# AVR194: Brushless DC Motor Control using ATmega32M1

8-bit AVR® Microcontroller

Application Note

## Features

- **BLDC Motor Basics**
- **Hardware Implementation**
- **Code Example**

## References

[1] ATmega32/64/M1/C1 data sheet

[2] AVR138: ATmega32M1 family PSC Cookbook

[3] AVR430: MC300 Hardware User Guide

[4] AVR470: MC310 Hardware User Guide

## 1. Introduction

This application note describes how to implement a brushless DC motor control in sensor mode using the ATmega32M1 AVR microcontroller.

The high performance AVR core fitted with Power Stage Controller module of ATmega32M1 allows to design high speed brushless DC motor applications.

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 sensor mode and we will also give a short description of the ATAVRMC310 and ATAVRMC300 boards used in this application note.

Software implementation is also discussed with software control loop using a PID filter.

This application note deals only with BLDC motor control application using Hall effect position sensors to control commutation sequence.

## 2. Theory of Operation

Brushless DC motors are used in a growing number of motor applications as they have many advantages:

They have no brushes so they require little or no maintenance.

They generate less acoustic and electrical noise than universal brushed DC motors.

They can be used in hazardous operation environments (with flammable products).

8138A–AVR–04/08

They also have a good weight/size to power ratio.

Such motors have little rotor inertia. The coils are attached to the stator. The commutation is controlled by electronics. Commutation times are provided either by position sensors or by coils Back Electromotive Force measurements.

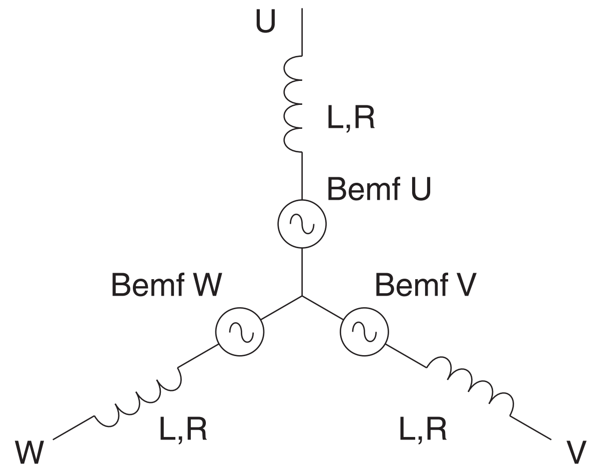
In sensor mode, Brushless DC motors usually consist of three main parts: a Stator, a Rotor and Hall Sensors.

## 2.1 Stator

A basic three phases BLDC motor Stator has three coils. In many motors the number of coils is replicated to have a smaller torque ripple.

Figure 1 shows the electrical schematic of the stator. It consists of three coils each including three elements in series, an inductance, a resistance and one back electromotive force.

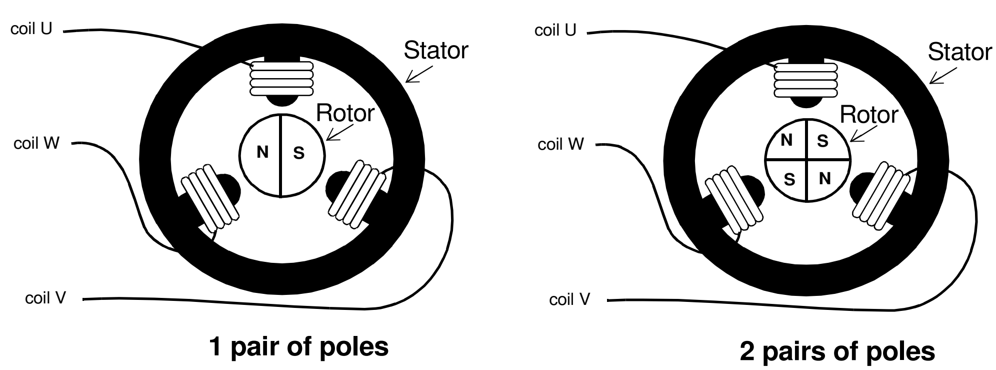
**Figure 1.** Stator Electrical Configuration (Three phases, three coils)




## 2.2 Rotor

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. The permanent magnets go from 1 to 5 pairs of poles. In certain cases it can go up to 8 pairs of poles.(Figure 2).

**Figure 2.** Three phase, three coil BLDC motor stator and rotor



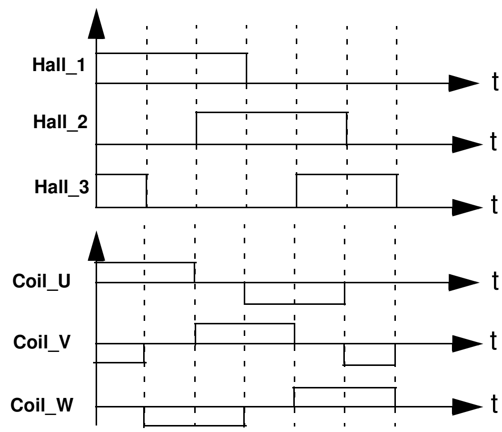
The coils are stationary while the magnet is rotating. The rotor of the BLDC motor is lighter than the rotor of a conventional universal DC motor where the coils are placed on the rotor.

