MOTOROLA
intelligence everywhere™

digital dna™

8-Bit Software Development Kit for Motor Control Targeting the MC68HC908MR32

User's Guide

SDKMR32UG/D
Rev. 1, 11/2002

M68HC08 Microcontrollers

MOTOROLA.COM/SEMICONDUCTORS

8-Bit Software Development Kit for Motor Control Targeting the MC68HC908MR32

User's Guide

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://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)
July, 2002	N/A	Original release	N/A
November, 2002	1.0	Added **Section 4. 3-Phase Sine Voltage Powered PM Motor Control Application with Hall Sensors** and **Section 5. 3-Phase ACIM Control with Dead-Time Distortion Correction**	37 and 47

List of Sections

Table of Contents

Section 1. MC68HC908MR32 Applications

Section 2. 3-Phase AC Induction Motor Control V/Hz — Open Loop

Section 3. 3-Phase BLDC Motor Control Application with Hall Sensors

Section 4. 3-Phase Sine Voltage Powered PM Motor Control Application with Hall Sensors

Section 5. 3-Phase ACIM Control with Dead-Time Distortion Correction

List of Figures and Tables

Section 1. MC68HC908MR32 Applications

1.1 Contents

1.2 Introduction

This section describes the common hardware configuration for motor control applications. For more information on the on the MC68HC908MR32 control board, optoisolation board, or 3-phase AC BLDC high-voltage power stage refer to the documnets shown in **Table 1-1**.

Table 1-1. Reference Documents

PC Board	Document Number
MC68HC908MR32 control board	MEMCMR32CBUM/D
Optoisolation board	MEMCOBUM/D
3-phase AC BLDC high-voltage power stage	MEMC3PBLDCPSUM/D

1.3 Settings for EVM Trimpots

Settings for the MC68HC908MR32 control board's required fault trimpots are shown in **Table 1-2**.

Table 1-2. Over-Current and Over-Voltage Adjustments

Power Stage	Over-Current Comparator U5B		Over-Voltage Comparator U5C	
EVM motor board	R34	2.8 Vdc	R35	1.24 Vdc
Low-voltage BLDC power stage	R34	3.3 Vdc	R35	2.5 Vdc
Low-voltage SR power stage	R34	3.3 Vdc	R35	2.5 Vdc
High-voltage AC BLDC power stage	R34	3.3 Vdc	R35	3.07 Vdc
High-voltage SR power stage	R34	3.3 Vdc	R35	3.07 Vdc

Adjust R34 such that the voltage at test point I_ref matches the value indicated in **Table 1-2**. The ground reference is GND_A.

Adjust R35 such that the voltage at test point V_ref matches the value indicated in **Table 1-2**. The ground reference is GND_A.

1.4 Communication Port Settings

If PC master software is to be used for real-time control of motor operations, it is necessary to set up RS-232 serial communication with a PC. To do this, connect a 9-conductor straight-through cable from the MC68HC908MR32 control board's DB-9 connector, J6, to the COM1 or COM2 serial port of the PC.

PC serial ports are wired as DTE (data terminal equipment) and the control board serial communications interface (SCI) port is wired as DCE (data communications equipment). Therefore, a 9-conductor cable wired straight through must be used.

WARNING: Do NOT use a null modem cable.

When utilizing the PC master software debugging tool, the PC's communication port requires the settings shown in **Table 1-3**.

Table 1-3. Communication Protocol Settings

Baud Rate:	9600
Data Bits:	8
Parity:	None
Stop Bit:	1
Flow Control:	None

Section 2. 3-Phase AC Induction Motor Control V/Hz — Open Loop

2.1 Contents

2.2 Introduction

This application demonstrates the principal of V/Hz control of an 3-phase AC induction motor using:

- MC68HC908MR32 control board
- Optoisolation board
- 3-phase AC BLDC high voltage power stage

2.3 Application Files

The 3-phase AC induction motor control V/Hz application is found in the following directory:

..\src\applications\68HC908MR32\3ph_ac_vhz

This application is composed of the following files located in the SDK installation directory:

- **3ph_ac_vhz.mcp (3ph_ac_vhz.prj),** application project file
- **sources\3ph_ac_vhz.c,** main application program
- **sources\3ph_ac_vhz.h,** application header file
- **sources\appconfig.h,** application configuration file for peripheral statical configuration
- **prms\hc908mr32.prm (prms\hc908mr32.lkf),** default linker command file
- **pcmaster\3ph_ac_vhz.pmp**, PC master software file

2.4 Specifications

This application performs principal control of the 3-phase AC induction motor using the MC68HC908MR32 processor. The control technique sets the speed ([rpm], [Hz]) of the magnetic field and calculates the phase voltage amplitude according to a V/Hz table. This table is private to the application and reflects:

- The AC induction motor parameters base voltage/frequency
- Boost voltage/frequency
- DC boost voltage

NOTE: *Protection against application faults over-current, over-voltage, under-voltage, and wrong hardware connections is provided.*

The application can run on:

- 3-phase AC BLDC high-voltage power stage
- 115 V or 230 V 50Hz or 60 Hz power source

The 3-phase AC induction motor control V/Hz application can operate in two modes:

1. Manual Operating Mode

   Refer to **Figure 2-1** for this description. 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 2-2** for this description. If the application runs and motor spinning is disabled (i.e., the system is ready), the green user light-emitting diode (LED) (D11) will blink. When motor rotation is enabled, the green user LED will be *on* and the actual state of the pulse-width modulator (PWM) outputs are indicated by PWM output LEDs, labeled PWM1 - PWM6. If over-current/over-voltage occurs or if the wrong system board is identified, the green user LED will start to flash quickly and the PC master software will signal the identified fault. This state can be exited only by an application reset.

NOTE: *It is strongly recommended you inspect the entire application to locate the source of the fault before starting it again.*

Refer to **Table 2-1** for a description of the application states.

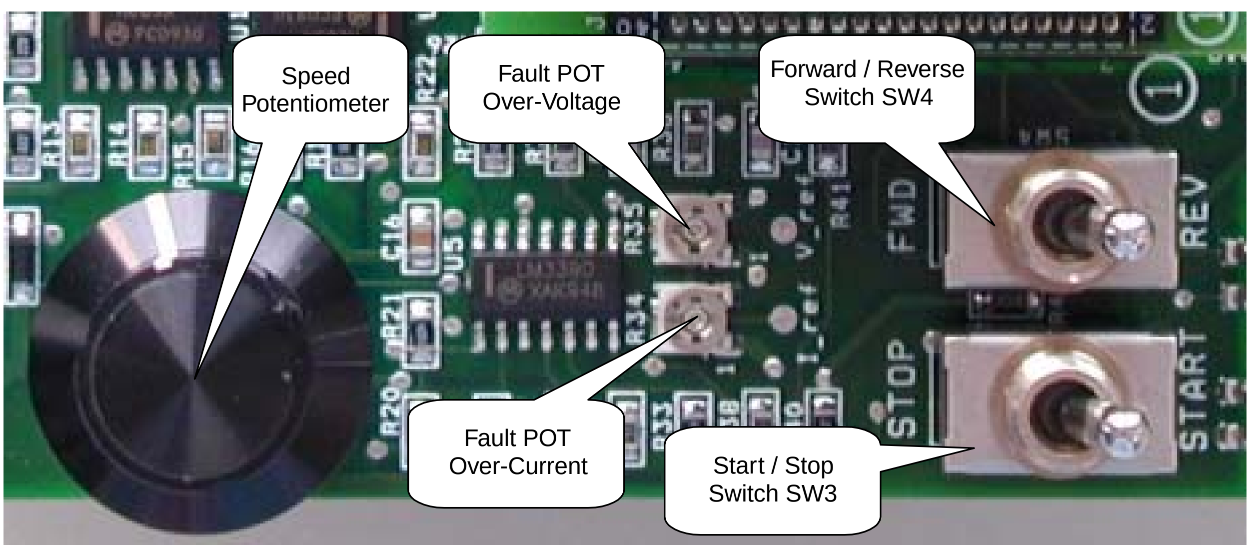

Figure 2-1. EVM Control Elements

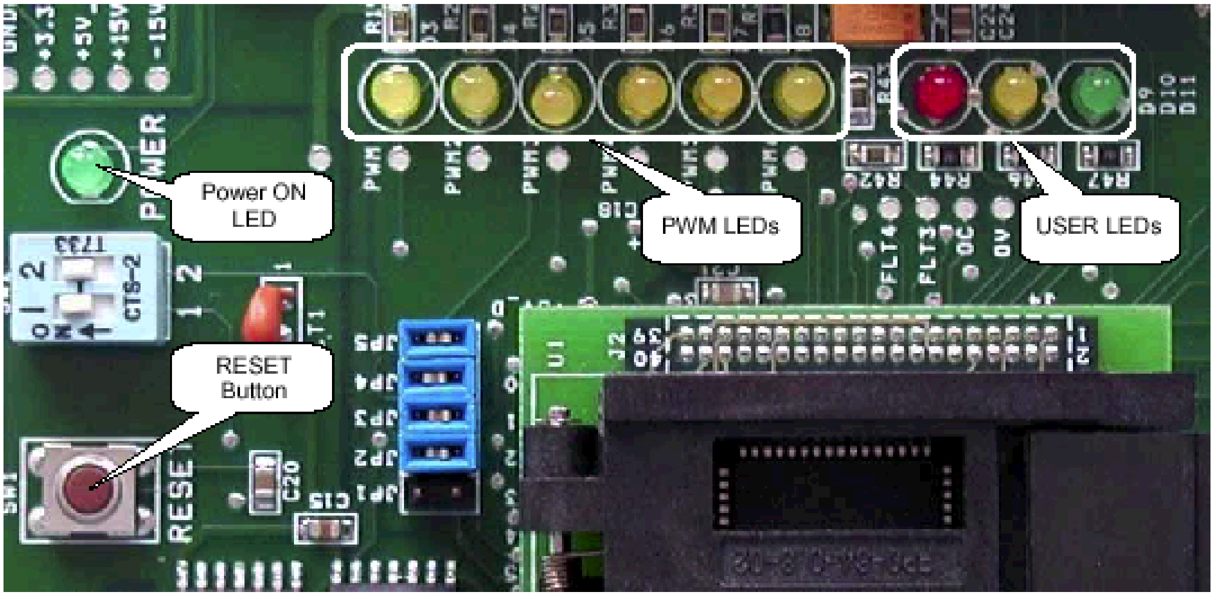

Figure 2-2. USER LEDs, PWM LEDs, and RESET

Table 2-1. 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

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 RUN/STOP switch, which can be used to safely stop the application at any time.

