BLDC motor control software using XMC

XMC1000, XMC4000

About this document

Scope and purpose

Brushless Direct Current (BLDC) motors are used in a diverse range of industries including appliance manufacturing, automotive, aerospace, consumer, medical, industrial automation equipment and instrumentation. This is largely because of their compact size, controllability and high efficiency. BLDC motors do not use brushes for commutation, but are electronically commutated instead.

This application notes describes the implementation of the BLDC Motor Control Software using the Infineon XMC1302 microcontroller. Features such as various control schemes, adaptive Hall pattern learning and motor parameter configuration are provided in the software.

Intended audience

This document is intended for customers who would like a highly configurable system for scalar control and to select between sensor-based control using Hall sensors or sensor-less control on XMC series microcontrollers.

References

[1] The User's Manual can be downloaded from **http://www.infineon.com/XMC**.

Table of contents

1 BLDC overview

Scalar control of 3-phase BLDC motors is an electronic commutation scheme, commonly known as trapezoidal commutation, or 120^0 commutation. In this control scheme, each phase conducts for 120^0 during the positive and negative half of a Back-EMF cycle, and is off or un-energized for the remainder of the cycle.

In terms of phase to phase conduction, each phase-pair conducts in steps of 60^0 electric degrees. A 3-phase BLDC motor is synchronous, therefore to produce the maximum torque for the applied stator current the stator magnetic fields must rotate in synchronism with the rotor, and its orientation should be in space quadrature to rotor magnetic field. To achieve these objectives, the trapezoidal control algorithm requires rotor position feedback for every 60^0 (electric degrees). Based on the rotor position feedback mechanism, trapezoidal commutation is characterized as:

- Sensor-based commutation in which Hall- sensor provides rotor position feedback.
- Sensor-less commutation scheme which derives the rotor position based on Back-EMF sensing of the un-energized phase.

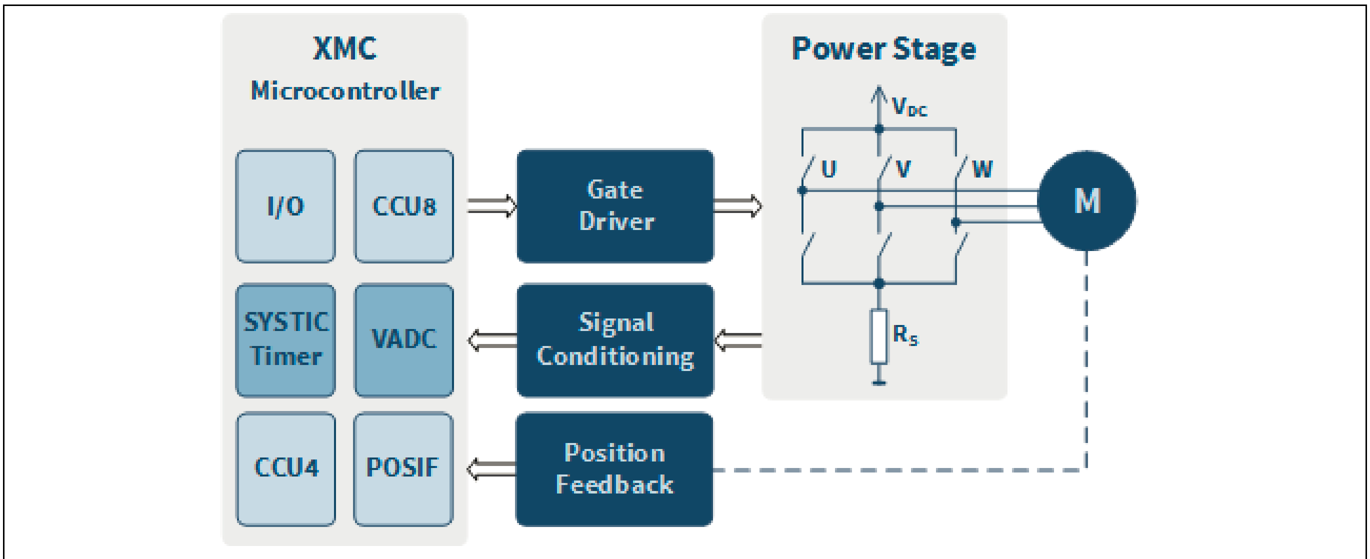


Figure 1 Overview block diagram of BLDC motor control

The key difference in the implementation between sensor-based and sensor-less motor control is the method to determine rotor position feedback mechanism.

Table 1 Method to determine the rotor position in BLDC motor control

	Sensor-based	Sensor-less
Motor Type	BLDC	-
Commutation	Block commutation with PWM	-
Current Measurement	Average current measurement / Direct DC link current measurement	-
Position Detection	3 Hall/ 2 Hall sensors	• Back-EMF detection at unpowered phase. • Detection of zero crossing at 30° using ADC.
Control Scheme	Hall sensors dictate the phase switching	• Zero crossing with 30° delay dictates the phase switching

1.1 Key features

Table 2 XMC BLDC motor control software features

Feature	Description	Sensor-based	Sensor-less
Control schemes	Allows the motor to be controlled in different operating schemes.	✓	✓
PWM modulation schemes	Selects the PWM modulation scheme to be used.	✓	✓
Seamless bi-directional control	Allows the motor direction to rotate in the reverse direction without stopping the motor first.	✓	-
Catch free-running motor	Used to catch the spinning motor at start up from existing speed without stopping the motor.	✓	✓
Hall pattern learning	Used to learn a new Hall-pattern from the motor during start-up.	✓	-
Protection	Protects the hardware against operating limits.	✓	✓
Low speed measurement	Enables the software to detect very low motor speed. Detection is achieved by using the floating prescaler feature in the XMC CCU4 peripheral module.	✓	✓
Enhanced current measurement	Improves the current measurement accuracy.	✓	✓
DC Bus voltage clamping	Limits the DC bus voltage during a sudden stop in a fast braking situation.	✓	✓

1.2 Peripherals

The following table lists the functions of the XMC microcontroller peripherals:

Table 3 Functions of peripherals

Peripherals	Description	Sensor-based	Sensor-less
Capture and Compare Unit 8 (CCU80)	Gives the microcontroller the capability to generate PWM signals and provides the proper time base for the sampling and controlling time base.	✓	✓
Capture and Compare Unit 4 (CCU40)	Captures the speed of the motor based on the commutation update pattern.	✓	✓
	Provides the time base for the generation of precise delay functions utilized for the POSIF0 peripheral.	✓	-
Position Interface (POSIF0)	Provides the commutation pattern update signal to the CCU80 peripheral.	✓	✓
	Decodes Hall sensor inputs.	✓	-
Versatile Analog-to-Digital Converter (VADC0)	Provides the analogue to digital conversion capabilities for analogue variables acquisition. Results are used in the protection schemes.	✓	✓
	Provides the limit checking with channel event generation for the out of boundary condition, used to detect BEMF zero crossing detection.	-	✓

1.3 Supported devices

The devices supported by the BLDC Motor Control software are described in our next table:

Table 4 XMC BLDC motor control software, supported devices

Software	Description	XMC1302	XMC4400
BLDC Scalar Control with Hall Sensors	Developed for 3-phase BLDC motor control using a 120 degree block commutation scheme. The commutation scheme derives the rotor position feedback for every 60 degree, provided by 3-Hall sensors.	✓	✓
Sensorless BLDC Scalar Control	Developed for 3-phase BLDC motor control using a 120 degree block commutation scheme. The commutation scheme derives the rotor position based on Back-EMF sensing of the un-energized phase using ADC zero-crossing detection.	✓	-

1.4 Limitations

1.4.1 Sensor-based control scope of use

In this application note, the software version used is BLDC SCALAR HALL v1.0.1.

At the time of release of this example software the following limitations apply:

- Only a single motor drive is supported.
  - Dual motor control support is not available.
- Linear ramp function is supported.
  - S curve ramp function is not available.
- Adaptive Hall pattern supports 3-Hall sensors placed at a relative angle of 120 degree electrical from each other and their transitions are aligned to zero crossing of phase to phase Back-EMFs. This represents the majority case for motor applications with Hall sensors. Hall pattern detection does not work in the following case:
  - 60 degree placed Hall sensor aligned to phase to phase zero crossing.
  - 120/60 degree placed Hall sensor aligned with phase to neutral zero crossing.

1.4.2 Sensor-less control scope of use

In this application note, the current software version used is BLDC SCALAR SL v1.0.0.

At the time of release of this example software the following limitations in use apply:

- XMC4000 devices are not supported in this software version.
- Only a single motor drive is supported.
  - Dual motor control support is not available.
- Linear ramp function is supported.
  - S curve ramp function is not available.

2 BLDC motor control software components

The major components of the BLDC motor control software are depicted in the following diagram. We will describe each of the modules referenced.

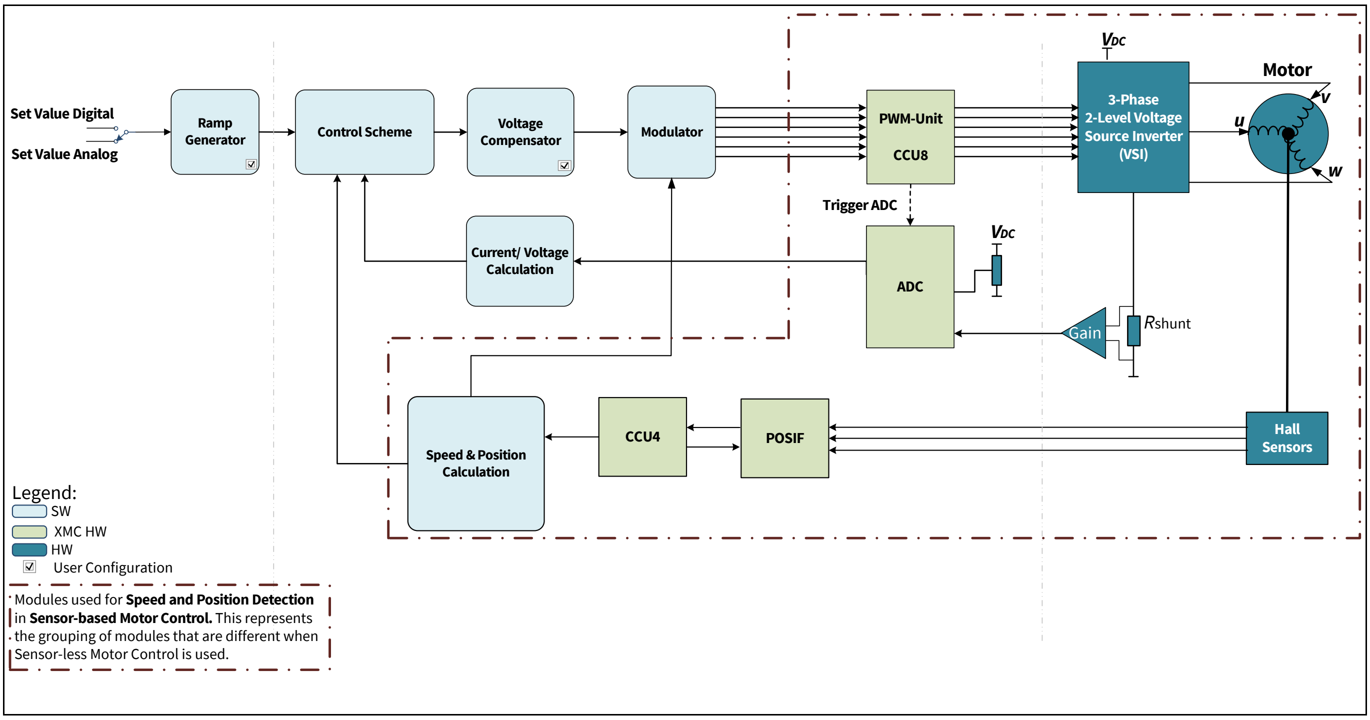


Figure 2 Sensor-based block diagram of BLDC motor control software

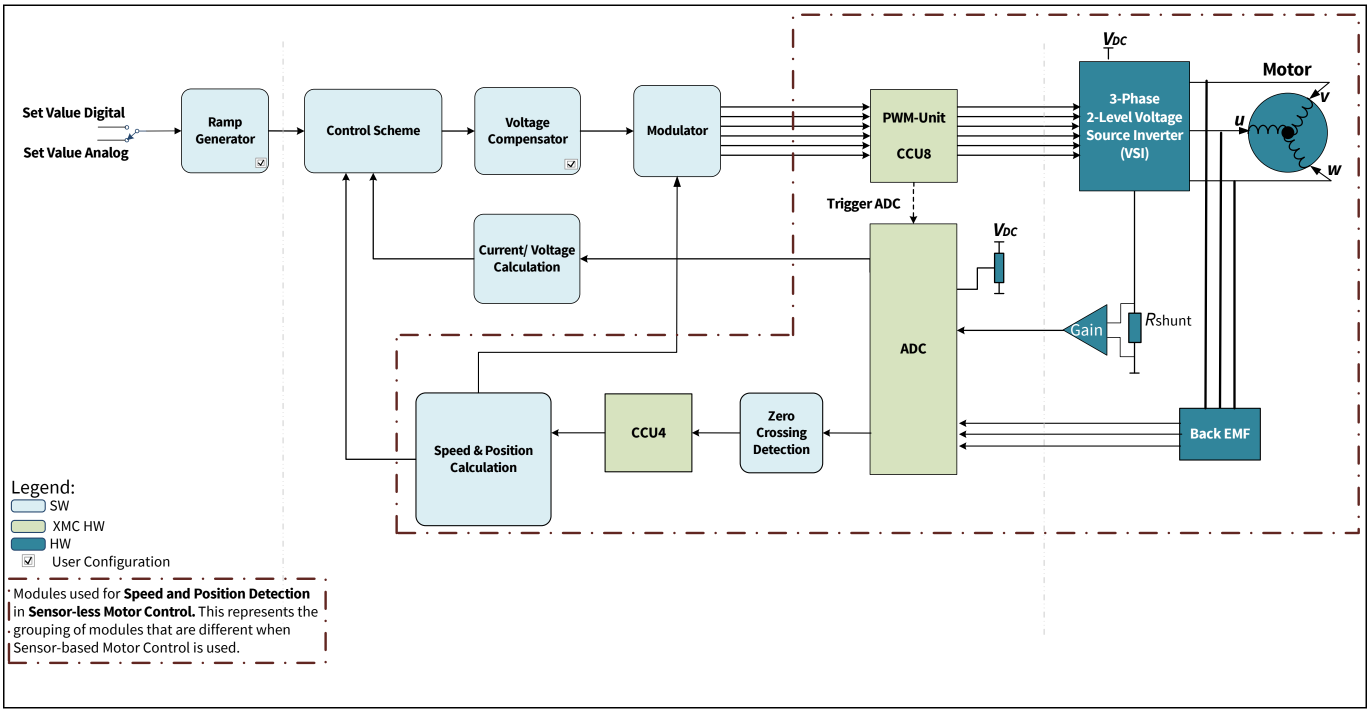


Figure 3 Sensor-less block diagram of BLDC motor control software

2.1 Control schemes

In this software block, the control schemes for the control of the 3-Phase BLDC motor can be either voltage, speed, current (torque), or speed with a current (torque) control scheme.

2.1.1 Voltage control

The voltage control scheme provides behavior comparable to a brushed DC motor. Although the position of the stator field is controlled by Hall sensors to be synchronous with the rotor, the speed and torque depends on the construction of the individual motor and the mechanical load. Voltage set point input can be connected to the ramp output (if ramp is enabled) or a user set value/ analog input.

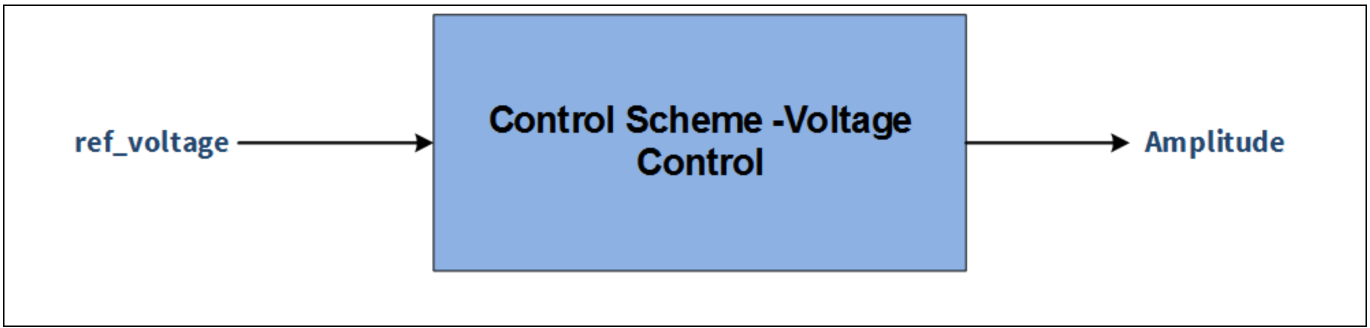


Figure 4 Voltage control block diagram

2.1.2 Speed control

A speed control scheme is a closed loop control, which adjusts the voltage according to the speed reference value. In case of dynamic load changes, the voltage at the motor is adjusted automatically and the speed is maintained at a constant.

The actual speed value is derived from the Hall sensor/phase voltage. Speed set point input will connect to ramp output (if ramp is enabled) or to a user set value or analog input, based on configuration.

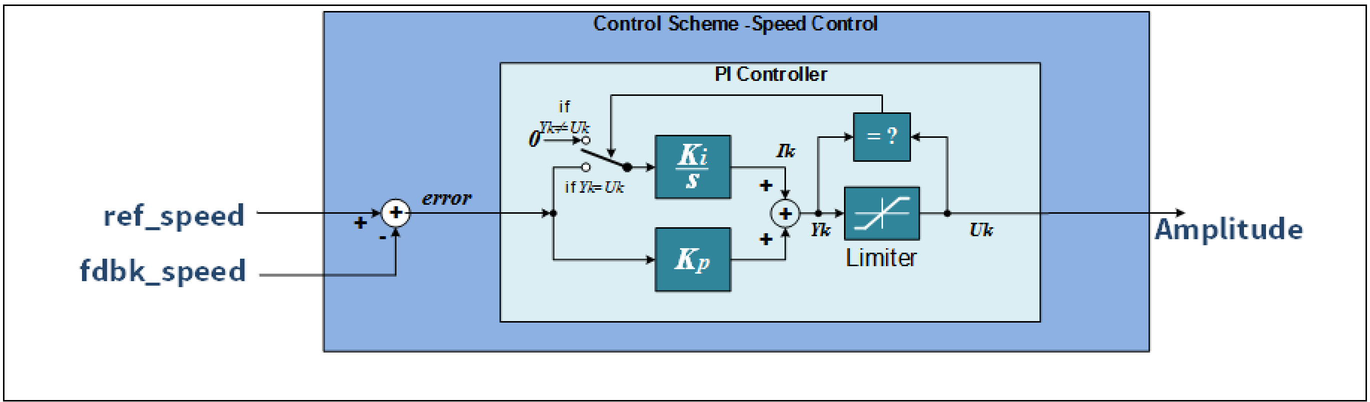


Figure 5 Speed control block diagram

2.1.3 Current control

The current control scheme requires a current measurement feedback and adjusts the voltage according to the required torque. With dynamic loads, the speed will vary, but the torque will remain constant. The current set point input will connect to ramp output (if ramp is enabled) or to a user set value or analog input, dependent on configuration.

