AN1160

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

Author: Adrian Lita, Mihai Cheles and Aldrin Abacan
Microchip Technology Inc.

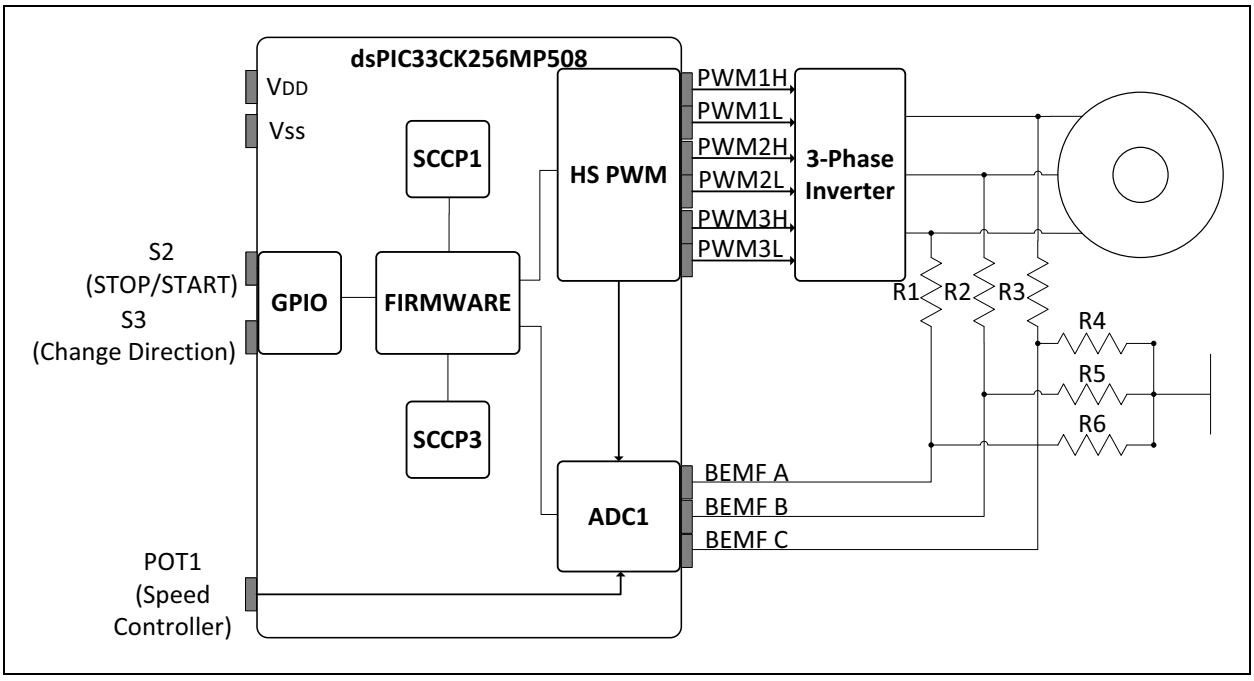
INTRODUCTION

This application note describes a sensorless Brushless Direct Current (BLDC) motor control algorithm that is implemented using a dsPIC® Digital Signal Controller (DSC). The algorithm works utilizing a majority filter function as the digital filter for 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 off-chip comparators and complicated Back-EMF sensing configurations. The algorithm that is discussed in this application note is implemented in a three-phase BLDC motor system. The motor control algorithm described here has five main parts:

- Sampling trapezoidal BEMF signals using the microcontroller's Analog-to-Digital Converter (ADC)
- PWM on-time ADC sampling to reduce noise and solve low-inductance problems
- Comparing the trapezoidal BEMF signals to a calculated virtual neutral point from the BEMF signals to detect the zero-crossing points
- Filtering the signals coming from the comparisons using a majority function filter
- Commutate the motor driving voltages in either of open or closed-loop control

This motor control technique, using a single-chip, 16-bit dsPIC DSC device, is introduced to minimize external hardware requirements for sensorless operation. Only a few resistors are added, reducing the BEMF signal amplitude to the range of the device's ADC module. Figure 1 shows the system block diagram of the sensorless BLDC system using a dsPIC33CK device.

FIGURE 1: SYSTEM BLOCK DIAGRAM



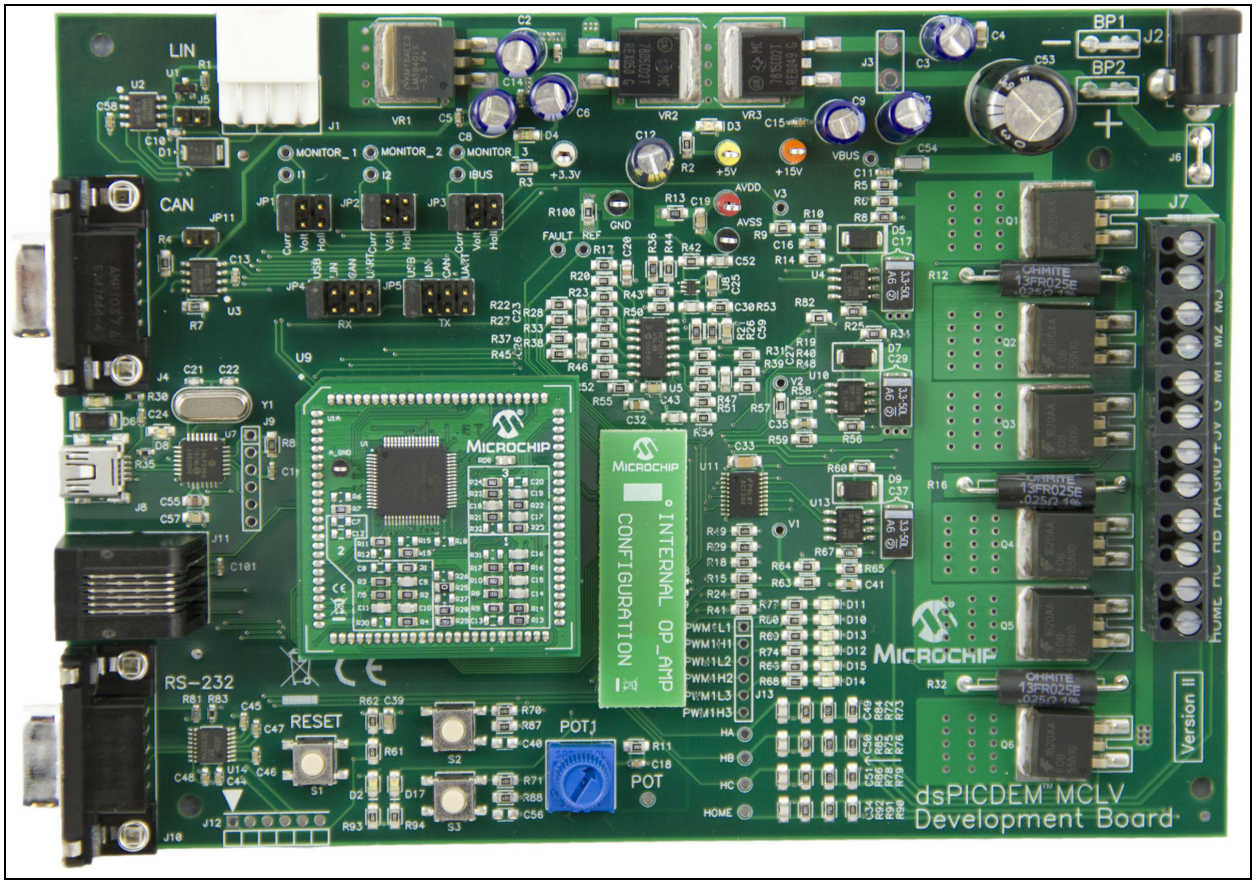
HARDWARE REQUIREMENT

The following hardware is used to implement the motor control application with the dsPIC33CK device:

- dsPICDEM™ MCLV-2 Development Board (DM330021-2) (Figure 2)
- dsPIC33CK256MP508 External Op Amp Motor Control PIM (MA330041-1)
- Hurst, 24V Three-Phase Brushless DC Motor – AC300020
- 24 VDC Power Supply

This motor control application is also compatible with Microchip's high-voltage motor control products, which can also be purchased online. Check the Development Tools section of MicrochipDirect for more information. For the complete setup of this project, refer to **Appendix A: "Hardware Setup"**.

FIGURE 2: MCLV-2 DEVELOPMENT BOARD



BLDC SENSORLESS CONTROL

The use of BLDC motors, due to their compact size, controllability and high efficiency, has increased in the past several years. It has become a staple in the industry because of its ability to cater to a wide range of application requirements, such as varying loads, constant torque, varying speed, etc., in the field of industrial control, automotive or aerospace applications. It eliminates belts and hydraulic systems to provide additional functionality, and to improve fuel economy, while reducing maintenance costs to zero.

