Utilizing the DRV10 Frequency Generator for Closed-Loop Speed Control

Sean Rauchfuss   Motor Driver Business Unit

ABSTRACT

This application report discusses a sensorless 3-phase BLDC motor control solution using the DRV10866 as the motor controller and MSP430 for closed-loop control. Discussion includes how to utilize the Frequency Generator (FG) output pin of the DRV10 family to achieve a constant user-defined motor RPM despite variables such as changing load. This is accomplished through a closed-loop control algorithm on the MCU. This document is provided as a supplement to the DRV10866 low voltage sensorless BLDC motor driver device's datasheet (SBVS206). The sample code described in this document can be downloaded from www.ti.com/lit/zip/SLVA557.

Contents

List of Figures

List of Tables

1 Introduction

BLDC motors have permanent magnets that rotate (rotor) and a fixed armature (stator). An electronic controller is needed for motor commutation instead of the brushed commutation used in brushed DC motors. BLDC motors offer many advantages over brushed DC motors, including increased speed versus torque efficiency, longer life (as no brushes are used), noiseless operation, and increased efficiency in converting electrical power to mechanical power (especially because there are no electrical or frictional losses due to brushed commutation). This application report discusses a sensorless commutation controller, DRV10866, and uses the MSP430 microcontroller to implement closed-loop control.

The DRV10866EVM is a complete solution for evaluating the DRV10866 5-V, three-phase sensorless BLDC motor driver. On its own, the DRV10866EVM is capable of open loop control. This open loop control can be handled either by using an onboard 555 timer to generate varying PWM signals, or by adjusting VCC on the device. Modifying the DRV10866EVM to allow for closed-loop speed control via a microcontroller is also quite simple to implement. The MSP430G2553 LaunchPad is ideal for this application thanks to its PWM capabilities and internal timers. Closed-loop control is used in applications that require more accurate and adaptive speed control of the system. These controls use feedback to direct the output states of a dynamic system. Closed-loop controls overcome the drawbacks of open-loop control to provide compensation for disturbances in the system, stability in unstable processes, and reduced sensitivity to parameter variations (dynamic load variation). Figure 1 visualizes the closed-loop control process mentioned above.
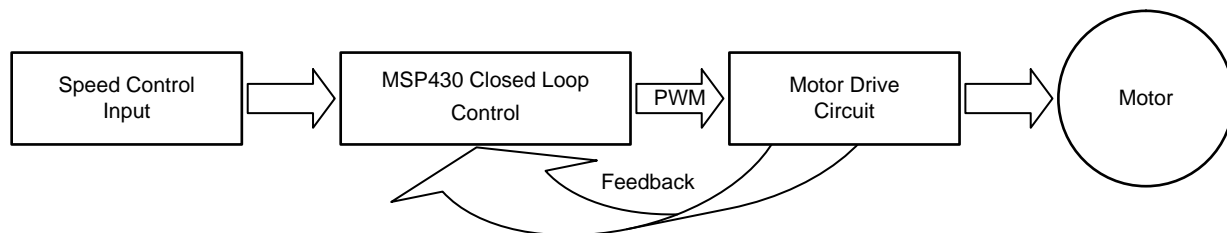
Figure 1. Closed-Loop Motor Control Block Diagram

2 Application

2.1 *DRV10866 Important Characteristics*

The DRV10866 is a three-phase, sensorless motor driver with integrated power MOSFETs with drive current capability up to 680-mA peak. The DRV10866 is specifically designed for low noise, low external component count fan motor drive applications. The DRV10866 has built in overcurrent protection with no external current sense resistor needed. The synchronous rectification mode of operation achieves increased efficiency for motor driver applications. The driver can output either FG or ½ FG to indicate motor speed with open-drain output through FGS pin selection. A 150° sensorless BEMF control scheme is implemented for a three-phase motor. A voltage surge protection scheme prevents the input VCC capacitor from overcharge during motor acceleration and deceleration modes. The DRV10866 has multiple built-in protection blocks including UVLO, overcurrent, lock and thermal shut down protection.

2.1.1 Speed Control

The DRV10866 can control motor speed through either the PWMIN or VCC pin. Motor speed will increase with higher PWMIN duty cycle or VCC input voltage. The curve of motor speed (RPM) versus PWMIN duty cycle or VCC input voltage is close to linear in most cases. However, motor characteristics will affect the linearity of this speed curve. The DRV10866 can operate at very low VCC input voltage, down to 1.65 V. The PWMIN pin is pulled up to VCC internally and the frequency range can vary from 15 kHz to 50 kHz. The motor driver MOSFETs will operate at constant switching frequency of 156 kHz. With this high switching frequency, the DRV10866 eliminates audible noise and reduces the ripple of VCC input voltage and current, and thus minimize EMI noise.

