Application Note

Three-Phase Hall Sensor BLDC Driver Using the Z51F3220 MCU

AN037102-1215

Abstract

This MultiMotor Series application note investigates the closed- and open-loop control of a 3-phase brushless direct current (BLDC) motor using the Z51F3220 MCU, a member of Zilog's Z8051 series of microcontrollers.. The Z8051 family of MCUs 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, and ease of firmware development for customized applications.

This document discusses methods of implementing motor control using a sensored feedback control system and fault protection. The results are based on using a MultiMotor MCU Module equiped with a Z51F3220 MCU, a 3-phase MultiMotor Development Board, and a 3-phase, 24VDC, 30W, 3200RPM BLDC motor with internal hall sensors.

Note: The source code file associated with this application note, AN0371-SC01, is available free for download from the Zilog website. This source code was compiled using the Keil µVision4 and Small Device C Compiler (SDCC) tools. The Keil µVision4 development tool is available from Keil; the SDCC tool is included in the Z8051 Software and Documentation set, which is available free for download from the Zilog Store. For this source code to work properly with other Z8051 MCUs, minor modifications to the source code may be required.

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
- UART Interface for PC control
- LED to indicate motor operation

- LED to indicate UART control
- LED to indicate a fault condition

Discussion

Z8051 Series Flash microcontrollers are based on Zilog's advanced CPU core. These devices set a standard of performance and efficiency with up to 16 MHz performance. The Z51F3220 MCU operates at near-single-cycle instruction execution. Up to 32 kilobytes of internal Flash memory are available in this microcontroller. Up to 256 bytes of internal RAM and 768 bytes of external RAM provide storage of data, variables, and stack operations.

The Z51F3220 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.

Additional features include up to sixteen single-ended channels of 12-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 to a hyperterminal window to communicate motor control parameters.

The Z51F3220 MCU offers a rich set of peripherals and other features, such as one 8-bit timer and two 16-bit timers, 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 Z51F3220 MCU architecture.

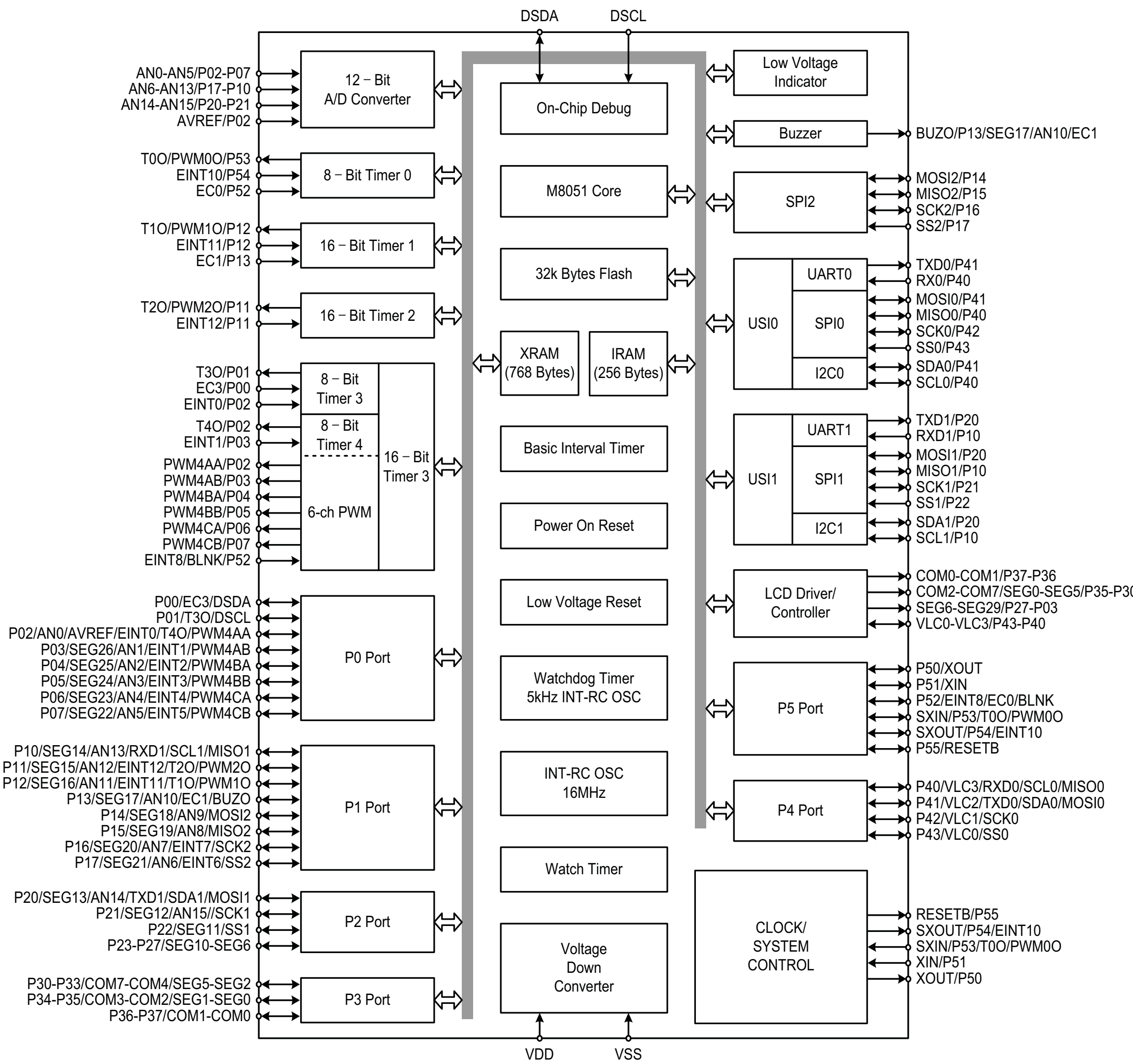

Figure 1. Z51F3220 MCU Block Diagram

Hardware Architecture

In a Brushed DC motor, commutation is controlled by brush position. However, in a BLDC motor, commutation is controlled by the supporting circuitry. Therefore, the rotor's position must 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, typically 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. Using Hall sensors to commutate the motor, the rotor position is available right at startup, when the motor is not turning yet. The disadvantage of using Hall sensors for commutation feedback is increased cost and reliability concerns due to addtional component counts.

