Hardware/Software Manual
For the SB-3622X
Digital-to-Synchro/Resolver
PCI Interface Card

MN-3622XXX-001

The information provided in this User's Guide is believed to be accurate; however, no responsibility is assumed by Data Device Corporation for its use, and no license or rights are granted by implication or otherwise in connection therewith.

Specifications are subject to change without notice.

Please visit our Web site at **www.ddc-web.com** for the latest information

105 Wilbur Place, Bohemia, New York 11716-2482

For Technical Support – 1-800-DDC-5757 ext. 7771
Headquarters U.S.A. - Tel: (631) 567-5600, Fax: (631) 567-7358
United Kingdom - Tel: +44-(0)1635-811140, Fax: +44-(0)1635-32264
France - Tel: +33-(0)1-41-16-3424, Fax: +33-(0)1-41-16-3425
Germany - Tel: +49-(0)89-1500 12-11, Fax: +49-(0)89-1500 12-22
Japan - Tel: +81-(0)3-3814-7688, Fax: +81-(0)3-3814-7689
World Wide Web – http://www.ddc-web.com

DATA DEVICE CORPORATION
REGISTERED TO ISO 9001:2000
FILE NO. A5976

REV G, December, 2007

RECORD OF CHANGE

Revision	Date	Pages	Description
A	10/2000	All	Original Issue
B	10/2000	All	Software/Clerical update
C	10/2001	All	Software/Clerical update
D	05/2002	All	Linux/Clerical update
E	10/2003	Various	Updated: Hardware Installation section, Program_Amplitude details, Specifications Table, Ordering Information, Appendix B, F. Added: New functions, LabVIEW section.
F	12/2005	Various	Software/Clerical update
G	12/2007	Various	Removed LabVIEW Section

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SOFTWARE LICENSE AND POLICIES

THIS PROGRAM LICENSE AGREEMENT IS DISPLAYED CONSPICUOUSLY ON THE PACKAGE SO THAT YOU CAN READ IT BEFORE OPENING THIS PACKAGE. RETAINING POSSESSION OF THE PACKAGE AND USING THE PROGRAMS AND MATERIALS CONTAINED IN IT INDICATES THAT YOU HAVE AGREED TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE UNOPENED TO DDC AND YOUR MONEY WILL BE REFUNDED.

LICENSE. Data Device Corporation, ("DDC") grants a license to the individual and/or corporate licensee to use the Program Product (the set of computer programs and machine readable and printed materials supplied with this package) for a term beginning on the date you accept this Agreement and continuing until termination. This license agreement is automatically terminated if you violate any of the terms and conditions of this License Agreement. You may terminate this agreement at any time by destroying all copies, modifications, and original materials supplied with the Program Product and destroying any portions of the DDC Program Product merged in any form.

The user License Agreement basically states the following:

- You may use the libraries on as many PC's as you wish.
- Portions of the object code may be included in User Programs that you write and develop.
- There are no royalties. The purpose of this product is for you to create applications or other computer programs (other than C library products) which you may distribute freely and which may contain portions of the object code from this library.

USE OF THE PROGRAM PRODUCT. You may use the Program Product only on the Card with which the product was purchased. You may copy the program for backup or transfer to any machine. Reverse compiling and disassembling of programs in object format are prohibited. Title to the program is not transferred to you by this license. Any attempted sublicense, assignment, or other transfer of the programs or the rights and obligations of this Agreement without the prior written consent of DDC shall be void.

RESTRICTIONS. You may not use, copy, modify, merge, or transfer the Program Product or any portion or modification thereof in whole or in part, except as expressly provided in this Agreement.

LIMITED WARRANTY. DDC warrants the distribution media upon which the Program Product is provided to be free from defects in materials and workmanship under normal use for a period of one year from the date of purchase by the original purchaser. FOR BREACH OF ANY WRITTEN OR IMPLIED WARRANTY ON THIS PRODUCT, THE CONSUMER IS LIMITED TO THE REPLACEMENT OF THE DEFECTIVE DISKETTE(S).

NOTE: SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

DDC MAKES NO OTHER WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AND NO WARRANTY THAT THE PROGRAM PRODUCT IS FREE OF ERRORS. DDC and its authorized dealers shall not be liable for the cost of any repair of correction required for defective DDC Program Product material except as stated in this paragraph.

