

# Buchla and Tiptop Audio – 248t

---

- [Manual PDF](#)
- 

[248t MARF Manual PDF](#)

## Using the Tiptop/Buchla 248t MARF to Create Full-Length Songs in Eurorack

---

The **248t MARF** is unusually well suited to solving the classic Eurorack problem:

- easy to make a cool loop
- hard to make a **piece with structure, variation, sections, transitions, and return**

That is because the MARF is not just a sequencer. It is really a **16-stage programmable control composition system** with:

- per-stage pitch
- per-stage duration
- per-stage pulse programming
- per-stage behavioral rules
- two independent function generators reading the same 16-stage data in different ways
- manual, clocked, voltage-addressed, or externally controlled access to stages
- preset storage
- external CV routing per stage
- quantized or continuous voltages
- programmable cycle boundaries, stops, sustains, and enables

In practice, this means the MARF can act like:

- a melodic sequencer
- a bass sequencer
- a phrase sequencer
- a scene sequencer
- a modulation timeline
- a transition generator
- a song-form controller
- a performance brain for an entire patch

Below is a musician-focused analysis of how to use it specifically for **full-length song construction**, not just short loops.

---

## Why the 248t is Strong for Song Writing

---

Most Eurorack sequencers are optimized for:

- one steady clock
- one pattern length
- one pitch lane
- one gate lane
- quick loop gratification

The MARF gives you things that are much more composition-friendly:

### 1. Per-stage time

---

Each stage can have its own duration.

That means sections can breathe:

- long intro notes
- short rhythmic verses
- stretched breakdowns
- accelerating fills

- held drones between sections

This is a huge step toward song form because songs are not just pitch sequences—they are **time architectures**.

## 2. Two independent function generators

---

The manual describes **two function generators (FGs)** that can independently access the same 16 programmed stages.

This enables: - melody and bass derived from the same underlying form - one FG running the “foreground” while the other controls transitions or macro modulation - one FG stepping normally while the other jumps or is manually/strobe addressed - one FG controlling pitch while the other controls arrangement events

This is one of the biggest reasons the module can make **full arrangements**, not just riffs.

## 3. Stage behaviors: stop, sustain, enable, first, last

---

These are compositional tools.

- **Stop** = wait for a cue
- **Sustain** = hold during a gate
- **Enable** = wait until a condition is met
- **First/Last** = define loop boundaries

That means you can build patches where the “song” does not just endlessly cycle—it can:

- wait at transition points
- be advanced by the performer
- react to another sequencer, gate source, footswitch, or clock divider
- loop only selected sub-sections until you decide to move on

That is exactly how arrangement works in music.

## 4. External voltage inputs A–D per stage

---

Each stage can choose internal or external source behavior for voltage and time.

This means the MARF can become a **programmable selector/router/composer of other modulation and sequencing sources**.

For example: - verse stages use internal pitch - chorus stages switch to external CV from another sequencer - breakdown stages use random CV - outro stages derive time from an envelope follower or slow LFO

So the MARF can define **when** each source is used across a song.

## 5. Presets

---

The module stores and recalls presets.

That means song sections can be prepared in advance:

- preset 1 = intro
- preset 2 = verse
- preset 3 = chorus
- preset 4 = bridge
- preset 5 = outro

Even if you do not use preset changes as hard scene jumps, presets are valuable for performance and rehearsal.

---

# Best Mental Model: Think of the MARF as a “Song Form Sequencer”

---

Instead of using the 248t like a normal 16-step sequencer, use it as a **hierarchical control system**.

Think in layers:

- **micro level:** notes, gates, glides, timing
- **phrase level:** 4–8 stage melodic cells
- **section level:** intro / verse / chorus / bridge
- **macro level:** filter openings, density, ratcheting, drum fills, sends, timbre shifts

The MARF excels when you let one layer control another.

