

## Bidirectional VF Control of Single and 3-Phase Induction Motors Using the PIC16F72

*Author: Padmaraja Yedamale  
Microchip Technology Inc.*

### INTRODUCTION

Single-phase induction motors are extensively used in appliances and industrial controls. The Permanent Split Capacitor (PSC) single-phase induction motor is the simplest and most widely used motor of this type. The classification, construction and working principle of single-phase induction motors are explained in detail in the application note “AC Induction Motor Fundamentals” (AN887) available from Microchip. For VF theory and basic operations, refer to AN887 mentioned above, and “Speed Control of 3-Phase Induction Motor Using PIC18 Microcontrollers” (AN843).

By design, PSC motors are unidirectional, which means they are designed to rotate in one direction. By adding either extra windings, and external relays and switches, or by using gear mechanisms, the direction of rotation can be changed. In this application note we will discuss in detail, how to control the speed of a PSC motor in both directions using a PIC16F72 microcontroller and power electronics.

The PIC16F72 microcontroller was chosen because it is one of the simplest and low-cost general purpose microcontrollers Microchip has in its portfolio. Even though it does not have the PWMs in hardware to drive complementary PWM outputs with dead band inserted, all PWMs are generated in firmware using timers and output to general purpose output pins.

### THEORY OF OPERATION

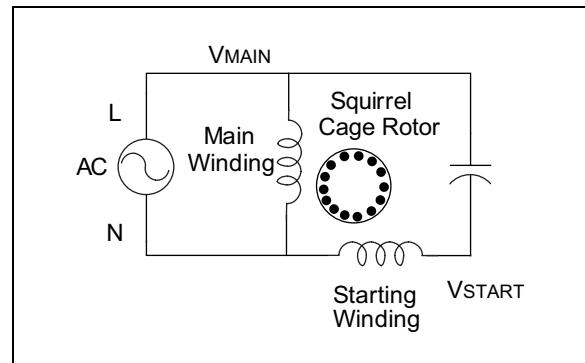
A PSC motor is usually a 2-phase asymmetrically wound motor. The main winding is designed to take the load current. The current flowing through the start winding is much less than the main winding. Therefore, the start winding will have a different electrical characteristic compared to the main winding. In order to produce the Magnetomotive Force (MMF) produced by the start winding very near to the main winding, the start winding has additional turns, higher resistance, and reduced current flowing through it. This makes the motor windings asymmetrical.

The motor is energized with a single-phase AC power supply, with a capacitor connected in series with the start winding (also called an auxiliary winding) as shown in Figure 1. The value of the capacitor is chosen so that the total impedance on the auxiliary winding produces sufficient phase shift in current to generate a rotating magnetic field in the air gap.

Typically, the current flowing through the start winding leads the current flowing through the main winding by 90 degrees.

By adding a microcontroller-based control circuit to the motor, the voltage across the main winding and start winding can be maintained at 90 degrees to each other. The other advantage is that the starting capacitor can be removed from the circuit, thus reducing the total system cost.

**FIGURE 1: ELECTRICAL EQUIVALENT CIRCUIT OF A PSC MOTOR**



A multiphase inverter can be used to drive the motor. Two methods are discussed in the next section.

## DRIVE TOPOLOGY

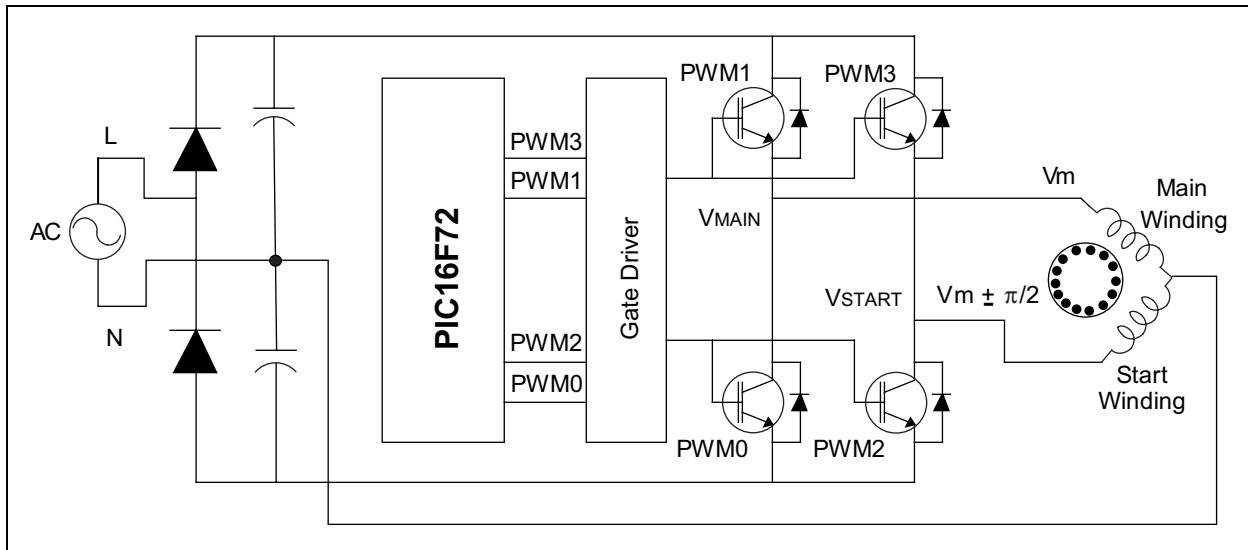
### Using an H-Bridge Inverter

The first approach is relatively easy as far as the power circuit and control circuit are concerned. On the input side, a voltage doubler is used and on the output side an H-bridge, or 2-phase inverter, is used as shown in Figure 2. One end of the main and start windings are connected to each half bridge and the other ends are connected together to the neutral point of the AC power supply, which also serves as the center point for the voltage doubler.

