MOTOROLA
intelligence everywhere™

digital dna™

3-Phase AC Induction Motor Drive with Dead Time Distortion Correction Using the MC68HC908MR32

Designer Reference Manual

M68HC08 Microcontrollers

DRM019/D
Rev. 0, 03/2003

MOTOROLA.COM/SEMICONDUCTORS

3-Phase AC Induction Motor Drive with Dead Time Distortion Correction Reference Design

Designer Reference Manual — Rev 0

by:

Radim Visinka, Ph.D.
Motorola Czech Systems Laboratories
Roznov pod Radhostem, Czech Republic

Revision history

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.motorola.com/semiconductors

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision history

Date	Revision Level	Description	Page Number(s)
January 2003	1	Initial version	N/A

List of Sections

Section 1. Introduction 13

Section 2. System Description. 15

Section 3. Hardware Design. 27

Section 4. Software Design 37

Section 5. System Setup 49

Appendix A. References. 61

Appendix B. Glossary. 63

List of Sections

Table of Contents

Section 1. Introduction

1.1 Contents 13

1.2 Application Functionality. 13

1.3 Benefits of the Solution. 13

Section 2. System Description

2.1 Contents 15

2.2 System Concept 15

2.3 Dead Time Distortion Correction 18

Section 3. Hardware Design

3.1 Contents 27

3.2 System Configuration 27

3.3 MC68HC908MR32 Control Board 29

3.4 3-Phase AC BLDC High Voltage Power Stage. 31

3.5 Optoisolation Board 33

3.6 Motor-Brake Specifications. 35

3.7 Hardware Documentation. 36

Section 4. Software Design

4.1 Contents 37

4.2 Introduction. 37

4.3 Data Flow 37

4.4 Algorithm of Dead Time Distortion Correction 43

Section 5. System Setup

5.1 Contents 49

5.2 Hardware Setup 49

5.3 Warning 50

5.4 Jumper Settings of Controller Board. 51

5.5 Required Software Tools 52

5.6 Building the Application 53

5.7 Executing the Application 53

5.8 Controlling the Application with PC Master Software 57

Appendix A. References

Appendix B. Glossary

List of Figures

Figure	Title	Page
2-1	System Block Diagram	16
2-2	Volt per Hertz Ramp	17
2-3	Dead Time Distortion	19
2-4	Topology of Current Polarity Sensing	21
2-5	Proposed Current Threshold for Correction Toggling	24
3-1	Hardware Configuration	28
3-2	MC68HC908MR32 Control Board	30
3-3	3-Phase AC High Voltage Power Stage	32
4-1	Data Flow	38
4-2	3-Phase Sine Waves with Amplitude of 50%	40
4-3	mcgen3PhWaveSine Data Explanation - Phase A	41
4-4	Dead Time Correction State Machine.	45
5-1	Setup of the Application	50
5-2	MC68HC908MR32 Jumper Reference.	52
5-3	Execute Make Command	53
5-4	Control Elements	54
5-5	USER LEDs, PWM LEDs, and RESET	55
5-6	PC Master Software Control Window	58

List of Figures

List of Tables

Table	Title	Page
2-1	PWM values loaded into registers PVAL1-6	22
2-2	PWM Prescaler	22
2-3	Sensing of the Current Polarity and Magnitude for Ph. 1	25
3-1	Electrical Characteristics of Control Board	30
3-2	Electrical Characteristics of Power Stage	33
3-3	Electrical Characteristics of Optoisolation Board	34
3-4	Motor - Brake Specifications	35
4-1	State Machine Flag Registers dtStateFlagsAB	46
4-2	State Machine Flag Registers dtStateFlagsC	46
4-3	dtCorrect_s Structure Elements	47
5-1	MC68HC908MR32EVM Jumper Settings	52
5-2	Motor Application States	56
5-3	Dead Time Distortion Correction	57

List of Tables

Section 1. Introduction

1.1 Contents

1.2 Application Functionality 13

1.3 Benefits of the Solution. 13

1.2 Application Functionality

This Reference Design describes the design of a 3-phase AC induction motor drive with dead time distortion correction. It is based on Motorola's MC68HC908MR32 microcontroller which is dedicated for motor control applications. The system is designed as a motor drive system for medium power three-phase AC induction motors and is targeted for applications in both industrial and appliance fields (e.g. washing machines, compressors, air conditioning units, pumps or simple industrial drives). The reference design incorporates both hardware and software parts of the system including hardware schematics with a bill of material, and a software listing.

1.3 Benefits of the Solution

The design of very low cost variable speed 3-phase motor AC control drives has become a prime focus point for the appliance designers and semiconductor suppliers. Replacing variable speed universal motors by maintenance-free, low noise asynchronous (induction) motors is a trend that supposes total system costs being equivalent.

Six-transistor inverter is the most used topology for AC motor drives. The dead time must be inserted between the turning off of one transistor in the inverter half bridge and turning on of the complementary transistor.

The dead time causes distortion to the generated voltage, and thus a non-sinusoidal phase current.

This distortion causes distortion of the motor performance. It is especially apparent in low speeds, when the dead time is comparable with the PWM pulse width. Also, the longer the dead time, the higher the influence it has over the motor performance.

Dead time distortion can be corrected by properly modulating the power stage control signals. The advantages of dead time distortion correction are:

- Smoother running motors
- Less torque ripple
- Quieter motors
- More efficient operation (less harmonic losses).

Section 2. System Description

2.1 Contents

2.2 System Concept 15

2.3 Dead Time Distortion Correction 18

2.2 System Concept

The application is designed to drive a 3-phase AC motor in an open speed loop mode with dead time distortion correction (see **Figure 2-1**). The desired speed is set-up in the user interface. The desired frequency and amplitude of the motor voltage sine wave is calculated according to the desired speed using Volt-per-Hertz table. The sine wave generator generates the PWM values for all three phases of the AC bridge inverter according to the selected type of dead time distortion correction algorithm.

The system incorporates the following hardware blocks:

- power supply rectifier,
- three-phase inverter including optoisolation,
- feedback sensors: DC-Bus voltage, DC-Bus current, temperature, polarity of phase currents,
- microcontroller MC68HC908MR32.

System Description

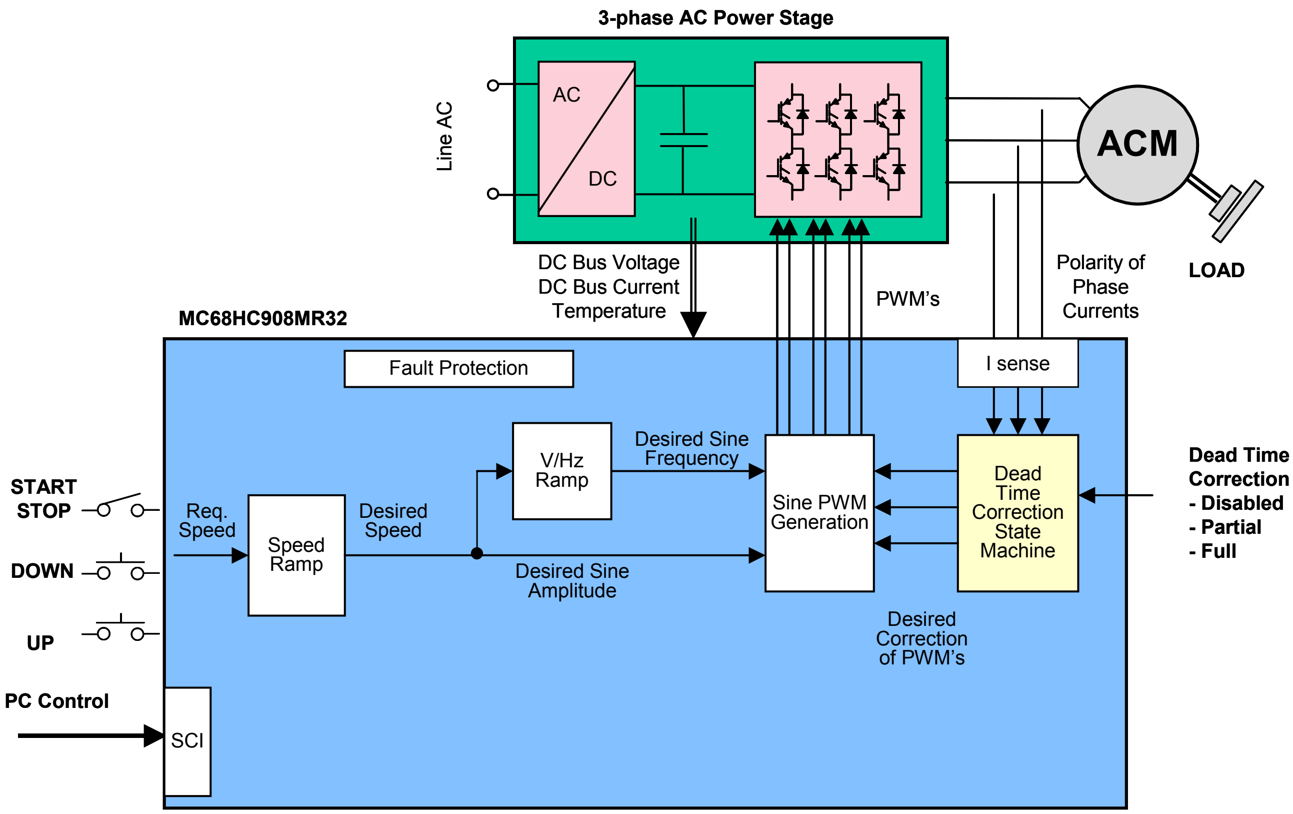

Figure 2-1. System Block Diagram

The drive is designed as a "Volt-per-Hertz" drive. It means that the control algorithm keeps constant magnetizing current (flux) of the motor by varying the stator voltage with frequency. The commonly used Volt-per-Hertz ramp of a 3-phase AC induction motor illustrates **Figure 2-2**.

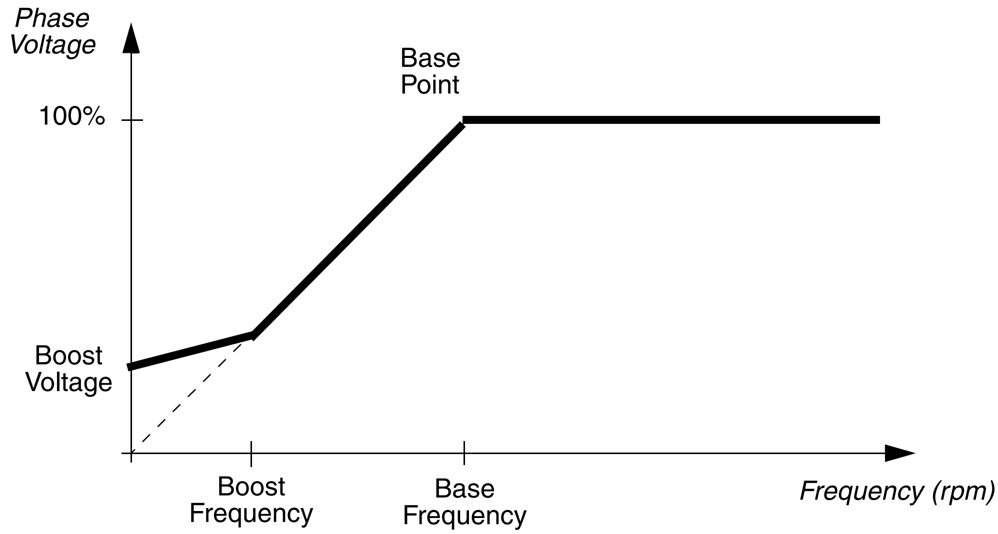

Figure 2-2. Volt per Hertz Ramp

The Volt per Hertz ramp is defined by following parameters:

- Base Point - defined by Base Frequency (usually 50Hz or 60Hz)
- Boost - Defined by Boost Voltage and Boost Frequency

The ramp profile is set to the specific motor and can be easily changed to accommodate different ones.