---

## Core Strategies for Building Full-Length Songs

---

### Strategy 1: Use One FG for Notes, the Other for Arrangement

---

A very effective setup:

#### FG1

Use for **melody or bass pitch sequencing**: - ART output or voltage output to oscillator pitch - pulse outputs for envelopes - quantize enabled on desired stages - sloped on selected stages for portamento

#### FG2

Use for **arrangement control**: - send voltage out to a sequential switch, VCA CV, filter cutoff, effect send, clock multiplier CV, mixer automation, or drum density control - use pulse outputs to trigger fills or transitions - set different stage lengths from FG1

#### Result

FG1 plays the musical line.

FG2 slowly changes the world around it.

This creates the feeling of a song developing over time without needing 20 separate sequencers.

### Example patch

- FG1 Voltage Out → VCO 1 pitch
- FG1 Pulse 1 → envelope for bass voice
- FG2 Voltage Out → filter cutoff on poly/lead bus
- FG2 Pulse 1 → trigger drum fill logic
- FG2 Pulse 2 → advance sequential switch selecting reverb send or alternate modulation

This works because the MARF's two FGs can produce **coordinated but independent timelines**.

---

## Strategy 2: Use the 16 Stages as Song Sections, Not Just Notes

---

A powerful trick is to stop thinking "16 notes" and instead think:

- stage 1 = intro
- stage 2 = intro variation
- stage 3 = verse A
- stage 4 = verse B
- stage 5 = pre-chorus
- stage 6 = chorus A
- stage 7 = chorus B
- stage 8 = fill
- stage 9 = verse return
- stage 10 = breakdown
- stage 11 = tension rise
- stage 12 = chorus peak
- stage 13 = bridge
- stage 14 = ambient drop
- stage 15 = final chorus
- stage 16 = outro

Then each stage does not necessarily represent one note. Instead it represents a **state**.

You do this by patching MARF outputs to control:

- transposition
- clock rates
- switch positions
- mixer VCAs
- drum muting
- effect sends
- sample selection
- wavefolder depth
- LPG decay
- logic thresholds

The actual note-by-note content can come from other sequencers. The MARF determines the **song progression**.

### **Great companion modules**

- sequential switches: Verbo Sequence Selector, Doepfer A-151, Noise Engineering Vice Virga
- matrix mixers / VCAs: 4ms VCA Matrix, Befaco Hexmix VCA expander style workflows
- clock tools: Pamela's Pro Workout, Shakmat, 4ms QCD
- logic modules: Joranalogue Compare 2, Doepfer logic, Klavis Two Bits
- performance mixers: Worng, Happy Nerdin, Befaco, WMD style mixers if available used

This is one of the most practical ways to get from loop to song.

---

## Strategy 3: Build “Conditional Sections” with Stop, Sustain, and Enable

---

The MARF manual’s most song-relevant features are these stage modifiers:

- **Stop**
- **Sustain**
- **Enable**
- **First / Last**

These can create real musical form.

### Stop stages as transition checkpoints

---

Program a stop stage at the end of a verse.

The sequence reaches it and waits.

You decide when the chorus starts by sending a start pulse.

This is fantastic live because it avoids getting trapped in automatic loops that move on too soon.

Use cases: - verse loops until you hit a manual trigger - breakdown holds until drummer/percussion line resolves - ambient intro waits until you launch beat

Patch ideas: - manual gate button module → MARF start - footswitch interface → MARF start - end-of-cycle from another sequencer → MARF start - comparator threshold from envelope follower → MARF start when energy rises

### Sustain stages as extended holds

---

Use sustain for: - held root note before drop - long pad during breakdown - tension note before chorus - drone ending

If a high gate is present at start input, the stage holds.

Patch that gate from: - manual pressure controller - keyboard gate - logic AND of bar counter and manual button - long gate from clocked trigger sequencer

## **Enable stages as “wait until event happens”**

---

Enable causes the FG to pause until voltage above 5V arrives at start input.

