AVR444: Sensorless control of 3-phase brushless DC motors

8-bit AVR® Microcontrollers

Application Note

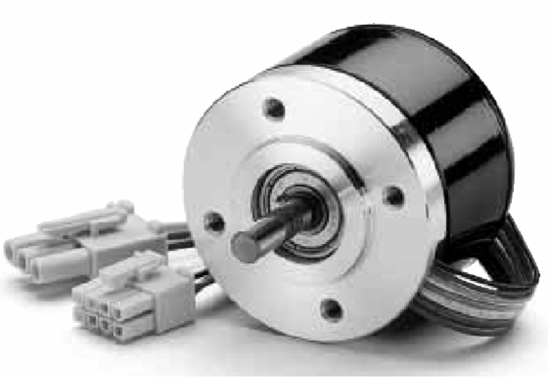
Features

- Robust sensorless commutation control.
- External speed reference.
- Overcurrent detection/protection.
- Basic speed controller included.
- Full source code in C.
- Source code can be adapted to a new motor by changing parameters.
- Several I/O pins/peripherals not used for motor control. Can be used for other purposes, such as status LEDs, communication or user input.
- Works without modification on the pin and source code compatible devices ATmega48, ATmega88 and ATmega168.

1 Introduction

This application note describes how to implement sensorless commutation control of a 3-phase brushless DC (BLDC) motor with the low cost ATmega48 microcontroller. A general solution, suitable for most 3-phase BLDC motors on the market is presented. The full source code is written in the C language, no assembly is required. Adaptation to different motors is done through the setting of parameters in the source code.

The ATmega48/88/168 devices are all pin and source code compatible. The only difference is memory sizes. This application note is written with ATmega48 in mind, but any reference to ATmega48 in this document also applies to ATmega88/168.



Rev. 8012A-AVR-10/05

2 Theory of operation

This document does not explain the theory behind brushless DC motors in detail. Instead it focuses on the aspects that are important to sensorless control. For an introduction to three phase brushless DC motors, please consult application note AVR443: Sensorbased control of three phase Brushless DC motor.

