the Deputy

Mark II

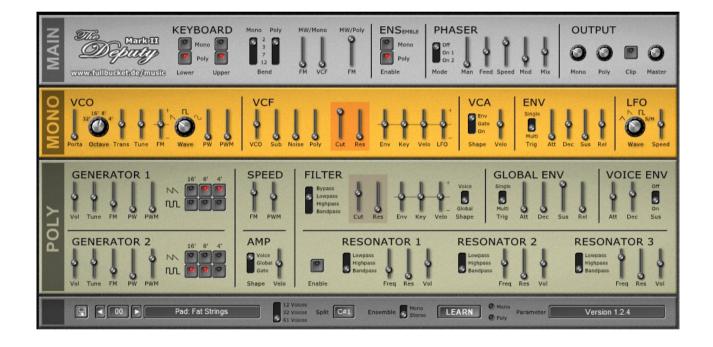### Polyphonic Ensemble

Version 1.2

© 2012-2020 by Björn Arlt @ Full Bucket Music
http://www.fullbucket.de/music

Table of Contents

Introduction

The *Deputy Mark II* is a software instrument plug-in for Microsoft Windows (VST) and Apple macOS (VST/AU) in the tradition of classic string machines and early "polyphonic" synthesizers, written in native C++ code. The main features are:

- Polyphonic string/synthesizer (*Poly*) section and monophonic synthesizer (*Mono*) section
- Built-in Ensemble and Phaser effects
- Flexible keyboard assignment for *Poly* and *Mono* section
- *Poly* section:
  - Up to 61 voices polyphony
  - Two band-limited frequency divider-driven oscillator banks
  - Sawtooth and PWM waveforms
  - 2-pole multi-mode filter and ADS envelope per voice
  - Global ADSR envelope for paraphonic modulation
  - Resonator bank (3 zero-delay feedback filters)
- *Mono* section:
  - Single VCO plus suboscillator and noise
  - 4-pole zero-delay feedback lowpass filter
  - ADSR envelope and LFO
  - Output of *Poly* section can be send to filter input
- Double precision audio processing
- All parameters can be controlled by MIDI controllers
- Skinning support
- Plug-in supports Windows and macOS (32 bit and 64 bit)

The *Deputy* emulates an instrument that *might have been built* this way in the 1970s. Due to technical and economical reasons it would have been a compromise between which features a musician expected, what he was willing to pay for it, and what fitted in a single box. The Digital Revolution already lured around the corner and manifested itself in frequency dividers and sophisticated keyboard controllers, but *sound* was still shaped in the good old analog fashion (if you ignore some ingenious machines like the PPG *Wave Computer* for the moment). As a conclusion, the *Deputy* is (technically speaking, and as to nowadays standards) a fairly limited instrument. However, I strongly believe that technical limitation can be a source of great creativity. Prost!

Architecture

The *Deputy* consists of three major blocks: the polyphonic *Poly* synthesizer section, the monophonic *Mono* synthesizer section, and the *Main* section hosting global effects, the mixer, and keyboard-related controls. Apart from that the *Deputy Mark II* has an additional section containing controls for program management and "tweak" parameters.

Each voice of the *Poly* section features two "generators" feeding a 2-pole multi-mode filter. The generators are driven by independent master oscillators and deliver sawtooth and/or pulse signals derived from frequency divider banks. Two simple LFOs are used for frequency and pulse width modulation. The amplitude and/or the filter cutoff can be controlled by an ADS envelope (per voice) or by an additional global ADSR envelope (for all voices). A resonator bank consisting of three 4-pole multi-mode filters is available to process the signal of the *Poly* section.

The *Mono* section features a single VCO plus suboscillator and white noise, a 4-pole lowpass VCF, an ADSR envelope, and a LFO offering four waveforms. The output of the *Poly* section can be sent to the VCF input as well.

Finally, in the *Main* section the *Poly and Mono* parts can be assigned to one or both of the two keyboard ranges while pitch and modulation wheels can be routed to several modulation targets. Furthermore, an Ensemble and a Phaser effect are available as well as an overall output mixer.