This is amazing for semi-generative song form: - wait for bar count signal - wait for kick pattern completion - wait for external sequencer reset event - wait for performer’s command - wait for comparator from audio envelope

This turns arrangement into a responsive system rather than a rigid timeline.

---

# **Song Construction Approaches**

---

## **Approach 1: Traditional Intro–Verse–Chorus–Bridge Song**

---

Here is a practical full-song architecture using the 248t.

### **Voice allocation**

---

- Voice 1: bass
- Voice 2: lead/melody
- Voice 3: pad/drone
- Drums: separate trigger sequencer or sample player
- FX: delay/reverb send

# MARF role

---

## FG1

Controls bass pitch and bass gate articulation.

## FG2

Controls section-level modulation: - filter cutoff - drum density - send levels  
- transposition offset through precision adder - switch between melodic sources

# Patch concept

---

- FG1 Voltage Out → quantized bass oscillator pitch
- FG1 Pulse 1 → bass envelope
- FG1 Pulse 2 → accent VCA / wavefolder burst
- FG2 Voltage Out → precision adder offset for lead sequencer transposition
- FG2 Time Out → modulate delay time or reverb send amount via attenuator
- FG2 All Pulses → clock divider reset or fill sequencer trigger

# Stage design example

---

## Stages 1–2: intro

- slow interval times
- sparse pulse programming
- low filter values via FG2
- maybe external voltage source for drifting pad timbre

## Stages 3–6: verse

- bass active
- lead muted or low in mix

- moderate pulse density
- stepped values

### **Stages 7–8: pre-chorus**

- shorter stage times
- sloped pitch transitions
- filter rises
- drum fill trigger from pulse 2

### **Stages 9–12: chorus**

- more pulse outputs active
- transposition up
- wider range or different external source selected
- effects send increased

### **Stage 13: break**

- sustain enabled
- drums muted through VCA control
- long held note

### **Stages 14–15: final chorus**

- return with more density
- faster times
- accent pulses on more stages

### **Stage 16: outro stop**

- stop stage
- lets you end or hold manually

This is already a song, not just a loop.

---

## Approach 2: MARF as Master Arranger for Other Sequencers

---

This is arguably the most useful real-world workflow.

Let other modules do what they are best at: - drum sequencing - ratcheting  
- Euclidean patterns - probabilistic melody

Use the MARF to decide **when each system is heard and how it evolves**.

### Example supporting modules

---

- drum sequencer: Steppy, Metron, Grids-style, Numeric Repetitor, Pam's
- melodic sequencer: Metropolis, Rene, Moskwa, Eloquencer, Bloom, etc.
- switch: A-151 or vice versa type switch
- mixer/VCA: performance mixer
- precision adder / quantizer
- filter / effects

### MARF jobs

---

- transpose melody sequencer per section
- mute/unmute voices using VCAs
- switch between multiple drum clocks
- gate fills on certain sections
- change the active modulation source through sequential switching
- alter time flow with Time Multiplier CV

### Example arrangement patch

- External sequencer provides 8-step melody
- MARF FG2 Voltage Out → precision adder → melody pitch transpose
- MARF FG2 Pulse 1 → open VCA for hi-hat bus only in chorus

- MARF FG2 Pulse 2 → trigger fill generator every 8th section
- MARF stages programmed with First/Last to define looped verse and chorus blocks
- Stop stage before bridge, manually advanced live

This gives you hands-on song structure while keeping your favorite sequencers in play.

---

## Approach 3: Through-Composed Pieces with Variable Timing

---

Because each stage has its own duration, the MARF is excellent for music that should not feel grid-locked.

This is perfect for: - ambient - kosmische - electroacoustic - generative-but-repeatable works - soundtrack style pieces - progressive pieces with rubato

### Patch example

- FG1 controls lead pitch with quantized stages, some sloped
- FG2 controls a drone oscillator or wavetable scan
- stage times vary widely
- some stages use external timing from slow random voltages
- stop stages inserted at major phrase boundaries
- all pulses output clocks occasional percussion
- reference output drives LPGs for plucked contours without needing separate envelopes

Because the reference output is a downward ramp over the stage time, you can shape phrases with very organic contour changes.

This is one of the most underused features for making music feel composed.

---

# Practical Full-Song Techniques

---

## 1. Create section contrast by changing time behavior, not just notes

---

A lot of weak Eurorack songs fail because every section uses the same rhythmic density.

With the MARF: - intro = long interval times - verse = moderate - chorus = shorter, more active - bridge = external time source - outro = long, decaying times again

You can create true section identity through timing.

## 2. Use pulse outputs as arrangement triggers

---

Each stage can independently emit Pulse 1 and/or Pulse 2.

Use these not only for notes, but to trigger: - drum fills - burst generators - reset events - sample playback - envelope changes - switch advances - effect ducking - scene changes on external modules

This is how the MARF becomes a **conductor**.

## 3. Use first/last markers for nested loops

---

Set: - stages 1–4 = verse loop - stages 5–8 = chorus loop - stages 9–12 = bridge loop

Then manually or externally reassign cycle boundaries during performance, or store alternate versions in presets.

Even if you do not dynamically rewrite stage programming live, you can organize the module so that a given block behaves like a mini-section.

## 4. Use stage address control for jumps and recalls

---

The stage address controls allow continuous or strobe addressing, with internal or external control.

This means you can: - jump to scenes via CV - scan manually across sections - use a joystick or pressure output to move between song regions - use another sequencer as a meta-sequencer to choose which MARF stage is active

This is huge for performance composition.

### Example

Patch a slow 4-step sequencer into stage address external input: - value 1 = intro stage - value 5 = verse stage - value 9 = chorus stage - value 13 = breakdown stage

Now the MARF is no longer merely stepping linearly; it is being **arranged from above**.

## 5. Use external sources A–D as “instrument roles”

---

Per the manual, in external source mode the stage slider can choose among A/B/C/D.

Patch: - A = main melody CV - B = random CV - C = keyboard CV - D = fixed offset or alternate sequencer

Now each stage decides which source becomes active.

This is incredibly compositional. Example: - verse uses melody CV - fill grabs random - chorus uses keyboard transpose line - breakdown uses static drone offset

Few modules let you **compose source selection per stage** like this.

---

# Excellent Module Pairings for Song Building

---

## 1. With a clock ecosystem

---

Examples: - Pamela's Pro Workout - 4ms QCD - Tempi - clock dividers / logic clocks

Why: - clocks can advance other sequencers while MARF controls phrase timing - MARF pulse outputs can reset clocks or launch fills - variable stage durations can produce phrase-level time movement against fixed subdivisions

Use case: - drums locked to master clock - MARF melody drifts in larger phrase timing - fills triggered by MARF pulses at section boundaries

This creates tension between machine precision and composed form.

---

## 2. With sequential switches

---

Examples: - Doepfer A-151 - Verbos Sequence Selector - Noise Engineering switches - Joranalogue Switch 4-type ideas

Why: - MARF voltages or pulses can choose which voice, modulation source, or drum pattern is active - lets one sequence become many song sections

Use case: - 3 different bass timbres patched to switch inputs - MARF pulse advances switch only during chorus stages - FG2 voltage controls crossfade or filter on the chosen voice

---

## 3. With precision adders and quantizers

---

Even though MARF can quantize internally, precision adders are useful for section transposition.

Use case: - another sequencer runs a motif - MARF outputs transposition values per section - intro = 0 semitones - pre-chorus = +2 - chorus = +5 - bridge = -3

This is one of the fastest ways to get “song” feeling from a loop.

---

## 4. With VCAs and performance mixers

---

A song is often really about **what is present and absent**.

