Application Note

MSPM0 Motor Control

Aaron Barrera

ABSTRACT

With TI’s new MSPM0 MCU portfolio optimized for motor control applications, MSP Motor Control is your one-stop shop for spinning your Brushed, Stepper, or 3-phase motor with an MSPM0 MCU in less than 10 minutes.

Table of Contents

List of Figures

List of Tables

Trademarks

Launchpad™ and Code Composer Studio™ are trademarks of Texas Instruments.
Eclipse® is a registered trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

MSP Motor Control is a middleware package integrated within MSPM0-SDK and to allow users to spin motors in 10 minutes or less using small-scale, simplified MSPM0 firmware examples with popular motor driver designs and topologies. Table 1-1 shows the types of motors and motor control algorithms are currently supported for industrial and automotive applications.

Table 1-1. MSP Motor Control Supported Motor Types, Drivers, and Interfaces

Motor Control Type	MSPM0 Launchpad™	Driver IC	Interface
Brushed-DC	LP-MSPM0L1306	• H-bridge motor driver • H-bridge gate driver	• PWM (2x) • Phase/Enable (PH/EN) • Half-bridge • Independent mode
Stepper	LP-MSPM0L1306	• Dual H-bridge motor driver with current regulation • Dual H-bridge intelligent motor driver with current sense and stall detect	• PWM (4x) • STEP (1x PWM)
Brushless-DC Sensored Trapezoidal	• LP-MSPM0L1306 • LP-MSPM0G3507	• 3-phase Gate Driver • 3-phase Motor Driver	• PWM (6x)
BLDC / PMSM / ACIM Sensorless Field-Oriented Control	LP-MSPM0G3507	• 3-phase Gate Driver • 3-phase Motor Driver	• PWM (6x)

Note

For BLDC motor control, MSPM0 software only supports 6x PWM control modes, but depending on driver selection, similar performance can be achieved with 3x or 1x PWM modes.

MSP Motor Control provides an abundance of resources inside MSPM0-SDK to support all motor control developments. Examples of resources include:

- Supported hardware designs using MSP control boards (LaunchPads) and motor driver evaluation modules (EVMs)
- Production-ready software using a Hardware Abstraction Layer (HAL) for scalable MSP and driver support
- Easy-to-evaluate Graphical User Interfaces (GUIs) to spin the motor in less than 10 minutes
- User guides and supporting documentation in TI Resource Explorer

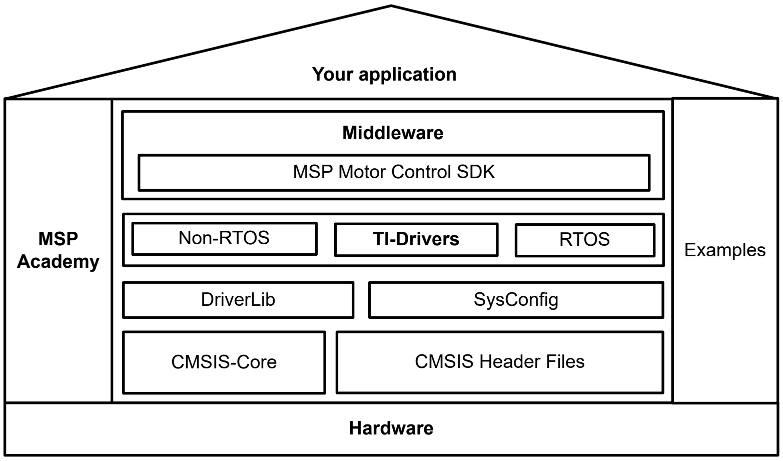


Figure 1-1. MSP Motor Control with MSPM0-SDK Development Resources

As shown in Figure 1-1, MSP Motor Control is embedded within MSPM0-SDK as a middleware package that can interface with Non-RTOS and RTOS drivers, MSPM0 DriverLib support, and graphical configuration tools such as SysConfig. Additionally, MSP Motor Control can pair with other middleware examples such as CAN and LIN drivers, and subsystem code examples can easily be added on top of MSP Motor Control to quickly build software solutions. It provides an easy-to-use ecosystem to evaluate motor control for a wide variety of applications using Brushed-DC, Stepper, BLDC, PMSM, and ACIM motors.

MSP Motor Control requires the following tools:

- Code Composer Studio™ v12.0.0 or newer
  - Support for Eclipse® and Theia packages
- MSPM0-SDK V1.00.00 or newer
- Future support for IAR Systems and Keil embedded development tools

2 Getting Started With MSPM0 Motor Control

To get started with MSPM0 Motor Control, follow these steps. For more information, see Table 2-1.

1. Order an MSPM0 LaunchPad and DRV8xxx motor driver EVM. See *Hardware Support* below.
2. Spin your motor in 10 minutes or less using methods. See *Software Support* below.
   a. Online provided GUI in the TI Gallery
   b. Software example from MSPM0-SDK inside Code Composer Studio (CCS) IDE
   c. Online using CCS Cloud
3. For more resources and documentation, please review the User's Gude resources inside TI Resource Explorer. See *User's Guide* below.

Table 2-1. MSP Motor Control Supporting Resources

	Brushed DC	Stepper	Sensored Trap	Sensorless FOC[1]
Software Support	See *Brushed DC Software Support*	See *Stepper Software Support*	See *Sensored Trap Software Support*	See *Sensorless FOC Software Support*
Hardware Support	See *Brushed DC Hardware Support*	See *Stepper Hardware Support*	See *Sensored Trap Hardware Support*	See *Sensorless FOC Hardware Support*
User's Guide	Brushed-DC User's Guide	Stepper User's Guide	Sensored Trap User's Guide	Sensorless FOC User's Guide

1. Sensorless FOC solution is currently for demo purposes only and not recommended for a full production solution. Motor identification, automated PI tuning, and additional closed-loop features with expanded hardware support will be available by January 2024.

3 Brushed-DC Motor Control

The main features of the MSPM0 Brushed-DC Motor Control designs include:

- 2 PWM control with complimentary outputs and PWM frequency up to 100 kHz
- Adjustable 0-100% duty cycle control
- Braking and direction control
- Voltage reference (VREF) for current limiting
- SPI interface to configure driver settings and diagnose driver faults (SPI devices only)
- Fault detection through fault pin

3.1 Background

Brushed-DC motor control is very simple: a voltage applied across the motor terminals changes the magnetic field on the rotor and create a continuous spinning motion. Despite having drawbacks such as heat dissipation, high rotor inertia, and electromagnetic interference, Brushed DC motors do not require current feedback and are easy to control, making them a simple low-cost design for many applications as shown in Figure 3-1.

Typically, Brushed-DC motors require high current at startup to overcome inertia then only small current to maintain steady state operation. To accomplish this, PWM duty cycle control is utilized to modulate the control waveforms so N-type MOSFETs (which control the switching current) are on for a portion of the PWM period rather than 100% of the time. Additionally, an H-bridge inverter is used to control the motor direction and decay modes for optimum motor braking and recirculation currents.

