AVR443: Sensor-based control of three phase Brushless DC motor

8-bit AVR Microcontrollers

Application Note

Features

- **Less than 5us 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**

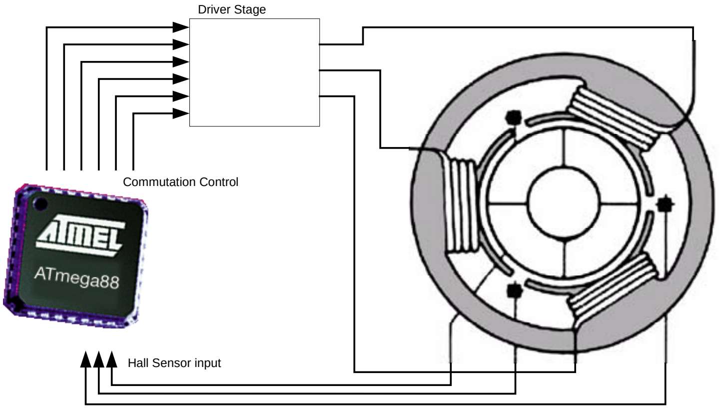
1 Introduction

The use of Brushless DC (BLDC) motors is continuously increasing. The reason is obvious: BLDC motors are having a good weight/size to power ration, have excellent acceleration performance, requires little or no maintenance and generates 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. Sensorless 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 described 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 1-1. ATmega48 controlling a BLDC motor with Hall sensors.



Rev. 2596B-AVR-02/06

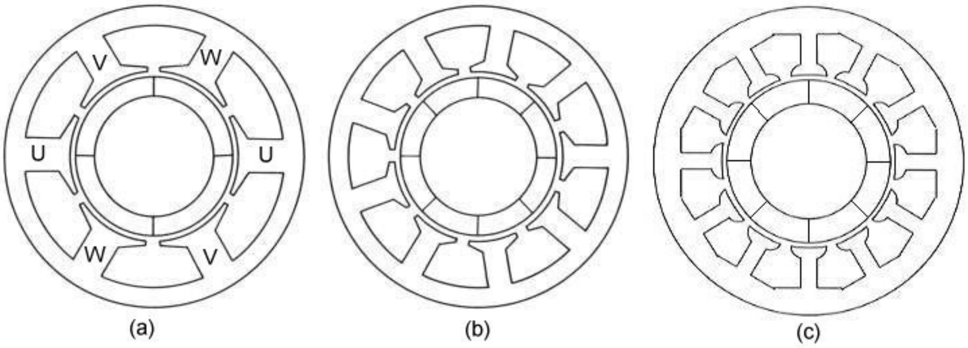
2 Theory of operation

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 has a number of coils. The fundamental three phase BLDC motor has three coils (see Figure 1-1). Usually the three coils are referred to as U, V and W. In many motors the fundamental number of coils are 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 gives smaller steps and less torque ripple. Figure 2-1 shows different configurations of motors with more that one fundamental set of coils and multiple poles.

Figure 2-1. 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 motor lighter than the rotor of a conventional universal DC motor where the coils are placed on the rotor.

2.1 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.

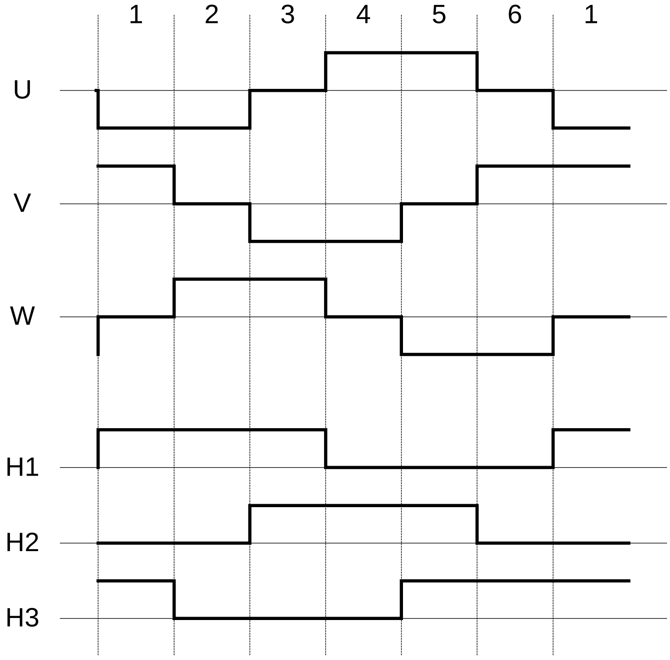
To make the motor rotate the coils are energized (or “activated”) in a predefined sequence, making the motor turn in one direction, say clockwise. Running the sequence in reverse order the motor run in the opposite direction. One should understand that the sequence defines the direction of the current 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 use four electrical rotation cycles to have one 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 on 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 trapezoidal 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 force and speed of the motor. By varying the current flow thought 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 2-2. 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.



