Application Note

MSPM33 Motor Control

Xiaodong Li, Ella Tang

ABSTRACT

With TI's new MSPM33 MCU portfolio optimized for motor control applications, MSP Motor Control is a one-stop shop for spinning a stepper or 3-phase motor with an MSPM33 MCU.

Table of Contents

List of Figures

List of Tables

Trademarks

Launchpad™ and Code Composer Studio™ are trademarks of Texas Instruments.

All trademarks are the property of their respective owners.

1 Introduction

MSPM33 Motor Control is a middleware package integrated within MSPM33-SDK and to allow users to spin motors in short time using small-scale, simplified MSPM33 firmware examples with popular motor driver designs and topologies. Table 1-1 shows the types of motors and motor control algorithms that are supported for industrial and automotive applications.

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

Motor Control Type	MSPM33 Launchpad™	Driver IC	Interface
Stepper FOC	LP-MSPM33C321A	Stepper Gate Driver	• PWM (4x) • Interposer board
Brushless-DC Sensored Trapezoidal	LP-MSPM33C321A	3-Phase BLDC Gate Driver	• PWM (6x) • Interposer board
BLDC / PMSM / ACIM Sensorless Field-Oriented Control	LP-MSPM33C321A	3-Phase BLDC Gate Driver	• PWM (6x) • Interposer board

MSP Motor Control provides an abundance of resources inside MSPM33-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)
- Supported interposer board that sits between MSPM33 LP and DRV EVM with targeted DRV EVM compatibility
- 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

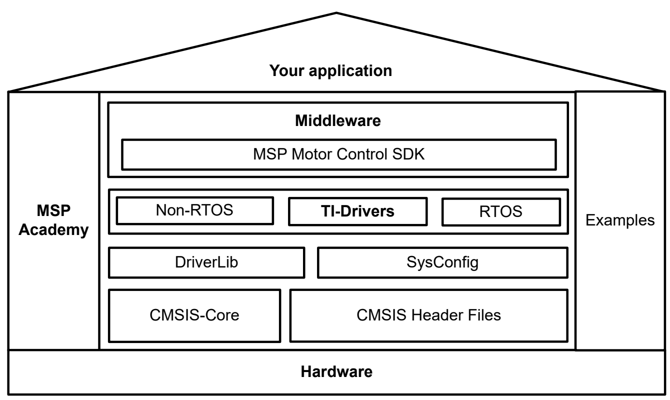


Figure 1-1. MSP Motor Control with MSPM33-SDK Development Resources

As shown in Figure 1-1, MSP Motor Control is embedded within MSPM33-SDK as a middleware package that can interface with Non-RTOS and RTOS drivers, MSPM33 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 designs. This provides a simple-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™ Theia v20.4.0 or newer
- MSPM33-SDK V1.00.00 or newer
- Support for IAR Systems and Keil embedded development tools

2 Getting Started With MSPM33 Motor Control

To address high-performance and high-efficiency motor control applications, the MSPM33 platform is designed to support three primary motor control types as shown in Table 1-1. At the current development stage, the Stepper FOC design is fully implemented and verified, while the remaining two types are under active research and development.

The verified Stepper FOC solution is executed on a dedicated hardware evaluation platform, based on MSPM33 LaunchPad, DRV8962 motor driver EVM, current amplifier and magnetic encoder. The software architecture implements Motor Sensored Vector Control. By utilizing real-time phase current feedback and rotor position tracking, the system decouples the magnetic flux and torque of the motor. This method allows the system to achieve real-time, closed-loop FOC for stepper motors.