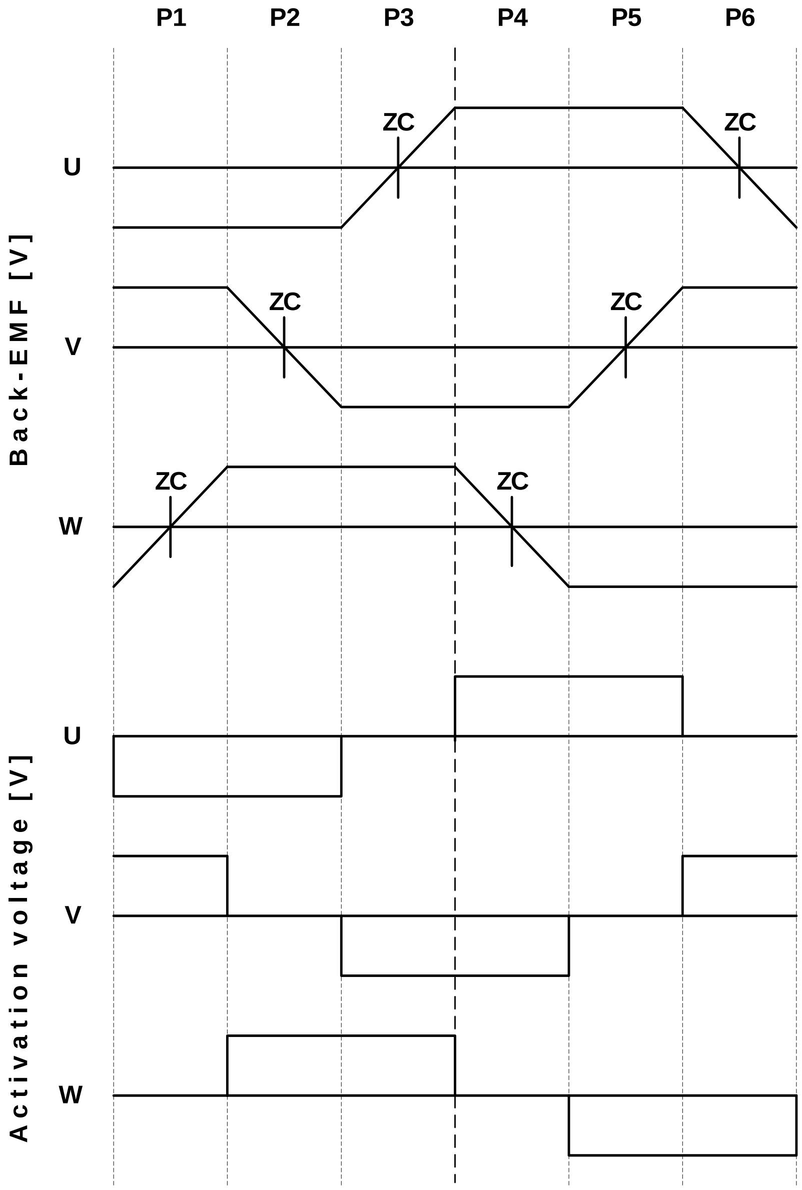
2.1 Waveforms

The generally accepted definition of a BLDC motor is a permanent magnet motor with trapezoidal back-EMF, as opposed to the sinusoidal back-EMF found in permanent-magnet synchronous motor. This application note applies to BLDC motors with trapezoidal back-EMF.

The typical trapezoidal back-EMF waveforms and corresponding driving voltages of a 3-phase BLDC are shown in Figure 1. In every commutation step, one phase winding is connected to positive supply voltage, one phase winding is connected to negative supply voltage and one phase is floating. The back-EMF in the floating phase will result in a “zero crossing” when it crosses the average of the positive and negative supply voltage. The zero crossings are marked as ZC in Figure 1.

The zero crossing occurs right in the middle of two commutations. At constant speed, or slowly varying speed, the time period from one commutation to zero-crossing and the time period from zero-crossing to the next commutation are equal. This is used as basis for this implementation of sensorless commutation control.

Figure 1. Waveforms



The floating phase, where the zero crossing must be detected, changes for every commutation step. One ADC channel for each phase winding is needed to detect zero crossings.

2.2 Startup

The magnitude of the back-EMF is directly proportional to the motor speed. This makes it extremely difficult to detect zero-crossings at low speed, since the signal to noise ratio is very small. The sensorless commutation scheme presented in this application note will thus not work during startup and at very low speeds. A number of strategies for sensorless startup of brushless DC motors have been proposed over the years. These differ in complexity and computational complexity, and there does not seem to be one solution that fits all. Furthermore, many of these startup methods are patented.

In this application note, a simple "blind" startup is used. A table of inter-commutation delays for the first few commutation is stored in flash. This sequence is executed without attention to the back-EMF feedback. The control is then passed over to the sensorless commutation controller. This rather simple method works very well when the motor load is known in advance.

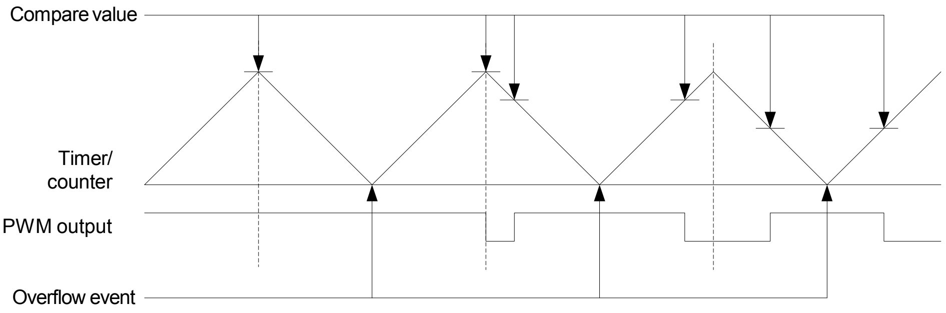
The inter-commutation delays are generated using the commutation timer (Timer/counter1) in the function 'StartupDelay'. See included source code documentation for usage if needed for other startup methods.

2.3 PWM/ADC

The motor speed/torque is controlled by pulse width modulation (PWM). It is important to understand how the PWM works and how it interacts with the analog to digital converter (ADC) in order to make reliable measurements in the noisy environment created by the PWM.

The PWM is used in what is called phase correct mode. This mode uses a counter in a dual slope operation that makes the PWM output symmetrical within one PWM period. Furthermore, the compare value that determines the duty cycle of the PWM output is buffered, so it is not updated in the middle of a PWM cycle. Figure 2 shows the relationship between counter value, compare value and PWM output. Each PWM period is separated by dashed lines in the figure. The figure also shows that an overflow event occurs when the timer reaches zero. This event can be used to automatically trigger an ADC sample. Unless the duty cycle is very low, this is a point where the PWM output has been stable for a long time. This is used to make sure that the ADC sample of the floating phase voltage is made when the PWM switching noise is low.

