

Microstepping Bipolar Drive of Two-Phase Hybrid Stepping Motor on TMS320F2808 DSC

*Chawakorn Yodsanti and Mongkol Konghirun**King Mongkut's University of Technology Thonburi*

ABSTRACT

This application report describes the implementation of a micro-stepping algorithm for a two-phase stepper motor using the TMS320F2808 digital signal controller (DSC). The "C" code can easily be utilized with other members of the C2000™ DSC platform of devices.

Project collateral and source code discussed in this application report can be downloaded from the following URL: <http://www.ti.com/lit/zip/SPRAAU7>.

Contents

1	Introduction	2
2	Microstepping Scheme With Current Controlled	2
3	Experimental Results	3
4	Conclusion	9
5	References	9

List of Figures

1	Dual H-Bridge Circuit Connecting to a Two-Phase Hybrid Stepping Motor	2
2	Overall System Using DSP	3
3	Overall Hardware Setup	4
4	Code Composer Studio Screen Capture With Graphs During Run-Time (the captured variables are discrete angle, reference current, feedback current, etc.)	4
5	Overall Software Flowchart	5
6	Responses for Precision = 1/8 Step	6
7	Responses for Precision = 1/16 Step	7
8	Responses for Precision = 1/100 Step	7
9	Phase Current and Rotor Position Responses for Different Precisions	8

List of Tables

1	Code Size, Number of Cycle and MIPS Used for Microstepping Bipolar Drive of Two-Phase Hybrid Stepping Motor	9
---	---	---

1 Introduction

Hybrid stepping motors are used in a wide variety of position controlled equipments such as plotters, CNCs, printers, robots, etc. However, in high precision applications, the microstepping scheme is necessary for accurate motor rotation. Two main advantages of this scheme have been well reported in literature such as: reduction of resonance behavior [1], [2], [4] and smooth movement with very low ripple torque [3]. In this application report, the microstepping scheme with configurable fractional step is implemented using a fixed-point TMS320F2808 DSC from Texas Instruments, Inc.

The discrete angle for discretized sinusoidal voltage commands is generated by the zero hold order (ZOH) module. The motor currents are controlled by using unipolar pulse width modulation (PWM) technique. The dual H-bridge circuit is primarily used to drive a two-phase hybrid stepping motor as shown in Figure 1. Each phase winding is connected to each H-bridge circuit, therefore, four switching devices are independently controlled to generate appropriate voltage for each phase winding. The main advantage of this circuit topology is the independent generation of bipolar voltage between two phases. In low-cost, low-precision applications, the conventional full or half step scheme is selected to implement without the PWM technique. However, high performance systems employ microstepping to achieve accurate motor phase current control.

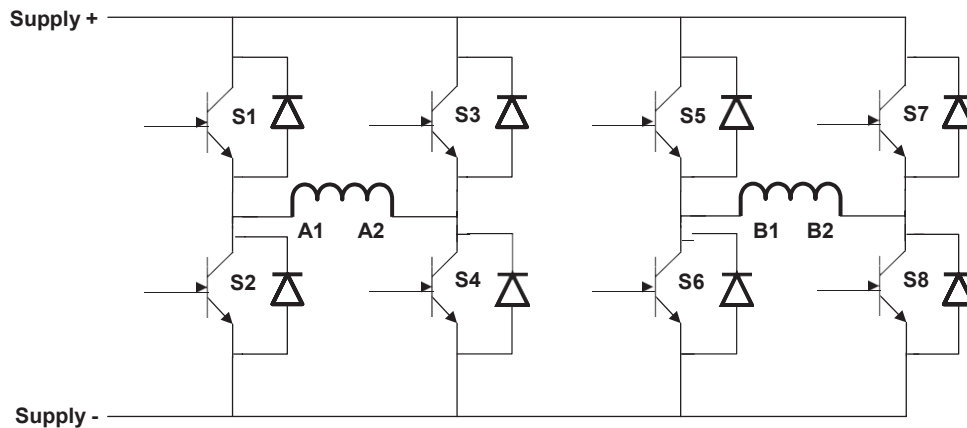


Figure 1. Dual H-Bridge Circuit Connecting to a Two-Phase Hybrid Stepping Motor

2 Microstepping Scheme With Current Controlled

Figure 2 shows the overall system of a microstepping bipolar drive using a TMS320F2808 DSC. The bipolar drive is well known for its advantage of 20%-30% increased torque capability, compared to the unipolar drive [4]. Notice that the reference variables are defined with a superscript * in Figure 2. In the system, the peak of the two-phase motor currents is controlled by a PI controller. The output of this PI is the peak command voltage V_p^* . The feedback of the peak current is calculated from the measured phase-a and -b currents by the following:

