TI Designs

Hall Sensor-Based Trapezoidal Control of 230-V, 900-W Mains Powered BLDC Motor Drive Using DRV92250

Abstract

This application report presents a solution for the control of brushless DC (BLDC) motors using the DRV92250 controller. The DRV92250 is a high-voltage BLDC motor controller, which integrates a 16-bit RISC microcontroller and pre-driver circuits in a 48-pin LQFP package with a Power PAD heat sink. 32 KB of single-cycle (40 ns) flash memory is embedded with the digital microcontroller sub-system. The 16-bit microcontroller is instruction set compatible with the TI's MSP430F5438 family. The DRV92250 controller that enables the cost-effective design of intelligent controllers for three-phase BLDC motors by reducing the system components and increasing efficiency. Using these devices, it is possible to realize far more precise control algorithms. A complete solution proposal is comprised of control structures, power hardware topology, and control hardware. The Hall Sensor-Based Trapezoidal BLDC Motor Control code is compatible with the board TIDA-00433, which is a "230V, 900W Mains Powered Three Phase BLDC Motor Drive for Vacuum Cleaner".

1. Introduction

The economic constraints and new standards legislated by governments place increasingly stringent requirements on electrical systems. New generations of equipment must have higher performance parameters such as better efficiency and reduced electromagnetic interference. System flexibility must be high to facilitate market modifications and to reduce development time. All these improvements must be achieved while, at the same time, decreasing system cost.

Brushless motor technology makes it possible to achieve these specifications. Such motors combine high reliability with high efficiency compared to a with brush motors. The BLDC motor is conventionally defined as a permanent magnet synchronous motor with a trapezoidal Back EMF (BEMF) waveform shape. Permanent magnet synchronous machines with trapezoidal BEMF and (120 electrical degrees wide) rectangular stator currents are widely used as they offer the following advantages first, assuming the motor has pure trapezoidal BEMF and that the stator phases commutation process is accurate, the mechanical torque developed by the motor is constant; secondly, the BLDC drives show a very high mechanical power density.

2. BLDC Motors

The BLDC motor is an AC synchronous motor with permanent magnets on the rotor (moving part) and windings on the stator (fixed part). Permanent magnets create the rotor flux and the energized stator windings create electromagnet poles. The rotor (equivalent to a bar magnet) is attracted by the energized stator phase. By using the appropriate sequence to supply the stator phases, a rotating field on the stator is created and maintained. This action of the rotor - chasing after the electromagnet poles on the stator - is the fundamental action used in synchronous permanent magnet motors. The lead between the rotor and the rotating field must be controlled to produce torque, and this synchronization implies knowledge of the rotor position.

On the stator side, three-phase motors are the most common. These motors offer a good compromise between precise control and the number of power electronic devices required to control the stator currents. For the rotor, a greater number of poles usually create a greater torque for the same level of current. On the other hand, by adding more magnets, a point is reached where, because of the space needed between magnets, the torque no longer increases. The manufacturing cost also increases with the number of poles. As a consequence, the number of poles is a compromise between cost, torque, and volume.

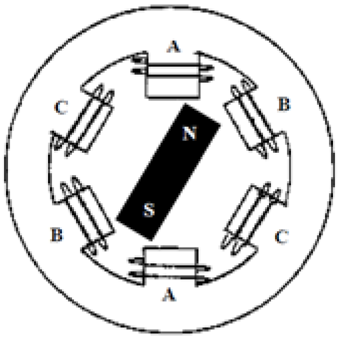


Fig 1. Three-Phase Synchronous Motor With One Permanent Magnet Pair Pole

Permanent magnet synchronous motors (PMSMs) can be classified in many ways; one classification depends on BEMF profiles: BLDC motor and PMSMs. This terminology defines the shape of the BEMF of the synchronous motor. Both BLDC motors and PMSM have permanent magnets on the rotor but differ in the flux distributions and BEMF profiles.

Table 1. Comparison of BLDC and PMSM Motors

BLDC	PMSM
Synchronous machine	Synchronous machine
Fed with rectangular currents	Fed with sinusoidal currents
Trapezoidal BEMF	Sinusoidal BEMF
Stator flux position commutation each 60 degrees	Continuous stator flux position variation
Only two phases ON at the same time	Possible to have three phases ON at the same time
Torque ripple at commutations	No torque ripple at commutations
Low order current harmonics in the audible range	Less harmonics due to sinusoidal excitation
Higher core losses due to harmonic content	Lower core loss
Less switching losses	Higher switching losses at the same switching frequency
Control algorithms are relatively simple	Control algorithms are mathematically intensive

- Both motor types are synchronous machines. The only difference between them is the shape of the induced voltage, resulting from two different manners of wiring the stator coils. The BEMF is trapezoidal in BLDC motors and sinusoidal in PMSM motors.

- BLDC machines can be driven with sinusoidal currents and PMSM with rectangular currents, but for better performance, PMSMs should be excited by sinusoidal currents and BLDC machines by rectangular currents.

- The control structure (hardware and software) of a sinusoidal motor required several current sensors and sinusoidal phase currents were hard to achieve with analog techniques. Therefore, many motors (sinusoidal like trapezoidal) were driven with rectangular current for cost and simplicity reasons (low-resolution position sensors and single low-cost current sensor), compromising efficiency, and dynamic behavior.

3. BLDC Motor Control

The key to effective torque and speed control of a BLDC motor is based on relatively simple torque and BEMF equations, which are similar to those of the DC motor. The BEMF magnitude can be written as:

$$E = 2NlrB\omega \tag{1}$$

and the torque term as:

$$T = \left(\frac{1}{2}i^2\frac{dL}{d\theta}\right) - \left(\frac{1}{2}B^2\frac{dR}{d\theta}\right) + \left(\frac{4N}{\pi}Brl\pi i\right) \tag{2}$$

where *N* is the number of winding turns per phase, *l* is the length of the rotor, *r* is the internal radius of the rotor, *B* is the rotor magnet flux density, ω is the motor's angular velocity, *i* is the phase current, *L* is the phase inductance, *θ* is the rotor position, *R* is the phase resistance.

The first two terms in the torque expression are parasitic reluctance torque components. The third term produces mutual torque, which is the torque production mechanism used in the case of BLDC motors. To sum up, the BEMF is directly proportional to the motor speed and the torque production is almost directly proportional to the phase current.

The BLDC motor is characterized by a two-phase ON operation to control the inverter. In this control scheme, torque production follows the principle that current should flow in only two of the three phases at a time and that there should be no torque production in the region of BEMF zero crossings. Figure 2 describes the electrical wave forms in the BLDC motor in the two phases ON operation.

This control structure has several advantages:

- Only one current at a time needs to be controlled.
- Only one current sensor is necessary (or none for speed loop only, as detailed in the next sections).
- The positioning of the current sensor allows the use of low cost sensors as a shunt.

The principle of the BLDC motor is, at all times, to energize the phase pair that can produce the highest torque. The combination of a rectangular current with a trapezoidal BEMF makes it theoretically possible to produce a constant torque. In practice, the current cannot be established instantaneously in a motor phase; as a consequence, the torque ripple is present at each 60-degree phase commutation.

