

### Atmel AVR443: Sensor-based Control of Three Phase Brushless DC Motor

#### Atmel AVR 8-bit Microcontrollers

#### Features

- Less than 5µs response time on Hall sensor output change
- Theoretical maximum of 1600k RPM
- Over-current sensing and stall detection
- Support for closed loop regulation
- UART, TWI and SPI available for communication

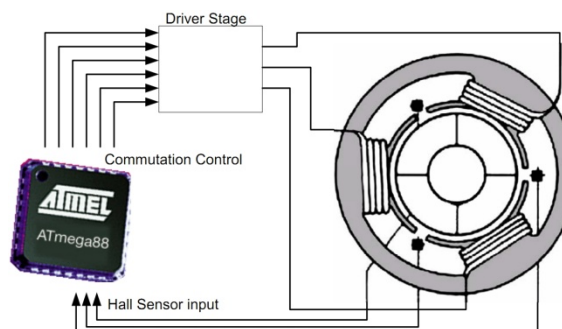
#### Introduction

The use of Brushless DC (BLDC) motors is continuously increasing. The reason is obvious: BLDC motors have a good weight/size to power ratio and excellent acceleration performance, require little or no maintenance and generate less acoustic and electrical noise than universal (brushed) DC motors.

In a Universal DC motor, brushes control the commutation by physically connecting the coils at the correct moment. In BLDC motors the commutation is controlled by electronics. The electronics can either have position sensor inputs that provide information about when to commutate or use the Back Electromotive Force generated in the coils. Position sensors are most often used in applications where the starting torque varies greatly or where a high initial torque is required. Position sensors are also often used in applications where the motor is used for positioning. Sensor-less BLDC control is often used when the initial torque does not vary much and where position control is not in focus, e.g. in fans.

This application note describes the control of a BLDC motor with Hall Effect position sensors (referred to simply as Hall sensors). The implementation includes both direction and open loop speed control.

**Figure 1. Atmel ATmega48 controlling a BLDC motor with Hall sensors.**



## Table of Contents

---

1. Theory of Operation .....	3
1.1 Operation of Fundamental BLDC Motor .....	3
1.2 Implementation – Hall Sensor Based Control of BLDC Motor .....	4
1.3 Software Description .....	6
1.4 Performance of Current Implementation .....	7
2. Revision History .....	8

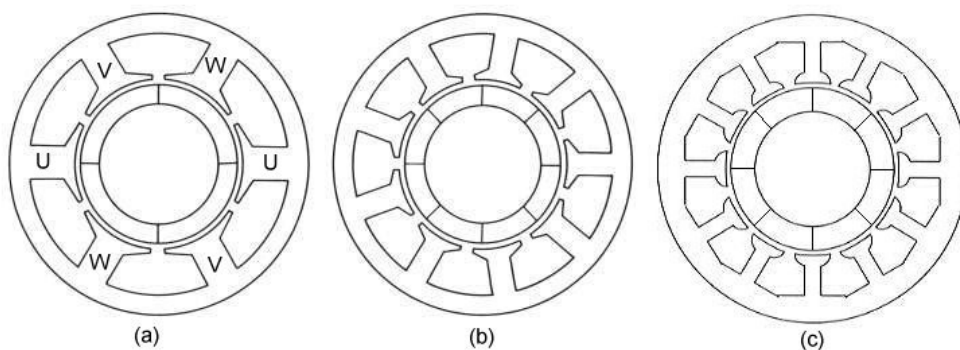
## 1. Theory of Operation

Control of a BLDC motor with position sensors can be implemented on sufficiently powerful microcontroller featuring basic hardware peripherals such as Analog to Digital Converter (ADC) and a timer with PWM output. The Atmel® ATmega48 covers the requirements for BLDC motor control well – with resources left for other tasks still. Other relevant tasks could e.g. be communication using SPI, UART or TWI protocols.

A three-phase BLDC consists of a Stator with a number of coils. The fundamental three-phase BLDC motor has three coils (see Figure 1). Usually the three coils are referred to as U, V and W. In many motors the fundamental number of coils is replicated to have smaller rotation steps and smaller torque ripple.

The rotor in a BLDC motor consists of an even number of permanent magnets. The number of magnetic poles in the rotor also affects the step size and torque ripple of the motor. More poles give smaller steps and less torque ripple. Figure 1-1 shows different configurations of motors with more than one fundamental set of coils and multiple poles.

**Figure 1-1. BLDC motors of different types. Motor (a) has two fundamental sets of coils and four poles, (b) has three sets of coils and eight poles and (c) has four sets of coils and eight poles.**



The fact that the coils are stationary while the magnet is rotating makes the rotor of the BLDC rotor lighter than the rotor of a conventional universal DC motor where the coils are placed on the rotor.

### 1.1 Operation of Fundamental BLDC Motor

To simplify the explanation of how to operate a three-phase BLDC motor a fundamental BLDC with only three coils is considered.

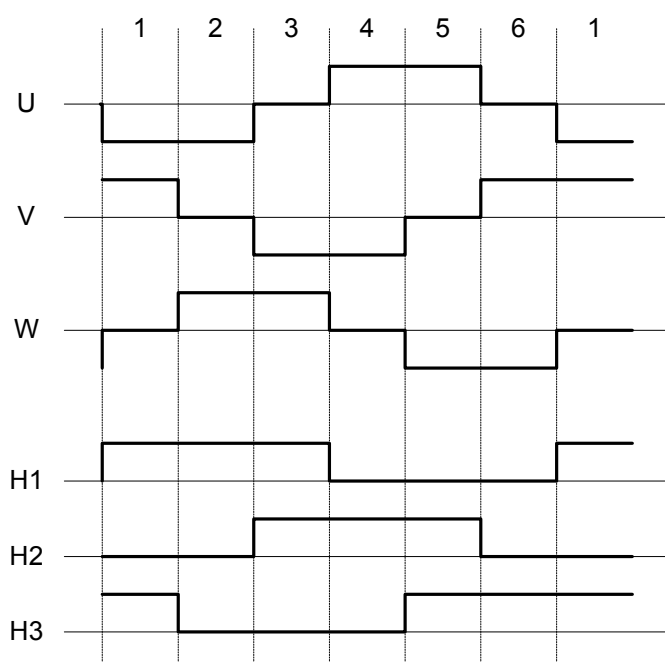
To make the motor rotate the coils are energized (or “activated”) in a predefined sequence, making the motor turn in one direction, say clockwise. Running the sequence in reverse order makes the motor run in the opposite direction. One should understand that the sequence defines the direction of the current flow in the coils and thereby the magnetic field generated by the individual coils. The direction of the current determines the orientation of the magnetic field generated by the coil. The magnetic field attracts and rejects the permanent magnets of the rotor. By changing the current flow in the coils and thereby the polarity of the magnetic fields at the right moment – and in the right sequence – the motor rotates. Alternation of the current flow through the coils to make the rotor turn is referred to as commutation.

A three-phase BLDC motor has six states of commutation. When all six states in the commutation sequence have been performed the sequence is repeated to continue the rotation. The sequence represents a full electrical rotation. For motors with multiple poles the electrical rotation does not correspond to a mechanical rotation. A four-pole BLDC motor uses two electrical rotation cycles per mechanical rotation. When specifying the number of Rotations per Minute subsequently, the number of electrical rotations is referred to unless otherwise mentioned.

The most elementary commutation driving method used for BLDC motors is an on-off scheme: A coil is either conducting (in one or the other direction) or not conducting. Connecting the coils to the power and neutral bus induces the current flow (accomplished using a driver stage). This is referred to as square wave commutation or block commutation. An alternative method is to use a sinusoidal type waveform. This application note covers the block commutation method.

