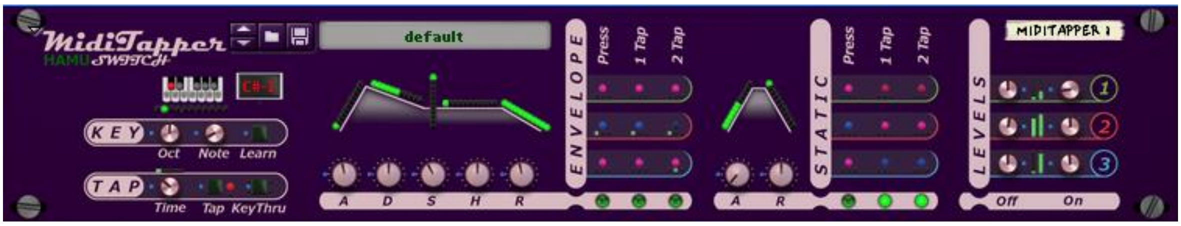
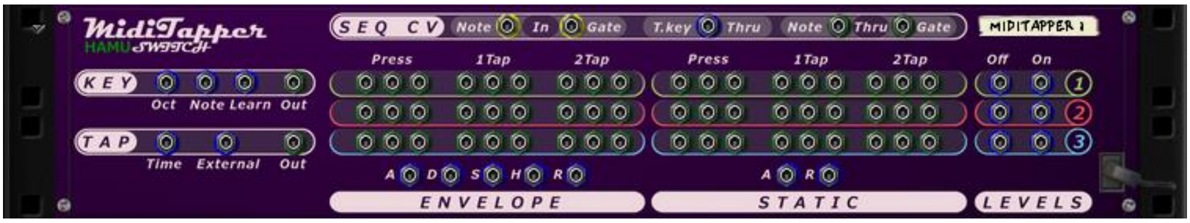
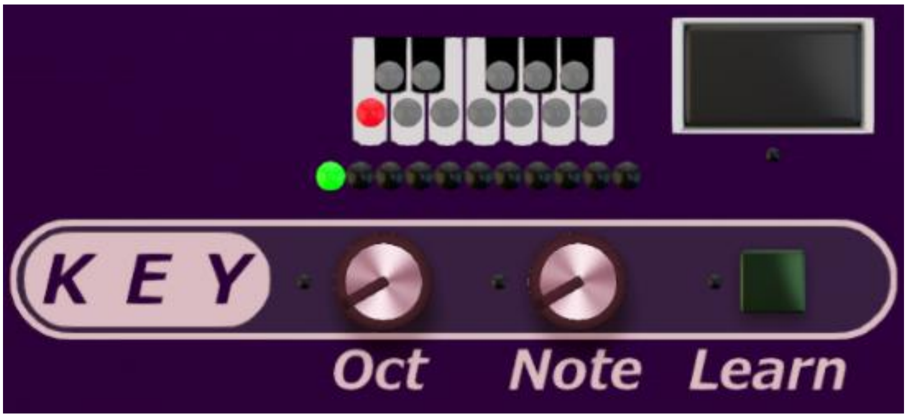
MidiTapper Manual

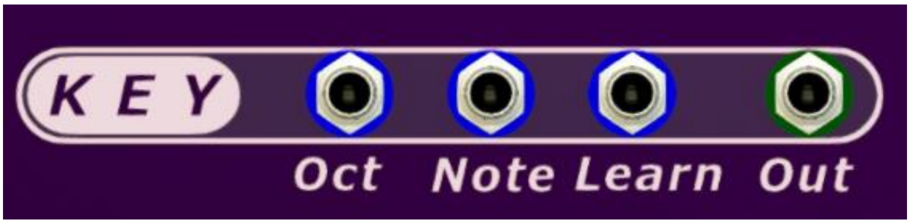




Key section



Front



Back

This is where the Tap Key is specified. The Tap Key is the note that activates – triggers – the device. This trigger key can be provided from midi input or from sequencer CV input.

The Key section includes three user controls. The octave knob, the note knob and the learn button. It also includes several feedback widgets, namely a LED bar indicating the selected octave, a keyboard pictogram with 12 LEDs showing the selected trig note and a display showing the selected key as text. There are also cable indicators showing which of the modulation (blue LED) and Tap Key out (green LED) cables are attached.