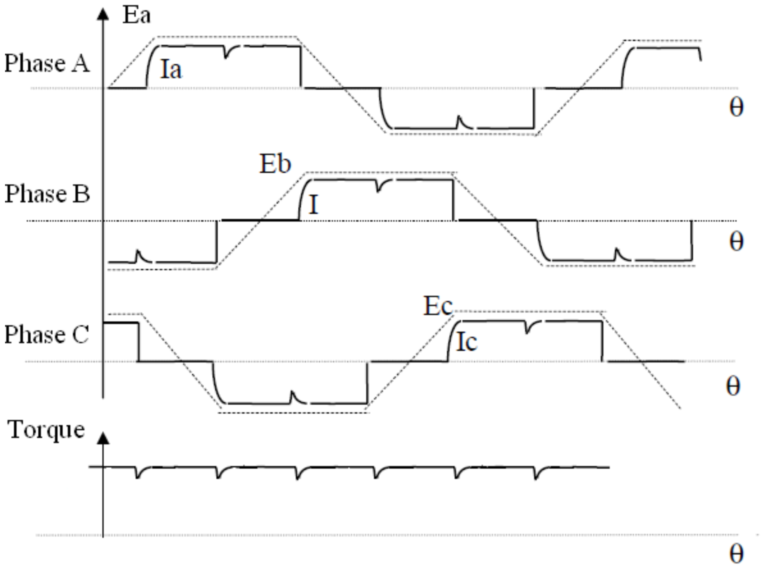


Fig 2. Electrical Waveforms in Two-Phase ON Operation and Torque Ripple

If the motor used has a sinusoidal BEMF shape, this control can be applied; however, the produced torque is:

- Firstly, not constant but made up from portions of a sine wave. This is due to its being the combination of a trapezoidal current control strategy and of a sinusoidal BEMF. Bear in mind that a sinusoidal BEMF shape motor controlled with a sine wave strategy (three phase ON) produces a constant torque.
- Secondly, the torque value produced is weaker.

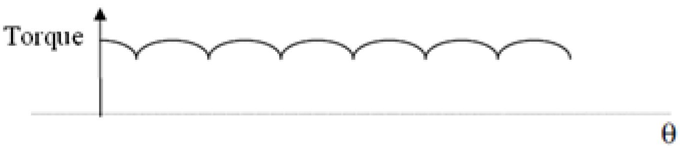


Fig.3 Torque Ripple in Sinusoidal Motor Controlled as BLDC

4. System Topology

4.1. Three-Phase Inverter

The BLDC motor control consists of generating rectangular currents in the motor phases. This control is subdivided into two independent operations: stator and rotor flux synchronization and control of the current value. Both operations are realized through the three-phase inverter depicted in Figure 4.

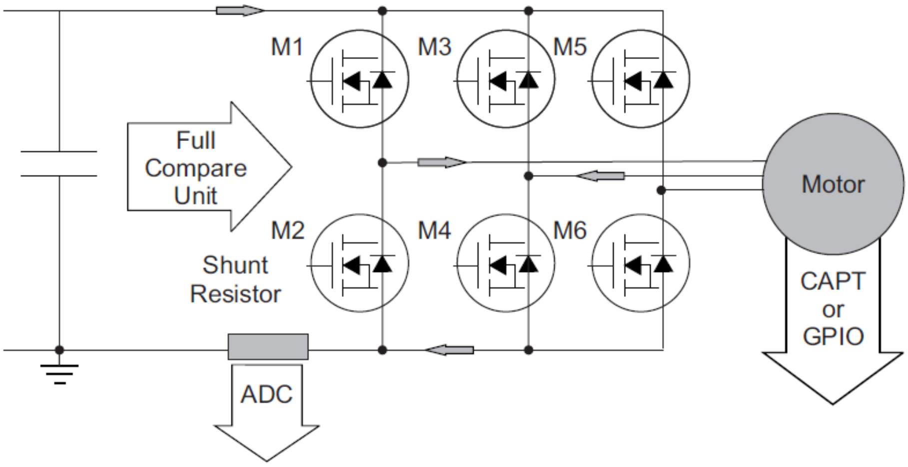


Fig.4 Three-Phase Inverter

The flux synchronization is derived from the position information coming from sensors or from sensorless techniques. From the position, the controller determines the appropriate pair of transistors (M1 to M6), which must be driven. The regulation of the current to a fixed 60-degree reference can be realized in pulse width modulation (PWM) mode or Hysteresis mode.

PWM Mode

The supply voltage is chopped at a fixed frequency with a duty cycle depending on the current error. Therefore, both the current and the rate of change of current can be controlled. The two-phase supply duration is limited by the two-phase commutation angles. The main advantage of the PWM strategy is that the chopping frequency is a fixed parameter; therefore, acoustic and electromagnetic noises are relatively easy to filter.

There are also two ways of handling the drive current switching: hard chopping and soft chopping.

In the hard chopping technique, both phase transistors are driven by the same pulsed signal: the two transistors are switched on and off at the same time. The power electronics board is then easier to design and is also cheaper as it handles only three pulsed signals. A disadvantage of the hard chopping operation is that it increases the current ripple by a large factor in comparison with the soft chopping approach.

The soft chopping approach allows not only a control of the current and of the rate of change of the current but a minimization of the current ripple as well. In this soft chopping mode the low side transistor is left ON during the phase supply and the high side transistor switches according to the pulsed signal. In this case, the power electronics board has to handle six PWM signals.

Hysteresis Mode

In the hysteresis-type current regulator, the power transistors are switched off and on according to whether the current is greater or less than a reference current. The error is used directly to control the states of the power transistors. The hysteresis controller is used to limit the phase current within a preset hysteresis band. As the supply voltage is fixed, the result is that the switching frequency varies as the current error varies. The current chopping operation is thus not a fixed chopping frequency PWM technique. This method is more commonly implemented in drives where motor speed and load do not vary too much, so that the variation in switching frequency is small. Here again, both hard and soft chopping schemes are possible. Because the width of the tolerance band is a design parameter, this mode allows current control to be as precise as desired. However, acoustic and electromagnetic noises are difficult to filter because of the varying switching frequency.

4.2. Current Sensing

A characteristic of the BLDC control is to have only one current at a time in the motor (two phases ON). Consequently, it is not necessary to put a current sensor on each phase of the motor; one sensor placed in the line inverter input makes it possible to control the current of each phase. Moreover, when using this sensor on the ground line, insulated systems are not necessary, and a low-cost resistor can be used. Its value is set such that it activates the integrated overcurrent protection when the maximum current permitted by the power board has been reached.

Each current measurement leads to a new PWM duty cycle loaded at the beginning of a PWM cycle. Note that, during Turn OFF, the shunt resistor does not have this current to sense, regardless of whether the inverter is driven in hard chopping or in soft chopping mode. Figure 5 depicts the shunt current in soft chopping mode and shows that in the Turn OFF operation the decreasing current flows through the M2 freewheeling diode and through the maintained closed M4 (so there is no current observable in the shunt in this chopping mode during Turn OFF). This implies that it is necessary to start a current conversion in the middle of the PWM duty cycle.

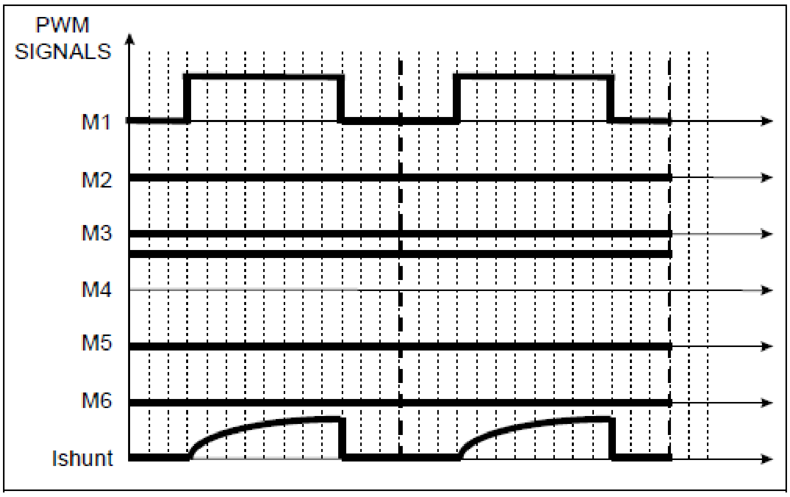


Fig. 5 Shunt Resistor Voltage Drop According to PWM Duty Cycles (Soft Chopping)

In the hard chopping mode during the Turn OFF, neither M1 nor M4 drive current, so that the decreasing phase current flows from ground through the shunt resistor via M2 and M3 freewheeling diodes and back to ground through the capacitor. In this chopping mode, it is possible to see the exponentially decreasing phase current across the shunt as a negative shunt voltage drop appears. Assuming that neither the power board nor the control board support negative voltages, this necessitates that the current be sensed in the middle of the Turn ON.

