Application Note

Sensorless Brushless DC Motor Control with the Z16FMC MCU

MultiMotor Series

AN035303-1015

Abstract

This MultiMotor Series application note investigates the closed-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 a sensorless feedback control system using a Phase-Locked Loop with back-EMF sensing. Test results are based on using a MultiMotor Development kit equipped with a Z16FMC MCU module and a 3-phase, 24VDC, 30W, 3200RPM BLDC motor.

Note: The source code file associated with this application note, AN0353-SC01, was tested with version 5.0.1 of ZDSII for ZNEO MCUs. Subsequent releases of ZDSII may require you to modify the code supplied with this application.

Features

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

- Smooth S-curve motor start-up 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
- 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 20MHz. 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 4KB of internal RAM provides storage of data, variables and stack operations.

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

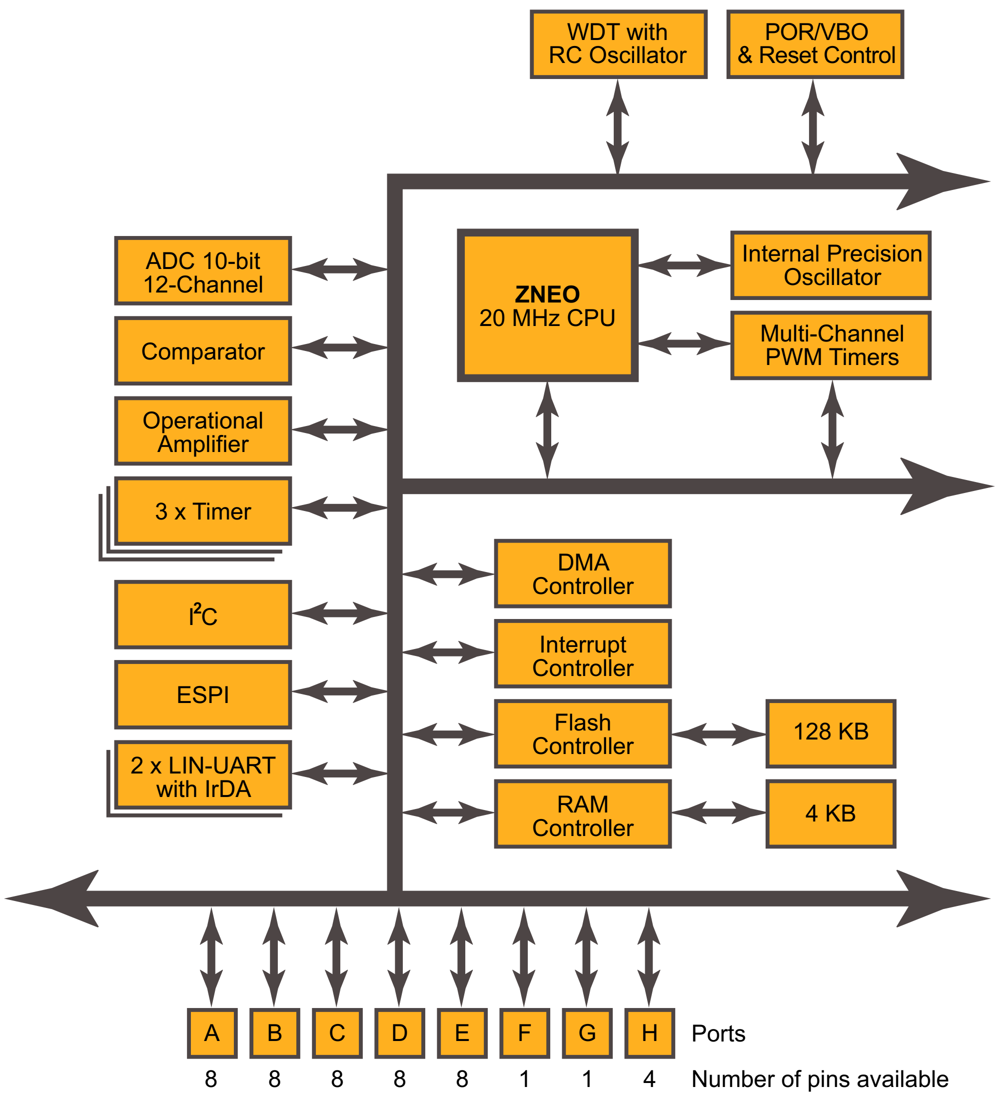

Figure 1. The Z16FMC MCU Architecture

In each of the Z16FMC products, the novel device architecture allows for realization of a number of enhanced control features:

- Time Stamp for Speed Control

- Integrated Operational Amplifier
- Multi-Channel PWM Timer

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 the output driving pulses can be synchronized and adjusted to properly regulate motor speed. An alternative approach based on the Z16FMC MCU eliminates the requirement for this comparator by instead employing an analog-to-digital converter (ADC) in conjunction with a timer. In such a scenario, the ADC samples the AC line voltage, with the timer running in the background.

When the ADC samples the line voltage's zero crossing, it reads the timer count and writes the result to a register. As a result, the timers are cued for the output Pulse Width Modulation (PWM) pulses to efficiently regulate the speed of the motor. This time stamp approach results in a very simple and cost-effective solution for smooth operation of the motor in a steady state.

