AN1160

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

Author: *Adrian Lita and Mihai Cheles*
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) or a PIC24 microcontroller. 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 four main parts:

- Sampling trapezoidal BEMF signals using the microcontroller's Analog-to-Digital Converter (ADC)
- PWM ON-side ADC sampling to reduce noise and solve low-inductance problems
- Comparing the trapezoidal BEMF signals to $V_{BUS}/2$ to detect the zero-crossing points
- Filtering the signals coming from the comparisons using a majority function filter
- Commutate the motor driving voltages in three different modes:
  - Classic Open Controller
  - Classic Closed-Loop Controller
  - Proportional-Integral (PI) Closed-Loop Controller

This new control method is a single-chip 16-bit PIC® MCU or dsPIC DSC device-based solution. The only external hardware required is a few resistors, used to reduce the BEMF signals to the operational voltage range of the device's ADC module.

SENSORED CONTROL VERSUS SENSORLESS CONTROL

The BLDC motor is used for both consumer and industrial applications due to its compact size, controllability and high efficiency. Increasingly, it is also used in automotive applications to eliminate 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, however:

- 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

If low cost is a primary concern, if low-speed motor operation is not a requirement, and if the motor load is not expected to change rapidly, sensorless trapezoidal control may be a better choice for your application. However, there are specific algorithms to overcome all of the above listed disadvantages.

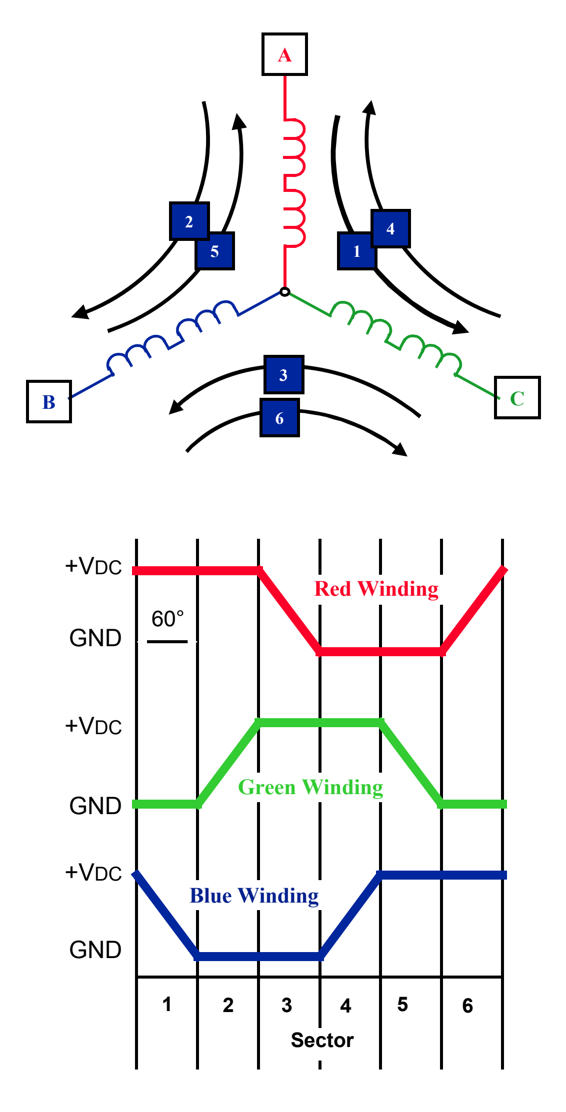
The BEMF zero-crossing technique described here is recommended for several reasons:

- It is suitable for use on a wide range of motors
- It can, in theory, be used on both Y and delta-connected 3-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 1 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 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 each of the six steps. Sequencing through these steps moves the motor through 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.

Generating and Sensing BEMF

When a BLDC motor rotates, each winding generates BEMF, which opposes the main voltage supplied to the windings in accordance with 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
- Angular velocity of the rotor
- Magnetic field generated by rotor magnets

BEMF can be calculated in terms of these parameters and angular velocity using Equation 4:

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 commutated voltage applied to the stator also has a direct impact on the correct functioning of the motor. For efficient control, the applied voltage must be at least enough to match to 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 times the current.

Generally speaking, 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

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

For the 3-phase BLDC motors considered, the phases are shifted 120° from each other, so a convenient method for having a rotating rotor flux in the stator is the six-step commutation scheme previously described, commutating each of the three-phase voltages 60 electrical degrees. 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, as it can be seen when comparing Figure 2 (high current, load applied) with Figure 3 (low current, no load). Figure 6 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 actually 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 60º 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 30º between the BEMF zero-crossing events and required commutation positions.