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. However, sensorless commutation requires a relatively complex control algorithm. When the speed of the motor is low, the generated BEMF is hard to detect, or too low to be detected and a well defined startup ramp must be implemented to assure reliable commutation at low speeds.

When a BLDC motor application requires high torque, the Hall sensor commutation technique is an appropriate choice. For example, a motor used in an electric bicycle application requires high initial torque and therefore, Hall sensor-based feedback may be the better choice for such types of applications.

Additionally, phase voltage modulation techniques can be applied based on the configuration of the supply-to-motor windings:

Sinusoidal. Sinusoidal voltages are 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, look at how it is 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

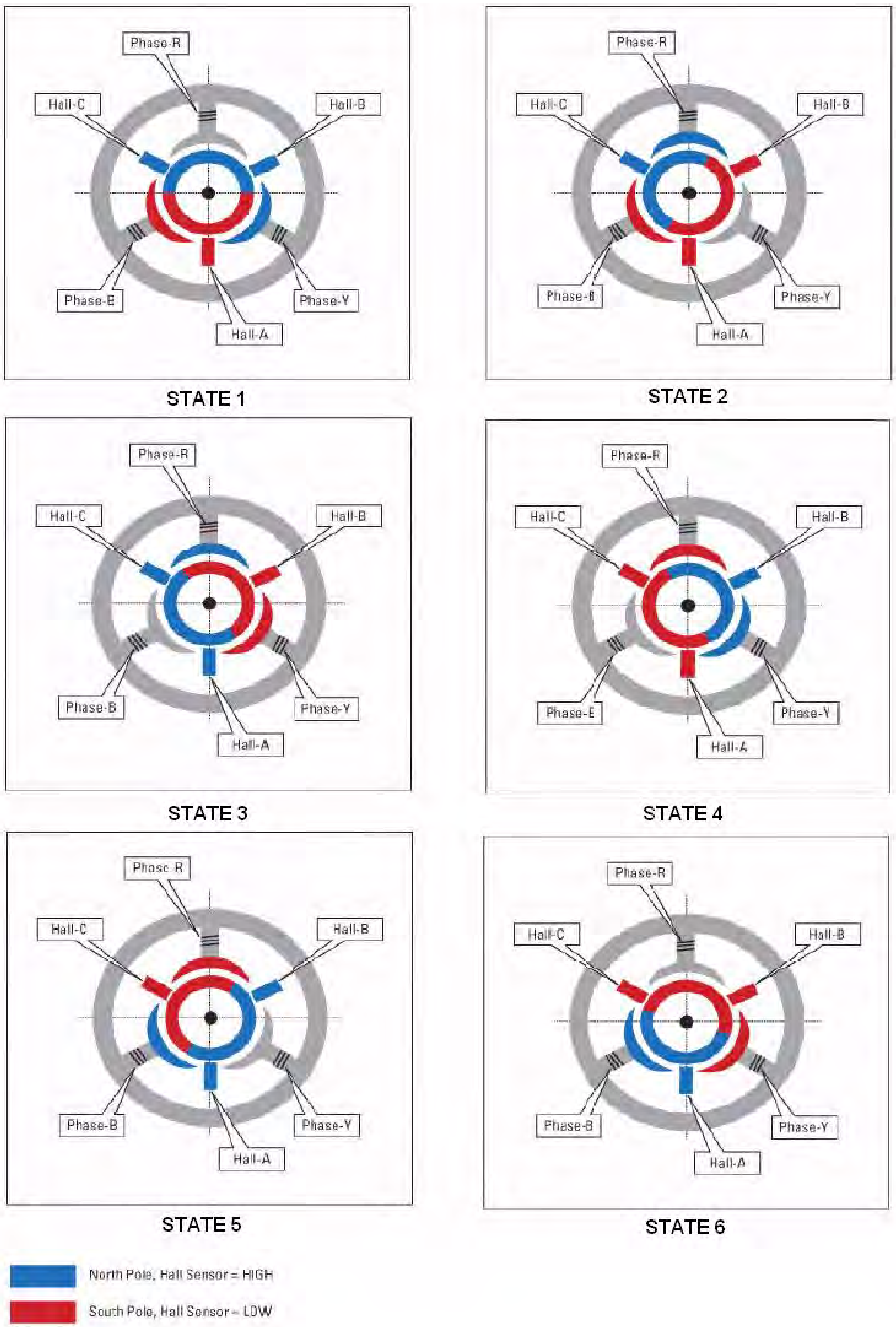

Figure 2. Hall Sensor Commutation States for a 2-Pole Motor

Using the Z8051 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 15.