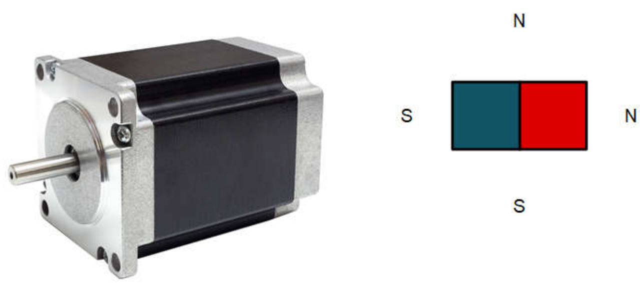
3 Stepper Motor Control

The main features of the MSPM33 stepper motor control designs include:

- 4-PWM stepper control up to 20kHz using FOC or sine-wave commutation
- Fully closed-loop execution for velocity and position tracking based on real-time feedback from a magnetic encoder
- Inline phase current sensing using external current sense amplifiers
- Low-power standby management using sleep pin
- Fault detection using fault pin

3.1 Background

Stepper motor can 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 3-1. 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.

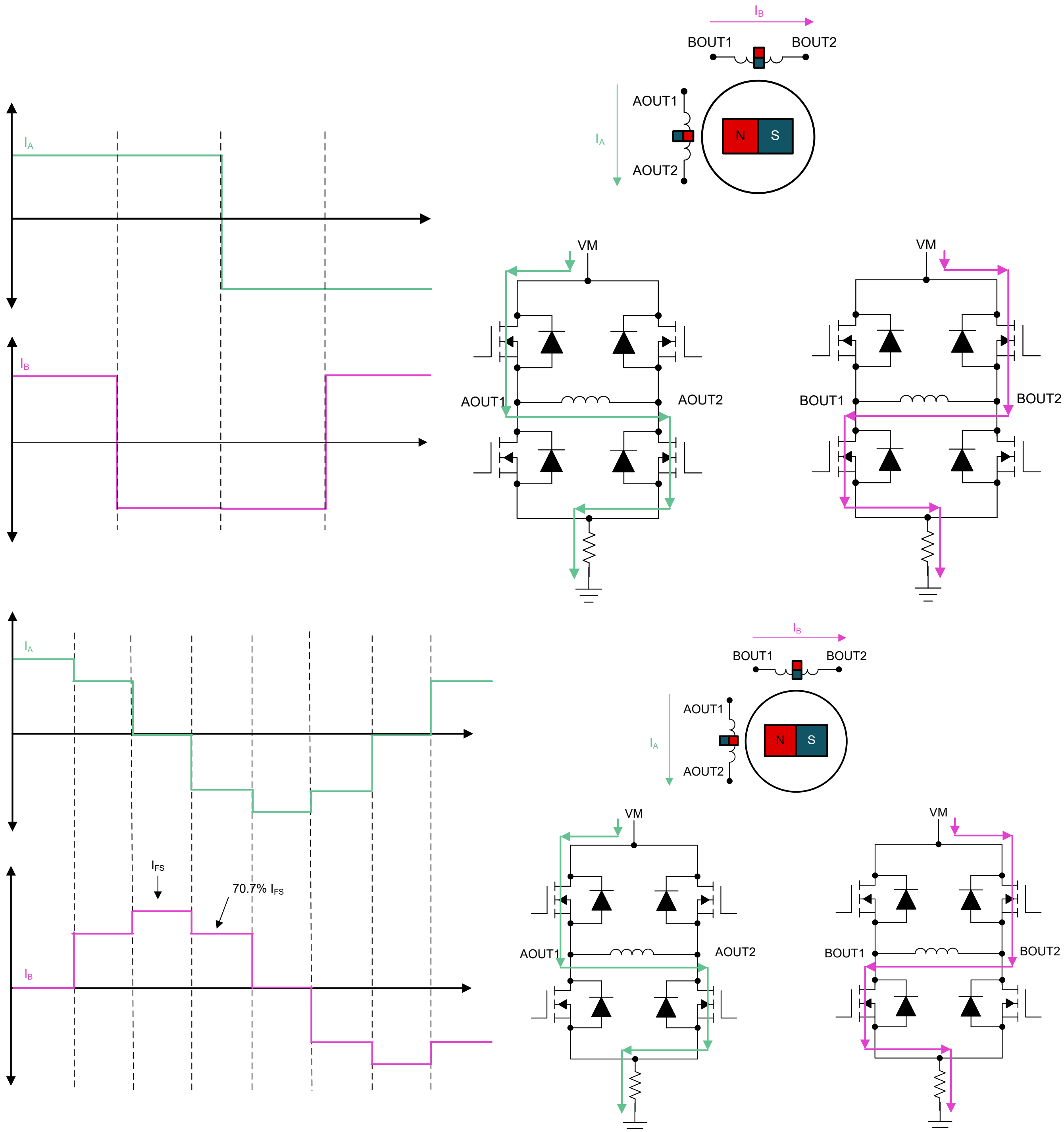


Figure 3-1. Full and Circular-Half Stepping in Stepper Motor Control

3.2 Software Architecture

The software architecture for this stepper motor control design features a modular, layered design. This framework decouples high-level application logic from low-level hardware dependencies, verifying excellent code portability and scalability. The software package is systematically divided into three functional layers, as shown in Figure 3-2:

Application and GUI Layer: This part manages the main application control logic and provides communication interfaces for Host/GUI data exchange.

Module Layer: This part Includes Hardware Abstraction Layer and Stepper Motor Library.