FIGURE 2: PHASE VOLTAGE AND PHASE CURRENT WHEN LOAD IS APPLIED

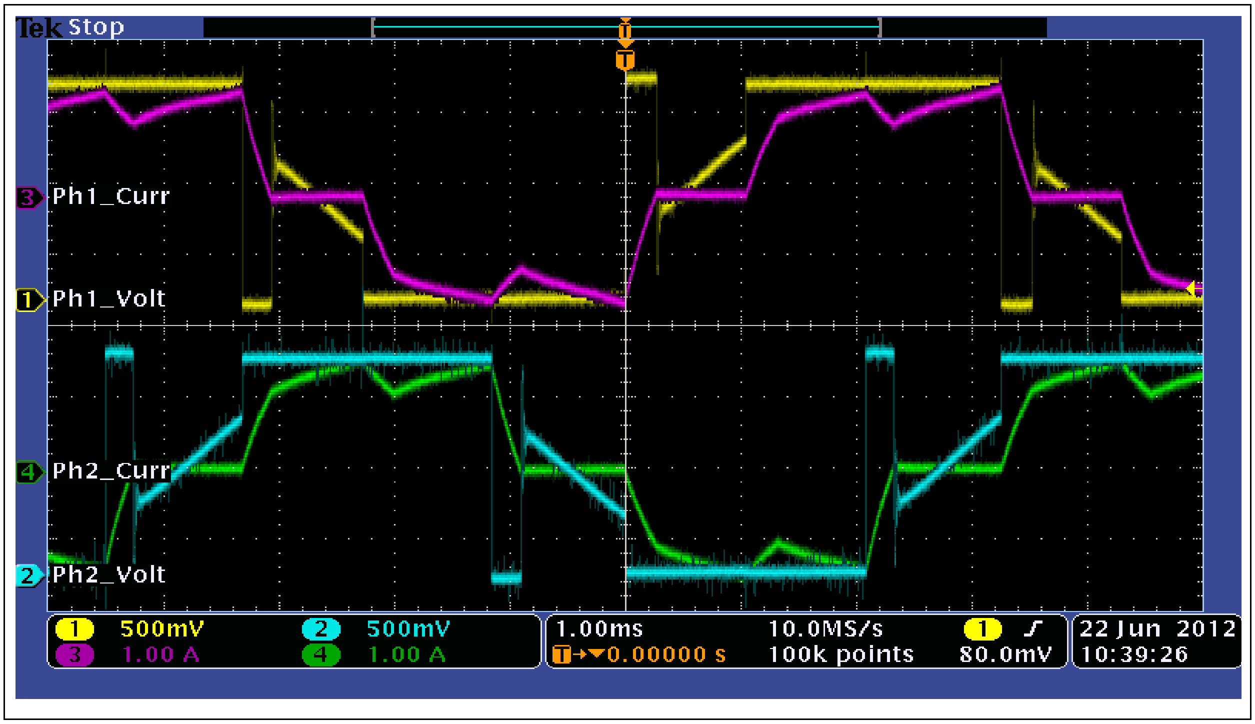
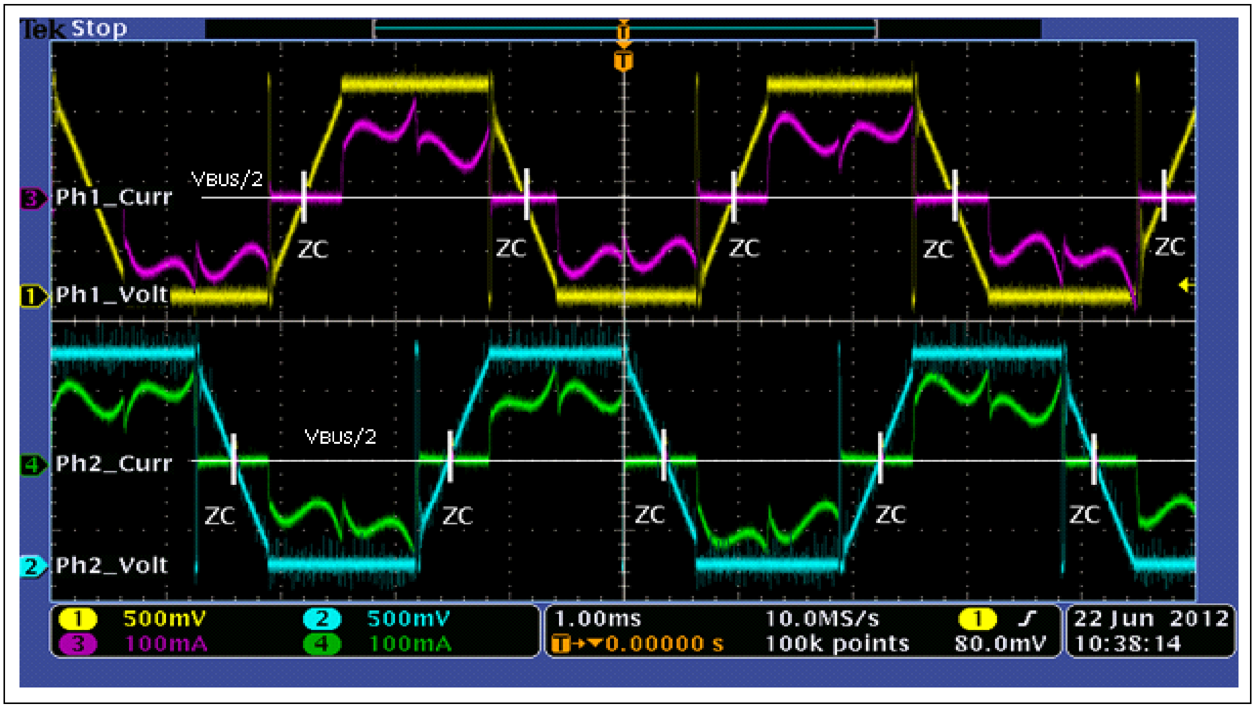


FIGURE 3: ZERO-CROSSING POINT EXACT OCCURRENCE



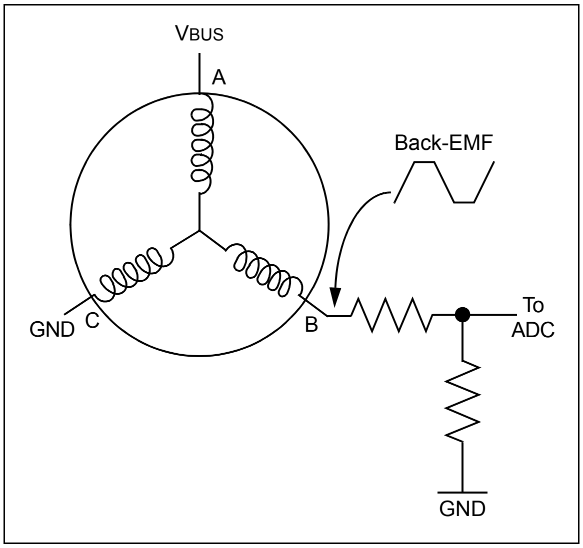
Detecting BEMF Zero-Crossing Signals

BEMF voltage zero-crossing signals can be detected by different methods. This section describes two different sensing methods. Both methods have advantages as well as drawbacks, which will be discussed for each case. Each method assumes that a wired neutral point is not provided or that the stators are wired in a delta configuration.

COMPARING THE BEMF VOLTAGE TO HALF THE DC BUS VOLTAGE

This method consists of comparing the BEMF voltage 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 4 shows the circuitry used to implement this method.

FIGURE 4: BEMF VOLTAGE COMPARED TO $V_{BUS}/2$



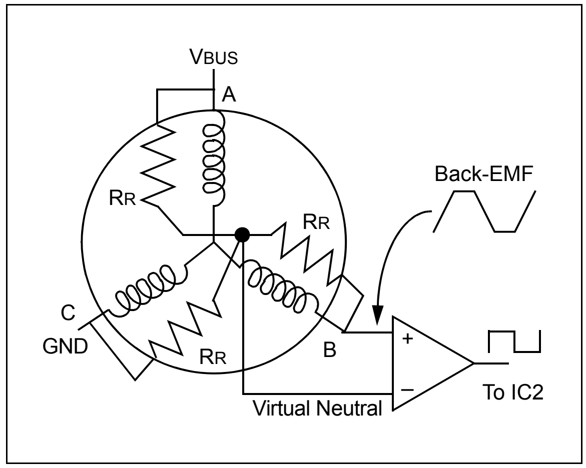
Assume that the motor is in commutation Step 1 (refer to Figure 1), in which Phase A is connected to $+V_{BUS}$ through an electronic switch, Phase C is connected to GND through an electronic switch and Phase B is open. The BEMF signal observed on Phase B has a negative slope and its minimum value is 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 $V_{DC}/2$ in order to determine the zero-crossing event. This is easily implemented with operational amplifiers configured as comparators.

