# TC-Data

User Guide 1.0

# TC-Data

Version 1.0

User Guide

May 31, 2014

Bit Shape Software, LLC

# Contents

## Getting Started

Getting Started 3

Introduction....3

TC-Data Overview....3

MIDI Quick Start....4

OSC Quick Start....4

Interface 6

Navigation....6

1.0 Navigation Bar....6

Views....7

2.0 Performance....7

3.0 Load / Share....11

4.0 Patch....15

5.0 Settings....19

Performance 23

Performance Basics....23

Voice Allocation....23

6.0 OSC and MIDI Voice Allocation....23

Device Motion....24

iPad Rotation....25

Performance Tips........................................................................25

## Patch Editing 27

Controllers Explained ..............................................................27

7.0 Voiced vs. Global Controllers............................................................27

8.0 Controller Types ...........................................................................28

Output Programming ..................................................................33

9.0 Outputs.........................................................................................33

10.0 Special Output Features...............................................................37

Module Programming.................................................................40

11.0 AHDSR .......................................................................................40

12.0 LFO.............................................................................................42

13.0 Table ...........................................................................................44

14.0 Sequencer ...................................................................................48

## Appendices 52

Troubleshooting..........................................................................52

14.0 Issues / Solutions .........................................................................52

# Getting Started

## Introduction

Thank you for purchasing **TC-Data**. Your support is greatly appreciated, and it means I can continue developing innovative music apps for iOS and Mac OS X. **TC-Data** is the product of countless hours of programming and research into controller design.

Your comments, feedback, and news are important to me. If you use **TC-Data** in a performance or recording, please let me know through the website www.bitshapesoftware.com or email feedback@bitshapesoftware.com. Your creative output is one of the reasons making software instruments is rewarding!

## TC-Data Overview

**TC-Data** is a programmable MIDI and OSC (Open Sound Control) controller the iPad, driven by multitouch and device motion controllers. TC-Data makes no sound; it is a controller interface for any instrument or software that can receive MIDI or OSC.

**TC-Data** does not have on-screen objects to control during performance. Instead, the performer's touches generate the control information. Some controllers are driven by a single touch. For example, the speed of the touch as it moves around the screen is its *Touch Speed*. Other controllers are generated with multiple touches, such as *Distance to Previous Touch*, where the positions of two touches are compared. Finally, the iPad's device motion capabilities can be used as controllers. The accelerometer, gyroscope and compass can be assigned as controllers to turn your iPad into a expressive motion controller.

## MIDI Quick Start

To connect **TC-Data** to another synth app on your iPad, open the target synth app and check for the following two settings:

1. Background Audio Mode
2. MIDI Input Mode

Be sure both of these options are on and available. If a synth app does not support external MIDI input or background audio mode, it will not receive any MIDI data from **TC-Data**.

Open **TC-Data**, and select a patch from the Quick Load Button popover within the 'MIDI' group (swipe the list once to the right). By default **TC-Data** broadcasts MIDI messages to any available target synth on iOS.

More setup options are available, including creating a virtual MIDI port within **TC-Data**, or passing incoming MIDI through to **TC-Data's** targets. See [5.1.1 General] for more information about these Settings.

## OSC Quick Start

To connect **TC-Data** to another device via OSC (Open Sound Control), run through the following steps:

1. Make sure both the iPad running **TC-Data** and the target device (computer) are on the same Wi-Fi network.
2. Be sure to double check IP address numbers after entering (more below)
3. Be sure to double check that the port number is the same for both the target and **TC-Data** (more below)

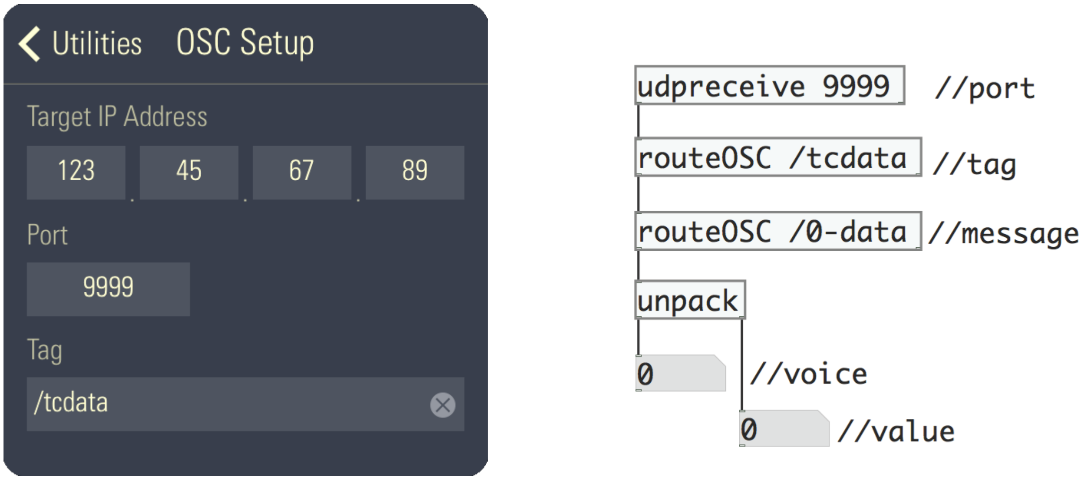
Find the IP address of the target device. Be sure it is the IP address of the Wi-Fi connection, and not another network connection (e.g., Ethernet).

Open TC-Data, and open an OSC patch from the Quick Load Button popover, and the 'OSC' group (swipe twice to the right).

Switch to the Patch view. Open the Utilities ☰ popover in the upper left corner and choose OSC Setup. Enter the IP address, and choose a port that will match your target software's expected port.

Open your target software, and configure the incoming port and message tags to receive data from TC-Data.

The messages sent from **TC-Data** are always a list of two float values: {voice value}. The voice value corresponds to a touch's voice allocation, and should be used to route the data to the proper polyphonic destination. See [6.0 OSC and MIDI Voice Allocation] for more information.



Target IP Address: 123 . 45 . 67 . 89

Port: 9999

Tag: /tcdata

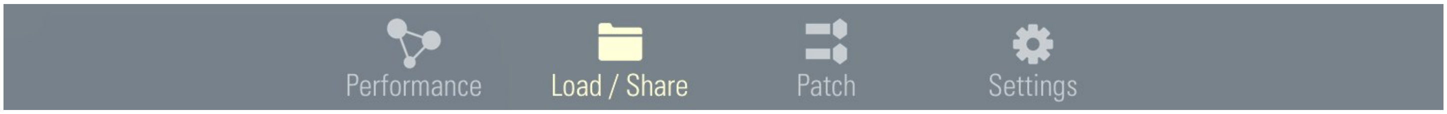
```
udpreceive 9999        //port
routeOSC /tcdata       //tag
routeOSC /0-data       //message
unpack
0        //voice
    0        //value
```

An example OSC setup between TC-Data and Pd (Pure Data).
Note the matching port number, and message routing.

# Interface

## Navigation

### 1.0 Navigation Bar



Show the Navigation Bar by double-tapping the Show Navigation Bar Button in the Performance view. Here you can switch between views. The Navigation Bar will hide automatically when a touch is played in the Performance view.

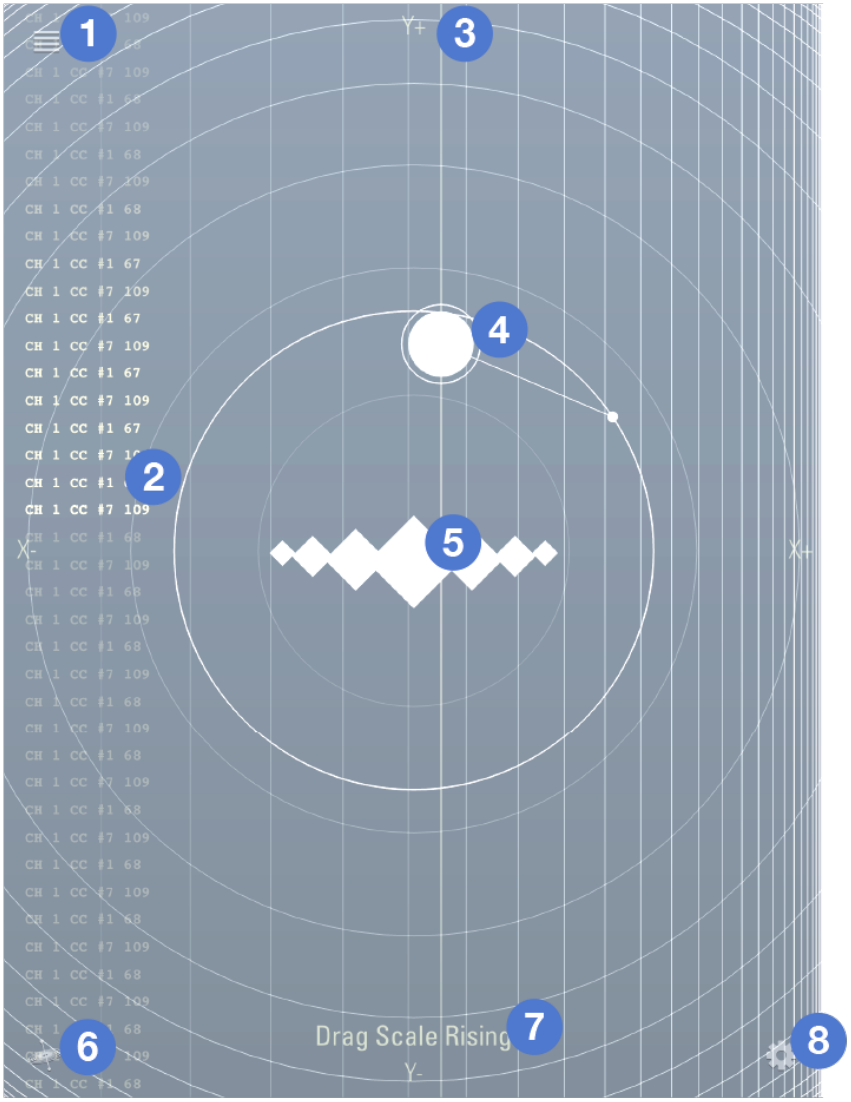
While the Navigation Bar is showing, TC-Data can rotate its orientation. See [2.3 Axis Labels] and [iPad Rotation] for more information.

### 1.1 TC-Data Views

The Performance view is where you can play the currently loaded patch. The Load / Share view is where you can load, save, duplicate, rename and share patches. In the Patch view, you can create outputs and modules to design your patch. The Settings view allows you to change the display style and colors, and has a quick help reference area.

# Views

## 2.0 Performance



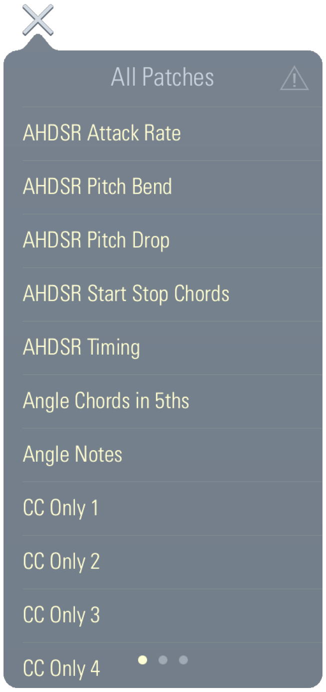
```
CH 1          109
CH            68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 67
CH 1 CC #7 109
CH 1 CC #1 67
CH 1 CC #7 109
CH 1 CC #1 67
CH 1 CC #7 109
CH 1 CC #1 67
CH 1 CC #7 10
CH 1 CC #1
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1 CC #1 68
CH 1 CC #7 109
CH 1       68
           109
CH 1 CC #1 68
```

1. Quick Load Button
2. Message List
3. Axis Labels
4. Performance Area
5. Device Motion Position
6. Gyroscope Reorientation Button
7. Patch Name
8. Navigation Bar Button

The **Performance** view is the edge-to-edge multitouch interface for generating touch control data. It will display the active controllers as you play, as well as an optional message list which shows all outgoing control messages.

