µRings SE

Eurorack Module

TD-URSE-B

Designed by Daniel & Michael Gilbert of *Tall Dog Electronics* in Western Massachusetts
tall-dog.com | **urings-se.com**

Based on the *Rings* module designed by Olivier Gillet of *Mutable Instruments*
mutable-instruments.net

Manufactured by *Elecrow* in Shenzhen, China
elecrow.com

Assembled at *Rust Temple* in Easthampton, MA
rusttemple.today

Source materials are available on *GitHub*
github.com/loglow/uRings_SE

Contents

About

Rings is a resonator, transforming unpitched excitations (clicks, bursts, etc.) into full-bodied pitched sounds. *Rings* models the bar, the tube, or the strings that you can make vibrate by providing an external signal.

This redesigned *µRings SE* (Special Edition) is a new revision of *Rings* that miniaturizes the module from 14HP to 8HP and redesigns its front panel layout.

µRings SE remains fully compatible with future upstream *Rings* firmware upgrades and any other alternate firmwares that are compliant with the standard *Rings* design.

Included Parts

µRings SE includes a standard 16-pin to 10-pin Eurorack power cable and two sets of mounting screws for racks/enclosures with either M2.5 or M3 threads.

The **TD-URSE-B/B** variant has a matte black front panel with gold markings made from FR-4 (glass-reinforced epoxy laminate) material. The included mounting screws are black.

The **TD-URSE-B/S** variant has a satin silver front panel with black markings made from anodized aluminum material via the Metalphoto process. The included mounting screws are silver.

History

The original *Rings* (**mutable-instruments.net/modules/rings**) was designed by Olivier Gillet of *Mutable Instruments* (**mutable-instruments.net/about**) under a CC BY-SA 3.0 (**creativecommons.org/licenses/by-sa/3.0**) license. *Rings* is available as a 14HP Eurorack module.

Source: **github.com/pichenettes/eurorack**

Warranty

This product is covered under warranty for one year following the date of purchase as indicated on the original sales receipt. This warranty covers any defect in the manufacturing of this product. This warranty does not cover any damage or malfunction caused by incorrect use—such as, but not limited to, power cables connected backwards, excessive voltage levels, exposure to extreme temperature or moisture levels, physical damage due to impact, or any aftermarket physical or electrical modifications or repairs.

This warranty covers either repair or replacement, at our sole discretion. This repair or replacement is subject to verification of the defect or malfunction and proof of purchase as confirmed by showing the model number on an original dated sales receipt. Shipping and handling fees are to be paid for by the customer.

Please contact **support@tall-dog.com** for a return authorization prior to shipping anything to us. Please understand that we will not be able to service units under warranty that have been modified or previously repaired by the customer or a third party.

Polyphony Selection

Rings operation is governed by two settings controlled by buttons located near the center of the module: the **POLY** button and the **MODE** button. Pressing either button cycles between three states. The current state of each setting is indicated by the color of its associated LED: green, amber, or red.

The **POLY** button selects the polyphony of the module:

LED Color	**Notes**
Green	1
Amber	2
Red	4

Enabling four note polyphony doesn't mean that four CV input jacks will magically appear on the module, but simply that four notes played in sequence will nicely overlap without cutting off each other's tails.

To play chords, you will need to *strum* the module by playing a rapid sequence of notes—something you might have already encountered with the *Braids* module's **PLUK** model.

Note that *Rings* might reduce the number of harmonics in the generated signals to cope with higher polyphony settings.

Resonator Selection

The **MODE** button selects from the three available types of resonators. They are:

Modal Resonator (Green)

Modal synthesis works by simulating the phenomena of resonance at play in vibrating structures, that is to say the way a string or plate (for instance) will absorb certain frequencies while it will *ring* at some other frequencies, called the *modes*.

When we pluck a string, strike a drum or blow in a tube, the short burst of energy of the blow/impact contains many frequencies. Some of these fall outside of the modes, and are absorbed. Some of these excite the modes, producing a stable, pitched sound. Each mode corresponds to a harmonic or partial in the spectrum of the sound, and is modeled by a band-pass filter.