The dead time distortion correction algorithms provide a correction of the PWM values with respect to the actual polarity of the phase currents. The current polarity is evaluated by sensing the phase voltage during the dead time and is carried out by the on-chip circuitry of the 908MR32 microcontroller. Two types of dead time distortion correction algorithms are implemented - partial, and full correction. The **partial** correction algorithm detects just the current polarity and the correction is done almost entirely by the on-chip PWM hardware. On the other hand, the full correction algorithm also detects the magnitude of the phase currents (low/high), and implements advanced s/w which improves the correction results. The user has the choice of selecting either of the correction algorithms. The type of dead time distortion correction is indicated by a yellow LED on 908MR32 controller board.

The PWM frequency can be changed at any time during the motor operation to one of the following values:

- 4kHz
- 8kHz
- 16kHz
- 32kHz

The drive incorporates fault protection, so in the case of DC-Bus over-current, DC-Bus over-voltage, or DC-Bus under-voltage faults, internal fault logic is asserted and the application enters a fault state. This state can be exited only if the fault disappears and it is acknowledged, by toggling the START/STOP switch through the STOP state. The application states are displayed by green LED on 908MR32 control board.

The application can operate in two modes:

1. Manual Operating Mode

   The drive is controlled by the START/STOP switch. The direction of the motor rotation is set by the FWD/REV switch. The motor speed is set by the SPEED potentiometer.
2. PC Master Software (Remote) Operating Mode

   The drive is controlled remotely from a PC through the serial communications interface (SCI) communication channel of the MCU device via an RS-232 physical interface. The drive is enabled by the START/STOP switch, which can be used to safely stop the application at any time.

2.3 Dead Time Distortion Correction

Six-transistor inverter is the most used topology for AC motor drives. The dead time must be inserted between the turning off of one transistor in the inverter half bridge and turning on of the complementary transistor. The dead time causes distortion to the generated voltage, and thus a non-sinusoidal phase current.

In order to achieve a sinusoidal phase current, and thus limit the harmonic losses, noise, and torque ripple, the dead time distortion correction needs to be implemented. The on-chip Pulse-Width-Modulation (PWM) module, of the MC68HC908MRxx family of Motorola microcontrollers, contains the patented hardware block that simplifies the task.

The dead time correction is based on the evaluation of the phase current polarity of the respective phase, and proper counter-modulation of the dead-time distortion. The basic situation is shown in **Figure 2-3**. The desired load voltage is affected by the dead time. During dead time, load inductance defines the voltage needed to keep inductive current flowing through diodes. So full positive or full negative voltage is applied to the phase, according to the phase current polarity. For positive current (i+), the actual voltage pulses are shortened by dead time, for negative phase current the voltage pulses are lengthened by dead time.

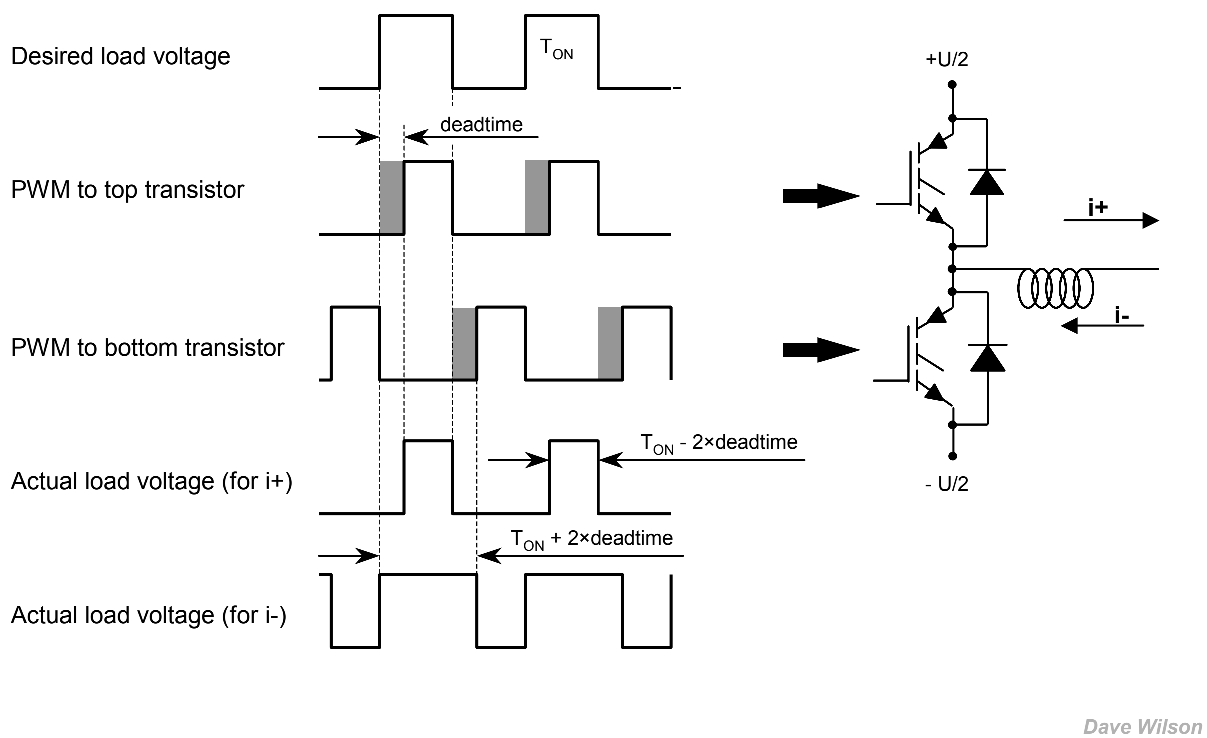

Figure 2-3. Dead Time Distortion

To achieve distortion correction, one of two different correction factors must be added to the desired PWM value, depending on whether the top or bottom transistor is controlling the output voltage during the dead time.

When the voltage pulse is shortened due to dead time, the control PWM signal is extended by dead time, so the actual voltage pulse matches the desired voltage. Vice versa, when the voltage pulse is lenghtened due to dead time, the control PWM signal is shortened by dead time, so again the actual voltage pulse matches the desired voltage. Therefore the actual signal equals the desired one, and the generated phase current is sinusoidal.

The dead time distortion correction utilizes phase current sensing. The on-chip PWM module of MC68HC908MRxx microcontrollers contains the block that enables them to evaluate the polarity and the size of the phase current without the need of an expensive current sensor. It is based on the sampling and evaluation of the phase voltage level during the dead time. The zero voltage during dead time reflects a positive phase current, the full DC-Bus voltage during dead time reflects a negative phase current. So comparing the phase voltage with the half DC-Bus voltage enables an evaluation of the current polarity. The topology is illustrated in **Figure 2-4**. The output of the comparator is connected to the current polarity sensing input of the MC68HC908MR32 microcontroller. The microcontroller contains the hardware that samples the current sensing inputs during dead time. It enables evaluation of the current polarity and also the region of low currents.

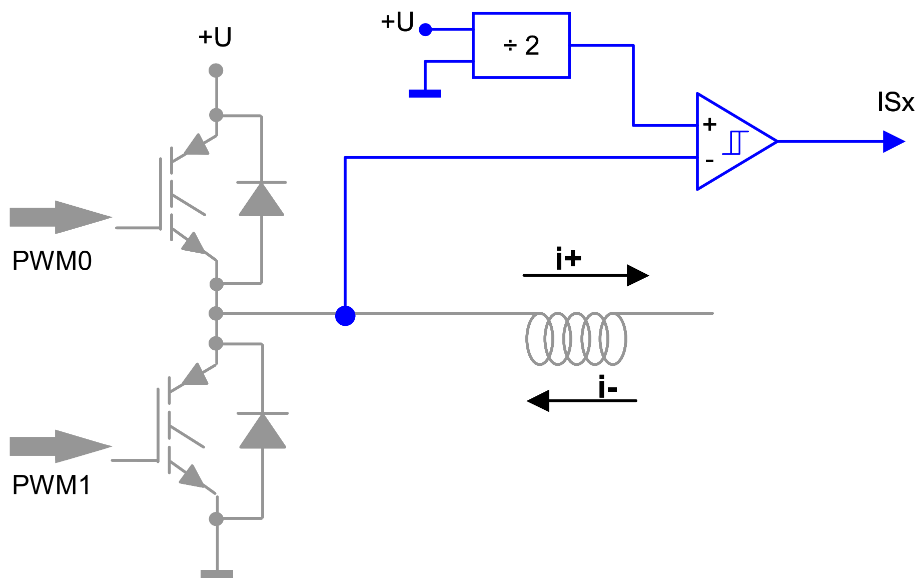

Figure 2-4. Topology of Current Polarity Sensing

During PWM reload ISR, the desired PWM values for all three phases are calculated as:

- PWM1 for phase 1
- PWM2 for phase 2
- PWM3 for phase 3

The values loaded into the individual PVAL registers of the separate phases are shown in **Table 2-1**. Since AC motor control utilizes center-aligned PWM modulation, only half of the dead time needs to be added to / substracted from the desired PWM duty cycle to achieve the distortion correction. Without dead time correction, the even PVAL registers are loaded with the required PWM value, but the odd PVAL registers are not used. When dead time correction is used, the even PVAL registers are loaded with the desired PWM *plus* half of the dead time (PWMx+DT/2), while the odd PVAL registers are loaded with the desired PWM *minus* half of the dead time (PWMx-DT/2).

Table 2-1. PWM values loaded into registers PVAL1-6

Phase	PVAL register	Required values in PVAL without dead time correction	Required values in PVAL with dead time correction	Actual values loaded into PVAL registers
Phase 1	PVAL1	PWM1	PWM1 + DT/2	PWM1 + DEADTM/2/PWM_PRESC
	PVAL2	-	PWM1 - DT/2	PWM1 - DEADTM/2/PWM_PRESC
Phase 2	PVAL3	PWM2	PWM2 + DT/2	PWM2 + DEADTM/2/PWM_PRESC
	PVAL4	-	PWM2 - DT/2	PWM2 - DEADTM/2/PWM_PRESC
Phase 3	PVAL5	PWM3	PWM3 + DT/2	PWM3 + DEADTM/2/PWM_PRESC
	PVAL6	-	PWM3 - DT/2	PWM3 - DEADTM/2/PWM_PRESC

When calculating the values to be loaded into the PVAL registers, the MRxx's Dead Time register can be used.

The dead-time register (DEADTM) holds an 8-bit value which specifies the number of CPU clock cycles to be used for the dead-time, when complementary PWM mode is selected. Dead-time is not affected by changes to the prescaler value. On the other hand, the PVAL values are affected by the prescaler of the PWM counter.

Therefore the value stored into the dead time register needs to be scalled by the PWM prescaler (PWM_PRESC in **Table 2-1**). The PWM Control Register 2 (PCTL2) contains the PWM generator prescaler. The buffered read/write bits, PRSC0 and PRSC1, select the PWM prescaler according to **Table 2-2**.

Table 2-2. PWM Prescaler

Prescaler bits PRSC0 and PRSC1	PWM Frequency	Prescaler PWM_PRESC
00	f_{OP}	1
01	$f_{OP}/2$	2
10	$f_{OP}/4$	4
11	$f_{OP}/8$	8

The on-chip PWM module of MC68HC908MRxx microcontrollers enables them to perform two types of dead time distortion correction:

- **Partial** correction
- **Full** correction

Partial dead time distortion correction is based only on polarity detection of phase current. The hardware, sensing the current polarity according to **Figure 2-4**, needs to be implemented. The software is responsible for calculating both compensated PWM values and placing them in an odd/even PWM register pair according to **Table 2-1**. The distortion correction is fully implemented by the on-chip PWM module according to the following scheme:

- If the **current** sensed at the motor for that PWM pair **is positive** (voltage on current pin ISx is low), **the odd PWM value is used** for the PWM pair.
- Likewise, if the **current** sensed at the motor for that PWM pair **is negative** (voltage on current pin ISx is high), **the even PWM value is used**.

