Micro-stepping for Stepper Motors

By
Marc McComb
Technical Training Engineer
Security, Microcontroller and Technology Development Division
Microchip Technology Inc.

Introduction

Stepper motor performance can be compromised by noise and vibration, but an advanced high-torque micro-stepping algorithm can minimise the impact of both noise and vibration as well as increasing the resolution of the rotations. The embedded microcontroller which hosts the micro-stepper algorithm also provides additional flexibility to integrate current-limiting as well as adjusting the drive characteristics in response to changes in the dynamics of the system.

White Paper

Stepper motors provide unique high-torque and positioning characteristics, along with the ability to hold a load at a specific rotational angle. Furthermore, these motors can be implemented as part of an open-loop positioning system without the additional costs associated with feedback circuitry. However, stepper motors are susceptible to vibration that could prove problematic at certain rotational speeds, especially when using lower-resolution stepping algorithms.

For example, using a two-phase bipolar stepping motor, a common practice is to implement a two-phase-on switching algorithm that energizes both phases simultaneously for each full step. Current direction in each phase is alternately changed from step to step, providing 41% more torque over full-stepping methods that energize one winding at a time. Problems may arise, as the rotor will tend to oscillate briefly on each step as a result of a number of system dynamics—including kinetic energy (momentum)—that cause the rotor to overshoot the detent position. The rotor, in an attempt to minimize reluctance, must then correct itself and travel in the opposite direction. Momentum will cause the rotor to overshoot once again, and the process repeats until energy finally dissipates.

What makes matters worse is that, as the step rate increases, there is a frequency at which this oscillation and the next step pulse in the sequence will coincide. This leads to missed steps, as the rotor never truly comes to rest at the previous detent position. The consequences of a missed step could range from simple vibration and noise issues to the catastrophic failure of an application, especially if the motor is used as part of an open-loop system.

Modern Hybrid stepping motors minimize the effects of overshoot by implementing teeth on both a permanent-magnet rotor and stators that can commonly reduce step angles to 0.9° and 1.8°. The distributed tooth construction dramatically increases magnetic field concentration, thereby increasing available torque. These motors, when used in conjunction with advanced stepping algorithms such as microstepping, can reduce noise and vibration while increasing rotational resolution.

Microstepping smoothes the rotor's rotation by dividing each full step by 4/8/16/32 smaller steps (typically, divisions of the full-step angle greater than 32 microsteps do not improve rotor motion). The method discussed in this article accomplishes this microstepping by varying the current in each phase in a sinusoidal fashion called Sine/Cosine microstepping. Two variations of a Sine/Cosine microstepping technique are shown in Figure 1.
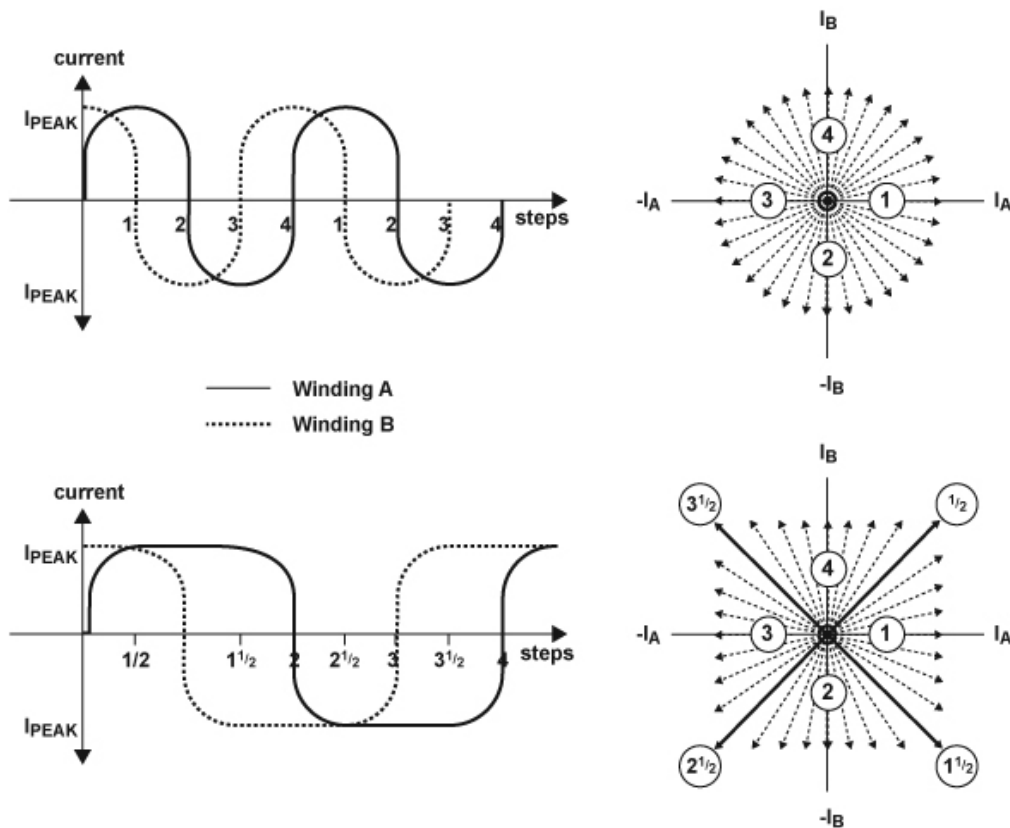
**Figure 1: Microstepping's Resulting Stator Current
(Constant Torque and High Torque Methods)**

A two-phase stepper motor is used as an example here, with the current in both windings separated by a 90° phase difference. It is important to note here that 360°, or one complete electrical cycle, differs from a 360° mechanical rotation of the rotor. The current diagrams shown in Figure 1 reveal that one full step actually represents 90° of electrical angle. The instantaneous current in each winding is calculated as follows:

Equation 1:

$$I_A = I_{PEAK} * \sin \theta$$
$$I_B = I_{PEAK} * \cos \theta$$

where:

I_A	= instantaneous current in Winding A
I_B	= instantaneous current in Winding B
I_{PEAK}	= manufacturer's specified maximum current rating per winding
θ	= electrical angle

On the right side of Figure 1 are the associated current-phase diagrams, with the x-axis representing the current in Winding A and the y-axis the current in Winding B. The length of each phasor represents the resulting vector sum of both currents at a given electrical angle, as calculated using Equation 2:

Equation 2:

$$\textbf{phasor length} = \sqrt{\textbf{I}_{\textbf{A}}{}^{2} + \textbf{I}_{\textbf{B}}{}^{2}}$$

Equation 3 demonstrates that the length of each phasor is also proportional to the torque at a given angle.

Equation 3:

$T_A = H * \sin \Theta$
$T_B = H * \cos \Theta$

where:

T_A = **instantaneous torque in Winding A**
T_B = **instantaneous torque in Winding B**
H   = **manufacturer's specified maximum holding torque**

The first method shown in Figure 1 maintains a constant torque (all phasors the same length) throughout rotor rotation, by constantly varying the current in both windings. A second method implements a high-torque microstepping algorithm. Here, the peak current level in one winding is maintained while the current in the other winding is gradually dropped to zero, reversed and then ramped up again. This sequence is then repeated for the other winding. Referring to the associated phasor diagram, note the effect on the torque. With the current in one winding held at its maximum while the other transitions, phasor length dynamically changes to provide higher average torque than generated in the first method. Note in particular the thicker phasors on the associated diagram at steps ½ (45°), 1½ (315°), 2½ (225°) and 3½ (135°). These points indicate the maximum torque positions in the algorithm, which naturally occur when both phases are energized to their maximum rated current. Note that these are the full-step points corresponding to the detent positions used by the two-phase-on, full-step switching algorithm discussed earlier.

A variety of methods can be used to dynamically change the current in each winding to implement High-Torque Sine/Cosine microstepping. The method described in this article utilizes a PIC16HV616 microcontroller (MCU) and associated peripherals.