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

The PC master software displays the following information:

- Actual and required speed of the motor
- Phase voltage amplitude (related to given DC bus voltage)
- Application mode — RUN/STOP
- DC bus voltage
- Fault status
- Identified hardware

In a case where over-current, over-voltage, or under-voltage fault occurs, internal fault logic is asserted and the application enters a fault state (user LED will start to flash quickly). This state can be exited only if the fault state is corrected and the fault is acknowledged by toggling the START/STOP switch through the STOP state.

NOTE: *It is strongly recommended that you inspect the entire application to locate the source of the fault before starting it again.*

Project files for the PC master software are located in:

..\nos\applications\3ph_ac_vhz\pcmaster\3ph_ac_vhz.pmp

Start the PC master software window's application and choose the PC master software project for the desired PC master software operating mode. Figure 2-3 shows the PC master software control window for ***3ph_ac_vhz.pmp***.

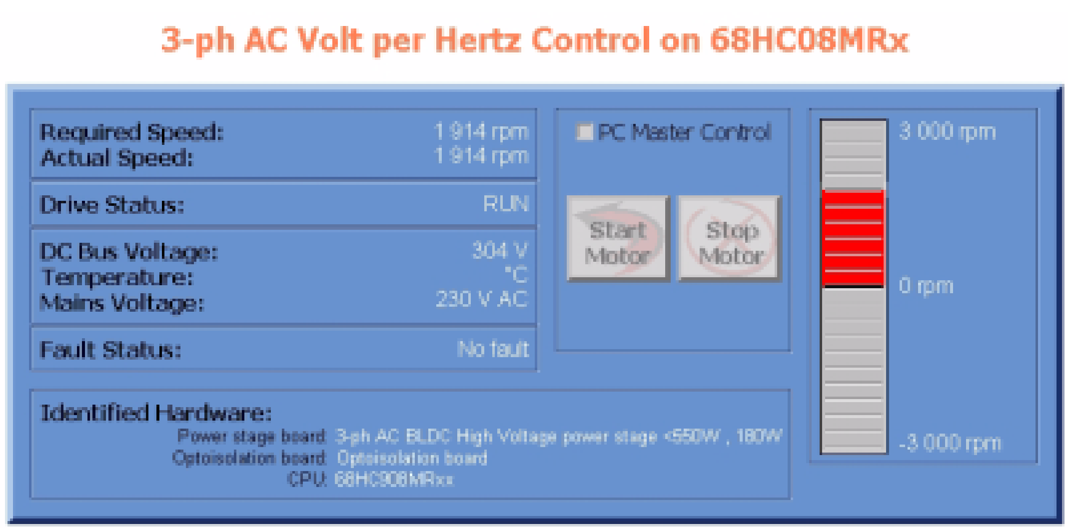

Figure 2-3. PC Master Software Control Window

2.5 Hardware Setup

Figure 2-4 illustrates the hardware setup for the 3-phase AC open loop motor control application. The correct phase order (phase A, phase B, phase C) for the AC induction motor shown is:

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

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

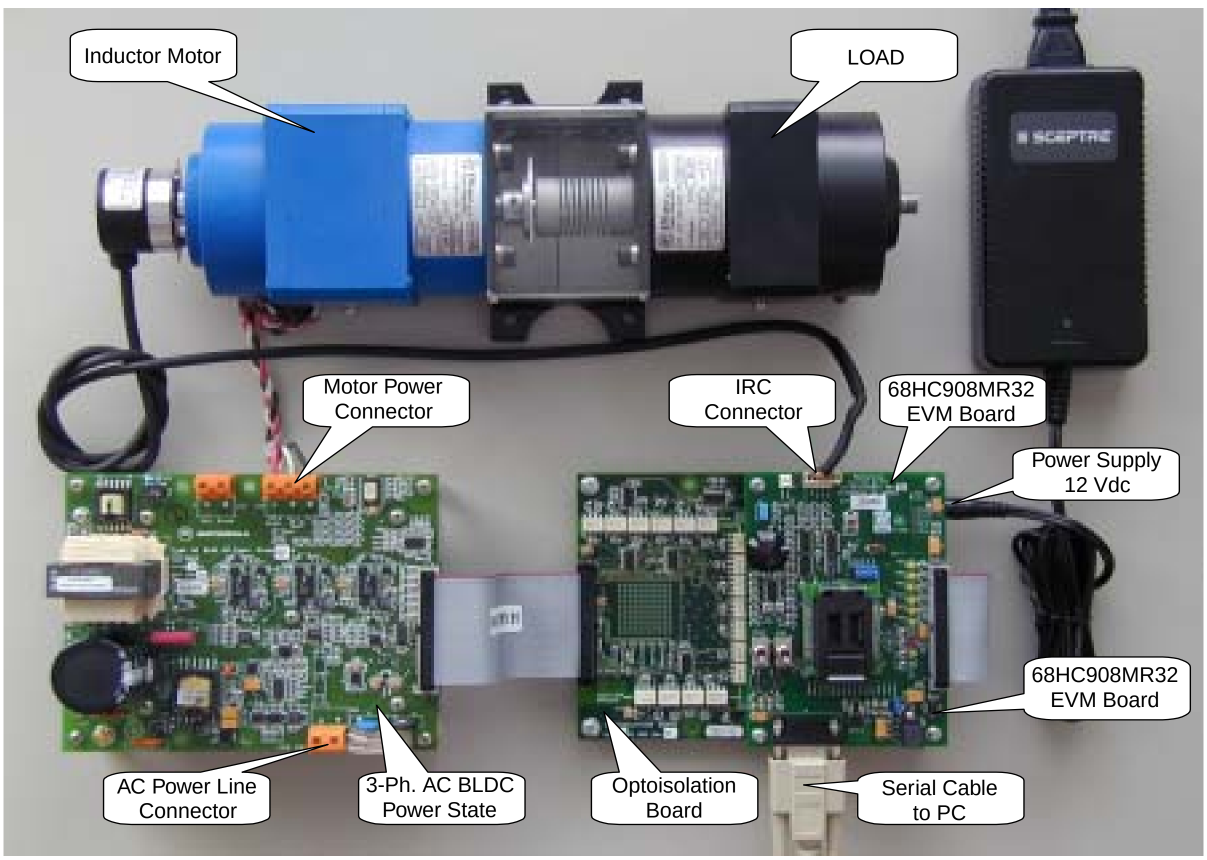

Figure 2-4. Setup of the 3-phase AC Induction Motor Control Application — Open Loop

2.6 EVM Jumper Settings

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

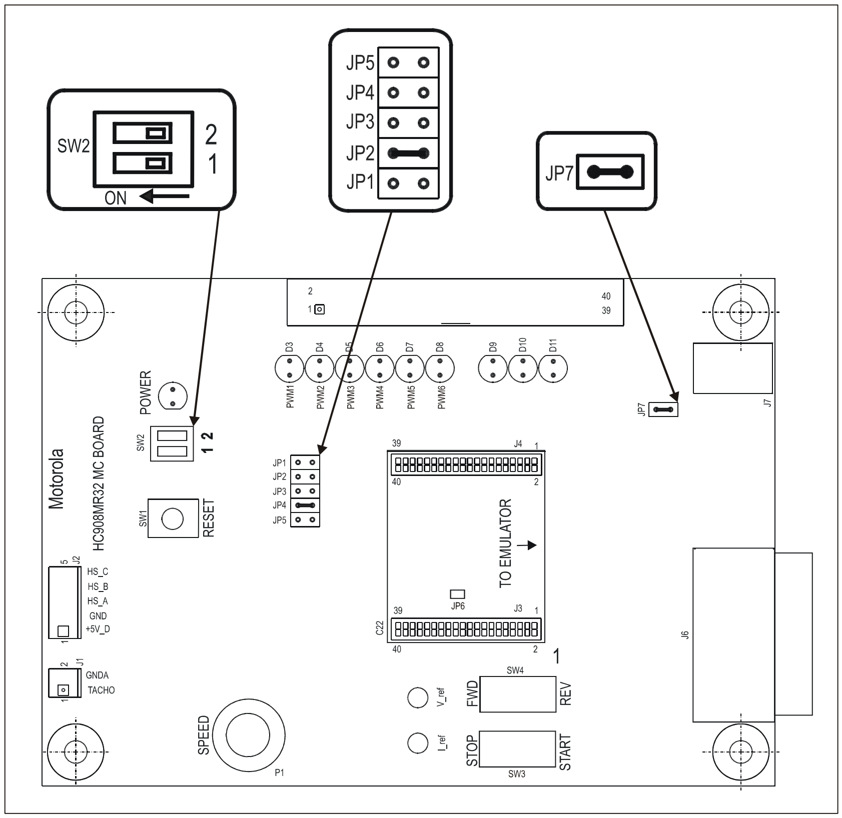

Figure 2-5. MC68HC908MR32 Jumper Reference

Table 2-2. 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

2.7 Building the Application

To build this application, open the ***3ph_ac_vhz.mcp*** project file and execute the *Make* command; see **Figure 2-6**. This command will build and link the 3-phase AC V/Hz motor control application along with all needed Metrowerks(1) and SDK libraries.

