



Freescale Semiconductor, Inc.

# 56800 Hybrid Controller

*3-Phase PM  
Synchronous  
Motor Control  
with Quadrature  
Encoder  
Using 56F805*

*Designer Reference  
Manual*

*DRM029/D  
Rev. 0, 03/2003*

[MOTOROLA.COM/SEMICONDUCTORS](http://MOTOROLA.COM/SEMICONDUCTORS)



**Freescale Semiconductor, Inc.**

**Freescale Semiconductor, Inc.**

**For More Information On This Product,  
Go to: [www.freescale.com](http://www.freescale.com)**

# 3-Phase PM Synchronous Motor Control with Quadrature Encoder Using 56F805

Designer Reference Manual — Rev 0

by:

Pavel Grasblum, Ph.D.  
Motorola Czech System Laboratories  
Roznov pod Radhostem, Czech Republic

**Revision history**

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

<http://www.motorola.com/semiconductors>

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**

Date	Revision Level	Description	Page Number(s)
January, 2003	1	Initial release	N/A

## List of Sections

<b>Section 1. Introduction . . . . .</b>	<b>15</b>
<b>Section 2. Target Motor Theory and Control . . . . .</b>	<b>21</b>
<b>Section 3. System Description. . . . .</b>	<b>31</b>
<b>Section 4. Hardware Design. . . . .</b>	<b>35</b>
<b>Section 5. Software Design . . . . .</b>	<b>41</b>
<b>Section 6. System Setup . . . . .</b>	<b>53</b>
<b>Appendix A. References. . . . .</b>	<b>65</b>
<b>Appendix B. Glossary. . . . .</b>	<b>67</b>



## Table of Contents

### Section 1. Introduction

1.1	Contents . . . . .	15
1.2	Application Benefit . . . . .	15
1.3	Motorola DSP Advantages and Features . . . . .	16

### Section 2. Target Motor Theory and Control

2.1	Contents . . . . .	21
2.2	PM Synchronous Motor Theory . . . . .	21
2.3	Digital Control of a PM Synchronous Motor . . . . .	23
2.3.1	Control Technique . . . . .	25
2.3.2	Position Sensing . . . . .	27
2.3.3	Position Alignment . . . . .	28
2.3.4	Speed Control . . . . .	28

### Section 3. System Description

3.1	Contents . . . . .	31
3.2	System Outline . . . . .	31
3.3	Application Description . . . . .	33

### Section 4. Hardware Design

4.1	Contents . . . . .	35
4.2	System Configuration . . . . .	35
4.3	DSP56F805EVM Controller Board . . . . .	36
4.4	EVM Motor Board . . . . .	38

4.4.1 Electrical Characteristics .....39  
 4.4.2 Motor Characteristics .....39  
 4.5 Hardware Documentation .....40

**Section 5. Software Design**

5.1 Contents .....41  
 5.2 Software Description .....41  
 5.3 Data Flow .....41  
 5.3.1 Read Latest Position .....43  
 5.3.2 Period Measuring and Velocity Calculation .....44  
 5.3.3 Speed Controller .....44  
 5.3.4 3-phase Sinewave generation .....44  
 5.4 Software Implementation .....44  
 5.4.1 Initialization .....45  
 5.4.2 Interrupts .....47  
 5.4.3 Drive State Machine .....47  
 5.5 Implementation Notes .....48  
 5.5.1 Scaling of Quantities .....48  
 5.5.2 Motor Constant Calculation .....50

**Section 6. System Setup**

6.1 Contents .....53  
 6.2 Operational Modes .....53  
 6.3 Application Set-Up .....56  
 6.4 PM Synchronous Motor versus BLDC Motor .....58  
 6.5 DSP56F805EVM Set-Up .....58  
 6.6 Projects Files .....60  
 6.7 Application Build & Execute .....62

**Appendix A. References**





**Appendix B. Glossary**



## List of Figures

Figure	Title	Page
2-1	PM Synchronous Motor - Cross Section . . . . .	21
2-2	Torque Optimal Control of PM Synchronous Motor . . . . .	23
2-3	Sinewave Voltage Output Applied onto a PM Synchronous Motor . . . . .	24
2-4	3-Phase Power Stage . . . . .	25
2-5	Stator Flux Generation . . . . .	26
2-6	PM Synchronous Motor Phasor Diagram . . . . .	27
2-7	Alignment of Rotor Position . . . . .	28
2-8	Speed Controller . . . . .	29
3-1	System Concept . . . . .	33
4-1	Low-Voltage Evaluation Motor HW System Configuration . . . . .	35
4-2	Block Diagram of the DSP56F805EVM . . . . .	37
4-3	EVM Motor Board . . . . .	38
5-1	Main Data Flow . . . . .	43
5-2	State Diagram - General Overview . . . . .	45
5-3	Drive State Machine Transitions . . . . .	48
6-1	RUN/STOP Switch and UP/DOWN Buttons at DSP56F805EVM . . . . .	54
6-2	USER and PWM LEDs at DSP56F805EVM . . . . .	54
6-3	PC Master Software Control Window . . . . .	56
6-4	Set-up of the BLDC Motor Control Application using DSP56F805EVM . . . . .	57
6-5	DSP56F805EVM Jumper Reference . . . . .	59
6-6	Target Build Selection . . . . .	62
6-7	Execute Make Command . . . . .	63



**Designer Reference Manual — 3-Ph PM SMC with Quadrature Encoder**

---

---

**List of Tables**

<b>Table</b>	<b>Title</b>	<b>Page</b>
1-1	Memory Configuration . . . . .	17
3-1	Specifications of the 3-Phase BLDC Motor . . . . .	32
4-1	Electrical Characteristics . . . . .	39
4-2	Motor Characteristics . . . . .	39
6-1	Motor Application States . . . . .	55
6-2	DSP56F805EVM Jumper Settings . . . . .	59



## Section 1. Introduction

### 1.1 Contents

1.2	Application Benefit . . . . .	15
1.3	Motorola DSP Advantages and Features . . . . .	16

### 1.2 Application Benefit

This Reference Design describes the design of a 3-phase PM (Permanent Magnet) Synchronous motor drive based on Motorola’s DSP56F80x dedicated motor control device.

PM Synchronous motors are very popular in a wide range of applications. Compared with DC motors, PM Synchronous motors are without a commutator, so they are more reliable than DC motors. Also, in comparison to AC induction motors, PM Synchronous motors have advantages. PM Synchronous motors generate the rotor magnetic flux with rotor magnets so that PM Synchronous motors are highly efficient. Therefore, PM Synchronous motors are used in high-end white goods (refrigerators, washing machines, dishwashers, etc.), high-end pumps, fans and in other appliances, which require high reliability and efficiency.

The concept of this application is a speed-closed loop PM Synchronous drive using a Quadrature Encoder. It serves as an example of a PM Synchronous motor control system design using a Motorola DSP56F805.

This Reference Design includes the basic motor theory, system design concept, hardware implementation and software design, including the PC master software visualization tool.

### 1.3 Motorola DSP Advantages and Features

The Motorola DSP56F80x family is well suited for digital motor control, combining the DSP's (Digital Signal Processor) calculation capability with the MCU's (Micro Controller Unit) features on a single chip. These DSPs offer many dedicated peripherals like a Pulse Width Modulation (PWM) module, an Analog-to-Digital Converter (ADC), Timers, communication peripherals (SCI, SPI, CAN), on-board Flash and RAM. Generally, all family members are well-suited for various motor controls.

A typical member of the family, the DSP56F805, provides the following peripheral blocks:

- Two Pulse Width Modulator modules (PWMA & PWMB), each with six PWM outputs, three Current Sense inputs, and four Fault inputs, fault tolerant design with deadtime insertion, supporting both center- and edge-aligned modes
- 12-bit Analog-to-Digital Converters (ADCs), supporting two simultaneous conversions with dual 4-pin multiplexed inputs; the ADC can be synchronized to the PWM modules
- Two Quadrature Decoders (Quad Dec0 & Quad Dec1), each with four inputs, or two additional Quad Timers A & B
- Two dedicated General Purpose Quad Timers totaling six pins: Timer C with two pins and Timer D with four pins
- A CAN 2.0 A/B Module with a 2-pin port used to transmit and receive
- Two Serial Communication Interfaces (SCI0 & SCI1), each with two pins, or four additional GPIO lines
- A Serial Peripheral Interface (SPI), with a configurable 4-pin port, or four additional GPIO lines
- A Computer Operating Properly (COP) timer
- Two dedicated external interrupt pins
- Fourteen dedicated General Purpose I/O (GPIO) pins, 18 multiplexed GPIO pins
- An external reset pin for hardware reset



- JTAG/On-Chip Emulation (OnCE)
- a software-programmable, Phase Lock Loop-based frequency synthesizer for the DSP core clock

**Table 1-1. Memory Configuration**

	DSP56F801	DSP56F803	DSP56F805	DSP56F807
Program Flash	8188 x 16-bit	32252 x 16-bit	32252 x 16-bit	61436 x 16-bit
Data Flash	2K x 16-bit	4K x 16-bit	4K x 16-bit	8K x 16-bit
Program RAM	1K x 16-bit	512 x 16-bit	512 x 16-bit	2K x 16-bit
Data RAM	1K x 16-bit	2K x 16-bit	2K x 16-bit	4K x 16-bit
Boot Flash	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit

Aside from to the fast Analog-to-Digital converter and the 16-bit Quad Timers, the most interesting peripheral from the PM Synchronous motor control point of view is the Pulse Width Modulation (PWM) module. The PWM module offers a high degree of freedom in its configuration, permitting efficient control of the PM Synchronous motor.

The PWM has the following features:

- Three complementary PWM signal pairs, or six independent PWM signals
- Features of complementary channel operation
- Deadtime insertion
- Separate top and bottom pulse width correction via current status inputs or software
- Separate top and bottom polarity control
- Edge-aligned or center-aligned PWM signals
- 15 bits of resolution
- Half-cycle reload capability
- Integral reload rates from 1 to 16
- Individual software-controlled PWM outputs

- Mask and Swap of PWM outputs
- Programmable fault protection
- Polarity control
- 20mA current sink capability on the PWM pins
- Write-protectable registers

The PM Synchronous motor control utilizes the PWM block set in the complementary PWM mode, permitting generation of control signals for all switches of the power stage with inserted deadtime. The PWM block generates three sinewave outputs mutually shifted by 120 degrees.

The Quad Timer is an extremely flexible module, providing all required services related to time events. It has the following features:

- Each timer module consists of four 16-bit counters/timers
- Count up/down
- Counters are cascadable
- Programmable count modulo
- Max count rate equals peripheral clock/2 when counting external events
- Max count rate equals peripheral clock when using internal clocks
- Count once or repeatedly
- Counters are preloadable
- Counters can share available input pins
- Each counter has a separate prescaler
- Each counter has capture and compare capability

The PM Synchronous motor application utilizes one channel of the Quad Timer module counting in quadrature mode. It enables sensing of the rotor position using the Quadrature Encoder. The second channel of the Quad Timer module is set to generate a time base for a speed controller.

The Quadrature Decoder is a module providing decoding of position signals from a Quadrature Encoder mounted on a motor shaft. It has the following features:

- Logic to decode quadrature signals
- Configurable digital filter for inputs
- 32-bit position counter
- 16-bit position difference counter
- Maximum count frequency equals the peripheral clock rate
- Position counter can be initialized by software or external events
- Preloadable 16-bit revolution counter
- Inputs can be connected to a general purpose timer to aid low speed velocity.

The PM Synchronous motor application utilizes the Quadrature Decoder connected to Quad Timer module A. It uses the Decoder's digital input filter, to filter the Encoder's signals, but does not make use of its decoding functions, so the decoder's digital processing capabilities are free to be used by another application.



## Section 2. Target Motor Theory and Control

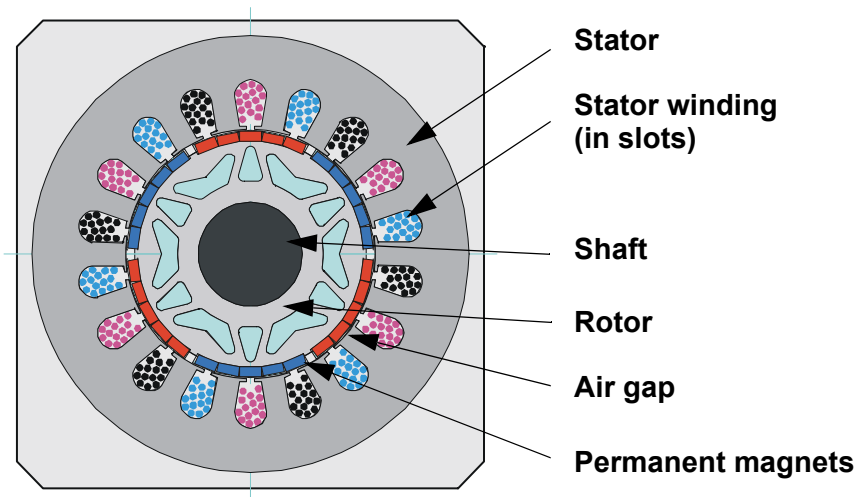
### 2.1 Contents

2.2 PM Synchronous Motor Theory .....21

2.3 Digital Control of a PM Synchronous Motor .....23

### 2.2 PM Synchronous Motor Theory

The PM Synchronous motor is a rotating electric machine where the stator is a classic three phase stator like that of an induction motor and the rotor has surface-mounted permanent magnets (see **Figure 2-1**).