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. Three resistors (R_R) are connected in parallel with the motor windings and connected together to generate a virtual neutral point, as shown in Figure 5.

FIGURE 5: 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 (Equation 2). The reconstructed motor neutral voltage is then compared to each BEMF signal to determine the zero-crossing events. An event occurs when the BEMF signals are equal to the motor neutral point.

EQUATION 2: VIRTUAL NEUTRAL POINT AND BEMF SIGNALS RELATIONSHIP

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

where: Vn is motor neutral voltage

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

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

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

Figure 6 shows the BEMF signals measured on all three phases.

Figure 7 shows the required circuitry for a complete BLDC control system.

FIGURE 6: BEMF ON ALL 3 PHASES

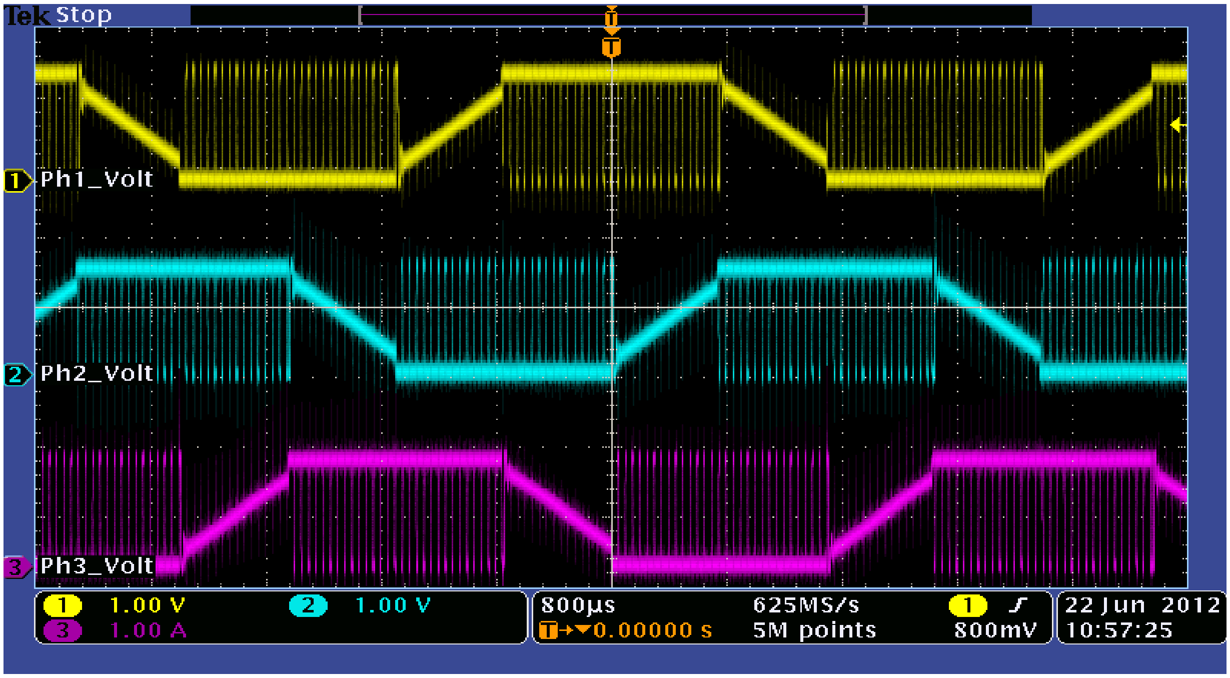


FIGURE 7: BEMF VOLTAGE MEASURED USING THE dsPIC® DSC ADC

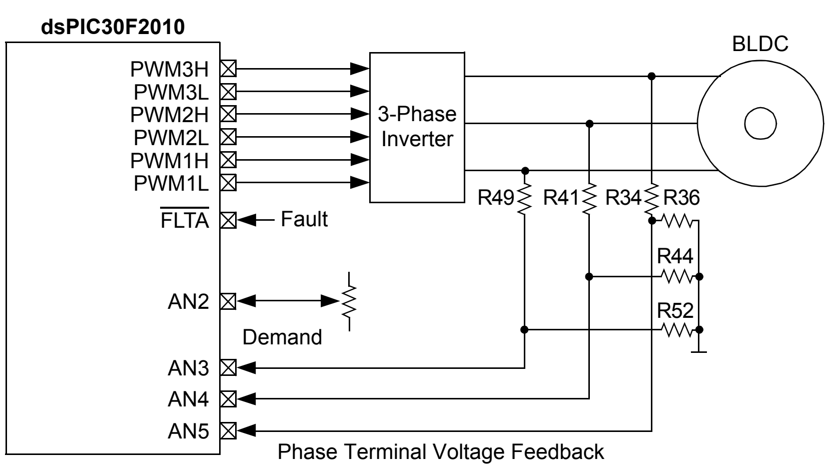


Figure 8 shows the sampling process over a sector (60°) period. The lower window provides an enhanced view of the time line while ADC sampling is taking place. The ADC_Read signal represents the times at which the ADC samples the BEMF voltage. Note that through all of the BEMF decay ramp, only the first half of the period is sampled. This is because once the zero-crossing event has been detected, no further sampling is done. This allows the top-level application more time to execute other important tasks.

Figure 9 and Figure 10 compare the differences in ADC sampling to the PWM duty cycle. Sampling and conversion are configured to occur as fast as the ADC and the controller will allow. At the same time, the sampling of the BEMF voltage is acquired only on the high side of the PWM signal, so the sampling rate and the PWM frequency are directly proportional. As shown in Figure 9 and Figure 10, the number of samples taken depends on the duty cycle. The value of the duty cycle (corresponding to PWM High ON state) is proportional to the voltage applied to the motor winding; at the same time, higher motor speeds require a higher voltage. Since the BEMF signal is sampled during the PWM High ON state, it can be deduced that higher duty cycles (and higher speeds) result in more ADC samples being taken. More samples result in a faster and more accurate detection of a zero-crossing event, with a net effect that sector transitions can be scheduled much quicker.

An advantage of using the VBUS/2 method is the need to sample only one ADC channel, as opposed to simultaneously sampling three channels sequentially, required for the neutral reconstruction method.

The challenge of this method consists of determining the correct time to sample the BEMF signals, since the samples acquired by the ADC may be affected by the resonant transition voltages caused by the PWM switching frequency. The ADC module is configured to take samples at the PWM ON time in order to avoid the ringing noise produced by the electronic switches and other noises (e.g., the high-voltage spikes produced when a motor winding de-energizes). These noises could create false zero-crossing events.

