square” signal of the *Trilogy/Stratus* (see section *Waveform Selection Section*) is constructed from the original 8' square tone signal plus a bit of the respective 16' signal (thus, it is not really a square).

## Organ Section

The *Organ* section is very simplistic and features four sliders to mix the 16', 8', 4' and 2' signals. These are smoothed by static lowpass filters and stem directly from the respective square tone signals of *both* Oscillator 1 and 2. Thus, you will get everything from a nice Chorus effect to total cacophony when detuning the oscillators.



## String Ensemble Section

The *String Ensemble* section is solely based on Oscillator 2: It receives the 16' and 8' "staircase" sawtooth signals and mixes them via the FOOTAGES knob. The resulting mix is fed into a classic Ensemble/Chorus effect consisting of three bucket brigade delays which are modulated by two LFOs. Using the TIMBRE knob you can mix the raw 16'/8' signal to the Ensemble signal, making the string sound less "swirly".

ATTACK and RELEASE control the respective timing of the *String Ensemble* section's envelope generator (again a single CEM 3310). And here we encounter another aspect of *Paralogy's* paraphony: The envelope is always *single-triggered* ("MONO" in *Trilogy/Stratus* terms) – the envelope starts with the first key pressed and ends with the last key released. If you play any note in between, it will have a gated envelope with a short release time (which can be adjusted in the *Tweak Section* section). Because of this it is a bit difficult to play a chord with long release since you have to release all keys at almost exactly the same time. But such a behavior is well-known from other string machines/synthesizer...



## Synthesizer Section

The *Synthesizer* section is based on the signals of both oscillators. We should get some decent synth sounds out of it, don't you think?

### Waveform Selection Section

Here you select the waveform: "(Staircase) Sawtooth", "Square" (which is a somehow impure square; see section *Waveform Generation*), or a mixture of both. Note that the oscillators will always have the same waveform!

Using the rather unique ALTERNATE SAW/SQ, it is possible to toggle between "Sawtooth" and "Square" waveforms when a new key is pressed. Time for another paraphonic encounter: The ENABLE switch selects if this should happen only for the first key ("MONO") or for each key ("MULTI") pressed – similar to the Glide envelope (see section *Oscillator Glide*).



### Envelope Generator Section

The *Synthesizer* section sports six individual *voices* with individual CEM 3310-based envelope generators. The parameters ATTACK, DECAY, SUSTAIN and RELEASE control the respective times/levels.

As already mentioned above in section *The Italian Job*, all the C and F# notes are assigned to voice 1, the C# and G notes to voice 2, the D and G# notes to voice 3, and so forth. Within a voice, each key re-triggers the envelope. Thus, if you hold an F#3 key and play a C1, the envelope will be re-triggered and applied to both notes. This can have interesting (and unexpected) results.



## Voltage Controlled Filter (VCF)

The CEM 3320 chip is a guarantee for a decent self-resonating four-pole lowpass filter, and there is one such VCF per voice. CUTOFF and RESON[ANCE] control the respective parameters, ENVELOPE sets the amount of cutoff frequency modulation (positive or negative).



Unfortunately there is no keyboard tracking parameter – this is due to the paraphonic nature of a voice itself. ☹

## Voltage Controlled Amplifier

This is an "invisible" section with no controls. The VCA of a voice is directly controlled by its envelope generator. In the *Trilogy/Stratus* a CEM 3330 chip is used.

## Volume/Pan Section

In this section, the outputs of the *Organ*, *Synthesizer* and *String Ensemble* sections are mixed together. *Paralogy* adds individual panorama sliders for the three sections. And due to **kraftraum**'s request (and my agreement) I took the liberty to add a Master Volume control – which is no part of the original hardware.



## Low Frequency Oscillator (LFO)

The LFO is routed to the oscillators, the VCF or the VCA. The waveform (WAVESHAPE) can be set to Triangle or Square. Once again we encounter paraphony with the RESET parameter which directly affects the LFO modulation amount (see next section).



## Modulation Section

This section controls the RATE and DEPTH (amount of modulation) of the LFO. DELAY sets the time it takes until the LFO modulation is applied after a key was pressed, while SLOPE controls the steepness of the onset of modulation.



And here comes the RESET parameter of the LFO into play: If RESET is set to "MONO", the delay phase will be started only by the first key pressed, while "MULTI" will restart the delay phase for each key pressed.

## Wheels Section

Both the *Trilogy* and the *Stratus* feature a joystick to control pitch bending and LFO modulation (similar to KORG keyboards of that era). I took the liberty to add a "Wheels section" instead where you can set the pitch bend amount and modulation wheel intensity received via MIDI.