Use the MARF to automate: - voice entrances/exits - effect send amounts - filter bus levels - sidechain/depth behaviors - percussion density

Use case: - FG2 voltage slowly opens chorus pad VCA - pulse 2 triggers accent envelopes on selected stages - stop stage freezes before the drop while mixer mutes drums

---

## 5. With LPGs and the Reference output

---

The reference output gives a downward ramp over the interval time of the stage.

This is gold for musical phrasing: - plucks - decays - acoustic-feeling swells - natural phrase tails

Patch it to: - 292t LPG - low-pass gate style modules - filter cutoff - VCA CV - effect return level

Then each stage carries its own contour.

That is very composition-friendly.

---

## 6. With ART oscillators like the 259t

---

The manual specifically notes strong integration with ART and recommends updated 259t firmware.

Benefits: - accurate pitch slides/glides - direct digital pitch control - fast response across staged voltages

This makes the MARF especially strong for: - melodic leads - precise bass writing - gliding hooks - recurring motifs that need exact intonation

For full songs, this helps keep the lead line consistent while the surrounding patch becomes more complex.

---

## Five Concrete Full-Length Song Recipes

---

### Recipe 1: Techno / Electro Arrangement Brain

---

#### Goal

Turn a strong 1-bar groove into a 6-minute track.

#### Patch

- drums from drum sequencer
- bass from MARF FG1
- synth stab from another sequencer
- FG2 controls arrangement

#### MARF use

- FG1: 8–16 stage bassline with quantized stages
- Pulse 1: bass envelope
- Pulse 2: occasional accent and fill trigger
- FG2 Voltage: filter bus / stab transposition
- FG2 pulses: mute logic or fill clock

## Song form

- stages 1–4: stripped intro
- stages 5–8: full groove
- stage 9: break
- stages 10–12: build
- stages 13–16: peak section

## Extra trick

Use a stop stage before the drop and manually restart on the one.

---

## Recipe 2: Ambient/Drone Composition

---

### Goal

A long evolving piece with defined sections.

### Patch

- FG1 → lead oscillator pitch via ART
- FG2 → wavetable position / filter / reverb amount
- reference output → LPG on second voice
- external A–D = random, joystick, keyboard, fixed offset

### MARF use

- long stage times
- sloped pitch transitions on selected stages
- external voltage source on some stages for unpredictability
- sustain stages for frozen drones
- first/last markers for looped midsection

## Song result

Feels composed and expansive, not like a repeating generative patch.

---

## Recipe 3: Berlin School / Kosmische

---

### Goal

Repeating sequences with evolving structure.

### Patch

- external 8-step sequencer for ostinato
- MARF FG1 for transposition and glide lead
- MARF FG2 for filter/effect evolution
- all pulses out used as auxiliary clock/reset source

### Song form

- intro: MARF lead only
- ostinato fades in
- transposition changes every several MARF stages
- bridge uses stop stage and external manual launch
- final section increases time multiplier and opens filter

The MARF becomes the phrase architect over a more repetitive engine.

---

## Recipe 4: Generative but Repeatable Composition

---

### Goal

A patch that can surprise you but still has recognizable song form.

### Patch

- random sequencer into external input A
- stable melody sequencer into B
- keyboard or pressure CV into C
- fixed voltage into D

Per-stage source selection chooses which world you are in.

## **Song use**

- intro selects D and B mostly
- verse uses B
- fills use A
- bridge uses C for live intervention
- outro returns to D

This is a very elegant way to make generative systems feel authored.

---

## **Recipe 5: Live Performance Song Builder**

---

### **Goal**

Perform arrangements by hand without losing synchronization.

### **Patch**

- manual gate button / footswitch to MARF start
- stop stages at section endings
- pulses sent to cue lights or trigger utility events
- FG1 for primary melodic line
- FG2 for voice presence and effects

### **Workflow**

- let verse cycle
- when ready, hit start to enter pre-chorus
- let sustain hold before drop
- release into chorus
- stop again at breakdown
- use stage address display to know where you are
- jump via external address if needed

This is where the MARF really shines as a performance composition module.