Since the electrical excitation must be synchronous to the rotor position, the BLDC motor is usually operated with one or more rotor position sensors. 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 BEMF voltage on an undriven motor terminal during one of the drive phases. There are some disadvantages to sensorless control, such as:

- The motor must be moving at a minimum rate to generate sufficient BEMF 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

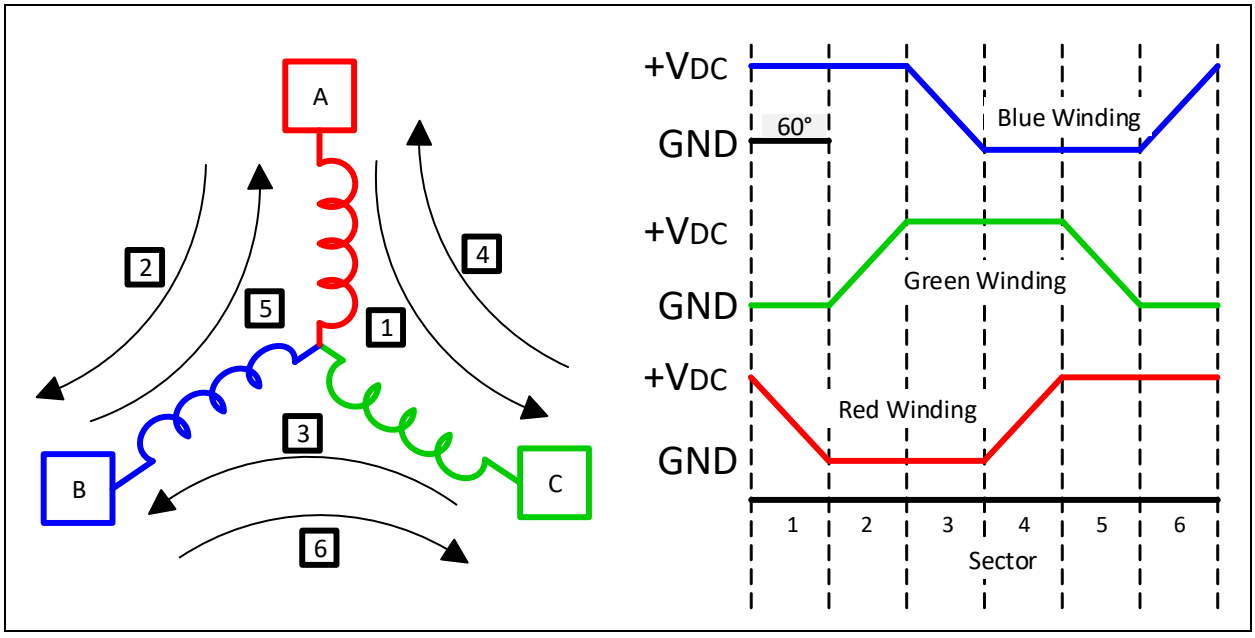
However, there are specific algorithms to overcome all the disadvantages listed. Sensorless trapezoidal control could be a better choice for your application if low cost is a primary concern, low-speed motor operation is not a requirement and the motor load is not expected to change rapidly. The BEMF zero-crossing technique described here is recommended for several reasons:

- It is suitable for use on a wide range of motors
- It can, in theory, be used on both Y and delta connected three-phase motors
- It requires no detailed knowledge of motor parameters
- It is relatively insensitive to motor manufacturing tolerance variations

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 3 shows how 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 3: 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 each of the six steps. Sequencing through these steps moves the motor through one electrical revolution.

The step commutation taken in one electrical revolution is summarized in Table 1. For every sector, two windings are energized and one winding is not. The fact that one of the windings is not energized during each sector is an important characteristic of six-step control, which allows the use of a sensorless control algorithm in this application.

TABLE 1: STEP COMMUTATION

Steps	Red	Green	Blue
Step 1	+	—	Not Driven
Step 2	+	Not Driven	—
Step 3	Not Driven	+	—
Step 4	—	+	Not Driven
Step 5	—	Not Driven	+
Step 6	Not Driven	—	+

Back-EMF (BEMF) Generation

When a BLDC motor rotates, each winding generates BEMF, which opposes the main voltage supplied to the windings according to 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 can be calculated in terms of these parameters and angular velocity using Equation 1.

EQUATION 1: 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

If magnetic saturation of the stator is avoided, or the dependency of the magnetic field on temperature is ignored (i.e., B is constant), the only variable term is the rotor's angular speed. Therefore, BEMF is proportional to the rotor speed; as the speed increases, the BEMF increases.

The frequency at which the sectors are sequenced determines the speed of the motor; the faster that the sectors are commutated, the higher the mechanical speed is achieved. The BEMF voltage is proportional to the rotor's speed. Because of this, detection of position using the BEMF at zero and very low speeds is not possible. Nevertheless, there are many applications (e.g., fans and pumps) that do not require positioning control or closed-loop operation at low speeds. For these applications, a BEMF sensing method is very appropriate.

The speed of the motor is measured using RPM, which is equivalent to the mechanical revolution per minute taken by the motor. RPM measurements are used in motor performance evaluation or as feedback in closed-loop operations. RPM can be calculated using Equation 2.

EQUATION 2: ELECTRICAL CYCLE TO MECHANICAL CYCLE RELATIONSHIP

$$\textit{Mechanical Revolution per Minute} = \textit{Electrical Revolution per Minute}/\#\ \textit{of Pole Pairs}$$

The commutated voltage applied to the stator also has a direct impact on motor performance. For efficient control, the applied voltage must be at least enough to match the generated BEMF, plus the voltage drop across the motor's windings due to torque production. This voltage drop, in turn, is equal to the impedance of the windings multiplied by the current.

If the commutated voltage is set to maximum, regardless of the motor's speed or torque production, the motor will be driven inefficiently with the wasted energy, heating the motor's windings. For the proper control necessary, Pulse-Width Modulation (PWM) is used to achieve the right voltage level. PWM is an efficient method of driving the motor, but it introduces some noise issues when attempting to acquire the control feedback signals (i.e., BEMF voltages).

To summarize, the important relationships for BLDC motors and sensorless control are:

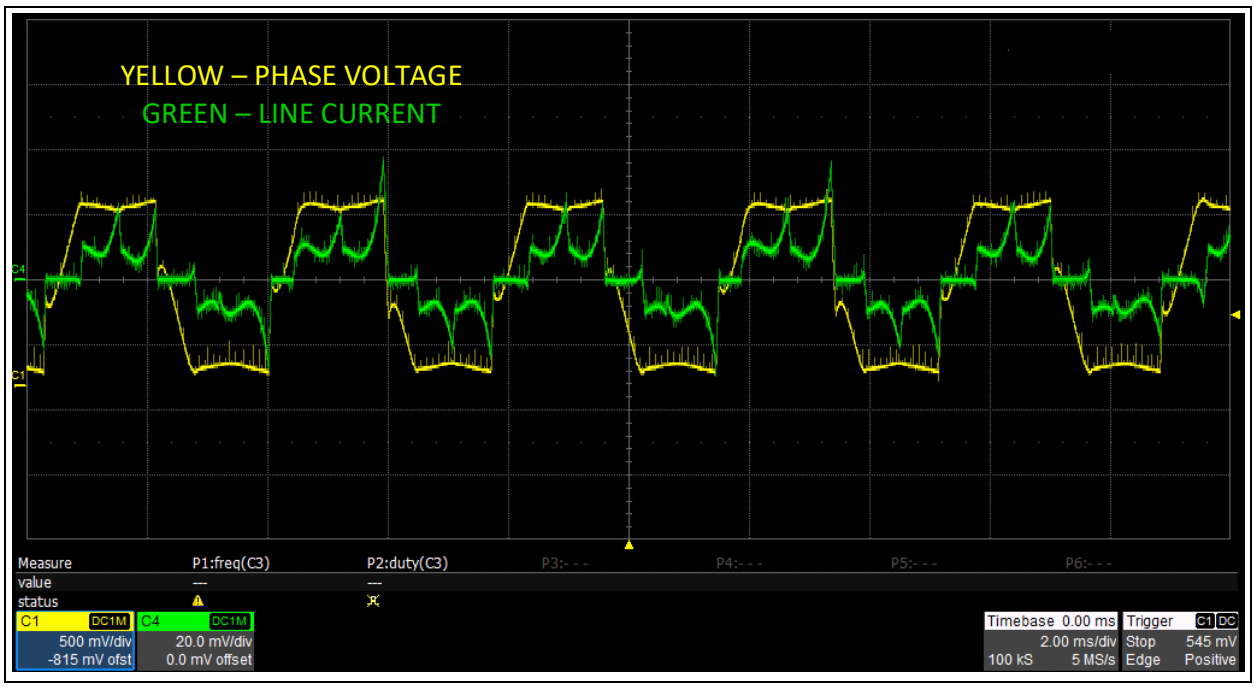
- The magnitude of the BEMF signal is proportional to speed
- The frequency of the BEMF signal is equal to the (mechanical) rotational speed times the number of poles pairs
- Motor torque is proportional to current (assuming the motor's temperature is constant)
- Motor drive voltage is equal to BEMF (proportional to speed) plus winding impedance voltage drop (proportional to current for a given torque)

Zero-Crossing Detection Methods

In BLDC motor control theory, the stator flux should be 90 electrical degrees ahead of the rotor flux for maximum torque generation. Consequently, for maximum torque, the phase current needs to be in-phase with the phase BEMF voltage.

In a three-phase BLDC motor, the phases are shifted 120° to each other. Therefore, using six-step (trapezoidal) commutation is a convenient method to create a rotating flux in the stator. In this method, the three phase voltages are commutated at 60 electrical degrees, and the phase voltage and line current should be in-phase as shown in Figure 4.

FIGURE 4: PHASE VOLTAGE AND LINE CURRENT RELATIONSHIP



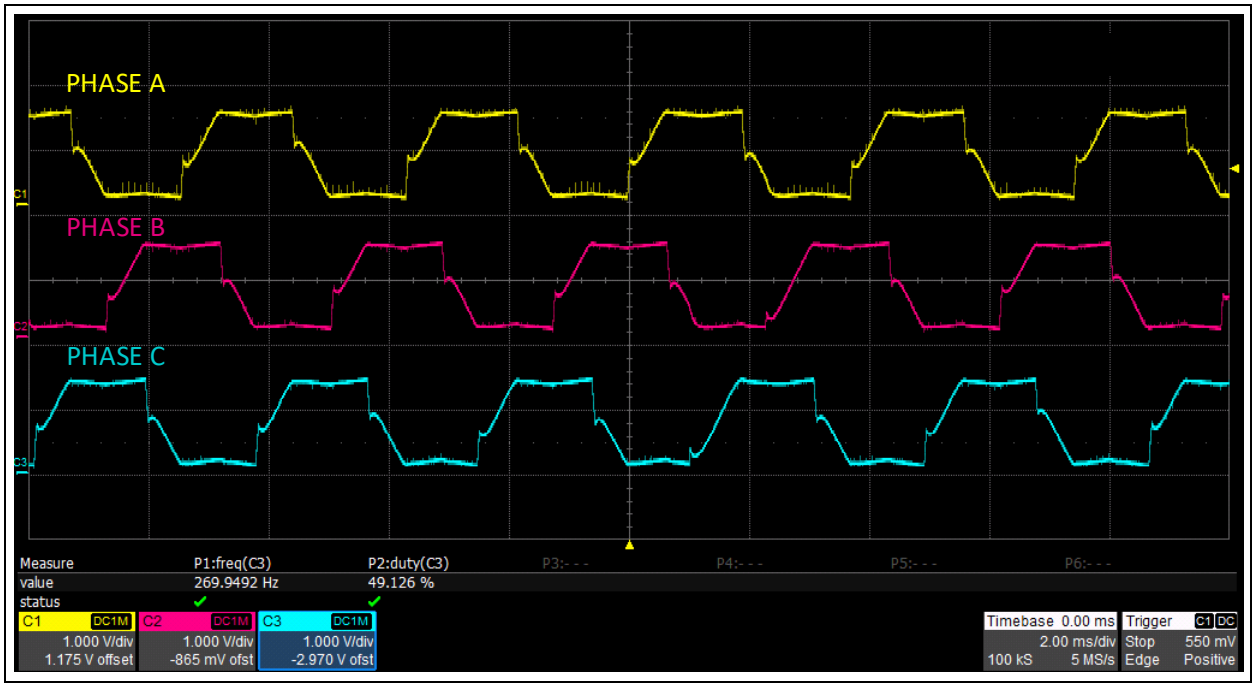
At maximum torque and full load, the phase current should have the same waveform as the driving voltage, neglecting the inductive reactance, and the two signals need to be in-phase. Figure 5 shows the individual idealized phase BEMF waveforms, as well as phase current, assuming an efficient commutation with a certain load.

The BEMF phase voltage is centered at one-half of the driving voltage. This means that any zero-crossing event indicates an intersection of the BEMF waveform with a point that is one-half of the supply voltage ($V_{BUS}/2$). The zero-crossing point occurs at 30 electrical degrees from the end of the last commutation, which is also 30 degrees from the next commutation point. The motor speed can thus be calculated from the time interval between two zero-crossing events. When the current zero-crossing event is identified, a precise schedule for future commutation steps can be achieved.

Each sector corresponds to one of six equal 60° portions of the electrical cycle (the sector numbering is arbitrary). Commutations occur at the boundary of each of the sectors. Therefore, the sector boundaries are what needs to be detected. There is an offset of 30° between the BEMF zero-crossing events and required commutation positions.

BEMF voltage zero-crossing signals can be detected by different methods. BEMF voltage can be either compared to half of the DC bus voltage or to a virtual motor neutral point. Both of these methods will be discussed to identify their advantages as well as their drawbacks.

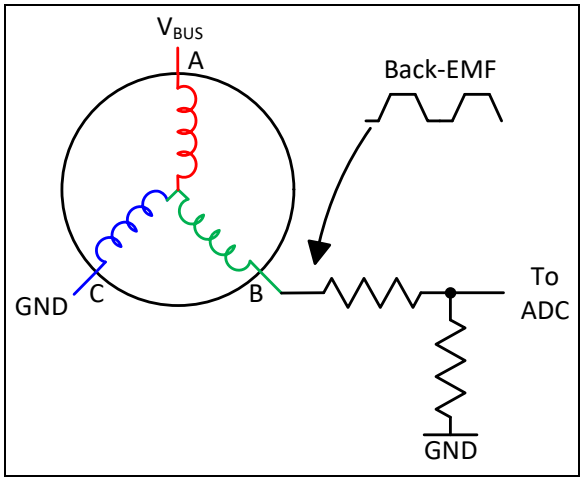
FIGURE 5: BEMF ON ALL THREE PHASES



COMPARING THE BEMF VOLTAGE TO HALF OF THE DC BUS VOLTAGE

In this method, the BEMF voltage is compared to one-half of the DC bus voltage ($V_{BUS}/2$) by using comparators, assuming that the zero-crossing events occur when BEMF is equal to $V_{BUS}/2$. Figure 6 shows the resistor configuration implemented using this method.

FIGURE 6: BEMF VOLTAGE COMPARED TO $V_{BUS}/2$



Assume that the motor is in commutation Step 1 (refer to Figure 3), in which Phase A is connected to $+V_{BUS}$ through an electronic switch, Phase C is connected to GND through an electronic switch and Phase B is open. The BEMF signal observed on Phase B has a negative slope and its minimum value is almost equal to $+V_{DC}$ just before the commutation Step 2 occurs. Phase B reaches the value of GND when commutation Step 2 occurs.

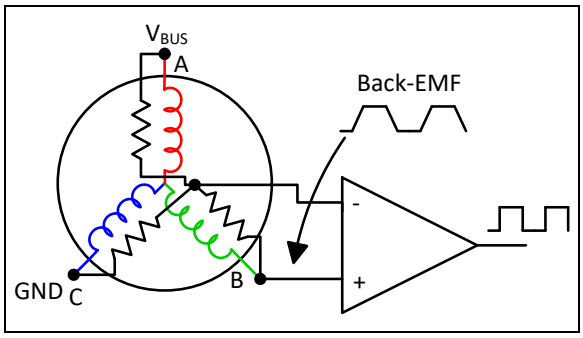
At that instant, Phase B is now connected to GND through an electronic switch, Phase C is now open and Phase A remains connected to V_{DC}. The BEMF signal observed on Phase C has a positive slope and its maximum value is almost equal to V_{DC} just before commutation Step 3 occurs. Both slopes observed on Phase B and Phase C are compared to $V_{DC}/2$ to determine the zero-crossing event.

This method is easily implemented with operational amplifiers configured as comparators; however, it has its own limitations. In comparing the BEMF voltage with $V_{BUS}/2$, motor winding parameters are assumed identical, which would pose a problem since it is too ideal.

COMPARING THE BEMF VOLTAGE TO THE MOTOR NEUTRAL POINT

The zero-crossing sensing method described previously can be simplified by using a variable threshold voltage point to detect the zero-crossing events. This variable voltage is the motor neutral point. The neutral point is not physically available for most BLDC motors. However, it can be generated by using a resistor network, as shown in Figure 7.

FIGURE 7: BEMF VOLTAGE COMPARED TO A VIRTUAL NEUTRAL POINT



The neutral point signal can also be reconstructed in software by averaging the values of three simultaneously sampled ADC channels, as shown in Equation 3. The reconstructed motor neutral voltage is then compared to each BEMF signal to determine the zero-crossing events. A zero-crossing event occurs (or is said to have occurred) when any one of the three Back-EMF voltages crosses over the $V_{DC}/2$ voltage in either direction

EQUATION 3: VIRTUAL NEUTRAL POINT AND BEMF SIGNALS RELATIONSHIP

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

Where:
- Vn is the motor neutral voltage
- $BEMF\ A$ is the BEMF voltage in Phase A
- $BEMF\ B$ is the BEMF voltage in Phase B
- $BEMFA\ C$ is the BEMF voltage in Phase C

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. The virtual neutral point is conveniently adjusted depending on the voltage reading, though it adds software overhead.

Back-EMF Sampling

One of the challenges in the implementation of the virtual neutral point method is determining the right time to sample the BEMF signal. The BEMF samples acquired by the ADC may be affected by the resonant transition voltages caused by the PWM switching frequency. To limit this, the dsPIC DSC is configured in a way that the ADC controller simultaneously samples all of the three BEMF signals at a sampling rate equal to the PWM frequency. Therefore, the ADC sampling rate is synchronized with the PWM reload event. It is also configured to take samples at the approximate middle of PWM on-time with the purpose of reducing 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.

Figure 8 shows the BEMF signals with respect to the created virtual neutral point. It is continuously compared to identify zero-crossing events. Figure 9 shows the sampling of the BEMF signal. Using center-aligned PWM, the ADC is triggered to sample every two PWM period values, which occur every half of PWM on-time.

FIGURE 8: ADC SAMPLING IN REFERENCE TO PWM PERIOD

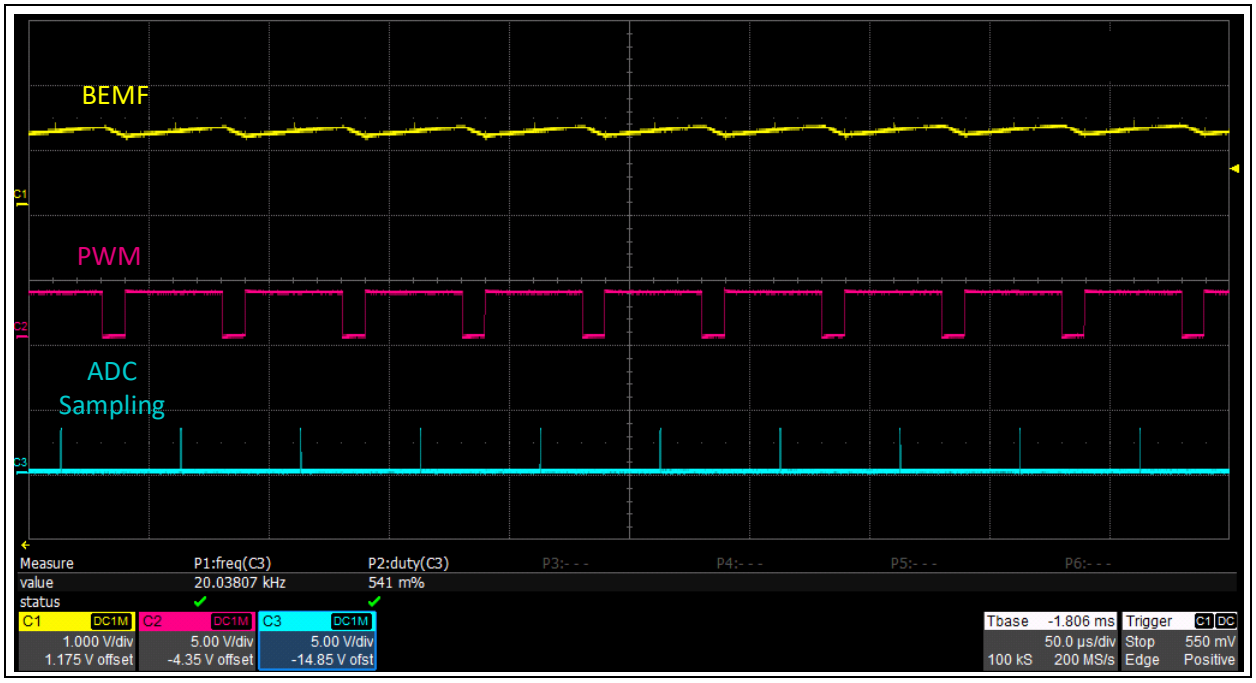
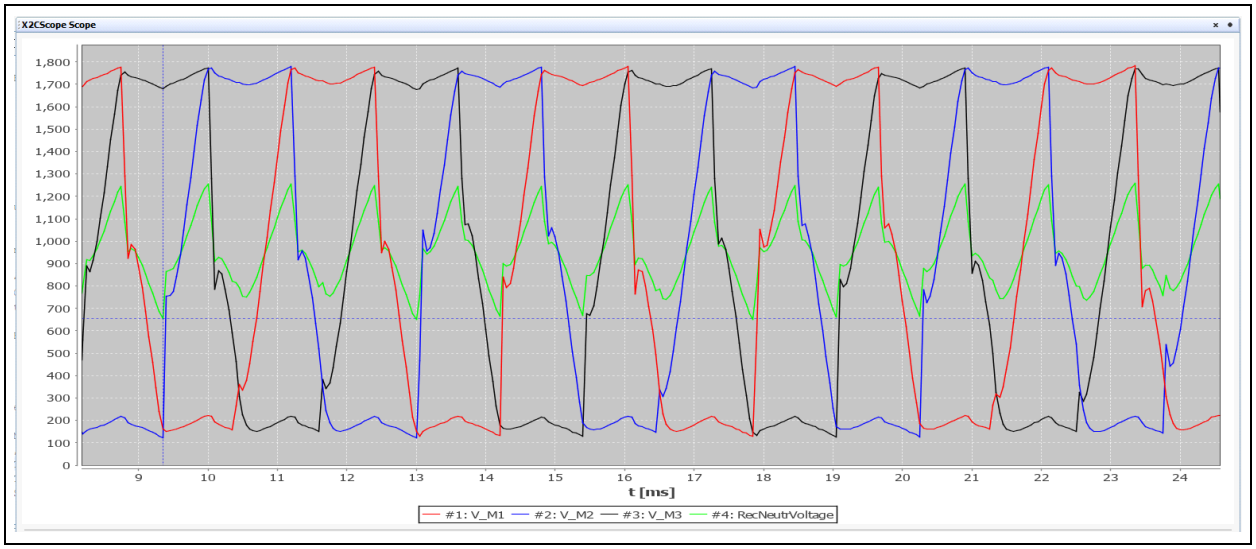


FIGURE 9: BEMF SIGNALS vs. VIRTUAL NEUTRAL POINT WITH THE PWM DUTY CYCLE



Digital Filter (Majority Function)

As previously noted, the BEMF signal can be adversely affected by PWM commutation in the other two energized windings. The coupling between the motor parameters, especially inductances, can induce ripple in the BEMF signal that is synchronous with the PWM commutation. This effect is less noticeable on motors with concentrated windings.

Since this induced ripple can cause faulty commutation, it is essential to filter the BEMF signal. There are, theoretically, two approaches: analog or digital. Analog filtering has the disadvantages of additional components and cost, as well as frequency-dependent phase and magnitude variations.

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 represent 66% of the numbers. The majority function always returns the value of the majority (> 50%) of the numbers. Table 2 shows an example of a three-input majority function. The majority of the values can be expressed using the AND (^) and OR (v) operators, as shown in Equation 4.

TABLE 2: THREE-INPUT MAJORITY FUNCTION

A	B	C	Output
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

EQUATION 4: BOOLEAN REPRESENTATION OF THE MAJORITY FUNCTION

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

The implementation of this nonlinear filter is based on a six-sample 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 corresponding to the existing Commutation state and an Exclusive-OR 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 composed of 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 26 possible combinations that the six-sample window could have for the active-masked BEMF signal. Each value on the Look-up Table (LUT) 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 listed in Table 3, which can be calculated using Equation 5.

EQUATION 5: CALCULATING ARRAY VALUES

First Half: $Array\ Value\ [N] = N * 2$

Second Half: $Array\ Value\ [N] = N * 2$

TABLE 3: 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 as listed in Table 4. The values pointed to by these unique indexes are replaced by '1' to indicate that a true-to-false condition has occurred. They are selected based on their 6-bit binary values using these majority function criteria:

- A majority of '1' (> 50%) in the three Most Significant bits (MSbs)
- A majority of '0' (> 50%) in the three Least Significant bits (LSbs)

TABLE 4: UNIQUE INDEX NUMBERS INDICATING A TRUE-TO-FALSE CONDITION

Number	6-Bit Binary Value
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 multiples of any of the 16 unique numbers. Table 5 provides some numbers that match this condition.

TABLE 5: NUMBERS THAT ARE UNIQUE NUMBER MULTIPLES

Number	6-Bit Binary	Number of Right Shifts	Unique Number Pointed To	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 one of the 16 unique numbers) are then pointed to their 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 6 shows the numbers that are not a multiple of a unique value. The complete array of filter coefficients, combining the initial array with Unique Number Pointers, is shown in Table 7.

TABLE 6: NUMBERS THAT NEVER POINT TO A UNIQUE VALUE

Number	6-Bit Binary	Number Pointed to Before Becoming Zero	Unique Number to be Pointed
1	000001	2, 4, 8, 16, 32	5
9	001001	18, 36, 8, 16, 32	5
36	100100	8, 16, 32	3
10	010001	34, 4, 8, 16, 32	5

TABLE 7: COMPLETE MAJORITY FILTER COEFFICIENTS

Array	Array Value	Array Unique Numbers	Array	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

For a more understandable representation, two example computations were given. Table 8 shows an example of the complete filtering process of a noiseless binary representation of the BEMF signals. Table 9 shows another example of the complete filtering process of a noisy binary representation of the BEMF signals.

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. From the moment of zero-crossing detection, commutation delay is equal to the timing of 30 electrical degrees, minus the time required to execute the digital filtering process. To implement the commutation delay, one of the device's general purpose timers is used to measure the amount of time elapsed from one zero-cross event to the next.

TABLE 8: DIGITAL FILTERING COMPUTATIONS USING NOISELESS BEMF SIGNALS

Angle	BEMF Phase			XOR Masked Phase			AND Masked Phase			Logical Test	Commutation Step	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	—	—
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	FALSE	—	—
132	1	0	1	0	0	0	1	0	0	1	3	4	TRUE	—	—

TABLE 9: DIGITAL FILTERING COMPUTATIONS USING NOISY BEMF SIGNALS

Angle	BEMF Phase			XOR Masked Phase			AND Masked Phase			Logical Test	Commutation Step	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	—	—
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	—	—

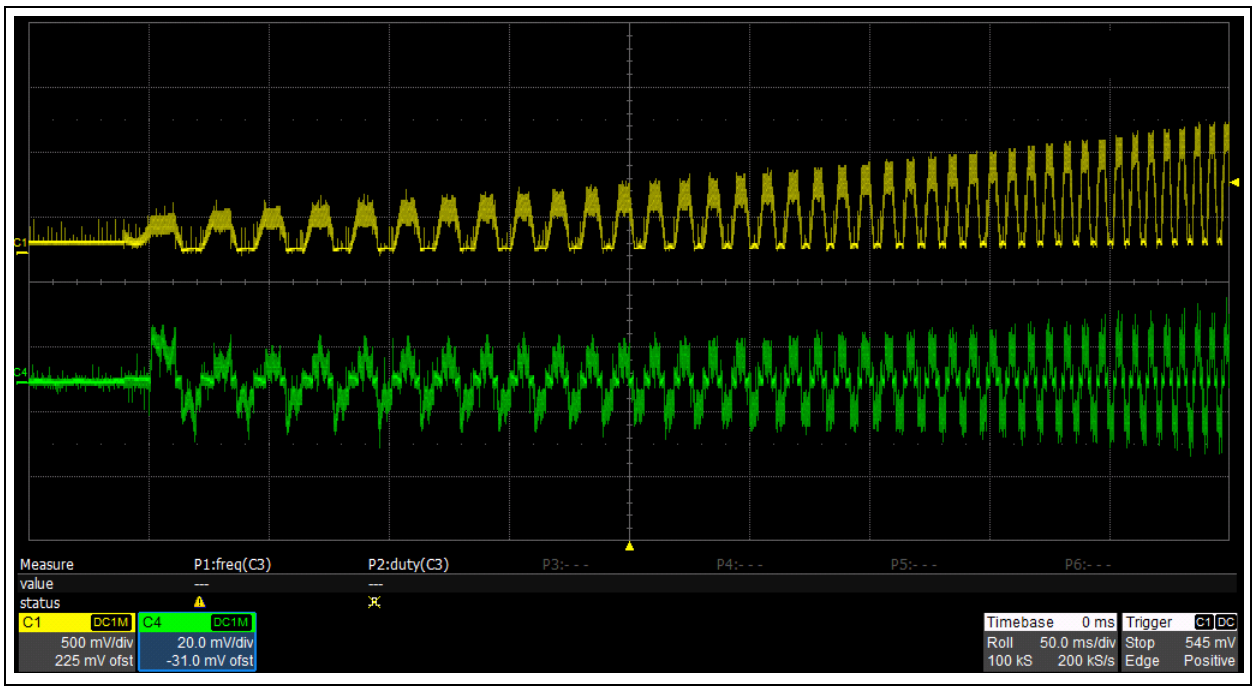
Start-up Sequence

During start-up, the amplitude of the Back-EMF is insufficient to determine zero-crossing. The motor needs to be accelerated in forced commutation, so that the motor generates BEMF signals that can be measured and processed by the ADC of the microcontroller.

For the start-up, the PWM duty cycle is predefined enough to break the stand still inertia, but not too high that the motor generates very high start-up current. The standard commutation step sequence, based on the PWM Look-up Table, is applied at a fixed rate based on the number of PWM ticks to break the motor from the Idle state.

After the forced commutation sequence, the running motor is already producing readable BEMF signals and the transition to open-loop or closed-loop sensorless operation will be initiated. Figure 10 shows the phase voltage and line current behavior during start-up.

FIGURE 10: START-UP VOLTAGE AND CURRENT SIGNAL RESPONSE



CONTROL LOOPS

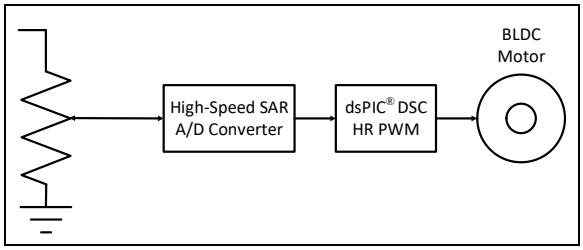
This application can operate in either of two control modes during sensorless operation, which can be configured in software. These modes are as follows:

- 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, a motor under constant torque can be operated open loop over its entire speed range. Therefore, an open-loop controller can be modeled by linking the PWM duty cycle to a control variable, which is generated by a potentiometer being sampled by an ADC. The block diagram of this mode is shown in Figure 11.

FIGURE 11: OPEN-LOOP CONTROL BLOCK DIAGRAM

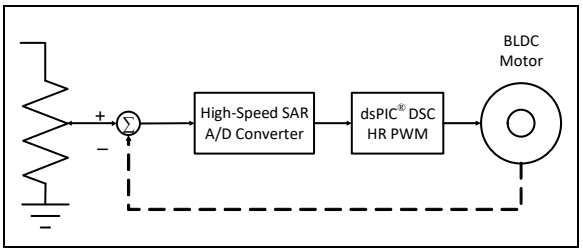


The potentiometer is used as an input to the ADC peripheral to control the speed of motor rotation. Using the latest dsPIC controllers, the ADC is configured to convert analog signals into 12-bit data in integer format. In this format, the ADC output can easily be multiplied to the maximum duty cycle as a way of scaling and obtaining the optimized range of speed, depending on the motor and user preference. The product of the ADC value and the maximum duty cycle is used as the desired PWM value, which will be periodically compared to the actual duty cycle value. Then, the actual duty cycle value will be incremented or decremented until it reaches the desired duty cycle value.

Proportional-Integral (PI) Closed-Loop Modes

Closed-Loop mode measures the current speed as feedback and adjusts the PWM duty cycle based on the desired set speed. The difference between the desired speed value and the set speed is used as an error signal to calculate the error correction factor for speed adjustments. Figure 12 shows the PI closed-loop block diagram.

FIGURE 12: PI CLOSED-LOOP BLOCK DIAGRAM



The actual speed of the motor can be calculated based on the method used to identify the zero-crossing points during commutation. In a Running Motor state, the number of clock ticks between two Zero-Crossing states signifies the time it takes to complete 60 degrees or one-sixth of an electrical revolution. By identifying the number of pole pairs of the motor, the actual number of clock ticks per one mechanical revolution can be calculated.

Once the current speed is calculated, it is then compared to the desired speed set from the potentiometer. The variable voltage of the potentiometer is converted to digital data by the ADC peripheral, which is scaled based on the maximum and minimum speed of the motor. The Proportional-Integral error between the desired speed and the current speed is calculated and then multiplied by the PI constants, as shown in Equation 6. The PI output is then scaled to match the range of the PWM duty cycle. Using Microchip's motor control library, PI closed-loop function is made simpler to implement the application software.

EQUATION 6: PI CONTROLLER COMPUTATIONS

$$Speed\ Error = Desired\ Speed - Current\ Speed$$

$$Integral\ Error = Integral\ Error + Speed\ Error$$

$$PI\ Output = (k_p) \bullet (Speed\ Error) + (K_i) \bullet (Integral\ Error)$$

SOFTWARE OVERVIEW

This section provides an overview of the BLDC control algorithm and the application software used in this application note. Figure 13 shows the main function, where all peripherals used are initialized. All variables are set in their initial state and will be waiting for any pressed button to make a corresponding action. By toggling these buttons, functions that trigger motor actions, such as running and stopping, are executed. These functions are shown in Figure 14.

FIGURE 13: APPLICATION MAIN FUNCTION

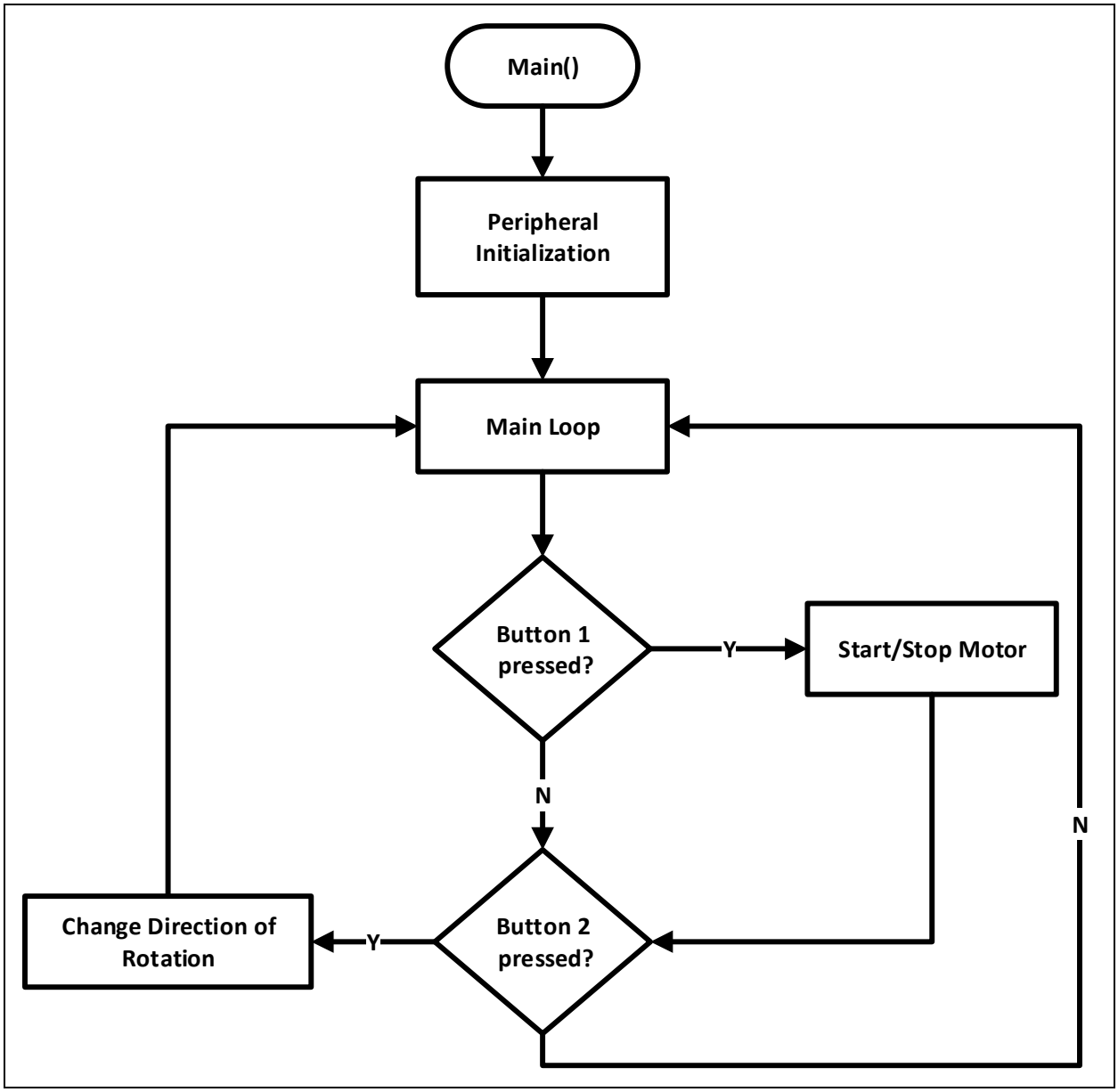
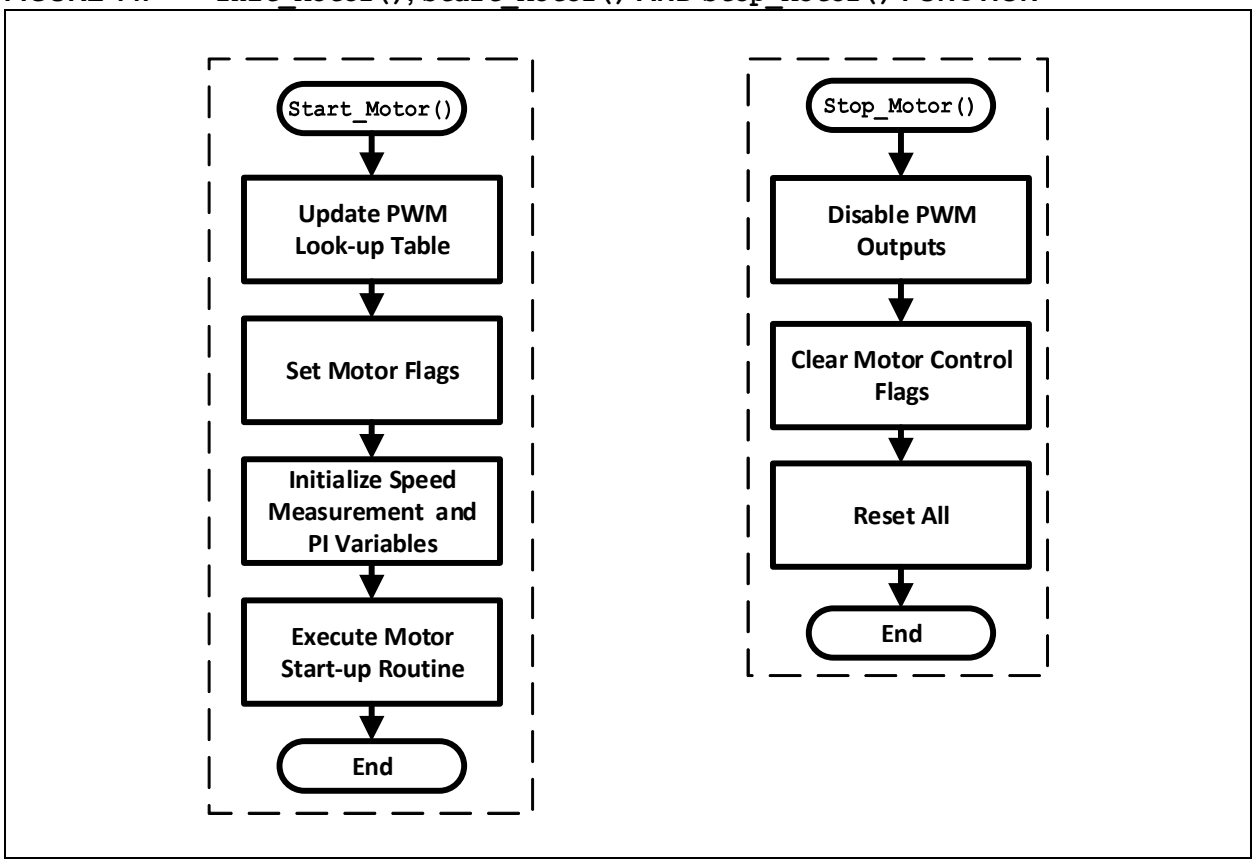
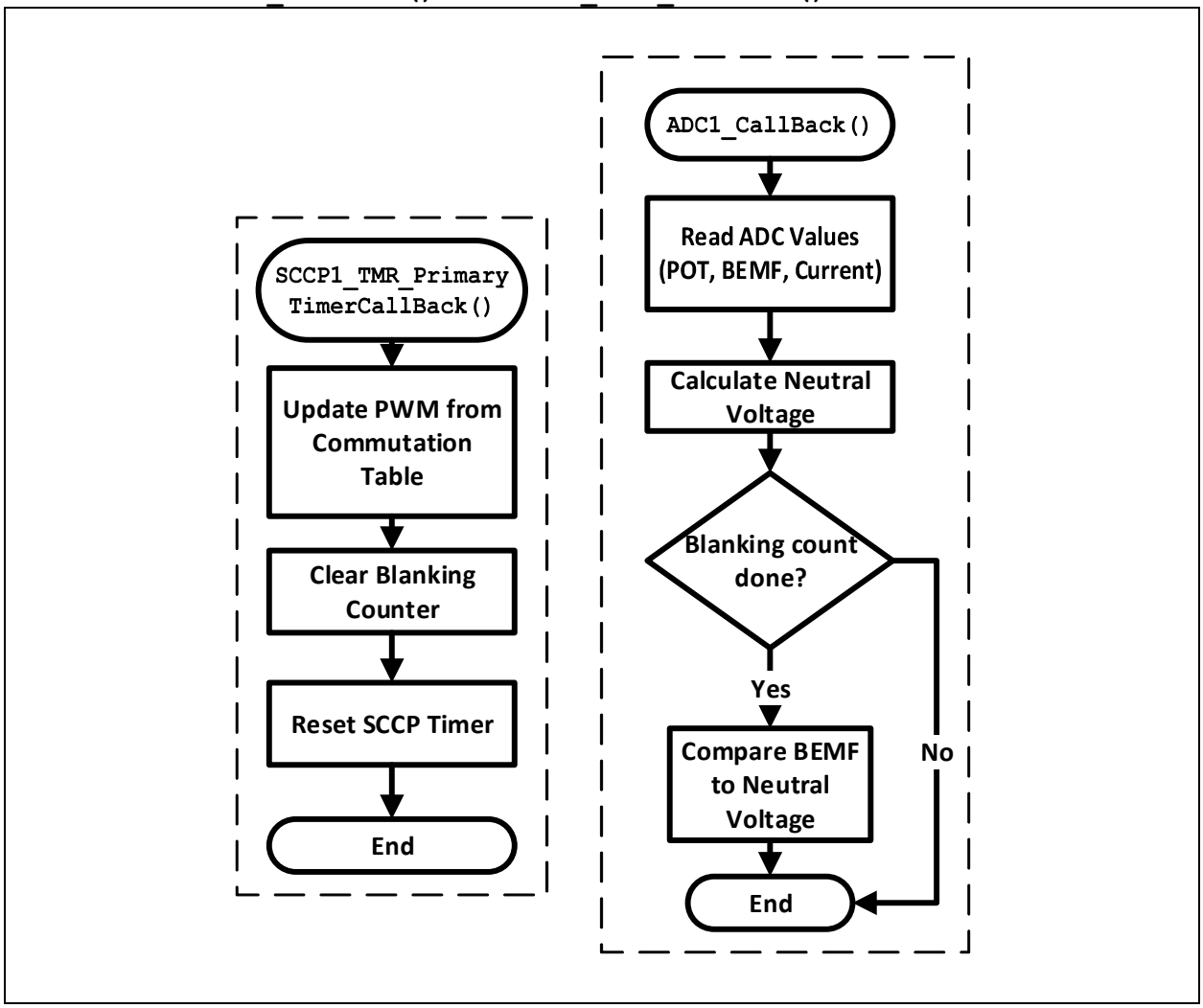


FIGURE 14: `Init_Motor()`, `Start_Motor()` **AND** `Stop_Motor()` **FUNCTION**



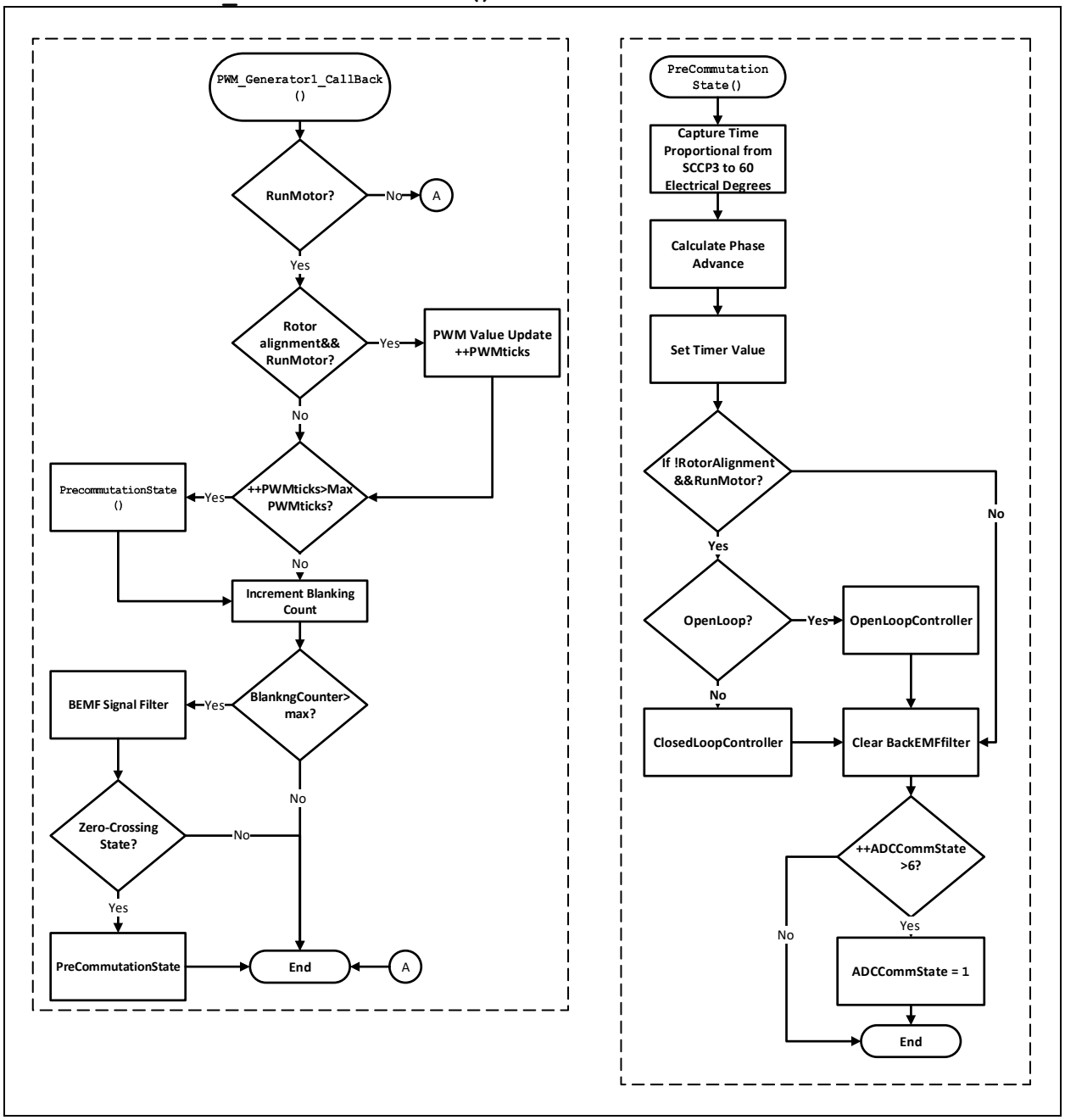
The ADC Interrupt Service Routine (ISR) in Figure 15 is executed with respect to the PWM reload event. It reads the sampled BEMF signals and potentiometer value. Neutral voltage is calculated and compared to the BEMF signal values which will be later used for the zero-crossing detection if blanking count is done. The SCCP ISR updates the PWM output configuration, depending on the PWM Look-up Table, based on the current commutation sector. The blanking count is cleared and resets the SCCP timer.

FIGURE 15: `ADC1_Callback()` **AND** `SCCP1_TMR1_CallBack()` **FUNCTION**



The PWM ISR in Figure 16 is executed periodically at the rate of 20 kHz. It contains the zero-crossing detection routine that is vital on sensorless BLDC control. The majority filter function is executed after the start-up routine. When a Zero-Crossing state is detected, Precommutatation state function is executed. It measures the time between the consecutive Zero-Crossing states, which will be used to trigger the next Commutation state. `ADCCommState` is also incremented, signifying the transition to the next commutation sector.

FIGURE 16: **`PWM_GeneratorCallBack()`**



CONCLUSION

This application note is intended for developers who want to drive a BLDC motor, using this new sensorless BLDC control technique, in a basic and simple form, without the use of off-chip comparators.

This sensorless control method, using a single-chip 16-bit device, does not require external hardware, except for a couple of resistors for BEMF signal conditioning to the operational voltage range of the ADC module. 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.

Digital filtering makes it possible to detect the zero-cross events more accurately 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 BLDC 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 a PMSM"*, Microchip Technology Inc., 2007

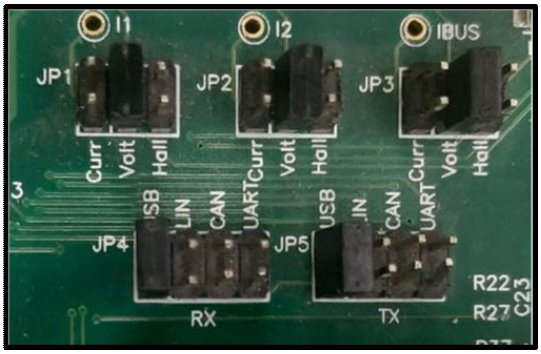
NOTES:

APPENDIX A: HARDWARE SETUP

The required connections for the dsPIC® MCLV-2 Development Board for this motor control application are presented in this section. Refer to the *"dsPICDEM™ MCLV-2 Development Board User's Guide"* for any clarification while setting up the hardware. When changing jumper connectors, make sure that the board is disconnected from power.

1. Set the following jumper connections for the BEMF voltage feedback signal. These pins are directly connected to the ADC pins of the dsPIC device used. Set the RX and TX jumpers to USB position for the diagnostic tool used in this application, which is X2C-Scope. Actual connections are shown in Table A-1.

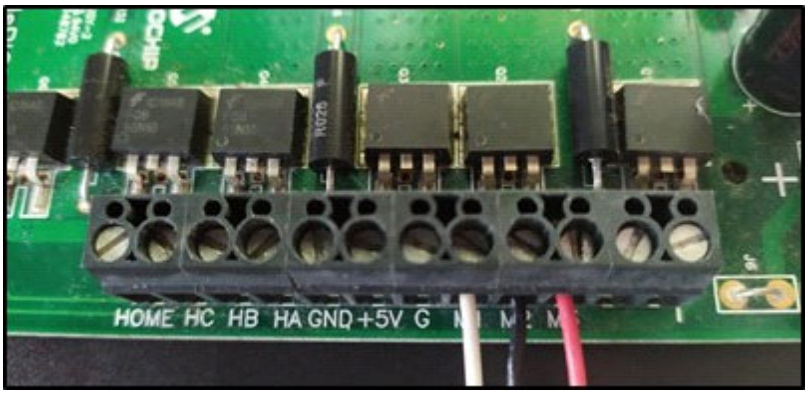
TABLE A-1: JUMPER CONNECTIONS

Jumper	Pin to Short	Board Reference
JP1	Volt Position	
JP2	Volt Position	
JP3	Volt Position	
JP4	USB Position	
JP5	USB Position	

2. Connect the three phase wires from the motor to the M1, M2 and M3 terminals of connector J7, provided on the Development Board, as mentioned in Table A-2.

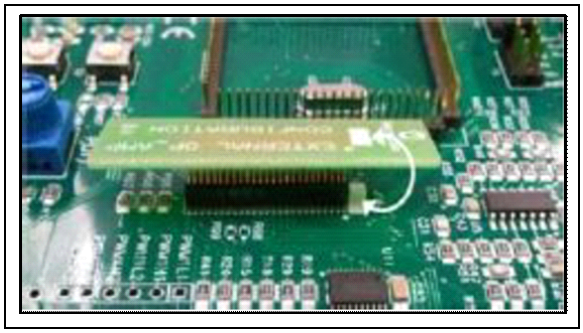
TABLE A-2: MOTOR CONNECTION

MCLV-2 Development Board	Hurst 75	
	Winding Terminals (color as per image below)	Molex® 39-01-2040 (Mating Connector)
M1	White	3
M2	Black	2
M3	Red	1



3. Connect the ‘External Op Amp Configuration Matrix Board’ to matrix board header J14, as shown in Figure A-1. Ensure that the matrix board is correctly oriented before proceeding.

FIGURE A-1: EXTERNAL OP AMP CONFIGURATION MATRIX



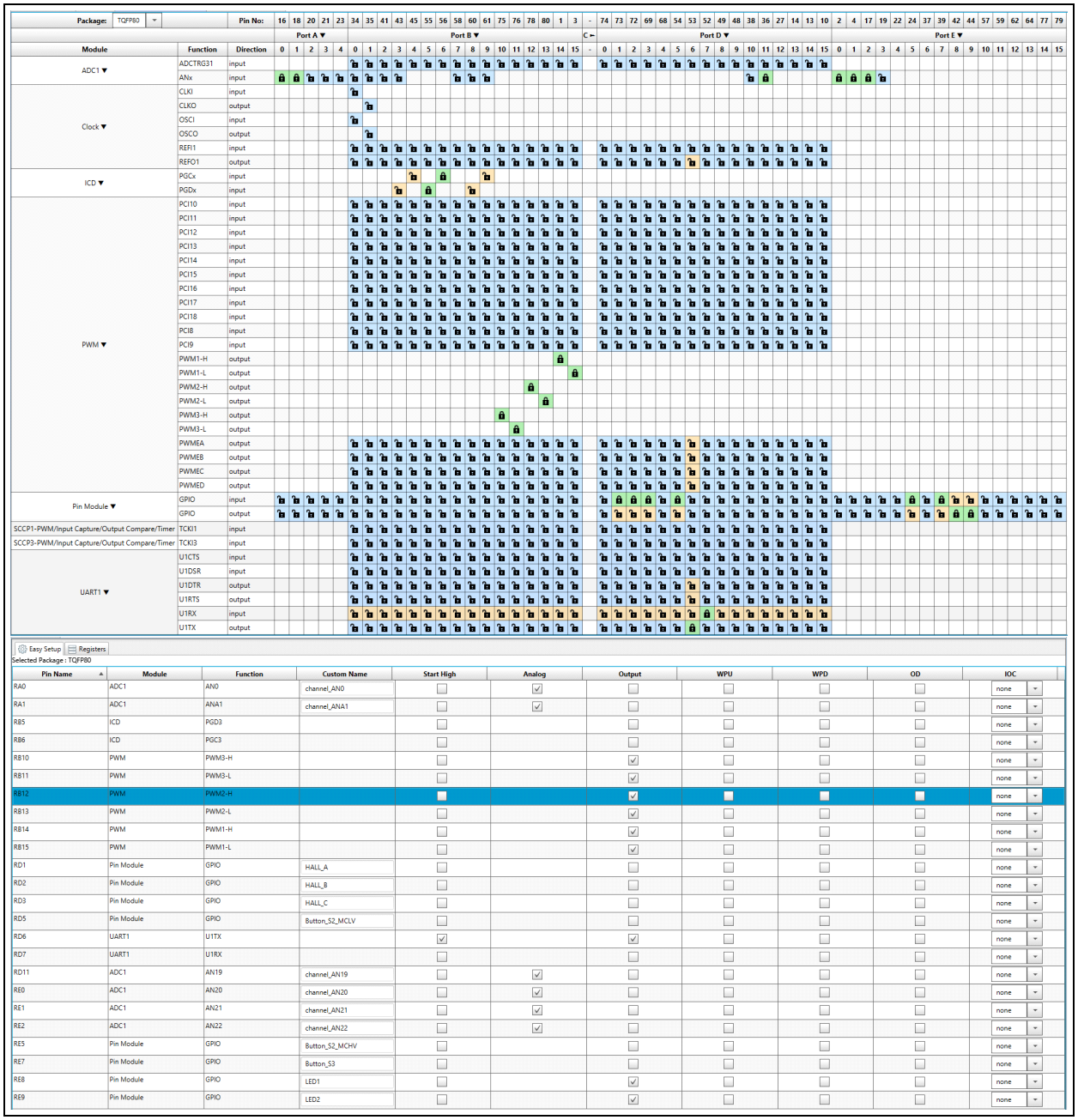
APPENDIX B: MCC GENERATION

In this section, the initialization and configuration of the peripherals used in this application note are shown. Microchip Code Configurator (MCC), a plug-in tool of MPLAB® X IDE, which provides a graphical environment for peripheral configuration, is used. MCC generates drivers in C code, which initializes the peripherals and provides functions that can be called on your firmware. Refer to the *"MPLAB® Code Configurator v3.xx User's Guide"* (DS40001829) for more information on how to install and set up the MCC in MPLAB X IDE.

The following steps provide the MCC settings of each peripheral used in this application.

1. In the System Module-Easy Setup tab, set the clock to FRC Oscillator with 8 MHz frequency. (**Note:** Make sure that the FNOSC bit of the FOSCEL register is set to Fast RC Oscillator with PLL rather than FRC only. It can be configured in the Register tab.) Enable the PLL and set the scalers as follows: Prescaler as 1:1, Feedback as 1:150, Postscaler 1 as 1:3 and Postscaler 2 as 1:1. This will set the clock to use 100 MHz frequency as FOSC/2.
2. For the SCCP1, select FOSC/2 as the clock source and set 1:64 as the prescaler. Enable the MCCP interrupt.
3. For measuring the time of a 60-degree rotation, the SCCP peripheral is configured. Enable the SCCP with FOSC/2 as the clock source. Set the prescaler as 1:64 and the 16-Bit Timer mode.
4. Set FOSC as the master clock to acquire 100 MHz frequency. Select PWM1, PWM2 and PWM3 as the required generators. For PWM master settings, select 200000000 Hz as the input clock selection with center-aligned as the PWM operation mode and Complementary PWM Output mode. Input 20 kHz as the requested frequency on the master PWM period part. For each PWM Generator, enable the master period. On trigger control settings, configure the PWM start of the cycle control. For PWM Generators 1/2/3, select self-trigger as the SoC trigger. For the trigger output selection, select EOC event on all PWM Generators. On PWM Generator 1, mark "none" and Trigger A compare as the ADC trigger. On the Register tab, enable the PWM Generator 1 interrupt.
5. For data monitoring using the X2C-Scope, enable UART1. Select FOSC/2 as the clock source and select 115200 as the baud rate.
6. For BEMF signal sensing and potentiometer reading, configure the ADC. Select FOSC/2 as the conversion clock source with a 1 µs target shared core sampling time. Enable AN19 (potentiometer control), AN20, AN21 and AN22 (BEMF signal detection) with a trigger source of 'PWM1 trigger1'. Enable the ADC common interrupt. On the Register tab, set 0xF on the SHRAMCx bits of ADCON2H.
7. In the PIN MANAGER configuration, set up the input/output pins of GPIO and the peripherals as shown in Figure B-1.
8. After configuring all the peripherals, click the 'Generate Code' button next to the Project Resources tab name in the top left corner. This will generate a `main.c` file to the project automatically. It will also initialize the module and leave an empty `while(1)` loop for custom code entry.

FIGURE B-1: MCC PIN MODULE AND PIN MANAGER GUIDE



APPENDIX C: REVISION HISTORY

Revision A (1/2008):

Original version of this document by D. Torres.

Revision B (9/2012):

Revision by A. Lita and M. Cheles to create a solution that only uses one ADC S/H circuitry, extending the algorithm compatibility to all 16-bit devices comprising a motor control PWM peripheral. The use of BEMF as a control modality and majority detect filtering is unchanged.

Revision C (3/2021):

Revision by A. Abacan. Extended the compatibility with new dsPIC33 devices. Microchip Code Configurator is used for peripheral setup for ease-of-use configuration. PI closed-loop control function is executed using the available Microchip's motor control library. For debugging and monitoring, DMCI is replaced with the X2C scope plug-in of MPLAB® X IDE. Core algorithm, such as zero-crossing detection and BEMF sensing have remained unchanged.

Revision D (4/2022):

Revision by A. Abacan. X2CScope library for debugging is integrated to the code using MCC. For the BEMF sensing, the single ADC sampling method is used to cater general motor specifications.

NOTES:

APPENDIX D: SOURCE CODE LISTING

The latest software version can be downloaded from the Microchip website (www.microchip.com). The user will find the source code appended to the electronic version of this application note.

NOTES:

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

ISBN: 978-1-6683-0292-7

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://www.microchip.com/support
Web Address:
www.microchip.com

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

Austin, TX
Tel: 512-257-3370

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
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

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

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

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

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

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

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

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

Italy - Padova
Tel: 39-049-7625286

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

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

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

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820