Z8FMC16100 Series of Flash MCUs

Motor Control Library

User Manual

UM019903-0307

www.zilog.com

Motor Control Library
Z8FMC16100 Series

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, please visit www.zilog.com.

UM019903-0307

Revision History

Each instance in Revision History reflects a change to this document from its previous revision. For more details, refer to the corresponding pages and appropriate link in the table below.

Date	Revision Level	Section/Description	Page No
March 2007	03	Globally updated for MC Library v2.0.0	All
July 2006	02	Globally updated MC Library to MC Library v1.1	All
December 2005	01	Original Issue	All

Table of Contents

Introduction

This User Manual describes ZiLOG's Motor Control (MC) Library v2.0.0 for motor control application development on ZiLOG's Z8 Encore! MC™ based Z8FMC16100 Series of Flash MCUs. The Library is a Class B compliant software and available as a part of Z8FMC16100 Series of Flash MCUs.

This manual serves as a guide for interfacing the user application with the on-chip peripherals of the Z8FMC16100 Series of devices. The MC Library v2.0.0 is available for **Brushless DC (BLDC)** and **AC Induction Motor (ACIM)** motors.

Note: MC Library APIs must be used in accordance with default Library Files provided along with software. Any updates to the default library may lead to the changes in the functionality.

About this Manual

We recommend you to read all the chapters and instructions provided in this manual before loading and using the product. The manual is designed to be used as a reference guide for application development using MC Library v2.0.0 APIs.

Intended Audience

This document is for ZiLOG customers familiar with 3-phase BLDC motor control concept, vector control concept for 3-phase AC Induction Motor, microcontrollers, assembly language, and 'C' language.

Manual Organization

This user manual is divided into four chapters and one appendix. A brief description of each chapter and appendix is provided below.

Introduction

This chapter introduces the MC Library in terms of its intended audience, manual organization, references, and conventions.

Chapter 1—Motor Control Library

This chapter provides an overview of MC Library and its corresponding modules.

Chapter 2—Getting Started

This chapter provides the hardware and software requirements, and installation procedure.

Chapter 3—Sample Application for BLDC Motor

This chapter provides instructions for building and executing sample application for BLDC motor through ZDS II using hyperterminal application.

Chapter 4—Sample Application for AC Induction Motor

This chapter provides instructions for building and executing sample application for AC Induction motor through ZDS II using Panel Class B GUI.

Appendix—List of APIs

This appendix provides the list of MC Library v2.0.0 APIs.

Related Documents

In addition to this manual, you must be familiar with the following documents:

- Z8FMC16100 Series Product Specification (PS0246)
- Motor Control Library API Reference Manual (RM0046)
- Z8FMC16100 Series Motor Control Development Kit (UM0192)
- ZiLOG Developer Studio II Product Brief (PB0098)
- eZ8 CPU User Manual (UM0128)
- Vector Control Development Kit User Manual (UM0214)
- Automatic electrical controls for household and similar use—Part 1 (IEC 60730-1)
- Household and similar electrical appliances—Safety—Part 1 (IEC 60335-1)

Abbreviations and Acronyms

Table 1 lists the abbreviations and acronyms used in this manual.

Table 1. List of Abbreviations and Acronyms

Abbreviation	Expansion
MC	Motor Control
API	Application Programming Interface
BLDC Motor	Brushless Direct Current Motor
ACIM	AC Induction Motor
ADC	Analog-to-Digital Converter

Table 1. List of Abbreviations and Acronyms (Continued)

Abbreviation	Expansion
DAC	Digital-to-Analog Converter
ISR	Interrupt Service Routine
MDS	Modular Development System
ESD	Electrostatic Discharge
PLL	Phase Locked Loop
ZDS II	ZiLOG Developer Studio II
PWM	Pulse Width Modulator
LED	Light Emitting Diode
IEC	International Electrotechnical Commission

Manual Conventions

Table 2 lists the conventions used in this manual.

Table 2. Manual Conventions

Convention	Description
`Courier font`	Code lines and fragments are distinguished from general text by appearing in the Courier typeface. This convention is also used within tables.
Bold	Window names, button names, and menu commands are written in bold to distinguish from general text. For example, Press **OK**.

Table 2. Manual Conventions (Continued)

Convention	Description
Symbol	**Caution:** Indicates a procedure or file may become corrupted if you do not follow directions.
MC	Each API is preceded with 'MC' to identify it as a motor control API. For example, `MCConfigLib()`.

Safeguards

Follow the precautions listed below to avoid permanent damage to the evaluation board:

- Use a grounding strap to prevent damage resulting from electrostatic discharge (ESD).
- Power-up precautions:
  (a) Power-up the PC and ensure that it is running properly.
  (b) Start the terminal emulator program on the PC.
- Power-down precautions:
  (a) Quit the monitor program.
  (b) Remove power from the development platform.


ZiLOG

Chapter 1

Motor Control Library

This chapter describes the MC Library under the following topics:

Overview

The MC Library supports Brushless DC (BLDC) motor using **sensorless (back-EMF sensing) control** and AC Induction Motor (ACIM) using **Vector Control algorithm**. This Library allows motor control application development through a set of well-defined APIs and header files coded using ANSI ‘C’ language (see Appendix—List of APIs). You must call the appropriate API when building the sample application.

This software library detect faults occuring in the appliance and provides exception handling feature to prevent unintended operation of the appliance. This feature makes the Library a Class B compliant software. The MC Library APIs control the basic functions required by motor control application to interface with the on-chip peripherals on Z8FMC16100 Series of MCUs. For more details on MC Library APIs, refer to *Motor Control Library API Reference Manual (RM0046)*.

Features

The MC Library includes the following features:

- IEC 60730-1 Class B compliancy
- Speed and direction control
- Fault response
- Constant speed mode and constant torque mode of operation
- BLDC control mode utilizes back-EMF sensing Phase Locked Loop (PLL) approach
- ACIM utilizes Vector Control approach
- Communication through ZiLOG's Panel Class B GUI (only for ACIM)
- Utilizes integrated operational amplifier of Z8FMC16100 Series of MCUs
- The MC Library also provides APIs for the following:
  - I^2C communication with Digital-to-Analog Converter (DAC) for debugging (only for ACIM)
  - Retrieving voltage, current, and speed of the motor useful for remote monitoring and control

IEC 60730-1 Class B Compliant Software

The MC Library follows the International Electrotechnical Commission (IEC, www.iec.ch) Class B standards. Class B software detects fault in the appliance and provides exception handling to prevent any unintended operation in the appliance due to the fault. Thus, many household appliances (for example, microwave oven, washing machine etc.) make use of Class B compliant software.

The Class B compliancy of MC Library enables fault detection for following system components:

- CPU Register
- Program Counter
- Random Access Memory (RAM)
- Flash Memory
- Clock Frequency
- Interrupt
- Analog Input (for example, voltage, current, etc.)

The above components are tested periodically for faults. If a fault is detected, the motor is brought to a stop to prevent any unintended operation resulting from the fault.

Speed and Direction Control

The MC Library APIs configure the speed and direction of the motor. The monitoring APIs help to monitor, debug, and interface the motor control application with other applications. The following parameters are monitored:

- ADC values of DC bus voltage and bus current
- Temperature
- Set-speed
- Motor's direction of rotation

Fault Response

If a fault is detected, the library contains code for exception handling which asserts a specific flag and stops the motor.

Communication Through Panel Class B GUI

The AC Induction Motor Vector Control module receives commands through a Panel Class B GUI. Based on the input, the motor speed is controlled.

