PAPHE-800

User Manual

Edition: version v2022 (8.8.2022)

Firmware version: GliGli v2022 by imogen

Editor / author: imogen / image-et-son

Contents

1 Introduction to the GliGli Prophet-600 Firmware Upgrade **1**

1.1 What you can expect from the Firmware Upgrade 3

1.2 On compatibility for those upgrading from 2.0 and 2.1 RC3 5

1.3 Credits and license 6

2 Finding your way around your new Prophet-600 **8**

2.1 Data Pad and basic operation 9

2.2 Panel 13

2.3 Keyboard and performance section 14

2.4 Rear, inputs and output 15

2.5 The concept 17

3 Functionality **18**

3.1 Operating modes: preset and live mode 18

3.2 Loading and storing patches 20

3.3 Oscillators and mix 21

3.4 Volume and master tune 25

3.5 Filter . . . 25
3.6 Envelopes and routing . . . 26
3.7 Polyphonic, unison and chord mode, voice assigner priority . . . 29
3.8 LFO . . . 31
3.9 Poly-mod and oscillator sync . . . 33
3.10 Arpeggiator / sequencer clock and MIDI support . . . 39
3.11 Arpeggiator . . . 40
3.12 Sequencer . . . 42
3.13 Vibrato . . . 43
3.14 Pitch bender . . . 44
3.15 Modulation wheel and modulation delay . . . 45
3.16 Glide . . . 46
3.17 Detune . . . 46
3.18 Vintage / spread . . . 46
3.19 External CV in . . . 47
3.20 Velocity sensitivity . . . 47

4 Synthesizer setup and maintenance **48**

4.1 Tuning . . . 48

4.2 Switchable panel layout 49
4.3 MIDI integration 50
4.4 Voice deactivation 53
4.5 Transposition 53
4.6 VCF limit 54
4.7 Managing sound libraries via SysEx 54
4.8 Scaling adjustment 56
4.9 Firmware upgrade 56
4.10 Technical notes on vulnerable exponential FM 59

5 Reference **62**

5.1 Patch parameter overview 62
5.2 Default patch 64
5.3 Settings parameter overview 66
5.4 Operation modes 67

6 MIDI control implementation **71**

6.1 MIDI CC Patch Parameters 71
6.2 MIDI CC Settings 72
6.3 Special parameters and technical information 73

Introduction to the GliGli Prophet-600 Firmware Upgrade

The original Sequential Circuits Prophet-600 from around 1982 is run by a Zilog Z80 processor. This processor (and the firmware written for it) is pushed beyond its limits for computing and applying the voltages, for scanning user inputs and managing MIDI events. The necessary compromises made the instrument steppy, glitchy and non-responsive. That is a pity, because the basic synth voice hardware of the Prophet-600 is good: unlike DCO synths which became popular and affordable at the time, the Prophet-600 is a 6 voice analog synthesizer driven by 2 voltage controlled oscillators (VCOs) and a 24db low pass filter per voice, all implemented using the renowned Curtis chips [4]. The hardware also provides hard sync, frequency modulation and even modulation of cut-off by VCO. It is very versatile. The shortcomings of the original Prophet-600 were due only to the unfortunate circumstance that the selected and commonly available micro-processors at the time were not powerful enough for the Prophet-600. So the idea to make use of a more modern processor 30 years later was interesting. Around 2013 GliGli [2] started reverse engineering the original firmware for the Z80 and found that the Z80 can be replaced by a (more) modern Teensy++ 2.0 USB programmable microprocessor [6] (with some hardware modifications). This was the start of what is now know as the *Prophet-600 GliGli firmware upgrade*. GliGli describes his motivation as follows:

> *I love vintage analog synthesizers, and to be honest, my dream synth would be a Prophet-5, but when I heard what the Prophet-600 was capable of tone wise, I immediately thought its major weaknesses – the lousy computer part, software envelopes and LFO – could become its strength with a remake; basically the whole internal synth in voltage-controlled from a nice 14bit DAC, so with a fast modern micro controller, it could become awesome, maybe even better than a Prophet-5!*

There are various modifications and modern CPU ”implants” for some vintage analog synthesizers. Since the Teensy++ 2.0 board fits in to the same 2x20 pin standard socket of the Z80, the decisive advantage of the GliGli approach is that it is an easy-to-install and non-destructive firmware drop-in replacement[1]. It is completely safe to try out the Teensy++ 2.0 upgrade because one can always remove the board and restore the Z80 without technical expertise or training.

The Teensy++ 2.0 offers completely new possibilities. Just looking at specs: the Z80 version of the Prophet-600 ran at 4 MHz

[1]There are two versions of the Prophet-600. One has the Z80 on a socket. In this case replacement is truly ”drop-in”. The other version has the Z80 soldered to the circuit board. In this case the Z80 needs to be removed and a socket has to be soldered in its place first.

while the ATMEL processor on the Teensy board is operated at 16 MHz. Also, the on-board flash memory provides much better storage and firmware handling possibilities compared to the EPROM which the Z80 uses. The firmware does not rely on a battery for keeping patch data in memory. With the replacement the obvious limitations of the original Prophet-600 could be overcome - but more than that, it opened the possibility for new and for enhanced features.

Developing the software was a major effort with low-level functions and basic hardware communication at the core, e.g. instruction to set voltages to the electronic components, MIDI (interrupt) implementation, multiplexer operations and DAC operations. On top of this there is the ”musical” part of coding such as envelopes, LFOs, modulations and sequencing functions as well as UI functions. Many people have contributed, most probably out of curiosity and passion for synthesizers and driven by the excitement that such a project could not only be conjured up but also be implemented and brought to life.

The first stable version of the firmware upgrade was version 2.0 in 2014. An intermediate release (or at least something close to a release) version 2.1 Release Candidate 3 (2.1 RC3) has been available from September 5th 2015 on GliGli’s webpages [2]. The latest release version v2022 is the result of continued development between September 2021 and March 2022. It is meant as a celebration and revival of the original idea and it provides various new features and also significant improvements in core functions. The latest firmware version can be found on the project page[1].

While there is a realistic backlog of additional ideas to come in the future, there is a clear end of Prophet-600 upgrade story in sight because of ultimate hardware limitations. Firstly, PJRC (the manufacturer of the Teensy boards) has stopped producing the Teensy++ 2.0 board, so that it has become difficult to find them. Secondly, after extensive performance testing in the latest firmware developments efforts, it has become clear that the performance (smoothness and responsiveness of controls, timely and accurate processing of information) is finally limited by the built-in hardware, in particular the DAC (limited by voltage rise times) and multiplexer (limited by switching times).

If you own a Teensy++ 2.0 upgraded Prophet-600 you can still enjoy some exciting new developments. The following is a compact summary of the improved and the new features. The manual generally refers to the latest version v2022. Differences between versions are marked or commented in the text for your information. Share the passion for analog synthesizers with the Prophet-600 upgrade story!

imogen / image-et-son

1.1 What you can expect from the Firmware Upgrade

General features compared to the original Z80 version

- Greater resolution of many of the sound parameters with an improved refresh rate that makes the instrument much more responsive, again improved in v2022 using a dynamic scheme for reading user controls.
- Faster and smoother ADSR envelopes with support for two speed regimes (slow, fast) and two shapes (linear, exponential). The latest version v2022 has new linear shapes which were inspired by Prophet-5 rev 1/2 envelopes and designed to be more organic and musical than the former linear shapes.
- Positive and negative envelope amount settings for filter and poly-mod
- A new LFO function generator with a wider frequency range from one cycle every 20 seconds to about 60Hz
- Assignable, random and up/down arpeggiator, optionally sync-able to external MIDI and analog clock
- Multiple note assignment modes including last/low/high note priority
- Full MIDI In control including amplitude and filter velocity sensitivity with an external keyboard controller, continuous controllers (CC) of all sound parameters, program change (PC) to choose current preset
- A dedicated vibrato which can be controlled by the modulation wheel
- Unison chord mode
- Four new LFO waveforms in addition to the original triangle and square including sine, random stepped, noise (like on the original Prophet-5, but non-periodic) and sawtooth (ramp up)
- Mix overdrive which now allows the output from both oscillators to drive the mix VCAs A and B harder as well as the Curtis 4 pole filter resulting in new sonic possibilities
- Pitch Wheel interval selection of plus/minus one octave, a whole tone, a minor third and a fifth
- Pitch Wheel reassignment to the VCF and Volume or off
- Modulation wheel intensity setting in four steps, with a smooth (exponential) action in the latest version v2022
- A new and improved tuning procedure
- Octave, chromatic and free oscillator course pitch control, with different options for oscillators A and B in the latest version v2022
- Debounce feature that prevents unintended re-triggering caused by the old keyboard

Specific features from version 2.1 RC3 onwards

- Polyphonic step sequencer
- Per note tuning
- Improved UI, introducing display of -50...50 range for bipolar dials and center deadband for better handling
- VCF limit option
- Maintenance mode / adjustment scaling
- Vintage (spread) function for voice variations in tune and envelopes, re-designed as a continuous parameter in version v2022 which can even be controlled by MIDI CC
- Support for pedal to sustain notes and to hold chords (in unison)
- On-the-fly transpose function for arpeggiator and sequencer

Features with version v2022

- Smoother and more responsive actions of all controls due to dynamic scanning scheme
- Softer (exponential) action of LFO control (frequency and amount) to provide more travel for common ”musical” value ranges
- Super accurate oscillator B fine tuning using a slow mid region
- Improved envelope shapes inspired by the Prophet-5 rev 1/2 shapes, replacing the linear shape in versions 2.0 and 2.1 RC3.
- Revised and more intuitive organisation of menu parameters in groups with corresponding ”cheat sheet”
- A new UI concept with pick-me-up behaviour, in order to avoid discontinuous value changes and to make the differences between panel and internal values transparent
- Switchable panel layout with choices: Mix / Glide (Sequential Circuits / SCI layout)and Volume A / Volume B (synthgraphics suported GliGli layout)
- Flexible envelope routing allowing the poly-mod to be modulated by either envelope generator
- Ability to sync the LFO to the arpeggiator and sequencer

- Better MIDI integration possibilities through new local off mode and MIDI input into sequencer and arpeggiator
- Improved and protected patch management via MIDI, supporting single patch dump and loading MIDI patch to active controls
- Changed arpeggiator patterns: omitting the top and bottom note repeat in the U/D arpeggiator pattern and removing note repeats in the random pattern
- Support for adjusting the external CV input amount
- Vibrato target VCA

In addition to this, several more technical / advanced changes were made such as the option to switch between ”round robin” and ”first” voice assign logic, the ability toggle a pulse width sync bug which has been introduced in the version 2.0 (and kept for compatibility in version 2.1 RC3), changes to remove the interference of internal and external (MIDI) pitch bend by adding the two (as in the original SCI firmware), stabilization of the external analog clock input.

1.2 On compatibility for those upgrading from 2.0 and 2.1 RC3

You can upgrade the firmware by SysEx or by directly loading the firmware onto the Teensy++ 2.0 board via USB (see section 4.9). When upgrading by SysEx, both patch data and settings are preserved. Backward compatibility is ensured, i.e. patches created in prior firmware version (including Z80) still work properly on new firmware versions. Note that as a general principle, before performing any upgrade the safekeeping of patches is not only best practice but also strongly recommended. The following contains some remarks which are relevant if you are upgrading to version v2022 from version 2.0 or 2.1 RC3.

Panel: If you have a synthgraphics overlay, there are some minor changes (also reflected in the figures in this manual). Firstly, the layout of the menu parameters has been changed - these are now grouped in themes and since there are more parameters, selecting a parameter requires three presses in some cases. So the ”cheat sheet” that some users have attached to the left of the number pad is not correct any more. Secondly (and less significant), what was formerly the ”amplitude envelope” is now referred to more generally as the ”2nd envelope”. There are additional envelope routing options so that this envelope has a more generalized role.

Vintage / Spread: With 2.1 RC3 a ”spread” switch was introduced as part of the global settings (not part of the patch). With version v2022 this has been removed from the settings and has been reintroduced as the patch parameter *vintage spread*. It serves the same purpose, but it can be controlled continuously and it is more randomized than the former spread switch for more realistic results.

Detune: The unison detune has undergone a scaling re-design in version v2022: the detune is now less pronounced at higher frequencies. This has been done to achieve the same subjective detuning effect across the whole frequency range. This is a change to the sound of the instrument and depending on how you use detune, your detuned unison patches may sound slightly different at higher registers compared to prior firmware versions. The idea is of course, that this is an agreeable change for everybody.

Arpeggiator: The arpeggiator patterns up/down and random have been changed in version v2022

MIDI CC: Due to the change in menu parameters (obsolete parameters) and subsequent reshuffling (re-designation of the CC values for the legacy LFO frequency range), the specification of MIDI CC has changed slightly. As a result of rescaling of parameters the MIDI CC value to action has changed for LFO and vibrato frequencies and LFO and vibrato amounts as well as decay and release times of envelopes with linear shape. When loading older patches, the changes are taken into account to ensure compatibility.

Envelopes: The linear envelope shape has been revised in version v2022. It is now a linear shape with an exponential tail (i.e. a softer roll-off) for decay and release. It was a deliberate decision to replace the linear shape altogether. Note that the smooth roll-off required a rescaling of the envelope decay and release controls for linear shapes. When loading older patches, the changes are taken into account to ensure compatibility: the effective decay and release times are preserved. However, the application of MIDI CC values is different, so if you are controlling sound by MIDI CC you may need to make adjustments.

1.3 Credits and license

The firmware upgrade project is an open source project. The GliGli code base (up to and including version 2.1 RC3) can be found in [2] and the new code base of the latest development can be found in [3]. When I first drafted this manual I hoped to put together a credits section by assembling a fair and comprehensive list of contributors. I have given up. There was no one who could, with authority, list and assess what came from where and who added what. From my research I can say that a **lot** of people have contributed and some have obviously spent considerable time to make this software as useful and consistent as possible.

Let me just give you some highlights and a brief roundup. First, I think that the original 77 page technical documentation of the Prophet-600 by Stanley Jungleib [8] is great and such documentation is both helpful and more than one can expect. Then there is an elaborate blog "Prophet-600 Spirit" by *minisystem* [11] which is legacy by now but still available. It contains a lot more technical information, Prophet-600 hardware related ground work, I would call it. I had the impression that it also helped the project. The contributors to the actual code can be obtained from the submit history of the code base on Github [3]. Apart from

GliGli (obviously) I found *snickell* and *Ricard Wanderlof* with many commits, although I haven't taken the time to disseminate their contributions in detail. Then there is the community, especially (but certainly not exclusively) on Gearspace (a common community platform for people to discuss music gear, not so long ago also known as Gearslutz): testing, giving feedback, spreading the word, supporting the project by appreciation and enthusiasm. Since I started picking up the development in November 2021 I joined up with a group of supporters and musicians loosely bound together by a forum thread on Gearspace. A number of active users have been especially helpful in prioritizing features, testing and proof reading. I neither want to list all of them, nor single someone out. Let me just say that developing the new envelopes and experimenting with different and sometimes really tricky aspects of the Prophet-600 hardware (e.g. multiplexers) would have been a lot less fun (and in parts even impossible) without an active community and dedicated co-workers.

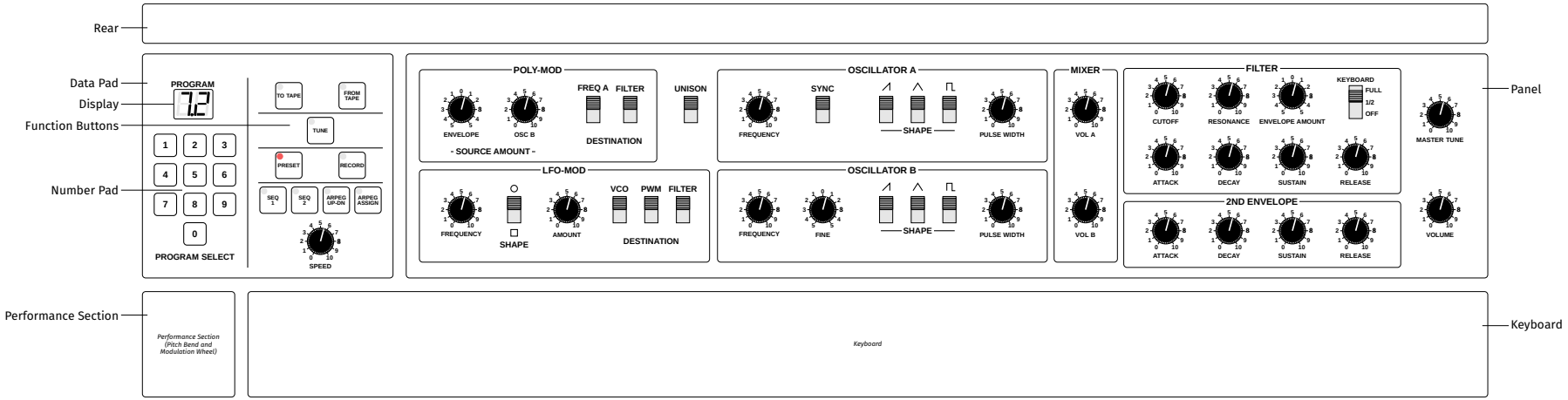
This user manual contains some texts and snippets written by other people. I compiled, merged, extended and in some case corrected the available information to an extent that makes it near impossible to trace everything back to its origins, let alone put in proper references. I just assume that all consent to the usage of their words. The overwhelming part of the text and the graphics are new.

