FOC firmware improvements

Improvements

In the next few slides, I am proposing possible improvements based on this old discussion, this old discussion and the attached slides, discussions with community members during GPN, hints I have seen in other BLDC firmwares (Smartesc , MESC, Hoverboard), and ideas I got while studying timers/PWM/ADC, STM documentation:
ADC modes
ADC accuracy
STM Single/dual FOC SDK
STM32 for motor control applications
Zanppa's firmware

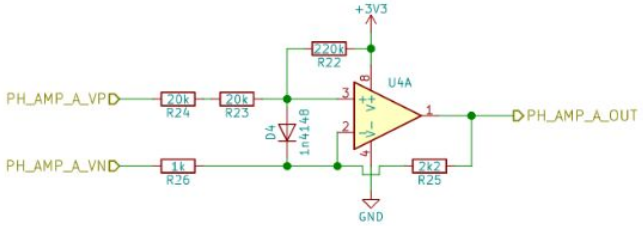
How FOC firmware works now

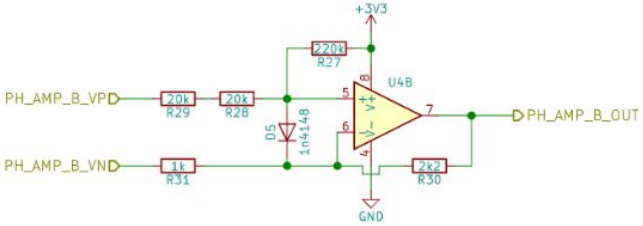
Hardware

For FOC to work, motor phase current has to be measured. Hoverboard mainboard use Opamps to amplify the low side mosfet drain-source on resistance(RDSon) and measure the current of 2 phases via ADC. A and B currents are available for the left wheel, B and C for the right wheel.

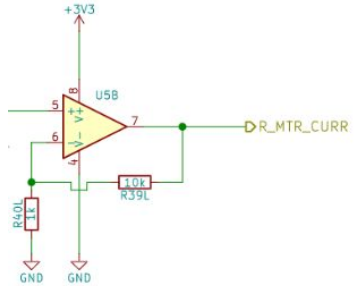
Based on Kirchhoff's law, Ia + Ib + Ic = 0, so we can calculate the 3rd phase current without measuring it (Ic = -Ia-Ib).

More details about Dual-Shunt Current Sensing in page 4 of this document.





The hoverboard mainboard also have shunt resistor and an Opamp to measure the DC current for each wheel, it is used for overcurrent protection.



