Application Note

Sensorless Brushless DC Motor Control with Z8 Encore! MC™ Microcontrollers

AN022606-0911

Abstract

This application note discusses the closed-loop control of a three-phase brushless direct current (BLDC) motor using the Z8 Encore! MC™ Family of Microcontrollers (MCUs). The Z8 Encore! MC product family is designed specifically for motor control applications and features an on-chip integrated array of application-specific analog and digital modules. The result is fast and precise fault control, high system efficiency, on-the-fly speed/torque and direction control, and ease of firmware development for customized applications.

This document further discusses how to implement a sensorless feedback control system using a phase-locked loop and Back EMF sensing. Test results are based upon the Zilog BLDC Motor Control Development Kit (Z8FMC160100KITG) which includes a Motor Control Drive System module with a 32-pin Z8FMC16100 MCU, a three-phase motor control application board and a 3-phase 24 VDC, 30W, 3200 RPM BLDC motor with internal Hall sensors.

Note: The source code files associated with this application note, AN0226-SC01 and AN0226-SC02, were tested with version 5.0.0 of ZDS II for Z8 Encore! MCUs. Subsequent releases of ZDS II may require you to modify the code supplied with this application note.

The sample project included in ZDS II v5.0.0 and the firmware in the Rev D (or earlier) version of the Z8FMC160100KITG Development Kit were preprogrammed with AN0226-SC01. The source code files contained in AN0226-SC02 are enhanced versions of AN0226-SC01 that allow users to easily change parameters to accommodate differing motor types.

Revision History

Each instance in the following table reflects a change to this document from its previous version. For more details, refer to the corresponding pages or appropriate links provided in the table.

Date	Revision Level	Description	Page Number
Mar 2011	05	AN0226-SC02 source code added to encompass multiple motor types; correction to TimerPrescale data in PLL flow, Figure 4.	1, 9
Dec 2010	04	Minor language changes to be consistent to AN0311.	All
Dec 2010	03	Changes to UART Control data.	17–18

Date	Revision Level	Description	Page Number
Aug 2010	02	Updated to include IXYS/Zilog logo.	All
Sep 2005	01	Original issue.	All

Features

The key features of this implementation include:

- Smooth S-curve motor startup with reduced starting current
- Sensorless (Back EMF) control using Phase Locked Loop feedback
- Microcontroller-based overcurrent protection
- Selectable speed or torque setting
- Selectable speed or torque control
- Selectable control of motor direction
- LED for max speed indication
- LED for motoring running indication
- LED for fault indication

Discussion

The use of BLDC motors has steadily increased over the last several years as the cost of these motors and the technology to control them has decreased and their benefits over other motor types has become more important. Variable-speed motor applications in industries such as white goods, automotive, aerospace, medical and industrial automation are increasingly using BLDC motors over other types of motors, such as brushed DC and AC induction.

The construction of a BLDC motor gives it several advantages when compared to other electric motors. First, because a BLDC motor uses electronic commutation, it has a longer life when compared with brushed DC motors and, because the brushes on the motor do not require cleaning and replacement, it requires less maintenance. A BLDC motor also runs much more quietly, both electrically and audibly, because the motor neither exhibits brush arcing nor the mechanical commutation of other types of motors. A BLDC motor generally produces a higher output per frame size because the windings are connected to the stator; the heat generated from running can be transferred directly to the motor housing to allow cooler operation. Finally, a BLDC motor will result in much lower electrical and friction losses because it is not required to transfer power via brushes. These losses are most prevalent at lower loads. The prevailing data shows that a standard BLDC motor operates at a 5-10% better efficiency than typical AC induction motors and 8-12% better efficiency than brushed DC motors.

Multiple control methods exist for BLDC motors; the selection is based on the requirements of the application. The most cost-effective is sensorless control. When using this method, the Back EMF of the un-energized coil is used to determine the rotor position. However, when starting the motor, no Back EMF is generated when the rotor is not in motion; therefore the motor can move in the wrong direction for a small period of time until the rotor position is determined. Sensorless control can be implemented with a few discrete components and a small amount of firmware which make it very attractive from a cost standpoint when small initial movement of the motor does not present a safety issue.

Theory of Operation

In a brushless DC motor, the rotor uses permanent magnets, while the stator windings are similar to those in AC induction motors. For a detailed discussion of BLDC motor fundamentals, as well as closed-loop control using sensorless techniques, refer to the Motor Control Electronics Handbook by Richard Valentine, McGraw-Hill, NY; 1998.

In a brushed DC motor, commutation is controlled by brush position. In a BLDC motor, however, commutation is controlled by the supporting circuitry. The rotor's position must therefore be fed back to the supporting circuitry to enable proper commutation.

Two different techniques can be used to determine rotor position:

Hall Sensor-Based Commutation. When employing a Hall sensor technique, three Hall sensors are placed inside the motor, spaced 120 degrees apart. Each Hall sensor provides either a High or Low output based on the polarity of the magnetic pole close to it. Rotor position is determined by analyzing the outputs of all three Hall sensors. Based on the output from Hall sensors, the voltages to the motor's three phases are switched.

The advantage of Hall sensor-based commutation is that the control algorithm is simple and easy to understand. Hall sensor-based commutation can also be used to run a motor at very low speeds. The disadvantages are that its implementation requires both separate Hall sensors inside the motor housing and additional hardware for the sensor interface.

Sensorless Commutation. With the sensorless commutation technique, the Back EMF induced in the idle phase is used to determine the moment of commutation. When the induced idle-phase Back EMF equals one-half of the DC bus voltage, commutation is complete.

The advantage of sensorless commutation is that it makes the hardware design simpler. No sensors or associated interface circuitry are required. The disadvantages are that it requires a relatively complex control algorithm and, when the magnitude of induced Back EMF is low, it does not support low motor speeds.

Furthermore, two voltage application techniques can be applied based on the configuration of the supply-to-motor windings:

Sinusoidal. Sinusoidal voltage is applied to the three-phase winding. Sinusoidal voltage provides a smooth motor rotation and fewer ripples.

Trapezoidal. Here, DC is applied to two phases at a time; the third phase remains idle. Trapezoidal voltage is simpler to implement and is less complex.

In this application, sensorless control with a trapezoidal waveform is implemented. This implementation is very common in small BLDC motors used in many white goods and other consumer-based products.

A block diagram of the BLDC motor control system, based on the Z8FMC16100 MCU, is shown in Figure 1. In a 3-phase commutation arrangement, at any given instance, only two phases are energized. The Back EMF voltage is subsequently generated in the unenergized phase winding, and the zero crossing of this induced voltage is detected for synchronization of the subsequent closed-loop control events. As discussed later, the innovative Time Stamp feature of the Z8FMC16100 MCU provides for robust, efficient implementation of this critical sensing function without the requirement for an additional comparator.

