



Manual

Version 1.184  
January 22, 2019

© Chaos Culture, Germany 2019

**ISOTONIK**  
STUDIOS



## Table of Contents

|  |    |
|--|----|
| Introduction .....   | 3  |
| Basics .....   | 3  |
| Max 8 Installation .....   | 4  |
| Help Guide .....   | 4  |
| Specifics .....  | 8  |
| Trigger Signals .....  | 8  |
| Clock Signals .....  | 8  |
| Connecting Modules .....   | 9  |
| Latency Devices & Detection .....  | 10 |
| Visual Feedback .....  | 11 |
| Monitoring .....   | 12 |
| Curves .....   | 12 |
| Automations .....  | 13 |
| Push Controller .....  | 14 |
| Control Voltage .....  | 14 |
| Options .....  | 15 |
| Errors .....   | 15 |
| "old version in use" .....   | 15 |
| "new (free) update available at isotonikstudios.com" .....                     | 15 |
| "current setup can't be saved, too many modules" .....                         | 16 |
| "reading and storing options is disabled, file path couldn't be located" ..... | 16 |
| Failed Initializations .....   | 16 |
| Modules .....  | 17 |
| Inputs / Outputs .....   | 17 |
| Audio In / Out .....   | 17 |
| Dial .....   | 17 |
| MIDI In / Out .....  | 18 |
| Note In .....  | 18 |
| Observer .....   | 18 |
| Remote .....   | 18 |
| Generators .....   | 19 |
| Envelope .....   | 19 |
| Follower .....   | 20 |
| LFO .....  | 20 |
| Noise .....  | 21 |
| Transformers .....   | 21 |
| Function .....   | 21 |
| Spread .....   | 22 |
| Effects .....  | 22 |
| Jitter .....   | 22 |
| Smooth .....   | 22 |
| Timing .....   | 23 |
| Clock .....  | 23 |
| Counter .....  | 23 |
| Delay .....  | 23 |
| Swing .....  | 24 |
| Basics .....   | 24 |
| Math .....   | 24 |
| Movement .....   | 24 |
| Range .....  | 24 |
| Scale .....  | 25 |
| Routing .....  | 25 |
| Mix .....  | 25 |
| Trigger Merge .....  | 25 |



## Introduction

Thank you for purchasing Signal. This is a passion project and it is my personal desire to make it as functional as it could possibly be. A lot of time has been spent on researching, testing and optimizing, but since it is a very complex device there is a chance that you might come across a problem. Please let me know, I'm very interested in fixing it as well as hearing your suggestions and feedback. You can get in contact with me using the Isotonik Studios support system. Simply log into your account and create a ticket. Alternatively you can also reach me directly at [ChaosCulture@aim.com](mailto:ChaosCulture@aim.com), I will always respond as quickly as possible.

There are many (free) updates planned and already in the works, whenever a new version gets released you will receive a mail from Isotonik Studios with your download link.

## Basics

Signal is a node based / modular playground for control signals such as LFOs, envelopes and automations. Most Max for Live control devices have a lot of issues and limit the possibilities by completely taking over the control of a parameter in Live. Signal uses audio signals for all its calculations and lets you route them however you want giving you a lot more options while keeping the highest quality possible along with staying sample accurate.

Additionally it eliminates chain mapping, where you would use an LFO on another one. Since every mapping in Live introduces a delay as well as CPU usage, chain mapping is not useful at all. With Signal you only need one mapping to make everything possible. You can always add another signal and keep expanding the setup. Especially by using multiple instances you can use Live's own routing to mix control signals, record them and play them back in any way using the session view, the Looper device or simply by triggering them using Simplifier/Sampler.

Additionally Signal is the only device on the market that provides a workaround to control parameters in sync with Live by compensating for the delay. You can find out more about this workaround on page 10.

The modules in Signal are versatile, you don't need a lot of them to create something useful. With a lot of additional features that haven't been done before, bypassing mapped parameters for example, Signal is the ultimate control device.

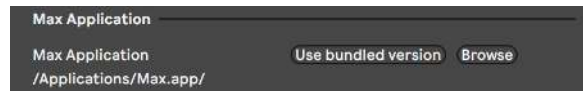


With the flexibility Signal becomes a meta language, you can basically create your own control devices exactly how you like it without actually having to program them in Max. Signal takes away all the work of getting the basics to work and leaves you only with the decision making.

## Max 8 Installation

This device makes use of recently introduced Max 8 features. Unfortunately Live 10 uses an outdated version at this time. The latest beta version already includes the a working Max 8 version and the next public Live release will include it as well. Until then you will have to download Max 8 from [cycling74.com/downloads](http://cycling74.com/downloads) and tell

Live where it is located under Preferences → File / Folder → Max Application.



