MING MECCA

VOLTAGE CONTROLLED VIDEOGAME CONSOLE

USER'S GUIDE

special stage systems

ABOUT SPECIAL STAGE SYSTEMS

Special Stage Systems was founded by Jordan Bartee in 2011. The company operates out of Special Stage Laboratories in Seattle, WA.

Special Stage Systems is:

—Jordan Bartee: Design, engineering, firmware development

—Chris Novello: Design

—Molly Roberts: Software development

FIRST EDITION
FIRST PRINTING—2014

MING MECCA USER'S GUIDE

TECHNICAL SPECIFICATIONS

SPEC	CONTROL CORE	WORLD CORE
HP width	14HP (2.8″)	56HP (11.2″)
Depth	1.75″	1.98″
Current consumption	5V INT +12V: 210mA +5V: 0mA -12V: 0mA 5V EXT +12V: 145mA +5V: 65mA -12V: 0mA	5V INT +12V: 250mA +5V: 0mA -12V: 0mA 5V EXT +12V: 165mA +5V: 85mA -12V: 0mA
CV input range	0-5V	0-5V / 0-1V selectable
CV output range	0-10V	N/A
Gate input threshold	+2.2V	+0.5V
Gate output level	+10V	+10V
Peripheral compatibility	NES gamepads —Original —"Dogbone" —Advantage	Displays NTSC Composite video SD CARDs Full size FAT16 / FAT32 format

TABLE OF CONTENTS

INTRODUCTION

Thank you for purchasing a **Special Stage Systems WORLD CORE**, the central module in the **MING MECCA** voltage controlled videogame console. The **WORLD CORE** is a video-generating **EURORACK** module that uses **TILEs**, **SPRITEs**, and **COLLISIONs** to create videogame-like graphical environments. We like to think of it as an "ontological toy"—a philosophical laboratory for creating, exploring, and experimenting with interactive virtual worlds. But the **WORLD CORE** can be used for all sorts of different purposes. In addition to its world-building capabilities, it's also extremely powerful as a deep and fully customizable digital video synthesizer, as well as a unique voltage controlled **GATE** and **TRIGGER** generator.

Although the **WORLD CORE** is fully functional as a stand-alone unit, it can also be connected to one or more **CONTROL CORE**s to form an enhanced **MING MECCA** system. The **CONTROL CORE** interfaces with NES-compatible gamepads and converts their output into modular-friendly voltages. For users who are interested in experimenting with videogame mechanics and world exploration, we highly recommend the purchase of at least one **CONTROL CORE** for moving **SPRITEs** and manipulating world-states. The full **CONTROL CORE** manual is included in **APPENDIX B**; if you own or plan to own a **CONTROL CORE**, we recommended that you read this first, since some of the **WORLD CORE** patch examples assume familiarity with it.

MING MECCA has been designed as an extendable platform, with many more **CORE** modules planned for eventual release. If you would like to be notified as new designs are announced, please consider signing up for the **Special Stage Newsletter** via our contact page at **specialstagesystems.com/contact.**

Finally, all of us at **Special Stage Systems** would like to thank you for joining us on this new adventure. We can't wait to see what strange and exciting worlds you discover.

1. SETUP

- Unpacking and Inspecting Your Module
- Selecting and Calibrating the CV Range
- Linking a **CONTROL CORE**
- Configuring and Connecting Power
- Connecting a Display
- Powering on the **WORLD CORE** for the First Time

UNPACKING AND INSPECTING YOUR MODULE

In addition to the **USER'S GUIDE**, your package should include the following items:

1. **WORLD CORE** module
2. 16-to-16 pin ribbon **POWER CABLE** (attached to the back of the module)
3. 16-to-10 pin ribbon **LINK CABLE** (unattached)

If any of these items are missing, please contact **Special Stage Systems** to request a replacement.

Before setting up your WORLD CORE, it's important to perform a quick visual inspection to make sure the module has not been damaged during shipping. First check the FRONT PANEL and then turn the module around to look at the MOTHERBOARD. When inspecting the MOTHEBOARD, take care to avoid directly touching any exposed components and connections, as this could result in electrostatic damage. Look closely at the board and verify that none of the header pins* are bent and that there is no obvious structural damage.

If you think your WORLD CORE has been damaged or is otherwise defective, do not attempt to install the module. Please get in touch with **Special Stage Systems** at **support@specialstagesystems.com** to open a support ticket.

*Headers are small rows of gold pins that are used to connect ribbon cables to the motherboard.

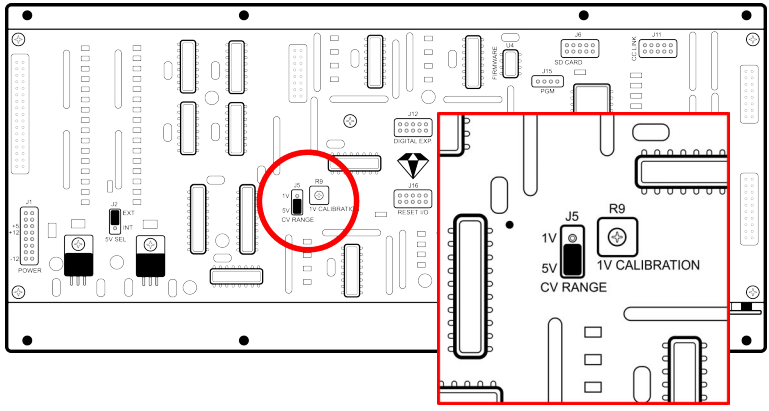
SELECTING AND CALIBRATING THE CV RANGE

Before you can install your WORLD CORE, you'll need to configure some basic settings. First we'll look at the control voltage or "CV" range—the span of voltage that the WORLD CORE's analog input jacks respond to. The WORLD CORE is configured at the factory to respond to a 0-5V CV range. It can also be set up to use an optional 0-1V CV range, which is the standard range used by the LZX VISIONARY SYSTEM, a suite of EURORACK video modules designed and manufactured by **LZX Industries.**

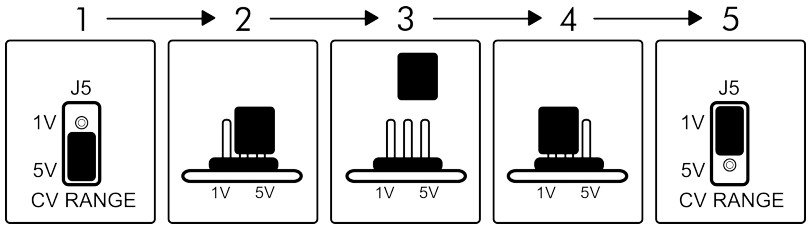
If you do not own an LZX VISIONARY SYSTEM, you may skip to the next section, LINKING A CONTROL CORE; the standard factory setting of 0-5V is the most compatible range for cross-patching with other EURORACK modules, and does not require any user calibration.

SELECTING THE LZX-COMPATIBLE 0-1V CV RANGE

To select the LZX-compatible 0-1V CV range, first locate the CV RANGE JUMPER on the MOTHERBOARD.



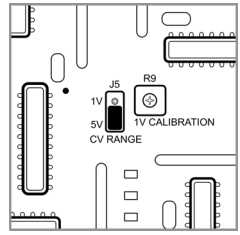
Remove the jumper and reinstall it in the 1V position, as shown in the diagram below.



The WORLD CORE will now use a 1V reference for its CV range. Digital GATE and TRIGGER inputs conform to a 0.5V logic threshold by default and require no user configuration for LZX compatibility.

If you're setting up your WORLD CORE for the first time you can now move on to the next section, LINKING A CONTROL CORE. The calibration instructions below are only necessary if you encounter problems after fully installing your WORLD CORE.

CALIBRATION



The 1V reference is set by a trimpot labeled 1V CALIBRATION, located to the right of the CV RANGE JUMPER. This trimpot is hand-calibrated by **Special Stage Systems** prior to shipment and usually requires no further adjustment.

Occasionally, the 1V reference can drift slightly during transport. This is usually due to mechanical vibration affecting the 1V CALIBRATION trimpot. If you ever find that the WORLD CORE is not responding to the full 1V range, the trimpot may need to be recalibrated. **Special Stage Systems** can perform this recalibration for you, or you can perform it yourself using the following instructions.

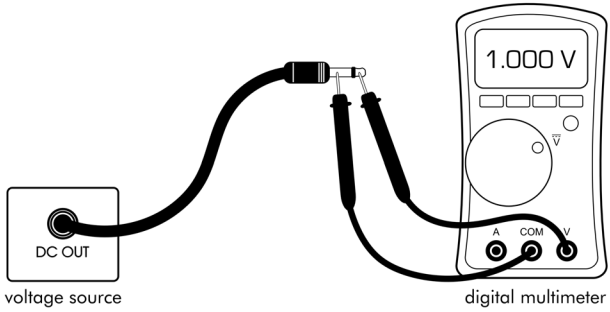
WARNING: This procedure requires that the **WORLD CORE** be powered on while it is removed from the case. Recalibration is therefore recommended for **advanced users** only. If you are not comfortable handling live circuitry, please contact **Special Stage Systems** to arrange a factory recalibration.

To calibrate the 1V reference you will need the following items:

1. A small Phillips-head screwdriver for adjusting the trimpot
2. A voltage source capable of generating a 1V DC offset
3. A digital multimeter with a 3-4 significant digit display

STEP 1 — Setting the voltage source

In order to calibrate the 1V reference we need to have an accurate 1V source to compare it to. Using your digital multimeter, set your voltage source to exactly 1.000V. This is most commonly achieved by inserting a patch cord into the voltage source's output jack, leaving the other side of the cord unpatched. You can then measure the voltage source by placing your multimeter's probes across the tip and sleeve of the patch cord.



STEP 2 – Removing and setting up the WORLD CORE

First make sure that you have dissipated any electrostatic charge you may be carrying by touching a grounded metal object. Touching a kitchen or bathroom faucet works well.

Turn off your modular synthesizer's power and remove the **WORLD CORE** from your case. Leave the power and video cables connected. Carefully hold or set the **WORLD CORE** on a stable surface, make sure your display is turned on, and then turn on your modular.

> Verify that the **WORLD CORE** boots without issue. If you see no video or the module is behaving erratically, immediately turn off the power and make sure nothing is shorting out on the **MOTHERBOARD**.

Sprite 1-1: DIGIMAN

Using the **X** and **Y** knobs, position **SPRITE 1** on the center of the screen. Once **SPRITE 1** is centered, turn the **OBJECT** and **ORIENT** knobs fully counter clockwise to select the **DIGIMAN SPRITE** in its default state. Now turn the **Y** knob fully clockwise and patch your **1V** voltage source into the **Y** CV input. **DIGIMAN** should now be hidden (or, depending on the level of miscalibration, partially hidden) behind the top border of the screen.

STEP 3 – adjusting the trimpot

Using your Phillips-head screwdriver, turn the trimpot counter-clockwise until **SPRITE 1** reappears on the screen. Now slowly turn the trimpot clockwise until the very bottom of **SPRITE 1** has just barely disappeared behind the top border. If the bottom of **SPRITE 1** is flickering in and out of the border, keep turning the trimpot until it stabilizes. The idea is to apply just enough voltage to position **SPRITE 1** fully be-

hind the top border. This may take several small adjustments to get perfect.

STEP 4 – verifying the calibration

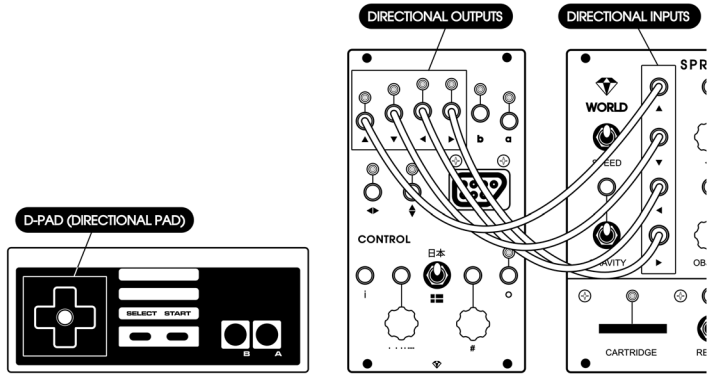
Once you've finished adjusting the trimpot, unpatch your voltage source from the Y CV input. Using the Y knob, sweep SPRITE 1 from the bottom to the top of the screen and make sure that it covers the complete range of positions. If SPRITE 1 doesn't completely disappear behind the top border, patch your voltage source back into the Y CV input and repeat STEP 3.

You have now successfully calibrated the 1V reference. You may turn off the power and reinstall the module in your case.

> Extremely advanced users may prefer to calibrate by directly measuring the voltage across the 1V-pin on the CV RANGE JUMPER. Ground can be tapped from the large ground-plane connected "via" hole directly above the trimpot. **Do not attempt this method if you have no experience with electronics—it is easy to slip and cause short circuits.**

LINKING A CONTROL CORE

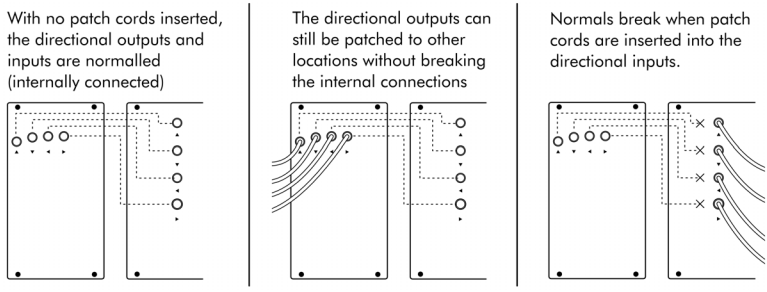
A CONTROL CORE can be used for many different purposes, but its most basic application is controlling the movement of SPRITEs—small graphical blocks that can be positioned anywhere on the screen. The gamepad's DIRECTIONAL PAD ("D-PAD") generates GATEs from the CONTROL CORE'S four DIRECTIONAL OUTPUTs (UP, DOWN, LEFT, and RIGHT). By patching these outputs into the DIRECTIONAL INPUTs located on the WORLD CORE, the gamepad can be used to freely move SPRITE 1 around the screen.



Gamepad Control Core World Core

If this is a little confusing right now, don't worry; you'll learn all about controlling SPRITEs in CHAPTER 4. For now we'll just say that the patch above is a common "starting position" for creating more complex MING MECCA patches.

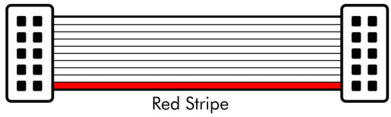
Because this patch is used so frequently, we've provided a way to hardwire it internally using a technique called NORMALIZATION. This automatically connects the DIRECTIONAL OUTPUTs to the DIRECTIONAL INPUTs without the use of patch cords, allowing SPRITE 1 to be moved by the gamepad without any set up.



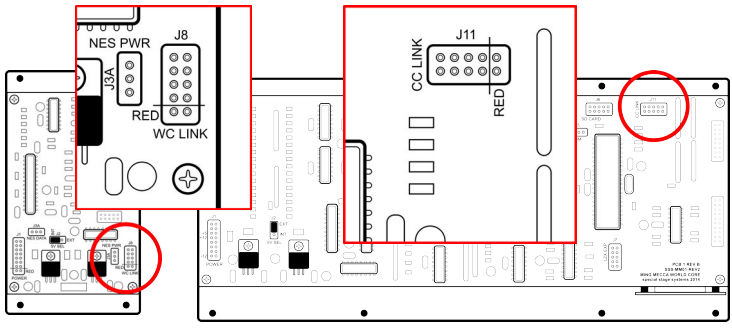
No functionality is lost when using **NORMALIZATION**. The **DIRECTIONAL OUTPUTs** can still be routed to other locations as usual, and the internal connections can be overridden by simply inserting a patch cord into the corresponding **DIRECTIONAL INPUT**.

> **NORMALIZATION** works using a mechanical switch inside the **DIRECTIONAL INPUTs**. When no patch cord is inserted, the switch is closed, making the connection to the **DIRECTIONAL OUTPUTs.** When a patch cord is inserted, it physically lifts the switch to override the connection.

To enable **NORMALIZATION**, the two modules must be **LINKED** by connecting their **MOTHERBOARDs** together. To **LINK** your modules, first locate the 10-pin **LINK CABLE** that was included with your **WORLD CORE**. Note the position of the cable's **RED STRIPE**.



Now locate the **CC LINK** and **WC LINK** header connectors on the **WORLD CORE** and **CONTROL CORE MOTHERBOARDs**.



Position the cable's **RED STRIPE** so that it aligns with the pins marked **RED** on the **WC LINK** and **CC LINK** header connectors. Due to its position on the **MOTHERBOARD**, you will have to rotate the cable 90 degrees to the left when connecting it to the **CC LINK** header on the **WORLD CORE**.

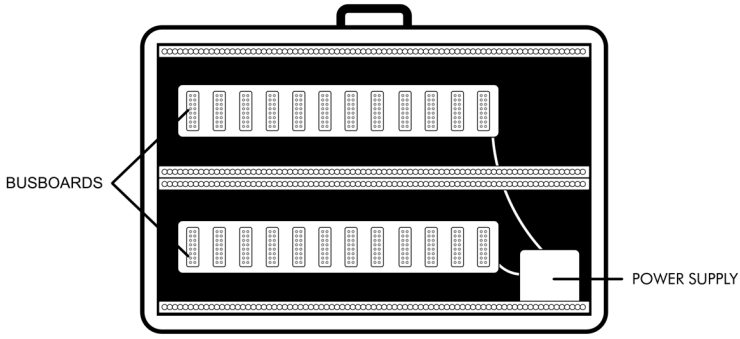
Double-check the **RED STRIPE** alignment, and then firmly press down on each side of the cable connector until the header pins are fully covered. Your **CONTROL CORE** and **WORLD CORE** are now **LINKED**. Only one **CONTROL CORE** can be **LINKED** to a **WORLD CORE** at a time. If you ever decide you would rather keep them functionally isolated, the modules can be **UNLINKED** by removing the cable.

CONFIGURING AND CONNECTING POWER

MING MECCA modules are designed for use within **EURORACK** modular synthesizer systems. In order to use your **WORLD CORE** module you will need to install it in a **EURORACK** case and supply it with **EURORACK**-compatible power. **EURORACK** cases usually have a power supply and power distribution system built in.

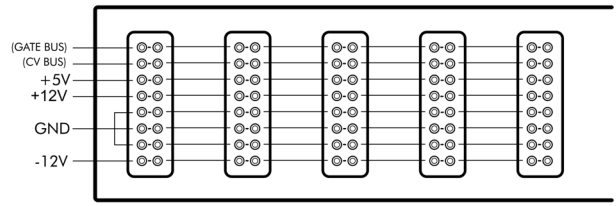
Caution should be exercised when installing any new module in your system. Although **Special Stage Systems** has taken steps to protect your **WORLD CORE** from inverted polarity, it is impossible to predict all potential scenarios given the open nature of the **EURORACK** standard. **Special Stage Systems accepts no liability for damage to the WORLD CORE or to any other connected hardware due to reversed, offset, or otherwise incorrect power connection.** Please follow this guide closely and double-check the ribbon cable before applying power to your modular.

EURORACK CASE



(generic illustration—actual case may vary)

BUSBOARD DETAIL



(standard Doepfer-style busboard shown. For "flying busboard" systems, please consult the manufacturer's documentation)

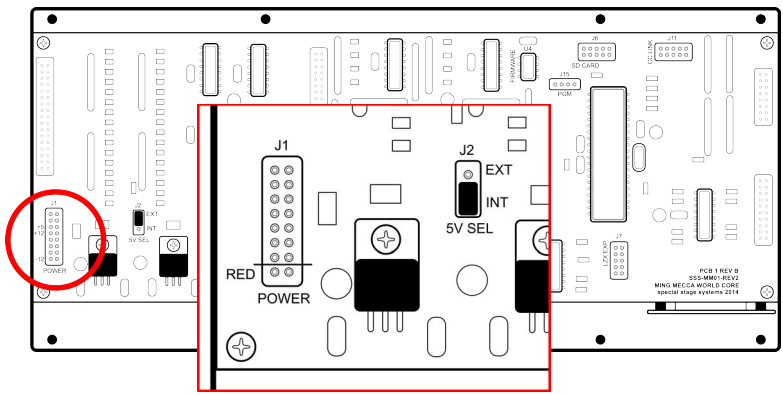
EURORACK modules receive power via ribbon cables that attach to the case's BUSBOARDS. Most BUSBOARDS use headers that are **unkeyed**, which means that it is **possible to plug power in backwards**. Accidentally inverting the polarity can damage not only the reversed module, but any modules connected to the same BUSBOARD as well.

CONNETING POWER

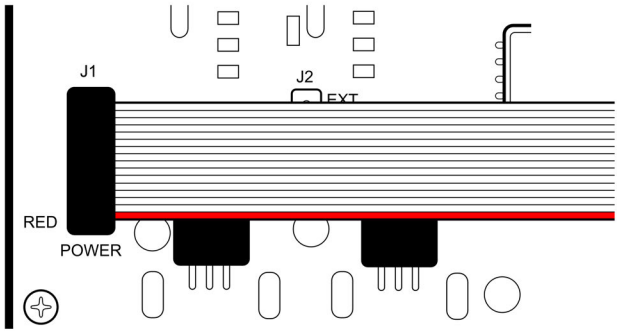
Before connecting power to your WORLD CORE, first verify that your case meets the minimum requirements for use with MING MECCA. You will need a minimum of **56HP** (11.2") of free horizontal space to install the WORLD CORE, and an additional **14HP** (2.8") for the optional CONTROL CORE. Your case must also be sufficiently deep to house the module's internal circuitry. The WORLD CORE measures **1.98"** deep, while the CONTROL CORE measures **1.75"**.

Your power supply must be able to provide at least **250mA** of current on the +12V rail. If you are also connecting a CONTROL CORE, you will need an additional **210mA** of current available for **460 mA** total.

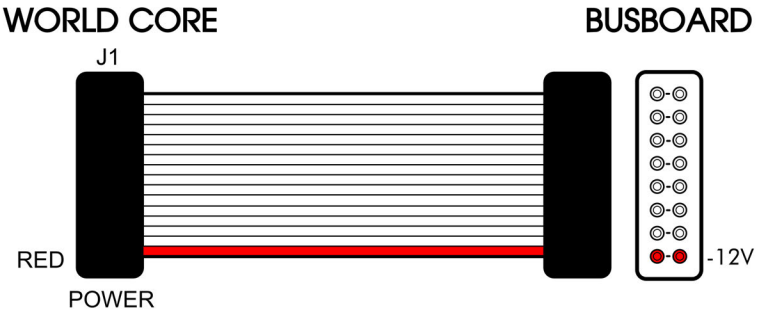
Now locate the 16-pin power connector on the MOTHERBOARD.



The 16-to-16 pin **POWER CABLE** should already be connected to the module. Note the position of the cable's **RED STRIPE**. Verify that the **RED STRIPE** is aligned with the location marked "**RED**" on the power connector. If the cable is not properly aligned, remove and reposition it so that the alignment is correct.



Connect the other end of the cable to your power supply, making sure that the **RED STRIPE** aligns with the **-12V** pins on the **BUSBOARD** header.



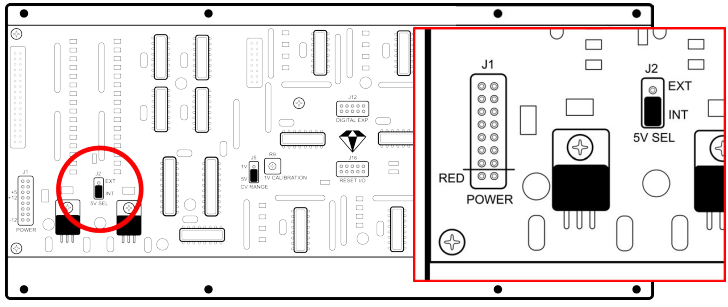
If you are unsure about the orientation of your case's BUSBOARD headers, consult the manufacturer's documentation for more information.