## 2.1 Quick Load Button

Double-tap the **Quick Load Button** to pop open a list of patches to load. Swipe left and right to change between folders of patches. There is an ⚠ **All Notes Off** button for quickly silencing stuck notes when in the Performance view.



All Patches

AHDSR Attack Rate
AHDSR Pitch Bend
AHDSR Pitch Drop
AHDSR Start Stop Chords
AHDSR Timing
Angle Chords in 5ths
Angle Notes
CC Only 1
CC Only 2
CC Only 3
CC Only 4

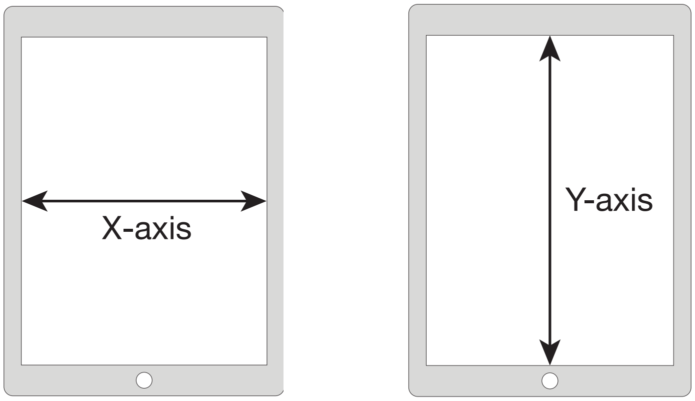
## 2.2 Message List

The **Message List** displays a scroll of MIDI and OSC messages generated during performance. It is useful for seeing live values when in the Performance view. Turning off the list can improve graphics performance and touch responsiveness.

## 2.3 Axis Labels

The Performance view is unique in that is does not rotate when the iPad is reoriented. This is because moving the device can be part of the performance (for motion control), and having the screen rotate beneath your fingers is an undesired effect.

You can reorient **TC-Data** whenever the Navigation Bar is visible to switch between portrait and landscape modes. Many controllers depend on X or Y-axis positions. To keep X / Y controllers consistent, those axes are always fixed as follows:



The X-axis is always the short axis, and the Y-axis is always the long axis, regardless of the interface orientation. You can invert the minimum and maximum of these axes in **Settings >> General >> Invert Axes**. The **Axis Labels** will remind you which way is up (labeled +) and which way is down (labeled -) for each axis.

## 2.4 Performance Area

The entire screen is used as the performance area. Many controllers require multiple touches to activate. As you touch the screen, the graphic display will update to show you which types of controllers are active, and what their positions or values are. If you wish to customize which interface objects appear in the view, choose **Settings >> Display >> Performance View**.

## 2.5 Gyroscope Reorientation Button

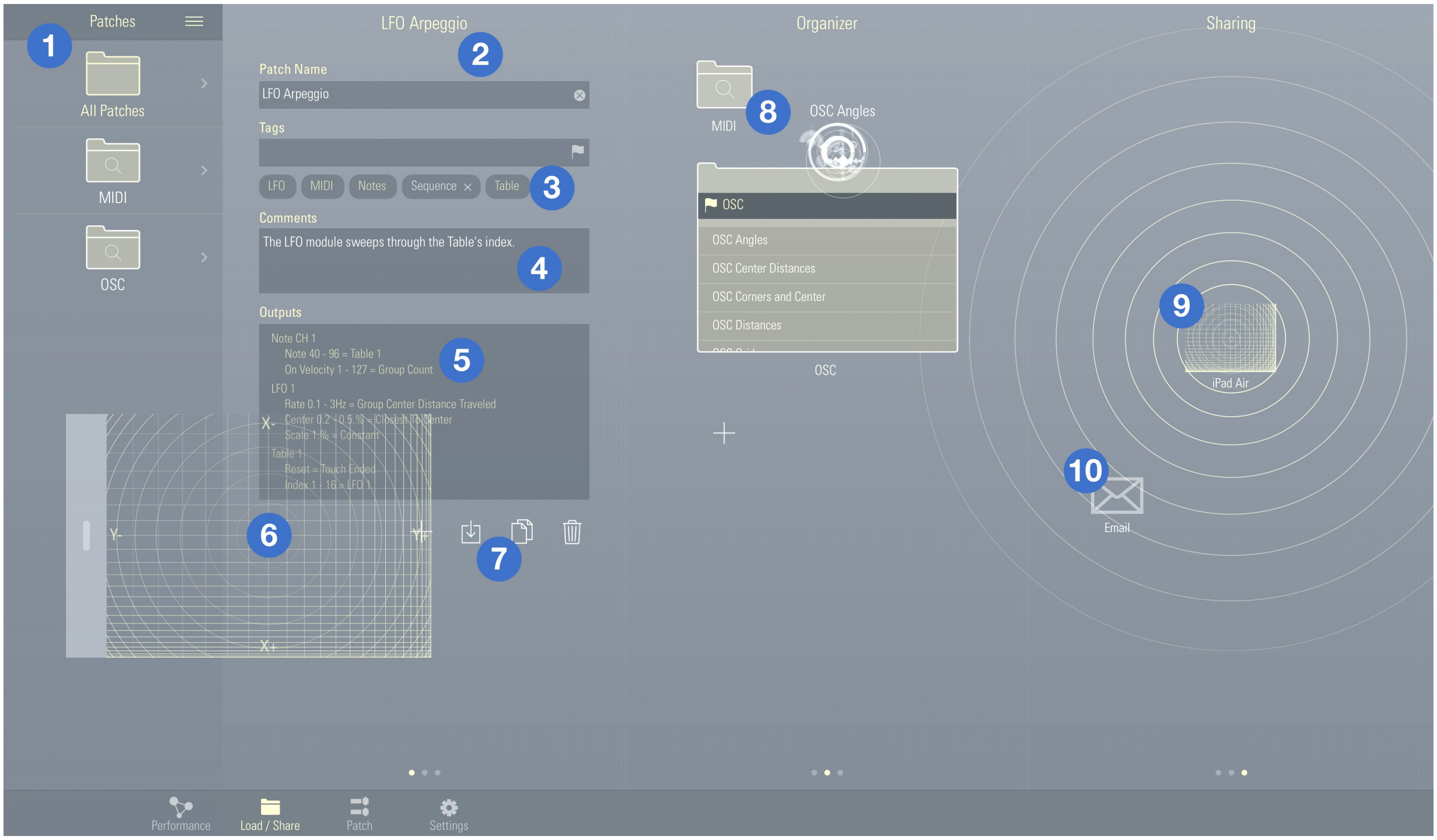
When the gyroscope is in use, you can reset the ‘natural’ position of the device by double-tapping this button. The pitch, roll, and yaw positions will all be zeroed. You can reset the orientation any time during performance by double-tapping the **Gyroscope Reorientation Button**.

## 2.6 Show Navigation Bar Button

Double-tap this button to show the **TC-Data** Navigation Bar.

• • •

## 3.0 Load / Share



Patch Name
LFO Arpeggio

Tags
LFO | MIDI | Notes | Sequence ✕ | Table

Comments
The LFO module sweeps through the Table's index.

Outputs
```
Note CH 1
    Note 40 - 96 = Table 1
    On Velocity 1 - 127 = Group Count
LFO 1
    Rate 0.1 - 3Hz = Group Center Distance Traveled
    Center 0.2 - 0.5 % = Closest To Center
    Scale 1 % = Constant
Table 1
    Reset = Touch Ended
    Index 1 - 16 = LFO 1
```

OSC
- OSC Angles
- OSC Center Distances
- OSC Corners and Center
- OSC Distances

1. Folders and Patches
2. Patch Information
3. Patch Tags
4. Comments
5. Outputs and Controllers List
6. Live Preview View
7. File Management Buttons
8. Patch Organizer
9. AirDrop Sharing
10. Email Sharing

The **Load / Share** view is your portal to loading, saving, duplicating, importing and exporting **TC-Data** patches. It is also a place to organize and share your patches.

When working on your own patches, **TC-Data** will keep a working copy saved in a temporary location in case you leave the app. However, it is a good idea to save your work permanently by visiting this view.

## 3.1 Folders and Patches

This area shows the patches installed on your device, organized by folder. The actual patch files can be found in the iTunes App File Sharing section of your iPad.

The **All Patches** folder is always available, and shows every patch installed. The other folders are Tag Searches and Groups and can be created or deleted to suit your needs [see 3.8 Patch Organizer].

When scrolled to either the **Organizer** or **Sharing** areas, tap and drag (or swipe right) a patch or folder to drag that patch into the view for organization or sharing.

## 3.2 Patch Information

This area shows the currently loaded patch information. The title, tags, and comments for a patch are set here.

## 3.3 Patch Tags

Add descriptive tags to patches for easy searching and organization later. Choose from a list of common tags by pressing the ⚑ button, or type in your own. Some tags are managed by **TC-Data**, and cannot be added or removed by the user.

## 3.4 Comments

Add comments to patches to explain particular setups or useful performance techniques.

## 3.5 Outputs and Controllers List

See a quick overview of all outputs and controllers in the loaded patch, plus the value ranges they are sending.

## 3.6 Live Preview View

The **Live Preview** view allows the currently loaded patch to be played when not in the Performance view. It can be dragged and thrown around the screen, and docks itself automatically on the right side of the screen.

## 3.7 File Management Buttons

+

Create a **New Patch**. You can choose from a list of standard configurations depending on the kind of patch you wish to create.

⤓

**Save** currently loaded patch. If a patch is not saved before loading another patch, it will lose its changes.

⧉

**Duplicate** currently loaded patch. Pressing this button creates a copy of the current patch, and loads the copy.

🗑

**Delete** the currently loaded patch. This operation is not undoable.

## 3.8 Patch Organizer

Create and edit patch organizing folders in this area. Make a **Tag Search** folder to filter all patches by tag, and a **Group** folder to collect any patches together into a folder. Drag patches from the **Folders and Patches** sidebar to add the patch to a Group, or add the tags in a Tag Search to that patch.

Some tags are managed by **TC-Data**. For example, the ‘MIDI’ tag is added automatically if a patch contains any output that sends MIDI. The user cannot add or remove the ‘MIDI’ tag manually. For this reason, dragging a patch into a Tag Search that only contains managed tags is not allowed.

> *Note:*
>
> *You can tell the difference between managed tags and normal tags by the color of the tag text. White tags are editable by the user, tinted tags are managed by TC-Data.*

## 3.9 AirDrop Sharing

If another device is on the same Wi-Fi or Bluetooth network, **TC-Data** will recognize the device and the two will appear on each other’s screen in the **AirDrop Sharing** area. Drag a patch or folder onto the floating screen to transfer the patches to the other device.

You can disable automatic searching by pressing your screen display in the middle of the AirDrop Sharing area.

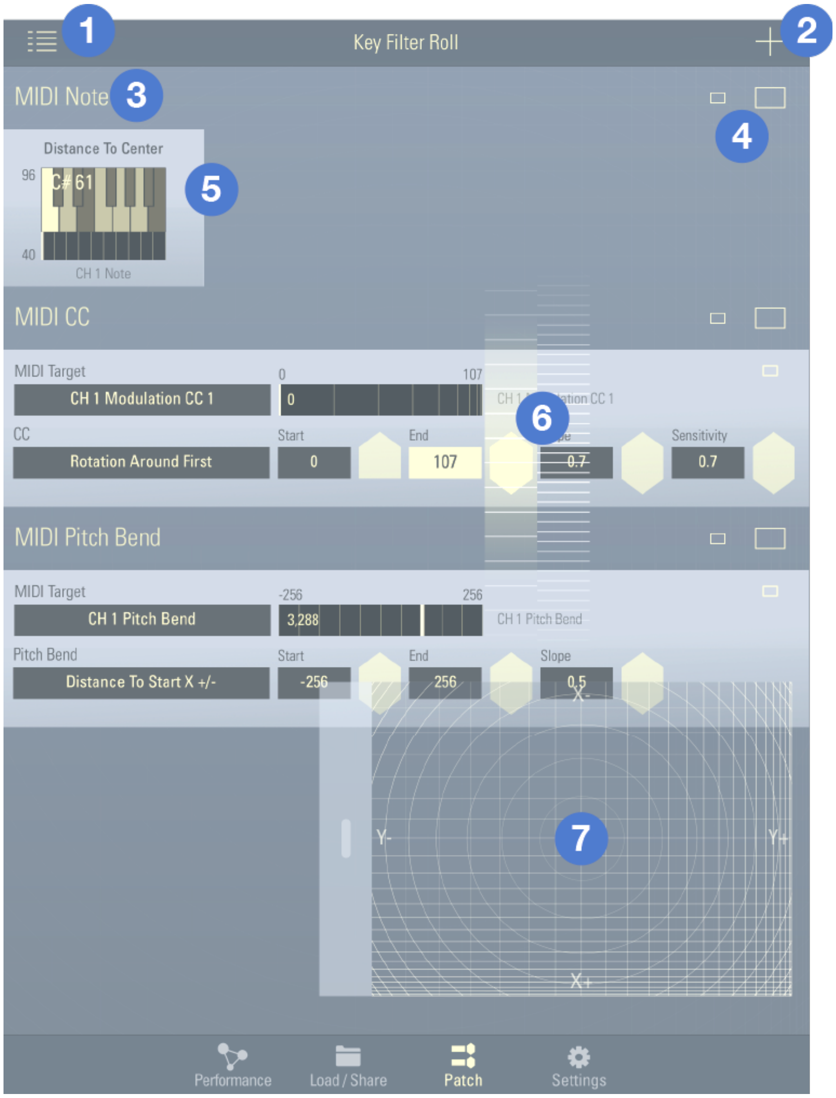
If the devices do not appear but are nearby, toggle automatic searching off / on for both devices.

## 3.10 Email Sharing

Drag a patch or folder onto the **Email Sharing** floating icon to email a patch or folder.

• • •

## 4.0 Patch

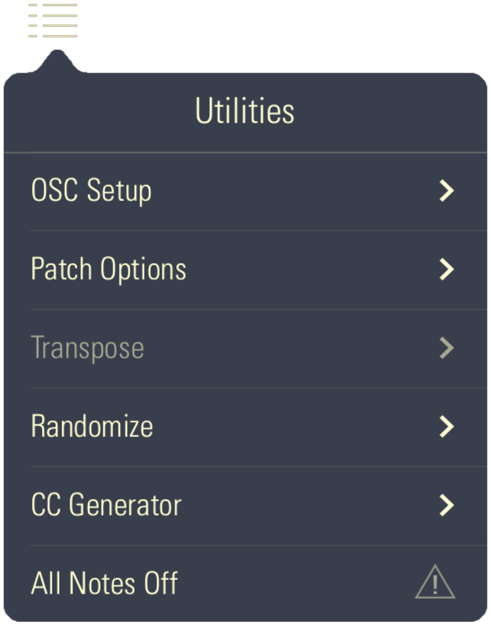


1. Utilities
2. Add Output
3. Output Sections
4. Minimize and Maximize Buttons
5. Outputs
6. Data Entry
7. Live Preview View

The **Patch** view contains the guts of a **TC-Data** patch, and always displays the currently loaded patch. It is the editing area.

## 4.1 Utilities

The **Utilities** popover shows functionality pertaining to the currently loaded patch, as well as OSC address entry and a dedicated CC generator for MIDI mapping.



**OSC Setup** is where the target IP address, port, and device message tag are entered.

**Patch Options** include setting custom grid divisions and suppressing the first touch. Choose **Suppress First Touch** if the patch uses multitouch controllers where the first touch may not be useful (like *Distance to First Touch*).

The **Transpose** section is for changing the key filters of all MIDI Note outputs. You can shift by interval, or set to a particular key or mode.

The **Randomize** section is a quick way to globally randomize controllers, triggers, and values.

The **CC Generator** is a utility that mutes all MIDI output except for a ramping CC value of your choice. Use this when a synthesizer supports MIDI mapping, but only by listening to incoming MIDI data. The CC Generator outputs on MIDI Channel 1.

Press **All Notes Off** ⚠ to stop stuck MIDI notes.

## 4.2 Add Output

Add outputs and modules from this popover. For more information on outputs and modules see [9.0 Outputs].

## 4.3 Output Sections

The Patch view is organized into sections by output type. The sections are:

- OSC
- OSC Trigger
- MIDI Note
- MIDI CC
- MIDI Pitch Bend
- MIDI Aftertouch
- MIDI Patch Change
- AHDSR
- LFO
- Table
- Sequencer

## 4.4 Minimize and Maximize Buttons

To conserve screen space, view multiple controller graphs, and speed up editing, outputs can be minimized. Each output has its own minimize button, and can be tapped when minimized to maximize. Entire sections can be minimized with the **Minimize** and **Maximize Buttons**.

## 4.5 Outputs

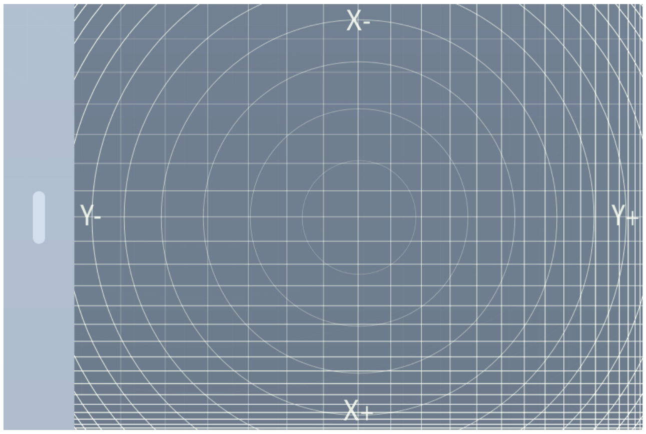
An **Output** is an OSC or MIDI generator driven by one or more of **TC-Data's** controllers. Outputs in the Patch view are organized by section. For more information on outputs and modules see [9.0 Outputs].

## 4.6 Data Entry

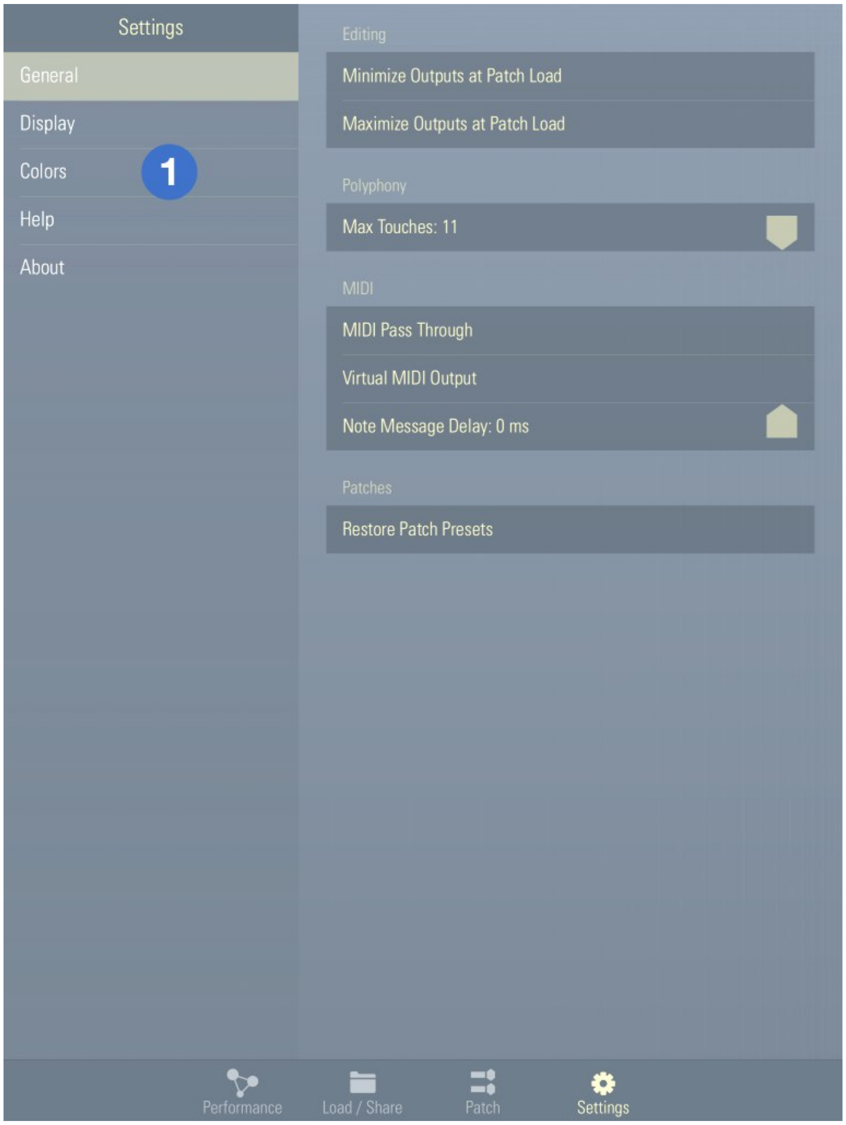
Enter values into output controllers with either a textfield or data slider. Each textfield has a **nub** to its right for quick value changes. Drag up or down to increase / decrease the value. There are two sensitivities, coarse and fine, which correlate to the left / right overlays that appear when dragging.

## 4.7 Live Preview View

The **Live Preview** view allows the currently loaded patch to be played when not in the Performance view. It can be dragged and thrown around the screen, and docks itself automatically on the right side of the screen.



# 5.0 Settings



Settings

General

Display

Colors

Help

About

Editing

Minimize Outputs at Patch Load

Maximize Outputs at Patch Load

Polyphony

Max Touches: 11

MIDI

MIDI Pass Through

Virtual MIDI Output

Note Message Delay: 0 ms

Patches

Restore Patch Presets

1. Settings Sections

The **Settings** view is where you can set global user options, such as interface options, display settings, and color changes. The Settings view also has a quick help reference section.

## 5.1 Settings Sections

### 5.1.1 General

The **General** section allows you to customize editing, performance, and MIDI options.

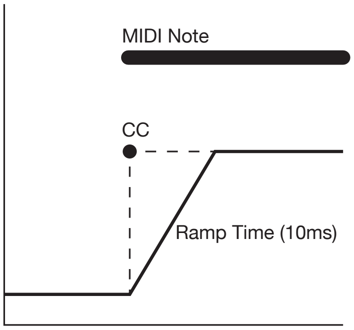
The **Minimize / Maximize Outputs at Patch Load** options will shrink or expand all outputs for newly loaded patches. If neither is selected, the patches own output minimize settings will be preserved.

**Polyphony** in **TC-Data** is how many touches are allowed during performance. For example, if **Max Touches** is set to five, a sixth touch will steal the earliest touch's polyphonic slot. Polyphony does not have anything to do with the number of MIDI notes being generated.

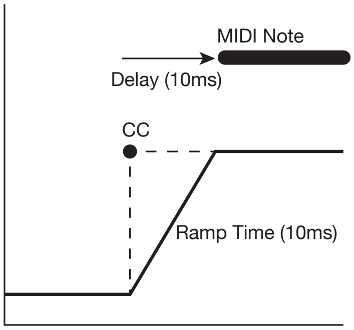
The **MIDI Pass Through** setting takes incoming MIDI messages, for example from a connected MIDI interface, and sends them out to all MIDI output destinations. Connect a MIDI keyboard controller to the MIDI IN port to combine the notes with **TC-Data's** continuous controller generation.

**Virtual MIDI Output** is a virtual port created by **TC-Data** for applications that do not automatically create their own virtual MIDI destination. **TC-Data** will appear as a valid input source with this option enabled.

The **Note Message Delay** amount is how many milliseconds a MIDI note message will be delayed before firing. This is useful when target a synthesizer ramps CC message values, and the note messages occur too quickly.