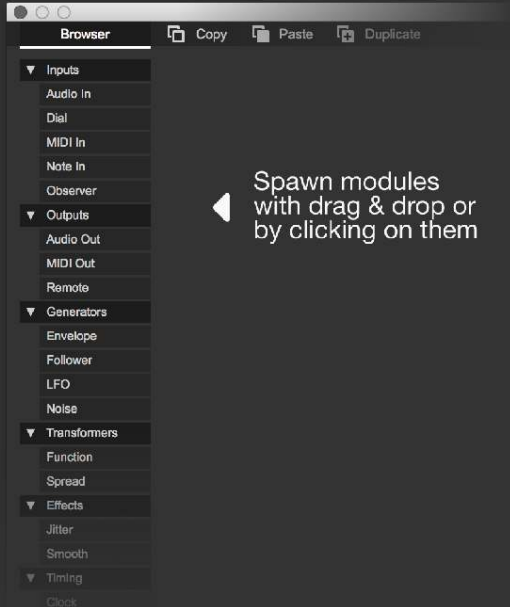
This also required to do for Live 9 users, Signal will not work without Max 8.

## Help Guide

It is heavily suggested to read this guide in the following pages before creating your own presets. It will give you an overview of the most important aspects of Signal.

Further explanations and additional informations about some of its subjects can be found in the next chapters.

You can open this guide within Signal itself by clicking the "Help" button.

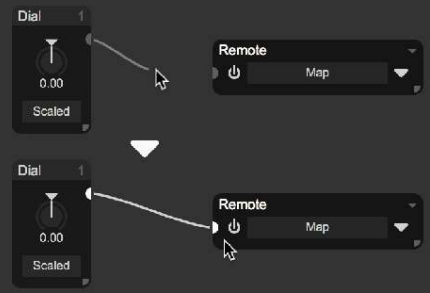


Browser

- Inputs
  - Audio In
  - Dial
  - MIDI In
  - Note In
  - Observer
- Outputs
  - Audio Out
  - MIDI Out
  - Remote
- Generators
  - Envelope
  - Follower
  - LFO
  - Noise
- Transformers
  - Function
  - Spread
- Effects
  - Jitter
  - Smooth
- Timing
  - Clock

Copy Paste Duplicate

Spawn modules with drag & drop or by clicking on them




Connect modules with drag & drop

Inputs are always on the left and outputs on the right

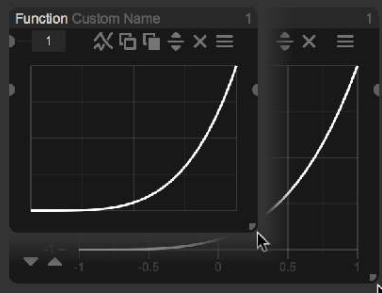
Hold CMD to change existing connections (CTRL on Windows)

Hold ALT to replace existing connections (ALT-click to delete connections)



Click on an input or output to monitor its signal

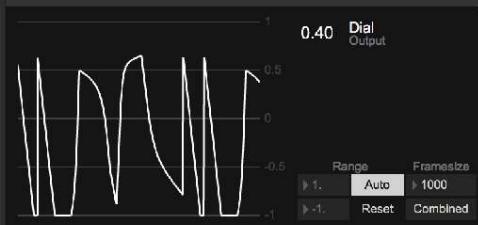
Hold SHIFT to monitor multiple signals



Double-click a module header to give it a custom name

Click and drag on module headers to move them around

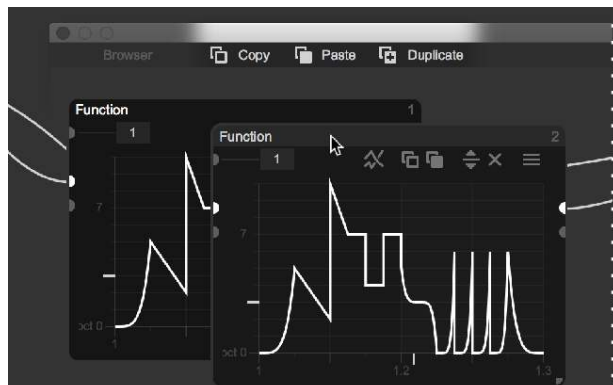
Use the handle in the bottom right to resize modules



0.40 Dial Output

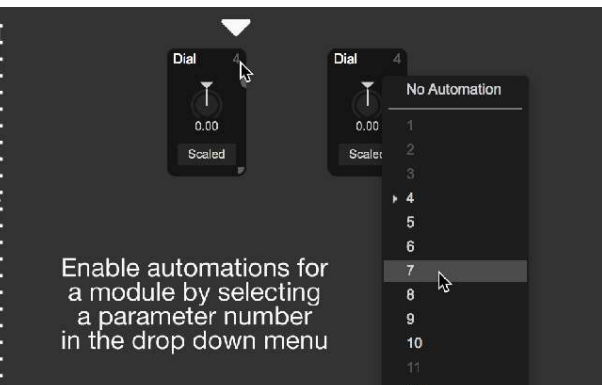
Range: Auto Frame size: 1000

Reset Combined




Copy & paste can also be used to transfer modules to different devices and even different Livesets

