Application Note Abstract
This application note describes the implementation of Sensorless Brushless DC (BLDC) Motor Control using the new generation PSoC® 3 device CY8C3866AXI. The integrated analog multiplexer and comparator with synchronous sampling make it easy to detect Back Electromotive Force (BEMF) zero-crossing.

Introduction
The sensorless BLDC Motor control implements BLDC commutation without a position sensor. Most sensorless BLDC controls are based on zero-crossing detection of BEMF. In this application, BEMF zero-crossing detection is implemented with the help of the PSoC 3 chip CY8C3866AXI without an external comparator, latch, or multiplexer component. The BLDC motor speed is in closed loop control. User interface helps to start or stop the motor and set parameters.

BLDC is constructed with a permanent magnet rotor and wire wound stator poles. Electrical energy is converted to mechanical energy by magnetic forces between the permanent magnet rotor and a rotating magnetic field induced in the wound stator poles.

The traditional brushed DC motor has been replaced by the BLDC motor that has higher reliability, efficiency, and lower noise. BLDC is popular in several fields such as consumer electronics, home appliances, and industrial controls.

Most BLDC motors have a three-phase winding topology with star connection. A motor with this topology is driven by energizing two phases simultaneously, while the other phase is kept afloat. The key to BLDC commutation is to sense the rotor position, then energize the phases that produce the maximum amount of torque. The rotor travels 60 electrical degrees at every commutation step. The appropriate stator current path is activated when the rotor is 120 degrees away from alignment with the corresponding stator magnetic field. It is then deactivated when the rotor is 60 degrees from alignment. Now, the next circuit is activated and the process repeats as shown in Figure 1.

To implement this sequence, it is important to identify the rotor position. The simplest way is to use rotor position sensors, such as Hall-effect sensors or an optical position encoder. However, sensors increase cost and create reliability problems in motors operating in harsh environments. As a result, with the increasing power of embedded computing, the sensorless control method has become more popular.

Most sensorless BLDC controls are based on the zero-crossing detection of BEMF. In Figure 1, every commutation sequence has one winding energized positive, the second negative, and the third winding is left open. The voltage polarity of Back EMF crosses from a positive to negative or from negative to positive (Zero-crossing) between two commutations. Ideally, the zero-crossing of BEMF occurs 30 electrical degrees after the last commutation and 30 electrical degrees prior to the next commutation. By measuring the zero-crossing of BEMF and the 30 degrees time interval, the controller can perform the commutation without a position sensor.
Back EMF Zero-crossing Detection Technology

Figure 2 is a typical drive circuit of a three-phase BLDC motor. In this figure, Va, Vb, and Vc are terminal voltages, and Vn is neutral voltage.

Assuming that phase C is the non-fed phase; you can derive the following equations for the three terminal voltages:

\[ V_a = RI_a + L \frac{dI_a}{dt} + E_a + V_n \]  
**Equation 1**

\[ V_b = RI_b + L \frac{dI_b}{dt} + E_b + V_n \]  
**Equation 2**

\[ V_c = E_c + V_n \]  
**Equation 3**

In these equations, \( E \) is Back EMF, \( I \) is phase current, and \( L \) is inductance.

Because only two currents flow in the stator windings simultaneously, the two phase currents are equal and opposite. Therefore,

\[ I_a = -I_b \]  
**Equation 4**

Adding the three terminal voltage equations, we get:

\[ V_a + V_b + V_c = E_a + E_b + E_c + 3V_n \]  
**Equation 5**

It is evident that at the BEMF zero-crossing points the sum of the three BEMFs equals zero. Therefore, the previous equation is reduced to:

\[ V_a + V_b + V_c = 3V_n \]  
**Equation 6**

For the non-fed phase (zero current flowing), the stator terminal voltage can be rewritten as:

\[ E_c = V_c - V_n \]  
**Equation 7**

From Equation 6 and Equation 7, we get:

\[ 3E_c = 2V_c - (V_a + V_b) \]  
**Equation 8**

Based on these equations, there are four existing BEMF zero-crossing detection methods. Those are as follows:

1. **Compared with Virtual Neutral Point**

   As shown in Equation 7, zero-crossing of BEMF phase C can be detected by comparing Vc and Vn. However, Vn is usually not available in a three-phase BLDC driven circuit. A resistor network is required to construct the virtual neutral point Vn as shown in Figure 3. Voltage dividers and low pass filters are necessary to process the signals. However, the signal/noise ratio at low speed is low because the BEMF amplitude is proportional to motor speed. The second problem is that the low pass filter causes phase delay at high speed.