Integrated Operational Amplifier

Appliance controllers almost invariably monitor motor speed by sensing the 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. With this process, an external operational amplifier is often used to convert the current signal to a voltage signal; the ADC next samples the voltage signal and outputs the result to the processor. The processor then synthesizes the PWM outputs to control motor speed.

In the case of the Z16FMC MCU, an on-chip integrated operational amplifier eliminates the requirement for an external component, thereby reducing overall system cost.

Multi-Channel PWM Timer

The Z16FMC MCU features a flexible PWM module with three complementary pairs – or six independent PWM outputs – supporting deadband operation and fault protection trip input. These features provide multiphase control capability for a variety of motor types and conduct safe operation of the motor by ensuring immediate shutdown of the PWM pins during a fault condition.

Theory of Operation

In a brushless DC motor, the rotor is comprised of permanent magnets, while the stator windings are similar to those in poly-phase 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 sensor-based control applications, the Hall elements are integrated, and are used to detect the position of the rotor for drive synchronization. In contrast, sensorless control employs the detection of back-EMF signals, which are generated (induced) by specific phase windings to synchronize the timing of the control loop.

A block diagram of the BLDC motor control system based on the Z16FMC MCU is shown in Figure 2. At any given instance in a 3-phase commutation arrangement, only two phases are energized. The back-EMF voltage is, in turn, 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 earlier, the innovative time stamp feature of the Z16FMC MCU provides for robust, efficient implementation of this critical sensing function without the requirement for an additional comparator.

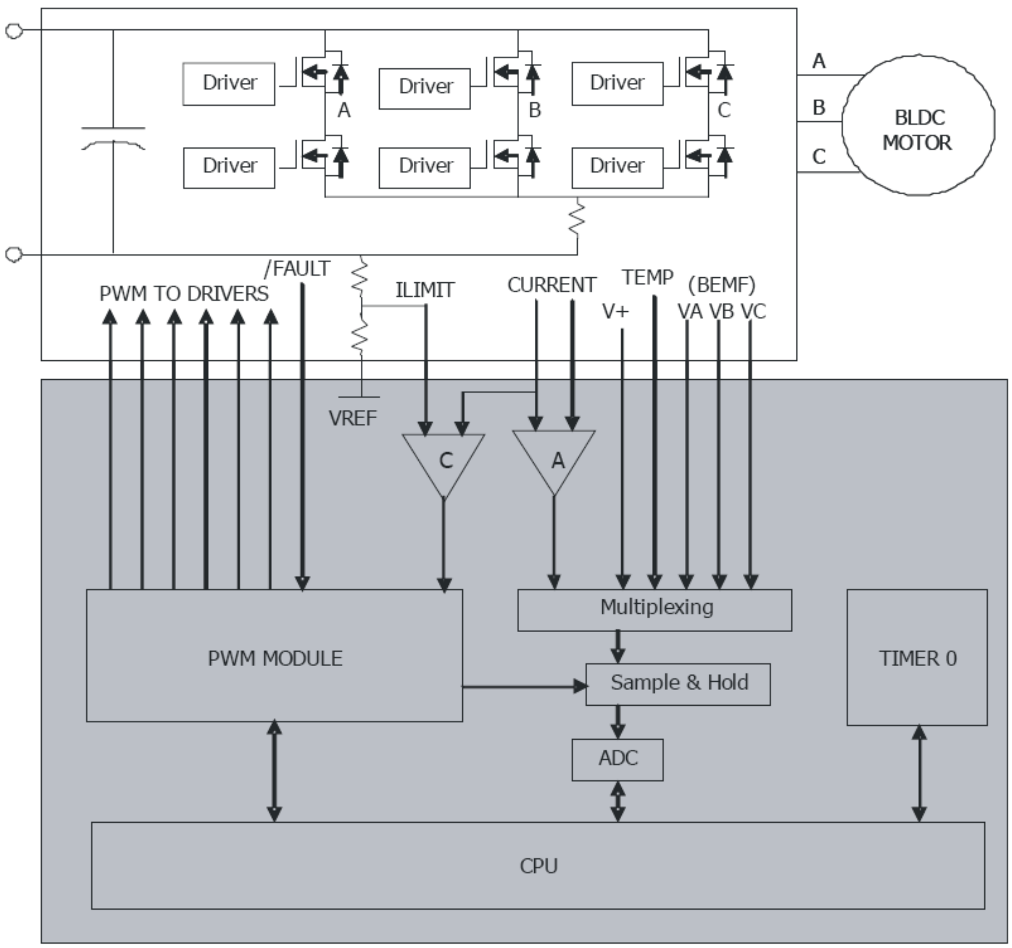

Figure 2. . A 3-Phase BLDC Motor Control System

The algorithm for back-EMF sensing is based on an implementation of a Phase-Locked Loop (PLL), which is described in Appendix C. Back-EMF Sensing Phase-Locked Loop on page 16. This algorithm is especially advantageous during startup, resulting in a very smooth increase in the motor speed, as well as a nearly-instantaneous reversal of direction of the rotation on command, as outlined below.