CONFIGURING THE +5V SUPPLY

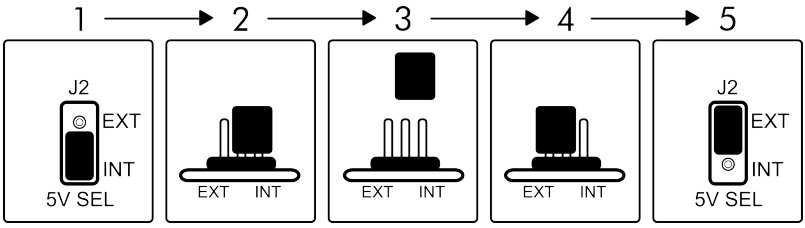
By default, the WORLD CORE generates all the power it needs from the +12V rail. The WORLD CORE can also be configured to use an external +5V supply to power its digital components. Not all EURORACK cases provide +5V power, and many that do are actually derivative of the +12V supply. In EURORACK cases that provide truly independent +5V power however, configuring the WORLD CORE to use the external supply can lessen the load on the +12V rail.

> If you don't know whether your case provides independent +5V power, it's best to leave the WORLD CORE in its default configuration. The option to use an external +5V supply is for advanced users who are looking to maximize the efficiency of their system's power supply.

To configure the WORLD CORE for use with an external +5V power supply, first locate the 5V SELECT JUMPER on the MOTHERBOARD.



Remove the jumper and reinstall it in the "EXT" position, as shown in the diagram below.



WARNING: the **5V SELECT JUMPER** must be installed in either the "**EXT**" or "**INT**" position before applying power. **Never attempt to power the WORLD CORE with the 5V SELECT JUMPER removed.**

WARNING: do not configure the **WORLD CORE** for external **+5V** power if your case does not have a working **+5V** rail. **Installing a WORLD CORE configured for external power in a case that doesn't supply +5V may damage the module.**

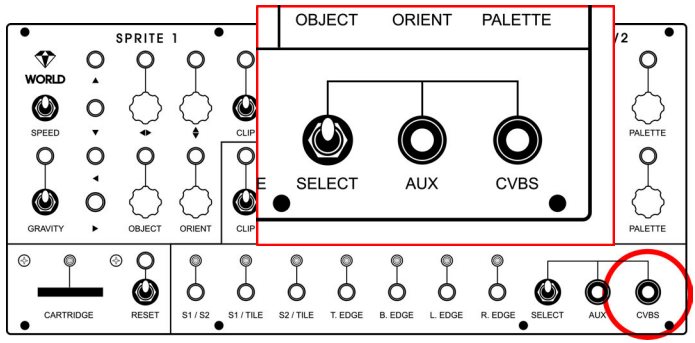
The **WORLD CORE** will now use an external **+5V** supply for its digital components. Note that the **+12V** rail is still required in order to power the analog sections of the **MOTHERBOARD**; always use a full 16-to-16 pin **POWER CABLE** when powering the module, regardless of the **5V SELECT JUMPER** setting.

CONNECTING A DISPLAY

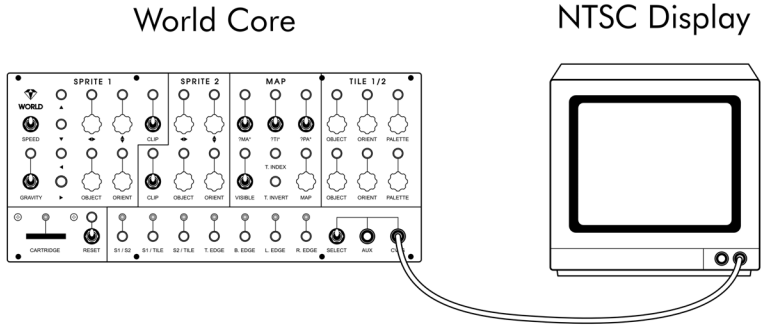
Now that you've connected power to your WORLD CORE, you can install the module in your case using the eight included screws. **But don't turn on the power just yet;** before using your module for the first time, you'll need to connect it to a compatible display.

The WORLD CORE can be connected to any display that accepts composite NTSC video. We recommend the use of a Cathode Ray Tube (CRT) television or monitor. Although these displays are no longer in production, they generally provide the best picture quality and response time when dealing with analog composite video sources like the WORLD CORE. If you do not have a CRT display available, the WORLD CORE will also work with any modern display, projector, or capture device that accepts composite video.

To connect the WORLD CORE to your display, locate the video output marked CVBS on the front panel. (CVBS is an initialism for composite video that stands for "Color, Video, Blanking, and Sync.")



Connect a composite video (RCA) cable to the CVBS jack. Make sure you don't accidentally plug into the AUX jack instead. Now plug the other end of the cable into your display's video input.



Finally, make sure the SELECT toggle is in the upright position. This toggle switch determines whether the CVBS jack will display the WORLD CORE's internal video output, or the signal patched into the AUX video input. When set in the downward position, it selects the AUX input, which will cause the WORLD CORE to display a blank screen when nothing is connected to the AUX jack.

A NOTE FOR USERS IN PAL TERRITORIES

Although the WORLD CORE is currently NTSC only, there are several workarounds available to PAL users:

- **Use a multi-format CRT (recommended)**
  Many higher-end CRTs will display both PAL and NTSC video sources natively.

- **Use a modern display**
  Most composite-compatible projectors and LCD displays are multi-format,but may introduce minor lag and ghosting.

- **Use a video capture device**

USB, FireWire, or Thunderbolt capture devices are usually multi-format, but they can also introduce significant lag.

- **Use an NTSC-to-PAL converter**.

Dedicated boxes can be bought that convert NTSC to PAL. Depending on the quality of the unit, some image degradation may be apparent as a result of the conversion process.

POWERING ON THE WORLD CORE FOR THE FIRST TIME

You are now ready to turn on your WORLD CORE! Make sure your display is turned on, and double-check the position of the SELECT toggle (it should be in the upward position).

As you turn on your modular, watch the LED above the CARTRIDGE SLOT. After a few seconds, the LED should flash once to indicate that the WORLD CORE has passed its internal system check.

Immediately after the LED flashes, you should see a fast sequence of rainbow bars on your display, followed by the **Special Stage Systems** boot logo.

> If you do not see anything on your display, or the CARTRIDGE LED fails to flash, turn off your modular immediately and proceed to the TROUBLEHSOOTING CHART in APPENDIX C.

Once the WORLD CORE has successfully booted, feel free to jump right in and start playing with the controls. Don't worry if some of them seem unresponsive or don't behave like you'd expect. The

WORLD CORE has a lot of settings that interact with each other in complex and sometimes unpredictable ways. You can't damage anything by using the controls, and as long as you stay within the EURORACK power range (-12V to +12V), you can't damage anything by patching into the jacks either.

When you're ready to learn more, reset your WORLD CORE (pull the RESET toggle switch next to the CARTRIDGE SLOT downwards, and release) and move on to CHAPTER 2.

2. BASIC OPERATION

- System Overview and Interface Guide
- Voltage Standards and General Control Paradigms
- How to Read Patch Schematics
- Returning to Default Settings

SYSTEM OVERVIEW AND INTERFACE GUIDE

The WORLD CORE's interface is divided into seven sections. The first four sections are explicitly labeled along the top of the module: SPRITE 1, SPRITE 2, MAP, and TILE 1/2. The final three sections are not labeled, and run along the bottom of the module: SYSTEM, COLLISION, and VIDEO.

The SPRITE 1 and SPRITE 2 sections contain all of the controls and CV inputs related to the position, display, and behavior of the WORLD CORE's two SPRITEs—small graphical objects that can be freely moved around the screen. SPRITE 1 has some special abilities that SPRITE 2 lacks, so its section is slightly larger with a greater number of inputs and controls.

The MAP and TILE 1/2 sections deal with the presentation and placement of TILEs, which are grid-based graphical elements used to create backgrounds and define terrain. The TILE 1/2 section affects the visual properties of TILE 1 and TILE 2, while the MAP section determines how the TILEs are placed across the screen to create larger patterns.

The SYSTEM section has only three elements: the RESET toggle, RESET OUTPUT, and CARTRIDGE SLOT. The COLLISION section contains only outputs, producing GATEs and TRIGGERs depending on the interaction between the SPRITEs and TILEs.

Finally, the VIDEO section contains the main CVBS video output, the AUX video input, and the SELECT toggle for choosing between them.

The following interface guide contains a complete list of every control from each section of the WORLD CORE, and where to find them on the panel.

SPRITE 1

1. **X** and **Y** knobs and CV inputs.
2. **OBJECT** and **ORIENT** knobs and CV inputs
3. **CLIP** toggle and GATE input
4. **DIRECTIONAL INPUTs**: **UP**, **DOWN**, **LEFT**, and **RIGHT**
5. **SPEED** toggle
6. **GRAVITY** toggle and GATE input

SPRITE 2

7. **X**, **Y**, **OBJECT**, **ORIENT**, and **CLIP** controls and inputs

TILE 1/2

8. **OBJECT**, **ORIENT**, and **PALETTE** controls and inputs for **TILE 1**
9. **OBJECT**, **ORIENT**, and **PALETTE** controls and inputs for **TILE 2**

MAP

10. **MAP SELECT** knob and CV input, **VISIBLE** toggle and GATE input, **TILE INDEX** CV input, and **TILE INVERT** TRIGGER input
11. **GLITCH** toggles and GATE inputs: **?MA*** (**MAPs**), **?TI*** (**TILEs**), and **?PA*** (**PALETTEs**)

COLLISION

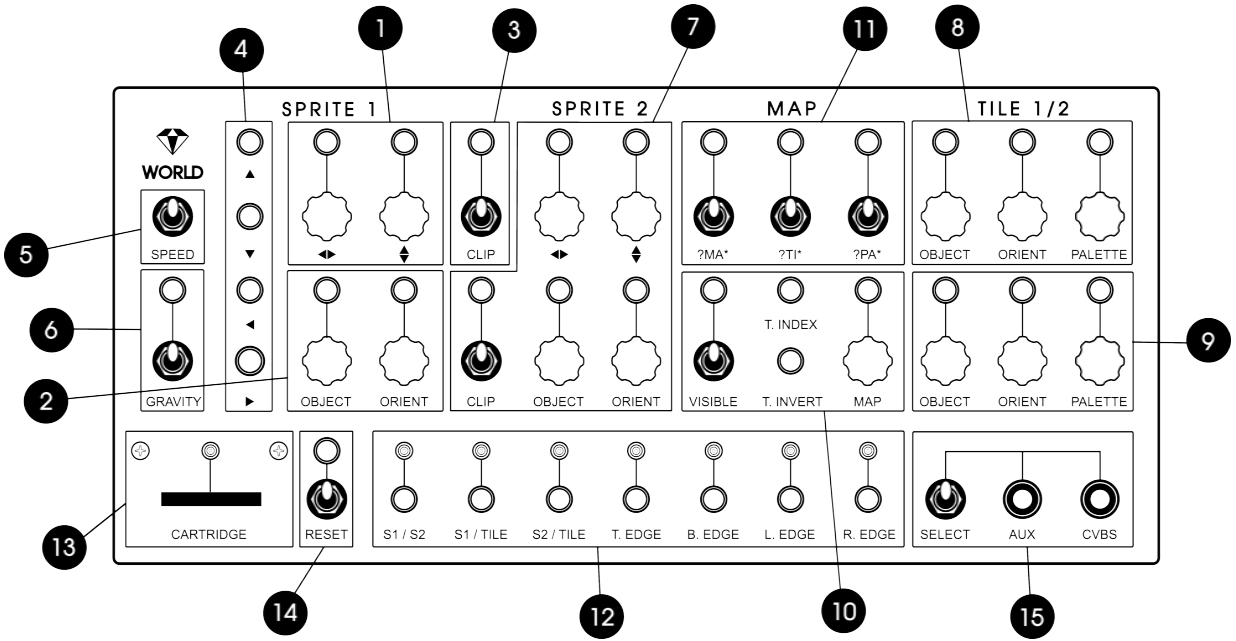
12. **COLLISION** GATE / TRIGGER outputs: **SPRITE 1 / SPRITE 2**, **SPRITE 1 / TILE**, **SPRITE 2 / TILE**, **TOP EDGE**, **BOTTOM EDGE**, **LEFT EDGE**, and **RIGHT EDGE**

SYSTEM

13. **CARTRIDGE SLOT** and **CARTRIDGE LED**
14. **RESET** toggle and **RESET OUTPUT**

VIDEO

15. **CVBS** composite video output, **AUX** composite video input, and **SELECT** toggle


SPRITE 1
SPRITE 2
MAP
TILE 1/2
WORLD
SPEED
GRAVITY
CLIP
?MA*
?TI*
?PA*
OBJECT
ORIENT
PALETTE
T. INDEX
VISIBLE
T. INVERT
MAP
CARTRIDGE
RESET
S1 / S2
S1 / TILE
S2 / TILE
T. EDGE
B. EDGE
L. EDGE
R. EDGE
SELECT
AUX
CVBS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

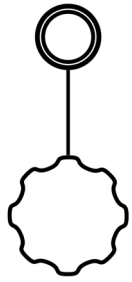
VOLTAGE STANDARDS AND GENERAL CONTROL PARADIGMS

- CV inputs: 0-5V (0-1V with CV RANGE JUMPER)
- GATE and TRIGGER inputs: +0.5V logic threshold
- GATE and TRIGGER outputs: +10V high
- Knobs transform into attenuators when using CV inputs
- Toggle switches transform into "mute" switches when using GATE and TRIGGER inputs

Any signal can be patched into the WORLD CORE's CV, GATE, and TRIGGER inputs so long as it doesn't exceed EURORACK power levels (-12V to +12V). In order to get the best results however, it's useful to attenuate and/or rectify signals to fit the WORLD CORE's responsive range (i.e., the range of voltages which produce noticeable effects).

The responsive range for CV inputs is 0-5V by default. If the CV RANGE JUMPER has been configured for LZX compatibility (see pg. 4), then the responsive range is 0-1V.

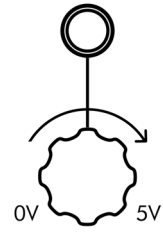
GATE and TRIGGER inputs are set to a +0.5V logic threshold. Any level below +0.5V is considered a logical low ("OFF"), and anything above +0.5V is considered a logical high ("ON").

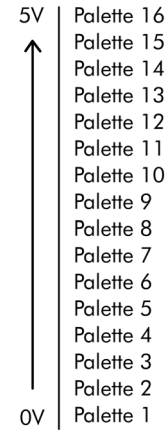


PALETTE

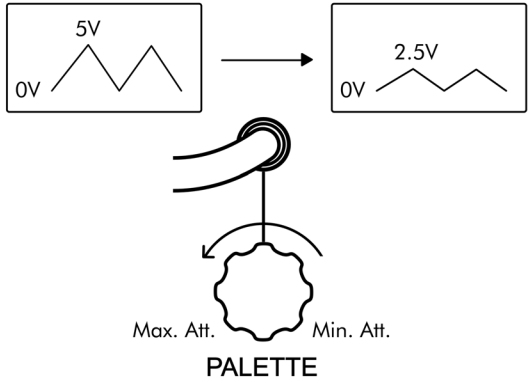
Most of the WORLD CORE's parameters are accessible through manual controls (knobs and toggle switches) as well as jack inputs (for use with CV, GATE, or TRIGGER signals). TILE 1's PALETTE parameter, for instance, consists of a manual knob and CV input jack. Both the knob and the jack modify the same parameter—the set of colors used to render TILE 1. Interface elements that affect the same parameter are depicted on the panel with a vertical connecting line.

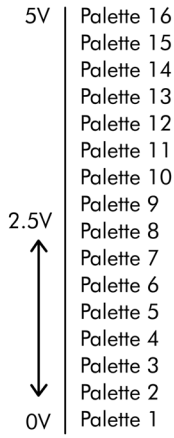
Knobs generate 0V when turned fully counterclockwise, and +5V (or +1V if configured for LZX) when turned fully clockwise. The number of states mapped to the voltage depends on the particular parameter. In the case of the PALETTE parameter, the 0-5V range indexes 16 unique PALETTEs.





When a signal is patched into the corresponding CV input, the knob no longer directly controls the parameter. Instead, it becomes an attenuator that scales the voltage at the CV input. When the knob is fully clockwise no attenuation is applied. Attenuation gradually increases as the knob is turned counterclockwise.

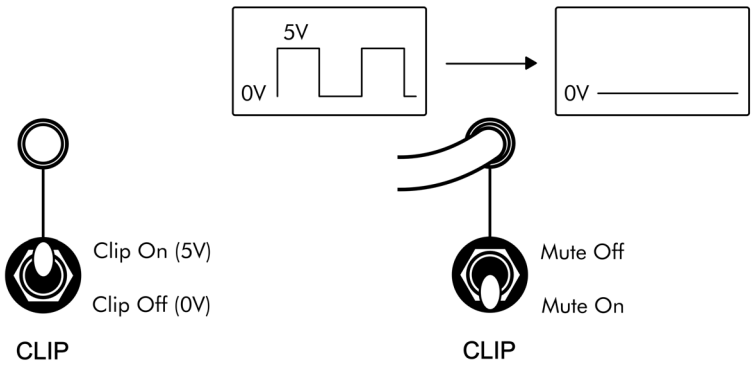




By adjusting the amount of attenuation, it's possible to format higher voltage signals to match the WORLD CORE's 0-5V responsive range, or to truncate the number of selectable states as shown above. Note that attenuation only affects the top end of the voltage

range. Additional (third party) modules are required to raise the low end of the range (DC offset summing), or to eliminate negative voltage (rectification).

Toggle switches usually appear with a corresponding GATE or TRIGGER input. Except for the SPEED, RESET, and SELECT toggles (which we'll look at in detail later) all toggles generate +5V when switched upwards ("ON"), and 0V when switched downwards ("OFF"). Unlike knobs, which select between several states, toggles are used for binary parameters.



Parameters can also be controlled by patching GATE or TRIGGER signals into the jack input. Just like knobs, toggles don't directly control parameters when their input is in use. Instead, they become "mute" switches that pass or block the signal patched into the input. The down position turns the mute "ON", overriding the signal with 0V, while the upward position turns the mute "OFF", allowing the signal to control the parameter.

If a parameter is not responding to external signals, the first thing to check is the knob or toggle position. Knobs should be fully clockwise and toggles should be placed upwards to avoid attenuating or muting signals.

The WORLD CORE's seven COLLISION outputs (along with the RESET OUTPUT) generate +10V GATEs. A relatively high voltage is desirable for GATE/TRIGGER outputs because it allows them to be patched to multiple simultaneous locations without experiencing problematic voltage drops. Each COLLISION output has a corresponding LED that lights to indicate its state.

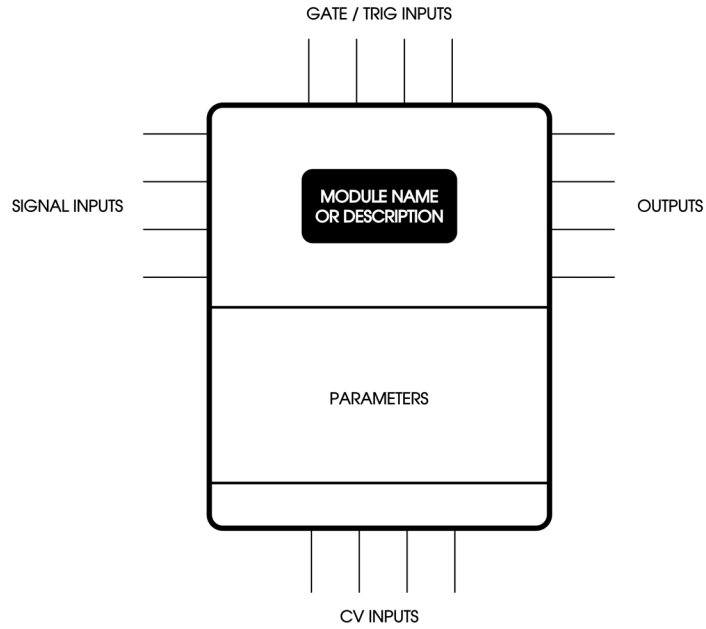
S1 / S2

HOW TO READ PATCH SCHEMATICS

Beginning in CHAPTER 3, the USER'S GUIDE will often provide visual patch schematics to illustrate the WORLD CORE's various features. Unfortunately, there is no universal standard for synthesizer patch notation like there is for electronic schematics. Many different systems have been proposed and used over the years, with the most popular ones having being developed during the 1970s and early 1980s.

Special Stage Systems has developed a new notational system that we hope may be of use not only for MING MECCA, but for patch documentation generally. We've tried to keep the best aspects of the '70s and '80s systems, while modernizing those elements that have become archaic or untenable in the context of modern modular synthesis.

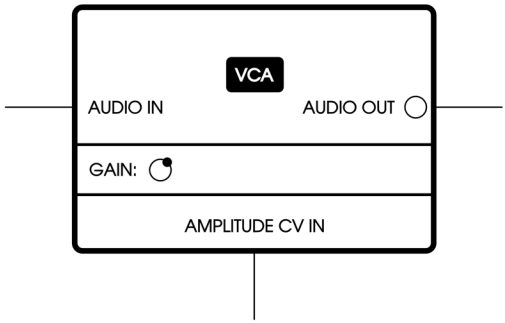
Modules are represented by rectangular boxes. When a specific module is needed, its name appears in the center of the box; if only a generic type is required, a general category will be listed instead, such as "VCA" or "STEP SEQUENCER."



Patch cord connections are represented by lines terminating at the edges of the box. Each side of the box is reserved for different categories of connections: Outputs always begin on the right side; CV (analog) inputs on the bottom side; GATE/TRIGGER (digital) inputs on the top side; and signal inputs (any voltage that will be processed and sent to an output) on the left side. Because outputs always appear on the left side of the boxes, and inputs always appear on the top, bottom, or left sides, there is no need to use arrows to indicate signal flow; directionality is instead indicated structurally.

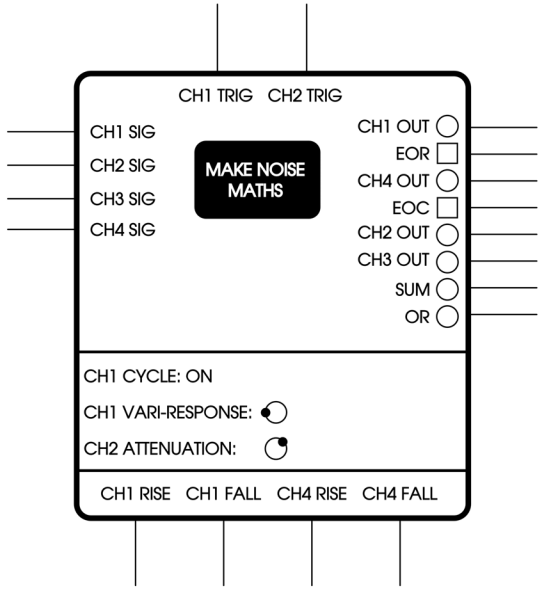
Inputs and outputs are labeled along the edges of the box, and outputs appear alongside corresponding symbols that indicate their waveform. The internal box labeled "parameters" is used to record state information about the module, such as knob and toggle positions. Let's look at a couple concrete examples to see the system in action.

A standard VCA with an audio input, audio output, CV input for amplitude, and a manual gain control would be drawn as follows:

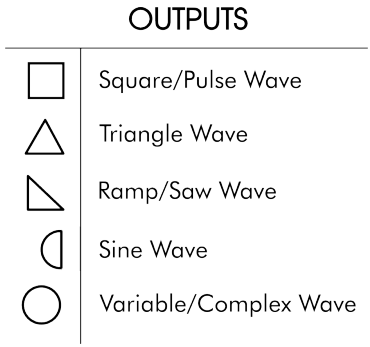


The position of the gain knob is noted graphically in the parameters box, and the inputs and outputs are labeled along the edges. The audio output is accompanied by a circle, which represents a variable or complex waveform. Since this VCA doesn't have any GATE or TRIGGER inputs, the top side of the module is unlabeled and has no connecting lines.

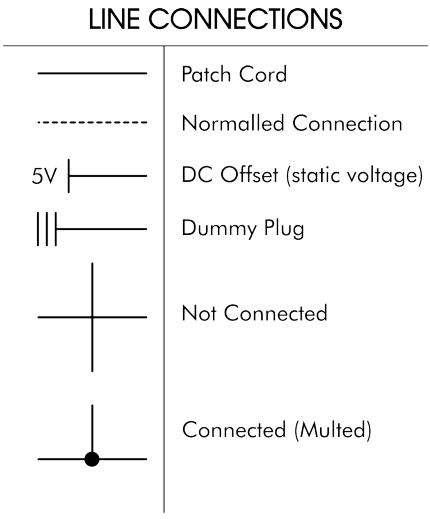
Now let's look at a more complex module: the **Make Noise MATHS**, a popular function generator.



