
Sensorless BLDC Control with Back-EMF Filtering Using a Majority Function

<i>Author: Daniel Torres Microchip Technology Inc.</i>
--

INTRODUCTION

This application note describes a sensorless Brushless Direct Current (BLDC) motor control algorithm that is implemented using dsPIC[®] Digital Signal Controller (DSC). The algorithm works utilizing a majority function for digitally filtering the Back-Electromotive Force (BEMF). Each phase of the motor is filtered to determine when to commutate the motor drive voltages. This control technique excludes the need for discrete, low-pass filtering hardware and off-chip comparators. It should be pointed out that all the discussions here, and the application software, assume a 3-phase motor has to be used. The motor control algorithm described here has six main parts:

- Sampling trapezoidal BEMF signals using the dsPIC Analog-to-Digital Converter (ADC)
- Reconstructing the Motor Virtual Neutral Point
- Comparing the trapezoidal BEMF signals to the reconstructed motor virtual neutral point to detect the zero crossing points
- Filtering the signals coming from the comparisons using a majority function filter
- Commutate the motor driving voltages
- Control loop

The purpose of this application note is to explain the different parts of the implementation of this new sensorless BLDC motor control technique in a basic and simple form. It also shows that this new control method is a single-chip dsPIC DSC device-based solution, which does not require external hardware except for a couple of resistors used to conditioning the BEMF signals to the dsPIC DSC device ADC operational voltage range.

SENSORED CONTROL VERSUS SENSORLESS CONTROL

The BLDC motor is used for both consumer and industrial applications owing to its compact size, controllability and high efficiency. Increasingly, it is also used in automotive applications as a part of strategy to eliminate belts and hydraulic systems, to provide additional functionality and to improve fuel economy. The continuing reduction in cost of magnets and the electronics required for the control of BLDC motors has contributed to its use in an increasing number of applications and at higher power levels.

The BLDC motor is usually operated with one or more rotor position sensors since the electrical excitation must be synchronous to the rotor position. For reasons of cost, reliability, mechanical packaging and especially if the rotor runs immersed in fluid, it is desirable to run the motor without position sensors, which is commonly known as sensorless operation.

It is possible to determine when to commutate the motor drive voltages by sensing the back-EMF voltage on an undriven motor terminal during one of the drive phases. The obvious cost advantage of sensorless control is the elimination of the Hall position sensors. Although there are some disadvantages to sensorless control:

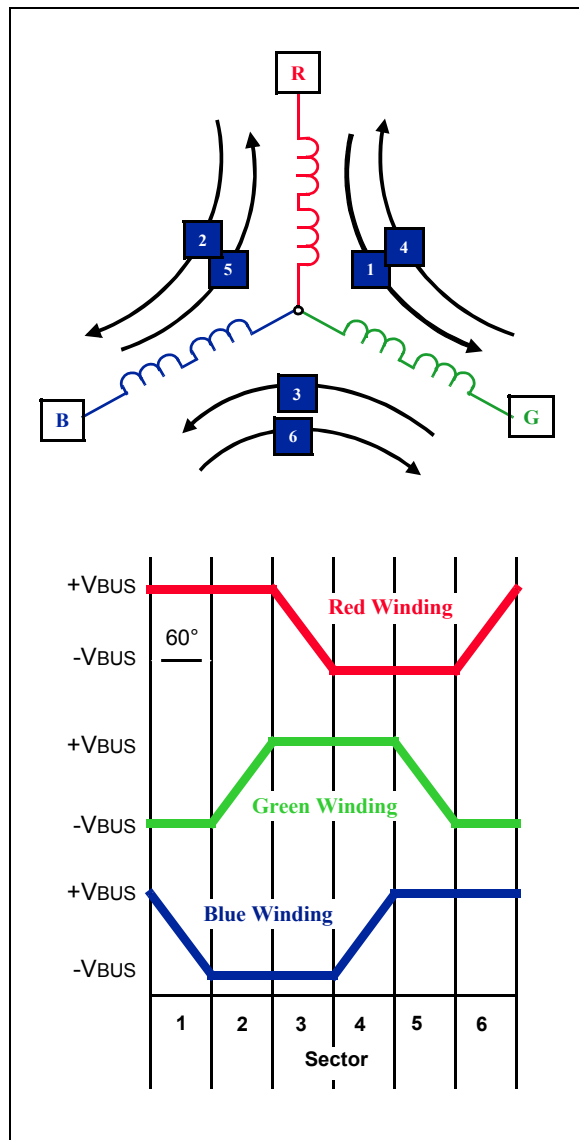
- The motor must be moving at a minimum rate to generate sufficient back-EMF to be sensed
- Abrupt changes to the motor load can cause the BEMF drive loop to go out of lock
- The BEMF voltage can be measured only when the motor speed is within a limited range of the ideal commutation rate for the applied voltage
- Commutation at rates faster than the ideal rate will result in a discontinuous motor response

If low cost is a primary concern and low-speed motor operation is not a requirement, and the motor load is not expected to change rapidly, sensorless control may be the better choice for your application. However, there are specific algorithms to overcome all the disadvantages listed above. The Sensorless BEMF method is quickly becoming the most popular solution.

SIX-STEP (Trapezoidal) Commutation

The method for energizing the motor windings in the sensorless algorithm described in this application note is six-step trapezoidal or 120° commutation. Figure 1 shows how the six-step commutation works. Each step, or sector, is equivalent to 60 electrical degrees. Six sectors make up 360 electrical degrees, or one electrical revolution.

FIGURE 1: SIX-STEP COMMUTATION



The arrows in the winding diagram show the direction in which the current flows through the motor windings in each of the six steps. The graph shows the potential applied at each lead of the motor during the six steps. Sequencing through these six steps moves the motor one electrical revolution.

STEP COMMUTATION

- Step 1
 - Red winding is driven positive.
 - Green winding is driven negative.
 - Blue winding is not driven.
- Step 2
 - Red winding remains positive.
 - Blue winding is driven negative.
 - Green winding is not driven.
- Step 3
 - Green winding is driven positive.
 - Blue winding is driven negative.
 - Red winding is not driven.
- Step 4
 - Green winding is driven positive.
 - Red winding is driven negative.
 - Blue winding is not driven.
- Step 5
 - Blue winding is driven positive.
 - Red winding is driven negative.
 - Green winding is not driven.
- Step 6
 - Blue winding is driven positive.
 - Green winding is driven negative.
 - Red winding is not driven.

For every sector, two windings are energized and one winding is not energized. The fact that one of the windings is not energized during each sector is an important characteristic of six-step control that allows for the use of a sensorless control algorithm.

This application note uses these terms to describe motor speed:

- Electrical revolutions per minute (RPM_{Elec})
- Electrical revolutions per second (RPSE_{Elec})

It is easier to discuss motor speed in these terms rather than mechanical RPM because when talking about electrical RPM, the number of motor poles need not be factored in. The relationship between mechanical and electrical RPM is seen in the following three equations:

EQUATION 1: M/E RPM RELATIONSHIP

$$RPM_{Mech} = \frac{2 \cdot RPM_{Elec}}{\text{Number of Motor Poles}}$$

EQUATION 2: E/M RPM RELATIONSHIP

$$RPM_{Elec} = \frac{RPM_{Mech} \cdot \text{Number of Motor Poles}}{2}$$

EQUATION 3: E/M RPS RELATIONSHIP

$$RPSElec = \frac{RPMElec}{60}$$

BEMF Sensing Methods

When a BLDC motor rotates, each winding generates BEMF which opposes the main voltage supplied to the windings according to the Lenz's law. The polarity of this BEMF is in the opposite direction of the energizing voltage. BEMF is mainly dependent on three motor parameters:

- Number of turns in the stator windings
- Rotor's Angular Velocity
- Magnetic field generated by rotor magnets

BEMF (in terms of motor parameters and angular velocity) can then be calculated using the expression given in Equation 4.

EQUATION 4: BACK-EMF (BEMF)

$$BEMF = NlrB\omega$$

where

N = number of windings per phase

l = length of the rotor

r = internal radius of the rotor

B = rotor magnetic field

ω = angular velocity

As seen on the Equation 4, the only variable term is the rotor angular speed. Therefore, the BEMF is proportional to the rotor speed; as the speed increases the BEMF increases.

The BEMF waveform of the motor varies as both a function of the rotor's position and speed. Detection of position using the BEMF at zero and very low speeds is, therefore, not possible. Nevertheless, there are many applications (for example, fans and pumps) that do not require positioning control or closed-loop operation at low speeds. For these applications, a BEMF method is very appropriate. There are many different methods of using the BEMF. The majority of these methods can be summarized as follows:

- Motor terminal voltage sensing
 - Either by direct measurement or inference (knowledge of switch states and DC bus voltage).
- Mid-point voltage sensing
 - Only works for Y- and delta-connected motors. Certain classes of winding connections may not work
 - 4th wire not actually required. Can recreate star point using the three motor phases
- Bus current gradient sensing
 - Relies on characteristic bus current shape due to commutation changing as rotor leads or lags
 - Cannot use fast bus current control

THE SELECTED BEMF SENSING METHOD

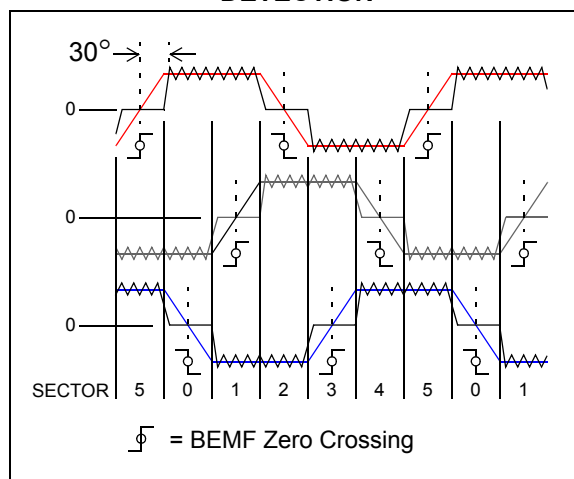
This application note is based on the Mid-Point voltage reconstruction and is also based on detecting the instances when the BEMF of an inactive phase is zero. Therefore, it is important to mention that the BEMF sensing method described in this application note can only be implemented using trapezoidal BEMF signals in order to have zero crossing events.

One important feature of this sensing method is the fact that only a few external components are required to determine the zero crossing points. Apart from the BEMF signal conditioning and the power switch gate drivers, the implementation is single-chip with the dsPIC DSC device providing all of the control functionality.

The BEMF zero-crossing technique was chosen because of the following reasons:

- It is suitable for use on a wide range of motors.
- It can be used on both Y- and delta-connected 3-phase motors in theory. Certain classes of connected motors may not work.
- It requires no detailed knowledge of motor properties.
- It is relatively insensitive to motor manufacturing tolerance variations.
- It will work for either voltage or current control.
- The zero-crossing technique is suitable for a wide range of applications where closed-loop operation near zero speed is not required. Provided the speed is greater than zero, there are only two positions per electrical cycle when the BEMF of a phase is zero, and these positions can be distinguished by the slope of the BEMF through the zero crossing as shown in Figure 2.

FIGURE 2: ZERO CROSSING DETECTION



Each sector corresponds to one of six equal 60° portions of the electrical cycle (the sector numbering is completely arbitrary). Commutations occur at the boundary of each of the sectors. Therefore, it is the sector boundaries that need to be detected. There is a 30° offset between the BEMF zero-crossings and required commutation positions, which must be compensated for to ensure efficient and smooth operation of the motor.

Figure 2 also shows the individual idealized phase BEMF waveforms. Assuming only the three motor leads are available for sensing the BEMF, then the voltage of the star point of the motor must be determined because the BEMF waveform will be offset by the star point voltage.

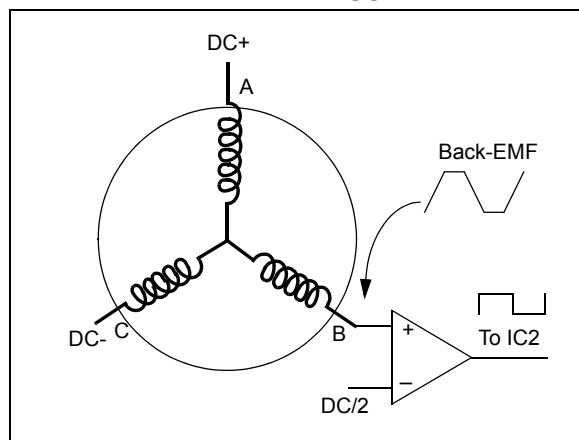
BEMF ZERO CROSSING SENSING METHODS

BEMF voltage zero-crossing signals can be detected by different methods. This section describes two different sensing methods. All of these methods have advantages and as well as drawbacks, which will be discussed in the next section. All these methods are based on the fact that most of the times the motor neutral point is not available, due to the fact that either it was not wired or the motor windings were built in a delta mode.

Comparing the BEMF Voltage to Half the DC Bus Voltage

This method consists of comparing the BEMF voltage to half the DC bus voltage by using comparators assuming that the zero crossing events occur when the BEMF is equal to $V_{DC}/2$. Figure 3 shows the circuitry used to implement this method.

FIGURE 3: BEMF VOLTAGE COMPARED TO HALF OF THE DC BUS



Assume that the motor is in the commutation Step 1 (according to Figure 1), in which Phase A is connected to +V_{BUS} through an electronic switch and Phase C is connected to -V_{BUS} through an electronic switch and Phase B is open. The BEMF signal observed on Phase B has a negative slope and its maximum value is almost equal to +V_{DC} just before the commutation Step 2 occurs. The Phase B reaches the +V_{DC} value when the commutation Step 2 occurs.

At that moment, Phase B is now connected to +V_{DC} through an electronic switch, Phase A is now open and Phase C remains connected to -V_{DC}. The BEMF signal observed on Phase A has a positive slope and its minimum value is almost equal to -V_{DC} just before the commutation Step 3 occurs. Both slopes observed on Phase B and Phase A are compared to $V_{DC}/2$ in order to determine the zero crossing event. This circuitry is easy to implement with three operational amplifiers configured as comparators.

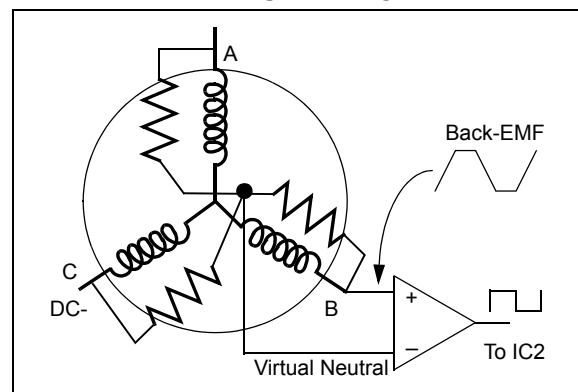
The drawbacks of this method are:

- This method assumes that the motor windings parameters are identical.
- The sensed BEMF signals have positive and negative phase shifts.
- Motor-rated voltage is less than the V_{DC} voltage most of the time; therefore, the zero crossing event not always occurs at $V_{DC}/2$.

Comparing the BEMF Voltage to the Motor Neutral Point

The zero crossing sensing method described before can be improved by having a variable threshold voltage point used to detect the zero crossing events. This variable voltage is in fact the motor neutral point. Often, the motor manufacturers do not wire the motor neutral point. However, it can be generated by using a resistor network. Three networks are connected in parallel with the motor windings and connected together to generate a virtual neutral point. These connections are shown in Figure 4.

FIGURE 4: BEMF VOLTAGE COMPARED TO A VIRTUAL NEUTRAL POINT



The method used in this application note is based on the same principle. However, the neutral point signal is reconstructed by software. The neutral voltage is equal to the average of the BEMF signals. Therefore, the zero crossing threshold value is expressed in Equation 5.

EQUATION 5: VIRTUAL NEUTRAL POINT AND BEMF SIGNALS RELATIONSHIP

$$V_n = \frac{BEMF\ A + BEMF\ B + BEMF\ C}{3}$$

where

V_n is motor neutral voltage

$BEMF\ A$ is the BEMF voltage observed in Phase A

$BEMF\ B$ is the BEMF voltage observed in Phase B

$BEMF\ C$ is the BEMF voltage observed in Phase C

Then the reconstructed motor neutral voltage is compared to each BEMF signal to determine the zero crossing events. A zero crossing event occurs when the BEMF signals are equal to the motor neutral point. Figure 5 shows the BEMF signals measured with the ADC.

The implementation of this method by software is discussed in the further sections. The challenges of this method consist of determining the correct time where the BEMF signals should be sampled since the samples acquired by the ADC may be affected by the resonant transition voltages caused by the PWM switching frequency. These samples may be also affected by the kickback currents produced by the windings de-energization. Figure 6 shows the BEMF signals and the motor virtual neutral point. Figure 7 shows the BEMF signals and the reconstructed virtual neutral point.

The advantage of this method is that it is more flexible in terms of measurement. When the speed varies, the winding characteristics may fluctuate, resulting in variation of the BEMF. In such situations the dsPIC DSC device has complete control over the determination of the zero crossing point. A digital filter is implemented to filter out the high-frequency switching noise components from the BEMF signal.

FIGURE 5: BEMF VOLTAGE MEASURED USING THE dsPIC® DSC ADC

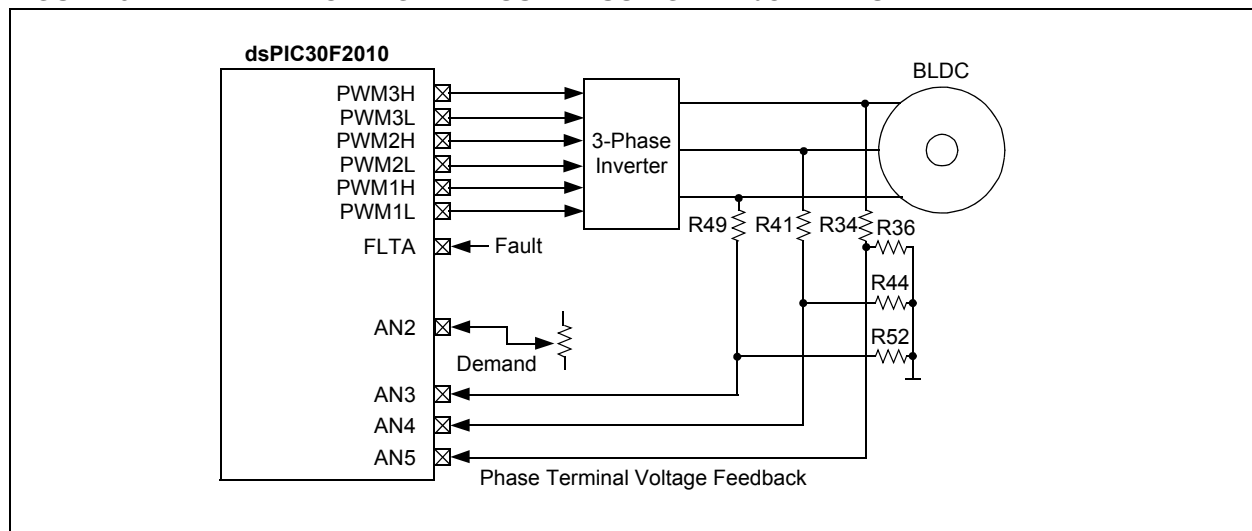


FIGURE 6: BEMF SIGNALS VERSUS VIRTUAL NEUTRAL POINT WHEN THE PWM DUTY CYCLE IS EQUAL TO 100%

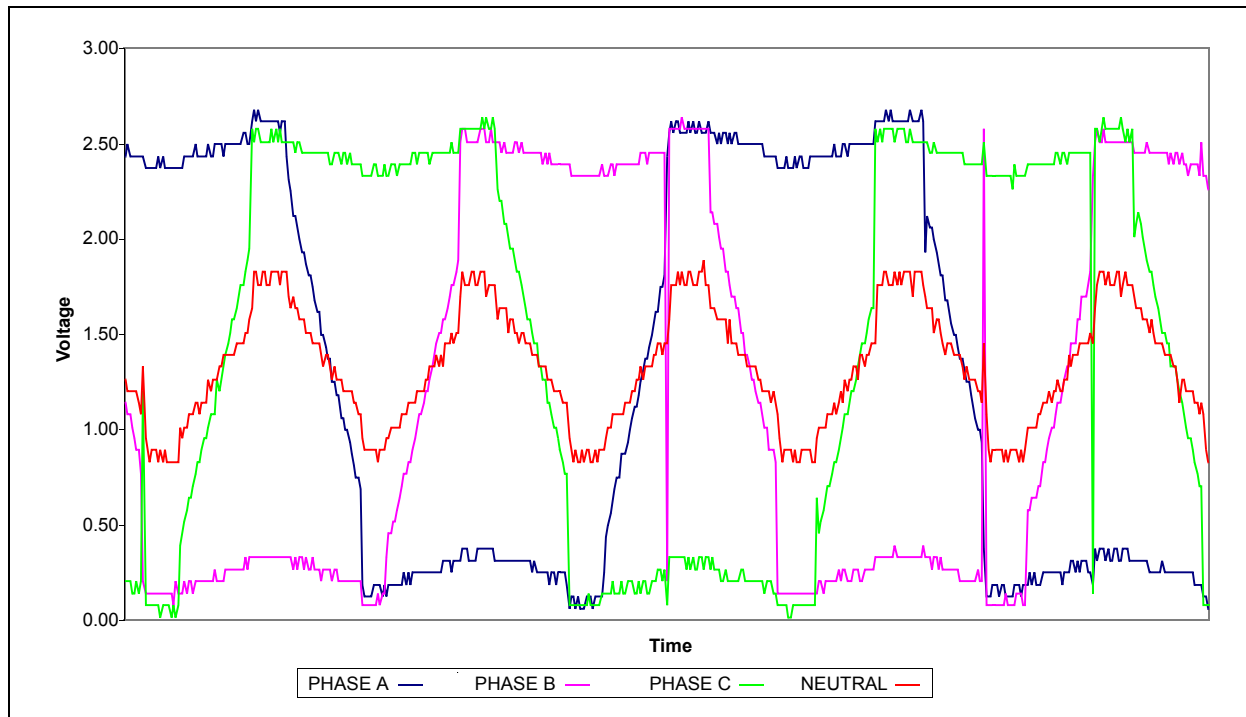
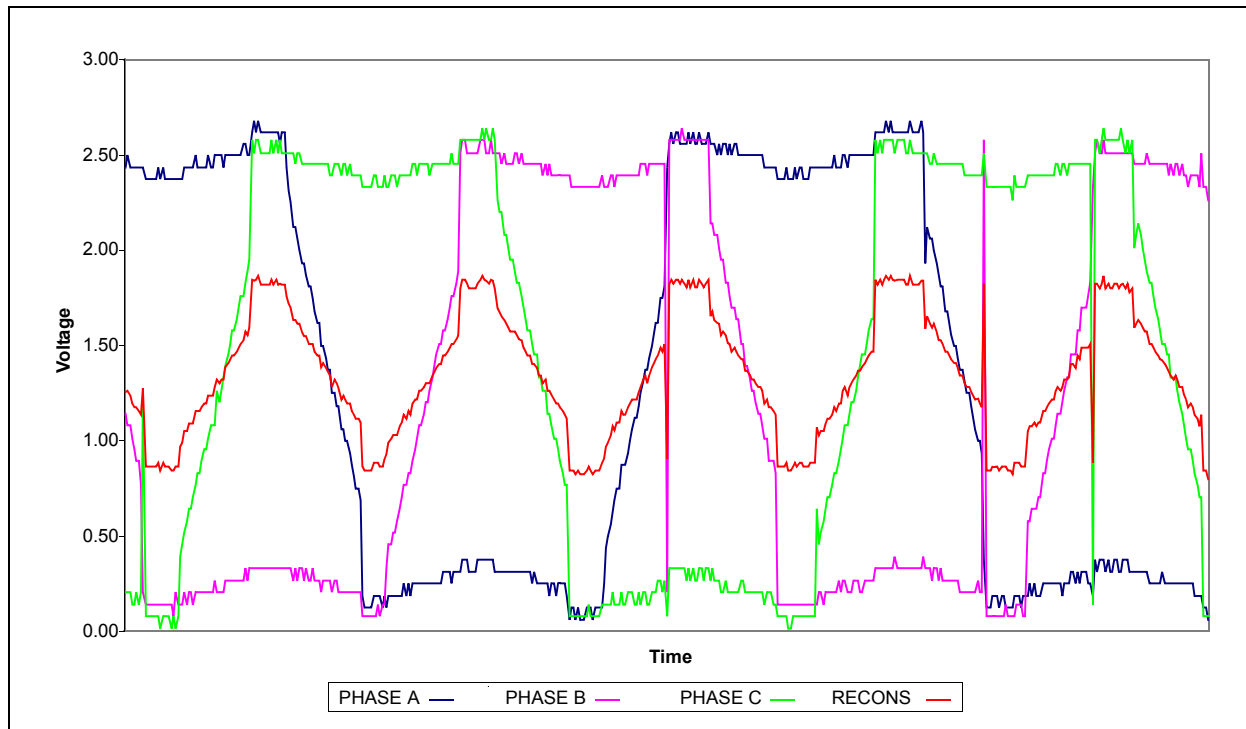


FIGURE 7: BEMF SIGNALS VERSUS RECONSTRUCTED VIRTUAL NEUTRAL POINT WHEN THE PWM DUTY CYCLE IS EQUAL TO 100%



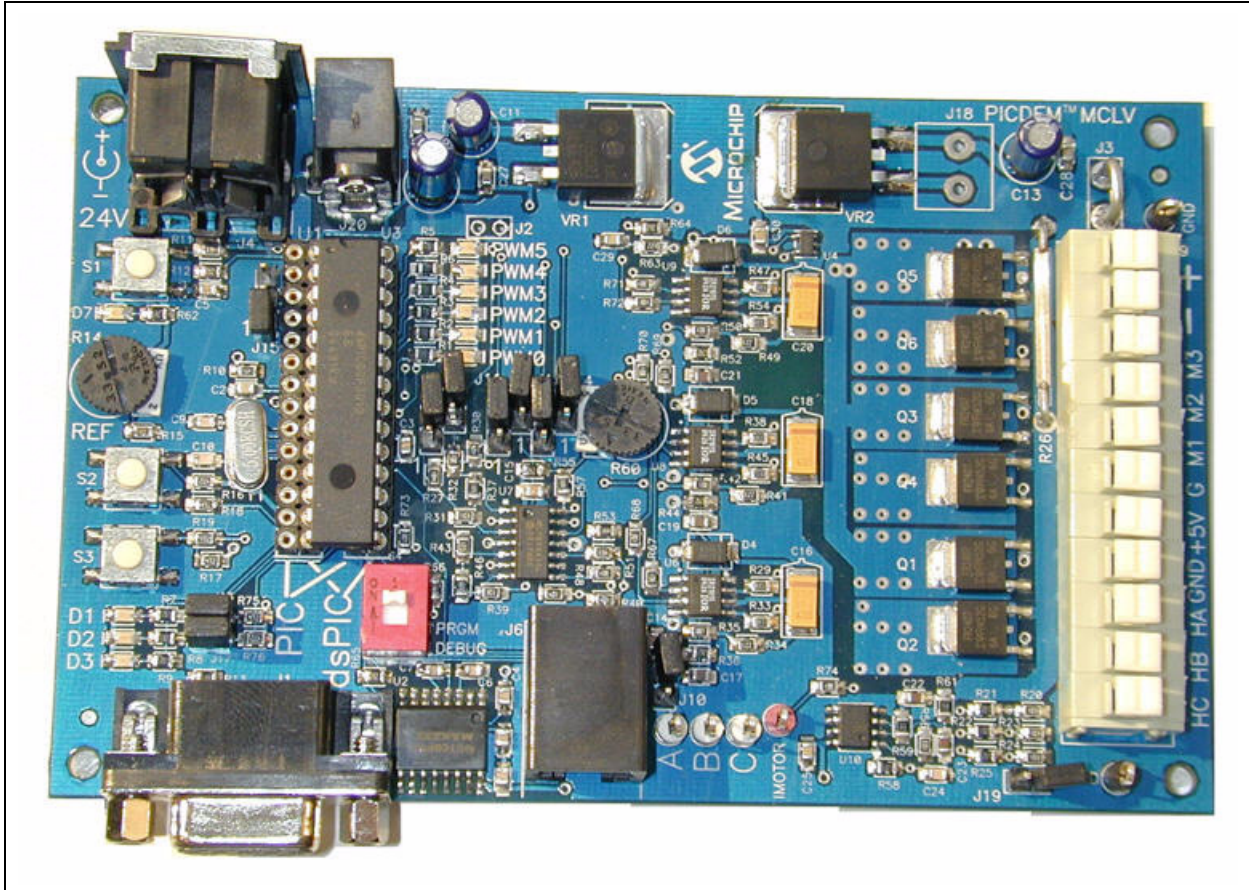
Required Hardware

The following hardware is required to run the BLDC motor control method described in this application note:

dsPIC30F SOFTWARE VERSION

- PICDEM™ MCLV Development Board (Figure 8)
- Hurst DMB0224C10002 CL B 6403 24V BLDC Motor
- 24 VDC Power Supply

FIGURE 8: PICDEM™ MCLV DEVELOPMENT BOARD



AN1160

dsPIC33F SOFTWARE VERSION

- dsPIC33FJ12MC202 PIM
- Explorer 16 Development Board
- Motor Control Interface PICtail™ Plus Daughter Board
- dsPICDEM™ MC1L 3-Phase Low Voltage Power Module
- Hurst DMB0224C10002 CL B 6403 24 V BLDC Motor
- 24 VDC Power Supply

FIGURE 9: HARDWARE CONNECTIONS FOR THE dsPIC33F SOFTWARE VERSION



These items can be purchased from Microchip as a complete kit or as individual components. Refer to the Development Tools section on the Microchip website for ordering information.

Hardware Modifications

The hardware block diagram previously shown in Figure 5 is a simplified block diagram of this motor control application. It shows the basic connections needed to get this algorithm work.

The required connections to get the hardware configuration shown in Figure 5 on the MCLV board are done through the jumper settings listed in Table 1.

TABLE 1: MCLV JUMPER CONFIGURATION

Jumper	Setting
J8	Open
J10	Open
J12	Open
J14	Open
J19	Open
J7	Short in position 2 - 3
J11	Short in position 2 - 3
J13	Short in position 2 - 3
J15	Short in position 2 - 3
J16	Short
J17	Short

These jumper settings used to configure the MCLV board are as follows:

- Pot R14 selects the demand for the speed. It is connected to the dsPIC DSC ADC channel AN4.
- BEMF signals are sensed using resistor networks R34/R36/R35, R41/R44/R42 and R49/R52/R50. The BEMF Phase A, B, and C signals are applied to the ADC channels AN3, AN4, and AN5 respectively.
- The capacitors C17, C19, and C21 used to filter the BEMF signals are disconnected since all the BEMF filtering is done by software.
- Fault input is received through a comparator circuit (U7D) connected with the current feedback circuit. The current is sensed using a 0.1Ω resistor (R26). The threshold point of the comparator can be adjusted using pot R60.

The required connections to achieve the dsPIC33FJ12MC202 hardware configuration shown in Figure 5 are listed below.

The default hardware configuration for the dsPIC33FJ12MC202 PIM must be modified to match the configuration shown in Table 2.

TABLE 2: PIM RESISTORS CONFIGURATION

Resistor	Setting
R29, R30, R8, R6, R20, R31, R27, R25, R9, R7, R5	Not populated
R15, R16, R17, R18, R19, R32, R33, R14, R10, R23, R22, R21, R28, R26, R24, R13, R11, R12	Populated

The default hardware configuration for the Explorer 16 Development Board must be modified to match the configuration shown in Table 3.

TABLE 3: EXPLORER 16 JUMPERS AND RESISTORS CONFIGURATION

Hardware modification	Setting
Jumper JP2	Short
Jumper J7	Short in position PIC24
Switch S2	Short in position PIM
Resistors R50, R51, R52	Not populated

The default hardware configuration for the Motor Control Interface PICTail™ Plus Daughter Board must be modified to match the configuration shown in Table 4.

TABLE 4: MOTOR CONTROL INTERFACE PICTail™ PLUS DAUGHTER BOARD JUMPER CONFIGURATION

Jumper	Setting
J12, J11, J10	Open
J1, J4	Open
J14, J15, J16	Open
J17	Short
J6, J7, J8	Open
J13	Open

The default hardware configuration of the 37W_DTYPE_PLUG LK resistors in the dsPICDEM™ MC1L 3-Phase Low Voltage Power Module must be configured as shown in Table 5.

Warning: Removing the top of the metal enclosure may result in electric shock. Wait at least 5 minutes after power is removed from the module before contacting the PCB in the enclosure. Follow the safety instructions described in **Section 1.6.2 “Accessing the System”** of the *dsPICDEM™ MC1L 3-Phase Low Voltage Power Module User’s Guide*, before removing the top of the metal enclosure.

TABLE 5: dsPICDEM™ MC1L 3-PHASE LOW VOLTAGE POWER MODULE RESISTORS CONFIGURATION

LK 51Ω resistors	Setting
LK22, LK24, LK25, LK26, LK30	Populated
LK19, LK20, LK21, LK23, LK27, LK28, LK29, LK31, LK32	Not populated

Digital Filter (Majority Function)

This BEMF sensing method is based on a nonlinear digital filter called ‘majority function’. In certain situations it is also known as ‘median operator’. The majority function is a Boolean function, which takes a number n of binary inputs and returns the value which is most common among them. For three Boolean inputs, it returns whichever value (true or false) occurs at least twice. In this case, two equal values represents 66% of the numbers. The majority function always returns the value of the majority ($> 50\%$) of the numbers. Table 6 shows an example of a 3-inputs majority function.

TABLE 6: AN EXAMPLE OF A MAJORITY FUNCTION USING THREE INPUTS

A	B	C	Majority
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

The majority of the values can be expressed using two logic operators, AND (\wedge) and OR (\vee) operators, as shown in Equation 6.

EQUATION 6: BOOLEAN REPRESENTATION OF THE MAJORITY FUNCTION

$$Majority = (A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$$

Implementing the Algorithm

In the earlier sections, it was pointed out that this BEMF method is based on the detection of zero-crossing events occurring on the BEMF signals. This section explains the implementation of this algorithm by the means of the dsPIC DSC device resources and peripherals.

SAMPLING THE BEMF SIGNALS

The first task to be performed is sampling the BEMF signals. To achieve such a task, the dsPIC DSC ADC is configured in such a way that it simultaneously samples the BEMF signals at a sampling rate equal to the PWM reload frequency; in this case it is 20 kHz. Therefore, the ADC is synchronized with the PWM reload event.

The dsPIC DSC ADC is also configured to take samples at the PWM ON time with the purpose of avoiding the ringing noise produced by the electronic switches and other noises such as the high voltage spikes produced by the winding de-energization event. These noises could create false zero-crossing events and therefore, a false commutation state.

The point in which the signals are sampled is variable across the PWM ON time depending on the motor speed. At low speeds, the dsPIC DSC device samples the BEMF signals at 50% of the PWM ON time. However, the sampling point moves forward according to the PWM duty cycle to reach the maximum point of 75% of the PWM ON time when the PWM duty cycle is equal to 100%. Figure 10 and Figure 11 illustrate the sampling points.

The motor neutral point is then reconstructed by software using the sampled BEMF signals A, B, and C. This reconstructed signal is compared against the sampled BEMF signals to identify the zero-crossing events. At this point, the external comparators have been emulated by software; the output of these software comparators are the binary representation of the sampled BEMF signals. The produced signals with these software comparisons still have some noise produced by the winding de-energization events, and the ringing noise produced by the electronic switches.

Sampling the BEMF signals at 20 kHz significantly reduces the switching noise on the sampled BEMF signals; consequently, it is easier to detect the zero crossing events. However, this aliasing trick is not enough to completely filter the BEMF signals. Therefore, the majority function filter is used.

FIGURE 10: BEMF SAMPLING POINTS AT 80% OF DUTY CYCLE

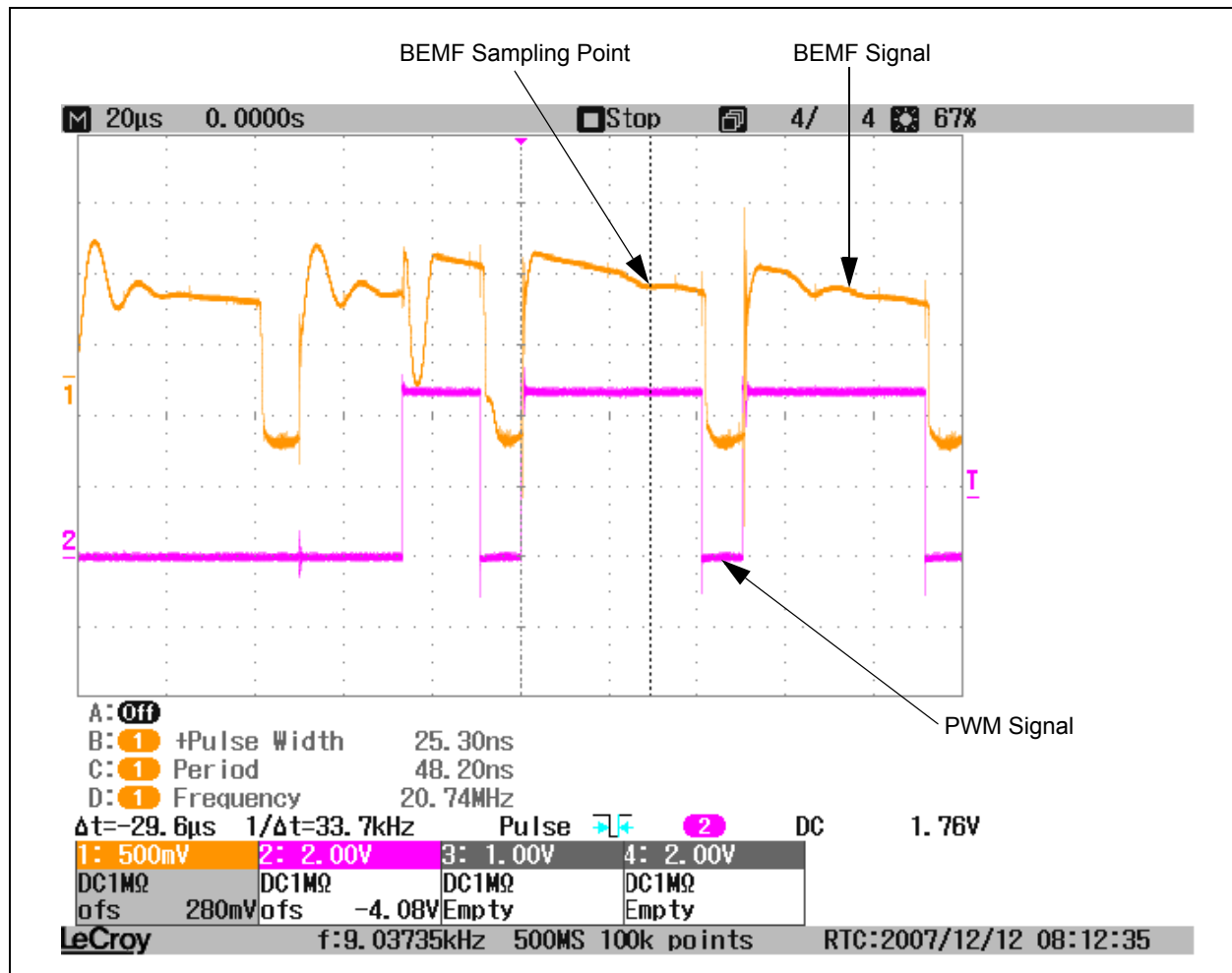
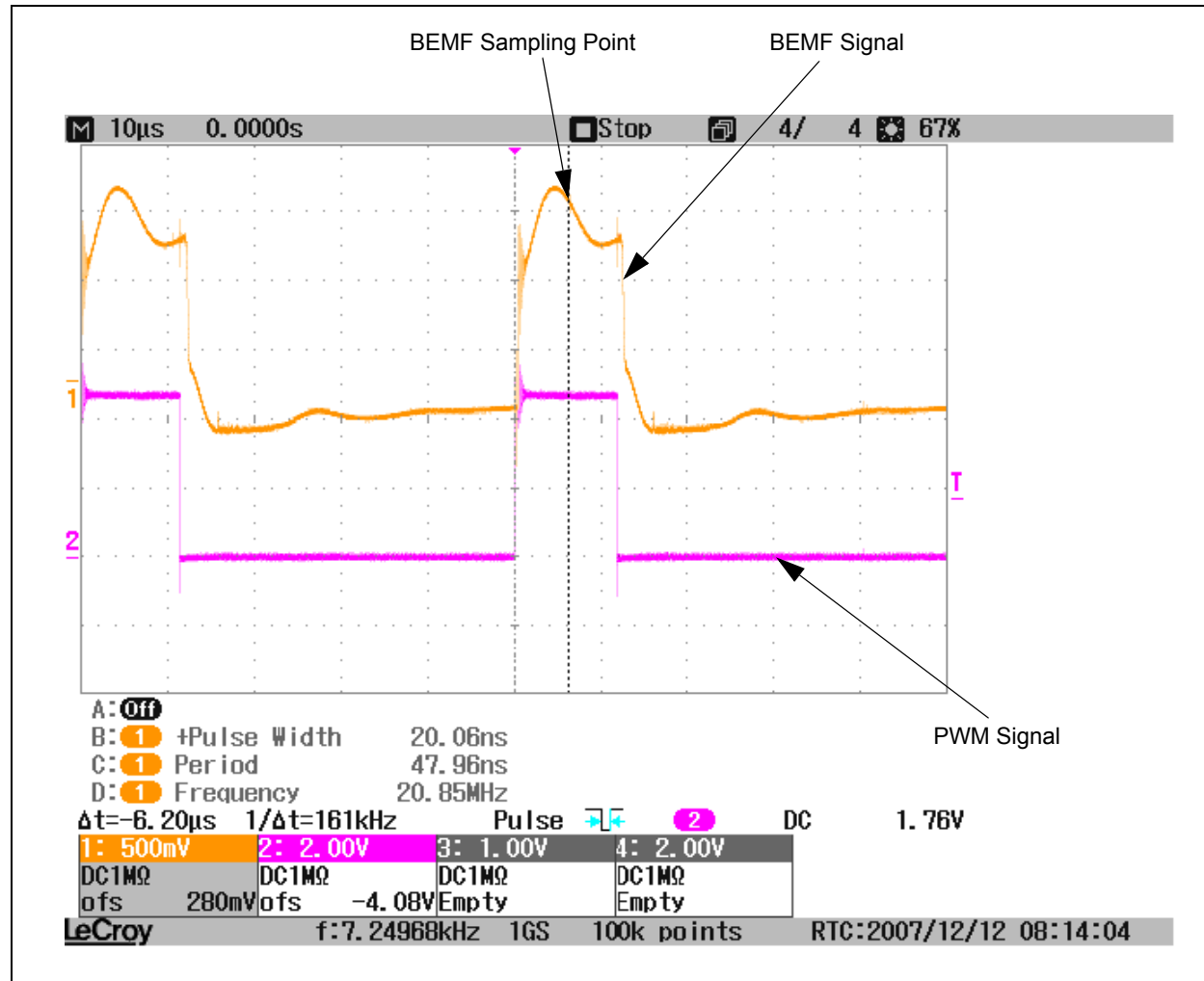


FIGURE 11: BEMF SAMPLING POINTS AT 20% OF DUTY CYCLE



FILTERING THE BEMF SIGNALS USING THE MAJORITY FUNCTION FILTER

The implementation of this nonlinear filter is based on a 6-samples window in which at least 51% of the three most significant samples should be equal to “1” and the three least significant samples should be equal to “0” for the purpose of identifying the occurrence of a zero-crossing event in the digitalized BEMF signals. This filtering step results in a more robust algorithm.

The first stage of the majority function filter is implemented using two logic operators, an AND operator for detecting the active BEMF signal correspondingly to the existing commutation state, and an Exclusive-OR (XOR) operator is used to detect the falling or rising edges on the active BEMF signal. The output of this logic operation is called “the active-masked BEMF signal” in the following sections.

The active-masked BEMF signal is then filtered using the majority detection filter. This filter is implemented with an array compounded by 64 values and a special logic test condition that is used to modify the pointer of the next data array. This logic test condition also identifies both the falling and rising edges of the active-masked BEMF signals; both edges are represented as a true-to-false event at the output of the logical test condition. The output of this condition is also used as an input to the majority detection filter.

The 64 values represent the 2^6 possible combinations that the 6-sample window could have for the active masked BEMF signal; each value on the look-up table is a pointer to the next signal state over time. The filter is always looking for a true-to-false change at the output of the logic test condition, if this true-to-false condition is detected, the filter looks for three consecutive false states to validate that a zero-crossing event occurred. A true-to-false condition at the output of the logic test represents a zero-crossing event and therefore a commutation on the motor, which occurs after a delay. This delay is equal to the timing of 30 electrical degrees minus the time required to execute the digital filtering. After the commutation a new BEMF signal is then monitored.

The 64 array values are determined as follows:

- The first 32 numbers are the index number multiplied by two, as shown in Equation 7

EQUATION 7: CALCULATING THE FIRST HALF OF THE ARRAY

$$\text{Array Value } [N] = N \cdot 2$$

- The last 32 values are filled out using Equation 8.

EQUATION 8: CALCULATING THE SECOND HALF OF THE ARRAY

$$\text{Array Value } [N] = (N - 32) \cdot 2$$

TABLE 7: ARRAY VALUES

Array Index [N]	Array Value	Array Index [N]	Array Value
0	0	32	0
1	2	33	2
2	4	34	4
3	6	35	6
4	8	36	8
5	10	37	10
6	12	38	12
7	14	39	14
8	16	40	16
9	18	41	18
10	20	42	20
11	22	43	22
12	24	44	24
13	26	45	26
14	28	46	28
15	30	47	30
16	32	48	32
17	34	49	34
18	36	50	36
19	38	51	38
20	40	52	40
21	42	53	42
22	44	54	44
23	46	55	46
24	48	56	48
25	50	57	50
26	52	58	52
27	54	59	54
28	56	60	56
29	58	61	58
30	60	62	60
31	62	63	62

There are 16 unique array index numbers that represent the true-to-false condition. These values are listed in order of appearance 24, 25, 26, 28, 40, 41, 42, 44, 48, 49, 50, 52, 56, 57, 58, 60. The values pointed by these unique indexes are replaced by “1” to indicate that a true-to-false condition occurred.

AN1160

The 16 unique index values are selected using the following majority function criteria. An array index number is a unique value when its binary representation contains a majority of “1” (> 50%) in the three most significant bits followed by a majority of “0” (> 50%) in the three least significant bits. Table 8 shows the 16 possible numbers that match these two conditions.

TABLE 8: 16 UNIQUE NUMBERS THAT NOTIFY A TRUE-TO-FALSE CONDITION IN THE ACTIVE MASKED BEMF

Number	6-bit Binary Representation
24	011000
25	011001
26	011010
28	011100
40	101000
41	101001
42	101010
44	101100
48	110000
49	110001
50	110010
52	110100
56	111000
57	111001
58	111010
60	111100

The 48 remaining array numbers are pointers to the unique values in case a true-to-false condition occurs. There are some values that never point to any of the unique values because they are not multiple of any of the 16 unique numbers. Table 9 provides some numbers that match this condition.

TABLE 9: NUMBERS THAT ARE A MULTIPLE OF A UNIQUE NUMBER

Number	6-bit Binary Rep.	Number of Times to be Right-Shifted	Unique Number to be Pointed	6-bit Binary Rep. of Unique Number
3	000011	3	24	011000
11	001011	3	24	011000
54	110110	1	44	101000
7	000111	2	28	011100

Those numbers (that never point to a 16 unique number) are then pointed to its multiple and they are trapped into a loop in such a way that the filter is waiting for a new value, which points to a unique number. Table 10 shows the numbers that are not multiple of a unique value.

TABLE 10: NUMBERS THAT NEVER POINT TO A UNIQUE VALUE

Number	6-bit Binary Rep.	Numbers Pointed Before Becoming Zero	Number of Times to be Right-Shifted
1	000001	2, 4, 8, 16, 32	5
9	001001	18, 36, 8, 16, 32	5
36	100100	8, 16, 32	3
17	010001	34, 4, 8, 16, 32	5

The complete filter coefficients are shown in Table 11.

TABLE 11: MAJORITY FILTER COEFFICIENTS

Array Index [N]	Array Value	Array (Unique Numbers)	Array Index [N]	Array Value	Array (Unique Numbers)
0	0	0	32	0	0
1	2	2	33	2	2
2	4	4	34	4	4
3	6	6	35	6	6
4	8	8	36	8	8
5	10	10	37	10	10
6	12	12	38	12	12
7	14	14	39	14	14
8	16	16	40	16	1
9	18	18	41	18	1
10	20	20	42	20	1
11	22	22	43	22	22
12	24	24	44	24	1
13	26	26	45	26	26
14	28	28	46	28	28
15	30	30	47	30	30
16	32	32	48	32	1
17	34	34	49	34	1
18	36	36	50	36	1
19	38	38	51	38	38
20	40	40	52	40	1
21	42	42	53	42	42
22	44	44	54	44	44
23	46	46	55	46	46
24	48	1	56	48	1
25	50	1	57	50	1
26	52	1	58	52	1
27	54	54	59	54	54
28	56	1	60	56	1
29	58	58	61	58	58
30	60	60	62	60	60
31	62	62	63	62	62

Table 12 shows an example of the complete filtering process. The inputs are the noiseless binary representation of the BEMF signals. Table 13 shows an example of the complete filtering process. In this case, the inputs are the noisy binary representation of the BEMF signals.

TABLE 12: EXAMPLE OF DIGITAL FILTERING COMPUTATIONS USING NOISELESS BEMF SIGNALS

ELECTRICAL ANGLE	BEMF Phase			XOR-Masked Phase			AND-Masked Phase			LOGICAL TEST CONDITION	COMMUTATION STEP	DIGITAL FILTER OUTPUT	ZERO-CROSSING EVENT	AND MASK	XOR MASK
	C	B	A	C	B	A	C	B	A						
0	1	1	0	0	0	0	0	0	0	0	0	0	FALSE	000	000
3	1	1	0	0	0	0	0	1	0	1	1	0	FALSE	010	000
6	1	1	0	0	0	0	0	1	0	1	1	2	FALSE	001	111
9	1	1	0	0	0	0	0	1	0	1	1	6	FALSE	100	000
12	1	1	0	0	0	0	0	1	0	1	1	14	FALSE	010	111
15	1	1	0	0	0	0	0	1	0	1	1	30	FALSE	001	000
18	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	100	111
21	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	000	000
24	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
27	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
30	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
33	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
36	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
39	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
42	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
45	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
48	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
51	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
54	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
57	1	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
60	1	0	0	0	0	0	0	1	0	0	1	62	FALSE	—	—
63	1	0	0	0	0	0	0	1	0	0	1	60	FALSE	—	—
66	1	0	0	0	0	0	0	1	0	0	1	1	FALSE	—	—
69	1	0	0	0	0	0	0	1	0	0	1	2	TRUE	—	—
72	1	0	0	1	1	1	0	0	1	1	2	4	FALSE	—	—
75	1	0	0	1	1	1	0	0	1	1	2	10	FALSE	—	—
78	1	0	0	1	1	1	0	0	1	1	2	22	FALSE	—	—
81	1	0	0	1	1	1	0	0	1	1	2	46	FALSE	—	—
84	1	0	0	1	1	1	0	0	1	1	2	30	FALSE	—	—
87	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—

TABLE 12: EXAMPLE OF DIGITAL FILTERING COMPUTATIONS USING NOISELESS BEMF SIGNALS (CONTINUED)

ELECTRICAL ANGLE	BEMF Phase			XOR-Masked Phase			AND-Masked Phase			LOGICAL TEST CONDITION	COMMUTATION STEP	DIGITAL FILTER OUTPUT	ZERO-CROSSING EVENT	AND MASK	XOR MASK
	C	B	A	C	B	A	C	B	A						
90	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
93	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
96	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
99	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
102	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
105	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
108	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
111	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
114	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
117	1	0	0	1	1	1	0	0	1	1	2	62	FALSE	—	—
120	1	0	1	1	1	1	0	0	1	0	2	62	FALSE	—	—
123	1	0	1	1	1	1	0	0	1	0	2	60	FALSE	—	—
126	1	0	1	1	1	1	0	0	1	0	2	1	FALSE	—	—
129	1	0	1	1	1	1	0	0	1	0	2	2	TRUE	—	—
132	1	0	1	0	0	0	1	0	0	1	3	4	FALSE	—	—

TABLE 13: EXAMPLE OF DIGITAL FILTERING COMPUTATIONS USING NOISY BEMF SIGNALS

ELECTRICAL ANGLE	BEMF Phase			XOR-Masked Phase			AND-Masked Phase			LOGICAL TEST CONDITION	COMMUTATION STEP	DIGITAL FILTER OUTPUT	ZERO-CROSSING EVENT	AND MASK	XOR MASK
	C	B	A	C	B	A	C	B	A						
0	1	1	0	0	0	0	0	0	0	0	0	0	FALSE	000	000
3	1	1	0	0	0	0	0	1	0	1	1	0	FALSE	010	000
6	1	0	1	0	0	0	0	1	0	0	1	2	FALSE	001	111
9	1	1	0	0	0	0	0	1	0	1	1	4	FALSE	100	000
12	1	1	0	0	0	0	0	1	0	1	1	10	FALSE	010	111
15	0	1	1	0	0	0	0	1	0	1	1	22	FALSE	001	000
18	1	1	0	0	0	0	0	1	0	1	1	46	FALSE	100	111
21	1	0	0	0	0	0	0	1	0	0	1	1	FALSE	000	000
24	1	1	0	0	0	0	0	1	0	1	1	2	FALSE	—	—
27	1	1	0	0	0	0	0	1	0	1	1	6	FALSE	—	—
30	1	1	0	0	0	0	0	1	0	1	1	14	FALSE	—	—
33	1	1	1	0	0	0	0	1	0	1	1	30	FALSE	—	—
36	0	1	0	0	0	0	0	1	0	1	1	62	FALSE	—	—
39	1	1	0	0	0	0	0	1	0	1	1	1	FALSE	—	—
42	1	1	0	0	0	0	0	1	0	1	1	2	FALSE	—	—
45	1	0	0	0	0	0	0	1	0	0	1	6	FALSE	—	—
48	1	1	0	0	0	0	0	1	0	1	1	12	FALSE	—	—
51	1	1	0	0	0	0	0	1	0	1	1	26	FALSE	—	—
54	1	1	0	0	0	0	0	1	0	1	1	54	FALSE	—	—
57	1	1	0	0	0	0	0	1	0	1	1	1	FALSE	—	—
60	1	0	0	0	0	0	0	1	0	0	1	2	TRUE	—	—
63	1	1	0	1	1	1	0	0	1	1	2	4	FALSE	—	—
66	0	0	0	1	1	1	0	0	1	1	2	10	FALSE	—	—
69	1	1	1	1	1	1	0	0	1	0	2	22	FALSE	—	—
72	1	1	0	1	1	1	0	0	1	1	2	44	FALSE	—	—
75	0	0	0	1	1	1	0	0	1	1	2	1	FALSE	—	—
78	1	0	1	1	1	1	0	0	1	0	2	2	FALSE	—	—
81	1	0	0	1	1	1	0	0	1	1	2	4	FALSE	—	—
84	0	1	0	1	1	1	0	0	1	1	2	10	FALSE	—	—
87	1	0	1	1	1	1	0	0	1	0	2	22	FALSE	—	—
90	0	1	0	1	1	1	0	0	1	1	2	44	FALSE	—	—
93	1	0	0	1	1	1	0	0	1	1	2	1	FALSE	—	—
96	1	0	1	1	1	1	0	0	1	0	2	2	FALSE		
99	1	1	0	1	1	1	0	0	1	1	2	4	FALSE	—	—
102	1	0	0	1	1	1	0	0	1	1	2	10	FALSE	—	—

TABLE 13: EXAMPLE OF DIGITAL FILTERING COMPUTATIONS USING NOISY BEMF SIGNALS (CONTINUED)

ELECTRICAL ANGLE	BEMF Phase			XOR-Masked Phase			AND-Masked Phase			LOGICAL TEST CONDITION	COMMUTATION STEP	DIGITAL FILTER OUTPUT	ZERO-CROSSING EVENT	AND MASK	XOR MASK
	C	B	A	C	B	A	C	B	A						
105	1	0	0	1	1	1	0	0	1	1	2	22	FALSE	—	—
108	1	1	1	1	1	1	0	0	1	0	2	46	FALSE	—	—
111	1	0	0	1	1	1	0	0	1	1	2	1	FALSE	—	—
114	1	1	0	1	1	1	0	0	1	1	2	2	FALSE	—	—
117	1	0	0	1	1	1	0	0	1	1	2	6	FALSE	—	—
120	1	0	1	1	1	1	0	0	1	0	2	14	FALSE	—	—
123	1	0	1	1	1	1	0	0	1	0	2	28	FALSE	—	—
126	1	0	1	1	1	1	0	0	1	0	2	1	FALSE	—	—
129	1	0	1	1	1	1	0	0	1	0	2	2	TRUE	—	—
132	1	0	1	0	0	0	1	0	0	1	3	4	FALSE	—	—

These computation examples recall the fact that this binary representation of the BEMF signals is generated after the comparison of the BEMF sampled signals to the Virtual Neutral Point. These two tables do not represent the 20 KHz sampling frequency, therefore the samples taken for this example were randomly sampled using a random sampling frequency.

When the zero-crossing event is detected, the dsPIC DSC device commutates after a delay the voltage drives according to the six-step commutation step. To keep the magnetic field in the stator advancing ahead of the rotor, the transition from one sector to another must occur at precise rotor positions for optimal torque.

This commutation delay is equal to the timing of 30 electrical degrees minus the time required to execute the digital filtering process. To determine the commutation delay, one of dsPIC DSC general purpose on-chip timers is used to measure the amount of time elapsed from one zero-cross event to the next. This time is equivalent to 60 electrical degrees.

Assuming there is no phase delay when a zero-cross event is detected, the next commutation should occur in 30 degrees. Dividing the timer capture value by two gives the time for 30 electrical degrees. This value is then loaded into another timer's period register for generating the commutation delay; this timer is also referred to as the commutation timer. When the interrupt for the commutation timer occurs, it is time to commutate the motor windings to the next state.

START-UP SEQUENCE

The motor start-up sequence is compounded by two stages.

- Applying 1024 pulses during 1 millisecond in a predefined commutation sequence to identify the rotor position.
- Then the motor is spun in Open Loop mode by applying the correct commutation step with the required minimum duty cycle to break the motor idle state.

For the HURST motor, the required PWM duty cycle to break the standstill inertia is 7.5%. Once the motor starts to spin, the BEMF signals are detected and the algorithm can run in either Open Loop mode or Closed Loop mode.

CONTROL LOOPS

An interesting property of BLDC motors is that they will operate synchronously to a certain extent. This means that for a given load, applied voltage, and commutation rate, the motor will maintain open loop lock with the commutation rate, provided that these three variables do not deviate from the ideal by a significant amount. The ideal is determined by the motor voltage and torque constants.

Consider a case in which the commutation rate is too slow for an applied voltage, the BEMF will be too low resulting in more motor current. The motor will react by accelerating to the next phase position, and then slows down waiting for the next commutation.

In the opposite case, the motor will snap to each position like a stepper motor until the next commutation occurs. Since the motor is able to accelerate faster than the commutation rate, rates much slower than the ideal can be tolerated without losing lock, but at the expense of excessive current.

If the commutation arrives so early that the motor can not accelerate fast enough to catch the next commutation, lock is lost and the motor spins down. This happens abruptly not very far from the ideal rate. The abrupt loss of lock looks like a discontinuity in the motor response which makes closed loop control difficult. An alternative to closed loop control is to adjust the commutation rate until self-locking open loop control is achieved.

This application software has two control modes that can be selected for use during sensorless operation. These modes are as follows:

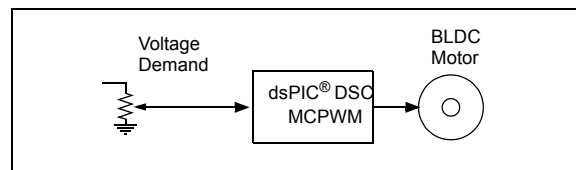
- Open Loop
- Closed Loop

Open Loop Mode

When the load on a motor is constant over its operating range, the response curve of motor speed relative to applied voltage is linear. If the supply voltage is well regulated, in addition to a constant torque load, the motor can be operated open loop over its entire speed range.

Assume that with pulse width modulation the effective voltage is linearly proportional to the PWM duty cycle. An open loop controller can be made by linking the PWM duty cycle to a 16-bit variable. The value of this 16-bit variable can be set by a potentiometer using an ADC channel to sense the voltage present across the potentiometer terminals. This block diagram of this mode is shown in Figure 12.

FIGURE 12: OPEN LOOP CONTROL



The Analog-to-Digital conversion value is delivered in a 10-bit unsigned integer format; therefore, the possible conversion values are within the range of 0 – 1024. It is required to scale this conversion value to match the PWM duty cycle range (Equation 9). In case of this application, the PWM duty cycle value varies from 0 to 1473.

EQUATION 9: CALCULATING THE PWM DUTY CYCLE RANGE

$$PWM \text{ Duty Cycle Range} = \frac{F_{CY}}{F_{PWM}} - 1$$

where

F_{CY} is the system frequency, approx. 29.4 MHz

F_{PWM} is the desired PWM frequency, which for this application is 20 kHz

After initializing the MCPWM, ADC, Ports and Timer3, the program waits for an activation signal (for example, a key press) to start spinning the motor (see Figure 14). When the key is pressed, the start-up sequence is executed and the BEMF signals and Pot value are sampled and then filtered.

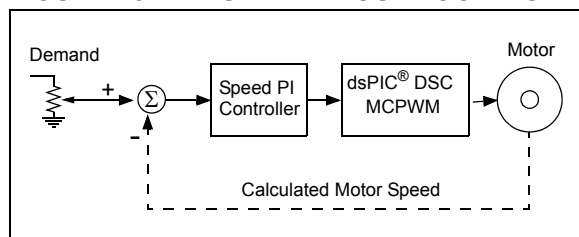
Based on the zero-crossing events and the current commutation state, a corresponding value is retrieved from the table and written to the PWM OVDCON register to set the electronic switches to the next commutation step after the commutation delay. The program flow is shown in Figure 16 and Figure 17.

Initially, the duty cycle value is held at a default 7.5%. On the very next ADC interrupt service routine, however, the potentiometer is read and scaled. Its value is inserted as the duty cycle when it is copied to the PWM PDCx registers. This action determines the speed of the motor. The higher the PDCx register value the faster the motor will spin.

Closed Loop Mode

In the Closed Loop mode a speed control loop is used to control the PWM duty cycle delivered to the motor. The speed demand is determined by the potentiometer value, which is scaled to achieve the desired speed range. The speed controller is implemented through a PI controller to compensate the error between the speed demand and the calculated motor speed. Figure 13 shows the block diagram of the speed Closed Loop mode.

FIGURE 13: SPEED CLOSE LOOP MODE



According to Equation 1, Equation 2 and Equation 3, it is possible to determine the motor speed if we know the number of pair poles and the electrical revolutions per second. Recalling the fact that one electrical revolution per second is equal to one six-step commutation cycle, the mechanical revolutions per second are directly related to the number of six-step commutation cycles.

For a 2-pair poles motor (or 4 motor poles), it is required to execute twice the six-step commutation cycle to achieve a complete mechanical revolution. Therefore, it is possible to measure the mechanical revolutions per second through counting the number six-step commutation cycles and then comparing them to the number of the motor pole pairs.

To measure the mechanical revolutions per second Timer3 is configured to operate in Free-Running Up-Counting mode, having as a time base the system clock frequency (approximately 29.4 MHz) divided by 256 (8.68 μ s). With this time base, Timer3 is able to count from 8.68 μ s to 568.84 ms. Therefore, it is possible to measure low speeds and high speeds with enough resolution to accurately determine the motor speed.

Timer3 is then triggered every time the N six-step commutation cycle is completed. The N factor is in fact the number of pole pairs. Therefore, Timer3 is triggered every one mechanical revolution per second.

Once the current speed is calculated, it is then compared to the desired speed set by the scaled value of the pot. The proportional and integral error between the desired speed and the current speed is calculated and then multiplied by the PI constants, as shown in Equation 10.

EQUATION 10: PI CONTROLLER COMPUTATIONS

$$Speed \text{ Error} = Desired \text{ Speed} - Current \text{ Speed}$$

$$Integral \text{ Error} = Integral \text{ Error} + Speed \text{ Error}$$

$$PI \text{ output} = (k_p) \{Speed \text{ Error}\} + (K_i) \{Integral \text{ Error}\}$$

The PI output is then scaled to match the range of the PWM duty cycle. The Figure 14, Figure 15 and Figure 16 illustrate the complete software flow for both the Open Loop mode and Closed Loop mode.

FIGURE 14: MAIN ROUTINE

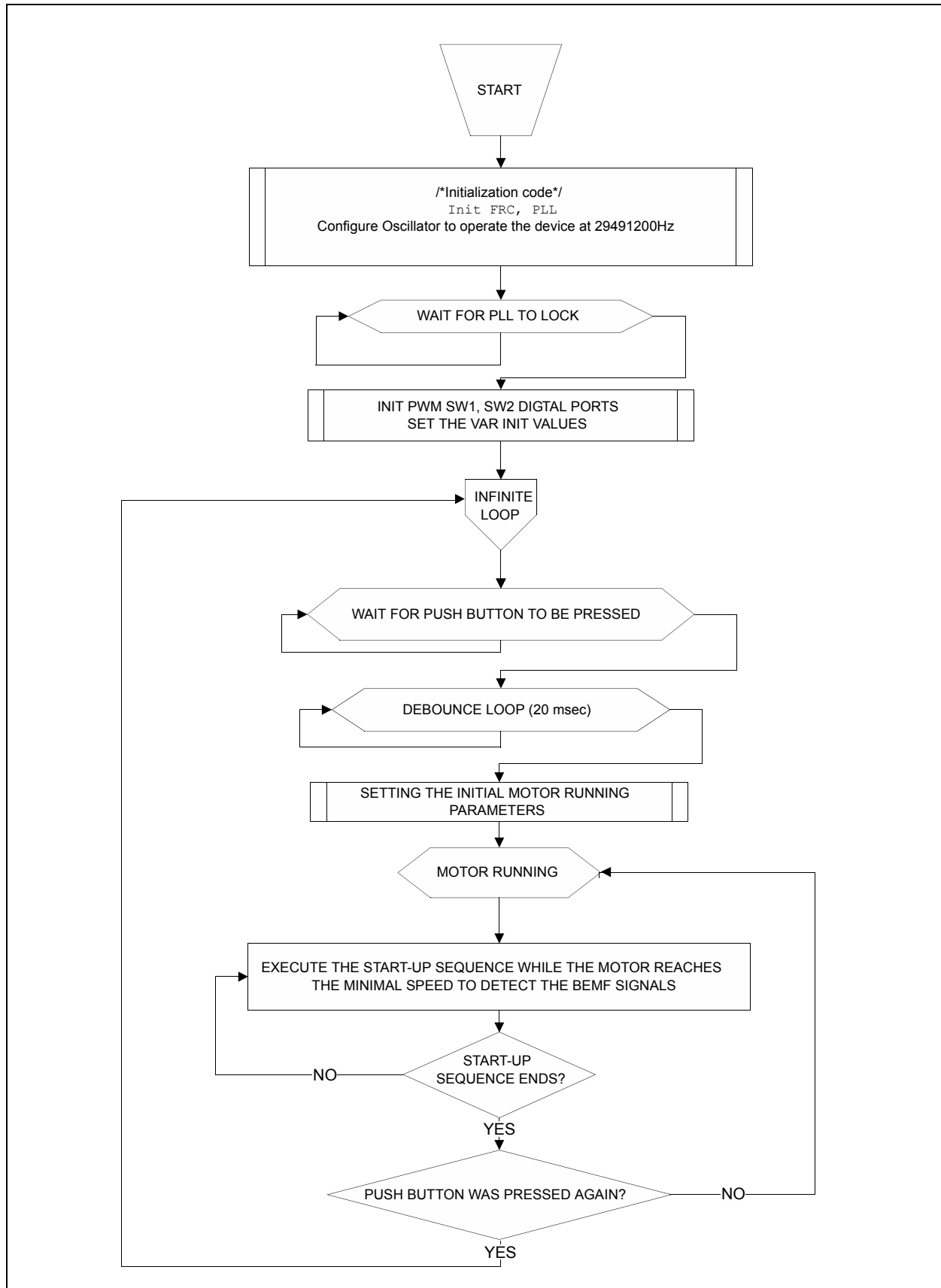


FIGURE 15: ADC INTERRUPT SERVICE ROUTINE

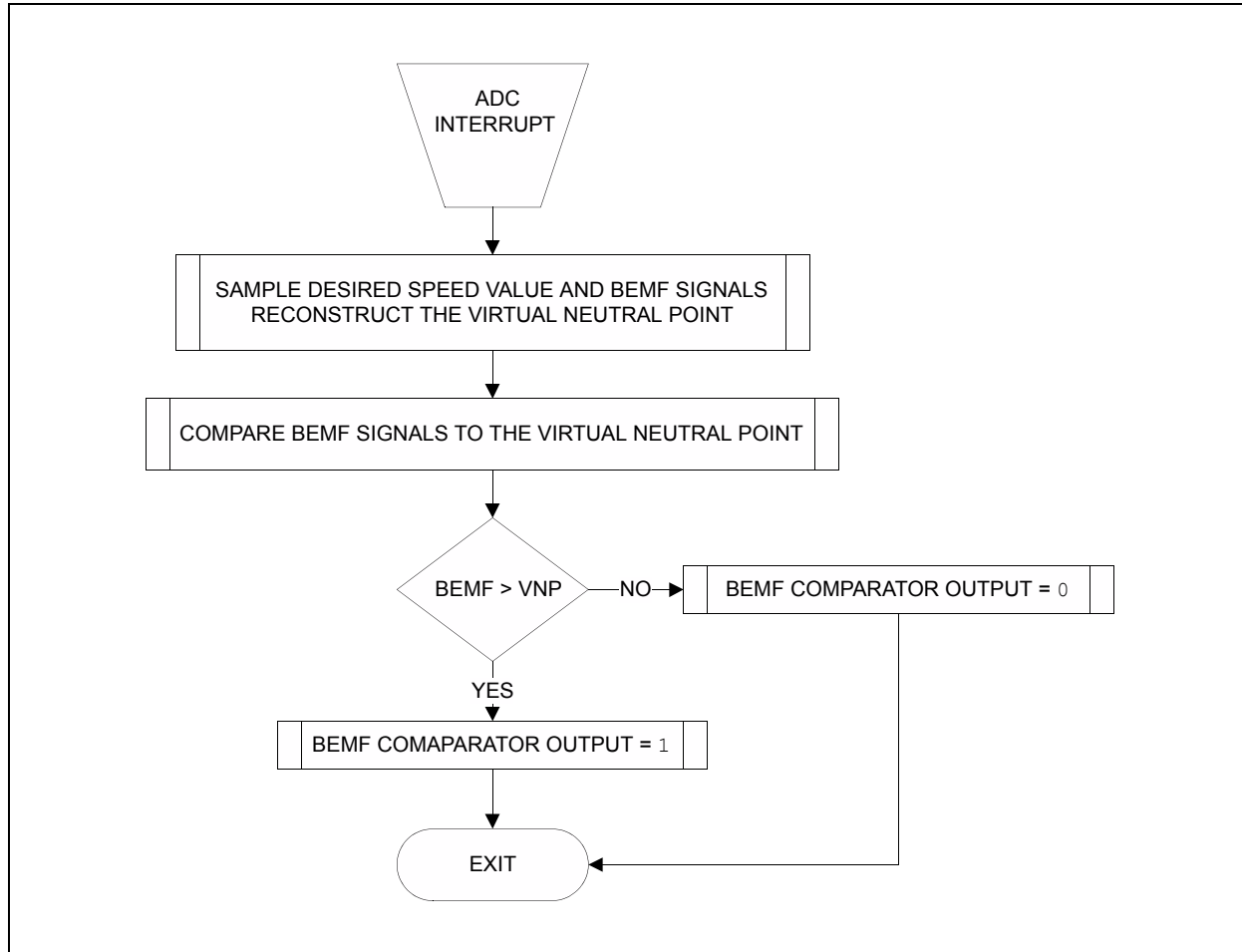


FIGURE 16: PWM INTERRUPT SERVICE ROUTINE

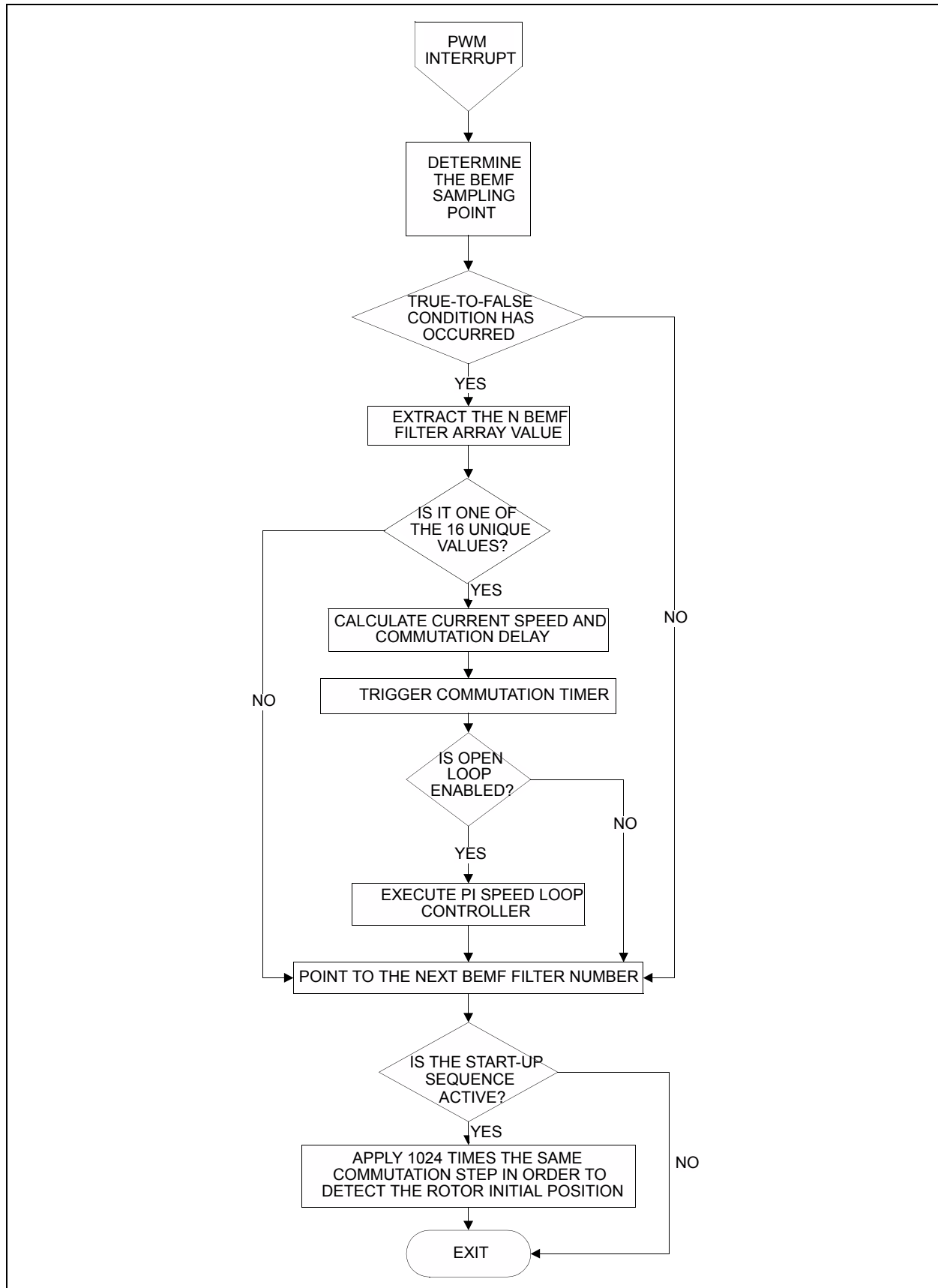
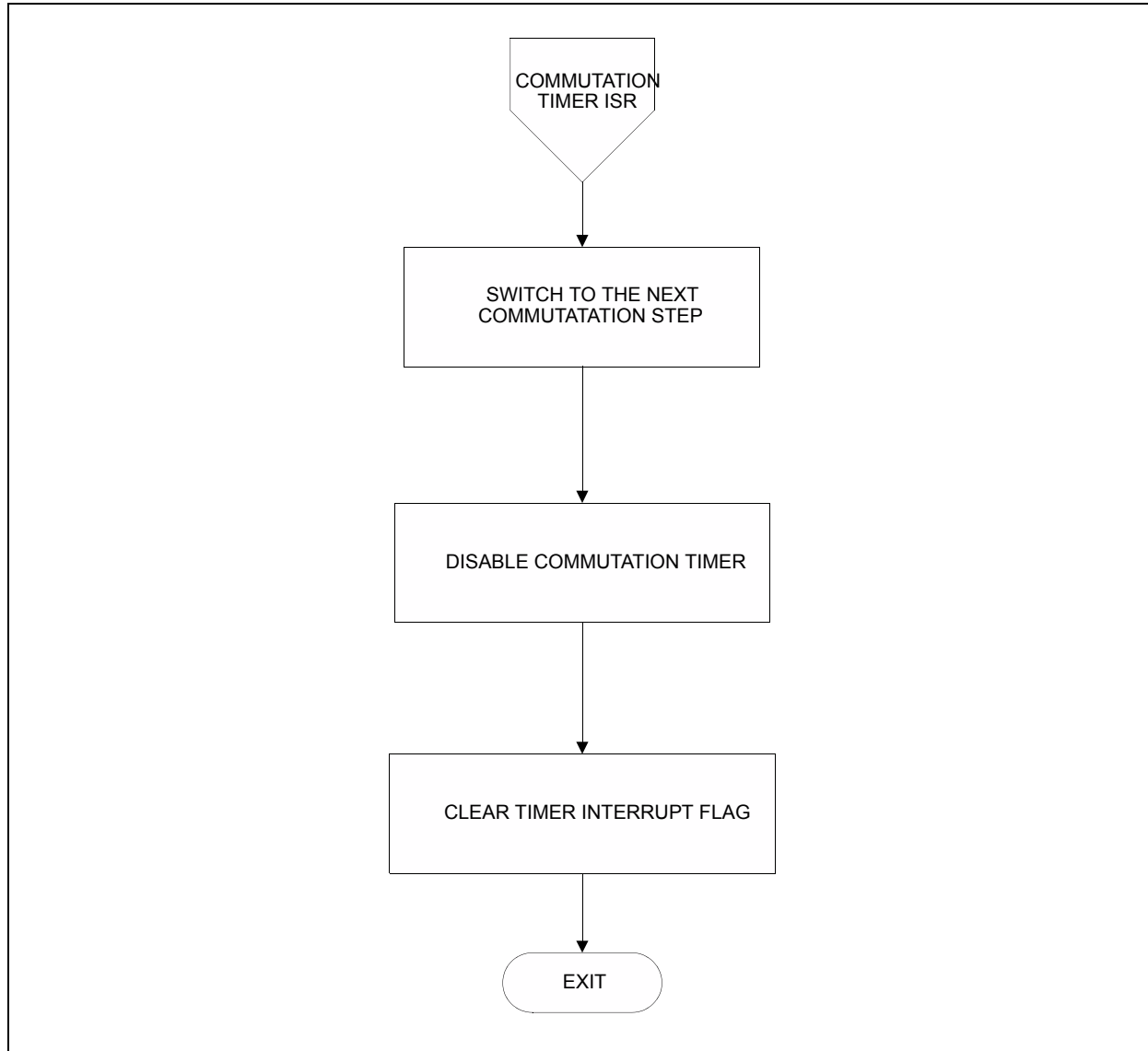


FIGURE 17: COMMUTATION TIMER INTERRUPT SERVICE ROUTINE



CONCLUSION

This application note is intended for the developer who wants to drive a sensorless BLDC motor using this new BLCD control technique in a basic and simple form, without the use of discrete, low-pass filtering hardware and off-chip comparators.

It also shows that this new control method is a single-chip dsPIC DSC device-based solution, which does not require external hardware, except for a couple of resistors used to conditioning the BEMF signals to the dsPIC DSC ADC operational voltage range. The algorithm described uses nonlinear digital filtering based on a majority detection function to sense the back-EMF signals generated by a rotating BLDC motor.

It is important to mention that this method is quite different from the ones described in previous Microchip application notes AN901, AN992 or AN1083 (see “References”). The most important differences between the method explained in this application note and the methods described in the previous application notes are presented below.

The ADC in AN1083 is used to sample the BEMF at a very high sampling frequency. These samples are then processed by an IIR filter to reconstruct the BEMF signals and the threshold point used to identify a zero-crossing event is $V_{BUS}/2$. It also uses timers to identify the commutation time due to the delay caused by the IIR filtering process. At low speeds it filters the three BEMF signals and one timer is used to ensure that the six-commutation steps occur on time. At high speeds it filters only one BEMF signal and uses three timers to ensure that the six-commutation steps occurs on time.

AN901 samples at the PWM OFF time, but the polarity of the PWM signals is inverted so the result is “similar to” sampling at the PWM ON time. In this case, the signals are sampled at a fixed point; which is close to the end of the PWM OFF time. The sampled signals are then compared against the $V_{BUS}/2$ to determine the zero-crossing events. This is achieved by means of an n -sample window. The samples enclosed in the n -sampling window are compared against the $V_{BUS}/2$. If the first half samples are lower than the $V_{BUS}/2$ and the second half samples are higher than the $V_{BUS}/2$, a true zero-crossing event has occurred.

Digital filtering makes it possible to more accurately detect the zero-cross events in the back-EMF signal. When detected by the dsPIC DSC device, zero-cross events provide the information needed by the algorithm to commutate the motor windings.

Accurately detecting the zero-cross events in a back-EMF signal is the key to sensorless control of a BLDC motor that is driven using six-step, or trapezoidal, commutation. The use of digital filtering, as

opposed to hardware filters or external comparators, requires less hardware, which equates to less cost and a smaller PCB.

REFERENCES

- Valiant, L. (1984), “*Short Monotone Formulae for the Majority Function*”, Journal of Algorithms 5: 363–366.
- “*Modern Power Electronics and AC Drives*”, B. Bose, Prentice Hall PTR, ISBN 0130167436
- “*Electric Motors and Drives*”, A. Hughes, Heinemann Newnes, ISBN 0750617411
- “*Brushless Permanent Magnet and Reluctance Motor Drives*”, T. Miller, Oxford Clarendon, ISBN 0198593694
- K. Iizuka et. al., “*Microcomputer Control for Sensorless Brushless Motor*”, IEEE Transactions on Industrial Applications, Vol. 21, No.4 1985, pp 595-601
- AN857, “*Brushless DC Motor Control Made Easy*”, Microchip Technology Inc., 2002
- AN901, “*Using the dsPIC30F for Sensorless BLDC Control*”, Microchip Technology Inc., 2007
- AN957, “*Sensored BLDC Motor Control Using dsPIC30F2010*”, Microchip Technology Inc., 2005
- AN970, “*Using the PIC18F2431 for Sensorless Motor Control*”, Microchip Technology Inc., 2005
- AN992, “*Sensorless BLDC Motor Control Using dsPIC30F2010*”, Microchip Technology Inc., 2005
- AN1017, “*Sinusoidal Control of PMSM Motors with dsPIC30F DSC*”, Microchip Technology Inc., 2005
- AN1078, “*Sensorless Field Oriented Control of PMSM Motors*”, Microchip Technology Inc., 2007

AN1160

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, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Linear Active Thermistor, MXDEV, MXLAB, 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, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, 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.

© 2008, 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 design centers in California and India. 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 - Hong Kong SAR

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

China - Nanjing

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

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 - Wuhan

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

China - Xiamen

Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai

Tel: 86-756-3210040
Fax: 86-756-3210049

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 - Daegu

Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul

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

Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang

Tel: 60-4-227-8870
Fax: 60-4-227-4068

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