Zilog
Embedded in Life
An IXYS Company

MultiMotor Series

Application Note

Three-Phase Hall Sensor BLDC Driver Using The Z16FMC MCU

AN035602-1015

Abstract

This MultiMotor Series application note investigates the closed- and open-loop control of a 3-phase brushless direct current (BLDC) motor using a Z16FMC MCU. Zilog's Z16FMC family of microcontrollers is designed specifically for motor control applications and, with this MultiMotor Series, features an on-chip integrated array of application-specific analog and digital modules using the MultiMotor Development Kit. The result is fast and precise fault control, high system efficiency, on-the-fly speed/torque and direction control, as well as ease of firmware development for customized applications.

This document further discusses ways in which to implement motor control using a sensored feedback control system, and fault protection. Test results are based on using a MultiMotor Development Kit equipped with a Z16FMC MCU, a 3-phase MultiMotor Development Board, and a 3-phase, 24 VDC, 30 W, 3200 RPM BLDC motor.

Note: The source code file associated with this application note, AN0356-SC01, was tested with version 5.0.1 of ZDS II for ZNEO MCUs. Subsequent releases of ZDS II may require you to modify the code supplied with this application.

Features

The power-saving features of this MultiMotor Series application include:

- Hall sensor commutation
- Motor speed measurement
- Motor protection logic
- Closed-loop or open-loop control for precise speed regulation
- Potentiometer-adjustable motor speed
- Selectable control of motor direction
- Storage of motor condition parameters into an external EEPROM (for more details, refer to the Zilog application note titled Implementing a Data Logger with Spansion SPI Flash (AN0360).
- UART Interface for PC control
- LED to indicate motor operation
- LED to indicate UART control

- LED to indicate a fault condition

Discussion

Z16FMC Series Flash microcontrollers are based on Zilog's advanced 16-bit ZNEO CPU core. These Z16FMC devices set a standard of performance and efficiency with up to 20 MIPS performance at 20 MHz. The Z16FMC MCU supports 16-bit internal bus widths and provides near-single-cycle instruction execution. Up to 128 kilobytes of internal Flash memory are accessible by the ZNEO CPU, 16 bits at a time, to improve processor throughput. Up to 4 KB of internal RAM provides storage of data, variables and stack operations.

The Z16FMC MCU features a flexible Pulse Width Modulation (PWM) module with three complementary pairs or six independent PWM outputs supporting dead-band operation and fault protection trip input. These features provide multiphase control capability for a variety of motor types, and ensure safe operation of the motor by providing pulse-by-pulse or latched fast shutdown of the PWM pins during a fault condition.

Also featured are up to twelve single-ended channels of 10-bit analog-to-digital conversion (ADC) with a sample and hold circuit. One operational amplifier performs current sampling, and one comparator performs overcurrent limiting or shutdown. A high-speed ADC enables voltage and current sensing, while dual-edge interrupts and a 16-bit timer provide a Hall-effect sensor interface.

Two full-duplex 9-bit UARTs provide serial, asynchronous communication and support an option for the Local Interconnect Network (LIN) serial communications protocol. The LIN bus is a cost-efficient single master/multiple slave organization that supports speeds up to 20 kbps.

The Z16FMC MCU offers a rich set of peripherals and other features, such as an additional 16-bit timer with capture/compare/PWM capability, an SPI, an I^2C master/slave for serial communication, and an internal precision oscillator.

The single-pin debugger and programming interface simplifies code development and allows easy in-circuit programming.

Figure 1 shows a block diagram of the Z16FMC MCU architecture.