The Q factor of the filter determines how sustained the oscillations are of the corresponding partial. Various materials or structures are characterized by different relationships between the frequencies of their modes, which *Rings* recreates.

Sympathetic Strings (Amber)

Some interesting string instruments (such as the sitar or sarod) make use of strings that are not directly struck/plucked by the musician, but which are just responding to the vibration of other strings, and add extra overtones or undertones.

Rings simulates this phenomenon with a bunch of virtual strings (made with comb filters) allowing the addition of extra tones to an incoming audio signal. The tuning ratio between these strings can be altered.

Modulated/Inharmonic String (Red)

This last method is perhaps the most familiar, since it's based on the extended Karplus-Strong method: the excitation signal is sent to a comb filter with an absorption filter, simulating the multiple reflection of a wave propagating on a string and being absorbed at its ends.

However, to bring more variety to the sound, *Rings* adds three extra ingredients to this classic: a delay-compensated all-pole absorption filter creating more drastic plucking effects, delay time modulation emulating the sound of instruments with a curved bridge (like the sitar or tanpura), and all-pass filters in the delay loop, shifting the position of the partials and recreating the tension of piano strings or completely bonkers inharmonic timbres.

Making Connections

Ideally, *Rings* would need three input signals:

1. A trigger signal on the **STRUM** input, which indicates that the currently playing note should fade away, and that a new note is starting.
2. A CV signal on the **V/OCT** input, which controls the frequency of the note.
3. An audio signal on the **IN** input, which will hit, strike or caress the resonator.

Because it might not always be possible to get these three signals in your setup, *Rings* makes the following assumptions:

1. If nothing is patched to the **IN** audio input, the module will synthesize an excitation signal whenever a note is strummed. This excitation signal is either a low-pass filtered pulse or a burst of noise, depending on the resonator type.
2. If nothing is patched to the **STRUM** audio input, the module will determine that a new string should be strummed either by:
    - Detecting note changes on the **V/OCT** input, or
    - Detecting sharp transients within the **IN** audio signal if nothing is patched to the **V/OCT** input.

Making Connections *Cont'd*

If there's one take-home message from this:

You can play *Rings* perfectly with just one CV output taken from a sequencer or sample & hold module!

The note changes on the CV input will be interpreted as note changes and the module will produce a suitable excitation signal internally for these note changes to be heard.

Front Panel

FREQ — Coarse frequency, adjusted by semitone increments. This control spans 5 octaves. Note that the **FREQ** CV input is normalized to 1/12 V, allowing its attenuverter to be used as a fine frequency control when no patch cable is inserted.

STRUCT — Harmonic structure. With the modal resonator, this parameter controls the frequency ratio between partials (and by doing so, the perceived structure—plate, bar, string). With the sympathetic string resonator, this parameter controls the set of frequency ratios between all strings (with virtual notches at octaves or fifths). Finally, with the modulated/inharmonic string resonator, this parameter controls the amount of modulation and detuning of the partials.

BRIGHT — Brightness. Adjusts the level of higher harmonics in the signal, by the simultaneous action of a low-pass filter on the exciter signal (closed at 8 o'clock, fully open at 12 o'clock), and the damping filter (or Q factor of the higher modes) on the rest of the course of the potentiometer. Low values simulate materials like wood or nylon. High values simulate materials like glass or steel.

DAMP — Damping. Controls the decay time of the sound, from less than 100 ms to about 10 seconds.

Front Panel *Cont'd*

POS — Excitation position. Controls on which point of the string/surface the excitation is applied. Applying the excitation right in the middle of the surface will cause, by symmetry, the even harmonics to cancel each other, resulting in a *hollow* sound reminiscent of a square wave. This setting will remind you of the PWM control on a square oscillator, or of the comb-filtering effect of a phaser.

POLY — Polyphony setting. Selects between monophonic, duophonic and quadriphonic operation.

MODE — Resonator type. Selects between modal, sympathetic and string resonators.

STRUM — Strumming trigger input, for polyphonic operation. Whenever a trigger is received on this input, the module freezes the currently playing voice and lets it decay, and starts a note on the next voice. Normalized to a step detector on the **V/OCT** input and a transient detector on the **IN** input.

IN — Audio input for the excitation signal. Modular levels are expected. Normalized to a pulse/burst generator that reacts to note changes on the **V/OCT** input.

V/OCT — CV input. Controls the main frequency of the resonator.

ODD and **EVEN** — Odd and even audio outputs:

- In monophonic mode, these two outputs carry two complementary components of the signal (odd and even numbered partials with the modal resonator, dephased components due to picking position and pickup placement with the string resonators).
- In polyphonic mode, the signal is split into odd and even numbered strings/plates.
- Note that you need to insert a jack into each output to split the signals. When only one jack is inserted, both signals are mixed together.

Calibration

The module is factory-calibrated using precision voltage sources. Follow this procedure only if you want to compensate for inaccuracies in your CV sources, or if your module has lost its calibration settings following a fault or the installation of alternative firmware.

To calibrate the unit:

1. Disconnect all CV inputs.
2. Connect the note CV output of a well-calibrated keyboard interface or MIDI-CV converter to the **V/OCT** input jack.
3. Connect a patch cable to the **FREQ** CV input. Leave the other end of the cable unplugged.
4. Press and hold both the **POLY** and **MODE** buttons for two seconds. The **POLY** LED will blink amber.
5. Play a C2 note, or send a 1V signal from your CV source.
6. Press the **POLY** button. The **MODE** LED will blink amber.
7. Play a C4 note, or send a 3V signal from your CV source.
8. Press the **POLY** button.

If calibration was successful, the module will return to its normal state. Otherwise, both LEDs will blink red for a few seconds.

Firmware Warnings

Before starting the audio firmware update procedure, please double-check the following:

- Make sure that no additional sound (such as email notification sounds, background music, etc.) from your computer will be played during the procedure.
- Make sure that your speakers/monitors are muted or not connected to your audio interface—the noises emitted during the procedure are aggressive and can harm your hearing.
- On non-studio audio equipment (for example the line output from a desktop computer), you might have to turn up the volume to the maximum.

Firmware Update

Unplug all inputs/outputs from the module. Connect the output of your audio interface or sound card to the **IN** input jack. Set the **FREQ** knob to 12 o'clock. Power on your modular system while holding the **POLY** button, then release. Both LEDs should blink amber.

When you are all set, play the firmware update file into the module. While the module receives data, the **POLY** LED will blink green, and the **MODE** LED will monitor the signal level. Signal reception is optimal when the **MODE** LED is green or yellow. Try adjusting the **FREQ** knob to change gain. The module will periodically pause to write data to its permanent memory. Both LEDs will be amber during these pauses. When the end of the audio file is reached, the module will automatically restart. If it doesn't restart, please retry the procedure.

If the signal level is too weak, the LEDs will blink red in an alternating pattern. Press the **POLY** button and retry with a higher gain. If this doesn't help, try the procedure from a different computer or audio interface, and make sure that no other equipment (such as an equalizer or FX processor) is inserted in the signal chain.

Serial Programming

The module can also be programmed using a serial data connection. This is most easily accomplished by using a USB-to-serial chip such as the popular *FT232R* which can be found in many standalone breakout boards and cables, including the *FTDI TTL-232R-3V3* as well as various equivalents from *Adafruit*, *SparkFun*, and others. Data signals should be at 3.3V levels.

The serial connection should be hooked up to the white shrouded 6-pin header labeled **FTDI** (H6) on the module's PCB. This is a *JST-XH* header that works very well with the 0.1″ pitch female connectors that are frequently found on these cables. Only pins **1 (GND)**, **4 (RX)**, and **5 (TX)** are used, and the location of pin 1 is closest to the top of the board. Power must be supplied separately via a Eurorack power cable connected to the **POWER** (H7) header.

To prepare the processor for serial programming:

1. Power-on the module.
2. Press and hold both the **BOOT** (S1) and **RESET** (S2) buttons at the same time.
3. Release the **RESET** button.
4. Wait a moment, then release the **BOOT** button.

The processor is now ready to be programmed.