MSPM33 Drive Library Layer: Based on the low-level firmware library of MCU, this part implements direct initialization and driving of hardware peripherals, including timers, high-speed Analog-to-Digital Converters and General-Purpose Input/Outputs.

- Quadrature encoder signals or SPI interface from the magnetic encoder to feed back real-time motor rotor angle and position (ABZ / SPI)

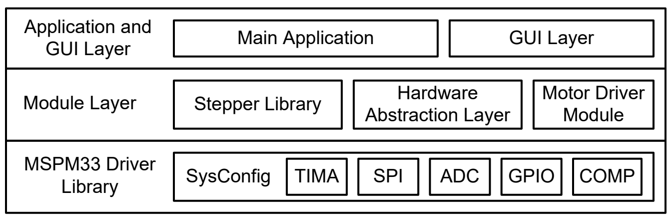


Figure 3-2. Software Architecture for Stepper Library with MSPM33

3.3 Block Diagrams

The system hardware architecture is built upon MSPM33C321A and DRV8962 four-channel half-bridge driver, forming a highly integrated, closed-loop field-oriented control (FOC) stepper motor drive system.

As shown in Figure 3-3, the interconnected signals and topology are described as follows:

- 4 PWM signals with center-aligned or edge-aligned synchronization (PWM 4x)
- 12-bit DAC reference voltage for hardware current regulation (VREF)
- Low-power mode signal for drivers with sleep mode pin (nSLEEP)
- Logic-low fault signal from driver (nFAULT)
- High-precision analog current feedback from external INA240A1 current sense amplifiers into the MCU high-speed ADC (Current)
- Quadrature encoder signals or SPI interface from the magnetic encoder to feed back real-time motor rotor angle and position (ABZ / SPI)

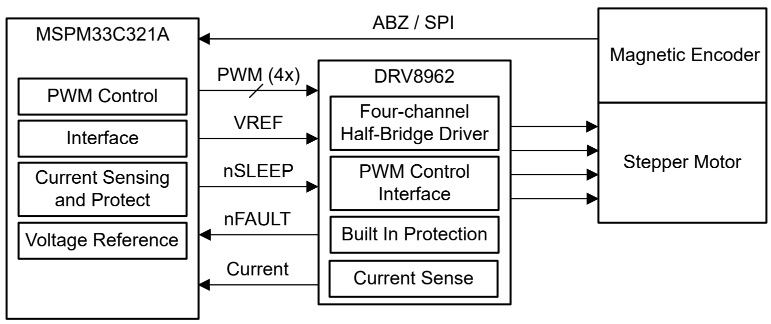


Figure 3-3. Block Diagram for 4-PWM Interface for Stepper Motor Control

3.4 Hardware Support

Table 3-1 lists the hardware supported with the MSPM33-SDK for Stepper motor control.