**Figure 2-1. PM Synchronous Motor - Cross Section**

In this respect, the PM Synchronous motor is equivalent to an induction motor where the air gap magnetic field is produced by a permanent magnet. It means that the rotor magnetic field is constant. PM

Synchronous motors provide a set of advantages for designing modern motion control systems. The use of a permanent magnet to generate a substantial air gap magnetic flux makes it possible to design highly efficient PM motors.

The PM Synchronous motor is described by the following equations:

$$u_S = r_S \cdot i_S + \frac{d\psi_S}{dt} \tag{EQ 2-1.}$$

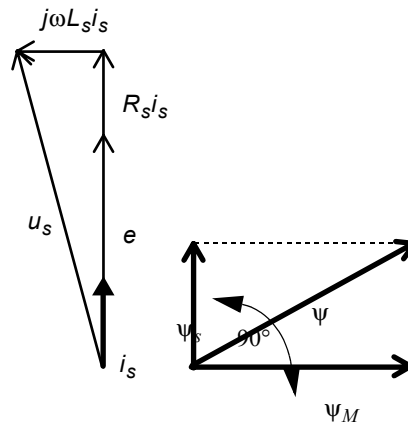
$$\psi_S = L_S \cdot i_S + \psi_M \tag{EQ 2-2.}$$

$$T_e = |i_S| \cdot |\psi_S| \cdot \sin(\angle i_S, \angle \psi_S) = |i_S| \cdot |\psi_M| \cdot \sin(\angle i_S, \angle \psi_M) \tag{EQ 2-3.}$$

where

- $u_S$  is the space phasor of stator voltage
- $i_S$  is the space phasor of stator current
- $r_S$  is the stator phase resistance
- $\Psi_S$  is the space phasor of stator magnetic flux
- $\Psi_M$  is the space phasor of rotor magnetic flux evoked by the permanent magnet
- $T_e$  is the electrical torque

As can be seen from equation (EQ 2-3), optimal torque is generated when the stator current vector is placed  $\pm 90^\circ$  relative to the rotor permanent magnet flux space vector. This situation is shown in [Figure 2-2](#).



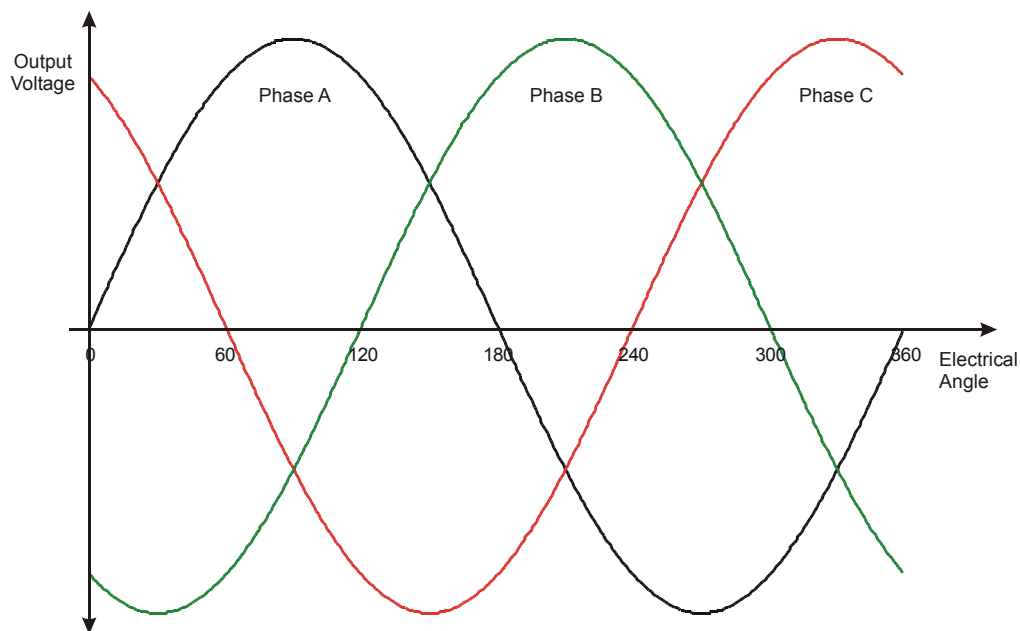
**Figure 2-2. Torque Optimal Control of PM Synchronous Motor**

where

- $R_S$  is the stator resistance
- $L_S$  is the stator inductance
- $e$  is the Back-EMF voltage
- $\Psi$  resultant magnetic flux

### 2.3 Digital Control of a PM Synchronous Motor

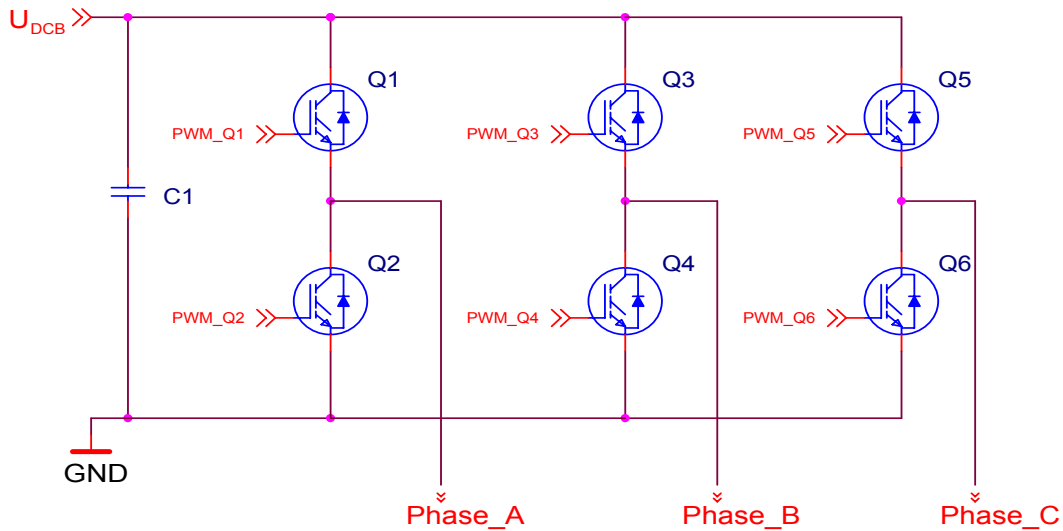
A PM Synchronous motor is driven by sinewave voltage coupled with the given rotor position. The generated stator flux together with the rotor flux, which is generated by a rotor magnet, defines the torque, and thus speed, of the motor. The sine wave voltage output have to be applied to the 3-phase winding system in a way that angle between the stator flux and the rotor flux is kept close to  $90^\circ$  to get the maximum generated torque. To meet this criterion, the motor requires electronic control for proper operation.



**Figure 2-3. Sinewave Voltage Output Applied onto a PM Synchronous Motor**

For a common 3-phase PM Synchronous motor, a standard 3-phase power stage is used. The same power stage is used for AC induction and BLDC motors. Such a power stage for 3-phase PM Synchronous motors is illustrated in [Figure 2-4](#). The power stage utilizes six power transistors with independent switching. The power transistors are switched in the complementary mode. The sinewave output is generated using a PWM technique.





**Figure 2-4. 3-Phase Power Stage**

### 2.3.1 Control Technique

The presented control algorithm demonstrates the principle of PM Synchronous motor control and use of the DSP56F80x peripherals. It means that this algorithm can be used as starting point for more sophisticated algorithms.

As is well known, the PM Synchronous (Permanent Magnet) motor is very similar to a Brushless DC motor. The PM Synchronous motors differ in three respects:

- sinusoidal distribution of magnet flux in the air gap
- sinusoidal current waveforms
- sinusoidal distribution of stator conductors.

Using a six-step control technique we get six flux vectors. This technique is commonly used for BLDC motors (see [10](#), [11](#)). In the case of sinusoidal voltage output, we are able to generate the stator flux in any position. The resultant flux is calculated as the sum of flux vectors of Phase A, Phase B and Phase C (see [Figure 2-5](#)). If the phase voltage changes sinusoidally over time, a fluent rotational flux field is generated.

As a result, a PM Synchronous motor runs smoother and quieter than a BLDC motor.

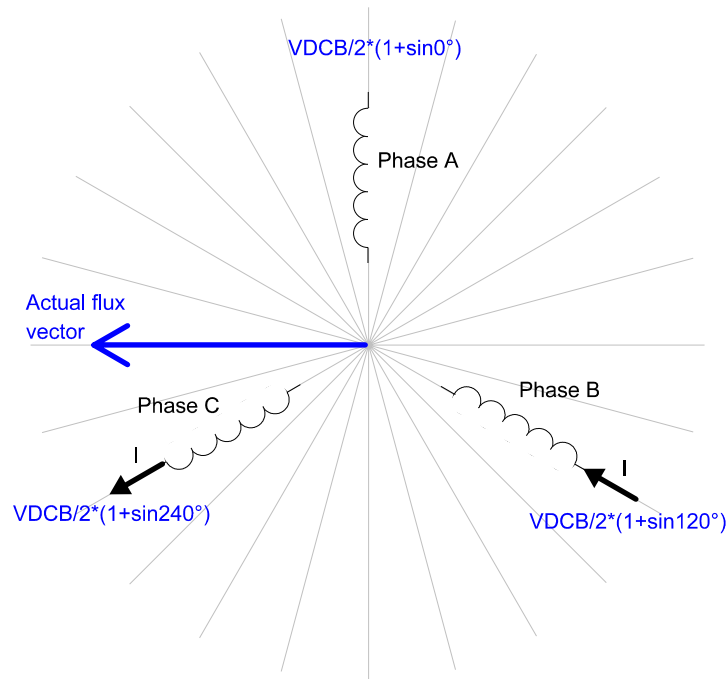


Figure 2-5. Stator Flux Generation

The motor runs with optimal torque generation when the angle between the stator and rotor flux is 90 electric degrees (see [Figure 2-6 a](#)).

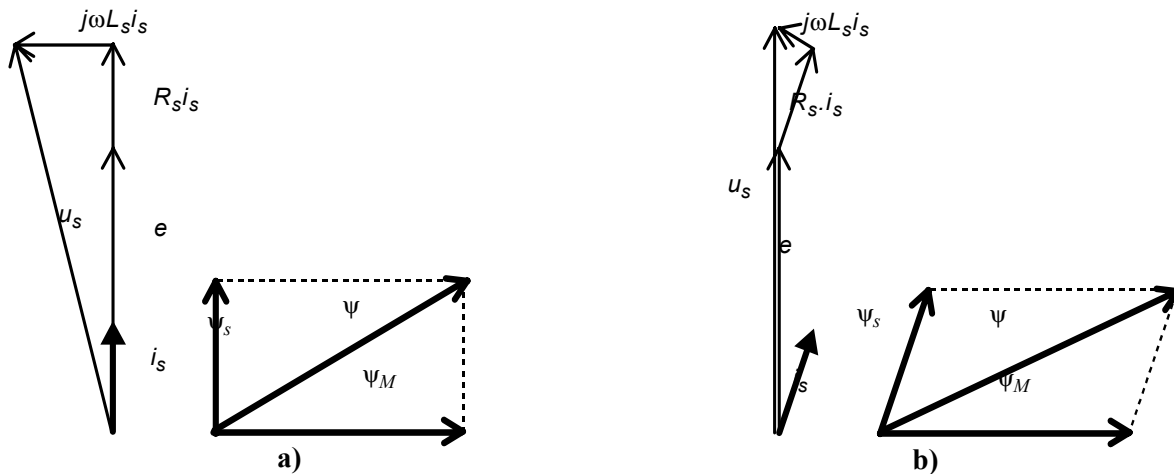
To ensure the angle between rotor and stator flux equals to 90 electric degrees it is necessary to know the position of the rotor and stator flux.

The position of the rotor flux is bound to the rotor position. Thus, by measuring rotor position we can get the exact position of the rotor flux.

The position of stator flux is bound to the vector of the stator current. To know the exact position of the stator flux, it requires measurement of the phase currents and the calculation of the stator current vector. Since the presented application does not measure current, there is no way to obtain the position of current vector.

