# AVR498: Sensorless control of BLDC Motors using ATtiny261/461/861

8-bit AVR® Microcontrollers

Application Note

## Features

- BLDC Motor Control in Sensorless Mode
- Timer 1 Waveform Generator Use
- Hardware Implementation
- External speed reference
- Code Example

## References


[1] ATtiny261/461/861 datasheet.

[2] AVR496: Brushless DC Motor Control using ATtiny861.

[3] AVR430: MC300 Hardware User Guide.

[4] AVR469: MC301 Hardware User Guide.

[5] AVR172: Sensorless Communtation of Brushless DC Motor (BLDC) using ATmega32M1 and ATAVRMC320


## 1 Introduction

This application note describes how to implement a brushless DC motor control in sensorless mode using the ATtiny861 AVR® microcontroller.

In this document, we will give a short description of brushless DC motor theory of operations, we will detail how to control a brushless DC motor in sensorless mode and we will also give a short description of the ATAVRMC301 and ATAVRMC300 boards used in this application note.

Sensorless commutation saves the cost of position sensors, wiring, and connectors compared to BLDC motors driven in sensor mode using Hall sensors. Without Hall sensors, the assembly of the motor is simplified. This reduces the motor and system costs.

The high performance AVR core a long with the Timer 1 of the ATtiny861 allows designing high speed brushless DC motor applications.

The ATtiny261/461/861 devices are all pin and source code compatible. The only difference is memory sizes. This application note is written with ATtiny861 in mind, but any reference to ATtiny861 in this document also applies to ATtiny261/461.

Rev. 8192A-AVR-04/09

# 2 Theory of operation

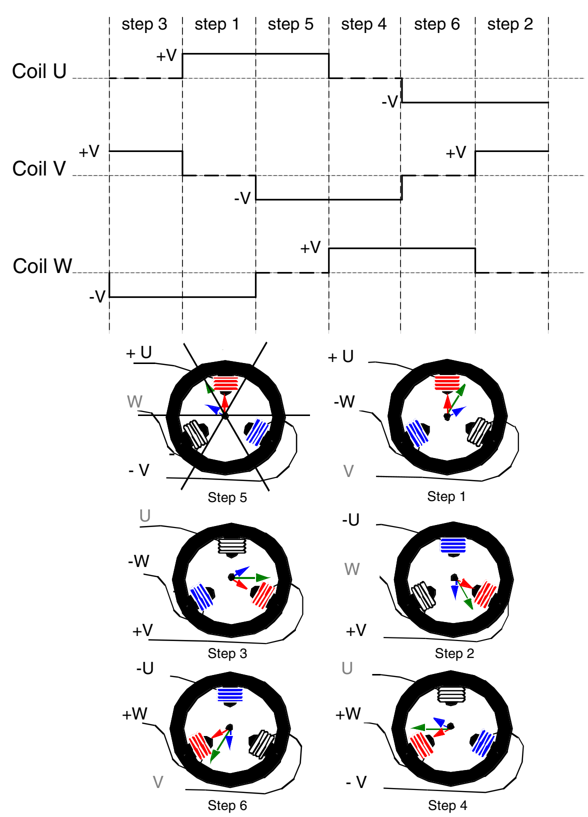
This document does not explain the theory behind brushless DC motors in detail. Instead, it focuses on the aspects that are important to sensorless control. For an introduction to three-phase brushless DC motors control in Sensorless Mode, please consult application note [5].

## 2.1 Block Commutation

The polarities of two coil currents with one coil left unconnected define six different positions for the rotor. Switching the currents in a way that the currents pull the rotor from the current position to the next position makes the rotor turning. Each position of the rotor is associated with a configuration of coil currents by a switching sequence. The coils are powered through fast switches (power MOSFETs...) which are controlled by a PWM block to provide the effective voltage value.

The block commutation scheme is outlined by Figure 1. For each commutation step there is one terminal connected to ground (ground symbol), one terminal is connected to the power supply (circle), and one terminal is left open (terminal name U, V, W). Permanent connection to ground and power supply drives the maximum current through the coils of the motor and will turn it with the maximum speed that is possible for a given motor load with a given supply voltage.

**Figure 1.** Commutation Sequence




Note: The step number are defined according to the value delivered by the Hall sensor outputs

For the block commutation, each sector of the rotor is mapped to the successive sector concerning current switching. So, commutation via interrupts becomes simple if each change of a Hall sensor signal forces an interrupt. Then, the actual set of Hall sensor signals defines the commutation sector. In other words, the block commutation can be described as a periodic sequence of 0Z11Z0 where 0 is connection to ground, Z represents an open terminal, and 1 is connection to the supply voltage source. This sequence is delayed by two steps for each successive terminal. The sequence 123456 of commutation steps is for U = Z00Z11, V = 11Z00Z, W = 0Z11Z0. For revolution into the opposite direction 654321 it is U = 11Z00Z, V = Z00Z11, W = 0Z11Z0.