The *Poly* Section

The *Poly* section is the heart of the *Deputy Mark II:* A fully polyphonic synthesizer (up-to 61 voices) plus a three-stage resonator bank. Its architecture resembles that of many well-known instruments like the *Polymoog* or the Korg *PS* series which also used electronic organ and classic string machine technology.

Generator Banks

The *Poly* synthesizer section follows a common, yet (at that time) highly discussed pattern of 1970s polyphonic keyboard designs: The tone signals are not generated directly by individual oscillators but by using overall "Master" oscillators driving a cascade of frequency dividers. While this design is rather cost-effective — in this case only one Master oscillator and a "Top-Octave Synthesizer" followed by octave dividers are needed to create a *fully polyphonic* generator bank — it also means that the waveform signals of different octaves of the same *key* (i.e. *C* to *B*) are totally correlated. Purists never stopped to point out that such an arrangement sounds "cold" and "unnatural", many even simply rejected it because this technology was used in (and originally developed for) electronic organs. Some 40 years later purists will assure you that this is *absolutely necessary* to reproduce the "warm" tone of, for example, famous string machines like the Eminent *Solina* etc.

A nice feature of such generator banks is that one gets the upper and lower octaves of a played key (almost) "for free" — think of choirs in (again) electronic organs. The *Poly* section provides two different waveform outputs (sawtooth and pulse) and three such octave signals (16', 8', and 4') which are individually selectable per waveform. To mitigate the static character of the generated signal it is possible to modulate the width of the pulse waveform by a dedicated LFO. However, the "real synthesizer feeling" is introduced by a second, identical generator bank which can be (de)tuned with respect to the first bank; this at least produces the famous "beating" when signals undergo a periodical phase cancellation.

Finally, frequency modulation of the generator banks by a separate LFO (vibrato) is available as well.

Filter

In contrary to other early polyphonic synthesizers (e.g. the *Polymoog* but also the later Korg *Poly 800*), the *Poly* section offers a multi-mode filter (lowpass, highpass, bandpass) *per voice*. Thus, each voice (in fact each of the 61 keys!) has its own filter in place — an enormous amount of additional hardware! To reduce the costs, the filter layout follows a simple 2-pole design which is *not* capable of self-oscillation (today an absolute *No-Go*). The filters' cutoff frequency can be modulated by key tracking, velocity (an uncommon feature in the 1970s), and the *Voice* or *Global* envelopes (see below). Here, it should be mentioned that both positive and negative modulation is possible.

Envelopes

There is a dedicated envelope per voice (the *Voice* envelope) which can be used to modulate the amplifier and/or the filter. To save costs, this envelope has a reduced control set, namely *Attack* and *Decay* plus a switchable *Sustain* level (*On* or *Off*) where *Decay* also controls the release time.

An additional *Global* envelope featuring the full ADSR control set is available as well. However, there is only one such envelope for the whole *Poly* section resulting in a *paraphonic* playing mode. Thus, the trigger mode can be changed from *Single* (a new note only triggers the envelope when all previous notes have been released) to *Multiple* (the envelope is triggered for each new note).

Amplifier

The amplifier section offers modulation by velocity (!) and a selectable envelope (*Voice*, *Global*, and *Gate*, a simple rectangular-shaped envelope). Again, there is one amplifier per voice, a luxury and only possible due to progress in large scale circuit integration.

Resonators

The *Deputy* is equipped with three independent 4-pole multi-mode filters (lowpass, highpass, bandpass) working in parallel mode on the overall output of the *Poly* section. Although it is not possible to modulate their cutoff frequency, these *Resonators* become very handy when it comes to create fixed formants.

The *Mono* Section

Many keyboards of the 70s were in fact "assemblies" of multiple instruments, e.g. String, Brass, Synthesizer, and Organ sections combined in one box, sharing a common keyboard controller; examples are the ARP *Quadra*, the Moog *Opus*, but also the Siel *Trilogy*, to name a few. Along with the *Poly* section the *Deputy* "just" features an additional monophonic synthesizer, the *Mono* section. Following a very basic VCO/VCF/VCA design, there is nothing spectacular about this little buddy except that the output of the *Poly* section can be routed through the *Mono*'s VCF as well.

