Flux

Liquid Tremolo MIDI Specification

MIDI In/Out/Thru

- The Liquid Tremolo responds to MIDI messages on its MIDI IN according to the documentation below.
- The Liquid Tremolo provides a MIDI THRU signal and therefore does not chain the incoming MIDI IN messages on its MIDI OUT.
- The MIDI OUT will send MIDI Clock messages (which are completely independent of MIDI Clock in) so that it can be used as a master sync.

MIDI In - Controller Change messages

Parameter	**CC#**	**Values**	**Comments**
Mode	19 13h	0-3	0 = Live (use knob setting) 1 = Manual 2 = TapSync 3 = Expression
Speed MSB Speed LSB	17 11h 49 31h	20-1440	Value is in BPM Send LSB first then MSB*
Depth	18 12h	0-127	0 to 40.5 dB
Contour	16 10h	0-8	0 = Live (use knob setting) 1 = Spin, 2 = Smooth, 3 = Thick 4 = Hard, 5 = Slice, 6 = Plink 7 = Whip, 8 = Custom
Stereo Field	31 1Fh	0-127	0 to 180°
Tap Tempo Divider**	20 14h	1-4	Denominator for bpm calcs (Numerator for tap period calcs)
Tap Tempo Multiplier**	21 15h	1-6	Numerator for bpm calcs (Denominator for tap period calcs)
TapSync/ReSync	93 5Dh	Any value	
Expression	100 64h	0-127	0 = Heel down 127 = Toe down
Bypass	102 66h	0 = Bypass 127 = Active	
MIDI Clock Receive On/off	51 33h	0 = Ignore 127 = On	Sets base tempo Default = On
MIDI Clock Transmit On/off	52 34h	0 = Off 127 = On	Follows base tempo Default = Off
MIDI Start Receive On/off	53 35h	0 = Ignore 127 = On	Re-sync's the contour Default = On
MIDI CC Transmit On/off	54 36h	0 = Off 127 = On	Sends CC on own channel Default = Off

* When the LSB is received it is held in waiting until the MSB is received at which point both are applied in unison.

** The combination of multiplier and divider give a wide range of tap sub divisions; for example dotted-eighths would use a multiplier of 4 and a divider of 3. In most cases the divider is set to 1 and the multiplier determines the final tempo. The final speed will be coerced within the allowable range if the combination of multiplier, divider and base tempo cause it to fall out of range.

MIDI In - Program Change messages

Preset	**PC#**	**Comment**
Engage Live	0	Revert to Live mode (local controls)
Engage Preset	1-48	Engage a stored preset
Save Preset	1-48	Saves the current settings to given PC# if *Bypass* & *Tapsync* are both held down. Note that PC#1 is the slot used for the local "FAV" preset. IMPORTANT: You must send PC# message within 5 seconds of holding down *Bypass* & *Tapsync* or the tremolo will automatically save to PC#1.

MIDI In - Realtime messages

Realtime	**Default**	**Comments**
Clock (F8) Receive	On	The received clock sets the base tempo
Clock (F8) Transmit	Off	The base tempo sets the clock for MIDI OUT
Start (FA) Receive	On	Causes the tremolo wave to re-sync; unlike CC#93 the speed will not be affected if *Start* is received in multiples
Reset (FF) Receive	On	Reboots the tremolo control system

MIDI Out - Clock messages

The Liquid Tremolo sends MIDI Clock messages on MIDI OUT so that it can be used as a master sync. MIDI Clock Out follows the base tempo (i.e. before multiplier is applied) and is completely independent of MIDI Clock In. MIDI Clock Out is off by default and can be turned on/off via CC#52.

MIDI Out - Controller Change messages

The Liquid Tremolo sends CC messages on MIDI OUT whenever a parameter is changed via local controls, preset engage, or direct MIDI CC receives. However, it is important to note that it sends them on its own channel. These are intended for future accessories such as a BPM readout for example. MIDI CC Out is off by default and can be turned on/off via CC#54.

Cabling

The Liquid Tremolo is shipped with an interface cable that connects a standard MIDI cable to the Liquid Tremolo MIDI IN. If you require connection to the Liquid Tremolo MIDI OUT and/or MIDI THRU these are available separately in our webshop for a small cost.

MIDI In - System Exclusive messages

Custom Contour Upload	
Hex	F0 00 21 21 01 1E 12 01 00 00 [*48 byte-pairs*] F7
Dec	240, 0, 33, 33, 1, 30, 18, 1, 0, 0, [*48 byte-pairs*], 247
Custom Contour Boost Value	
Hex	F0 00 21 21 01 1E 12 01 00 30 *XX YY* F7
Dec	240, 0, 33, 33, 1, 30, 18, 1, 0, 48, *X*, *Y*, 247

Custom Contour Upload

The data bytes are ordered in 48 pairs, each pair representing the contour wave level at equal time-divisions of 1/48th of the entire contour shape. The first byte of each pair is the MSB and is either 0 or 1, the second byte of each pair is the LSB and ranges from 0-127. When combined each value has a range of 0-255. These values are inverted: a value of 0 represents the wave crest (or unity), and a value of 255 represents the wave trough. The values are also logarithmic rather than linear, where 0 = 0dB (unity) and 255 = -29.8dB. Below are the formulae to convert between Value and Level:

$$\text{Value} = \text{LOG}_{10}(\text{Level}) / -0.005847$$ *Level ranges from 0.0323 to 1.0*

$$\text{Level} = 10^{(\text{Value} * -0.005847)}$$

The first value is the wave sync point, and is followed by 47 additional points to comprise the full contour shape.

Custom Contour Boost Value

One pair of bytes where the first byte is either 0 or 1 and the second byte ranges from 0-127, giving a range of 0-255. However, the allowable range is 7-254, and 255 is reserved for disabling the custom contour and reverting back to the factory contour. Values of 0-6 will create undesirable side effects! This value determines how much boost is applied to compensate for the volume drop resulting from the specific wave shape. There is no formula to calculate the ideal value for a given wave shape so some level of experimentation is necessary. The boost value is counter intuitive in that it works in reverse; a boost value of 254 will give no boost whereas a boost value of 7 will give maximum boost (approx +22 dB). For example, the on-board "Slice" contour which spends almost all of its time at unity has a low boost and thus a fairly high value of 145, whereas the "Spin" contour has a high boost and thus a low value of 17. Contour shapes that have roughly balanced crest-to-trough spread (such as a sine wave or triangle wave) should have a boost value of around 32, which is a good place to start when experimenting.