## 2.3 Hall Sensor

For the estimation of the rotor position, the motor is equipped with three hall sensors. These hall sensors are placed every 120°. With these sensors, 6 different commutations are possible. Phase commutation depends on hall sensor values.

Power supply to the coils changes when hall sensor values change. With right synchronized commutations, the torque remains nearly constant and high.

**Figure 3.** Hall Sensors signals for CW rotation

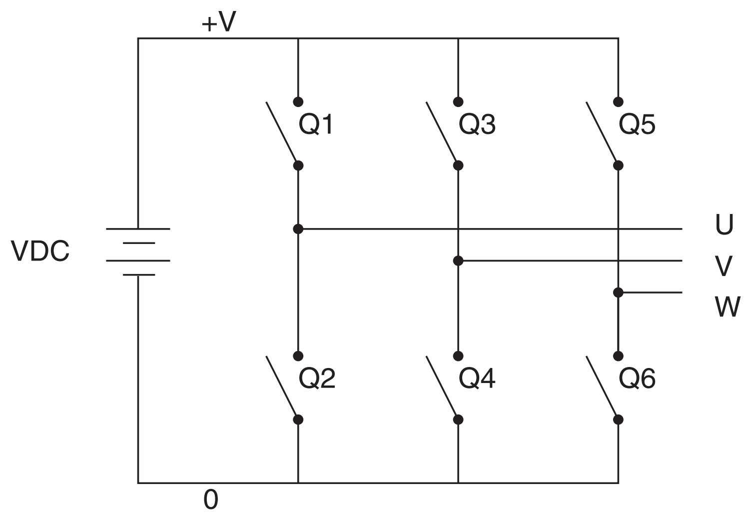



## 2.4 Phase Commutations

To simplify the explanation of how to operate a three phase BLDC motor, a typical BLDC motor with only three coils is considered. As previously shown, phases commutation depends on the hall sensor values. When motor coils are correctly supplied, a magnetic field is created and the rotor moves. The most elementary commutation driving method used for BLDC motors is an on-off scheme: a coil is either conducting or not conducting. Only two windings are supplied at the same time and the third winding is floating. Connecting the coils to the power and neutral bus induces the current flow. This is referred to as trapezoidal commutation or block commutation.

To command brushless DC motors, a power stage made of 3 half bridges is used. Figure 4 below shows a 3 half bridge schematic.

**Figure 4.** Power Stage




Reading hall sensor values indicates which switch should be closed.

**Table 1.** Switches commutation for CW rotation

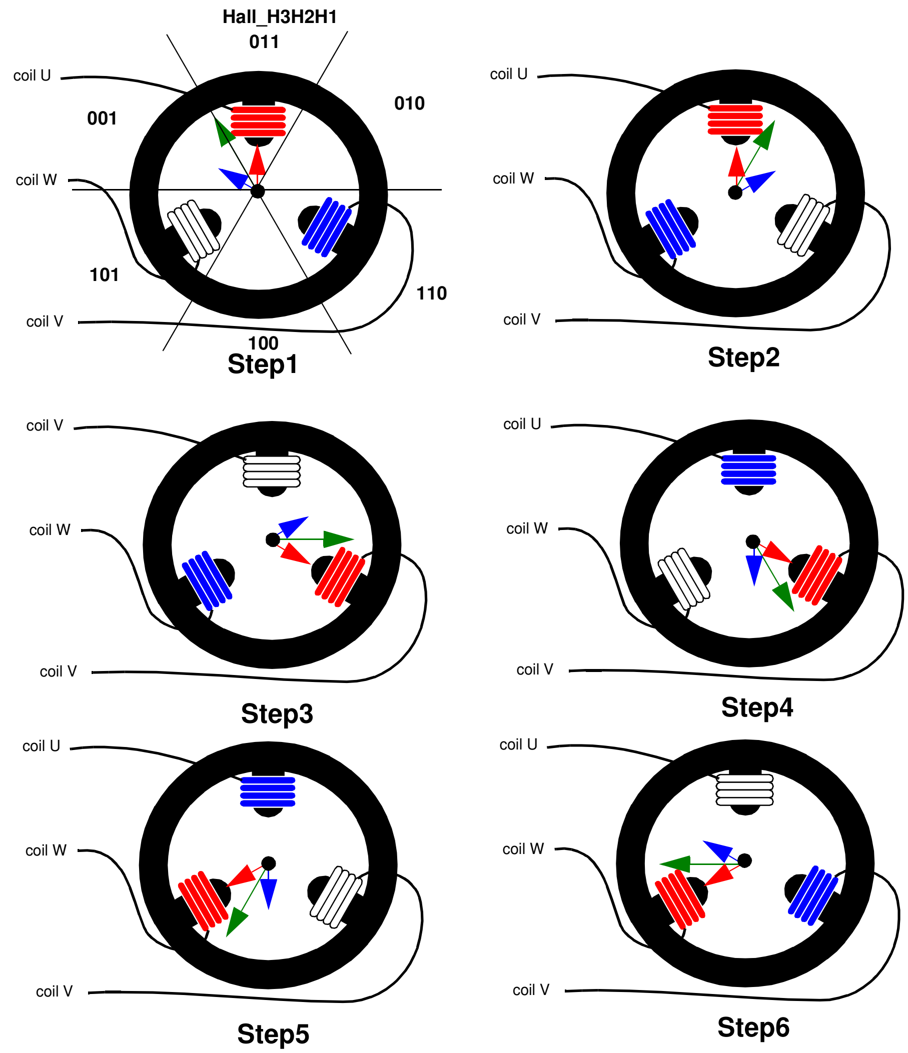
| Hall Sensors Value (H3 H2 H1) | Phase | Switches |
|---|---|---|
| 101 | U-V | Q1 ; Q4 |
| 001 | U-W | Q1 ; Q6 |
| 011 | V-W | Q3 ; Q6 |
| 010 | V-U | Q3 ; Q2 |
| 110 | W-U | Q5 ; Q2 |
| 100 | W-V | Q5 ; Q4 |

For motors with multiple poles the electrical rotation does not correspond to a mechanical rotation. A four pole BLDC motor uses four electrical rotation cycles to have one mechanical rotation.