The firmware is under GPL v3 license, except files / libraries / other open source components that have their own license in the header. The usage of the software and related documentation (including descriptions and references to descriptions for making modifications to hardware) are understood to be at the user's own risk. The firmware program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

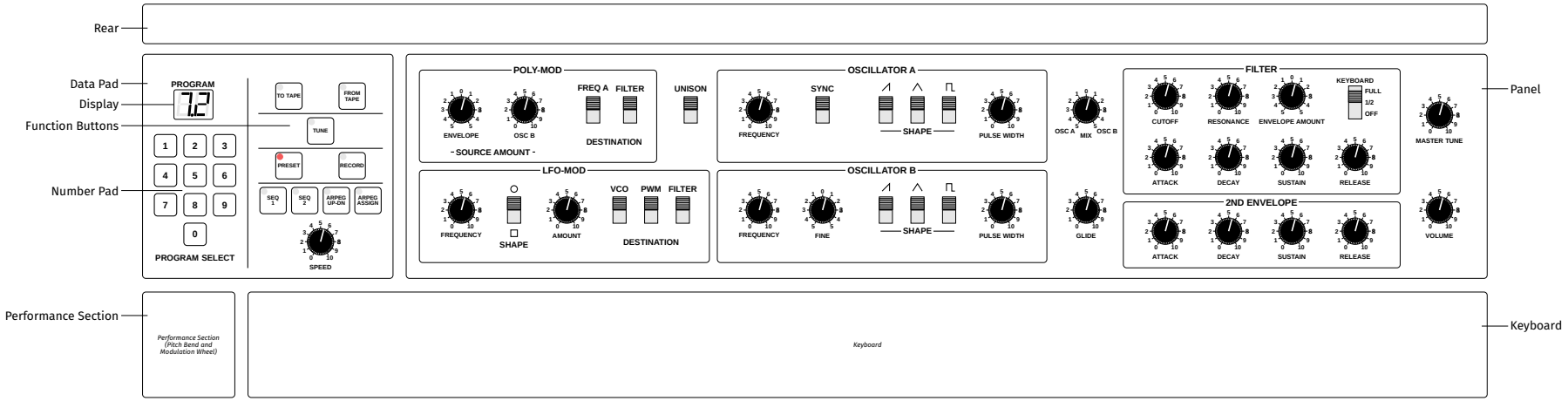
Finding your way around your new Prophet-600

The different controls and interfaces of the Prophet-600 are shown schematically below. Throughout the document the corresponding nomenclature is used. In the original GliGli firmware upgrade the dials for oscillator A vs. B mix had been replaced by separate dials for oscillator A volume and oscillator B volume, at the expense of demoting the glide amount to a menu parameter. In the newest firmware version you have the choice to either use the *GliGli* layout or the original *Sequential Circuits (SCI)* layout. For more details on how to change the panel and on the parameters in each case see section 4.2.

GliGli panel layout (like synthgraphics overlay)



Sequential Circuits panel layout



The following sections contain a brief explanation of the controls and interfaces: the **Rear**, **Data Pad**, **Panel** (with sub-panels), **Performance Section** and **Keyboard**.

2.1 Data Pad and basic operation

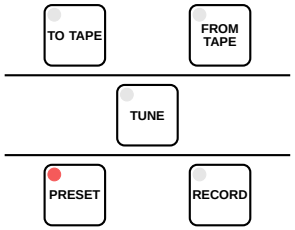
The **Data Pad** consists of the **Display** (2 two digit 7 segment display with one dot[1]), the **Number Pad**, the **Function Buttons** and the **Data/Speed Dial**.

Function Buttons

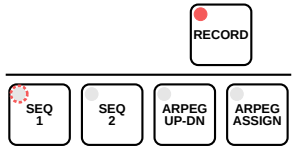
The block of 9 **Function Buttons** is the main control from where you activate and toggle the operation modes and functions of the Prophet-600. This block consists of an upper part (controlling essential functions) and a block of buttons relevant for the sequencer and arpeggiator, as shown below. (The **Record** button has a function of both.)

[1]The were good ideas for using a second display dot to make the UI more sophisticated, but - unfortunately - even though the display has the second dot, the Prophet-600 hardware supports control of only one dot.

Upper block of Function Buttons:



Buttons relevant for sequencer and arpeggiator:



The **Function Buttons** have LEDs which can be off, be on solid and on blinking. In the entire document the following convention for representing the button LED status off / on / blinking is used.

LED Status:  = LED on solid  = LED on blinking  = LED off

The different functions of the buttons are described in the general modes description in section 3.1, and throughout the chapter 3 in the context of specific functions in which they are used.

Number Pad

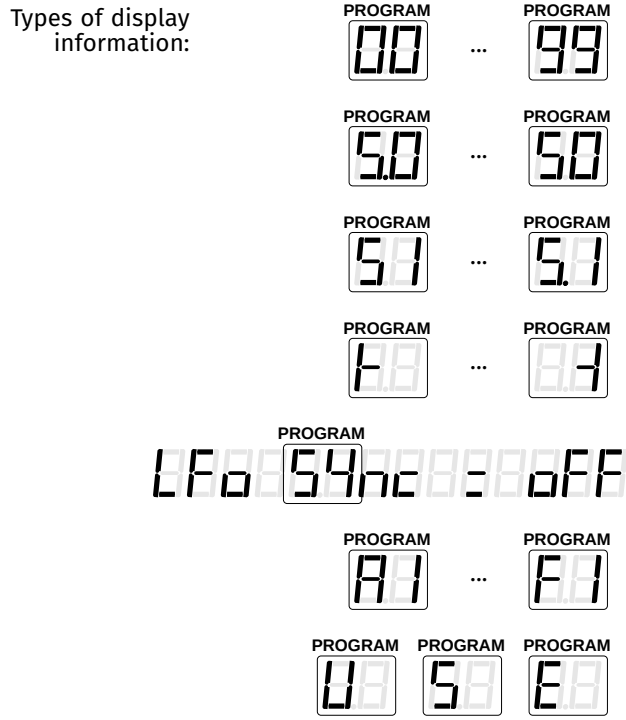
This pad has three usages depending on mode and selected functions:

- Selecting a patch number from one of the internal 100 patch storage slots to load in **Preset Patch Mode** (see section 3.1)
- Selecting a patch number for storing a patch to one of the internal 100 patch storage slots in **Storage Mode** (see section 3.2)
- Selecting a patch number to export a patch as MIDI SysEx in **Patch Management Mode** (see section 4.7)
- Selecting a function / setting in **Shift Mode** and **Shift Lock Mode** (see miscellaneous settings below and overview in section 5.3)
- Selecting a parameter in the additional patch parameter menu (see section 2.1) in **Preset Panel Mode** and in **Live Mode**

Display

The 2 digit 7 segment display is used for different purposes depending on the mode and selected function as shown below. There are three distinct types of information: display of numeric values (parameters values), scrolling of (short) texts, indicator symbols for pick-me-up in Preset Panel mode, etc.

Types of display information:



Value display of positive parameters with a range 0...MAX (**Live Mode**, **Preset Panel Mode**)

Value display of bipolar parameters with a range -MAX...MAX. The dot indicates negative values. (**Live Mode**, **Preset Panel Mode**)

Display of patch number with a range 0...99. In **Preset Patch Mode** the dot indicates that the patch has been changed compared to the stored patch.

Display of pick-me-up indicators in **Preset Panel Mode** mode for dials. The left/right symbol indicates that the current dial position is lower/higher than current active patch parameter value.

Longer messages are scrolled through the display, e.g. parameter names, message, etc.

Status of tuning procedure from oscillator A (1..6) through oscillator B (1...6) to filter F (1..6), see section 4.1 on tuning

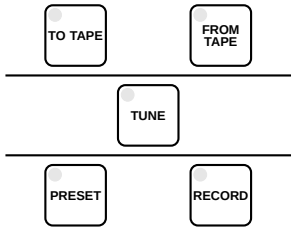
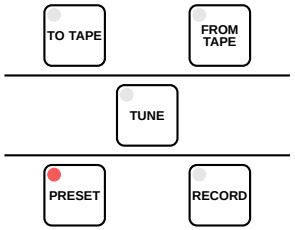
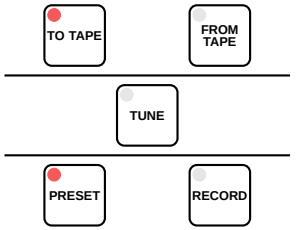
Status in SysEx firmware update mode: ready for upload, success, error.

Note that while numeric values are shown in the range -50 to 50 or 0 to 99, the internally used accuracy and dial sensitivity is much higher. As a result of this, noticeable patch changes can be achieved by turning a dial slightly while the display continues to show the same value. The display is a rough indication of the current values. However, it can be used accurately when centering bipolar dials.

Preset panel mode, live mode and additional patch parameters menu

The Prophet-600 operates in two modes, **Preset Mode** and **Live Mode**, see section 3.1. The mode of operation determines what the display shows and what functions are invoked when you press the number pad buttons, as shown in the table below.

Basic modes:

	Live Mode	**Preset Patch Mode**	**Preset Panel Mode**
LED			
Display	Parameter values	Patch number	Parameter values
Number pad	Select add. parameter	Select patch	Select add. parameter
Enter / leave	From **Preset** using **Preset** button	From **Live Mode** using **Preset**, from **Preset Panel Mode** mode using **To Tape**	From **Preset Patch Mode** mode using **To Tape**

The following section describes how patch parameters are accessed and modified in live mode and in **Preset Panel Mode**. Patch parameters are - by definition - stored, exported and loaded per patch. Most patch parameters can be controlled using dials and switches on the panel (see section 2.2). With the firmware upgrade the Prophet-600 provides a range of **Additional Patch Parameters**. There are *continuous* parameters (numeric) and *stepped* parameters (select options). Changing these parameters can only be done in **Preset Panel Mode** or in **Live Mode**. You must select the parameter first and then change the value using the **Data/Speed Dial**. A parameter is selected by pressing the buttons 0...9 on the **Number Pad** (or pressing them twice or in some cases three times). If you select a new parameter, the respective parameter name is scrolled through the display. The display then shows the current value. Numeric values are shown permanently and choice settings are scrolled through the display.

Each parameter is explained in the respective functional context in the next chapter. For a quick reference of additional patch parameters please see section 5.1.

Note 1: The parameter accessed using the number 0 is an exception. The parameter **(0) Arpeggiator/Sequencer Speed** controls the internal sequencer and arpeggiator speed or external clock divide and this speed is stored as an instrument setting (see

next section) and not as a patch parameter. For details on **(0) Arpeggiator/Sequencer Speed** see 3.10.

Shift mode and miscellaneous settings

The **Miscellaneous Settings** are patch independent instrument settings and functions. These settings are stored and re-loaded after a power cycle. The miscellaneous settings are accessed via the **Number Pad** in **Shift Mode**. To activate this mode either hold down the button **From Tape** while selecting one of the additional functions or double press **From Tape** to enter **Shift Lock Mode** (permanent shift) indicated by a blinking **From Tape** LED. To exit the mode push the button again.

In **Shift Mode** the numbers 0...9 on the **Number Pad** as well as some of the function buttons let you access the **Miscellaneous Settings**. The settings are edited by pressing the respective buttons repeatedly to cycle through values or to initiate actions. The first time a button is pressed the display scrolls the current value or mode. Pressing the same number key again cycles through the values of that setting or confirms a certain action or mode change.

Some important functions which you access in **Shift Mode** and **Shift Lock Mode** are ...

- ... the numbers 0...9 access MIDI setup, voice (de-)activation, panel layout, pitch bender calibration, panel layout switch, etc.
- ... the keys on the keyboard are used to set the keyboard transposition (see section 4.5)
- ... the **Tune** button activates the **Per Note Tuning Mode** (see section 4.1).
- ... pressing the **Record** button activates the **Patch Management Mode** (see section 4.7)
- ... pressing the **Preset** (and confirmation) loads the default patch (see section 5.2)

The list of miscellaneous settings can be found in section 5.3. Each setting is explained in the respective functional context in the chapter 3 and in the chapter 4 on instrument setup tasks.

2.2 Panel

Two slightly different panel layouts are supported. In *SCI* layout the panel is organized in 6 sub-panels, 3 additional patch controls (**Unison**, **Mixer** and **Glide Time**) plus two global controls, **(Master) Tune** and **(Master) Volume**. In *GliGli* layout the panel

is organized in 7 sub-panels (an additional mixer sub-panel), the patch parameter **Unison** and the two global controls **(Master) Tune** and **(Master) Volume**.

- Poly-Mod sub-panel (see section 3.9)
- LFO sub-panel (see section 3.8)
- Oscillator A sub-panel (see section 3.3)
- Oscillator B sub-panel (see section 3.3)
- Mixer sub-panel (in GliGli panel layout only, see section 3.3).
- Filter sub-panel (see section 3.5 and 3.6)
- 2nd Envelope sub-panel (see section 3.6)
- Unison switch (see section 3.7)
- Mix dial (in Sequential Circuits panel layout only, see section 3.3)
- Glide dial (in Sequential Circuits panel layout only, see section 3.16)

These controls are all patch parameters, which means they are stored, loaded, imported and exported with a patch. The panel controls **(Master) Tune** and **(Master) Volume** to the very right of the panel are not part of the patch. These controls are always applied as they are. They are meant to adjust the instrument to the performance environment (e.g. spontaneous tuning to a band/ensemble and adjustment of the volume). Also, you can always reliably start the instrument with volume turned to zero.

2.3 Keyboard and performance section

Without modifications, the Prophet-600 comes with a non-weighted, 5 octave (61 key) keyboard. The keyboard does not support touch sensitivity (although the upgraded Prophet-600 supports velocity sensitivity for amplitude and filter via MIDI, see section 3.20). Apart from playing, the keyboard is used for arpeggiator and sequencer input (see section 3.12) and for transposition (see section 4.5).

The Prophet-600 offers a **Pitch Bender** (unfortunately without a spring mechanism) and a **Modulation Wheel**. The mid position of **Pitch Bender** can be calibrated and there are several different bend targets to choose from, see section section 3.14. You can adjust the effect strength or action profile of the modulation wheel, and you choose if it should modulate the LFO or the vibrato, see section 3.15.

2.4 Rear, inputs and output

Power Switch

When turned on, the Prophet-600 starts in normal mode if no buttons are held down. The Prophet-600 can be powered up in two different modes related to maintenance tasks: firmware upgrade mode (see section 4.9) and scaling adjustment mode (see section 4.8).

When you start the instrument in normal mode the specification of the installed version is scrolled through the display. After power-up the instrument remembers that last state. It will starts in **Live Mode** or **Preset Mode** and in **Preset Mode** it activates and loads the last selected patch.

Note: earlier versions of the firmware upgrade went into tuning mode directly when powered up. From version 2.1 RC3 on this is no longer the case. In practice, the instrument needs to warm up and stabilize before tuning makes sense. For details on tuning see section 4.1. If the tuning data is invalid or there is no tuning data (for example after a firmware upgrade using USB), the Prophet-600 goes into tuning mode on start-up.

Line voltage selector

The selector next to the main plug provides the option to switch between 110V and 220V mains voltage, whichever is appropriate for the line voltage where the instrument is operated. Use a flat head screw driver to rotate the line voltage selector to the appropriate setting. Instrument should be powered down and unplugged when line voltage selector is adjusted.

Fuse holder

Fit the fuse according to the selected line voltage, i.e. 110V 1/2A, slo-blo, 220V 1/4A slo-blo.

Audio out / headphone jack

To drive a preamp or an amp, a standard monophonic cable can be used. Or if there are two audio destinations (e.g. one to

remain “dry” while the other is processed), a stereo cable may be more convenient.

The Prophet-600 has a monophonic output signal, but the jack is wired so that both sides of standard stereo headphones can be driven. The headphones have a minimum impedance of 1200 Ohms per element (600 Ohms in parallel).

Filter CV In

This jack accepts a 0-10 V control voltage (CV) which is applied to the filter cut-off frequency (all voices). This enables remote and spontaneous increase (but not decrease) of the filter cut-off making the sound brighter. The CV can be usually provided by an accessory voltage pedal or an external LFO. The strength of the external CV influence is controlled by an additional patch parameter (see section 3.19). The default value (cf. default patch) of this parameter is zero.

Note that popular ”noise mods” for the Prophet-600 use the external CV In control as an auxiliary control voltage for the noise amount. For such modifications the patch parameter controls the noise level.

Foot switch

The foot switch jack allows for the use of a sustain pedal (only supported from version 2.1 RC3 onwards). The pedal should be contact open at rest.

Cassette In/Out (disabled with GliGli mod)

Cassette In/Out can be used for sending a pulsed clock to the Prophet-600 to sync to the sequencer and/or arpeggiator. However, in the upgraded version the functionality to receive or send patch data via **Cassette In/Out** has been omitted.

When you set the instrument to use external analog clock (using the setting **Clock Sync**), the **Data/Speed Dial** (when set to control **(0) Arpeggiator/Sequencer Speed**) selects the clock divider rather than a clock speed. See section 3.10 for details.

MIDI In/Out

The Prophet-600 is equipped with standard MIDI In and MIDI Out connections. The MIDI receive and send channels can be configured in the miscellaneous settings, see section 4.3 on MIDI integration.

2.5 The concept

With the clearly laid out panel controls the Prophet-600 is both, a hands-on performance instrument and a sound sculpting synthesizer at the same time. The upgrade preserves that basic character. The controls on the panel have largely the same function as in original.

However, the new functionalities of the upgrade required additional parameters and these had to be placed in new hidden menus. This has obvious disadvantages. Diving into menus interrupts the work flow. It is abstract and unsatisfying. It is only acceptable for parameters which are rarely used and certainly not suited for live tweaking.

Another important aspect is the preservation of the general user interface concept. In particular, for each function there should be a unique way to do it, not a mix of say, dials and menus. This is one reason why the upgrade also contains ”only” one LFO and two envelopes even though one might have added more with the new processor. This corresponds to the layout of the Prophet-600 and anything more would have meant at least a partial dive into menus. There are some notable exceptions, such as the LFO shape, which is a mix of flipping a switch on the control panel and choosing options in the additional patch parameter menu. In most cases an additional patch parameter either has completely separate function form what is available on the control panel (such as the vibrato, which is fully controlled by menu parameters) or it had the function to modify the panel control behavior (such as envelope ranges, etc.).

A useful work flow of the upgraded Prophet-600 for creating or editing patches could be as follows:

1. Pre-production: set basic overall patch parameters, such as envelope shapes and routing, LFO targets and shape
2. Production: use the panel controls to interactively design the sound
3. Post-production: add additional features, such as vibrato, bend modulation, spread, etc. for a final touch

Ideally, menu diving should be largely restricted to steps 1 and 3 so that uninterrupted hands-on and interactive work with the synthesizer is possible in step 2.

Functionality

3.1 Operating modes: preset and live mode

As with the original Prophet-600 firmware, there are two modes of operation, a **Preset Mode** and a **Live Mode**. You can change between the two by pressing the **Preset** button, see section 2.1.

