



CONTROL FORGE

Programmable
Universal CV Generator

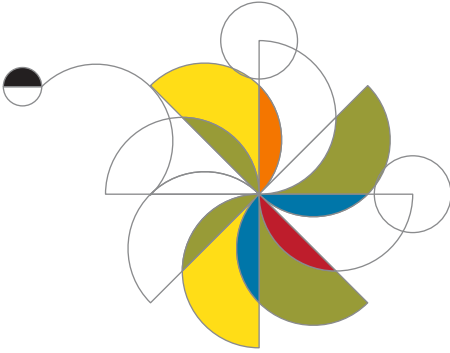
Operation Manual

v1.7_102616

Contents

1. Introduction	3
2. Module Installation	4
3. Overview	5
4. How to Use This Manual	7
5. Basic Functionality	9
6. Main Displays	14
7. Managing Presets	16
8. Preset Globals	18
9. Linear Contours	21
10. Randomness	27
11. Sequences	30
12. Conditional Jumps	33
13. AutoChain	38
14. Triggers	40
15. Real Time Control	42
16. Audio Rate Contours	44
17. Preset Sequencer	45
18. Utilities	49
19. Preset Templates	55
20. Demo Presets	58
21. Transition Shape Plots	64
22. Specifications	67
23. From Dave's Lab	68
24. Acknowledgements	69

1. Introduction



Support

In the unlikely event that you have a problem with your Control Forge, tell us about it here:

<http://www.rossum-electro.com/support/support-request-form/>

... and we'll get you sorted out.

If you have any questions, comments, or just want to say "Hi!," you can always get in touch here:

<http://www.rossum-electro.com/about-2/contact-us/>

... and we'll get back to you.

Happy music making!

Thanks for purchasing (or otherwise acquiring) the Rossum Electro-Music Control Forge Programmable Universal CV Generator.

This manual will give you the information you need to get the most out of Control Forge. However, the manual assumes you already have a basic understanding of synthesis and synthesizers. If you're just starting out, there are a number of good reference and tutorial resources available to get you up to speed. One that we highly recommend is:

Power Tools for Synthesizer Programming
(2nd Edition)
By Jim Aikin
Published by Hal Leonard
HL00131064

2. Module Installation

As you will have no doubt noticed, the rear of Control Forge is a circuit board with exposed parts and connections. When handling Control Forge, it's best that you hold it by the edges of the front panel or circuit board. It's not particularly easy to blow up, but why take chances?

More specifically, the biggest risk (to the extent that there's a risk), is damage by static electricity. Particularly on dry, cold days (or if you've just shuffled across your shag carpet in fuzzy slippers), make a point of touching the metal panel first, before touching any other part of the module.

While all Rossum Electro-Music modules are protected against reverse polarity damage, both to your module and your system, care should still be taken to connect the power cable correctly. (For more detail on our unique protection method, check out Dave's discussion of Circuit Protection in **Chapter 23**.)

Plug the included 16-pin connector into the header on the rear of the module such that the red stripe on the cable (the -12V side) is on the same end of the header as the "Red Stripe (-12V)" text on the PCB.

Control Forge requires, at most, 125mA of +12V and 25mA of -12V.

We have included both M3 and M2.5 (for vector rails) mounting screws. Use what fits your system.

If rack rash is of concern to you, use the included nylon washers when mounting Control Forge in your case.

3. Overview



Control Forge's core functionality is drawn from the E-mu Morpheus's insanely powerful (but hard to program on a two-line display) Function Generators. But now expanded into a supremely flexible modulation source

encompassing elaborate one-shot contours, evolving cyclical patterns and sequences of arbitrary length and complexity.

At its most basic, Control Forge is an 8-segment function generator with variable time and target voltage level for each segment.

Each segment's level can be defined either absolutely or relative to the ending level of the previous segment and can optionally be quantized to the nearest 1/12 volt.

Each segment's level may have a user defined range of randomness of either linear or gaussian distribution.

The time scale may be modulated under CV control or manually via the encoder during runtime.

There are 67 different transition shapes between segments (indicated graphically on the display during selection), from linear and exponential shapes to various flavors of random, chaotic, and many more that don't lend themselves to one-word descriptions, including the ability to pass CVs directly through to the output for individual segments.

Also included are "DC" shapes that allow the module to be used as a sequencer with all of the module's available programmability. When combined with Control Forge's Preset Sequencer, sequences can be hundreds of steps long and modified in real time manually or by CVs.

Each segment may optionally have a conditional jump that results in a jump to another selected segment (or to a completely different preset) if a particular condition is met. E.g., a CV being above or below the selected conditional value, the presence or lack of a gate or logic high at their respective inputs, rising or falling edges at the inputs, etc. So, in addition to one-shot shapes, extremely elaborate cyclical patterns can be programmed that evolve based on the various conditional inputs.

Control Forge's AutoChain functionality allows the creation of two very specific kinds of multi-preset patterns: A multi-preset retriggerable pattern (like a long syncable LFO), and a multi-preset "envelope-like" pattern with retriggerability, an optional sustain section, and a "release" section that's jumped to on gate fall.

Two independent triggers can be programmed to fire in response to a variety of events and can be used to affect either Control Forge itself or external modules.

500 presets can be saved and recalled.

A preset sequencer allows stepping through up to 200 user-defined series of presets under trigger, clock or manual control. This allows contours and sequences of almost unlimited length and complexity.

All presets and sequences can be transferred to any number of Satellite modules, which can then operate as stand-alone modulation sources, completely independent of the Control Forge.

In the following sections, we'll look at each of Control Forge's functions in turn.

4. How to Use This Manual

Control Forge is powerful. There's almost nothing you'd want to do with control voltages that you can't do with it. But with that power comes a lot of capabilities to become familiar with.

In designing the Control Forge interface, one of our main goals was to organize the features such that if all you wanted to do was program a simple 8-segment linear contour, you could do that without having to trip over all the rest of its many functions.

In keeping with that philosophy, we've organized this manual to take you sequentially from the most basic use scenarios to the most arcane, making it easy to learn just what you need for the task at hand. If you don't need conditional jumps or relative random target voltages, you can simply ignore them until some time when you do.

With that in mind, here is what the following chapters offer:

Chapter 5: Basic Functionality

This chapter introduces you to Control Forge's basic functional principles, including operational modes (Play/Program), inputs and outputs, setting parameters, and loading and saving presets.

Chapter 6: Main Displays

This chapter describes the elements of the main **Play Mode** and **Program Mode** displays.

Chapter 7: Managing Presets

This chapter describes saving and loading your Control Forge presets.

Chapter 8: Preset Globals

This chapter describes setting the global parameters that affect each entire preset.

Chapter 9: Linear Contours

This chapter takes you through the process of creating linear one-shot and cyclical ("LFO-style") contours using the **Time, Level, Level Mode**, and **Transition Shape** controls.

Chapter 10: Randomness

This chapter describes the addition of programmable amounts and distributions of randomness to individual segment's levels, as well as to the initial level of the preset when triggered or reset.

Chapter 11: Sequences

This chapter covers the use of the "DC" transition shapes and the **Quantize** function to create traditional (or not-so-traditional) step sequences.

Chapter 12: Conditional Jumps

This chapter covers Control Forge's extensive jumping and branching system, describing how the **Jump Mode, Conditional Value**, and **Jump Target** parameters can be used to create CV contours of arbitrary lengths that can change their shapes in response to Control Forge's real time controls or CVs from other modules in your system.

Chapter 13: AutoChain

This chapter describes Control Forge's ability to create extended multi-preset retriggerable "LFO-style" and multi-preset retriggerable "envelope-style" contours with optional "sustain" section(s), and a "release" section that's jumped to on gate fall.

Chapter 14: Triggers

This chapter covers the generation of Control Forge's two programmable triggers, which can be used either to affect Control Forge itself (e.g., controlling the **Preset Sequencer** or driving the **Logic** or **Gate/Trig** inputs) or to provide triggers or gates to other modules in your system.

Chapter 15: Real Time Control

This chapter describes the various ways that your CV contours can be modified in real time while they are playing. Options include internal controls (**CV 3**, **CV 4**, real time **Quantize** and **Randomize**, **Time Scaling**, manual **Preset Sequencer** control, and manual trigger generation), external triggers (**Gate**, **Logic** and **Preset Sequencer** control), and CVs (**CV 1**, **CV 2**, and **Time Scale CV**).

Additionally, Control Forge can be put in **Program Mode** while a preset is running, allowing you to reprogram segments while the preset continues to run.

Chapter 16: Audio Rate Contours

Describes running Control Forge at audio rate frequencies, essentially implementing a wavetable audio source.

Chapter 17: Preset Sequencer

This chapter covers the functionality of the **Preset Sequencer**, which allows you to program sequences of presets and navigate through the sequences either manually, using the panel controls, or under the control of triggers from Control Forge itself or from external sources.

Chapter 18: Utilities

This chapter describes various utility functions, including sending program data to a Rossum Electro-Music Satellite or another Control Forge, loading updated firmware or boot software, saving and reloading a backup of all your presets and sequences, calibrating your Control Forge, and more.

Chapter 19: Preset Templates

To help get you up and running quickly, we've provided templates of basic functions like envelopes, LFOs and step sequences to customize for your specific needs. This chapter describes those templates.

Chapter 20: Demo Presets

Some of our talented testers have been generous enough to provide a selection of presets that can give you an idea of some of the more creative Control Forge capabilities. They're described here.

Chapter 21: Transition Shape Plots

Graphic representations of Control Forge's Transition Shapes.

Chapter 22: Specifications

Yup, specifications.

Chapter 23: From Dave's Lab

Dave describes our unique approach to circuit protection.

Chapter 24: Acknowledgements

Thanks to the talented folks who contributed in one way or another to the development of Control Forge.

5. Basic Functionality

Before we jump into individual functions, let's take a look at Control Forge's basic organization.

Power Up

When power is applied to the module for the first time, it begins in **Play Mode** with Preset 001 as the Current Preset, Sequence 001 as the Active Sequence, and the contour stopped.

On subsequent power ups, the Preset and Sequence that were current at power off are restored.


Modes



The **Mode** button toggles the module between the **Play** and **Program** modes. One of the associated LEDs light to indicate the current mode.

As you've probably guessed, **Program Mode** is selected to

create your desired CV contours. Select **Play Mode** to play them.

 **NOTE:** It's possible to switch to **Program Mode** while a contour is playing and modify settings and values while it continues to play. This can be quite powerful in live performance.

Presets

CV contours created in **Program Mode** can be saved using the **Save Preset** function. A preset contains the individual settings for the 8 segments and a variety of **Global** settings that affect the overall preset.

In **Play Mode**, presets are played back. Presets can be manually loaded using the **Load Preset** function, loaded as the result of a **Conditional Jump**, as part of an **AutoChain**, or loaded by the **Preset Sequencer**.

Inputs



Gate/Trigger

This input, as its name so aptly suggests, accepts triggers and/or gates, i.e., a digital signal that can be high or low. Analog signals will be converted to high or low at a threshold of 1.65V. In order to be detected, the pulse width, either high or low, must exceed 25 microseconds.

Depending on settings in the **Preset Globals** menu (see Chapter 8), signals appearing here can either control a contour's start, reset, and jump to a "release" segment, or can be tested for a conditional jump.

The LED in the **Manual Gate** pushbutton will light while a gate is active, either as the result of a **Gate/Trigger** input signal or the **Manual Gate** button being pressed.

Logic

The **Logic** input accepts digital logic signals for evaluation for true/false conditional jumps. Analog signals will be converted to high or low at a threshold of 1.65V.

The associated Logic LED will light whenever the Logic input is high.

CV 1 & CV 2

These inputs accept CVs in the range of -5V to +5V. Voltages outside of that range are clipped to the appropriate limit.

The CVs at these inputs are tested for various conditional jump states.

Additionally, the CVs at these inputs are passed directly through to the output during any segment whose shape is **CV 1** or **CV 2 Passthrough**.

Time Scale CV

This input accepts CVs in the range of -5V to +5V. Voltages outside of that range are clipped to the appropriate limit.

The value of the CV modifies the **Time** values of all of a preset's segments such that a 1 volt change increases the Time Scale Rate value by a factor of 2. Therefore, overall range of modification is from 1/32 of the set time to 32 times the set time.

In Play Mode, the **Encoder** also controls time scale. Its value is multiplied by the Time Scale CV to provide the final time scale modification value.

Sequencer Dec, Inc, and Reset

Depending on the Preset Sequencer state, rising edges at these inputs control the loading of a sequence's programmed presets. See **Chapter 17** for details.

Outputs

OUTPUTS



+ OUTPUT



- OUTPUT



TRIGGER 1



TRIGGER 2

+ Output

Outputs the CV contour in the range of -5V to +5V. Accuracy is approximately +/-1mV.

- Output

Outputs a precisely inverted version of the CV contour in the range of -5V to +5V.

Trigger 1

Outputs a +10V trigger or gate pulse whenever the condition set in the **Assign Trigger 1** menu is met or the **Conditional Value/Trigger 1** button is pressed while in Play Mode.

Trigger 2

Outputs a +10V trigger or gate pulse whenever the condition set in the

Assign Trigger 2 menu is met or the **Assign Triggers/Trigger 2** button is pressed while in Play Mode.

Controls

Data/Time Scale Encoder



The Encoder is used in **Program Mode** to set the values of parameters selected via the parameter push buttons or knobs.

The Encoder includes an integrated pushbutton that is typically used to enter the currently selected value (see individual parameters for details).


In **Play Mode**, the encoder serves as a real time control over Time Scale. The time scale range is between X 0.031 and X 32X in 80 approximately logarithmic steps. Pressing and holding the encoder switch and turning it quantizes the Time Scale values to factors of 2.

Global/Pause/Utilities Button

In **Play Mode**, if Control Forge is running, quickly pressing this button causes playback to pause. Pressing it again causes playback to resume from the paused location. Pressing and holding this button brings up the **Utilities Menu** described in **Chapter 18**.

In **Program Mode**, quickly pressing this button brings up the **Preset Globals** menu described in **Chapter 8**. If Control Forge is running, pressing and holding this button causes pausing and resuming playback.

NOTE: Yes, we realize the short press/long press functionality is reversed between Play and

 Program Modes. While we prefer things to be consistent, after long discussion we came to the conclusion that it was more important that the things you were most likely to do in each mode were accessed with a short press.

Manual Gate Button



The **Manual Gate** button functions in both Play and Program Modes in the same manner as the **Gate/Trigger** input, allowing the generating of a gate for test or real-time performance purposes. The button and input are logically ORed together to create the Gate.

light while a gate is active, either as the result of a Gate/Trigger input signal or the Manual Gate button being pressed.

Segment Buttons

The LED in the Manual Gate pushbutton will



In **Play Mode**, while Control Forge is running, the buttons' LEDs light to indicate the active segment. No segment LED is lit when Control Forge is stopped.

select segments for editing. The LED of the segment being programmed will always be fully lit. If Control Forge is running, the LED of the active segment (if it is not the segment being programmed) will be more dimly lit.

Pressing one of the Segment Buttons (apart from Segment 1) while Control Forge is stopped causes it to immediately jump to that segment and to begin running from 0 Volts. If Segment 1 is pressed while stopped, it is treated as a **Manual Gate**, including activating the global Start/Reset parameters.

If Control Forge is running, pressing the button of the segment being edited will cause the generator to jump to that segment.

Pressing one of the Segment Buttons while Control Forge is running is treated as a **Jump Immediate** from the active segment to the pressed segment.


During the naming portion of the **Save Preset** and **Save Sequence** operations, the **Segment Buttons** can be used to enter alpha numeric characters.

Time and Level Knobs

In **Program Mode**, the buttons are used to



In **Program Mode**, these knobs set the values of the associated parameters.

 very wide range of possible values, when their editing window is visible, you can use the encoder to fine tune your selection. Pressing, holding, and turning the encoder, will quantize the selection to useful values.

In **Play Mode**, they allow manual real time setting of **CV 3** and **CV 4** for conditional jump tests.

Parameter Buttons

NOTE: Since both of these parameters have a



In **Program Mode**, the black parameter buttons are used to select parameters for editing and to load and save presets and sequences. In **Play Mode**, they are used for real time control input as indicated by their blue function labels.

Setting Parameters

As just mentioned, the **Time** and **Level** parameters are each set by adjusting their respective knobs. Whenever their value is displayed for editing, the **Encoder** can be used for fine adjustment.

The **Level Mode** is set by repeatedly pressing the **Mode/Quantize** button until the desired value is displayed in the **Program Mode** display. See **Chapter 9** for details.

All other parameters are set by pressing their parameter button to display the parameter's current value and then using the Encoder to set the desired value.

The value change is always effective immediately. Pressing the parameter's button while the parameter is displayed, or pressing the encoder, will exit the screen. The new parameter value remains in effect in either case.