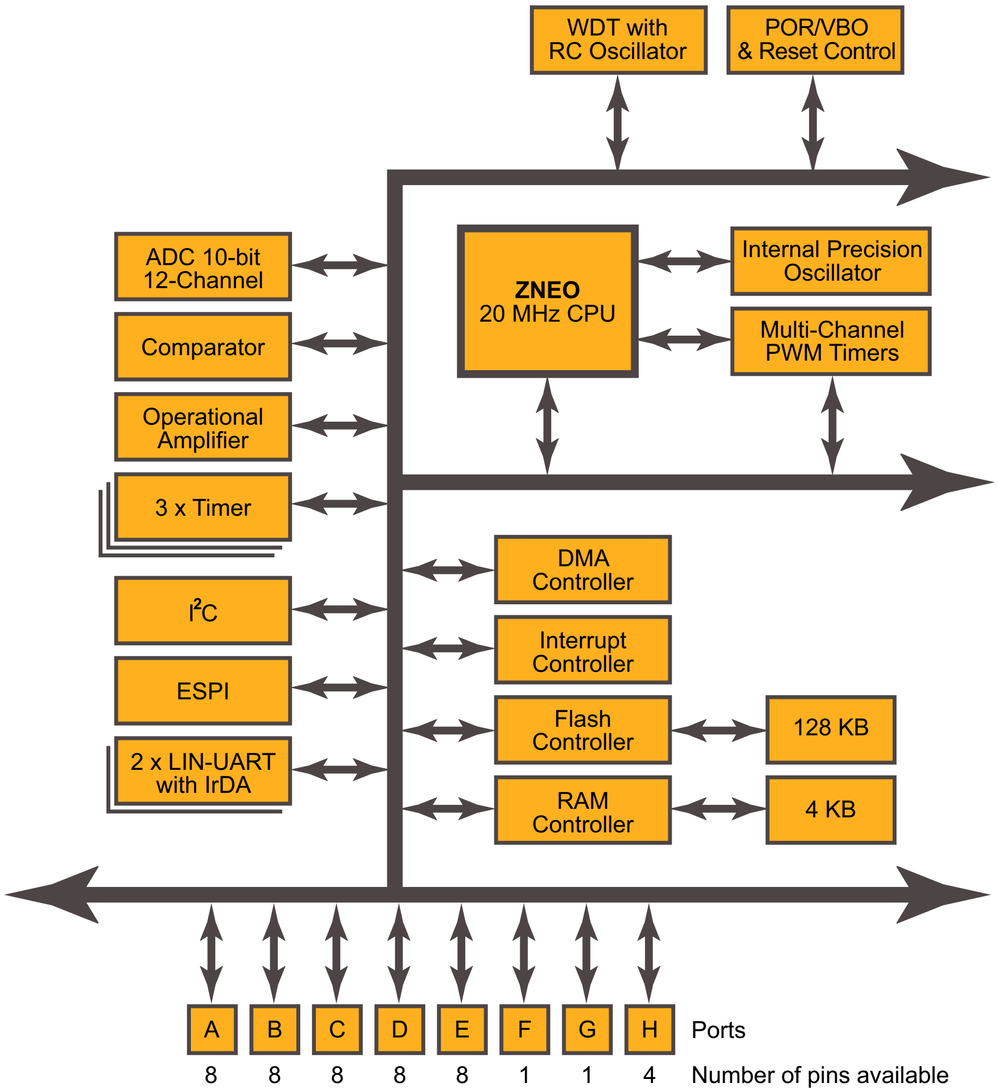

Figure 1. Z16FMC MCU Block Diagram

Hardware Architecture

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. In the 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 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 the motor at very low speeds. The disadvantages are that its implementation requires both separate Hall sensors inside the motor housing and additional hardware for sensor interface.

Sensorless Commutation. In 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.

When a BLDC motor application requires high torque, or when the motor is moving from a standstill, the Hall sensor commutation technique is an appropriate choice. A motor used in an electric bicycle application, for example, requires high initial torque and is a perfect application for Hall sensor commutation.

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

Sinusoidal. Sinusoidal voltage is continuously applied to the three phases. Sinusoidal voltage provides a smooth motor rotation and fewer ripples.

Trapezoidal. DC voltage is applied to two phases at a time, and the third phase remains idle. Trapezoidal voltage is less complex to implement. The idle phase is generating the BEMF from the rotating magnet that passes the unenergized idle phase and provides the BEMF zero-crossing data.

How Hall Sensor Commutation Works

To better understand how Hall sensor commutation works, let's look at how it's implemented with a two-pole motor. Six different commutation states are required to turn the rotor one revolution. The motor’s commutation states are shown in Figure 2.

Table 1 indicates the relationship between the Hall sensor output and phase switching operations shown in Figure 2.

Table 1. Relationship Between Hall Sensor Output and Phase Switching

State	Hall A	Hall B	Hall C	Phase B	Phase C	Phase A
1	0	1	1	0	$+V_{DC}$	$–V_{DC}$
2	0	0	1	$+V_{DC}$	0	$–V_{DC}$
3	1	0	1	$+V_{DC}$	$–V_{DC}$	0
4	0	1	0	0	$–V_{DC}$	$+V_{DC}$
5	1	1	0	$–V_{DC}$	0	$+V_{DC}$
6	1	0	0	$–V_{DC}$	$+V_{DC}$	0

