

Space Vector Modulation of a 3-Phase BLDC Motor with the Z16FMC MCU

AN035404-1015

Abstract

Space vector modulation techniques can be applied for AC induction motors, permanent magnet synchronous motors, and BLDC motor types. PMSM motors can be more efficient at smaller motor frame sizes compared with an ACIM machine of the same size.

A 3-phase BLDC motor can be controlled by creating a rotating voltage reference vector within a hexagon; the speed of rotation of this voltage reference vector determines the frequency of motor rotation. The space vector modulation application discussed in this application note uses a BLDC type motor with three Hall sensors for angular position feedback.

Constant cost pressure and increased consumer expectations have driven design engineers to seek minimal hardware solutions that extract maximum performance from motors used in consumer goods. This application note demonstrates how Zilog's Z16FMC MCU can implement efficient, cost-conscious vector modulation of a BLDC motor.

-
- **Note:** The source code file associated with this application note, [AN0354-SC01](#), is available free for download from the Zilog website. This source code has been tested with ZDSII – ZNEO version 5.0.1. Subsequent releases of ZDSII may require you to modify the code supplied with this application note.
-

Z16FMC Series Flash Microcontrollers

The Z16FMC Series of Flash MCUs is based on Zilog's advanced 16-bit ZNEO CPU core. The MCUs in this series are optimized for motor control applications and support control of single- and multiphase variable-speed motors. Target applications are consumer appliances, HVAC, factory automation, refrigeration, and automotive applications, among others.

To rotate a 3-phase motor, three AC voltage signals must be supplied and phase-shifted 120 degrees from each other and the MCU must provide six Pulse Width Modulation (PWM) outputs. To control a 3-phase motor, the MCU must provide six Pulse Width Modulation (PWM) outputs. The Z16FMC Series Flash 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 various motor types, and ensure safe operation of the motor by providing immediate shutdown of the PWM pins during a fault condition.

Discussion

An electric motor consists of a stator and a stationary frame in which a rotating component, or rotor, is mounted on a shaft and bearings. In a 3-phase BLDC motor, the stator is laced with three sets of inductor windings energized by three AC voltage inputs that are phase-offset 120 degrees from each other to produce a rotating field of magnetic flux. This stator flux field exerts a magnetic force on a rotor's permanent magnet flux field, resulting in torque on the output shaft.

In a 3-phase motor control application, the input to the motor is produced by a 3-phase inverter bridge. A bridge contains three complementary source/drain transistor pairs which connect either ground or high-voltage DC to each of its three outputs in response to digital control signals from the microcontroller. The microcontroller uses PWM on the bridge control signals to generate three approximately-sinusoidal AC waveforms on the bridge outputs, with the required 120-degree phase offset.

The duty cycle of each microcontroller PWM output is varied to control the period and amplitude of the generated AC signal which, in turn, determines the speed and torque of the motor.

Theory of Operation

Similarly to third harmonic-injected sinusoidal PWM, the Space Vector Modulation method utilizes about fifteen percent more of the available bus voltage, therefore increasing the efficiency of motor operation.

Unlike a non-third harmonic-injected sinusoidal PWM, the neutral point of the phase voltages is constrained to one-half of the bus voltage, as illustrated in Figure 1.

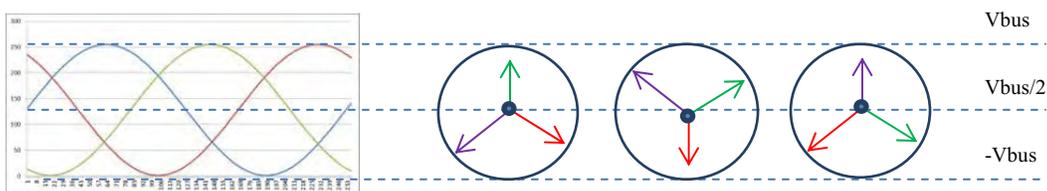


Figure 1. The Rotating Vectors are Constrained by $\pm V_{BUS}$ and the Center of the V_{BUS} Voltage

Space vector modulation is not confined to the limits of the V_{BUS} and the center voltages, and can float in space as illustrated in Figure 2.

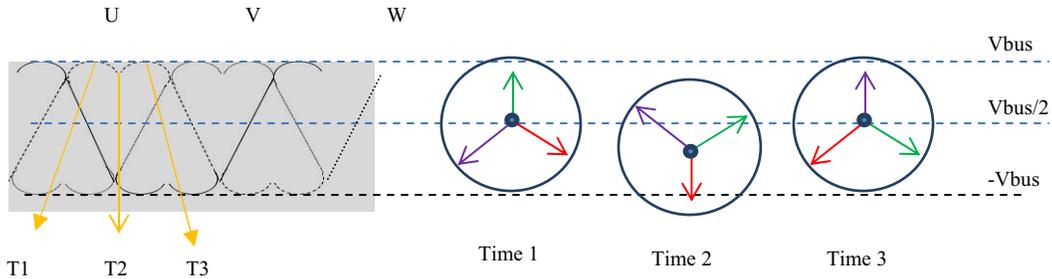


Figure 2. Center Voltage of the Space Vector Floats in Space

Unlike sinusoidal PWM which generates sinusoidal currents separately in each push/pull stage of the inverter, space vector modulation operates the entire inverter as a single unit to produce the sinusoidal currents. In doing so, the inverter is operated in eight different states within the hexagon, two of which are referred to as zero vectors because they produce no voltages, and six states which produce non-zero voltages.

The rotating reference voltage V_s within this hexagon, seen in Figure 3, is represented by a space vector using the following equation:

$$\underline{V}_s = V_s e^{j\theta}$$

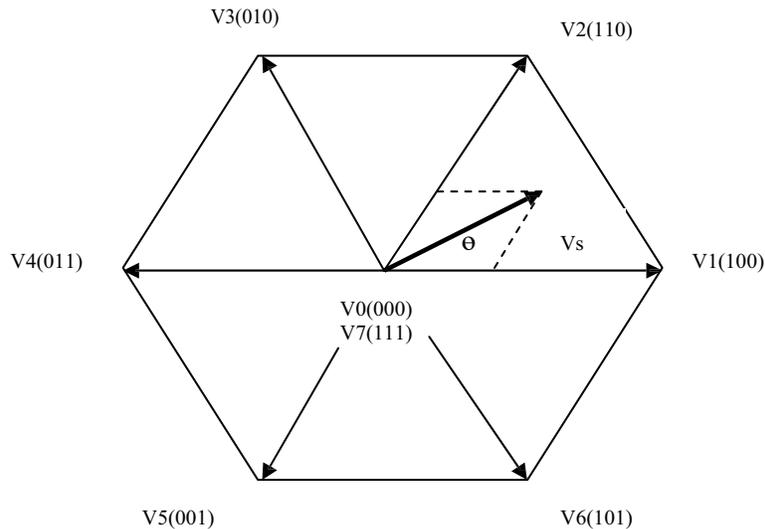


Figure 3. The Rotating Reference Vector V_{REF} within the Hexagon

With vector addition, the resulting reference space vector is:

$$V_s = (r_1 \times V_1) + (r_2 \times V_2)$$

To produce this rotating vector, the angular position within any two base vectors must be known, as shown in Figure 4.

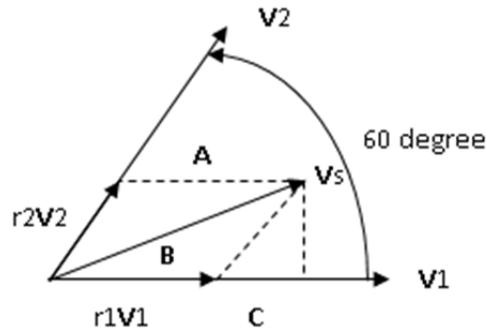


Figure 4. The Location of the Reference Vector in between any of Two Base Vectors

Knowing the angle of rotation and the adjacent base vectors within the hexagon, the scalar coefficients for the adjacent base vector r_1 and r_2 must be calculated to time-modulate the base vectors V_1 and V_2 toward generating the resulting voltage reference vector, V_s .

After the angular information is obtained, the scalar quantities r_1 and r_2 can be calculated using the following equations:

$$r_1 = m \times \sqrt{3} \times \sin(60 - \Delta\theta)$$

$$r_2 = m \times \sqrt{3} \times \sin(\Delta\theta)$$

In the above equations, m is the magnitude of the rotating space vector, V_s .