VCO

The VCO offers three waveforms (sawtooth, pulse, sine) in four octaves (32', 16, 8', 4'). The pulse width can be modulated by the LFO, and it is also possible to tune the VCO in a range of ±7 semitones.

VCF

This 4-pole lowpass filter is capable of self-oscillation and takes the outputs of the VCO, the rectangular suboscillator (one octave below the VCO frequency), the white noise generator, and the *Poly* section as input signals. Cutoff frequency will be positively or negatively modulated by velocity, keyboard tracking, LFO, and the ADSR envelope.

VCA

To modulate the amplitude of the *Mono* synthesizer, one can select the ADSR or a simple gate envelope. Furthermore, it is possible to set the VCA into a permanent "On" mode; this becomes handy when the *Mono* section is configured to process only the output of the *Poly* section. Last but not least, the VCA can be modulated by velocity.

ENV and LFO

The ADSR envelope of the *Mono* section is identical to the *Global* envelope of the *Poly* section and is used as a modulation source for the VCF and the VCA.

The LFO features four waveforms (triangle, sawtooth, rectangle, sample/hold) which can modulate the VCO frequency and pulse width and/or the VCF cutoff.

The *Main* Section

This section provides common functionality for both the polyphonic and the monophonic synthesizer sections: keyboard controls, effects, and mixer.

Keyboard

The keyboard of the *Deputy* has 61 keys (C-1 to C-6), and each key makes up one individual voice (one can decrease the number of total voices in the plug-in version to reduce computational load). Thus, only MIDI keys C-1 to C-6 generate a sound when played.

The keyboard is divided into two *zones*, the *Lower* and the *Upper* zone — the split point between both regions was originally fixed to C-3 but is variable in the plug-in version). The *Poly* and the *Mono* sections can be assigned to one or both of these zones independently, allowing for split, layered, or combinations of both modes.

The amount of pitch wheel modulation (2, 3, 7 or 12 semitones) is selectable for both sections as well; this also holds for the modulation wheel which controls the intensity of vibrato and LFO respectively. Furthermore, the modulation wheel can be assigned to manipulate the VCF cutoff frequency.

Ensemble

Many classic string machines like the Eminent *Solina* are famous (or infamous) for their modulation effects (often called *Chorus*, *Ensemble*, or simply *Modulation*). While these effects are very important for the typical sound characteristics, the original rationale for integrating them into such an instrument was to beef up the rather static output of the single generator bank. In contrary, the *Polymoog* (like the *Deputy*) features two individual generator banks with a detuning option resulting in a "fat" sound. However (and unlike the *Polymoog*), the *Deputy* does have a built-in *Ensemble* effect that can be individually activated for both the *Poly* and the *Mono* section.

Phaser

The *Phaser* effect modulates the sum of both section outputs. Implemented by a classic 4-pole allpass filter the *Phaser* comes with two different modes and features feedback control plus a separate LFO for modulation.

Output

Last but not least, the *Output* section sums up the individual *Mono* and *Poly* output signals. There is also a very simple signal limiter (called *Clip*) which in the signal path lies before the master volume control. However, for "loud" signals this circuit works more like an overdrive effect.

General Controls

Some general controls live in a section at the very bottom of the *Deputy*'s user interface. Besides the program selector there are also some "tweak" parameters: The number of *Poly* voices (12, 32, 61) to reduce computational load, the split point between the keyboard zones, and the *Ensemble* mode (mono or stereo). Furthermore, the MIDI Learn switch (see section *MIDI Learn*) is followed by two LEDs that report the status of the *Mono* and *Poly* sections (on or off).

A quite handy tool is the parameter display: At start-up it displays the software version while as soon as one hoovers over a parameter the respective value is shown.

Program and File Menu

When clicking on the *File* button (disk symbol on bottom left of the screen), a context menu opens with the following options:

Copy Program	Copy current program to internal clipboard
Paste Program	Paste internal clipboard to current program
Load Program	Load a program file containing a patch to the *Deputy's* current program
Save Program	Save the *Deputy's* current program to a program file
Load Bank	Load a bank file containing 64 patches into the *Deputy*
Save Bank	Save the *Deputy's* 64 patches to a bank file
Init Program	Initialize the current program
Reload Configuration	Reload the *Deputy's* configuration file (see section *The deputy.ini Configuration File*)
Save Configuration	Save the *Deputy's* configuration file (see section *The deputy.ini Configuration File*)
Select Startup Bank	Select the bank file that should always be loaded when the *Deputy* is started
Load Startup Bank	Load the Startup bank file; can also be used to check what the current Startup bank is
Unselect Startup Bank	Unselect the current Startup bank
Check Online for Update	When connected to the Internet, this function will check if a newer version of the *Deputy* is available at fullbucket.de
Alternative Skin	Switch between alternative built-in skins
Visit fullbucket.de	Open fullbucket.de in your standard browser

The deputy.ini Configuration File

The *Deputy* is able to read some settings from a configuration file (`deputy.ini`) located in the same directory as the *Deputy* VST DLL (`deputy.dll` or `deputy64.dll`) or Mac VST/AU (`Deputy.component` or `Deputy.vst`) itself. After you have edited this INI file in a text editor, you have to reload it using the *Reload Configuration* command from the *File* menu (see section *Program and File Menu*).

Skinning Support

The *Deputy* features some very rudimentary skinning support. Within the INI file, each of the bitmaps used in the UI can be replaced by specifying the full path to an alternative bitmap file; bitmaps that you don't want to replace can be set to default (see example below), or left out of the INI file completely.

```
[Editor]
Bitmap_Back = default
Bitmap_AltBack = default
Bitmap_Handle = default
Bitmap_Button = default
Bitmap_Knob = default
Bitmap_Switch1 = default
Bitmap_Switch2 = default
...
```

Important: The replacing graphics need to have the same size and control placement to work correctly. Also knob/slider graphics need to have the exact same size and number of frames to work correctly. It is not possible to alter the placement of controls, as that is handled by code in the plug-in.

MIDI Control Change Messages

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

```
[MIDI Control]
# trying to follow GM2 definitions here ;-)
CC7  = 0  # Volume
CC70 = 37 # Poly Filter Mode
CC71 = 39 # Poly Filter Resonance
CC72 = 61 # Poly Voice Decay/Release
CC73 = 60 # Poly Voice Attack
CC74 = 38 # Poly Filter Cutoff
CC75 = 62 # Poly Voice Sustain
CC76 = 13 # Poly FM Rate
...
```

The syntax is straight forward:

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

Given the above example, controller 7 directly controls the overall *Volume* parameter, controller 74 the *Poly Filter Cutoff* etc. As you can see, comments are introduced by

the Pound sign (#); they are here just for description purposes and completely optional.

The *parameter ID* of one of the *Deputy's* parameters is given in the section *Parameters* below. Note that the *controller number* can run from 0 to 110, with the exception of 1 (*Modulation Wheel*) and 64 (*Sustain Pedal*); the latter two are simply ignored.

MIDI Learn

The easiest way to assign MIDI controllers to *Deputy* parameters is to use the MIDI Learn function. To activate *MIDI Learn*, click on the respective button and wiggle both the MIDI controller and the *Deputy*'s parameter that you want to link. If you want to unlearn the assignment, right-click the *MIDI Learn* button (the label now reads "UNLEARN") and activate it. Now wiggle the MIDI controller or the parameter that you want to unlearn.