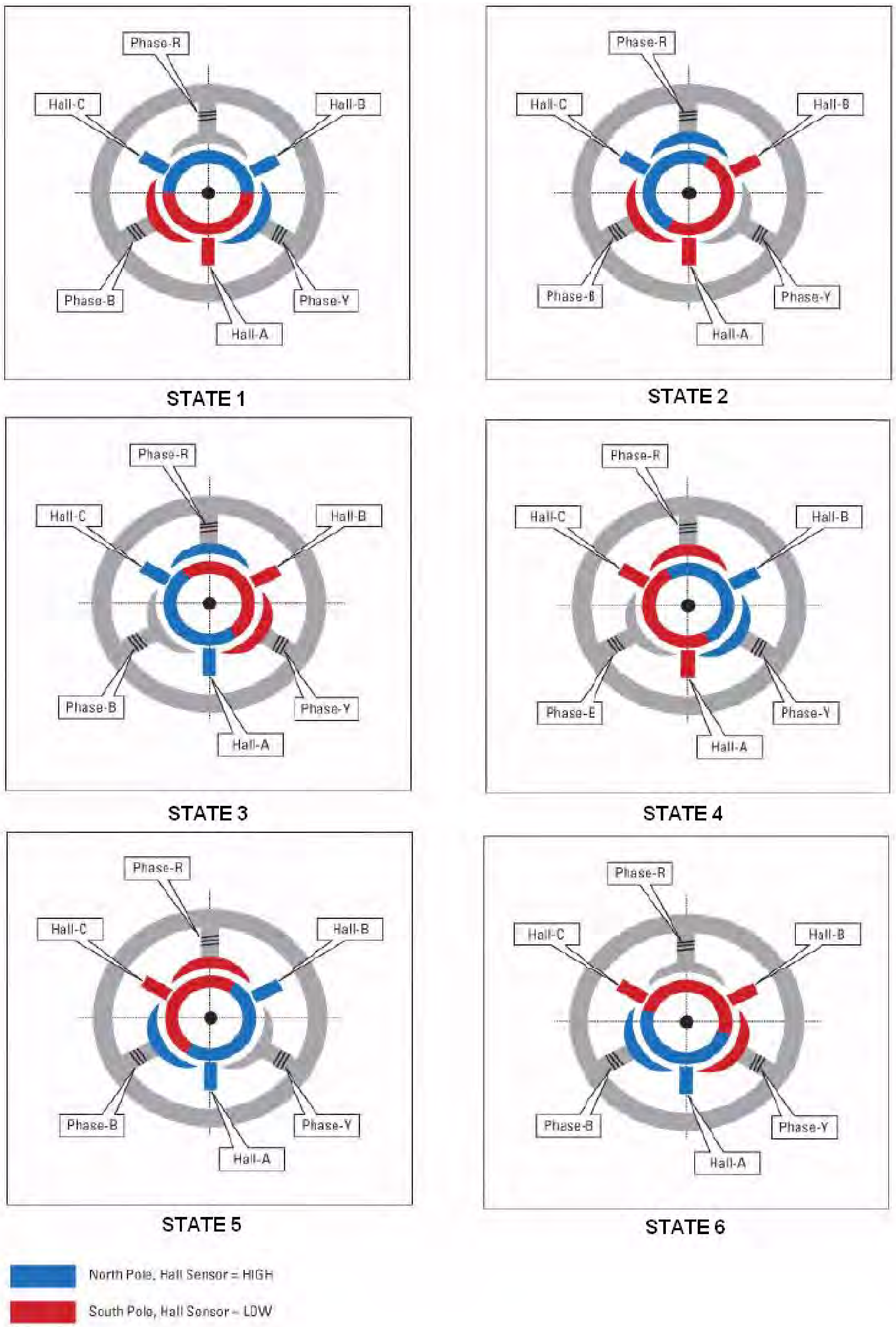

Figure 2. Hall Sensor Commutation States for a 2-Pole Motor

Using the Z16FMC28 MCU with a 3-Phase Hall Sensor BLDC Motor Controller

Figure 3 offers a visual overview of the 3-phase Hall sensor BLDC motor controller. For more details about hardware connections, see Appendix A. Schematic Diagrams on page 13.

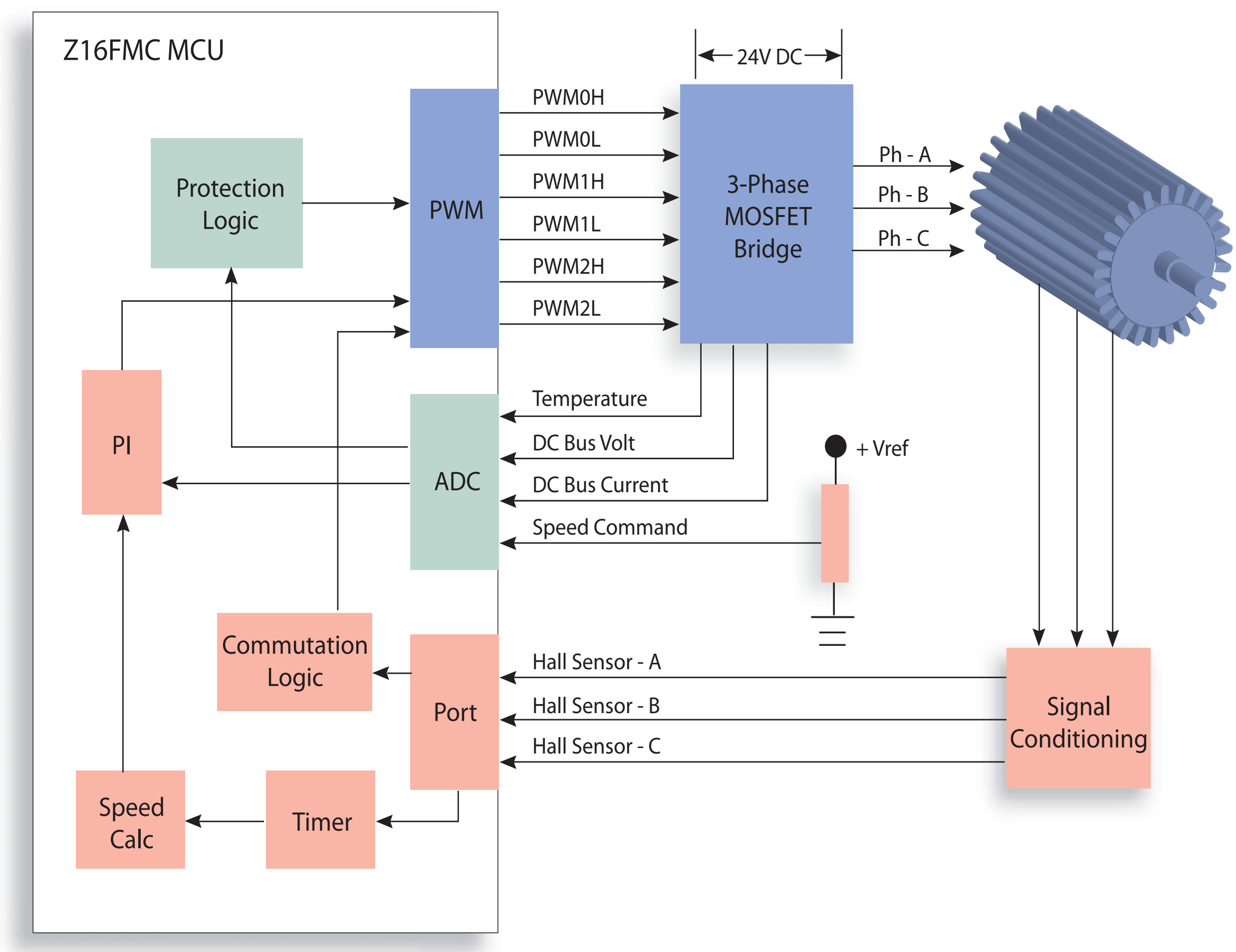

Figure 3. 3-Phase Hall Sensor BLDC Motor Controller Block Diagram

Hardware Design

The design involves running the BLDC motor in a closed loop or an open loop, with speed as set by a potentiometer. As shown in the architecture diagram, the design generates PWM voltage via the Z16FMC MCU's PWM module to run the BLDC motor.

After the motor is running, the states of the three Hall sensors change based on the rotor position. Voltage to each of the three motor phases is switched based on the state of the sensors (commutation). Hall sensor interrupts capture timer ticks every sixty degrees to measure the rotor speed of the motor. Other peripheral functions can be used to protect the system in case of current overload, under- or overvoltage, and overtemperature.

The hardware is described in the following sections.

Three-Phase Bridge MOSFET

The three-phase bridge MOSFET consists of six MOSFETs connected in bridge fashion used to drive the three phases of the BLDC motor. The DC bus is maintained at 24 V, which is same as the voltage rating of the BLDC motor. A separate Hi-Lo gate driver is used for each high- and low-side MOSFET phase pair, making the hardware design simpler and robust. The high-side MOSFET is driven by charging the bootstrap capacitor.