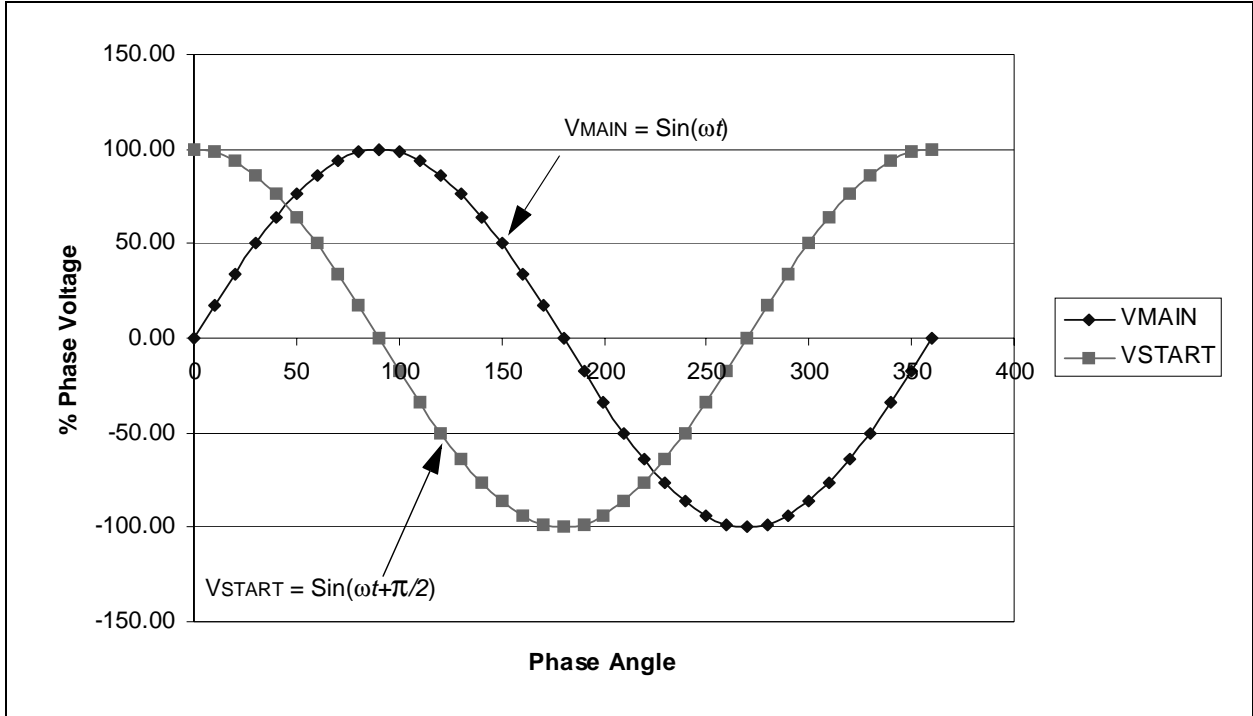
The control circuit requires four PWMs with two complementary pairs with sufficient dead band between the complementary outputs. PWM0-PWM1 and PWM2-PWM3 are the PWM pairs with dead band. The PIC16F72 does not have PWMs designed in the hardware to output the way we need. Therefore, the PWMs should be generated in firmware and output to the port pins. Using PWMs, the DC bus is synthesized to give two sine voltages at 90 degrees out of phase with varying amplitude and varying frequency according to the VF profile. If the voltage applied to the main winding lags the start winding by 90 degrees, the motor runs in one (i.e., forward) direction. To reverse the direction of rotation, the voltage supplied to the main winding should lead the voltage supplied to the start winding.

Figure 3 and Figure 4 show the main and start winding voltages in forward and reverse respectively.

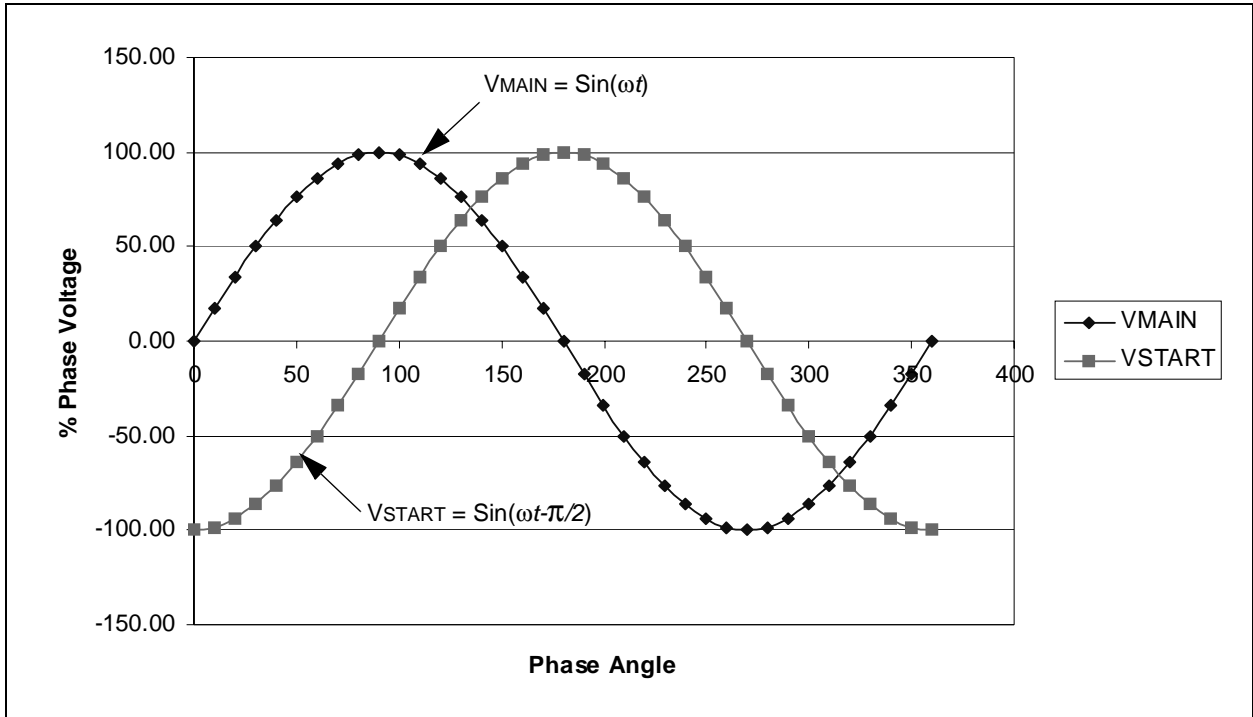
**FIGURE 2: PSC DRIVE WITH AN H-BRIDGE**



**FIGURE 3: MOTOR RUNNING IN FORWARD DIRECTION**



**FIGURE 4: MOTOR RUNNING IN REVERSE DIRECTION**



# AN967

This method of controlling a PSC type motor has a few disadvantages.

- Because the main winding and start windings have different electric characteristics, the current flowing through each switch is unbalanced. Over time, this may lead to premature breakdown of switching devices in the inverter.
- The common point of the windings is directly connected to the neutral power supply. This may increase the switching signal coming into the mains power supply and increase the noise emitted onto the line. This also may limit the EMI level of the product, violating certain design goals and regulations.
- The effective DC voltage handled is high due to the input voltage doubler circuit.
- Lastly, the cost of the voltage doubler circuit itself is high because of two large power capacitors.

A better solution to minimize these problems would be to use a 3-phase inverter bridge, as discussed in the next section.

## Using a 3-Phase Inverter Bridge

The input section is replaced with a standard diode bridge rectifier. The output section has a 3-phase inverter bridge. The main difference from the previous scheme is the way the motor windings are connected to the inverter. One end of the main winding and start windings are connected to one half bridge each. The other ends are tied together and connected to the third half bridge, as shown in Figure 5.

Refer to **Appendix B: “Schematics”** for the schematics that reference the PIC16F72 with a 3-phase inverter bridge.

With this drive topology, control becomes more efficient; however, the control algorithm becomes more complex. The voltages  $V_a$ ,  $V_b$  and  $V_c$  should be controlled to achieve the phase difference between the effective voltages across the main and start windings to have a 90 degree phase shift to each other.

The turn ratio of the start winding to the main winding is defined by:

### EQUATION 1:

$$\alpha = \frac{V_{MAIN}}{V_{START}}$$

where  $\alpha$  is the turn ratio, and  $V_{MAIN}$  and  $V_{START}$  are the effective voltage across the main winding and the start winding.