For partial correction, the on-chip dead time correction block is set in the automated mode - current sense correction bits ISENS1:ISENS0 of PWM Control Register 0 (PCTL1) are set to 10).

The disadvantage of the partial correction is that some dead time distortion still exist - the current is flattened out at the zero crossings.

Full dead time distortion correction (implemented in **dtCorrectFull algorithm**) improves the partial dead time correction by sensing not only the polarity, but also the magnitude of the actual phase current.

In the full dead time correction method, the threshold, where the correction values should be toggled is not in the zero level, but slightly advanced. The threshold is illustrated in **Figure 2-5**. Toggling of the correction offset needs to occur before the current has a chance to flatten out at a current zero-crossing. So, the current sense scheme must sense that the current waveform is approaching the zero-crossing.

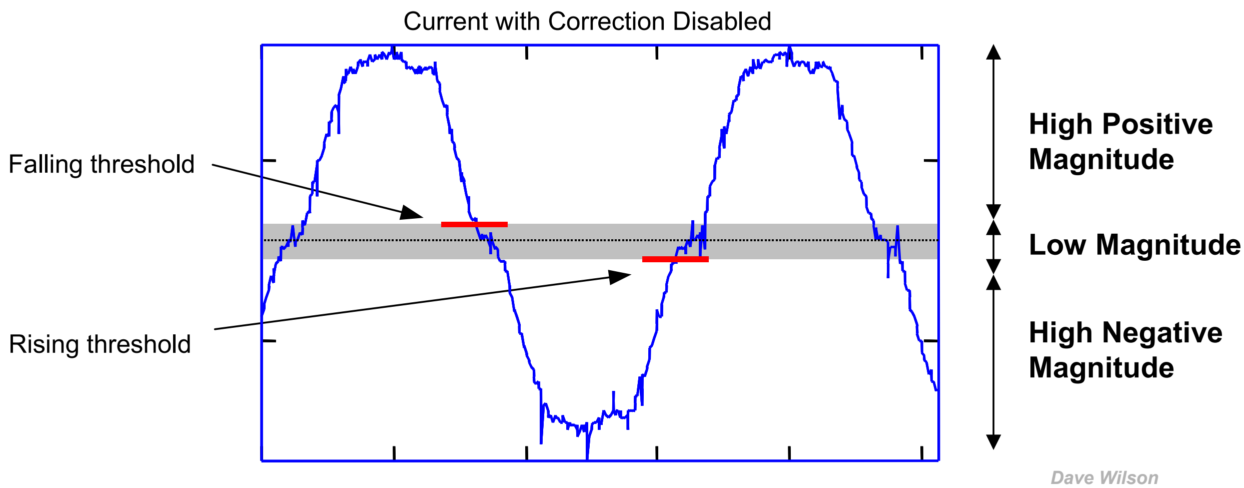

Figure 2-5. Proposed Current Threshold for Correction Toggling

To achieve the full distortion correction, again one of two different correction factors must be added to the desired PWM value, depending on whether the top or bottom transistor is controlling the output voltage during the dead time. The software is responsible for calculating both compensated PWM values and placing them in an odd/even PWM register pair. Then the s/w needs to determine which PWM value is to be used, according to the following scheme:

- If the **current** sensed at the motor for that PWM pair **is positive and of high magnitude,** or **negative and of small magnitude in a trend approaching zero crossing**, **the odd PWM value is used** for the PWM pair.
- Likewise, if the **current** sensed at the motor for that PWM pair **is negative,** or **positive and of small magnitude in a trend approaching zero crossing**, **the even PWM value is used**.

The MR32 contains a hardware circuitry that enables it to sense the current polarity together with the magnitude. The current polarity and magnitude is sensed using the DT-DT6 of FTACK register in ’908MR32 microcontroller. For Phase 1, the bits DT1 and DT2 are used as shown in **Table 2-3**.

Table 2-3. Sensing of the Current Polarity and Magnitude for Ph. 1

DT1	DT2	Current Condition of Phase 1
0	0	high magnitude I+
1	1	high magnitude I-
0	1	low magnitude, either polarity

For phase 2, bits DT3 and DT4 are used. For phase 3, bits DT5 and DT6 are used.

As was stated the determination of the correct PVAL used for the PWM generation is done purely by software. The on-chip dead time correction block is set in the manual mode - current sense correction bits ISENS1:ISENS0 of PWM Control Register 0 (PCTL1) are set to 00 or 01.

Section 3. Hardware Design

3.1 Contents

3.2 System Configuration 27
3.3 MC68HC908MR32 Control Board 29
3.4 3-Phase AC BLDC High Voltage Power Stage. 31
3.5 Optoisolation Board 33
3.6 Motor-Brake Specifications. 35
3.7 Hardware Documentation. 36

3.2 System Configuration

The application is designed to drive the 3-phase AC motor. It consists of the following modules (see **Figure 3-1**):

- MC68HC908MR32 Control Board
- 3-phase AC/BLDC High Voltage Power Stage
- Optoisolation Board
- 3-phase AC Induction Motor

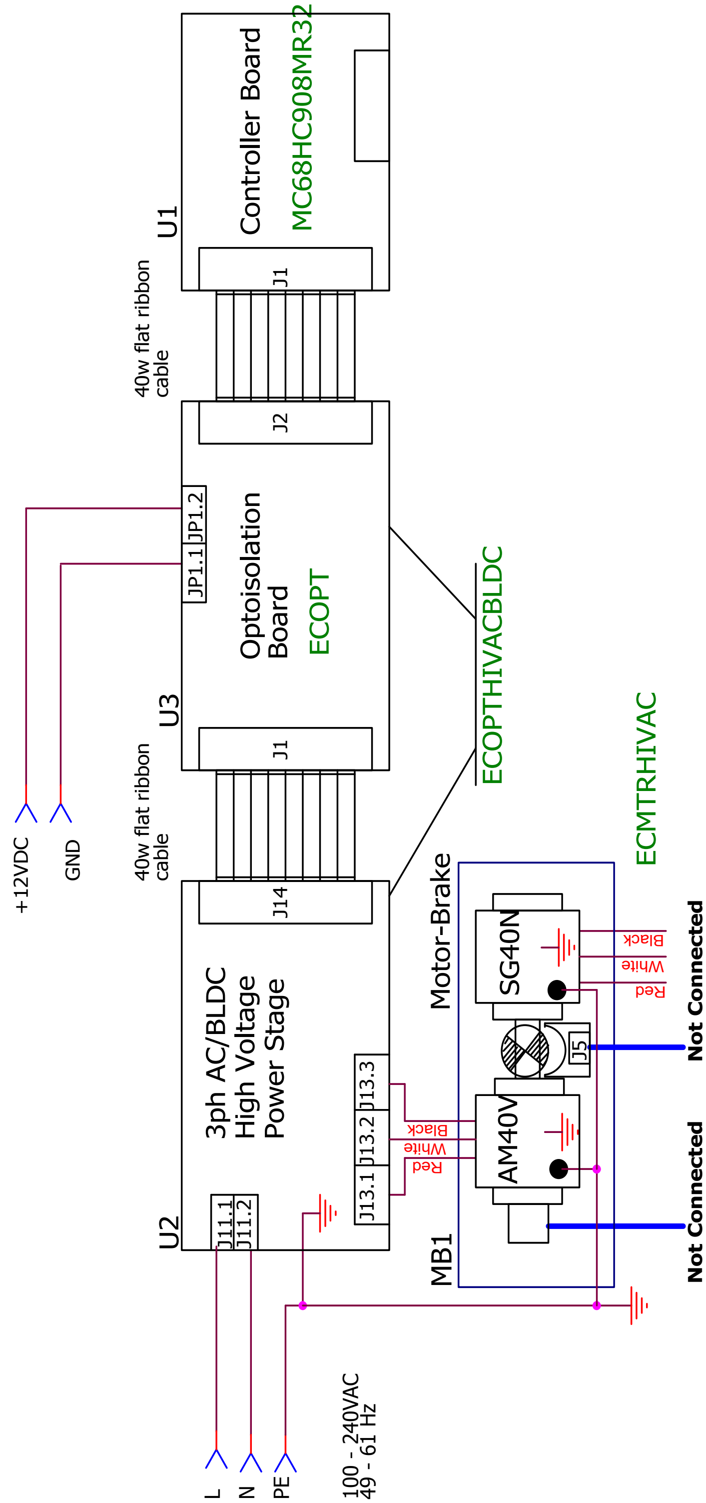

Figure 3-1. Hardware Configuration

3.3 MC68HC908MR32 Control Board

Motorola's embedded motion control series MR32 motor control board is designed to provide control signals for 3-phase AC induction, 3-phase brushless DC (BLDC), and 3-phase switched reluctance (SR) motors. In combination with one of the embedded motion control series power stages, and an optoisolation board, it provides a software development platform that allows algorithms to be written and tested without the need to design and build hardware. With software supplied on the CD-ROM, the control board supports a wide variety of algorithms for AC induction, SR, and BLDC motors. User control inputs are accepted from START/STOP, FWD/REV switches, and a SPEED potentiometer located on the control board. Alternately, motor commands can be entered via a PC and transmitted over a serial cable to DB-9 connector. Output connections and power stage feedback signals are grouped together on 40-pin ribbon cable connector. Motor feedback signals can be connected to Hall sensor/encoder connector. Power is supplied through the 40-pin ribbon cable from the optoisolation board or low-voltage power stage.

The control board is designed to run in two configurations. It can be connected to an M68EM08MR32 emulator via an M68CBL08A impedance matched ribbon cable, or it can operate using the daughter board. The M68EM08MR32 emulator board may be used in either an MMDS05/08 or MMEVS05/08 emulation system.

Figure 3-2 shows a block diagram of the board's circuitry.

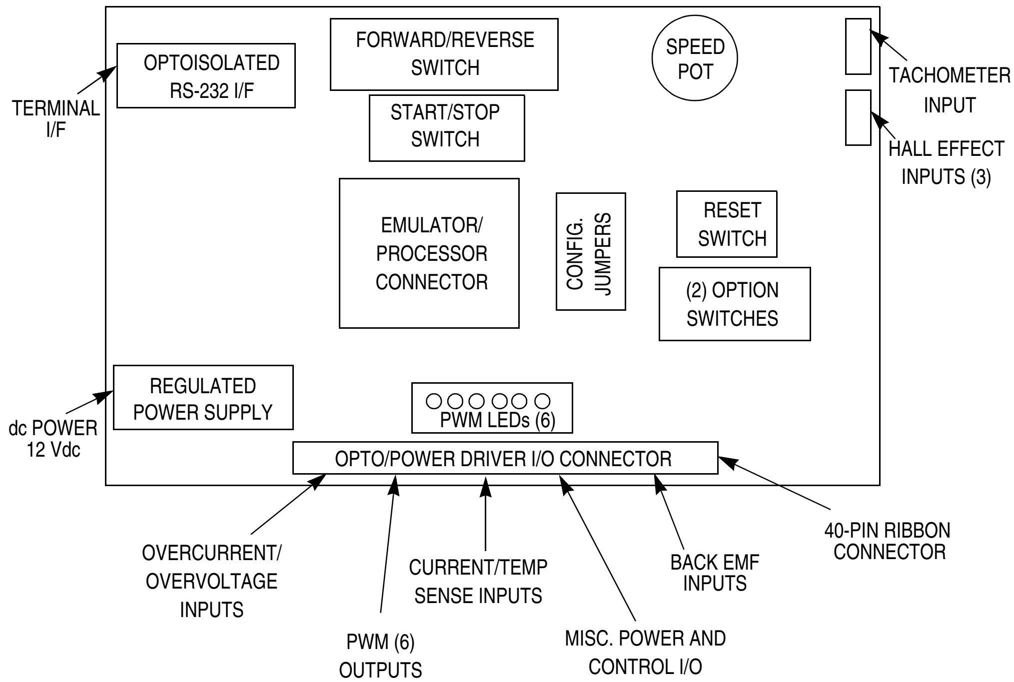

Figure 3-2. MC68HC908MR32 Control Board

The electrical characteristics in Table 3-1 apply to operation at 25°C.