The strength of the magnetic field determines the force and speed of the motor. By varying the current flow through the coils, the speed and torque of the motor can be adjusted. The most common way to control the current flow is to control the average current flow through the coils. PWM (Pulse Width Modulation) is used to adjust the average voltage and thereby the average current, inducing the speed. For example, the PWM frequency selected is the range from 10kHz to 200kHz according to the application (commutation losses, audible frequency...).

For a three phase, three coil BLDC motor, the rotating field is described in Figure 5.

**Figure 5.** Commutation steps and rotating field




Commutation creates a rotating field. Step1 Phase U is connected to the positive DC bus voltage by Q1 and Phase V is connected to ground by Q4, Phase W is unpowered. Two flux vectors are generated by phase U (red arrow) and phase V (blue arrow). The sum of the two vectors give the stator flux vector (green arrow). Then the rotor tries to follow the stator flux. As soon as the rotor reaches a given position, the hall sensors state changes its value from “010” to “011” a new voltage pattern is selected and applied to the BLDC motor. Then Phase V is unpowered and Phase W is connected to the ground, resulting in a new stator flux vector ‘Step2’.

By following the commutation schematic Figure 3 and Table 1, we obtain six differents stator flux vectors corresponding to the six commutation steps. The six steps provide one rotor revolution.

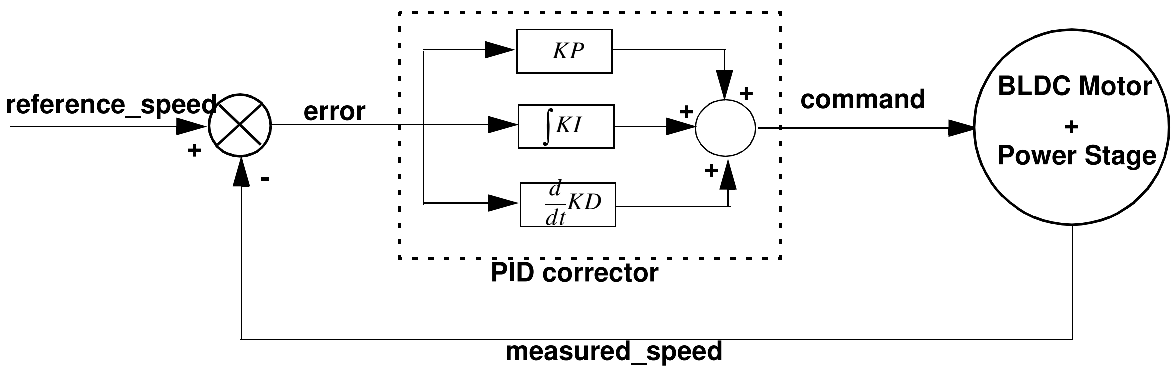
## 2.5 PID Regulation

After this brief theoretical presentation of Brushless DC Motor Control, the practical implementation will be introduced with the help of an example. The next part of this application note will deal with the hardware and the software implementation based on the starter kit ATAVRMC300 & ATAVRMC310 running with the Atmel ATmega32M1 microcontroller .

The software includes the control of the speed through a PID corrector. Such a corrector is composed of three main coefficients : KP, KI, and KD.

KP is the proportional gain coefficient, KI is the integral gain coefficient and KD is the derivative gain coefficient. The error between the desired speed and the real speed (called error in the Figure 2-1) is multiplied by each gain. Then, the sum of the three terms gives the command to apply to the motor to get the right speed (Figure 2-1).

**Figure 2-1.** PID diagram




$$command(t) = KP \times eror(t) + KI\int error(t)d(t) + KD\frac{d}{dt}error(t)$$

$$error(t) = reference\_speed(t) - measured\_speed(t)$$

The KP coefficient determines the motor response time, the KI coefficient is used to cancel the static error and the KD is used in particular for position regulation (refer to regulation loop in the software description for tuning of the coefficients).

# 3. ATmega32M1 microcontroller

The ATmega32M1 has been developed to provide an integrated solution for advanced motor control applications with CAN and LIN connectivity.

Based on the high performance AVR 8-bit RISC architecture, the ATmega32M1 integrates all of the basic peripherals necessary to satisfy the needs of complex algorithms. It integrates analog blocks like 10-bit ADC, with differential amplifiers and programmable gain options. Analog comparators with selectable comparison levels, and interrupts on pin change I/Os. The microcontrollers provide all necessary resources to control BLDC motors in their system environments.

