

Abstract

This application note discusses the control of a 3-phase brushless BLDC motor in Sinusoidal PWM Modulation mode using Zilog's ARM Cortex M3 Z32F128 MCU with an 80MHz internal clock and 128KB of Flash memory. The Z32F128 microcontroller, a member of Zilog's ZNEO32! 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 and on-the-fly speed/torque control, as well as ease of firmware development for customized applications.

This document also discusses ways in which to implement sinusoidal PWM modulation and phase-angle synchronization with Hall sensor feedback. Test results are based on using a MultiMotor Development kit equipped with a Z32F128 MCU module and a 3-phase, 24VDC, 30W, 3200RPM BLDC motor with internal Hall sensors.

► **Note:** The source code file associated with this application note, [AN0380-SC01.zip](#), is available free for download from the Zilog website. This source code has been tested with Keil μ Vision version 5.14.0.0. Subsequent releases of Keil μ Vision may require you to modify the code supplied with this application note.

Program Features

The power-saving features of this Z32F128 application code include:

- Smooth motor start-up with reduced starting current
- 3-Hall sensor feedback sinusoidal PWM modulation
- Microcontroller-based overcurrent protection
- Adjustable speed and current (frequency and sine magnitude)
- 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
- Non-volatile data logger function to record motor running conditions

Figure 1 shows a block diagram of the Z32F128 MCU architecture.

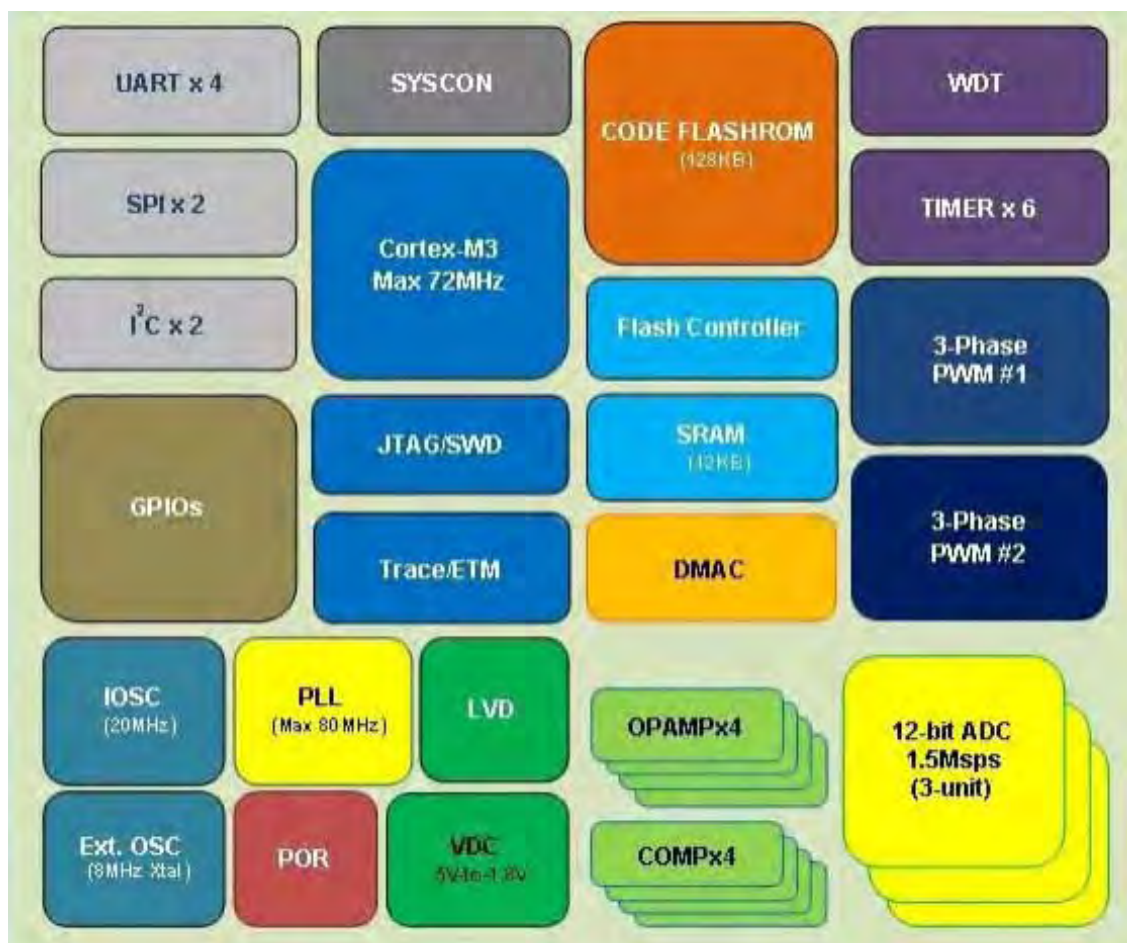


Figure 1. The Z32F128 MCU Architecture

Discussion

The Z32F128 microcontroller upon which this Sinusoidal PWM driver has been conceived has a high performance core, executing instructions at up to 80MHz internal clock frequency.

Up to 128KB internal Flash memory is accessible by the CPU, 32 bits at a time, to improve processor throughput. Up to 12KB of internal RAM provides storage of data, variables and stack operations.

PWM sinusoidal operation has certain advantages over block-commutated PMSM motor driving approaches, most notably its lower electrical and lower acoustical noise signatures. By comparison, the block commutation method causes harsh current transitions through the PMSM motor coils by turning the phase windings of the motor on and off between commutations. The PWM sinusoidal method does not create these harsh current

transitions through the motor coils, because the current and phase voltages are sinusoidal in nature. Motor operation via sinusoidal PWM control can run the motor at higher efficiencies than block commutated motors.

Because of the advantages of a PWM sine driver scheme's attributes, PWM sinusoidal operation may be a better option for certain applications in which the life of ripple capacitors and ball bearings are concerns, as well as electrical noise.

Sinusoidal PWM driving schemes can be used to drive either PMSM- or BLDC-type motors, however, to take advantage of a sinusoidal driving scheme, a PMSM-type motor is likely to show the best results due to its sinusoidal wound phases which also causes the motor to run virtually cogging-free with a minimum amount of ripple torque.

In each of the Z32F128 products, the novel device architecture allows for the realization of the following enhanced control features; each is described in this section.

- Time stamp for speed control
- Integrated operational amplifier
- Highly configurable multichannel PWM timer with interrupt capabilities on period and duty cycles
- Overcurrent shut down

Time Stamp for Speed Control

The Capture feature of the 16-bit timers can be used to take a time stamp of the Hall sensor's electrical timing periods. Upon a predefined Hall state, the asynchronously operating timer is read and its value is compared against a calculated speed reference value using PI closed loop control.

Integrated Operational Amplifier

Appliance controllers almost invariably monitor motor speed by sensing current through the motor 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. For the Z32F128 MCU, an on-chip integrated operational amplifier eliminates the requirement for an external component, thereby reducing overall system cost.

Multichannel PWM Timer

Each Z32F128 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 ensure safe operation of the motor by providing 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 polyphase motors.