The strength of the magnetic field determines the torque and speed of the motor. By varying the current flow through the coils the speed and torque of the motor can be varied. The most common way to control the current flow is to control the (average) current flow through the coil. This can be accomplished by switching the supply voltage to the coils on and off so that the relation between on and off time defines the average voltage over the coil and thereby the average current.

**Figure 1-2. Current flow through the coils/magnetic field generated by the coils U, V and W in the six commutation states for a BLDC motor. Hall sensor outputs are also shown.**



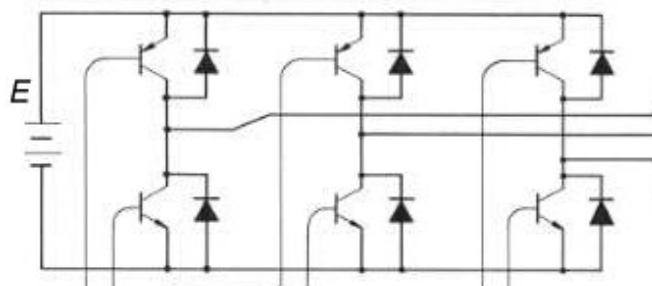
For BLDC motors the commutation control is handled by electronics. The simplest way to control the commutation is to commute according the outputs from a set of position sensors inside the motor. Usually Hall sensors are used. The Hall sensors change their outputs when the commutation should be changed (see [Figure 1-2](#)). Quite simple!

Secondary functions for the electronics in a BLDC motor control application is to ensure that the speed is as desired either by open or closed loop control. In either case it is however also recommended to have stall detection (blocked motor) and overload detection.

## 1.2 Implementation – Hall Sensor Based Control of BLDC Motor

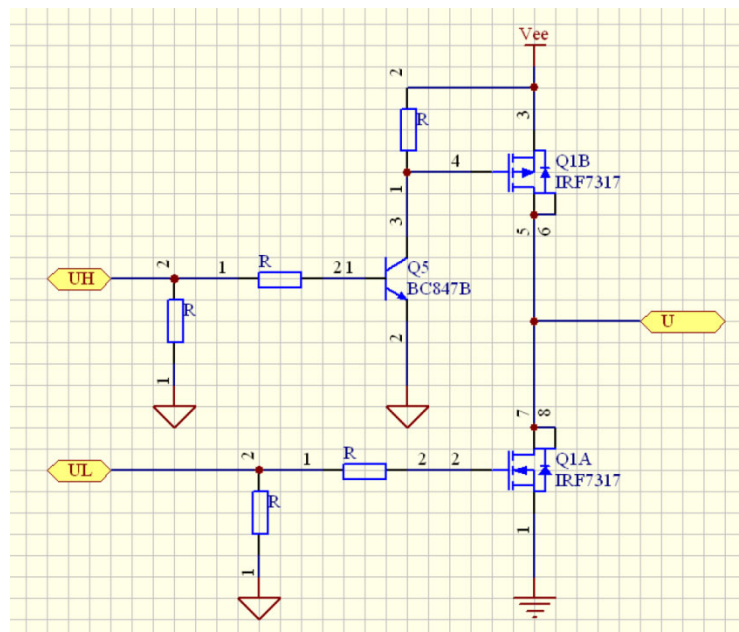
The implementation is controlling a BLDC motor in open loop. The motor current is measured and speed is monitored, to be able to respond to stall and overload situations. Three PWM channels are connected to the low side of the driving Half-bridges to control the speed of the motor. A BLDC motor driver stage, consisting of three half-bridges, can be seen in [Figure 1-3](#).

Figure 1-3. Typical driver-bridge for a three-phase BLDC motor.



The driver stage is implemented slightly different in practice to accommodate for the lacking possibility to control the high side FETs directly from logic output levels from the AVR®. An implementation using dedicated FET drivers is used for the Atmel ATAVRMC100 development board, an alternative implementation could be the one described in [Figure 1-4](#).

Figure 1-4. Driver circuit for the U, V and W motor coils (only U driver shown).



Three PWM channels, OC0A, OC0B and OC2B, control the low side of the driver bridge (e.g. UL on [Figure 1-4](#)). This gives the possibility to control the current flow using hardware based PWMs with a minimum of timer resources in use. This controls the speed of the motor: by varying the duty cycle of the PWM output the current flow and thereby the speed (and torque) of the motor is controlled.

It is also possible to have PWM based control of the high side of the bridge, but that would require all the Atmel ATmega48 timers. Further, it would require either that shoot through protection is integrated in the driver circuit or that dead time is handled in software. If active braking is used it can be desired to use PWM channels for both high and low side of the drivers to distribute the power dissipation more evenly over the effect transistors. However, in most applications this is not required.

A single ADC channel is used to measure the current flow. The ADC has a resolution of 10 bits and uses an external 2.5V reference; this gives an accuracy of approximately 2.4mV, which is sufficient for over-current detection as the voltage over a 0.22Ω shunt resistor is 220mV when 1A flows through it. If required the ADC can be triggered by the PWM to measure current when not switching or run continuously with a given sampling frequency. A second ADC channel is used to measure an analog voltage (e.g. a potentiometer) used as speed reference for adjusting the motor speed.

The Hall sensor outputs are connected to three pins on PORTB, which all features interrupt on level change (pin change interrupt). In case the Hall sensors outputs change their logic levels, an interrupt is executed and the commutation state corresponding to the new Hall sensor output is determined. Note that the lowest pins on a PORT are used intentionally to speed optimize the decoding of the Hall signals.

An overview of the resources used is listed in [Table 1-1](#), more details can be found in the software documentation (readme.html).

**Table 1-1. Resources used for motor control.**

Resource	Usage
ADC0 and ADC4	Speed reference and current measurements
PORTD[3] – Timer Counter 2: OC2B	Control of low side drivers (WL)
PORTD[5,6] – Timer Counter 0: OC0[A,B]	Control of low side drivers (VL, UL)
PORTD[7,4,2]	Control of high side drivers (UH, VH, WH)

It is worth mentioning that the hardware resources for UART, SPI and TWI communication are still available if required. Note that the increased response time to hall sensor change should be considered if interrupts are used for communication (or other peripherals).

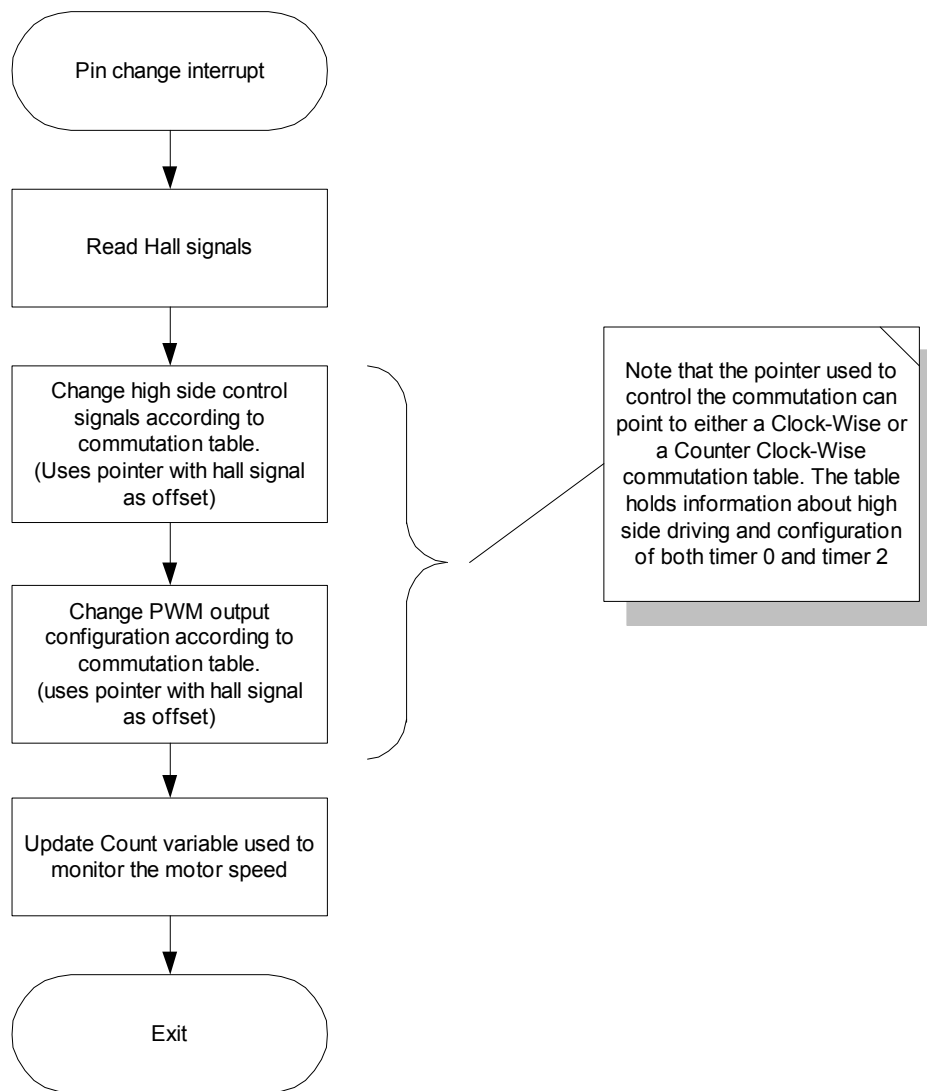
## 1.3 Software Description

All code is implemented in C language using the IAR™ EWAVR 4.12A compiler (free up to 4kB of binary output). The source code is documented using Doxygen - refer to the readme.html file found along with the source code for code documentation. The readme.html will open the html based code documentation.

The most important function, the Pin Change Interrupt routine, handling the commutation change upon a change in the Hall sensor output, is described by the flowchart found in [Figure 1-5](#).

Note that the implementation locks a number of registers for certain variables to ensure fast execution of the interrupt handling the commutation. The compiler rarely uses the registers locked - unless standard libraries functions for handling strings are used. If a conflict should emerge this, it can be taken care of by recompiling the standard libraries.

**Figure 1-5. Flowchart of the pin change interrupts handling the communication.**



## 1.4 Performance of Current Implementation

- 8-bit resolution on the speed control (limited by PWM resolution)
- Code size is 427 bytes (high code size optimization)
- Response time to Hall sensor signal changes is below 5µs
- Pin-Change interrupt routine (Hall input) takes app 50 CPU cycles. At 8MHz this gives a giving a theoretical maximum of 1600k RPM ( $8\text{MHz}/(50 \text{ cycles} \times 6 \text{ commutation states}) \times 60 \text{ sec/min}$ ) - if over-current control and communication is not considered

## 2. Revision History

Doc. Rev.	Date	Comments
2596C	07/2013	Syntax correction and new document template
2596B	02/2006	Doxygen comments updated and corrected
2596A		Initial document release



**Atmel Corporation**

1600 Technology Drive  
San Jose, CA 95110  
USA

**Tel:** (+1)(408) 441-0311

**Fax:** (+1)(408) 487-2600

[www.atmel.com](http://www.atmel.com)

**Atmel Asia Limited**

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG

**Tel:** (+852) 2245-6100

**Fax:** (+852) 2722-1369

**Atmel Munich GmbH**

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY

**Tel:** (+49) 89-31970-0

**Fax:** (+49) 89-3194621

**Atmel Japan G.K.**

16F Shin-Osaki Kangyo Building  
1-6-4 Osaki, Shinagawa-ku  
Tokyo 141-0032  
JAPAN

**Tel:** (+81)(3) 6417-0300

**Fax:** (+81)(3) 6417-0370

© 2013 Atmel Corporation. All rights reserved. / Rev.: 2596C-AVR-07/2013

Atmel®, Atmel logo and combinations thereof, AVR®, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.