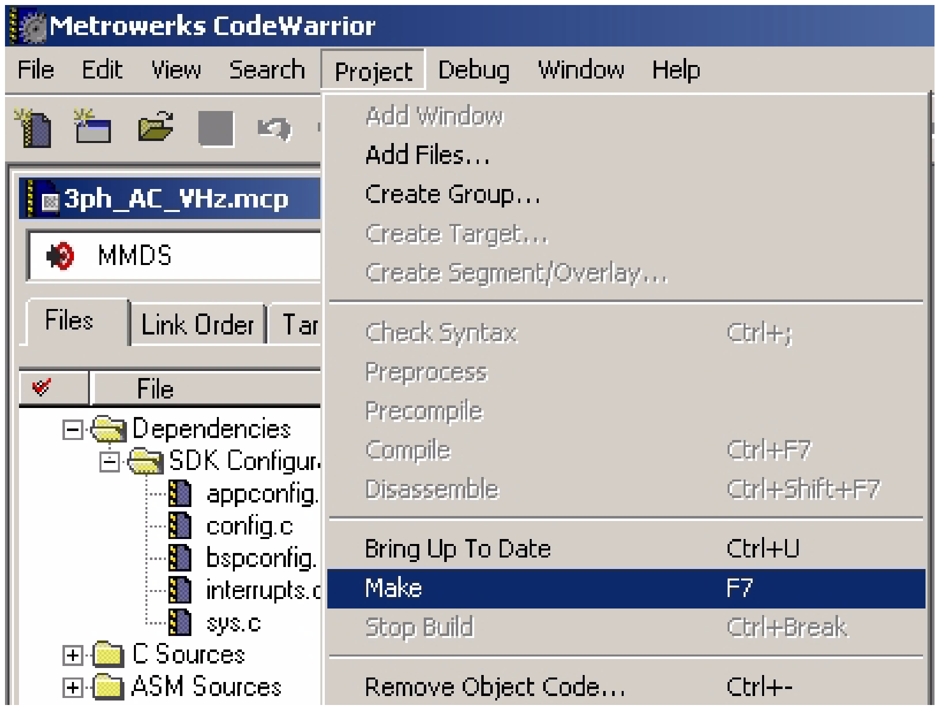

Figure 2-6. Execute *Make* Command

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

2.8 Executing the Application

To execute the 3-phase AC V/Hz motor control application, choose the *Program/Debug* command in the CodeWarrior®[(1)] IDE, followed by the *Run* command.

For more help with these commands, refer to the CodeWarrior tutorial documentation in the following file located in the CodeWarrior installation directory:

<...>\info\CodeWarrior\IDE_User_Guide.pdf

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

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 RUN/STOP switch is set to the RUN position when the application starts, toggle the RUN/STOP switch between the STOP and RUN positions to enable motor spinning. This is a protection feature preventing the motor from starting 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.

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

Section 3. 3-Phase BLDC Motor Control Application with Hall Sensors

3.1 Contents

3.2 Introduction

This application exercises simple control of the BLDC motor with Hall sensors on the MC68HC908MR32 control board and the EVM motor kit. The application is found in directory:

..\src\68HC908MR32\Applications\3ph_bldc_hs

3.3 Application Files

The BLDC motor control application with Hall sensors is composed of the following files:

- **3ph_bldc_hs.mcp (3ph_bldc_hs.prj)**, application project file
- **sources\3ph_bldc_hs.c**, main program
- **sources\3ph_bldc_hs.h**, main program header file
- **sources\appconfig.h**, application configuration file for static periphery configuration
- **prms\hc908mr32.prm (prms\hc908mr32.lkf)**, linker parameters file
- **pcmaster\3ph_bldc_hs.pmp**, PC master software file

These files are located in the 8-bit SDK installation directory.

3.4 Specifications

This application performs a 3-phase, 4-quadrant sensor BLDC motor drive with voltage and angle control technique. It is based on Motorola's MC68HC908MR32 family dedicated to motor control applications. The software design uses the 8-bit software development kit (SDK) developed by Motorola.

The concept of the application allows both closed and open-loop speed control. It serves as an example of a sensor BLDC motor control system using Motorola's M68HC08 Family with 8-bit SDK support. It also illustrates the usage of dedicated motor control on chip peripherals, software drivers, and software libraries which are included in the 8-bit SDK.

This BLDC motor control application with Hall sensors can operate in two modes:

1. Manual operating mode
2. PC master software (remote) operating mode

3.5 Manual Operating Mode

Refer to MC68HC908MR32 control board shown in **Figure 3-1** for this description:

- The drive is controlled by the START/STOP switch (SW3), see [1].
- The motor speed is set by the SPEED potentiometer (P1), see [4].
- If the application runs and motor rotation is disabled (i.e., the system is ready), the green USER LED (D11) shown in [9] will blink.
- When motor rotation is enabled, the USER LED is *On*.

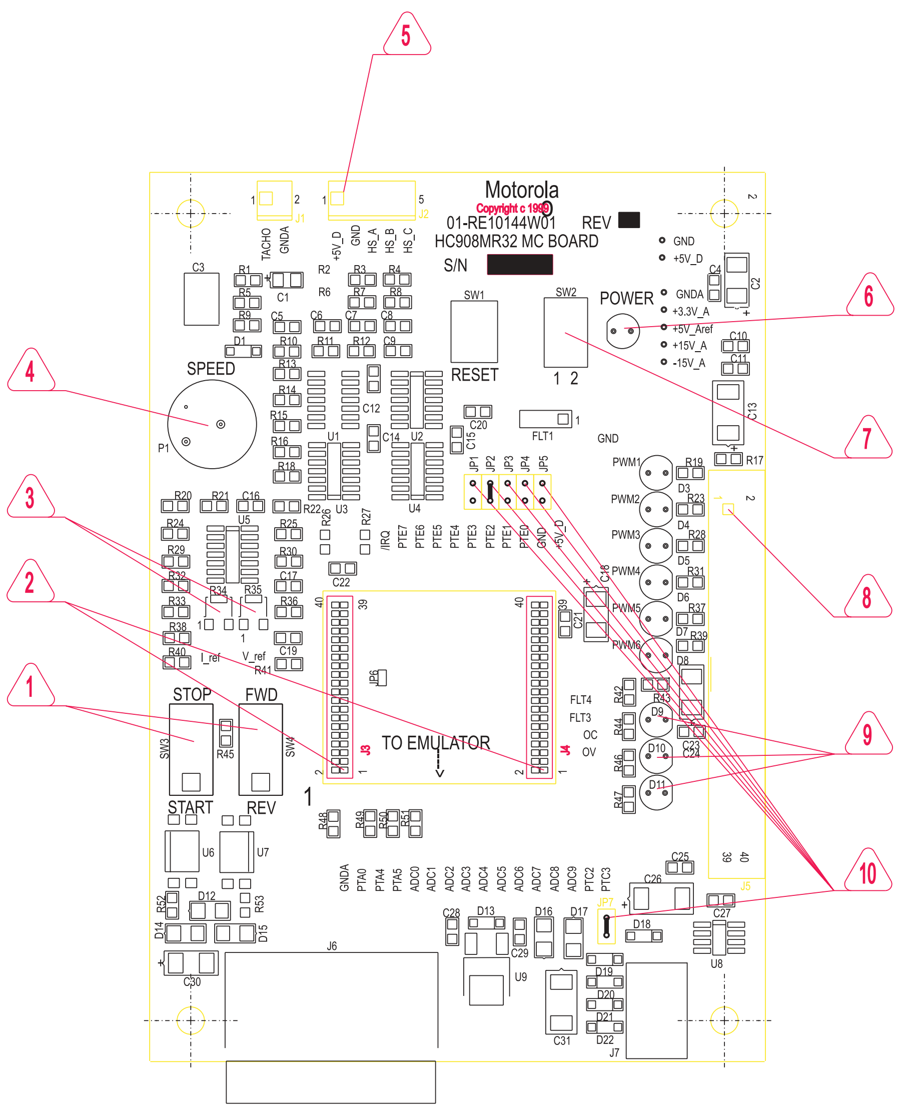

Figure 3-1. MC68HC908MR32 Board

Application states are defined in **Table 3-1**.

Table 3-1. Motor Application States

Application State	Motor State	Green LED State	Yellow LED State	Red LED State
Stopped	Stopped	Blinking at a frequency of 2 Hz	—	—
Running	Spinning	On	—	—
Running	De-excitation	On	On	—
Fault (under voltage, sensor fault)	Stopped	Blinking at a frequency of 8 Hz	—	—
Critical Fault (over voltage, over current)	Stopped	Blinking at a frequency of 8 Hz	—	On

The following control actions are supported:

- Set the required speed of the motor
- Motor reverse
- Switch between close loop and open loop
- Allow Hall sensor identification
- Control board

EVM control elements (see **Figure 3-1**):

1. Main board switches
2. Emulator / MC68HC908MR32 microcontroller socket
3. Over-current and over-voltage trimpot
4. Speed potentiometer
5. Hall sensor connector
6. Power indicator
7. PC board switches
8. Connector to power stage
9. User light emitting diode (LED)
10. Hardware configuration jumpers

3.6 PC Master (Remote) Operating Mode

The drive is controlled remotely from a PC via an RS-232 physical interface. The drive is enabled by the RUN/STOP switch. This switch can be used to safely stop the application at any time.

Targeting Motorola's MC68HC908MR32 Platform, the PC master software displays the following information:

- Applied voltage
- Required voltage
- Speed
- Direction
- RUN/STOP switch status
- CLOSE LOOP/OPEN LOOP
- Application mode

Project files for PC master software are located in:

PC master software file
...\applications\3ph_bldc_hs\sources\pcmaster\3ph_bldc_hs.pmp

To start the PC master software's window application
3ph_bldc_hs.pmp

NOTE: *If the PC master project (**.pmp** file) is unable to control the application, it is possible the wrong load map (**.map** file) has been selected. The PC master software uses the load map to determine addresses for global variables being monitored. Once the PC master project has been launched, this option may be selected in the PC master window under "Project/Select other Map File Reload".*

The PC master software control window is shown in **Figure 3-2**.

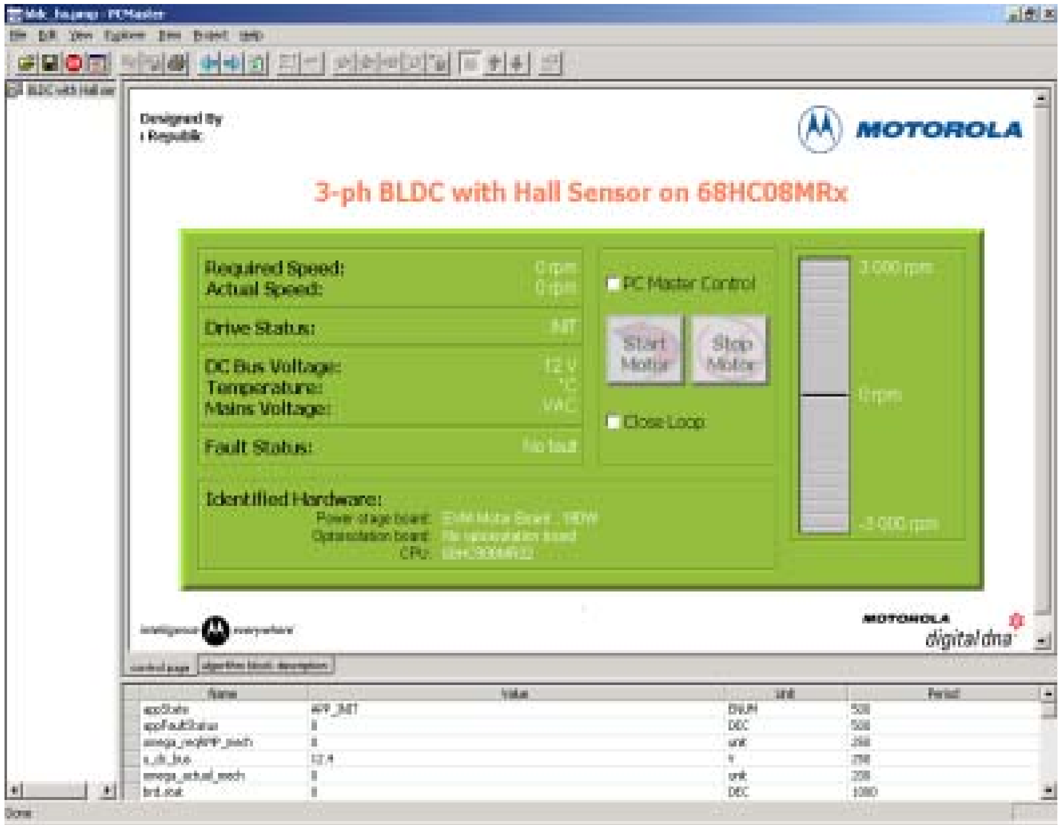

Figure 3-2. PC Master Control Window

3.7 Hardware Setup

The hardware setup for the BLDC motor control application with Hall sensors includes:

- Motorola MC69HC908MR32 control board
- 3-phase AC BLDC low voltage power stage or EVM motor board
- BLDC motor with Hall sensors
- Power supply
- Serial cables to PC

For more detailed information, refer to the *MC68HC908MR32 Control Board User's Manual* (Motorola document order number MEMCMR32CBUM/D).

3.8 EVM Jumper Settings

To execute the BLDC motor control application with Hall sensors, the MC68HC908MR32 control board requires the jumper settings shown in **Table 3-2**.

NOTE: *The JP2 jumper must be connected.*

Table 3-2. MCHC908MR32 Board Jumper Settings

Jumper Group	Comment	Connections
JP1	Tacho	No connection
JP2	Encoder	1–2
JP3	BEMF_z_c	No connection
JP4	PFC_z_c	No connection
JP5	PFC_PWM	No connection
JP7	GND_Connection	1–2

3.9 Building the Application

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

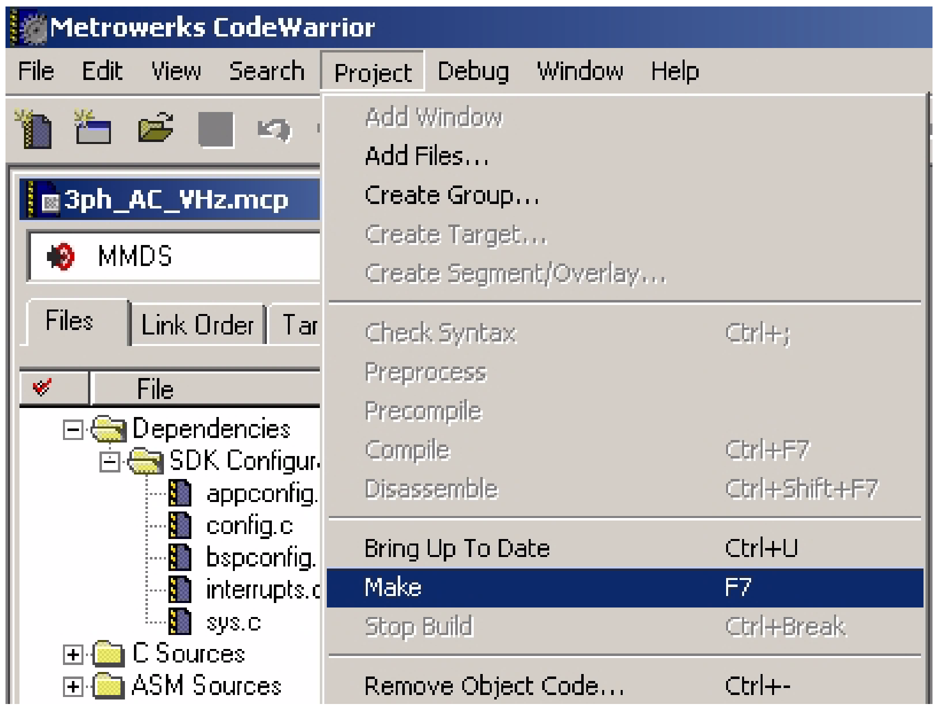

Figure 3-3. Target Build Selection

3.10 Executing the Application

To execute the BLDC motor control application with Hall sensors, select *Project\Debug* in the CodeWarrior IDE, followed by the *Run* command. For more help with these commands, refer to the CodeWarrior tutorial documentation in the following file located in the CodeWarrior installation directory:

<...>\info\CodeWarrior\IDE_User_Guide.pdf

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

3.10.1 Starting the Motor in Manual Mode

Switch the START/STOP switch to the START position and set the required speed by the SPEED potentiometers. If successful, the BLDC motor will be spinning.

NOTE: *If the START/STOP switch is set to the START position when the application starts, toggle the START/STOP switch between the STOP and START positions to enable motor spinning. This is a protection feature preventing the motor from starting 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 frequency of 2 Hz. If a fault occurs, the green LED will blink at a frequency of 8 Hz.

3.10.2 Switches SW2

In board control mode, the SW2–1 switch on the CPU board determines close/open loop (close loop is at position On). SW2–2 allows Hall sensor identification (identification is allowed in position On). When Hall sensor identification is allowed, before first start of the motor the sensor identification algorithm will make a table with rotor positions assigned for each Hall sensor code. Otherwise, the table for a standard arrangement is used.

3.10.3 PC Master Mode Control

To set the PC master control, perform the following steps:

1. The RUN/STOP switch must be in the STOP position
2. Check the PC master mode on the PC master control page
3. Enabled the application by setting the RUN/STOP switch in the RUN position
4. Start the motor by pressing the Start PC Master Push Button and stop the motor by releasing the button
5. Set the speed with the bar graph
6. The motor can be stopped any time with the RUN/STOP switch on the EVM. When the RUN/STOP switch on the EVM is in the STOP position, manual mode can be set again by unchecking PC master mode on the PC master control page.

Section 4. 3-Phase Sine Voltage Powered PM Motor Control Application with Hall Sensors

4.1 Contents

4.2 Introduction

This application exercises simple control of the PM motor with Hall sensors on the MC68HC908MR32 control board and the EVM motor kit. The application is found in directory:

..\src\68HC908MR32\Applications\3ph_pm_sin_3hs

4.3 Application Files

The PM motor control application with Hall sensors is composed of the following files:

- **3ph_pm_sin_3hs.mcp**, application project file
- **sources\3ph_pm_sin_3hs.c**, main program
- **sources\3ph_pm_sin_3hs.h**, main program header file
- **sources\appconfig.h**, application configuration file for static periphery configuration
- **prms\hc908mr32.prm**, linker parameters file
- **pcmaster\3ph_pm_sin_3hs.pmp**, PC master software file

These files are located in the 8-bit SDK installation directory.

4.4 Specifications

This application performs a 3-phase, sine voltage powered PM motor drive with hall sensor in each phase. It is based on Motorola's MC68HC908MR32 Family dedicated to motor control applications. The software design uses the 8-bit software development kit (SDK) developed by Motorola.

The concept of the application allows both closed and open-loop speed control. It serves as an example of a sensor PM motor control system using Motorola's M68HC08 Family with 8-bit SDK support. It also illustrates the usage of dedicated motor control on chip peripherals, software drivers, and software libraries which are included in the 8-bit SDK.

The application was derived from the application described in *Sine Voltage Powered 3-Phase Permanent Magnet Motor with Hall Sensor* (Motorola document order number AN2357). The difference is that the torque limitation was disabled and three Hall sensors were used instead of one.

This PM motor control application with Hall sensors can operate in two modes:

1. Manual operating mode
2. PC master software (remote) operating mode

4.5 Manual Operating Mode

Refer to MC68HC908MR32 control board shown in **Figure 4-1** for this description:

- The drive is controlled by the START/STOP switch (SW3), see [1].
- The motor speed is set by the SPEED potentiometer (P1), see [4].
- If the application runs and motor rotation is disabled (i.e., the system is ready), the green USER LED (D11) shown in [9] will blink.
- When motor rotation is enabled, the USER LED is *On*.

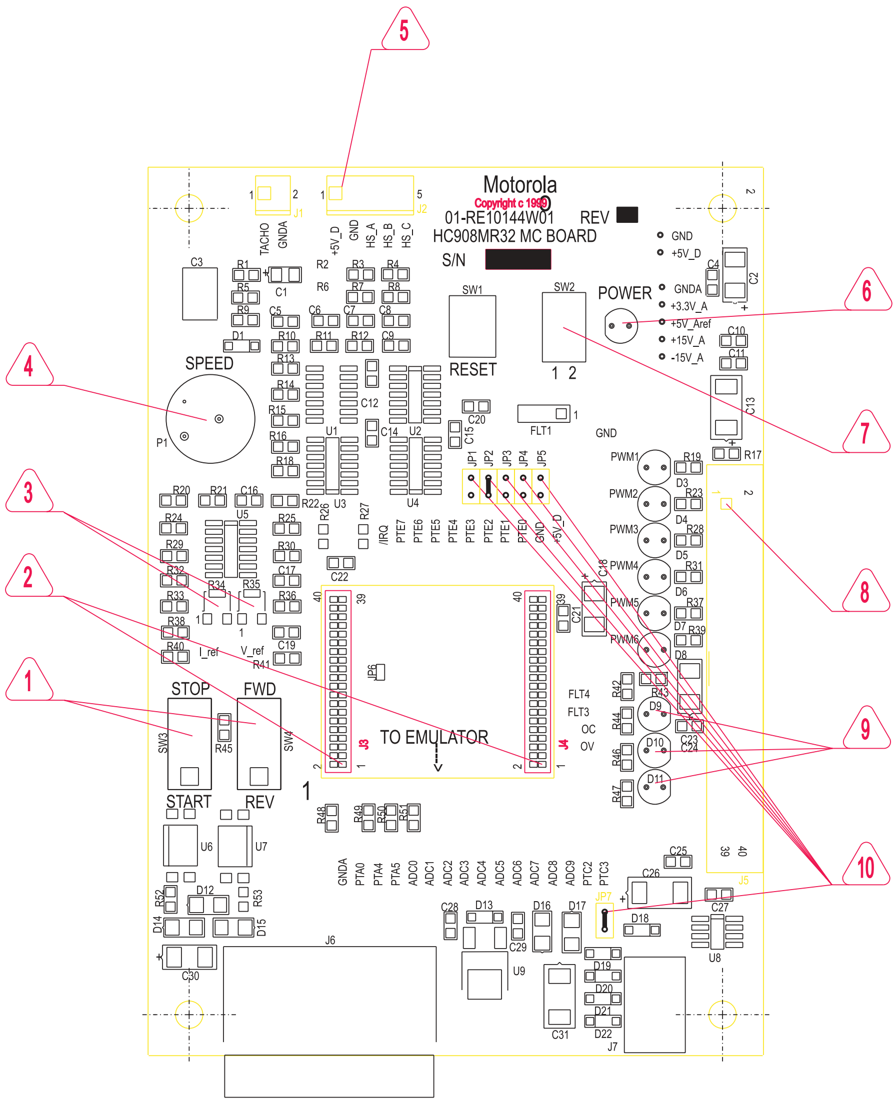

Figure 4-1. MC68HC908MR32 Board

Application states are defined in Table 4-1.

Table 4-1. Motor Application States

Application State	Motor State	Green LED State	Yellow LED State	Red LED State
Stopped	Stopped	Blinking at a frequency of 2 Hz	—	—
Running	Spinning	On	—	—
Running	De-excitation	On	On	—
Fault (under voltage, sensor fault)	Stopped	Blinking at a frequency of 8 Hz	—	—
Critical Fault (over voltage, over current)	Stopped	Blinking at a frequency of 8 Hz	—	On

The following control actions are supported:

- Set the required speed of the motor
- Switch between close loop and open loop
- Start/Stop

EVM control elements (see Figure 4-1):

1. Main board switches
2. Emulator / MC68HC908MR32 microcontroller socket
3. Over-current and over-voltage trimpot
4. Speed potentiometer
5. Hall sensor connector
6. Power indicator
7. PC board switches
8. Connector to power stage
9. User light emitting diode (LED)
10. Hardware configuration jumpers

4.6 PC Master (Remote) Operating Mode

The drive is controlled remotely from a PC via an RS-232 physical interface. The drive is enabled by the RUN/STOP switch. This switch can be used to safely stop the application at any time.

Targeting Motorola DSP56F80X Platform, the PC master software displays the following information:

- Applied voltage
- Required voltage
- Speed
- Direction
- RUN/STOP switch status
- CLOSE LOOP/OPEN LOOP
- Application mode

Project files for PC master software are located in:

PC master software file

...\applications\3ph_pm_sin_3hs\sources\pcmaster\3ph_pm_sin_3hs.pmp

To start the PC master software's window application

3ph_pm_sin_3hs.pmp

NOTE: *If the PC master project (**.pmp** file) is unable to control the application, it is possible the wrong load map (**.map** file) has been selected. The PC master software uses the load map to determine addresses for global variables being monitored. Once the PC master project has been launched, this option may be selected in the PC master window under "Project/Select other Map File Reload".*

The PC master software control window is shown in Figure 4-2.

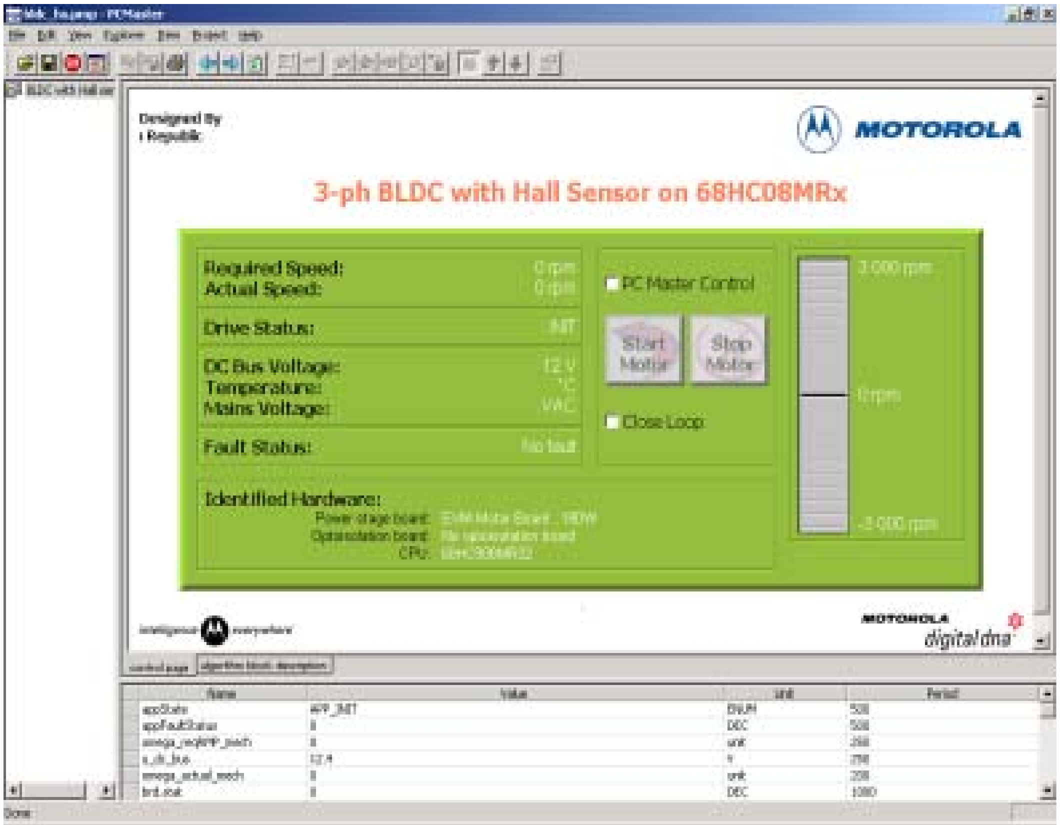

Figure 4-2. PC Master Control Window

4.7 Hardware Setup

The hardware setup for the PM motor control application with Hall sensors includes:

- Motorola MC69HC908MR32 control board
- 3-phase AC BLDC low-voltage power stage or EVM motor board
- PM motor with Hall sensors
- Power supply
- Serial cables to PC

For more detailed information, refer to the *MC68HC908MR32 Control Board User's Manual* (Motorola document order number MEMCMR32CBUM/D).

4.8 EVM Jumper Settings

To execute the PM motor control application with Hall sensors, the MC68HC908MR32 control board requires the jumper settings shown in **Table 4-2**.

NOTE: *The JP2 jumper must be connected.*

Table 4-2. MCHC908MR32 Board Jumper Settings

Jumper Group	Comment	Connections
JP1	Tacho	No connection
JP2	Encoder / Hall Sensor	1–2
JP3	BEMF_z_c	No connection
JP4	PFC_z_c	No connection
JP5	PFC_PWM	No connection
JP7	GND_Connection	1–2

4.9 Building the Application

To build this application, open the ***3ph_pm_sin_3hs.mcp*** project file and execute the *Make* command; see **Figure 4-3**. This will build and link PM motor control application with Hall sensors along with all needed Metrowerks and SDK libraries.

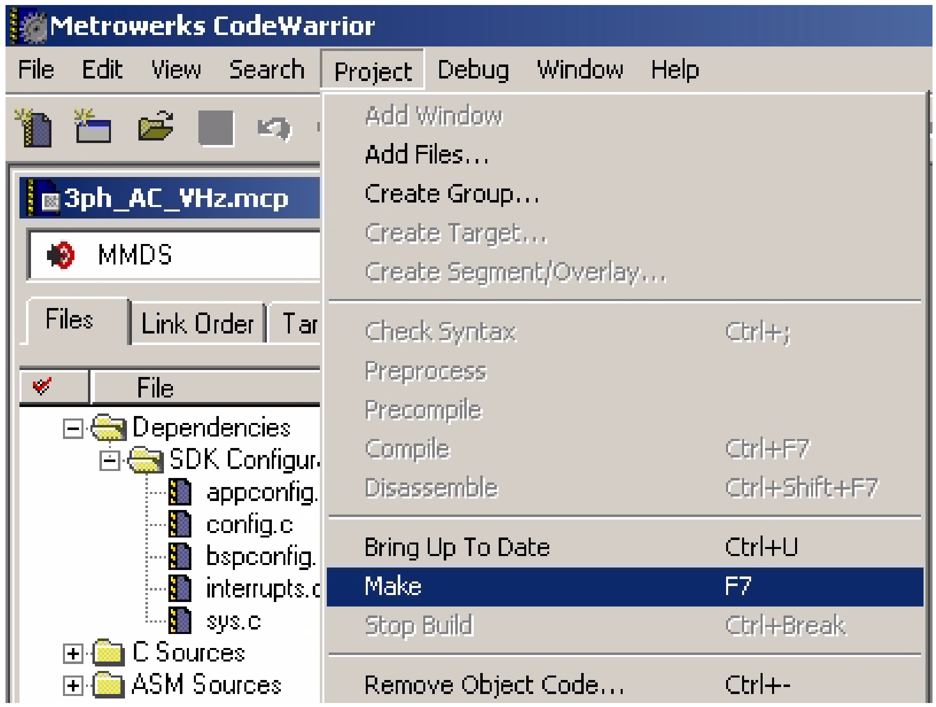

Figure 4-3. Target Build Selection

4.10 Executing the Application

To execute the PM motor control application with Hall sensors, select *Project\Debug* in the CodeWarrior IDE, followed by the *Run* command. For more help with these commands, refer to the CodeWarrior tutorial documentation in the following file located in the CodeWarrior installation directory:

<...>\info\CodeWarrior\IDE_User_Guide.pdf

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

4.10.1 Starting the Motor in Manual Mode

Switch the START/STOP switch to the START position and set the required speed by the SPEED potentiometers. If successful, the PM motor will be spinning.

NOTE: *If the START/STOP switch is set to the START position when the application starts, toggle the START/STOP switch between the STOP and START positions to enable motor spinning. This is a protection feature preventing the motor from starting 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 frequency of 2 Hz. If a fault occurs, the green LED will blink at a frequency of 8 Hz.

4.10.2 Switches SW2

In board control mode, the SW2–1 switch on the CPU board determines close/open loop (close loop is at position On).

4.10.3 PC Master Mode Control

To set the PC master control, perform the following steps:

1. The RUN/STOP switch must be in the STOP position
2. Check the PC master mode on the PC master control page
3. Start the motor by pressing the Start PC Master Push Button and stop the motor by releasing the button
4. Set the speed with the bar graph

Section 5. 3-Phase ACIM Control with Dead-Time Distortion Correction

5.1 Contents

5.2 Introduction

This application demonstrates the principal of dead-time distortion correction of a 3-phase AC induction motor using:

- MC68HC908MR32 control board
- Optoisolation board
- 3-phase AC BLDC high-voltage power stage at 115 V or 230 V 50 Hz or 60 Hz power source

The dead time is a short delay that needs to be inserted between the turning off of one transistor in the power stage half bridge, and the turn-on of its complementary transistor. As a result, distortion is introduced to the phase voltage during dead time.

This phase voltage distortion causes distortion of the phase current, and thus a deterioration of the motor performance. It is especially apparent in low speeds, when the dead time is comparable with the pulse-width modulator (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)

5.3 Application Files

The application of 3-phase AC induction motor control with dead-time correction is found in the following directory:

..\src\applications\68HC908MR32\3ph_acim_dt_correct

This application is composed of the following files located in the SDK installation directory:

- **3ph_acim_dt_correct.mcp (3ph_acim_dt_correct.prj),** application project file
- **sources\3ph_acim_dt_correct.c,** main application program
- **sources\3ph_acim_dt_correct.h,** application header file
- **sources\appconfig.h,** application configuration file for peripheral statical configuration
- **prms\hc908mr32.prm (prms\hc908mr32.lkf),** default linker command file
- **pcmaster\3ph_acim_dt_correct.pmp**, PC master software file

5.4 Specifications

The application is designed to drive a 3-phase AC motor in an open speed loop mode with dead-time distortion correction. 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. The sine wave generator generates the PWM values for all three phases of the AC bridge inverter.

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 MC68HC908MR32 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 dead-time correction algorithm also detects the magnitude of the phase currents (low/high), and implements advanced software which improves the correction results. The user has the choice of selecting either of the correction algorithms. Both algorithms are described in detail in:

- The algorithm documentation
- Application note entitled *Making Low-Distortion Motor Waveforms with the MC68HC708MP16* (Motorola document order number AN1728),
- The dead-time correction algorithm description that is part of the 8-Bit SDK motor control algorithms library

The drive incorporates fault protection, so in the case of over-current, over-voltage, or under-voltage faults, internal fault logic is asserted and the application enters a fault state (green status light emitting diode (LED) will start to blink quickly). This state can be exited only if the fault disapears and it is acknowledged, by toggling the START/STOP switch through the STOP state.

NOTE: *It is strongly recommended that you inspect the entire application to locate the source of the fault before starting it again.*

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 5-1** for this description.

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 a supported control action.

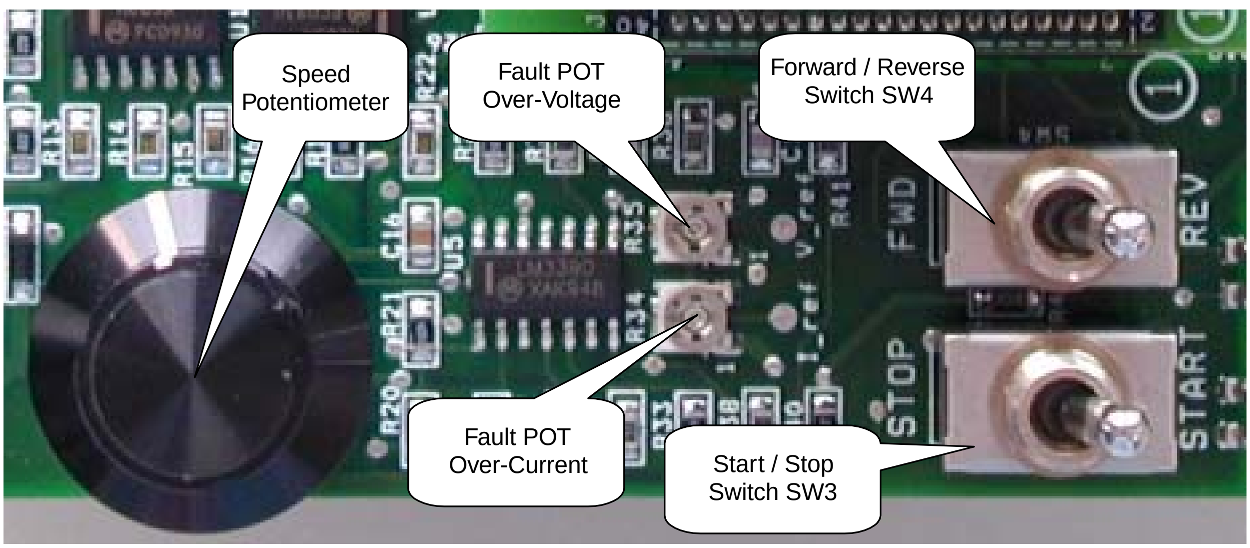

Figure 5-1. EVM Control Elements

The application states are displayed by on-board LED's. Refer to **Table 5-1** for a description of the application states and to **Figure 5-2** for the on-board LED's position.

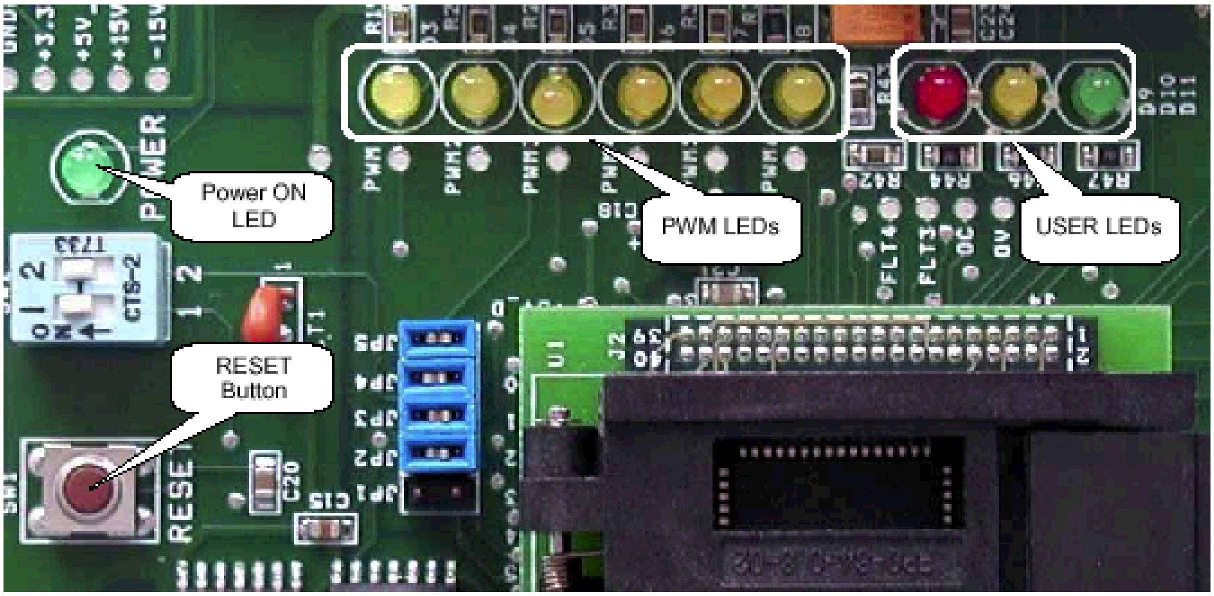

Figure 5-2. USER LEDs, PWM LEDs, and RESET

Table 5-1. Motor Application States

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

If the application runs and motor spinning is disabled (i.e., the system is ready), the green status LED (D11) will blink. When motor rotation is enabled, the green status LED will be on, and the actual state of the PWM outputs are indicated by 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 by an application reset.

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
- Identified hardware

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 type of dead-time distortion correction is indicated by a yellow LED on the MR32 controller board. Refer to **Table 5-2** for a description of the LED states and to **Figure 5-2** for the on-board LED's position. When the dead-time distortion correction is disabled, the yellow LED is turned off. When partial correction is selected, the LED flashes with 2-Hz frequency. With full correction, the LED is turned on.

Table 5-2. Dead Time Distortion Correction

Distortion Correction	Yellow LED State
Disabled	Off
Partial (hardware)	Blinking at a frequency of 2 Hz
Full (software)	On

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

Project files for the PC master software are located in:

..\nos\applications\3ph_acim_dt_correct\pcmaster\3ph_acim_dt_correct.pmp

Start the PC master software application window and choose the appropriate PC master software project. **Figure 5-3** 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 (4 kHz/8 kHz/16 kHz/32 kHz) can be selected in the variables pane, as shown in **Figure 5-3**.

NOTE: *The desired dead time can be set in the application configuration file appconfig.h, where all on-chip modules of the MC68HC908MR32 microcontroller are initialized.*

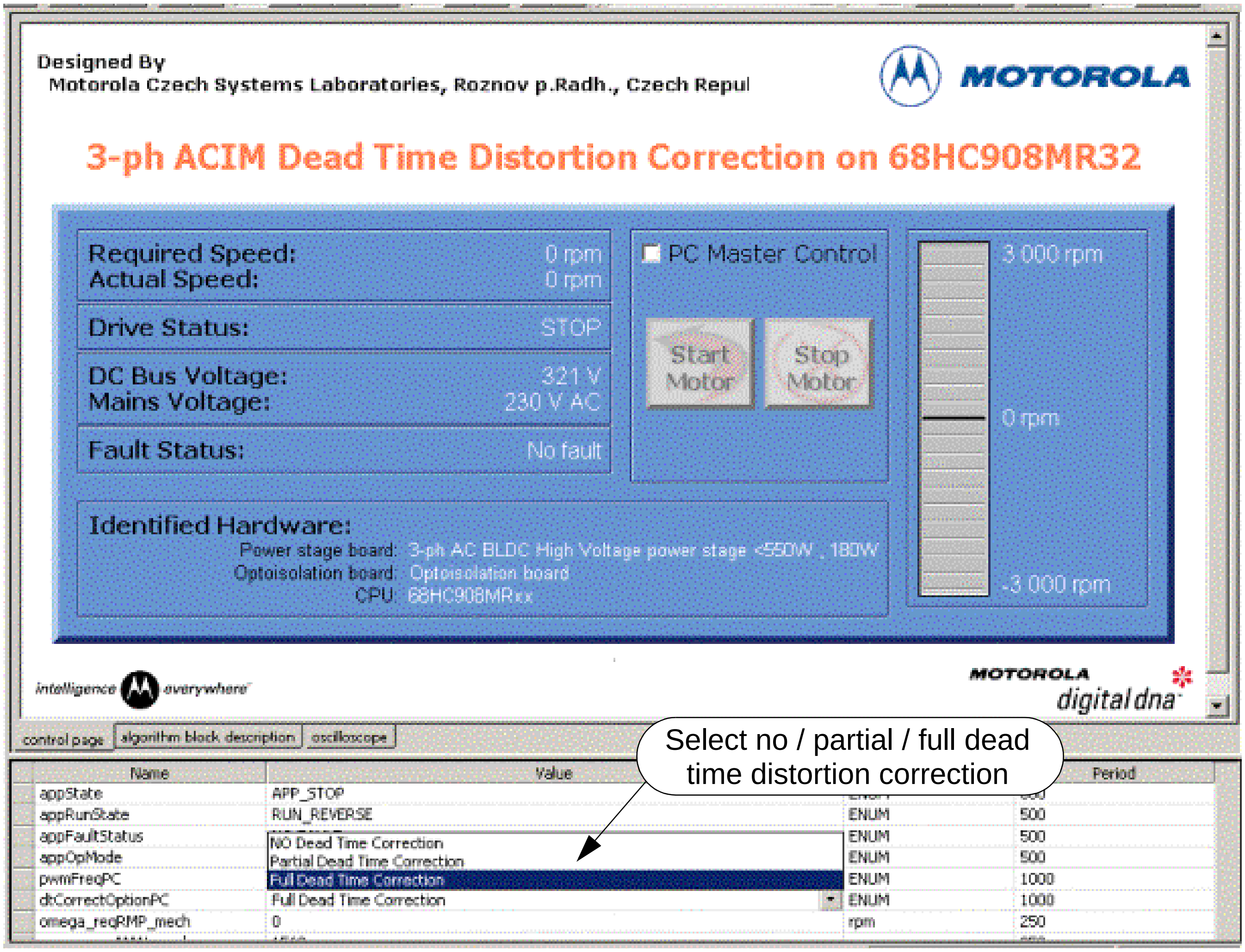

Figure 5-3. PC Master Software Control Window

5.5 Hardware Setup

Figure 5-4 illustrates the hardware setup for the application. The correct phase order (phase A, phase B, phase C) for the AC induction motor shown 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).

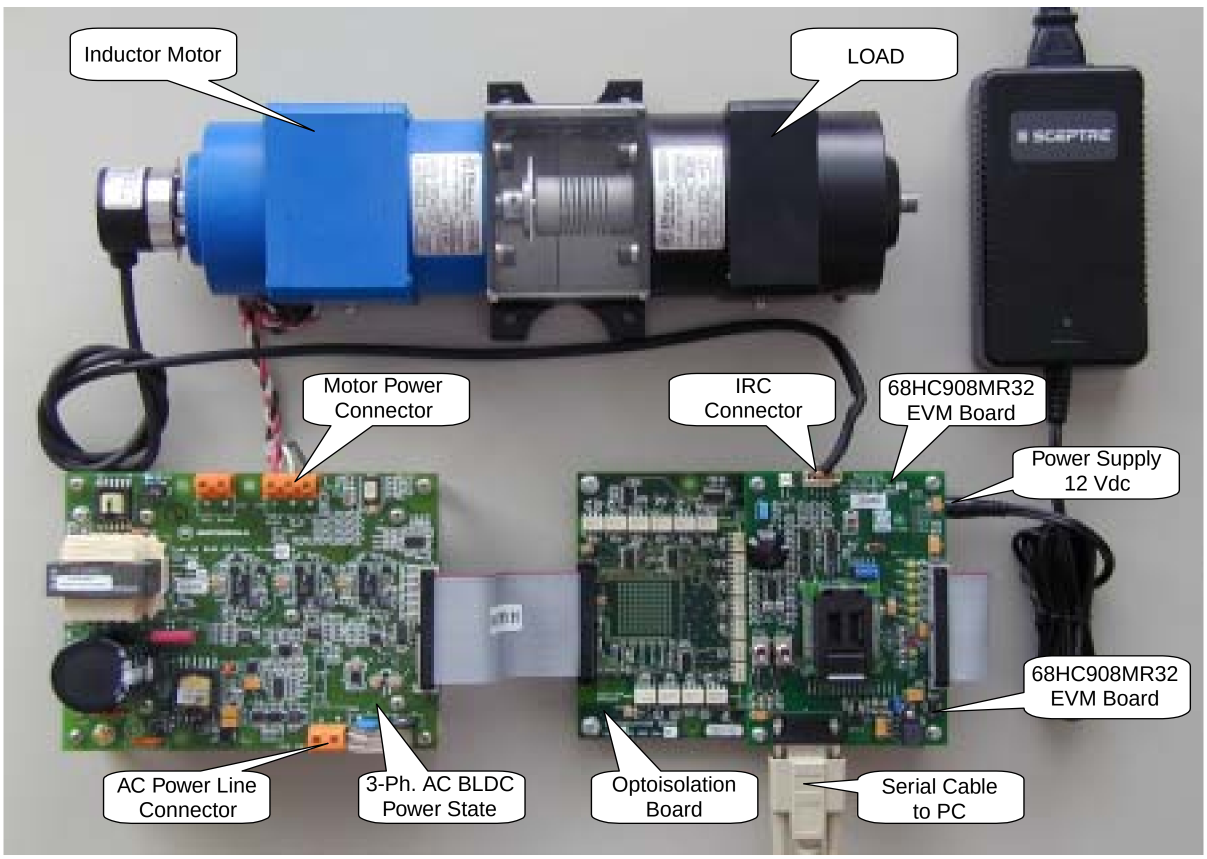

Figure 5-4. Setup of the Application 3-phase AC Induction Motor Control with Dead-Time Distortion Correction EVM Jumper Settings

The MC68HC908MR32 control board jumper settings shown in **Figure 5-5** and **Table 5-3** 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).

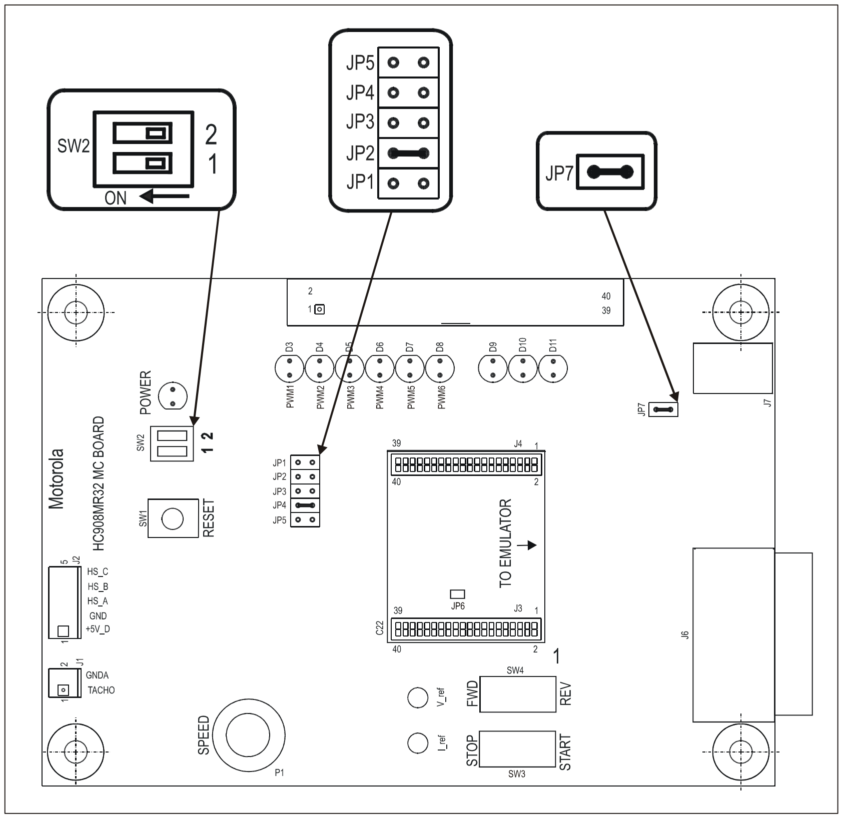

Figure 5-5. MC68HC908MR32 Jumper Reference

Table 5-3. 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

5.6 Building the Application

To build this application, open the ***3ph_acim_dt_correct.mcp*** project file and execute the *Make* command; see **Figure 5-6**. This command will build and link the motor control application along with all needed Metrowerks and SDK libraries.

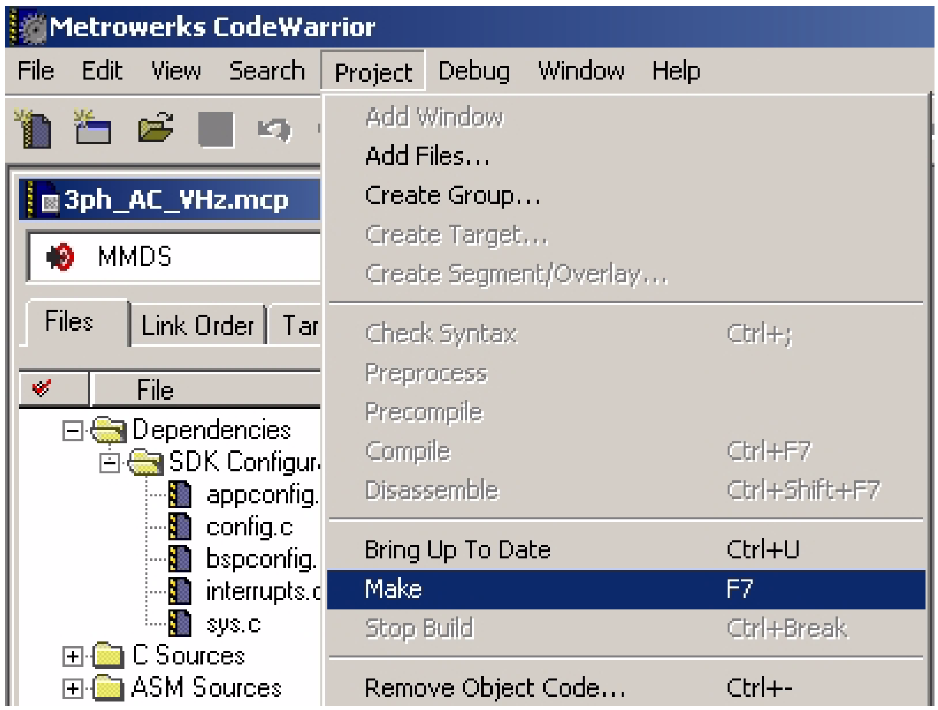

Figure 5-6. Execute *Make* Command

5.7 Executing the Application

To execute the motor control application, choose the *Program/Debug* command in the CodeWarrior IDE, followed by the *Run* command.

For more help with these commands, refer to the CodeWarrior tutorial documentation in the following file located in the CodeWarrior installation directory:

<...>\info\CodeWarrior\IDE_User_Guide.pdf

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

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 from starting 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 motor can be observed using the oscilloscope, and the efficiency of dead-time distortion correction can be evaluated. The yellow LED indicates the selected type of dead-time distortion correction algorithm.

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

SDKMR32UG/D
Rev. 1
11/2002