UM3468

User manual

STM32 motor control SDK, 6step startup configuration

Introduction

This document provides guidelines to easily configure the start-up parameters of the 6step sensorless algorithm. For further details about 6step firmware algorithm implemented buy the STM32 Motor Control Software Development Kit (X-CUBE-MCSDK), please refer to the related user manual included in the documentation package.

1 Acronyms and abbreviations

Table 1. Acronyms and abbreviation

Acronym	Description
MCSDK	STM32 Motor Control Software Development Kit (X-CUBE-MCSDK)
HW	Hardware
IDE	Integrated Development Environment
MCU	Micro-Controller Unit
GPIO	General Purpose Input Output
ADC	Analog to Digital Converter
VM	Voltage Mode (driving)
SL	Sensorless (driving)
BEMF	Back Electromotive Force
FW	Firmware
ZC	Zero-Crossing
GUI	Graphical user interface
MC	Motor control
OCP	Over current protection
PID	Proportional-integral-derivative (controller)
SDK	Software development kit
UI	User interface
MC Workbench	Motor control workbench tool, part of MCSDK
MC Profiler	Motor control profiler tool, part of MCSDK
Motor Pilot	Motor Pilot tool, part of MCSDK

2 Overview

In the 6step sensorless driving mode, the firmware exploits the Back Electromotive Force (BEMF) sensed at the floating phase to synchronize power stage driving with rotor position. Since the BEMF signal is proportional to the speed, the rotor position cannot be determined when the motor is stopped or at very low speed. Therefore, the motor must be accelerated in open-loop until a sufficient BEMF voltage, that allows the synchronization of the step commutation with the rotor position.

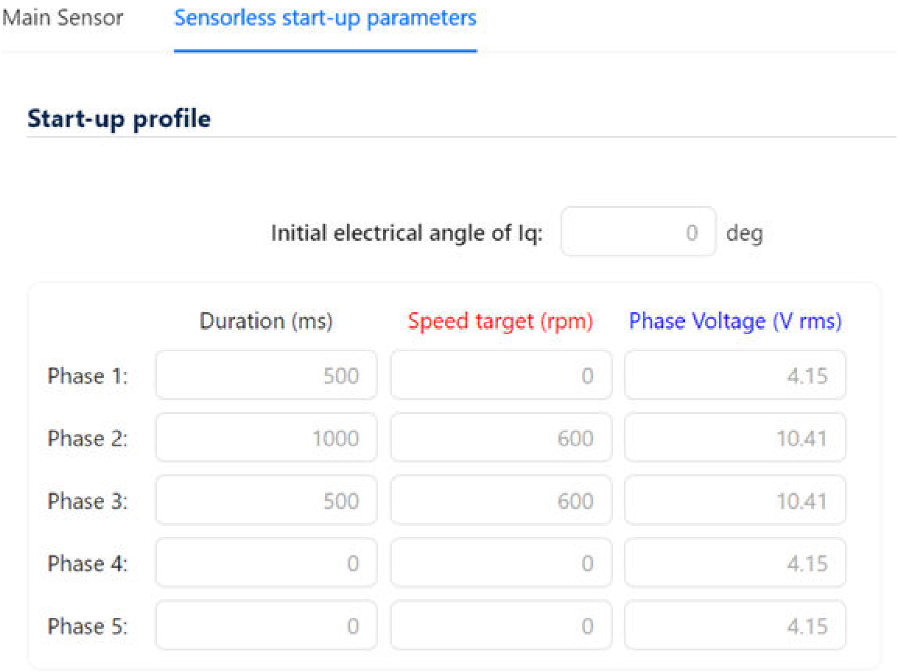
The start-up procedure is usually made up of a sequence of three stages:

1. **Alignment.** The rotor is aligned at a predetermined position.
2. **Open loop acceleration.** The motor leads are in a predetermined sequence to create a magnetic field that will cause the rotor to start rotating. The sequence’s rate is progressively increased to allow the rotor to reach a certain speed.
3. **Switch-over.** Once the rotor has reached a target speed, then tries switching to closed-loop 6step control sequence to maintain control of the motor's speed and direction.

As shown in Figure 1, the user can customize the start-up parameters in the MC Workbench before generating the code according to two different driving modes:

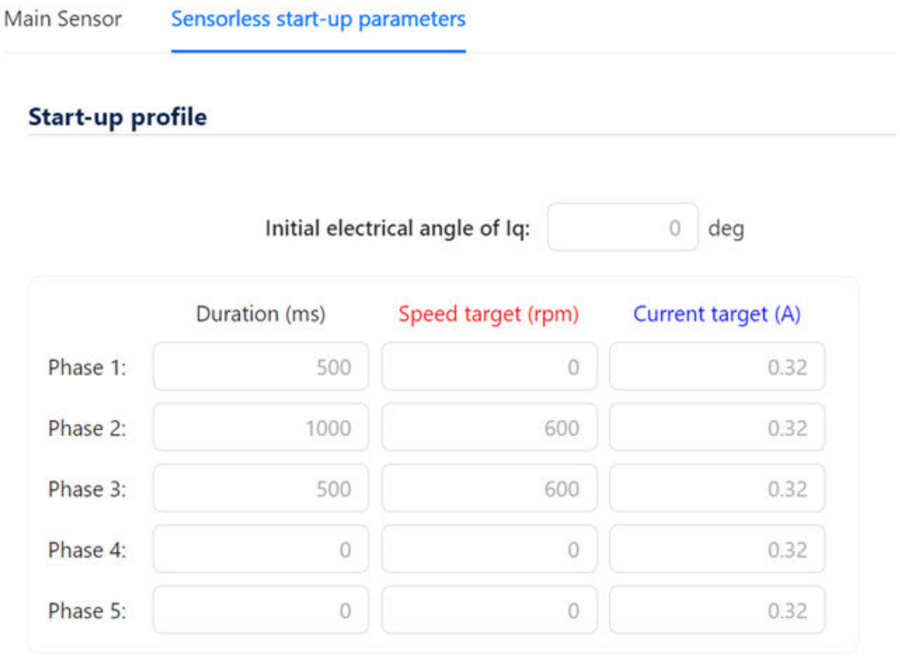
- Voltage mode. The algorithm controls the speed by varying the duty cycle of the PWM applied to the motor phases: a target *Phase Voltage* is defined for each segment of the start-up profile. This mode is activated by default when creating a project with the MC Workbench.
- Current mode. The algorithm controls the speed by varying the threshold of a current limiter circuitry: a *Current target* is defined for each segment of the start-up profile. This mode is activated enabling the option “Current mode” in the “Current sensing” tab of the MC Workbench.

Figure 1. Voltage mode - start-up parameters in the MC Workbench



Initial electrical angle of Iq: 0 deg

	Duration (ms)	Speed target (rpm)	Phase Voltage (V rms)
Phase 1:	500	0	4.15
Phase 2:	1000	600	10.41
Phase 3:	500	600	10.41
Phase 4:	0	0	4.15
Phase 5:	0	0	4.15



Start-up profile

Initial electrical angle of Iq: 0 deg

	Duration (ms)	Speed target (rpm)	Current target (A)
Phase 1:	500	0	0.32
Phase 2:	1000	600	0.32
Phase 3:	500	600	0.32
Phase 4:	0	0	0.32
Phase 5:	0	0	0.32

Figure 2. Current mode - start-up parameters in the MC Workbench

2.1 Alignment

In Figure 1, the *Phase 1* corresponds to the alignment step (i.e. target speed is 0). The rotor is aligned to the 6step position closest to the “Initial electrical angle”.

By default, the duration of the *Phase 1* is 200ms and during this step the phase voltage or the current limiter threshold are linearly increased up to the target value (*Phase Voltage* for Voltage mode and *Phase Current* for Current mode).

2.2 Open loop acceleration

In Figure 1, the steps from *Phase 2* to *Phase 5* correspond to the acceleration phases. The 6step sequence is applied to speed up the motor in open loop, hence, the rotor position is not perfectly synchronized with the 6step sequence and the efficiency is low (the phase currents are then higher than optimum).

The user can define one or more acceleration segments (*Phase 2, 3, 4* and *5*): in particular, in the case of a bulky motor, it is recommended to accelerate it with a slower ramp to overcome the inertia before performing a steeper ramp. During each segment the phase voltage or the current limiter threshold are linearly increased to reach the final target value within the defined time frame. The same applies to the rotation speed.

2.3 Switch-over

At the end of the open loop acceleration, if the debug option “Open loop” is not activated, the switch-over to closed speed loop is performed. During this step the algorithm exploits the sensed BEMF to synchronize the 6step sequence with the rotor position.

After a valid BEMF zero crossing detection signal the algorithm switches to close loop operation.

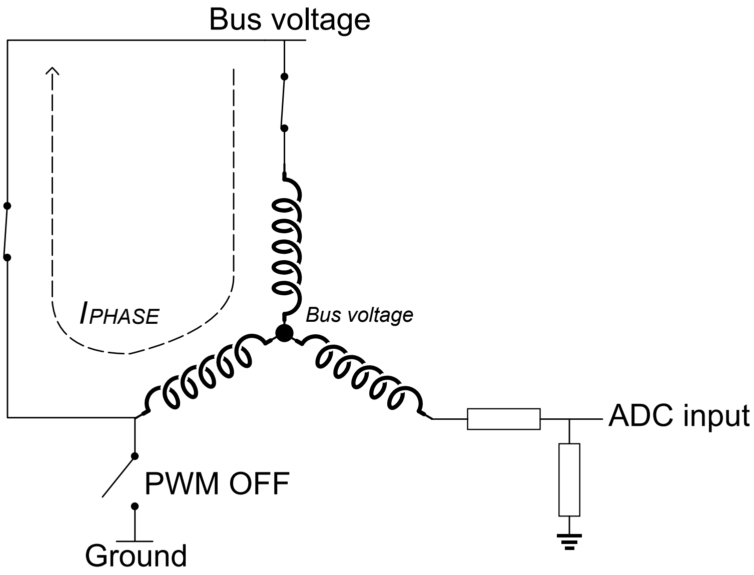
2.3.1 BEMF zero-crossing thresholds

The proper setting of BEMF thresholds is crucial for good performances of the algorithm and successful switch-over operation.

BEMF sample is compared with the zero-crossing thresholds to determine if a BEMF zero-crossing event has occurred. In the MC Workbench, three types of thresholds are defined, based on the sampling point of the BEMF and driving mode of the outputs. The algorithm automatically switches from one threshold to another according to the instantaneous driving conditions.

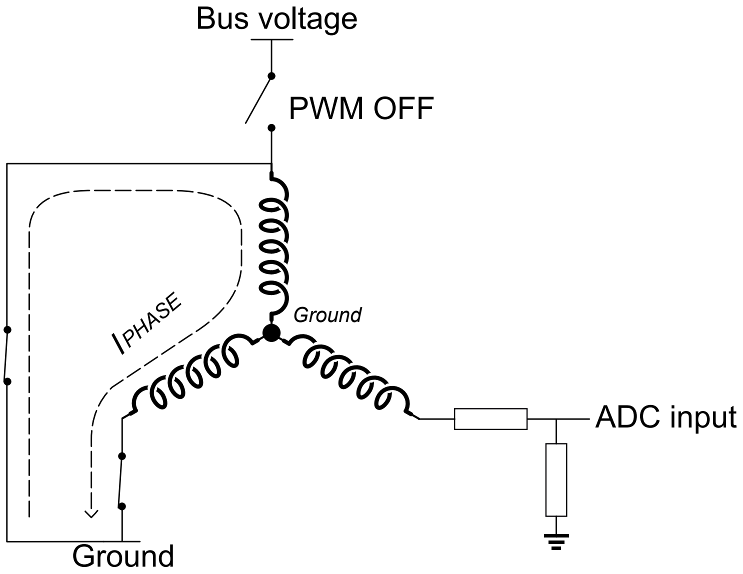
- Threshold with high side recirculation. This threshold is taken into account when the BEMF is sensed during the off time of the PWM and the PWM signal is applied to the low side MOSFETs. In this condition, during the off time of the PWM, only the high side MOSFETs are turned on and the recirculating current pulls the motor center point to the bus voltage. The BEMF threshold, defined as a percentage of the bus voltage, is by default 100%.

Figure 3. **High side recirculation**



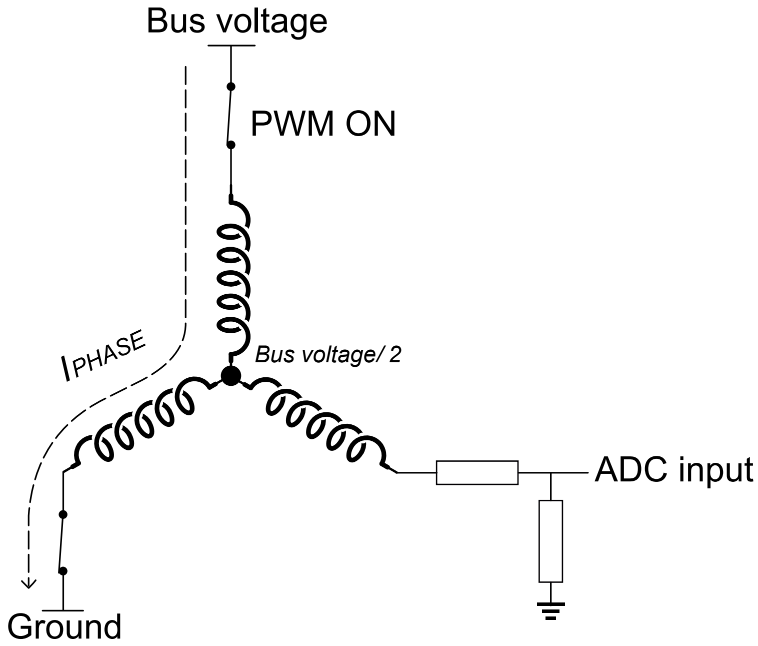
- Threshold with low side recirculation. This threshold is taken into account when the BEMF is sensed during the off time of the PWM and the PWM signal is applied to the high side MOSFETs. In this condition, during the off time of the PWM, only the low side MOSFETs are turned on and the recirculating current pulls the motor center point to ground. The BEMF threshold, defined as a percentage of the bus voltage, is by default 1%.

Figure 4. low side recirculation



- Threshold with PWM on. This threshold is taken into account when the BEMF is sensed during the on time of the PWM. In this condition, the load current flows from one high side MOSFET, through two motor phase into one low side MOSFET. The motor center point would then be set to half the bus voltage. The BEMF threshold, defined as a percentage of the bus voltage, is by default 50%.

Figure 5. low side recirculation

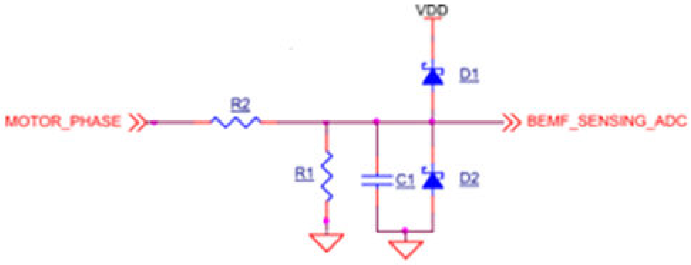


2.4 BEMF sensing network design

In Figure 6 shows the typical circuitry to sense the BEMF. Its purpose is to scale the motor phase voltage to be properly acquired by the ADC.

D1 and D2 diodes are optional and should be added only in case of risk of violating the BEMF sensing ADC channel maximum ratings.

Figure 6. BEMF sensing network



R2 and R1 value should be chosen according to the bus voltage level. User should be aware that implementing a R1 / (R2 + R1) ratio much lower than needed, the BEMF signal may result in too low and the control not robust enough.

On the other hand, a ratio higher than needed would lead in frequent turning-on/off of the D1 protection diode whose recovery current may inject noise. The recommended value is:

$$\frac{R1}{R1+R2} = \frac{0.95 \times V_{DD}}{V_{BUS}} \tag{1}$$

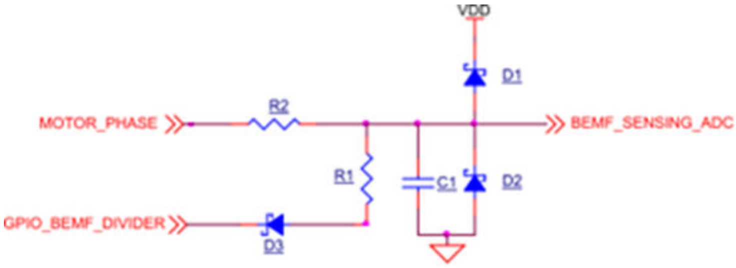
In this way the full motor supply voltage (V_{BUS}) can be mapped on the full-scale range of the ADC (V_{DD}).

Very low values for R1 and R2 should be avoided to limit current tapped from the motor phase.

C1 is for filtering purposes and should not limit the signal bandwidth in the PWM frequency range; the recommended value is in the range of 47 pF to 220 pF.

It is possible to connect R1 to a GPIO instead of GND, allowing the network to be run-time enabled or disabled (see Figure 7).

Figure 7. BEMF sensing network – GPIO variant

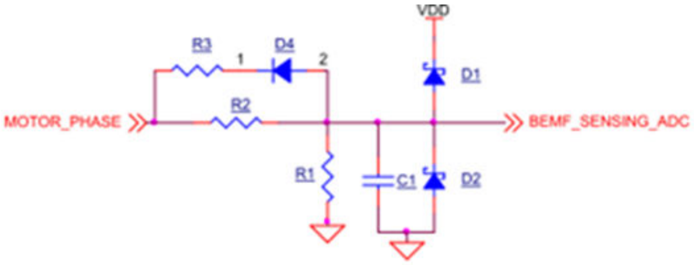


In this case the 6step firmware keeps the divider enabled forcing the GPIO in reset state. However, the presence of D3 must be considered when setting the BEMF thresholds for sensing during the PWM on time: it usually adds 0.5 V to 0.7V to the ideal threshold. This BEMF network schematic is deprecated and not recommended for new designs.

$$BEMF\ threshold\ (ON) = V_F\left(D3\right) + \frac{V_{BUS}}{2} \tag{2}$$

In high voltage inverters, an additional circuitry composed by D4 and R3 is recommended for a fast discharge of BEMF_SENSING_ADC node during PWM commutations (see Figure 8).

Figure 8. **BEMF sensing network – high voltage variant**



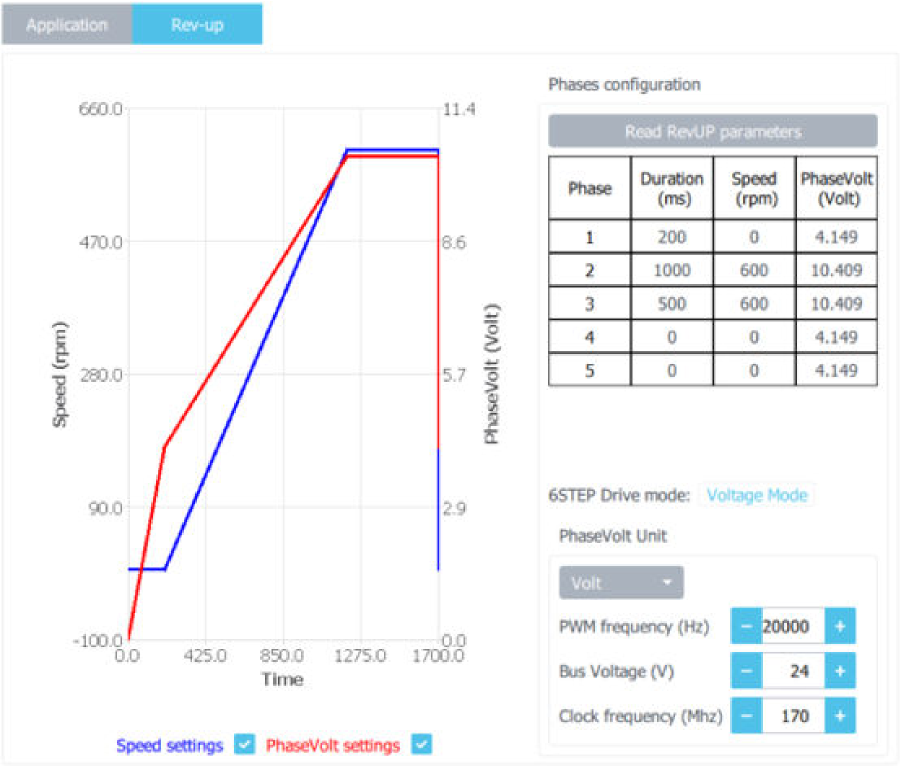
2.5 Tuning the parameters using the motor pilot

The values applied during the motor start-up can be customized in the Motor Pilot GUI, rev-up section, without the need to regenerate the code.

In the table shown in Figure 9, the user can change the target voltage/current, the target speed and the duration of each phase.

Once the user finds the proper values that suit best his motor, they can be feedbacked into the MC Workbench project to regenerate the code and apply this default value.

Figure 9. **Start-up parameters in the motor pilot**



Phase	Duration (ms)	Speed (rpm)	PhaseVolt (Volt)
1	200	0	4.149
2	1000	600	10.409
3	500	600	10.409
4	0	0	4.149
5	0	0	4.149

6STEP Drive mode: Voltage Mode

PhaseVolt Unit: Volt

PWM frequency (Hz) − 20000 +

Bus Voltage (V) − 24 +

Clock frequency (Mhz) − 170 +

3 Optimization of the start-up

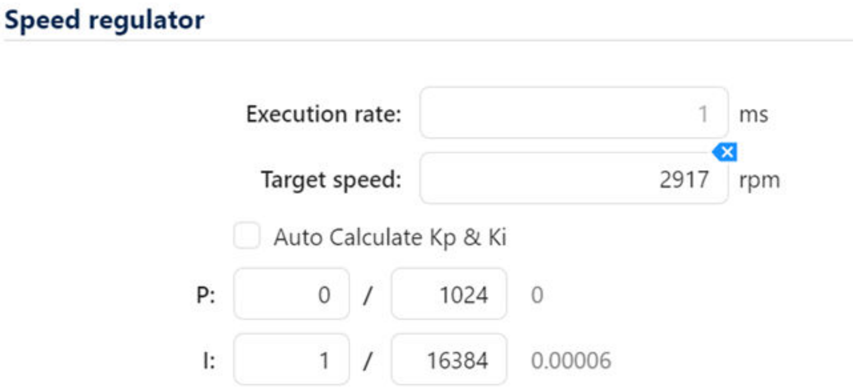
The following procedure is meant to help the user in finding the right values of the target speed and the target voltage/current to perform a smooth and successful sensorless motor start-up.

3.1 Preliminary steps

To optimize the start-up process, it is recommended to follow the steps below:

1. In the “Drive Settings” section of the Workbench, set speed KP gain (proportional gain of the speed loop) to 0 and speed KI gain (integral gain of the speed loop) to 1 (see Figure 10).

Figure 10. **PI gains**



Speed regulator

Execution rate: 1 ms

Target speed: 2917 rpm

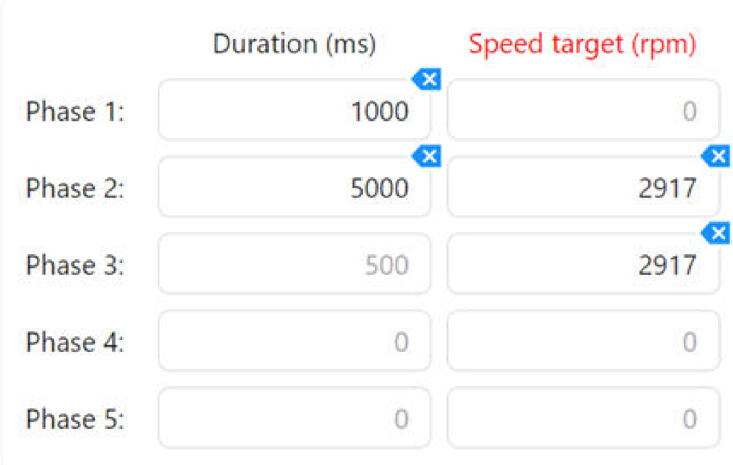
☐ Auto Calculate Kp & Ki

P: 0 / 1024 0

I: 1 / 16384 0.00006

2. The first two phases of the rev-up procedure, as described in Section 2.1 and Section 2.2, are the alignment and the acceleration step respectively. In the “Speed Sensing Config.” section of the Workbench, set the alignment duration to 1 second and the acceleration step to 5 seconds (see Figure 11).

Figure 11. **Start-up profile configuration**



	Duration (ms)	Speed target (rpm)
Phase 1:	1000	0
Phase 2:	5000	2917
Phase 3:	500	2917
Phase 4:	0	0
Phase 5:	0	0

3. While the target speed of the alignment step is imposed to 0 RPM, the target speed of the acceleration step should be chosen in accordance with the BEMF constant of the motor and the BEMF network divider of the board. This is to ensure that at the end of the acceleration the BEMF signal is strong enough to allow a reliable zero-crossing detection and the rotor synchronization. It is advisable that:

   $$Target_{RPM} = \frac{\sqrt{2} \times BEMF_{ADC}(@maxSpeed) \times BEMFdivider}{K_{BEMF}} \tag{3}$$

   Where:
   $BEMF_{ADC}$(@maxSpeed) = maximum BEMF at the ADC input at the end of the acceleration step in mV
   BEMFdivider = BEMF divider factor implemented on the board
   K_{BEMF} = BEMF constant in Vrms / kRPM
   A *$BEMF_{ADC}$(@maxSpeed)* value of 150 mV to 200 mV is usually enough to ensure a reliable zero crossing detection.

Note: the above considerations don't relieve the user of properly sizing the BEMF network as explained in the Section 2.4.

   As an example, in Figure 11, with *$BEMF_{ADC}$(@maxSpeed)* = 150 mV, the target speed for the BullRunning motor (K_{BEMF} = 0.4 *Vrms / kRPM*) with X-NUCLEO-IHM07M1 (BEMFdivider = 5.5) is set to 2917 RPM.

4. Keep the Phase Voltage or Phase Current values (according to voltage/current driving mode selection) as pre-calculated by the MC Workbench.
5. Generate the code through the MC Workbench, compile and download it into the MCU. The next steps of the optimization procedure are managed through the Motor Pilot, in the tab dedicated to the “Rev-up”.
6. If voltage mode driving is applied follow the procedure in the Section 3.2. If current mode is selected, skip to the Section 3.3.

3.2 Voltage mode

3.2.1 Definition of the target of the alignment phase

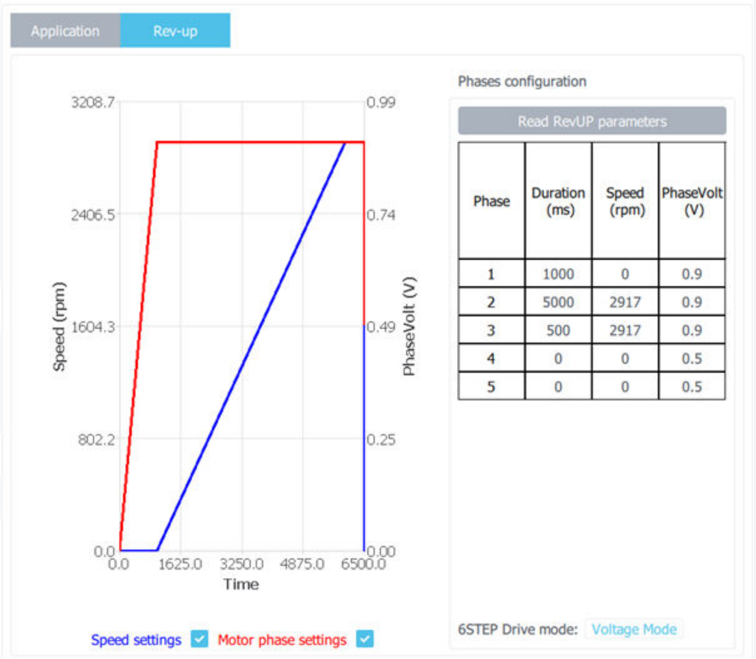
To optimize the alignment phase in voltage mode driving:

1. Set the phase voltage of all the rev-up steps at a low value. A good starting point is usually 20% of the expected BEMF level at the final target speed (Figure 12).

$$PhaseVoltage_{INIT}[Vrms] = 0.2 \times BusVoltage[V] \times \sqrt{\frac{\sqrt{2} \times targetSpeed[RPM] \times K_{BEMF}[\frac{Vrms}{kRPM}]}{1000 \times BusVoltage\,[V]}} \quad (4)$$

   The motor is expected to stall and not complete the procedure.
2. Progressively increase this value for all the rev-up steps until the motor starts spinning. If so, even if the motor stalls before completing the acceleration, the alignment is considered successful and skip to Section 3.3.
3. .If an overcurrent event occurs and motor has not even started spinning, the target speed is not feasible with this set-up. Return to step 4 of the preliminary steps section (3.1) and recalculate the target speed with a smaller *BEMF*$_{ADC}$*(@maxSpeed)* value or consider to increase the BEMF signal at the ADC input changing the BEMF divider value.

Figure 12. Alignment phase voltage settings



Phase	Duration (ms)	Speed (rpm)	PhaseVolt (V)
1	1000	0	0.9
2	5000	2917	0.9
3	500	2917	0.9
4	0	0	0.5
5	0	0	0.5

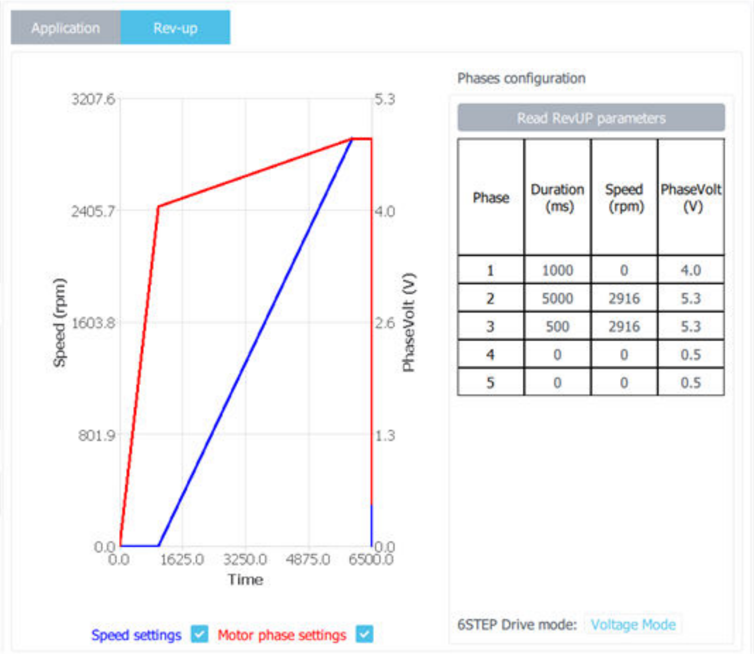
6STEP Drive mode: Voltage Mode

3.2.2 Definition of the target the acceleration phase

To optimize the acceleration phase in voltage mode driving:

1. Without changing anymore the alignment step, carry on increasing the phase voltage of the acceleration steps until the motor properly reaches the RUN state.

Figure 13. Acceleration phase voltage settings



Phases configuration

Phase	Duration (ms)	Speed (rpm)	PhaseVolt (V)
1	1000	0	4.0
2	5000	2916	5.3
3	500	2916	5.3
4	0	0	0.5
5	0	0	0.5

6STEP Drive mode: Voltage Mode

2. At this point, the timings can be reduced and the speed loop controller gains can be increased and optimized.
3. If an overcurrent event occurs before the motor reaches the RUN state, the target speed is not feasible with this set-up. Return to step 4 of the preliminary steps section (Section 3.1) and recalculate the target speed with a smaller *$BEMF_{ADC}$(@maxSpeed)* value or consider to increase the BEMF signal at the ADC input changing the BEMF divider value.

3.3 Current mode

To optimize the alignment phase in current mode driving:

1. Set the phase current reference of all the rev-up steps at a low value. A good starting point is usually 10% of the rating of the motor (Figure 12).

$$PhaseCurrent_{INIT}[Arms] = 0.1 \times MotorCurrentRating[A] \qquad (5)$$

The motor is expected to stall and not complete the procedure.

2. Progressively increase this value until the motor starts spinning and accelerates.

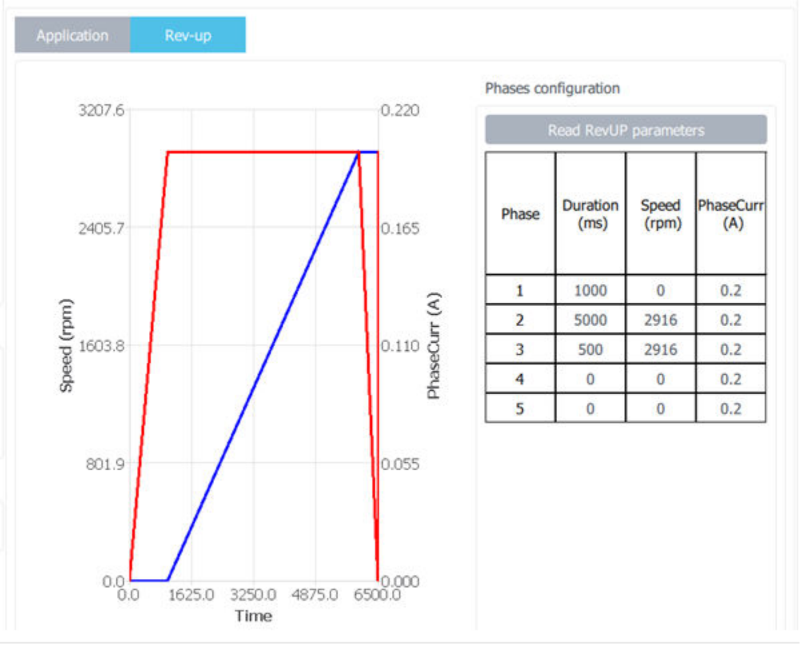
3. Carry on increasing the phase voltage of the acceleration steps until the motor properly reaches the RUN state

4. At this point, the timings can be reduced and the speed loop controller gains can be increased and optimized.

5. If an overcurrent event occurs before the motor reaches the RUN state, the target speed is not feasible with this set-up.

Return to step 4 of the preliminary steps section (Section 3.1) and recalculate the target speed with a smaller BEMFADC(@maxSpeed) value or consider to increase the BEMF signal at the ADC input changing the BEMF divider value.

Figure 14. Phase current settings



Phase	Duration (ms)	Speed (rpm)	PhaseCurr (A)
1	1000	0	0.2
2	5000	2916	0.2
3	500	2916	0.2
4	0	0	0.2
5	0	0	0.2

Revision history

Table 2. Document revision history

Date	Version	Changes
16-Jan-2026	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 STMicroelectronics – All rights reserved