2. **Compared with 0V during PWM OFF**

   Assume that PWM is only applied on the high side switch and the low side is on during the entire step, ignoring the voltage drop on Vmos and Vd. The current flow during PWM-Off is depicted in Figure 4. In this period:

\[ V_c = V_b = 0 \]  
**Equation 9**

Combine Equations 8 and 9:

\[ 3E_c = 2V_c \]  
**Equation 10**

Therefore, it is not necessary to detect the neutral voltage. The BEMF signal is directly proportional to the float phase voltage. As a result, to detect zero-crossing of BEMF Ec, just sample the terminal voltage Vc and compare it with a voltage of a little more than 0V.
This method requires only a few external components and the applicable speed range is large. This sensing technique can be used in a high voltage or low voltage system with no effort to scale the voltage. However, a low side switch must always be ON. In addition, a voltage reference which is a little higher than 0V for a comparator cannot be easily selected.

3. Compared with Half DC Bus during PWM-ON

Assume that PWM is only applied on the high side switch, and the low side is ON during the complete step. During PWM-ON, ignore the voltage drop on Vmos and Vd. The following equation is obtained:

\[ V_a = V_{bus} \quad \text{Equation 11} \]
\[ V_b = 0 \quad \text{Equation 12} \]

As a result, from Equation 8 on page 2:

\[ 3E_c = 2V_e - V_{bus} \quad \text{Equation 13} \]

To judge zero-crossing of BEMF Ec (Ec=0) is exactly the same as judging the sign of (2Vc-Vbus). This means that we only need to sample terminal voltage Vc and compare it with half Vbus.

This method requires only a few external components and has a high detection range for different speeds. The PWM duty cycle can be up to 100 percent. This sensing technique can be used in a high voltage or low voltage system with no effort to scale the voltage.

There are two typical solutions to implement this control algorithm. One solution is based on a high speed ADC with the capability of synchronous sampling aligning with the PWM. This means that the firmware samples the float terminal voltage and the DC bus voltage in every PWM ON period. Then, perform a comparison in the firmware to do the zero-crossing judgment.

The second solution is to use comparators to perform the voltage comparison with half DC-bus voltage. Then, the judgment is done in the software. In this case, the comparators’ output always contains the PWM carrier waveform in it. Therefore, it is necessary to have a complex firmware to achieve that or use a special PWM scheme or a RC filter to simplify the firmware.

Figure 5. Detecting During PWM-On

These methods require a high MIPS controller to handle the detection algorithm.

4. Simultaneous sampling of Three Phase Terminal Voltages by ADC

According to Equation 8 on page 2, this scheme to detect the zero crossing of BEMF Ec (Ec=0) simultaneously samples the three-phase voltages Va, Vb, and Vc. The float terminal voltage Vc is compared with half of the fed phase voltage (Va+Vb) by the firmware.

The algorithm is simple and does not require a filter and comparator. It does not rely on the PWM mode and can sample voltage signals at any moment of the PWM period. However, it requires high speed and multi-channel ADC. It also requires a high MIPS controller.

Introduction to PSoC 3

PSoC 3 is a new generation PSoC device. Its architecture includes high precision, programmable analog resources that can be configured as ADCs, DACs, TIAs, Mixers, PGAs, OpAmps, and more. They also include enhanced programmable logic based digital resources that can be configured as 8, 16, 24, and 32-bit timers, counters, and PWMs. They can also be configured as more advanced digital peripherals such as Cyclic Redundancy Check (CRC), Pseudo Random Sequence (PRS) generators, and quadrature decoders. Designers have a unique ability to customize this digital system through full featured general purpose PLD-based logic available in PSoC 3. The new architecture also supports a wide range of communication interfaces such as full speed USB, I2C, SPI, UART, CAN, LIN, and I2S.

A block diagram of the PSoC 3 CY8C3866AXI device is shown in Figure 6, highlighting the blocks used in the BLDC sensorless application.

Figure 6. PSoC 3 (CY8C3866AXI) Block Diagram
Digital Subsystem
The PSoC 3 digital subsystem provides unique configurability of functions and interconnect. The BLDC motor control uses these digital resources to implement timers, pulse width modulator (PWM) blocks, control registers, and a hardware look-up table (LUT).

Implementation of BEMF Zero-Crossing Detection in PSoC 3

In this application, the BEMF zero-crossing detection scheme is based on the Compared with Half DC Bus during PWM-ON method described on page 3. Because of the versatility of the PSoC 3 chip, this BEMF zero-crossing detection scheme does not require a high speed ADC or external comparator. The internal multiplexer and comparator with input clock simplifies this procedure. Figure 7 shows the block diagram of PSoC implementation:

Figure 7. PSoC Internal Configuration Diagram

“A, B, and C” are three terminal voltage inputs divided by resistor nets. Half Vbus is an input of DC bus also divided by resistor nets.

A brief introduction of the circuit implementation in PSoC 3 follows:

- The multiplexer can switch between 3-channel BEMF signals in the sampling circuit according to the commutation status.