Table 3-1. Electrical Characteristics of Control Board

Characteristics	Symbol	Min	Typ	Max	Units
DC power supply voltage	Vdc	10.8*	12*	16.5*	V
Quiescent current	I_{CC}	—	80	—	mA
Min logic 1 input voltage (MR32)	V_{IH}	2.0	—	—	V
Max logic 0 input voltage (MR32)	V_{IL}	—	—	0.8	V
Propagation delay (Hall sensor/encoder input)	t_{dly}	—	—	500	ns
Analog input range	V_{In}	0	—	5.0	V
RS-232 connection speed		—	—	9600	Baud
PWM sink current	I_{PK}	—	—	20	mA

* When operated and powered separately from other Embedded Motion Control tool set products

3.4 3-Phase AC BLDC High Voltage Power Stage

Motorola's embedded motion control series high-voltage (HV) AC power stage is a 180-watt (one-fourth horsepower), 3-phase power stage that will operate off of DC input voltages from 140 to 230 volts and AC line voltages from 100 to 240 volts. In combination with one of the embedded motion control series control boards and an optoisolation board, it provides a software development platform that allows algorithms to be written and tested without the need to design and build a power stage. It supports a wide variety of algorithms for both AC induction and brushless DC (BLDC) motors.

Input connections are made via 40-pin ribbon cable connector J14. Power connections to the motor are made on output connector J13. Phase A, phase B, and phase C are labeled Ph_A, Ph_B, and Ph_C on the board. Power requirements are met with a single external 140- to 230-volt DC power supply or an AC line voltage. Either input is supplied through connector J11. Current measuring circuitry is set up for 2.93 amps full scale. Both bus and phase leg currents are measured. A cycle-by-cycle over-current trip point is set at 2.69 amps.

The high-voltage AC power stage has both a printed circuit board and a power substrate. The printed circuit board contains IGBT gate drive circuits, analog signal conditioning, low-voltage power supplies, power factor control circuitry, and some of the large, passive, power components. All of the power electronics, which need to dissipate heat, are mounted on the power substrate. This substrate includes the power IGBTs, brake resistors, current sensing resistors, a power factor correction MOSFET, and temperature sensing diodes. **Figure 3-3** shows a block diagram.

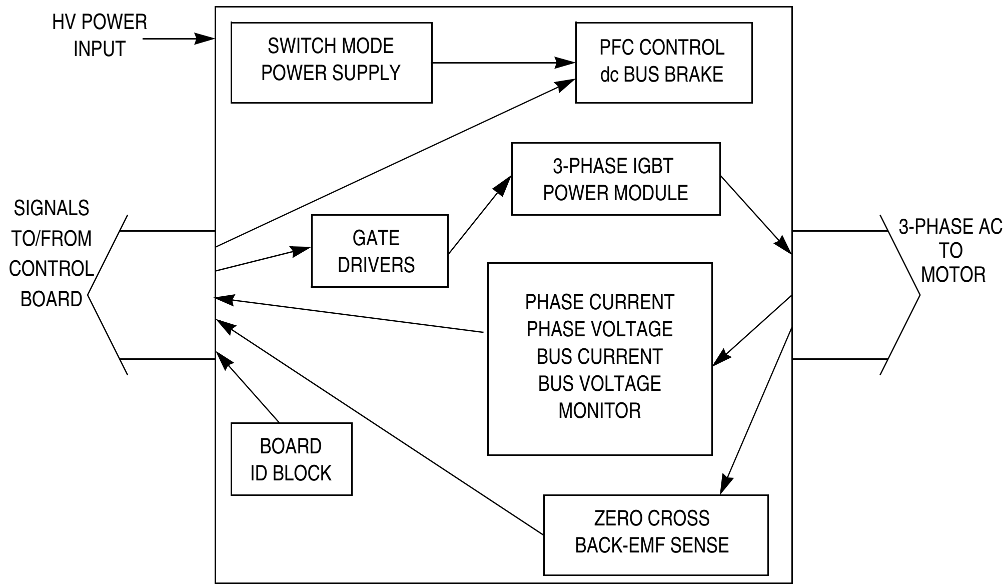

Figure 3-3. 3-Phase AC High Voltage Power Stage

The electrical characteristics in **Table 3-2** apply to operation at 25°C with a 160-Vdc power supply voltage.

Table 3-2. Electrical Characteristics of Power Stage

Characteristics	Symbol	Min	Typ	Max	Units
DC input voltage	Vdc	140	160	230	V
AC input voltage	Vac	100	208	240	V
Quiescent current	I_{CC}	—	70	—	mA
Min logic 1 input voltage	V_{IH}	2.0	—	—	V
Max logic 0 input voltage	V_{IL}	—	—	0.8	V
Input resistance	R_{In}	—	10 kΩ	—	
Analog output range	V_{Out}	0	—	3.3	V
Bus current sense voltage	I_{Sense}	—	563	—	mV/A
Bus voltage sense voltage	V_{Bus}	—	8.09	—	mV/V
Peak output current	I_{PK}	—	—	2.8	A
Brake resistor dissipation (continuous)	P_{BK}	—	—	50	W
Brake resistor dissipation (15 sec pk)	$P_{BK(Pk)}$	—	—	100	W
Total power dissipation	P_{diss}	—	—	85	W

3.5 Optoisolation Board

Motorola's embedded motion control series optoisolation board links signals from a controller to a high-voltage power stage. The board isolates the controller, and peripherals that may be attached to the controller, from dangerous voltages that are present on the power stage. The optoisolation board's galvanic isolation barrier also isolates control signals from high noise in the power stage and provides a noise-robust systems architecture.

Signal translation is virtually one-for-one. Gate drive signals are passed from controller to power stage via high-speed, high dV/dt, digital optocouplers. Analog feedback signals are passed back through HCNR201 high-linearity analog optocouplers. Delay times are typically

250 ns for digital signals, and 2 µs for analog signals. Grounds are separated by the optocouplers' galvanic isolation barrier.

Both input and output connections are made via 40-pin ribbon cable connectors. The pin assignments for both connectors are the same. For example, signal PWM_AT appears on pin 1 of the input connector and also on pin 1 of the output connector. In addition to the usual motor control signals, an MC68HC705JJ7CDW serves as a serial link, which allows controller software to identify the power board.

Power requirements for controller side circuitry are met with a single external 12-Vdc power supply. Power for power stage side circuitry is supplied from the power stage through the 40-pin output connector.

The electrical characteristics in **Table 3-3** apply to operation at 25°C, and a 12-Vdc power supply voltage.

Table 3-3. Electrical Characteristics of Optoisolation Board

Characteristic	Symbol	Min	Typ	Max	Units	Notes
Power Supply Voltage	Vdc	10	12	30	V	
Quiescent Current	I_{CC}	70(1)	200(2)	500(3)	mA	DC/DC converter
Min Logic 1 Input Voltage	V_{IH}	2.0	—	—	V	HCT logic
Max Logic 0 Input Voltage	V_{IL}	—	—	0.8	V	HCT logic
Analog Input Range	V_{In}	0	—	3.3	V	
Input Resistance	R_{In}	—	10	—	kΩ	
Analog Output Range	V_{Out}	0	—	3.3	V	
Digital Delay Time	t_{DDLY}	—	0.25	—	µs	
Analog Delay Time	t_{ADLY}	—	2	—	µs	

1. Power supply powers optoisolation board only.
2. Current consumption of optoisolation board plus DSP EVM board (powered from this power supply)
3. Maximum current handled by DC/DC converters

3.6 Motor-Brake Specifications

The AC induction motor-brake set incorporates a 3-phase AC induction motor and attached BLDC motor brake. The AC induction motor has four poles. The incremental position encoder is coupled to the motor shaft, and position Hall sensors are mounted between motor and brake. They allow sensing of the position if required by the control algorithm. Detailed motor-brake specifications are listed in **Table 3-4**. In a target application a customer specific motor is used.

Table 3-4. Motor - Brake Specifications

Set Manufactured	EM Brno, Czech Republic	
Motor Specification:	eMotor Type:	AM40V 3-Phase AC Induction Motor
	Pole-Number:	4
	Nominal Speed:	1300 rpm
	Nominal Voltage:	3 x 200 V
	Nominal Current:	0.88 A
Brake Specification:	Brake Type:	SG40N 3-Phase BLDC Motor
	Nominal Voltage:	3 x 27 V
	Nominal Current:	2.6 A
	Pole-Number:	6
	Nominal Speed:	1500 rpm
Position Encoder	Type:	Baumer Electric BHK 16.05A 1024-12-5
	Pulses per Revolution:	1024

3.7 Hardware Documentation

All the system parts are supplied and documented according to the following references:

- U1 - MC68HC908MR32 Control Board:
  - supplied as: ECCTR908MR32
  - described in: *Motorola Embedded Motion Control MC68HC908MR32 Control Board User's Manual* MEMCMR32CBUM/D
- U2 - 3-ph AC/BLDC High Voltage Power Stage
  - supplied in kit with Optoisolation Board as: ECOPTHIVACBLDC
  - described in: *Motorola Embedded Motion Control 3-Phase AC BLDC High-Voltage Power Stage User's Manual* MEMC3PBLDCPSUM/D
- U3 - Optoisolation Board
  - supplied with 3-ph AC/BLDC High Voltage Power Stage as: ECOPTHIVACBLDC
  - or supplied alone as: ECOPT - optoisolation board
  - described in: *Motorola Embedded Motion Optoisolation Board User's Manual* MEMCOBUM/D
- MB1 Motor-Brake AM40V + SG40N
  - supplied as: ECMTRHIVAC

Detailed descriptions of individual boards can be found in comprehensive User's Manuals belonging to each board. The manuals are available on the Motorola web. The User's Manual incorporates the schematic of the board, description of individual function blocks and a bill of materials. An individual board can be ordered from Motorola as a standard product.

Section 4. Software Design

4.1 Contents

4.2 Introduction 37

4.3 Data Flow 37

4.4 Algorithm of Dead Time Distortion Correction 43

4.2 Introduction

This section describes the design of the software blocks of the drive. The software will be described in terms of -

- Software Data Flow
- Algorithm Dead Time Distortion Correction

4.3 Data Flow

The requirements of the drive dictate that software takes some values from the user interface and sensors, processes them and generates 3-phase PWM signals for motor control.

The control algorithm of closed loop AC drive is described in **Figure 4-1**. It consists of processes described in the following sub-sections. The dead time distortion correction algorithm is described separately in the successive section.

