ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Diamond Systems Hercules HRC550-5A128* at our website: *Click HERE*

Navigator®-PC/104

User's Guide



Performance Motion Devices, Inc.
55 Old Bedford Road
Lincoln, MA 01773

Revision 1.0, August 2003

NOTICE

This document contains proprietary and confidential information of Performance Motion Devices, Inc., and is protected by federal copyright law. The contents of this document may not be disclosed to third parties, translated, copied, or duplicated in any form, in whole or in part, without the express written permission of Performance Motion Devices, Inc.
The information contained in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form, by any means, electronic or mechanical, for any purpose, without the express written permission of Performance Motion Devices, Inc.

Copyright 1998, 1999, 2000, 2001, 2002, 2003 by Performance Motion Devices, Inc.
Navigator®, Pro-Motion® and C-Motion® are registered trademarks of Performance Motion Devices, Inc.

Warranty

Performance Motion Devices, Inc. (PMD) warrants performance of its products to the specifications applicable at the time of sale in accordance with Performance Motion Devices, Inc.'s standard warranty. Testing and other quality control techniques are utilized to the extent Performance Motion Devices, Inc. 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. 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 Performance Motion Devices, Inc. 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

Performance Motion Devices, Inc. assumes no liability for applications assistance or customer product design. Performance Motion Devices, Inc. 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 Performance Motion Devices, Inc. covering or relating to any combination, machine, or process in which such products or services might be or are used. Performance Motion Devices, Inc.'s publication of information regarding any third party's products or services does not constitute Performance Motion Devices, Inc.'s approval, warranty or endorsement thereof.

Related Documents

Navigator Motion Processor User's Guide (MC2000UG)
How to use all members of the Navigator Motion Processor family.

Navigator Motion Processor Programmer's Command Reference (MC2000PR)
Descriptions of all Navigator Motion Processor commands, with coding syntax and examples, listed alphabetically for quick reference.

1.0 Installation 7
1.1 Software 8
1.2 Accessory Products 9
1.3 Documentation 9
1.4 Installation Sequence 9
1.5 Required Hardware 10
1.6 Preparing the Card for Installation 11
1.7 Connection Summary for Navigator-PC/104 Cards 12
1.7.1 DC Brush Motors 13
1.7.2 Brushless DC Motors 13
1.7.3 Pulse & Direction Motors 14
1.7.4 Microstepping Motors 14
1.8 Applying Power 15
1.9 Software Installation 15
1.10 First Time System Verification 15
1.10.1 Step #1: Set the Motor Amplifier Type 16
1.10.2 Step #2: Initialize the Commutation 17
1.10.3 Step #3: Check Commutation 18
1.10.4 Step #4: Set Filter Parameters 19
1.10.5 Step #5: Set the Motor Command 19
1.10.6 Step #6: Make a Trajectory Move 20

2.0 Operation 21
2.1 Card Function Overview 22
2.2 Navigator Motion Processor 22
2.3 Card Specific Functions 23
2.3.1 General Purpose Digital I/O 23
2.3.2 Amplifier Enable 24
2.3.3 DAC Output Enable 26
2.3.4 Serial Transceiver 27
2.3.5 Watch Dog Timer 28
2.3.6 Under Voltage Monitor 28
2.3.7 Reset 29
2.3.8 Reset Monitor 30
2.3.9 Card ID 31
2.4 Signal Processing and Hardware Functions 32
2.4.1 Home, AxisIn, AxisOut, Limits, Hall Sensors 32
2.4.2 QuadA, QuadB, Index 32
2.4.3 Analog Input 33
2.4.4 Pulse and Direction 34
2.4.5 PWM Out 34
2.4.6 Motor Command 34

3.0 Using Pro-Motion 35
3.1 Communication 36
3.1.1 Axis Setup Wizard 37
3.2 Motion Configuration 38
3.2.1 Step #1: General Settings: Amplifier Type 38
3.2.2 Step #1a: General Settings: Time Units 39
3.2.3 Step #1b: General Settings: Scale Settings 39
3.2.4 Step #2: Initialize Signal Sensing 40
3.2.5 Step #3: Initialize Encoders 40

3.2.6 Step #4: Initialize Commutation 41
3.2.7 Step #4a: Check Commutation 41
3.2.8 Step #5: Initialize Stepper Parameters 42
3.2.9 Step #6: Initialize Microstepper Parameters 42
3.2.10 Step #7: Set Servo Loop Parameters 43
3.2.11 Step #8: Tracking Parameters 44
3.2.12 Step #9: Motion Dynamics Settings 44
3.2.13 Step #10: Initiate Motion Shuttle (optional) 45
3.3 Command Window 45
3.3.1 Commands Available within the Command Window 47
3.4 Scope Window 48
3.4.1 Scope Control Bar 48
3.4.2 Scope Window Settings 49
3.4.3 Interactive Scope Features 50
3.4.4 Printing Scope Data 50
3.4.5 Exporting Scope Data 50

4.0 Developing Your Own Applications with C-Motion 51
4.1 Overview 51
4.2 Using C-Motion 52

5.0 Navigator-PC/104 Electrical Reference 55
5.1 User-Settable Components 55
5.1.1 Switch S1 56
5.1.2 Mode Jumper 57
5.1.3 Resistor Packs RS1, RS2, RS3 57
5.2 Connectors 57
5.2.1 J1 and J7 Connector 57
5.2.2 J1 and J7 using DC Brush Cards 58
5.2.3 J1 and J7 using Brushless DC Cards, Microstepping Cards, and Mixed Motor Cards 59
5.2.4 J1 and J7 using Pulse & Direction Cards 60
5.2.5 Option Con Connector 61
5.2.6 Option Con using Brushless DC and Mixed Motor Cards 61
5.2.7 Option Con using Microstepping Cards 62
5.2.8 Serial I/F Connector 63
5.3 Connections Summary - Motor Amplifiers 63
5.3.1 DC Brush Motor Connections 64
5.3.2 Brushless DC Motor Connections 64
5.3.3 Microstepping Motor Connections 65
5.3.4 Pulse & Direction Motor Connections 65
5.4 Command Summary - ReadIO, WriteIO 66
5.4.1 General Purpose Digital I/O Control Register (Address +0) 66
5.4.2 Amplifier & DAC Enable Control Register (Address +1) 66
5.4.3 Reset Monitor Control Register (I/O Space Address +2) 67
5.4.4 Card ID Control Register (Address +0xFF) 67
5.5 Command Summary - Card-Specific Functions 68
5.6 Environmental and Electrical Ratings 69

1.0 Installation

In This Section

- Software
- Accessory Products
- Documentation
- Installation Sequence
- Required Hardware
- Preparing the Card for Installation
- Connection Summary for Navigator-PC/104 Cards
- Applying Power
- Software Installation
- First Time System Verification

The PMD Navigator-PC/104 cards are high performance PC/104-bus cards that provide motion control for DC brush, brushless DC, and step motors. These cards are based on PMD Navigator motion processors which perform motion command interpretation and many other real time functions.

The following product selector table shows the relationship between card part numbers (P/N), Navigator motion processor part numbers (P/N), the number of axes supported, and the type of motors supported.

Navigator-PC/104 card P/N	Navigator P/N	Number of axes	Motor type
MB802140	MC2140	4	DC brush
MB802120	MC2120	2	DC brush
MB802110	MC2110	1	DC brush
MB802340	MC2340	4	brushless DC
MB802320	MC2320	2	brushless DC
MB802310	MC2310	1	brushless DC
MB802540	MC2540	4	pulse & direction step motor
MB802520	MC2520	2	pulse & direction step motor
MB802510	MC2510	1	pulse & direction step motor
MB802440	MC2440	4	microstepping step motor
MB802420	MC2420	2	microstepping step motor
MB802410	MC2410	1	microstepping step motor
MB802840	MC2840	4	mixed motor - DC brush and brushless DC or microstepping
MB802820	MC2820	2	mixed motor - DC brush and brushless DC or microstepping

DC brush cards output 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 cards provide multi-phase motor command signals, using Hall-based or sinusoidal commutation located on the Navigator-PC/104 card. These cards are intended to interface with brushless DC amplifiers and motors, although they can also control other 3-phase devices such as AC Induction motors.

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

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

Mixed motor cards provide the ability to control DC-brush, brushless DC or microstepping motors within the same card. The output formats are software programmable, allowing various combinations of motor types and axis number to be configured.

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

1.1 Software

Two major software packages are provided with the Navigator-PC/104 cards: Pro-Motion, an interactive Windows-based exerciser program and C-Motion, a C-language library which simplifies the development of motion applications for Navigator-PC/104 cards.

Pro-Motion is a sophisticated, easy to use exerciser program that allows you to set and view all card parameters, and exercise all card features. Pro-Motion features include:

- Motion oscilloscope graphically displays processor parameters in real-time
- Interactive servo tuning
- Project window for accessing card 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
- Command window for direct text command entry. It also serves as a communications monitor that echoes all commands sent by Pro-Motion to the card.

C-Motion provides a convenient set of callable routines that comprise all of the code required for controlling your Navigator-PC/104 card. C-Motion includes the following features:

- Axis virtualization
- The ability to communicate to multiple Navigator-PC/104 cards
- The ability to link easily to any "C/C++" application

Pro-Motion is described in *section 3, Using Pro-Motion, page 35,* C-Motion is described in detail in *section 4, Developing Your Own Applications with C-Motion, page 51.*

1.2 Accessory Products

The Navigator-PC/104 cards can be enhanced by a number of hardware accessory products, listed below:

Component part number	Description
Cable-2003	50 position, 3-foot long ribbon cable to connect Navigator-PC/104 card to IM-1000 interconnect module.
IM-1000	Breakout interconnect module that provides convenient jack-screw type terminators for the 50 pin cable. Used with cable-2003.
DC-1000	Parallel encoder input adaptor. This daughter card module allows parallel-word and other encoders which use the SSI interface format to be directly connected.
Cable-4003	3-foot long serial port cable which connects to a serial I/F connector. This cable allows serial communication to the PC/104 card. Only used if serial port communications are desired.

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

1.3 Documentation

There are three manuals specifically associated with the Navigator-PC/104 cards. A brief description of each is listed below.

Component part number	Name	Description
MB80XUG	Navigator-PC/104 User's Guide	This is the first document you will use to get started, make connections, and exercise the card. This document will answer questions such as "How do I connect my amplifiers and motors to the card?" and "What jumper options do I set to use the card?" This document also provides a complete electrical description of the card, and a description of the associated software products Pro-Motion and C-Motion.
MC2000UG	Navigator Motion Processor User's Guide	This is a functional description of the Navigator motion processor, which is the chipset that is at the heart of the Navigator-PC/104 cards. This document will answer questions such as, "How do I make a trapezoidal move?" or "How do I load servo parameters."
MC2000PR	Navigator Motion Processor Programmer's Command Reference	This document provides a complete listing of all motion commands supported by the Navigator motion processors.

To download these documents, or request that they be sent to you, you can visit the PMD website at www.pmdcorp.com or contact your PMD representative.

1.4 Installation Sequence

In this manual it is assumed that development of your PC/104-based motion application will occur using a Windows-based PC with a PC/104 bus, or with an ISA to PC/104 converter device. Such converters are available from a number of vendors.

For a normal installation of a Navigator-PC/104 card, you will need to configure your card for the hardware/bus system and motor hardware that you will connect it to. Configuration of the Navigator-PC/104 cards is described in detail in *section 1.6, Preparing the Card for Installation, see page 11.*

Next you will need to connect your system's motors, encoders, amplifiers, and sensors as desired to operate your motion hardware. A description of the connections that are made for the various Navigator-PC/104 cards is found in *section 1.7, Connection Summary for Navigator-PC/104 Cards, see page 12*.

Once this hardware configuration is complete, you should then install the software. Installation of the software is described in *section 1.9, Software Installation, see page 15*.

The final step to finish the installation is to perform a functional test of the finished system. This is described in *section 1.10, First Time System Verification, see page 15*.

Once all of the above has been accomplished installation is complete, and you are ready to operate the card.

1.5 Required Hardware

To install a Navigator-PC/104 card and run Pro-Motion, the Windows-based exerciser program, you will need the following hardware:

1. The recommended platform is an Intel (or compatible) processor, Pentium or better, one available PC/104 slot, 5MB of available disk space, 32MB of available RAM, and a CDROM drive. The PC operating system required is Windows 9X/ME/NT/2000/XP.
2. 1 to 4 pulse and direction, PWM, or analog-input amplifiers. The type of amplifier depends on the card type you are using.
3. 1 to 4 step motors or servo motors. These motors may or may not provide encoder position feedback signals depending on the type of card being used. Servo motors must have encoder feedback, while step motors can optionally do so.
4. Additional connectors as required to connect the Navigator-PC/104 card to the amplifiers and the motors. Two 50-pin dual-header-type connectors will be needed to interface to the card. See *section 5.3, Connections Summary, page 63,* for more information on setting up your connections.

1.6 Preparing the Card for Installation

The following diagram shows the location of the jumpers, resistor packs and connectors for the PC/104 cards:

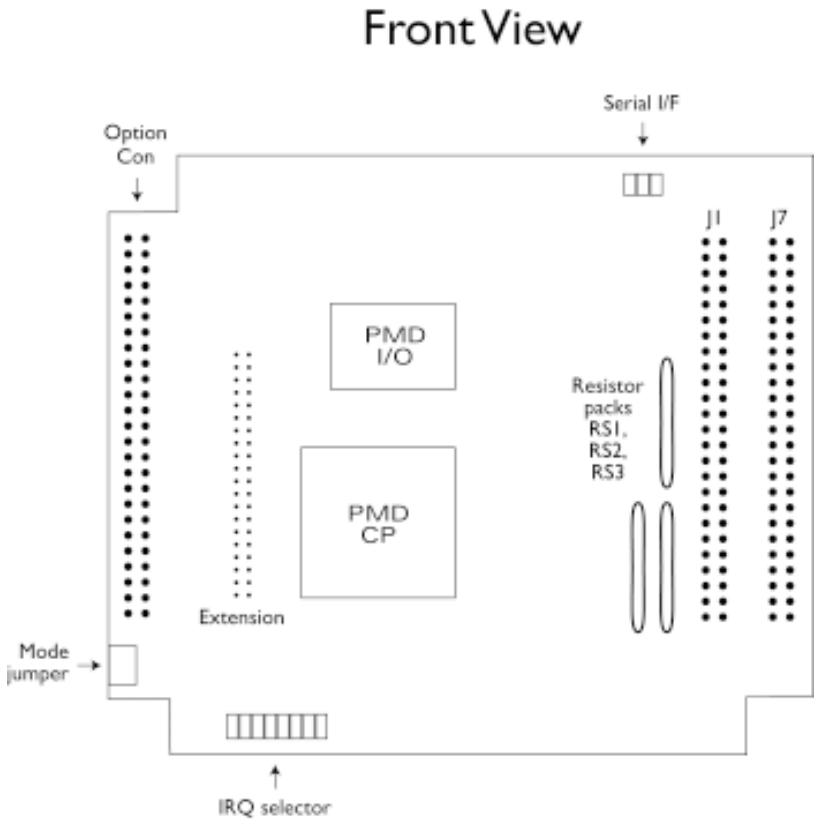


Figure 1-1a. Location of various board elements, front view

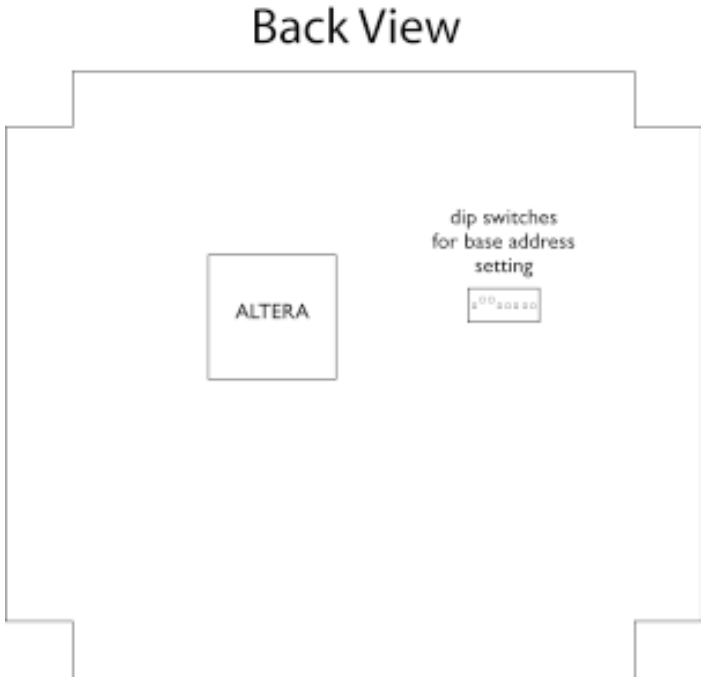


Figure 1-1b. Location of various board elements, back view

To prepare your card for installation the following user-settable hardware option should be checked:

Item	How to set	Description
Switch S1	S1-1 8 (hex) S1-2 10 (hex) S1-3 20 (hex) S1-4 40 (hex) S1-5 80 (hex) S1-6 100 (hex) S1-7 200 (hex) S1-8 400 (hex) *300 (hex) is the default base address*	Switch S1 sets the card address on the PC/104 bus. The selected card address is the additive value of the switch settings indicated to the left. Moving the switch to the 'off' position adds that value to the final base address. For example, to select a base address of 340 (hex), the following switch settings would be selected: S1-1 on S1-2 on S1-3 on S1-4 off S1-5 on S1-6 off S1-7 off S1-8 on The card address selected using S1 must match the address expected by your software. In addition, this address must not be used by other cards. From the base address selected, 8 addresses are used by the Navigator-PC/104 card. For example, if 300 (hex) is selected, then locations 300 - 307 will be used by the card, and cannot be used by other cards on the bus.
Resistor packs RS1, RS2, RS3	Installed *this is the default setting*	If you are using differential connections leave these resistor packs installed. *See section 2.4.2, QuadA, QuadB, Index, page 32,* for more information on encoder inputs.
	Removed	If you are using single-ended encoder connections, remove the resistor packs. *See section 2.4.2, QuadA, QuadB, Index, page 32,* for more information on encoder inputs.