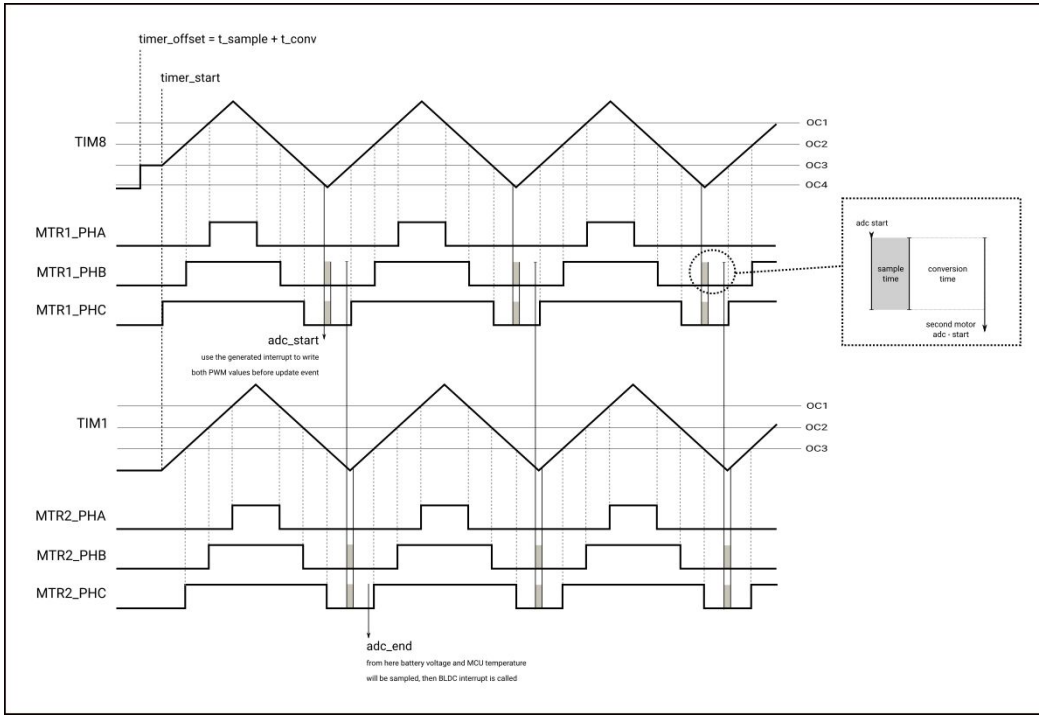
PWM/Timer

MCU is running at 64Mhz, PWM is running at 16Khz. The PWM period is MCU frequency / PWM frequency : 64Mhz / 16Khz = 4000 cycles =62.5us (center aligned)

As ADC cannot measure all currents at the same time for both wheels, TIM1 and TIM8 are started at the same time as master/slave, but TIM8's counter is 80 cycles ahead so both motor's phase current is sampled at the right time :
7.5(sampling) + 12.5(conversion) = 20 ADC cycles * 4 = 80 system clock cycles

ADC sampling is triggered at the middle of the ON time of the low side mosfet(center aligned). DC current is sampled first to delay the sampling of phase current and help avoid the hysteresis when the mosfet turns ON, but only half of the PWM On is available for the measurement window.

Each Timer has 3 channels, and complementary PWM.



ADC sampling time

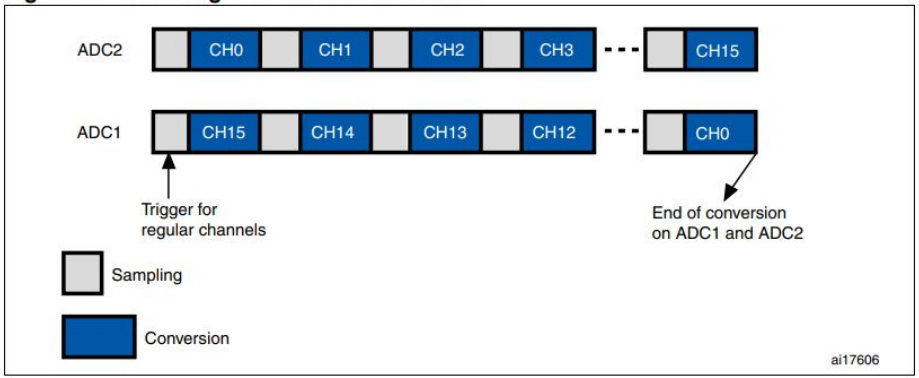
ADC frequency is 64Mhz/4 = 16Mhz. It's boosted compared to the max 14Mhz mentioned in the datasheet to be able to sample faster. The current sampling times are in below table, conversion time is fixed (12.5 cycles).

Rank	ADC1	Sampling (cycles)	Total time (us)	ADC2	Sampling (cycles)	Total time (us)
1	Right DC(PC1/CH11)	1.5	0.875	Left DC(PC0/CH10)	1.5	0.875
2	Left U(PA0/CH0)	7.5	1.25	Left V(PC3/CH13)	7.5	1.25
3	Right U(PC4/CH14)	7.5	1.25	Right V(PC5/CH15)	7.5	1.25
4	Bat V(PC2/CH12)	7.5	1.25	ADC_IN1(PA2/CH1)	7.5	1.25
5	Temp(CH16)	239.5	15.75	ADC_IN2(PA3/CH3)	7.5	1.25
Total			20.375			5.875

ADC Mode

Currently all adc measurements are done in Dual regular simultaneous mode.
It takes 20.375us for ADC1 because of the temperature measurement.
It's not impacting the measurement window as the first signals to be sampled and converted are the most critical ones (currents), but it leaves less time for foc calculation to run in the same PWM cycle.

Figure 7. Dual regular simultaneous mode



FOC calculation

When all the ADC values have been written to the DMA, an interrupt is triggered.
ADC measurement + FOC calculation + duty cycle update shouldn’t exceed approx 30us which is half the PWM period.
Now it take around 70us for both wheels.
I think there is a pwm cycle of lag from output perspective at least for one of the wheels.

