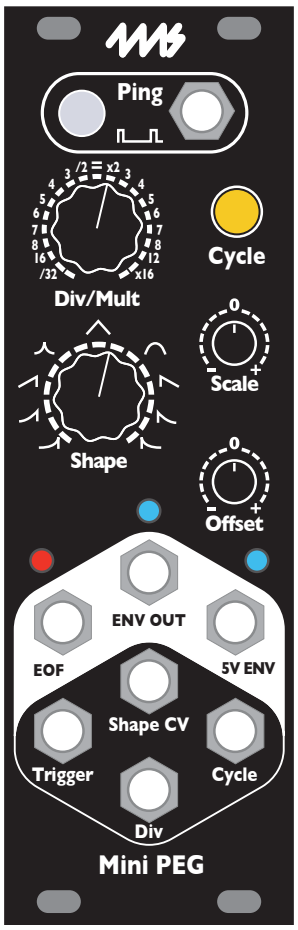
Mini PEG
4ms Company
User Manual 1.0a – September 12, 2022



The **Mini PEG** is a clock-synced envelope generator and LFO whose envelope times are set by the time between clock pulses or “pings”. The **Mini PEG** is the latest version of the classic 4ms Pingable Envelope Generator.

Mini PEG features:

- Envelope length set by incoming clocks or “pings”
- **Ping** button (tap tempo) or external clock/triggers set the envelope time
- **Div/Mult** knob and CV jack divide and multiply ping clock from /32 to x16
- **Shape** knob with full CV control for waveshaping the output envelope, includes various combinations of exponential, linear, logarithmic and interpolated curves
- **Cycle** button and gate jack for looping envelopes
- **Scale** knob acts as attenuating inverter for main envelope output
  - Maximum 9.5V peak-to-peak
- **Offset** knob for setting DC offset, ranging from -9.4V to +9.4V
- **End-of-Fall** (**EOF**) output jack goes high when envelope finishes a fall portion and low when envelope begins a fall portion
  - Jack can also be configured as End-of-Rise, Half-Rise, or Tap Clock Out
- **ENV OUT** jack outputs scaled/offset envelope
- **5V ENV** jack output always produces a 0V to +5V envelope, independent from scale/offset
- **Trigger** jack starts or re-starts an envelope
  - Can be configured as Quantized Trigger, Async Trigger, or Async Gate
- 8HP Eurorack module

Table of Contents

Setting up your Mini PEG

1. Power off your Eurorack system.
2. On the back of the **Mini PEG** you will see a 16-pin header. Connect one end of the power cable to a 16-pin Eurorack power header on your power supply distribution board and the other end to the **Mini PEG** with the red stripe on the power cable oriented towards the bottom of the module.
3. Using the included screws, securely attach the **Mini PEG** to the rails of your case.
4. Power on your Eurorack system.

*Note: The **Mini PEG** is reverse-polarity protected, but incorrectly connecting any module in any system can damage other modules on the power bus.*

Overview

The **Mini PEG** is a “pingable” envelope generator with the envelope time set by the duration between pulses (“pings”). The tempo can be set by tapping the **Ping** button, or by patching an external clock into the **Ping** jack. The ping clock is then multiplied or divided by the amount set by the **Div/Mult** knob and CV, from /32 to x16. The envelope will be as long as the period of the divided/multiplied clock. Various waveshapes can be selected using the **Shape** knob and CV. The envelope can also be scaled, inverted and offset using the **Scale** and **Offset** knobs.
Sending a trigger into the **Trigger** jack will produce an envelope. The **Mini PEG** can also function as an LFO by producing envelopes constantly when the **Cycle** button is engaged, or when a gate is held high on the **Cycle** jack.

Beginning Patch

Step 1: Prepare the Mini PEG

Unplug all cables, turn **Div/Mult**, and **Shape** to center, turn **Scale** and **Offset** all the way up. Make sure **Cycle** is off (not lit up).
Plug the **ENV OUT** jack into something you want to modulate — perhaps a filter, or the pitch of an oscillator.

