

Ruismaker FM

Bram Bos



User Manual
version 2.0

Version history

Version history (this document's state reflects the latest available software version:

2.0 August 21, 2017 Initial publication

Introduction

Ruismaker FM is like the crazy twin-sister to the classic Ruismaker drum synth. Instead of using virtual analog modelling, it sports a single, custom FM ('Frequency Modulation') synthesis engine capable of creating a vast bandwidth of different types of sounds.

The theory behind FM is too complex to go into right here, but essentially it works by taking an oscillator (called 'the carrier', like a sine wave or a pulse wave) and changing its frequency so fast that all sorts of overtones - known as 'sidebands' - become audible in the signal. Although FM can be done using analog equipment, the most effective way to do it is using digital algorithms (analog's charming quirkiness tends to introduce side effects that sometimes makes FM hard to control properly in a musical way).

Ruismaker FM uses this type of digital FM synthesis to create percussive sounds. It is capable of creating powerful synth kicks, noisy snares and metallic hihats, but truly comes to life when doing more unconventional sounds (zaps, blips, whirrs, clicks, hollow or metallic effects) or even bass lines.

Because it is designed to do percussion, it comes with some specific parameter behavior and waveforms that are optimised for percussive sounds and effects - typically not found on other FM synths.

Euclidean Sequencer

With the introduction of version 2.0 of Ruismaker and Ruismaker FM, we have added a cool standalone feature to the existing plugin: a Euclidean sequencer. The aim of the sequencer is to help you come up with drum patterns you would normally not come up with. We hope you'll enjoy this playful addition to the app.

1. The Audio Unit / Drum Synth Engine

We'll first dive into the controls of the synth engine (which constitutes the AU plugin part of the app).

1.1 Basic Parameters

The basic parameters can all be set using the on-screen user interface. There are

- channel settings that are specific to each of the 6 available channels,
- and global parameters that apply to the total plugin.



INSTRUMENT SETTINGS ('channel settings')

Oscillator

This is the main waveform responsible for the basic sound of the instrument. If the modulation amount is set to 0 this is the only sound you'll hear. Some waveforms have simple frequency buildup (e.g. a sine wave only covers a single frequency and sounds 'dull' on its own), whereas noise generally covers a wide band of the frequency spectrum like a blanket (more on that later). There are 6 oscillator waves to choose from.

Pitch

This is the frequency at which the oscillator waveform will be played when the sound is triggered. Pitch values are chromatically tuned in Ruismaker FM, making it easy to fit sounds in your music in a musical way (or even to create simple melodic bass lines, etc.).

When 'noise' is selected as the oscillator the pitch value applies a band-pass filter letting you tune the character of the noise signal.

Modulator

This is the waveform of the modulator signal. It determines how the frequency of the oscillator changes. For example, when the modulator wave is a sine wave, the oscillator's frequency will go up and down with a smooth movement. When the modulator type is a square wave, the oscillator's frequency will change from its highest point to its lowest point very abruptly.

There is one special modulator: the pitch drop. When you select this 'wave', the pitch won't go up and down, but only up *or* down. If you want to make a kick drum, this is the modulator you need.

Speed

The speed at which the frequency of the oscillator changes can be adjusted with this control. It can range from low frequency changes (like a pitch LFO, or vibrato) to 'audio-rate' changes, resulting in the typical FM overtones. At very high speeds, digital artefacts can appear, useful for making noise, inharmonic metallic frequencies and other 'chaotic' sound effects.

Amount

Use this to adjust how much the oscillator is modulated by the modulator wave (the 'depth' of the modulation). When the centre value (0) is selected, no modulation is applied and only the oscillator can be heard.

Positive values (0-63) lock the modulation frequency to the pitch of the oscillator. This results in sounds that can be tuned while keeping their overtones constant (e.g. for making toms or bass sounds at different pitches).

Negative values (-1 - -64) keep their speed static. In other words, regardless of the oscillator pitch the speed will be fixed.

When the modulator wave is the "pitch drop", positive amount values will make the pitch drop, and negative amount values will make the pitch rise.

Attack

This parameter changes how the envelope affects the start of the sound. Neutral (0) results in the shortest attack time for the envelope. Negative values increase the attack time; -64 being the longest attack time. Positive values (1-63) keep the attack to its minimal time, but also add a clicky transient to the attack, with increasing strength. For snares you'll want to go all out with this one, and very deep kicks can benefit from a minor click to give them some presence in a track.

