

fadeOS Version 3.04

New firmware and editor program for the Peavey StudioMix

© 2002-2007 by Oliver Dahlmann

ATTENTION ATTENTION ATTENTION ATTENTION ATTENTION ATTENTION

To use the new editor, you also need the appropriate firmware (fadeOS V3.02 or higher). Try the editor without the new firmware, if you want, but it's not possible to transmit configuration data or motor profiles into your StudioMix without it.

ATTENTION ATTENTION ATTENTION ATTENTION ATTENTION ATTENTION

table of contents

1.Installation.....	3
Installation of the upgrade (Hardware).....	3
Software – editor-program installation.....	4
Software – activation.....	4
2.The Beginning.....	7
What is a configuration?.....	7
Messages and groups.....	8
Example: Creation of a new configuration	8
3.Basics.....	11
Hexadecimal?.....	11
The MIDI standard.....	11
MIDI messages.....	12
4.Construction of messages.....	15
Modes.....	15
Placeholder.....	15
Range.....	16
5.Motor profiles.....	18
6.MIDI monitor.....	22
7.Import und Export.....	23
Export a configuration.....	23
Export a group.....	23
8.Q&A.....	25
Rien ne va plus!.....	25
Faders and Knobs doesn't send any messages.....	25
Faders doesn't send positions.....	25
Can't alter configurations, groups or whatever.....	25
Assign SysEx Data.....	25
Other questions?.....	25
Have fun!.....	25

1. Installation

Installation of the upgrade (Hardware)

Attention: Before you are going to open the device, switch it off and unplug all cables, especially the power cord. Now wait a few minutes, before you open the StudioMix. Remove the backplate. Lay down the device on a soft layer on its frontside. Now you'll see the PCB containing the microcontroller (the chip with 40 pins) and the eeprom (the little one with 8 pins).



Attention: Discharge yourself (static electricity) by touching a metallic surface of a radiator.

Gently remove your original microcontroller with a screwdriver from its socket.

Place the delivered microcontroller in the empty socket.

Attention again: Look for Pin 1 (the „u“-shape in the microcontroller package has to show in the same direction as you can see above).

On the same PCB, in the left upper edge, there is a little EEPROM that you have to change, too.

Repeat the same procedure for this part. The delivered EEPROM has a soldered wire on its top due to slightly different pinouts. Look for the alignment of Pin 1, too.

Software – editor-program installation

Requirements: The software was tested with Microsoft Windows 98SE, Windows Me, Windows 2000 and Windows XP. The program and its documentation needs approximately 5MB of free harddisk space.

Run *setup_fadeOS_V3_04.exe*. The Installer will lead you through the installation process. After successful installation you can start the program for the first time.

The default language for the editor is german, but you can change the user interface to english language settings (Datei→Einstellungen→English).

Now choose the MIDI interface, where you have connected your StudioMix controller to (File→Preferences→MIDI-Interface).

Software – activation

Before you can start your work with your upgraded StudioMix controller, you need to activate the firmware. Without activating it, you can transmit configurations, but you will not be able to send fader or knob-positions (faders and knobs are deactivated, the gray buttons are working).

Assuming that you have made your language- and midi-interface-settings as described above, you will have to do some very simple steps:

First, click on the menuitem 'File→Preferences→Enter Activation Code'.



Now, the following window appears:



Click on the button 'Show Serialnumber'. Now a dialog containing your firmwares serialnumber appears.



Write an email with the subject *activation* addressed to info@sonicware.de with this serialnumber (in this example, you would simply write B00EC526 in the Email-Body, the Mail will be read by human eyes, so don't worry to much about email composition).

I will respond as fast as possible and send a reply with your activation code.

If you received your activation code, enter it in the 4 editfields, press on OK, done.

Normally, you will never need to do this again, even if you buy a new pc and reinstall the editor software, because the activation number is 'burned' into your StudioMix-EEPROM, but it may be a good idea to write down the activation code.

Now, have fun with your studiomix's upgraded possibilities.

2.The Beginning

With the StudioMix editor, you can configure the behaviour of each fader, push button or rotary encoder. Look at the picture; this is how the editor software looks right after installation.



What is a configuration?

With 'configuration' i mean a collection of faders, buttons and rotary-controllers, which are assigned to (self) defined MIDI messages.

The user can define a lot of different configurations. They all are saved in one file (studiomix.xml). You can import and export your configurations to exchange it with other users. The resulting XML files are small, only some kilobytes in size, so you can send them via email without problems.

A configuration doesn't have to consist of ALL faders, buttons and rotarys. It's possible to define subsets for the faders only, or only for 5 buttons and 2 rotary-controllers. What is it good for? You can 'stack' your configurations: For example, load the configuration for the faders first, send it to the StudioMix, load another configuration for the buttons, send it to the StudioMix, too, and so on. Knobs, which are used in a configuration, are colored blue. If there are buttons, faders and rotarys which are not in use, they are painted grey.

Each configuration has a name (shown in the combobox) and a description text, shown in the right upper corner (the green chars on black background you see in the picture or on your screen). If you alter the description, it will be saved automatically.

Messages and groups

To prevent long terms of searching your messages, they are sorted in groups. The group 'Volume', for example, consists of the messages 'Volume MIDI Channel 1' to 'Volume MIDI Channel 16', in the group 'Standard Peavey' are all original NRPN-Messages. (look Panel 'Choose message' in the editor window).

Example: Creation of a new configuration

With the menu 'set' → 'create' you can create a new configuration.

Give it a name:



After creating the new set, your editor window looks like this:



All buttons, knobs and faders are coloured grey, so you can see: none of them has a message assigned to itself.

Click on the leftmost fader with your left mousebutton. The fader will become green, it's

selected now. Now choose a group:



Choose a message contained in this group.



Ready. The selected message is now assigned to fader 1. If you select another fader with your mouse, you will see that fader 1 now is coloured blue.

If you click on the button 'Send to StudioMix', the actual configuration will be transmitted to

the StudioMix controller. In this example it would only affect the message assigned to fader 1, all other faders, buttons, etc. will remain their old message assignment.

There are some messages predefined. You can select them simply with your mouse. (Click on a fader, a rotary knob or a button, select a message, done).

But if you want to define your own messages, you will have to learn about hexadecimal numbers, MIDI specifications and the following topic:

3.Basics

Hexadecimal?

If you know, what hexadecimal numbers are, you can stop reading and jump to the next chapter. All others can continue reading. If you press a button, rotate a knob or move a fader, the StudioMix will send a number of hexadecimal numbers on the MIDI output port. These numbers are bytes, MIDI. A Byte consists of 8 bits, which are Zeros or Ones. With 8 bits you can code 256 different conditions, for example the decimal range of numbers from 0 to 255.

Our decimal system is some kind of 'incompatible' with the computers world, so you better use hexadecimal numbers, if you don't want to juggle with Zeros and Ones.

A hexadecimal number counts from 0 to 9, then to A, B, C, D, E, F and G. So you can code 16 different conditions, which is equal to 4 bits ($0_{16}=0000_2$, $F_{16}=1111_2$).

Two hexadecimal digits are enough to represent a byte ($00_{16} = 00000000_2 = 0$, $FF_{16} = 11111111_2 = 255$).

The following table is a practical help to convert a number from its decimal to its hexadecimal representation (and vice versa). So the decimal number 120 is a composition of the hexadecimal digits **7** and **8** →

$120_{10} = 78_{16}$ (other spellings for 78_{16} are e.g. 0x78, \$78, 078h)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 – 0000	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 – 0001	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2 – 0010	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3 – 0011	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4 – 0100	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5 – 0101	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6 – 0110	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7 – 0111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8 – 1000	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9 – 1001	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A – 1010	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B – 1011	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C – 1100	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D – 1101	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E – 1110	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F – 1111	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

If you define your own messages, use only comma-separated hexadecimal values without any extensions or prefixes (no '0x', '\$' or 'h' is allowed).

The MIDI standard

In the following text you will find the basics about MIDI and the construction of MIDI messages. If you want more informations: google is your friend...

The MMS (MIDI Manufacturers Association), has standardized the MIDI interface in the 80's of the last century. It is a digital, serial point-to-point transmission with a speed of

31.250 bits per second.

MIDI permits the control of electronic instruments in realtime. Each physical MIDI port can transmit on 16 different channels. The differentiation in 16 Channels allows to generate different simultaneous voices with only one synthesizer, or to control more than one device connected to only one MIDI port (as MIDI is a point-to-point interface, the devices are daisy-chained).

MIDI messages

The communication between devices works with very short messages, consisting only a few bytes. A MIDI message begins with a status byte (Bit7 is set, so the byte has a hexadecimal value greater or equal \$80), followed by one or more data bytes (a data byte is always less than \$80).

The status byte consists information about the MIDI channel (Bit3..Bit0) and the kind of the message (Bit6..Bit4). The kind of the messages are standardized.

If you press the key C-3 on your masterkeyboard (as hard as you can), which sends on e.g. MIDI channel 1, it will send the following:

Example: C-3 is pressed

90 3F 7F Note On, MIDI channel 1, Key C3, Velocity 127 (Maximum)

A synthesizer, which listens on channel 1 and receives this message, would generate the appropriate sound.

A controller message, let's say for volume control, is similar to that: The first byte is a statusbyte, which is \$Bx, where x is the nibble specifying the MIDI channel, followed by the controllernumber, which is form \$00 to \$7F. For volume control, the controllernumber ist \$07. After that follows a value, which has a range of 7 Bit: \$00 .. \$7F, where \$00 means the volume will be zero, \$7F means maximum volume.

Example: Setting the volume of a midi channel

B1 07 55 Controller, MIDI channel 2, Volume, \$55

Some Functions afford a higher precision than 7Bit, so they simply use two 7bit values, which will be used as one 14bit value. The two parts of this are called MSB and LSB (Most Significant Byte, Least Significant Byte) or Coarse and Fine. Sometimes you will read High and Low. With 14bit, you will have a range of 16384 different values.

A good example for this type of messages is the modulation wheel, which has the controllernumber \$01 for the coarse and \$33 for the fine part of the value.

Example: Modulation wheel sends the value 578 (%0000100 1000010)

B0 01 04 Controller midi channel 1, Mod Wheel coarse, \$04
 B0 33 42 Controller midi channel 1, Mod Wheel fine, \$42

Info: $578_{10} = \$04 * 2^7 + \42
 or like this: $578 = (0 * 16 + 4) * 128 + (4 * 16 + 2)$

Sometimes you don't need to send both the MSB and the LSB parameter, if you only send one of them, the other part will not be influenced and remains at its last value. If 7bit precision is enough, you will only send the Coarse Parameter, the LSB Parameter will not be changed, if the midi implementation is programmed correct.

All 128 different controllernumbers are assigned to specific functions, defined by the MIDI specification. The Controllers \$62 - \$65 have a special significance: \$62 and \$63 are the MSB/LSB of the so called 'non registered parameter', where \$64 and \$65 are the MSB/LSB of the 'registered parameter'. The RPN/NRPN MSB and LSB are specifying the function, the value will be specified in form of a data entry coarse/fine message. 'Registered' parameters means registered by the MMA, whereas 'non registered' parameters can be used from each manufacturers without informing the MMA.

Example: Specifying the range of PitchBend

B0 65 00 Controller midi channel 1, RPN coarse, \$00
 B0 64 00 Controller midi channel 1, RPN fine, \$00
 B0 06 02 Controller midi channel 1, data entry coarse, \$02 (2 halftones)
 B0 26 04 Controller midi channel 1, data entry fine, \$04 (4/128 halftones)

To save time, the statusbyte needs only to be send if it changes.
 So the following example has the same effect than the previous one:

„B0 65 00 64 00 06 02 26 04“.

With its original firmware, the StudioMix only sends NRPN-Messages defined by Peavey.

\$BF	\$63	\$0B	\$62	\$5C	\$06	\$7F	\$26	\$7F
Controller, channel 16	NRPN highbyte follows	Highbyte (Parameter Nr.)	NRPN lowbyte follows	Lowbyte (Parameter No.)	Value highbyte follows	Highbyte (Value)	Value lowbyte follows	Lowbyte (Value)

*Table 1: Example of a NRPN-Message:
 Moving one of the rotary encoders one step to the left*

Not all kind of music software can interpret these kind of messages. Some software, which can be taught to use some midi messages to control their functions, has problems with such long RPN- or NRPN-messages. It seems, that some programs only look for the statusbyte and the twop following bytes, so such a program would only see \$BF \$63 \$0B, to stay at the previous example.

So you have to assign volume-messages or generic controller to the faders, and perhaps panorama to the rotary encoders, to use such a software with the StudioMix.

A fadeOS message can have a maximum count of 16 bytes.

The following messages are predefined, you can use it directly:

- The default NPRN messages to use it with Cakewalk.
- Note on/off, midi channel 1, notes 64 .. 7F (hexadecimal)
- Volume midi channel 1 .. 16
- Panorama midi channel 1 .. 16
- PitchBend midi channel 1 .. 16 (a good example, how to transmit values with 14bit resolution)
- 'Generic controller' :-) 00 .. 0F, midi channel 1

You can extend the groups and configurations with your own messages.

4. Construction of messages

Modes

Like you can see in the following window, a messageset is consisting of one or two messages, depending of the defined mode.

- You can turn a rotary encoder to the left or to the right, each event can cause a message to be send (relative, step up/down), so you have to define two messages. The second way is to let the rotary encoder count the absolute steps and send this 7bit or 14bit value. In this case you only need to define one message including a placeholder which will be replaced by the actual value when sended.
- Faders only send one message with their position, either in 7bit or 14bit resolution.
- The jogwheel is very special: You can define 16 different messages, one for each jogwheel position.

A message is always a sequence of comma-separated, hexadecimal notated bytes.

If you use the modes 'Analog, absolute 7bit' or 'Analog, absolute 14bit', you have to insert placeholders for the positional value (only the placeholder F1 for 7bit messages or F1 and F2 for 14bit messages).

What to send depends of the device you want to control. Look on the last pages of the manuals (for the devices you want to control) for the midi implementation charts.

Placeholder

Generally, the StudioMix will send the defined message byte after byte, but as you have seen, some bytes have special meanings:

F1	F1 is a placeholder and will be exchanged for the coarse part of your 14bit value or for the only value of your 7bit message.
F2	F2 is the placeholder for the fine part of your 14bit value.
FE	internal use, don't enter in message definitions. Avoid FF (which is midi reset) and F7 (which is midi clock), too.

Example: The definition of the message 'MIDI Volume Kanal 1' is:

B0, 07, F1

so each movement of the fader will send the bytes B0, 07 followed by the faderposition as 7bit value.

The next example is a standard NRPN message, which sends its value in 14bit resolution:

Range

Perhaps you don't want to use the complete range from 0..16383, to send a faderposition. Maybe you want to control a filter only in a little range, let's say from value 1000 to value 2000. Since V3.02, you can define those range limitations for faders and rotary encoders.

If you use the input fields inside the message definition groupbox, your range settings will be saved inside the message definition. So each configuration, which uses this message, will have the limited range.

If you don't want that, you can use the alternative range settings, which will be saved in the actual configuration, not in the message definition. So all other configurations will not be affected.

The rotary encoders has to additional parameters. The value in the editfield 'Init' is the initial value of this knob after power on. So, if you assign a function to a rotary encoder, which has a center position, you can define it here.

Also it's possible to enter a stepsize for rotary encoders. For example, if you enter 100, each snap to the right/left will add or subtract 100 from the actual position value.

The faders ignore 'init' and 'step'.

In the following picture you can see the settings described in the example above.

If you press the button 'normal', the fader will get its hole range back.

The screenshot shows a configuration window with two main sections: 'Select message' and 'Edit message'.
 - 'Select message': Group is 'Standard Peavey', Message is 'Standard Peavey NRPN Fader 1'.
 - 'Edit message': Fader (high: F1, low: F2) is 'BF,63,1B,62,2C,06,F1,26,F2'. A green checkmark button is circled in red.
 - 'Mode' section: Radio buttons for 'Analog, absolute 14 Bit' (selected), 'Analog, absolute 7 Bit', 'Button or digital value +/-', and 'Jogwheel position (-8 .. 7)'.
 - 'Range' section: Radio buttons for 'full', 'scaled, abs.' (selected), and 'scaled, %'. Input fields for Max (2000), Step (1), Min (1000), and Init (0) are visible.

Important: All settings will just be saved, if you press the button containing the green hook.

You can also define the range by entering values on a percentage basis, if you select 'scaled, %'.

The following is ambivalent to the last example, but the range limitations will be written in the configuration instead of the message definition.

