Prodigy™-PCI Motion Card

User's Guide

P M D

NOTICE

Warranty

PMD warrants performance of its products to the specifications applicable at the time of sale in accordance with PMD's standard warranty. Testing and other quality control techniques are utilized to the extent PMD deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Performance Motion Devices, Inc. (PMD) reserves the right to make changes to its products or to discontinue any product or service without notice, and advises customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

Safety Notice

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage. Products are not designed, authorized, or warranted to be suitable for use in life support devices or systems or other critical applications. Inclusion of PMD products in such applications is understood to be fully at the customer's risk.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent procedural hazards.

Disclaimer

PMD assumes no liability for applications assistance or customer product design. PMD does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of PMD covering or relating to any combination, machine, or process in which such products or services might be or are used. PMD's publication of information regarding any third party's products or services does not constitute PMD's approval, warranty or endorsement thereof.

Related Documents

Magellan Motion Processor User's Guide

> Complete description of the Magellan Motion Processor features and functions with detailed theory of its operation.

Magellan Motion Processor Programmer's Command Reference

> Descriptions of all Magellan Motion Processor commands, with coding syntax and examples, listed alphabetically for quick reference.

Pro-Motion User's Guide

> User's guide to Pro-Motion, the easy-to-use motion system development tool and performance optimizer. Pro-Motion is a sophisticated, easy-to-use program which allows all motion parameters to be set and/or viewed, and allows all features to be exercised.

Other Documents

Magellan Motion Processor Electrical Specifications

> Booklets containing physical and electrical characteristics, timing diagrams, pinouts, and pin descriptions of each series:
>
> MC58000 Series, for DC brush, brushless DC, Microstepping, and Pulse & Direction motion processors
>
> MC55000 Series, for Pulse & Direction motion processors

Magellan Motion Processor Developer's Kit Manual

> How to install and configure the DK58000 series and DK55000 series developer's kit PC board.

ION Digital Drive User's Manual

> How to install and configure ION Digital Drives.

Prodigy-PC/104 Motion Card User's Guide

> How to install and configure the Prodigy-PC/104 motion board.

Table of Contents

This page intentionally left blank.

List of Figures

This page intentionally left blank.

1. Installation

In This Chapter

▶ Prodigy-PCI Motion Card Types
▶ Prodigy-PCI Motion Card Motor Types
▶ Software
▶ Accessory Products
▶ Installation Sequence
▶ Required Hardware
▶ Software Installation
▶ Preparing the Board for Installation
▶ Connection Summary for Prodigy-PCI Motion Cards
▶ Applying Power
▶ First-Time System Verification

1.1 Prodigy-PCI Motion Card Types

The Prodigy-PCI Motion Cards are high-performance PCI-bus boards which provide motion control for DC brush, brushless DC, and step motors. These boards are based on PMD's Magellan Motion Processors, which perform motion command interpretation and many other real-time functions. The following product selector table shows the relationship between board part numbers (Prodigy-PCI Motion Card P/N), Magellan Motion Processor part numbers (Magellan P/N), the number of axes supported, and the type of motors supported. In the following table, "all motor types" refers to DC brush, brushless DC, microstepping, and step (pulse & direction) motors.

Prodigy-PCI Motion Card P/N	Magellan P/N	Number of Axes	Motor Type
PR9258120	MC58120	1	All motor types
PR9258220	MC58220	2	All motor types
PR9258320	MC58320	3	All motor types
PR9258420	MC58420	4	All motor types
PR9255120	MC55120	1	Step (pulse & direction) only
PR9255220	MC55220	2	Step (pulse & direction) only
PR9255320	MC55320	3	Step (pulse & direction) only
PR9255420	MC55420	4	Step (pulse & direction) only

1.2 Prodigy-PCI Motion Card Motor Types

DC brush: output is a single-phase motor command; either in PWM (pulse width modulated), or analog (± 10V) output format. They are intended to control DC brush motors, or brushless DC motors using an amplifier which performs commutation.

Brushless DC: provides multi-phase motor command signals, either in PWM (pulse width modulated), or analog (± 10V) output format, using Hall-based or sinusoidal commutation. The output is intended to interface with brushless DC amplifiers and motors.

Microstepping: output is multi-phase analog (± 10V) or PWM (pulse width modulation) waveforms. They are designed to control 2- or 3-phase step motors using amplifiers which accept this command format.

Pulse & direction: output is standard pulse & direction signals, and are intended to interface with amplifiers which accept that command format.

For complete information on motor output formats and other information, see the *Magellan Motion Processor User's Guide.*

1.3 Software

Three major software packages are provided with the Prodigy-PCI Motion Cards. Pro-Motion, an interactive Windows-based exerciser program, C-Motion, a C-language library, and VB-Motion, a set of Active-X objects. C-Motion and VB-Motion are designed to simplify the development of motion applications for Prodigy-PCI Motion Cards.

Pro-Motion is a sophisticated, easy-to-use exerciser program which allows all board parameters to be set and/or viewed, and allows all board features to be exercised. Pro-Motion features include:

- Motion oscilloscope graphically displays processor parameters in real-time
- Axis Wizard to automate axis setup and configuration
- Interactive DC brush and brushless DC tuning
- Project window for accessing board parameters
- Ability to save and load current settings
- Distance and time units conversion
- Motor-specific parameter setup
- Axis shuttle performs continuous back and forth motion between two positions

Pro-Motion is described in the *Pro-Motion User's Guide.*

C-Motion provides a convenient set of callable routines comprising all of the code required for controlling the Prodigy-PCI Motion Card. C-Motion includes the following features:

- Axis virtualization
- The ability to communicate to multiple Prodigy-PCI Motion Cards
- Can be easily linked to any C/C++ application

VB-Motion provides a complete set of methods and properties for developing applications in Visual Basic and other Active-X enabled programs. All the features of C-Motion are also available in VB-Motion.

C-Motion and VB-Motion are documented in the *Magellan Motion Processor Programmer's Command Reference.*

1.4 Accessory Products

The Prodigy-PCI Motion Cards can be enhanced with the addition of any or all of the following list of hardware accessory products:

Component Part Number	Description
Cable-1003	100 position, 3-foot long, shielded round cable with locking connectors to connect Prodigy-PCI Motion Card to IM-1000 interconnect module.
Cable-1006	100 position, 6-foot long, shielded round cable with locking connectors to connect Prodigy-PCI Motion Card to IM-1000 interconnect module.
IM-1000	Breakout interconnect module that provides convenient jack-screw type terminators for the 100-pin cable. Used with cable-1003 or cable-1006.
DC-1000	Parallel encoder input adaptor. This daughter board module allows parallel-word and other encoders which use the SSI interface format to be directly connected.
Cable-3003	Interface cables used with 68-pin Option Con connector. This connector is only used with the brushless DC and microstepping board versions.
Cable-4203	3-foot-long RS232 cable that connects to the Serial connector. This cable allows serial RS232 communication to the PCI board. Only used if serial port communication is required.

For information on ordering these accessory products, please contact your PMD representative.

1.5 Installation Sequence

1 Before using the board, the software must be installed. Installation of the software is described in Section 1.7, "Software Installation."

2 For a normal installation of a Prodigy-PCI Motion Card, you will need to configure the board for the specific motor hardware to which it will be connected. Configuration of the Prodigy-PCI Motion Cards is described in detail in Section 1.8, "Preparing the Board for Installation."

3 Next, connect the system's motors, encoders, amplifiers, and sensors to operate the motion hardware. A description of the available connections and options for the Prodigy-PCI Motion Card is found in Section 1.9, "Connection Summary for Prodigy-PCI Motion Cards."

4 Install the Prodigy card into an available PCI slot. Refer to the PC manufacturer's documentation or website for additional information regarding the PC's cover removal, etc.

5 If installing the DC-1000, see Section 4.2, "DC-1000 SSI Option Board."

6 Once this hardware configuration is complete, the final step to finish the installation is to perform a functional test of the finished system. This procedure is described in Section 1.11, "First-Time System Verification."

Once these steps have been accomplished, the installation is complete, and the board is ready for operation.

1.6 Required Hardware

To install a Prodigy-PCI Motion Card, the following hardware is required:

1 The recommended platform is an Intel (or compatible) processor, Pentium or better, one available PCI slot, 30 MB of available disk space, 32MB of available RAM, and a CD-ROM drive. The supported PC operating systems are Windows 9X/ME/NT/2000/XP.

2 One to four pulse and direction, PWM, or analog-input amplifiers. The type of amplifier depends on the type of motor being used.

3 One to four step, DC brush, or brushless DC motors. These motors may or may not provide encoder position feedback signals, depending on the type of motor being used. Encoder feedback is a requirement for DC brush and brushless DC motors; for step motors, it's an option.

4 Additional connectors as required to connect the Prodigy-PCI Motion Card to the amplifiers and the motors. See Section 3.3, "Connections Summary— Motor Amplifiers," for more information on setting up these connections.

1.7 Software Installation

Locate the CD-ROMs containing the Pro-Motion, C-Motion and VB-Motion software. These CDs contain software to exercise the Prodigy board, and source code enabling you to develop your own motion applications. All software applications are designed to work with Windows 9X/ME/NT/2000/XP.

If autorun is enabled, the installation process will begin when the CD-ROM is inserted. If autorun is not enabled, go to Start > Run…, and in the Open textbox, type D:\setup.exe, where D: is the drive letter of your computer's CD-ROM drive. Follow the on-screen prompts to complete the installation process. Upon completion of the installation process, the following components will be installed.

1 Pro-Motion – an application for communicating to and exercising the installed board. Refer to the *Pro-Motion User's Guide* for operating instructions.

2 C-Motion – source code which may be used for developing motion applications in C/C++ based on the Magellan Motion Processor.

3 VB-Motion - Active-X DLLs and example source code which may be used for developing motion applications in Visual Basic based on the Magellan Motion Processor.

4 PDF versions of the *Prodigy-PCI Motion Card User's Guide, Magellan Motion Processor Programmer's Command Reference,* and *Magellan Motion Processor User's Guide.* The Adobe Acrobat Reader is required for viewing these files. If the Adobe Acrobat Reader is not installed on your computer, it may be freely downloaded from http://www.adobe.com.

1.8 Preparing the Board for Installation

1.8.1 Resistor Pack Settings

The Prodigy-PCI Motion Card has minimal jumper settings; most settings are software configurable. To prepare the board for installation, the following user-specified resistor pack options should be checked.

Item	Setting	Description
Resistor packs RS1, RS2, RS3	Installed; this is the default setting of resistor packs RS1 - RS3.	If differential connections are being used, leave these resistor packs installed.
	Removed	If single-ended encoder connections are being used, remove the resistor packs.

The following diagram shows the location of the resistor packs RS1, RS2, RS3, along with other components such as connectors. The component side of the board is shown, with the PCI slot connector at the bottom. All component locations in this manual refer to this orientation.
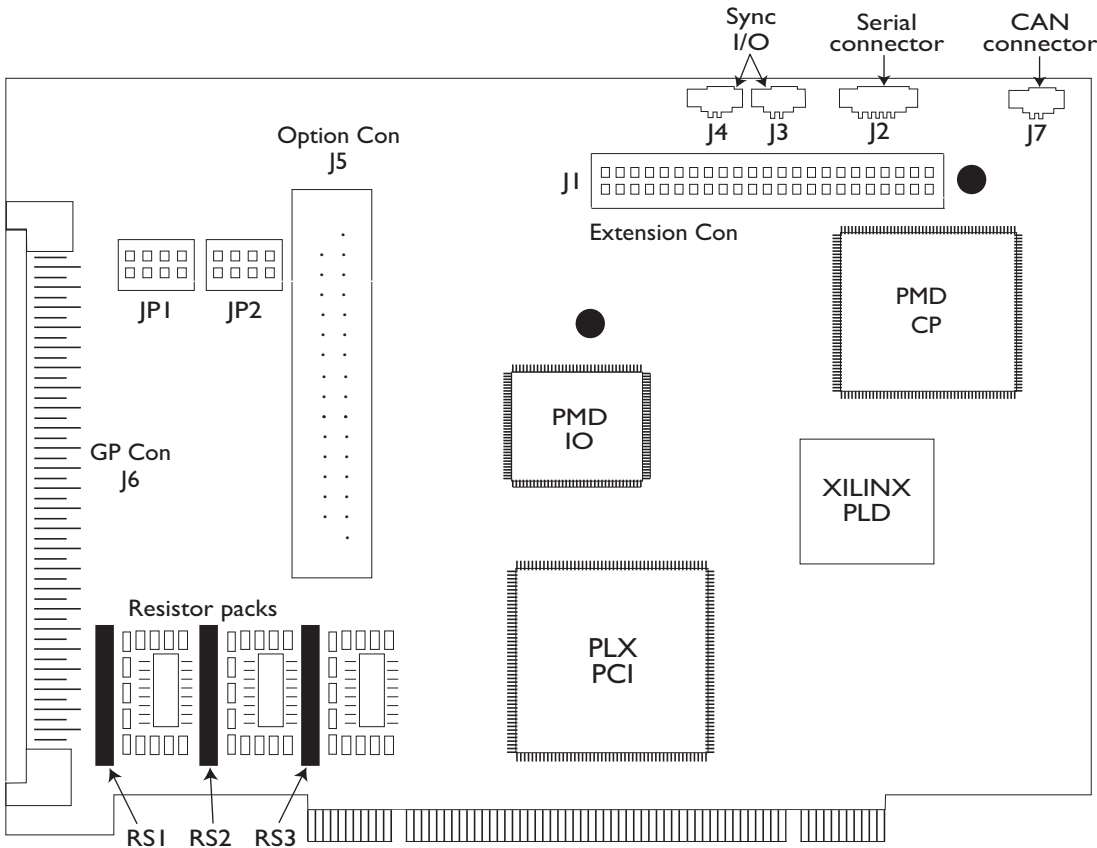
Figure 1-1: Prodigy-PCI Motion Card components

1.8.2 Motor Jumper Settings

Various possibilities exist for setting up the Prodigy-PCI Motion Card for use with either DC brush, brushless DC (with or without on-board commutation), microstepping, or step (pulse & direction) motors. The default mode at power-on is for DC brush motor on all axes.

The jumpers JP1 and JP2 consist of four jumper blocks each, relating to axis 1, 2, 3, or 4. The following table describes the correct jumper settings for each motor. See Figure 1-1 for JP1 and JP2 locations and jumper numbering. In the following table, the individual jumpers are numbered from left to right on each jumper block, each jumper number preceeded by its jumper block: JP1-1 is the jumper farthest to the left on JP1; JP1-2 is second from the left, and so on.

An example of open and closed jumpers is shown in Figure 1-2. Motor type selection determines the default output mode. See the *Magellan Motion Processor User's Guide* for more information.

Axis	Pulse & Direction	DC Brush, Brushless DC
1	JP1-1 installed	JP2-1 installed
2	JP1-2 installed	JP2-2 installed
3	JP1-3 installed	JP2-3 installed
4	JP1-4 installed	JP2-4 installed

NOTE: Caution should be used when setting motor jumpers. Be sure that the jumper settings are valid for the specified motor type.

**Figure 1-2:
Example of
pulse &
direction
jumper
settings**
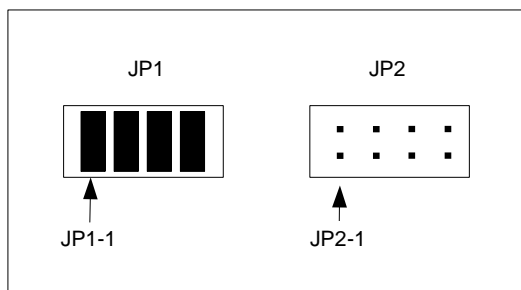
1.9 Connection Summary for Prodigy-PCI Motion Cards

The following sections summarize the recommended connections for various motor types. Generally, the motor type to be installed is specified by the type of Prodigy-PCI Motion Card (see table on page 9*)*. However, with the PR9258x20 boards, DC brush, brushless DC, microstepping, and step (pulse & direction) motors may be connected to the same board.