Decay

This does what it says on the tin. It applies a decay curve to the sound and the value of this parameter determines how long the decay will take (from very short to pretty-long-for-a-drum-sound).

Saturate

This takes the raw signal (before EG, filters, and VCA are applied) and uses a combination of soft-limiter and overdrive to add high frequency harmonics to the sound.

Pan

Changes the stereo image of the sound.

Level

Controls the output level of the sound.

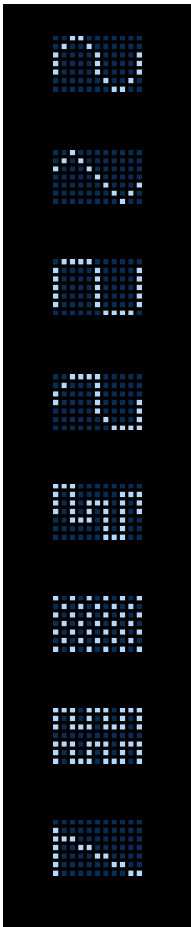
HPF

Applies a high-pass filter (without resonance) to the output signal. This can be used to remove low end frequencies from a sound. The HPF value represent the cutoff frequency of the high-pass filter. Setting it to 0 means no effect (all frequencies pass the filter unaffected).

PKF

This is a peaking filter, which adds strong emphasis to a narrow frequency band. This can be used to boost low end (low values) or stress mid to high end frequency ranges (higher values). Setting it to maximum value (127) effectively disables the filter.

Available wave forms



Sine - the simplest wave form there is, perfect for kick drums and clean sounds*

Triangle - resembles a sine wave, but adds some subtle frequencies*

Square - good for rich, hollow sounds and metallic modulations*

Sine Pulse - a harsh hybrid between a sine wave and a pulse wave*

Sample & Hold - randomises the modulation effect at variable speeds

Noise - covers a wide band of the frequency spectrum*

Metal - complex stream of pulse-waves, useful for generating metallic sounds*

Pitch-drop - lets the modulator drop or raise the oscillator pitch

* these wave forms are available as Oscillator types. All wave forms are available for modulation.

Mini Tutorial 1 - Making a Kick Drum



The most basic kick drum can be made using a sine wave that drops its pitch very quickly. That's what we'll create right now.

First, choose the 'sine' as your oscillator. Choosing other waveforms can lead to special kick sounds, but will add many overtones that you typically don't want for a conventional kick sound.

As discussed before, we want to modulate the oscillator using a pitch drop, so select the pitch drop curve as your modulation shape. The speed value determines how fast the pitch drops and ranges from short

punchy blibs to long drawn-out whines. You'll want the speed to be fairly fast, so crank it up all the way. You can fine-tune later.

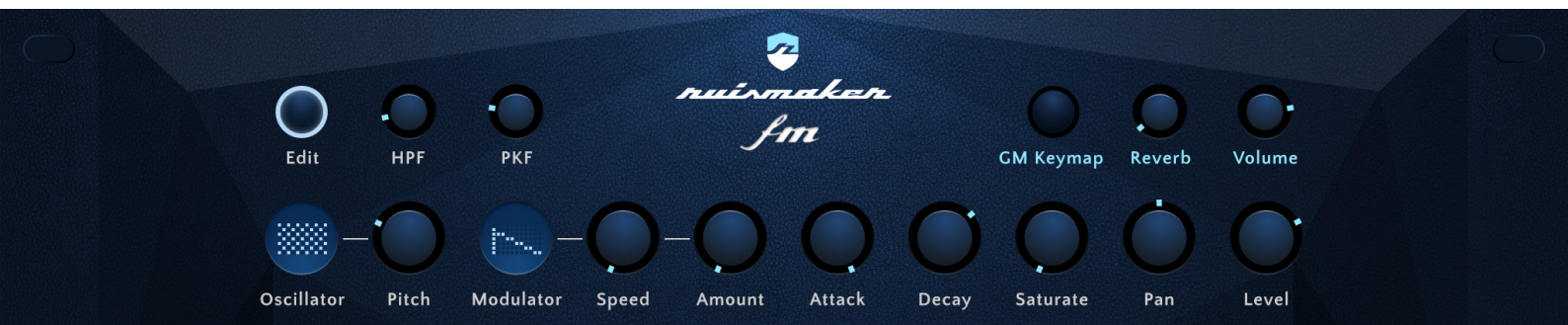
The amount knob does two things: tell the synth engine whether the pitch should go up or down (we want down, obviously) and how much difference there should be between its highest frequency and its lowest frequency. The lowest frequency is determined by the oscillator pitch, so you'll want that to be in the bottom 25% of the range. Now adjust the amount knob to taste.