Live mode vs. preset mode

The idea of the **Live Mode** (LED of preset button is off) is that the patch that is playing corresponds exactly to the parameters as currently set by all dials, switches and additional patch parameters. The synth produces the sound as set and seen. In **Preset Mode** (LED of preset button is on), in contrast, the parameter values as stored in the patch memory are applied. The synth produces the sound of the preset. Still, once you change a dial or a switch on the panel or an additional parameter , this change will take effect also in **Preset Mode**. In this case a ”mixed state” applies, partly patch (untouched controls) and partly panel controls (touched controls). You can then store the current sound in the same patch or a different patch (See section 3.2). The patch will be stored as heard.

Note that you also load a MIDI SysEx patch into the active patch parameters in **Preset Mode**, see section 4.7.

Preset patch vs. preset panel mode

In **Preset Mode** the default setup is **Preset Patch Mode** in which the **Number Pad** is used to select and load presets. However, in **Preset Mode** you can also decide to switch to **Preset Panel Mode** in which the **Number Pad** selects additional patch parameters similar to **Live Mode** and the display shows the value of the last touched panel control. To activate this mode from **Preset Patch Mode**, press **To Tape**, after which the LED is on. To leave the **Preset Panel Mode**, press **To Tape** again.

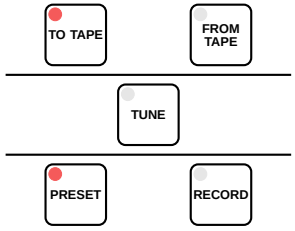
Pick-me-up mode in preset panel mode

The fact that the controls in **Preset Mode** do not automatically correspond to the current active parameter values has two disadvantages. Firstly, once a control is touched it is immediately applied, irrespective of where it is currently pointing and

how far away that value is from the one used in the patch you are hearing. This may lead to unwanted, disruptive sound changes. Secondly, it may be difficult or cumbersome to retrieve an original patch value once it is changed using a control. Therefore, as of version v2022 the upgraded Prophet-600 supports a *pick-me-up* control logic for dials in **Preset Panel Mode**. It works as follows:

If you touch and change a dial the display indicates if the current active value is below the current dial position (arrow on the left) or above it (arrows on the right). As long a these arrows are shown the value of the dial will not be applied, it is not ”picked up”. Once you touch the current active value closely enough, the display switches to showing the numerical value instead, indicating that the dial is ”picked up”. From that moment on all movements of the dial are always applied. If you move a dial fast enough you can hover over the pick-up point. The logic is shown below.

Preset Panel Mode example: modifying an active preset value = 57







Upper ”arrow” indicates that the dial value (ca. 40) is lower than the active internal value. The dial has no effect.



”picked up”: display shows the active value which is equal to the dial value. The active value follows the dial movement.



Lower ”arrow” indicates that the dial value (ca. 85) is higher than the active value. The dial has no effect.

The pick-me-up mechanism avoids discontinuous value changes and it provides a means to dial the current panel into the patch you are hearing. Note, however, that it is important to observe the display and current pick-up status of dials to avoid confusion. If you want a direct application of a dial press **To Tape** to switch back to standard **Preset Patch Mode** where all control changes are directly applied. The pick-up status of each control is remembered when switching between the two modes.

There is no pick-up mechanism for switches on the panel[1]. The mechanism is also irrelevant for additional patch parameters as there is no conflict between physical controls and internally applied values. Furthermore, the logic applies only to patch parameters. It is not applied to **(Master) Tune**, **(Master) Volume**, **Pitch Bender** and **Modulation Wheel**.

Note that the pick-up relies on that your dials can reach the maximum value. If your maximum value as shown on the display

[1]The attempt to do this was made, but surprisingly, the hardware is not reliable enough for this, in particular, the three-way tracking switch.

is less than 99 you may find that you cannot pick-up patch parameter values close to 99. In this case you should check the DAC gain to see if your instrument falls short of the required 4.9V which correspond to the maximum value. Please consult the service manual on this [8]. As a work around you can always switch to **Preset Patch Mode**, change the dial in question and then return to **Preset Panel Mode**.

Default Patch

Pressing the **Preset** button (with confirmation) in **Shift Mode** or **Shift Lock Mode** loads a default patch (see section 5.2). After loading the default patch the Prophet-600 is in **Preset Patch Mode**.

3.2 Loading and storing patches

Patches are stored in **Storage Mode**. To enter this mode from **Preset Mode** or **Live Mode** press the **Record** button, whose LED starts blinking. The Prophet-600 is then waiting for you to enter the (two-digit) target patch number on the **Number Pad**. Once the second digit is entered, the patch is stored automatically and the LED of the **Record** button deactivates.

Loading patches is done in **Preset Patch Mode** by selecting the patch on the **Number Pad**. Once selected and loaded the display then shows the selected patch number. If there is no valid patch stored at the selected page then no new patch is loaded and the Prophet-600 remains at the active patch. You can also re-load the current active patch, for example if you changed something and you would like to go back to the original.

For even easier and faster access, the patch selection can also be done using the **Data/Speed Dial**. If you are in **Preset Mode** and hold **From Tape** then **Data/Speed Dial** selects the patch. Notice that in contrast to the patch selection using the number pad, the page selected using the **Data/Speed Dial** will always ”load” even if there is no patch stored there. This means that the display shows the selected page number in any case, but the instrument remains silent if there is no patch stored there. To be precise, the Prophet-600 loads the default patch with no waveform activated in this case.

While the Prophet-600 waits for patch number entry (in **Storage Mode** or **Preset Patch Mode**) other functions of the **Number Pad** and display (e.g. in **Preset Panel Mode** or **Live Mode**) are temporarily suppressed. If you find you have accidentally started typing a digit in these modes while expecting to select a patch parameter you can switch / cancel the mode after the first digit by pressing **To Tape** in **Preset Patch Mode** or pressing **Record** again in **Storage Mode**.

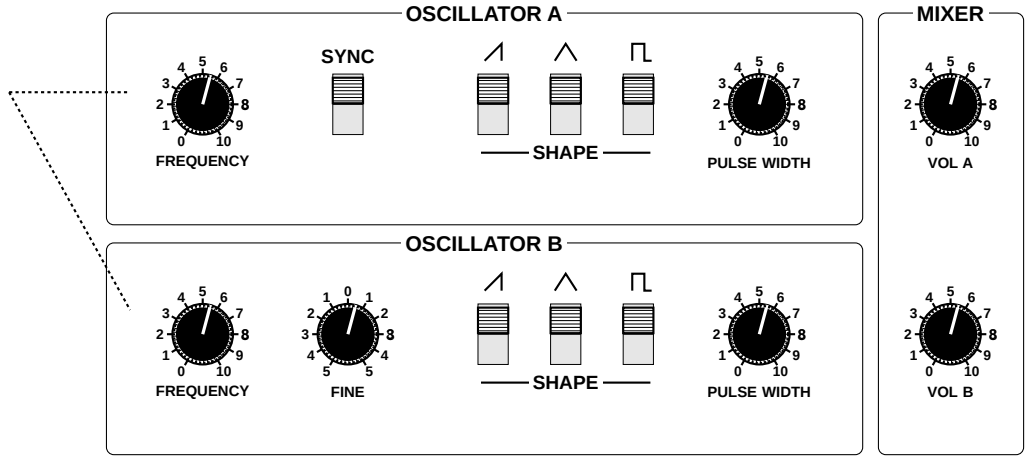
For exporting and importing patches via MIDI SysEx see section 4.7.

3.3 Oscillators and mix

Each of the six voices of the Prophet-600 has two voltage controlled oscillators (VCOs). VCO A and VCO B have dedicated sub-panels. The oscillators are basically identical. They offer three waveforms: *sawtooth*, *triangle* and *pulse*. The width of the pulse waveform can be adjusted using the corresponding **(OSC A/B) Pulse Width** dial. All three shapes can be activated simultaneously. The figures below are for *GliGli* panel layout and *SCI* panel layout, which differ in the way you control the oscillator volumes.

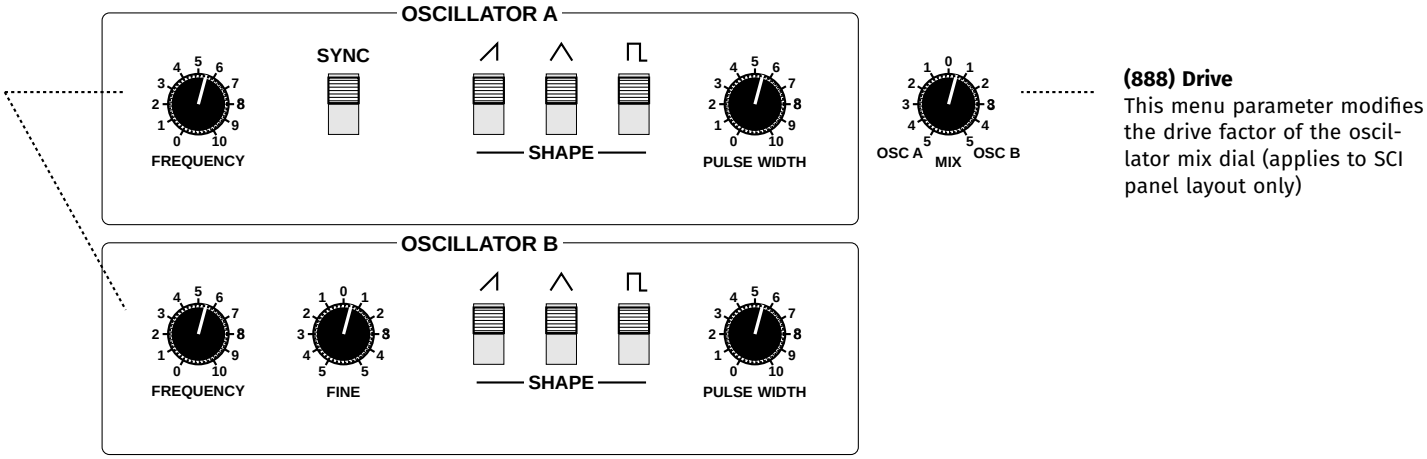
Oscillators in GliGli panel layout

(4) Oscillator Pitch Mode
This menu parameter oscillator sets the units of the two oscillator frequency dials: *free*, *semi*, *octave*, or mixed (different units for A and B.)



Oscillators in SCI panel layout

(4) Oscillator Pitch Mode
This menu parameter sets the units of the two oscillator frequency dials: *free*, *semi*, *octave*, or mixed (different units for A and B.)



(888) Drive
This menu parameter modifies the drive factor of the oscillator mix dial (applies to SCI panel layout only)

Oscillator controls

Both oscillators have a **(OSC A/B) Frequency** dial which determines the frequency range of the oscillators when played from the keyboard or via MIDI. Using the additional patch parameter **(4) Oscillator Pitch Mode** you can choose the granularity of the frequency dials:

- *free*: the base frequency can be chosen freely. The dial value selection is continuous and the **Display** shows approximate values between 0 and 99 in the normal way
- *semitone*: the base frequency can be selected in semitones. The dial value selection is discrete and the **Display** shows values from 0 to 63 (semitones).
- *octave*: the base frequency can be selected in octaves. The dial value selection is discrete and the **Display** shows values c0, c1, c2, c3, c4 or c5.

In addition there are also mixed options, *octave-semitone*, *semitone-octave*, *octave-free* and *free-octave*, which provide a different granularity for oscillators A and B. When the **(OSC A) Sync** switch is active, it may be more convenient to be able to choose the frequency range of oscillator A freely while oscillator B shold remain fixed on octaves. Depending on the patch design different choices of **(4) Oscillator Pitch Mode** may be more convenient.

The additional **(OSC B) Fine** dial for oscillator B is bipolar and provides frequency fine tuning with a range of about ± 1 semitone. The main purpose of separate frequency settings per oscillator is to be able to detune them against one another. This can be used to create thick and lively patches (when the pitch difference is small), to create complex timbres with harmonics (when pitch difference is large, typically octaves) or even create intervals (for example fifths). The frequency setting is also an important aspect of poly-mod, see section 3.9. To be able to tune the relative frequency very accurately, **(OSC B) Fine** has a very sensitive (slow changing) mid region between -1 and 1.

Oscillator A has an additional **(OSC A) Sync** switch. The hardware of the Prophet-600 offers hard sync of oscillator A to oscillator B: when **(OSC A) Sync** is activated, oscillator A is reset to the cycle start when oscillator B completes a cycle.

Finally, the outputs of both oscillators are mixed into one signal before entering the filter and output stages. The volume of oscillators A and B are either set by the **(OSC A) Volume** and **(OSC B) Volume** dials in GliGli panel layout or the **Mixer** and **(888) Drive** parameters in SCI panel layout as described in the following. For details on switching the panel layout, see section 4.2.

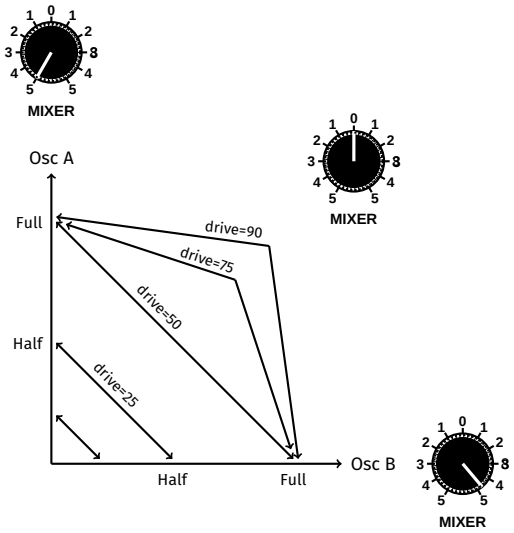
Oscillator mix in GliGli panel layout

In GliGli panel layout the volume of each oscillator has a dedicate control in the mixer sub-panel. The normalised output of either oscillator is about half way of the respective dial. Beyond this, the amplifiers into the filter can be over-driven. This is a new feature of the upgraded firmware from version 2.0 on.

Oscillator mix in SCI panel layout

In the *SCI* panel layout the two dials are designated **Mixer** (becoming a bipolar dial) and **Glide Time** and these control the oscillator mix and the glide time, respectively. For glide amount see section 3.16. In order to cover the entire range of oscillator volumes which the GliGli controls provide, there is an additional menu parameter **(888) Drive** which becomes available when the SCI panel layout is selected. The original Sequential Circuits mix function corresponds to a half-way value for **(888) Drive** at which oscillators A and B are mixed linearly. At **Mixer** =-50 oscillator A is at full volume and oscillator B is silent. At **Mixer** =+50 oscillator B is at full volume and oscillator A is silent. For smaller and larger values of **(888) Drive** the oscillator mix function is modified as shown below.

Oscillator mix function
In SCI panel layout the volumes of oscillator A and B are set by using the **Mixer** control and the menu parameter **(888) Drive**.



3.4 Volume and master tune

To the right of the panel there are two global controls, **(Master) Tune** and **(Master) Volume**. The master tuning has a range of approximately ± 1 semitone. Its purpose is to be able to tune the synth to other instruments without touching the internal, relative frequencies of the oscillators and thereby potentially changing the patch.

(Master) Volume and **(Master) Tune** are neither patch parameters nor settings. The two values are applied as set on the panel. The only exception is external MIDI control of volume (see section 6). When the master volume is set via MIDI CC that volume level continues to be applied until either a new value is set by MIDI or until the **(Master) Volume** control is touched/changed.

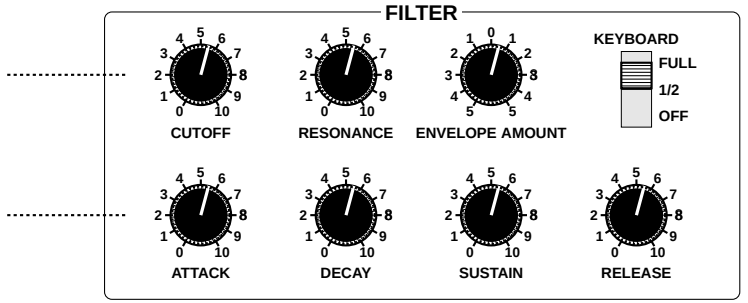
After power-up the instrument volume is always set to the value position of the **(Master) Volume** control.

3.5 Filter

The mix of the two VCOs is routed through a resonant 24db low pass filter (built on a Curtis CEM3372 filter). The control for the filter consists of a **(Filter) Cut-Off Frequency** dial regulating the cut-off frequency, and a **(Filter) Resonance** dial for the resonance. The **(Filter) Envelope Amount** dial regulates how strongly the cut-off frequency is modulated by the filter envelope. This dial is bipolar. It supports positive and negatives values (effectively an inverse envelope setting). The display will show values between -50 and 50 depending on position. This is major enhancement compared to the original Prophet-600 firmware which featured only positive values.

Filter Limit
Using the miscellaneous setting options the cut-off frequency can be limited to 22kHz, see section 4.6 for details.

(5) Filter Envelope Shape
This modifies the overall shape and speed range of the envelope. It influences the attack, decay and release phases, see 3.6 for details



The **(Filter) Keyboard Tracking** switch has the setting *Off*, *1/2* and *Full*. It lifts the cut-off frequency depending on keyboard note, i.e. the cut-off frequency is higher for higher notes. The effect applies to the cut-off frequency resulting from the fixed **(Filter) Cut-Off Frequency** setting plus any envelope and LFO modulation.

Equal tempered tuning of the filter cut-off frequency

In the Prophet-600 the filter cut-off frequency is tuned in the same manner as the oscillator frequencies. Furthermore **(Filter) Keyboard Tracking** is implemented in a way to allow equal tempered play using the filter:

- When **(Filter) Keyboard Tracking** is set to *half* then the filter cut-off is increased by a semitone every two semitones on the keyboard
- When **(Filter) Keyboard Tracking** is set to *full* then the filter cut-off is increased by a semitone with every semitone on the keyboard, i.e. if follows the scale

This feature essentially provides a third oscillator per voice. You can play harmonically using the filter at self-resonance even without oscillator A or B active - you should set the **(Filter) Envelope Amount** dial to zero, at least in first experiments. The **(Filter) Cut-Off Frequency** acts as a tuning dial in this case. You can also use the self-resonant filter in combination with oscillators A and B if you ”tune” the filter frequency appropriately. You can even use cross-modulation using the poly-mod function to modulate the filter cut-off using oscillator B (see section 3.9). However, there is no way to control the volume of the filter self-resonance relative to the volume of oscillators A and B.