Generally, there are two methods for determining motor position and speed: sensed control and sensorless control. In sensor-based control applications, the Hall elements are integrated into the motor and used to detect the position of the rotor for drive and sine wave synchronization. In contrast, sensorless control employs the detection BEMF which is generated by the phase windings in the non-energized state. Both methods use the feedback information to synchronize the rotor and stator fields at 90 degree to each other.

An inverter bridge consisting of six N-channel 55V/64A IXYS MOSFETS is used to drive the PWM sine generated currents through the BLDC motor windings, as shown in Figure 2.

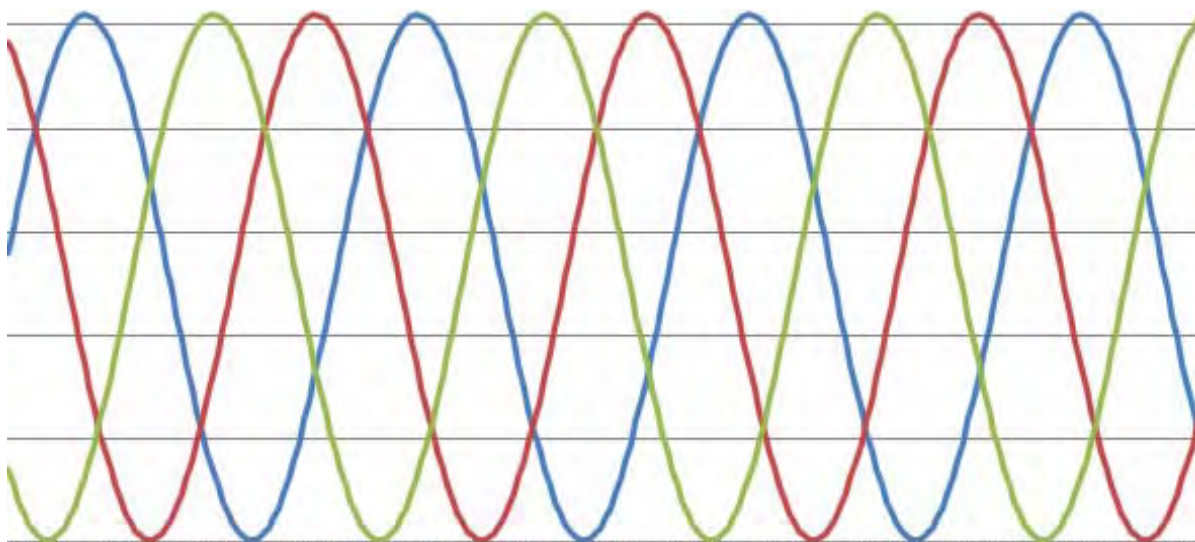


Figure 2. PWM-Generated Sinusoidal Currents In The Motor Windings

The algorithm for Hall sensing is based on an implementation using three I/O ports which are configured for an *interrupt on edge* change of the Hall sensor's signals. One of the advantages of using Hall sensors is that the angular position of the motor is known upon startup of the motor, therefore minimizing erratic start-up behavior and the need for a start-up ramp until BEMF zero crossings are detected. The PWM duty cycle present at the motor windings then produces the torque to start the motor. The rotating motor generates the Hall signals that vector into a single I/O service interrupt routine, which determines the next commutation state to energize the next motor phase.

Another advantage of using Hall sensors as opposed to BEMF sensing is that under sudden and strong load changes, the information of the commutation angle is not in jeopardy of becoming lost. In sensorless feedbacks, an extreme load increase can cause an inductive spike – which results from the stored magnetic energy of the previously turned off phase –

to become wide enough to suppress the BEMF information. As a result, the commutation angle information is lost and could cause the motor to stall.

The Hall sensor's angular position provides the information to energize all three phase voltages at the correct commutation angle. As opposed to trapezoidal or block commutation, in which two of the three phases are energized for each commutation step, the sinusoidal commutation requires all three phases to be energized at any given time, which involves more extensive and sophisticated methods to detect the BEMF.

To save computation time, the firmware implements a look up table in which the precalculated sine values are stored. The PWM timer interrupt service routine interrupts every 50µs and is used to fetch the sine values from the sine table and update the PWM sine frequency for Phase A, Phase B, and Phase C. This method provides very regular time intervals to update the sine frequency and scaling of the sine magnitude for all three phases. Right before exiting the PWM timer interrupt service routine, these three PWM channels are updated with the new PWM modulation values.

For this application, a PWM timer frequency of 20kHz was chosen to minimize linear switching power losses in the MOSFETs, and to be out of the audible noise range.

PWM Frequency Calculations

Using every value in the 256 sine array, the frequency is:

$$\frac{1}{(\text{PWM period} \times 256)} = \frac{1}{50\mu\text{s} \times 256} = 78.125\text{Hz}$$

If every second sine value is used instead, then the frequency is effectively doubled and becomes:

$$\frac{1+n}{(\text{PWM period} \times 256)} = \frac{2}{50\mu\text{s} \times 256} = 156.25\text{Hz}$$

In this second equation, the numerator represents the 1 + nth number of an offset to the array elements; the larger the numerator, the higher the sine frequency. A better way of obtaining a wider sine frequency range and resolution of the sine wave is to use a 16-bit interpolating table index of which only the upper byte is used to fetch the next PWM sine value from the look-up table. Depending on the frequency demand, the values of the sine index can change with higher granularity, hitting each sine array value more or less times while the sine index continuously rolls over. Using this method, the lowest period using a 16-bit pointer to a 256-element sine table is:

$$65535 \times 50\mu\text{s} = 3.277 \text{ seconds}$$

Because of the interpolating index method, the values in the numerator can be small, changing only in fractions to achieve higher frequencies.

Assuming the sine frequency is 60Hz, the offset value for the sine table pointer is:

$$\text{SineIndexOffset} = \frac{60 \times 65536}{20000} \approx 196$$

The resolution of the generated sine wave is a function of the sine frequency. Using the equation above, the calculated offset value is, in essence, the speed value, to be integrated to form the theta value.

Phase-Angle Adjustment

Hall sensor interrupts are generated six times – once every sixty degrees – therefore providing data about the rotor position which is used to synchronize the motors stator field at 90 degrees to the motors rotor field.

[Figure 2](#) on page 4 illustrates the generation of three 120-degree shifted sine waves based on values from a look-up table (LUT) which are then reconstructed in the PWM interrupt service routine.

Speed Calculations

The angular period times of the rotor are captured every one-sixth of an electrical commutation, wherein Timer0 represents the number of timer ticks. These timer ticks are then compared against the demand speed coming from a potentiometer – also represented in timer ticks – and processed in a PI closed loop to adjust the sine magnitude of the motor. If the motor is operated with open-loop speed control, the speed demand together with the Hall period information is used to calculate the demand frequency and magnitude values to change the motor frequency.

The angular speed calculation is:

$$\omega = \frac{d\phi}{dt}$$

In this equation, $d\phi$ is the angular displacement and dt is the time taken for the angular displacement.

The position information is provided by the Hall sensor binary state, and the time between angular positions is measured by Timer0 timer ticks.