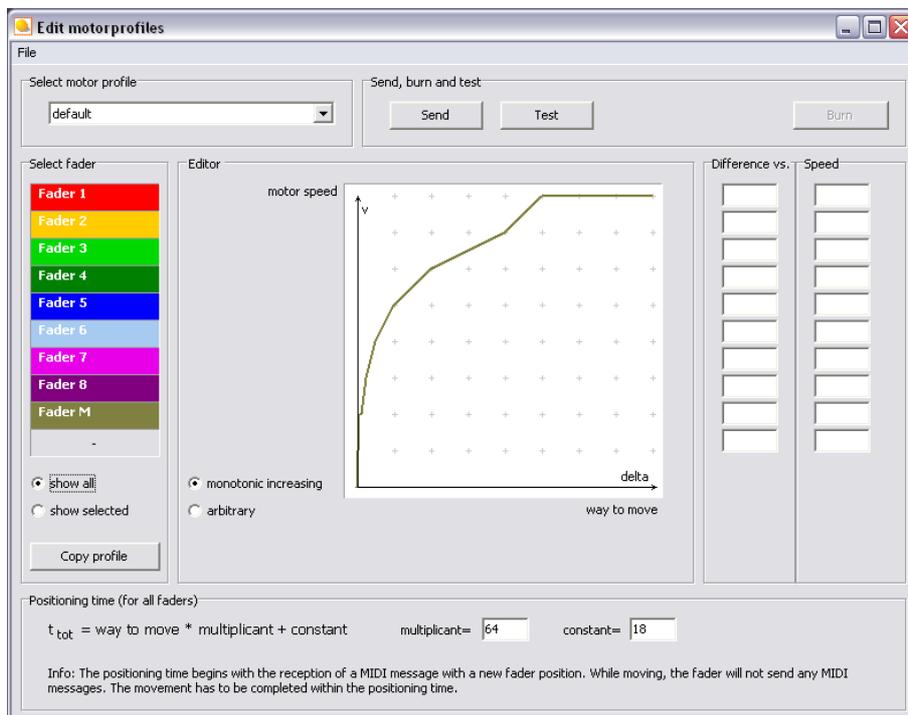
This screenshot shows the same configuration window as above, but with additional options for range selection.
 - 'Select message' and 'Edit message' sections are identical to the previous image.
 - 'Range' section: Radio buttons for 'from message' and 'from configuration' (selected). A green checkmark button is circled in red.
 - 'Alternative (configuration)' section: Input fields for Max (2000), Step (1), Min (1000), and Init (0) are present, mirroring the settings in the previous image.

You also have to press the green hook if you want your settings be saved.

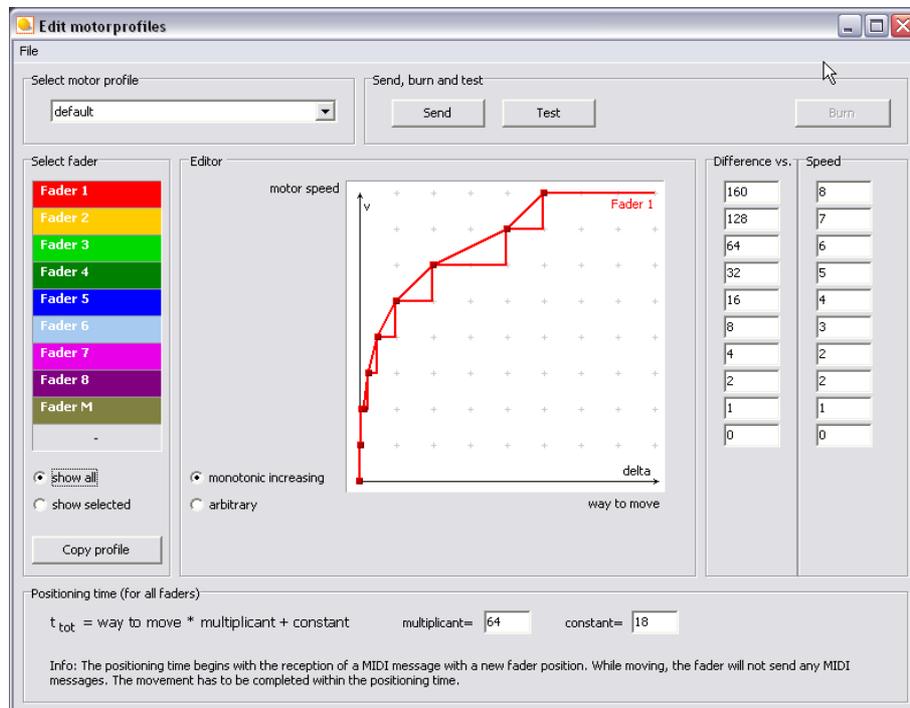
5.Motor profiles

Since version 3.02, there is the possibility to define your own motor profiles (Menu 'Motor profiles').

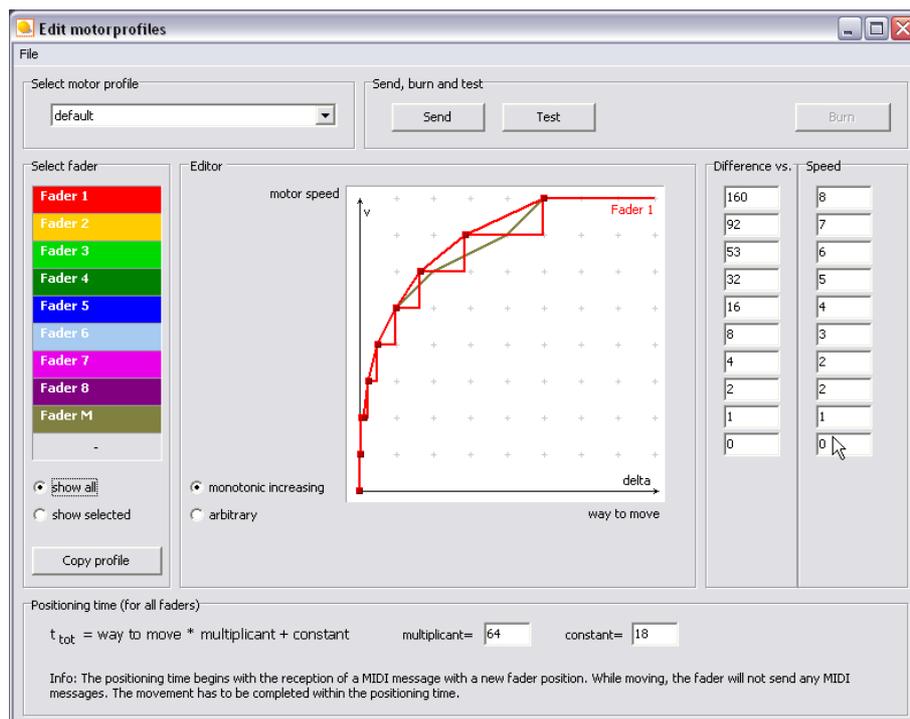
After the window has opened, you'll see the following: The 'default' motor profile is opened, all profiles are plotted one upon the other. Because all of them has the same shape, you'll see only one curve.



Now click on the red colored panel with the caption 'Fader 1'.



Now the curve shape for the motor of the first fader is selected, the quadratic spots can be moved with the mouse when holding the left mousebutton. You see the resulting vectors in the table on the right, you can also enter values here.



In the graphic representation the x axis is indicated as 'position difference', meaning the difference between the actual value and the desired value, received via midi.

The higher the difference, the faster the motor should turn to move the fader to the destination in sufficient time.

The speed of the motor is layed on the y axis, the higher the value, the faster the motor will drive.

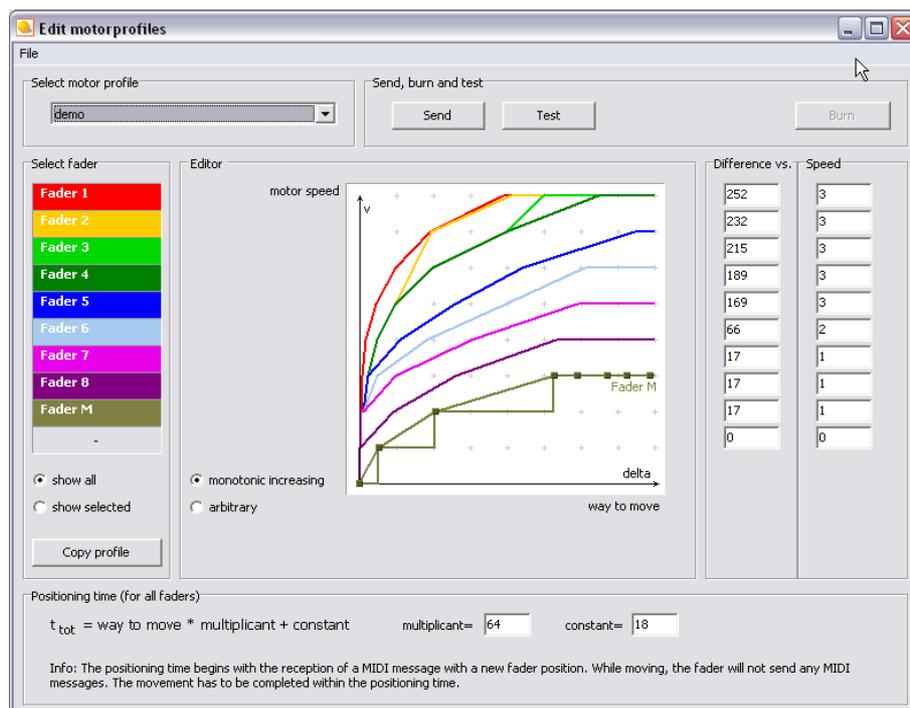
Use monotonic increasing, deselecting it can make the positioning of the quadratic spots with the mouse more simple, especially if they lie on top of each other.

You can copy the profile of the selected fader to another one, if you click on the copy button, followed by a click on the panel of the destination fader. You can also copy the profile to all other faders, you will see how after clicking the copy button.

In the groupbox 'positioning time', you can enter two values which affect the duration of the time, which controls the motor power. After receiving a new desired position via MIDI, an internal timer for this fader begins to count down. if it reaches zero, the motor stops. The time definition consists of a constant value and another value, which will be multiplied by the difference between actual and desired position. Try it out...

In the 'file' menu you'll find the menu items to save or delete your configuration.

With the 'Transmit' button you can send your complete profile, consisting the curves for all motors, to the StudioMix. You can use it and test it now, but it is not volatile. If you switch off the StudioMix, the Motorprofile will be restored to the default when switching on the next time. To avoid this, press the button 'Burn' after transferring the profiles.



The button 'test' is a very simple possibility to test the motor behaviour. When you click on it, the editor will send the midi volume messages for the midi channels 1..9 with the value 0x40, which means (if you didn't limited the range): Drive to the center position. So, if you move all faders up or all faders down, you will see whether your profile is working properly or not. To use this test, you have to assign the midi volume messages for the channels 1..9 to the faders before.

With the 'demo' profile, you can get a feel for the motor profile definition. Try it out and test it like described above.

6.MIDI monitor

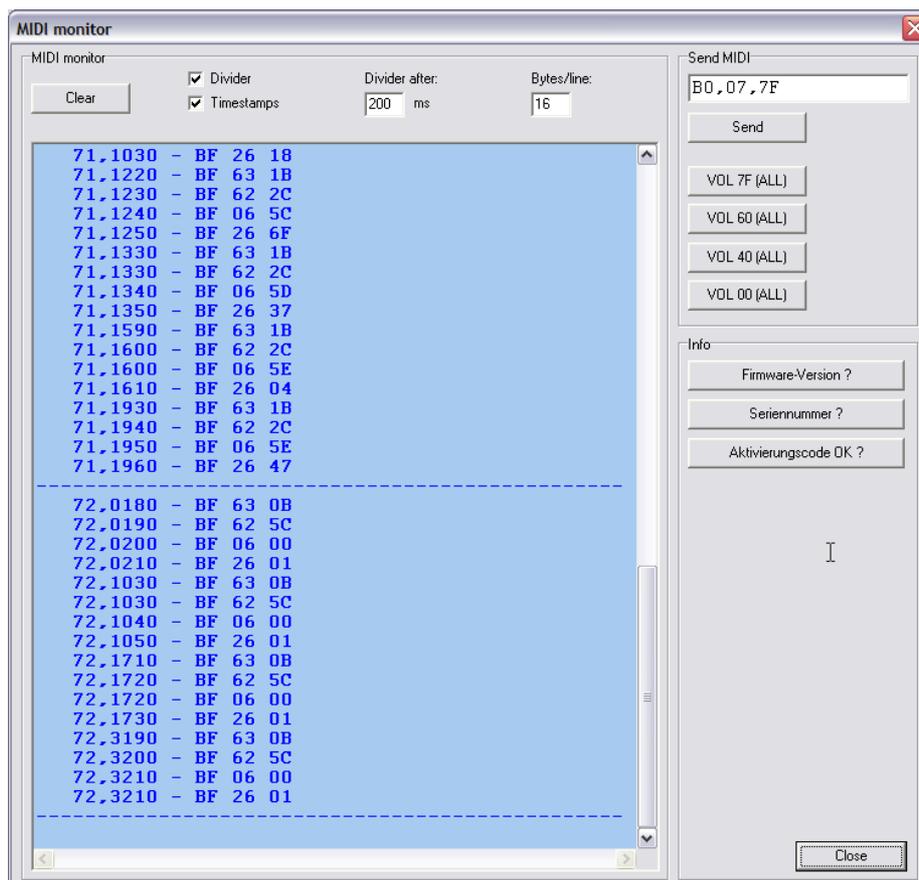
Menu -> MIDI monitor.

Now, if you press, shift or turn any input element of your StudioMix, you'll see the corresponding midi data.

With 'Timestamps' active, each data packet gets a prefix with a relative timestamp in seconds, milliseconds.

The divider is a line, which appears after some milliseconds of inactivity. So, if the monitor displays a midi message which is made up of more than one line, it provides a better oversight.

With the edit box 'Send' you can send an arbitrary sequence of bytes to the StudioMix to test something. The input string has to be a sequence of comma-separated, hexadecimal values like in the example.



The volume buttons will send the volume value from their caption on the midi channels 1 to 9 – Good to try out the motorfaders and the motor profiles.

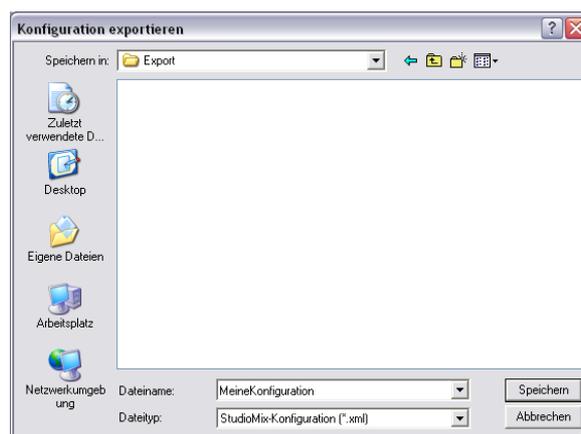
7.Import und Export

Export a configuration

In the following example you'll see how to export a configuration, if you want to exchange especially this configuration and not your complete StudioMix.xml with other users.



Give it a name:



Now you can send it via Mail. Another User can import the file in his own configuration.

Export a group

(Group -> Export)

The same for a specified group. The result is a group with all used messages, but no

configuration.

8.Q&A

Rien ne va plus!

When everything is blocked up, don't panic!
You can simply restore the default configuration.

- Switch the StudioMix off
- Press the button of the first channel and hold it
- Switch the StudioMix on
- Fader1 drives up
- Release the button
- Fader1 drives down after a few seconds.
- You can use the StudioMix.

The configuration and the motor profile is set to default values.

Faders and Knobs doesn't send any messages

Maybe you haven't activated your firmware? Please read the chapter 'Software – activation', beginning at page 4.

Faders doesn't send positions

Set the mode to 'Absolute 7Bit' or 'Absolute 14Bit'.

Use the placeholders F1 or F2.

Open the MIDI monitor and look what happens if you move the fader.

Can't alter configurations, groups or whatever

Maybe you made a backup of your editor on CD-R and restored it. The files (especially StudioMix.xml) now are write protected. Remove the write protection.

Assign SysEx Data

You can send SysEx data with the buttons, faders and rotary encoders, but maybe the maximum length of each message (16 Bytes) is a limiting factor.

However, faders, buttons and rotary encoders can not receive SysEx data for internal reason.

Other questions?

If an unknown problem appears, write a mail to me (info@sonicware.de).

Have fun!