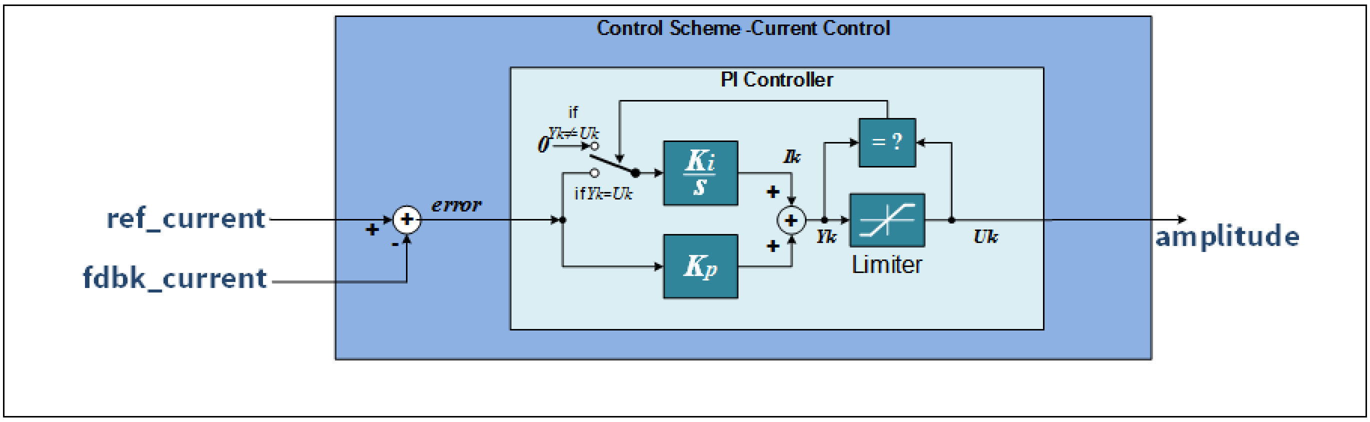


Figure 6 Current control block diagram

2.1.4 Speed with current control

A speed with torque control scheme provides a cascaded control scheme, where the inner control loop adjusts the current (torque) by changing the voltage at the motor and the outer control loop provides the current reference value in order to control the speed. Speed set point input will connect to ramp output (if ramp is enabled) or to a user set value or analog input, depending on configuration.

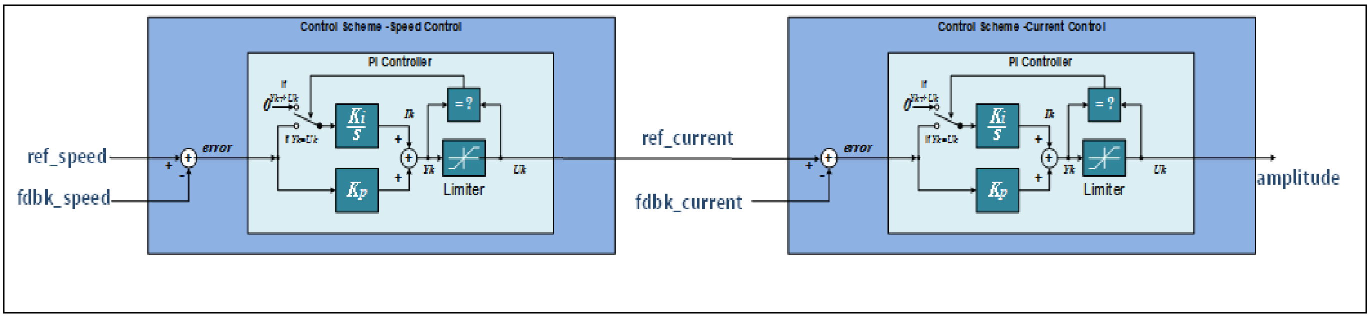


Figure 7 Speed with current control block diagram

2.2 Voltage compensator

DC link voltage is measured every PWM period and compensates the variation in the DC bus voltage. An increase or decrease is applied to the voltage based on the actual DC link voltage and the configured DC link voltage, so that the voltage applied to the motor will be maintained even when there is a variation in DC link voltage. A PT1 filter is used to attenuate high frequency noises.

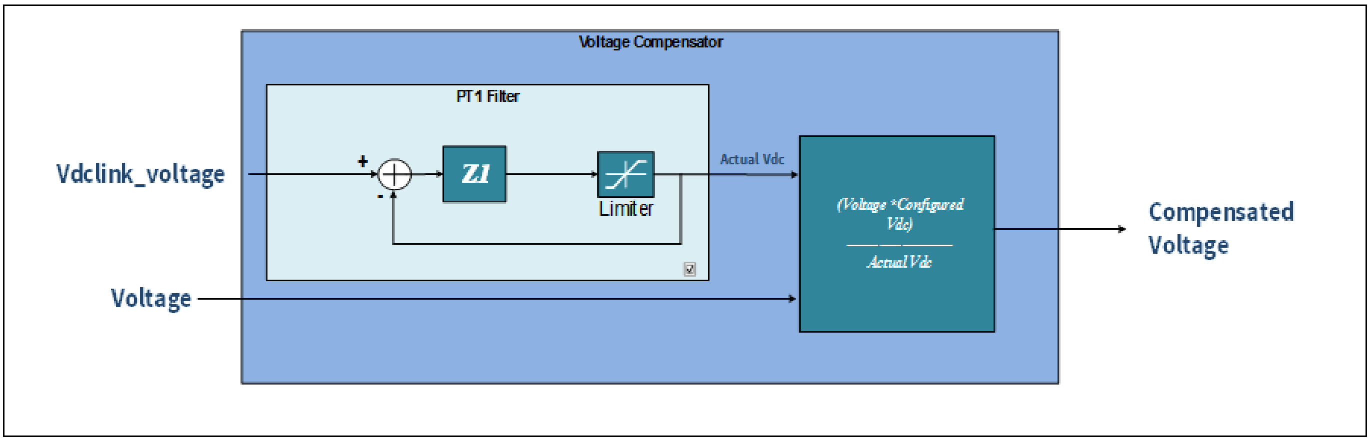


Figure 8 Voltage compensator block diagram

2.3 PWM generator

In the BLDC motor control software, the PWM updates the commutation pattern and controls the modulation scheme used.

2.3.1 PWM pattern update

The PWM pattern update is made using the POSIF peripheral.

In the software configured to support a sensor-based solution, the pattern update is made using the POSIF peripheral configured in Hall Sensor Control with Multi-Channel Mode. For a sensor-less solution, the PWM pattern update is made using the POSIF peripheral configured in Stand-Alone Multi-Channel Mode.