Step 2: Set the ping clock

*Before the **Mini PEG** can generate an envelope, you need to supply a ping clock. The ping clock is the basic reference that determines the timing of the envelope.*

Tap the **Ping** button two or three times, about a second between taps. It's easier to see what's going on with a slow envelope. The **Ping** button should be flashing white at the tempo you tapped. If you give a third tap, the time between taps will be averaged.
You could also run an external clock into the **Ping** jack (when a trigger is received on the **Ping** jack, the internal tap tempo clock is stopped).

Step 3: Provide a trigger source

*Like any envelope module, the **Mini PEG** will produce an envelope when it receives a trigger. Also like many envelope modules, it can be set to self-trigger so that it'll cycle without any external trigger.*

Press the **Cycle** button (it will light up orange). Notice the red/blue light above the **ENV OUT** jack starts flashing. The envelope is now running freely, in time with the ping clock. Adjust the other module(s) you are running the **Mini PEG** into so you can hear the modulation.

You could also turn **Cycle** off and run a manual trigger into the **Trigger** jack (e.g. try the gate output from a sequencer, or a slow LFO, or perhaps a slow clock output from a QCD, RCD, or other clock module.)

Step 4: Adjust the output level

Adjust the **Scale** and **Offset** knobs to get a voltage range that works well with whatever you're modulating.

Step 5: Set the Division/Multiplication amount

Turn the **Div/Mult** knob to various settings and watch the **Ping** button flash faster or slower. Notice how the tempo doesn't change gradually, but instead jumps from speed to speed. This is because each speed is an integer multiple or division of the original tempo (e.g. three times as fast, or half as slow). You also can modulate this parameter with the **Div CV** jack at the bottom.

Step 6: Adjust the shape

Fiddle with the **Shape** knob to get an envelope shape you like. Notice the shapes in the center section of the knob are all symmetrical (same rise / fall shape and time), while the shapes at the start and end of the knob's range are asymmetrical (different curves and times for rise / fall). Of course, you can modulate this parameter with the **Shape CV** jack at the bottom.

That's it, a basic **Mini PEG** patch!

Control Details

Ping
The **Ping** button and jack set the tempo of the ping clock. This clock is later divided or multiplied to set the period of the envelope.
The button flashes to show the tempo. In *Sync* mode it flashes cyan (light blue), and in *Async* mode it flashes white (See Sync and Async Modes).

Tapping a tempo
Two taps on the **Ping** button set the ping clock period. If you give a third tap, it will be averaged with the first two taps, unless the timing period set by the third tap is widely different than the first two taps. In this case the second and third taps will be used.
If you want visual feedback as you tap, set the **Div/Mult** knob to "=" so that the **Ping** button flashes at the same tempo that you tap.

Syncing to an external clock
To synchronize to an external clock or gate signal, patch it into the **Ping** jack. The timing between the last two pulses sets the ping clock.
Unlike the **Ping** button, the **Ping** jack does not average pulses, it simply takes the period of the last two pulses. You can plug in a manual trigger/gate module and just tap in two pulses: the ping clock will continue to run at that tempo even if you don't provide any more trigger pulses. *Note: in System Mode, you can disable this free-running feature (thus, giving the **Mini PEG** two external pulses will output just one envelope and then stop) See the System Mode section.*

When a gate is received on the **Ping** jack, the tap tempo clock is disabled. Conversely, when the **Ping** button is tapped, the external clock will be overridden. Try tapping a fast tempo while you have a slow clock going into the **Ping** jack — the fast tempo will take effect immediately, but will revert back to the slow tempo when the next clock pulse is received on the jack.

Clearing the tempo
Hold the **Ping** button down for 2 seconds to clear the tap tempo. The light will go off and the envelope will stop. You can also clear an externally generated ping if the external unit has stopped sending pulses or is unpatched.

Div/Mult
Once a ping clock has been established, you can use the **Div/Mult** knob and jack to divide or multiply the tempo from 1/32nd of the speed to 16 times the speed, in whole number increments. The resulting clock defines the length of the envelope output. The **Ping** button flashes at the rate of this divided/multiplied ping clock.
The **Div/Mult** knob has detents for each of the 19 divide/multiply amounts, and any CV that's applied to the **Div CV** jack will be added to the knob's setting.

Changing the divide/multiply amount while an envelope is running will immediately change the slope of the envelope. Depending on whether the module is in *Sync* or *Async* mode, the envelope may re-sync to the ping clock. See Sync and Async Modes section below.

The **Div CV** jack ranges from -5V to +5V and offsets the knob's position.

