AN202488

FM4 MB9BF568x Series MCU – PMSM Servo Motor Speed Control

Author: Arthur Zhong

Associated Part Family: FM4

Associated Code Examples: Servo Motor Speed Control Firmware

Related Application Notes: None

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN202488.

AN202488 explains the permanent magnet synchronous motor (PMSM) control theory and the system scope, hardware design, software design, and test results of the servo motor speed control solution. A code example using low-voltage motor control starter-kit is included to demonstrate the solution.

Contents

1 Introduction....1
2 PMSM Control Theory....1
2.1 Structure of a 3-Phase PMSM....1
2.2 Field-Oriented Control Principle....2
2.3 FOC Structure....3
2.4 Incremental Encoder....4
3 System Scope....5
4 Hardware Design....6
5 Software Design....6
5.1 Firmware File Structure....6
5.2 Control Implementation....7
6 Test Results....10
6.1 Current Waveform....12
6.2 Acceleration and Deceleration....13
7 Summary....14
Document History....15
Worldwide Sales and Design Support....16

1 Introduction

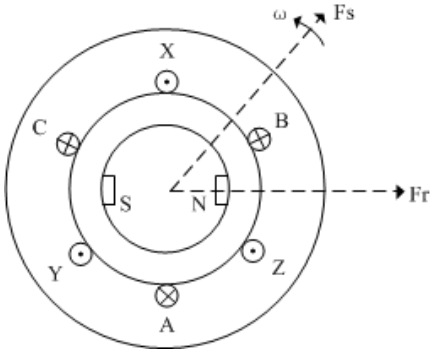
This document describes PMSM servo motor speed control based on the FM4 MB9BF568x Series MCU, including the system scope, hardware design, software design, and test results.

2 PMSM Control Theory

2.1 Structure of a 3-Phase PMSM

A 3-phase PMSM has two parts: the stator and the rotor, as shown in Figure 1. At the stator side, the 3-phase windings are coiled on the stator core. Each winding is separated from the others by 120 degrees to generate a rotating magnetic field (Fs) when a 3-phase AC current traverses the 3-phase windings. The 3-phase winding separated by 120 degrees is referred to as “3-phase symmetric.”

Figure 1. Structure of 3-Phase PMSM

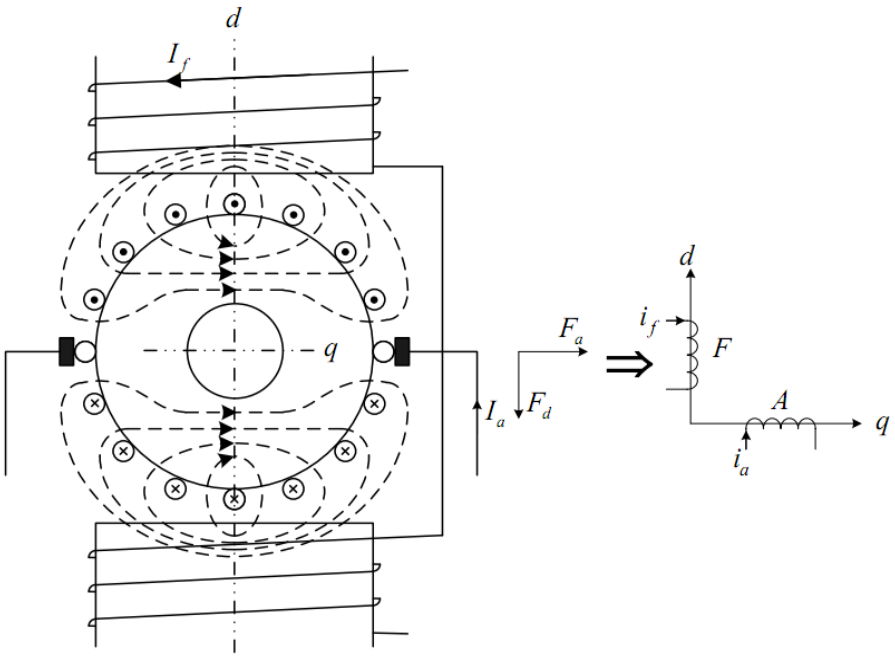


At the rotor side, one or more pairs of permanent magnetic poles are mounted to offer a constant rotor magnetic field (Fr). Because Fs is a rotating magnetic field, Fr will be dragged and follow Fs. If Fr cannot catch up with Fs, the rotor will not rotate continuously. If the 3-phase current in the 3-phase windings disappears, Fs will disappear at the same time, and the rotor will stop.

2.2 Field-Oriented Control Principle

The brushless DC (BLDC) motor is a conventional DC motor whose torque control and magnetizing control are decoupled, making it easy to control. Figure 2 shows the decoupled control.

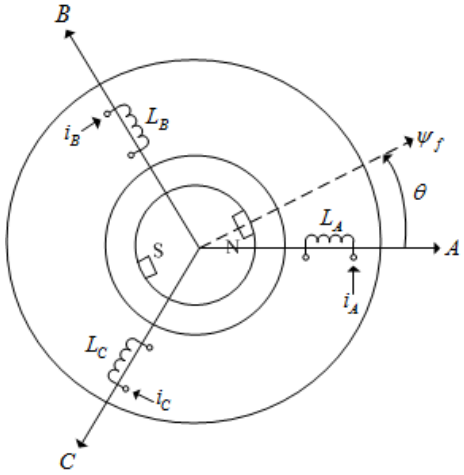
Figure 2. BLDC Motor Decoupled Control



Magnetizing is controlled by the magnetizing current (I_f), and the torque is controlled by the torque current (I_a). The direction of the magnetizing magnetic field is parallel to the d-axis (vertical direction), and the direction of the torque magnetic field is parallel to the q-axis (horizontal direction). So these two magnetic fields do not influence each other. That is to say, they are decoupled, so the motor's magnetizing and torque can be adjusted individually. For example, the torque control formula is $T_e = C_m \emptyset I_a$, which means that the torque is controlled only by the torque current I_a.