## Phaser And Delay

The most common effects used along with string machines are Phasers and Delays (think *Oxygène* and *Equinoxe*...). I decided to rip those two effects from my *Oxid* plugin and add them to *Paralogy* for your convenience.



The Phaser is loosely based on the famous Electro Harmonix *Small Stone* with its 4-stage allpass filter design. SPEED sets the modulation rate and FEEDBACK the positive or negative feedback level – extreme settings can make your ears tingle. The Phaser works in stereo and can be activated for each of the three sound sections individually.

The Delay is modeled after the classic bucket brigade devices (BBD) that popped up in the early 70s. It works in stereo on the output of the Phaser section.

You can set the delay time from 75ms to 600ms or, when the button DOUBLE is active, from 150ms to 1200ms. FEEDBACK controls the amount of the delay output

signal that is fed back into the input, and MIX controls the balance between the original and the effect signal.

SYNC enables the synchronization of the delay time to the host's tempo. VIBE adds some slight modulation of the delay time to produce bright stereo effects, and P/P activates the Ping-Pong mode where the output of the left delay channel is fed into the input of the right channel and vice versa.

## Tweak Section

The tweak parameters are accessible when you click the yellow "high voltage" symbol on the lower left of *Paralogy's* panel.



*Trilogy* and *Stratus* do not feature a *common* knob to tune *both* oscillators – which is a bit annoying. Thus I added the [Master] TUNE parameter here.

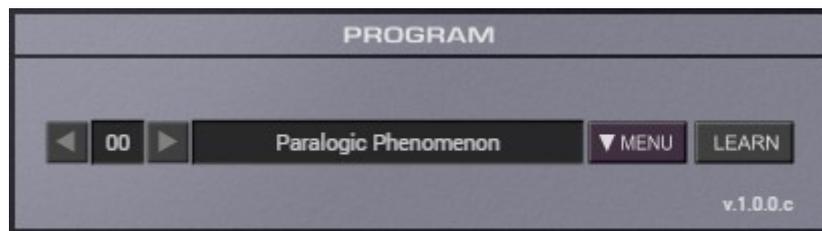
The next problem has already been discussed in the *Oscillator Section*: When SYNC is activated, the oscillators get randomly out of phase. The tweak activated parameter SYNC PHASE will set Oscillator 2 to a definite phase with respect to Oscillator 1.

KEY REL[EASE] sets the release time for a single key (see *String Ensemble Section*). Due to the nature of *Trilogy's* and *Stratus's* tone generation system, a tone signal coming from the frequency dividers is *gated* by its respective key on the keyboard – it does not have its own envelope generator! The only thing that can be set for the gate is the release time, and on *Trilogy/Stratus* this is hard-wired (roughly about 10 milliseconds). But *Paralogy* allows you to manipulate it. 😊

ENS. RATE and ENS. DEPTH set the rate and modulation depth of the two LFOs of the *String Ensemble* effect while DELAY VIBE does the same for the LFO used to create the Delay "Vibe".

## Program Section

All settings of *Paralogy* can be saved as a named program. You can also select one of the 64 programs via the arrow buttons or by clicking on the program number.



## Options Menu

When clicking the MENU button in the Program section, a context menu opens with the following options:

|                                       |  |
|---------------------------------------|--|
| <b>Copy Program</b>                   | Copy current program to internal clipboard   |
| <b>Paste Program</b>                  | Paste internal clipboard to current program  |
| <b>Init Program</b>                   | Initialize the current program   |
| <b>Load Program</b>                   | Load a program file containing a patch to <i>Paralogy's</i> current program  |
| <b>Save Program</b>                   | Save <i>Paralogy's</i> current program to a program file   |
| <b>Load Bank</b>                      | Load a bank file containing 64 patches into <i>Paralogy</i>  |
| <b>Save Bank</b>                      | Save <i>Paralogy's</i> 64 patches to a bank file   |
| <b>Select Startup Bank</b>            | Select the bank file that should always be loaded when <i>Paralogy</i> is started  |
| <b>Load Startup Bank</b>              | Load the Startup bank file; can also be used to check what the current Startup bank is                                       |
| <b>Unselect Startup Bank</b>          | Unselect the current Startup bank  |
| <b>Default Path for Program Files</b> | Sets the default path for program and bank files   |
| <b>MIDI Thru</b>                      | Set globally if MIDI data sent to <i>Paralogy</i> should be sent through to its MIDI output (stored in configuration file)   |
| <b>Ignore Program Change</b>          | Set globally if MIDI Program Change data sent to <i>Paralogy</i> should be ignored (stored in configuration file)            |
| <b>Reload Configuration</b>           | Reload <i>Paralogy's</i> configuration file  |
| <b>Save Configuration</b>             | Save <i>Paralogy's</i> configuration file  |
| <b>Check Online for Update</b>        | When connected to the Internet, this function will check if a newer version of <i>Paralogy</i> is available at fullbucket.de |
| <b>Visit fullbucket.de</b>            | Open fullbucket.de in your standard browser  |