Without MIDI Note Message Delay



With MIDI Note Message Delay

Pressing the **Restore Patch Presets** button will prompt you to restore any deleted preset patches that come with **TC-Data**. This function will copy missing presets into the User Documents folder of **TC-Data**. It will not overwrite any existing patches.

### 5.1.2 Display

The **Screen Axes** options **Invert X Axis** and **Invert Y Axis** will flip the minimum (-) and maximum (+) edges for the Performance area and Live Preview view. See [2.3 Axis Labels] for more information about the screen orientation.

The **Performance View** options provide a choice for user interface buttons and labels to be visible on screen. Show or hide the Quick Load Button, Patch Title, and Axis Labels. **TC-Data** can show a representation of MIDI note on velocity for each touch. Also, a scrolling Message List can display all outgoing messages being generated.

> *Note:*
>
> *Turning off **Show Message List** can improve interface responsiveness in the Performance view on some devices.*

**Reduced Drawing Mode** is an option to turn off anti-aliasing in the Performance view. Some devices will see a marked improvement in interface responsiveness with this option enabled.

**Custom Controller Drawing** is a way to enjoy the many controller graphics displays in the Performance view. Normally, only the controllers used in a patch will display their graphics cues. With **Override Patch Controller Display** enabled, you can choose to display whichever cues you want for a custom display.

Note that Override Patch Controller Display is a purely aesthetic option. The controllers displayed are not doing anything (except looking good).

### 5.1.3 Colors

Choose the color set that best suites you. The entire interface will change to reflect the color set chosen.

### 5.1.4 Help

Here there are a number of in-app videos to watch to learn about connecting OSC and MIDI from **TC-Data**, as well as some interface lessons.

### 5.1.5 About

Contact information and links to the App Store to rate **TC-Data**. There is no in-app reminder to rate **TC-Data**, so consider this my one moment of asking, and thanks!

• • •

# Performance

## Performance Basics

**TC-Data** was designed from the ground up as a multitouch driven controller. That is to say, the multitouch data itself is what generates control data. Some apps have buttons, sliders, and other on-screen objects that are controlled with a multitouch interface. In these systems, the multitouch capability is meant to enhance the interface accessibility, but it is not the means of control.

**TC-Data** takes a different approach. The multiple touches on the screen are examined, and relationships between them are used as controller data streams. Take this example: in most controllers, there may be a knob that targets a synthesizer's volume. The user touches the screen to move the knob, which changes the volume. With **TC-Data**, that job will be given to a touch controller. Perhaps the volume level will be controlled by the *Average Distance to Group Center*. The farther apart the touches are spread, the louder the volume.

The power of this system is the ability to create complex synth responses using sets of simple touch controllers. When a patch uses multiple touch controllers, the performative capabilities change dramatically depending on the style of performance.

## Voice Allocation

### 6.0 OSC and MIDI Voice Allocation

Each touch in the Performance view is given its own polyphonic voice allocation. Users who have created polyphonic synthesis graphs in programs like Pd or Max/MSP may be familiar with this technique.

## 6.1 OSC Voice Allocation

When the very first touch occurs, it is assigned the voice value 0. The next touch to occur will be voice 1, followed by voice 2, and so on until the maximum polyphony value is reached. When that occurs, **TC-Data** will cycle back to the beginning to search for a new free voice slot. In short: any given touch will have a voice value between 0 - 10.

Touches keep their voice values for the duration of the touch lifecycle. If voice 2 is released before 1 and 3, nothing changes about the other voice assignments. If the maximum polyphony is exceeded, the earliest created touch will be destroyed to make space for the newly created touch.

Finally, group controllers and device motion controllers have a special voice assignment of -1. This is because there is only one control value created no matter how many touches are on screen. For example, the *Average Angle To Center* averages every touch on screen to create one control value. A voice value of -1 indicates it is not tied to any particular touch.

## 6.2 MIDI Voice Allocation

For MIDI users, note that assigning a voiced controller (one which has voiced values for each touch) to an output like the MIDI CC output is a special case. To avoid the zippering and wasted messages that would occur if every touch's value were sent to the CC target, only the most recently created touch will send its value.

# Device Motion

The physical motion of the iPad, or **Device Motion**, can be used as a controller. You can turn and twist the device to change filters and effects, or even create vibrato through movement. The screen will display an orientation graphic that will move based on the motion of the device. Follow the graphic carefully to learn how to maneuver the iPad.

For more information about Device Motion, see [7.3 Device Motion Controllers].

## iPad Rotation

Since physically moving the iPad can be part of the synthesis generation, device rotation is disabled while in the Performance view. To temporarily enable rotation, double tap the Show Navigation Bar Button. You can then rotate the device to your current orientation.

The axes of the device are treated statically throughout **TC-Data**. The rule is: the X-axis is always the left / right axis when holding the device with the Home button along the bottom, and the Y-axis is always up / down. There is a strong reason for this. When programming, it is significant which axis is assigned to a parameter. If the axes changed arbitrarily, it would be impossible to retain consistency in performance.

## Performance Tips

When first performing any patch, try a variety of touch interactions. Some patches are designed to be held down for long durations, others are meant to be ‘plucked’ with short touches. Here are a few common gestures that can trigger complex responses from patches:

- Hold multiple touches down, and move them around the screen
- Change the number of touches on the screen
- Move touches around the screen quickly, or slowly
- ‘Throw’ the touches by moving them quickly, then releasing
- ‘Pluck’ the screen by tapping and releasing quickly
- ‘Stretch’ a group of touches by moving them farther apart, then closer together
- ‘Twist’ a group of touches like you are turning a knob

Be aware that while some patches may feel as though they are sensitive to touch pressure, the iPad does not respond to touch pressure. Use a light touch when performing **TC-Data**. The capacitive touch screen is sensitive to the charge of your finger, not the pressure it exerts on the screen (that is a resistive touch screen).

If you are performing **TC-Data** in a live show, you may wish to turn off your iPad's automatic sleep function (Auto-Lock). You can turn off this function in the iPad's Settings, under General.

Prevent accidental Home Button taps in the System Preferences under General >> Accessibility >> Guided Access.

Also, be sure to turn off Multitasking Gestures in the iPad's Settings, under General. Otherwise, moving groups of touches may close the app, or switch to another open app.

• • •

# Patch Editing

## Controllers Explained

At the heart of **TC-Data** is a complex system that pulls as much user interaction as possible from the iPad. This system analyzes multitouch relationships and the movement of the device to create controllers. Each controller represents a single mode of interaction.

It is important to understand the different types of controllers to make the most out of TC-Data in your projects.

### 7.0 Voiced vs. Global Controllers

Some controllers generate their values by following a single touch voice [6.0 OSC and MIDI Voice Allocation]. These controllers are **Voiced** controllers. A simple example of a voiced controller is *Touch X Position*. Each touch has its own position along the x-axis, so the controller reports a value for every active touch.

Other controllers may depend on the entire collection of touches to generate their value. These controllers are **Global** controllers. A basic global controller is *Group Count*. This controller sends a value based on the number of currently held touches (between 0-8). This number is not unique to a single touch voice, and so the controller will report the same value to *every* voice that follows it.

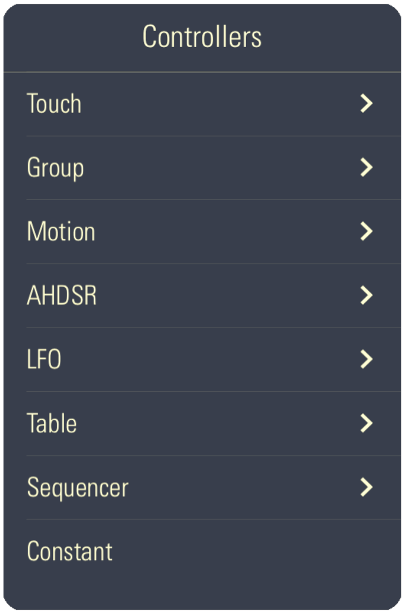
When assigning controllers to outputs, it is important to know whether the output parameters should follow a voiced or global controller. More information about controllers can be found in the following section.

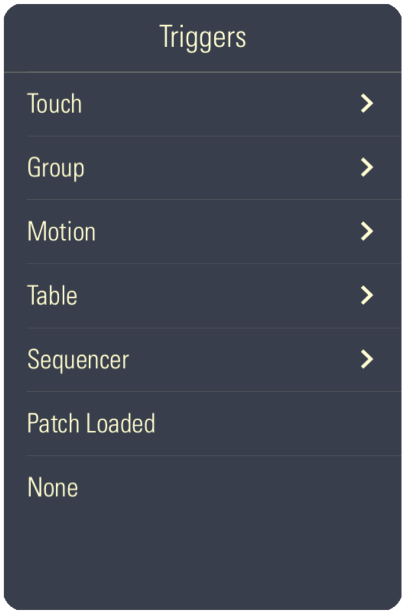
## 8.0 Controller Types

TC-Data has a variety of controllers to use when building patches, most of which pull data directly from multitouch relationships.  There are four types of controllers:

1. Touch-based controllers
2. Device Motion controllers
3. Module controllers (AHDSR, LFO, Table, Sequencer)
4. Special controllers (Constant, Patch Loaded, None)

In this section, the different controller types will be explained.





## 7.2 Multitouch Controllers

Multitouch controllers are created by touching the screen in the Performance view. Multitouch data streams are built by comparing these touches to one another.

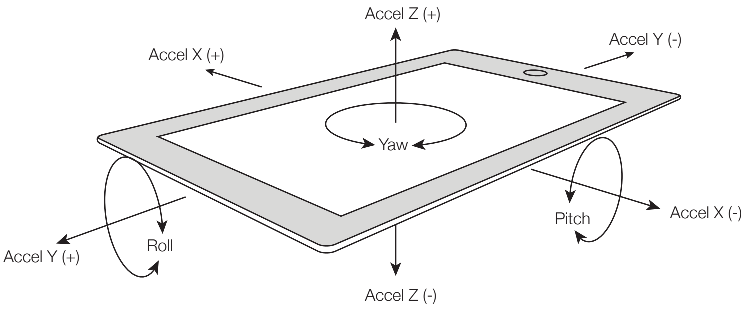
Multitouch controllers are split into two categories. **Touch** controllers are voiced controllers that have values for every active touch. Note that in many cases more than one touch is needed to generate the data. For example, the *Angle To Previous Touch* checks the position of two touches to find the angle between them, and assigns that value to the latter touch.

**Group** controllers are global controllers that examine the entire collection of active touches (the group) to generate their data. One example is *Total Distance To Center*, which sums all of the touches' distances to the center of the screen.

Like the **Controllers** popover list, there is a corresponding **Triggers** popover list that shows triggers generated by touch data, device motion, and modules. The same voiced and group categorization applies.

## 7.3 Device Motion Controllers

**Device Motion** controllers are global controllers that pull data from the physical movement of the iPad. There are three device motion sensors: the accelerometer, the gyroscope, and the compass. The accelerometer measures the rate of change in the device's motion through three axes. The gyroscope measures the device's position in space by describing its rotation around three axes. Finally, the compass compares the direction of the device to a fixed heading: magnetic north.



Accelerometer (x, y, z) and gyroscope
(pitch, roll, yaw) motions

**TC-Data** uses these data sources as continuous controllers. When the iPad's orientation is reset, gyroscope controllers will output their minimum value. As the device is rotated, controller values will change based on the deviation from that position. Accelerometer controllers do not require an orientation reset.

The gyroscope can reset its natural position (orientation). Double-tap the Gyroscope Reorientation Button to zero the rotation values when you are in a comfortable resting position, or when you wish to offset the position to hold a value.

Since the compass reports the same motion as gyroscope yaw, it will not reset to a new fixed heading with the Gyroscope Reorientation Button is pressed. Use the compass when you want a permanently fixed spatial reference in your patch.

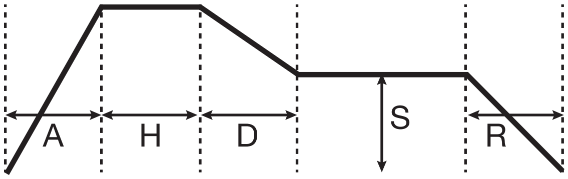
## 7.4 Module Controllers

**TC-Data** has four different **Module** types: **AHDSRs**, **LFOs**, **Tables**, and **Sequencers**. In this section their functionality will be explained. For a detailed look at programming **Modules**, see [Module Programming, p. 41].