In order to have equal voltage stress on all devices, thus improving the device utilization and provide the maximum possible output voltage for a given DC bus voltage, all three inverter phase voltages are kept at the same amplitude as follows:

### EQUATION 2:

$$|V_a| = |V_b| = |V_c| = |V_l|$$

The effective voltage across the main and start winding is given as:

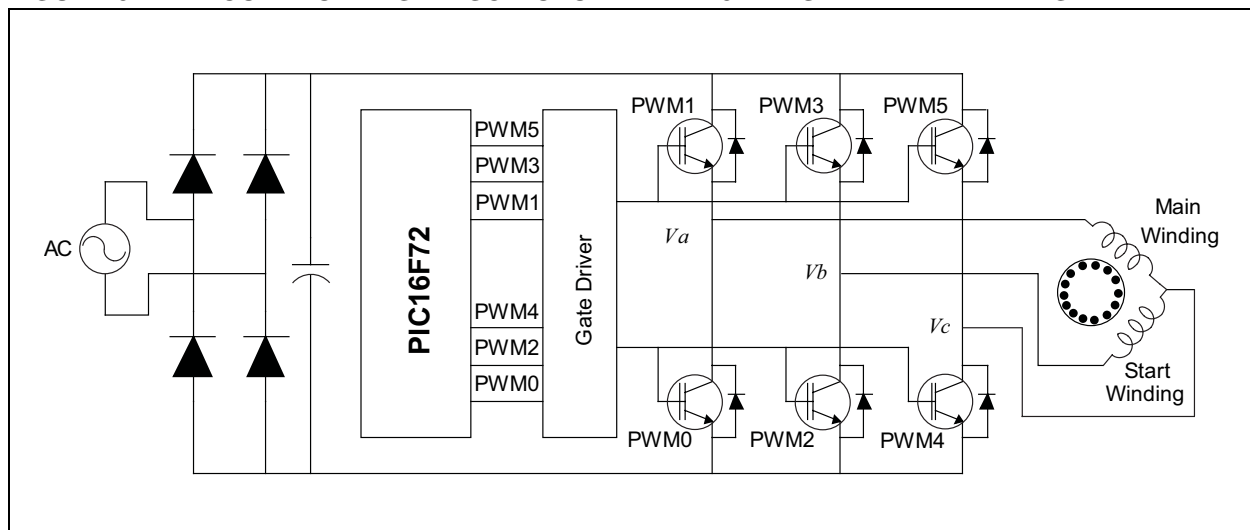
### EQUATION 3:

$$V_{MAIN} = V_a - V_c$$

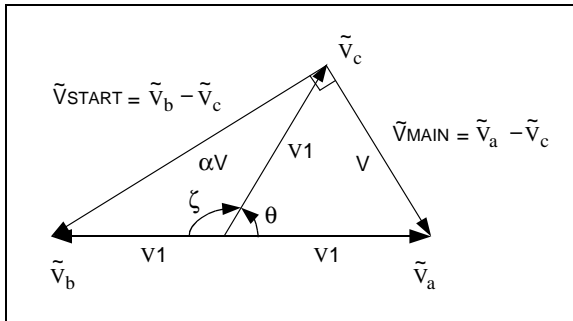
$$V_{START} = V_b - V_c$$

The voltages are shown in the phasor diagram in Figure 6.

**FIGURE 5: CONTROLLING A PSC MOTOR WITH A 3-PHASE INVERTER BRIDGE**



**FIGURE 6: VOLTAGE PHASOR DIAGRAM**



As seen in the phasor diagram in Figure 6, the voltages across phase A and phase B are out of phase, and the phase difference between phase A and phase C is  $\theta$  degrees. By applying basic trigonometry,  $\theta$  can be calculated by:

**EQUATION 4:**

$$\text{Angle } \theta = 180^\circ - 2 \tan^{-1}(\alpha).$$

By applying the Pythagorean Theorem, the voltage vector  $V_1$  can be calculated as:

**EQUATION 5:**

$$V_1 = V \times \frac{\sqrt{1 + \alpha^2}}{2}$$

Because the turn ratio remains constant for a given motor,  $\alpha$  can be a compile time option. With this,  $\theta$  and  $V_1$  can be precomputed for a given motor. This simplifies the run time calculation. Based on the phase angle, phase voltages  $V_a$ ,  $V_b$  and  $V_c$  can be calculated as:

**EQUATION 6:**

$$V_a = V_1 \times \cos(\omega t) + \frac{V_{dc}}{2}$$

$$V_b = -V_1 \times \cos(\omega t) + \frac{V_{dc}}{2}$$

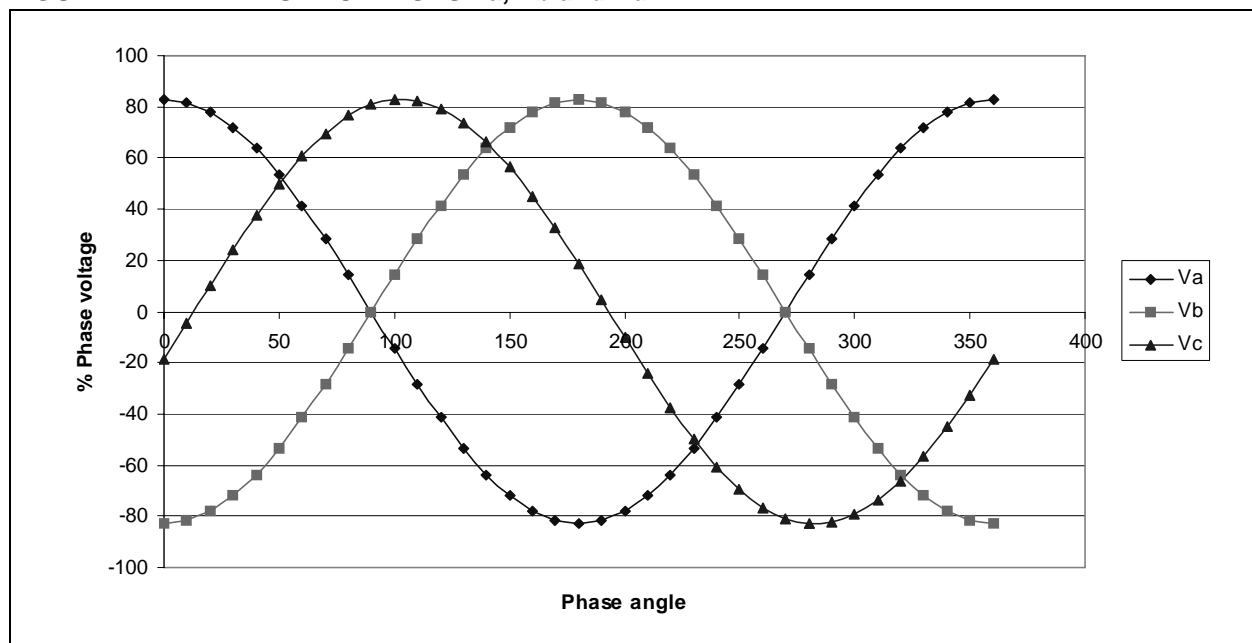
$$V_c = V_1 \times \cos(\omega t \pm \theta) + \frac{V_{dc}}{2}$$

$V_{dc}$  is the DC bus voltage, and  $\omega t$  is the angular velocity of the electrical cycle. The direction of rotation can be easily controlled by adding or subtracting  $\theta$  in the  $V_c$  calculation.

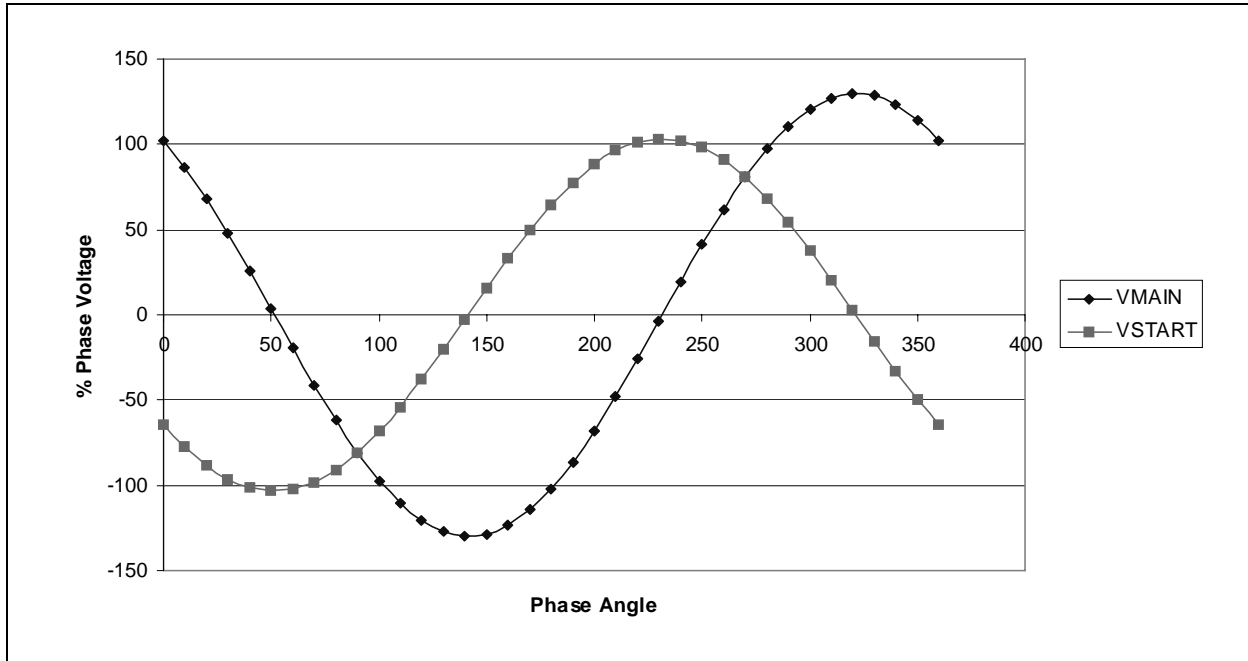
Figure 7 shows the phase voltages  $V_a$ ,  $V_b$  and  $V_c$ , and Figure 8 shows the effective voltages across the main winding ( $V_{MAIN}$ ) and the start winding ( $V_{START}$ ).

Figure 8 also shows that the effective phase difference between the voltages is 90 degrees and the effective voltage ratio is  $\alpha$ .

**FIGURE 7: PHASE VOLTAGES  $V_a$ ,  $V_b$  and  $V_c$**



**FIGURE 8: VOLTAGE ACROSS MAIN WINDING (V<sub>MAIN</sub>) AND START WINDING (V<sub>START</sub>)**



## IMPLEMENTATION USING THE PIC16F72

In order to control a 3-phase inverter bridge, we need three PWM pairs with complementary outputs. In addition, each complementary pair of PWMs needs dead time in between any OFF and ON switch events to avoid a DC bus short circuit. The PIC16F72 does not have these features in the hardware. However, this can be easily implemented using a general purpose timer and six output pins as shown in Figure 9.

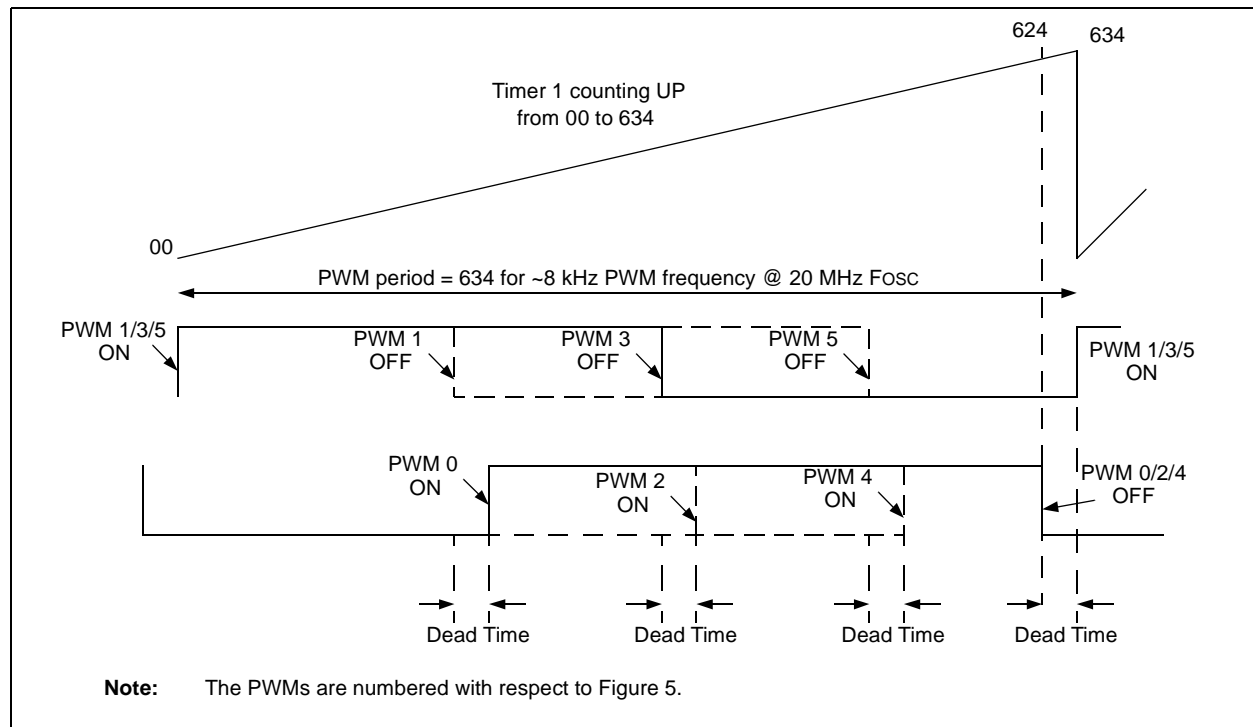
### Generating Software PWMs

The scheme shown here gives a fixed PWM frequency of approximately 7.9 kHz. Timer1 (1:4 prescale) is counted up from 00 to 634. At the beginning of the cycle (when Timer1 = 00), the PWMs controlling the upper switches (i.e., PWM1, PWM3 and PWM5) are turned ON. Based on the individual PWM duty cycle, the corresponding PWM output is turned OFF.