The RPM of a sinusoidal operated motor is calculated using the following equation:

$$\text{RPM} = \frac{120 \times f}{N}$$

In the above equation, N is the number of pole pairs. By substituting for f , the following equation is obtained:

$$\text{RPM} = \frac{\left(120 \times \frac{\text{table index value}}{\text{PWM period} \times \text{table size}}\right)}{N}$$

Figure 3 illustrates how these calculations can influence the PWM sine operation of a 3-phase BLDC motor.

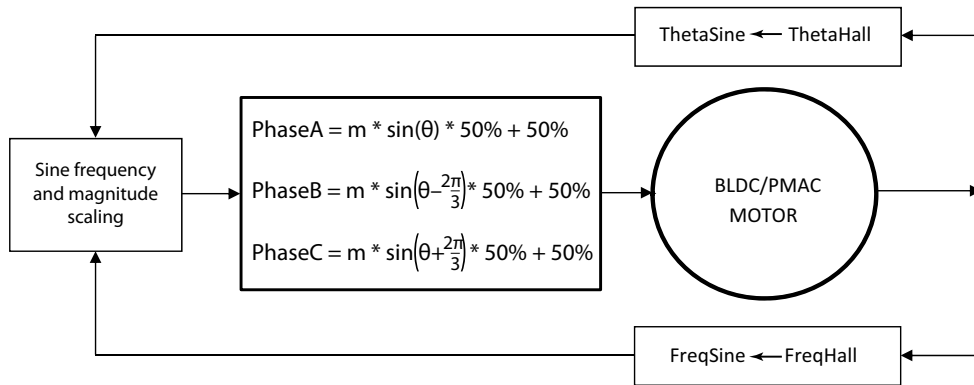


Figure 3. Simplified PWM Sine Operation of a BLDC Motor

The hardware used to realize the sinusoidal PWM motor driver approach discussed above is shown in the block diagram in Figure 4.

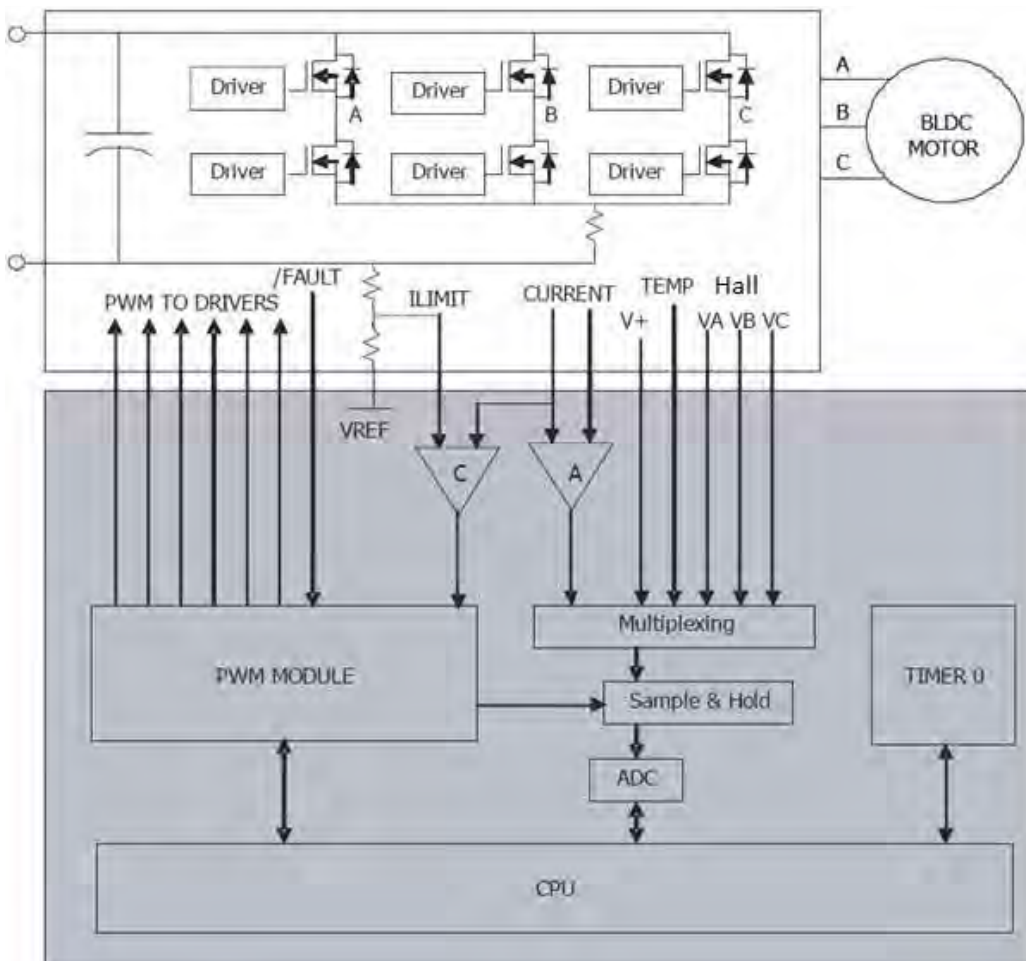


Figure 4. 3-Phase BLDC Motor Control System

Overcurrent Protection

Currents can reach excessive amounts during startup, load changes, or catastrophic failures, for which a motor and electronics must be protected. A key feature of the Z32F128 MCU is the direct coupling of the on-chip integrated comparator to the PWM module to enable a fast, cycle-by-cycle shutdown during an overcurrent event. Oscilloscope-generated waveforms representing this sequence of events are shown in Figure 5.