$$i_p = \sqrt{i_a^2 + i_b^2} \quad (1)$$

The continuous angle (θ_{con}) is discretized to obtain discrete angle (θ_{disc}) by the ZOH block. The number of fractional steps can be adjusted by this block. Then, θ_{disc} is used to compute the discrete sinusoidal command voltages with the peak determined by PI output expressed as follows:

$$v_a^* = v_p^* \cos \theta_{disc} \quad (2)$$

$$v_b^* = v_p^* \sin \theta_{disc} \quad (3)$$

Once the command voltages (2) and (3) are computed, then the duty cycle of each switching device in the dual H-bridge is determined by using the unipolar PWM technique. Based on this technique, the command voltages are compared with a triangle signal (V_{tri}) with a fixed switching frequency. The rules of four switching devices for phase-a are determined as follows:

- when $V_a^* > V_{tri}$, S1 on and S2 off
 - Otherwise, S1 off and S2 on
 - when $-V_a^* > V_{tri}$, S3 on and S4 off
 - Otherwise, S3 off and S4 on
- (4)

Similarly, the same rules are applied for other four switching devices (S5-S8) for phase-b winding, using phase-b command voltage V_b^* . The main advantage of the unipolar PWM technique is the reduction of ripple currents due to output voltages of the double switching frequency.