Octave knob

With the Octave knob you select octave from -2 up to +8. The chosen value is shown by a green lit LED in the LED above the knobs, as well as in the Tap Key display above the Learn button. The setting of the Octave knob is saved and restored with patches.

There is a modulation input socket for the octave, so you can use CV to set the octave. When selected by CV, the octave is displayed and used as if set by the knob, but it will not be saved with patches.

Note knob

With the Note knob you select the note of the Tap Key within the selected octave. Selection is shown in the keyboard pictogram above the knobs and in the Tap Key display to its right. The setting of the Note knob is saved and restored with patches.

There is a modulation input socket for the note, so you can use CV to set the note. When selected by CV, the note is displayed and used as if set by the knob, but it will not be saved with patches.

Learn button

The Learn button is momentary, stays on only while you push it, and is used to set the Tap Key from the incoming note. Both midi input and sequencer input is listened to – and shown in LED bars and a text display – while you push the button. The last received note before you release the Learn button will be used as Tap Key. The Tap Key selected by the learn function is not saved with patches, but loading a patch back will always select the Tap Key according to the Tap Octave and Note knobs settings of the saved patch.

User input, panel meters, modulation inputs and patch files

The front panel of an RE consists of different 'widgets' (parts) which can either input widgets for the user to manipulate, or feedback widgets where information about the inner status of the RE is displayed for the user. A typical example from the MidiTapper is when a knob is rotated to a specific position, this is obvious by the looks of the knob. It is not, however, possible for the RE to turn a knob in response to a modulation signal. Instead we use lamps, meters, LED bars etc. to show on the panel what value is currently used for the parameter.

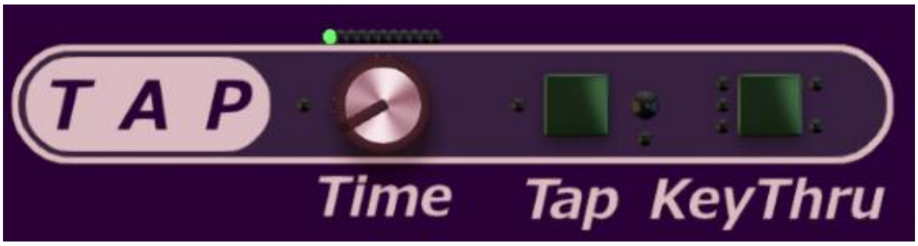
In most cases, on the MidiTapper, any modulation values are *added* to the value set by a knob, so it makes most sense to use bipolar signal to modulate parameters.

A patch file will by definition only save the settings of user input widgets of the rack extension, i.e. buttons, knobs, faders, displays for user input etc.

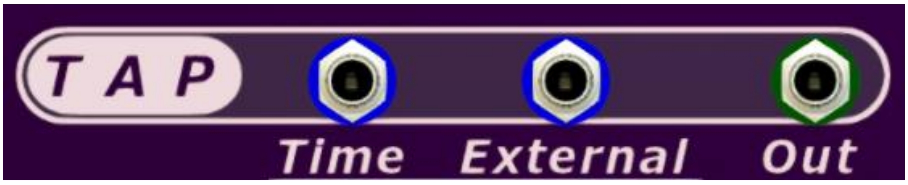
With a device like MidiTapper, having lots of modulation inputs, this means that only the basics will be save in patches. A limited number of patch files can be useful as you build a combinatory setup from scratch, or add a MidiTapper to an existing setup. Examples are include in the RE download, but you may want to create your own. After a MidiTapper has been fully integrated in a setup, it may be more useful to think in terms of variation of certain parameters, and maybe try out and save patches fit for the specific environment it's working in. A setup could become completely erratic if you load a totally different patch into a MidiTapper after the setup is already fine tuned.

So; save variation to MidiTapper patches only for quick variations, and save combinatory patches to preserve the environment setup.

Tap section



Front



Back

The Tap section is where tap inputs are detected and converted to trig events.