GENERAL TERMS. This is the only agreement between User and DDC regarding this Program Product and it may be modified only by a written agreement between those parties. This agreement cannot be modified by purchase orders, advertising or other representation by any person. If any provision of this agreement shall be held invalid, illegal, or unenforceable, then the validity, legality and enforceability of the remaining provisions shall not be affected or impaired thereby. This agreement shall be governed by the laws of the State of New York. By opening or keeping the package, you acknowledge that you have read this agreement, understand it and agree to be bound by its terms and conditions.

Trademarks

All trademarks are the property of their respective owners.

Update Policy

DDC will update the product as new functions are developed, to the extent of the warranty period. Further information will be posted, as it becomes available.

Technical Support

In the event that problems arise beyond the scope of this manual contact DDC by calling:

US Toll Free Technical Support: 1-800-DDC-5757, ext. 7771

Outside US Technical Support: 1-631-567-5600, ext. 7771

Europe:
44 (1635) 811140 Fax: 44 (1635) 32264

Asia/Pacific:
81 (3) 3814-7688 Fax: 81 (3) 3814-7689

Or by faxing 1-631-567-7358 to the attention of
Motion Feedback Technologies Applications.

Customer support:

1-800-DDC-5757, ext. 7382

Headquarters:

1-800-DDC-5757

Web Site

DDC also has an Internet World Wide Web site, which allows customers to easily download new revisions of software and documentation. The Internet address is www.ddc-web.com

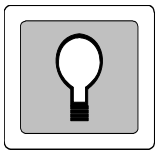
HOW TO USE THIS MANUAL

This manual uses typographical and iconic conventions to assist the reader in understanding the content. This section will define the text formatting and icons used in the rest of the manual. This manual is formatted with a 'Scholar Margin' where many tips, symbols or icons will be located.

Text Usage

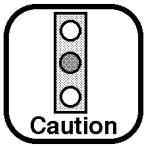
- **BOLD**–text that is written in bold letters indicates important information and table, figure, and chapter references.
- ***BOLD ITALIC***–will designate DDC Part Numbers.
- `Courier New`–is used to indicate code examples.
- <…> - Indicates user entered text or commands.

Symbols and Icons



The Idea/Tip icon will be used to identify a handy bit of supplementary information that may be useful to the user.

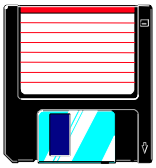
The Note Icon signifies important supplementary information that will be useful to the user.



The Caution icon identifies important information that presents a possibility of damage to the product if not heeded.

Much stronger than a Caution, the Warning icon presents information pertaining to hazards that will cause damage to the product and possible injury to the user.

The Reference icon indicates that there is related material in this manual or in another specified document.



The Disk Icon describes information that is related to software.

Special Handling and Cautions

The ***SB-3622X series*** uses state-of-the-art components, and proper care should be used to ensure that the device will not be damaged by Electrical Static Discharge (ESD), physical shock, or improper power surges and that precautions are taken to avoid electrocution.

- Turn off power to the PC and unplug from wall.
- **NEVER** insert or remove card with power turned on.
- Ensure that standard ESD precautions are followed. As a minimum, one hand should be grounded to the power supply in order to equalize the static potential.
- Do not store disks in environments exposed to excessive heat, magnetic fields or radiation.

INTRODUCTION

The ***SB-3622X*** provides intelligent interfacing between a PCI Bus and a Synchro or Resolver's input. The ***SB-3622X*** is packaged on a full-size IBM PC/AT printed circuit board with a PCI bus interface. Features of the board include:

- Plug-and-Play (PnP) compatible for easy installation
- Utilizes up to six ***DSC-11520***, ***DSC-11524*** or ***DRC-11525*** converters
- On-Board Oscillator option
- Transformer Isolation Available for 60 Hz or 400 Hz

Application software has the capability to simulate position for up to six channels of single speed or three channel pairs of two-speed. Included with each ***SB-3622X*** PCI Card is the DDC Toolbox library with example source code that allows users to easily write their own application software for the card.

What is included in this manual?

This manual contains a detailed installation guide for the ***SB-3622X*** PCI Card and a basic overview of the software supplied with the card. Toolbox Library, DLL, and Driver installations for Windows® 95, 98, 2000, Windows NT® and Linux will be covered in this manual.