Analog Subsystem
The PSoC analog subsystem provides the device its second half of unique configurability. The BLDC motor uses dedicated comparators, voltage DACs, and programmable gain amplifiers (PGA) for BEMF detection and protection.

- A comparator inside the PSoC 3 chip has the synchronous sampling feature. The input clock enables the comparator input to be sampled on the rising edge of the clock. By connecting the PWM output to the clock input, the comparator is able to latch BEMF during PWM-On and compares it with half voltage of the DC bus. In other words, the comparator’s function is to hold/sample the phase voltage input. This feature can remove the PWM carrier noise on the comparator output and simplify the firmware implementation. The integrated comparator also has the hysteresis feature. The 10 mv hysteresis ensures the slowly moving voltages or slightly noisy voltage do not cause the comparator output to oscillate when the two input voltages are almost equal. Figure 8 shows the input waveform and output waveform of the comparator. The yellow waveform is the single terminal voltage of the motor. The green one is the output of comparator that is like a Hall sensor output except for the 30 electrical degree phase shift. The microcontroller senses the rotor position by polling on the output of the comparator.

Figure 8. Comparator Input and Output

Unlike the solution of PSoC 1 BEMF detection (refer to the application note AN47936), the PSoC 3 implementation does not require configuration of the sample/hold circuit and filter circuit. As a result, it costs less components (user modules), less pins, and less external components.

Figure 9 shows the detailed implementation in the PSoC® Creator™ schematic design.

Figure 9. BEMF Zero-crossing Detection in PSoC Creator
Firmware Verification of BEMF Zero-crossing Event & Commutation

Figure 10 illustrates BEMF zero-crossing and commutation procedure in the firmware. In this figure, the picture on top is the single phase terminal voltage signal and the picture below is the algorithm implementation for three-phase BEMF zero-crossing signal.

Figure 10. Firmware Process Sequence

- When commutation begins, there is an incorrect pulse impact on the BEMF zero-crossing signal because of the flywheel of the diode. To neglect this impact, F/W discards the first few PWM period samples of BEMF after commutation. This period is called “Blank” period.
- After the “Blank” period, F/W probes the zero-crossing signal in every PWM period.
- Zero-crossing validation causes some delay during the real zero-crossing moment. F/W compensates for this delay based on computing and experiments. The compensating value is different when running in different speeds.
- In ideal cases, the zero-crossing of BEMF occurs 30 electrical degrees after the previous commutation and 30 electrical degrees before the next commutation (refer to Figure 1 on page 1). As result, when a zero-crossing is checked, a 30 electrical degree delay time should be recorded to evoke the next commutation. This is done by a Timer16 component. This component is also used as a free running timer to stamp the commutation and zero-crossing period for period calculation.

BLDC Commutation

The Universal Digital Block (UDB) in the PSoC 3 chip consists of a combination of uncommitted logic similar to programmable logic devices (PLDs), structured logic (datapaths), and a flexible routing scheme.

In addition to the flexibility of the UDB array, PSoC 3 provides configurable digital blocks targeted at specific functions. These blocks can include 16-bit timer/counter/PWM blocks, I2C slave/master/multi-master, full speed USB, and CAN 2.0.

All these features in the PSoC 3 digital system are useful for implementing BLDC commutation logic by HW.
Free Run Stages of BLDC Motor

The BEMF amplitude of BLDC is proportional to the motor speed. If the motor is still, there is no BEMF. When the motor runs at low speed, the BEMF is too weak to be detected. Therefore, before the BEMF can be measured by the firmware, there is a stage called the free run stage. This stage drives the BLDC motor step by step to acquire the initial BEMF zero-crossing signal before running sensorless control.

When run at a constant speed with open-loop stepping stimulation, the rotor position is approximately 90 electrical degrees ahead of that when run correctly under sensorless control. As a result, the BEMF zero crossings cannot be sensed. To observe the zero-crossings, it is necessary to accelerate the motor at a certain rate.

Therefore, at this stage, the PWM duty cycle increases gradually and the commutating period is longer than usual. A ramp-up table in F/W defines the timeout of every commutating period. The value of the table content results from experiments based on different motor types. When sufficient valid zero-crossing events are detected, F/W enters the normal synchronous running stage. If F/W cannot detect sufficient valid zero-crossing events during a period of time, an error occurs and BLDC must stop and wait for the next round of free running stage.

Figure 13. BLDC Starting Sequence

PI Speed Close Loop

The PI control algorithm is useful in a continuous control system. There are two basic PI control algorithms: absolute mode and increment mode PI control algorithm. The following equation is a discrete expression of the position mode of the PI algorithm.

\[ u_k = K_p e_k + \sum_{i=1}^{k-1} K_i e_i + u_0 \] \hspace{1cm} Equation 14

In this equation, \( e_k \) is the speed error, \( K_1 \) is the integration coefficient, and \( K_p \) is the proportional coefficient.

The disadvantages of the absolute mode PI algorithm are:

- Switching between closed-loop and open loop has a system impulsive force, which results in unstable running of motor.
- Output of integration action is related to all past status. This increases the workload of MCU and accumulates systematic error.

These disadvantages can be solved by using the increment mode PI algorithm, shown in the following equation:

\[ \Delta u_k = u_k - u_{k-1} = K_p (e_k - e_{k-1}) + K_i e_k \] \hspace{1cm} Equation 15

The control increment is the output, which is added to the current control input. The implementation on an 8-bit controller becomes easier.

Figure 14 shows the block diagram of PSoC PI closed-loop speed control.

Figure 14. Speed PI Close Loop

Speed Calculation

Motor speed is calculated with a few codes by using the digital system of the chip. See Figure 15 for details.

Speed is calculated once in a commutation period. LUT_spd is a 3*1 look up table. It generates one rising edge every commutation period. A 16-bit counter captures the counter value to a storage FIFO. An interrupt is hung up so that the CPU can read the counter value immediately. At the same time, the counter resets to zero and restarts.

Figure 15. Speed Calculation
Over Current Protection

In this application, over current protection is done by analog components in PSoC 3. As shown in Figure 16, CurrIn is voltage on the current sensing resistor. This current signal is amplified through the internal PGA and compared with a reference value through the internal comparator. The maximum current reference is set by an 8-bit voltage DAC. The user can dynamically change this maximum value by FW. The output of the comparator is ANDed by a logic circuit with PWM so that it can disable the PWM when the motor is in the over current situation.

Figure 16. Current Sample Circuit

Meanwhile, firmware can also probe the output of the comparator. After FW detects output of comparator at high level for a short period of time, the controller stops the motor for overcurrent protection.

User Interface

The DVK board provides two CapSense buttons and a 5-segment slider for menu/parameter selection and a character LCD panel for information display.

The capacitive sensing (CapSense) component provides a versatile and efficient means to measure capacitance in applications such as touch sense buttons, sliders, and proximity detection.

Reference

For more details, refer to the following Cypress application notes:

1. Sensorless BLDC Motor Control Based on CY8C24x33, AN47936
2. BLDC Closed Loop Control Based On CY8C24x33, Bill Jiang and Jemmey Huang, AN42102
3. Brushless DC Motor Control, Andrey Magarita, AN2227
4. PSoC 3 Sensored Motor Control, AN53595
Appendix 1: Board Photograph

Figure 18. Photograph of CY8CKIT-021 Kit
Appendix 2: Board Schematics
Test Point

Power Supplier

Test
PSwC3 BLDC / Stepper Motor Control Kit

<table>
<thead>
<tr>
<th>Size</th>
<th>Document Number</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Power Supplier</td>
<td>V1.0</td>
</tr>
</tbody>
</table>

Date: Thursday, September 17, 2010

Sheet: 4 of 7

[+] Feedback
J8, J9, J10 select BLDC working mode.
For Hall sensor mode, connect pin1 with pin2;
For sensor-less mode, connect pin2 with pin3.

Voltage Divider
HALL Input

Motor

- R60: 3.3K
- R79: 3.3K
- R76: 3.3K
- R70: 1K
- R71: 1K
- R72: 1K
- C10: 10nF/10V
- C6: 10nF/10V
- C8: 10nF/10V

HALL_VCC

C41
0.1uF/25V

J11
3
2
1

C42
HALL_VCC

M_Phaso1
M_Phaso2
M_Phaso3
M_Phaso4

J6
1
2
3
4

C44X1

0.1uF/25V

+5V

- HALL_VCC

Motor

Feedback
About the Authors

Name: Dino Gu
Title: Application Engineer Sr.
Contact: qdgu@cypress.com

Name: Jemmey Huang
Title: Product Apps Manager Sr.
Contact: jhu@cypress.com
Document History
Document Title: Sensorless BLDC Motor Control based on CY8C3866AXI
Document Number: 001-58261

<table>
<thead>
<tr>
<th>Revision</th>
<th>ECN</th>
<th>Orig. of Change</th>
<th>Submission Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>2829303</td>
<td>QDGU/JHU</td>
<td>12/16/2009</td>
<td>New application note</td>
</tr>
<tr>
<td>*A</td>
<td>3012573</td>
<td>QDGU</td>
<td>08/19/2010</td>
<td>Update example project to fit PSoC Creator 1.0 Beta 5.0</td>
</tr>
</tbody>
</table>

PSoC is a registered trademark of Cypress Semiconductor Corp. PSoC Creator is trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

© Cypress Semiconductor Corporation. 2009-2010. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and/or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress’ product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.