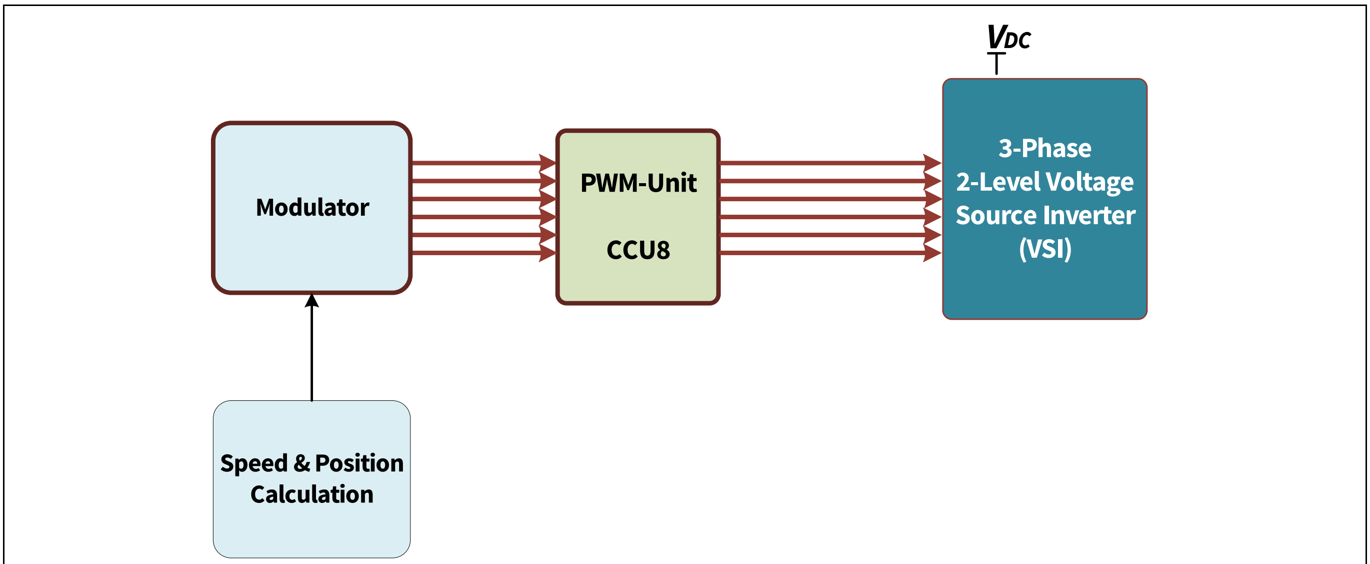


Figure 9 PWM generation module block diagram

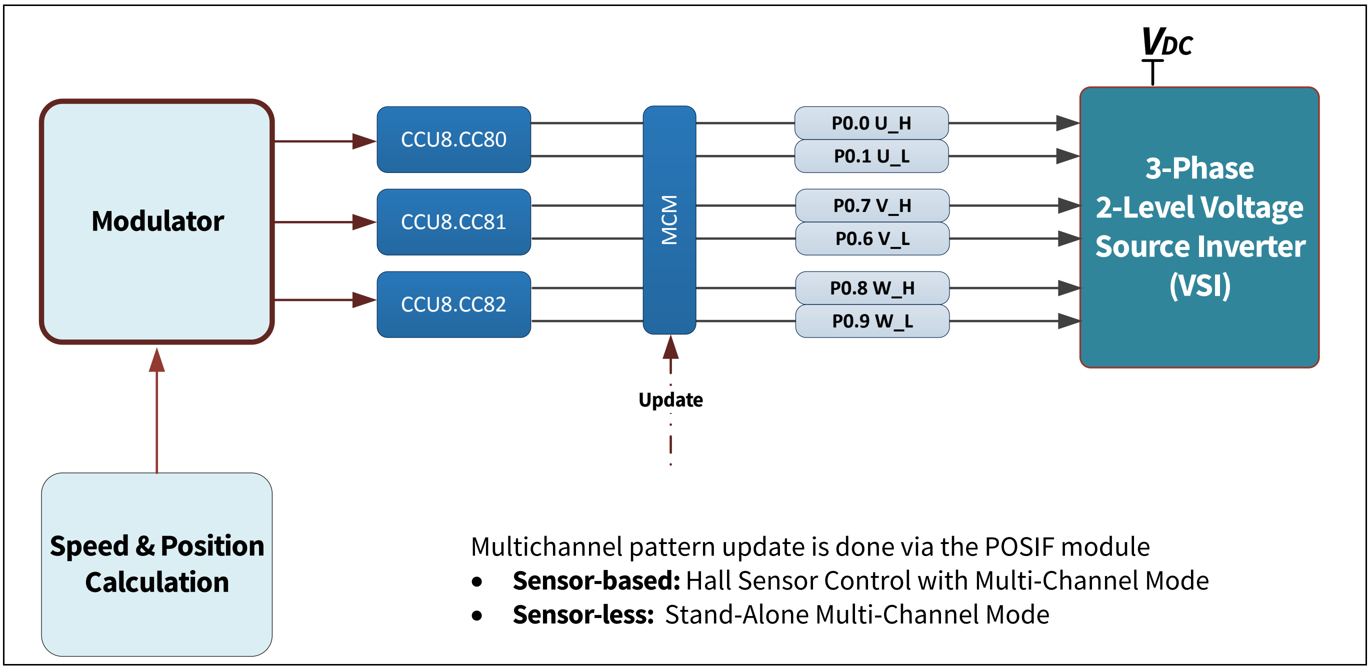


Figure 10 Hardware block interconnect for PWM generation module

Table 5 PWM generation resource description

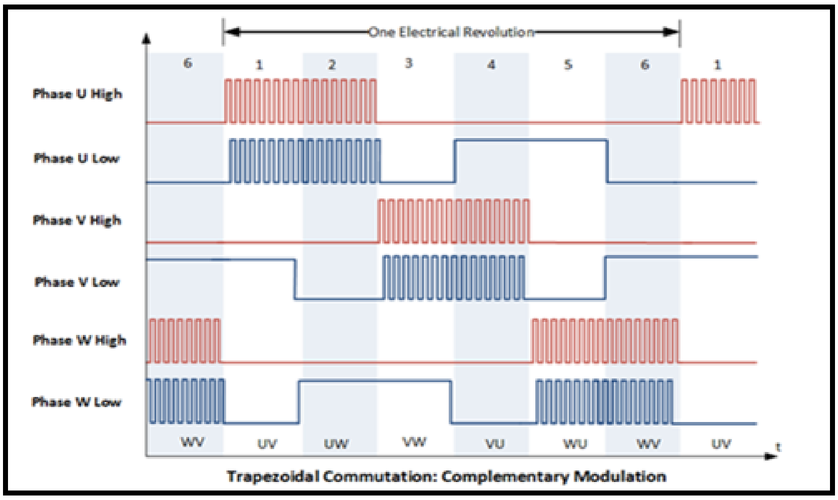
Resource	Description	Connections
POSIF0	PWM commutation pattern with the POSIF peripheral configured in: • Hall Sensor Control with Multi-Channel Mode for sensor-based solution. • Stand-Alone Multi-Channel Mode for sensor-less solution.	-
CCU80.80	Complementary PWM output for U-Phase.	-
CCU80.81	Complementary PWM output for V-Phase.	-
CCU80.82	Complementary PWM output for W-Phase.	-

2.3.2 PWM scheme

- The rotation of the motor depends on the commutation sequence.
  - A commutation sequence in a correct order ensures the proper rotation of the motor.
- Motor speed depends upon the amplitude of the applied voltage.
- The amplitude of the applied signal is adjusted by using Pulse Width Modulation (PWM).

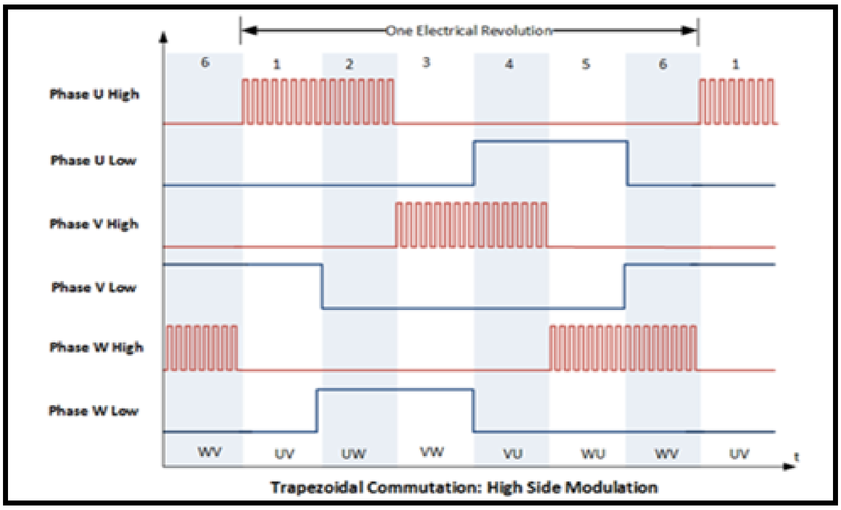
Table 6 Supported PWM schemes

Modulation scheme	Description
High Side Modulation	Modulation is applied to high side switches.
Low Side Modulation	Modulation is applied to low side switches.
High Side Modulation with Synchronous Rectification	Modulation is applied to high side switches with a complementary PWM on the low side switches. This helps to reduce diode losses.

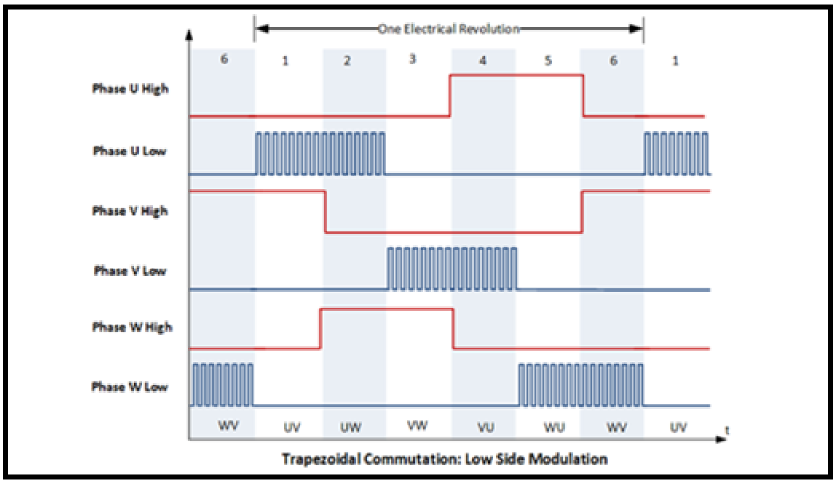


High side sync rectification PWM modulation

+++-



High side PWM modulation



Low side PWM modulation

Figure 11 Supported PWM schemes

2.4 Current and voltage calculation

In the BLDC motor control software this module is used to measure motor current and DC link voltage using the VADC module. The measurement can be triggered based on a software or hardware trigger. The current measurements are either an average or a direct measurement.

Direct current measurements are synchronized with the PWM. The ADC trigger is configurable and is based on the Channel 2 compare match value for Phase-V PWM.

Average current value is calculated in the software, using the PT1 filter and duty cycle value. The calculations are based on configuration settings. The current amplifier offset will be calculated during start-up for average and direct DC link current measurement.

Note: In XMC1300 devices, the ADC on-chip amplifier can be used for current measurement.