The library software provides a level of abstraction such that it is not necessary to understand the operation of the chip set. This manual provides an introduction to the library.

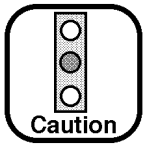
System Requirements

The PC must be configured with a Windows 95, 98, 2000, Windows NT or Linux operating system.

HARDWARE INSTALLATION

Card Installation

The ***SB-3622X*** card is a PCI Target device and may be inserted into any PCI 2.1, or higher, compatible slot. The card is designed for a +5V PCI slot, and will not operate in a +3.3V PCI slot. When installing the card, the following should be observed:



NEVER insert or remove the card with the power turned on.

ALWAYS take proper precautions to guard against static damage. Use a wrist strap if available, or ensure proper static grounding by touching the power supply cover **WITH POWER OFF**.

Insert the card at a slight angle so that the connectors first protrude from the rear opening and then gently press the card into the motherboard connector. Secure with proper hardware.

Make sure that adjacent cabling and wiring do not hinder the airflow around the card.

This card is designed as a Plug-and-Play device and as such, there are no jumpers or switches to be set for address and interrupt selection. **Figure 1** shows a block diagram for the Digital-to-Synchro/Resolver PCI Card.

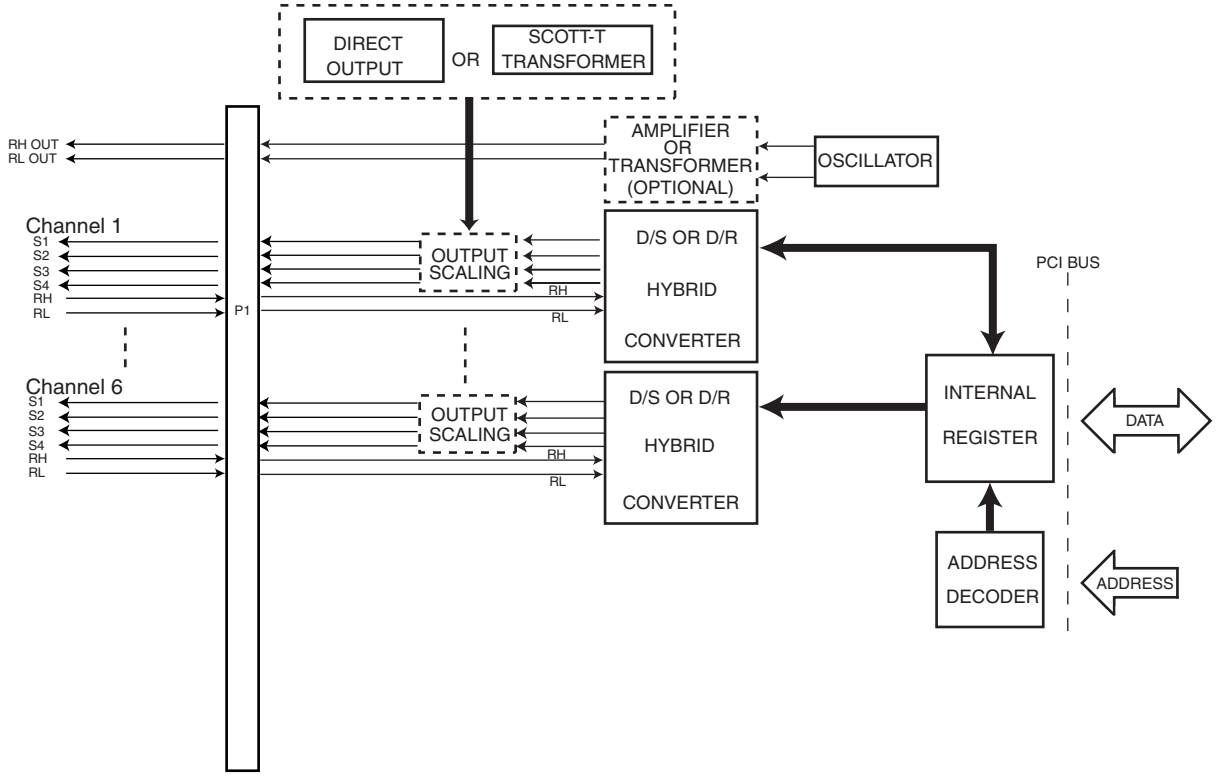


Figure 1. SB-3622X PCI Card Block Diagram

Windows 9x/2000 Driver Installation

After successfully installing the hardware as described in the **Card Installation** section on page 2, install the drivers. For a Windows 9x/2000 operating system continue with the following installation instructions. If you are using Windows NT, refer to page 7 for installation instructions. If you are using Linux, refer to page 7 for installation instructions.