The Enhanced Capture/Compare/PWM (ECCP) peripheral is capable of generating a Pulse-Width-Modulated waveform with a software-programmable duty cycle. This peripheral also features a half-bridge output mode that can modulate two pins simultaneously. Varying the duty cycle of a PWM output pin connected to the drive circuitry of a winding will vary the current in the winding proportionally, as shown in Equation 4.

Equation 4:

$I_{WINDING}$ = **Duty Cycle x** I_{PEAK}

For example, if a 3V stepper motor has a rated current of 0.5 amps, modulating a 3V supply with a 50% duty cycle across the coil results in an average current of 0.25 amps.

To implement the high-torque method, firmware on the PIC16HV616 utilizes a look-up table of duty cycles to produce sinusoidal current transitions with the number of table values corresponding to the number of microsteps. Since each full step represents 90° of electrical angle, an 8 microsteps/full-step algorithm can be calculated as per Equation 5 to generate the table values shown in Table 1.

Equation 5:

90°/8 = 11.25°/step

Duty Cycle (step #) = (sin (step # * 11.25)) * 100 (to obtain duty cycle in %)

Example:

Calculate Step 2 Duty Cycle

Duty Cycle (step 2) = (sin (2*11.25°))*100 = 38% (rounded to a whole number)

Step Number	Winding A Duty Cycle (%)	Winding B Duty Cycle (%)	
1	20	100	Full-Step 1 (0 to 90°)
2	38	100	
3	56	100	
4	71	100	
5	83	100	
6	92	100	
7	98	100	
8	100	100	
1	100	100	Full-Step 2 (90° to 180°)
2	100	98	
3	100	92	
4	100	83	
5	100	71	
6	100	56	
7	100	38	
8	100	20	

Table 1: PWM Duty Cycles for 1/8 Microstepping to Complete 2 Full Steps

Two complete full steps are shown. The first half of the table varies the duty cycle to the drive circuitry of Winding A, transitioning the current across the coil from 0 to I_{PEAK} while maintaining a maximum current in Winding B. At the end of the first eight microsteps, 90° of electrical angle or one full step has been completed. In the next eight microsteps, the current in Winding A is now held, while the current in Winding B is decreased by simply reversing the order of the duty-cycle values used to ramp Winding A in the previous 90° of transition.

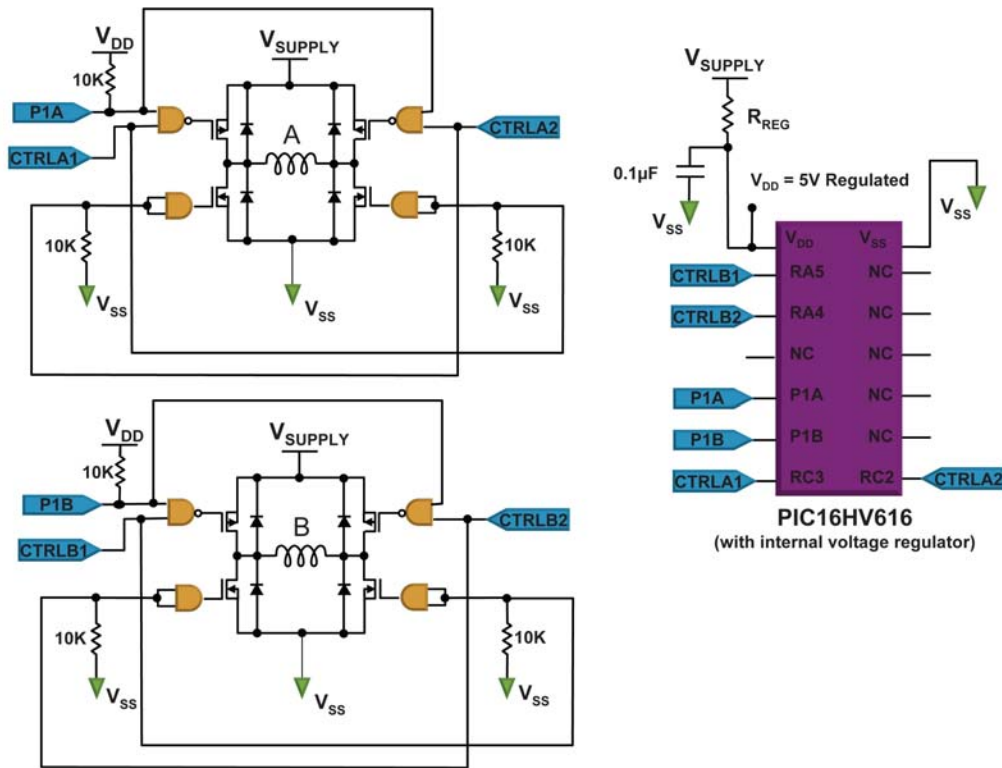An example circuit diagram using the PIC16HV616 is shown in Figure 2.

Figure 2: Microstepping Circuit Using the 8-bit PIC16HV616 Microcontroller

The 8-bit PIC16HV616 MCU features a permanent internal 5V shunt regulator that can help limit external regulation components while allowing higher supply voltages to be used, optimizing the current and therefore the torque generated by the winding. The size of the required external current-limiting resistor (R_{REG}) is determined using equations provided in the PIC16HV616 datasheet.

The drive circuit consists of AND and NAND buffer ICs to interface the high side of each H-Bridge with the PWM output pins P1A and P1B and four General Purpose I/O PORT pins RA5 (CTRLB1), RA4 (CTRLB2), RC3 (CTRLA1) and RC2 (CTRLA2). Driving a PORT pin HIGH will configure the coil current direction by turning on the associated n-channel MOSFET while enabling the signal present on the PWM pin. The PWM output pins are configurable as either input or output. When configured as an output, the PWM waveform is present on the pin. When configured as an input, the PWM pin is in a high-impedance state, allowing the pull-up resistors connected to the pin to clamp the line HIGH.

To implement the first eight microsteps shown in Table 1, the RC3 pin connected to CTRLA1 is driven HIGH, turning on the lower right n-channel MOSFET on the Winding A bridge. The P1A pin is configured as an output and firmware begins stepping through the look-up table, changing the duty cycle of the PWM output to the values calculated at each step. The Winding B current is held at a maximum with firmware driving the CTRLB1 pin HIGH and configuring the P1B pin as a high-impedance input. The 10K resistor connected to P1B clamps the line HIGH and maximum current flows across the winding. The remainder of the algorithm is simply a variation on these steps.

In summary, High-Torque microstepping provides a means of electronically damping the stepper motor's response through overshoot suppression. The application example described in the preceding article, easily accomplishes the task by implementing a PIC16HV616 microcontroller along with some common peripherals. Integrating an embedded solution into the design further adds a level of intelligence that can accommodate more advanced drives incorporating current limiting techniques; or even modify the behavior of the drive as system dynamics change. In this way, designers now have a few extra tools at their disposal to help take full advantage of the benefits associated with this truly unique motor.

###

About Microchip Technology
Microchip Technology Inc. (NASDAQ: MCHP) is a leading provider of microcontroller and analog semiconductors, providing low-risk product development, lower total system cost and faster time to market for thousands of diverse customer applications worldwide. Headquartered in Chandler, Arizona, Microchip offers outstanding technical support along with dependable delivery and quality. For more information, visit the Microchip website at http://www.microchip.com.

References:

de Silva, C.W., 2005, *Mechatronics: An Integrated Approach,* CRC Press, New York, NY, 1312 p.

Condit, R, 2004, Microchip Technology, *AN906: Stepper Motor Control Using the PIC16F684 (DS00906B),* http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en012150 (December 2, 2008)

Yedamale, P, Chattopadhyay, S, 2002, Microchip Technology, *AN822: Stepper Motor Microstepping with PIC18C452 (DS00822A),* http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en011997 (December 2, 2008)

###

MCA486wp