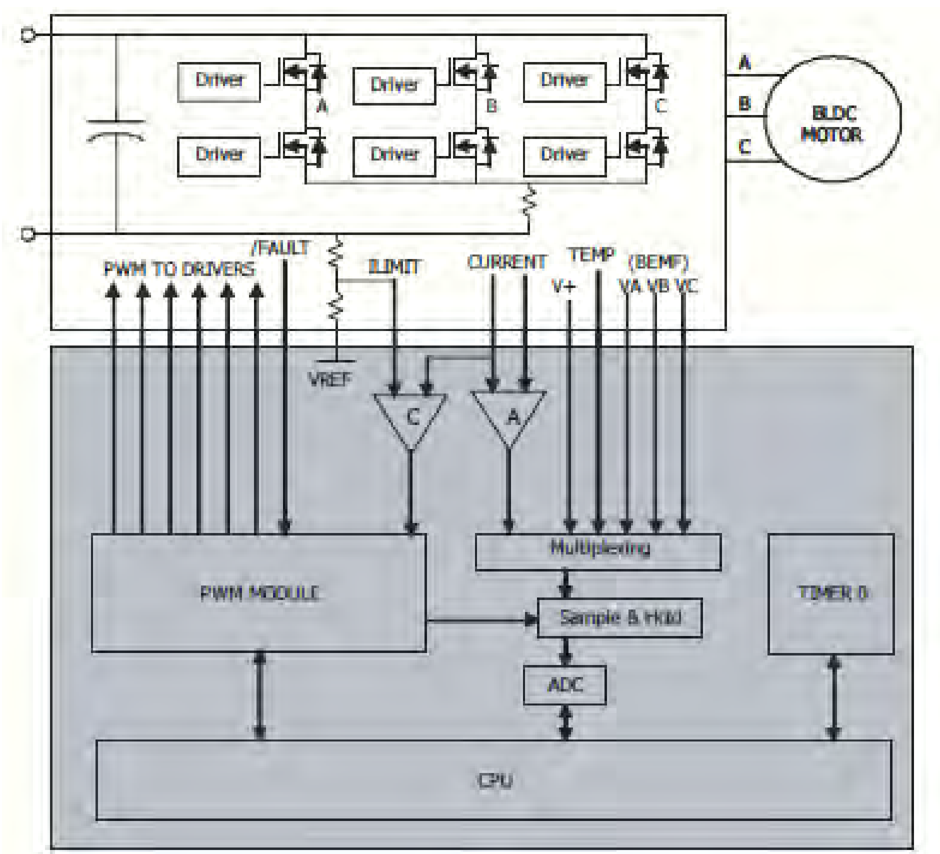

Figure 1. A Three-Phase Motor Control System

The algorithm for Back EMF sensing is based on the implementation of a Phase Locked Loop (PLL). A PLL is especially advantageous during startup, resulting in a gradual increase in the motor speed as well as a nearly instantaneous reversal of direction of rotation on command, as outlined below.

In the conventional approach, during the startup sequence, power is applied to the windings to place the rotor in a known starting position, followed by commutation and start of Back EMF sensing and control. In contrast, the PLL-based approach implemented in this implementation makes it possible to lock to the Back EMF signal from the very onset of the start-up phase without the need for the initial placement of the rotor in a specific position. Moreover, this approach significantly reduces the erratic movement of the motor during startup or reversal of direction.

Following the start-up phase, during the normal operation phase, torque/current mode control is achieved via sensing of the voltage generated across a sense resistor in the motor drive circuit. This voltage is routed to the on-chip integrated ADC, after which data processing by the CPU based on a predefined computational algorithm results in the regulation of the Pulse Width Modulation (PWM) commutation signal period.

Another key feature of the Z8FMC16100 MCU is the direct coupling of the on-chip integrated comparator to the PWM module to enable fast, cycle-by-cycle shutdown during an overcurrent fault event.

In conjunction with the integrated on-chip hardware blocks, the 3-phase BLDC motor control software developed in this implementation allows for ease of programming to achieve the desired closed-loop control characteristics. The routines that enable the sensing of the motor's Back EMF and current are all interrupt-driven. It is critical that the highest interrupt priority is assigned to the Back EMF sensing event, as this is a critical step for subsequent synchronization of the commutation events. In this case, Timer 0 is used for the Time Stamp function as well as for updating the commutation period, if necessary.

Hardware Architecture

The functional block diagram of the Z8 Encore! MC Sensorless Brushless Motor Controller shown in Figure 1 (see the previous section) is divided into a control section (highlighted in grey) and a power conversion section. In the control section, the Z8FMC16100 MCU is operating with an external 20MHz crystal.

Table 1 describes the pin functions of the Z8FMC16100 MCU and their associated use in this design.

Table 1. Z8FMC16100 MCU Pin Functions

Pin #	Pin Description	Function	In/Out/PWR	Application Use
1	PB2/ANA2/TOIN2	ANA2	Input	Phase C BEMF
2	PB1/ANA1/TOIN1	ANA1	Input	Phase B BEMF
3	PB0/ANA0/TOIN0	ANA0	Input	Phase A BEMF
4	AVVD	AVVD	PWR	3.3V Supply
5	AVVS	AVVS	PWR	Ground
6	VREF	VREF	PWR	Voltage Reference
7	PA0/OPINN	OPINN	Input	Current Sense
8	PA1/OPINP/CINN	OPINP	Input	Current Sense
9	PA2/CINP	CINP	Input	Current Sense
10	PA7/FAULT1/T0OUT…	PA7	Output	Fault(Red) LED
11	RESET/FAULT0	RESET	Input	STOP/RESET

Table 1. Z8FMC16100 MCU Pin Functions (Continued)

Pin #	Pin Description	Function	In/Out/PWR	Application Use
12	DBG	DBG	Input/Output	DEBUG
13	PC0/T0OUT	PC0	Input	Direction
14	PWML2	PWML2	Output	Phase C Gate Low
15	PWMH2	PWMH2	Output	Phase C Gate High
16	PWML1	PWML1	Output	Phase B Gate Low
17	PWMH1	PWMH1	Output	Phase B Gate High
18	PWML0	PWML0	Output	Phase A Gate Low
19	PWMH0	PWMH0	Output	Phase A Gate High
20	VSS	VSS	PWR	Ground
21	XOUT	XOUT	Output	Clock
22	XIN	XIN	Input	Clock
23	VDD	VDD	PWR	3.3V Supply
24	PA3/TXDE/SCK/SCL	PA3	Output	Power(Yellow) LED
25	PA4/RXD/MISO	RXD	Input	UART Receive
26	PA5/TXD/MOSI	TXD	Output	UART Transmit
27	PA6/CTS/SS/SDA	CTS	Input	UART Clear
28	PB7/ANA7	PB7	Output	Run(Green) LED
29	PB6/ANA6	ANA6	Input	Speed
30	PB5/ANA5	ANA5	Output	ISR Test Pin
31	PB4/ANA4/CINN	ANA4	Input	DC Bus Voltage
32	PB3/ANA3/OPOUT	ANA3	Input	Current Sense

The power conversion section contains the DC bus, gate drivers, MOSFETs, power supply, Back EMF dividers and temperature sensor. The MOSFETs used in this design are IXYS high-efficiency trench gate power devices. To control the small 30W BLDC motor, IXYS part number IXTP64N055T was selected. However, this design is scalable to meet the needs of the majority of 3-phase BLDC motors, from 1 watt to 5kilowatts. To support larger motors, the major design changes are in the power conversion section, which includes the fuse and the MOSFETs. The IXYS family of power MOSFETs, which includes both modules and discrete devices, is available in a wide variety of packages to meet the specific mechanical requirements of the application. They are also available in a wide range of power ratings and can support current in excess of 500A.

Software Architecture

The Z8 Encore! MC™ family of Microcontrollers features up to 16 KB of Flash memory and is based on Zilog's advanced eZ8 8-bit CPU core, which provides closed-loop control of single- and multiphase variable-speed motors. Target applications are major appliances, HVAC, industrial automation and consumer electronics. In each of Zilog's Z8 Encore! MC products, the novel device architecture allows for the realization of a number of enhanced control features including a Time Stamp for speed control, an integrated operational amplifier, and fault response.

Time Stamp for Speed Control

Most microcontrollers use at least one dedicated comparator to detect the zero crossing of the input AC voltage signal so that their output-driving pulses can be synchronized and adjusted to properly regulate motor speed. An alternative approach based on Zilog's motor control MCU eliminates the need for this comparator by instead employing an ADC in conjunction with a timer. In this case, the ADC samples the AC line voltage, with the timer running in the background.

After the ADC samples the line voltage zero crossing, the timer count is read and the result is written to a register. As a result, the timers for the output PWM pulses are cued to efficiently regulate the speed of the motor. The Time Stamp approach results in a very simple and cost-effective solution for the smooth operation of the motor in a steady state.

Integrated Operational Amplifier

Motor controllers almost invariably monitor motor speed by sensing current through the windings, using sensor and sensorless techniques in conjunction with the ADC. Ordinarily, sampling instances by the ADC are synchronized by the MCU.

In this process, an operational amplifier is used to convert the current signal to a voltage signal, respectively. The ADC in turn samples the voltage signal and outputs the result to the processor. The processor will then synthesize the PWM outputs to control motor speed.