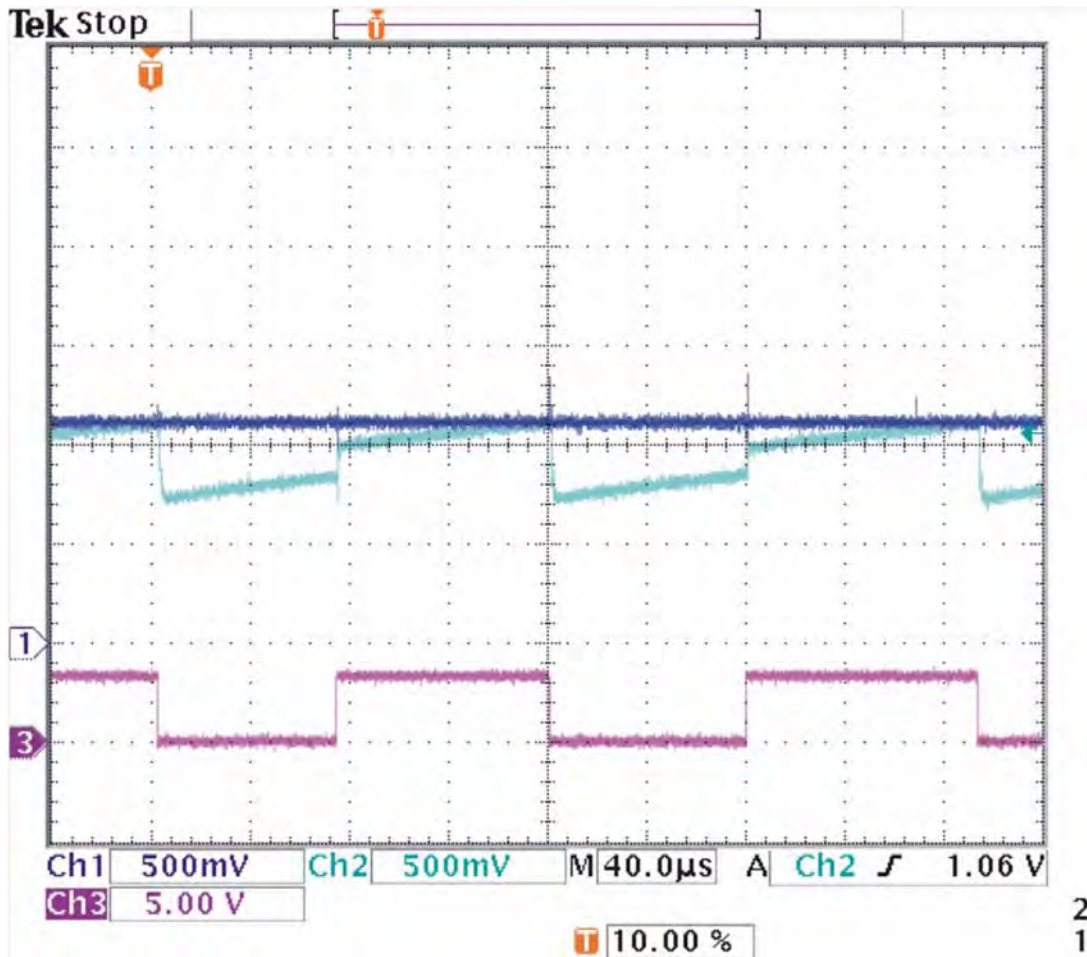


Figure 5. Cycle-by-Cycle Shutdown

Testing

This section describes how to run the code and demonstrate this sensored sinusoidal PWM 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)
- Z32F128 MultiMotor MCU Module (99C1461-001G) – Order separately
- Keil ARM ULINK2 Debugger – Order separately
- Digital Oscilloscope or Logic Analyzer
- PC with Internet access and at least two open USB ports

Hardware Setup

Figure 6 shows the application hardware connections.



Figure 6. The MultiMotor Development Kit with Z32F128 MCU Module and Keil ULINK2 Debugger

Procedure

Observe the following procedure to test the 3-Phase Sensorless Sinusoidal Motor Control demo program on the Z32F128 MultiMotor MCU Module.

1. Download and install Keil MDK-ARM μ Vision IDE version 5.14.0.0 (or newer) on your PC from the [Keil website](#).
2. Download the [AN0380-SC01.zip](#) source code file from the Zilog website and unzip it to an appropriate location on your PC.
3. Connect the hardware as shown in Figure 6.
 - a. The cables from the Keil ULINK2 debugger and the UART-to-USB adapter must be connected to two of the PC's USB ports.
 - b. Download and install the drivers for the Keil ULINK2 and the UART-to-USB adapter, if required.
4. Power up the MultiMotor Series Development Board using the 24 V DC adapter that is included in the Kit.
5. Using a serial terminal emulation program such as HyperTerminal, TeraTerm, or RealTerm, configure the serial port to 57600-8-N-1-N. A console screen should appear on the PC which will show the status of the motor and allow changes to the motor's operation.
6. Launch Keil μ Vision version 5.14.0.0 (or newer), select **Open Project** from the **Project** menu, browse to the directory on your PC to which the AN0380-SC01 source code was downloaded, locate the AN0380_SC01.zdsproj file, highlight it, and select **Open**.
7. Ensure that the RUN/STOP switch on the Z32F128 MCU Module is in the STOP position.
8. In Keil μ Vision, compile and flash the firmware to the Z32F128 MCU Module by selecting **Rebuild All target files** from the **Project** menu. Next, select **Debug** \rightarrow **Start/Stop Debug Session**, followed by **Debug** \rightarrow **Run**.
9. Set the RUN/STOP switch on the Z32F128 MCU Module to RUN. The motor should begin turning.
10. 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 Keil ULINK2 debugger controls the reset pin of the MCU. After debugging and running your code, detach the Keil ULINK2 debugger from J1 of the MultiMotor MCU Module to free the Reset pin and apply a power cycle to reset the MCU from Debug Mode.
-

Results

Linix BLDC-type and Teknik/Hudson PMSM-type motors were tested to compare their corresponding voltage and current waveforms. During operation of the BLDC motor, three oscilloscope probes were connected to the Hall sensors, and a scope probe was connected to one of the three motor phase BEMF resistor dividers to show the three 120-degree shifted Hall sensors in conjunction with one of three sine wave phase voltages. The scope channel was set to AC so that the positive and negative half of the sine wave modulates with respect to the midpoint. These three voltages and one current waveform are shown for the BLDC and PMSM motors in Figures 7 and 8, respectively.