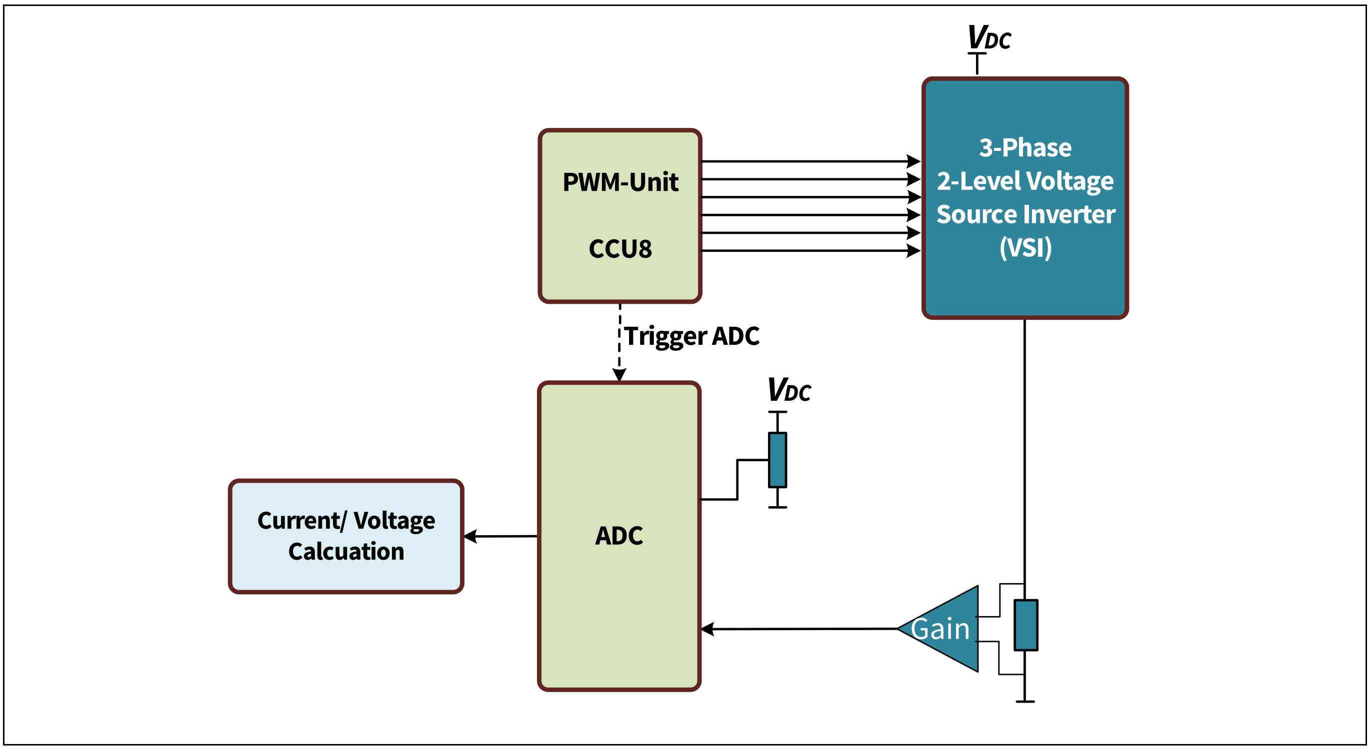


Figure 12 Block diagram of voltage and current measurement module

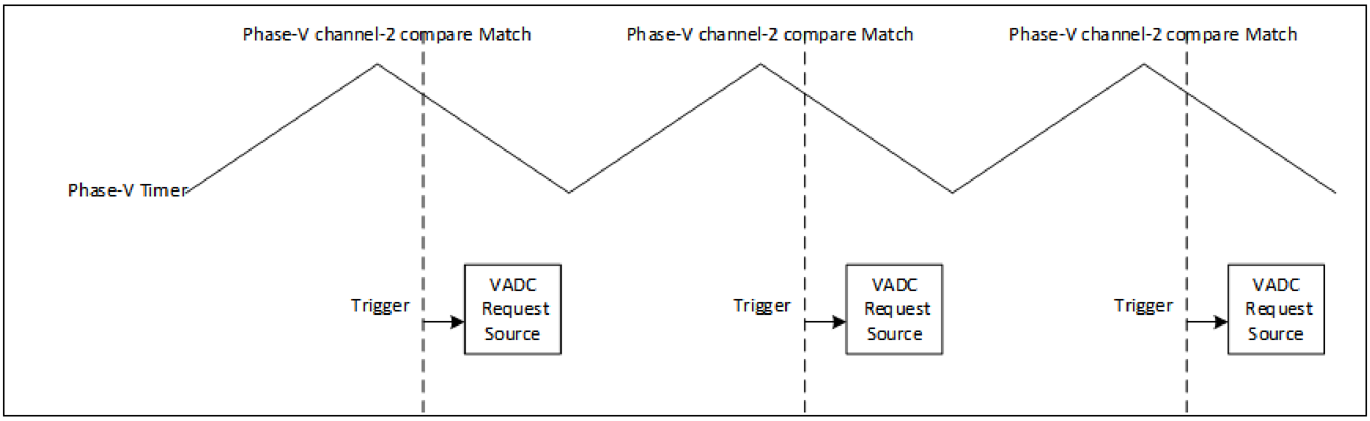


Figure 13 Voltage and current measurement is triggered by Phase V compare match of CCU8

Table 7 Voltage and current measurement resource description

Resource	Description	Connected to
CCU80.SR2	Triggers the ADC channel measurement.	VADC0
VADC_G1.CH5	DC Link current measurement.	-
VADC_G1.CH6	IDC average current measurement.	-
VADC_G1.CH1	VDC link current measurement.	-
VADC_G1.CH7	Potentiometer measurement.	-
VADC_G0.CH6	User defined measurement 1.	-
VADC_G0.CH2	User defined measurement 2.	-
VADC_G0.CH3	User defined measurement 3.	-
VADC_G0.CH0	User defined measurement 4.	-

2.5 Speed and position calculation

Motor position and speed is determined in the Speed and Position Calculation block. Speed and position is detected based on the changing states of the motor outputs as the motor is rotating.

- For sensor-based control, Hall sensors are used to detect the changes in the motor outputs. The Position Interface (POSIF) module is used to decode the Hall sensor outputs.
- For sensor-less control, the VADC module is used to detect the zero-crossing events from the Back-EMF of the motor.

2.5.1 Speed and position calculation for sensor-based control

In a BLDC motor deploying sensor-based motor control, 3 Hall sensors are separated at 120 degree. The Hall sensors are used to get the motor position and speed. The Hall sensor inputs are then interfaced to the Position Interface (POSIF) peripheral of the XMC. When used with the Capture and Compare Unit 4 (CCU4), the motor speed and position can be determined.

