Renoise[1]

FaderPort driver

v1.1.0

**Copyright 2010-2014 Matthias Ehrmann[2],
Reviewed 2010, Organic IO**

---

1 Renoise is a registered trademark by the Renoise makers (Eduard Müller and Team), Germany
2 AKA Airmann

Table Of Contents

Introduction

Hi :-),

this is the manual for *"Airmann's FaderPort Renoise driver"*. For those who don't know what the FaderPort is, here an excerpt from the official PreSonus manual[3]:

"The FaderPort is a powerful and affordable USB DAW controller. Featuring the same ultra high-quality Alps fader that is used in professional control surfaces and digital mixing console, the FaderPort utilizes a dual servo motorized drive belt system for fast-acting, accurate, and smooth operation. The FaderPort's 100 mm long throw fader delivers 1024-step resolution for writing ultra-fine and accurate fades and adjustments. In addition, the FaderPort puts complete automation and transport control packed in its small footprint for a cluster free workspace."

---

3   http://www.presonus.com/media/manuals/FaderPort_manual_v2x-web.pdf. PreSonus is a registered trademark of
    PreSonus Audio Electronics, USA

What's so special about the FaderPort ?

Ok, what's so special about it ? Admittedly: it has only one fader and pan control and some buttons !

It's so special, because it has a touch-sensitive ALPS high-quality motor fader with a 10 bit resolution (1024 values). That's the same fader which is built into mixing consoles which cost thousands of dollars[4]. Of course, one can argue that such consoles have more than one fader and a lot of other fancy stuff, but with the right DAW integration even one motor fader can be a decent weapon.

Why ? The problem of standard faders for digital audio workstations is well known: e.g. the track volume is adjusted by using controller fader 1. And bingo: as soon as you modify the volume by using the mouse the fader is out of sync with the DAW's volume value.

This problem is among other solutions[5] solved by using motor-driven faders that are kept in sync with the DAW's values. As soon as you change a value inside the DAW, the corresponding motor fader moves it's knob "magically" to a new position.

Thereby it's important that the fader does this precisely, quickly and quietly. Moreover, a longer fader offers more precise adjustments than a short one. The problem about lower quality faders: they usually tend to suck in one ore more discipline. In contrast: the FaderPort's fader fulfills all requirement of a high-quality fader. The other FaderPort controls/knobs are also not bad, though the pan control is not as good as the fader (mine sucks a bit).

Conclusion: it's better to have one high-quality fader that really can be used, than 24 low-quality ones that suck. Such a fader is very worthwhile for mixing and creation of automation envelopes.

That said, proper DAW integration of this "one" fader is of course also very important. Therefore, PreSonus ships FaderPort with drivers for all major DAW's like Cubase, ProTools, Logic and so on.

---

4   E.g. DigiDesign's D-Command
5   e.g. touch-sensitive endless controllers like in the Novation Nocturn series

Even for Reaper there exists an open source driver[6]. And for Renoise now there exists one too :-).

Features of Renoise Integration

The main objective was to create an ergonomic, sophisticated and stable working tool, which should integrate seamlessly into Renoise. Focus was laid on mixing, automation envelope creation, and general DAW control. The result is a piece of software which really can improve your workflow and mixing process. Most of the features should be self-explanatory. Nonetheless, there are features and modes which need some explanation, so it's highly recommended to study this manual.

If you have a question, find a bug or have suggestion for improvement, just let me know. Positive feedback is always welcome :). Now to the features:

- support of all FaderPort buttons, touch-sensitive motor fader, endless pan control and lights.
- footswitch support for hands-free operation. You can connect an usual on/off footswitch to the FaderPort. Footswitch has the same function as transport play. It's especially useful for hands-free recording of audio and midi data.
- transport: play, play from position, stop, panic stop, forward, rewind, record / edit, loop, block loop, block loop forward/backward
- track: mute, solo, select next track, select previous track, select master track, jump to song start / end, record sample
- pre-fx / post-fx volume / pan, and any DSP device parameter can be controlled
- **currently selected track is automatically bound to FaderPort. Binding is displayed in status bar.**
- currently selected device/plug-in etc. can automatically be bound to FaderPort.
- customizable navigation through device / plug-in parameters. Relevant device parameter lists can be defined via preferences. Presets for all native devices are included.
- **full automation envelope support: read, write, touch, latch mode**
- full support of bidirectional communication for controls and lights
- 10 bit high resolution (1024 steps) for fader values (allows for precise mixing)
- endless pan control support with speed feature[7], adjustable virtual resolution (default 7 bit, 128 steps), auto down scaling and adjustable "anti-suck" protection
- fader and pan controller can be swapped (e.g. fader controls stereo panning)
- fader value can be reset to Renoise default value (e.g. volume to 0 dB)
- fine trim mode: fader values can be fine trimmed via shift+pan control
- switch between views: mixer, pattern editor, sample editor
- sticky mode support: FaderPort can be sticked / bound to a specific track or device
- undo / redo support[8]
- Renoise status bar support (displays parameter bindings and value changes)
- Emulator mode which emulates real FaderPort hardware. Useful if real hardware is not available or 3rd party midi controllers shall be used.

---

6   http://www.reaper.fm
7   the faster the pan is moved the higher the value. Thus, values can be dialed in much quicker
8   is still problematic (Renoise 2.7) !

Installation

Requirements

For running Renoise FaderPort driver Renoise 2.6 or higher is required. For emulator mode Renoise 3.0 or higher is required. Driver was only fully tested on Windows 7. Please have a look at the Renoise homepage / forums for scripting tool installation on other platforms (Mac OS, Linux). Also you need to install the latest FaderPort PreSonus driver for your OS platform. Don't know about Linux drivers.

Installation and Configuration

For installation open Renoise and drag&drop the file **de.airmann.FaderPort.xrnx** to the Renoise Windows. You will be asked if you want to install the tool. That's the same for upgrading to a new version. After installation there'll appear a new entry "FaderPort" in the "Tools" menu. On Windows FaderPort driver should work "out of the box" without any further configuration[9].

If you experience problems check the Troubleshooting section of this manual.

In certain cases it can be necessary to modify the **config.xml** XML file depending on your MIDI setup, OS platform and FaderPort hardware quality. To open the config.xml select "Tools: Tool Browser", right click on FaderPort entry and select "Reveal in Explorer". Open and modify config.xml with a text editor of your choice. Hint: you should save the modified file in UTF-8 format !

If FaderPort is not working for your OS, open the Renoise midi settings and find out the names of input and output FaderPort midi ports. Enter those names in the config.xml: midi_in_name, midi_out_name.

Finally, you can try to optimize the "anti_suck_min_turns" value for faster pan control response, depending on your FaderPort's hardware quality. Try values between 1 and 3. A lower value is faster, but also more error prone.

For everything else read this manual carefully. Especially "Preferences" and "Troubleshooting" sections.

Deactivation / Uninstallation

In order to temporarily deactivate the FaderPort driver open "Tools: Tool Browser" and select "disable". Hint: this kind of "deactivation" is still a bit unclean (e.g. the FaderPort light state isn't reset and so on). The better and recommended way for doing this: select "Disconnect" in FaderPort menu and disable "Auto Connect" in FaderPort menu.

For complete uninstallation open "Tools: Tool Browser" and select uninstall. **BEWARE: this also deletes your current configuration / preferences ! In order to backup it open "Tool: Tool Browser" and right click on "de.airmann.FaderPort.xrnx", select "Reveal in Explorer", and copy "config.xml" to a safe location.**

Control Overview

The following sections contain a quick overview about the buttons and various control features of Renoise FaderPort driver. The handling of the various controls should actually be pretty self-explanatory. Nonetheless, there are features and modes which need some explanation. They are further described in the following chapters.