Figure 2. PWM generation



2.4 Hardware

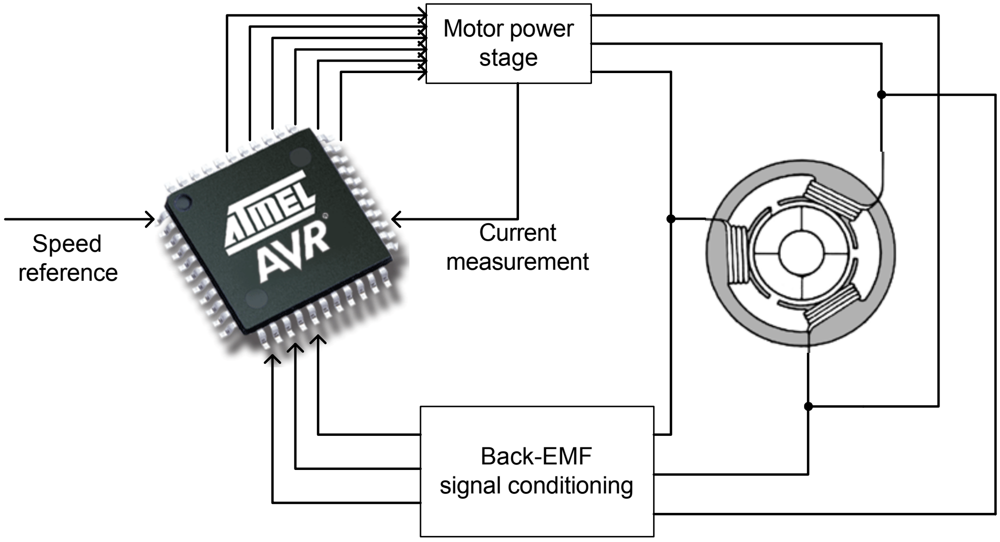
The suggested solution for sensorless control of 3-phase motors consists of six parts:

- AVR ATmega48/88/168 microcontroller
- 3-phase motor
- Motor power stage
- Back-EMF signal conditioning circuit

- Speed reference input
- Current measurement circuit

The interconnection of these parts is shown in Figure 3.

Figure 3. Sensorless control setup



The motor power stage is a standard triple half-bridge setup, often used for 3-phase motors. Figure 4 shows the conceptual schematics for this circuit. U, V and W are the three phase windings of the motor. The signals labeled XX_PWM are the control signals from the microcontroller. The switches are normally implemented with transistors, or similar semiconductors that can act as switches. The implementation details of the power stage circuit will differ and is outside the scope of this document. The ATAVRMC100 motor control kit gives an example of one such power stage circuit suitable for a wide range of motors. Schematics are available from www.atmel.com.

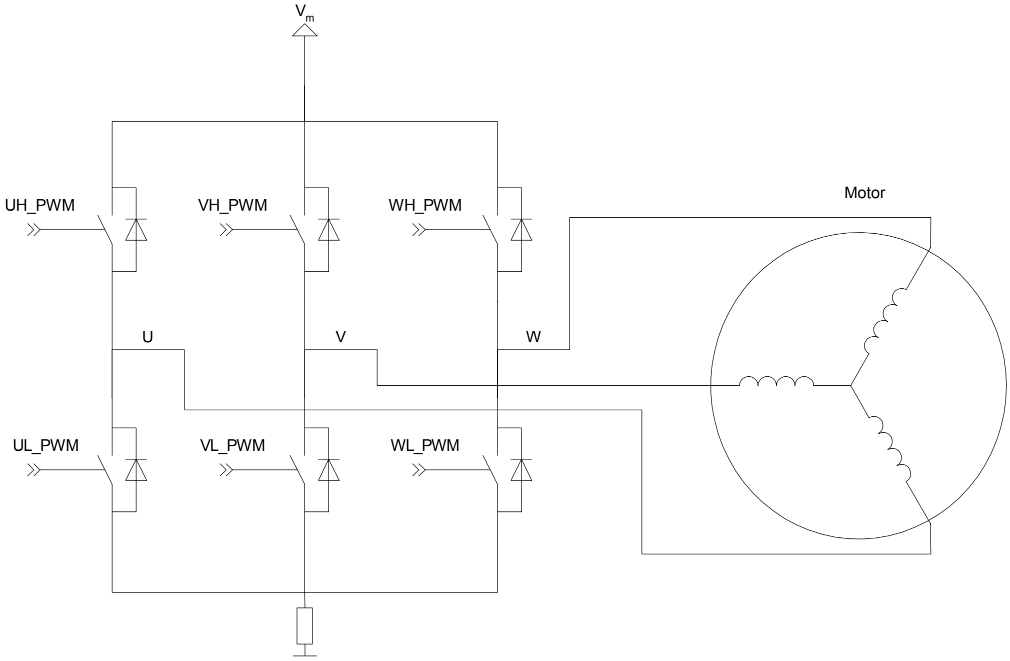
The switches connected to V_m are called high side switches. The switches connected to ground are similarly called low side switches. When operating a BLDC motor as shown in Figure 1, only one high side and one low side switch is open at any time, creating a closed circuit where the current runs into one phase winding and out of another phase winding, while the last phase winding is left floating.