1) Turn on the computer.

2) The **Add New Hardware Wizard** window will appear after boot-up:

Figure 2. Add New Hardware Wizard Screen

3) Click on the **Next** button.

Figure 3. Search For New Driver Screen

4) Click on Search for the best driver for your device.

5) Click on the **Next** button.

Figure 4. Search This Location For Driver Screen

6) Insert the MFT CD into the CD drive.

7) Select the **CD-ROM drive** option.

8) Select the **Specify a location** option.

9) Type the path of the ***SB-3622XS0*** folder on the MFT CD (Example: E:\Software\SB36220 – where E is the CD drive letter)), or Browse to locate the SB36220 folder.

10) Click on the **Next** button to locate the ***SB-3622X*** device driver.

Figure 5. Install Driver From Location Screen

11) Click on the **Next** button to install the device driver.

Figure 6. Driver Installation Complete Screen

12) Click on the **FINISH** button to confirm the installation.

13) Proceed to the **Hardware Configuration** section on page 10.

Windows NT Driver Installation

After successfully installing the hardware as described in the **Card Installation** section on page 2, install the drivers. For a Windows NT operating system continue with the following installation instructions.

1) Turn on the computer.

2) Insert the MFT CD into the CD drive.

3) The CD will auto start.

4) Select the ***SB-36220*** card software to install.

5) Follow any instructions that may appear on the screen. The computer will need to be rebooted for the device driver to run. Once rebooted, proceed to the **Motion Feedback Board Manager** section on page 13.

Linux Driver Installation

Linux is a free Unix-type operating system developed under the GNU General Public License. The source code for Linux is freely available to everyone. Since it's inception in 1991, individuals, university students and companies began distributing Linux with their own choice of packages bound around the original kernel. This is where the concept of the "distribution" was born. This also complicates providing Linux support for our hardware because there are over 150 distributions of Linux. Red Hat Linux is one of the most popular distributions in the world. For this reason we have developed and tested our Linux support on one of the many Red Hat distributions. **This guide was prepared using Red Hat® Linux® 7.2.**

To complete the Linux setup, the user should be familiar with basic Linux procedures and commands (mounting, compiling, etc.). If you are unfamiliar with Linux, try visiting some of these sites for helpful documentation:

- www.ora.com
- www.kernel.org
- www.linux.org
- www.linuxmall.com
- www.amazon.com
- www.ssc.com
- www.linuxhq.com

Documentation can also be found at any of the following major distribution sites for Linux:

- www.redhat.com

- www.caldera.com
- www.debian.org
- www.slackware.com
- www.suse.com

The following installation procedure is also described in the Install.txt and Readme.txt files provided in the *SB3622X_X_Y_Z.TGZ* archive.

1. **Create a new (working) directory called '*synchro*' and go to it.**

   ```
   mkdir synchro
   cd synchro
   ```

2. **Go to super user 'su' (root) mode.**

   ```
   su
   ```

3. **Copy the synchro archive to the working directory and uncompress (TAR/G-ZIP) it.**

   ```
   tar –xzpPf SB3622X_X_Y_Z.TGZ
   ```

4. **Create the shared library links and cache for runtime linking of the synchro library.**

   ```
   /sbin/ldconfig
   ```

5. **Change ownership of all of the files to normal user and then exit superuser mode.**

   ```
   chown –R users.username *
   exit
   ```

6. **As superuser locate the PCI card information by viewing the pci section of the proc kernel interface.**

   ```
   cat /proc/pci
   ```

The Vendor ID is '4ddc', and the Device ID is '0600' for the *SB-36220* card. Make a note of the data listed for the Logical Device #, Vendor ID/Device ID, Bus #, and Device # (or Slot #).