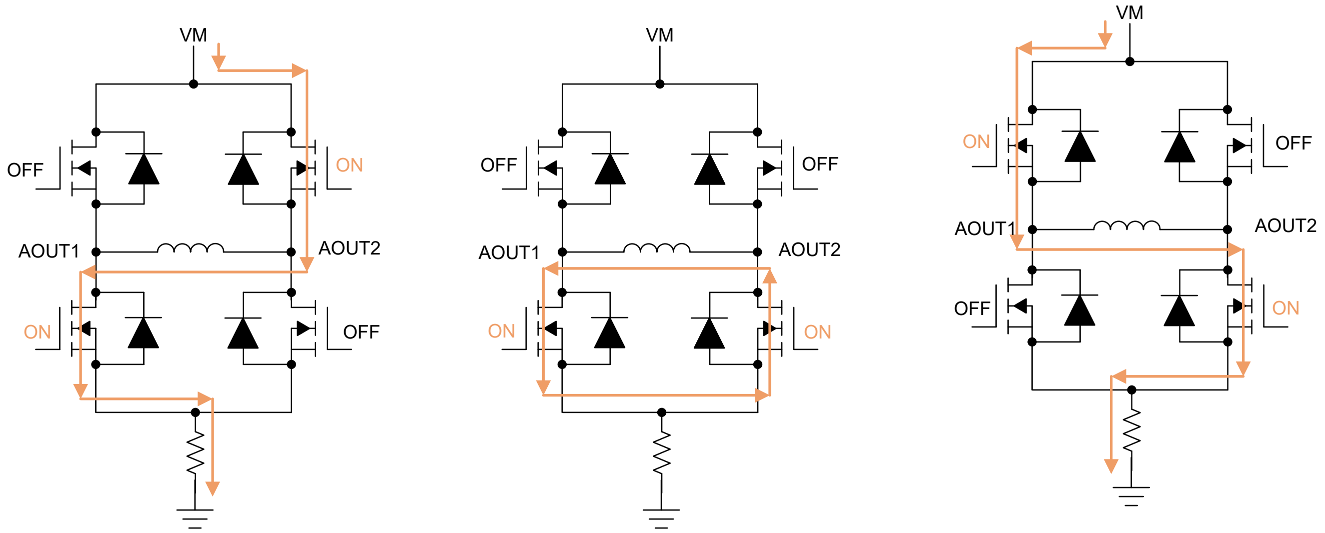


Figure 3-1. H-Bridge Motor Control for Brushed-DC Motors

In systems that require current feedback to the microcontroller for advanced commutations such as ripple counting or torque control, a current shunt resistor can be placed at the low-side of the H-bridge or inline to the motor phase.

H-bridge drivers can have various types of power and analog control integration to reduce the system form factor. The MCU provides the control algorithm, and the analog integration can be integrated into the MCU or the driver.

3.2 Software Architecture

As shown in Figure 3-2, this software package uses the MSPM0 System Configuration tool to initialize the necessary peripherals for Brushed-DC motor control and an abstraction layer to interface with a variety of discrete and integrated stepper driver designs.

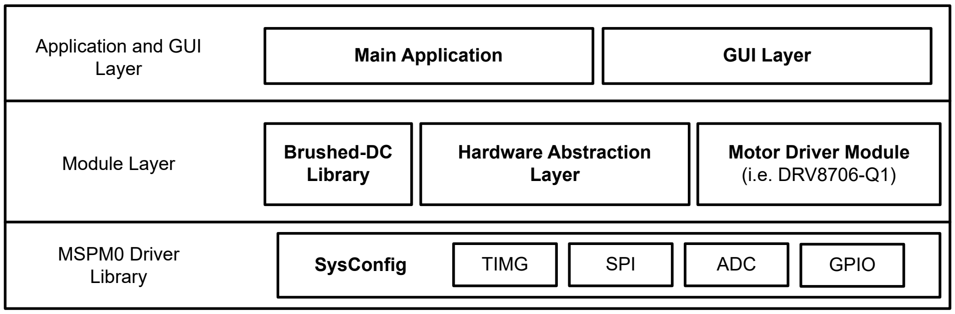


Figure 3-2. Software Architecture for Brushed-DC Library with MSPM0

3.3 Block Diagrams

The block diagrams showcase supported MSPM0L1xxx designs for Brushed DC motors in the Brushed Motor Control examples.

- 2-PWM control interface (PWM)
- Phase-Enable interface (PH/EN)
- Half-bridge control
- Independent mode control

The two topologies that are supported are H-Bridge Motor Driver and H-Bridge Gate Driver.

3.3.1 H-Bridge Motor Driver

MSPM0L1xxx can support 2 PWM control for a low-cost, low voltage H-bridge motor driver to drive small-power BDC motors. These motor drivers are typically very small and integrate the 4 N-type MOSFETs, and these system architectures are used in personal electronics, personal care items, smart locks, robotics/toys, and more. Bus overcurrent detection or basic torque control can be implemented using the integrated OPA or COMP inside the MSPM0. A low-cost MSP MCU provides the PWM, GPIO control, and analog integration while detecting fault conditions using the driver's fault pin or optional overcurrent detection.

As shown in Figure 3-3, the signals used in *H-bridge motor driver* control are:

- 2-PWM signal with adjustable duty cycle and pulse width (PWM)
- Low-power mode signal for drivers with sleep mode pin (nSLEEP)
- Logic-low fault signal from driver (nFAULT)
- Optional low-side current sense from shunt resistor[1]

[1] Current sense can be realized using integrated COMP or OPA, available in MSPM0L130x only.

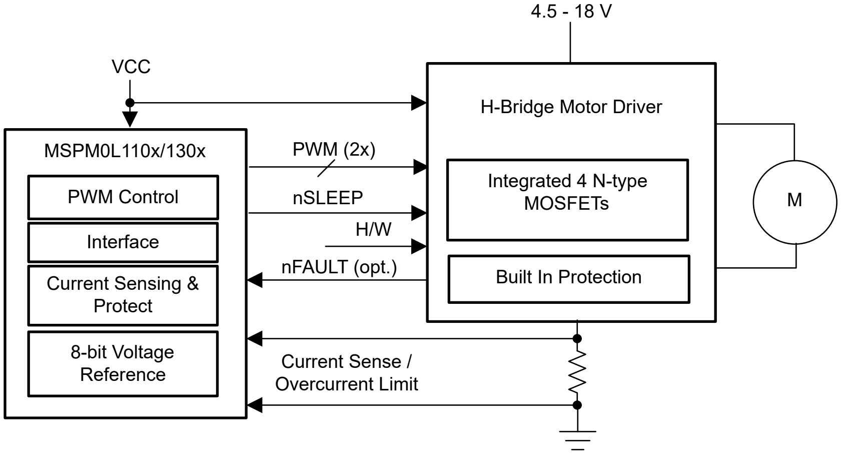


Figure 3-3. Block Diagram of Brushed-DC Motor Control Using MSPM0L1xxx and H-Bridge Motor Driver

3.3.2 H-Bridge Gate Driver

MSPM0L1xxx can support 2 PWM control for a gate driver with optional inline or low-side current sensing to drive high power, higher voltage BDC motors with closed-loop torque control. These gate drivers provide current to drive 4 external N-type MOSFETs, which have higher current and temperature ratings compared to integrated MOSFETs. These system architectures are used in high-power automotive applications such as power modules, door modules, windows, sunroofs, trunks, and other body applications. A low-cost MSP MCU provides the PWM, GPIO, SPI communication interface, and analog integration to support these H-bridge gate drivers.

As shown in Figure 3-4, the signals used in *H-bridge gate driver with inline current sense amplifiers* are:

- 4-PWM signal with adjustable duty cycle and pulse width (PWM)
- Low-power mode signal for drivers with sleep mode pin (nSLEEP)
- Optional SPI read/write interface (for drivers with SPI interface)
- Logic-low fault signal from driver (nFAULT)
- CSA reference (VREF)[1]
- Optional low-side current sense from shunt resistor[2]

(1) CSA reference voltage can be output using 8-bit DAC from the integrated COMP.

(2) Current sense can be realized using integrated COMP or OPA. Both COMP and OPA available in MSPM0L130x only.

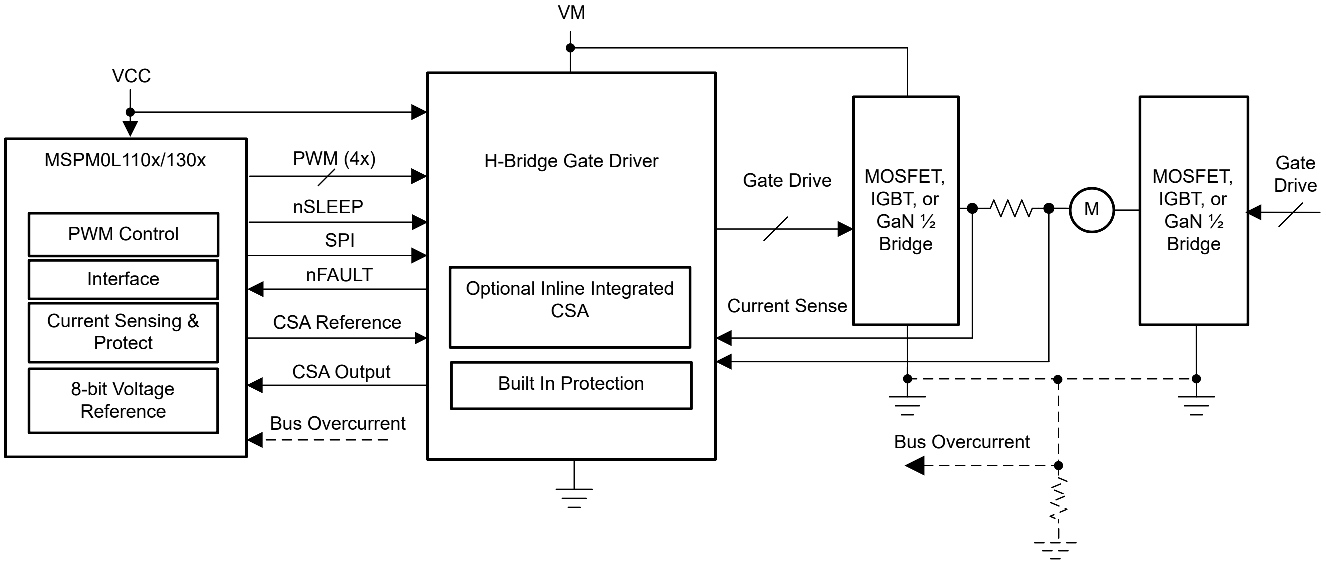


Figure 3-4. Block Diagram of Brushed-DC Motor Control Using MSPM0L1xxx and H-Bridge Gate Driver With Inline Current Sense Amplifiers