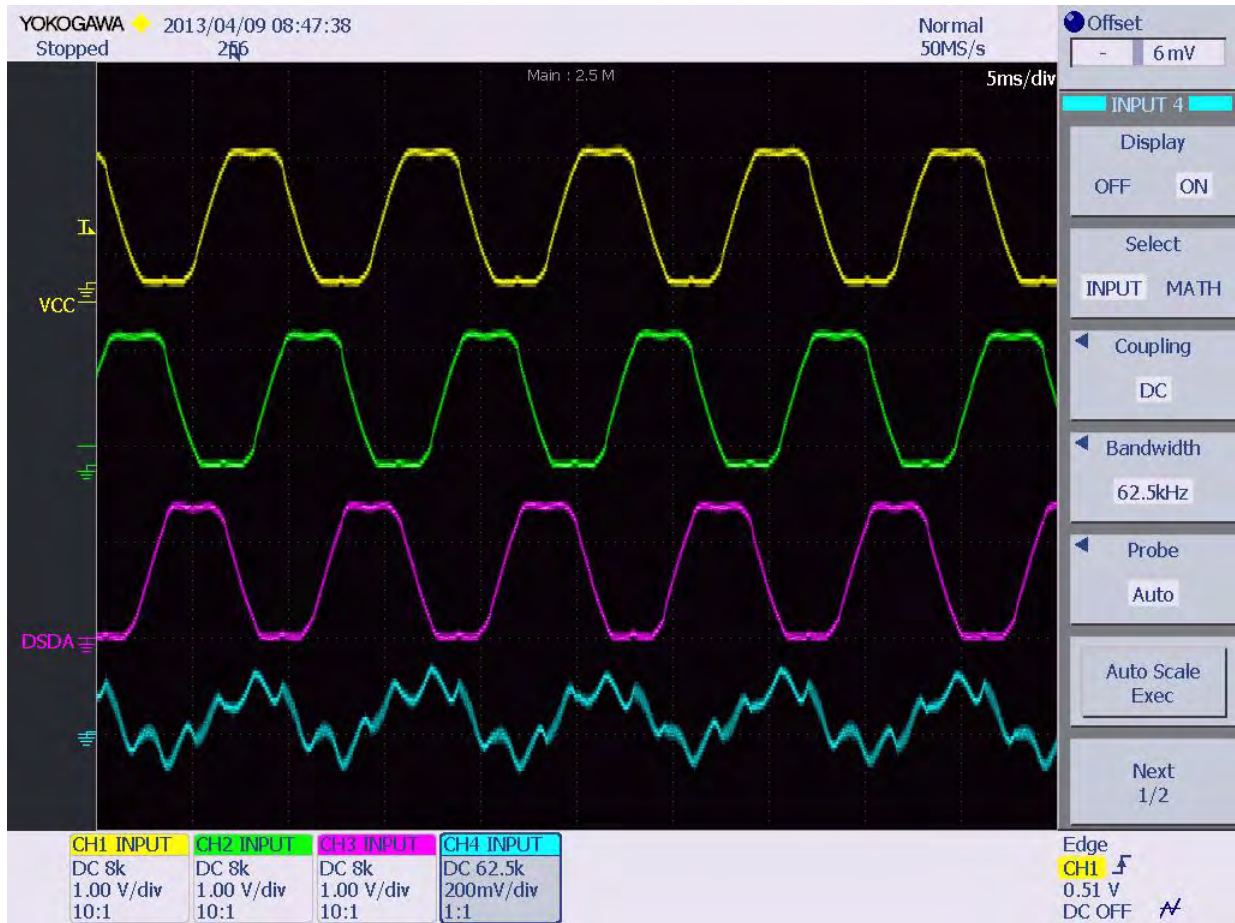


Figure 7. Linix BLDC Motor Phase Voltages and One Current Waveform

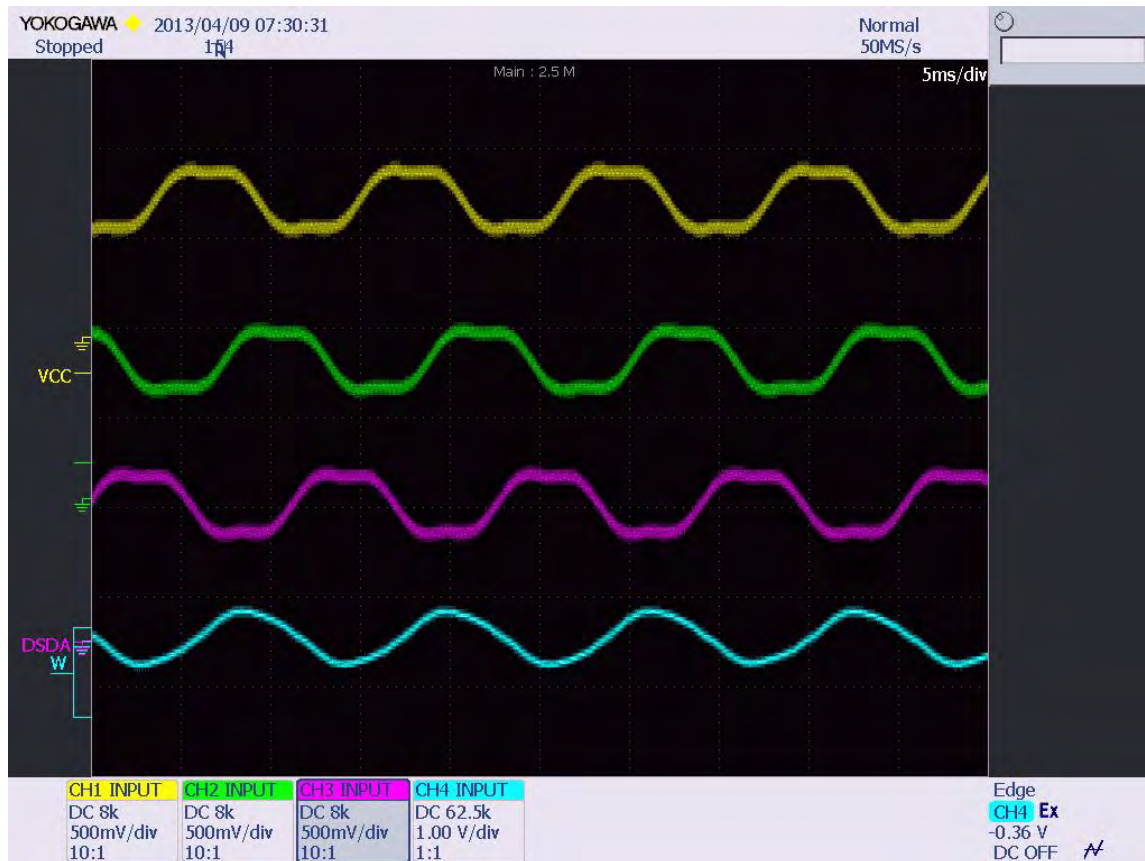


Figure 8. Hudson Teknik Motor Phase Voltages and One Current Waveform

Speed Control Performance in a Closed Loop Operation

To monitor performance of the speed control function while operating in a closed loop, the motor speed was set to 2000RPM at a nominal operating voltage of 24V. As this operating voltage was increased and decreased by plus and minus 4V, motor speed was observed to remain constant. To test the PI loop under load, the motor load was increased, which caused the PI to quickly ramp up the current to maintain the set speed. PI loop stability was verified by observing the voltage sine wave while loading the running motor, a condition for which the sine wave period time must be maintained constant in both amplitude and frequency.

Speed Control Performance in an Open Loop Operation

To monitor performance of the speed control function while operating in an open loop, the motor speed was set to 2000RPM at a nominal operating voltage of 24V. As this operating voltage was increased and decreased by plus and minus 4V, motor speed was observed to vary. Motor load was then increased, which caused the motor current to be increased while its speed slightly dropped.

Summary

The purpose of this application was to demonstrate the operation of a BLDC or PMSM type machine using the sinusoidal PWM technique.

To generate sinusoidal voltages and currents 120 degrees apart for a BLDC machine, a sine look up table (LUT) was implemented to reconstruct the three sine waves and formulas have been shown to calculate the motor frequency. Since the frequency calculations include the PWM period, all sinusoidal wave constructions are executed in the PWM interrupt service routine. The execution time for the sine wave reconstruction in the PWM service interrupt routine takes approximately 4 μ S. The execution time of the Hall interrupt service routine takes 7 μ s. Both execution times are based on a 80MHz internal PLL clock.

To maintain the correct 90 degree phase-angle between the stator and rotor field, the Hall interrupt service routine captures the binary Hall state upon each interrupt and fetches the corresponding reference angle from a Look Up Table (LUT).

The high byte of the PWM sine Look Up Table index is used to fetch the next value from the Sine LUT. Any offset value to the high byte of the PWM sine Look Up Table index will increase the frequency of the sine wave.

Sinusoidal PWM operation has the advantage of commutating the BLDC or PMSM with less acoustical and electrical noise, because the sine current through the windings has no steep current transitions. This allows for higher life expectancy of ripple current capacitor and ball bearings because the sinusoidal commutation approach causes no torque or current ripple in a PMSM or BLDC type motor. Besides electrical and acoustical noise reduction, the PWM sine approach also increases the efficiency in a BLDC-/PMSM-type motor. The efficiency can be further increased if a 3rd harmonic is injected into the PWM sine wave.