Hold ALT while moving modules to duplicate them on the fly

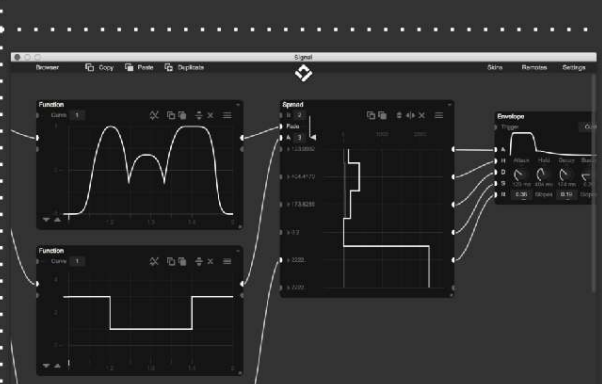


Enable automations for a module by selecting a parameter number in the drop down menu




Press the backspace or delete key to delete selected modules or connections

Drag on the background or hold CMD for multi-selections (CTRL on Windows)



Click and drag within the background while holding CMD to scroll through the project (CTRL on Windows)

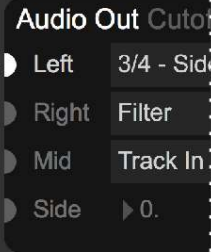
Hold shift additionally to zoom in and out  
Double-click to reset offset or zoom




All modules run in audio rate and every connection is an audio signal

Use audio in and out modules to exchange signals between devices or to send and receive control voltage

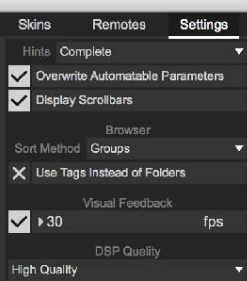
A signal can be any number and is not limited to a range



Activate complete hints in the settings for detailed informations while hovering over interface elements

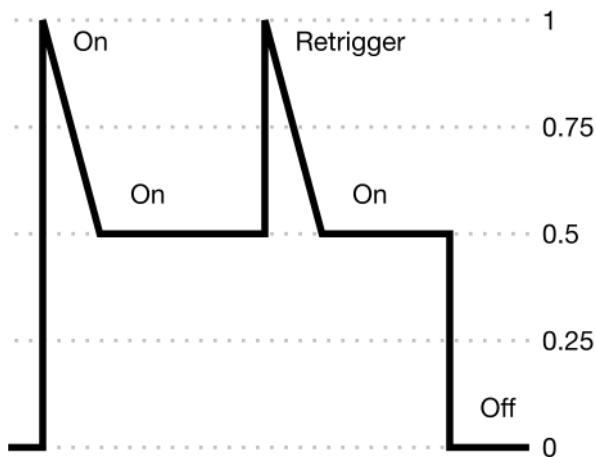


the value of this parameter is scaled to the full range defined below



## Specifics

### Trigger Signals



A trigger signal can be in 3 different states: 0, 0.5 and 1.

If the signal is set below 0.25, modules will recognise it as an OFF signal.

Both 0.5 and 1 will be treated as an ON signal, but if the signal moves from 0.5 to 1 (crossing 0.75), modules will retrigger without the need of an OFF signal beforehand.

This way you can also use an automated Dial module to draw your own trigger signals or use standard 0 to 1 gate signals from other sources such as control voltage gates.

Modules that create trigger signals will only hold the retrigger position (1) for a single sample. The monitoring will not be able to catch and display this spike, because in order to show a large timeframe that is needed to display control signals in a useful manner the monitoring has to average the incoming signals.

If you want to use absolute sample accurate trigger signals, you can use the "Trigger Generator" preset. This includes a sampler device with a specific audio file. The output of this device will be a perfect trigger signal and it is being triggered by MIDI notes and without the use of Max at all. This is important, because MIDI in Max is not sample accurate.

### Clock Signals

A clock signal is just a linear count of quarter notes (beats). A clock signal of 2.5 means 2 quarter notes and 1 eighth note have passed. How long it takes for the clock signal to move up depends on the BPM.





Some modules make use of a clock signal in order to stay synchronized to Live. They will give you the option to input a custom clock signal, but it is never required. Whenever a module needs a clock signal it will create its own internally unless you connect one to it.

By default all clock signals within the modules as well as the Clock module will use Lives transport to generate the clock signal. Transport in Live is unfortunately not latency compensated yet, so the clock signal will be out of sync whenever there is a latency introducing device in front of Signal. This is the same for any device and plugin.

As a workaround, the Clock module allows you to use automation instead of Lives transport, which is latency compensated unless you disable it in Live. This way it is possible to create a 100% accurate clock signal since all automations in Signal are sample accurate.

## Connecting Modules

To connect modules simply click and drag from an input or output and connect it to anything.