A tap can be a detected tap key on either midi or note CV inputs, or from the direct tap button. Depending on the times between taps in a sequence, and the selected tap timeout setting, one out of three kinds of trig events will occur. These are the ‘Press’, ‘1 Tap’ and ‘2 Tap’ events, similar to the computer mouse events of ‘click and hold’, ‘click’ and ‘double click’. These trig events are channeled on into the Envelope and Static sections.

The Tap section includes three user controls. These are the Time knob for the tap timeout, the direct Tap button and the Key Block button. There are the usual cable indicators showing which of the modulation (blue LED) cables are attached, but also several other LEDs which are described in more detail along with the user controls below.

Tap Time knob

The Time knob specifies how long the tap-to-trig converter should wait before deciding what kind of trig event a tap input sequence represents. There are 25 steps to this knob, selecting one of the following times: 0.00s, 0.01s,

The selected time is visualized in the bar of 13 LEDs above the knob.

There is a modulation input socket for the tap time, so you can use CV to select the time. When selected by CV, you still have the same 25 times to choose from, and the socket input level is rounded to the nearest time. The selection is indicated in the LED bar and used as if set by the knob, but it will not be saved with patches.

Tap button

The tap button is a direct tap input, and can be used in parallel with the midi and CV note inputs. This is as expected a momentary button. Since you cannot press the tap button and save to a patch file simultaneously, this button is never pressed when you load a patch.

There is a modulation input socket for the direct tap too, so you can create trig events from a common on/off CV signal. Since all tap methods work in parallel, you can create a programmed sequence of trig events via this modulation input, while creating additional trig events to the same output from your tap key.

Tap output

There is a red LED beside the Tap button indicating the tap status (on or off, pressed or released). This is useful as an indicator, but there is also a direct output socket corresponding to this LED, with its own green cable connection indicator right below.

Tap-to-trig conversion

Here we describe in detail how a tap input sequence is interpreted into a specific trig event.

For tap inputs every signal flank is counted as a step, for instance when the Tap button is pressed this is one step, releasing it again is next step. Starting from a status of 'no tap input', steps are counted as long as there is a new tap input before the timeout is reached. If 4 tap steps are counted without any timeout, a 2Tap trig is immediately generated and the tap step counter is reset to 'no tap input'. The following list shows what trig event is generated from each possible tap sequence:
1 tap step before timeout → a Press trig event
2 tap steps before timeout → a 1Tap trig event
3 tap steps before timeout → a 1Tap trig event *and* a Press trig event
4 tap steps before timeout → a 2Tap trig event

A long tap sequence example

Keep on tapping without waiting for a timeout, and you will get several trig events. For instance 7 quick tap steps will create all three trig events. The first 4 trig a 2Tap event, while the following 3 taps trig both the 1Tap *and* the Press events. If you started with tap input 'off', you will after 7 tap steps end up pressing the tap key or tap button. In other words you made a double tap, followed by a single tap and finally a press.

Inverted tapping

Since the tap-to-trig converter counts flanks (note off as well as note on), it's perfectly possible to create 1Tap and 2Tap events also while keeping the tap key pressed. For instance a quick release and then press again, will create a 1Tap trig event, while the Press switch remains unaffected.

Tap Key Thru button

This toggle button is used to block or allow the tap key input to pass through the device to the sequencer CV note and gate outputs. Normally you don't want the tap key to be heard, but you might find interesting use of doing so. For instance you can use a number of MidiTappers to apply different – but predefined – effects for different notes. Maybe do some detuning for micro intervals?

There is a modulation input socket for the Tap Key Block button, with corresponding blue indicator close to the button. I cannot advice you on how to use it, it's there because it *should* be there, and I have great confidence that one fine day some creative user will find a use for it.

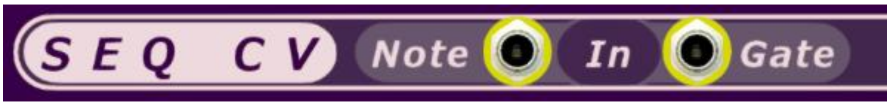
Sequencer CV inputs and Thru outputs

The cable lamps are in the Tap section, beside the Tap Key Thru button. Please refer to next section for more details.