In the case of the Z8 Encore! MC Family of Microcontrollers, an on-chip integrated operational amplifier eliminates the need for an external component, hence reducing overall system cost.

Fault Response

Overcurrent faults can result from many different causes and are sometimes destructive. Shorted motor windings, shorted motor leads, problems in mechanical drives and linkages, a stuck rotor or changes in the load, breakdowns or misfiring of power devices and a number of other problems can arise - some of them permanent, some merely temporary. Whatever may be the origin of an overcurrent condition, motor rotation must be halted. In this scenario, fast response time is a key criterion for the design of the fault protection system. However, rather than triggering a hard shutdown of the entire system when a fault is detected, it is better to disable the motor drive outputs on a cycle-by-cycle basis, with normal operation resuming once the fault condition is no longer detected. In this case, if the overcurrent condition persists, a hard shutdown then ensues.

Motor control microcontrollers typically incorporate input elements (such as a comparator) for sensing overcurrent conditions. In many cases, the current signal is routed to the ADC. This approach has a major drawback due to the excess time associated with data processing before the outcome can disable the PWM. The resulting data processing latency could in turn delay system shutdown beyond the next switching cycle, and catastrophic damage could result.

In the Z8 Encore! MC Family of Microcontrollers, to avoid a processing delay inherent with an ADC, an overcurrent comparator is directly coupled to the PWM module, thereby guaranteeing that the shutdown can truly occur in a cycle-by-cycle mode. This approach not only improves the controller's fault response characteristics, but also circumvents a vulnerability that is inherent with the conventional approach. Namely, if the MCU's clock were to stop functioning, there would be no risk of executing a shutdown in response to an overcurrent fault as there could be if an ADC is involved.

All of the algorithms have been developed in the C programming language using the Zilog ZDS II Integrated Development Environment for the Z8 Encore!® family of products. Figure 2 shows the main control loop.

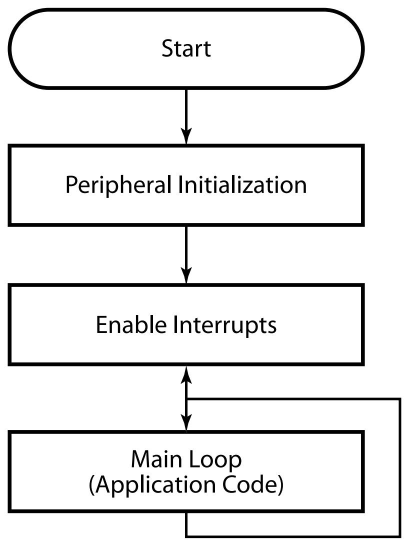

Figure 2. Initialization and Application Code Space Flow

This implementation provides precise control of the motor while leaving sufficient resources for additional application code. Even when using a very small and cost-effective 8-bit MCU, an additional 13 KB of Flash and 420 bytes of RAM are available for additional user application code.

Back EMF sensing of the Phase Lock Loop is unique to this implementation. The details of the algorithm are described in the following figures and tables; the Back EMF sensing loop is shown in Figure 3.

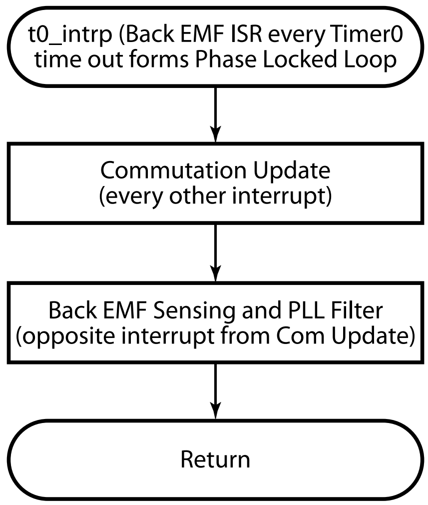

Figure 3. Back EMF Sensing Phase Locked Loop Flow

The Phase Locked Loop Back EMF algorithm, implemented to provide a smooth start-up of the motor, is shown in Figures 4 and 5. Additional details about the specific formulas in these figures are shown in Table 2; a description of these calculations follows.