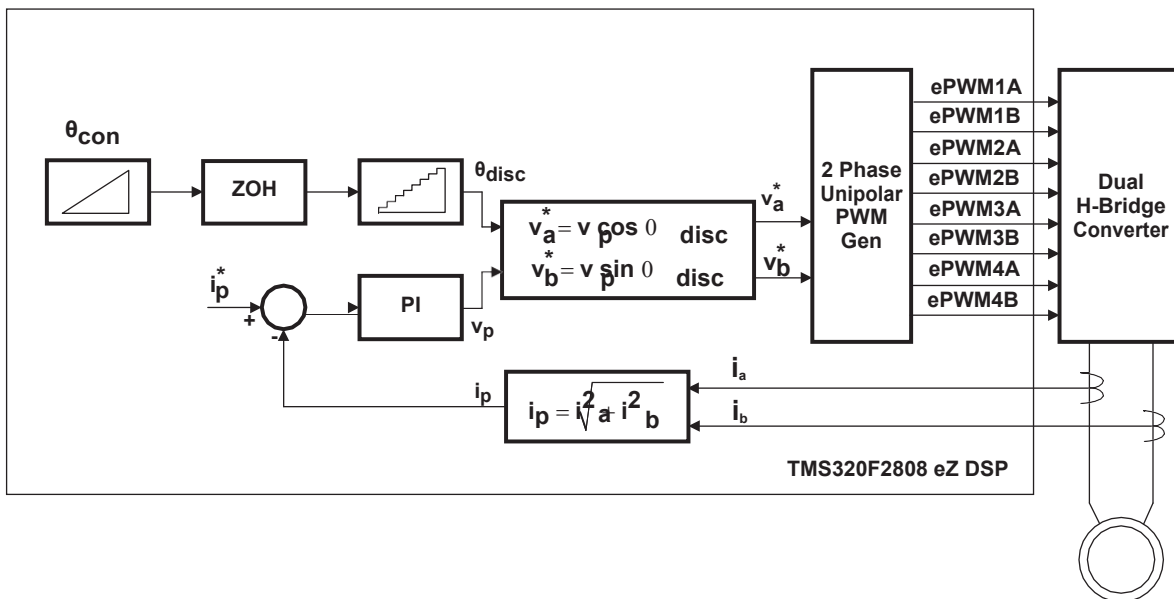


Figure 2. Overall System Using DSP

3 Experimental Results

The overall system shown in Figure 2 is implemented using a fixed-point TMS320F2808 DSC operating at 100 MHz system clock [5]. A photograph of the overall hardware setup is shown in Figure 3. In this system, the switching frequency is set at 10 kHz. This is also the frequency of the interrupt service routine (ISR) where overall algorithms are executed. In addition, two measured currents are sampled at this frequency using the on-chip 12-bit analog-to-digital (ADC) converter of DSC. The DSC is configured to generate the 1-μs of dead time for upper and lower switching devices. The two-phase stepping motor is rated 3A and 1.8° step angle. Code Composer Studio™ V3.1 is used as a DSP development tool and is capable of plotting graphs of any variables in the codes during run-time [6]. Figure 4 shows a screen capture of Code Composer Studio with graphs. Some of the experimental results in this application report are conveniently obtained from the Code Composer Studio graphs. The software flowchart of the overall implementation can be shown in Figure 5.

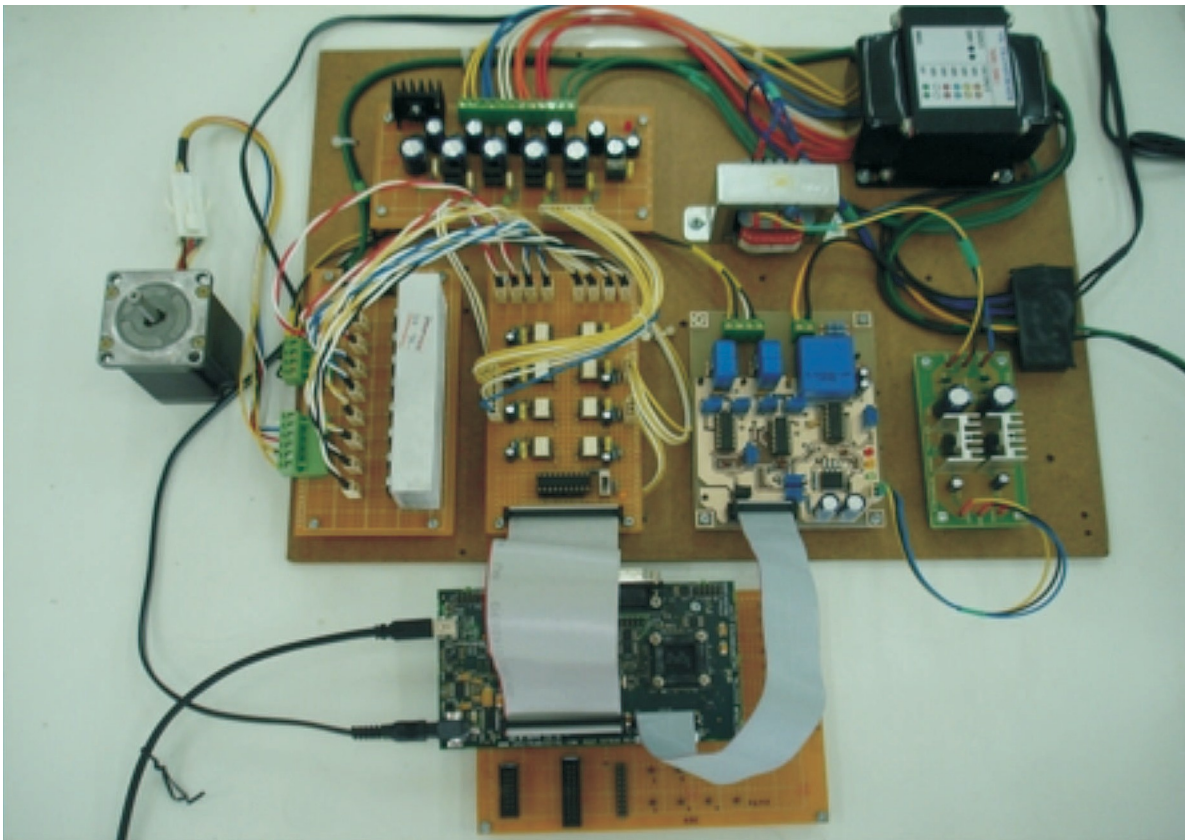


Figure 3. Overall Hardware Setup

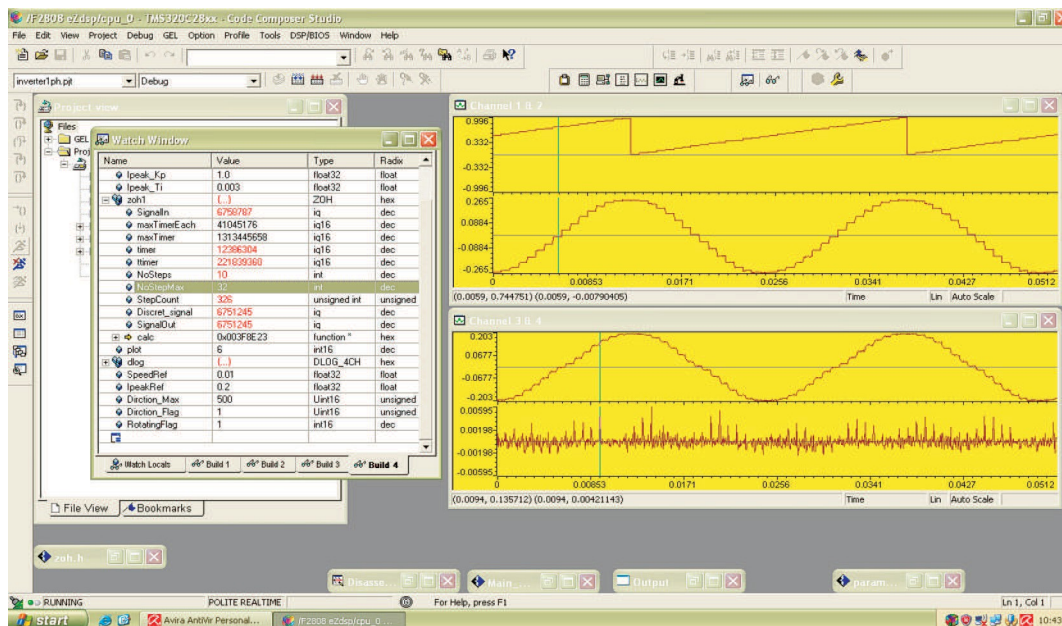


Figure 4. Code Composer Studio Screen Capture With Graphs During Run-Time (the captured variables are discrete angle, reference current, feedback current, etc.)

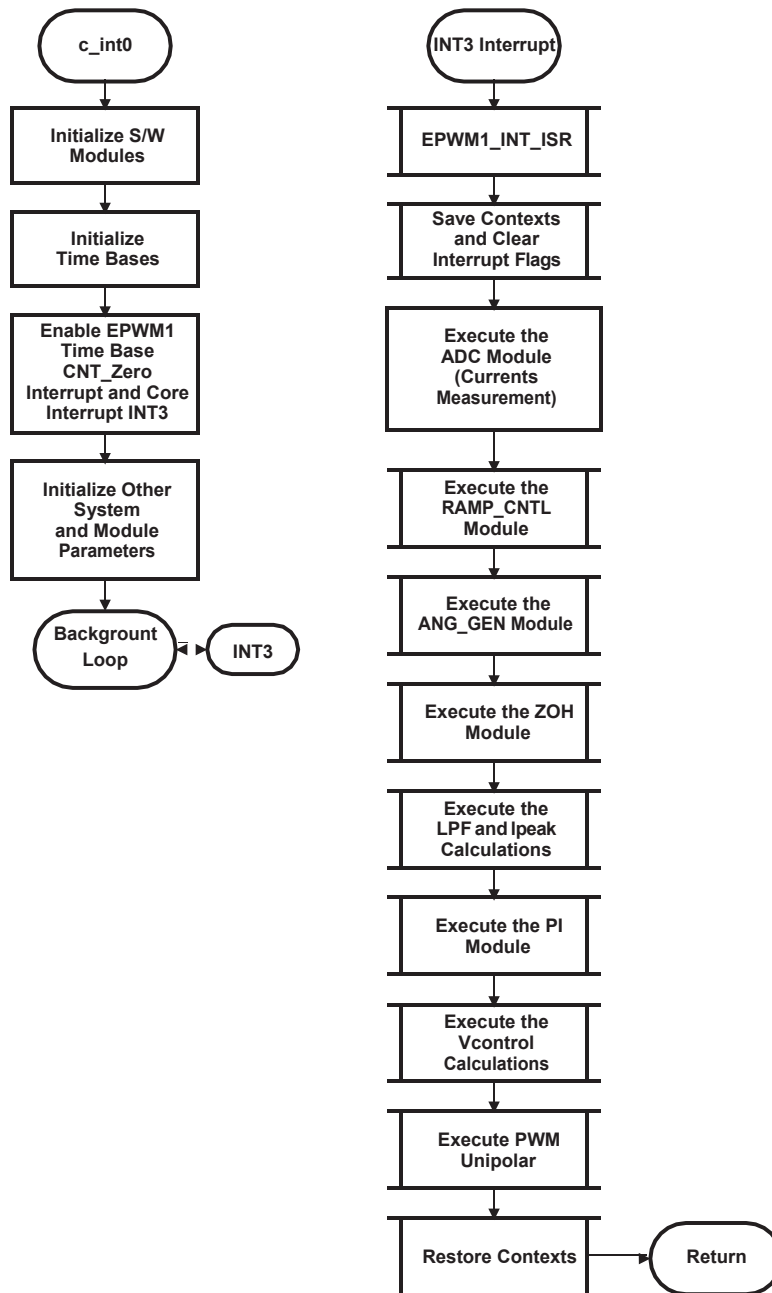
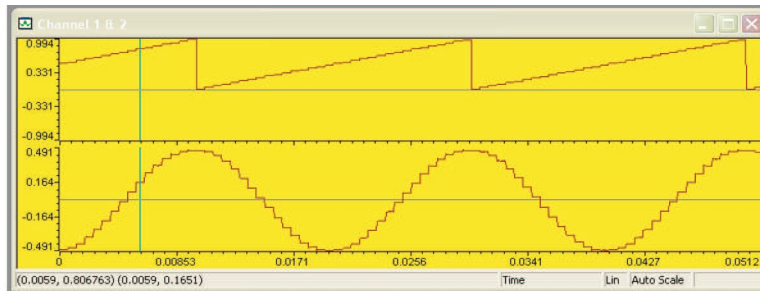
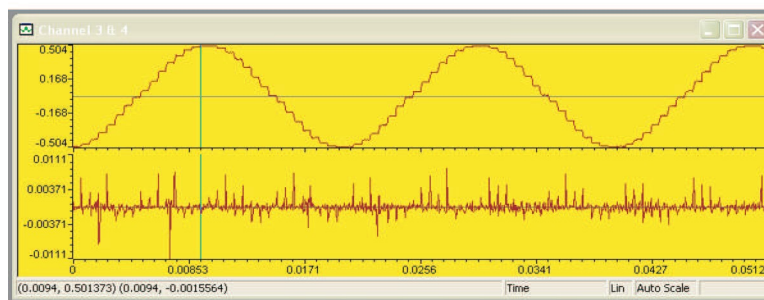


Figure 5. Overall Software Flowchart

Figure 6 through Figure 8 show the captured Code Composer Studio graphs of discrete angle, reference current, feedback current, and current error when the step precision is configured by the ZOH block for precision = 1/8, 1/16, and 1/100 step, respectively. As seen from these figures, the current is successfully controlled with a peak of 0.5 per-unit (base current of 2A) regardless of the step precision. The current errors in Figure 6(b), Figure 7(b), and Figure 8(b) are shown to validate the successful implementation of current control. As the fractional step is reduced, the current waveforms become less and less discretized and eventually sinusoidal. When the currents are sinusoidal waveforms, the smooth rotation of the stepping motor is obtained with small ripple torque. This is a necessary requirement in high precision applications.

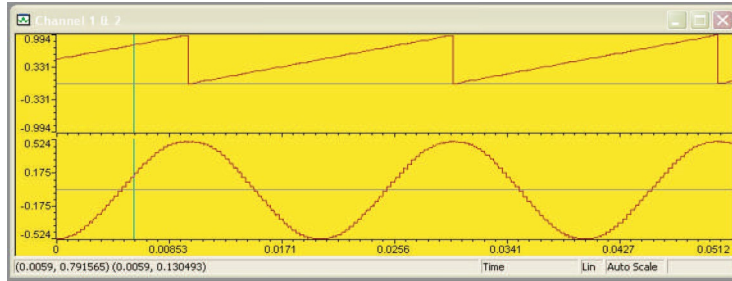


(a) Discrete Angle and Reference Current

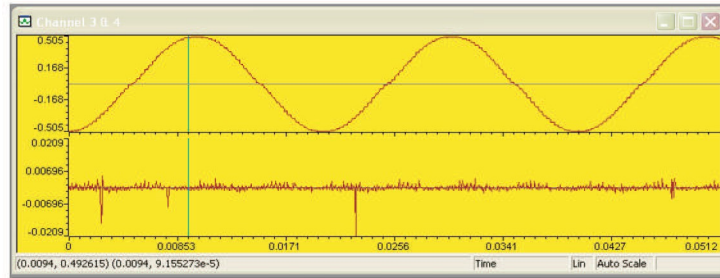


(b) Feedback Current and Current Error

Figure 6. Responses for Precision = 1/8 Step

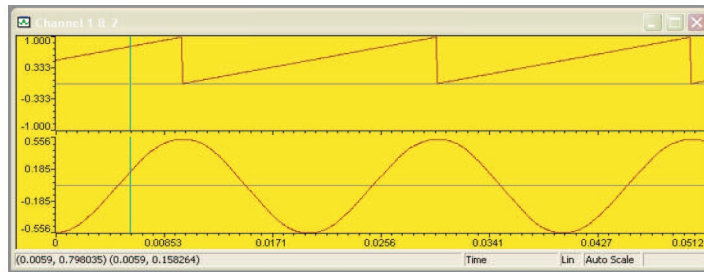


(a) Discrete Angle and Reference Current

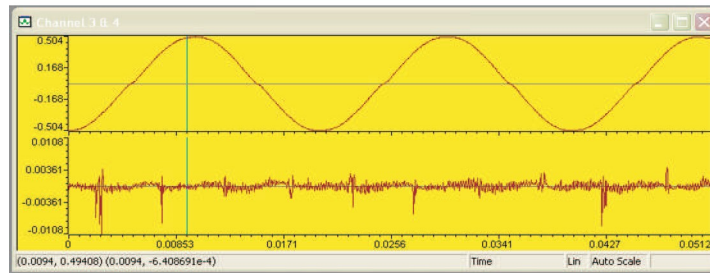


(b) Feedback Current and Current Error

Figure 7. Responses for Precision = 1/16 Step



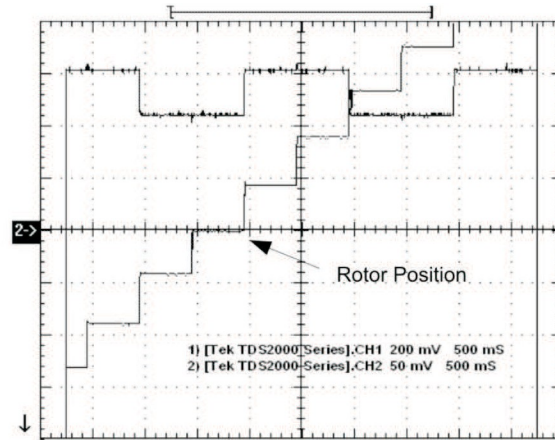
(a) Discrete Angle and Reference Current



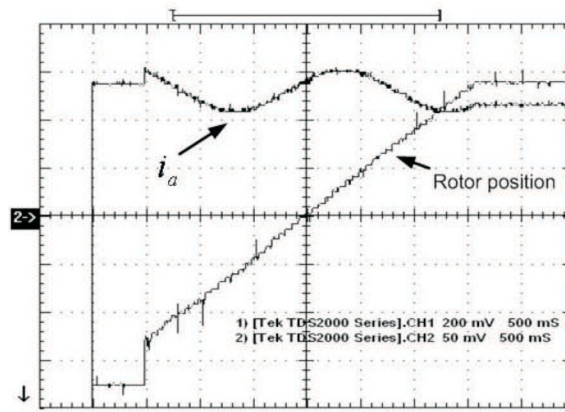
(b) Feedback Current and Current Error

Figure 8. Responses for Precision = 1/100 Step

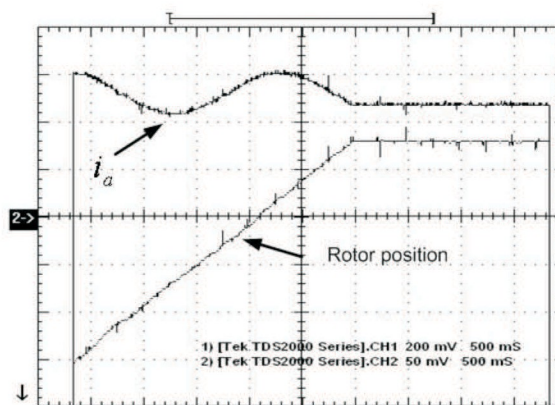
In Figure 9, phase-a current and rotor position of the stepping motor are shown for (a) full step, (b) 1/8 step, and) 1/100 step. The experiments are tested under the same peak current controlled and same stepping rate. As seen in Figure 9, the peak of current is always constant for three different step precisions because of successful current control. However, the movement of the rotor is extremely smooth for precision = 1/100 step, compared to the full step angle (Figure 9(a)) and 1/8 step (Figure 9(b)) precisions.



(a) Full Step



(b) 1/8 Step



(b) 1/100 Step

Figure 9. Phase Current and Rotor Position Responses for Different Precisions

Using the profiling capabilities of the Code Composer Studio, [Table 1](#) summarizes the code size and number of cycles used for key functional blocks in the system:

Table 1. Code Size, Number of Cycle and MIPS Used for Microstepping Bipolar Drive of Two-Phase Hybrid Stepping Motor

	Code Size (words)	Cycles Used	MIPS Used ⁽¹⁾ (%)
RAMP_CNTL	80	54	0.54
ANG_GEN	65	95	0.95
ZOH	75	236	2.36
PI	100	146	1.46
PWM	311	115	1.15
Total Functional Blocks	3104	1581	15.81

⁽¹⁾ The MIPS used is calculated based on an ISR frequency of 10 kHz.

4 Conclusion

Microstepping is a drive technique of the stepping motor that allows the smooth movement of the rotor in a fraction of the motor's full step angle. This application report presents a digital implementation of a microstepping bipolar drive system with current control using a fixed-point TMS320F2808 DSC. The dual H-bridge converter is designed to drive the two-phase stepping motor. The fractional step can be adjusted/configured using the ZOH block. The rotor movement can be controlled at 1/100 step resolution. The demo code is written in "C" and can be downloaded from TI web site at www.ti.com.

5 References

1. N.Q. Le and J.W. Jeon, *An Open-Loop Stepper Motor Driver Based on FPGA*, IEEE International Conference on Control, Automation and Systems (ICCAS '07), pp. 1322-1326, 2007.
2. S-M. Yang and E-L Kuo, *Damping a Hybrid Stepping Motor With Estimated Position and Velocity*, IEEE Transaction on Power Electronics, vol. 18, no. 3, pp. 880-887, May 2003.
3. J.D. Wale and C. Pollock, *A Low-Cost Sensorless Technique for Load Torque Estimation in a Hybrid Stepping Motor*, IEEE Transaction on Industrial Electronics, vol, 46, no. 4, pp. 833-841, August 1999.
4. T. Kenjo, *Stepping Motors and Their Microprocessor Controls*, Clarendon Press, Oxford, 1984.
5. eZdsp™ F2808 USB: *Technical Reference*, Spectrum Digital Inc., Texas, 2005.
6. *Code Composer Studio Development Tools v3.3: Getting Started Guide* ([SPRU509](#))

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated