

2hp – Arp

- [Manual PDF](#)
-

[Manual PDF](#)

2hp Arp – melodic use in a Eurorack system

The **2hp Arp** is a **gate-driven arpeggiator** that turns a selected chord into a stepped **V/Oct melodic sequence**. It's very compact, but musically it can do a lot: chord-based riffs, repeating ostinatos, pseudo-basslines, harmonic motion, and generative melody when paired with clocks, gates, modulation, and quantized pitch destinations.

What the module does

At its core, **Arp outputs pitch CV** based on:

- a **selected chord**
- a **root note**
- an **arpeggio mode**
- incoming **trigger pulses**
- optional **reset pulses**
- optional CV over **root** and **chord**

Main controls and I/O

- **Trig Input**
- Advances the arpeggio one step per gate/trigger
- Threshold: **0.4V**

- **Reset Input**
 - Restarts the pattern at the **root note**
 - Threshold: **0.4V**
- **Root Knob**
 - Sets the base pitch / chord root
- **Root CV Input**
 - CV control over root
 - Range: **0V to +5V**
- **Chord Knob**
 - Selects chord type
- **Chord CV Input**
 - CV control over chord selection
 - Range: **0V to +5V**
- **Mode Knob**
 - Selects playback direction/order
- **Output**
 - **V/Oct pitch CV output**
 - Range: **0V to +5V**

Available chords

The module can arpeggiate these chord qualities:

- Major
- Major 7
- Dominant 7
- Minor

- Minor 7
- Diminished
- Half Diminished 7
- Full Diminished 7
- Augmented
- Augmented 7
- Sus 4
- Sus 4 Maj 7
- Sus 4 Min 7

This makes it much more than a simple “up/down notes” utility – it’s really a **harmony-to-melody converter**.

Available arpeggio modes

Playback modes include:

- Ascending one octave
- Ascending two octaves
- Descending one octave
- Descending two octaves
- Pendulum one octave
- Pendulum two octaves
- Random one octave
- Random two octaves

These modes strongly affect the melodic character:

- **Ascending/descending** = orderly, classic arpeggios
 - **Pendulum** = more lyrical, looping shapes
 - **Random** = generative, less repetitive
 - **Two-octave** = more dramatic range and movement
-

How to use 2hp Arp for melody

1. Basic chord-to-melody patch

This is the most direct use.

Patch

- Send a **clock or trigger sequence** to **Trig Input**
- Patch **Arp Output** to your oscillator's **V/Oct**
- Send the oscillator to a **VCA/filter/voice path**
- Trigger an envelope from the same clock or from a related gate source
- Choose:
 - **Root**
 - **Chord**
 - **Mode**

Result

Each trigger advances to the next note in the chosen chord, creating a melodic line that stays harmonically coherent.

Musical use

- bass arps
- plucks
- techno sequences
- synthwave ostinatos
- Berlin-school style step motion

2. Use reset for phrase structure

The **Reset input** is one of the most important features musically.

Patch idea

- Fast clock to **Trig**
- Slower pulse every 4, 8, or 16 steps to **Reset**

Result

The arp restarts on the root at regular intervals, producing a repeating phrase instead of endlessly drifting.

Why it matters

Without reset, the pattern may feel more circular or continuous.

With reset, it becomes more **song-like**: - downbeats land on root notes - phrases become easier to hear - basslines feel more intentional

This is especially strong in: - techno - trance - arpeggiated pop lines - sequenced ambient

3. Animate harmony with Root CV

The **Root CV input** lets you transpose the entire arpeggio.

Patch idea

- Send a slow sequencer, precision adder source, keyboard CV, or quantized CV to **Root CV**
- Keep a steady trigger stream into **Trig**

Result

The arpeggio pattern stays structurally similar, but shifts to new tonal centers.

Musical applications

- chord progressions

- key changes
- moving bass foundations
- verse/chorus transitions

Example progression idea

Send stepped voltages to Root CV corresponding to: - I - vi - IV - V

Then set the chord quality manually or by CV.

This gives you a complete **harmonic progression generator** with only a few signals.

4. Modulate chord type for harmonic movement

The **Chord CV Input** is where things get especially interesting.

Patch idea

- Send a stepped CV source to **Chord CV**
- Clock that CV more slowly than the arpeggio triggers

Result

The chord quality changes while the note stepping continues.

Musical applications

- alternating major/minor colors
- tension/release with 7ths
- unstable cinematic diminished motion
- suspended chords for modal textures

Tip

For musical results, clock chord changes at slower divisions: - note changes every 16th - chord changes every bar or half-bar

This creates a strong hierarchy: - fast motion = melody - slow motion = harmony

5. Build full progressions with both Root CV and Chord CV

This is where Arp becomes a compact composition tool.

Patch concept

- **Trig:** steady clock
- **Reset:** phrase reset every 8 or 16 steps
- **Root CV:** progression source
- **Chord CV:** chord-quality source
- **Output:** oscillator V/Oct

Result

You get a melody that reflects: - harmonic root movement - chord quality changes - pattern direction - octave span

This is enough to generate: - evolving arpeggiated hooks - soundtrack patterns - melodic techno motifs - generative tonal lines

Best pairings with other Eurorack modules

The manual only covers the Arp itself, but musically, here's how it works with common module types.

With a clock source

A clock or trigger sequencer is essential because Arp is **gate-driven**.

Use: - master clocks - trigger sequencers - Euclidean triggers - clock dividers/multipliers - manual gate buttons

Why

The timing source determines: - note density - groove - syncopation - phrasing

A rigid clock gives machine-like precision.

Irregular trigger sources give broken, human, or generative melodies.

With an oscillator or full voice

Arp outputs **pitch CV only**, so it needs a sound source.

Good destinations: - analog VCO - wavetable oscillator - FM voice - pluck voice - physical modeling voice - any full synth voice with V/Oct input

Why

Different voices change the role of the arp: - sine/triangle = bass or minimal melody - saw/pulse = classic synth arps - wavetable = modern animated sequences - FM = metallic/cinematic patterns

With envelopes and VCAs

To hear discrete notes, patch triggers to an envelope and VCA.

Typical patch

- Clock mult:
- one copy to **Arp Trig**

- one copy to **Envelope Gate/Trig**
- Envelope to VCA CV
- Oscillator through VCA
- Arp output to oscillator pitch

Result

Each pitch step becomes an articulated note.

Variation

Use a different rhythmic trigger for the envelope than for the arp advance: - arp moves on every 16th - envelope opens only on selected steps

This creates **implied melodies** and rhythmic variation.

With filters

If you patch the voice through a filter, you can make the melodic line feel more expressive.

Patch ideas

- envelope to filter cutoff
- LFO to cutoff
- accent gate to cutoff CV
- reset pulse to a burst envelope affecting brightness

Result

The harmony may stay constant while timbre adds motion, making the arp more musical and less static.

With sequencers

Sequencers pair beautifully with Arp, especially if you use them not for direct pitch, but for **meta-control**.

Use a sequencer to control: - **Root CV** - **Chord CV** - **Mode changes** via manual performance between takes - resets and phrase length

Why this is powerful

Instead of sequencing every note directly, you sequence the **rules** that generate the notes.

That gives: - coherent tonal output - less programming - more happy accidents

With random / sample-and-hold / chaos sources

These are great for generative work.

Good uses

- random stepped CV to **Root CV**
- restrained random CV to **Chord CV**
- irregular gates to **Trig**
- slow random pulse to **Reset**

Result

A self-moving melodic network that still sounds harmonically structured because Arp constrains note choices to chord tones.

This is one of the best uses of the module:
randomness with harmonic containment.

With quantizers

You may not always need a quantizer after Arp, since it already outputs pitch in a structured harmonic system. But quantizers can still help upstream or downstream.

Useful placements

- Quantize CV going into **Root CV**
- Quantize external transposition sources
- Use a precision adder + quantizer combo for key control

Why

This helps keep chord roots aligned to a chosen scale or song key.

With precision adders / offsets

A precision adder is very useful if you want to transpose the arp while preserving interval relationships.

Use cases

- add keyboard CV to Arp output
 - add bass transposition
 - layer multiple pitch influences
 - create call-and-response by offsetting copies
-

Melodic patch recipes

Patch 1: Classic synth arpeggio

- Clock → Trig

- Reset every 8 steps → Reset
- Arp Out → VCO V/Oct
- Clock → Envelope trig
- VCO → VCF → VCA
- Set chord to **Minor 7**
- Mode = **Ascending one octave**

Sound: tight, classic, musical arp line

Patch 2: Bass ostinato with harmonic change

- 16th-note trigger → Trig
- Bar reset → Reset
- Slow sequencer → Root CV
- Chord fixed to **Minor** or **Dominant 7**
- Arp Out → bass oscillator

Sound: repeating bass figure that follows a progression

Patch 3: Generative ambient melody

- Irregular trigger source → Trig
- Slow random pulse → Reset
- Slow random stepped CV → Root CV
- Very slow stepped modulation → Chord CV
- Mode = **Random two octaves**
- Arp Out → mellow voice or resonator

Sound: evolving melodic fragments with harmonic identity

Patch 4: Suspended tension pattern

- Clock → Trig
- Reset every 16 steps
- Chord set to **Sus 4**, **Sus 4 Maj 7**, or **Sus 4 Min 7**

- Mode = **Pendulum one octave**
- Bright plucked voice

Sound: unresolved, floating melodic texture great for intros and breakdowns

Patch 5: Dark cinematic motion

- Clocked triggers → Trig
- Root sequence moves slowly
- Chord CV alternates between:
 - Diminished
 - Half Diminished 7
 - Full Diminished 7
- Mode = **Descending two octaves**

Sound: ominous, unstable, tension-heavy melodic movement

Performance strategies

Use reset as a musical downbeat

If you manually or rhythmically fire reset at section starts, the pattern feels intentional and anchored.

Perform chord changes live

The **Chord knob** is performance-friendly. Sweeping between major, minor, suspended, augmented, and 7th-based chords can dramatically change mood without repatching.

Switch modes for arrangement changes

Even with the same root and chord: - **Ascending** = verse energy - **Pendulum** = flowing bridge - **Random** = breakdown or ambient section - **Two-octave** = lift or climax

Use slower modulation for structure

Fast note triggers with slow root/chord modulation give a musically readable hierarchy.

Layer multiple voices

Mult the Arp output to: - a bass voice - a higher pluck voice - a delayed or reverb-heavy texture

If transposed or timbrally separated, one arp stream can become a complete melodic stack.

Practical notes from the manual

Voltage ranges

- **Root CV:** 0V to +5V
- **Chord CV:** 0V to +5V
- **Output:** 0V to +5V

That means it plays in a fairly standard Eurorack pitch space and should work well with most V/Oct inputs.

Trigger/reset thresholds

- **Trig threshold:** 0.4V
- **Reset threshold:** 0.4V

So standard Eurorack triggers and gates should work fine.

Size and power

- **Width:** 2HP
- **Depth:** 45mm
- **Power:** +12V 40mA, -12V 3mA, +5V 0mA

Very easy to fit into a small system, especially if you want a compact melodic utility.

Bottom line

The **2hp Arp** is best thought of as a **melodic engine** rather than just an effect or helper module. It works especially well when combined with:

- a **clock/trigger source**
- a **voice or oscillator**
- **envelopes + VCA**
- optionally a **sequencer** for root/chord movement
- optionally **random/modulation** sources for generative behavior

Its strength is that it creates melodies that are: - rhythmically driven - harmonically constrained - easy to transpose - compact to patch - performable in real time

So if you want **musical melodic content quickly**, this module is excellent for: - arpeggios - ostinatos - basslines - harmonic motifs - generative tonal melodies

Generated With [Eurorack Processor](#)