Cycle

The **Cycle** function makes the envelope self-cycle (LFO mode). When this button is lit, the envelope will keep running without needing any external triggers. Pressing the **Cycle** button will toggle the cycle state on/off.

The **Cycle** jack toggles the cycling state. By default, when a gate is applied to this jack, the **Cycle** button will toggle state (on→off and off→on). When the gate is released, cycling will revert to its previous state. In System Mode, this jack can be changed to toggle the cycling state each time it receives a trigger. So if **Cycle** is on, a trigger will turn it off, and a subsequent trigger will turn it on again.
Depending on whether the module is in *Sync* or *Async* mode, the envelope will start playing from a different point (starting phase) when the **Cycle** button is pushed. See Sync and Async Modes section.

Trigger

The **Trigger** jack initiates an envelope when a trigger is received. By default, the envelope starts immediately and the module is put into *Async* mode. Holding the trigger high will have no effect. In System Mode, this jack can be configured as a Quantized jack, so that receiving a trigger will cause an envelope to output on the next ping clock. When used this way, the module will be put into *Sync* mode when it receives a trigger. Holding the **Trigger** jack high will cause the envelope to cycle until released.
The jack can also be configured as an Async Gate jack. This is the same as the default behavior (Async Trigger), but the envelope will sustain if the gate is held high past the end of the rise phase. When the gate is released, the fall stage will start.
Async modes allow you to re-trigger an envelope that's already running, while the Quantized mode does not. See Sync and Async Modes section and System Mode sections.

Shape

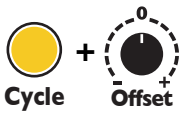
The **Shape** knob and jack control both the skew (or ratio of rise to fall times) and the selection of waveshapes (logarithmic, linear, exponential). While there is a continuous spectrum of waveshapes, the range can be divided into three zones.
The first zone is from fully counter-clockwise to around 9:30. Turning the knob clockwise, the envelope starts as a ramp-up exponential shape and slowly turns to a linear ramp-up shape.
The transition from the first zone to the second zone goes from a ramp-up linear shape to a balanced exponential shape (equal rise and fall times) at about 10:00.
The second zone begins with the balanced expo shape. It then transitions to a balanced linear shape at the knob's center position, and then to a balanced log shape at about 2:00.
The transition to the third zone turns the balanced log shape into a ramp-down linear shape at about 2:30
The third zone mirrors the first zone, going from ramp-down linear to ramp-down exponential.
The CV jack ranges from -5V to +5V and offsets the knob's position.

Scale

The **Scale** knob is an attenuverter for the main envelope output, **ENV OUT**. If the scale knob is right of center, the output voltage will increase when the envelope is in the rise phase, and decrease when the envelope is falling (non-inverted). This is normally how envelopes are used. If the **Scale** knob is left of center, the envelope will be inverted and the output voltage will decrease as the envelope rises, and increase as it falls.
As you turn **Scale** towards center, the amplitude will decrease, moving the peaks towards zero. At center, there will be no output (technically, it produces a steady DC voltage which is set by the **Shift** feature).

Offset

The **Offset** knob shifts the envelope on the **ENV OUT** jack up and down by adding DC offset. The offset is applied *before* **Scale** is applied. For a positive-only envelope, **Offset** would typically be set at maximum, and **Scale** should be right of center. For a bi-polar envelope, **Offset** should be set to center. For a negative-only envelope (non-inverted), **Offset** should be set to minimum.



Shift
By holding down the **Cycle** button while turning the **Offset** knob, the **Shift** amount can be controlled. This is similar to **Offset**, but is applied *after* the envelope is offset and scaled. The DC level of the **Shift** amount is the center-line across which the **Scale** knob attenuates and inverts the envelope. **Shift** essentially shifts the scaled/offset envelope up and down, causing clipping at the -9.4V and +9.4V rails.
By default, **Shift** is 0V, which is the most intuitive setting. Any envelope voltages (amplitude and offset) that the **Mini PEG** is capable of generating can be produced without adjusting **Shift**. However, with a proper understanding of DC offset, amplitude, inversion, and clipping, changing **Shift** can help in special-case situations by allowing the **Scale** knob to invert the envelope across a voltage other than 0V.