Achieving a BLDC speed control requires three control layers to be performed. The innermost one is to get the rotor position to correctly commutate the stator flux. Once the rotor position is known, the magnitude of the stator flux has to be generated and controlled. Assuming that the stator flux is proportional to the current flowing in the stator coils, the control of the stator flux magnitude is equivalent to the control of the input current. The outermost control loop is the speed regulation loop.

4.3. Position and Speed Sensing

The motor in this application is equipped with three Hall effect sensors. These sensors are fed by the power electronics board. The sensor outputs are directly wired to the GPIO pins. The Hall effect sensors give three 180° overlapping signals, thus providing the six mandatory commutation points: The rising and falling edges of the sensor output are detected, and the corresponding flags are generated. The system first determines which edge has been detected, then computes the time elapsed since the last detected edge and commutates the supplied phases.

The speed feedback is derived from the position sensor output signals. As mentioned in the previous paragraph, there are six commutation signals per electrical cycle. In other words, between two commutation signals there are 60 electrical degrees. It is possible to get the speed from the computed elapsed time between two captures. Between two commutation signals, the angle variation is constant as the Hall effect sensors are fixed relative to the motor, so speed sensing is reduced to a simple division.

5. System Overview

This document describes the “C” real-time control framework used to demonstrate the trapezoidal control of BLDC motors. The “C” framework is designed to run on DRV92250 controllers on Code Composer Studio.

The functions given in Table 2 are used in the trapezoidal control of BLDC motors using Hall position sensors.

Table 2. Functions Used in Trapezoidal Control of BLDC Motors Using Hall Sensors

Function names	Explanation
Init.c	Initialization of system variables and the analog front-end registers of the DRV92250
Global.h	User interface variables
Global.c	Defines the following functions: *Commutate()* - to commutate the switching based on position feedback *ChargeBootCap()* – Initial charging of bootstrap capacitors *SetPWMDutyCycle()* – Assign the duty cycle to the TIMERB0 register *HighImpedance()* - Function which turn off all the six PWMs *ReadPotiSpeed()* – Read the potentiometer for changing motor speed *ReadVCC()* – Read the DC bus voltage
ISRs.c	Define the TIMERB0 interrupt and PORT interrupt

The DRV92250 is used to generate six PWM signals. The motor is driven by a discrete IGBT-based three-phase inverter with the gate driver UCC27714, by means of BLDC specific PWM technique. Phase voltages, the DC bus voltage, and the DC bus return current are measured and fed to the DRV92250.

Table 3. System Features

Development /Emulation	Code Composer Studio v5.5
Target Controller	DRV92250
PWM Frequency	10-kHz PWM (Default), programmable for higher and lower frequencies
PWM Mode	Asymmetrical with no dead band
Interrupts	CPU Timer B0 – Implements 10-kHz ISR execution rate PORT1 – Over current Interrupt
Peripherals Used	TIMER B0 for motor control PWM (P2.1 –P2.6) ADC7 – Temperature sense ADC8 – DC bus voltage sense ADC9 – Potentiometer voltage sensing for changing speed. ADC11, ADC12, ADC13 – Motor winding voltage sensing (applicable for sensor- less control only) P3.1, P3.3, P3.3- Hall sensor inputs DACB- current limit reference DC-DC controller

The overall system implementing a three-phase sensor-based BLDC control is depicted in Figure 6.

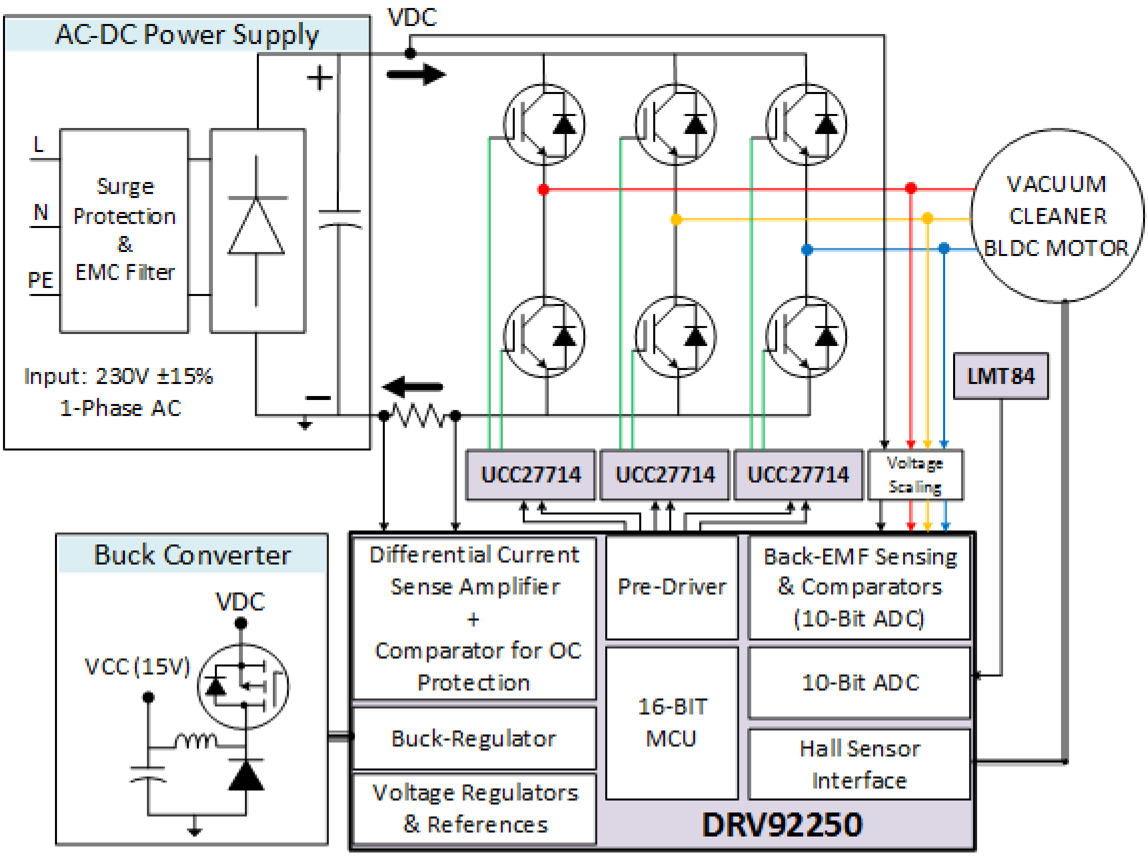
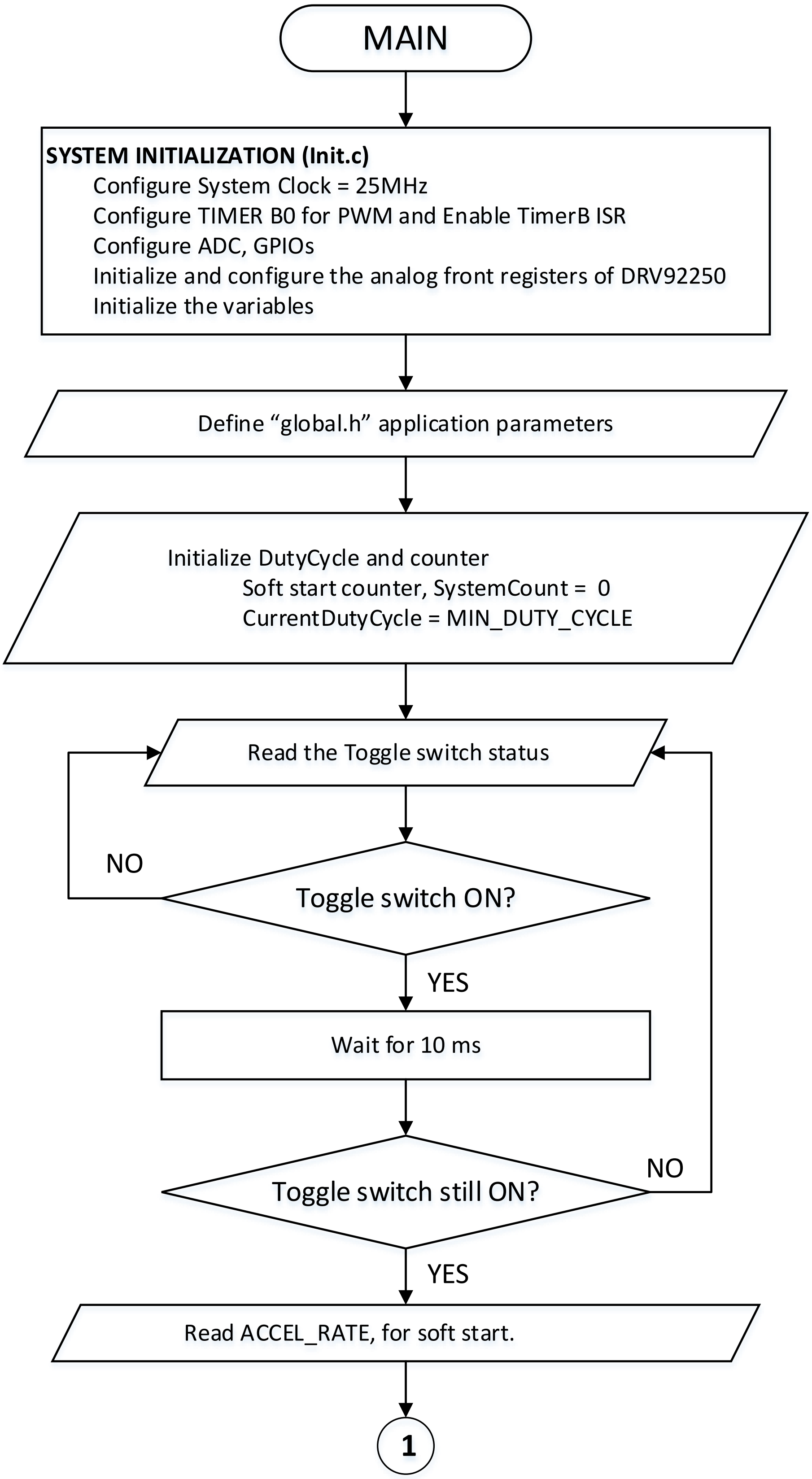


Figure 6. Overall Implementation of BLDC Motor Drive Using DRV92250

6. Software Flowchart

The software flow is described in Figure 7.



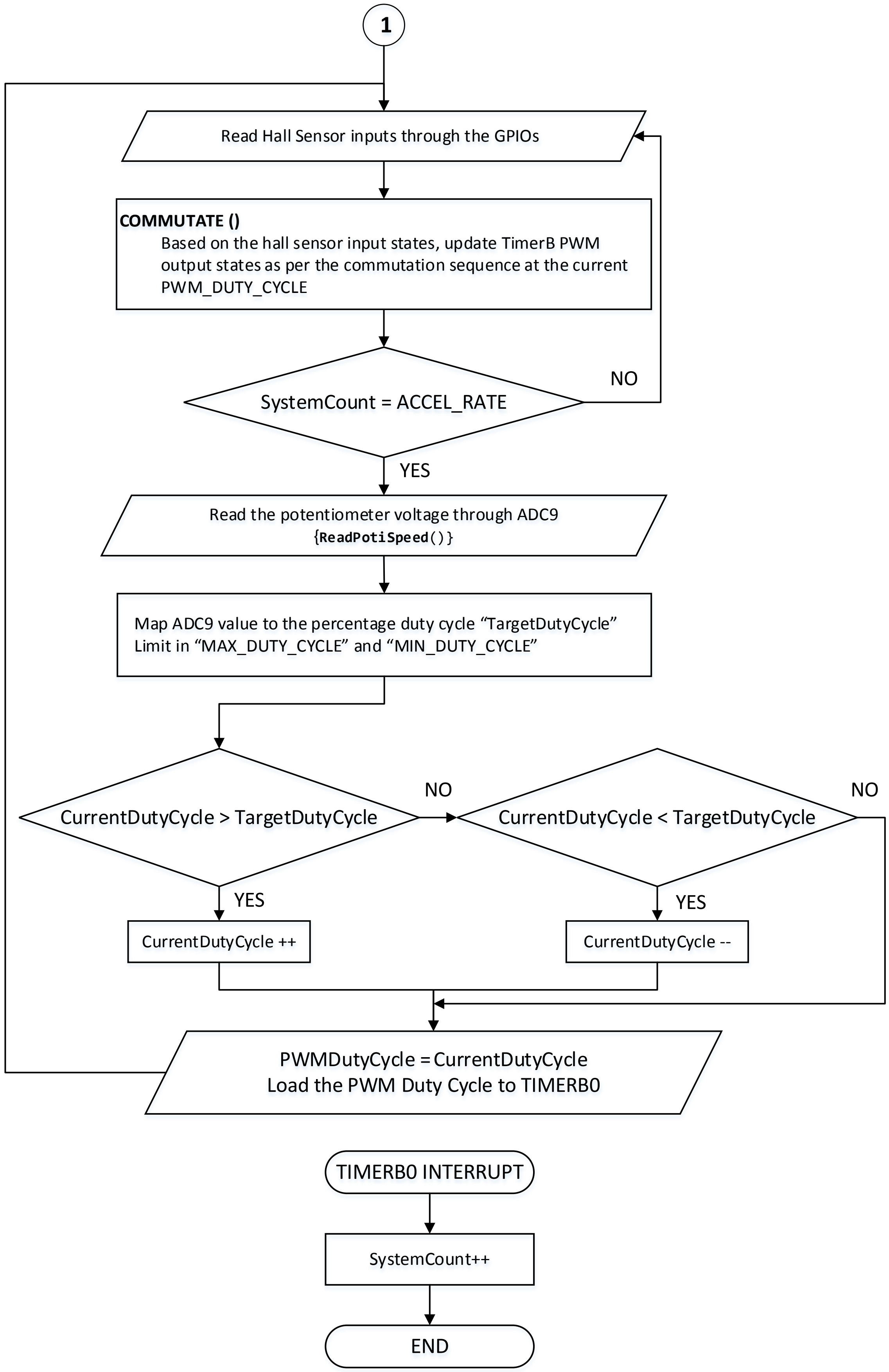


Figure 7. Flowchart of Sensor-Based Control Using DRV92250

7. Instructions for Programming the DRV92250

Gather the following pre-requisites to program the DRV92250.

Table 4. Pre-requisites for Programming DRV92250

Programmer kit	MSP-FET430UIF (MSP430 USB Debugging Interface)
Platform	Code Composer Studio 5.5

The programmer tool has a 14-pin JTAG connector. The programming of DRV92250 is done by means of 2-wire Spy-Bi-Wire. The four-pin connector between the programmer and the DRV92250 can be formed as shown in Figure 8.

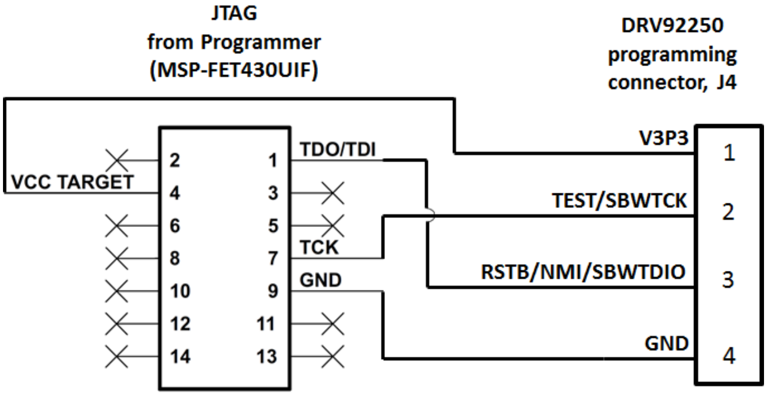


Figure 8. Programming Connector

In the board, J4 is the programming connector. Figure 9 shows the connection of the 14-pin JTAG to the 4-pin connector in the board.

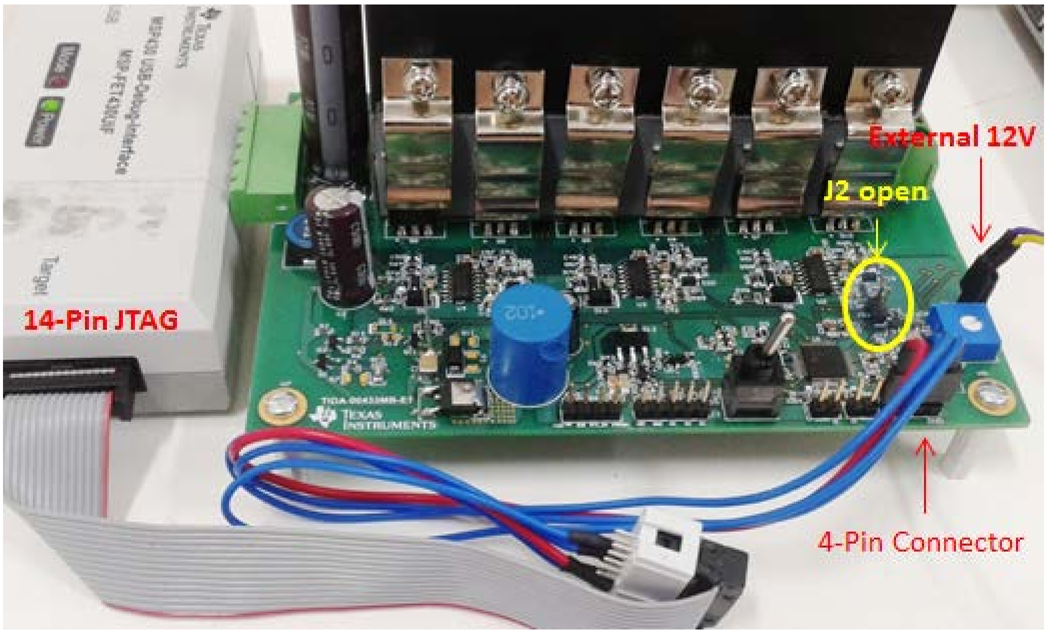


Figure 9. Programming of DRV92250

The header J6 is used as the provision for external 12-V supply for programming the DRV92250. By default, the external 12-V is connected to the board. If the jumper J2 is closed, the onboard 15 V will be connected to the DRV92250. Figure 10 shows the terminal connector J6 for external 12-V programming supply.

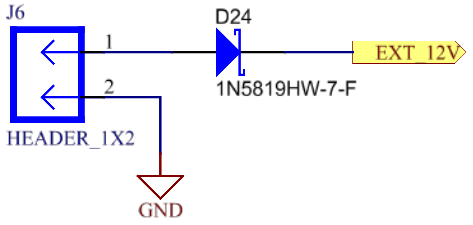


Figure 10. External Power Supply for Programming

During programming, the header J2 should be kept open and apply the external 12 V at J6 with correct polarity as shown in Figure11.

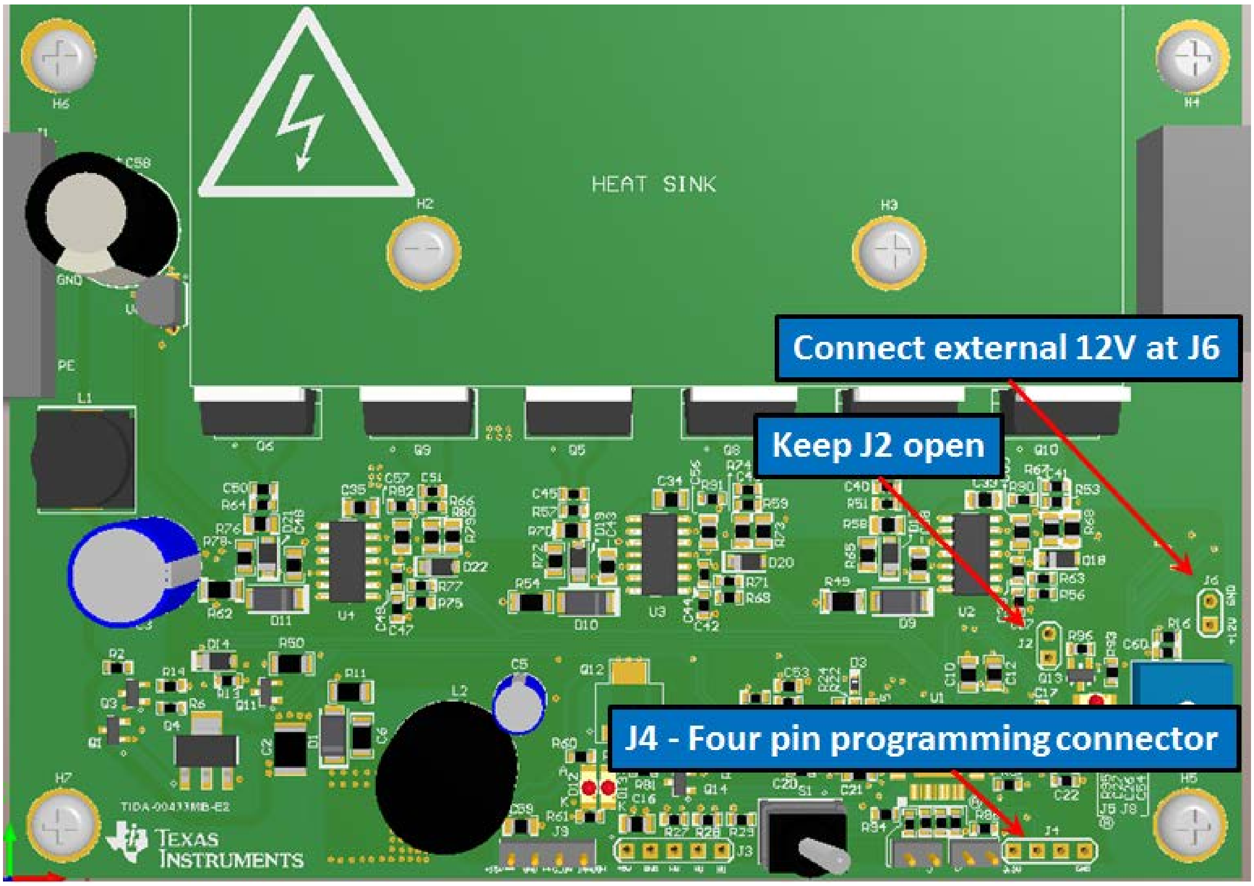


Figure 11. External Power Supply for Programming

Follow these steps to program the board:

1. Switch off the board.
2. Open J2 connection.
3. Give the external 12 V to J6 with the correct polarity, and turn on the 12-V supply.
4. Connect the programmer to the computer.
5. Connect the -pin connector from the programmer to the terminal J4 of the board.
6. Open Code Composer Studio, build, and debug the program to burn the code.

8. System Interfaces and Connectors

8.1. External Hall Sensor Interface

Figure 12 shows the connector J3 used to interface the Hall sensors from the motor to the board. The 5.35 V generated by the DRV92250 is used as the power supply for the Hall sensor. Usually, the Hall sensors have an open drain or open collector configuration. R27, R28, and R29 are used as the pull-up resistors.
R87, R88, and R89 are used as voltage dividers so that the voltage to pins HS1, HS2, and HS3 can be properly scaled. Here the scaling is done to make the logic high voltage to 3.3 V. R30, R31, and R35 along with C18, C19, and C20 form noise filtering at the Hall sensor input.

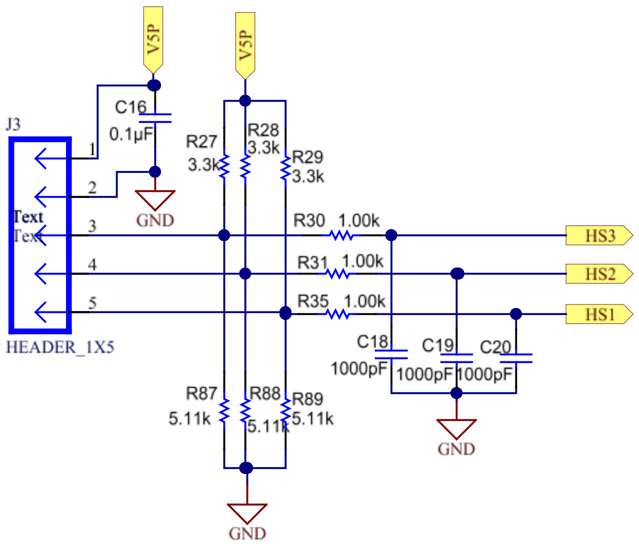


Figure 12. Hall Sensor Connector

Note:. The Hall sensor connection should match with the winding for proper operation of the BLDC motor. The winding phases U, V, and W are named as MU, MV, and MW and the Hall sensors as HU, HV, and HW in the inverter board. The Hall sensors inside the motor are placed such that the connection matches Table 5. To change the direction of rotation, the two-phase winding and the corresponding Hall sensors should be interchanged.

Table 5. Hall Sensor and Motor Winding Connection Matching

Hall sensor signal	Hall sensor	Matching Winding phase	Comments
HS1	HU	MU	Hall sensor HU placed at the start of the U phase winding
HS2	HV	MV	Hall sensor HV placed at the start of the V phase winding
HS3	HW	MW	Hall sensor HW placed at the start of the W phase winding

8.2. Connector Configuration of TIDA-00433MB

1. **Five-pin terminal connector J1:** This is the DC input interface male connector to the inverter board. This is used to interface to the boards TIDA-00433DB/TIDA-00443. The first terminal of this terminal is connected to positive DC bus, the third terminal to negative of DC bus, and the fifth terminal is connected to PE (Earth)
2. **Three-pin connector J7:** These are the motor winding connection terminal. The terminal is marked as MU, MV, and MW used to connect a three-phase BLDC motor winding terminals.
3. **Four-pin connector J9:** This is the interfacing connector to the active PFC board TIDA-00443. It gives the 15-V auxiliary power supply to the PFC board. It has the provision to give PFC_ON signal to the TIDA-00443, which is used to turn on the PFC circuit.
4. **Five-pin connector J3:** This is the Hall sensor interface. It has 5.35 V to be used as the power supply for Hall sensors. It receives the three sensor signals from the Hall sensors mounted in the motor.
5. **Four-pin connector J4:** This is the programming connector for the DRV92250.
6. **Two-pin connector J6:** This is the two-pin connector for giving an external 12 V to the DRV92250 for programming.

8.3. Jumper Configurations of TIDA-00433MB

1. **J2:** To connect the 15-V supply to the gate drivers and to J9. When using the external 12 V for programming, open the jumper J2 to isolate the gate drive sections. For normal working of the board short J2.
2. **J5, J8:** Jumpers used as digital inputs to the DRV92250. When these jumpers are CLOSE, the corresponding digital input to the DRV92250 is logic LOW. When these jumpers are OPEN, the corresponding digital input to the DRV92250 is logic HIGH.

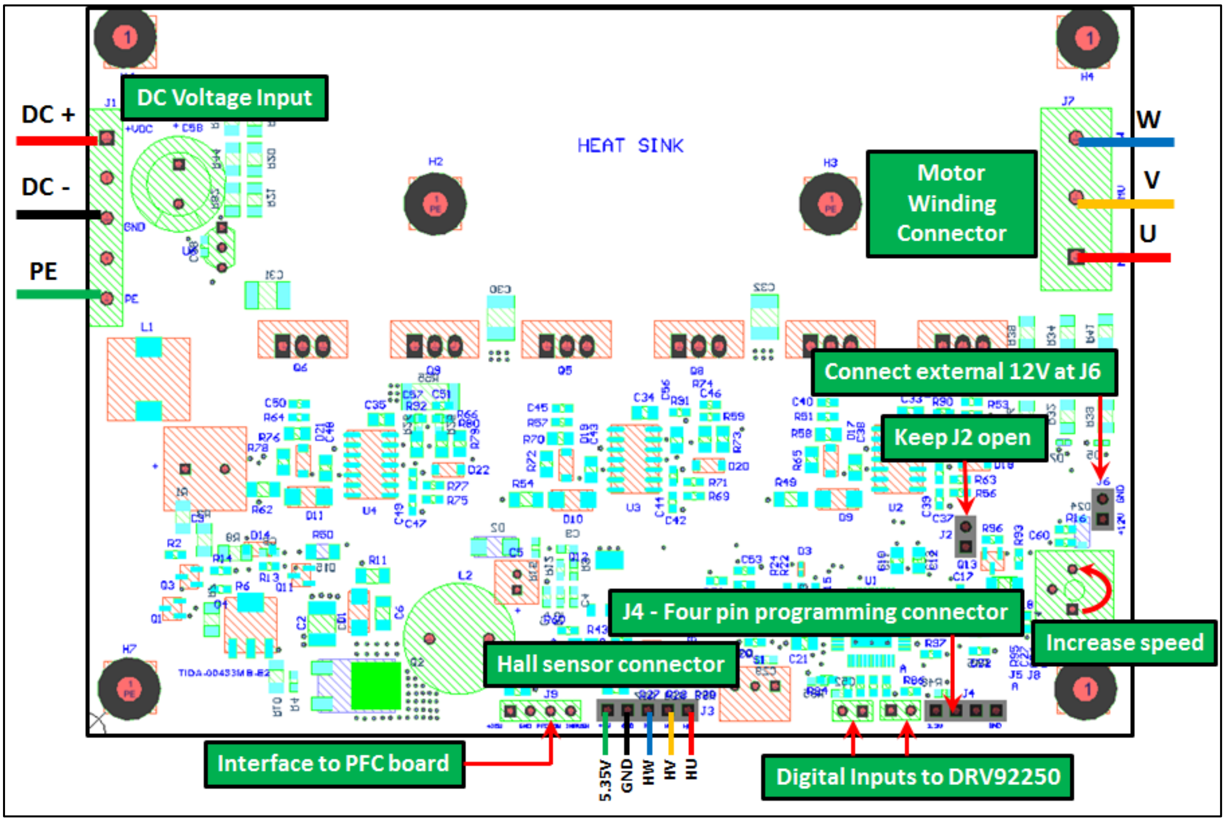


Figure 13. System Interfaces and Connectors

9. Customizing the Reference Code

To modify the sensor-based code, the end user must have Code Composer Studio (CCS) downloaded and the DRV92250 configuration files installed. The following section describes the different user adjustable parameters and how to select an optimized value for a specific application. Start by loading the project into CCS and locating the parameters.