The ATmega32M1 includes independent positive and negative comparator inputs to allow sensorless motor control with no external active component. Three individual comparators are available for back Electro Magnetic Field (EMF) measurements. An additional comparator is available for over-current detection. Its reference (comparison level) can be fixed via the DAC output or any external reference voltage. Clocked up to 64-MHz, the 12-bit versatile synchronous Power Stage Controller generates 6 complementary programmable high speed and precision signals to control the 3 half bridges of a motor. The maximum frequency is 64 kHz, with a resulting voltage resolution of about 1/1000. Hardware fault detection will automatically and immediately put the motor in a safe position in case a failure is detected.

## 3.1 ATmega32M1 Main Features

- **Data and Non-Volatile Program Memory**
  - **32K Bytes Flash of In-System Programmable Program Memory**
  - **1024 Bytes of In-System Programmable EEPROM**
- **2048 Bytes Internal SRAM**
- **Peripheral Features**
  - **One 12-bit High Speed PSC (Power Stage Controller)**
    - **Non Overlapping Inverted PWM Output Pins With Flexible Dead-Time**
    - **Variable PWM duty Cycle and Frequency**
    - **Synchronous Update of all PWM Registers**
    - **Auto Stop Function for Emergency Event**
  - **One 8-bit General purpose Timer/Counter with Separate Prescaler, Compare Mode and Capture Mode**
  - **One 16-bit General purpose Timer/Counter with Separate Prescaler, Compare Mode and Capture Mode**
  - **CAN 2.0A/B with 6 Message Objects**
  - **LIN 2.1 and 1.3 Controller or 8-Bit UART**
  - **One Master/Slave SPI Serial Interface**
  - **10-bit ADC**
    - **Up To 11 Single Ended Channels and 3 Fully Differential ADC Channel Pairs**
    - **Programmable Gain (5x, 10x, 20x, 40x) on Differential Channels**
    - **Internal Reference Voltage**
    - **Direct Power Supply Voltage Measurement**
  - **10-bit DAC for Variable Voltage Reference (Comparators, ADC)**
  - **Four Analog Comparators with Variable Threshold Detection**
  - **100µA ±2% Current Source (LIN Node Identification)**
  - **Interrupt and Wake-up on Pin Change**
  - **Programmable Watchdog Timer with Separate On-Chip Oscillator**
  - **On-chipTemperature Sensor**
- **Special Microcontroller Features**
  - **Low Power Idle, Noise Reduction, and Power Down Modes**
  - **Power On Reset and Programmable Brown Out Detection**
  - **In-System Programmable via SPI Port**
  - **High Precision Crystal Oscillator for CAN Operations (16 MHz)**
  - **Internal Calibrated RC Oscillator ( 8 MHz)**
  - **On-chip PLL for fast PWM ( 32 MHz, 64 MHz) and CPU (16 MHz)**

Note: Refer to the ATmega32M1 data sheet for the complete description of the ATmega32M1 microcontroller.

# 4. Hardware Description

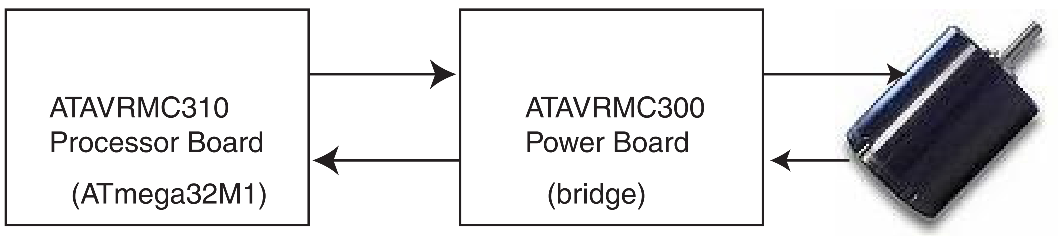
This application has been developed and tested with ATAVRMC300 and ATAVRMC310 boards.

The ATAVRMC300 board is the power board which embeds the bridge while the ATAVRMC310 is the processor board built arround the ATmega32M1 processor.

Refer to the 'AVR430: MC300 Hardware User Guide' and 'AVR470: MC310 Harware User Guide' in depth descriptions of these two boards. The schematics are also available with these application notes.

Figure 4-1.gives a block diagram of the ATAVRMC310 used with an ATAVRMC300 board.

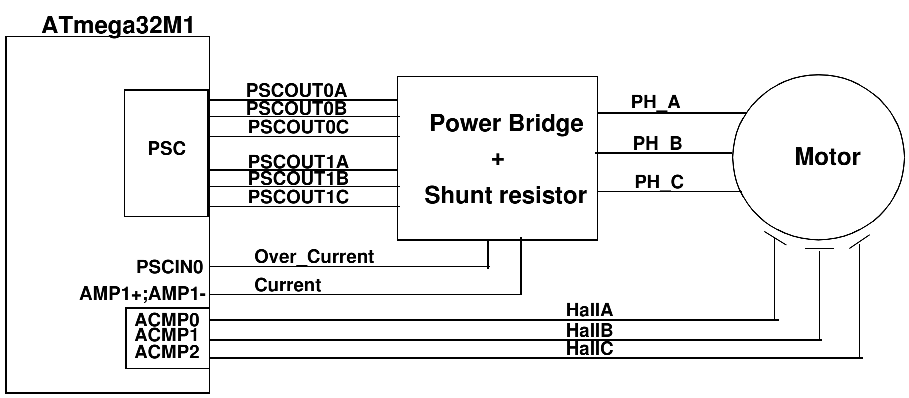
**Figure 4-1.** ATAVRMC310 & ATAVRMC300 connection




As shown in Figure 6 the microcontroller contains a Power Stage Controller (PSC). The PSC can be seen as a Pulse Width Modulator with six output signals. To avoid cross conduction a Dead Time control is integrated (see ATmega32M1 datasheet for more information about PSC or Figure 8 below).