MATHS has signal, TRIGGER, and CV inputs, as well as a variety of outputs, so all four sides of the box are used. Various module states are noted in the parameters box, including the position of the cycle toggle on Channel 1. GATE outputs like end-of-rise (EOR) and end-of-cycle (EOC) are accompanied by a square symbol. The full legend of possible output symbols is shown to the right.

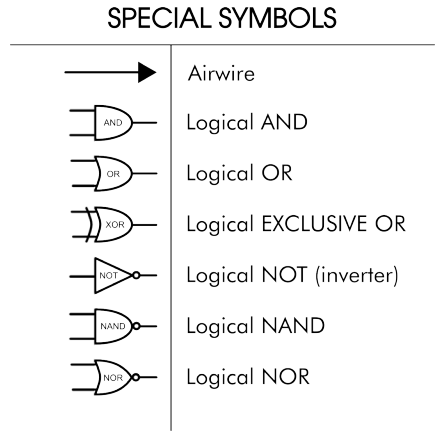


Square, triangle, and saw waves are represented by their corresponding shapes. Pure sine waves are represented by a half circle. All other wave types are represented by a full circle, including noise waveforms, complex waveforms such as those generated by FM or wavetable oscillators, outputs where the precise wave shape is variable, unknown, or unimportant, and high frequency signals such as video.

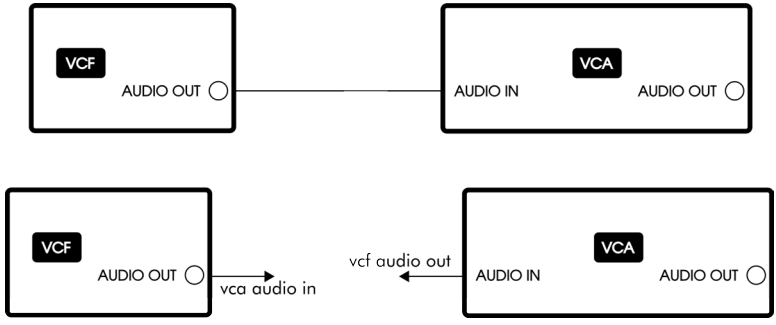


When multiple modules are patched into one another, standard patch cord connections are shown as solid lines. Normalled connections are shown as dotted lines. DC offsets appear as a single vertical bar, and dummy plugs as three vertical bars. Multed connections are shown as branching lines joined by a small dot. When no dot is present, the lines are simply overlapping visually but do not imply an electrical connection.

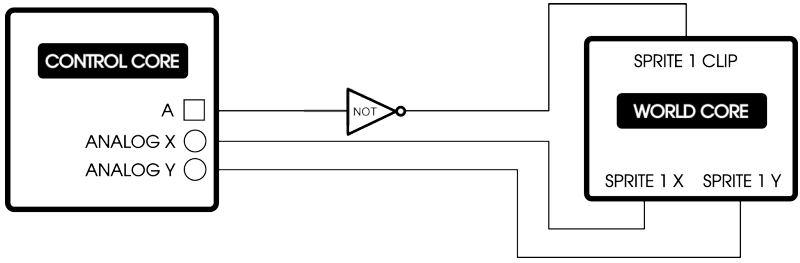
Sometimes a patch schematic can have so many modules and overlapping connection wires that it becomes difficult to read. To reduce the number of individually drawn modules, commonly used logic functions are drawn in-situ as individual symbols; it's left up to you to select and patch the proper combination of logic modules to realize the patch.



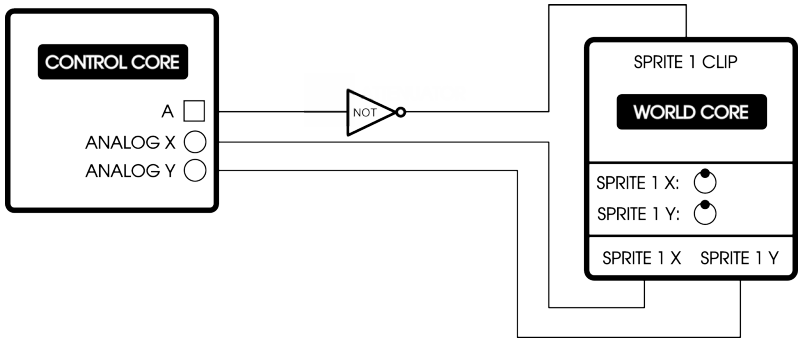
In schematics with very dense signal flow, certain connections may be shown by an "airwire" instead of a normal solid line. Airwires are pairs of solid lines that terminate in arrows. Each arrow has the name of its paired connection written next to it. For instance, the two patches below are equivalent:



Additionally, only inputs and outputs that are in active use are included within the module boxes. In the following simple patch, the majority of the WORLD CORE and CONTROL CORE I/O is not shown in the schematic; only the connections relevant to the patch are labeled.



Knobs and toggles at signal destinations are assumed to be fully open unless otherwise noted. In the above patch for instance, the SPRITE 1 X and Y knobs are fully clockwise (0% attenuation) and the SPRITE1 CLIP toggle is in the upward or "ON" position. Non-default positions are simply noted in the parameters box along with the rest of the controls, as shown in the modified version below:



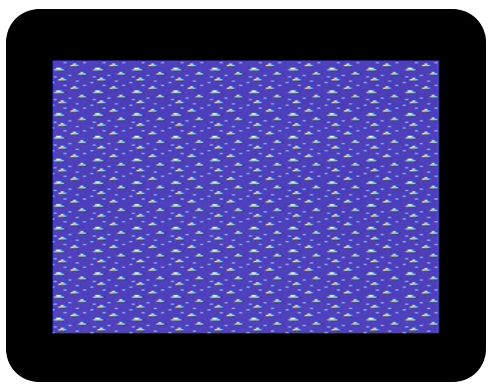
Since the CONTROL CORE's analog X and Y outputs generate 0-10V signals, the modified patch schematic notes that the SPRITE 1 X and Y knobs should be at high noon to achieve 50% attenuation, scaling the voltage to the WORLD CORE's 0-5V responsive range.

RETURNING TO DEFAULT SETTINGS

With so many system parameters, it's easy to encounter WORLD CORE states where things don't behave as you might expect them to, especially when you're just getting started. To avoid confusion, the USER'S GUIDE will occasionally ask you to return the WORLD CORE to "default settings" before constructing a patch or starting a tutorial.

To return to default settings, perform the following steps:

1. Remove all patch cords
2. Place the SPEED toggle in the center position
3. Place the SELECT toggle in the upward position
4. Place all other toggles in the downward position
5. Turn all knobs fully counterclockwise
6. If present, eject any SD CARD from the CARTRIDGE SLOT
7. Pull the RESET toggle down and release to reboot the system



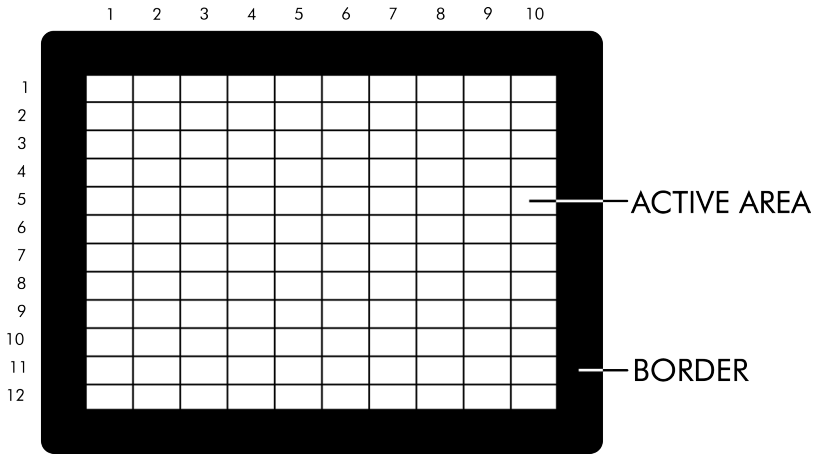
If you've followed the directions correctly, your display should now show only white clouds on a blue sky, with both SPRITEs off-screen.

3. TILES

- Overview
- Data Structure
- Building Simple Environments
- **MAPs**
- **GLITCH MODEs**
- **DYNAMIC MAP DESTRUCTION (DMD)**

OVERVIEW

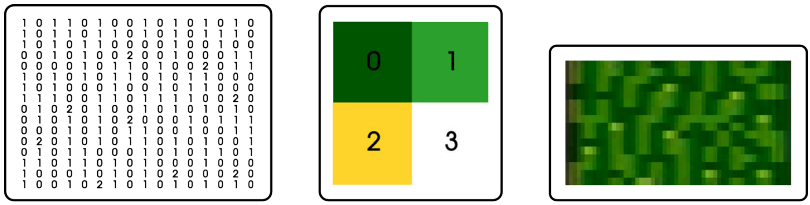
The WORLD CORE graphics system is divided into two primary subsystems: TILE GRAPHICs and SPRITE GRAPHICs. SPRITEs are small graphical objects that can be freely positioned anywhere on the display, and will be introduced in the next chapter. TILEs are rectangular graphical blocks that repeat in grid patterns to build large scenes, architecture and terrain. A good way to understand the difference between the TILE and SPRITE subsystems is to think of TILEs in terms of building an environment, and SPRITEs in terms of populating that environment with avatars, beings, or other objects.



The display consists of an active area with a 160 x 192 pixel resolution, surrounded by an inactive black border. The active area is divided into 120 locations, organized in a 10 x 12 grid. Each grid location is further subdivided into a 16 x 16 pixel bitmap displaying one of two currently selected TILEs. In order to understand how these TILEs are selected, rendered, and organized into larger patterns, we will now take a look at how the WORLD CORE processes graphical data.

DATA STRUCTURE

TILE DATA is composed of 16 x 16 pixel bitmapped images. Each pixel in the bitmap is stored as a number ranging from 0-3, which represents one of four different colors. TILE DATA is combined with a PALETTE to render the final TILE. PALETTEs contain color definitions that map each number in the grid to a different color; in this way the graphical structure of the TILE and its color are logically decoupled, allowing the TILE coloring to be changed dynamically.

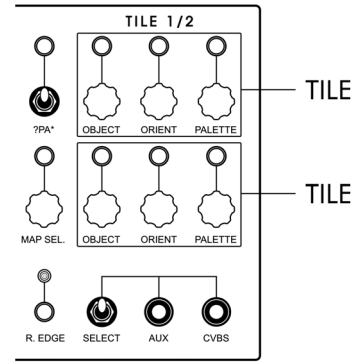


The WORLD CORE can store a total of 32 TILE DATA bitmaps and 16 PALETTEs, which are loaded either from a user-created bank (if a WORLD PACK is loaded into the CARTRIDGE SLOT) or from the default internal memory. Although the complete contents of the TILE and PALETTE data banks are always accessible, the WORLD CORE can only render two TILEs onscreen at any given time.

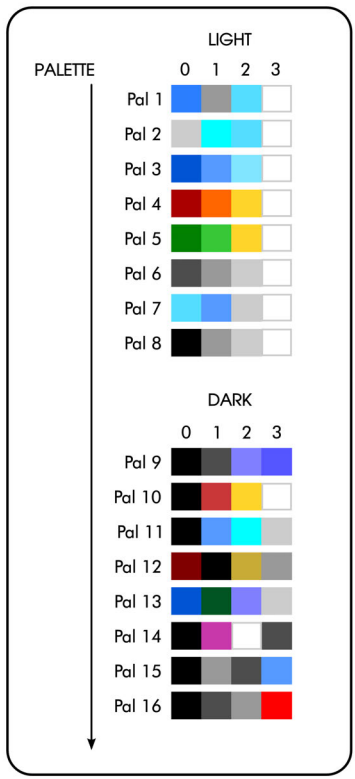
These TILEs are referred to as TILE 1 and TILE 2 respectively, and are best thought of as "containers" that are filled with TILE DATA and colored by PALETTEs. TILE DATA is selected using the OBJECT and ORIENT knobs, and PALETTEs are selected using the PALETTE knob. TILE 1 and TILE 2 each have a dedicated set of controls, color-coded in YELLOW (TILE 1) and BLUE (TILE 2).

Return your WORLD CORE to the default settings, so that the entire display is filled with white clouds and blue sky. Now try changing the TILE DATA for TILE 2 by turning its OBJECT and ORIENT knobs. You

should see the bottom 3rd of the screen change from white clouds to a variety of other patterns. Once you find a pattern you like, try experimenting with the **PALETTE** knob to see how different coloring options affect the **TILE DATA**. If you'd like, feel free to change the **TILE 1** settings as well, or patch a signal into the corresponding **CV** inputs to animate the bitmap patterns and colors.



PALETTE TABLE



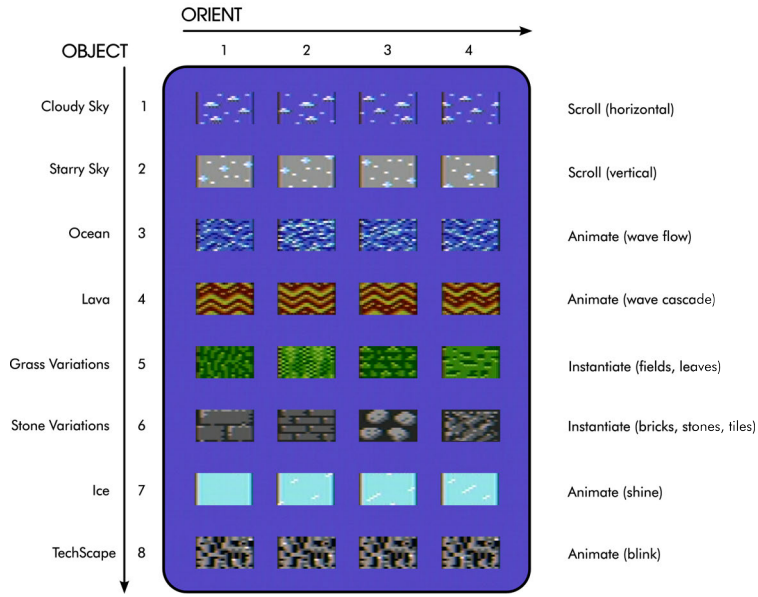
The default **PALETTEs** are loosely organized into two groups of eight ("light" and "dark"), and are shown in the table to the left. All 16 **PALLETES** are indexed using the **PALETTE** knob or a **0-5V** control voltage applied to the **CV** input. Note that although the **WORLD CORE** contains two **PALETTE** knobs—one for **TILE 1** and one for **TILE 2**—they both draw from the same data bank of 16 **PALETTEs**. Think of each **TILE** as an "instance," rendered using different elements from the same underlying data set.

The same is true for the bitmap itself; **TILE 1** and **TILE 2** may each be "filled" independently using their individual **OBJECT** and **ORIENT** controls, but they both refer-

ence the same TILE DATA bank of 32 bitmaps.

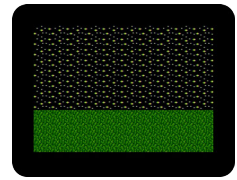
Whereas PALETTEs are indexed using a simple 0-5V control voltage, TILE DATA is indexed using a pair of control voltages that scan through a 2-dimensional array. The position of the OBJECT and ORIENT knobs (or their CV inputs) is multiplied to select a bitmap from the array, as shown below.

TILE TABLE



The OBJECT CV has eight positions, while the ORIENT CV has four. When multiplied together they specify a single position in the TILE table. In the default world, the table is organized so that the OBJECT CV specifies a broad "category" of TILE, while the ORIENT CV selects states within that category. Depending on the TILE DATA, those states can be frames in an animation, scroll positions, or variations on a theme, and are noted to the right of the table.

BUILDING SIMPLE ENVIRONMENTS



Following the steps below, see if you can build a simple environment by selecting specific TILEs from the table.

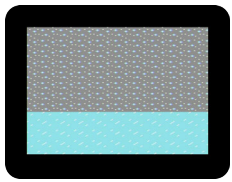
1. Use the TILE 1 controls to select the STARRY SKY TILE. Starting with the OBJECT knob fully counterclockwise, slowly turn it to the right until the CLOUDY SKY TILE (position "1") is replaced by the STARRY SKY TILE (position "2").

2. Finish building the sky by slowly turning the TILE 1 PALETTE knob until you reach PALETTE 10, producing a black background with white and yellow stars.

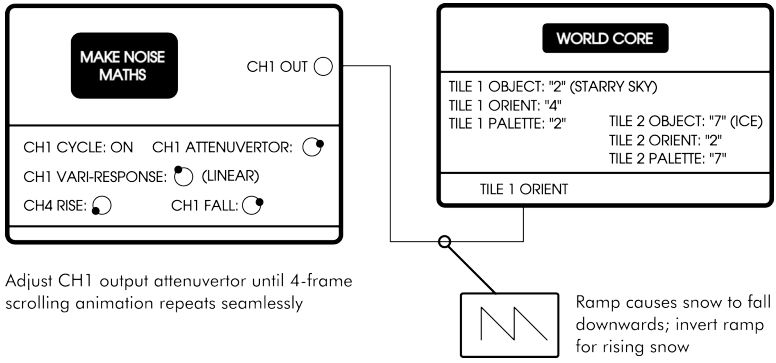
3. Select the first GRASS TILE by setting TILE 2's OBJECT knob to "5" and ORIENT to "1"

4. Use TILE 2's PALETTE knob to select PALETTE 5. You should now see a green field underneath a star-filled nighttime sky.

PALETTE selection is just as important as TILE DATA selection when constructing environments, and simple color changes can have huge effects. Try turning the TILE 1 PALETTE knob to select PALETTE 2, a color set composed of greys and light blues. The stars now resemble snowflakes suspended in a cloudy winter sky. Complete the scene by selecting one of the ICE TILEs for the terrain, and color it using PALETTE 7.



Turning the TILE 1 ORIENT knob will scroll the snowflakes a few frames up and down across the screen. Let's automate this using a control voltage to create the appearance of falling snow.

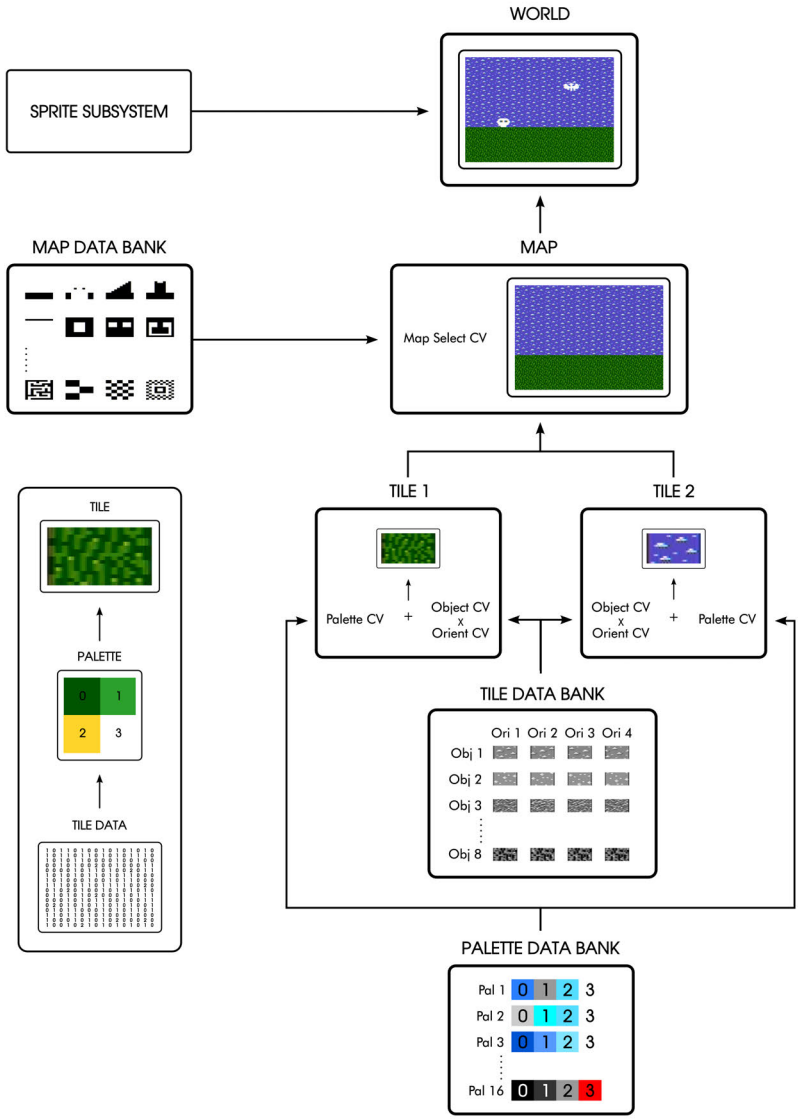


By building the patch above, the four frames indexed by the **ORIENT CV** can be animated in a continuous loop, giving the impression that the snow is falling infinitely. If you don't own a **Make Noise MATHS** you can substitute any standard **LFO** or cycling **ENVELOPE** generator, so long as it has a **0-5V** range and you can control the phase of the ramp (i.e., set a fast attack and slow decay, or vice versa).

The **TILE 1 ORIENT** knob will transform into an attenuator when patching into its **CV** input, so make sure the knob is fully clockwise to accept the full range of voltage. Once you've patched in the function, set the frequency to taste, and adjust the **MATHS** Channel 1 attenuvertor (or **WORLD CORE ORIENT** knob if using a module without built-in range control) until the snow falls smoothly. The key is to minimize the amount of "pause" during the last frame of the animation before he start of the next ramp cycle. If you run into trouble, double-check that the **MATHS** vari-response is set to linear; non-linear shapes can lead to choppy or asymmetric animation cycles.

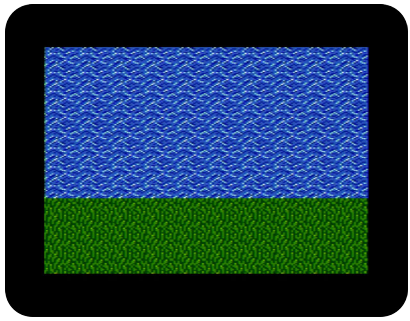
Inverting the phase of the ramp will make the snow rise instead of fall, and changing the shape to a triangle will make the snow move up and down in a loop. Experiment with these parameters, as well as the frequency, to get an idea of the range of possible effects.

TILE SUBSYSTEM



MAPS

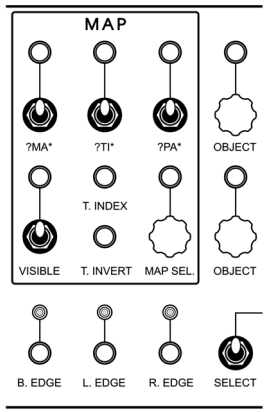
Keeping the same patch from the previous section, change **TILE 1** to display the **OCEAN TILE SET** (**OBJECT** position "3"), and color it using **PALETTE 3**. Set **TILE 2** to display the first **TILE** of the **GRASS VARIATIONS** (**OBJECT** position "5") and color it with **PALETTE 5**.



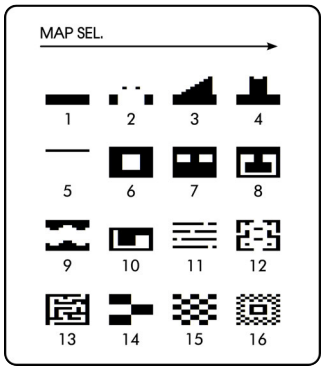
You should now see a grassy bank that gives way to a blue ocean with animated waves. Notice how a simple **TILE** swap has completely changed the world's presentation: when the upper portion of the screen was filled with a sky **TILE**, it seemed as if we were looking out across a terrain towards a vanishing point; now that it's filled with water, the perspective has shifted to a bird's-eye view, as if looking down on a coastline from above.

Another way to play with the world's physical structure is through direct topographical manipulation. The **WORLD CORE** stores 16 **MAPs** that determine how **TILE 1** and **TILE 2** are patterned across the display.