The time periods for which the adjacent base vectors are modulated to obtain the reference vector can be calculated using equations 1, 2, and 3:

Equation 1

$$t_0 = (1 - t_1 - t_2) \times T$$

Equation 2

$$t_1 = r_1 \times T$$

Equation 3

$$t_2 = r_2 \times T$$

In these three equations, T is the sum of t_0 , t_1 , and t_2 , and cannot be greater than the time period of the PWM. Next, t_0 becomes the time period for which either or both zero vectors are applied in combination with t_1 and t_2 , as illustrated in Figure 5.

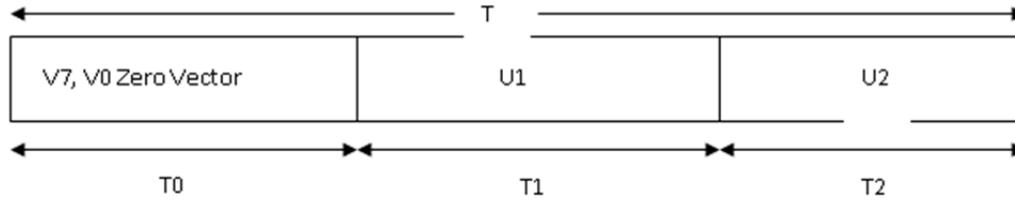


Figure 5. The Switching Times for Base Vectors V_1 , V_2 , and Zero Vectors

To find the time periods t_0 , t_1 , and t_2 , the angles are determined by using V_1 as the reference axis in a counterclockwise direction to determine the base vector angle to be subtracted from the angle of the reference vector.

Example 1

Using equations 1, 2, and 3 above, if the bus voltage is 24V and the desired reference vector magnitude is 12V, then the following equation can be calculated:

$$V_s = \frac{\text{Mag}}{V_{\text{bus}}} \times e^{j \cdot 190}$$

In the above equation, the angle is 190 degrees and the adjacent base vector is V_4 , which is 180 degrees. Therefore, r_1 and r_2 can be calculated using the following equations:

$$r_1 = \sqrt{3} \times \frac{12V}{24V} \times \sin[60^\circ - (190^\circ - 180^\circ)] = 0.663$$

$$r_2 = \sqrt{3} \times \frac{12V}{24V} \times \sin(190^\circ - 180^\circ) = 0.15$$

If the PWM period is $T = 50\mu\text{s}$, then the time duration for either zero vector V_0 , V_7 is:

$$t_0 = (1 - r_1 - r_2) \times T = 9.3\mu\text{s}$$

The time duration for t_1 is:

$$t_1 = r_1 \times T = 33.2\mu\text{s}$$

And the time duration for t_2 is:

$$t_2 = r_2 \times T = 7.5\mu\text{s}$$

Space vector control allows for different switching combinations using t_1 and t_2 based on the choice of the null vectors which are applied for duration of time t_0 . Applying the zero vectors V_0 , V_7 , or both V_0 and V_7 results in different switching patterns to generate either less total harmonic distortion or to reduce linear switching power losses in the switching devices. Using these V_0 , V_7 zero vectors can serve to obtain a regenerative braking effect, especially when using ACIM.

Example 2

To reduce linear switching losses with either V_0 or V_7 , zero vectors can be applied in sequence; i.e, using V_0 as the zero vector in the sequence $t_1 \rightarrow t_2 \rightarrow t_0$ or using the V_7 zero vector in the same sequence. In both cases, each of the three phases in the inverter does not switch for one-third of the time in a cycle.

However, different combinations of switching sequences have different effects on the inverter circuit, depending on the size of the bootstrap capacitors used for the high- and low-side drivers. If the V_0 zero vector is used, the bootstrap will still work because the capacitors can discharge; however, such may not be the case when using the V_7 vector as a zero vector.

The resulting phase waveforms are shown in Figures 6 through 10.

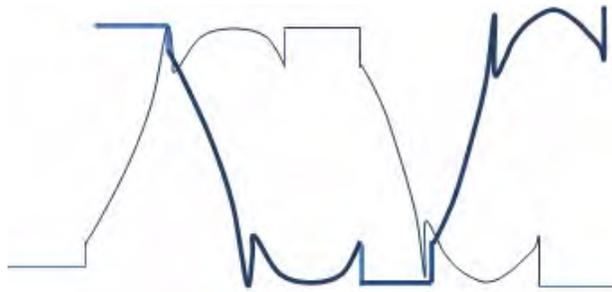


Figure 6. Using V_7 as the Null Vectors in Sector 1, 3, 5 and V_0 in Sector 2, 4, 6

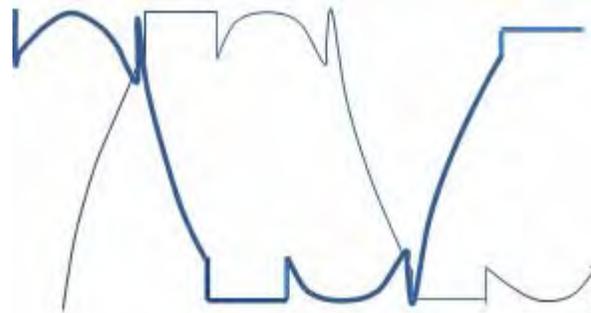


Figure 7. Using V_0 as the Null Vectors in Sectors 1, 3, and 5 and V_7 in Sectors 2, 4, and 6

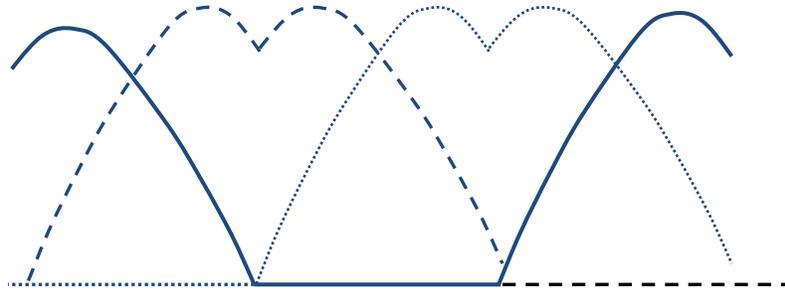


Figure 8. Always Using V_0 as Zero Vector

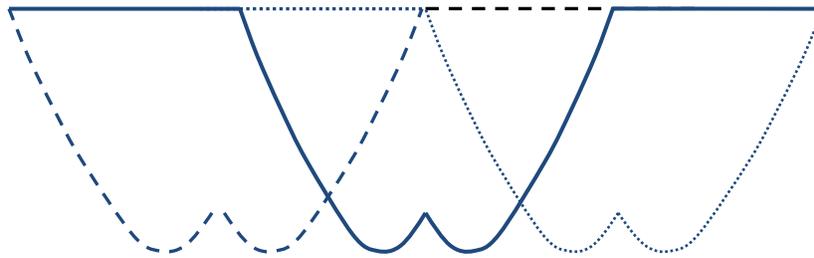


Figure 9. Always Using V_7 as a Zero Vector

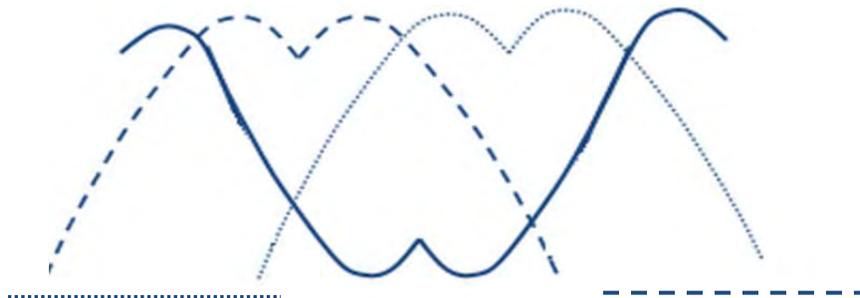


Figure 10. Using Alternate Reverse Switching Modes by Alternating Zeroes for Each Sequence and Reverse Sequence After Each Zero Vector

Examples for PWM timings using null vectors V_0 , V_7 across the six hexagon sectors are listed in Table 1.

Table 1. PWM Timings for Each of Six Sectors

Sector 1	Sector 2	Sector 3	Sector 4	Sector 5	Sector 6
PhsA = $t_1 + t_2$	PhsA = $t_0 + t_1$	PhsA = 0	PhsA = t_0	PhsA = t_2	PhsA = 100%
PhsB = t_2	PhsB = 100%	PhsB = $t_1 + t_2$	PhsB = $t_0 + t_1$	PhsB = 0	PhsB = t_0
PhsC = 0	PhsC = t_0	PhsC = t_2	PhsC = 100%	PhsC = $t_1 + t_2$	PhsC = $t_0 + t_1$

Application

To apply space vector theory, the Z16FMC microcontroller’s PWM module is configured as three complementary output pairs. Each output pair controls one complementary source/drain transistor pair in the inverter bridge. The PWM module is configured to insert a deadband between ON states to prevent leakage that might occur if one transistor in a pair turns on before the other is fully off.

Each PWM output pair produces a stream of complementary on/off pulses to activate the corresponding source or drain transistor in the inverter bridge. The voltage of each bridge output varies with the source/drain pulse duty cycle.

The period of each PWM cycle is configured to be 50µs; the PWM module generates an interrupt request at the end of each cycle to calculate the PWM timings for of the space vector modulation signals. These signals are loaded into the three PWM registers for Phase A, Phase B, and Phase C. Therefore, the primary goal of the ISR is to update the duty cycle value for each PWM channel, as required, to produce the appropriate AC waveforms at the inverter bridge outputs.

The frequency of the rotating vector is calculated as:

$$\text{Freq} = \frac{\text{LUToffset}}{\text{PWMperiod} \times \text{LUTsize}}$$

In the above equation, the *LUToffset* value is a 16-bit integer index, of which only the upper byte is used to select the Look-Up Table (LUT) entries.

The synchronous speed of the rotor can then be calculated as:

$$\text{Speed} = 120 \times \frac{\text{Freq}}{\text{Poles}}$$

Utilizing Timer 0, the time period of the rotor is measured in terms of timer ticks. This information is then used in the PI speed control loop.

$$\text{TimerTicks} = \frac{\frac{\text{MCUclockFreq}}{\text{TimerPrescaler}}}{\frac{\text{LUToffset}}{\text{PWMperiod} \times \text{LUTsize}}}$$

Equipment Used

The following equipment is used for the setup to demonstrate the space vector modulation technique. The first four items are included 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
- PC with Internet access and at least two open USB ports

Hardware Setup

Figure 11 illustrates the application hardware connections required to operate the motor with space vector modulation.

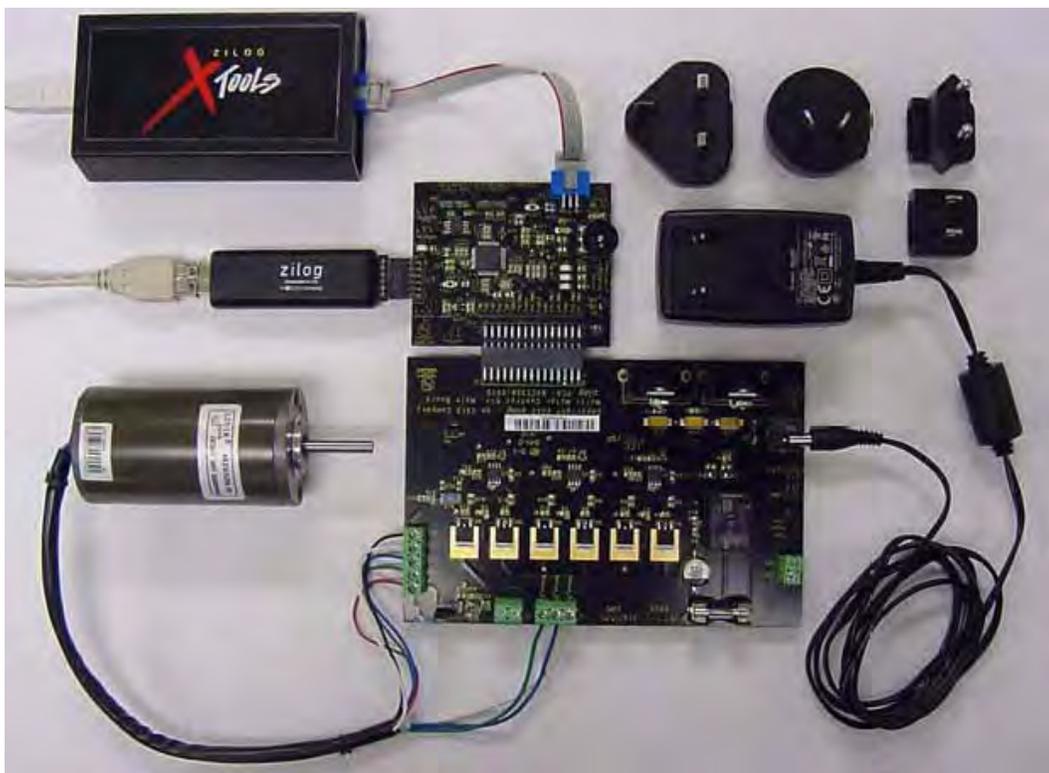


Figure 11. The MultiMotor Development Kit with Z16FMC MCU Module and SmartCable

Testing Procedure

Observe the following procedure to test space vector modulation on the Z16FMC MCU Module.

1. Download and install ZDSII – ZNEO 5.0.1 (or newer) on your PC from the [Zilog Store](#).
2. Download the [AN0354-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 11.
 - a. The cables from the Opto-Isolated USB SmartCable 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 SmartCable and the UART-to-USB adapter, if required.
 - c. For additional assistance, refer to the [MultiMotor Series Development Kit Quick Start Guide \(QS0091\)](#).
4. Power up the MultiMotor Series Development Board using the 24V 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 ZDSII – ZNEO, select **Open Project** from the **File** menu, browse to the directory on your PC in which the AN0354-SC01 source code was downloaded to, locate the AN0354_SC01.zdsproj file, highlight it, and select **Open**.
7. Ensure that the RUN/STOP switch on the Z16FMC MCU Module is in the STOP position.
8. In ZDSII, compile and flash the firmware to the Z16FMC MCU Module by selecting **Rebuild All** from the **Build** menu. Next, select **Debug** → **Download code**, followed by **Debug** → **Go**.
9. Set the RUN/STOP switch on the Z16FMC MCU Module to RUN. The motor should begin turning.
10. In the GUI terminal console, enter the letter `u` to switch to UART control; a menu similar to the example shown in Figure 12 should appear. As a result, commands can now be entered using the console to change the motor's operation.

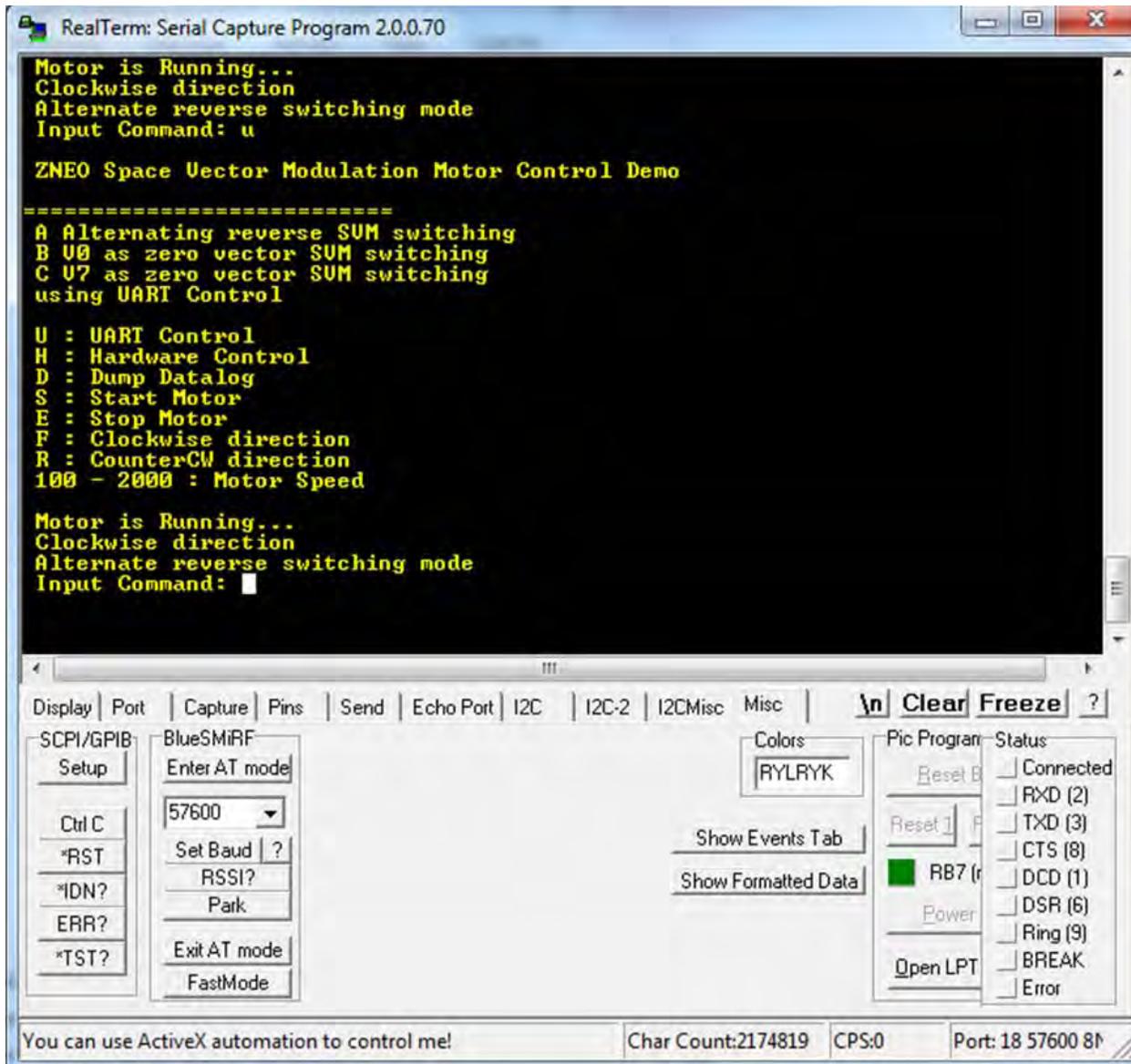


Figure 12. GUI Terminal Showing the Alternate-Reverse Switching Pattern

11. At the Input Command: prompt, enter the letter A to indicate an alternate-reverse switching pattern. There will be no apparent change in the motor's operation; however, the signals going to the motor will change and can be displayed as shown in Figures 15 through 20, beginning on [page 14](#).
12. While the motor is running, enter the B character at the HyperTerminal prompt to indicate the V_0 Zero Only Vector switching pattern, as shown in Figure 13.

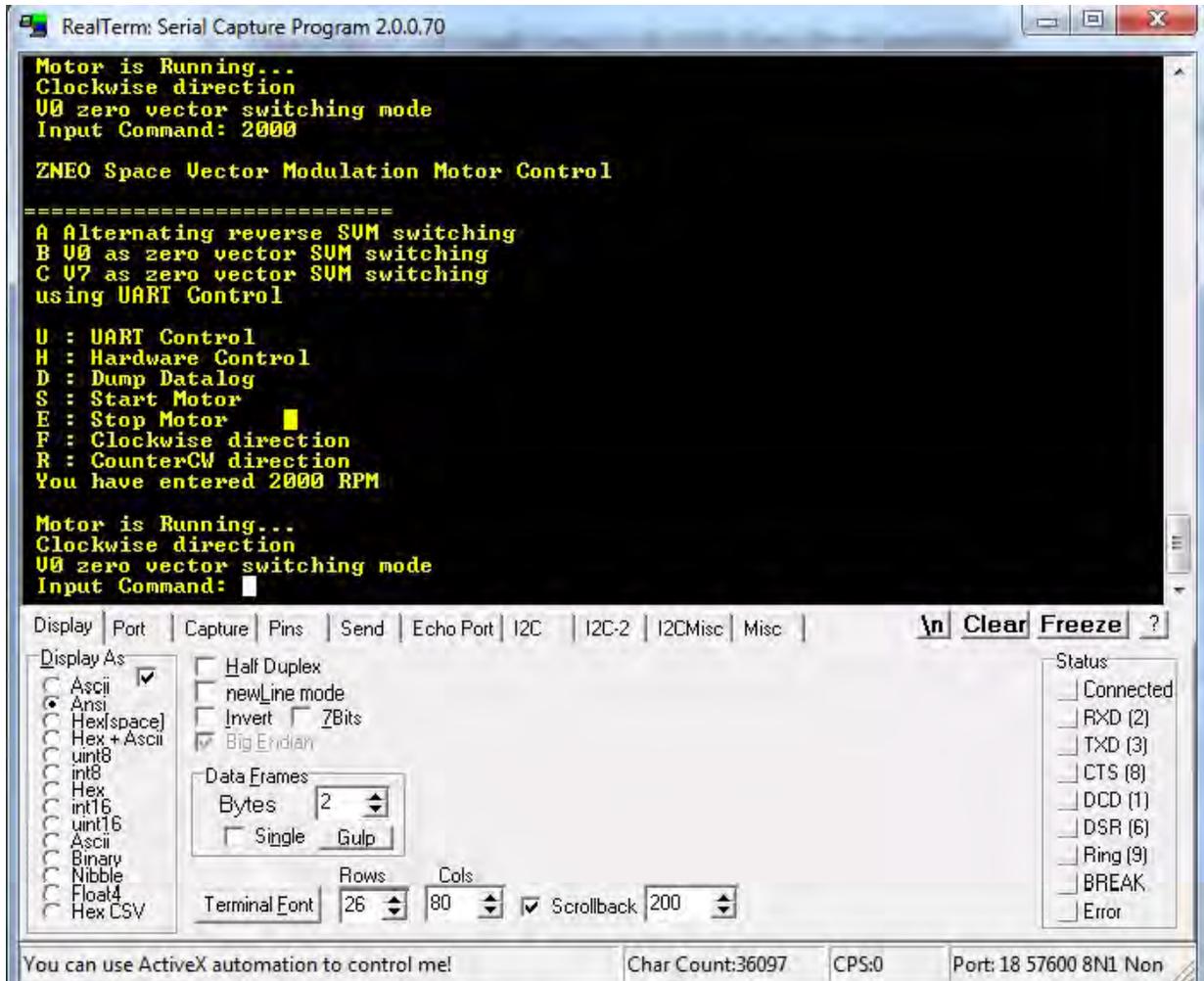


Figure 13. GUI Terminal Showing the V_0 Zero Vector Only Switching Pattern

13. While the motor is running, enter the C character at the HyperTerminal prompt to indicate the V_0 , V_7 Zero Vector switching pattern, as shown in Figure 14.

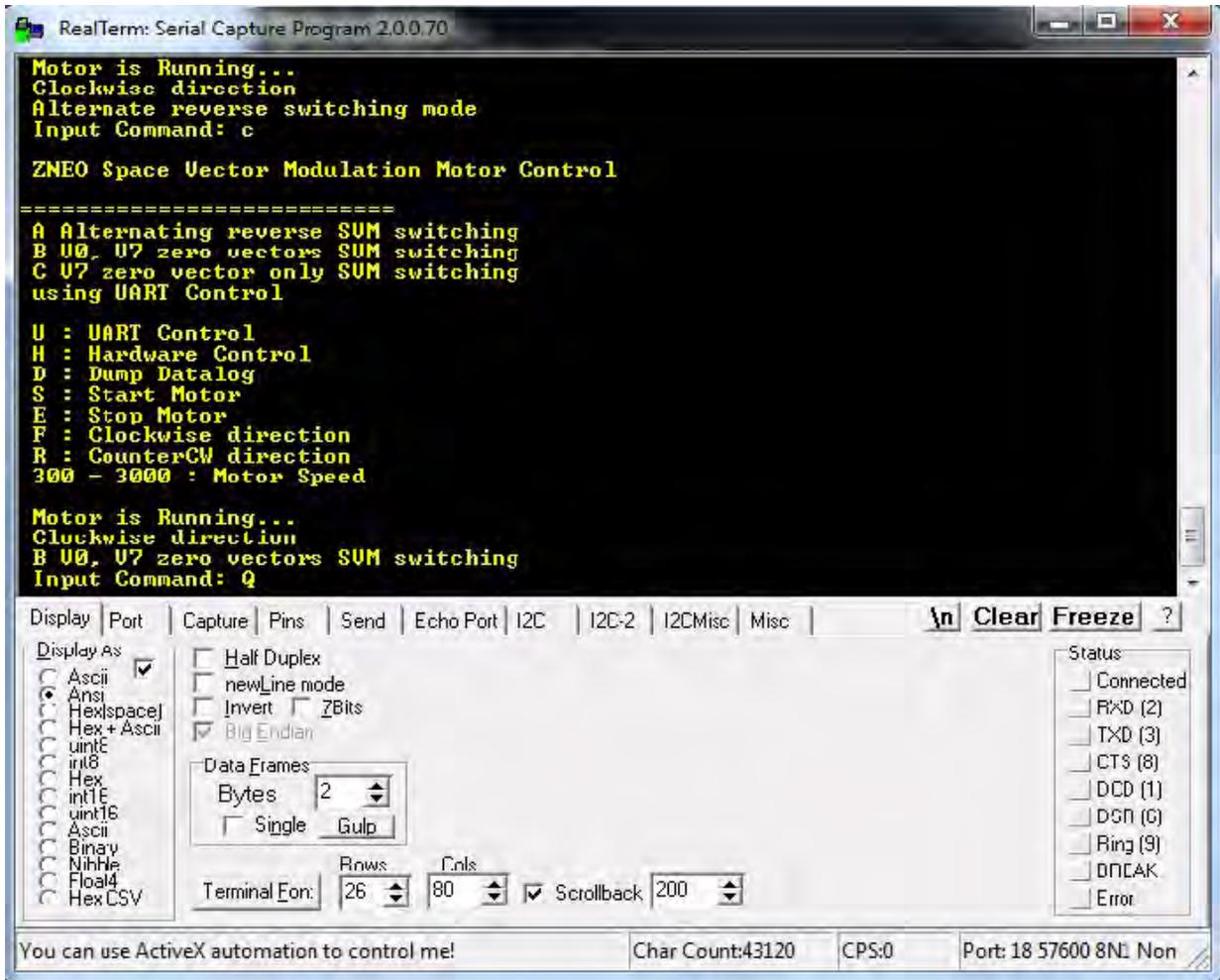


Figure 14. GUI Terminal Showing the "V₀, V₇ Zero Vector" Switching Pattern

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

Results

In this application, three oscilloscope probes are connected to the Phase A, Phase B, and Phase C offsets of the MultiMotor Series Development Board to show three different switching patterns. These scope probes were also connected to BEMF voltage dividers to monitor the generated BEMF voltages and, ultimately, to view the associated switching pattern waveforms.

Figure 15 illustrates the alternate-reverse space vector modulation pattern on Phase A, Phase B, and Phase C.

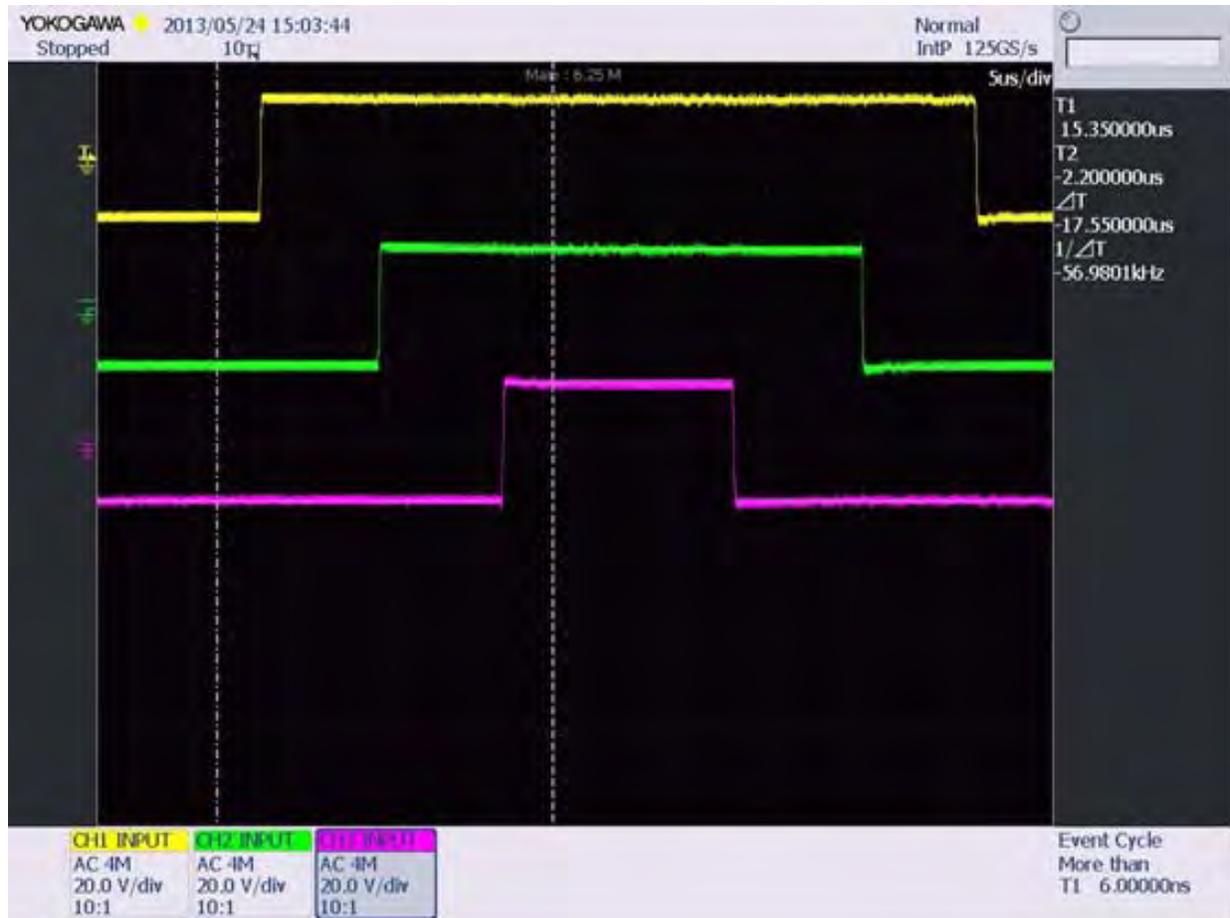


Figure 15. Alternate-reverse Space Vector Modulation Pattern

Figure 16 illustrates the V_0 , V_7 switching pattern on Phase A, Phase B, and Phase C.



Figure 16. V_0 , V_7 Space Vector Modulation Pattern

Figure 17 illustrates the V_7 Zero Vector Only switching pattern on Phase A, Phase B, and Phase C.



Figure 17. V_7 Zero Vector Only Switching Pattern

Figure 18 illustrates the alternate-reverse waveform on Phase A, Phase B, and Phase C.

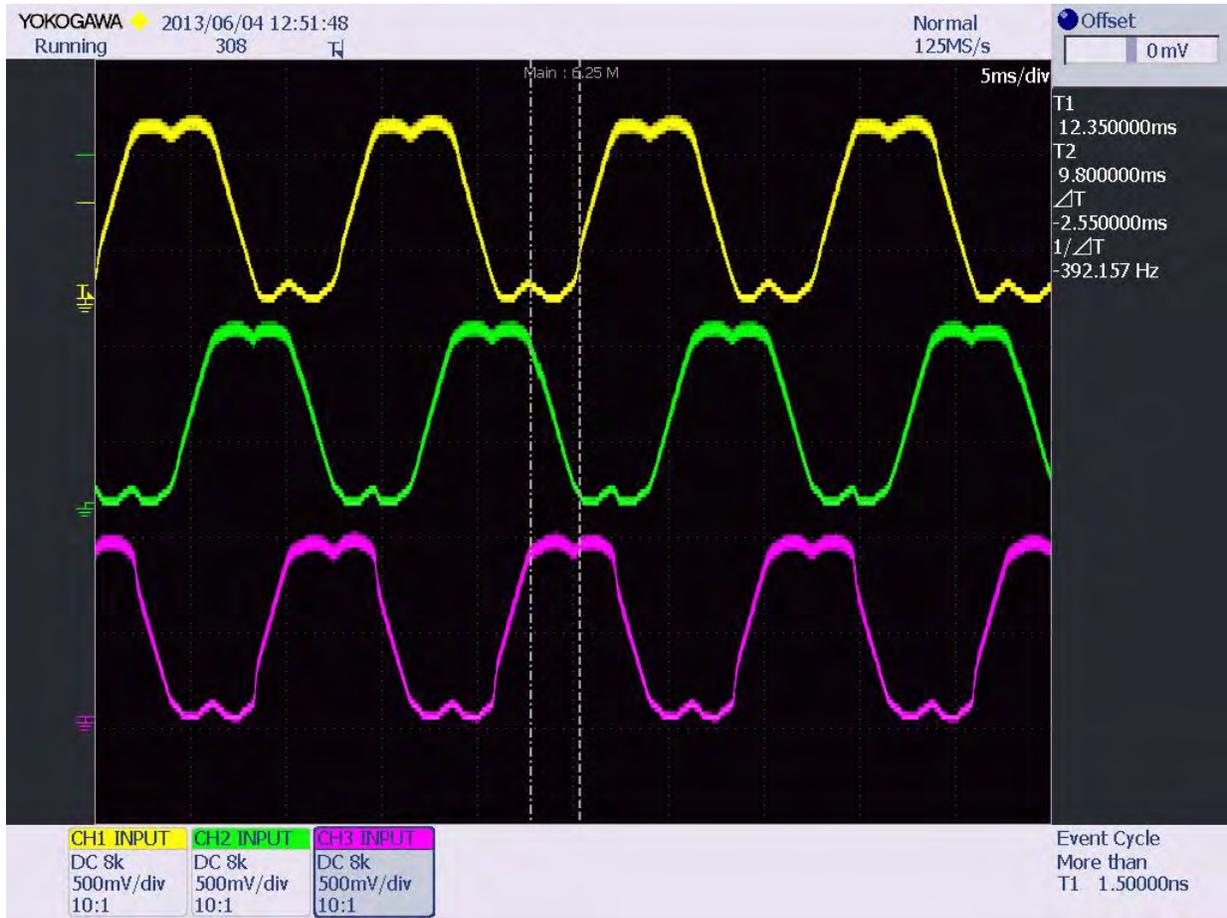


Figure 18. Alternate-Reverse Waveforms

Figure 19 illustrates the V_0 , V_7 Zero Vector Only Waveform on Phase A, Phase B, and Phase C.

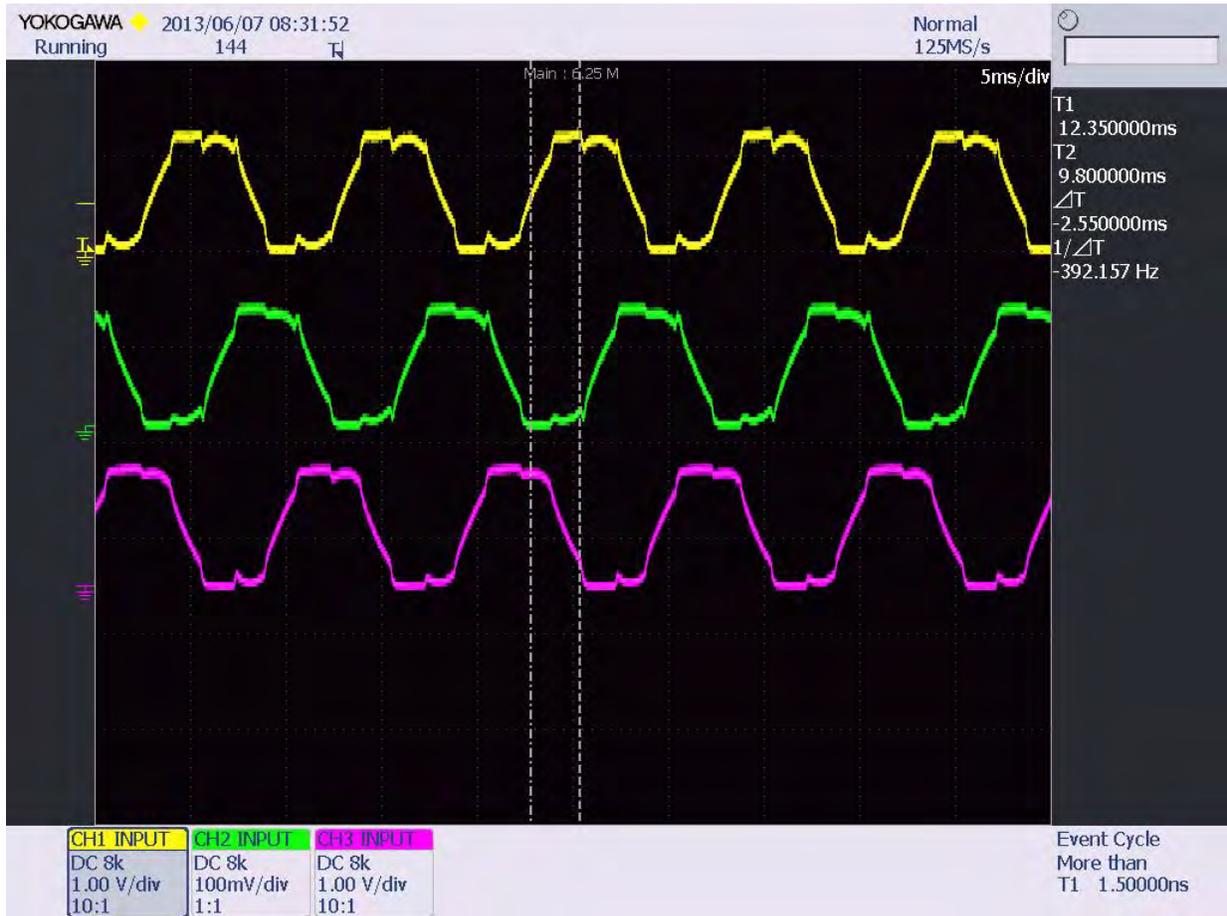


Figure 19. V_0 , V_7 Zero Vector Only Waveforms

Figure 20 illustrates the V_7 Zero Vector Only Waveform on Phase A, Phase B, and Phase C.

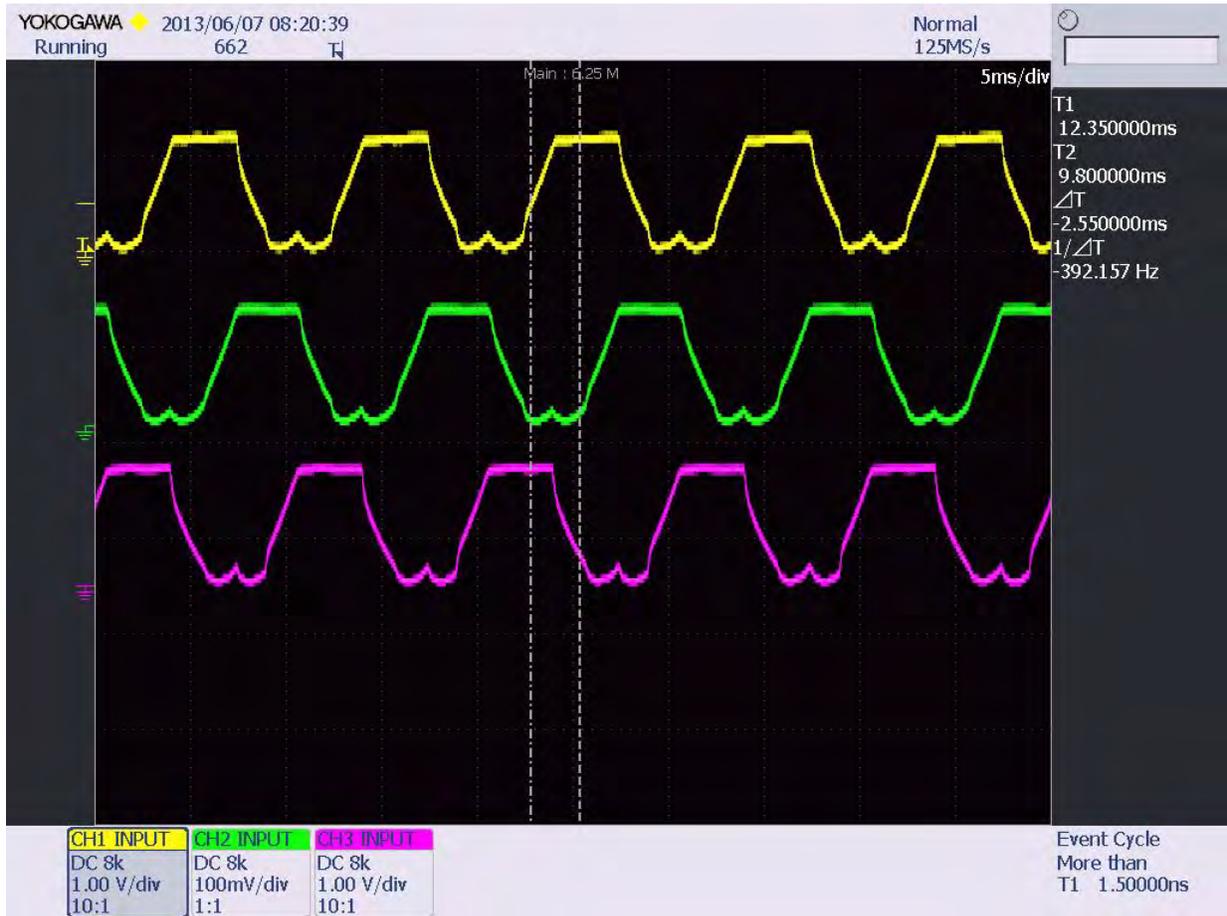


Figure 20. V_7 Zero Vector Only Waveforms

Closed-Loop Control Performance

To monitor closed-loop speed control performance, the motor speed was set to 2000RPM at a nominal operating voltage of 24 V. As this operating voltage was increased and decreased by $\pm 4V$, motor speed was observed to remain constant. To test the PI loop under load, the motor load was increased, which caused the PI loop to quickly ramp up the current to maintain the set speed. The 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 with no jitter. The closed-loop speed control essentially provides power operation of the motor which remains constant.

Open-Loop Control Performance

To monitor the 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 or decreased by $\pm 4V$, motor speed was observed to vary accordingly. The motor load was then increased, which caused the motor current to be increased while its speed dropped.

Summary

The purpose of this application is to demonstrate the operation of a BLDC- or PMSM-type machine using the Space Vector Modulation technique. To generate sinusoidal voltages and currents, a voltage reference vector is rotated 360 degrees within a hexagon. Each of the six sectors within this hexagon creates unique switching patterns for the space vector modulation.

Space vector modulation has the advantage of utilizing about fifteen percent more of the available bus voltage. Formulas discussed in this document have been shown to calculate the space vector modulation timings and resulting motor frequency. Because the frequency calculations include the PWM period, all space vector 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 is approximately $6\mu s$. The execution time of the Hall interrupt service routine is approximately $8\mu s$. Both execution times are based on a 20MHz external clock.

To maintain synchronization and commutation angle between the reference vector frequency and rotor frequency, 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 then used to fetch the next value from the Sine Look-Up Table (in which the LUT index is interpolating). Any positive or negative offset value to this high byte of the PWM sine look-up table will accordingly increment or decrement the frequency of rotation of the reference vector.

Space vector modulation has the advantage of commutating a BLDC or PMSM motor with less acoustical and electrical noise, because the sine current through the windings has no steep current transitions. The effects of total harmonic distortions and linear switching power losses can be further manipulated by applying different space vector modulation switching schemes. Such manipulations allow for higher life expectancy of ripple current capacitors and ball bearings because the sinusoidal commutation approach causes virtually no torque or current ripple in a PMSM or BLDC motor. In addition to electrical and acoustical noise reduction, the PWM sine approach also increases the efficiency of a BLDC-/PMSM-type motor due to its fifteen percent higher bus voltage utilization. The program uses unsigned integer-type variables only, to avoid additional execution times.

The example application and techniques described in this document should prove helpful for anyone who intends to develop motor control applications based on the Z16FMC Series family of microcontrollers.

Appendix D. References

Documents associated with the Z16FMC Series of products are listed below. Each of these documents can be obtained from the Zilog website by clicking the link associated with its document number where indicated.

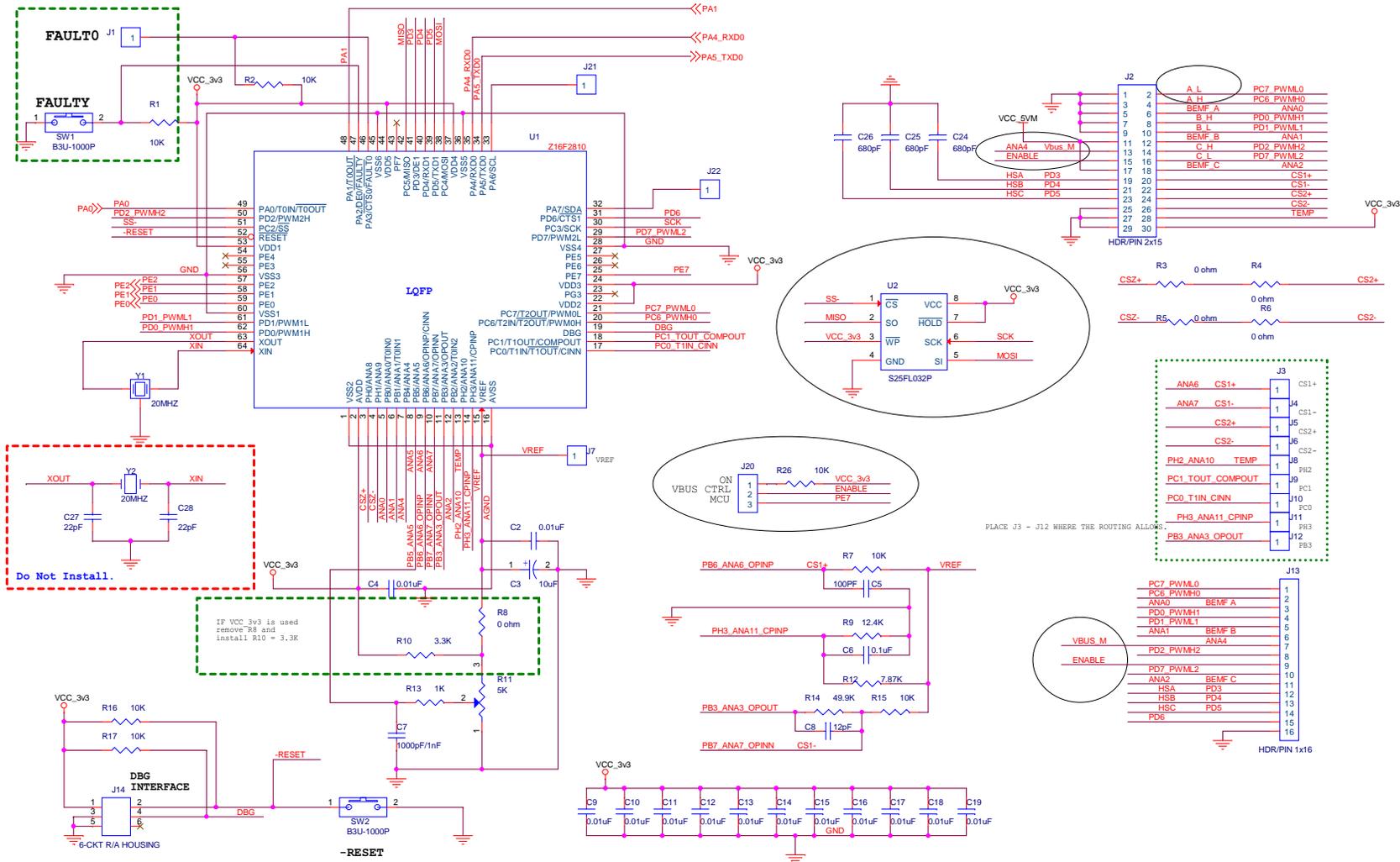
- ZNEO CPU User Manual ([UM0188](#))
- Z16FMC Series Motor Control Product Specification ([PS0287](#))
- MultiMotor Series Development Kit Quick Start Guide ([QS0091](#))
- MultiMotor Series Development Kit User Manual ([UM0262](#))
- MultiMotor Control with Parameter Monitoring Using the Z16FMC MCU Application Note ([AN0343](#))
- 3-Phase Sensorless Brushless DC Motor Control Application Note ([AN0353](#))
- Hall Sensor Sinusoidal PWM Modulation Brushless DC Motor Control Application Note ([AN0355](#))
- 3-Phase Hall-Sensor Brushless DC Motor Control Application Note ([AN0356](#))
- Implementing a Data Logger with Spansion SPI Flash Application Note ([AN0360](#))

The following external documents offer sound fundamentals for understanding motor control concepts.

- *Motor Control Electronics Handbook*, Richard Valentine; McGraw Hill.
- *Short Course on Electric Drives: Understanding Basics to Advanced Control & Encoder-Less Operation*, Ned Mohan, University of Minnesota, 2005: a recording of the Internet-based short course presented on May 12, 2005 by Professor Mohan and edited to fit on a DVD.
- *Lehrstuhl fuer Elektrische Antriebssysteme und Leistungselektronik*, Prof. Dr. Ing. Ralph Kennel, Technische Universitaet Muenchen.
- *Electric Machinery*, Peter F. Ryff.

Appendix E. Schematic Diagrams

Figures 21 and 22 present the schematic diagrams for the Z16FMC MCU Module.



Figures 23 and 24 present the schematic diagrams for the MultiMotor Main Board.

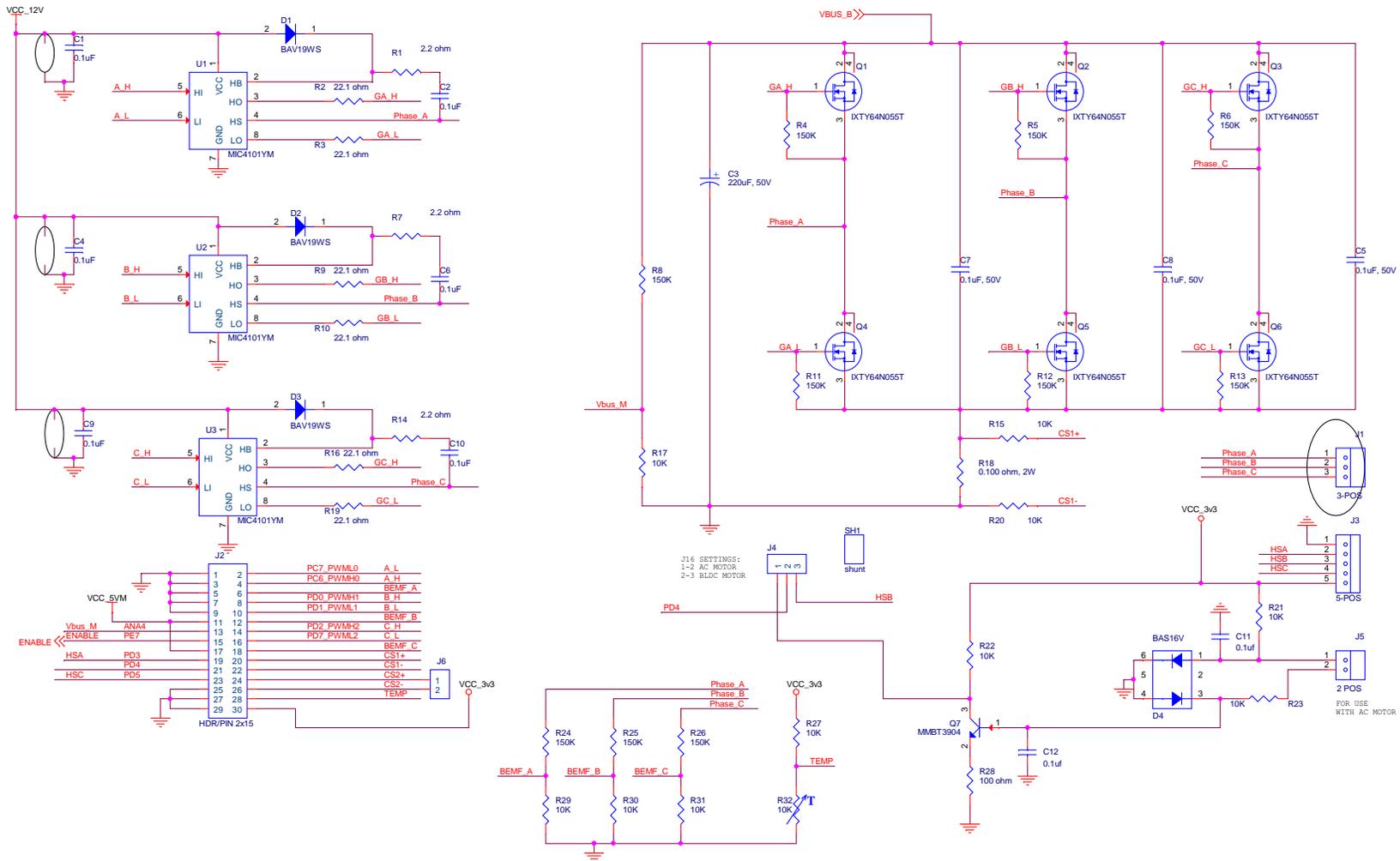


Figure 23. Space Vector Modulation Schematics, #3 of 4

Appendix F. Flow Charts

Figure 25 presents the typical flow of the space vector control routine.

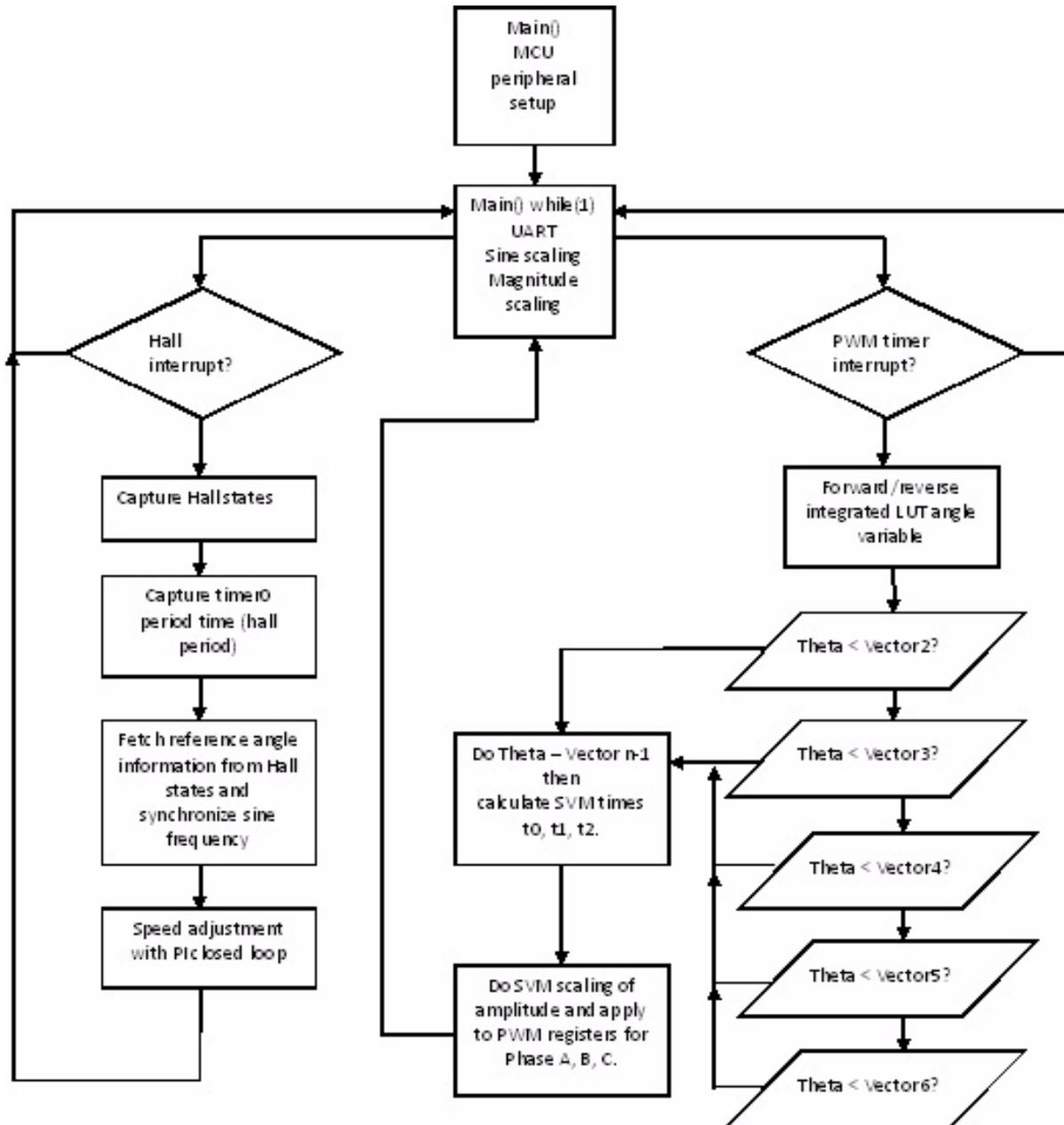


Figure 25. Space Vector Modulation Flow

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

©2015 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.

ZNEO and Z16FMC are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.