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

During normal operation following the start-up period, phase torque/current mode control is achieved with a 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 PWM commutation signal period(s).

As discussed earlier, another key feature of the Z16FMC 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. Oscilloscope-generated waveforms representing this sequence of events are shown in Figure 3.

In conjunction with the integrated on-chip hardware blocks, the 3-phase BLDC motor control software developed for this application 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 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.

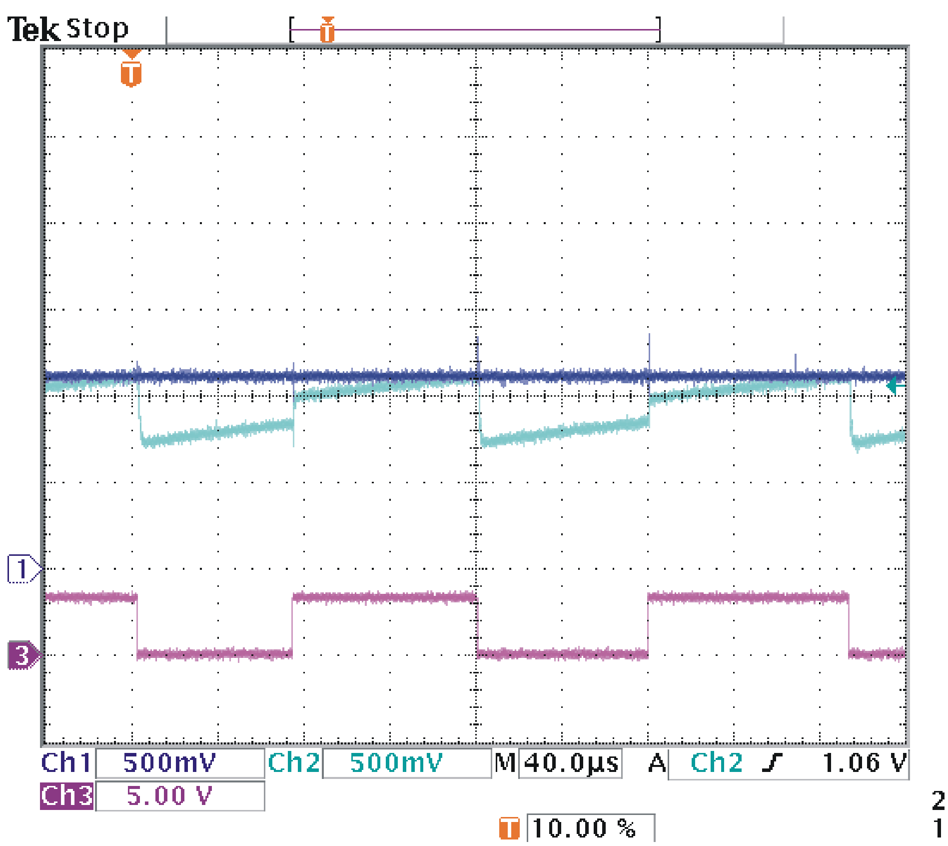

Figure 3. Cycle-By-Cycle Shutdown

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.

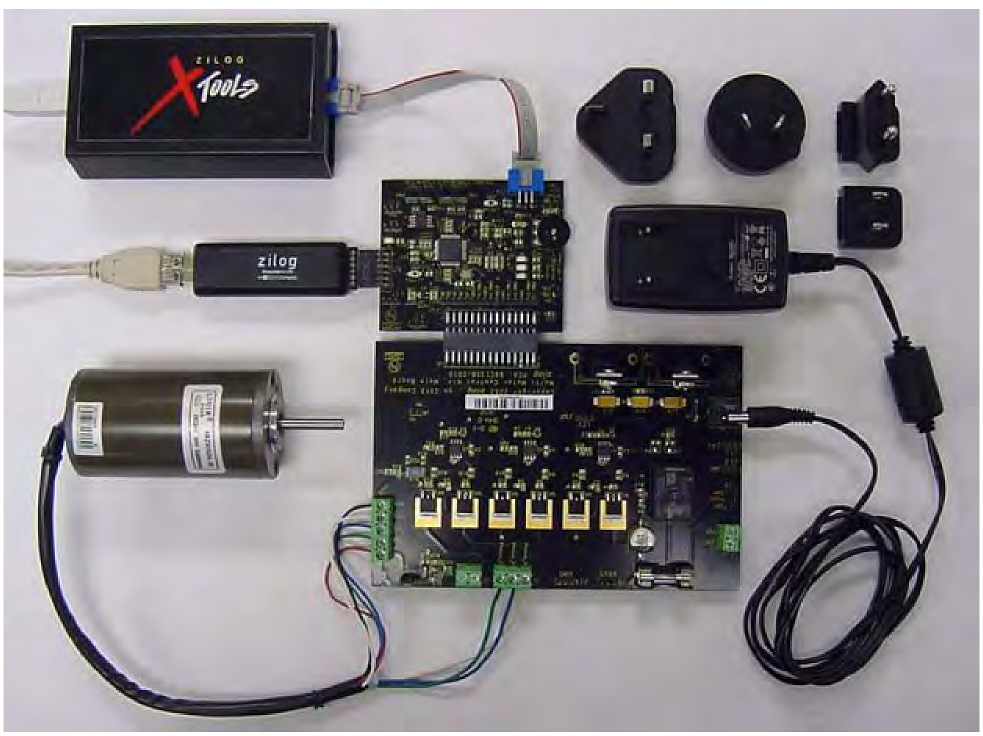

Figure 4. The MultiMotor Development Kit with Z16FMC MCU Module and SmartCable

Figure 5 displays 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 ZDS II – 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 24 VDC adapter included in the kit.
5. Open the AN0353-SC01 project in ZDS II for ZNEO.