1.7 Connection Summary for Navigator-PC/104 Cards

The following sections summarize the connections you should make for various motor types. Generally, the motor type you will install is specified by the card type purchased. *See table, page 7.* However with the mixed-motor cards (part number MB802820, MB802840) both brushless DC, DC brush and microstepping motors can be connected to the same card.

1.7.1 DC Brush Motors

The following table summarizes connections to the Navigator-PC/104 card when DC-brush motors are used. Depending on the specific card that you ordered, between 1 and 4 axes should be connected. All connections are made through J1 and J7, the two 50-pin header connectors indicated in *figure 1.1, page 11*. For a detailed list of connections, *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

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 analog output used)*	Analog out (DAC output)
Other control signals: *(optional, per axis)*	Home signal input limit switch inputs AxisIn input AxisOut output
Miscellaneous signals:	Digital GND +5 V (for encoder power)

1.7.2 Brushless DC Motors

The following table summarizes connections to the Navigator-PC/104 card when brushless DC motors are used. Depending on the specific card that you ordered, between 1 and 4 axes should be connected. Note that some of these connections are made through J1 and J7, the two 50-pin header connectors indicated in *figure 1.1, page 11,* and some are made through Option Con, the other 50-pin connector. For detailed signal descriptions *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

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 analog output used)	Analog out (phase A) Analog 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:	GND +5 V (for encoder power)

1.7.3 Pulse & Direction Motors

The following table summarizes connections to the Navigator-PC/104 card when pulse & direction interface step motors are used. Depending on the specific card that you ordered, between 1 and 4 axes should be connected. All connections are made through through J1 and J7, the two 50-pin header connectors, indicated on *figure 1-1, see page 11*. For detailed signal descriptions *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

Signal category	Signal description
Encoder input signals: (optional, 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:	GND +5 V (for encoder power)

1.7.4 Microstepping Motors

The following table summarizes connections to the Navigator-PC/104 card when microstepping-interface step motors are used. Depending on the specific card that you ordered, between 1 and 4 axes should be connected. Note that some of these connections are made through J1 and J7, the two 50-pin header connectors indicated in *figure 1.1, page 11,* and some are made through Option Con, the other 50-pin connector. For detailed signal descriptions *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

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 magnitude (phase A) PWM magnitude (phase B) PWM direction (phase A PWM direction (phase B)
Amplifier output signals: (per axis, if PWM 50/50 used)	PWM magnitude (phase A) PWM magnitude (phase B)
Amplifier output signals: (per axis, if analog output used)	Analog out (phase A) Analog out (phase B)
Other control signals: (optional, per axis)	Home signal channel input Positive limit switch input Negative limit switch input AxisIn input AxisOut output
Miscellaneous signals:	GND +5 V (for encoder power)

1.8 Applying Power

Once you have installed the Navigator-PC/104 card in your system and 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 card 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 card and check the amplifier and encoder connections. If anomalous behavior is still observed, call PMD for application assistance.

1.9 Software Installation

Locate the CDROM which contains C-Motion and Pro-Motion software. This CD contains software to exercise your board, and source code that will enable you to develop your own motion applications. The exercise software is designed to work with Windows 95/98/ME or Windows NT/2000/XP.

If you have autorun enabled, the installation process will start when you insert the CDROM. The installation program will guide you through installing the software. Upon completion of the installation process, the following components will be installed:

1. Pro-Motion – an application for communicating to and exercising the installed card. Refer to *section 3, Using Pro-Motion*, *page 35* for operating instructions.
2. PMD Board Setup – used to set the base address for communication with PC/104-based cards.
3. C-Motion – source code that can be used for developing your own motion applications based on the Navigator motion processor. Refer to *section 4, Developing Your Own Applications with C-Motion, page 51,* for further information. These files are installed in the "C-Motion" folder, a sub-folder of the installation folder.
4. "PDF" versions of the *Navigator-PC/104 User's Guide*, *Navigator Motion Processor Programmer's Command Reference* and *Navigator Motion Processor User's Guide*. The Adobe Acrobat Viewer is required for viewing these files. If the Adobe Acrobat Viewer is not installed on your computer, you can download it from http://www.adobe.com.

1.10 First Time System Verification

To verify that the Navigator-PC/104 card and Pro-Motion software program have been properly installed, it is useful to have each axis of the system perform a short move. The instructions shown on the following pages are a summary. Before executing these commands you may want to refer to the section of the *Navigator Motion Processor User's Guide* to familiarize yourself with operation of the card's motion processor.

The following table summarizes what is required for each motor type. Note that the step numbers reference specific steps that are detailed in the next section. Perform only the setup step sequences for the card/motor type installed in your system.

Motor type	Step #	Operation
DC brush	1	Set amplifier type (PWM sign/mag, PWM 50/50, DAC)
	4	Set filter parameters
	6	Make a trajectory move
Brushless DC	1	Set amplifier type (PWM 50/50, DAC)
	2	Initialize commutation
	3	Check commutation
	4	Set filter parameters
	6	Make a trajectory move
Microstepping	1	Set amplifier type (PWM 50/50, DAC)
step motor	5	Set the motor output power
	6	Make a trajectory move
Pulse and direction step motor	6	Make a trajectory move

It is assumed that you will initialize each axis of your system one at a time. This being the case, after executing Pro-Motion, type one of the axis setting commands (#axis 1, #axis 2, #axis 3, or #axis 4) in the command window and go through the whole initialization for that axis. To initialize other axes, enter a new axis number and initialize that axis entirely, etc.

For example, if you want to exercise axis #1 you would type in the command **#Axis 1**. All subsequent commands will now be directed to axis 1. This would be followed by an <ENTER> to process the command. Upon successfully accepting this command Pro-Motion will return a prompt ">". If there is some problem with the command, Pro-Motion will indicate the nature of the error, followed with a ">" prompt.

For more information on Pro-Motion, refer to *section 3, Using Pro-Motion, page 35*.

1.10.1 Step #1: Set the Motor Amplifier Type

The motion processor must be told what type of motor output mode to use, PWM sign/mag, PWM 50/50, or DAC. This can be set using the command **SetOutputMode** followed by the output mode; 0 for DAC, 1 for PWM sign/mag, and 2 for PWM 50/50. For example, to specify the output mode as PWM 50/50 the command **SetOutputMode 2** would be typed in.

If the output mode is set to DAC, the DAC outputs need to be enabled as they are disabled after a hard reset. The command **WriteIO 0x8080** enables the DAC output.

If we want to describe the 0x8080 parameter, bit 7 is the DAC enable bit state and bit 15 is the change mask for the DAC enable bit.

1.10.2 Step #2: Initialize the Commutation

NOTE: THIS SECTION APPLIES TO BRUSHLESS OR MIXED MOTOR CARDS ONLY

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

SetCommutationMode 1

No other commands are necessary and you may proceed to step #3.
If you are controlling a brushed type motor, set the number of phases to 1 as follows:

SetNumberPhases 1

No other commands are necessary and you may proceed to step #3.

If you will be commutating using a sinusoidal technique you must initialize the commutation phasing. There are two ways this can be done. You will need to decide whether to initialize using Hall-based or algorithmic methods. See the *Navigator Motion Processor User's Guide* for more information on this.

Each of these two-phase initialization methods requires a separate sequence, as follows (note that // indicates a comment and should not be typed in):

Hall-based initialization command sequence:

```
SetNumberPhases x          // where x is 2 or 3 depending on type of motor
SetPhaseCounts yyyy        // yyyy is # of encoder counts per electrical cycle
SetPhaseInitializeMode 1   // set phase initialize mode to 'Hall-based'
InitializePhase
```

Algorithmic-based initialization command sequence:

```
SetNumberPhases x          // x is 2 or 3 depending on type of motor
SetPhaseCounts yyyy        // yyyy is # of encoder counts per electrical cycle
SetPhaseInitializeMode 0   // set phase initialize mode to 'algorithmic'
SetMotorMode 0             // places axis in open loop mode, required for algorithmic initialization.
SetPhaseInitializeTime zzzz // zzzz is # of motion processor cycles to initialize for
SetMotorCommand wwww       // wwww is motor command.
InitializePhase
```

To determine the values of x, yyyy, zzzz, and wwww you should refer to the *Navigator Motion Processor User's Guide - Commutation Section.*

For more information on Pro-Motion, refer to *section 3, Using Pro-Motion, page 35*.

If your system has one or more of the following conditions present then the above sequence will need to be expanded. To handle such systems you will need to use the **SetSignalSense** command as well as the **SetPhasePrescale** command. Refer to the commutation section of the *Navigator Motion Processor User's Guide* for more information.

1. One or more Hall signals must be inverted to commutate or initialize the commutation correctly
2. Number of encoder counts per electrical cycle exceeds 32,767

1.10.3 Step #3: Check Commutation

NOTE: THIS SECTION APPLIES TO BRUSHLESS MOTOR CARDS AND MIXED MOTOR CARDS.

After phase initialization has been completed it is useful to check the smoothness of the motor rotation in open loop mode to verify that the motor phasing initialization and commutation is correct. To do this use the following command sequence:

```
SetMotorMode 0              // set axis for open loop operation
SetMotorCommand xxxx        // xxxx is the motor command from 0 to 32,767 to output
Update
```

The xxxx value represents the fraction of the value 32,768 of total power that will be applied to the motor. For example a value of 1000 sends roughly 3% (1000/32,768) of the total power to the motor.



When the motor mode is set to off (**SetMotorMode 0**) the motor is not under servo control. Be aware that the motor may spin rapidly after a motor command value is applied. Use small values and increase slowly.

After this command sequence the motor should smoothly spin in one direction or the other. 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 roughly, in the wrong direction, or if it moves a short distance and then abruptly stops there may be a problem with the commutation. Check your wiring and re-test. Once the motor is spinning smoothly in both directions under open loop control, re-enable closed-loop servo control by executing the following command:

```
SetMotorMode 1
```

1.10.4 Step #4: Set Filter Parameters

For motion to occur, some amount of feedback gain must be specified. Initially use just a proportional gain with a very low value between 1 and 25. Later you can add integral or derivative gains 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, making them active.

```
SetKp xxxx                  // xxxx is the desired proportional gain
SetKd yyyy                  // yyyy is the desired derivative gain
SetKi zzzz                  // zzzz is the desired integral gain
SetIntegrationLimit aaaa    // aaaa is the desired integration limit
Update                      // make the values active.
```

It is not necessary to specify all 3 gains. Just Kp, followed by an **Update** can be specified, just a Kd, etc.



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

1.10.5 Step #5: Set the Motor Command

NOTE: THIS SECTION APPLIES TO THE MICROSTEPPING STEP MOTOR CARDS ONLY.

In order for motion to occur, the magnitude of the output must be set. Motor control output is described in detail in the *Navigator Motion Processor User's Guide, section 9.4, Motor Command Output, page 55*. A value between 0 and 32,767 represents an amplitude of 0 – 100%. A value of around 5000 should be satisfactory to start with.

Here is the command sequence to use:

```
SetMotorCommand xxxx        // set the motor output level
Update                      // execute the move
```

1.10.6 Step #6: Make a Trajectory Move

To test that the motor is being driven properly, set up and execute a small trapezoidal move. Specify a small distance of (for example) 5000 counts, and a low velocity and acceleration of (for example) 10,000, and 10 respectively. With a cycle time of 400 μSec, these values correspond to roughly 381 counts/sec, and 954 counts/sec2, respectively.



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

Here is the command sequence to use:

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

After entering this sequence of commands you should see the axis smoothly move for about 15 seconds (if the suggested values are used and the cycle time of the motion processor is 400 μsec) If you do not see the axis moving, or if the axis jumps rapidly in one direction or the other, there may be a problem with the board or software settings. Re-check and review the board setup procedures, as well as the exerciser parameter settings.

If you are still having problems after re-checking your system, call PMD for applications assistance.

2.0 Operation

2

In This Section

- Card Function Overview
- Navigator Motion Processor
- Card Specific Functions
- Signal Processing and Hardware Functions

The PMD Navigator-PC/104 cards are high performance PC/104-bus cards that provide motion control for DC brush, brushless DC, and step motors. These cards are based on PMD Navigator motion processors which perform motion command interpretation and other real time functions.