The coils of the motor can be connected in star (Y connection) or triangle (Delta connection). Whatever the connection, the idea is to get an access to the null point to be able to measure the BEMF. Some motors allow this access via an additional wire. Direct access to the null point N enables direct measurements of the BEMF. The voltage of the null point N depends on the supply voltage and on the switching PWM scheme. If needed, the voltage of the null point N over ground can be electrically reconstructed by hardware or, for different switching PWM schemes, it is possible to calculate it.

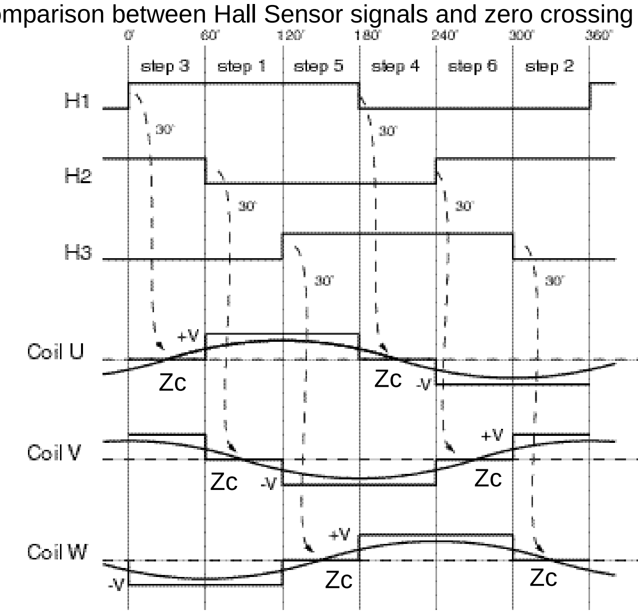
## 2.2 BEMF vs Hall Sensors

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 sinusoidal back-EMF.

The typical sinusoidal back-EMF waveforms and corresponding driving voltages of a 3-phase BLDC are shown in Figure 2. 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 2.

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 2.** Comparison between Hall Sensor signals and zero crossing of BEMF




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.

## 2.3 Ramp Up

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.

The method adopted in this application is described on the application note [5]. 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 'delay_us'. See included source code documentation for usage if needed for other startup methods.

# 3 ATtinyx61 microcontroller

## 3.1 Generating PWM signals with dead-time with ATtiny261/461/861

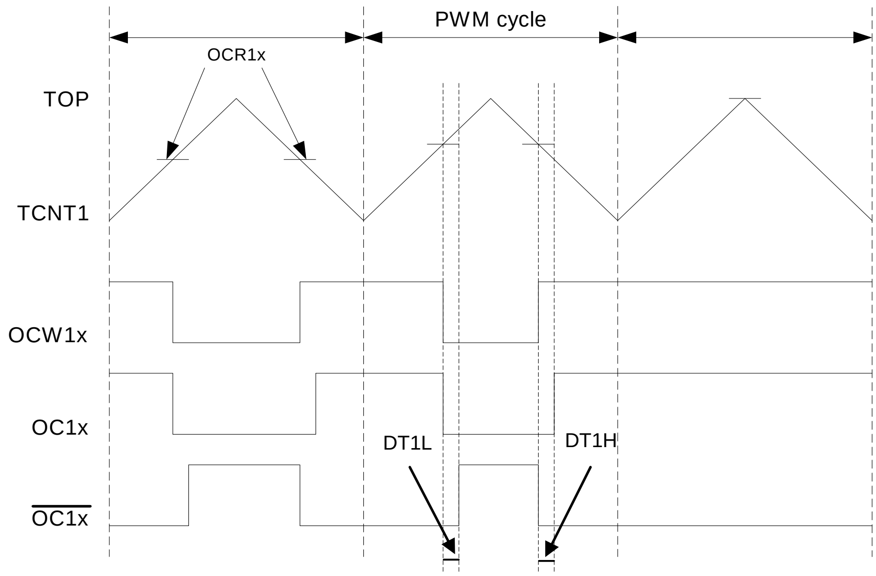
The Timer/Counter1 (TC1) module of the ATtinyX61 family is very well suited for driving three-phase motors. It can be run from a 64MHz PLL, and has a resolution of 10 bits. Six PWM outputs can be generated from this timer, grouped into three complementary output pairs with configurable dead-time. The “PWM6” mode is perfect for block commutation of brushless DC motors. For a maximum safety, a hardware fault protection unit can disable the PWM drivers without any intervention from the CPU.

To control a triple half-bridge driver stage, the “Phase and frequency correct PWM mode” of TC1 is used. In this mode, three pairs of complementary PWM outputs with hardware dead-time insertion can be generated. This is exactly what is needed to generate three-phase sinusoidal drive waveforms with a triple half-bridge driver stage and chosen in the PWM generation.