Next, adjust the attack knob into the positive range and listen what happens (it adds a short 'click' to the attack of the kick). Choose a value that sounds right to you.

The final touch is choosing the Decay length. Again; tune this to what sounds right in your mix. You'll typically want this to be < 50 for a punchy kick drum, but set it by ear to make it fit your project.

A final note: there's a huge interplay between the Pitch, Speed and Amount knobs. Play around with different combinations of the those three knobs to completely change the character of your kick.

Mini Tutorial 2 - Making a Snare Drum



Ruismaker FM's engine offers various methods for creating snaredrums and snaredrum-like sounds. Modulating a sine oscillator with either noise or fast sample-and-hold modulators and cutting the low end with the high pass filter can give very interesting results. I'll let you experiment with that.

The easiest way to make a snappy, 808-like snaredrum in Ruismaker FM is using only its filtered noise. Start by selecting the noise oscillator. Remember that selecting the noise oscillator will actually turn the pitch knob into a bandpass filter? Very useful now! Move the pitch into the 35-40 range and listen what changing the values does to the noise's character.

Another consequence of choosing noise as an oscillator is that you can ignore the entire modulation section. You can't modulate noise in a very meaningful way using FM, so why waste precious CPU power on it? Simply ignore the Modulator, Speed and Amount controls.

Still, the sound you've created so far probably sounds like a dull noise burst. Not exactly what you were looking for. To turn this into a tasty, snappy snaredrum you need to crank the Attack knob to its maximum value. This will make the envelope's attack much more aggressive and adds a strong transient click to the beginning of the noise burst. Presto, we have a fairly convincing synth snare sound.

As a finishing touch you can play around with the Decay length and the HPF/PKF cutoff frequencies to change the personality and snappiness of the sound.

Mini Tutorial 3 - Making a whirring FM effect



Time to do some actual frequency modulation! We're going to make a whirring sound effect, because it will let us tinker with the basics of how frequency modulation gets its signature sound.

To keep things simple we're going to use the sine-wave (the most basic sound in the universe) for both the Oscillator (the 'carrier') and the Modulator. Start with an 'init'-patch and make it so.

Keep the modulator's Amount to 0 for now and choose a moderately high Pitch value (70-ish). Listen to the sound. If you haven't made any other changes, it will be a dull beep. This will not get crowds dancing any time soon.

Now set the modulator's Amount knob to a negative value, like -10 (this will fix the modulation rate to a fixed frequency, set with the Speed knob). Drop the Speed knob to a very low value... say 4. If your decay time is long enough you'll hear the same beep but with a funny pitch wobble applied to it. Yay, we're modulating the frequency of the Oscillator... FM, get it?

Now increase the speed to 40 and listen again. The wobble is still there, but it's now so fast that it has become a sound wave on its own and you don't perceive it as a wobble anymore, but as a metallic audible sound. This is where FM gets interesting, because different modulator waves will result in rather different sound characteristics, even when applied to something as mundane as a sine oscillator.

Back to our whirrrrrrrr..... Decrease the Speed value again to 15. We're now in the intriguing grey area where our modulation speed is transitioning from LFO to audio-rate sound. But because the frequency of the modulator is still quite low for a sound wave there is a bit of low-end rumble in our sound.

