

# TINKER

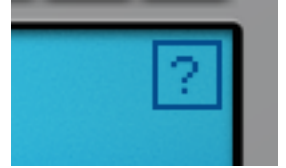
## Operation Manual

Introduction	2
Front	3
Entering a Formula	3
Variables	4
Constants	4
Operators	4
Functions	4
Formula Parsing and Evaluation	6
Inputs Display	7
Description Display	7
Rate Settings	7
Help Menu	8
Patch Browser	8
Knob (K) and Switch (S)	9
Back	10
Variable Inputs (A, B, C, D, W, X, Y, Z)	10
Description Label	10
Result Output	11
Tips and Tricks	12
MIDI / Remote Implementation	14

# Introduction

Tinker CV Math Assistant is a Rack Extension that lets you enter mathematical formulas to manipulate CV signals. This lets you accomplish a wide variety of CV-related tasks with one compact rack device.

Read on to learn how it works. And remember, you can access Tinker's integrated help menu at any time by clicking the question mark icon in the upper right corner of its display.

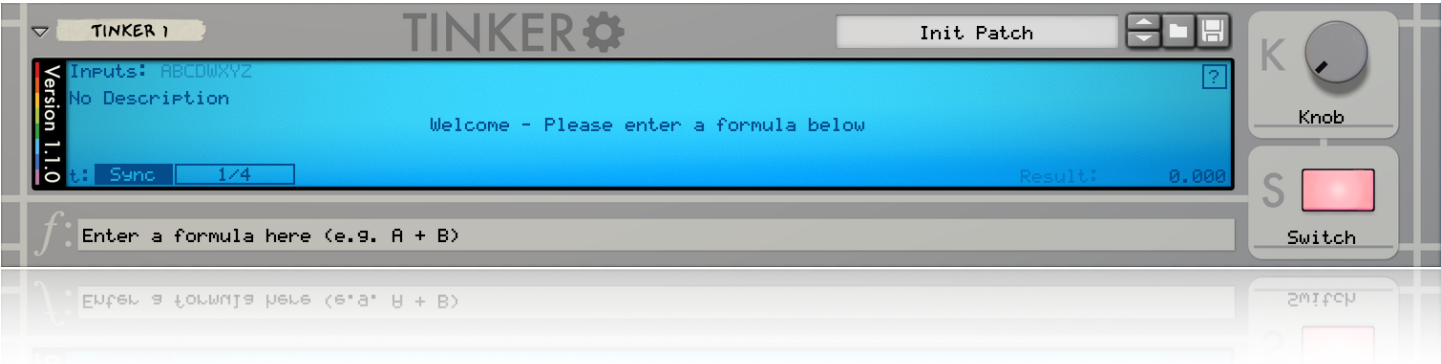


**Help Icon**

For contact information, visit <https://re.pepinhazan.com/Contact>

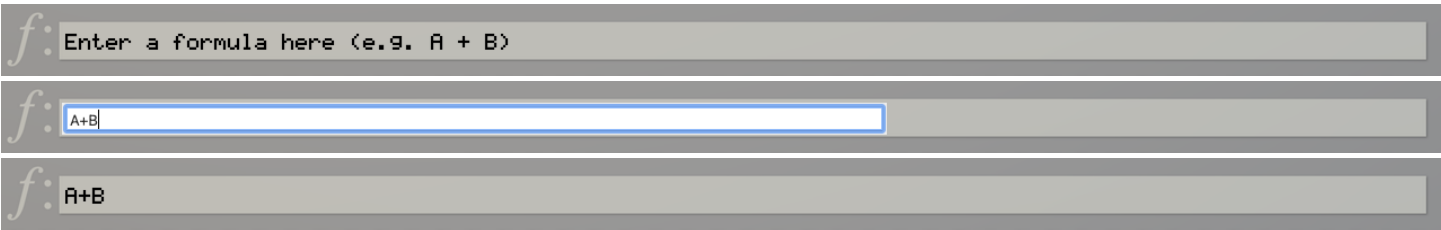
Happy formulating!  
Pepin

# Front



## Entering a Formula

The majority of your time using Tinker will likely be spent entering and editing formulas. To enter a formula, double click the text entry area beside the “f” at the bottom of the device, type in a formula, and press enter/return. On Tinker’s default patch, you will initially see the prompt “Enter a formula here (e.g. A + B).”



### Entering the formula A+B

“A + B” is one example of a formula. In this case, it adds together the values of CV inputs A and B, outputting the result. This formula contains two variables (A and B) as well as an operator (+). In general, formulas can contain numbers, variables, constants, operators, functions, and parentheses.

Numbers	Literal numbers such as “1.58”
Variables	Named changing values, such as “A,” which represents the first CV input on the back of the device
Constants	Named constant values, such as “pi”
Operators	Binary arithmetic operators such as “+” (addition) as well as unary negation
Functions	Named procedures that take 0 or more arguments and output a result, such as “sin(A)”
Parentheses	“(” and “)” can be used to change order of evaluation

### Function Components

All supported variables, constants, operators, and functions are listed below. A condensed version of this list can also be accessed through Tinker’s integrated help menu.

## Variables

A	CV Input A (0 if disconnected)
B	CV Input B (0 if disconnected)
C	CV Input C (0 if disconnected)
D	CV Input D (0 if disconnected)
W	CV Input W (0 if disconnected)
X	CV Input X (0 if disconnected)
Y	CV Input Y (0 if disconnected)
Z	CV Input Z (0 if disconnected)
K	Current setting of knob labeled “K” on the right side of Tinker (0.0 left to 1.0 right)
S	Current setting of switch labeled “S” on the right side of Tinker (0.0 off or 1.0 on)
t	Position within synced or free cycle configured using the rate controls on Tinker’s display (value in [0.0, 1.0) )
qpm	Current song tempo in quarter notes per minute

## Constants

pi	3.14159...
e	2.71828...

## Operators

– (unary)	Negation (e.g. -A)	Precedence 3 (highest)
* (binary)	Multiplication (e.g. A*B)	Precedence 2
/ (binary)	Division (e.g. A/B)	Precedence 2
% (binary)	Remainder (e.g. A%B)	Precedence 2
+ (binary)	Addition (e.g. A+B)	Precedence 1 (lowest)
– (binary)	Subtraction (e.g. A-B)	Precedence 1 (lowest)

**Note: All binary operators have left-to-right associativity**

## Functions

abs(A)	Returns the absolute value of A
toUni(A)	Converts a standard bipolar signal [-1,1] to a unipolar signal [0,1]

<code>toBi(A)</code>	Converts a standard unipolar signal [0,1] to a bipolar signal [-1,1]
<code>clamp(A,B,C)</code>	Clamps A between B and C. In other words: If A is greater than the maximum of B and C, that maximum is returned. If A is less than the minimum of B and C, that minimum is returned. Otherwise, A is returned unchanged.
<code>trunc(A)</code>	Returns the integral part of A
<code>round(A)</code>	Rounds A to the nearest integer
<code>sin(A)</code>	Sine function
<code>cos(A)</code>	Cosine function
<code>pow(A,B)</code>	Returns A to the power B
<code>ln(A)</code>	Natural logarithm
<code>min(P1,P2,...)</code>	Returns minimum of 1 or more arguments
<code>max(P1,P2,...)</code>	Returns maximum of 1 or more arguments
<code>select(A,P1,P2,...)</code>	<p>Uses the value of A to select from one or more arguments which are evenly distributed across the range [0,1]. If A is below 0, the first of those arguments is returned. If A is above 1, the last of those arguments is returned.</p> <p>Example: <code>select(A, 3, 7, 5, 11)</code></p> <ul style="list-style-type: none"> <li>• When A is between 0 and .25, the function will return 3.</li> <li>• When A is between .25 and .5, the function will return 7.</li> <li>• When A is between .5 and .75, the function will return 5.</li> <li>• When A is between .75 and 1.0, the function will return 11.</li> <li>• When A is less than 0, the function will return 3.</li> <li>• When A is greater than 1, the function will return 11.</li> </ul>
<code>lerp(A,P1,P2,...)</code>	<p>Uses the value of A to interpolate one or more arguments which are evenly distributed across the range [0,1]. If A is below 0, the first of those arguments is returned. If A is above 1, the last of those arguments is returned.</p> <p>Example: <code>lerp(A, 3, 7, 5, 11)</code>:</p> <ul style="list-style-type: none"> <li>• When A is between 0 and <math>0.\bar{3}</math> (<math>\frac{1}{3}</math>), the function will interpolate from 3 to 7.</li> <li>• When A is between <math>0.\bar{3}</math> (<math>\frac{1}{3}</math>) and <math>0.\bar{6}</math> (<math>\frac{2}{3}</math>), the function will interpolate from 7 to 5.</li> <li>• When A is between <math>0.\bar{6}</math> (<math>\frac{2}{3}</math>) and 1.0, the function will interpolate from 5 to 11.</li> <li>• When A is less than 0, the function will return 3.</li> <li>• When A is greater than 1, the function will return 11.</li> </ul>

**Note: Arguments A, B, C, etc. used in the functions above are placeholders. They are unrelated to the similarly named input variables.**

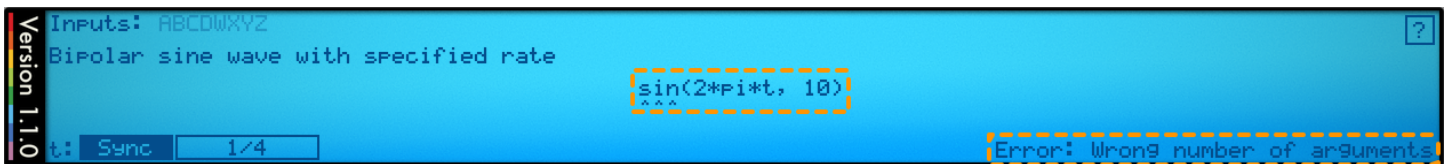
## Formula Parsing and Evaluation

After you enter a formula, it appears in the formula area at the center of Tinker's display.



**Formula display**

If you've entered an invalid formula, you will see an error message in the lower right of the display. The location of the error will also be underlined.



**Invalid formula display**

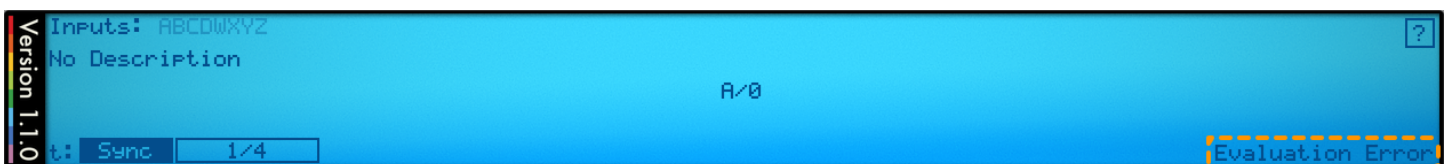
If you've entered a valid formula, then the formula will repeatedly be evaluated, and the result will display in the lower right of the display. The word "Result" will be lighter colored if the output CV is currently disconnected. If the result of your calculation produces a finite value outside of the valid CV range (-10,000 to 10,000), then you will see "Result (clamped)" and the output will be clamped to that range.

Result: 123.000    Result: 123.000    Result (clamped): 10000.000

**Left to right: Disconnected result, Connected result, Clamped result**

Note that only the final output value is clamped. Intermediate results of calculations may exceed the valid CV range as long as they remain finite. Producing a non-finite value during any step of evaluation will cause an evaluation error.

An evaluation error occurs if a structurally valid formula nevertheless produces an error when it is evaluated. Some examples are dividing by zero or raising a negative number to a fractional power. In those cases, you will see "Evaluation Error" in the lower right of the display, and the output will be 0.

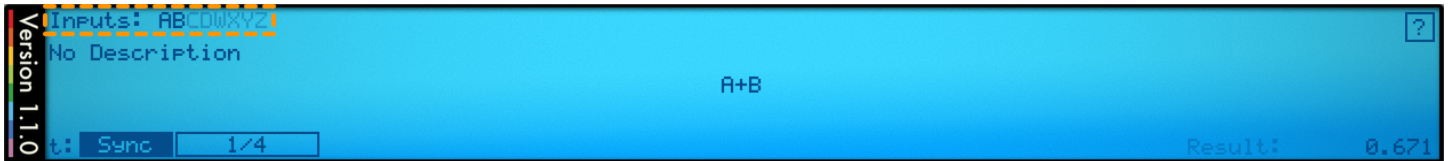


**Evaluation Error caused by dividing by zero**

That covers the basics of writing formulas. Now, let's cover the rest of Tinker's front panel features.

## Inputs Display

At the upper left corner of Tinker's display, you can see a list of all the CV input variables that are available to you. Disconnected inputs will appear fainter than connected inputs.