### 7.4.1 AHDSR

An **AHDSR** is an envelope generator. AHDSR stands for *attack*, *hold*, *decay*, *sustain*, and *release*. This module ramps smoothly along its envelope shape. Each part of the AHDSR envelope shape can be changed in real-time by another controller. Plus, the output of the AHDSR can be scaled, allowing for complex data streams.

**TC-Data** differs from **TC-11** in how **Hold** is implemented. In **TC-Data**, hold is the duration after the attack time finishes where the maximum point of the envelope is held.



### 7.4.2 LFO

**LFOs** are low frequency oscillators. They can be used to add a slow periodic change to a Parameter, or a wild vibrating effect. Each LFO has an assignable waveform that cycles at controllable rate. In addition, the cycling waveform can be scaled (amplitude adjusted), and shifted (center position change). When a Parameter is controlled by an LFO, it will be modulated by the incoming slow-cycling waveform.

### 7.4.3 Table

A **Table** is an indexed array of values. Its **Index** parameter takes incoming controller data and maps it to read through the Table from start to finish.

Think of the Table as a way to discretize a controller's values. For example, if a Table's **Index** parameter is set to *Touch X Position*, the incoming X position is used to look up the value at the corresponding position in the Table. Instead of a smooth stream of values that comes from *Touch X Position*, a set of discrete values would be output to create a 'stair-step' data stream.

*Note:*

*When a new index position is read from a Table, the value will only be sent once, and the Table will wait for a new index to be accessed. This prevents duplicate messages from clogging the output stream.*

### 7.4.4 Sequencer

**TC-Data** has a powerful step **Sequencer**. The Sequencer stores up to 128 values which it can play through to generate a customized stream of values. All Sequencer parameters are controllable: playback rate, value adjustment, sequence length and position, and playback controls.

For more information about using Sequencers, see [14.0 Sequencers].

## 7.5 Special Controllers

### 7.5.1 Constant

Most parameters can be set to a **Constant** controller (the exception is the OSC Output, which must have a variable controller). Parameters set to constant controllers have their values set once by the user.

Outputs driven by constant controllers send their values when the following events occur:

1. The user changes the constant value
2. A patch finishes loading

### 7.5.2 Patch Loaded

The **Patch Loaded** trigger is useful for firing a trigger once when the patch first loads. For example, starting playback of a Sequencer immediately after loading a patch.

### 7.5.3 None

As the name suggests, the **None** trigger allows a parameter to have no trigger control.

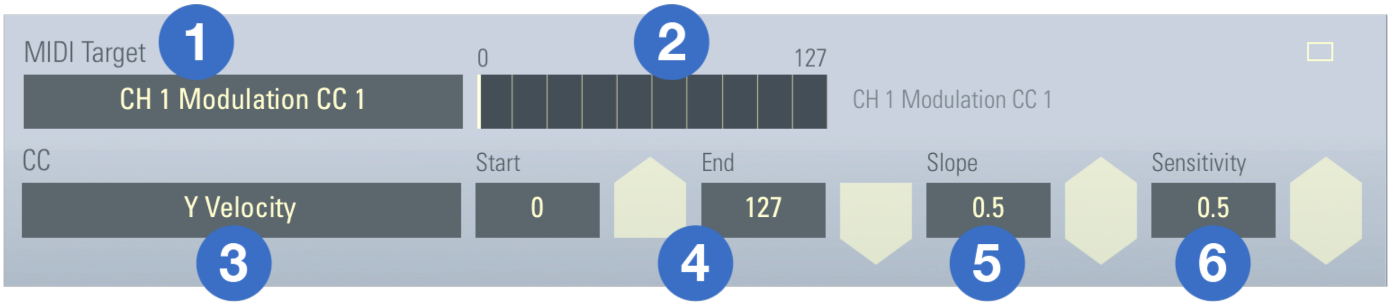
• • •

# Output Programming

## 9.0 Outputs

An **Output** has one main function: translating **TC-Data** controllers into standard MIDI or OSC messages. Outputs are organized by their outgoing message type, for example MIDI Note or OSC Trigger.

To create an output, use the + Add Output button. To delete an output, tap and hold the output and choose the Delete menu option. To duplicate the output, choose Duplicate.



MIDI Target: CH 1 Modulation CC 1

CH 1 Modulation CC 1

CC: Y Velocity

Start: 0

End: 127

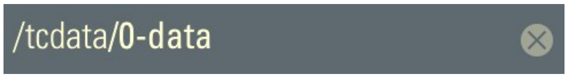
Slope: 0.5

Sensitivity: 0.5

1. Output Target
2. Controller Graph
3. Controller Selection
4. Start and End Values
5. Slope
6. Sensitivity

### 9.1 Output Target

The **Target** for any output is the destination where the converted output message is sent. For OSC outputs, the target is a textfield where a custom message tag is entered:



/tcdata/0-data

Note that the first part of the message is always the **Tag** set in **Utilities >> OSC Setup**.

For MIDI outputs, the target is a **MIDI Port** and **Channel** as detected by the CoreMIDI system. By default, **TC-Data** sets MIDI outputs to broadcast on all available MIDI ports, except for the Network Port.

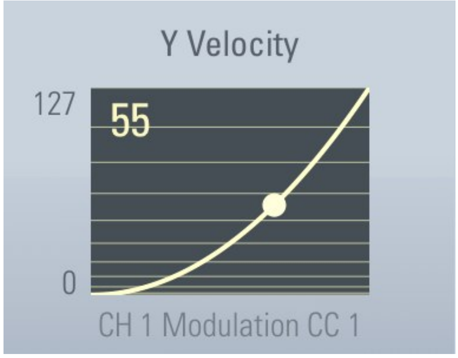
> *Note:*
>
> *The Network Port is ignored when choosing 'All' for your MIDI output target. This is because the WiFi connection used by Network MIDI can introduce a significant amount of lag in messaging.*

You can choose a specific MIDI port and channel for each individual MIDI output. This allows a single patch to discretely control multiple synthesizers separately.

Note that if the specific target is not available (for example, if another iOS app is not running), **TC-Data** will temporarily set the output to broadcast on 'All' available ports.

## 9.2 Controller Graph

The **Controller Graph** shows a real-time readout of the controller data translated into the output's value range. This is useful to view how the slope and sensitivity of your controller are performing as you play, as well as viewing the exact values sent by outputs.



A minimized output showing the controller graph in full view

## 9.3 Controller Selection

The **Controller Select Button** allows you to choose which controller will be watched by that parameter. When you press the button, you will be shown a list of controller categories that are appropriate for that parameter. You can navigate the menu to choose a controller. When you tap the controller name, it will be selected as the source for that parameter.

This is where most of the fun happens in programming **TC-Data**. Choosing a novel way of controlling a parameter may open up a new sound or style of performance. Some controllers are driven very directly by the player, like touch-based controllers. Others, like an LFO module, will run by themselves with as much (or as little) user interaction as you decide. When constructing a patch, deciding which areas of your patch you want to control directly is often a good first step.

For more information about controllers, see [Controllers Explained].

## 9.4 Start and End Values

All controllers generate their own particular range of values. A parameter will take that range and map it to its own **Start** and **End** values.

The most basic example of this is the *Touch X Position* parameter. The far left of the screen is the controller's minimum limit (0 pixels), and the opposite side of the screen is the maximum limit (768 pixels). The position of the touch within that range would set the parameter's value, mapped to whatever minimum / maximum value range the user has set (for example, 0-127 for many MIDI messages).

To 'invert' the range of a parameter, simply flip the start and end values (for example, 127-0).

> *Note:*
>
> *When controlling a MIDI Note output with a Table or Sequencer, keep in mind that the sequence values will only represent actual MIDI notes when the start and end values of the Note parameter are 0-127.*

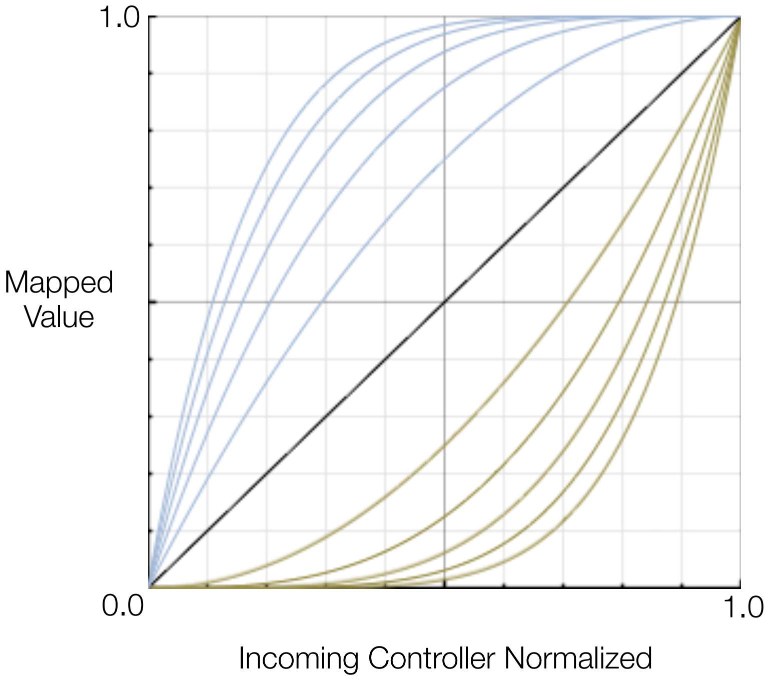
## 9.5 Slope

The **Slope** value weighs incoming controller data along a curve. This makes controllers cover more distance over a particular range and less of another. Setting the slope well can be the key to making a controller ‘feel’ right.

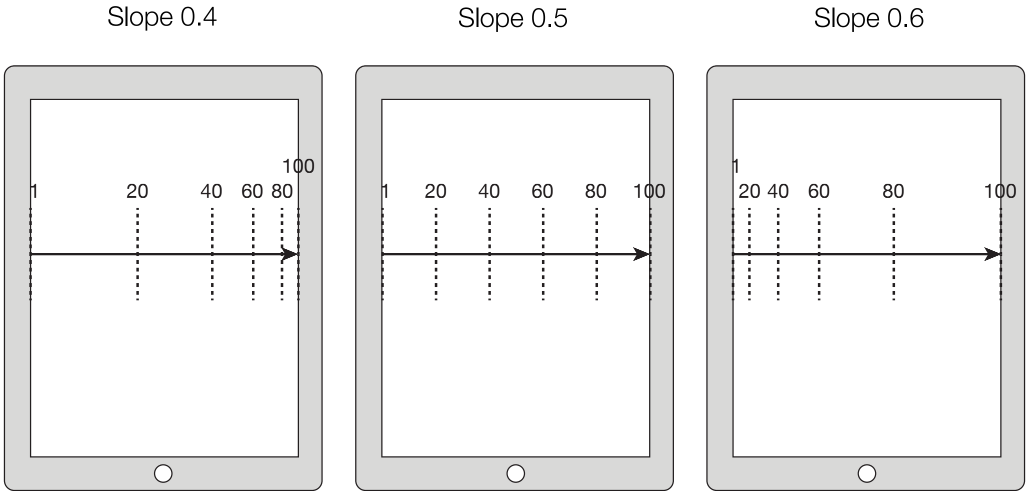
The following table shows exactly how the sloping calculations work, along with a graph to visualize the change. Output controller graphs will also change to reflect how the slope is set.

Slope Calculations

| | |
|---|---|
| **0.0** | **y = x^6** |
| **0.1** | **y = x^5** |
| **0.2** | **y = x^4** |
| **0.3** | **y = x^3** |
| **0.4** | **y = x^2** |
| **0.5** | **y = x** |
| **0.6** | **y = 1 - (1 - x)^2** |
| **0.7** | **y = 1 - (1 - x)^3** |
| **0.8** | **y = 1 - (1 - x)^4** |
| **0.9** | **y = 1 - (1 - x)^5** |
| **1.0** | **y = 1 - (1 - x)^6** |



Let us look at one example to see how slope can affect a patch. If a MIDI Note On Velocity parameter is controlled by *Touch X Position*, with a range from 1-100, changing the slope will change how much of the screen will be playing stronger notes, and how much of the screen with be playing softer notes. For mostly softer notes, choose 0.4 or lower. For an even spread, choose 0.5. For mostly louder notes, choose 0.6 or higher.



Differences in slope, with the *Touch X Position* mapped to
a parameter with a range of 1-100

## 9.6 Sensitivity

Some controllers have a user-definable **Sensitivity** setting. For example, you can decide whether *Touch Speed* reacts to slow movements in a drastic way (high sensitivity), or if it should require a violent gesture just to set it going (low sensitivity). Not only does this allow flexibility in performance, but it can dramatically change how an output parameter reacts to a controller.

In particular, Sensitivity should be addressed by the user when using controllers in the **Time** category. For example, *Creation Time Since Previous Touch* can be used to simulate chording if the sensitivity is high enough.

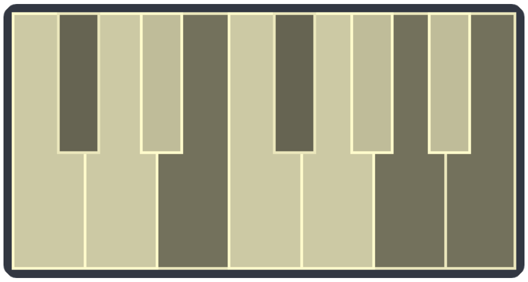
The sensitivity value will appear automatically for controllers that support it. A higher value makes a controller more sensitive, a lower value makes it less so.

# 10.0 Special Output Features

### 10.0.1 MIDI Note Output

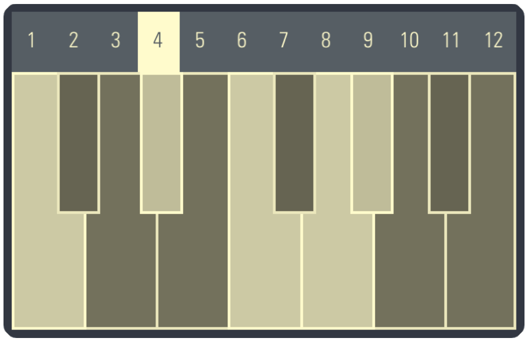
The default MIDI Note output only shows the **Note** and **On Velocity** parameters. You can also control **Off Velocity** and **Polyphonic Aftertouch** (Poly) by tapping and holding the output, and choosing **+ Off Velocity** or **+ Poly**. If you remove those parameters, they will revert to constants.

Each MIDI Note output has an optional **Key Filter**. Tap the Note parameter graph to open the key filter popover. Highlighted notes will play, and notes generated that do not fit the custom scale will be shifted to the nearest acceptable note.



A key filter will C#, E, F#, A, and B filtered out

Note outputs also have a controllable **Key Filter Parameter** which allows up to 12 filters to be shifted through during performance. Tap and hold the MIDI Note output, and select **+ Key Filter** to add the parameter control. Multiple scales can be constructed and changed on the fly using any of the available controllers.



A key filter with active controllable parameter

Try using different types of controllers to shift through different keys. For example, connect *Gyroscope Roll +/-* to the Key Filter parameter to tilt the device through three different scales or chords.

### 10.0.2 MIDI CC Output

When MIDI programs and hardware receive messages, they hold the new value until a new value occurs. This can be a problem when one patch sets a CC value that another patch ignores. For example, if CC #7 Volume is set to 0, and the user loads a new patch which does not send CC #7, the target synthesizer will stay at zero volume.

To compensate for this, **TC-Data** sends the following common CC values at the beginning of every patch load:

| | |
|---|---|
| CC #1 | 0 |
| CC #7 | 127 |
| CC # 10 | 64 |
| CC #11 | 127 |
| CC #64 | 0 |

CC messages sent at patch load

### 10.0.3 MIDI Patch Change Output

You can add the **Bank Change** parameter by tapping and holding the output, and choosing **+ Bank**.

If the Patch Change parameter is set to a constant controller, the constant value is sent at launch. It receives a special dispatch order over other constant parameters. Patch Change constants are sent first, followed by a 100ms delay, followed by the rest of the initial values.

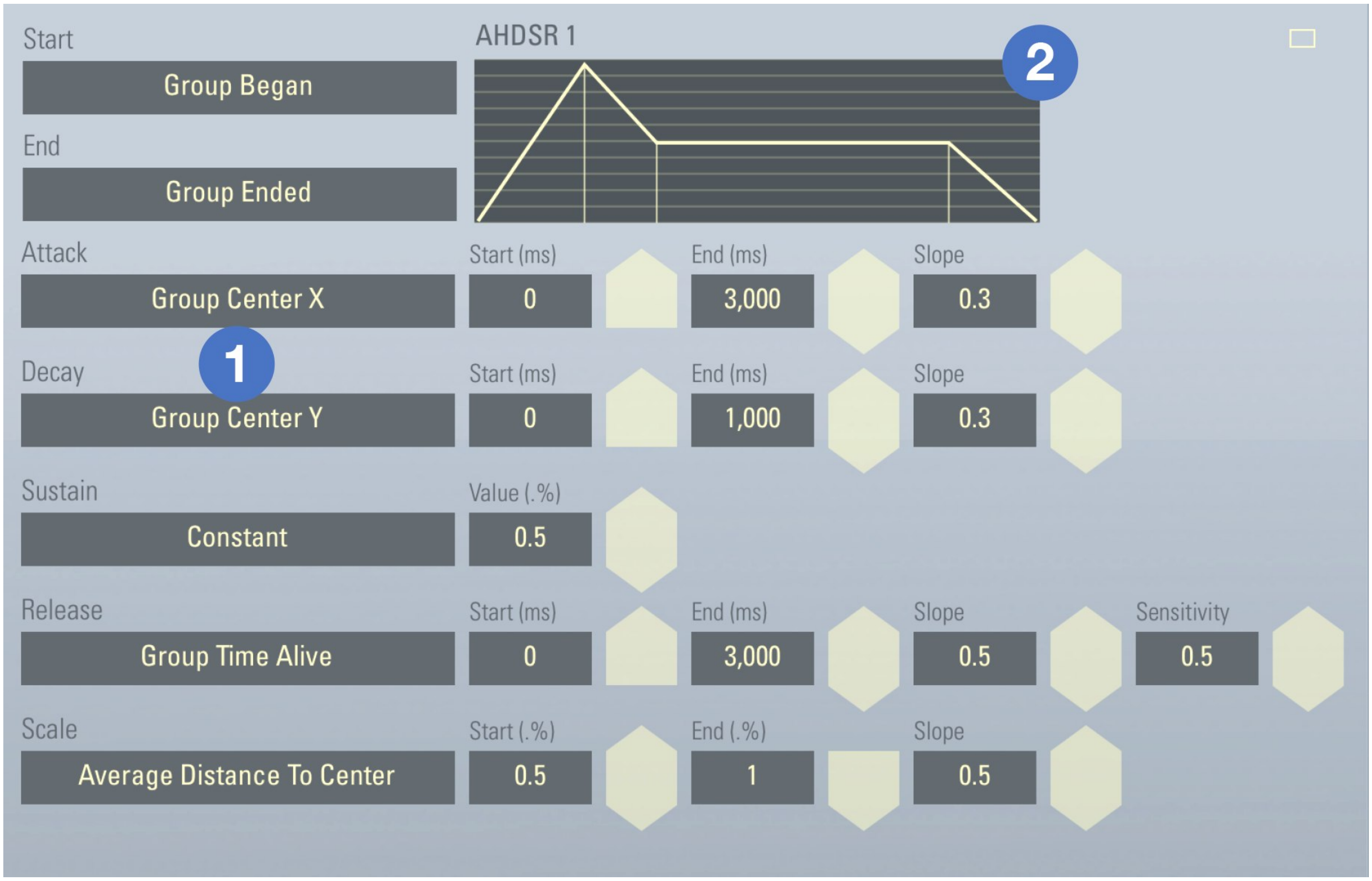
• • •

# Module Programming

The three **Module** controllers in **TC-Data** are the **AHDSR**, **LFO**, and **Sequencer**. In this section we will look at their functionality and some tips for using them in your patches.

## 11.0 AHDSR

The **AHDSR** is an envelope generator. When triggered, it ramps a control value along a user-defined envelope shape. Like traditional **AHDSRs**, the envelope starts at a zero [0] value, then ramps to its full value before returning to zero [0].



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Start | Group Began | | | | | | | | |
| End | Group Ended | | | | | | | | |
| Attack | Group Center X | Start (ms) | 0 | End (ms) | 3,000 | Slope | 0.3 | | |
| Decay | Group Center Y | Start (ms) | 0 | End (ms) | 1,000 | Slope | 0.3 | | |
| Sustain | Constant | Value (.%) | 0.5 | | | | | | |
| Release | Group Time Alive | Start (ms) | 0 | End (ms) | 3,000 | Slope | 0.5 | Sensitivity | 0.5 |
| Scale | Average Distance To Center | Start (.%) | 0.5 | End (.%) | 1 | Slope | 0.5 | | |

1. AHDSR Parameters
2. AHDSR Graph

## 11.1 AHDSR Parameters

The **Start** and **End** parameters are the two triggers that start and stop value ramping. A start trigger that occurs before the envelope has finished its cycle will re-trigger the envelope from the beginning.

**Attack** is the time (ms) from zero to the maximum envelope value. **Hold** is the time (ms) where it will stay at maximum before the initial decay. **Decay** is the time (ms) to ramp down to the **Sustain** value, which is the percentage of the maximum where it will stay until the end trigger is received. **Release** is the time (ms) to ramp back down to zero.

**Scale** is a parameter which live scales the entire envelope as it is running. Use this parameter to add an extra mode of control to the envelope output.

To use the Hold and Scale parameters, tap and hold the AHDSR output and choose **+ Hold** or **+ Scale**.

## 11.2 AHDSR Graph

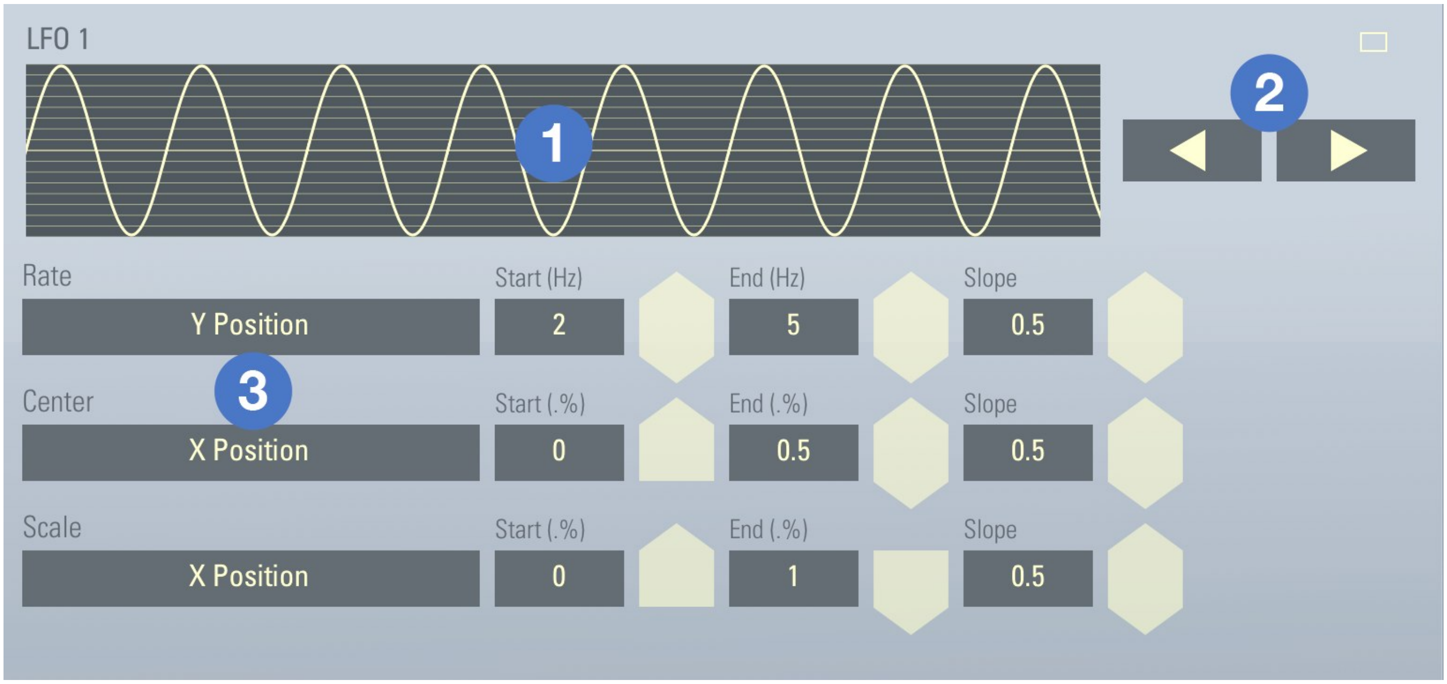
The **AHDSR Graph** is a live update of how the AHDSR is shaped with the current parameter controls. Use it to watch how your envelope shape changes based on your performance.

