

# Sensorless BLDC Motor Control Using MC9S08AW60

DRM086  
Rev. 0.1  
01/2007





# Sensorless BLDC Motor Control Using MC9S08AW60

## Designer Reference Manual

---

by: Pavel Grasblum  
Freescale Czech Systems Center  
Roznov pod Radhostem, Czech Republic

To provide the most up-to-date information, the revision of our documents on the World Wide Web are the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to: <http://www.freescale.com>

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.  
This product incorporates SuperFlash® technology licensed from SST.

© Freescale Semiconductor, Inc., 2007. All rights reserved.

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

## Revision History

<b>Date</b>	<b>Revision Level</b>	<b>Description</b>	<b>Page Number(s)</b>
November 2006	0	Initial release	62
January 2007	0.1	Changes in chapter 3 and 5 to reflect latest version of the software	62

## Chapter 1 Introduction

1.1	Introduction .....	7
1.2	Freescale MC9S08AW60 Advantages and Features .....	8

## Chapter 2 Control Theory

2.1	Brushless DC Motor (BLDC Motor) .....	11
2.2	Power Stage .....	14
2.3	Mathematical Description of Brushless DC Motor .....	14
2.3.1	Power Stage – Motor System Model .....	14
2.3.2	Mathematical Model .....	15
2.3.3	Back-EMF Sensing .....	17

## Chapter 3 System Concept

3.1	System Specification .....	21
3.2	Sensorless Drive Concept .....	22
3.3	System Blocks Concept .....	23
3.3.1	PWM Voltage Generation for Brushless DC Motor .....	23
3.3.1.1	Complementary PWM Mode .....	23
3.3.1.2	Independent PWM Mode .....	25
3.3.1.3	Conclusion .....	25
3.3.2	ADC Sampling Mechanism .....	26
3.3.3	Back-EMF Zero Crossing Sensing .....	27
3.3.4	Current (Torque) Sensing .....	28
3.3.5	Sensorless Commutation Control .....	29
3.3.5.1	Alignment .....	29
3.3.5.2	Start Up .....	29
3.3.5.3	Stabilization .....	30
3.3.5.4	Run .....	30

## Chapter 4 Hardware

4.1	Hardware Implementation .....	31
4.2	Component Descriptions .....	32
4.2.1	MC9S08AW60 Evaluation Board .....	32
4.2.2	Micro Power Board .....	33
4.2.3	Motor IB23810 (Produced by MCG) .....	35

## Chapter 5 Software Design

5.1	Introduction .....	37
5.2	Main Software Flowchart .....	37
5.3	Data Flow .....	38
5.4	Processes Description .....	40
5.4.1	Process Zero-Crossing Detection .....	40

5.4.2	Process Torque and Speed Calculation . . . . .	41
5.4.3	Process Standstill Detection . . . . .	41
5.4.4	Process Commutation . . . . .	41
5.4.5	Process Manual Interface . . . . .	41
5.4.6	Process Application State Machine . . . . .	41
5.4.7	Process Speed and Torque PI Controller . . . . .	42
5.4.8	Process Ramp . . . . .	43
5.4.9	Process FreeMASTER . . . . .	43
5.4.10	Process PWM Update . . . . .	43
5.5	Constant Calculation . . . . .	45
5.5.1	Alignment Constants . . . . .	45
5.5.2	Start-Up Constants . . . . .	45
5.5.3	Speed Constants and Scaling . . . . .	46
5.5.4	Torque Constants and Scaling . . . . .	47
5.5.5	DC-Bus Voltage Scaling . . . . .	47

**Chapter 6  
Application Setup**

6.1	Hardware Setup . . . . .	49
6.2	Software Setup . . . . .	52
6.2.1	Application Software Files . . . . .	52
6.2.2	Application FreeMASTER Software Control Files . . . . .	52
6.2.3	Application Build . . . . .	53
6.2.4	Programming the MCU . . . . .	54

**Chapter 7 Sensorless BLDC Demo Operation**

7.1	Application Control . . . . .	55
7.2	BLDC Drive Operational Modes . . . . .	55
7.2.1	Speed Operational Mode . . . . .	55
7.2.2	Torque Operational Mode . . . . .	55
7.3	Manual/Remote Operation . . . . .	56
7.3.1	Manual Operation . . . . .	56
7.3.2	Remote Operation . . . . .	56

**Appendix A  
References**

**Appendix B  
Glossary**

# Chapter 1

## Introduction

### 1.1 Introduction

This designer reference manual describes a sensorless, 3-phase, brushless DC (BLDC) motor drive based on Freescale's MC9S08AW60 dedicated motor control device.

BLDC motors are very popular in a wide application area. Because the BLDC motor lacks a commutator, it is more reliable than the DC motor. The BLDC motor also has advantages when compared to an AC induction motor. Because it achieves a higher efficiency by generating the rotor magnetic flux with rotor magnets, a BLDC motor is used in high-end white goods (such as refrigerators, washing machines, dishwashers, high-end pumps, fans) and in other appliances requiring high reliability and efficiency.

The concept of the application is a speed and torque, closed-loop BLDC drive using a sensorless back-EMF zero crossing technique. It is an example of a BLDC motor control design using a Freescale HCS08 family MCU.

This reference design includes basic motor theory, system design concept, hardware implementation, and software design, including the FreeMASTER software visualization tool.



**Figure 1-1. Sensorless 3-Phase BLDC Motor Demo Using MC9S08AW60**

## 1.2 Freescale MC9S08AW60 Advantages and Features

The MC9S08AWx is a family of devices based on the high-performance HCS08 core, supporting 5-volt applications. It is a highly integrated, high-performance family packed with valuable features:

- 16 – 60K flash memory
- 1 – 2K RAM
- A flexible internal clock generator that eliminates the need for external components, low voltage detection, high-performance, analog-to-digital converter (ADC), serial communication modules, etc.

The MC9S08AWx family performs well in a variety of environments and is qualified for automotive applications. This family is suitable for appliance (white goods), industrial, and automotive applications.

The HCS08 family incorporates these standard features:

- 40-MHz HCS08 CPU
- HC08 instruction set with BGND instruction
- Internal background debugging system
- Breakpoint capability to allow single breakpoint setting during in-circuit debugging (plus two more breakpoints in the on-chip debug module)
- Debug module:
  - Contains two comparators, nine trigger modules, and eight deep FIFO for storing change-of-flow addresses and event-only data
  - Supports tag and force breakpoints
- Support for up to 32 interrupt/reset sources
- System protection features:
  - Optional computer operating properly (COP) reset
  - Low voltage detection with reset or interrupt
  - Illegal opcode detection with reset
  - Illegal address detection with reset (some devices don't have illegal addresses)

The MC9S08AW60 incorporates these features:

- 60 Kbytes of on-chip, in-circuit programmable flash memory with block protection and security options
- 2 Kbytes of on-chip RAM
- 16-channel, 10-bit analog-to-digital converter (ADC)
- Two serial communications interface (SCI) modules
- Serial peripheral interface (SPI) module
- External voltage monitor (EVM) allowing input level monitoring with selectable trigger point and interrupt generation
- Clock source options including crystal, resonator, external clock, or an internally generated clock with precision NVM trimming
- Inter-integrated circuit (IIC) bus module to operate at up to 100 kbps
- One 2-channel and one 6-channel, 16-bit timer/pulse-width modulator (TPM) modules
  - Selectable input-capture, output-compare, and edge-aligned pulse-width-modulation (PWM) capability on each channel
  - Each timer module may be configured for buffered, centered PWM (CPWM) on all channels



- 16-bit periodic interrupt timer (PIT) counter
- 8-pin keyboard interrupt (KBI) module
- Eight high-current pins (limited by package dissipation)
- Software selectable pullups on ports when used as input (selection is on an individual port bit basis; during output mode, pullups are disengaged)
- Software selectable slew rate control on ports when used as inputs
- Master reset pin and power-on reset (POR)
- Internal pullup on  $\overline{\text{RESET}}$  and IRQ pins to reduce customer system cost
- 52 general-purpose input/output (I/O) pins
- 64-pin low-profile quad flat package (LQFP) and 64-pin quad flat package (QFP)

Table 1-1 shows memory configuration of the MC9S08AW60 family. Table 1-2 shows peripheral availability according to package type.

**Table 1-1 Memory Configuration**

Device	Flash Memory	RAM	Package
MC9S08AW60	63280	2048	64 QFP 64LQFP 48QFN 44LQFP
MC9S08AW48	49152		
MC9S08AW32	32768		
MC9S08AW16	16384	1024	

**Table 1-2 Peripherals Availability per Package Type**

Feature	Package Option		
	64-pin	48-pin	44-pin
ADC	16-ch	8-ch	8-ch
IIC	yes		
IRQ	yes		
KBI1	8	7	6
SCI1	yes		
SCI2	yes		
SPI1	yes		
TPM1	6-ch	4-ch	4-ch
TPM1CLK	yes	no	no
TPM2	2-ch		
TPM2CLK	yes	no	no
I/O pins	54	38	34

The key peripherals for 3-phase BLDC motor control are the TPM/PWM module and an analog-to-digital converter (ADC) converter. The MC9S08AWx includes a 6-channel TPM1 and a separate 2-channel TPM2. The 6-channel TPM1 module allows the generation of any PWM pattern for any 3-phase motor control. The TPM module features include:

- Eight channels:
  - Each channel may be input capture, output compare, or buffered edge-aligned PWM
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
  - Selectable polarity on PWM outputs
- Each TPM may be configured for buffered, center-aligned PWM (CPWM) on all channels
- Clock source to the prescaler for each TPM is independently selectable as the bus clock, fixed system clock, or an external pin:
  - Prescale taps for divide by 1, 2, 4, 8, 16, 32, 64, or 128
  - External clock inputs TPM1CLK for TPM1, and TPM2CLK for TPM2 (only available in the 64-pin package)
- 16-bit free-running or up/down (CPWM) count operation
- 16-bit modulus register to control the counter range
- Timer system enable
- One interrupt per channel and a terminal count interrupt for each TPM module

The ADC has the following features:

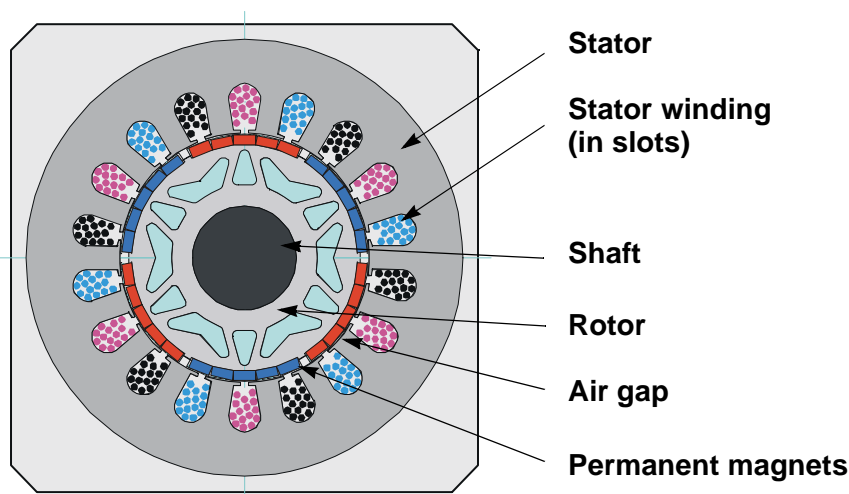
- Linear successive approximation algorithm with 10 bits resolution
- Up to 28 analogue inputs
- Output formatted as 10- or 8-bit right-justified
- Single or continuous conversion (automatic return to idle after single conversion)
- Configurable sample time and conversion speed/power
- Conversion complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in wait or stop3 modes for lower noise operation
- Asynchronous clock source for lower noise operation
- Selectable asynchronous hardware conversion trigger
- Automatic compare with interrupt for less-than, or greater-than or equal-to, programmable value

The ADC evaluates back-EMF, zero-crossing detection without any external comparators and senses other analog quantities necessary for BLDC motor control.

## Chapter 2 Control Theory

### 2.1 Brushless DC Motor (BLDC Motor)

The brushless DC motor (BLDC motor) is a rotating electric machine with a classic 3-phase stator like an induction motor; the rotor has surface-mounted permanent magnets. It is also referred to as an electronically commutated motor. There are no brushes on the rotor, and the commutation is performed electronically at certain rotor positions. The stator is usually made from magnetic steel sheets. A typical cross section of a BLDC Motor is shown in [Figure 2-1](#). The stator phase windings are inserted in the slots (distributed winding) or wound as one coil onto the magnetic pole. Because the air gap magnetic field is produced by permanent magnets, the rotor magnetic field is constant. The magnetization of the permanent magnets and their displacement on the rotor is chosen so the back-EMF (the voltage induced into the stator winding due to rotor movement) shape is trapezoidal. This allows the DC voltage (see [Figure 2-2](#)), with a rectangular shape, to create a rotational field with low torque ripples.

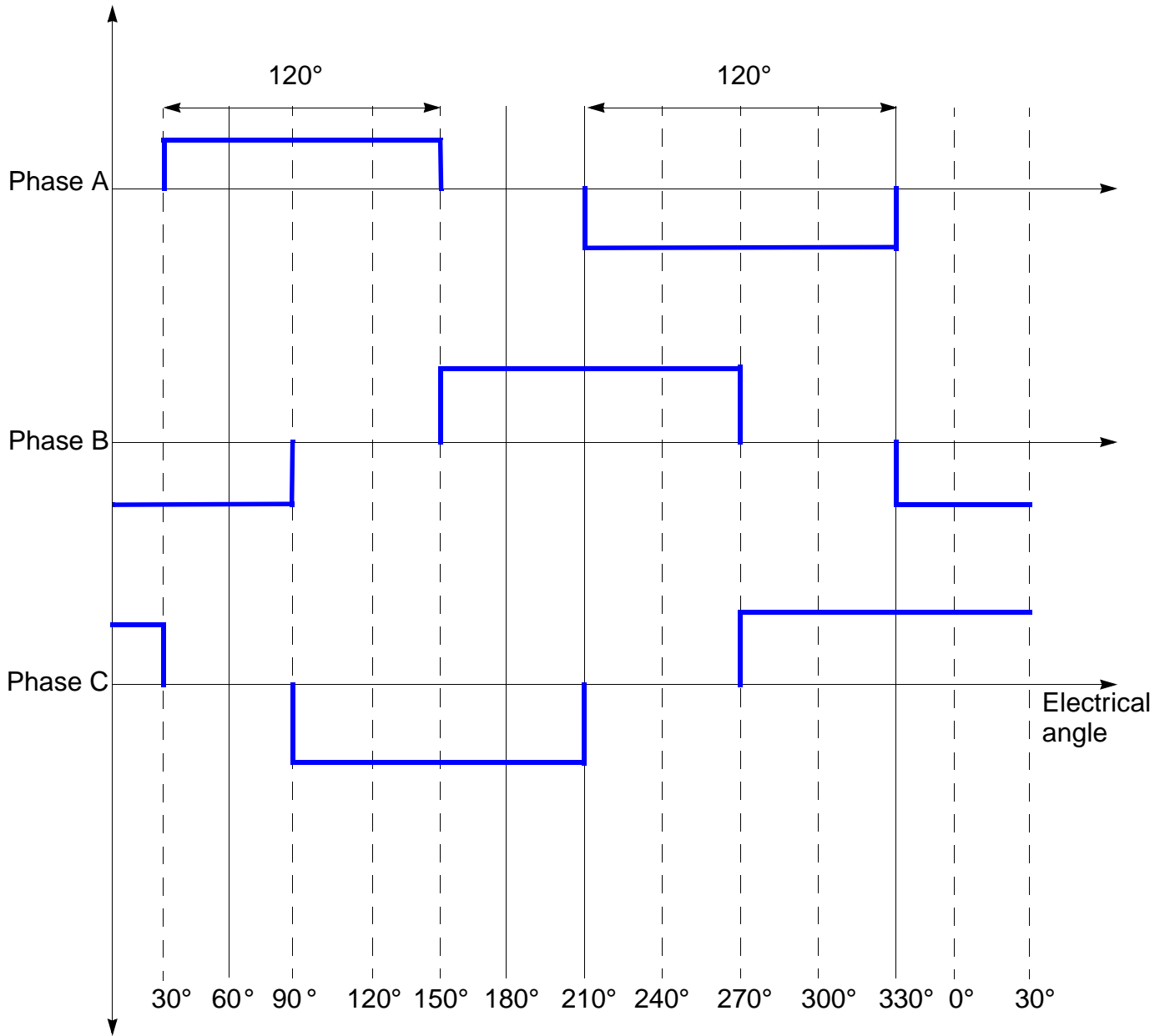


**Figure 2-1. BLDC Motor / Cross Section**

The motor can have more than just one pole-pair per phase. The pole-pair per phase defines the ratio between the electrical revolution and the mechanical revolution. For example, the shown BLDC motor ([Figure 2-2](#)) has three pole-pairs per phase, which represent the three electrical revolutions per one mechanical revolution.

The rectangular, easy-to-create shape of the applied voltage ensures the simplicity of control and drive. However, the rotor position must be known at certain angles to align the applied voltage with the back-EMF. The alignment between back-EMF and commutation events is very important. Under this condition the motor behaves as a DC motor and runs at the best working point. Therefore, the simplicity

of control and performance makes the BLDC motor the best choice for low-cost, high-efficiency applications.



**Figure 2-2. 3-Phase Voltage System for BLDC Motor**

Figure 2-3 shows the number of waveforms, the magnetic flux linkage, the phase back-EMF voltage, and the phase-to-phase back-EMF voltage. The magnetic flux linkage was measured by calculating the integration phase back-EMF voltage, which was measured on the non-fed motor terminals of the BLDC motor. The back-EMF shape is approximately trapezoidal, and the amplitude is a function of the actual speed. During speed reversal, the amplitude changes, and its sign and the phase sequence change too.