Creating endless loops will deactivate affected inputs and outputs until the loop is cleared. The device will not break during this time, however the internal latency compensation will be deactivated. This means other modules will still run and work as usual, but their timings might be off if there are any latency introducing modules involved.

In order to work quickly Signal includes a few shortcuts that might come in handy.

If you drag one end of an existing connection while holding the CMD key (CTRL on Windows) you will change this routing without affecting the other end of it. This also works for multiple connections that are connected to this input or output.

Additionally if some of the connections are selected, Signal will only modify those.

Clicking on a connection while holding the ALT key will delete it. This means you don't have to select it and press the delete (or backspace) key. This also works while creating new connections. Whenever you start a drag it will delete all previous connections of this input or output if you hold down the ALT key while doing so. The same thing happens when you stop the drag for whatever input or output you want this new connection to go to. The ALT key is used both times when you press down and when you let go of the mouse button.

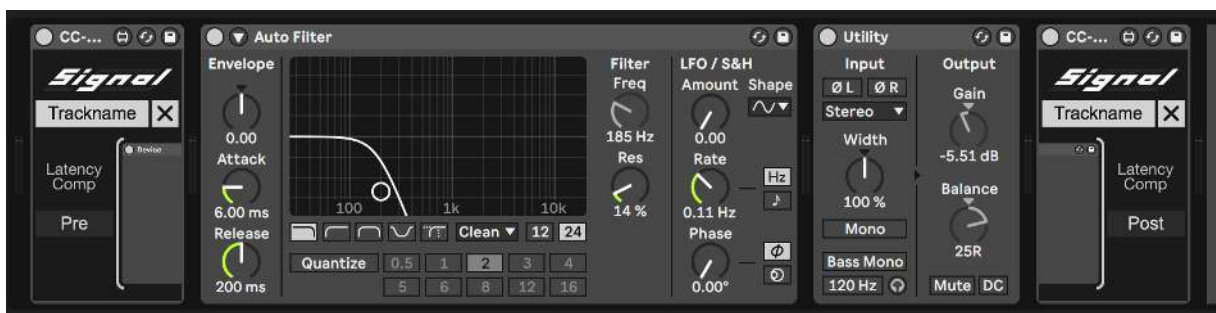


The SHIFT key allows you to perform the same task for all selected modules of the same type. All other shortcuts can be combined with holding the SHIFT key.

## Latency Devices & Detection

The latency devices provide a workaround that allow you to control parameters in Live accurately.

If you control a parameter in Live with the Remote module, Live will add a small delay to it depending on latencies and your buffer size. This is the same for any Max for Live device (even the default control devices by Ableton). If this delay is not a big issue, you can simply use Signal without these Latency Devices, it will work just as well.

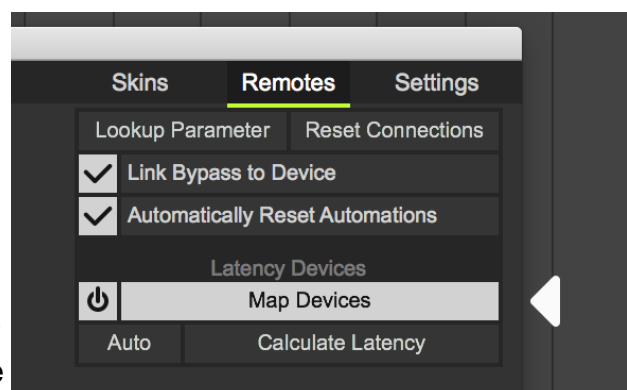


But if you would like to compensate for Lives delay when controlling parameters, you need to place one Latency Device before and after the devices / plugins that you're controlling with Signal.

Set the Latency Device to "Pre" or "Post" to let it know where it is within your signal flow as shown in the picture above.

Open the "Remotes" settings at the top within Signal and activate the "Map Devices" button. After that you'll be able to press the "Map" button within the latency devices to connect them.

Once they are connected, they will be talking to each other and compensate for Lives delay.



In order to compensate for it correctly, Signal has to calculate the latency of your project, which you can do by pressing the "Calculate Latency" button below.



Whenever the latency changes of devices and plugins in front of Signal, you can manually press this button or simply whenever you notice the timings being off.

Pressing this button while holding CMD (CTRL on Windows) will cause all instances of Signal in your Liveset to perform this action simultaneously.

The latency is also calculated once on load, this means most of the times you don't need to worry about it at all. However if the latency of the Signal device changes constantly, by inserting or changing plugins before Signal for example, you can also activate the "Auto" button to keep calculating the latency forever. This uses quite a bit of CPU, so it is advised not to use this feature unless it is absolutely needed.

You need to be aware that this is a workaround. Every device and plugin that is enclosed within the Latency Devices will be offset in time and normal automations of them will now be out of sync.

## Visual Feedback

Some modules will display values and states based on their audio inputs.

This is used to make Signal more intuitive and helps a lot to visualize what is actually happening. But Max can slow down if you have too many interface elements constantly changing at the same time.

