AVR452: Sensor-based Control of Three Phase Brushless DC Motors Using AT90CAN128/64/32

8-bit AVR® Microcontrollers

Application Note

1. Features

- **Less than 3µs response time on Hall sensor output change**
- **Theoretical maximum of 3478K RPM (Electrical RPM)**
- **Support for closed loop regulation, over current, stall and overload detection.**
- **CAN, UART, TWI and SPI available for communication**

2. Introduction

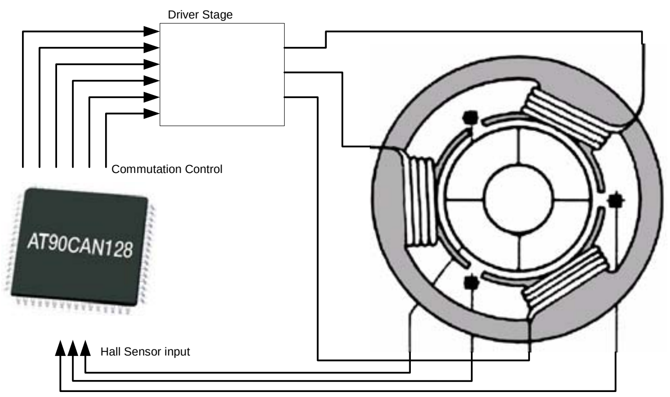
The use of Brushless DC (BLDC) motors is continuously increasing. The reason is obvious: BLDC motors have a good weight/size to power ratio, have excellent acceleration performance, requires little or no maintenance and generates less acoustic and electrical noise than universal (brushed) DC motors.

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.

7616A–AVR–03/06

Figure 2-1. AT90CAN128 controlling a BLDC motor with Hall sensors



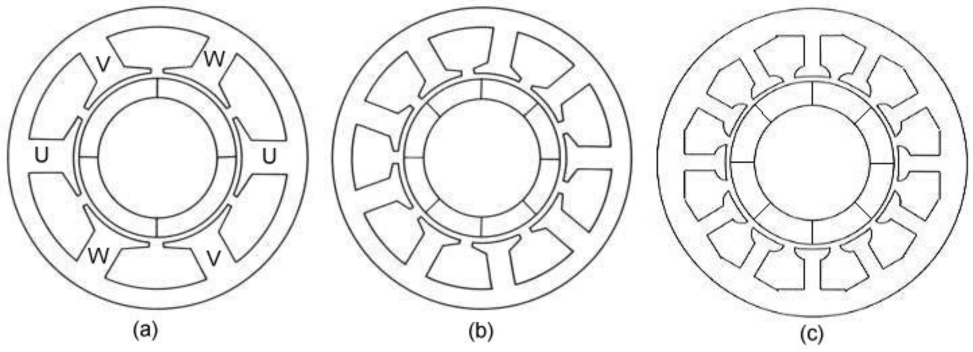
3. Theory of Operation

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 AT90CAN128 covers the requirements for BLDC motor control well – with resources left for other tasks still. Other relevant tasks could include for example, communication using CAN, SPI, UART or TWI protocols.

A three phase BLDC consists of a Stator which has a number of coils. The fundamental three phase BLDC motor has three coils (see Figure 2-1). 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 3-1 shows different configurations of motors with more than one fundamental set of coils and multiple poles.

Figure 3-1. 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.

3.1 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 and two magnetic poles is considered.