## The *paralogy.ini* Configuration File

*Paralogy* is able to read some settings from a configuration file (*paralogy.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 *Paralogy* can be controlled by MIDI controllers, or more precise: Each MIDI controller (except *Modulation Wheel* and *Sustain Pedal*) can control one of *Paralogy's* parameters. The mapping is defined in the *paralogy.ini* for example like this:

```
[MIDI Control]
CC7  = 0  # Pitch Bend Amount
CC70 = 25 # VCF Cutoff
CC71 = 26 # VCF Resonance
...
```

The syntax is straight forward:

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

Given the above example, controller 7 directly controls the *Master Volume* parameter, controller 70 the *VCF Cutoff* etc. As you can see, comments are introduced by the Pound sign (#); they are here just for description purposes and completely optional. Note that the *controller number* can run from 0 to 110, with the exception of 1 (*Modulation Wheel*) and 64 (*Sustain Pedal*); the latter two are simply ignored.

## MIDI Learn

The easiest way to assign MIDI controllers to *Paralogy* parameters is to use the *MIDI Learn* function. To activate MIDI Learn, click on the LEARN button and wiggle both the MIDI controller and the *Paralogy's* parameter that you want to link. 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.

# Parameters

## General

| parameter                | id | description                                |
|--------------------------|----|--|
| <i>Pitch Bend Amount</i> | 0  | Pitch Bend amount                          |
| <i>Mod. Wheel Amount</i> | 1  | Modulation Wheel amount                    |
| <i>Master Volume</i>     | 65 | Master Volume                              |
| <i>Synth: Volume</i>     | 2  | Volume of <i>Synthesizer</i> section       |
| <i>Organ: Volume</i>     | 3  | Volume of <i>Organ</i> section             |
| <i>String: Volume</i>    | 4  | Volume of <i>String Ensemble</i> section   |
| <i>Synth: Pan</i>        | 5  | Panorama of <i>Synthesizer</i> section     |
| <i>Organ: Pan</i>        | 6  | Panorama of <i>Organ</i> section           |
| <i>String: Pan</i>       | 7  | Panorama of <i>String Ensemble</i> section |

## Organ Section

| parameter         | id | description         |
|-------------------|----|---------------------|
| <i>Organ: 2'</i>  | 8  | Level of 2' signal  |
| <i>Organ: 4'</i>  | 9  | Level of 4' signal  |
| <i>Organ: 8'</i>  | 10 | Level of 8' signal  |
| <i>Organ: 16'</i> | 11 | Level of 16' signal |

## Oscillator/Glide Section

| parameter                | id | description                               |
|--------------------------|----|---|
| <i>Osc.1 Fine Tune</i>   | 12 | Tuning of Oscillator 1                    |
| <i>Osc.1 Octave</i>      | 13 | Octave of Oscillator 1 ("Off", "Off")     |
| <i>Osc.2 Coarse Tune</i> | 14 | Tuning of Oscillator 2                    |
| <i>Osc.2 Octave</i>      | 15 | Octave of Oscillator 2 ("Off", "Off")     |
| <i>Osc.: Sync</i>        | 16 | Oscillator synchronization ("Off", "Off") |
| <i>Osc.: Octave Mod.</i> | 17 | Octave modulation by LFO ("Off", "Off")   |
| <i>Glide: Amount</i>     | 18 | Glide amount                              |
| <i>Glide: Speed</i>      | 19 | Glide speed                               |
| <i>Glide: Enable</i>     | 20 | Glide re-trigger ("MONO" or "MULTI")      |
| <i>Glide: Direction</i>  | 21 | Glide mode ("A", "B", "C", "D")           |

## Synthesizer Section

| parameter                            | id | description                                       |
|--------------------------------------|----|---|
| <i>Waveform Selection: Waveform</i>  | 22 | Waveform ("Sawtooth", "Square", "Mix")            |
| <i>Waveform Selection: Alternate</i> | 23 | Alternating waveform on key press                 |
| <i>Waveform Selection: Enable</i>    | 24 | Alternate waveform re-trigger ("MONO" or "MULTI") |
| <i>VCF: Cutoff</i>                   | 25 | VCF cutoff frequency                              |
| <i>VCF: Resonance</i>                | 26 | VCF resonance                                     |
| <i>VCF: Envelope</i>                 | 27 | VCF envelope amount                               |
| <i>Envelope: Attack</i>              | 28 | Envelope Attack time                              |
| <i>Envelope: Decay</i>               | 29 | Envelope Decay time                               |
| <i>Envelope: Sustain</i>             | 30 | Envelope Sustain level                            |
| <i>Envelope: Release</i>             | 31 | Envelope Release time                             |

