

Multi-Rate Phase-Current Sampling Plan

Plan for multi-rate phase-current sampling on the STM32H723ZG to support ripple-aware MPC.

Document ID	OV-SW-MAINMCU-MPC-SAMPLING
Version	1.0
Date	2026-07-13
Status	draft
Applies to	openvvvf-control-module, software-target-main-mcu
Product line	openvvvf

1. Plan: Multi-Rate Phase-Current Sampling for MPC / Ripple Separation

Status: Deferred - implement after SimpleFOC is working.

Goal: Sample phase currents multiple times per PWM period so a software model can separate switching ripple from fundamental current, enabling ripple-aware MPC or advanced current control.

Hardware: STM32H723ZG

- TIM1: center-aligned PWM + ADC trigger generator.
 - ADC1 + ADC2: share ADC12 clock domain, can run dual-mode or independently.
 - ADC3: independent ADC on separate clock domain.
-

1.1. Sampling Requirements

- **All three phases measured every PWM period** for safety oversight and cross-checking (`iu + iv + iw == 0` is a check, not a substitute).
 - **Mixed resolution per trigger:**
 - One phase at **16-bit** with oversampling (high-res path).
 - The other two phases at **12-bit** (fast snapshot path).
 - **Timestamped samples** aligned to PWM/switching angle.
 - **Spare ADC capacity** for other peripherals (DC link, throttle, temps) via time-division multiplexing.
 - **Target PWM frequency:** 6 kHz center-aligned (software cap).
-

1.2. Hardware Constraints

Resource	Capability
ADC1 + ADC2 dual mode	Two simultaneous 16-bit samples per trigger. Both ADCs must use same resolution and sample time.
ADC1/ADC2/ADC3 independent	All three can trigger on the same TIM edge but use different resolutions and channels. Slightly different sample-window lengths, but start time is identical.
TIM1 CH4	Can generate two ADC trigger edges per center-aligned PWM period via <code>OC4REF_RISINGFALLING</code> .
HRTIM	Alternative for 4+ triggers per period; requires peripheral redesign.

Important: Mixed-resolution simultaneous sampling requires ADC1/ADC2/ADC3 to run in **independent** mode, not dual mode. The current `PhaseCurrentADC` dual-mode path must be replaced for MPC.

1.3. Clocking and Timing Budget

From current RCC/ADC config:

- ADC1/ADC2 clock = PLL2P (96 MHz) ÷ 4 = **24 MHz**
- ADC3 clock = PLL2P (96 MHz) ÷ 1 = **96 MHz**

With `SAMPLETIME_32CYCLES_5` :

Conversion	ADC Clock Cycles	Time
ADC1/ADC2 16-bit, 16× oversampled	$16 \times (32.5 + 16.5)$	32.7 μs
ADC1/ADC2 12-bit	$32.5 + 12.5$	1.9 μs
ADC3 12-bit	$32.5 + 12.5$	0.47 μs

At **6 kHz center-aligned PWM**, period = **166.7 μs**.

1.4. Recommended Two-Trigger Architecture

Use **TIM1 CH4** as trigger source with `TIM1 TRGO = OC4REF_RISINGFALLING`. In center-aligned mode this gives **two common trigger edges per PWM period**.

1.4.1. Per-Period Sampling Schedule

Trigger	ADC1	ADC2	ADC3
T0	Phase U @ 16-bit OS	Phase V @ 12-bit	Phase W @ 12-bit
T1	Phase V @ 16-bit OS + Phase W @ 16-bit OS (scan)	Spare / TDM	Spare / TDM

Result every PWM period:

- U: one 16-bit sample
- V: one 12-bit + one 16-bit sample
- W: one 12-bit + one 16-bit sample
- Two spare ADC slots for other sensors

Total conversion load: $\sim 32.7 \mu\text{s} + 65.3 \mu\text{s} = \sim 100 \mu\text{s}$, leaving $\sim 66 \mu\text{s}$ headroom.

1.4.2. Trigger Placement

- Place CH4 compare so the two edges are separated by at least $\sim 40 \mu\text{s}$ to allow the 16-bit oversampled conversion to finish.
- Example: `CCR4 = 3 × ARR / 4` gives edges near 3/8 and 5/8 of the period (spacing $\sim 42 \mu\text{s}$).
- Position relative to actual PWM switching angles can be adjusted by changing `CCR4` per operating point if desired.

1.5. Three-Trigger Option (Future)

If the MPC model needs denser data, rotate the 16-bit phase so no trigger carries a 2-channel 16-bit scan:

Trigger	ADC1	ADC2	ADC3
T0	U @ 16-bit OS	V @ 12-bit	W @ 12-bit
T1	V @ 16-bit OS	W @ 12-bit	U @ 12-bit
T2	W @ 16-bit OS	U @ 12-bit	V @ 12-bit

Conversion load: $3 \times \sim 32.7 \mu\text{s} \approx 98 \mu\text{s} + \text{margin}$. Fits in $166.7 \mu\text{s}$.

Caveat: TIM1 alone cannot generate three common trigger edges per center-aligned period.

Implementing this requires either:

- HRTIM (recommended), or
- Split trigger sources (e.g., ADC1 on TIM1 TRGO, ADC2/ADC3 on TIM1 TRGO2), which breaks simultaneity.

1.6. Data Flow and Timestamping

1. DMA transfers ADC results to RAM_D1 circular buffers.
2. In DMA half/complete callbacks, record:
3. Raw ADC values
4. Trigger index (T0/T1)
5. `TIM1->CNT` latched at callback time (or derive angle from trigger index)
6. Current PWM duty cycles / switching state at that instant
7. Software model uses:
8. Known inductance, resistance, DC link voltage
9. Switching state and duty cycle
10. Timestamps
to separate ripple and reconstruct fundamental currents.

1.7. Implementation Steps (When Started)

1. **Disable dual-mode ADC** in `PhaseCurrentADC` ; reconfigure ADC1/ADC2/ADC3 as independent.
 2. **Reconfigure TIM1:**
 3. Keep CH1/CH2/CH3 for center-aligned PWM.
-

-
4. Add CH4 in PWM mode for ADC trigger generation.
 5. Set `TIM1->CR2.MMS = OC4REF_RISINGFALLING` .
 6. **Set up ADC trigger:**
 7. ADC1, ADC2, ADC3 all use `ADC_EXTERNALTRIG_T1_TRGO` with `RISINGFALLING` edge.
 8. **Set up DMA:**
 9. One DMA stream per ADC (ADC1, ADC2, ADC3) to `RAM_D1` buffers.
 10. Circular mode, word/half-word alignment as needed.
 11. **Implement per-trigger channel scheduling:**
 12. Reconfigure ADC channels between triggers, or pre-configure scan sequences.
 13. Use DMA double-buffer or circular buffer indexing to know which trigger produced which data.
 14. **Add timestamp/duty-cycle logging** alongside raw ADC data.
 15. **Add TDM schedule** for non-phase ADC channels (DC link, throttle, temps).
 16. **Validate:** check sample spacing, ensure no ADC overrun, verify ripple reconstruction.
-

1.8. Open Decisions

- Exact trigger angle(s) relative to switching events.
- Whether to keep 16× oversampling or trade it for more triggers.
- Whether to sample reference pins differentially or use calibrated single-ended offset.
- Whether to migrate to HRTIM for three or more triggers per period.