I^2C Communication with DAC

The MC Library APIs allows I^2C communication with DAC, which in turn can be used to measure the motor parameters on Cathode Ray Oscilloscope (CRO) or can be used for debugging

Constant Speed Mode and Constant Torque Mode of Operation

The library consists of Proportional Integral (PI) loop which operates by maintaining constant speed or constant current depending on the user input when configuring the library. This provides the flexibility to run the motor either in Torque mode or Speed mode. Only Speed Mode is implemented for AC Induction Motor.

BLDC Control Mode Utilizes Back-EMF Sensing PLL Approach

The MC Library utilizes the PLL based approach for locking to the back-EMF signal during startup phase. This eliminates the need for initial placement of the rotor in a specific position. It also reduces the jerky movement of the motor during startup or direction reversal.

AC Induction Motor Utilizes Vector Control Approach

Vector Control approach decouples the orthogonal component of current in ACIM into magnetizing current and torque producing current. Thus, each component can be controlled individually providing higher efficiency, fast response, and low-operating costs.

Utilizes Integrated Operational Amplifier of Z8FMC16100 MCU

The MC Library APIs utilize the integrated operational amplifier of Z8FMC16100 Series of MCUs which converts the current signal to a voltage signal resulting in a simpler hardware design.

3-phase Sensorless BLDC Motor Control

The MC Library is an effort to provide an off-the-shelf solution to the application developer. It consists of APIs for the 3-phase sensorless brushless direct current (BLDC) motor control applications. The APIs in this library controls the basic motor functions to interface with the on-chip peripherals and are used as building blocks for motor control application development.

3-phase AC Induction Motor Control

MC Library supports the control of a 3-phase AC Induction Motor using Vector Control algorithm. The APIs helps you to build a Vector Control application for AC Induction Motor, executable on an 8-bit microcontroller (Z8FMC16100 Series of MCUs). The Class B feature of the library ensures the detection and correction of all the system faults.

The Vector Control approach provides efficient and accurate control of the motor’s speed and torque. In library code, Z8FMC16100 microcontroller's PWM module is configured as three complementary output pairs with frequency set at 20 kHz, which results in PWM interrupt generated at a period of 50 µs. The PWM interrupt service routine (ISR) leaves the PWM interrupt disabled while it executes, and is tuned to execute within 200-250 µs. Therefore, the PWM ISR is executed every five PWM cycles or 250 µs.

The primary goal of the ISR is to update the duty cycle value for each PWM channel to produce the desired AC waveforms at the inverter bridge outputs. The PWM duty cycle values are derived from a feedback loop based on rotor speed and rotor flux position. The stator current

waveform is determined by sampling two phases of the inverter output current and reconstructing the third phase. The rotor speed is sampled by a counter/timer configured to measure the period of a magnetic position sensor on the motor. The measured speed is periodically compared to the requested speed received through UART from Panel Class B GUI.

The speed command is used to create the vector to produce the necessary flux and torque amplitude. Finally, the PWM registers are updated by values calculated using space vector modulation technique. For more details, refer *Vector Control of a 3-Phase AC Induction Motor Using FMC16100 MCU Application Note (AN0247)*.

Architecture

Figure 1 illustrates the MC Library v2.0.0 architecture.

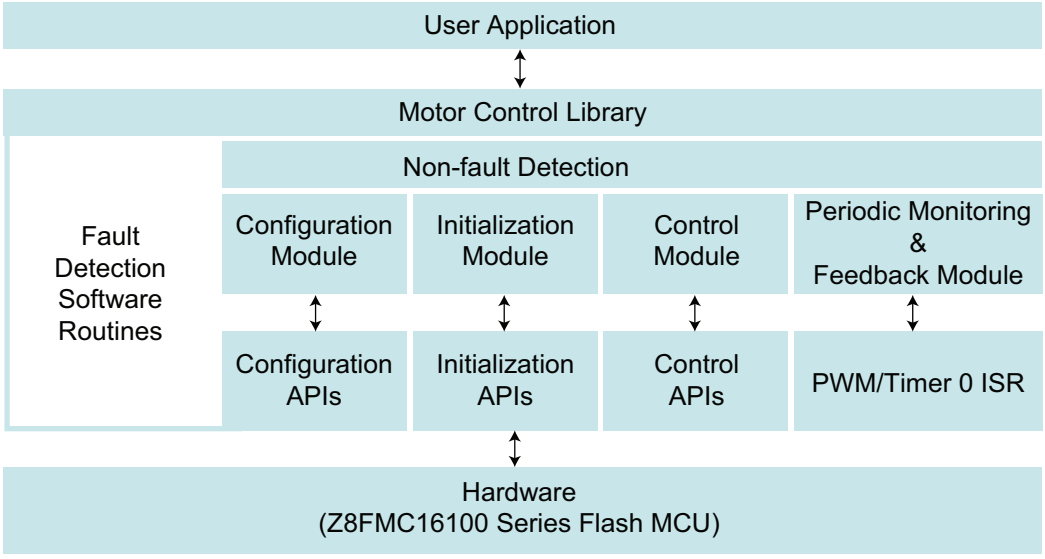

Figure 1. MC Library Architecture

MC Library is divided into the following segments based on their functionality:

- Fault Detection
- Non-Fault Detection

Fault Detection

The fault detection segment consists of software routines that checks for the hardware faults (for example, RAM and Flash faults) in the system. If a fault is detected, the motor is brought to a stop to prevent any unintended operation. As a part of fault detection, some tests are performed during startup while other tests are run periodically to detect the run-time faults. For more details on faults, theory of detection, and response to each fault, refer to the *Implementation of Class B Features White Paper (WP0016)* available for download at www.zilog.com.

Non-Fault Detection

The non-fault detection segment consists of the following major API modules:

- Configuration Module
- Initialization Module
- Control Module
- Periodic Monitoring and Feedback Module

Configuration Module

The Configuration Module allows you to set the default values for the on-chip peripherals associated with Z8FMC16100 Series of MCUs. It allows changes in the input parameters through enumerations used by the MC Library APIs.

The configuration module which consists of **`MCConfigLib()` API** loads all the General Purpose Input/Output (GPIO) ports with default values along with the motor operation mode selection and startup test.

This module also performs an initial test to detect any fault in RAM or CPU register, during startup. If a fault is detected, the motor is stopped to prevent any unintended operation.

Initialization Module

The Initialization Module consists of APIs to initialize the on-chip peripherals required to drive 3-phase sensorless BLDC motor or 3-phase ACIM configured with Z8FMC16100 Series of MCUs. Table 3 lists the initialization module APIs for BLDC and ACIM. The default values required for these initialization must be provided by the configuration module. The peripherals initialized by the initialization module are as follows:

- External Oscillator
- GPIO Ports
- Comparator
- Analog-to-Digital Converter (ADC)
- Timer0
- Pulse Width Modulator (PWM)
- Operational Amplifier

Table 3. Initialization Module APIs

BLDC Motor	AC Induction Motor
MCInitOsc	MCInitOsc
MCInitOpAmp	MCInitOpAmp
MCInitComparator	MCInitComparator
MCInitAdc	MCInitAdc
MCInitAdcPort	MCInitAdcPort
MCInitTimer0	MCInitTimer0
MCInitPwm	MCInitPwm
MCInitDirectionControl	MCInitI2C
	MCInitUart
	MCInitIdleCurrent
	MCInitRelay

Control Module

The Control Module APIs allows you to control the motor parameters like speed, voltage, current, temperature, and direction. Table 4 lists the Control Module APIs for BLDC and AC Induction Motor.