FIGURE 8: ADC SAMPLING vs. PWM AND BEMF SIGNAL

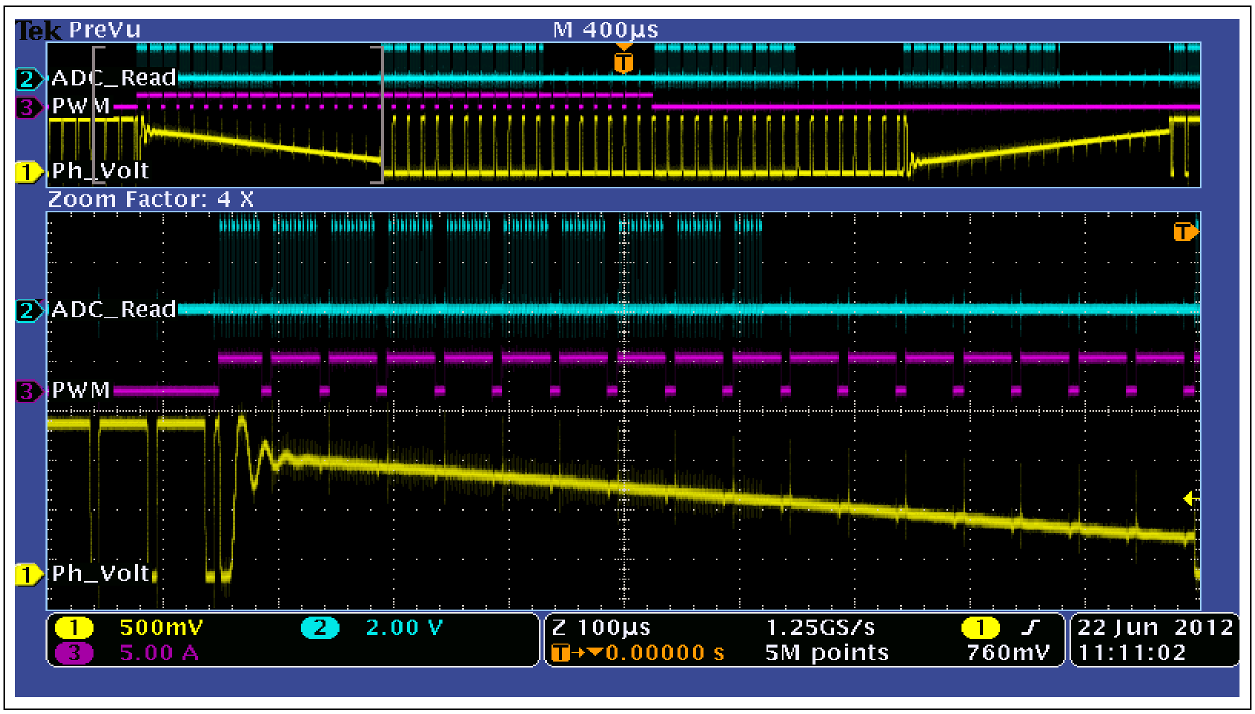


FIGURE 9: BEMF SAMPLING OVER A PWM PERIOD (LOW PWM DUTY CYCLE)

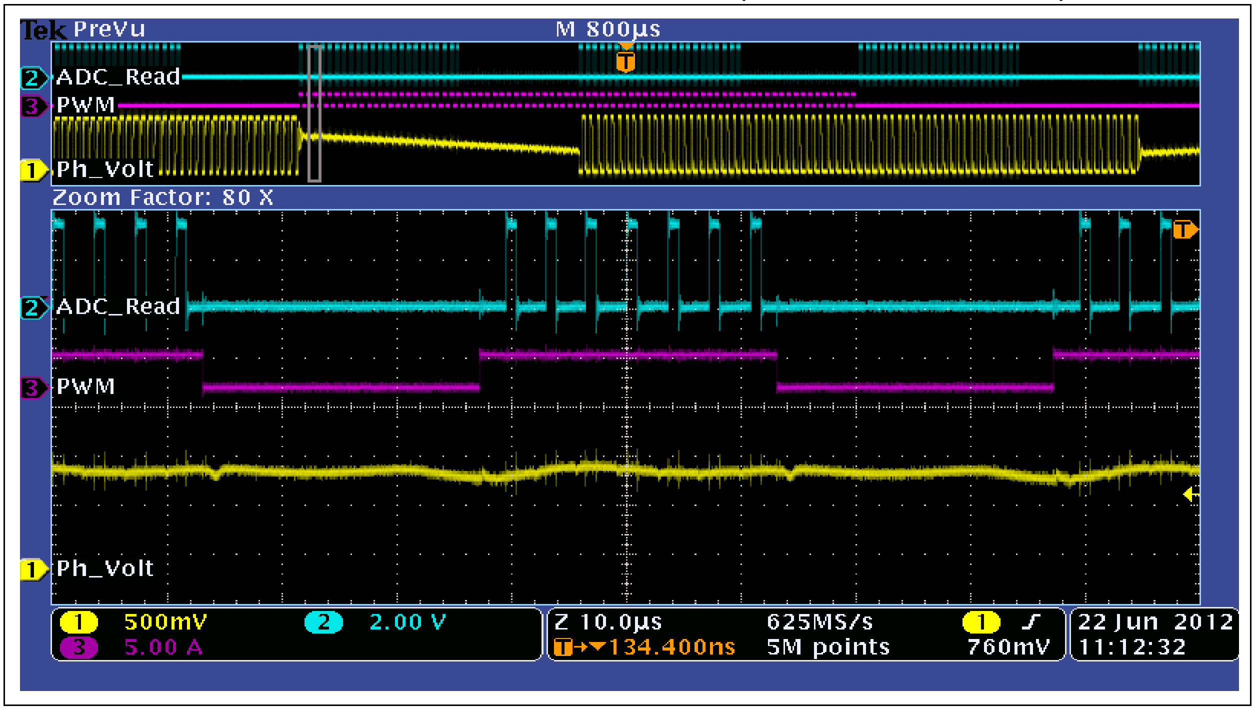
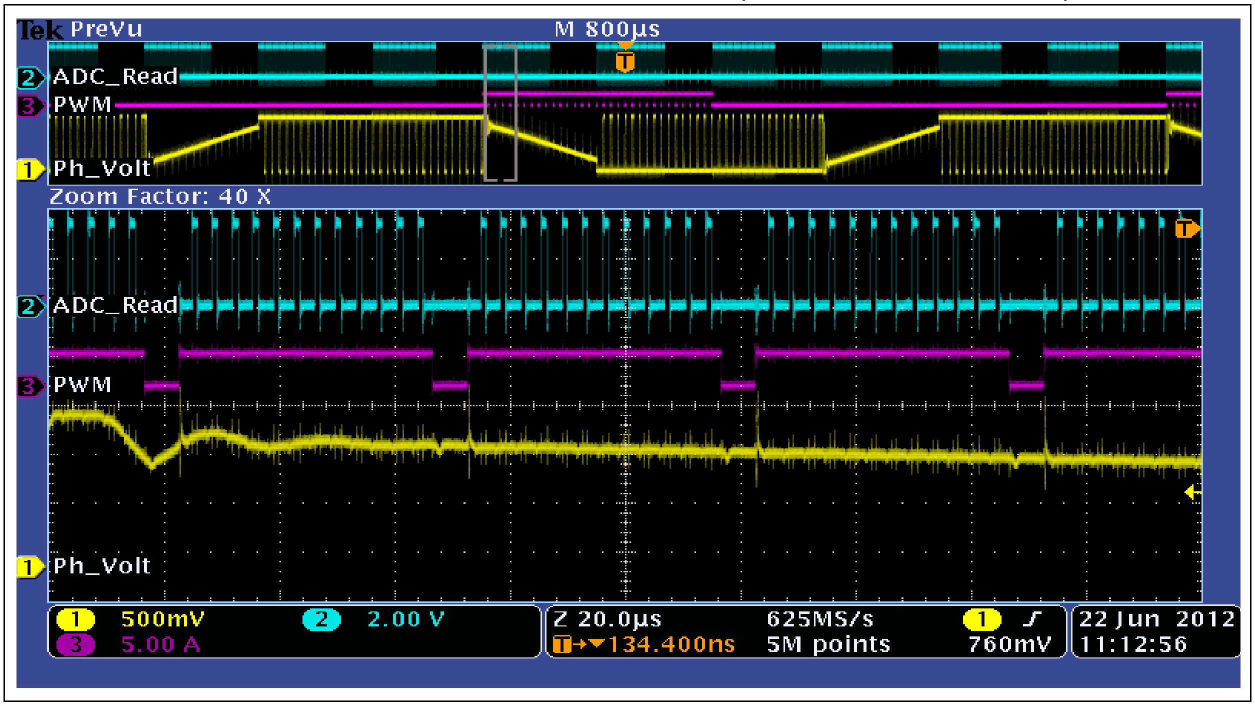