Task	**Duration (us)**
ADC sampling	20
Battery voltage filtering	1
Applying offsets	1*2
Current chopping	1*2
Buzzer processing	1
Apply margin for FOC	0.6
Reading hall sensor	0.6*2
Initializing FOC input parameters	1x2
FOC calculation	17*2
Clamping and applying new duty cycle	3*2
Total	~70

PWM duty cycle

To have enough time for current measurement, a margin is used to clamp the duty cycle at maximum 94.5% (0<110<1000<1890<2000).
This should allow a window of 2,6875us for current measurement (half ON time at 11% duty cycle minus deadtime).
I have measured 6us minimum ON time on the oscilloscope.
It is not enough to cover the 3.375us needed for DC and phase current measurements ? (Could be the 0,75us deadtime was missed in the calculation, more details in the next slide)

Midpoint clamp is used for FOC and is used here.
For SIN, 3rd Harmonic injection is used here (look up table).
Nice visualization of different zero sequence modulation.

PWM - deadtime

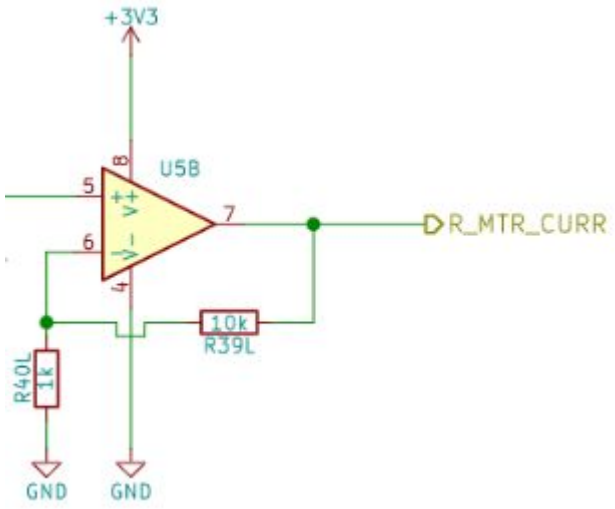
A dead-time of 750ns (48 cycle at 64Mhz) is inserted to prevent high side and low side mosfet to be ON at the same time(shoot-through).
Is the current Dead-time too high and impacting the performance ? Too low and causing shoot-through ?
Based on datasheet of mosfets commonly found on hoverboards, it seems far enough (fall and rise time not considered), even if picking the mosfet with highest turn OFF and the one with lowest turn ON.
Should it also cover the fall and rise time, and how can we estimate the gate driver propagation time ?

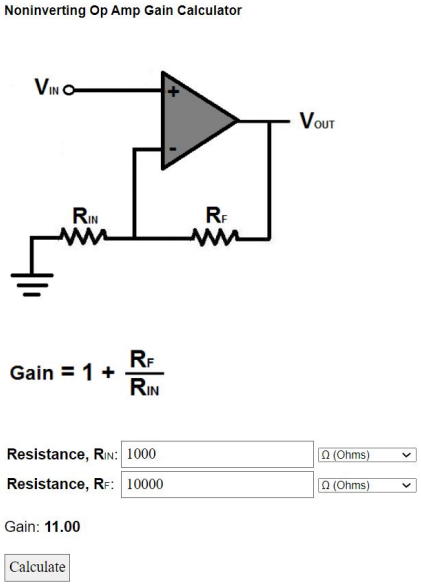
Mosfet	Turn ON delay (ns)	Turn OFF delay (ns)	turn OFF - Turn ON + 20% (ns)
STP110N7F6	23	103	96
P75NF75	25	66	49.2
SIN75Y06	13	29	19.2
100N8F6	33	103	84
IRFB3607	16	43	32.4
SSF7509	17.3	52	41.6

Over-current protection

Current chopping is done by disabling timer's output (BDTR_MOE) if DC current is higher than the max current in Amps, based on the Amp to Bit conversion factor(A2BIT_CONV) that is derived from the shunt resistance, ADC resolution, Reference voltage and Opamp factor.

A2BIT_CONV = 0.0035(shunt resistance) *1A * 11 (FACTOR) * 4096(12 bit ADC) / 3.3v(REF voltage) = 47.7





FOC firmware improvements

FOC - Current Sensing

Could the DC current be used to calculate the 3rd current via this formula ?
Idc = K1*Ia+K2*Ib+K3*Ic with K1,K2 and K3 being the Pwm signals.

Shouldn’t we have to calibrate DC and Phase currents ? The calibration I mentioned in pt2 of the next slide could help.

Update : probably not worth it, the DC current is not precise enough

ADC calibration

1. Use HAL_ADCEx_Calibration_Start for self-calibration could reduce ADC errors(RM0008,ADC accuracy)

11.4 Calibration

The ADC has an built-in self calibration mode. Calibration significantly reduces accuracy errors due to internal capacitor bank variations. During calibration, an error-correction code (digital word) is calculated for each capacitor, and during all subsequent conversions, the error contribution of each capacitor is removed using this code.

2. Current offset calculation is a kind of soft filter, average would work better (beware of overflow)
3. Can VREFINT be used to mitigate reference voltage fluctuation and have a more accurate current measurement? Channel 17 is the ADC value for 1.20V
4. The RDSon of the mosfets on the board might differ, especially if different mosfets were mixed after a replacement.(e.g STP100N8F6 0.008 to 0.009Ω, STP110N7F6 has 0.0055 to 0.0065Ω, IRFB3607 has 0.00734 to 0.009Ω). Would it be possible to run a calibration by powering each phase separately and comparing phase current with DC current as a reference and compensate ? This could also serve as a test to make sure mosfets are not shorted and opening, and help find the correct Phase/hall mapping (autodetect)
5. RDSon varies depending on the temperature of the mosfets. Could the previously mentioned phase current calibration be used to compensate ? When could it be ran ?

Calibration - autodetect/diagnostic

How it could work:

For each motor, and each phase:

- calibrate adc offset for 2000 cycles,
- power the phase at 50% duty cycle for 2000 cycles
- memorize which hall sensor is ON, generate an error if none (the FOC model also checks if none of the halls are ON and generates an error while driving)
- measure DC current, if lower than a threshold:
  - Opamp is either not working or is missing on this board(can happen)
  - Mosfet is not opening ? Gate driver problem ?
- measure Phase current if available, if lower than a threshold
  - Opamp is not working ?
  - Phase measurement of the 2 phases is swapped ?
- If DC current and phase current are available, calculate compensation factor
- Calculate mosfet resistance ? KV ?
- Anti cogging calibration ?
- Spin the motor in open loop using the mechanical angle input, and measure the average angle for each hall position, this will give the correct hall to angle map

In case a mosfet or gate driver is shorted, it should be picked by the ADC watchdog.

This should run on demand (after the input calibration ?), and values should be saved to eeprom

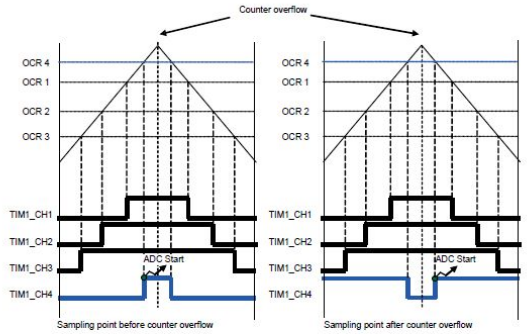
ADC synchronization

Channel 4 of the timers can be used to trigger ADC sampling before the middle of the period, this could help increase the measurement window.

Initialize TIM_CHANNEL_4 from TIM1 with Pulse(ccr) equal to the pwm period / 2 - 1? Or before ? How to be sure it's after hysteresis?

Trigger ADC measurement at OCR4, it works only with Injected ADC (see next slide).

Figure 14. **PWM and ADC Synchronization**



In this way, supposing that the sampling point must be set before the counter overflow, that is, when the TIM counter value matches the OCR4 register value during the up-counting, the A/D conversions for current sam are started. If the sampling point must be set after the counter overflow, the PWM 4 output has to be inverte modifying the CC4P bit in the TIM1_CCER register. Thus, when the TIM1 counter matches the OCR4 regis value during the down-counting, the A/D samplings are started.

After execution of the FOC algorithm, the value to be loaded into the OCR4 register is calculated to set the sampling point for the next PWM period, and the A/D converter is configured to sample the correct channels

Table 3. **Three-shunt current reading, used resources (single drive, F103 LD/MD)**

Adv. timer	DMA	ISR	ADC master	ADC slave	Note
TIM1	DMA1_CH1 DMA1_CH5	None	ADC1	ADC2	DMA is used to enable ADC injected conversion external trigger. Disabling is performed by software.

Table 4. **Three-shunt current reading, used resources (Dual drive,F103 HD, F2x, F4x)**

Adv. timer	DMA	ISR	ADC	Note
TIM1	DMA1_CH1	TIM1_UP	ADC1 ADC2	Used by first or second motor configured in three-sl according to user selection. ADC is used in time sh Trigger selection is performed in the TIM_UP ISR.
TIM8	None	TIM8_UP	ADC1 ADC2	Used by first or second motor configured in three-sl according to user selection. ADC is used in time sh Trigger selection is performed in the TIM_UP ISR.