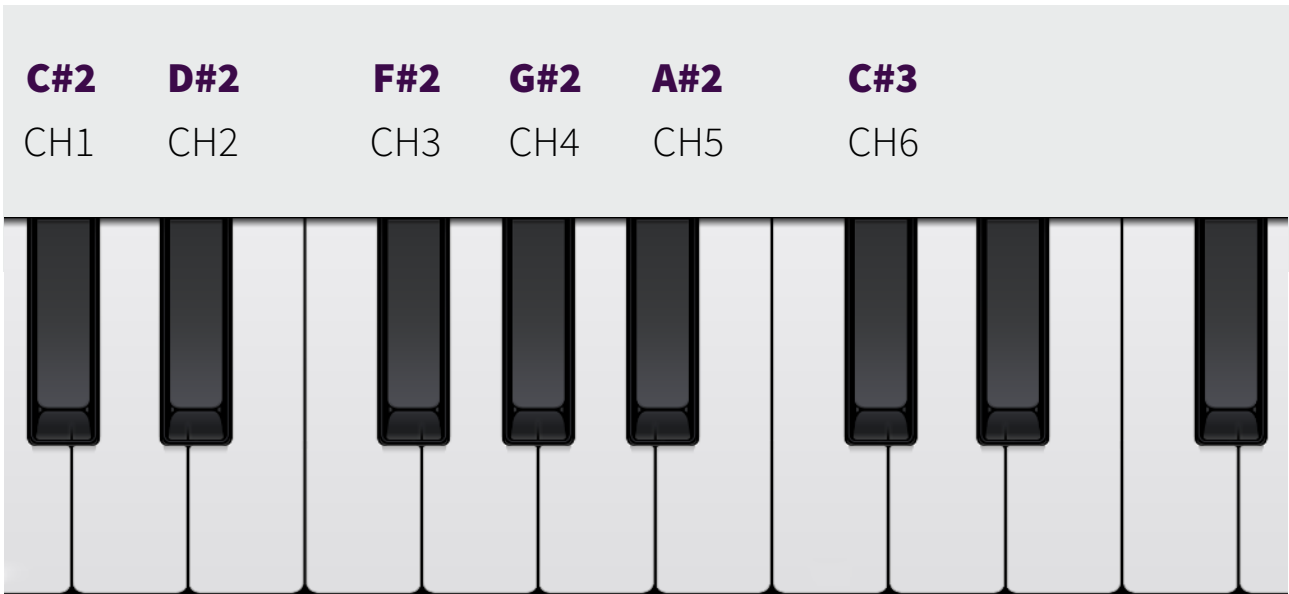
We want a clean high-pitched, nasal whirrrrr, so how are we going to get rid of that low end? HPF to the rescue. The high pass filter was added to the synth engine specifically for such purposes - to sculpt our engine's output signal according to what we need for our percussion. In this case we want all the low and mid-end gone, so crank up the HPF cutoff setting. There it is: our FM whirrrrrrrr.

We can spice it up by putting the PKF cutoff frequency in a more interesting frequency band, adding a little Saturation and tweaking the Attack setting.

Note: you may be wondering why there is no low pass filter. Very rarely will FM result in too much high frequency content. Whenever there is, you can usually tame it by decreasing the modulation Amount. No need to waste CPU (and increase the complexity of the engine and the GUI) when there are better ways to achieve the same thing, right?

1.2 Triggering notes

By default the notes of the 6 channels are mapped to the black keys in octaves 2 and 3 (octave mapping may vary between MIDI sequencers). Compared to the old GM key mapping this ‘black key’ mapping is much more convenient and intuitive for programming drum patterns using MIDI keyboards.



The corresponding MIDI key/note numbers are:

	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	
Default map	49	51	54	56	58	61	
GM keymap	36	38	39	45	48	50	

The standard setup used in most preset kits is:

	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	
Instrument	Kick	Snare	Perc1	Perc2	Perc3	Hihat	

You are free to map any instrument to any channel; the table above is just a guideline to make it easier to change out drum kits between tracks.

1.3 Controller lists

Every aspect of the Ruismaker FM plugin that can be accessed via the on-screen user interface can also be controlled and automated using AU parameters and MIDI CC codes. The following controller numbers are for MIDI CC. The respective Audio Unit parameters use their corresponding names.

Channel parameters

Channel parameters are used to control each of the 6 individual channels.

	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6
Osc Wave	30	42	54	67	79	91
Osc Pitch	31	43	55	68	80	92
Mod Wave	32	44	56	69	81	93
Mod Speed	33	45	57	70	82	94
Mod Amount	34	46	58	71	83	95
Decay	35	47	59	72	84	96
Attack	36	48	60	73	85	97
Saturate	37	49	61	74	86	98
Pan	38	50	62	75	87	99
Level	39	51	63	76	88	100
HPF	40	52	65	77	89	101
PKF	41	53	66	78	90	102

Example: to set the pan of channel 2 to full left, send MIDI CC#50 with value 0

'Active channel' parameters

These are a special set of MIDI CC codes (not available as AU Parameters) specifically added to make it easy to tweak sounds using external midi controller hardware. They are 12 CC codes, mapped to the channel parameters of the currently active channel.