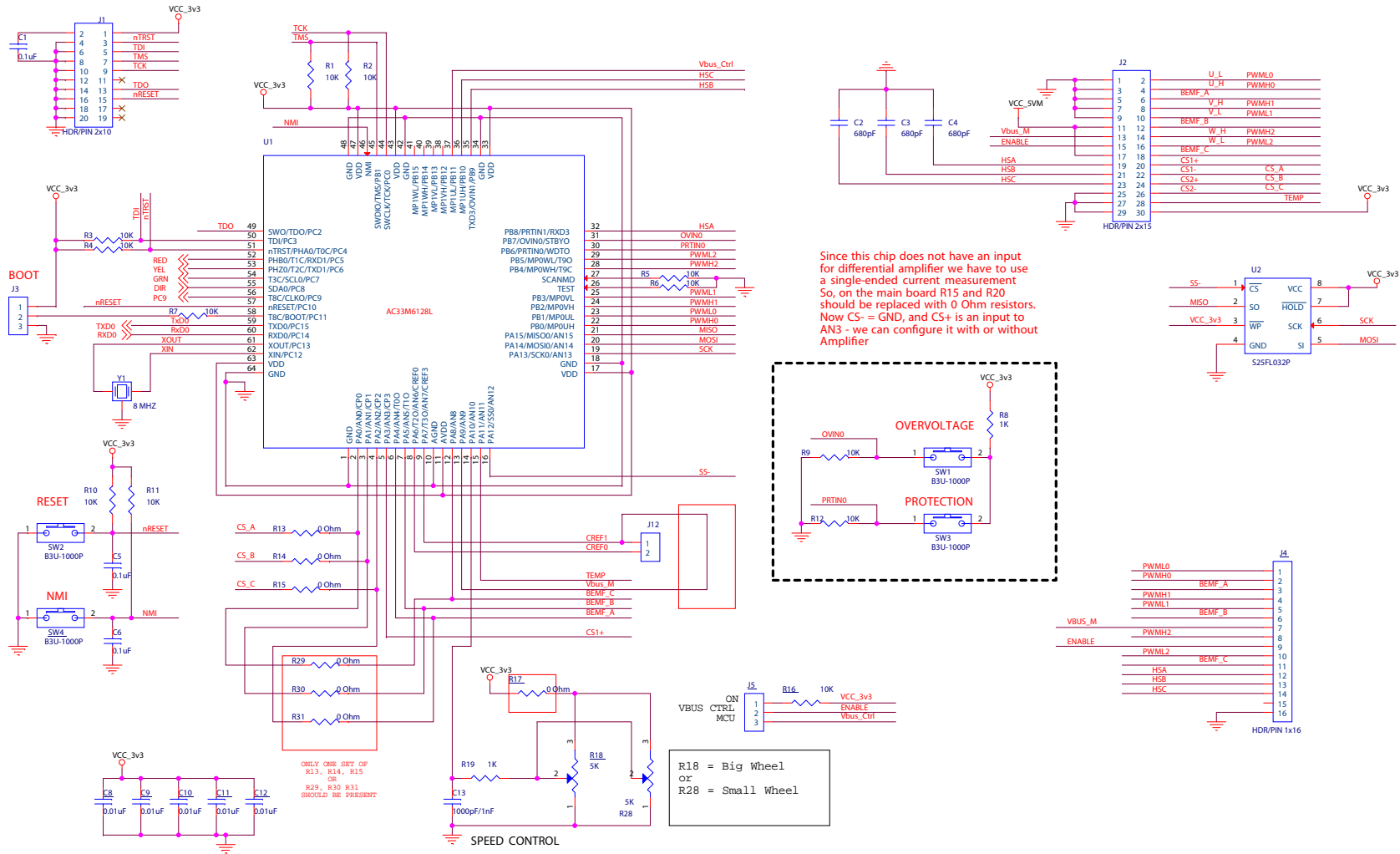
References

The following documents are each associated with the Z32F128 Series of Motor Control MCUs; each is available free for download from the Zilog website.

- [Z32F128 MCU Product Specification \(PS0345\)](#)
- [Z32F128 Evaluation Kit User Manual \(UM0277\)](#)
- [Space Vector Modulation of a 3-Phase BLDC Motor with the Z32F128 MCU \(AN0379\)](#)
- [MultiMotor Series Development Kit Quick Start Guide \(QS0091\)](#)
- [MultiMotor Series Development Kit User Manual \(UM0262\)](#)

Appendix C. Schematic Diagrams

Figures 9 and 10 show the schematics of the Z32F128 MCU Module.



Since this chip does not have an input for differential amplifier we have to use a single-ended current measurement. So, on the main board R15 and R20 should be replaced with 0 Ohm resistors. Now CS- = GND, and CS+ is an input to AN3 - we can configure it with or without Amplifier

R18 = Big Wheel
or
R28 = Small Wheel

Figure 9. Z32F128 MultiMotor MCU Module, #1 of 2

BLDC Motor Control Using Sensored Sinusoidal PWM Modulation with the Z32F128 MCU Application Note

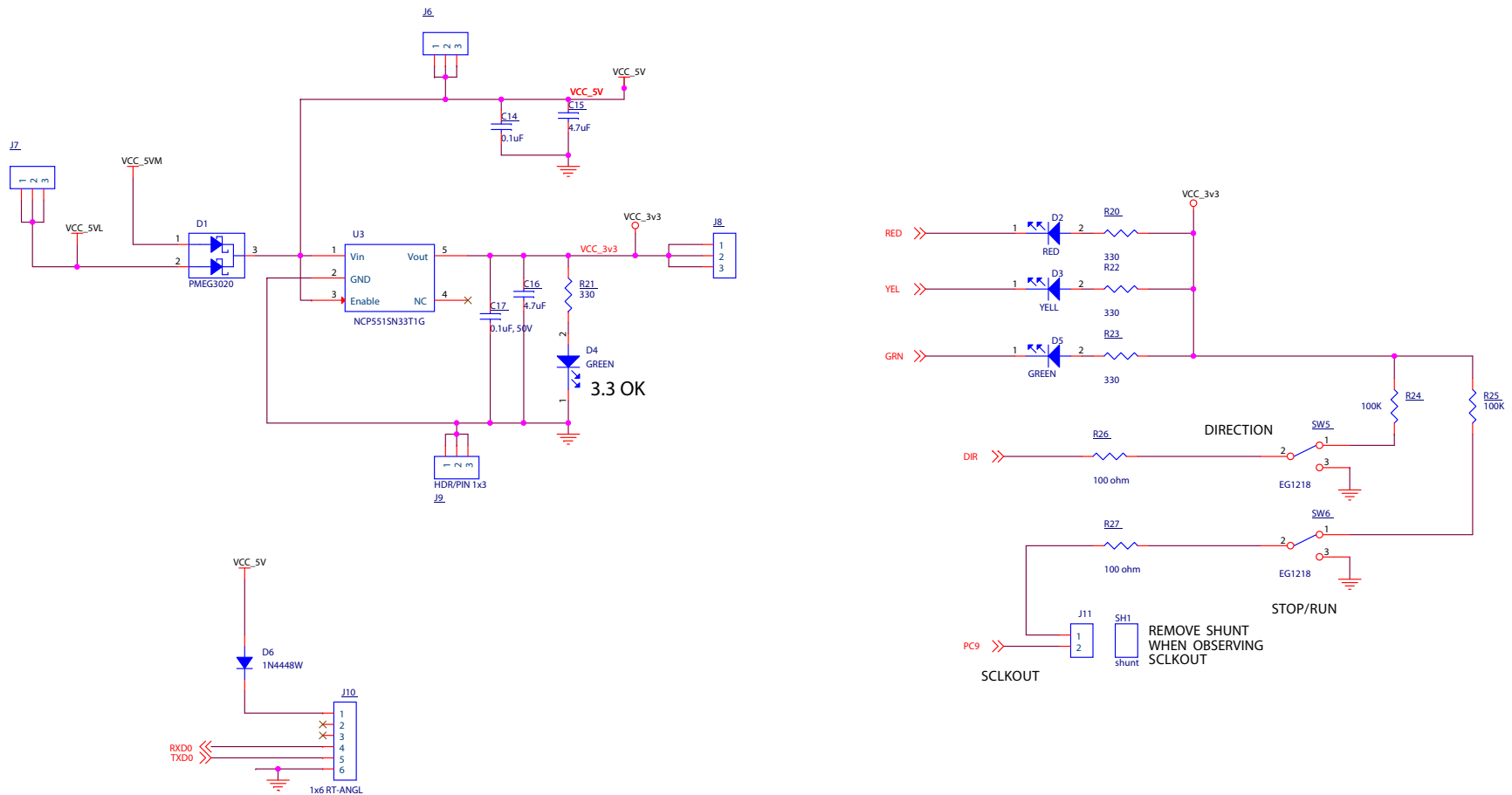


Figure 10. Z32F128 MultiMotor MCU Module, #2 of 2

Figures 11 and 12 show the schematics for the MultiMotor Main Board.

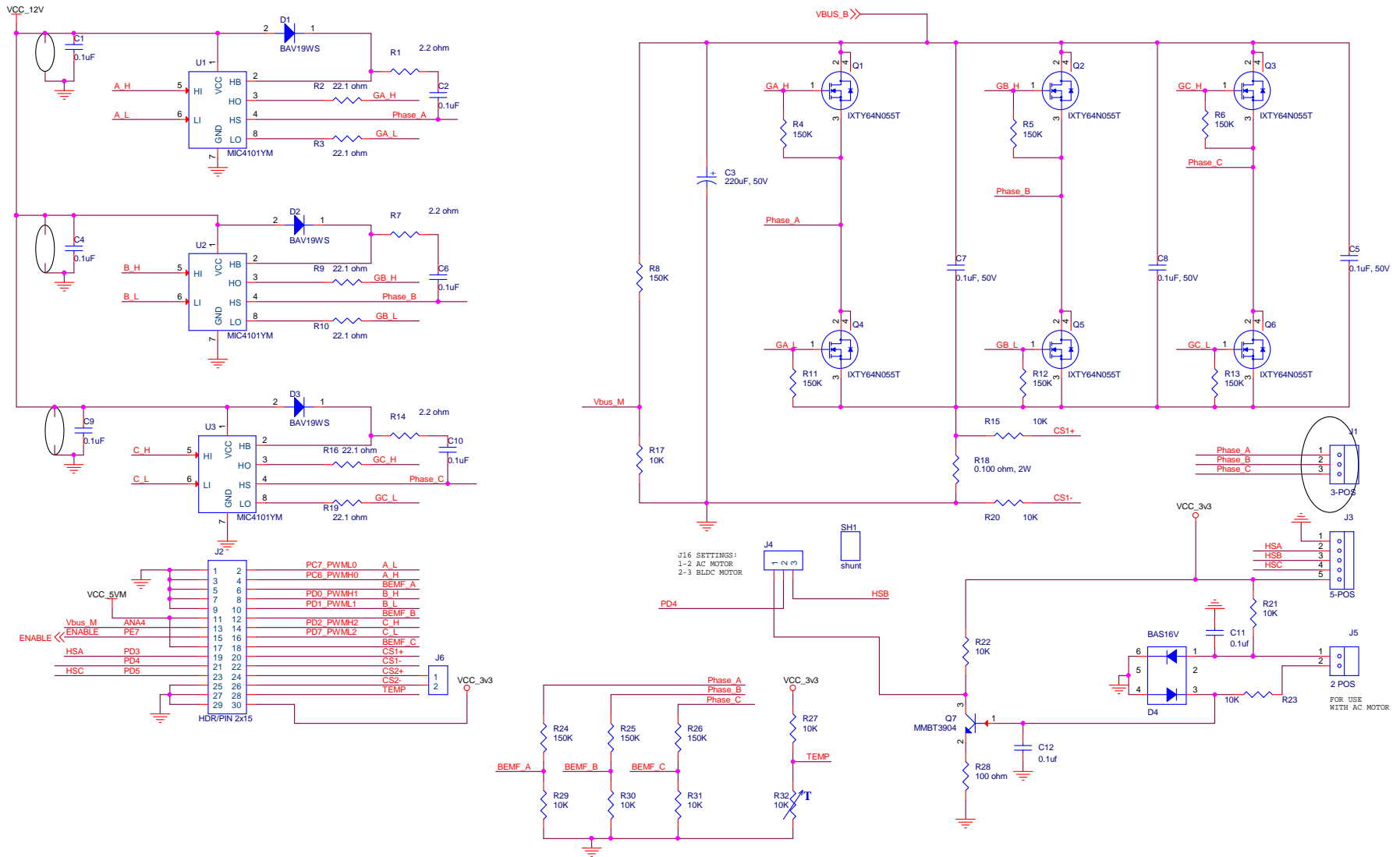


Figure 11. MultiMotor Development Board, #1 of 2

BLDC Motor Control Using Sensored Sinusoidal PWM Modulation with the Z32F128 MCU Application Note