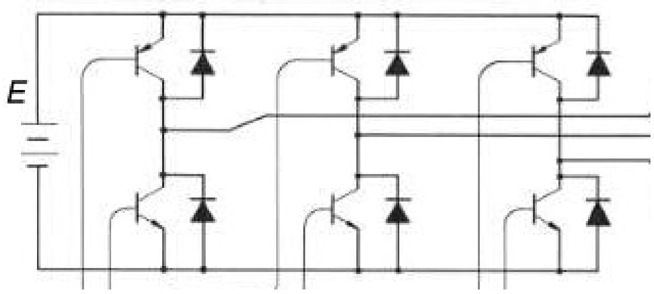
For BLDC motors the commutation control is handled by electronics. The simplest way to control the commutation is to commutate according the outputs from a set of position sensors inside the motor. Usually Hall sensors are used. The Hall sensors change their outputs when the commutation should be changed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

2.2 Implementation - Hall sensor based control of BLDC motor

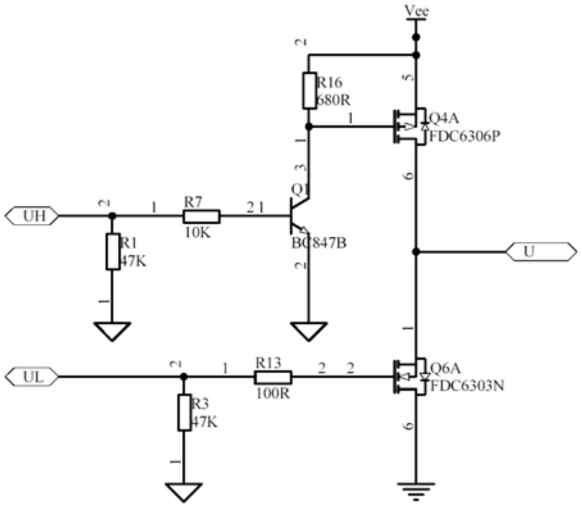
The implementation is controlling a BLDC motor in open loop. The motor current is measured and speed is monitored, to be able to respond to stall and overload situations. Three PWM channels are connected to the low side of the driving Half-bridges to control the speed of the motor. The typical driver stage for a BLDC motor can be seen in Figure 2-3.

Figure 2-3. 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. Figure 2-4 shows the actual implementation of the driver for each coil. Other implementations can be used if desired. The default state of the drivers is off. The driver stage can deliver app 2 A continuously at 12V.

Figure 2-4. 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 2-4). 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 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 breaking 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 0.22 ohm 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 a potentiometer voltage for setting the motor speed.

The Hall sensor outputs are connected to the 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 are listed in Table 2-1.

Table 2-1. Resources used for motor control.

Resource	Usage
ADC	Current measurements
PORTD[3] – Timer Counter 2: OC2B	Control of low side drivers – W coil
PORTD[5,6] – Timer Counter 0: OC0[A,B]	Control of low side drivers
PORTD[7,4,2]	Control of high side drivers

It is worth mentioning that the hardware resources for UART, SPI and TWI communication are still available if required. Note that it is not recommended to use interrupts for communication, unless the potential effect on the commutation response time is considered first.

2.3 Software description

All code is implemented in C language using the IAR EWAVR 3.20C compiler (free up to 4kB of binary output). The functions available in the implementation are listed below. Only the most important function, the Pin Change Interrupt routine, handling the commutation change upon a change in the Hall sensor output, is described by flowchart.

Note that the implementation locks a number of registers for certain variables to ensure fast execution of the interrupt handling the commutation. The registers locked are rarely used when not using the compilers standard libraries for handling strings. Even if a conflict should emerge this can be taken care of by recompiling the standard libraries.

```
void Init_MC_timers( void )
```

Initialize the Timer 10 and timer 2 to run in Phase and frequency correct PWM mode (symmetric PWM). The base frequency is set to 32kHz (can be reduced at the expense of lower resolution on the speed control). The functions also ensures that the timers are counting in synch.

```
void Init_MC_Pin_Change_Interrupt( void )
```

Sets up the pins used to sense the Hall sensor signals to generate interrupt if the pin level changes (both rising and falling edge).

```
void Init_ADC( void )
```

Sets up the ADC with prescaler value 4, which means a maximum sample speed of CPU frequency divided by 52 (13*4). With the ADC measuring the speed set point and shunt voltage, this gives a reaction time of two samples for detecting over-current.

```
void Set_Direction( unsigned char direction )
```

Set the commutation table pointer up to point at either the clockwise or counter clockwise direction table. Note that it is not recommended to change direction without first reducing the speed of the motor, preferably stopping it fully.

```
void Set_Speed( unsigned char speed )
```

Updates the output compare registers of the timer 0 and timer 2 which control the duty cycle of the PWM output and thereby the speed of the motor. The method used ensures that that all PWM channels are behaving same duty cycle.

```
unsigned char Get_Speed( void )
```

Returns the speed of the rotor. Not implemented.

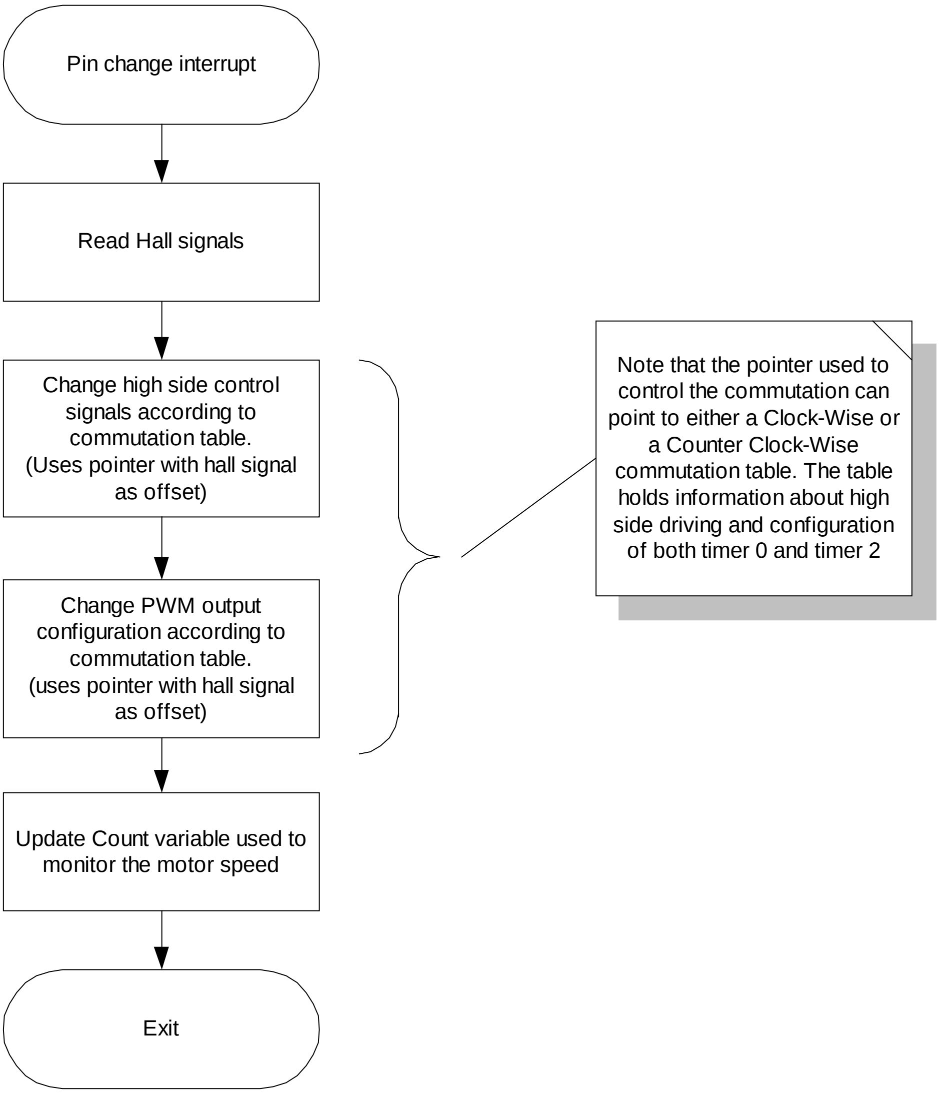
```
__interrupt void PCINT0_ISR( void )
```

Updates the PWM outputs controlling the low side of the driver and the IO controlling the high side of the driver. To ensure a speed optimal interrupt the variables used in the interrupt are placed and in reserved registers (locked for this purpose only). Further, the information required to do the commutation is placed in tables that can be accessed very efficiently using the Hall sensor input signals as offset. The interrupt is described by the flowchart in figure Figure 2-5.

```
void Release_motor( void )
```

Floats the outputs from the AVR connected to the driver stage. This will disable the drivers to ensure that not current flows into the motor coils. Not implemented.

Figure 2-5. Flowchart of the pin change interrupt handling the commutation.



2.4 Performance of current implementation

- 8 bit resolution on the speed control.
- Code size is app 500 bytes (current implementation is 350 bytes)
- Response time to Hall sensor signal changes is below 5us.
- Pin-Change interrupt routine (Hall input) takes app 50 CPU cycles. At 8MHz this gives a giving a theoretical maximum of 1600k RPM (8MHz/(50 cycles * 6 commutation states) * 60 sec/min) - if over-current control and communication is not considered.

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

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 ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, 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 OF 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 product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2006. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are®, AVR®, AVR Studio® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.