Table 3-1. Supported Hardware for Stepper Control with MSPM33

LaunchPad	Motor Driver Hardware	Control Interface	Current Sense	SPI Interface	Motor Voltage Range	Motor Current (Peak)	Motor Power
LP-MSPM33C321A	DRV8962EVM	PWM (4)	4	Yes	4.5V-65V	5A	<300W

4 BLDC Sensored Trap Control

The main features of the MSPM33 BLDC sensored trap control designs include:

- Six PWM control with complimentary outputs, adjustable dead band with nanosecond-level resolution, and PWM frequency up to 200kHz
- 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 and undervoltage detection
- Hardware fault brake and cycle-by cycle overcurrent protection

4.1 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 4-1.

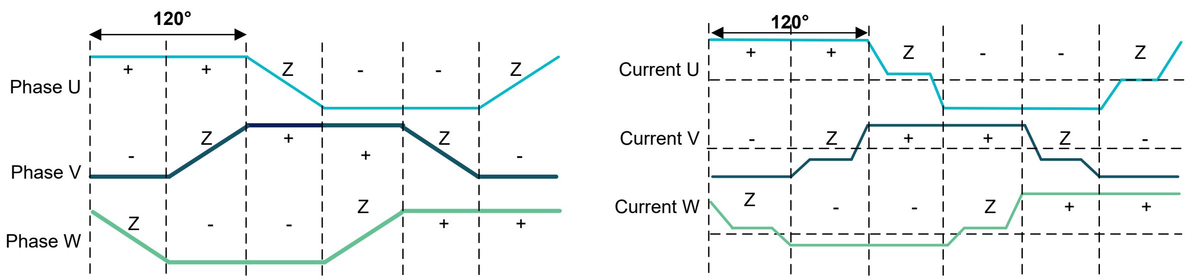


Figure 4-1. Trapezoidal Commutation Phase Voltage and Current Waveforms

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 4-2. 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.

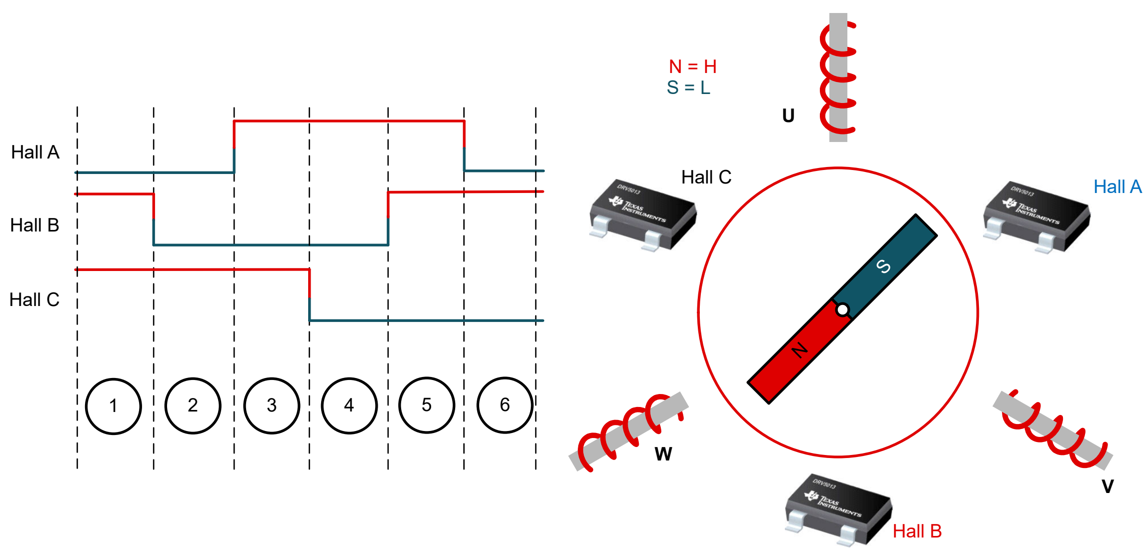


Figure 4-2. Hall Sensor A/B/C Waveforms for Six-Step Commutation

4.2 Software Architecture

As shown in Figure 4-3, this software package uses the MSPM33 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.