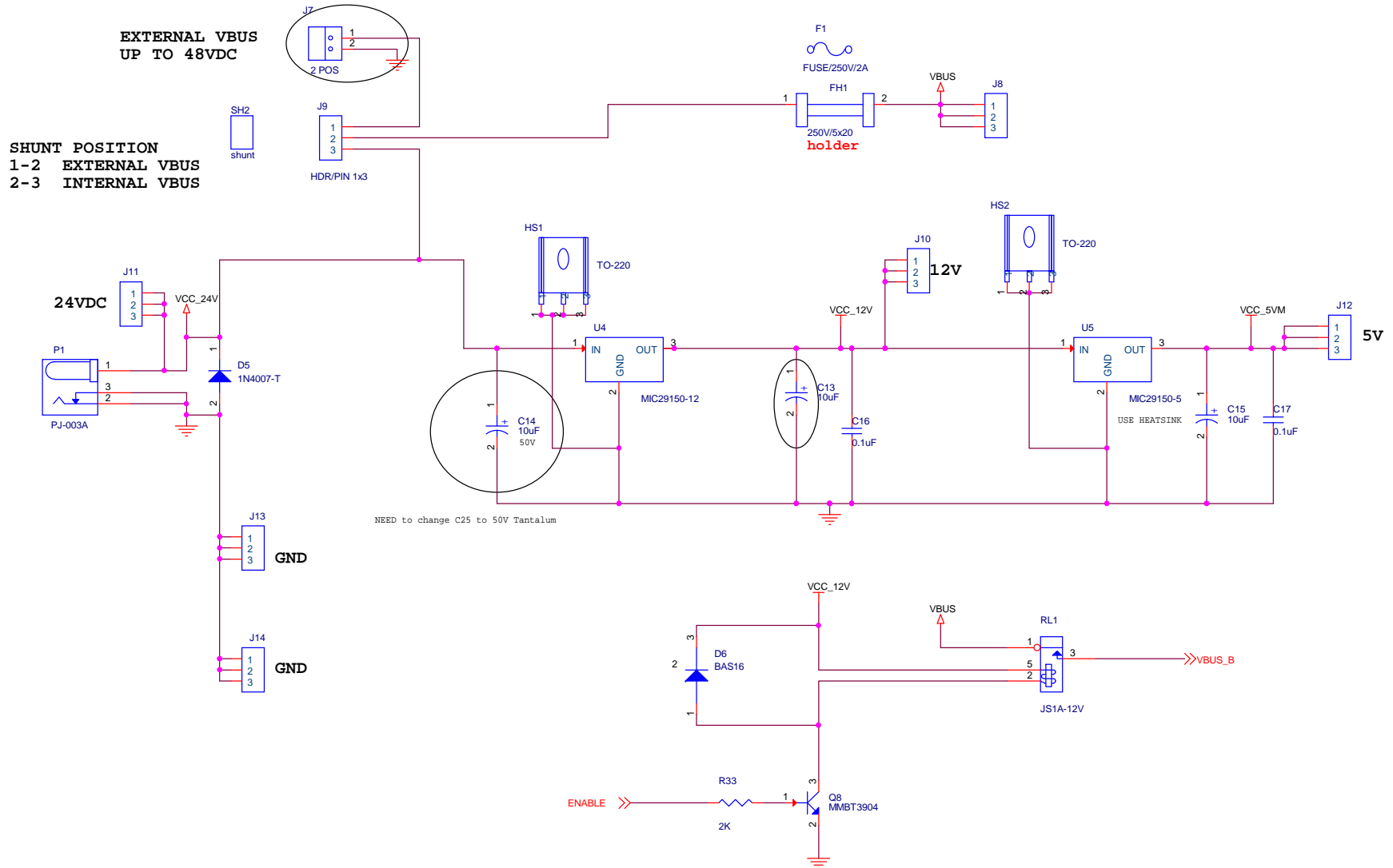


Figure 12. MultiMotor Development Board, #2 of 2

Appendix D. Flow Charts

Figure 13 presents an algorithm by which a 3-phase BLDC motor can be controlled using the Z32F128 MCU.

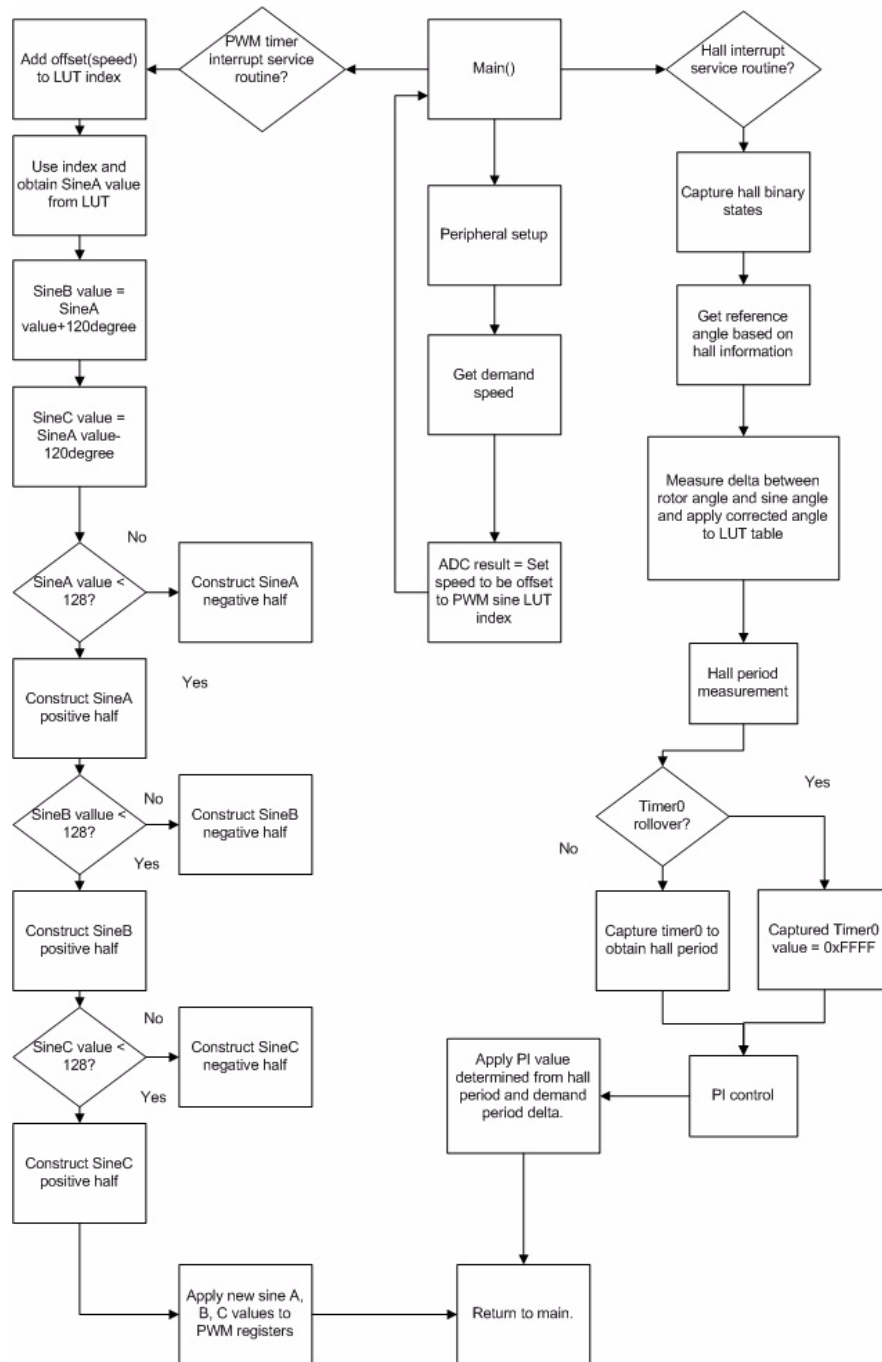


Figure 13. Simplified Control Algorithm

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 facts 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

©2016 Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

ZNEO32! and Z32F128 are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.