## LFO Section

| parameter             | id | description   |
|-----------------------|----|---|
| <i>LFO: Routing</i>   | 32 | Destination of LFO ( <i>FM</i> , <i>VCF</i> or <i>VCA</i> ) |
| <i>LFO: Reset</i>     | 33 | LFO reset/re-trigger ("MONO" or "MULTI")                    |
| <i>LFO: Waveshape</i> | 34 | LFO Waveform (Triangle or Square)                           |
| <i>LFO: Rate</i>      | 35 | Rate of the LFO   |
| <i>LFO: Slope</i>     | 36 | Steepness of LFO modulation onset                           |
| <i>LFO: Delay</i>     | 37 | Delay until LFO modulation starts                           |
| <i>LFO: Depth</i>     | 38 | Depth of LFO modulation                                     |

## String Ensemble Section

| parameter               | id | description                               |
|-------------------------|----|---|
| <i>String: Footages</i> | 39 | 16'/8' signal mix                         |
| <i>String: Timbre</i>   | 40 | Ensemble/raw signal mix                   |
| <i>String: Attack</i>   | 41 | Attack time of the <i>String</i> section  |
| <i>String: Release</i>  | 42 | Release time of the <i>String</i> section |

## Phaser Section

| parameter               | id | description   |
|-------------------------|----|---|
| <i>Phaser: Speed</i>    | 43 | Phaser speed  |
| <i>Phaser: Feedback</i> | 44 | Phaser feedback   |
| <i>Phaser: Synth</i>    | 45 | Activates Phaser for the <i>String</i> section          |
| <i>Phaser: Organ</i>    | 46 | Activates Phaser for the <i>Organ</i> section           |
| <i>Phaser: String</i>   | 47 | Activates Phaser for the <i>String Ensemble</i> section |

## Delay Section

| parameter               | id | description                           |
|-------------------------|----|---------------------------------------|
| <i>Delay: Time</i>      | 48 | Delay time                            |
| <i>Delay: Feedback</i>  | 49 | Delay feedback                        |
| <i>Delay: Mix</i>       | 50 | Delay dry/wet mix                     |
| <i>Delay: Double</i>    | 51 | Activates doubled delay time          |
| <i>Delay: Sync</i>      | 52 | Synchronizes delay time to host tempo |
| <i>Delay: Vibe</i>      | 53 | Activates the delay vibe modulation   |
| <i>Delay: Ping-Pong</i> | 54 | Activate the Ping-Pong mode           |

## Tweak Section

| parameter                   | id | description   |
|-----------------------------|----|---|
| <i>Master Tune</i>          | 55 | Master tune   |
| <i>Sync Reset</i>           | 56 | Activates the phase reset of Osc. 2 on SYNC             |
| <i>Sync Reset Phase</i>     | 57 | Phase of Osc. 2 on SYNC reset                           |
| <i>Key Release</i>          | 58 | Key Release time  |
| <i>Ensemble LFO 1 Speed</i> | 59 | Speed of LFO 1 of the <i>String Ensemble</i>            |
| <i>Ensemble LFO 2 Speed</i> | 60 | Speed of LFO 2 of the <i>String Ensemble</i>            |
| <i>Ensemble LFO 1 Depth</i> | 61 | Modulation depth of LFO 1 of the <i>String Ensemble</i> |
| <i>Ensemble LFO 2 Depth</i> | 62 | Modulation depth of LFO 2 of the <i>String Ensemble</i> |
| <i>Delay Vibe Speed</i>     | 63 | Modulation speed of the delay's Vibe effect             |
| <i>Delay Vibe Depth</i>     | 64 | Modulation depth of the delay's Vibe effect             |

## Frequently Asked Questions

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

Just copy the files `paralogy.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 *Paralogy* VST2 plug-in the next time you start it.

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

Just copy the file `paralogy64.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 *Paralogy* VST2 plug-in the next time you start it.

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

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

Just copy the files `paralogy.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 *Paralogy* VST3 plug-in the next time you start it.

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

Copy the file `paralogy_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 *Paralogy* AAX plug-in the next time you start it.

### ***How do I install Paralogy (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 Paralogy?***

The ID is `P1 g y`.