The overall card function is divided amongst a number of modules. These modules are indicated in the block diagram below:

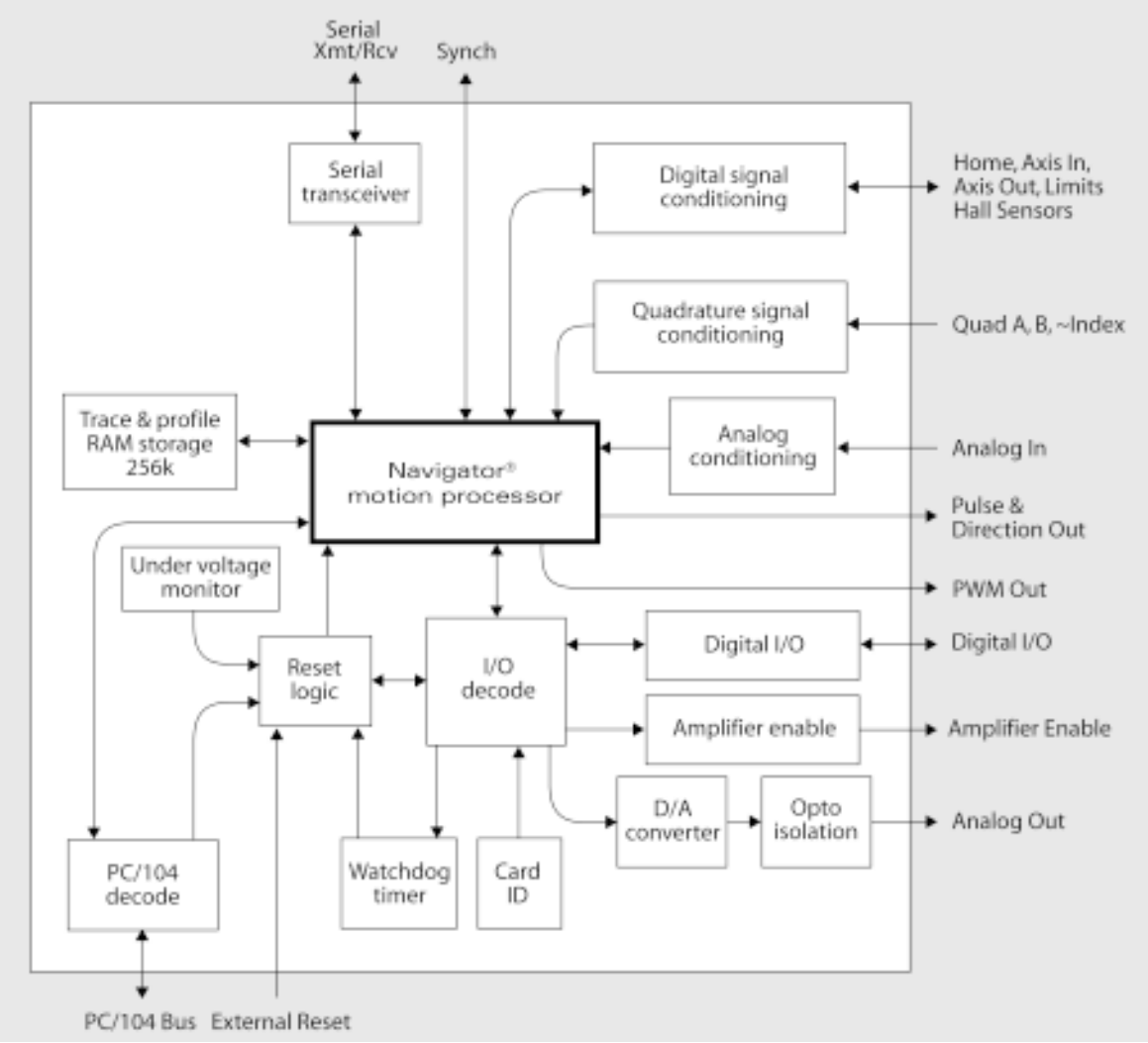


Figure 2-1. Navigator-PC/104 Internal Block Diagram

2.1 Card Function Overview

Navigator-PC/104 card resources can be broken into three overall categories:

Navigator motion processor functions - These are programmable functions which reside in the motion processor such as profile generation, servo loop closure, and much more. These functions are accessed using Navigator motion processor commands, of which there are roughly 150 in total, to allow sophisticated control of the card's overall behavior.

Card-specific functions - These are programmable functions which are controlled by the motion processor using commands **ReadIO** and **WriteIO**, but which reside in various portions of the card circuitry. These functions include general purpose digital I/O, and other card-specific capabilities.

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

2.2 Navigator Motion Processor

The Navigator motion processor pictured in *figure 2.1, the Internal Block Diagram, page 21,* is comprised of 2 ICs, a CP (command processor) and an I/O (input/output) IC. A summary list of the functions provided by the motion processor is as follows:

- Profile generation
- Motor output signal generation (PWM and analog)
- Quadrature counting, index capture
- Servo loop closure (for DC brush or brushless DC motors only)
- Breakpoint processing
- PLC-function processing (AxisIn and AxisOut signals)
- Trace
- Motion error detection, tracking and settle windows indicator
- Limit switches

Access to the Navigator-PC/104 cards may occur through the PC/104 port, or through the serial port. In either case, a complete set of C-Motion function calls, one for each Navigator command, is used to communicate to the card. For a complete list of Navigator commands see the *Navigator Motion Processor Programmer's Command Reference*.

The system on which the Navigator-PC/104 card is installed can control the card through the pre-compiled program Pro-Motion, or through a program of your own construction, using C-Motion calls as the basic interface. During axis setup (see *section 4, Developing Your Own Applications with C-Motion, page 51* for more details) the communications method (PC/104-bus or serial port), and other parameters, are specified which allow C-motion to create a virtualized axis handle, that from then on is the reference for all C-motion commands.

Available C-Motion commands correspond one-for-one with those listed in the *Navigator Motion Processor Programmer's Command Reference*. All C-Motion commands preface the Navigator command with the letters 'PMD', so the Navigator command (for example) **SetPosition** becomes the C-callable routine

PMDSetPosition



Example

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

PMDSetProfileMode(axis_handle, trapezoidal);
PMDSetPosition(axis_handle, position_value);
PMDSetVelocity(axis_handle, velocity_value);
PMDSetAcceleration(axis_handle, acceleration_value);
PMDSetDeceleration(axis_handle, deceleration_value);
PMDUpdate(axis_handle);

Two separate manuals describe how the Navigator motion processor operates and how it is programmed, the *Navigator Motion Processor User's Guide,* and the *Navigator Motion Processor Programmer's Command Reference*.

2.3 Card Specific Functions

Card-specific functions are those functions that are mapped through the motion processor's ReadIO and WriteIO facility, but are implemented in the card circuitry.

Card-specific functions are detailed in this document, the *Navigator-PC/104's User's Guide,* rather than the *Navigator Motion Processor User's Guide*, or the *Navigator Motion Processor Programmer's Command Reference*.

2.3.1 General Purpose Digital I/O

In addition to numerous special-purpose digital signals that are input or output to the card such as ***AxisIn, AxisOut, Home, QuadA,*** etc., the Navigator-PC/104 cards support 8 general-purpose inputs, and 8 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 motion processor's **ReadIO** and **WriteIO** command, the signals present at these various connections do not directly affect the motion processor's behavior. Thus the motion processor simply passes them through to the host.

ReadIO and WriteIO commands
The 8 inputs and outputs are read using the **ReadIO** command and **WriteIO** command, with an I/O address of 0. The table below shows this, along with the bit locations of the input and output signals.

Address	Bit location	Signals
0	0-7	***DigitalOut0-7***
	8-15	***DigitalIn0-7***

To read the 8 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 J1 and J7 connectors, pins 36-39, for those inputs. To write new signal values to the 8 digital outputs, a **WriteIO** command to address offset 0 is sent, and the values on bits 0-7 will be output to the signal connectors at J1 and J7, pins 41-44. The value of bits 8-15 are ignored.



Example

To write a sequence 0xAA to bits 0-7, the C-Motion command (*see section 4, Developing Your Own Applications with C-Motion, page 51,* for more information on C-Motion commands) **PMDWriteIO**(axis_handle, 0, 0xAA) is used. Assuming that the signal pattern 0x55 is present on the 8 input connections, if the command **PMDReadIO**(axis_handle, 0, &load_reg) is used, load_reg will contain 0x55AA. The upper 8 bits reflect the present value of the output signals, while the lower 8 bits reflect the 8-bit value being input.

C-Motion commands

In addition to accessing these signals using the C-Motion commands **PMDReadIO** and **PMDWriteIO**, you can use card-specific C-Motion functions for this purpose. Card specific functions have a 'MB' after the letters PMD. The available card specific functions for this feature are:

C-Motion command	Arguments	Function description
PMDMBWriteDigitalOutput	axis handle, write_value	This function writes to the 8 general-purpose digital I/O signals (digitalOut0-7). Write_value holds the 8 signals in its low order 8 bits.
PMDMBReadDigitalInput	axis_handle, read_value	This function reads the value of the signals DigitalIn0-7 and returns them in the low order 8 bits of read_value.
PMDMBReadDigitalOutput	axis_handle, read_value	This function reads the value of the signals DigitalOut0-7 and returns them in the low order 8 bits of read_value.

Connections & associated signals.
The general-purpose I/O signals are direct single-signal digital inputs and outputs. There are no associated connections that need to be made for these signals to function properly, however one or more of the digital grounds must be connected. The default value, upon powerup, for all general-purpose digital outputs is low.

For a complete description of the pinout connections to/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55.*

2.3.2 Amplifier Enable

The signals ***AmpEnable1-4*** provide up to four digital outputs which, if desired, can be used as amplifier enable signals, although they can also be used as general-purpose digital outputs. Similar to the general-purpose digital inputs and outputs (*section 2.3.1, page 23*), these signals are not directly affected by the motion processor's behavior, however they can be read or written through the motion processor'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 I/O address of +1. The table below shows this.

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, and the change mask bits corresponding to the signals that will be changed must be loaded at bits 8-11 and the value(s) to be loaded must be loaded in bits 0-3.

Example



To set the amplifier enable outputs to high (1) for axes 1 & 3, the C-Motion command **PMDWriteIO**(axis_handle, +1, 0x505) is used. To then disable the output only for axis 3, the command **PMDWriteIO**(axis_handle, +1, 0x400) is used, and to then disable the output for axis 1, the command **PMDWriteIO**(axis_handle, +1, 0x100) is used.

C-Motion commands

In addition to accessing these signals using the C-Motion commands **PMDReadIO** and **PMDWriteIO**, you can use card-specific C-Motion functions for this purpose. The available card specific functions for this feature are:

C-Motion command	Arguments	Function description
PMDMBSetAmplifierEnable	axis handle, mask, write_value	This function writes to the 4 amplifier enable signals (AmpEnable1-4) using mask and write_value. When a 1 appears in mask, the corresponding bit position in write_value is written to the corresponding signal. The values for mask and write_value are all 0- shifted, that is they are stored in the lowest order 4 bits.
PMDMBGetAmplifierEnable	axis_handle, read_value	This function reads the value of the AmpEnable1-4 and returns them in the low order 4 bits of read_value

Connections & associated signals

AmpEnable1-4 are direct single-signal digital outputs. There are no associated connections that need to be made for these signals to function properly; however, one or more of the digital grounds must be connected. The default value, upon powerup, for all amplifier enable signals is low. In addition, these signals are driven low upon a hard reset, which is a reset via the PCI bus, or via the external ***Reset*** signal. *See section 2.3.7, Reset, page 29,* for more information.

For a complete description of the pinout connections to/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

2.3.3 DAC Output Enable

In addition to the amplifier enable outputs described above, there is a dedicated card 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 motion processor (DAC enabled). This 'shunting' occurs at a hardware level outside the motion processor itself, and represents 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 I/O address of +1. *See the table in section 2.3.2 on the previous page* for a summary of the bit functions for I/O address word +1.

To read the status of the DAC output enable function, **ReadIO** is used with address +1. The value currently in use will appear in bit 7. A 1 value indicates DAC output is enabled, which means the voltage being output by the DACs is programmed by the motion processor, while a 0 value indicates it is disabled, which means 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 with an address of +1. The change mask bit at bit 15 must be loaded with a 1, and bit 8 must be loaded with the desired value of 0 to disable, and 1 to enable output.

The default value, upon powerup, 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 2.3.7, Reset, page 29,* for more information.



Since, for safety reasons, the default condition of the DAC Output is disabled, the DAC Output must be enabled by the user for proper DAC-based operation to occur.



Example

To set the DAC output enable function to enabled, the C-Motion command **PMDWriteIO**(axis_handle, +1, 0x8080) is used. To disable the DAC output the command **PMDWriteIO**(axis_handle, +1, 0x8000) is used.

C-Motion commands

In addition to accessing these signals using the C-Motion commands **PMDReadIO** and **PMDWriteIO**, you can use card-specific C-Motion functions for this purpose. The available card specific functions for this feature are:

C-Motion command	Arguments	Function description
PMDMBSetDACOutputEnable	axis handle, write_value	This function sets the DACOutputEnable status. A 1 value written enables DAC output, while a 0 disables it.
PMDMBGetDACOutputEnable	axis_handle, read_value	This function reads the value of the DACOutputEnable function. A 1 indicates enabled, a 0 indicates not enabled.

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

2.3.4 Serial Transceiver

This module and associated signals provides the capability to operate the Navigator-PC/104 card using an asynchronous serial port, or to allow certain monitoring operations to be performed through the serial port, even while the PC/104 bus is used to command motion sequences.

To operate the serial port as the primary communication channel to the card and disable the PC/104 interface, jumper mode should be set to the 2-3 position. To use the PC/104 bus for primary communications, and allow the serial port to be used for certain diagnostic operations, 'mode' should be set to 1-2 configuration. For a complete description of primary and diagnostic communication modes, *see Navigator Motion Processor User's Guide*. The following table shows the jumper options for mode:

Jumper	Jumper location	Description
Mode	1-2 *this is the default setting of mode*	PC/104 bus is main communications channel, serial port can be used in diagnostic mode.
	2-3	PC/104 bus is disabled, serial port is primary communication channel.

The serial port can be operated at various baud rates from 1,200 to 416,667, and different configurations of stop, start, and parity codes. In addition, three connection modes are provided, point-to-point, multi-drop (address bit mode), and multi drop (idle line mode). The Navigator commands **SetSerialPortMode** and **GetSerialPortMode** are used to set, and read back, the serial port communication parameters. A complete description of the serial port, and its usage modes is provided in the *Navigator Motion Processor User's Guide*. A complete description of Navigator commands can be found in the *Navigator Motion Processor Programmer's Command Reference*.

When operated in the serial-only mode (mode set to 2-3) the card will power-up with a default serial communications parameters set of 9600 baud, no parity, one stop bit, and point to point mode. To alter these parameters in serial-only mode, the user should set new desired serial port parameters using the **SetSerialPortMode** command, while communicating at the default parameters, and then switch to the new communications parameters.

Connections & associated signals
A special 6-pin connector is used to connect to the serial port. It is also possible to order a convenient cable with this connector installed, Cable-4003. *See section 1.2 Accessory Products, page 9. Section 5.2.8, Serial I/F Connector, page 63,* provides a detailed signal description of the serial I/F connector.

For a complete description of the pinout connections to/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

2.3.5 Watch Dog Timer

To enhance the overall safety of the card a watch dog function has been included, allowing the motion processor to be automatically reset if communications to/from the host, for whatever reason, should be lost. Resetting the motion processor will have the result of setting all motor command outputs to zero, thereby letting the motors come to a safe stop.

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

If no command is sent to the +4 location, then watchdog will remain disabled. Thus watchdog disabled is the default condition upon power-up.

C-Motion commands
In addition to accessing these signals using the C-Motion command **PMDWriteIO**, you can use card-specific C-Motion functions for this purpose. The available card specific functions for this feature are:

C-Motion command	Arguments	Function description
PMDMBSetWatchDog	axis handle	This function writes to the correct value to the watchdog register, so that for the next 104 milliseconds the card will not be reset by the watchdog circuitry.

Connections & associated signals
The Watchdog timer is an internal function of the card. Thus there are no signals directly associated with this function.

2.3.6 Under Voltage Monitor

To enhance reliability under a variety of electrical conditions, a special under voltage detection circuit triggers a motion processor reset when the voltage has dropped to an unsafe level. Resetting the motion processor will have the result of setting all motor command outputs to zero, thereby letting the motors come to a safe stop.

An under voltage condition is detected when the 5V supply to the card, provided through the PC/104 bus, drops below a specific voltage specified by an under voltage supervisory device. The typical under voltage threshold is 4.625V, while the min. and the max. values are 4.50V, and 4.75V, respectively.

Connections & associated signals
The under voltage detector is an internal function of the card. Thus there are no signals directly associated with this function.

2.3.7 Reset

Although a reset occurs automatically during power up, it is sometimes desirable to reset the motion processor explicitly, through a user-initiated command or action.

There are several methods by which you can explicitly reset the motion processor. They are summarized in the table below:

Method	Type of reset	Description
External reset	Hard reset	When external signal ***Reset*** on J7 is brought low, a reset occurs. Normally this signal is pulled high by the card. This type of reset is a 'hard' reset, which resets both the card circuitry and the motion processor. *See table below for more information.*
Reset through motion processor	Soft reset	The C-Motion command **PMDReset** sends the command **Reset** to the motion processor, which causes a 'soft' reset of the motion processor only. *See table below for more information.*
Reset through PC/104 bus	Hard reset	The C-Motion command **PMDHardReset** uses the PC/104 bus to perform a 'hard' reset of both the card circuitry and the motion processor. *See table below for more information.*

After a reset occurs, the motion processor and other related output signals will be driven to known states, depending on the type of reset performed. These are summarized in the table below:

Signal name	Soft reset	Hard reset
AxisOut1-4	High	High
PWMMag1A-4C	For DC-brush or mixed motor cards: low	For DC-brush or mixed motor cards: low
	For brushless-DC cards: 50/50 high/low	For brushless-DC cards: 50/50 high/low
	For microstepping cards: low	For microstepping cards: low
PWMSign1A-4B	Low	Low
DAC1A-DAC4B	0.0 volts	0.0 volts
DigitalOut0-7	No change	Low
AmpEnable1-4	No change	Low
Dac On/Off	No change	Off

C-Motion commands

The available C-Motion callable functions for this feature are:

C-Motion command	Arguments	Function description
PMDHardReset	axis_handle	This function causes a 'hard' reset of the motion processor. Unlike all other card-specific commands, this command is processed directly through the PC/104 card interface.

Connections & associated signals

The reset feature has an external signal input, ***Reset***, associated with it.

2.3.8 Reset Monitor

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

ReadIO and WriteIO commands

To determine the source of a card 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 shows the encoding of this I/O address word:

Address	Bit location	Signals
2	0-11	Reserved
	12	Software command: a 1 value in this bit indicates reset caused by a software command by the user.
	13	Under voltage detection: a 1 value in this bit indicates reset caused by under voltage detection.
	14	External signal: a 1 value in this bit indicates reset caused by external signal. Such a reset originates from the J1 and J7 connectors on the connector 'Reset'. When this signal is brought low, a reset condition occurs.
	15	Watch dog timeout: a 1 value in this bit indicates reset caused by watch dog timeout.

Once a reset condition has occurred, the reset status stored in location +2 described above can be cleared by a **WriteIO** command to address +2 with a value of 0.



Example

To determine that a reset has occurred, and to determine what the reset condition is, the C-Motion command **PMDReadIO**(axis_handle, 2, &reset_value) is used. Assuming that a watchdog timer event has occurred, the value returned would be 0x8000. To reset the reset monitor word, the command **PMDWriteIO**(axis_handle, 2.0) is sent.

C-Motion commands

In addition to accessing these signals using the C-Motion commands **PMDReadIO** and **PMDWriteIO**, you can use card-specific C-Motion functions for this purpose. The available card specific functions for this feature are:

C-Motion command	Arguments	Function description
PMDMBGetResetCause	axis_handle, reset_cause	This function returns the reset cause in the variable reset_cause and also clears the reset condition.

Connections & associated signals

The reset monitor is an internal function of the card. Thus there are no signals directly associated with this function.

2.3.9 Card ID

This module allows the user to query the card for a card ID. This may be useful for verifying the type of card in situations where different types of cards are employed.

ReadIO and WriteIO Commands
To read this value the **ReadIO** command is used with an address of 0xff. The following table shows the encoding of the bits returned when this function is called:

Address	Bit location	Signals
0xFF	0-3	Major PLD revision: binary coded 4-bit word encoding the major PLD revision. This value can range from 0 to 15.
	4-7	Minor PLD revision: binary-coded 4-bit word encoding the minor PLD revision. This value can range from 0 to 15.
	8-11	Board revision: binary-coded 4-bit word encoding the board revision. This value can range from 0 to 15.
	12-15	Board type: binary-coded 4-bit word encoding the board type. This value can have one of the following values: 0 - Navigator-ISA family card 1 - Navigator-PCI family card 2 - Unused 3 - Navigator-PC/104 family card 4-15 - Unused

Example



To read the card ID the C-Motion command **ReadIO**(axis_handle, +0xff, &card_id) is used. Assuming the value returned is 0x3104, this can be interpreted as:

Navigator-PC/104 card, board revision 1, PLD revision 4.0

C-Motion commands
In addition to accessing these signals using the C-Motion command **PMDReadIO**, you can use card-specific C-Motion functions for this purpose. The available card specific functions for this feature are:

C-Motion command	Arguments	Function description
PMDMBReadCardID	axis_handle, card_ID	This function returns the card ID, encoded as defined in the table above.

Connections & associated signals
The Card ID is an internal function of the card. Thus there are no signals directly associated with this function.

2.4 Signal Processing and Hardware Functions

Signal processing and hardware functions are card functions which are not directly user-programmable. These are card characteristics which are encoded in hardware. Primarily this consists of various types of signals. The following sections lists these related groups of signals and provides information that may be helpful when connecting your motion system.

2.4.1 Home, AxisIn, AxisOut, Limits, Hall Sensors

These signals are conditioned by the card, but are output or input directly to the motion processor. The *Navigator 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 being used.

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.
This group of signals are direct single-signal digital inputs to the card, with the exception of ***AxisOut*** which is a single-ended output. There are no associated connections that need to be made for these signals to function properly, however one or more of the digital grounds must be connected. The default value, upon powerup, for all ***AxisOut*** signals, is high.

For a complete description of the pinout connections to/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55*.

2.4.2 QuadA, QuadB, Index

Quad A, B, Index
This group of signals provides position feedback to the controller which is used to track motor position, and for servo motors, to generate a motor command. For DC brush and brushless DC motors, they are required for proper operation. For microstepping or pulse and direction motors, they are optional.

The encoder-processing circuitry provides a multi-pass digital filter of the ***QuadA, QuadB***, and ***Index*** signals for each axis. This provides additional protection against erroneous noise spikes, thereby improving reliability and motion integrity.

These signals are named ***quadA1+*** through ***quadB4-*** (16 signals), and ***Index 1+*** through ***Index4-*** (8 signals).

Connections & associated signals
This group of signals are connected in one of two ways. Single-ended termination means that only one wire per signal is used, while differential (dual) mode means two wires encode each signal (labeled + and -). Differential encoding 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- is left floating. If differential connections are used, both the + and - signals are used. Differential or single-ended termination must be selected through resistor pack installation. *See table in section 1.6, Preparing the Card for Installation, page 11, for details.* 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, if desired, 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 motion processor. *See Navigator Motion Processor User's Guide,* for more information.

Associated connections that are supported by the card 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 LEDs or other 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/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55.*

2.4.3 Analog Input

The ***Analog1-8*** signals provide general purpose input of 8 analog signals. If connected, the voltages present at these various connections do not directly affect the motion processor's behavior. However they can be read through the motion processor, and thus provide a convenient way of bringing in analog signal levels that may be acted upon by the user's application code located on the PC. These signals are read using the Navigator command **ReadAnalog**. For more information on reading the value of these analog inputs, *see Navigator Motion Processor User's Guide.*

These signals are named ***Analog1-8***

The minimum allowed input voltage is 0.0V, and the maximum allowed input voltage is 4.096V. To determine the numerical value that will be read by the motion processor given a specific voltage, the following formula is used:

ReadValue = AnalogVoltage * 65,472 / 4.096V

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

AnalogVoltage = ReadValue * 4.096V / 65,472

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 motors these signals provide a stream of pulse and direction data, compatible with a wide variety of off-the-shelf step motor amplifiers. These signals are directly generated by the Navigator motion processor. For more information on output waveforms, pulse rates, and related information, *see Navigator Motion Processor User's Guide.*

These signals are named ***pulse1-4, direction1-4***.

The default value, upon powerup, for all pulse & direction output signals is high.

Connections & associated signals
This group of signals are direct single-signal digital outputs. There are no associated connections that need to be made 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/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55.*

2.4.5 PWM Out

For servo or microstepping motors these signals provide PWM (pulse width modulated) motor command signals when the motor output mode is set to **PWMSignMagnitude** or **PWM5050Magnitude**. See the *Navigator Motion Processor User's Guide* for complete information. The number of signals per axis varies depending on the motor type being connected to, the number of phases that the motor has, and the motor drive method (sign/magnitude or 50/50). *Section 5, Navigator-PC/104 Electrical Reference, page 55,* has complete connection tables for various motor configurations.

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

Connections & associated signals
This group of signals are direct single-signal digital outputs. There are no associated connections that need to be made 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/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55.*

2.4.6 Motor Command

For servo or microstepping motors, these signals contain the analog motor command when the motor output mode is set to DAC (digital to analog converter). These signals vary between -10V and +10V. S*ee Navigator Motion Processor User's Guide* for more information. The number of signals per axis varies depending on the motor type being connected to.

These signals are named ***DAC1A - DAC4B*** (4 signals)

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/from the card, *see section 5, Navigator-PC/104 Electrical Reference, page 55.*

3.0 Using Pro-Motion

3

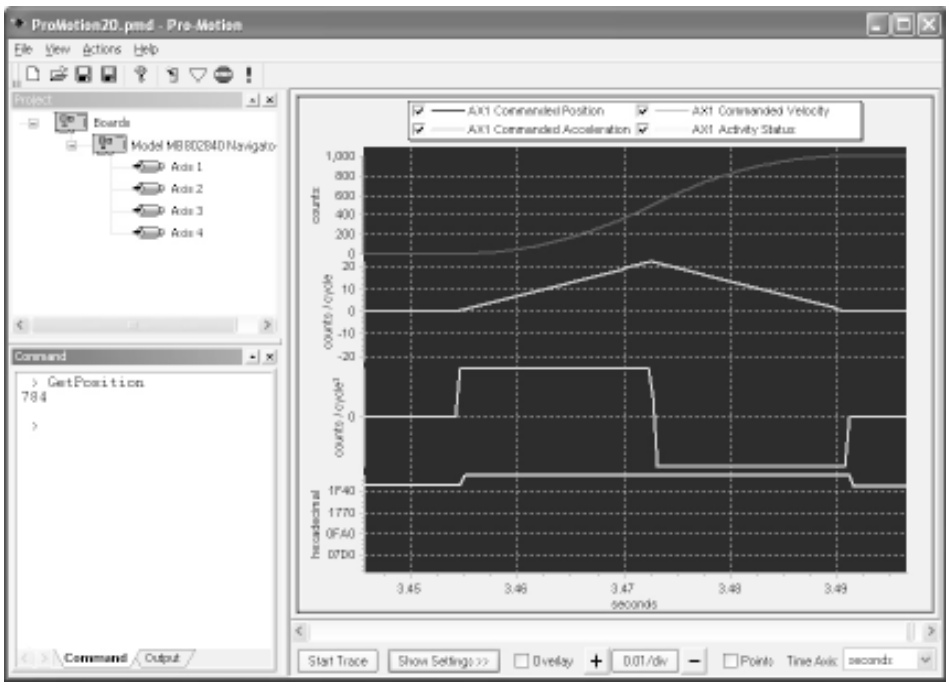
In This Section

- Communication
- Motion
- Command Window
- Scope Window

The Pro-Motion program facilitates the exercising of the PMD motion processor installed in your PMD Navigator-PC/104. All processor parameters can be viewed and modified via standard Windows controls.

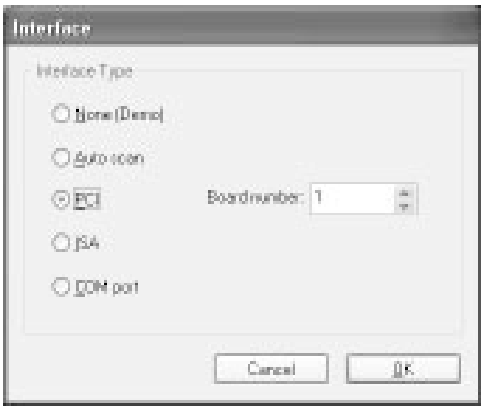
Pro-Motion features:

- Project window for accessing processor parameters.
- Command window for direct text command entry. It also serves as a communications monitor that echoes all commands sent by Pro-Motion to the card.
- Axis shuttle performs continuous back and forth motion between two positions.
- Oscilloscope graphically displays processor parameters in real-time.
- Print and export oscilloscope trace data.



3.1 Communication

When Pro-Motion is started (or file/new is selected) the Navigator-PC/104 card communication interface is selected via the interface selection dialog. Make sure the interface setting matches your hardware configuration. Click OK to proceed to the axis setup wizard.



If the serial interface (COM port) is selected in the interface selection dialog, the serial communication configuration screen is displayed.



Make sure the optional serial cable is connected from the host computer's COM port to the Navigator-PC/104 card. Set the serial interface parameters a according to the settings on the Navigator-PC/104 card and the host COM port settings. In order for Pro-Motion to control the motion processor, the Diagnostic port mode must be set to full. Note that some functions of the program may react slowly due to the slower data rate of the serial interface.

3.1.1 Axis Setup Wizard

Once communication has been established with the Navigator-PC/104 card, the connected motors may be put in motion by setting the appropriate axis parameters and running the motion shuttle.

The axis setup wizard will start automatically to configure of the Navigator-PC/104 card's axis controllers. To continue through the configuration process for each of the Navigator-PC/104 card's axes, click 'next'. To accept the default values, click 'cancel'. Individual axis settings may be modified later via Pro-Motion's axis properties dialog (access by right-clicking an axis icon  in the project window and selecting 'properties').

The axis setup wizard may be re-run at any time by selecting 'new' from Pro-Motion's file menu.



3.2 Motion Configuration

The following table summarizes the steps necessary to obtain motion depending on the card used.

Board part number	Step#	Operation
DC brush	1-3	General settings, signal sense, encoder settings
(MB802140,	7	Set filter parameters
MB802120,	8	Motion tracking
MB802110)	9	Motion dynamics
Brushless DC &	1-3	General settings, signal sense, encoder settings
mixed motor	4	Initialize commutation
(MB802340, MB802320,	4a	Check commutation
MB802310, MB802840,	8	Motion tracking
MB802820)	9	Motion dynamics
Microstepping	1-3	General settings, signal sense, encoder settings
step motor	6	Microstepper parameters
(MB802440, MB802420,	8	Motion tracking
MB802410)	9	Motion dynamics
Pulse & direction	1-3	General settings, signal sense, encoder settings
(MB802540,	5	Stepper motor parameters
MB802520,	8	Motion tracking
MB802510)	9	Motion dynamics

3.2.1 Step #1: General Settings: Amplifier Type

On the general settings page of the wizard, select the appropriate motor output mode. Make sure the motor mode is set to on and the axis enabled button is checked.



If the motor output mode is set to 'DAC', Pro-Motion will query the user about enabling the DAC outputs; if this is safe for the current hardware configuration click 'yes' otherwise click 'no'. If the DACs are not enabled at this point, they must be enabled before attempting to initiate motion *(see section 3.2.13: step #10: Initiate Motion Shuttle, page 45).*

3.2.2 Step #1a: General Settings: Time Units

Pro-Motion can allow a motion engineer to work in time units other than motion processor cycles. The program automatically converts data values to cycles before transmitting them to the processor.



3.2.3 Step #1b: General Settings: Scale Settings

Pro-Motion can allow a motion engineer to specify distance and angle values in real world units other than encoder counts or motor steps. The software translates these values into counts or steps using the scaling information supplied in this dialog before transmission to the motion processor. For example, a motor speed in a motion profile can be specified in terms of revolutions per second using the rotary scaling feature.



3.2.4 Step #2: Initialize Signal Sensing

This step of the axis setup wizard is used to select the signal sense parameters. Users should check the invert sense checkbox in the first column for any input signals which are active low.



3.2.5 Step #3: Initialize Encoders

This step of the axis setup wizard is used to select the axis encoder parameters. If the current axis does not have an encoder, set the encoder source setting to 'none' and click 'next' to continue.

Set the encoder source setting based on the axis encoder type. For axes that use an incremental quadrature encoder (differential or non-differential) select incremental. For axes that have a resolver or absolute encoder, select parallel.

The capture source setting controls high speed position capture. It can either be set to home which triggers when the home switch is triggered, or index which triggers on the encoder's index pulse.



3.2.6 Step #4: Initialize Commutation

This step of the axis setup wizard is used to select the commutation parameters. If the commutation mode is set to sinusoidal, then the initialize phase button should be clicked after all the parameters have been correctly set and before any motion is attempted or the open loop start button is pressed. Please *see chapter 12 in the Navigator Motion Processor User's Guide* for further information on phase initialization.



3.2.7 Step #4a: Check Commutation

The commutation may be verified by running the motor in open loop mode. Set the motor command parameter to a low value between 1% and 15%, keeping in mind that the motor may start to move at a rate proportional to the motor command setting. Click the open loop start button. If the motor is commutating properly, motion should occur in a single direction. Click the stop button to stop motion.

3.2.8 Step #5: Initialize Stepper Parameters

This step of the axis setup wizard is used to select the stepper motor parameters. If this stepper motor has an encoder, positions can be expressed in either motor steps or encoder counts. Set the actual position units menu to the desired value.