The DC bus voltage is monitored by reducing it to suitable value using a potential divider. The DC bus current is monitored by putting a shunt in the DC return path. An NTC-type temperature sensor provides an analog voltage output proportional to temperature.

PWM Module

The Z16FMC MCU contains a six-channel, 12-bit PWM module configured in this application to run in Complementary Mode. The switching frequency is set to 20KHz. The PWM outputs are controlled according to the inputs from the Hall sensors.

The inputs from the Hall sensors determine the sequence in which the three-phase bridge MOSFET is switched. The Duty cycle of the PWM is directly proportional to the accelerator potentiometer input. The change in the duty cycle controls the current through the motor winding, thereby controlling motor torque.

Commutation Logic

The Hall sensors are connected to ports PD3, PD4, and PD5 on the Z16FMC MCU. An interrupt is generated when the input state on any pin changes. An interrupt service routine checks the state of all three pins and accordingly switches the voltage for the three phases of the motor.

Trapezoidal commutation is used for this application to make implementation simple. In this process of commutation, any two phases are connected across the DC bus by switching the top MOSFET of one phase and bottom MOSFET of another phase ON. The third phase is left un-energized (both top and bottom MOSFET of that phase are switched OFF).

Speed Measurement

One out of the three Hall sensors is used to capture the Timer0 ticks, which represent the actual Hall period for closed loop calculations.

Closed Loop Speed Control

Closed-loop speed control is implemented using a PI loop, which works by reducing the error between the speed set by the potentiometer and actual motor speed. The output of this PI loop changes the duty cycle of the PWM module, thereby changing the average voltage to the motor, and ultimately changing the power input. The PI loop adjusts the speed at the same rate as the Hall frequency from one of three Hall sensors.

Protection Logic

The ADC module periodically checks DC bus voltage, DC bus current, and temperature. If these values go beyond the set limits, the motor is shut down. These checks are timed by Timer0 interrupt.

Over-Current Hardware Protection

The Z16FMC MCU has a built-in comparator that is used to shut down the PWM for over-current protection. When the current exceeds the set threshold, a PWM Comparator Fault is generated to turn OFF the PWM Module.

Software Implementation

During implementation of the software, the following actions are performed:

Initialization. Hardware modules are initialized for the following functions:

- Switch from an internal to an external oscillator for system operation
- Enable alternate functions on the respective pins for the ADC, Comparator, and UART, and to drive the LEDs
- Configure Timer0 to run in Continuous Mode to capture the Hall period timing
- Configure the comparator to shut down the PWM module when an overcurrent results
- Enable the Op Amp to measure the DC bus current flowing to the motor
- Configure the ADC to read analog values such as DC bus voltage, current, temperature, and acceleration potentiometer (only one channel at a time)
- Configure the PWM module for the individual mode of operation with a 20 kHz switching frequency, control output depending on the values in the PWMOUT Register, and drive the PWMOUT as defaulted to a low off state at Power-On Reset and at any Reset

Interrupt. The Port D interrupt controls commutation. The Hall sensor output is read on pins PD3:5, the software performs its filtering operation, and the switching sequence of the MOSFET is determined. The PWM timer interrupt is used to time periodically occurring tasks and for the background loop to read analog values from different channels and average these values, update the LED indicator status, and update the read parameters on the UART.

For a visual representation of the application, see Appendix B. Flowcharts on page 17.

Testing

This section describes how to run the code and demonstrate this sensorless brushless motor application including its setup, implementation and configuration, and the results of testing.

Equipment Used

The following equipment is used for the setup; the first four items are contained in the MultiMotor Development Kit (ZMULTIMC100ZCOG).

- MultiMotor Development Board (99C1358-0001G)
- 24V AC/DC power supply
- LINIX 3-phase 24VDC, 30W, 3200RPM BLDC motor (45ZWN24-30)
- Opto-Isolated UART-to-USB adapter (99C1359-001G)
- Z16FMC MultiMotor MCU Module (99C1357-001G) – Order separately
- Opto-Isolated USB SmartCable (99C0968) – Order separately
- Digital Oscilloscope or Logic Analyzer

Hardware Setup

Figure 4 shows the application hardware connections.