To avoid the current measurement there is one of possible solution which aligns the vector of the applied voltage to be orthogonal to the rotor position (see **Figure 2-6 b**). As can be seen, the angle between the stator and rotor flux is not exactly 90 electric degrees because the voltage drop on the stator inductance is not compensated. The real angle is lower than 90 electric degrees and depends on the load.

For low-cost applications, such a solution is fully sufficient. For high-end applications, the current measurement and the stator flux need to be evaluated. (see **12**)



**Figure 2-6. PM Synchronous Motor Phasor Diagram**

### 2.3.2 Position Sensing

The rotor position is obtained from a Quadrature Encoder mounted on the rotor shaft. The encoder transfers the rotational movement into signal pulses corresponding to the position. The Quadrature Encoder output signals are connected to the on-chip Quadrature Decoder input. The signals go through a digital filter to the Quad Timer. The Quad Timer is set to count in quadrature mode. During alignment, the Quad Timer is preset to a value which represents a shift of the rotor position by 90 electrical degrees. Thus, the applied voltage is aligned with Back-EMF according to **Figure 2-6 b**.

### 2.3.3 Position Alignment

Since the Quadrature Encoder doesn't give the absolute position, we need to know exactly the rotor position before the motor is started. One possible, and very easy implementable, method is the rotor alignment to a predefined position. The motor is powered by a selected static voltage pattern (usually the zero position in the sinewave table) and the rotor aligns to the predefined position. The alignment is done only once during first motor start. **Figure 2-7** shows the position of the aligned rotor. After alignment the position counter is set to 90 electric degrees from alignment position in order to preset the angle between stator and rotor flux.

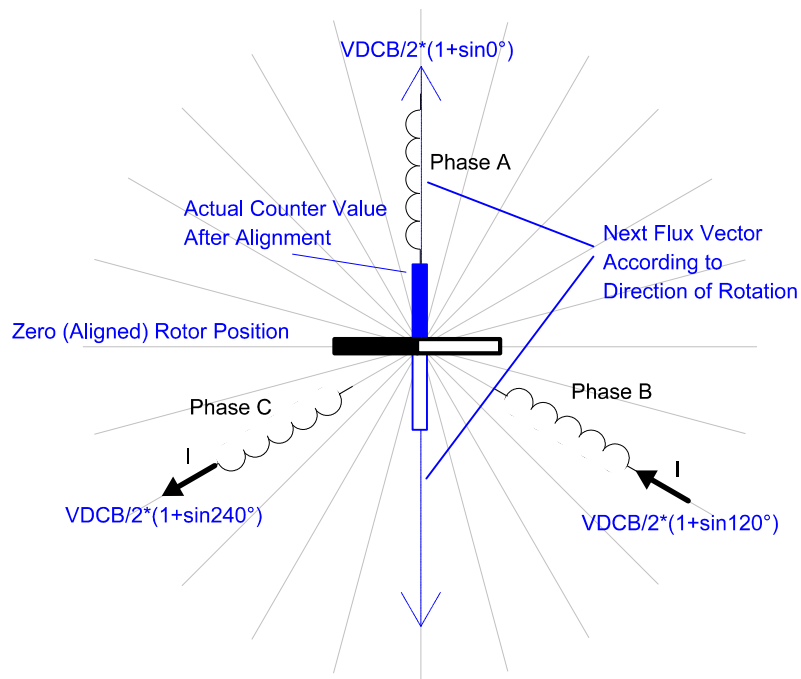
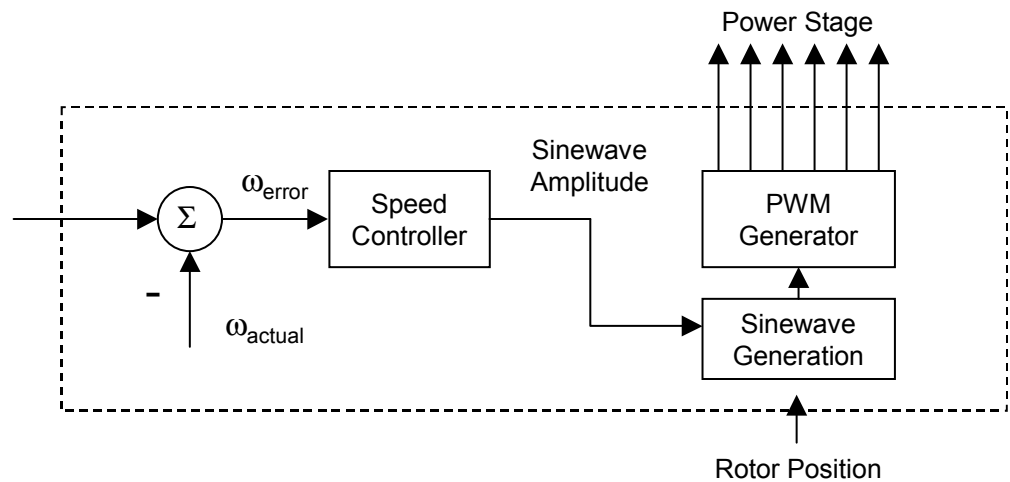


Figure 2-7. Alignment of Rotor Position

### 2.3.4 Speed Control

The correct shift between the rotor and stator flux ensures that the PM Synchronous motor generates a torque. The torque amplitude depends on the amplitude of the applied voltage. It means that the motor speed

is also controlled by the amplitude of the applied voltage. The amplitude of the applied voltage is changed by the PWM technique. The required speed is controlled by a speed controller. The speed controller is implemented as a conventional PI controller. The PI controller compares the actual and required speeds. The difference between the actual and required speed is input to the PI controller and based on this difference. The PI controller calculates the duty cycle which corresponds to the voltage amplitude required to keep the required speed.



**Figure 2-8. Speed Controller**

The speed controller calculates a Proportional-Integral (PI) algorithm according to the equations below:

$$u(t) = K_c \left[ e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau \right] \quad \text{(EQ 2-4.)}$$

After transformation to a discrete time domain using an integral approximation by a Backward Euler method, we get the following equations for the numerical PI controller calculation:

$$e(k) = w(k) - m(k) \tag{EQ 2-5.}$$

$$u(k) = u_p(k) + u_I(k) \tag{EQ 2-6.}$$

$$u_p(k) = K_c \cdot e(k) \tag{EQ 2-7.}$$

$$u_I(k) = u_I(k-1) + K_c \frac{T}{T_I} \cdot e(k) \tag{EQ 2-8.}$$

where:

- $e(t), e(\tau)$  is the input error in time  $t, \tau$
- $e(k)$  is the input error in step  $k$
- $w(k)$  is the desired value in step  $k$
- $m(k)$  is the measured value in step  $k$
- $u(k)$  is the controller output in step  $k$
- $u_p(k)$  is the proportional output portion in step  $k$
- $u_I(k)$  is the integral output portion in step  $k$
- $u_I(k-1)$  is the integral output portion in step  $k-1$
- $T_I$  is the integral time constant
- $T$  is the sampling time
- $K_c$  is the controller gain
- $t, \tau$  is the time
- $p$  is the Laplace variable

## Section 3. System Description

### 3.1 Contents

3.2	System Outline . . . . .	31
3.3	Application Description . . . . .	33

### 3.2 System Outline

The system is designed to drive a 3-phase PM Synchronous motor. The application meets the following performance specification:

- Voltage control of PM Synchronous motor using Quadrature Encoder
- Targeted for DSP56F805EVM
- Running on a 3-phase EVM Motor Board
- Control technique incorporates:
  - Voltage PM Synchronous motor control with speed-closed loop
  - Both directions of rotation
  - Motoring mode
  - Start from any motor position without rotor alignment
  - Minimum speed 50 RPM
  - Maximum speed 1000 RPM (limited by power supply)
- Manual interface (Start/Stop switch, Up/Down push button control, Led indication)
- PC master software control interface (motor start/stop, speed set-up)

**System Description**

- PC master software monitor
  - PC master software graphical Control Page (required speed, actual motor speed, start/stop status, DC-Bus voltage level, system status)
  - PC master software Speed Scope (observes actual & desired speeds)
- DC-Bus under-voltage fault protection

The introduced PM Synchronous drive is designed to power a low-voltage PM Synchronous motor equipped with a Quadrature Encoder, which is supplied with the EVM Motor Board. The motor has the following specifications:

**Table 3-1. Specifications of the 3-Phase BLDC Motor**

Motor Specification:	eMotor Type:	3-Phase BLDC Motor 4 Poles
	Speed Range:	< 5000 RPM
	Line Voltage:	60V
	Phase Current:	2A
Position Sensor Specification:	Sensor 1 Type:	3-Phase Hall Sensors
	Sensor 2 Type:	Quadrature Encoder 500 Pulses Per Revolution

**NOTE:** *The application SW is targeted for PM Synchronous motor with sine-wave Back-EMF shape. In this particular demo application the BLDC motor is used instead. This is due to the availability of the BLDC motor supplied as ECMTREVAL. Although the Back-EMF shape of this motor is not ideally sine-wave, it can be controlled by the application SW. The drive parameters will be even better when PMSM motor with exactly sine-wave Back-EMF shape is used.*



### 3.3 Application Description

A standard system concept is chosen for the drive (see [Figure 3-1](#)). The system incorporates the following hardware boards:

- Power Supply 12V DC, 4Amps
- EVM Motor Board
- BLDC Motor IB23810 with Quadrature Encoder
- Evaluation Board DSP56F805

The DSP runs the main control algorithm. According to the user interface and feedback signals it generates 3-phase PWM output signals for the AC/BLDC inverter.

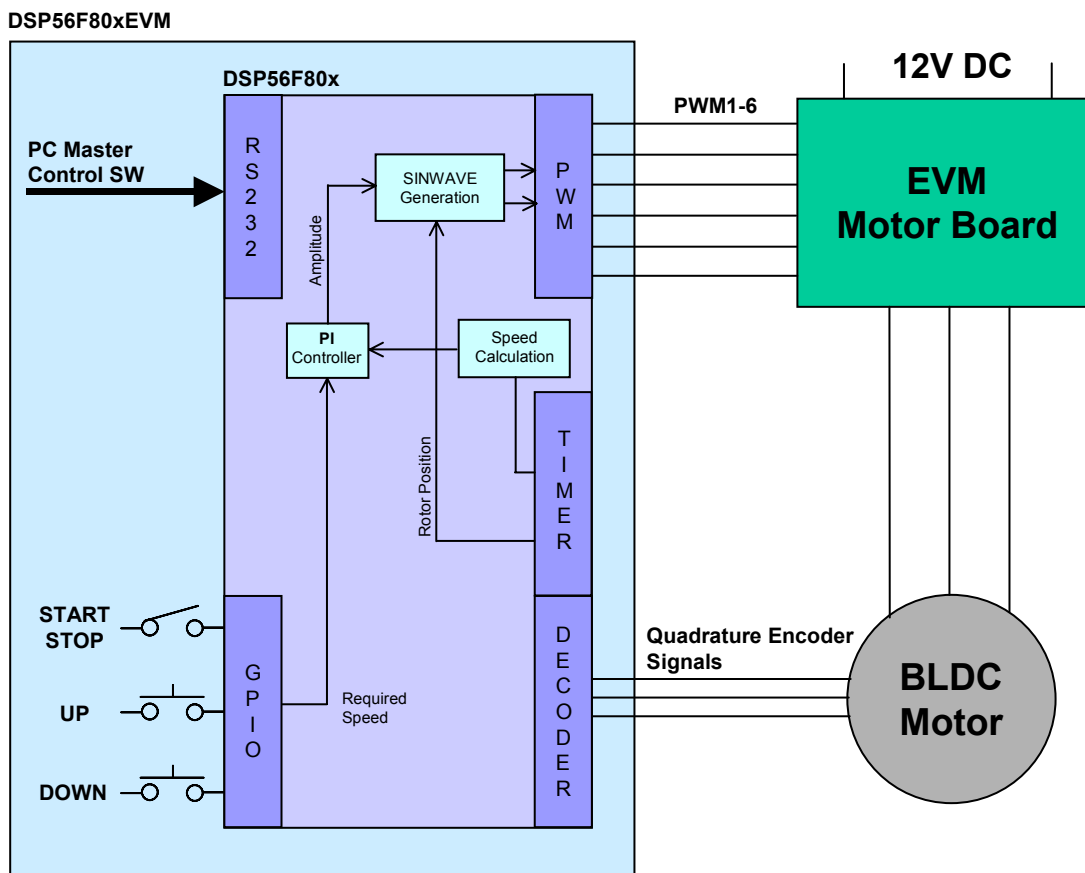


Figure 3-1. System Concept

The control process is as follows:

The state of the user interface is periodically scanned while the speed of the motor is measured on each new coming edge from the Quadrature Encoder (only one phase is used for speed measurement). According to the state of the control signals (Start/Stop switch, speed up/down buttons) the speed command is calculated. The comparison between the actual speed command and the measured speed generates a speed error. The speed error is brought to the speed PI controller that generates a new corrected amplitude of the sinewave output. The rotor position is also periodically scanned together with sinewave generation. The sinewave generation generates 3-phase sinewaves, shifted by 120 electrical degrees according to actual rotor position and the required amplitude. The output of sinewave generation defines directly the duty cycle of the PWM output signals for the power stage.