PMSM control is much more complex than BLDC motor control. The magnetic field of a 3-phase symmetric winding is a coupled magnetic field. The following torque control formula reveals the complex coupled relationship, illustrated in Figure 3.

Figure 3. Coupled Magnetic Flux of PMSM



$$T_e = \frac{1}{2} n_p [I_{ABC}]^T \frac{\partial [L_{ABC}]}{\partial \theta} [I_{ABC}]$$

$$[L_{ABC}] = \begin{bmatrix} L_A & M_{AB} & M_{AC} \\ M_{BA} & L_B & M_{BC} \\ M_{CA} & M_{CB} & L_C \end{bmatrix} (\text{M is mutual inductance}), [I_{ABC}] = \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix}$$

From the expression of Te, it is easy to see that the torque is determined by all 3-phase inductances (including self-inductance and mutual inductance) and currents. Obviously, torque control is more complex than in a BLDC motor.

Coordinate transformation is the way to simplify PMSM torque control. By coordinate transformation, a PMSM control model is converted from an A-B-C coordinate to a d-q coordinate. The torque control formula is also converted into a d-q coordinate, according to the following formula:

$$T_e = \frac{3}{2} n_p \psi_d I_q$$

The simple formula in the d-q coordinate makes torque control of the PMSM as easy as that of the BLDC motor.

2.3 FOC Structure

As explained previously, the advantage of FOC is that it makes torque control of the PMSM as easy as that of a BLDC motor through motor rotor magnetic field orientation technology. In this technology, the coordinate transformation method turns the motor module from the u-v-w coordinate to the rotational d-q coordinate, and the d-q coordinate rotational speed is the same as the stator magnetic field rotational speed. Thus, the control of a PMSM is simplified, and the control performance is almost the same as that of a BLDC motor.

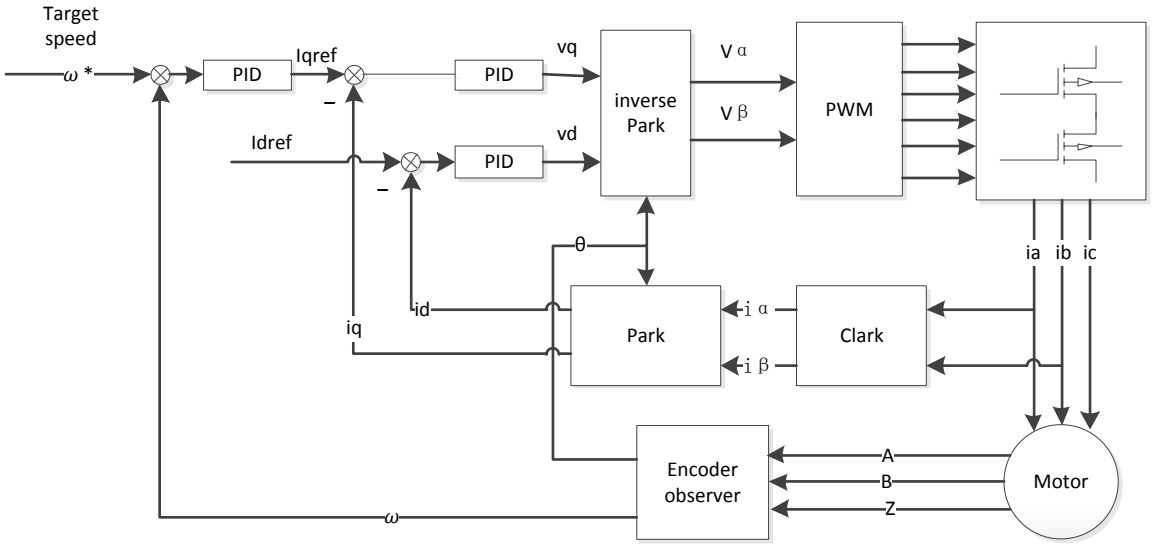
Some proportional-integral-derivative (PID) regulators are added to adjust the motor output according to the given input. By setting different PID parameters, the system achieves a different dynamic and static performance.

Space vector pulse width modulation (SVPWM) technology is applied to accept the driving voltage in an α-β coordinate and to output a set of switching instructions to control the six switches in the full-bridge inverter.

A position and speed estimator is used to observe the real-time motor speed resulting from the motor-driving voltage and current. The estimated motor speed is compared with the expected speed, and the compare result acts as the input of the speed PID regulator. The estimated rotor position angle is used by the coordinate transformation unit.

Figure 4 shows the control structure.