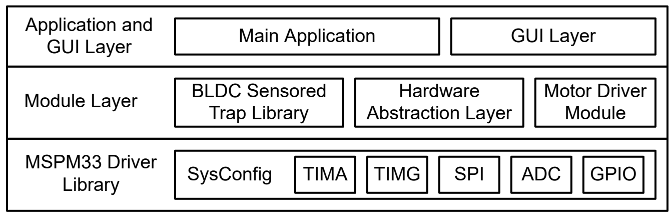


Figure 4-3. Software Architecture for BLDC Sensored Trap Library with MSPM33

4.3 Block Diagrams

The block diagram for MSPM33C321A MCU and a 3-phase gate/motor drivers showcases supported in the MSP Motor Control is shown in Figure 4-4. 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)

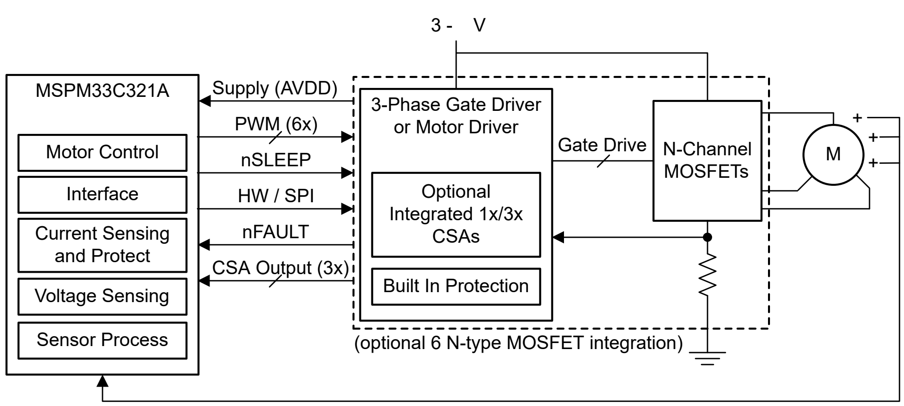


Figure 4-4. Block Diagram for 6-PWM Hall-Sensored Trapezoidal Control for BLDC Motors Using MSPM33C321A and 3-Phase Gate or Motor Driver

4.4 Hardware Support

Table 4-1 shows the hardware supported with the MSPM33-SDK for BLDC sensored trap motor control.

Table 4-1. Supported Hardware for BLDC Sensored Trap Control with MSPM33

LaunchPad	Supported Motor Driver Hardware	Current Sense Amplifiers	SPI Driver Support	Motor Voltage Range (rec.)	Recommended Motor Power
LP-MSPM33C321A3	DRV8316REVM	3	Yes	4.5V-40V	< 75W
	DRV8334EVM	3	Yes	12V-24V	< 300W
	DRV8363-Q1AEVM	3	Yes	8V-85V	< 500W
	DRV7308EVM	0-3	Yes	15V-450V	< 250W

5 3-Phase Sensorless FOC Control

The main features of the MSPM33 sensorless FOC design includes:

- 2-shunt or 3-shunt based current sense design
- Enhanced sliding mode observer and 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 100kHz) and deadband

5.1 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 5-1.

