AN202483

FM0+ S6E1A1 Series MCU – Low-Voltage 3-Phase BLDC and PMSM Control

Author: Arthur Zhong

Associated Part Family: FM0+

Associated Code Examples: 3-Phase Low Voltage BLDC/PMSM Platform Firmware

Related Application Notes: AN93637

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN202483.

AN202483 explains the control theory and the system scope, hardware design, software design, and test results of a low-voltage brushless DC (BLDC) and permanent magnet synchronous motor (PMSM) control solution using the S6E1A1 MCU. A code example using a low-voltage motor control starter kit is included to demonstrate the solution.

Contents

1 Introduction

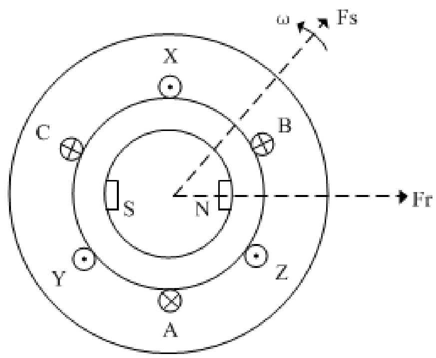
This document describes 3-phase BLDC/PMSM low-voltage motor control using the FM0+ S6E1A1 series MCU. The system scope, hardware design, software design, and test results are included.

2 PMSM Control Theory

2.1 Structure of a 3-Phase PMSM

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

Figure 1. Structure of a 3-Phase PMSM

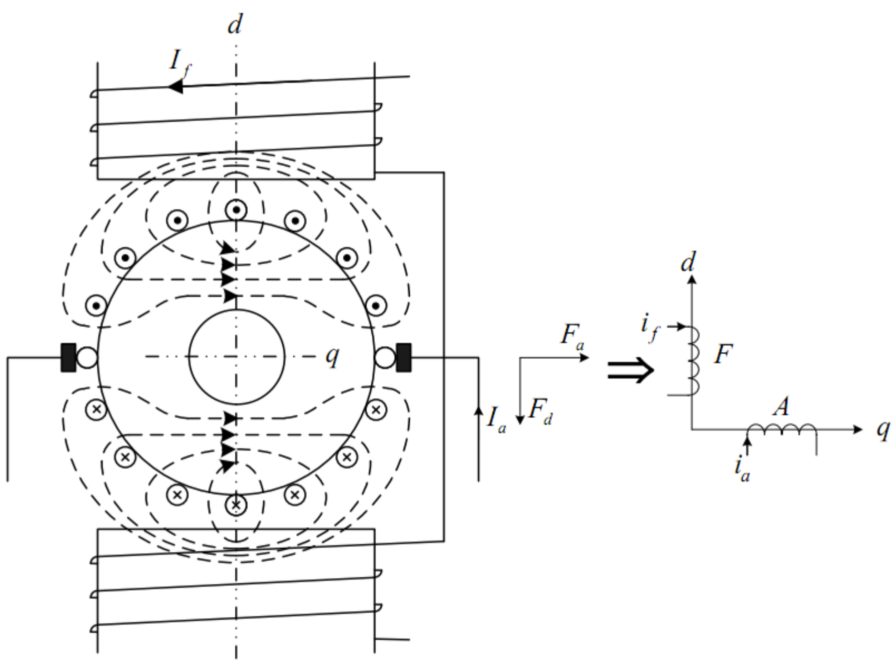


The rotor has one or more pairs of permanent magnetic poles that create a constant rotor magnetic field (Fr). Because Fs is a rotating magnetic field, Fr is dragged and follows Fs. If Fr cannot catch up with Fs, the rotor does not rotate continuously. If the 3-phase current in the 3-phase windings disappears, Fs disappears at the same time, and the rotor stops.

2.2 Field-Oriented Control Principle

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

Figure 2. BLDC Motor Decoupled Control

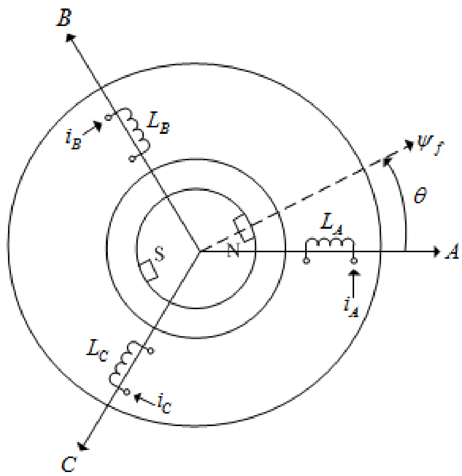


Magnetizing is controlled by the magnetizing current (I_f), and the torque is controlled by the torque current (I_a). The direction of the magnetic field is parallel to the d-axis (vertical direction), and the direction of the torque magnetic field is parallel to the q-axis (horizontal direction). So, 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 torque is controlled only by the torque current, I_a.

PMSM motor control is much more complex than BLDC motor control. The magnetic field of a 3-phase symmetry 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 a 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 of a PMSM is more complex than that of the 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 magnetic field orientation technology. In this technology, the coordinate transformation method transforms the motor model 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 the 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 reference 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.

The motor position can be obtained by two approaches: one is physical sensor (either an encoder or a Hall sensor), the other is sensorless estimator realized by software. This application note includes both ways to get the position information, as shown in Figure 4 and Figure 5.