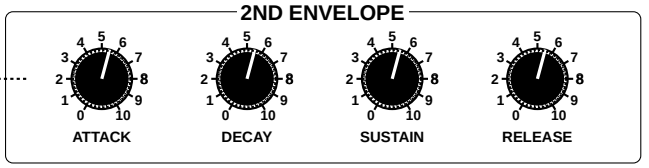
3.6 Envelopes and routing

Like the original version, the upgraded Prophet-600 offers two ADSR envelopes. The first one is the filter envelope, which modulates the filter cut-off frequency. Its parameters are located in the filter section of the panel, see section 3.5. The second envelope has a dedicated sub-panel below the filter section[2].

Both envelopes work identically in that they have a classic ADSR setup with an **(Envelope) Attack** control, a **(Envelope) Decay** control, a **(Envelope) Sustain** control and a **(Envelope) Release** control.

[2] In the original Prophet-600 as well as in the upgrade up to version 2.1 RC3 the 2nd envelope is a dedicated amplitude envelope and on the original panel and on the existing panel overlays the sub-panel is marked ”Amplifier”. With the envelope routing option in the latest firmware version the envelope and its sub-panel are more appropriately called ”2nd Envelope” in this document.

(55) 2nd Envelope Shape
This modifies the overall shape and speed range of the second envelope. It influences the attack, decay and release phases of the envelope



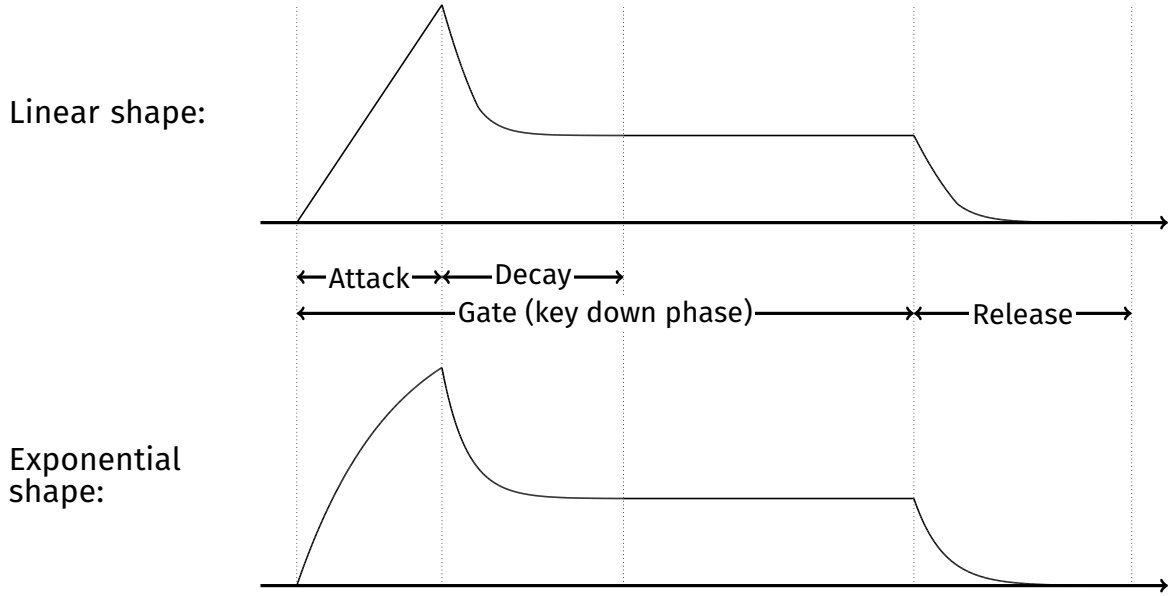
The upgraded Prophet 600 provides faster and smoother envelopes, with two different envelope time scales (*slow* and *fast*) and two envelope shapes (*linear* and *exponential*)

You can set the shape and speed range independently for the filter and the 2nd envelope using two additional patch parameters. The parameter **(55) 2nd Envelope Shape** changes the 2nd envelope and the parameter **(5) Filter Envelope Shape** changes the filter envelope. Each of the two parameters can take the following values:

- *slow -linear*
- *slow - exponential*
- *fast - linear*
- *fast - exponential*

The envelope shapes are shown below. The *linear* shape is strictly linear in the entire attack phase and in the main attenuation phases of decay and release, but it tails out exponentially. [3]

[3] Up to version 2.1 RC3 the linear envelope shape was strictly linear in the decay and release without a soft tail. The new shape sounds more organic as it does not drop hard to zero. It was inspired by the Prophet-5 rev 1/2 envelope behaviour.



The phase durations of the ADSR envelopes range between $< 30ms$ to $> 50s$.

Envelope modulation targets

The envelopes have three available targets: filter cut-off frequency, amplitude and poly-mod (for this see section 3.9). The filter envelope is always routed to the filter cut-off frequency and the modulation strength is controlled by the **(5) Filter Envelope Shape** control. For the amplitude and poly-mod targets, different routing options are provided as described in detail in the section on poly-mod 3.9. The additional patch parameter **(555) Envelope Routing** is used to configure which envelope modulates which target. The main benefit from the routing options is to make poly-mod independent from the filter dynamics. The default routing is that the poly-mod is modulated by the filter envelope and the amplitude is modulated by the 2nd envelope. This is the original routing and the routing of versions 2.0 and 2.1 RC3.

The modulation strength of the amplitude is fixed and positive in the Prophet-600. The modulation strength of the poly-mod is set by the bipolar **(Poly-Mod) Envelope** control as explained in section 3.9.

3.7 Polyphonic, unison and chord mode, voice assigner priority

You can play the upgraded Prophet-600 in the following voice modes

- *polyphonic*: This mode is set by switching **Unison** to *off*. In this mode any new note is assigned to one of the 6 voices. This is the default patch setting.
- *unison*: This mode is activated by switching **Unison** *on* without holding any keyboard key pressed. In this mode all 6 voices play the same note. Apply the additional patch parameter **(8) Unison Detune Amount** to create thickened sounds in unison mode, see section 3.17.
- *chord*: This mode is activated by switching **Unison** *on* while holding down one or more keys (the ”chord”). In this mode the synthesizer transposes that chord over the whole keyboard. Note that independently of how many notes the chord contains (and therefore how many voices would be left in principle), only one chord can be played at a time, similar to unison mode.

Unison is a patch parameter – i.e. it is stored with the patch.

Note that in unison/chord modes the foot switch does not hold/sustain notes as in polyphonic mode but instead latches patterns of notes / chords on-the-fly without the need to leave and reactivate the unison/chord mode. To use on-the-fly latching, hold down the chord you want to latch in unison/chord mode and press the foot switch. In order to return to normal unison mode, press the foot switch with no key held down. Using the foot switch with **Unison** on re-triggers the currently held notes.

The assignment of voices to newly played notes can be customized using the additional patch parameter **(7) Priority**. New notes will be assigned to voices according to the following priority rules:

- *last*: New notes always play and if 6 voices are already playing then the oldest played note is ”stolen”
- *low*: A newly played note only plays if it is lower than the highest note already playing. The highest note is ”stolen”. With **Unison** on, legato is active. This is the preferred setting when bass accompaniment is important and the sudden stop of bass notes would create unwanted disruptions.
- *high*: A newly played note only plays if it is higher than the lowest note already playing. The lowest note is ”stolen”. With **Unison** on, legato is active. This is the preferred setting for a synthesizer lead situation where the sudden stop of a high note would create unwanted disruptions.

The assigner priority of voice mode is a patch parameter – i.e. it is stored with the patch.

Note on the return of ”stolen” notes: when more than 6 notes are held in priority *last*, only the newest 6 notes play. If one of those keys is released then this note is cut and a stolen note is re-assigned and will sound, i.e. it will re-appear. The note is not re-triggered in this case but rather takes over from the envelope positions of the note that was just released and cut. The note to re-appear is the oldest of the stolen notes. However, with the re-assignment this note becomes the newest assigned note. When another key is release the next note to re-appear will therefore be the one which originally the second oldest, and so forth. Glide is applied to re-appearing notes.

Voice assignment logic

As described above the assigner priority determines which notes are played or terminated once there are more notes to play than the available 6 voices (or less if one or more are deactivated). A separate part of the assigner logic is concerned with the selection of the target voice when a new note is to be assigned. There are two choices for this logic which you can toggle using the menu parameter **(77) Assigner**:

- *first*: the first free voice (in the sense of the fixed internal numbering of voices from 1 to 6) is selected. I.e. when only one note is played only the first voice is used.
- *cycle*: also known as ”round robin”, this logic always selects a new voice in order to distribute the load evenly among the voices. I.e. when only one note is played each voice will be used in turn.

In practise the first logic this means that when a note is released and played again the same voice is used. Since voices can have different characteristics in analogue synthesizer such as the Prophet-600, this type of voice allocation logic favours more stable and consistent note sequences. Using the second logic, on the other hand, you can make deliberate use of the variations in voice characteristics to create more lively and animated sequences. This becomes very pronounced when vintage spread is activated (see section 3.18). Note also that glide is based on voice, i.e. once a note is assigned to a voice the glide start note will be the last note of that voice (rather than, for example, the last played note). Therefore the parameter **(77) Assigner** also has a strong impact on the behaviour of glide in polyphonic mode. For glide see section 3.16.

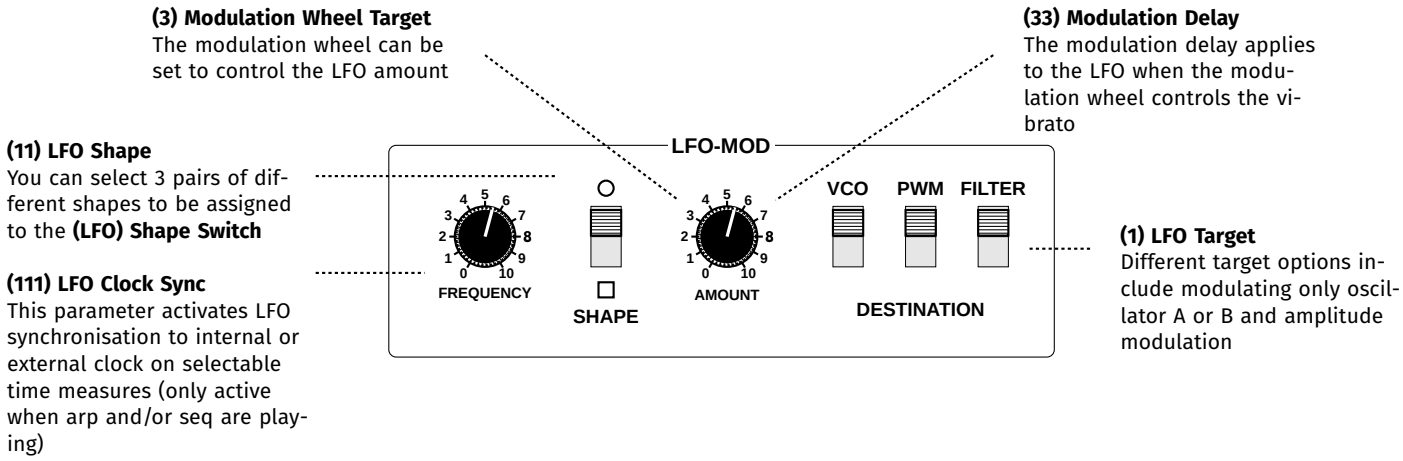
Ideas for solo voices

You can use the assigner priority effectively in combination with the voice modes to customize solo voices. For example combine the priority *high* (or *low*) with ...

- ...*unison* mode for a massive legato solo voice
- ...*chord* mode with a one key ”chord” (hold down only C3) to achieve a more delicate legato mono solo voice
- ...*chord* mode with octaves (hold down C3 plus C4 plus more if desired) for a more assertive mono solo voice

3.8 LFO

Like the original model, the upgraded Prophet-600 offers a single main LFO[4]. However, the functionality is substantially enhanced. The frequency range of the LFO is wider, from 1/20 to 60 Hz. There are additional waveforms (a total of 6). The upgraded Prophet-600 offers more modulation target options and, finally, the user has the option to either control the LFO amount using the modulation wheel or to set a modulation delay for the LFO. As a result the controls on the LFO sub-panel shown below are affected by 5 menu parameters. Still, the most important LFO controls can be changed hands-on using the dedicated LFO controls.



LFO speed

[4]There is also a vibrato, see section 3.13

The LFO speed / frequency is set in the panel using the **(LFO) Frequency** dial[5]

LFO modulation amount and targets

There are four modulation targets which can be active simultaneously. Three targets are activated using the switches **(LFO) VCO** (modulation of oscillator pitch), **(LFO) PWM** (modulation of pulse width) and **(LFO) Filter** (modulation of the filter cut-off frequency) in the LFO sub-panel. The modulation amount is set using the **(LFO) Initial Amount** control[6] and this amount applies to all activated modulation targets at fixed relative strength. The pitch modulation is harmonic and the maximum **(LFO) Initial Amount** corresponds to 30 semitones.

In the upgraded Prophet-600 you can customize the LFO target. The additional patch parameter **(1) LFO Target** offer the settings:

- *A & B*: Both oscillators are modulated, this is the standard of the original Prophet-600
- *A*: Only VCO A is modulated
- *B*: Only VCO B is modulated
- *VCA*: Amplitude modulation, and both oscillators are selected for all targets

The choices for oscillators A and B affect the targets *VCO* and *PWM* but obviously not *Filter* as there is only one filter per voice. When the target *VCA* is selected, both oscillators are targets for *VCO* and *PWM* modulation (if the respective switches are on).

LFO shapes

The upgraded Prophet-600 supports not only the waveform *Pulse* and *Triangle* of the original model but four additional shapes, *Sin*, *Sawtooth*, *Random* and *Noise*. On the sub-panel there is one simple up/down **(LFO) Shape Switch** switch. To accommodate 6 waveforms an additional patch parameter **(11) LFO Shape** is used. Selecting the desired waveform therefore requires selecting the respective shape pair and then using the **(LFO) Shape Switch** switch to toggle between the two shapes.

[5] Up to the upgrade version 2.1 RC3 a menu parameter was available to chose between slow and fast frequency ranges for the LFO. In the new version the range selection using the **(LFO) Frequency** control has been made exponential and the control range covers the entire frequency range. The range has also been extended further compared to 2.0 and 2.1 RC3 both on the low end and on the fast end.

[6] The action of the **(LFO) Initial Amount** dial has been made exponential and therefore with a slower onset in the lower value range. In particular when pitch modulation is active this provides a longer travel for commonly useful modulation amounts, typically up to 2-3 on the control scale.

(11) LFO Shape set to:	**(LFO) Shape Switch** *up*	**(LFO) Shape Switch** *down*
tri-pulse	LFO shape is triangle	LFO shape is square
sin-random	LFO shape is sin	LFO shape is random
saw-noise	LFO shape is sawtooth	LFO shape is noise

LFO delay and modulation of LFO

The strength LFO effect itself can be modulated by either the modulation wheel or a delay function. Using the menu parameter **(3) Modulation Wheel Target** your can choose *vibrato* or *LFO*. When set to *LFO* the modulation wheel controls the LFO amount. The LFO starts at initial amount as set by **(LFO) Initial Amount** on the sub-panel and turning the modulation wheel up adds LFO strength. The wheel action itself is set using the menu parameter **(333) Modulation Wheel Range**. If the modulation wheel target is set to *vibrato* then, automatically, the modulation delay is applied to the LFO, i.e. the LFO onset is delayed by the amount set by the menu parameter **(33) Modulation Delay**. For details on modulation wheel related parameters see section 3.15.

LFO re-triggering

As a new feature in version v2022, the re-triggering behavior of the LFO can be customized. The default is a free running LFO, but you can also set the LFO to re-trigger with new played keys (i.e. when no key is held and a new key is played) or to re-trigger depending on (external or internal) clock. The latter effectively syncs the LFO to the arpeggiator and/or sequencer. The menu parameter **(111) LFO Clock Sync** provides the choices *off* (free running LFO), *key* and *1*, *2*, *3*, *4*, *5*, *6*, *8* to have the LFO reset after 1, 2, 3, 4, 5, 6 or 8 beats or the arpeggiator and/or sequencer. The LFO clock sync is only active if arpeggiator and/or sequencer is running. For details on clock settings see 3.10.

3.9 Poly-mod and oscillator sync

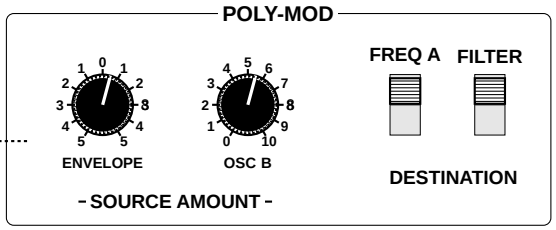
The functions associated with the poly-mod section are very powerful. They provide tools for non-linear modulation: cross-modulation of the two voice oscillators and modulation of the filter cut-off frequency by VCO B.

A familiar and common synthesizer feature is that oscillators and/or filters (frequency, filter cut-off frequency, pulse width, etc.) are modulated by a *low frequency oscillator (LFO)*. An LFO does not operate in the audible frequency range (it is ”low frequency”) and it has a fixed frequency which is independent from the note played. Poly-mod is different in that the modulation itself is in the audible range and the resulting non-linearity changes the frequency spectrum drastically. The resulting effect is

comparable to FM synthesis known from digital FM synthesizers[7], although in the Prophet-600 it is more raw and fully analog. Like the oscillator sync function, cross-modulation is hardware implemented in the Prophet-600. To be precise, the Prophet-600 implements *exponential* FM by feeding the audio signal directly into the voltage control of the oscillator.

The poly-mod control can be found in the poly-mod sub-panel.

(555) Envelope Routing
The routing options determine whether poly-mod is driven by the filter or the 2nd envelope



There are two modulation sources, **(Poly-Mod) Envelope** and **(Poly-Mod) Oscillator B** and two destinations, the pitch of oscillator A and the filter cut-off frequency. Each source is activated using the respective panel switch, **(Poly-Mod) Frequency Target** and **(Poly-Mod) Filter Target**. The **(Poly-Mod) Oscillator B** determines how strongly the oscillator B modulates the poly-mod targets (destinations). It is instructive to consider the effect of oscillator B modulation on the two targets first and then explore the effect of envelope modulation.