The encoder to step ratio fields must be set to reflect the ratio of encoder counts to motor steps, if encoder positioning or stall detection is in use.



3.2.9 Step #6: Initialize Microstepper Parameters

This step of the axis setup wizard is used to select the microstepper parameters.

Set the number of phases field to match the number of electrical phases used by your microstepping motor (usually 2).

If this motor has an encoder, input positions can be expressed in either motor steps or encoder counts. Set the actual position units menu to the desired value.

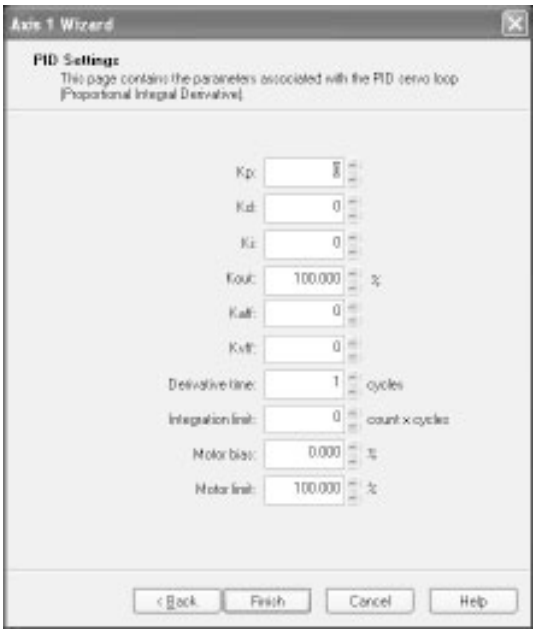
The phase counts parameter represents the number of microsteps per electrical cycle (4 times the desired number of microsteps). For example, a phase counts value of 256 sets the number of microsteps to 64 (256/4). The maximum number of microsteps that can be generated per full step is 256, giving a maximum phase counts value of 1024.

Set the motor command parameter to a low value between 1% and 15%, keeping in mind that the motor may start to move at a rate proportional to the motor command setting.



3.2.10 Step #7: Set Servo Loop Parameters

Use the PID page of the axis setup wizard to set the Kp parameter to a low value between 1 and 25 initially to prevent any oscillations.



3.2.11 Step #8: Tracking Parameters

Motion tracking parameters can initially be left at the default settings; *see the Monitoring Motion Performance section of the Navigator Motion Processor User's Guide* for more information on these parameters.

If you have connected a limit switch to the limit switch input, verify that the limit switch mode is set to 'on'. If no limit switch is connected, set limit switch mode to 'off'. If auto stop mode is enabled, the motion process will automatically stop motion on the axis when the difference between the actual and commanded position of the axis exceeds the position error limit setting on this page.



3.2.12 Step #9: Motion Dynamics Settings

Set the profile mode parameter to trapezoidal. Set the velocity to a low value (for example, 1 to 10 counts/cycle). Set the acceleration and deceleration parameters to a low value (between 0.01 and 1.0 counts/cycle for example). Click the finish button to update the motion processor parameters.



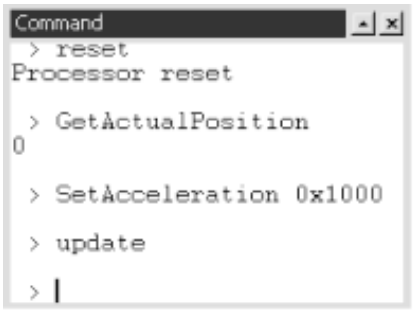
3.2.13 Step #10: Initiate Motion Shuttle (optional)

After completing the axis setup wizard, test motion may be initiated using the axis motion shuttle interface (access by right-clicking an axis icon in the project window and selecting 'shuttle').

Verify that the velocity is set to a low value (for example, 1 to 10 counts/cycle). Set the acceleration and deceleration parameters to a low value (between 0.01 and 1.0 counts/cycle for example). If desired, set the starting and ending positions for the motion (the position 1 and position 2 settings). Upon clicking 'start', the selected axis should begin to shuttle back-and-forth between the two selected positions. The motion may be tracked in real-time, by clicking 'start trace' on the scope control bar, below the motion scope window.



3.3 Command Window



The command window in Pro-Motion allows you to issue commands directly to the PMD motion processor installed in your Navigator-PC/104 card. The window has a command line style interface that accepts all of the motion processor commands. The *Navigator Motion Processor Programmer's Command Reference* contains a full list of commands along with their required parameters. The command window presents you with the command prompt '>'. The sequence on the following page shows a typical command session.

```
> Reset
Processor reset
> SetKp 25
> Update
> SetProfileMode 0
> SetVelocity 200000
> SetAcceration 256
> SetDeceleration 256
> SetPosition 123456
> Update
```

This sequence of commands:

1. Resets the Navigator motion processor;
2. Sets the Kp PID filter parameter;
3. Sets up and executes a trapezoidal move.

The command window is not case sensitive so commands can be entered in any combination of upper and lower case characters. As shown above, commands are entered as a sequence of command name followed by up to 2 numeric parameters. Parameters can represent a single 16-bit word of data or a 32-bit double word of data depending on what is required by the particular command.

```
> SetBreakpointValue 0 1000
```

In this example the first parameter represents a 16-bit word that contains the selected breakpoint number (0 or 1), and the second parameter represents a 32-bit word that contains the breakpoint value.

All of the "get" commands display the value returned by the chipset.

```
> GetEventStatus
0x0309
```

Some "get" commands require a parameter for selecting the desired value.

```
> GetBreakpointValue 0
0
```

In this example, 0 selects the breakpoint number for which the value is retrieved. The command window accepts numeric parameters in either decimal or hexadecimal format. Pre-fixing a numeric parameter with "0x" enters that number using hexadecimal format. For example,

```
> SetKp 0xA
```

sets the Kp to 10 (decimal). When a numeric parameter is entered without any prefix it is assumed to be in decimal format.

The command window automatically inserts the axis number into the command word before sending it to the Navigator motion processor. At startup, the axis number is set to 1. For motion processors that support more than one axis, you can select the axis number using the following commands:

```
> #axis 1          // selects axis # 1
> #axis 2          // selects axis # 2
> #axis 3          // selects axis # 3
> #axis 4          // selects axis # 4
```

The command window is capable of executing sequences of commands stored in external files using the file-input command "<" followed by the name of the file to execute.

```
> <setup.txt
```

As an example, the file "setup.txt" could have the following command sequence that initializes DAC output mode and sets the servo parameters for the attached hardware.

```
Reset
SetOutputMode 0
SetKp 25
SetKd 200
Update
```

This file can then be used any time the system needs to be re-started. Any sequence of valid commands can be contained within the script file.

The tilde symbol '~' can be prefixed to any line to make that line a comment which will not be executed.

3.3.1 Commands Available within the Command Window

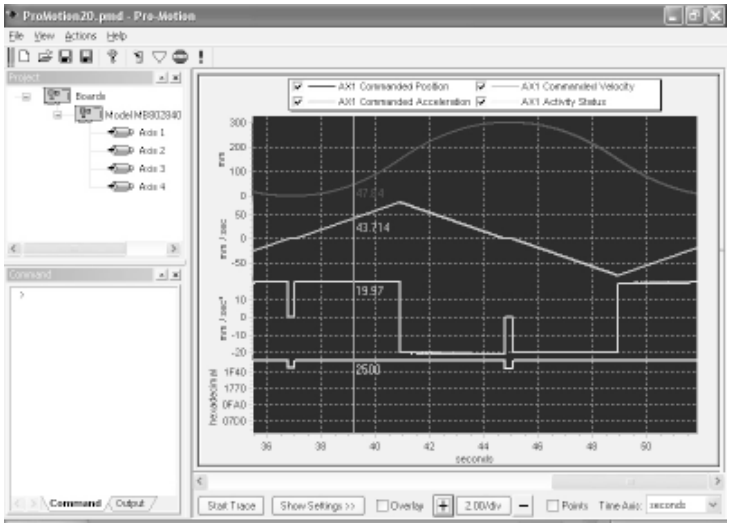
CTRL-R:	Repeat the last command entered.
TAB:	Automatic command name completion. Press this key after having entered the first few characters of the command you wish to execute. If more than one command begins with these characters, a window containing a list of commands that match the starting characters will appear. Select the correct command using the keyboard or mouse.
UP-ARROW:	Sequentially scrolls back through previously entered commands.
DOWN-ARROW:	Sequentially scrolls forward through previously entered commands.
ESC:	Clears the command line.
#axis *number:*	Selects the current axis (e.g., #axis 3 selects axis number 3.) All future commands will be executed on the selected axis.
<filename	Script file-input command. Executes the commands contained in the specified file.
TraceDownload *filename:*	Reads trace data from the external memory and stores it to the specified *filename*. Prior to executing this command, an actual trace must have been started as described in section 7 of the *Navigator Motion Processor User's Guide*.

For more information click on *Help* in *Pro-Motion*.

3.4 Scope Window

The Pro-Motion oscilloscope window displays trace data from the motion processor's hardware trace buffer in a flexible graphical format. The scope window's features include:

- Virtual buffering of trace data, allowing for display of traces many times longer than the hardware trace buffer.
- Interactive zooming and scrolling of the trace data display.
- Interactive probing of individual data samples.
- Separated and overlaid trace display modes.
- Save/restore of scope window data.
- Printing and exporting of scope window data.



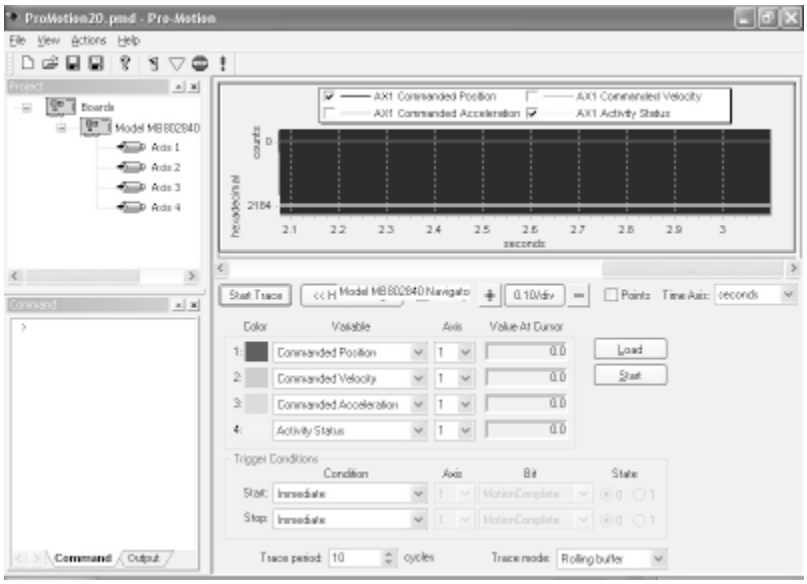
3.4.1 Scope Control Bar

On the scope control bar (just below the scope display window) are the following controls.

Control	Description
Start trace	Activates oscilloscope trace. When trace is active, this button changes to read **stop trace** and clicking it deactivates the running trace.
Show settings	Displays the scope settings window. When the settings window is showing, this button changes to read **hide settings** and clicking it hides the settings window.
Overlay	Selecting this checkbox changes the graph display to overlay the individual scope traces. This can be useful for closely comparing related variables. In overlay mode, it is often helpful to use the checkboxes in the legend box at the top of the scope display to disable unneeded traces.
+, -, Zoom level	These buttons control the zoom level of the trace display. Clicking **+** displays more samples in the trace window; clicking **–** shows fewer samples. The current zoom level is displayed on the zoom level indicator in terms of the selected time units. Clicking the zoom level indicator resets the zoom level to the scope default.
Points	Activates the data sample highlighting feature or the scope. This feature is most useful at high zoom levels, allowing the motion engineer to differentiate between trace data points and interpolated display values.
Time axis	This dropdown list controls the units displayed along the time (bottom) axis of the scope display. Note that changes to this setting only take effect upon the next trace start.

3.4.2 Scope Window Settings

Clicking the show settings button on the scope window control bar opens the scope settings window below the oscilloscope display.



The scope window can simultaneously display trace data for 1 to 4 motion processor variables. Individual traces are selected and configured using the controls in this window as specified.

Control	Description
Variable, axis	These dropdown lists select which motion processor variables from which axes shall be displayed in the trace window. Selecting none for any entry disables that entry and any of the following entries in the trace list.
Value at cursor	Displays the value under the data probe cursor when cursor is active.
Load	Load the trace buffer settings from the motion processor into Pro-Motion.
Trigger: start, axis, bit, state	The controls are used to set the scope trace to start and stop automatically when the selected motion processor state is selected.
Trace period	Sets the number of motion processor cycles between trace data samples. Valid values for this setting range from 1 to 65535.
Trace mode	Select between rolling buffer (continuous trace mode) and one-time trace mode. One-time trace mode will automatically end the trace when the motion processor's buffer is filled; rolling buffer mode will continuously capture trace data and store it in the host computer's memory until host virtual memory is exhausted.

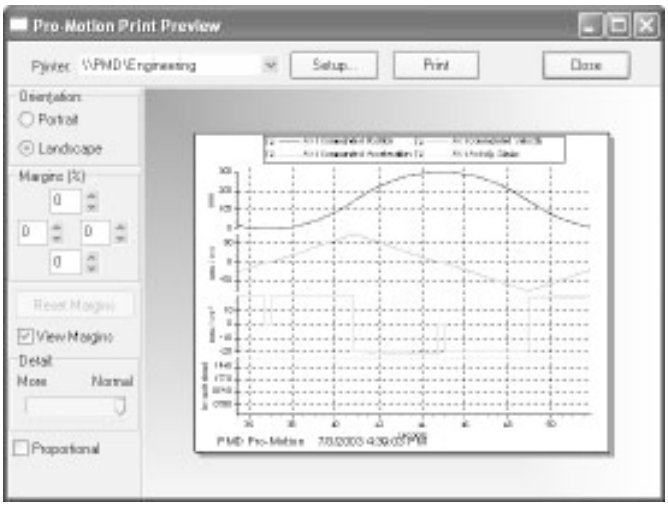
3.4.3 Interactive Scope Features

The motion engineer can use the mouse to probe and manipulate the scope display with the following mouse button combinations:

Mouse Button	Description
Button 1	Data probe: Pressing and dragging mouse button one over the scope window displays the data probe cursor. This cursor prints the value of each trace at the cursor position on the scope display, as well as printing the value in the value at cursor field of the scope settings window.
Shift + Button 1	Zoom on range: Pressing shift + mouse button one and dragging the displayed cursor over a range of data from left to right zooms the display to the selected range. Dragging the shift + button one in a right to left motion zooms out to the default zoom level.
Button 2	Scroll: Pressing and dragging mouse button two scrolls the graph in the direction of the mouse movement.

3.4.4 Printing Scope Data

Selecting print from the application file menu brings up the scope print window. The trace window is printed as it is displayed, at the current zoom level and scroll position.



3.4.5 Exporting Scope Data

Selecting export from the application file menu exports all of the trace data from the scope window to a comma-separated ASCII file format, which is compatible with major spreadsheet applications. The data file contains one column of data for each trace, with two lines of header data defining the trace variables and units in use.

4.0 Developing Your Own Applications with C-Motion

4

In This Section

- Overview
- Using C-Motion

4.1 Overview

C-Motion is a "C" source code library that contains all the code required for communicating to your Navigator-PC/104 card.

C-Motion includes the following features:

- Axis virtualization
- The ability to communicate to multiple Navigator-PC/104 cards
- The ability to link easily to any "C/C++" application

C-Motion callable functions are broken into two groups, those callable functions that encapsulate direct motion processor commands, and those callable functions that encapsulate card-specific capabilities.

Motion Processor Commands
The motion processor specific commands are detailed in the *Navigator Motion Processor Programmer's Command Reference*. They are the primary commands that you will use to control the major motion features including profile generation, servo loop closure, motor output signal generation (PWM and analog), breakpoint processing, trace operations, and many other functions.

Each Navigator motion processor command listed in the *Navigator Motion Processor Programmer's Command Reference* has a C-Motion command of the identical name, but prefaced by the letters 'PMD'. For example the Navigator command **SetPosition**, is called **PMDSetPosition**.

Card-Specific Commands
The card-specific C-Motion commands are detailed in *section 5.5, Command Summary: Card-Specific Functions, page 68*. These functions provide access to features that are located in the card, as opposed to in the motion processor. This is a much smaller set of features, and includes:

General Purpose Digital I/O function
Amplifier enable outputs
DAC output enable
Watchdog timer
Reset monitor
Card ID

4.2 Using C-Motion

The following files make up the C-Motion distribution:

C-Motion.h/C-Motion.c	Definition/declaration of the PMD Navigator command set
PMDpar.h/PMDpar.c	Parallel interface functions
PMDW32ser.h/PMDW32ser.c	Windows serial communication interface functions
PMDdrv.h/PMDdrv.c	Windows PC/104 driver communication interface functions
PMDPC/104.h/PMDPC/104.c	Windows PC/104 driver communication interface functions
PMDutil.h/PMDutil.c	General utility functions
PMDtrans.h/PMDtrans.c	Generic transport (interface) functions
PMDecode.h	Defines the PMD Navigator and C-Motion error codes
PMDocode.h	Defines the control codes for Navigator commands
PMDtypes.h	Defines the basic types required by C-Motion
PMDdiag.h	Defines a string structure for each command code that can be used for debugging or diagnostics



If you use Visual Studio 6.0 or later, it may be advantageous to use the pre-made Visual Studio Project MB80.dsp in the C-Motion Examples folder.

C-Motion can be linked to your application code by including the above "C" source files in your application. Then, for any application source file that requires access to the motion processor #include "C-Motion.h". In addition, the required interfaces need to be #defined as shown below. Only the required interfaces need to be included.

```
#define PMD_PARALLEL_INTERFACE      // use this for a standard parallel interface under DOS or Win9x
#define PMD_W32SERIAL_INTERFACE     // use this for a standard serial interface under Win9x/NT/2000/XP
#define PMD_DRIVER_INTERFACE        // use this for a standard parallel interface under Win9x/NT/2000/XP
```

By customizing the base interface functions, C-Motion can be ported to virtually any hardware platform. An example would be a memory-mapped IO scheme that uses the parallel interface. This would be built using the PMDPar.c/.h source files as a basis.

C-Motion is a set of functions that encapsulate the motion processor command set. Every command has as its first parameter an axis handle. The axis handle is a structure containing information about the interface to the motion processor and the axis number that the handle represents. Before

communicating to the motion processor, the axis handle must be initialized using the following sequence of commands:

```
    // the axis handles
PMDAxisHandle hAxis1, hAxis2;
    // open interface to PMD processor and initialize handle to axis one
PMDSetupAxisInterface_Parallel( &hAxis1, PMDAxis1, 0 );
    // initialize handle to the second axis
PMDCopyAxisInterface( &hAxis2, &hAxis1, PMDAxis2 );
```

The above is an example of initializing communication using the parallel communication interface. Each interface .c source file contains an example of initializing the interface.

Once the axis handle has been initialized, any of the motion processor commands can be executed. C-Motion.h includes the prototypes for all motion processor commands as implemented in C-Motion. Refer to this file for the required parameters for each command. *The Navigator Motion Processor Programmer's Command Reference* is the primary source for information about the operation and purpose of each command.

As part of a normal startup procedure, the motion processor is often reset. This command only needs to be executed ONCE for each motion processor, and not for every axis. The following code demonstrates a chip reset using C-Motion.

```
PMDuint16 result;
PMDuint16 status;
    // reset the PMD chip that this axis resides on
    // if more than one chip(set) is present, all of them should be
    // reset here
result = PMDReset(handle);
    // in the serial interface mode if an error occured it is returned
    // immediately with no need to call GetHostIOError, so in this
    // code we check for any error before continuing.
    // with the parallel interface the result code will always be
    // PMD_ERR_CommandError since that bit is set whenever a reset
    // occurs.  If it ISN'T set then there is some other error
if ( (result != PMD_ERR_ChipsetReset) && (result != PMD_ERR_CommandError))
{
    printf("Error: %s\n", PMDGetErrorMessage(result));
    return 0;
}

    // after the reset the chip will be in the PMDChipsetReset state
result = PMDGetHostIOError(handle, &status);

    // the above command should execute without error but we need to check
if ( (result != PMD_ERR_OK) || (status != PMD_ERR_ChipsetReset) )
{
    printf("Error: %s\n", PMDGetErrorMessage(result));
    printf("Status: %s\n", PMDGetErrorMessage(status));
    return 0;
}
return 1;
```

An example of the reset procedure is also included in the PMDutil.c source file. Every motion processor command returns a status code of type PMDuint16. The return code for every command executed should be checked before attempting to execute more commands.

```
PMDuint16 result,status;
result = PMDUpdate(&hXAxis);
if (result != PMD_ERR_OK)
{
    status = result;
    if (result == PMD_ERR_CommandError)
    PMDGetHostIOError( &hXAxis, &status );
    printf("Error: %s\n", PMDGetErrorMessage(status));
    return;
}
```

Internally, C-Motion checks the status of the HostIOError bit after issuing a command. If this bit is set it will return **PMD_ERR_CommandError**. The application should then call **PMDGetHostIOError** to clear the error bit and determine the cause of the error.

Many commands require additional parameters. Some standard values are defined by C-Motion and can be used with the appropriate commands. Refer to PMDtypes.h for a complete list of defined types. An example of calling one of the C-Motion functions with the pre-defined types is shown below.

```
PMDSetBreakpoint(&hXAxis, PMDBreakpoint1, PMDAxis2, PMDBreakpointActionAbruptStop,
PMDBreakpointActualPositionCrossed);
```

The following functions are provided by C-Motion in addition to the standard chip command set:

C-Motion command	Arguments	Function description
PMDuint16 PMDGetStatus	axis_handle	Returns the result of executing an IO status read operation. Only returns a valid result when the interface is parallel.
PMDuint16 PMDHasError	axis_handle	Returns 1 if the error bit is set in the IO status word and returns 0 if it is not set. Only returns a valid result when the interface is parallel.
PMDuint16 PMDIsReady	axis_handle	Returns 1 if the ready bit is set in the IO status word and returns 0 if it is not set. Only returns a valid result when the interface is parallel.
PMDuint16 PMDHasInterrupt	axis_handle	Returns 1 if the interrupt bit is set in the IO status word and returns 0 if it is not set. Only returns a valid result when the interface is parallel.
void PMDCloseAxisInterface	axis_handle	Should be called to terminate an interface connection.
char *PMDGetErrorMessage	errorCode	Returns a character string representation of the corresponding PMD chip or C-Motion error code.
void GetCMotionVersion	MajorVersion MinorVersion	Returns the major and minor version number of C-Motion.

5.0 Navigator-PC/104 Electrical Reference

5

In This Section

- User-Settable Components
- Connectors
- Connections Summary - Motor Amplifiers
- Command Summary - ReadIO, WriteIO
- Command Summary - Card-Specific Functions
- Environmental and Electrical Ratings

5.1 User-Settable Components

The figure below shows the locations of the principal components of the Navigator-PC/104 cards.

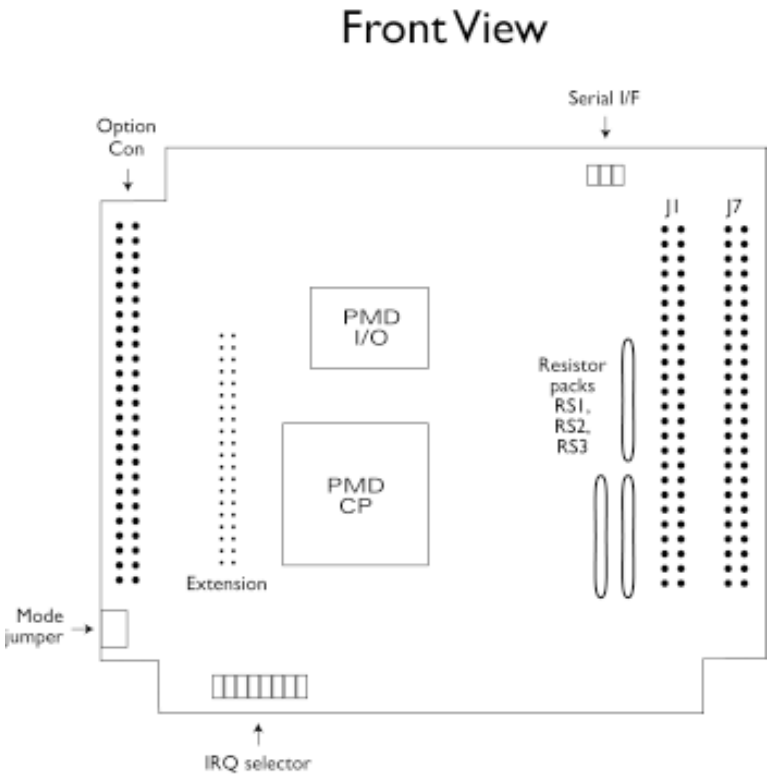
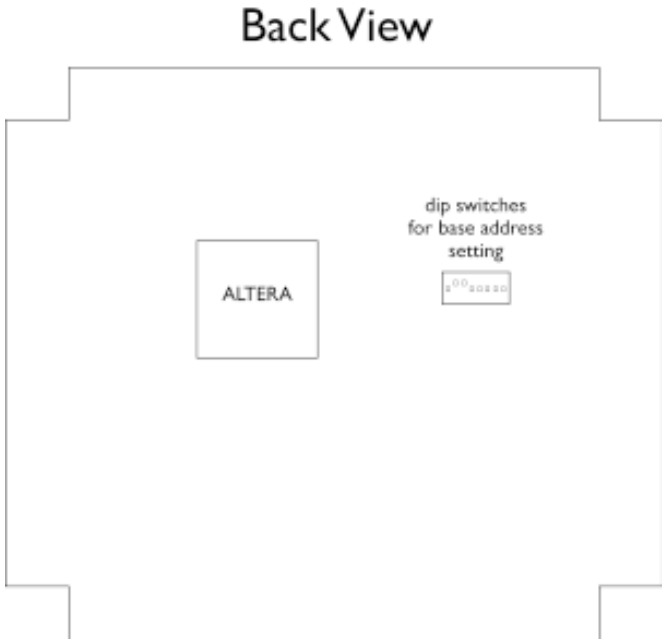


Figure 5-1a. Location of various board elements, front view

Figure 5-1b. Location of various board elements, back view



The user-accessable components of the card are:

- Switch S1
- Mode jumper
- Resistor packs RS1, RS2, and RS3

To prepare your card for installation the following user-settable hardware options should be checked:

5.1.1 Switch S1

The following table shows the function of the switch S1:

Item	How to set	Description
Switch S1	S1-1 8 (hex) S1-2 10 (hex) S1-3 20 (hex) S1-4 40 (hex) S1-5 80 (hex) S1-6 100 (hex) S1-7 200 (hex) S1-8 400 (hex) *300 (hex) is the default base address*	Switch S1 sets the card address on the PC/104 bus. The selected card address is the additive value of the switch settings indicated to the left. Moving the switch to the 'off' position adds that value to the final base address. For example, to select a base address of 340 (hex), the following switch settings would be selected: S1-1 on S1-2 on S1-3 on S1-4 off S1-5 on S1-6 off S1-7 off S1-8 on The card address selected using S1 must match the address expected by your software. In addition, this address must not be used by other cards. From the base address selected, 8 addresses are used by the Navigator-PC/104 card. For example if 300 (hex) is selected, then locations 300 - 307 will be used by the card, and cannot be used by other cards on the bus.

5.1.2 Mode Jumper

The following table shows the function of the mode jumper:

Jumper	Jumper location	Description
Mode	1-2 *this is the default setting*	PC/104 bus is main communications channel, serial port can be used in diagnostic mode.
	2-3	PC/104 bus is disabled, serial port is primary communication channel.

5.1.3 Resistor Packs RS1, RS2, RS3

The following table shows the function of the resistor packs:

Item	How to set	Description
Resistor packs RS1, RS2, RS3	Installed *this is the default setting*	If you are using differential connections leave these resistor packs installed. *See section 2.4.2, QuadA, QuadB, Index, page 32,* for more information on encoder inputs.
	Removed	If you are using single-ended encoder connections, remove the resistor packs. *See section 2.4.2, QuadA, QuadB, Index, page 32,* for more information on encoder inputs.

5.2 Connectors

There are three user-accessable connectors on the Navigator-PC/104 card

- J1 and J7 are two 50-pin headers used to connect external amplifier and motor elements.
- Option Con provides analog input signals. It is a 50-pin header-style connector.
- Serial I/F connector allows the card to be controlled via a serial port, or to allow monitoring over a separate serial port to occur

5.2.1 J1 and J7 Connector

The J1 and J7 connectors are used to connect various motion peripherals such as encoders, amplifiers, etc. They are two 50-pin header-style connectors.

Depending on what type card is used, various configurations of J1 and J7 pin connections apply.

5.2.2 J1 and J7 using DC Brush Cards

(part numbers MB802140, MB802120, MB802110)

This table shows connections to the J1 and J7 connectors when the card type is DC brush. Unless otherwise noted, 1 indicates axis 1, 2 indicates axis 2, etc.

Pin	Connection	Description	Pin	Connection	Description
J7-1	QuadA1+	Quadrature A+ encoder input (axis 1)	J1-1	QuadA3+	Quadrature A+ encoder input (axis 3)
J7-2	QuadA1-	Quadrature A- encoder input (axis 1)	J1-2	QuadA3-	Quadrature A- encoder input (axis 3)
J7-3	QuadB1+	Quadrature B+ encoder input (axis 1)	J1-3	QuadB3+	Quadrature B+ encoder input (axis 3)
J7-4	QuadB1-	Quadrature B- encoder input (axis 1)	J1-4	QuadB3-	Quadrature B- encoder input (axis 3)
J7-5	Index1+	Index+ input (axis 1)	J1-5	Index3+	Index+ input (axis 3)
J7-6	Index1-	Index- input (axis 1)	J1-6	Index3-	Index- input (axis 3)
J7-7	Vcc	+5V	J1-7	Vcc	+5V
J7-8	GND	Ground	J1-8	GND	Ground
J7-9	PosLim1	Pos. direction limit switch input (axis 1)	J1-9	PosLim3	Pos. direction limit switch input (axis 3)
J7-10	NegLim1	Neg. direction limit switch input (axis 1)	J1-10	NegLim3	Neg. direction limit switch input (axis 3)
J7-11	Home1	Home input (axis 1)	J1-11	Home3	Home input (axis 3)
J7-12	GND	Ground	J1-12	GND	Ground
J7-13	AxisOut1	AxisOut output (axis 1)	J1-13	AxisOut3	AxisOut output (axis 3)
J7-14	PWMMag1A	PWM magnitude output (axis 1)	J1-14	PWMMag3A	PWM magnitude output (axis 3)
J7-15	PWMSign1A	PWM sign output (axis 1)	J1-15	PWMSign3A	PWM sign output (axis 3)
J7-16	AxisIn1	AxisIn input (axis 1)	J1-16	AxisIn3	AxisIn input (axis 3)
J7-17	DAC1A	Analog mtr cmd output (axis 1), ±10V	J1-17	DAC3A	Analog mtr cmd output (axis 3), ±10V
J7-18	AGND	Ground for analog motor command	J1-18	AGND	Ground for analog motor command
J7-19	QuadA2+	Quadrature A+ encoder input (axis 2)	J1-19	QuadA4+	Quadrature A+ encoder input (axis 4)
J7-20	QuadA2-	Quadrature A- encoder input (axis 2)	J1-20	QuadA4-	Quadrature A- encoder input (axis 4)
J7-21	QuadB2+	Quadrature B+ encoder input (axis 2)	J1-21	QuadB4+	Quadrature B+ encoder input (axis 4)
J7-22	QuadB2-	Quadrature B- encoder input (axis 2)	J1-22	QuadB4-	Quadrature B- encoder input (axis 4)
J7-23	Index2+	Index+ input (axis 2)	J1-23	Index4+	Index+ input (axis 4)
J7-24	Index2-	Index- input (axis 2)	J1-24	Index4-	Index- input (axis 4)
J7-25	Vcc	+5V	J1-25	Vcc	+5V
J7-26	GND	Ground	J1-26	GND	Ground
J7-27	PosLim2	Pos. direction limit switch input (axis 2)	J1-27	PosLim4	Pos. direction limit switch input (axis 4)
J7-28	NegLim2	Neg. direction limit switch input (axis 2)	J1-28	NegLim4	Neg. direction limit switch input (axis 4)
J7-29	Home2	Home input (axis 2)	J1-29	Home4	Home input (axis 4)
J7-30	AxisOut2	AxisOut output (axis 2)	J1-30	AxisOut4	AxisOut output (axis 4)
J7-31	PWMMag2A	PWM magnitude output (axis 2)	J1-31	PWMMag4A	PWM magnitude output (axis 4)
J7-32	PWMSign2A	PWM sign output (axis 2)	J1-32	PWMSign4A	PWM sign output (axis 4)
J7-33	AxisIn2	AxisIn input (axis 2)	J1-33	AxisIn4	AxisIn input (axis 4)
J7-34	DAC2A	Analog mtr cmd output (axis 2), ±10V	J1-34	DAC4A	Analog mtr cmd output (axis 4), ±10V
J7-35	AGND	Ground for analog motor command	J1-35	AGND	Ground for analog motor command
J7-36	DigitalIn0	General purpose digital input 0	J1-36	DigitalIn4	General purpose digital input 4
J7-37	DigitalIn1	General purpose digital input 1	J1-37	DigitalIn5	General purpose digital input 5
J7-38	DigitalIn2	General purpose digital input 2	J1-38	DigitalIn6	General purpose digital input 6
J7-39	DigitalIn3	General purpose digital input 3	J1-39	DigitalIn7	General purpose digital input 7
J7-40	AmpEnable1	Amplifier enable signal (axis 1)	J1-40	AmpEnable3	Amplifier enable signal (axis 3)
J7-41	DigitalOut0	General purpose digital output 0	J1-41	DigitalOut4	General purpose digital output 4
J7-42	DigitalOut1	General purpose digital output 1	J1-42	DigitalOut5	General purpose digital output 5
J7-43	DigitalOut2	General purpose digital output 2	J1-43	DigitalOut6	General purpose digital output 6
J7-44	DigitalOut3	General purpose digital output 3	J1-44	DigitalOut7	General purpose digital output 7
J7-45	AmpEnable2	Amplifier enable signal (axis 2)	J1-45	AmpEnable4	Amplifier enable signal (axis 4)
J7-46	Reset	Hardware reset input	J1-46	AnalogGND	Gnd for general purpose analog inputs
J7-47	Analog1	General purpose analog input 1	J1-47	Analog5	General purpose analog input 5
J7-48	Analog2	General purpose analog input 2	J1-48	Analog6	General purpose analog input 6
J7-49	Analog3	General purpose analog input 3	J1-49	Analog7	General purpose analog input 7
J7-50	Analog4	General purpose analog input 4	J1-50	Analog8	General purpose analog input 8

5.2.3 J1 and J7 using Brushless DC Cards, Microstepping Cards, and Mixed Motor Cards

(part numbers MB802340, MB802320, MB802310, MB802440, MB802420, MB802410, MB802840, MB802820)

The following table shows connections to the J1 and J7 connectors when the card type is brushless DC, microstepping, or mixed motor. Unless otherwise noted, 1 indicates axis 1, 2 indicates axis 2, etc. Note that in addition to the J1 and J7 connectors detailed below, proper operation of the brushless DC, microstepping, or mixed motors cards also requires connections to the Option Con connector. *See section 5.2.6, Option Con using Brushless DC Cards, page 61,* and *section 5.2.7, Option Con using Microstepping Cards, page 62,* for a detailed list of Option Con connections.

Pin	Connection	Description	Pin	Connection	Description
J7-1	QuadA1+	Quadrature A+ encoder input (axis 1)	J1-1	QuadA3+	Quadrature A+ encoder input (axis 3)
J7-2	QuadA1-	Quadrature A- encoder input (axis 1)	J1-2	QuadA3-	Quadrature A- encoder input (axis 3)
J7-3	QuadB1+	Quadrature B+ encoder input (axis 1)	J1-3	QuadB3+	Quadrature B+ encoder input (axis 3)
J7-4	QuadB1-	Quadrature B- encoder input (axis 1)	J1-4	QuadB3-	Quadrature B- encoder input (axis 3)
J7-5	Index1+	Index+ input (axis 1)	J1-5	Index3+	Index+ input (axis 3)
J7-6	Index1-	Index- input (axis 1)	J1-6	Index3-	Index- input (axis 3)
J7-7	Vcc	+5V	J1-7	Vcc	+5V
J7-8	GND	Ground	J1-8	GND	Ground
J7-9	PosLim1	Pos. direction limit switch input (axis 1)	J1-9	PosLim3	Pos. direction limit switch input (axis 3)
J7-10	NegLim1	Neg. direction limit switch input (axis 1)	J1-10	NegLim3	Neg. direction limit switch input (axis 3)
J7-11	Home1	Home input (axis 1)	J1-11	Home3	Home input (axis 3)
J7-12	GND	Ground	J1-12	GND	Ground
J7-13	AxisOut1	AxisOut output (axis 1)	J1-13	AxisOut3	AxisOut output (axis 3)
J7-14	n.c.	No connection	J1-14	n.c.	No connection
J7-15	n.c.	No connection	J1-15	n.c.	No connection
J7-16	AxisIn1	AxisIn input (axis 1)	J1-16	AxisIn3	AxisIn input (axis 3)
J7-17	DAC1A	Phase A analog mtr cmd output (axis 1), ±10V	J1-17	DAC3A	Phase A analog mtr cmd output (axis 3), ±10V
J7-18	AGND	Ground for analog motor command	J1-18	AGND	Ground for analog motor command
J7-19	QuadA2+	Quadrature A+ encoder input (axis 2)	J1-19	QuadA4+	Quadrature A+ encoder input (axis 4)
J7-20	QuadA2-	Quadrature A- encoder input (axis 2)	J1-20	QuadA4-	Quadrature A- encoder input (axis 4)
J7-21	QuadB2+	Quadrature B+ encoder input (axis 2)	J1-21	QuadB4+	Quadrature B+ encoder input (axis 4)
J7-22	QuadB2-	Quadrature B- encoder input (axis 2)	J1-22	QuadB4-	Quadrature B- encoder input (axis 4)
J7-23	Index2+	Index+ input (axis 2)	J1-23	Index4+	Index+ input (axis 4)
J7-24	Index2-	Index- input (axis 2)	J1-24	Index4-	Index- input (axis 4)
J7-25	Vcc	+5V	J1-25	Vcc	+5V
J7-26	GND	Ground	J1-26	GND	Ground
J7-27	PosLim2	Pos. direction limit switch input (axis 2)	J1-27	PosLim4	Pos. direction limit switch input (axis 4)
J7-28	NegLim2	Neg. direction limit switch input (axis 2)	J1-28	NegLim4	Neg. direction limit switch input (axis 4)
J7-29	Home2	Home input (axis 2)	J1-29	Home4	Home input (axis 4)
J7-30	AxisOut2	AxisOut output (axis 2)	J1-30	AxisOut4	AxisOut output (axis 4)
J7-31	n.c.	No connection	J1-31	n.c.	No connection
J7-32	n.c.	No connection	J1-32	n.c.	No connection
J7-33	AxisIn2	AxisIn input (axis 2)	J1-33	AxisIn4	AxisIn input (axis 4)
J7-34	DAC2A	Phase A analog mtr cmd output (axis 2), ±10V	J1-34	DAC4A	Phase A analog mtr cmd output (axis 4), ±10V
J7-35	AGND	Ground for analog motor command	J1-35	AGND	Ground for analog motor command
J7-36	DigitalIn0	General purpose digital input 0	J1-36	DigitalIn4	General purpose digital input 4
J7-37	DigitalIn1	General purpose digital input 1	J1-37	DigitalIn5	General purpose digital input 5
J7-38	DigitalIn2	General purpose digital input 2	J1-38	DigitalIn6	General purpose digital input 6
J7-39	DigitalIn3	General purpose digital input 3	J1-39	DigitalIn7	General purpose digital input 7
J7-40	AmpEnable1	Amplifier enable signal (axis 1)	J1-40	AmpEnable3	Amplifier enable signal (axis 3)
J7-41	DigitalOut0	General purpose digital output 0	J1-41	DigitalOut4	General purpose digital output 4
J7-42	DigitalOut1	General purpose digital output 1	J1-42	DigitalOut5	General purpose digital output 5
J7-43	DigitalOut2	General purpose digital output 2	J1-43	DigitalOut6	General purpose digital output 6
J7-44	DigitalOut3	General purpose digital output 3	J1-44	DigitalOut7	General purpose digital output 7
J7-45	AmpEnable2	Amplifier enable signal (axis 2)	J1-45	AmpEnable4	Amplifier enable signal (axis 4)
J7-46	Reset	Hardware reset input	J1-46	AnalogGND	Gnd for general purpose analog inputs
J7-47	Analog1	General purpose analog input 1	J1-47	Analog5	General purpose analog input 5
J7-48	Analog2	General purpose analog input 2	J1-48	Analog6	General purpose analog input 6
J7-49	Analog3	General purpose analog input 3	J1-49	Analog7	General purpose analog input 7
J7-50	Analog4	General purpose analog input 4	J1-50	Analog8	General purpose analog input 8

5.2.4 J1 and J7 using Pulse & Direction Cards

(part numbers MB802540, MB802520, MB802510)

The following table shows connections to the J1 and J7 connectors when the card type is pulse & direction. Unless otherwise noted, 1 indicates axis 1, 2 indicates axis 2, etc.

Pin	Connection	Description	Pin	Connection	Description
J7-1	QuadA1+	Quadrature A+ encoder input (axis 1)	J1-1	QuadA3+	Quadrature A+ encoder input (axis 3)
J7-2	QuadA1-	Quadrature A- encoder input (axis 1)	J1-2	QuadA3-	Quadrature A- encoder input (axis 3)
J7-3	QuadB1+	Quadrature B+ encoder input (axis 1)	J1-3	QuadB3+	Quadrature B+ encoder input (axis 3)
J7-4	QuadB1-	Quadrature B- encoder input (axis 1)	J1-4	QuadB3-	Quadrature B- encoder input (axis 3)
J7-5	Index1+	Index+ input (axis 1)	J1-5	Index3+	Index+ input (axis 3)
J7-6	Index1-	Index- input (axis 1)	J1-6	Index3-	Index- input (axis 3)
J7-7	Vcc	+5V	J1-7	Vcc	+5V
J7-8	GND	Ground	J1-8	GND	Ground
J7-9	PosLim1	Pos. direction limit switch input (axis 1)	J1-9	PosLim3	Pos. direction limit switch input (axis 3)
J7-10	NegLim1	Neg. direction limit switch input (axis 1)	J1-10	NegLim3	Neg. direction limit switch input (axis 3)
J7-11	Home1	Home input (axis 1)	J1-11	Home3	Home input (axis 3)
J7-12	GND	Ground	J1-12	GND	Ground
J7-13	AxisOut1	AxisOut output (axis 1)	J1-13	AxisOut3	AxisOut output (axis 3)
J7-14	Pulse1	Pulse output (axis 1)	J1-14	Pulse3	Pulse output (axis 3)
J7-15	Direction1	Direction output (axis 1)	J1-15	Direction3	Direction output (axis 4)
J7-16	AxisIn1	AxisIn input (axis 1)	J1-16	AxisIn3	AxisIn input (axis 3)
J7-17	AtRest1	Atrest indicator output (axis 1)	J1-17	AtRest3	Atrest indicator output (axis 1)
J7-18	GND	Ground	J1-18	GND	Ground
J7-19	QuadA2+	Quadrature A+ encoder input (axis 2)	J1-19	QuadA4+	Quadrature A+ encoder input (axis 4)
J7-20	QuadA2-	Quadrature A- encoder input (axis 2)	J1-20	QuadA4-	Quadrature A- encoder input (axis 4)
J7-21	QuadB2+	Quadrature B+ encoder input (axis 2)	J1-21	QuadB4+	Quadrature B+ encoder input (axis 4)
J7-22	QuadB2-	Quadrature B- encoder input (axis 2)	J1-22	QuadB4-	Quadrature B- encoder input (axis 4)
J7-23	Index2+	Index+ input (axis 2)	J1-23	Index4+	Index+ input (axis 4)
J7-24	Index2-	Index- input (axis 2)	J1-24	Index4-	Index- input (axis 4)
J7-25	Vcc	+5V	J1-25	Vcc	+5V
J7-26	GND	Ground	J1-26	GND	Ground
J7-27	PosLim2	Pos. direction limit switch input (axis 2)	J1-27	PosLim4	Pos. direction limit switch input (axis 4)
J7-28	NegLim2	Neg. direction limit switch input (axis 2)	J1-28	NegLim4	Neg. direction limit switch input (axis 4)
J7-29	Home2	Home input (axis 2)	J1-29	Home4	Home input (axis 4)
J7-30	AxisOut2	AxisOut output (axis 2)	J1-30	AxisOut4	AxisOut output (axis 4)
J7-31	Pulse2	Pulse output (axis 1)	J1-31	Pulse4	Pulse output (axis 4)
J7-32	Direction2	Direction output (axis 1)	J1-32	Direction4	Direction output (axis 4)
J7-33	AxisIn2	AxisIn input (axis 2)	J1-33	AxisIn4	AxisIn input (axis 4)
J7-34	AtRest2	Atrest indicator output (axis 1)	J1-34	AtRest4	Atrest indicator output (axis 1)
J7-35	GND	Ground	J1-35	GND	Ground
J7-36	DigitalIn0	General purpose digital input 0	J1-36	DigitalIn4	General purpose digital input 4
J7-37	DigitalIn1	General purpose digital input 1	J1-37	DigitalIn5	General purpose digital input 5
J7-38	DigitalIn2	General purpose digital input 2	J1-38	DigitalIn6	General purpose digital input 6
J7-39	DigitalIn3	General purpose digital input 3	J1-39	DigitalIn7	General purpose digital input 7
J7-40	AmpEnable1	Amplifier enable signal (axis 1)	J1-40	AmpEnable3	Amplifier enable signal (axis 3)
J7-41	DigitalOut0	General purpose digital output 0	J1-41	DigitalOut4	General purpose digital output 4
J7-42	DigitalOut1	General purpose digital output 1	J1-42	DigitalOut5	General purpose digital output 5
J7-43	DigitalOut2	General purpose digital output 2	J1-43	DigitalOut6	General purpose digital output 6
J7-44	DigitalOut3	General purpose digital output 3	J1-44	DigitalOut7	General purpose digital output 7
J7-45	AmpEnable2	Amplifer enable signal (axis 2)	J1-45	AmpEnable4	Amplifer enable signal (axis 4)
J7-46	Reset	Hardware reset input	J1-46	AnalogGND	Gnd for general purpose analog inputs
J7-47	Analog1	General purpose analog input 1	J1-47	Analog5	General purpose analog input 5
J7-48	Analog2	General purpose analog input 2	J1-48	Analog6	General purpose analog input 6
J7-49	Analog3	General purpose analog input 3	J1-49	Analog7	General purpose analog input 7
J7-50	Analog4	General purpose analog input 4	J1-50	Analog8	General purpose analog input 8

5.2.5 Option Con Connector

When either the brushless DC, microstepping, or mixed motor cards are used, the Option Con connector provides additional signals for multi-phase motor output and input of signals such as Hall sensors.

5.2.6 Option Con using Brushless DC or Mixed Motor Cards

(part numbers MB802340, MB802320, MB802310, MB802840, MB802820)

The following table shows connections to the Option Con connector when the card type is brushless DC, or mixed motor. Unless otherwise noted, 1 indicates axis 1, 2 indicates axis 2, etc., and A indicates phase A, B indicates phase B, etc. Note that in addition to the Option Con connector detailed below, proper operation of the brushless DC, or mixed motor cards, also requires connections to the J1 and J7 connectors. *See section 5.2.3, J1 and J7 using Brushless DC Cards, Microstepping Cards and Mixed Motor Cards, page 59,* for a detailed list of J1 and J7 connections.