3.4 Hardware Support

Table 3-1 shows the hardware supported with the MSPM0-SDK for Brushed-DC motor control.

Table 3-1. Supported Hardware for Brushed-DC Control With MSPM0

LaunchPad	Motor Driver Hardware	Hardware Guide	Control Interface	Current Sense	SPI Driver	Motor Voltage Range	Recommended Motor Power
LP-MSPM0L1306	DRV8706S-Q1EVM	DRV8706S-Q1EVM	PWM (2x), PH/EN, Half-bridge, Independent	1	Yes	4.9 V-37 V	>5W

3.5 Software Support

Table 3-1 shows the software supported for Brushed-DC motor control in TI Resource Explorer.

Table 3-2. Software Support for Brushed-DC Control

Quick Start Guide	Library Software Overview	Code Example(s)	GUI(s)	GUI User Guide
Brushed quick start guide	Brushed library overview	DRV8706-Q1	DRV8706-Q1	Brushed GUI User Guide

3.6 Evaluating Brushed-DC with MSP Motor Control SDK

For full instructions and information, visit the Brushed-DC Motor Control User's Guide in TI Resource Explorer.

To spin a Brushed-DC motor using MSPM0, follow these three simple steps:

1. Order a supported LaunchPad and DRV8xxx EVM. Connect the hardware and motor to the Hardware User's Guide for the specified EVM.
2. Launch the specified GUI from the TI Gallery. Enable the driver, spin the motor, and evaluate device features using the MSPM0 MCU.
3. Evaluate the firmware in Code Composer Studio or on the CCS Cloud IDE.

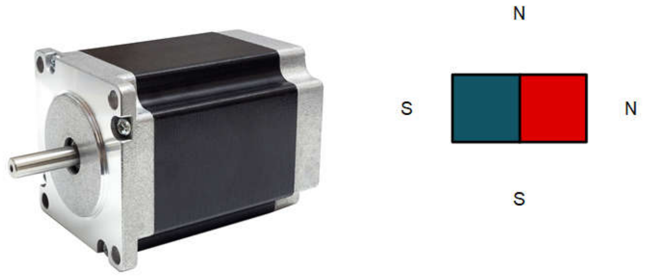
4 Stepper Motor Control

The main features of the MSPM0 Stepper Motor Control designs include:

- 4-PWM stepper control up to 100 kHz using full- or half-stepping, or 1-PWM stepper control up to 500 kHz using 1/256 microstepping
- Spin or step motor in open loop based on selected step size
- Voltage reference for current limiting
- SPI interface to configure driver settings and diagnose driver faults (SPI devices only)
- Fault detection using fault pin

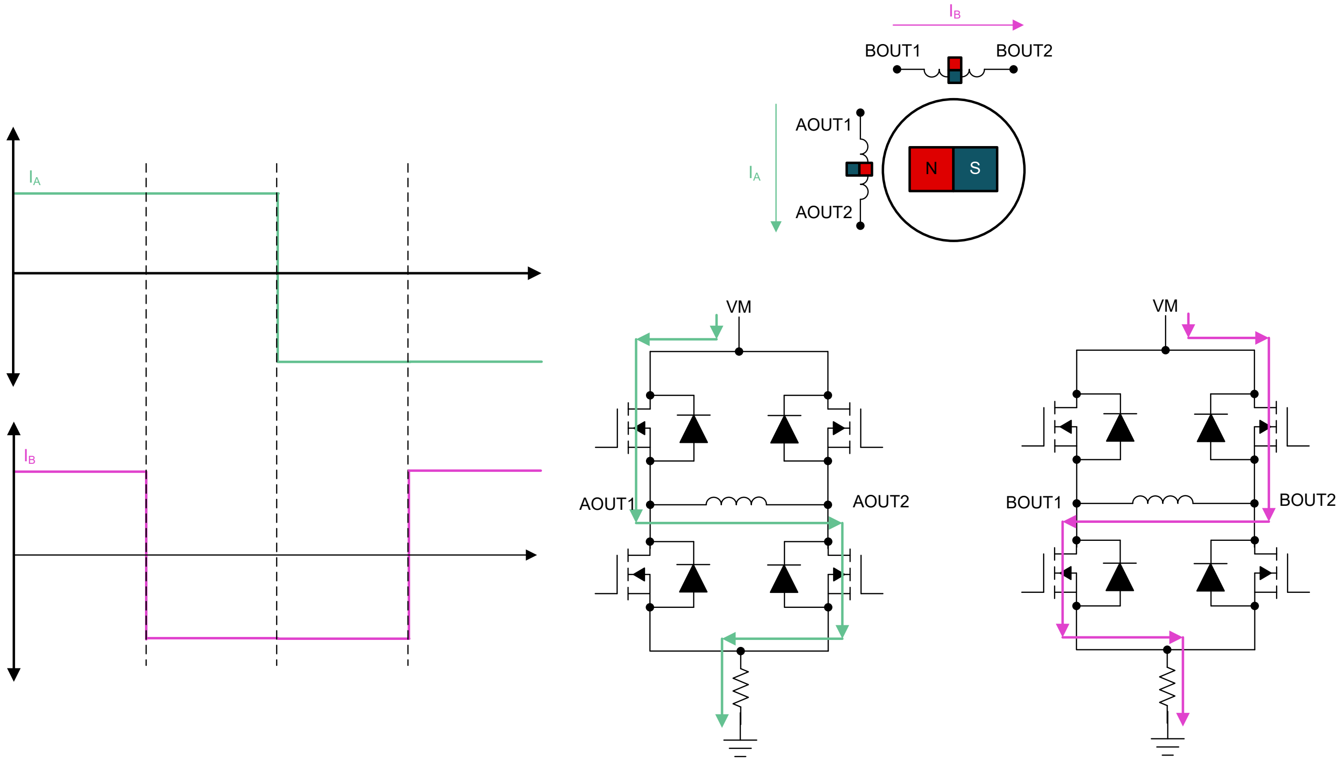
4.1 Background

Stepper motors are prevalent because they allow for fine position control without requiring external sensors or complex control algorithms. This is known as open loop position control, which can be used to allow the stepper motor to provide continuous motion of maintain a fixed rotor position depending on the system requirements. These features make steppers easy to implement and a low-cost design for system designers.



Stepper motors are commutated by energizing two phases, which requires two full H bridges to provide current direction and magnitude for each of the phase currents. This is used to move the rotor in a stepper motor with position and torque control, depending on the application. Bipolar stepper motors also have a specified resolution in degrees per full step to indicate how many mechanical degrees a stepper will commutate when a full electrical period is applied on each phase.

Typically, *full-stepping* refers to a full electrical period using four indexes to energize the two windings in all four current directions to commutate the stepper the specified mechanical period. To provide more precise position and torque control, *non-circular* or *circular half-stepping* provides more resolution and current regulation to *smoothen* the electrical commutation pattern by using eight indexes to energize the two windings. Both full-stepping and circular half-stepping current waveforms are shown in Figure 4-1. These methods of full stepping, non-circular half-stepping, and circular half-stepping are simple to implement using a microcontroller's GPIO or PWM pins and a simple stepper driver with a PWM interface.



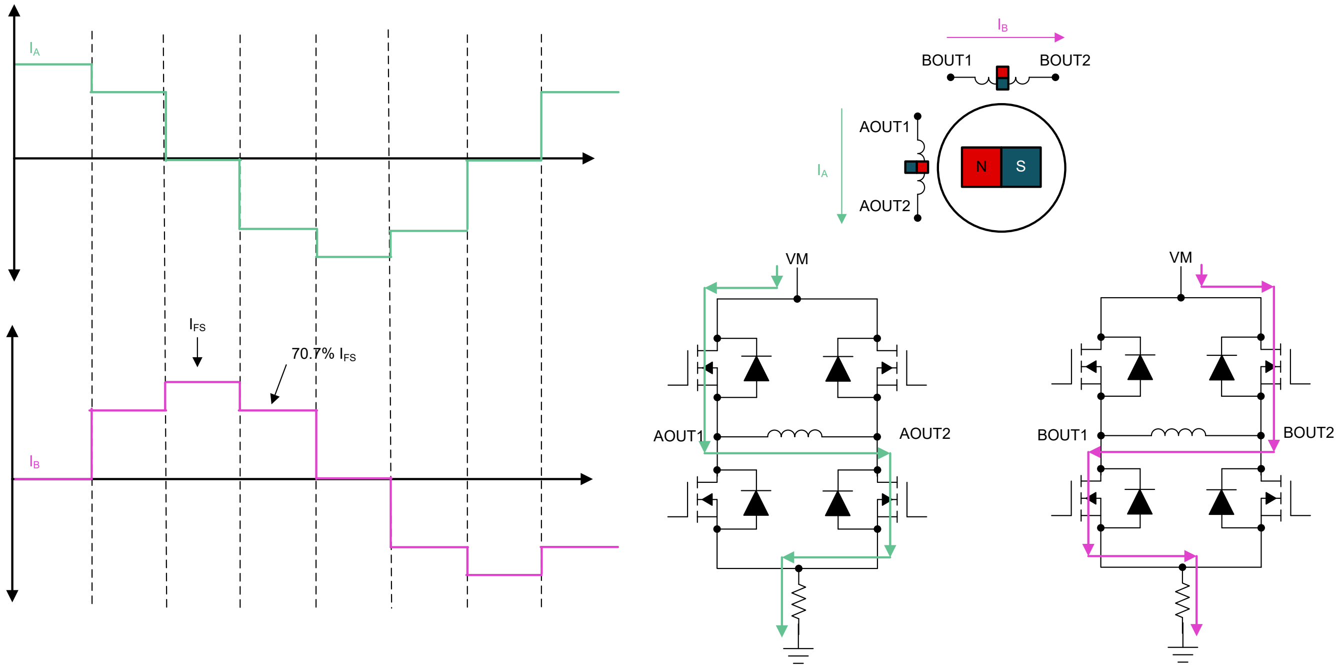


Figure 4-1. Full and Circular-Half Stepping in Stepper Motor Control

Smaller step sizes can be implemented by factors of half (1/4, 1/8, 1/16, 1/32, etc.) down to 1/256 to create a sinusoidal current drive and reduce the noise and torque implemented in the stepper motor. Using step sizes less than 1/4 is generally referred to as *microstepping* because an indexer and current regulation is used to implement discrete intervals for precise position and torque control. An example of using 1/16 microstepping is shown in Figure 4-2. Microstepping is more difficult to implement in software, so using an intelligent stepper motor driver and a simple MCU with one PWM and analog integration is optimal for low-noise, high-precision stepper drives.

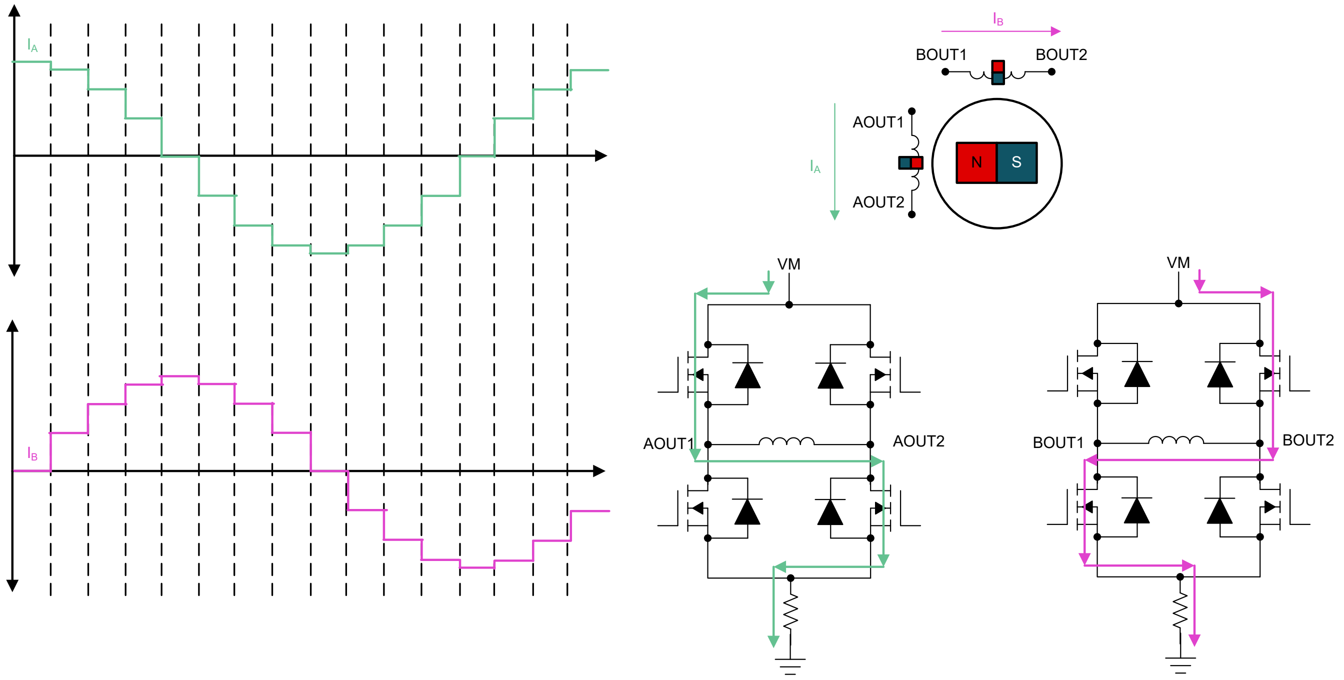


Figure 4-2. Microstepping in Stepper Motor Control

4.2 Software Architecture

As shown in Figure 4-3, this software package uses the MSPM0 System Configuration tool to initialize the necessary peripherals for stepper motor and an abstraction layer to interface with a variety of discrete and integrated stepper driver designs.

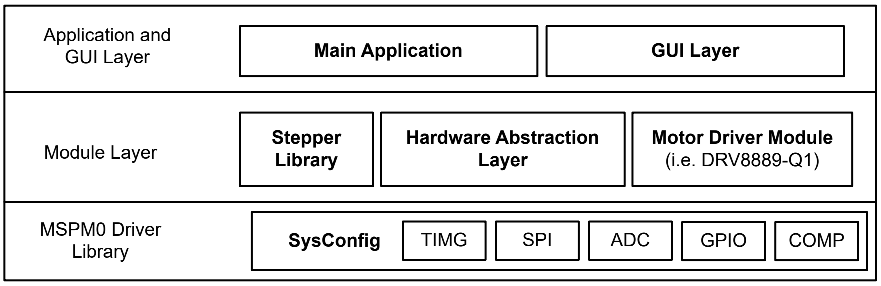


Figure 4-3. Software Architecture for Stepper Library with MSPM0

4.3 Block Diagrams

The block diagrams showcase two supported MSPM0L1xxx designs for stepper motors in the MSPM0L1xxx Stepper SDK:

- 4-PWM control (PWM interface)
- 1-PWM control (STEP interface)

As shown in Figure 4-4, the signals used in *4-PWM control (PWM interface)* are:

- 4 PWM signals with edge-aligned synchronization (PWM 4x)
- 8-bit DAC reference voltage for current regulation (VREF)[1]
- Logic-low fault signal from driver (nFAULT)
- Gate driver shutoff signal (DRVOFF)
- Low-power mode signal for drivers with sleep mode pin (nSLEEP)
- ADC current sense feedback from integrated current sense to measure motor phase currents

[1] VREF available using 8-bit DAC from COMP module, available in MSPM0L130x only.