E.g. if channel 2 is active, they will control the sound of the that channel. If channel 6 is active, they will control the sound of channel 6 instead. For that reason you should typically never record these CC changes in your sequencer, because the results may be unpredictable when playing back (i.e. they will still apply to whatever channel is active, which may not be the same channel that was active when you recorded the session). *So only use these if you're designing sounds using a MIDI controller.*

	Osc Type	Osc Pitch	Mod Type	Mod Speed	Mod Amount	Decay
CC#	14	15	16	17	18	19

	Attack	Saturate	Pan	Level	HPF	PKF
CC#	20	21	22	23	24	25

Example: in the image below channel 2 is active. So sending MIDI CC #15 with value 127 will change the pitch of the oscillator in channel 2 to its highest tuning setting.



It is also possible to use a MIDI controller for selecting the active channel. Use the controller value to select channels 0-5 (which represent instruments 1-6).

	Select channel
CC#	110

Global parameters

These controllers set parameters that apply to the entire plugin, e.g. the overall volume level.

Notes:

- The 'Reverb' effect emulates an old school plate reverb mechanism that adds a subtle presence to the sound when used in small amounts. It is by default set to a very minimal level. Set this effect to 0 to get a completely dry signal.

Main Volume	103	Reverb	105
Randomize	109		

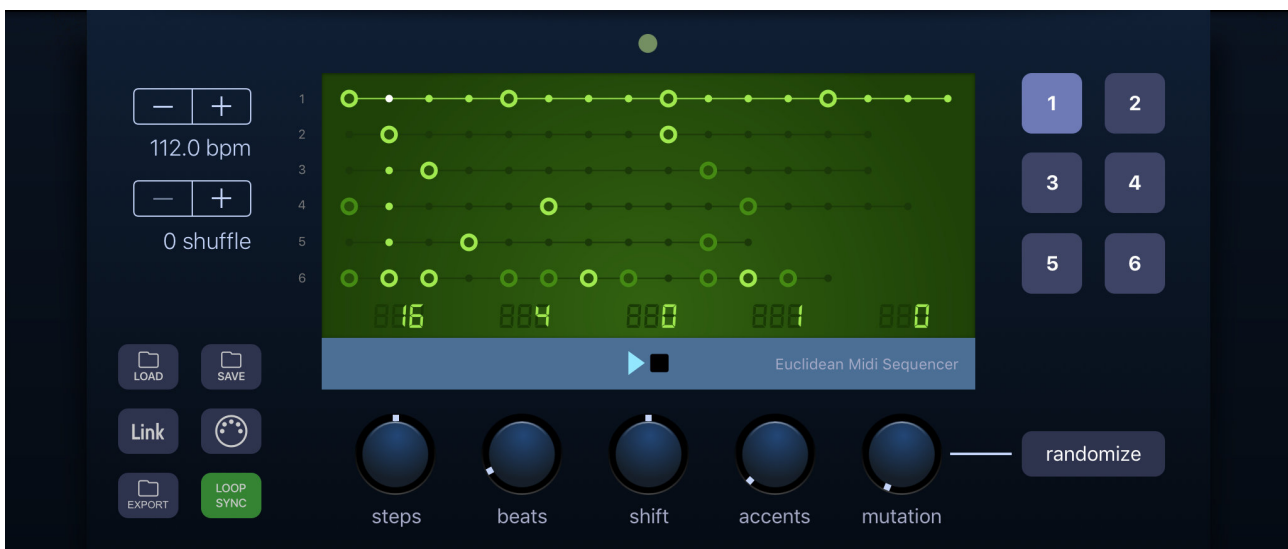
Example: to fully disable the reverb of this plugin instance, send MIDI CC #105 with value 0

2. The sequencer section

When using Ruismaker as a standalone app, it lets you drive the Ruismaker drum synth using a playful little Euclidean sequencer. Under the surface the sequencer is actually a normal MIDI sequencer sending bog-standard midi commands to the synth module.

Once you have designed (or randomly generated) a nice pattern, you can export your pattern as a standard MID file and send it directly into your DAW of choice to build a full track out of it (taking advantage of the Audio Unit Plugin format of Ruismaker).

Note: exported MIDI files do not contain swing or shuffle and all notes are quantized to a straight grid. It is assumed that the sequencer/DAW will apply its own swing algorithm to make the MIDI pattern sit better in the receiving track.



2.1 Euclidean Patterns

There's lots of interesting reading material about the theory behind Euclidean rhythms on the interwebs, so we'll just give you the executive summary here.

Euclid was a Greek mathematician from 300 B.C. who came up with an algorithm to calculate the Greatest Common Divisor (GCD) for any set of two numbers.

So far, it has nothing to do with music nor drums. But some time ago, it was discovered that many seemingly complex traditional/ethnic drum patterns follow Euclidean logic in their composition.



Recommended reading: “The *Euclidean* Algorithm Generates Traditional Musical Rhythms” by Godfried Toussaint is a remarkably clever - and highly influential - paper on Euclidean Rhythms. <http://cgm.cs.mcgill.ca/~godfried/publications/banff.pdf>

When we apply Euclid's algorithm to rhythmic patterns, it lets us take a pattern of a given number of (16th) steps and distribute a number of beats (drum hits) onto it, as evenly as mathematically possible.

For example, when we take a pattern of 16 steps and 4 beats and apply Euclid's algo, the distribution will look like this:

Steps: 16 Beats: 4 Shift: 0



Now if we apply 5 beats instead, note how the distribution changes (and how the 'excess' step introduces an interesting irregularity in the pattern):

Steps: 16 Beats: 5 Shift: 0



If we want a simple snare drum/hand clap on the second and fourth beat, we take 16 steps and 2 beats, and then shift (or rotate) the entire pattern 4 steps to the right:

Steps: 16 Beats: 2 Shift: 4



By playing with the number of steps and beats we can create anything from standard 4-to-the-floor patterns to authentic latin and highly complex african rhythms - not to mention lots of interesting experimental patterns.

Especially when different channels with different patterns and lengths are combined, we can create polyrhythms of massive complexity using just these few simple variables.

2.2 Ruismaker FM's Euclidean Sequencer



Ruismaker FM's implementation of the algorithm works as follows: each channel has its own pattern. When the playhead of a pattern reaches the end of a cycle, it will loop and start at the beginning.

Each channel pattern is controlled using 5 variables:

Steps - the total length of this channel's pattern

Beats - the number of drum hits distributed over this pattern

Shift - the rotation to the right (positive) or left (negative) of the beats in this pattern

Accents - the accent distribution: 0 = off, 1 = every step accented, 2 = every 2nd step accented, etc.

Mutation - the probability (0-100%) that a step will do the opposite of what it's programmed to do

The 5 knobs for changing the pattern can be controlled using MIDI CC:

	Steps	Beats	Shift	Accents	Mutation			
CC#	112	113	114	115	116			

Loop Sync

Loop Synchronization mode is a switch that lets you select how polyrhythms are handled. If Loop Sync is enabled, then the entire sequence will restart once the longest pattern has reached its end. This makes for nicely predictable drum loops.

If Loop Sync is disabled then every channel will run freely, making the sequence a complex free-running polyrhythm. This lets you mix different time signatures or create intricate rarely-repeating patterns.

Random

This button randomly generates a pattern for you. Mutations will be reset to 0 on all channels.

Play/Stop

This button starts and stops the playback of the pattern. If Ableton Link is enabled there may be a brief count-in before the pattern starts playing to ensure your pattern will be in sync with the other Ableton Link enabled apps on the network.

Link

This lets you control the Ableton Link feature. You can use this to sync the app with other Ableton Link compatible apps and devices which are on the same device or on the same network as Ruismaker. Changing the tempo will then change the tempo in all linked apps and conversely Ruismaker will also respond to tempo changes initiated by other apps.