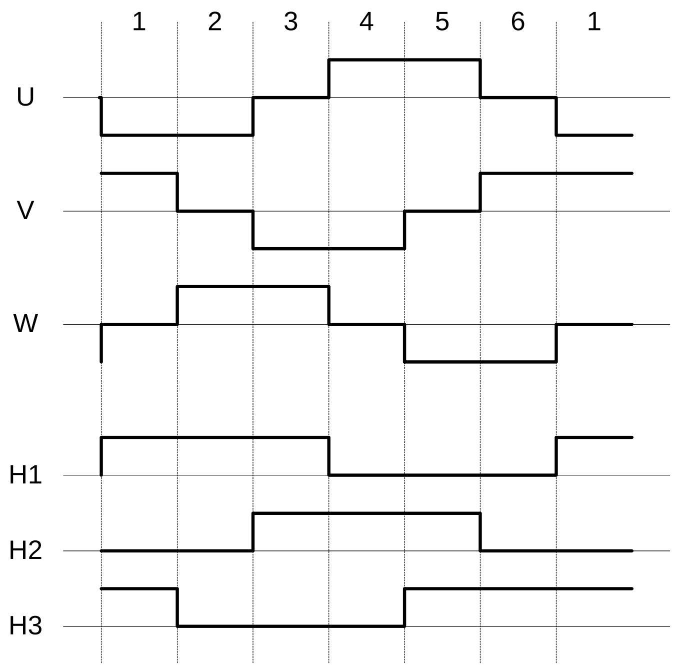
To make the motor rotate the coils are energized (or “activated”) in a predefined sequence, making the motor turn in one direction, say clockwise. Running the sequence in reverse order the motor run in the opposite direction. One should understand that the sequence defines the direction of the current 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 two electrical rotation cycles to have one mechanical rotation.

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 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 3-2. 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



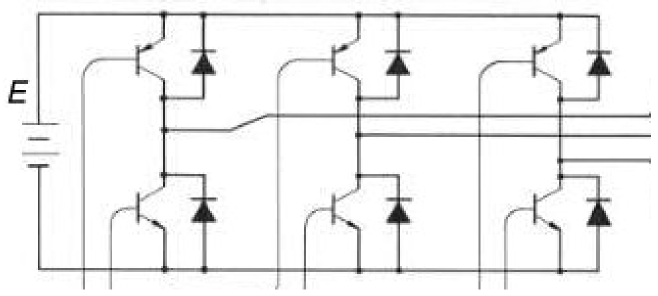
For BLDC motors the commutation control is handled by electronics. The simplest way to control the commutation is to commutate according the outputs from a set of position sensors inside the motor. Usually Hall sensors are used. The Hall sensors change their outputs when the commutation should be changed (see Figure 3-2).

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.

3.2 Implementation - Hall sensor based control of BLDC motor

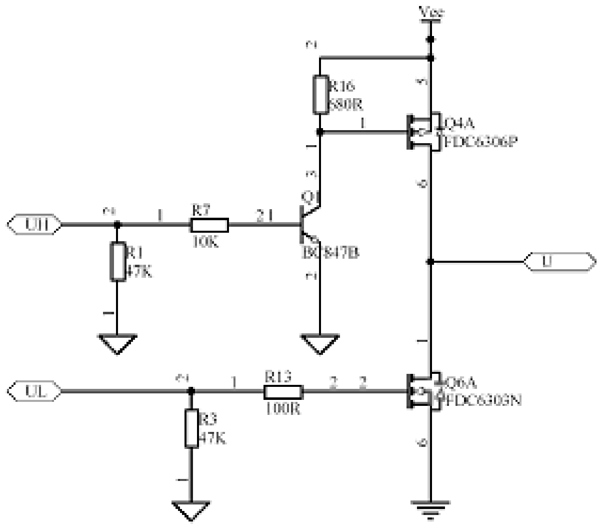
The implementation is controlling a BLDC motor in open loop. The motor speed is measured, the motor current can be monitored (not implemented) to be able to respond to stall and overload situations. Three PWM channels are connected to the low side of the driving Half-bridges to control the speed of the motor. The typical driver stage for a BLDC motor can be seen in Figure 3-3.

Figure 3-3. 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 3-4 shows the actual implementation of the driver for each coil. Other implementations can be used if desired.

Figure 3-4. Driver circuit for the U, V and W motor coils (only U driver shown)



Three PWM channels, OC1A, OC1B and OC1C, control the low side of the driver bridge (e.g. UL on Figure 3-4. 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 as well, that would require the AT90CAN128 timer 3 in addition to timer 1. In the implementation distributed with this application note the high side drivers are controlled by general purpose IO.

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 can be use used to measure the current flow (not implemented). The ADC has a resolution of 10 bits and can use the internal 2.56V reference; this gives an accuracy of approximately 2.4 mV, which is sufficient for over-current detection as the voltage over a 0.22Ohm shunt resistor is 220 mV 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 can be used to measure a potentiometer voltage for setting the motor speed (useful if a digital communication interface is not used to control the speed of the motor).

The Hall sensor outputs are connected to the three pins on PORTE which all features interrupt on level change (External interrupt). In case of 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

An overview of the resources used are listed in Resources used for motor control

Table 1. Resources used for motor control

Resource	Usage
PORTB[5,6,7] – Timer Counter 1: OC1[A,B,C]	Control of low side drivers [UL,VL,WL]
PORTB[2,3,4]	Control of high side drivers [UH,VH,WH]
PORTE[5,6,7]	Hall sensor input [A,B,C]

It is worth mentioning that the hardware resources for CAN, 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.

3.3 CAN interface

The implementation use CAN network to control the motor speed and direction, it is possible to start and stop the motor. There is also specific frames to retrieve the measured speed of the motor through CAN network. Table 2.2 details frames content to interface with the firmware.

Table 2. Frame used for Can communication

Type	ID	Data	Action
Data frame	0x120	don't care	Stop the motor
Data frame	0x121	don't care	Start the motor
Data frame	0x122	HH LL	Set motor speed (speed=OxHHLL)
Data frame	0x123	XX	Set motor direction (XX=O=CCW or XX=1=CW)
Remote frame	0x124	DLC=2	To get back measured speed value

CAN communication is developed under interruption, for high speed it should be better to implement polling mechanism for communication process with CAN bus.

Average CAN interrupt duration is 160 machine cycles, CAN interrupt is executed only when a new command is send to the AT90CAN128/64/32 to modify direction, speed or to get measured speed.

3.4 Software description

All code is implemented in C language using the IAR EWAVR 4.11A. compiler (free up to 4kB of binary output). The available functions in the implementation are listed below. Only the most important function, the External 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 IAR compiler rarely used these specific registers, except when 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_timer1_pwm( void )
```

Initialize the Timer 1 to run in fast PWM mode.

OCRx is cleared on compare match.

```
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 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 int speed )
```

Update the output compare registers of the timer 1 which control the duty cycle of the PWM output and thereby the speed of the motor. The method used ensures that all PWM channels are updated at the same time (and keep the same duty cycle).

```
void Init_Speed_Timer0(void)
```

Initialize the Timer 0 to overflow at Clkio/1024, Timer 0 (8bits) overflow every 32 ms with 8 MHz Crystal. This time base is used to measure the motor speed.

```
__interrupt void TIMER0_OVF_ISR(void)
```

Timer 0 interrupt occurs every 32 ms with 8 MHz crystal, Measured_speed variable is updated there.

```
void Run_motor(void)
```

Enable the OCRx output to run the motor

```
void Stop_motor(void)
```

Disable the OCRx output to stop the motor, 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.

```
void Hall_ISR(void)
```

Update the PWM outputs controlling the low side of the driver and the IO controlling the high side of the driver. To ensure an optimal speed 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 3-5.

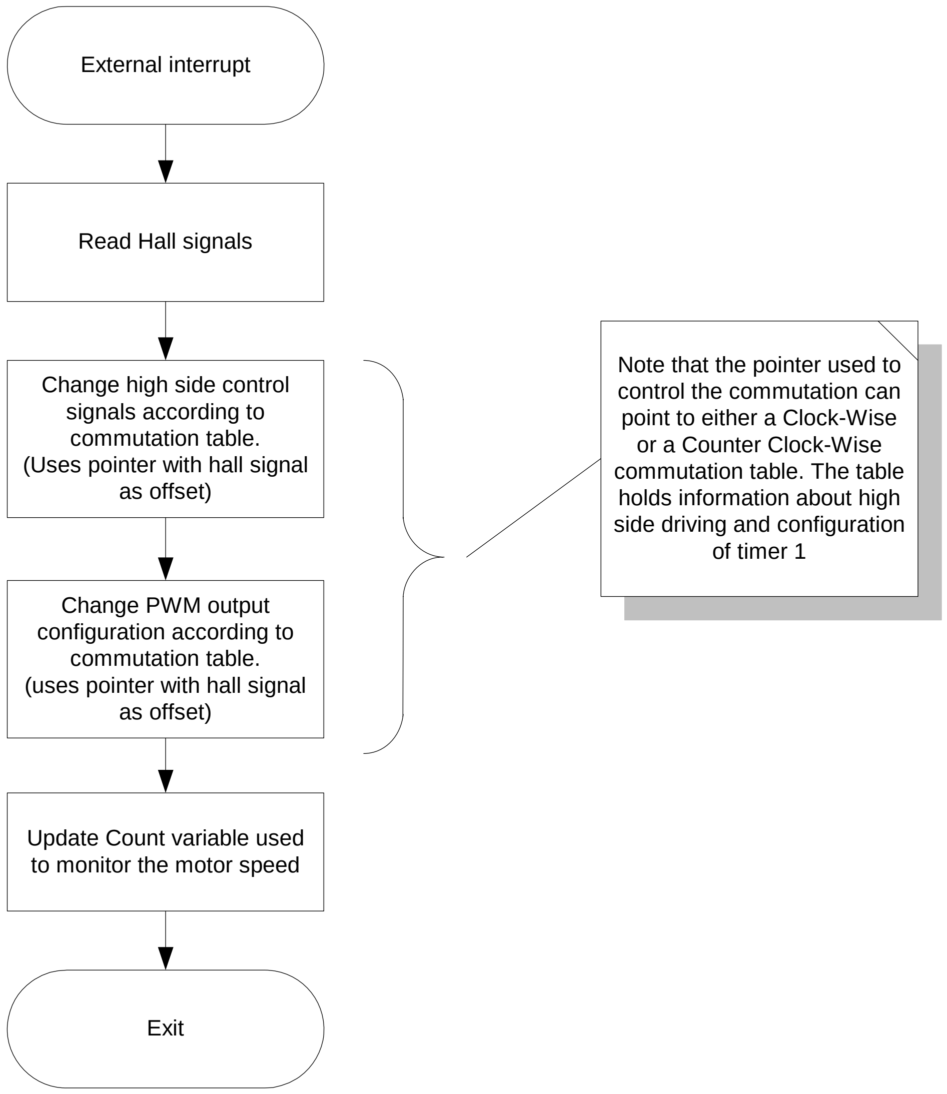
```
void can_init(void)
```

Initialize CAN Macro for 100 Kbps with 8 MHz crystal, CAN channel 0 receive all frames, CAN channel 1 send frames with measured speed.

```
__interrupt void CAN_ISR(void)
```

Can interrupt subroutine, update speed, direction variable regarding can frame received, send back measured speed, start and stop the motor.

Figure 3-5. Flowchart of the external interrupt handling the commutation


External interrupt
Read Hall signals
Change high side control signals according to commutation table. (Uses pointer with hall signal as offset)
Change PWM output configuration according to commutation table. (uses pointer with hall signal as offset)
Update Count variable used to monitor the motor speed
Exit
Note that the pointer used to control the commutation can point to either a Clock-Wise or a Counter Clock-Wise commutation table. The table holds information about high side driving and configuration of timer 1

3.5 Performance of current implementation

- 10-bit resolution on the speed control.
- Code size is app 1400 bytes.
- Response time to Hall sensor signal changes is below 5us.

External interrupt routine (Hall input) takes app 23 CPU cycles. At 8 MHz this gives 17,25 µs (50 cycles * 6 commutations) by electrical rotation. It gives 3478K electrical rotation per minute, If using a motor with 4 pair of poles it gives a theoretical maximum of 869K mechanical RPM. (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