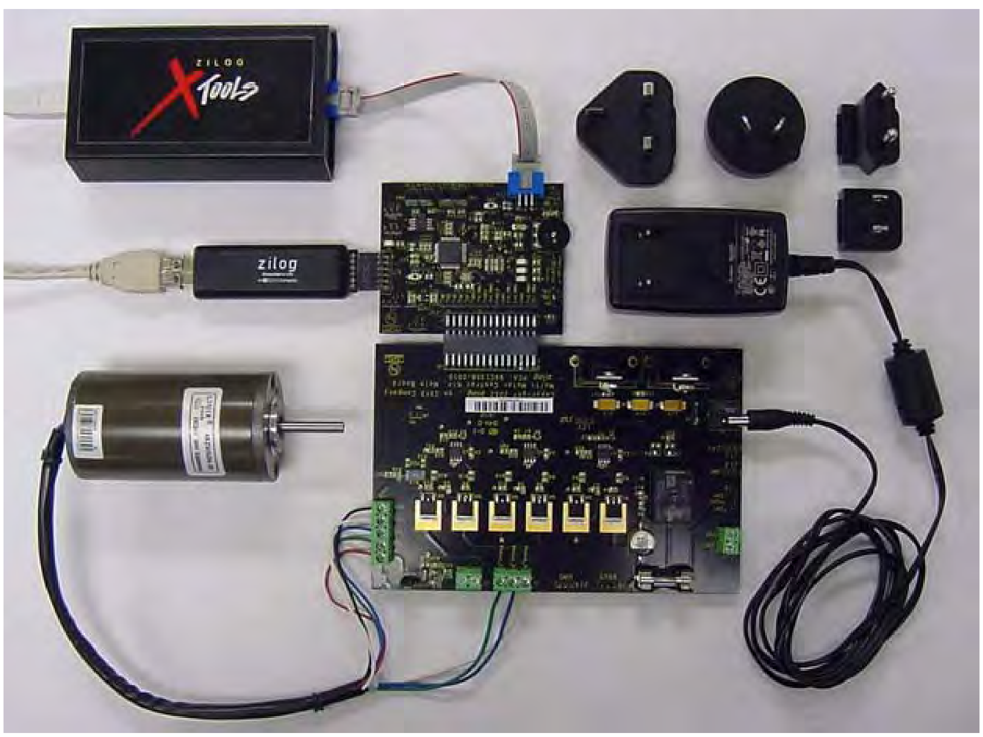

Figure 4. The MultiMotor Development Kit with Z16FMC MCU Module and SmartCable

Figure 5 shows the proper port settings in the terminal emulation program.

Figure 5. Example: Terminal Display Settings

Procedure

Observe the following procedure to test the 3-Phase Sensorless BLDC Motor Control demo program on the Z16FMC MultiMotor MCU Module.

1. Install the ZDSII – ZNEO version 5.0.1 (or newer) software on your PC.
2. Connect the Opto-Isolated USB SmartCable to the PC.
   - To install the driver of the Opto-Isolated USB SmartCable, refer to the installation guide for the Opto-Isolated USB SmartCable that is included in your MultiMotor Development Kit.
3. Connect the hardware as shown in Figure 4 on the previous page.
4. Power up the MultiMotor Development Board using the 24VDC adapter included in the kit.
5. Open the AN0356-SC01 project in ZDSII for ZNEO.
6. Compile the application and download the code to the Z16FMC MultiMotor MCU Module.

7. In ZDSII, stop the Debug Mode. Unplug the power supply from the MultiMotor Development Board, then disconnect the Opto-Isolated USB Smart Cable.
8. Ensure that the RUN/STOP switch on the MultiMotor MCU Module is in the STOP position.
9. Connect the 24V DC supply source to the MultiMotor Development Board.
10. Set the RUN/STOP switch on the MultiMotor MCU Module to RUN.
11. Set the direction of rotation of the motor by changing the position of the direction switch on the MultiMotor Development Board.

You can now add your application software to the main program to experiment with additional functions.

Note: While debugging your code, ensure that the Opto-Isolated USB SmartCable controls the reset pin of the MCU. After debugging and running your code, detach the Opto-Isolated USB SmartCable from J14 of the MultiMotor MCU Module to free the Reset pin and apply a power cycle to reset the MCU from Debug Mode.

Results

This three phase, sensored, brushless motor control application was tested with a 3-phase BLDC motor connected to Zilog's MultiMotor Development Board. Testing of the Z16FMC MultiMotor MCU Module confirms a seamless start-up of the motor from an idle mode to full operational speed, plus on-the-fly reversal of the direction of rotation, an extremely fast fault-detection cycle, and a lower total solution cost.

- Maximum motor speed: 3200RPM
- The motor can be controlled using two methods:
  - Manually using the Stop/Run & Direction switches and the speed pot on the MultiMotor MCU Module
  - Using menu-driven commands on a PC terminal emulator connected to the MultiMotor MCU Module through the UART connections

- The Green LED illuminates when the motor is running
- The Yellow LED illuminates when under UART control
- The Red LED flashes when the motor is stopped or a fault is detected

References

The following documents are associated with the Z16FMC Series of Motor Control MCUs; each is available for download on www.zilog.com.

- Z16FMC Series Motor Control MCU Product Specification (PS0287)
- MultiMotor Series Development Kit Quick Start Guide (QS0091)
- MultiMotor Series Development Kit User Manual (UM0262)
- Zilog Developer Studio II – ZNEO User Manual (UM0171)
- ZNEO CPU Core User Manual (UM0188)
- Sensorless Brushless DC Motor Control with the Z16FMC MCU (AN0353)
- Space Vector Modulation of a 3-Phase AC Induction Motor with the Z16FMC MCU (AN0354)
- BLDC Motor Control on the Z16FMC MCU Using Sensored Sinusoidal PWM Modulation (AN0355)
- Implementing a Data Logger with Spansion SPI Flash (AN0360)

zilog
Embedded in Life
An IXYS Company

Appendix A. Schematic Diagrams

Figures 6 and 7 show the schematics for the Z16FMC MCU Module.