---

9   nonetheless the config.xml can be optimized

Transport

BUTTON(S)	ACTION
Play	start/stop playing (or recording)
Shift+Play	start/stop playing from/at position
Footswitch	The same as Play (shift and recording also works)
Stop	stop playing
Stop x 2	panic stop. All sounds are stopped
Forward	increase position in pattern sequencer
Rewind	decrease position in pattern sequencer
Shift+Forward	jump to song end position
Shift+Rewind	jump to song start position
Record	edit mode on/off
Loop	pattern loop mode on/off
Punch	block loop mode on/off
Shift+Punch	decrease block loop position
Shift+User	increase block loop position

Application

BUTTON(S)	ACTION	NOTES
Mix	switch middle frame to mixer view	
Proj	switch middle frame to pattern editor	
Trns	switch middle frame to sample editor	Only if sticky mode support is disabled in preferences !
Undo	Renoise undo	
Redo	Renoise redo	
Menu: View: FaderPort Device Infos	show/hide FaderPort device Info dialogue	can also be bound to keyboard shortcut
Menu: Tools: FaderPort: Reset FaderPort	reset FaderPort and re-initialize MIDI I/O, but don't alter state	can also be bound to keyboard shortcut
Menu: Tools: FaderPort: Connect	connect FaderPort to Renoise and initialize MIDI I/O	can also be bound to keyboard shortcut
Menu: Tools: FaderPort: Disconnect	disconnect FaderPort from Renoise and close MIDI I/O	Releases also resources and decreases CPU load. Can also be bound to keyboard shortcut.
Menu: Tools: FaderPort: Auto Connect	Enables/disables automatic FaderPort driver connection after a song is created/loaded. The state is persistently stored in the preferences	If disabled, you have to manually connect (activate) FaderPort driver for each song

BUTTON(S)	ACTION	NOTES
Menu: Tools: FaderPort: Undo Workaround	enable/disable undo workaround	can also be bound to keyboard shortcut

Mixer

BUTTON(S)	ACTION	NOTES
Mute	mute/unmute current track	
Solo	solo/unsolo current track.	for master track: it's like muting everything !
Rec	show/hide sample recording dialogue	
Fader	change volume	this can be either pre-fx volume or post-fx volume. Controls swapped: pre-fx pan / post-fx pan .
Pan	change panning setting	Controls swapped: change volume.
User	swap/unswap fader and pan control	fader -> pan pan -> volume
Shift+Loop	display fader/pan binding	display current binding in status bar
Shift+Pan	fine trim volume	controls swapped: fine trim pan. Has no effect if volume/pan is automated.
Output	toggle pre-fx and post-fx mixer	default is pre-fx mixer
Shift+Output	select master track	
Channel select left	select previous track	
Channel select right	select next track	
Rewind+Forward	resets volume to default value	controls swapped: reset pan to default value. Hint: not possible if volume is automated !
Trns	sticky mode on/off	only if sticky mode support is enabled in preferences !
Bank	device mode on/off	See Device

Device

BUTTON(S)	ACTION	NOTES
Fader	change parameter1 value	
Pan	change parameter2 value	
User	swap/unswap fader and pan control	fader -> parameter2 pan -> parameter1
Shift+Pan	fine trim parameter1	controls swapped: fine trim parameter2
Shift+Loop	display fader/pan binding	display current binding in status bar
Channel select left	select previous parameter pair	device parameter pairs must be defined in the preferences !

Channel select right	select next parameter pair	device parameters pairs must be defined in the preferences !
Rewind+Forward	reset parameter1 to default value	controls swapped: reset parameter2 to default value. Hint: not possible if parameter is automated !
Trns	sticky mode on/off	only if sticky mode support is enabled in preferences !
Bank	device mode on/off	

Fader Automation

Hint: in order to enable a fader mode an automatable parameter must be bound to fader !

BUTTON(S)	ACTION	NOTES
Read	read mode on/off	read/ follow automation data
Write	latch mode on/off	playing mode: automation data is written after replay has been started and fader has been touched. Until fader has been touched automation data is read. non-playing mode: if the fader is touched an automation point is added/modified at the current edit position. If necessary a new automation envelope is created.
Shift+Write	write mode on/off	automation data is written immediately after song has been started. Has no effect in non-playing mode.
Touch	touch mode on/off	playing mode: automation data is written after replay has been started and fader has been touched. If fader is untouched/released automation data is read. non-playing mode: if the fader is touched an automation point is added/modified at the current edit position. If necessary a new automation envelope is created.
Off	switch fader temporarily off	as long as off button is pressed, no fader values are sent from FaderPort to Renoise. **Doesn't switch off any fader modes !**

Controls in Detail

Fader Control

As already mentioned the FaderPort provides a high resolution touch-sensitive motor fader. The fader is completely supported by the Renoise FaderPort driver. This means: as soon as you touch/move the fader, touch-state and a value between 0..1023 is sent from FaderPort via MIDI to

the Renoise driver. Afterwards the driver replaces the value of the currently selected Renoise parameter with the received one. And vice versa: as soon as you change the parameter value in Renoise, FaderPort's fader-knob is moved by the motor to a new position.

Now, one may assume that the fader and the Renoise value are always in perfect sync, but unfortunately this is not completely the case. Despite it's high resolution the fader can only be positioned on 1024 different positions. In contrast the Renoise parameter can have a much higher resolution. E.g. it's possible to enter values with multiple decimal places using the keyboard and so on.

Thus, there can arise a slight difference (rounding error) between the actual Renoise value and the corresponding fader position[10]. The consequence: if the fader is touched/moved, the Renoise value "micro-jumps" sometimes. But in praxis that shouldn't be a real problem.

Another thing that matters here: whenever the fader position is set near/to the extrema (0,1023), the knob isn't completely moved to the edges[11]. Vice versa: in order to set a Renoise parameter value to its extrema[12], the fader-knob must be moved almost all the way to the edges. This means there can occur a slight "mechanical difference/tolerance" between fader read and write position[13]. My impression is that this is more significant near the extrema than in the middle positions, but I have no prove of that. But the same here: in praxis this should be no biggy.

Also worth to mention in this context is that Renoise uses non-linear value scales for various parameters. E.g. for volume a logarithmic dB scale is used. That means for the extrema (e.g. -40 dB): a small fader position change causes a bigger value change than in the middle parts (e.g. around 0 dB).

Special case: Emulator's Fader Control

Hint regarding the resolution of the fader of FaderPort's emulator: the resolution of Renoise UI slider/fader controls (viewbuilder) is always 100. Means: despite the emulator's fader can receive and send values between 0..1023 it's resolution and precision is always limited to 100 ! Effective mapping: 0..99 → 0..1023 → Renoise control target.

Moreover, any 3rd party midi controller which is mapped to the emulator's fader, has a max. resolution of 128 (0..127), so the effective mapping is: 0..127 → 0..1023 → Renoise control target. This is a little bit higher compared to emulator GUI only.

Further, the emulators fader control emulates touch and untouch state of the original hardware as follows: whenever a new fader value shall be sent to Renoise, a touched event is sent before the fader value is sent, and an untouched event is sent after the fader value was sent.

---

10 This should not be the case if the value is of type integer and has a range <= 1024
11 I assume this is a desired behavior that shall protect the fader from taking damage. But no prove, here.
12 e.g. maximum: 3 dB, minimum -INF dB
13 sorry can't describe that better

Pan Control

Endless Hardware Pan Encoder

The pan control is an endless encoder which sends two different values to the Renoise FaderPort driver when being turned left or right. Since the control is endless the sent values are always relative. Means: actually there are two different signals sent: "one position left" or "one position right"[14]. After receiving the signals, FaderPort driver translates them into absolute Renoise parameter values.

Virtual Pan Control

For translation the driver uses a virtual model of an absolute pan control. All received values are first forwarded to the virtual pan control. From there they're forwarded to the current selected Renoise parameter. Vice versa: if the Renoise parameter value changes, the value change is forwarded to the virtual pan control. But unlike fader, nothing is sent to FaderPort, because the real pan control doesn't need to be adjusted.

Resolution

The resolution of the virtual pan control is 128 steps by default. This is a standard value for many midi controllers. Nonetheless, if you want to control a parameter which has a higher resolution than 128, you can increase the resolution in the FaderPort driver preferences[15]. For lower values than 128 the resolution is automatically scaled down.

Speed and Down-Scaling Feature

Furthermore, the virtual pan control has an automatic speed control feature, which is helpful to quickly adjust values within larger value ranges. Means: the more quickly you turn the hardware pan, the larger is the value change. E.g. you want to adjust the pre-fx pan setting of track A to 40L. Just turn the pan encoder more quickly and the value jumps to 30-40L. Without speed feature you had to turn the pan encoder several times[16].

On the other hand there are parameters which have a small discrete value range[17]. E.g. the oscillator selection parameter of the native RingMod device:

This means there are only 5 different values to choose from. In this case it's enough to use a resolution of 5. Otherwise you had to turn the pan encoder also several times to select a specific oscillator type[18]. It's much more convenient to navigate in such selection lists with a single turn.

---

14 This is not completely true: if you turn the pan very (!) quickly there may be different signals sent. But these are not usable in praxis
15 a lower value is also possible. If it's too low precise adjustments are not possible
16 That's e.g. the case in Reaper FaderPort driver
17 with quantum 1
18 128/5 = approx. 26 !

Therefore, FaderPort driver detects discrete value ranges[19] automatically and down-scales the resolution to a reasonable value.

Fake Touch/Release Support

One problem about the FaderPort endless encoder is that it is not touch-sensitive and there's no signal sent which indicates the end of the control action (last value received)[20]. To solve this problem the virtual pan has a time-based "fake touch/release" feature. This means: whenever a pan value is received, it is assumed that the pan encoder was touched and an "touched" event is fired. For release detection the time after receiving the last pan encoder value is internally measured. After 0.5 seconds it is assumed that the pan encoder isn't touched anymore and a "released" event is fired.

Snap to Center Support

For bipolar parameters it's important that the exact center values can be dialed in[21]. It shouldn't matter what resolution or scaling is used. Therefore, the FaderPort driver snaps/rounds the values always to center if they're near the center.

Anti-Suck Protection and Anti-suck "turn calibration"

Since the pan encoder of my FaderPort controller sucks from time to time, I added an anti-suck protection feature.

Background: whenever I move my pan encoder, it sometimes sends incredibly high values despite I move it only a little bit. In some cases it even crashes the complete FaderPort and I have to manually reset it. Moreover the pan value flutters from time to time (jumps between values). Let's be polite here and say the pan encoder works sometimes a bit "unprecisely".

Anyway, the anti suck feature filters out those "value bursts" and counts the number of pan encoder hardware turns[22]. Only if a specific number of turns is reached, the virtual pan value is increased or decreased. This protects from value fluttering. The drawback of this is that one have to turn the pan encoder more often for a specific value adjustment.

So if your FaderPort works more precisely than mine, you can decrease the number of minimum needed turns. The best value would be 1. My value is 3. You can change the value in the FaderPort driver's preferences[23]. Hint: if you chose lower values, the speed is implicitly increased. Higher values decrease the speed.

Trim Control

If the shift-button is pressed, the pan encoder values are routed to the so called "virtual trim control" instead of the pan control. The virtual trim control has the same features (speed, anti-suck, etc.) like the virtual pan control. The trim control is further described in the following sections.

Fader Fine Trim Control

As already mentioned the fader - thanks to it's high resolution - works very precisely. Nonetheless, there may be situations where you need to "fine tune" parameters even more. E.g. volume or pan.

Therefore, the FaderPort driver provides a fine trim control for the fader. To activate it press the shift button and move the pan control. Now all movements fine adjusts the parameter value, which

---

19 All parameters with quantum1. Moreover, for ranges <= 20 the speed feature is automatically switched off
20 This is e.g. important for undo-data creation
21 think of e.g. the pan settings. You want the exact center, not +0.001 or something
22 Every time you turn the pan encoder it makes those clicky sound. This is one turn.
23 anti_suck_min_turns = x

is currently bound to fader[24].

Hint: right now it is not possible to fine trim automation envelope points !

Swapping Of Fader And Pan Control

It's possible to swap the parameters which are currently bound to fader and pan control by pressing the user button. This is e.g. helpful for quick and precise adjustments of the pre-fx / post-fx pan parameters. But it works of course for any parameters which are bound to fader and pan.

Reset Fader Parameter To Default Value

Whenever "rewind" button and "fast fwd" button are pressed at the same time, the fader parameter is reset to it's default value. That works for any parameter. E.g. if a volume parameter is bound to the fader it is reset to 0 dB. If a pan parameter is bound to the fader it is reset to center and so on.

Optional Footswitch

It's possible to connect an usual footswitch to the FaderPort for hands-free operations. The footswitch is treated like a button and fires simple pressed/release events. The default function for the footswitch is the same as for the play button. This means also shift+footswitch works. The footswitch can e.g. be very useful for hands-free recording of wave and midi data.. A footswitch that was tested and works is e.g. the Korg PS-1

Hint: if the cable is not long enough, you can use an extender like e.g. Planet Waves Coupler Adapter PW-P047T.

Another thing is the press/release behavior of the footswitch. It seems that some footswitches send a "0" signal when footswitch is pressed down and a "1" signal when footswitch is released. Therefore, you can define if the "pressed" signal is either "1" or "0" in FaderPort driver preferences.

Undo/Redo Handling

Modes In Detail

The FaderPort driver provides several modes for different typical working / controlling scenarios. Those modes are explained in the following sections.

Mixer Mode

After Renoise is started or song changes, FaderPort driver is switched into mixer mode. In mixer

---

24 The resolution of the fine adjustments is $\dfrac{parameter\,value\,range * 0.5}{1000}$

mode either track pre-fx or post-fx volume is bound to fader, and either pre-fx or post-fx pan is bound to endless pan encoder. You can define in the FaderPort driver preferences if you want to start in post-fx or pre-fx mixer mode by default[25]. The output button indicates if post-fx mixer is on/off.
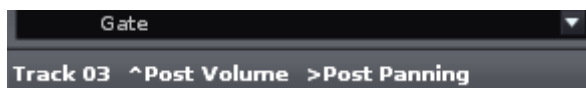
Track Selection And Automatic Parameter Binding

Whenever a track is selected in mixer mode, FaderPort driver automatically binds the track's volume and pan parameters to the fader and pan control. Moreover, if the FaderPort's additional power supply is connected, the fader-knob is adjusted to the current fader parameter value by the motor. This is very convenient and accelerates the workflow.

Important hint: if the parameter is automated, then the parameter value can't be modified by moving the fader or pan control, because the envelope has priority. Nonetheless, you can modify the envelope by using the fader.

The current parameter bindings and fader / pan control value changes are also displayed in the Renoise status bar.

E.g. for Track 03 post volume is bound to fader (^) and post panning is bound to pan control (>) and this is display in the status bar:

E.g. for Track 06, reverb's room size parameter is bound to fader (^) and Width parameter is bound to pan control (>) and this is displayed in the status bar:
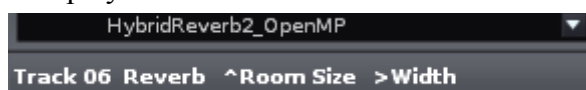
E.g. the fader is moved and room size value change is displayed accordingly in the status bar:
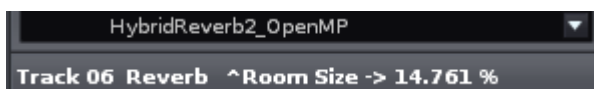
Especially in sticky mode this is worthwhile for getting feedback of the value changes

Hint: you can press FaderPort Shift+Loop buttons to display the current parameter binding in the status bar.

Track Navigation

To navigate between tracks you can use the channel select left/right buttons. But tracks can also be selected by using the mouse, of course.

Mute, Solo, Rec

Mute and solo buttons mutes or solos the currently selected track. The master track can not be muted, but it can be soloed. If it's soloed it's the same as it was muted. If a send track track is soloed the sending track (source track) is not muted.

---

25 default_is_post_fx_mixer = true/false

Recording Audio

The rec button shows or hides the sample recording dialogue. When the recording dialogue is opened one can press the user button to start/stop recording or arm recording. This depends on the "sync start&stop" setting. If "none" is selected, user button starts/stops recording immediately. If "pattern" is selected, then recording is just armed and one have to start playing to start/stop recording.

Hands-free Recording By Using The footswitch

Hint: since the footswitch starts/stops playing, it's really helpful for hands-free audio recording in "pattern sync" mode. Especially if one's playing an instrument etc.. Of course this is also helpful for recording midi data.

Device / FX Mode

Pressing the "bank" button activates device / fx mode. In this mode the fader and pan control are bound to various parameters of the currently selected device. Important: the device must have been registered to be controlled by FaderPort driver before. This means: before you can control the parameters of a specific device, you have to define a parameter binding list for those device in the FaderPort's preferences. For all native devices a default binding list is already included. For 3rd party devices this is not the case.

A device can be any native Renoise device[26], or any VSTi/e, AU or LADSPA plug in. All kind of parameters are supported: float, discrete integer, unipolar, bipolar.

Important hint: if the parameter is automated, then the parameter value can't be modified by moving the fader or pan control, because the envelope has priority. Nonetheless, you can modify the envelope by using the fader.

Parameter Binding List

As mentioned above, one have to define a parameter binding list for every device that shall be controlled by FaderPort driver. Right now this must be done by modifying the FaderPort driver's XML preference file "config.xml" by using a text editor. The section in the preference file starts with "<dsp_binding_list>":

```
<dsp_binding_list>
 <dsp_binding>Delay</dsp_binding>
 <dsp_binding>5</dsp_binding>
 <dsp_binding>6</dsp_binding>
 <dsp_binding>1</dsp_binding>
 <dsp_binding>2</dsp_binding>
 <dsp_binding>3</dsp_binding>
 <dsp_binding>4</dsp_binding>
 <dsp_binding>9</dsp_binding>
 <dsp_binding>10</dsp_binding>
 <dsp_binding>12</dsp_binding>
 <dsp_binding>13</dsp_binding>
 <dsp_binding>-1</dsp_binding>
```

Every device has it's own list. The list starts with the device name and ends with "-1". The bindings are defined as pairs for fader and pan. So the overall number of "<dsp_binding> XXX </dsp_binding>" entries must be even ! The numbers define the positions of the device parameters. It's possible to enter 0. This means: no parameter shall be bound to fader or pan. Moreover it's

---

26 All native devices, means also meta and routing and so on !

possible to define the same parameter for fader and pan, so that fader and pan control the same parameter.

The order of the list is descending. Means: the first binding pair - in our example [5,6] - is bound first to fader and pan. The last pair [12,13] is last.

In order to find out the necessary device name and parameter position FaderPort driver provides a "Device Info" dialogue (View Menu -> FaderPort Device Infos):

The first line contains the name of the currently selected device. To define a new parameter binding list for a device just copy it's name to the preferences file. the second line displays the currently defined parameter binding list of the device. The big text field displays the numbers and names of all device parameters.

Binding Pair Selection / Navigation

It's possible to navigate through the various parameter binding pairs by pressing "channel select left" or "channel select right" button. Whenever a new device is selected, the first parameter binding pair is bound to fader and pan.

Sticky Mode

As mentioned above, FaderPort driver binds fader and pan control automatically to volume and pan parameters of the selected track (mixer mode). The same for devices and their parameters in device mode. But there may be situations where this default behavior is not desired. E.g. if you want to adjust volume and stereo panning of a specific track in relation to other tracks etc..

Therefore, it's possible to make the actual parameter binding "sticky". Just press the "trns" button to make the current binding sticky. Press the "trns" button again to end sticky mode.

Hints:

• If you move a sticky track in mixer mode to a new position, the track remains sticky. If you delete a sticky track, sticky mode is disabled automatically.

• if you move a sticky device in device mode to a new position of the origin track, the device remains sticky. If you move it to another track or delete it, sticky mode is disabled automatically.

• Switching between device mode and post- / pre-mixer modes doesn't disable sticky mode

• Automation envelope creation also works in sticky mode

• Parameter bindings and value changes are also displayed in the Renoise status bar. This is especially helpfull in sticky mode to get feedback of the value changes

- The below described undo workaround doesn't function in sticky mode
- if you don't need sticky mode at all, it's possible to disable it in the FaderPort driver preferences. In this case the "trns" button switches the Renoise middle frame to the sample editor view.

Envelope Automation And Fader Modes

As you might know Renoise supports recording of automation envelopes by using the right mouse button on device parameter sliders[27]. Alternatively you can bind any midi controller to a device parameter / slider and record the incoming midi data. Also with FaderPort it is possible to do that. But in addition to Renoise core, FaderPort driver distinguishes between several modes for reading and writing automation envelopes.
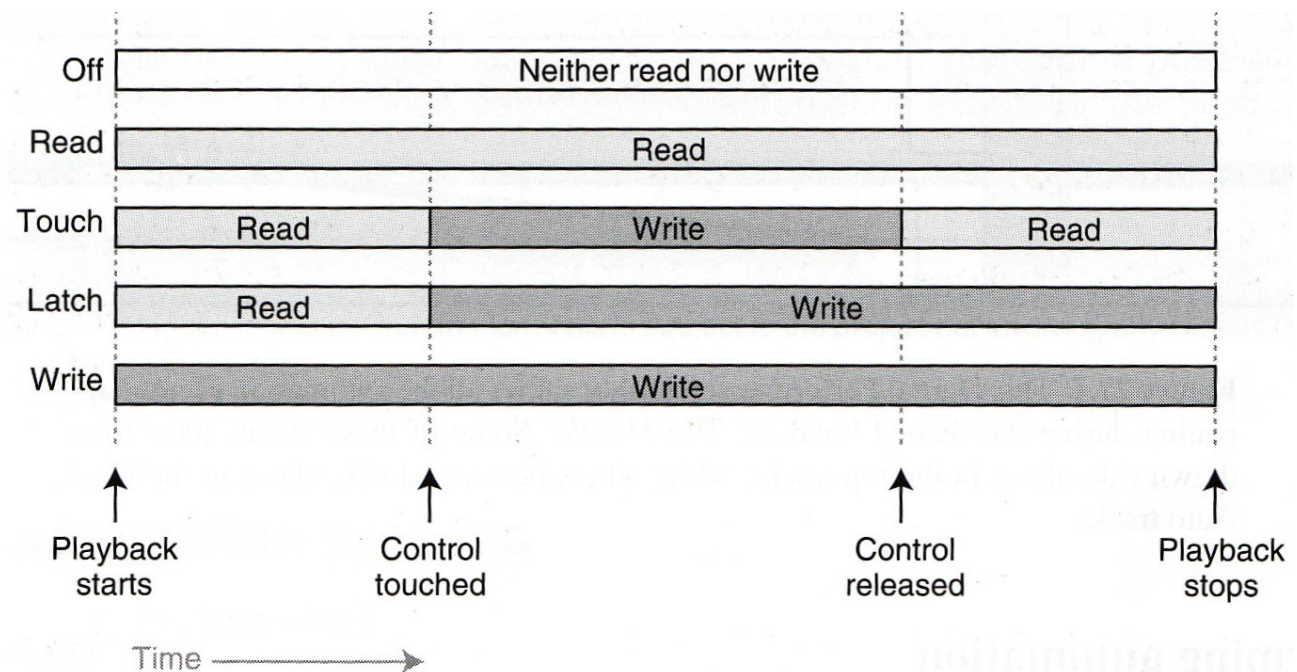
Hint: recording automation data into patterns/tracks is not supported !

Modes In Detail

The supported fader automation modes are:

- Read
- Write
- Touch
- Latch

These modes are standard for most major DAW's and are further explained in the following sections. The picture below[28] illustrates the different modes in detail:

- **Off:** automation data is neither read nor written. This is the default mode of FaderPort driver. In this mode the lights of all mode buttons are off. Important: the "off" button switches off sending fader values on the hardware itself.
  Means: as long as this button is pressed, no new fader MIDI values are sent from FaderPort. Therefore, the button can be used for fader adjustments without actually sending new values. Nonetheless, the last received value is written to envelope, whenever fader is

---

27 works in playing and non-playing mode
28 Taken from "mixing audio, concepts practices and tools", Roey Izhaki, Focal Press 2008

touched ! Otherwise FaderPort driver just ignores this button.

- **Read:** previously written automation data is read, but new fader changes are not written
- **Touch**: new automation data is written as long as the fader is touched, otherwise previous automation data is read.
- **Latch:** automation data is written from the moment the fader is touched until playback stops. This is the default write mode of FaderPort driver. Means: if you press "write" button, this mode is activated[29].
- **Write:** new automation data is written as long as the playback is running, overriding previously recorded automation. Attention: don't forget to disable write mode after the recording session. There's no automatic prevention of unwanted automation overrides. To enable this mode press "shift"+"write" buttons.

Generally: in order to enable a specific mode, just press the specific button. In order to disable a mode press the button again or select a different automation mode.

For write/touch mode: if no automation envelope exists for a specific parameter, a new envelope is automatically created. The same if automation envelopes are recorded over multiple patterns.

All read modes only work if the additional power supply is connected to FaderPort !

Whenever a new parameter is bound to fader, all automation modes are disabled ! Moreover, it's not possible to enable any fader mode, if there's no parameter bound to fader or if parameter is post-fx volume / pan.

Recording / Step-Editing in non-Playing Mode

Touch and latch-mode also work in non-playing mode[30]. In order to create automation data, switch to "pattern view", open the automation envelope which is bound to fader, and move the fader while pressing left/right on the keyboard. I call this technique step-editing[31]. It's useful for envelope fine-corrections etc..

Technical Hints And Limitations

Additional Power Supply Is Needed

It's a good idea to connect the additional power to FaderPort when working with the fader and/or using the fader automation modes. Otherwise the motor doesn't work, and therefore read, touch and latch modes doesn't function properly. Nonetheless, writing and fader touch state should always work, even if the power supply is disconnected.

Recording Gaps

Since there are still some technical limitations of the Renoise scripting API, it's right now not possible to record automation envelopes without "gaps". Means: if you record at high BPM / LPB settings or with high CPU load, it may happen that some positions are skipped. To avoid this you can temporarily lower the BPM / LPB settings while recording, and/or reduce the CPU load.

Missing Undo / Redo Support

Right now it's not possible to undo/redo a recording of many envelope points in one action. Means: you have to activate undo/redo for every single automation point that was recorded. That's

---

29 You can also set a different default write mode in the FaderPort driver preference
30 Latch- and touch-mode are the same in non-playing mode
31 That's also possible with other midi controllers

especially a problem for longer recordings over multiple patterns. But you can hold down "strg-z/y" which works very quickly. Hint: the below described undo workaround doesn't work for automation envelopes !

Ugly Undo Workaround Mode

Right now Renoise doesn't support a proper undo/redo mechanism for received real-time data. Moreover, Renoise uses internally an undo grouping feature that is activated as soon as a key or the mouse is pressed. While the undo grouping is active, all actions are bundled in one undo/redo action.

Unfortunately this leads to a number of technical problems for the FaderPort driver. E.g. if you move the fader or pan control Renoise creates lots (!) of undo / redo points and thus kind of "pollutes" the undo/redo stack. Right now, there's no proper solution for the problem, because it must be solved inside Renoise.

Nonetheless, I added an ugly[32] workaround which reduces the number of created undo points. With the workaround undo points are only created after the fader or pan control has been released / untouched. BUT: this only works as long as the undo grouping is not activated, means no mouse or key is pressed. Otherwise it undoes things that shouldn't be undone (no bug, just a shitty feature ;).

Because of the undo grouping, the workaround is really dangerous to use. Therefore, I don't recommend to use it !

The only real use case is post processing / final mixing, where proper undo/redo is really important. But you have to work disciplined here: don't use fader or pan control while pressing mouse or keys. Then it should work - at least for me it does ;-). Another hint: the workaround can also cause some crackling while moving the fader/pan control. That's ok and nothing to worry about.

The workaround can be enabled/disabled via menu entry: "Tools: FaderPort: Undo Workaround".

IMPORTANT: if you want to use the workaround, it's a good idea to always connect the additional power supply to FaderPort. Though it's no absolute must, it can avoid trouble

As soon as Renoise provides an alternative this mode will be removed !

Emulator Mode

Since v1.1.0 the FaderPort driver package contains an emulator mode which emulates the real FaderPort hardware controller 1:1. So why create an emulator when there exists the real hardware ?

Firstly, the day will come when the FaderPort hardware won't be produced anymore, and without further development the driver will be useless.

Secondly, since the driver's design and stable code base is regarded as useful, it should also usable for non-FaderPort owners. So a 1:1 FaderPort software emulator with GUI was developed:

---

32 ugly because it's technically not possible to create a clean solution from outside Renoise
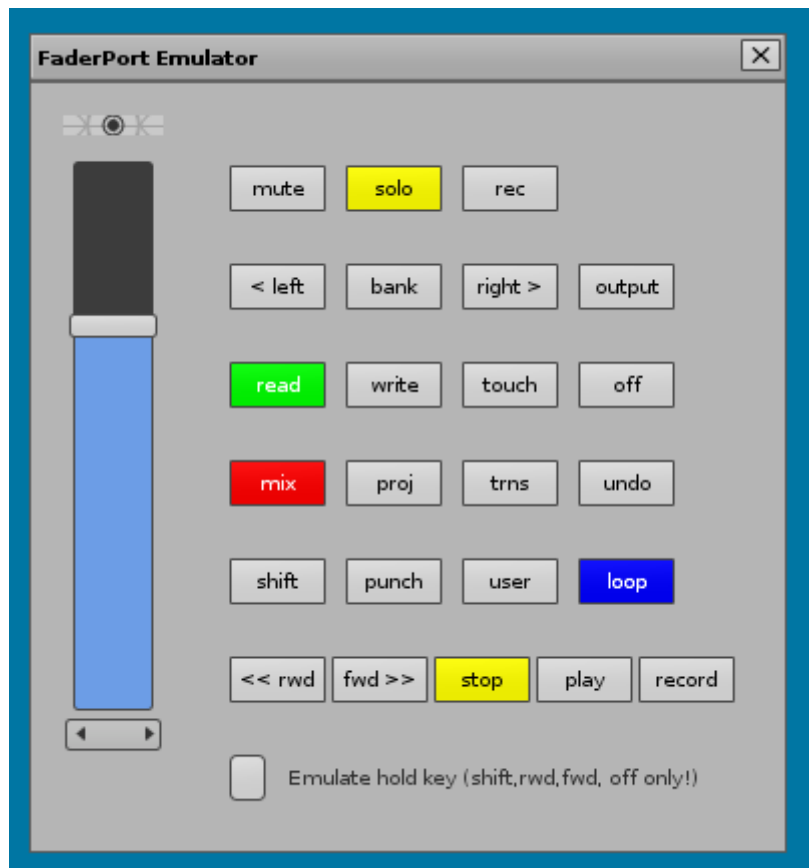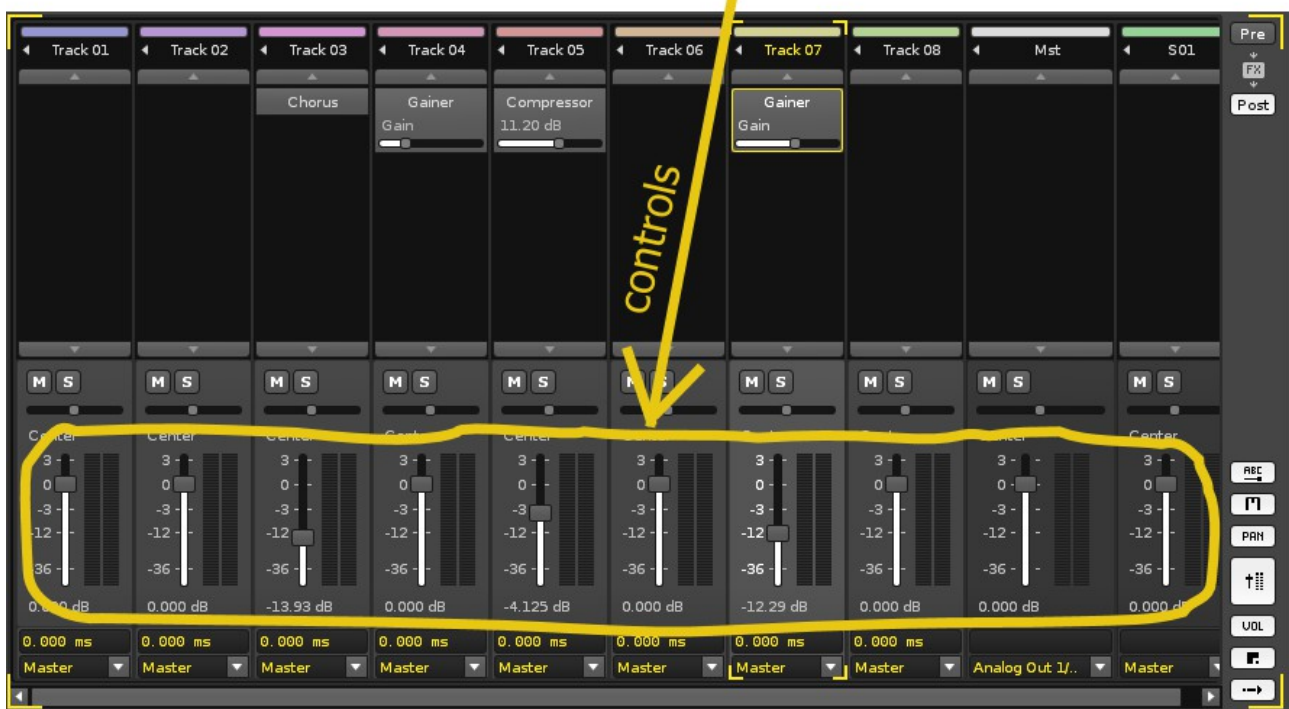
Emulator Features

- fully integrated into Renoise 3.0
- resembles 1:1 with the design of the real hardware
- acts almost like the real hardware
- All buttons and controls - especially the fader - can be MIDI-mapped to any 3rd party midi controller. Thus, the driver's code can be used by any midi controller.

Renoise Integration Of Real FaderPort Hardware

The FaderPort's motorized fader controls the volume level of the currently selected track. Whenever another track is selected, the FaderPort changes it's position according to the new track's volume level.

Analogue the FaderPort's buttons (like play etc.) control Renoise and also reflect Renoise's state.
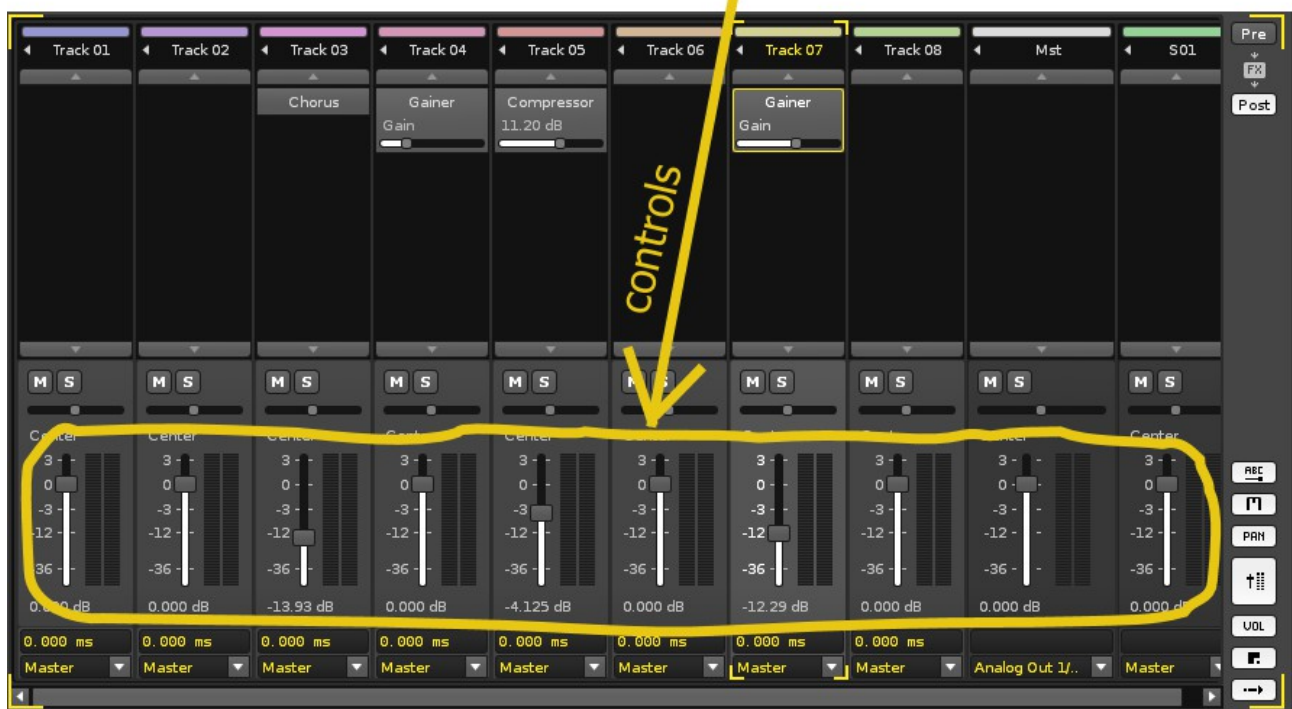
Renoise Integration Of FaderPort Emulator
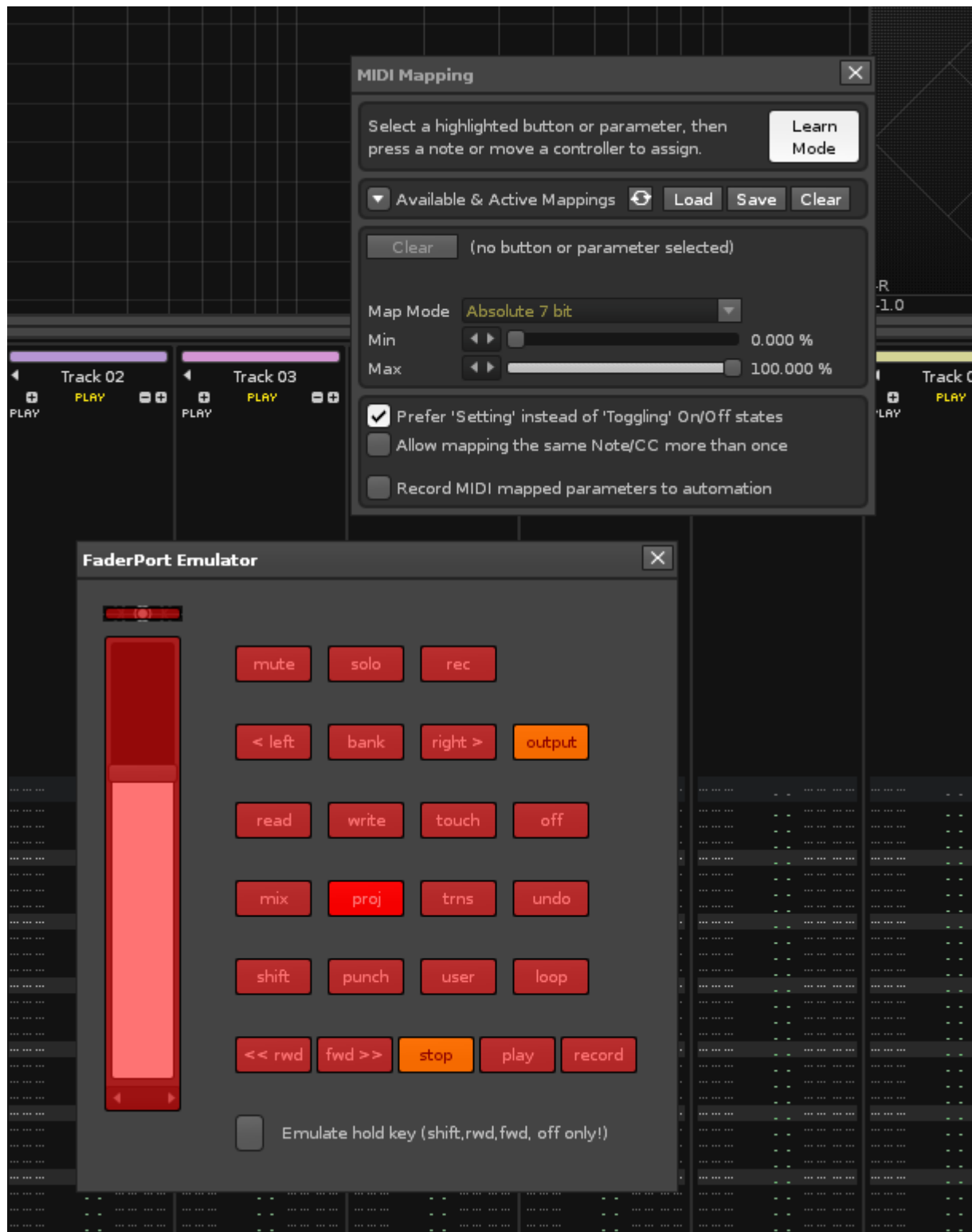

The emulator's fader controls the volume level of the currently selected track. Whenever another track is selected, the fader changes it's position according to the new track's volume level.

Moreover, it is possible to midi map any midi controller to the fader, pan and buttons of the emulator GUI dialog:

In the example above an endless rotary knob is mapped to the emulator's fader. control Means: whenever the knob is turned, the volume level of the currently selected track is changed. **The big advantage: regardless of how many tracks a song has, only one rotary knob can control them all ! There's no need for a huge motorized mixing console etc..**

IMPORTANT HINTS:

- **for fader control mapping only absolute midi map mode is supported. Relative sources are not possible !**

- **For pan control mapping only relative midi map mode is supported. Absolute sources are not possible !**

The emulator mode can be enabled/disabled via the FaderPort tools menu:
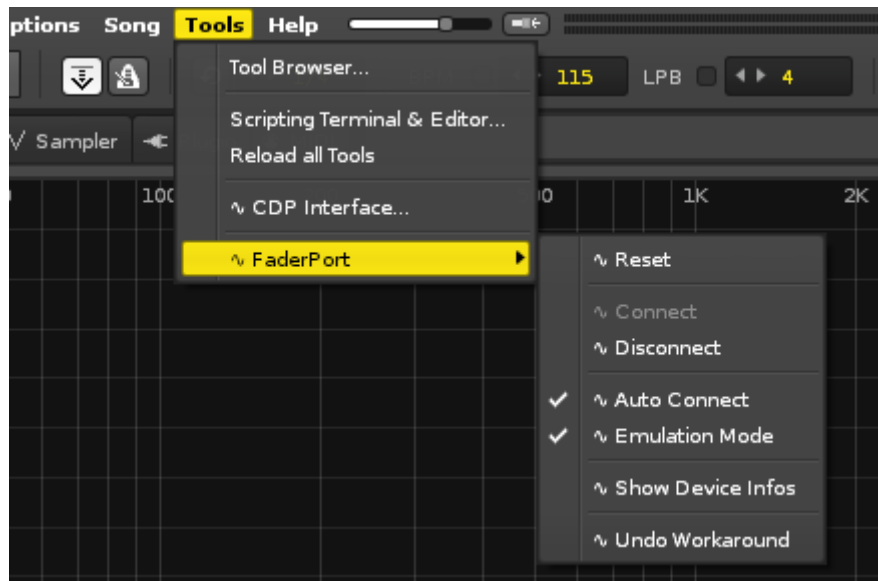
Preferences

FaderPort driver can be configured via XML configuration file. For Windows the file path is e.g.
C:\Users\USERNAME\AppData\Roaming\Renoise\V3.0.0\Scripts\Tools\de.airmann.FaderPort.xrnx\config.xml.

Alternatively you can also select "Tools: Tool Browser:", right click FaderPort entry and select "Reveal in Explorer". This opens the folder which contains config.xml

You can open and modify the file with any text editor. Hint: you should always edit and save the file in UTF-8 format. Options are provided as XML tags:

OPTION	DEFAULT	DESCRIPTION
default_is_post_fx_mixer	true	true: FaderPort drivers starts in post fx mixer mode, false: in pre-fx mixer mode
latch_is_default_write_mode	true	true: default fader write mode is "latch", false: default fader write mode is "write"
sticky_mode_support	true	true: sticky mode available (via trns button), false: sticky mode unavailable
auto_connect	true	true: Renoise connects to FaderPort automatically during start up
emulation_mode	false	true: instead of real hardware an emulation dialog is opened. Real hardware is ignored. false: emulation dialog is closed.
undo_workaround	false	true: use dirty undo workaround to avoid massive creation of undo data. NOT RECOMMENDED TO USE. IMPORTANT: doesn't work without FaderPort additional power supply !
anti_suck_min_turns	3	the minimum number of hardware pan

		encoder turns for a virtual pan value change. The lower the value (minimum 1), the faster the pan response, but also the higher the error. Adjust this value according to your pan encoder hardware state (calibration)
virtual_pan_resolution	128	resolution of virtual pan control. Increase this value for higher pan control precision.
midi_in_name	FaderPort	name of MIDI in device
midi_out_name	FaderPort	name of MIDI out device
footswitch_pressed_signal	1	Should either be 0 or 1: this is helpful for compensation of varying footswitch wiring
dump_midi	false	true: dumps received midi data to terminal. Primarily for developers !

Menu Entries and Key Bindings

See category Tools:FaderPort

Troubleshooting / Known Errors

PROBLEM	SOLUTION(S)
FaderPort doesn't function after Renoise was started, or a new song was loaded / created	• Make sure FaderPort is properly connected BEFORE Renoise is started. Check USB connection AND the USB PowerSupply. TRY TO UNPLUG / PLUG THE USB CONNECTOR !!! • Make sure that the auto connect feature is enabled. (Tools: FaderPort: Auto Connect). Otherwise try to connect manually (Tools: FaderPort: Connect) • Try to reset FaderPort driver (Tools: FaderPort: Reset FaderPort). If that doesn't help wait some time and try to reset again • Make sure that no other DAW or Renoise instance is using FaderPort • Make sure that FaderPort is not used as general MIDI device for Renoise (see Renoise Preferences) • Check if the FaderPort MIDI in/out device name is correctly set in FaderPort driver preferences. • If nothing else helps try to unplug/plug USB connection, restart Renoise and reset driver. • Re-Install the FaderPort Presorts drivers according to PreSonus manual, and upgrade the firmware.
FaderPort is not responding	This either can happen if FaderPort crashes as described below, or if FaderPort driver crashes because of a bug. In both cases, resetting FaderPort should help. If it's reproduceable then it's probably a bug. Please report, thanks !
Fader motor doesn't work	Make sure that the additional power supply is connected to

	FaderPort
Though fader isn't really moved, fader sends a lot of values	This is an effect that can occur if the motor is off. It can also pollute Renoise's undo/redo queue. Make sure that the additional power supply is connected to FaderPort.
FaderPort crashes when moving the pan control (e.g. very slowly)	This seems to be a FaderPort Firmware bug. Try to reset FaderPort driver whenever this happens. (Tools: FaderPort: Reset FaderPort). If that doesn't help wait some time and try to reset again.
FaderPort endless pan encoder produces value bursts / jumps	This seems to be another FaderPort Firmware bug. The anti-suck feature should help. Try to increase the anti_suck_min_turns value in the FaderPort driver preferences
FaderPort doesn't work for multiple Renoise instances, or while other DAW's are active.	Right now FaderPort can't be used by more than one Renoise instance or DAW at the same time[33] . This is especially a problem when using Renoise via Rewire and both programs try to connect to FaderPort. In this case FaderPort must be disabled for one program. To disable FaderPort in Renoise just disconnect it (Tools: FaderPort: Disconnect)
When recording automation envelopes some positions are skipped	This can happen with high BPM/LPB setting is, or if the CPU load is high. Just lower the speed temporarily while recording. This is no error, but a technical Renoise limitation
When recording automation envelopes the related Renoise slider is bouncing / jumping	That's ok. It jumps between the old and newly recorded values.
Trim mode doesn't work for automation envelope points	That's ok, not implemented
When using the undo workaround strange things happen	Don't press mouse or keyboard while moving fader or pan control. Otherwise all kind of actions are undone. Make also sure that the additional power supply is connected
fader and pan can't be used at the same time when undo workaround is enabled	that's right, they are "mutually excluded" in undo workaround mode.
To many undo points are created	This is a technical Renoise limitation. Use strg-z/y.
FaderPort driver doesn't function for Linux or MAC OS	FaderPort driver was only tested for Windows, but should work for Linux/MAC, too. Make sure that the names of the MIDI I/O devices in the FaderPort driver preferences are correct.
Track width parameter is not controllable	that's right, it's the only Renoise parameter which is not adjustable by the FaderPort driver right now. As a workaround you can use the native stereo expander device, or any other 3rd party stereo widening plug-in
undo workaround doesn't function in sticky mode	that's right, it's disabled in sticky mode
parameter value doesn't change	check if the parameter is automated. Automated parameters

---

33 I've made some experimental code to solve this, though

when moving fader or pan control	can't be directly controlled by fader / pan, because the envelope has priority. Nonetheless, you can modify the envelope by using the fader
whenever a track is selected in mixer mode, or a device is selected in device mode, the fader mode is disabled. The same for switching between post-fx / pre-fx mixer	This is correct, whenever a new parameter is bound to fader, all fader modes are disabled.
I can't enable a fader mode in mixer mode nor in device mode	It's not possible to enable any fader mode, if there's no parameter bound to fader or if parameter is post-fx volume / pan.

Developer Infos

Debugging

Enable dump_midi in config.xml. Have a look at Debug.lua. There you can define regexp filters for tracing function calls. Almost all functions are traceable. The function midi_callback() is also traceable, but you have to hold the FaderPort's shift key while tracing it. Last but not least _AUTO_RELOAD_DEBUG = true in main.lua.

Regression Tests

The following table contains a collection of regression tests which should be applied for every new Renoise release, Renoise OS version, scripting API release, and after every modification of the code itself. It should also be run with emulation mode on and off !

The used config.xml is the default one. The number of tests is by far not complete. I've tried to chose tests which are not so obvious[34], but nonetheless very significant for overall stability.

TEST	EXPECTED RESULT
FaderPort XRNX Installation via drag&drop	success
Renoise start up, FaderPort not connected	success, driver disconnected
Renoise start up, FaderPort connected, power supply connected	success, driver connected, fader motor on
Renoise start up, FaderPort connected, power supply disconnected	success, driver connected, fader motor off
Renoise start-up, auto connect disabled	success, driver disconnected, MIDI I/O closed
disconnect FaderPort	success, driver disconnected, MIDI I/O closed
connect FaderPort	success, driver connected, MIDI I/O opened
reset FaderPort	success, MIDI I/O closed/re-opened
disconnect, Reset, while driver disconnected	success, nothing happens
connect, while driver connected	success, nothing happens
disconnect USB connection while FaderPort is connected to Renoise, afterwards re-establish USB connection	success, FaderPort doesn't react any more, Renoise must be restarted in order to use it again.

---

34 none like e.g. press button, move fader etc.

shutdown Renoise	success, all FaderPort lights are cleared, MIDI I/O is closed
disable auto connect and load or create new song	success, new song is activated and FaderPort is disconnected
load Song while moving, touching or pressing controls (especially fader / pan)	success, nothing happens
launch second Renoise instance	xfail at least for one instance, MIDI I/O problems (Windows: MME device error)
press multiple buttons at once	success, actions are executed
move fader and pan at the same time	success, actions are executed, values change
quickly navigate between tracks in mixer mode, power supply connected	success, fader adjusts accordingly to the selected track
switch quickly between device mode and mixer mode pre-fx / post-fx, power supply connected	success, fader adjusts accordingly to the selected device / track parameters.
mute / solo master track	success, mute doesn't function, solo mutes all tracks but master track which is the same as the mute
play a song with different pattern lengths and press the forward / rewind buttons quickly and multiple times	success, forward / rewind works as expected
move fader and pan very long and check Renoise memory consumption	success, undo stack grows. For a longer movement Renoise allocates some KB'S or MB's
move pan very slowly and very fast	success / xfail, seldom FaderPort might crash, otherwise works as expected
iterate back and forth through a dsp binding pair list in device mode	success, list should be iterated in both directions endlessly
move tracks / sticky track in sticky mixer mode	success, tracks are moved as expected. When sticky track is moved it remains sticky
delete tracks / sticky track in sticky mixer mode while moving fader / pan	success, tracks are deleted. If track was in sticky mode, sticky mode is disabled automatically. Next track is selected automatically. If fader / pan are moved, volume / pan on next track is modified (seamless transition)
change volume / pan of sticky track in sticky mixer mode, while changing volume / pan of another track	success, track parameter values change in parallel
move devices / sticky device in sticky device mode	success, devices are moved as expected. When sticky device is moved to a new position of the origin track it remains sticky. When sticky device

	is moved to another track, sticky mode is disabled
delete devices / sticky device in device mixer mode while moving fader / pan	success, devices are deleted. If device was in sticky mode, sticky mode is disabled. If fader / pan are moved, volume / pan on next track is modified (seamless transition)
change value of sticky device in sticky device mode, while changing value of another device on the same or another track, or change volume / pan of any track.	success, parameter values change in parallel
Switch between device and mixer modes in sticky mode	success, sticky mode isn't disabled
use write, touch, latch for creating new automation envelope in playing mode, try also in sticky mode	success, new envelope should be created and written automatically, also over multiple patterns, and also for sticky track / device
use read, write, touch, latch, for already present automation envelope in playing mode, try also in sticky mode	success, present envelope should be read or modified, also over multiple patterns, and also for sticky track / device
use touch, latch for creating new automation envelope in non-playing mode, try also in sticky mode	success, new envelope should be created and written automatically. Should work also in sticky mode
use write for creating new automation envelope in non-playing mode	success, has no effect
use read, write, touch, latch for already present automation envelope in non-playing mode. Navigate through envelope points using the arrow keys (pattern editor / step editing)	success, envelope should be read or modified.
try to recording automation envelopes with high BPM / LPB settings or with high CPU load	success, but some positions are skipped. Increasing LPB should have the most impact (?).
press off button while changing parameter values with the fader, or recording automation envelopes	success, no new values are sent from FaderPort while off button is being pressed. Nonetheless, the last received value is written to envelope, whenever fader is touched !
try to use trim mode for an automated parameter	success, trim mode not possible for automated parameters
try to activate a fader mode in post-fx mixer mode and record an automation envelope for volume or pan	success, fader mode can't be enabled, no envelope is created, nothing is recorded
try to activate a fader mode in device mode if no device is selected	success, fader mode can't be enabled if no parameter is bound to fader
enable undo workaround while power supply is disconnected. Move the fader and pan control	success, should work,

move fader and pan at the same time while undo workaround is enabled	success, mutual exclusion
move fader or pan while keyboard or mouse is pressed	success, but unwanted group undo actions occur
use trim mode while undo workaround is enabled	success, should work as expected
enabled sticky mode while undo workaround is enabled	success, undo workaround is disabled for sticky mode
switch on/off sticky mode in undo workaround mode while moving fader. Press undo	success, in sticky mode lots of undo data points are created. Otherwise undo workaround functions as expected
modify preferences / config.xml	success, modifications should function as expected. Renoise doesn't have to, but can optionally be restarted.
use unusual / forbidden or extreme values (0,-1 etc.) for various options in preferences	xfail, so far no validity check. FaderPort driver should crash
use unusual / forbidden or extreme values for dsp binding pairs in preferences, switch to device mode and select a native dsp	xfail, assertion fails and dsp selection doesn't function correctly

FaderPort MIDI Charts

If you're unhappy with the button layout of the FaderPort driver, it's possible to modify MidiHandler.lua. The following MIDI charts may help you[35]

MIDI SEND EVENT (ON/OFF)	DESCRITPION
0xa0, 0x15, 0/1	mute light state
0xa0, 0x16, 0/1	solo light state
0xa0, 0x17, 0/1	rec light state
0xa0, 0x14, 0/1	channel select left light state
0xa0, 0x13, 0/1	bank light state
0xa0, 0x12, 0/1	channel select right light state
0xa0, 0x11, 0/1	output light state
0xa0, 0x0d, 0/1	read light state
0xa0, 0x0e, 0/1	write light state
0xa0, 0x0f. 0/1	touch light state
0xa0, 0x10, 0/1	off light state - switching on disables fader !!
0xa0, 0x0c, 0/1	mix light state
0xa0, 0x0b, 0/1	proj light state
0xa0, 0x0a, 0/1	trns light state

---

35 See also http://en.wikipedia.org/wiki/Musical_Instrument_Digital_Interface

0xa0, 0x09, 0/1	undo light state
0xa0, 0x05, 0/1	shift light state
0xa0, 0x06, 0/1	punch light state
0xa0, 0x07, 0/1	user light state
0xa0, 0x08, 0/1	loop light state
0xa0, 0x04, 0/1	rewind light state
0xa0, 0x03, 0/1	forward light state
0xa0, 0x02, 0/1	stop light state
0xa0, 0x01, 0/1	play light state
0xa0, 0x00, 0/1	record light state
0xb0, 0x00, vv	fader value, MSB
0xb0, 0x20, vv	fader value, LSB
0x91, 0x00, 0x64	reset / switch on faderport

MIDI RECEIVE EVENT	DESCRIPTION
0xb0, 0x00, vv	fader value, MSB
0xb0, 0x20, vv	fader value, LSB
0xe0, 0x00, vv	pan control value (left 0x00, right 0x7f)
0xa0, 0x7f , 0/1	fader touched / released
0xa0, 0x7e, 0/1	footswitch pressed / released
0xa0, 0x7d, 0/1	pan control touched / released (virtual event)
0xa0, 0x7c, 0/1	trim control touched / released (virtual event)
0xa0, 0x00, 0/1	user pressed / released
0xa0, 0x01, 0/1	punch pressed / released
0xa0, 0x02, 0/1	shift pressed / released
0xa0, 0x03, 0/1	rewind pressed / released
0xa0, 0x04, 0/1	forward pressed / released
0xa0, 0x05, 0/1	stop pressed / released
0xa0, 0x06, 0/1	play pressed / released
0xa0, 0x07, 0/1	record pressed / released
0xa0, 0x08, 0/1	touch pressed / released
0xa0, 0x09, 0/1	write pressed / released
0xa0, 0x0a, 0/1	read pressed / released
0xa0, 0x0b, 0/1	mix pressed / released
0xa0, 0x0c, 0/1	proj pressed / released
0xa0, 0x0d, 0/1	trns pressed / released
0xa0, 0x0e, 0/1	undo pressed / released

0xa0, 0x0f, 0/1	loop pressed / released
0xa0, 0x10, 0/1	rec pressed / released
0xa0, 0x11, 0/1	output pressed / released
0xa0, 0x12, 0/1	mute pressed / released
0xa0, 0x13, 0/1	channel select left pressed / released
0xa0, 0x14, 0/1	bank pressed / released
0xa0, 0x15, 0/1	channel select right pressed / released
0xa0, 0x16, 0/1	output pressed / released
0xa0, 0x17, 0/1	off pressed / released

History / Change Log

- v1.0
  - initial release, tested against Renoise 2.6 RC2
- v1.0.1
  - Bugfix Release
  - Manual reviewed and corrected by Organic IO
- v1.0.2
  - Slight modifications for Renoise 2.7
  - changed behaviour of output button. Output toggles now post-fx / pre-fx mixer and indicates if post-fx / pre-fx mixer is selected
  - removed quick switch output mode
- v1.1.0
  - Slight modifications for Renoise 3.0
  - New midi mapable emulator mode with 1:1 emulation of real FaderPort hardware
  - Reorganized tools menu
  - Reorganized keyboard and midi bindings