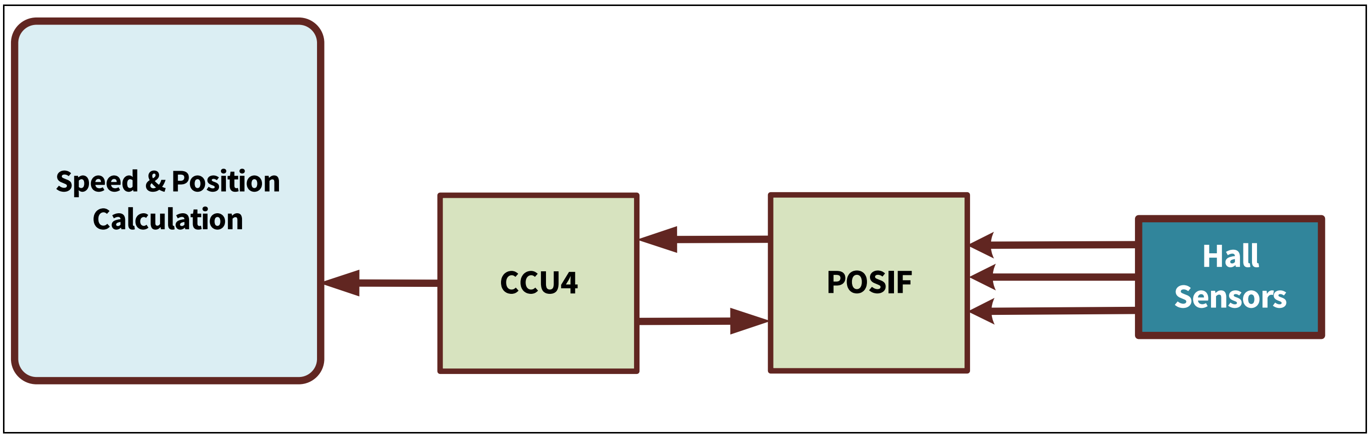


Figure 14 Block diagram of speed and position calculation module

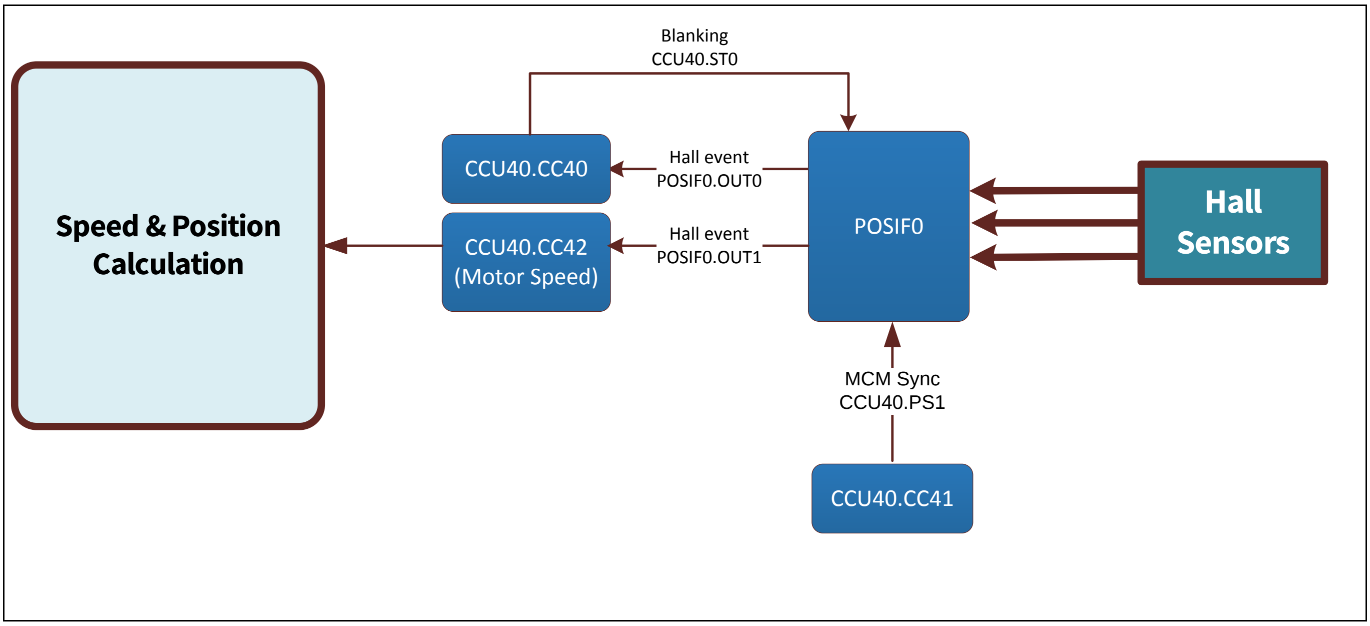


Figure 15 Hardware block interconnect for speed and position calculation module

Table 8 Speed and position resource description

Resource	Description	Connected to
POSIF0.OUT0	Hall inputs edge detection trigger.	CCU40.CC40
POSIF0.OUT1	Hall Correct Event.	CCU40.CC41
CCU40.CC40	Generated the delay (or blanking time) between the edge detection of the Hall Inputs and the actual sampling. This helps to avoid any noise in the detected Hall signal.	POSIF0.HSD[B..A]
CCU40.CC41	This slice is configured for multi-channel pattern synchronization.	POSIF0.MSYNC[D..A]
CCU40.CC42	This slice is configured in Capture Mode, to capture the time between Correct Hall Events (storing the motor speed between two correct Hall events). The POSIFx.OUT1 of the POSIF is used as capture trigger for the Slice.	-

2.5.2 Speed and position calculation for sensor-less control

In sensor-less block commutation, Back-EMF (BEMF) of the un-energized phase is used to sense the motor position. The BLDC motor is characterized by a two phase ON operation used to control the inverter. In this control scheme the torque produced follows the principle that the current flows in only two of the three phases at a time, and no torque is produced in the region of the BEMF zero crossings.

In the BLDC motor control software using sensor-less control, the POSIF peripheral is configured in standalone multi-channel mode for updating the commutation pattern. The VADC peripheral is used to detect the BEMF zero crossing point. This provides the input to the CCU4 module to determine the motor speed. The commutation pattern is updated in-between the zero crossings.

The following figure describes the electrical waveforms in the BLDC along with BEMF zero crossing points and commutation points in each phase.