ADC Mode

Using injected ADC for critical measurements (phase and DC current) would help reduce the total conversion time to only 2.625us.
Values must be read from registers instead of DMA.
Other measurements (ADC input, battery voltage, temperature) can have less priority and be measured less often.
Battery Filter would need to be moved out of bldc code and adjusted.
Measure each wheel separately and reconfigure ? (see next slide)

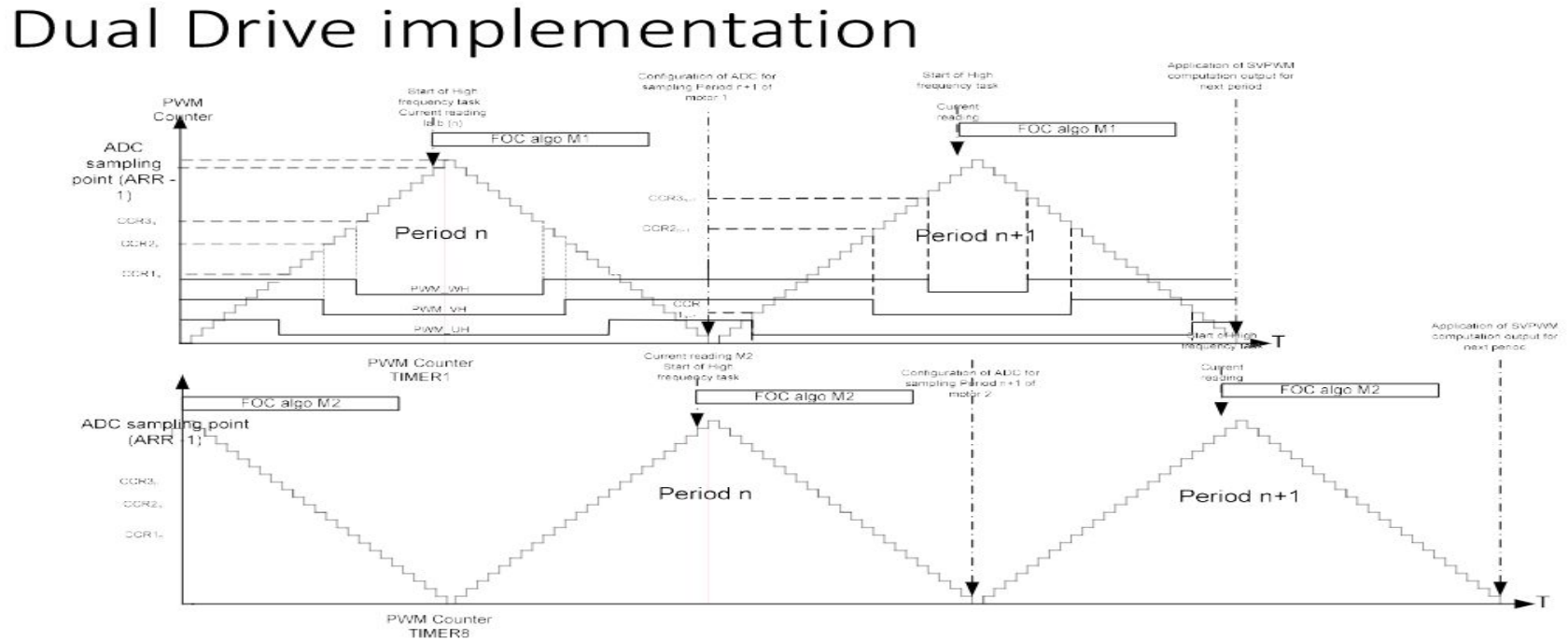
Can we use the 3rd ADC and use 1.5 cycle sampling time be for phase current? That would reduce the needed measurement window to 0,875us.

Do we even have to wait for the conversion ?

	ADC1	ADC2	ADC3
at TIM1 CCR4	Left U(PA0/CH0)	Left V(PC3/CH13)	Left DC(PC0/CH0)
at TIM8 CCR4	Right U(PC4/CH4)	Right V(PC5/CH15)	Right DC(PC1/CH11)

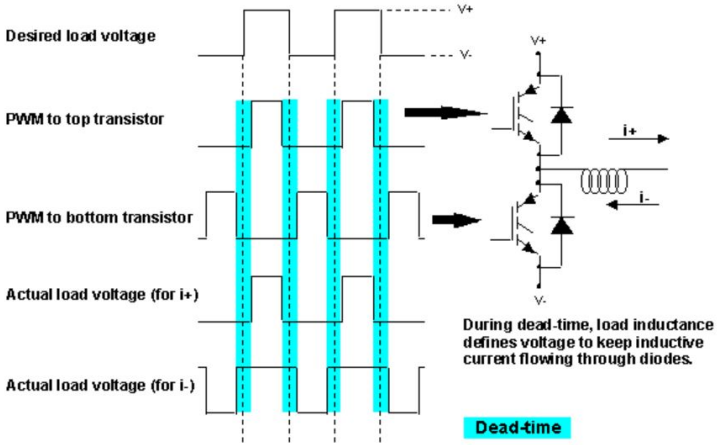
FOC calculation

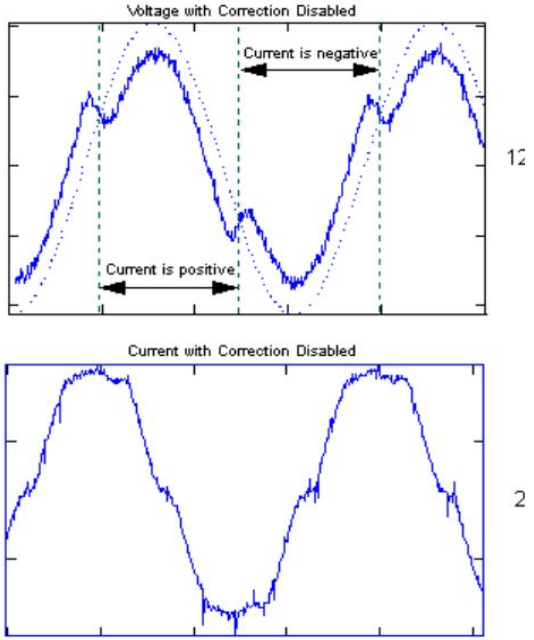
Would it be possible to shift PWMs /current sensing so that the foc model runs for each wheel separately ?
It would take around 23us of the available 30us(half pwm period) for the whole calculation for one wheel.
That would spread the workload across the PWM period, split the needed measurement window for each wheel, and both wheels would have recent current measurement to use.



PWM - dead-time distorsion

Dead-time insertion impacts by reducing the expected ON time for a given duty cycle. This generates dead-time distorsion.
There are techniques to compensate it by increasing the duty cycle artificially to have the right ON time.





A simple solution can be to add +-48 (sign depends on the current) to the duty cycle to compensate the deadtime, with a deadzone close to 0.
This is the value Motor workbench is using to compensate
Can Mosfet On and Off time be measured and deadtime be adjusted ? (example)

After first tests, it's really improving the current curves, but

Over-current protection

Timer's output (BDTR_MOE) is disabled if DC current is higher than the max current in Amps, based on the Amp to Bit conversion factor(A2BIT_CONV) that is derived from the shunt resistance, ADC resolution, Reference voltage and Opamp factor.

A2BIT_CONV = 0.0035(shunt resistance) *1A * 11 (OPAMP FACTOR) * 4095(12 bit ADC) / 3.3v(REF voltage) = 47.7

- There can be different shunt resistors values on different mainboards, replacing A2BIT_CONV parameter's fixed value(50) by the formula could let you adjust the different variables.
- Can ADC_CHANNEL_VREFINT(ADC_CHANNEL_17) be used to improve the measurement? It doesn't seem to be available on GD32 ?
- Don't allow Max current higher than 35A ?
- ADC Watchdog could be used as last protection. I react faster to over-current as It would kick in directly after DC current sampling + conversion(0.875us), if the measured adc value is not within Low and High values. Now it's checked only in the interrupt triggered by the DMA, after all ADC measurements are done. The threshold could be close to the limits like MESC is doing.
- On board where DC current is not available, use the break input instead, it's discussed here

HardFault protection

Timer’s output can be disabled when Hardfault interrupt is triggered for safety

FOC - MTPA

Can we use MTPA to maximize the torque per Amp?
Eg. VESC

Measure Ld-Lq with vesc to see if MTPA is worth investigating.

Some more motor parameters would be require:

- D axis inductance
- Q axis inductance
- Flux linkage