1) Open CCS and load the reference project "***TIDA-00433_sensor_based_BLDC_V1.0***". Note if this project is zipped it must be extracted.
2) Select the file '*global.h*'. At the top of the header file are some parameters that can be optimized. Below snippet is showing these parameters, there is a brief description commented in the '*global.h*' file.

```
// USER DEFINES

#define ACCEL_RATE          2000    //Accleration to full scale value.
                                    //Ramp up time to 100% duty cycle = 100* ACCEL_RATE *PWM period
#define PWM_PERIOD          1200    // PWM Period time = 25 Mhz/((1200*2)-1) ~ 100us, Up Down Mode,10kHz
#define MAX_DUTY_CYCLE      80      // Maximum duty cycle relative to %
#define MIN_DUTY_CYCLE      5       // Minimum duty cycle relative to %
```

ACCEL_RATE

ACCEL_RATE defines how fast the motor will accelerate. For a motor with greater inertia or needs a longer time to accelerate, set this number to a high value such as 2000. Motors that can quickly ramp up can use a smaller ACCEL_RATE to decrease the startup time.

In the application program,

Ramp up time to full scale duty cycle = (Full scale %duty cycle) * ACCEL_RATE * PWM_PERIOD

$$ACCEL\_RATE = \frac{\text{Ramp up time to full scale duty cycle}}{\text{(Full scale \%duty cycle)*PWM\_PERIOD}} \quad (3)$$

For example: To ramp up to 80% duty cycle in 10 seconds, provided the PWM frequency = 10 kHz,

$$ACCEL\_RATE = \frac{10\text{ s}}{80 * 100\mu\text{s}} = 1250$$

PWM_PERIOD

The PWM_PERIOD is used to set the value in TimerB capture/compare register 0. Because TimerB is used as the PWM generator, this value specifies at what timer count the timer generates an interrupt and restarts from zero. TimerB is initialized to operate at 25 Mhz; see the equation below for calculating the PWM frequency. The TIMERB PWM is configured in UP-DOWN Mode.

$$PWM\ Frequency\ (Hz) = \frac{25MHz}{((2*PWM\_PERIOD)-1)} \quad (4)$$

For example, here PWM_PERIOD = 1200

Therefore, $PWM\ Frequency\ (Hz) = \frac{25MHz}{((2*1200)-1)} = 10.42\ kHz$

MAX_DUTY_CYCLE

MAX_DUTY_CYCLE sets the maximum threshold the input duty cycle command is allowed to. Every time the input is read the duty cycle input command is compared to MAX_DUTY_CYCLE and if it exceeds it, the target duty cycle is set to MAX_DUTY_CYCLE. This number is in percentage.

MIN_DUTY_CYCLE

MIN_DUTY_CYCLE sets the minimum duty cycle that can be applied to the motor. This number is in relation to percentage.

9.1. DRV92250 Analog Device Serial Registers

The DRV92250 provides a total of seventeen 8-bit registers (or 136 bits) of programmable register space on the analog device, which is accessible through the serial port. These registers can be set as needed to control operation in different modes. Out of the 17 registers, 13 (address 0x00 to address 0x0D) are available during normal operation while four registers are reserved for test and debug (address 0x0E to address 0x11). All register bits are reset to 0 at power-up. The following example shows the initialization of these registers by writing into them the equivalent hex values:

```
drv922xxSpiWrite(ASSI_ICOMP_WRITE, 0x51);
drv922xxSpiWrite(ASSI_GPIO1IN_WRITE, 0x00);
drv922xxSpiWrite(ASSI_GPIO1OUT_WRITE, 0x70);
drv922xxSpiWrite(ASSI_GPIO1GP_WRITE, 0x80);
drv922xxSpiWrite(ASSI_GPIO2GP_WRITE, 0x00);
drv922xxSpiWrite(ASSI_GPIO3GP_WRITE, 0x00);
drv922xxSpiWrite(ASSI_ADC1_WRITE, 0x00);
drv922xxSpiWrite(ASSI_ADC2_WRITE, 0x00);
drv922xxSpiWrite(ASSI_DACA_WRITE, 0x00);
drv922xxSpiWrite(ASSI_DACB_WRITE, 0x1E);  //DAC for over current limit threshold
drv922xxSpiWrite(ASSI_HALLSNS_WRITE, 0x50);
drv922xxSpiWrite(ASSI_MISC1_WRITE, 0x00);
drv922xxSpiWrite(ASSI_MISC2_WRITE, 0x34);
drv922xxSpiWrite(ASSI_MISC3_WRITE, 0x30);
drv922xxSpiWrite(ASSI_MISC4_WRITE, 0x20);
```

The status of these registers can be read using the read command as shown below.

```
unsigned char test = drv922xxSpiRead(ASSI_MISC1_READ);
```

The command read the status of the register MISC1 into the variable “test”.

For detailed information on the register settings, refer to the DRV92250 datasheet.

9.2. Overcurrent Limit

The DRV92250 has an integrated 8-bit DAC with an output swing from 0 to 1.2 V. The DAC is used to set the current-limit threshold for the overcurrent comparator. The DAC digital input can be set directly by programming the serial port register DACB.

```
drv922xxSpiWrite(ASSI_DACB_WRITE, 0x1E);
```

The following calculation can be done to find out the value to be stored in the “DACB” register.

R_{SENSE} = The sense resistor value

I_{OC} = Required overcurrent limit

For the 8-bit DAC, the maximum hex value is 0xFF (255 Decimal) and the corresponding DAC output voltage is1.2 V.

Therefore, the equivalent decimal value to be stored in the DACB registers:

$$DACB_{dec} = \left(\frac{I_{OC} * R_{SENSE}}{1.2}\right) * 255 \quad (5)$$

Approximate $DACB_{dec}$ to the nearest integer value, convert into the equivalent hex, and write in the DACB register.

For example:

In the reference design, to set the current limit at 7 A, I_{OC} = 7 A, using R_{SENSE} = 20 mΩ

Using the equation 5,

$$DACB_{dec} = \left(\frac{7*0.02}{1.2}\right) * 255 = 29.75$$

The nearest integer value is 30.

The equivalent hex value of 30_D is $0x1E_H$.

Therefore, the value to be stored in DACB register is **0x1E.**

Table 6. Settings of DRV92250 for Current Limit Protection

Bit(s) Name	Number of Bits	Bit Location	Register	R/W	Address	Description
OCFromGPIO	1	7	GPIO2GP	R/W	03	When set to 0, internal comparator is used to detect over-current. When set to 1, external over-current comparator is used. External over-current comparator output should be connected to pin 37 (ADC8/CSAVG/PA1.4). In addition, the state of this multi-function pin must be configured as a digital input (G1p4[WR,RD] = 01).
TRKHSPWM	1	4	GPIO1 OUT	R/W	01	When set to 0, all six pre-drivers are disabled when an over-current event is detected. When set to a 1, only the low-side pre-drivers are disabled if HSDPWM = 0 or only the high-side pre-drivers are disabled if HSDPWM = 1.
HSDPWM	1	4	Icomp/ PreDrv	R/W	09	Set to 1 when high-side drivers get PWM signal from the MCU. Set to 0, when the low-side drivers get PWM signal. In addition, if control bit TRKHSPWM = 1, only high-side pre-drivers get disabled when an OC is detected if HSDPWM = 1. If TRKHSPWM = 1 but HSDPWM = 0, only the low-side pre-drivers get disabled when an OC is detected.
PreDrvHLD	1	6	GPIO2GP	R/W	03	When set to a 0, pre-driver circuits automatically resume normal operation following an over-current event. When set to a 1, an over-current event disables the pre-drivers. Normal operation can be restored by resetting PreDrvHLD to 0.
CURLIM[1:0]	2	[2:1]	Misc1	R/W	0B	Control bits used to set leading edge blanking interval for over-current detection. The four settings are 800 nsec, 400 nsec, 200 nsec and 0.
IcompHys[1:0]	2	[1:0]	Icomp/ PreDrv	R/W	09	Hysteresis control for over-current comparator. 00 is 20 mV, 01 is 0 mV, 10 is 40 mV and 11 is 80 mV.
DACB [7:0]	8	[7:0]	DACB	R/W	06	Set the reference comparison voltage for the hardware over-current comparator. The range is 0 to 1.2 V.