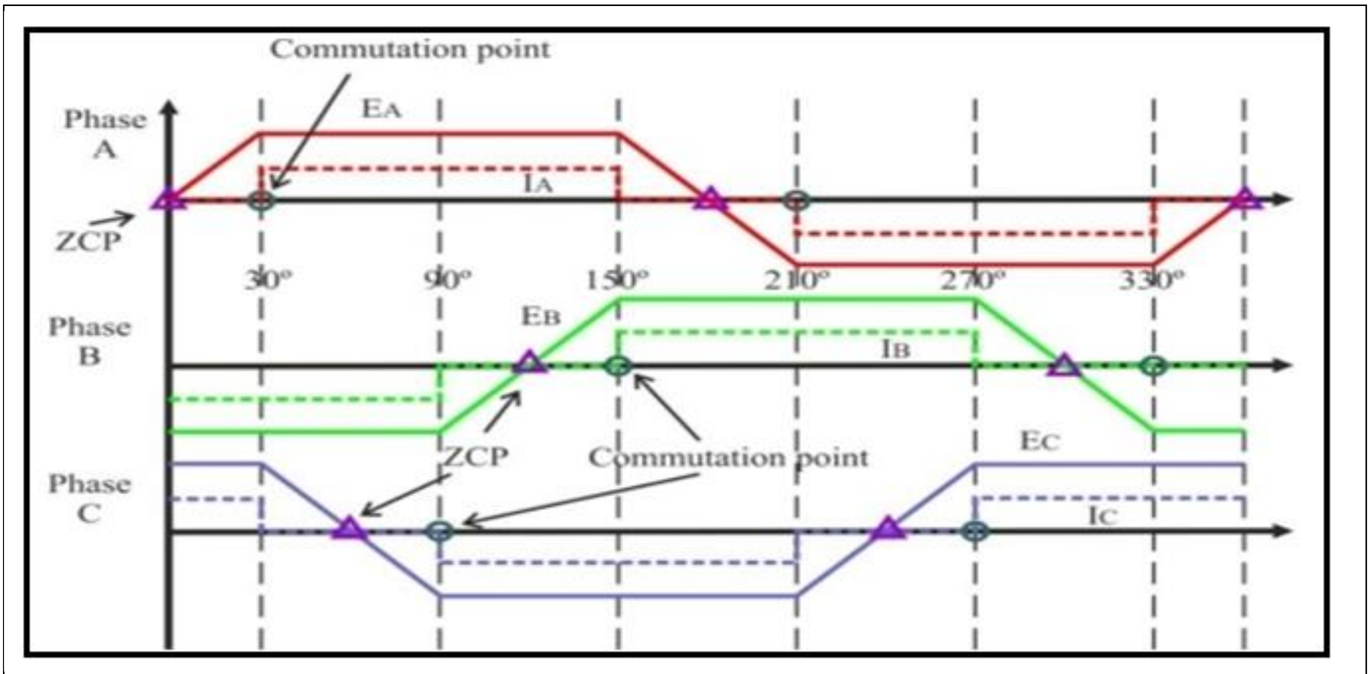


Figure 16 BLDC commutation in sensorless mode

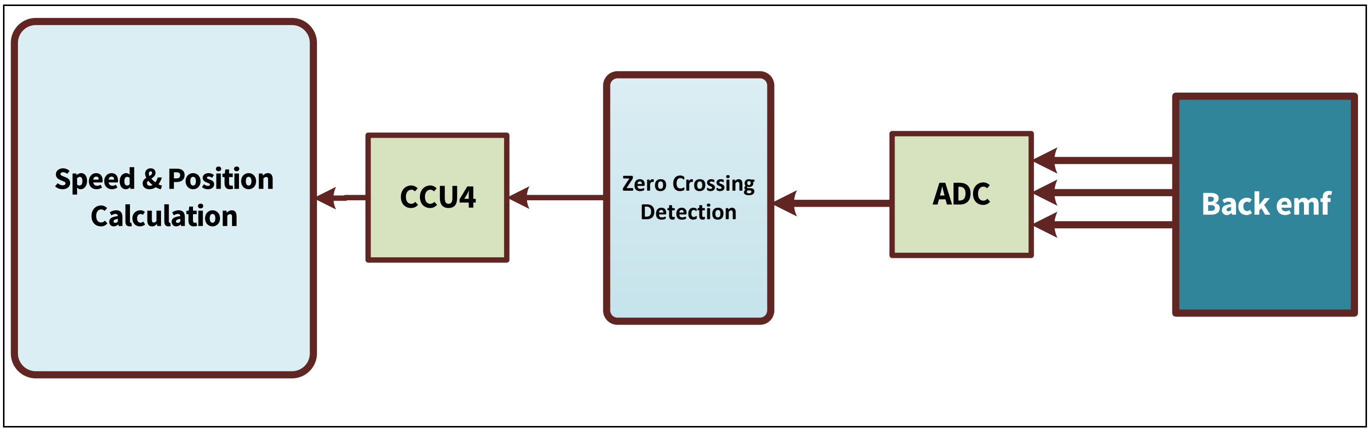


Figure 17 Block diagram of speed and position calculation module

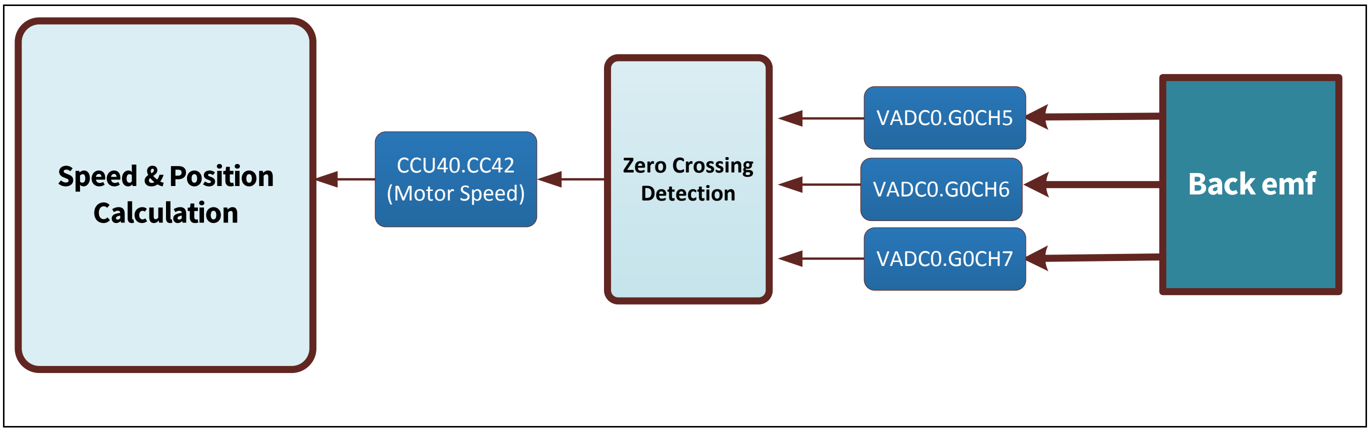


Figure 18 Hardware block inter-connect for speed and position calculation module

Table 9 Speed and position resource description

Resource	Description	Connected to
CCU40.CC40	Commutation timer (prescaler value calculated in zero crossing event).	-
CCU40.CC41	This slice is configured for multi-channel pattern synchronization.	POSIF0.MSYNC[D..A]
CCU40.CC42	This slice is configured in Capture Mode, to capture the time between two consecutive zero-crossing events.	-
VADC_G0.CH5	Phase U BEMF Channel configured for inbound event with global boundary to determine the BEMF zero-crossing point.	-
VADC_G0.CH6	Phase V BEMF Channel configured for inbound event with global boundary to determine the BEMF zero-crossing point.	-
VADC_G0.CH7	Phase W BEMF Channel configured for inbound event with global boundary to determine the BEMF zero-crossing point.	-

2.6 Adaptive Hall pattern learning

In a sensor-based motor control solution, Hall sensors are used to provide rotor position feedback to determine the speed and position of the motor. In addition, Hall sensor-based motor cannot be driven if the Hall pattern information is not available. The sequence of the Hall pattern excitation needs to be determined to ensure that the motor operates correctly.

In the BLDC motor control software, an adaptive Hall learning feature provides a method to generate the correct 6-step commutation pattern for 3-phase BLDC motors. This feature is supported for motors in which 3-Hall sensors are placed at relative angle of 120 degree electrical from each other, and their transitions are aligned to zero crossing of phase-to-phase BEMFs.

2.6.1 Input settings for adaptive Hall pattern learning

To run the Hall based BLDC motor requires a proper sequence of excitation of motor phases with respect to binary code generated from 3-Hall sensors. The Hall learning technique captures and defines these sequences automatically. Adaptive Hall learning is achieved by exciting the motor phase windings with a pre-defined excitation pattern, aligning the rotor to each commutation sequence, and reading the Hall signal code.

For successful tuning, it is important that the rotor is aligned every time to a new applied sequence. This operation is equivalent to forcing the motor to run in an open loop, step-by-step manner. Each commutation pattern is applied to the motor for a defined period.

For the adaptive Hall pattern learning, the two input settings are:

- Open Loop Speed
  - Period that the commutation pattern is applied to the motor.
- Open Loop Voltage
  - PWM duty cycle applied to drive the motor.

These open loop settings (speed and voltage) are defined by the load on the motor, and are required to get the motor locked at some position. At this point Hall sensor output is read to get the Hall pattern corresponding to the applied commutation pattern. The next commutation pattern is applied to move the motor forward and get the Hall pattern. This procedure is repeated to capture the required sequences.

Note: If the motor does not rotate, the motor voltage needs to increase gradually until the motor rotates correctly.

2.7 Sensor-less Motor control start-up mechanism

In a sensor-less solution, the start-up is the most important part to ensure a successful sensor-less operation.

During the start-up, the Back-EMF is very small (or even zero). This makes it difficult to sense an accurate zero crossing and leads to incorrect detection of the rotor position. This leads to the software being unable to control the motor properly.

Attention: The effect of wrong phase energization can lead to reverse rotation in start-up. This is a condition that must be avoided.

3 BLDC motor control software configuration

3.1 Software organization and file structure

BLDC motor control software is developed based on a well-defined layered approach. The layered architecture is designed in such a way as to separate modules into groups. This allows different modules in a given layer to be replaced without affecting the performance in other modules and the complete system.

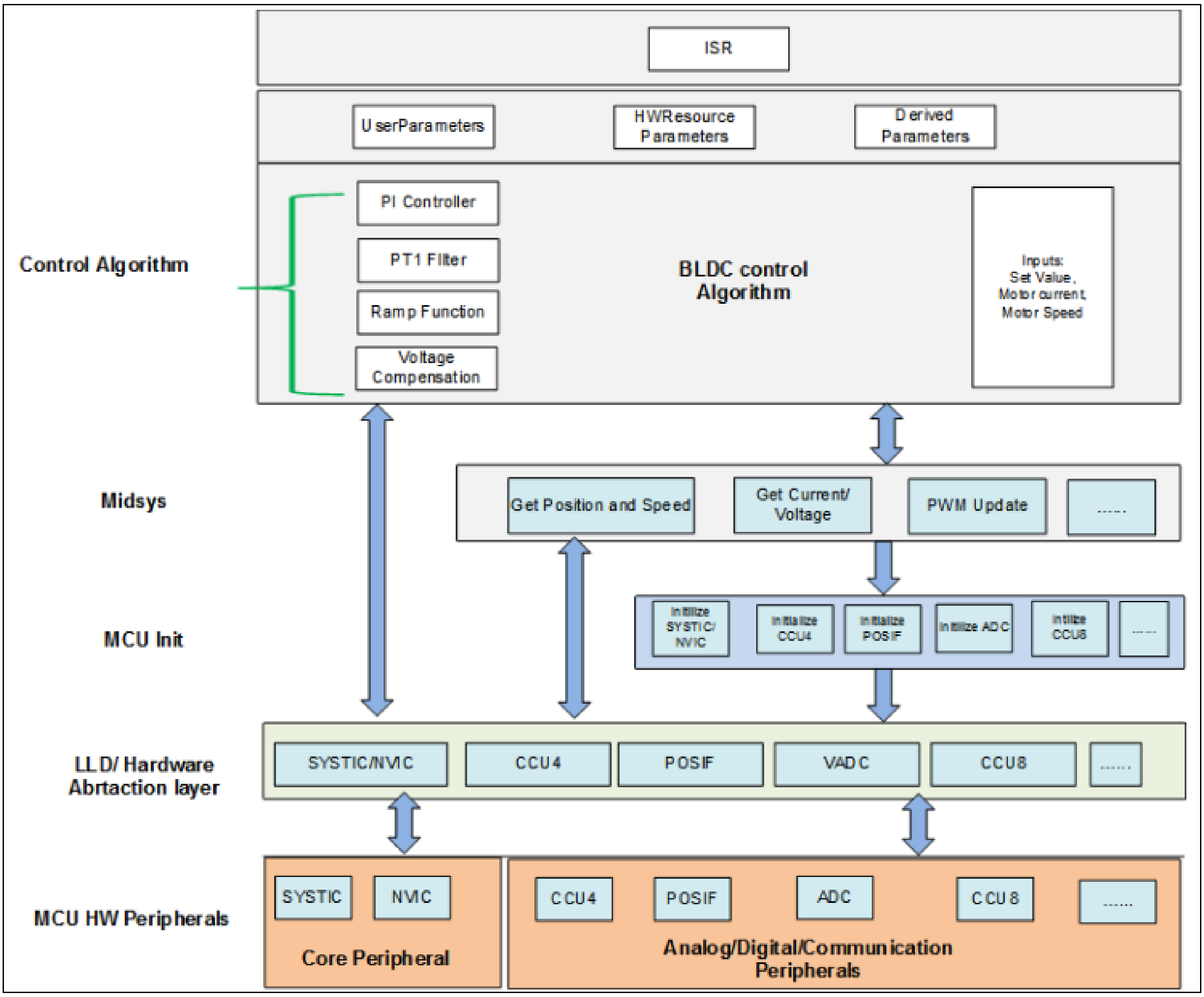


Figure 19 Software layer structure of BLDC scalar control library

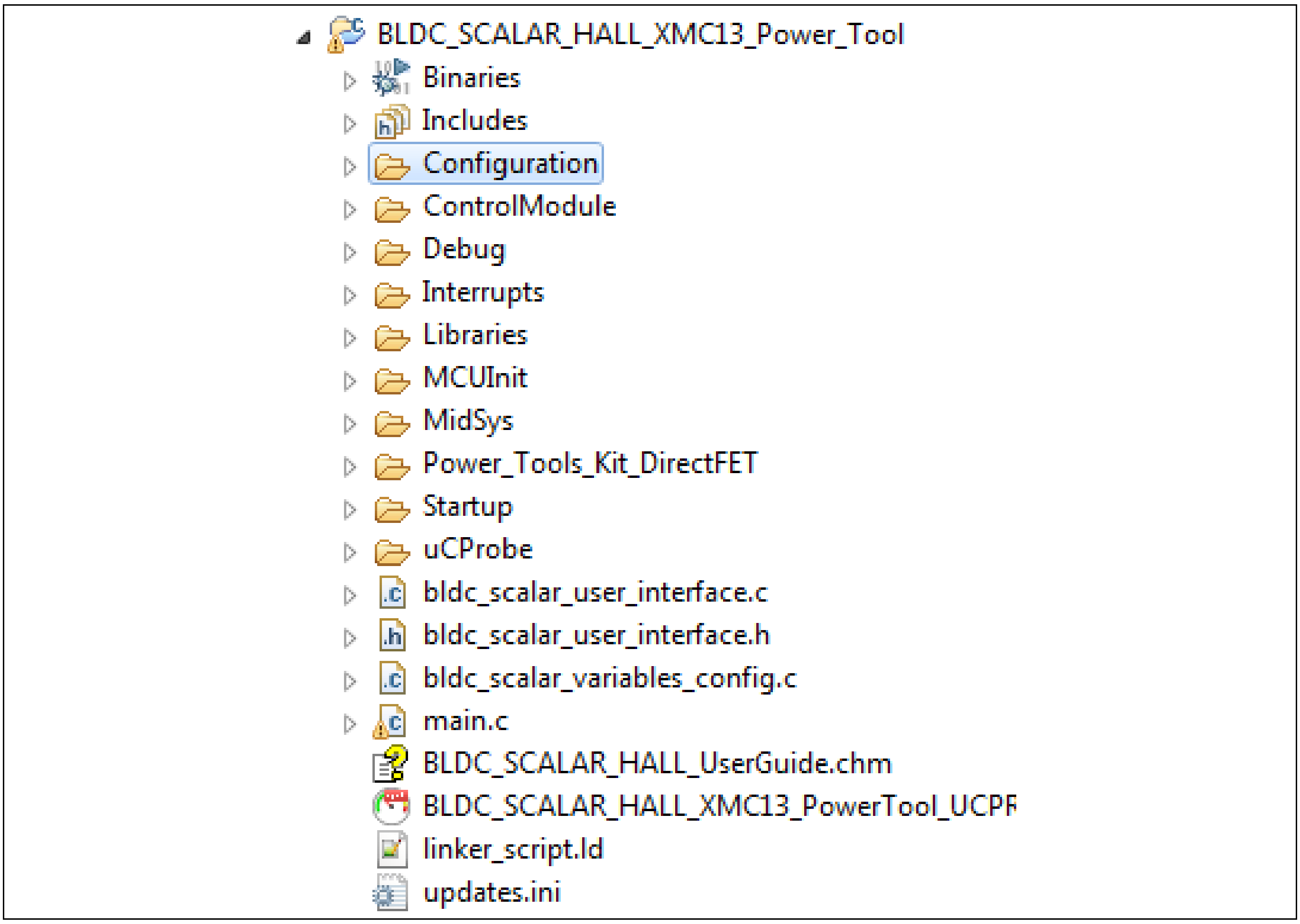


Figure 20 File/folder structure of the BLDC scalar control library

Table 10 Descriptions of the software layers

Layers	Description	Folders
Control Algorithm	This layer consists of software Control library modules. This includes speed control, torque control, voltage compensation, PI, and PT1. All the software control library routines mentioned above are called from Interrupt Service Routines (ISRs). In order to provide flexibility to the user to choose a different sampling frequency for a high priority control task versus a slower control task, the design provides two independent ISRs. For example, you could choose to execute the current PI controller, a PWM update, and voltage compensation in a fast Control loop (CCU8 period match), while a ramp function can be implemented in SYSTICK ISR. The advantage is that the ramp and state machine task does not get affected by a PWM frequency change.	• Configuration • ControlModule • Interrupts
Functional/Middle System	Provides routines for PWM and position, speed, and ADC measurements. The main purpose of this layer is to give flexibility to add/remove sensors for control feedback purposes. You could for example modify files in this layer to change the ADC current reading either from DC_link or low-side phase-current sense, without modifying modules in the top layer interface.	• MidSys
MCU Level (MCUInit)	Contains the initialization of the MCU Peripheral. It contains LLD data structure initialization and peripheral initialization functions. This layer closely interacts with XMC LLD and the MIDSYS layer to configure each peripheral.	• MCUInit
LLD layer	This is the Hardware Abstraction Layer to the MCU peripherals.	• Libraries • Configuration

Table 11 Software configuration project names

Project Name	XMC1302	XMC4400	Sensor-based	Sensor-less
BLDC_SCALAR_HALL_XMC13	√	-	√	-
BLDC_SCALAR_HALL_XMC44	-	√	√	-
BLDC_SCALAR_SL_ADC_XMC13	√	-	-	√

Table 12 Software configuration files

Folder	Files	Sensor-based	Sensor-less
Configuration	bldc_scalar_common.h	√	√
	bldc_scalar_derived_parameter.h	√	√
	bldc_scalar_mcuhw_config.h	√	√
	bldc_scalar_user_config.h	√	√
ControlModule	bldc_scalar_control_scheme.c	√	√
	bldc_scalar_control_scheme.h	√	√
	bldc_scalar_pi.h	√	√
	bldc_scalar_pt1_filter.h	√	√
	bldc_scalar_ramp_generator.c	√	√
	bldc_scalar_ramp_generator.h	√	√
	bldc_scalar_control_hall.c	√	-
	bldc_scalar_control_hall.h	√	-
	bldc_scalar_control_sensorless.c	-	√
	bldc_scalar_control_sensorless.h	-	√
	bldc_scalar_inductive_sensing.c	-	√
	bldc_scalar_inductive_sensing.h	-	√
	bldc_scalar_bemf_zero_cross.c	-	√
Interrupts	bldc_scalar_control_loop.c	√	√
	bldc_scalar_ctrap.c	√	√
	bldc_scalar_protection_error.c	√	√
	bldc_scalar_hall_event.c	√	-
	bldc_scalar_pattern_shadowtx.c	√	-
	bldc_scalar_state_machine.c	√	-
	bldc_scalar_sl_state_machine.c	-	√
MidSys	bldc_scalar_current_motor.c	√	√
	bldc_scalar_current_motor.h	√	√
	bldc_scalar_pwm_bc.c	√	√
	bldc_scalar_pwm_bc.h	√	√
	bldc_scalar_volt_dcbus.c	√	√
	bldc_scalar_volt_dcbus.h	√	√

Folder	Files	Sensor-based	Sensor-less
	bldc_scalar_volt_potentiometer.c	√	√
	bldc_scalar_volt_potentiometer.h	√	√
	bldc_scalar_volt_userdef.c	√	√
	bldc_scalar_volt_userdef.h	√	√
	bldc_scalar_speed_pos_hall.c	√	-
	bldc_scalar_speed_pos_hall.h	√	-
	bldc_scalar_volt_dcbus.c	√	-
	bldc_scalar_volt_dcbus.h	√	-
	bldc_scalar_speed_pos_sl.c	-	√
	bldc_scalar_speed_pos_sl.h	-	√
	bldc_scalar_volt_3phase.c	-	√
	bldc_scalar_volt_3phase.h	-	√
Root	bldc_scalar_user_interface.c	√	√
	bldc_scalar_user_interface.h	√	√
	bldc_scalar_variables_config.c	√	√

3.2 Configuring the BLDC motor control software

To configure the BLDC motor control software for a new motor requires only configuration changes to files in the following folders:

- Configuration folder
  - Hardware and user configuration.
- Interrupt folder
  - State machine customization.

Table 13 Folders

File	Options	Sensor-based	Sensor-less
Configuration folder			
bldc_scalar_mcuhw_config.h	Pin Selection.	√	√
bldc_scalar_user_config.h	– Motor and Power Board selection. – ADC Measurement configurations. – Hall Pattern Learning configurations. – Protection modes.	√	√
Interrupts folder			
bldc_scalar_state_machine.c	System timer events used for state machine.	√	-
bldc_scalar_sl_state_machine.c		-	√

4 Resources

- XMC1000 motor control application kit.
  http://www.infineon.com/cms/en/product/productType.html?productType=db3a30443ba77cfd013baec9ca5c0caa
- XMC4400 motor control application kit.
  http://www.infineon.com/cms/en/product/evaluation-boards/KIT_XMC44_AE3_001/productType.html?productType=db3a30443cd75eda013cd984f125047e
- BLDC Motor Control 3-Hall Sensor Example with uC Probe for XMC1300 series.
  http://www.infineon.com/cms/en/product/productType.html?productType=db3a30443ba77cfd013baec9ca5c0caa#ispnTab12
- BLDC Motor Control Sensorless Example with uC Probe for XMC1300 series.
  http://www.infineon.com/cms/en/product/productType.html?productType=db3a30443ba77cfd013baec9ca5c0caa#ispnTab12
- BLDC Motor Control 3-Hall Sensor Example with uC Probe for XMC4400 series.

Revision history

Major changes since the last revision

Page or Reference	Description of change
All pages	Initial release

Trademarks of Infineon Technologies AG
AURIX™, C166™, CanPAK™, CIPOS™, CoolGaN™, CoolMOS™, CoolSET™, CoolSiC™, CORECONTROL™, CROSSAVE™, DAVE™, DI-POL™, DrBlade™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPACK™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, Infineon™, ISOFACE™, IsoPACK™, i-Wafer™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OmniTune™, OPTIGA™, OptiMOS™, ORIGA™, POWERCODE™, PRIMARION™, PrimePACK™, PrimeSTACK™, PROFET™, PRO-SIL™, RASIC™, REAL3™, ReverSave™, SatRIC™, SIEGET™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, SPOC™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Trademarks updated August 2015

Other Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2017-01-03

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2017 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AP32359

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.