---

## A Good “Minimum System” Around the MARF for Song Writing

---

If your goal is full songs, a very effective surrounding system would be:

- **248t MARF**
- 1–2 oscillators or one complex oscillator
- one bass voice and one lead/pad voice
- drum voice set or sampler
- a clock source / divider
- at least one sequential switch
- at least one precision adder
- several VCAs
- one LPG or VCA/filter combo
- performance mixer
- delay/reverb
- logic/comparator utilities

That combination lets the MARF govern: - pitch - phrase length - transitions  
- instrument entrances - fills - section changes - song ending

---

## Specific Tips from the Manual That Matter Musically

---

### Quantize / Continuous

---

Use **quantize** for: - basslines - tonal hooks - transposition lanes

Use **continuous** for: - filter moves - drones - transitions - addressing - FX automation

## Sloped / Stepped

---

Use **sloped** not just as glide, but as section glue: - verse to chorus pitch rise - melting breakdown lines - softer transitions between phrases

## Half range and limited range

---

The manual notes full, half, and limited ranges.

For musical writing, **half range** is often more practical for pitch composition because it keeps the sequence in a useful register and improves precision.

## External time mode

---

This is a sleeper feature.

If time source is external, the time sliders/time outputs can behave like another compositional CV layer.

That means you can decouple “time programming” from “time output use,” which opens interesting parallel structure patches.

## All pulses output

---

Use it as: - reset for another sequencer - barline trigger - sample advance - clock for percussion ornamentation - input to clock divider for section-based trigger events

---

# A Useful Workflow for Actually Finishing Songs

---

## Step 1: Program a compelling 4–8 stage core

---

Do not start with all 16 stages.  
Create one strong motif.

## Step 2: Duplicate its function across stages as sections

---

Use the rest of the stages to create: - sparse version - denser version - transposed version - gliding version - held version - breakdown version

## Step 3: Assign pulse outputs for section events

---

Do not use pulses only for note gates.  
Reserve some for: - fill triggers - drum mutes/unmutes - switch advances - effect throws

## Step 4: Insert stop/sustain points

---

These create performable structure.  
Without them, you are back to looping.

## Step 5: Use FG2 as the macro layer

---

Even if FG1 handles the music, FG2 should control: - timbre - mix - density - transposition - switching

## Step 6: Rehearse section transitions

---

The MARF is deep.

Practice: - when to start/stop - when to hold - when to launch chorus -  
when to recall presets - when to change stage addressing

That performance aspect is what turns a patch into a song.

---

## The Big Idea

---

The 248t MARF is one of the rare Eurorack modules that can genuinely help with **musical form**, because it does not only sequence notes—it sequences:

- time
- behavior
- source selection
- transitions
- phrase boundaries
- structural events

If most sequencers give you a **loop**, the MARF gives you a way to design a **timeline**.

The best use of it for full-length songs is usually not “make one 16-step melody,” but rather:

- let one function generator handle a musical line
- let the other handle arrangement and macro change
- use stop/sustain/enable as section logic
- use external inputs and switches to swap compositional roles
- use pulses to trigger fills and transitions in the rest of the system
- think in terms of sections, not steps

Used that way, the MARF becomes less of a sequencer and more of a **song form instrument**.

---

# Summary of Best Full-Song Uses

---

The 248t MARF is especially strong for:

- **section-based composition**
- **live arrangement of looping material**
- **variable phrase timing**
- **macro control over other sequencers**
- **transposition and voice management**
- **performable transitions**
- **through-composed modular pieces**
- **hybrid deterministic/generative song forms**

If you want, I can also turn this into any of the following:

1. a **practical patch cookbook** with 10 specific song patches
2. a **genre-focused guide** for techno / ambient / Berlin school / experimental
3. a **“starter template” patch** for building full songs with the 248t and a small system
4. a **signal-flow diagram** showing how to patch the MARF as a song arranger

Generated With [Eurorack Processor](#)