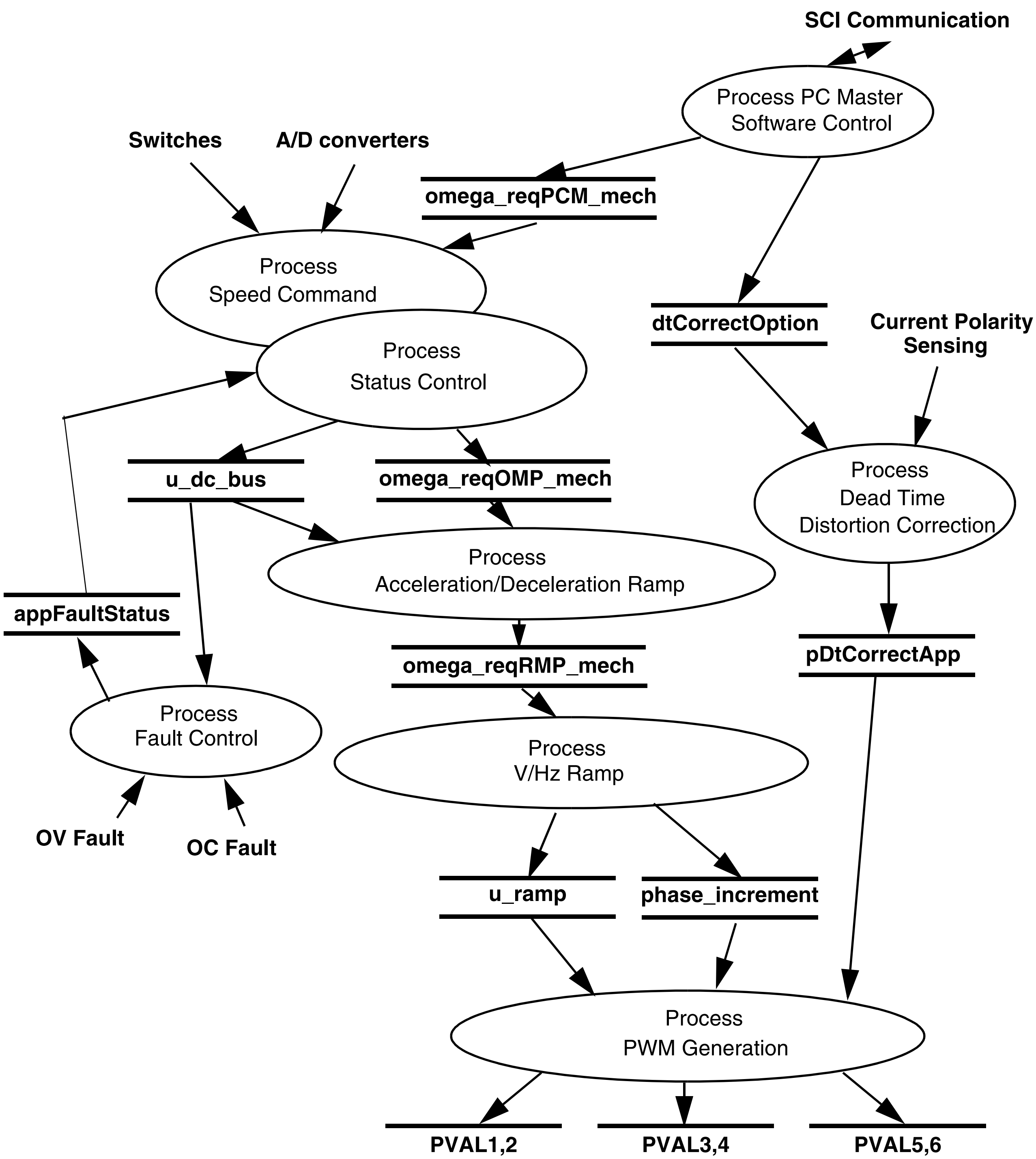

Figure 4-1. Data Flow

4.3.1 Speed Command & Status Control

In the Manual Operating Mode, the required speed is set by speed potentiometer and switches (start/stop, forward/reverse). In the PC Master Software (Remote) Operating Mode, the required speed is set by PC. In the process, the input parameters are evaluated and the speed command is calculated accordingly. Also the DC-Bus voltage is measured. The application fault status is analyzed and the state of the drive is set. The status LED's are controlled according to the system state.

4.3.2 Acceleration/Deceleration Ramp

The process calculates the new speed command based on the required speed according to the acceleration / deceleration ramp.

4.3.3 V/Hz Ramp

This process provides voltage calculation according to V/Hz ramp. The input of this process is the generated inverter frequency *omega_req_RMP_mech*. The outputs of this process are the output sine wave parameters required by PWM generation process: the table increment *phase_increment* that corresponds to the frequency *omega_req_RMP_mech* and is used to roll through the wave table in order to generate the output inverter frequency, and the corresponding amplitude of the generated inverter voltage *u_ramp*.

4.3.4 Process PWM Generation

This process generates a system of three phase sinewaves shifted 120^{o} each other. The function *mcgen3PhWaveSine* is used for the sine wave calculation.

The *mcgen3PhWaveSine* function calculates an immediate value of the three-phase sinusoidal system from given amplitude and actual phase pointer:

- Phase A — *sPhaseVoltage.PhaseA*

- Phase B — *sPhaseVoltage.PhaseB*
- Phase C — *sPhaseVoltage.PhaseC*

The individual waves are shifted 120° each other. The shape of the generated waveforms depends on the data stored in the sine table. In motor control applications, data usually describes a pure sinewave or a sinewave with addition of the third harmonic component.

Figure 4-2 shows the duty cycles generated by the *mcgen3PhWaveSine* function when amplitude is 50%.

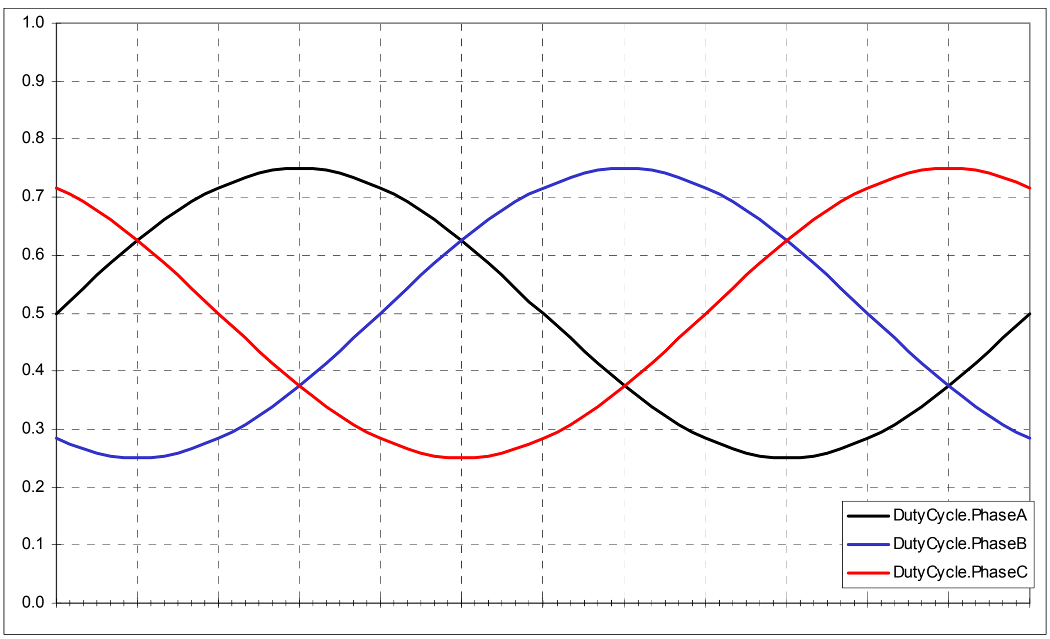

Figure 4-2. 3-Phase Sine Waves with Amplitude of 50%

The calculation is based on the wave table stored in FLASH memory of the microcontroller. The table describes either a pure sinewave or a sinewave with the third harmonic addition. The second case is often preferred because it allows one to generate the first harmonic sine voltage equal to the input AC line voltage. The format of the stored wave table data is from #0x0000 (for ZERO Voltage) up to 0x7fff (for the 100% Voltage). Thus the proper data scaling is secured (see **Figure 4-3**).

It is important to note that 50% PWM (or 50% of PWM Modulus loaded to the corresponding PVAL registers) corresponds to the ZERO phase voltage. But in the wave table, the ZERO phase voltage corresponds to the number #0x0000. Therefore the fetched wave value from the table must be added to the 50% PWM Modulation for quadrant 1 and 2 or substracted from the 50% PWM Modulation for quadrant 3 and 4. Thus the correct PWM value is loaded.

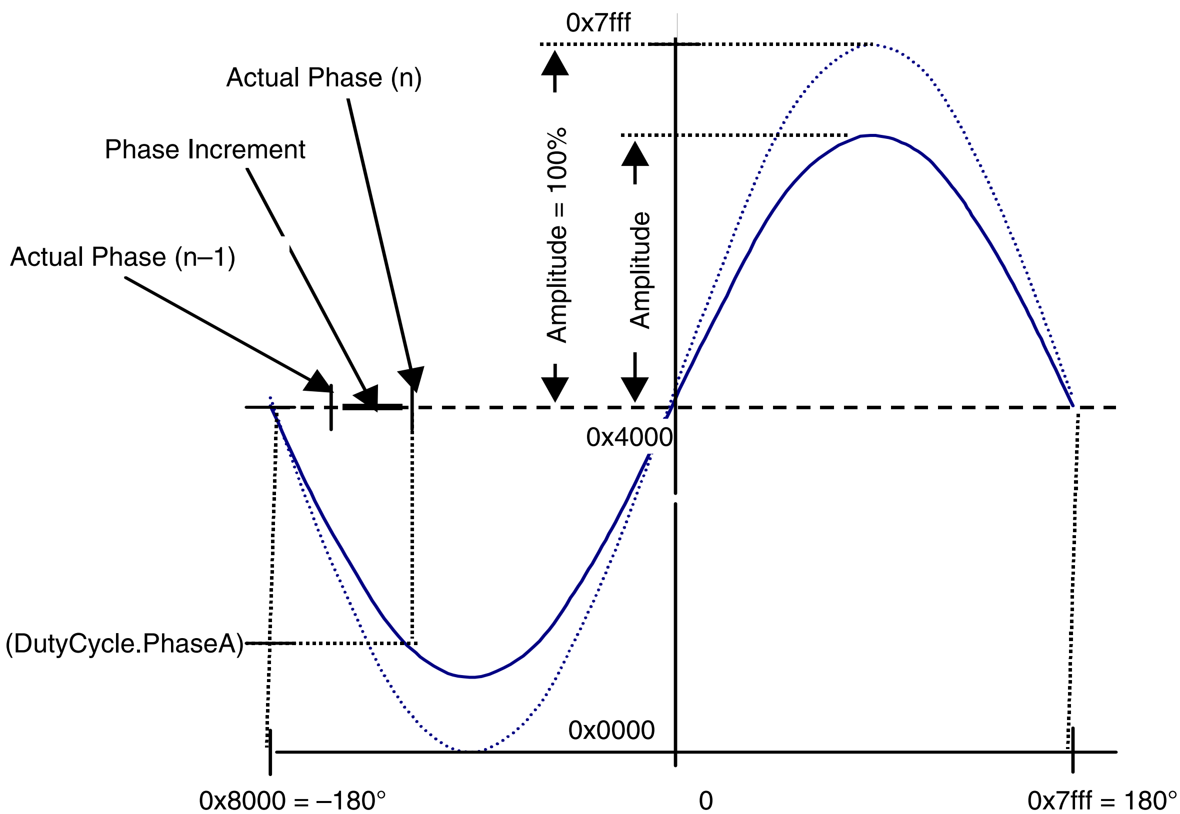

Figure 4-3. mcgen3PhWaveSine Data Explanation - Phase A

The output parameters of the process are:

- PWM value for phase A: PVAL1 register
- PWM value for phase B: PVAL3 register
- PWM value for phase C: PVAL5 register

In case of dead time distortion correction, the corrected PVAL values PVAL1-6 are calculated and used for the PWM generation according to the detected phase current polarity.

The process can be described by following points:

- Wave pointer for phase A is updated by the table increment.
- Based on the wave pointer, the PWM values for all three phases are calculated.
- PWM values are rescaled according to the PWM modulo (PWM frequency) and loaded into PVAL1, 3, 5 registers. Registers PVAL2, 4, 6 are loaded automatically because of complementary PWM mode selected during the PWM module initialisation.
- In case of dead time distortion correction, the corrected values PVAL1-6 are calculated and used for PWM generation according to the detected phase current polarity.