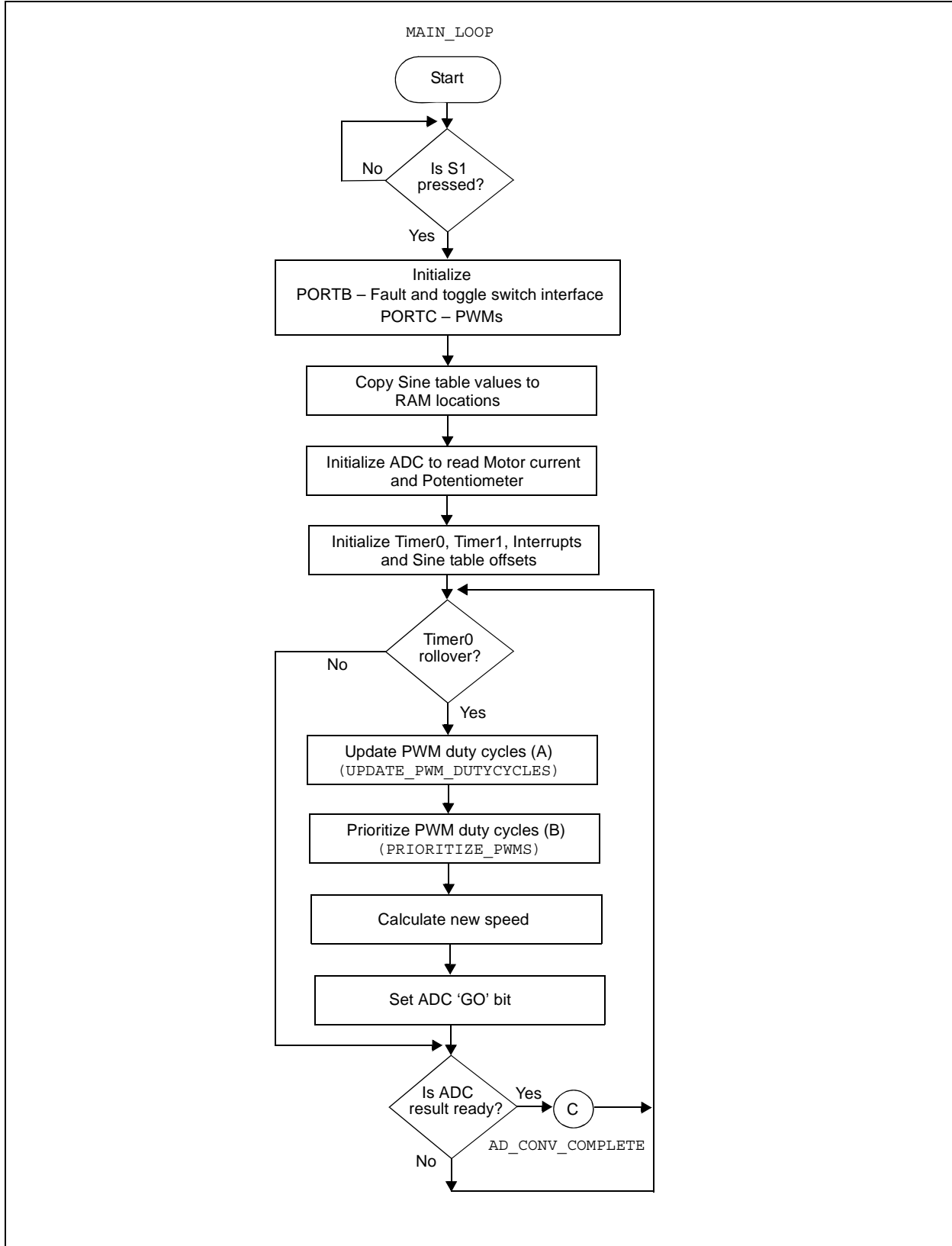
After five instruction cycles, the complementary PWM is turned ON. This gives a dead time of 1  $\mu$ s when the microcontroller is running at 20 MHz. When the count reaches 624, all outputs are turned OFF. A new PWM cycle starts after 10 instruction cycles.

In the source code provided with this application note (see **Appendix A: "Source Code"**), instead of comparing the Timer1 values corresponding to the duty cycle values, a relative count is calculated with respect to the previous PWM duty cycle. This count is subtracted from 0xFF (Timer1 is configured as an 8-bit timer) and loaded to the Timer1 register. When the Timer1 value overflows, in Timer1 overflow ISR, the corresponding Odd PWM is turned OFF and the corresponding Even PWM is turned ON after five instruction cycles. This is done for all three pairs of PWM. Refer to the flowcharts in Figure 10 through Figure 11 for more details.