ENV OUT
ENV OUT is the main envelope output. Its range is from -9.4V to +9.4V. The red and blue lights above the **ENV OUT** jack are helpful in determining if the envelope is positive (blue), negative (red), or both.

5V ENV
The **5V ENV** jack is an envelope output that always produces a 0V to +5V envelope regardless the **Offset**, **Scale**, and **Shift** settings.

EOF
The End-of-Fall (**EOF**) jack outputs a gate that goes high when the fall segment ends. If the envelope is stopped, the jack will remain high. It will only go low when a rise segment ends and a sustain or fall segment begins. Another name for this jack might be “Envelope is Rising or Stopped”.
In System Mode, this jack can be configured as an End-of-Rise jack, a Half-Rise jack, or a Tap Clock output jack. End-of-Rise (EOR) goes high when a fall segment starts (and stays high if it stops), and goes low when a rise segment starts (and stays low if it sustains). This jack could be called “Envelope is Falling or Stopped”.
Half-Rise goes high when the rise portion is half-way finished, stays high through any sustain period, and goes low when the fall portion is half-way finished.
Tap Clock runs freely even if the envelope is not running. It outputs the raw ping clock (before division/multiplication) that was tapped on the **Ping** button, ignoring any external clocks. See the Tap Clock Output section for details.

The **EOF** jack outputs a square wave by default, but in System Mode it can be changed to output triggers instead.

Maximum and Minimum Clock Times

The **Mini PEG** can process incoming clocks and output envelopes up to about 4kHz, with some jitter, glitches, and imprecision. When outputting frequencies higher than 500Hz, the output will contain increasing amounts of noise and glitches, but still holds a solid fundamental frequency and waveshape. Since it goes well into the audio range, the **Mini PEG** can be used as a rudimentary lo-fi harmonizer (e.g. sub-octave generator). Be aware that when using the **Mini PEG** as an audio generator, small amounts of jitter on the incoming clock will cause large amounts of audible distortion and glitches on the output.

The maximum envelope time is about 29 hours (1750 minutes). The maximum ping clock time is also about 29 hours, so if you fire a clock pulse into the **Ping** jack once a day, the **Div/Mult** setting must be = or multiplying in order to produce an accurate output.

Sync and Async Modes

The **Mini PEG** can operate in *Sync* or *Async* mode. Most of the time, the difference between the two modes is not important and the mode will automatically change in an intuitive way.
In *Sync* mode, the envelope will always begin and end on the divided/multiplied ping clock, but in *Async* mode the envelope can begin at any point in time. Thus, *Sync* mode locks the phase of the envelope and the divided/multiplied ping clock together, while *Async* mode allows for any phase difference.

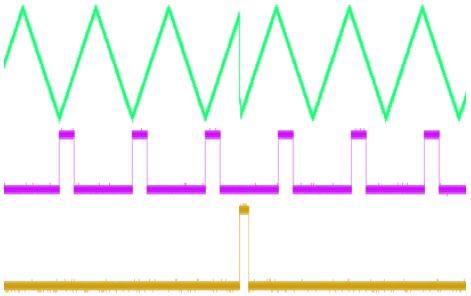


Figure 1: Envelope (green) is synced to Ping clock (magenta) until an Async Trigger (gold) is received

Changing Between Sync and Async Mode

- To enter *Sync* mode, fire a trigger into the **Trigger** jack while it's configured as a Quantized Trigger jack. Or, you can fire a trigger into the **Cycle** jack while it's configured to enter Sync mode. See System Mode for instructions on how to configure the jacks.
- To enter *Async* mode, fire a trigger into the **Trigger** jack when it's configured as an Async Trigger or Async Gate jack.
- You can toggle modes by holding down **Cycle**, tapping **Ping**, then releasing **Cycle**.

In *Sync* mode, the **Ping** button flashes cyan. In *Async* mode, the **Ping** button flashes white.

Figure 1 illustrates envelopes in each mode. The **Mini PEG** begins cycling in *Sync* mode: each envelope starts on a ping clock pulse. When a trigger is fired into the **Trigger** jack (which is set to Async Trigger or Gate mode), the envelope immediately resets and continues cycling out-of-phase with the ping clock. The module has now changed to *Async* mode automatically. In this case, it's not important that you know what mode the module is in, it's just doing what you told it to do.

Cycle Button in Sync and Async Mode

When using the **Cycle** button, it can be important to know whether the **Mini PEG** is in *Sync* or *Async* mode.
When you turn on the **Cycle** button in *Sync* mode, the envelope will immediately start outputting from the same phase as the ping clock. You can imagine this as if the envelope was always running (in sync with the ping clock), and the **Cycle** button un-mutes it. See Figure 2. This can cause some discontinuities, but the envelope will always be in sync with the ping clock. When you turn **Cycle** off, the output will stop after the current envelope finishes.
If you turn on the **Cycle** button in *Async* mode, the envelope will immediately start outputting from zero. One or more complete envelopes will always be output, but they will be synced to your button press, not to the ping clock. See Figure 3.

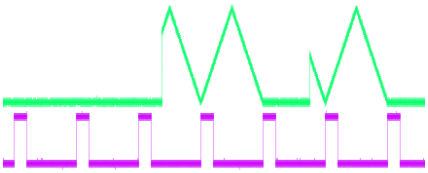


Figure 2: Pressing Cycle button in Sync mode.

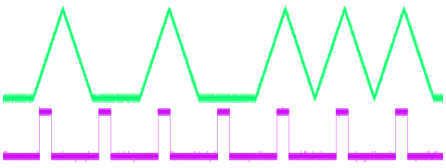


Figure 3: Pressing Cycle button in Async mode.

Modulating Div/Mult in Sync and Async Mode

If you change the **Div/Mult** amount while an envelope is running, the module's response will depend on whether it's in *Sync* or *Async* Mode. In both modes, the envelope slope will track the **Div/Mult** amount as long as the amount keeps changing.
In *Sync* mode, if no change to the **Div/Mult** is made for 50ms, then the envelope will re-sync to the ping clock based on the new **Div/Mult** amount. See Figure 4. The **Div/Mult** knob was turned from /2 to = at the black arrow. Between the black and red arrows, the envelope makes a quick transition to a new envelope which is in sync with the ping clock. Notice how every envelope starts and ends exactly on a ping clock, even though we changed envelope lengths.
In *Async* mode there is no requirement that the envelopes sync to the ping clock, so it will transition smoothly between waveforms without any re-syncing. See Figure 5. The envelope tracks the changes in the **Div/Mult** amount but never needs to

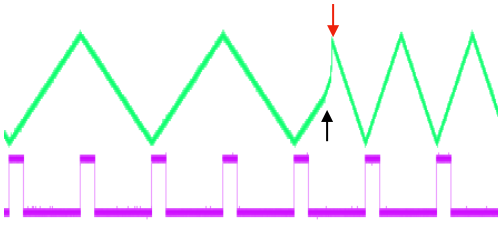


Figure 4: Changing Div/Mult in Sync Mode

re-sync to the ping clock. Notice how the envelopes do not necessarily start or end on a ping clock.

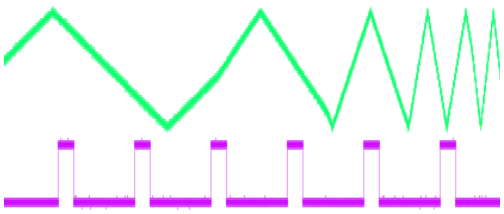


Figure 5: Changing Div/Mult in Async Mode

Using the Trigger Jack in Sync and Async Mode

When the **Trigger** jack is configured as Quantize and it receives a trigger, the module will wait for the next divided/multiplied ping clock before staring the envelope. See Figure 6.

When the **Trigger** jack is set to Async Trigger or Async Gate, an envelope will begin immediately upon receiving a trigger, regardless of whether the ping clock is firing at the same moment. See Figure 7.

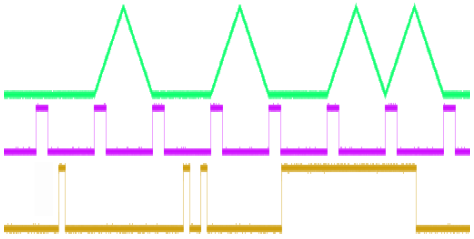


Figure 6: Quantized Trigger jack (gold)

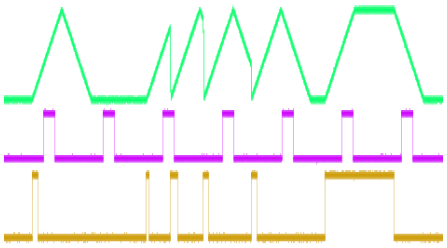


Figure 7: Async Gate jack (gold).

Quantize Trigger Re-Phasing

In *Sync* mode, when **Div/Mult** is set to dividing and the **Cycle** button is on, firing a trigger into the **Trigger** jack configured as Quantize will re-start (re-phase) the envelope on the next incoming ping clock. For example, if **Div/Mult** is set to /4 then the ping clock represents quarter notes, and we could count measures as 1-2-3-4-1-2-3-4 etc. The envelope starts on the “1”, reaches its peak on the “3” and finishes just as the next “1” occurs. If we were to fire a trigger into the Quantize **Trigger** jack somewhere in between the “2” and “3”, then the envelope would re-phase so that it starts on the “3”. See Figure 8.

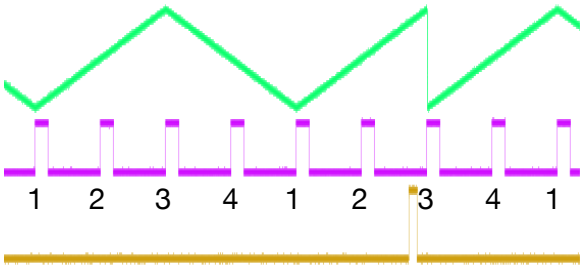


Figure 8: Re-phasing a /4 envelope.

This can be useful for multi-phase outputs (e.g. Quadrature patch).

Tap Clock Output

The **EOF** jack can be configured to output a tap tempo clock. This is done in System Mode (see System Mode section). When configured as Tap Clock, whatever tempo you tap on the **Ping** jack will be steadily output on the **EOF** jack. Unlike configuring the **EOF** jack as End-of-Rise, End-of-Fall, or Half-Rise, the **Div/Mult** parameter will not affect the tempo being output on this jack. More importantly, the tempo you tapped will continue to be output even if you are clocking the **Mini PEG** with an external clock at a different tempo.

This feature allows you to use the **Ping** button and **EOF** jack as a semi-independent tap tempo sub-module: tapping the **Ping** button will only change the tempo on the **EOF** jack and not the tempo of the envelopes.

Bus Clock

Clock Bus is a 1:1 clock that runs along the gate pin of the Eurorack power bus. This feature allows for a compatible module to send a clock signal over the power bus, and for one or more modules to synchronize to this clock. The **MiniPEG**, **DLD**, and **QCD** from 4ms are all capable of sending and

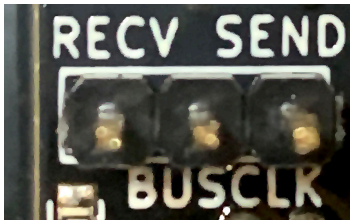
receiving a clock over the Clock Bus. The **RCD** and **SCM Plus** can receive only. Modules from other manufacturers that send Clock Bus are also likely to be compatible, such as the Malekko Varigate 8+.

On the back of the **MiniPEG** is a jumper to enable sending or receiving clocks over the Clock Bus.

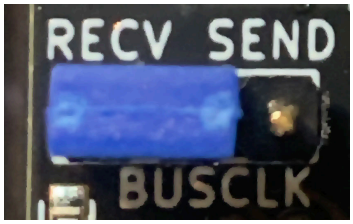
When this jumper is installed in the RECV position, clock pulses on the Clock Bus will be sent to the **Ping** jack as if they were patched directly into the jack. When a cable is plugged into the **Ping** jack, the Clock Bus signal will be ignored.

When this jumper is installed in the SEND position, the ping clock (before division or multiplication) will be sent over the Clock Bus. In other words, if you tap a tempo on the **Ping** button or if you patch an external clock into the **Ping** jack, this tempo will be sent 1:1 to the Clock Bus. Changing the **Div/Mult** amount will not change the signal on the Clock Bus. When using the **Mini PEG** to send a clock, you must ensure there are no other modules configured to send a clock over the same Clock Bus. Having two modules send clocks on the same Clock Bus will result in unpredictable behavior, possibly causing erratic tempos and/or making all Clock Bus modules fail to work properly.

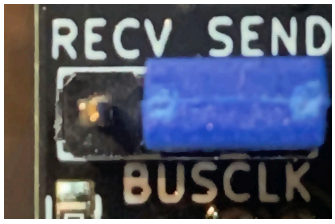
To disable use of the Clock Bus, remove the jumper entirely. It's safe to "park" the jumper by installing it on only one pin.



Bus Clock Disabled



Receive Bus Clock



Send Bus Clock

Updating Firmware

The **Mini PEG** has a boot loader that allows for updating the firmware by playing a special audio file into the module. When a firmware update is available, it will be posted on 4mscompany.com as a WAV file (or zipped WAV file).

Updating Firmware

1. Download the audio firmware file and prepare to play it on your computer or mobile device.
2. Set your computer or device volume to maximum, both in the system/OS controls and also in the audio player application.
3. Power off the **Mini PEG** and unpatch all cables.
4. Patch the computer/device output to the **Shape CV** jack (any mono or stereo cable will work).
   - Note: If you have a 4ms **Listen IO** or other module that boosts line level to Eurorack level, run the audio through it before patching into the **Shape CV** jack.
5. Hold down **Ping** and **Cycle** while you power on the **Mini PEG**.
6. The module will boot into bootloader mode, with the **Ping** light blinking green, and the **ENV OUT** and **5V ENV** lights purple.
   - If you wish to exit bootloader mode, press the **Ping** button while it's flashing green.
7. Begin playing the audio firmware file. You can monitor the signal by listening to the **EOF** jack.
   - The **Ping** and **Cycle** buttons will alternate blue and white to indicate the signal is being received
8. If the update fails, the buttons will flash red. Press **Ping** to try again.
   - If it fails more than once, you may need to adjust the threshold. See below.
9. When the update succeeds, the lights will flash rainbow colors. Press **Ping** to start running the new firmware.

Adjusting the Threshold

The **Mini PEG** can compensate for a variety of signal levels by adjusting the threshold. This technique is only recommended if you don't have access to a line-to-Eurorack level conversion module such as the **Listen IO**.

The **ENV OUT** and **5V ENV** lights show the current threshold. Each light can be one of five possible colors: off, blue, purple, red, white. Off is the lowest value and white is the highest value. There are 25 possible values. The **ENV OUT** light is weighted more than the **5V ENV** light.
The lowest threshold is represented by both lights being off. The next higher threshold value is when **ENV OUT** is off and **5V ENV** is blue. The highest value is when both lights are white.

The module starts bootloader mode with the threshold near the middle of the range: both lights purple. Each time you press **Cycle,** the threshold will increase and the lights will change. Once you get to both lights being white (highest threshold), it will cycle back around to both lights being off (lowest threshold). If you have trouble loading a firmware file and have already tried a different device and checked your volume, then adjusting the threshold may help. The best approach is to try different values until you find one that works.

Troubleshooting

- Make sure the volume on your device and app are at 100%.
- Try a different computer or device.
- Do not use sophisticated playback software such as Ableton, as these can inadvertently alter the encoded audio. Use the most basic playback program available.
- Adjust the threshold (see above)
- If you have an oscilloscope or a way to view waveforms, compare the output of the **EOF** jack with the output of the computer or device. Adjust the threshold using the **Cycle** button until the pulses are similar (accounting for about 40µs-80µs of latency, and of course amplitude differences).
- Use a line-to-Eurorack module, such as the **Listen IO**, between the computer and the **Mini PEG**.

Viewing the Firmware Version

On power-up, the **Mini PEG** displays its firmware version by flashing two colors on the **Ping** button. The first color turns on immediately after power-up, then the light turns off for a moment and the second color turns on. As of this manual printing, the latest (and only) version number is 1.0.

First Color	Second Color	Version
Red	White	1.0

System Mode

System Mode allows you to change the way the **Mini PEG** works. To enter System Mode, hold down the **Cycle** button for 3 seconds. When the lights flash to confirm entry into System Mode, release the button.

You can exit System Mode at any time by holding down either the **Ping** button or the **Cycle** button for 3 seconds. If you exit with the **Cycle** button, your changes will be active but will not be saved after you power down (the previous saved values will be restored). If you exit with the **Ping** button, your changes will be saved.

If you enter System Mode with **Cycle** on, the envelope will continue to cycle while you remain in System Mode, but you will be unable to adjust its parameters.

Saving the Clock Time and Cycle State

The ping clock period and the state of the **Cycle** button can be saved so that the next time you power-on, the **Mini PEG** will load the saved values.

If you are using an external clock patched into the **Ping** jack or received over Clock Bus, the tempo of that clock will be saved. If you tapped a tempo on the **Ping** button, that tempo will be saved. In either case the tempo before **Div/Mult** is applied will be saved.

Whether or not the module is cycling will also be saved. If the **Cycle** button is illuminated at the moment you enter System Mode, then the module will turn **Cycle** on when you next power-on.

There are six parameters in System Mode. Tapping the **Cycle** button will advance the parameter, looping around to the first after the sixth. Tapping the **Ping** button will change the parameter value.

1. Trigger Jack Function:
The **EOF** light will be on, and the **ENV OUT** light will be red. The color of the **Ping** button shows the function of the **Trigger** jack. Press the **Ping** button to toggle.

Orange: Async trigger*	*Green*: Async gate (sustain)	*Blue*: Quantized trigger

2. Cycle Jack Function:
The **Cycle** button will be flashing and all other lights will be off. The color of the **Cycle** button shows the function of the **Cycle** jack. When it's set to Gate toggle, cycling will be toggled by a gate on the jack. When it's set to Trigger toggle, each trigger will toggle cycling. Entering *Sync* mode upon receiving a gate is also an option (See Sync and Async Modes section). Press the **Ping** button to toggle.

Orange: Gate toggles*	*Blue*: Gate toggles and enters *Sync* mode	*Red*: Trigger toggles

3. EOF Jack Function:
The **EOF** light will be on, the **Ping** button will be lit, and all other lights will be off. The color of the **Ping** button shows the function of the **EOF** jack. Press the **Ping** button to change functions.

Orange: End-of-Fall*	*Red*: End-of-Rise	*Green*: Half-Rise	*White*: Tap Clock

4. EOF Jack Trig/Gate:
The **EOF** light will flash/flicker. All other lights will be off except for **Ping** button. The color of the **Ping** button and the pulse width of the **EOF** light shows whether the **EOF** jack outputs gates or triggers. When set to gates, the **EOF** light will flash equally on and off. When set to triggers, the **EOF** light will flicker briefly on. Press the **Ping** button to toggle.

Ping Orange, and EOF flashes: Gate Output*	*Ping White, and EOF flickers*: Trigger Output

5. Free-running Ping:
The **Ping** button light will be flashing white or red. All other lights will be off. The **Ping** button shows whether the ping clock keeps running after an external clock patched into the **Ping** jack stops. Press the **Ping** button to toggle.

Ping flashes white: Free-running ping*	*Ping solid red*: No free-running ping

6. Slew Limit:
The **ENV OUT** light will be blue. All other lights will be off except for **Cycle** button. The color of the **Cycle** button shows whether the maximum slew time is limited. When slew is not limited, popping or clicking can be heard when using the **Mini PEG** with a VCA. Slew limiting prevents this by limiting the fastest rise or fall time. Slew is never limited if the envelope period is in the audio range. Press the **Ping** button to toggle.

Orange: Slew not limited*	*White*: Slew limited

*default

Electrical and Mechanical Specifications

- **Mini PEG**
  - 8HP Eurorack format module
  - 0.95” (24mm) maximum depth (includes power cable)
  - 16-pin Eurorack power header
- **Power consumption**
  - +12V: 55mA
  - -12V: 30mA
- **Envelope Outputs**
  - Maximum voltage range (**ENV OUT**): -9.4V to +9.4V
  - Maximum amplitude (**ENV OUT**): 9.5Vpp
  - Voltage range (**5V OUT**): 0V to +5V
  - Frequency range: +/-0.1dB DC to 1kHz, -1.0dB@2.5kHz, -2.0dB@4kHz
  - 12-bit, 40kHz sample rate
- **Clock Range**
  - Period range: 29.16 hours to 250µs (4kHz)
- **CV Inputs (Div/Mult and Shape)**
  - Voltage range: -5V to +5V
- **Gate Inputs (Ping, Trigger, Cycle)**
  - Rising edge threshold: 2.5V