> You may have noticed that the **OBJECT** and **PALETTE** positions are often synchronized—**OBJECT 1** is paired with **PALETTE 1**, **OBJECT 2** with **PALETTE 2**, and so on. This is due to a deliberate feature of the **WORLD CORE's** built-in graphics. Each **TILE SET** has been designed with an associated "default" **PALETTE**, such as green colors for grass (**OBJECT 5 / PALETTE 5**) or blue shades for water (**OBJECT 3 / PALETTE 3**). There are exactly twice as many **PALETTEs** (16) as there are **TILE SETS** (8), with **PALETTEs 9-16** representing an alternative set of defaults built out of darker shades, useful for building nighttime environments.



MAP TABLE



Find the MAP section (located to the immediate right of the TILE 1/2 section) and try turning the MAP SELECT knob. You should see the ocean transform into a variety of lakes, ponds, rivers, streams, and other structures.

The table above shows all 16 available MAPs. Like PALETTEs, MAPs are indexed by a single control voltage that sweeps from 1 to 16 as the voltage increases. Also like PALETTEs, MAPs are not named, but are referred to by number alone.

GLITCH MODES

Now turn your attention to the toggle switches located at the top of the MAP section. These toggle switches control three independent GLITCH MODEs that replace the contents of MAP, TILE, and PALETTE memory with arbitrary data. The toggles are labeled as follows:

?MA* — MAP glitches
?TI* — TILE glitches
?PA* — PALETTE glitches

Try flipping the ?MA* toggle upwards (to the "ON" position). You should now see the currently selected MAP change into a chaotic new pattern. Flipping the toggle back down will turn off the GLITCH MODE, returning the MAP to its normal state. Every time the GLITCH MODE is *cycled* (moving from an "OFF" to an "ON" state) a new chaotic pattern is loaded into memory.

The ?MA* toggle affects not only the currently selected MAP, but all 16 MAPs in memory. With ?MA* toggled "ON," turn the MAP SELECT knob to see the variation of patterns possible within a single set of glitched MAPs. Try cycling ?MA* a few times, auditioning the full set of MAPs each time. You may notice that certain sets have some similarity between successive MAPs, following a visual logic or even appearing to coarsely "scroll" structures when turning the MAP SELECT knob. This is because the MAPs generated by ?MA* are—despite their chaotic appearance—not actually random, but generated according to a highly structured process.

During normal operation, the WORLD CORE's MAP rendering system looks to a particular segment of RAM where MAP DATA is stored. Engaging ?MA* causes the rendering system to look at random areas of RAM instead, including areas occupied by the WORLD CORE's own firmware. In effect, glitched MAPs are actually a visual representation of the WORLD CORE's live program code, offering a self-reflexive glimpse of its own inner workings. The ?TI* and ?PA* modes work in the same way, using random segments of the WORLD CORE's RAM to generate TILE DATA and PALETTE DATA, respectively.

> Because the program code is visualized live, you may come across glitched MAP DATA or TILE DATA that is animated. This animation reflects real processes occurring in the WORLD CORE's firmware. Note that PALETTE DATA is derived from a more complex process, and does not reflect live program code, but only the state of code at the moment that ?PA* is cycled.

Just like the ?MA* toggle, ?TI* and ?PA* populate their entire memory banks with each cycle, generating 32 TILE DATA bitmaps and 16 PALETTEs. Try adding glitched PALETTEs to standard MAPs and TILEs, or turn all three modes on simultaneously to generate fully glitched screens.

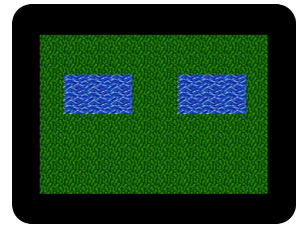
DYNAMIC MAP DESTRUCTION (DMD)

MAP DATA can also be modified in realtime via the TILE INDEX and TILE INVERT inputs. TILE INDEX sweeps through each TILE location consecutively, beginning in the lower-left corner of the display. Sending a TRIGGER to TILE INVERT will invert the TILE DATA at that location (i.e., swap TILE 1 settings for TILE 2 settings and vice versa).

We call this process DYNAMIC MAP DESTRUCTION, or DMD for short. There are two main uses for DMD:

1. Editing maps by hand using the CONTROL CORE or some other interface module, usually as part of a "patch setup" that makes permanent and static geographical changes.

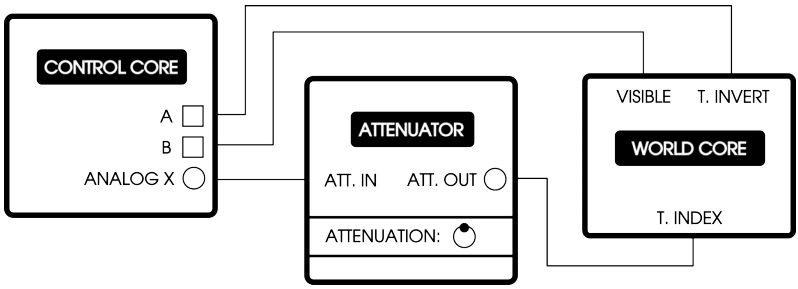
2. using control voltage sources to "animate" terrain procedurally, in order to create appearing/disappearing platforms, bridges, doors, rooms, islands, and so on.



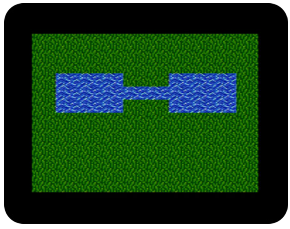
Let's start out by building a "manual" DMD patch using the CONTROL CORE. If you don't have a CONTROL CORE, you can substitute an analog joystick and manual TRIGGER source, or any other combination of modules providing a DC offset and button or switch-style digital output.

First, call up the scene shown above by setting your WORLD CORE to MAP 7. Fill in the terrain with green grass and the ponds with blue water.

Now construct the following patch. The CONROL CORE's ANALOG X output generates a 0-10V signal; in order to scale it to the 0-5V responsive range of the TILE INDEX input, we need to run it through an attenuator set to 50% attenuation. The B and A GATE outputs are patched directly into the VISIBLE and TILE INVERT inputs, respectively.

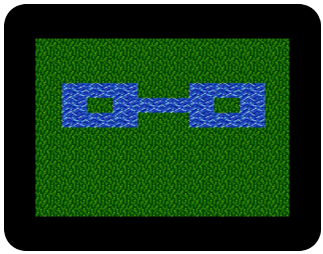


The VISIBLE toggle / GATE input controls a highlight that displays the currently selected TILE location, useful for previewing which TILE you are about to invert. In our simple patch, the VISIBLE toggle is patched directly to the CONTROL CORE's B output, meaning that in order to view the highlight, B must be held down on the gamepad. This allows the highlight to be engaged only when actively modifying the MAP, disappearing when not in use.



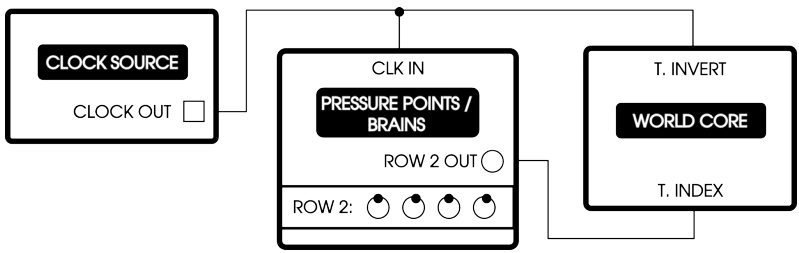
Try holding B while moving the D-PAD LEFT and RIGHT to sweep through the TILE locations. If the highlight is moving too quickly, try pressing SELECT to slow down the motion of the CONTROL CORE's ANALOG X output; if it's moving too slowly, press START to speed it up.

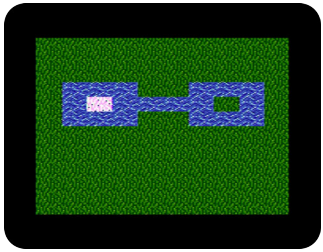
Now let's join the ponds together as shown above. Position the highlight over each TILE and press A to convert the grass into water.



Use the same process to place a small island in each pond, as shown. When you're finished, try turning the MAP SELECT knob to another MAP, and then return to MAP 7. Note that the changes made to MAP 7 are preserved in memory. Changes persist until the WORLD CORE is power cycled or rebooted using the RESET toggle switch.

By using a STEP SEQUENCER such as the **Make Noise** PRESSURE POINTs / BRAINs, we can create islands that appear and disappear in a cyclical pattern. Leaving the WORLD CORE set to your modified MAP 7, remove all patch cords and construct the following patch. Leave the clock source unpatched (or switched off) for now.





The ROW 2 knob positions are approximate: the idea is to have each step of the sequence invert one of the island TILEs. To set up the sequencer, turn the WORLD CORE's VISIBLE toggle to "ON", and move the first step (ROW 2, STEP 1) until the highlight is on top of the leftmost island. Then move the the second step (ROW 2, STEP 2) until the highlight is on top of the rightmost island. Since PRESSURE POINTs is a 4-position STEP SEQUENCER, but we're only modifying two islands, we'll set up steps 3 and 4 to simply repeat the pattern of steps 1 and 2. Set the third

step (**ROW 2, STEP 3**) to the leftmost island, and the fourth step (**ROW 2, STEP 4**) to the rightmost island.

Now set your clock source to a low (control-rate) frequency, and patch the output to both the **PRESSURE POINTs CLK** input and the **TILE INVERT** input as shown in the patch diagram. The **VISIBLE** toggle maybe turned off or on according to taste. You should now see the two islands appear and disappear sequentially, at a rate determined by the clock source.

See if you can expand the patch by adding some percussive sound effects to the disappearing islands, using the clock source to trigger an **ENVELOPE / VCA**, and the **ROW 2** outputs to alter an oscilator's pitch.

Because the **TILE INDEX** input is indexing 120 unique **TILE** locations, the "voltage resolution" is very high. This can make it somewhat difficult to perfectly select each island by hand. Additionally, small drifts in ambient temperature can occasionally nudge the **TILE** location to the left or right. Make sure your modular is fully warmed up, take your time, and readjust if necessary.

As you experiment with clock speed, you may find that fast clock rates cause the "order" of the island pattern to change. This is due to **TILE** locations failing to invert on a given step. The **TILE INDEX** input uses active processing to prevent false triggering, but the tradeoff is that extremely fast operation can sometimes result in errors. For best result, keep clock speeds below 10 Hz.

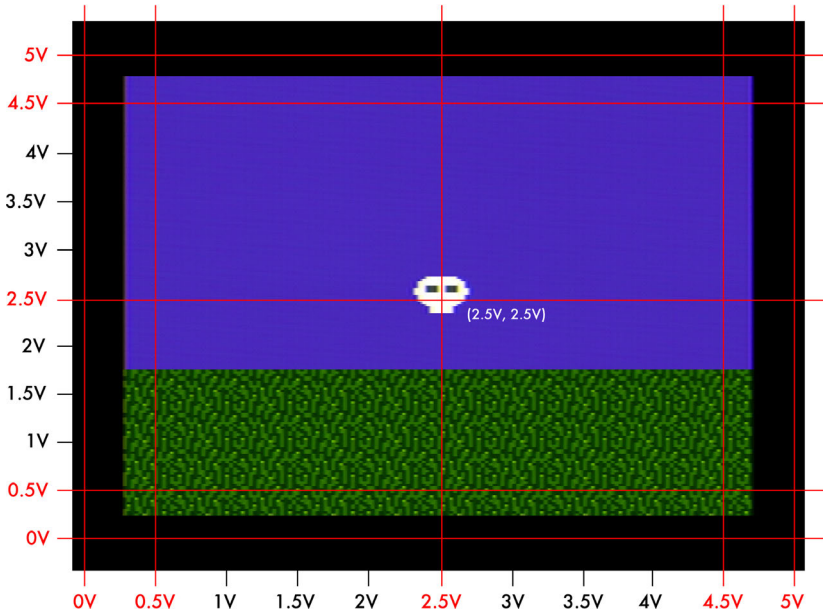
4. SPRITES

- Overview
- Data Structure
- CLIPPING
- COLLISION
- SPRITE 1: DIRECTIONAL INPUTs
- SPRITE 1: GRAVITY

OVERVIEW

The SPRITE GRAPHICs subsystem allows your environments to be populated with autonomous or player-controlled objects. SPRITEs are essentially small moveable pictures, and are similar to TILEs in their basic data structure. Instead of being repeated in grid patterns however, they are freely positioned on the screen at arbitrary, dynamic coordinates. These coordinates are specified by X and Y control voltages that move in 1-pixel increments. At the extreme ends of the X and Y CV range, SPRITEs can be placed behind the screen borders, hiding them from view.

SPRITE POSITION: VOLTAGE GRAPH

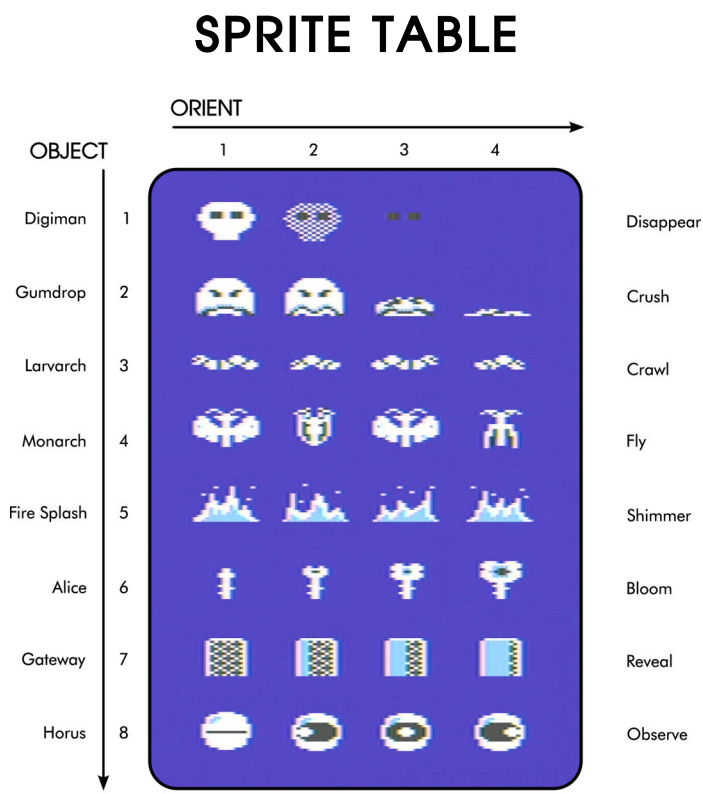


The WORLD CORE contains two independently addressable SPRITEs: SPRITE 1 and SPRITE 2. SPRITE 1 features some special attributes not

available to SPRITE 2, such as relative GATE-based DIRECTIONAL INPUTs and GRAVITY simulation. These special attributes are discussed at the end of the chapter.

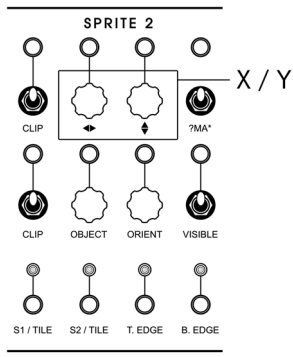
DATA STRUCTURE

Just like TILE DATA, SPRITE DATA is organized in a 2-dimensional array and accessed by OBJECT and ORIENT control voltages.



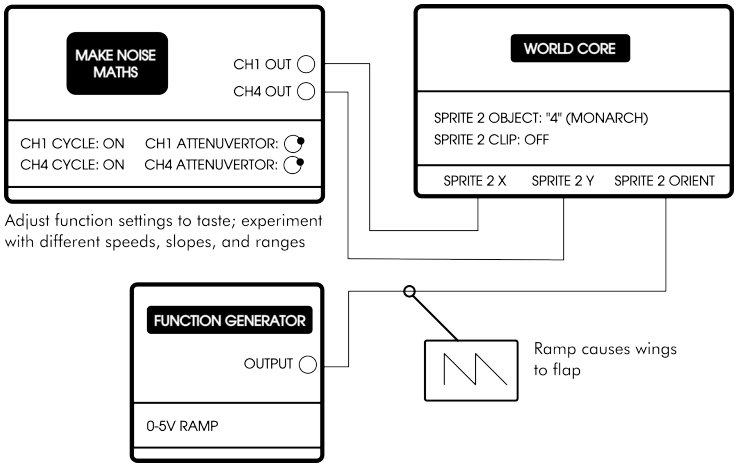
Each SPRITE object contains four frames of animation indexed by the ORIENT CV. These animation behaviors are noted to the right of the table.

Return the WORLD CORE to default settings, making special note of the CLIP toggles, which should be in the "OFF" position. Now locate the SPRITE 2 area of the control panel and use the X and Y knobs to reposition the SPRITE. Note that at the extreme boundaries of the X and Y ranges, the SPRITE is hidden behind the black borders, allowing it to be quickly and easily hidden or "turned off."



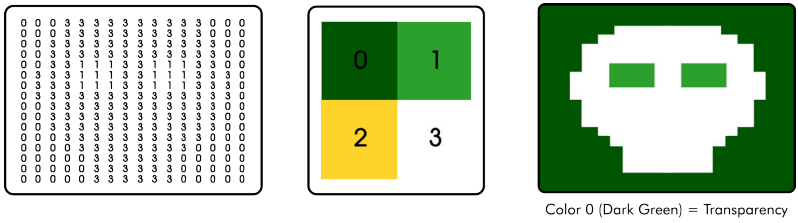
Position the SPRITE in the center of the display by setting both the X and Y knobs to approximately high noon. Just like we did with TILEs, try using the OBJECT and ORIENT knobs to audition the SPRITE DATA. Select the MONARCH SPRITE (OBJECT position "4"), and then patch an LFO or cycling ENVELOPE into the ORIENT CV input to animate its wings.

Now try using a pair of control voltages to animate the SPRITE's position using the X and Y CV inputs, as shown below.



Once you've gotten acquainted with the range of movements and patterns, disconnect the function generators to regain manual control, and return to the default DIGIMAN SPRITE (OBJECT position "1"). Call up MAP 6, filling the central box with gray bricks and the outer boundaries with blue water. Position SPRITE 2 so that DIGIMAN is contained within the box of bricks. Now use the X knob to slowly sweep DIGIMAN's position left and right across the screen.

Pay close attention to DIGIMAN's eyes. See how they change color depending on the underlying TILE? This is because SPRITEs do not have dedicated PALETTEs; instead they "inherit" the PALETTE of whatever TILE they are currently on top of. The WORLD CORE's default graphics data has been designed so that the main SPRITE color (white) goes mostly unchanged across PALETTEs, but depending on the currently selected SPRITEs, PALETTEs, and especially the WORLD PACK design, SPRITE bodies may change color when moving across TILE 1 and TILE 2 boundaries. With a little practice, this can be utilized as an expressive effect.



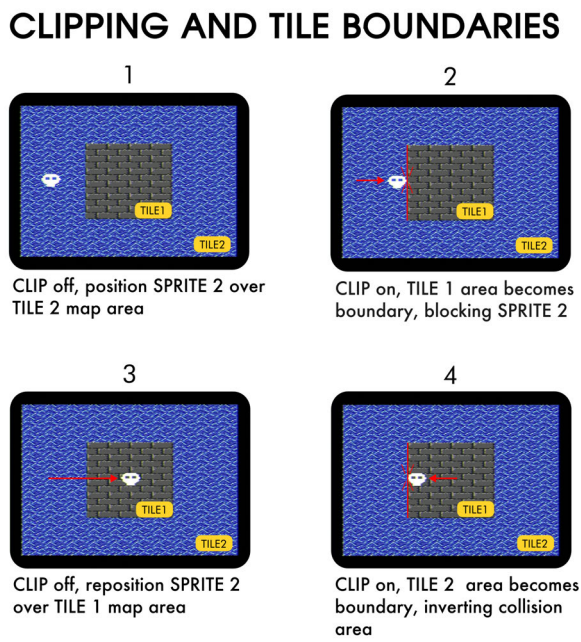
Try filling the box with different PALETTEs and see how they alter DIGIMAN and other SPRITEs. The effects will be more drastic on some SPRITEs than others, depending on which PALETTE color (1, 2, or 3) dominates the SPRITE. The FIRE SPLASH SPRITE, for instance, makes extensive use of Color 2, while DIGIMAN doesn't use Color 2 at all.

When you're done, leave the WORLD CORE on MAP 6; we'll use this same patch in the next section.

CLIPPING

So far we've been able to move SPRITE 2 unimpeded across the display. To build more sophisticated worlds however, it is often necessary to define boundary-interactions between SPRITEs and the TILE geographies they inhabit.

The WORLD CORE's integrated CLIPPING and COLLISION system allows us to restrict SPRITE movement to either TILE 1 or TILE 2, transforming the opposing TILE into a boundary or obstacle. Referring to step 1 of the chart below, position SPRITE 2 outside of the brick box, and then turn on the SPRITE 2 CLIP toggle.



Now follow step 2 and slowly sweep SPRITE 2's X position from left to right, noting that the SPRITE gets "caught" on the TILE 1 / TILE 2 boundary, preventing it from entering the TILE 1 area. Although the SPRITE may still move along the Y-axis, its X-axis progression is sus-

pended when it comes into contact ("collides") with the brick box. CLIPPING is contextual, depending on the relative position of SPRITEs and TILEs: if we were to position the SPRITE underneath the box, for instance, its Y-axis would become impeded, while its X-axis would become free.

Move the SPRITE back over the ocean and turn CLIP back off. Following steps 3 and 4, reposition the SPRITE over the brick box, and turn on CLIP again. The TILE boundary relationship has now been inverted, transforming TILE 2 into the obstacle. Since the TILE 2 area forms an enclosure, DIGIMAN's movement is now restricted in all four directions, trapped on the brick terrain.

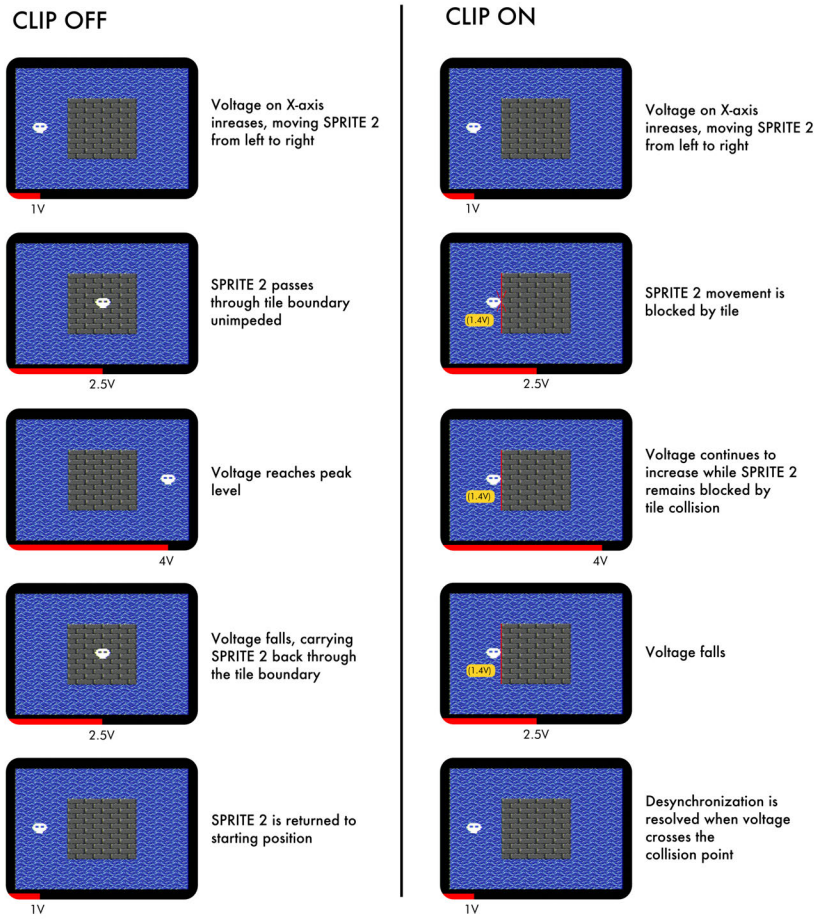
> The CLIPPING system determines which TILE area is an obstacle based on the SPRITE's position when CLIP is engaged: if the SPRITE is over TILE 1, TILE 2 will become the obstacle; if it's over TILE 2, TILE1 will become the obstacle instead.