Table 4. Control Module APIs

BLDC Motor	AC Induction Motor
MCGetCurrent	MCGetCurrent
MCGetVoltage	MCGetVoltage
MCGetTemp	MCGetTemp
MCGetSpeed	MCGetSpeed
MCGetDirection	MCGetDirection
	MCSendTOGUI
	MCGetMotorSpeed

Periodic Monitoring and Feedback Module

Periodic Monitoring and Feedback Module is an internal interrupt driven module of the MC Library. This module constantly monitors the back-EMF, DC voltage supply to the motor, and commutation time. Based on the feedback, the PWM duty cycle is adjusted. It also controls the direction switching and speed of the motor.

Periodic monitoring and feedback module also periodically checks for occurance of any fault. If a fault is detected, the motor is brought to a stop to prevent any unintended operation resulting from the fault. Table 5 lists the Periodic Monitoring and Feedback APIs for BLDC and AC Induction Motor.

Table 5. Periodic Monitoring and Feedback Module APIs

BLDC Motor	AC Induction Motor
MCGetLastError	MCGetLastError
MCGetLastErrorLocation	MCGetLastErrorLocation
MCCheckInterrupt	MCCheckInterrupt
	MCSpeedRamp
	MCSpeedPIRegulator
	MCPeriodicModifiedChecksum
	MCPeriodicStaticRAMTest
	MCPeriodicStaticRegisterTest_R0R7
	MCPeriodicStaticRegisterTest_R8R15
	MCPeriodicAnalogCheck

Chapter 2

Getting Started

This chapter provides the detailed description of system requirements and installation under the following headings:

- System Requirements
- Downloading MC Library
- Installing MC Library
- Installing ZDS II—Z8 Encore!® v4.10.0 Software
- Installation and Settings

System Requirements

The hardware and software for ACIM and BLDC motors is as follows:

- BLDC Motor
- AC Induction Motor

BLDC Motor

The MC Library is tested with LINIX 45ZWN24-30 (3-phase BLDC) on the ZiLOGs standard development board. This section provides hardware and software requirements for executing the sample application using MC Library with 3-phase sensorless BLDC motor configured with Z8FMC16100 Series of MCUs.

Hardware

Hardware required for MC Library includes:

- ZiLOG's Z8FMC16100 modular development system (MDS) board
- ZiLOG's 3-phase motor control application board
- 24 V 3-phase BLDC motor (3200 rpm)
- 24 V and 5 V DC power supply
- Opto-Isolated USB Smart Cable

Software

Software required for MC Library includes:

- WinXP operating system
- ZiLOG Developer Studio (ZDS II), Z8 Encore! v4.10.0 or latest version
- HyperTerminal application
- 3-phase BLDC motor control source code

AC Induction Motor

MC Library is tested with BOSCH (3-phase ACIM) on ZiLOG's Vector Control Development Board. Following hardware and software are required for executing the sample application:

Hardware

Hardware required for MC Library includes:

- Z8FMC16100 MCU based Vector Control Development Board
- USB to TTL serial cable (TTL-232R)
- 3-phase AC Induction Motor with tachometer

- 220 V, 50 Hz or 110 V, 60 Hz or 50 V DC power supply
- Opto-Isolated USB Smart Cable

Software

Software required for MC Library includes:

- WinXP operating system
- ZiLOG Developer Studio (ZDS II), Z8 Encore! v4.10.0 or latest version
- 3-phase AC Induction Motor control source code

Downloading MC Library

The MC Library is available for download at www.zilog.com/application/library.asp.

Installing MC Library

Follow the steps below to install the MC Library:

1) Double-click `setup.exe` file to launch the **InstallShield Wizard**.
2) Click **Next**.
3) Accept the licence agreement on the **InstallShield Wizard** and click **Next**.
4) Enter the appropriate User Name and Organization and click **Next**.
5) Click **Finish** to complete the installation. On successful installation, MC Library (see Figure 2) is installed into the following default filepath:

   `C:\Program Files\ZiLOG\MC_2.0.0_SRC`

Installing ZDS II—Z8 Encore!® v4.10.0 Software

ZDS II v4.10.0 is available for free download at www.zilog.com. Follow the steps below to install the ZDSII–Z8 Encore!:

1) Unplug the power supply from the demo board.
2) Load the ZDSII CD into your CD-ROM drive.
3) The CD automatically launches the DemoShield application and provides a menu to install the product and documentation. Selecting **Install Products** followed by **Install ZDSII** displays the Installation Wizard.

The following directory is created on the host PC, assuming all installation settings remain at their defaults:

```
C:\Program Files\ZiLOG\ZDSII_Z8Encore!v4.10.0
```

Notes:
1) Software versions illustrated are for reference only. You may have an updated version.
2) You must install ZDSII–Z8 Encore! v4.10.0 software prior to installing MC Library.

MC Library Directory Structure

Figure 2 illustrates the hierarchy of the MC Library v2.0.0 directory structure.

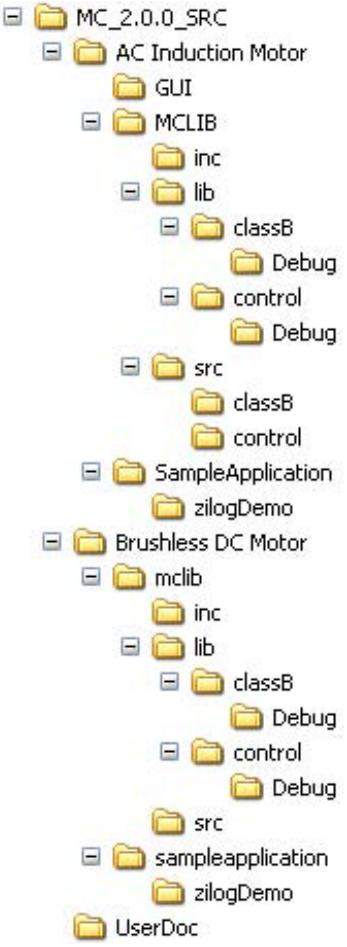

Figure 2. MC Library Directory Structure

Table 6 lists the folder contents of the MC Library directory.

Table 6. MC Library v2.0.0 Directory Contents

Folder	File	Description
AC Induction Motor\GUI	Executable file	Panel Class B GUI application.
AC Induction Motor\MCLIB \inc	Header file	AC Induction Motor header file and Class B header file that must be included in the project workspace.
AC Induction Motor\MCLIB \lib\classB\Debug	Library file	Class B library file that must be included in the application workspace.
AC Induction Motor\MCLIB \lib\control\Debug	Library file	AC Induction Motor library file that must be included in the application workspace.
AC Induction Motor\MCLIB \src\classB	Source file	Class B function source file.
AC Induction Motor\MCLIB \src\classB	Source file	AC Induction Motor source file.
AC Induction Motor\MCLIB \src\control	Sample project	Sample application project workspace that demonstrates a sample application using MC Library APIs. The header file and library files are already added along with the project.
AC Induction Motor\ SampleApplication \zilogDemo	Header file	BLDC motor header file and Class B header file that must be included in the project workspace.
Brushless DC Motor\mclib \inc	Library file	Class B Library file that must be included in the application workspace.

Table 6. MC Library v2.0.0 Directory Contents (Continued)

Folder	File	Description
`Brushless DC Motor\mclib \lib\classB\Debug`	Library file	BLDC motor Library file that must be included in the application workspace.
`Brushless DC Motor\mclib \src`	Source file	Class B function and BLDC motor source file.
`Brushless DC Motor\ SampleApplication\ zilogDemo`	Sample project	Sample application project workspace that demonstrates a sample application using MC Library APIs. The header file and library files are already added along with the project.
`UserDoc`		This directory contains MC Library related documents, including this document.

Installation and Settings

Following hardware and software setup are required for BLDC and AC Induction Motor:

- BLDC Motor
- AC Induction Motor

BLDC Motor

Hardware Setup

Required hardware connections are provided below:

1) Connect the 3-phase BLDC to voltage lines—A, B, and C of motor control application board.
2) Connect the Z8FMC16100 MDS board to the 3-phase motor control application board. For default settings, refer to *Z8FMC16100 Series Motor Control Development Kit User Manual (UM0192)*.

3) Connect the 5 V power supply to the Z8FMC16100 MDS board.
4) Connect the 24 V power supply to the motor control application board.
5) Connect the debug port of Z8FMC16100 MDS board to the PC using Opto-Isolated USB Smart Cable.

Configuring HyperTerminal Application

Follow the steps below to configure the HyperTerminal application:

1) Connect the Serial Port (DB-9) of Z8FMC16100 MDS Board to the COM1 or COM2 of the PC through a serial cable.
2) Launch the HyperTerminal application by navigating to **Start** → **Programs** → **Accessories** → **Communications** → **HyperTerminal**. The **Connection Description** dialog box appears (see Figure 3).

Figure 3. Connection Description Dialog Box

3) Enter the name for a new connection in the **Connection Description** dialog box and click **OK**. The **Connect To** dialog box appears (see Figure 4).

Figure 4. Connect To Dialog Box

4) In the **Connect Using** text field, select the port (COM1 or COM2) to which the serial cable is connected. Click **OK**. The **Port Settings** dialog box appears for the selected port (see Figure 5).

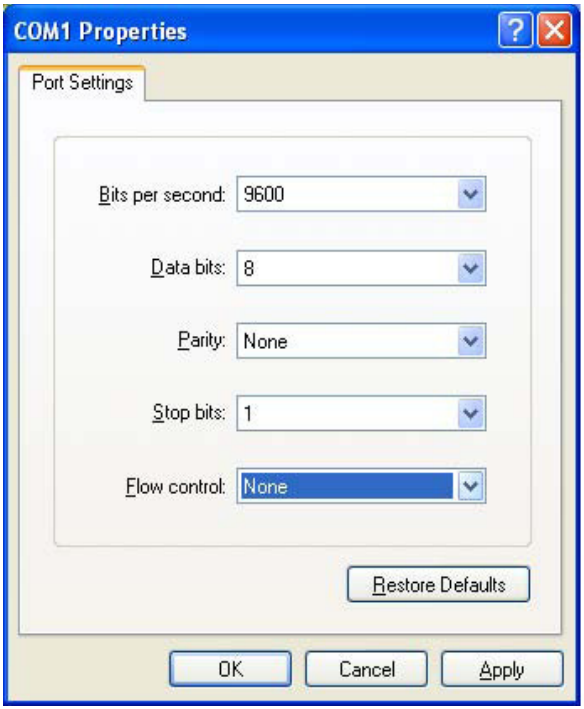

Figure 5. Port Settings Dialog Box

5) In the **Port Settings** dialog box, enter the following values in their respective fields.
   (a) Bits per second: 9600
   (b) Data bits: 8
   (c) Parity: None
   (d) Stop Bits: 1
   (e) Flow control: None
6) Click **OK**. The connection is established between Z8FMC16100 MDS Board and PC through serial port.

AC Induction Motor

Note: Apply 50 V DC power supply to the Vector Control Development Board if powering for the first time. During testing, run the board on 50 V DC power supply for safety purpose.

Hardware Setup (50 V DC Power Supply)

The connections for hardware setup are provided below:

1) Connect the 3-phase leads of ACIM to terminals PH-A, PH-B, and PH-C of Vector Control Development Board.
2) Connect the tachometer leads of ACIM to terminals T1 and T2 of Vector Control Development Board.
3) Connect the +VDC of 50 V DC supply to AC - 1 and Return to AC - 2.
4) Now follow Step 7 through Step 10 of Hardware Setup (AC Power Supply)

Hardware Setup (AC Power Supply)

The connections for hardware setup are provided below:

1) Connect the 3-phase leads of ACIM to terminals PH-A, PH-B, and PH-C of Vector Control Development Board.
2) Connect the tachometer leads of ACIM to terminals T1 and T2 of Vector Control Development Board.
3) Connect the frame of ACIM to FGND (Frame Ground) terminal of Vector Control Development Board.
4) If supply voltage is 110 V, 60 Hz; connect the Line of supply to AC-2 and neutral to C-T (see Figure 6).
5) If supply voltage is 220 V, 50 Hz; connect the Line of supply to AC-1 and neutral to AC-2 (see Figure 7).

6) Connect FGND to Earth ground of AC-supply.
7) Install the USB to TTL serial cable driver software (for installation details, check with the manufacturer for USB to TTL serial cable). Once you connect the USB to TTL Serial cable to PC, the cable appears as a new COM port as illustrated in Figure 14 on page 39.
8) Connect Vector Control Development Board to PC using USB to TTL serial cable (TTL-232R).
9) Connect the debug port of Z8FMC16100 MCU on Vector Control Development Board to PC through Opto-Isolated USB Smart Cable.
10) Connect the power supply to the Vector Control Development Board.

Figure 6 illustrates the hardware setup for AC Induction Motor for AC 110 V and 60 Hz supply voltage.

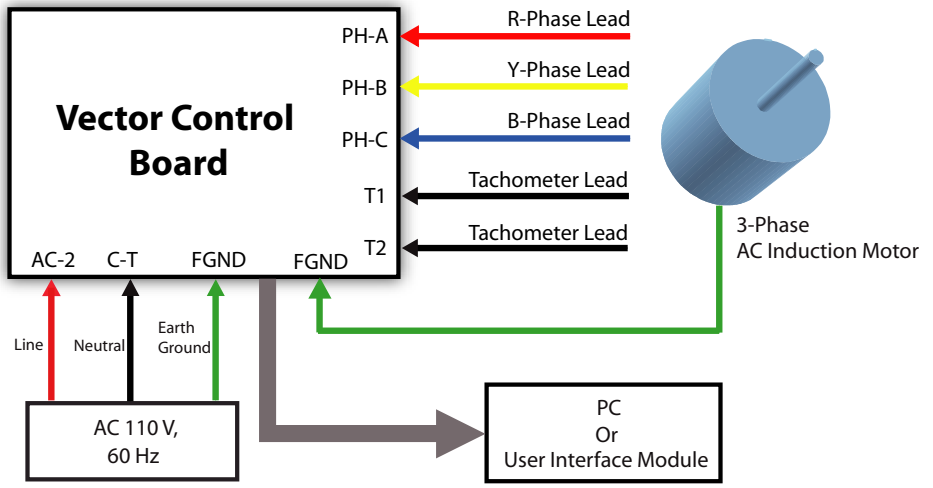

Figure 6. Hardware Setup for ACIM with 110 V, 60 Hz Supply Voltage

Figure 7 illustrates the hardware setup for AC Induction Motor for AC 220 V and 50 Hz supply voltage.

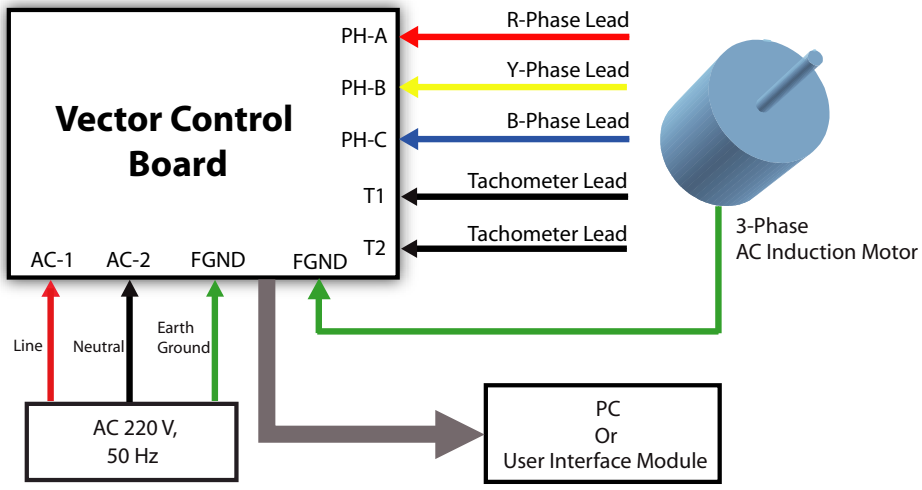

Figure 7. Hardware Setup for ACIM with 220 V, 50 Hz Supply Voltage

Chapter 3

Sample Application for BLDC Motor

This chapters describes the procedure to execute the sample application under the following topics:

- Downloading Demo Program to Board
- ZDS II Project Settings
- Running Sample Application

Downloading Demo Program to Board

A sample application written for Z8FMC16100 Series forms a part of MC Library. This section discusses the procedure to execute this sample application for Z8FMC16100 Series of MCUs only.

Follow the steps below to download the sample application to the Z8FMC16100 MDS Board:

1) Install the Opto-Isolated USB Smart Cable Driver Software. Refer installation guide of Opto-Isolated USB Smart Cable included in the Z8FMC16100KIT.
2) Connect the Opto-Isolated USB Smart Cable to the PC.
3) Connect the USB Smart Cable to the Z8FMC16100 MDS board debug port.
4) Powerup the Z8FMC16100 MDS board using 5 V power supply.
5) Launch ZDS II by navigating from the **Start** menu to **Programs** → **ZiLOG ZDS II-Z8 Encore! v4.10.0** → **ZDS II Z8 Encore! v4.10.0**.
6) Select **Open Project** from the **File** menu. The **Open Project** dialog box is displayed.

7) Browse to the **zilogDemo** folder for the **demoApp.zdsproj** file located in the below filepath:.

```
C:\Program Files\ZiLOG\MC_2.0.0_SRC\Brushless DC Motor\SampleApplication\zilogDemo
```

8) Open the **demoApp.zdsproj** project file. The initial ZDS II screen is displayed. To view the project source files, double-click the Project Files folder on left side of IDE interface. Double-click on the individual file to open in ZDS II file editor.

9) From the **Project** menu in ZDS II, select **Settings**. The **Project Settings** dialog box appears. Check for all the parameters (see Figure 8 and Figure 9) and do not change them. By default, ZDS II stores your files in the same directory as the project files:

```
C:\Program Files\ZiLOG\MC_2.0.0_SRC\Brushless DC Motor\SampleApplication\zilogDemo
```

10) Click **OK** to close the **Project Settings** dialog box.

11) Download the application code by selecting **Debug** → **Reset**. When the **IDE** Warning dialog box (see Figure 12) is displayed, click **Yes** to continue.

ZDS II Project Settings

This section provides the ZDS II project settings for BLDC and AC Induction Motor. Select **Settings** from the **Project Menu** to view the following default project settings:

- General Settings
- Linker Settings

General Settings

Figure 8 illustrates the General Settings in **Project Settings** dialog box.

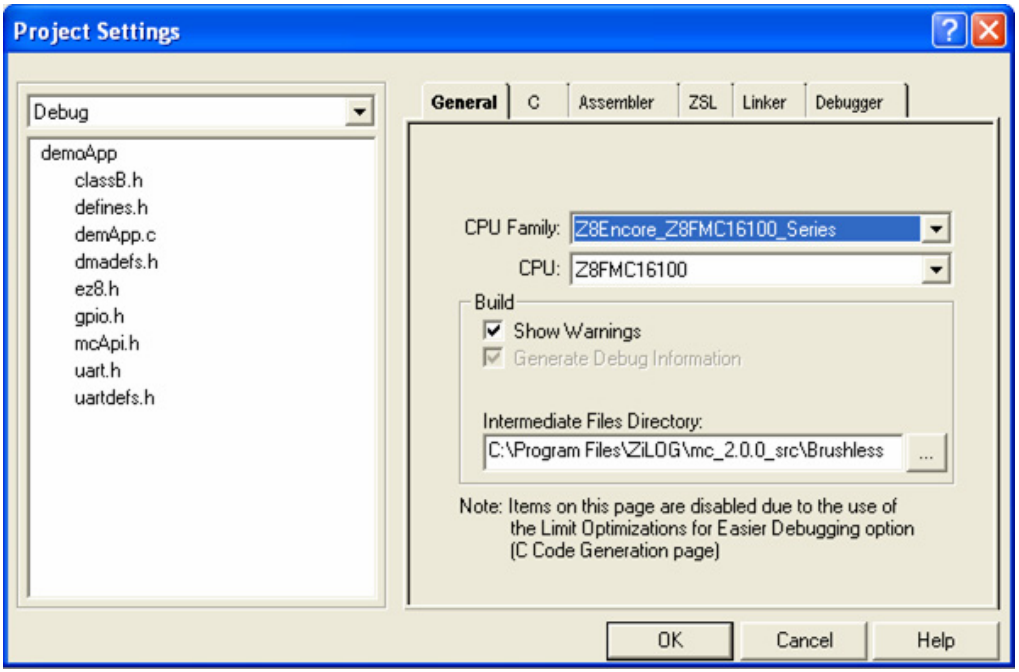

Figure 8. General Settings for Z8FMC16100 MCU in ZDS II

Figure 8 illustrates the General Settings for the following fields:

- **CPU Family**: Z8Encore_Z8FMC16100_Series
- **CPU**: Z8FMC16100
- **Intermediate Files Directory**: `C: \Program Files\ZiLOG \mc_2.0.0_src \Brushless DC Motor\sampleapplication \zilogDemo\Debug\`

You can change the path for the Intermediate Files Directory.

Linker Settings

Figure 9 illustrates the Linker Settings in **Project Settings** dialog box.

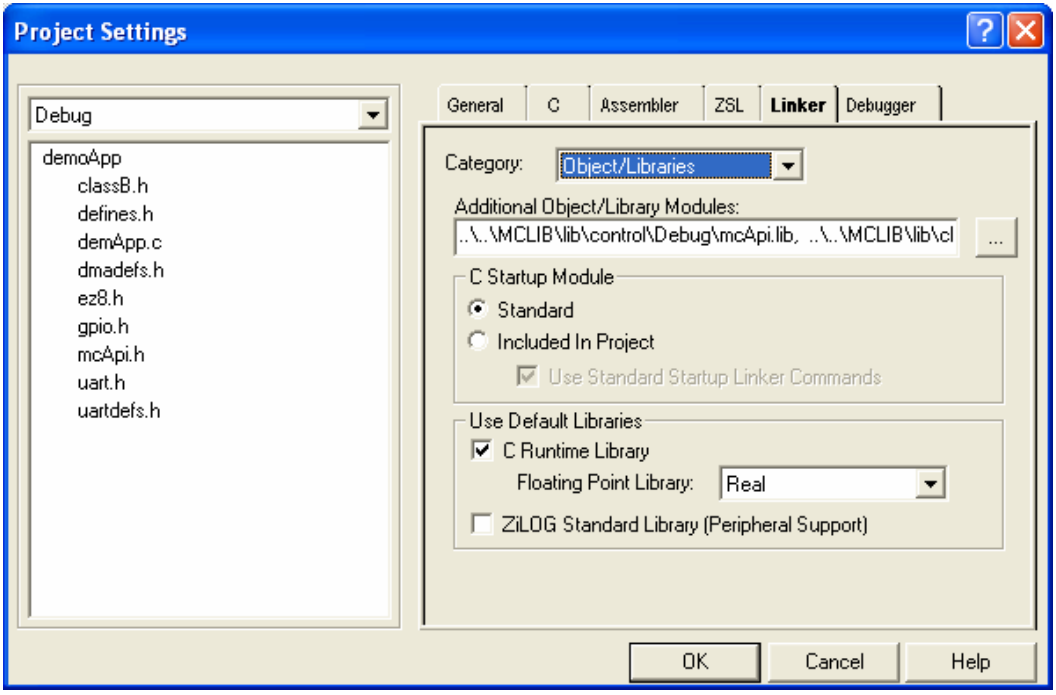

Figure 9. Linker Settings for Z8FMC16100 MCU in ZDS II

Running Sample Application

The sample application for BLDC motor can be tested in simulation mode or Z8FMC16100 MCU as the target.

Simulation Mode

To run the sample application in simulation mode, click the **Debugger** tab of the **Project Settings** dialog box and select the **Simulator** option from the Debug Tool drop-down list (see Figure 10).

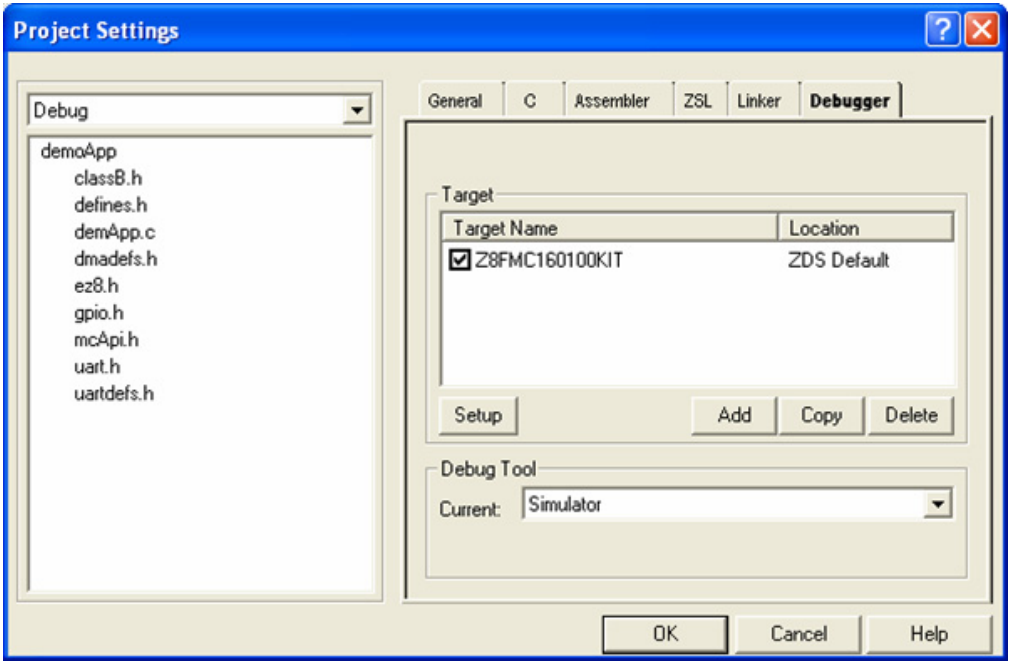

Figure 10. Simulation Mode Settings for Z8FMC16100 MCU in ZDS II

Z8FMC16100 MCU mode

To run the sample application in Z8FMC16100 MCU mode, click the **Debugger** tab of the **Project Settings** dialog box and select the **USB-SmartCable** from the Debug Tool drop-down list (see Figure 11).

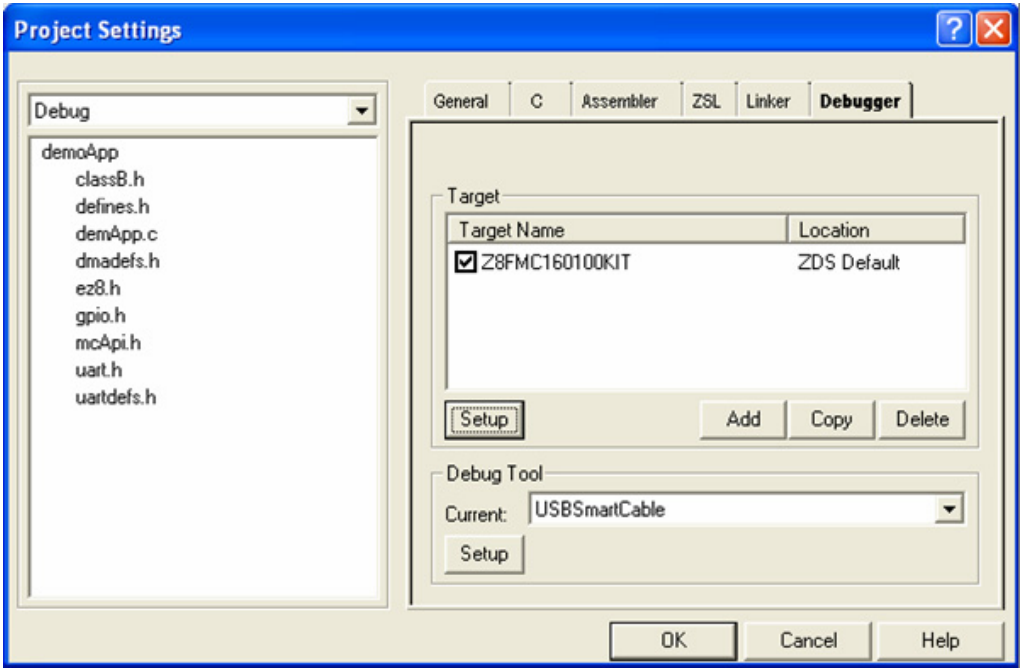

Figure 11. Z8FMC16100 MCU Mode Settings in ZDS II

Perform the following steps to run the sample application either in simulation mode or Z8FMC16100 MCU as the target:

1) Select **Build → Debug → Go**.
2) On successful execution of the sample application, loading message is displayed on ZDS II **Debug window**:

```
Loading IEEE-695 Absolute file
"C:\PROGRA~1\ZiLOG\MC_1.1.0_S\SAMPLE~1\ZILOGD~1\
demoApp.lod"
```

An output similar to following code block is displayed in the HyperTerminal application:

```
CURRENT = 00000000
****CURRENT VALUE ONLY FOR REFERENCE****
VOLTAGE = 00000024
SET SPEED = 00000050
TEMPERATURE = 00000000
DIRECTION = FORWARD
```

Chapter 4

Sample Application for AC Induction Motor

This section describes the procedure to execute the sample application for AC Induction Motor. It covers the following topics:

- Downloading Demo Program to Board
- Running Sample Application

Downloading Demo Program to Board

Follow the steps below to download the sample application to the Z8FMC16100 Vector Control Development Board:

1) Install the Opto-Isolated USB Smart Cable Driver Software. Refer installation guide of Opto-Isolated USB Smart Cable included in the Z8FMC16100KIT.
2) Connect the Opto-Isolated USB Smart Cable to the PC.
3) Connect the USB Smart Cable to the Z8FMC16100 Vector Control Development Board debug port.
4) Powerup the board with AC-supply.
5) Launch ZDS II by navigating from the **Start** menu to **Programs** → **ZiLOG ZDS II-Z8 Encore! v4.10.0** → **ZDS II Z8 Encore! v4.10.0**.
6) Select **Open Project** from the **File** menu. The **Open Project** dialog box is displayed.
7) Open the **classB.zdspro**j project file and **classB.h** header file.

8) To run the Vector Control Development Board with 50 V DC power supply, set the value for `SET_MIN_VOLTAGE` as `0x00`. To run the board with AC power supply, set the value for `SET_MIN_VOLTAGE` as `0xA1`.

9) Select **Build** → **Rebuild All**.

10) Browse to the **zilogDemo** folder for **demoApp.zdsproj** file located in the directory structure.

```
C:\Program Files\ZiLOG\mc_2.0_src\AC Induction
Motor\SampleAppplication\zilogDemo
```

11) Open the **demoApp.zdsproj** project file. The initial ZDS II screen is displayed. To view the project source files, double-click the **Project Files** folder on left side of IDE interface. Double-click on the individual file to open in ZDS II file editor.

12) From the **Project** menu in ZDS II, select **Settings**. The **Project Settings** dialog box appears. Check for all the parameters (see Figure 8 and Figure 9) and do not change them. By default, ZDS II stores your files in the same directory as the project files:

```
C:\Program Files\ZiLOG\mc_2.0_src\AC Induction
Motor\SampleAppplication\zilogDemo
```

13) Click **OK** to close the **Project Settings** dialog box.

14) Download the application code by selecting **Debug** → **Reset**. When the **IDE** Warning dialog box (see Figure 12) is displayed, click **Yes** to continue.

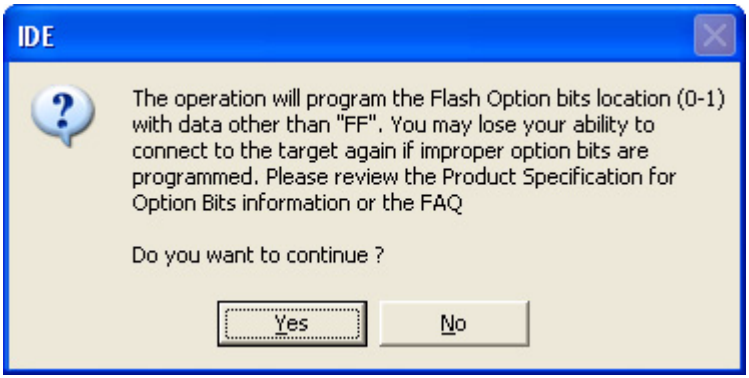

Figure 12. IDE Warning Dialog Box

Running Sample Application

This sections discusses configuring the Panel Class B GUI and the procedure to run the sample application for AC Induction Motor.

Follow the steps below to run the sample application:

1) Double click `Panel Class B.exe` to open the **Panel Class B GUI** application (see Figure 13) from the below path:

   `C:\Program Files\ZiLOG\mc_2.0_src\AC Induction Motor\GUI`

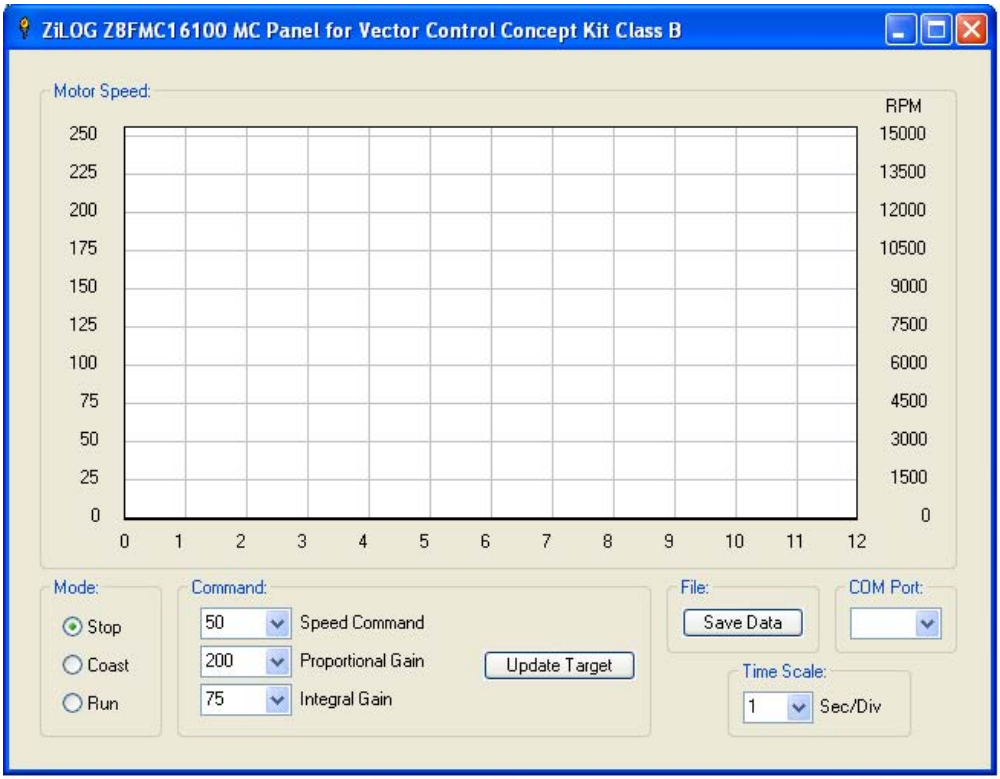

Figure 13. Panel Class B GUI Opening Screen

2) Click on the **Com Port** on Panel Class B GUI window. A drop-down list of available COM Ports is displayed. Select the appropriate COM port. You can determine the COM Port from the Device Manager window (see Figure 14) after installing the USB to TTL serial cable.

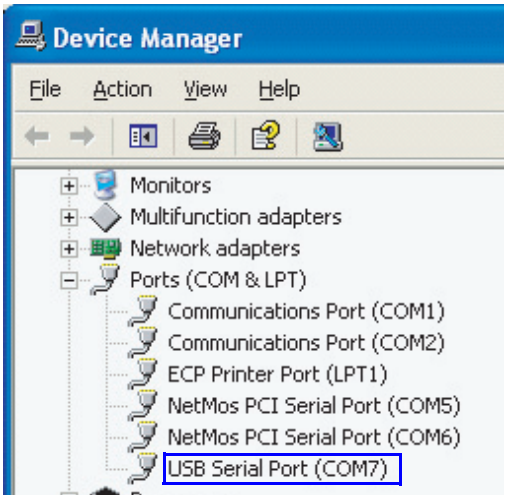

Figure 14. Device Manager Window

3) To set the Command:
   (a) Click on **Speed Command** and select the appropriate value from the drop-down list.
   (b) Click on **Proportional Gain** and select the appropriate value from the drop-down list
   (c) Click on **Integral Gain** and select the appropriate value from the drop-down list.
4) Click **Update Target**.
5) To run the motor, select radio button **Run**.
6) While the motor is running, to change Speed, Proportional Gain or Integral Gain, select the appropriate value from drop-down list and click on **Update Target**.
7) To stop the motor gradually, select the radio button **Coast**.

8) To stop the communication between the target board and Panel Class B GUI, select the radio button **Stop**.

9) To save the recent speed value on the GUI, click on **Save Data**. The data is saved in `.csv` format.

10) If the motor is stopped due to an external factor (for example, power cut, fault, etc.) repeat from Step 1 to reset and run the motor.

The output is displayed as waveform in the **Panel Class B GUI** as illustrated in Figure 15.

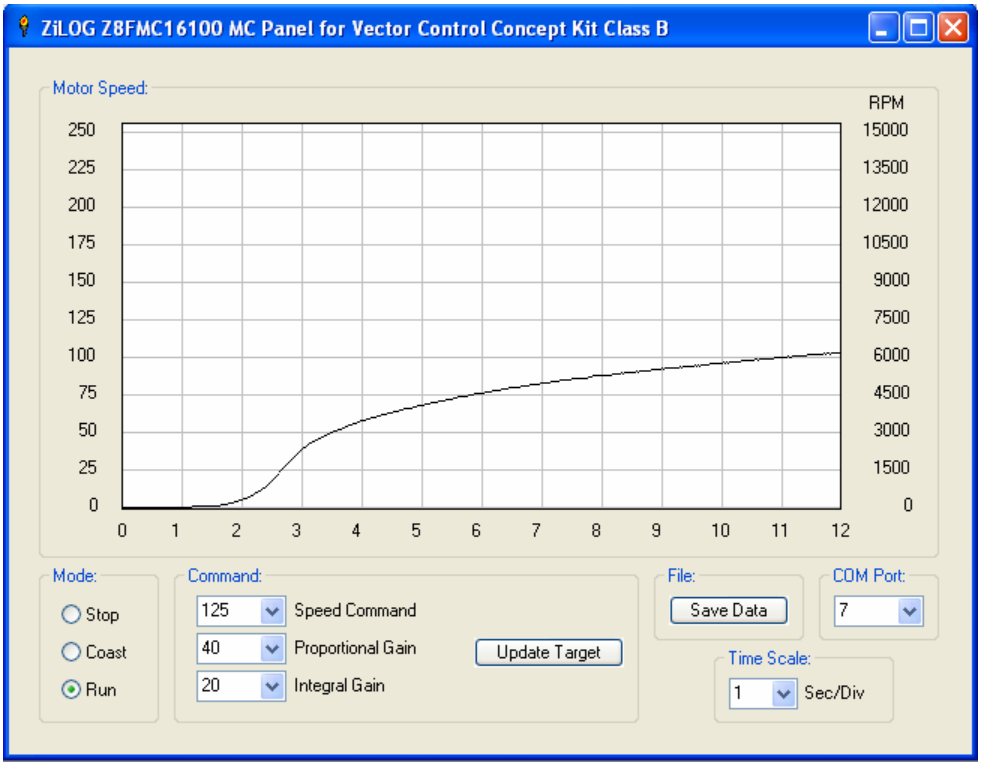

Figure 15. Panel Class B GUI Output Waveform

Appendix—List of APIs

The MC Library v2.0.0 assists you in building your own motor based applications. Table 7 provides a brief description of each API. For detailed description of each MC Library API, refer to *Motor Control Library API Reference Manual (RM0046)* available for download at www.zilog.com.

Table 7. MC Library APIs Quick Reference

API Name	Motor Type	Synopsis
`MCConfigLib`	BLDC/ACIM	This is the first API to be called before calling any other MC Library API. Configures the library with default values, sets the motor control operation mode and performs an initial test to detect any fault during startup.
`MCInitOsc`	BLDC/ACIM	Enables external oscillator.
`MCInitOpAmp`	BLDC/ACIM	Enables operational amplifier.
`MCInitComparator`	BLDC/ACIM	Enables comparator.
`MCInitAdc`	BLDC/ACIM	Enables ADC.
`MCInitRelay`	ACIM	Enables power supply to IGBTs after a delay of 3 seconds by turning on relay and then waiting for one second for relay to stabilize.
`MCInitIdleCurrent`	ACIM	Calculates the current present on the bus during idle state.
`MCInitUart`	ACIM	Sets the UART for the appropriate baud rate.
`MCInitI2C`	ACIM	Sets the I2C for the appropriate data rate.

Table 7. MC Library APIs Quick Reference (Continued)

API Name	Motor Type	Synopsis
`MCSendToGUI`	ACIM	Transmits the data to Panel Class B GUI through the UART.
`MCGetMotorSpeed`	ACIM	Calculates the motor speed from Timer-0 captured value and retrieves the motor speed in the range of 0 (0 - rpm) to 255 (15,000 - rpm).
`MCSpeedPIRegulator`	ACIM	Regulates the motor speed with respect to speed command by running a PI-loop.
`MCSpeedRamp`	ACIM	Maintains a constant acceleration and deceleration rate for the motor speed.
Class B Test APIs		
`MCPeriodicAnalogCheck`	ACIM	Checks for the analog parameters (for example, voltage, current, etc.) and stops the motor if any parameter exceeds the specified limit. These limits are set in `classB.h` header file.
`MCPeriodicStatic RegisterTest_R8R15`	ACIM	Performs static memory test on registers R8 to R15 and stops the motor if any register fails the static memory test.
`MCPeriodicStatic RegisterTest_R0R7`	ACIM	Performs static memory test on registers R0 to R7 and stops the motor if any register fails the static memory test.
`MCPeriodicStatic RAMTest`	ACIM	Performs static memory test on RAM (one memory location at a time) and stops the motor if any RAM location fails the static memory test.
`MCPeriodicModified Checksum`	ACIM	Calculates the checksum for the program memory (16-location at a time).
`MCInitAdcPort`	BLDC/ACIM	Initializes any ADC channel.

Table 7. MC Library APIs Quick Reference (Continued)

API Name	Motor Type	Synopsis
`MCInitTimer0`	BLDC/ACIM	**BLDC Motor**—Sets the flags to initial value and enables the16-bit reloadable Timer0 and Timer0 interrupt service routine (ISR). **AC Induction Motor** —Sets the timer-0 in capture mode for speed feedback.
`MCInitPwm`	BLDC/ACIM	Enables the Pulse Width Modulator and sets the PWM interrupt service routine.
`MCInitDirectionControl`	BLDC	Initializes the specified port pin as motor direction control pin.
`MCGetCurrent`	BLDC/ACIM	Retrieves ADC value of the motor driving current.
`MCGetVoltage`	BLDC/ACIM	Retrieves ADC value of the DC bus voltage.
`MCGetTemp`	BLDC/ACIM	Retrieves the value of temperature.
`MCGetSpeed`	BLDC/ACIM	Retrieves the ADC value of the speed which is set through potentiometer.
`MCGetDirection`	BLDC/ACIM	Retrieves direction of the motor.

Customer Feedback

For answers to technical questions about the product, documentation, or any other issues with ZiLOG's offerings, please visit ZiLOG's Knowledge Base at:

http://www.zilog.com/kb

For any comments, detail technical questions, or reporting problems, please visit ZiLOG's Technical Support at:

http://support.zilog.com