JTAG Programming

The module can also be programmed using a JTAG programmer connected to the black shrouded header labeled **JTAG** (H5) on the module's PCB. Power must be supplied separately via a Eurorack power cable connected to the **POWER** (H7) header.

An example of this kind of programmer is the *Olimex ARM-USB-OCD*. Their *ARM-JTAG-20-10* adapter is also necessary in order to accommodate the module's miniature 0.05″ pitch 10-pin header.

No additional steps are necessary to prepare the module for programming via this method.

If you are interested in serial or JTAG programming, please explore the *Vagrant environment for Mutable Instruments modules hacking* which can be found at **github.com/pichenettes/mutable-dev-environment**

Other programming resources can be found at **forum.mutable-instruments.net/t/4336** (*Mac Tutorial: How to compile and upload the firmware of MIs eurorack modules*) and at **medium.com/music-thing-modular-notes/a08173cec317** (*How to get started writing your own firmware for Mutable Instruments Clouds*), the latter of which was written for the *Clouds* module but is also applicable to *Rings*.

Documentation

Much of the information in this manual was created by Olivier Gillet of *Mutable Instruments* and released under the *CC BY-SA 3.0* license.

Modifications and additions to this material were created by *Tall Dog* and released under the compatible *CC BY-SA 4.0* license.

For an overview of the implications and terms of these licenses, please read the following page.

CC BY-SA 4.0

This is a human-readable summary of (and not a substitute for) the license, which can be found at **creativecommons.org/licenses/by-sa/4.0/legalcode**

You are free to:

- **Share** — copy and redistribute the material in any medium or format.
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Support

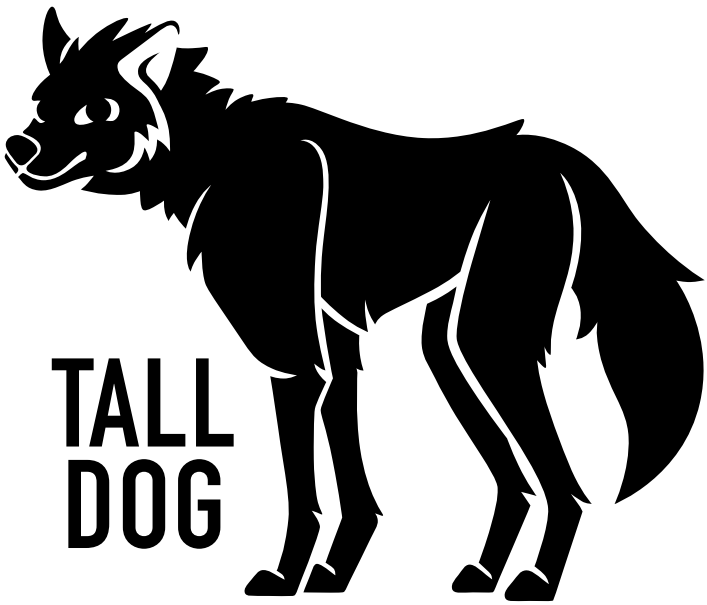
For all support inquiries, please send an email to **support@tall-dog.com**

Who We Are

Tall Dog Electronics is located in the Pioneer Valley region of Western Massachusetts. *Tall Dog* has primarily focused on producing a variety of breakout boards for the *Teensy* microprocessor development platform and conducts the majority of its business via the *Tindie* marketplace. *Tall Dog* released their first Eurorack module, *µBraids SE*, in late 2017.

Michael Gilbert is a composer, recording artist, and teacher of electronic music, for over 40 years. His music is a creative mix of electronic, jazz, world, and contemporary classical idioms, and is available on 9 albums of original work as *Michael William Gilbert*. The music has featured Adam Holzman, Mark Walker, Peter Kaukonen, David Moss, and Tony Vacca. He has also designed and built electronic music equipment, using it in his own studio and making it available to other musicians.

Daniel Gilbert is a designer and engineer with a background in film, photography, and animation. After graduating from *Hampshire College* he spent the next several years working in the Los Angeles film industry. He now resides in Easthampton, Massachusetts, where he designs, builds, and distributes *Tall Dog* products.



µRings SE User Manual
Revised 2018-11-05