2.1.2 Frequency Generator

The FG pin outputs a square waveform in the normal operation condition. The frequency of the FG signal represents the motor speed and phase information. The FG pin is an open drain output, so an external pull-up resistor is needed when connected to an external system. During startup, FG output stays at high impedance until the motor speed reaches a certain level and BEMF is detected. During lock protection condition, FG output remains high until the motor restarts and the startup process is completed. The DRV10866 can output either FG or 1/2 FG to indicate motor status with open-drain output through FGS pin selection. When FGS is pulled to VCC, the frequency of FG output is half of that when FGS is pulled to GND. Motor speed is calculated based on the FG frequency when FGS is pulled to GND, which equals to:

$$RPM = \frac{FG \times 60}{\# \text{ of pole pairs}}$$

(1)

Where FG is in hertz (Hz). Table 1 shows RPM and FG data for motors with 2 or 4 pole pairs.

Table 1. RPM & FG Correlation for 2 & 4 Pole Motors

2 Pole Pairs		4 Pole Pairs	
RPM	FG (Hz)	RPM	FG (Hz)
500	16.67	500	33.33
1000	33.33	1000	66.67
2500	83.33	2500	200
5000	166.67	5000	333.33

2.2 DRV10866EVM Setup

Ready the DRV10866EVM for external PWM control by removing resistor R6 such that its terminals leave an open circuit. See Figure 2 for locating R6. Additionally, jumper J1 needs to be closed to set FG to normal frequency output. Close jumper J2 to send the FG signal off the board while still using pull-up resistor R4.
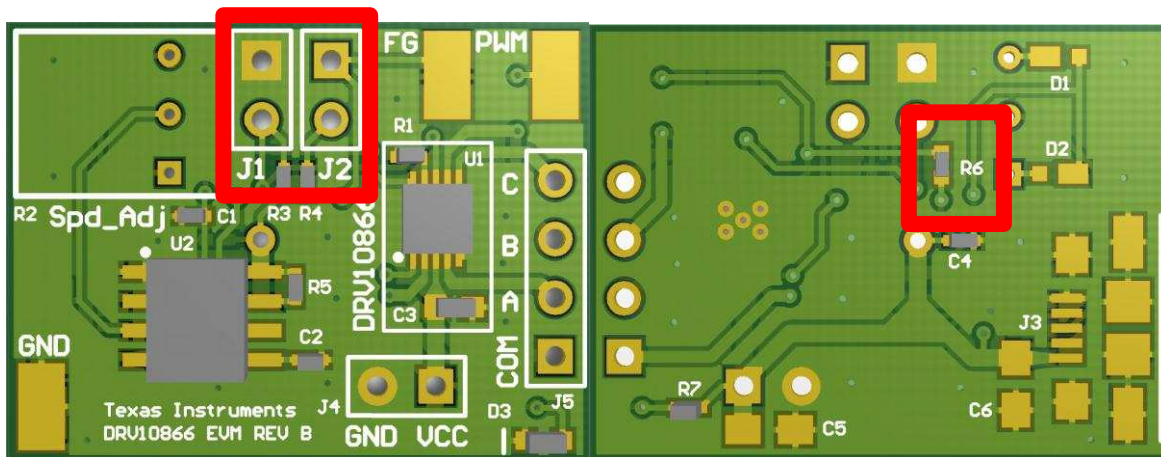

Figure 2. DRV10866EVM Capture

2.3   DRV10866EVM and MSP430G2553 LaunchPad Setup

The following pins need to be connected for proper operation using the code examples attached. A schematic of the connections needed is shown in Figure 3.

1. XIN (P2.6) of MSP430G2553 connected to PWM connector of DRV10866EVM
2. P2.3 of MSP430G2553 connected to FG connector of DRV10866EVM
3. GND Pin of MSP430G2553 connected to GND connector of DRV10866EVM
    a. Note when programing the MSP430, this connection may need to be broken
4. Motor connected to the U, V, W, and COM pins of the DRV10866EVM
5. VCC pin of MSP430G2553 connected to VCC pin of the DRV10866EVM
6. MSP430G2553 can be powered in one of two ways (don't do both):
    a. Through LaunchPad USB port
    b. 1.8–3.6 V supplied to VCC pin
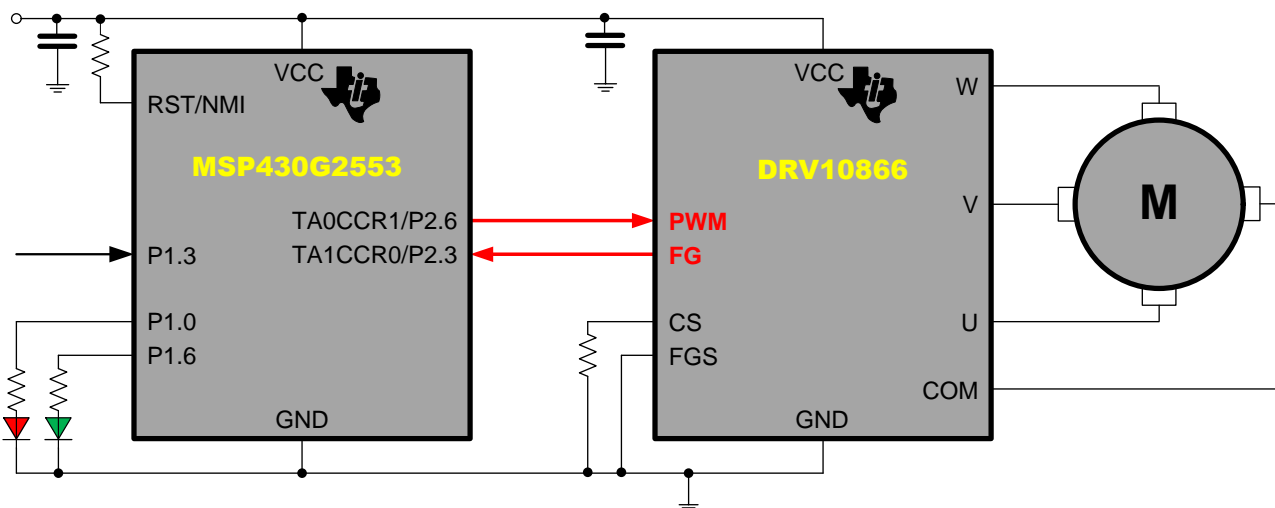
Figure 3. DRV10866EVM and MSP430G2553 LaunchPad Connections

2.4   Code Objectives

The goal of the code implemented on the MSP430 isn't to commutate the motor as that is taken care of by the DRV10866. Instead the chief objective is to successfully implement closed-loop motor control. Success for this is achieved when the system is able to keep the motor spinning at a constant user-defined speed and provide feedback to the user when the motor is and isn't at the desired speed.

Constant speed can be defined as maintaining the RPMs of a motor within an allowable range of accuracy. In this case the FG output of the DRV10866 is critical because it is directly related to the motor's RPM. The code examples associated with this app note can be edited to obtain and maintain a certain RPM on the users motor. The MSP430 utilizes an IF statement to determine based on the FG value if the PWM duty cycle needs to be increased or decreased. Through closed-loop control the MSP430 will settle to the proper PWM duty cycle for the desired FG value under the current motor load. When load changes causing the FG value to once again change the MSP430 will re-enter the IF statement to determine the new PWM duty cycle needed to again achieve the FG value desired. To best display how quickly the system is designed to adjust for a desired FG value, the S2 (P1.3) switch of the MSP430 LaunchPad is enabled such that when held, the motor will adjust to a second constant user-defined speed.

Visual feedback is provided to the user using the two onboard LEDs of the MSP430G2553 LaunchPad. LED1 (P1.0) and LED2 (P1.6) emit red and green light, respectively, when activated. The green LED2 is turned on when the desired FG value is met. When the MSP430 is adjusting the PWM duty cycle, the red LED1 is turned on to signify that the desired FG value is not currently being met. When the MSP430 detects that the motor is in lock, both the red LED1 and green LED2 are turned on.

The code utilizes interrupt based timers to capture the FG value sent by the DRV10866; more on this is in FG capture. Interrupt-based timers are also used for lock-detection status and to update the PWM value at a certain clock edge, more on this in the following sections.

2.5 Pseudo code

1. Perform all initializations and set clocks
2. PWM duty cycle value updated (on 5.c interrupt)
3. Check FG status (on 5.a interrupt)
4. Is FG at desired value?
   a. Yes
      i.   LED1 ON
      ii. LED1 OFF
      iii. Return to step 3
   b. No
      i.   LED1 ON
      ii. LED2 OFF
      iii. Too fast?
              1. Decrease PWM duty cycle
              2. Return to step 2 to update PWM
      iv. Too slow?
              1. Rotation Lock?
                 a. Yes
                    i. Interrupt 5.b
                 b. No
                    i. Increase PWM duty cycle
                    ii. Return to step 2 to update PWM
5. The following interrupt-based clocks also operate
   a. on FG signal detection
      i.   Update current FG status
      ii. Restart timer for next FG interrupt
      iii. Restart countdown timer for lock detection interrupt
   b. Lock detection
      i.   LED1 ON
      ii. LED2 ON
   c. PWM duty cycle update ready
      i.   Update PWM duty cycle
      ii. Check if button is pulled low?
              1. Yes
                    a. FG = FG desired / 2
              2. No
                    a. FG = FG desired

2.6 Code Variants

2.6.1 FG Capture

The ability to determine the current RPM of the motor is crucial to this application as the feedback is required for closed-loop control. There are several methods of capturing and using the FG frequency of the DRV10866 with the MSP430G2553 to determine the motor's RPM.

2.6.1.1 *Number of Clock Cycles Between FG Interrupts*

This is the method used in this application note to determine the motor's RPM through the FG frequency of the DRV10866. It works by capturing the number of clock cycles between two FG pulses. The advantage behind this method is that it allows for fantastic accuracy due to the high number of clock cycles that occur between each FG pulse. A second advantage is that this method allows for very fast adjustments in speed as load changes as adjustments can be made on each capture. In this situation the MSP430 is used to capture the number of clock cycles on timer A that happen, at 1 MHz for the example code provided, between each FG interrupt. The number of clock cycles to expect for a given RPM can be calculated by using Equation 2.

$$\# \text{ of cycles } = \frac{60 \times 10^6}{\text{RPM} \times \# \text{ of pole pairs}}$$

(2)

With the motor provided with the DRV10866EVM, which has 6 pole pairs, if an RPM of 4000 is desired, based on the equation above the ideal number of cycles would be 2500 between each FG capture. At 2500 cycles per FG capture the resolution error is 0.05%.

2.6.2 Rotation Lock Detection

The included code example also utilizes an additional interrupt which becomes set in the event that no FG interrupt is detected for 50 milliseconds. This delay is calculated based on the 50,000 cycle delay at clock speed of 1 MHz. When this event occurs the MSP430 turns on both the LaunchPad's red and green LEDs to signal that a lock has been detected. The DRV10866 has built-in lock protection to prevent the device from damage and overheating. The lock detect time of the DRV10866 is 3 seconds with a lock release time of 5 seconds later. During the entire lock period both LEDs on the LaunchPad will be illuminated.

2.6.3 RPM Accuracy versus Fluttering

Depending on the RPM range desired, some fluttering could be seen, depending on the characteristics of the motor used. This fluttering is often most obvious during startup and while adjusting for varying loads. There are several ways to correct any flutter which may occur. The simplest is to increase the range of acceptable FG values; the downside to this method is that it sacrifices precise RPM control. A second method would be to use a different control algorithm such as PID control which is able to settle on a desired FG value quicker.

2.6.4 Button Pulling

To allow for improved demonstration of the MSP430 as a closed-loop controller, the switch S2 can be used to pull between various desired RPMs. This allows the user to see how the control system reacts to adjust the PWM for a new desired RPM. The button pull code has been included in the example associated with this app note. The default speed of the motor shipped with the DRV10866EVM is 4000 RPM when the code is unchanged, and without pressing S2. When S2 is pressed, the motor speed halves to 2000 RPM.

2.6.5 CPU Overhead

In its entirety the coding attached with this app note requires less than 7% of the total CPU capability the MSP430G2553 is capable of. This is calculated with the MSP430 clock running at 8 MHz; it is capable of 16 MHz where the coding needed for this solution would occupy less than 3.5% of the total capacity.

2.6.6 Alternative FG Capture – Number of FG Interrupts in an Interval

A second method which could be practiced but is not used in this app note would be to count the number of FG pulses in a given time duration. For example, the watchdog timer interval of the MSP430 could be used to count the number of FG hits in a 128 ms interval. The desired motor RPM can be calculated using Equation 3.

$$\text{RPM} = \frac{\text{FG} \times 60}{\# \text{ of pole pairs}}$$

(3)

To instead calculate the optimal FG value for a desired RPM, the equation can be rewritten as:

$$FG = \frac{RPM \ \times \ \# \ of \ pole \ pairs \ \times \ interval \ duration}{60}$$

(4)

Here interval duration is in seconds. The advantages of this method are simplicity and minimal CPU overhead required. For example, with an interval duration of 128 ms, a motor with 6 pole pairs, and a desired RPM of 4000, then the number of FG interrupts during each interval that would be targeted is 51.2.

3 Code Example

The code example is attached online in .c source format.

4 References

1. *DRV10866 Data Sheet* SBVS206
2. *DRV10866EVM User Guide* SBVU020
3. *MSP-EXP430G2 LaunchPad Experimenter Board User's Guide* SLAU318C
4. *Sensored 3-Phase BLDC Motor Control Using MSP430* SLAA503

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