Figure 4. FOC Diagram

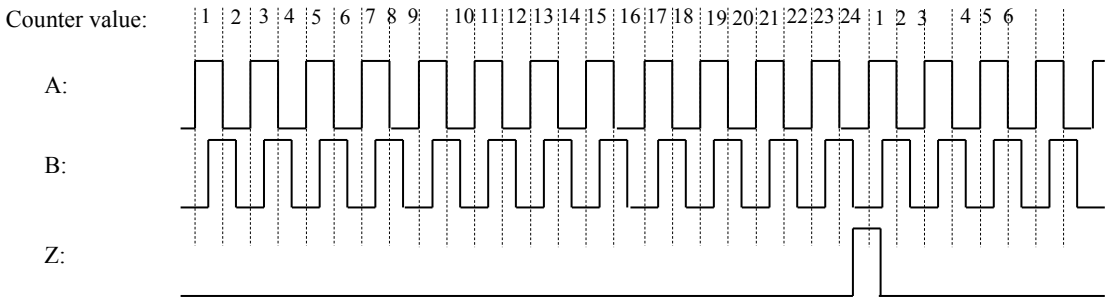


2.4 Incremental Encoder

A quadrature position encoder contains two types of signals: quadrature-phase signal A and B, and zero-match signal Z, as shown in Figure 5. The Z-signal is generated every mechanical cycle. It is used to calculate the cycles of rotation or the correct estimated motor position.

By integrating the A or B pulses, the motor position is estimated. The encoder peripheral is integrated with the MB9BF568R MCU to count the A or B signal. A and B pulses indicate the position of the rotor, and their pulse frequency is related to the precision of position estimation.

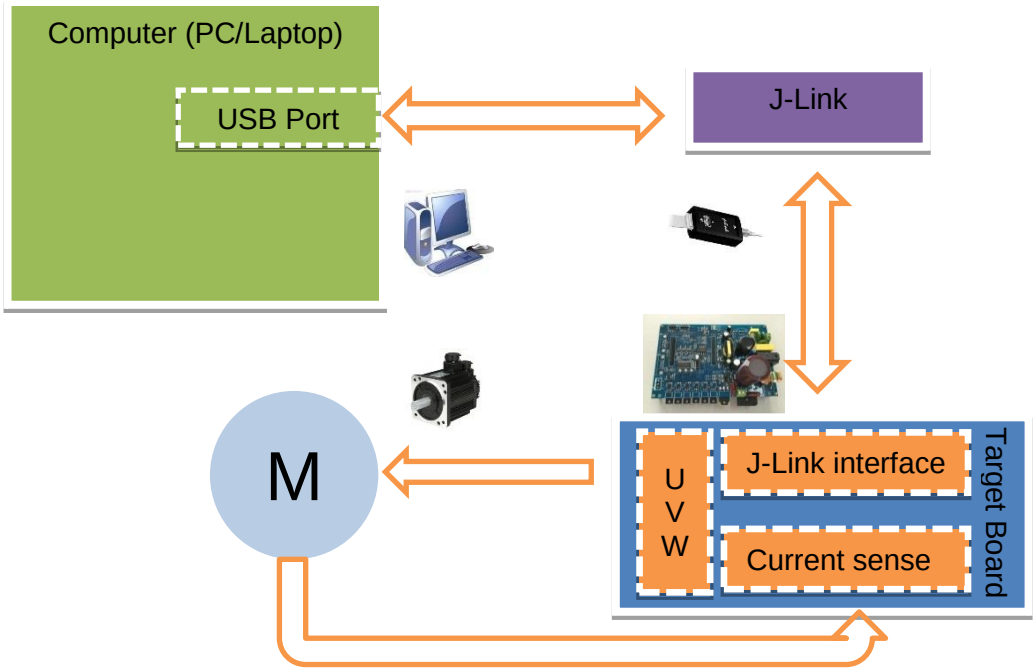
Figure 5. Encoder Signal



3 System Scope

This section describes the driving system of a 3-phase PMSM. Figure 6 shows the system structure and driving performance.

Figure 6. System Structure



- MB9BF568x is the target controller with a configured 160-MHz main clock and 80-MHz bus clock.
- Motor combines an incremental encoder with 360 PWM pulses per cycle.
- System specifications:
  - Three-phase Hall current sensor for sampling
  - Full high- and low-voltage supplier separation
  - Auto Z-signal position detection
  - Wide speed range: 100 rpm ~ 3500 rpm
  - Rapid acceleration within 200 ms from 0 rpm to 3500 rpm
  - Rapid deceleration within 300 ms from 3500 rpm to 0 rpm
  - Field-weaken function is not implemented in this system.
  - Accurate speed controlling with less than 1 percent target error
  - Bidirectional rotation
- Firmware development environment
  - Windows XP or above
  - IAR Embedded Workbench 7.3

4 Hardware Design

The hardware design for servo motor control is different from that of typical motor control hardware. The servo motor is applied to industrial control. For more information about hardware, refer to the hardware design application note.

The following are some hardware specifications:

- AC-DC power supply
- Three-shunt current sample
- Support for J-Link connection
- Combined Hall sensor interface (HA, HB, HC, 5V, GND)
- Combined encoder interface (AIN, BIN, ZIN, 5V, GND)
- Interior permanent magnet (IPM) motor drive

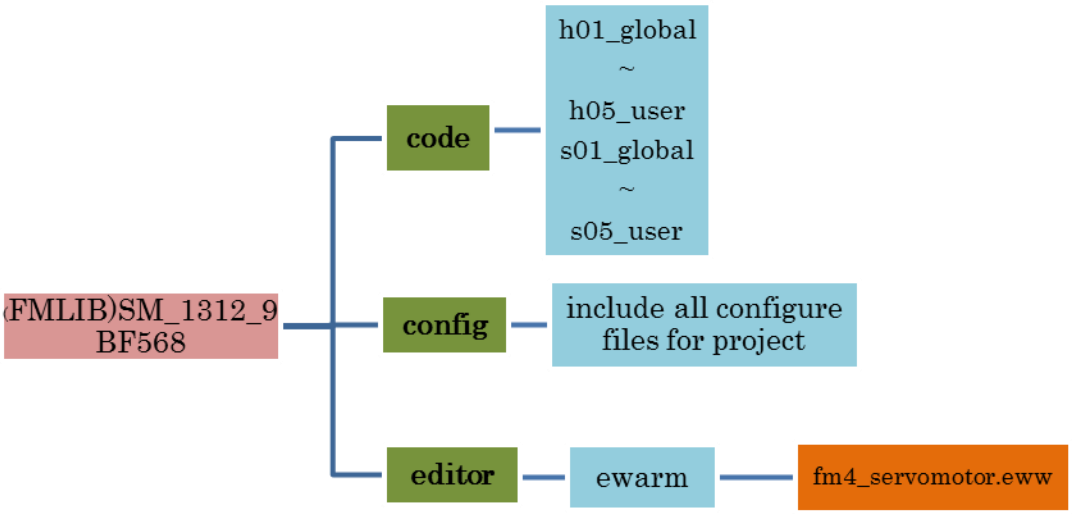
5 Software Design

This section describes servo motor speed control implementation and addresses firmware version, firmware structure, and control process.

5.1 Firmware File Structure

Figure 7 shows the firmware file structure.

Figure 7. Firmware File Structure



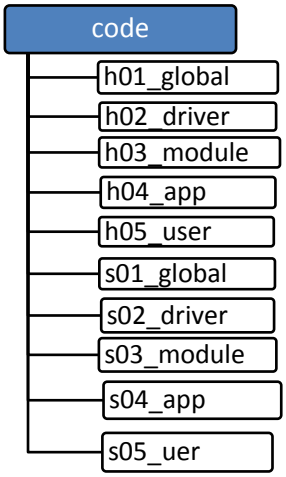
The firmware contains three subfolders: code, config, and editor. All source code is stored in the code folder, including header files and C source files. Configuration and MCU description files are stored in the config folder.

Double-click on the *FM4_ServoMotor.eww* file to open the project.

Code Folder

Source codes are divided into five types by function and stored in five different folders named “global,” “driver,” “module,” “app,” and “user” respectively, as shown in Figure 8.

Figure 8. Structure in Code Folder



- Global layer is empty.
- Driver layer stores the MCU header file and macro definition file.
- Module layer stores independent functions.
- App layer is related to the actual project. The function in this folder can be changed depending on the system.
- User layer is open for configuration and debugging.

5.2 Control Implementation

This section explains the peripherals and interrupts used in the firmware and the control process flow.

5.2.1 Firmware Peripherals

All peripherals used in the firmware are configured in *init_mcu.c*, stored in the "s05_user" folder. For more details on peripheral initialization, refer to the MCU datasheets. Table 1 lists the functions of each peripheral.

Table 1. Peripheral Functions

Peripheral	Function
Clock	Configure system main clock and bus clock.
Nested vectored interrupt controller (NVIC)	Enable or disable interrupts, configure priorities.
QPRC	Count encoder signal pulses to detect motor position.
Base timer	Measure the width of the encoder signal to calculate motor current speed.
ADC	Sample phase current; ADC unit 0 is used.
Multifunction timer(MFT)	Generate PWM signals to control three half-bridges to drive motor running; MFT unit 0 is used.
Watchdog	Reset MCU when program goes wrong.

- Clock settings
  - SCM_CTL: System clock mode control
  - BSC_PSR: Base clock mode control
  - APBC0_PSR: APB0 prescaler register
  - APBC1_PSR: APB1 prescaler register
  - APBC2_PSR: APB2 prescaler register
- NVIC settings
  - NVIC_SetPriority(IRQn, x): Priority setting*
  - NVIC_EnableIRQ(IRQn): Enable priority
  - IRQn: IRQ number
  - X: Priority number

- QPRC settings
  - PC_Mode2 and RC_Mode0 is selected.
  - QCR: QPCR control register
  - QICRL: QPCR interrupt control register
- Base timer settings
  - PWC function is selected in this firmware.
  - TMCR: Timer control register
  - STC: Status control register
  - DTBF: Data buffer control
- ADC settings
  - Scan interrupt is enabled in this firmware; priority mode interrupt is not used.
  - ADCR: AD control register
  - ADSR: AD status register
  - SCCR: Scan conversion control
- MFT settings
  - FRT, OCU, WFG, and ADCMP are used in this firmware.
  - FRT selects up and down count mode. Complementary output of WFG with dead time is selected.
  - For configuration details, refer to the MCU datasheet.
- Watchdog settings
  - WdogControl: Software watchdog timer control register
  - WDG_CTL: Hardware watchdog timer control register

*For priority setting, the lower the digit, the higher the priority. For example:

- NVIC_SetPriority(ADC0_IRQn,1)
- NVIC_SetPriority(FRT0_ZERO_IRQn,2)

ADC0_IRQn priority is higher than FRT_ZERO_IRQn through this setting.

5.2.2 Interrupts in Firmware

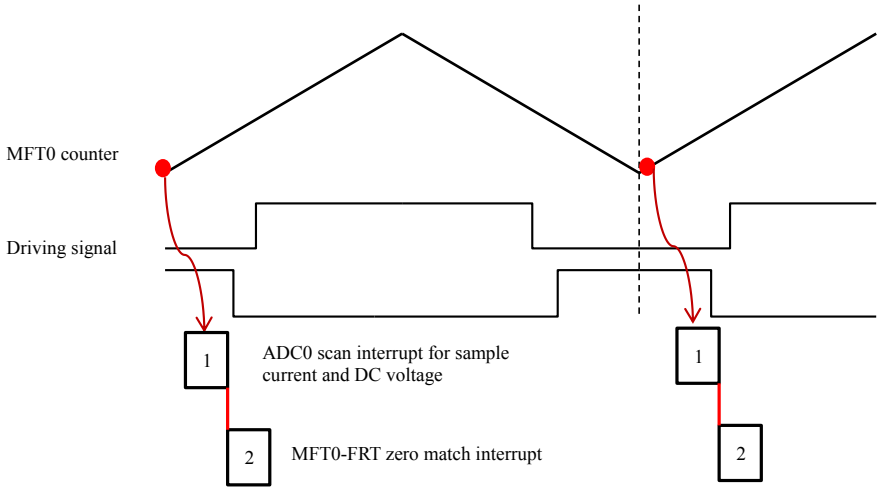
Table 2 lists the interrupts used in the firmware. The function “InitMcu_Nvic()” in *init_mcu.c* is for interrupt control. For more details about interrupt control, refer to the Cotex-M4 Technical Reference Manual.

Table 2. Interrupt Functions

Interrupt Type	Function
MFT zero-match interrupt	Execute FOC algorithm.
ADC scan interrupt	For DC voltage and 3-phase current sampling; triggered by MFT zero matching
MFT DTIF interrupt	For hardware overcurrent protection
Software watchdog interrupt	Upon software watch overflow, motor stops running.

Figure 9 illustrates MFT and ADC interrupts execution. The MFT and ADC interrupts include all functions relating to motor control. They are triggered every PWM cycle.

Figure 9. Interrupt Call Relationship in Firmware



5.2.3 Control Process Flow

Because of its higher priority, the ADC interrupt executes before the MFT zero-match interrupt. Figure 10 illustrates the basic control flow. The main flow for the control process contains three primary parts: ADC interrupt, MFT interrupt, and main function. Figure 11 shows the control process state diagram.

Figure 10. Control Flow Diagram

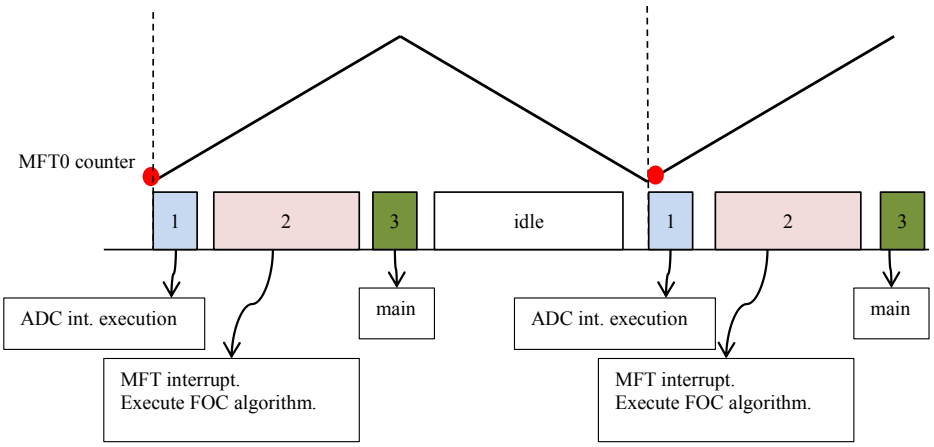
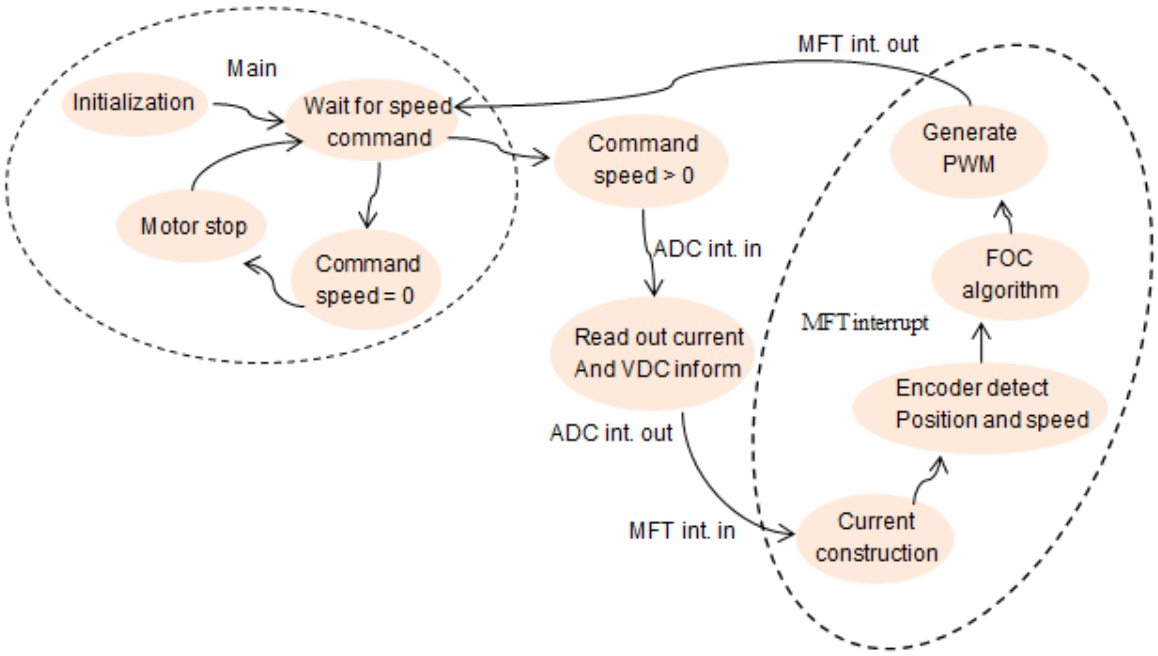


Figure 11. Firmware State Diagram



6 Test Results

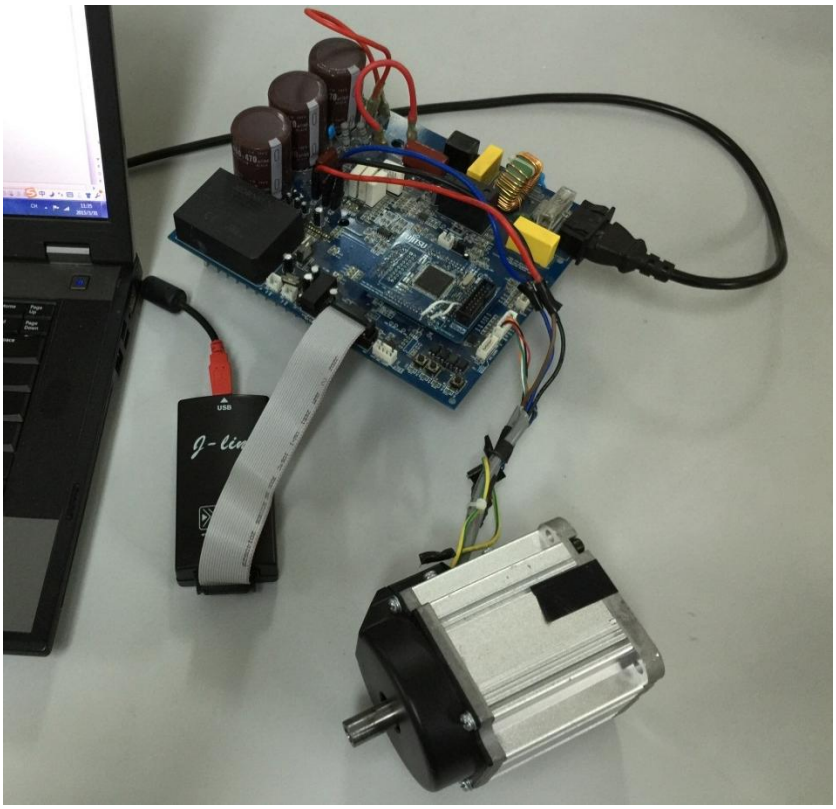
In order to demonstrate the design described above and show the control performance, the test platform is established including the starter kit, a PMSM with 360 encoder pulses, firmware, and oscilloscope. Table 3 lists the motor parameters.

Table 3. Test Parameters

Motor Parameter	Maximum	Unit
Phase current (peak)	15	A
Speed range	100~5000	rpm

Figure 12 shows the system connections.

Figure 12. System Connection Diagram



6.1 Current Waveform

Figure 13, Figure 14, and Figure 15 show the 100 rpm, 1800 rpm, and 3500 rpm current waveforms. The peak value of the phase current is less than 300 mA.

Figure 13. Motor Phase Current at 100 rpm

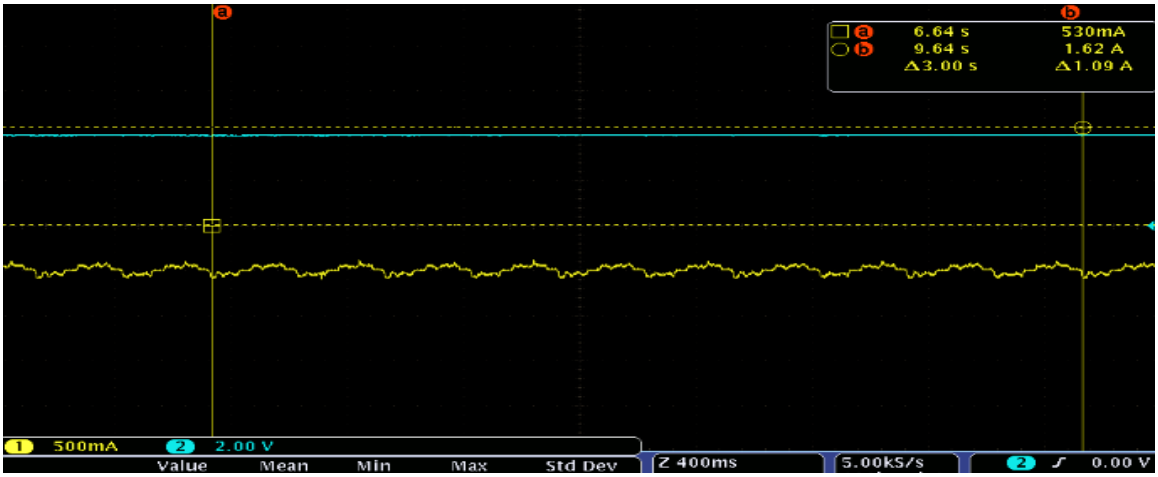


Figure 14. Motor Phase Current at 1800 rpm

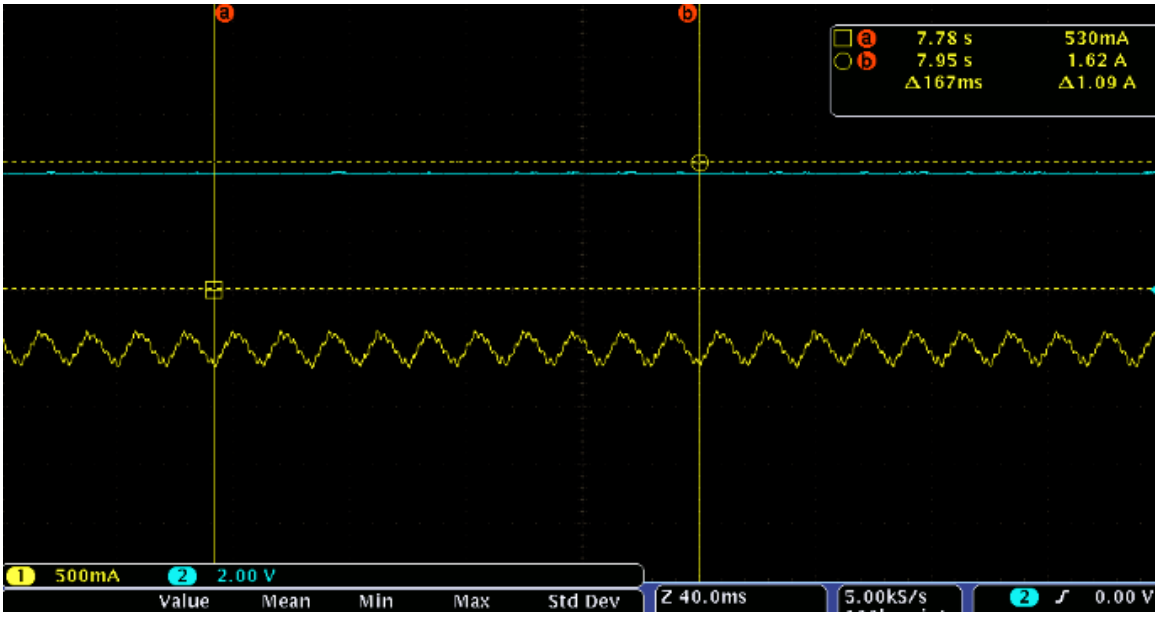
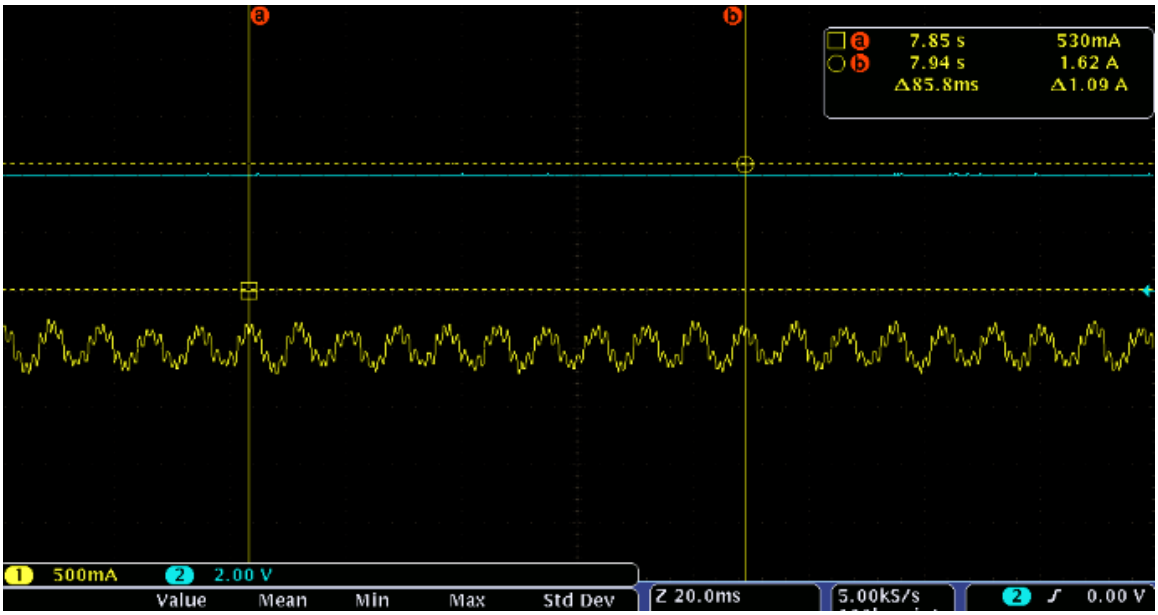


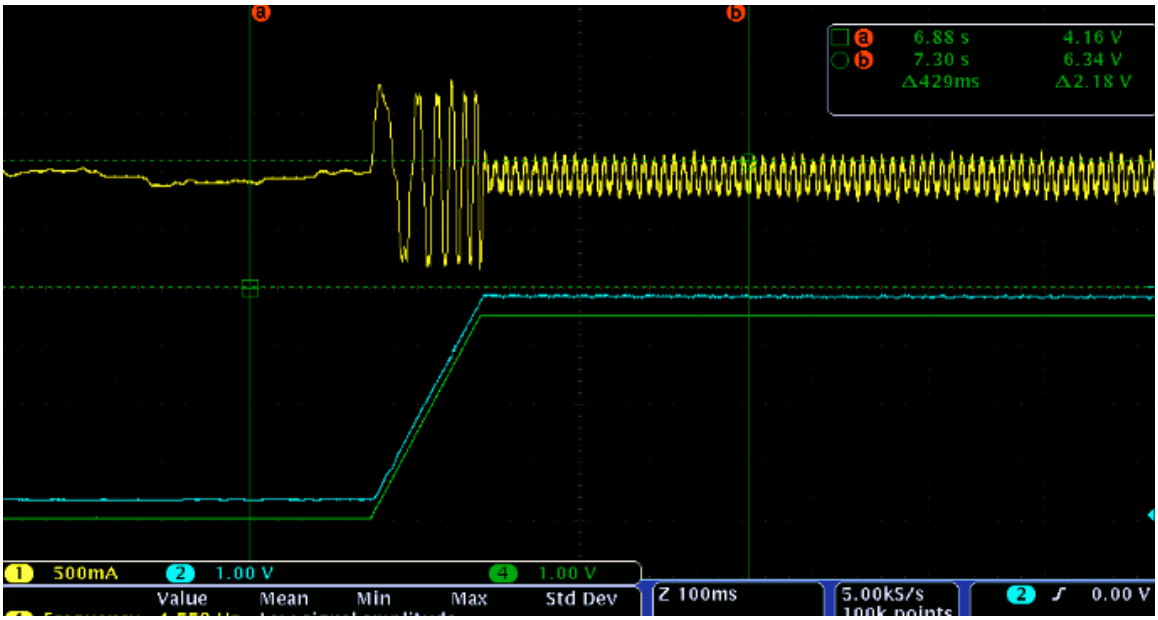
Figure 15. Motor Phase Current at 3500 rpm



6.2 Acceleration and Deceleration

Motor acceleration from 100 rpm to 3500 rpm is controlled within 200 ms, as shown in Figure 16. Speed overshoot is not obvious. The maximum phase current peak value is controlled under 1.5 A.

Figure 16. Motor Phase Current When Accelerating from 100 rpm to 3500 rpm Within 200 ms

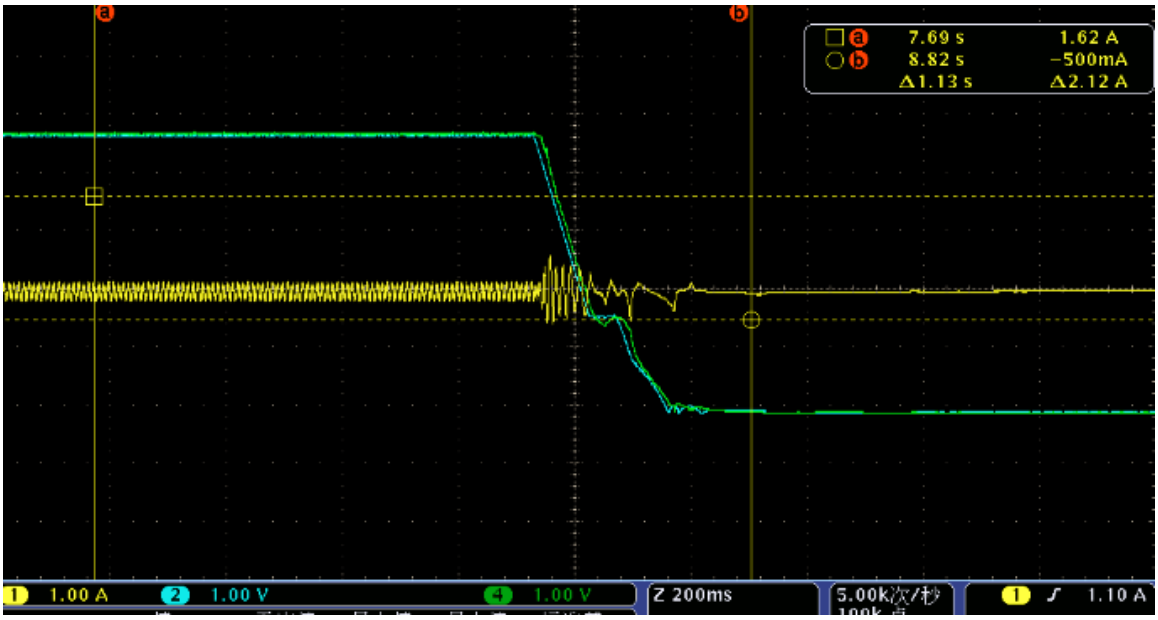


Notes:

1. Green curve indicates motor target speed.
2. Blue curve represents motor speed estimated by encoder signal.

Figure 17 shows an overview of motor deceleration from 3500 rpm to 0 rpm and stopping at a certain position, within 300 ms.

Figure 17. Motor Phase Current When Decelerating from 3500 rpm to 0 rpm Within 300 ms



Notes:

1. Green curve indicates motor target speed.
2. Blue curve represents motor speed estimated by encoder signal.

7 Summary

This application note detailed the servo motor speed control solution on the FM4 MCU MB9BF56x and S6E2HG Series. In doing so, it explained the PMSM control theory, system scope, hardware design, software design, and test results. An associated code example based on the motor control starter kit, “SK-MC-3P-LVPS-0” and “ADPT-FM4-9B560-MC” was created to demonstrate the application note content.

Document History

Document Title: AN202488 - FM4 MB9BF568x Series MCU – PMSM Servo Motor Speed Control

Document Number: 002-02488

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4938204	CBZH	11/05/2015	New application note.
*A	5799309	AESATMP9	07/05/2017	Updated logo and copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP| PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/go/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709