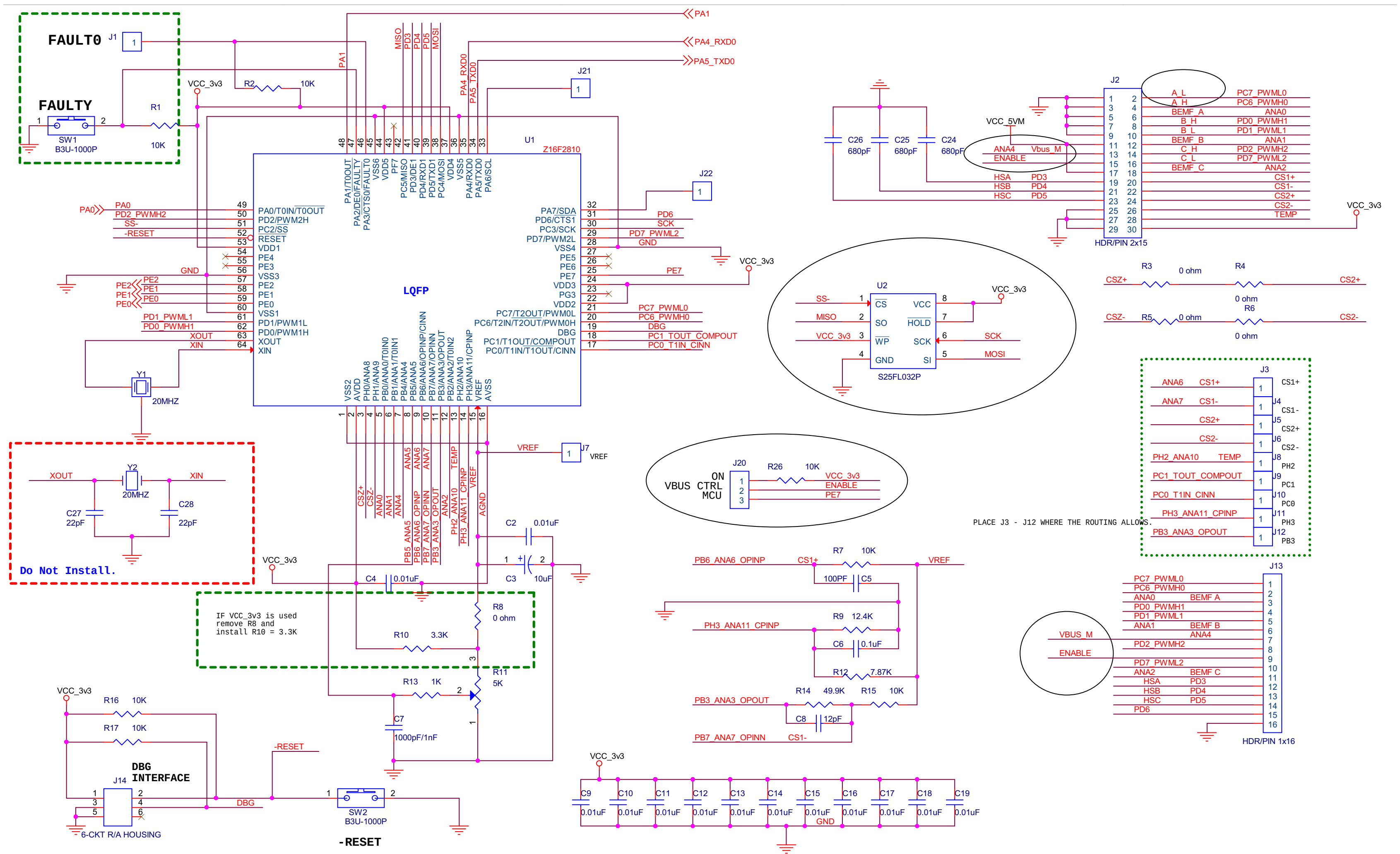

Figure 6. Z16FMC MultiMotor MCU Module, #1 of 2

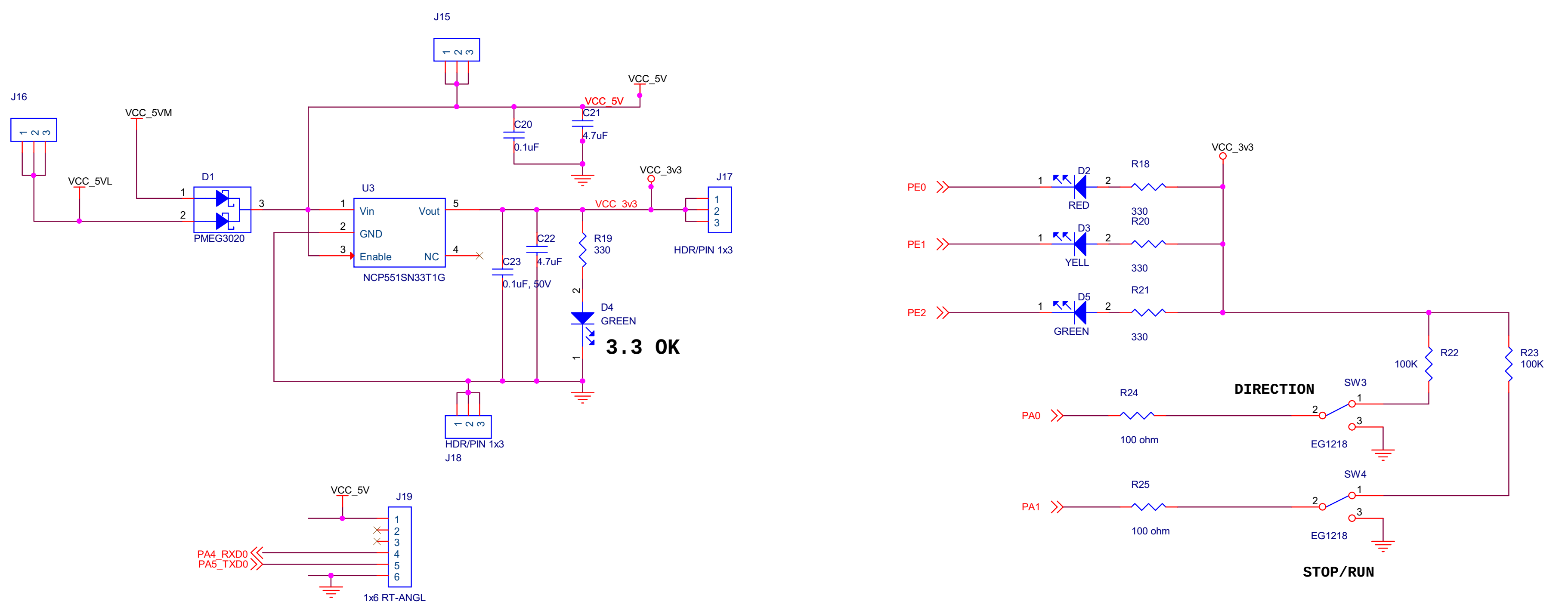

Figure 7. Z16FMC MultiMotor MCU Module, #2 of 2

Figures 8 and 9 show the schematics for the MultiMotor Main Board.

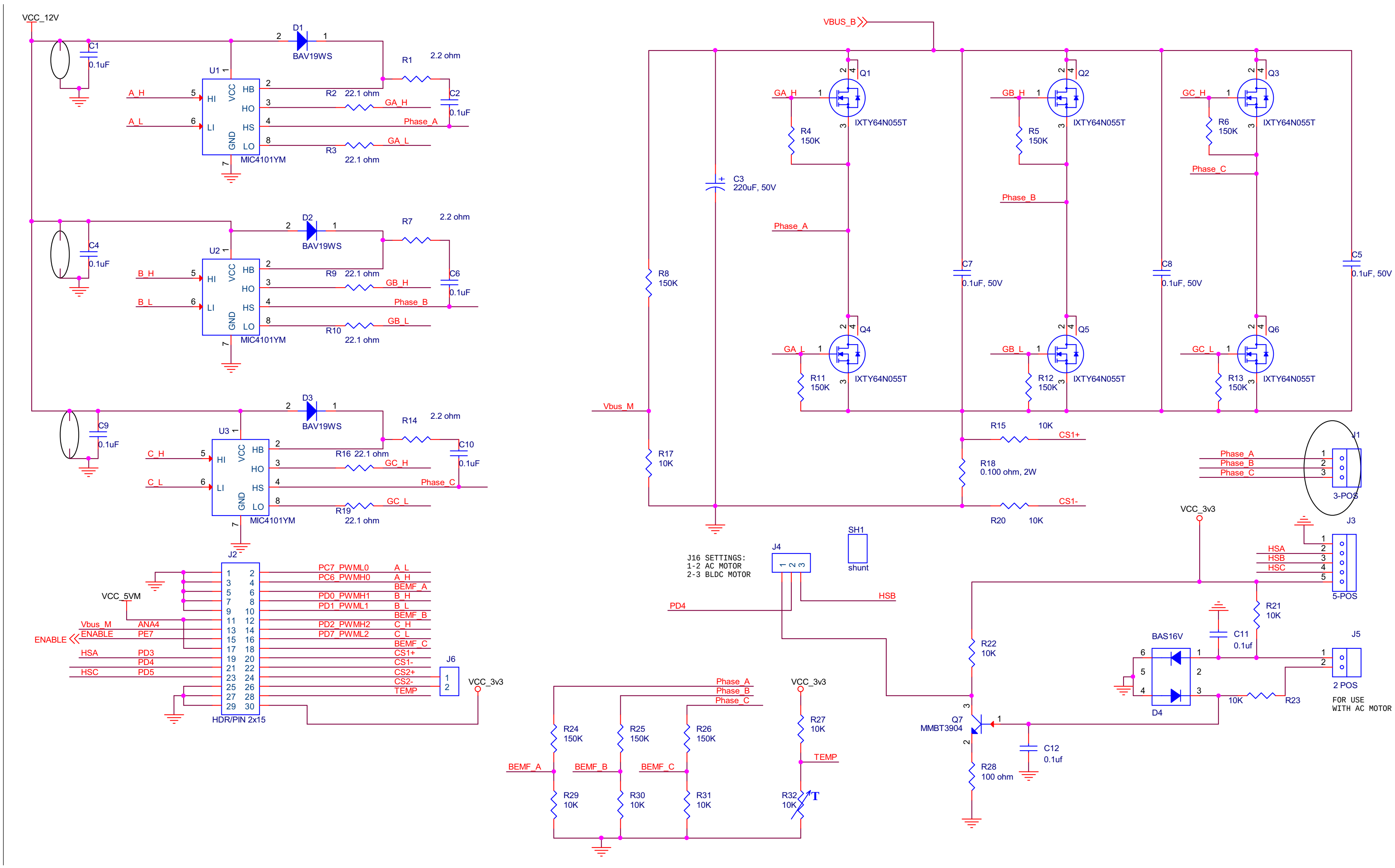

Figure 8. MultiMotor Development Board, #1 of 2

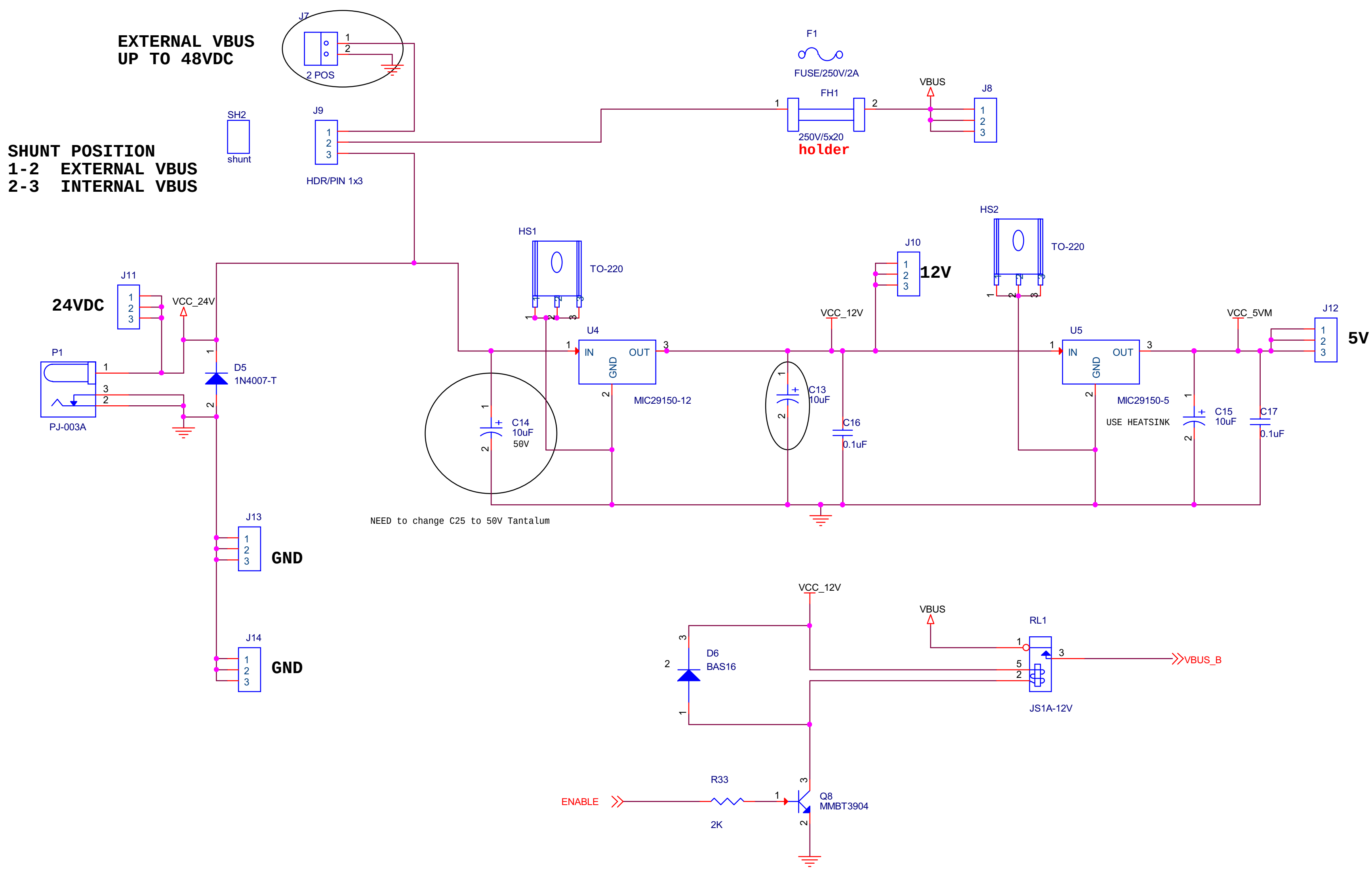

Figure 9. MultiMotor Development Board, #2 of 2

Appendix B. Flowcharts

Figure 10 presents a simple flow chart of the main, timer