The filled areas in the top of the phase back-EMF voltage waveforms indicate the intervals where the particular phase power stage commutations are conducted. The power switches are cyclically commutated through the six steps. The crossing points of the phase Back-EMF voltages represent the natural commutation points. In a normal operation, the commutation is performed here. Some control techniques lead the commutation by a defined angle to control the drive above the PWM voltage control.

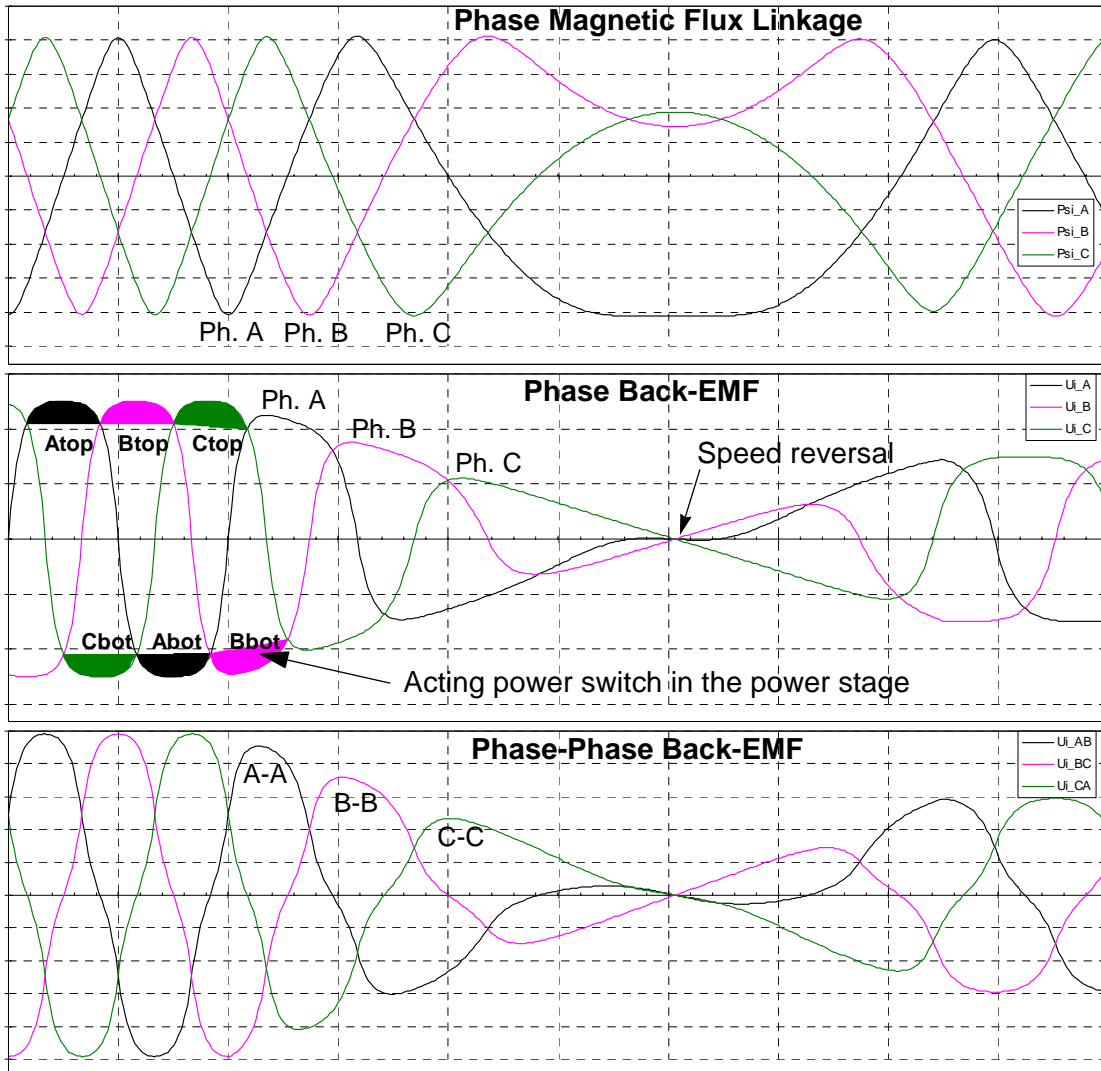


Figure 2-3. BLDC Motor / Back-EMF and Magnetic Flux

## 2.2 Power Stage

The voltage for a 3-phase BLDC motor is provided by a typical 3-phase power stage. The MCU on-chip TPM/PWM module controls the 3-phase power stage, which creates the desired switch-control signals (Figure 2-4).

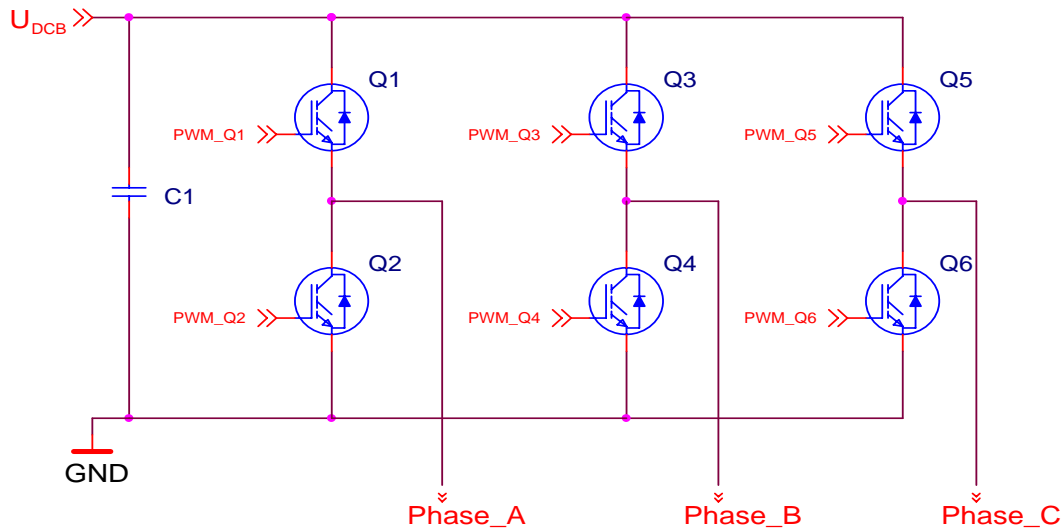
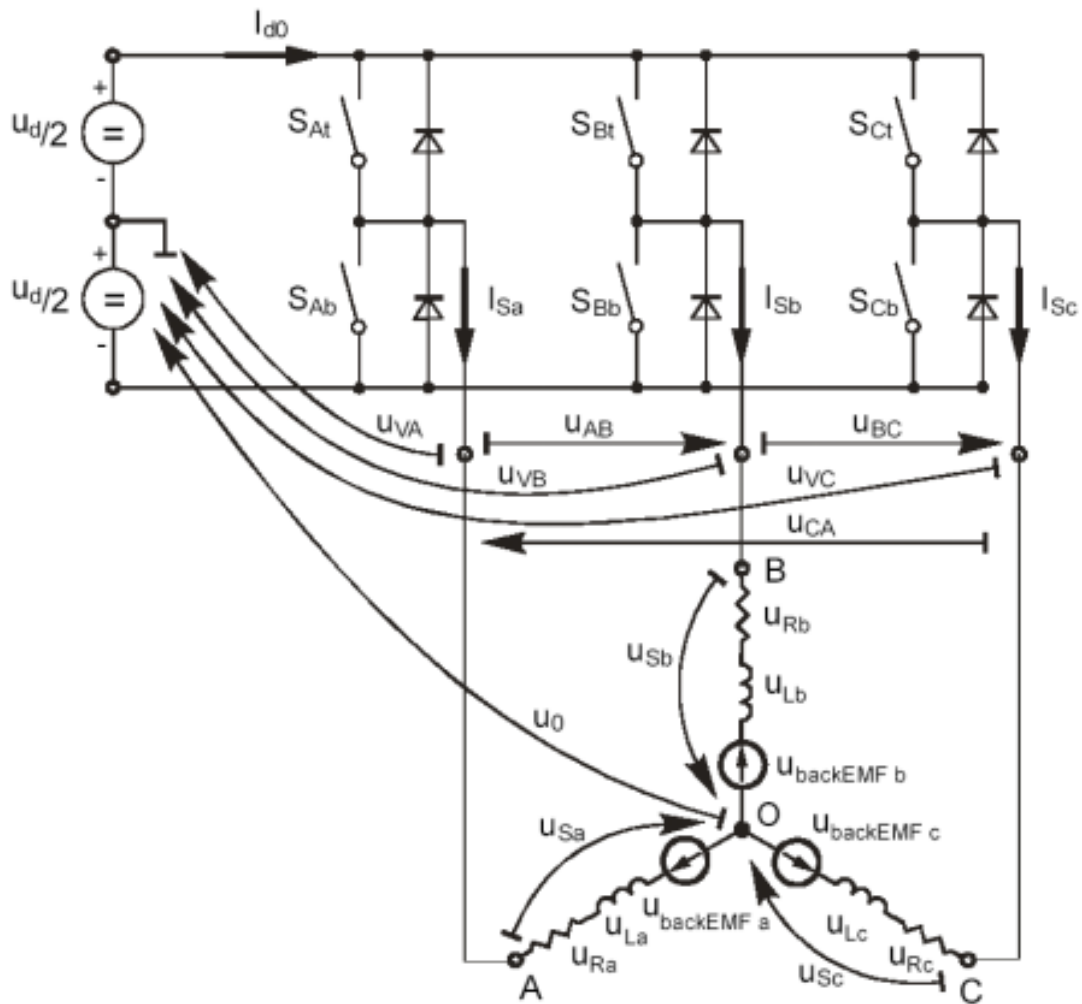


Figure 2-4. 3-Phase BLDC Motor Power Stage

## 2.3 Mathematical Description of Brushless DC Motor

### 2.3.1 Power Stage – Motor System Model

To explain and simulate back-EMF sensing techniques, a simplified mathematical model based on the basic circuit topology is provided (Figure 2-5).



**Figure 2-5. Power-Stage and Motor Topology**

The model goal is to discover how the motor characteristics depend on the switching angle. The switching angle is the angular difference between a real switching event and an ideal one (at the point where the phase-to-phase back-EMF crosses zero).

The motor-drive model consists of a 3-phase power stage plus a brushless DC motor. A voltage source ( $U_d$ ) provides the power for the system. Six semiconductor switches ( $S_{A/B/C t/b}$ ), controlled elsewhere, allow the rectangular voltage waveforms to be applied (Figure 2-2). The semiconductor switches and diodes are simulated as ideal devices. The natural voltage level of the whole model is applied at half the DC-bus voltage. This simplifies the mathematical expressions.

### 2.3.2 Mathematical Model

The BLDC motor is usually symmetrical. All phase resistances, phase and mutual inductances, and flux-linkages can be thought of as equal to, or as a function of, the position with a 120 electrical degree displacement. The electrical BLDC motor model then consists of the set of the following stator voltage equations (2-1).

The equations in (2-4) can be rewritten, taking into account the motor phase resistance and the inductance. The mutual inductance between the two motor-phase windings can be neglected because it is very small and has no significant effect for our abstraction level.

$$\begin{bmatrix} u_{Sa} \\ u_{Sb} \\ u_{Sc} \end{bmatrix} = R_S \begin{bmatrix} i_{Sa} \\ i_{Sb} \\ i_{Sc} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \Psi_{Sa} \\ \Psi_{Sb} \\ \Psi_{Sc} \end{bmatrix} \quad (2-1)$$

The following set of equations is valid for the presented topology:

$$\begin{aligned} u_{Sa} &= u_{VA} - u_0 \\ u_{Sb} &= u_{VB} - u_0 \\ u_{Sc} &= u_{VC} - u_0 \end{aligned} \quad (2-2)$$

$$u_0 = \frac{1}{3} \left[ \sum_{x=A}^C u_{Vx} - \sum_{x=a}^c u_{backEMFx} \right] \quad (2-3)$$

$$\begin{aligned} u_{Sa} &= \frac{1}{3} \left[ 2u_{VA} - u_{VB} - u_{VC} + \sum_{x=a}^c u_{backEMFx} \right] \\ u_{Sb} &= \frac{1}{3} \left[ 2u_{VB} - u_{VA} - u_{VC} + \sum_{x=a}^c u_{backEMFx} \right] \\ u_{Sc} &= \frac{1}{3} \left[ 2u_{VC} - u_{VA} - u_{VB} + \sum_{x=a}^c u_{backEMFx} \right] \end{aligned} \quad (2-4)$$

where:

$u_{VA} \dots u_{VC}$	Branch voltages between one power stage output and its natural zero
$u_{Sa} \dots u_{Sc}$	Motor-phase-winding voltages
$u_{backEMFa} \dots u_{backEMFc}$	Phase back-EMF induced in the stator winding
$u_0$	Differential voltage between the central point of the star connection of motor winding and the power stage natural zero
$i_{Sa} \dots i_{Sc}$	Phase currents
$R_S$	Phase resistances



$$\begin{aligned}
 u_{VA} - u_{\text{backEMFa}} - \frac{1}{3} \left[ \sum_{x=A}^C u_{Vx} - \sum_{x=a}^c u_{\text{backEMF}x} \right] &= R_s i_{Sa} + L_s \frac{di_{Sa}}{dt} \\
 u_{VB} - u_{\text{backEMFb}} - \frac{1}{3} \left[ \sum_{x=A}^C u_{Vx} - \sum_{x=a}^c u_{\text{backEMF}x} \right] &= R_s i_{Sb} + L_s \frac{di_{Sb}}{dt} \\
 u_{VC} - u_{\text{backEMFc}} - \frac{1}{3} \left[ \sum_{x=A}^C u_{Vx} - \sum_{x=a}^c u_{\text{backEMF}x} \right] &= R_s i_{Sc} + L_s \frac{di_{Sc}}{dt}
 \end{aligned} \tag{2-5}$$

where:

$R_s, L_s$  phase resistances and inductances

The internal torque of the motor itself is defined as:

$$T_i = \frac{1}{\omega} \sum_{x=a}^c u_{\text{backEMF}x} \cdot i_{Sx} = \sum_{x=a}^c \frac{d\psi_{Sx}}{d\theta} \cdot i_{Sx} \tag{2-6}$$

where:

$T_i$  internal motor torque (no mechanical losses)

$\omega, \theta$  rotor speed and rotor position

$\psi_{Sx}$  magnetic flux of phase winding x

### 2.3.3 Back-EMF Sensing

The back-EMF sensing technique is based on the fact that only two phases of a Brushless DC motor are energized at a time (see [Figure 2-2](#)). The third phase is a non-fed phase that can be used to sense the back-EMF voltage.

Let us assume the situation where phases A and B are powered and phase C is non-fed. No current passes through this phase. Assume the following conditions are met:

$$\begin{aligned}
 S_{Ab}, S_{Bt} &\leftarrow \text{PWMswitching} \\
 u_{VA} &= \mp \frac{1}{2} u_d, u_{VB} = \pm \frac{1}{2} u_d \\
 i_{Sa} &= -i_{Sb}, i_{Sc} = 0, di_{Sc} = 0 \\
 u_{\text{backEMFa}} + u_{\text{backEMFb}} + u_{\text{backEMFc}} &= 0
 \end{aligned} \tag{2-7}$$

The branch voltage  $u_{VC}$  can be calculated when considering the above conditions:

$$u_{VC} = \frac{3}{2}u_{\text{backEMFC}} \quad (2-8)$$

Figure 2-6 illustrates that the branch voltage of phase C—between the power stage output C and the natural voltage level—can be sensed. Therefore, the back-EMF voltage is obtained and the zero crossing can be recognized.

The general expression can be found by:

$$u_{Vx} = \frac{3}{2}u_{\text{backEMFx}} \quad (2-9)$$

where:

$$x = A, B, C$$

Two conditions must be met:

- Top and bottom switch (in diagonal) must be driven by the same PWM signal
- No current is going through the non-fed phase used to sense the back-EMF

Figure 2-6 shows branch and motor-phase-winding voltages during a 0–360° electrical interval. Shaded rectangles designate the validity of the equation (2-9). Basically, the back-EMF voltage can be sensed during designated intervals.

This is more difficult in reality because the sensed “branch” voltage also contains some ripples.

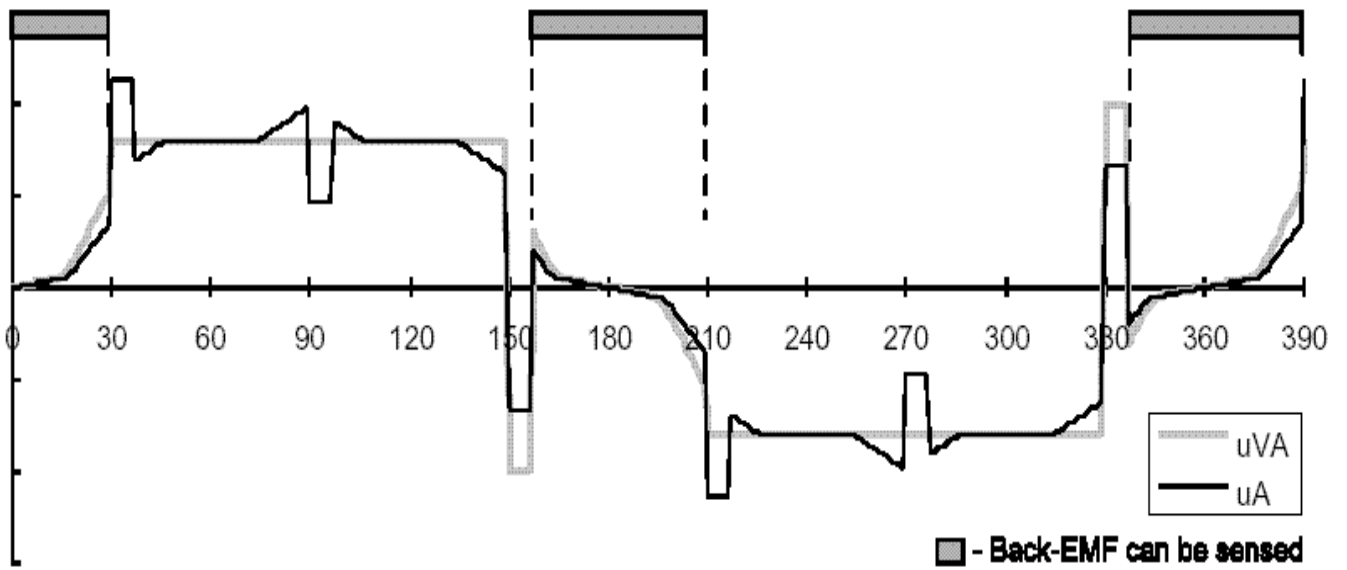


Figure 2-6. Phase Voltage Waveform



# Chapter 3

## System Concept

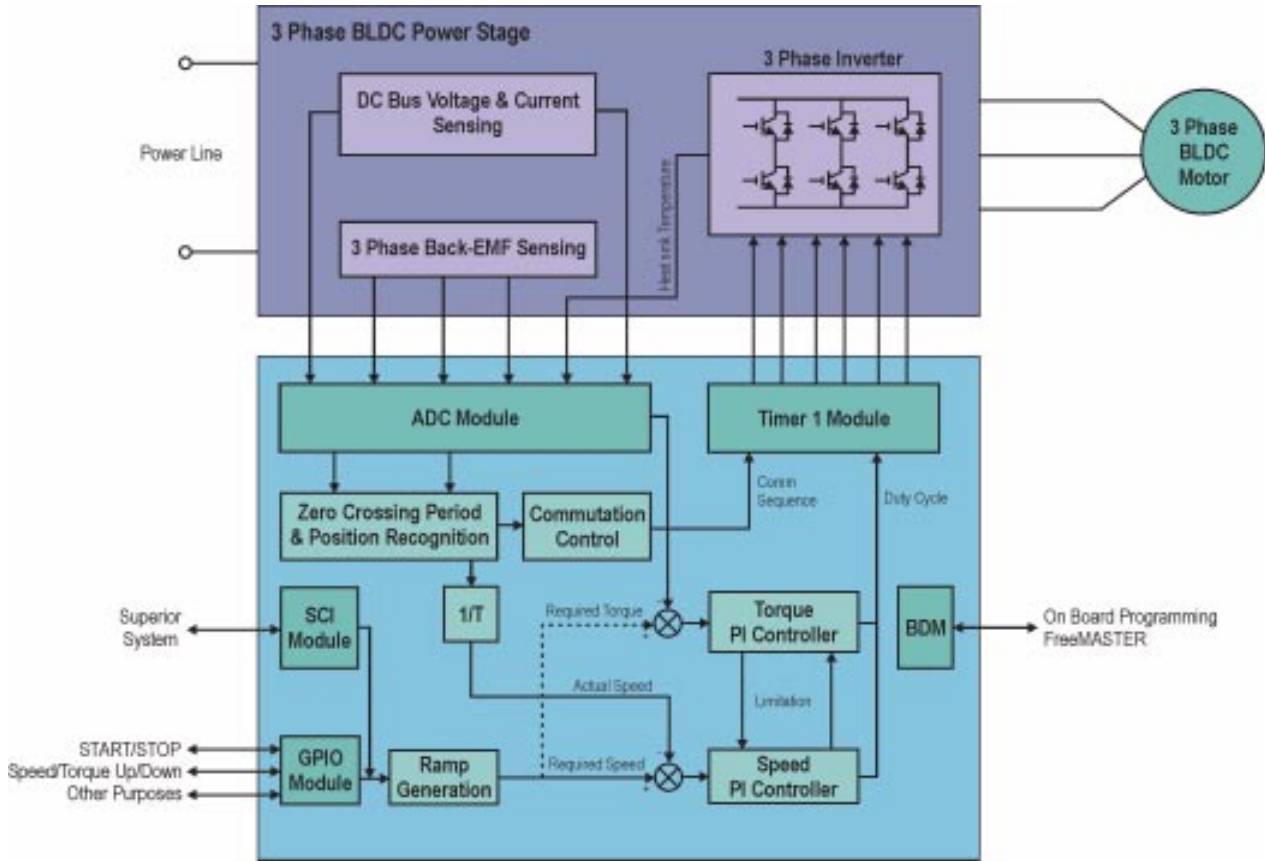
### 3.1 System Specification

The motor-control system is designed to drive a 3-phase, brushless-DC (BLDC) motor in a speed and torque closed loop. The application meets these performance specifications:

- Sensorless brushless DC motor control by back-EMF zero-cross sensing
- Targeted for the MC9S08AWx platform
- Running on a low voltage (12 V) 3-phase, BLDC-control-development platform
- Control technique incorporates:
  - Sensorless control with speed and torque closed loop
  - ADC converter for zero-cross sensing
  - Rotation in both directions
  - Full 4-quadrant operation
  - Start from any motor position with rotor alignment
  - Manual interface (START/STOP switch, up/down push-button control)
  - FreeMASTER software control interface (motor START/STOP, speed/torque setup)
  - FreeMASTER software remote monitor

### 3.2 Sensorless Drive Concept

The system concept is shown below (Figure 3-1).



**Figure 3-1. Drive Concept**

The basic operational theory of the system is as follows:

- The background loop executes the application state machine (ASM), consisting of init, stop, alignment, start up, run and error states.
- After an MCU reset, the ASM goes through the init state to the stop state.
- The START/ STOP switch and up/down buttons are continuously monitored over the GPIO interface.
- As soon as the user turns the START/STOP switch to the START position, the ASM goes into the alignment state, where the motor is aligned to a known position.
- After a predefined time, the ASM continues to the start-up state.
- At start up, the six commutations are performed without any feedback. The timings among the commutations are defined by the start-up table.
- After start up, the ASM goes into the run state, where zero crossing is employed and the control loop is closed.
- During the run state, requests for new speeds and torques are accepted and maintained by the PI controllers.

- If a standstill of the rotor is detected, the ASM goes again into the alignment state and tries to execute a new start-up sequence. In case of any fault, the ASM returns to the error state and then into the stop state.
- After a fault condition occurs, the application switches to the fault state, stops the motor, and waits for the fault reset over the START/ STOP switch. Beyond this, the FreeMASTER interface enables monitoring and adjustment of all system variables.

## 3.3 System Blocks Concept

### 3.3.1 PWM Voltage Generation for Brushless DC Motor

A 3-phase voltage system, as described, must be created to run the BLDC motor. This is provided by 3-phase power stage with six power switches (IGBTs or MOSFETs) controlled by the MC9S08AW60 on-chip TPM/PWM module.

When generating PWM signals for the BLDC-motor-control application, the bottom and top power switches of the non-fed phase must be switched off. (See [Figure 3-2](#) and [Figure 3-3](#))

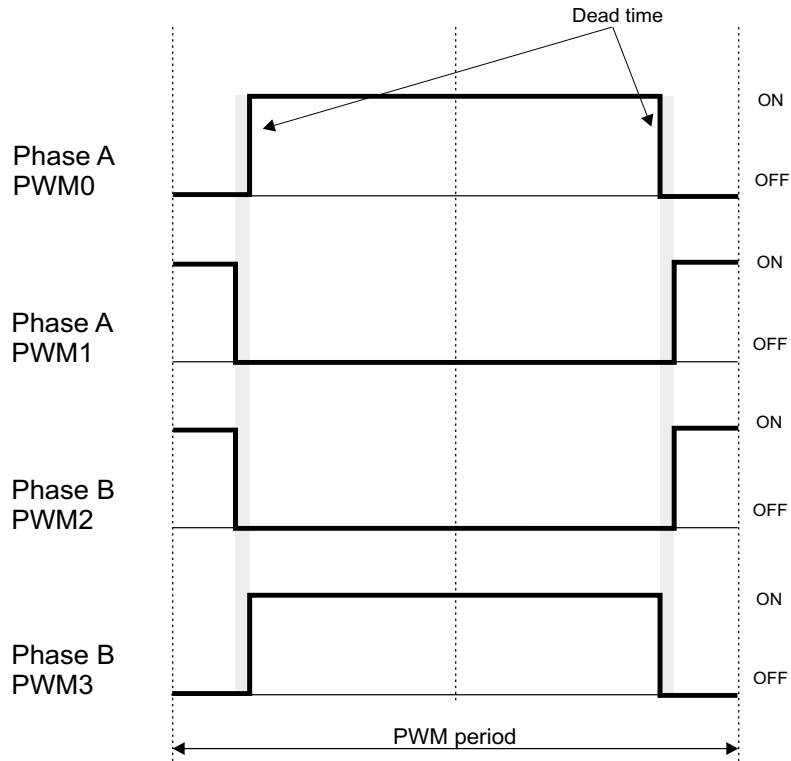
For the BLDC-motor-control application, PWM signals can be created in two ways:

- Complementary PWM Mode
- Independent PWM mode.

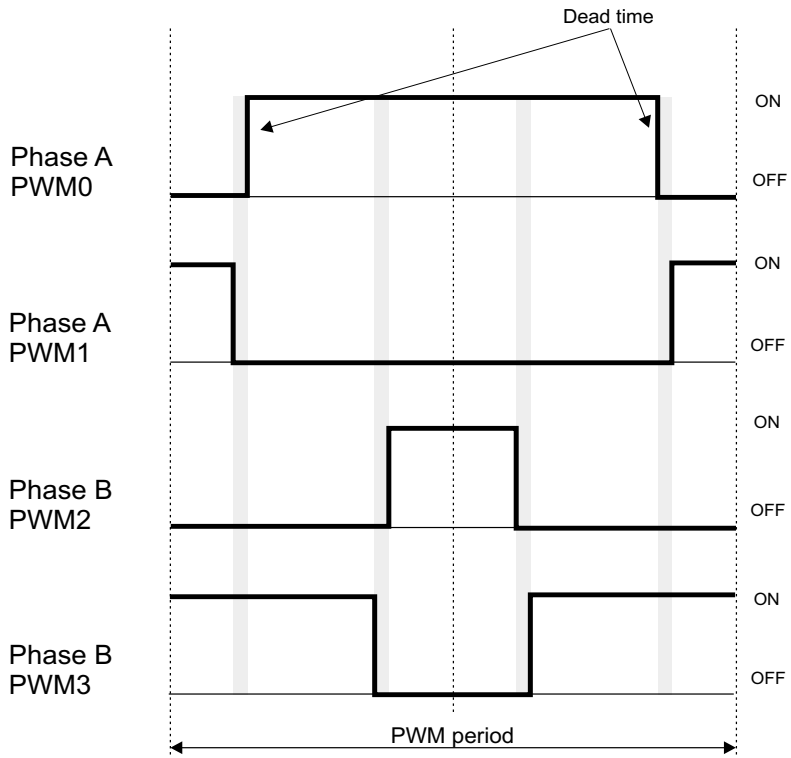
Because of the flexible MC9S08AW60 TPM/PWM module, both modes are available.

#### 3.3.1.1 Complementary PWM Mode

In complementary PWM mode, the top and bottom switches of a phase are operated complementarily: if one switch is on, the other is off. This mode must be used if four quadrant drive operation is required. This mode needs dead time insertion between the top and bottom switches to avoid any phase short-circuit. The complementary switching can be implemented in a bipolar or unipolar manner. The unipolar switching leads to lower switching losses and current ripple. However, from a back-EMF perspective, the bipolar switching is the better choice because this allows a duty cycle in the range of 50 percent to 100 percent. It significantly simplifies the back-EMF voltage and current sensing.



**Figure 3-2. Complementary Bipolar Switching Patterns for BLDC Motor**



**Figure 3-3. Complementary Unipolar Switching Patterns for BLDC Motor**



### 3.3.1.2 Independent PWM Mode

In independent PWM mode, the top and bottom switches of a phase are operated independently over a commutation period. If a top switch performs a PWM, the bottom switch is off, and vice versa. In this mode, the drive can operate in two quadrants; the bipolar and unipolar switching are available.

Using independent bipolar switching has significant drawbacks. The drive can work with discontinuous or continuous currents. During the transition from discontinuous to continuous current, the drive significantly changes its behavior: tuning of the speed controller is very difficult, and the drive exhibits low torque at a discontinuous current. Using unipolar switching is not recommended due to problematic back-EMF voltage and current sensing at narrow pulses.

Figure 3-2 through Figure 3-5 show details of one PWM period for the BLDC motor. Phase A is connected to a positive DC bus voltage, phase B is connected to a negative DC bus voltage. Phase C is non-fed and not shown. The duty cycle is 50 percent.

### 3.3.1.3 Conclusion

The complementary bipolar switching is optimal for the selected back-EMF voltage sensing method when implemented on the MC9S08AW60. The details of back-EMF voltage sensing method are shown in Section 3.3.3. However there is one limitation when the complementary switching is implemented. If the phase is disabled the TPM one pair of the timer channels is disabled by setting of ELSnB:ELSnA bits to zero. If the TPM timer channel is disabled the polarity of output generated signal is unknown since the polarity is defined by ELSnB:ELSnA bits. Therefore the update of TPM channels must be done within first quarter of the PWM period (starting from mid-point of the PWM pulse). The later update may cause generation of the incorrect PWM output during the commutation.

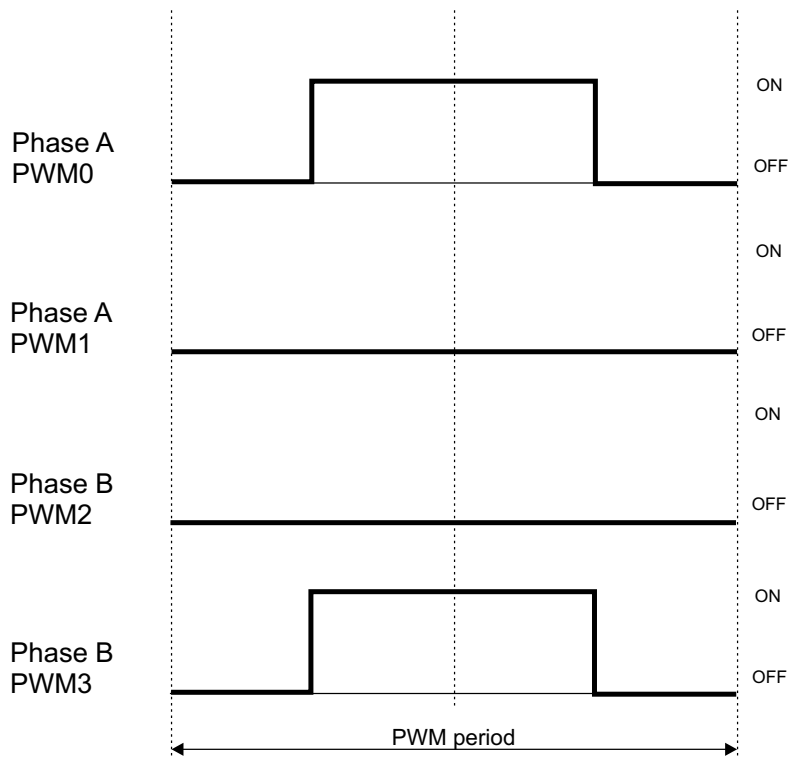
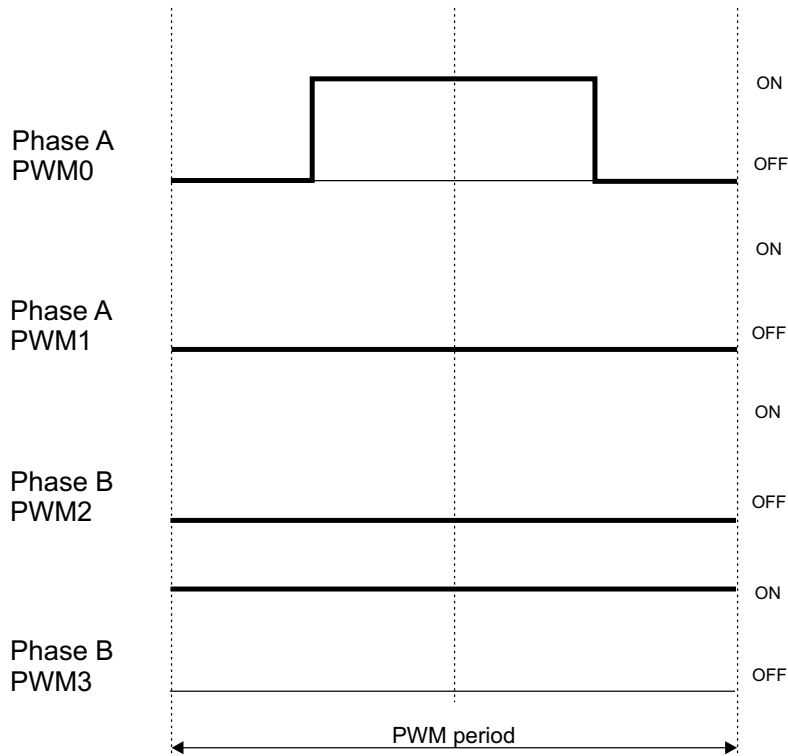


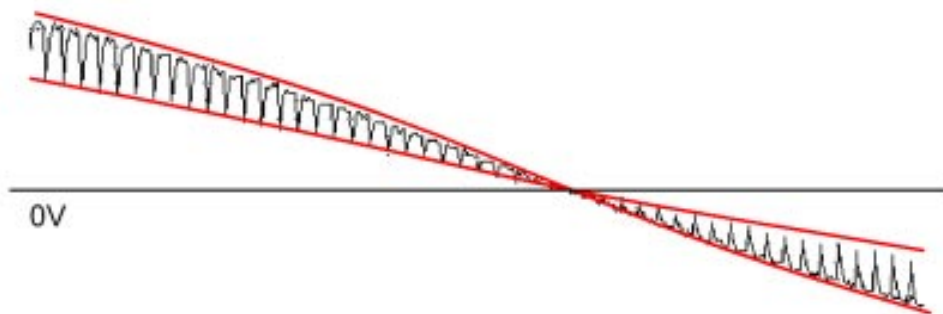
Figure 3-4. Independent Bipolar Switching Patterns for BLDC Motor



**Figure 3-5. Independent Unipolar Switching Patterns for BLDC Motor**

### 3.3.2 ADC Sampling Mechanism

Power stage PWM switching causes high voltage spikes on the phase voltages. These voltage spikes are generated on the non-fed phase because of mutual inductances and mutual capacitor couplings between motor windings. Non-fed phase “branch” voltage is then disturbed by PWM switching. [Figure 3-6](#) shows the effect on the non-fed phase, because of the mutual inductance.



**Figure 3-6. Effect of Mutual Inductance on Back-EMF**

The non-fed phase “branch” voltage is disturbed at the PWM switching edges. Therefore, the presented BLDC-motor-control application synchronizes the back-EMF zero crossing detection with PWM. The analog-to-digital conversion of phase branch voltages is triggered in the middle of a PWM pulse. Then the voltage for back-EMF is sensed at those moments when the non-fed phase branch voltage is stabilized.

The MC9S08AWx is not equipped with synchronization of the TPM/PWM and ADC modules. Therefore several assumptions must be considered during the ADC sampling implementation. The first limiting factor is the choice of PWM modulation. From this perspective, the complementary bipolar switching is the best choice. This PWM switching can be implemented so the duty cycle is in the range of 50 percent to 100 percent at any time of motor operation. It gives a good condition for ADC sampling because at least 1/4 of the PWM period is available for ADC sampling. With unipolar switching, the duty cycle starts from 0 percent. This means that there is a limit on ADC sampling at narrow pulses, regardless of whether hardware synchronization is available. In this case, a different sensing method must be chosen, where the back-EMF voltage is compared to zero instead of half DC-bus voltage. Secondly, the TPM/PWM module must be properly set to ensure ADC sampling at mid-pulse.

See [Figure 3-7](#) for the right setting for the TPM/PWM module. The TPM/PWM module is configured to run in center-aligned mode. In this case, the overflow interrupt event is generated in the middle of each pulse. The overflow interrupt starts the analog-to-digital conversion practically in the middle of the pulse. The result of the conversion can be read on the conversion complete interrupt. Because the A/D converter has a single result register, the conversion of more channels must be serviced in the same interrupt. See [Figure 3-7](#) for the sampling process.

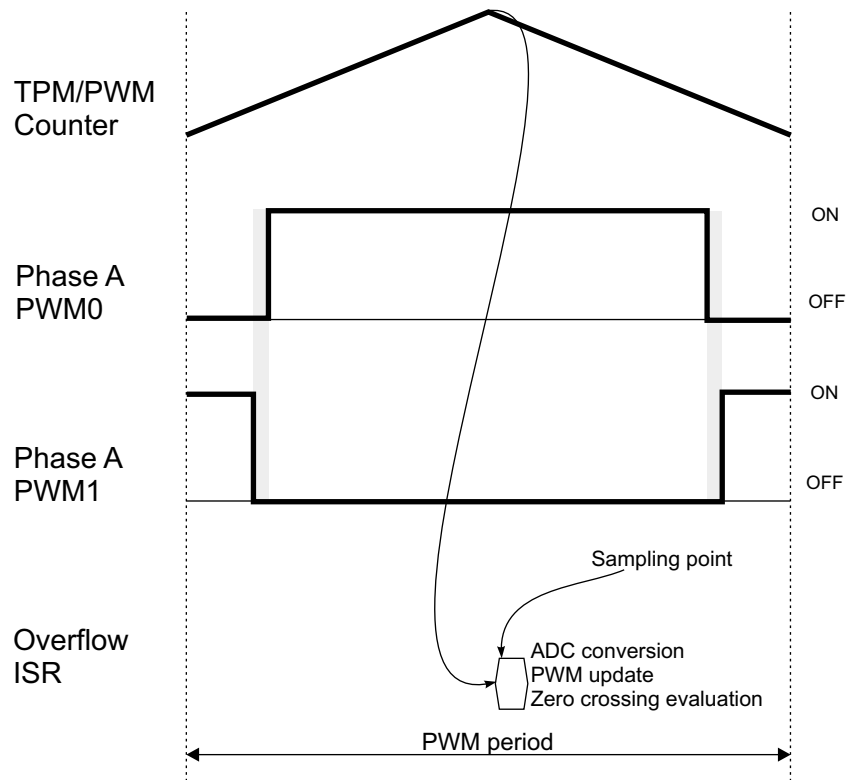


Figure 3-7. TPM/PWM to ADC Synchronization

### 3.3.3 Back-EMF Zero Crossing Sensing

Back-EMF zero-crossing is detected by sensing the motors non-fed phase “branch” voltage ( $u_{vi}$  in [Section 2.3.3](#)) and DC-bus voltage  $u_d$  using the ADC. (Refer to [Section 2](#)).

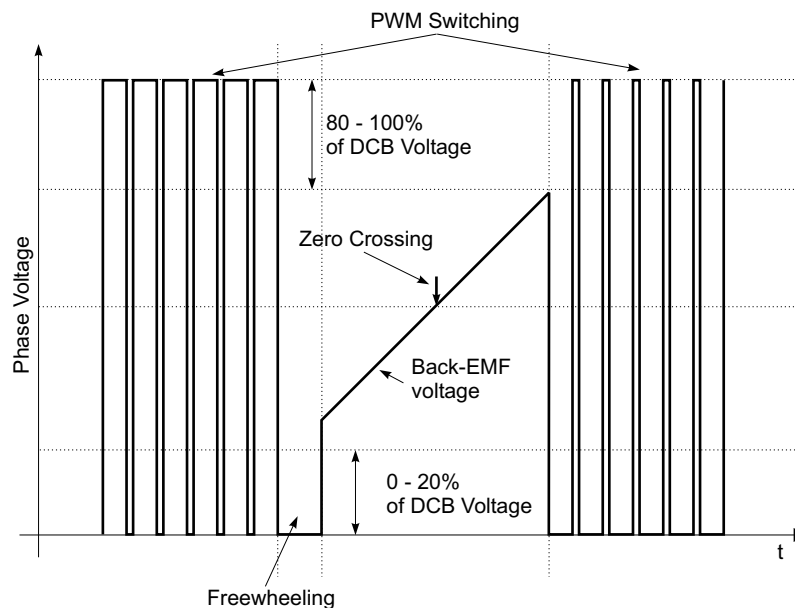
To get the correct back-EMF voltage, two conditions must be met:

- The same PWM signal drives top and bottom switches (in diagonal).
- No current is going through the non-fed phase used to sense the back-EMF.

The first condition is achieved by the TPM/PWM module to ADC synchronization. The second condition can be directly detected from the sensed back-EMF voltage. When the phase disconnects from the DC bus, a current continues flowing through the freewheeling diode. The conduction time depends on the momentary load of the motor. In some circumstances, the conduction time is so long that it disallows the detection of back-EMF voltage. The conduction freewheeling diode connects the released phase to positive or negative DC-bus voltage. The next two steps detect current in the non-fed phase:

1. The first three samples after commutation are not considered for back-EMF voltage detection due to transient event.
2. The back-EMF samples within the range of 0 percent to 20 percent; 80 percent to 100 percent of the DC bus voltage are ignored.

When the back-EMF voltage falls into the range of 20 percent to 80 percent of the DC-bus voltage, evaluation of zero crossing starts. The back-EMF voltage samples are compared with half DC bus voltage in relation to an expected positive or negative slope. If zero crossing is recognized, the new period and commutation event is calculated. The zero crossing detection process can be seen in [Figure 3-8](#).



**Figure 3-8. Zero Crossing Detection Process**

### 3.3.4 Current (Torque) Sensing

Besides the speed-control mode, the drive can run in torque control mode. In this mode the torque is limited to the desired value. In BLDC-motor theory the torque is proportional to motor current. In six-step control, the current amplitude varies during the commutation period. To get a stable value as the actual motor torque, the current is sampled at a zero-crossing event. The resultant value is calculated as the average value from the last six commutation periods. Therefore, the acquired value can be used as the actual motor torque for the torque-control loop. This value cannot be used for current control cycle by cycle. The IGBT driver performs over current limitation.

### 3.3.5 Sensorless Commutation Control

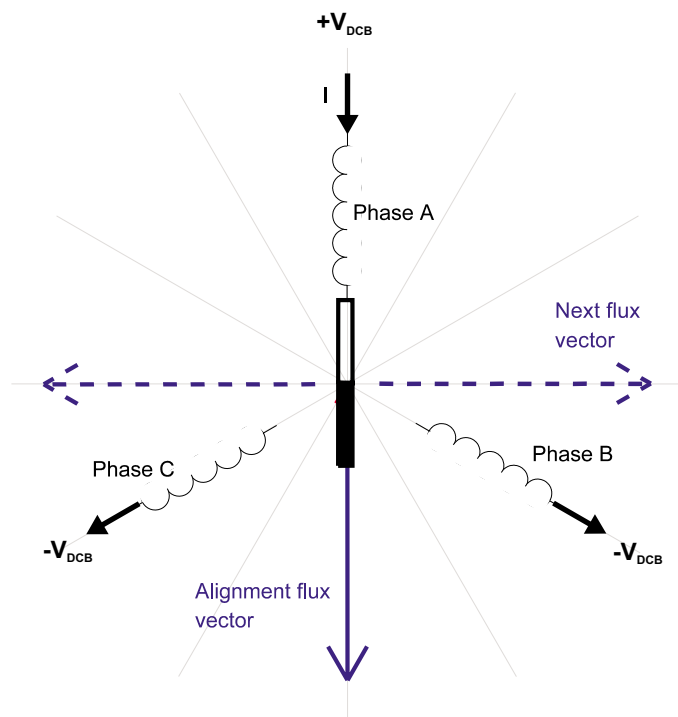
This section presents sensorless BLDC-motor commutation with the back-EMF-zero-crossing technique. To start and run the BLDC motor, the control algorithm must go through these processes:

1. Alignment
2. Start up
3. Stabilization
4. Run

First, the rotor is aligned to a known position without the position feedback. When the rotor moves, the back-EMF is induced on the non-fed phase and sensorless position detection can be employed. As a result, the position is known and can be used to calculate the speed and process the commutation in the run state.

#### 3.3.5.1 Alignment

The main aim of alignment is to align the rotor to a known position. This known position is necessary to start rotation in the proper direction and to generate maximal torque during start up. During alignment, all three phases are powered. Phase A is connected to the positive DCB voltage, and phases B and C are connected to the negative DCB voltage. The resultant rotor position gives the same condition for start up, regardless of rotational direction (see [Figure 3-9](#)). The alignment time depends on the mechanical constant of the motor including load. The time must ensure the rotor steady state at the end of the alignment process.



**Figure 3-9. Rotor Alignment**

#### 3.3.5.2 Start Up

The back-EMF technique can be used when the rotor is moving. The start-up process ensures the initial rotor movement, followed by back-EMF evaluation. The start-up process starts immediately after the

alignment process. The first flux vector applied on the motor is orthogonal to the alignment flux vector. The right flux vector is chosen based on the required direction of rotation (Figure 3-9). The flux vector is applied for a period defined as a constant value during drive tuning. The next five flux vectors follow to create a rotational field. The period timing of all six periods is established by (3-1):

$$T_{\text{NEW}} = k \cdot T_{\text{PREVIOUS}} \quad (3-1)$$

The constant  $k$  must be less than 1; a value within range 0.7 to 1 gives an approximate constant acceleration. Another way is to define a start up table with period timing. The initial period must be tuned based on load inertia.

The start-up tuning is the heaviest task in the back-EMF sensorless technique. The tuning difficulty depends on the load type. On a passive load, where the load torque starts at 0 Nm and grows with the motor speed, it is easy. A more difficult situation can be observed on drives having non-zero starting torque or a variable torque within one revolution such as compressors. If load inertia is variable, the number of starting flux vectors can be changed. A number smaller than six brings about a more robust start up at variable-load inertia, however, a number less than three leads to an unreliable start up. In other cases, the six starting flux vectors (one electrical revolution) seems to be an optimal value.

After the last flux vector executes, back-EMF voltage evaluation is enabled and the motor enters a sensorless mode.

### 3.3.5.3 Stabilization

The stabilization process immediately follows the start-up sequence and is optional. The purpose of this process is to stabilize the measured values. As described in Section 3.3.4, the torque is calculated as the average value from the last six commutations; the speed is calculated in the same manner. Because the periods are unstable during the start-up sequence, the speed and torque values can be wrong. Therefore, the stabilization process executes. This process lasts at least six additional periods, where the motor is kept in an open loop. During these additional periods, the speed and torque values stabilize and the control loop can close.

### 3.3.5.4 Run

After the control loops close, the motor enters the run state. Here, the motor remains at the desired speed or torque, according to the selected operational mode and user commands. Besides controlling the desired quantity, the motor is checked for locked rotor. This might happen when the motor is suddenly overloaded. Standstill detection is performed as a comparison between the actual period and the average period. If the difference is too high for a predefined time, a standstill or incorrect commutation is detected. If the actual period does not correspond to the applied voltage on the motor, the error is indicated. In the case of a standstill or incorrect commutation, the motor stops and the start-up sequence repeats.

# Chapter 4 Hardware

## 4.1 Hardware Implementation

The BLDC sensorless application runs on Freescale's MC9S08AW60 evaluation board, micro-power board, and IB23810 motor.

**NOTE**

*This reference design demonstrates the application using low-voltage power stages but can be adapted to high-voltage power stages and applications.*

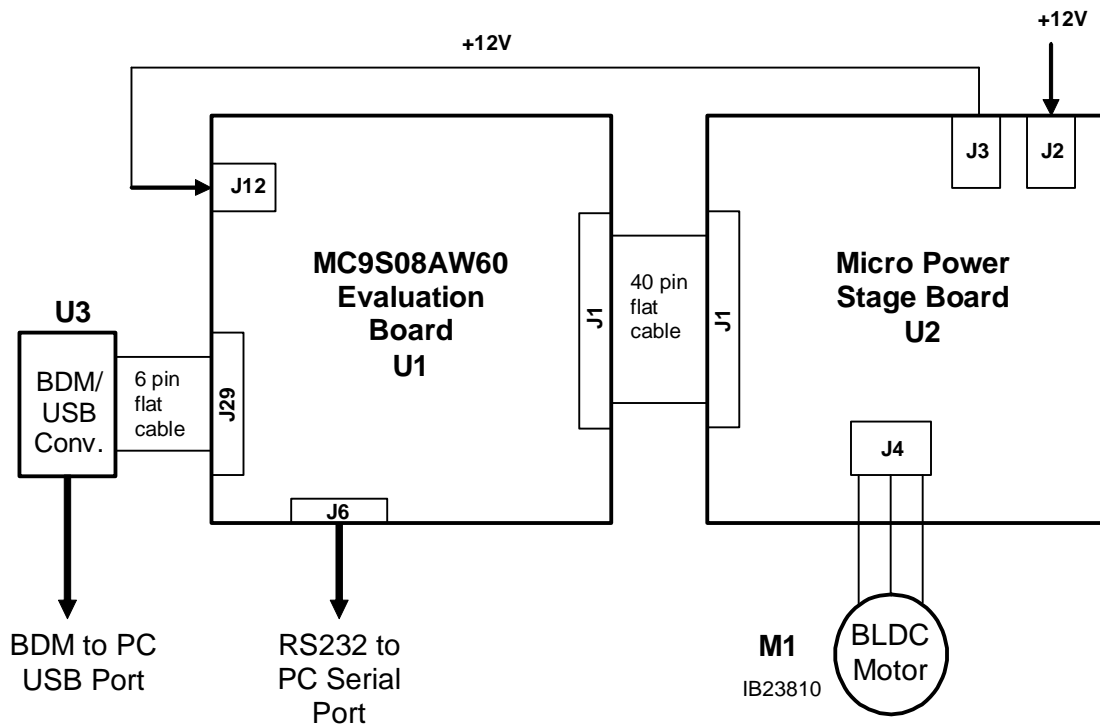


Figure 4-1. Hardware Diagram with Micro-Power Stage

## 4.2 Component Descriptions

### 4.2.1 MC9S08AW60 Evaluation Board

The MC9S08AW60 controller board is a versatile development card for developing real-time software and hardware products to support a new generation of general-purpose, servo, and motor-control applications. The power of the 8-bit MC9S08AW60 MCU combined with the hall-effect/quadrature encoder interface, PWM interface with dead-time insertion circuit, tacho-generator interface for digital/analog sensing, motor over-current logic, and motor over-voltage logic, makes the MC9S08AW60 controller board ideal for developing and implementing many motor-controlling algorithms, as well as for learning the MC9S08AW60 microcontroller architecture and instruction set.

MC9S08AW60 controller board main features:

- MC9S08AW60 8-bit +3.3 V Microcontroller operating at 40 MHz
- BACKGROUND interface header for an external debug host target interface
- RS-232 interface with galvanic isolation for easy connection to a host computer or FreeMASTER development tool
- Header allowing users to attach their own port A GPIO compatible peripheral
- Header allowing users to attach their own port D (GPIO, ADC, KBI, TPM) compatible peripheral
- Header allowing users to attach their own port CE (GPIO, IIC, SCI, SPI) compatible peripheral
- Header allowing users to attach their own ADC compatible peripheral
- Header allowing users to attach their own PWM compatible peripheral
- Header allowing users to attach their own KBI/port G GPIO compatible peripheral
- Header allowing users to attach a ZigBee module or other SCI compatible peripheral
- On-board power regulation from an external 12 V DC supplied power input
- Light emitting diode (LED) power indicator
- Six on-board PWM monitoring LEDs
- One on-board PWM fault monitoring LED
- One on-board general-purpose User LED
- UNI-3 motor interface
  - DC-bus-voltage sensing
  - DC-bus-current sensing
  - Back-EMF sensing
  - Temperature sensing
  - PWM
  - BRAKE, PFC, and PWM signals
- Encoder/hall-effect interface
- Tacho-generator interface
- PWM dead time insertion circuit
- DC-bus-over-current- and over-voltage-protection logic
- DC-bus-current-peak-detector and average-value sensing
- Manual RESET push-button
- General-purpose push-button for UP on GPIO PTG1



- General-purpose push-button for DOWN on GPIO PTG0
- General-purpose toggle switch for START/STOP control on GPIO PTC4

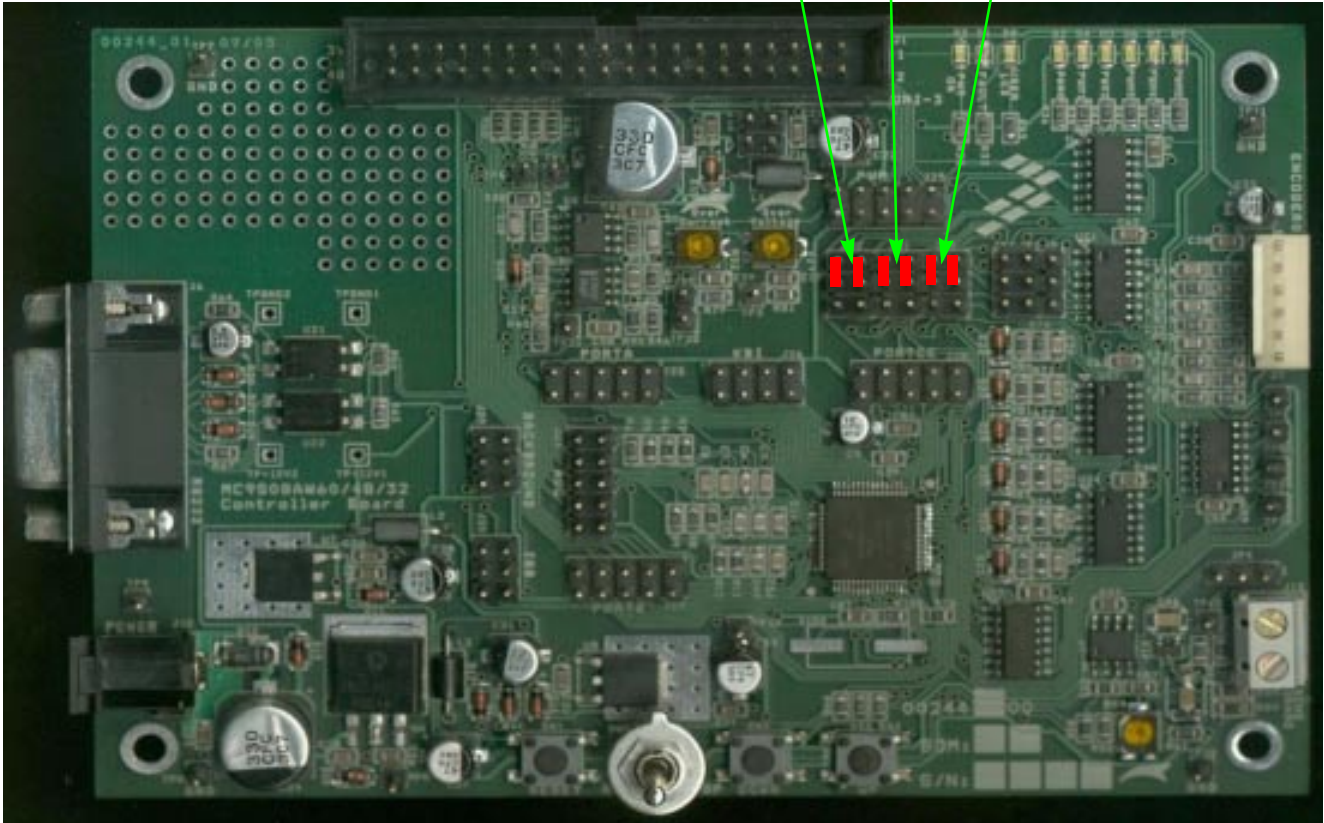
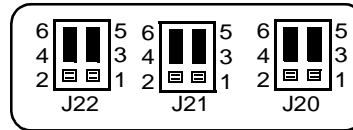


Figure 4-2. MC9S08AW60 Evaluation Board View/ Mandatory Jumper Settings

### 4.2.2 Micro Power Board

Freescale’s embedded motion control series 3-phase, micro-power stage is a 12 V, 3 A, surface-mounted power stage. With one of the embedded motion control series control boards, it provides a software development platform allowing algorithms to be written and tested without designing and building a power stage. It supports algorithms using hall sensors and back-EMF (electromotive force) signals for sensorless control.

The 3-phase, micro-power stage does not have any over-current protection independent of the control board; when using a lower impedance motor, take care in its setup and use. With the motor supplied in the kit, the power-output stage withstands a full-stall condition without the need for over-current protection. Current measuring circuitry is set up for 8.25 A full scale. In a 25°C ambient operation, up to 3A continuous RMS output current is within the board’s thermal limits.

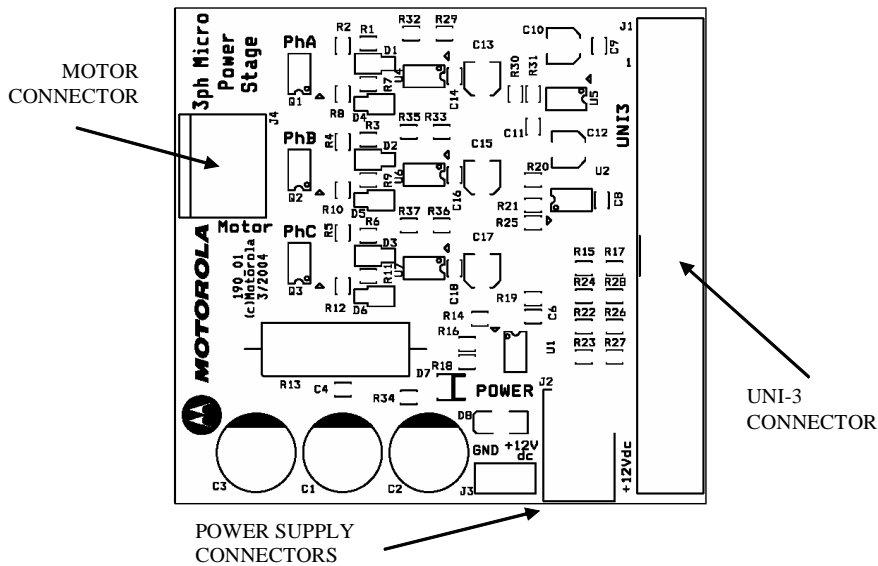


Figure 4-3. Micro-Power Stage View

Input connections are made via the 40-pin ribbon cable connector J1. Power connections to the motor are made on the output connector J4. Phase A, B, and C are labeled on the board. Power requirements are met by a single external 12 V, 4 A power supply. Two connectors (J2 and J3) are provided for the 12 V DC power supply. While J2 is used for 12 V DC input, J3 is used for 12 V DC output to power the controller board. Both are located in a corner of the board.

Table 4-1. Electrical Characteristics of Micro-Power Board

Characteristic	Symbol	Min	Typ	Max	Units
Power-Supply Voltage	V <sub>dc</sub>	10	12	16	V
Quiescent Current	I <sub>CC</sub>	—	18	—	mA
Minimum Logic 1 Input Voltage	V <sub>IH</sub>	2.4	—	—	V
Maximum Logic 0 Input Voltage	V <sub>IL</sub>	—	—	0.8	V
Input Resistance	R <sub>In</sub>	—	10	—	kΩ
Analog Output Range	V <sub>Out</sub>	0	—	3.3	V
Bus Current Sense Voltage	I <sub>Sense</sub>	—	200	—	mV/A
Bus Voltage Sense Voltage	V <sub>Bus</sub>	—	206.25	—	mV/V
Power MOSFET On Resistance (Top Switch)	R <sub>DS(On)</sub>	—	60	70	mΩ
Power MOSFET On Resistance (Bottom Switch)	R <sub>DS(On)</sub>	—	170	200	mΩ
Continuous Output Current	I <sub>D</sub>	—	—	3	A
Pulsed Output Current	I <sub>DM</sub>	—	—	15	A
Total Power Dissipation	P <sub>diss</sub>	—	—	2	W

### 4.2.3 Motor IB23810 (Produced by MCG)

The following motor is used by the BLDC sensorless application. Other motors can be adapted to the application by defining and changing the motor-related parameters. See [Figure 4-2](#) for motor specifications.

**Table 4-2. Electrical Characteristics of MCG IB23810 Motor**

Characteristic	Symbol	Min	Typ	Max	Units
Reference Winding Voltage	$V_t$	—	—	170	V
Speed @ $V_t$	$\omega$	—	—	6000	RPM
Torque Constant	$K_t$	—	0.0840	—	Nm/A
		—	11.90	—	oz-in/A
Voltage Constant	$K_e$	—	8.8	—	V/krpm
Terminal Resistance	$R_t$	0.13	—	0.18	$\Omega$
Winding Inductance	L	—	6.8	—	mH
Continuous Current	$I_{cs}$	—	—	1.8	A
Number of Pole Pairs	$J_m$	—	2	—	—
Temperature Rating	t	-10	—	80	$^{\circ}\text{C}$
	t	14	—	176	$^{\circ}\text{F}$



# Chapter 5

## Software Design

### 5.1 Introduction

This section describes the design of the drive's software blocks. The software description is comprised of these topics:

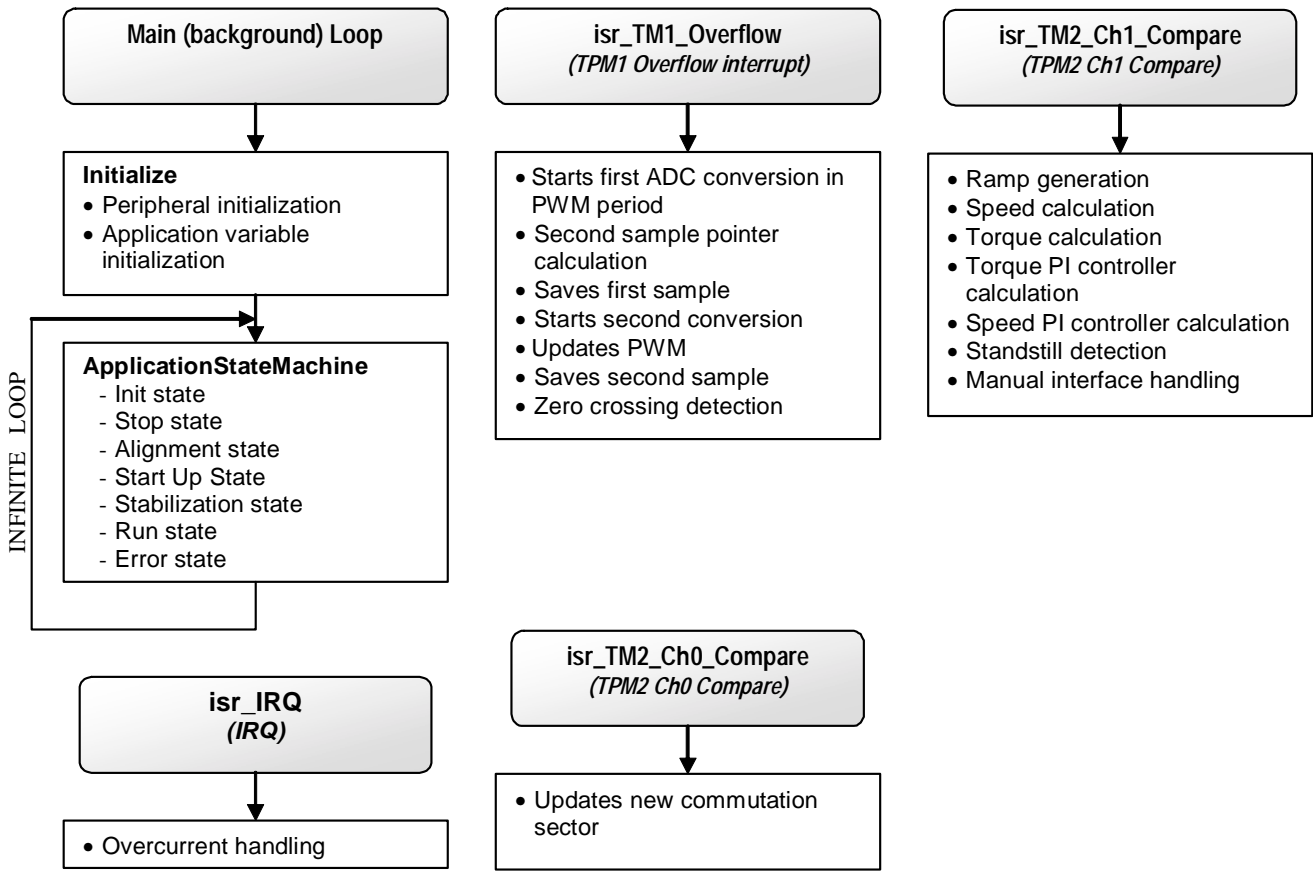
- Main software flow chart ([Section 5.2](#))
- Data flow ([Section 5.3](#))

### 5.2 Main Software Flowchart

The main software flowchart ([Figure 5-1](#)) incorporates the main routine (background loop) entered from reset and interrupt states. The main routine includes the initialization of the microcontroller and the main loop.

After initialization, the main routine enters an endless loop, which executes the application state machine consisting of initialization, stop, alignment, start-up, run, and error states. The transition between the states depends on state of the START/ STOP switch, UP/DOWN buttons, and drive status. In the case of remote control—besides the motor speed/torque—the operational mode and the rotation direction can be set from FreeMASTER. Apart from the background loop, the software incorporates the two periodic and two event interrupts:

- TPM 1 overflow interrupt  
Execution period: 62.5  $\mu$ s  
The routine starts ADC conversions, provides a PWM update and zero-crossing detection
- TPM 2 Ch0 compare interrupt  
Execution period: event interrupt called every commutation  
The routine updates the commutation sector
- IRQ interrupt  
Execution period: event interrupt called in the case of overcurrent detection  
The routine handles the DC bus overcurrent event
- TPM 2 ch1 compare interrupt  
Execution period: 5 ms  
The routine provides ramp generation, speed/torque PI controllers, standstill detection and manual interface handling (UP/DOWN buttons, RUN/STOP switch)



**Figure 5-1. Main Software Flow Chart**

### 5.3 Data Flow

The BLDC motor sensorless control drive control algorithm is described in [Figure 5-2](#) flowcharts.

These variables contain the results of the analog-to-digital conversion:

- PhA\_voltage
- PhB\_voltage
- PhC\_voltage
- *dcb\_voltage*
- *dcb\_current*

The zero-crossing-detection process compares the phase voltage to *dcb\_voltage*/2. The appropriate phase voltage is chosen based on *pwm\_sector*, containing the actual commutation sector. The result of zero-crossing-detection process defines the time difference for the next commutation, saved in *periodNextCommutation*.

For further speed calculation, individual commutation periods are also stored for every commutation in variables:

- *periodZC\_F\_PhA*

- *periodZC\_R\_PhA*
- *periodZC\_F\_PhB*
- *periodZC\_R\_PhB*
- *periodZC\_F\_PhC*
- *periodZC\_R\_PhC*

In every zero crossing event, the value of *dcb\_current* is copied to the variables:

- *torque\_F\_PhA*
- *torque\_R\_PhA*
- *torque\_F\_PhB*
- *torque\_R\_PhB*
- *torque\_F\_PhC*
- *torque\_R\_PhC*

The variable *torque\_Filtered* is calculated as an average value of variables:

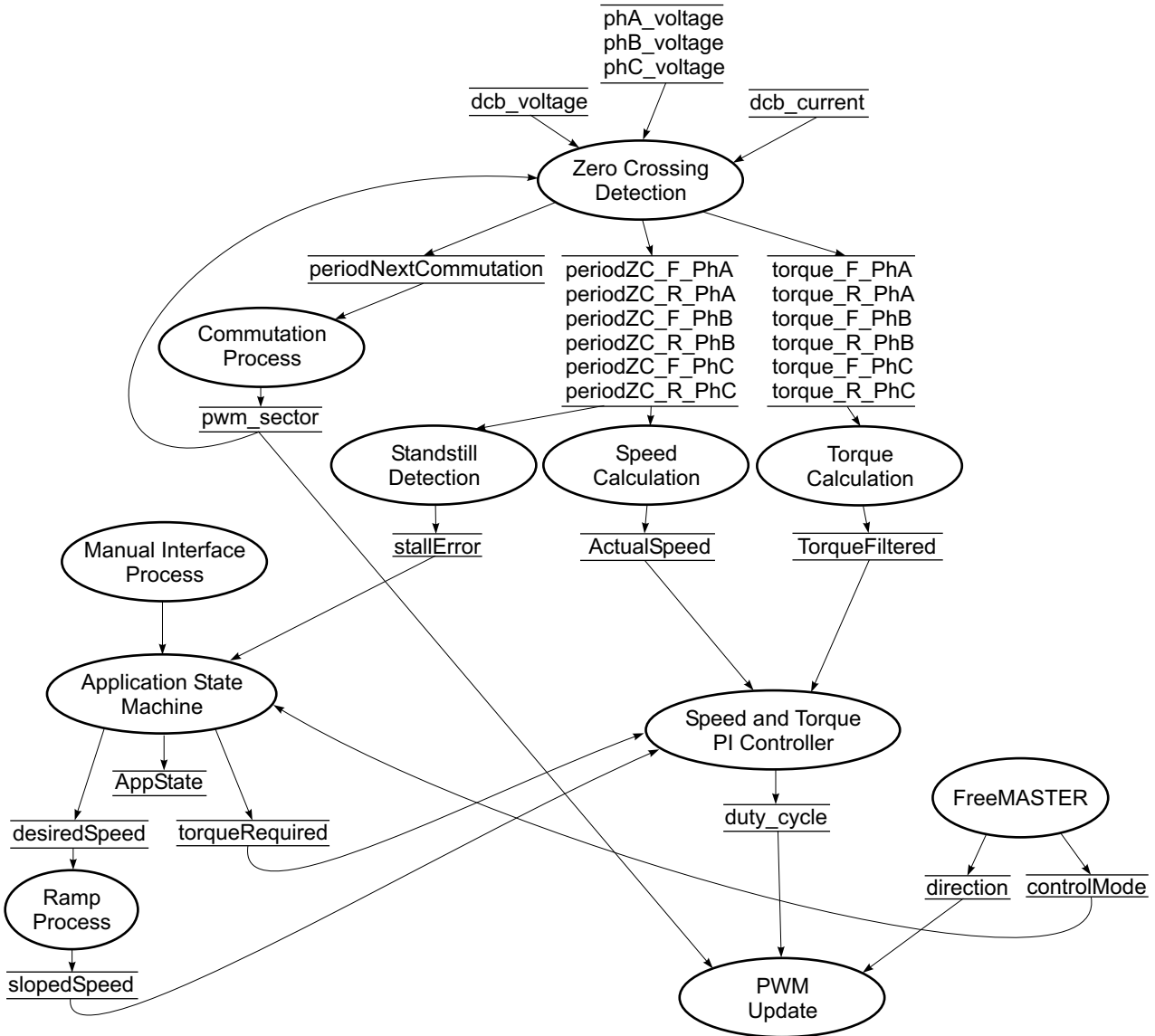
- *torque\_F\_PhA*
- *torque\_R\_PhA*
- *torque\_F\_PhB*
- *torque\_R\_PhB*
- *torque\_F\_PhC*
- *torque\_R\_PhC*

The commutation period is also filtered from variables (and recalculated as the motor speed *actual\_speed*):

- *periodZC\_F\_PhA*
- *periodZC\_R\_PhA*
- *periodZC\_F\_PhB*
- *periodZC\_R\_PhB*
- *periodZC\_F\_PhC*
- *periodZC\_R\_PhC*

Both values are input into the speed and torque PI controllers as actual values. The required values *desiredSpeed/slopedSpeed* and *torqueRequired* are updated based on commands from the user. A ramp limits the required speed to avoid exceeding the maximum motor current. The resulting speed and torque PI Controllers *duty\_cycle* combined with the actual commutation sector *pwm\_sector* and the required direction of rotation *direction*, define the duty cycle of every PWM channel.

The application state machine sets its state *AppState* based on inputs from the manual interface, standstill detection, FreeMASTER interface, and state variables not mentioned in main data flowchart.



**Figure 5-2. Data Flowchart**

## 5.4 Processes Description

The main processes of the BLDC-motor-sensorless-drive-application can be in [Figure 5-2](#).

### 5.4.1 Process Zero-Crossing Detection

This process is called every 62.5  $\mu$ s in `isr_TPM1_Overflow` interrupt. The process is called after the second analog-to-digital conversion, when a new sample of phase voltage is available. This new sample is checked for appropriate voltage level and compared to the midpoint of the DC bus voltage.

[Section 3.3.3](#) describes the zero-crossing detection. If zero crossing is detected, the new commutation



period is calculated and saved for speed calculation. The new commutation event is then calculated and set to the timer TPM1 channel 0 as a new compare event.

### 5.4.2 Process Torque and Speed Calculation

Both torque and commutation periods are filtered to get suitable input values to the torque and speed PI controllers. Both values are calculated as the averages from the last six commutation events. Then the averaged commutation period is recalculated as the motor speed. This process executes every 5 ms in the `isr_TM2_Ch0_Compare` interrupt.

### 5.4.3 Process Standstill Detection

This process compares all six commutation periods. If the difference between periods is higher than defined, a standstill is detected. It also detects incorrect commutation during start up or loss of synchronization between commutation sector and sensed back-EMF voltage. At any error the motor is stopped and the start up sequence is repeated. This process executes every 5 ms in the `isr_TM2_Ch0_Compare` interrupt.

### 5.4.4 Process Commutation

This process updates the variable `pwm_sector` only. The PWM update based on the new commutation sector is performed in the PWM Update process. The commutation process is called on every new commutation event. Therefore, the execution period depends on the motor speed.

### 5.4.5 Process Manual Interface

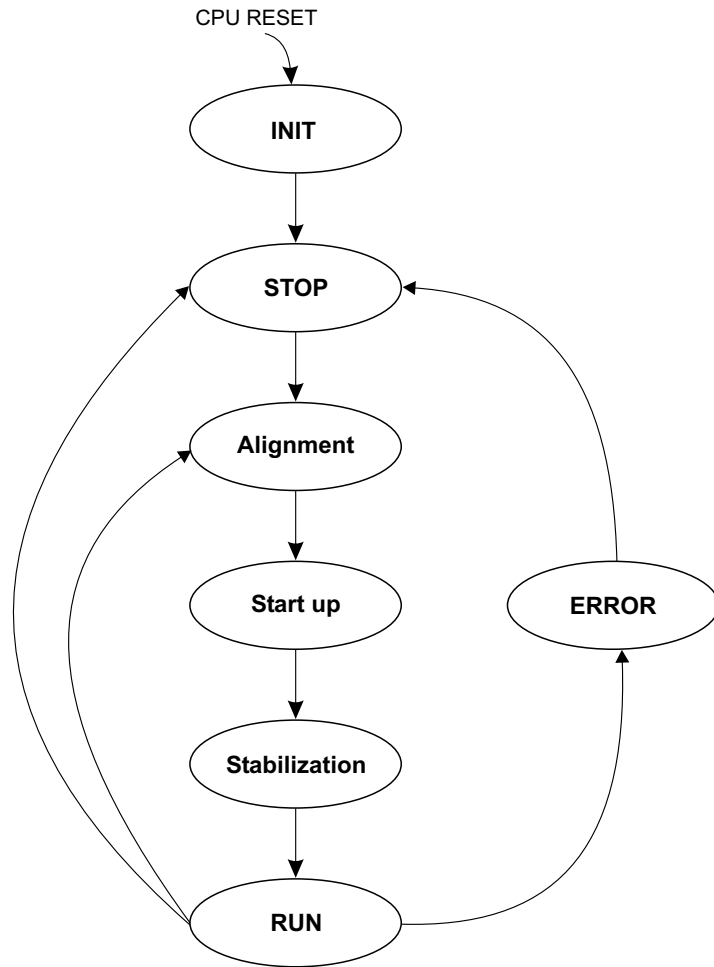
This process checks the state of the input pins where the START/STOP switch and UP/DOWN buttons are connected. If a change is detected the particular status bit is set. These status bits are evaluated in the application state machine. The process executes every 5 ms in `isr_TM2_Ch0_Compare` interrupt.

### 5.4.6 Process Application State Machine

The application state machine (ASM), consists of init, stop, alignment, start-up, stabilization, run, and error states ([Figure 5-3](#)). After an MCU reset, the ASM goes through the init state to the stop state.

1. As soon as the user turns the START/STOP switch to the START position, the ASM goes into the alignment state, where the motor is aligned to a known position.
2. After a predefined time, the ASM continues to the start-up state.
3. At the start-up state, the six commutations perform without feedback. The start-up table defines the timings between the communications.
4. After the start-up state, the ASM moves into the stabilization state, where the measurements of the torque and speed are stabilized. This state goes through the next six commutations, where zero crossing is already employed.
5. After the stabilization state, the ASM moves into the run state, where the control loop is closed.
6. During the run state, the PI controllers accept and maintain requests for new speeds and torques.

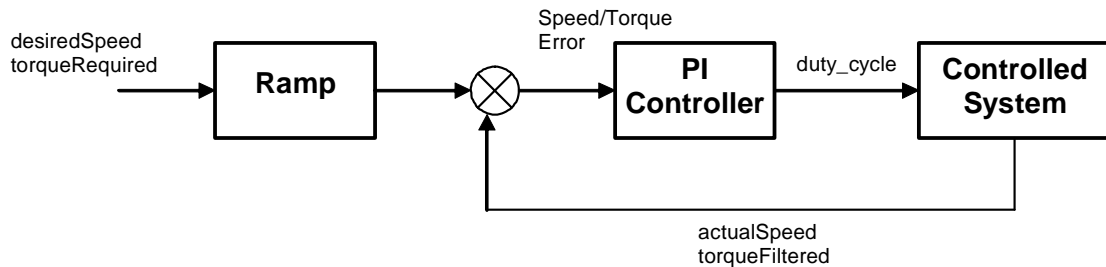
If a standstill of the rotor is detected, the ASM returns into the alignment state and tries to execute a new start-up sequence. In case of any fault, the ASM moves into the error state and then into the stop state. The ASM executes in the background loop.



**Figure 5-3. Application State Machine**

**5.4.7 Process Speed and Torque PI Controller**

Figure 5-4 shows the general principle of the speed PI control loop.



**Figure 5-4. Closed-Loop Control System**

The speed and torque closed loop control is characterized by the actual motor speed and torque feedback. This information is compared with the reference set point, and the error signal generates. The

magnitude and polarity of the error signal corresponds to the difference between the actual and desired speed/torque. Based on the speed/torque error, the PI controller generates the corrected motor voltage to compensate for the error.

The application allows operation in two modes: speed and torque. Independent of the operational mode, the speed and torque PI controllers are calculated. The operational mode defines the required values of the speed and torque at start-up only. During start-up, if the drive runs in speed-operational mode, the required speed is set to the minimal speed and torque to the allowed maximum. Vice versa, in torque-operational mode, the required torque is set to the minimum torque, and the required speed is set to the maximum speed. The lesser value of the PI controllers is chosen as the duty cycle and is applied to the motor. To avoid the saturation of the integral parts of the PI controllers, these parts are limited. If the speed-PI-controller output is applied to the output, the torque PI-controller integral part is limited by the speed PI controller integral part, and vice versa. This type of cross limitation allows simultaneously running the drive under speed and torque limitations. The speed/torque PI controllers' proportional and integral constants were set experimentally. The PI controllers execute every 5 ms in the `isr_TM2_Ch0_Compare` interrupt.

#### 5.4.8 Process Ramp

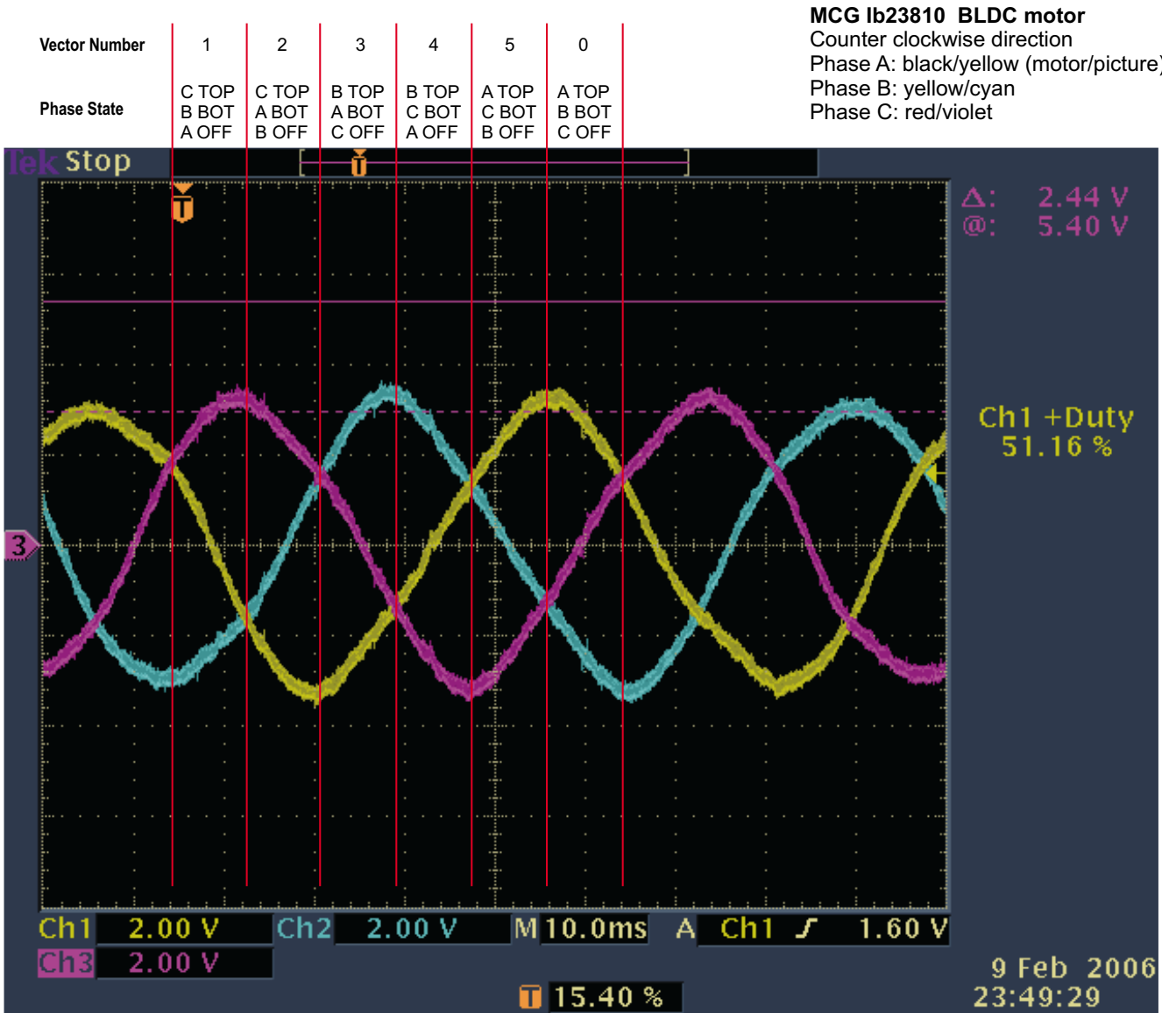
The ramp process slopes some quantities to ensure a smooth motor rotation. The ramp process is used at start up to slope the alignment voltage and for the required motor speed. The process is executed every 5 ms in the `isr_TM2_Ch0_Compare` interrupt.

#### 5.4.9 Process FreeMASTER

The FreeMASTER process is not part of the application software. The BDM interface fully implements FreeMASTER communication, so no additional software support is necessary to provide remote control. FreeMASTER writes new values directly into data RAM, where particular variables are located. This operation is hidden from the application due to non-intrusive debug support for the MC9S08AW60 via the BDM interface.

#### 5.4.10 Process PWM Update

The PWM-update process generates the correct voltage pattern on the motor based on the actual PWM sector, direction of rotation, and required output voltage. The process is called every 62.5  $\mu$ s in `isr_TM1_overflow`. The commutation table derives from back-EMF voltage measured on the motor (Figure 5-5.). Besides the proper commutation pattern, the back-EMF sensing can be evaluated from Figure 5-5. For example, the falling back-EMF voltage of phase A must be evaluated for zero-crossing if phase C connects to positive DC-bus voltage, and phase B connects to negative DC-bus voltage. The resultant commutation table includes sensing of the back-EMF voltage, as shown in Table 5-1 for counter-clockwise direction, and in Table 5-2 for clockwise direction.



**Figure 5-5. Motor Back-EMF Measurement and Evaluation**

**Table 5-1 Commutation for Counter-Clockwise Direction**

Vector Number	Phase			Voltage Sensing
	A	B	C	
0	DCB+	DCB-	NC	Phase C rising
1	NC	DCB-	DCB+	Phase A falling
2	DCB-	NC	DCB+	Phase B rising
3	DCB-	DCB+	NC	Phase C falling
4	NC	DCB+	DCB-	Phase A rising
5	DCB+	NC	DCB-	Phase B falling

**Table 5-2 Commutation for Clockwise Direction**

Vector Number	Phase			Voltage Sensing
	A	B	C	
5	DCB+	NC	DCB-	Phase B rising
4	NC	DCB+	DCB-	Phase A falling
3	DCB-	DCB+	NC	Phase C rising
2	DCB-	NC	DCB+	Phase B falling
1	NC	DCB-	DCB+	Phase A rising
0	DCB+	DCB-	NC	Phase C falling

## 5.5 Constant Calculation

### 5.5.1 Alignment Constants

```
#define ALIGNMENT_TIME          100
```

---

This value multiplied by 5 ms gives the alignment time:

```
#define ALIGNMENT_VOLTAGE      43
```

---

This value defines the alignment voltage level. The value is tuned experimentally. The conversion to volts can be done as:

$$\frac{2 \cdot \text{ALIGNMENT\_VOLTAGE}}{\text{PWM\_MODULO}} \cdot \text{DC-bus voltage} \quad (5-1)$$

### 5.5.2 Start-Up Constants

```
#define START_PERIOD           6000
```

---

This value defines the first commutation period during start up. The value is tuned experimentally. The conversion to the time in seconds can be done as:

$$\text{START\_PERIOD} \cdot \frac{\text{TPM2\_PRESCALER}}{\text{TPM2\_INPUT\_CLOCK}} \quad (5-2)$$

where:

TPM2\_PRESCALER = 128

TPM2\_INPUT\_CLOCK = 20 MHz

Note: The first period is calculated as START\_PERIOD/2. The constant START\_PERIOD is used for calculating of other periods during start up.

```
#define START_ACCELERATION          243
```

---

This value defines other commutation periods during start up (see [Section 3.3.5.2](#)). The value is calculated as a constant from the range (0,1) multiplied by 256.

Example:

If the acceleration constant is set at 0.95, and START\_PERIOD is set at 6000, then START\_ACCELERATION is 243 and start-up periods are calculated as: 3000 (see note at START\_PERIOD), 5700, 5415, 5145, 4887, and 4643,

```
#define START_SPEED                 1610
```

---

This constant defines the motor speed after the start up sequence. For scaling, see [Section 5.5.3](#)

```
#define START_TORQUE                10
```

---

This constant defines the motor current (torque) after the start up sequence. For scaling, see [Section 5.5.4](#)

```
#define START_CMT_NUMBER           6
```

---

This constant defines the number of start-up-commutation periods (see [Section 3.3.5.2](#)).

```
#define STABILIZATION_CMT_NUMBER   6
```

---

This constant defines the number of stabilization commutation periods (see [Section 3.3.5.3](#)).

### 5.5.3 Speed Constants and Scaling

The motor speed scaling for motor IB23810 can be calculated as:

$$\omega_{[\text{SCALED}]} = \frac{16777216 \cdot \text{TPM2\_PRESCALER} \cdot 6 \cdot p \cdot \omega_{[\text{RPM}]}}{\text{TPM2\_INPUT\_CLOCK} \cdot 60 \cdot 2} \quad (5-3)$$

where:

TPM2\_PRESCALER = 128

TPM2\_INPUT\_CLOCK = 20 MHz

p = number of pole pairs (2 for motor IB23810)

```
#define MAX_SPEED          12880    // 1200 RPM
```

---

This constant defines the maximum required motor speed.

```
#define MIN_SPEED          1610     // 150 RPM
```

---

This constant defines the minimal required motor speed.

```
#define SPEED_STEP         1610     // 150 RPM
```

---

This constant defines the step increment for the required motor speed. This increment is used for increasing/decreasing the required speed when the UP/DOWN buttons are used.

```
#define SPEED_RAMP_INCREMENT 50
```

---

### 5.5.4 Torque Constants and Scaling

The current scaling on the micro power stage can be calculated as:

$$I_{[SCALED]} = I_{[A]} \cdot \frac{0.2V/A}{3.3V} \cdot 1024 \quad (5-4)$$

```
#define MAX_TORQUE          124    // 2 Amps
```

---

This constant defines the maximal required motor current (torque).

```
#define MIN_TORQUE          12    // 0.2 Amps
```

---

This constant defines the minimal required motor current (torque).

```
#define TORQUE_STEP         6    // 0.1 Amps
```

---

This constant defines the step increment for the required motor current (torque). This increment is used for increasing/decreasing the required motor current (torque) when the UP/DOWN buttons are used.

### 5.5.5 DC-Bus Voltage Scaling

The DC-bus voltage scaling on the micro-power stage can be calculated as:

$$V_{DCB[SCALED]} = V_{DCB[V]} \cdot \frac{0.20625V/V}{3.3V} \cdot 1024 \quad (5-5)$$





# Chapter 6

## Application Setup

### 6.1 Hardware Setup

The sensorless BLDC demo is mounted onto a piece of acrylic glass that provides a fixed base for the BLDC motor, the micro-low-voltage power stage, and the MC9S08AW60 controller board (see [Figure 6-1](#)).



**Figure 6-1. Sensorless BLDC Demo**

To operate the sensorless BLDC demo:

1. Plug the low-voltage output cable of the power supply to the micro-power-stage connector J2.
2. Connect the BDM cable to connector J29 on the controller board. The USB(PARALLEL)/BDM convertor is necessary only for remote control from the PC and software update.
3. Connect the USB/PARALLEL cable to the PC.
4. Check the jumper settings of J20, J21, and J22 on the controller board according to [Figure 6-4](#).
5. Connect the power supply to the main input cable, the power LED D7 on the micro power stage,

and D3 on the controller board lights up.

The demo setup is shown in [Figure 6-2](#).



**Figure 6-2. Sensorless BLDC Demo Setup**

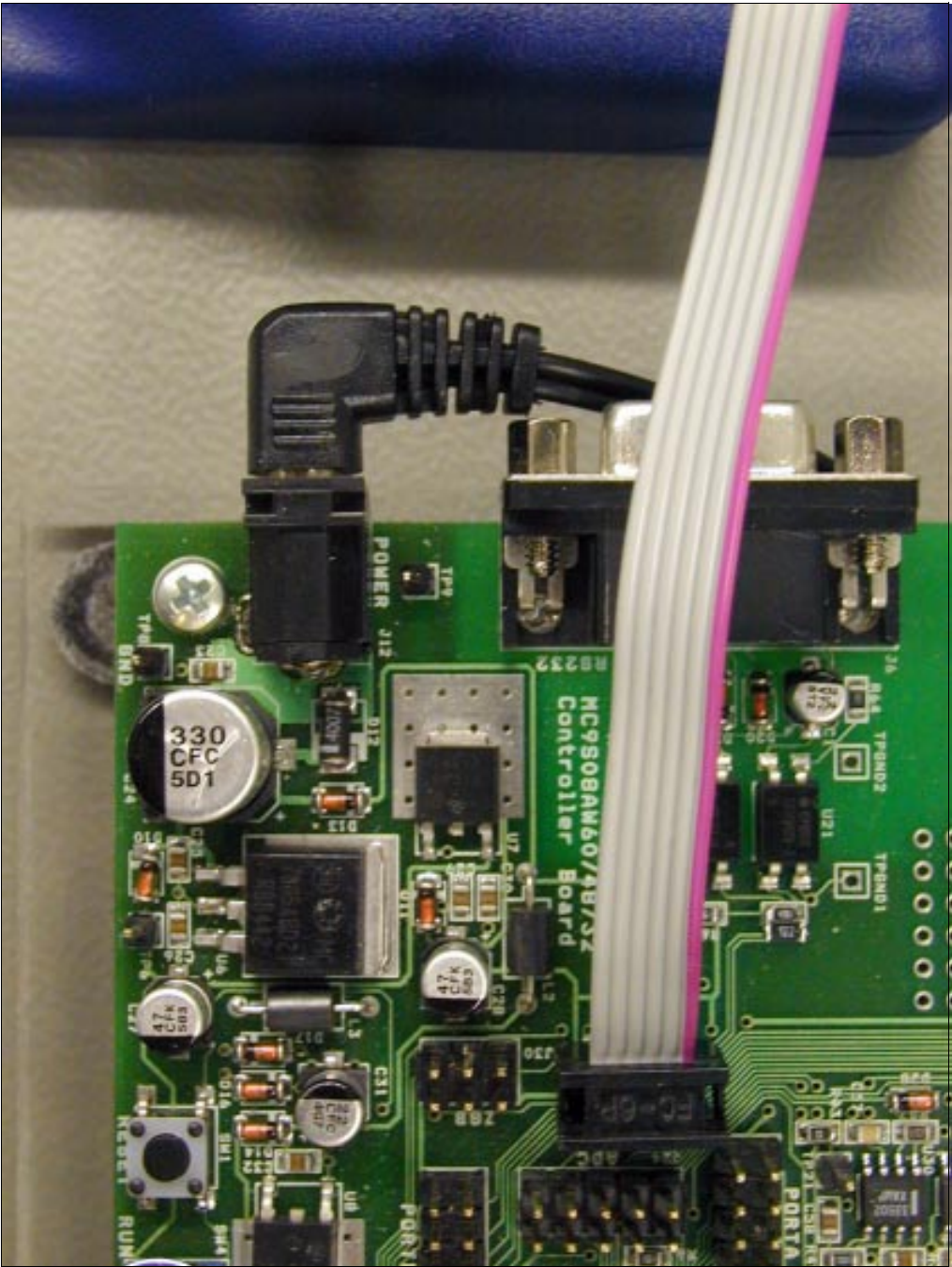


Figure 6-3. Detail of BDM Connection



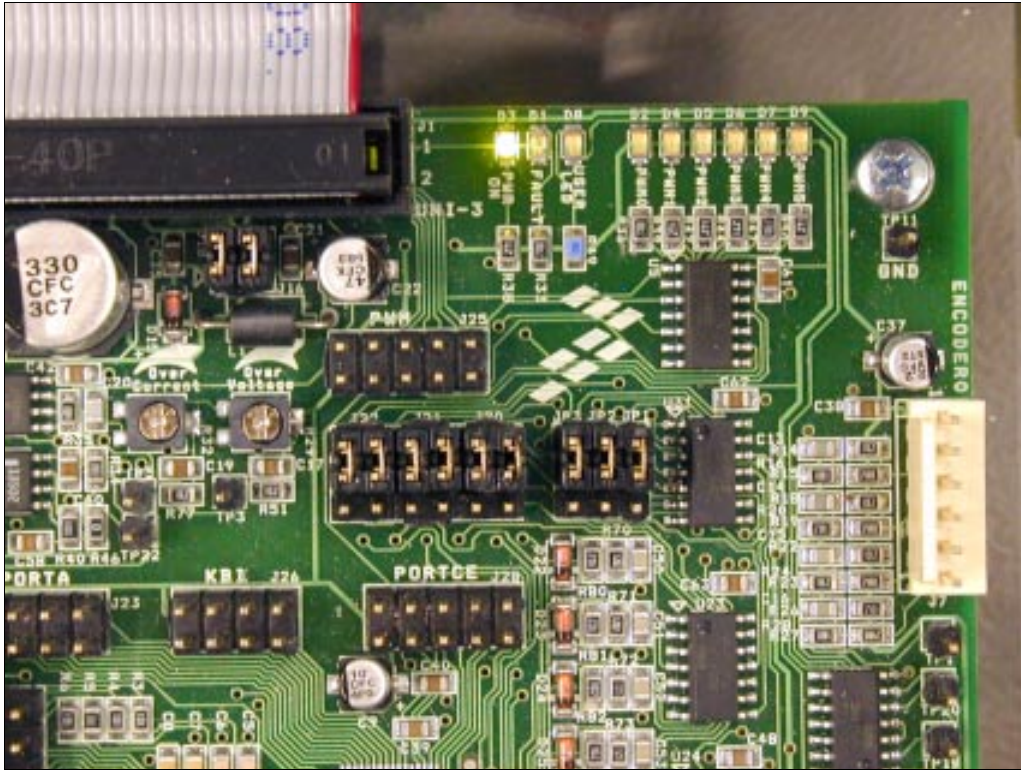


Figure 6-4. Jumper Setup

## 6.2 Software Setup

### 6.2.1 Application Software Files

The application software files are:

- `...\software\DEMO\bldc_AW60.mcp`, application project file
- `...\software\DEMO\sources\main.c`, main program
- `...\software\DEMO\sources\main.h`, header file for main code
- `...\software\DEMO\sources\math.c`, library of mathematical functions
- `...\software\DEMO\sources\math.asm`, library of mathematical functions
- `...\software\DEMO\sources\math.h`, header file for mathematical functions
- `...\software\DEMO\sources\START08.C`, start up code for HCS08 stationary

### 6.2.2 Application FreeMASTER Software Control Files

The application FreeMASTER software control files are:

- `...\software\DEMO\Freemaster\BLDC_BEMF_DEMO_S08AW60.pmp`, FreeMASTER software project file
- `...\software\DEMO\Freemaster\source`, directory with FreeMASTER software control page files

### 6.2.3 Application Build

To build the sensorless BLDC demo application, open the BLDC\_AW60.mcp project file and execute the *Make* command (Figure 6-5). This builds and links the application with all the required CodeWarrior™ libraries.

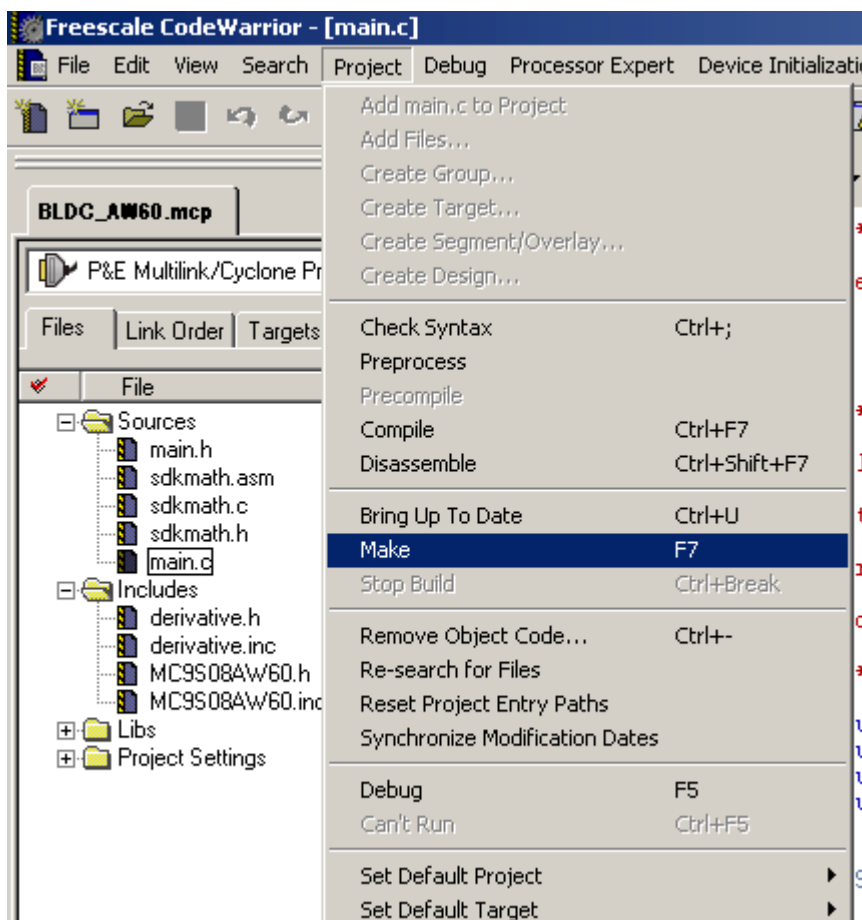
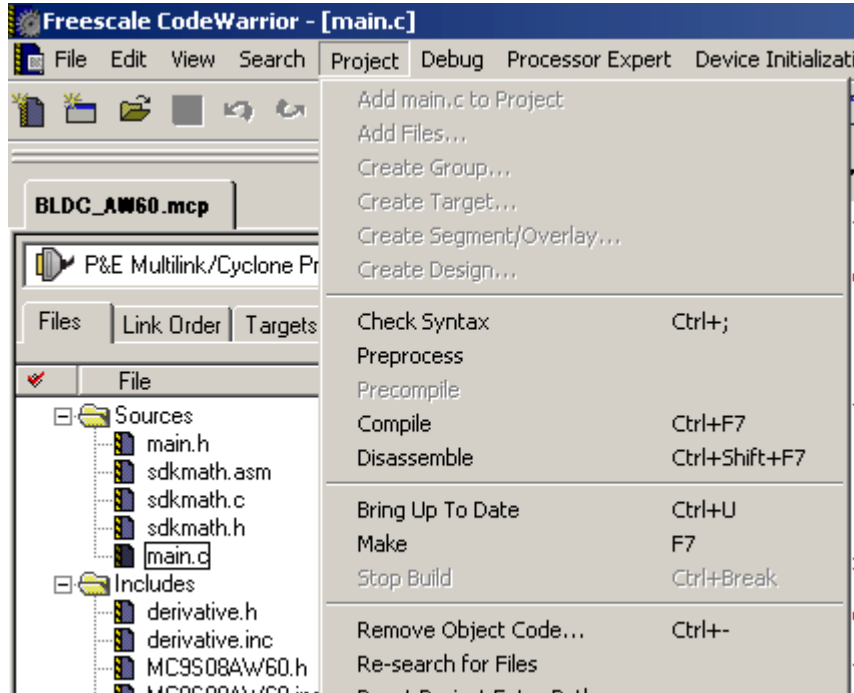


Figure 6-5. Execute Make Command

### 6.2.4 Programming the MCU

To download the software into microcontroller, use an USB/PARALLEL to BDM converter. Before downloading new software, stop the motor by the START/STOP switch and execute the *Debug* command as shown in [Figure 6-6](#).



**Figure 6-6. Execute Debug Command**

## Chapter 7 Sensorless BLDC Demo Operation

### 7.1 Application Control

The BLDC motor control application can be controlled by the manual interface, consisting of the START/STOP switch and UP/DOWN buttons (Figure 7-1) or remotely by PC. The BLDC drive can operate in two speed and torque operational modes.

### 7.2 BLDC Drive Operational Modes

#### 7.2.1 Speed Operational Mode

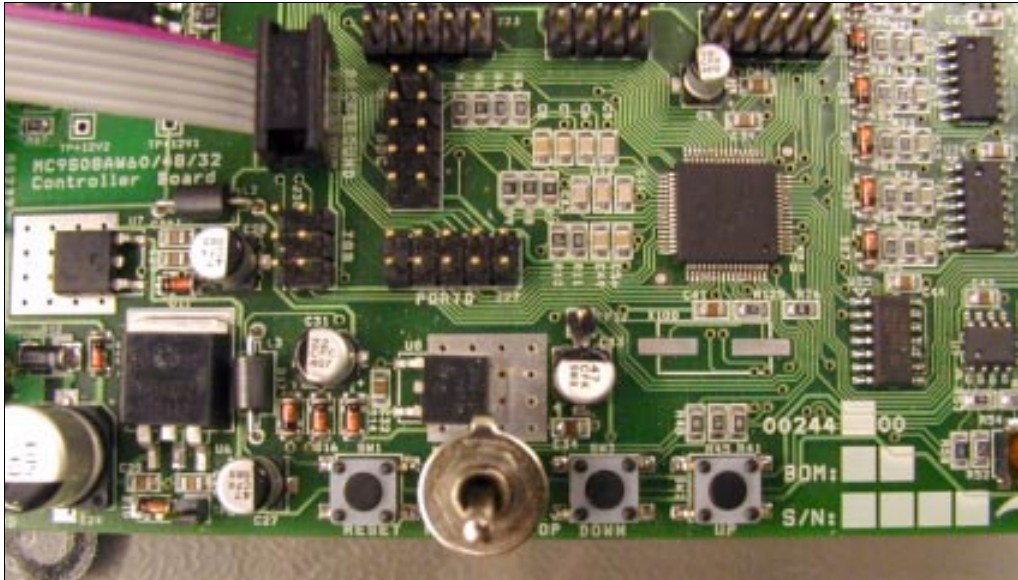
If the application is set to the speed operational mode, the user can set the required speed manually or by the PC. The DC bus current (torque) is automatically set to the maximal value of 2 A. In this case, the BLDC drive maintains the required speed until the maximum DC-bus current (torque) is exceeded.

#### 7.2.2 Torque Operational Mode

Similarly, in torque operational mode the user can set the DC bus current (torque) of the BLDC motor. The required value can be set by the UP/DOWN buttons or remotely by the PC. The speed limit is set to a maximal speed of 1200 rpm. In this case, the BLDC drive maintains the required DC bus current (torque) until the maximum BLDC motor speed is achieved.

The operational mode can be changed only when the motor is stopped (START/STOP switch is in the STOP position). The default setting is speed-operational mode.

## 7.3 Manual/Remote Operation



**Figure 7-1. Manual Interface**

### 7.3.1 Manual Operation

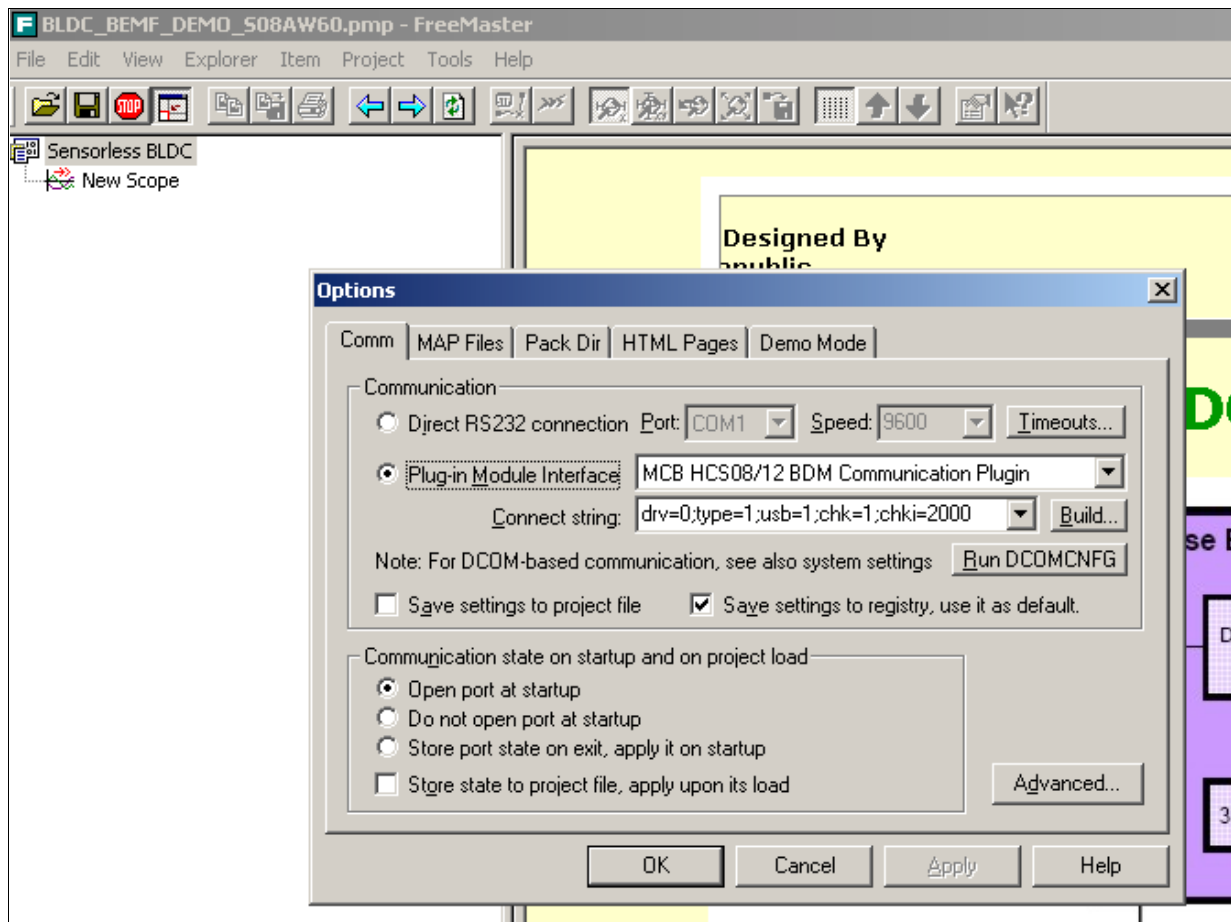
To run the BLDC motor, switch the START/STOP switch to the START position (toward the middle of the controller board). If the START/STOP switch is in the START position when applying the power supply, turn the switch to the STOP position before running the BLDC motor.

After the START/STOP switch is in the START position, the motor starts running with a minimum speed of 150 rpm or with a minimum DC-bus current of 0.2 A based on the operational mode. To increase/decrease the speed/DC bus current (torque), use the UP/DOWN buttons. The speed increment is 150 rpm and the maximal motor speed is 1200 rpm. The DC bus current increment is 0.1 A and the maximal DC bus current is 2 A. The default direction of rotation is clockwise rotation. The default operational mode is speed mode. The direction of rotation and operational mode cannot be changed manually. To change the direction of rotation and operational mode, use the FreeMASTER software.

### 7.3.2 Remote Operation

Remote operation can be provided by FreeMASTER software via the BDM interface. For correct FreeMASTER software, go to Project/Option/Comm and set communication via the plug-in BDM module (see [Figure 7-2](#)). If the BDM option is unavailable, install the latest version of FreeMASTER on your computer.





**Figure 7-2. FreeMaster Communication Settings**

The FreeMASTER software control page offers the same interface as the manual controls, adds some other features such as the change-of-rotation direction, operational mode settings, and displays some additional status information (Figure 7-3).

To start a remote operation, run the FreeMASTER software and open the project file located at **...\software\DEMO\Freemaster\BLDC\_BEMF\_DEMO\_S08AW60.pmp**.

The remote operation is active when the START/STOP switch is in the START position. The values, which can be changed, are displayed in red. The motor speed can be set to any value within the range of 150 to 1200 rpm. The DC bus current (torque) can be set to any value within the range of 0.1 to 2 A. The direction of rotation and operation modes can be changed only when the motor has stopped.

When you switch the START/STOP switch to STOP position, the motor stops and the remote control disables.

The FreeMASTER software also displays some variables in the scope (Figure 7-4).

The FreeMASTER software defines a speed stimulator for an automatic demonstration of the sensorless BLDC demo. To use the speed stimulator, switch the START/STOP switch to the START position. The speed changes according to a predefined ramp.

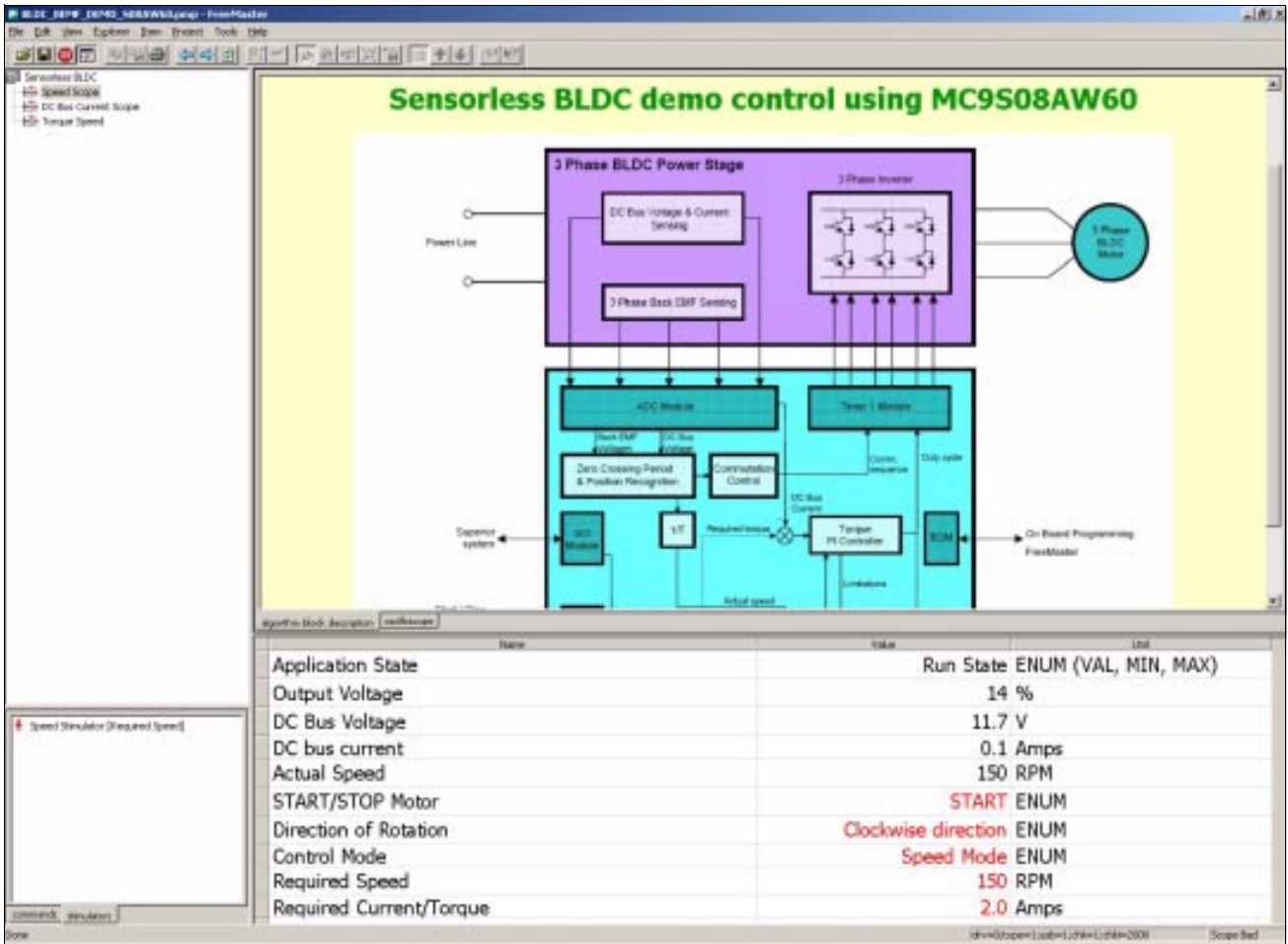


Figure 7-3. BLDC Demo Project in FreeMASTER

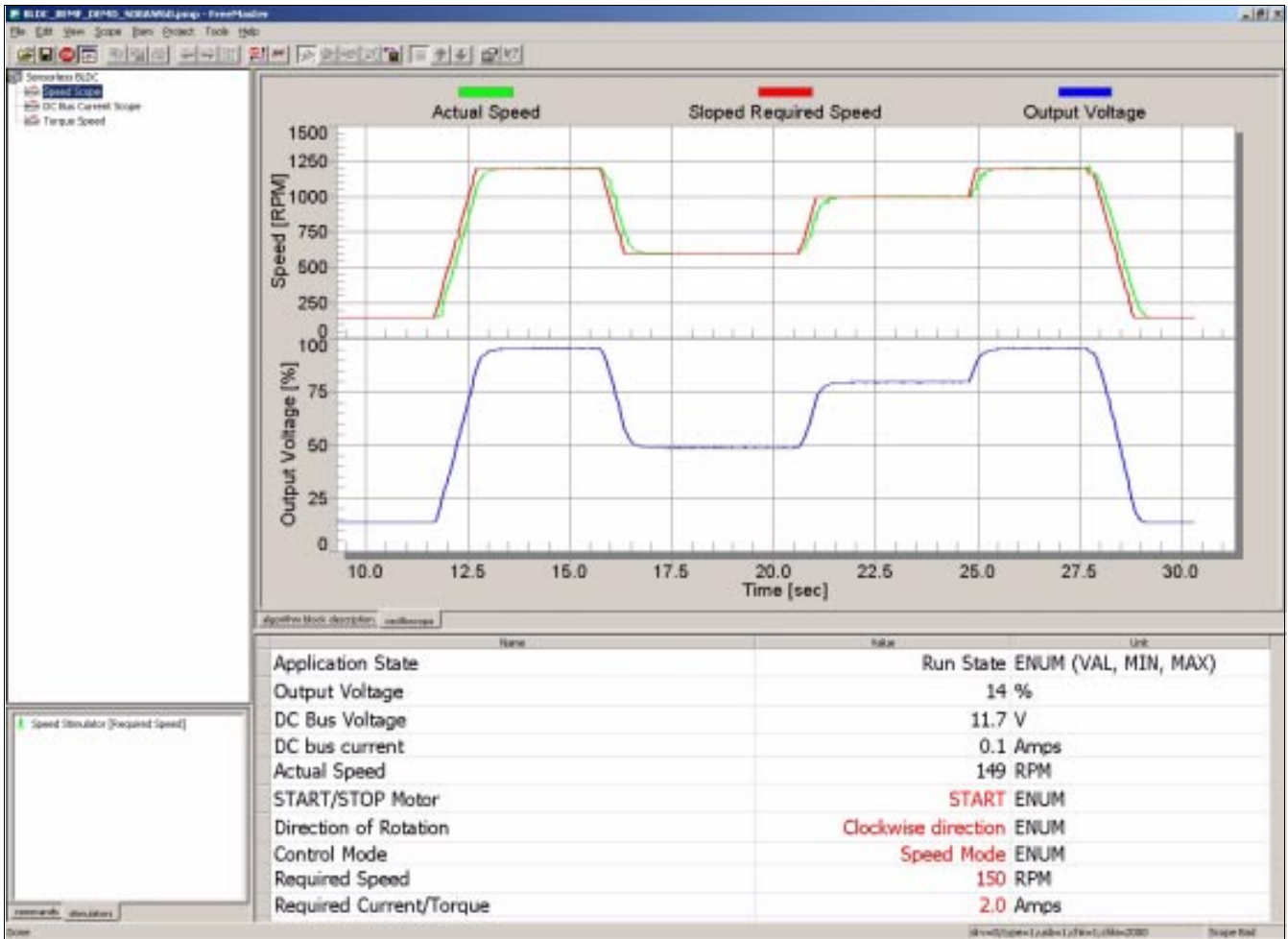


Figure 7-4. Scope Variables During Enabled Stimulator



## Appendix A

### References

1. *Design of Brushless Permanent-Magnet Motors*, J.R. Hendershot JR and T.J.E. Miller, Magna Physics Publishing and Clarendon Press, 1994.
2. *Sensorless Vector and Direct Torque Control*, P. Vas (1998), Oxford University Press, ISBN 0-19-856465-1, New York.
3. *Sensorless BLDC Motor Control Using the MC68HC908MR32*, DRM028, Freescale 2003
4. *3-phase BLDC Motor Control with Sensorless Back-EMF ADC Zero Crossing Detection using DSP 56F80x*, AN1913, Freescale 2005.
5. *3-Phase BLDC Motor Sensorless Control Using MC56F8013*, DRM70, Freescale 2005.
6. *Brushless DC Motor Control using the MC68HC708MC4*, AN1702, John Deatherage and Jeff Hunsinger, Freescale.
7. *FreeMaster for Embedded Applications User Manual*, Freescale 2004.
8. *MC9S08AW60: Technical Data Sheet for MC9S08AW60/48/32/16 MCUs*, Freescale 2006.
9. *MC9S08AW60 Controller Board Hardware User's Manual*, MC9S08AW60CB\_UM, Freescale 2005.
10. *Three-Phase Micro Power Stage Reference Manual*, MEMC3PMBLDCRM, Freescale 2004.
11. Freescale web page: <http://www.freescale.com>.



## Appendix B

### Glossary

**AC** — Alternating Current

**ADC** — See “analogue-to-digital converter”

**BDM** — Background Debug Mode interface allowing On-Chip Emulation and Programming

**brush** — A component transferring electrical power from non-rotational terminals, mounted on the stator, to the rotor

**BLDC** — Brushless dc motor

**commutation** — A process that creates a rotational field by the switching of power transistors (electronic replacement of brush and commutator)

**commutator** — A mechanical device alternating the DC current in a DC commutator motor and providing rotation of a DC commutator motor

**COP** — Computer Operating Properly timer

**DC** — Direct Current

**DT** — see “Dead Time (DT)”

**Dead Time (DT)** — the short time that must be inserted between the turning off of one transistor in the inverter half bridge and the turning on of the complementary transistor, due to the limited switching speed of the transistors

**duty cycle** — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

**GPIO** — General Purpose Input/Output

**Hall Sensors** — A position sensor giving six defined events (each 60 electrical degrees) per electrical revolution (for a 3-phase motor)

**HV** — High Voltage (115 V AC or 230 V AC)

**interrupt** — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine

**input/output (I/O)** — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal, and writes to an output to change the level on an external signal

**logic 1** — A voltage level approximately equal to the input power voltage ( $V_{DD}$ )

**logic 0** — A voltage level approximately equal to the ground voltage ( $V_{SS}$ )

**LV** — Low Voltage (12 V DC)

**MC9S08AWx** — A Freescale family of 8-bit MCUs suitable for motor control

**PI controller** — Proportional-Integral controller

**phase-locked loop (PLL)** — A clock generator circuit in which a voltage controlled oscillator produces an oscillation which is synchronised to a reference signal

**PM** — Permanent Magnet

**PMSM** — Permanent Magnet Synchronous Motor

**PWM** — Pulse Width Modulation

**Reset** — To force a device to a known condition

**RPM** — Revolutions per minute

**SCI** — See “serial communication interface module (SCI)”

**software** — Instructions and data that control the operation of a microcontroller

**software interrupt (SWI)** — An instruction that causes an interrupt and its associated vector fetch

**timer** — A module used to relate events in a system to a point in time





## **How to Reach Us:**

### **USA/Europe/Locations not listed:**

Freescale Semiconductor Literature Distribution  
P.O. Box 5405, Denver, Colorado 80217  
1-800-521-6274 or 480-768-2130

### **Japan:**

Freescale Semiconductor Japan Ltd.  
SPS, Technical Information Center  
3-20-1, Minami-Azabu  
Minato-ku  
Tokyo 106-8573, Japan  
81-3-3440-3569

### **Asia/Pacific:**

Freescale Semiconductor H.K. Ltd.  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T. Hong Kong  
852-26668334

### **Learn More:**

For more information about Freescale Semiconductor products, please visit <http://www.freescale.com>

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.  
© Freescale Semiconductor, Inc. 2006.