Moving SPRITEs around with CLIPPING engaged often results in a condition called POSITIONAL DESYNCHRONIZATION, where the SPRITE's displayed position and the X and/or Y voltage drift out of sync with each other. Consider a simple patch in which a cycling ramp slowly moves SPRITE 2 back and forth across the X-axis in an endless loop. If CLIPPING is turned off, the SPRITE will follow the full 0-5V trajectory, moving across the entire screen and hiding behind the borders on either side.

If CLIPPING is turned on however, the SPRITE will treat TILE 1 as a boundary, getting caught at the TILE transition about 1/3 of the way through the screen. The control voltage source, being oblivious to the WORLD CORE's internal clipping states, will continue to increase in voltage until it hits its maximum value of +5V. During this time, the X-axis voltage will specify a position that SPRITE 2 cannot physically access because it's obstructed. The SPRITE will remain frozen until the ramp begins the decay portion of its cycle and lowers to "meet" the SPRITE at its current location. This "meeting point" occurs at the precise location at which the SPRITE's physical position and its voltage

broke sync: in our example, about 1.4V. Once the control voltage source and the SPRITE have been realigned the POSITIONAL DESYNCHRONIZATION is "resolved."

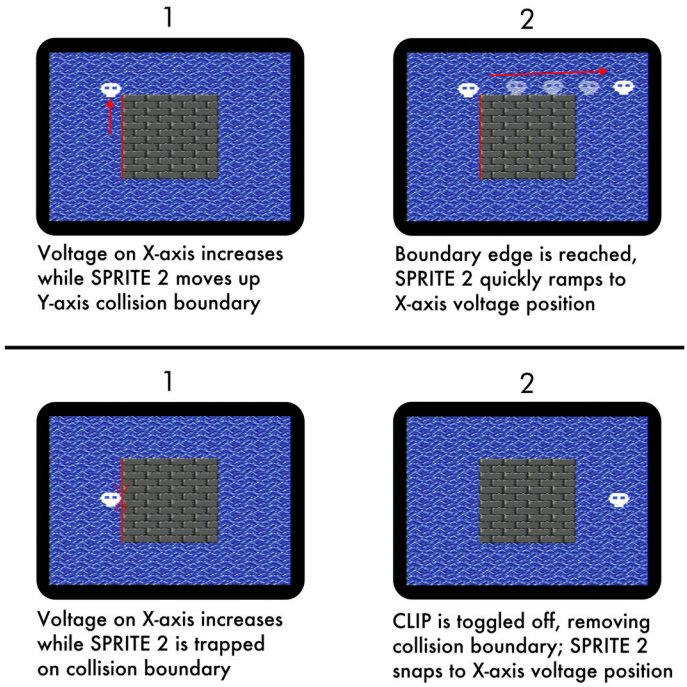
POSITIONAL DESYNCHRONIZATION



There are two additional ways that POSITIONAL DESYNCHRONIZATION can be resolved. The first of these is by removing the CLIPPING obstacle, either through changing the SPRITE's position across the op-

posing axis (in our example, moving the SPRITE up or down so that it "slides" past the TILE), or by erasing the TILE using TILE INDEX and TILE INVERT. The second is by manually cycling the CLIP toggle to defeat the CLIPPING system entirely. Both situations produce the same effect—the return of the SPRITE position to the absolute value specified by the X and/or Y CV input—but with a subtle shift in behavior in each case.

DESYNCHRONIZATION RESOLUTION



Position the SPRITE above the ocean and turn on CLIPPING, then turn the X knob to the right so that the SPRITE is caught up against the brick box. Keep turning the knob until you hit about the 3 o'clock position. Now slowly turn the Y knob to the left, raising the vertical position of the SPRITE. As it passes the edge of the box, you should

see the SPRITE quickly ramp across the screen to meet the X-axis voltage position, as shown in the top half of the chart.

Move the SPRITE back to the left side of the box, once again turning the X knob to about 3 o'clock so that the SPRITE is "loaded" against the box's edge. This time, instead of using Y-axis navigation to resolve the DESYNCHRONIZATION, simply turn off the CLIP toggle instead. Note that instead of quickly moving across the screen, the SPRITE now snaps to the X-axis voltage position instantly.

These two behaviors provide the best of both worlds, giving SPRITE movement a natural look when navigating CLIP-persistent TILE geography, while still retaining full control when manually disengaging CLIPPING.

> Sometimes CLIPPING can be accidentally defeated when modulating a SPRITE's position at very fast speeds, resulting in the SPRITE becoming stuck or "embedded" into the boundary TILEs. This is particularly likely to happen in corners where two diagonal TILEs meet. This is due to a limitation of videogame CLIPPING algorithms. CLIPPING errors of this sort were relatively common in older videogames, often resulting in the player character becoming embedded in the wall or floor, necessitating a system reset. Luckily, resolving these errors on the WORLD CORE is as simple as flipping a switch: simply cycle the CLIP toggle to release the trapped SPRITE and resume normal operation.

COLLISION

While experimenting with CLIPPING states, you may have noticed the row of LEDs at the bottom of the module occasionally changing in response to your actions. The WORLD CORE has seven dedicated

COLLISION outputs that generate 10V GATEs in response to the interaction of SPRITEs and TILEs. The outputs are organized as follows:

S1/S2 — SPRITE 1 collides with SPRITE 2

S1/TILE — SPRITE 1 collides with TILE 1 / TILE 2, as selected by CLIP
S2/TILE — SPRITE 2 collides with TILE 1 / TILE 2, as selected by CLIP

T. EDGE — SPRITE 1 collides with the TOP EDGE of the screen
B. EDGE — SPRITE 1 collides with the BOTTOM EDGE of the screen
L. EDGE — SPRITE 1 collides with the LEFT EDGE of the screen
R. EDGE — SPRITE 1 collides with the RIGHT EDGE of the screen

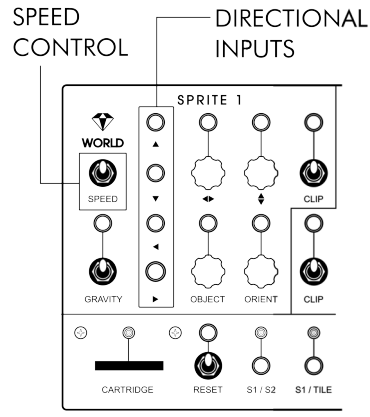
Each output flips high according to different conditions. The S1/S2 output flips high if and only if SPRITE 1 and SPRITE 2 are touching. The S1/TILE and S2/TILE outputs are contextual, depending on the current CLIPPING state. If TILE 1 has been defined as a SPRITE's obstacle, then the corresponding output will flip high when the SPRITE touches TILE 1; ditto if TILE 2 has been defined as the obstacle. The EDGE outputs flip high when SPRITE 1 has hit the extreme boundaries of the screen, behind the black borders.

> The COLLISION outputs are always active, even when CLIP is turned off, and will continue to fire when the SPRITE crosses the boundary last selected by the CLIP toggle. This can be used to determine what region of the MAP the SPRITE is currently in without restricting the SPRITE's movement.

> By modifying the WORLD CORE's CONFIG.TXT, several alternate COLLLISION modes can be selected that change the behavior of the COLLISION outputs. See CHAPTER 5 for more information.

SPRITE 1: DIRECTIONAL INPUTS

So far we've been using SPRITE 2 exclusively. This is because SPRITE 1 has some additional features that make it a little more complex (but much more flexible) than SPRITE 2. SPRITE 1 can do everything that SPRITE 2 can, including X/Y analog position, OBJECT/ORIENT graphical indexing, and CLIPPING. But it is also capable of being positioned via four dedicated DIRECTIONAL INPUTs, as well as placed in a simple GRAVITY simulation.



The DIRECTIONAL INPUT system consists of separate UP, DOWN, LEFT, and RIGHT GATE inputs, combined with a 3-stage SPEED toggle.

The CV-based X/Y system we've been using so far is an "absolute" positional system, meaning that no matter where the SPRITE is currently located, applying 2.5V to the X- and Y-axis will always position the SPRITE at the center of the screen, and so on across the range of voltages. The DIRECTIONAL INPUT system is, by contrast, a "relative" positional system, moving SPRITE 1 according to fixed increments relative to its starting location.

When the WORLD CORE detects a GATE signal at SPRITE 1's UP input, for instance, it will begin incrementing SPRITE 1's Y-axis position from its current position. SPRITE 1 will continue to move upwards until the GATE at the UP input disappears, at which point it will come to rest at its new location. The rate at which SPRITE 1's position is changed is controlled by the SPEED toggle, which selects between slow, medium, and fast speeds. The default medium SPEED is selected by placing the toggle switch in the center position.

The WORLD CORE does not require that the user select an explicit mode in order to use the relative DIRECTIONAL INPUT system. Simply

patching a GATE signal into one or more of the DIRECTIONAL INPUTs will automatically engage relative control for the appropriate direction. Exiting the system is similarly straightforward: to regain absolute X/Y control of SPRITE 1, simply patch a voltage into its X or Y CV input (or wiggle its corresponding knob).

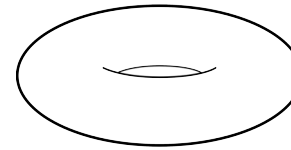
Let's try moving SPRITE 1 using the DIRECTIONAL INPUTs. If you've linked a CONTROL CORE, all you need to do is start pressing the D-PAD buttons—the DIRECTIONAL OUTPUTs are already normalled to the DIRECTIONAL INPUTs. If your CONTROL CORE is unlinked, or if you don't have a CONTROL CORE, you'll need to patch into the GATE inputs manually.

You should see SPRITE 1 move around the screen in response to the DIRECTIONAL INPUT activity. Try changing the SPEED toggle to get a feel for the different rates of motion. If GATE signals are sent to adjacent DIRECTIONAL INPUTs simultaneously, a diagonal motion will be produced (for instance, UP and LEFT, or DOWN and RIGHT). If opposing directions are triggered simultaneously (LEFT and RIGHT, or UP and DOWN) SPRITE 1 will simply freeze in place.

Now let's move back to the absolute X/Y positional system by turning the X and Y knobs. You should see SPRITE 1 "snap" to the voltage position specified by the knob, indicating that the axis has been placed back under absolute control.

The WORLD CORE decreases its sensitivity to the X/Y inputs when using the DIRECTIONAL INPUTs. This is to avoid problems where small fluctuations in voltage could "steal" control away from the DIRECTIONAL INPUTs, causing SPRITE 1 to jump in position unexpectedly. If you find that SPRITE 1 is not changing position in response to the X or Y knobs, try wiggling the knob faster or over a greater range of values.

The positional system you use has deep cosmological implications for your worlds. While using the DIRECTIONAL INPUTs you probably saw SPRITE 1 occasionally "wraparound" when it hit the sides of the screen, appearing to move smoothly from one side to the other. This creates a toroidal shape in which SPRITE 1 moves through a finite but unbounded universe, somewhat like gliding across the surface of a doughnut. Toroidal worlds were used by classic videogames like *Asteroids* and *Pac-Man* to increase the player's freedom of movement. This topology is only available using the relative positional system; when using the absolute X/Y system, the universe necessarily becomes bounded.



> Relative and absolute control can be mixed and matched. Try using the gamepad to control the X-axis via the DIRECTIONAL INPUTs, and then patch a sine or triangle wave into the Y input.

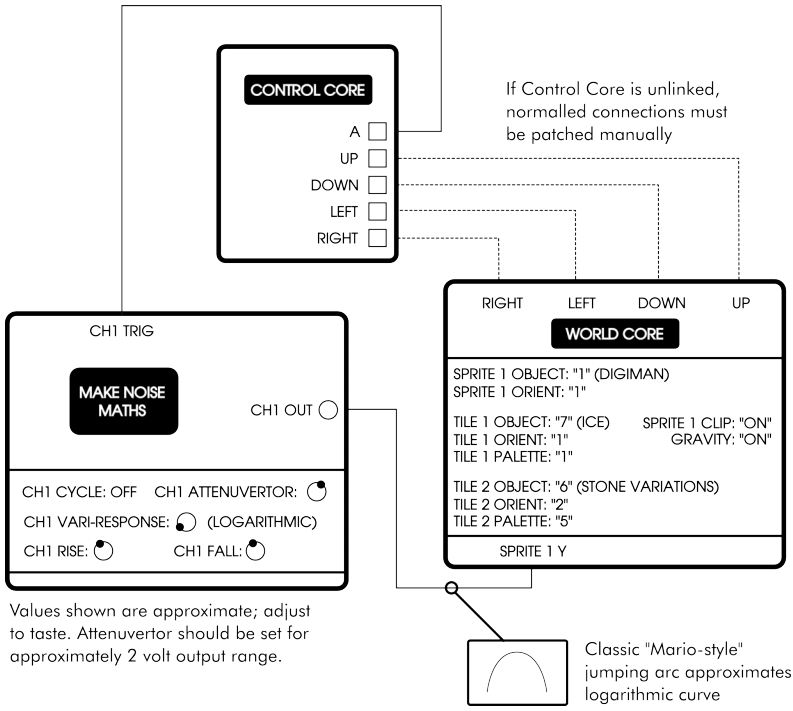
> For a simple "artificial intelligence" patch, try sending random or pseudo-random GATE signals into the DIRECTIONAL INPUTs. The frequency of the GATE signals in conjunction with the SPEED toggle will control the density of the resulting motion, while changes to each GATE signal's pulse width can be used to bias the SPRITE movement in particular directions.

SPRITE 1: GRAVITY

SPRITE 1 also features a dedicated GRAVITY mode inspired by the simple physics of classic platforming games. When GRAVITY is turned on, the Y-axis CV input is used to control jumping, while SPRITE 1 is

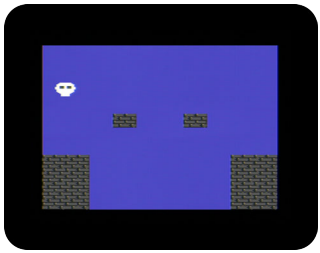
forced downwards. Either relative or absolute systems may still be used for the X-axis, although relative, gamepad-based control is recommended for traditional "Mario-like" movement mechanics.

The patch below builds a simple platforming world with jumping control over SPRITE 1. The CONTROL CORE DIRECTIONAL OUTPUTs are connected to the WORLD CORE's DIRECTIONAL INPUTs (either through normalled connections or manually), while the A button output triggers a MATHS ENVELOPE. This ENVELOPE controls the SPRITE 1 Y-Axis position to generate a jump.



At some point while building the patch, SPRITE 1 was likely grounded off-screen. The UP and DOWN inputs are disabled when GRAVITY is engaged; to regain control, turn CLIPPING and GRAVITY off, then use the D-PAD to move SPRITE 1 over the lower brick platform on the

left-hand side of the screen. Turn CLIPPING back on, but leave GRAVITY turned off for now. Instead, simply hit the A button to trigger a jumping arc and try moving SPRITE 1 around the world a little.



This patch already does a pretty good job of creating simple jumping physics, but several problems are immediately obvious:

1. Pressing up or down on the D-PAD defeats the jumping physics, allowing SPRITE 1 to hover at will.

2. Because the MATHS ENVELOPE is controlling the absolute Y-axis position, SPRITE 1's jump height is proportional to the vertical starting position. Whether the player starts on the ground or on a platform block, the jump always ends at the same place on the screen. This has the effect of making total jumping height longer or shorter depending on where the SPRITE starts on the screen. Assuming you've set the MATHS attenuvertor according to the patch, you probably can't reach the higher platforms when starting from the lower platforms; and if you can, jumping back off the higher platform is likely impossible since it is beyond the reach of the MATHS voltage function.

3. For the same reason, there's significant input delay that increases as the SPRITE moves up to higher platforms. Since the MATHS ENVELOPE always starts from 0V, when jumping off a platform SPRITE 1 waits until the ENVELOPE intersects its position before beginning the jump.

4. SPRITE 1 can continue to jump and maneuver even after falling into the bottomless pit, breaking the illusion of persistent GRAVITY.

To fix these problems, we need to use the GRAVITY system to apply a more intelligent physics. Position SPRITE 1 over the leftmost lower platform again, and turn GRAVITY back on (making sure that CLIPPING is also turned on). SPRITE 1 should fall downwards and eventually come to rest on the ground.

Now try jumping on and off of the higher platform. If you fall in the pit, disengage both GRAVITY (to enable the relative Y-axis inputs) and CLIPPING (to easily move through obstacles) and use the D-PAD to reposition SPRITE 1 over the terrain.

You should notice several differences now that GRAVITY is turned on:

1. Pressing UP or DOWN on the D-PAD no longer has any effect; the relative Y-Axis inputs have been disabled.

2. The jumping arc covers the same distance when starting from either platform height. This means that you can now reach higher areas of the screen by jumping off the high platform, possibly even jumping past the top of the screen (depending on the MATHS settings).

3. The arc begins immediately upon pressing the A button, even on the higher platform.

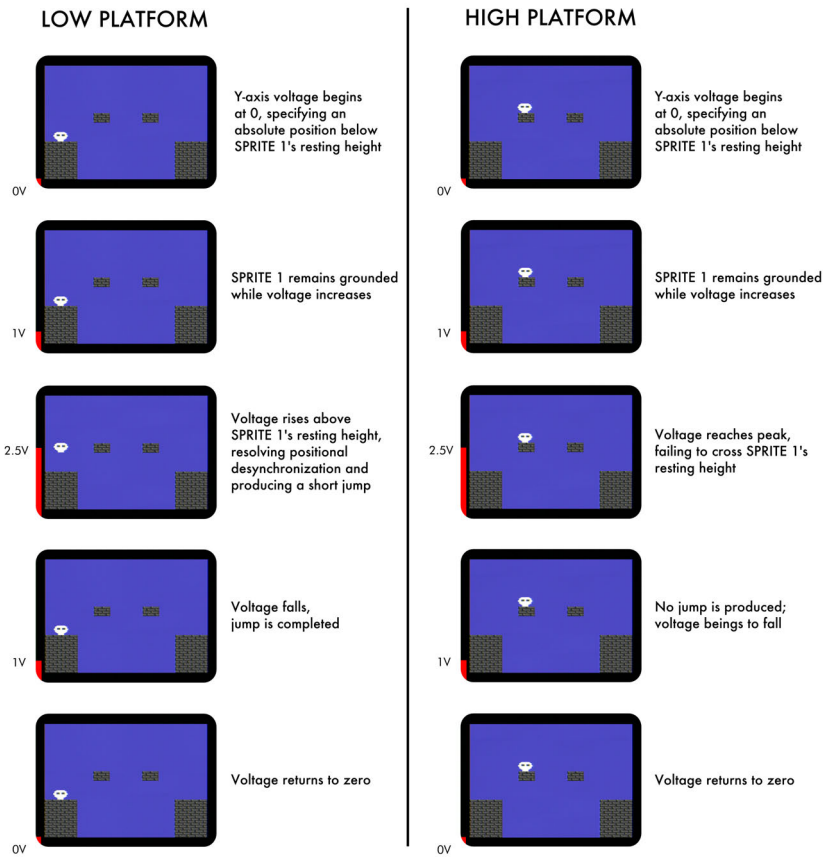
4. If you fall in the pit you should no longer be able to move or jump.

The GRAVITY system uses a technique based on a virtual CV offset generator and a dynamic input analyzer to achieve its performance. Instead of treating 0V as an absolute position, enabling GRAVITY causes the Y input to dynamically position 0V at SPRITE 1's current resting height. An easy way to think about it is to imagine summing a static DC offset with the MATHS ENVELOPE feeding the Y input. With the DC offset at zero, the MATHS ENVELOPE's 0V-point remains unchanged, positioning the SPRITE at the bottom of the display. As the

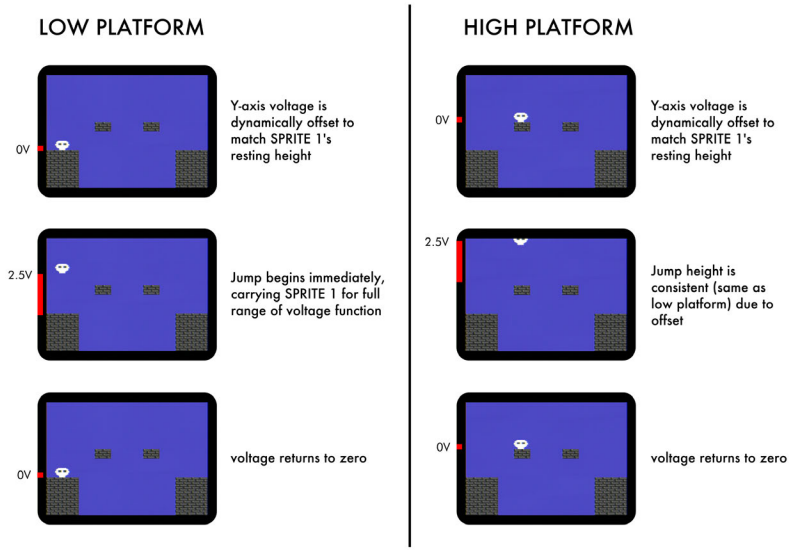
DC offset increases however, the 0V-point is effectively raised in height, causing the jumping arc to begin from a higher point of origin. The WORLD CORE simulates this DC offset internally, calculating the amount of offset dynamically based on the resting height of SPRITE 1. Note that the overall shape and amplitude of the ENVELOPE is unchanged by this offset; only its absolute coordinates have shifted up or down across the screen.

The following two diagrams demonstrate the effect visually: first with GRAVITY turned off, and then with GRAVITY turned on.

JUMPING ARC — GRAVITY OFF

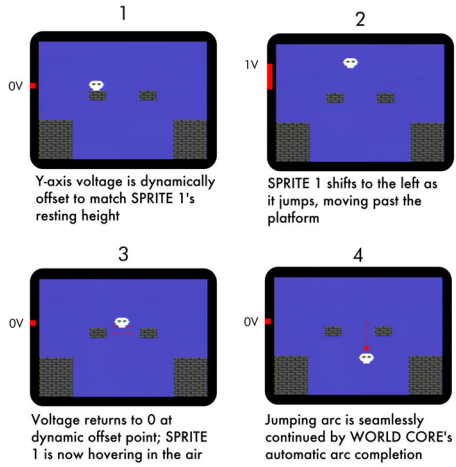


JUMPING ARC — GRAVITY ON



The final component of the GRAVITY system is a process called AUTOMATIC ARC COMPLETION, or AAC. When jumping off of a higher platform to land on a lower one, a problem arises when the SPRITE passes through the dynamically offset 0V-point—i.e., when it must descend lower than its previous resting height. Since there is literally no voltage function to carry it down to the lower platform (the MATHS ENVELOPE has already reset to 0V by the time the SPRITE crosses the platform line), the WORLD CORE generates an internal

AUTOMATIC ARC COMPLETION



ramp to complete the arc instead. This arc is generated by analyzing the incoming **Y CV** input voltage and computing its speed in realtime, so that no matter the speed of the incoming arc, the **WORLD CORE** will complete it seamlessly.

In practice, this allows the gravitational properties of your worlds to be based entirely on the speed and shape of the analog functions used to drive the **Y** Input. Slow **ENVELOPEs** produce conditions similar to a small moon, while fast **ENVELOPEs** evoke the crushing weight of Jupiter. You can even produce esoteric conditions by experimenting with exponential slopes, or complex non-linear arcs achieved with multi-stage **ENVELOPEs** like **ADSRs.**

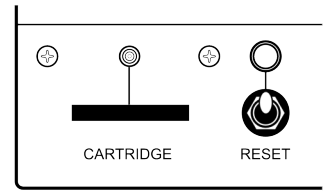
Although the **AAC** subsystem is generally quite accurate, it will occasionally miscalculate the incoming arc's speed and produce an internal ramp that is too fast or slow. When this happens, the problem will generally self-correct within one or two additional jumps. The **AAC's** current speed setting can also be "flushed" manually by cycling the **GRAVITY** toggle. This will reset the **AAC** to a default moderate speed.

Occasionally the **SPRITE** will come to rest before the **MATHS ENVELOPE** has returned to **0V** (such as when it bumps against an overhead platform and ricochets to the ground). When this occurs, the **WORLD CORE** prevents another jump from triggering until the **ENVELOPE** has returned to **0V**. During this delay, the **SPRITE** will shake left and right to visually indicate that the **Y** input voltage is out of sync.

5. WORLD PACKS

- Using the CARTRIDGE SLOT
- Creating and Modifying WPACK.TXT Files
- Creating and Modifying CONFIG.TXT Files
- Generating WPACK.TXT Files with WPACKer
- Building Better Worlds

USING THE CARTRIDGE SLOT



The WORLD CORE's CARTRIDGE SLOT allows you to load custom graphics data and system settings via standard SD CARDs. Data is stored on the CARDs in two special text files: WPACK.TXT, which stores graphics collections called WORLD PACKs, and CONFIG.TXT, which stores system settings called CONFIGURATIONs. A CARD can contain both files together, or only one file in isolation (to alter the graphics without modifying system settings, or vice versa).

WORLD PACKs and CONFIGURATIONs are extremely powerful tools for personalizing the WORLD CORE, but they are also somewhat complicated to create. To start with, let's download a premade WORLD PACK and learn how to load it into the WORLD CORE. Begin by following the steps below:

1. Locate or purchase a full size SD CARD. Brand and speed class are unimportant.

2. Make sure the CARD is formatted as either FAT16 or FAT32.

3. Visit **specialstagesystems.com/wpacks** and download the *Tangram Park* WORLD PACK.

4. Unzip the contents of *Tangram Park* (you should see both a WPACK.TXT and a CONFIG.TXT), and place them in the root directory of the SD CARD.

5. **Do not rename the files.** The WORLD CORE only recognizes "WPACK" and "CONFIG" as valid filenames.

Although you may insert the SD CARDs while the module is powered off, it is not necessary to do so. CARDs are hot swappable and may be inserted or removed at any time except during boot, which is when the WORLD CORE accesses the CARTRIDGE SLOT and loads the SD CARD contents into RAM. During this portion of the boot sequence, the CARTRIDGE LED will flash to indicate that the SD CARD is currently in use and should not be tampered with.

> **WARNING**: Do not insert or remove the SD CARD while the CARTRIDGE LED is active. Doing so may result in corrupt data and damage to the WORLD CORE and / or SD CARD.

Once a CARD has been inserted or swapped, the WORLD CORE must be rebooted using the RESET toggle switch in order for the new data to be loaded. Before we insert the CARD and reboot the module, let's look at how the RESET switch works.

The RESET toggle is a momentary, spring-loaded switch. It may be flicked up or down, but always returns to the center position when you release it. The RESET OUTPUT, located immediately above the switch, generates a +10V GATE in response to the position of the switch. The behavior of the RESET SWITCH and OUTPUT is as follows:

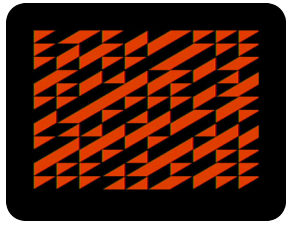
- RESET held down: Produces a +10V GATE and RESETS the WORLD CORE
- RESET held up: Produces a +10V GATE only

The **RESET OUTPUT** can be used to initialize other modules like sequencers or clock dividers in tandem with the **WORLD CORE**. When decoupled from the **WORLD CORE** using the upward switch position, the **RESET OUTPUT** can function as a sort of "soft reset" in which other patch elements are initialized without flushing **WORLD CORE RAM**.

Now let's load up *Tangram Park*, starting with the **WORLD CORE** fully powered to demonstrate its hot swap capability.

1. Make sure your **WORLD CORE** is turned on and has completed its boot sequence, and then insert the **SD CARD** into the **CARTRIDGE SLOT**.

2. Flick the **RESET** switch downwards to reboot the **WORLD CORE**, and wait a few seconds while the module power cycles (the screen will go black).



You should soon see a blue progress bar begin to fill the screen as the **WORLD CORE** copies the **CARD**'s contents into **RAM**. The process can take 10-15 seconds to complete, so be patient. Once it's completed, the **WORLD CORE** will continue to boot as normal. When it's finished booting, feel free to have a look around the new **WORLD PACK** and explore its **SPRITEs**, **TILEs**, **MAPs**, and **PALETTEs**.

When you're finished looking around, remove the **SD CARD** from the **CARTRIDGE SLOT** and flick the **RESET** switch down again. With no **CARD** present, the **WORLD CORE** will now reboot using its default graphics and settings, returning the system to normal.

If the **WORLD CORE** never displays the progress bar, or if the progress bar hangs for more than 10-15 seconds, it's possible that the **CARD** contains improperly named or located data. Turn off the

WORLD CORE by powering down your modular synthesizer, remove the CARD, and inspect it on your computer. The WPACK.TXT and CONFIG.TXT files must be placed in the root directory, and they must not be renamed. If all else fails, try another SD CARD or contact **Special Stage Systems** for assistance.

CREATING AND MODIFYING WPACK.TXT FILES

Before we open up and inspect the contents of the *Tangram Park* WPACK.TXT, let's have a quick review of the main elements that make up WORLD CORE graphics. The WORLD CORE uses four primary asset types to create a scene, which we're already familiar with from the proceeding chapters:

1. 32 blocks of **TILE DATA**, each 16 x 16 pixels in size and consisting of four potential colors, organized into eight sets of four, indexed with the OBJECT and ORIENT controls.

2. 32 blocks of **SPRITE DATA**, each 16 x 16 pixels in size and consisting of three potential colors and a transparency layer (color 0), indexed by their own OBJECT and ORIENT controls in a manner identical to TILE DATA.

3. 16 **MAPs**, each consisting of 10 x 12 screen locations that specify where TILE 1 and TILE 2 data will be drawn, indexed by the MAP SELECT control.

4. 16 **PALETTEs** that are applied to the TILE DATA blocks, each consisting of four colors, and indexed by two PALETTE controls for TILE 1 and TILE 2. PALETTEs are indirectly applied to SPRITEs as well, which inherit the PALETTE of the TILE they are drawn on top of.

Each of these four asset types is represented as raw text in the WPACK.TXT file. The format is similar to ASCII-art, in that it is designed to be human-readable, as graphical as possible, and easily editable with any simple text editor.

Open the *Tangram Park* WPACK.TXT file in the text editor of your choice and have a look around. Note the introductory text at the top of the file. Before any data is declared there is a COMMENT AREA demarcated by two pound signs ("#"). This area is typically used to record the WORLD PACK title, author name, and so on—although it can be used for other purposes as well. Any character can be used in this area except for "#," since it's used to start and end the COMMENT AREA.

> COMMENT AREAS can appear anywhere in the file so long as they don't appear in the middle of TITLEs, ASSET BLOCKs, or DATA LINEs

Now scroll down the file and note the different sections. Other than the "#" demarcated COMMENT AREAS, WPACK.TXT files consist of the following elements:

1. **SECTION TITLEs** "/x/" (/BOOT LOGO/, /MAPS/, etc.)
2. **ASSET TITLEs** "[x]" ([Special Stage Emerald], [1], [2], etc.)
3. **ASSET BLOCKs**
4. **DATA LINEs**

SECTION TITLEs end and begin in forward slashes ("/"), and describe the primary file regions. ASSET TITLEs precede each individual ASSET BLOCK and are surround by brackets ("[" and "]").

- Every section must be proceeded by a "/x/" SECTION TITLE
- Every asset block must be preceded by a "[x]" ASSET TITLE

It doesn't matter what goes between the [x] brackets and /x/ slashes, as long as the brackets and slashes themselves are present. The text could consist of numerals, letters, short descriptions; whatever is most useful for organizing the WORLD PACK.

WARNING: The **WORLD CORE** processes **WPACK.TXT** files based on the order the elements appear in, not on the text that appears in **SECTION TITLEs** or **ASSET TILEs**. If you try to put **PALETTEs** first, or move **TILEs** ahead of **SPRITEs**, for instance, the **WORLD CORE** will not be able to interpret the file and will crash during boot.

As long as the **SECTION** and **ASSET TITLEs** remain undisturbed, it's possible to omit individual **ASSET BLOCKs.** When loading the **WORLD PACK**, the **WORLD CORE** will automatically replace any missing **ASSET BLOCKs** with the corresponding data from its default graphics. This is useful if you only want to modify certain areas of the default data, such as replacing the **MAPs** while using the default **SPRITEs** and **TILEs**, etc.

```
/MAPS/ ← SECTION TITLE

[1] ← ASSET TITLE
0 0 0 0 0 0 0 0 0 0  ← DATA LINEs
1 1 1 1 1 1 1 1 1 1  ←
0 0 0 0 0 0 0 0 0 0  ←
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
```

To the right is an example of an **ASSET BLOCK**. This **ASSET BLOCK** is the first **MAP** in the *Tangram Park* **WPACK.TXT** file. Like all **ASSET BLOCKs**, it is defined by a series of individual **DATA LINEs.**

DATA LINEs take different forms depending on the type of ASSET BLOCK. MAP blocks are binary, using "0" to represent TILE 1, and "1" to represent TILE 2.

TILE and SPRITE blocks, on the other hand, consist of DATA LINEs expressed using four digits ("0," "1," "2," and "3"), as shown in the TILE block below. For TILEs, each of these represents one of four colors. For SPRITEs, Color 0 represents transparency.

```
[10]
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

This TILE block only contains "0" and "1," since the *Tangram Park* WORLD PACK uses simple two-color designs. "2" and "3" are also valid digits in TILE and SPRITE blocks, and designate the corresponding colors in the PALETTE assets. Note its graphical nature—the actual TILE DATA will follow this block 1-to-1, producing a diagonally divided, dual color rectangle.

The spaces between ASSET BLOCK characters are used to make the WORLD PACK more human-readable. They do not affect how the WORLD CORE processes data. If you'd like, you can omit the spaces for thinner-looking ASSET BLOCKs, or add extra spaces to make them fatter. Two spaces between each character will look closest to the WORLD CORE's rendering dimensions, but can make the text harder to visualize at a glance.

Although the **WORLD CORE** indexes **TILE** and **SPRITE DATA** in a 2-dimensional array, their **ASSET BLOCKs** are presented as a single linear sequence in **WPACK.TXT** files. Blocks 1-4 represent the **ORIENT** indexes of **OBJECT** "1," while blocks 5-8 represent the **ORIENT** indexes of **OBJECT** "2," and so on.

Unlike the graphical **MAP**, **TILE**, and **SPRITE** blocks, **PALETTEs** are defined verbally. A **PALETTE ASSET BLOCK** consists of four **DATA LINEs**, each representing an individual color.

```
[5]
CYAN2, 0  ← Color 0
BLACK, 2  ← Color 1
CYAN2, 0  ← Color 2
BLACK, 2  ← Color 3
```

The **DATA LINEs** are written in order from color "0" to color "3." Each line consists of a color name ("**GREEN1**") followed by a brightness value ("3"), separated by a comma. There are eight possible brightness values from 0-7, although some of these have unpredictable results (as discussed below).

- **BLACK** 2-7
- **BLUE1** 3-6
- **BLUE2** 3-6
- **PURPLE** 3-6
- **MAGENTA1** 3-6
- **MAGENTA2** 3-6
- **RED** 3-6
- **ORANGE** 3-6
- **BROWN** 3-6
- **YELLOW1** 3-6
- **YELLOW2** 3-6
- **GREEN1** 3-6
- **GREEN2** 3-6
- **GREEN3** 3-6
- **CYAN1** 3-6
- **CYAN2** 3-6
- **CYAN3** 3-6

There are 17 color names to choose from in total. Numbered names simply represent a mixture of the preceding and following colors. For example, **MAGENTA2** is a mix of **MAGENTA1** and **RED**, while **CYAN2** is a mix of **CYAN1** and **CYAN3**. **BLACK** is a special color, since depending on its brightness value it can be transformed into white or shades of grey.

For all colors except **BLACK**, the legal range of brightness is from 3-6. For **BLACK**, the range is extended to 2-7. Stretching beyond this range causes the **WORLD CORE's** analog video output to start deforming the **SYNC** and **COLOR BURST** signals, which can have destabilizing effects on the image.

Depending on the specific combination of colors in a given PALETTE, illegal brightness values can sometimes be used to achieve interesting special effects. Experimentation is encouraged, but be aware that the results are often unpredictable and can vary from display to display.

IMPORTANT: For all colors other than BLACK, specifying a brightness value of 0 leads to a special set of **hyper-saturated colors**. Once again, these will often look different depending on the display, so some caution is necessary—when used carefully however, they can add extra vibrancy to a PALETTE.

The final PALETTE in WPACK.TXT (**"PALETTE 17"**) is used for the TILE INDEX highlight color enabled by the VISIBLE toggle.

CREATING AND MODIFYING CONFIG.TXT FILES

Now let's have a look at the CONFIG.TXT file. This file allows several alternate modes to be selected for various WORLD CORE system settings. Just like the WPACK.TXT file, it begins with a COMMENT AREA demarcated by pound signs ("#"). In the *Tangram Park* CONFIG.TXT file, this area lists the possible settings for the file parameters and includes some short explanations of their functionality.

Below the COMMENT AREA are the CONFIGURATION DATA LINEs. Parameter names and their corresponding values are separated by

an equal sign ("="). Just like in WPACK.TXT, the order in which the entries appear is critical, since the WORLD CORE processes the file based on entry order, not title.

```
INTRO = 1
COLLISION MODE = 0
SLOW SPRITE = 0
PALETTE DESYNC = 1
BORDER COLLISION = 0
```

There are five parameters in total: INTRO, COLLISION MODE, SLOW SPRITE, PALETTE DESYNC, and BORDER COLLISION. *Tangram Park* uses a standard CONFIGURATION except for the PALETTE DESYNC setting, which is normally set to "0."

The following list describes each parameter and its available settings. The default setting (i.e., the setting used by the WORLD CORE if no CONFIG.TXT is present) is listed in parentheses following the parameter title.

INTRO (Default 1)

Enable or disable the **Special Stage** intro animation (uses /BOOT LOGO/ graphic if WPACK.TXT is supplied).

- "1" = on, "0" = off.

COLLISION MODE (Default 0)

Switch between 5 behaviors for the EDGE COLLISION outputs.

- "0" selects the standard mode, in which the outputs correspond to SPRITE 1 crossing the edges of the screen.

- "1" selects a mode where the outputs correspond to SPRITE/TILE COLLISION on individual sides of SPRITE 1. This allows you to detect when the SPRITE has collided on its left vs. its right side, for instance.

- "2" selects an experimental variant of this mode where individual sides are processed for SPRITE/SPRITE COLLISION instead of SPRITE/TILE.

- "3" selects a mixture of the standard and alternate modes, where the LEFT EDGE and RIGHT EDGE outputs operate as normal, but the TOP EDGE and BOTTOM EDGE outputs correspond to the top and bottom of SPRITE 1 in SPRITE/TILE COLLISION.

- "4" selects a inverted version of mode "3" where the LEFT EDGE and RIGHT EDGE outputs correspond to SPRITE sides, and the TOP EDGE and BOTTOM EDGE outputs preserve the standard EDGE COLLISION.

SLOW SPRITE (Default 0)

Slow down the movement of SPRITE 1's slowest speed setting when using the DIRECTIONAL INPUTs.

- "0" = normal, "1" = slow.

PALETTE DESYNC (Default 0)

Enable the extended range of "illegal" brightness values when using the ?PA* switch to generate glitched PALETTEs. Normally, the WORLD CORE restricts ?PA* generated PALETTEs to the legal range. The illegal range can produce interesting effects (especially on CRTs), but varies significantly from display to display

- "1" = illegal range, "0" = legal range.

BORDER COLLISION (Default 0)

Restrict SPRITE motion to the visible frame, preventing them from passing behind the screen border. The EDGE COLLISION outputs are also adjusted accordingly, so that SPRITE 1 will produce an EDGE COLLISION when hitting the border, instead of when passing fully behind it.

- "1" = restrict to borders, "0" = pass behind borders.

A CONFIG.TXT file can exist in addition to a WPACK.TXT on the same CARD, as in the case of the *Tangram Park* WORLD PACK. However, if you only want to alter system settings without loading custom graphics, a CONFIG.TXT file can also exist in isolation. The inverse is also true: if you prefer to use default settings, you can simply use the WPACK.TXT file in isolation.

GENERATING WPACK.TXT FILES WITH WPACKer

Although WORLD PACKs can be created and edited entirely as raw text, it is often desirable to use graphics software when creating more complex SPRITEs, TILEs, and MAPs. WPACKer is a Mac and Windows compatible software tool that generates WPACK.TXT files from graphical SPRITESHEETS. A SPRITESHEET is a single image file containing a complete WORLD PACK of SPRITE, TILE, and MAP data. WPACKer accepts standard .PNG image files for its SPRITESHEETS, so you can work in virtually any graphics software you'd like.

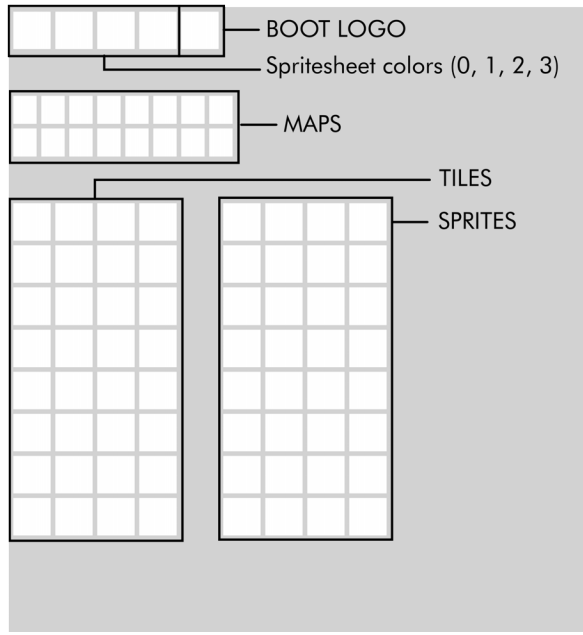
- WPACKer is available for free from **Special Stage Systems**. Simply navigate to **specialstagesystems.com/wpacks** and look for the WPACKer download.

The download includes the WPACKer program, a short readme file, and a .PNG template image to use when constructing your SPRITESHEETS. Using WPACKer is very straightforward: simply place your completed SPRITESHEET in the same folder as WPACKer, and then launch the program. WPACKer will automatically load the SPRITESHEET

and convert it into the WPACK.TXT format. The newly generated WPACK.TXT file will be placed in the same folder as WPACKer and the SPRITESHEET.

> **IMPORTANT**: the resulting text file will be named "Your_World" followed by a string of digits. The file must be renamed to WPACK.TXT in order to be identified and loaded by the WORLD CORE.

Elements in the SPRITESHEET must be arranged in a particular order and appear exactly in the grid spaces outlined by the template. Let's open the template image and examine its different sections.



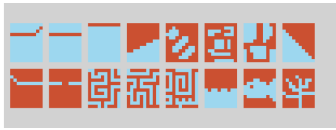
The top row of elements contains the SPRITESHEET colors, followed by the BOOT LOGO. The first four boxes should contain solid blocks of color. These are the colors that WPACKer will use to interpret the rest of the SPRITESHEET, and have no relation to the actual PALETTE colors used by the WORLD CORE. The BOOT LOGO replaces the **Special Stage** emerald used in the boot sequence animation.

Here's what a completed version of this section might look like:

The light blue, orange, dark blue, and light yellow colors are mapped to colors "0," "1," "2," and "3" respectively, and are used to convert the BOOT LOGO into text. The body of the crab is orange, so it's converted to "1," while the blue background is converted to "0." If you look closely, you can see that the crab's eyes have been converted as "2," since they are dark blue.

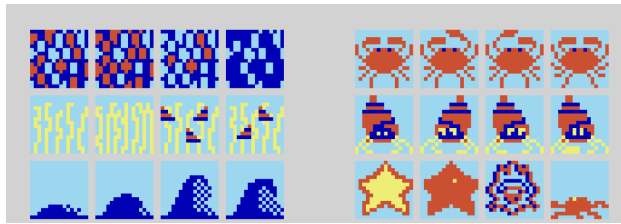
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 0
0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0
1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1
1 0 0 1 0 1 2 1 1 2 1 0 1 0 0 1
1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1
0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
0 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0
0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0
0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0
0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0
0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
```

The four colors you define at the top of the template are used by WPACKer to interpret all the other assets in the file, so it's important to follow them closely. Using a slightly different shade of orange, for instance, would result in unconvertible MAPs, TILEs, or SPRITEs.



The MAP section only uses colors "0" and "1," since all it does is specify TILE 1 and TILE 2 locations. The MAP assets have a different aspect ratio from the others, since they are ultimately translated into a 10 x 12 grid. The MAP assets are laid out from left to right, beginning with MAP 0 in the upper left corner and ending with MAP 16 in the lower- right corner.

TILEs and SPRITEs appear exactly as they do in the corresponding tables from CHAPTERS 3 and 4. The OBJECT CV corresponds to the columns, and the ORIENT index corresponds to the rows. All four colors (i.e., digits 0-3) are valid.



WPACKer does not currently implement a conversion process for PALETTEs. Although WPACKer automatically generates the correct SECTION and ASSET TITLEs for PALETTEs, the actual PALETTE DATA must be entered by hand.

> Assuming that the SECTION and ASSET TITLEs are left undisturbed, the WORLD CORE will simply use its default PALETTEs when loading the file if no custom PALETTEs are present. The same is true for missing MAPs, TILEs, and SPRITEs.

BUILDING BETTER WORLDS

Just like the retro videogame hardware that inspired it, the WORLD CORE possesses a unique constellation of technological constraints and creative possibilities. Designing content that navigates those constraints can be a challenge—but with practice and consideration, you'll be steering MING MECCA towards new dimensions in no time.

This section contains a few general tips for designing your first WORLD PACKs. Since MING MECCA is such an experimental and

open-ended system, it's important to remember that these tips are meant only as a kind of basic "orientation." As you advance in your WORLD PACK creation, you may find yourself deliberately ignoring some of the advice contained here. The key is to always stay true to your own taste—if you follow your vision diligently and honestly, there is no "bad design."

USE COLOR TO YOUR ADVANTAGE

By far the most difficult aspect of WORLD PACK design is coming to grips with the idiosyncrasies of the color system. This is partly due to the inherent unpredictability of analog composite video, which can make it difficult to know exactly how colors will appear on different displays, and how adjacent colors will interact with one another due to various color artifacts. As you spend more time with your WORLD CORE, you'll start to get a feel for working with composite video, and perhaps even exploit some of those artifacts as effects.

A deeper issue however concerns the WORLD CORE's lack of dedicated PALETTEs for SPRITEs. Because SPRITEs inherit their PALETTE from underlying TILEs, it can be difficult to make them "pop" against the background; a brown, red, black, and yellow SPRITEs, for instance, can be difficult to see against an identically colored TILE. A good way to avoid this problem is with a clearly defined "color topology"—a system for deciding which colors will be used by SPRITEs, which will be used by TILEs, and how both will interact in various PALETTEs.

As a general rule, it's important to design your assets so that at least one or two colors are reserved for SPRITEs. The WORLD CORE's default graphics, for instance, reserve color "3" for SPRITEs, which is a bold WHITE in almost every PALETTE. Color "3" forms the basis of most of the SPRITE DATA, but rarely appears in TILE DATA, allowing for an overall look that features brightly lit SPRITEs against comparatively dark backgrounds.

Your WORLD PACKs don't need to have simple single-color SPRITEs of course, and in fact the opposite approach (using two or three colors for SPRITEs and reserving one or two colors for TILEs) is just as valid. The point is to design some sort of SPRITE / TILE separation into your assets so that your worlds end up with enough graphical contrast between their elements.

KEEP THE INTERFACE IN MIND

Ultimately, the WORLD PACKs you design will be rendered and manipulated via the WORLD CORE's physical interface. Try to imagine how your assets will be accessed and explored once they're behind the module's control surface.

For instance, because SPRITE DATA is indexed by two separate control voltages (eight OBJECT states and four ORIENT states), it makes sense to design your assets into eight groups of four, or four groups of eight. As an example, the default graphics use the OBJECT index to select between different base designs, and the ORIENT index to select animation frames. If you design your assets as one giant lump, or if you split related ideas across the ORIENT and OBJECT indexes, the results can be confusing to navigate using the physical controls.

Similarly, the order in which assets appear can be extremely important, especially if it has any spatial significance. Consider the first TILE set in the default graphics, CLOUDY SKY. CLOUDY SKY has four frames of ORIENT-indexed animation that scrolls horizontally. The frames have been organized so that an increase in voltage scrolls to the right, and a decrease scrolls to the left. The voltage increases as the ORIENT knob is turned clockwise, so the direction of the knob rotation aligns with the scrolling direction. If the frames were organized in the reverse order, the clouds would scroll left when turning the knob to the right—a behavior that feels "wrong" in terms of interface expectations.

DESIGN SYSTEMS, NOT GRAPHICS

It's easy to think of each asset you create as a self-contained statement, but once your MAPs, SPRITEs, and TILEs are loaded into the WORLD CORE, they'll begin to interact with each other in complex real-time systems. What do your SPRITEs communicate about how they should be used? A door SPRITE implies very different procedural possibilities than an owl SPRITE, for example; one is probably stationary, serving as a link between geographic zones, while the other is probably in motion, driven by LFOs or ENVELOPEs as it dances across the world.

How do your different SPRITE designs work together? Perhaps there should be a mouse to escape the owl, or a key to open the door. Your TILE designs are equally important. Why not create a grass field for the mouse to hide in, or a brick wall for the door to sit against? Even completely abstract designs can form evocative geometric or visual relationships, such as "Tetris" style blocks that interlock, or detailed stripe patterns that alias.

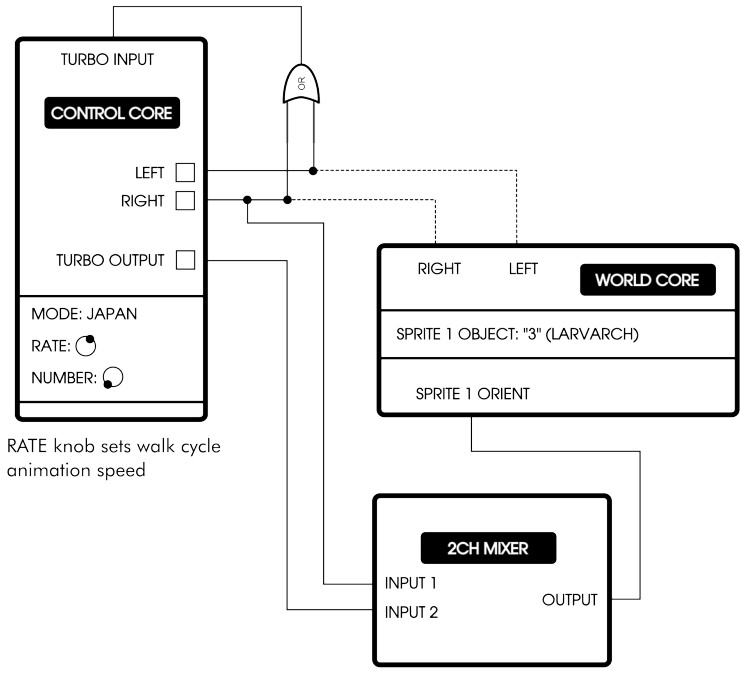
Once you've thought about the interaction between your SPRITEs and TILEs, how do your MAPs support them? MAPs in particular have a huge impact on the procedural structure of the world. Try to think of how your large-scale MAP elements translate to COLLISION events, and compose them accordingly.

APPENDIX A: ADVANCED TECHNIQUES

- Walk Cycles
- **SPRITE** Multiplexing and **ASFM**
- Multi-Map Composition

WALK CYCLES

When moving **SPRITEs** around the screen, it's often desirable to create motion-triggered animations called "walk cycles." Walk cycles are short 2-4 frame animations that give the impression of walking, running, hopping, and so on, and can help breath life into your patches. There are a variety of ways to create walk cycles, some more complex than others. The method described here uses only the **WORLD CORE**, **CONTROL CORE**, a 2-channel mixer, and a single OR logic gate.



RATE knob sets walk cycle animation speed

Attenuate CH1 to span first 2 ORIENT frames
Attenuate CH2 to create offset for last 2 ORIENT frames

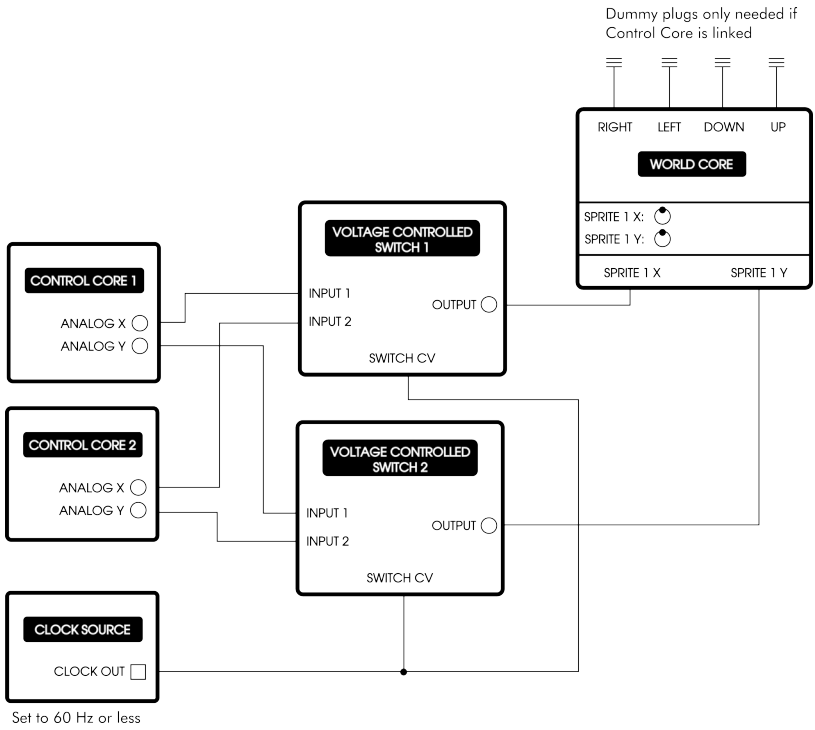
The patch uses the LARVARCH SPRITE (OBJECT "3"), which contains an ORIENT index split into two dual-frame animations: the lower half of the index (frames 1-2) animate LARVARCH crawling to the left, and the upper half (frames 3-4) animate him crawling to the right. The CONTROL CORE's TURBO OUTPUT is used to toggle between the two animation frames, while the OR-ed LEFT and RIGHT outputs are used to trigger the TURBO OUTPUT, so that no matter which direction LARVARCH is moved in, he will always animate. The RIGHT output creates an offset that selects the upper two frames when moving LARVARCH to the right; when moving him to the left, no offset is applied, thereby selecting the lower two frames. The speed of the animation is determined by the CONTROL CORE's RATE knob.

SPRITE MULTIPLEXING AND ASFM

Although surprisingly diverse worlds can be built using only two SPRITEs, certain situations demand the ability to draw several objects simultaneously. A technique called "multiplexing" can be used to give the visual impression of multiple independently positioned objects by rapidly changing the X/Y positions of SPRITE 1 or SPRITE 2. Since the WORLD CORE redraws its entire display 60 times per second, it's possible to modulate the SPRITE's position so quickly that it "splits" into multiple parallel locations. Because the SPRITE is actually drawn in only a single position at a time, multiplexed SPRITEs appear to flicker. The severity of the flicker depends on both the cycling speed and the number of positions (i.e., the number of visually perceived SPRITEs) being cycled.

The patch below multiplexes SPRITE 1 using two VOLTAGE CONTROLLED SWITCHES (or a single dual switch such as the **Doepfer** A-150) paired with two CONTROL CORES. The switches feed SPRITE 1's X/Y

inputs by cycling between each CONTROL CORE's ANALOG X/Y outputs at 60 Hz or less. By clocking the switches at 60 Hz, the SPRITE will change position approximately once per frame, with each position flickering 30 times per second (60/2).



Note that the voltage sources do not have to be CONTROL CORES; any sources can be used, including LFOs, random voltage generators, cycling ENVELOPES, or even static DC offsets. However, for patches where the SPRITE locations are stationary, a better solution is to simply use a STEP SEQUENCER clocked at 60 Hz, eliminating the need for VOLTAGE CONTROLLED SWITCHES.

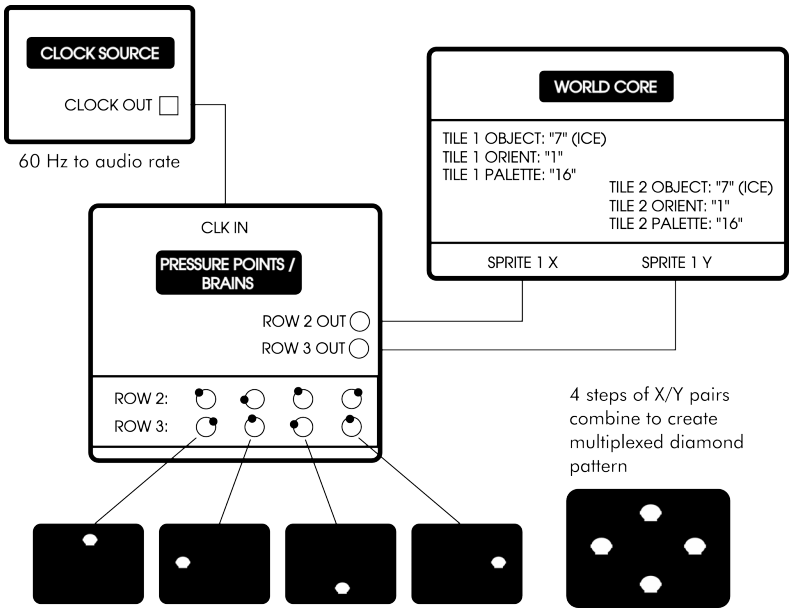
SPRITE multiplexing is often best when used sparingly: for instance, limiting the multiplexing to 2:1, instead of 4:1. Because external

modulators are not synchronized to the video clock, the WORLD CORE cannot guarantee a SPRITE position change will fall exactly during each frame refresh, occasionally resulting in the SPRITE flickering elsewhere on the screen. Although problematic for certain patch types, this aliasing can also be harnessed artistically to produce amazingly complex SPRITE motion, as explained below.

ASYNCHRONOUS SUPER-FRAME-RATE MODULATION (ASFM)

While experimenting with the previous patch you most likely encountered a "rogue copy" of the multiplexed SPRITE occasionally appearing in unexpected areas of the screen. This is an aliasing artifact caused by the lack of synchronization between the SPRITE's position-modulating clock and the refresh rate of the WORLD CORE's NTSC video clock. Even when the position-modulator is tuned to 60 Hz, the two clocks will slowly drift in and out phase with each other. The output of the VOLTAGE CONTROLLED SWITCHES is also not perfect; being an analog voltage, it doesn't immediately snap to position, but instead quickly slews across a range of voltages when adjusting its output. When coupled with the phase drift of the two clocks, the WORLD CORE will occasionally draw a frame just at the exact moment the switch is slewing through an "in-between" value, producing a kind of momentary ghost SPRITE.

By driving the positional changes at speeds higher than the 60 Hz frame-rate, this aliasing artifact can be exploited to produce highly complex patterns of emergent SPRITE movement—an effect we call ASYNCHRONOUS SUPER-FRAME-RATE MODULATION, or ASFM for short. To get started with ASFM, begin by constructing the STEP SEQUENCER-based multiplexing patch below:



TILE 1 and TILE 2 are configured so that a single black screen is created, allowing us to focus on the behavior of the bright red SPRITEs. Each PRESSURE POINTs step consists of an X/Y voltage pair specifying one of SPRITE 1's multiplexed positions. When clocked at 60 Hz, the resulting pattern is a diamond shape.

Begin with the clock source at 60 Hz and slowly increase its frequency by hand. As you increase the STEP SEQUENCER's speed above the frame-rate, you should see the multiplexed diamond pattern explode into a chaotic cloud of SPRITEs. By employing a light touch (and your clock source's fine tune control, if available), it's possible to dial in harmonic ratios that produce stable patterns of SPRITE movement, such as figure eights, spinning wheels, and a variety of strange loops, bifurcations, and lattices.

The process is somewhat analogous to frequency modulation (FM), in which two audio-rate oscillators are placed in a carrier-modulator relationship. In simple FM synthesis, the modulator controls the frequency of the carrier, producing complex sidebands when pitched

higher than the carrier's base frequency. ASFM and FM are so similar, in fact, that it's possible to "sonify" ASFM patterns by constructing a basic FM patch in parallel. By modulating the carrier oscillator's frequency with the same clock source that drives the STEP SEQUENCER, the ASFM and FM branches of the patch are effectively synchronized. Furthermore, if the carrier is tuned to 60 Hz, the FM sidebands and ASFM patterns will synchronize precisely, linking stable visual patterns to consonant FM ratios and chaotic visual patterns to dissonant ones.

ASFM can be achieved with almost any voltage source capable of oscillating at audio rates. Try experimenting with LFOs, cycling ENVELOPEs, audio/video oscillators, and especially oscillators that allow continuous control of wave shape (interpolating from saws to triangles, for instance). Interesting results can also be achieved by only applying ASFM to a single axis, leaving the other axis free for slower movements.

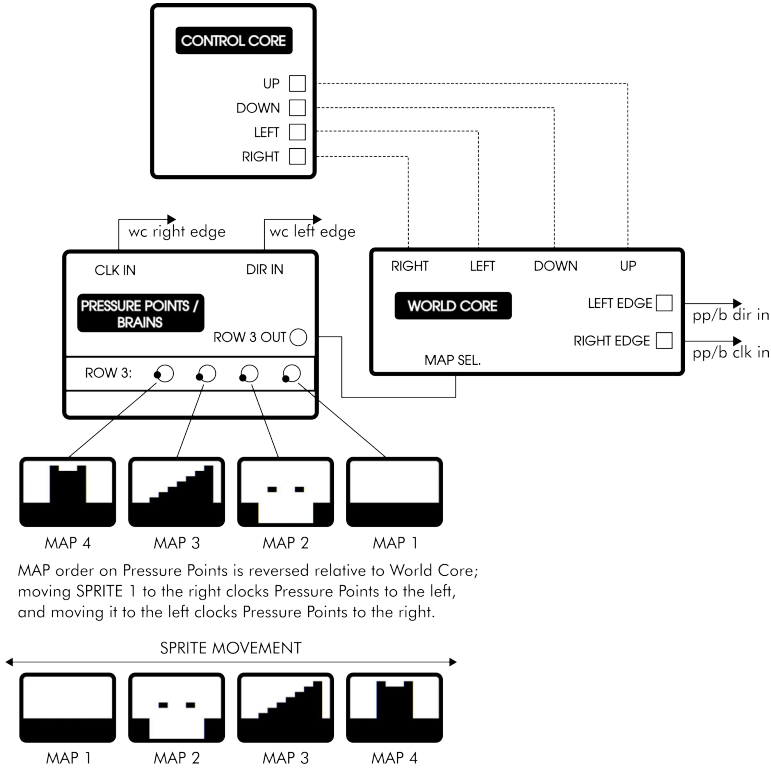
When CLIPPING is turned on, the WORLD CORE performs significantly more operations per second when updating SPRITE position for each frame, limiting the temporal resolution of the SPRITE. When experimenting with SPRITE multiplexing or ASFM patches, CLIPPING should be turned off for best results.

MULTI-MAP COMPOSITON

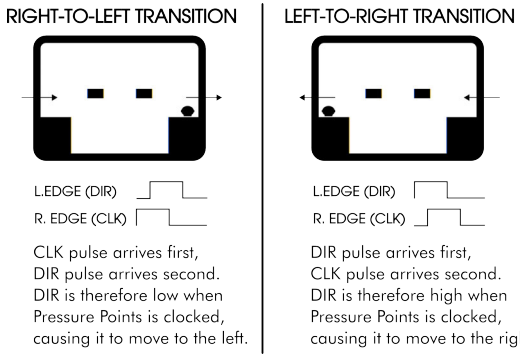
A single MAP can contain plenty of complexity, but sometimes a world design requires more geography than a single screen can hold. Building large multi-MAP worlds with MING MECCA can be challenging, but with the right tools it's possible to create expansive environments that reward the time spent patching them.

The only requirement for composing with multiple **MAPs** is a **STEP SEQUENCER** and a properly organized **WORLD PACK**, but there's a catch: the sequencer must have a "direction" input that uses a **GATE** signal to determine left vs. right step clocking. Without a direction input, your **SPRITEs** will be limited to moving through the environment in a single direction only.

In the patch example below, **PRESSURE POINTs / BRAINs** is used to sequence four **MAPs** into a condensed "Mario-like" environment that consisting of a field, a pit, stairs, and a castle. The basic idea is to detect when **SPRITE 1** is crossing the wrap transition at the edge of the screen, and clock in the next **MAP** in the sequence when it does so. This produces a "flip screen" effect where a large composite world is navigated through interconnected single-screen segments.



As SPRITE 1 moves across the wrap boundary, it causes two closely spaced EDGE COLLISION events. If SPRITE 1 is moving to the right, for example, it will cause the RIGHT EDGE output to fire as it approaches the edge of the screen, immediately followed by the LEFT EDGE output when it reappears on the opposite side The same effect occurs if the SPRITE is moving to the left, except that the order of the EDGE COLLISION events is reversed.



The RIGHT and LEFT EDGE COLLISION outputs are used to clock PRESSURE POINTs and set its direction, respectively. The pulse logic aligns in such a way to clock PRESSURE POINTs left/right in response to SPRITE 1's edge transitions, only backwards: when SPRITE 1 moves to the right, PRESSURE POINTs clocks to the left; when SPRITE 1 move to the left, PRESSURE POINTs clocks to the right. This is why the MAP order on PRESSURE POINTs is backwards in the patch schematic. When the order is inverted according to SPRITE 1's direction of travel, the MAP sequence appears in the correct order, beginning with the bare field and ending in the castle.

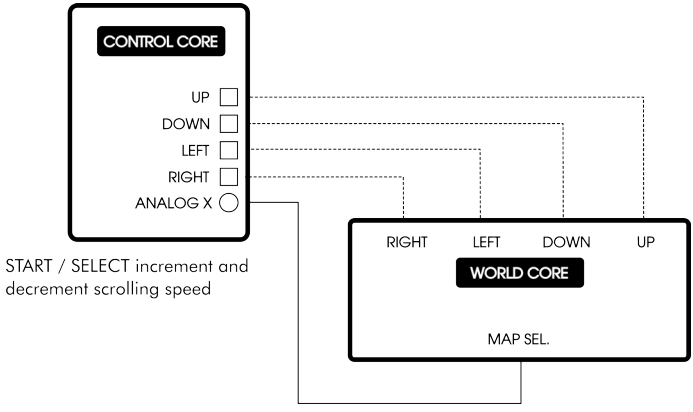
Try combing this patch with a simple gravity-enabled jumping patch and/or walk cycle patch to create more complex worlds.

PRESSURE POINTs ROW 1 and **ROW 2** can be used to sequence other **WORLD CORE** parameters. Try changing the sky color in each **MAP**, or positioning **SPRITE 2** in different locations depending on the scene.

COARSE SCROLLING

In addition to the "flip screen" style of environment just described, it's also possible to create "side scrolling" environments that seamlessly scroll left and right across the display. The **WORLD CORE** isn't capable of performing so-called "smooth scrolling," where the world's terrain moves in single-pixel increments. "Coarse scrolling," however—in which the terrain moves in single-**TILE** increments—can be achieved by combining a very simple **CONTROL CORE** patch and custom designed **WORLD PACK**s.

Special Stage Systems provides an example **WORLD PACK** called *Snake Scroller* to demonstrate coarse scrolling. To download the file, navigate to **specialstagesystems.com/wpacks**. Once *Snake Scroller* has been loaded into the **WORLD CORE**, construct the following patch:



The MAPs in *Snake Scroller* have been built so that each successive screen moves the terrain a single TILE to the left. By using the CONTROL CORE's ANALOG X output to sweep through MAP SELECT, the terrain will scroll in the opposite direction of SPRITE 1's movement. The rate of the ANALOG X function, modified by pressing the START and SELECT buttons, determines the speed of scrolling. The speed of SPRITE 1's movement is adjusted as normal, using the WORLD CORE's SPEED toggle switch. Make sure to select appropriate scrolling and SPRITE movement speeds so that the two are roughly synchronized.

Just like split screen patches, coarse scrolling can be combined with jumping patches to create interactive videogame-like environments.

SPRITE 2 can be "pinned" to a region of the scrolling terrain by modulating its X position from an inverted and offset copy of the ANALOG X output. The **Make Noise** MATHS easily provides this functionality using channels 2 and 4.

APPENDIX B: CONTROL CORE MANUAL

- Introduction
- Unpacking and Inspecting Your Module
- Linking a **WORLD CORE**
- Configuring and Connecting Power
- System Overview and Interface Guide
- Voltage Standards and General Control Paradigms
- Gamepad Operation
- **TURBO** Modes
- Unleashing the **CHAOTIX** Oscillator

INTRODUCTION

Thank you for purchasing a **Special Stage Systems CONTROL CORE**. The **CONTROL CORE** is a gamepad-to-CV interface that supports NES-compatible peripherals. Although it was designed for use with a **WORLD CORE** as part of the **MING MECCA** voltage controlled videogame console, it can also function stand-alone as a powerful performance controller for your modular.

The **CONTROL CORE** features six digital outputs, two analog outputs, and a sophisticated **TURBO** oscillator, all driven by the gamepad's **DIRECTIONAL PAD** and **FACE BUTTONS**. When used together with a **MING MECCA WORLD CORE**, the **CONTROL CORE** facilitates videogame-like manipulation of **SPRITEs**, as well as control of various world-states and video parameters. As a performance controller, the **CONTROL CORE** can be used to generate both discreet pulses (for triggering percussion, for example) and continuous voltage (for controlling oscillator pitch, filter cutoff, etc.). The long cord connecting the module to the gamepad makes it ideal for "remote control" applications where patches need to be manipulated or performed at a distance.

MING MECCA has been designed as an extendable platform, with many more **CORE** modules planned for eventual release. If you would like to be notified as new designs are announced, please consider signing up for the **Special Stage Newsletter** via our contact page at **specialstagesystems.com/contact**.

From all the robots and miscellaneous life forms at **Special Stage Systems**, thank you again, and we hope you enjoy your time with the **CONTROL CORE**.

UNPACKING AND INSPECTING YOUR MODULE

Your package should include the following items:

1. **CONTROL CORE** module
2. 16-to-16 pin ribbon **POWER CABLE** (attached to the back of the module)

If any of these items are missing, please contact **Special Stage Systems** to request a replacement.

Before setting up your CONTROL CORE, it's important to perform a quick visual inspection to make sure the module has not been damaged during shipping. First check the FRONT PANEL and then turn the module around to look at the MOTHERBOARD. When inspecting the MOTHEBOARD, take care to avoid directly touching any exposed components and connections, as this could result in electrostatic damage. Look closely at the board and verify that none of the header pins are bent and that there is no obvious structural damage.

If you think your CONTROL CORE has been damaged or is otherwise defective, do not attempt to install the module. Please contact **Special Stage Systems** at **support@specialstagesystems.com** to open a support ticket.

*Headers are small rows of gold pins that are used to connect ribbon cables to the motherboard.

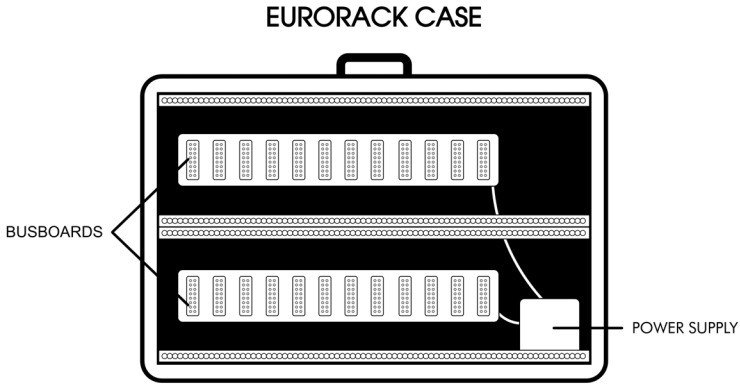
LINKING A WORLD CORE

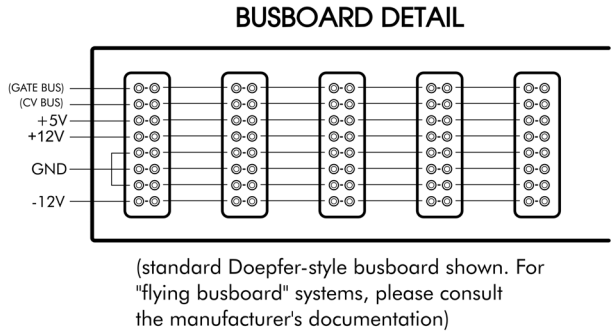
If you will be using your CONTROL CORE in conjunction with a WORLD CORE, it's recommended that you LINK the two together. The LINK CABLE is supplied with your WORLD CORE, along with full instructions on performing the procedure in the MING MECCA USER'S GUIDE.

CONFIGURING AND CONNECTING POWER

MING MECCA modules are designed for use within EURORACK modular synthesizer systems. In order to use your CONTROL CORE module you will need to install it in a EURORACK case and supply it with EURORACK-compatible power. EURORACK cases usually have a power supply and power distribution system built in.

Caution should be exercised when installing any new module in your system. Although **Special Stage Systems** has taken steps to protect your CONTROL CORE from inverted polarity, it is impossible to predict all potential scenarios given the open nature of the EURORACK standard. **Special Stage Systems accepts no liability for damage to the CONTROL CORE or to any other connected hardware due to reversed, offset, or otherwise incorrect power connection.** Please follow this guide closely and double-check the ribbon cable before applying power to your modular.



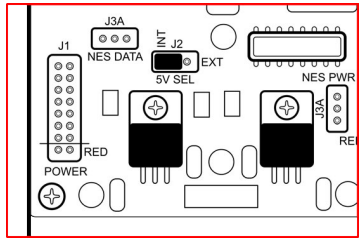


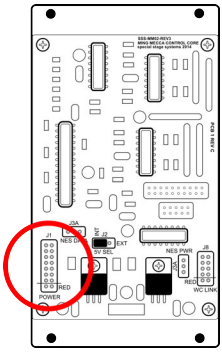
EURORACK modules receive power via ribbon cables that attach to the case's BUSBOARDS. Most BUSBOARDS use headers that are **unkeyed**, which means that it is **possible to plug power in backwards**. Accidentally inverting the polarity can damage not only the reversed module, but any modules connected to the same BUSBOARD as well.

CONNETING POWER

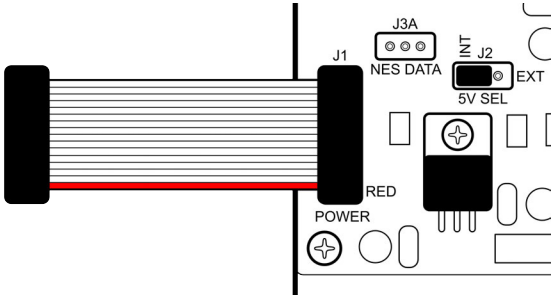
Before connecting power to your CONTROL CORE, first verify that your case meets the minimum. You will need a minimum of 14HP (2.8") of free horizontal space to install the CONTROL CORE. Your case must also be 1.7" deep to house the module's internal circuitry. Finally, your power supply must be able to provide at least 210mA of current on the +12V rail.

Now locate the 16-pin power connector on the MOTHERBOARD.

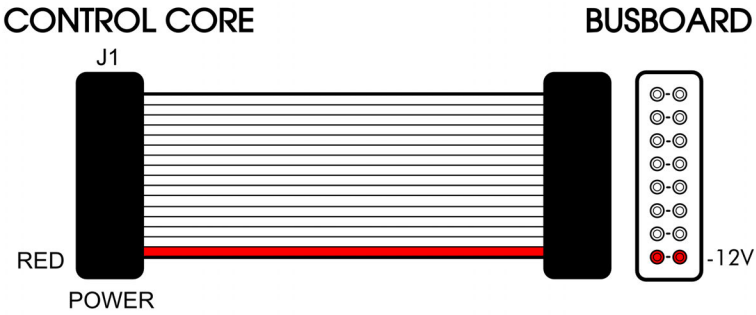




The 16-to-16 pin POWER CABLE should already be connected to the module. Note the position of the cable's RED STRIPE. Verify that the RED STRIPE is aligned with the location marked "RED" on the power connector. If the cable is not properly aligned, remove and reposition it so that the alignment is correct.



Connect the other end of the cable to your power supply, making sure that the RED STRIPE aligns with the -12V pins on the BUSBOARD header.



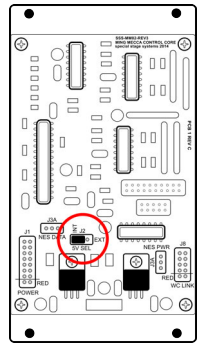
If you are unsure about the orientation of your case's BUSBOARD headers, consult the manufacturer's documentation for more information.

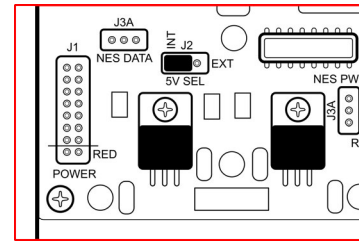
CONFIGURING THE +5V SUPPLY

By default, the CONTROL CORE generates all the power it needs from the +12V rail. The CONTROL CORE can also be configured to use an external +5V supply to power its digital components. Not all EURORACK cases provide +5V power, and many that do are actually derivative of the +12V supply. In EURORACK cases that provide truly independent +5V power however, configuring the CONTROL CORE to use the external supply can lessen the load on the +12V rail.

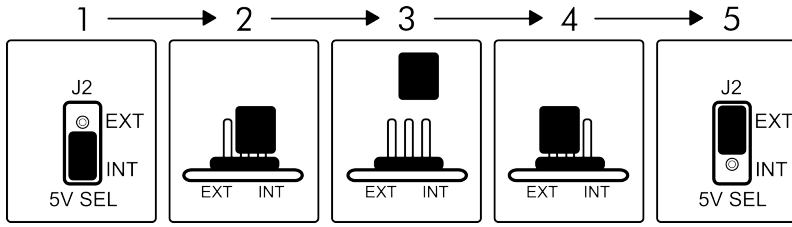
> If you don't know whether your case provides independent +5V power, it's best to leave the CONTROL CORE in its default configuration. The option to use an external +5V supply is for advanced users who are looking to maximize the efficiency of their system's power supply.

To configure the CONTROL CORE for use with an external +5V power supply, first locate the 5V SELECT JUMPER on the MOTHER-BOARD.





Remove the jumper and reinstall it in the "EXT" position, as shown in the diagram below.



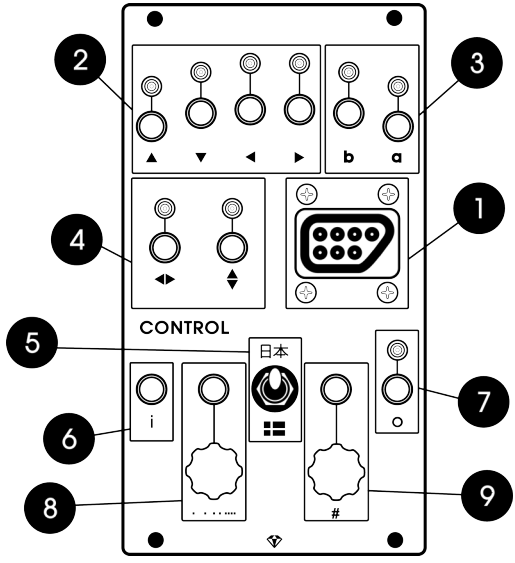
WARNING: the 5V SELECT JUMPER must be installed in either the "EXT" or "INT" position before applying power. **Never attempt to power the CONTROL CORE with the 5V SELECT JUMPER removed.**

WARNING: do not configure the CONTROL CORE for external +5V power if your case does not have a working +5V rail. **Installing a CONTROL CORE configured for external power in a case that doesn't supply +5V may damage the module.**

The CONTROL CORE will now use an external +5V supply for its digital components. Note that the +12V rail is still required in order to power the analog sections of the MOTHERBOARD; always use a full 16-to-16 pin POWER CABLE when powering the module, regardless of the 5V SELECT JUMPER setting.

SYSTEM OVERVIEW AND INTERFACE GUIDE

The CONTROL CORE's interface is divided into two main sections. The top half of the module deals with the gamepad-to-CV conversion, while the bottom half contains the complex TURBO oscillator and its corresponding CV inputs. The following interface guide contains a complete list of every control on the CONTROL CORE, and where to find them on the panel.



1. **GAMEPAD PORT**
2. **DIRECTIONAL OUTPUTs** (GATE outputs) (**UP**, **DOWN**, **LEFT**, and **RIGHT**)
3. **FACE BUTTON** GATE outputs (**A** and **B**)
4. **ANALOG X** and **ANALOG Y** CV outputs

5. **TURBO MODE** toggle
6. **TURBO INPUT** (GATE / TRIGGER input)
7. **TURBO OUTPUT** (GATE / TRIGGER output)
8. **RATE** knob and CV input
9. **NUMBER** knob and CV input

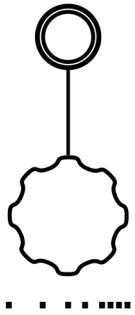
VOLTAGE STANDARDS AND GENERAL CONTROL PARADIGMS

- CV inputs: 0-5V
- GATE and TRIGGER inputs: +2.2V logic threshold
- GATE and TRIGGER outputs: +10V high
- CV outputs: 0-10V
- Knobs transform into attenuators when using CV inputs
- GAMEPAD PORT: NES-compatible controllers including the original NES gamepad, "dogbone" gamepad, and the NES Advantage joystick controller.

Any signal can be patched into the CONTROL CORE's CV and GATE / TRIGGER inputs so long as it doesn't exceed EURORACK power levels (-12V to +12V). In order to get the best results however, it's useful to attenuate and/or rectify signals to fit the CONTROL CORE's responsive range (i.e., the range of voltages which produce noticeable effects).

The responsive range for CV inputs is 0-5V. GATE and TRIGGER inputs are set to a +2.2V logic threshold. Any level below +2.2V is con-

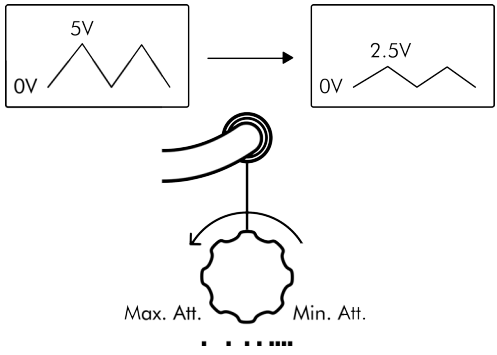
sidered a logical low ("OFF"), and anything above +2.2V is considered a logical high ("ON").



The RATE and NUMBER parameters are accessible through manual knobs as well as CV input jacks. Knobs and their corresponding jacks are depicted on the panel with a vertical connecting line.

The knobs generate 0V when turned fully counterclockwise, and +5V when turned fully clockwise. When a signal is patched into the corresponding CV input, the knob no longer directly controls the parameter. Instead, it becomes an attenuator that scales the voltage at the CV input. When the knob is fully clockwise no attenuation is applied. Attenuation gradually increases as the knob is turned counterclockwise.

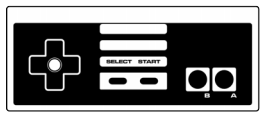
By adjusting the amount of attenuation, it's possible to format higher voltage signals to match the CONTROL CORE's 0-5V responsive range, or to truncate the number of selectable states as shown above. Note that attenuation only affects the top end of the voltage range. Additional (third party) modules are required to raise the low end of the range (DC offset summing), or to eliminate negative voltage (rectification).



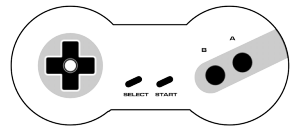
The CONTROL CORE's DIRECTIONAL PAD and FACE BUTTON outputs generate +10V GATEs. A relatively high voltage is desirable for GATE/TRIGGER outputs because it allows them to be patched to multiple simultaneous locations without experiencing problematic voltage drops. The ANALOG X and ANALOG Y CV outputs follow suit, generating 0-10V. Each output has a corresponding LED that lights to indicate its state.

GAMEPAD OPERATION

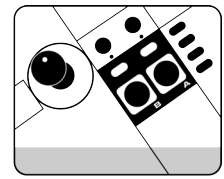
Before you can start using your CONTROL CORE, you need to select a gamepad to use with it. The CONTROL CORE has been designed to support NES (Nintendo Entertainment System) gamepads. The following gamepads are supported:



Original Gamepad



"Dogbone" Gamepad



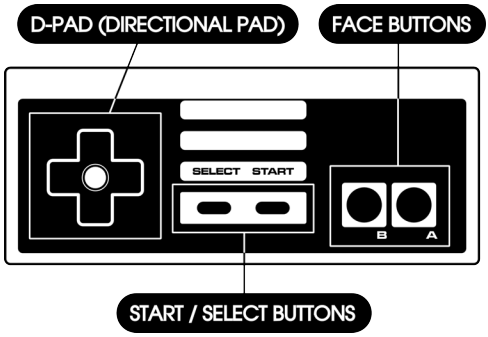
NES Advantage

Once you've obtained the gamepad of your choice, simply plug its interface cable into the GAMEPAD PORT. Gamepads can be hot-swapped without power cycling the CONTROL CORE. Some random GATE activity may be generated by the D-PAD and FACE BUTTON outputs when connecting a new gamepad—this is normal and no cause for alarm.

> All NES accessories other than those listed above are **not supported**, including the Power Glove, Zapper light gun, and Power Pad. Use of unsupported accessories may damage the CONTROL CORE. **Special Stage Systems accepts no responsibility for damage caused by the connection of incompatible NES accessories.**

Now try selecting a direction on the D-PAD (UP, DOWN, LEFT, or RIGHT) or pressing one of the FACE BUTTONS (A or B). You should see the corresponding GATE output LEDs flash on the CONTROL CORE in response. All 8 GATE outputs are fully-buffered and produce 10V sig-

nals. Outputs will remain high as long as the button is depressed, and flip low when the button is released.



Dedicated **ANALOG X** and **ANALOG Y** outputs provide additional control by converting **D-PAD** activity into analog **CV**. This is best understood as a kind of "virtual analog joystick." Holding **UP** will increase the voltage at the **Y** output, while holding **DOWN** will decrease it; similarly, holding **RIGHT** will increase the voltage at the **X** output, while holding **LEFT** decreases it. The voltage will retain its level even after the button is released.

The rate at which the **ANALOG X** and **ANALOG Y** voltages change is controlled by the **START** and **SELECT** buttons. Pressing **START** will increase the rate of change, while pressing **SELECT** will decrease it. There are nine different rates to choose from in total.

> The **START/SELECT** buttons modify the **ANALOG X/Y** voltage in discreet increments; holding either button will **not** produce steadily increasing or decreasing speed. To increase the speed by three units, for instance, the **START** button should be pressed three times sequentially.

TURBO MODES

The TURBO oscillator may be placed into two separate modes, each with their own unique behaviors: JAPAN and SWEDEN, as indicated by the Japanese text and Swedish flag labeling the MODE toggle switch.

JAPAN MODE

In JAPAN mode, the CONTROL CORE mimics the functionality of rapid-fire modes found on traditional videogame controllers. When a GATE signal is applied to TURBO IN, TURBO OUT will generate a square wave at a frequency determined by the RATE CV. The square wave will continue to be generated as long as the GATE at TURBO IN is held high, and will immediately terminate when the GATE flips low. The NUMBER CV is used to introduce a small delay between the detection of a GATE at TURBO IN and the generation of the square wave at TURBO OUT.

> TURBO IN is internally normalled to the B output, facilitating the easy creation of held-button style rapid-fire patches.

SWEDEN MODE

SWEDEN mode was inspired by the "retrigger" functionality of the Elektron Machinedrum, a popular digital drum machine. When a GATE is applied to TURBO IN, TURBO OUT generates a square wave of a fixed number of cycles before terminating automatically.

The square wave frequency is once again determined by the RATE CV, and the number of cycle repetitions is determined by the NUM-

BER CV. The number of repetitions ranges from 1-32, not including the first cycle (i.e., selecting one repetition will produce two pulses in total, and so on). When the NUMBER knob is fully clockwise (or +5V is applied to its CV input), the TURBO oscillator is set for infinite repetitions, and will cycle indefinitely until the NUMBER CV is changed or SWEDEN mode is exited.

When the NUMBER CV is set to 0V, the TURBO oscillator is placed into a special "0-repetition" sub-mode where the resulting square wave is only one cycle long. Applying a GATE to TURBO IN will therefore result in a single pulse appearing at TURBO OUT, making this sub-mode very useful as a GATE-to-TRIGGER converter. The RATE CV controls the width of the resulting TRIGGER.

UNLEASHING THE CHAOTIX OSCILLATOR

The CONTROL CORE can be optionally transformed into an unusual digital noise oscillator by entering the famous "Konami Code" when the module boots. To summon the CHAOTIX OSCILLATOR, follow these steps:

1. Make sure your gamepad is connected to the CONTROL CORE, and then power on your modular while holding the gamepad's SELECT and B buttons.

2. When the top row of LEDs begins to flash, release the SELECT and B buttons and quickly enter the following code:

   UP, UP, DOWN, DOWN, LEFT, RIGHT, LEFT, RIGHT, B, A, START

Once the CONTROL CORE has transformed into a CHAOTIX OSCILLATOR, the gamepad will cease to function. The TURBO OUTPUT now

becomes an audio output, and the RATE and NUMBER CV inputs adjust the behavior of the oscillator. To resume normal operation, simply power cycle the module.

APPENDIX C: TROUBLESHOOTING CHART

SYMPTOM	CAUSE	SOLUTION
Display is black or shows "No Signal" warning	Video SELECT switch is in the downward position	Place the SELECT switch in the upward position
	Module is not receiving power	Double check POWER CABLE direction and 5V SELECT JUMPER position (see pg. 15)
	Video cable is defective or improperly connected	Swap for known-good cable; make sure cable is connected to the CVBS jack, not the AUX jack
	Corrupt WORLD PACK inserted into CARTRIDGE SLOT	Eject the corrupt CARD and RESET the system
	Display is defective / incompatible	Try another display
Display is garbled or shows unexpected color / graphics	GLITCH MODEs are engaged	Make sure the ?MA*, ?TI*, and ?PA* toggles are in the downward position
	Display is not NTSC-compatible	Try another display
One or more SPRITEs is not visible and / or unresponsive to positional controls	CLIPPING is turned on, trapping the SPRITE	Place the CLIP toggle in the downward position
	GRAVITY is turned on, forcing the SPRITE off-screen	Place the GRAVITY toggle in the downward position

SPRITEs not visible / unresponsive (continued)	SPRITE is showing SPRITE DATA that is completely transparent	Change OBJECT and ORIENT knobs or CV; eject WORLD PACK (if applicable) and test with default graphics
	CV range is miscalibrated	See below
CV inputs / knobs are unresponsive, over-responsive, or do not cover full control ranges	1V CALIBRATION trimpot is set incorrectly	Perform calibration procedure (see pg. 4)
	CV SELECT JUMPER is missing	Locate or replace jumper
LINKED CONTROL CORE fails to move SPRITE 1 or behaves unexpectedly	LINK CABLE is inverted or misaligned	Correct cable orientation

INDEX

D

E

F

G

N

O

P

R

S

T

U

V

W

X

?

INTO THE UNKNOWN

This guide has demonstrated **MING MECCA**'s basic capabilities through simple patch examples and tutorials. We've deliberately left complex, fully-developed patches out of the discussion, not only for pedagogical reasons, but also because the things you can create with **MING MECCA** are so diverse. Exploring the possibilities is half the fun, and we wouldn't want to spoil the adventure.

Instead, we'll leave you with a small word of big-picture advice. The systems **MING MECCA** generates can be described in the language of "videogames," but in truth they're closer to something like "procedural etudes"—playground worlds featuring a small number of deceptively simple mechanisms that, when investigated, reveal the deep ambiguity and behavioral richness of the analog networks they interact with.

MING MECCA wraps spatial metaphors around voltage controlled machines, transforming them into the ontological building blocks of toy realities. Don't be afraid to combine those blocks in experimental, unconventional ways. The most exciting worlds are often the most surprising ones, discovered by creatively repurposing familiar tools into unfamiliar toolboxes.

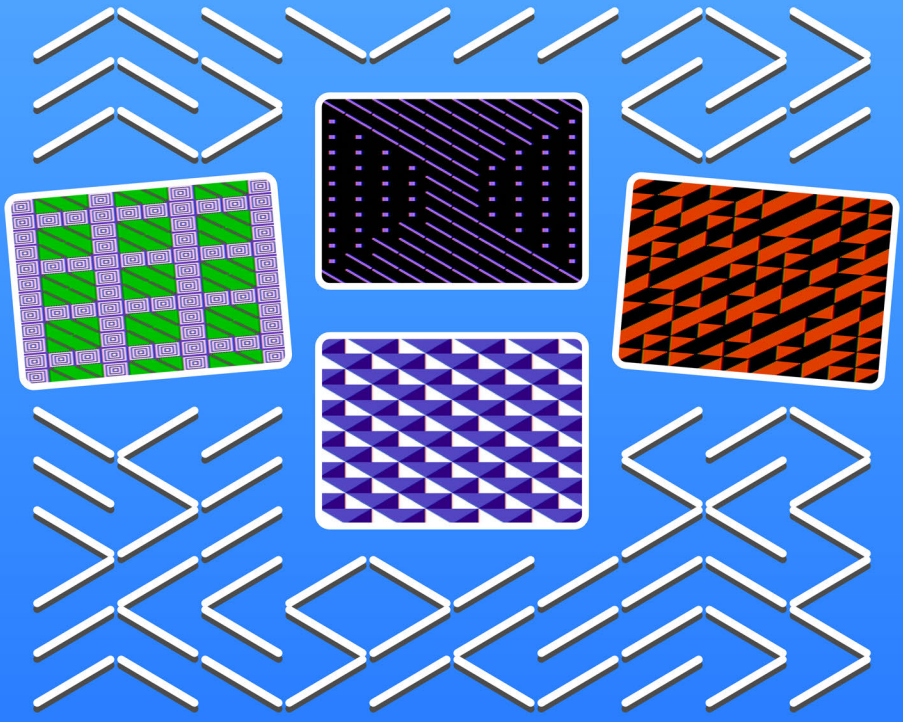
We wish you many strange journeys and exciting discoveries as you push at the frontiers of your voltage controlled worlds. And we'd love to see what you've been building. If you have recordings of patches that you'd like to share, please get in touch with us!

Show us your worlds at
worlds@specialstagesystems.com

Or reach us on facebook at
facebook.com/specialstagesystems

NOTES

TANGRAM PARK

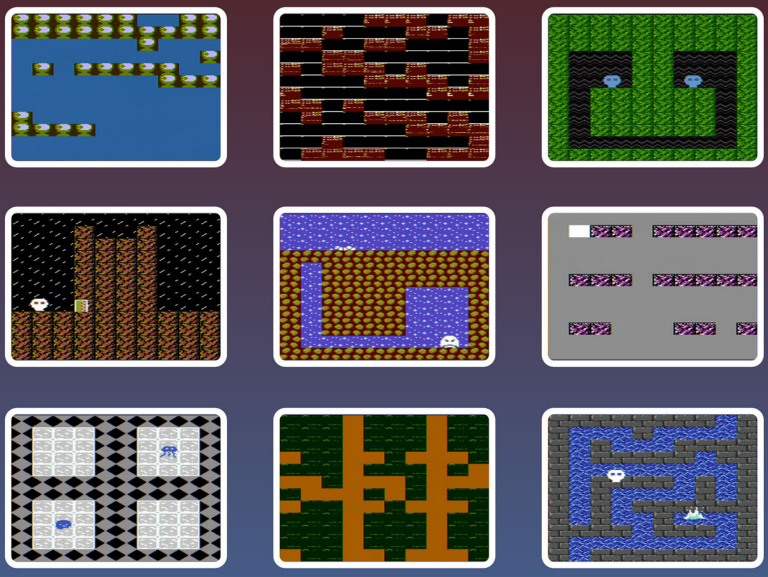


SPECIALSTAGESYSTEMS.COM/WPACKS

Lose yourself in Tangram Park, an abstract World Pack from Special Stage Systems.

- **Psychedelic animations**
- **Complex patterns**
- **Color-cycling palettes**
- **And more!**

INCLUDES 10 PRINT construction kit



special stage systems

New worlds in electric voltage.

Need help with installation, operation, or repairs?
Email support@specialstagesystems.com