1.9.1 DC Brush Motors

The following table summarizes connections to the Prodigy-PCI Motion Card when DC brush motors are used. Between one and four axes may be connected depending on the specific Prodigy board and application requirements. All connections are made through connector GP Con, the primary 100-pin connector indicated in Figure 1-1. For a detailed list of connections, see Chapter 3, "Prodigy-PCI Electrical Reference."

Signal Category	Signal Description
Encoder input signals:*(per axis)*	A quadrature channel input
	B quadrature channel input
	Index pulse channel input
Amplifier output signals: *(per axis, if PWM sign, magnitude used)*	PWM direction
	PWM magnitude
Amplifier output signals: *(per axis, if PWM 50/50 used)*	PWM magnitude
Amplifier output signals: (per axis, if DAC output used)	DAC out
Other control signals: (optional per axis)	Home signal input
	Limit switch inputs
	AxisIn input
	AxisOut output
Miscellaneous signals:	Digital GND, AmpEnable,+5V (for encoder power)

1.9.2 Brushless DC Motors

The following table summarizes connections to the Prodigy-PCI Motion Card when brushless DC motors are used. Between one and four axes may be connected depending on the specific Prodigy board and application requirements. All connections are made through the connectors GP Con and Option Con, shown in Figure 1-1. For a detailed list of connections, see Chapter 3, "Prodigy-PCI Electrical Reference."

Signal Category	Signal Description
Encoder input signals:(per axis)	A quadrature channel input
	B quadrature channel input
	Index pulse channel input
Amplifier output signals: *(per axis, if PWM 50/50 used)*	PWM magnitude (phase A)
	PWM magnitude (phase B)
	PWM magnitude (phase C)
Amplifier output signals: (per axis, if DAC output used)	DAC out (phase A)
	DAC out (phase B)
Hall inputs:	Hall (phase A)
	Hall (phase B)
	Hall (phase C)
Other control signals: (optional per axis)	Home signal channel input
	Positive limit switch input
	Negative limit switch input
	AxisIn input
	AxisOut output
Miscellaneous signals:	Digital GND, AmpEnable,+5V (for encoder power)

1.9.3 Step Motors

The following table summarizes connections to the Prodigy-PCI Motion Card when pulse & direction interface step motors are used. Between one and four axes may be connected depending on the specific Prodigy board and application requirements. All connections can be made through connector GP Con, the primary 100-pin connector indicated in Figure 1-1. For a detailed list of connections, see Chapter 3, "Prodigy-PCI Electrical Reference."

Signal Category	Signal Description
Encoder input signals:*(per axis)*	A quadrature channel input
	B quadrature channel input
	Index pulse channel input
Amplifier output signals:	Pulse
	Direction
Other control signals: (optional per axis)	AtRest signal output
	Home signal channel input
	Positive limit switch input
	Negative limit switch input
	AxisIn input
	AxisOut output
Miscellaneous signals:	Digital GND, AmpEnable, +5V (for encoder power)

As an additional feature, differential pulse and direction outputs using line drivers are available on the Option Con connector. Using these signals with differential input pulse and direction step drivers will provide a higher immunity to noise. See Chapter 3, "Prodigy-PCI Electrical Reference," for more information on these differential signals.

1.9.4 Microstepping Motors

The following table summarizes connections to the Prodigy-PCI Motion Card when microstepping-interface step motors are used. Between one and four axes may be connected depending on the specific Prodigy board and application requirements. All connections are made through the connectors GP Con and Option Con, shown in Figure 1-1. For a detailed list of connections, see Chapter 3, "Prodigy-PCI Electrical Reference."

Signal Category	Signal Description
Encoder input signals:*(per axis)*	A quadrature channel input
	B quadrature channel input
	Index pulse channel input
Amplifier output signals:	PWM magnitude (phase A)
(per axis, if PWM sign, magnitude used)	PWM magnitude (phase B)
	PWM direction (phase A)
	PWM direction (phase B)
Amplifier output signals:	PWM magnitude (phase A)
(per axis, if PWM 50/50 used)	PWM magnitude (phase B)
Amplifier output signals:	DAC out (phase A)
(per axis, if DAC output used)	DAC out (phase B)
Other control signals:	Home signal channel input
(optional per axis)	Positive limit switch inputs
	Negative limit switch inputs
	AxisIn input
	AxisOut output
Miscellaneous signals:	Digital GND, AmpEnable,+5V (for encoder power)

1.10 Applying Power

Once you have installed the Prodigy-PCI Motion Card in your PC and have made the necessary connections to your external amplifiers and motor encoders, hardware installation is complete, and the board is ready for operation. Upon power-up, the board will be in a reset condition. In this condition, no motor output will be applied. Therefore, the motors should remain stationary. If the motors do move or jump, power down the board and check the amplifier and encoder connections. If anomalous behavior is still observed, call PMD for application assistance. Complete PMD contact information is listed on the last page of this manual.

If the PR9258x20 is being used, there may be a situation where undesired motion may occur when the processor is powered up. In particular, if the connected amplifier is a PWM 50/50 amplifier, the motor will receive 100% power because the PR9258x20 Prodigy-PCI Motion Card defaults to PWM sign/magnitude on reset. To avoid this situation, use AmpEnable as an enable/disable signal for the amplifier, and set the motor type for each axis before enabling the amplifier. See Section 2.3.2, "Amplifier Enable."

1.11 First-Time System Verification

The best way to setup and configure the Prodigy-PCI motion system is to run the Axis Wizard in Pro-Motion. The Axis Wizard performs a step-by-step software configuration and verification of each function for each axis. Refer to the *Pro-Motion User's Guide* for complete information on the Pro-Motion applicaton and its Axis Wizard. You may also want to refer to the *Magellan Motion Processor User's Guide* to familiarize yourself with operation of the board's motion processor.

The following sections are provided as a summary for users who are developing applications using C-Motion or VB-Motion. They show the minimum set of steps required to get an axis up and running. The Axis Wizard in Pro-Motion performs a more exhaustive list of software configuration steps incuding limit switch configuration, index capture, etc.

Since the configuration steps differ depending on motor type, the following table provides an overview listing each step by number. These steps are then detailed in subsequent sections. Perform only the setup step sequences for the board/motor type installed in your system.

Motor Type	Step #	Operation
DC brush	1	Set motor type
	2	Set output mode
	5	Set filter parameters
	6	Check encoder direction
	8	Make a trajectory move
Brushless DC	1	Set motor type
	2	Set output mode
	3	Initialize commutation
	4	Check commutation
	5	Set filter parameters
	6	Check encoder direction
	8	Make a trajectory move
Microstepping	1	Set motor type
	2	Set output mode
	7	Set the Motor Command
	8	Make a trajectory move
Step (pulse & direction)	1	Set motor type
	8	Make a trajectory move

It is assumed that each axis of your system will be verified one at a time. The Steps below are for Axis 1. To verify other axes, use a new axis number and verify each axis accordingly.

1.11.1 Step 1: Set the Motor Type

The Prodigy-PCI Motion Card must be told which type of motor will be driven by which axis. To do this, use the command:

    SetMotorType Axis1, type   // Sets the motor type for axis 1

The operation sets the number of phases for commutation on the axis, as well as internally configuring the motion IC for the motor type. See the *Magellan Motion Processor Programmer's Command Reference* for additional information.

The following table lists the number of phases to be commutated.

Motor Type	Value	Commutation
Brushless DC	0	3 phase
Brushless DC	1	2 phase
Microstepping	2	3 phase
Microstepping	3	2 phase
Step (pulse & direction)	4	None
DC brush	7	None

1.11.2 Step 2: Set the Output Mode

The Prodigy-PCI Motion Card must be told what type of motor output mode to use: PWM Sign Magnitude, PWM 50/50, or parallel DAC. This can be set using the command:

    SetOutputMode Axis1, mode   // Sets the output mode for axis 1

See the *Magellan Motion Processor Programmer's Command Reference* for additional information.

1.11.3 Step 3: Initialize the Commutation

NOTE: This section applies to brushless DC motors only.

For the motor to be properly controlled, the Prodigy-PCI Motion Card must select and possibly initialize the commutation phasing. If Hall-based commutation will be used, then no initialization is necessary. Simply specify this to the Prodigy-PCI Motion Card using the command:

    SetCommutationMode Axis1, 1   // Sets the commutation mode to Hall for Axis 1

No other commands are necessary, and you may proceed to step 4.

If commutating using a sinusoidal technique, the commutation phasing must be initialized. There are two ways this can be done; either by using Hall-based or by algorithmic methods. See the *Magellan Motion Processor User's Guide* for more information. Each of these two-phase initialization methods requires a separate sequence, as demonstrated in the following examples:

Hall-based initialization command sequence:

    SetPhaseCounts Axis1, yyyy   // yyyy is # of encoder counts per electrical cycle
    SetPhaseInitializeMode, Axis1, 1   // sets phase initialize mode = 1 (Hall-based) for Axs 1
    InitializePhase Axis1   // Initiates the phase initialization sequence for Axis 1

Algorithmic-based initialization command sequence:

    SetPhaseCounts Axis1, yyyy   // yyyy is # of encoder counts per electrical cycle
    SetPhaseInitializeMode Axis1, 0   // sets phase initialize mode = 0 (Algorithmic)
    SetOperatingMode, Axis1, 3   // disables the position loop, required for algorithmic initialization.
    SetPhaseInitializeTime Axis1, zzzz   // zzzz is # of motion processor cycles to initialize for
    SetMotorCommand Axis1, wwww   // wwww is motor command
    InitializePhase Axis1   // Initiates the phase initialization sequence for Axis 1

To determine the values of yyyy, zzzz, and wwww, please refer to the *Magellan Motion Processor User's Guide and the Magellan Motion Processor Programmer's Command Reference.*

If your system has one or more of the following conditions present:

    1   one or more Hall signals must be inverted to commutate or initialize the commutation correctly; or:

    2   the number of encoder counts per electrical cycle exceeds 32,767

then the preceeding command sequences will need to be expanded. For these systems, you will need to use the SetSignalSense command, along with the SetPhasePrescaleMode command. Refer to the commutation section of the *Magellan Motion Processor User's Guide* for more information.

> When the operating mode is set to 3, the motor is not under closed-loop position control. Be aware that the motor may spin rapidly after a motor command value is applied. To begin, use small values, and then gradually increase the values.

> Contact PMD for additional information on the Phase Initialization application note.

1.11.4 Step 4: Check Commutation

NOTE: This section applies to brushless DC motors only.

After phase initialization has been completed, we recommend checking the smoothness of the motor rotation in open loop mode, to verify that the motor phasing initialization and commutation is correct. For this verification, use the following command sequence:

```
SetOperatingMode Axis1, 3          // mode = 3 disables the positon loop of Axis 1
SetMotorCommand Axis1, xxxx        // xxxx is the motor command
Update Axis1                       // updates the buffered parameters and starts the move
```

The motor command value represents the fraction of 32,767 (where 32,767 is equivalent to total power) which will be applied to the motor. For example, a value of 1000 sends roughly 3% (1000/32,767) of the total power to the motor. After executing this command sequence, the motor should spin smoothly in the specified direction. The motor command is a signed number, and the sign controls the rotation direction. When a positive motor command is given, the motor should rotate in the positive (increasing encoder counts) direction. If the motor spins either roughly, in the wrong direction, or if it moves a short distance and then stops abruptly, there may be a problem with the commutation. Check the wiring and re-test. Once the motor is spinning smoothly in both directions under open-loop control, re-enable closed-loop DC brush and brushless DC control by executing the command:

```
SetOperatingMode Axis1, 0x33       // mode = 0x33 enables the trajectory generator, position loop and motor
                                   // output
```

1.11.5 Step 5: Set Filter Parameters

For motion to occur, some amount of feedback gain must be specified. Initially, use a proportional gain with a very low value between 1 and 25. Later, integral or derivative gains may be added, as well as feedforward gains if desired. The following sequence shows how to set the P, I, and D terms of the filter and how to update them, thus making them active.

```
SetPositionLoop Axis1, Param0, xxxx    // Param0 selects Kp and xxxx is the desired proportional gain
SetPositionLoop Axis1, Param3, yyyy    // Param3 selects Kd and yyyy is the desired derivative gain
SetPositionLoop Axis1, Param1, zzzz    // Param1 selects Ki and zzzz is the desired integral gain
SetPositionLoop Axis1, Param2, aaaa    // Param2 selects Ilimit and aaaa is the desired integration limit value
Update Axis1                           // make the values active
```

It is not necessary to specify all three gains. Just Kp, followed by an **Update** command can be specified; just a Kd, followed by **Update**, etc.

> When exercising the motor, use extreme caution. It is the responsibility of the user to observe safety precautions at all times.

1.11.6 Step 6: Check Encoder Direction

NOTE: This Section applies to DC brush and brushless DC motors only.

The next step is to ensure that the sense of direction between the PID output and the encoder is synchronized. A positive PID output should induce a direction of torque in the motor corresponding to positive encoder incrementation. In other words, a positive motor torque (which creates motion), should cause a positive encoder displacement.

This condition can be tested immediately after completing Step 5. This is because the system will respond to a position error once the filter parameters have been set. If the motor "runs away" after setting the filter parameters, then the encoder direction does not correspond to the motor torque direction. If the motor remains stationary, manually rotate the motor to induce a position error. If the motor runs away, then the encoder direction does not correspond to the motor torque direction. The simple solution for this is to use the C-Motion command **SetSignalSense** 0x0001 to invert the direction of the encoder. If the motor does not run away, but remains "servoed," then no action is necessary.

1.11.7 Step 7: Set the Motor Command

NOTE: this section applies to microstepping motors only.

In order for motion to occur, the magnitude of the output must be set. Refer to the *Magellan Motion Processor User's Guide* for more information. A value between 0 and 32,767 represents an amplitude of 0 – 100%. As a starting point, a value of around 5000 should be satisfactory.

Here is the command sequence to use:

```
SetMotorCommand Axis1, xxxx          // sets the motor output level for Axis 1
Update Axis1                         // execute the move
```

1.11.8 Step 8: Make a Trajectory Move

To verify that the motor is being controlled properly, set up and execute a short trapezoidal move. For example, a short distance of 5000 counts, a low velocity of 10,000, and acceleration of 10. With a cycle time of 256μsec, these values correspond to roughly 596 counts/sec, and 2.4×10^{-3} counts/sec^2, respectively.

The command sequence to use is as follows:

```
SetProfileMode Axis1, 0              // sets the profile mode to trapezoidal for Axis 1
SetPosition Axis1, 5000             // 5000 is the desired destination position
SetVelocity, Axis1, 10000          // 10,000 is the desired maximum velocity
SetAcceleration Axis1,10            // 10 is the desired acceleration
SetDeceleration Axis1,10            // 10 is the desired deceleration
Update Axis1                        // execute the move
```

After entering this sequence of commands, the axis should move smoothly for about 15 seconds (if the suggested values are used and the cycle time of the Prodigy-PCI Motion Card is 256μsec). If the axis is not moving, or if the axis jumps rapidly in one direction or the other, there may be a problem with the motor connections or software settings. Re-check and review the board setup procedures, as well as the exerciser parameter settings. If problems still exist after re-checking the system, contact PMD for applications assistance. PMD contact info is located on the final page of this manual.

> Whichever profile values you use, be sure that they are safe for your system.

2. Operation

In This Chapter

▶ Board Function Overview
▶ Magellan Motion Processor Functions
▶ Board-Specific Functions
▶ Signal Processing and Hardware Functions

The Prodigy-PCI Motion Cards are high-performance PCI-bus boards which provide motion control for DC brush, brushless DC, step, and microstepping motors. These boards are based on Magellan Motion Processors, which perform motion command interpretation and other real-time functions. The overall board function is divided among a number of modules, as illustrated in the following diagram.
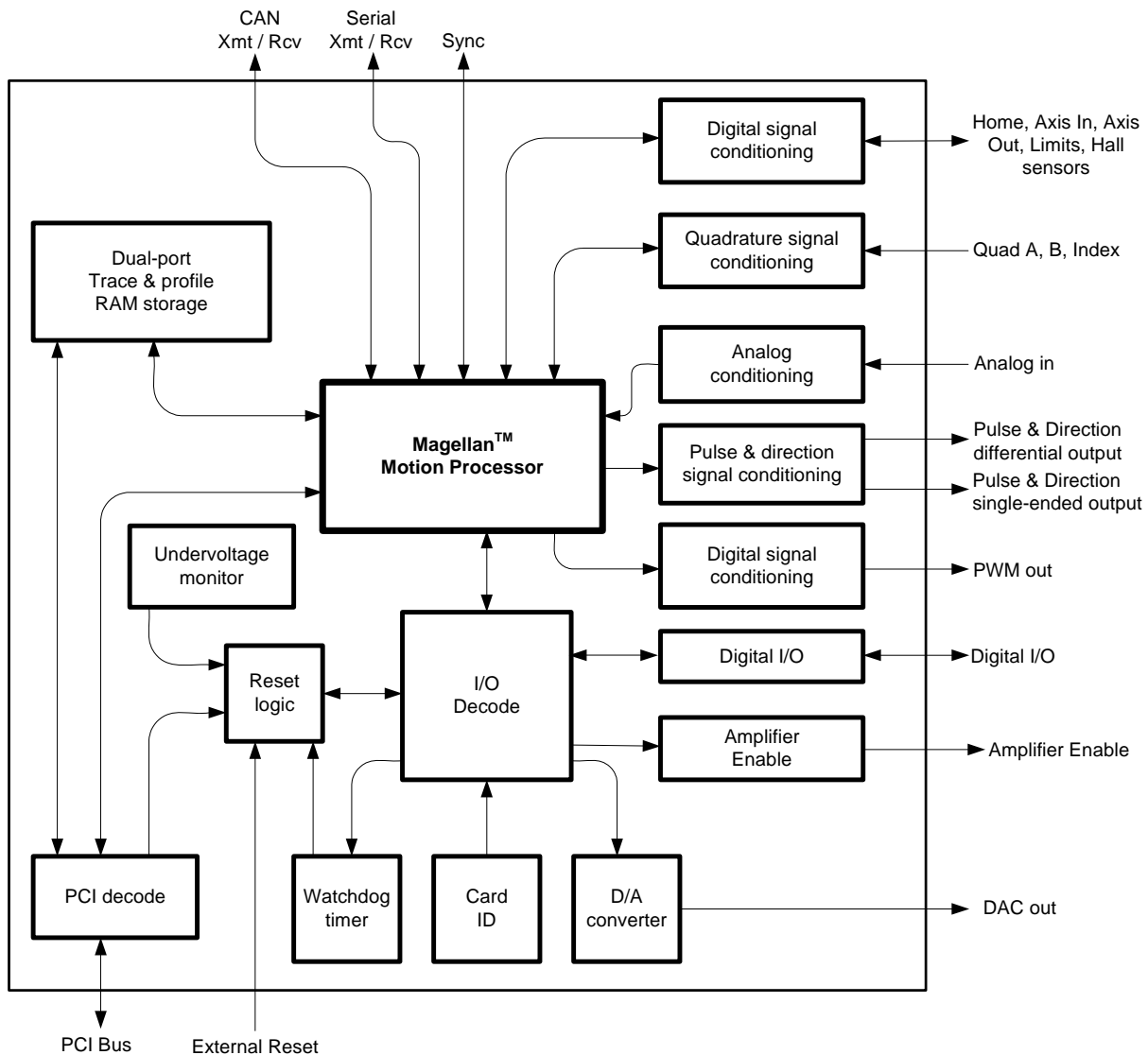

Figure 2-1: Prodigy-PCI internal block diagram

2.1 Board Function Overview

The Prodigy-PCI Motion Card's resources can be broken down into three overall categories:

Magellan Motion Processor functions - These are programmable functions which reside in the Magellan Motion Processor chipset. Included are profile generation, DC brush and brushless DC loop closure, microstep generation and much more. These functions are accessed through the Magellan API commands, which allows for sophisticated control of the board's overall behavior.

Board-specific functions - These are programmable functions which are controlled by the Magellan Motion Processor using the ReadIO and WriteIO commands, but which reside in various portions of the board circuitry. These functions include general purpose digital I/O, and other board-specific capabilities.

Signal processing & hardware functions - A substantial portion of the board provides signal conditioning and other functions associated with non-programmable, signal-related processing.

2.2 Magellan Motion Processor Functions

The Magellan Motion Processor block pictured in Figure 2-1 consists of two ICs: a CP (command processor) chip, and an IO (input/output) chip. A summary list of the functions provided by the Magellan Motion Processor is as follows:

- Profile generation
- Motor output signal generation (PWM and analog)
- Quadrature encoder counting and index capture
- DC brush and brushless DC loop closure
- Breakpoint processing
- AxisIn and AxisOut signal processing
- Trace
- Motion error detection, tracking windows, and at-settled indicator
- Limit switches

Access to the Magellan Motion Processor on the Prodigy-PCI Motion Card occurs through the PCI bus, the serial port, or the CAN port. Two manuals describe how the Magellan Motion Processor operates and how it is programmed: the *Magellan Motion Processor User's Guide,* and the *Magellan Motion Processor Programmer's Command Reference.*

The Magellan instruction set is very flexible and powerful. The following example, which would be used to set up and execute a simple trapezoidal profile, illustrates just a small part of the overall command set.

```
SetProfileMode Axis1, trapezoidal    // set profile mode to trapezoidal for axis 1
SetPosition Axis1, 12345             // load a destination position for axis 1
SetVelocity Axis1, 223344            // load a velocity for axis 1
SetAcceleration Axis1, 1000          // load an acceleration for axis 1
SetDeceleration Axis1, 2000          // load a deceleration for axis 1
SetUpdateMask Axis1, Profile         // specify that an update of profile parameters only
                                     // is to occur
Update Axis1                         // Double buffered registers are copied into
                                     // the active registers, thereby initiating the move
```

2.3 Board-Specific Functions

Board-specific functions are those functions which are mapped through the Prodigy-PCI Motion Card's **ReadIO** and **WriteIO** facility, but are implemented in the board circuitry.

Board-specific functions are detailed in this document, rather than the *Magellan Motion Processor User's Guide* or the *Magellan Motion Processor Programmer's Command Reference*.

2.3.1 General-Purpose Digital I/O

In addition to numerous special-purpose digital signals which are input or output to the board such as *AxisIn, AxisOut, Home, QuadA,* etc., the Prodigy-PCI Motion Cards support eight general-purpose inputs, and eight general-purpose outputs. These signals provide a convenient way of accessing additional general-purpose digital I/O. Although access to these signals occurs through the Prodigy-PCI Motion Card's **ReadIO** and **WriteIO** commands, the signals present at these various connections do not directly affect the Prodigy-PCI Motion Card's behavior; the Prodigy-PCI Motion Card simply passes them through.

> The general purpose digital I/O voltage range is 0-5V.

ReadIO and WriteIO commands

The eight inputs and eight outputs are read using the **ReadIO** command and **WriteIO** command, with an I/O address of 0. This is illustrated in the following table, along with the bit locations of the input and output signals.

I/O Address	Bit Location	Signals
0	0 - 7	DigitalOut0-7
	8 - 15	DigitalIn0-7

To read the eight general-purpose digital I/Os, a **ReadIO** command is performed at address offset 0. The 16-bit read word returns the current output values (set using the **WriteIO** command) in bits 0 - 7, while bits 8 - 15 hold the digital values corresponding to the signal levels at the connector for those inputs. To write new signal values to the eight digital outputs, a **WriteIO** command to address offset 0 is sent, and the values on bits 0-7 will be output to the signal connections. The value of bits 8 - 15 are ignored.

Example

To write the value 0xAA to bits 0 - 7, the command **WriteIO** is used. Assuming that the signal pattern 0x55 is present on the eight input connections, then if the command **ReadIO** is used, the value returned will be 0x55AA. The upper eight bits reflect the present value of the input signals, while the lower eight bits reflect the 8-bit value being output.

Related commands

In addition to the low-level **ReadIO** and **WriteIO** commands, the following utility commands are also supported in C-Motion and VB-Motion: **WriteDigitalOutput**, **ReadDigitalInput**, and **ReadDigitalOutput**. These utility commands provide a simpler interface by handling the high byte - low byte shifting. See the *Magellan Motion Processor Programmer's Command Reference* for more information.

Connections & associated signals

The general-purpose I/O are direct digital inputs and outputs. There are no associated connections required for these signals to function properly, however, one or more of the digital grounds must be connected. Digital inputs are pulled up through 4.7 kOhm resistors to 5V. The power-up default value for all general-purpose digital outputs is low.

For a complete description of the pinout connections to and from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

2.3.2 Amplifier Enable

The signals *AmpEnable1-4* provide four digital outputs which may be used as amplifier enable signals. They can also be used as general-purpose digital outputs. Similar to the general-purpose digital inputs and outputs (see Section 2.3.1, "General-Purpose Digital I/O"), these signals are not directly affected by the Prodigy-PCI Motion Card's behavior; however, they can be read or written through the Prodigy-PCI Motion Card's ReadIO and WriteIO commands.

ReadIO and WriteIO commands

These outputs are read using the ReadIO command, and written to using the WriteIO command, using an address of 1, as shown in the following table:

I/O Address	Bit Location	Signals
1	0-3	Amplifier enable outputs (0-3)
	4-6	Unused
	7	DAC enable status (1 = enabled; 0 = disabled)
	8-11	Change mask for bits 0-3; amplifier enable outputs (1 = change; 0 = don't change)
	12-14	Unused
	15	Change mask for DAC enable (1 = change; 0 = don't change)

To read the status of the amplifier enable outputs, the command ReadIO is used at address 1. The values currently being output will appear in bits 0 - 3. To write values to the amplifier enable output signals, the WriteIO command is used with an address of 1. The change mask bits corresponding to the signals which will be changed must be loaded at bits 8 - 11, and the value(s) to be loaded must be loaded in bits 0 - 3.

Examples

```
WriteIO Address1, 0x0505        // write 0x0505 to I/O Address 1 to enable Amplifiers 1 & 3
WriteIO Address1, 0x0400        // write 0x0400 to I/O Address 1 to disable Amplifier 3
WriteIO Address1, 0x0100        // write 0x0100 to I/O Address 1 to disable Amplifier 1
```

Related commands

In addition to the low-level ReadIO and WriteIO commands, the following utility commands are also supported in C-Motion and VB-Motion: SetAmplifierEnable and GetAmplifierEnable. These utility commands provide a simpler interface by handling the bit shifting. See the *Magellan Motion Processor Programmer's Command Reference* for more information.

Connections & associated signals

AmpEnable1-4 are direct digital outputs. There are no associated connections required for these signals to function prop-erly, however, one or more of the digital grounds must be connected. The power-up default value for all amplifier enable signals is low (disabled). For a complete description of the pinout connections from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

2.3.3 DAC Output Enable

In addition to the amplifier enable outputs, there is a dedicated board function which allows the DAC output signals to be shunted to 0 volts for safety purposes (DAC disabled), or allowed to be set by the Magellan Motion Processor (DAC enabled). This shunting occurs at a hardware level outside the motion processor itself, and provides an additional safety layer to control the motor command.

ReadIO and WriteIO commands

The status of the DAC output enable function can be read using the **ReadIO** command, and the DAC output enable status can be set using the **WriteIO** command, with an address of, as shown in the following table:

I/O Address	Bit Location	Signals
1	0-3	Amplifier enable outputs (0-3)
	4-6	Unused
	7	DAC enable status (1 = enabled; 0 = disabled)
	8-11	Change mask for bits 0-3; amplifier enable outputs (1 = change; 0 = don't change)
	12-14	Unused
	15	Change mask for DAC enable (1 = change; 0 = don't change)

To read the status of the DAC output enable function, **ReadIO** is used. The value currently in use will appear in bit 7. A value of 1 indicates DAC output is enabled, meaning that the voltage being output by the DACs is controlled by the motion processor. A value of 0 indicates that it is disabled, meaning that the voltage being output by the DAC is forced to 0.0 volts.

To enable or disable the DAC enable function, the **WriteIO** command is used. The change mask bit located at bit 15 must be loaded with a 1. Bit 8 must be loaded with a value of 0 to disable, or a value of 1 to enable output.

The powerup default value for DAC Output Enable is disabled. In addition, the DAC Output Enable is disabled upon a hard reset, which is a reset via the PCI bus, or via the external *Reset* signal. See Section 2.3.8, "Reset," for more information.

Examples

        WriteIO Address1, 0x8080   // write 0x8080 to I/O Address 1 to enable all DACs
        WriteIO Address1, 0x8000   // write 0x8000 to I/O Address 1 to disable all DACs

Related commands

In addition to the low-level **ReadIO** and **WriteIO** commands, the following utility commands are also supported in C-Motion and VB-Motion: **SetDACOutputEnable** and **GetDACOutputEnable**. These utility commands provide a simpler interface by handling the bit shifting. See the *Magellan Motion Processor Programmer's Command Reference* for more information.

Connections & associated signals

DAC output enable is an internal function of the board. Thus there are no signals directly associated with this function.

2.3.4 Serial Communications

The Prodigy-PCI Motion Card offers an asynchronous serial communications port. All functions of the card can be commanded through this port or it can be used to monitor operations, even while the PCI bus is used to command motion sequences.

The serial port can be operated at various baud rates from 1,200 to 460,800 and in varying configurations of stop, start, and parity codes. In addition, two serial protocols are supported: point-to-point, and multi-drop (idle line) mode. The Magellan commands **SetSerialPortMode** and **GetSerialPortMode** are used to set and read the serial port communication parameters. A complete description of the serial port, and its usage modes are provided in the *Magellan Motion Processor User's Guide.* A complete description of Magellan commands can be found in the *Magellan Motion Processor Programmer's Command Reference.* Also see Section 3.2.7, "Serial Connector."

After a reset or at power-up, the board sets the communication parameters to 57,600 baud, no parity, one stop bit, and point-to-point mode. To alter these parameters, new serial port parameters should be set using the **SetSerialPortMode** command, while communicating at the default parameters, and then switch to the new communications parameters.

Connections & associated signals

A special 5-pin connector is used to connect to the serial port. A convenient cable with this connector installed may also be ordered. See Section 1.4, "Accessory Products," and Section 3.2.7, "Serial Connector," for a detailed signal description of the serial connector.

2.3.5 CAN Communications

The Prodigy-PCI Motion Card offers a CAN port. All functions of the card can be commanded through this port or it can be used to monitor certain operations, even while the PCI bus is used to command motion sequences.

The CAN port may be operated at various baud rates from 1,200 to 1,000,000. The Magellan commands **SetCANMode** and **GetCANMode** are used to set CAN communication parameters. A complete description of the CAN connector and its usage modes are provided in the *Magellan Motion Processor User's Guide.* A complete description of Magellan commands can be found in the *Magellan Motion Processor Programmer's Command Reference.*

After a reset or at power-up, the board sets the CAN default parameters to 20,000 bps, and node ID to 0. To alter these parameters, the user may set new CAN parameters using the **SetCanMode** command, while communicating at the default parameters, and then switch to the new communications parameters.

The Prodigy-PCI Motion Card will integrate with a CAN 2.0B network. It will coexist, but not communicate with, other CAN 2.0B nodes on that network. The CAN functionality receives commands, sends responses, and (optional) sends asynchronous event notifications.

Connections & associated signals

A special 2-pin connector is used to connect to the CAN port. See Section 3.2.9, "CAN Connector," for a detailed signal description of the CAN connector.

2.3.6 Watchdog Timer

To enhance the overall safety of the board, a watchdog function has been included. The watchdog will automatically trigger a hard reset if communication from the host should be lost. Resetting the Prodigy-PCI Motion Card will have the result of setting all motor command outputs to zero, thus allowing the motors to come to a safe stop.

ReadIO and WriteIO commands

To enable the watchdog timer, the **WriteIO** command is used to send a value of 0x5562 to address 4. Once enabled, the watchdog timer will time out, causing a hard reset, if another write to address 4 with a value of 0x5562 is not received within 104 milliseconds. As long as a watchdog value is written to address 4 within the 104-millisecond interval, no reset will occur and motion operations will proceed normally.

After powerup or any hard reset, if no command is sent to the watchdog address, then the watchdog will remain disabled. The watchdog is disabled by default at power-up. When the watchdog timer times out and triggers a hard reset, it also disables itself.

Related commands

In addition to the low-level **WriteIO** command, the following utility command is also supported by C-Motion and VB-Motion: **SetWatchDog**. This utility command provides a simpler interface by automatically sending the value 0x5562. See the *Magellan Motion Processor Programmer's Command Reference* for more information.

Connections & associated signals

The watchdog timer is an internal function of the board. There are no signals directly associated with this function.

2.3.7 Undervoltage Monitor

To enhance reliability under a variety of electrical conditions, an undervoltage detection circuit has been included. This circuit triggers a hard reset when the voltage has dropped to an unsafe level. Resetting the Prodigy-PCI Motion Card will have the result of setting all motor command outputs to zero, thus allowing the motors to come to a safe stop. An undervoltage condition is detected when the 3.3V internal supply on the board drops below 95% of its rated value. To determine if a reset was caused by an undervoltage condition, see Section 2.3.9, "Reset Monitor."

Connections & associated signals

The undervoltage detector is an internal function of the board. There are no signals directly associated with this function.

2.3.8 Reset

Although a reset occurs automatically during power-up, it is sometimes desirable to reset the Prodigy-PCI Motion Card explicitly through a user-initiated command or action. There are several methods by which the Prodigy-PCI Motion Card may be reset. The methods are summarized in the following table:

Method	Type of Reset	Description
External reset	Hard reset	When the external signal *Reset* on GP Con is brought low, a "hard" reset occurs, which resets all of the board's circuitry and the Magellan Motion Processor.
Reset through the Magellan Motion Processor	Soft reset	Sending the **Reset** command to the Prodigy-PCI Motion Card causes a "soft" reset of the Magellan Motion Processor only.
Reset through PCI bus	Hard reset	The **HardReset** command uses the PCI bus to perform a "hard" reset, which resets all of the board's circuitry and the Magellan Motion Processor.

After a reset occurs, the Prodigy-PCI Motion Card and other related output signals will be driven to known states, depending on the type of reset performed. These are summarized in the following table:

Signal Name	Soft Reset	Hard Reset
AxisOut1-4	High	High
PWMMag1A-4C	Low	Low
PWMSign1A-4B	Low	Low
DAC1A-DAC4B	No change	0.0 volts
DigitalOut0-7	No change	Low
AmpEnable1-4	No change	Low
DAC On/Off	No change	Off
Watchdog Timer	No change	Disabled

Related commands

A HardReset utility command is provided in C-Motion and VB-Motion. This utility command uses the PCI interface to force a Hard reset of the entire board. See the *Magellan Motion Processor Programmer's Command Reference* for more information on this command.

Connections & associated signals

The reset feature has an external signal input, *Reset*, associated with it. This active low signal is located on the GP Con. It is pulled up through a 4.7 kOhm resistor to 5 V.

2.3.9 Reset Monitor

In addition to resets which are explicitly requested by the user (as detailed in the previous section), some reset conditions may occur automatically. During normal operations, the Prodigy-PCI Motion Card is only reset during power-up. A reset serves the purpose of initializing values and bringing the Prodigy-PCI Motion Card to a known and consistent state. On occasion, the Prodigy-PCI Motion Card will be reset due to one of a few anomalous conditions.

ReadIO and WriteIO commands

To determine the cause of a board reset, special instructions to read the reset source have been provided. The command ReadIO with an address of 2 should be used. The following table details the encoding of this I/O address word.

I/O Address	Bit Location	Signals
2	0-11	*Reserved*
	12	Hard Reset: a 1 value in this bit indicates a reset commanded over the bus.
	13	Undervoltage detection: a 1 value in this bit indicates a reset caused by undervoltage detection.
	14	External signal: a 1 value in this bit indicates a reset caused by the external *Reset* signal.
	15	Watchdog timeout: a 1 value in this bit indicates a reset caused by watchdog timeout.

Once a reset condition has occurred, the reset status stored at address 2 (described in the preceeding table) can be cleared by a WriteIO command to address 2 with a value of zero (0).

Example

To determine that a reset has occurred, and to determine the cause of the reset, the command ReadIO is used. Assuming that a watchdog timer event has occurred, the value returned would be 0x8000. To reset the reset monitor word, the command WriteIO is sent to address 2 with a value of zero (0).

Related commands

In addition to the low-level ReadIO and WriteIO commands, a GetResetCause utility command is also supported by C-Motion and VB-Motion. This utility command returns the cause and also clears the reset condition. See the *Magellan Motion Processor Programmer's Command Reference* for more information.

Connections & associated signals

The Reset monitor is an internal function of the board. There are no signals directly associated with this function.

2.3.10 Card ID

This feature allows the user to query the card for a Card ID. This may be helpful for verifying the type of Prodigy Motion Card in situations where multiple cards of varying types are installed.

ReadIO and WriteIO Commands

To read the Card ID, the ReadIO command is used with an address of 0xFF. The encoding of the bits returned is detailed in the following table:

Address	Bit Location	Signals
0xFF	0-3	Major board revision: This nibble encodes the major board revision. This value can range from 0 to 15.
	4-7	Minor board revision: This nibble encodes the minor board revision. This value can range from 0 to 15.
	8-11	Board generation: This nibble encodes the board generation. This value can range from 8 to 15. (0 to 7 are reserved for older motion board families.)
	12-15	Board type: This nibble encodes the board type and has one of the following values: 0 = ISA Bus 1 = PCI Bus 2 = CompactPCI 3 = PC/104 4 = MIPS 5 = RS232 6 = CAN

Example

To read the Card ID, the command ReadIO is used. Assuming the value returned is 0x1805, this can be interpreted as: Prodigy-PCI Motion Card, board generation 8, board revision 5.0.

Related commands

In addition to the low-level ReadIO command, a ReadCardID utility command is also supported by C-Motion and VB-Motion. This utility command returns the Card ID in the format described above. See the *Magellan Motion Processor Programmer's Command Reference* for more information.

Connections & associated signals

The Card ID is an internal function of the board. There are no signals directly associated with this function.

2.3.11 Memory Configuration

Trace capture is a powerful feature of the Prodigy-PCI Motion Card which allows various Magellan Motion Processor parameters and registers to be continuously captured and stored to an external memory buffer. The captured data may later be downloaded by the host using standard memory buffer access commands. Data traces are useful for optimizing DC brush and brushless DC performance, verifying trajectory behavior, capturing sensor data, or to assist with any type of monitoring where a precise time-based record of the system's behavior is required.

Maintaining the trace buffer requires the motion processor to perform extra work during each cycle. Most applications should disable trace capture once the motion application has been debugged. It is also possible to access the on-board memory independent of the trace function, which allows it to be used for generic storage. The remaining on-board memory available for trace must be calculated accordingly. See the "Trace Capture" section of the *Magellan Motion Processor User's Guide* for more information on these subjects.

The Prodigy-PCI Motion Card has 40 KBytes of on-board dual-ported memory (DPRAM) which is interfaced to the motion processor as well as the PCI decode circuitry allowing two paths of communication. In many applications, the Magellan Motion Processor can be used to store and access all trace data. This method of operation is described in the *Magellan Motion Processor User's Guide.* Alternatively, for moving large amounts of data, the host computer can directly interface to the DPRAM so that the memory can be used in a real-time circular buffer mode. In this configuration, the motion processor stores data to the DPRAM and the host processor reads the data using the PCI interface instead of using motion processor commands. The PCI interface offers a much higher read and write rate than is available through the motion processor. This allows an almost unlimited amount of "virtual" trace storage space, as the host computer's memory or hard drive is used to store the data.

For both the host computer and the Magellan Motion Processor, the 40 KBytes of DPRAM memory is accessed from offset 0 to offset 10,240, where the offset is calculated in 4-byte (32 bit) dwords. The number of dwords read or written can be 0 to 10,240, but must never be greater than (10,240 - *offset_in_dwords*).

The following commands are used by the host to read and write directly to the DPRAM:

> ReadDPRAM
> WriteDPRAM

By using these commands, blocks of 32-bit data can be read or written to the DPRAM. While a trace is in progress, the location to which the motion processor is currently writing may be determined by using the motion processor command **GetBufferWriteIndex**. Only data behind the current write index location should be read.

These commands are documented in the *Magellan Motion Processor Programmer's Command Reference.*

2.4 Signal Processing and Hardware Functions

These functions are implemented in hardware and are not directly user-programmable. The following sections are organized into related groups of signals, and provide information which may be helpful when connecting the motion system.

2.4.1 Home, AxisIn, AxisOut, Limits, Hall Sensors

These signals are conditioned by the board, and then input or output directly to the Magellan Motion Processor. The *Magellan Motion Processor User's Guide* explains the functions provided in connection with these various signals. Most of the signals are optional, and are connected depending on the nature of the application.

These signals are named *Home1-4, AxisIn1-4, AxisOut1-4, PosLim1-4* (positive direction limit input), *NegLim1-4* (negative direction limit input), and *Hall1A-4C* (12 signals in all).

Connections & associated signals

These signals are single-ended digital inputs to the board, with the exception of *AxisOut,* which is a single-ended output. There are no associated connections required for these signals to function properly, however, one or more of the digital grounds must be connected. The input signals are pulled up through 4.7 kOhm resistors to 5V. The default power-up value for all *AxisOut* signals is high.

For a complete description of the pinout connections to and from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

2.4.2 QuadA, QuadB, Index

These signals provide position feedback to the motion controller which is used to track motor position. For DC brush and brushless DC motors, they are required for proper operation. For microstepping or step (pulse & direction) motors, they are optional.

The encoder-processing circuitry provides a multi-stage digital filter of the *QuadA, QuadB*, and *Index* signals for each axis. This provides additional protection against erroneous noise spikes, thus improving reliability and motion integrity. These signals are named *QuadA1+* through *QuadB4-* (16 signals), and *Index1+* through *Index4-* (8 signals).

Connections & associated signals

These signals can be connected in one of two ways. Single-ended means that only one wire per signal is used, while differential means two wires encode each signal (labeled + and -). Differential transmission is generally recommended for the highest level of reliability, because it provides greater noise immunity than a single-ended connection scheme.

If single-ended connections are used, only the + signal is connected, and the - signal should be left floating. For example, in connecting to the A quadrature input, *QuadA1+* connects to the signal, and *QuadA1-* remains floating. If differential connections are used, both the + and - signals are used. Differential or single-ended termination must be selected through resistor pack installation. For details see the table in Section 1.8, "Preparing the Board for Installation." Note that all quadrature and index connections should be in either single-ended or differential mode. It is not possible to mix on a signal-by-signal basis. When using the system with differential connections, the polarity of the differential signal can be reversed by swapping the + and - connections. This may be useful for altering the motor and/or encoder direction; however, this same function can also be accomplished through commands to the Prodigy-PCI Motion Card. See the *Magellan Motion Processor User's Guide* for more information. Associated connections supported by the board are the +5V output signals. These are provided as a convenience, as they are generally connected to a corresponding input on the encoder to power its internal circuitry. As was the case for the digital input signals, one or more of the digital grounds must also be connected.

For a complete description of the pinout connections to and from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

2.4.3 Analog Input

The *Analog1-8* signals provide general purpose input of up to eight analog signals. The voltages present at these various connections do not directly affect the Prodigy-PCI Motion Card's behavior. However, they can be read through the Prodigy-PCI Motion Card, thus providing a convenient way of importing analog signal levels which may be acted upon by the user's application code located on the host PC. These signals are read using the Magellan command **ReadAnalog**. For more information on reading the value of these analog inputs, see the *Magellan Motion Processor User's Guide*. The minimum allowed input voltage is 0.0V, and the maximum allowed input voltage is 3.3V. To determine the numerical value that will be read by the Prodigy-PCI Motion Card given a specific voltage, the following formula is used:

ReadValue = AnalogVoltage * 65,536 / 3.3V

Conversely, given a read value, the voltage at the connection is calculated as:

AnalogVoltage = ReadValue * 3.3V / 65,536

Connections & associated signals

For analog voltages to be read correctly, in addition to the analog signal itself, *AnalogGND* (analog ground) must be connected.

2.4.4 Pulse and Direction

For pulse & direction applications, these signals provide a stream of pulse and direction data, and are compatible with a wide variety of off-the-shelf step motor amplifiers. These signals are generated by the Magellan Motion Processor and are named *Pulse1-4* and *Direction1-4*. The default value at power-up and reset for all pulse and direction output signals is: pulse signal is high; direction signal is low.

Connections & associated signals

Both single-ended and differential line driver versions of these signals are output from the Prodigy-PCI Card. There are no associated connections required for these signals to function properly, however, one or more of the digital grounds must be connected. For a complete description of the pinout connections to and from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

2.4.5 PWM Out

For DC brush, brushless DC or microstepping motors, these signals provide PWM (pulse width modulated) motor command signals when the motor output mode is set to *PWMSignMagnitude* or *PWM5050Magnitude*. The number of signals per axis varies, depending on factors such as the motor type, the number of phases of the motor, and the motor drive method (sign/magnitude or 50/50). See Chapter 3, "Prodigy-PCI Electrical Reference," for complete connection tables for various motor configurations.

These signals are named *PWMMag1A-4C* (12 signals) and *PWMSign1A-4B* (8 signals).

Connections & associated signals

These signals are generated by the Magellan Motion Processor. There are no associated connections required for these signals to function properly, however, one or more of the digital grounds must be connected. For a complete description of the pinout connections to and from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

2.4.6 DAC Out

For DC brush, brushless DC or microstepping motors, this is the analog motor command when the motor output mode is set to DAC (digital-to-analog converter). These signals are named *DAC1A - DAC4B* (8 signals), and vary between -10V and +10V. The number of signals per axis depends upon the motor type. See Section 3.4, "Environmental and Electrical Ratings," and the *Magellan Motion Processor User's Guide* for more information.

Connections & associated signals

For analog voltages to be output correctly, *AGND* (motor command ground) must be connected. For a complete description of the pinout connections to and from the board, see Chapter 3, "Prodigy-PCI Electrical Reference."

3. Prodigy-PCI Electrical Reference

In This Chapter

▶ User-Settable Components
▶ Connectors
▶ Connections Summary—Motor Amplifiers
▶ Command Summary—Board-Specific Functions
▶ Environmental and Electrical Ratings

3.1 User-Settable Components

Figure 3-1 illustrates the locations of the principal components of the Prodigy-PCI Motion Cards. The important user-accessible components of the board are listed in the following table:

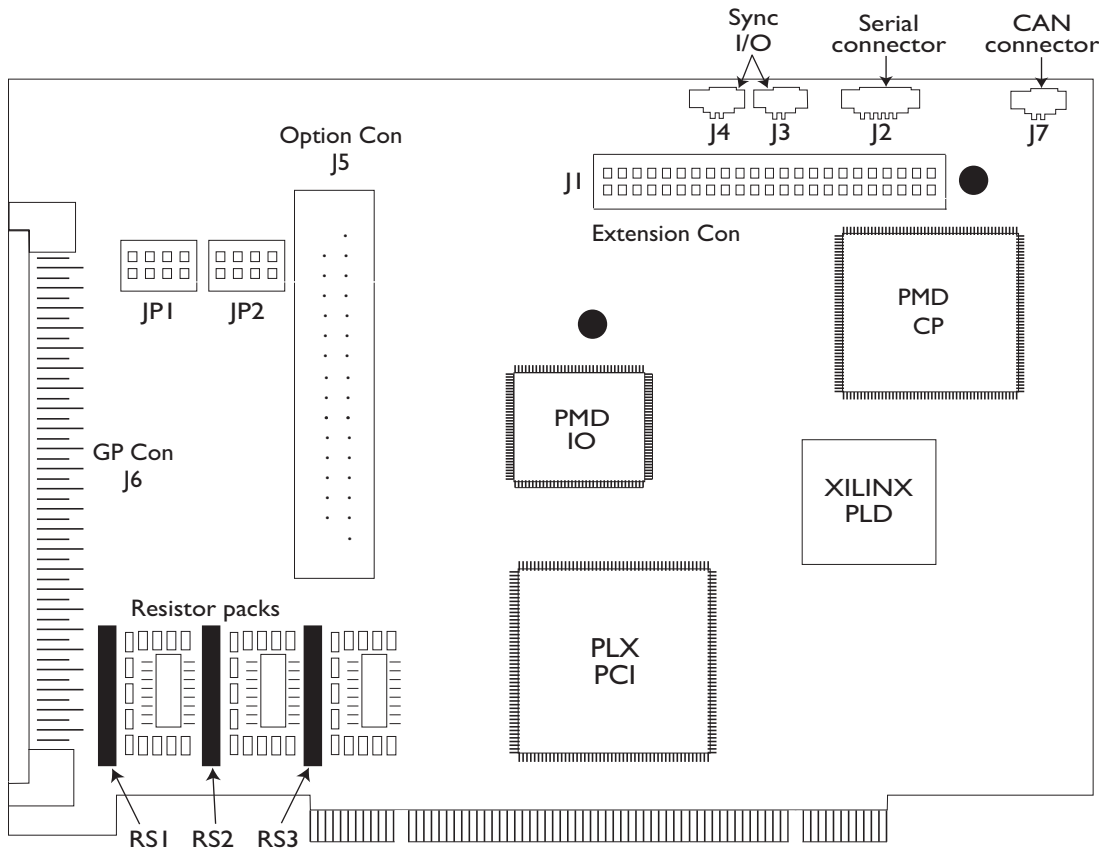| Component	Function
Resistor packs RS1, RS2, and RS3	Sets the encoder termination.
JP1 and JP2	Selects pulse & direction (step) or DC brush and brushless DC mode for each axis.

Figure 3-1: Component locations on the Prodigy-PCI

3.1.1 Encoder Settings and Resistor Packs

The RS1, RS2, and RS3 resistor packs are installed on the Prodigy-PCI Motion Card by default. Encoder inputs may be connected differentially, with two wires for *QuadA*, *QuadB*, and *Index* signals, or with just one wire per signal. If differential connections are being employed, resistor packs RS1, RS2, and RS3 should be installed. If single-ended encoders are used, remove all three resistor packs (see Figure 3-1), and connect encoder signals to the positive encoder input only. The negative input may remain unconnected. Encoder connections are detailed in the following tables:

Encoder connections when using differential encoder input:

Signal	GP Con J6 Pin Connections			
	Axis 1	**Axis 2**	**Axis 3**	**Axis 4**
QuadA*n*+	1	19	51	69
QuadA*n*-	2	20	52	70
QuadB*n*+	3	21	53	71
QuadB*n*-	4	22	54	72
Index*n*+	5	23	55	73
Index*n*-	6	24	56	74
Vcc	7	25	57	75
GND	8	26	58	76

Encoder connections when using single-ended encoder input:

Signal	GP Con J6 Pin Connections			
	Axis 1	**Axis 2**	**Axis 3**	**Axis 4**
QuadA*n*	1	19	51	69
QuadB*n*	3	21	53	71
Index*n*	5	23	55	73
Vcc	7	25	57	75
GND	8	26	58	76

The Prodigy-PCI Motion Card has minimal jumper settings; most settings are software configurable. To prepare the board for installation, the following user-specified resistor pack options should be checked.

Item	Setting	Description
Resistor packs RS1, RS2, RS3	Installed; this is the default setting of resistor packs RS1 - RS3.	If differential connections are being used, leave these resistor packs installed.
	Removed	If single-ended encoder connections are being used, remove the resistor packs.

3.1.2 Motor Jumpers

Various possibilities exist for setting up the Prodigy-PCI Motion Card for use with either DC brush, DC brushless (with or without on-board commutation), microstepping, or step (pulse & direction) motors. The default mode at power-on is for DC brush motor on all axes.

The jumpers JP1 and JP2 consist of four jumper blocks each, relating to axis 1, 2, 3, or 4. The following table describes the correct jumper settings for each motor. See Figure 3-1 for JP1 and JP2 locations and jumper numbering. In the following table, the individual jumpers are numbered from left to right on each jumper block, each jumper number preceded by its jumper block: JP1-1 is the jumper farthest to the left on JP1; JP1-2 is second from the left, and so on.

An example of open and closed jumpers is shown in Figure 3-2. Motor type selection determines the default output mode. See the *Magellan Motion Processor User's Guide* for more information.

Axis	Pulse & Direction	DC Brush, Brushless DC
1	JP1-1 installed	JP2-1 installed
2	JP1-2 installed	JP2-2 installed
3	JP1-3 installed	JP2-3 installed
4	JP1-4 installed	JP2-4 installed

NOTE: Caution should be used when setting motor jumpers. Be sure that the jumper settings are valid for the specified motor type.
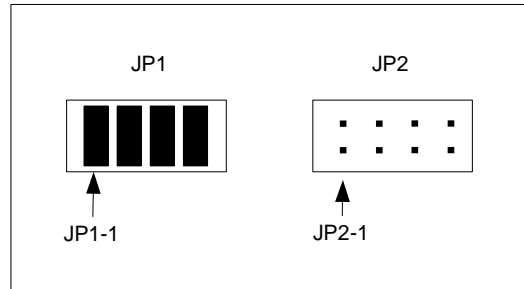
3.2 Connectors

There are seven user-accessible connectors on the Prodigy-PCI Motion Card; the connectors and their functions are outlined in the following table:

Connector	Functionality
GP Con	Provides connections to an external amplifier and motor elements.
Option Con	For Brushless DC and microstepping motors, the Option Con connector provides additional connections to external amplifier and motor components.
Serial	The Serial connector allows the Prodigy-PCI Motion Card to be controlled via a serial port, or to enable monitoring over a separate serial port.
Sync I/O*	The two Sync I/O connectors are in parallel. For both connectors, pin 1 is the sync-in and sync-out pin, determined by the master and slave card. Pin 2 is the ground (GND).
CAN	The CAN connector allows for integration with a CAN 2.0B network.
Extension Con	The extension connector allows the DC-1000 card to be installed on the Prodigy-PCI Motion Card.

*Sync I/O is not available on the 9255xxx series motion boards.

3.2.1 GP Con

The GP Con connector is used to connect to various motion peripherals such as encoders, amplifiers, etc. It is a 100-position, high-density connector. The following table provides part ordering information for those users interested in developing their own cables for this connector.

Part Description	Vendors and Part Numbers
Board connector	AMP: 2-5178238-9 3M: N102A0-52B2PC
Mating connector (for shielded cable)	AMP: 2-5175677-9 (plug) AMP: 1-176793-0 (shell) 3M: 101A0-6000EC (plug) 3M: 103A0-A200-00 (shell)

Depending on the motor type, various configurations of GP Con pin connections apply.

3.2.2 GP Con Using DC Brush Motors

Pin	Connection	Description	Pin	Connection	Description
1	QuadA1+	Quadrature A+ encoder input (axis 1)	51	QuadA3+	Quadrature A+ encoder input (axis 3)
2	QuadA1-	Quadrature A- encoder input (axis 1)	52	QuadA3-	Quadrature A- encoder input (axis 3)
3	QuadB1+	Quadrature B+ encoder input (axis 1)	53	QuadB3+	Quadrature B+ encoder input (axis 3)
4	QuadB1-	Quadrature B- encoder input (axis 1)	54	QuadB3-	Quadrature B- encoder input (axis 3)
5	Index1+	Index+ input (axis 1)	55	Index3+	Index+ input (axis 3)
6	Index1-	Index- input (axis 1)	56	Index3-	Index- input (axis 3)
7	Vcc	+5V	57	Vcc	+5V
8	GND	Ground	58	GND	Ground
9	PosLim1	Pos. direction limit switch input (axis 1)	59	PosLim3	Pos. direction limit switch input (axis 3)
10	NegLim1	Neg. direction limit switch input (axis 1)	60	NegLim3	Neg. direction limit switch input (axis 3)
11	Home1	Home input (axis 1)	61	Home3	Home input (axis 3)
12	GND	Ground	62	GND	Ground
13	AxisOut1	AxisOut output (axis 1)	63	AxisOut3	AxisOut output (axis 3)
14	PWMMag1A	PWM magnitude output (axis 1)	64	PWMMag3A	PWM magnitude output (axis 3)
15	PWMSign1A	PWM sign output (axis 1)	65	PWMSign3A	PWM sign output (axis 3)
16	AxisIn1	AxisIn input (axis 1)	66	AxisIn3	AxisIn input (axis 3)
17	DAC1A	Analog mtr cmd output (axis 1), ±10V	67	DAC3A	Analog mtr cmd output (axis 3), ±10V
18	AGND	Ground for analog motor command	68	AGND	Ground for analog motor command
19	QuadA2+	Quadrature A+ encoder input (axis 2)	69	QuadA4+	Quadrature A+ encoder input (axis 4)
20	QuadA2-	Quadrature A- encoder input (axis 2)	70	QuadA4-	Quadrature A- encoder input (axis 4)
21	QuadB2+	Quadrature B+ encoder input (axis 2)	71	QuadB4+	Quadrature B+ encoder input (axis 4)
22	QuadB2-	Quadrature B- encoder input (axis 2)	72	QuadB4-	Quadrature B- encoder input (axis 4)
23	Index2+	Index+ input (axis 2)	73	Index4+	Index+ input (axis 4)
24	Index2-	Index- input (axis 2)	74	Index4-	Index- input (axis 4)
25	Vcc	+5V	75	Vcc	+5V
26	GND	Ground	76	GND	Ground
27	PosLim2	Pos. direction limit switch input (axis 2)	77	PosLim4	Pos. direction limit switch input (axis 4)
28	NegLim2	Neg. direction limit switch input (axis 2)	78	NegLim4	Neg. direction limit switch input (axis 4)
29	Home2	Home input (axis 2)	79	Home4	Home input (axis 4)
30	AxisOut2	AxisOut output (axis 2)	80	AxisOut4	AxisOut output (axis 4)
31	PWMMag2A	PWM magnitude output (axis 2)	81	PWMMag4A	PWM magnitude output (axis 4)
32	PWMSign2A	PWM sign output (axis 2)	82	PWMSign4A	PWM sign output (axis 4)
33	AxisIn2	AxisIn input (axis 2)	83	AxisIn4	AxisIn input (axis 4)
34	DAC2A	Analog mtr cmd output (axis 2), ±10V	84	DAC4A	Analog mtr cmd output (axis 4), ±10V
35	AGND	Ground for analog motor command	85	AGND	Ground for analog motor command
36	DigitalIn0	General purpose digital input 0	86	DigitalIn4	General purpose digital input 4
37	DigitalIn1	General purpose digital input 1	87	DigitalIn5	General purpose digital input 5
38	DigitalIn2	General purpose digital input 2	88	DigitalIn6	General purpose digital input 6
39	DigitalIn3	General purpose digital input 3	89	DigitalIn7	General purpose digital input 7
40	AmpEnable1	Amplifier enable signal (axis 1)	90	AmpEnable3	Amplifier enable signal (axis 3)
41	DigitalOut0	General purpose digital output 0	91	DigitalOut4	General purpose digital output 4
42	DigitalOut1	General purpose digital output 1	92	DigitalOut5	General purpose digital output 5
43	DigitalOut2	General purpose digital output 2	93	DigitalOut6	General purpose digital output 6
44	DigitalOut3	General purpose digital output 3	94	DigitalOut7	General purpose digital output 7
45	AmpEnable2	Amplifier enable signal (axis 2)	95	AmpEnable4	Amplifier enable signal (axis 4)
46	Reset	Hardware reset input	96	AnalogGND	Gnd for general purpose analog inputs
47	Analog1	General purpose analog input 1	97	Analog5	General purpose analog input 5
48	Analog2	General purpose analog input 2	98	Analog6	General purpose analog input 6
49	Analog3	General purpose analog input 3	99	Analog7	General purpose analog input 7
50	Analog4	General purpose analog input 4	100	Analog8	General purpose analog input 8

3.2.3 GP Con Using Brushless DC or Microstepping Motors

Pin	Connection	Description	Pin	Connection	Description
1	QuadA1+	Quadrature A+ encoder input (axis 1)	51	QuadA3+	Quadrature A+ encoder input (axis 3)
2	QuadA1-	Quadrature A- encoder input (axis 1)	52	QuadA3-	Quadrature A- encoder input (axis 3)
3	QuadB1+	Quadrature B+ encoder input (axis 1)	53	QuadB3+	Quadrature B+ encoder input (axis 3)
4	QuadB1-	Quadrature B- encoder input (axis 1)	54	QuadB3-	Quadrature B- encoder input (axis 3)
5	Index1+	Index+ input (axis 1)	55	Index3+	Index+ input (axis 3)
6	Index1-	Index- input (axis 1)	56	Index3-	Index- input (axis 3)
7	Vcc	+5V	57	Vcc	+5V
8	GND	Ground	58	GND	Ground
9	PosLim1	Pos. direction limit switch input (axis 1)	59	PosLim3	Pos. direction limit switch input (axis 3)
10	NegLim1	Neg. direction limit switch input(axis 1)	60	NegLim3	Neg. direction limit switch input (axis 3)
11	Home1	Home input (axis 1)	61	Home3	Home input (axis 3)
12	GND	Ground	62	GND	Ground
13	AxisOut1	AxisOut output (axis 1)	63	AxisOut3	AxisOut output (axis 3)
14	n.c.	No connection	64	n.c.	No connection
15	n.c.	No connection	65	n.c.	No connection
16	AxisIn1	AxisIn input (axis 1)	66	AxisIn3	AxisIn input (axis 3)
17	DAC1A	Phase A analog mtr cmd output (axis 1), ±10V	67		DAC3A Phase A analog mtr cmd output (axis 3), ±10V
18	AGND	Ground for analog motor command	68	AGND	Ground for analog motor command
19	QuadA2+	Quadrature A+ encoder input (axis 2)	69	QuadA4+	Quadrature A+ encoder input (axis 4)
20	QuadA2-	Quadrature A- encoder input (axis 2)	70	QuadA4-	Quadrature A- encoder input (axis 4)
21	QuadB2+	Quadrature B+ encoder input (axis 2)	71	QuadB4+	Quadrature B+ encoder input (axis 4)
22	QuadB2-	Quadrature B- encoder input (axis 2)	72	QuadB4-	Quadrature B- encoder input (axis 4)
23	Index2+	Index+ input (axis 2)	73	Index4+	Index+ input (axis 4)
24	Index2-	Index- input (axis 2)	74	Index4-	Index- input (axis 4)
25	Vcc	+5V	75	Vcc	+5V
26	GND	Ground	76	GND	Ground
27	PosLim2	Pos. direction limit switch input (axis 2)	77	PosLim4	Pos. direction limit switch input (axis 4)
28	NegLim2	Neg. direction limit switch input (axis 2)	78	NegLim4	Neg. direction limit switch input (axis 4)
29	Home2	Home input (axis 2)	79	Home4	Home input (axis 4)
30	AxisOut2	AxisOut output (axis 2)	80	AxisOut4	AxisOut output (axis 4)
31	n.c.	No connection	81	n.c.	No connection
32	n.c.	No connection	82	n.c.	No connection
33	AxisIn2	AxisIn input (axis 2)	83	AxisIn4	AxisIn input (axis 4)
34	DAC2A	Phase A analog mtr cmd output (axis 2), ±10V	84		DAC4A Phase A analog mtr cmd output (axis 4), ±10V
35	AGND	Ground for analog motor command	85	AGND	Ground for analog motor command
36	DigitalIn0	General purpose digital input 0	86	DigitalIn4	General purpose digital input 4
37	DigitalIn1	General purpose digital input 1	87	DigitalIn5	General purpose digital input 5
38	DigitalIn2	General purpose digital input 2	88	DigitalIn6	General purpose digital input 6
39	DigitalIn3	General purpose digital input 3	89	DigitalIn7	General purpose digital input 7
40	AmpEnable1	Amplifier enable signal (axis 1)	90	AmpEnable3	Amplifier enable signal (axis 3)
41	DigitalOut0	General purpose digital output 0	91	DigitalOut4	General purpose digital output 4
42	DigitalOut1	General purpose digital output 1	92	DigitalOut5	General purpose digital output 5
43	DigitalOut2	General purpose digital output 2	93	DigitalOut6	General purpose digital output 6
44	DigitalOut3	General purpose digital output 3	94	DigitalOut7	General purpose digital output 7
45	AmpEnable2	Amplifier enable signal (axis 2)	95	AmpEnable4	Amplifier enable signal (axis 4)
46	Reset	Hardware reset input	96	AnalogGND	Gnd for general purpose analog inputs
47	Analog1	General purpose analog input 1	97	Analog5	General purpose analog input 5
48	Analog2	General purpose analog input 2	98	Analog6	General purpose analog input 6
49	Analog3	General purpose analog input 3	99	Analog7	General purpose analog input 7
50	Analog4	General purpose analog input 4	100	Analog8	General purpose analog input 8

3.2.4 GP Con Using Step (Pulse & Direction) Motors

Pin	Connection	Description	Pin	Connection	Description
1	QuadA1+	Quadrature A+ encoder input (axis 1)	51	QuadA3+	Quadrature A+ encoder input (axis 3)
2	QuadA1-	Quadrature A- encoder input (axis 1)	52	QuadA3-	Quadrature A- encoder input (axis 3)
3	QuadB1+	Quadrature B+ encoder input (axis 1)	53	QuadB3+	Quadrature B+ encoder input (axis 3)
4	QuadB1-	Quadrature B- encoder input (axis 1)	54	QuadB3-	Quadrature B- encoder input (axis 3)
5	Index1+	Index+ input (axis 1)	55	Index3+	Index+ input (axis 3)
6	Index1-	Index- input (axis 1)	56	Index3-	Index- input (axis 3)
7	Vcc	+5V	57	Vcc	+5V
8	GND	Ground	58	GND	Ground
9	PosLim1	Pos. direction limit switch input (axis 1)	59	PosLim3	Pos. direction limit switch input (axis 3)
10	NegLim1	Neg. direction limit switch input (axis 1)	60	NegLim3	Neg. direction limit switch input (axis 3)
11	Home1	Home input (axis 1)	61	Home3	Home input (axis 3)
12	GND	Ground	62	GND	Ground
13	AxisOut1	AxisOut output (axis 1)	63	AxisOut3	AxisOut output (axis 3)
14	Pulse1	Pulse output (axis 1)	64	Pulse3	Pulse output (axis 3)
15	Direction1	Direction output (axis 1)	65	Direction3	Direction output (axis 3)
16	AxisIn1	AxisIn input (axis 1)	66	AxisIn3	AxisIn input (axis 3)
17	AtRest1	Atrest indicator output (axis 1)	67	AtRest3	Atrest indicator output (axis 3)
18	GND	Ground	68	GND	Ground
19	QuadA2+	Quadrature A+ encoder input (axis 2)	69	QuadA4+	Quadrature A+ encoder input (axis 4)
20	QuadA2-	Quadrature A- encoder input (axis 2)	70	QuadA4-	Quadrature A- encoder input (axis 4)
21	QuadB2+	Quadrature B+ encoder input (axis 2)	71	QuadB4+	Quadrature B+ encoder input (axis 4)
22	QuadB2-	Quadrature B- encoder input (axis 2)	72	QuadB4-	Quadrature B- encoder input (axis 4)
23	Index2+	Index+ input (axis 2)	73	Index4+	Index+ input (axis 4)
24	Index2-	Index- input (axis 2)	74	Index4-	Index- input (axis 4)
25	Vcc	+5V	75	Vcc	+5V
26	GND	Ground	76	GND	Ground
27	PosLim2	Pos. direction limit switch input (axis 2)	77	PosLim4	Pos. direction limit switch input (axis 4)
28	NegLim2	Neg. direction limit switch input (axis 2)	78	NegLim4	Neg. direction limit switch input (axis 4)
29	Home2	Home input (axis 2)	79	Home4	Home input (axis 4)
30	AxisOut2	AxisOut output (axis 2)	80	AxisOut4	AxisOut output (axis 4)
31	Pulse2	Pulse output (axis 2)	81	Pulse4	Pulse output (axis 4)
32	Direction2	Direction output (axis 2)	82	Direction4	Direction output (axis 4)
33	AxisIn2	AxisIn input (axis 2)	83	AxisIn4	AxisIn input (axis 4)
34	AtRest2	Atrest indicator output (axis 2)	84	AtRest4	Atrest indicator output (axis 4)
35	GND	Ground	85	GND	Ground
36	DigitalIn0	General purpose digital input 0	86	DigitalIn4	General purpose digital input 4
37	DigitalIn1	General purpose digital input 1	87	DigitalIn5	General purpose digital input 5
38	DigitalIn2	General purpose digital input 2	88	DigitalIn6	General purpose digital input 6
39	DigitalIn3	General purpose digital input 3	89	DigitalIn7	General purpose digital input 7
40	AmpEnable1	Amplifier enable signal (axis 1)	90	AmpEnable3	Amplifier enable signal (axis 3)
41	DigitalOut0	General purpose digital output 0	91	DigitalOut4	General purpose digital output 4
42	DigitalOut1	General purpose digital output 1	92	DigitalOut5	General purpose digital output 5
43	DigitalOut2	General purpose digital output 2	93	DigitalOut6	General purpose digital output 6
44	DigitalOut3	General purpose digital output 3	94	DigitalOut7	General purpose digital output 7
45	AmpEnable2	Amplifier enable signal (axis 2)	95	AmpEnable4	Amplifier enable signal (axis 4)
46	Reset	Hardware reset input	96	AnalogGND	Gnd for general purpose analog inputs
47	Analog1	General purpose analog input 1	97	Analog5	General purpose analog input 5
48	Analog2	General purpose analog input 2	98	Analog6	General purpose analog input 6
49	Analog3	General purpose analog input 3	99	Analog7	General purpose analog input 7
50	Analog4	General purpose analog input 4	100	Analog8	General purpose analog input 8

3.2.5 Option Con Connector

When the Prodigy-PCI Motion Card is used with either brushless DC or microstepping motors, the Option Con connector provides additional signals for multi-phase motor output and input of signals such as Hall sensors. The Option Con connector is a 68-pin SCSI-style connector. The following table provides ordering information for those users interested in developing their own cables for this connector.

Part description	Vendors and Part Numbers
Board connector	Molex: 1587-0205
Mating connector plug (for AWG 30, .025 ribbon)	Molex: 70498-4068 Harting: 60 06 068 5440

3.2.6 Option Con Using Brushless DC, Microstepping, or Step (Pulse & Direction) Motors

Pin	Connection	Description	Pin	Connection	Description
1	PWMMag1A	Phase A PWM magnitude output (axis 1)	35	PWMMag1B	Phase B PWM magnitude output (axis 1)
2	PWMMag1C / PWMSign1B	Phase C PWM magnitude output (axis 1)	36	GND	Ground
3	PWMSign1A	Phase B PWM sign output (axis 1)	37	GND	Ground
4	PWMMag2A	Phase A PWM magnitude output (axis 2)	38	PWMMag2B	Phase B PWM magnitude output (axis 2)
5	PWMMag2C / PWMSign2B	Phase C PWM magnitude output (axis 2)	39	GND	Ground
6	PWMSign2A	Phase B PWM sign output (axis 2)	40	GND	Ground
7	PWMMag3A	Phase A PWM magnitude output (axis 3)	41	PWMMag3B	Phase B PWM magnitude output (axis 3)
8	PWMMag3C / PWMSign3B	Phase C PWM magnitude output (axis 3)	42	GND	Ground
9	PWMSign3A	Phase B PWM sign output (axis 3)	43	GND	Ground
10	PWMMag4A	Phase A PWM magnitude output (axis 4)	44	PWMMag4B	Phase B PWM magnitude output (axis 4)
11	PWMMag4C / PWMSign4B	Phase C PWM magnitude output (axis 4)	45	GND	Ground
12	PWMSign4A	Phase B PWM sign output (axis 4)	46	GND	Ground
13	Hall1A	Phase A Hall sensor input (axis 1)	47	Hall1B	Phase B Hall sensor input (axis 1)
14	Hall1C	Phase C Hall sensor input (axis 1)	48	GND	Ground
15	Hall2A	Phase A Hall sensor input (axis 2)	49	Hall2B	Phase B Hall sensor input (axis 2)
16	Hall2C	Phase C Hall sensor input (axis 2)	50	GND	Ground
17	Hall3A	Phase A Hall sensor input (axis 3)	51	Hall3B	Phase B Hall sensor input (axis 3)
18	Hall3C	Phase C Hall sensor input (axis 3)	52	GND	Ground
19	Hall4A	Phase A Hall sensor input (axis 4)	53	Hall4B	Phase B Hall sensor input (axis 4)
20	Hall4C	Phase C Hall sensor input (axis 4)	54	GND	Ground
21	axis 1 pulse +	Axis 1 differential pulse +	55	axis 1 pulse -	Axis 1 differential pulse -
22	axis 2 pulse +	Axis 2 differential pulse +	56	axis 2 pulse -	Axis 2 differential pulse -
23	axis 3 pulse +	Axis 3 differential pulse +	57	axis 3 pulse -	Axis 3 differential pulse -
24	axis 4 pulse +	Axis 4 differential pulse +	58	axis 4 pulse -	Axis 4 differential pulse -
25	axis 1 direction +	Axis 1 differential direction +	59	axis 1 direction -	Axis 1 differential direction -
26	axis 2 direction +	Axis 2 differential direction +	60	axis 2 direction -	Axis 2 differential direction -
27	axis 3 direction +	Axis 3 differential direction +	61	axis 3 direction -	Axis 3 differential direction -
28	axis 4 direction +	Axis 4 differential direction +	62	axis 4 direction -	Axis 4 differential direction -
29	VCC	+5V	63	VCC	+5V
30	AGND	Ground for analog motor command	64	AGND	Ground for analog motor command
31	DAC1B	Phase B analog mtr cmd output (axis 1), ±10V	65	DAC1A	Phase A analog mtr cmd output (axis 1), ±10V
32	DAC2B	Phase B analog mtr cmd output (axis 2), ±10V	66	DAC2A	Phase A analog mtr cmd output (axis 2), ±10V
33	DAC3B	Phase B analog mtr cmd output (axis 3), ±10V	67	DAC3A	Phase A analog mtr cmd output (axis 3), ±10V
34	DAC4B	Phase B analog mtr cmd output (axis 4), ±10V	68	DAC4A	Phase A analog mtr cmd output (axis 4), ±10V

3.2.7 Serial Connector

The following table gives the pinout for the 5-position Molex 1.25 mm header used for serial communications.

Pin	Connection	Description
1	SrlEnable	Serial enable (used only for RS422/485)
2	SrlXmt	Serial transmit output
3	SrlRcv	Serial receive input
4	GND	Ground
5	Vcc	+3.3V

For more information on communicating to the Magellan Motion Processor via the serial port, see the *Magellan Motion Processor User's Guide*.

3.2.8 Sync I/O Connector

The two *Sync I/O* connectors located on the back of the Prodigy-PCI Motion Card (J3 and J4 in Figure 3-1) allow for the synchronization of multiple Prodigy-PCI Motion Cards within a single system. This configuration enables operation within the same cycle period. If multiple boards are installed, yet not inter-connected, any additional boards would begin working after the first card (master) was initialized. However, none of the axes would be synchronized. With *Sync I/O* activated, the servo loops of all slave boards are synchronized to the servo loop of the master board. This allows for precise synchronization of all implemented axes.

To connect two or more Prodigy-PCI Motion Cards for synchronization, a *Sync I/O* cable (one cable for each set of two boards) is required. This cable may be connected to either of the two *Sync I/O* connectors on the boards. Both connectors function as either an input or output; the two sync connectors are wired in parallel. For more information on synchronizing multiple Prodigy-PCI Motion Cards, see the *Magellan Motion Processor User's Guide*.

The pinouts for the *Sync I/O* connectors are defined in the following table:

Pin Number	Signal
1	Sync-in or sync-out pin, depending on whether the card is master or slave.
2	GND

The two *Sync I/O* connectors are 2-position Molex 1.25 mm headers, Molex Part # 53398-0271.

NOTE: Sync I/O is not available on the PR9255xx series boards. Additionally, if any axis is set for pulse & direction, the synchronization option cannot be used.

Figure 3-3 shows three synchronized Prodigy Motion Cards.
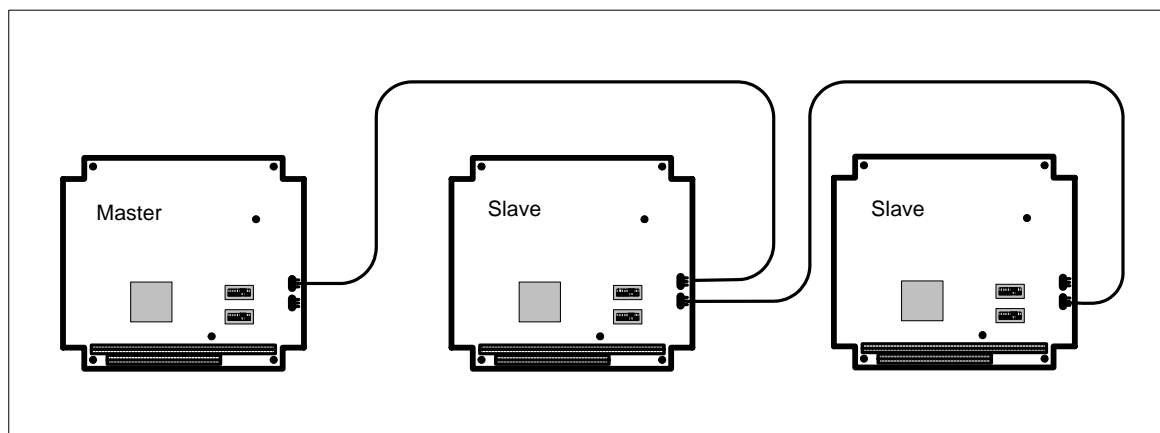
Figure 3-3: Sync I/O connector to three boards

3.2.9 CAN Connector

The Prodigy-PCI Motion Card's controller area network (CAN) transceivers are designed for use with CAN controllers, or with equivalent devices. They are intended for use in applications employing the CAN serial communication physical layer in accordance with the ISO 11898 standard. The transceiver provides differential transmit and differential receive capability to/from a CAN controller at speeds up to 1 Mbps.

The pinouts for the CAN connector are as follows:

Pin Number	Signal
1	CANH
2	CANL

The CANH and CANL connections are made via the J7 connector on the Prodigy-PCI Motion Card (see Figure 3-1). It is a 2-position Molex 1.25 mm header, Molex Part # 53398-0271.

3.2.10 Extension Connector

This is a 46-pin connector (labeled J1 in Figure 3-1), which connects the DC-1000 expansion board to the Prodigy-PCI. The DC-1000 is a a Synchronous Serial Interface (SSI) for absolute encoders, and connects to the Prodigy-PCI Motion Card's parallel input connector. The DC-1000, its functionality, and installation instructions, are fully detailed in Chapter 4, "Interconnect Modules."

3.2.11 Connector Parts Reference

The following table provides part ordering information for those users interested in developing their own cables.

Label	Description	Board Connector	Mating Connector
J1	Extension Con	Samtec TMM-123-01-L-D-SM	
J2	Serial	Molex 53261-0571	Molex 51021-0500
J3, J4	Sync I/O	Molex 53261-0271	Molex 51021-0200
J5	Option Con	Molex 1587-0205	Molex 70498-4068 Harting 60 06 068 5440
J6	GP Con	AMP 2-5178238-9 3M N102A0-52B2PC	AMP 2-5175677-9 (plug) AMP 1-176793-0 (shell) 3M 101A0-6000EC (plug) 3M 103A0-A200-00 (shell)
J7	CAN	Molex 53261-0271	Molex 51021-0200

3.3 Connections Summary— Motor Amplifiers

The Prodigy-PCI Motion Card supports four methods of output to motor amplifiers:

Motor Type	Signal Output
DAC	Analog signals from the onboard D-A converters
PWM sign-magnitude	Pulse width modulated signals with separate magnitude and sign signals per output phase
PWM 50/50	Pulse width modulated square-wave signals with a single PWM signal per output phase
Pulse & Direction	One signal representing stepping information, and a signal representing direction

In addition, each motor axis may have one, two, or three output phases associated with it. For DC brush motors, the number of phases is one; for multi-phase motors such as Brushless DC or microstepping motors, the number of phases can be two or three, depending on the output waveform programmed into the Prodigy-PCI Motion Card. For more information, see the *Magellan Motion Processor User's Guide*.

The following tables provide convenient summaries of amplifier connections for various common configurations of motor output method and motor type. These outputs should be connected from the designated connector pins to the appropriate amplifier inputs. Note that the names of the pins may vary among amplifiers. Common names are shown.

3.3.1 DC Brush Motor Connections

Motor Output Method	Prodigy-PCI Connection Name	Amplifier Input Connection Name*	Connection			
			Axis 1	Axis 2	Axis 3	Axis 4
DAC	DAC1-4A	Ref+ or V+	GP-17	GP-34	GP-67	GP-84
	AGND	Ref- or Gnd	GP-18	GP-35	GP-68	GP-85
PWM sign/magnitude	PWMMag1-4A	PWM magnitude	GP-14	GP-31	GP-64	GP-81
	PWMSign1-4A	PWM direction	GP-15	GP-32	GP-65	GP-82
	GND	Ground	GP-8	GP-26	GP-58	GP-76

Names of amplifier connections may vary. Common names are shown.

3.3.2 Brushless DC Motor Connections

Motor Output Method	Prodigy-PCI Connection Name	Amplifier Input Connection Name*	Connection			
			Axis 1	Axis 2	Axis 3	Axis 4
DAC	DAC1-4A	Ref1+ or V1+	Option-65	Option-66	Option-67	Option-68
	DAC1-4B	Ref2+ or V2+	Option-31	Option-32	Option-33	Option-34
	AGND	Ref- or Gnd	Option-30	Option-64	Option-30	Option-64
PWM 50/50	PWMMag1-4A	PWM phase 1	Option-1	Option-4	Option-7	Option-10
	PWMMag1-4B	PWM phase 2	Option-35	Option-38	Option-41	Option-44
	PWMMag1-4C	PWM phase 3	Option-2	Option-5	Option-8	Option-11
	GND	Ground	Option-36	Option-39	Option-42	Option-45

Names of amplifier connections may vary. Common names are shown.

3.3.3 Microstepping Motor Connections

Motor Output Method	Prodigy-PCI Connection Name	Amplifier Input Connection Name*	Connection			
			Axis 1	Axis 2	Axis 3	Axis 4
DAC	DAC1-4A	Ref1+ or V1+	Option-65	Option-66	Option-67	Option-68
	DAC1-4B	Ref2+ or V2+	Option-31	Option-32	Option-33	Option-34
	AGND	Ref- or Gnd	Option-30	Option-64	Option-30	Option-64
PWM sign/magnitude	PWMMag1-4A	PWM magnitude	Option-1	Option-4	Option-7	Option-10
	PWMSign1-4A	PWM direction	Option-3	Option-6	Option-9	Option-12
	PWMMag1-4B	PWM magnitude	Option-35	Option-38	Option-41	Option-44
	PWMSign1-4B	PWM direction	Option-2	Option-5	Option-8	Option-11
	GND	Ground	Option-36	Option-39	Option-42	Option-45

Names of amplifier connections may vary. Common names are shown.

3.3.4 Step (Pulse & Direction) Motor Connections

Motor Output Method	Prodigy-PCI Connection Name	Amplifier Input Connection Name*	Connection			
			Axis 1	Axis 2	Axis 3	Axis 4
Pulse & direction (single-ended)	Pulse1-4	Pulse or step	GP-14	GP-31	GP-64	GP-81
	Direction1-4	Direction	GP-15	GP-32	GP-65	GP-82
	GND	Ground	GP-8	GP-26	GP-58	GP-76
Pulse & direction (differential)	Pulse +	Pulse + or step +	Option-21	Option-22	Option-23	Option-24
	Pulse -	Pulse - or step -	Option-55	Option-56	Option-57	Option-58
	Direction +	Direction +	Option-25	Option-26	Option-27	Option-28
	Direction -	Direction -	Option-59	Option-60	Option-61	Option-62
Pulse & direction	AtRest 1-4	At Rest	GP-17	GP-34	GP-67	GP-84

Names of amplifier connections may vary. Common names are shown.

3.4 Environmental and Electrical Ratings

Dimensions:	PCI short card, 4.2" (107mm) x 5.9" (150mm)
Storage temperature:	-40 to +125 degrees C (-40° F to +257° F)
Operating temperature:	0 to +70 degrees C (32° F to +158° F)
Power requirements:	4.8V to 5.25V operating range, 0.6A (no outputs on)
Supply voltage limits:	-0.3V to +7V
Analog (DAC) output range:	-10.0V to +10.0V, ± 3mA min/axis, short circuit protected
Analog input range:	0 to 3.3V, 1.4 KOhm input impedance
Digital I/O voltage range:	0V to 5V, TTL thresholds, inputs pulled up to 5V through 4.7 kOhm resistors
Digital outputs drive capacity:	DC output source or sink current: ± 50mA
CAN communications:	2.0B compliant, non-isolated, 1 Mbps
Serial communications:	3.3V signaling, unbuffered (no transceiver)

3.5 User I/O Memory Map

The Magellan Motion Processor reserves the sector from address 1000h to 10FFh in peripheral space for user-defined I/O devices. The Prodigy-PCI Motion Card uses this sector as shown in the following table:

Address	Device	Description
1000h	General Purpose I/O	Includes the 8 digital inputs and 8 digital outputs
1001h	Amplifier Enable register	Also includes the DAC output enable
1002h	Reset Monitor register	See Section 2.3.9
1003h	Reserved	
1004h	Watchdog register	See Section 2.3.6
1005h	Reserved	
1006h - 100Fh	Reserved	
1010h - 101Fh	DC-1000	Includes the SSI clock, resolution, and absolute position registers
1020h - 10CFh	None	Available for use with custom peripherals over Extension Con
10D0h - 10DAh	Reserved	Used by Prodigy FPGA
10DBh	Build register	Build number of Prodigy FPGA
10DCh - 10DFh	Reserved	Used by Prodigy FPGA
10E0h - 10EFh	None	Available for use with custom peripherals over Extension Con
10F0h - 10FDh	Reserved	
10FEh	Model type	For compatibility with older Magellan Motion Card family
10FFh	Card ID	See Section 2.3.10

See the *Magellan Motion Processor User's Guide* for more information on peripheral memory space.

This page intentionally left blank.

4. Interconnect Modules

In This Chapter

▶ IM-1000 Interconnect Module
▶ DC-1000 SSI Option Board

4.1 IM-1000 Interconnect Module

The IM-1000 is an interconnect module which assists in the set up and configuration of the Prodigy-PCI Motion Card, and also provides complete options for external connections. All wiring to and from the Prodigy-PCI Motion Card, the amplifiers, power supplies, IOs and encoder feedback may easily be connected to this central point.
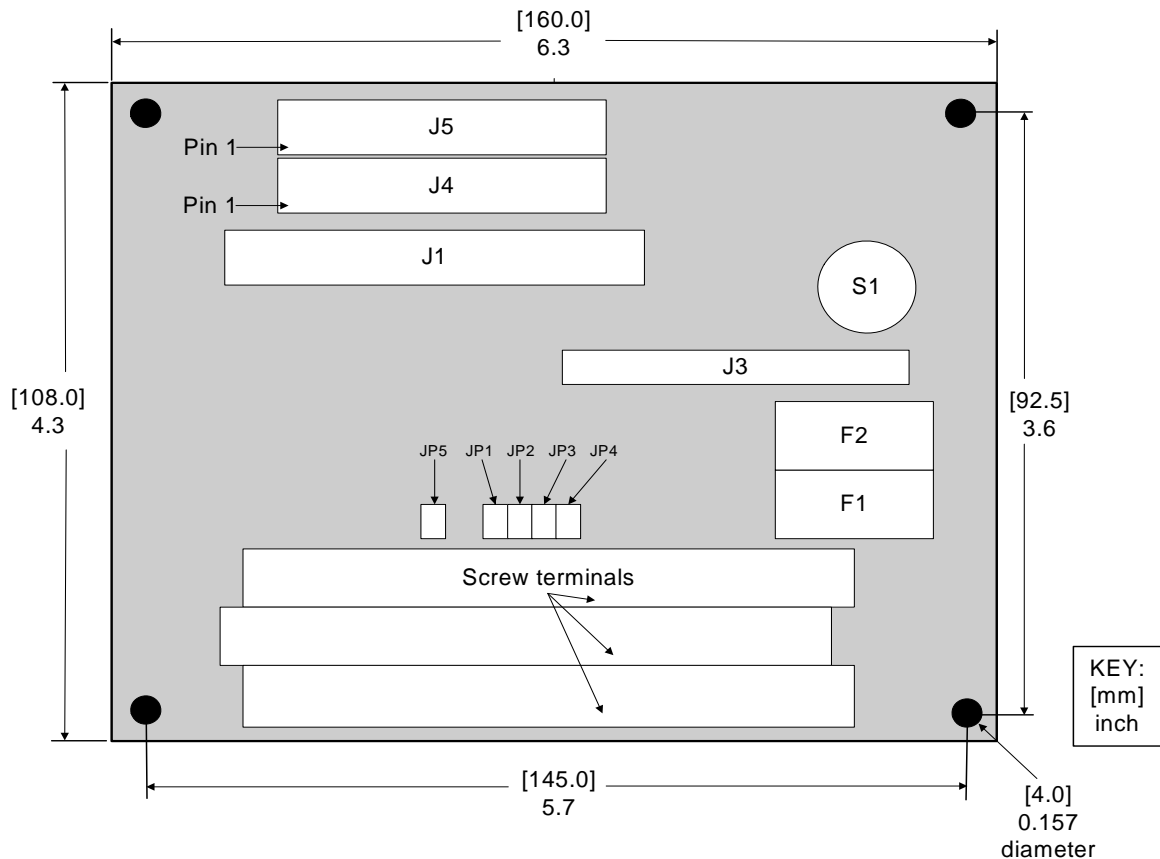
Figure 4-1: IM-1000 location of components

Connector	Description
F1	1 Amp fuse for power supply
F2	1 Amp fuse for power supply
J1	100-position connector
J3	OPTO22 connector
J4	50-position connector
J5	50-position connector
S1	External hardware reset button
Screw terminals	(3) 100-terminal connection blocks for connecting to one 100-position, or two 50-position cables.

The enclosure of the board is a standard Phoenix DIN rail mounting system. Alternatively, the enclosure may be re-moved, and the board mounted to other systems via the 3.5 mm mounting holes located on the corners of the board.

The pinout descriptions that follow provide a detailed illustration of the IM-1000's connections. Please observe the differences between the DC brush and brushless DC and stepping motor versions. The screw terminal numbers cor-respond to the pinout description for the Prodigy-PCI Motion Card in the following way:

PCI	IM-1000	Screw Terminals
J6	J1	1-100

The following table lists the functions of the remaining screw terminals:

Screw Terminals	Function
101, 103	5V from J1 through 1 Amp fuse F2
102, 104, 105	GND
106	External power in
107	GND
108	Unused

The Prodigy-PCI Motion Card supplies 5V DC from the computer to pins J6-7, J6-25, J6-57, and J6-75 through a 1 Amp fuse to power encoders. When the jumpers JP1, JP2, JP3 and JP4 are installed 1-2, this 5V is passed on to screw terminals 7, 25, 57, and 75, respectively. Alternatively, it is also possible to use an external power supply in the range of 5 - 12 VDC for the encoders. To do this, intall jumpers JP1, JP2, JP3 and JP4 on pins 2-3 and connect the external power to screw terminal 106 and its ground to 107. This external power will be limited to 1 Amp by fuse F1.

Connections to J3 (the OPTO 22 connector) are as follows:

J1	J3	Description
	49	5V if JP5 jumper is installed
36	47	DigitalIn0
37	45	DigitalIn1
38	43	DigitalIn2
39	41	DigitalIn3
86	39	DigitalIn4
87	37	DigitalIn5
88	35	DigitalIn6
89	33	DigitalIn7
41	31	DigitalOut0
42	29	DigitalOut1
43	27	DigitalOut2
44	25	DigitalOut3
91	23	DigitalOut4
92	21	DigitalOut5
93	19	DigitalOut6
94	17	DigitalOut7
40	15	AmpEnable1
45	13	AmpEnable2
90	11	AmpEnable3
95	09	AmpEnable4
13	07	AxisOut1
30	05	AxisOut2
63	03	AxisOut3
80	01	AxisOut4
	All even-numbered pins	GND

4.2 DC-1000 SSI Option Board

The DC-1000 is an optional expansion board for the Prodigy-PCI Motion Card. The DC-1000 is a Synchronous Serial Interface (SSI) for absolute encoders, and connects to the Prodigy-PCI Motion Card's parallel input connector (see Figure 4-1). The data of the absolute encoders is read through the DC-1000 interface, and is then transmitted to the Prodigy-PCI's motion controllers. The electronics of the absolute encoder converts the parallel-format data into serial-format data, and transmits this data to the DC-1000. The DC-1000 converts the incoming data into a 16-bit data word, from Gray code to binary code, and transmits the data to the Prodigy-PCI Motion Card. The DC-1000 can accept connections of up to four axes, and the data of each axis can be programmed for a read rate of up to 20 kHz.

Some of the major features of the DC-1000 are as follows:

- Up to four axes can be read

- Resolution can be programmed for 10, 12, 13, or 25 bits

- Compatible with single- and multi-turn absolute encoders

- Programmable frequencies of 1.1 MHz, 550 kHz, 275 kHz, and 137.5 kHz

- Default jumper settings can be overwritten with software commands

- Update rates of up to 20 kHz for all four axes

- Automatic recognition and addition of position overflow

- Index search not required

- Absolute position is read at power-on

- Incremental and absolute encoders can be mixed in any combination

- Real-time switching between incremental and absolute modes

- Compatible with any absolute encoders which support the Synchronous Serial Interface (SSI), such as Haidenhein, Stegmann, Thalheim

Unlike incremental encoders, optical encoders always supply the exact absolute position. At power-on, the encoder will report its exact position. This precludes the need for the machine to locate a machine reference. Absolute encoders are grouped into two categories: single-turn and multi-turn. Encoders with a resolution of up to 360 degrees are single-turn encoders. Those which can resolve several turns (normally up to 4,096) are classified as multi-turn encoders. The key components of an absolute encoder consist of an optical code disk, an opto-electronic sensor system, and pulse-shaping circuitry and amplification.

The code discs of most absolute encoders generate position data in Gray Code. This differs from binary and BCD code, as Gray code guarantees higher reliability for increased accuracy of data transmission. The requirements for the transmission of absolute position data is one data line for each bit of information. An encoder with 13 bits of resolution would therefore require 13 data lines; an encoder with 25 bits of resolution would require 25 data lines. A system with four encoders would require 100 (4 x 25) data lines to transmit the data to the controller. The SSI interface requires only four data lines, regardless of the encoder's resolution. This is accomplished by converting the parallel data into serial data. Each bit of the parallel data is the sent on the data lines +/- synchronously to the +/- clock lines. The clock speed of the DC-1000 is programmable between 1.1 MHz and 137.5 kHz. The clock speed depends on the length of the data cables between encoder and controller.

The DC-1000 converts the received Gray code to binary code serial data, and stores it in parallel format in registers. This data can then be read by the Prodigy-PCI Motion Card. For a system with four absolute encoders and a cable length of up to 50 meters, the position of every encoder is read 20,000 times per second by the Prodigy-PCI Motion Card. This guarantees timely and accurate data. The DC-1000 can be programmed for encoder resolutions of 10, 12, 13 and 25 bits. The transmission speed may be programmed to accommodate cable lengths of greater than 300 meters, and both single-turn encoders and multi-turn encoders may be employed. The Prodigy-PCI Motion Card will capture the absolute position of single-turn encoders for a full 32 bits. When a roll-over occurs, it will be detected by the Prodigy-PCI Motion Card and the correct position will be stored. The direction of motion will also be correctly detected.

4.2.1. DC-1000 Specifications

- 40-position mating connector for the Prodigy-PCI Motion Card

- 2 stand-offs with screws for mounting to Prodigy-PCI Motion Card

- Encoder mating connector for 5V DC or 12V DC from PC power supply

- Jumper for selecting 5V or 12V supply voltage for the encoders

- 26-position IDC connector and mating stub IDC ribbon cable

- DIP-switches for setting encoder resolution for each axis

- Four position registers (one for each axis) maintain absolute positions

- All positions may be read by the Prodigy-PCI Motion Card at any time

- Four read and write registers (one for each axis) for programmatically setting resolution and clock speed

- Resolution and clock speed settings can be different for every axis

The following diagram illustrates the location of the DC-1000's main components and connectors. The component side of the board is shown.

Figure 4-2: DC-1000 location of components
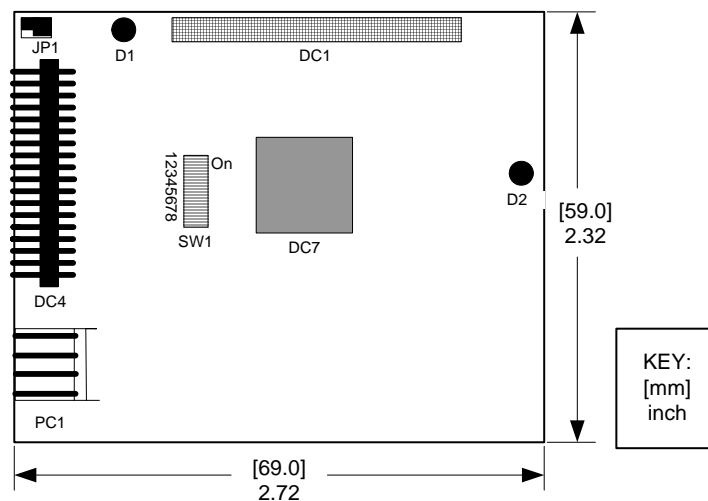
The names and descriptions of the main components of the DC-1000 are detailed in the following table:

Label	Name	Description
DC1	Parallel connector	This socket connects to the parallel input connector of the Prodigy-PCI board.
D1, D2	Mounting holes	These holes (2) are in direct alignment with the mounting holes on the Prodigy-PCI board, and are used for the mounting screws and stand-offs.
JP1	Jumper	This jumper selects either 5V or 12V supply voltage. 5V DC is the default setting.
DC4	Encoder connector	The encoders are connected to this pin block.
PC1	Power connector	Use this pin block to connect a small Molex connector for DC power.
SW1	DIP switches	This block of DIP switches is used to set encoder resolution for each axis.
DC7	PLD	This is the processor for the DC-1000.

4.2.2. DC-1000 Connections

The pinouts for connector DC4 are outlined in the following table:

IDC Pin No.	Description	Axis
1	+5 or +12 VDC to Encoder	X
2	Clock 1+	X
3	Clock 1-	X
4	GND	X
5	DIN 1+	X
6	DIN 1-	X
7	+5 or +12 VDC to Encoder	Y
8	Clock 2+	Y
9	Clock 2-	Y
10	GND	Y
11	DIN 2+	Y
12	DIN 2-	Y
13	No connection	
14	No connection	
15	+5 or +12 VDC to Encoder	Z
16	Clock 3+	Z
17	Clock 3-	Z
18	GND	Z
19	DIN 3+	Z
20	DIN 3-	Z
21	+5 or +12 VDC to Encoder	W
22	Clock 4+	W
23	Clock 4-	W
24	GND	W
25	DIN 4+	W
26	DIN 4-	W

The settings for the jumper (labeled JP1 in Figure 4-2) are outlined in the following table:

Pins Jumpered	Setting
1 - 2	12V DC
2 - 3	5V DC (default)

4.2.3. DC-1000 Default Parameters

At power-on or after a reset, the system clock will be set to 1.1 MHz, and the default system resolution will be determined by the settings of the DIP switches. Additional parameters are defined in the following tables:

Switches (0…1)	Resolution
00	10 bits
01	12 bits
10	13 bits
11	25 bits

For the DIP switches, ON = 1; OFF = 0. Default settings are all DIP switches set to OFF (0).

DIP Switches	Axis
1 - 2	X-axis
3 - 4	Y-axis
5 - 6	Z-axis
7 - 8	W-axis

NOTE: These values may be overwritten by software.

4.2.4. Installing the DC-1000

The DC-1000 attaches directly to the Prodigy-PCI Motion Card, with the component sides of both boards facing each other. The DC-1000's parallel input socket DC1 connects to the Prodigy-PCI board's Extension Con connector J1 (see Figure 1-1).

1 To mount the DC-1000 to the Prodigy-PCI, remove the Prodigy-PCI Motion Card from the computer (if installed). Orient the card as shown in Figure 4-3 with the GP Con connector (J6) on the left.

2 Position the DC-1000's parallel connector socket (DC1) over the Prodigy-PCI board's Extension Con. Ensure that the right edge of the mating connectors are aligned. The DC1 connector socket is shorter than the Prodigy-PCI board's Extension Con, and when the right edges of both are properly aligned, three sets of pins on the Extension Con will be visible at the left of the DC-1000's connector when viewed from above the boards. Before pressing the DC-1000 into place, view the boards from the front in the same orientation as Figure 4-3. Verify that the two sets of mounting holes are in exact alignment.

3 When this alignment is correct, press firmly and evenly to seat the pins of the Prodigy-PCI board's Extension connector into the sockets of the DC-1000's parallel connector.

4 Place the two standoffs between the boards so that they are aligned with the two mounting holes, and insert the mounting screws through the mounting holes and standoffs.

5 Once the boards are mated, attach the supplied IDC cable to the encoder connector on the DC-1000. Pin 1 of the connector is located nearest the JP1 jumper, and the cable's connector must be oriented so that the red-striped wire of the IDC cable attaches to pin 1. The red marking on the cable will then be located toward the top of the boards.

6 To connect DC power to the DC-1000, attach a small Molex plug from the computer's power supply to the 4-pin white connector on the DC-1000 (labeled PC1 in Figure 4-2). These are the small power connectors, which are used to connect a floppy drive.

The DC-1000 mounted to the Prodigy-PCI Motion Card is illustrated in the following figure:
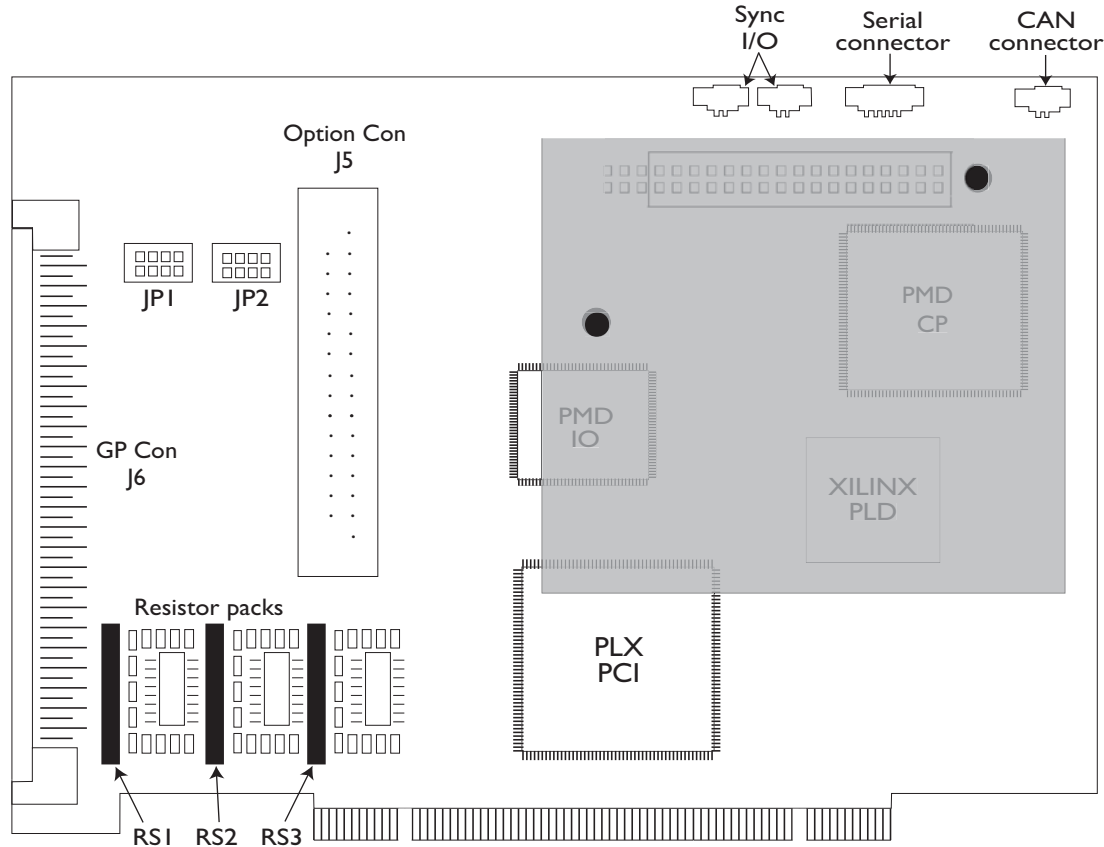
**Figure 4-3:
DC-1000
mounted on
Prodigy-PCI**

4.2.5. DC-1000 Software Commands

The DC-1000 may be operated in mixed-mode. Any or all of the four axes may be configured in any combination of incremental or absolute modes. The following commands are required to configure the absolute encoder mode for the DC-1000.

    SetEncoderSource Axis n, mode   //Sets the mode to 0 = incremental or1 = absolute for Axis n

Setting the mode to absolute (parallel) deactivates the incremental mode and setting to incremental deactivates the absolute (parallel) mode.

    SetEncoderModulus Axis n, modulus   //Sets the modulus (0 to 65535) for Axis n

Modulus settings are required for single-turn absolute encoders in order to capture the full position when the encoder exceeds a full turn. The Prodigy-PCI Motion Card will accumulate the full 32 bits for the position. The modulus must always be one-half of the encoder's resolution. Assuming an encoder with 12 bits (4,096), the resulting modulus to be programmed would be 2,048.

One more command is required to program the SSI clock and resolution on the DC-1000.

    SetSSIRegister Axis n, resolution, frequency   //This function sets the resolution and clock frequency for Axis n

Resolution can be set to one of 4 values: 10 bits, 12 bits, 13 bist or 25 bits. The frequency can also be set to one of 4 values: 1.1 MHz (default), 550 kHz, 275 kHz or 137.5 kHz. Each axis can be programmed independently.

See the *Magellan Motion Processor Programmer's Command Reference* for more information.

This page intentionally left blank.

Index

Numerics

100-terminal connection blocks 45
26-position IDC connector 48
40-position mating connector 48
50-position connector 45

A

absolute encoder 47
absolute mode. 51
Accessory 11
Algorithmic-based initialization command sequence 18
AmpEnable 16
amplifier enable output signals 24
Analog input range 43
Applying Power 16
axis verification 17
AxisIn 30
AxisOut 30

B

BCD code 47
Breakout interconnect module 11
Brushless DC 10
Brushless DC Motor Connections 42
Brushless DC Motor connections 14
brushless DC motors 14

C

Cable 11
CAN 2.0B network 26
CAN 2.0B nodes 26
CAN Connector 47
CAN serial communication physical layer 41
CANH connection 40, 41
CANL connection 41
Card Function Overview 21
Card-Specific Functions 21
change mask bits 24
Check Commutation 19
C-Motion 10, 12
C-Motion features 10
commutation 19
Connection Summary 14
Connections Summary 41
Connectors 35

This page intentionally left blank.

For additional information, or for technical assistance,
please contact PMD at (978) 266-1210.

You may also e-mail your request to support@pmdcorp.com

Visit our website at http://www.pmdcorp.com