Parameters

Main — Keyboard

parameter	ID	description
Lower/Mono	4	activates *Mono* section for *Lower* keyboard zone
Lower/Poly	3	activates *Poly* section for *Lower* keyboard zone
Upper/Mono	6	activates *Mono* section for *Upper* keyboard zone
Upper/Poly	5	activates *Mono* section for *Upper* keyboard zone
Bend/Mono	8	pitch bend amount (± 2, 3, 7, 12 semitones) for *Mono* section
Bend/Poly	7	pitch bend amount (± 2, 3, 7, 12 semitones) for *Poly* section
MW/Mono FM	10	mod wheel amount for frequency modulation of *Mono* section
MW/Mono VCF	11	mod wheel amount for cutoff frequency of VCF (*Mono* section)
MW/Poly FM	9	mod wheel amount for frequency modulation of *Poly* section

Main — Ensemble

parameter	ID	description
Enable/Mono	97	activates *Ensemble* effect on *Mono* section
Enable/Poly	96	activates *Ensemble* effect on *Poly* section

Main — Phaser

parameter	ID	description
Mode	98	*Phaser* effect mode (*Off*, *On – Mode 1*, *On – Mode 2*)
Man	99	manual setting (i.e. center frequency)
Feed	100	feedback
Speed	101	modulation speed
Mod	102	modulation amount
Mix	103	dry/wet balance

Main — Output

parameter	ID	description
Mono	86	volume of the *Mono* section
Poly	46	volume of the *Poly* section
Clip	1	controls if the overall output is unlimited or clipped to unity
Master	0	overall volume

Mono — VCO

parameter	ID	description
Porta	68	portamento time (0 to 5 seconds)
Octave	69	octave (32', 16', 8', 4')
Trans	70	transpose (0 to ±7 semitones)
Tune	71	fine tuning (±1 semitone)
FM	75	amount of frequency modulation by LFO
Wave	72	waveform (sawtooth, pulse, sine)
PW	73	pulse width
PWM	74	amount of pulse width modulation by LFO

Mono — VCF

parameter	ID	description
VCO	76	volume of VCO signal
Sub	77	volume of suboscillator signal (rectangle, -1 octave)
Noise	78	volume of white noise signal
Poly	79	volume of *Poly* section output signal
Cut	80	cutoff frequency
Res	81	resonance (up to self-oscillation)
Env	82	amount of cutoff frequency modulation by envelope
Key	83	amount of cutoff frequency modulation by keyboard tracking
Velo	84	amount of cutoff frequency modulation by velocity
LFO	85	amount of cutoff frequency modulation by LFO

Mono — VCA

parameter	ID	description
Shape	88	envelope selection (*Env*, *Gate*, *On*)
Velo	87	amount of amplitude modulation by velocity

Mono — Env

parameter	ID	description
Trig	89	envelope trigger selection (*Single*, *Multiple*)
Attack	90	attack time
Decay	91	decay time
Sustain	92	sustain level
Release	93	release time

Mono — LFO

parameter	ID	description
Wave	94	LFO waveform (triangle, sawtooth, rectangle, Sample & Hold)
Speed	95	LFO speed (0 to 30 Hz)

Poly — Generator 1 & 2

parameter	ID	description
Vol	15; 26	volume of the generator bank
Tune	16; 27	fine tuning (±1 semitone)
FM	19; 30	amount of frequency modulation
PW	17; 28	pulse width
PWM	18; 29	amount of pulse width modulation
Saw 16'	20; 31	activates the *16' Sawtooth* register
Saw 8'	21; 32	activates the 8' *Sawtooth* register
Saw 4'	22; 33	activates the 4' *Sawtooth* register
Pulse 16'	23; 34	activates the *16' Pulse* register
Pulse 8'	24; 35	activates the 8' *Pulse* register
Pulse 4'	25; 36	activates the 4' *Pulse* register

Poly — Speed

parameter	ID	description
FM	13	speed of frequency modulation (0 to 30 Hz)
PWM	14	speed of pulse width modulation (0 to 30 Hz)

Poly — Amp

parameter	ID	description
Shape	44	envelope selection (*Voice*, *Global*, *Gate*)
Velo	45	amount of amplitude modulation by velocity

Poly — Filter

parameter	ID	description
Mode	37	filter mode (bypass, lowpass, highpass, bandpass)
Cut	38	cutoff frequency
Res	39	resonance
Env	41	amount of cutoff frequency modulation by envelope
Key	41	amount of cutoff frequency modulation by keyboard tracking
Velo	43	amount of cutoff frequency modulation by velocity
Shape	40	envelope selection (*Voice*, *Global*)