To combat this, you can manually lower the framerate of visual feedback in order to leave more room for interactions and other Max applications.

Additionally you can completely turn off visual feedback, which will change the appearance of some modules by hiding or showing certain controls to reserve full functionality.

Without visual feedback, the interface of Signal will be the most efficient and less demanding on your Computer. The actual signal processing is not affected by the visual feedback however.

If you notice the interface of Signal as well as any other Max for Live device in your Liveset getting slower, lower the framerate or deactivate it.

Obviously, whenever the window of Signal is closed, visual feedback is bypassed automatically. Additionally monitoring signals is excluded from the visual feedback settings. You can still monitor signals without visual feedback, you won't ever lose functionality.



## Monitoring

Signals are not limited to a specific range, the values can jump to any number at any time.

This makes a lot of things possible (making use of different units, offsetting without clipping, rounding to integers, clock signals and much more), but it's easy to lose track of ranges.

Monitoring signals is essential to understand how modules work and to figure out what you're actually doing.

By clicking on an input or output of any module, you enable monitoring on it and a small window in the bottom left will appear displaying its value along with a history graph.

Clicking on an input or output while holding shift lets you monitor multiple signals at the same time, which is very useful to figure out the relation of signals to each other.

Since every module has the ability to introduce latency to allow certain features, the Monitoring will ensure that what you're seeing in the graph is always in sync. Any time differences shown in the Monitoring are actual real time differences and not caused by the latencies.

## Curves

Some Modules include a curve editor and even though some features might be disabled for some, they all work in the same way.

You can add and delete points with a double click or a CTRL (Mac)

click, which might work as a right click depending on your mouse and OS.

Holding the CMD key (CTRL on Windows) will snap points to the grid while moving or creating points.

Drag between two points while holding the ALT key to alter its curve. Double clicking will reset it.

Moving points while holding the ALT key will duplicate them first.





You can draw steps on the grid if you hold the SHIFT and ALT keys simultaneously.

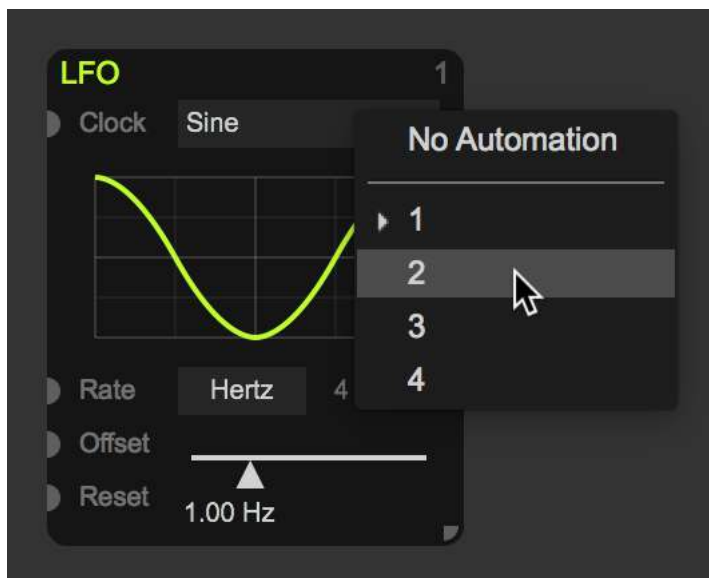
You can select points by clicking on them or by dragging in the background. Holding SHIFT while doing so will not reset previous selections to allow selecting multiple points easily.

Offset and zoom can be altered with the axes on the sides. Holding shift while doing so will lock to either zoom or offset depending on the direction of your cursor drag.

Alternatively you can hold the CMD key (CTRL on Windows) while dragging on the curve background to offset the canvas and zoom in and out while holding the SHIFT key as well. This works exactly in the same way as the overall background of Signal.

## Automations

Some modules allow you to use automations and modulations in Live. Unfortunately because of the nature of Live parameters the amount of automatable parameters is limited.



To enable automations for a module you have to open the drop down menu for it in the top right. There you can assign any unused parameter number to the module or deactivate automations to make them available for other modules.

Most modules will spawn with an automation number already assigned. A triangle symbol is shown if no

automations are enabled and the module doesn't have any automatable parameters if nothing at all is shown in the top right of the module.

If you open the automation menu in Live for Signal you will see a big list of all possible parameters. Every parameter name is labeled with the module type and automation index number. The drop down menu simply lets you choose which parameters to use.



Every automatable parameter can also be controlled by a module input. This means a module that is not automated is not limited in any way. The only exception is the Clock module, because it uses automations for a clock signal as a workaround for the lack of transport latency compensation in Live. Using a signal defeats the point of it.

## Push Controller

The Push controller will show all automatable parameters in use using multiple banks if necessary. Click on the device name on the controller to access all banks.

Parameters are sorted into their module types and parameter automation numbers. Only usable parameters are shown, for example using the LFO module with intervals will hide the frequency slider. Also the automatable parameter within the Clock module is always hidden, because the purpose of it being automatable is to function as a workaround.

## Control Voltage

In order to use Signal with hardware synths and modular systems, you can simply output and input audio signals from it.

All signals and modules in Signal are being processed in audio rate (your samplerate).

Simply use the Audio In and Out modules to receive and send signals out of the device and back into it when needed. Sidechain inputs and outputs as well as direct routing capabilities were introduced in Live 10.

To avoid introducing latency, don't make use of negative delays and latency devices for controlling parameters in Live.

You can use the Scale module to change the range of a signal before sending it to your hardware in order to fix any range issues depending on your setup and interface.

Received audio from your interface can also be used in Signal to do certain things. Every module in Signal that requires a trigger signal will compensate for noise. Your trigger signal (gate) doesn't have to be perfectly clean.



## Options

The options/settings are available in the top right corner of the Signal window.

All open instances of Signal will be updated automatically, they all share the same options.

In order to recall them in the future, Signal will store them within a small file on your computer called "CC Signal Options.txt". This file is located in the same folder as your installed Live packs. This ensures that your options will be recalled even after updating Signal.

Please do not alter this file in any way, it can create problems within the device.

You can delete this file to reset all options to their default values, Signal will automatically create a new one whenever it can't be found.

## Errors

In some situations Signal might display following messages in the top right of the window.

### **"old version in use"**

Signal is updated regularly and because to the nature of Max for Live devices and Live parameters, some updates can't be backwards compatible. Whenever such an update gets released, the Live pack will include an old version of Signal that will be used by old projects of yours.

This message lets you know that this version of Signal you're using is outdated and does not include new features and changes.

You can manually replace it with the new version with Lives file manager. It might work depending on the setup of your Signal preset. Simply drag the new version on top of the old Signal device header in Live. You can find all versions in the Live pack under the "Versions" folder. The last folder (with the highest date number) will include the latest version.

### **"new (free) update available at isotonikstudios.com"**

Signal will display this message once a new update is available for you to download. You can get it at [isotonikstudios.com/my-account/downloads](https://isotonikstudios.com/my-account/downloads) if you log into your account.



### **"current setup can't be saved, too many modules"**

This message lets you know that you've exceeded the amount of data Signal can store and recall. Everything you do from this point on will not be stored in Live and next time you open this Liveset it won't recall all the changes you did from this point on.

It takes a lot of data to reach this limit and in most situations it makes more sense to use multiple instances of Signal anyways.

If you still reached this limit, you'll have to delete modules or their content (especially points in curve editors) to reduce the amount of data that has to be stored. As soon as it doesn't hit the limit anymore, Signal will hide this message and continue to store all data as expected.

### **"reading and storing options is disabled, file path couldn't be located"**

This message will be shown whenever Signal couldn't find the location of the options file and therefore can't import or store any of them. The device will work just fine apart from that. You can find more informations about the options file on page 15.

## **Failed Initializations**

Signal might not be able to fully load under some circumstances. Instead of showing the "Open" and "Help" button, Signal will display an error message explaining the problem.

Keep in mind Signal requires Max 8.0.1 or above. At the time of release, Live 10 has an early version of Max 8 build in, which doesn't support Signal yet. Make sure to download the latest Max version from [cycling74.com/downloads](https://cycling74.com/downloads) and connect it with Live (Preferences → File Folder → Max Application). You don't need a full Max license to do this and to use Signal.

If you have any questions or unexpected errors, you are always welcome to get in contact by logging into your account at [IsotonikStudios.com](https://isotonikstudios.com) and use the support system.





## Modules

### Inputs / Outputs

#### Audio In / Out

Receive and send audio using the device inputs and outputs.

Sidechain inputs and outputs have been introduced in Live 10 and are therefore not available in Live 9.

Signal allows as many inputs and outputs as technically possible (64 channels for both ways).

In Live inputs and outputs are always stereo pairs of two channels.

Routing (introduced in Live 10) can be used additionally, but is not required. Without routing the audio is still being received or send to the correct channels just like any other plugin.

Before routing you need to select the input or output channel of Signal with the first dropdown menu. The routing menus below are tied to this channel, not to the module itself. This means you can have two modules using the same sidechain input and the routing will always be the same for both, because the routing is done within Live and has nothing to do with Signal.

The additional bypass button can be used to mute the incoming or outgoing signal. You can mute control signals before routing them to make sure they never reach your speakers.

Using negative values in the timing options below will introduce latency, make sure Live's latency compensation is active when doing so.

These modules are disabled and won't show up in the browser if you're using the MIDI version of Signal since Max MIDI effects do not support audio inputs and outputs.

#### Dial

This module simply outputs the dial value, which can be automated.

Without scaling, the range of the dial uses the displayed range of -1 to 1.



Scaling allows you to quickly set up a custom range for the output.

The split feature will separate the range into two equal segments with a fixed middle point.

## **MIDI In / Out**

Receives and sends MIDI data.

When sending MIDI, the incoming audio will be converted into MIDI signals. Since MIDI is not as fast as Audio, you will always lose signal resolution.

The additional update rate control lets you alter the resolution if needed.

MIDI Out is only available in the MIDI version of Signal and the audio version does not support MIDI Inputs and outputs at all.

## **Note In**

Receives MIDI notes and outputs a trigger, pitch and velocity signal.

You can filter out a specific note range and select a specific MIDI control change number to be used for the sustain pedal.

Pitch and velocity is scaled to the default -1 to 1 value, but it's possible to output the unaltered values (0 to 127) instead.

Max audio effects do not support MIDI inputs and therefore this module is deactivated in the audio version of Signal.

## **Observer**

This module outputs the value of any parameter in Live.

It's impossible to get sample accurate and latency compensated values, be cautious about using it for time sensitive tasks.

You can scale the output to a -1 to 1 range or use the parameters own range without altering it in any way.

Using negative values in the timing options below will introduce latency, make sure Lives latency compensation is active when doing so.

## **Remote**

Controls parameters in Live with the incoming signal.



You can customize the range of the parameter along with a reference input points. By default -1 will set the parameter to its lowest point (0%) and 1 will set it to its highest point (100%).

The assign buttons will read the current parameter value and use it as either the highest or lowest points of the range. This is especially useful if you want to set a specific value by entering in Live instead of Signal (bypass the connection first to be able to alter the parameter in Live).

Range mapping (CMD / CTRL click) will not immediately take over the control once you selected a parameter in Live. Instead it will constantly check the parameter value and remember its lowest and highest values until you confirm the mapping. This allows you to quickly map a parameter while at the same time set the range for it.

The update rate control at the bottom can be used to increase or decrease the resolution. 0 ms will result in absolute sample accurate modulations, but require more CPU. Keep in mind not every parameter is capable of using sample accurate informations, so it is often pointless to set it to 0 ms.

Using negative values in the timing options below will introduce latency, make sure Lives latency compensation is active when doing so.

The bypass button can be automated if you activate it using the automation drop down menu. Be aware that as of yet Live does not support this during rendering.

At the top navigation bar of Signal you will find a "Remotes" tab with a few global settings for all Remote modules. The lookup button will bypass all mapped parameters and as soon as you select one of them in Live, Signal will display the correct module that is mapped to it. The reset button will make sure all Remote modules are actually connected to the parameters. This might happen sometimes, especially if you move the Signal device in your Liveset. Both the lookup and reset buttons can be used while holding the CMD key (CTRL on Windows) to perform this action for all instances of Signal in your Liveset at the same time.

## Generators

### Envelope

Generates an AHDSR curve controlled by a trigger signal input with an output range of 0 to 1.



With legato mode enabled the attack phase will start from the current output value instead of resetting at 0. This ensures smooth transitions if a new attack phase is started before the release phase has ended.

Without the sustain mode the envelope will not wait for a trigger off signal before starting the release phase. The attack and decay phase will be played through completely though.

With the custom mode this module will output linear values describing the current position of the envelope. This allows you to use a Function module with custom curves instead.

The attack phase uses the range 0 to 1. Decay uses 1 to 2 and the release phase goes from 2 to 3. Depending on the hold time and sustain mode, the output signal will pause between the phases (at 1 and 2) if needed.

## **Follower**

This module expects an audio signal input and generates a signal that follows the incoming amplitudes.

The range starts at 0, but how far the values goes up depends on the incoming amplitudes. Remember that you can always use monitoring to figure out the range of any signal.

With the rise, fall and average timeframe values you can smooth the signal however you like.

## **LFO**

The output of this module is within the range of -1 to 1.

The curve is customizable and the presets in the list above can be used as a starting point.

Copy and pasting works across modules, Signal devices and Livesets.

Changing the interval rate might result in jumps, because instead of calculating the frequency for the LFO, it uses a clock signal and divides it to the desired speed. This way the LFO is always locked to the clock to give you more control and predictable results along with the phase offset parameter.



A clock input is not required, the module will generate its own clock signal if nothing is connected to it.

The reset input expects a trigger signal input and will set a secondary phase offset along with the custom phase offset. It will restart the phase at the position of the default phase offset. Whenever you disconnect a signal from this reset input, this hidden secondary phase offset will be reset in order to not affect the output anymore.

Using the custom output mode, the module will send out a linear value from 0 to 1 that represents the LFO position. You can then use a Function module and its functionality for the LFO curve instead.

The additional trigger output within the custom output mode sends a trigger ON value everytime the phase starts over and when the phase crosses the middle point (0.5) the trigger signal will be set to OFF again.

## Noise

This module outputs random values within the range specified below (including the low and high values themselves).

When using a trigger input, a new random value is only generated whenever a trigger ON value is received.

## Transformers

### Function

This module translates an incoming signal using a customizable curve.

The x axis (horizontal) represents the input and the y axis (vertical) the output.

There is no fixed range, you can zoom and offset the curve axis and use any value.

Within the options you can change the units for input and output if needed.

With the curve number on top you can switch between different curves and you can change the amount of available curves in the options. There is no limit on the amount of curves, but if you use automations the curve number parameter itself will always stay within the default range of 16. Using a signal input for the curve you can still access every curve number though.



The lossless quality is efficient, but especially if you move across a lot of points in a short time it might make sense to use lower quality settings. The lossless quality will however always be accurate, while the other options will use a pre-rendered curve and therefore interpolate between rendered values.

Copy and pasting works across modules, Signal devices and Livesets.

You can apply this curve to multiple individual signals. The additional inputs and outputs will be shown whenever the previous ones are being used. This way you don't have to duplicate this module just to apply the same transformation on multiple signals.

## **Spread**

This module will take multiple inputs and adds a value to it specified with the number directly next to the input.

You can quickly generate these values by using a curve on the right.

Additionally you have multiple curves available and you can crossfade between them using the controls in the top left.

Just like the Function module, the amount of curves is unlimited and can be changed in the options.

You don't have to connect anything to the inputs and just use the outputs as signal generators.

Additionally you could also use a single signal for all inputs to spread it.

Copy and pasting works across modules, Signal devices and Livesets.

## **Effects**

### **Jitter**

This module creates a jitter signal with a customizable frequency, which you can then mix together with other signals.

The output range is -1 to 1 if the amount is 100% active (set to 1).

### **Smooth**

This module takes a signal and smoothes it out to avoid quick movements.



The amount simply scales the ramp up and down values. This way the amount is automatable and the actual values are still endless, because automatable parameters always require a specific range.

## Timing

### Clock

This module generates a clock signal that loops after a set amount of bars.

By default the module will use Lives signature to determine the length of a single bar, but you can also use a custom signature instead.

The offset works in quarter notes, since a clock signal is just a quarter note count.

You can enable automations for this module with the automation drop down in the top right. This will show a parameter you then have to automate (or modulate) to go up in a straight line for the same amount of bars as the bars duration value at the top. This is a workaround for the lack of transport latency compensation in Live. It allows you to create a clock signal that is latency compensated.

Instead of automations you can also use modulations, which are locked to a clip in Live. This way you can loop this clip or duplicate it, which is a lot easier to manage than automations.

### Counter

Whenever this module receives a trigger ON value it will count the Milliseconds, samples or beats (quarter notes / clock signal) that have passed.

The scaled output will always use a range of 0 to 1, but it requires a stop point.

Connected to the Function module you could use this to "play back" a curve.

### Delay

This module can delay a signal with positive values as well as negative, which will add latency.

By default the dial will be scaled to the range defined below. Set to 0 will make no difference to the signal, 1 will delay the signal with the max value and -1 will use a negative delay defined with the min value.

Additionally you can use a raw dial to enter and automate specific delay values.



When using a negative delay with the unit set to intervals, you have to set a BPM value. This will let the module know what the lowest possible BPM of your song will be in order to not introduce too much latency for no reason. This BPM value is not actually being used for calculating delay times however, it will always use Lives actual BPM.

## Swing

With this module you can apply swing onto any signal.

The clock input is not required, the module will use its own internal clock whenever no external clock is connected.

The curve describes the type of swing and you can customize it to your liking or to fit with other swinging elements in your song.

You can also apply swing onto a clock signal itself and get a swinging clock that can then be used for other modules such as the LFO.

Copy and pasting the curve works across modules, Signal devices and Livesets.

## Basics

### Math

You can perform basic math operations by selecting a method in the drop down.

The values below are being used unless a signal is connected to its inputs on the left.

### Movement

This module analyses an incoming signal and reports its movement.

It uses an average of a customizable time span to avoid constant jumping caused by signals with a low resolution and to give similar results on different samplerates while using the milliseconds unit.

### Range

The incoming signal will be kept within a customizable range.

There are different methods of keeping the signal within the range, warp could be used to create a clock divider.





## Scale

This module allows you to scale a signal by using 4 reference points. If the module receives the value specified in the top left, it will result in the value shown in the top right. The same goes for the bottom two values.

The signal will not be limited within the specified ranges however.

If the learn button is activated, the module will analyse the incoming signal and determine the input low and high values on the left side. This way you don't have to figure out yourself what range a signal is using. The learn mode will automatically deactivate itself after a while if you don't deactivate it yourself.

It's good to know that the values are only being changed once the learn mode has been turned off and Lives undo history will only be altered at this point.

## Routing

### Mix

This module simply mixes multiple signals together.

Since you can connect multiple signals to a single input, it only makes sense to use this module if you want to alter the amount of each signal, which is what the values next to the inputs are for.

### Trigger Merge

This simple module will analyse multiple trigger signals and create a single.

Mixing trigger signals together like any other signal without this module can also work in most cases, but you will lose the retrigger functionality.