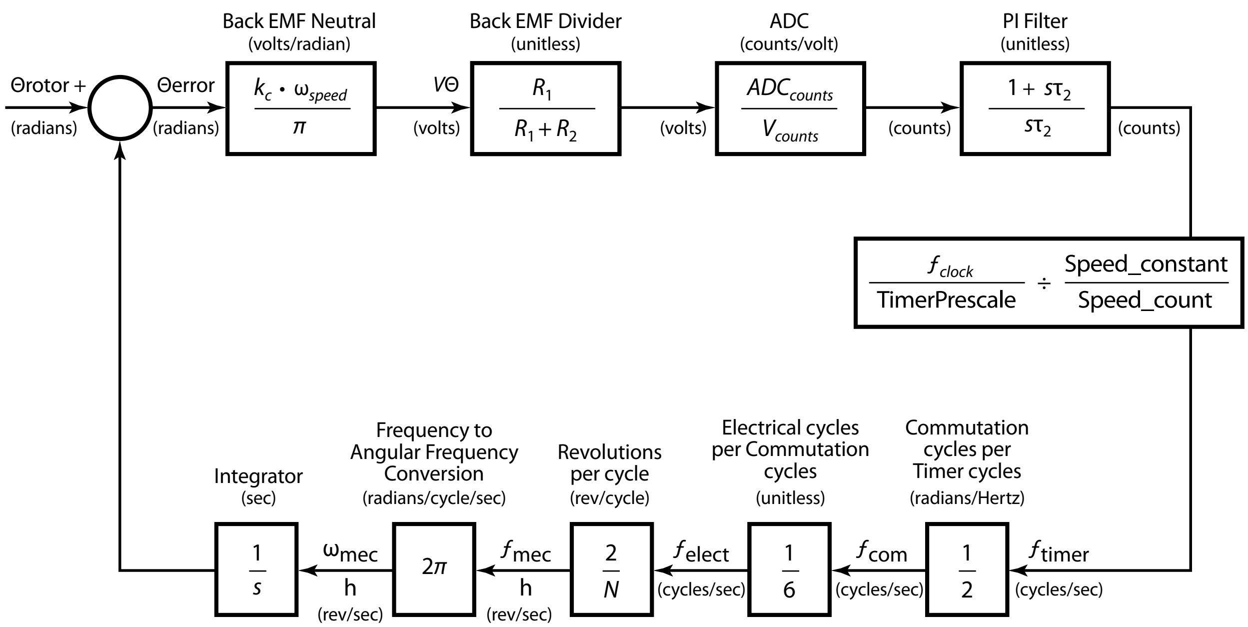

Figure 4. Back EMF Sensing Phase Locked Loop Algorithm

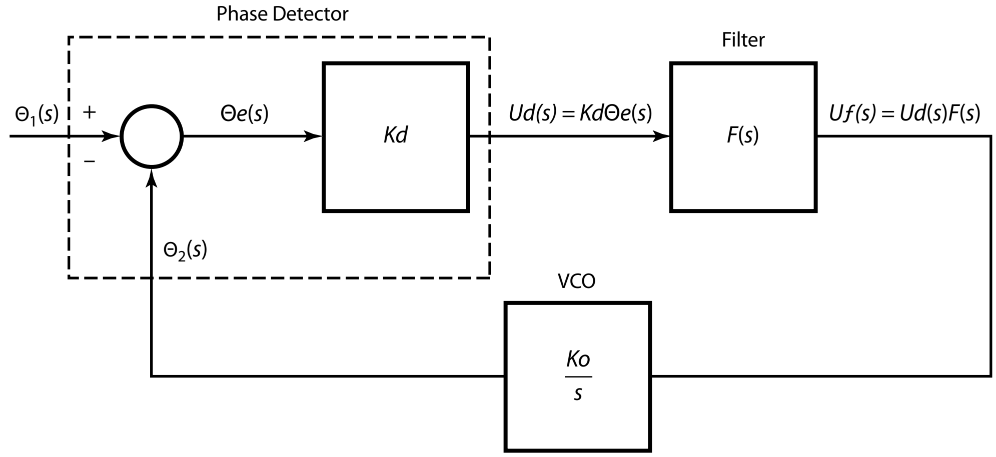

Figure 5. Proportional Integral (PI) Filter Representation for Back EMF Sensing

Table 2. Back EMF Sensing Phase Locked Loop

$D = \frac{R_1}{R_1 + R_2}$	Divider Ratio
$K_d = \frac{K_e \cdot \omega speed \cdot D}{2\pi}$	(volts/rad) ωspeed = current speed of motor
$F(s) = \frac{1 + s\tau_2}{s\tau 1}$	τ_1 = numeric constant τ_2 = numeric constant
$K_o = \frac{ADCcounts \cdot fclock \cdot 2 \cdot 2\pi}{Vref \cdot Prescaler \cdot 6 \cdot N \cdot K_{speed}}$	(rad/sec/volt)
$K_{speed} = \frac{(2 \cdot 2\pi \cdot fclock \cdot speed\_count\_max)}{(2 \cdot 6 \cdot N \cdot \omega_{max} \cdot Prescaler)}$	Speed_count_max = counts at max speed ω max = Maximum motor speed (rad/sec)
$K_o = \frac{ADCcounts \cdot \omega_{max}}{Vref \cdot speed\_count\_max}$	(rad/sec/volt)
$\omega n = \sqrt{Kd \cdot Ko \cdot \tau_1} = \omega_{max}$	Natural frequency
$\zeta = \frac{\omega n \cdot \tau_2}{2} = 0.707$	Damping factor
$H(s) = \frac{\theta_2(s)}{\theta_1(s)} = \frac{2 \cdot s \cdot \zeta \cdot \omega_n + \omega_n^2}{s^2 + 2 \cdot s \cdot \zeta \cdot \omega_n + \omega_n^2}$ $A_{ol}(s) = Kd \cdot F(s) \cdot \frac{Ko}{s} = Kd \cdot Ko \cdot \frac{1 + s\tau_2}{s^2\tau_1}$	Closed Loop Transfer Function
$A_{ol}(s) = \frac{s \cdot Kd \cdot Ko \cdot \tau_2 + Kd + Ko}{s^2\tau_1}$	Open Loop Gain

We begin with the transfer function of the Proportional Integral (PI) Filter in the s-plane:

$$F(s) = \frac{Y(s)}{R(s)} = \frac{1 + s\tau_2}{s\tau_1}$$

Next, by using the bilinear transform identity:

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

where T = the sampling period, yields the following equation.

$$F(z) = \frac{Y(z)}{R(z)} = \frac{1 + \left(\frac{2z-1}{Tz+1}\right)\tau_2}{\left(\frac{2z-1}{Tz+1}\right)\tau_1}$$

When multiplying by:

$$Tz - T$$

the calculations that follow are:

$$F(z) = \frac{Y(z)}{R(z)} = \frac{Tz + T + (2\tau_2)(z - 2\tau_2)}{(2\tau_1)z - 2\tau_1}$$

$$zY(z) - Y(z) = \left(\frac{T + 2\tau_2}{2\tau_1}\right)zR(z) + \left(\frac{T - 2\tau_2}{2\tau_1}\right)R(z)$$

$$zY(z) - Y(z) = a_0 zR(z) + a_1 R(z)$$

where:

$$a_0 = \frac{T + 2\tau_2}{2\tau_1} \qquad\qquad a_1 = \frac{T - 2\tau_2}{2\tau_1}$$

and:

$$Y(z) = z^{-1}Y(z) + a_0 R(z) + a_1 z^{-1} R(z)$$

Collecting terms and dividing by *z* yields the following result:

$$y(n) = y(n-1) + a_0 r(n) + a_1 r(n-1)$$

When writing this computation as a computer program, it takes the form of a recursive filter, with the coefficients A0 and A1:

$$Y0 = Y1 + A0 * R0 - A1 * R1$$

where:

- Y0 = Current output
- Y1 = Output at the last sample period
- R0 = Current ADC sample of Back EMF (phase voltage – $V_{BUS}/2$)
- R1 = Most recent sample of Back EMF from ADC
- A0 = a0
- A1 = –a1

A flow chart of the PWM loop is shown in Figure 6. This PWM loop can also be used for specific application code, such as communications or additional user interfaces.

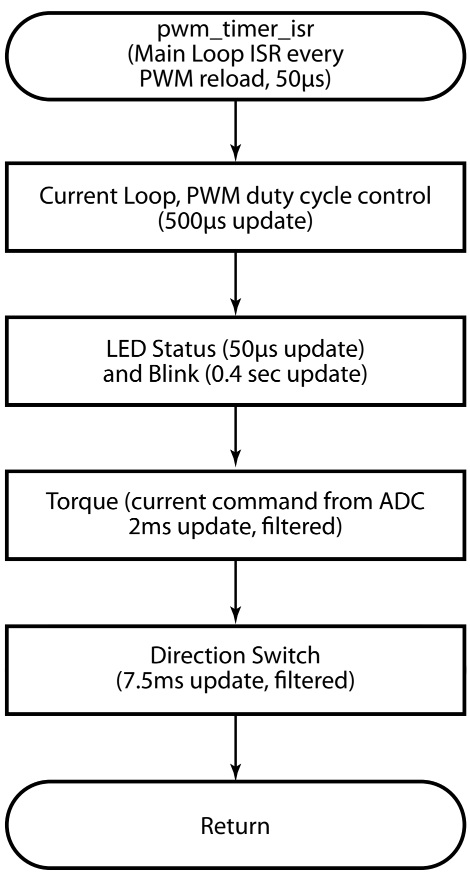

Figure 6. Current Loop and Timed Housekeeping

The Phase Locked Loop back EMF algorithm is critical to smooth startup and operation. Precise control of the PWM is required to create constant waveforms to the motor, resulting in its quiet operation. These waveforms are shown in Figure 7, which captures the voltage on the gate for the High and Low side MOSFETs on Phase A while the motor is running with 24VDC input and at the highest speed setting.