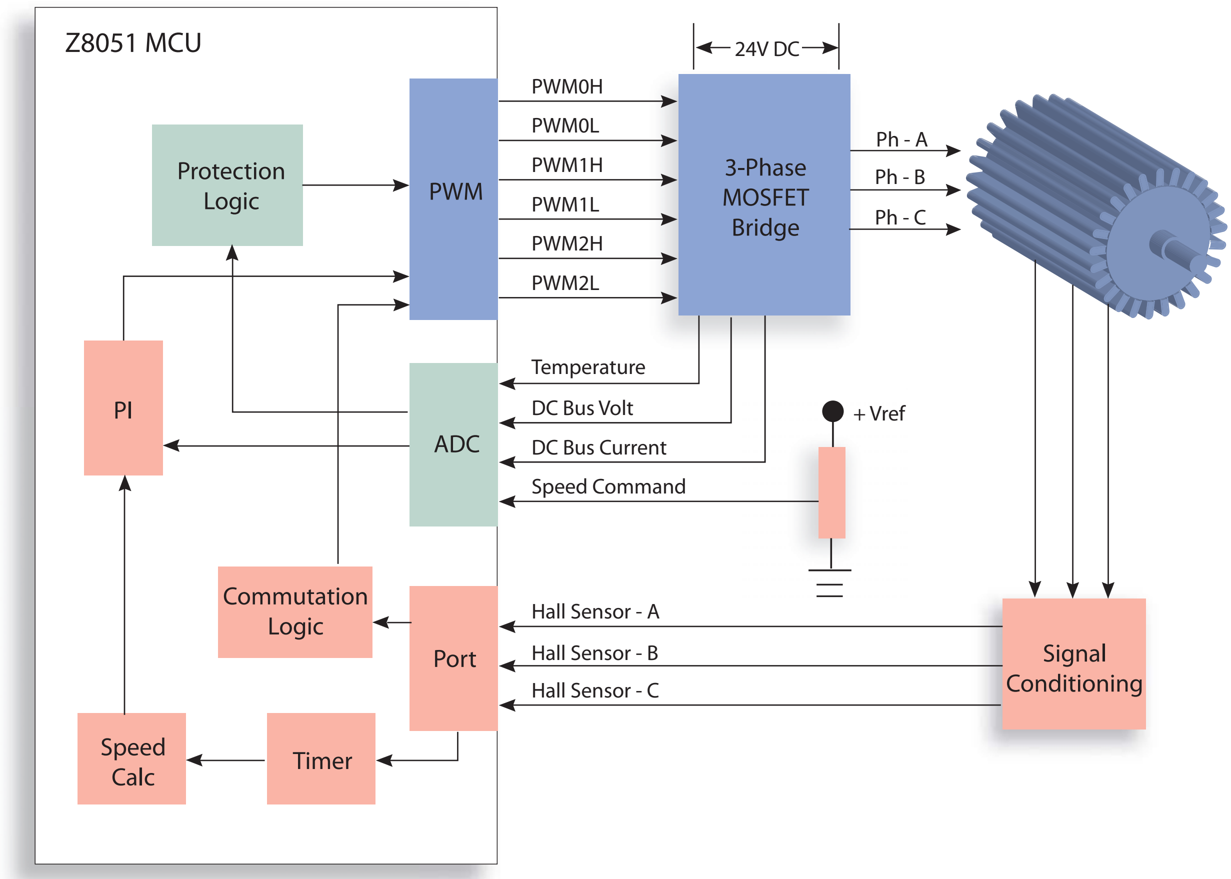

Figure 3. 3-Phase Hall Sensor BLDC Motor Controller Block Diagram

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 Z8051 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 IXYS 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 the 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 simple and robust. The high-side MOSFET is driven by charging the bootstrap capacitor.

The DC bus voltage is monitored by reducing it to a 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 Z51F3220 MCU contains a six-channel, 12-bit PWM module configured to run in Complementary Mode in this application. The switching frequency is set to 20 KHz. The PWM outputs are controlled according to the inputs from the Hall sensors.

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 P11, P12, and P16 of the Z8051 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.

To simplify implementation, trapezoidal commutation is used for this application. In this process of commutation, any two phases are connected across the DC bus by switching the top MOSFET of one phase and the 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 the Timer0 interrupt.

Over-Current Hardware Protection

The Z51F3220 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:

- 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 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 P11, P12, P16 interrupts control commutation. The Hall sensor output is read on pins P11, P12, and P16, 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 19.

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)
- Z51F3220 MultiMotor MCU Module (99C1399-001G) – Order separately
- Z8051 USB On Chip Debugger II Kit (Z51FOCD2000ZACG) – Order separately
- Digital Oscilloscope or Logic Analyzer

Hardware Setup

Figure 4 shows the application hardware connections.

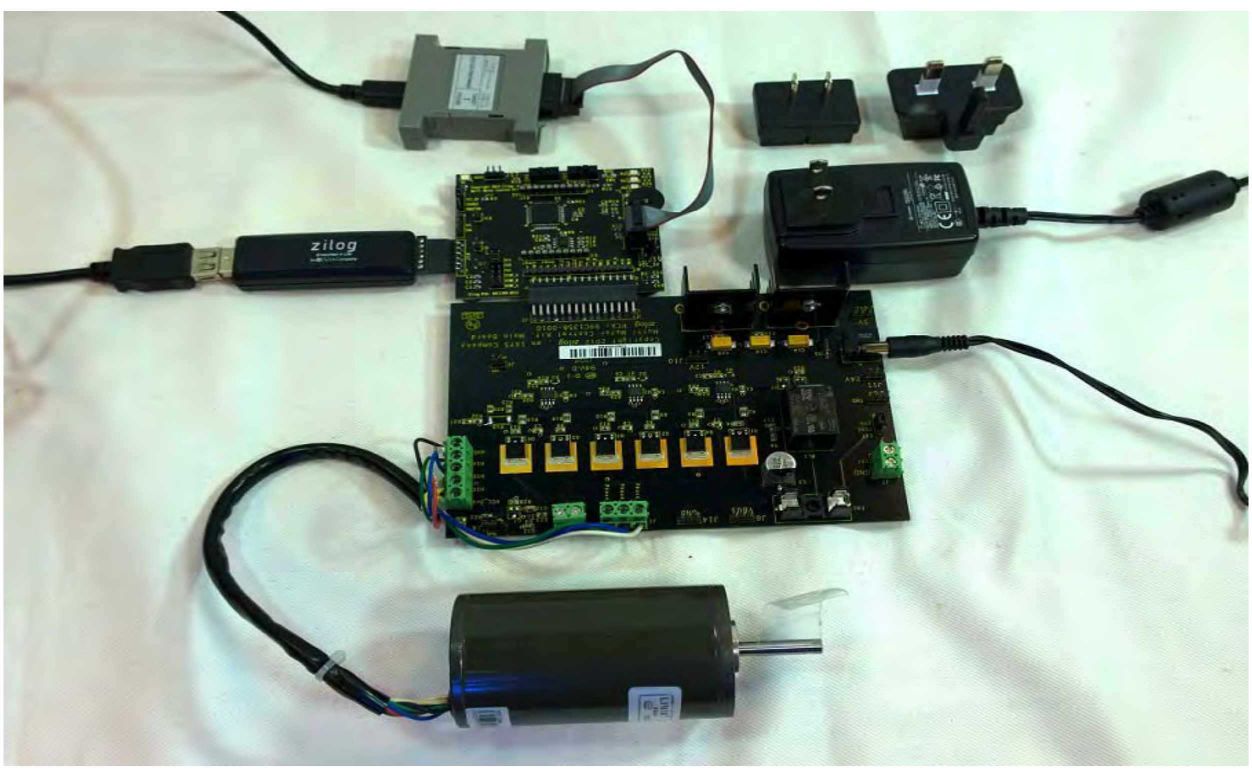

Figure 4. The MultiMotor Development Kit with the Z51F3220 MCU Module and OCD Cable

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

Figure 5. Example: Terminal Display Settings

Testing/Demonstrating the Application

This application includes sample code to show how to load the AN0371-SC01 compiled code. There are two methods for compiling and testing this code:

- Use the Keil compiler for compiling and debugging
- Use the SDCC compiler for compiling the code, and use Zilog's Z8051 OCD software to download a hex file to the Z8051 MCU

This section describes the procedure for enabling each of these two methods.

Test the Code Using the Keil Compiler

Observe the following procedure to test the 3-Phase Sensorless BLDC Motor Control demo program using the Z8051 MultiMotor MCU Module.

1. Install the Keil Z8051 development tool C51 version 9.53.0.0 (or newer) software on your PC.

2. Connect the Zilog Z8051 On-Chip Debugger (OCD) to the host PC's USB port.
3. Connect the other end of the 10-circuit cable connector to the programming connector (J10) on the Z8051 MCU Module. Pin 1 of the cable connector is indicated by a red stripe.
4. Connect the hardware as shown in Figure 4.
5. Power up the MultiMotor Development Board using the 24 VDC adapter included in the kit.
6. Download the AN0371-SC01.zip file and unzip it to a convenient location on your PC.
7. Launch the Keil compiler. From the **Project** menu, select **Open Project** and locate the AN0371-SC01 folder on your PC.
8. From the **Project** menu, select **Rebuild all target files** to recompile all project files.
9. From the **Flash** menu, select **Download** to download the code to the MCU.
10. In Keil, stop the Debug Mode. Unplug the power supply from the MultiMotor Development Board, then disconnect the Zilog Z8051 USB OCD cable.
11. Ensure that the RUN/STOP switch on the MultiMotor MCU Module is in the STOP position.
12. Connect the 24 V DC supply source to the MultiMotor Development Board.
13. Set the RUN/STOP switch on the MultiMotor MCU Module to RUN.
14. Set the direction of rotation of the motor by changing the position of the direction switch on the MultiMotor Development Board.

To experiment with additional functions, add your application software to the main program.

Test the Code Using the SDCC Compiler

Observe the following procedure to compile the code using the SDCC compiler and download a hex file to the Z51F3220 MCU Module using the Z8051 OCD software.

Note: Debugging is not possible when using the SDCC compiler. The following procedure is intended merely as a guide to compiling and downloading the code to the MCU using the SDCC compiler.

1. Connect the Zilog Z8051 On-Chip Debugger (OCD) to the host PC's USB port.
2. Connect the other end of the 10-circuit cable connector to the programming connector (J10) on the Z8051 MCU Module. Pin 1 of the cable connector is indicated by a red stripe.
3. Connect the hardware as shown in Figure 4.

4. Power up the MultiMotor Development Board using the 24 VDC adapter included in the kit.
5. Download the AN0371-SC01.zip file and unzip it to the following path:

   <Z8051 Installation>\Z8051_<version_number>\samples
6. Browse to the location where the AN0371-SC01 file was installed and double-click the `build_sdcc.bat` batch file to build the project. An example of a typical file location is:

   `C:\Program Files\ZILOG\Z8051_2.2\samples\AN0371-SC01\build_sdcc.bat`
7. When the build is complete, the command window will prompt the user to press any key to continue. Proceed by pressing any key on the keyboard; as a result, a hex file will be created.
8. Launch the Zilog Z8051 OCD software.
9. If a *Disconnected* message is displayed by the debugger, unplug the power supply from the MultiMotor Development Board (P1) and then plug it back in. Ensure that the green LED (D2) is on, indicating power is applied to the board.
10. From the Debugger's **File** menu, select **Load Hex**. The Object File dialog will appear.
11. Click the **Browse…** button, then locate and open the `AN0371-SC01.hex` file in the following path:

    <Z8051 Installation>\Z8051_<version_number>\samples\Sdcc_out
12. In the **Object File** dialog, click the **Download** button. The Configuration dialog will appear.
13. Click the **Write** button to download the code to the MCU.
14. Unplug the power supply from the MultiMotor Development Board, then disconnect the Zilog Z8051 USB OCD Cable from P1.
15. Ensure that the RUN/STOP switch on the MultiMotor MCU Module is in the STOP position.
16. Connect the 24 VDC supply source to the MultiMotor Development Board.
17. Set the RUN/STOP switch on the MultiMotor MCU Module to RUN.
18. Set the direction of rotation of the motor by changing the position of the direction switch on the MultiMotor Development Board.

Refer to the the Z8051 On-Chip Debugger II User Manual (UM0270) or the Z8051 Tools Product User Guide (PUG0033) for additional information.

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 Z8051 MultiMotor MCU Module confirms a seamless startup 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: 3200 RPM
- 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 illuminates when the motor is stopped or a fault is detected

References

The following documents are associated with the Z8051 Series of Motor Control MCUs; each is available for download at www.zilog.com.

- MultiMotor Series Development Kit Quick Start Guide (QS0091)
- MultiMotor Series Development Kit User Manual (UM0262)
- Z51F3220 Product Specification (PS0299)
- Z8051 Tools Product User Guide (PUG0033)
- Z51F3220 Development Kit User Manual (UM0243)
- Z8051 On-Chip Debugger II User Manual (UM0270)

Appendix A. Schematic Diagrams

Figures 6 and 7 show the schematics for the Z8051 MultiMotor MCU Module.

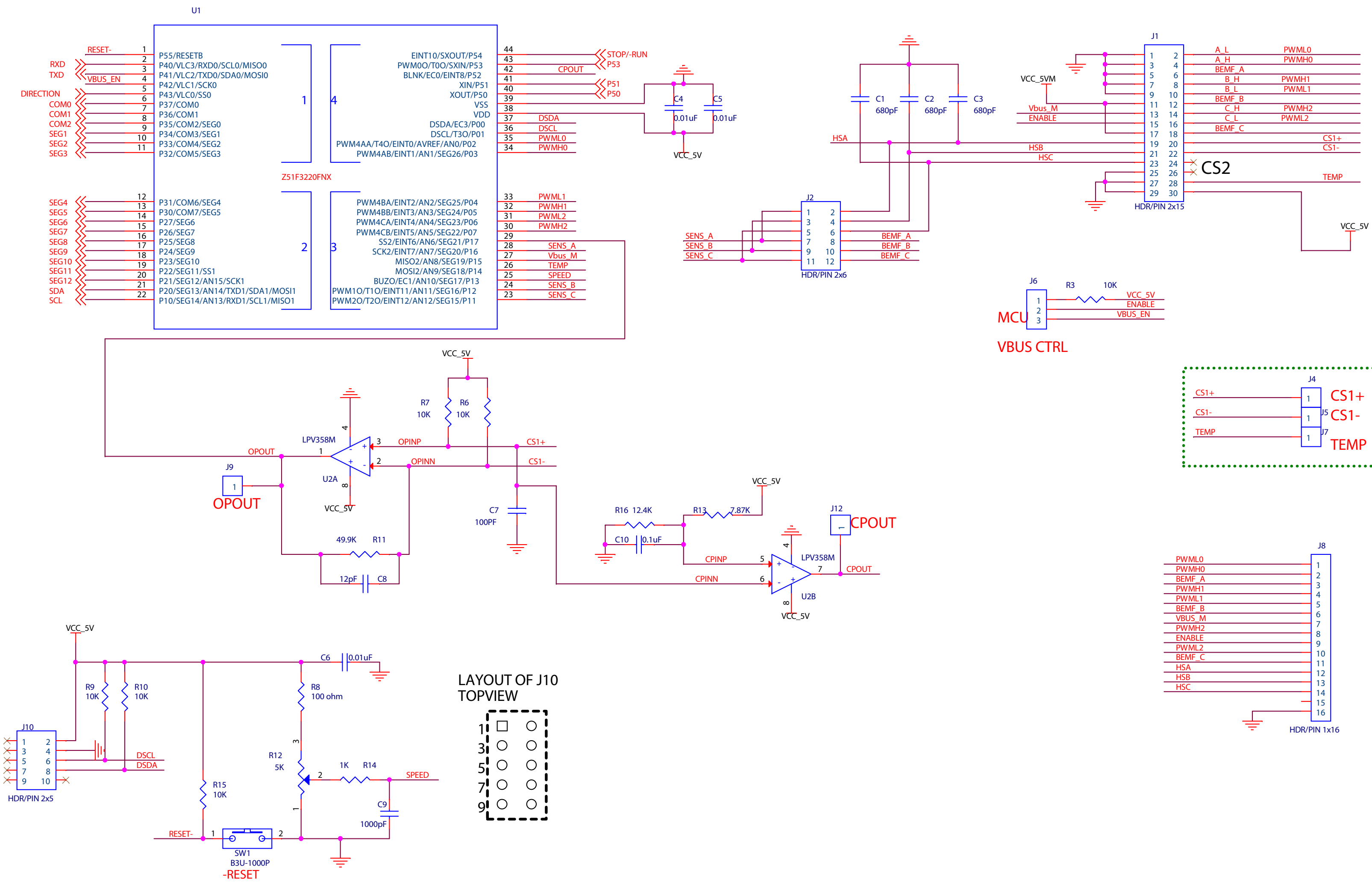

Figure 6. Z8051 MultiMotor MCU Module, # 1 of 2

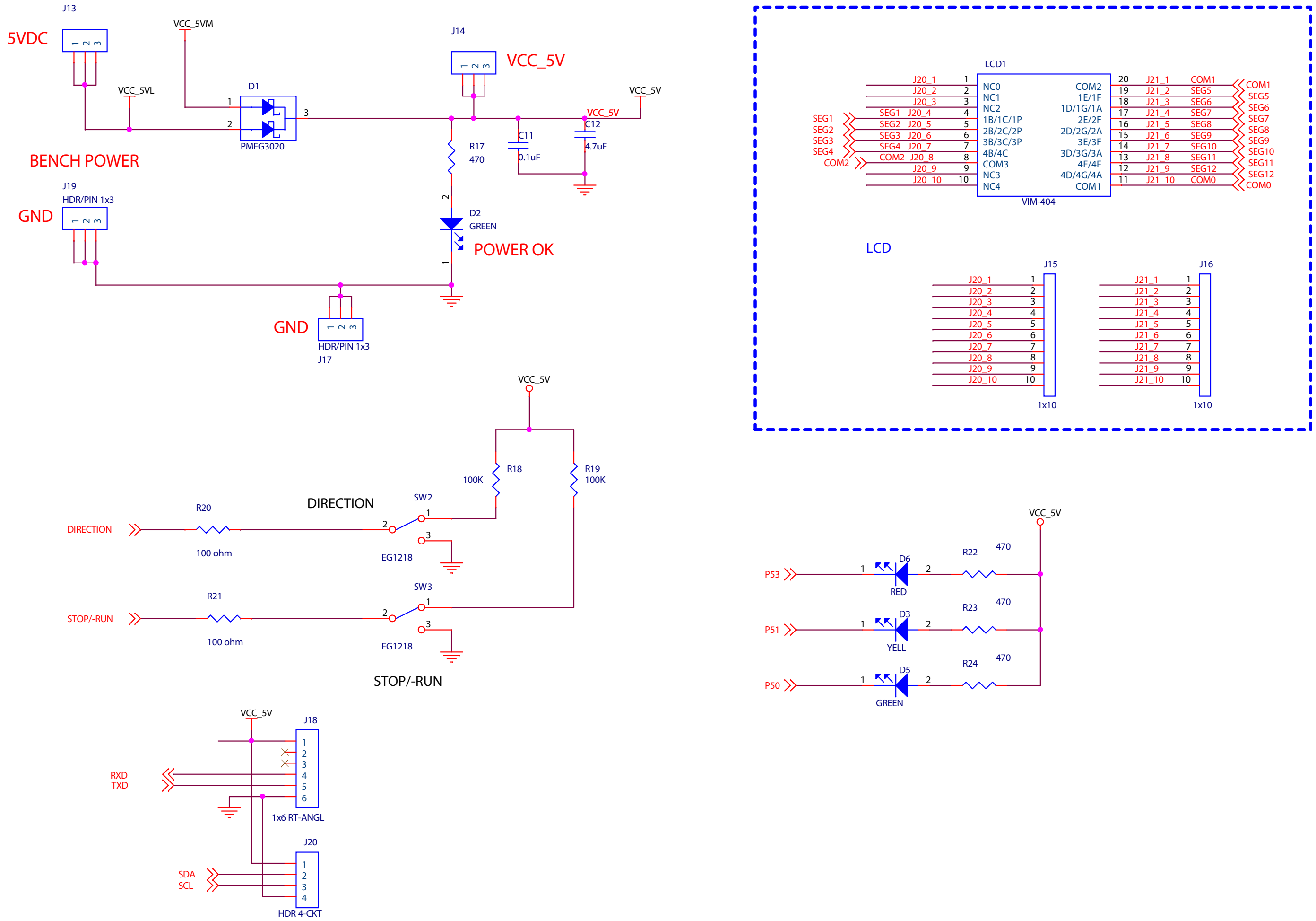

Figure 7. Z8051 MultiMotor MCU Module, #2 of 2

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

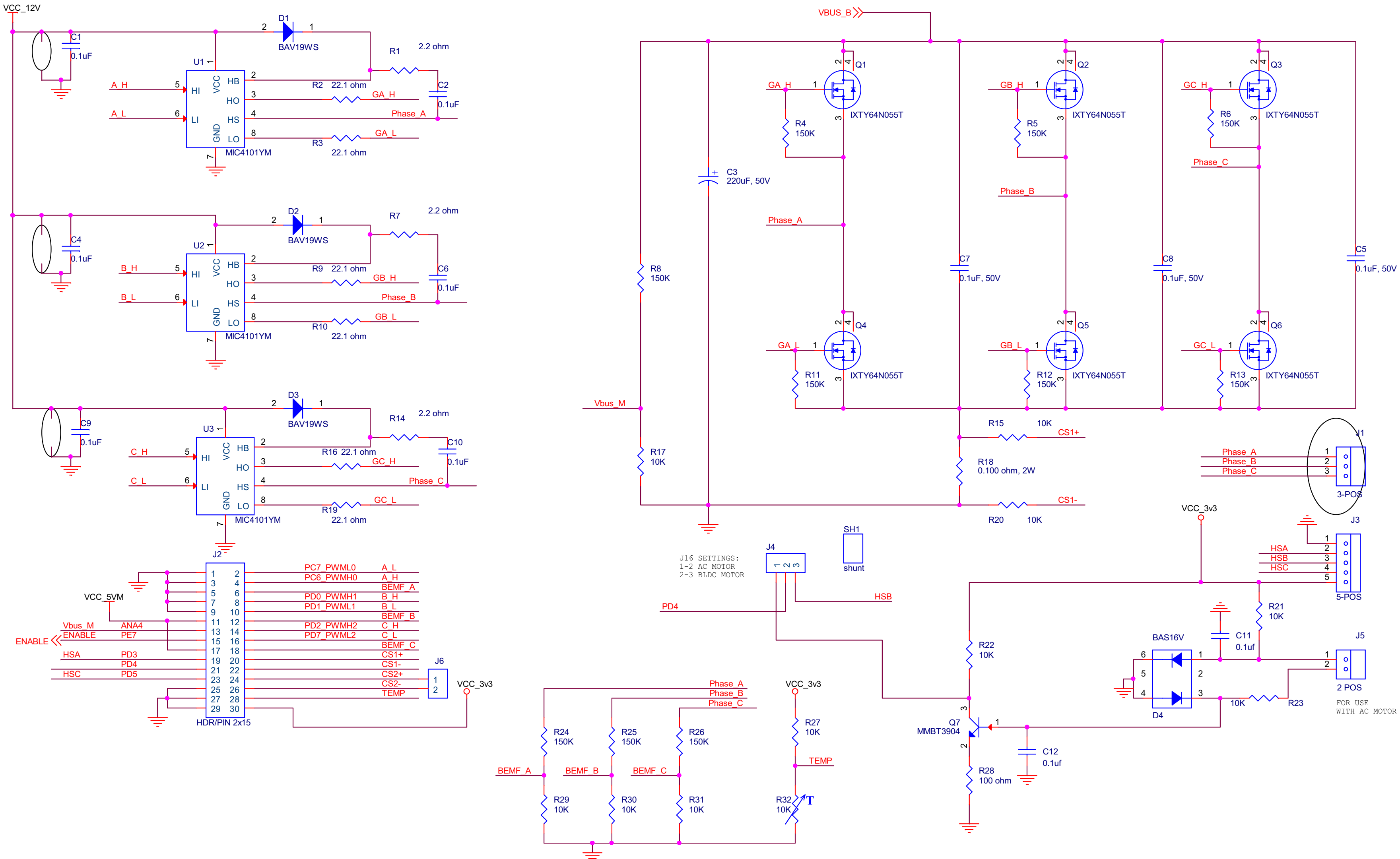

Figure 8. MultiMotor Development Board, #1 of 2

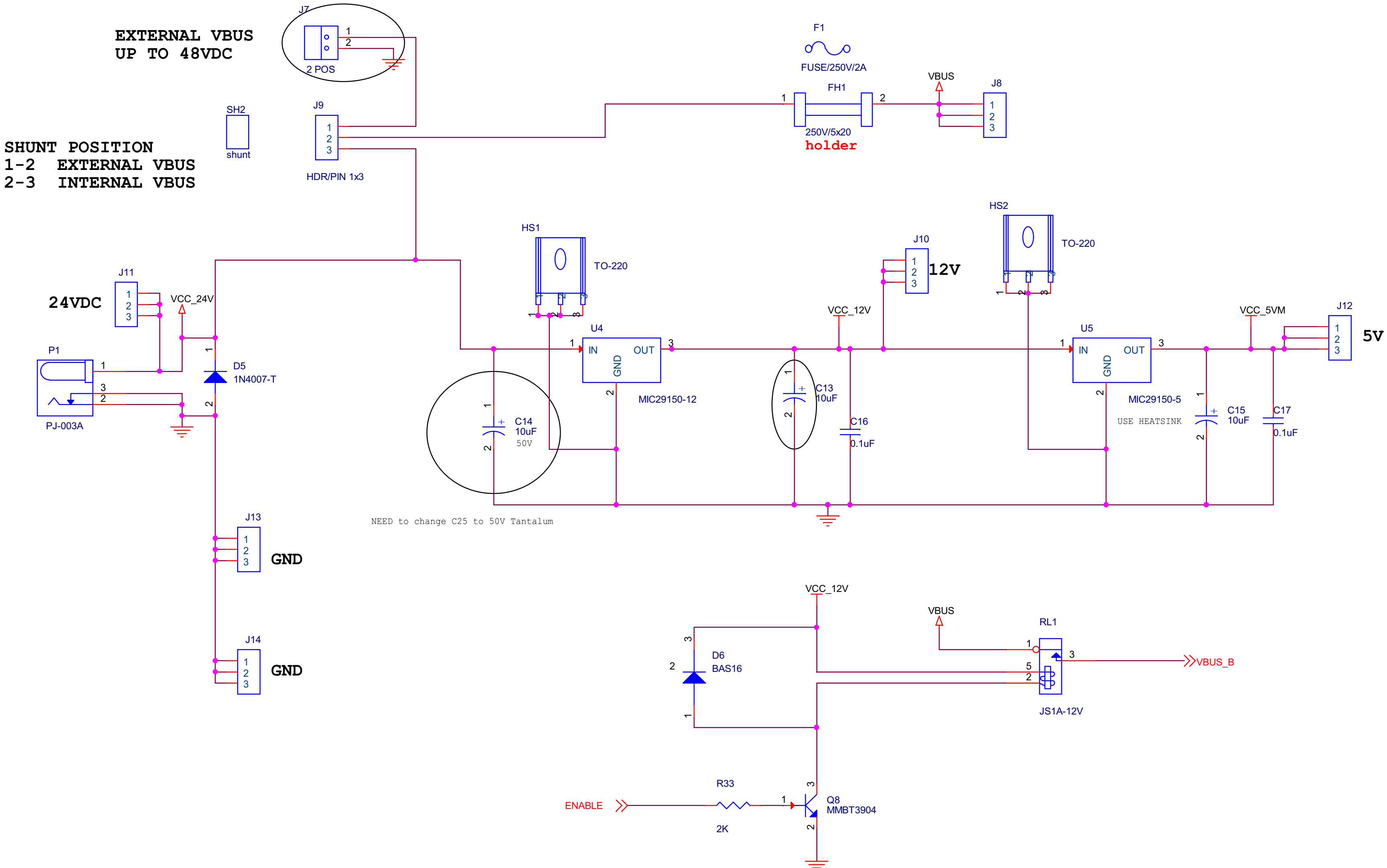

Figure 9. MultiMotor Development Board, #2 of 2

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.

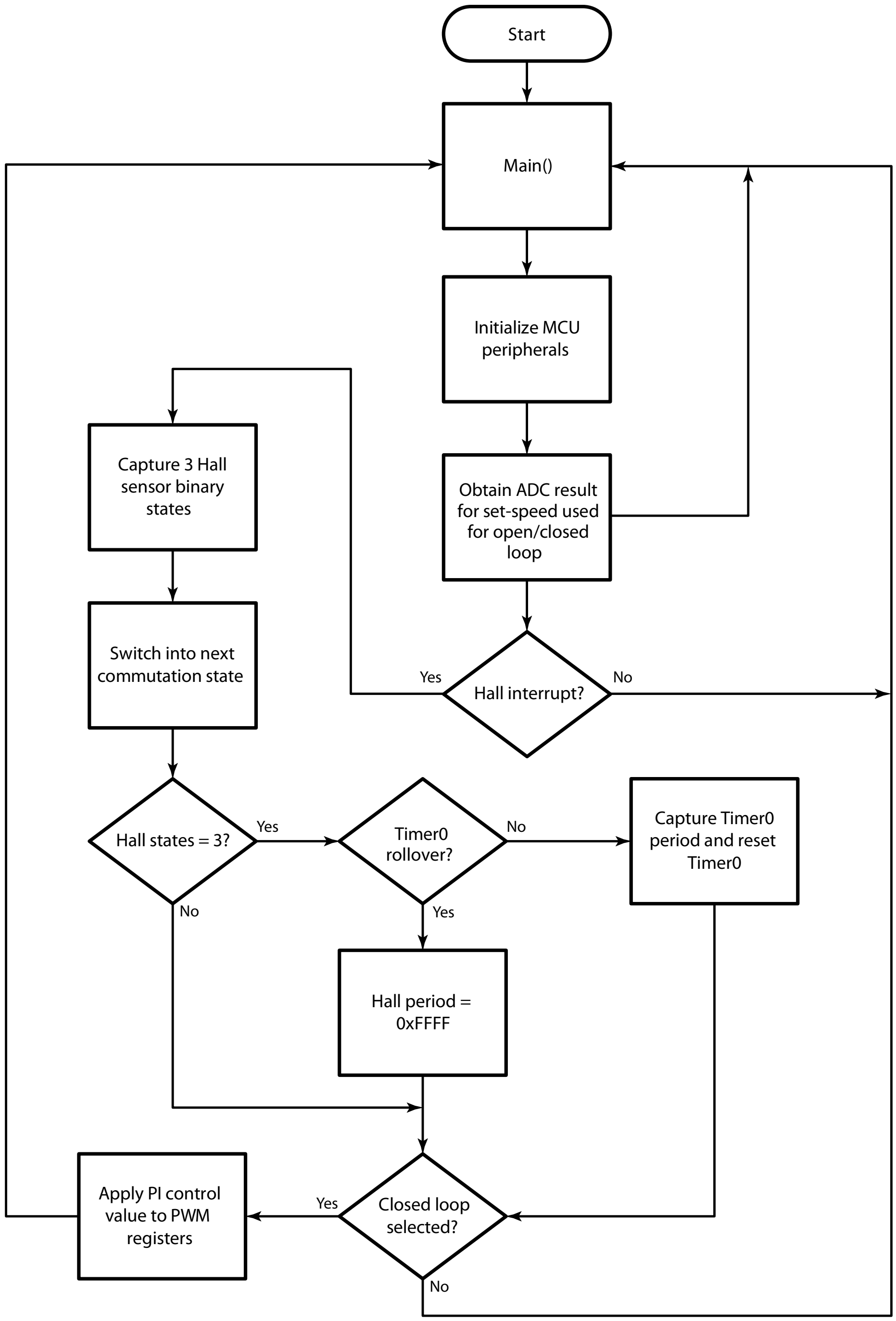

Figure 10. 3-Phase Hall Sensor BLDC Motor Control Application Flowchart

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