**FIGURE 9: PWM SOFTWARE IMPLEMENTATION**

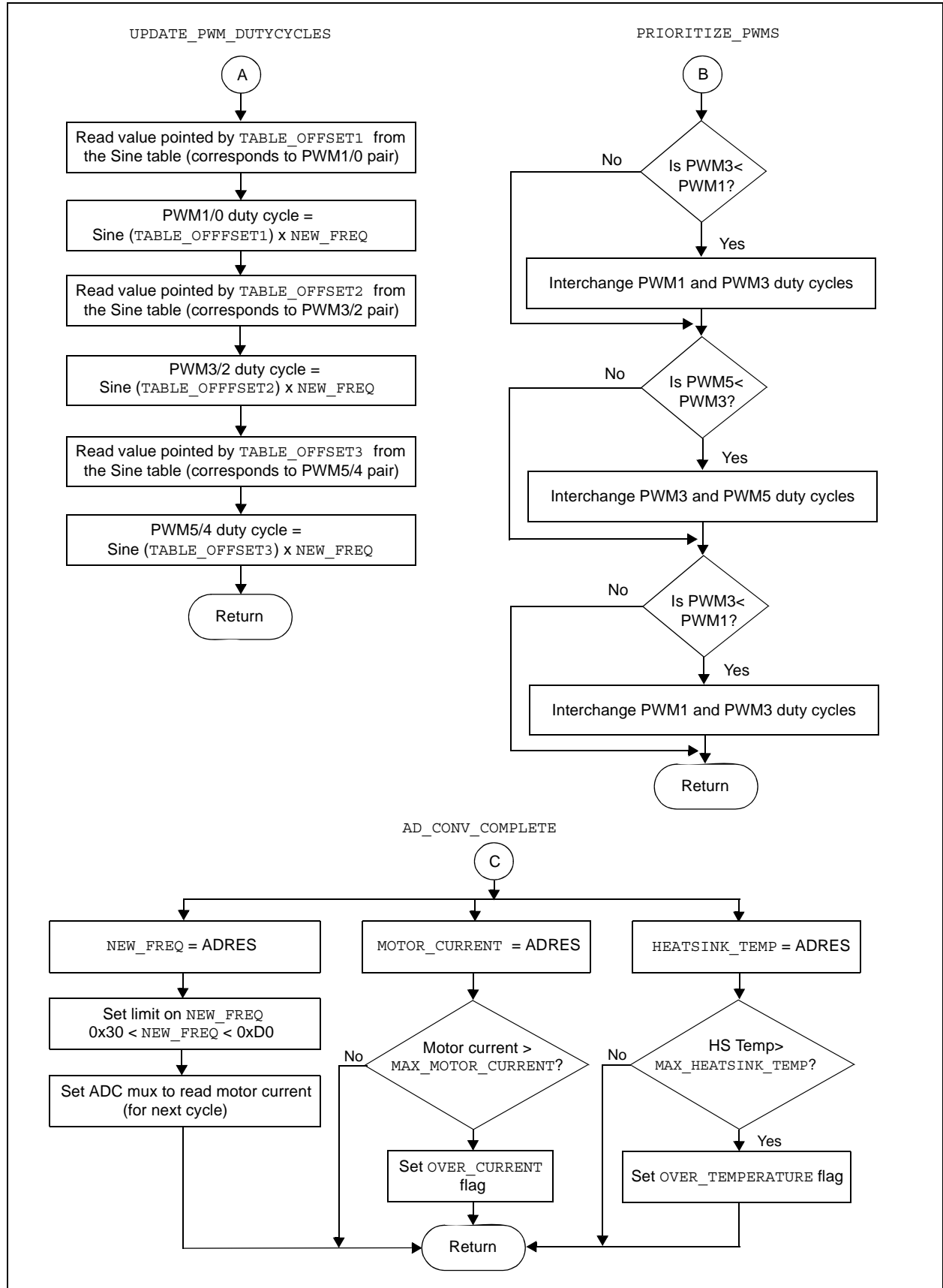


**FIGURE 10: MAIN LOOP**

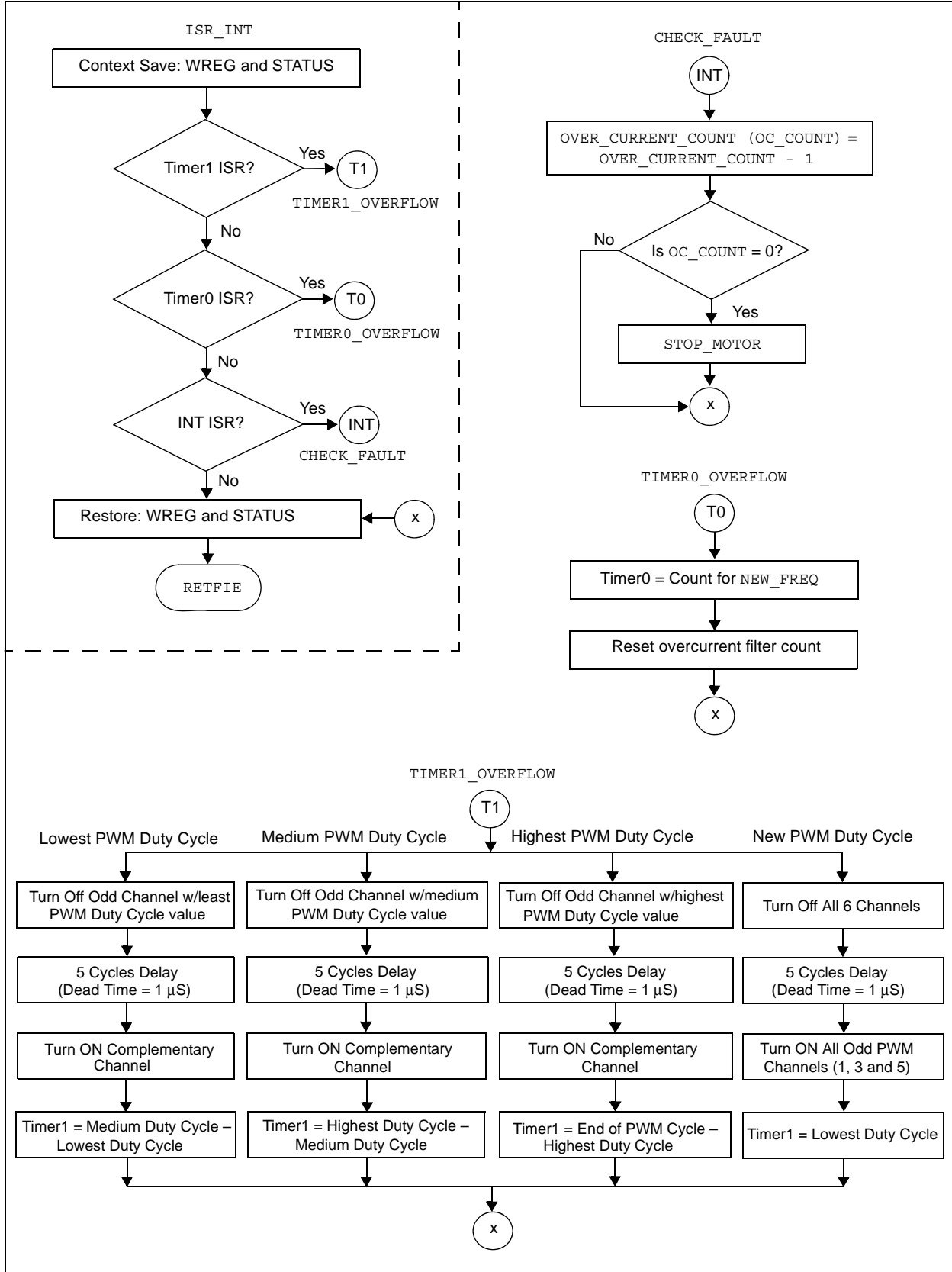




**FIGURE 10: MAIN LOOP (CONTINUED)**



**FIGURE 11: INTERRUPT SERVICE ROUTINES (ISR)**



## Initialization

- PORTC<0:5> are initialized to output PWMs.
- Timer1 is initialized with 8-bit operation and 1:4 prescale.
- A/D channels are initialized to read frequency reference (AN0), motor current (AN1) and heatsink temperature (AN2).
  - Frequency reference is read using a potentiometer connected to A/D Channel 0.
  - Motor current is read using a shunt resistor in the DC return path. The voltage corresponding to the motor current is amplified and connected to A/D channel 1.
- Timer0 is used for setting the motor frequency based on the potentiometer setting. On every Timer0 overflow, new PWM duty cycles are advanced by 10 electrical degrees on the Sine table.
- The Sine table is loaded into RAM
- Timer0 overflow, Timer1 overflow and INT interrupts are enabled.

## Main Routine

These tasks are done in the MAIN\_LOOP routine:

- New PWM duty cycle is calculated by the subroutine UPDATE\_PWM\_DUTYCYCLES  
 Three pointers pointing to three different values on the Sine table corresponding to the phase difference between  $V_a$ ,  $V_b$  and  $V_c$ . The sine table is drawn to maximum duty cycle available when the sine value reaches 90 degrees. Every value is scaled down based on the frequency input to follow a linear VF profile.
- PWM duty cycle sorting is handled by the subroutine PRIORITIZE\_PWMs  
 PWM duty cycles calculated earlier are sorted in ascending order, so that the duty cycle with minimum ON time can be addressed first and PWM with maximum duty cycle last. Corresponding Flags are set to indicate which PWM duty cycle corresponds to which PWM output.

- Timer0 reload value is calculated by the subroutine TIMER0\_OVERFLOW  
 Timer0 is used for setting the motor frequency. The Timer0 reload value is calculated based on three factors: first is the frequency reference input from the potentiometer, second is the number of sine table values, and third is the MCU operating frequency.
- Polling for the ADC result is handled by the subroutine AD\_CONV\_COMPLETE  
 ADC conversion is poled in the main routine. Alternatively, frequency reference (AN0), motor current (AN1) and heatsink temperature (AN2) are selected and converted.

## Interrupt Service Routines (ISRs)

- **Timer1 ISR:** In the first three Timer1 overflow ISRs, the corresponding Odd PWM output is turned off in each ISR. The complementary output is turned on after a dead time of five cycles (1  $\mu$ s). In the fourth Timer1 overflow ISR, the PWM cycle is restarted. All PWMs are turned OFF and the timer is loaded with the value corresponding to the lowest duty cycle value. This is repeated for each PWM cycle.
- **Timer0 ISR:** A flag is set to indicate that the Sine output should advance by 10 degrees on the Sine table. The Timer0 registers are reloaded with the value corresponding to the motor frequency reference.
- **INT ISR:** The INT pin is used to interface hardware overcurrent fault. Motor current is compared with a fixed voltage reference using an op amp comparator. Each time the motor current exceeds the reference, in INT ISR a count (OC\_COUNT) is decremented. If the count reaches zero in one Timer0 cycle, then the motor is stopped and overcurrent is indicated. This count is reset in every Timer0 ISR.

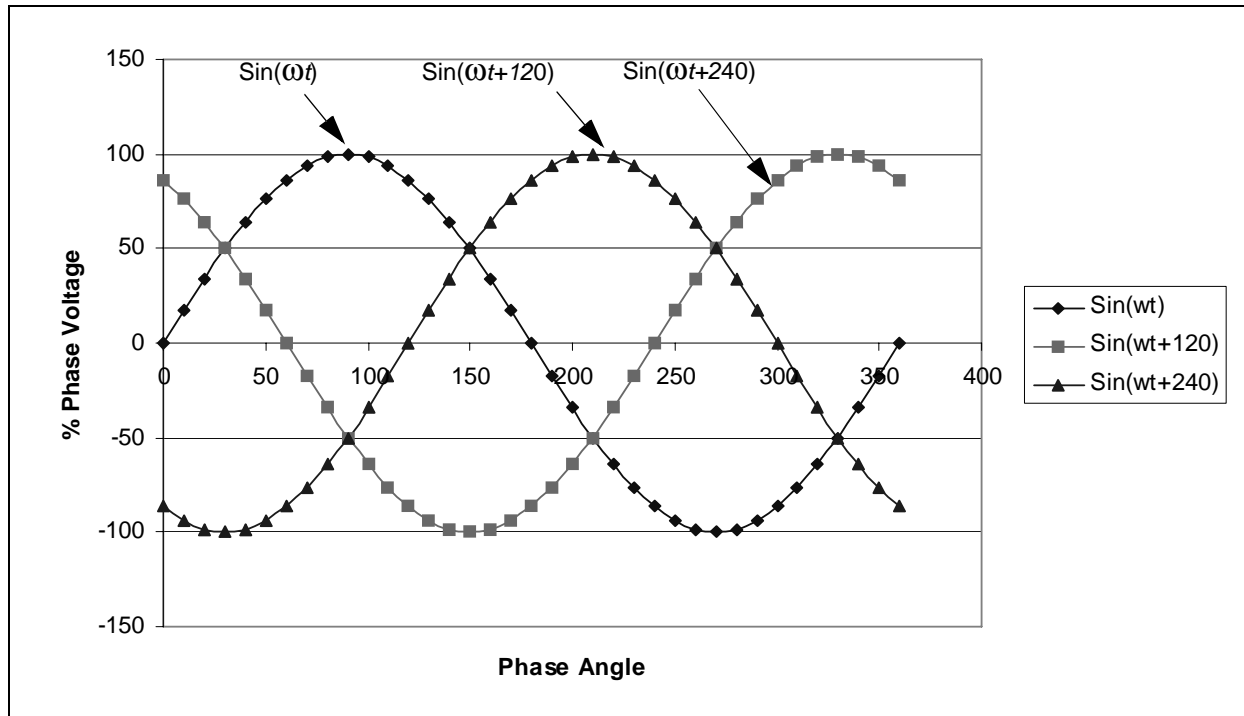
# AN967

## Controlling a 3-Phase Induction Motor

The advantage of the drive topology shown in Figure 5 is the same hardware can be used to control the speed of a 3-phase induction motor. In order to control a 3-phase induction motor with a Variable Frequency (VF) technique, three pairs of complementary PWMs with dead band are required.

The DC bus is modulated with these PWMs to give three Sine voltage outputs with 120 degrees phase shift to each other as shown in Figure 12. This can be done by changing the offset pointers on the Sine table. Firmware needs to be recompiled and the part should be reprogrammed. To reverse the motor direction, the PWM loaded to two of the phases should be interchanged.

FIGURE 12: PHASE VOLTAGE FOR 3-PHASE INDUCTION MOTOR CONTROL



## PIC16F72 LIMITATIONS

Using the PIC16F72 for control has the following limitations:

- **MIPS:** Generating a software PWM takes significant processing power. This limits any other application that may be required to run on the same MCU.
- **PWM Frequency Range:** The PWM frequency range is limited and requires firmware changes. The safe PWM frequency range is from 6 kHz to 10 kHz with an operating frequency of 20 MHz. The PWM cycle may have an error up to  $\pm 5\%$ . The resolution is also limited between 6 to 8 bits. In addition to this, generating PWM outputs in firmware uses the on-chip Timer, program and data memory.

## CONTROL USING A PIC18F2431/4431

The PIC18FXX31 MCUs have a state-of-the-art Power Control PWM (PCPWM) module on-chip. This module can provide three pairs of complementary PWMs with programmable dead band. Programming is greatly simplified using PIC18FXX31 devices, because it reduces the task of generating the software PWMs, which takes most of the processor resource on a PIC16F72. The source code included with this application note (see **Appendix A: "Source Code"**) also includes single-phase VF control using the PIC18F2431.

To learn more about using PIC18FXX31 microcontrollers for different types of motor controls, refer to application notes AN899, *Brushless DC Motor Control Using PIC18FXX31 Microcontrollers* (DS00899) and AN900, *Controlling 3-Phase AC Induction Motors Using the PIC18F4431* (DS00900) available from Microchip.

## CONCLUSION

Microcontroller-based control for a PSC motor makes the system easy to implement and have control over the motor in two directions. Implementing the algorithm using a 3-phase inverter bridge gives flexibility and efficiency of control. In addition, a 3-phase induction motor can be controlled using the same hardware by making minor modifications to the firmware. The PIC16F72 is a popular low cost general purpose microcontroller from Microchip, that can be used to implement the control algorithm. To a great extent, PIC18FXX31 devices can provide flexibility in implementing motor control algorithms.

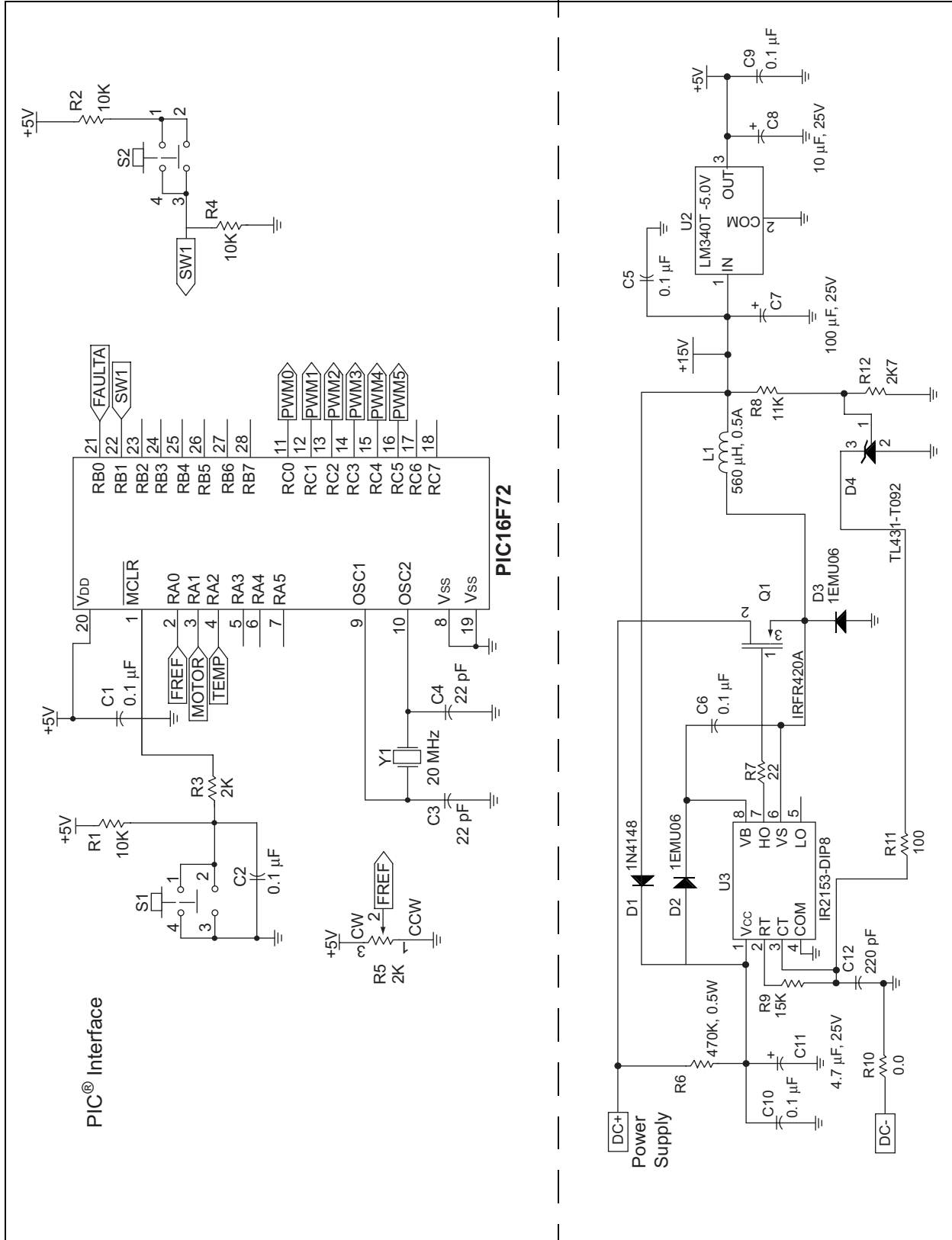
## APPENDIX A: SOURCE CODE

The complete source code is available for download as a single archive file from the Microchip corporate web site at:

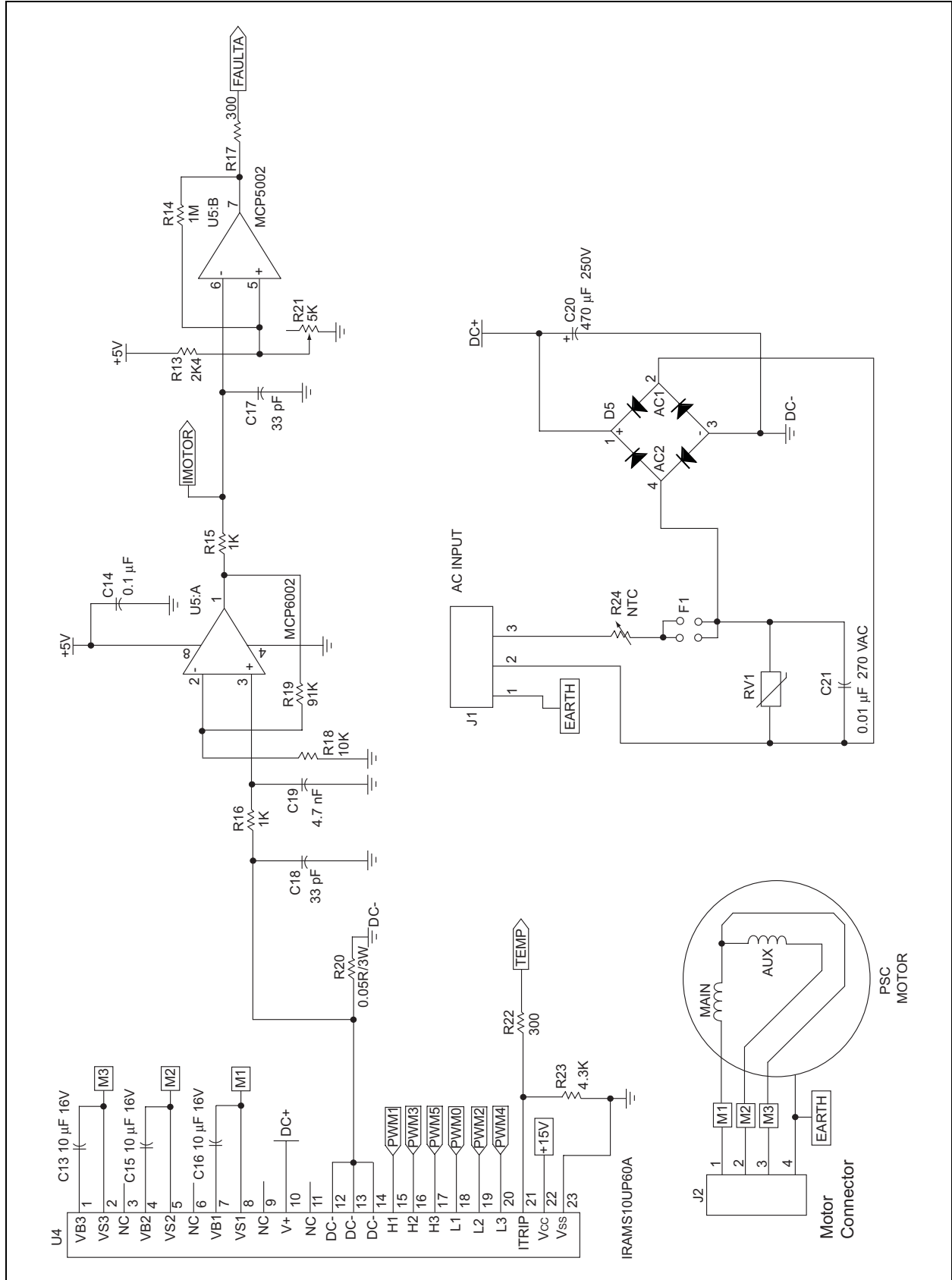
[www.microchip.com](http://www.microchip.com)

## APPENDIX B: SCHEMATICS

FIGURE B-1: PIC® INTERFACE AND POWER SUPPLY (SHEET 1 OF 2)



**FIGURE B-2: POWER SECTIONS AND MOTOR CURRENT MEASUREMENT (SHEET 2 OF 2)**



# AN967

---

NOTES:



---

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, MPASM, MPLIB, MPLINK, MPSIM, PICKit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2005, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949:2002 ==**

*Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*



---

---

## Worldwide Sales and Service

---

---

### AMERICAS

**Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://support.microchip.com>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

**Atlanta**  
Alpharetta, GA  
Tel: 770-640-0034  
Fax: 770-640-0307

**Boston**  
Westford, MA  
Tel: 978-692-3848  
Fax: 978-692-3821

**Chicago**  
Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

**Detroit**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

**Kokomo**  
Kokomo, IN  
Tel: 765-864-8360  
Fax: 765-864-8387

**Los Angeles**  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

**San Jose**  
Mountain View, CA  
Tel: 650-215-1444  
Fax: 650-961-0286

**Toronto**  
Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

**Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**China - Beijing**  
Tel: 86-10-8528-2100  
Fax: 86-10-8528-2104

**China - Chengdu**  
Tel: 86-28-8676-6200  
Fax: 86-28-8676-6599

**China - Fuzhou**  
Tel: 86-591-8750-3506  
Fax: 86-591-8750-3521

**China - Hong Kong SAR**  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**China - Shanghai**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**China - Shenyang**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**China - Shenzhen**  
Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

**China - Shunde**  
Tel: 86-757-2839-5507  
Fax: 86-757-2839-5571

**China - Qingdao**  
Tel: 86-532-502-7355  
Fax: 86-532-502-7205

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-2229-0061  
Fax: 91-80-2229-0062

**India - New Delhi**  
Tel: 91-11-5160-8631  
Fax: 91-11-5160-8632

**Japan - Kanagawa**  
Tel: 81-45-471- 6166  
Fax: 81-45-471-6122

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

**Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**Taiwan - Kaohsiung**  
Tel: 886-7-536-4818  
Fax: 886-7-536-4803

**Taiwan - Taipei**  
Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

**Taiwan - Hsinchu**  
Tel: 886-3-572-9526  
Fax: 886-3-572-6459

### EUROPE

**Austria - Weis**  
Tel: 43-7242-2244-399  
Fax: 43-7242-2244-393

**Denmark - Ballerup**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**France - Massy**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Ismaning**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**England - Berkshire**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820