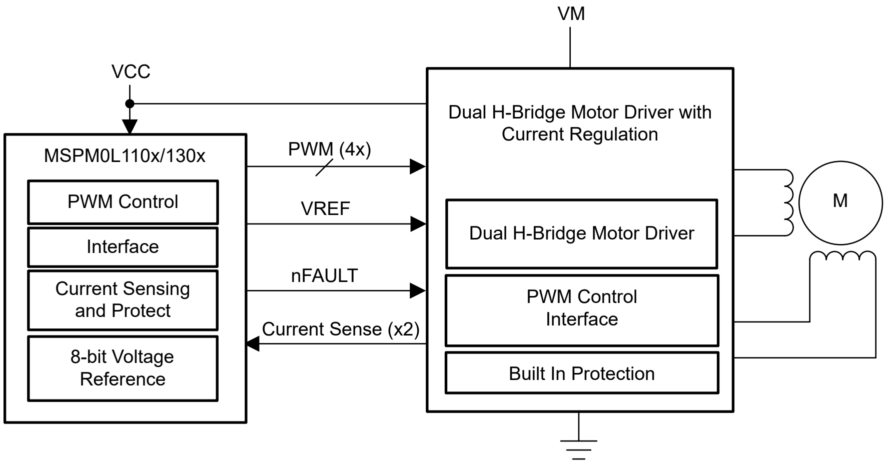


Figure 4-4. Block Diagram for 4-PWM Interface for Stepper Motor Control

As shown in Figure 4-5, the signals used in *1-PWM Control (STEP interface)* are:

- 1-PWM signal with adjustable duty cycle and pulse width (STEP)
- Direction pin (DIR)
- 8-bit DAC reference voltage for current regulation (VREF)
- Logic-low fault signal from driver (nFAULT)
- Gate driver shutoff signal (DRVOFF)
- Low-power mode signal for drivers with sleep mode pin (nSLEEP)
- Optional SPI read/write interface (for drivers with SPI interface)

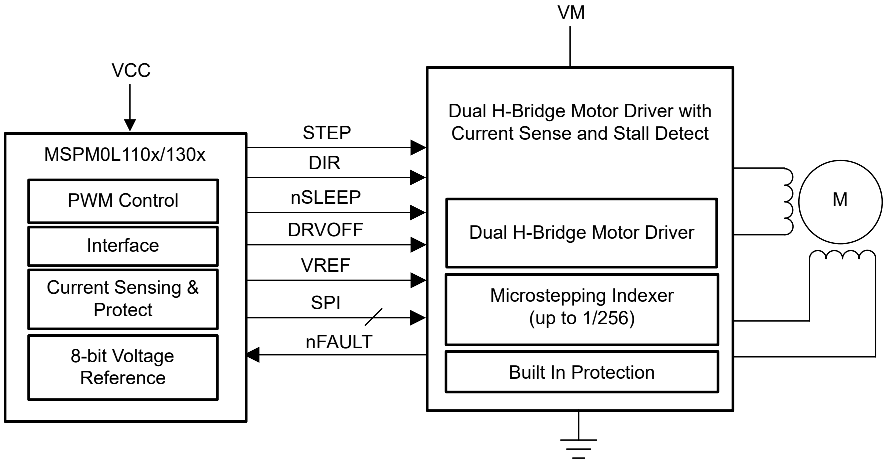

VM
VCC
MSPM0L110x/130x
PWM Control
Interface
Current Sensing & Protect
8-bit Voltage Reference
STEP
DIR
nSLEEP
DRVOFF
VREF
SPI
nFAULT
Dual H-Bridge Motor Driver with Current Sense and Stall Detect
Dual H-Bridge Motor Driver
Microstepping Indexer (up to 1/256)
Built In Protection
M

Figure 4-5. Block Diagram for 1-PWM Interface for Stepper Motor Control

4.4 Hardware Support

Table 4-1 shows the hardware supported with the MSPM0-SDK for Stepper motor control.

Table 4-1. Supported Hardware for Stepper Control with MSPM0

LaunchPad	Motor Driver Hardware	Hardware User Guide	Control Interface	Current Sense	SPI Interface	Motor Voltage Range	Motor Current (Peak)	Motor Power
LP-MSPM0L1306	DRV8411AEVM	DRV8411AEVM	PWM (4)	2	No	1.65 V-11 V	2.5 A	5-10W
	DRV8889-Q1EVM	DRV8889-Q1EVM	STEP (1)	1	Yes	4.5 V-45 V	2.4 A	<75W

4.5 Software Support

Table 3-1 shows the software supported for Stepper motor control in TI Resource Explorer.

Table 4-2. Software Support for Stepper Control

Library Software Overview	Code Example(s)	GUI(s)	GUI User Guide
Stepper library overview	DRV8411A, DRV8889-Q1	DRV8411A, DRV8889-Q1	DRV8411A, DRV8889-Q1

4.6 Evaluating Stepper With MSP Motor Control SDK

For more information, visit the Stepper Motor Control User's Guide in TI Resource Explorer.

To spin a stepper motor using the MSPM0, follow these three simple steps:

1. Order a supported LaunchPad and DRV8xxx EVM. Connect the hardware and motor to the Hardware User's Guide for the specified EVM.
2. Launch the specified GUI from the TI Gallery. Enable the driver, spin the motor, and evaluate device features using the MSPM0 MCU.
3. Evaluate the firmware in Code Composer Studio or on the CCS Cloud IDE.

5 BLDC Sensored Trap Control

The main features of the MSPM0 BLDC Sensored Trap Control designs include:

- 6 PWM control with complimentary outputs, adjustable dead band up to 2 µs, and PWM frequency up to 200 kHz
- Adjustable 0-100% duty cycle control
- Braking and direction control
- Bus voltage, phase voltage, and phase current monitoring
- SPI read/write to configure driver settings and diagnose driver faults (optional)
- Software overcurrent and Overvoltage/Undervoltage detection

5.1 Background

Trapezoidal commutation is the most basic method of spinning a 3-phase Brushless-DC motor. This is accomplished by energizing the windings in a 6-step pattern every 60 electrical degrees so that one phase is sourcing motor current, another phase is sinking motor current, and the last phase remains unconnected (Hi-Z). This produces a 120° trapezoidal-shaped current waveform for each phase as shown in Figure 5-1.

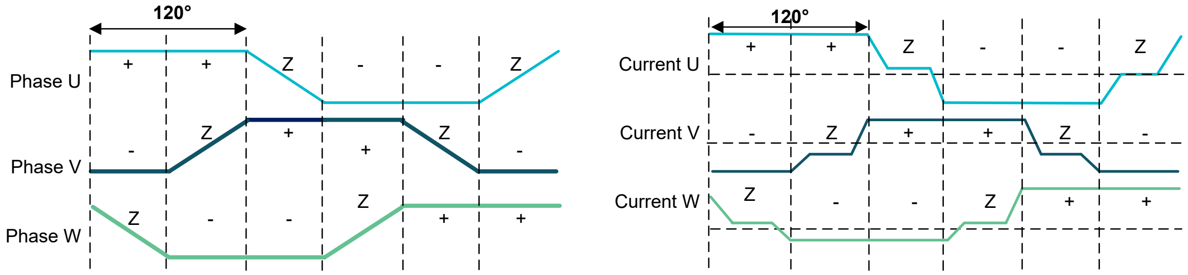


Figure 5-1. Trapezoidal Commutation Phase Voltage and Current Waveforms

To determine the real-time position of the motor and commutate the motor to the next state, Hall sensors are commonly placed between the 3 motor phases to measure the motor's magnetic field and output digital signals to the MCU as shown in Figure 5-2. The three hall sensor signals are accepted as logic inputs and a 6-step table is used to decode the next PWM control state. This is commonly referred to as *Hall-sensored trapezoidal* commutation, which is a low-cost, simple solution to implement that can generate high amounts of torque and speed and minimal MOSFET switching losses. However, it is low resolution and results in torque ripple and audible noise due to a non-ideal current drive.

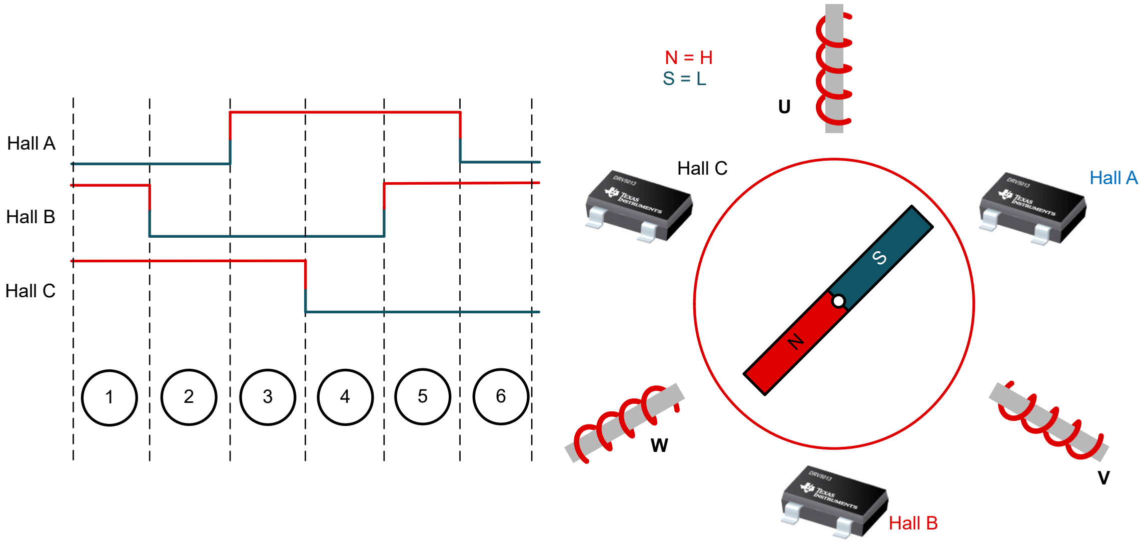


Figure 5-2. Hall Sensor A/B/C Waveforms for Six-Step Commutation

5.2 Software Architecture

As shown in Figure 5-3, this software package uses the MSPM0 System Configuration tool to initialize the necessary peripherals for sensored trapezoidal and an abstraction layer to interface with a variety of 3-phase motor drivers and gate driver solutions.

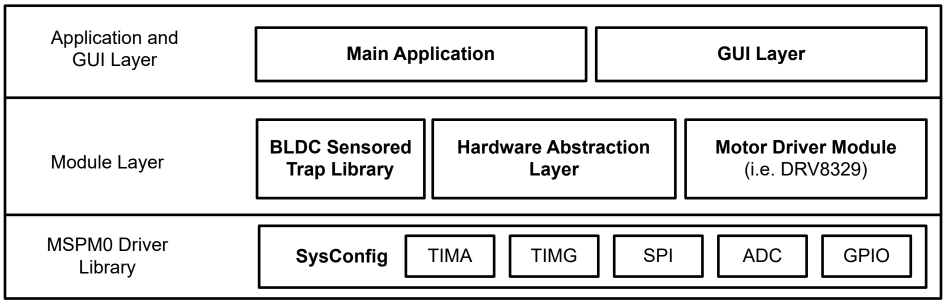

Application and GUI Layer
Main Application
GUI Layer
Module Layer
BLDC Sensored Trap Library
Hardware Abstraction Layer
Motor Driver Module (i.e. DRV8329)
MSPM0 Driver Library
SysConfig
TIMA
TIMG
SPI
ADC
GPIO

Figure 5-3. Software Architecture for BLDC Sensored Trap Library with MSPM0

5.3 Block Diagrams

The block diagram for MSPM0Lxxx / MSPM0Gxxx MCUs and a 3-phase gate/motor drivers showcases supported in the MSP Motor Control is shown in Figure 5-4. The control signals, feedback signals, and interfaces include:

- 6 PWM signals with adjustable dead band (PWM 6x)
- Logic-low fault signal from driver (nFAULT)
- Gate driver shutoff signal (DRVOFF)
- Low-power mode signal for drivers with sleep mode pin (nSLEEP)
- ADC current sense feedback from integrated or external current sense amplifiers to measure motor phase current
- Optional SPI read/write interface (for drivers with SPI interface)

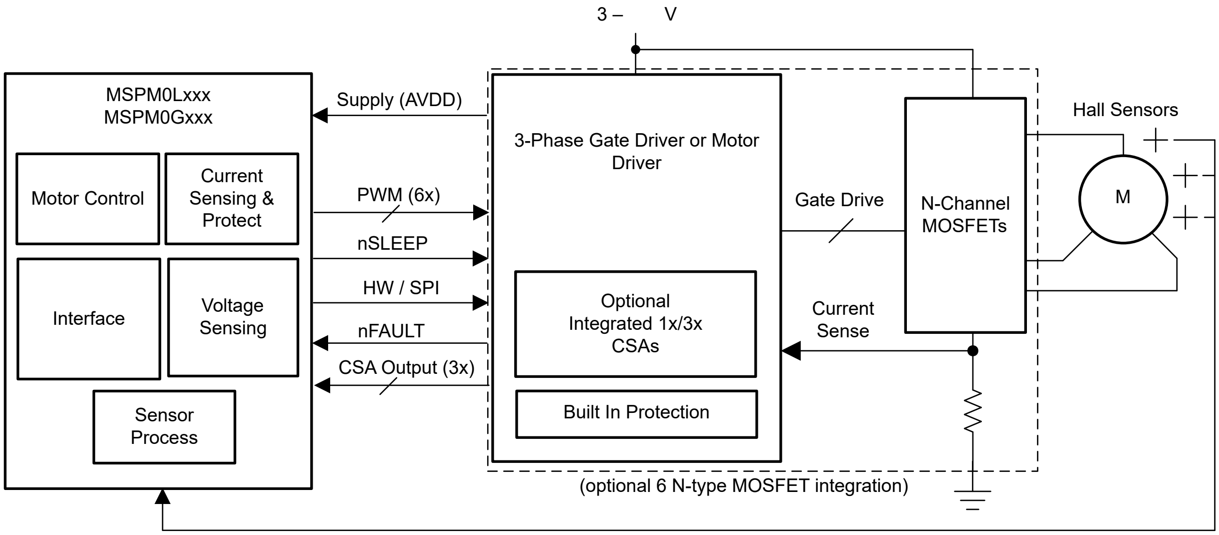


Figure 5-4. Block Diagram for 6-PWM Hall-Sensored Trapezoidal Control for BLDC Motors Using MSPM0Lxxx/MSPM0Gxxx and 3-Phase Gate or Motor Driver

5.4 Hardware Support

Table 5-1 shows the hardware supported with the MSPM0-SDK for BLDC sensored trap motor control.

Table 5-1. Supported Hardware for BLDC Sensored Trap Control with MSPM0

LaunchPad	Supported Motor Driver Hardware	Hardware User Guide	Current Sense Amplifiers	SPI Driver Support	Motor Voltage Range (rec.)	Recommended Motor Power
LP-MSPM0L1306	DRV8329AEVM	DRV8329AEVM	1	No	4.5 V-60 V	<1000 W
LP-MSPM0G3507	DRV8311HEVM	DRV8311HEVM	3	No	3 V-20 V	<25 W
	DRV8316REVM	DRV8316REVM		Yes	4.5 V-40 V	<75 W
	DRV8317HEVM	DRV8317HEVM		No	4.5 V-20 V	<25 W
	DRV8328AEVM	DRV8328AEVM	0		4.5 V-60 V	<1000 W
	DRV8329AEVM	DRV8329AEVM	1			
	BOOSTXL-DRV8323RH	BOOSTXL-DRV8323RH	0		6 V-60 V	
			3			
	DRV8300DIPW-EVM	DRV8300DIPW-EVM	0		4.6 V-100 V	<2000 W
	DRV8353RH-EVM	DRV8353RH-EVM	0		8 V-100 V	
			3			

5.5 Software Support

Table 3-1 shows the software supported for Hall-sensored trap motor control in TI Resource Explorer.

Table 5-2. Software Support for Stepper Control

MSPM0 LaunchPad	Library Software Overview	Software User Guide	Code Example(s)	GUI(s)	GUI User Guide
LP-MSPM0G3507	BLDC Sensored trap library overview	BLDC Sensored trap software user guide	DRV8300, DRV8311, DRV8316, DRV8317, DRV8323RH, DRV8328, DRV8329, DRV8353RH, TIDA-010251	DRV8300, DRV8311, DRV8316, DRV8317, DRV8323RH, DRV8328, DRV8329 DRV8353RH	DRV8300, DRV8311, DRV8316, DRV8317, DRV8323RH, DRV8328, DRV8329, DRV8353RH
LP-MSPM0L1306			DRV8329	DRV8329	

5.6 Evaluating Sensored Trap with MSP Motor Control

For more information, visit the BLDC Motor Control User's Guide in TI Resource Explorer.

To spin a BLDC or PMSM motor with sensored trap using the MSPM0, follow these three simple steps:

1. Order a supported LaunchPad and DRV8xxx EVM. Connect the hardware and motor to the Hardware User's Guide for the specified EVM.
2. Launch the specified GUI from the TI Gallery. Enable the driver, spin the motor, and evaluate device features using the MSPM0 MCU.
3. Evaluate the firmware in Code Composer Studio or on the CCS Cloud IDE.

6 3-Phase Sensorless FOC Control

The main features of the MSPM0 Sensorless FOC design includes:

- 2-shunt based current sense design
- Enhanced sliding mode observer + phase-locked loop observer (eSMO+PLL)
- Configurable user motor parameters and target RPM speed
- Align, slow first cycle, or IPD startup method
- Auto-handoff from started to closed-loop speed control
- Adjustable PWM frequency (up to 50kHz) and deadband

Note

The current solution is currently a demo and is meant for evaluation only. Future MSPM0-SDK releases will include essential sensorless FOC control features, such as motor parameter identification, automated PI tuning, field weakening control, and more.

6.1 Background

This design uses an incremental system build to demonstrate a complete fixed-point sensorless Field-oriented control (FOC) design across a wide range of speeds and takes into consideration torque changes with transient phases by processing a dynamic model of the motor. Each incremental build uses modular software blocks to create a full sensorless-FOC design using two current sensors, macros for angle and speed estimation, speed and torque regulation, Park, Clarke, inverse Park, and space vector PWM generation (SVGEN). A block daigram for Sensorless FOC using two current sensors is shown in Figure 6-1.

For more information on the theoretical background of Sensorless FOC implementation with this design, see Sensorless FOC of PMSM.

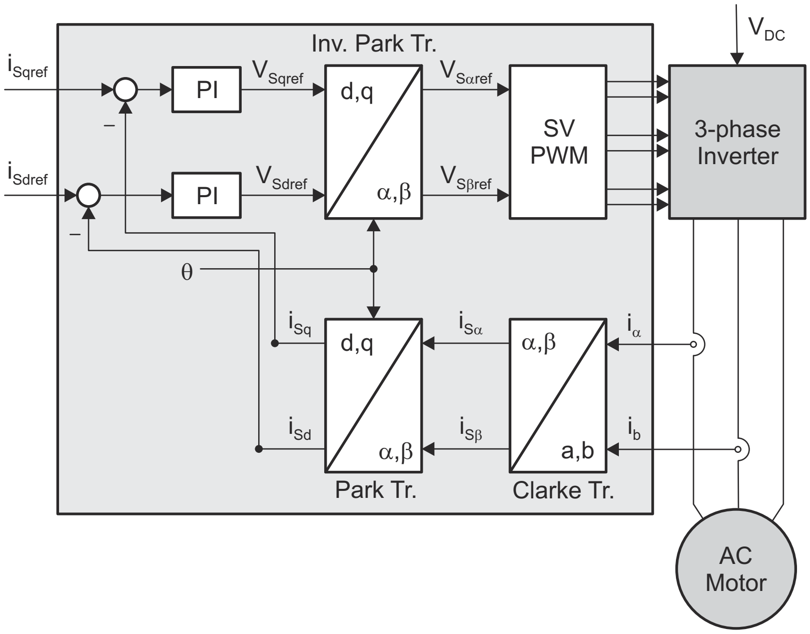


Figure 6-1. Sensorless Field-Oriented Control Using Two Current Sensors

Angle estimation is implemented using an Enhanced Sliding Mode Observer (eSMO) with phased lock-loop (PLL) to accurately estimate the real-time speed of a BLDC/PMSM motor. The motor state of the algorithm is controlled by using a state variable to automatically transition the motor from startup to open loop, and from open loop to closed loop. The closed-loop state allows for real-time motor control with minimal reference current, and additionally provides tunable speed regulation across torque changes or load conditions.

6.2 Software Architecture

As shown in Figure 6-2, this software package uses the MSPM0 System Configuration tool to initialize the necessary peripherals for sensorless FOC and a HAL layer to interface with a variety of 3-phase motor drivers and gate driver designs.

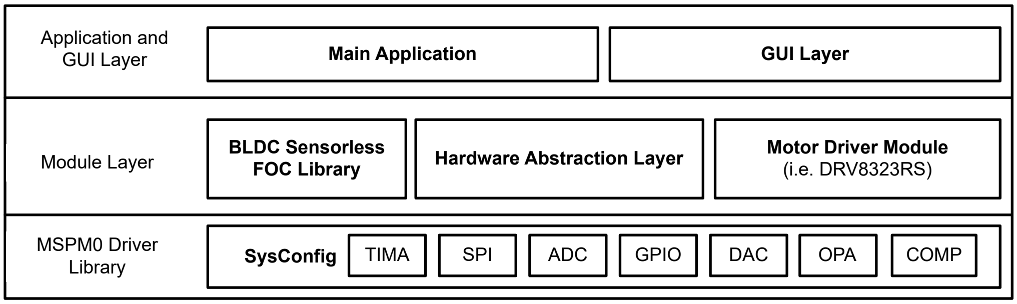

Application and GUI Layer
Main Application
GUI Layer
Module Layer
BLDC Sensorless FOC Library
Hardware Abstraction Layer
Motor Driver Module (i.e. DRV8323RS)
MSPM0 Driver Library
SysConfig
TIMA
SPI
ADC
GPIO
DAC
OPA
COMP

Figure 6-2. Software Architecture for Sensorless FOC Library With MSPM0

6.3 Block Diagrams

The are two topologies for sensorless FOC depending on whether the analog signal chain required for phase current feedback is integrated in the MSPM0Gxxx MCU or external in the system.

6.3.1 MSPM0Gx10x and Gate Driver with Analog/MOSFET Integration

In lower voltage FOC applications, many gate driver or motor driver devices integrate up to three current sense amplifiers with programmable gain, which offloads analog requirements from the MSPM0Gx device. MSPM0Gx10x devices without analog integration are available in packages as small as VSSOP-20 and 24-VQFN to reduce system size and provide low cost. MSPM0 can accurately sense the motor phase voltage, bus voltage, current, and speed quickly to provide feedback to the FOC algorithm using 12-bit simultaneous sampling 4-Mbps ADCs. This topology is designed for FOC applications that are low-cost and small form factor, such as pumps, fans, blowers, and small appliances.

As shown in Figure 6-3, the signals used this system topology are:

- 6 PWM signals with adjustable deadband (6x PWM)
- Logic-low fault signal from driver (nFAULT)
- ADC current sense feedback from integrated or external current sense amplifiers to measure motor phase current, and bus sense voltage
- Optional SPI read/write interface (for drivers with SPI interface)

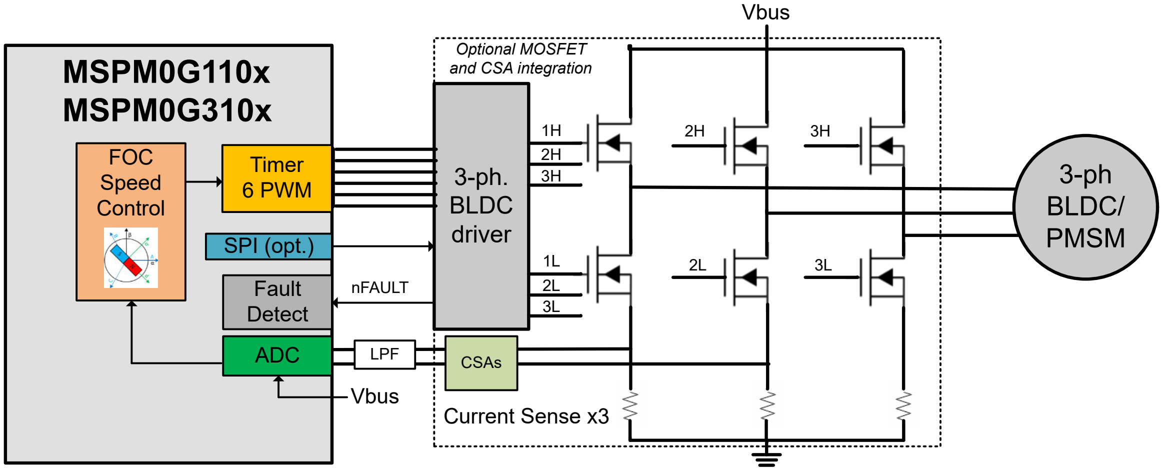


Figure 6-3. Block Diagram for Sensorless FOC for BLDC Motors using MSPM0Gx10x and 3-Phase Gate or Motor Driver with Optional CSA and MOSFET Integration

6.3.2 MSPM0Gx50x Analog Integration and Gate Driver

With the help of scalable analog integration, the MSPM0Gx50x devices can accurately sense the motor phase currents and voltage, compute the real-time motor speed and error calculations, and output space-vector generated PWM signals to close the speed loop. Two programmable gain amplifiers (PGAs) amplify the difference between phase currents sensed through two shunt resistors and a scaled DAC output voltage, and the output of the PGAs directly can be sampled by internal ADCs. Additionally, the MSPM0G350x family of devices provide a hardware math accelerator (MATHACL) that increases the computational throughput of math functions required in FOC transformations, such as multiply, divide, arctangent, sine, and cosine.

This topology provides a low-cost alternative to using a discrete analog signal chain and is designed for BLDC, PMSM, and ACIM motors that require high voltages and high efficiency. Some examples of applications include servo drives, HVAC motors, and large appliances.

As shown in Figure 6-4, the signals used this system topology are:

- 6 PWM signals with adjustable deadband (6x PWM)
- Logic-low fault signal from driver (nFAULT)
- Two integrated Opamps with programmable gain in inverting- or non-inverting configuration
- ADC current sense from output of integrated Opamps and bus sense voltage
- 12-bit DAC reference for external current sense signal biasing
- Optional bus overcurrent protection using integrated low-side comparator

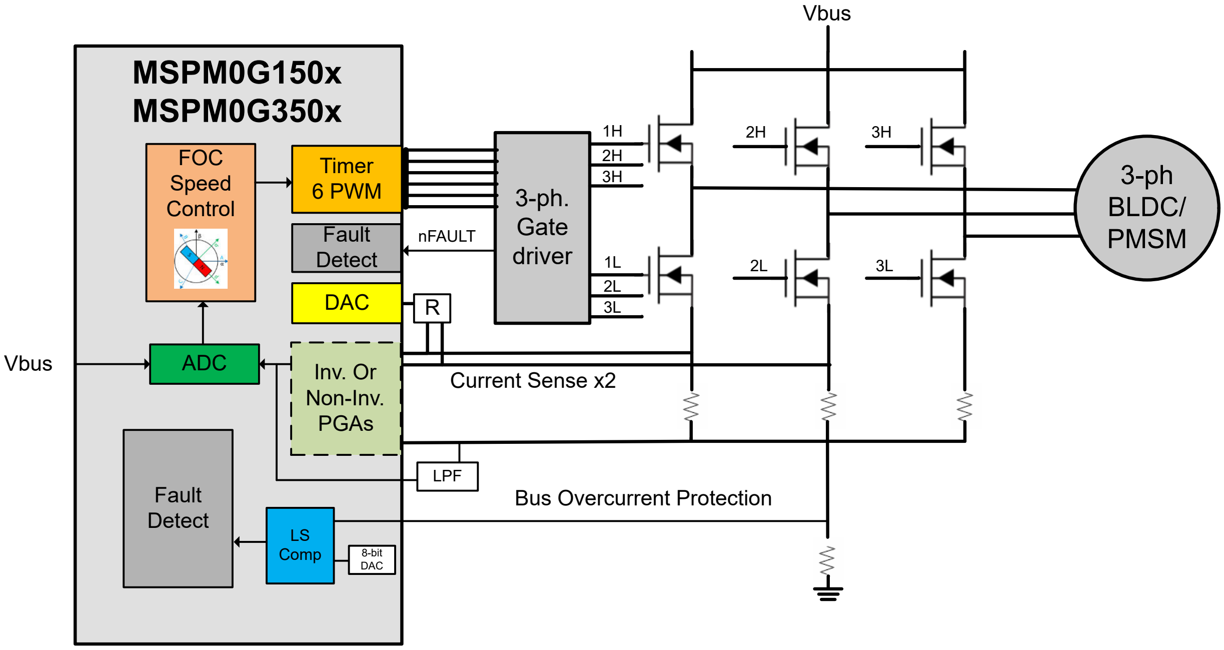


Figure 6-4. Block Diagram for Sensorless FOC for BLDC/PMSM/ACIM Motors using MSPM0Gx50x with Analog Integration and 3-Phase Gate Driver Powerstage

6.4 Hardware Support

Table 6-1 shows the hardware supported and tested motors with MSPM0-SDK for 3-phase Sensorless FOC motor control.

Table 6-1. Supported Hardware for Sensorless FOC using MSPM0

LaunchPad	Motor Driver Hardware	Hardware User Guide	Current Sense Amplifiers	SPI Driver Support	Motor Voltage Range (rec.)	Recommended Motor Power
LP-MSPM0G3507	BOOSTXL-DRV8323RS	BOOSTXL-DRV8323RS	3	Yes	6-60V	< 1000 W

6.5 Software Support

Table 3-1 shows the software supported for Sensorless FOC control in TI Resource Explorer.

Table 6-2. Software Support for Stepper Control

Library Software Overview	Software User Guide	Code Example(s)	GUI(s)	GUI User Guide
Sensorless FOC library overview	Sensorless FOC software user guide	DRV8323RS	DRV8323RS	DRV8323RS

6.6 Evaluating Sensorless FOC with MSP Motor Control

For more information, visit the Sensorless FOC Control User's Guide in TI Resource Explorer.

To spin a BLDC or PMSM motor with sensorless FOC using the MSPM0, follow these four simple steps:

1. Order a supported LaunchPad and DRV8xxx EVM. Connect the hardware and motor to the Hardware User's Guide for the specified EVM.
2. Obtain the motor's phase resistance (Rs), phase inductance (Ls), number of poles, and maximum speed (RPM).
3. Launch the specified GUI from the TI Gallery. Specify the motor parameters, enable the driver, spin the motor, and evaluate device features using the MSPM0 MCU.
4. Evaluate the firmware in Code Composer Studio or on the CCS Cloud IDE.

6.7 Sensorless FOC Performance

Table 6-3 and Table 6-4 shows the 3-phase sensorless FOC design performance with the following motors tested.

Sensorless FOC Design Specifications

- PWM frequency = FOC = 10 kHz – Sensorless
- Compiler - Code Composer Studio v. 12.0 (CCS)

Table 6-3. Sensorless FOC Design Specifications

MCU	CPU Clock (MHz)	Cortex	Current sensing	Total workload	Total code (Kb)	Total SRAM (kB)
MSPM0G	80	M0	2 Shunt	55%	12.8	1.64

Motor Performance

- BOOSTXL-DRV8323RS, VM = 24 V
- Math Accelerator (MATHACL) enabled
- PWM frequency = FOC loop frequency

Table 6-4. Sensorless FOC Motor Performance

	LVSERVOMTR	LVBLDCMTR
Closed loop speed range (RPM)	200 – 4500 RPM	280 – 5830 RPM
Max electrical frequency range (kHz)	265 Hz	389.5 Hz
Speed accuracy error	<1%	<1%
PWM frequency (kHz)	15 kHz	15 kHz
Maximum power (W)	approximately100-400 W	approximately100-400 W

7 References

- Texas Instruments, *Sensorless Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors*

Revision History

Changes from Revision A (July 2023) to Revision B (October 2023) — Page

- Updated Section 1 3
- Updated Section 2 4
- Updated Section 3.4 8
- Added Section 3.5 8
- Updated Section 4.4 12
- Added Section 4.5 12
- Updated Section 5.4 15
- Added Section 5.5 15
- Updated Section 6.1 16
- Updated Section 6.4 18
- Added Section 6.5 19

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265