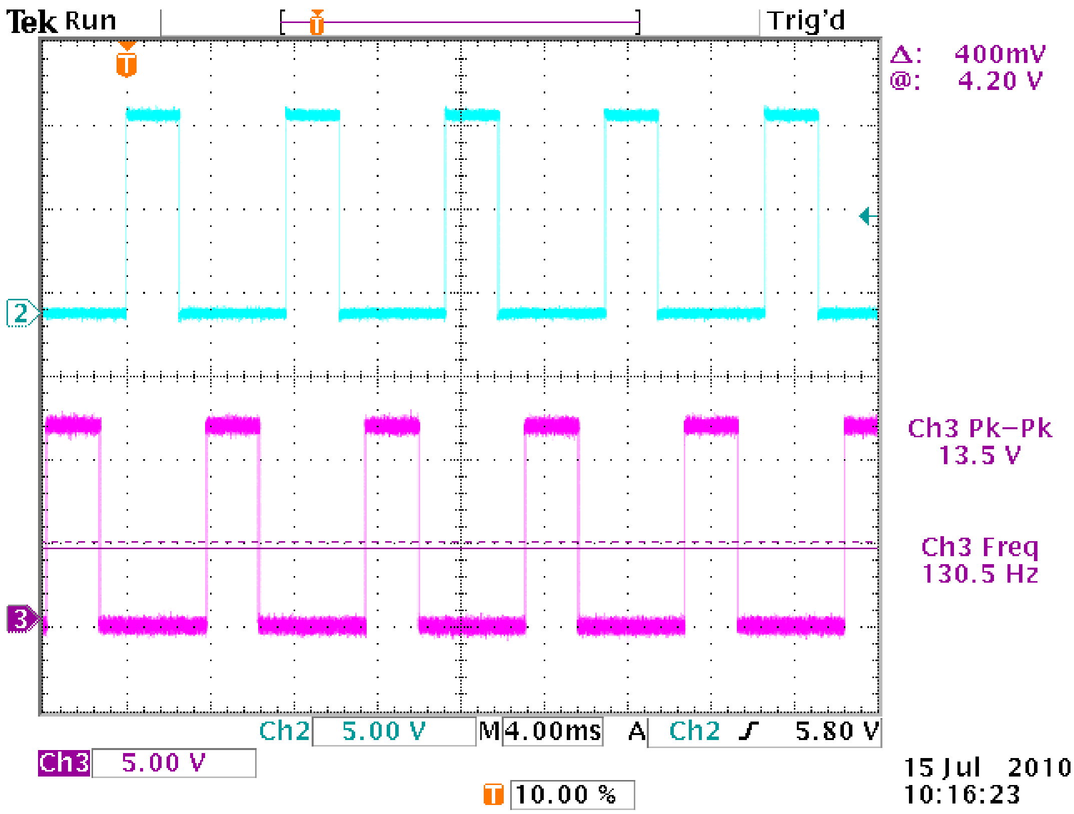

Figure 7. MOSFET Gate Signal

To verify fast shutdown capability during an overcurrent event, this implementation was set up with an oscilloscope tied to the PWM output and to the current sense resistor. The load was then connected to the BLDC Motor Control Development Kit and set up to gradually increase to an overcurrent state. The resulting oscilloscope-generated waveforms representing this sequence of events are shown in Figure 8.

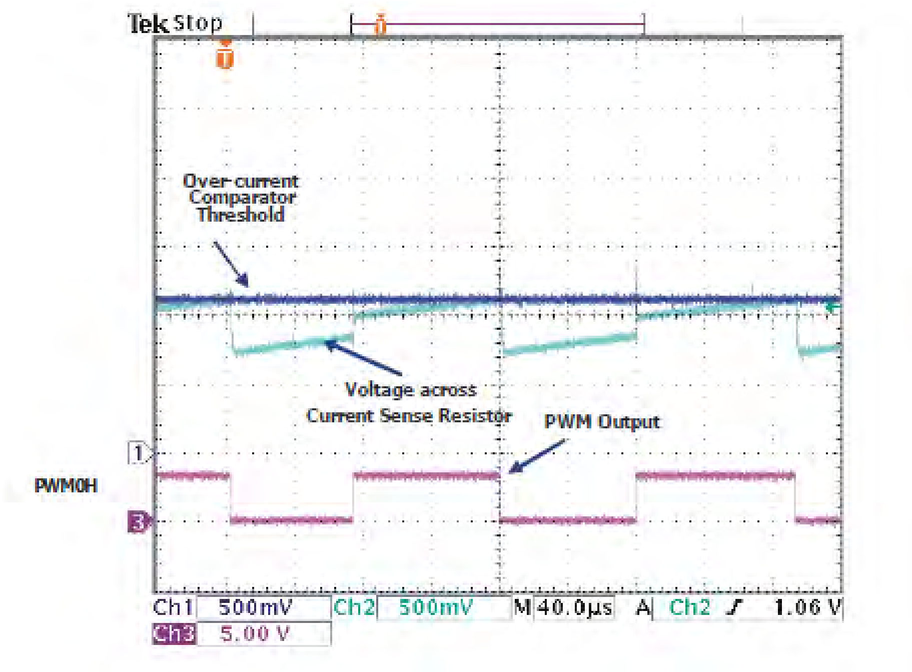

Figure 8. Cycle-by-Cycle Shutdown

Testing

This section provides information about how to run the code and demonstrate this application, including the equipment used to build the implementation, its configuration and the results of its testing.

Equipment Used

The following equipment is used for the setup:

- Zilog Z8FMC16 MDS Module (99C0987-001) - Z8FMC16100KITG
- Zilog 3-Phase Motor Control Application Board (99C0960-001)
- LINIX 3-Phase BLDC motor 24VDC, 30W 3200RPM (45ZWN24-30)
- Opto-Isolated USB Smart Cable
- 5V DC power supply for the Z8FMC16 MCU
- 24 V DC power supply for the BLDC motor
- Digital oscilloscope or logic analyzer

Hardware Setup

Figures 9 and 10 illustrate the application hardware connections.

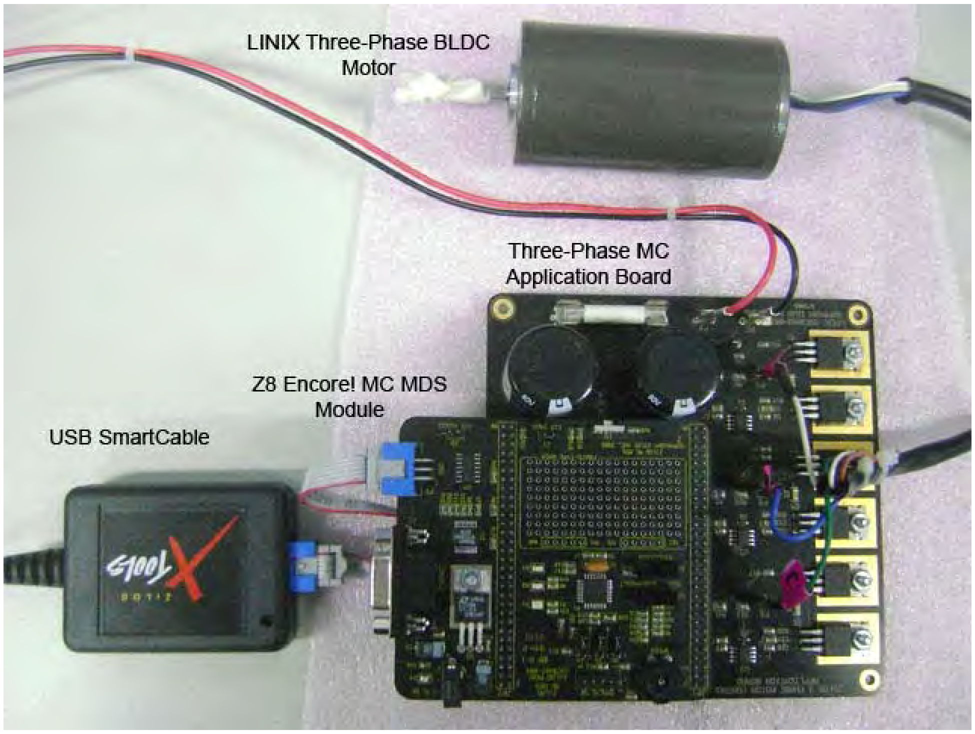

Figure 9. Z8FMC16100KITG and Motor Control Application Board

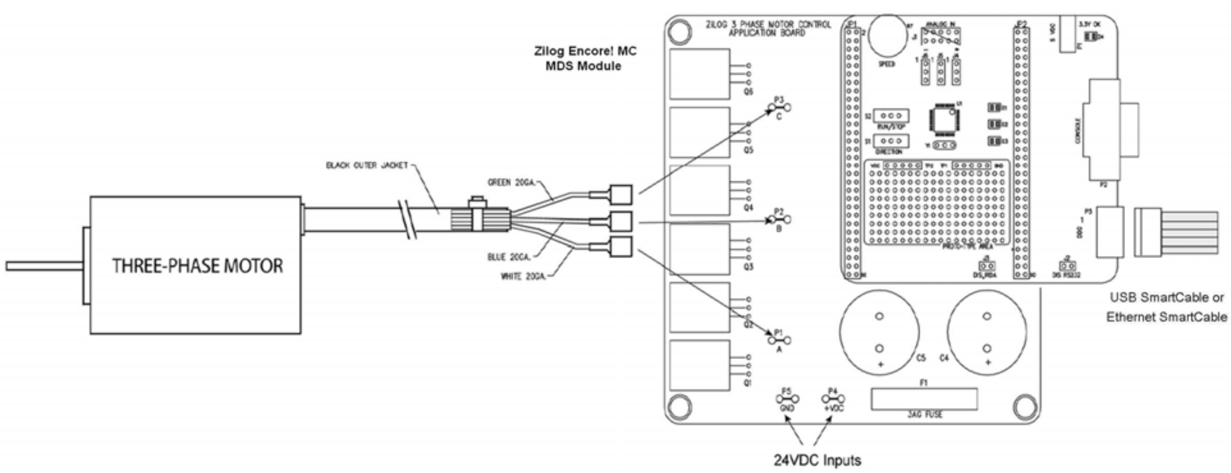

Figure 10. 3-Phase BLDC Hardware Setup with the Z8FMC16 Module

Connecting the Hardware

Observe the following steps to connect and configure the implementation.

