MPETuner

Just intonation on keyboards in real time! Microtonal modulation! Strange and novel sonorities! Welcome to MPETuner! Here's a quick guide to get you up and running. Please let me know if you have any questions.

Installation and Setup

Installing on macOS

Just run the installer.

Installing on Windows

Just run the installer. You may need to add C:\Program Files (x86)\Steinberg\VST3 to the plugin locations in your DAW.

Installing on Linux

Currently there is no Linux support for MPETuner, but if you're interested, let me know and I will try to add it!

Setting Up in Logic

1) Create a software instrument track.
2) Under MIDI FX in the track inspector, select Audio Units > dmitrivolkov > MPETuner.
3) Select an MPE-compatible instrument, such as Sampler, Alchemy, EFM1, ES2, EXS24, Retro Synth, Sculpture, or Vintage Clav. Alternatively, add an external instrument plugin (see "Setting Up Instrument Plugins" below).
4) In the dropdown at the bottom of the selected instrument's GUI window, make sure to enable MIDI Mono Mode and that the pitch bend range matches that set in MPETuner.
    a) Common base channel 1 corresponds to low MPE base channel in MPETuner.
    b) the default pitch bend range should be 48, both for the Logic instrument and for MPETuner.

Setting Up in Reaper

1) Create a new track.
2) Open up the FX rack and add MPETuner.
3) Add an MPE-compatible virtual instrument to the FX rack after MPETuner. See "Setting Up Instrument Plugins" below.
4) Make sure the MPE settings match between the virtual instrument and MPETuner!

Setting Up in Bitwig

Bitwig does not expect plugins to return MPE data, so setting up MPETuner requires using an external program to use MPETuner as a virtual MIDI device, essentially. Unfortunately, this method does not work with MIDI data already inputted into Bitwig; it only works when initially inputting MIDI data through a controller. Also, due to how Windows handles audio devices, this method does not work on Windows.

As of now, I have not figured out how to get this to work with Windows. I believe it is possible, but requires using a non-default audio driver. If you're interested in trying to get this to work on Windows, please let me know and I will gladly assist.

Create a virtual MIDI device.

a) On macOS, open the "Audio MIDI Setup" application, make sure MIDI Studio is open (Window > Show MIDI Studio), then make sure IAC Driver is active (double click IAC driver and make sure "Device is online" is checked).

2) Set up MPETuner to process MIDI through the virtual device.
   a) Open an audio plugin host that supports arbitrary input/output mapping. For the purposes of this guide, I am going to use a free release of Element, downloadable at https://github.com/kushview/Element/releases/tag/0.41.1 .
   b) Insert the virtual MIDI device, both as an input and output. In Element, right-click, then under MIDI Input Device and MIDI Output Device select the virtual device created in step 1.
   c) Insert MPETuner. In Element, right-click, then under Plugins find MPETuner (it may be under "dmitrivolkov" or "Unverified."
   d) Wire everything up. In Element, wire the output of whatever MIDI device you want to use as input to the input of MPETuner, and the output of MPETuner to the input of the virtual MIDI device. If using Element, make sure to press play at the top.

3) Add a “Seaboard RISE” controller to Bitwig, and set the input and output of this controller to the virtual MIDI device.
4) Make sure that “Seaboard RISE” is selected for any track you want to use MPETuner with. Now MIDI recorded on that track will be justly tuned!

Setting Up in Other DAWs

1) Create a MIDI track or equivalent.
2) Insert MPETuner as a MIDI effect, or at the front of the effects chain.
3) Insert an MPE-compatible virtual instrument and make sure the MPE settings match with MPETuner.

This is the general idea, but the exact process may differ from DAW to DAW. If confused, please contact me and I will try to help.

Setting Up Instrument Plugins

For a plugin to work with MPETuner, it must be MPE-compatible. Some plugins may call this “midi mono mode.” Make sure to set the MPE settings in your instrument the same as the MPE settings in MPETuner!

If you’re looking for a free plugin to use to experiment with MPETuner, I recommend the Surge synth plugin: https://surge-synthesizer.github.io/ .

Setting Up MPETuner

Set up MPETuner in a DAW or other audio plugin host as specified above. Make sure MPETuner’s MPE settings match the MPE settings of the DAW or audio plugin host you are using! If “Low MPE Base Channel” corresponds to an MPE main channel of 1, and unchecked corresponds to an MPE main channel of 16.

Next, set the default pitch shift. For tuning around A=440hz, this should be 0.0c. For tuning around A=432hz, this should be -31.77c. For tuning around A=442hz, this should be 7.85c. For tuning around A=444hz, this should be 15.67c. Honestly, I should probably change this setting just to accept the default tuning of A. That would probably make more sense.

Next, set the MIDI CCs which control key center and pitch center locking. The default is that key center locking is controlled by MIDI CC 64 (sustain pedal) and pitch center locking is controlled by MIDI CC 1 (mod wheel). The details of how key and pitch center locking is explained below.

Using MPETuner

Core Concepts of MPETuner

Just intonation: a way of tuning notes such that they sound in-tune to the human ear. While "just intonation" is a nebulous concept with no single concrete definition, this plugin attempts to achieve one form of it by tuning all notes in pure intervals to a key center.

Key center: in the context of MPETuner, this is the note to which all the other notes are being justly tuned.

Pitch center: in the context of MPETuner, this is the exact cent value to which all notes are being justly tuned.

Pitch deviations: the difference (usually expressed in cents) between how a note is being tuned and how a note is usually tuned in equal temperament. Pitch deviation can be relative to the current pitch center, or to a default pitch center such as A=440hz.

Microtonal modulation: When the pitch center changes due to changes in key center. For example, changing from a key center of C to a key center of Eb results in the pitch center increasing by 15.54 cents (because Eb justly tuned to C has a pitch deviation of 15.54 cents).

Key center locking: This prevents the key center from changing, and maintains the pitch deviations tuned to the key center. This effectively disables microtonal modulates, but also enables interestingly detuned chords (e.g. locking the key to C then playing an A minor).

Pitch center locking: This prevents the pitch center from changing. This effectively disables microtonal modulation, but the key center can still change.

Using MPETuner

In MPETuner, the key center is determined by whichever note has been held down the longest. E.g., a root position C major triad as a chord arpeggiated upwards justly tunes all the notes to C, whereas arpeggiating downwards justly tunes all notes to a G.

The pitch center or key center can be locked using MIDI CCs; the default is for the sustain pedal to control pitch center locking and the modulation wheel to control pitch center locking, though these can of course be remapped. Both pitch and key center are unlocked by default, and interacting with a MIDI CC (e.g. by pressing the sustain pedal) toggles the lock.

MPETuner Key Center Lock Modes

Default lock: the default key center lock state.

Passthrough: when enabled, the MIDI CC is passed through to the instrument, e.g. if the MIDI CC used to control key center locking is sustain and passthrough is enabled, the instrument will receive the sustain message as well, and sustain the held notes.

Lock Mode None: activating the MIDI CC does not cause any change in the lock.

Lock Mode Toggle Lock: activating the MIDI CC causes the lock to be toggled. e.g. if using the sustain pedal (CC 64) and when unlocked by default, pressing the sustain pedal will lock the key center.

Lock Mode Presustain: This “renders out” sustained notes into MIDI, in addition to toggling the lock. This can be useful when the virtual instrument does not change the pitch of notes sustained by the sustain pedal; by having those notes appear as continuously played, the virtual instrument should change the pitch accordingly. This can also be used to control the sustain of notes with MIDI CCs other than the sustain pedal. I recommend avoiding this mode unless necessary; Lock Mode Toggle Lock and Sustain Passthrough should handle most cases.

MPETuner Pitch Center Lock Modes

Default lock: the default pitch center lock state.

Passthrough: when enabled, the MIDI CC is passed through to the instrument, e.g. if the MIDI CC used to control key center locking is modulation and passthrough is enabled, the instrument will receive the modulation message as well, and modulate.

Lock Mode None: activating the MIDI CC does not cause any change in the lock.

Lock Mode Toggle Lock: activating the MIDI CC causes the lock to be toggled. e.g. if using the mod wheel pedal (CC 64) and when unlocked by default, fully activating the mod wheel will lock the pitch center.

Lock Mode Bendback: When enabled, this makes it possible to reset the pitch center to the default value by activating, then deactivating the lock. To use this, first make sure that pitch center locking is being controlled by a MIDI CC with multiple values (e.g. mod wheel, NOT sustain pedal). Then, in a place where the current pitch center is different from the default, slowly return to an unlocked state. The pitch should ”bend back” towards the default, similar to a pitch bend wheel but automatically lining up with the current pitch center and default pitch center! Once the MIDI CC is back at a minimum, pitch center locking will be toggled back to the default state.

Credits

Development

Developed by Dmitri Volkov, interning with Ben Bloomberg.

Testing

Thank you to members of The Hideaway server on Discord for testing out the plugin!