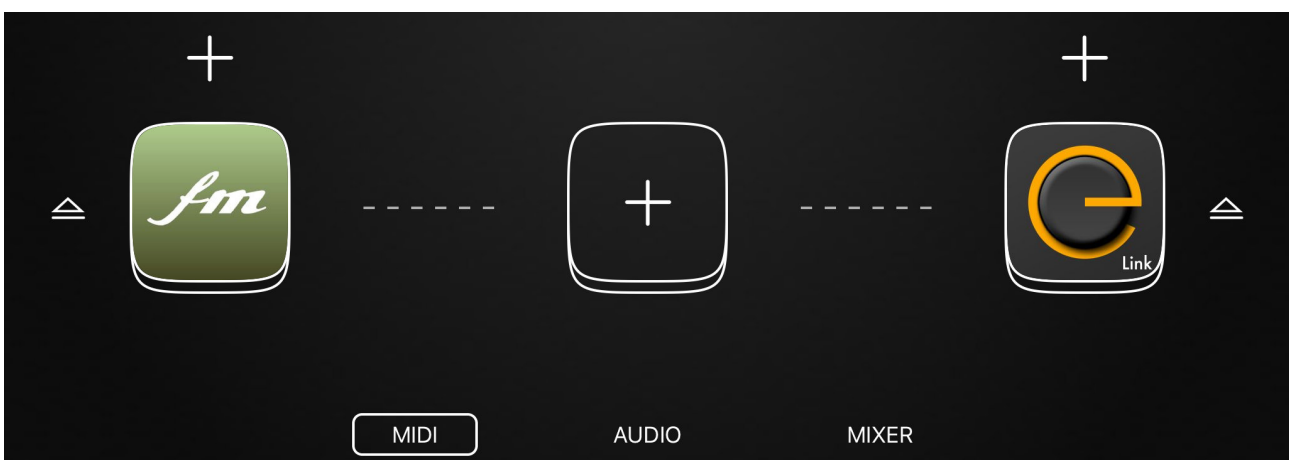
Note: when a Link session is active, Ruismaker FM will respect its tempo rules. Opening a pattern with a different tempo will not change the active Link session's tempo.

MIDI settings & Audiobus 3 MIDI

This lets you change basic settings for midi input (channel) and select Bluetooth MIDI devices. Ruismaker is fully compatible with CoreMIDI, Virtual MIDI and BTLE MIDI on iOS and will respond to midi input while running in background mode.

Note: Audiobus 3 MIDI IN/OUT is also supported. No settings need to be made for it; they are enabled automatically when Ruismaker is loaded into an AB3 MIDI slot.

When using Ruismaker FM in an Audiobus 3 MIDI slot you can use the built-in sequencer to drive other Audiobus 3 compatible apps and Audio Unit Instruments. In this case Ruismaker's internal synth engine is disabled and only MIDI notes are transmitted - not audio.

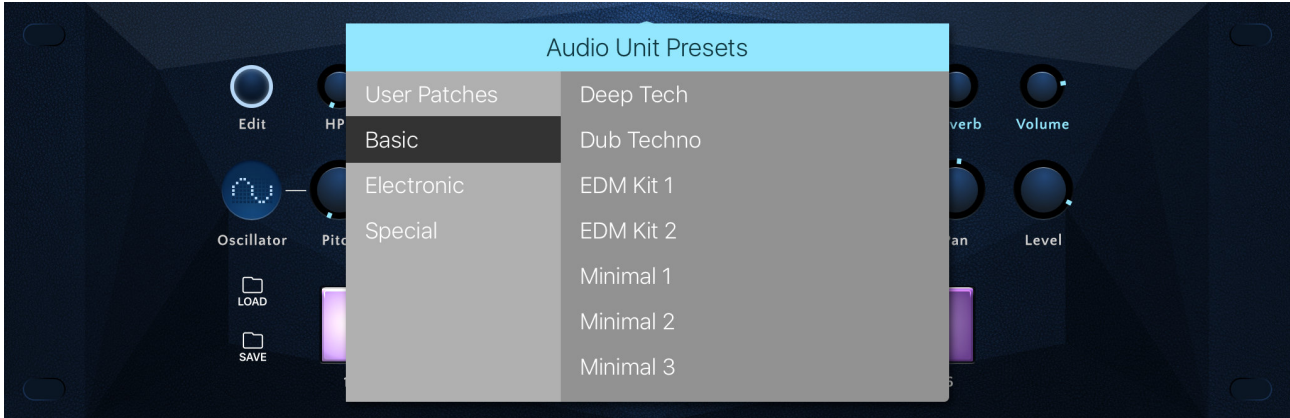


Note: the Ruismaker FM sequencer always uses the 'black keys' key mapping for output. The receiving app must be configured to respond correctly to these MIDI notes.