Cross-modulation

- When you set the **(Poly-Mod) Frequency Target** switch to *on*, oscillator B modulates the frequency of oscillator A (frequency modulation, FM). Depending on the modulation strength the result is a wide and rich frequency spectrum of oscillator A which contains among others the sum and difference of multiples of the two oscillators frequencies. The spectrum strongly depends on the frequency difference and may be totally non-harmonic but also harmonic for carefully adjusted frequency settings. Typical spectra are often described as bell sounding. Note that in contrast to digital FM synthesizers which use highly accurate sin waves, the oscillator output of the Prophet-600 does not have the same basic accuracy and the available waveforms already contain higher harmonics (triangle, sawtooth, etc.). Therefore, the FM result in the Prophet-600 is typically more drastic and dirtier and requires classic subtractive methods for shaping the sound, i.e. the

[7] FM synthesizers typically use phase modulation. However, the term *FM synthesis* is commonly used for the family of synthesis methods or modulations in which an audio range signal is used to modulate the frequency control.

application of the low pass filter. Another important difference to FM synthesizers is that the Prophet-600 (like most analog poly synthesizers with cross-modulation capability) uses exponential FM. It is characteristic of this feature that the modulation strength changes the pitch of the sound. If you want a harmonic patch for chord playing, you should treat **(Poly-Mod) Oscillator B** as a static parameter once it is adjusted.

- When you set the **(Poly-Mod) Filter Target** switch to *on*, oscillator B modulates the filter cut-off frequency of the low pass filter. Like oscillator FM this is a non-linear modulation. This effect can also produce a wide frequency spectrum which may or may not be harmonic, depending on the relative frequency of oscillators A and B. The effect is more pronounced at low filter cut-off frequencies. The resonance also has a strong influence on the effect.

Note that unless you use oscillator sync at the same time, poly-mod is very pitch sensitive so that patches which use strong poly-mod may need to be fine tuned using **(OSC B) Fine** or **(OSC A/B) Frequency** (for this, the *free* option of the parameter **(4) Oscillator Pitch Mode** is useful). Also, poly-mod patches may sound off on a cold unit due to temperature dependent drift and generally it maybe necessary to retune the synth more often when using extreme poly-mod patches.

You should also be aware that the harmonic integrity of a patch may vary from instrument to instrument and - more importantly - that there can be voice-to-voice differences. Therefore, medium to strong poly-mod patches that sound good on one instrument may not immediately sound good on another unit or may even not be fully reproducible. In most cases you can compensate by adjusting **(OSC A/B) Frequency** (better A than B), **(OSC B) Fine** and/or **(Poly-Mod) Oscillator B**. However, you may find that in your unit the differences among the voices are so strong that beyond a certain poly-mod strength, patches start sounding detuned (for example in chord play). Please refer to section 4.10 for more background on the design implications of poly-mod in the Prophet-600 and for measures you can take to reduce any detuning effects of poly-mod. Measures include scaling adjustment and the avoidance of the oscillator B pulse waveform in medium to strong poly-mod patches.

Note that when you use oscillator sync, pitch related problems of poly-mod are not an issue.

Envelope pitch modulation

The second poly-mod modulation source is the poly-mod envelope. When the **(Poly-Mod) Frequency Target** switch is *on*, the envelope modulates the pitch of oscillator A at the strength set by the **(Poly-Mod) Envelope** dial. The **(Poly-Mod) Envelope** control is bipolar. It provides negative (inverse modulation, lowering the pitch) and positive modulation (raising the pitch). The pitch shift is harmonic and in the sustain phase of the envelope the maximum obtainable pitch shift (i.e. maximum positive or negative value of **(Poly-Mod) Envelope** and sustain at full value) corresponds to ± 30 semitones.

Note that the poly-mod envelope only affects the frequency of oscillator A (and never the filter cut-off frequency) even if the second destination, the **(Poly-Mod) Filter Target** switch, is also *on*. The filter cut-off frequency is already modulated by the filter envelope. Depending on the menu parameter **(555) Envelope Routing** the source envelope is either the filter envelope or the 2nd envelope, see below for details on the options.

Pitch modulation by envelope is not actually a cross-modulation function. However, the association with poly-mod is useful because of the strong effect pitch (difference) has on cross-modulation. What can it be used for? Envelope modulation is a form of time modulation. It can change the sound from attack through hold to release. By time modulating the pitch in combination with cross-modulation (and/or sync), the sound can be made to sweep through different spectra, ranging from apocalyptic noise through bell sounds to delicate transients. The way humans perceive natural sounds is strongly influenced by the frequency characteristics of the transients (typically short lived high frequency components) in the attack and decay phases of the sound. This is best modelled using an envelope. So, being able to modulate the composition of frequency spectrum by envelope is a versatile tool for advanced sound design. This was probably the original thought behind putting the **(Poly-Mod) Envelope** control in the poly-mod section.

In order to further expand the sound design possibilities, the latest firmware upgrade offers an envelope routing option, which is inspired by a similar setup in the Roland JX8P synthesizer. Having different envelopes modulating the filter and the poly-mod opens up new possibilities in the sound and spectrum dynamics. The following section explains the routing options.

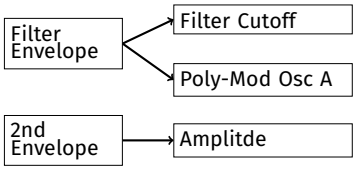
The menu parameter **(555) Envelope Routing** determines if the filter envelope or the 2nd envelope is used in poly-mod[8].

- *standard*: In this ”standard” routing the amplitude is modulated by the 2nd envelope and the poly-mod is modulated by the filter envelope. This is the setup of the original Prophet-600 and versions 2.0 and 2.1 RC3. It is also the default in v2022.
- *poly-amplitude*: In this routing both the amplitude and the poly-mod are modulated by the 2nd envelope.
- *poly*: In this routing the amplitude is modulated by the filter envelope and the poly-mod is modulated exclusively by the 2nd envelope. Therefore the poly-mod time modulation is completely free, at the expense of having filter modulation and amplitude tied together.
- *gate*: In this routing the amplitude is not modulated at all, but behaves like a gate (similar to electronic organs). The poly-mod is modulated exclusively by the assignable envelope. Therefore the poly-mod modulation is completely free at the expense of a less dynamic sound evolution.

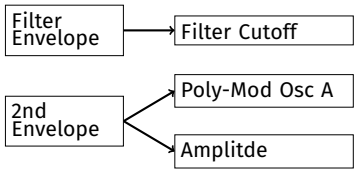
[8] Up to version 2.1 RC3 the filter envelope was ”hard wired” to poly-mod.

Envelope routing options using the additional patch parameter **(555) Envelope Routing**

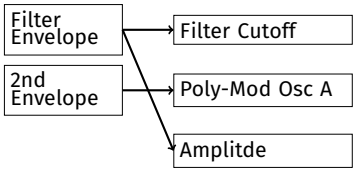
standard routing:



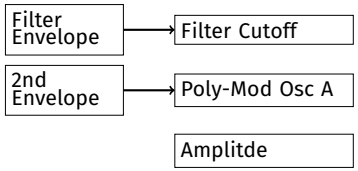
poly-amp routing:



poly routing:



gate routing:



Oscillator sync and poly-mod

As explained in section 3.3, you can sync oscillator A to oscillator B using the **(OSC A) Sync** switch in the sub-panel for oscillator B. This can also be view as a variant of cross-modulation. The sync function can also be combined with frequency modulation as described above.

Combining oscillator sync with poly-mod pitch modulation by envelope can be used to produce interesting short transients or slow sound evolutions. For example, when oscillator A is a pulse wave, then sync'ing this wave to oscillator B and time modulating the pitch of oscillator A has the effect of pulse width modulation by envelope. It creates a change in timbre which can be short to simulate plucking an brassy attacks or long to produce a sweep of the sound spectrum.

Starting points for patch design with poly-mod and sync

Using poly-mod in combination with oscillator sync, different modulations strengths for different oscillator pitches with different filter and modulation shapes - this offers a very wide field of experimentation. Here are two prototypical starting points:

- Set the envelope routing to *standard*, tune both oscillators to the same frequency. In the poly-mod section turn source VCO B to zero and deactivate the filter modulation. Then set VCO A to sync and switch on the poly-mod target frequency oscillator A. Experiment with the poly-mod envelope source amount, oscillator A frequency and filter envelope attack and

decay. This produces an evolving sync sound which, for high envelope amounts, can be tuned to a very assertive synth solo sound.

- Set the envelope routing to *standard*, tune both oscillators to the same frequency. In the poly-mod section turn the poly-mod envelope source amount to zero, deactivate the filter modulation and switch on the poly-mod target frequency oscillator A. Switch off VCO A sync. Now experiment with the poly-mod envelope source amount and oscillator B fine tune. Either you enjoy the drastic spectra you will get for most of the settings or you carefully adjust the frequencies to find the sweet spots at which the spectrum clears. This is the typical way to produce very rich and brassy timbres but normally this also needs some filter cut-off to softly cut away the highest frequencies. For stronger poly-mod, oscillator B fine tune may not be sufficient and it may be necessary to also adjust the frequency of oscillator A. For this it is useful to set the granularity of the frequency dials to free/octave (i.e. oscillator A free and oscillator B octave). Notice that other controls also influence the pitch, such as the oscillator B wave selection and pulse width (see description of bleed bug below) so that you may find that you need to compensate for this effect with fine tuning.

Technical excursion: pulse width bleed bug

The firmware versions 2.0 and 2.1 RC3 contain a software error which incorrectly resets the pulse widths of oscillators A and B. This manifests itself under the following conditions: **(Poly-Mod) Oscillator B** is turned up and the poly-mod target **(Poly-Mod) Frequency Target** is active. The wave shape pulse of oscillator B is not activated. In this case the incorrect reset of the pulse widths leads to a wrong frequency in oscillator A. The effect becomes stronger with higher settings of **(Poly-Mod) Oscillator B**. In version v2022 this bug is corrected but in order to ensure backwards compatibility of older patches, the bug is automatically switched on when loading such patches to faithfully reproduce the behaviour of prior firmware versions. The bug toggle is the **(444) Pulse Reset Bug Switch** which can be set to *off* and *on*. The default value (default patch) is *off*.

3.10 Arpeggiator / sequencer clock and MIDI support

The Prophet-600 offers a polyphonic sequencer with two tracks and an arpeggiator with three distinct pattern types. The details of operation are explained in sections 3.11 (arpeggiator) and 3.12 (sequencer).

The sequencer and arpeggiator share the same clock, which can simultaneously run both sequencer tracks and one arpeggiator pattern. In order to be able to operate the sequencer and/or arpeggiator in sync with other instruments, the upgraded Prophet-600 provides three different clock source methods which are configured in the miscellaneous setting **Clock Sync**.

- *internal*: the internal clock is used. The **(0) Arpeggiator/Sequencer Speed** is edited like a menu patch parameter associated with number 0
- *MIDI*: the clock is synced to the MIDI clocked received on the MIDI receive channel as set in the miscellaneous settings
- *tape*: the clock is synced to a fraction of a pulse clock received via the Tape In input on the rear of the instrument.

When the clock source is set to external (either *MIDI* or *tape*) the **(0) Arpeggiator/Sequencer Speed** sets the clock divide. The values are: 192,168,144,128,96,72,48,36,24,18,12,9,6,4,3,2.

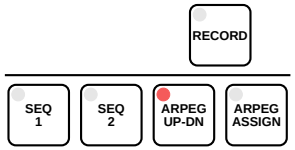
It is also possible to sync the LFO to the sequencer / arpeggiator clock (internal or external). The menu parameter **(111) LFO Clock Sync** provides the options *off* (no sync, free running LFO) or one of the following *1,2,3,4,5,6,8*. The latter sets the number of clock beats after which the LFO in reset.

Note that you can also operate the sequencer and/or arpeggiator in unison mode with all features of that mode (see section 3.7). However, the voices are not shared between the tracks and the arpeggiator. The priority is as follows: arpeggiator, sequencer track 2, sequencer track 1. If activated, the arpeggiator will use all voices in unison mode and when the sequencer tracks 1 and 2 have a note on the same beat, only track 2 will play. Unison mode should be activated before starting the sequencer or arpeggiator.

3.11 Arpeggiator

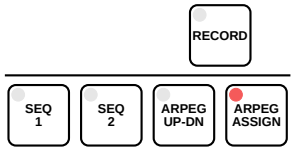
The upgraded Prophet-600 supports the three arpeggiator patterns *up/down*[9], *assign* and *random*[10] which are activated using the arpeggiator buttons as described in the following.

Arpeggiator pattern *up/down*:



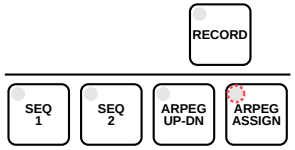
This pattern plays an up/down note sequence. For example, holding down C3 C4 E4 G4 will play: C3 C4 E4 G4 E4 C4 C3 C4 and so on. n notes will result in a pattern which repeats itself after $2n-2$ beats. To start the arpeggiator in *up/down*, press the key **Arpeg Up/Dn** on the key pad. Its LED will light solid when active.

Arpeggiator pattern *assign*:



This pattern plays the sequence of notes in the order in which they are played on the keyboard. To start the mode press the **Arpeg Assign** button on the key pad and play the notes in the order you want them sequenced. The key LED will light solid.

Arpeggiator pattern *random*:



In random mode the arpeggiator replays the notes at random. To enter this mode press the **Arpeg Assign** key twice until its LED blinks and then play the notes.

It is possible to sync the LFO to the arpeggiator using the additional patch parameter **Clock Sync**, see section 3.10.

Latch mode

[9] The pattern is different in firmware versions 2.0 and 2.1 RC3: the lowest and highest note in the up/down pattern were played twice (e.g. ”1234432112...”) while now they are only played once (e.g. ”12343212...”).

[10] In firmware versions 2.0 and 2.1 RC3 the random pattern was slightly different. Formerly, a single note could be randomly repeated two or more times while now a note is never repeated (unless there is only one note to play).

With all arpeggiator modes, press the **Record** button on the key pad to enter the arpeggiator latch mode in which played notes are held, i.e. are sustained after the keys are released. The LED of the **Record** button will light solid. Playing additional notes in this mode will add them as additional notes to the existing sequence, up to a maximum of 128 notes. To clear the notes from the sequence, press the **Record** button to switch the latch mode off. The LED of the Record button will go off.

The foot switch has the same function as pressing **Record** while the arpeggiator is running. It toggles the arpeggiator latch mode as indicated by the **Record** LED.

Transposition

The arpeggiator pattern can be transposed on the fly. To do so use the keyboard in **Shift Mode** or **Shift Lock Mode**, see section 4.5.

Controlling the arpeggiator using MIDI

It is possible to input notes into the arpeggiator via MIDI. The MIDI behaviour in this context depends on the MIDI mode (*local on* or *local off*, see section 4.3) in the following way when the arpeggiator is running:

- *local on*: Keys played on the keyboard are entered into the arpeggiator. Transposition works normally. MIDI notes sent to the Prophet-600 play normally. This means that they are not sent to the arpeggiator but play on top of the running arpeggiator.
- *local off*: Keys played on the keyboard are not entered into the arpeggiator but are sent out as MIDI in the normal way. Transposition of the arpeggiator works from the keyboard and from external MIDI. MIDI notes sent to the Prophet-600 are input into the arpeggiator. MIDI cannot play normally on top of the arpeggiator.

Troubleshooting

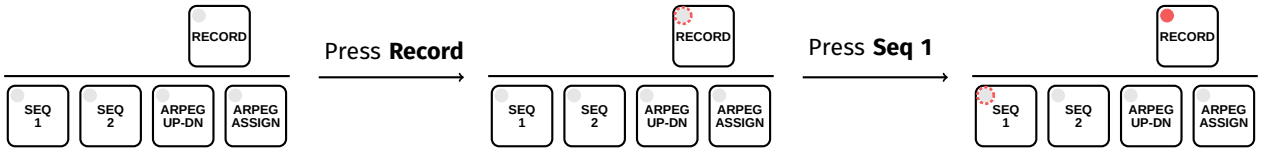
If the arpeggiator is activated and keys are held down or have been latched and still no sound plays, then it is worth checking the following potential reasons: Is the sync set to *internal* and is the **(0) Arpeggiator/Sequencer Speed** greater than zero? If the sync is not *internal* does the instrument receive a clock via MIDI on the correct channel or via tape in? If the instrument in *local on* mode (see section 4.3)?

3.12 Sequencer

The live sequencer of the original Prophet-600 has been replaced by a step sequencer which supports polyphony. You can program two tracks which can play simultaneously. Sequence 1 is started by pressing the button **Seq 1**, sequence 2 is started by pushing the button **Seq 2**. The LED of the running sequencer is always on. Live sound control via panel controls, menu parameters and MIDI CC as well as playing via keyboard and/or MIDI is fully supported while the sequencer is running. To stop either sequence, press the respective sequencer button again. The LED of the stopped sequencer is off.

Creating / editing a sequence

The following is an example work flow for adding notes, rests and ties to a sequence that covers all editing functions. To enter the editing mode press the **Record** button and then the respective sequencer track button as shown below (example for editing sequencer track 1):



1. Press the **Record** button. Its LED starts flashing.
2. Press the sequencer button of the sequence you would like to edit (**Seq 1** or **Seq 2**). The LED of the respective sequencer button starts flashing, and the **Record** button LED becomes solid. The sequence is in **Sequencer Edit Mode** and awaits notes, ties and rest to be added to the sequence. Note: The sequencer is running in this mode, so as soon as the first note is entered this note is repeatedly replayed at the current sequencer/arpeggiator tempo. If notes have previously been added to the sequence, this sequence is played and any new notes are added.
3. Adjust the tempo. Note that you can set the **Data/Speed Dial** to control the clock tempo before starting the sequencer editing mode. The parameter selection is preserved. Be careful when selecting the parameter in record mode. It requires that you press and hold down the **To Tape** button. Otherwise pressing the **button 0** has a different, sequencer specific function: it clears the sequence!
4. Play the notes to be added to the sequence. The display screen shows the number of steps already added, i.e. it is the current length of the recorded sequence including rests and ties.

5. Pressing **button 0** resets the sequence and deletes any notes. Note: there is no confirmation step.
6. In **Sequencer Edit Mode** the **button 2** on the **Number Pad** inserts rests or ties. It adds a rest when no key is pressed, and a tie when a key is pressed.
7. Added notes (or chords), ties and rests can be successively deleted/undone using the **button 1** on the number pad.
8. In order to complete the sequence editing and exit the **Sequencer Edit Mode**, press the respective sequencer button (**Seq 1** or **Seq 2**). The sequencer button LED becomes solid, and the **Record** button LED does off. The sequence is saved.
9. To start playing the sequence, press the respective button, (**Seq 1** or **Seq 2**). The respective LED is on when a sequence is running.

Transposing a sequence

To transpose a sequence use the keyboard in **Shift Mode** or **Shift Lock Mode**. The C3 key is zero or standard tuning, and will not transpose. Playing C sharp 3 will transpose the sequence up a half step, playing D3 will transpose the sequence up a whole step, etc. See section 4.5 for details.

Troubleshooting

By default the sequences are empty, so even with the started sequencer no sound will play. If either sequence is activated and no sound plays, then it is worth checking the following potential reasons: Has a note been added? Is the sync set to *internal* and is the speed greater than zero? If the sync is not *internal* does the instrument receive a clock (via MIDI on the correct channel or via tape in)?

3.13 Vibrato

The upgraded Prophet-600 features a dedicated vibrato. Its parameters are available in the additional patch parameter menu as follows.

- The **(2) Vibrato Speed** sets the vibrato frequency in a range of approximately 1/20 to 60 Hz
- The **(22) Vibrato Amount** sets the vibrato strength

- The **(222) Vibrato Target** sets the vibrato target. The options are *VCO* (modulating the pitch of oscillators A and B), *VCA* (modulating the amplitude) as well as *VCO A* and *VCO B*(modulating only the pitch of one oscillator A or B, respectively)

The vibrato shape is a triangle wave. The vibrato effect can be controlled by either the modulation wheel or a modulation delay function. The additional patch parameter **(3) Modulation Wheel Target** can be set to *vibrato* or *LFO*. When set to *vibrato*, pushing the **Modulation Wheel** up adds vibrato strength. The strength of the **Modulation Wheel** is controlled by the additional patch parameter **(333) Modulation Wheel Range**. If you set the **Modulation Wheel** target to *LFO*, on the other hand, the modulation delay is automatically applied to the vibrato, i.e. the vibrato onset is delayed by the amount set by the additional patch parameter **(33) Modulation Delay**. For setting the parameters of the modulation wheel see section 3.15. For complementary functions for LFO see section 3.8.

3.14 Pitch bender

The Prophet-600 comes with a (pitch) **Bender** (without automatic return to the middle position). With the upgrade it is possible to customize the range/action and the target of the bender as part of the patch confguration (see below).

Note: External MIDI pitch bend is added to the internal pitch bend[11]. Full internal and external (MIDI) bend covers twice the range set by the bender range. As a consequence, the external MIDI bend value cannot be reset/overridden using the built in bender, but the external value is reverted to zero during the bender calibration process (see below).

- The bender can target the oscillator pitch, the filter cut-off frequency or the amplitude. This is set using the additional patch parameter **(6) Bender Target** which provides the options *off*, *AB* (bend of the pitch of both oscillators), *VCF* (modulating the cut-off frequency), *volume* (modulating the amplitude) and *b* (bending the pitch of oscillator B only).
- The range of the bender can be configured using the additional patch parameter **(66) Bender Range**. It offers the choices *2nd* (bend a full tone), *3rd* (bend 3 semi-tones), *5th* (bend 5 semi-tones) and *octave* (bending an entire octave).

The upgraded Prophet-600 also offers the option to calibrate the bender middle position using one of the miscellaneous settings/function (see section 5.3). This function is useful to remove jittery behaviour of the bender around the middle position. In order to calibrate the bender follow the steps below.

[11]This is the behaviour of the original firmware. Since the bender does not return to zero a bend override by external MIDI leads to unwanted discontinuous pitch changes when the bender of the Prophet-600 is touched. The addition of both bends is therefore more consistent.

1. Move the bender to the middle position.
2. Activate bender calibration by pressing **button 3** in **Shift Mode** or **Shift Lock Mode** (the display will show that the bender calibration mode is active). Pressing **button 3** again applies and confirms the calibration and exits the calibration mode.

As mentioned above, the current active external (MIDI) bend value is set to zero during the calibration process. To be precise, it is set to zero with the first press of **button 3** in **Shift Mode** or **Shift Lock Mode**.

3.15 Modulation wheel and modulation delay

The upgraded Prophet-600 has two methods for controlling the modulation effect of the LFO and the vibrato and these properties are stored with the patch. The first one is the **Modulation Wheel**. The target of the modulation wheel can be set using the menu patch parameter **(3) Modulation Wheel Target** which provides the options *LFO* and *vibrato*.

The second method is a modulation delay which delays the onset of the modulation effect. The delay is always applied to the modulation source which is not controlled by the **Modulation Wheel**. So when the modulation wheel target is set to *LFO* the delay applies to the vibrato and when it is set to *vibrato* the delay is applied to the LFO.

The target setting as well as the modulation wheel strength and the delay time are set as follows.

- **(3) Modulation Wheel Target** provides choices *LFO* and *vibrato*
- **(33) Modulation Delay** is a continuous menu parameter (range 0...99) which controls the delay time
- **(333) Modulation Wheel Range** offers 4 options: *touch*, *soft* (this is the default), *high* and *full*

The parameter **(333) Modulation Wheel Range** affects the **Modulation Wheel** action for all variations except *VCA*. If the target is *vibrato* and the vibrato target is *VCA* the wheel action is always effectively *full*. The reason is that amplitude modulation is a weak effect and there is no practical value in making the **Modulation Wheel** softer in this case.

Note: the **Modulation Wheel** action has been modified in all settings compared to version 2.0 and 2.1 RC3. It is now exponential to provide a smooth onset for better playability.

3.16 Glide

The Prophet-600 offers a glide function in polyphonic and unison mode. Generally speaking, the glide time determines the how fast the pitch glides from the start note to the target (played) note. Glide is based on voice, i.e. once a note is assigned to a voice the glide start note will be the last note of that voice (rather than, for example, the last played note).

The way you set the glide time depends on the current active panel layout. When the panel layout is set to *GliGli* you must use the menu parameter **(777) Glide Time** to set the glide time. When the panel layout is set to *SCI*, the menu parameter **(777) Glide Time** is suppressed and the glide time is set by the **Glide Time** dial instead.

For details on the panel layout options see section 4.2.

Note that since glide is voice based, the voice assignment logic (as set using the parameter **(77) Assigner**, see 3.7) has a strong impact on the behaviour of glide in polyphonic mode. The voice assignment logic determines the glide start note once a new note is played.

3.17 Detune

The upgraded Prophet-600 provides a method to detune the 6 voices against one another. This effect is used to make unison sounds broader and more lively. Detune is only applied when unison is activated. The menu parameter for setting detune is **(8) Unison Detune Amount**. Note: the effect of the vintage spread (see section 3.18) is applied in all modes and is therefore added to detune in unison mode.

The detune functions has been changed in version v2022. Compared to versions 2.0 and 2.1 RC3 the detune is scaled with pitch is less pronounced for higher pitches. This change is designed such that the perceived detune is constant (or at least more constant) over the available pitch range.

3.18 Vintage / spread

The upgraded Prophet-600 offers a continuous parameter to introduce deliberate variations in voice tuning and envelope times (both envelopes) in order to emulate imperfections of hardware implementations. This menu parameter **(88) Vintage/Spread Amount** ranges from 0 (no variations) to 99 (strongest variation). The variations in envelope times are relative, i.e. the variations are smaller at shorter times. At full spread the differences in attack and release can be significant, especially at longer times. The sustain levels of the two envelopes are not changed. They are the same for all voices. Note that in unison mode, the spread

is applied in addition to the detune (if activated). The tuning differences introduced using **(88) Vintage/Spread Amount** are generally less pronounced than unison detune.

Note that **(88) Vintage/Spread Amount** is implemented as a menu parameter. This patch parameter replaces the former spread switch in the miscellaneous settings menu, which has been omitted in version v2022.

3.19 External CV in

The Prophet-600 offers the option to control the filter cut-off frequency using an external voltage source fed into the Prophet-600 via the **Filter CV In** plug on the rear panel. Applying a voltage to that input raises the filter cut-off frequency.

In the upgraded Prophet-600 the strength of the external CV input effect can be controlled using the menu parameter **(44) External Voltage**. The parameter has been newly introduced in version v2022. In versions 2.0 and 2.1 RC3 the external CV control strength is fixed and it is set to maximum by default. In the new version on the other hand the patch parameter value is zero and it therefore needs to be changed before using the external voltage control.

One application of the external voltage control is that it is used in the popular ”noise mod” of the Prophet-600 and in this mod the parameter controls the level of an additional noise source.

3.20 Velocity sensitivity

The Prophet-600 is equipped with a non-weighted keyboard without support for velocity. However, when played via MIDI, the upgraded Prophet-600 supports velocity sensitivity of amplitude and filter cut-off frequency. The respective velocity amounts are configured using the two menu parameters **(9) Amplitude Velocity** and **(99) Filter Velocity**.

By default the **(9) Amplitude Velocity** is set to half-way and the **(99) Filter Velocity** is set to zero. Amplitude and filter velocity are not supported in arpeggiator and sequencer.

Synthesizer setup and maintenance

4.1 Tuning

The Prophet-600 is a synthesizer using microchips that implement voltage controlled oscillators. This means that the oscillator frequency is set by applying a voltage to the control pin of a microchip. The relationship between absolute voltage and resulting frequency is not precise. It depends on many factors, such as smallest differences in the manufacturing process of the chip itself and the surrounding circuit components and operating temperature (the temperature of the components also changes during operation). Therefore such synthesizers need to be tuned regularly. The Prophet-600 has a tuning procedure which determines the correct voltages which need to be applied to oscillators A and B and to the filter frequency control in order to achieve a perfect pitch. There is one measurement point per octave over several octaves (over 8 octaves in principle, even though the lowest and highest are extrapolated). When a note is played, the voltage to achieve the corresponding pitch is interpolated. This is how the Prophet-600 stays in tune.

The tuning data is stored in the settings so it is readily available after the instrument is switched on. However, while the tuning primarily takes into account the individual characteristics of the chips one should be aware that voltage-to-pitch characteristics depend on different factors, notably on temperature, and can therefore ”drift”. Therefore the tuning procedure must be started manually by pressing the **Tune** button (while not in **Shift Mode** or **Shift Lock Mode** as pressing **Tune** in this case activates the per note tuning mode, see below) when there is the need to do so. Typically, tuning is best after the instrument (or rather: the chips and components) has been given time to warm up to a stable operating temperature, which could be 10-30 minutes into operation. The warm up time can depend on changes in air temperature, humidity and transportation of the instrument, as well as the condition of hardware such as the power supply. Allowing the instrument to warm up will result in less need to run the tune procedure and overall longer tuning stability. It may be necessary to retune the synth during operation. You should trust your ear on this.

Per note tuning

The automated tuning procedure described above applies 12 note equal tempered tuning. To achieve other tunings each of the 12 notes can be tuned manually. To enter the per note tuning press the **Tune** button in **Shift Mode** or **Shift Lock Mode**. The **Tune**

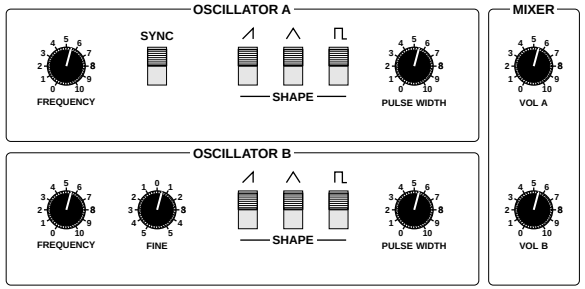
button flashes. In this mode the last played note (on any octave) can be tuned using the mod wheel. From the middle position the tune range is $\pm 1/2$ semitone from equal tempered. To leave the per note tuning mode press **Tune** again.

The per note tuning is stored as part of the patch (in contrast to the overall tuning, which is stored as part of the settings and applies to all presets) and it is retained after full tuning. However, the treatment is different for the active patch in **Live Mode**. The Prophet-600 stores all additional patch parameters of the active patch in **Live Mode** such that they can be recalled when **Live Mode** is (re-)entered. The patch parameters include the per note tuning data. However, when the active patch parameters is recalled upon activation of the **Live Mode**, the per note tuning is reset to equal tempered tuning. So, when you edit the per note tuning for a patch you must stored it in a page in order to keep it.

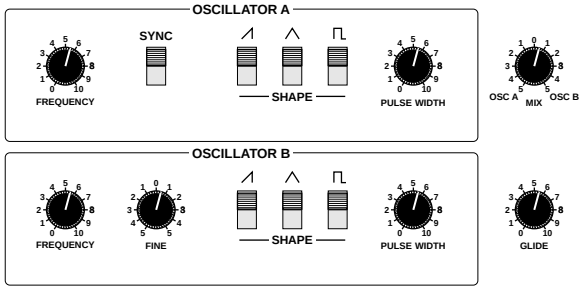
4.2 Switchable panel layout

The original Prophet-600 panel layout (referred to as *SCI*) in the following provides the panel controls **Mixer** (oscillator A vs. B mix) and **Glide Time** (glide time). The *GliGli* panel layout re-designated the **Mixer** to control **(OSC A) Volume** and the **Glide Time** to control **(OSC B) Volume**. During the main development phase of the firmware upgrade project a panel overlay from synthgraphics [9] was available which - among other differences - reflects the change in function of those two controls. The two layouts are shown below.

GliGli panel layout



SCI panel layout



Since then many users have asked for the original panel layout to be made available again. With version v2022 it is now possible to switch between the two layouts, *GliGli* and *SCI*. However, this panel switching has some technical implication. Concretely, the single **Mixer** control does not cover the entire value range provided by the separate **(OSC A) Volume** and **(OSC B) Volume** controls. In order to be able to translate between the two ”perspectives” a new menu parameter **(888) Drive** has been introduced which, in combination with **Mixer**, provides an alternative but fully equivalent way of setting the volume of oscillators A and B. For

details on this function see section 3.3. Technically, the patch storage uses volume **(OSC A) Volume** and **(OSC B) Volume** rather than **Mixer** and **(888) Drive**. This means that there is also no MIDI CC equivalent of **Mixer** and **(888) Drive**.

To change the panel layout press **button 6** in **Shift Mode** (i.e. while pressing **From Tape**) or **Shift Lock Mode** (i.e. after double-pressing **From Tape** and blinking **From Tape** LED). Pressing **button 6** once shows the current active layout and pressing **button 6** twice changes the panel layout.

The layout switch does not only affect the function of the two controls on the panel. It also modifies the parameter menu. In *GliGli* layout the parameter **(888) Drive** is suppressed while in *SCI* layout the menu parameter **(777) Glide Time** is suppressed. The parameters are preserved. The value selected using the **Glide Time** in *SCI* layout is transferred to the menu parameter **(777) Glide Time**. And the value of **(888) Drive** used in *SCI* layout is used to determine the value for **(OSC A) Volume** and **(OSC B) Volume** when switching from *SCI* to *GliGli* layout.

4.3 MIDI integration

For supporting the integration of the Prophet-600 in a studio or live environment using MIDI, several MIDI functions are supported.

- Note send and receive
- Pitch bend and modulation wheel actions
- Receive of all patch parameters using MIDI CC
- Local on and local off modes
- SysEx support for patch management (for this see section 4.7)

The MIDI send and receive channels can be set using dedicated miscellaneous settings as follows. In **Shift Mode** or **Shift Lock Mode**:

1. **button 1** cycles through the MIDI receive channels, from *OMNI* through channels *Ch1* to *Ch16*
2. **button 2** cycles through the MIDI send channels, from *Ch1* to *Ch16*

Notes, performance and MIDI CC

The Prophet-600 sends the following MIDI signals.

1. Key on, key off in all modes
2. Pitch bender
3. Hold pedal (in polyphonic mode, not in unison/chord mode or running arpeggiator or sequencer)
4. Modulation wheel (technically a MIDI CC event)
5. Program change (technically a MIDI CC event)

All of the events can also be received and executed with the following exceptions/modifications.

1. In arpeggiator mode incoming MIDI note on/off events are not entered into arpeggiator but play normally when in *local on*. To enter notes to the arpeggiator using MIDI you need to switch to *local off* mode.
2. In **Shift Mode** or **Shift Lock Mode** incoming MIDI notes do no cause transpositions (but are ignored) in *local on* mode.
3. External MIDI pitch bend is added to the pitch bend from the on-board bender
4. MIDI Hold Pedal events are also applied in unison/chord mode or running arpeggiator or sequencer where it has the effect of latching notes/chords
5. MIDI program change messages are only applied in **Preset Mode** and are ignored otherwise

The instrument does not send MIDI CC (apart from modulation wheel and program change). However, the Prophet-600 supports a comprehensive list of general and specific MIDI CC events. For reference see section 6. Incoming MIDI CC events are only applied in **Preset Mode**.

Local off mode

The Prophet-600 supports a *local off* mode which is activated using **button 0** with **Shift Mode** or **Shift Lock Mode**. In *local off* mode various internal events (such as notes) are not executed in the Prophet-600 directly but they are still sent as MIDI events.

The typical setup would be to connect to a MIDI recorder and feed the recorded MIDI messages back to the instrument for replay. The *local off* mode ensures that there is no doubling of events.

Note that in *local on* mode external MIDI does not play into the arpeggiator but plays over the arpeggiator while the local keyboard plays into the arpeggiator. Since the arpeggiator and the playing the external MIDI notes share the 6 voices this can lead to interference of the two, such as missed arpeggiator notes, depending on what play using MIDI. When in *local off* mode MIDI notes play into the arpeggiator (if active) and in this mode you can also use external MIDI notes for transposition of arpeggiator and sequencer like you would with the internal keyboard (i.e.. in **Shift Mode** or **Shift Lock Mode**). However, transposition does never affect normal playing via MIDI in any mode. The reason is that the Prophet-600 sends MIDI notes with transposition already applied. For consistency reasons, incoming MIDI (which may in fact be a recording of MIDI notes from the Prophet-600) must therefore not be transposed again. If you need transposition from your external MIDI controller you would need to use whatever function the MIDI controlled provides to transpose the MIDI notes sent to the Prophet-600.

Overview of MIDI integration

	local on			*local off*		
	Apply local	Send MIDI	Apply MIDI	Apply local	Send MIDI	Apply MIDI
Keys on/off playing	Yes	Yes	Yes	No	Yes	Yes
Keys into arpeggiator	Yes	Yes	No[1]	No	Yes	Yes
Keys into sequencer	Yes	Yes	Yes	No	Yes	Yes
Modulation Wheel	Yes	Yes	Yes	No	Yes	Yes
Pitch Bend	Yes	Yes	Yes[2]	No	Yes	Yes
Program Change	Yes	Yes	in preset mode	Yes	Yes	in preset mode
Pedal Hold (Latch)	Yes	No	No	Yes	No	No
Pedal Hold (Sustain)	Yes	Yes	Yes	No	Yes	Yes
Transpose	Yes	No	No	Yes	No	Yes[3]
Other MIDI CC	Yes	No	in preset mode	Yes	No	in preset mode

[1] MIDI notes will play normally / freely on top of the running arpeggiator. note that voice sharing between arpeggiator and playing MIDI notes can lead to unwanted interference such as missed arpeggiator notes etc.

[2] Internal and external pitch bends are added, see 3.14

[3] You can transpose the sequencer and the arpeggiator by MIDI, but this has no effect on normal MIDI playing.

4.4 Voice deactivation

The upgraded Prophet-600 provides a method for switching single voices off. This can be useful to suppress a voice that is damaged or for troubleshooting, e.g. isolating voices for testing. The ”voice masking” is configured using the miscellaneous settings with **button 4** and **button 5** in **Shift Mode** or **Shift Lock Mode** as follows

1. Press **button 4** to cycle through the 6 voice of the Prophet-600. This selection determines the functionality of the key **button 5** in the setting menu.
2. Press **button 5** to toggle between on and off of the voice selected using **button 4**.

Note: the voice masking is stored in the settings. It therefore applies to all patches and all modes and it is retained through a power cycle. If you are missing a voice you should check the voice masking.

4.5 Transposition

In the upgraded Prophet-600 there are two types of transposition which you set in the same way. The first one is the transposition of the internal keyboard. The second is the on-the-fly transposition of the sequencer and arpeggiator. In **Shift Mode** or **Shift Lock Mode** the last pressed key will define the transposition, the middle C3 being zero transposition. The transposition is also scrolled through the display for conformation. As stated above, adjusting the transposition affects both, playing from the internal keyboard and the sequencer and arpeggiator.

You can transpose the sequencer and arpeggiator on-the-fly by MIDI too (play a MIDI note while in **Shift Mode** or **Shift Lock Mode**) but only in *local off* mode. Otherwise MIDI notes are ignored. Note that while transposition (from internal keyboard or from external MIDI) transposes playing from the internal keyboard, the transposition will never affect how notes from external MIDI are played. This is for consistency reasons because MIDI notes sent by the Prophet-600 already include any transposition, both for normal playing and for sequencer and arpeggiator. When this data is recorded and played back to the Prophet-600 the transposition must not be applied (again) to the incoming MIDI notes. The keyboard transposition should be thought of as a configuration of the internal keyboard rather than a transposition of the patch. If you need transposition from an external MIDI keyboard you have to rely on the functionalities of your MIDI keyboard.

4.6 VCF limit

In the upgraded Prophet-600 it is possible to limit the cut-off frequency of the filter to 22kHz in order to avoid strange filter behaviour in the ultrasonic range. The limit is switch on using **button 9** in **Shift Mode** or **Shift Lock Mode**.

4.7 Managing sound libraries via SysEx

For the following instructions to work you need to have your Prophet-600 connected via MIDI to a device where you can receive, store and send MIDI SysEx files. Make sure that the MIDI receive channel on the Prophet-600 and the MIDI send channel of the sending device are the same (see section 4.3).

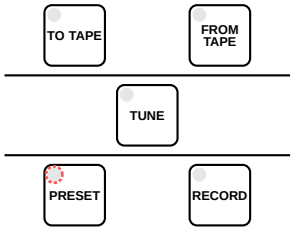
In **Preset Mode** the Prophet-600 loads patch data is receives via MIDI into the active controls. This is like loading a patch from memory where the panel controls may not (and typically don't) show the values of the patch being played. You can edit and store the patch in the normal way. This way of loading a patch is useful if you want to try out a patch from MIDI SysEx before deciding if it should be kept and without having to allocate a storage number in memory. The Prophet-600 will never write incoming patch MIDI SysEx to memory unless the **Patch Management Mode** is activated.

Patch Management Mode

In order to manage patches in memory via SysEx you need to enter the **Patch Management Mode**. In this mode any patch data received via MIDI SysEx is loaded directly into the storage number as indicated in the SysEx data. This extends to the load of the entire patch bank. In patch management mode you can also dump single patches as well as the entire patch bank.

To enter the patch management mode use **Shift Mode** or **Shift Lock Mode** and press the **Record** button. The **Patch Management Mode** is indicated by a flashing **Preset** button. To leave the mode, press the **Record** button.

Patch management mode:



To Tape	
From Tape	(normal function)
Tune	(normal function)
Preset	Dump entire patch bank
Record	Leave patch management mode
Number pad	Patch number entry
Display	Patch number input, confirmation text
Keyboard	(normal function)
Performance	(normal function)
Panel controls	(normal function)

Dumping the patch library as SysEx

To dump the entire patch bank press the **Preset** button. The screen will scroll a confirmation message. Note, however, that only valid presets are exported. Any unused patch slots are skipped.

Dumping single patches as SysEx

To dump a single patch in patch management mode simply enter the patch number on the number pad. Note, that if there is no valid patch data stored in the selected patch slot, the entry will be accepted but there will be no dump.

Note: the Prophet-600 also supports MIDI dump requests. When sending a SysEx patch request for a particular patch number to the Prophet-600 you will receive the corresponding single SysEx back (on the specified MIDI send channel) provided the specified patch number is between 0 and 99 and a valid patch is stored at this number. This can be done in all modes at any time, not just in **Patch Management Mode**.

Loading patches

Loading MIDI SysEx patches to storage is done by sending the patch SysEx containing the data to the Prophet-600 in **Patch Management Mode**. Each patch contains a patch number. If the patch number is between 0 and 99 the data will stored in that patch slot overwriting the existing stored patch. Therefore, loading a SysEx patch library will overwrite the entire library stored on the Prophet-600 when **Patch Management Mode** is activated. When a patch has been loaded to storage the number of the patch number is shown on the display.

In normal **Preset Mode** patch MIDI is only loaded into the active control values as described above. In **Live Mode** incoming patch MIDI SysEx is ignored. Requiring **Patch Management Mode** to be activated for MIDI-to-storage operations protects your patches from accidental overwrite. Still, it is always advisable to regularly archive patches in SysEx files outside the instrument.

4.8 Scaling adjustment

Due to ageing and changing component characteristics it may become necessary to calibrate the oscillator and filter components. Off calibration, the worst case could be that the tuning procedure fails to tune one of the frequencies. Calibration or ”scaling adjustments” are done using the dedicated variable resistors next to the chips. The user should refer to the Sequential Circuits maintenance manual [8] and available documentation of the circuit board to locate the components.

For performing the procedure the Prophet-600 must be started in **Maintenance Mode** by shortening ”TP301 SCALE” and then powering up. In this mode the display shows the current oscillator/filter (A1..A6,B1..B6,F1..F6) and a value that needs to be made close or equal to zero by tweaking the corresponding adjustable resistor. Press **button 1** to go to next oscillator/filter, press **button 2** to go the previous one.

When done, turn off the instrument and unshort TP301. After the next start the Prophet-600 will be in normal mode.

Note: the scaling adjustment can also be done by refitting the Z80 processor and following the instructions in the Sequential Circuits maintenance manual [8]. The procedure is different but essentially achieves the same.

4.9 Firmware upgrade

For performing the upgrade on a Prophet-600 with an original Z80 processor please refer to the GliGli modification instructions [5]. The following section is for owners of a modified Prophet-600 with Teensy board and a prior version of the GliGli upgrade installed. If modifying a Prophet-600 for the time, you may directly flash the Teensy board with software version v2022. It is not necessary to first install a lesser version and then upgrade.

Firmware upgrades are provided as MIDI SysEx files (*.syx) and as code files (*.hex) for USB transfer to the Teensy board.

Firmware upgrade via SysEx

In order to perform the upgrade follow the steps below.

1. Store and save the patch bank and make a note of important settings
2. Switch off the instrument
3. Connect your computer via a MIDI interface to the instrument (the only required data direction is from computer to the Prophet-600) Switch on the instrument while holding down the buttons **To Tape** and **From Tape**. The display should show an ”U” for upgrade on the left digit.
4. Send the firmware SysEx file to the instrument. This process takes a while. The display shows a spinning segment. Once the process is completed, the display shows an ”S” for success. If the upgrade ends with “E” for error on the display, the upgrade has failed and the upgrade procedure must be repeated until it succeeds. It is not recommended to start the instrument in normal mode after a failed upgrade.
5. Switch the instrument off and on again. The scrolled welcome message should state the firmware version

You are recommended to follow the MIDI SysEx part of the modification instructions [5]. In particular, the ”Delay between buffers” and the ”delay after F7” should be set to at least 250ms.

The settings and patch parameters stored on a unit are untouched when the firmware is done by MIDI. After the upgrade, the sound of patches will unchanged provided the previously installed version is one for which backward compatibility is supported. This applies to version 2.0 and 2.1 RC3. Downgrading the firmware version is technically possible but preservation of settings and patch parameters is not guaranteed in that case.

Firmware upgrade via USB

The Teensy board has a USB port which can be used to program the processor. Loading the firmware code to the Teensy board in this way via USB has a lot of advantages but several aspects need to be taken into consideration. Here are some of the most important differences.

- While upgrading by by MIDI works via the MIDI In socket at the rear, the USB transfer requires opening up the Prophet-600
- USB transfer requires downloading and installing a PJRCc software on a computer
- USB transfer erases all patch and settings data
- USB transfer is very fast (seconds) and highly reliable

- Unless the Teensy board is temporarily removed for the upgrade procedure, the USB transfer requires access to the Teensy board from the side which is blocked by the storage battery of the Prophet-600
- Unless the Teensy board is temporarily removed for the upgrade procedure, the transfer has to be done with power switched on and with the Prophet-600 panel lid open in at least one task

To reach the Teensy board, unscrew the top 2 screws on the wooden side panels on either side. After this, the panel can be lifted up from top end of the keyboard. It is advisable to be careful when opening the panel lid because there are delicate wire connections from the keyboard and the circuit boards inside the Prophet-600 and the circuits board attached to the panel lid. These connections can break, in particular when they have become brittle with age. There is no real danger of serious damage but if broken, the wires need to be re-soldered.

Attention: the following instructions involve opening up the Prophet-600 and performing little tasks inside the unit while the power is on. Obviously, there is the danger of electrical shock. You do this at your own risk and you must consult a professional in case of doubt.

The most convenient method for upgrading the Teensy code via USB is to use the power supplied by the Prophet-600 when it is in the socket or to provide an alternative power source for example using a bread board. In both cases the procedure does not require that the 5V bridge in the Teensy modification be reconnected (see GliGli modification instructions [5]).

Method 1: Use the Prophet-600 as the power source. For this the Teensy 2.0++ board stays in place and the USB connected with the Teensy 2.0++ board in the unit. One practical problem is that the USB socket of the Teensy board in place is blocked by the battery. The battery is only needed to store patches when the Prophet-600 is operated with the Z80 in place. With the Teensy upgrade it is no longer needed. To solve the problem there are three possibilities: 1) remove the battery, 2) stack several 2x20 sockets on top of each other so that the Teensy board / the USB socket is above the battery or 3) implement a different solution for the battery for example a battery holder attached at a convenient place inside the Prophet and connected with wires to the proper $\pm$ connections.

Method 2: Take the Teensy 2.0++ out of the Prophet-600 an place it onto a standard bread board. Follow the board specifications to supply 5V to the correct pins.

1. Download the *.hex file for the firmware version you want to install

2. Connect a USB cable to the USB socket of the Teensy on the side which faces the rear side of the unit. You need a USB cable with a type A plug for this. In method 2 this will need to be done with the panel lid open and power off
3. Connect the cable to your computer
4. Download an install the Teensy software from PJRC [10] and open the application
5. Power up the Teensy 2.0++ board (in method 1 power up the Prophet-600, in method 2 switch on the voltage to the bread board) and then press the button on the Teensy board to activate code transfer mode. Follow the instructions provided with the Teensy loader software.
6. When the code transfer mode is active, select the *.hex file provided for the release version you want to install and click *program*
7. If using method 1, simply click *reboot*, if using method 2 refit the Teensy 2.0++ and then power up. Note that the Prophet-600 will always go into tuning when first started after USB firmware upgrade. Instead of rebooting your may also switch the unit off and on again.
8. After the tuning procedure is complete, the instrument is ready to play. All patch slots will be empty and the active patch in preset mode is the default patch.

4.10 Technical notes on vulnerable exponential FM

The Prophet-600 has a hardware design flaw which leads to a bleed of the DC component of the oscillator B output to the pitch of oscillator A when **(Poly-Mod) Oscillator B** with destination **(Poly-Mod) Frequency Target** is active. This section describes the consequences of this design, methods to assess the impact in your unit (such as detuning effects) and also some measures to compensate for and/or avoid detuning.

While the CEM3340 oscillator chip has a dedicated support for linear frequency modulation, Sequential Circuits chose no to use that option but instead designed the circuitry such that the output of oscillator B is fed directly into the control voltage of oscillator A. The control voltage input of oscillator A (i.e. the input that determines the frequency) is the sum of the note related voltage (i.e. the note which the oscillator is supposed to play) and the output of oscillator B. The amount of oscillator B contribution is set by **(Poly-Mod) Oscillator B**. This is effectively exponential FM: the frequency doubles with each 1V of control voltage input. Exponential FM is common in polyphonic synthesizers and you find the same basic mechanism in the Prophet-5,

for example. Exponential FM has the disadvantage that the poly-mod amount changes the pitch. Since we are dealing with non-calibrated, CV controlled components here, this leads to potential differences between different units as far as the amount of poly-mod is concerned. Medium to strong poly-mod patch may not be immediately transferable from one unit to another, as a result. Empirically, on most instruments the overall differences can be compensated by adjusting **(OSC A/B) Frequency** (better A than B), **(OSC B) Fine** and/or **(Poly-Mod) Oscillator B** for a given poly-mod patch.

There is another problem with the Prophet-600 design, however. The DC component of oscillator B output is not suppressed before the signal is fed into oscillator A. For using the linear FM input, the CEM3340 specification recommends using a capacitor to remove any direct current. No such mechanism is in place in the Prophet-600 exponential FM. As a result, any DC voltage component from oscillator B directly changes the basic frequency of oscillator A. While you may accept the general effect that **(Poly-Mod) Oscillator B** changes the pitch, the presence of the DC voltage has the distinctive disadvantage that chip specific (and therefore voice specific) variations affect the pitch. In the worst case, poly-mod completely detunes the Prophet-600 and makes chord playing impossible. Patches that sound clear and rich on one instrument may sound detuned and outright bad one another.

There are three main sources for poly-mod induced voice-to-voice variations in pitch. First, away from the pulse width mid-position, pulse waveforms have a DC component and therefore oscillator B pulse width creates a shift in oscillator A frequency. The pulse width response is chip specific (with variations within production specifications). For the timbre of the sound such variations do not matter much (unless you are toying with the break-up points on either side) but in poly-mod it means that each voice can produce a slightly different pitch in oscillator A, depending on the pulse width variation of oscillator B. Second, the waveform output can have a general, chip specific DC voltage offset. This affects also the sawtooth waveform. Third: scaling differences in the oscillator A chips. The chip specific relation of voltage to frequency is compensated for by the built-in tuning mechanism. The FM voltage input in poly-mod is not corrected for scaling differences. As a result, the DC components from the voices lead to different voltage shifts depending on the scaling characteristics of the oscillator A chips that are targeted by poly-mod, even in the absence of other variations in the DC components.

The bottom line is that instrument specific voice-to-voice variations in poly-mod response can make strong poly-mod difficult, especially for chord playing. You can try the following measures to reduce the detuning effect of poly-mod.

- *Differences in pulse width*: If your find that pulse width produces a strong voice-to-voice variation in your instrument, you should avoid using the pulse waveform on oscillator B for medium to strong poly-mod patches.
- *Differences in scaling*: It is generally recommended to not rely solely on tuning but also to regularly perform careful scaling adjustment (see 4.8).

- *Sawtooth waveform*: There can be inherent, chip specific differences in the DC offset of output waveforms. Empirically, this seems to be more likely/pronounced in the sawtooth and pulse wave forms. If voice-to-voice variations persist you may try to also avoid sawtooth waveforms on oscillator B for medium to strong poly-mod patches.
- *Chips and circuitry*: In case everything else fails, you may want to consider replacing selected CEM3340 chips. One tested Prophet-600 hardware modification involves inserting a capacitor into the poly-mod circuitry (following the Curtis recommendations for linear FM input, a similarly sized capacitor can be inserted at resistor R430 and so on, see [8]) in order to the remove the DC offset of oscillator B. This change is reported to make the instrument much more consistent but it also alters the characteristics significantly.

Practical suggestion for assessing the voice-to-voice differences in poly-mod response: Create a strong poly-mod patch and calibrate it to be harmonic (i.e. no beating sounds) using one note (voice). Make sure to choose a high sustain and short release. You can then start from key C and play and hold 6 notes up the scale. Each note is assigned to a different voice. When you release all but one note you will hear only that one voice. Check if there is a beating sound. You can either assess the level detuning by listening or ”measure” it by adjusting the frequency of oscillator A to remove the beating. If you choose the **(77) Assigner** option *first* (see section 3.7) and use zero release (filter and amplitude), the relation of note to voice will be completely reproducible. Activating and deactivating voices one-by-one (see 4.4) is also possible (but more cumbersome).