In the case of under-voltage, the PWM outputs are disabled and the fault state is displayed by an on-board LED.

## Section 4. Hardware Design

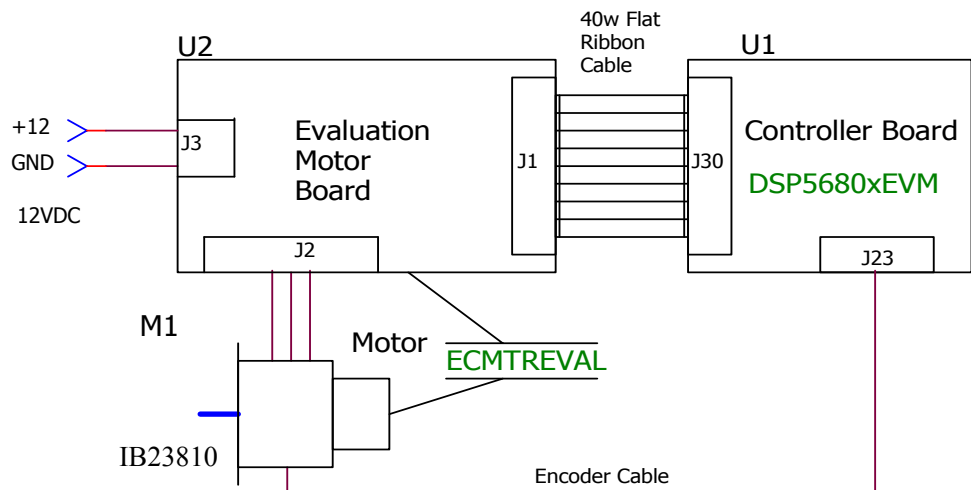
### 4.1 Contents

4.2	System Configuration .....	35
4.3	DSP56F805EVM Controller Board .....	36
4.4	EVM Motor Board .....	38
4.5	Hardware Documentation .....	40

### 4.2 System Configuration

The application is designed to drive the 3-phase PM Synchronous motor. It consists of the following modules (see **Figure 4-1**):

- DSP56F805EVM Controller Board
- Evaluation Motor Board
- 3-phase BLDC Motor



**Figure 4-1. Low-Voltage Evaluation Motor HW System Configuration**

**NOTE:** *The application SW is targeted for PM Synchronous motor with sine-wave Back-EMF shape. In this particular demo application the BLDC motor is used instead. This is due to the availability of the BLDC motor supplied as ECMTREVAL. Although the Back-EMF shape of this motor is not ideally sine-wave, it can be controlled by the application SW. The drive parameters will be even better when PMSM motor with exactly sine-wave Back-EMF shape is used.*

### 4.3 DSP56F805EVM Controller Board

The DSP56F805EVM is used to demonstrate the abilities of the DSP56F805 and to provide a hardware tool allowing the development of applications that use the DSP56F805.

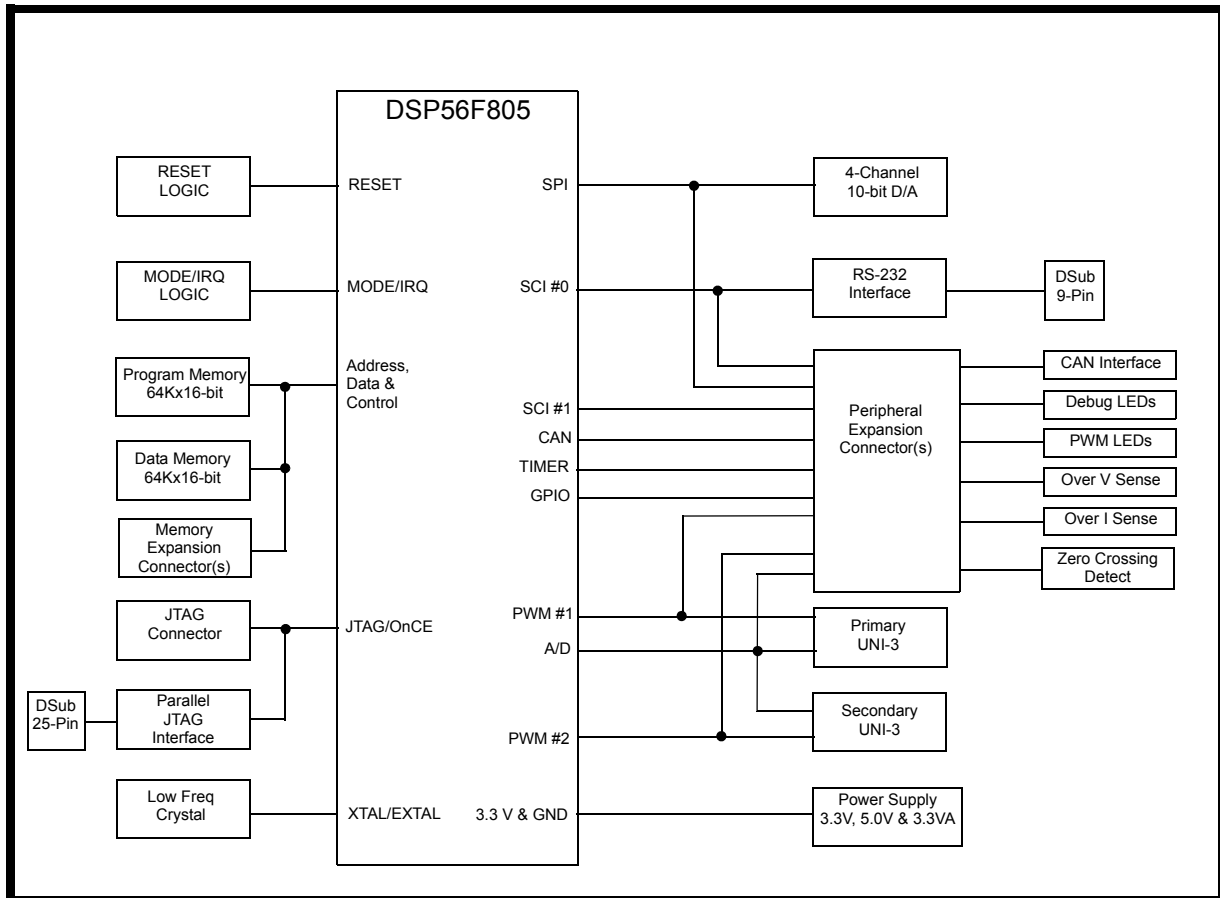
The DSP56F805EVM is an evaluation module board that includes a DSP56F805 part, peripheral expansion connectors, external memory and a CAN interface. The expansion connectors are for signal monitoring and user feature expandability.

The DSP56F805EVM is designed for the following purposes:

- Allowing new users to become familiar with the features of the 56800 architecture. The tools and examples provided with the DSP56F805EVM facilitate evaluation of the feature set and the benefits of the family.
- Serving as a platform for real-time software development. The tool suite enables the user to develop and simulate routines, download the software to on-chip or on-board RAM, run it, and debug it using a debugger via the JTAG/OnCE™ port. The breakpoint features of the OnCE port enable the user to easily specify complex break conditions and to execute user-developed software at full-speed, until the break conditions are satisfied. The ability to examine and modify all user accessible registers, memory and peripherals through the OnCE port greatly facilitates the task of the developer.
- Serving as a platform for hardware development. The hardware platform enables the user to connect external hardware peripherals. The on-board peripherals can be disabled, providing the user with the ability to reassign any and all of the DSP's

peripherals. The OnCE port's unobtrusive design means that all of the memory on the board and on the DSP chip are available to the user.

The DSP56F805EVM provides the features necessary for a user to write and debug software, demonstrate the functionality of that software and interface with the customer's application-specific device(s). The DSP56F805EVM is flexible enough to allow a user to fully exploit the DSP56F805's features to optimize the performance of their product, as shown in **Figure 4-2**.



**Figure 4-2. Block Diagram of the DSP56F805EVM**

### 4.4 EVM Motor Board

Motorola's evaluation motor board (EVM motor board) is a 12-volt, 4-amp power stage that is an integral part of Motorola's embedded motion control series of development tools. It is supplied in kit number ECMTREVAL, along with a small brushless dc motor, an encoder, an encoder cable, a 40-pin ribbon cable, and mounting hardware. In combination with one of the embedded motion control series control or evaluation boards, it provides a ready-made software development platform for small brushless dc motors. The motor is capable of being controlled with either Hall sensors, an optical encoder, or with sensorless techniques. **Figure 4-3** is an illustration of the board.

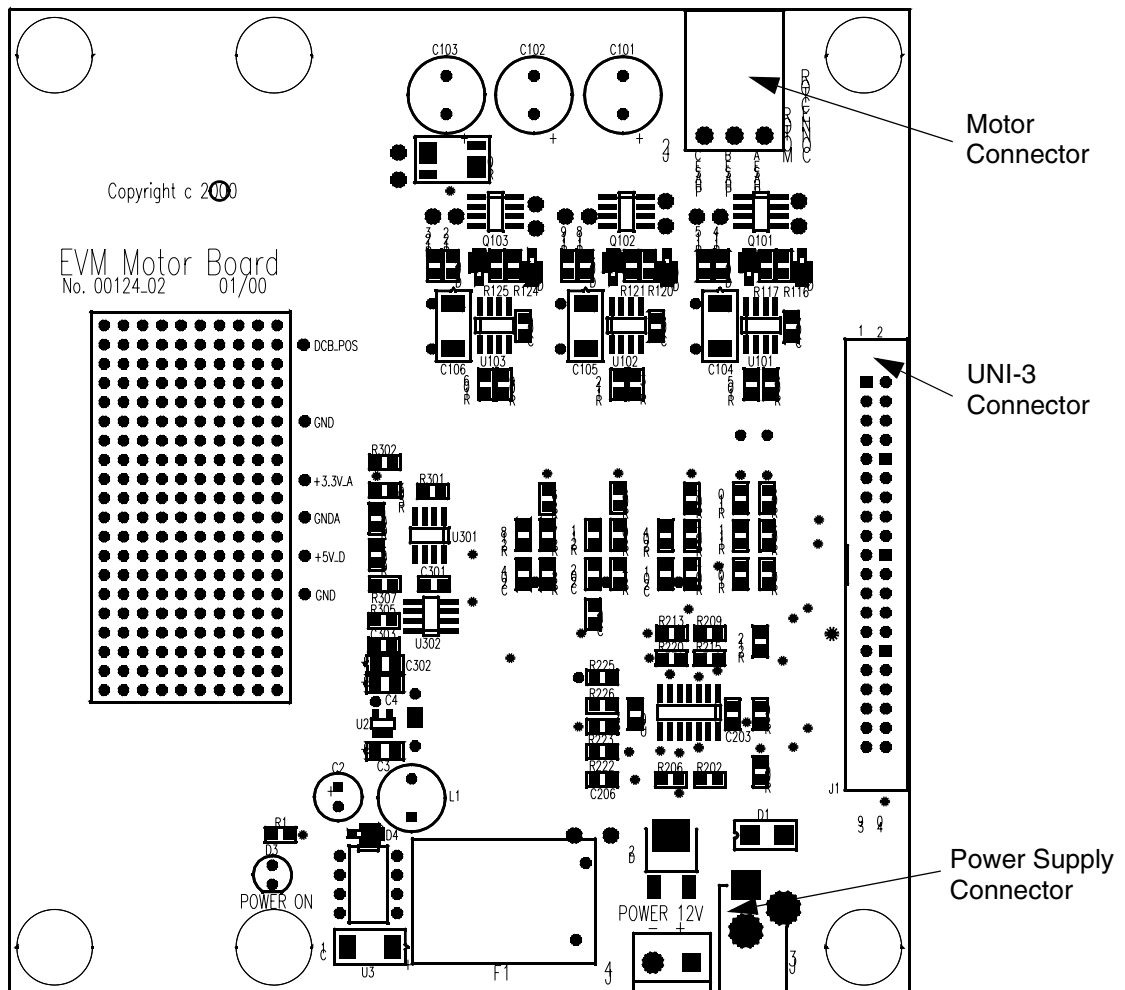


Figure 4-3. EVM Motor Board

#### 4.4.1 Electrical Characteristics

The electrical characteristics in [Table 4-1](#) apply to operation at 25°C and a 12-Vdc power supply voltage.

**Table 4-1. Electrical Characteristics**

Characteristic	Symbol	Min	Typ	Max	Units
Power Supply Voltage	V <sub>dc</sub>	10	12	16	V
Quiescent Current	I <sub>CC</sub>	—	50	—	mA
Min Logic 1 Input Voltage	V <sub>IH</sub>	2.4	—	—	V
Max 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>	—	412	—	mV/A
Bus Voltage Sense Voltage	V <sub>Bus</sub>	—	206	—	mV/V
Power MOSFET On Resistance	R <sub>DS(On)</sub>	—	32	40	MΩ
RMS Output Current	I <sub>M</sub>	—	—	6	A
Total Power Dissipation	P <sub>diss</sub>	—	—	5	W

#### 4.4.2 Motor Characteristics

The motor characteristics in [Table 4-2](#) apply to operation at 25°C.

**Table 4-2. Motor Characteristics**

Characteristic	Symbol	Min	Typ	Max	Units
Terminal Voltage	V <sub>t</sub>	—	—	60	V
Speed @ V <sub>t</sub>		—	5000	—	RPM
Torque Constant	K <sub>t</sub>	—	0.08	—	Nm/A
Voltage Constant	K <sub>e</sub>	—	8.4	—	V/kRPM
Winding Resistance	R <sub>t</sub>	—	2.8	—	Ω

**Table 4-2. Motor Characteristics**

Winding Inductance	L	—	8.6	—	mH
Continuous Current	$I_{cs}$	—	—	2	A
Peak Current	$I_{ps}$	—	—	5.9	A
Inertia	$J_m$	—	0.075	—	kgcm <sup>2</sup>
Thermal Resistance		—	—	3.6	°C/W

## 4.5 Hardware Documentation

.All the system parts are supplied and documented according the following references:

- M1 - IB23810 Motor
  - supplied in kit ECMTREVAL - Evaluation Motor Board Kit
- U2 EVM Motor Board:
  - supplied in kit with IB23810 Motor: ECMTREVAL - Evaluation Motor Board Kit
  - described in: *Evaluation Motor Board User's Manual*
- U1 CONTROLLER BOARD for DSP56F805:
  - supplied as: DSP56805EVM
  - described in: *DSP Evaluation Module Hardware User's Manual*

Detailed descriptions of individual boards can be found in comprehensive User's Manuals belonging to each board. The manuals are available on the Motorola web. The User's Manual incorporates the schematic of the board, description of individual function blocks and a bill of materials. An individual board can be ordered from Motorola as a standard product.



## Section 5. Software Design

### 5.1 Contents

5.2	Software Description . . . . .	41
5.3	Data Flow . . . . .	41
5.4	Software Implementation . . . . .	44
5.5	Implementation Notes . . . . .	48

### 5.2 Software Description

This section describes the design of the software blocks for the drive. The software will be described in terms of:

- Control Algorithm Data Flow
- Software Implementation

### 5.3 Data Flow

The control algorithm of a close loop PM Synchronous drive is described in **Figure 5-1**. The individual processes are described in the following sections.

The main data flow can be divided to four parts:

- Speed control
- Velocity calculation
- 3-phase sinewave generation
- DC-Bus voltage measurement

Speed control starts with the required speed `omega_required_mech`. This variable is set by user buttons or remotely by the PC within allowed limits. The variable `omega_required_mech` is copied to `omega_desired_mech` at a defined moment. This variable is used as a shadow variable to avoid change of the required speed from the PC at any time. The variable `omega_desired_mech` is input to the speed PI controller as a reference value.

`MeasuredTime` incorporates a time period of one phase of the Quadrature Encoder. The time period is used for speed calculation. Calculated speed, `omega_actual_mech`, is input to the speed PI controller as a secondary input. The PI controller output determines the amplitude of the generated sinusoidal output signals.

For the rotor position scanning the Timer A0, set as a quadrature counter, is used. The Timer A0 gives the actual rotor position shifted by 90 electrical degrees after initialization. This rotor position `RotorPosition` is input to the 3-phase sinewave modulation together with required amplitude `Amplitude`. The result of the sinewave modulation is written directly to the PWM block. The rotor position scanning with sinewave modulation is performed by an interrupt routine, which is called each PWM reload (16 kHz). The next task, which is provided by an interrupt routine, is the calculation of the spin direction. The result, `DirectionSpinning`, is used for the speed calculation.

The variable `u_dc_bus` contains the actual DC-Bus voltage. The value is used for an under-voltage detection.

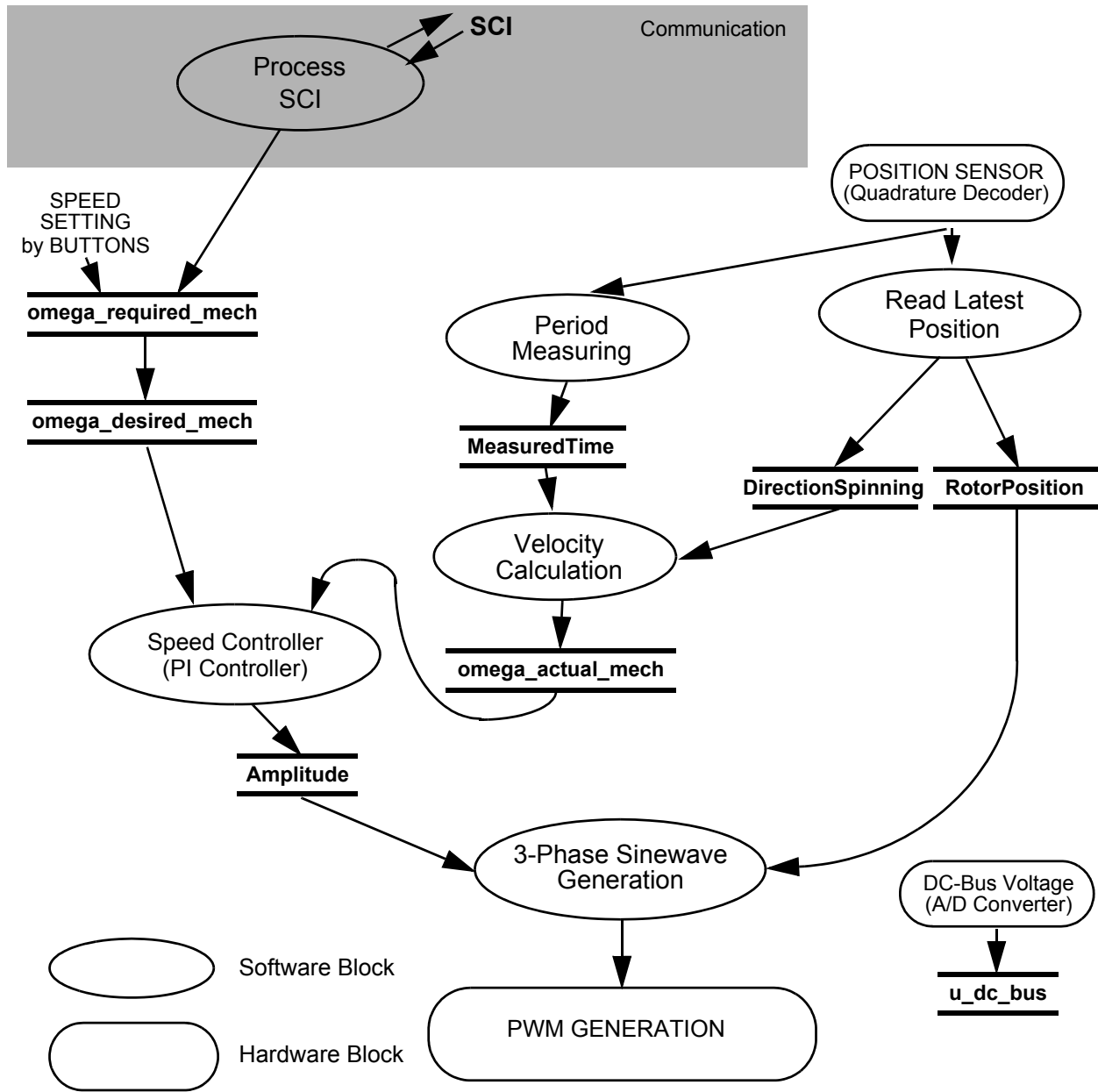


Figure 5-1. Main Data Flow

### 5.3.1 Read Latest Position

The process Read Latest Position is executed with each PWM Reload interrupt (16 kHz). The process reads the actual rotor position and calculates the direction of spinning.

### 5.3.2 Period Measuring and Velocity Calculation

The processes, Period Measuring and Velocity Calculation, read the time between the adjacent edges of one phase of the Quadrature Encoder and calculates the actual motor speed `omega_actual_mech`.

### 5.3.3 Speed Controller

This process compares the required and actual speed and calculates the duty cycle of the PWM output signals. For detailed information see [Section 2.3.4, Speed Control](#)

### 5.3.4 3-phase Sinewave generation

The process 3-phase Sinewave Generation calculates the PWM output from the actual rotor position and the required sinewave amplitude. The output is written to the PWM module. This process is performed within the PWM Reload interrupt.

## 5.4 Software Implementation

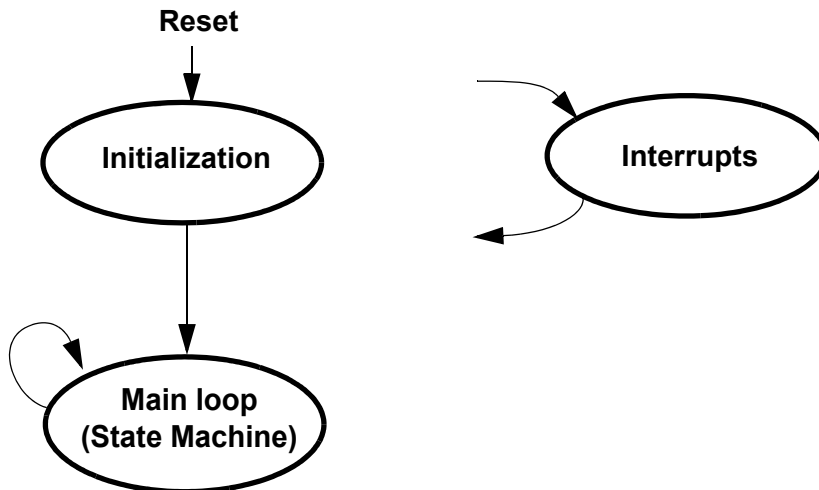
The general software diagram shows the Main routine entered from Reset and the interrupt states (see [Figure 5-2](#)).

The Main routine initializes both the DSP and the application, then enters into an infinite background loop. This loop contains an application State Machine.

The following interrupt service routines are utilized:

- PWM Reload ISR- services signals generated by the Quadrature Encoder and generates the 3-phase sinewave output
- Input Capture ISR (TimerA1) - services period measurement for speed calculation
- Timer ISR - services the speed controller and LED diode blinking
- Push Button Up ISR and Push Button Down ISR - service the Up and Down push buttons

- SCI ISR - services communication with the PC master software



**Figure 5-2. State Diagram - General Overview**

### 5.4.1 Initialization

The Main Routine provides initialization of the DSP:

- Disables Interrupts
- Initializes DSP PLL
- Disables COP and LVI
- Initializes the Timer for time base reference 1 ms
- Initializes the LED
- Initializes the PWM module:
  - Center-aligned complementary PWM mode, positive polarity
  - PWM modulus - defines PWM frequency
  - PWM deadtime - defines PWM deadtime
  - Disable faults
- Initializes Quadrature Decoder
  - Sets on-chip digital filter of the Quadrature Decoder inputs

- Connects Quadrature Decoder signals to QuadTimerA
- Initializes QuadTimerA - channel A0
  - set Count Mode to Quadrature Count
  - set Input Source to Input 0
  - set Input Polarity to Normal
  - set Secondary Input Source to Input 1
  - set Count Frequency to Repeatedly
  - set Count Length to Until Compare
  - set Count Direction to Down
  - disable Capture Mode
- Initializes QuadTimerA - channel A1
  - set Count Mode to Count
  - set Input Source to Bus Clock / 128
  - set Input Polarity to Normal
  - set Secondary Input Source to Input 1
  - set Count Frequency to Repeatedly
  - set Count Length to Past Compare
  - set Count Direction to Up
  - set Capture Mode = RisingEdges
  - associate Callback On Input Edge to CallbackOnNewEdge
  - associate CallbackOnOverflow to CallbackOnOverload
- Sets-up I/O ports (brake, switch, push buttons)
  - Brake, LED, switch on GPIO
  - Push buttons on interrupts IRQ0, IRQ1
- Initializes the Analog-to-Digital Converter
  - ADC set for sequential sampling, single conversion
  - Channel 0 = DC-Bus voltage

- Initializes control algorithm (speed controller, control algorithm parameters)
- Enables interrupts
- Starts ADC conversion

## 5.4.2 Interrupts

The interrupt handlers have the following functions:

- *PWM Reload* reads the actual rotor position, calculates the 3-phase sinewave output and spin direction and updates PWM Value Registers.
- *Input Capture Interrupt Handler (Timer A1)* reads the time between the two subsequent IC edges one phase of the Quadrature Encoder, which is used for speed calculation.
- *Timer Interrupt Handler* generates the time base 1ms. The routine, called within this time base, blinks the green LED diode, reads the result of the ADC conversion, calculates the speed and provides the speed controller.
- *Push Button Interrupt Handler* takes care of the push button service. The *UpButton Interrupt Handler* increments the desired speed, the *DownButton Interrupt Handler* decrements the desired speed.
- *PC and SCI Interrupt Handlers* provide SCI communication and service routines for the PC master software. These routines are fully independent of the motor control tasks.

## 5.4.3 Drive State Machine

The drive can be in any of the states shown in [Figure 5-3](#), which shows the transition conditions between the drive states. The user is able to recognize the current state, by a blinking green LED diode. In the case of the *init* and *stop* state, the green LED diode blinks at a frequency of 2 Hz. In the *fault* state, the green LED diode blinks at a frequency of

8 Hz. During the *running* state, the green LED diode is continuously turned on.

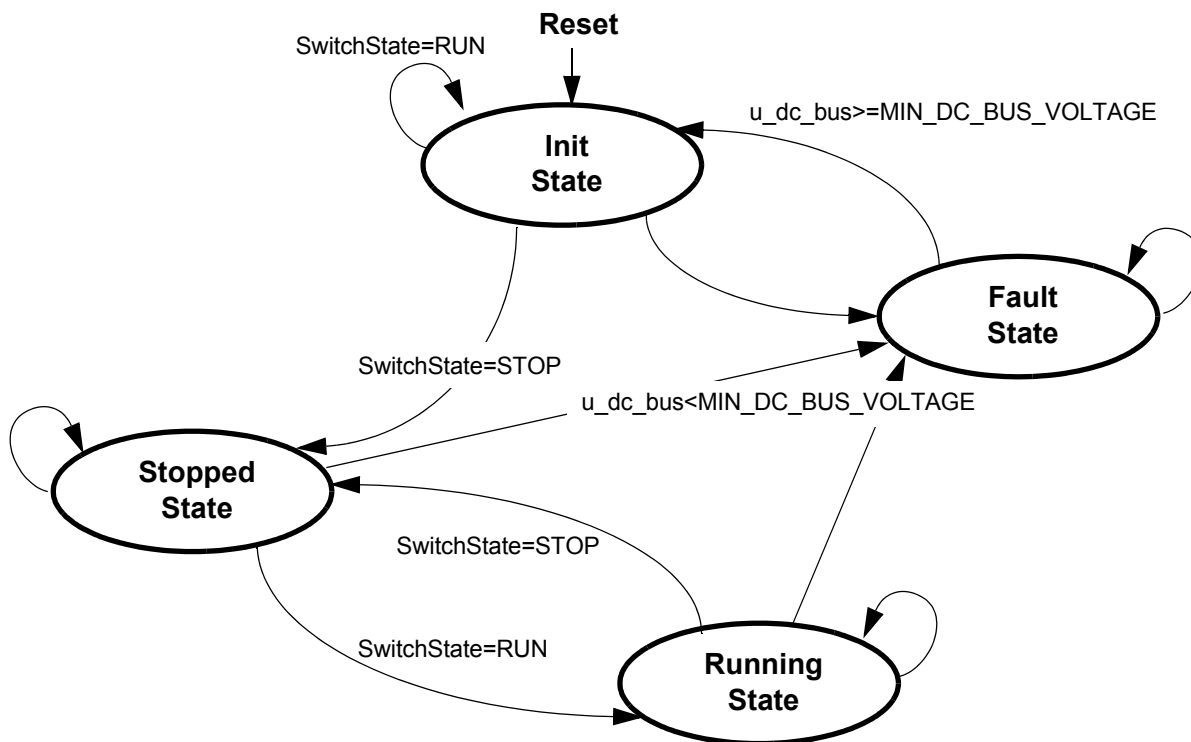


Figure 5-3. Drive State Machine Transitions

## 5.5 Implementation Notes

The following chapter describes calculation of application constants and quantity scaling.

### 5.5.1 Scaling of Quantities

The PM Synchronous motor control application uses a fractional representation for all real quantities except time. The N-bit signed



fractional format is represented using 1.[N-1] format (1 sign bit, N-1 fractional bits). Signed fractional numbers (SF) lie in the following range:

$$-1.0 \leq SF \leq +1.0 \cdot 2^{-[N-1]} \quad \text{(EQ 5-1.)}$$

For words and long-word signed fractions, the most negative number that can be represented is -1.0, whose internal representation is \$8000 and \$80000000, respectively. The most positive word is \$7FFF or  $1.0 - 2^{-15}$ , and the most positive long-word is \$7FFFFFFF or  $1.0 - 2^{-31}$ .

The following equation shows the relationship between real and fractional representations:

$$\text{Fractional Value} = \frac{\text{Real Value}}{\text{Real Quantity Range}} \quad \text{(EQ 5-2.)}$$

where:

*Fractional Value* is a fractional representation of the real value [Frac16]

*Real Value* is the real value of the quantity [V, A, RPM, etc.]

*Real Quantity Range* is the maximum range of the quantity, defined in the application [V, A, RPM, etc.]

### 5.5.1.1 DC-Bus Voltage Scaling

The DC-Bus voltage sense is defined by the following equation:

$$u\_dc\_bus = \frac{V_{DC\_BUS}}{V_{MAX}} \cdot 32767$$

Where:

$u\_dc\_bus$  = variable of DC-Bus voltage

$V_{DC\_BUS}$  = measured DC-Bus voltage

$V_{MAX}$  = max. measurable DC-Bus voltage.

**NOTE:**  $V_{MAX} = 16V$  for the EVM Motor Board

### 5.5.1.2 PI Controller Parameters

The P constant was chosen as 0.2 ( $26214 * 2^{-17}$ ) and the I constant was chosen as 0.3 ( $31457 * 2^{-20}$ ) or 0.12 ( $31457 * 2^{-18}$ ). To get better response to error speed, the I constant is changed according the actual speed. The I constant equals 0.3 from 50 to 200 RPM. Over 200 RPM the I constant equals 0.12. The controller parameters were experimentally tuned.

### 5.5.1.3 Velocity Calculation

The constant OMEGA\_ACTUAL\_MECH\_CONST is defined by the following equations:

position difference = 1/500 rev (given by each rising edge of one phase of Quadrature Encoder and two pole pairs motor)

max. period time = 0.008 s (chosen according to required min. speed)

$v_{\min} = 60 * (\text{position difference}) / (\text{max. period time}) = 15 \text{ RPM}$

$v_{\max} = 100 * v_{\min} = 1500 \text{ RPM}$  (chosen according to required max. speed)

$\text{OMEGA\_ACTUAL\_MECH\_CONST} = 32767 * v_{\min} / v_{\max} = 327$

## 5.5.2 Motor Constant Calculation

The PM Synchronous motor control application uses the constants, which depend on a motor type (number of pole pairs) and on a Quadrature Encoder type (number of pulses per revolution). The depended constants are:

- PULSES\_PER\_REVOLUTION
- VOLTAGE\_SHIFT
- SIN\_TABLE\_MULTIPLIER

The following paragraphs explain the constant calculations. The range for all constants is unsigned integer.

5.5.2.1 Constant *PULSES\_PER\_REVOLUTION*

The constant *PULSES\_PER\_REVOLUTION* defines the number of pulses of the Quadrature Encoder per electrical revolution. Since the Quadrature Encoder counts both rising and falling edges, the value is multiplied by four. The resultant value must be an integer.

$$PULSES\_PER\_REVOLUTION = \frac{4 \times \text{number of pulses per mech. revolution}}{\text{number of pole pairs}} - 1$$

(EQ 5-3.)

In the case of presented application the constant is equal:

$$PULSES\_PER\_REVOLUTION = \frac{4 \times 500}{2} - 1 = 999$$

**NOTE:** *In case that the constant is not an integer, it is necessary to set the constant *PULSES\_PER\_REVOLUTION* to a value which is equal to the number of pulses per mechanical revolution minus one. Then the actual rotor position has to be recalculated from the mechanical to electrical revolution.*

5.5.2.2 Constant *VOLTAGE\_SHIFT*

The constant *VOLTAGE\_SHIFT* defines the shift of applied voltage by 90 el. degree and is calculated as:

$$VOLTAGE\_SHIFT = \frac{(PULSES\_PER\_REVOLUTION + 1)}{4} \quad \text{(EQ 5-4.)}$$

Then for the presented application the constant is equal to:

$$VOLTAGE\_SHIFT = \frac{(999 + 1)}{4} = 250$$

### 5.5.2.3 Constant *SIN\_TABLE\_MULTIPLIER*

The constant *SIN\_TABLE\_MULTIPLIER* rescales the rotor position, which is defined in pulses per electrical revolution, to the sinewave table, which is scaled from -1 to 1  $(-\pi; \pi)$ . The constant is calculated as:

$$SIN\_TABLE\_MULTIPLIER = \frac{65535}{PULSES\_PER\_REVOLUTION} \times 256 \quad (\text{EQ 5-5.})$$

For the presented calculation can be calculated:

$$SIN\_TABLE\_MULTIPLIER = \frac{65535}{999} \times 256 = 16794$$

## Section 6. System Setup

### 6.1 Contents

6.2	Operational Modes . . . . .	53
6.3	Application Set-Up . . . . .	56
6.4	PM Synchronous Motor versus BLDC Motor . . . . .	58
6.5	DSP56F805EVM Set-Up . . . . .	58
6.6	Projects Files . . . . .	60
6.7	Application Build & Execute . . . . .	62

### 6.2 Operational Modes

This BLDC Motor Control Application with Hall Sensors can operate in two modes:

1. Manual Operating Mode  
 The drive is controlled by the RUN/STOP switch (S6). The motor speed is set by the UP (S2-IRQB) and DOWN (S1-IRQA) push buttons; see [Figure 6-1](#). If the application runs and motor spinning is disabled (i.e., the system is ready) the USER LED (LED3, shown in [Figure 6-2](#)) will blink. When motor spinning is enabled, the USER LED is *On*. Refer to [Table 6-1](#) for application states.

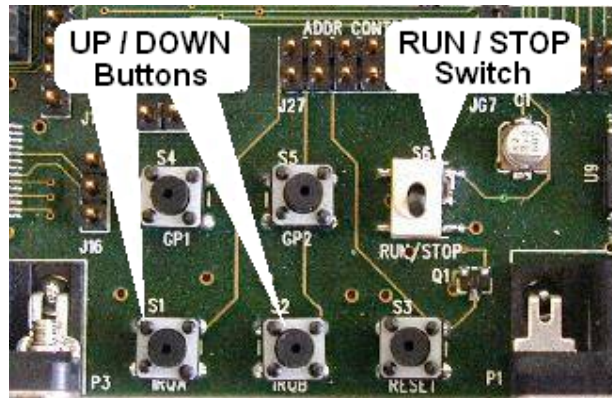


Figure 6-1. RUN/STOP Switch and UP/DOWN Buttons at DSP56F805EVM

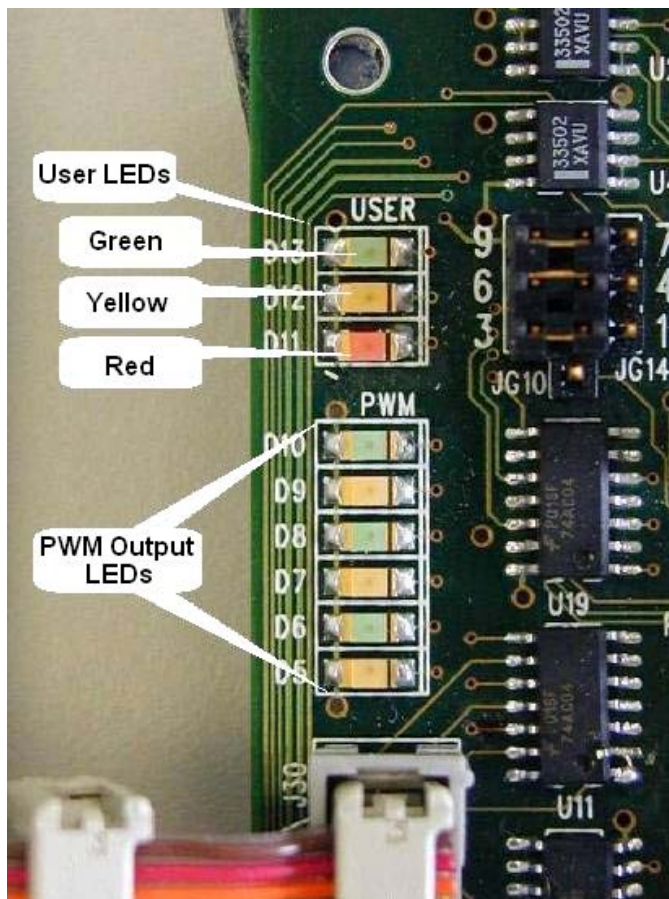


Figure 6-2. USER and PWM LEDs at DSP56F805EVM

**Table 6-1. Motor Application States**

Application State	Motor State	Green LED State
Stopped	Stopped	Blinking at a frequency of 2Hz
Running	Spinning	On
Fault	Stopped	Blinking at a frequency of 8Hz

2. PC master software (Remote) Operating Mode

The drive is controlled remotely from a PC through the SCI communication channel of the DSP device via an RS-232 physical interface. The drive is enabled by the RUN/STOP switch, which can be used to safely stop the application at any time. MC master software enables to set the required speed of the motor

PC master software displays the following information:

- Required Speed
- Actual Speed
- Amplitude
- DC-Bus Voltage
- RUN/STOP Switch Status
- Application Mode

Start the PC master software window's application, *BLDC\_synchro\_pm\_quad\_encoder.pmp*. **Figure 6-3** illustrates the PC master software control window after this project has been launched.

**NOTE:** *If the PC master software project (.pmp file) is unable to control the application, it is possible that the wrong load map (.elf file) has been selected. PC master software uses the load map to determine addresses for global variables being monitored. Once the PC master software project has been launched, this option may be selected in the PC master software window under Project/Select Other Map File Reload.*

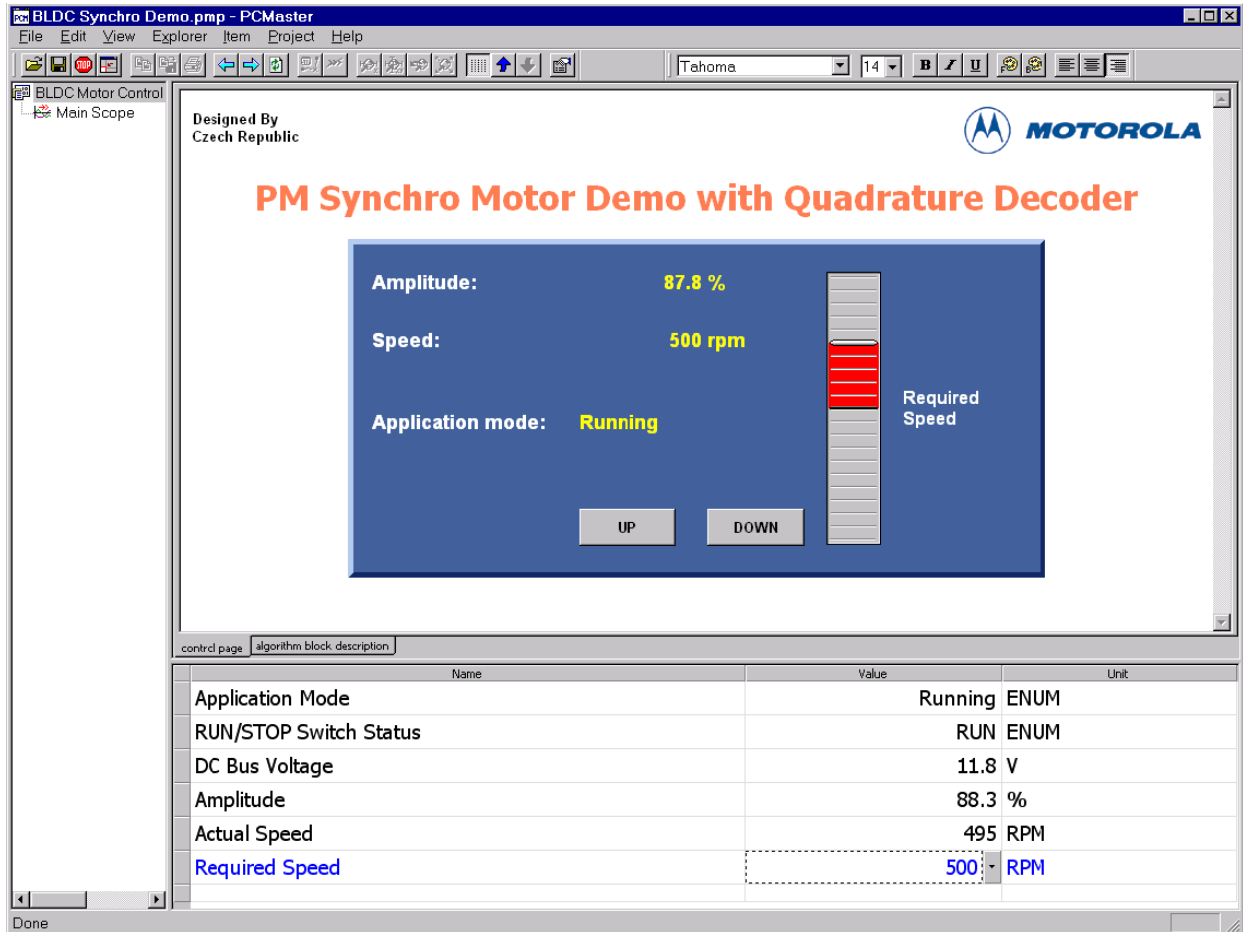
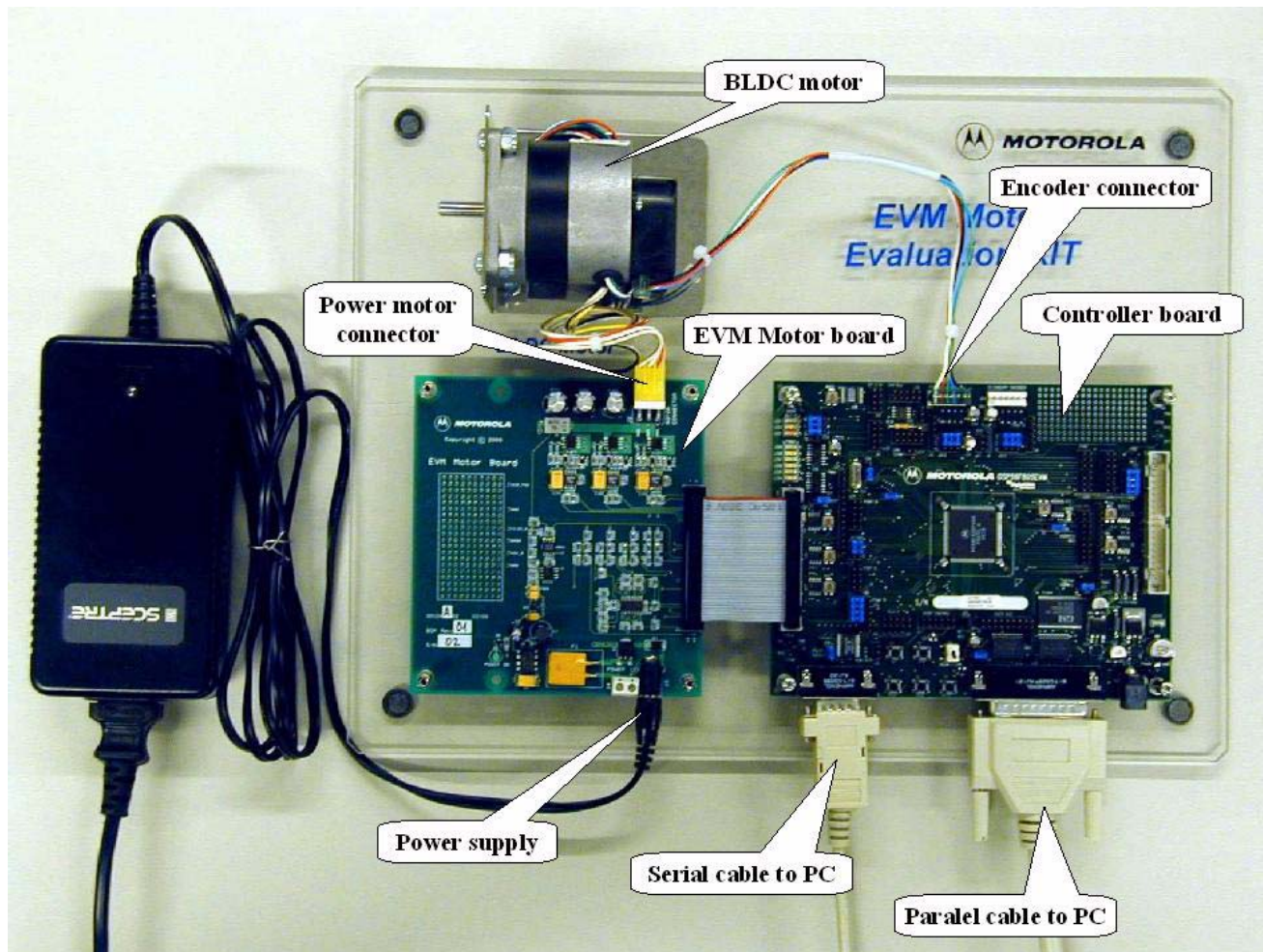


Figure 6-3. PC Master Software Control Window

### 6.3 Application Set-Up

Figure 6-4 illustrates the hardware set-up for the Synchronous PM Motor Control Application with Quadrature Encoder.





**Figure 6-4. Set-up of the BLDC Motor Control Application using DSP56F805EVM**

The system consists of the following components:

- BLDC Motor IB23810
  - supplied in kit ECMTREVAL - Evaluation Motor Board Kit
- EVM Motor Board:
  - supplied in kit with IB23810 Motor: ECMTREVAL - Evaluation Motor Board Kit
- DSP56F805 Board:

- DSP56F805 Evaluation Module, supplied as DSP56F805EVM
- or DSP56F805 Controller Board
- The serial cable - needed for the PC master software debugging tool only.
- The parallel cable - needed for the Metrowerks Code Warrior debugging and s/w loading

For detailed information, refer to the dedicated application note (see References).

## 6.4 PM Synchronous Motor versus BLDC Motor

The application SW is targeted for PM Synchronous motor with sine-wave Back-EMF shape. In this particular demo application the BLDC motor is used instead. This is due to the availability of the BLDC motor supplied as ECMTREVAL. Although the Back-EMF shape of this motor is not ideally sine-wave, it can be controlled by the application SW. The drive parameters will be even better when PMSM motor with exactly sine-wave Back-EMF shape is used.

## 6.5 DSP56F805EVM Set-Up

To execute the Synchronous PM Motor Control with Quadrature Encoder, the DSP56F805EVM board requires the strap settings shown in [Figure 6-5](#) and [Table 6-2](#).

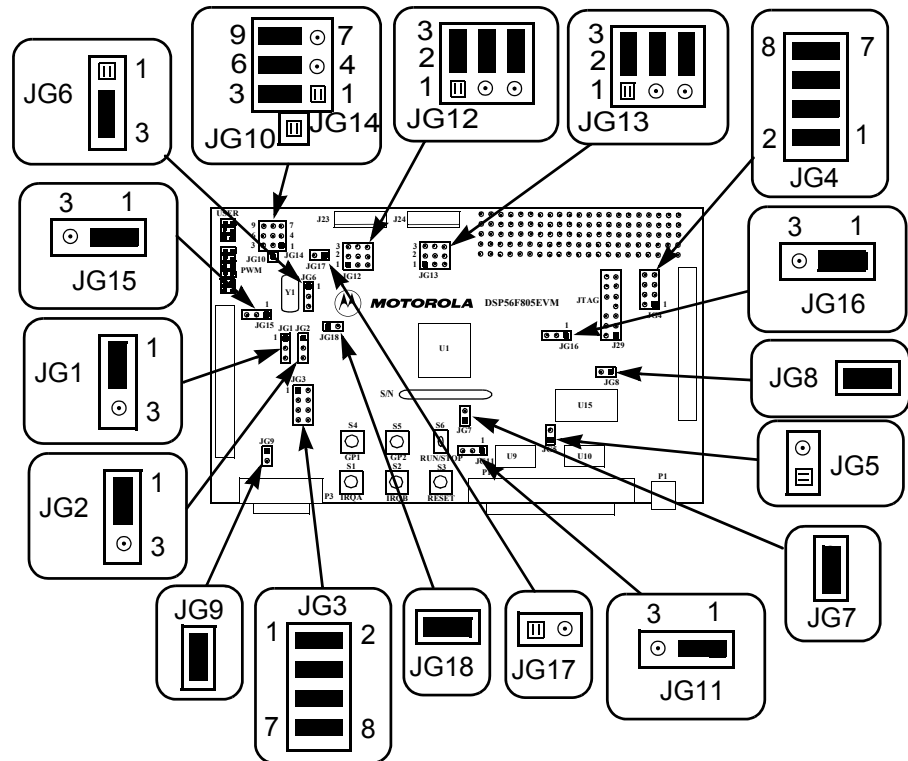


Figure 6-5. DSP56F805EVM Jumper Reference

Table 6-2. DSP56F805EVM Jumper Settings

Jumper Group	Comment	Connections
JG1	PD0 input selected as a high	1-2
JG2	PD1 input selected as a high	1-2
JG3	Primary UNI-3 serial selected	1-2, 3-4, 5-6, 7-8
JG4	Secondary UNI-3 serial selected	1-2, 3-4, 5-6, 7-8
JG5	Enable on-board parallel JTAG Command Converter Interface	NC
JG6	Use on-board crystal for DSP oscillator input	2-3
JG7	Select DSP's Mode 0 operation upon exit from reset	1-2
JG8	Enable on-board SRAM	1-2
JG9	Enable RS-232 output	1-2

Table 6-2. DSP56F805EVM Jumper Settings

Jumper Group	Comment	Connections
JG10	Secondary UNI-3 Analog temperature input unused	NC
JG11	Use Host power for Host target interface	1-2
JG12	Primary Encoder input selected for quadrature encoder signals	2-3, 5-6, 8-9
JG13	Secondary Encoder input selected	2-3, 5-6, 8-9
JG14	Primary UNI-3 3-Phase Current Sense selected as Analog Inputs	2-3, 5-6, 8-9
JG15	Secondary UNI-3 Phase A Over-current selected for FAULTA1	1-2
JG16	Secondary UNI-3 Phase B Over-current selected for FAULTB1	1-2
JG17	CAN termination unselected	NC
JG18	Use on-board crystal for DSP oscillator input	1-2

**NOTE:** When running the EVM target system in a stand-alone mode from Flash, the JG5 jumper must be set in the 1-2 configuration to disable the command converter parallel port interface.

## 6.6 Projects Files

The Synchronous PM Motor Control application is composed of the following files:

- ...\`bldc_synchro_pm_quad_encoder_sa`\bldc\_synchro\_pm\_quad\_encoder\_sa.c, main program
- ...\`bldc_synchro_pm_quad_encoder_sa`\bldc\_synchro\_pm\_quad\_encoder\_sa.mcp, application project file
- ...\`bldc_synchro_pm_quad_encoder_sa`\ApplicationConfig\appconfig.h, application configuration file
- ...\`bldc_synchro_pm_quad_encoder_sa`\SystemConfig\ExtRam\linker\_ram.cmd, linker command file for external RAM

- `...\bldc_synchro_pm_quad_encoder_sa\SystemConfig\Flash\linker_flash.cmd`, linker command file for Flash
- `...\bldc_synchro_pm_quad_encoder_sa\SystemConfig\Flash\flash.cfg`, configuration file for Flash
- `...\bldc_synchro_pm_quad_encoder_sa\PCMaster\bldc_synchro_pm_quad_encoder.pmp`, PC master software file

These files are located in the application directory.

Motor Control algorithms used in the application:

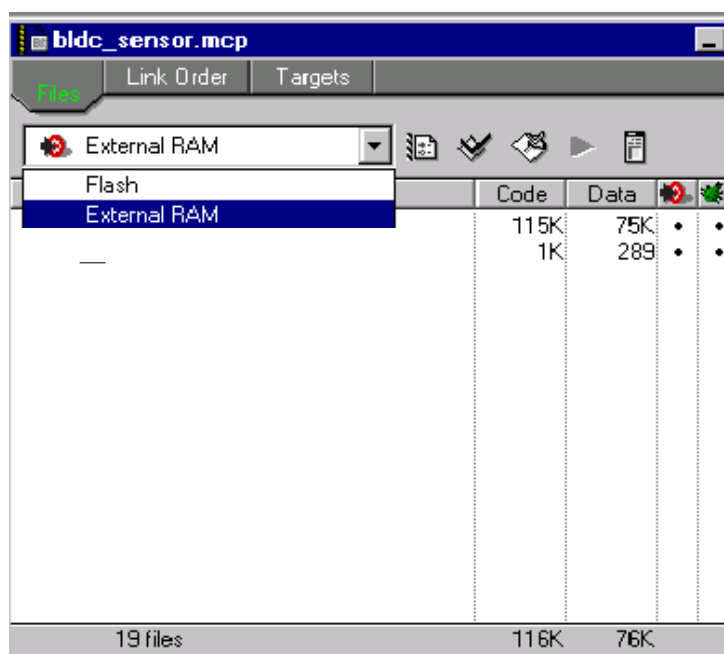
- `...\controllers.c, .h`: source and header files for PI controller
- `...\ramp.c, .h`: source and header files for ramp controller
- `...\sinqquad.c, .h`: source and header files with the sine look-up table
- `...\trigon.c, .h`: source and header files for sine calculation function
- `...\mcgen.c, .h`: source and header files for three-phase sine wave generation

In stand-alone application, all the necessary resources (algorithms and peripheral drivers) are part of the application project file:

- `...\bldc_synchro_pm_quad_encoder_sa\src\include`, folder for general C-header files
- `...\bldc_synchro_pm_quad_encoder_sa\src\dsp56805`, folder for the device specific source files, e.g. drivers
- `...\bldc_synchro_pm_quad_encoder_sa\src\pc_master_support`, folder for PC master software source files
- `...\bldc_synchro_pm_quad_encoder_sa\src\algorithms\`, folder for algorithms

## 6.7 Application Build & Execute

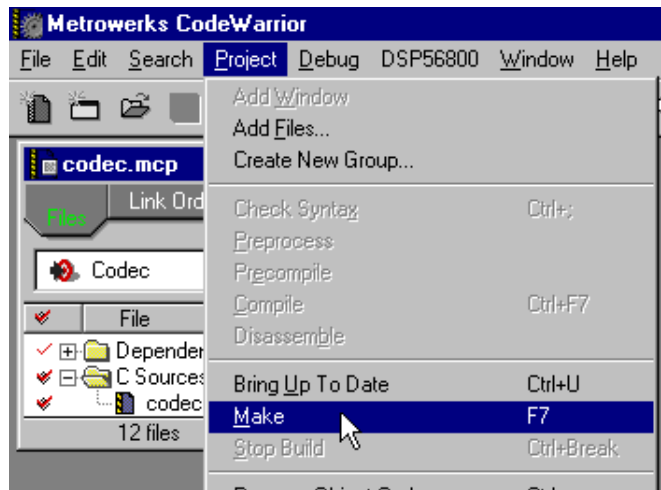
When building the Synchronous PM Motor Control Application with Quadrature Encoder, the user can create an application that runs from internal *Flash* or *External RAM*. To select the type of application to build, open the *bldc\_synchro\_pm\_quad\_encoder.mcp* project and select the target build type, as shown in **Figure 6-6**. A definition of the projects associated with these target build types may be viewed under the *Targets* tab of the project window.



**Figure 6-6. Target Build Selection**

The project may now be built by executing the *Make* command, as shown in **Figure 6-7**. This will build and link the Synchronous PM Motor Control Application with Quadrature Encoder and all needed Metrowerks and Quick\_Start libraries.





**Figure 6-7. Execute *Make* Command**

To execute the Synchronous PM Motor Control application, select *Project\Debug* in the CodeWarrior IDE, followed by the *Run* command. For more help with these commands, refer to the CodeWarrior tutorial documentation in the following file located in the CodeWarrior installation directory:

<...>\CodeWarrior Documentation\PDF\Targeting\_DSP56800.pdf

If the Flash target is selected, CodeWarrior will automatically program the internal Flash of the DSP with the executable generated during *Build*. If the External RAM target is selected, the executable will be loaded to off-chip RAM.

Once Flash has been programmed with the executable, the EVM target system may be run in a stand-alone mode from Flash. To do this, set the JG5 jumper in the 1-2 configuration to disable the parallel port, and press the RESET button.

Once the application is running, move the RUN/STOP switch to the RUN position and set the required speed using the UP/DOWN push buttons. Pressing the UP/DOWN buttons should incrementally increase the motor speed until it reaches maximum speed. If successful, the BLDC motor will be spinning.

**NOTE:** *If the RUN/STOP switch is set to the RUN position when the application starts, toggle the RUN/STOP switch between the STOP and RUN positions to enable motor spinning. This is a protection feature that prevents the motor from starting when the application is executed from CodeWarrior.*

You should also see a lighted green LED, which indicates that the application is running. If the application is stopped, the green LED will blink at a 2Hz frequency. If an Undervoltage fault occurs, the green LED will blink at a frequency of 8Hz.



## Appendix A. References

1. **Brushless DC Motor Control using the MC68HC708MC4**, John Deatherage and Jeff Hunsinger, AN1702/D, Motorola
2. **DSP56F80x MC PWM Module in Motor Control Applications**, Leos Chalupa, AN1927/D, Motorola
3. **Design of Brushless Permanent-magnet Motors**, J.R. Hendershot JR and T.J.E. Miller, Magna Physics Publishing and Clarendon Press, 1994
4. **CodeWarrior for Motorola DSP56800 Embedded Systems**, CWDSP56800, Metrowerks 2001
5. **DSP56F800 16-bit Digital Signal Processor, Family Manual**, DSP56F800FM/D, Motorola 2001
6. **DSP56F80x 16-bit Digital Signal Processor, User's Manual**, DSP56F801-7UM/D, Motorola 2001
7. **DSP56F805 Evaluation Module Hardware User's Manual**, DSP56F805EVMUM/D, Motorola 2001
8. **Evaluation Motor Board User's Manual**, MEMCEVMBUM/D, Motorola
9. **Motorola SPS web page:** *<http://www.motorola.com/>*
10. **3-Phase BLDC Motor Control with Hall Sensors Using DSP56F80x**, Pavel Grasblum, AN1916/D, Motorola 2001

## References

11. **3-Phase BLDC Motor Control with Quadrature Encoder Using DSP56F80x**, Pavel Grasblum, AN1915/D, Motorola 2001
12. **3-Phase PM synchronous Motor Vector Control using DSP56F80x**, Libor Prokop and Pavel Grasblum, AN1931/D, Motorola 2001
13. **3-Phase PM Synchronous Motor Control with Quadrature Encoder Using DSP56F80x**, Pavel Grasblum, AN1917/D, Motorola 2001

## Appendix B. Glossary

**AC** — Alternative Current.

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

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

**BLDC** — Brushless dc motor.

**commutation** — A process providing the creation of a rotation field by switching of power transistor (electronic replacement of brush and commutator)

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

**COP** — Computer Operating Properly timer

**DC** — Direct Current.

**DSP** — Digital Signal Processor.

**DSP56F80x** — A Motorola family of 16-bit hybrid controller dedicated for motor control.

**DT** — see “Dead Time (DT)”

**Dead Time (DT)** — short time that must be inserted between the turning off of one transistor in the inverter half bridge and 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 3-phase motor)

**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.

**JTAG** — Interface allowing On-Chip Emulation and Programming.

**LED** — Light Emitting Diode

**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 ( $V_{SS}$ ).

**PI controller** — Proportional-Integral controller.

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

**PM** — Permanent Magnet

**PMSM** - Permanent Magnet Synchronous Motor.

**PWM** — Pulse Width Modulation.

**Quadrature Decoder** — A module providing decoding of position from a quadrature encoder mounted on a motor shaft.

**Quad Timer** — A module with four 16-bit timers.

**reset** — To force a device to a known condition.

**RPM** — Revolutions per minute.

**SCI** — See "serial communication interface module (SCI)."

**serial communications interface module (SCI)** — A module that supports asynchronous communication.

**serial peripheral interface module (SPI)** — A module that supports synchronous communication.

**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.

**SPI** — See "serial peripheral interface module (SPI)."

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





**Freescale Semiconductor, Inc.**

**Freescale Semiconductor, Inc.**

**For More Information On This Product,  
Go to: [www.freescale.com](http://www.freescale.com)**

**HOW TO REACH US:****USA/EUROPE/LOCATIONS NOT LISTED:**

Motorola Literature Distribution;  
P.O. Box 5405, Denver, Colorado 80217  
1-303-675-2140 or 1-800-441-2447

**JAPAN:**

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

**ASIA/PACIFIC:**

Motorola Semiconductors H.K. Ltd.;  
Silicon Harbour Centre, 2 Dai King Street,  
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong  
852-26668334

**TECHNICAL INFORMATION CENTER:**

1-800-521-6274

**HOME PAGE:**

<http://motorola.com/semiconductors>

Information in this document is provided solely to enable system and software implementers to use Motorola 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.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part.



Motorola and the Stylized M Logo are registered in the U.S. Patent and Trademark Office. digital dna is a trademark of Motorola, Inc. All other product or service names are the property of their respective owners. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© Motorola, Inc. 2003

DRM029/D

**For More Information On This Product,  
Go to: [www.freescale.com](http://www.freescale.com)**