The process is accessed regularly in the rate given by the set PWM frequency and the selected PWM interrupt prescaler.

4.3.5 PC Master Software Control

The process provides SCI communication with PC using PC master software service routines. These routines are fully independent on the motor control tasks. They enable for example to set the desired speed, the PWM frequency and the type of dead time distortion correction.

4.3.6 Fault Control

This process is responsible for fault handling. The software accommodates three fault events: DC-Bus over-current, DC-Bus over-voltage and DC-Bus under-voltage.

DC-Bus Over-current: In case of DC-Bus over-current, the external hardware provides a rising edge on the DC-Bus over-current fault input of the microcontroller. This signal disables all motor control PWM outputs (PWM1 - PWM6) and sets the application fault status.

DC-Bus Over-voltage: In case of DC-Bus over-voltage, the external hardware provides a rising edge on the DC-Bus over-voltage fault input of the microcontroller. This signal disables all motor montrol PWM outputs (PWM1 - PWM6) and sets the application fault status.

DC-Bus Under-voltage: The sensed DC-Bus voltage is compared with the limit within the software. In case of DC-Bus under-voltage, all motor control PWM outputs (PWM1 - PWM6) are disabled and the application fault status is set.

If any of the faults occurs, the application status is changed into the Fault Status.

4.3.7 Dead Time Distortion Correction

The process defines the value registers to be used for PWM generation according to the type of dead time distortion correction and the state of the immediate phase current polarity.

- If no dead time correction is required, the PVAL1,3,5 are used, the complementary PVAL values are calculated by on-chip PWM peripheral automatically.
- If partial dead time correction is required, the PVAL value is selected by on-chip PWM peripheral automatically according to the phase current polarity sensing
- If full dead time correction is required, the process selects the desired PVAL registers according to the dead time distortion correction state machine.

In the following section the dead time distortion correction algorithm is described in detail.

4.4 Algorithm of Dead Time Distortion Correction

The algorithm **dtCorrectFull** calculates the IPOL bits defining the PVAL registers to be used for MC68HC908MR32 PWM generation for full dead time correction. The IPOL bits are determined according to the phase current polarity detection bits DT1-6, actual sine wave pointer, and the actual state of the algorithm state machine.

The algorithm state machine samples the actual state of the phase current, and selects appropriate PVAL registers to be used for PWM generation. The state machine, implemented in the **dtCorrectFull** algorithm, is illustrated in **Figure 4-4**.

When the algorithm is enabled, the state machine is entered from initial state 0. It is waiting till the high magnitude of positive current is detected (State 1, confirmed by State 2), then the algorithm enters the state machine (State 3). The state machine is performed in circle 3-4-5-6-3. As soon as the low magnitude of negative current is detected, the IPOL is changed to 1, requesting the even-numbered PWM registers to be used for PWM generation, the actual value of the wave pointer is recorded (θ_C), and State 4 is entered. State 4 is preserved for 80 electrical degrees, until a high negative current can be expected. Then State 5 is entered. As soon as the low magnitude of positive current is detected, the IPOL is changed to 0, requesting the odd-numbered PWM registers to be used for PWM generation, the actual value of the wave pointer is recorded (θ_C), and State 6 is entered. State 6 is preserved for 80 electrical degrees, until a high positive current can be expected. Then State 3 is entered and the state machine loop is repeated. In this way, it is ensured that the required IPOL changes when a small amplitude of respective current is detected by the hardware. Please note, that the wave pointer is recorded into the algorithm variable *PointA*, *PointB,* or *PointC,* in the moment when the respective phase current crosses the low current threshold.

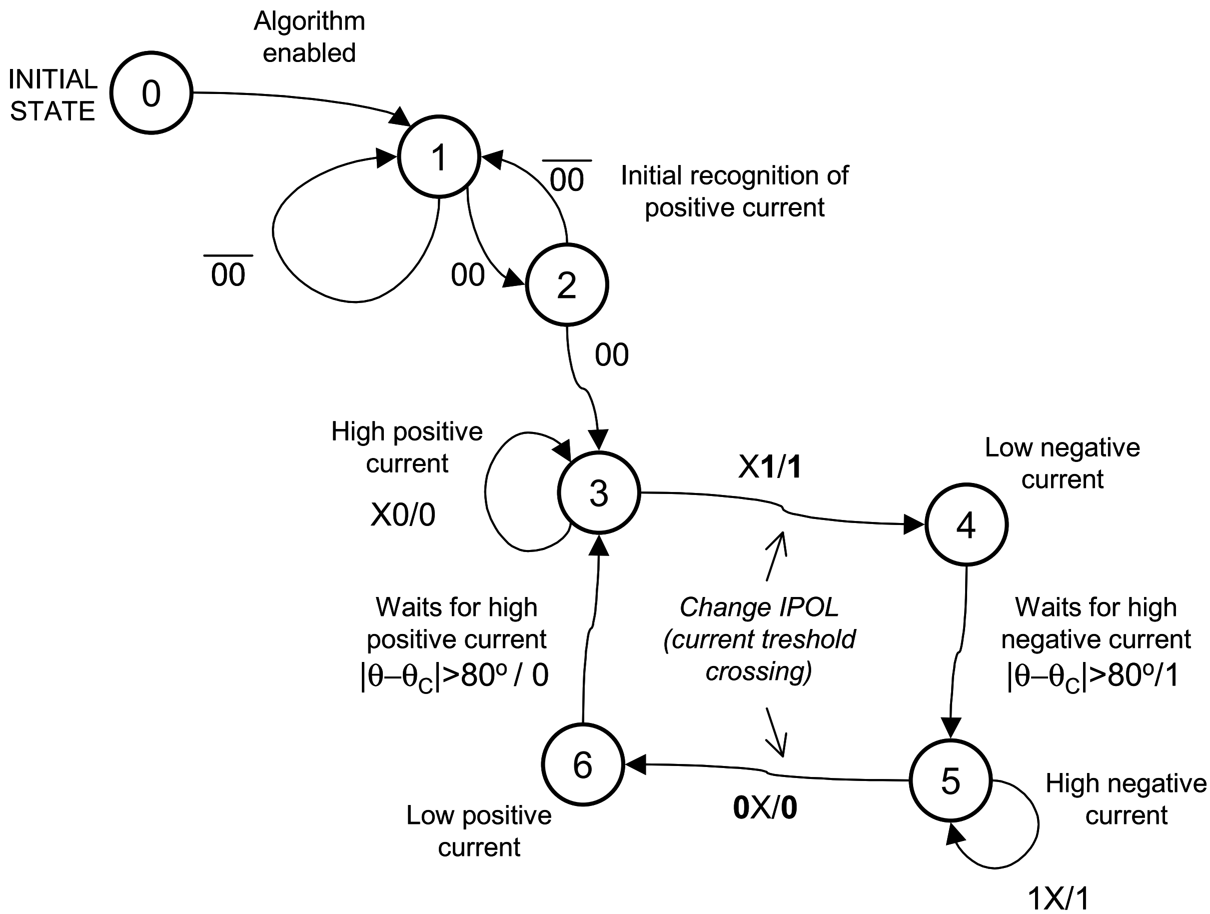

Figure 4-4. Dead Time Correction State Machine

Such a state machine is independently implemented for each phase (A, B, C). The algorithm contains 2 flag variables, determining actual state of the state machine for individual phases. Flag variable *dtStateFlagsAB* determines state of the state machine for phases A & B, *dtStateFlagsC* determines state of the state machine for phase C.

The meaning of individual bits of *dtStateFlagsAB* is listed in **Table 4-1**. The meaning of individual bits of *dtStateFlagsC* is listed in **Table 4-2**.

Table 4-1. State Machine Flag Registers *dtStateFlagsAB*

phase	bits	State					
		1	2	3	4	5	6
phase A	bit0 - lock	0	0	1	1	1	1
	bit1	0	1				
	bit2			0	1	0	1
	bit3			0	0	1	1
phase B	bit4 - lock	0	0	1	1	1	1
	bit5	0	1				
	bit6			0	1	0	1
	bit7			0	0	1	1

Table 4-2. State Machine Flag Registers *dtStateFlagsC*

phase	bits	State					
		1	2	3	4	5	6
phase C	bit0 - lock	0	0	1	1	1	1
	bit1	0	1				
	bit2			0	1	0	1
	bit3			0	0	1	1
reserved	bit4	x	x	x	x	x	x
	bit5	x	x	x	x	x	x
	bit6	x	x	x	x	x	x
	bit7	x	x	x	x	x	x

NOTE: *Detailed explanation of the dead time distortion correction can be found in a comprehensive application note of Motorola, AN1728 “Making Low-Distortion Motor Waveforms with the MC68HC708MP16“ by David Wilson. Note, that MC68HC708MP16 is the predecessor of MC68HC908MRxx family and contains identical on-chip PWM block.*

Algorithm Data Structure:

Algorithm data structure is defined in *dtCorrect.h* header file.

See **Table 4-3**.

```
typedef struct {
    UByte dtBits;
    UByte ipolBits;
    type_uBits dtStateFlagsAB;
    type_uBits dtStateFlagsC;
```

```
    SByte pointA;
    SByte pointB;
    SByte pointC;
    SByte pointerA;
} dtCorrect_s;
```

Table 4-3. dtCorrect_s Structure Elements

Variable	Explanation
dtBits	INPUT: actual status of the dead time bits DT1-6, format | x | x |DT6|DT5| |DT4|DT3|DT2|DT1| fits to FTACK of 'MR32
ipolBits	OUTPUT: ipolBits - new top/bottom correction bits IPOL1-3, format | x | x | x |IPOL1| |IPOL2|IPOL3| x | x | fits to PCTL2 of 'MR32
dtStateFlagsAB	internal dead-time correction flags for phases AB
dtStateFlagsC	internal dead-time correction flags for phase C
pointA	internal capture of the pointer for phase A
pointB	internal capture of the pointer for phase B
pointC	internal capture of the pointer for phase C
pointerA	INPUT: actual pointer of the generated wave phase A

The dead time correction algorithm **dtCorrectFull** adds the correction factor to originally calculated sine wave. It is necessary to ensure that the calculated PWM duty cycles do not exceed the PWM modulus.

The **dtCorrectInit** function must be called before starting any call to the **dtCorrectFull** function**,** to ensure proper functionality.

Section 5. System Setup

5.1 Contents

5.2 Hardware Setup 49

5.3 Warning 50

5.4 Jumper Settings of Controller Board. .51

5.5 Required Software Tools 52

5.6 Building the Application 53

5.7 Executing the Application 53

5.8 Controlling the Application with PC Master Software 57

5.2 Hardware Setup

Figure 5-1 illustrates the hardware setup of the application. It incorporates the following modules:

- MC68HC908MR32 Control Board
- 3-phase AC/BLDC High Voltage Power Stage
- Optoisolation Board
- 3-phase AC Induction Motor

The correct phase order (phase A, phase B, phase C) for the shown AC induction motor is:

- Phase A — red wire
- Phase B — white wire
- Phase C — black wire

If you view the motor looking into the shaft end, and the phase order is phase A, B, C, the motor shaft should rotate in a clockwise direction (i.e., positive direction, positive speed).

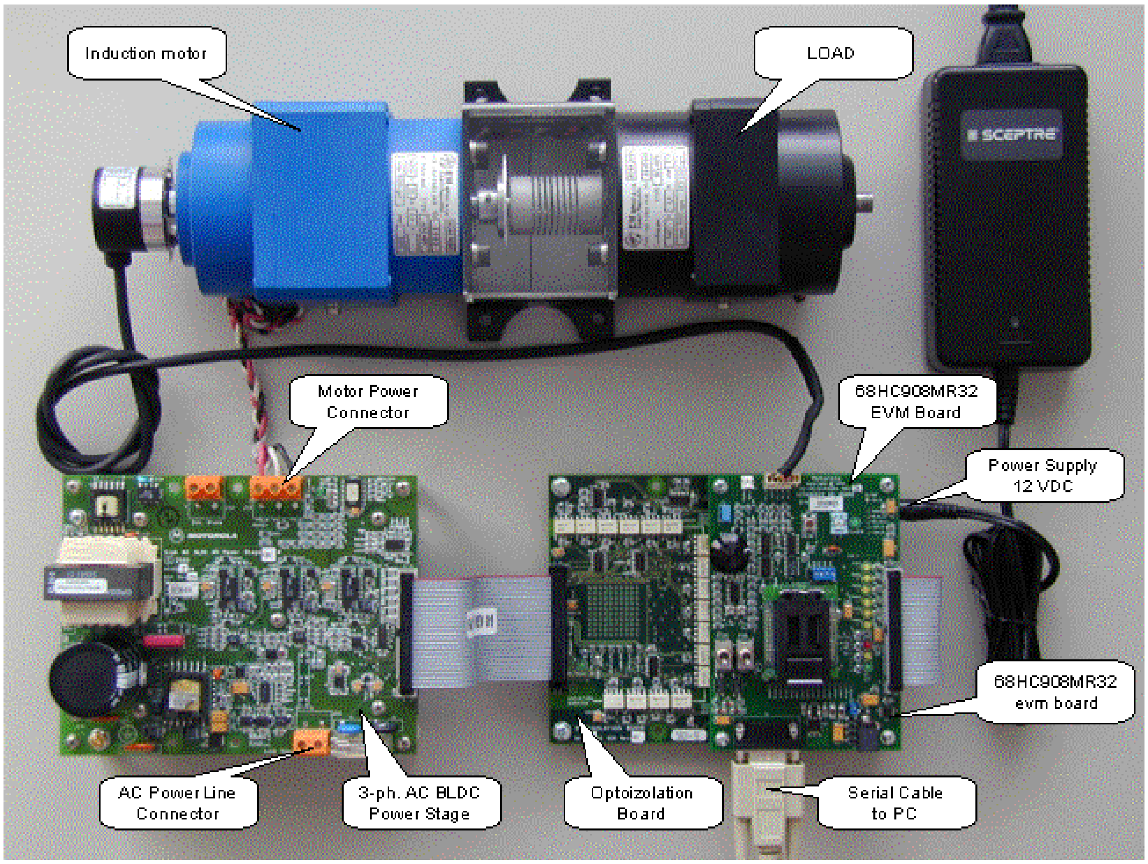

Figure 5-1. Setup of the Application

5.3 Warning

This application operates in an environment that includes dangerous voltages and rotating machinery.

Be aware, that the application power stage and optoisolation board are not electrically isolated from the mains voltage - they are live with risk of electric shock when touched.

An isolation transformer should be used when operating off an AC power line. If an isolation transformer is not used, power stage grounds and oscilloscope grounds are at different potentials, unless the oscilloscope is floating. Note, that probe grounds and, therefore, the case of a floated oscilloscope are subjected to dangerous voltages.

The user should be aware, that:

- Before moving scope probes, making connections, etc., it is generally advisable to power down the high-voltage supply.
- To avoid inadvertent touching live parts, use plastic covers.
- When high voltage is applied, using only one hand for operating the test setup minimizes the possibility of electrical shock.
- Operation in lab setups that have grounded tables and/or chairs should be avoided.
- Wearing safety glasses, avoiding ties and jewelry, using shields, and operation by a personnel trained in high-voltage lab techniques is also advisable.
- Power transistors, the PFC coil, and the motor can reach temperatures hot enough to cause burns.
- When powering down; due to storage in the bus capacitors, dangerous voltages are present until the power-on LED is off.

5.4 Jumper Settings of Controller Board

The MC68HC908MR32 control board jumper settings shown in **Figure 5-2** and **Table 5-1** are required to execute the 3-phase AC motor control application with dead time distortion correction. For a detailed description of the jumper settings, refer to the *MC68HC908MR32 Control Board User's Manual* (Motorola document order number MEMCMR32CBUM/D).

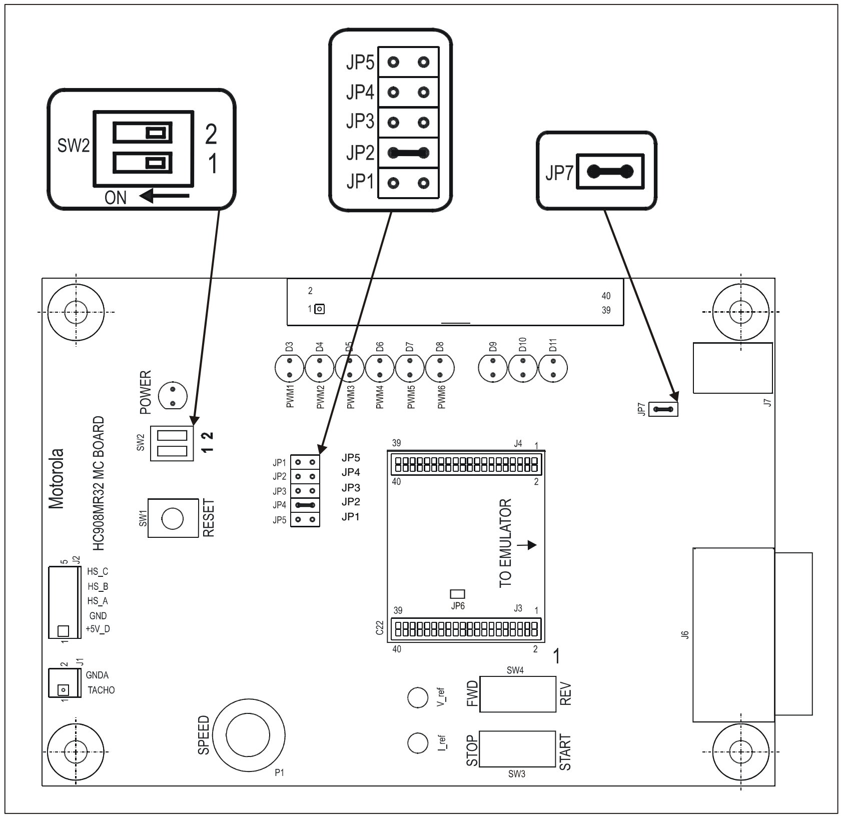

Figure 5-2. MC68HC908MR32 Jumper Reference

Table 5-1. MC68HC908MR32EVM Jumper Settings

Jumper Group	Comment	Connections
JP1	Tachometer input selected	No connection
JP2	Encoder input selected	1–2
JP3	Back EMF signals selected	No connection
JP4	Power factor correction — zero cross signal selected	No connection
JP5	Power factor correction — PWM signal selected	No connection
JP7	Power Supply connected to jack J3	1–2

5.5 Required Software Tools

The application requires the following software development tools:

- Metrowerks[1]CodeWarrior®[2] for MC68HC08 microcontrollers version 1.2 or later.
- PC master software version 1.2.0.11 or later

5.6 Building the Application

To build this application, open the ***3ph_acim_dt_correct.mcp*** project file and execute the *Make* command; see **Figure 5-3**. This command will build and link the motor control application along with all needed Metrowerks libraries.

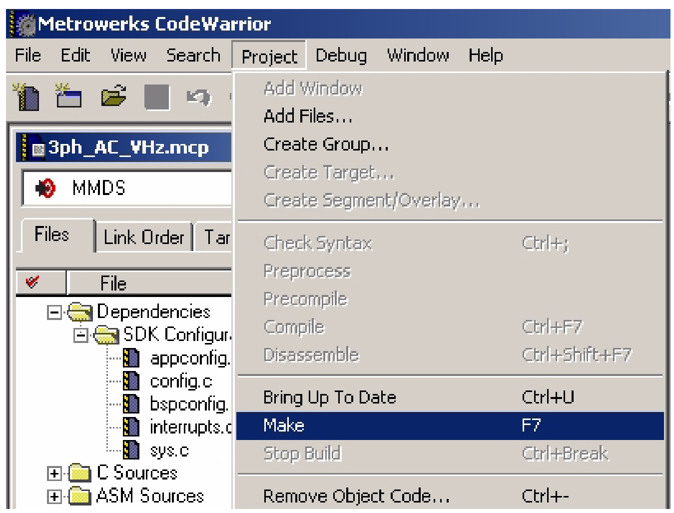

Figure 5-3. Execute *Make* Command

5.7 Executing the Application

To execute the motor control application, in the pull-down menu choose the *Project/Debug* command in the CodeWarrior® IDE, followed by the *Run* command.

1. Metrowerks® and the Metrowerks logo are registered trademarks of Metrowerks, Inc., a wholly owned subsidiary of Motorola, Inc.

2. CodeWarrior® is a registered trademark of Metrowerks, Inc., a wholly owned subsidiary of Motorola, Inc.

If the MMDS target is selected, CodeWarrior will automatically download to the MMDS05/08 emulator.

The application can operate in two modes:

1. Manual Operating Mode

   The drive is controlled by the START/STOP switch (SW3). The direction of the motor rotation is set by the FWD/REV switch (SW4). The motor speed is set by the SPEED potentiometer (P1). Refer to **Figure 5-4** for this description.

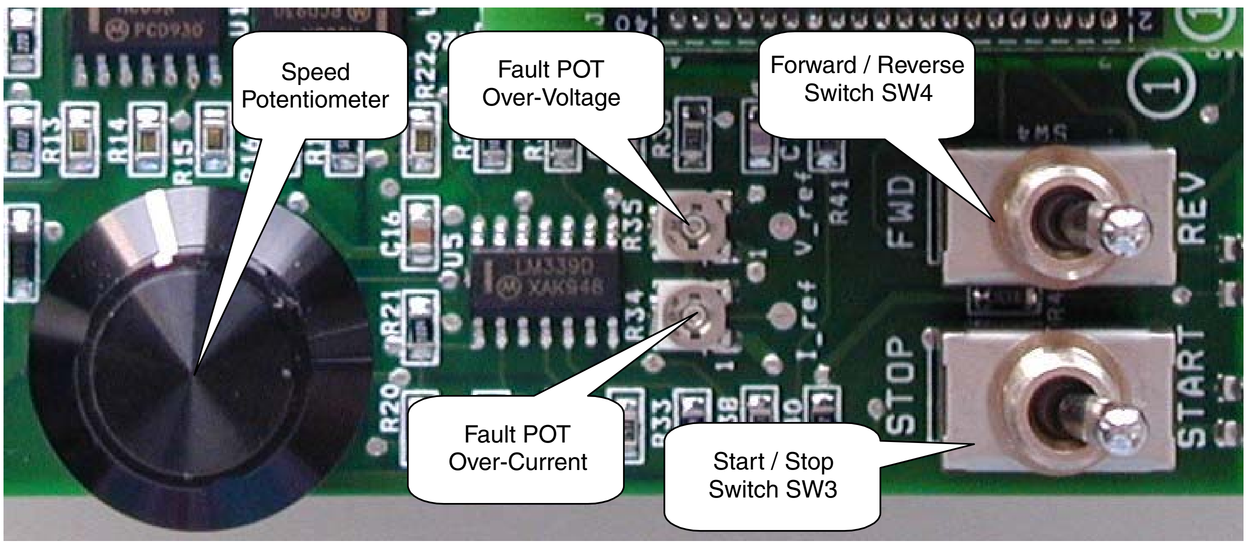

Figure 5-4. Control Elements

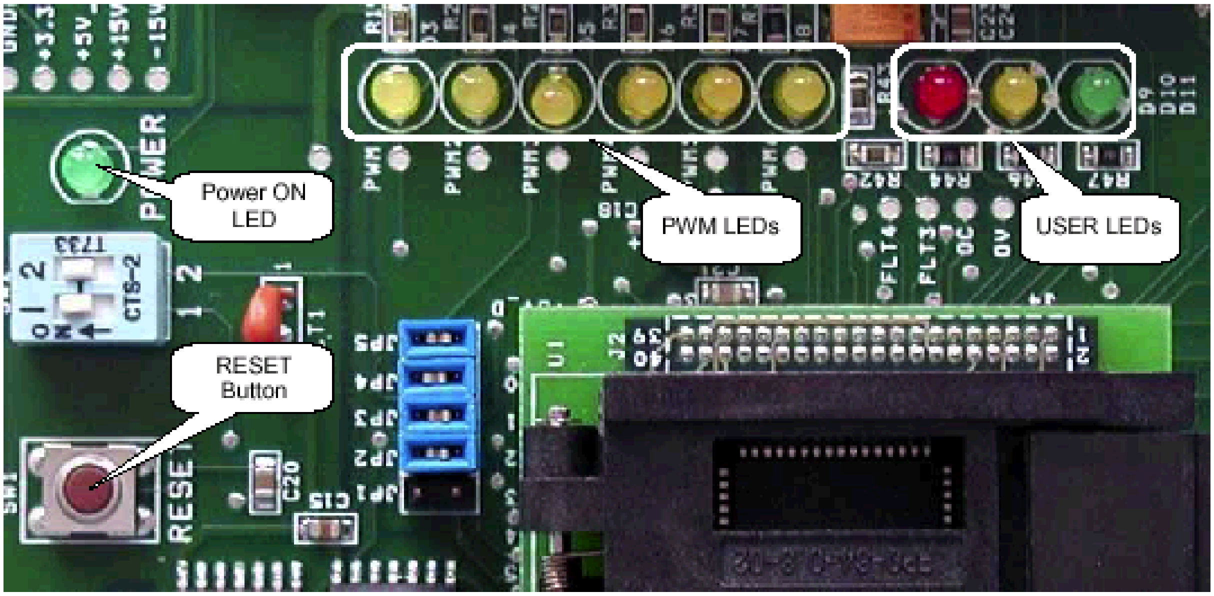

Figure 5-5. USER LEDs, PWM LEDs, and RESET

2. PC Master Software (Remote) Operating Mode

   The drive is controlled remotely from a PC through the serial communications interface (SCI) communication channel of the MCU device via an RS-232 physical interface. The drive is enabled by the START/STOP switch, which can be used to safely stop the application at any time.

   Setting the required speed of the motor is the supported control action.

The application states are displayed with on-board LEDs. Refer to **Figure 5-5** for the LED positions. If the application runs and motor spinning is disabled (i.e., the system is ready), the green status LED will blink. When motor rotation is enabled, the green status LED will be on, and the actual state of the pulse-width modulator (PWM) outputs are indicated with PWM output LEDs, labeled PWM1 - PWM6. If DC-Bus over-current / DC-Bus over-voltage occurs, or if the wrong system board is identified, the green status LED will start to flash quickly and the PC master software will signal the identified fault. This state can be exited only with the application reset.

Refer to Table 5-2 for a description of the application states and their corresponding LED indications.

Table 5-2. Motor Application States

Application State	Motor State	Green LED State
Stopped	Stopped	Blinking at a frequency of 2Hz
Running	Spinning	On
Fault	Stopped	Blinking at a frequency of 8Hz

Once the application is running:

- Move the START/STOP switch (SW3) from STOP to START
- Select the direction of rotation by the FWD/REV switch (SW4)
- Set the required speed by the SPEED potentiometer

If successful, the 3-phase AC induction motor will be spinning.

NOTE: *If the START/STOP switch is set to the START position when the application starts, toggle the switch between the STOP and START positions to enable motor spinning. This is a protection feature preventing the motor to start spinning when the application is executed from CodeWarrior.*

You should also see a lighted green LED indicating the application is running. If the application is stopped, the green LED will blink at a 2-Hz frequency.

When the application is started, the type of dead time distortion correction and desired PWM frequency can be selected using the PC master software control page. The phase voltage and motor current can be observed using the oscilloscope, and the efficiency of dead time distortion correction can be evaluated.

The type of dead time distortion correction is indicated by a yellow LED on MR32 controller board. When the dead time distortion correction is disabled, the yellow LED is turned off. When partial correction is selected, the LED flashes with 2Hz frequency. With full correction, the LED is turned on (refer to Table 5-3).

Table 5-3. Dead Time Distortion Correction

Distortion Correction	Yellow LED State
Disabled	Off
Partial (h/w)	Blinking at a frequency of 2Hz
Full (s/w)	On

5.8 Controlling the Application with PC Master Software

Project file for the PC master software is located in:

..\pcmaster\3ph_acim_dt_correct.pmp

Start the PC master software application window and choose the appropriate PC master software project. **Figure 5-6** shows the PC master software control window for ***3ph_acim_dt_correct.pmp***. The type of dead time distortion correction (no/partial/full), and the PWM frequency (4kHz/8kHz/16kHz/32kHz) can be selected in the variables pane, as shown in **Figure 5-6**.

NOTE: *The desired dead time can be set in application configuration file appconfig.h, where all on-chip modules of the 68HC908MR32 microcontroller are initialized.*

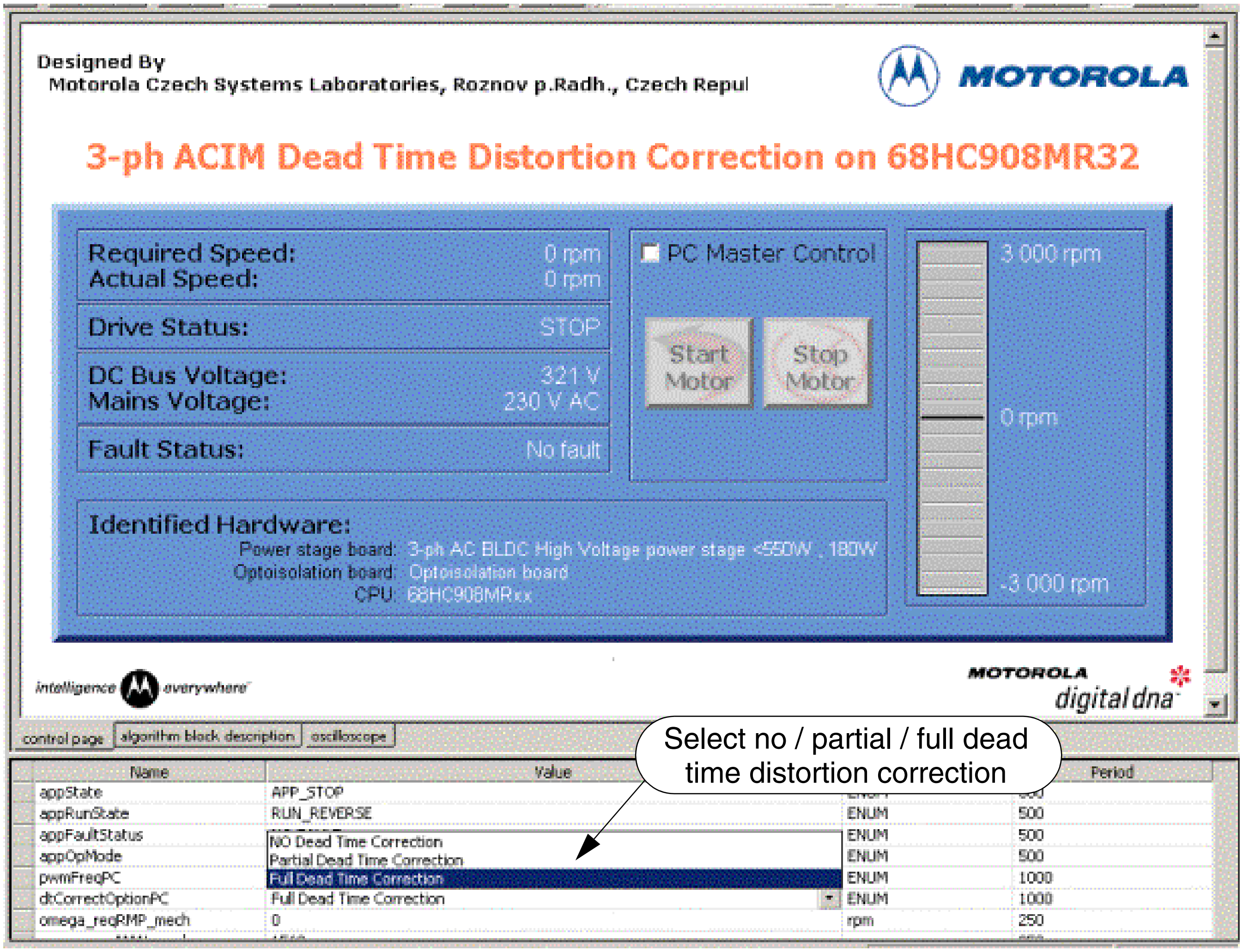

Figure 5-6. PC Master Software Control Window

The PC master software displays the following information:

- required and actual speed of the motor
- phase voltage amplitude (related to given DC-Bus voltage)
- application mode — START/STOP
- DC-Bus voltage
- fault status

The PC master software allows the user to

- set the PWM frequency (the frequency can be changed at any time during the motor operation):
  - 4 kHz
  - 8 kHz
  - 16 kHz
  - 32 kHz
- select dead time distortion correction (the selection can be done at any time during the motor operation):
  - no
  - partial
  - full

The PWM frequency and type of dead time distortion correction can be selected in both the manual and the PC master modes, using the PC master software. It is possible to use the oscilloscope to display the phase currents and voltages for dead time distortion evaluation.

Appendix A. References

1. Motorola, Inc. (2001). *68HC908MR32 User's Manual*, MC68HC908MR32/D
2. Motorola, Inc. (2000). *Motorola Embedded Motion Control MC68HC908MR32 Control Board User's Manual*, MEMCMR32CBUM/D
3. Motorola, Inc. (2000). *Motorola Embedded Motion Control 3-Phase AC BLDC High-Voltage Power Stage User's Manual*, MEMC3PBLDCPSUM/D
4. Motorola, Inc. (2000). *Motorola Embedded Motion Optoisolation Board User's Manual*, MEMCOBUM/D
5. Motorola, Inc. (1997). *Making Low-Distortion Motor Waveforms with the MC68HC708MP16,* AN1728

Appendix B. Glossary

AC — Alternating Current

ACIM — AC Induction Motor

ADC — Analogue-to-Digital Converter

BLDC — brushless DC motor

DC — Direct Current

DT — see “Dead Time (DT)”

DTC — Dead Time Correction, see “Dead Time (DT)”

Dead Time (DT) — short time that must be inserted between the turning off of one transistor in the inverter half bridge and turning on of the complementary transistor due to the limited switching speed of the transistors.

duty cycle — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

MC68HC08 — A Motorola family of 8-bit MCUs

MCU — Microcontroller Unit. A complete computer system, including a CPU, memory, a clock oscillator, and input/output (I/O) on a single integrated circuit.

MR32 (908MR32, MC68HC908MR32) — See "MC68HC08"

phase-locked loop (PLL) — A clock generator circuit in which a voltage controlled oscillator produces an oscillation which is synchronized to a reference signal.

PVAL — PWM value register of motor control PWM module of MC68HC908MR32 microcontroller. It defines the duty cycle of generated PWM signal.

PWM — Pulse Width Modulation

reset — To force a device to the known condition.

SCI — See "serial communications interface module (SCI)"

serial communications interface module (SCI) — A module that supports asynchronous communication.

serial peripheral interface module (SPI) — A module that supports synchronous communication.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See "serial peripheral interface module (SPI)"

SR — switched reluctance motor

timer — A module used to relate events in a system to a point in time.

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://motorola.com/semiconductors

MOTOROLA

DRM019/D