Note that the AHDSR Graph is not to scale, but rather does its best to represent the changes in values in a clear visual way.

• • •

## 12.0 LFO

The **LFO** is a low frequency oscillator. It has a selectable waveform and control over the rate of oscillation and amplitude. **LFOs** are generally used for periodic control of parameters, but can also facilitate quasi-random parameter control.



| | | | | | | |
|---|---|---|---|---|---|---|
| Rate | Y Position | Start (Hz) | 2 | End (Hz) | 5 | Slope | 0.5 |
| Center | X Position | Start (.%) | 0 | End (.%) | 0.5 | Slope | 0.5 |
| Scale | X Position | Start (.%) | 0 | End (.%) | 1 | Slope | 0.5 |

1. LFO Waveform Graph and Selection
2. LFO Waveform Select Buttons
3. LFO Parameters

## 12.1 LFO Waveform Graph and Selection

The **LFO Graph** shows a live image of how the rate, center, and scale parameters are working together to make the current LFO wave.

Tap the LFO Graph to bring up the **LFO Waveform Select** popover. Scroll through the dozens of waveforms to find the shape you need.

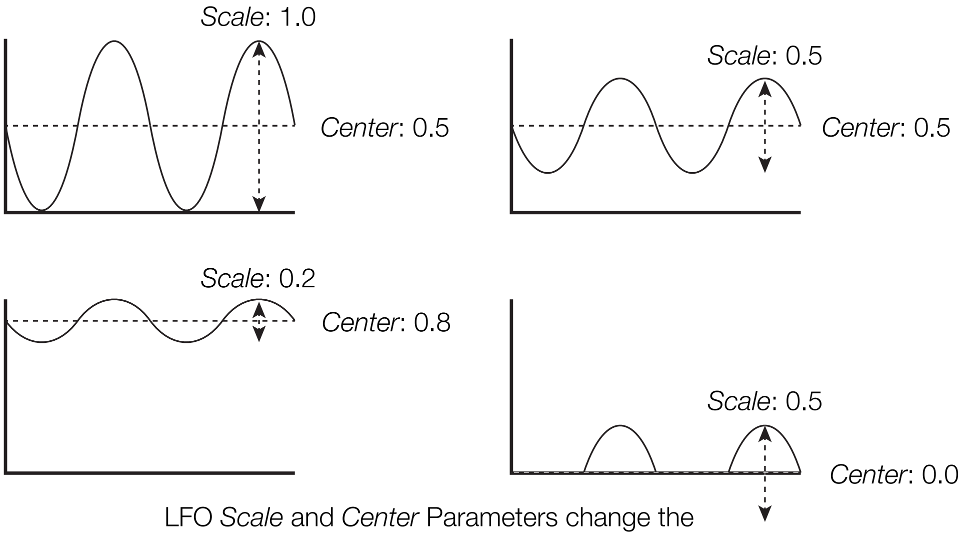
## 12.2 LFO Waveform Select Buttons

There are two **LFO Waveform Select Buttons** to allow for quick cycling through the available waveforms. Tap them to move forward and backward through the list.

## 12.3 LFO Parameters

The **Rate** parameter controls the LFO rate, and has a range of 0-30Hz. The **Center** parameter changes the vertical center of the wave. The **Scale** parameter changes the amplitude scale of the wave.

Use the Center and Scale parameters to make the LFO stretch over the range you need.

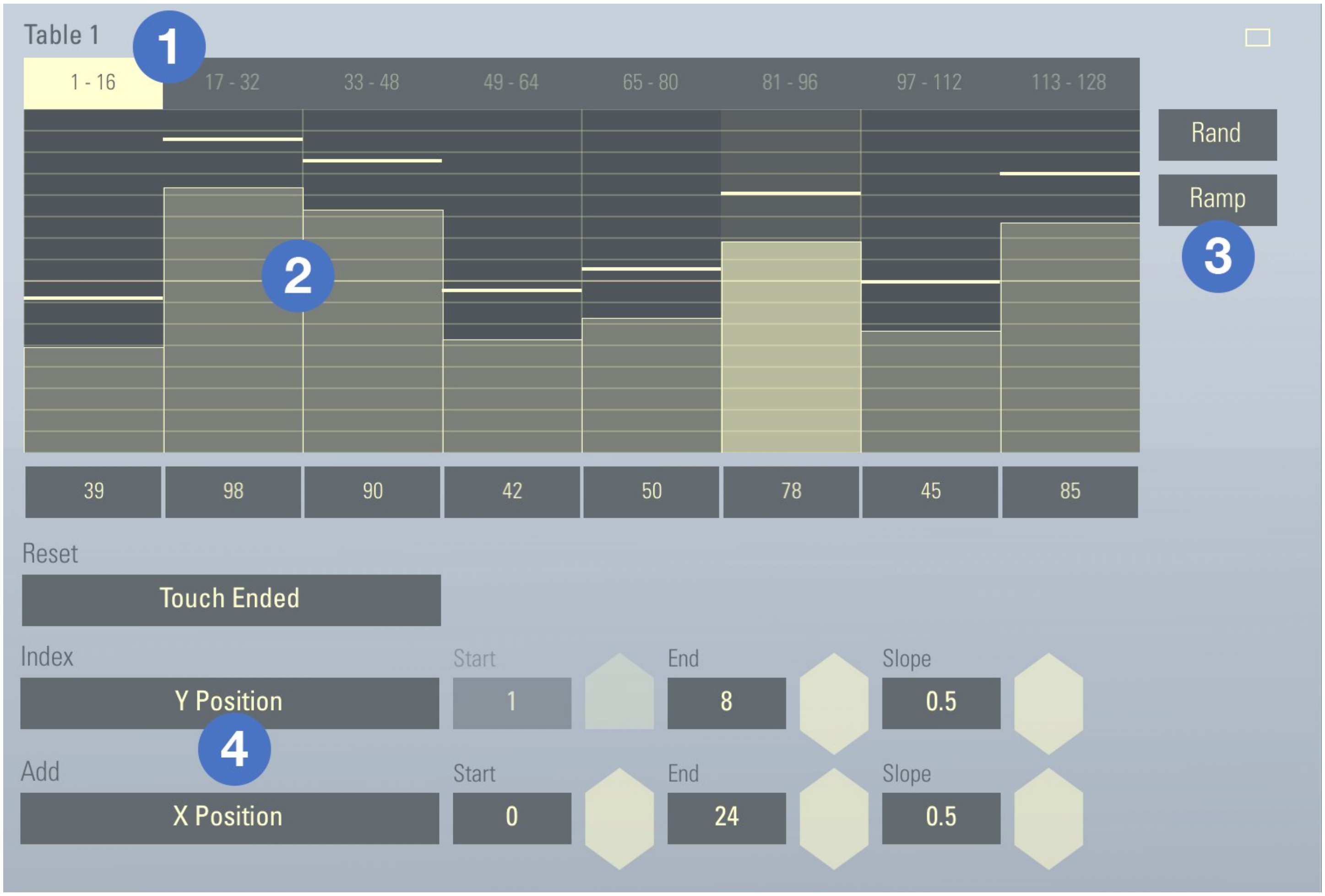


LFO *Scale* and *Center* Parameters change the position and amplitude of the LFO

• • •

## 13.0 Table

A **Table** is an indexed array of values. Its main function is to take an incoming controller to scrub through the table indexes and output the indexed value. This can make an otherwise continuous controller discrete.



39 98 90 42 50 78 45 85

Reset: Touch Ended

Index: Y Position — Start: 1 — End: 8 — Slope: 0.5

Add: X Position — Start: 0 — End: 24 — Slope: 0.5

1. Table Graph Range Buttons
2. Table Graph
3. Table Function Buttons
4. Table Parameters

## 13.1 Table Graph Range Buttons

The Table has up to 128 values, which can be viewed by tapping the range buttons. The Table will automatically switch to the range of values that contains the current index position when a controller changes it.

## 13.2 Table Graph

Set and view the Table values with the **Table Graph**. Tap a value slider, or drag over the graph to change a range of values. Each value is between 0-127 (floating point).

If the Table has an active Add parameter, lines will appear above and below the graph's sliders. These indicate the actual reported value of the Table when read.

The graph will update to show the currently active index by highlighting the appropriate index slider, including switching the view to the necessary range if the Table length is greater than 16.

## 13.3 Table Function Buttons

The **Rand** function sets each value in the active Table length to a random value. The **Ramp** function will evenly step from 0 to 127 over the Table length.

## 13.4 Table Parameters

The **Index** parameter uses an incoming controller to set the current Table index. For example, if *Touch Y Velocity* is used, when the velocity is zero, the Table index would be at the very beginning. As the velocity reaches its maximum, the Table index would be near the end.

The Table will send its value only when a new index is set. This prevents the Index parameter from rapidly reading the same index as it receives its controller values. To force a re-trigger of the same index, choose a trigger for the **Reset** parameter.

For example, if *Touch Y Position* controls the Index parameter, a simple kind of keyboard screen division can be created. Drag a touch from Y- to Y+ to scrub through the Table. If a touch taps a location, then wishes to tap the same location again, a reset command must be sent to the Table to allow the re-triggering of the same index. In this case, a good candidate is *Touch Ended*.

> *Example:*
>
> *Create a new 'Scale' patch. This default patch divides the Y-axis of the screen into 10 divisions. Note how the* Touch Ended *trigger allows for the same index to be sent multiple times in a row.*

The **Add** parameter will add a value to the outgoing Table, allowing for live shifting of the Table values. To use the Add parameter, tap and hold the Table, and choose **+ Add**.

The Add parameter can be used in some clever ways to extend a Table's scale. For example, use another Table to add specifics amounts, like 0 or 12 for an octave step. For more information about using two or more Tables together, see [13.3.2 Table Order].

> *Example:*
>
> *Load the patch* ***Three Octave Scale*** *to see how two Tables work together to create a scale on the Y-axis and octave leaps on the X-axis.*

## 13.3 Table Special Functionality

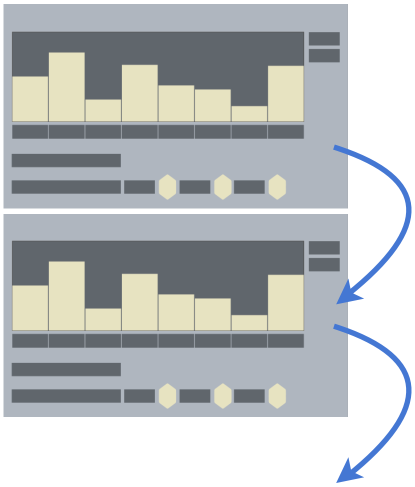
### 13.3.1 Setting Custom Grids

Because Tables are often used to divide the screen, special shortcuts are available to customize the axes into checkered boxes. After setting the Index parameter End value, tap and hold the Table and choose an appropriate axis with **Set X Grid**, **Set Y Grid**, or **Set R Grid**. You can undo your change in **Utilities >> Patch Options**.

Note that grids have a maximum custom division of 32 for the X and Y axes, and 16 for the R-axis.

### 13.3.2 Table Order

When controlling one Table with another, it is important to get the order correct. The rule is: Tables send their values in order from topmost Table downward.



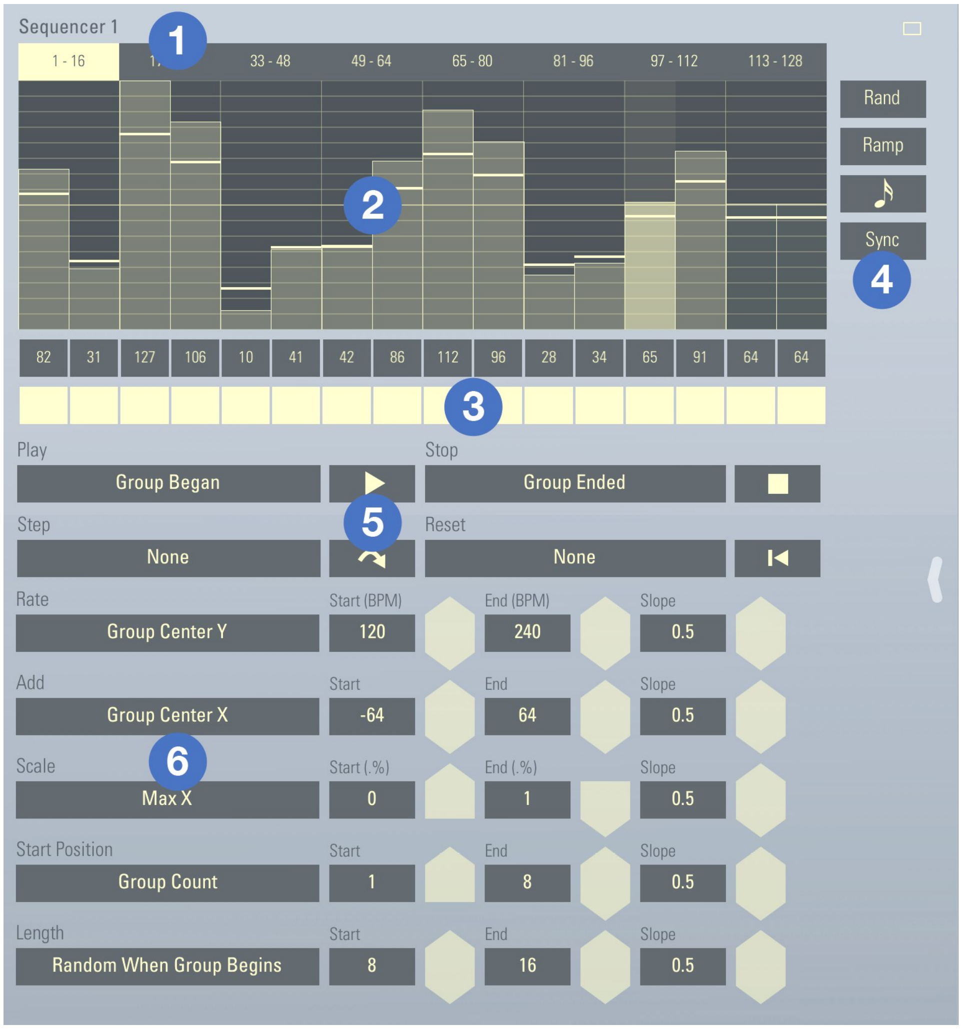
Tables send their values in order from top down

Because of this, Tables which have parameters controlled by other Tables should have their controlling Tables above them. This ensures the proper order of execution for parameter updates.

• • •

## 14.0 Sequencer

A **Sequencer** is playable sequence of values. Like the Table module, Sequencers have a 128 index array where the user can design a series of specific values. Sequencers have the ability to step through the sequence at a user controlled rate, and have more options for live data transforming.



| 82 | 31 | 127 | 106 | 10 | 41 | 42 | 86 | 112 | 96 | 28 | 34 | 65 | 91 | 64 | 64 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Play: Group Began ▶

Stop: Group Ended ■

Step: None

Reset: None

| | | | | | | |
|---|---|---|---|---|---|---|
| Rate | Group Center Y | Start (BPM) | 120 | End (BPM) | 240 | Slope 0.5 |
| Add | Group Center X | Start | -64 | End | 64 | Slope 0.5 |
| Scale | Max X | Start (.%) | 0 | End (.%) | 1 | Slope 0.5 |
| Start Position | Group Count | Start | 1 | End | 8 | Slope 0.5 |
| Length | Random When Group Begins | Start | 8 | End | 16 | Slope 0.5 |

1. Seq. Graph Range Buttons
2. Seq. Graph
3. Seq. Position Gates
4. Seq. Function Buttons
5. Seq. Playback Parameters
6. Seq. Parameters

## 13.1 Sequencer Graph Range Buttons

Sequencers have up to 128 values, which can be viewed by tapping the range buttons. The Sequencer will automatically switch to the range of values that contains the current index position during playback.

## 13.2 Sequencer Graph

Set and view the sequence values with the **Sequencer Graph**. Tap a value slider, or drag over the graph to change a range of values. Each value is between 0-127 (floating point).

If the Sequencer has an active Add or Scale parameter, lines will appear above and below the graph's sliders. These indicate the actual reported value of the Sequencer after being shifted and scaled.

The graph will update to show the currently active index by highlighting the appropriate index slider, including switching the view to the necessary range.

The graph will also dim values that are skipped because their position gate value is off.

## 13.3 Sequencer Position Gates

The **Position Gate** buttons show whether a sequence index will be played or skipped. By default all indexes are set to play. You can create a rhythm by turning some position gates off.

## 13.4 Sequencer Function Buttons

The **Rand** function sets each value in the active sequence length (from its current start position to the end of its length) to a random value. The **Ramp** function will evenly step from 0 to 127 over the sequence length.

The ♪ **Rate Format** button allows the Rate parameter unit to be changed to a preferred musical format. If the rate format is **Hz** (Hertz), the rate of sequence playback will read out in cycles per second. Otherwise, the rate format is displayed in **BPM** (beats per minute), where each step is equal to the note value selected. For example, if the rate format is set to a 16th note, and the Rate parameter is a constant 120 BPM, each step will occur 0.125 seconds after the previous. This is equivalent to a rate of 8Hz.

The **Sync** button allows one Sequencer to have all of its playback, start position, and length parameters synchronized to another **TC-Data** Sequencer. Use this to lock performance of two or more Sequencers together.

## 13.5 Sequencer Playback Parameters

The four playback parameters are **Play**, **Step**, **Stop**, and **Reset**. Play will begin the regular stepping through the sequence. Step will instantly step one index forward. Stop immediately halts the sequence. Reset sets the next sequence index back to the beginning.

There are buttons in the view added for editing convenience. Use these buttons while designing your Sequencer, but consider which triggers you wish to use to control playback during performance.

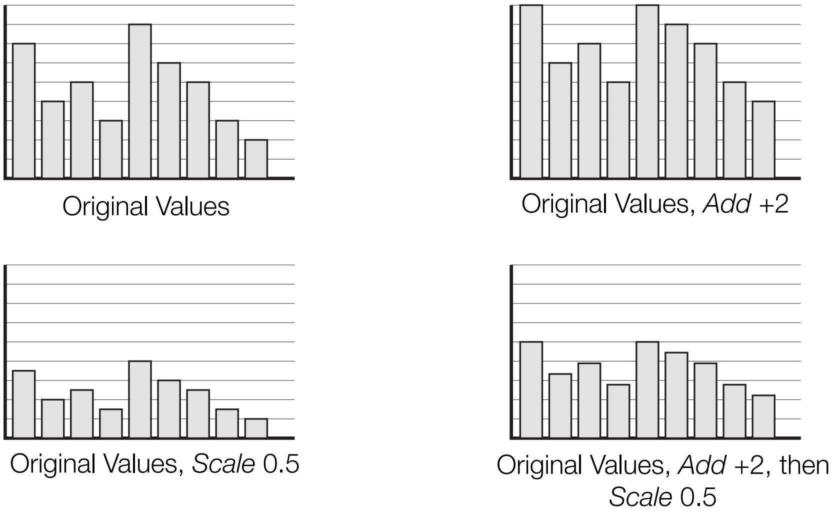
## 13.6 Sequencer Parameters

The **Rate** parameter controls the Sequencer playback rate. Its range is from 0.1Hz to 30Hz.

The **Add** parameter will add a value to the entire sequence, allowing for live shifting of the sequence values. To use the Add parameter, tap and hold the Sequencer, and choose **+ Add**.

The **Scale** parameter scales the sequence values. It can scale between 0-100% of the sequence value. To use the Scale parameter, tap and hold the Sequencer, and choose **+ Scale**.

When using both the Add and Scale parameters, keep in mind that the Add parameter acts first, and then the value is scaled.


Original Values

Original Values, *Add* +2

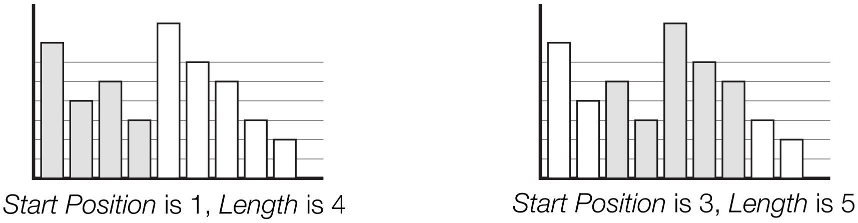
Original Values, *Scale* 0.5

Original Values, *Add* +2, then *Scale* 0.5

Sequencer *Add* and *Scale* Parameters

The **Start Position** parameter allows for the beginning index of playback to move freely through the sequence. Use this to construct long sequences of values which can then be shifted through during playback. To use the Start Position parameter, tap and hold the Sequencer, and choose **+ Start Pos**.

The **Length** parameter determines the sequence length. Sometimes it can be useful to set the sequence length to 1, and simply use the Add and Scale parameters to shift the one playing value around.


*Start Position* is 1, *Length* is 4

*Start Position* is 3, *Length* is 5

*Start Position* and *Length* Parameters

# Appendices

## Troubleshooting

If you are experiencing odd behavior from **TC-Data**, check the following common solutions for assistance. If your issue is not addressed here, please email feedback@bitshapesoftware.com with a description of the problem. If the issue is an app crash, feel free to include the crash log, located at ~/Library/Logs/CrashReporter/MobileDevice/*iPadName/TC-Data_iPad.crash.* This will help keep **TC-Data** as solid of an app as possible.

### 14.0 Issues / Solutions

| | |
|---|---|
| *Target synthesizer makes no sound* | • Check that the target synth is Background Audio enabled<br>• Check that the target synth accepts incoming MIDI ports<br>• Load a new patch, or send CC#7 and CC#11 MIDI messages to reset any volume controls |
| *Graphical slowdown* | • Turn off the Message List in Settings >> Display >> Show Message List<br>• Turn on Reduced Draw Mode in Settings >> Display >> Reduced Draw Mode |
| *The notes created are not what are expected* | • Check the MIDI Note output Key Filter<br>• Check the Note parameter value range, especially when using a Table or Sequencer (0-127 expected) |

| | |
|---|---|
| *OSC messages are intermittent* | •Connect to a different WiFi network<br>•Create a peer-to-peer WiFi network directly with your target computer |
| *OSC message do not appear* | •Make sure the iPad WiFi is connected<br>•Double check the IP address<br>•Double check the port number<br>•Test to see if any messages arrive. If so, check that the message tag routing matches the incoming message tags<br>•Did you really check the IP address? |