Figure 4. FOC with Hall Sensor

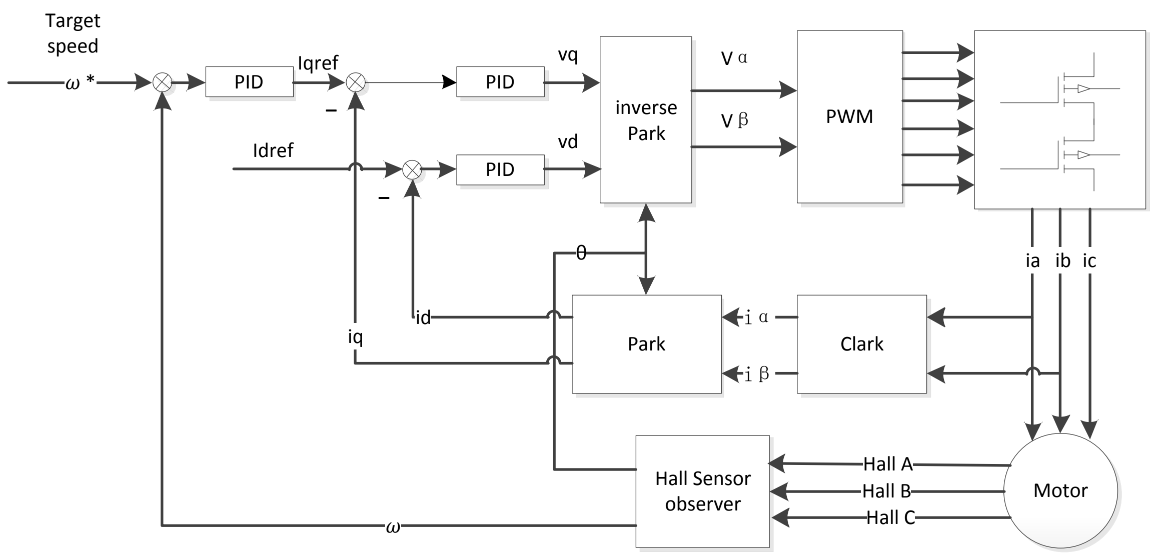
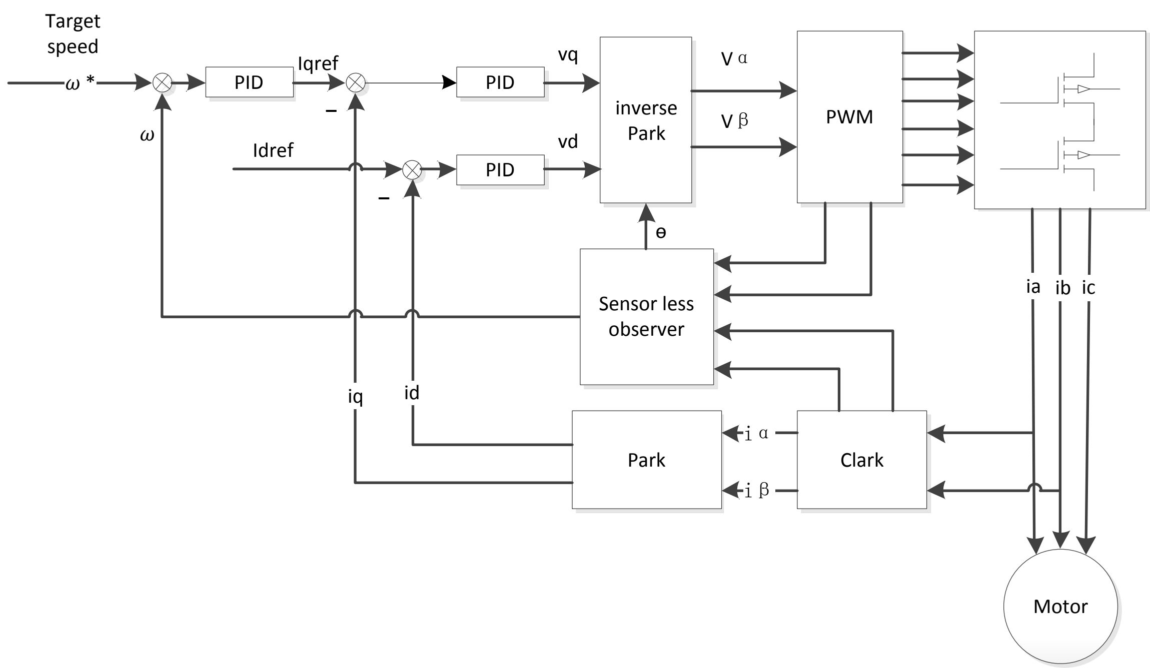


Figure 5. FOC without Sensor

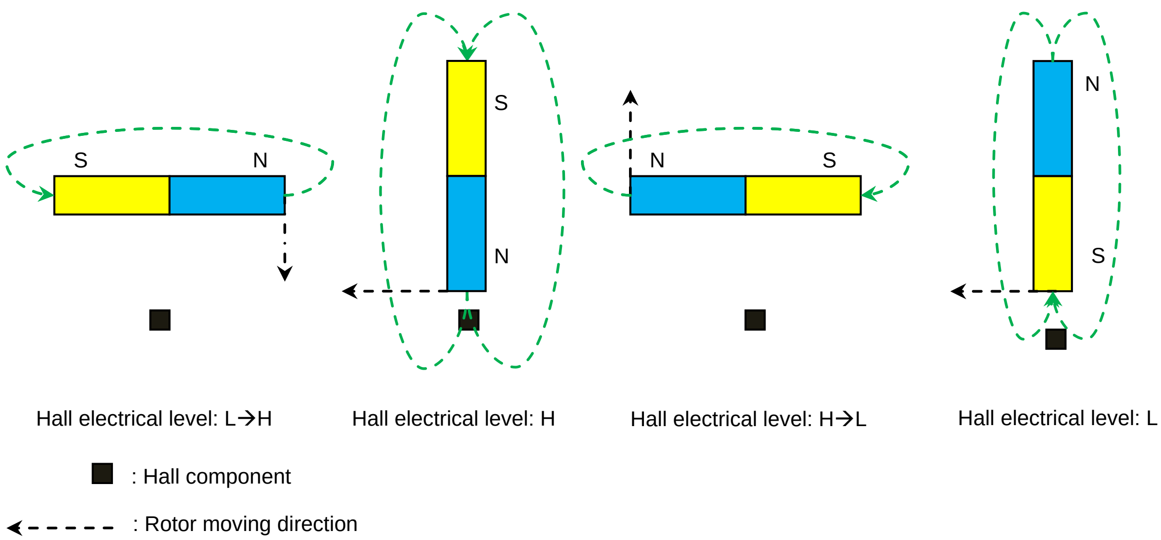


2.4 Hall Sensor

The Hall sensor can sense the PMSM/BLDC motor's rotor position. Two Hall sensors can check the rotor's 4-point position, and three Hall sensors can check the 6-point position in one electrical cycle. The Hall sensor is installed in the motor's stator as an independent physical component.

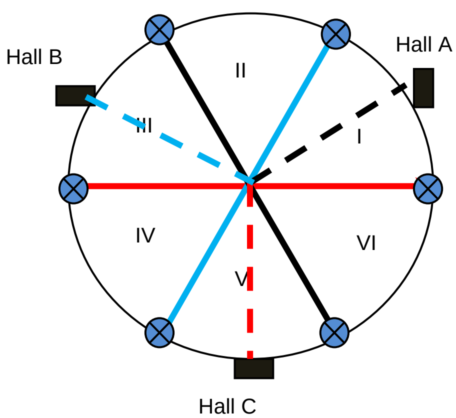
The Hall sensor signal has only two statuses: high level and low level. Figure 6 shows the change process.

Figure 6. Hall Sensor Electrical Status Change Process



Three Hall sensors have six statuses by logical combination, as shown in Figure 7.

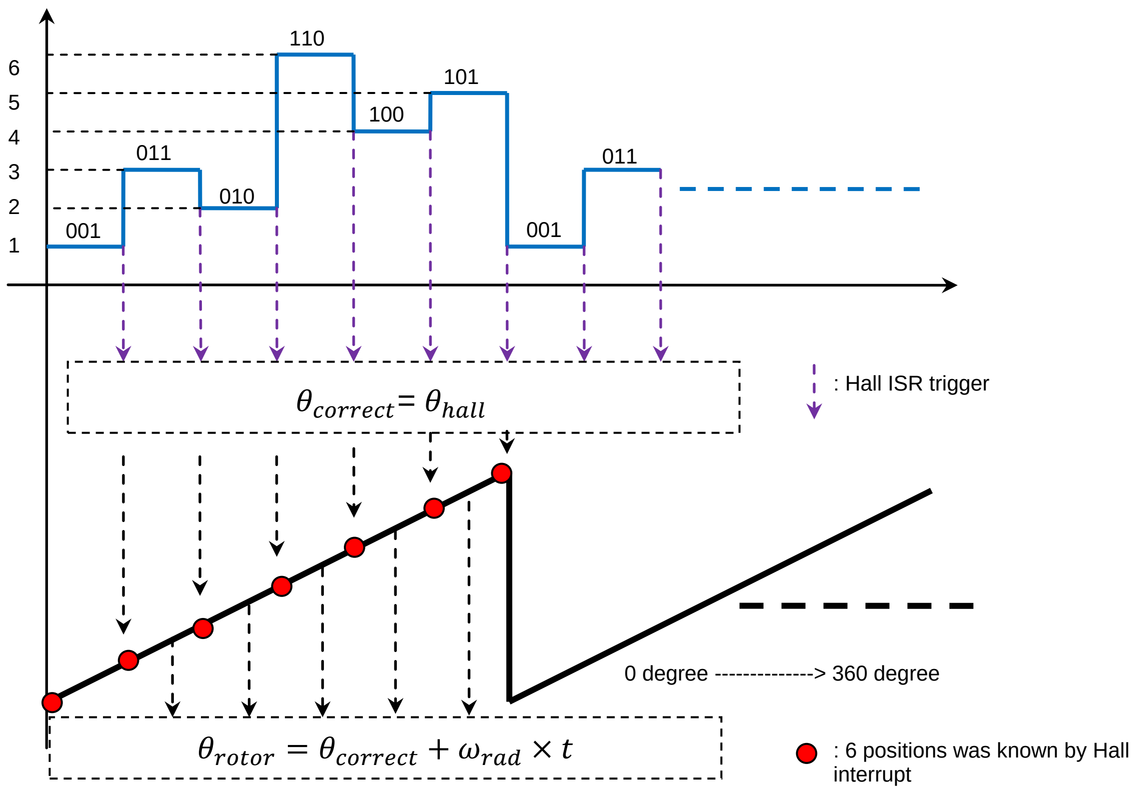
Figure 7. Three Hall Sensors' Logical Combination



	C Level	B level	A level	Hall Statues
I	0	0	1	1
II	0	1	1	3
III	0	1	0	2
IV	1	1	0	6
V	1	0	0	4
VI	1	0	1	5

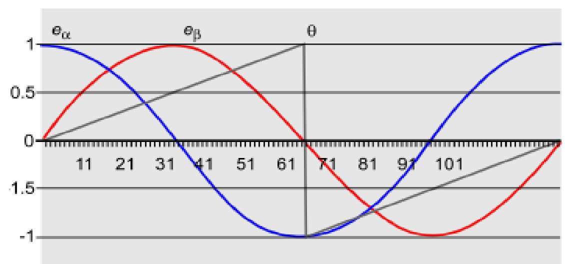
Hall sensor signals are connected to the MCU's I/O port. Once the status changes, an interrupt will occur, and the firmware module will calculate the rotor's position and speed, as shown in Figure 8.

Figure 8. Rotor Angle Calculated by Hall Edge Trigger Interrupt



The motor's position can also be obtained from the back electromotive force (EMF) when no sensor is installed in the motor. To make the low-voltage platform more compatible, this solution introduces one basic rotor position observation method. Figure 9. shows the back EMF waveform in a-b axis.

Figure 9. Back EMF Voltage Wave Variation with Rotor Phase Angle



The rotor position can be calculated in every PWM interrupt time:

$$e_\alpha = \mathrm{LPF}(V_\alpha - R_s i_\alpha - L\frac{di_a}{di_t})$$

$$e_\beta = \mathrm{LPF}(V_\beta - R_s i_\beta - L\frac{di_\beta}{di_t})$$

$$\theta_e = \arctan\frac{-e_\alpha}{e_\beta}$$

LPF is a first-order low-pass filter. As the filter is used in the calculation process, the angle must have a little phase delay compared with the real rotor position. An offset angle needs to be added to the calculated angle.

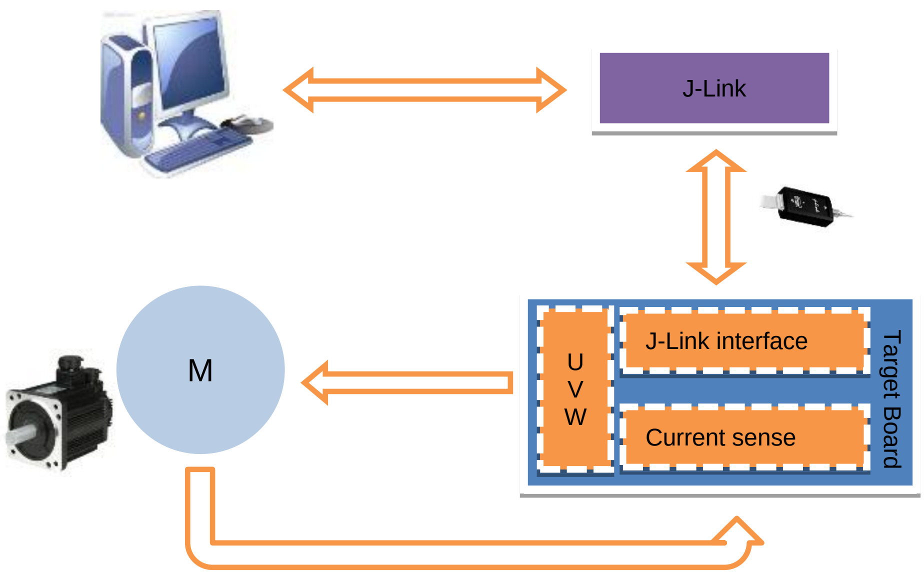
$$\theta_{e\_correct} = \theta_e + \theta_{offset}$$

Inner R_s can be checked by the Stator Resistor Check module, and L can be checked by the inductor component.

3 System Scope

This section describes the driving system of a 3-phase PMSM. Figure 10 shows the system structure and driving performance.

Figure 10. System Structure



- S6E1A1 is the target controller with a configured 40 MHz main clock.
- Motor can work with Hall sensor or be sensorless.
- System specification of low-voltage motor control starter kit:
  - Auto Hall sensor phase angle detect
  - Wide speed range: 360 rpm ~ 4000 rpm
  - Rapid speed acceleration up to 2000 rpm/s
  - Accurate speed controlling with less than 1 percent speed error
  - Input voltage range from 24 V to 48 V; maximum current: 3 A
  - PWM frequency can be changed online
  - All protection functions implementation to protect the motor
  - The system does not implement Field weaken algorithm which could expand speed range
- Firmware development environment
  - Operating system: Windows XP or later
  - IDE: IAR Embedded Workbench 7.30

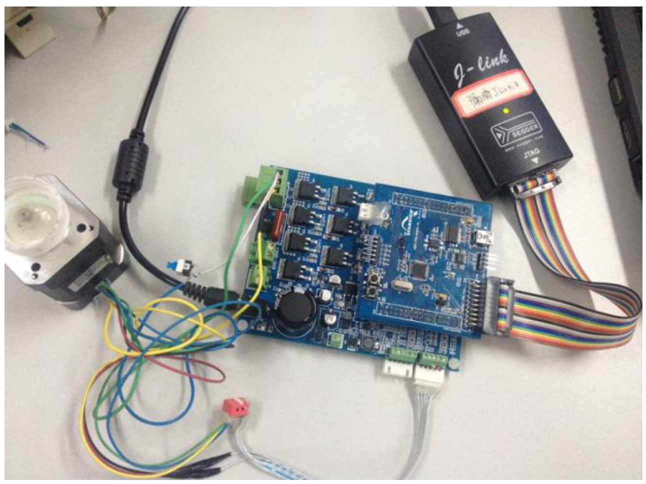
4 Hardware Design

The following are some hardware specifications of low-voltage motor control starter kit. For more information about hardware, refer to the hardware design application note.

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

Figure 11 shows the hardware system connections.

Figure 11. System Hardware Structure

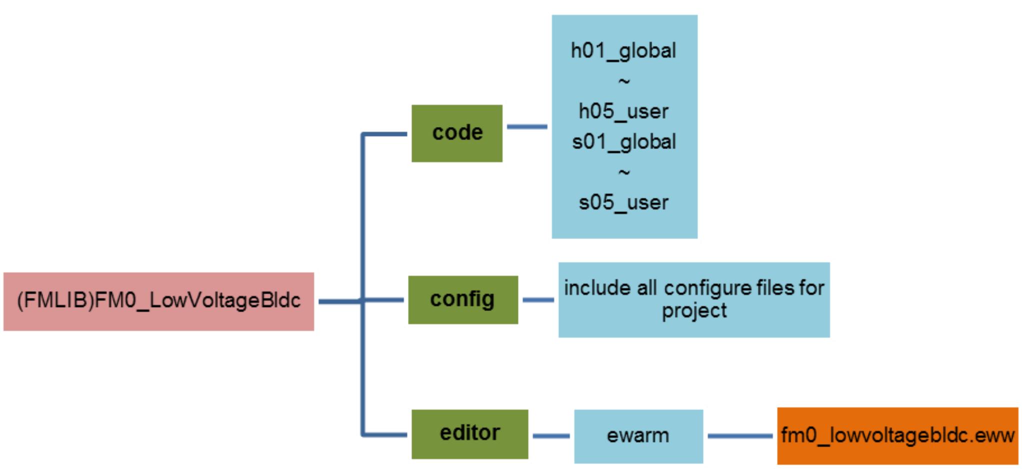


5 Software Design

5.1 Firmware File Structure

Figure 12 shows the firmware file structure.

Figure 12. Firmware File Structure



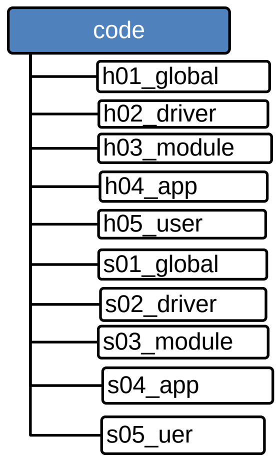
The firmware contains three subfolders: code, config, and editor. All source code is stored in the code folder, including header and C source files. Configuration and MCU description files are stored in the config folder.

Double-click on file *fm0_lowvoltagebldc.eww* to open the project in the IAR IDE.

5.1.1 Code Folders

The source code is divided into five layers by function and stored in five different folders. The five layers are named “global,” “driver,” “module,” “app,” and “user”, as Figure 13 shows.

Figure 13. Firmware Structure in Project



- Global layer is empty.
- Driver layer stores the MCU header file and macro definition files.
- Module layer stores independent functions.
- App layer is related to the actual project. The function in this folder is changeable depending on 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 S6E1A1 MCU Peripherals Used

All peripherals used in the firmware are configured in *init_mcu.c*, stored in folder “s05_user.” For more details on peripheral initialization, refer to the MCU datasheets.

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.
Base timer	Measure width of Hall signal to calculate motor current speed.
ADC	Used to sample phase current; ADC unit 0 is being used.
Multifunction timer (MFT)	Generate PWM signals to control three half-bridge circuits to run the motor; MFT unit 0 is being used.
Watchdog	Reset the MCU when the program does not execute correctly.

- 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
- Base timer settings
  - PWC (Pulse width counter) 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.
- Watch dog 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

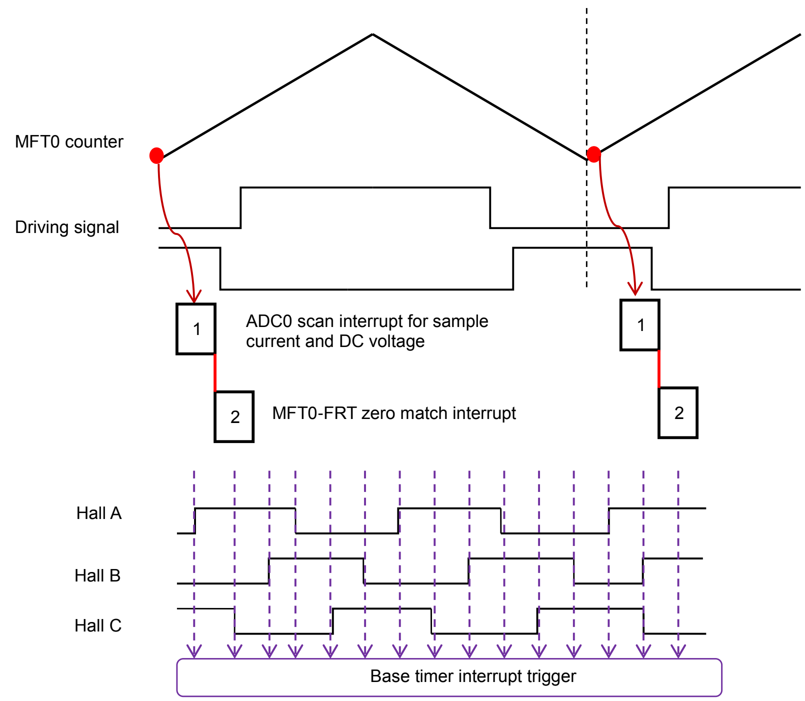
Table 2 shows the interrupts used in the system. The function “InitMcu_Nvic()” in *init_mcu.c* is for interrupt control. For more details about interrupt control, refer to the Cortex-M0 Technical Reference Manual.

Table 2. Interrupt Functions

Interrupt	Function
MFT zero match interrupt	Execute FOC algorithm.
ADC scan interrupt	For DC voltage and three-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.
Base timer hall capture interrupt	Capture Hall sensor signal edge change.

Figure 14 illustrates MFT and ADC interrupts execution. The MFT and ADC interrupts include all functions relating to motor control. They are triggered every PWM cycle. The base timer interrupt is triggered by a Hall sensor edge change.

Figure 14. Interrupt Trigger Process



5.2.3 Control Process Flow

Because of its higher priority, the ADC interrupt executes before the MFT zero match interrupt. Figure 15 illustrates the basic control theory. The main flow for the control process contains three primary parts: ADC interrupt, MFT interrupt, and main function, as shown in Figure 16.

Figure 15. ADC and MFT Interrupt in Motor Control

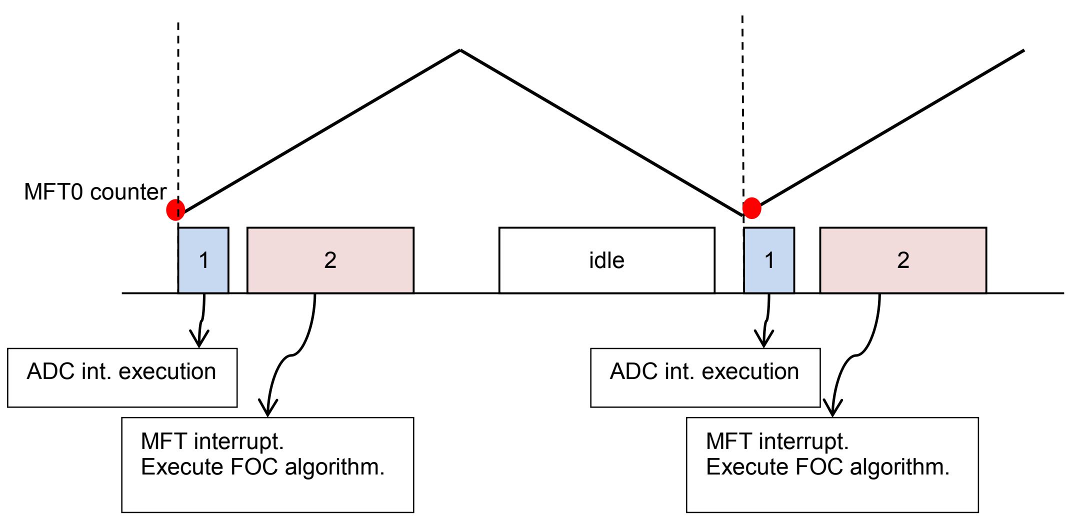
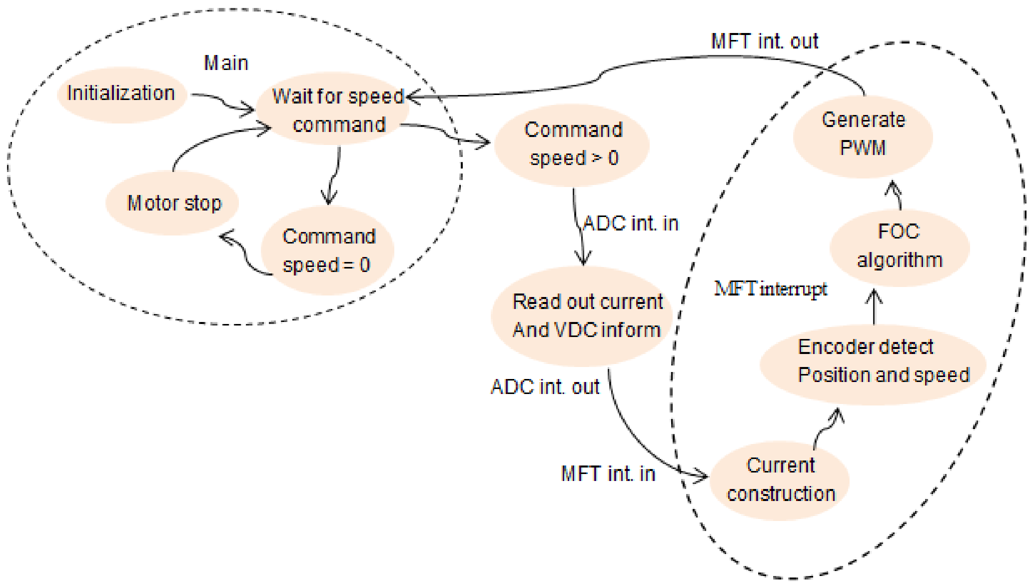


Figure 16. System Run Machine State



6 Test Results

In order to demonstrate the design described above and show the control performance, the test platform is established including low-voltage motor control starter kit, BLDC motor, firmware and oscilloscope. The control system can run in sensor mode or sensorless mode. Table 3 lists the BLDC motor parameters.

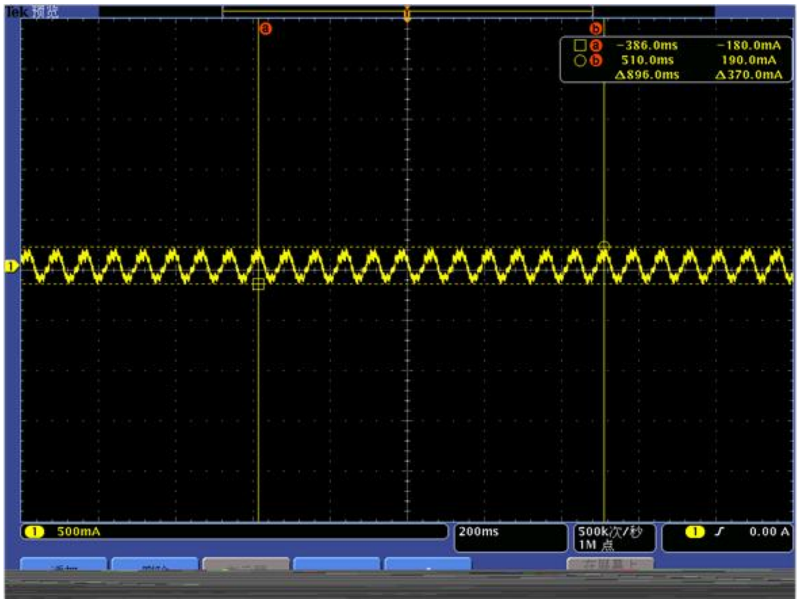
Table 3. Motor Parameters in Test Platform

Motor Parameter	Maximum	Unit
Phase current(peak)	2	A
Speed range	360~4000	rpm
Ld	0.65	mH
Lq	0.85	mH
Rs	0.5	Ω
Ke	2.8	Vrms/krpm
Pole pairs	2	N/A

6.1 Current Waveform with Sensorless Control

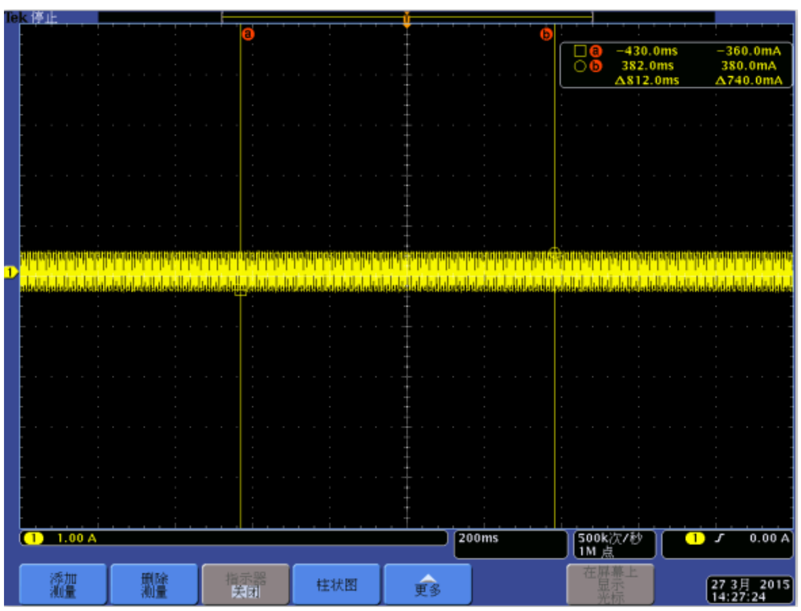
Figure 17 and Figure 18 show the 360-rpm and 4000-rpm current waveform in the sensorless control mode.

Figure 17. Motor Phase Current Wave at 360 rpm Without Hall Sensor Control



The motor’s phase current peak value is less than 190 mA. The motor running is very stable at a low speed. The system reached the low-speed motor running requirement.

Figure 18. Motor Phase Current in 4000 rpm Without Hall Sensor Control



From the test wave, it can be seen that the motor running at the maximum speed is very stable, and current enveloping is much smoother. It also indirectly reflects that noise is much lower.

6.2 Current Waveform with Three Hall Sensor Control

Figure 19 and Figure 20 show 360 rpm and 4000 rpm current waveform with three Hall sensor control.

Figure 19. Motor Phase Current at 360 rpm with Three Hall Sensor Control

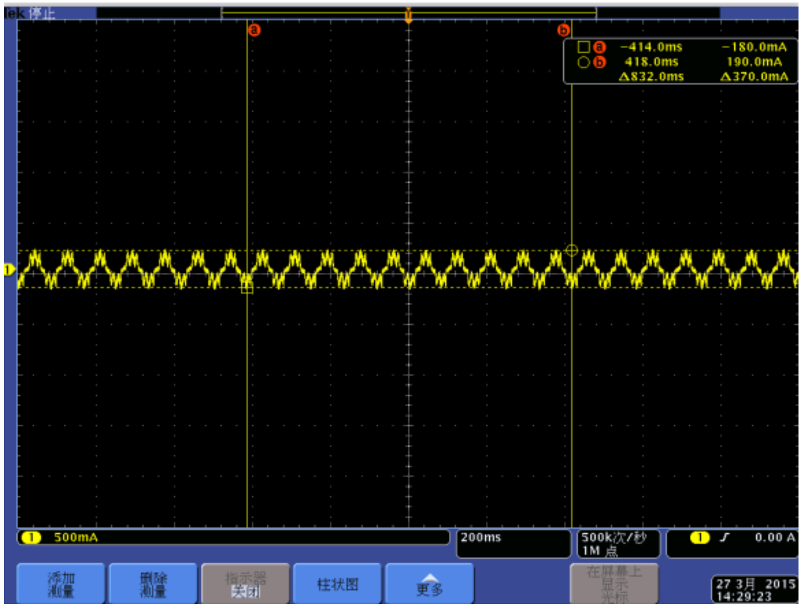
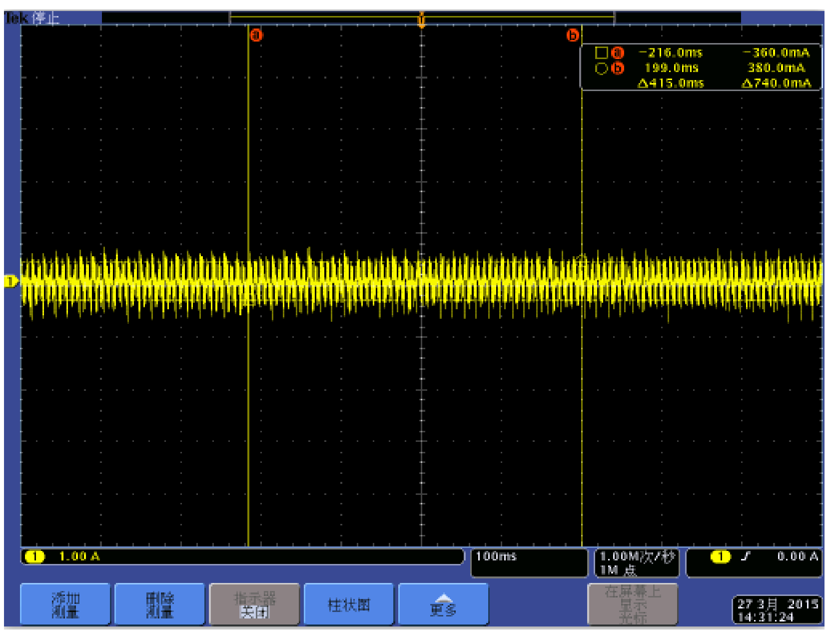


Figure 20. Motor Phase Current at 4000 rpm with Three Hall Sensor Control



From the test wave, it can be seen that the Hall sensor solution with the motor's phase current at 4000 rpm is not better than the sensorless solution. The position calculation result is different in each solution. One is from the physical Hall sensor, and the other is estimated based on the sampled current and back EMF. So at high speed, the harmonic and noise of the Hall sensor solution are much worse than that of the sensorless solution.

6.3 Lock Rotor

Figure 21 and Figure 22 show motor phase current waveform with and without hall sensor, with a locked rotor.

Figure 21. Motor Phase Current Wave with Locked Rotor with Hall Sensor at 1000 rpm

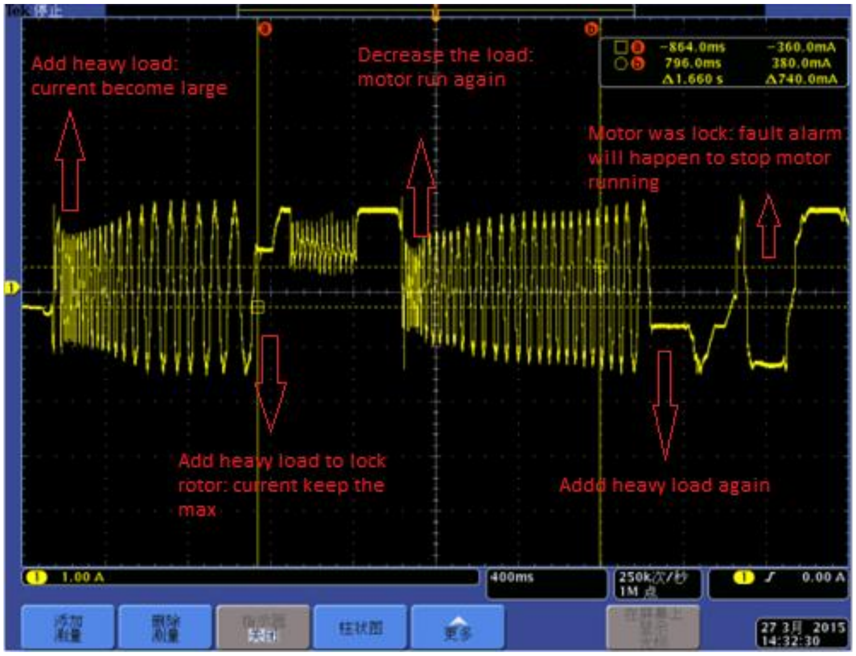


Figure 22. Motor Phase Current Wave With Locked Rotor Without Hall Sensor at 1000 rpm

7 Summary

This application note introduced a BLDC and PMSM low-voltage motor control solution using the FM0+ S6E1A1 Series MCU. In doing so, it explained the PMSM control theory, system scope, hardware design, firmware design, and test results. An associated code example based on a low-voltage motor control starter kit, including two boards SK-MC-3P-LVPS-0 and ADPT-FM0-S6E1A1-MC, was created to demonstrate the application note content.

Document History

Document Title: AN202483 – FM0+ S6E1A1 Series MCU – Low-Voltage 3-Phase BLDC and PMSM Control

Document Number: 002-02483

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4932915	CBZH	11/05/2015	New application note.
*A	5711713	AESATP12	04/26/2017	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/support

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