Table 7. Settings of DRV92250 Reference Design Program for current limit protection

Register Bit	Binary setting	Remarks
OCFromGPIO	0	On-chip voltage comparator for OC detection
TRKHSPWM	1	Number of pre-drivers disabled depends on HSDPWM
HSDPWM	1	Only the three high-side pre-drivers are disabled
PreDrvHLD	0	pre-driver circuits automatically resume normal operation following an over-current event
CURLIM [1:0]	00	Leading-edge current limit blanking interval = 800 ns
IcompHys [1:0]	01	0-mV hysteresis control for overcurrent comparator
DACB [7:0]	0x0D	8-bit reference for the current limit

9.3. DC-DC Controller

The DRV92250 integrates a buck converter controller. An adjustable DC-DC buck converter can be implemented using external power transistors or FETs and passive components. The DC/DC buck converter controller on the DRV92250 has special protection features designed in to prevent any damage to external components in case of a floating VCC pin.

Table 8. Settings of DRV92250 DC-DC Controller

Bit(s) Name	Number of Bits	Bit Location	Register	R/W	Address	Description
DCDeGlitch	1	7	GPIO1GP	R/W	02	When set to a 1, any pulses shorter than 1 µsec into the DC/DC switcher are stretched out to be at least 1µsec wide. This sets the floor on the minimum duty-cycle for the DC/DC switcher.
DcDcWidth[1:0]	2	[3:2]	comp/ PreDrv	R/W	09	Maximum duty-cycle control for the DC/DC switcher. The four settings are 50%, 30%, 17% and no limit.
HYSDCDC[1:0]	2	[6:5]	Icomp/ PreDrv	R/W	09	Hysteresis control for DC/DC converter. The four settings are 10 mV, 20 mV, 30 mV, and 40 mV
FLYBACKENBL	1	2	Misc2		OC	When set to a 1, the pin SWDR is modulated during the high state using a 62.5-kHz clock.

10. Running the Project in CCS

To run this project in CCS:

1. Install CCS using the *CCS Installation and install Configuration for DRV9x*.
2. Following the above guide, import this project "***TIDA-00433_sensor_based_BLDC_V1.0***".
3. Read through section 9 to tune the control for the specific motor.
4. Power up the board with external supply as described in section 7 and connect the MSP-FET430UIF to the computer and the programming pins of DRV92250.
5. Build and debug the modified project to download the code to the DRV92250.

11. Power Up Sequence

1. Program the DRV92250.
2. Make sure that all the connectors and jumpers configured properly as explained in Section 8.
3. Make sure that the jumper J2 is closed.
4. Connect the AC–DC converter board (TIDA-004533DB) to the inverter board (TIDA00-433MB) through the interface J1 of the inverter board.
5. Disconnect the BLDC motor terminals.
6. Increase the input AC voltage slowly so that the DC bus voltage is around 100 V (70-V AC).
7. Check all the power supplies 15 V, 5.35 V, and 3.3 V and ensure that all are proper.
8. Increase the AC input voltage so that DC bus voltage is 390V and check 15 V, 5.35 V, and 3.3 V.
9. Switch off the AC input and wait for 1 minute so that the DC bus capacitor will discharge completely.
10. Connect the BLDC motor terminals to J7 of TIDA-00433MB, Hall sensors to J3.
11. Increase the AC voltage so that the DC bus voltage is around 120 V.
12. Move the toggle switch ‘S1’ towards right to start the program and hence to start the motor. If ‘S1’ is already in right position, the program execution starts on power up.
13. If the motor is getting stuck or did not move, please check the Hall sensor combination and interchange the hall sensor connection if required to match the combination mentioned in Table 8.1.
14. After getting a satisfactory performance, go to full voltage of 230-V AC and then to 265-V AC.
15. Adjust the motor speed using the potentiometer R17 if necessary. Rotating the potentiometer clockwise reduces the speed and counter-clockwise increases the speed.

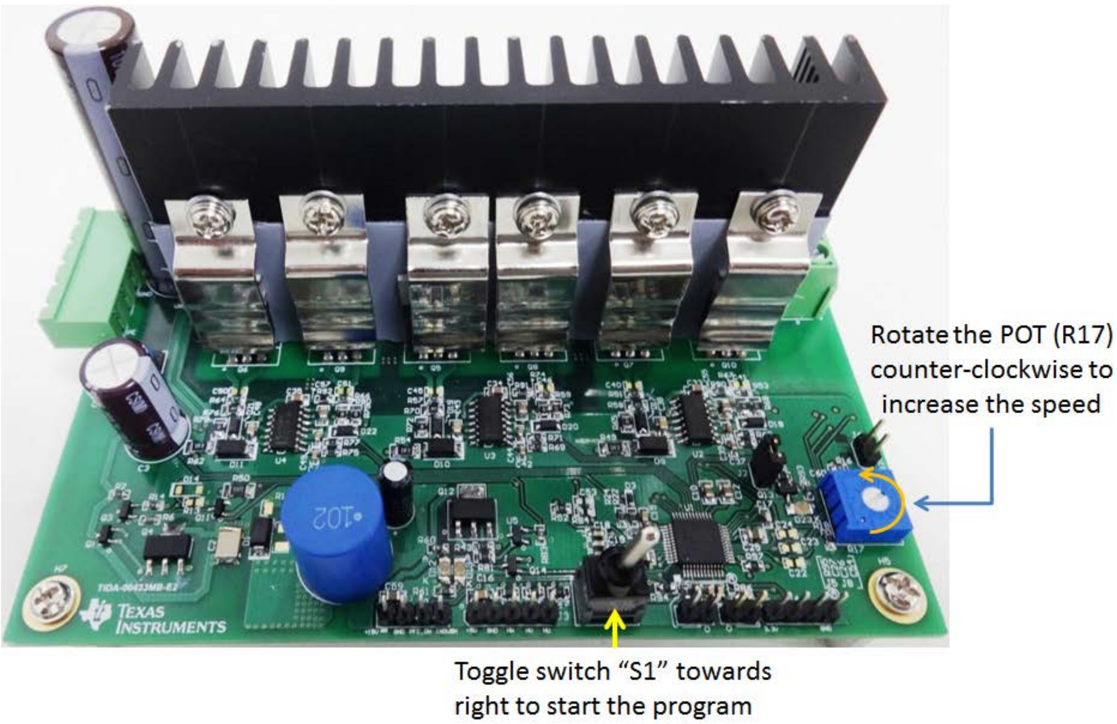


Figure13. Program Start Up and Speed Variation Control

CAUTION: The inverter bus capacitors remain charged for a long time after the power line supply is switched off or disconnected. Proceed with caution!

References

1. Texas Instruments Datasheet, *Highly Integrated High Voltage Brushless DC (BLDC) Motor Controller,* DRV92250
2. Texas Instruments Application Report, *Sensorless Trapezoidal Control of BLDC Motors,* (SPRABQ7)

About the Author

Manu Balakrishnan is a systems engineer at Texas Instruments where he is responsible for developing subsystem design solutions for the Industrial Motor Drive segment. Manu brings to this role his experience in power electronics, analog and mixed signal designs. He has system level product design experience in permanent magnet motor drives. Manu earned his bachelor of technology in electrical and electronics engineering from the University of Kerala and his master of technology in power electronics from National Institute of Technology Calicut, India.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265