1. Attach an RS-232 Cable to the Z8FMC16 MDS Board's Console connector, P2.
2. Adjust the speed potentiometer to mid-position.
3. Configure a terminal emulation client, such as Hyper Terminal, to reflect the following settings: 57600, 8, N, 1.
4. Connect the Opto-Isolated USB Smart Cable to the Z8FMC16 MDS Board's Debug connector, P3.
5. Apply power to the P1 connector on the Z8FMC16 MDS board; as a result, LED D4 should illuminate.

Test Procedure

Follow the steps below to test the 3-Phase Sensorless BLDC Motor Control Application demo program to the Z8FMC16 MDS Module:

1. Launch ZDS II - Encore! version 4.11.0 or later.
2. Connect the Opto-Isolated USB Smart Cable to the PC.
3. Install the driver for the Opto-Isolated USB Smart Cable. (For assistance, refer to the installation guide for the Opto-Isolated USB Smart Cable, which is included in the Z8FMC16100KITG kit).
4. Connect the hardware as shown in Figure 10.
5. Power up the Z8FMC MDS board using the 5VDC adapter included in the kit.
6. Open the AN0226-SC01 project from within ZDS II.
7. In the main.c file, choose your preferred mode of motor control from the following loop select definition statement:

   ```
   #define LOOP_SELECT_VALUE 1u // 0u = torque loop, 1u =
   speed loop
   ```

8. Compile the application and download the code to the Z8FMC16 MDS Module.
9. In ZDS II, stop the debug mode. Switch off the power supply to the Z8FMC16 MDS board and disconnect the Opto-Isolated USB Smart Cable.
10. Connect the 24 VDC supply source to the motor control application board.
11. Ensure that the RUN/STOP switch on the Z8FMC16 development board is in the Stop position.
12. Turn on the Z8FMC16 development board supply, then turn on the 24V supply to the motor control application board.
13. Move the switch position on the Z8FMC16 development board to the RUN position.

- If SPEED mode is selected, the speed of the motor is adjusted by varying the potentiometer on the Z8FMC16 development board.
- If TORQUE mode is selected, the motor speed is decreased with application of force on the shaft of the motor.

14. Set the motor's direction of rotation by changing the position of the direction switch on the Z8FMC16 development board.

    You can now add your specific application software to the main program to realize additional functions.

Note: While debugging your code, ensure that the Opto-Isolated USB Smart Cable controls the reset pin of the MCU. Additionally, to run the code, do not switch S2 to the Reset position. The ZDS tool cannot communicate with the MCU if the Reset is in the Low state.

After debugging and running your code, detach the Opto-Isolated USB Smart Cable from pin P3 to free the Reset pin and apply a power cycle to reset the MCU from Debug mode.

Results

The motor control application was tested with the Z8FMC16 MDS board connected to the Zilog 3-Phase Motor Control application board. The BLDC motor specifications are:

- Manufacturer: Linix
- Motor type: 3-wire, 3-phase brushless DC motor
- Voltage rating: 24 V
- Power rating: 30 W
- Maximum speed of rotation: 3200 RPM

LED indicator

- RED: indicates DC power loss to the motor/fault condition
- YELLOW: indicates UART control
- GREEN: indicates that the motor is running

The terminal emulation program should show the following example display:

```
Zilog Encore! MC>>> Press ESC for commands
```

Note: By default, this application uses hardware control because the UART control function, while it can allow adjustments for motor direction, changes in motor speed and reading speed via software, is a supplementary control element included merely for demonstration purposes. To use UART control, ensure that S2 is in the RUN position.

Pressing the Escape key (Esc) displays the following list of commands:

```
I : User interface
U : Give back to hardware
S : Start motor
E : Stop motor
F : Forward direction
R : Reverse direction
V : DC Voltage reading
C : Current speed reading
M : Set motor speed
```

Select `I` to control the motor using UART commands. The terminal emulation program inherits the motor's start/stop settings, its forward/reverse direction, DC voltage and current speed readings, and motor speed settings, ranging across 32–9B hexadecimal bytes. Select `U` to return control to the hardware.

Summary

This application note described the closed-loop control of a sensorless BLDC motor using the advanced on-chip integrated features of the Z8 Encore! MC Family of Microcontrollers. The Z8 Encore! MC product line is ideally suited for such applications, providing for a seamless startup of the motor from idle mode to full operational speed, on-the-fly reversal of the direction of rotation, an extremely fast fault detection cycle, and lower total solution cost. These features, along with the powerful eZ8 CPU core and some of the best development tools available in the industry, result in less complex board designs and reduced design cycle time.

References

The following documents are associated with the Z8 Encore! MC Family of Microcontrollers; each is available for download on www.zilog.com.

- Z8FMC16100 Series Product Brief (PB0166)
- Z8FMC16100 Series Product Specification (PS0246)
- Z8FMC16100 Series Motor Control Development Kit Quick Start Guide (QS0054)
- Z8FMC16100 Series Motor Control Development Kit User Manual (UM0192)

- Z8FMC16100 Series In-Circuit Emulator and Development Tool User Manual (UM0190)
- BLDC Motor Control Using the Z8FMC16100 Application Brief (AB0005)
- Sensorless Brushless DC Motor Control with the Z16FMC MCU Application Note (AN0311)

Appendix A. Schematics

Figures 11 through 13 show the schematics for the 3-Phase Motor Control Application Board.