FIGURE 10: BEMF SAMPLING OVER A PWM PERIOD (HIGH PWM DUTY CYCLE)



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 1 shows an example of a 3-input majority function.The majority of the values can be expressed using the AND (^) and OR (v) operators, as shown in Equation 3.

TABLE 1: 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

EQUATION 3: BOOLEAN REPRESENTATION OF THE MAJORITY FUNCTION

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

FILTERING THE BEMF SIGNALS USING THE MAJORITY FUNCTION FILTER

The implementation of this nonlinear filter is based on a 6-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 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 listed in Table 2. They are calculated as follows (Equation 4):

- The first 32 numbers are the index number multiplied by two
- The last 32 values are the index number minus 32, then multiplied by two

EQUATION 4: CALCULATING ARRAY VALUES

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

Second Half: $$Array\ Value\ [N] = (N-32) \cdot 2$$

TABLE 2: ARRAY VALUES

Array Index [N]	Array Value
0	0
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16
9	18
10	20
11	22
12	24
13	26
14	28
15	30
16	32
17	34
18	36
19	38
20	40
21	42
22	44
23	46
24	48
25	50
26	52
27	54
28	56
29	58
30	60
31	62

Array Index [N]	Array Value
32	0
33	2
34	4
35	6
36	8
37	10
38	12
39	14
40	16
41	18
42	20
43	22
44	24
45	26
46	28
47	30
48	32
49	34
50	36
51	38
52	40
53	42
54	44
55	46
56	48
57	50
58	52
59	54
60	56
61	58
62	60
63	62

There are 16 unique array index numbers that represent the true-to-false condition. 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 3 shows the 16 possible numbers that match these two conditions. The values pointed to by these unique indexes are replaced by '1' to indicate that a true-to-false condition has occurred.

TABLE 3: 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 4 provides some numbers that match this condition.

TABLE 4: NUMBERS THAT ARE UNIQUE NUMBER MULTIPLES

Number	6-Bit Binary	Number of Right Shifts	Unique Number Pointed To	6-Bit Binary 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 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 5 shows the numbers that are not multiples of a unique value.

The complete array of filter coefficients, combining the initial array with unique number pointers, is shown in Table 6.

TABLE 5: NUMBERS THAT NEVER POINT TO A UNIQUE VALUE

Number	6-Bit Binary	Numbers Pointed To 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

TABLE 6: COMPLETE 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 7 shows an example of the complete filtering process. The inputs are the noiseless binary representation of the BEMF signals.

Table 8 shows another example of the complete filtering process. In this case, the inputs are the 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 7: DIGITAL FILTERING COMPUTATIONS USING NOISELESS BEMF SIGNALS

ANGLE	BEMF Phase			XOR Masked Phase			AND Masked Phase			Logical Test	Commutation Step	Filter Output	Zero-Cross 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	TRUE	—	—
132	1	0	1	0	0	0	1	0	0	1	3	4	FALSE	—	—

TABLE 8: DIGITAL FILTERING COMPUTATIONS USING NOISY BEMF SIGNALS

Angle	BEMF Phase			XOR Masked Phase			AND Masked Phase			Logical Test N	Commutation Step	Filter Output	Zero-Cross 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	—	—

CONTROL LOOPS

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

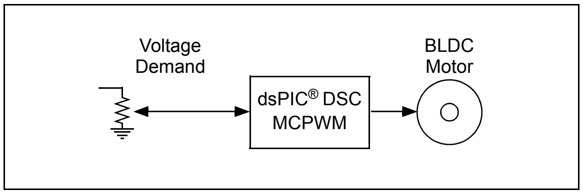
- Open Loop
- Closed Loop
- Closed Loop (PI Controller)

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.

Assume that with PWM, 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, 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



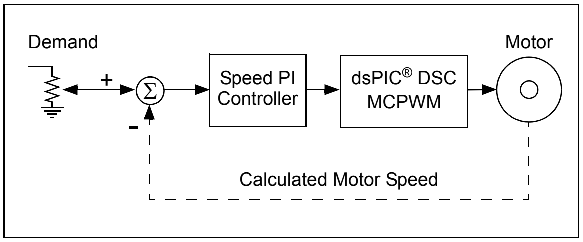
The Analog-to-Digital conversion value is delivered in a 10-bit unsigned integer format; therefore, the possible conversion values are within the range of 0 to 1024. It is required to scale this conversion value to match the PWM duty cycle range. Assuming a PWM frequency of 20 kHz, the PWM duty cycle value varies from 0 to 2000 for microcontrollers running at 40 MIPS, or 0 to 3500 for devices at 70 MIPS (i.e., 16-bit Microchip devices.)

Closed-Loop and Proportional-Integral (PI) Closed-Loop Modes

Closed-Loop mode implemented in the software has two options. The classic Closed-Loop mode attempts to maintain a constant speed by increasing and decreasing the duty cycle with a unit each run.

The Proportional-Integral (PI) mode uses a PI speed controller to calculate the difference between the motor's calculated speed and the speed demand value, and apply the appropriate corrections. The speed demand is typically set as determined by a potentiometer value, scaled to the desired speed range. Figure 12 shows the block diagram of the speed PI Closed-Loop mode.

FIGURE 12: CLOSED-LOOP CONTROL



If we know the number of pair poles and the electrical revolutions per second, it is possible to determine the motor speed. For a motor with two pole pairs (or 4 poles), it is necessary to execute the six-step commutation cycle twice to achieve a complete mechanical revolution. Therefore, it is possible to measure the mechanical revolutions per second through counting the number of six-step commutation cycles and then comparing them to the number of the motor pole pairs.

To measure the mechanical speed (RPM), Timer2 is used for the 30-degree measurement. If it is known that N ticks of Timer2 correspond to 30 electrical degrees, the final mechanical RPM is calculated using the basic motor control formulas.

Once the current speed is calculated, it is then compared to the desired speed set by the scaled value of the potentiometer. 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 5. The PI output is then scaled to match the range of the PWM duty cycle.

EQUATION 5: PI CONTROLLER COMPUTATIONS

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

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

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

Start-up Sequence

The motor start-up sequence is composed of two stages: a user-definable linear start-up ramp and a ramp sustaining time. This sequence is common to all control modes.

During the start-up ramp and the sustaining time, the motor is run in forced spinning commutation. During these phases, the BEMF is not checked. For the start-up ramp, the parameters that can be defined are:

- Ramp Length (in time): How long the ramp takes
- Ramp Final RPM: The target motor speed at the end of the ramp
- Ramp Start-up Duty Cycle: The PWM duty cycle used to spin the motor in forced commutation

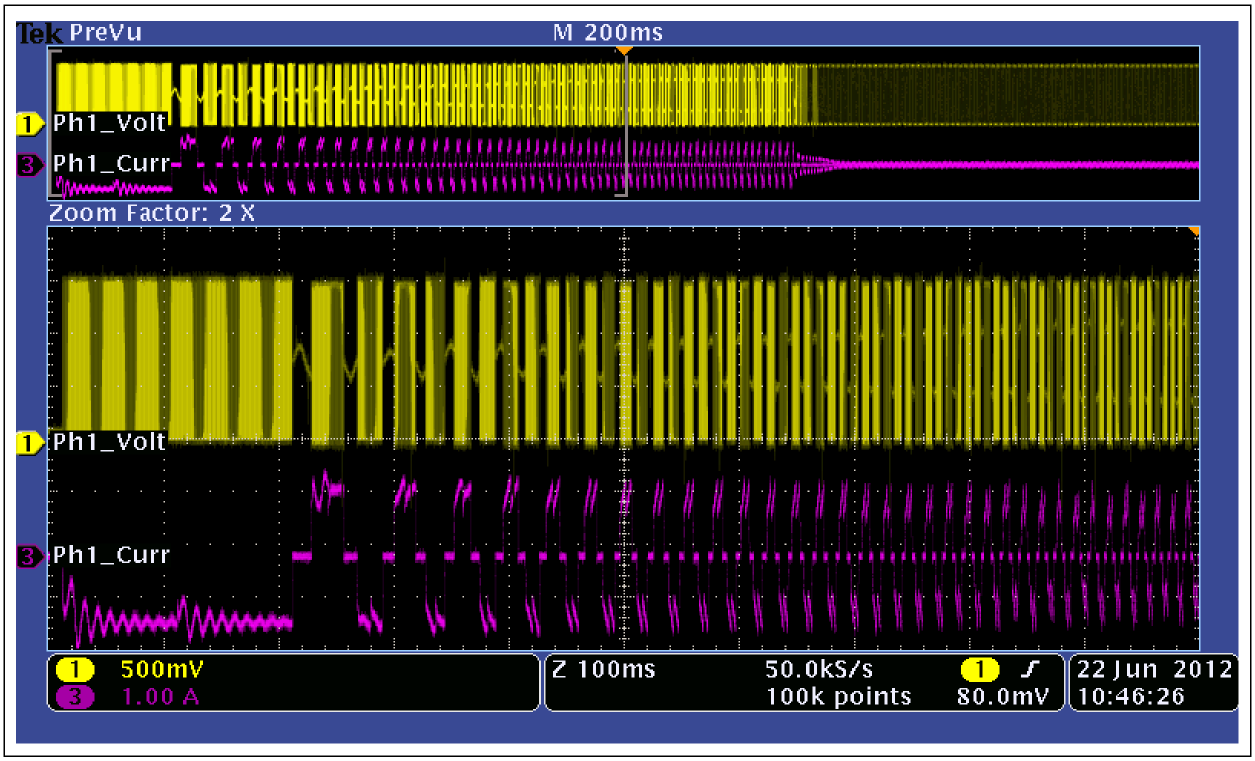
The sustaining time is the required time to keep the motor spinning in forced commutation immediately after the ramp finishes. During this period, if PI Closed-Loop mode is enabled, then the PI controller is trained for the required speed.

For open-loop control, the sustaining time should be kept very low as it does not affect anything. Also note that the ramp's slope is calculated in the accompanying Excel® tuning file.

The scope capture in Figure 13 shows the sequence, from left to right:

- Rotor alignment sequence (exactly 200 ms)
- Start-up ramp, including sustaining time forced commutation (up to 60% of the top screen)
- Switching from forced commutation to closed loop (at about 60% of the unzoomed screen when the current drops very low)
- Closed-loop operation

FIGURE 13: START-UP SEQUENCE



SOFTWARE OVERVIEW

Figure 14, Figure 15 and Figure 16 provide an overview of the BLDC control algorithm's flow.

The state machine (Figure 14) Acknowledges the following states:

- `STATE_STARTING`: The state in which the motor starts. The start-up ramp is executed here.
- `STATE_STARTED`: The motor is running in the selected mode (Open or Closed-Loop mode).
- `STATE_STOPPING`: The state in which a command is issued for motor stopping. Automatically transitions to `STATE_STOPPED`.
- `STATE_STOPPED`: The motor is stopped.
- `STATE_FAULT`: When current Fault or stalling occurs. Basically, the same as `STATE_STOPPING`, but the stop comes from a Fault, not by user command.

The ADC Interrupt Service Routine (ISR) (Figure 15) sets the pre-commutation state, which is the state after the zero-crossing point has been detected. It lasts for 30 electrical degrees, plus the delay the majority filter adds, minus the phase advancing degrees. During the pre-commutation state, the ADC does not sample any signal and the CPU is almost free (as seen in Figure 8, when the BEMF voltage reaches its halfway point).

The pre-commutation state is cleared in Figure 16. When the state is cleared, commutation occurs, the ADC restarts sampling after the blanking counter reaches the preset macro value, and the whole process restarts.

PI Closed-Loop control includes a new PI training routine, that is invoked whenever the sustaining time is not zero. The routine involves calculating the output of the PI controller using real measured values for the input and the reference; the output is not applied until the sustaining time has passed. This process assures a smoother switching from forced sector commutation to PI Closed-Loop operation.

FIGURE 14: APPLICATION MAIN ROUTINE AND STATE MACHINE

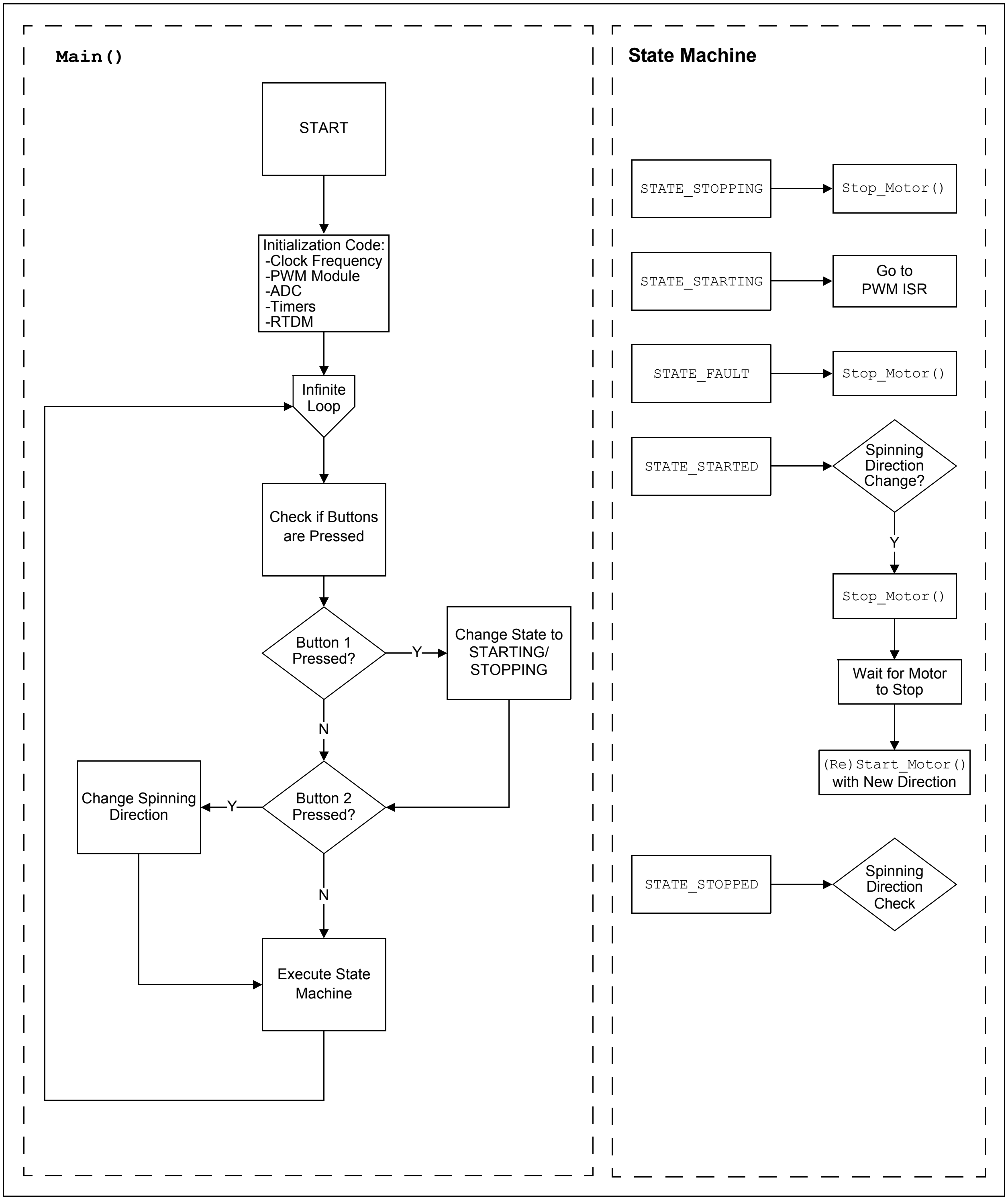


FIGURE 15: INTERRUPT ROUTINE FOR ADC (SENSORLESS COMMUTATION)