2.3. Saving and Exporting

AU Presets

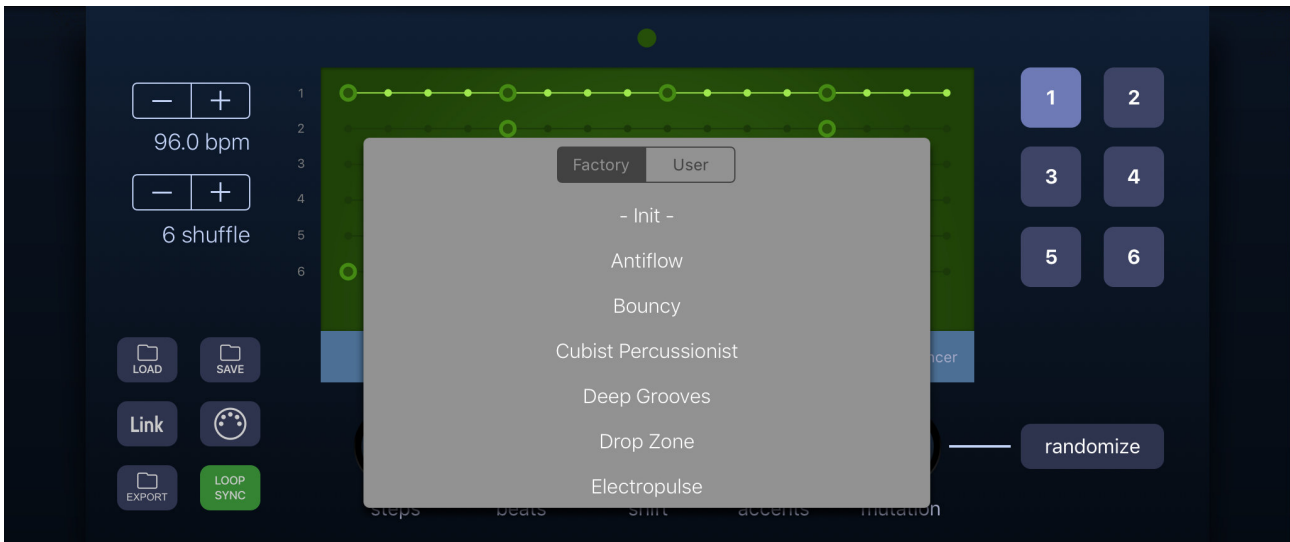
Synth settings can be saved and loaded from inside the Audio Unit plugin. This ensures that all your favorite presets and homemade patches are always available, across different AU hosts and the standalone mode of Ruismaker. To access the AU Presets, tap the SAVE and LOAD buttons in the top-right



corner of the plugin.

Patterns

A Ruismaker FM pattern contains the sequencer note data, the current synth settings, the tempo and shuffle settings. Patterns can be saved and loaded from the standalone sequencer version of Ruismaker. They can be accessed from the SAVE and LOAD buttons in the left hand margin of the sequencer.



Note: you can delete user presets and patterns from the list by swiping-to-the-left and then confirming you want to delete them. Factory patterns/presets can't be deleted.

Exporting

Several things can be exported (and imported) from the standalone version of Ruismaker. Tap the EXPORT button to bring up the export options window...

Pattern as WAV - Exports the 4 bars of the current sequencer pattern & synth settings as a WAVE file

Pattern as MID - Exports a standard MIDI file with the current sequencer pattern for use in your DAW

All Patterns ZIP - Packs all your user patterns into a Zip file and sends it to wherever you like

AU Presets ZIP - Packs all your homemade AU Presets and exports them to your preferred destination

Importing

Patterns and AU presets can be reimported as Zip files. To make sure they are valid and end up in the right place there is a special rule for how to name your Zip files:

Pattern Zip file: [yourname]_patterns.zip e.g. *ruismakerFM_patterns.zip*

AU Preset Zip file: [yourname]_presets.zip e.g. *ruismakerFM_presets.zip*

Other filenames are ignored.

2.4. Using Ruismaker FM in plugin mode

When using Ruismaker FM as plugin inside third party hosts and DAWs it will behave as any regular MIDI instrument. You can run as many simultaneous instances as your device can handle.

The Delay section will automatically synchronize with the AU host's tempo settings.

GM Keymap settings are local: they only apply to this particular instance of the AU plugin and must be set accordingly for each instance.



This document, Ruismaker FM and the Ruismaker logo © Bram Bos 2016-2017
www.ruismaker.com