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

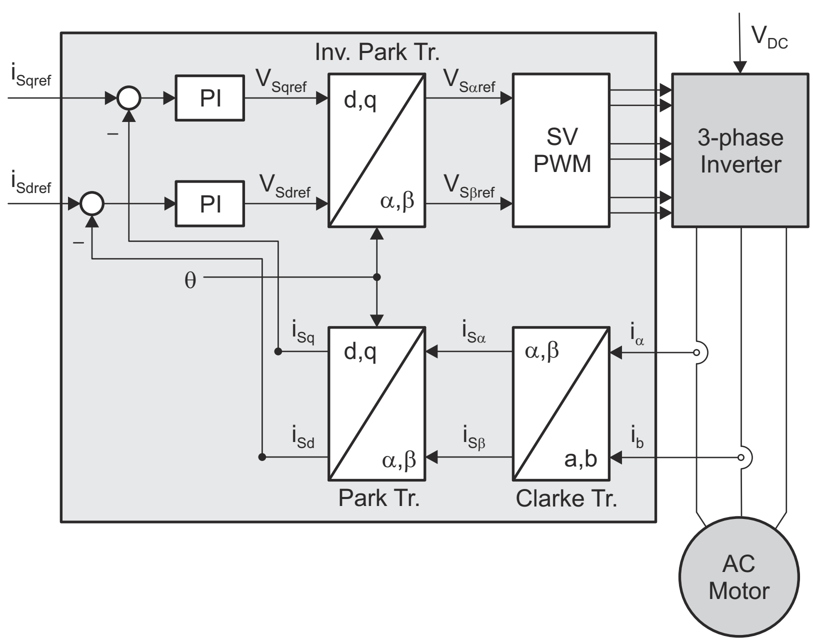


Figure 5-1. Sensorless Field-Oriented Control Using Two Current Sensors

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.

5.2 Software Architecture

As shown in Figure 5-2, this software package uses the MSPM33 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.

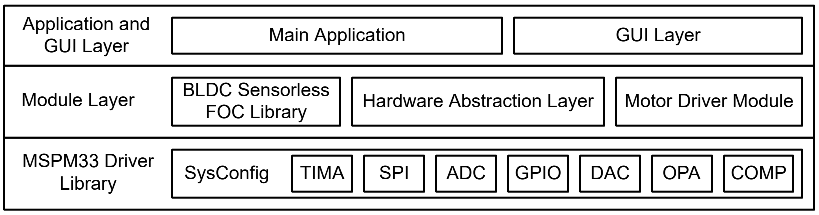


Figure 5-2. Software Architecture for Sensorless FOC Library With MSPM33

5.3 Block Diagrams

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 MSPM33 device. MSPM33 devices without analog integration are available in packages as small as 48-VQFN and 64-LQFP to reduce system size and provide low cost. MSPM33 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-MSPS 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 5-3, the signals used this system topology are:

- Six 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)

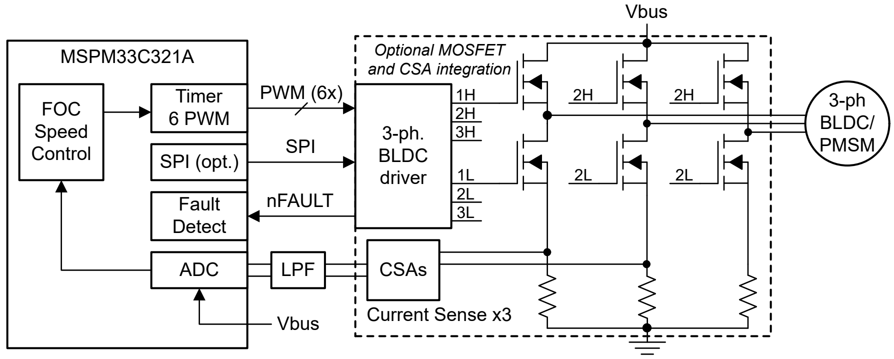


Figure 5-3. Block Diagram for Sensorless FOC for BLDC Motors using MSPM33C321A and 3-Phase Gate or Motor Driver with Optional CSA and MOSFET Integration

5.4 Hardware Support

Table 5-1 shows the hardware supported and tested motors with MSPM33-SDK for 3-phase Sensorless FOC motor control.

Table 5-1. Supported Hardware for Sensorless FOC using MSPM33

LaunchPad	Supported Motor Driver Hardware	Current Sense Amplifiers	SPI Driver Support	Motor Voltage Range (rec.)	Recommended Motor Power
LP-MSPM33C321A3	DRV8316REVM	3	Yes	4.5 V-40 V	< 75 W
	DRV8334EVM	3	Yes	12 V-24 V	< 300 W
	DRV8363-Q1AEVM	3	Yes	8 V-85 V	< 500 W
	DRV7308EVM	0-3	Yes	15 V-450 V	< 250W

6 Summary

This application note provides an overview of the motor control middleware package among MSPM33 family and highlights technical designs for stepper, BLDC sensored trap and 3-phase sensorless FOC control. In addition to basic knowledge, the document also lists references and further reading materials for users.

7 References

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