interrupt and Port D interrupt routines for a 3-phase Hall sensor BLDC motor control application.

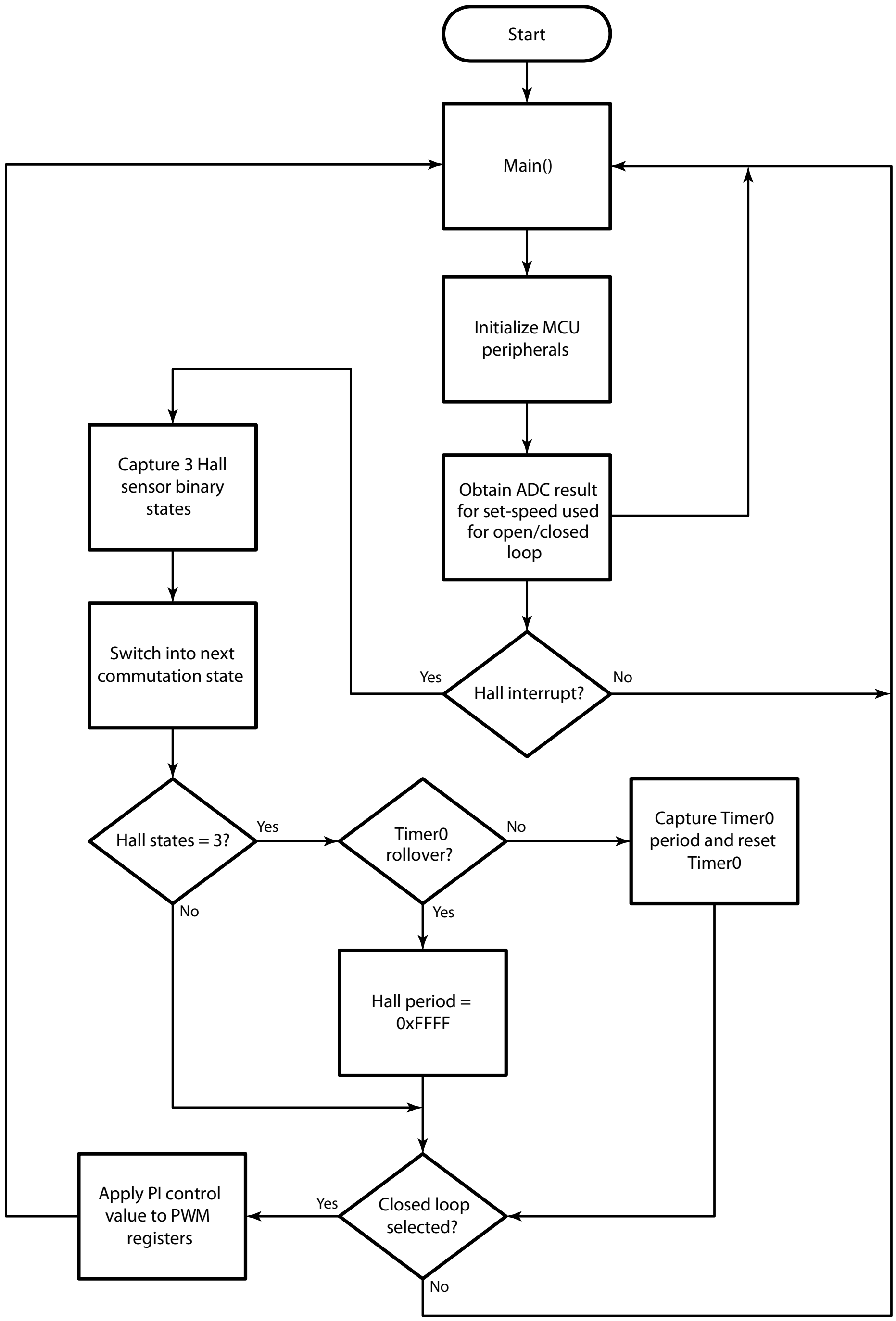

Figure 10. 3-Phase Hall Sensor BLDC Motor Control Application Flowchart

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