A fault input (Over_Current) is linked to PSCIN. This fault input enables the microcontroller to disable all PSC outputs.

**Figure 6.** Hardware implementation




**PSC : Power Stage Controller**
**ACMPi : Analog Comparator Positive input (i = 0,1,2)**
**AMPi+/- : Analog Differential Amplified channel Positive/Negative inputs (i = 0,1,2)**

It's possible to measure the current with two differential amplified channels with programmable 5, 10, 20 and 40 gain stage. The Shunt resistor has to be adjusted to the amplifier conversion range.

The Over_Current signal comes from an internal comparator. The comparator's reference can be adjusted by the internal DAC.

The phase commutation has to be done according to hall sensors value. HallA, HallB and HallC are connected to external interrupt sources or to the three internal comparators. The comparators generate the same type of interrupts as the external interrupts. Figure 7 shows how the microcontroller I/O ports are used in the starter kit.

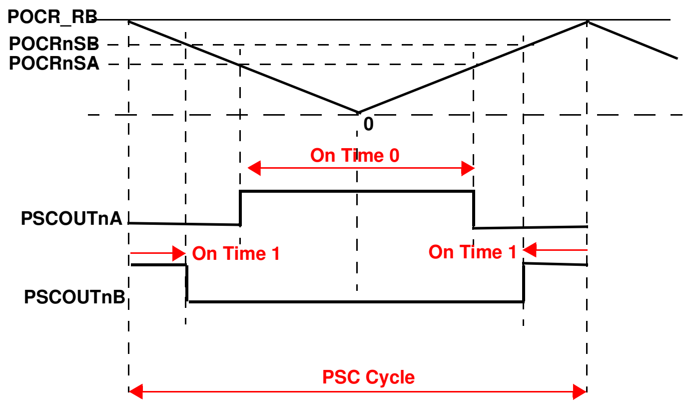
**Figure 7.** Microcontroller I/O Ports use (SO32 package)

| | | | |
|---|---|---|---|
| PB2/ADC5/INT1/ACMPN0 | **20** | | |
| PB3/AMP0- | **27** | | |
| PB4/AMP0+ | **28** | ***Vm'*** | |
| PB5/ADC6/INT2/ACMPN1/AMP2- | **30** | | |
| PB6/ADC7/PSCOUT1B | **31** | **VL** | **Q4** |
| PB7/ADC4/PSCOUT0B/SCK | **32** | **UL** | **Q2** |
| PORTC | | | |
| PC0/INT3/PSCOUT1A | **2** | **VH** | **Q3** |
| PC1/PSCIN1/OC1B | **7** | | ***Depends on the communication mode*** |
| PC2/T0/TXCAN | **10** | ***TxCAN*** | |
| PC3/T1/RXCAN | **11** | ***RxCAN*** | |
| PC4/ADC8/AMP1-/ACMPN3 | **21** | **GNDm** | **Current_Shunt-** |
| PC5/ADC9/ACMP3/AMP1+ | **22** | **ShCo** | **Current_Shunt+** |
| PC6/ADC10/ACMP1 | **26** | **H2** | **Hall Sensor 2** |
| PC7/D2A/AMP2+ | **29** | **DAC_OUT** | **DAC_OUT** |
| PORTD | | | |
| PD0/PSCOUT0A/XCK/SS_A | **1** | **UH** | **Q1** |
| PD1/PSCIN0/CLKO | **4** | **Fault** | *Not Used with MC300* |
| PD2/PSCIN2/OC1A/MISO_A | **5** | **MISO** | ***Depends on the communication mode*** |
| PD3/TXD/TXLIN/OC0A/SS/MOSI_A | **6** | **MOSI / LIN TxD / TxD** | |
| PD4/ADC1/RXD/RXLIN/ICP1A/SCK_A | **16** | **SCK / LIN RxD / RxD / POT** | |
| PD5/ADC2/ACMP2 | **17** | **H3** | **Hall Sensor 3** |
| PD6ADC3/ACMPM/INT0 | **18** | | |
| PD7/ACMP0 | **19** | **H1** | **Hall Sensor 1** |
| PORTE | | | |
| PE0/RESET/OCD | **3** | **RST** | ***Depend on the communication mode*** |
| PE1/OC0B/XTAL1 | **14** | ***Quartz CAN*** | |
| PE2/ADC0/XTAL2 | **15** | ***Quartz CAN*** | |

Vm' is implemented but not used. It can be used to get the value of the DC bus voltage.

The outputs UH, UL, VH, VL, WH & WL are used to control the power bridge. As previously seen, they depend on the Power Stage Control (PSC) which generates PWM signals. For such applications, the recommended mode is the Center Aligned Mode (see Figure 8), the register POCRnRA is used to adjust ADC synchronization for current measurement.

**Figure 8.** PSCOUTnA & PSCOUTnB Basic Waveforms in Center Aligned Mode




Please refer to the ATmega32M1 data sheet to get the value of all timings according to register values. The input clock of PSC comes from the internal PLL.

Two strategies can be used for PWM signals applied to the power stage. The first one (fast decay) is to apply the PWM signals on the high side AND the low side of the power bridge and the second one (slow decay) is to apply the PWM signals only on the high side of the power bridge.

# 5. Software Description

HTML documentation is delivered with the AVR194 software package. It can be opened thanks to the readme.html file located in the source directory.

Atmel provides libraries to control Brushless DC motors. The first step is to configure and initialize the microcontroller.

The function to be used is mc_init(). It calls hardware and software initialization functions and sets up all motor parameters (motor direction, motor speed, motor run or motor stop).

## 5.1 Interface Functions

After the microcontroller configuration and initialization, the motor can be started. Only a few functions are needed to control the motor. All user functions are defined in the mc_interface.h file:

**void** mci_run(**void**)

Used to start the motor. The regulation loop function is launched to set the duty cycle. Then the first phase commutation is executed.

**Bool** mci_motor_is_running(**void**)

Gets the command motor state. If ‘TRUE’ the motor is running. If ‘FALSE’ the motor is stopped.

**void** mci_stop(**void**)

Used to stop the motor.

**void** mci_set_ref_speed(**U8** speed)

Sets the reference speed.

**U8** mci_get_ref_speed(**void**)

Returns the reference speed.

**void** mci_forward()

Sets the motor direction ‘CW’ (clock wise).

**void** mci_backward()

Sets the motor direction ‘CCW’ (counter clock wise).

**U8** mci_get_motor_direction(**void**)

Returns the direction of rotation of the motor.

**U8** mci_set_measured_speed(**U8** measured_speed)

Saves the measured speed in the measured_speed variable.

**U8** mci_get_measured_speed(**void**)

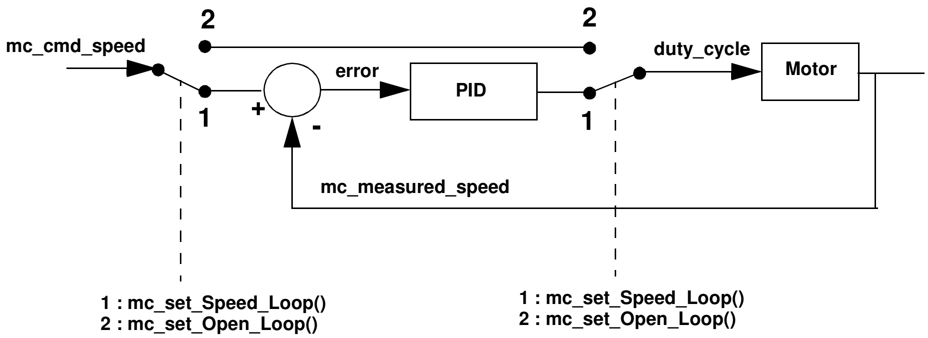
Gets the measured speed.

## 5.2 Regulation Loop

Two functions select the regulation loop. Open loop and speed loop. Figure 9 shows the regulation loop implemented in the software.The two functions are mc_set_speed_loop() and mc_set_open_loop().

In open loop mode, the reference speed is used as duty cycle for the PWM.

**Figure 9.** Regulation Loop



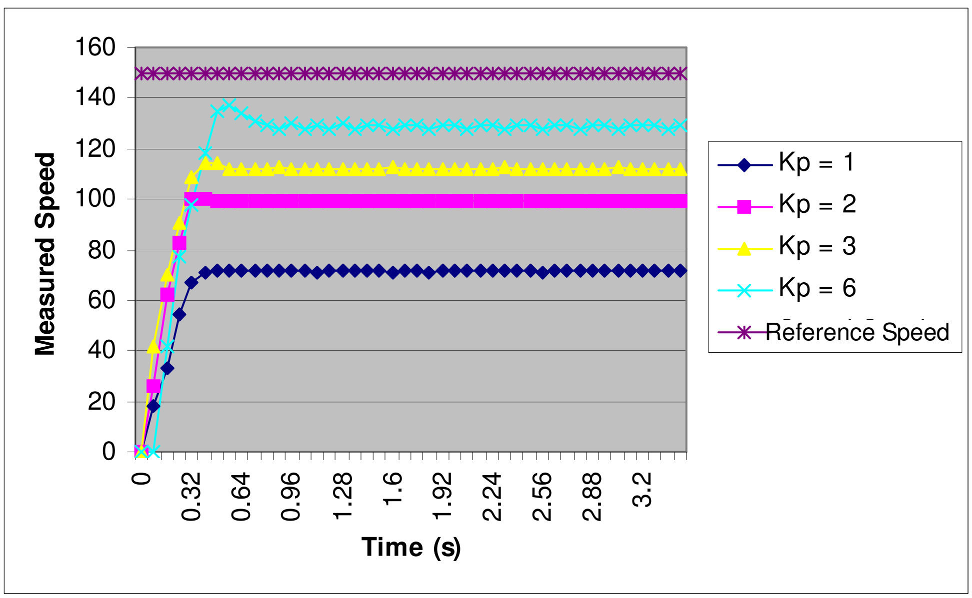

The speed loop consists in a speed regulation loop with the PID corrector.

We will further explain how to adjust the coefficients KP and KI. The KD coefficient is present in the regulation loop but not used.

As previously shown, the KP coefficient is used to regulate the motor response time. First time set KI and KD to '0'. Try different values of KP to get a correct motor response time.

- If the response time is too slow, increase KP gain.
- If the response time is quick, but motor speed is unstable, decrease KP gain.

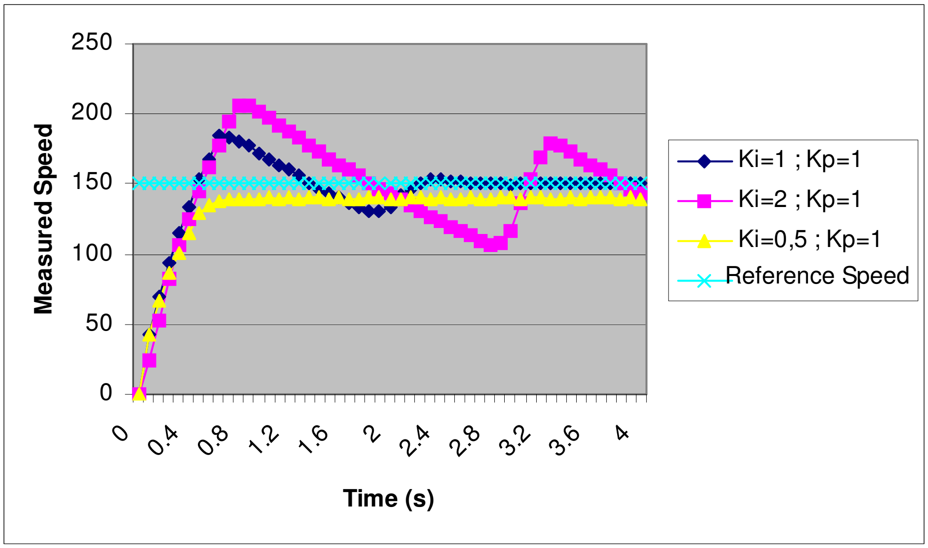
**Figure 10.** Kp Tuning Example




KI parameter is used to cancel the static error. Leave the KP coefficient unchanged and set the KI parameter.

- If the error is still different from zero increase KI gain.
- If the error is cancelled but oscillations appears, decrease KI gain.

**Figure 11.** Ki tuning example




In Figure 10 and Figure 11, the right parameters are KP = 1, KI = 0.5 and KD = 0.

To adjust the KD parameter:

- If the reponse is still slow, increase KD gain.
- If the reponse is still unstable, decrease KD gain.

Another significant point is the sampling time. It has to be chosen according to the response time of the system. The sampling time must be at least twice as small as the response time of the system (according to the Shannon-Nyquist criteria).

Two functions are available for the sampling time configuration (explained previously). They result in a global variable called g_tick which is set every 250us. With this variable it is possible to configure the sampling time.

## 5.3 CPU & Memory Usage

All measurements have been realized with Fosc = 16MHz. They also depend on the motor type (numbers of pair of poles). With a motor of 4 pairs of poles, hall sensor frequency is four times faster than the motor rotation.

All results in Figure 5-1 are obtained with a three phases BLDC motor with four pairs of poles and a maximum speed of 6900 rpm. (Motor provided with the ATAVRMC300 kit)

**Table 5-1.** Microcontroller utilization rate

| Function | Parameters | activation time | activation period | Ratio uc % |
|---|---|---|---|---|
| mc_estimation_speed() | Speed = 6900 rpm | 15 us | 2 ms | 0.75 |
| mc_switch_commutation() | Speed = 6900 rpm | 8 us | 300 us | 2.7 |
| mc_regulation_loop() | Open Loop | 1.4 us | 80 ms | 0.0175 |
| | Close Loop | 20 us | 80 ms | 0.025 |

In the worst case, the microcontroller utilization ratio is about 3.5% with a sampling time of 80ms at 6900 rpm.

All ratio measurement have been made with the same software. No communication modes are used (no UART, no LIN...).

In these conditions, the microcontroller memory usage is:

- 5500 bytes of CODE memory (Flash occupation = 17% ).
  Including communication protocol through UART
- 488 bytes of DATA memory (SRAM occupation = 24%).
  Including stack and communication protocol through UART

# 6. ATAVRMC300 & ATAVRMC310 Configuration and Use

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

**Table 6-1.** ATAVRMC300 jumper settings

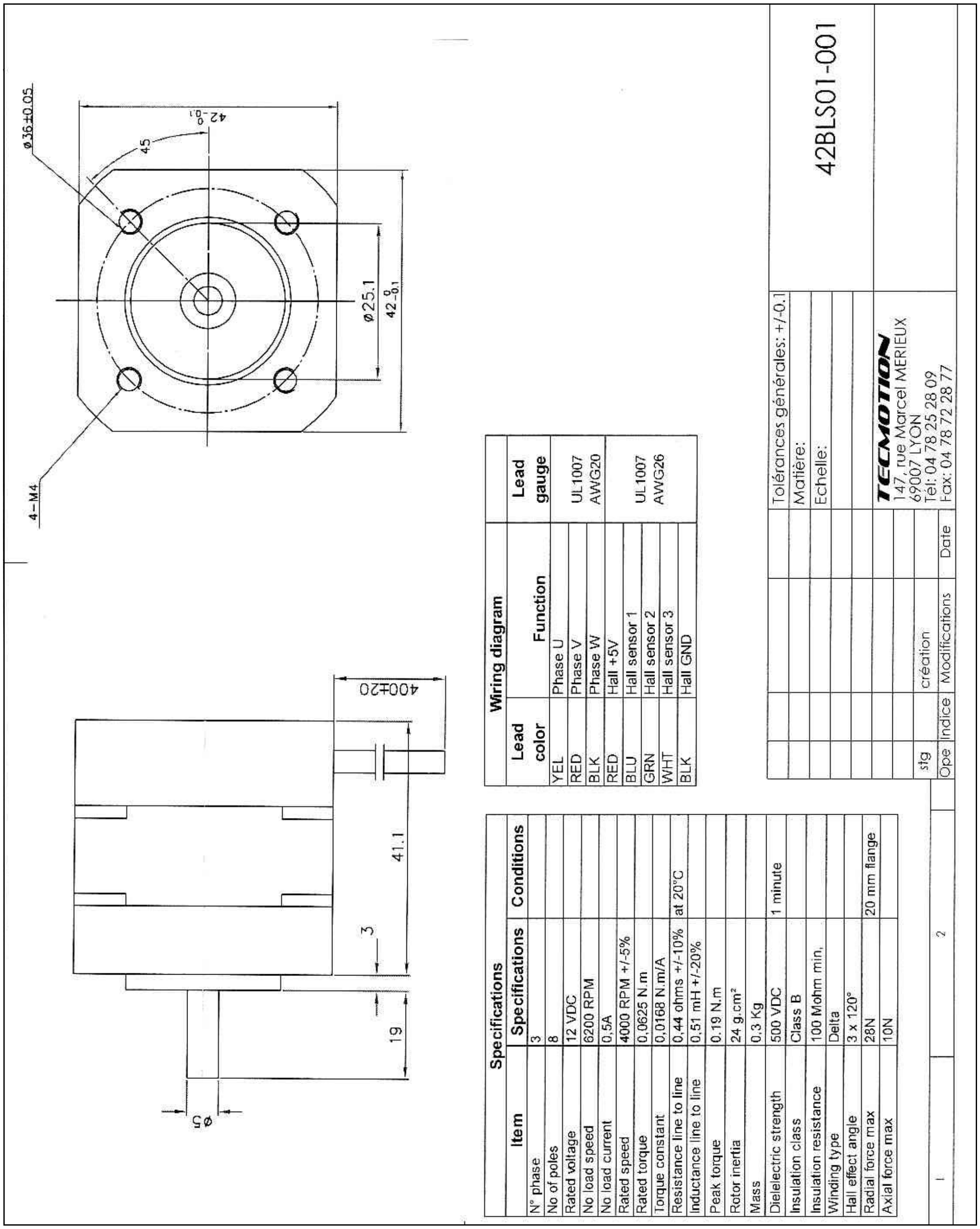
| Jumper | Position | Comment |
|---|---|---|
| J1(VHa) | Pin1 & 2 shorted | VHa = +5V |
| J2(VCC) | Open | Vcc = +5V |

**Table 6-2.** ATAVRMC310 jumper settings

| Jumper | Position | Comment |
|---|---|---|
| J5 | Vm’ | PB4 = Vm’ |
| J6 | Undefined | PB3 = x |
| J7 | RxCAN | PC3 = RxCAN |
| J8 | ShCo | PC5 = ShCo (Shunt+) |
| J9 | GNDm | PC4 = GNGm (shunt-) |
| J21 | Open | PB2 is not used |
| J22 | Hall | PD7 = H1 |
| J23 | Open | PB5 is not used |
| J24 | Hall | PC6 = H2 |
| J25 | Open | PD6 is not used |
| J26 | Hall | PD5 = H3 |

# 7. Appendix

**Figure 7-1.** 42BLS01-001 Motor Characteristics




| Item | Specifications | Conditions |
|---|---|---|
| N° phase | 3 | |
| No of poles | 8 | |
| Rated voltage | 12 VDC | |
| No load speed | 6200 RPM | |
| No load current | 0,5A | |
| Rated speed | 4000 RPM +/-5% | |
| Rated torque | 0,0625 N.m | |
| Torque constant | 0,0168 N.m/A | |
| Resistance line to line | 0,44 ohms +/-10% | at 20°C |
| Inductance line to line | 0,51 mH +/-20% | |
| Peak torque | 0.19 N.m | |
| Rotor inertia | 24 g.cm² | |
| Mass | 0,3 Kg | |
| Dielelectric strength | 500 VDC | 1 minute |
| Insulation class | Class B | |
| Insulation resistance | 100 Mohm min, | |
| Winding type | Delta | |
| Hall effect angle | 3 x 120° | |
| Radial force max | 28N | 20 mm flange |
| Axial force max | 10N | |

| Lead color | Function | Lead gauge |
|---|---|---|
| YEL | Phase U | UL1007 AWG20 |
| RED | Phase V | |
| BLK | Phase W | |
| RED | Hall +5V | UL1007 AWG26 |
| BLU | Hall sensor 1 | |
| GRN | Hall sensor 2 | |
| WHT | Hall sensor 3 | |
| BLK | Hall GND | |

## Headquarters

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

## International

***Atmel Asia***
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
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***
www.atmel.com

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

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

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