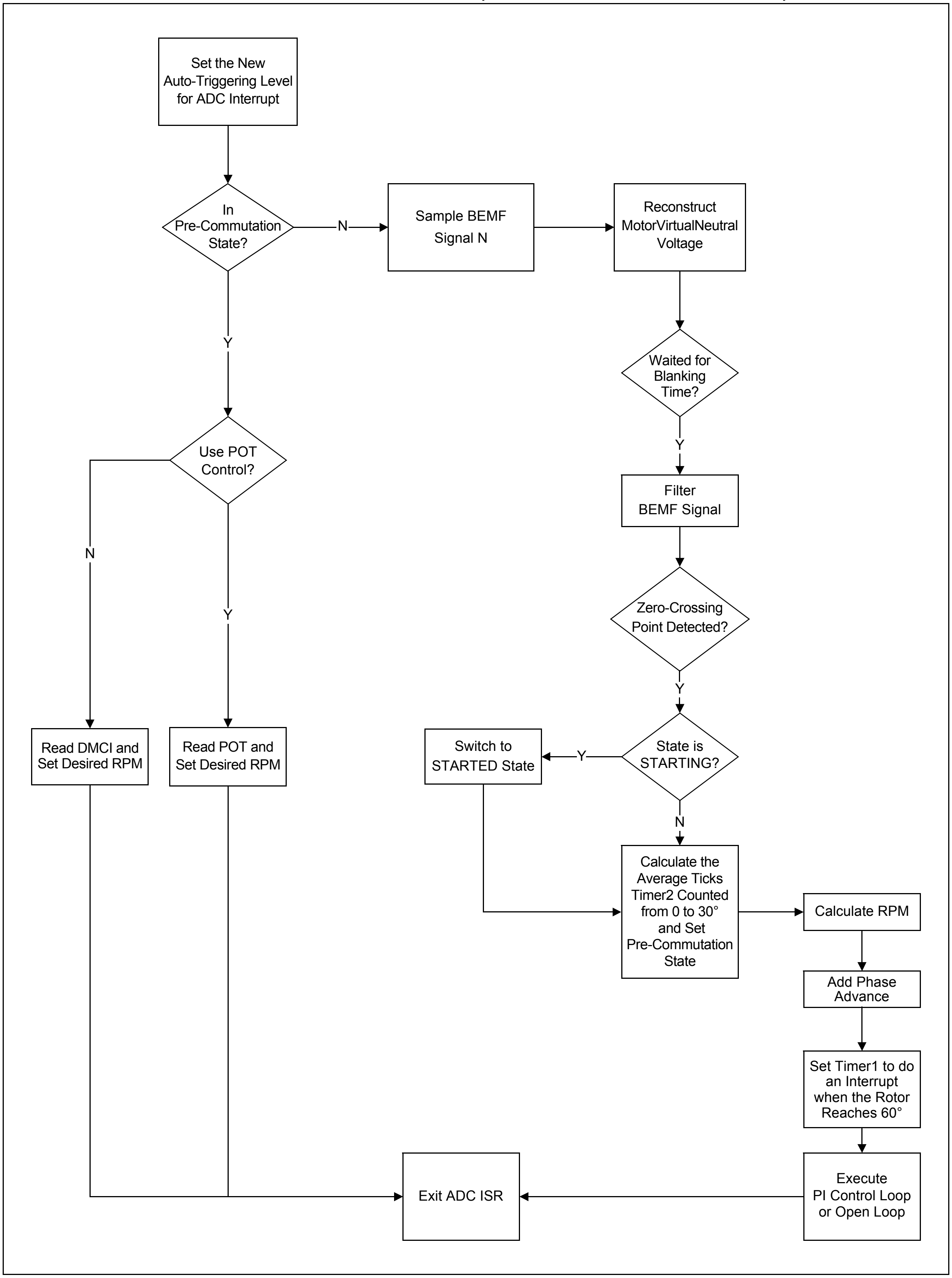
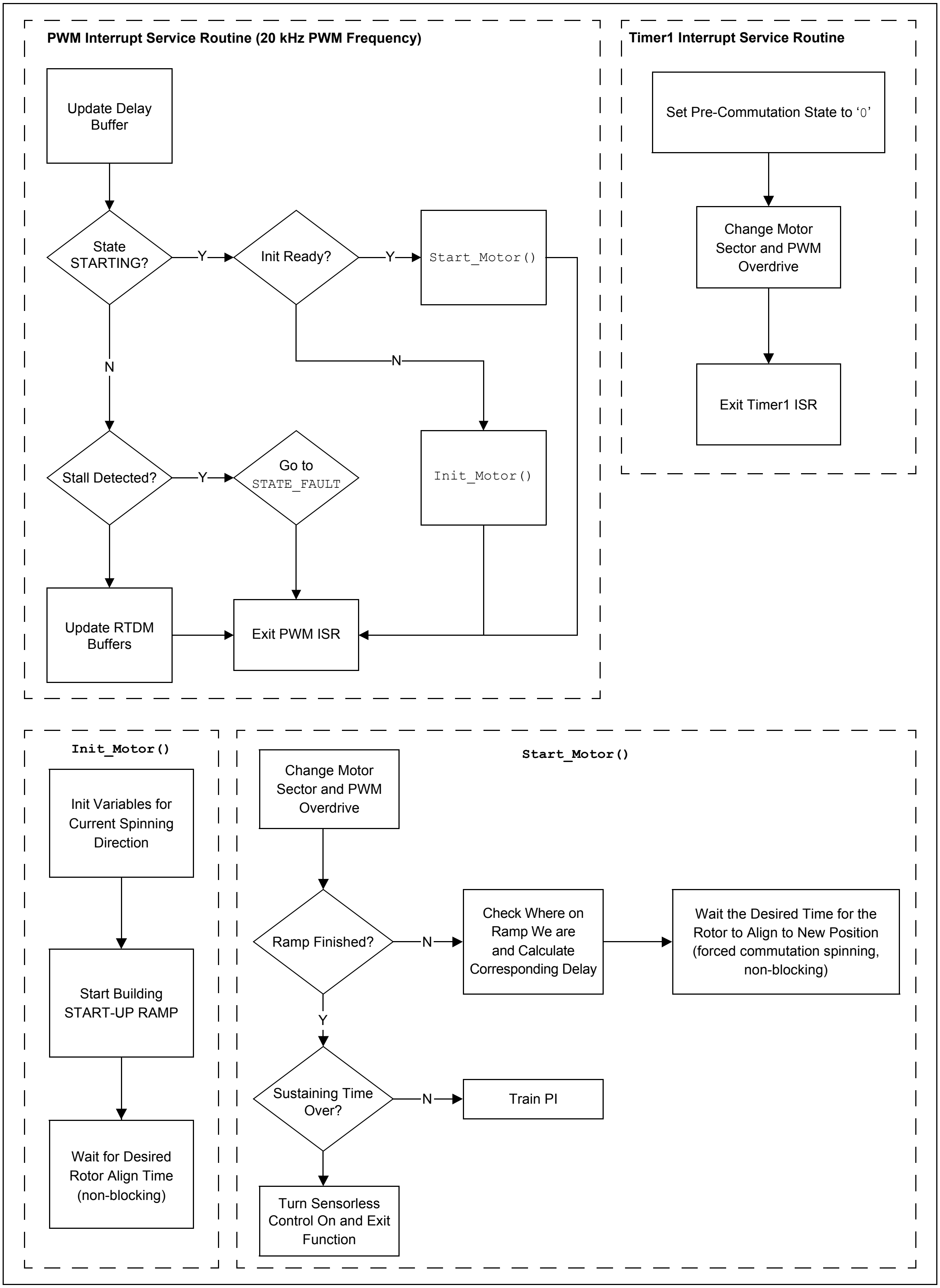


FIGURE 16: INTERRUPT ROUTINES FOR PWM AND TIMER1



CONCLUSION

This application note is intended for the developer who wants to drive a sensorless BLDC motor, using this new BLDC 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 16-bit device-based solution, which does not require external hardware, except for a couple of resistors used to condition the BEMF signals 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 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 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

APPENDIX A: 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.

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.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

== ISO/TS 16949 ==

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale 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.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

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

Printed on recycled paper.

ISBN: 978-1-62076-594-4

Microchip received ISO/TS-16949:2009 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://www.microchip.com/support
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

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

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

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

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

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: 86-10-8569-7000
Fax: 86-10-8528-2104

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

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

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

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

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

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

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 - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

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: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

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

11/29/11