
Sensorless Field Oriented Control of PMSM Motors

*Author: Jorge Zambada
Microchip Technology Inc.*

INTRODUCTION

Designers can expect environmental demands to continue to drive the need for advanced motor control techniques that produce energy efficient air conditioners, washing machines and other home appliances. Until now, sophisticated motor control solutions have only been available from proprietary sources. However, the implementation of advanced, cost-effective motor control algorithms is now a reality thanks to the new generation of Digital Signal Controllers (DSCs).

An air conditioner, for example, requires fast response for speed changes in the motor. Advanced motor control algorithms are needed to produce quieter units that are more energy efficient. Field Oriented Control (FOC) has emerged as the leading method to achieve these environmental demands.

This application note discusses implementation of a sensorless FOC algorithm for Permanent Magnet Synchronous Motors (PMSMs) using the Microchip dsPIC[®] DSC family.

Why Use the FOC Algorithm?

The traditional control method for BLDC motors drives the stator in a six-step process, which generates oscillations on the produced torque. In six-step control, a pair of windings is energized until the rotor reaches the next position, then the motor is commutated to the next step. Hall sensors determine the rotor position to electronically commutate the motor. Advanced sensorless algorithms use the back-EMF generated in the stator winding to determine the rotor position.

The dynamic response of six-step control (also called trapezoidal control) is not suitable for washing machines because the load is changing dynamically within a wash cycle, and varies with different loads and selected wash cycle. Further, in a front load washing machine, the gravitational power works against the motor load when the load is on the top side of the drum. Only advanced algorithms such as FOC can handle these dynamic load changes.

This application note focuses on the PMSM-based sensorless FOC control of appliances because this control technique offers the greatest cost benefit in appliance motor control. The sensorless FOC technique also overcomes restrictions placed on some applications that cannot deploy position or speed sensors because the motor is flooded, or because of wire harness placement constraints. With a constant rotor magnetic field produced by a permanent magnet on the rotor, the PMSM is very efficient when used in an appliance. In addition, its stator magnetic field is generated by sinusoidal distribution of windings. When compared to induction motors, PMSM motors are powerful for their size. They are also electrically less noisy than DC motors, since brushes are not used.

Why Use Digital Signal Controllers for Motor Control?

dsPIC DSCs are suitable for appliances like washing machines and air conditioner compressors because they incorporate peripherals that are ideally suited for motor control, such as:

- Pulse-Width Modulation (PWM)
- Analog-to-Digital Converters (ADCs)
- Quadrature Encoder Interface (QEI)

When performing controller routines and implementing digital filters, dsPIC DSCs enable designers to optimize code because MAC instructions and fractional operations can be executed in a single cycle. Also, for operations that require saturation capabilities, the dsPIC DSCs help avoid overflows by offering hardware saturation protection.

DSCs need fast and flexible analog-to-digital (A/D) conversion for current sensing—a crucial function in motor control. The dsPIC DSCs feature ADCs that can convert input samples at 1 Msp/s rates, and handle up to four inputs simultaneously. Multiple trigger options on the ADCs enable use of inexpensive current sense resistors to measure winding currents. For example, the ability to trigger A/D conversions with the PWM module allows inexpensive current sensing circuitry to sense inputs at specific times (switching transistors allow current to flow through sense resistors).

MOTOR CONTROL WITH DIGITAL SIGNAL CONTROLLERS

The dsPIC30F Motor Control family is specifically designed to control the most popular types of motors, including:

- AC Induction Motors (ACIM)
- Brushed DC Motors (BDC)
- Brushless DC Motors (BLDC)
- Permanent Magnet Synchronous Motors (PMSM)

Several application notes have been published based on the dsPIC30F motor control family (see “**References**” Section). These application notes are available on the Microchip web site (www.microchip.com).

This application note demonstrates how the dsPIC30F6010A takes advantage of peripherals specifically suited for motor control (motor control PWM and high-speed ADC) to execute sensorless field oriented control of PMSM. The DSP engine of the dsPIC30F6010A supports the necessary fast mathematical operations.

Data Monitoring and Control Interface

The Data Monitor and Control Interface (DMCI) provides quick dynamic integration with MPLAB® IDE for projects in which operational constraints of the application depend on variable control of range values, on/off states or discrete values. If needed, application feedback can be represented graphically. Examples include motor control and audio processing applications.

The DMCI provides:

- 9 slider controls and 9 boolean (on/off) controls shown in Figure 1
- 35 input controls (see Figure 2)
- 4 graphs (see Figure 3)

The interface provides project aware navigation of program symbols (variables) that can be dynamically assigned to any combination of slider, direct input or boolean controls. The controls can then be used interactively to change values of program variables within MPLAB IDE. The graphs can be dynamically configured for viewing program generated data.

<p>Note: The characteristics of the DMCI tool are subject to change. This description of the DMCI tool is accurate at the date of publication.</p>

Application Highlights

The purpose of this application note is to illustrate a software-based implementation of sensorless, field oriented control for permanent magnet synchronous motors using Microchip digital signal controllers.

The control software offers these features:

- Implements vector control of a Permanent Magnet Synchronous Motor.
- Position and speed estimation algorithm eliminates the need for position sensors.
- An air conditioner compressor rated at 1.5 kW is targeted as the main motor.
- The speed range is from 500 to 7300 RPM.
- With a 50 μ sec control loop period, the software requires approximately 21 MIPS of CPU overhead (about 2/3 of the total available CPU).
- The application requires 450 bytes of data memory storage. With the user interface, approximately 6 Kbytes of program memory are required. The memory requirements of the application allow it to run on the dsPIC30F2010, which is the smallest and least expensive dsPIC30F device at the time of this writing.
- An optional diagnostics mode can be enabled to allow real-time observation of internal program variables on an oscilloscope. This feature facilitates control loop adjustment.
- This application note applies to all motor control devices in the dsPIC30F and dsPIC33F families.

FIGURE 1: DYNAMIC DATA CONTROL INTERFACE

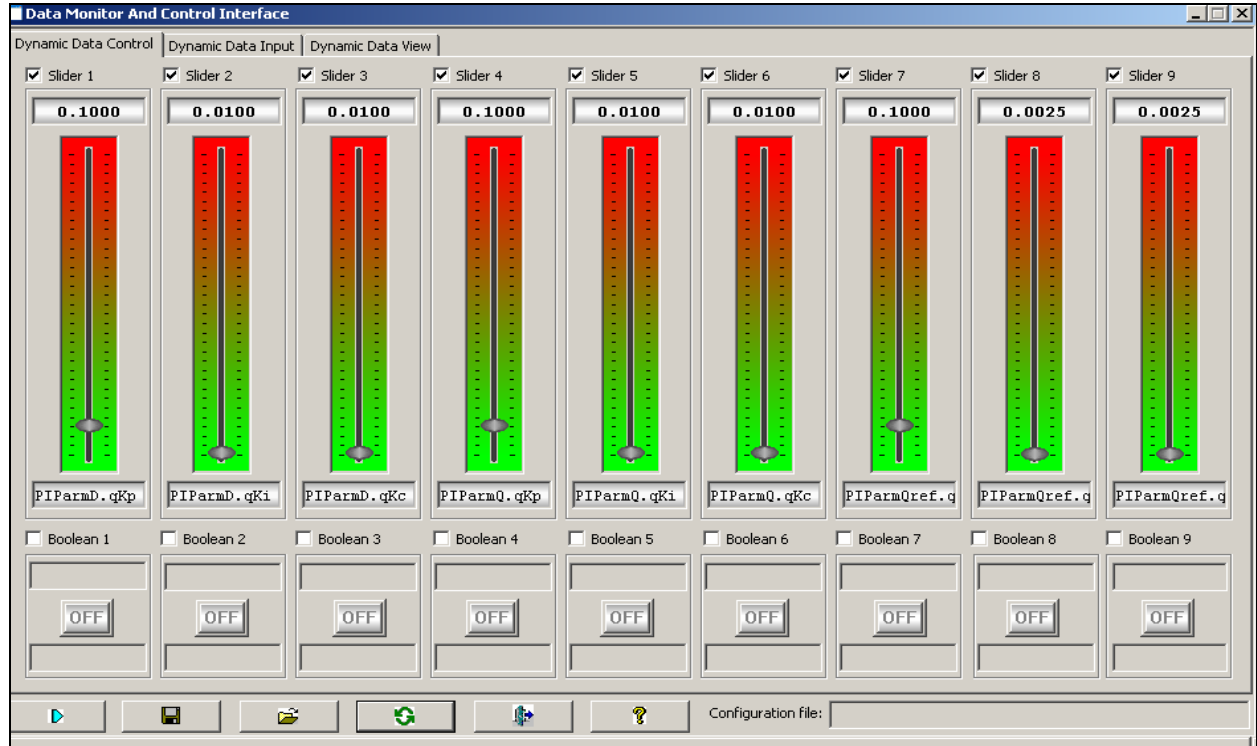


FIGURE 2: USER-DEFINED DATA INPUT CONTROLS

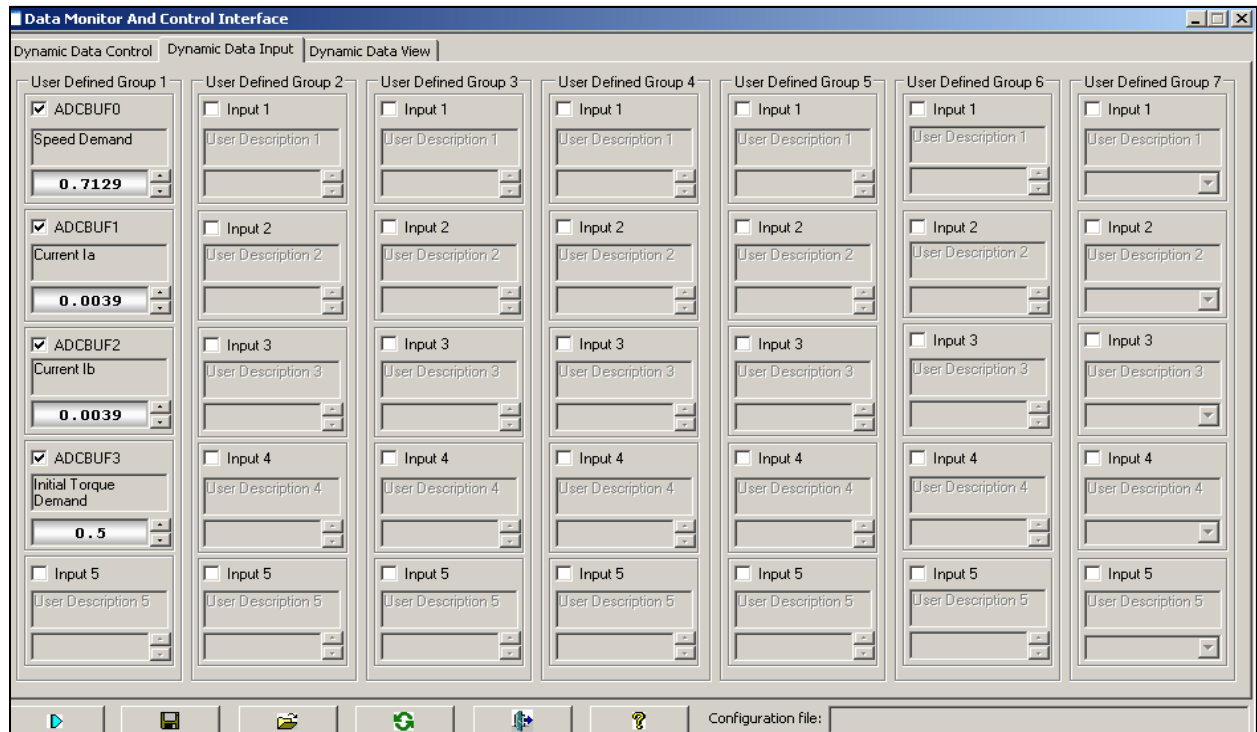
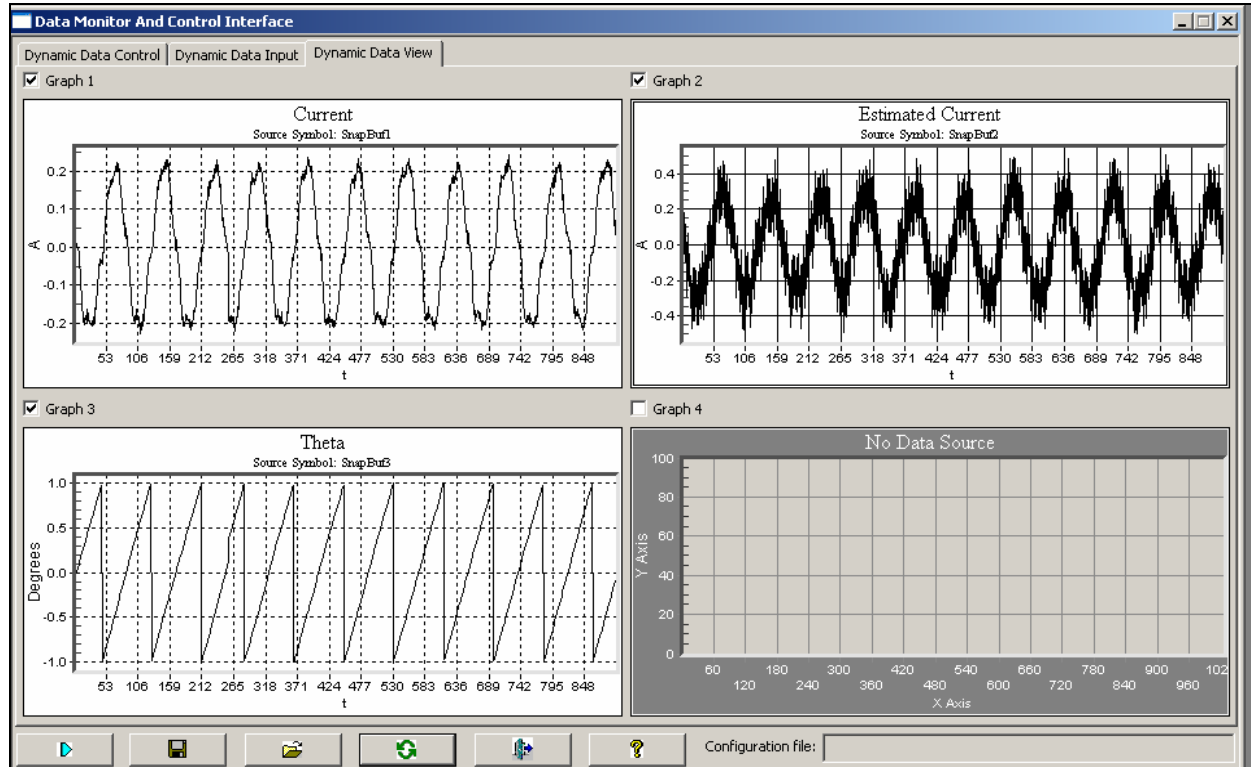


FIGURE 3: GRAPHICAL DATA VIEW



SYSTEM OVERVIEW

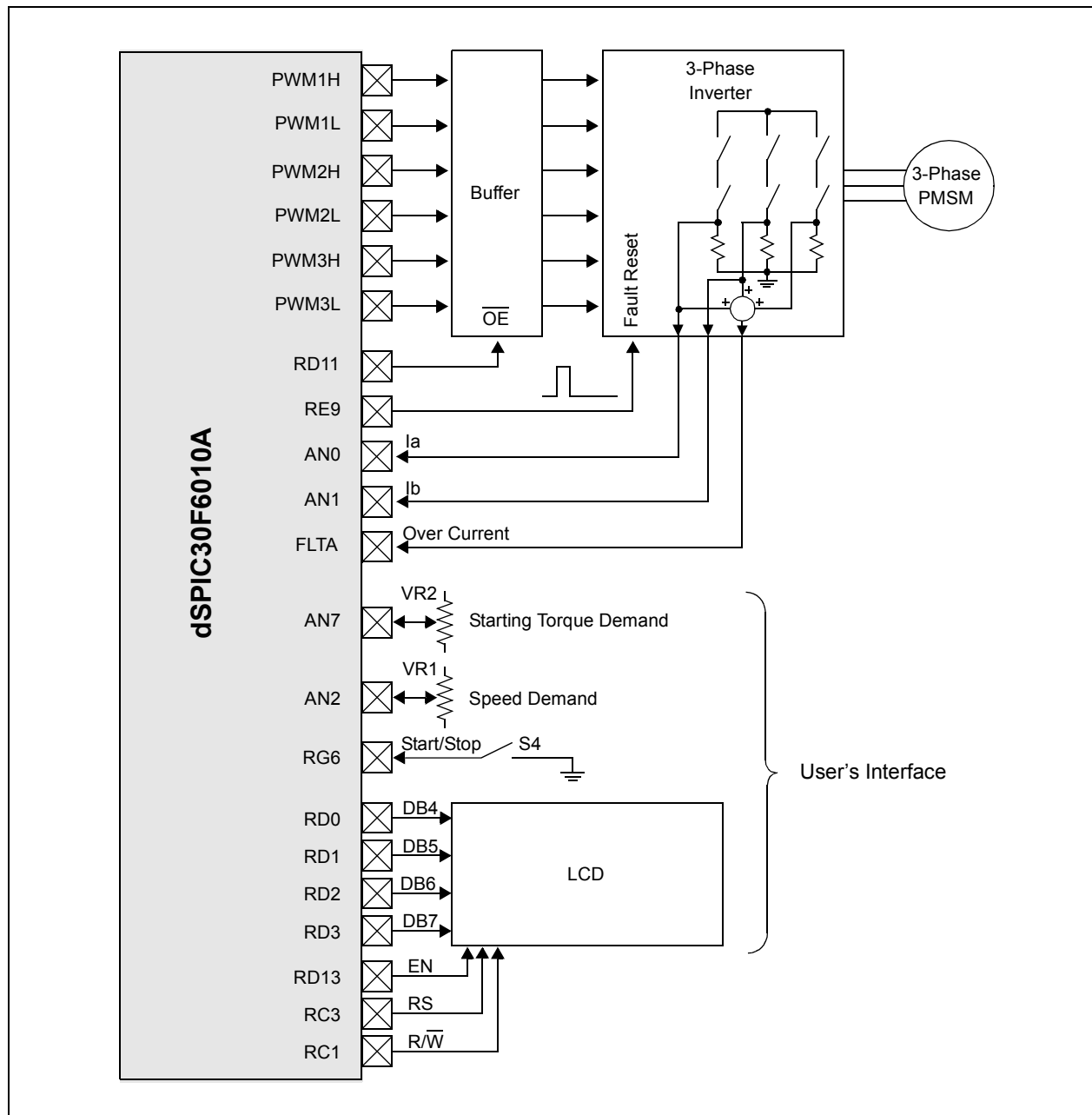
As shown in Figure 4, there are no position sensors attached to the motor shaft. Instead, low inductance resistors, which are part of the inverter, are used for current measurements on the motor. A 3-phase inverter is used as the power stage to drive motor windings. Current sensing and fault generation circuitry built into the power inverter protects the overall system against over currents.

Figure 5 shows how the 3-phase topology, as well as the current detection and fault generation circuitry, are implemented.

The first transistor shown on the left side of the inverter is used for Power Factor Correction (PFC), which is not part of this application note.

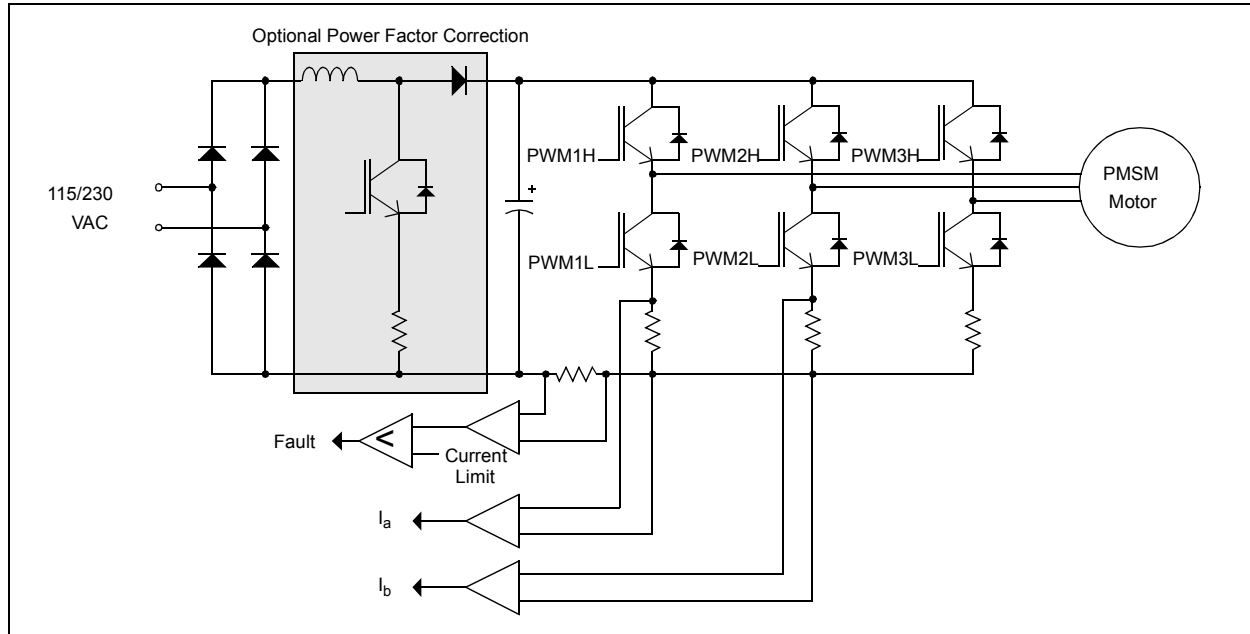
Hardware referenced in this application note is the dsPICDEM™ MC1 Motor Control Development Board (DM300020), available at the Microchip website (www.microchip.com). Two power modules are available: one for low voltages (up to 50V) and one for high voltages (up to 400V). A modified High-voltage Power Module (DM3000021) was used for this application note. These modules are explained in the appendix.

FIGURE 4: SYSTEM OVERVIEW



AN1078

FIGURE 5: 3-PHASE TOPOLOGY



FIELD ORIENTED CONTROL

A Matter of Perspective

One way to understand how FOC (sometimes referred to as vector control) works is to form a mental image of the coordinate reference transformation process. If you picture an AC motor operation from the perspective of the stator, you see a sinusoidal input current applied to the stator. This time variant signal generates a rotating magnetic flux. The speed of the rotor is a function of the rotating flux vector. From a stationary perspective, the stator currents and the rotating flux vector look like AC quantities.

Now, imagine being inside the motor and running alongside the spinning rotor at the same speed as the rotating flux vector generated by the stator currents. If you were to look at the motor from this perspective during steady state conditions, the stator currents look like constant values, and the rotating flux vector is stationary.

Ultimately, you want to control the stator currents to obtain the desired rotor currents (which cannot be measured directly). With coordinate reference transformation, the stator currents can be controlled like DC values using standard control loops.

Vector Control Summary

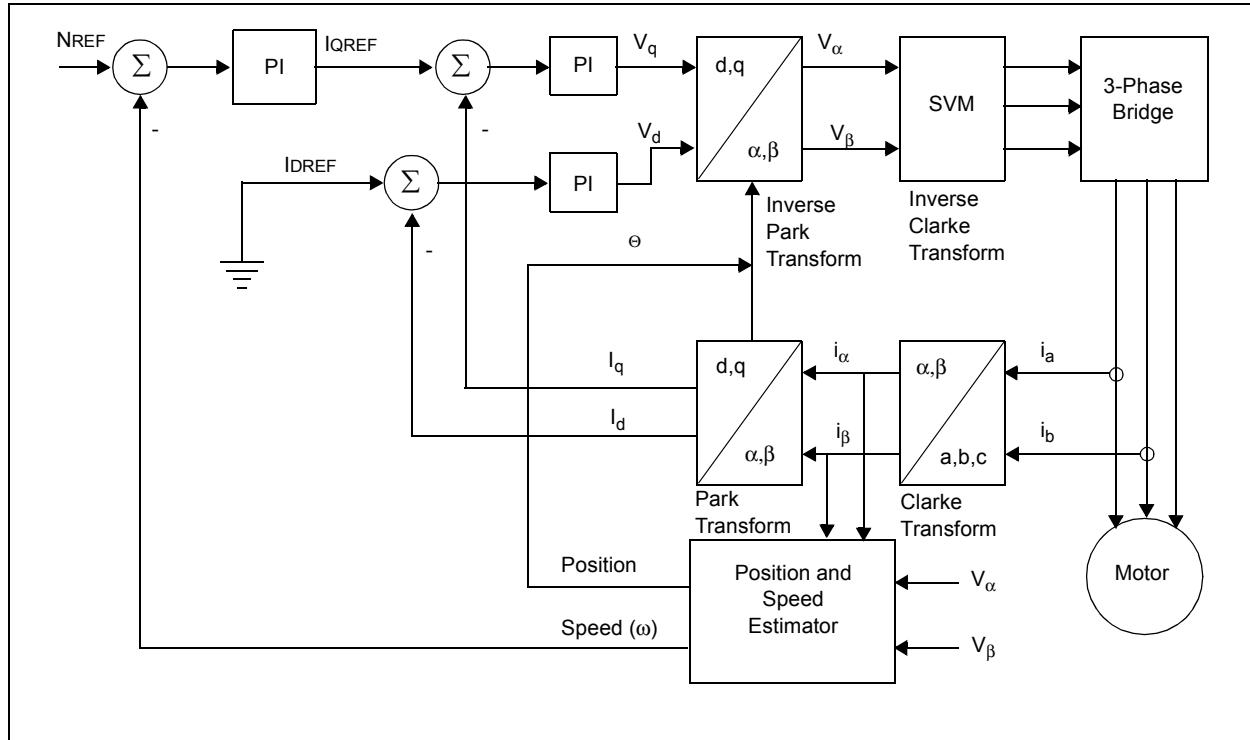
Indirect vector control can be summarized as follows:

1. The 3-phase stator currents are measured. These measurements provide values i_a and i_b . i_c is calculated because i_a , i_b and i_c have this relationship:
$$i_a + i_b + i_c = 0.$$
2. The 3-phase currents are converted to a two axis system. This conversion provides the variables i_α and i_β from the measured i_a and i_b and the calculated i_c values. i_α and i_β are time-varying quadrature current values as viewed from the perspective of the stator.
3. The two axis coordinate system is rotated to align with the rotor flux using a transformation angle calculated at the last iteration of the control loop. This conversion provides the i_d and i_q variables from i_α and i_β . i_d and i_q are the quadrature currents transformed to the rotating coordinate system. For steady state conditions, i_d and i_q are constant.
4. Error signals are formed using i_d , i_q and reference values for each.
 - The i_d reference controls rotor magnetizing flux.
 - The i_q reference controls the torque output of the motor.
 - The error signals are input to PI controllers.
 - The output of the controllers provide v_d and v_q , which is a voltage vector that will be sent to the motor.
5. A new transformation angle is estimated where v_α , v_β , i_α and i_β are the inputs. The new angle guides the FOC algorithm as to where to place the next voltage vector.
6. The v_d and v_q output values from the PI controllers are rotated back to the stationary reference frame using the new angle. This calculation provides the next quadrature voltage values v_α and v_β .
7. The v_α and v_β values are transformed back to 3-phase values v_a , v_b and v_c . The 3-phase voltage values are used to calculate new PWM duty cycle values that generate the desired voltage vector. The entire process of transforming, PI iteration, transforming back and generating PWM is illustrated in Figure 6.

The next sections of this application note describe these steps in greater detail.

AN1078

FIGURE 6: VECTOR CONTROL BLOCK DIAGRAM



COORDINATE TRANSFORMS

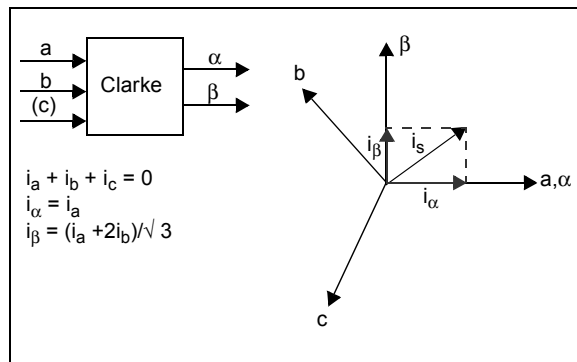
Through a series of coordinate transforms, you can indirectly determine and control the time invariant values of torque and flux with classic PI control loops. The process begins by measuring the 3-phase motor currents. In practice, the instantaneous sum of the three current values is zero. Thus by measuring only two of the three currents, you can determine the third. Because of this fact, hardware cost can be reduced by the expense of the third current sensor.

A single shunt implementation for 3-phase current measurement is also possible with the dsPIC DSC. Contact Microchip for more information.

Clarke Transform

The first coordinate transform, called the Clarke Transform, moves a three axis, two dimensional coordinate system, referenced to the stator, onto a two axis system, keeping the same reference (see Figure 7, where i_a , i_b and i_c are the individual phase currents).

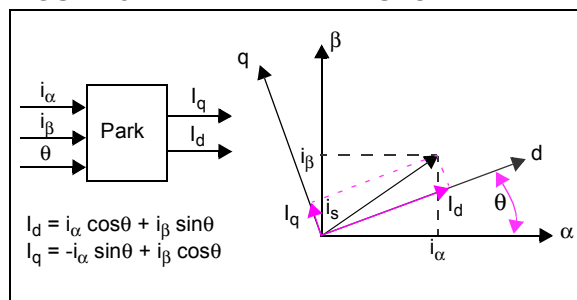
FIGURE 7: CLARKE TRANSFORM



Park Transforms

At this point, you have the stator current represented on a two axis orthogonal system with the axis called α - β . The next step is to transform into another two axis system that is rotating with the rotor flux. This transformation uses the Park Transform, as illustrated in Figure 8. This two axis rotating coordinate system is called the d-q axis. θ represents the rotor angle.

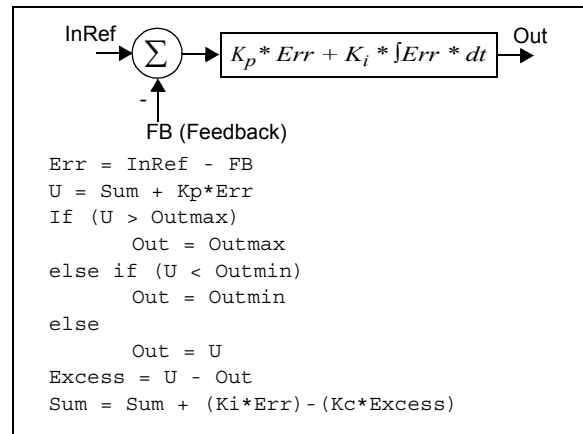
FIGURE 8: PARK TRANSFORM



PI Control

Three PI loops are used to control three interactive variables independently. The rotor speed, rotor flux and rotor torque are each controlled by a separate PI module. The implementation is conventional and includes term ($K_c \cdot \text{Excess}$) to limit integral windup, as illustrated in Figure 9. Excess is calculated by subtracting the unlimited output (U) and limited output (Out). The term K_c multiplies the Excess, and limits the accumulated integral portion (Sum).

FIGURE 9: PI CONTROL



PID CONTROLLER BACKGROUND

A complete discussion of Proportional Integral Derivative (PID) controllers is beyond the scope of this application note, but this section provides you with some basics of PID operation.

A PID controller responds to an error signal in a closed control loop and attempts to adjust the controlled quantity to achieve the desired system response. The controlled parameter can be any measurable system quantity such as speed, torque or flux. The benefit of the PID controller is that it can be adjusted empirically by varying one or more gain values and observing the change in system response.

A digital PID controller is executed at a periodic sampling interval. It is assumed that the controller is executed frequently enough so that the system can be properly controlled. The error signal is formed by subtracting the desired setting of the parameter to be controlled from the actual measured value of that parameter. The sign of the error indicates the direction of change required by the control input.

The Proportional (P) term of the controller is formed by multiplying the error signal by a P gain, causing the PID controller to produce a control response that is a function of the error magnitude. As the error signal becomes larger, the P term of the controller becomes larger to provide more correction.

The effect of the P term tends to reduce the overall error as time elapses. However, the effect of the P term diminishes as the error approaches zero. In most systems, the error of the controlled parameter gets very close to zero but does not converge. The result is a small remaining steady state error.

The Integral (I) term of the controller is used to eliminate small steady state errors. The I term calculates a continuous running total of the error signal. Therefore, a small steady state error accumulates into a large error value over time. This accumulated error signal is multiplied by an I gain factor and becomes the I output term of the PID controller.

The Differential (D) term of the PID controller is used to enhance the speed of the controller and responds to the rate of change of the error signal. The D term input is calculated by subtracting the present error value from a prior value. This delta error value is multiplied by a D gain factor that becomes the D output term of the PID controller.

The D term of the controller produces more control output as the system error changes more rapidly. Not all PID controllers will implement the D or, less commonly, the I terms. For example, this application does not use D terms due to the relatively slow response time of motor speed changes. In this case, the D term could cause excessive changes in PWM duty cycle that could affect the operation of the algorithms and produce over current trips.

Adjusting the PID Gains

The P gain of a PID controller sets the overall system response. When you first tune a controller, set the I and D gains to zero. You can then increase the P gain until the system responds well to set point changes without excessive overshoot or oscillations. Using lower values of P gain will 'loosely' control the system, while higher values will give 'tighter' control. At this point, the system will probably not converge to the set point.

After you select a reasonable P gain, you can slowly increase the I gain to force the system error to zero. Only a small amount of I gain is required in most systems. The effect of the I gain, if large enough, can overcome the action of the P term, slow the overall control response and cause the system to oscillate around the set point. If oscillation occurs, reducing the I gain and increasing the P gain will usually solve the problem.

This application includes a term to limit integral windup, which occurs if the integrated error saturates the output parameter. Any further increase in the integrated error does not affect the output. The accumulated error, when it does decrease, will have to fall (or unwind) to below the value that caused the output to saturate. The Kc coefficient limits this unwanted accumulation. For most situations, this coefficient can be set equal to Ki.

All three controllers have a maximum value for the output parameter. These values can be found in the `UserParms.h` file and are set by default to avoid saturation in the `SVGen()` routine.

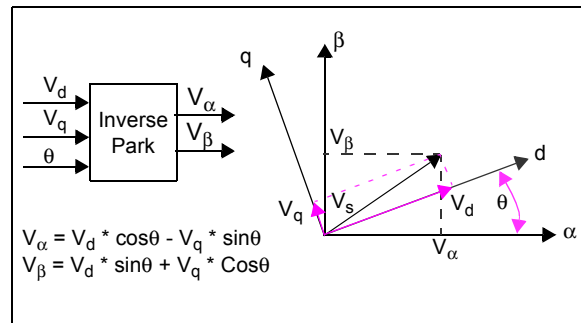
Control Loop Dependencies

There are three interdependent PI control loops in this application. The outer loop controls the motor velocity. The two inner loops control the transformed motor currents, I_d and I_q . As mentioned previously, the I_d loop is responsible for controlling flux, and the I_q value is responsible for controlling the motor torque.

Inverse Park

After the PI iteration you have two voltage component vectors in the rotating d-q axis. You will need to go through complementary inverse transforms to get back to the 3-phase motor voltage. First you transform from the two axis rotating d-q frame to the two axis stationary frame α - β . This transformation uses the Inverse Park Transform, as illustrated in Figure 10.

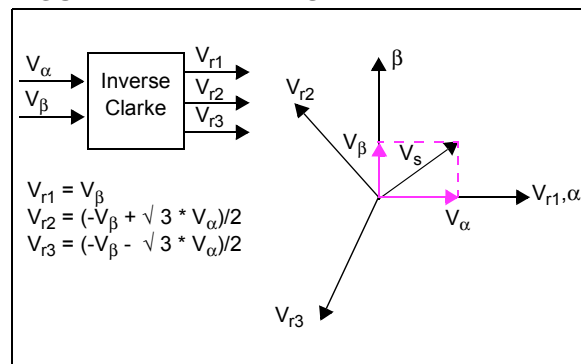
FIGURE 10: INVERSE PARK



Inverse Clarke

The next step is to transform from the stationary two axis α - β frame to the stationary three axis, 3-phase reference frame of the stator. Mathematically, this transformation is accomplished with the Inverse Clark Transform, as illustrated in Figure 11.

FIGURE 11: INVERSE CLARKE



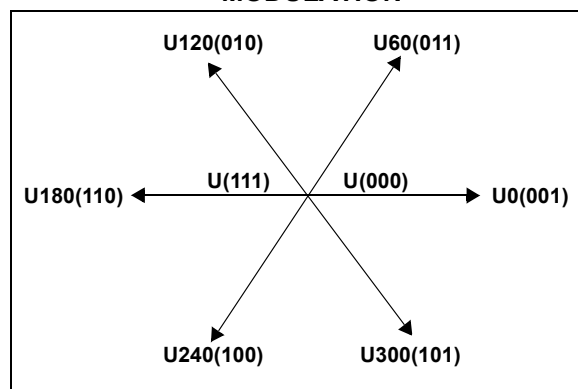
Space Vector Modulation (SVM)

The final step in the vector control process is to generate pulse-width modulation signals for the 3-phase motor voltage signals. If you use Space Vector Modulation (SVM) techniques, the process of generating the pulse width for each of the three phases is reduced to a few simple equations. In this implementation, the Inverse Clarke Transform has been folded into the SVM routine, which further simplifies the calculations.

Each of the three inverter outputs can be in one of two states. The inverter output can be connected to either the + bus rail or the - bus rail, which allows for $2^3 = 8$ possible states that the output can be in (see Table 1).

The two states in which all three outputs are connected to either the + bus or the - bus are considered null states because there is no line-to-line voltage across any of the phases. These are plotted at the origin of the SVM Star. The remaining six states are represented as vectors with 60 degree rotation between each state, as shown in Figure 12.

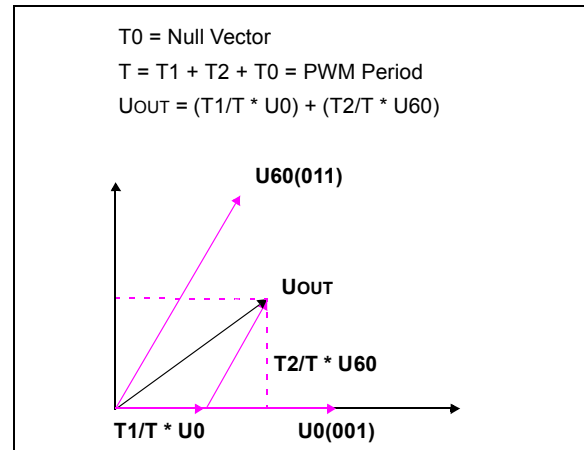
FIGURE 12: SPACE VECTOR MODULATION



The process of Space Vector Modulation allows the representation of any resultant vector by the sum of the components of the two adjacent vectors. In Figure 13, U_{OUT} is the desired resultant. It lies in the sector

between U_{60} and U_0 . If during a given PWM period T , U_0 is output for T_1/T and U_{60} is output for T_2/T , the average for the period will be U_{OUT} .

FIGURE 13: AVERAGE SPACE VECTOR MODULATION



T_0 represents a time where no effective voltage is applied into the windings; that is, where a null vector is applied. The values for T_1 and T_2 can be extracted with no extra calculations by using a modified Inverse Clark transformation. If you reverse V_α and V_β , a reference axis is generated that is shifted by 30 degrees from the SVM star. As a result, for each of the six segments, one axis is exactly opposite that segment and the other two axes symmetrically bound the segment. The values of the vector components along those two bounding axes are equal to T_1 and T_2 . See the `CalcRef.s` and `SVGen.s` files in the source code for details of the calculations.

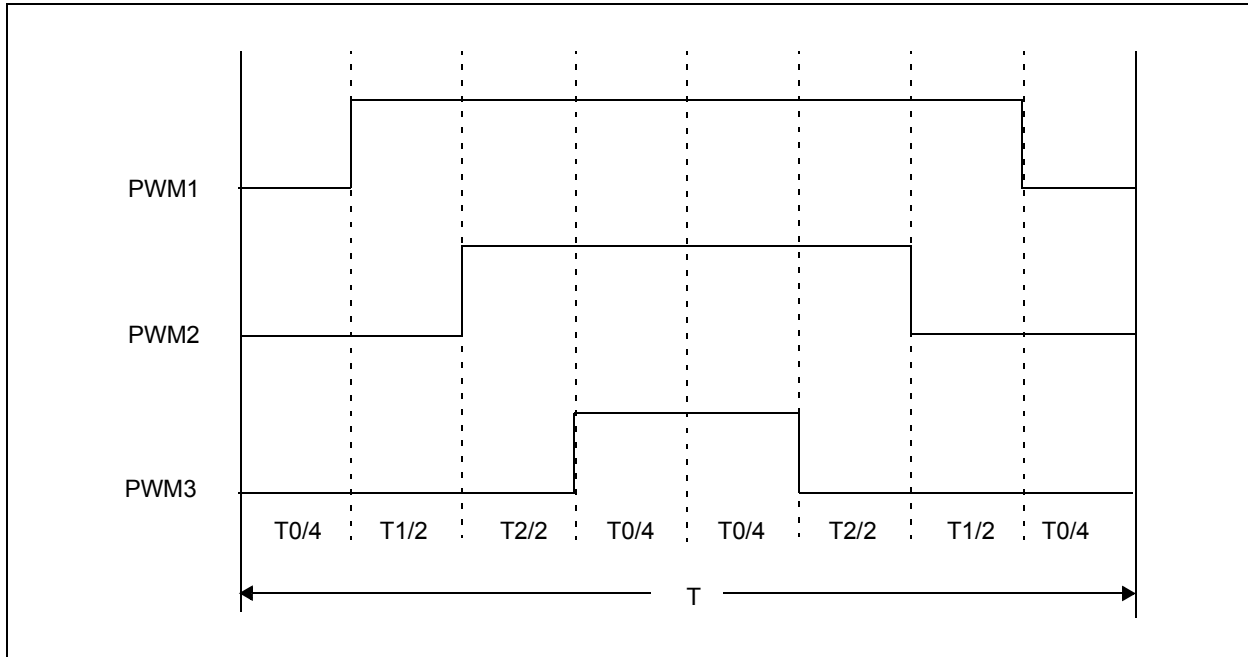
You can see from Figure 14 that for the PWM period T , the vector T_1 is output for T_1/T and the vector T_2 is output for T_2/T . During the remaining time the null vectors are output. The dsPIC DSC is configured for center-aligned PWM, which forces symmetry about the center of the period. This configuration produces two pulses line-to-line during each period. The effective switching frequency is doubled, reducing the ripple current while not increasing the switching losses in the power devices.

TABLE 1: SPACE VECTOR MODULATION INVERTER STATES

Phase C	Phase B	Phase A	V_{ab}	V_{bc}	V_{ca}	V_{ds}	V_{qs}	Vector
0	0	0	0	0	0	0	0	$U(000)$
0	0	1	V_{DC}	0	$-V_{DC}$	$2/3V_{DC}$	0	U_0
0	1	1	0	V_{DC}	$-V_{DC}$	$V_{DC}/3$	$V_{DC}/3$	U_{60}
0	1	0	$-V_{DC}$	V_{DC}	0	$-V_{DC}/3$	$V_{DC}/3$	U_{120}
1	1	0	$-V_{DC}$	0	V_{DC}	$-2V_{DC}/3$	0	U_{180}
1	0	0	0	$-V_{DC}$	V_{DC}	$-V_{DC}/3$	$-V_{DC}/3$	U_{240}
1	0	1	V_{DC}	$-V_{DC}$	0	$V_{DC}/3$	$-V_{DC}/3$	U_{300}
1	1	1	0	0	0	0	0	$U(111)$

AN1078

FIGURE 14: PWM FOR PERIOD T



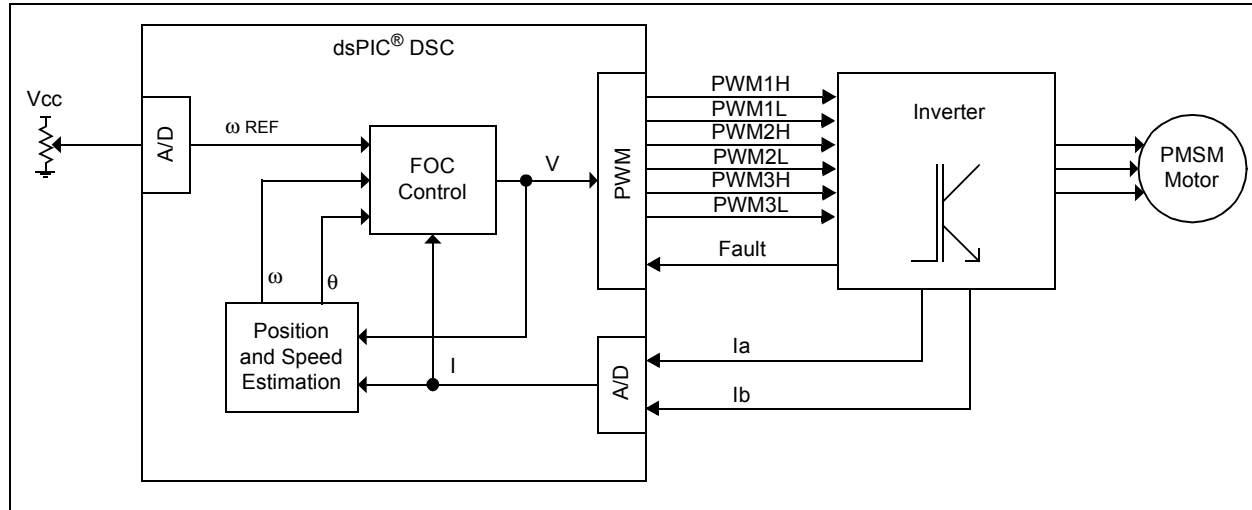
SENSORLESS FIELD ORIENTED CONTROL FOR PMSM MOTORS

An important piece of the algorithm is how to calculate the commutation angle needed for FOC. This section of the application note explains the process of estimating commutation angle (θ) and motor speed (ω).

The sensorless control technique implements the FOC algorithm by estimating the position of the motor without using position sensors. Figure 15 is a simplified block diagram of the position estimator function.

Motor position and speed are estimated based on measured currents and calculated voltages.

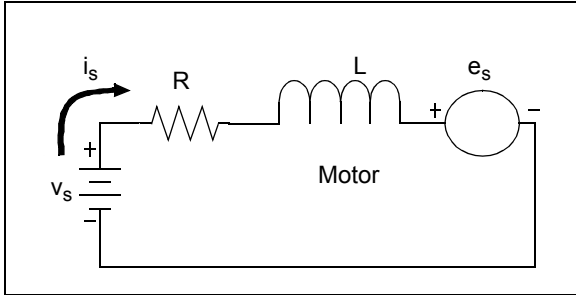
FIGURE 15: POSITION ESTIMATOR FUNCTION BLOCK DIAGRAM



Motor Model

You can estimate the PMSM Motor position by using a model of a DC Motor, which can be represented by winding resistance, winding inductance and back-EMF, as shown in Figure 16.

FIGURE 16: MOTOR MODEL



From the motor model, the input voltage can be obtained by Equation 1.

EQUATION 1: DIGITIZED MOTOR MODEL

$$v_s = Ri_s + L \frac{d}{dt}i_s + e_s$$

Where:

- i_s = Motor Current Vector
- v_s = Input Voltage Vector
- e_s = Back-EMF Vector
- R = Winding Resistance
- L = Winding Inductance
- T_s = Control Period

Motor current is obtained by solving for i_s :

$$\frac{d}{dt}i_s = \left(-\frac{R}{L}\right)i_s + \frac{1}{L}(v_s - e_s)$$

In the digital domain, this equation becomes:

$$\frac{i_s(n+1) - i_s(n)}{T_s} = \left(-\frac{R}{L}\right)i_s(n) + \frac{1}{L}(v_s(n) - e_s(n))$$

Solving for i_s :

$$i_s(n+1) = \left(1 - T_s \cdot \frac{R}{L}\right)i_s(n) + \frac{T_s}{L}(v_s(n) - e_s(n))$$

or

$$i_s(n+1) = F \cdot i_s(n) + G \cdot (v_s(n) - e_s(n))$$

where

$$F = 1 - T_s \cdot \frac{R}{L}$$

$$G = \frac{T_s}{L}$$

Calculating F and G Parameters

The motor model has two parameters that need to be modified for a particular motor. These two parameters are F and G gains, where:

$$F = 1 - T_s \cdot \frac{R}{L}$$

$$G = \frac{T_s}{L}$$

Constants R and L are measured using a simple multi-meter. For example, if a line to line resistance is measured, the R used for F and G gains is the measurement divided by two, since the phase resistance is needed. The same procedure applies to inductance calculation L. For example, if a new motor is run with this algorithm at 8 kHz control frequency, where the line to line resistance measured is 5.0Ω, and line to line inductance is 10 mH, then the motor model parameters are:

$$F = 1 - T_s \cdot \frac{R}{L} = 1 - (1/8 \text{ kHz}) \cdot \frac{(5.0\Omega/2)}{(10 \text{ mH}/2)} = 0.9375$$

$$G = \frac{T_s}{L} = \frac{(1/8 \text{ kHz})}{(10 \text{ mH}/2)} = 0.025$$

Current Observer

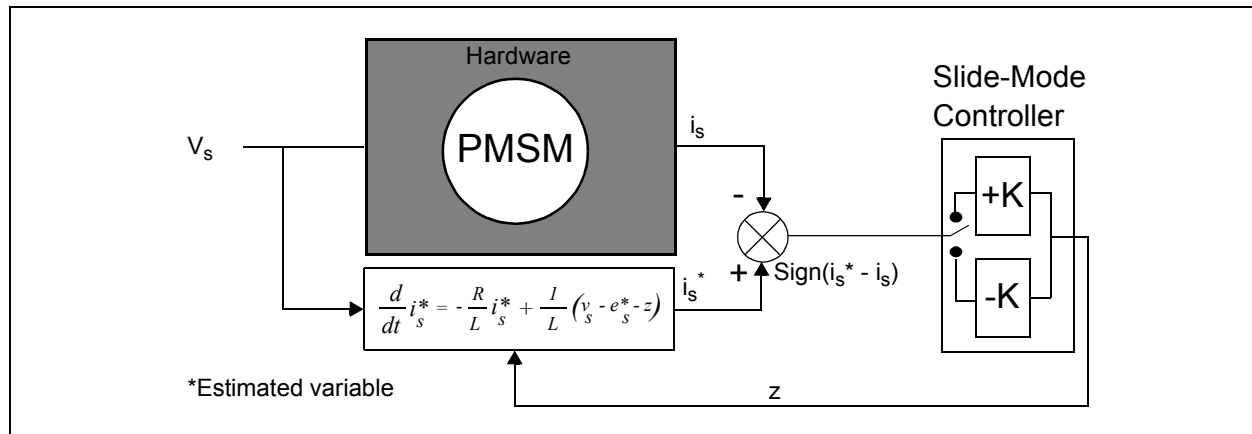
The position and speed estimator is based on a current observer. This observer is a digitized model of the motor, as represented by Equation 1. Variables and constants include:

- Motor Phase Current (i_s)
- Input voltage (v_s)
- back-EMF (e_s)
- Winding resistance (R)
- Winding inductance (L)
- Control period (T_s)
- Output Correction Factor Voltage (z)

The digitized model provides a software representation of the hardware. However, in order to match measured current and estimated current, the digitized motor model needs to be corrected using the closed loop shown in Figure 17.

Considering two motor representations, one in hardware (shaded area) and one in software, with the same input (v_s) fed into both systems, and matching the measured current (i_s) with estimated current (i_s^*) from the model, we can presume that back-EMF (e_s^*) from our digitized model is the same as the back-EMF (e_s) from the motor.

FIGURE 17: CURRENT OBSERVER BLOCK DIAGRAM

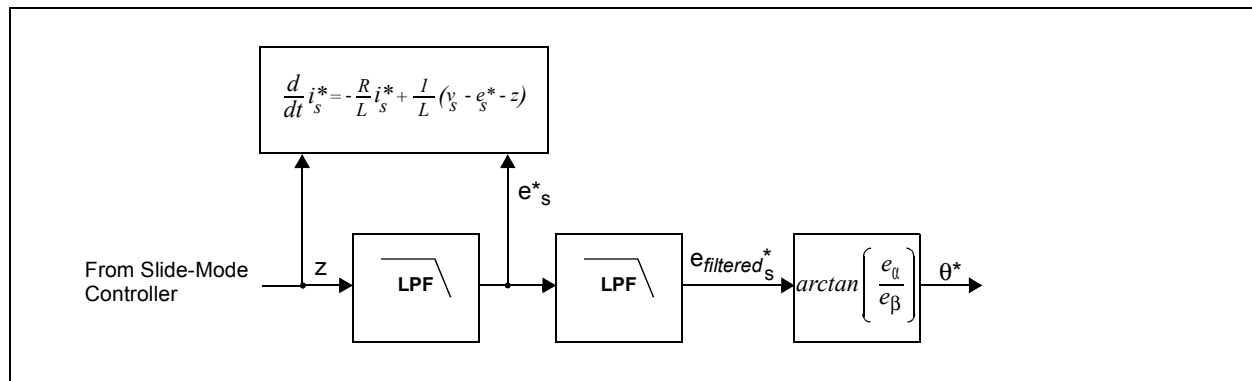


A slide mode controller, or SMC, is used to compensate the digitized motor model. An SMC consists of a summation point that calculates the sign of the error between measured current from the motor and estimated current from the digitized motor model. The computed sign of the error (+1 or -1) is multiplied by an SMC gain (K). The output of the SMC controller is the correction factor (Z). This gain is added to the voltage term from the digitized model, and the process repeats every control cycle until the error between measured current (i_s) and estimated current (i_s^*) is zero (i.e., until measured current and estimated current match).

Back-EMF Estimation

After compensating the digitized model, you have a motor model with the same variable values for the input voltage (v_s) and for current (i_s^*). Once the digitized model is compensated, the next step is to estimate back-EMF (e_s^*) by filtering the correction factor (Z), as shown in Figure 18. The back-EMF estimation (e_s^*) is fed back to the model to update the variable e_s^* after every control cycle. Values e_{α} and e_{β} (vector components of e_s) are used for the estimated theta calculation.

FIGURE 18: BACK-EMF ESTIMATION MODEL



Back-EMF Filtering

To provide the filtering, a first-order, digital low-pass filter is used with Equation 2.

EQUATION 2: FIRST-ORDER DIGITAL LOW-PASS FILTER:

$$y(n) = y(n-1) + T2\pi f_c \cdot (x(n) - y(n))$$

To filter z to obtain e^* , we substitute in the equation for 8 kHz and we get:

$$e(n) = e(n-1) + \left(\frac{1}{f_{pwm}}\right) \cdot 2\pi f_c (z(n) - e(n))$$

Where:

- $e(n)$ = Next estimated back-EMF value
- $e(n-1)$ = Last estimated back-EMF value
- f_{pwm} = PWM frequency at which the digital filter is being calculated
- f_c = Cutoff frequency of the filter
- $z(n)$ = Unfiltered back-EMF, which is output from the slide mode controller

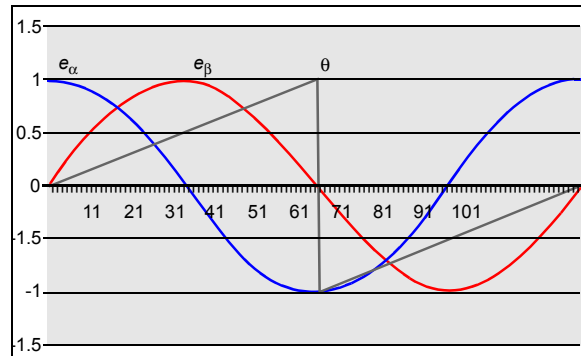
The value of the cutoff frequency depends on the selection of the slide-mode controller gain and is set experimentally.

The output of the first filter is used in two blocks. The first block is the model itself, used to calculate the next estimated current (i_s^*), and also to calculate the estimated theta (θ^*). A second, first-order filter is used to calculate a smoother signal coming out of the motor model.

Relationship Between Back-EMF and Rotor Position

Once the back-EMF has been filtered for the second time, theta is calculated. The relationship between e_s and θ can be explained based on the graph shown in Figure 19.

FIGURE 19: BACK-EMF AND THETA RELATIONSHIP



The plot shows a trigonometric function relating the vector components of the back-EMF (e_α and e_β) and rotor angle (θ). Arctangent is computed on the back-EMF vector components to calculate theta. Equation 3 illustrates how the function is implemented in software:

EQUATION 3: THETA CALCULATION

$$\theta = \arctan(e_\alpha, e_\beta)$$

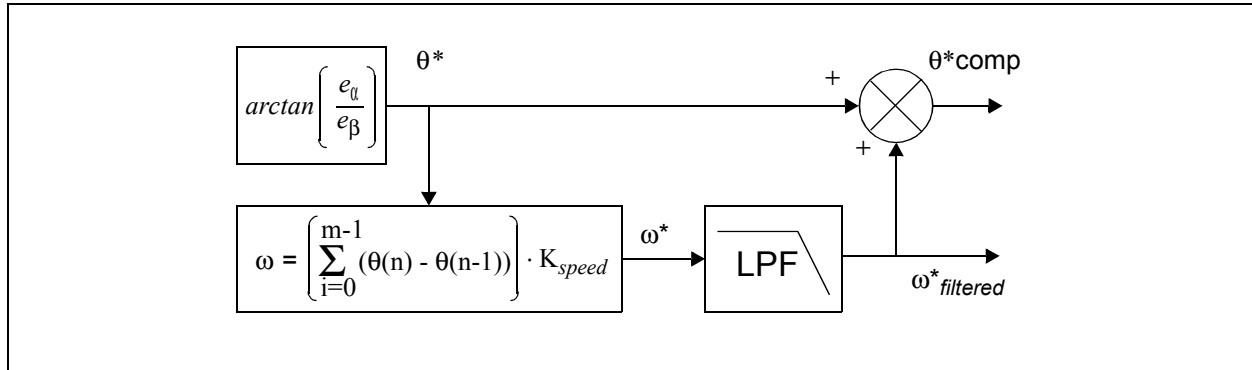
The actual implementation uses a numeric and iterative algorithm called CORDIC (COordinate Rotation by DIgital Computer), which is fast, yet takes less memory than a floating point implementation. Discussion of the CORDIC algorithm is beyond the scope of this application note.

Speed Calculation

Due to the filtering function applied during the theta calculation, some phase compensation is needed before the calculated angle is used to energize the motor windings. The amount of theta compensation depends on the rate of change of theta, or speed of the motor. The theta compensation is performed in two steps:

- First, the speed of the motor is calculated based on the uncompensated theta calculation.
- Then the calculated speed is filtered and used to calculate the amount of compensation, as shown in Figure 20.

FIGURE 20: SPEED CALCULATION BLOCK DIAGRAM



Speed is calculated by accumulating theta values over m samples and then multiplying the accumulated theta by a constant. The formula used in this application note for speed calculation is shown in Equation 4.

EQUATION 4: SPEED CALCULATION

$$\omega = \sum_{i=0}^m (\theta_n - \theta_{n-1}) \cdot K_{speed}$$

Where:

Omega (ω)	=	Angular velocity of the motor
Theta (θ_n)	=	Current Theta value
PrevTheta (θ_{n-1})	=	Previous Theta value
Kspeed	=	Amplification factor for desired speed range
m	=	Number of accumulated Theta deltas

To secure a smoother signal on the speed calculation, a first order filter is applied to Omega (ω^*) to obtain FilteredOmega ($\omega^*_{filtered}$). First order filter topology is the same as the ones used for back-EMF filtering.

Phase Compensation

After uncompensated theta and filtered speed have been calculated, the delays introduced in the filtering process must be removed. This is accomplished by adding a compensating offset theta (θ_{offset}), which is determined by motor speed, to the uncompensated theta, as follows:

$$\theta^*_{comp} = \theta^* + \theta_{offset}$$

It is recommended that phase compensation be fine tuned for any particular motor. In this application, we separated phase compensation into eight speed ranges. Each speed range has its own slope and constant phase compensation component. Table 2 lists the phase compensation formulas used in this equation.

TABLE 2: PHASE COMPENSATION FORMULAS

Speed Range (FilteredOmega)	Phase Compensation Formula (θ_{offset})
Minimum - 0.09375	$\theta_{offset} = 0$
0.09375 - 0.1875	$\theta_{offset} = 4\omega_{filtered} + 0.375$
0.1875 - 0.2876	$\theta_{offset} = 2\omega_{filtered}$
0.2876 - 0.38095	$\theta_{offset} = \omega_{filtered} + 0.2876$
0.38095 - 0.475	$\theta_{offset} = \omega_{filtered} + 0.2876$
0.475 - 0.568	$\theta_{offset} = 0.5\omega_{filtered} + 0.5251$
0.568 - 0.756	$\theta_{offset} = 0.25\omega_{filtered} + 0.6671$
0.756 - Maximum	$\theta_{offset} = 0.125\omega_{filtered} + 0.748275$

Changing Phase Compensation Formulas

In the ZIP file containing the source code of this application note, there is a spreadsheet that will help the users calculate the angle compensation formulas. The user is required to type Speed in RPMs and Theta to be compensated at that particular speed. For example, if at 500 RPMs the phase compensation needs to be 30 degrees, then the user needs to type 500 under the "Speed (RPMs)" field, and 30 under the "Angle Compensation (Degrees)" field. The spreadsheet will calculate the slope and constant for the line equation.

In the same spreadsheet, the user needs to define the following parameters to translate speed from RPMs to Q15 format:

Ts: PWM Period

Kspeed: Speed Constant

m: Number of Accumulated Theta Deltas per Speed Calculation

Pole Pairs: Number of Motor Pole Pairs

FLOW CHARTS

The FOC algorithm is executed at the same rate as the PWM. It is configured so that the PWM triggers A/D conversions for two windings using two shunt resistors and a potentiometer that sets the reference speed of the motor. Interrupts of the A/D are enabled to perform the algorithm. Figure 21 shows the general execution of the A/D Interrupt subroutine.

FIGURE 21: A/D INTERRUPT SUBROUTINE

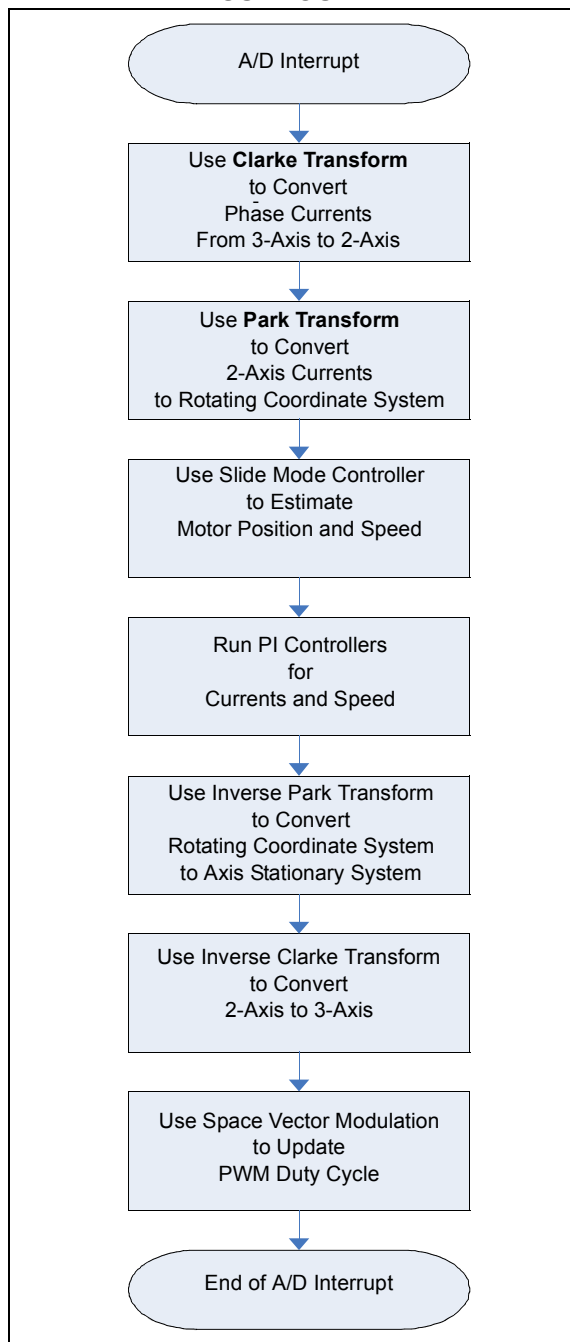
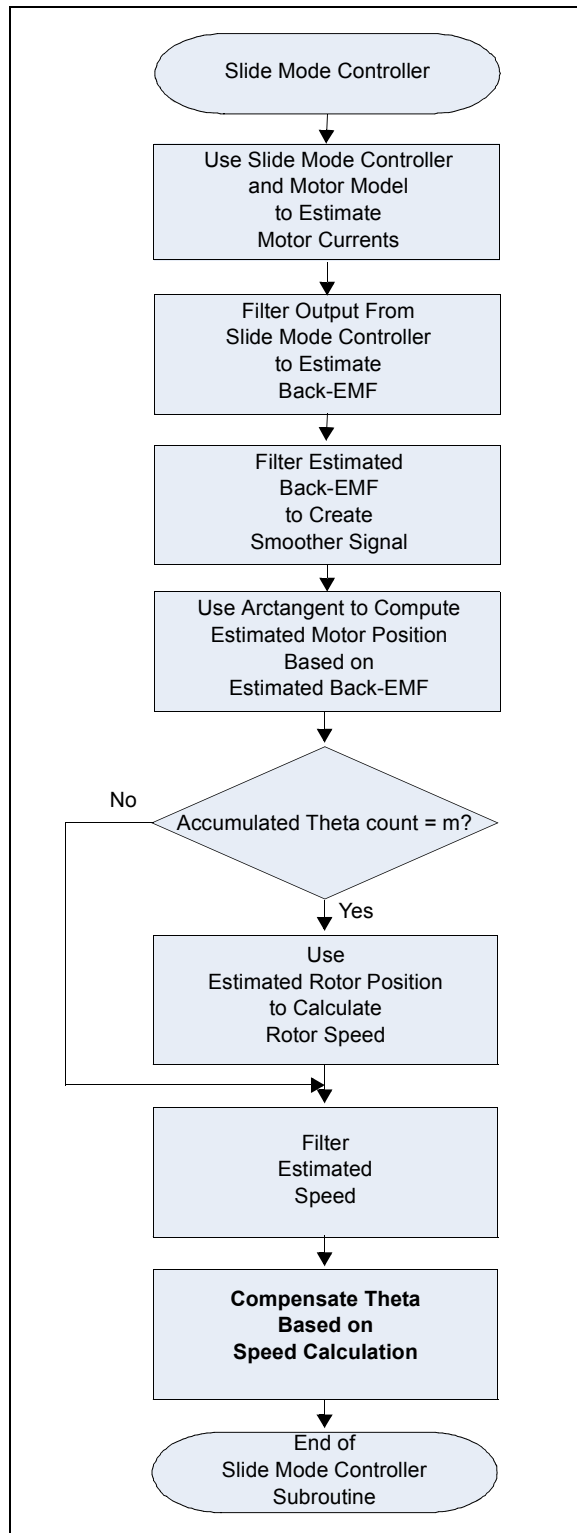


Figure 22 shows the process of using the Slide Mode Controller to estimate the position and speed of the motor.

FIGURE 22: MOTOR POSITION AND SPEED ESTIMATION



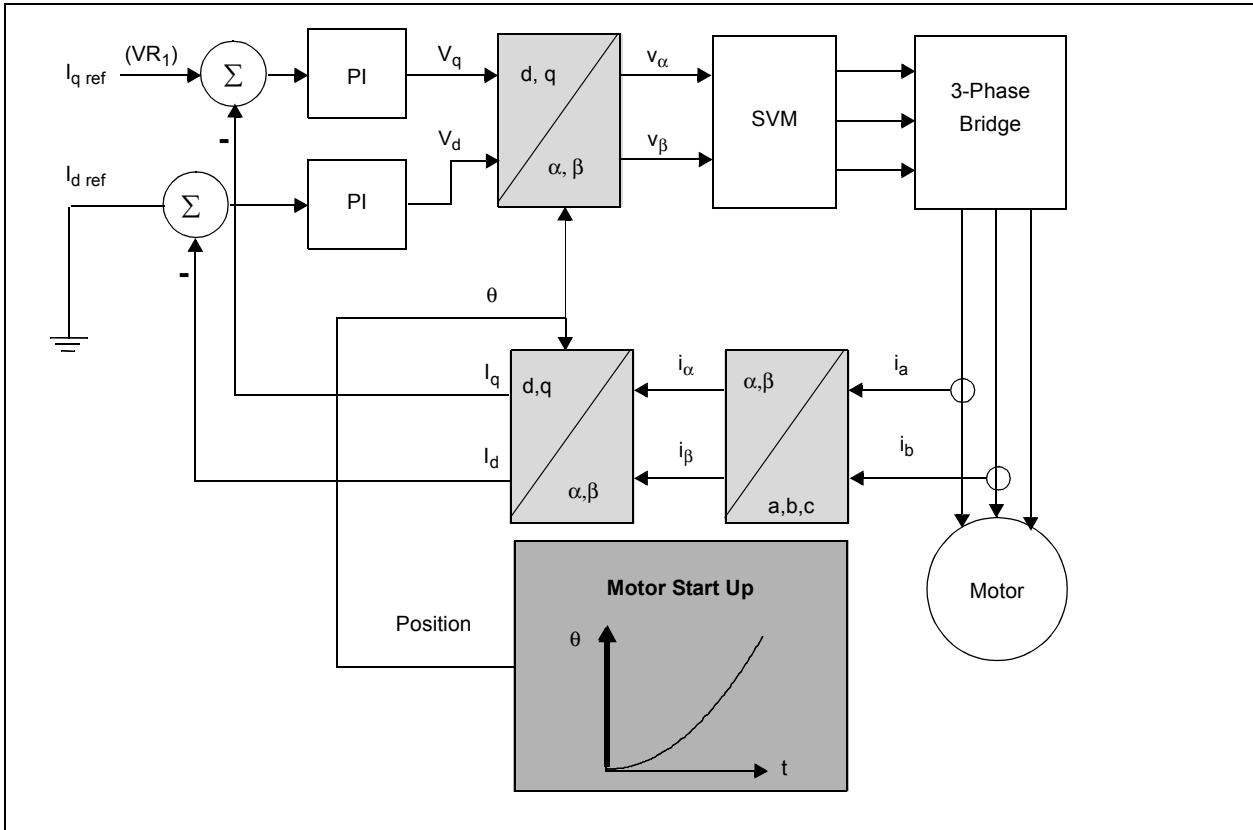
MOTOR START-UP

Since the sensorless FOC algorithm is based on the back-EMF estimation, a minimum speed is needed to get the estimated back-EMF value. Therefore, the motor windings must be energized with the appropriate estimated angle. To handle this, a motor start-up subroutine (see Figure 23) was developed.

When the motor is at standstill, and the start/stop button has been pressed, the dsPIC DSC generates a series of sinusoidal voltages to get the motor spinning. The motor spins at a fixed acceleration rate, and the FOC algorithm controls the currents I_d and I_q . The angle Theta (commutation angle) is incremented based on the acceleration rate.

As shown in the diagram, phase angle is incremented at a squared rate to get a constant acceleration on the motor. Even if Theta is being generated by the open loop-state machine, Field Oriented Control blocks are still being executed and are controlling torque component current and flux component current. An external potentiometer is used to set the desired torque required to start the motor. This potentiometer is set experimentally depending on mechanical load characteristics. This start-up subroutine provides a constant torque to start up the motor. At the end of the start-up ramp, the software switches over to closed-loop, sensorless control, taking Theta from the position and speed estimator shown in Figure 6.

FIGURE 23: MOTOR START-UP



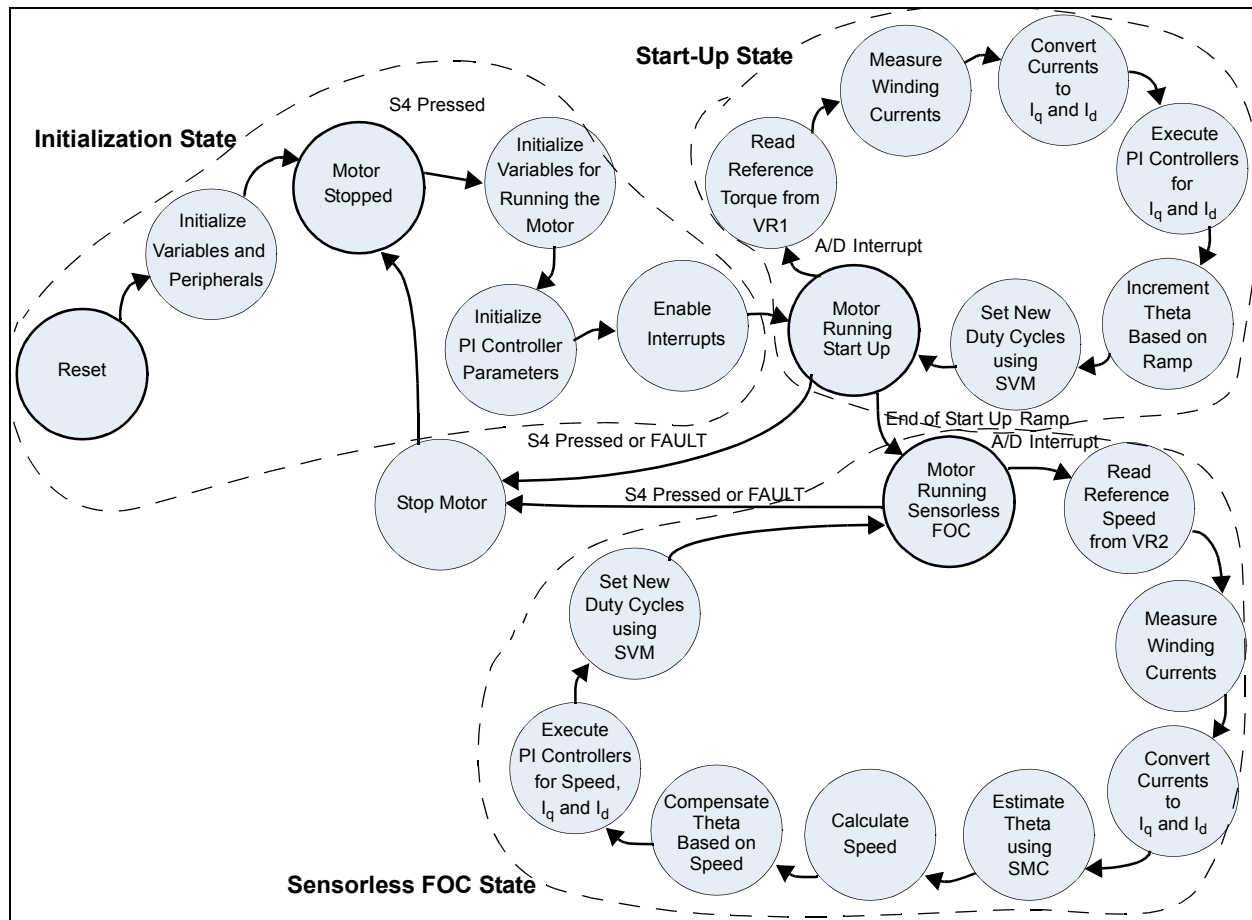
MAIN SOFTWARE STATE MACHINE

It is helpful to visualize the FOC algorithm as a software state machine (see Figure 24). First, the motor windings are de-energized and the system waits for the user to press the start/stop button (S4 on the development board). Once the user presses start/stop button, the system enters initialization state, where all variables are set to their initial value and interrupts are enabled. Then the start-up subroutine is executed, where current components for torque (I_q) and flux production (I_d) are being controlled, and commutation angle (Theta) is being generated in a ramp fashion to get the motor spinning.

After going through the start-up subroutine, the system switches over to sensorless FOC control, where the speed controller is added to the execution thread, and the Slide Mode Controller (SMC) starts estimating theta as previously explained. When the motor enters sensorless FOC control state, the reference speed is continuously read from an external potentiometer and the start/stop button is monitored to stop the motor.

Any fault in the system causes the motor to stop and return to Motor Stopped state until S4 is pressed again. The state diagram shows all the different states of the software and the conditions that make the system transition to a different state.

FIGURE 24: MAIN SOFTWARE STATE MACHINE



BENEFITS OF DSC-BASED FOC CONTROL

A major advantage of deploying DSCs in motor control is the practicality of a common design platform, which makes the production of appliances more efficient. This means appliance makers now have an economical way to offer a range of appliance models that use PMSM or other type of motors with sensorless FOC algorithm control.

These software-based motor control designs enable rapid customization to address multiple markets by changing only the control parameters.

Protection of firmware Intellectual Property (IP) is another major issue for manufacturers who frequently deploy appliance design teams that collaborate across many geographies. It is easy to imagine a scenario in which the implementation of FOC for an appliance might have come from place A, user-interface board from place B and final system integration being done in place C.

In developing their designs, it is more than likely that these design teams will have claims to their own IP. Microchip's dsPIC DSC family offers the CodeGuard™ Security feature, which helps to protect IP in collaborative design environments (for more information, see www.microchip.com/codeguard).

CONCLUSION

This application note illustrates how designers can take advantage of DSCs to implement advanced motor control techniques such as the Sensorless FOC algorithm in appliance applications. Since programming the dsPIC DSCs is similar to programming the MCUs, appliance designers can quickly design their motor control algorithm and test their prototypes.

Fine tuning motor control is made easy, thanks to the powerful IDE-based tools such as the DMCI, which allows the designers to easily port their algorithms across a variety of motor platforms including PMSM, BLDC, Brushed DC (BDC) and ACIM motors.

Microchip has various resources to assist you in tuning these parameters. Contact your Microchip sales or application engineer if you would like further support.

APPENDIX

Hardware Resources

The hardware used to implement this sensorless FOC application consists of the following components:

- dsPICDEM™ MC1 Motor Control Development Board (DM300020)
- dsPICDEM™ MC1H 3-Phase High Voltage Power Module (DM300021)
- Toshiba Compressor Model HD187X1-S12FD.

MC1 Development tools can be obtained from the Microchip web site (www.microchip.com). Customers with their own compressor may refer to the following motor parameters:

Number of poles:	4 (2 pole pairs)
Speed range:	900 to 7200 RPM
Nominal Output Power (P):	750 W
Winding Resistance (R):	0.70
Winding Inductance Average (L):	7.35 mH
Nominal Current (I):	6.0 ± 0.3 A
Line-to-line back-EMF constant:	0.0228 V _{rms} /RPM
Torque constant:	0.39 N m/A

Hardware Modifications

Out of the box, the high voltage power module has a nominal power output of 0.8 kVA. To drive a PMSM air conditioner compressor with a load represented by the listed parameters, the module must be modified. The board must be removed from its housing to complete portions of this modification.

Certain modifications, as described below, have been allowed for in the design of the system. Clearly, any additional modifications that the user chooses to make can not be guaranteed to be functional or safe. It is assumed that relevant qualified personnel only will use the system.

Note: Refer to the “dsPICDEM™ MC1H 3-Phase High Voltage Power Module User’s Guide” (DS70096) for additional information.

MODIFYING THE BOARD (REFER TO FIGURE 25)

Step One:

Establish a common power and digital ground by adding a high-current jumper

Step Two:

Replace the existing motor current-sensing shunt resistors

Step Three:

Configure power module jumper settings

Step Four:

Bypass Fault signals from Hall Effect sensors

Step Five:

Activate feedback current-sensing circuit by installing selection resistors

Hall type current sensors cannot be used for current sensing after these modifications. Hall over-current fault is also disabled with these modifications; however, shunt over-current fault is still active for over-current faults.

ACCESSING THE BOARD

Before removing the lid (see Figure 25), the following procedure should be rigidly followed:

- Turn off all power to the system.
- Wait a minimum of 3 minutes so that the internal discharge circuit has reduced the DC bus voltage to a safe level. The red LED bus voltage indicator visible through the top ventilation holes should be out.
- Verify with a voltmeter that discharge has taken place by checking the potential between the + and - DC terminals of the 7-pin output connector before proceeding. The voltage should be less than 10V.
- The system is now safe to work on.
- Remove all cables from the system.
- Remove the screws fixing the lid to the chassis and heat sink on the top and bottom.
- Slide the lid forwards while holding the unit by the heat sink.

After the board is out of the housing, follow the detailed procedures shown in Figures 26 through 30.

FIGURE 25: dsPICDEM™ MC1H 3-PHASE HIGH VOLTAGE POWER MODULE WITH COVER REMOVED

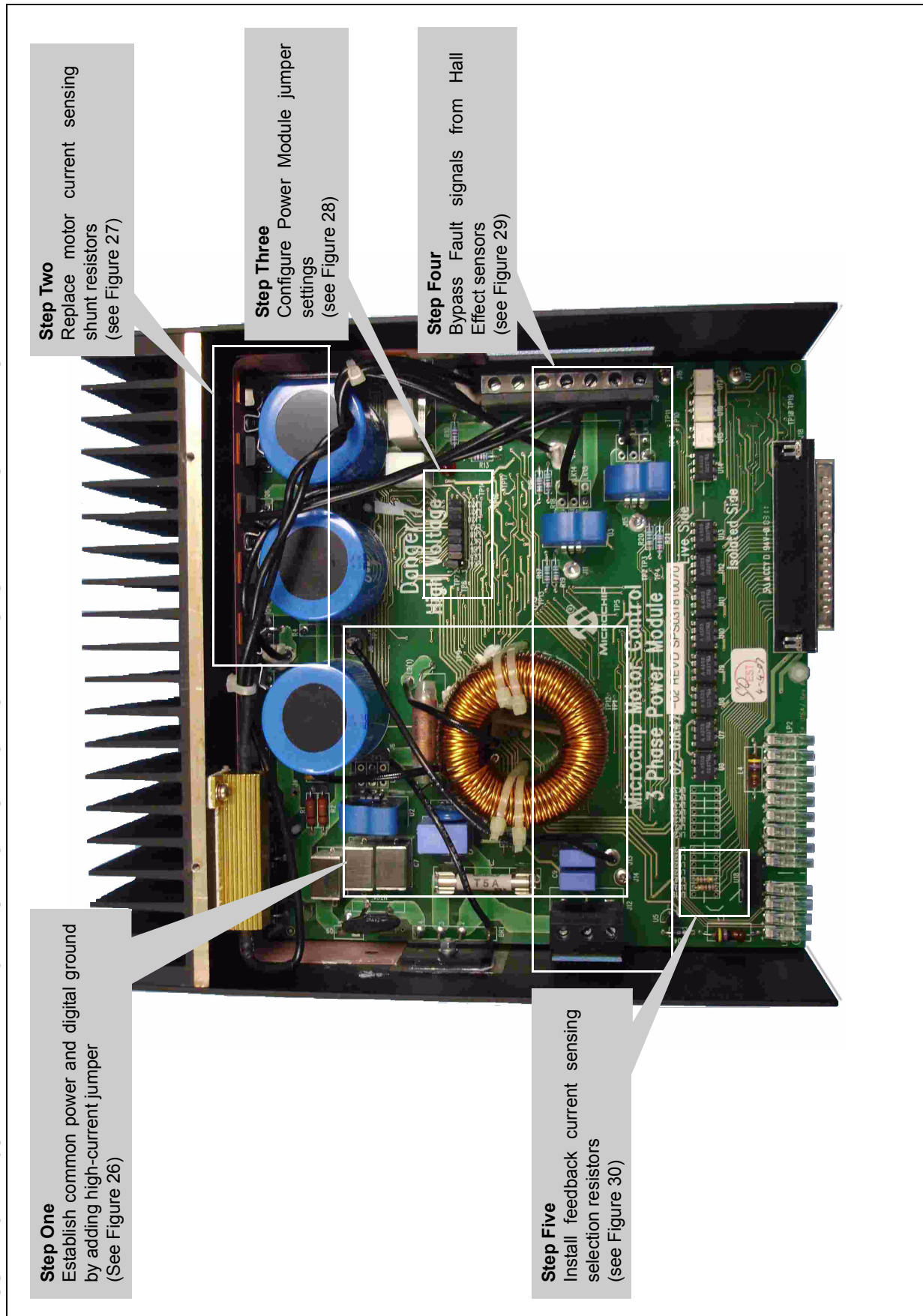
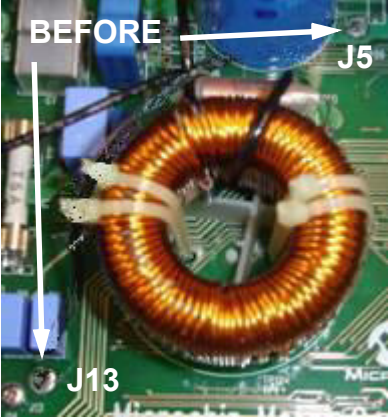


FIGURE 26: ESTABLISH COMMON POWER AND DIGITAL SIGNAL GROUND

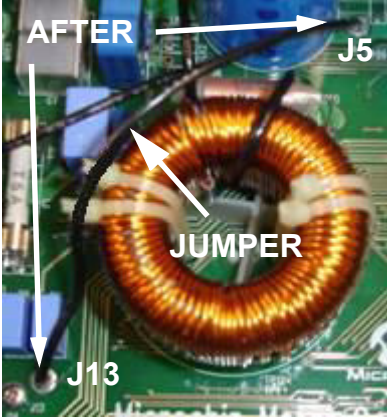
Step One: Establish common power and digital signal ground



BEFORE

Because shunt resistors are used to sense current from the motor, power and digital signals must use the same ground.

Solder a high-current jumper wire (AWG 18 minimum) between J5 and J13.

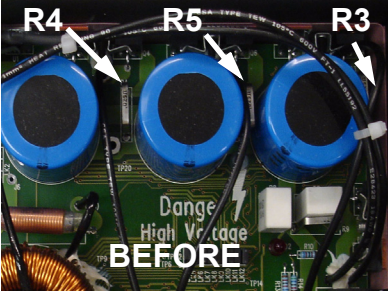


AFTER

JUMPER

FIGURE 27: REPLACE CURRENT-SENSING SHUNT RESISTORS

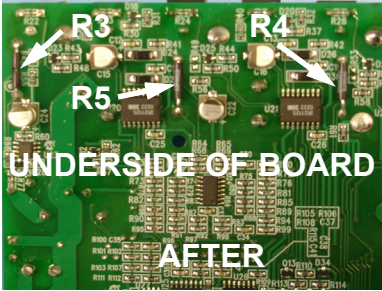
Step Two: Replace motor current-sensing shunt resistors



BEFORE

To accommodate higher current, the current-sensing shunt resistors must be changed from the default value of 25 mΩ/3W to 10 mΩ/3W.

The recommended method is to cut shunt resistors R3, R4 and R5 from the top side of the board and solder replacement shunt resistors on the underside of the board.




UNDERSIDE OF BOARD

AFTER

FIGURE 28: CONFIGURE JUMPER SETTINGS

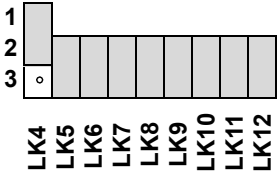
Step Three: Configure Power Module jumper settings



The illustrated jumper settings ensure a gain of 16.67, which provides a feedback of 6A per 1V. Jumper LK4 provides an offset of 2.5V to enable measurement of both positive and negative currents.

The new formula for current measurement is:

$$V = 2.5 + I/6$$



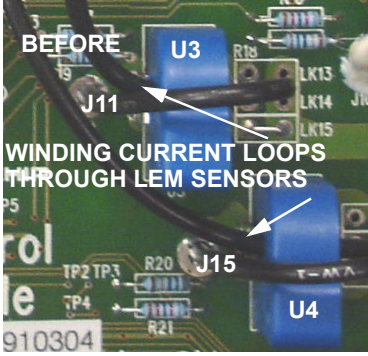
1	2	3	4	5	6	7	8	9	10	11	12
		o									
LK4	LK5	LK6	LK7	LK8	LK9	LK10	LK11	LK12			

FIGURE 29: DISABLE HALL EFFECT SENSORS

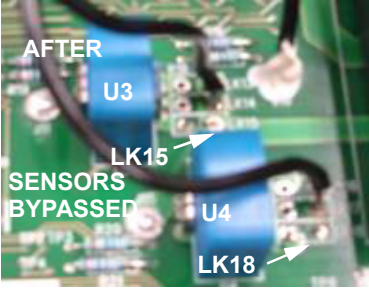
Step Four: Bypass Fault signals from Hall Effect sensors

The dsPICDEM™ MC1H 3-Phase High Voltage Power Module uses three Hall Effect sensors. Two sensors are used with winding currents; the third is used with the I_{BUS} current. These sensors, and their associated feedback circuitry, generate Fault signals that must be bypassed for the air conditioning compressor application. Follow these steps:

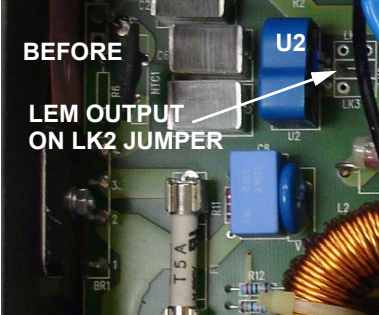
1. To bypass Sensor U3, unsolder the wire at J11 and unloop it from the sensor.
2. Resolder wire to the right side of jumper LK15.
 - 2a. Cut the jumper on LK15.
 - 2b. Cut the jumper on LK18.
3. Repeat for U4, moving from J15 to LK18.
4. To bypass the I_{BUS} sensor (U2), cut the jumper on LK2.
5. Solder a high-current (AWG 18) wire from the right side of jumper LK2 to pin 1 of the input diode bridge



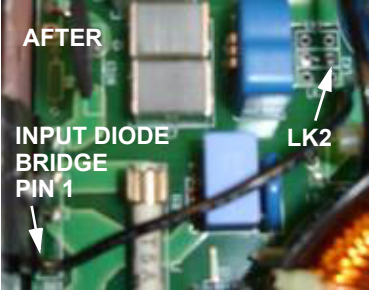
BEFORE
WINDING CURRENT LOOPS THROUGH LEM SENSORS



AFTER
SENSORS BYPASSED



BEFORE
LEM OUTPUT ON LK2 JUMPER



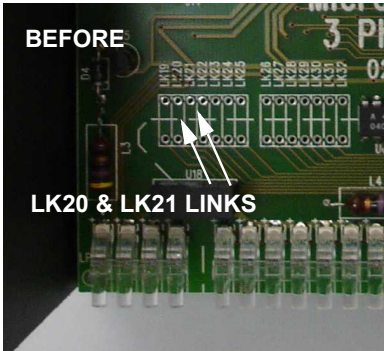
AFTER
INPUT DIODE BRIDGE PIN 1

FIGURE 30: INSTALL FEEDBACK CURRENT SELECTION RESISTORS

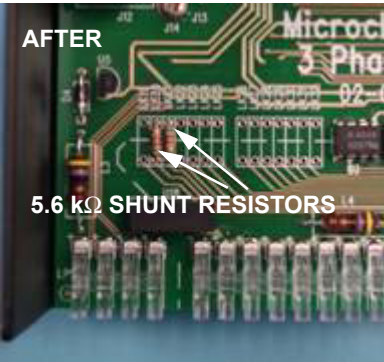
Step Five: Install feedback current sensing selection resistors

To obtain feedback current, the circuit links must be completed.

To activate the current feedback for this application, populate links LK20 and LK21 with 5.6 kΩ resistors.



BEFORE
LK20 & LK21 LINKS



AFTER
5.6 kΩ SHUNT RESISTORS

REFERENCES

Several application notes have been published by Microchip Technology describing the use of DSCs for motor control.

For ACIM control see:

- AN984, “An Introduction to AC Induction Motor Control Using the dsPIC30F MCU” (DS00984)
- AN908, “Using the dsPIC30F for Vector Control of an ACIM” (DS00908)
- GS004, “Driving an ACIM with the dsPIC® DSC MCPWM Module” (DS93004)

For BLDC motor control see:

- AN901, “Using the dsPIC30F for Sensorless BLDC Control” (DS00901)
- AN957, “Sensored BLDC Motor Control Using dsPIC30F2010” (DS00957)
- AN992, “Sensorless BLDC Motor Control Using dsPIC30F2010” (DS00992)
- AN1083, “Sensorless BLDC Control with Back-EMF Filtering” (DS01083)

For PMSM control see:

- AN1017, “Sinusoidal Control of PMSM Motors with dsPIC30F DSC” (DS01017)

For information on the dsPICDEM MC1 Motor Control Development Board see:

- “dsPICDEM™ MC1 Motor Control Development Board User’s Guide” (DS70098)
- “dsPICDEM™ MC1H 3-Phase High Voltage Power Module User’s Guide” (DS70096)
- “dsPICDEM™ MC1L 3-Phase Low Voltage Power Module User’s Guide” (DS70097)

These documents are available on the Microchip web site (www.microchip.com).

AN1078

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==**

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://support.microchip.com>
Web Address:
www.microchip.com

Atlanta

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston

Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago

Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas

Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit

Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo

Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles

Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara

Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto

Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing

Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Fuzhou

Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR

Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen

Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde

Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian

Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore

Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune

Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama

Tel: 81-45-471-6166
Fax: 81-45-471-6122

Korea - Gumi

Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul

Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang

Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila

Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore

Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu

Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung

Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei

Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok

Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen

Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan

Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen

Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham

Tel: 44-118-921-5869
Fax: 44-118-921-5820

12/08/06