A future version of the upgraded firmware may contain a calibration and correction mechanism in order to compensate for chip specific differences in poly-mod response.

Reference

5.1 Patch parameter overview

Number	Group	1st press	2nd press	3rd press
0	Speed	Seq/Arp Speed *numeric 0...98*		
1	LFO	LFO Shape *tri-square, sin-random, saw-noise*	LFO Target *A&B, A, B, A & B & VCA*	LFO Clock Sync *off, key, 1, 2, 3, 4, 5, 6, 8*
2	Vibrato	Vibrato Speed *numeric 0...98*	Vibrato Amount *numeric 0...98*	Vibrato Target *VCO, VCA, VCO A, VCO B*
3	Modulation Wheel	Modulation Target *LFO, vibrato*	Modulation Delay *numeric 0...98*	Moduation Wheel Range *touch, soft, high, full*
4	Configuration	OSC Pitch Mode *free, semi-tones, octaves*	External Voltage *numeric 0...98*	Pulse Reset Bug *on, off*
5	Envelope	Filter Envelope Shape *fast-lin, fast-exp, slo-lin, slo-exp*	2nd Envelope Shape *fast-lin, fast-exp, slo-lin, slo-exp*	Envelope Routing *std, poly-amp, poly, gate*
6	Bender	Bend Target *off, VCO, VCF, amp*	Bend Range *2nd, 3rd, 5th, octave*	
7	Play Mode	Note Priority *last, low, high*	Voice Assigner *first, cycle*	Glide*) *numeric 0...98*
8	Oscillator	Detune *numeric 0...98*	Vintage (Spread) *numeric 0...98*	Drive*) *numeric 0...98*
9	Touch Sensitivity	Amplitude Velocity *numeric 0...98*	Filter Velocity *numeric 0...98*	

*) The availability of parameters **(777) Glide Time** and **(888) Drive** depend on the panel layout. In *GliGli* panel layout **(777) Glide Time** is available and drive is obsolete. In *SCI* layout the parameter **(888) Drive** is available while the glide parameter is set

using the dedicated dial on the panel.

5.2 Default patch

When powered up, the Prophet-600 starts either in **Preset Mode** (with the last selected patch loaded) or in **Live Mode**. In either mode the user can load a default setup to start a new patch. To load the default patch press the **Preset** button twice in **Shift Mode** or **Shift Lock Mode** (upon first press the display shows that another press will load the default patch). When the default patch is loaded, the Prophet-600 automatically switches to **Preset Panel Mode**. Note that loading the default patch overwrites the patch menu parameters and therefore any unsaved patch parameter values in **Preset Mode** are lost.

The default patch is intended to be a very basic/minimal setup. The values are shown in the table below.

Continuous Parameter	**Value**	**Stepped Parameter**	**Value**
Osc A Frequency	*c0*	Osc A Saw	*on*
Osc A Volume	*full*	Osc A Triangle	*off*
Osc A Pulse Width	*half*	Osc A Square	*off*
Osc B Frequency	*c0*	Osc B Saw	*off*
Osc B Volume	*zero*	Osc B Triangle	*off*
Osc B Pulse Width	*half*	Osc B Sqr	*off*
Osc B Fine	*zero*	Sync	*off*
Cutoff	*full*	Poly Mod Oscillator A Destination	*off*
Resonance	*zero*	Poly Mod Filter Destination	*off*
Filter Envelope Amount	*zero*	LFO Shape	*triangle-pulse*
Filter Release	*zero*	LFO Sync	*off*
Filter Sustain	*zero*	LFO Targets	*A & B*
Filter Decay	*zero*	Keyboard Filter Tracking	*off*
Filter Attack	*zero*	Filter Envelope Shape	*linear*
Amp Release	*zero*	Filter Envelope Fast/Slow	*slow*
Amp Sustain	*full*	2nd Envelope Shape	*linear*
Amp Decay	*zero*	Unison	*off*
Amp Attack	*zero*	Assigner Priority Mode	*last*
Poly Mod Filter Amount	*zero*	Bender Range (semitones)	*5 semitones*
Poly Mod Osc B Amount	*zero*	Bender Target	*pitch A & B*
LFO Frequency	*mid*	Mod Wheel Range	*soft*

Continuous Parameter	**Value**	**Stepped Parameter**	**Value**
LFO Amount	*zero*	Osc pitch mode	*octave-octave*
Glide	*zero*	Mod Wheel Target	*LFO*
Amp Velocity	*mid*	Vibrato Target	*pitch*
Filter Velocity	*zero*	2nd Envelope Fast/Slow	*slow*
Modulation delay	*zero*	Sync Bug	*off*
Vibrato frequency	*mid*	Voice Assign	*first*
Vibrato amount	*zero*	Envelope Routing	*standard*
Unison detune	*zero*		
External CV amount	*zero*		
Vintage / Spread	*zero*		

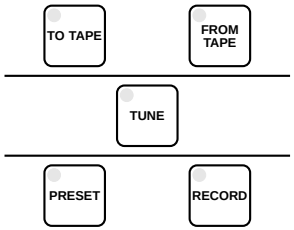
5.3 Settings parameter overview

Number	Purpose	1st press	Repeated press
0	MIDI mode	Show current selected voice status	Toggle MIDI mode *local on, local off*
1	MIDI Receive	Show current MIDI receive channel	Cycle through channels *OMNI, Ch1 ... Ch16*
2	MIDI Send	Show current MIDI send channel	Cycle through channels *Ch1 ... Ch16*
3	Bender Calibration	Activate calibration mode	Confirmation
4	Voice Selection	Show current selected voice	Cycle through voices *1 ... 6*
5	Voice Deactivate	Show current selected voice status	Toggle voice status *on, off*
6	Panel Layout	Show current selected panel layout	Toggle panel layout *GliGli, SCI*
7	Unused		
8	Clock Sync	Show current clock sync	Cycle through sync types *internal, MIDI, tape*
9	VCF limit	Show current setting	Cycle through settings *on, off*
Preset	Default Patch	Activate default patch load	Confirmation
Tune	Per Note Tuning	Activate per note tuning	Deactivate/save per note tuning
Record	MIDI Patch Receive Mode	Activate MIDI patch receive mode	End MIDI patch receive mode

5.4 Operation modes

Live mode

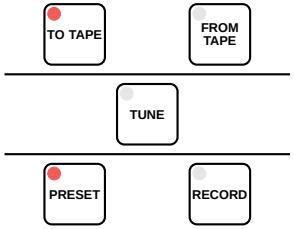
Live mode:



To Tape	
From Tape	Shift and shift-lock modes (double press)
Tune	Start tuning procedure
Preset	Leave live mode and enter preset patch mode
Record	Enter record mode
Number pad	Select patch parameter
Display	Show parameter values
Keyboard	(normal function)
Performance	(normal function)
Panel controls	(normal function)

Preset panel mode

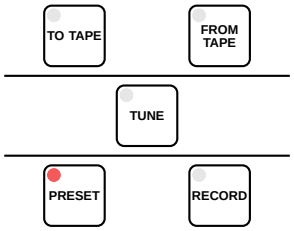
Preset panel mode:



To Tape	Enter preset patch mode
From Tape	Shift and shift-lock modes (double press)
Tune	Start tuning procedure
Preset	Leave preset mode and enter live mode
Record	Enter record mode
Number pad	Select patch parameter
Display	Pick-me-up display (value or relative position)
Keyboard	(normal function)
Performance	(normal function)
Panel controls	Pick-me-up function

Preset patch mode

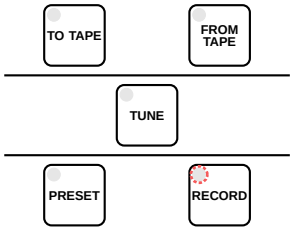
Preset patch mode:



To Tape	Enter preset panel mode
From Tape	Shift and shift-lock modes (double press)
Tune	Start tuning procedure
Preset	Leave preset mode and enter live mode
Record	Enter record mode
Number pad	Enter patch number
Display	Display patch number
Keyboard	(normal function)
Performance	(normal function)
Panel controls	(normal function)

Record mode

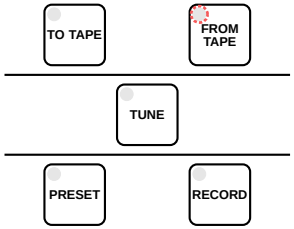
Record mode:



To Tape	
From Tape	(normal function)
Tune	(normal function)
Preset	Leave record mode
Record	Leave record mode
Number pad	Enter patch number
Display	Display patch number
Keyboard	(normal function)
Performance	(normal function)
Panel controls	(normal function)

Shift-lock mode

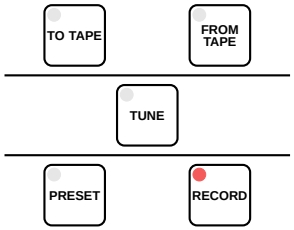
Shift-lock mode:



To Tape	
From Tape	Leave shifted mode
Tune	Enter per note tuning mode
Preset	Load default patch (requires confirmation)
Record	Enter patch management mode
Number pad	Miscellaneous settings
Display	Display of miscellaneous settings and messages
Keyboard	Transposition
Performance	(normal function)
Panel controls	(normal function)

Sequencer editing mode

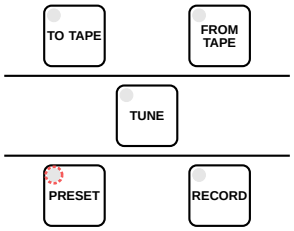
Sequencer editing mode:



To Tape	Hold to access patch parameters via number pad
From Tape	(normal function)
Tune	(normal function)
Preset	
Record	Leave editing mode
Number pad	Special functions of buttons 0, 1 and 2
Display	Display the number of steps
Keyboard	Enter steps in the sequence
Performance	(normal function)
Panel controls	(normal function)

Patch management mode

Patch management mode:



To Tape	
From Tape	(normal function)
Tune	(normal function)
Preset	Dump entire patch bank
Record	Leave patch management mode
Number pad	Patch number entry
Display	Patch number input, confirmation text
Keyboard	(normal function)
Performance	(normal function)
Panel controls	(normal function)

MIDI control implementation

Two types of MIDI events are implemented in the ,upgraded Prophet-600:

- MIDI CC Continuous parameters: value resolution of 0-16383 using 2 CCs (fine and coarse), or value resolution of 0-127 using only the coarse one
- MIDI CC Stepped parameters: 0-127, variable number of steps. They work by dividing the 0-127 range in as many zones as there are choices for the parameter. E.g.: ”Unison” is an on-off parameter and therefore has s choices. In this case it is *off* for 0-63 and it is **on** for 64-127.
- Performance events (note events, pitch bend)
- Real time events
- SysEx (for different purposes)

The Prophet-600 receives Continuous Controllers (CC) which correspond to patch parameters in **Preset Mode** only.

6.1 MIDI CC Patch Parameters

Continuous Parameter	Type	CC Coarse	CC Fine	Stepped Parameter	Type	CC
Osc A Frequency	Continuous	16	80	Osc A Saw	Stepped	48
Osc A Volume	Continuous	17	81	Osc A Triangle	Stepped	49
Osc A Pulse Width	Continuous	18	82	Osc A Square	Stepped	50
Osc B Frequency	Continuous	19	83	Osc B Saw	Stepped	51
Osc B Volume	Continuous	20	84	Osc B Triangle	Stepped	52
Osc B Pulse Width	Continuous	21	85	Osc B Sqr	Stepped	53
Osc B Fine	Continuous	22	86	Sync	Stepped	54

Continuous Parameter	Type	CC Coarse	CC Fine	Stepped Parameter	Type	CC
Cutoff	Continuous	23	87	Poly Mod Oscillator A Destination	Stepped	55
Resonance	Continuous	24	88	Poly Mod Filter Destination	Stepped	56
Filter Envelope Amount	Continuous	25	89	LFO Shape	Stepped	57
Filter Release	Continuous	26	90	LFO Targets	(see 6.3)	59
Filter Sustain	Continuous	27	91	Keyboard Filter Tracking	Stepped	60
Filter Decay	Continuous	28	92	Filter Envelope Shape	Stepped	61
Filter Attack	Continuous	29	93	Filter Envelope Fast/Slow	Stepped	62
2nd Release	Continuous	30	94	2nd Envelope Shape	Stepped	63
2nd Sustain	Continuous	31	95	Unison	Stepped	65
2nd Decay	Continuous	32	96	Assigner Priority Mode	Stepped	66
2nd Attack	Continuous	33	97	Bender Range (semitones)	(see 6.3)	67
Poly Mod Filter Amount	Continuous	34	98	Bender Target	Stepped	68
Poly Mod Osc B Amount	Continuous	35	99	Mod Wheel Range	Stepped	69
LFO Frequency	Continuous	36	100	Osc pitch mode	Stepped	70
LFO Amount	Continuous	37	101	Mod Wheel Target	Stepped	71
Glide	Continuous	38	102	Vibrato Target	Stepped	72
Amp Velocity	Continuous	39	103	2nd Envelope Fast/Slow	Stepped	73
Filter Velocity	Continuous	40	104	Sync Bug	Stepped	74
Modulation delay	Continuous	41	105	Voice Assign	Stepped	75
Vibrato frequency	Continuous	42	106	Envelope Routing	Stepped	76
Vibrato amount	Continuous	43	107	LFO Sync	Stepped	77
Unison detune	Continuous	44	108			
External CV amount	Continuous	46	110			
Vintage / Spread	Continuous	47	111			

6.2 MIDI CC Settings

Unlike MIDI CC event for patch parameters, which are only applied in **Preset Mode**, the following MIDI CC controls are always applied.

Parameter	Type	CC	CC Fine	Comment on CC value
Toggle Live/Preset Mode	Stepped	0		0 = live mode, 1 = preset mode
Mod Wheel	Continuous	1		
Master Volume	Continuous	7		
Sequencer / arpeggiator clock	Continuous	45	109	
Hold Pedal	Stepped	64		0 = release, >0 = hold/latch
All Sound Off	Stepped	120		127 = sound off
Local On/Off	Stepped	122		<=63 = local on, >63 = local off
All Notes Off	Stepped	123		0 = all notes off

6.3 Special parameters and technical information

LFO Targets (MIDI CC 59)

MIDI CC 59 LFO Target is a stepped parameter. It combines the effects of the LFO panel controls (separate switches for targeting the VCO, the filter cutoff and the pulse width) with the additional patch parameter which provides further options (targeting VCA, targeting oscillators A and B separately). These options are encoded bitwise in the parameter as shown in the table below. Each bit represents *off* or *on* for the respective switch/option.

Bit	Value	Function
1	1	VCO
2	2	Filter
3	4	VCA
4	8	Pulse Width
5	16	Only A
6	32	Only B

Example: if the filter and the VCO targets are active and the modulation is restricted to oscillator B then the parameter takes the following value:

LFO Targets (MIDI CC 59) = 1 (VCO) + 2 (filter) + 32 (only B) = 35

Note that the same parameter encoding is also used in the patch MIDI SysEx implementation. Not all combinations of bits are possible (at least when edited on the unit) and meaningful. Some can only be achieved using MIDI CC or SysEx.

- The options ”Only A” and ”Only B” only affect the targets VCO and pulse width (VCA and filter are shared by the two oscillators)
- When editing a patch from the Prophet-600 it is impossible to set bits 5 and 6 simultaneously. Technically the effect of setting both bits (by sending such a patch configuration to the Prophet-600 via SysEx or MIDI CC) would be that VCO and pulse width are modulated for neither oscillator even if those targets are activated.
- When editing a patch from the Prophet-600 it is impossible to activate the VCA target and at the same time restrict the VCO or pulse width modulation to one of the oscillators. This is purely due to user interface restrictions. It is perfectly possible to choose this option using MIDI CC or SysEx.

Bender Range (semitones) (MIDI CC 67)

The MIDI CC 67 parameter which controls the bender range behaves like a stepped parameter. However, unlike other stepped parameters whose value corresponds to the index of the selected option, the value of the bend range parameter contains the actual number of semitones. The allowed values and the corresponding option names as shown on the Prophet-600 display are as follows.

Value	Text
2	2nd
3	3rd
5	5th
8	Octave

No other values are allowed.

Changes in MIDI CC implementation from version 2.1 RC3 to v2022

The following MIDI CC have been changed.

- For the filter envelope release and decay and the 2nd envelope release and decay the time value range has been rescaled for the *linear* envelope shape. This was an unavoidable side effect of changing the strictly linear shapes to linear shapes with soft tails. The exponential envelopes are unaffected.
- The **(LFO) Frequency** has been totally rescaled. The stepped frequency range parameter has been omitted and the value range of the **(LFO) Frequency** dial now covers the entire frequency range. However, the dial to frequency relation has also been changed from linear to exponential, making it much softer at low settings. Consequently, the MIDI CC support for the LFO frequency range (up to firmware version 2.1 RC3) has been omitted. In order to avoid potential conflicts with MIDI tools developed for firmware versions 2.0 and 2.1 RC3, the legacy MIDI CC 58 has not been re-designated. MIDI CC 58 is ignored.
- The value to frequency relation for the **(2) Vibrato Speed** has been changed from linear to exponential.
- The **(333) Modulation Wheel Range** has been reconfigured with new value which also produce different wheel actions.

Bibliography

[1] Latest firmware project page (to follow)

[2] GliGli blog and project pages, http://gliglisynth.blogspot.com/, http://gligli.github.io/p600fw/

[3] Imogen Synth blog and project pages, https://prophet600revisited.blogspot.com/, https://github.com/image-etson/p600fw

[4] Curtis Electro Music on Webarchive, https://web.archive.org/web/20130502132536/http://curtiselectromusic.com/Customers_and_Instruments.html

[5] Prophet-600 Modification Instructions, see GliGli project pages and links above

[6] Teensy++ 2.0 manufacturer information: http://www.pjrc.com/teensy/index.html

[7] Prophet-600 Owner's Manual, Sequential Circuits

[8] Prophet-600 Service Manual, Sequential Circuits

[9] Synthgraphics (http://www.synthgraphics.com/). The provider of panel overlays for modded synthesizers created an overlay for the GliGli firmware panel layout of the Prophet-600.

[10] Teensy code loader application from pjrc, https://www.pjrc.com/teensy/loader.html

[11] Prophet-600 Spirit blog, http://prophet600.blogspot.com/