The diodes across each switch, usually called flyback diodes, protect the switches from the high voltage spikes that can occur when switching highly inductive loads like a motor. Some transistors include this diode in the package.

The single shunt resistor between the power stage and ground is used to measure the current consumption of the motor. Large currents can pass through this resistor, so the resistance should be as small as possible, but large enough to give ADC readings with sufficient accuracy. Also make sure that this resistor is rated for the power dissipated at maximum current.

Figure 4 shows how the power stage is connected to the motor windings and the control signals for the switches.

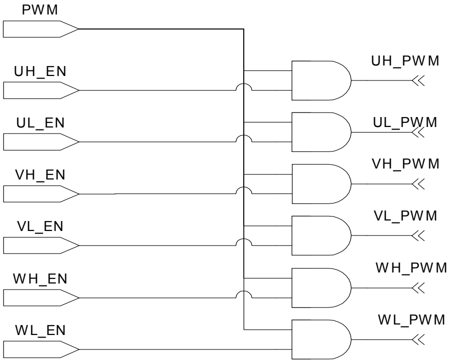
Figure 4. Power stage



Six synchronized PWM signals that can be individually turned on and off are needed to control the power stage of Figure 4. The ATmega48 microcontroller can generate six PWM signals, but that requires the use of all timer/counter modules. It is desirable to have one timer/counter unit available for commutation timing. The solution to this is to generate only one PWM signal, and route this signal to the active switches. This is easily achieved by using six external AND-gates. Each AND-gate controls one power stage switch. The PWM-signal is routed to one input of the AND-gate. Six general-purpose I/O pins from the microcontroller is used as enable-signal to each AND-gate. The PWM distribution logic schematics are shown in Figure 5.

In order to prevent floating driver signals while the microcontroller is in reset, it is recommended to place a pull-down resistor on the PWM line. This will effectively turn off all switches when the PWM line is floating.

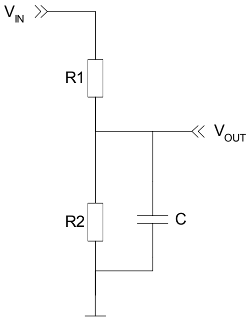
Figure 5. PWM distribution logic



2.4.1 Voltage divider/low pass filter circuit

One circuit is used several times throughout the design: the passive voltage divider and low pass filter (VD/LPF). The layout of this circuit is shown in Figure 6.

Figure 6. ADC voltage divider and low pass filter circuit



The component values in this circuit depend on the desired DC gain and low pass filter corner frequency. The following observations should assist the choice of components:

- At low frequencies, the circuit behaves as a normal voltage divider. $V_{OUT} = (R_2/(R_1+R_2))\ V_{IN}$. In other words, the DC gain equals that of the voltage divider made up by R_1 and R_2.
- For high frequencies, the capacitor C behaves as a short to ground, filtering away these frequencies.
- The corner frequency of the filter is given by the equation: $\omega_0=(R_1+R_2)/(R_1R_2C)$
- R1+R2 should be large (10-100 kΩ) to avoid large currents through the filter.

2.4.2 ADC reference voltage

The zero-crossing happens when the floating phase crosses the average voltage of the two supply rails. In this application note, it is assumed that the negative supply is at ground level, which makes the zero-cross voltage half the motor supply voltage. This dependence on motor supply voltage makes it impractical to use a fixed zero cross voltage threshold. Instead, the motor supply voltage (or scaled down version) is used as ADC reference voltage. The motor supply voltage needs to be low pass filtered before it is fed to the ADC. The VD/LPF of Figure 6 should be used for this purpose. The DC gain should be selected so that the voltage will be in the allowable range for the ADC, between 1.0V and AV_{CC}.

Note that the DC gain of this circuit will be affected by the relatively low-impedant input on the AREF pin. This must either be compensated for in choice of component values or in software.

2.4.3 Back-EMF signal conditioning

The three phase voltages should be connected to the ADC through 3 VD/LPFs. The filters should have the same DC gain as the ADC reference in order to utilize the full ADC voltage range. The low pass filter should be designed to filter out as much high frequency noise as possible without introducing notable delay to the back-EMF signal.

2.4.4 Current measurement

The current through the shunt resistor is full of high frequency components because of PWM switching and commutations. In order to make meaningful ADC samples of the current consumption, it must be low pass filtered. The voltages over the shunt will typically be so low, that a voltage division is unnecessary. If not, a smaller shunt resistor should be considered to avoid unnecessary power dissipation. A passive low pass filter (series resistor and capacitor to ground) should be sufficient for filtering.

2.4.5 Fixed voltage reference

Since the ADC reference voltage (AREF pin) is varying with motor supply voltage, a fixed, known voltage reference is needed in order to make accurate current measurements. The ATmega48 has an internal band gap voltage reference, which can be measured relative to AREF by the ADC, but stabilizes too slow for this purpose. Instead an external reference is used in this application note. Any source can be used for this voltage reference, as long as it is stable and the voltage level is known at compile time. In this application note, the V_{CC} voltage is fed through a VD/LPF to produce the known reference. The DC gain of the VD/LPF should in this case be selected such that the scaled fixed reference is never higher than the AREF voltage. The capacitor should be large enough, to remove any ripple on V_{CC}.

2.4.6 Speed reference

An analog signal has been used as speed reference in this application note. The speed reference could be any signal, e.g. a temperature sensor reading. In this application note, a simple potentiometer circuit has been used. A VD/LPF like Figure 6, using a potentiometer as R_2 resistor, with motor supply voltage as input and same voltage division as AREF ensures that the full ADC range is used.

2.4.7 Overcurrent detection/protection

The ADC current measurements are made once each PWM cycle, or approximately every 50µs at 20kHz PWM base frequency. If faster reaction to a critical overcurrent situation is needed, the analog comparator can be used. The filtered shunt voltage is connected to one analog comparator input. A fixed voltage representing the shunt voltage at maximum current consumption is connected to the other input. It is thus possible to get an interrupt that cuts power if the current is critically high. Note that any running interrupt must finish before the analog comparator interrupt is run, so there can be a small latency before power is cut. An external circuit that disables the PWM signal can be used if it is critical that power to the motor is shut down very quickly.

2.4.8 ATmega48 connections

The connections between the AVR and the different hardware subsystems are listed in Table 1. All signals on PORTB can be interchanged with each other. The same goes for the ADC channels.

Table 1. ATmega48 connections

ATmega48 pin	Connected to	Direction
AREF	Motor supply reference	In
PB0	WH_EN	Out
PB1	WL_EN	Out
PB2	VH_EN	Out
PB3	VL_EN	Out
PB4	UH_EN	Out
PB5	UL_EN	Out
PC0	Phase U ADC input	In
PC1	Phase V ADC input	In
PC2	Phase W ADC input	In
PC3	Shunt voltage	In
PC4	Speed reference	In
PC5	V_{CC} reference	In
PD5	PWM signal	Out
PD6	Analog comparator input 0	In
PD7	Analog comparator input 1	In

2.4.9 Peripheral usage

The following on-board peripheral units are used:

Table 2. Peripheral usage

Peripheral unit	Usage
Timer/counter0	PWM generation ADC sample triggering
Timer/counter1	Commutation timing Zero-cross holdoff timing
Analog to digital converter	Zero-cross detection Current measurement Speed reference input
Analog comparator	Overcurrent detection/prevention

2.5 Software

The software implementation is divided into four main parts:

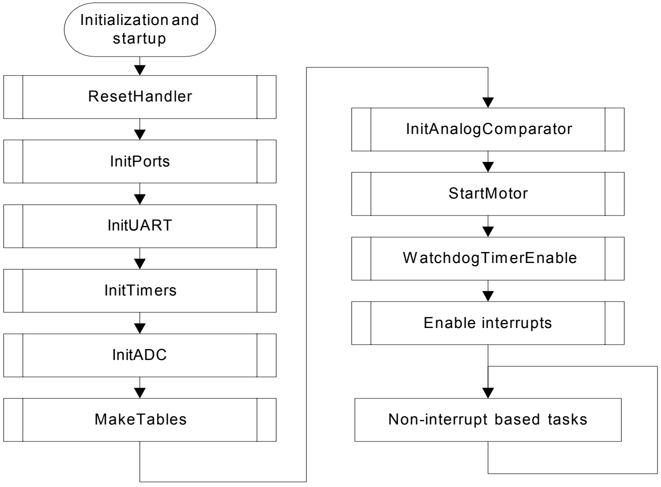
- Initialization and startup
- Sensorless commutation control

- Speed and current control
- Stall and overcurrent detection

2.5.1 Initialization and startup

The initialization and startup part initializes all peripherals, runs the motor startup sequence and hands over control of motor commutation to the fully interrupt-driven sensorless commutation control. At the same time, an eternal loop is entered that runs all non-interrupt based tasks. The only such task in the included example is the speed and current control. Figure 7 shows the flowchart for the Initialization and startup. The block labeled 'Enable interrupts' marks the point where sensorless commutation control is handed over to the interrupt-driven commutation controller.

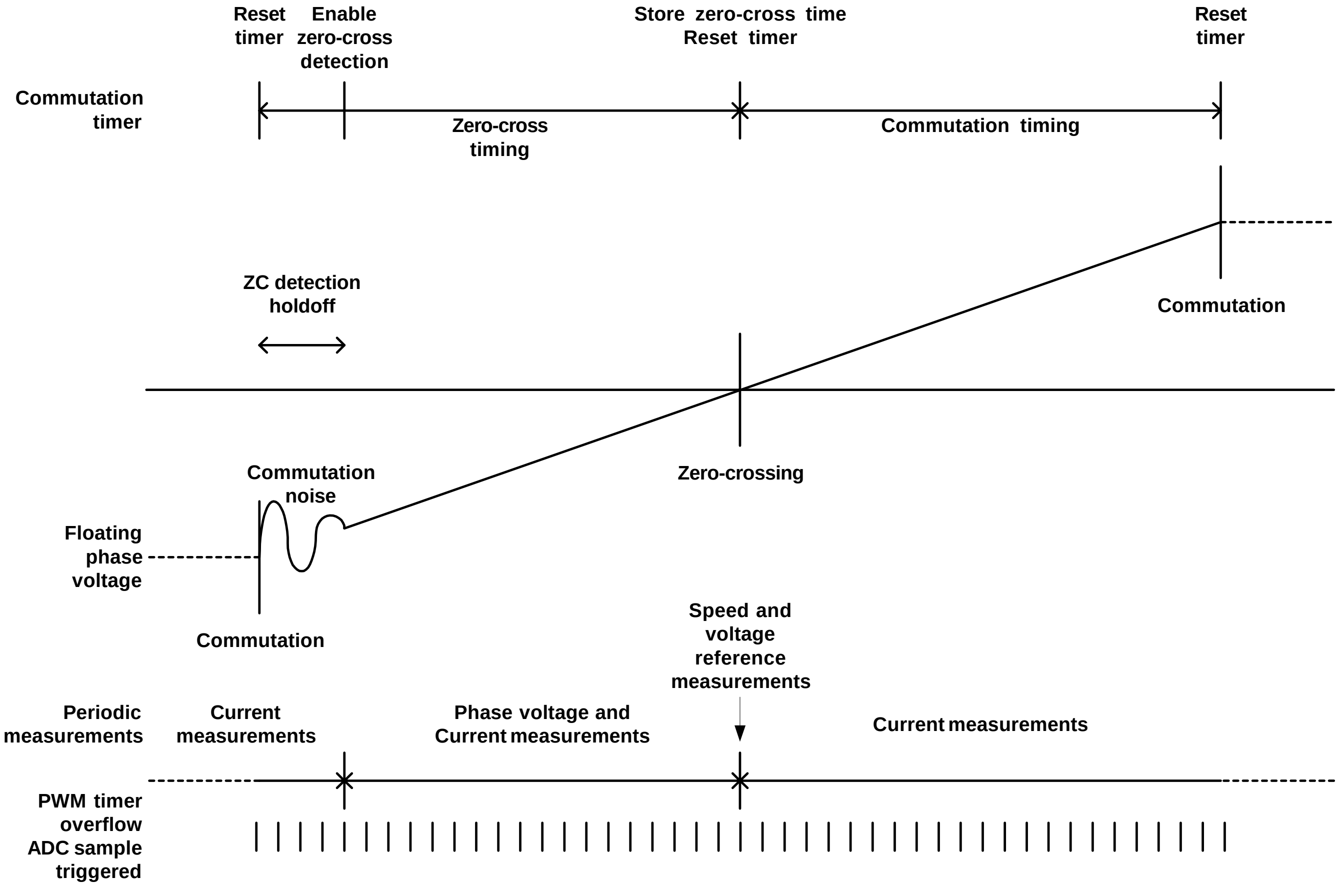
Figure 7. Initialization and startup



2.5.2 Sensorless commutation

In order to understand how the sensorless commutation is implemented, it is useful to take a look at the timing of the events that occur between two commutations. Figure 8 shows a close-up view of the voltage of a floating phase between two commutations.

Figure 8. Close-up view of the commutation timing



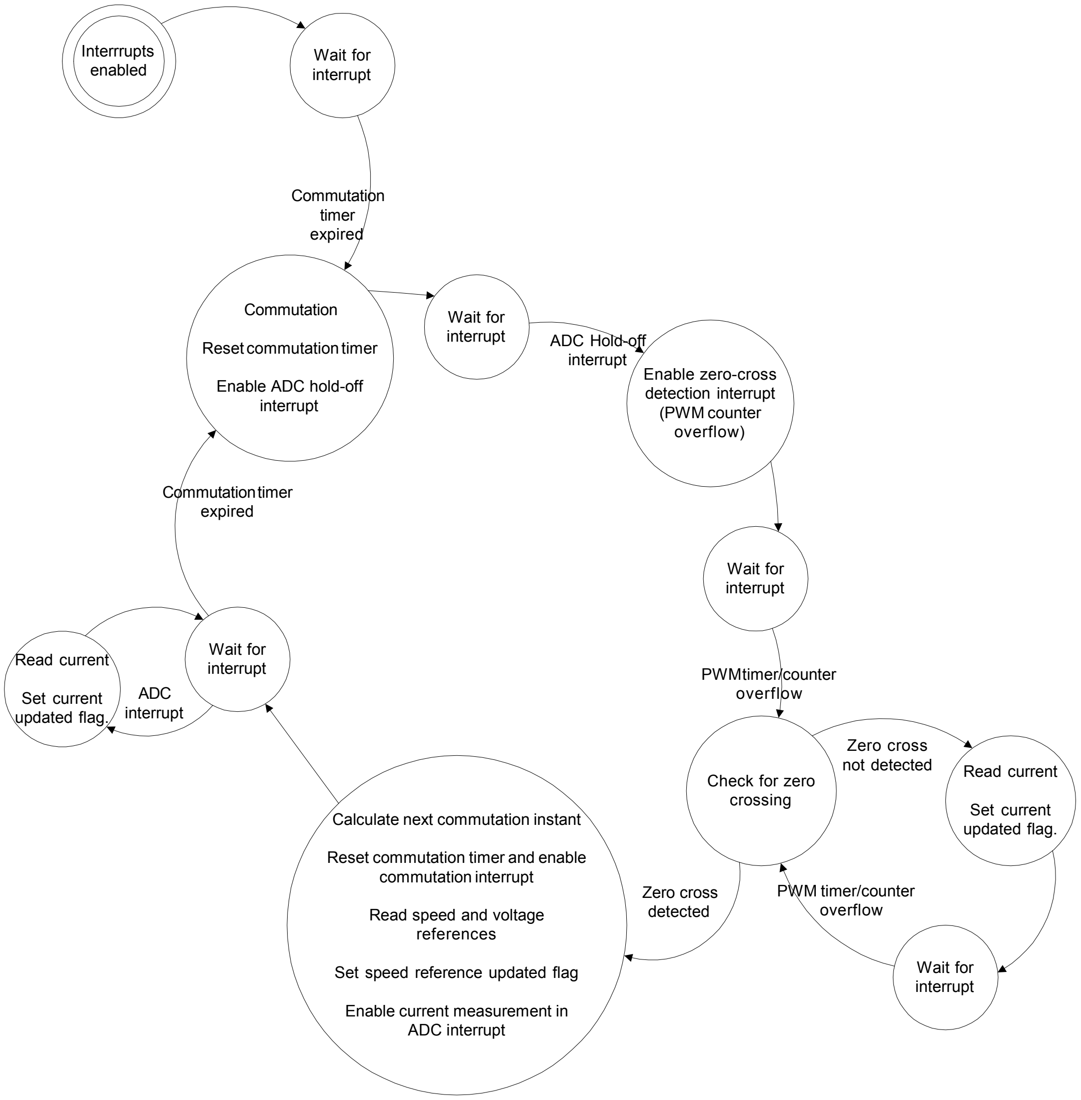
The sensorless commutation control is implemented using several interrupts. Interrupts are enabled and disabled during different stages of the commutation cycle. Table 3 shows the responsibility of each interrupt. The state diagram in **Figure** 9 shows how the interrupts cooperate to perform sensorless commutation.

The eternal loop in main is running every time the state machine is in one of the states labeled ‘Wait for interrupt’.

Table 3. Interrupt responsibility

Interrupt	**Responsibility**
Timer/counter0 (PWM timer) overflow	Zero-cross detection
Timer/counter1 (commutation timer) compare A	Commutation timing
Timer/counter1 (commutation timer) compare B	ZC detection hold-off timing
ADC complete	Current measurement
Analog comparator	Overcurrent detection/prevention

Figure 9. Sensorless commutation state machine


Interrrupts enabled
Wait for interrupt
Commutation timer expired
Commutation
Reset commutation timer
Enable ADC hold-off interrupt
Wait for interrupt
ADC Hold-off interrupt
Enable zero-cross detection interrupt (PWM counter overflow)
Wait for interrupt
PWMtimer/counter overflow
Check for zero crossing
Zero cross not detected
Read current
Set current updated flag.
Wait for interrupt
PWM timer/counter overflow
Zero cross detected
Calculate next commutation instant
Reset commutation timer and enable commutation interrupt
Read speed and voltage references
Set speed reference updated flag
Enable current measurement in ADC interrupt
Wait for interrupt
ADC interrupt
Read current
Set current updated flag.
Commutation timer expired

2.5.3 Commutation filter

Noisy back-EMF measurements and the fact that zero-cross detection is performed only once per PWM cycle can lead to zero-cross detections with a small deviation from middle of two commutations. Assuming constant or slowly varying speed, a digital filter with low-pass characteristics will improve commutation timing. In the included source code, an Infinite impulse response (IIR) filter has been used to smooth the effects of incorrect zero-cross detections. The filter takes the form of Equation 1.

Equation 1. Commutation IIR filter

$$y_n = \frac{ax_n + by_{n-1}}{a+b},$$

where y_n is is the filtered time at time step n, x_n is the measured time at time step n and a and b are weighing factors.

For computational efficiency and numerical stability, a+b should be a power of 2. In the included source code, a = 1, b = 3 has been used.

2.5.4 Speed and current control

With no sensors available, the speed must be calculated using information from the commutation controller. The commutation controller stores the time between commutation and zero crossing in a global variable that is also used to calculate the rotational speed. At the same time a flag is set that tells the speed controller that a new speed measurement is available. The external speed reference is also sampled right after a zero-cross is detected, so fresh measurements of these are available at the same time.

Since the speed information is only updated at every back-EMF zero crossing, the update rate is proportional to the motor speed. This can be a problem, since the parameters of a discrete-time controller are dependent on time step. E.g. to have a P-regulator with constant gain, the proportional gain parameter must be calculated from the current time step. The alternative is to use a fixed value and accept that the controller gain varies with speed.

The shunt voltage is measured once every PWM cycle, approximately every 50µs at 20kHz. In addition the external fixed voltage reference is measured once for every commutation step to compensate for varying motor supply voltage. The fixed voltage reference is used to calculate the motor-supply derived AREF voltage. Once the AREF voltage is known, the shunt voltage can be calculated and thus the current through the shunt.

2.5.5 Stall detection

The rotor is considered stalled if there has not been a commutation for a pre-determined period of time. The watchdog timer is used for this purpose. The watchdog timeout period is set during initialization and the watchdog timer is reset at every commutation. This triggers both the watchdog interrupt service routine and resets the microcontroller. The reset handler function that is run at startup determines the reset source and makes it possible to act differently when the microcontroller is first powered on and after a watchdog reset. This allows for powerful error detection and recovery.

2.5.6 Overcurrent detection/protection

The overcurrent detection/protection is based on the analog comparator. A shunt voltage that exceeds the overcurrent limit will trigger an analog comparator interrupt. This interrupt simply cuts power to the motor and waits in an infinite loop until the watchdog timer resets the microcontroller.

3 Implementation

A working implementation written in C is included with this application note. Full documentation of the source code and compilation information is found by opening the ‘readme.html’ file included with the source code.

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. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.