Midi input, sequencer CV input and sequencer Thru output

The sequencer thru output, mingling midi and sequencer input, is a side effect benefit of the MidiTapper. More details about this in this section.

Note and gate inputs

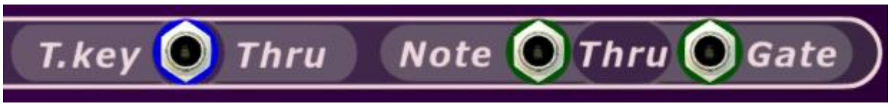


Back

There are two ways of feeding note and gate information to the MidiTapper. Primary input is – of course – the midi signals received by the device. Secondary input is from the sequencer CV input sockets.

Midi and CV inputs work in parallel, so it's perfectly possible to trig a regular pattern of MidiTapper tap events from a Matrix or similar, while adding triggers manually from midi input.

Note and gate outputs – 'Thru' output sockets

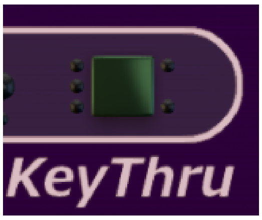


Back

The 'thru' outputs are combined from the two input sources. This enables you to not only trig the MidiTapper envelope and static outputs, but also 'mingle' the 'melodies' from sequencer inputs with the notes from the midi input.

The output is a sequencer CV flow including your midi input, which can be connected to instruments, additional MidiTappers or whatever you please. In this way several MidiTappers can be chained and react to different keys in the flow.

Using the Key Thru button



Front

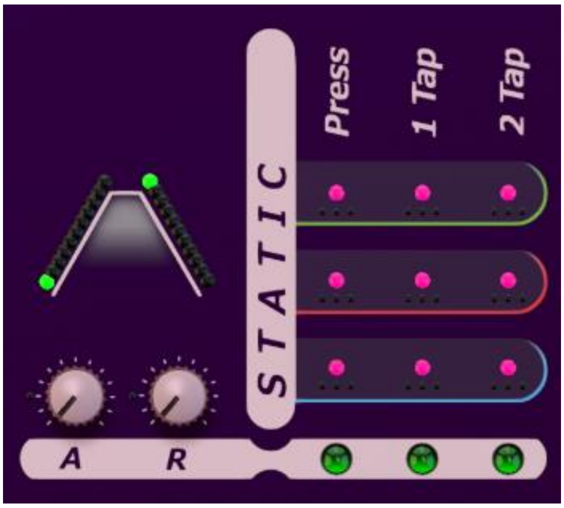
Depending on how you use the sequencer thru output, you may want to include or exclude the tap key note events from the flow. I must urge you to experiment with this, and remember; for each MidiTapper you can decide if it should receive input from midi, sequencer CV or both.

Static and one shot Envelopes

When a trig event (Press, 1Tap or 2Tap) is detected, the corresponding output levels will traverse the corresponding curve. Common for both envelope and static outputs is that the transition time for each envelope section can be varied in 25 steps from 0s to about 12s.

Should a new event occur before the previous transition is completed, the new transition will be from the current level to the new 'target' level. This means that you have no surprises like sudden jumps in the output levels. Modulations of levels are handled in the same way, so if a transition time of MidiTapper is slower than the changes of an applied level modulation, the output will still never move faster than set by the MidiTapper transition times. If you want to modulate levels at a higher frequency you should try something like a Mesher device *after* the MidiTapper.

Static curve for switches

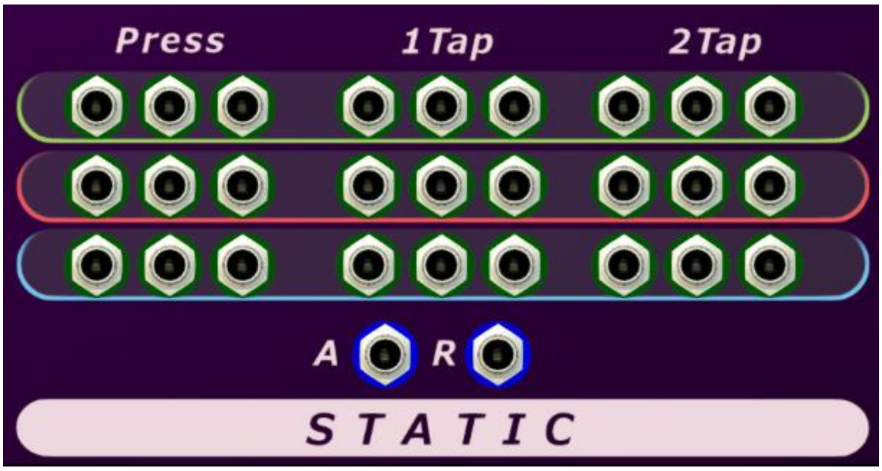


Front

The static outputs have only attack and release timing, with the A and R times set by the knobs.

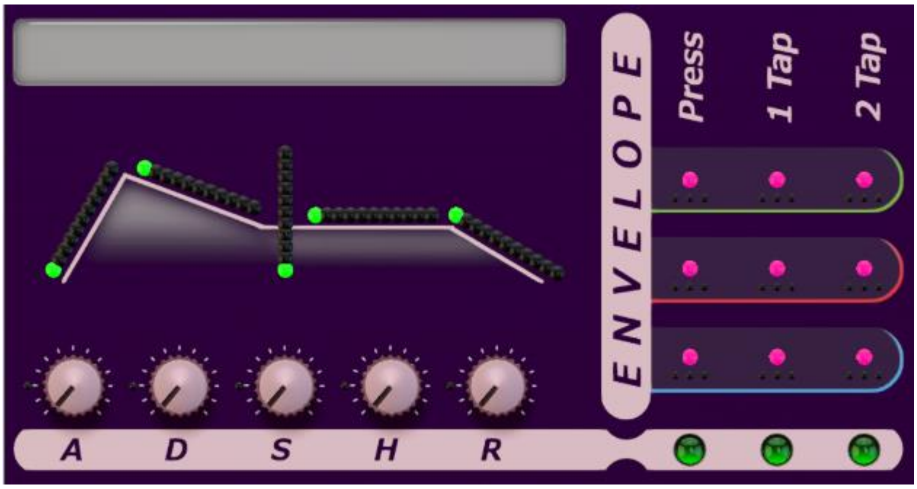
Press switch goes from off to on level at the attack rate when the tap key is pressed, and back to off level at the release rate when you stop pressing.

The 1Tap and 2Tap switches are turned on or off with each single tap or double tap event, i.e. they switch back and forth with the transition times set by each of the knobs.



Back

Envelope for one shot pulses

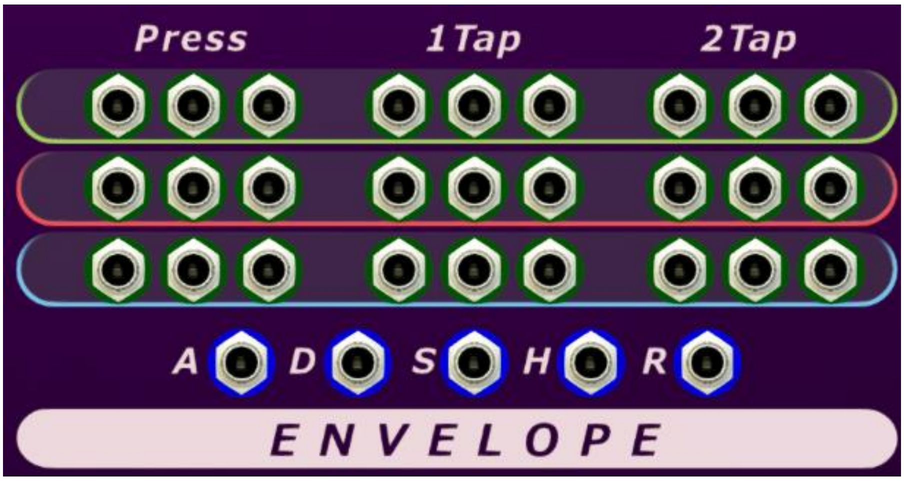


Front

The 'Envelope' section applies to the one shot output only. It's a classical ADSHR curve, where the sustain level is set as percentage of the distance between the 'off' and 'on' levels.

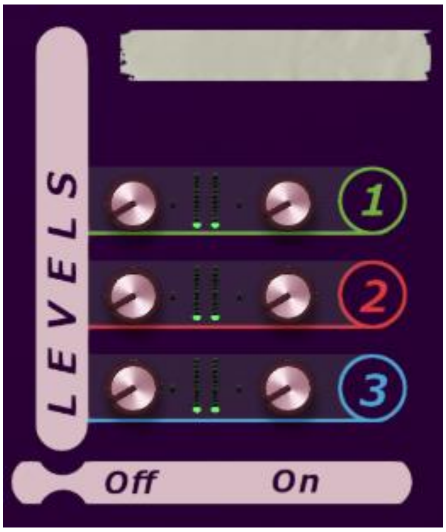
A special case occurs when you set the attack to 0.00s. With this setting, the envelope will start from 'off' level, but never move to the 'on' level. Instead the first move will be from off to sustain level, at the rate set by the decay. If you really want to have an 'immediate' transition to the 'on' level, you should set the attack to 0.01s, which will go to 'on' in a short enough time to be inaudible.

There is a difference between the Press on one hand and 1Tap and 2Tap on the other. The Press envelope is triggered 'only' when the tap key is pressed, not when it's released. 1Tap and 2Tap envelopes are triggered each time you tap or double tap, completely independent from the status of the corresponding static switch.

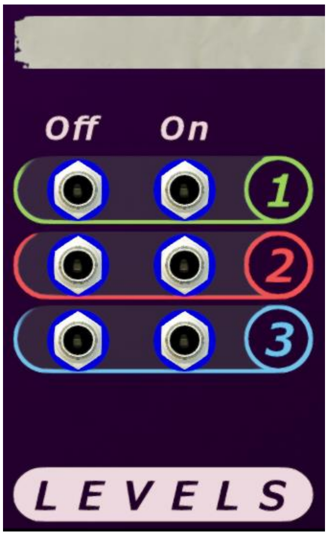


Back

Output levels



Front



Back

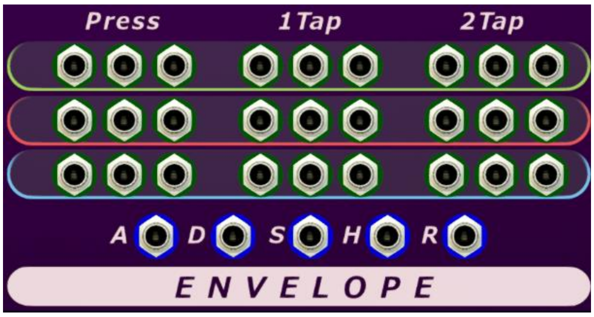
Three output level groups

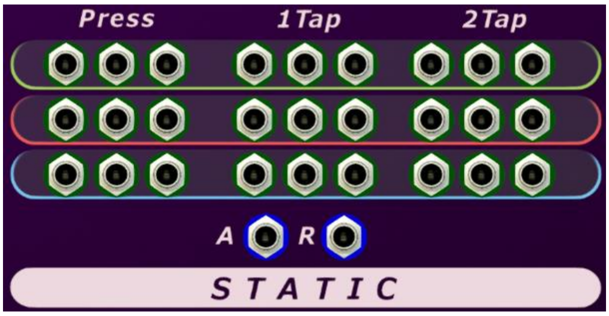
The MidiTapper has three separate output level groups, with separate levels for each group.

There are levels for ‘off’ and ‘on’ statuses, for the three groups.

Output sockets

All output sockets on the MidiTapper are marked with a green color, and are found in the envelope and static section of the back panel:





The colored lanes correspond to each level group, and there are three socket copies for each output signal.

Defaults and modulation

The default patch, and ‘reset device’ will set the levels to 0 / 1 for group 1, 1 / 0 for group 2 and -1 / 1 for group 3. This gives a unipolar level for group 1, inverted unipolar level for group 2 and bipolar level output for group 3.

Modulation can be applied, but remember this is mainly for slow adjustments, and will be affected by the transition times of the envelope and static curves.