Display with inputs A and B connected

## Description Display

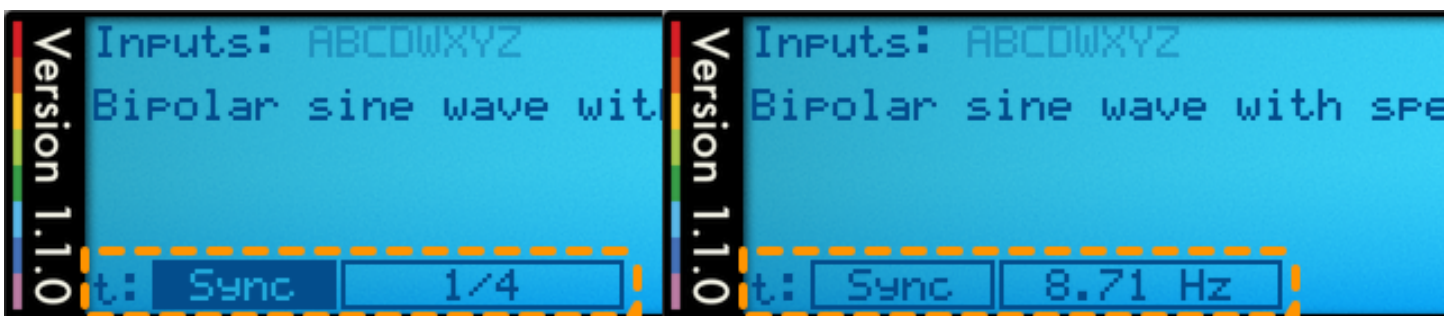
Below the inputs display is an optional description of the current patch. This can be edited on the back of the device. It's useful for explaining how your formula works, how each input variable is used, and so on.



Patch description describing a formula

## Rate Settings

At the lower left of Tinker's display are the rate settings, which are used to control how quickly the variable "t" cycles from 0 to 1. Click the Sync button to switch between free and tempo-synced modes. Then click and drag the box to the right to configure the tempo-synced or free rate. The variable "t" will increase linearly from 0 to 1 at the selected rate, effectively producing a unipolar sawtooth wave.



Left: Tempo-Synced Rate, Right: Free Rate

The following tempo-synced rates are supported (T = triplet, D = dotted):

8/1	6/1	4/1	3/1	2/1	1/1	1/2D	1/1T	1/2
1/4D	1/2T	1/4	1/8D	1/4T	1/8	1/16D	1/8T	1/16
1/32D	1/16T	1/32	1/64D	1/32T	1/64			

Free rate ranges from 0.01 Hz to 99.99 Hz.

## Help Menu

Click the question mark icon in the upper right corner of Tinker's display to enter the help menu. The help menu lists all variables, constants, operators and functions that can appear in a formula. You can click and drag the help text horizontally to scroll, or you can click the buttons at the top to skip to a specific section. Click the back button in the upper right to exit the help menu.



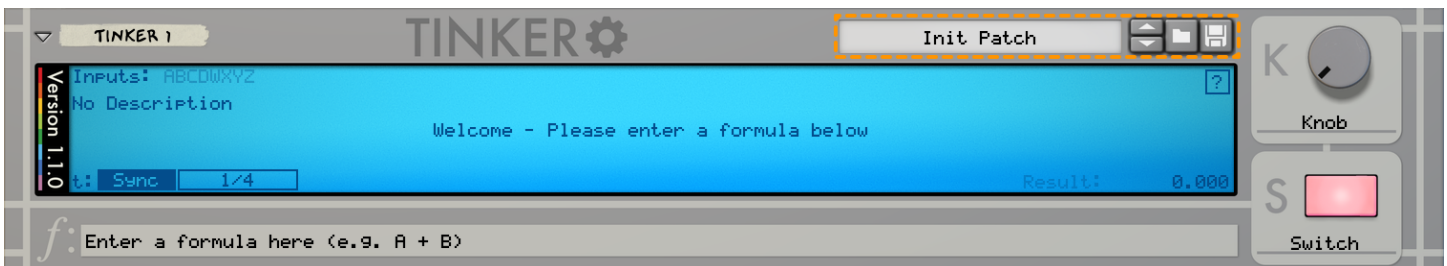
Help Button



Left: Section buttons, Center: Draggable area, Right: Back button

## Patch Browser

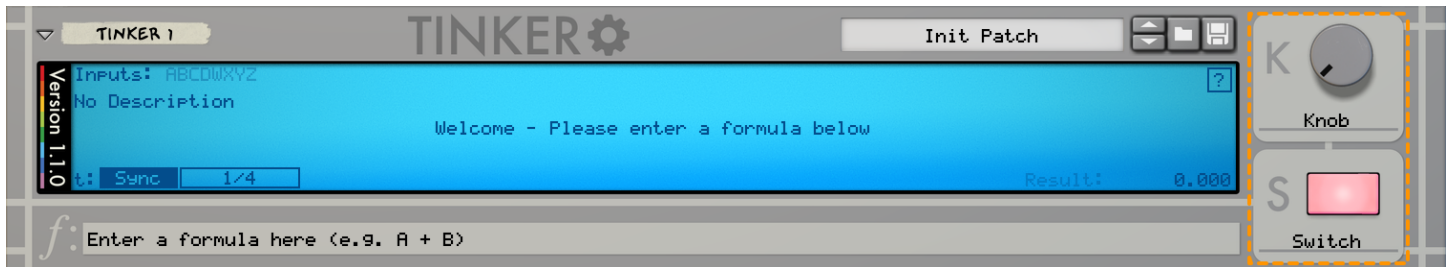
As with many Reason devices, Tinker lets you save patches. Tinker also includes a number of patches that serve as examples of its functionality.



Patch browser

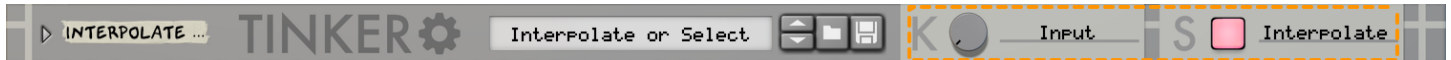
## Knob (K) and Switch (S)

On the right side of Tinker are a knob and switch that can be referenced in formulas using the variables K and S respectively. K ranges from 0 (fully left) to 1 (fully right). S equals 1 when on and 0 when off. Each control has a customizable label below it.



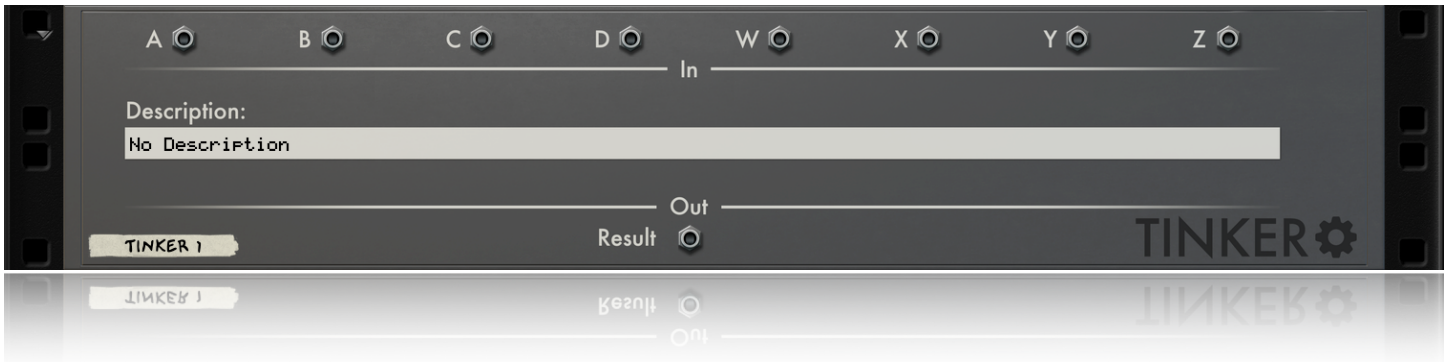
**Knob (K) and Switch (S) with labels below**

For convenience, K and S are also visible when Tinker is folded:



**Knob (K) and Switch (S) in folded view**

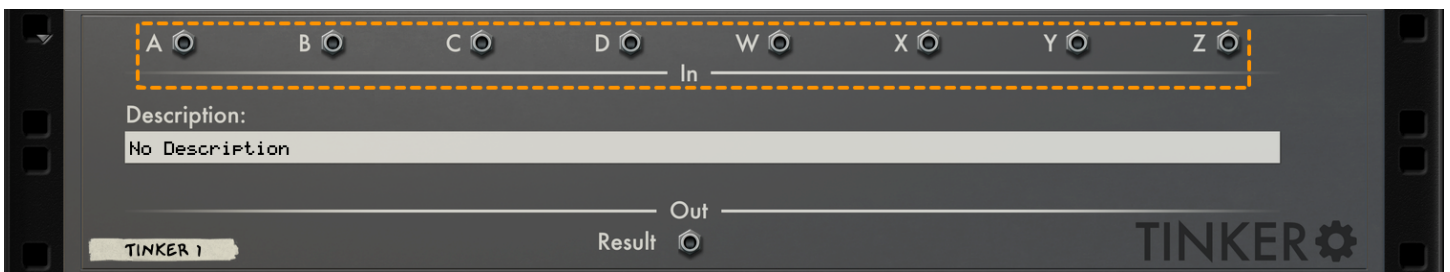
## Back



On the back of Tinker, you'll find its CV connections and an editable patch description.

### Variable Inputs (A, B, C, D, W, X, Y, Z)

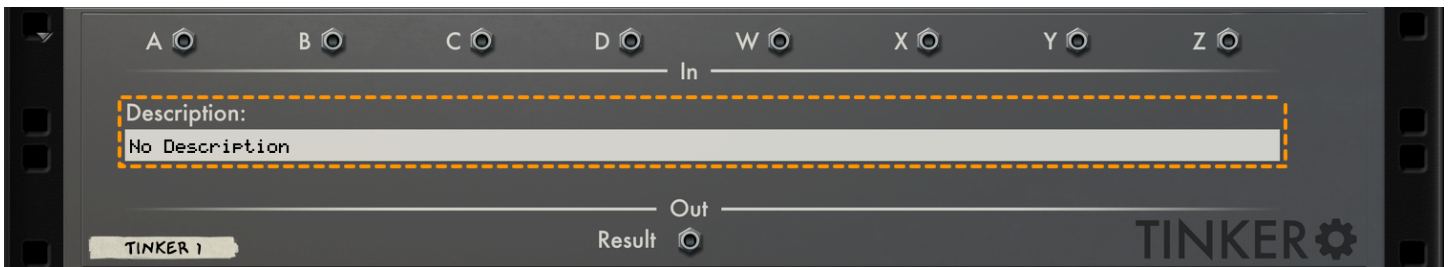
These eight CV inputs are labeled with the variables that can be used to reference them in formulas. When an input is disconnected, the corresponding variable will equal 0.



Variable CV Inputs

### Description Label

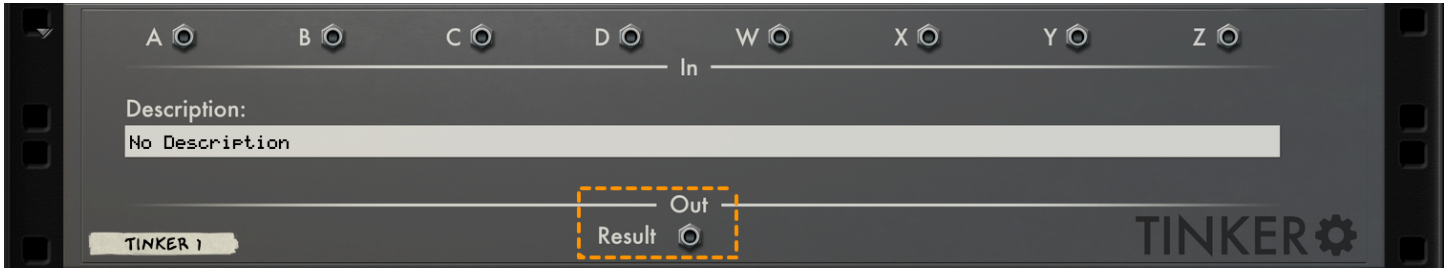
Double-click this label to enter a patch description that will be visible on Tinker's display.



Editable Description Label

## Result Output

This CV output transmits the result of evaluating the current formula. If an error occurs, this outputs 0.

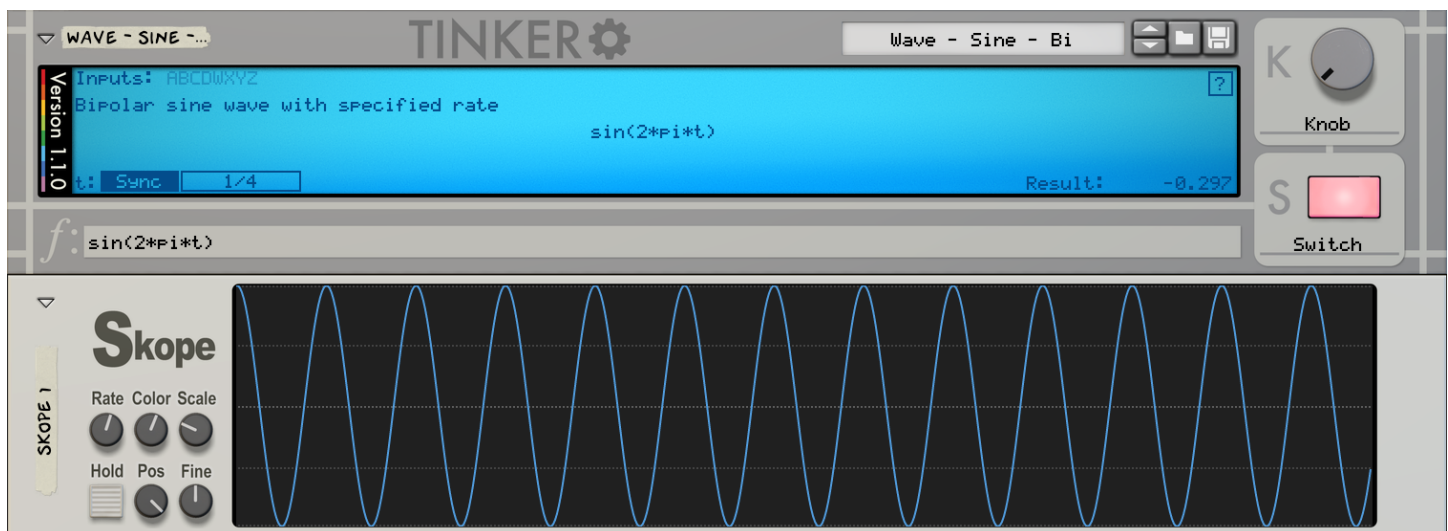


Result CV Output

## Tips and Tricks

This section contains some additional ideas for using Tinker more effectively, as well as recommendations for free Rack Extensions that you may find helpful.

- **Combine Multiple Tinkers:** Try using multiple instances of Tinker together. Use a Tinker's output as a variable in another. This can enable more complex formulas and help to reduce repetition.
- **Create a Bypass Switch:** Try using the select function to add a bypass switch to your patch. For example:
  - This formula will output  $\sin(A)$  when switch S is off (bypass disabled) and A when switch S is on (bypass enabled): `select(S, sin(A), A)`
- **Use Thor for More Controls:** By routing Thor's rotaries and buttons to its CV outputs, you gain four additional controls that can be hooked to Tinker's inputs. And going one step further, you can use the Combinator to build a uniform interface for all these controls.
- **Switchable Logic:** When fed binary signals (0 and 1), the min and max functions can serve as AND and OR respectively. This can be used to build complex conditions using multiple switches. For example:
  - Suppose you have four switches hooked to inputs A, B, C, and D. And suppose you want to check that A is off and all the others are on. This formula will output 1 in that situation and 0 otherwise: `min(1-A, B, C, D)`
  - Going one step further, we can use the select function to output W in that situation and X otherwise: `select(min(1-A, B, C, D), X, W)`
  - Try nesting min, max, and select to create more complex logic.
- **Use an Oscilloscope:** An oscilloscope is often useful when composing formulas. Some free options are [Skope](#) from Lectric Panda and [CVA-7](#) from pongasoft.



Tinker with Skope

- **Try Tinker's Siblings:** The inputs on the back panel of Tinker align with the outputs of [FRONT-8](#) and [RVL-1](#) (both free). RVL-1 is especially useful since it can provide labeled metering for Tinker's eight inputs. It can also split those inputs across multiple instances of Tinker.



Tinker with RVL-1

## MIDI / Remote Implementation

Parameter	MIDI CC	Remote Identifier
Tempo Sync	20	Tempo Sync
Rate Free	21	Rate Free
Rate Synced	22	Rate Synced
K (Knob)	23	K (Knob)
S (Switch)	27	S (Switch)