Poly — Global Env

parameter	ID	description
Trig	63	envelope trigger selection (*Single*, *Multiple*)
Attack	64	attack time
Decay	65	decay time
Sustain	66	sustain level
Release	67	release time

Poly — Voice Env

parameter	ID	description
Attack	60	attack time
Decay	61	decay/release time
Sustain	62	sustain level (on or off)

Poly — Resonators

parameter	ID	description
Enable	47	activates the resonators
Mode 1–3	48;52;56	filter mode (lowpass, highpass, bandpass)
Cut 1–3	50;54;58	cutoff frequency
Res 1–3	51;55;59	resonance
Vol 1–3	49;53;57	volume

General Controls

parameter	ID	description
Program	—	program selector (programs 1 – 64)
Voices	12	number of *Poly* voices (12, 32, 61)
Split	2	split point between *Lower* and *Upper* zone
Ensemble	104	*Ensemble* mode (mono or stereo)
Knob	—	knob mode (linear or circular)

Hidden Controls (*Ensemble*)

parameter	ID	description
LFO1 Speed	105	Speed of *Ensemble* LFO 1
LFO2 Speed	106	Speed of *Ensemble* LFO 2
Mod. Amount 1	107	Modulation amount *Ensemble* for delay line 1
Mod. Amount 2	108	Modulation amount *Ensemble* for delay line 2
Mod. Amount 3	109	Modulation amount *Ensemble* for delay line 3

Frequently Asked Questions

How do I install the Deputy *(Windows 32 bit version)?*

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

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

Just copy the file `deputy64.dll` and `deputy.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST plug-in folder. Your DAW should automatically register the *Deputy* VST plug-in the next time you start it.

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

How do I install the Deputy *(Mac VST/AU universal 32/64 bit)?*

Locate the downloaded PKG package file `deputy_1_2_4_mac.pkg` in Finder (!) and do a right- or control-click on it. In the context menu, click on "Open". You will be asked if you really want to install the package because it comes from an "unidentified developer" (me ☺). Click "OK" and follow the installation instructions.

What is the plug-in ID of the Deputy*?*

The ID is **T D M 2** .

How can I decrease the Deputy's *CPU load?*

Whenever it does not degrade the *sound* you need, try this:

- Reduce polyphony, i.e. the number of voices.
- Deactivate a section, an effect, or the resonators.
- If you don't need a waveform/octave to be generated by one or both generator bank, switch off the respective buttons.

How can I prevent output clipping?

Decrease the sections' *Volume* levels. Alternatively, switch on the *Clip* parameter in the *Main* section but that might lead to distortion effects. Hey, maybe that's exactly what you need? ;-)

There is no sound when I play a key below C-1 or above C-6?

This is by design: The *Deputy* has 61 keys, each making up a complete voice of the instrument. For keys beyond this range there is no voice left over.

Can I tweak the Ensemble effect parameters?

Yes, but not from the *Deputy*'s front panel. Like on the old *Solina* thy shall seek and thy will be given.

How do I know if a new version of the Deputy *is available?*

When connected to the Internet, open the *Program and File Menu* by clicking the disk icon and select the entry "Check Online for Updates". If a new version of the *Deputy Mark II* is available on fullbucket.de the respective information will be shown in a message box.

Your skins suck! How can I change them to something flashy?

By making your own bitmaps and editing the `deputy.ini` — see *The deputy.ini Configuration File*.

Seriously, the Deputy *has never existed as a real "hardware" instrument…!?*

I never said it would! The *Deputy Mark II* is an emulation of a *fictive* polyphonic synthesizer from the 1970s of an alternative history. But the point is that an instrument like this *could have been built* just this way — see the appendix.

So why did you create the Deputy*?*

Because I love those weird machines with their strange concepts and technical designs. They tell us where the sophisticated instruments of today originally come from blablabla. And because it's fun!

What does "Prost!" mean?

You do not know this famous German word? It means something like "To your health!". Typically, German words tend to be indefinitely longer than their English counterparts but in this case the Germans came to the point.