Pin	Connection	Description	Pin	Connection	Description
1	PWMMag1A	Phase A PWM magnitude output (axis 1)	26	HALL2B	Phase B hall sensor input (axis 2)
2	PWMMag1B	Phase B PWM magnitude output (axis 1)	27	HALL2C	Phase C hall sensor input (axis 2)
3	PWMMag1C	Phase C PWM magnitude output (axis 1)	28	GND	Ground
4	PWMSign1A	Phase A PWM sign output (axis 1)	29	HALL3A	Phase A hall sensor input (axis 3)
5	GND	Ground	30	HALL3B	Phase B hall sensor input (axis 3)
6	PWMMag2A	Phase A PWM magnitude output (axis 2)	31	HALL3C	Phase C hall sensor input (axis 3)
7	PWMMag2B	Phase B PWM magnitude output (axis 2)	32	GND	Ground
8	PWMMag2C	Phase C PWM magnitude output (axis 2)	33	HALL4A	Phase A hall sensor input (axis 4)
9	PWMSign2A	Phase A PWM sign output (axis 2)	34	HALL4B	Phase B hall sensor input (axis 4)
10	GND	Ground	35	HALL4C	Phase C hall sensor input (axis 4)
11	PWMMag3A	Phase A PWM magnitude output (axis 3)	36	GND	Ground
12	PWMMag3B	Phase B PWM magnitude output (axis 3)	37	AGND	Ground for analog motor command
13	PWMMag3C	Phase C PWM magnitude output (axis 3)	38	DAC1A	Phase A analog mtr cmd output (axis 1), ±10V
14	PWMSign3A	Phase A PWM sign output (axis 3)	39	DAC2A	Phase A analog mtr cmd output (axis 2), ±10V
15	GND	Ground	40	DAC3A	Phase A analog mtr cmd output (axis 3), ±10V
16	PWMMag4A	Phase A PWM magnitude output (axis 4)	41	DAC4A	Phase A analog mtr cmd output (axis 4), ±10V
17	PWMMag4B	Phase B PWM magnitude output (axis 4)	42	DAC1B	Phase B analog mtr cmd output (axis 1), ±10V
18	PWMMag4C	Phase C PWM magnitude output (axis 4)	43	DAC2B	Phase B analog mtr cmd output (axis 2), ±10V
19	PWMSign4A	Phase A PWM sign output (axis 4)	44	DAC3B	Phase B analog mtr cmd output (axis 3), ±10V
20	GND	Ground	45	DAC4B	Phase B analog mtr cmd output (axis 4), ±10V
21	HALL1A	Phase A hall sensor input (axis 1)	46	AGND	Ground for analog motor command
22	HALL1B	Phase B hall sensor input (axis 1)	47	GND	Ground
23	HALL1C	Phase C hall sensor input (axis 1)	48	GND	Ground
24	GND	Ground	49	Vcc	+5V
25	HALL2A	Phase A hall sensor input (axis 2)	50	Vcc	+5V

5.2.7 Option Con using Microstepping Cards

(part numbers MB802440, MB802420, MB802410)

The following table shows connections to the Option Con connector when the card type is microstepping. Unless otherwise noted, 1 indicates axis 1, 2 indicates axis 2, etc., and A indicates phase A, B indicates phase B, etc. Note that in addition to the Option Con connector detailed below, proper operation of the microstepping cards also requires connections to the J1 and J7 connectors. *See section 5.2.3, J1 and J7 using Brushless DC Cards, Microstepping Cards and Mixed Motor Cards, page 59,* for a detailed list of J1 and J7 connections.

Pin	Connection	Description	Pin	Connection	Description
1	PWMMag1A	Phase A PWM magnitude output (axis 1)	26	n.c.	No connection
2	PWMMag1B	Phase B PWM magnitude output (axis 1)	27	n.c.	No connection
3	PWMSign1B	Phase B PWM sign output (axis 1)	28	GND	Ground
4	PWMSign1A	Phase A PWM sign output (axis 1)	29	n.c.	No connection
5	GND	Ground	30	n.c.	No connection
6	PWMMag2A	Phase A PWM magnitude output (axis 2)	31	n.c.	No connection
7	PWMMag2B	Phase B PWM magnitude output (axis 2)	32	GND	Ground
8	PWMSign2B	Phase B PWM sign output (axis 2)	33	n.c.	No connection
9	PWMSign2A	Phase A PWM sign output (axis 2)	34	n.c.	No connection
10	GND	Ground	35	n.c.	No connection
11	PWMMag3A	Phase A PWM magnitude output (axis 3)	36	GND	Ground
12	PWMMag3B	Phase B PWM magnitude output (axis 3)	37	AGND	Ground for analog motor command
13	PWMSign3B	Phase B PWM sign output (axis 3)	38	DAC1A	Phase A analog mtr cmd output (axis 1), ±10V
14	PWMSign3A	Phase A PWM sign output (axis 3)	39	DAC2A	Phase A analog mtr cmd output (axis 2), ±10V
15	GND	Ground	40	DAC3A	Phase A analog mtr cmd output (axis 3), ±10V
16	PWMMag4A	Phase A PWM magnitude output (axis 4)	41	DAC4A	Phase A analog mtr cmd output (axis 4), ±10V
17	PWMMag4B	Phase B PWM magnitude output (axis 4)	42	DAC1B	Phase B analog mtr cmd output (axis 1), ±10V
18	PWMSign4B	Phase B PWM sign output (axis 4)	43	DAC2B	Phase B analog mtr cmd output (axis 2), ±10V
19	PWMSign4A	Phase A PWM sign output (axis 4)	44	DAC3B	Phase B analog mtr cmd output (axis 3), ±10V
20	GND	Ground	45	DAC4B	Phase B analog mtr cmd output (axis 4), ±10V
21	n.c.	No connection	46	AGND	Ground for analog motor command
22	n.c.	No connection	47	GND	Ground
23	n.c.	No connection	48	GND	Ground
24	GND	Ground	49	Vcc	+5V
25	n.c.	No connection	50	Vcc	+5V

5.2.8 Serial I/F Connector

The following connector is used during serial communications. The Serial I/F connector is a 6-position Molex MLX Micro-4-SMD style connector.

Pin	Connection	Description
1	SrlEnable	Serial enable (only used for RS422/485)
2	SrlXmt	Serial transmit output
3	SrlRcv	Serial receive input
4	GND	Ground
5	Vcc	+5V
6	n.c.	No connection

For more information on communicating to the card by serial port, see the *Navigator Motion Processor User's Guide*.

5.3 Connections Summary - Motor Amplifiers

The Navigator card supports three methods of output to motor amplifiers:

DAC	Analog signals from the on-board 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

In addition, each motor axis may have 1, 2, or 3 output phases associated with it. For DC brush motors, the number of phases is one, while for multi-phase motors such as brushless DC or microstepping motors, the number of phases can be 2 or 3, depending on the output waveform programmed into the Navigator motion processor. For more information, see the *Navigator Motion Processor User's Guide.*

The tables on the following pages provide convenient summaries of amplifier connections for various 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.

5.3.1 DC Brush Motor Connections

			Connection			
Motor output method	**Navigator-PC/104 connection name**	**Amplifier input connection name***	**Axis 1**	**Axis 2**	**Axis 3**	**Axis 4**
DAC	DAC1-4A	Ref+ or V+	J7-17	J7-34	J1-17	J1-34
	AGND	Ref- or Gnd	J7-18	J7-35	J1-18	J1-35
PWM	PWMMag1-4A	PWM magnitude	J7-14	J7-31	J1-14	J1-31
sign/magnitude	PWMSign1-4A	PWM direction	J7-15	J7-32	J1-15	J1-32
	GND	Gnd	J7-8	J7-26	J1-8	J1-26
PWM 50/50	Unused					

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

5.3.2 Brushless DC Motor Connections

			Connection			
Motor output method	**Navigator-PC/104 connection name**	**Amplifier input connection name***	**Axis 1**	**Axis 2**	**Axis 3**	**Axis 4**
DAC	DAC1-4A	Ref1+ or V1+	Option Con-38	Option Con-39	Option Con-40	Option Con-41
	DAC1-4B	Ref2+ or V2+	Option Con-42	Option Con-43	Option Con-44	Option Con-45
	AGND	Ref- or Gnd	Option Con-37	Option Con-46	Option Con-37	Option Con-46
PWM sign/magnitude	Unused					
PWM 50/50	PWMMag1-4A	PWM phase 1	Option Con-1	Option Con-6	Option Con-11	Option Con-16
	PWMMag1-4B	PWM phase 2	Option Con-2	Option Con-7	Option Con-12	Option Con-17
	PWMMag1-4C	PWM phase 3	Option Con-3	Option Con-8	Option Con-13	Option Con-18
	GND	Gnd	Option Con-5	Option Con-10	Option Con-15	Option Con-20

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

5.3.3 Microstepping Motor Connections

			Connection			
Motor output method	**Navigator-PC/104 connection name**	**Amplifier input connection name***	**Axis 1**	**Axis 2**	**Axis 3**	**Axis 4**
DAC	DAC1-4A	Ref1+ or V1+	Option Con-38	Option Con-39	Option Con-40	Option Con-41
	DAC1-4B	Ref2+ or V2+	Option Con-42	Option Con-43	Option Con-44	Option Con-45
	AGND	Ref- or Gnd	Option Con-37	Option Con-46	Option Con-37	Option Con-46
PWM sign/magnitude	PWMMag1-4A	PWM magnitude	Option Con-1	Option Con-6	Option Con-11	Option Con-16
	PWMSign1-4A	PWM direction	Option Con-4	Option Con-9	Option Con-14	Option Con-19
	PWMMag1-4B	PWM magnitude	Option Con-2	Option Con-7	Option Con-12	Option Con-17
	PWMSign1-4B	PWM direction	Option Con-3	Option Con-8	Option Con-13	Option Con-18
	GND	Gnd	Option Con-5	Option Con-10	Option Con-15	Option Con-20
PWM 50/50	Unused					

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

5.3.4 Pulse & Direction Motor Connections

			Connection			
Motor output method	**Navigator-PC/104 connection name**	**Amplifier input connection name***	**Axis 1**	**Axis 2**	**Axis 3**	**Axis 4**
Pulse & direction	Pulse1-4	Pulse or step	J7-14	J7-31	J1-14	J1-31
	Direction1-4	Direction	J7-15	J7-32	J1-15	J1-32
	GND	Gnd	J7-8	J7-26	J1-8	J1-26

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

5.4 Command Summary - ReadIO, WriteIO

Card-specific features are accessed through the Navigator's general-purpose I/O addressing scheme. These card-specific functions can be accessed using the **ReadIO** and **WriteIO** commands, or through convenient 'packaged' board-specific C-Motion commands. *See section 4, Developing Your Own Applications with C-Motion, page 51,* for more information.

The following table shows the overall address map for the ReadIO and WriteIO functions supported by the Navigator-PC/104 cards.

Address	Register	Description
0	Digital I/O	Register which allows 8 general purpose digital input and output signals to be read from and written to.
1	Amplifier & DAC enable	Register which allows AmpEnable1-4 output signals (which can also be used as a general purpose outputs to be written to and verified, and which allows DAC enable output function to be written to and verified).
2	Reset monitor	Register which allows reset monitor to be checked and reset.
3	Unused	
4	Watchdog	Register which allows watchdog function +0 be activated and updated.
0xFF	Card ID	Register which allows Card ID word to be read.

5.4.1 General Purpose Digital I/O Control Register (Address +0)

The following table details the digital I/O control register

Address	Bit location	Signals
0	0-7	Digital input signals
	8-15	Digital output signals

5.4.2 Amplifier & DAC Enable Control Register (Address +1)

The following table details the amplifier & DAC enable register

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)

5.4.3 Reset Monitor Control Register (Address +2)

The following table details the reset monitor register

Address	Bit location	Signals
2	0-11	Reserved
	12	Software command: a 1 value in this bit indicates reset caused by a software command by the user.
	13	Under voltage detection: a 1 value in this bit indicates reset caused by under voltage detection.
	14	External signal: a 1 value in this bit indicates reset caused by external signal. Such a reset originates from the J1 and J7 connectors on the connector 'Reset'. When this signal is brought low, a reset condition occurs.
	15	Watch dog timeout: a 1 value in this bit indicates reset caused by watch dog timeout.

5.4.4 Card ID Register (Address +0xFF)

The following table details the card ID register

Address	Bit location	Signals
0xFF	0-3	Major PLD revision: binary coded 4-bit word encoding the major PLD revision. This value can range from 0 to 15.
	4-7	Minor PLD revision: binary-coded 4-bit word encoding the minor PLD revision. This value can range from 0 to 15.
	8-11	Board revision: binary-coded 4-bit word encoding the board revision. This value can range from 0 to 15.
	12-15	Board type: binary-coded 4-bit word encoding the board type. This value can have one of the following values: 0 - Navigator-ISA family card 1 - Navigator-PCI family card 2 - Unused 3 - Navigator-PC/104 family card 4-15 - Unused

5.5 Command Summary - Card-Specific Functions

The following tables provide convenient summaries for card-specific C-Motion commands. The much larger list of Navigator motion processor commands is detailed in *Navigator Motion Processor Programmer's Command Reference*.

C-Motion command	Arguments	Function description
PMDMBWriteDigitalOutput	axis handle, write_value	This function writes to the 8 general-purpose digital I/O signals (digitalOut0-7). Write_value holds the 8 signals in its low order 8 bits.
PMDMBReadDigitalInput	axis_handle, read_value	This function reads the value of the signals DigitalIn0-7 and returns them in the low order 8 bits of read_value
PMDMBReadDigitalOutput	axis_handle, read_value	This function reads the value of the signals DigitalOut0-7 and returns them in the low order 8 bits of read_value
PMDMBSetAmplifierEnable	axis handle, mask, write_value	This function writes to the 4 amplifier enable signals (AmpEnable1-4) using mask and write_value. When a 1 appears in mask, the corresponding bit position in write_value is written to the corresponding signal. The values for mask and write_value are all 0-shifted, that is they are stored in the lowest order 4 bits.
PMDMBGetAmplifierEnable	axis_handle, read_value	This function reads the value of the AmpEnable1-4 and returns them in the low order 4 bits of read_value
PMDMBSetDACOutputEnable	axis handle, write_value	This function sets the DACOutputEnable status. A 1 value written enables DAC output, while a 0 disables it.
PMDMBGetDACOutputEnable	axis_handle, read_value	This function reads the value of the DACOutputEnable function. A 1indicates enabled, a 0 indicates not enabled.
PMDMBSetWatchDog	axis handle	This function writes to the correct value to the watchdog register, so that for the next 104 milliseconds the card will not be reset by the watchdog circuitry.
PMDHardReset	axis_handle	This function causes a 'hard' reset of the motion processor. Unlike all other card-specific commands, this command is processed directly through the PC/104 card interface
PMDMBGetResetCause	axis handle, reset_cause	This function returns the reset cause in the variable reset_cause, and also clears the reset condition.
PMDMBReadCardID	axis handle, card_ID	This function returns the card ID, encoded as defined in the table above.

5.6 Environmental and Electrical Ratings

Dimensions:	standard PC/104 (96mm x 110.5mm)
Storage Temperature:	-40 to +125 degrees C
Operating Temperature:	0 to +70 degrees C (standard commercial version)
Power requirement:	4.8 V to 5.25 V operating range, 1 Amp no load current: 0.6 A (no outputs on)
Supply voltage limits:	-0.3 V to +7 V
Analog Output Range:	-10.0 .0 V to +10.0 V, ±5 mA/axis, ±20 mA max./axis
Analog Input Range:	0 - 4.096 V
Digital outputs drive capacity:	DC output source or sink current: +/- 50 mA
Under voltage detection:	The under voltage supervisory device used is a MCP120T-475I. Under voltage specs are Min: 4.50 V, Max: 4.75 V, Typical: 4.625 V.

Performance Motion Devices, Inc.
55 Old Bedford Rd
Lincoln, MA 01773

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP