Buchla and Tiptop Audio — 248t

- Manual PDF

Manual PDF

Tiptop/Buchla 248t MARF for dense, hyper-complex percussion

The **248t MARF** is extremely powerful for percussion because it is not just a sequencer. It is a **dual arbitrary function generator**, meaning each side can independently move through 16 programmed stages, with per-stage:

- voltage
- time
- pulse assignment
- quantize / slew behavior
- stop / sustain / enable logic
- first / last cycle boundaries
- internal or external control of time and stage addressing

That makes it ideal for:

- **irregular trigger streams**
- **polyrhythms**
- **non-repeating bar lengths**
- **ratchety / elastic timing**
- **selective accents**
- **meta-sequencing other percussion sequencers**
- **drum voice modulation with stage-specific CV**

In short: this module excels at **composed rhythmic structure that can still mutate**.

Why the 248t is great for percussion

From the manual, the key percussion-relevant features are:

- **Two independent Function Generators**
- **Two independent programmed pulse outputs per stage**
- **All Pulses output** for every stage change
- **Per-stage interval time**
- **Per-stage pulse assignment**
- **Per-stage operating conditions** like Stop, Sustain, Enable
- **External inputs A–D** for using external CV as time or voltage source
- **Reference output**: a downward ramp over stage duration
- **Time output**: CV from interval-time sliders
- **Independent stage-address behavior**
- **First/Last cycle markers** for custom loop lengths

Those features let you treat the MARF as:

1. a **master polyrhythm brain**
2. a **trigger composer**
3. a **CV animator for percussion voices**
4. a **timing deformer**
5. a **burst / fill / reset control center**

Core percussion mindset for the 248t

Instead of thinking “16-step drum sequencer,” think:

- each stage is an **event cell**
- the **time between events is programmable**

- the **trigger behavior is programmable**
- the **length of the loop is programmable**
- the **way the loop advances can itself be modulated**

That means you can build rhythms that feel like:

- 5 over 4
- 7 over 3
- evolving Balkan-like asymmetry
- machine funk with unstable subdivisions
- fake tuplets
- rotating accents
- broken-grid electro
- generative percussion structures

Best ways to patch it for dense rhythmic music

1. Use the two Function Generators as two independent rhythmic layers

A very effective starting point:

- **FG1** = main pulse lattice
- **FG2** = accents / fills / reset logic / modulation

Example:

- FG1 pulse output 1 → kick trigger
- FG1 pulse output 2 → closed hat trigger
- FG2 pulse output 1 → snare / clap
- FG2 pulse output 2 → open hat / rim / burst source
- FG1 or FG2 voltage output → accent CV to drum VCA / decay / pitch
- Reference output → LPG or VCA CV for natural percussive contour

Because the two FGs can have different stage timings and loop boundaries, you immediately get **phase-shifting percussion**.

---

2. Build polyrhythms by using different loop lengths with First/Last stages

The **Cycle First / Last** modifiers are one of the best rhythmic tools here.

Program:

- FG1 loop from stage 1 to stage 5
- FG2 loop from stage 1 to stage 7

Now you have a **5-against-7** cycle relationship.

Other strong pairings:

- **4 vs 5**
- **3 vs 7**
- **5 vs 8**
- **7 vs 9**
- **11 vs 4**

Because each stage can also have different durations, this is much more interesting than plain Euclidean overlap.

Tip

Use one FG for a relatively stable pulse network and the other for longer-cycle asymmetry. That keeps the groove intelligible while still becoming complex.

---

3. Create complex time signatures by varying stage interval times

The lower slider row sets **interval time per stage**. This is where the MARF becomes a monster.

Instead of keeping every stage equal, deliberately create uneven phrase lengths.

For example, to suggest a **7/8-like pattern**, make a loop of 7 stages with grouped durations like:

- long
- short
- short
- long
- short
- short
- short

This can imply groupings such as:

- **2+2+3**
- **3+2+2**
- **2+3+2**

For more fractured time feels:

- **5/8** = 2+3 or 3+2
- **9/8** = 2+2+2+3 or 3+3+3
- **11/8** = 3+3+3+2 or 2+3+2+2+2
- **13/8** = 3+3+3+2+2

You are not limited to notation, either. You can make timings that sit between conventional subdivisions and feel “elastic.”

---

4. Use the two pulse outputs as interlocking drum voices

Each stage can independently emit **Pulse 1**, **Pulse 2**, both, or neither.

That means one FG can already do two intertwined trigger patterns.

Example on one Function Generator:

- Pulse 1 = kick

- Pulse 2 = snare
- All Pulses = hat clock or shaker

Stage design idea:

- some stages only trigger kick
- some only snare
- some both for flam-like emphasis
- some neither, but still advance time

This creates **rests and ghost stages** that shape groove without always sounding a drum.

Great use:

Patch:

- Pulse 1 → kick
- Pulse 2 → hat
- All Pulses → clock divider / burst module / logic module

Now every addressed stage contributes to larger rhythmic behavior, even if no drum sounds directly on it.

---

5. Use “ghost stages” to create syncopation

Very important: not every stage needs a pulse.

Program stages that:

- advance time
- alter voltage or time modulation
- do not trigger anything

These silent stages are excellent for:

- off-balance phrasing
- fake rests
- delayed accents
- making fills land unexpectedly

- creating “holes” in dense patterns

This is one of the easiest ways to get **complicated but musical** percussion.

Advanced rhythmic strategies

6. Use per-stage external time source for elastic subdivisions

The manual states that time can be sourced from **internal sliders** or **external CV via inputs A–D**.

This is huge.

You can patch an LFO, random source, envelope, another sequencer, or even the other MARF side into A–D, then program certain stages to use **external time source**.

Result:

- some steps are fixed
- some steps stretch or compress dynamically
- certain hits rush
- certain gaps drag
- complex groove breathes instead of staying rigid

Patch idea

- External input A = slow triangle LFO
- On selected stages, set time source to external
- Use internal time on other stages

This makes only specific rhythmic cells drift, which is far more interesting than modulating the whole clock.

---

7. Use Time Multiplier CV for phrase-level rhythmic morphing

The module has a **global time multiplier** with **CV attenuverter**, scaling stage times from **half-time to 4x longer**.

For percussion, this is ideal for:

- fill transitions
- tempo swells
- broken-time intros
- dropping one layer into halftime while another stays dense

Strong patch

- FG1 time multiplier CV ← stepped random or a slow sequencer
- FG2 left stable

Now FG1 breathes in and out rhythmically while FG2 anchors the groove. This creates sophisticated cross-rhythmic motion.

---

8. Use the All Pulses output as a master event clock

The **All Pulses output** emits a pulse every time a new stage is addressed.

This is one of the module's secret weapons.

Use it to clock:

- a burst generator
- a clock divider
- a logic module
- a sequential switch
- a sample and hold
- another sequencer
- envelope retriggers for hi-hat chatter

Because MARF stage lengths are variable, the All Pulses output creates an **irregular event clock**. This is perfect for non-grid percussion ecosystems.

Example

- All Pulses → clock input of a switch
- switch alternates among noise, FM percussion, click source, metallic source
- MARF pulse outputs separately trigger envelopes

Now the timbre rotates while the rhythm is already complex.

---

9. Use Stop, Sustain, and Enable as rhythmic logic states

These are not just utility conditions; they are compositional tools.

Stop

A stop stage halts until a start pulse is received.

Use this for:

- fill waiting points
- manual performance intervention
- call-and-response structures
- “barline” control points

Example: - patch a gate from another sequencer or manual button into Start - MARF runs until a programmed stop stage - external event releases it into the next phrase

This is excellent for **semi-controlled percussion arrangements**.

Sustain

With high pulse at Start input, the stage is held as long as the gate stays high.

Use this to: - stretch a rhythmic cell - create held rests - freeze on an accent - lock into a micro-loop

Enable

The FG pauses until voltage above 5V arrives at Start input.

Use a sparse gate source to “authorize” advancement. This can produce: - gated complexity - phrase-dependent fills - polyrhythms that only emerge when another layer lines up

These three modifiers can make the MARF behave more like **rhythmic state machine logic** than a normal sequencer.

---

10. Use continuous stage addressing for scrubbing and burst-like percussion

The stage address section includes:

- **Cont / Strobe**
- internal or external addressing
- stage address knob / CV

When in **Continuous**, stage position sweeps through stages using address control and **stops the internal clock**.

This is amazing for percussive sound design.

Instead of hearing discrete sequence advance, you can use the MARF as a **scan-able bank of programmed voltages/times/pulses**.

Percussion applications

- sweep stage address externally with a fast LFO for pseudo-granular trigger selection
- strobe addressed stages with another clock
- create burst clusters by rapidly strobing neighboring programmed pulse stages

- use stage address CV from an envelope to “throw” the pattern through different regions

This is not conventional drum sequencing; it becomes **voltage-addressed rhythm topology**.

Using MARF CV to make percussion more unique, punchy, and animated

Even if you mainly care about triggers, the voltage side is where the groove becomes alive.

11. Use Voltage Output as accent, pitch, or decay control

The **Voltage Output** follows the programmed output voltage and modifiers.

Patch it to drum parameters such as:

- kick pitch
- snare tone
- clap decay
- hat filter cutoff
- LPG level
- FM amount
- distortion amount
- wavefolder depth
- sample player start position

Then each stage has its own character, not just its own trigger.

Best uses for punch

- kick pitch envelope depth

- transient click level
- LPG amount
- decay shortening on dense stages
- saturation amount on accents

A dense rhythm sounds clearer when each event has controlled articulation, not constant full energy.

---

12. Use Quantize for tuned percussion lines

If a stage is set to **Quantize**, voltage output becomes **1V/oct** scaled. You can also choose key and scale.

This means MARF can sequence:

- tom melodies
- tuned kicks
- resonant pings
- FM percussion pitches
- modal metallic percussion

For dense percussion music, a great trick is to keep one drum layer tuned:

- low tom line cycling 5 stages
- metallic FM voice cycling 7 stages
- hats/noise on pulse-only layers

That creates a hybrid of rhythm and melody without needing another sequencer.

---

13. Use Sloped stages for gliding percussion modulation

Per-stage **Sloped/Stepped** is excellent for percussion, especially when the voltage output is not going to pitch but to timbre.

Use sloped transitions to modulate:

- filter cutoff between hits
- wavefolder amount
- FM depth
- decay time
- reverb send
- distortion bias
- noise color

This gives a “morphing machine percussion” effect between events.

Nice trick

Patch Voltage Output to kick pitch FM amount or tom oscillator pitch, with some stages sloped and some stepped. You get a mix of: - hard discrete hits - sliding tuned drum motion - liquid pitch dives between accents

---

14. Use Limited or Half Range for more precise percussion control

The manual says:

- Full = 0–10V
- Half = 0–5V
- Limited ranges = 2V spans with octave offsets

For percussion modulation, **Half** or **Limited** is often better than Full because it gives more precision.

Use reduced range when controlling:

- decay
- LPG response
- pitch around a sweet spot
- FM index
- filter resonance
- click level

This helps keep the patch punchy instead of wild and unfocused.

Great percussion patch concepts

Patch 1: Polyrhythmic drum brain

Goal: Dense interlocking drums with long non-repeating cycle.

- FG1 loop: 5 stages
- FG2 loop: 7 stages
- FG1 Pulse 1 → kick
- FG1 Pulse 2 → closed hat
- FG2 Pulse 1 → snare
- FG2 Pulse 2 → metallic percussion
- FG1 Voltage Out → kick accent/pitch
- FG2 Voltage Out → snare decay or metallic timbre
- All Pulses from FG1 → shaker/burst trigger
- All Pulses from FG2 → reset/advance another modulation source

Use uneven interval times on both sides. This gives immediate polymeter and evolving phrase overlap.

Patch 2: One side clocks, the other side ornaments

Goal: Stable groove plus chaotic micro-detail.

- FG1 = main pattern, mostly regular times
- FG2 = short times, irregular pulses, no consistent bar length
- FG1 Pulse 1 → kick
- FG1 Pulse 2 → snare
- FG2 Pulse 1 → hats
- FG2 Pulse 2 → bursts, rims, clicks
- FG2 Time Multiplier CV modulated by slow random

- FG2 Voltage Out → hat decay / filter / noise level

This creates a clear backbone with hyperactive top-end percussion.

Patch 3: Asymmetric techno / electro phrasing

Goal: Groove that implies changing meter.

Program FG1 as an 8-stage loop but with stage durations grouped: - 3 short - 2 medium - 3 short

Program FG2 as a 6-stage loop with a different grouping.

Then: - kick on FG1 Pulse 1 - snare on FG1 Pulse 2 only on certain long stages - FG2 pulses drive offbeat hats and ghost percussion

Add silent stages and use voltage output to open the VCA more on sparse hits, less on dense ones.

This makes the percussion feel deliberate instead of cluttered.

Patch 4: Controlled fills with Stop stages

Goal: phrase-aware fills you can release manually or via logic.

- Main drum cycle runs normally
- A programmed **Stop** stage near the fill region halts progression
- External pulse to Start releases it
- Use another clock divider or manual gate button to decide when fills happen

This is powerful live: - regular groove continues until you “approve” the fill - then the sequence advances into a dense, weird section - it returns to main loop later

Very performance-friendly.

Patch 5: Reference output as percussion envelope

The **Reference Output** gives a downward ramp over the interval time.

Patch it to: - LPG CV - VCA CV - filter cutoff - FM amount - wavefolder symmetry

This is especially useful for: - bongos - woodblock-like pings - toms - plucked noise percussion - LPG-driven hats

Longer stage times produce longer decay-like contours; shorter stages produce tighter ticks.

This can make the MARF function like both sequencer and envelope source at once.

Patch 6: External input selected per stage

Since stage programming can choose external voltage source **A/B/C/D**, you can feed different modulation sources and select among them per stage.

For example: - A = random stepped CV - B = envelope - C = LFO - D = fixed offset

Then set certain stages to external voltage source and choose which input is active via the stage slider behavior.

Great for: - changing snare pitch source on only some hits - selecting among several accent curves - alternating between deterministic and random modulation

This is excellent for complex percussion timbral variation.

How to keep dense rhythms musical

The MARF can become chaotic fast. A few practical rules help.

1. Make one layer legible

If everything is irregular, nothing reads as groove.

Keep one of these relatively stable: - kick pattern - hat pulse stream - phrase length - accent system

Let the rest mutate around it.

2. Use silence on purpose

Empty stages are crucial. Density comes from contrast.

3. Use CV for articulation, not only pitch

For percussion, better targets are often: - decay - LPG level - click/transient - tone - distortion amount

4. Keep one Function Generator slower

Use one FG as phrase architecture and the other as detail engine.

5. Use Half/Limited range often

It makes percussive modulation tighter and easier to tune.

Specific ways to make voices punchy and percussive with MARF

If you’re sequencing drum voices or oscillator-based percussion, try these:

Kick

- Pulse output triggers envelope
- Voltage output modulates pitch amount or decay
- Reference output to LPG/VCA for natural decay
- Use short stages for tight hits, longer ones for booming accents

Snare / noise percussion

- Pulse triggers envelope
- Voltage output controls noise/filter balance or decay
- Sloped stages change tone between hits
- External time source on select stages creates drunken ghost notes

Hats

- Use one pulse stream for regular ticks
- Use second pulse stream for open-hat events
- Voltage output to filter cutoff or decay
- Use very short interval times on select stages for clustered chatter

Metallic FM percussion

- Quantized voltage output to pitch carrier/modulator
- Sloped stages for gliss between metallic hits

- Reference output into VCA index or FM amount
- Different loop length than kick for constant phase evolution

LPG percussion

- Pulse out triggers event
- Reference output directly opens 292t/LPG-like module
- Time slider becomes effectively per-hit decay control
- Uneven stage times produce organic percussion envelopes

That last one is especially Buchla-like and very strong.

A high-value workflow for building hyper-complex patterns

Method: build from structure outward

Pass 1: phrase architecture

Set: - loop lengths with First/Last - rough interval times - stop points if needed

Pass 2: trigger map

Assign: - Pulse 1 main hits - Pulse 2 counter-hits - silent stages for air

Pass 3: articulation

Program voltage rows for: - accent - pitch - decay - timbre

Pass 4: instability

Add: - external time source to a few stages - time multiplier CV - enable or sustain logic - different loop lengths on FG1/FG2

Pass 5: performance access

Reserve one or two useful controls: - manual Start release - time multiplier amount - stage address strobe - switching gate source for ART / pulse behavior

That gives complexity you can still perform.

Best module pairings for this purpose

The 248t will shine with:

- drum voices with CV over decay/pitch/tone
- LPGs
- burst generators
- logic modules
- sequential switches
- clock dividers/multipliers
- sample and hold
- noise sources
- FM percussion voices
- resonant filters used as percussion
- oscillators with fast pitch response

Especially good pairings: - LPG + Reference out - logic + pulse outs - burst generator + All Pulses - switch + external CV inputs A–D - another sequencer clocked by All Pulses

---

Simple “starter patch” for instant complex percussion

If you want one practical first patch:

FG1

- stages 1–5 active as loop
- uneven interval times
- Pulse 1 on stages 1, 3, 5 → kick
- Pulse 2 on stages 2, 4 → hat
- Voltage out → kick accent

FG2

- stages 1–7 active as loop
- shorter average interval times
- Pulse 1 on scattered stages → snare/clap/rim
- Pulse 2 on dense cluster of 2–3 adjacent stages → hat flurry
- Voltage out → hat decay or snare tone

Extra

- All Pulses from FG2 → burst module or switch
- Reference output from FG1 → LPG CV for kick/tom contour
- Modulate FG2 Time Multiplier slowly

That one patch already gives: - polymeter - asymmetry - accents - fills - evolving overlap - timbral change

---

Bottom line

For hyper-complex percussion, the **248t MARF** is best used as a **rhythmic composition engine**, not just a trigger sequencer.

Its strengths are:

- per-stage time variation
- independent dual generators
- per-stage pulse programming

- custom loop boundaries
- stage-conditioned behavior
- external control over time and voltage
- useful auxiliary outputs like All Pulses, Time, and Reference

If you want dense rhythmic music with: - polyrhythms - odd meters - evolving phrase lengths - articulated accents - non-repeating drum systems

the winning strategy is:

1. give the two FGs different loop lengths
2. vary stage durations intentionally
3. use pulse 1 and pulse 2 as interlocking trigger layers
4. reserve some stages as silent or logic-only
5. use voltage/reference outputs to animate drum parameters
6. add external time modulation only to selected stages

That will get you from “sequenced drums” to **highly structured rhythmic machinery**.

Generated With Eurorack Processor