IMPORTANT NOTE: *Whenever there are any unsaved parameter changes in a preset or sequence, a red warning (P for Preset and S for Sequence) will appear in both the Play and Program displays. If you select a new preset or sequence before saving them, the changes will be lost.*

Copying Segments and Parameters

For ease of programming multiple segments that may share some or all of their parameters, the following copying functions are available:

To copy a particular parameter value from one segment to another:

In Program Mode:

1. Select the segment whose parameter value you want to copy and bring the parameter's value up for editing.
2. Press and hold the source segment button and, while holding it, press and hold the segment button of the segment you want to copy the parameter value to.
3. After holding the destination button for about a second, the message "Parameter copied" will appear in the display and, not unexpectedly, the parameter value will have been copied to the destination segment and the destination segment will be selected.
4. The destination segment will be selected, but if you continue holding the original source segment button, you can repeat the procedure and sequentially copy the same source parameter value to as many of the other segments as you wish.

To copy all of the parameter values from one segment to another:

In Program Mode:

1. Select the segment whose parameter values you want to copy. Do not bring up any parameter edit boxes.
2. Press and hold the source segment button and, while holding it, press and hold the segment button of the segment you want to copy all the parameter values to.
3. After holding the destination button for

about a second, the message "Segment copied" will appear in the display and the parameter values will have been copied to the destination segment and the destination segment will be selected.

- 4 The destination segment will be selected, but if you continue holding the original source segment button, you can repeat the procedure and sequentially copy the same source parameter values to as many of the other segments as you wish.

6. Main Displays

Control Forge offers two main display layouts, depending on whether you are in **Play** or **Program** modes. Additionally, there are dedicated layouts for loading and saving presets and sequences, programming sequences, setting **Preset Globals**, and executing **Utility** functions.

In this chapter, we'll describe the main **Play** and **Programming** displays. The rest will be described in their own chapters.



Play Mode Display



When operating in **Play Mode**, the display provides the following:

- > A real-time graphic representation of the output voltage level with a range of -5V to +5V
- > A display of the time progress through the segments used in the preset (including any conditional jumps)

 **NOTE:** This display disappears when Control Forge is stopped. It flashes while Control Forge is paused.

- > The number and name of the currently loaded preset
- > The current numeric **Time Scale** multipliers. The first number is the Encoder multiplier and the second is the result of the **Time Scale CV** input.
- > The number and name of the currently loaded sequence
- > A display of the current segment's **Transition Shape**
 -  **NOTE:** The **Transition Shape** graphic will be inverted if the the segment's **Target Voltage** is lower than its starting voltage.
- > Graphical indicators of the real-time values of **CV 1**, **CV 2**, **CV 3**, **CV 4**, and the **Time Scale CV** (top to bottom in that order).
- > A sequencer status indicator which displays the current sequence step number (if the step contains a preset) or an indication that the sequence is in a **Halt** or **Pause** state.
 -  **NOTE:** If **Sequence Off** has been selected, this status display is blank.
- > Yellow status indicators:
 - Q:** Lights when real-time quantize is active, either because the button is being pressed or because it has been latched on.
 - R:** Lights when real-time randomize is active, either because the button is being pressed or because it has been latched on.
 - T1 & T2:** Light when a trigger has fired, either because its trigger condition has been met or because it's been triggered by the real-time TR 1 or TR 2 buttons.

INC, DEC, RST: Light when the Sequencer has received an Inc, Dec, or Reset command, either via triggers at the sequencer input jacks or because they've been triggered by the real-time DEC, INC or RESET buttons.

- > Red indicators that light when the current preset (P) and/or the current sequence (S) have unsaved changes.
- > An alphanumeric display of the currently active segment.

> **TIME** The programmed time for the selected segment. .0000 seconds to 9999 seconds.

The total time scale multiplier (Encoder x Time Scale CV) at the moment Program Mode was entered.

> **LEVEL** The Target Level in Volts. The Target Level Mode:

Abs: Absolute

AbsQ: Absolute Quantized

Rel: Relative

RelQ: Relative Quantized

Randomness in +/- Volts

Distribution:

Lin: Linear

Gau: Gaussian

> **SHAPE**

Shape name

Shape graphic

> **JUMP**

Jump Target (greyed out if not applicable)

Jump Mode

Conditional Value (greyed out if not applicable)

> An alphanumeric display of the segment currently being programmed.

> Trigger programmed indicators (**1&2**): Light when the respective trigger(s) have been programmed (i.e., aren't set to **Off**).

> Red indicators that light when the current preset (P) and/or the current sequence (S) have unsaved changes.

Program Mode Display



When operating in **Program Mode**, with no parameter selected for modification, the display provides the following:

- > A real-time graphic representation of the output voltage level with a range of -5V to +5V.
- > A display of the time progress through the segments used in the preset (including any conditional jumps).

 **NOTE:** This display disappears when Control Forge is stopped. It flashes while Control Forge is paused.


7. Managing Presets

Loading Presets



Press the **Load Preset** button to bring up the **Load Preset** display. Use the encoder to scroll through all of the existing presets, displaying the number and name of each one. Empty or erased presets are not displayed.

Press the encoder switch to select the desired preset and load it into the Control Forge and exit the Load Preset screen.

 **NOTE:** A long press of the **Load Preset** button will also load the preset displayed in the Load Preset screen. This makes reloading the current preset to easily and deliberately erase unwanted edits simply a matter of pressing **Load Preset**, and then immediately long pressing it.

Saving Presets



Press the **Save Preset** button to bring up the **Save Preset** display. The current preset is initially displayed.

Selecting a Location

When first entering this screen, the encoder controls the **Preset Number** field. Scroll through the numbers to select a destination location (or don't scroll to resave to the current preset's existing location). Note that this scrolling does include **--empty--** presets. At the top of the scrolling list is a special item, **Erase Preset**.

Saving without Changing Name

To save the current preset to the selected location without changing the location's existing name, either press the encoder for 2 seconds, or press the **Save Preset** button for 2 seconds. The display briefly displays "**Saving in 2 Seconds...**" and then (after 2 seconds) displays "**Save Completed**" and exits the Save Preset screen.

If you release the button before 2 seconds have elapsed, the display briefly shows "**Save canceled**" and exits the screen without executing the save operation.

 **NOTE:** The **Save Preset** operation can also be exited at any time without saving by pressing any other function button (other than **Segment** or **Manual Gate** buttons).

Changing the Preset Name



To change the name of the selected target location, press the encoder. A cursor will highlight the first character in the **Name** field. (If the selected location was **--empty--**, the Name field will change to the name of the current preset.)

The name can be changed using the **Segment Buttons** as follows:

- > Use the encoder to navigate to the desired character position(s).
- > Use the **Segment Buttons** to enter the desired character into each character location. Press a Segment Button repeatedly to cycle through that button's characters.
- > Press Segment 1 and 2 simultaneously to shift between lower and upper case.

Optionally, the name can also be changed using just the encoder:

- > After a highlighted character has been selected using the encoder, press the encoder to select that character for changes.
- > Rotate the encoder to select the new character. (At this point the segment buttons can also be used to enter characters.)


- > Press the encoder to choose the character and return the encoder to character location selection operation.
- > Repeat as necessary.


Saving after a Name Change

To save the renamed preset to the selected location, either long press the encoder, or long press the **Save Preset** button. This can be done while either in the character location selection or character changing operation for the encoder. The display briefly displays **"Saving in 2 Seconds..."** and then (after 2 seconds) displays **"Save Completed"** and exits the Save Preset screen. If you release the button before 2 seconds have elapsed, the display briefly shows **"Save canceled"** and exits the screen without executing the save operation.

Erasing the Current Preset

To erase a preset, press **Save Preset** and then scroll to the beginning of the menu to display **### **ERASE**** (where **###** is the preset number of the current preset which will be erased). While ****ERASE**** is displayed, either long press the **Encoder**, or long press the **Save Preset** button. The display will briefly display **Erasing Preset** and then exit the screen. If you release the button while **Erasing Preset** is displayed, the display briefly shows **Erase canceled** and exits the screen without executing the erase operation.

 **NOTE:** Erasing a preset that is a member of a Sequence will change the name of that preset to **"--erased--"** in the sequence step list and it will be skipped over when the sequence reaches that step.

 **ANOTHER NOTE:** If you, for some bizarre reason, erase all of the presets in your module, the next time the module boots up, the module will contain **Preset 001** named **--erased--** and set to the default preset values.

8. Preset Globals

The parameters in the **Preset Globals** menu apply to the entire preset, as opposed to specific segments. Since they define the basic behavior of a preset, you should typically set them before you otherwise start programming a new preset.



To access the **Preset Globals** menu, press the **Global** button while in **Program Mode**.



Initially, the display will show an overview of the current values of all of the global parameters. Rotate the encoder to scroll through the available parameter list. Click the encoder to choose the selected parameter for editing. The encoder will now adjust the parameter among available values, which become immediately effective. Click the encoder again to return its function to scrolling through the available parameters.

The menu is dismissed by:

- Pressing the **Global** button again.
- Selecting another parameter to set.
- Exiting **Program Mode**


Here's what the **Global** parameters do:

Gate Rise

This parameter defines what happens when a gate transition from low to high appears at the **Gate/Trig** input or as the result of pressing the **Manual Gate** button.

If **AutoChain** is not turned on for this preset (don't worry about AutoChain right now, we'll get to it later), the choices are:

- Off:** Nothing happens (unless a **Gate Rise** function has been programmed for a conditional jump in the active segment).
- Start/Reset:** Starts the contour at Segment 1 if it is not currently running, and forces an immediate jump to Segment 1 if it is running.

 **IMPORTANT NOTE:** *If the **Gate Rise** function has been programmed for a different conditional in the active segment, the segment function takes priority.*

If **AutoChain** is turned on for this preset, the choices are:

- Off:** Just like above, nothing happens.
- Start:** Forces an immediate jump to Segment 1 of the **Start Preset** in the chain.

The following settings all refer to the voltage level at the beginning of the contour:

Start Level: Reset or Current

If **Reset**, the contour starts at the voltage defined by the three **Reset** parameters below. If **Current**, the contour starts at the current voltage at the instant of the **Gate Rise**.

Reset Level: The voltage level at the beginning of the contour. -5.000 - +5.000 in 1mV steps. While the value is displayed for editing, press, hold, and turn the encoder to quantize the selected value to even whole volts. Release the encoder to continue editing if desired.

Reset Random Level: The range within which the voltage level at the beginning of the contour will vary from the selected **Start Level** level. +/-0.00V (no random variance) thru +/-5.00V

Reset Random Shape: Linear or Gaussian

Defines the distribution of potential random values within the randomization range. If **Linear**, there is an equal probability that any value within the range may be selected. If **Gaussian**, values nearer to the target level are more likely to be selected than values nearer the extremes of the range.

Gate Fall Jump

Defines what happens when a gate transition from high to low appears at the **Gate/Trig** input or as the result of releasing the **Manual Gate** button. Unlike a regular **Conditional Jump**, which is specific to an individual segment, the **Gate Fall Jump** is triggered anywhere in the contour, regardless of which segment is active.

The **Gate Fall Jump** choices are:

Off: A Gate transition from high to low has no special consequences (unless programmed in an individual segment).

Segment # (1-8): A gate transition from high to low causes an immediate jump to the selected segment (unless the **Gate**

Fall function has been programmed for a different conditional in the active segment, in which case the segment function takes priority).

If **AutoChain** is on, a gate transition from high to low forces an immediate jump to Segment 1 of the **Release Preset** in the chain (again, we'll get to **AutoChain** later).



NOTE: *The primary intended use of **Gate Fall Jump** is to provide a mechanism for creating the equivalent of the "Release" portion of a conventional ADSR envelope generator. However, we're sure you can come up with lots of other creative things to do with it.*

Free Run

On/Off: If **On**, the contour starts running at Segment 1 as soon as the Free Run preset is loaded. If Segment 8 is reached and no other jump programmed for it, the contour jumps back to Segment 1 and continues looping until the heat death of the universe (or the end of civilization as we know it, whichever comes first).

If **AutoChain** is on, if the last preset in the chain is reached and no other jump programmed for it, it jumps back to Segment 1 of the **Start Preset** in the chain.

AutoChain: Off/Start/Continue/Release

Any choice other than **Off** identifies the preset as a member of an **AutoChain** and identifies its role within the chain. (See **Chapter 13** for an explanation of **AutoChain**.)

Randomize Level


Sets the range of randomization of a segment's **Target Level** when pressing the real-time **Randomize** button while a contour is playing. The range is from +/-0.00V (no random variance) thru +/-5.00V

Randomize Shape

Linear/Gaussian: Defines the distribution of potential random values within the randomize range when pressing the **Randomize** button while a contour is playing. If **Linear**, there is an equal probability that any

value within the range may be selected. If **Gaussian**, values nearer to the target level are more likely to be selected than values nearer the extremes of the range.

NOTE: *While these last two parameters*


 *look pretty much identical to the **Reset Random Level** and the **Reset Random Shape** parameters above, they are completely independent of each other. The "**Reset**" parameters only apply to the level at the beginning of a contour. These last two only apply to real-time pressing of the **Randomize** button during play.*

Scale All Times

This function recalculates all of the preset's segments' **Time** parameters by the selected multiplier. Use it to adjust the overall duration of a contour while keeping the relative time relationships of all of the segments intact.


The range of multiplier is from 0.031x to 32.0x. The initial value when the editor appears is the most recently captured **Time Scale** value set by the encoder from **Play Mode** (if any). Once completed, the **Encoder Time Scale** is reset to 1.


NOTE: *This is a destructive function. After*


 *executing a time scaling, if you examine the individual **Time** values of the preset's segments, you'll find that they have been changed to the multiplied values. Of course, if you later change your mind, you can always execute another scaling that's the inverse of any previous one.*

9. Linear Contours

Whew! Now that we've covered all of that infrastructural stuff, it's time to make some presets.

 **NOTE:** If you'd rather get going without having to make anything from scratch, check out the template presets described in **Chapter 19**. However, if you want to customize them to your own needs (which, of course, you do), you should at least familiarize yourself with the basic segment parameters described in this and the next few chapters.

 **A TIP:** While Control Forge can be a modulation source for anything that accepts CVs, when programming or auditioning contours, it's easiest to hear the effect of your programming by routing the **+ Output** to a frequency CV input of an oscillator or other audio source.

 **ANOTHER TIP:** While everyone will develop their own routine for programming contours, we've found one that feels very efficient to us. Instead of selecting a segment, moving from parameter to parameter to program everything in that segment, and then proceeding to the next segment and programming everything in that one, we start with the first segment and program one parameter. Then while that parameter is still displayed for editing, we press the next **Segment Button** and program that parameter for that segment, continuing until that parameter is programmed for all 8 (or however many you're using) segments. Then return to Segment 1, select another parameter, and continue the process until all segments are programmed. We find it saves a lot of button pressing. Give it a try.

At its most basic, a Control Forge CV contour is made up of up to 8 segments, each one of which has a **Target Level** (the voltage the CV will reach at the end of the segment) and a **Time** (the amount of time, in seconds, the segment will take to reach its Target Level). A very simple contour might look like this:

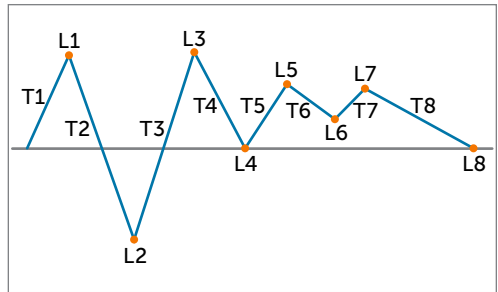



FIGURE 1: SIMPLE CONTOUR


Time Control



The **Time** knob sets the Time parameter of the currently selected segment. The range of times is .0001 second to 9999 seconds (over 2.75 hours!).

Initially turning the knob brings the Time value to the fore on the display and modifies it. While the Time value is thus displayed, you can use the Encoder to fine tune your selection.

 **NOTE:** While the value is displayed for editing, press, hold, and turn the encoder to quantize the selected value to useful even time values. Release the encoder to continue editing if desired.

 **ANOTHER NOTE:** To tweak a previously programmed Time value (without having to wiggle the knob and change the setting), with no parameter editing box visible, click the encoder to display the current value of the segment's Time or Level (whichever was most recently edited). If that's the Level rather than the Time parameter, click the encoder twice more to dismiss the Level parameter and display the Time parameter.

Once modified, the display can be dismissed by:

- Pressing the encoder button
- Selecting another parameter to edit

Programming Rhythm

There are two basic approaches to programming your contour's rhythm, absolute segment duration and note-based ratios. Which you use will depend on how you intend to use your contour.

Absolute Segment Duration

Use this method when it's important to you that each segment last for an exactly defined duration. In **Play Mode**, use the encoder to ensure that Time Scale is set to 1.00. Then enter **Program Mode** and program the duration of each segment to whatever length you desire.


Note-Based Ratios

With this method, you think in terms of note values (e.g., quarter notes, half notes, etc.) and start by arbitrarily selecting a note duration to represent a specific note value (e.g., let 1.00 seconds = an eighth note), set all your segments to the desired ratios of that value, and then use Time Scaling to define your tempo.

An Example: As above, use the encoder in **Play Mode** to ensure that Time Scale is set to 1.00. Then enter **Program Mode**. If, as suggested above, you let **1.00 seconds** represent an **eighth note**, program the value of each segment in your contour using the following table:

NOTE	SECOND(S)
32 nd note	0.250
16 th note triplet	0.334
Dotted 32 nd	0.375
16 th note	0.500
8 th note triplet	0.667
Dotted 16 th note	0.750
8 th note	1.000
Quarter note triplet	1.334
Dotted 8 th note	1.500
Quarter note	2.000
Half note triplet:	2.667
Dotted quarter note	3.000
Half note	4.000
Dotted half note	6.000
Whole note	8.000
Etc...	

Once your contour is programmed, use the encoder to select your desired tempo by adjusting the **Time Scale** in **Play Mode**. All of the notes will retain their correct rhythmic relationship.

 **A TIP:** Once you've adjusted the **Time Scale** to provide your desired tempo, note the **Time Scale** multiplier value and use the **Scale All Times** function in the **Preset Globals** menu to permanently scale all of the segment times to produce that tempo (keeping in mind that you can always still modify it later by once again using the encoder to scale the new times).

Level Controls


The Segment's **Target Level** is set by a combination of the **Level** knob, the **Mode/Quantize** button, and the **Randomness** button. We'll cover the first two here.


Randomness will be covered on its own in the next chapter.


Level Knob

The **Level** knob sets the base **Target CV Level** (i.e., the CV level that will be reached at the end of the segment) of the currently selected segment. The range of Levels is -5.000V to +5.000V.

Initially turning the knob brings the **Level** value to the fore on the display and modifies it. While the **Level** value is displayed, you can use the encoder to fine tune your selection.

 **NOTE:** While the value is displayed for editing, press, hold, and turn the encoder to quantize the selected value to even while volts. Release the encoder to continue editing if desired.

 **ANOTHER NOTE:** If, while the **Level** value is displayed, you hold the **Mode/Quantize** button and turn the **Level** knob, the resulting values will be quantized to exact semitones (i.e., $1/12^{\text{th}}$ volt) and displayed as octaves and semitones (e.g., "+2 oct +06 semi"). Once the editing display is dismissed, the **Level** value on the main display will be in the format of the currently selected mode (see below).

 **YET ANOTHER NOTE:** To tweak a previously programmed **Level** value (without having to wiggle the knob and change the setting), with no parameter editing box visible, click the encoder to display the current value of the segment's **Time** or **Level** (whichever was most recently edited). If that's the **Time** rather than the **Level** parameter, click the encoder twice more dismiss the **Time** parameter and display the **Level** parameter.

Once modified, the display can be dismissed by:

- Pressing the encoder button
- Selecting another parameter to edit

Mode/Quantize Button

Sequentially pressing this button cycles through the four available **Level Modes**. The current mode appears on the main display. An associated LED lights to indicate whether the current mode is **Absolute** or **Relative**.


The four modes are:

Absolute (Abs): The **Target Level** for the segment is the level set by the **Level Knob**.

Absolute Quantized (AbsQ): The **Target Level** for the segment is the level set by the **Level Knob** quantized to the nearest $1/12^{\text{th}}$ volt, displayed as octaves and semitones.

Relative (Rel): The **Target Level** for the segment is the sum of the ending level of the previous segment and the level set by the **Level Knob**.

Relative Quantized (RelQ): The **Target Level** for the segment is the sum of the ending level of the previous segment and the level set by the **Level Knob**, quantized to the nearest $1/12^{\text{th}}$ volt, displayed as octaves and semitones.


 **NOTE:** In either **Relative Mode**, if the resulting level would be above +5V or below -5V, the actual target is set at +5V or -5V.

A Relative Mode Example: Assume Segment 2 is in **Relative Mode** and its **Level** is +1.000 volt.

If Segment 1 has a **Target Level** of +1.500 volts and is followed by Segment 2, the actual **Target Level** of Segment 2 would be +2.500 volts.

However, if Segment 2 is instead jumped to from Segment 6, which has a **Target Level** of -2.250 volts, the actual **Target Level** of Segment 2 would then be -1.250 volts.

The point being that a Segment in **Relative Mode** can have radically different effective **Target Levels** depending on its immediately previous segment.

 **NOTE:** The quantization function also affects any external CVs that are routed to the output via one of the CV Passthrough shapes.

Transition Shape



TRANSITION
SHAPE/DEC

In the graphic at the beginning of this chapter, each Segment progresses in a straight line from its beginning voltage level to its target level. But one of the most



powerful features of Control Forge is that each Segment can travel in a trajectory defined by one of 67 different transition shapes, from linear

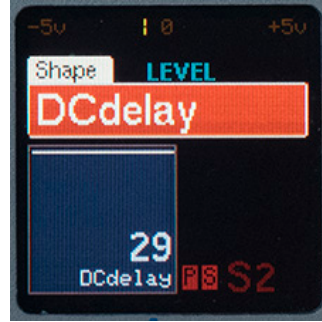
and exponential shapes to various flavors of random, chaotic, and many more unusual shapes, including the ability to pass CVs directly through to the output for individual stages.

The available shapes include:

- Linear
- Exponential 1-7
- Circle 1.4, 1.6, 1.8, 1.16
- Squeeze
- Fast Line 1-3
- Medium Line 1-2
- Slow Ramp 1-2
- Bloom and Bloom 2
- Circle 1.16, 1.8, 1.6, 1.4 Reverse
- Slow Curve 1-2
- Delay DC and DC Delay
- Curve 2x
- Curve 2x B-C
- Zig Zag 1-3
- Chaos 03, 06, 12, 16, 25, 33, 37, 50, 66, 75, 95, 99
- Linear Shuffle and Linear Shuffle 2
- Random A-L, Z
- CV1-4 Passthrough

You'll find graphs of all the shapes in **Chapter 21**.

About Delay DC and DC Delay Shapes



These two shapes are special cases designed primarily for use in step sequences, where you'd typically like each segment's "note" to remain

unchanged for the entire length of the segment.

With **DC Delay**, the segment's CV jumps immediately to the **Target Level** and remains there for the entire length of the segment.

With **Delay DC**, the segment's CV stays at the ending voltage of the previous segment for the entire length of the segment and then jumps to its own **Target Level** at the very end of the segment.

About the ZigZag 1, 2, and 3 Shapes

These shapes appear to have two separate curves in the same space, but what happens when they're selected is that the CV output alternates back and forth, point by point, between the two curves. (Check out their plots in **Chapter 21** or listen to them driving a VCO and it will be clear how each works.)

About CV 1-4 Passthrough Shapes

When one of the four **CV Passthrough Shapes** are selected for a segment, during the **Time** programmed for that segment, the CV output is exactly the CV that is present at the **CV 1** or **CV 2** inputs or the CV that results from the positions of the **CV 3** or **CV 4** knobs. This vastly expands the range of possible CV contours, as well as offers dramatic effects when Control Forge is operated at audio rates (see **Chapter 16** for details).

To select a **Transition Shape**, press the **Transition Shape** button. This brings the **Transition Shape** parameter to the fore on the display along with a graphic representation of the shape. Select an available shape using the encoder.

The display is dismissed by:

Pressing the **Transition Shape/Dec** button again.

Pressing the encoder.

Selecting another parameter to set.

Programming Simple One-Shot Contours

Now that we know what the controls do, let's program a preset. For this first one, we'll make a simple one-shot contour that starts when it receives a gate or trigger at the **Gate/Trigger** input (or you push the **Manual Gate** button) and stops when it reaches the end of the final segment.

1. To start, use the **Load Preset** function to load any preset. Immediately use the **Save Preset** function to save the preset to an empty location. Name it "**1 Shot**" (or whatever else you prefer).
2. Make sure your Control Forge is in **Program Mode**. Press the **Global** button to access the **Preset Globals** screen.
3. Set the Global parameters to the following values and then press **Global** again to dismiss the menu:

Gate Rise: Start/Reset

Start Level: Reset

Reset Level: +0.000

Random Level: +/-0.00

Random Shape: Linear

Gate Fall Jump: Off

Free Run: Off

AutoChain: Off

Randomize Level: +/-0.00

Randomize Shape: Linear

4. Press the **Segment 1 Button** to edit the first segment.

5. Turn the **Time Knob** to bring up the editing display. Use the knob and encoder to select your desired time for **Segment 1**.
6. While the selected time is still displayed, press the **Segment 2 Button** and set its time. Continue until all 8 segments' time parameters have been set.
7. Return to **Segment 1** and turn the **Level Knob** to bring up its editing display. Again, use the knob and encoder to select your desired level and continue as before to program levels for all 8 segments.
9. Return to **Segment 1** and press the **Transition Shape** button to bring up its editing display. Use the encoder to select your desired transition shape and continue once again to program transition shapes for all 8 segments.
10. With the main **Program Mode** screen visible, press each **Segment Button** in turn and confirm that the **Level Mode** for each segment is **Absolute (Abs)**, each segment's **Jump Mode** is set to "**Never**" and the **Randomize** parameter in the **Level** section is set to +/-0.00. If any aren't, use the appropriate controls to set them to those values.
11. Finally, you're ready to audition your contour! Connect the **+ Output** to the frequency input of a VCO, switch your Control Forge into **Play Mode**, press the **Manual Gate** button, and listen to the result. You'll see the CV level indicator tracing out the shape of your contour while the segment progress indicator will step through the 8 segments and then stop at the end of **Segment 8**.
Note that if you press the **Manual Gate** button while the contour is playing, it will immediately jump back to **Segment 1** and continue from there.
12. It's unlikely that any contour will be just what you want on the first try. Here're some things to try:

- > In **Play Mode**, turn the encoder to speed up or slow down your contour, while maintaining the segments' time relationships.
 - > Adjust the **Times** or **Levels** of individual segments.
 - > Change the **Transition Shapes** of individual segments.
 - > Change the **Level Modes** of individual segments to **Absolute Quantized (AbsQ)** and listen to the effect.
 - > Change the **Level Modes** of individual segments to **Relative (Rel)** and listen to the effect.
 - > Patch a CV from another module into the CV 1 input and set one of the segment's Transition Shapes to CV 1 Passthrough.
13. At this point, resave your preset to the same location, so you can come back to it later if desired and continue editing.

Programming Cyclical (LFO-Style) Contours

For this next one, we'll make a simple repeating contour that starts as soon as it's loaded and retriggers from the start whenever it receives a subsequent trigger or gate at the **Gate/Trig** input.

1. Press the **Global** button to access the **Preset Globals** screen and set the Global parameters to the following values (they're the same as the last example except for turning **Free Run** on). Then press **Global** again to dismiss the menu:

Gate Rise: Start/Reset

Start Level: Reset

Reset Level: +0.000

Random Level: +/-0.00

Random Shape: Linear

Gate Fall Jump: Off

Free Run: On

AutoChain: Off

Randomize Level: +/-0.00

Randomize Shape: Linear

10. Randomness

Random variations in CV levels can bring a subtlety organic feeling to a repeating pattern, as well as allow the creation of much more dramatic randomized contours. Control Forge provides three independent sources of randomization and lets you select both the range within which randomization will occur and the probabilistic distribution of the random results within that range, independently for each source.

Randomization Range



This parameter defines the symmetrical range, in volts, within which the programmed CV Level will be randomly modified.

The range of values is from

+/-0V to +/-5V in 10mV steps. For example, if you select a randomization range +/-1.5V, the actual CV will be somewhere between 1.5V below the programmed CV level and 1.5V above it.

Randomization Distribution



This parameter defines the distribution of potential random values within the randomization range. If **Linear**, there is an equal

probability that any value within the range may be selected. If **Gaussian**, values nearer to the target level are more likely to be selected than values nearer the extremes of the range.

Here are the places you can program randomization:

Initial Level Randomization

When a contour is started or retriggered from Segment 1, it starts at the level set by the **Start Level** parameters in the **Preset Globals** menu. The **Reset Random Level** and **Reset Random Shape** parameters allow you to select an optional amount and distribution of randomness that varies the **Start Level** each time the contour is started or retriggered.

Real Time Randomization

As will be described in detail in **Chapter 15**, while a contour is playing, you can add randomization to the active segments in real time by pressing the **Randomize** button. The amount and distribution of this randomization is set by the **Randomize Level** and **Randomize Shape** parameters in the **Preset Globals** menu. (Note that these parameters are separate and independent from the **Initial Level Randomization** parameters described above.)

Segment Randomization

Each segment can optionally have a programmed amount of random variation applied to its programmed **Target Level**. The randomness parameters are programmed via the **Randomness Button** and interact with the **Level Mode** as follows:

Randomness Button



Pressing this button allows you to edit the **Randomness** parameter. Pressing the **Randomness** button a second time allows you to select

between **Linear** and **Gaussian** random distribution shapes, as described above. Pressing the encoder again, or pressing the **Randomness** button from here exits back to **Program Mode**.

This **Randomness** setting interacts with the **Level Mode** setting as follows:

- > If **Level Mode** is set to **Abs** or **AbsQ**, the **Randomness Amount** defines the symmetrical range above and below the **Target Level** setting within which the final **Target Level** will be randomly set.

An example: If the **Target Level** is set to 0.75V and the **Randomness Amount** is set to 1.25V, the actual Target values will fall somewhere between 2.00V and -0.50V.

- > If **Level Mode** is set to **Rel** or **RelQ**, the **Randomness Amount** defines a symmetrical range above and below the **Relative Target Level** setting within which the **Target Level** will be randomly set. The resulting value is summed with the previous segment's final value to obtain the actual **Target Level** value.

Another example: If the **Relative Target Level** is set to -1.2V and the **Randomness Amount** is set to 0.25V, the actual Target value will fall somewhere between 0.95V and 1.45V **below** the previous segment's final value.

Some Randomness Examples Sample and Hold in One Segment

With the **Randomness** parameter, we can program a flexible sample and hold using only one segment. Here's how:

1. Use the **Load Preset** function to load any preset. Immediately use the **Save Preset** function to save the preset to an empty location. Name it "**1 SEG S&H**" (or whatever else you prefer).

2. Enter **Program Mode** and program **Segment 1** as follows:

Time: 1.000 sec.

Level: 0.000V Absolute

Randomness: +/-3.00V Linear

Transition Shape: 29 DC Delay

Jump Mode: Always at End

Jump Target: Segment 1

Assign Trigger 1: Begin Segment 1

3. Patch the Control Forge's **+ Output** to the 1V/Oct input of an oscillator and the **Trigger 1** output to the Gate In of an envelope.
4. Patch the output of your oscillator through whatever processing you want and into a VCA controlled by the envelope.
5. Switch to **Play Mode** and press Control Forge's **Manual Gate** button to start your contour running and listen to the result.
6. Use the Encoder to control the rate of your sample and hold.

So, here's what's happening: You've created a loop that continually plays Segment 1 over and over. Since it's using the DC Delay shape, the voltage output remains steady each time through the segment. With randomness set to +/-3V, each time through the segment, the voltage will be somewhere between -3V and +3V (i.e. a +/- 3V range around the programmed **Target Level** of 0.000V).

Here are some things you can do to modify your sample and hold:

- > Change the **Randomness** amount. The overall range of the sample and hold can be anywhere from a few cents up to a full 10 octaves with **Randomness** set to +/- 5.00V.
- > Change the **Randomness** mode to **Gaussian** to maintain the full range of the amount setting, but have more of the values be closer to the Target Level of 0V.
- > Set a different **Target Level** to offset the **Randomness** range. Instead of 0V, a **Target Level** of 1.500V for example,

would result in a sample and hold range of -1.500V to 4.500V

- > Change the **Level Mode** to **Absolute Quantized** to have all of your sample and hold values be quantized to semitones.

Drifting Random Variation

Here, we'll use **Relative Level Mode** and **Randomness** to program a contour that drifts randomly as it alternates between positive and negative-going segments.

1. As above, load a preset and then immediately save it to an empty location. Name it "**Rand Drift**" (or whatever else you prefer).
2. Enter **Program Mode** and program **Segment 1** as follows:
Time: 1.000 sec.
Level: +0.350V Relative
Randomness: +/-0.19V Linear
Transition Shape: 1 Linear
Jump Mode: Never
Jump Target: (Not Used)
3. Program **Segment 2** as follows:
Time: 1.000 sec.
Level: -0.350V Relative
Randomness: +/-0.19V Linear
Transition Shape: 1 Linear
Jump Mode: Always at End
Jump Target: Segment 1
4. Switch to **Play Mode** and press Control Forge's **Manual Gate** button to start your contour running and listen to the result.
5. Use the encoder to control the rate.

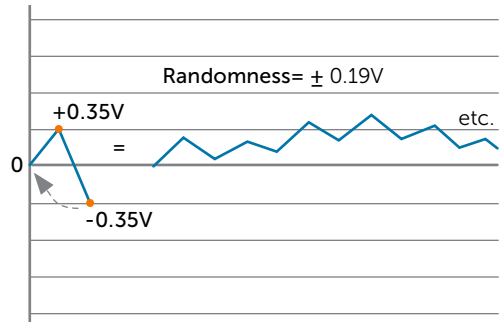


FIGURE 2: DRIFTING RANDOM CONTROL

Here's what's happening: Starting at 0V, **Segment 1** heads up to somewhere between 0.160V and 0.540V (0.350V +/-0.19V). Then **Segment 2** heads down to between 0.160V and 0.540V **below** wherever **Segment 1** ended up, and then jumps back to **Segment 1**, which starts up from there, again going to between 0.160V and 0.540V **above** wherever **Segment 2** ended up. This continues for as long as you let the contour run.

As in the previous example, you can modify your contour by changing **Transition Shapes**, **Target Levels**, **Randomness** amounts and distributions, etc.

11. Sequences

While strictly speaking, *any* series of CVs output from Control Forge can be considered a sequence, in this chapter we're going to address using Control Forge more in the style of a traditional step sequencer (although it's capable of sequencing that's anything but traditional).

Quantization

On the (admittedly not necessarily accurate) assumption that you will be creating "melodic" sequences, you will most likely want to set your **Level Mode** to **Absolute Quantized (AbsQ)**. In this mode, **Target Level** is quantized to the nearest 1/12 volt and is specified in octaves and semitones, from **"-5o +00 semi"** to **"+5o +00 semi."**


Obviously, if you want something other than diatonic semitones, you can select **Absolute (Abs)** mode and program your own intervals accurate to the thousandth of a volt.

Transition Shape



Again, if you want to program a series of "notes" (whether diatonic or not), you will most likely want to select **Transition Shape #29, DC Delay**.

When **DC Delay** is selected for a segment, Control Forge's output immediately jumps to the **Target Level** at the beginning of the segment and remains there for the segment's entire length. In practice, that means that each segment will output its programmed note in sequence.

 **A TIP:** You can alternate **DC Delay** segments with other transition shapes to create sequences that intersperse sustained notes with dynamic transitions.

Programming Rhythm and Tempo

There are two basic approaches to programming your sequence's rhythm, absolute note duration and note-based ratios. Which you use will depend on how you intend to use your sequence.

Absolute Note Duration

Use this method when it's important to you that each note last for an exactly defined duration. In **Play Mode**, use the encoder to ensure that **Time Scale** is set to 1.00. Then enter **Program Mode** and program the duration of each note to whatever length you desire.


Note-Based Ratios

With this method, you think in terms of note values (e.g., quarter notes, half notes, etc.) and start by arbitrarily selecting a note duration to represent a specific note value (e.g., let 1.00 seconds = an eighth note), set all your notes to the desired ratios of that value, and then use **Time Scaling** to define your tempo.

An Example: As above, use the encoder in **Play Mode** to ensure that Time Scale is set to 1.00. Then enter **Program Mode**. If, as suggested above, you let 1.00 seconds represent a quarter note, program the value of each note in your sequence using the following table:

NOTE	SECOND(S)
32 nd note	0.250
16 th note triplet	0.334
Dotted 32 nd	0.375
16 th note	0.500
8 th note triplet	0.667
Dotted 16 th note	0.750
8 th note	1.000
Quarter note triplet	1.334
Dotted 8 th note	1.500
Quarter note	2.000
Half note triplet	2.667
Dotted quarter note	3.000
Half note	4.000
Dotted half note	6.000
Whole note	8.000
Etc...	

Once your sequence is programmed, use the encoder to select your desired tempo by adjusting the **Time Scale** in **Play Mode**. All of the notes will retain their correct rhythmic relationship.

 **A TIP:** *Once you've adjusted the Time Scale to provide your desired tempo, note the Time Scale multiplier value and use the Scale All Times function in the Preset Globals menu to permanently scale all of the segment times to produce that tempo (keeping in mind that you can always still modify it later by once again using the encoder to scale the new times).*

Long Sequences

There are a number of ways to create sequences longer than the 8 segments of a single preset. Which one you use depends on how you intend to organize your sequence. Here are some options:

Linear: If you are creating one long sequence that's intended to play straight through or repeat in its entirety, it's probably easiest to simply program a **Jump Always at End** in the last segment of each preset in your sequence, and set its **Jump Target** to the next preset in the sequence. Just keep on repeating the process until the entire sequence is programmed. The last segment of the last preset can be programmed either to stop or to jump back to the first preset in the sequence and repeat.

Use the Preset Sequencer: If you have structured your sequence out of separate preset sections (e.g., an intro, a chorus, a verse, a break, a different chorus, etc. (although, of course, you don't have to use song structure sections)), use the **Preset Sequencer** to sequence them in whatever order you need. An advantage of this method is that you can use the same section in multiple locations in your performance without having to duplicate their programming.

Depending on how you want to play back your sequence, you can:

- > Program one of Control Forge's triggers to fire at the end of the last segment of each preset and patch that trigger to the sequencer's **Inc** input to automatically move to the next preset.
- > Patch a trigger or gate from another module in your system to control the progress of the presets.
- > Move from preset to preset manually using the **Dec**, **Inc**, and **Reset** buttons

Using Conditional Jumps

With **Conditional Jumps**, you can create extended sequences with sections whose order changes based on external CVs or gates. For example, you can program a conditional jump in a segment of a preset that jumps to another preset if a gate is high at the **Logic** input, while if it is low, the sequence would continue to the next segment where, if the Logic remains low, it would jump to a yet different preset.

As should be obvious, the options here are almost limitless.

Use AutoChain

Although not the primary intended purpose of **AutoChain**, you can certainly create an **AutoChain** that serves as an extended sequence. See **Chapter 13** for details.

12. Conditional Jumps

Conditional Jumps are, without a doubt, one of the most powerful of Control Forge's capabilities. They give you the ability to create CV contours and sequences that actually change their content based on input from other modules in your system or from manual controls on Control Forge itself.

Overview



For each segment in a Control Forge

preset, you can program a conditional jump that results in a jump to another selected segment (or to a completely different preset) if a particular condition is met. Examples of conditions include a CV at the selected conditional value, the presence or lack of a gate or logic high at their respective inputs, rising or falling edges at the inputs, etc. So, in addition to one-shot shapes, extremely elaborate cyclical patterns and sequences can be programmed that evolve based on the various conditional inputs.

Conditional Jumps are programmed using the **Jump Mode**, **Jump Target**, and **Conditional Value** parameters.



Jump Mode

The **Jump Mode** parameter selects which of the available conditions will be tested to decide if a jump will be executed. In **Program**

Mode, pressing the **Jump Mode** button brings the parameter to the fore on the display. Select an available mode with the encoder.

Available **Jump Modes** are:

Never: Never jumps; the segment always proceeds to the next segment in numerical order.

NOTE: *If the preset is not set to be **Free Run**, the contour stops at the end of **Segment 8**. If it is set to **Free Run**, the preset automatically loops back to **Segment 1** from the end of **Segment 8** (unless something else has been expressly programmed).*

Always at End: Always jumps to the **Jump Target** at the end of the current segment.

on Gate Rise: Jumps to the **Jump Target** immediately upon a rising edge at the **Gate/Trig** input.

on Gate Fall: Jumps to the **Jump Target** immediately upon a falling edge at the **Gate/Trig** input.

wait4GateRise: Jumps to the **Jump Target** immediately upon a rising edge at the **Gate/Trig** input. If the end of the segment is reached without a gate rise happening, the segment stops at its end,

maintaining its **Target Level**, until a gate rise does arrive, at which time it jumps to its **Jump Target**.

if@end GateHi: Jumps to the **Jump Target** at the end of the segment if the **Gate** is high at that time.

if@end GateLo: Jumps to the **Jump Target** at the end of the segment if the **Gate** is low at that time.

on Logic Rise: Jumps to the **Jump Target** immediately upon a rising edge at the **Logic** input.

on Logic Fall: Jumps to the **Jump Target** immediately upon a falling edge at the **Logic** input.

wait4LogicRise: Jumps to the **Jump Target** immediately upon a rising edge at the **Logic** input. If the end of the segment is reached without a **Logic** rise happening, the segment stops at its end, maintaining its **Target Level**, until a **Logic** rise does arrive, at which time it jumps to its **Jump Target**.

if@end LogicHi: Jumps to the **Jump Target** at the end of the segment if **Logic** is high at that time.

if@end LogicLo: Jumps to the **Jump Target** at the end of the segment if **Logic** is low at that time.

when CV# (1-4) > : Jumps to the **Jump Target** immediately when CV# exceeds the specified voltage.

when CV# (1-4) < : Jumps to the **Jump Target** immediately when CV# drops below the specified voltage.

if@end CV# (1-4) > : Jumps to the **Jump Target** if CV# exceeds the specified voltage at the end of the segment.

if@end CV# (1-4) < : Jumps to the **Jump Target** if CV# is below the specified voltage at the end of the segment.

The display is dismissed by:

- Pressing the **Jump Mode/Inc** button again
- Pressing the encoder
- Selecting another parameter to edit

NOTE: The two "wait" modes are designed to let you program contours that are clocked from segment to segment by a clock patched to the **Gate/Trig** or **Logic** inputs, independent of the duration of the individual segments. If you used a regular "on Gate Rise" jump mode, if any of the segments were shorter in duration than the clock frequency, they would proceed to the next segment before the next clock pulse arrived, losing sync. With the "wait" modes, short segments will wait at their end until the clock does arrive, ensuring sync is maintained.

ANOTHER NOTE: When using the "wait" modes, it's usually useful to use the **Logic** input to clock your contour, as you can then use the **Gate/Trig** input independently to start and reset it.



Jump Target

The **Jump Target** parameter selects where Control Forge will jump if the test defined by the **Jump Mode** is evaluated to be true.

In **Program Mode**, pressing the **Jump Target** button brings the parameter to the fore on the display. Select an available mode with the encoder.

Available **Jump Targets** are:

- to Segment # (1-8):** Jump to the selected segment in the current preset
- to Stopped:** Stop the preset
- to Seq Dec:** Send the **Preset Sequencer** a **Dec** trigger
- to Seq Inc:** Send the **Preset Sequencer** an **Inc** trigger

to Seq Reset: Send the **Preset Sequencer** a **Reset** trigger

to Preset ### (any existing Preset): Jump to the selected Preset

The display is dismissed by:

Pressing the **Jump Target** button again

Pressing the encoder

Selecting another parameter to edit



Conditional Value

If the selected **Jump Mode** is one of the 16 modes that test a CV (either at the **CV 1** or **CV 2** inputs or via the **CV 3** and **CV 4** knobs) this

parameter sets the level of the CV to be tested.

In **Program Mode**, pressing the **Conditional Value** button brings the parameter to the fore on the display. Adjust the value using the encoder.

The **Conditional Value** range is from -5V to +5V in 10mV increments.

If the current segment does not use a **Conditional Jump** mode that involves a CV test, the value will be ignored and browned out in the **Program Mode** screen.

The display is dismissed by:

Pressing the **Conditional Value** button again

Pressing the encoder

Selecting another parameter to edit

Some Examples of Conditional Jumps

If we tried to document all of the possibilities opened up by **Conditional Jumps**, this already frighteningly long manual would grow to truly epic proportions. Instead, here are just some of the basic structures **Conditional Jumps** make possible.

Cyclical (LFO-style) Contour

Probably the simplest is to program an **Always at End** jump in the last segment of your preset to jump back to Segment 1. This creates a cyclical pattern that will repeat indefinitely (unless it gets a rising edge at the Gate/Trig input, in which case it will immediately restart from **Segment 1**).

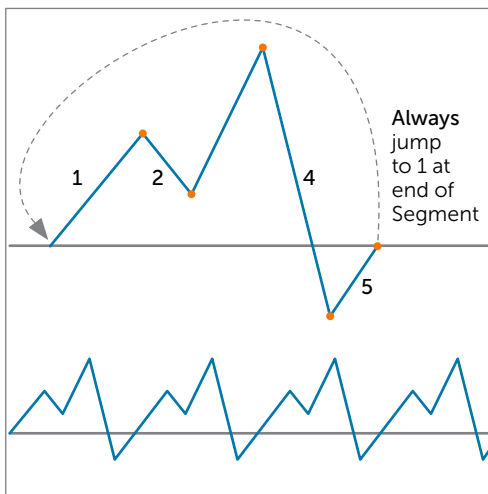


FIGURE 3: CYCLICAL (LFO-STYLE) CONTOUR

Modify Contours Based on CV

In the example below, we have an envelope-like contour with a bit of a "blip" after the initial rise. As long as the signal at the **CV 2** input is greater than 1.75 volts, the contour will proceed through **Segments 2 and 3** to create the blip.

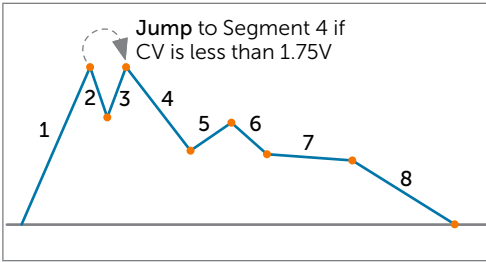


FIGURE 4: MODIFY CONTOURS BASED ON CV

However, by programming in **Segment 1**:

Jump Mode: if@end CV2 <

Conditional Value: 1.75v

Jump Target: to Segment 4

When the contour gets to the end of Segment 1, it checks the CV 2 input and, if the input is below 1.75v, it immediately jumps to Segment 4, eliminating the blip.

Modify Contours Based on Gates

This next example demonstrates how you would program a conventional ADSR Envelope using a combination of **Conditional Jumps** and the **Gate Fall Jump** function in the **Preset Globals** menu.

NOTE: While, for simplicity, this example contains only the standard four segments of an ADSR, with the Control Forge, each of the functional sections could themselves be made up of multiple segments of multiple shapes for much more dynamic modulation contours.

For this example, program in **Segment 3**:

Jump Mode: if@end GateHi

Conditional Value: (not used)

Jump Target: to Segment 3

In the **Preset Globals** menu, set **Gate Rise to Start/Reset** and **Gate Fall Jump** to Segment 4.

With this programming, when Control Forge sees a rising edge at the **Gate/Trig** input, the contour starts at the beginning and (assuming the Gate remains high) continues through **Segments 2 and 3**.

At the end of **Segment 3**, the **Gate/Trig** input is checked to see if the Gate is still high. If it is, the contour immediately jumps back to the beginning of **Segment 3** and runs through it again. It will keep looping through **Segment 3** as long as the gate is high (think of this as the Sustain portion of the envelope).

Because **Gate Fall Jump** is set to Segment 4, if the gate should go low at *any* point during Segments 1, 2, or 3, there will be an immediate jump to the beginning of Segment 4 (essentially the Release portion of the envelope).

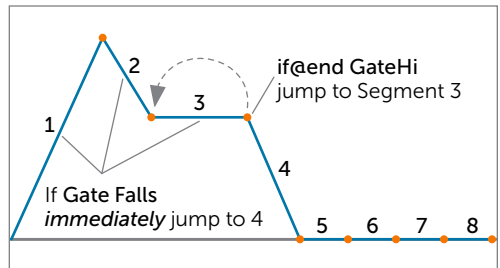


FIGURE 5: MODIFY CONTOURS BASED ON GATES

BONUS IDEA: Instead of making Segment 3 an absolute value, set its **Level Mode** to **Relative (Rel)**, set its **Level** to something small (say 0.10v), and set **Randomness** to +/- 0.25 Gaussian. Now, while the gate remains high during the sustain portion of the envelope, instead of having a static value, the level will subtly vary in a random manner, resulting in a more organic effect.

Jump to Different Presets Based on CVs

While it's simple to program a jump to a different preset based on any **Conditional Jump** evaluation, since each segment can have only one **Conditional Jump** programmed, it takes a bit more programming to set up a decision point that results in the contour jumping to one preset if a certain condition is true and a different one if it isn't. Here's how to do it.

Suppose that we have a preset where we are using the first 6 segments and at the end of Segment 6, we want to jump to **Preset 045** if CV 1 is greater than 3.40v and to **Preset 179** if CV 1 is less than or equal to 3.40v.

For Segment 6, program:

Jump Mode: if@end CV1>
Conditional Value: 3.40v
Jump Target: to Preset 045

For Segment 7, program:

Jump Mode: when CV1<
Conditional Value: 3.41v
Jump Target: to Preset 176

At the end of **Segment 6**, if **CV 1** is greater than 3.40v, then we jump to **Preset 045**.

If it's not greater than 3.40v, we immediately continue on to **Segment 7**, where the immediate evaluation of **CV 1** (which we know will be equal to or less than 3.41v) results in the jump to **Preset 179**.

13. AutoChain




AutoChain is an alternate way to link multiple presets in extended chains. Unlike simply using **Conditional Jumps** to jump from one preset to the next,

or using the **Preset Sequencer** to move from one to the next under trigger control, **AutoChain** was created to handle two very specific circumstances:

- > Use **AutoChain** when you want to create a multi-preset “envelope-like” pattern where a gate rise will cause the chain to reset to the first preset in the chain, regardless of which preset is active when the gate rise arrives, and can optionally contain a “release” section preset that’s jumped to on gate fall, again regardless of which preset is active when the gate fall arrives.
- > Use **AutoChain** when you want to create a multi-preset retriggerable cyclic pattern (like a long, syncable LFO), where, again, a gate rise will cause the pattern to reset to the first preset in the chain, regardless of which preset is active when the gate rise arrives.

Here’s how it works:

1. An **AutoChain** pattern is created in **numerically consecutive preset locations**.
2. Absent any other specific segment programming, each preset in an **AutoChain** will automatically proceed to the next preset without the need to expressly program any jumps.
3. All member presets of an **AutoChain** must be set to one of the member functions (**Start**, **Continue**, or **Release**) in their **Preset Globals** menus.
4. Typically, the first preset in the chain will be set to **Start**. Subsequent presets will be set to **Continue**. If the pattern is to be an envelope-style, there should be a **Release** preset at (or near) the end.
5. The **Gate Rise** function of **every** member preset is set to **Start**. This causes a **Gate Rise** in any of the **AutoChain** members to trigger a reset to **Segment 1** of the **Start** preset.
6. If the pattern is to be an envelope style, the **Gate Fall** function of **every** member preset should be set to **Release**. This causes a **Gate Fall** in any of the **AutoChain** members to trigger a jump to **Segment 1** of the **Release** preset.
 **NOTE:** *There can be additional **Continue** presets after the **Release** preset, so you can have very long, complex releases.*
7. An **AutoChain** continues until it encounters any of the following:
 - > A **Conditional Jump to Stop** in any segment
 - > An empty preset location
 - > A preset that is not set to **AutoChain**
 - > Another preset that is defined as an **AutoChain Start** preset

14. Triggers



Control Forge provides two independent triggers that can fire based on a variety of circumstances (including being triggered manually in **Play Mode** by the **TR 1** and **TR 2** buttons). They can be used to affect other modules in your system or can be routed to Control Forge's own inputs. The triggers are

programmed via the **Assign Triggers** button.

Programming Triggers



In **Program Mode**, press the **Assign Triggers** button to bring up the **Assign Trigger 1** menu on the display. Use the encoder to make a selection.



Available assignments are:

Off: No trigger is automatically generated. Triggers are fired only by the **Play Mode** buttons.

Gate Rise: A **trigger only** is generated when the **Gate** (manual or input) rises.

Gate Fall: A **trigger only** is generated when the **Gate** (manual or input) falls.

Logic Rise: A **trigger only** is generated when the **Logic Input** rises.

Logic Fall: A **trigger only** is generated when the **Logic Input** falls.

On Stop: A **trigger only** is generated when the contour stops.

Begin Any Segment: A **gate or trigger** is generated when any segment begins.

End Any Segment: A **gate or trigger** is generated when any segment ends. A segment that stops the contour creates a **trigger only**.

Any Jump: A **gate or trigger** is generated when any jump to a segment occurs.

Begin Segment #: A **gate or trigger** is generated when the specified segment begins.

End Segment #: A **gate or trigger** is generated when the specified segment ends. A segment that stops the contour creates a **trigger only**.

Jump To Segment #: A **gate or trigger** is generated when a jump to the specified segment occurs.

Jump From Segment #: A **gate or trigger** is generated when a jump from the specified segment occurs.

Once you have selected the **Trigger 1** assignment, press the **Assign Triggers** to bring up the **Trigger 1 Width** menu. Use the encoder to make your choice.





NOTE: Depending on the circumstance that fires them and the setting of the **Trigger Width** parameters, Control Forge can output an actual "trigger" (i.e., a very short pulse) or a gate whose duration is represented as a percentage of the length of the segment in which it appears. Some triggering actions provide no meaningful gate length reference and are considered "trigger only" (as noted in the options above). If a trigger-only assignment is made and its **Width** parameter is set to any of the percentage values, a **1ms Hi** trigger will be generated instead.

Trigger Width options are:

Fast: Trigger is low for 48 microseconds, then goes high for 48 microseconds, then low.

1ms Hi: Trigger is low for 48 microseconds, then goes high for 1 ms, then low.

1ms Hi/Lo: Trigger is low for 1 ms, then goes high for 1 ms, then low.

10ms: Trigger is low for 1 ms, then goes high for 10 ms, then low.

12.5%, 25%, 33.3%, 37.5%, 50%, 62.5%, 66.7%, 75%, 87.5%: When the associated segment begins, the trigger will be low for 48 microseconds, then go high for 48 microseconds and remain high until the segment reaches the specified percentage completion, then go low.

99%: When the associated segment begins, the trigger will be low for 48 microseconds, then go high for 48 microseconds and remain high until the segment reaches its end, then go low.

100%: When the associated segment begins, unless the previous segment had a 100% width trigger, the trigger will be low for 48 microseconds, then go high for 48 microseconds and remain high for the entire duration of the segment. If the next segment calls for a trigger at its start, there will **not** be a low period between the two triggers.

NOTE: A **99% Trigger Width** will remain high for the entire length of its segment and then fall briefly, ensuring that if the next segment has a trigger at its start, there will be a rising edge at the trigger output. With a **100% Trigger Width**, if the next segment has a trigger at its start, the 100% trigger will **not** fall before the start of the next segment, so there will not be a rising edge at the trigger output. This allows you to program gates that span multiple segments. (Think of it as a kind of "legato" mode.)

Once you've programmed the **Trigger 1 Width**, press the **Assign Triggers** button again to bring up the **Assign Trigger 2** menu and continue to program Trigger 2's assignment and width.

15. Real Time Control

When Control Forge is in **Play Mode**, there are a wide variety of ways to influence the progress of a contour, both manually and via signals from other modules in your system (and, in fact, from Control Forge itself). This provides a lot of creative flexibility in live performance situations.

We've covered many of them earlier in this manual, but here are all of the real-time control options in one place:

Gate/Trig Input

If **Gate Rise** is set to **Start/Reset**, a rising edge at this input, (or a manual push of the **Manual Gate** button) will cause the contour to reset to Segment 1.

If **Gate Rise** is set to **Off**, a rising or falling edge or high or low signal at this input (or push of the **Manual Gate** button) can be used as a **Conditional Jump** test in any segment.

Logic Input

A rising or falling edge or high or low signal at this input can be used as a **Conditional Jump** test in any segment.

CV 1 and CV 2 Inputs

The CV levels at these inputs can be tested for a **Conditional Jump** condition in any segment.

If a segment's shape is set to one of the **CV 1** or **CV 2 Passthrough** shapes, the CV levels at these inputs will be passed directly through to the Control Forge's output during that segment.

Time Scale CV Input

The CV at this input will scale the entire contour while preserving the segment's time relationships. This CV is multiplied by the encoder time scale value (see below) to arrive at the final **Time Scale** value.

Time Scale (Data Encoder)

In **Play Mode**, the encoder serves as a real time control over time scale. Its value is multiplied by the value of the **Time Scale CV Input** to arrive at the final **Time Scale** value.

CV 3 (Time Knob) and CV 4 (Level Knob)

Turning these knobs set the **CV 3** and **CV 4** values, which can be tested for a **Conditional Jump** condition in any segment.

If a segment's shape is set to one of the **CV 3** or **CV 4 Passthrough** shapes, the CV levels at these inputs will be passed directly through to the Control Forge's output during that segment.

Quantize Button


Pressing this button while a contour is playing will quantize the contour output to the nearest 1/12th volt for as long as the button is pressed.

If, while pressing this button, you simultaneously press the encoder, the quantize function is locked on and will remain in effect when the button is released. The "Q" indicator in the display will remain lit to indicate this.

Click the **Quantize** button again to unlock it.



NOTE: *If you switch to **Program Mode** while quantization is active, it remains active.*


 **ANOTHER NOTE:** *The quantize function also affects any external CVs that are routed to the output via one of the CV Passthrough shapes.*

Randomize Button

Pressing this button while a contour is playing will randomize the voltage output for as long as the button is pressed. The range of randomization and the randomization distribution is set in the **Preset Globals** menu.

If, while pressing this button, you simultaneously press the encoder, the randomization function is locked on and will remain in effect when the button is released. The “R” indicator in the display will remain lit to indicate this.

Click the **Randomize** button again to unlock it.

 **NOTE:** *If you switch to Program Mode while randomization is active, it remains active.*

Trigger 1 and Trigger 2 Buttons

In **Play Mode**, pressing either of these buttons causes a manual gate to be generated at the associated output. The “1” and/or “2” indicators in the display light to indicate the presence of the gate(s).

Segment Buttons

In **Play Mode**, pressing one of the **Segment Buttons** while Control Forge is stopped causes it to immediately jump to that segment and begin playing from 0 Volts. If the **Segment 1** button is pressed while stopped, this treated the same as a **Manual Gate**, including activating the **Global Start/Reset** parameters.

If a contour is running at the time that a segment button is pressed, it is treated as a **Jump Immediate** from the active segment to the pressed segment.

Sequence Control Inputs (Dec, Inc, and Reset Buttons)

In **Play Mode**, pressing any of these buttons sends the associated trigger to the **Preset Sequencer**, giving you manual real-time control over the progression through the sequence of presets. The “DEC,” “INC,” and “RST” indicators in the display light to indicate the presence of the triggers.

Entering Program Mode While a Preset is Running

It is possible to enter **Program Mode** and modify the preset’s or individual segment’s parameters while the contour continues to play. Parameter changes take affect as they are made.

In particular, changing a segment’s level, time, and/or shape while a contour is playing can provide interesting creative options, both for a control structure and when Control Forge is operating at audio rate as a wavetable audio source.

16. Audio Rate Contours


Since Control Forge will operate well into the audio frequency range, it can be used as a wavetable audio source. Simply patch one of its CV outputs to a filter or other audio path module and listen to the result.


The best way to start experimenting with Control Forge as a wavetable oscillator is to start by programming a variety of shapes to a preset's eight segments, setting their **Time** parameters to a few milliseconds, and setting alternating **Target Levels** of +5V and -5V. Use the encoder in **Play Mode** to vary the frequency.

From that start, try the following:

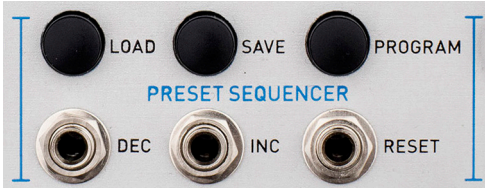
- > Vary the **Time** parameters of individual segments. Since the preset will continue to play when you enter **Program Mode**, you can select individual segments for editing and listen to the effects of changing their **Time** parameter as you change it.
- > Alternating **Target Levels** of +5V and -5V, as suggested above, result in cycles that are two segments long. You can create longer, more complex cycles by setting consecutive **Target Levels** to: +5V, 0V, -5V, and 0V. And, of course, you can create arbitrarily longer cycles by programming even more intermediate **Target Level** values.
- > Use **Conditional Jumps** in your audio rate contour to allow changing the segments (and therefore the timbre of your audio) based on external inputs or by manually adjusting **CV 3** or **CV 4**.
- > Link multiple audio rate presets together using **Preset Chaining**, the **Preset Sequencer**, or **AutoChains** for continually evolving audio.

- > One of our favorite techniques is to program one or more segments in your preset(s) to **CV 1** or **CV 2 Passthrough**. Patch an audio rate signal from one or more other modules in your system into the **CV 1** and/or **CV 2** input(s) and modify those signals as the Control Forge preset plays. In our experience, the results are often dramatic!

 **NOTE:** *While the frequency response to CVs at the **Time Scale CV** input is approximately 1V/oct, it's not always precisely 1V/oct. As such, Control Forge is best seen as a source of rich audio textures as opposed to a precise source of diatonic melody. (On the other hand, it may just be good enough for what you have in mind.)*

 **ANOTHER NOTE:** *An alternate approach that does offer precise 1V/oct response (assuming the oscillator used below is itself properly calibrated), is to set each segment to jump On Gate Rise to the next segment. Then patch the square wave output from a VCO into the Gate/Trig input. The VCO will then control the frequency of Control Forge.*

17. Preset Sequencer



The **Preset Sequencer** allows you to create an ordered list of presets (and optionally, some navigational commands) and then step through them under trigger or manual control.

Sequence Structure

A **Preset Sequence** is made up of a list of steps containing one or more presets, and, optionally, one or more commands that affect the progress of the sequence.

Sequences are numbered from 000 to 200. Sequence 000 is the special case **"Off."** Sequences 001-200 each have a name of up to 12 characters. The default name of a sequence location that has never been saved to is "--empty--".

The progress of the **Preset Sequencer** is controlled by triggers at the **Dec**, **Inc**, and **Reset** input jacks and, in **Play Mode**, the **Dec**, **Inc**, and **Reset** buttons, as described below.

Load Sequence



Loading sequences functions exactly like loading presets as described in **Chapter 7**. Rather than duplicate all that here, refer to that

chapter and, in your mind, replace "Preset" with "Sequence."

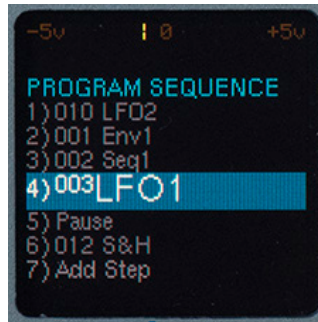


Sequence 000 (**Off**) turns off the sequencer. When Sequence 000 is loaded, triggers at the sequence control jacks and presses of the manual sequence

control buttons are ignored.

When loaded, a sequence (other than Sequencer Off) begins at Step 001 and immediately loads the associated preset for that step.

Program Sequence



Pressing the **Program Sequence** button brings up the **Program Sequence** screen and allows the selection or editing of the presets and commands of

the currently active sequence.

While in the **Program Sequence** screen, the display will show (in addition to the usual CV output indicator and segment progress display) up to seven sequence steps (fewer if the sequence contains less than seven steps) and their current content.

To program or edit a step, use the encoder to scroll to that step (it will be displayed at a larger size than the unselected steps) and press the encoder. The current content of the step will be highlighted in red and turning the encoder will select the desired content or action for that step.

Possible step contents include:

- > Any existing **preset**
- > A navigational **command**:

Pause
Halt
Bumper

These commands will be described below.

In addition, once a step is selected, you can perform the following **Actions**:

Insert Step: A new step containing a **Pause** is created and inserted at the current step number. It becomes the current step and is selected for editing.

Delete Step: The current step of the sequence is deleted, and you are returned to the **Program Sequence** screen.

Pressing the encoder while an action is highlighted executes that action. Pressing the encoder while a command or preset is highlighted replaces the current step contents with that command or preset.

Adding Steps to a Sequence

To add one or more steps to your sequence, scroll to end of the step list until **Add a Step** is highlighted. Press the encoder to add a step to the end of the sequence. Repeat as needed.

Sequence Commands

The Sequence Commands give you some additional ways to control the flow of your sequences. Here's how they work:

When a new sequence is loaded, it begins at the **Step 001**. Subsequently, any **Reset** rising edge (or button press) acts to jump to **Step 001** and load the preset or command

associated with it, and any **Inc** or **Dec** rising edge will cause the sequence to move (circularly) to the next or previous step. If that step is a preset, it will be immediately loaded; if that step is a **command**, it will take effect per the below descriptions:

Halt: When a **Halt Command** is loaded, the sequencer enters the **Sequence Halt** state. In this state, subsequent **Inc** or **Dec** rising edges (or button presses) are ignored. Only a **Reset** rising edge (or the loading of a new sequence) will return the Sequencer to the normal state. A **Reset** rising edge also acts to jump to **Step 001**.

Pause: When a **Pause Command** is loaded, the sequencer enters the **Sequence Pause** state. In this state, subsequent **Inc** or **Dec** rising edges (or button presses) are ignored until receipt of a **Reset** rising edge, which will return it to the normal state. At that point, the **Pause** has been satisfied, and subsequent **Inc** or **Dec** signals act normally, and any **Reset** rising edge acts to jump to **Step 001**.

Bumper: When the sequencer is adjacent to a step containing a **Bumper Command**, the sequencer remains in a normal state, but ignores any **Inc** or **Dec** signals that would advance it to the **Bumper** step. Essentially, the sequence sticks at the step adjacent to the **Bumper** step until an **Inc** or **Dec** rising edge moves it away from the **Bumper** step, or until a **Reset** rising edge returns it to **Step 001**.



NOTE: *Since sequences can progress circularly, it's possible to encounter commands from either side.*

Reset, Dec, and Inc Inputs

If the Sequencer is **Off**, these inputs (and equivalent button presses) are ignored. If the Sequencer is on (i.e., any sequence except **#000** is loaded), each of these inputs is logically ORed with its associated Play Mode button and responds to a rising edge as follows:

Reset: If the Sequencer is in the **normal** or **Halt** state, it immediately jumps to **Step 001** of the current sequence, loading the associated preset or command. The rules for running the loaded preset are followed.

If the sequence is in the **Pause** state, the sequencer will exit that state and return to the normal state, taking no other action until it receives another **Dec, Inc** or **Reset**.

Inc: If the sequencer is in the normal state, it immediately attempts to advance to the next higher (circular) step. If that step is a **Bumper**, it does not advance, but stays at the current step. Otherwise, it succeeds in advancing to the next step. If that step is a **Halt** or **Pause** Command, it enters the corresponding state. If that step is a valid preset, that preset is immediately loaded, and the rules for running the loaded preset are followed. If that step is an **Empty Preset***, the step is jumped and it attempts to advance to yet another step iteratively.

If the sequence is in the **Pause** or **Halt** state, the sequencer will take no action.

Dec: If the sequencer is in the normal state, it immediately attempts to decrement to the next lower (circular) step. If that step is a **Bumper**, it does not decrement, but stays at the current step. Otherwise, it succeeds in decrementing to the previous step. If that step is a **Halt** or **Pause** Command, it enters the corresponding state. If that step is a valid preset, that preset is immediately loaded, and the rules for running the loaded preset are followed. If that step is an **Empty Preset***, the step is jumped and it attempts to decrement to yet another step iteratively.

If the Sequence is in the **Pause** or **Halt** state, the sequencer will take no action.

* While it is not possible to program an empty preset into a sequence, it would be possible to program an existing preset into a sequence and subsequently erase that

preset. In that case, the sequence step the erased preset previously occupied would be named "--empty--". If you find yourself in that situation, it'd probably be wise to delete that step.

Other Stuff to Know about Sequences

- > In addition to being off because you've selected **Sequence #000**, the sequencer is also off when you are in **Program Mode** (so it won't be changing presets on you while you're trying to program one) and while a preset is being saved. If the sequencer is Off due to **Program Mode** or **Save Preset**, once normal **Play Mode** is re-entered, the sequencer will resume its previous state. If you've selected **Sequencer Off**, it can be turned on only by selecting another sequence using **Load Sequence**.
- > Logically, all sequences loop at their end. Any step can be any entry (i.e., any **preset** or **command**) with the recognition that some entries or combinations make little functional sense or can even create completely dysfunctional sequences. Specifically, placing a **Halt** at **Step 001** will result in a sequence that can never progress beyond Step 001.
- > Special case: A **Bumper** placed at **Step 001** acts as a (satisfied) **Pause** (i.e., an **Inc** or **Dec** will move to adjacent steps).

Save Sequence



Again, saving Sequences (including naming them) functions exactly like saving Presets as described in **Chapter 7**.

Sequences vs. Preset Chaining

As mentioned back in the **Chapter 11**, if you're creating an extended, multi-preset contour or sequence that's designed to play sequentially from beginning trend, it's probably easiest to program that by simply using a **Jump Always at End** in the last segment of each preset in your sequence, and set its **Jump Target** to the next preset in the sequence. Just keep on repeating the process until the entire sequence is programmed.

But if you have structured your contour or sequence out of separate preset sections that you want to use in multiple places in your sequence, you can use the **Preset Sequencer** to sequence them in whatever order you need. An advantage of this method is that you can use the same section in multiple locations in your performance without having to duplicate their programming.

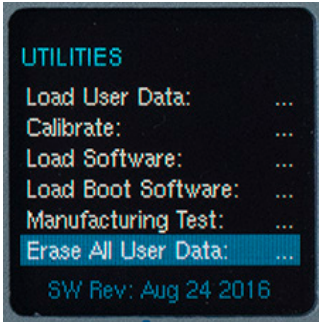
Depending on how you want to play back your contour, you can:

- > Program one of Control Forge's triggers to fire at the end of the last segment of each preset and patch that trigger to the sequencer's **Inc** input to automatically move to the next preset.
- > Patch a trigger or gate from another module in your system to control the progress of the presets.
- > Move from preset to preset manually using the **Dec**, **Inc**, and **Reset** buttons

18. Utilities



The **Utilities** menu is where you'll find Control Forge's various housekeeping and maintenance functions.



To access the **Utilities** menu, press and hold the **Global/Pause/Utilities** button while in **Play Mode**. Turn the encoder to scroll through the available function list. Click the

encoder to choose the desired function.

The menu is dismissed by pressing the **Global/Pause/Utilities** button again.

Here are the available **Utility** functions:

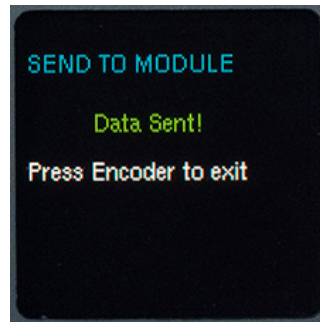
Send to Module



This function lets you send all of the presets and sequences in your Control Forge to a Satellite module or another Control Forge.

To send the data:

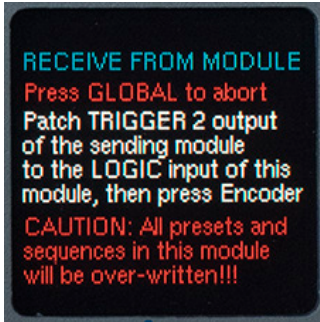
1. Connect the **Trigger 2** output from your Control Forge to the **Logic** input of the receiving module
2. Select **Utilities > Send to Module** on your Control Forge.
3. Select **Utilities > Receive From Module** on the receiving module, then click the receiving module's encoder.
4. Click the encoder on your Control Forge. The display will show the progress of the transfer.
5. If an error is detected on the receiving module during reception or on completion of reception, an error message will be displayed there and you



can press the sending Control Forge's encoder to try again or press the **Global/Pause/Utilities** button to exit.

Once transfer is complete, click the encoder to return to the **Utility** menu.

Receive from Module



This function lets you load the presets and sequences from a Satellite or another Control Forge into your Control Forge.

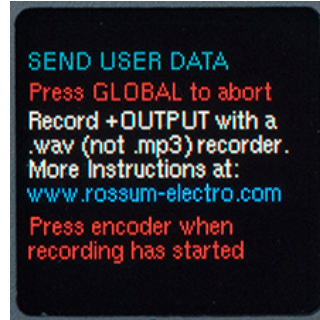
IMPORTANT!: This function will overwrite (i.e., delete) all of the presets and sequences in your module. **Be careful.**

To receive the data:

1. Connect the **Trigger 2** output from the Satellite or Control Forge sending the data to the **Logic** input of your Control Forge.
2. Select **Utilities > Receive from Module** on your Control Forge and click the encoder. The display will indicate that it is waiting for data.
3. Select **Utilities > Send to Module** on the module sending the data, then click that module's encoder to send the data.
4. When the transfer begins, a progress bar on your Control Forge's display will show its progress.
5. If an error is detected during reception or on completion of reception, an error message will be displayed and you can click the encoder to try again or press the **Global/Pause/Utilities** button to exit.

Once transfer is complete, click the encoder to return to the **Utility** menu.

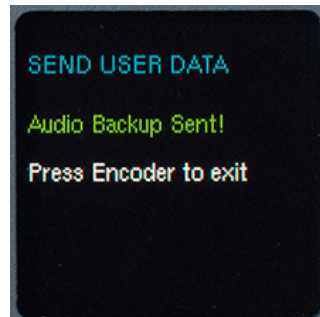
Save User Data



This function lets you save all your presets and sequences to an audio .wav file for backup or transfer purposes.

To save your data to a .wav file:

1. Connect the Control Forge's **+ Output** to the input of your recording device.
- NOTE:** *The recording device must record uncompressed WAV data. Do not use an MP3 recorder.*
2. Select **Utilities > Save User Data** on your Control Forge.
3. Upon selecting Save User Data, Control Forge will begin sending a test tone for the purpose of setting your record level. Using this tone, set the record level on your recording device to -6dBFS.




4. Start recording on your recording device.
5. Click the encoder on your Control Forge to send the data.

Once transfer is complete, stop recording on your recording device and click the encoder to return to the **Utility** menu.

Verify User Data

This function lets you play a previously recorded (using the **Save User Data** function) backup into your Control Forge to ensure that it is intact and being played back at sufficient level.

This does **not** actually load the data into your Control Forge. It's purpose is to confirm that the backup file is valid, so that you can avoid loading an invalid file and deleting your existing data.

 **NOTE:** *If you are accessing the .wav file from the internet (as opposed to having recorded it yourself), perhaps because you are loading presets that someone else developed and shared online, see the **Hints for Playing Back WAV Files** section below for advice on successful playback.*

To verify your saved data:

1. Connect the output from your playback device to your Control Forge's **CV 1** input. We've found that a regular Eurorack patch cord works fine. (See the **Hints for Playing Back WAV Files** section below for hints for successfully playing the files into your Control Forge.)
2. Select **Utilities > Verify User Data** on your Control Forge.
3. Click the encoder as instructed.
4. Start playback of the .wav file. You should see a green percentage progress message in the display.
5. If an error is detected during reception or on completion of reception, an error message will be displayed and you can try again (try playing back at a hotter level) or press the **Global/Pause/Utilities** button to exit.
6. If you don't receive an error, you can proceed to load the data into your Control Forge using the **Load User Data** function described below.

Once the process is complete, click the encoder to return to the **Utility** menu.

Merge User Data

This allows you to read in a data file (presets and sequences) and copy any preset or sequence in that file to your module, as long as its number is not already taken up with a saved preset or sequence already in your module.

So, if you have saved presets 003 and 006, and you load in a file that has presets 001-009, you'll end up with the following:

001 and 002 from the merge file

003: your existing preset

004 and 005 from the merge file

006: your existing preset

007, 008, and 009 from the merge file


 **NOTE:** *003 and 006 from the merge file will not be transferred.*

Sequences work the same, with the exception that it would be possible to exceed the 20,000 step limit. In that case, any sequences that exceeded the limit would not be transferred.

Load User Data



This function lets you play a previously recorded (using the **Save User Data** function) backup into your Control Forge to load that data into your module's memory.

 **IMPORTANT!** *This function will also overwrite (i.e., delete) all of the presets and sequences in your module. So again, be careful. We highly recommend using the **Verify User Data** function described above to confirm that your file is good before actually loading it.*

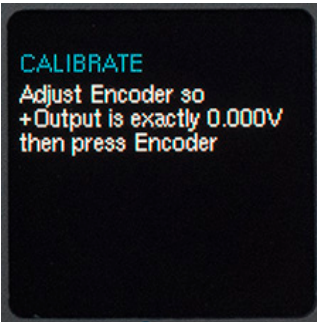
NOTE: If you are accessing the .wav file from the internet (as opposed to having recorded it yourself), perhaps because you are loading presets that someone else developed and shared online, see the **Hints for Playing Back WAV Files** section below for advice on successful playback.

To load your saved data:

1. Connect the output from your playback device to your Control Forge's **CV 1** input. We've found that a regular Eurorack patch cord works fine. (See the **Hints for Playing Back WAV Files** section below for hints for successfully playing the files into your Control Forge.)
2. Select **Utilities > Load User Data** on your Control Forge.
3. Click the encoder as instructed.
4. Start playback of the .wav file. You should see a green percentage progress message in the display.
5. If an error is detected during reception or on completion of reception, an error message will be displayed and you can try again or press the **Global/Pause/Utilities** button to exit.

Once transfer is complete, click the encoder to return to the **Utility** menu.

Calibrate



Your Control Forge is calibrated at the factory to precisely 1V/oct. So, in normal circumstances, you shouldn't have a need to recalibrate it. However

should abnormal circumstances present themselves, this function allows you to recalibrate your Control Forge.

NOTE: This calibration procedure assumes you have access to an accurate voltmeter. The result will only be as accurate as your meter.

Select **Utilities > Calibrate** on your Control Forge. The procedure is self-guiding. Simply follow the instructions on each screen.

The first 3 steps require you to measure the voltage at the **+ Output** with your meter and then adjust it if necessary. The remaining steps use that output to calibrate the CV inputs.

When the encoder is pressed after the last step, the screen will briefly display **"New Calibration Data Saved"** then return to normal operation.

Pressing any other button during the calibration process will briefly display **"Calibration Aborted"** then return to normal operation.

NOTE: If you don't actually have a need to recalibrate, it's best to leave this function alone. Playing around with it could completely mess up your calibration. You don't want that to happen.

Load Software



This function lets you load updated operating software into your Control Forge. The process is pretty much identical to the **Load User Data** process described above.

1. Connect the output from your playback device to your Control Forge's **CV 1** input.
2. Select **Utilities > Load Software** on your Control Forge.

3. Click the encoder as instructed.



4. Start playback of the .wav file. You should see a green percentage progress message in the display. Again, see the **Hints for Playing Back WAV Files** section below if you're having any problems.
5. If an error is detected during reception or on completion of reception, an error message will be displayed and you can try again or press the **Global/Pause/Utilities** button to exit.

Once transfer is complete, click the encoder to reboot your Control Forge and use your new software.

Load Boot Software



This function loads an updated version of the special software that's responsible for booting up your Control Forge.

The process is identical to the **Load Software** function above, except, of course, that you select **Utilities > Load Boot Software** on your Control Forge and load an appropriate Boot Software file.

Manufacturing Test

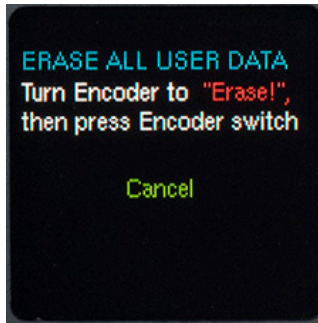


This function provides a suite of tests to be used during manufacturing to ensure that everything on your Control Forge is working correctly.

Unless something goes horribly wrong, there shouldn't be a reason that you ever need to worry about it.

However, like **Calibrate**, it's self-guiding, so if you're just curious, you're welcome to play with it, since, unlike **Calibrate**, it can't do any harm.

Erase All User Data



This does exactly what it says, i.e., erases **all of the presets and sequences** in your Control Forge. There's no undo, so be sure you **really** want to do this before proceeding.

To erase your data:

1. Select **Utilities > Erase All User Data**.
2. Initially, the word **"Cancel"** is displayed in green. Turn the encoder until it changes to **"Erase!"** in red.
3. With **Erase!** displayed, press the encoder to proceed.

4. If you change your mind before completing Step 3, either turn the encoder until **Cancel** is displayed again and press the encoder, or press the **Global/Pause/Utilities button** to abort the process.

> If you do get an error message during a transfer (e.g., the level was too low, your computer generated extraneous audio, etc.), you can just correct the problem and try again. However, we've found that rather than just playing the file again from the beginning, it's usually advisable to close the file, reopen it, and then start playback.

Hints for Playing Back WAV Files

The basic requirements for successful WAV file playback for software updates and data transfer is that the file should be played back at the **highest possible level without distortion**. As long as it doesn't distort, the hotter the better.

We've had good luck playing WAV files directly from a computer (but see the warnings below) as well as smartphones and tablets.

In theory, any device that will play a WAV file without distorting the data should work. However, here are some caveats:

- > If you are streaming the file directly from the web, be sure that your data rate is sufficient to keep up. If necessary, wait until the file has been completely buffered before starting playback.
- > If you are playing the file from a computer or phone, quit Facebook, your email client, and any other software that produces audio. Any other audio injected into the stream will cause the transfer to fail.
- > If you're using a smartphone, temporarily placing it in Airplane Mode will prevent any incoming call or message notifications from interfering with playback.
- > We have found that **Safari** on the **Mac** plays the file in a way that often keeps it from being successfully recognized and decoded. If you are using **Safari**, we recommend downloading the WAV file to your computer and playing the file from the local copy using **Quicktime Player**.
- > In fact, if you are having **any** problem streaming the file, downloading the file locally and playing it from there is a good alternative.

19. Preset Templates

The following presets are designed to be starting points for a variety of common types of contours. Their Global parameters have been set up, conditional jumps, if used, have been programmed, and stand-in values have been provided for Time, Level, and Transition Shape. We'd expect you to change those parameters to suit your specific needs.

To use the templates, load the one you want and then **immediately resave it to a different location** and rename it as desired. That way the original template is still available unchanged for when you want to use it again.

In most cases, you can check out the templates with the basic patch described below. If a different patch is required for a particular template, it will be described in that preset's listing.

The basic patch

- > Patch the **+Output** to the CV in of a VCO and/or VCF and/or VCA or other processor. (It's easiest to clearly hear what's happening with a preset by listening to its effect on a VCO.)
- > Patch the **Trigger 1** output to the gate input of an envelope or other triggered CV source.
- > Patch the output from a gate or trigger source into the **Gate/Trig** input.
- > Patch a CV source into the **Time Scale CV** input (preferably through an attenuator).

300 One Shot

This is a simple 8-segment one shot contour. A rising edge at the Gate/Trig input (or press of the Manual Gate button) starts it and resets it to the beginning if it's already running. It automatically stops at the end of Segment 8.

Trigger 1 fires at the beginning of Segment 1

Trigger 2 fires at the beginning of every segment.

Customize the template by selecting new levels, times, shapes and time scale.

301 1 Shot Pass

This is the same contour as 301 above, except Segment 3 has been set to CV 1 Passthrough and Segment 6 has been set to CV 2 Passthrough. Patch external signals into CV 1 and CV 2 inputs and they will be passed through to the output during their respective segments.

Trigger 1 fires at the beginning of Segment 1

Trigger 2 fires at the beginning of every segment.

Customize the template by selecting new levels, times, shapes and time scale.

302 Cond LFO

This template provides a basic sine LFO in the first four segments. As long as CV 3 is below 2V, Segment 4 will continuously loop back to Segment 1.

When CV 3 is greater than 2V at the end of Segment 4, the contour will continue on to the remainder of the segments and then loop back to Segment 1. If CV 3 falls back below 2V, the contour will return to looping segments 1-4.

In Play Mode, the encoder sets the tempo of the LFO.

At Gate rise will instantly reset the contour to Segment 1.

Trigger 1 fires at the beginning of Segment 1.

Customize the template by selecting new levels, times, and shapes. Change the conditional test from CV 3 to CV 1 or CV 2 to have an external signal control the conditional jump.

303 DAHDSR

This is a template for a 6 stage envelope: Delay, Attack, Hold, Decay, Sustain, Release.

Default values (which you will be changing for your own use) for the segments are:

Segment 1 (Delay): 0V, 1 Second, DC Delay

Segment 2 (Attack): 5V, 1 Second, Expo 5

Segment 3 (Hold): 5V, 1 Second, DC Delay

Segment 4 (Decay): 2V, 1 Second, Circ1.4

Segment 5 (Sustain1): +0.012Vrel, +/-0.09. Rand gaussian, 5 Second, Expo 3

Segment 6 (Sustain2): -0.012Vrel, +/-0.09. Rand gaussian, 5 Second, Expo 3, Jump to 5 if@endGateHi

Segment 7 (Release1): +0.579V, 1 Second, Expo 5

Segment 8 (Release2): 0V, 1 Second, Expo 3

The way this works is that there is an initial delay defined by the time of Segment 1 (set it to 0 if you don't want a delay).

Segment 2 defines Attack time and level.

Segment 3 is an optional Hold time.

Segment 4 defines Decay time and level.

Segment 5 and 6 loop for as long as the Gate is high. The relative Level values and slight amount of randomness means the sustain portion has a bit of variation, as opposed to the static sustains of most envelopes.

When the Gate falls, the envelope jumps to Segment 7. It and Segment 8 define the release portion of the envelope.

304 Shape Keys

This is one of our favorites for real-time fun. The template defines a "keyboard" made up of the 8 Segment Buttons. Simply press a key to "play" its shape. Each segment stops at the end of its shape.

The template is arranged as 4 pairs of shapes. In each pair, the first segment rises to 5V and the next segment is the same shape dropping back down to 0V. The best result comes from alternating rising and falling shapes.

Trigger 1 fires at the start of every segment.

Customize the template by selecting new shapes, levels, times, and trigger width.

305 Audio Rate

This template implements an audio signal generator. Patch the + Output directly into your audio path (a filter would be good) and adjust the encoder in Play Mode to change the pitch.

Change shapes, levels and times to modify the timbres. Set a few segments to CV 1 or CV 2 Passthrough and patch in some other audio sources that will be passed through doing their segments.

306, 307 and 308 AutoChain

These three presets use AutoChain to implement a long "envelope-style" contour. Refer to the AutoChain explanation in **Chapter 13** for details.

306 AC Start: This preset is the AutoChain "Start."

307 AC Continue: This preset is set to loop as long as the Gate is high.

308 AC Release: When the Gate falls, this preset is immediately jumped to.

In all three presets, Trigger 1 fires at the beginning of Segment 1.

Trigger 2 fires at the beginning of every segment.

Customize these by changing pretty much everything.

Presets 310–315

The following presets were created for us by composer and technology writer, **Jim Aikin**.

310 Seq Starter

A template for creating a melodic sequence.

Has an AbsQ value in Segment 1 and RelQ values in 2 through 8, so as to play a melodic pattern.

Change the Levels, Time, and Shapes as desired.

Trigger 1 fires at the start of every segment.

311 Xpose Seq

A template for creating a melodic sequence that can be transposed by an external CV.

It's identical to 310, except that the value in Segment 1 comes directly from the CV 1 input. By plugging a keyboard-type source (such as Make Noise Pressure Points) into this input, you can transpose the entire sequence while it's running.

Alternatively, change the CV 1 input to CV 3 and control the transposition with the CV 3 knob.

Trigger 1 fires at the start of every segment.

312 XposSeqLim

Is identical to 311 except that it uses Kurt Kurasaki's neat trick of controlling the length of the pattern from CV 3. In play mode, this knob will give you from 2 Segments up to all 8. Play with this at a high tempo in real time for intricate rhythmic and tonal variation.

Trigger 1 fires at the start of every segment.

313 4 Tri LFOs

This preset provides triangle LFOs in Segments 1-2, 3-4, 5-6, and 7-8, each twice as fast as the one before it. By tapping the buttons in Play mode, you can double or halve the speed of the LFO on the fly.

Try changing the curve shapes for a wide variety of modulation effects.

314 LoopEnv

This is a basic ADSR envelope template. Segments 1 and 2 are the attack and decay, Segments 3 through 6 loop for as long as the Gate/Trig input is high. Segments 7 and 8 are the release. Customize by changing the Times, Levels, and Shapes.

315 LoopEnvExt

This preset is a more complex envelope that loops for a variable number of times depending on the level of an external CV source patched into CV 1.

The preset sends a trigger from Segment 2, which can be used to trigger your choice of external signal sources, such as a Maths. Patch the Maths output back into CV 1. You can now send Control Forge a short gate to start the envelope, and the envelope will loop until the Maths's decay level falls to zero, at which point the release segments will play. If the length of the Maths contour is being voltage-controlled from another module, successive Control Forge envelopes can have different numbers of loops.

20. Demo Presets

The following presets were provided by some of our talented beta testers to highlight some of Control Forge's capabilities. In most cases, you can check them out with the basic patch described below. If a different patch is required for a particular preset, it will be described in that preset's listing.

The basic patch

- > Patch the **+Output** to the CV in of a VCO and/or VCF and/or VCA or other processor. (It's easiest to clearly hear what's happening with a preset by listening to its effect on a VCO.)
- > Patch the **Trigger 1** output to the gate input of an envelope or other triggered CV source.
- > Patch the output from a gate or trigger source into the **Gate/Trig** input.
- > Patch a CV source into the **Time Scale CV** input (preferably through an attenuator).

Presets 400–446

Presets 400 to 446 were created by composer and sound designer **Kurt Kurasaki**. With a few exceptions, they are programmed to use the CV 3 and CV 4 knobs to modify the presets' behavior in one way or another without having to patch up external CVs. Once you understand what they are capable of, you can, of course, replace the CV 3 and CV 4 conditional tests with CV 1 and/or CV 2 from other modules in your system.

In many of Kurt's presets, adjusting **CV 4** controls the **number of segments** in the pattern. Doing this while the preset is running can result in intricate rhythmic and tonal variation. Be sure to try it. It's clever and addicting.

For quick access to the various preset categories, you can load **Preset Sequences 1-4** that each contain the presets of their own category.

Presets 400–409: Sequences

In each preset, **Trigger 1** fires at the beginning of every segment. **Trigger 2** fires at the beginning of Segment 1.

400 [Sequencer]

(Group marker of Sequencer Presets)

Sequence of volt/oct values that shift relative to segment 1

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

401 CV3 5ths

Sequence of volt/oct values that shift relative to segment 1

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

402 CV3 oct

Sequence of volt/oct values that shift by octaves relative to segment 1.

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

403 CV3 Slides

Sequence of volt/oct values that shift relative to seg 1, 3, 7.

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

404 CV1 5ths

Sequence of volt/oct values that shift by octaves and fifths relative to segment 1.

CV1 Controls the Quantized Value of Seg 1. (Requires CV 1 input)

CV4 Controls the Number of Segments in the Pattern.

405 CV3 Swing

Sequence of volt/oct values that shift relative to segment 1.

Segment times have various durations to create a shuffle.

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

406 Reverse

CV3 Controls a base quantized pitch value of the sequence

CV4 Controls the **Direction of Playback**

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

407 GateHiForward

Sequence that plays in reverse from normal operation

Gate/Trig In or **Manual Gate** Button switches direction.

CV3 Controls a base quantized pitch value of the sequence

CV4 Controls the Number of Segments in the Pattern.

408 Gate Step

A fixed Note Sequencer preset.

Gate/Trig or **Manual Gate** advances the sequence to the next segment.

CV3 Controls the Quantized Value of Seg 1.

CV4 Controls the Number of Segments in the Pattern.

409 BUCHLA CV

Fixed Note Sequencer Preset with 1.2 volt/octave values that drives Buchla 200/200e oscillators.

CV4 Controls the Number of Segments in the Pattern.

Presets 410–419:

Random CV contours

In each preset, **Trigger 1** fires at the beginning of every segment. **Trigger 2** fires at the beginning of Segment 1.

410 [Random]

(Group marker of Random CV Contours)

CV3 Sets Segment 2 voltage

CV4 Sets Segment 3 voltage

411 Rand 5th

Segment 1 is a Random Quantized volt/oct value.

Consecutive segments shift the value by 5th and Octaves.

CV4 Controls the Number of Segments in the Pattern.

412 Rand Ascend

Segment 1 is a Random Quantized volt/oct value.

Consecutive segments shift the voltage up by 1 semitone.

CV4 Controls the Number of Segments in the Pattern.

413 Rand Curves

Segment 1 and 6 are Random Quantized volt/oct values.

CV3 Sets Segment 5 voltage.

CV4 Controls the Number of Segments in the Pattern.

414 CV3&Random

Odd Segments are Quantized volt/oct values set by CV3.

Even Segments are Random Quantized volt/oct values.

CV4 Controls the Number of Segments in the Pattern.

415 CV1 S/H Q

Sample and Hold Generator that outputs Quantized volt/oct values.

CV1 is the signal (noise) source input.

Any type of signal can be sampled on CV1.

416 CV1 S/H

Sample and Hold Generator (not quantized)
CV1 is the signal (noise) source input.
Any type of signal can be sampled on CV1.

417 S/H+Slide

Sample and Hold with Randomized segment
2 target voltage.

Creates a sliding voltage curve.

CV1 is the signal (noise) source input.

418 Rand 5th+7

Segment 1 is a Random Quantized volt/oct
value.

Consecutive segments shift the value by 5ths
and Octaves.

CV4 Controls the Number of Segments in
the Pattern.

419 Rand Ratchet

Random Voltages with exponential time
changes on each segment.

CV4 Controls the Number of Segments in
the Pattern.

Presets 420–429

Filter Pattern Contours

Kurt designed them to drive an oscillating
Rossum Electro-Music Evolution filter, but
you can try them with anything.

In each preset, **Trigger 1** fires at the beginning
of every segment.

420 [Filter Pts] (Group marker of Filter Pattern Contours)

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

421 Filter Rn 1

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

422 Filter Rn 2

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

423 Filter Rn 3

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

424 Filter Rn 4

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

425 Filter Rn 5

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

426 Filter Rn 6

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

427 Filter Rn 7

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern..

428 Filter Rn 8

CV3 sets various segment voltages.

CV4 Controls the Number of Segments in
the Pattern.

429 Filter F/B

CV3 sets various segment voltages.

CV4 Controls the **Playback Direction** of the
segments.

Trigger 2 Pulses at the beginning of segment
1.

Presets 430–445

(gltch rtm 1–gltch rtm 15) are used in a Preset Sequence

These presets are used by **Preset Sequence 004**: “—gltch rtm—.” According to Kurt, this sequence creates high-paced CVs that can be used to modulate envelopes and other modulation destinations for IDM style tracks.

The presets are programmed to generate internal **Preset Sequencer Inc** triggers, so no need to patch physical triggers into the trigger inputs. Just load **Preset Sequence 004** and go.

In all presets, **Trigger 1** fires at the beginning of every segment. **Trigger 2** fires at the beginning of **Segment 1**.

Presets 450–462

Presets 450 to 462 are melodic sequences created by musician and composer **David Phipps**. The presets use a variation of the basic patch as follows:

- > Patch the **+Output** to the CV in of a VCO.
- > Patch a clock (gate or trigger source) into the Control Forge’s **Logic Input**.
- > Patch the **Trigger 1** output to the gate input of an envelope or other triggered CV source.
- > **Trigger 2** will pass through the clock from the Logic Input.

450 Phi-Env x

A dynamic one-shot contour. Trigger with a Gate Rise. (Does not use the Logic Input.)

451 Phi-Env ~

A another dynamic one-shot contour. Trigger with a Gate Rise. (Does not use the Logic Input.)

452 Phi-Cycle 3d

An ascending melodic pattern. Trigger and retrigger with Gate Rises. The clock at the Logic Input sets the tempo. If the Gate is sustained, the pattern will repeat and continue rising until it’s out of hearing range (if you’re not a dog).

453 Phi-MajrrAN

An arpeggiating quantized major pattern. Trigger and retrigger with Gate Rises. The clock at the Logic Input sets the tempo. Unlike 452, the pattern will automatically repeat without the need for additional Gate Rises.

454 Phi-Minorl

An arpeggiating quantized minor pattern. Trigger and retrigger with Gate Rises. The clock at the Logic Input sets the tempo. Hold down the Manual Gate for a short rhythmic pattern using Segments 1, 2, and 8. Release the gate for the whole pattern. You can “play” the Manual Gate button for rhythmic and melodic variation.

455 Phi-MinorA

Another arpeggiating quantized minor pattern. Trigger and retrigger with Gate Rises. The clock at the Logic Input sets the tempo.

456 Phi-MajorW

An ascending melodic pattern with the Cir1.8R shape on segment one for an interesting “slide.” Trigger and retrigger with Gate Rises. The clock at the Logic Input sets the tempo.

457 Phi-1

A dynamic (and glitchy) repeating contour. Trigger with a Gate Rise. (Does not use the Logic Input.)



NOTE: *We discovered (after releasing version 1.0 software) that there is an error in this preset. It’ll be fixed in the next version, but in the meantime, you can fix it yourself by changing the Jump Target of Segment 8 to Segment 1 and resaving the preset.*

458 Phi-Chase prtA

A melodic patten that uses a conditional jump in Segment 8 to link to **Phi-Chase PrtB** below.

459 Phi-Chase PrtB

The continuation of the melodic patten in **Phi-Chase prtA**. It uses a conditional jump in Segment 8 to link back to **Phi-Chase PrtA**. The two presets then repeating in alternation.

460 Phi-Ascend

An ascending melodic pattern that uses Relative Target Levels to continue to rise with each repetition until reaching the upper voltage limit and then repeating at that level. Trigger with a Gate Rise. The clock at the Logic Input sets the tempo. Another Gate Rise will reset the pattern to its low beginning pitch.

461 Phi-GateAdvMjr

A major scale that can be manually advanced by repeatedly pressing the Manual Gate button or can be played like a keyboard using the Segment Buttons.

462 Phi-Env w

A dynamic one-shot contour. Trigger with a Gate Rise. (Does not use the Logic Input.)

Presets 470–481

Presets 470 to 482 were created by composer and sound designer **Richard Devine**. In addition to the basic patch, add a clock (gate or trigger source) into the Control Forge's **Gate Input**.

470 Rhyt-2

A Gate Rise starts a one-shot contour with random rhythmic variation. A second Gate Rise during Segment 1 results in a jump to Segment 5. A Gate Rise during any other segment retriggers the contour from the start. Works best with a random or slow evolving CV envelope into the Time Scale input.

471 Rhyt-3

A variation based on Rhyt-2, but with different transition shapes. Works best with random or slow envelopes/LFOs into the Time Scale input.

472 Rhyt-4

Another variation based on Rhyt-2, but with the CV 3 Passthrough shape on Segment 5 and the CV 4 Passthrough shape on Segment 8. Slow the Time Scale down and twiddle those knobs!

473 Rhyt-5

Another variation based on Rhyt-2, but with the CV 4 Passthrough shape on Segment 5, this preset uses more Chaos Transition waves, making it better to use on Envelopes or Filter. Use the a fast rate CV source (LFO/Random/Envelope) into the Time Scale input to get more glitchy results.

474 Rhyt-6

Based on patch Rhyt-5, this preset uses a different combination of Chaos transitions shapes, making it better to use for Envelopes, VCA's, CV effects, or a Filter cutoff. Use a CV Random CV source that changes at different speeds or rates to get more interesting variations.

475 Octo-1

A Gate Rise starts a one-shot contour which plays Segment 1 and then waits for a second Gate Rise to trigger an ascending sequence of pitches. Fun for a simple fast melody.

476 Octo-2

IGate Rise triggers a sequence of pitches in an alternating sequence. Try something fun into the Time Scale CV input, to create more organic repetitions and pulses, offset Sine waves, and LFO's.

477 Octo-3

Gate Rises and Falls during various segments trigger conditional jumps. Picking a clock frequency that is offset to the basic rhythm of the contour will result in more variation. Try something fun into the Time Scale CV input, to create more organic repetitions and pulses, offset Sine waves, and LFO's.

478 S&H-1

Trigger with a gate input, at medium to high rates for the best effect. Uses the randomization function to generate different DC Delay variations. Press the Quantization button while running to temporarily quantize the output to semitones.

479 S&H-2

Variation of S&H-1, Trigger with a gate input, at medium to high rates for the best effect. Try using both positive and negative outputs at the same time to get two different variations.

480 S&H-3

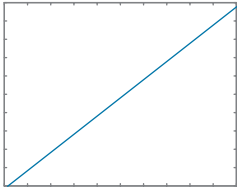
Variation of S&H-2 using the FastLn3 shape for most of the segments for a very different sonic effect.

481 Quanta-1

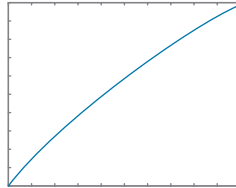
Best used as a modulation source, if CV 3 is greater than -5V (i.e., not turned all the way counterclockwise), the contour will not move past Segment 4. When CV 3 is above -5V, the contour will proceed to successive segments. The interaction between the clock frequency into the Gate input, the position of CV 3, and the time scale of the contour will produce a variety of effects.

21. Transition Shape Plots

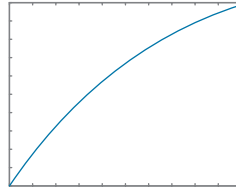
The following are plots of the **Transition Shapes** available in Control Forge.



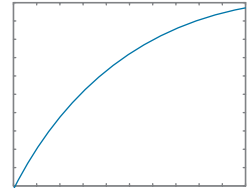
1: LINEAR



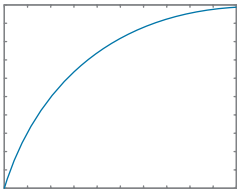
2: EXPONENTIAL 1



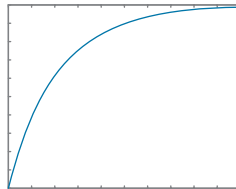
3: EXPONENTIAL 2



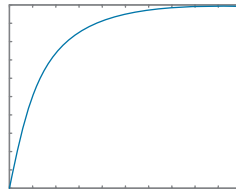
4: EXPONENTIAL 3



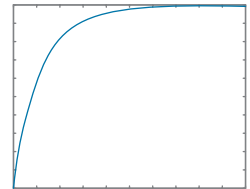
5: EXPONENTIAL 4



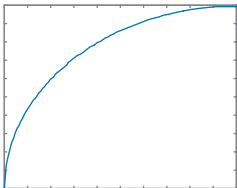
6: EXPONENTIAL 5



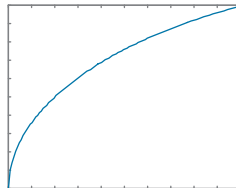
7: EXPONENTIAL 6



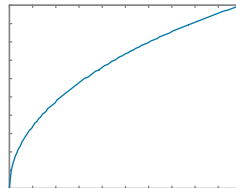
8: EXPONENTIAL 7



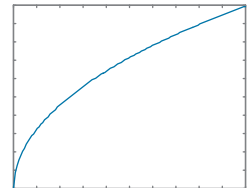
9: CIRCLE 1.4



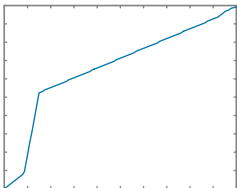
10: CIRCLE 1.6



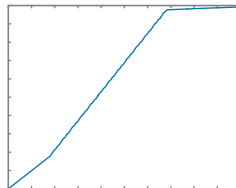
11: CIRCLE 1.8



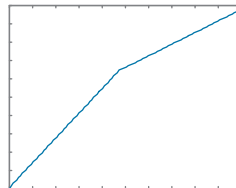
12: CIRCLE 1.16



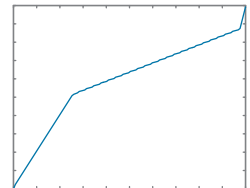
13: SQUEEZE



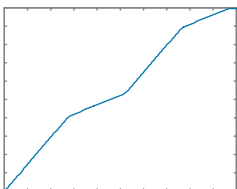
14: FAST LINE 1



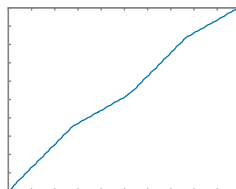
15: FAST LINE 2



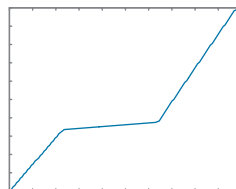
16: FAST LINE 3



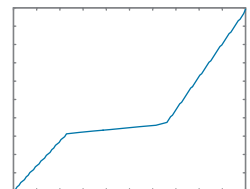
17: MEDIUM LINE 1



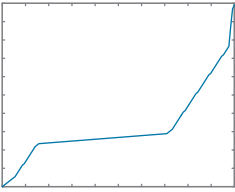
18: MEDIUM LINE 2



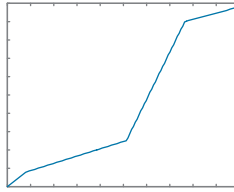
19: SLOW RAMP 1



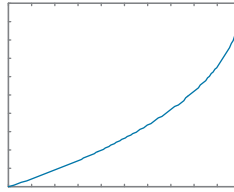
20: SLOW RAMP 2



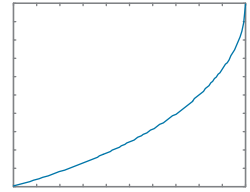
21: BLOOM



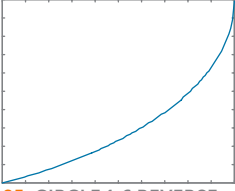
22: BLOOM 2



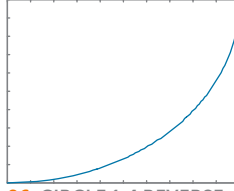
23: CIRCLE 1.16 REVERSE



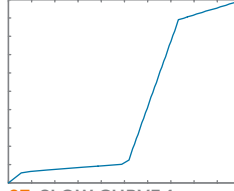
24: CIRCLE 1.8 REVERSE



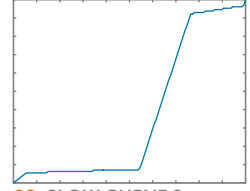
25: CIRCLE 1.6 REVERSE



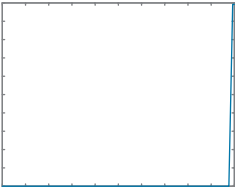
26: CIRCLE 1.4 REVERSE



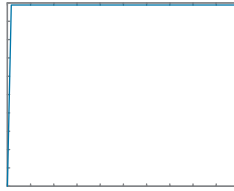
27: SLOW CURVE 1



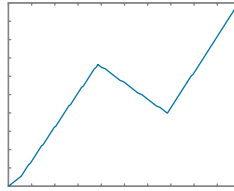
28: SLOW CURVE 2



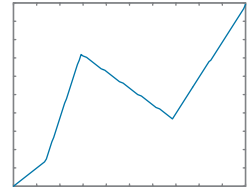
29: DELAY DC



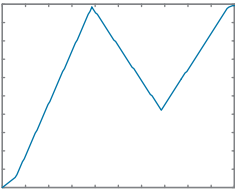
30: DC DELAY



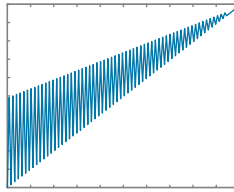
31: CURVE 2X



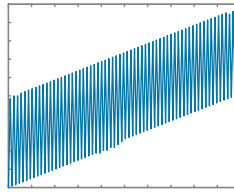
32: CURVE 2X B



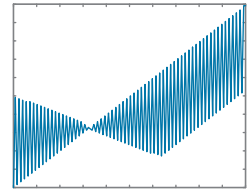
33: CURVE 2X C



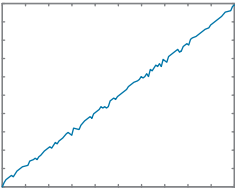
34: ZIG ZAG 1



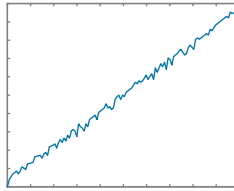
35: ZIG ZAG 2



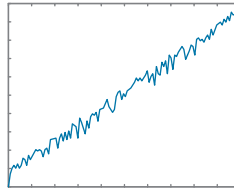
36: ZIG ZAG 3



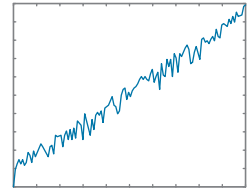
37: CHAOS 03



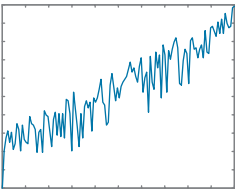
38: CHAOS 06



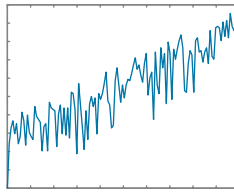
39: CHAOS 12



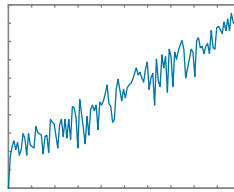
40: CHAOS 16



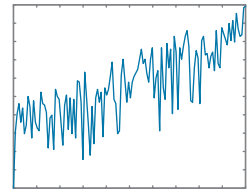
41: CHAOS 25



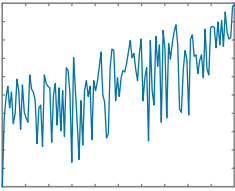
42: CHAOS 33



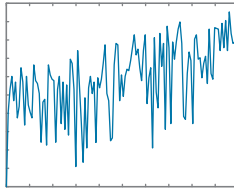
43: CHAOS 37



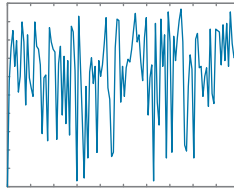
44: CHAOS 50



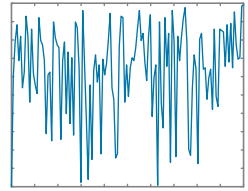
45: CHAOS 66



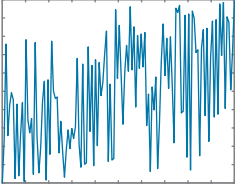
46: CHAOS 75



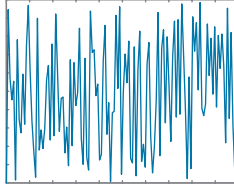
47: CHAOS 95



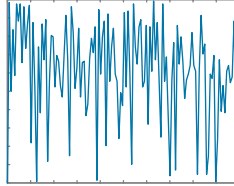
48: CHAOS 99



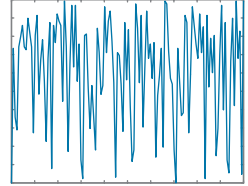
49: LINEAR SHUFFLE



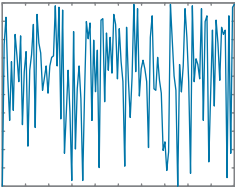
50: LINEAR SHUFFLE 2



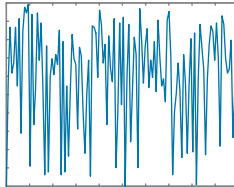
51: RANDOM A



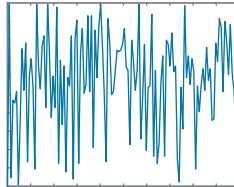
52: RANDOM B



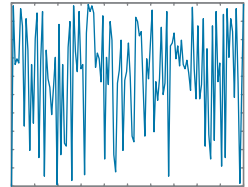
53: RANDOM C



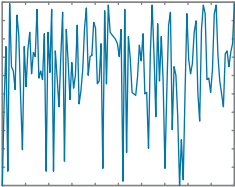
54: RANDOM D



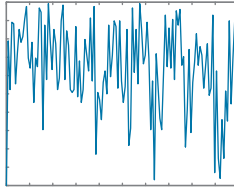
55: RANDOM E



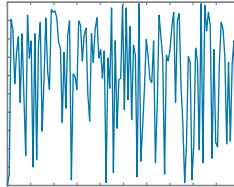
56: RANDOM F



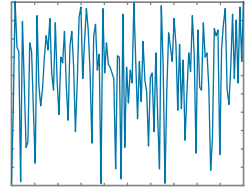
57: RANDOM G



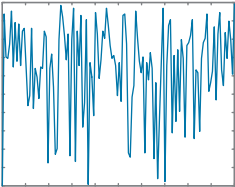
58: RANDOM H



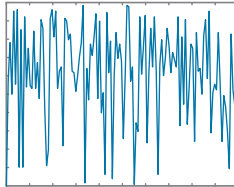
59: RANDOM I



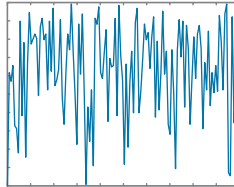
60: RANDOM J



61: RANDOM K



62: RANDOM L



63: RANDOM Z

22. Specifications

PRESETS

500

PRESET SEQUENCES

200

SEQUENCE STEPS

20,000 Dynamically Allocated

INPUTS

GATE/TRIG

1x 3.5mm mono socket
100k Ω Input Impedance
1.6V threshold

LOGIC

1x 3.5mm mono socket
100k Ω Input Impedance
1.6V threshold

CV 1

1x 3.5mm mono socket
100k Ω Input Impedance
-5V to +5V

CV 2

1x 3.5mm mono socket
100k Ω Input Impedance
-5V to +5V

TIME SCALE CV

1x 3.5mm mono socket
100k Ω Input Impedance
-5V to +5V

PRESET SEQUENCER DEC

1x 3.5mm mono socket
100k Ω Input Impedance
1.6V threshold

PRESET SEQUENCER INC

1x 3.5mm mono socket
100k Ω Input Impedance
1.6V threshold

PRESET SEQUENCER RESET

1x 3.5mm mono socket
100k Ω Input Impedance
1.6V threshold

OUTPUTS

+ OUTPUT

1x 3.5mm mono socket
<1 Ω Output Impedance
-5V to +5V

- (INVERTED) OUTPUT

1x 3.5mm mono socket
<1 Ω Output Impedance
-5V to +5V

TRIGGER 1

1x 3.5mm mono socket
1k Ω Output Impedance
10V
Selectable Width

TRIGGER 2

1x 3.5mm mono socket
1k Ω Output Impedance
10V
Selectable Width

POWER REQUIREMENTS

+/-12V via 16-pin, Doepfer-style connector

CURRENT DRAW

125mA +12V, 25mA -12V (maximum)

DIMENSIONS

22HP (W); Panel to power connector (with connector plugged in) 25mm (D)

SUPPLIED ACCESSORIES

1x 16-pin, Doepfer-style cable
4x M3 screws
4x M2.5 screws
4x Nylon washers
1x Quickstart Guide

23. From Dave's Lab: Circuit Protection

Eurorack suffers from the problem of power connector reversal. When 10 pin connectors are used, mis-insertion results in a swap of +12V and -12 V, and protection is easily accomplished using various techniques such as series diodes.

But more systems are providing the +5V supply and thus use the full 16 pin connector. When this is reversed, a diode-protected module is still safe, but the six connected ground pins in the module will short together the system's +5V and +12V supplies, potentially damaging the power supply and any modules that use +5V.

To prevent this, Rossum Electro-Music modules deviate from the standard Eurorack power connector by leaving power connector pins 9 and 10 open, rather than connecting them to ground. When plugged in backwards, this leaves the system +12V supply disconnected. Since ground is still supplied by four pins as well the chassis and any patch cords connected to the module, the dropping of these two pins has no measurable effect on circuit performance, but it means that if a Rossum Electro module is accidentally plugged in backwards, no stress is placed on the +5V supply or modules that use it.

24. Acknowledgements

A number of wonderful people generously provided help, advice, encouragement, and inspiration during the development of Control Forge.

Many thanks from the Rossum Electro-Music team to:

Jim Aikin

Bob Bliss

Patrick Brede

Janis Chafin

Richard Devine

Nancy Enge

Mihai Ionescu

Kurt Kurasaki

William Mathewson

Gur Milstein

David Phipps

Bill Putnum

Allan Shaffer

Kirk Southwell

Tyler Thompson

Tony Wride

Andy Zenczak (and the crew at Gadgetbox Studios)

And, it goes without saying, but we'll say it anyway, our families for understanding all the late nights and weekends spent not having fun (or doing chores) with them.

