



Freescale Semiconductor, Inc.

*3-Phase BLDC
Motor Control
with Hall Sensors
Using 56F805*

*Designer Reference
Manual*

*56800
Hybrid Controller*

*DRM025/D
Rev. 0, 03/2003*

MOTOROLA.COM/SEMICONDUCTORS

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

3-Phase BLDC Motor Control with Hall Sensors 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

Section 1. Introduction 13

Section 2. Target Motor Theory and Control 19

Section 3. System Description. 29

Section 4. Hardware Design. 33

Section 5. Software Design 39

Section 6. System Setup 51

Appendix A. References. 63

Appendix B. Glossary. 65

Table of Contents

Section 1. Introduction

1.1	Contents	13
1.2	Application Benefit	13
1.3	Motorola DSP Advantages and Features	14

Section 2. Target Motor Theory and Control

2.1	Contents	19
2.2	Brushless DC Motor Theory	19
2.3	Digital Control of a BLDC Motor	20

Section 3. System Description

3.1	Contents	29
3.2	System Outline	29
3.3	Application Description	30

Section 4. Hardware Design

4.1	Contents	33
4.2	System Configuration	33
4.3	DSP56F805EVM Controller Board	34
4.4	EVM Motor Board	35
4.5	Hardware Documentation	38

Section 5. Software Design

Table of Contents

5.1 Contents39
5.2 Software Description.....39
5.3 Data Flow39
5.4 Software Implementation44
5.5 Implementation Notes.....48

Section 6. System Setup

6.1 Contents51
6.2 Operational Modes51
6.3 Application Set-Up54
6.4 DSP56F805EVM Set-Up56
6.5 Projects Files58
6.6 Application Build & Execute59

Appendix A. References

Appendix B. Glossary

List of Figures

Figure	Title	Page
2-1	BLDC Motor - Cross Section	19
2-2	Voltage Strokes Applied To the 3-ph BLDC Motor	20
2-3	3-phase BLDC Power Stage.	21
2-4	Independent Switching of Power Transistors	22
2-5	Complementary Switching of Power Transistors	23
2-6	Stator Flux Vectors at Six-Step Control	24
2-7	Situation Right Before Commutation	25
2-8	Situation Right After Commutation	25
2-9	Speed Controller.	27
3-1	System Concept.	31
4-1	Low-Voltage Evaluation Motor HW System Configuration	33
4-2	Block Diagram of the DSP56F805EVM	35
4-3	EVM Motor Board.	36
5-1	Main Data Flow.	40
5-2	PWM Swapping and Masking for BLDC Motor.	44
5-3	State Diagram - General Overview.	45
5-4	Drive State Machine Transitions.	48
6-1	RUN/STOP Switch and UP/DOWN Buttons at DSP56F805EVM	52
6-2	USER and PWM LEDs at DSP56F805EVM.	52
6-3	PC Master Software Control Window	54
6-4	Set-up of the BLDC Motor Control Application using DSP56F805EVM.	55
6-5	DSP56F805EVM Jumper Reference	56
6-6	Target Build Selection.	59
6-7	Execute Make Command	60

List of Tables

Table	Title	Page
1-1	Memory Configuration	15
2-1	Commutation Sequence for Clockwise Rotation	26
2-2	Commutation Sequence for Counterclockwise Rotation	26
3-1	Specifications of the Motor and Hall Sensors	30
4-1	Electrical Characteristics	37
4-2	Motor Characteristics	37
6-1	Motor Application States	53
6-2	DSP56F805EVM Jumper Settings	57

Section 1. Introduction

1.1 Contents

1.2	Application Benefit	13
1.3	Motorola DSP Advantages and Features	14

1.2 Application Benefit

This Reference Design describes the design of a 3-phase BLDC (Brushless DC) motor drive based on Motorola's DSP56F805 dedicated motor control device.

BLDC motors are very popular in a wide application area. Compared with a DC motor, the BLDC motor uses the electric commutator (replacing the mechanical commutator) and so it is more reliable than the DC motor. In BLDC motors, rotor magnets generate the rotor's magnetic flux so that BLDC motors achieve higher efficiency. Therefore BLDC motors may be 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 the application is a speed-closed loop BLDC drive using a Hall position sensor. It serves as an example of a BLDC 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 visualization tool.

1.3 Motorola DSP Advantages and Features

The Motorola DSP56F80x family is well suited for digital motor control, combining on a single chip the DSP's calculation capability with the MCU's controller features. These DSPs offer many dedicated peripherals like a Pulse Width Modulation (PWM) module, 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
- Twelve-bit Analog-to-Digital Converters (ADCs), supporting two simultaneous conversions with dual 4-pin multiplexed inputs; the ADC can be synchronized by 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 6 pins: Timer C with 2 pins and Timer D with 4 pins
- CAN 2.0 A/B Module with 2-pin ports used to transmit and receive
- Two Serial Communication Interfaces (SCI0 & SCI1), each with two pins, or four additional GPIO lines
- Serial Peripheral Interface (SPI) with configurable 4-pin port, or four additional GPIO lines
- Computer Operating Properly (COP) timer
- Two dedicated external interrupt pins
- Fourteen dedicated General Purpose I/O (GPIO) pins, 18 multiplexed GPIO pins
- External reset pin for hardware reset
- JTAG/On-Chip Emulation (OnCE)

- 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

In addition to the fast Analog-to-Digital converter and the 16-bit Quad Timers, the most interesting peripheral from the BLDC 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 BLDC 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 one to 16
- Individual software-controlled PWM outputs
- Mask and swap of PWM outputs

- Programmable fault protection
- Polarity control
- 20-mA current sink capability on PWM pins
- Write-protectable registers

The BLDC 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 outputs can be controlled separately by software allowing the setting of the control signal to logical 0 or 1. This feature is used to enable/disable control signal by software. The PWM module's swap function allows the immediate swap of top and bottom control signals in the same phase. These functions permit the separation of the rotor commutation and speed control to the two independent program parts. The state of the control signals can be changed immediately when required by the motor position (phase commutation) without changing the content of the PWM value registers. These changes can be accomplished asynchronously to the PWM duty cycle update.

The Quad Timer is provides 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 BLDC motor application utilizes three channels of a Quad Timer module in Input Capture mode. It enables sensing of the rotor position using position Hall sensors. The fourth channel of the Quad Timer module is set to generate time base for a speed controller.

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

- Includes 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 SW or external events
- Preloadable 16-bit revolution counter
- Inputs can be connected to a general purpose timer to aid low speed velocity.

Since the Quadrature Decoder also has all input pins connected to the Quad Timer module A, the BLDC application uses the digital input filter to filter Hall sensor signals.

Section 2. Target Motor Theory and Control

2.1 Contents

2.2	Brushless DC Motor Theory	19
2.3	Digital Control of a BLDC Motor	20

2.2 Brushless DC Motor Theory

A brushless DC (BLDC) 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**).

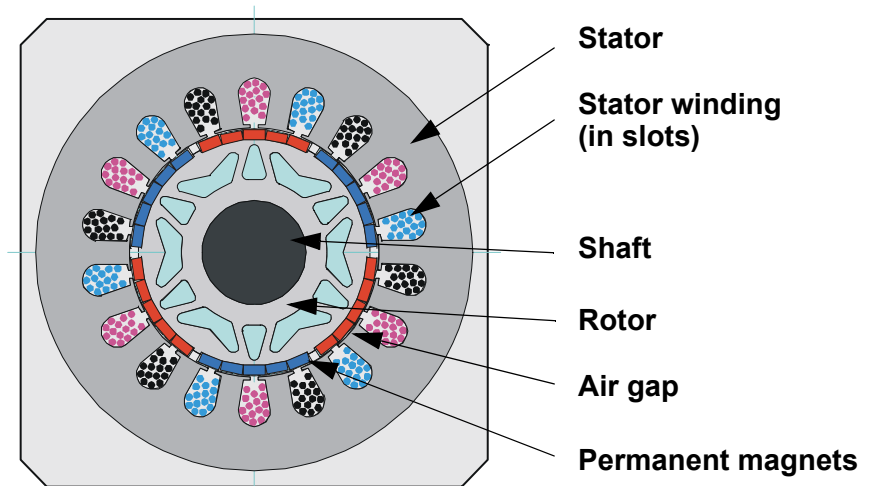


Figure 2-1. BLDC Motor - Cross Section

In this respect, the BLDC motor is equivalent to a reversed DC commutator motor, in which the magnet rotates while the conductors remain stationary. In the DC commutator motor, the current polarity is

altered by the commutator and brushes. On the contrary, in the brushless DC motor, the polarity reversal is performed by power transistors switching in synchronization with the rotor position. Therefore, BLDC motors often incorporate either internal or external position sensors to sense the actual rotor position or the position can be detected without sensors.

2.3 Digital Control of a BLDC Motor

The BLDC motor is driven by rectangular voltage strokes coupled with the given rotor position (see [Figure 2-2](#)). The generated stator flux interacts with the rotor flux, which is generated by a rotor magnet, defines the torque and thus speed of the motor. The voltage strokes must be properly applied to the two phases of the three-phase winding system so that the angle between the stator flux and the rotor flux is kept close to 90° to get the maximum generated torque. Due to this fact, the motor requires electronic control for proper operation.

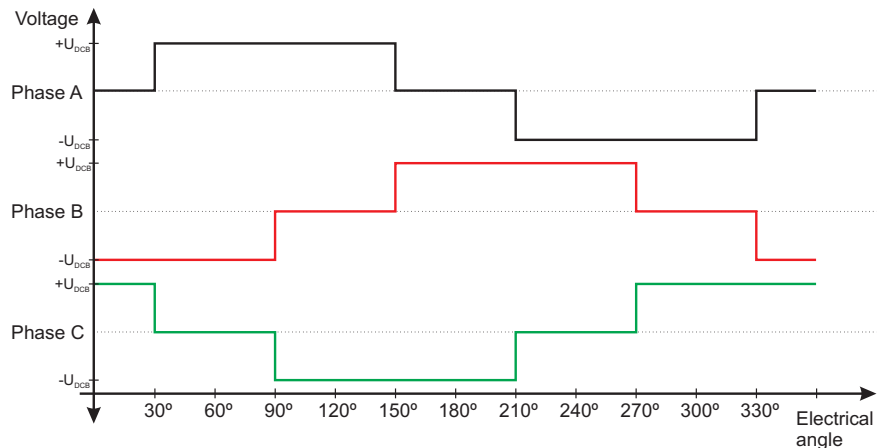


Figure 2-2. Voltage Strokes Applied To the 3-ph BLDC Motor

For the common 3-phase BLDC motor a standard 3-phase power stage is used as is illustrated in [Figure 2-3](#). The power stage utilizes six power transistors with switching in either the independent mode or complementary mode.

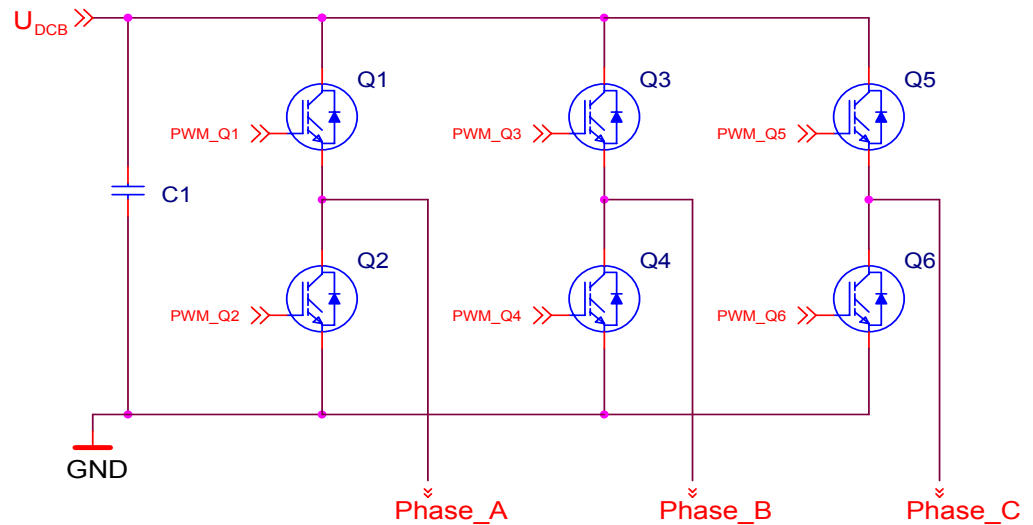


Figure 2-3. 3-phase BLDC Power Stage

In both modes the 3-phase power stage energizes two motor phases concurrently. The third phase is unpowered (see Figure 2-2). Thus, we get six possible voltage vectors that are applied to the BLDC motor using a PWM technique (see Figure 2-3 and Figure 2-4). There are two basic types of power transistor switching: independent switching and complementary switching.

2.3.1 Independent Switching of Power Transistors

With independent switching, only two transistors are switched on when current is conducted from the power supply to the phase of the BLDC motor. In one phase, the top transistor is switched on; in second phase the bottom transistor is switched on and the third phase is not powered. During freewheeling all transistors are switched off (see Figure 2-4).

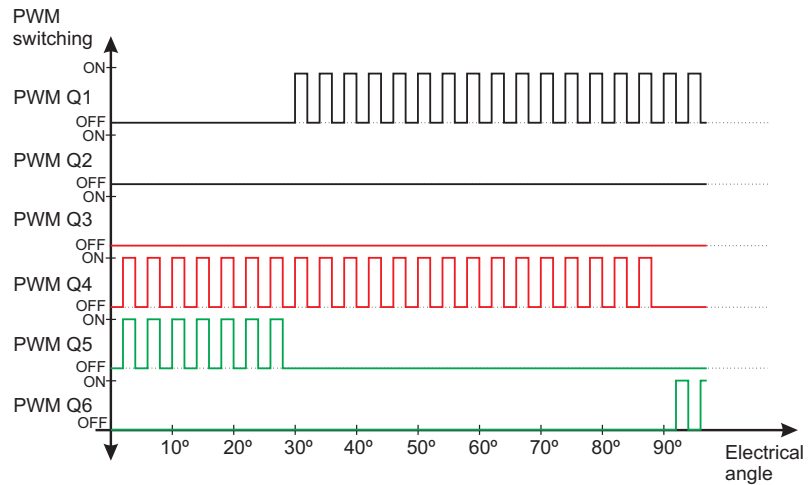


Figure 2-4. Independent Switching of Power Transistors

2.3.2 Complementary Switching of Power Transistors

With complementary switching, two transistors are switched on when the phase of the BLDC motor is connected to the power supply. But there is a difference during freewheeling. With independent switching all the transistors are switched off. The current continues to flow in the same direction through freewheeling diodes and until it falls to zero. Contrary to this, with complementary switching the complementary transistors are switched on during freewheeling. Thus the current may be able to flow in the opposite direction. **Figure 2-5** depicts the complementary switching.

NOTE: Both described switching modes are able to work in bipolar or unipolar mode. For detailed information about bipolar and unipolar modes see the Application Note *DSP56F80x MC PWM Module in Motor Control Applications*. **Figure 2-4** and **Figure 2-5** illustrate the bipolar switching mode. The presented application utilizes the complementary unipolar PWM mode.

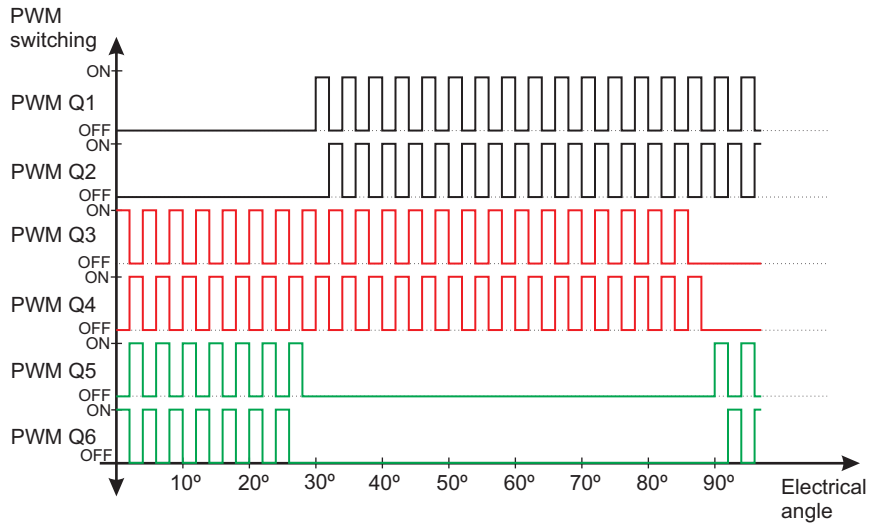


Figure 2-5. Complementary Switching of Power Transistors

2.3.3 Commutation

Commutation provides the creation of a rotation field. As was explained, for proper operation of a BLDC motor it is necessary to keep the angle between stator and rotor flux close to 90°. With six-step control we get a total of six possible stator flux vectors. The stator flux vector must be changed at a certain rotor position. The rotor position is usually sensed by Hall sensors. The Hall sensors generate three signals that also comprise six states. Each of Hall sensors' states corresponds to certain stator flux vector. All Hall sensor states with corresponding stator flux vectors are illustrated in [Figure 2-6](#). The same figure is illustrated in tables [Table 2-1](#) and [Table 2-2](#).

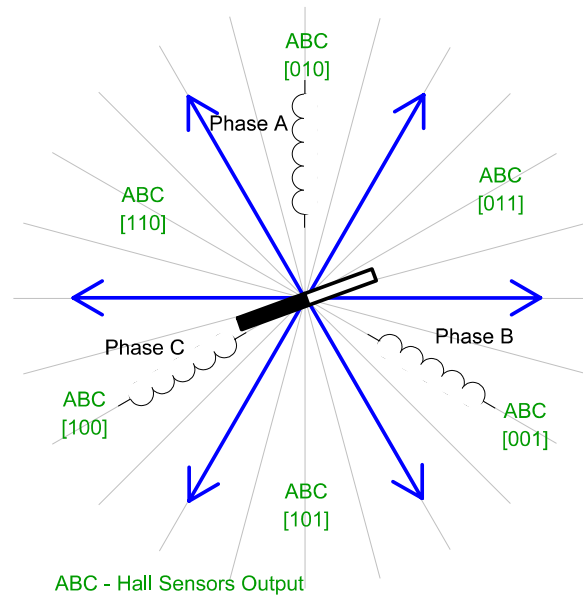


Figure 2-6. Stator Flux Vectors at Six-Step Control

The next two figures depict the commutation process. The actual rotor position in **Figure 2-7** corresponds to the Hall sensors' state ABC[110] (see **Figure 2-6**). The actual voltage pattern can be derived from the **Table 2-1**. Phase A is connected to the positive DC bus voltage by the transistor Q1; phase C is connected to the ground by transistor Q6 and phase B is unpowered.

As soon as the rotor reaches a certain position (see **Figure 2-7**) the Hall sensors' state changes its value from ABC[110] to ABC[100]. From **Table 2-1** a new voltage pattern is selected and applied to the BLDC motor.

As can be seen, using a six-step control technique there is no possibility of keeping the angle between the rotor flux and the stator flux precisely at 90° . The real angle varies from 60° to 120° .

The commutation is repeated per each 60 electrical degrees. The commutation event is critical for its angular (time) accuracy. Any deviation causes the torque ripples and hence speed variation.

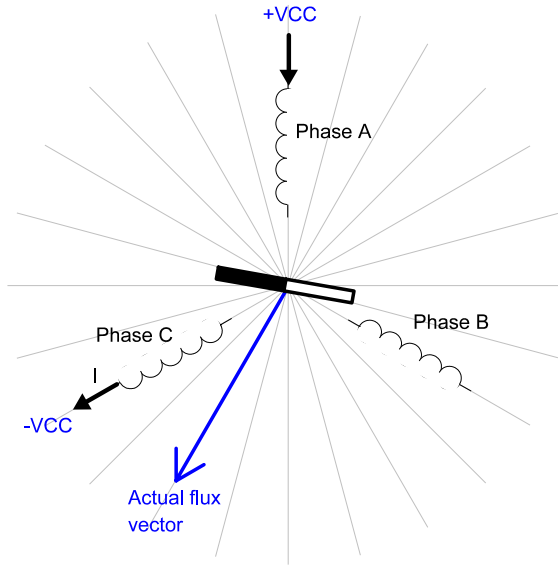


Figure 2-7. Situation Right Before Commutation

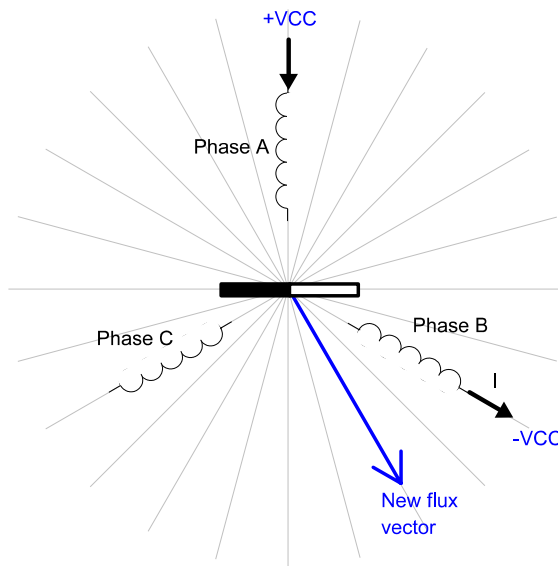


Figure 2-8. Situation Right After Commutation

Table 2-1. Commutation Sequence for Clockwise Rotation

Hall Sensor A	Hall Sensor B	Hall Sensor C	Phase A	Phase B	Phase C
1	0	0	$-V_{DCB}$	$+V_{DCB}$	NC
1	0	1	NC	$+V_{DCB}$	$-V_{DCB}$
0	0	1	$+V_{DCB}$	NC	$-V_{DCB}$
0	1	1	$+V_{DCB}$	$-V_{DCB}$	NC
0	1	0	NC	$-V_{DCB}$	$+V_{DCB}$
1	1	0	$-V_{DCB}$	NC	$+V_{DCB}$

Table 2-2. Commutation Sequence for Counterclockwise Rotation

Hall Sensor A	Hall Sensor B	Hall Sensor C	Phase A	Phase B	Phase C
1	0	0	$+V_{DCB}$	$-V_{DCB}$	NC
1	1	0	$+V_{DCB}$	NC	$-V_{DCB}$
0	1	0	NC	$+V_{DCB}$	$-V_{DCB}$
0	1	1	$-V_{DCB}$	$+V_{DCB}$	NC
0	0	1	$-V_{DCB}$	NC	$+V_{DCB}$
1	0	1	NC	$-V_{DCB}$	$+V_{DCB}$

2.3.4 Speed Control

Commutation ensures the proper rotor rotation of the BLDC motor, while the motor speed only depends on the amplitude of the applied voltage. The amplitude of the applied voltage is adjusted by using the PWM technique. The required speed is controlled by a speed controller. The speed controller is implemented as a conventional PI controller. The difference between the actual and required speed is input to the PI controller and, based on this difference, the PI controller controls duty cycle of PWM pulses which corresponds to the voltage amplitude required to keep the required speed.

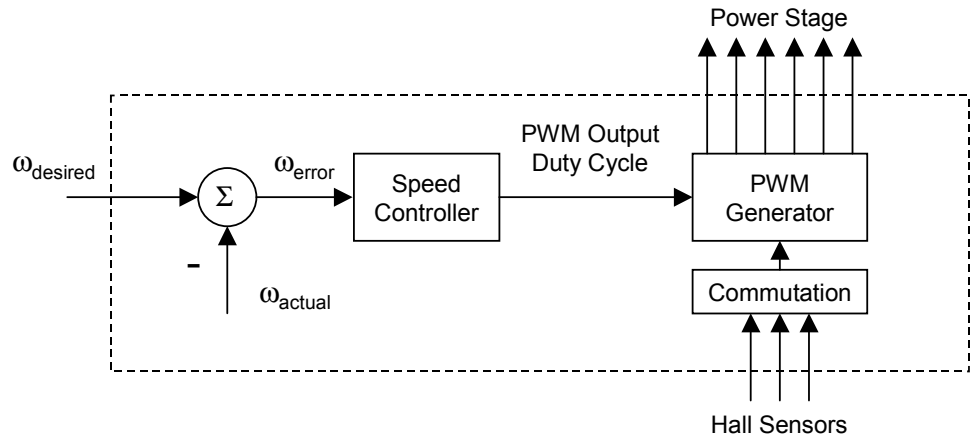


Figure 2-9. Speed Controller

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

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

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

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

$$u(k) = u_p(k) + u_I(k) \tag{EQ 2-3.}$$

$$u_p(k) = K_c \cdot e(k) \tag{EQ 2-4.}$$

$$u_I(k) = u_I(k-1) + K_c \frac{T}{T_I} \cdot e(k) \tag{EQ 2-5.}$$

where:

- $e(k)$ = Input error in step k
- $w(k)$ = Desired value in step k
- $m(k)$ = Measured value in step k
- $u(k)$ = Controller output in step k

$u_p(k)$ = Proportional output portion in step k

$u_i(k)$ = Integral output portion in step k

$u_i(k-1)$ = Integral output portion in step k-1

T_I = Integral time constant

T = Sampling time

K_c = Controller gain

Section 3. System Description

3.1 Contents

3.2	System Outline	29
3.3	Application Description	30

3.2 System Outline

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

- Voltage control of BLDC motor using Hall sensors
- Targeted for DSP56F805EVM
- Running on 3-phase EVM Motor Board
- Control technique incorporates:
 - Voltage BLDC motor control with speed-closed loop
 - Both directions of rotation
 - Motoring mode
 - Start from any motor position without rotor alignment
 - Minimal speed 500 RPM
 - Maximal 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)
- 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)
- PC master software Hall sensors Scope (observes actual Hall sensors' state)
- DC Bus under-voltage fault protection

The introduced BLDC drive is designed to power a low-voltage BLDC motor equipped with Hall sensors, which is supplied with the EVM Motor Board. The motor has the following specifications:

Table 3-1. Specifications of the Motor and Hall Sensors

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

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 Hall sensors
- 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 a 3-phase inverter.

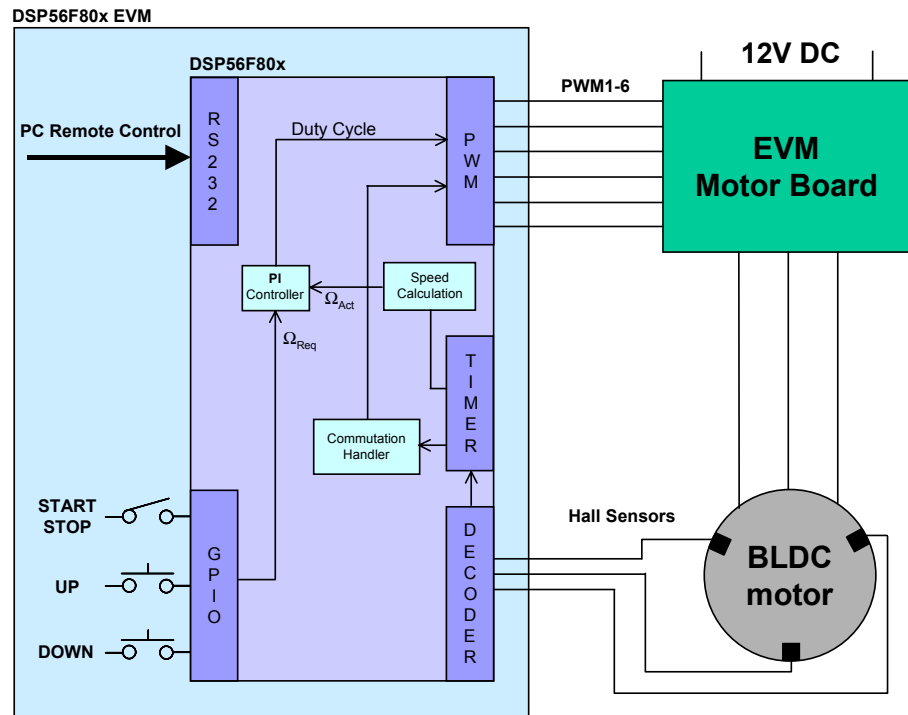


Figure 3-1. System Concept

The **control process** is as follows:

The state of the user interface is periodically scanned, while the speed of the motor is measured each new coming edge from the Hall sensors (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 input to the speed PI controller that generates a new corrected duty cycle. The duty cycle value together with commutation algorithm creates the PWM output signals for the BLDC power stage.

The Hall sensor signals are scanned independently on a speed control. Each new coming edge of any Hall sensor signal calls the interrupt routine which provides the commutation algorithm.

In case of under-voltage, the PWM outputs are disabled and the fault state is displayed.

Section 4. Hardware Design

4.1 Contents

4.2	System Configuration	33
4.3	DSP56F805EVM Controller Board	34
4.4	EVM Motor Board	35
4.5	Hardware Documentation	38

4.2 System Configuration

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

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

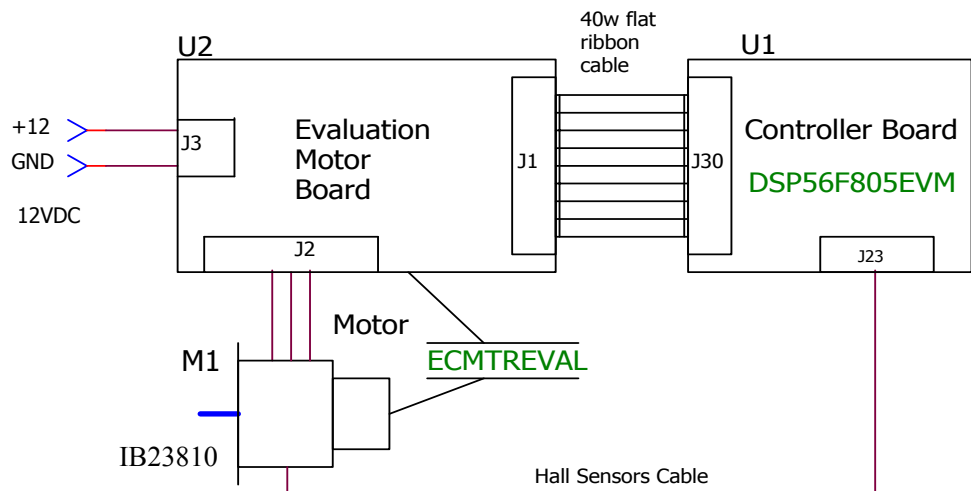


Figure 4-1. Low-Voltage Evaluation Motor HW System Configuration

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**.

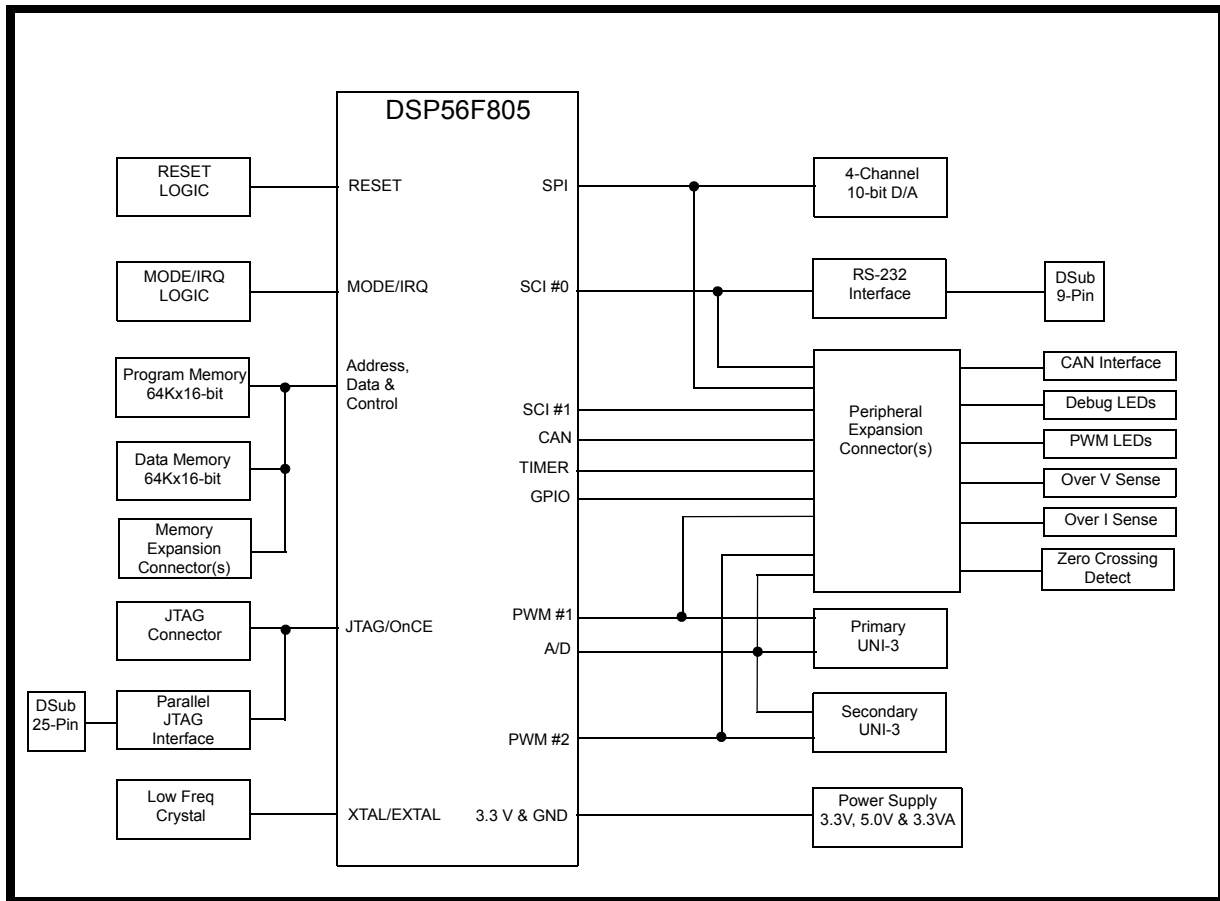


Figure 4-2. Block Diagram of the DSP56F805EVM

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.

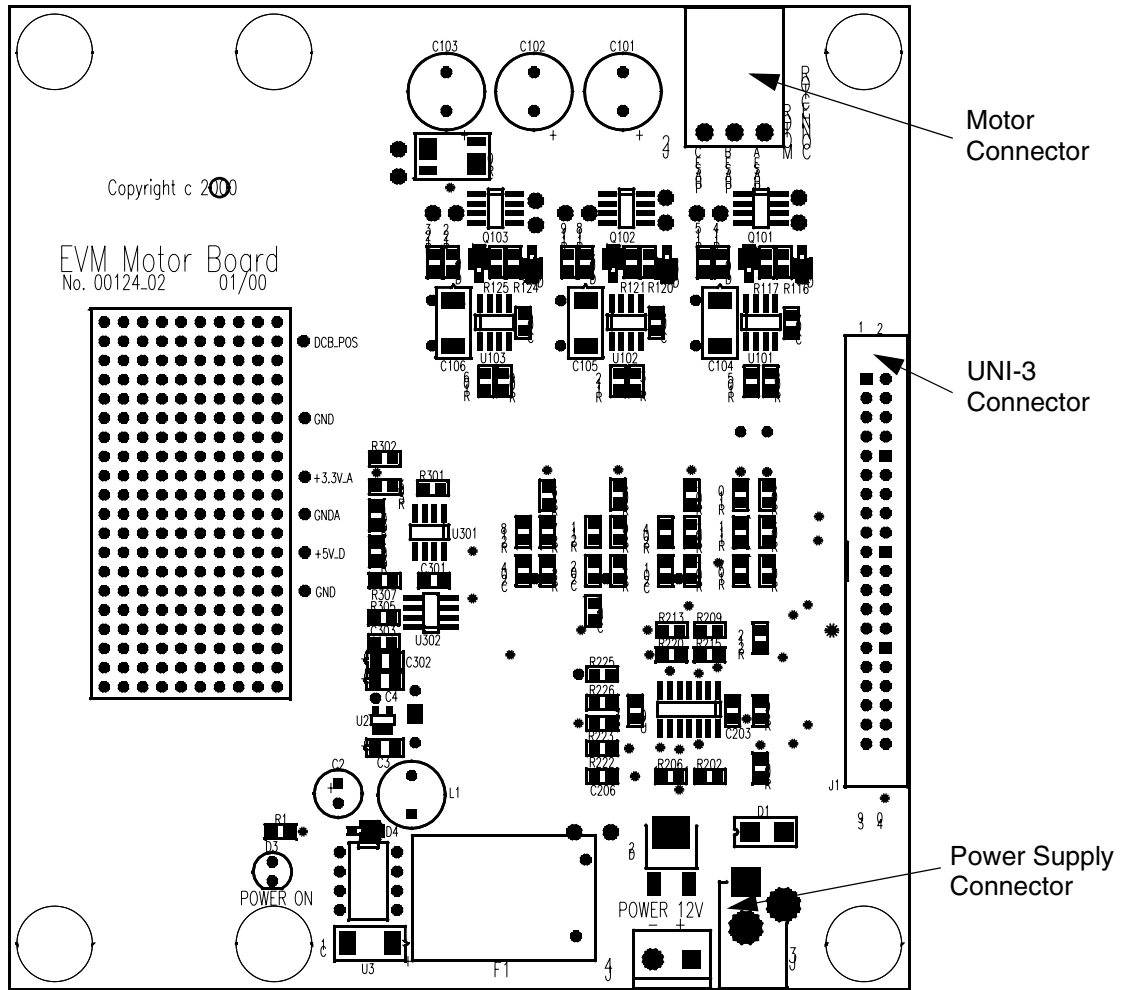


Figure 4-3. EVM Motor Board

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 _{dc}	10	12	16	V
Quiescent Current	I _{CC}	—	50	—	mA
Min Logic 1 Input Voltage	V _{IH}	2.4	—	—	V
Max Logic 0 Input Voltage	V _{IL}	—	—	0.8	V
Input Resistance	R _{In}	—	10	—	kΩ
Analog Output Range	V _{Out}	0	—	3.3	V
Bus Current Sense Voltage	I _{Sense}	—	412	—	mV/A
Bus Voltage Sense Voltage	V _{Bus}	—	206	—	mV/V
Power MOSFET On Resistance	R _{DS(On)}	—	32	40	MΩ
RMS Output Current	I _M	—	—	6	A
Total Power Dissipation	P _{diss}	—	—	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 _t	—	—	60	V
Speed @ V _t		—	5000	—	RPM
Torque Constant	K _t	—	0.08	—	Nm/A
Voltage Constant	K _e	—	8.4	—	V/kRPM
Winding Resistance	R _t	—	2.8	—	Ω
Winding Inductance	L	—	8.6	—	mH
Continuous Current	I _{cs}	—	—	2	A
Peak Current	I _{ps}	—	—	5.9	A

Table 4-2. Motor Characteristics

Inertia	J_m	—	0.075	—	kgcm ²
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	39
5.3	Data Flow	39
5.4	Software Implementation	44
5.5	Implementation Notes	48

5.2 Software Description

This section describes the design of the software blocks of 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 closed loop BLDC drive is described in **Figure 5-1**. The individual processes are described in the following sections. The main data flow can be divided into four parts:

- Speed control
- Velocity calculation
- Rotor commutation
- DC Bus voltage measurement

Speed control starts with the `omega_required_mech` variable. 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.

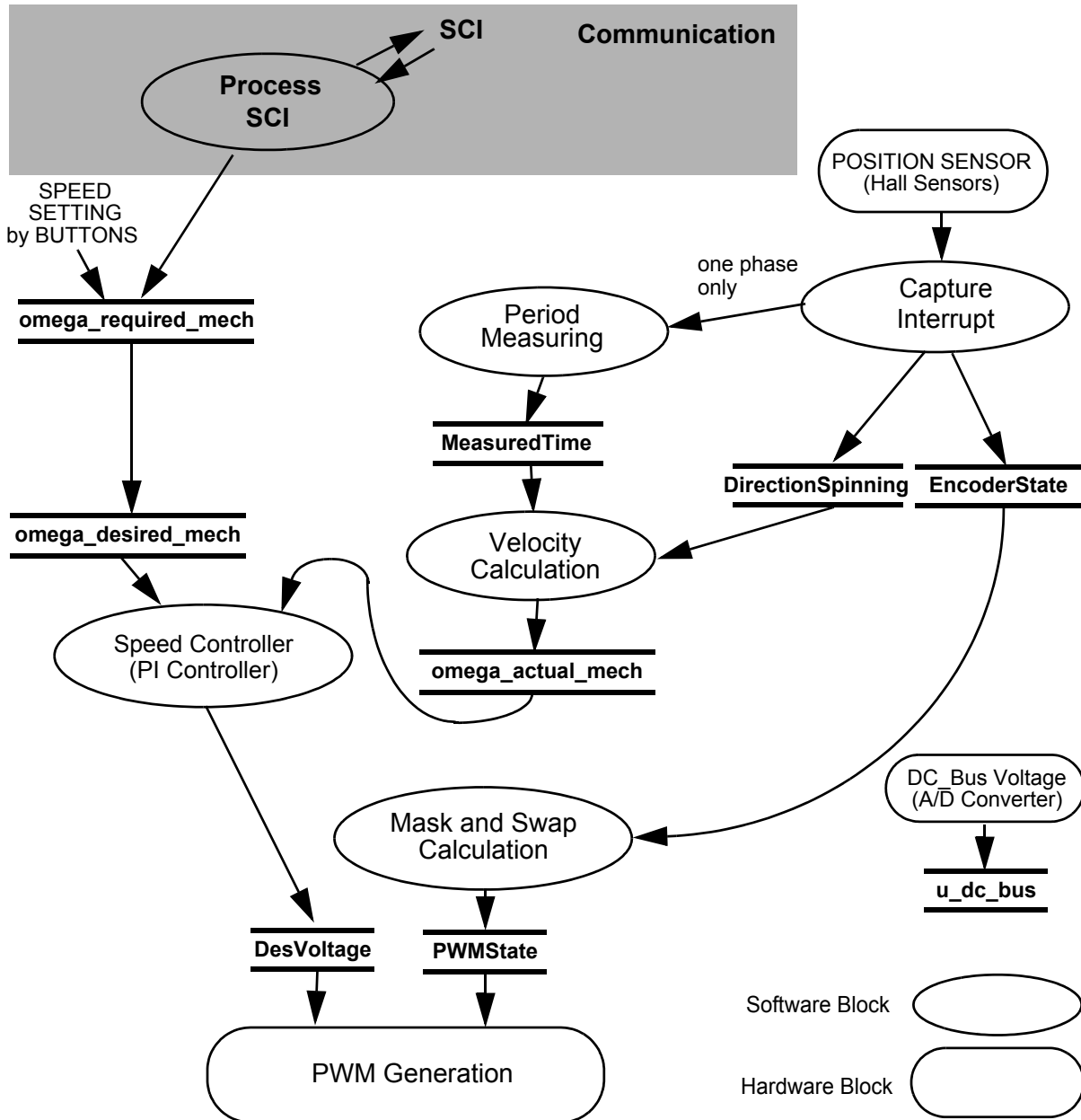


Figure 5-1. Main Data Flow

`MeasuredTime` incorporates a time period of one Hall sensor phase. The phase time period is used for a speed calculation. Calculated speed, `omega_actual_mech`, is input to the speed PI controller as a secondary input. The PI controller output determines the duty cycle of the generated PWM output signals.

Each new edge of the Hall sensor signals a capture interrupt to be called. The interrupt routine saves the actual Hall sensor state to `EncoderState`. The `EncoderState` variable is input to mask and swap calculation, which calculates the final shape of the output voltage. The output variable `PWMState` is written directly to the PWM block. The next task, which is provided by interrupt routine, is calculation of the spin direction. The result, `DirectionSpinning`, is used for the speed calculation.

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

5.3.1 Latest Position Capture, Period Measuring and Velocity Calculation

Latest Position Capture, Period Measuring and Velocity Calculation processes relate to the Hall sensor inputs. The sensors generate streams of pulses that are captured (separately for each sensor) by the Input Capture (IC) function.

The process, Latest Position Capture, is called by an input capture interrupt and captures the latest state of the Hall sensors.

The processes, Period Measuring and Velocity Calculation, read the time between the adjacent edges of Hall sensor output (one phase only), and calculates the actual motor speed `omega_actual_mech`.

5.3.2 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](#).

5.3.3 Mask and Swap Calculation

This process performs a rotor commutation. As already mentioned, the only two phases are powered by a six-step control. The proper PWM output can be generated by changing the PWM value (duty cycle) registers only. This has two disadvantages. The first is that the speed controller, which changes the duty cycle, affects the commutation algorithm (performed by changing the duty cycle). The second disadvantage is that a change in the duty cycle is synchronized with PWM reload which may cause a delay between a proper commutation moment and the PWM reload. This is especially pronounced at high speed when the commutation period is very short.

The DSP56F80x family has two features dedicated to BLDC motor control: the ability to swap odd and even PWM generator outputs, and the ability to mask (disable) any PWM generator outputs. These two features allow creation of a rotational field without changing the contents of the PWM value registers. The influence of masking and swapping on the PWM generator outputs is illustrated in [Figure 5-2](#). Detailed information about PWM settings for BLDC motors can be found in the Application Note DSP56F80x MC PWM Module in Motor Control Applications.

The commutation algorithm `bldchsCommHandlerComp` calculates, based on the actual Hall sensor state, the output `PWMState`. The structure of `PWMState` consists of two parts (variables). The first part (`PWMState.Swap`) defines the swapping of phases and the second part (`PWMState.Mask`) defines phase masking. Then these values are written directly to the PWM generator. The swapping value is written to the PWM Channel Control Register and the masking value is written to the PWM Output Control Register. The following paragraph describes in detail the commutation transition from the one to the next commutation sector as is depicted on [Figure 2-7](#), [Figure 2-8](#) and [Figure 5-2](#).

The even PWM value registers are set e.g. to 75% duty cycle. The odd PWM value registers are set to complementary value $100\% - 75\% = 25\%$ duty cycle. The rotor in [Figure 2-7](#) is situated in sector ABC[110]. The state of masking and swapping is following (see [Table 2-2](#)):

- Phase A
not masked, not swapped (the Phase A is connected to positive DC-Bus voltage)
- Phase B
masked/disabled
- Phase C
not masked, swapped (the Phase C is connected to negative DC-Bus voltage. The swap of Phase B provides the identical pulses on the upper switch of Phase A and the lower switch of Phase C and, conversely, on the lower switch of Phase A and the upper switch of Phase C)

As soon as the rotor passes the sector border ([Figure 2-8](#)) the commutation is done in two steps:

- swap Phase B (unconnected phase)
- mask/disable Phase C; unmask/enable Phase B

After commutation the state of masking and swapping is (see [Table 2-2](#)):

- Phase A
not masked, not swapped (the Phase A is connected to positive DC-Bus voltage)
- Phase B
not masked, swapped (the Phase B is connected to negative DC-Bus voltage.)
- Phase C
masked/disabled

This commutation process is repeated six times per electrical revolution.

NOTE: *In complementary switching mode, it is necessary to use the software control feature for masking. For independent mode, it is possible to use the masking feature in the PWM Channel Control Register. This feature works properly in independent mode only.*

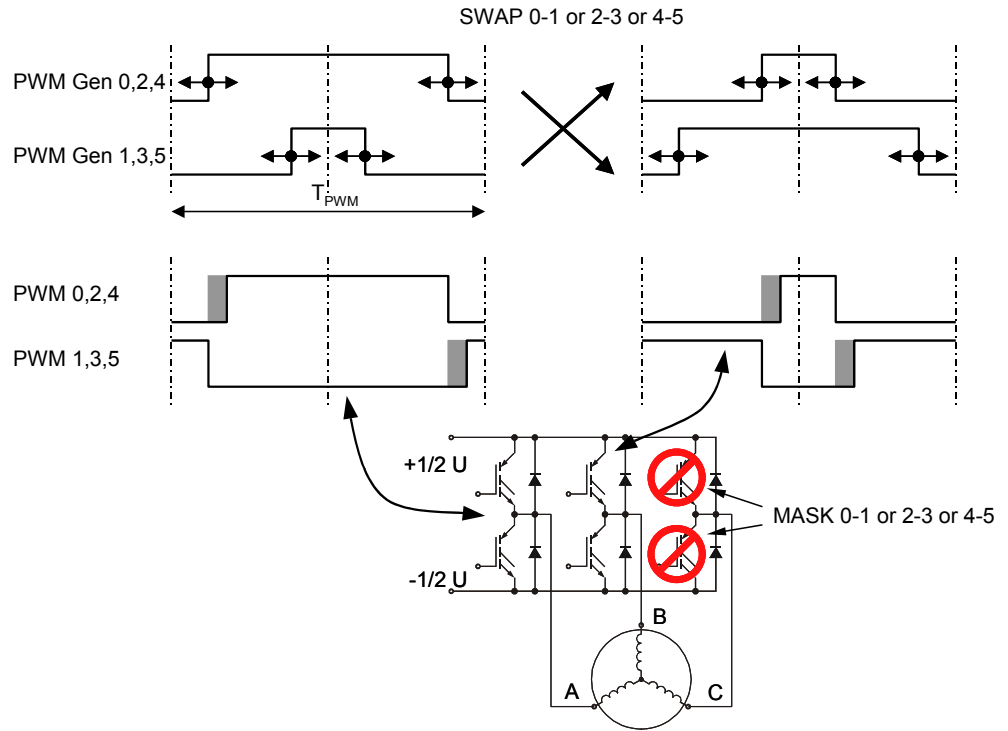


Figure 5-2. PWM Swapping and Masking for BLDC Motor

5.4 Software Implementation

The general software diagram incorporates the Main routine entered from Reset and the interrupt states (see [Figure 5-3](#)).

The Main routine provides the initialization of the DSP, initialization of the application and then it enters an infinite background loop. The background loop contains an application state machine.

The following interrupt service routines are utilized:

- Input Capture ISR (TimerA0, A1, A2) - services signals generated by the Hall sensors
- Input Capture ISR (TimerA3) - 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 - services Up and Down push buttons
- SCI ISR - services communication with the PC master software

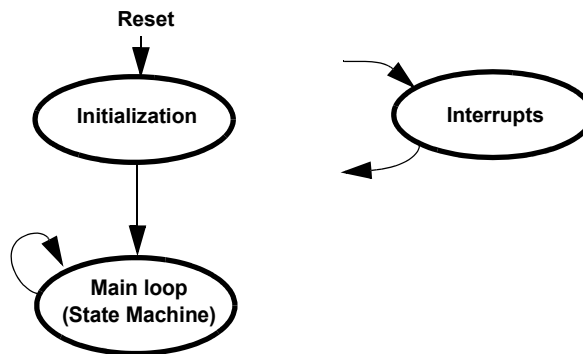


Figure 5-3. State Diagram - General Overview

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 20ms
- Initializes 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 Quad TimerA

- Initializes Quad TimerA - channels A0, A1, A2
 - Sets Count Mode to Count
 - Sets Input Source to Bus Clock / 128
 - Sets Input Polarity to Normal
 - Sets Secondary Input Source to Input 0, 1, 2
 - Sets CountFrequency to Repeatedly
 - Sets Count Length to Past Compare
 - Sets Count Direction to Up
 - Sets Capture Mode = BothEdges
 - Associates Callback On Input Edge to ISRQTimer
- Initializes Quad TimerA - channel A3
 - Sets Count Mode to Count
 - Sets Input Source to Bus Clock / 128
 - Sets Input Polarity to Normal
 - Sets Secondary Input Source to Input 2
 - Sets Count Frequency to Repeatedly
 - Sets Count Length to Past Compare
 - Sets Count Direction to Up
 - Sets Capture Mode = BothEdges
 - Associates Callback On Input Edge to CallbackOnNewEdge
 - Associates 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
- Identifies the voltage level according to the identified power stage

5.4.2 Interrupts

The interrupt handlers have the following functions:

- *Input Capture Interrupt Handler (Timer A0, A1, A2)* reads the actual status of the Hall sensors. That value is input to the commutation algorithm. The description of the commutation is in [Section 2.3.3](#) and [Section 5.3.3](#). The order of Hall sensor edges determines the direction of spin.
- *Input Capture Interrupt Handler (Timer A3)* reads the time between the two subsequent IC edges of one Hall sensor phase, which is used for speed calculation.
- *Timer Interrupt Handler* generates the time base 20ms. 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 with the increment, the *DownButton Interrupt Handler* decrements the desired speed with the decrement.
- *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 one of the following states as shown in [Figure 5-4](#), which also shows transition conditions among the drive states. The user is able to recognize the actual state by a blinking green LED diode. In

case of the *init* and *stop* state, the green LED diode blinks with a frequency of 2 Hz. In the *fault* state, the green LED diode blinks with a frequency of 8 Hz. During the *running* state the green LED diode is continuously lit.

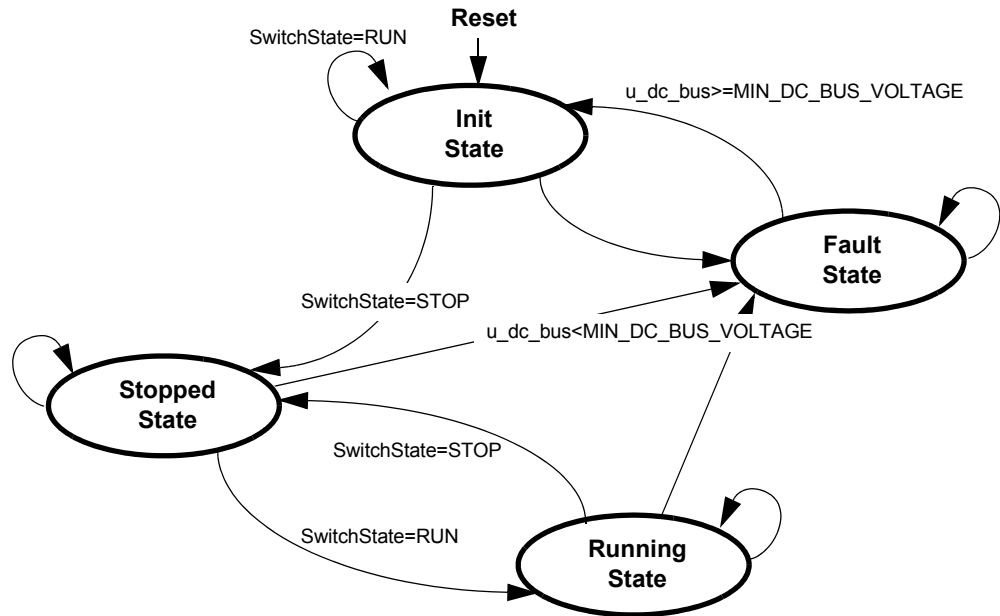


Figure 5-4. Drive State Machine Transitions

5.5 Implementation Notes

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

5.5.1 Scaling of Quantities

The BLDC 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:

$$-1.0 \leq SF \leq +1.0 \cdot 2^{-[N-1]} \tag{EQ 5-1.}$$

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 maximal range of the quantity, defined in the application [V, A, RPM, etc.]

5.5.2 DC Bus Voltage Scaling

The DC Bus voltage sense is defined by following equation:

$$u_dc_bus = \frac{V_{DC_BUS}}{V_{MAX}} \cdot 32767$$

Where:

u_dc_bus is variable of DC Bus voltage

V_{DC_BUS} is measured DC Bus voltage

V_{MAX} is max. measurable DC Bus voltage.

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

5.5.3 PI Controller Parameters

The P constant was chosen as $0.12207 (32000 * 2^{-18})$ and the constant I was chosen as $0.095367 (25000 * 2^{-18})$. The controller parameters were experimentally tuned.

5.5.4 Velocity Calculation

The constant OMEGA_ACTUAL_MECH_CONST is defined by the following equations:

position difference = 1/4 rev (given by each edge of one Hall sensors phase and two pole pairs motor)

max. period time = 0.2 s (chosen according to required min. speed)

$v_{\min} = 60 * (\text{position difference}) / (\text{max. period time}) = 75 \text{ RPM}$

$v_{\max} = 20 * v_{\min} = 1500 \text{ RPM}$ (chosen according to required max. speed)

$\text{OMEGA_ACTUAL_MECH_CONST} = 32767 * v_{\min} / v_{\max} = 1638$

Section 6. System Setup

6.1 Contents

6.2	Operational Modes	51
6.3	Application Set-Up	54
6.4	DSP56F805EVM Set-Up	56
6.5	Projects Files	58
6.6	Application Build & Execute	59

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 (LED D13, 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.

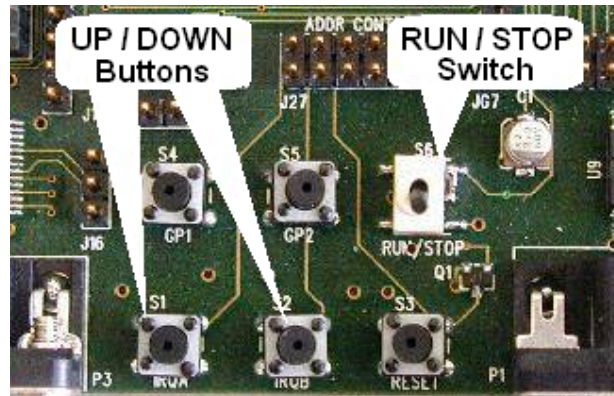


Figure 6-1. RUN/STOP Switch and UP/DOWN Buttons at DSP56F805EVM

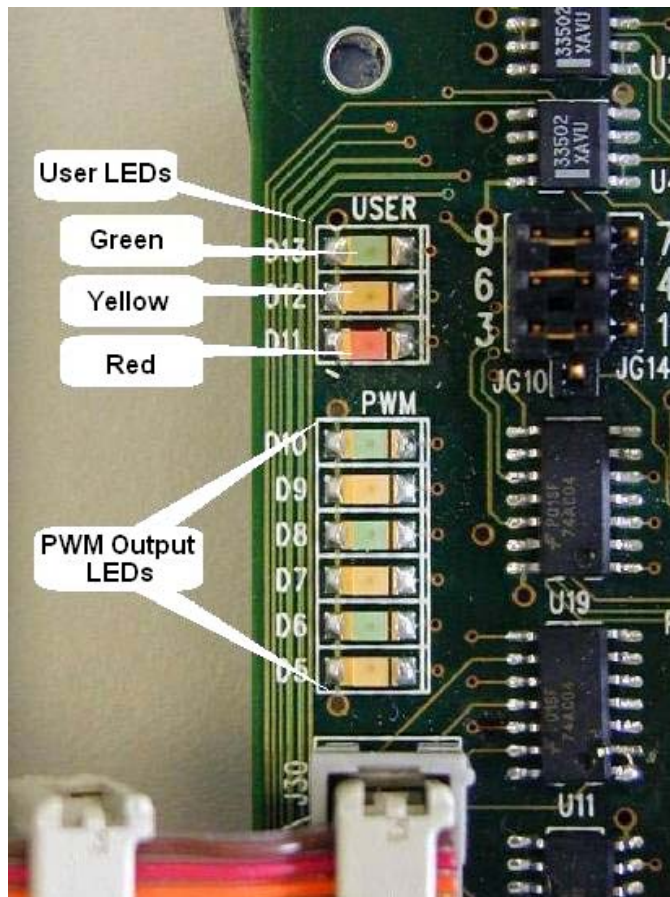


Figure 6-2. USER and PWM LEDs at DSP56F805EVM

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. PC master software enables to set the required speed of the motor.

PC master software displays the following information:

- Applied Voltage
- Required Voltage
- Speed
- RUN/STOP Switch Status
- Application Mode

Start the PC master software window's application, *bldc_hall_sensors.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 FileReload.*

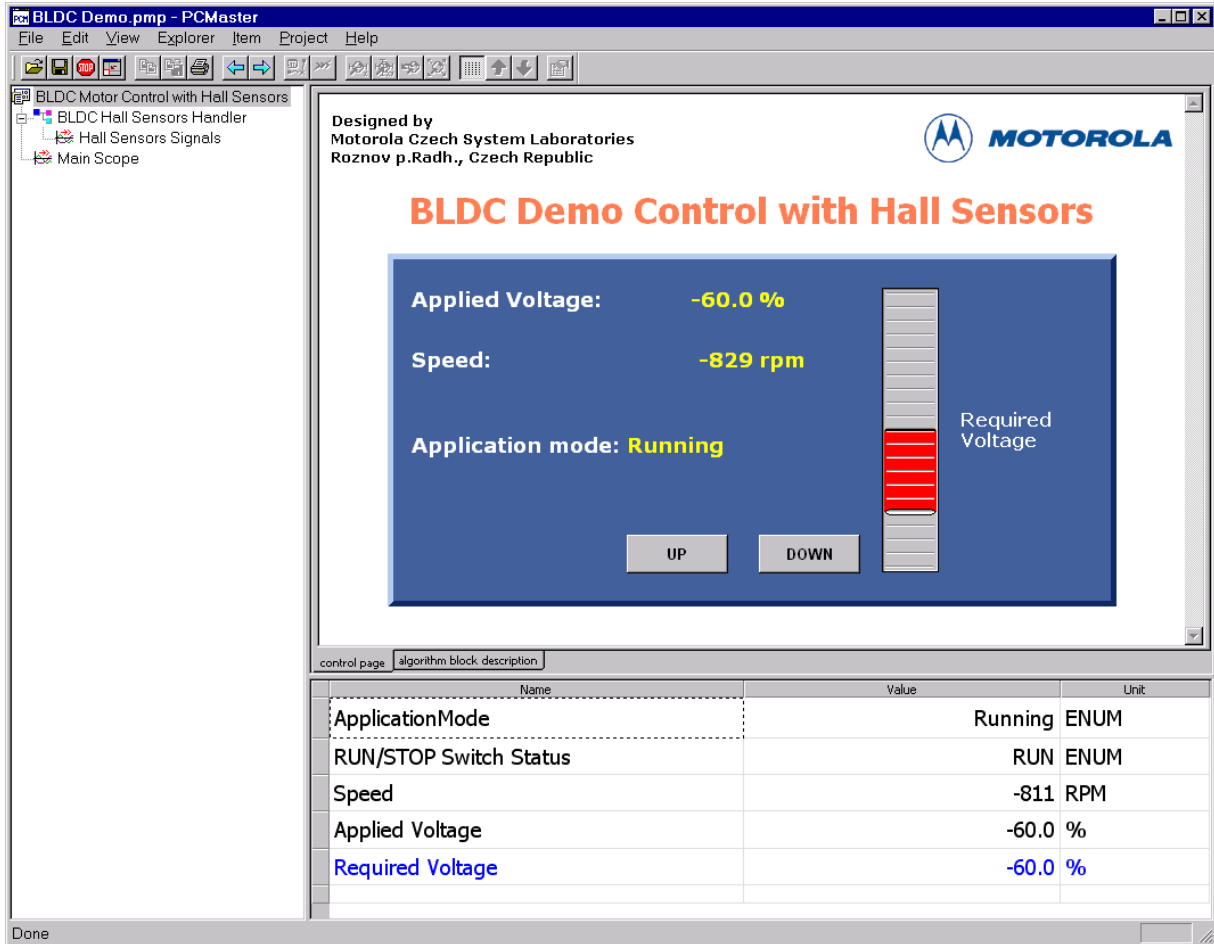


Figure 6-3. PC Master Software Control Window

6.3 Application Set-Up

Figure 6-4 illustrates the hardware set-up for the BLDC Motor Control Application with Hall Sensors.

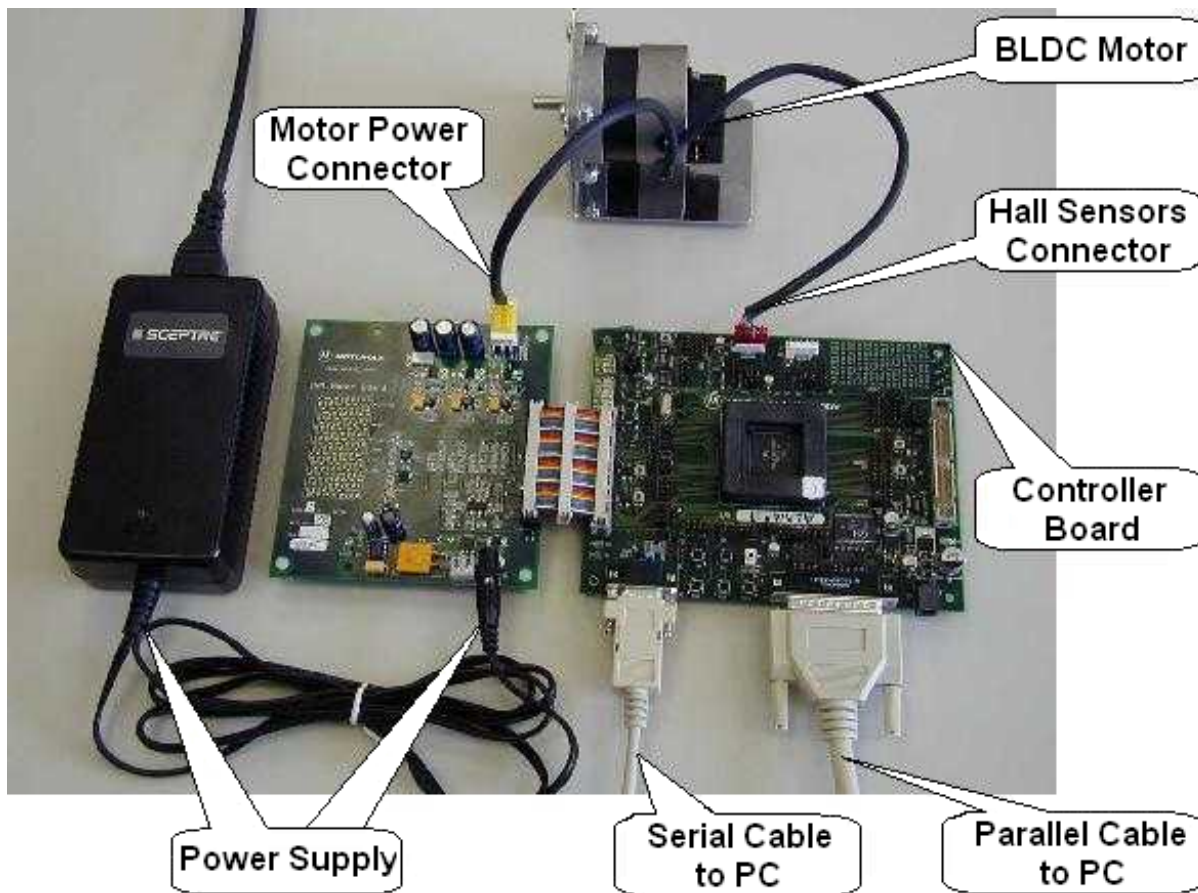


Figure 6-4. Set-up of the BLDC Motor Control Application using DSP56F805EVM

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

- 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.

6.4 DSP56F805EVM Set-Up

To execute the BLCD Motor Control with Hall Sensors, the DSP56F805EVM board requires the strap settings shown in [Figure 6-5](#) and [Table 6-2](#).

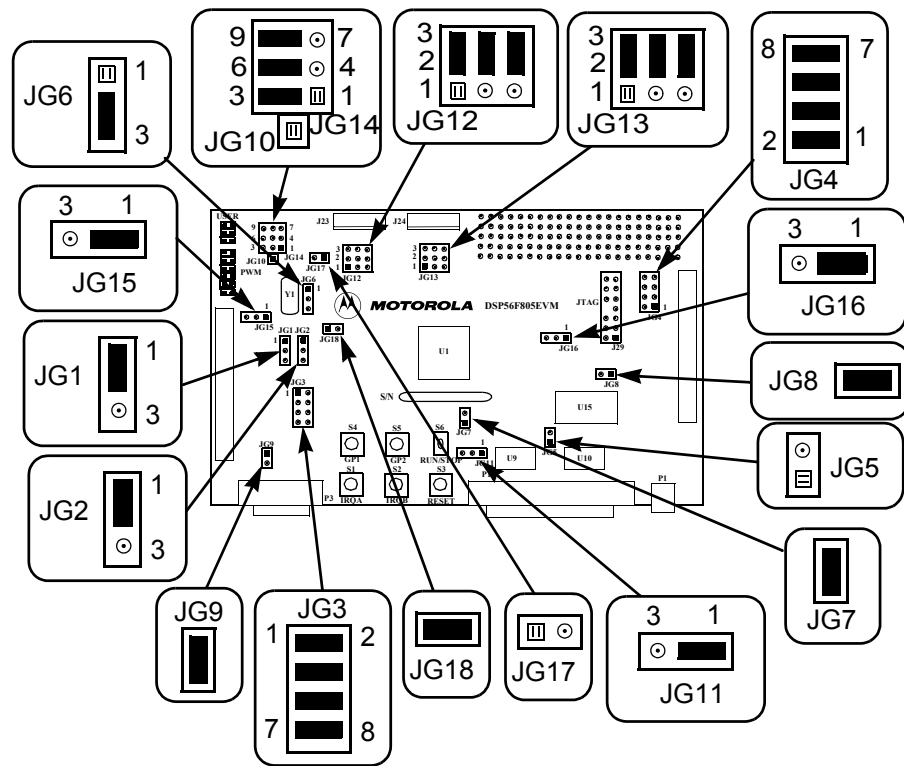


Figure 6-5. DSP56F805EVM Jumper Reference

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
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 Overcurrent selected for FAULTA1	1-2
JG16	Secondary UNI-3 Phase B Overcurrent 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.*

6.5 Projects Files

- The BLDC Motor Control application is composed of the following files:
- `...\bldc_hall_sensors_sa\bldc_hall_sensor_sa.c`, main program
- `...\bldc_hall_sensor_sa\bldc_hall_sensor_sa.mcp`, application project file
- `...\bldc_hall_sensor_sa\ApplicationConfig\appconfig.h`, application configuration file
- `...\bldc_hall_sensor_sa\SystemConfig\ExtRam\linker_ram.cmd`, linker command file for external RAM
- `...\bldc_hall_sensor_sa\SystemConfig\Flash\linker_flash.cmd`, linker command file for Flash
- `...\bldc_hall_sensor_sa\SystemConfig\Flash\flash.cfg`, configuration file for Flash
- `...\bldc_hall_sensor_sa\PCMaster\bldc_hall_sensors.pmp`, PC master software file

These files are located in the application folder.

Motor Control algorithms used in the application:

- `...\controllers.c, .h`: source and header files for PI controller

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

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

6.6 Application Build & Execute

When building the BLDC Motor Control Application with Hall Sensors, 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_hall_sensors.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.

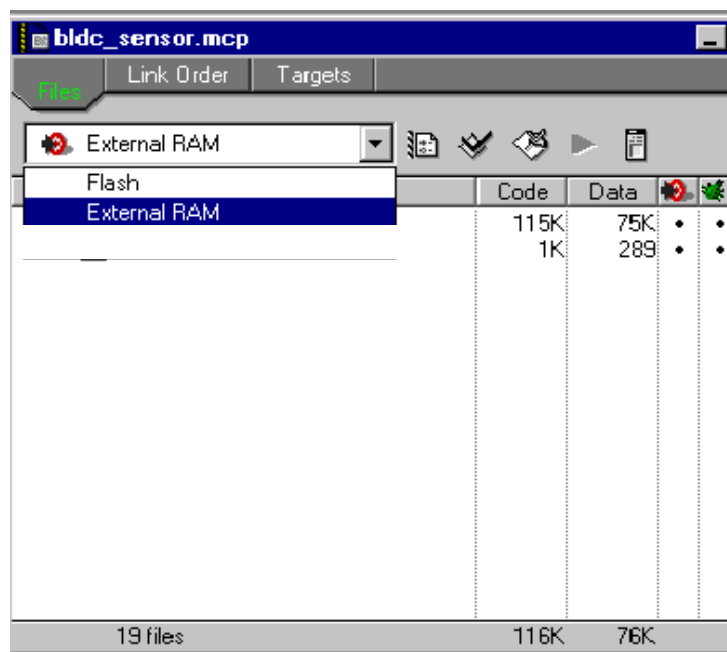


Figure 6-6. Target Build Selection

The project may now be built by executing the *Make* command, as shown in **Figure 6-7**. This will build and link the BLDC Motor Control Application with Hall Sensors and all needed Metrowerks libraries.

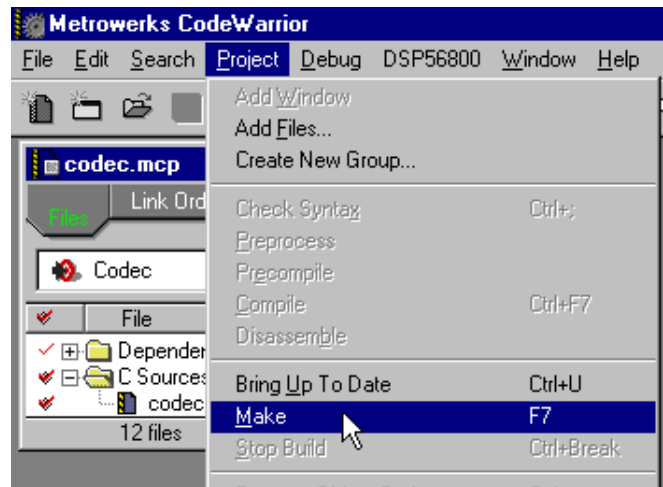


Figure 6-7. Execute *Make* Command

To execute the BLDC 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 folder:

<...>\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. **3-Phase BLDC Motor Control with Hall Sensors Using DSP56F80x**, Pavel Grasblum, AN1916/D, Motorola 2001
10. **Motorola SPS web page:** *<http://www.motorola.com/>*

Appendix B. Glossary

AC — Alternating Current

ADC — 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 DSPs dedicated for motor control.

DT — see “Dead Time (DT)”

Dead Time (DT) — A 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**

Freescale Semiconductor, Inc.

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

DRM025/D

**For More Information On This Product,
Go to: www.freescale.com**