The operation of one of the three complementary output pairs can be seen in Figure 3. The counter counts in an up-down counting mode. The OCR1x register specifies the current duty cycle. An intermediate waveform, labeled as OCW1x in the figure, is generated by clearing the output on a compare match when up-counting and setting the output on compare match when down-counting. This intermediate waveform feeds the dead time generator, which in turn generates two outputs from this waveform. The non-inverted output, OC1x, follows OCW1x, except it will not go high until a specified dead-time period, DT1H, has elapsed after OCW1x goes high. The inverted output, OC1x, follows the inverted OCW1x signal, except it will not go high until a specified dead-time period, DT1L, has elapsed after OCW1x goes low. The result is a pair of complementary outputs with dead-time.

To control the duty cycle of one half-bridge, only one register, OCR1x needs to be changed. The average voltage output of the half-bridge will be proportional to its value.

**Figure 3.** Generating PWM signals

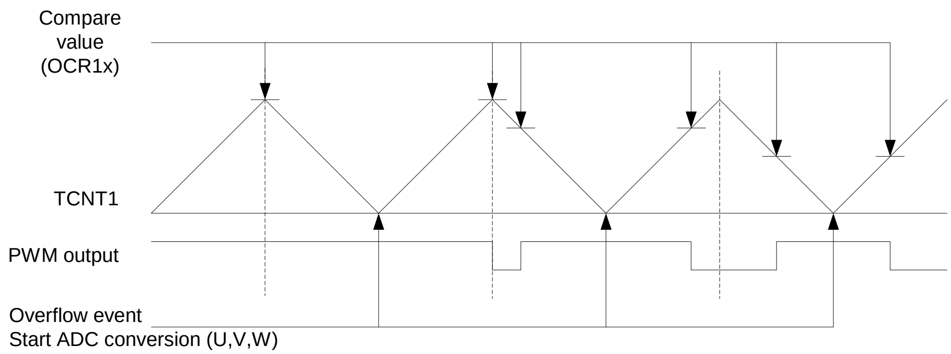



## 3.2 ADC Triggering

The Analog to Digital Converter (ADC) module of the ATtinyX61 has a resolution of 10 bits and a 0-Vcc input voltage range. The three phases U, V and W are measured through three channels on the ADC. The voltage reference input can be selected through the REFSx fields in the ADMUX register. This block is perfectly adapted for motor control as the triggering source can be selected through the ADTSx fields in the ADCSRB register: it is possible to synchronize ADC measure with PWM signals.

Indeed, Figure 4 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 occurs when the PWM switching noise is low.

**Figure 4.** ADC trigger source on TC overflow




## 3.3 PWM Base frequency

The PWM base frequency is controlled by the Timer/counter resolution and clock frequency.

The Timer/Counter1 resolution can be controlled by setting the TOP value in OCR1C. When operated in phase and frequency correct PWM mode, the counter counts up to the TOP value and down to zero during each PWM cycle. In this application note, the full 10 bit resolution of Timer/Counter1 is utilized, giving a PWM cycle period of 2046 timer clock cycles.

Timer/Counter1 is clocked from the system clock or the fast peripheral clock running at a nominal speed of 64 or 32MHz. In this application note, a clock frequency of 64MHz has been used to ensure a PWM frequency well above the audible frequency spectrum. The frequency of the Timer/counter1 clock will be referred to as fCLK,T/C1 in the following.

The PWM frequency as a function of Timer/Counter1 clock frequency can be calculated from Figure 5.

**Figure 5.** PWM base frequency as a function of timer clock frequency

$$f_{PWM} = \frac{f_{CLK},TC1}{2.TOP - 2}$$

The PWM frequency with a Timer/Counter1 clock frequency of 64 MHz and 10 bit resolution is thus 31,28kHz.

# 4 Hardware Description

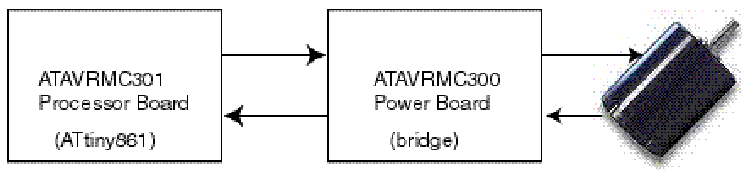
This application has been developed and tested with ATAVRMC300 and ATAVRMC301 boards.

The ATAVRMC300 board is the power board which embeds the bridge while the ATAVRMC301 is the processor board built around the ATtiny861 processor.

Refer to the 'AVR430: MC300 Hardware User Guide' and 'AVR469: MC301 Hardware User Guide' in depth descriptions of these two boards. The schematics are also available with these application notes.

Figure 6 shows a hardware assembly using the ATAVRMC301used with an ATAVRMC300 board.

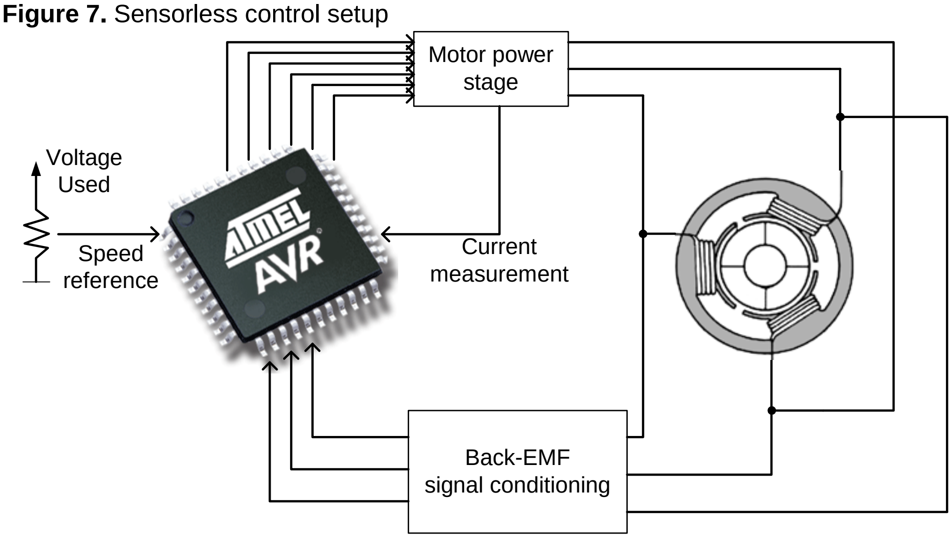
**Figure 6.** ATAVRMC301 & ATAVRMC300 connection




The power bridge (Q1 to Q6) is controlled through 6 signals UL, UH, VL, VH, WL, WH which are modulated by the PWM signal to adjust the motor DC voltage.

The speed reference can be adjusted with a potentiometer.

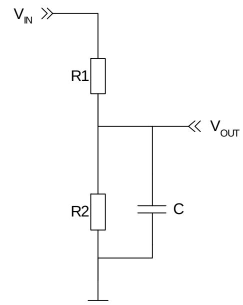
**Figure 7.** Sensorless control setup



## 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 8.

**Figure 8.** ADC voltage divider and low pass filter circuit




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

- At low frequencies, the circuit behaves as a normal voltage divider. $V_{OUT} = (R_2/(R_1+R_2)) \times V_{IN}$. 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 cutoff frequency of the filter is given by the equation:

$f_c$= 1/(2*PI*Req*C) : fc should be lower than PWM frequency in order to reject commutation noise

with

Req = $(R_1R_2)/(R_1+R_2)$

For example:

| |
|---|
| With a PWM frequency of **10Khz**: |
| R1 = 2.2K , R2 = 10K and C1 = 22nF, the **fc = 4.01Khz < PWM frequency** |

## 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, connected to AREF input. The motor supply voltage needs to be low pass filtered before it is fed to the ADC. The VD/LPF of Figure 8 should be used for this purpose.

By referring the AVR430 document, the $V_{motor'}$ is used as a reference of ADC. The maximum voltage reference should be on the range 0-2.56V.

In the current application, the power bridge used is powered by 12V.

So

R1 = 10k

R2 = 2.2k (To be mounted on schematic)

C = 22nF

$V_{OUT} = 0.180 \,.V_{IN}$ or $V_{m'} = 0.180 \,.V_m$

## 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.

For voltage measurement, the values are:

R1 = 33k

R2 = 5.6k

C = 22nF

$V_{OUT} = 0.145 .V_{IN}$

For example, in case the U signal is measured:

U’ = 0.145.U

## 4.4 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 8, 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.

## 4.5 Zero crossing threshold

The ADC has a 10-bit resolution. During zero crossing detection, the threshold is detected in the middle of the range of ADC resolution.

This value could be detected as:

Measured voltage: U’ = 0.145.U

Reference voltage: $V_{m'} = 0.180 .V_m$

Zero crossing threshold = $((U'/ V_{m'})*1024)/2 = 412$

## 4.6 ATtiny861 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 ADC channels.

**Table 1.** ATtiny861 connections

| ATtiny861 pin | Connected to | Direction |
|---|---|---|
| AREF | Motor supply reference (Vm’) | In |
| PB5 | WH | Out |
| PB4 | WL | Out |
| PB3 | VH | Out |
| PB2 | VL | Out |

| | | |
|---|---|---|
| PB1 | UH | Out |
| PB0 | UL | Out |
| PA1 | Phase U ADC input (U') | In |
| PA4 | Phase V ADC input (V') | In |
| PA5 | Phase W ADC input (W') | In |
| PA2 | Speed reference | In |

## 4.7 Peripheral usage

The following on-board peripheral units are used:

**Table 2.** Peripheral usage

| **Peripheral unit** | **Usage** |
|---|---|
| Timer/counter1 | PWM generation<br>ADC sample triggering |
| Timer/counter0 | Commutation timing |
| Analog to digital converter | Zero-cross detection<br>Speed reference input |

# 5 Software

## 5.1 Preamble

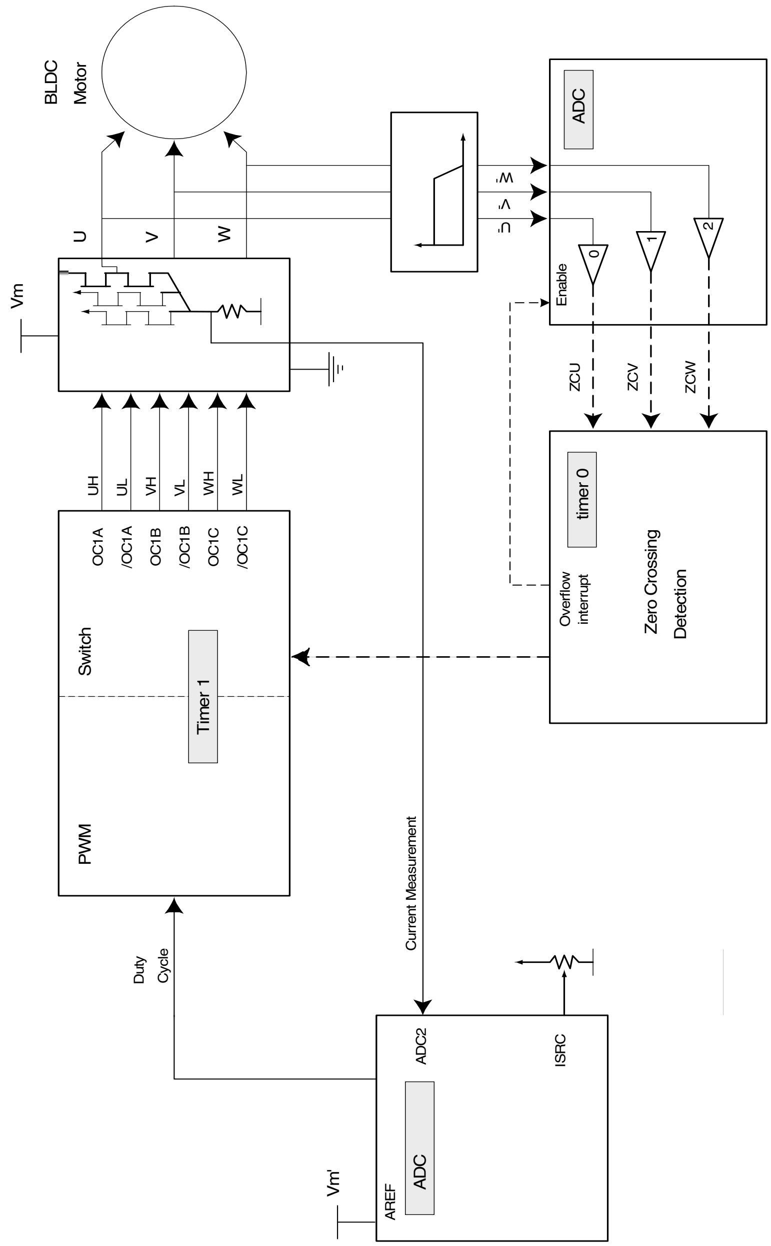
***Html documentation is delivered with the AVR498 software package. It can be opened thanks to the readme.html file located in the source directory.***

The software implementation is divided into three main parts:

- Initialization and startup
- Sensorless commutation control
- Speed control

## 5.2 BLDC Motor Control Implementation

**Figure 9.** Hardware Block Diagram




The Timer 1 is used to generate PWM outputs. The 'timer 1 compare match unit' and the 'dead time generators' provide the 6 PWM signals to drive the 6 power switches.

The 'Overflow interrupt on Timer 1' is used to detect zero crossing on phase U,V and W.

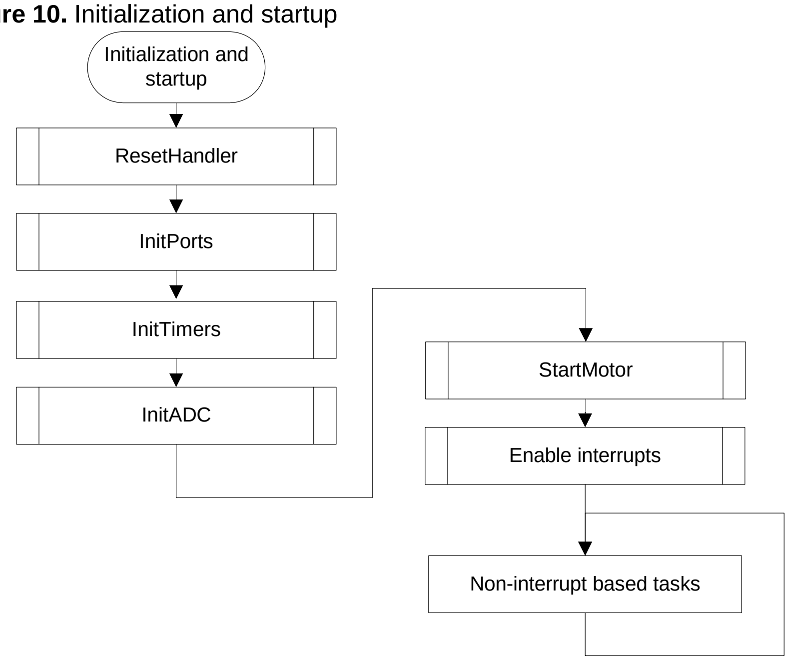
The ADC measures the potentiometer to set a duty cycle and measure the current which can be monitored by a regulation loop or sent through the TWI communication port.

The Timer 0 is used to determine the period of thirty degrees.

## 5.3 Initialization and startup

The initialization and startup part initializes all peripherals and runs the motor startup sequence in open loop. During the startup sequence, an alignement should be done between the mechanical position of the rotor and the electrical field during a time slot. When the ramp-up sequence is over, the application switches in sensorless commutation control. At the same time, an external loop is entered that runs all non-interrupt based tasks. The only such task in the included example is the speed control.  Figure 10 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.

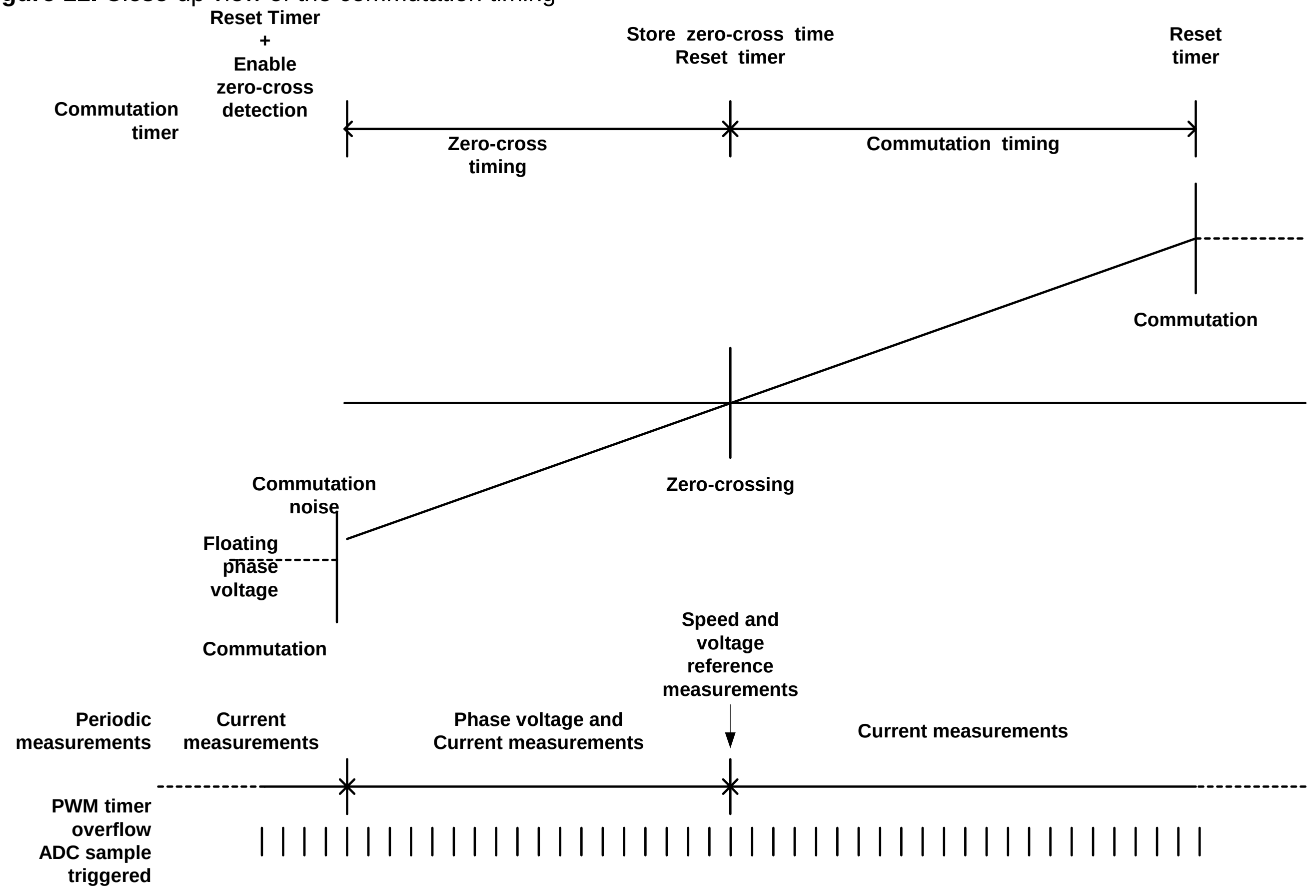
**Figure 10.** Initialization and startup




### 5.3.1 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 11 shows a close-up view of the voltage of a floating phase between two commutations.

**Figure 11.** Close-up view of the commutation timing



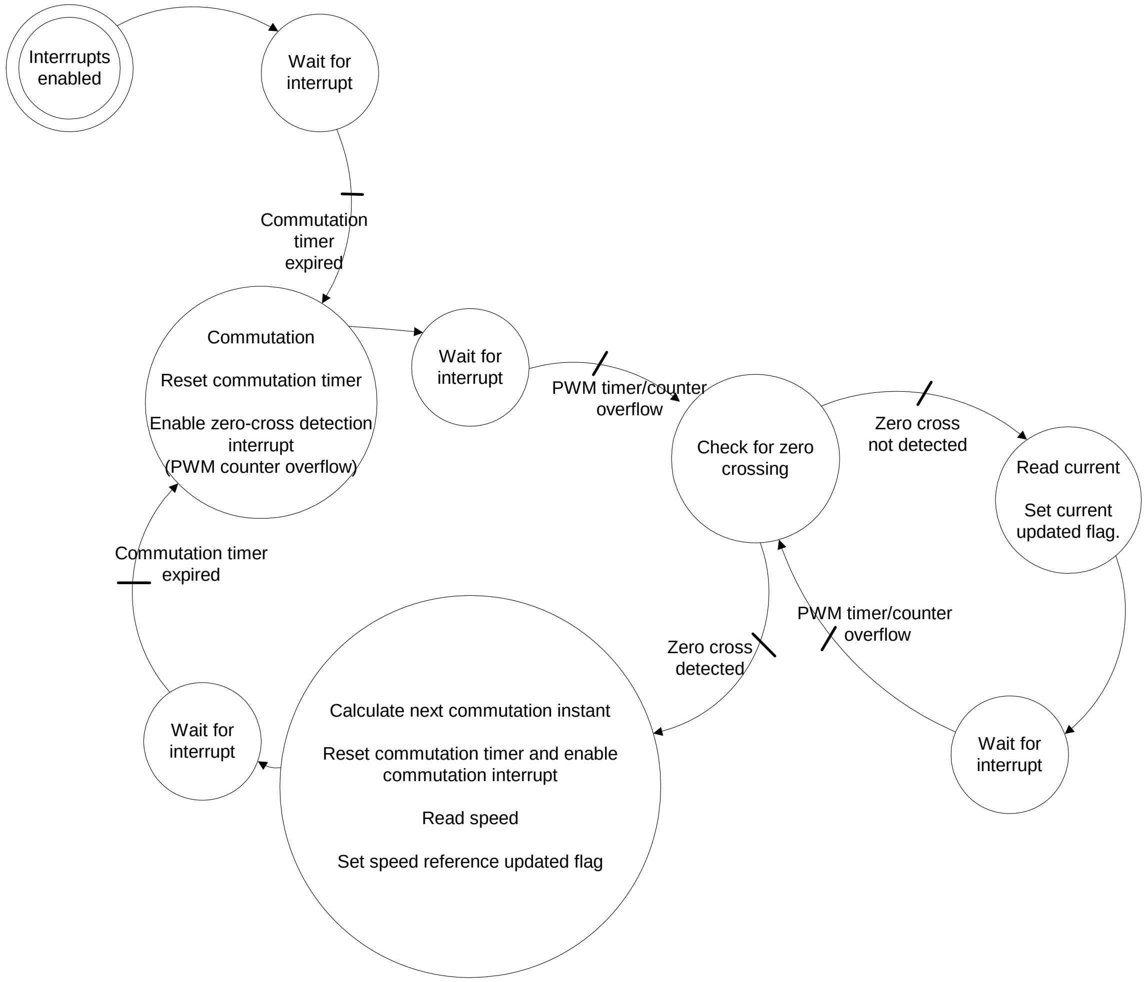

The sensorless commutation control is implemented using several interrupts. Interrupts are enabled and disabled during different stages of the commutation cycle. Table 3 shows the responsibility of each interrupt. The state diagram in Figure 12 shows how the interrupts work together to perform sensorless commutation.

The infinite 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 |

**Figure 12.** Sensorless commutation state machine




### 5.3.2 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. In addition, the external fixed voltage reference is measured once 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.

# 6 ATAVRMC300 & ATAVRMC301 Configuration and Use

The power board must be supplied with a 12V, 2A, DC Power Supply.

**Table 4.** ATAVRMC301 Jumper Settings

| Jumper | Position | Comment |
|---|---|---|
| J9 | UH,UL,VL | Closed |
| J10 | PA3,VM | Closed |
| J11 | | Opened |
| J12 | | Opened |
| J13 | | Opened |
| J15 | PA1, UCond | Closed |
| J16 | PA4, Vcond | Closed |
| J17 | PA5,Wcond | Closed |

# 7 Conclusion

This application note provides software and hardware solution for sensorless brushless DC motors applications. All the source code is available on the Atmel web site.

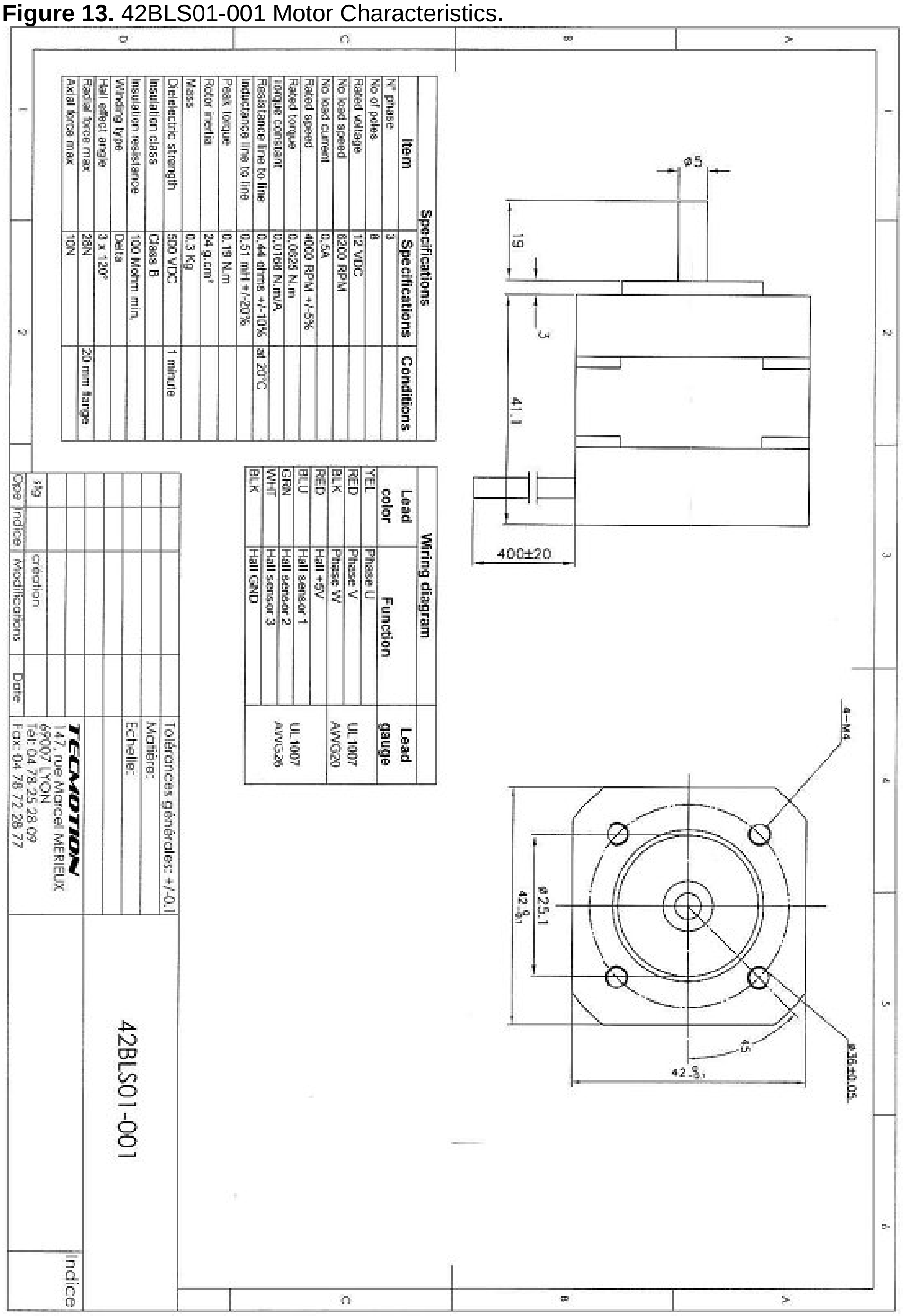
The software library provides functions to start and control the speed of any three phases brushless DC motors in sensorless mode.

The hardware is based on the minimal design with minimal external components required for control sensor brushless DC motors.

The ATtiny861 CPU and memory usage are low enough to allow for more complex applications.

# 8 Appendix

**Figure 13.** 42BLS01-001 Motor Characteristics.



## Headquarters


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


## International


***Atmel Asia***
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

***Atmel Europe***
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

***Atmel 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


## Product Contact

***Web Site***
http://www.atmel.com/

***Technical Support***
avr@atmel.com

***Sales Contact***
www.atmel.com/contacts

***Literature Request***
www.atmel.com/literature