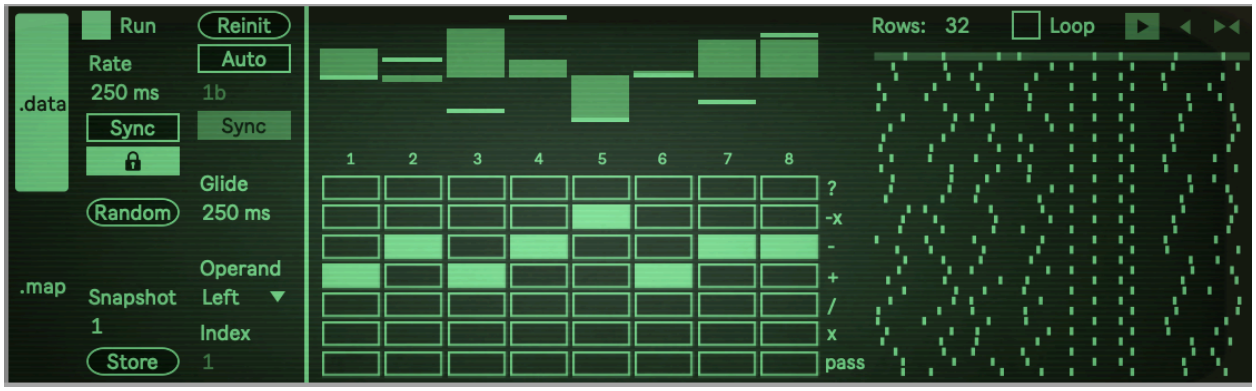
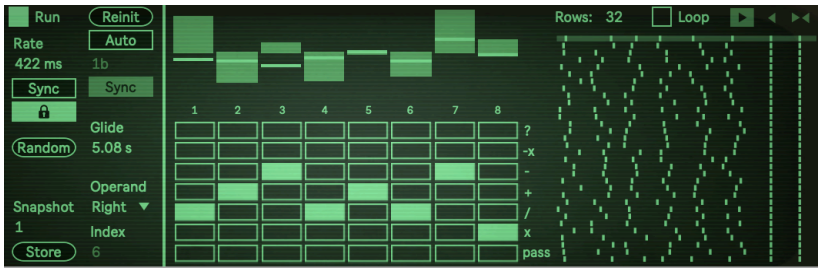
data.mod User Manual



To install:
Unzip the folder and drop the folder called “data.mod” in this exact location in order for presets to load correctly: *ableton/user library/presets/audio effects/max audio effect*
For best results or if you are having issues, make sure you are using the latest version of max/msp. You do not need to have a license if you are using Live suite. Download the newest version here: https://cycling74.com/downloads and once downloaded go to the ableton Preferences > Library and set the newly downloaded version of max to the one ableton should use. (Also you can try to see if it works fine with your bundled version first).

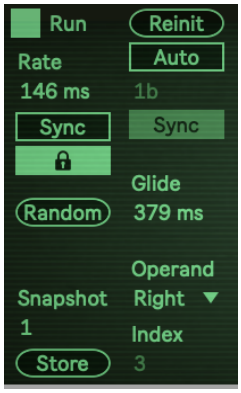
Synopsis
data.mod is a max for live audio effect that is a series of modulators which can be mapped to various parameters in Live. It modulates the parameters via an ever changing array of values in the “data” section. Each frame, these values are changed via a selection of operations. Depending on the initial values and the selected operations, these values can produce various types of oscillations, from simple to complex.

THE “data” SECTION

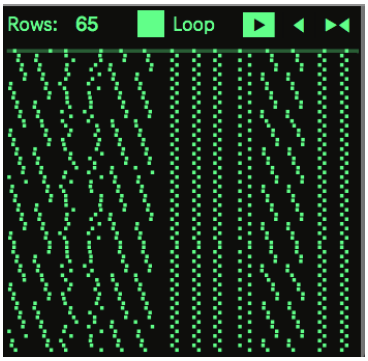


The **multisliders** on top show the modulation values of each parameter in the device. The bar multisliders set the original modulation values (from -1 to 1). These are also called the “seed” values. The brighter green horizontal bars layered over them display the current modulation values. Below each multislider is a label of the parameter that it modulates. “?” gives a random value each frame, “-x” negates the value, “-” is subtraction, “+” is addition, “/” is division, “x” is multiplication, and “pass” keeps the same value. The button **matrix** below the multisliders set what operation the value undergoes each new

frame. Each column of the matrix is associated with the labeled modulation destination, while each row relates to an operation which is listed to the right side of the matrix. For example, in the picture above, the “2” modulation undergoes an addition (+) operation every frame, which means its current value is added to another value to get the next modulation value. The value which it is added to is determined by the Operand parameter, which is described below.

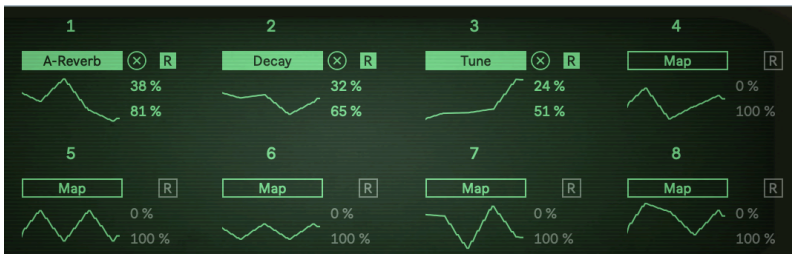


The leftmost side of this section controls the settings of the frame processing of the data. **Run** starts/stoprs the data processing. You can set the **Rate** in milliseconds or clock **synced** intervals. **Reinit** resets the modulation values to the original “seed” values. You can also **Auto** reinit the values either in milliseconds intervals or clock **synced** intervals. **Glide** allows for gliding between new modulation value changes for smooth transitions. You can also **Randomize** all of the parameters in the “data” section and store/recall **Snapshots** of all values in this section. The **Operand** parameter sets which value becomes the right hand operand in the value operation on each frame. “Left” sets the operand to the value left of the current value, “Right” sets the value to the value to the right of the current value, “Op” sets the right hand value to the last value that went through the same operator, “Self” sets the right hand value to itself, “Index” sets the value to a specific value from the list set by the **Index** parameter below. So, following our example described in the paragraph above about the index “2” modulation with the addition (+) operator… if Operand is set to left, then the index “2” modulation value will be added with the index “1” modulation value to get its new value. For Right, it would be added with the index “3” modulation value, for Op it would be added with the “drive” modulation value, as that is the last value to also go through the addition (+) operator in the example picture. For Self it would be added to itself, and for Index it would again be added with the index “3” modulation value as that is the selected Index parameter in the example picture in the above left.



In this section a history of frames of modulation values is visualized based on the **Rows** parameter. If **Loop** is off, this history does nothing functionally other than show the changes of the modulation values. If Loop is on, then these values are looped and repeated instead of new values being calculated. The arrows to the right of Loop set the **direction** of looping (forward, backward, foreback).

THE “map” SECTION



This section is much more straightforward. These are where you map the 8 modulating values to a parameter destination in Live. Simply press the “Map” button for each value and then click a parameter in Live to map it and start modulating. Below each Map button is a monitor of the modulating value. To unmap a value just press the “X” button that appears after mapping. The “R” button allows you to go from Remote modulation mode to “Mod” mode (**Mod mode is ONLY FOR LIVE 12!)** which allows you to modulate parameters relative to the current value instead of taking them over. If in Remote mode, you have a minimum and maximum scaling of the value range from 0% to 100% so that you can choose the parameter value range that is modulated. For Mod mode you instead attenuate the amount of modulation from the set parameter value (keep in mind this can be a negative percentage to inversely modulate). There is also a toggle in Mod mode “+-” to change from the modulation being bipolar or polar.

I hope you enjoy this device! Please email me if you have bugs or other issues:
dillonbastan@gmail.com
More: http://dillonbastan.com