6. From the `main.c` source file, choose the following mode for the Motor Control application:

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

7. Compile the application and download the code to the Z16FMC MultiMotor MCU Module.
8. In ZDSII, stop the Debug Mode. Unplug the power supply from the MultiMotor Development Board, then disconnect the Opto-Isolated USB Smart Cable.
9. Ensure that the RUN/STOP switch on the MultiMotor MCU Module is in the STOP position.
10. Connect the 24V DC supply source to the MultiMotor Development Board.
11. Set the RUN/STOP switch on the MultiMotor MCU Module to RUN.
12. Additionally, observe the following points:
    - If Speed Mode is selected, the speed of the motor can be varied by adjusting the potentiometer on the MultiMotor Development Board.
    - If Torque Mode is selected, the motor speed is decreased with application of force on the shaft of the motor.
    - The direction of rotation of the motor is set 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, sensorless, 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 Modules confirms a seamless start-up of the motor from an idle mode to full operational speed, a safe 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
- Two methods of controlling the motor:
  - Manually using the Stop/Run & Direction switches and the speed pot on the 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

Summary

This application note describes the closed-loop control of a sensorless BLDC motor using the advanced on-chip integrated features of the Z16FMC MCU. The software algorithm implemented in this application demonstrates how a three-phase BLDC motor is operated with a minimum set of peripherals using the ADC module for BEMF detection and a Phase Lock Loop for PI speed control. With this implementation, the need for an open loop start-up ramp was eliminated without sacrificing smooth motor start.

The results of this application confirm why the Z16FMC MCU is ideally suited for sensorless brushless motor control applications. The Z16FMC MCU's features, along with the powerful ZNEO 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 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)
- 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)
- Three-Phase Hall Sensor BLDC Driver Using The Z16FMC MCU (AN0356)
- Implementing a Data Logger with Spansion SPI Flash (AN0360)

Appendix A. Schematic Diagrams

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

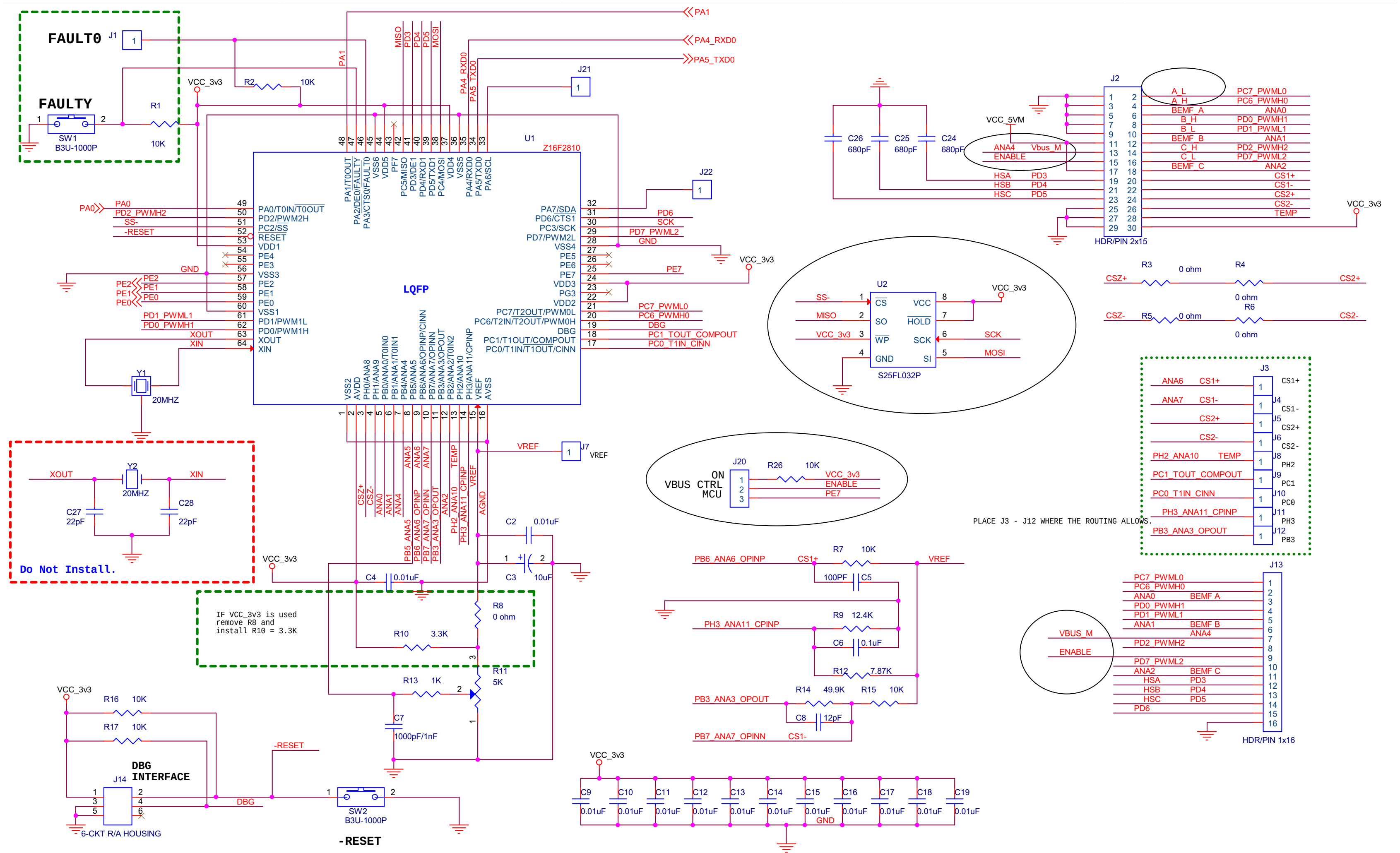

Figure 6. Z16FMC MultiMotor MCU Module, #1 of 2

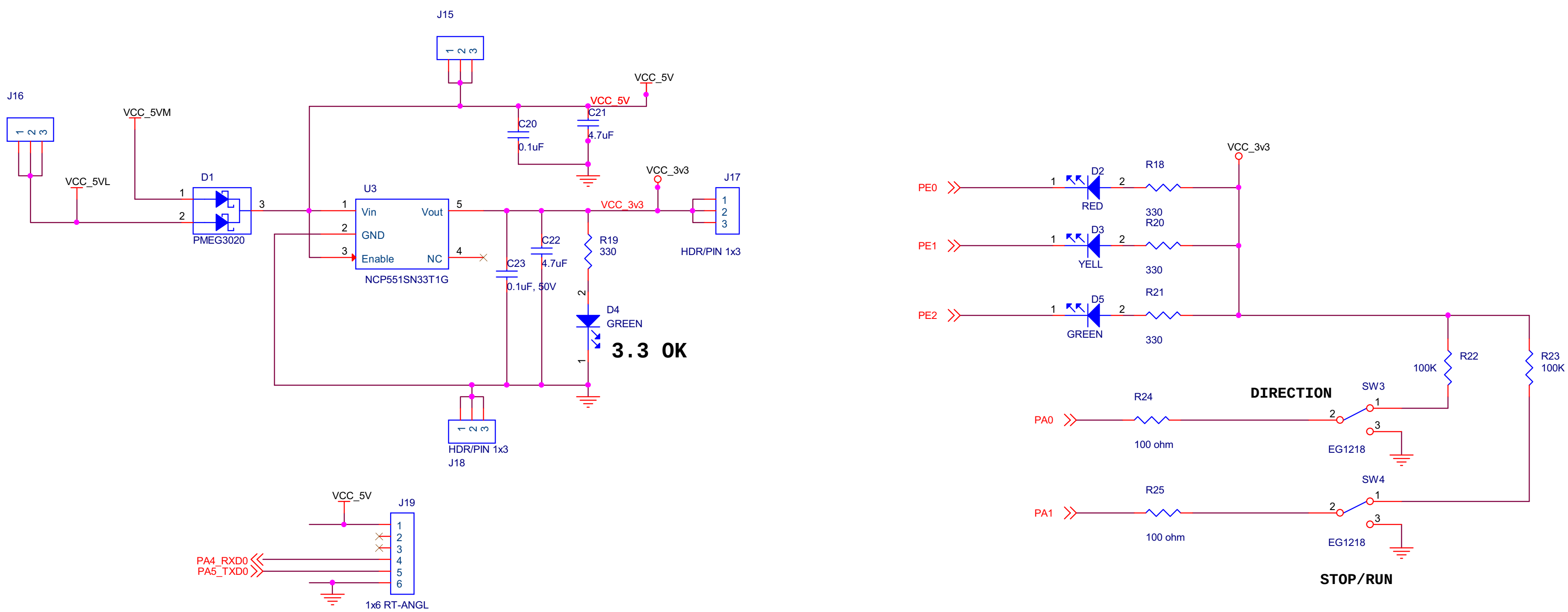

Figure 7. Z16FMC MultiMotor MCU Module, #2 of 2

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

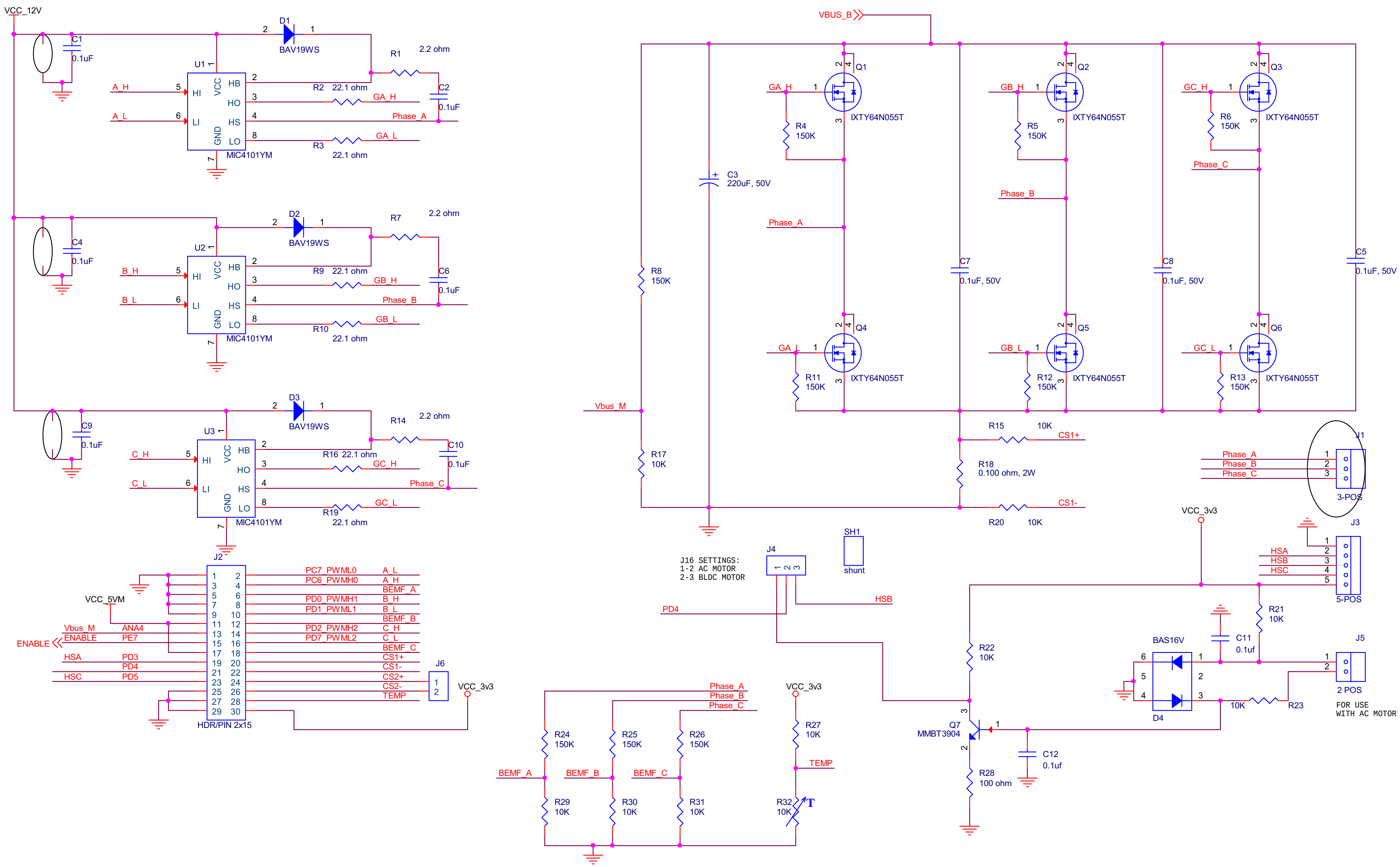

Figure 8. MultiMotor Development Board, #1 of 2

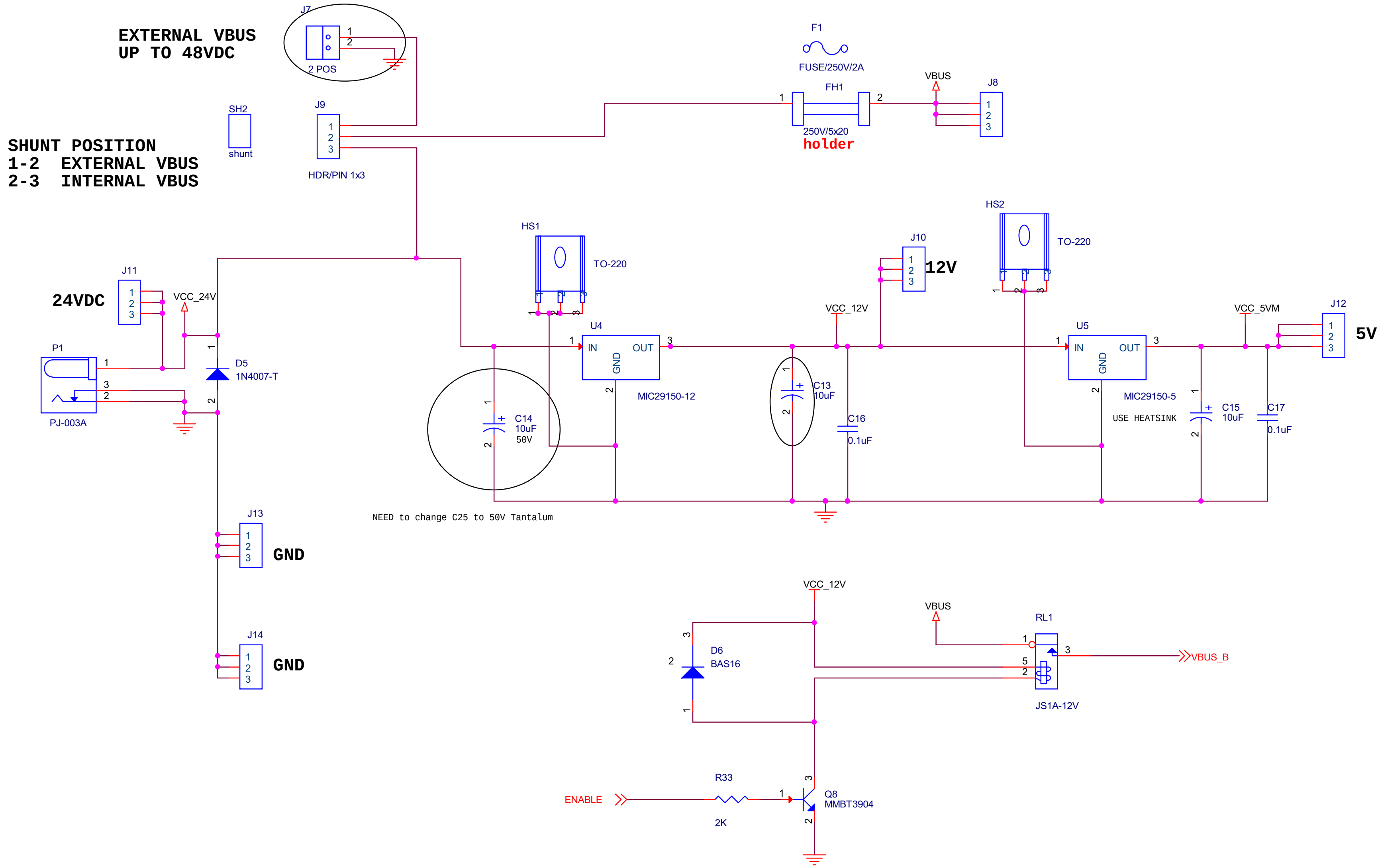

Figure 9. MultiMotor Development Board, #2 of 2

Appendix B. Flowcharts

This appendix displays flow charts that diagram the Main Function and the Read and Write APIs.

Figure 10 shows the main control loop.

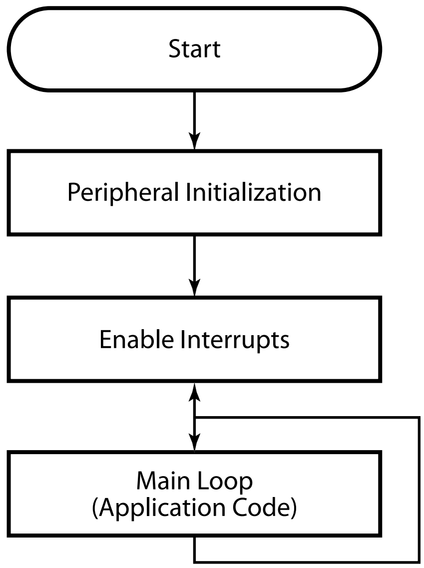

Figure 10. Initialization and Application Code Space

The back-EMF sensing loop is shown in Figure 11.

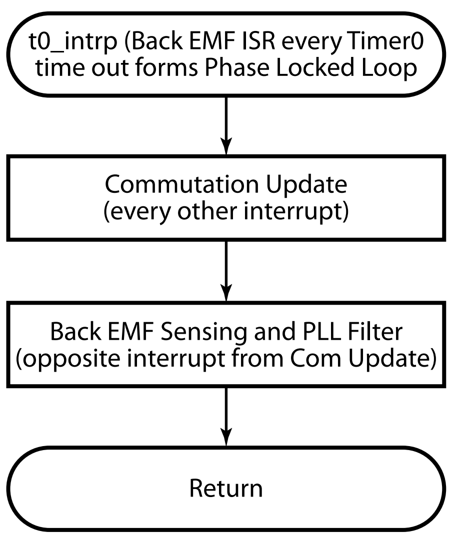

Figure 11. Initialization and Application Code Space

A flow chart of the PWM loop is shown in Figure 12. This PWM loop can also be used for specific application code, such as communications or additional user interfaces.

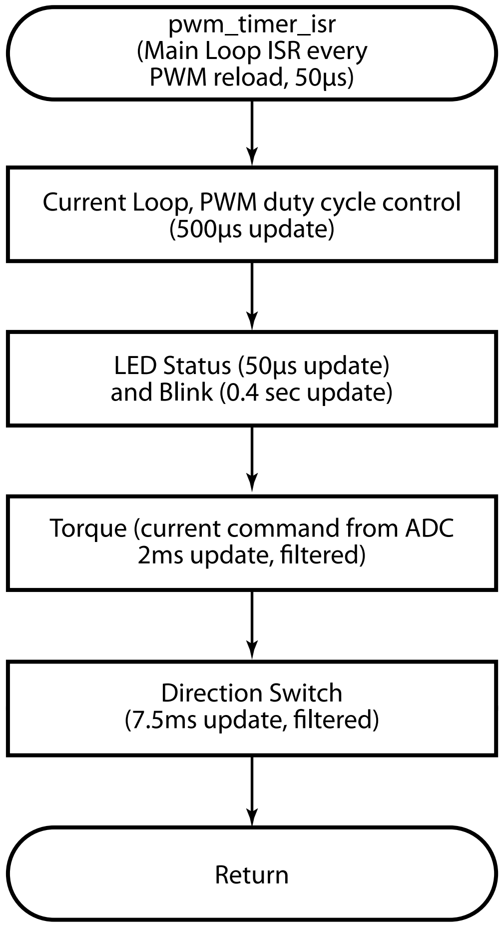

Figure 12. Current Loop and Timed Housekeeping

Appendix C. Back-EMF Sensing Phase-Locked Loop

The Phase-Locked Loop back-EMF algorithm, implemented to provide a smooth start-up of the motor, is shown in Figures 13 and 14. Additional details about the specific formulas in these figures are shown in Table 1; a description of these calculations follows.

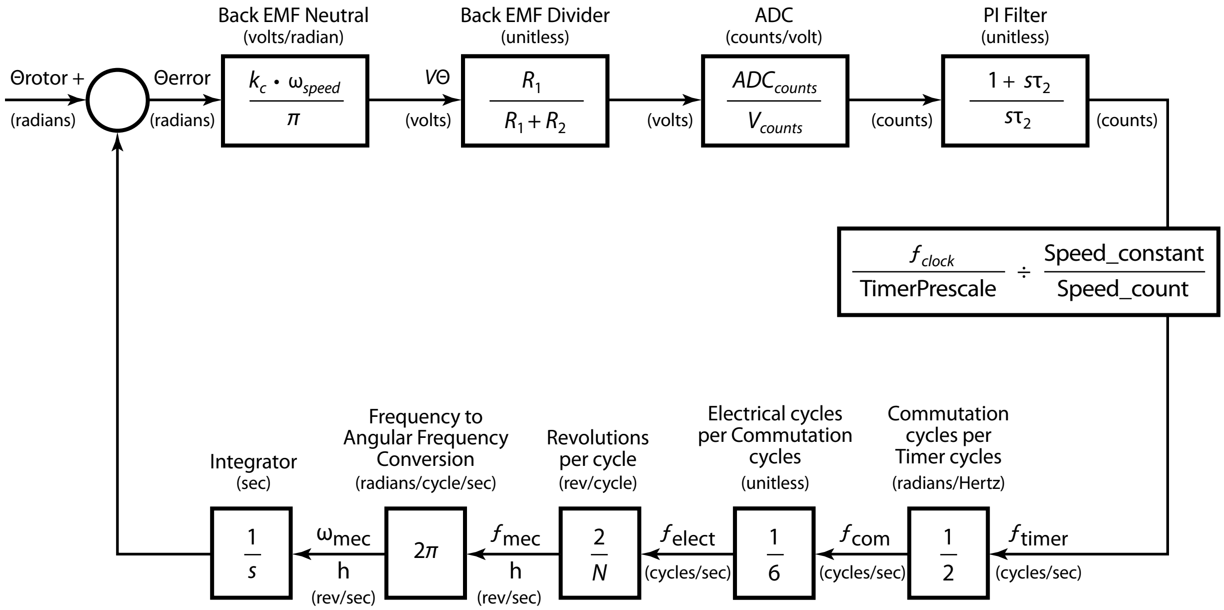

Figure 13. Back-EMF Sensing Using the Phase-Locked Loop Algorithm

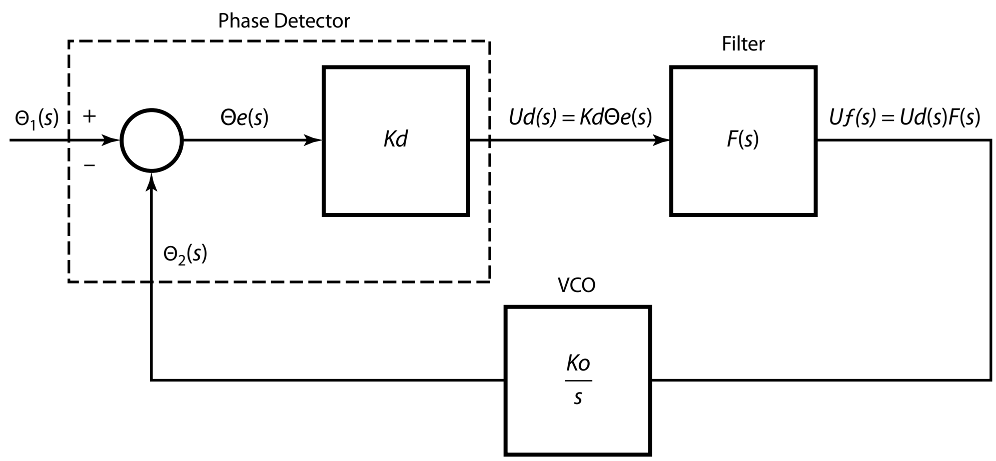

Figure 14. Proportional Integral (PI) Filter Representation for Back-EMF Sensing

Table 1. 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:

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

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

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