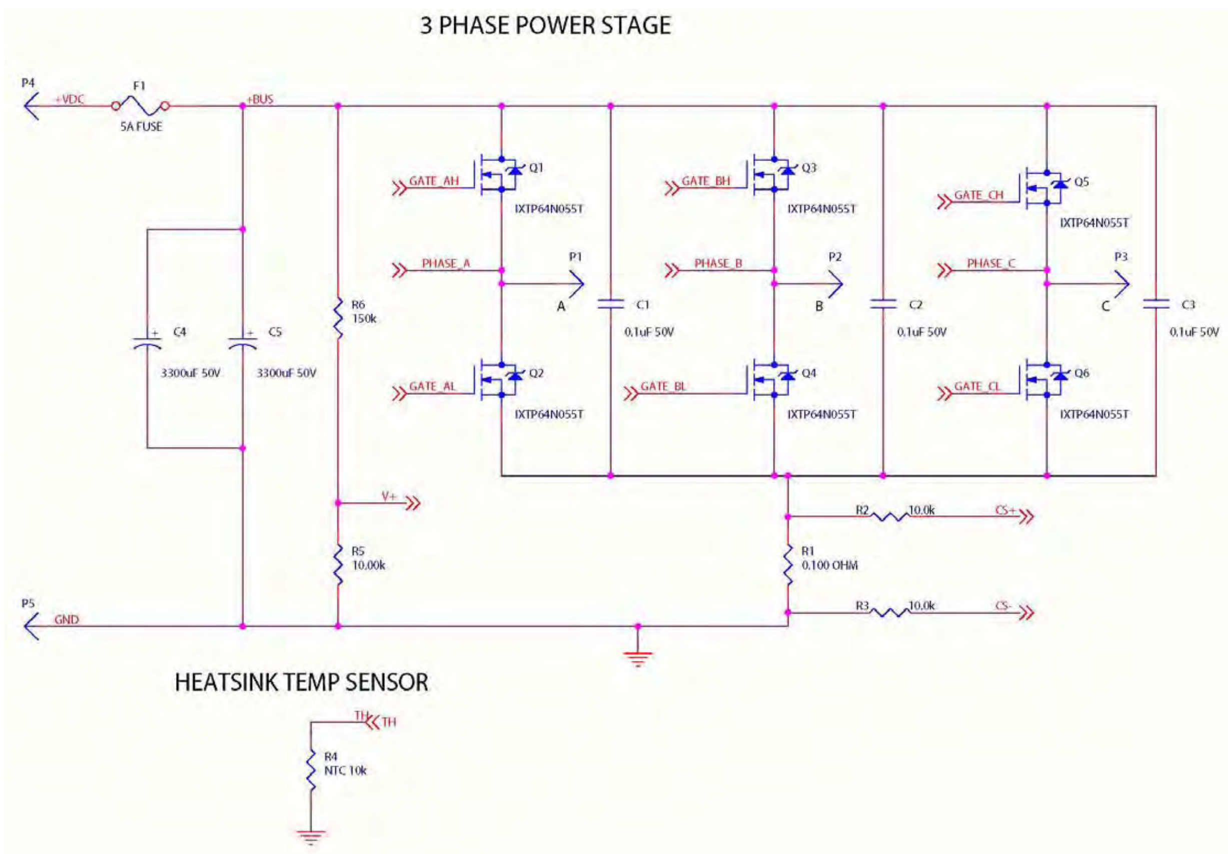

Figure 11. 3-Phase Motor Control Application Board, Part 1 of 3

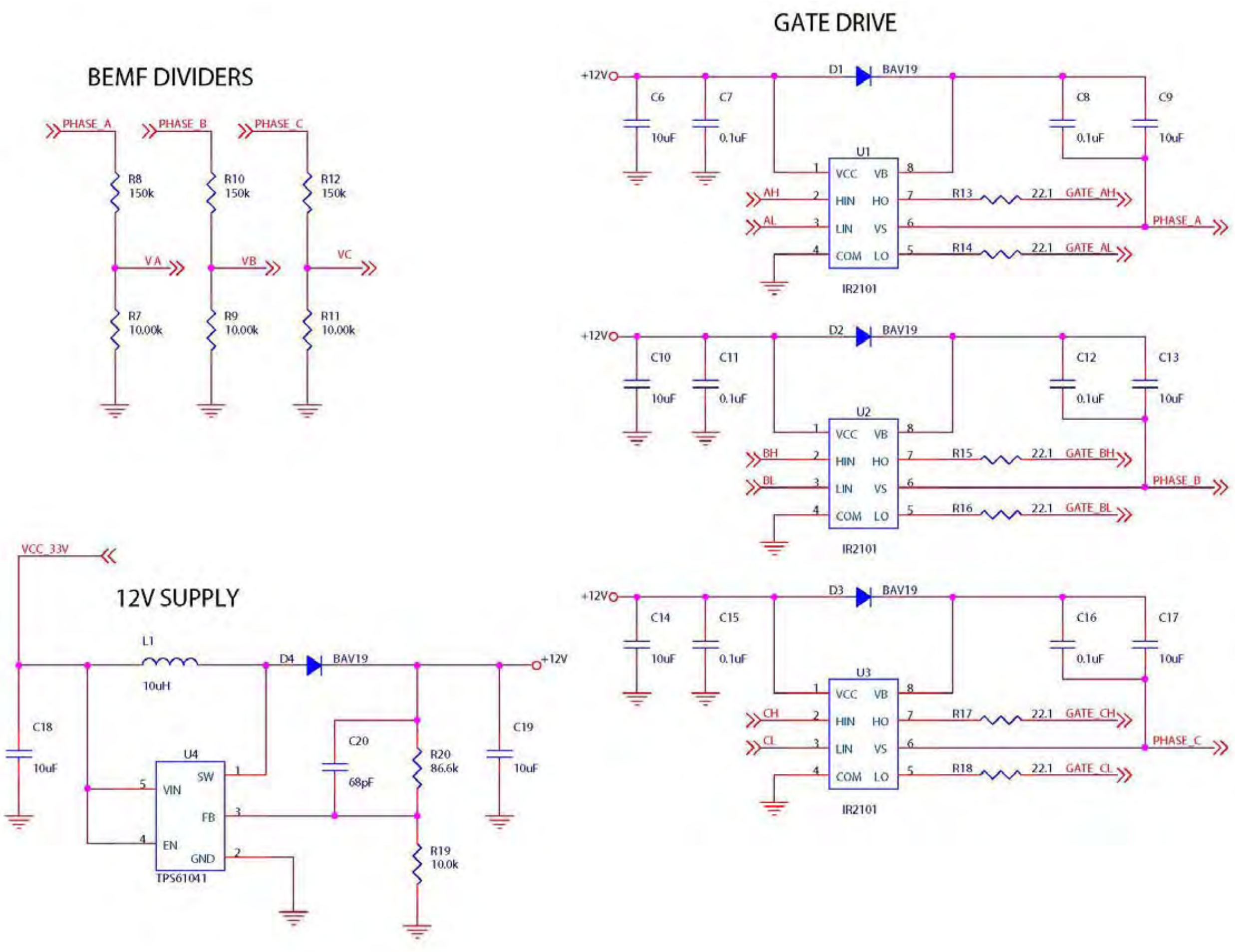

Figure 12. 3-Phase Motor Control Application Board, Part 2 of 3

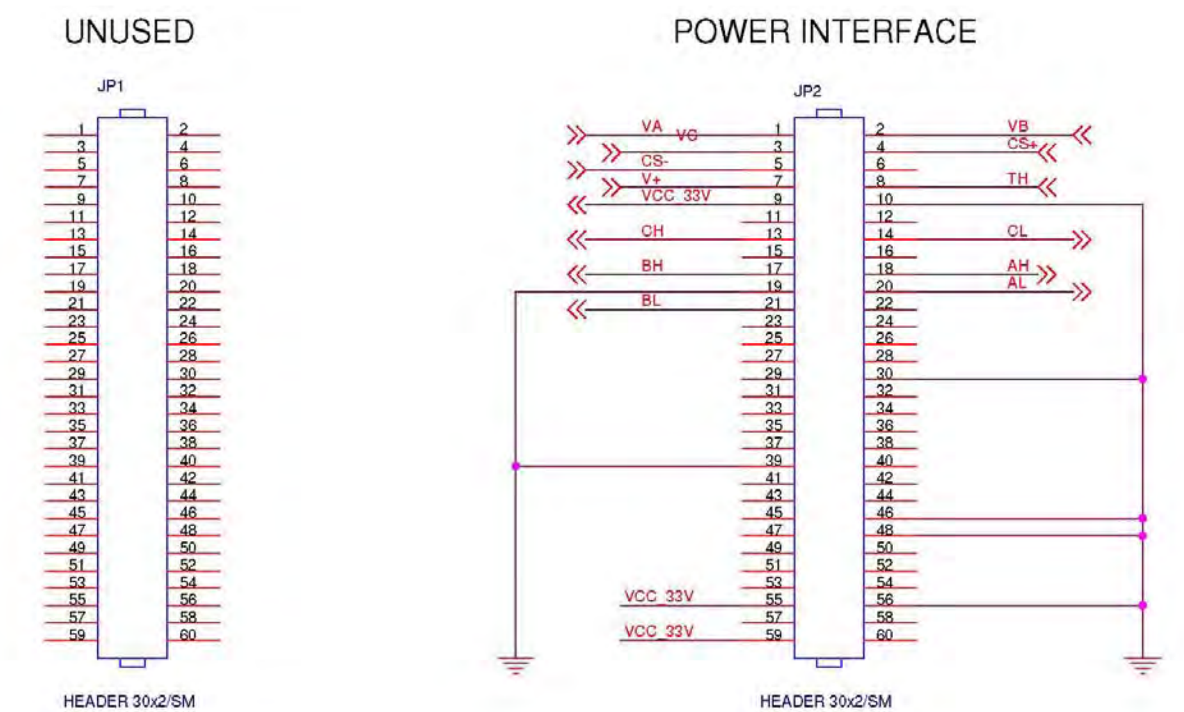

Figure 13. 3-Phase Motor Control Application Board, Part 3 of 3

Figures 14 and 15 show the schematics for the Motor Control MDS Module.

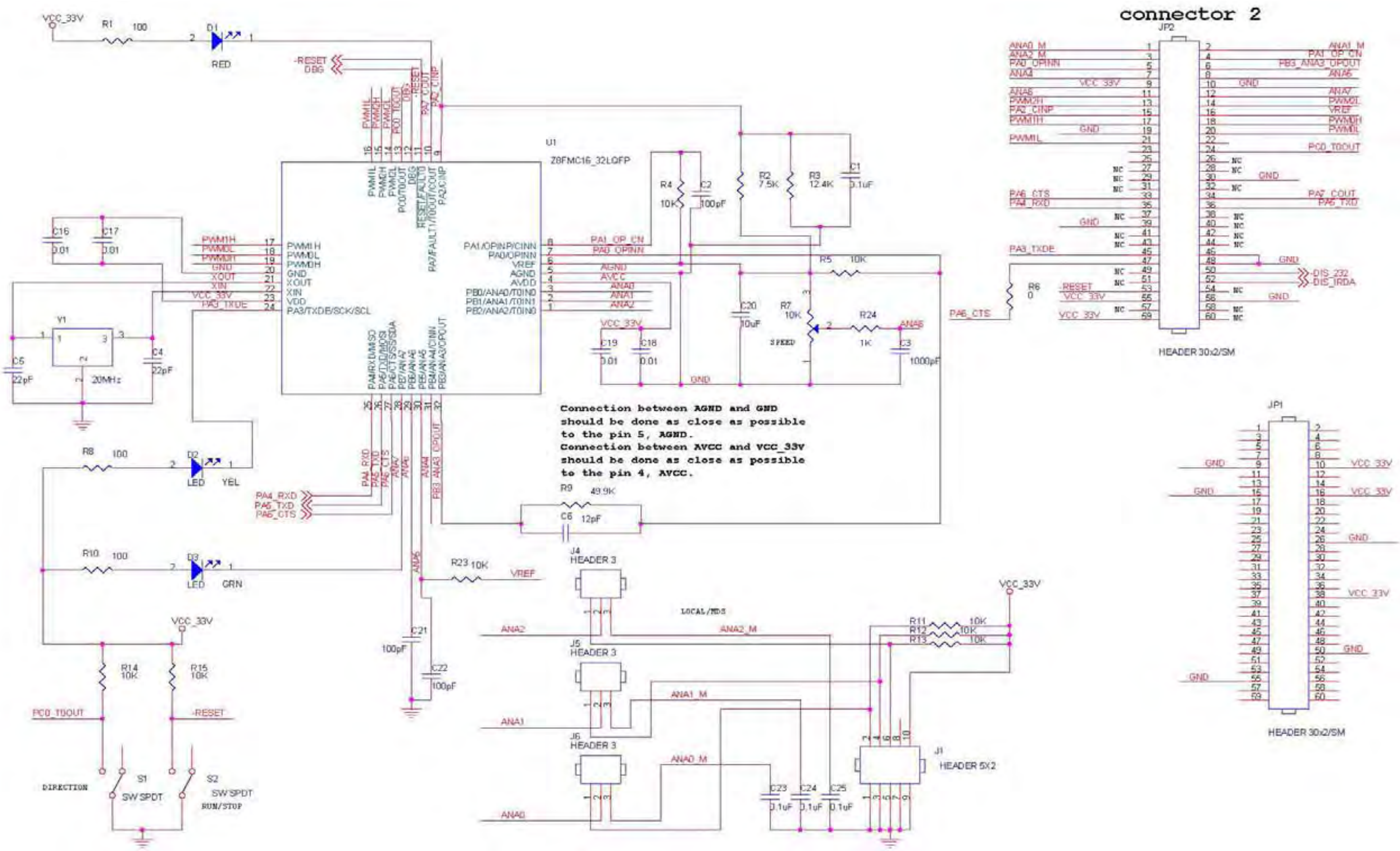

Figure 14. Z8FMC16 MCU and MDS Connectors

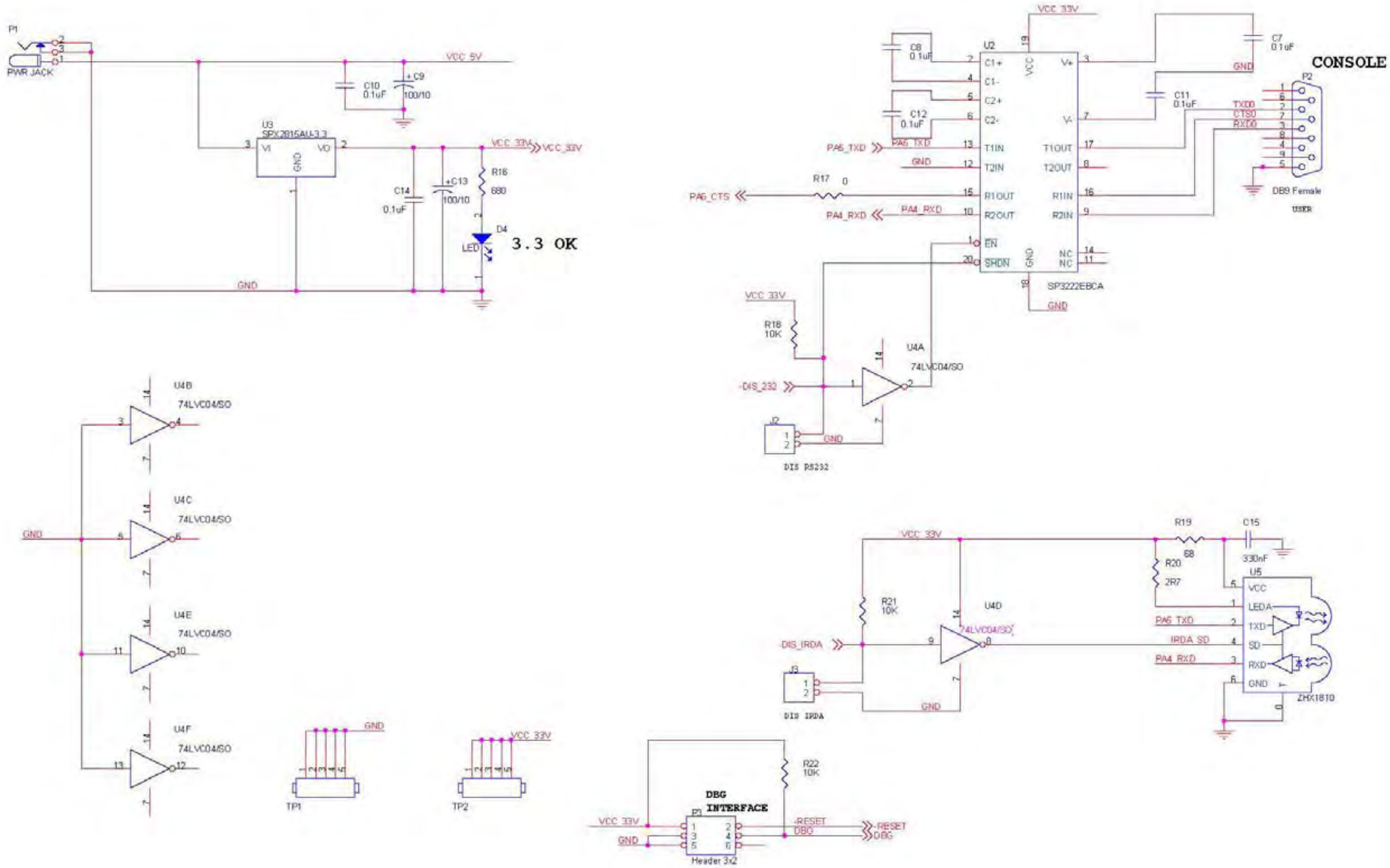

Figure 15. Z8FMC16 MCU and MDS Power and Communications

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer