Trigger Finger

Users Guide

What is *Trigger Finger*?

Play your favorite synth or sampler by kicking a drum, plucking a string, or creating any other decidedly percussive acoustic racket. The louder you play, the louder your synth plays. As for what note it plays, pick one manually or better yet, provide your own set of pitches for it to cycle through with every hit. That's right, composition meets electronic music you actually play with acoustic instruments.

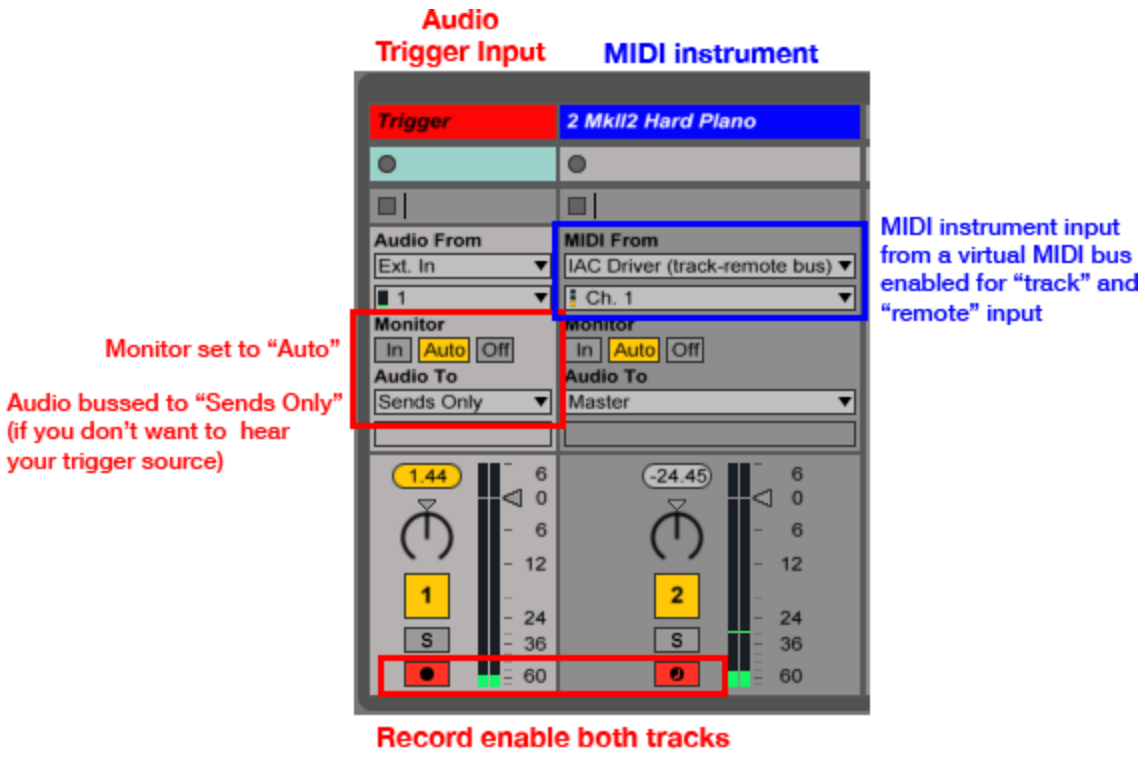
Setup

Trigger Finger is a mono Ableton Live Audio Device implemented using Max for Live (M4L), **currently available only as a Mac OS device for 32 bit Live**. M4L using at least Max version 6 is required to use this device. Installation is easy: after downloading and expanding the zip archive you'll find two items. Before opening Live, copy the **trigger-cycles.json text file** anywhere into Max's search path, for example *Applications/Max 6.1/patches***.**

Open your Live set and simply drag and drop the **TriggerFinger.amxd device** into a new audio track. If you don't want to monitor your input trigger audio, set the output to "Sends Only" and make sure the Monitor is set to "Auto". Record enable the track.

Trigger Finger is able to send MIDI notes from an audio track using the power of virtual MIDI busses. On Mac, this is trivial using the built-in IAC bus. On Windows, several free third-party options are available. First, follow this guide to setup a virtual MIDI bus for use in Ableton Live. In Ableton Live Preferences > MIDI Sync, enable this bus for both **"Track" and "Remote" Input** and **"Track" Output**. In the *Trigger Finger* device on your audio track, select this virtual MIDI channel from the **output** dropdown.

Create a new MIDI track in Ableton Live and drop an instrument in you'd like to trigger. Set the input to this track ("MIDI from") to receive from the virtual MIDI bus and the output channel that is specified above *Trigger Finger's* **output** pulldown (default is channel one). Record enable the track. Now *Trigger Finger* can send MIDI events from its audio track to this MIDI track.



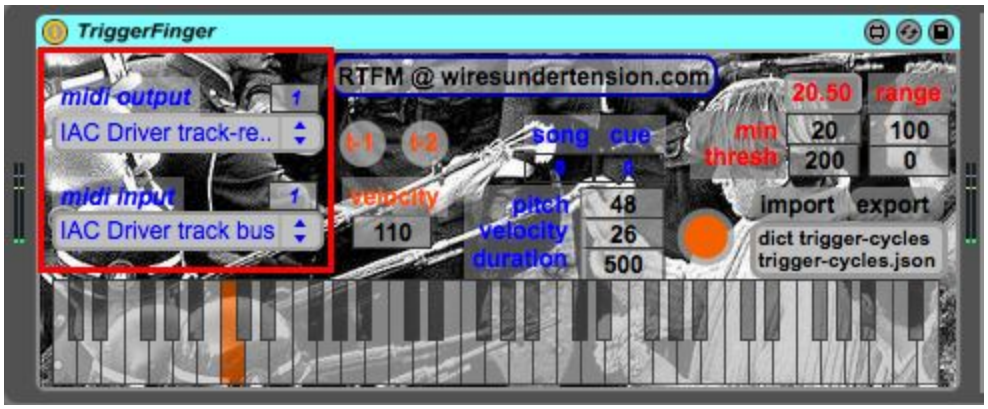
Ableton Live *Trigger Finger* channel setup

Audio triggering devices like *Trigger Finger* work best with isolated audio inputs and percussive playing, for example, tapeing a contact microphone to a drum head or using dedicated drum triggers. An open microphone will likely produce "false triggers" due to ambient noise, including the triggered sound itself (unless headphones are used for monitoring). Also, the event detection algorithm (Miller Puckette's MSP bonk~ object) looks for sharp amplitude shifts in spectrum as one would find from struck percussion. It's not meant to detect note changes within cello runs for example and will likewise produce indeterminate results. If such hijinks are what you're after, indeed, ignore these guidelines!

Because this device triggers from audio in real time, lower audio latencies will be preferred. In Live's Audio Preferences using the smallest audio buffer size possible for your set will offer the most responsive results.

The device uses audio input via the left channel. Stereo output is passed through for ease of use with subsequent devices but as the source is typically a drum trigger, track audio output will typically be disabled by the user (set to "Sends Only"). The device is designed to be cpu friendly; running Live 9.18 on a 2.66 GHz Intel Core i7 MacBook Pro with Max OS 10.6.8, using this device alone with the built in Core Audio driver drove Live's CPU meter to 1%.

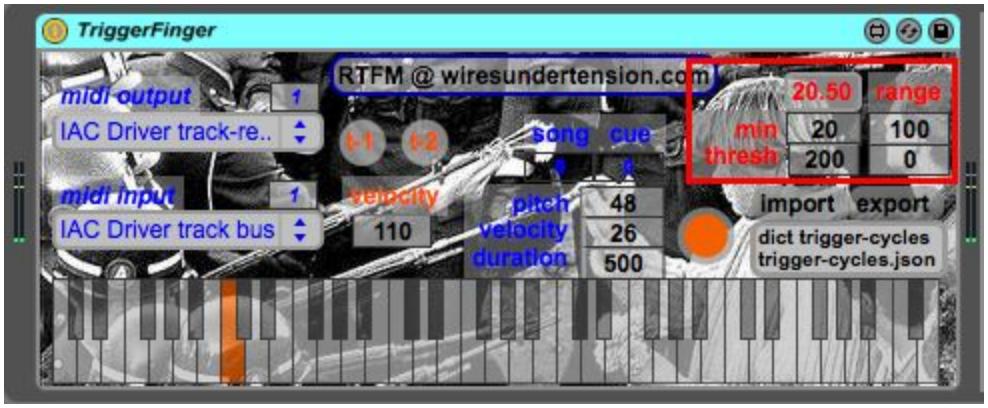
MIDI Routing



Select the desired virtual MIDI bus for playback output from the **midi output dropdown** at the top of the highlighted region. The number box above / right of the dropdown indicates the desired **MIDI channel** you would like *Trigger Finger* to send notes to on this channel.

Trigger Finger receives MIDI messages for various features (see below). The device can be used without these features if desired. To enable these features select a new virtual MIDI bus from the **midi input dropdown**, also indicating the **MIDI channel** in the number box above / right this dropdown. ***The input MIDI channel or virtual MIDI bus should be different from the one used for output to avoid MIDI feedback behavior.***

Trigger Calibration



Audio trigger calibration is set-up using the controls highlighted above.

The number box at top with red text indicates the **current level** of the incoming audio (where 100.0 indicates roughly a gain of 1.0).

The **min** number box indicates the lowest input level value that can output a trigger. This is useful for eliminating noise floors.

The **thresh** number box indicates the minimum number of milliseconds between trigger events. This is useful for eliminating “double triggers” and the like.

The **range** column of number boxes indicates the target range for scaling the audio signal level into the 0-127 MIDI velocity range. The lower number box indicates the bottom of this range - all signal levels <= this value will output velocity 0 (no trigger). The upper number box indicates the top of this range - all signal levels >= this value will output velocity 127. These boxes are useful for controlling how much velocity variation you want in your output. Often ***less sensitivity using a tighter range*** between these numbers is more useful, trading volume sensitivity for more consistent results. Tune appropriately for your needs and style of playing.

When a trigger occurs, the orange trigger button below these controls flashes yellow. You can click this button manually at any time to send a trigger too.

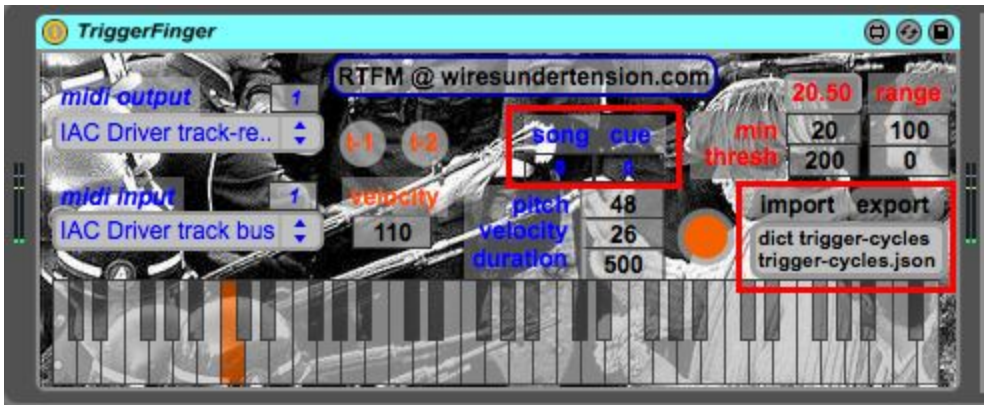
Note Values

The pitch, velocity, and duration of the midi notes output from each trigger are indicated in the highlighted number boxes above. The **velocity** is auto-populated by the audio level as described above or the midi-trigger velocity (see below) with each trigger event. The **pitch** value indicates the pitch of the next note to be triggered. It can be set manually in the number box or by selecting a note from the keyboard. The **duration (**in milliseconds) of each note triggered is set manually using the number box.

MIDI Triggers

In addition to audio triggering, two MIDI trigger buttons are provided that can be parameter mapped to your midi controller. The velocity of these notes is fixed at the value indicated within the number box highlighted above.

Pitch Cycle Scripting



Trigger Finger uniquely lets you provide cycles of pitch values that will be incrementally played with each trigger event. By providing this simple score, you can use percussion to play an additional part, controlling its feel through the timing and velocity of your playing.

Pitch Cycles

Pitch cycle scores are stored in a simple text file on disk as a JSON dictionary. The file “trigger-cycles.json” is loaded into the dictionary object highlighted at right above when the device initializes. **This file should always remain anywhere within the Max search-path (for example, *Applications/Max 6.1/patches*).** Double click on the dictionary object to bring up the editor:

```
{
      "1-1" : [ 48, 50, 52, 53, 55, 57, 59 ],
      "1-3" : [ 54, 69, 66, 54, 69, 66, 54, 52, 71, 64, 52, -1, 69 ]
}
```

This dictionary contains two pitch cycles indicated by the bracketed array of pitch values on each line to the right of the colon. The first plays an ascending C Major scale. The second has a more elaborate melody, including a rest indicated by the value -1 at the second to last note.

Trigger Finger organizes entries in this dictionary using a **"<song>-<cue>"** format for the dictionary keys. Above, each pitch cycle is associated with song 1. The first cycle is also associated with cue 1 within this song, the second with cue 3. This allows you to break up your songs into different sections, each associated with a different pitch cycle.

Trigger events play each note, one at a time, in order, from the activated cycle, repeating from the beginning in a loop. Cycle arrays support any valid MIDI pitch. Additionally, a value of -1 will indicate the trigger should rest, not playing a note. This lets you create output rhythms that deviate from the input rhythm, skipping input events.

Editing Pitch Cycles

Double clicking on the dictionary object opens the editor. The dictionary editor enforces a strict format in these entries to produce valid JSON. All entries in the pitch array must be separated by commas excluding the last entry. Likewise, note the comma following the first array, separating entries in the dictionary. All entries are separated by commas excluding the last entry. If you make a mistake in the format the editor will alert you to the line where the error is detected before letting you close and loading your changes into the dictionary.

The dictionary editor's capabilities are quite limited. For example, there is no undo-redo support. For elaborate changes, consider copying the entire contents into a real text editor, making the required changes and then replacing the entire contents back in the dictionary editor.

In order to preserve your changes for next time, you need to overwrite the trigger-cycles.json text file on disk. This is done by clicking the **export** button highlighted above. **Note that saving your Live set WILL NOT save changes you've made to your trigger-cycle dictionary. You need to export trigger-cycle files manually following any changes.**

The **import** button above will let you read in any valid JSON dictionary file, replacing the contents of the dictionary. This is useful for keeping alternate versions of your pitch-cycles in different files, for example as a backup of previous ideas. Also, if you change the dictionary using the editor, don't like the changes and would like to revert to the saved version, simply re-import the file.

Activating Pitch Cycles

The pitch cycles associated with each <*song*>-<*cue*> entry can be activated using the **song** and **cue** number boxes highlighted above. When the dictionary contains a valid entry for the indicated <*song*>-<*cue*>, the number boxes' background becomes a solid white indicating the cycle is active. Note these number boxes are also MIDI parameterizable in Live. When a cycle is first selected, the first pitch will be cued up, displayed in the **keyboard** and **pitch** number box.

<*song*>-<*cue*> entries can also be selected using MIDI input from the source configured via the routing dropdown. Continuous controller messages received from this source will be interpreted as <*song*>-<*cue*> selections where the controller number indicates the song and the controller value indicates the cue.

Controlling the Pitch Remotely

The pitch **keyboard** that specifies which pitch will be triggered next can also be controlled using MIDI input from the source configured via the routing dropdown. Any note-on event received from this channel will be interpreted as a pitch for this **keyboard**. This feature is automatically disabled when a pitch cycle is activated. This allows you to have pitches for certain parts of the song scripted using pitch cycles where others are specified live using a MIDI controller.

Find this Manual

With an active internet connection, this manual can be accessed anytime by clicking the button highlighted above. A browser will open and direct you to the wiresundertension.com page for this guide.

The Second Amendment is Anachronistic Vaguery

The second amendment is anachronistic and vague. Guns should be licensed and registered.