7. **Edit */usr/lib/ddc/DDC_synchro.cfg* to set the correct data for the Logical Device #, Vendor ID/Device ID, Bus #, and Device # (or Slot #) previously noted in step 6. Exit superuser mode.**

Logical Device #	Bus Type	Device ID	Bus #	Slot#	Channel #
0	1	4ddc0600	0	10	0

```
pico /usr/lib/ddc/DDC_synchro.cfg
exit
```

Note: *The bus and channel numbers are not used for the* ***SB-3622x*** *card and should be set to 0. All numbers are in decimal format.*

8. **The driver is distributed as a versioning enabled loadable kernel module, therefore module support (CONFIG_MODULES) and versioning (CONFIG_MODVERSIONS) must be compiled into the kernel in order to use the card and RTL. The driver source is included with the distribution in the event that any kernel symbols or data types are not compatible with the driver binary. Incompatibility is evident when the driver fails to load into the kernel. If a rebuild is required, simply run 'make SB3622.0' in the 'synchro' directory as unprivileged user then copy 'SB3622.0' to '/usr/lib/ddc' as 'root'.**

9. **Loading and unloading of the driver is accomplished by running the appropriate script as superuser. 'usr/sbin/SB3622X_load' will create the required device special files and load the driver into the kernel, and 'usr/sbin/SB3622X_unload' does the reverse. Calls to these scripts can also be made at startup or login time in the runtime command (rc) section of the '/etc'tree, or user startup files, respectively. To gain a better understanding of what the scripts do, it is suggested that the user view them.**

10. **Compile the sample application.**

```
cd synchro
make 3622 test
cd samples
```

11. **Execute the sample application .**

```
./3622_linux
```

After successfully installing the drivers, refer to step 3 in the **Console Application Section** on Page 28.

Hardware Configuration

The ***SB-3622X*** is a PCI device, and as such does not require jumpers or switches to set the Base address or interrupt values. The job of configuration for Plug-and-Play PCI configuration is performed by the PC BIOS. During the initial power on boot process, the BIOS performs an enumeration of the PCI bus and provides a configuration in the system that satisfies the card requirements. The card provides the BIOS with details of how much memory it requires, and any other operating parameters that the system needs to know by way of configuration registers built into the card. These registers are configured at the factory to contain the optimum values for the operation of the ***SB-3622X***. The ***SB-3622X*** PCI card and software drivers allow for shared interrupts, thus simplifying the installation and reducing the risk of device conflicts.

Once the card has been properly installed, its configuration information is stored to the Windows Registry. The card information stored in the Registry is accessed and controlled through the use of the **Motion Feedback Board Manager** Application. The **Motion Feedback Board Manager** Application can be used to view the status of installed cards, and map an installed card to a device number. In order to access an installed card you **must** assign it a device number.

Proceed to the **Motion Feedback Board Manager** section on page 13.

APPLICATIONS SOFTWARE INSTALLATION

Windows Software Overview

There are several software applications which are used in conjunction with ***SB-3622X*** PCI Cards. These applications are included with your card, and include a Console Application, a Graphical User Interface and a Synchro.dll file.

If you did not receive the application software described on the following pages with your card, please contact DDC customer service.

MOTION FEEDBACK BOARD MANAGER APPLICATION

A MOTION FEEDBACK BOARD MANAGER Application contained in the CONTROL PANEL is used to define each Synchro/Resolver card type in the system and the address for each Synchro/Resolver card. The MOTION FEEDBACK BOARD MANAGER Application is installed in the CONTROL PANEL during the setup.exe installation. The MOTION FEEDBACK BOARD MANAGER Application will route the information to the appropriate routine for the card type selected.

CONSOLE APPLICATION

An executable application using the synchro.dll is included with the ***SB-3622X*** PCI card to demonstrate the capabilities of the card. Upon installation, there will also be source code examples of a program that can be compiled.

SYNCHRO.DLL FILE (FUNCTION CALLS)

The synchro.dll (Dynamic Link Library) has been created to provide a programmer with routines (Function Calls) that perform the low-level read and write functions to the registers on the Synchro/Resolver cards. The **Function Calls** section of this manual describes the routines available from the synchro.dll.

DATA SIMS

dataSIMS software support is available, contact DDC Software Applications for details.

Windows Software Installation

The ***SB-3622X*** PCI Card is supplied with a GUI, a console application, and Runtime Libraries for Windows 95/98 and Windows NT/2000.

The runtime library provides the user with a hardware abstraction layer for the DDC ***SB-3622X*** PCI series Digital-to-Synchro/Resolver. This software layer includes routines that dramatically reduce software development time by providing a high level interface to the PCI card.

16-bit operating systems not supported

If you are installing under Windows perform the steps below:

1) Insert the MFT CD into the CD-ROM drive, and allow the CD to auto start.

2) Choose your product (SB36220).

3) Choose your software (SB36220).

4) Click on the Install Software icon.

5) Follow the on screen instructions.

6) When the installation is successfully completed, proceed to **Motion Feedback Board Manager** section to configure the card. See page 13.

MOTION FEEDBACK BOARD MANAGER

Once you have installed the hardware and software, you are now ready to configure the card. Device configuration is performed through the use of the **Motion Feedback Board Manager** Application.

1) Turn on the computer.

2) Click on Windows **START**.

3) Click on **SETTINGS**.

4) Click on **CONTROL PANEL**.

5) Double-click on the **Motion Feedback Board Manager** icon (circled in the screen image below).

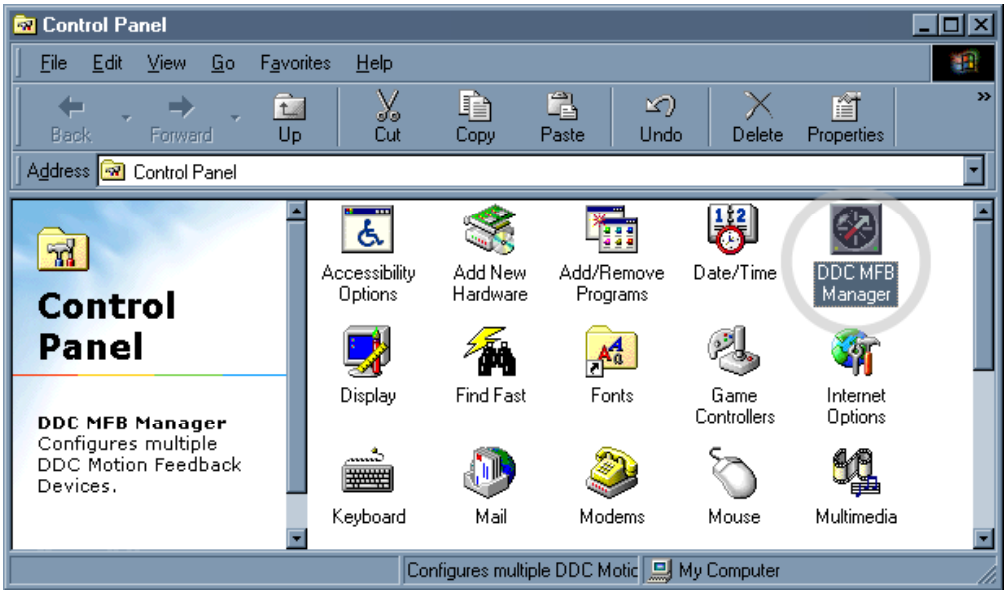


Figure 7. Control Panel window

6) Select **Show installed devices** from the Motion Feedback Devices window to display status information for the installed card(s).

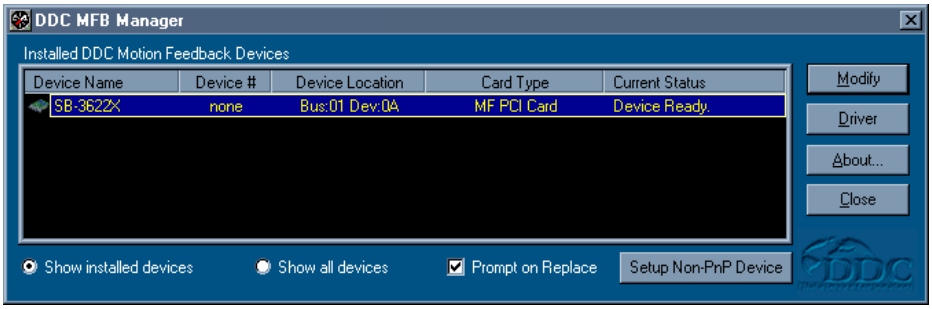


Figure 8. Motion Feedback Devices window

Note: *The **Device #** will default to none. Each card must be assigned to a unique (unused) device number from 0 to 31, card numbers cannot be shared.*

7) To assign the card to a new Device number, select **Modify** to open up the Modify Device window.

Figure 9. Modify Device window

8) Click on the **Device Number** arrow to open up a pull down window of device number selections.

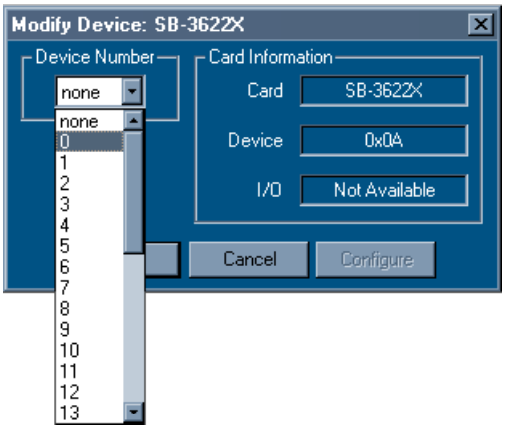


Figure 10. Modify Device pull down menu

9) Select an unused device number from the pull down menu (Currently only device numbers 0 to 7 can be selected).

Note: *If a previously assigned device number is selected, and the* ***Prompt on Replace*** *option is checked on the Motion Feedback Devices menu, the following warning will appear. Clicking on* ***Yes*** *will overwrite the previously assigned device number. If the* ***Prompt on Replace*** *option was not selected, a warning will not be displayed and the new device number will automatically overwrite the previously assigned device number.*

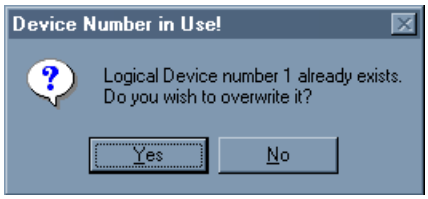


Figure 11. Device Number in Use! warning

Figure 12. Modify Device Window (close)

10) Select **OK** on the Modify Device window to return to the Motion Feedback Devices menu.

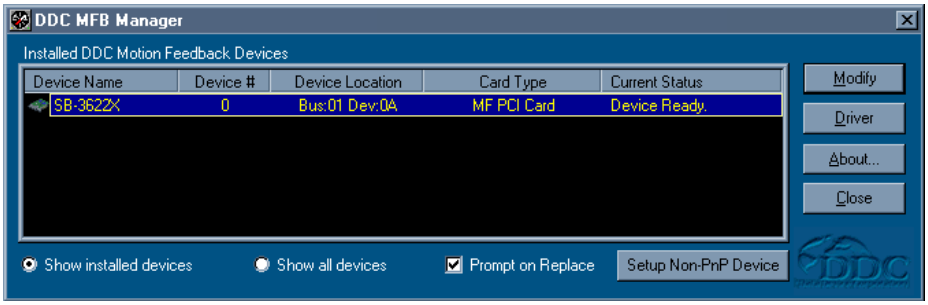


Figure 13. Motion Feedback Devices window (Device # assigned)

Note: *The **Motion Feedback Devices** window should display the device assigned to the selected device number.*

11) To show a log of all devices installed (even if the card is not currently inserted), select **Show all devices** from the **Motion Feedback Devices** window.

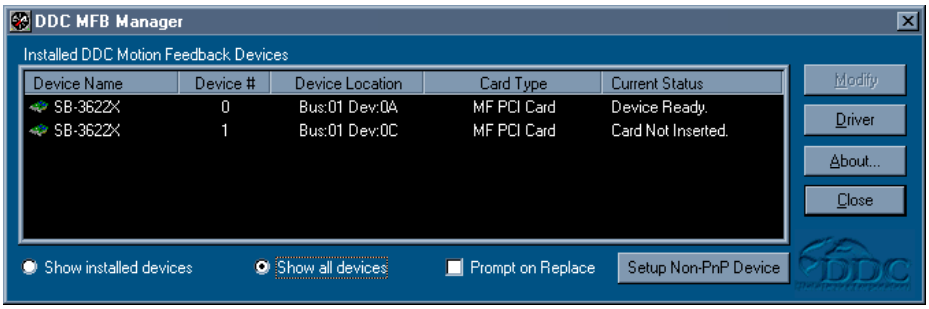


Figure 14. Motion Feedback Devices window (Show all devices)

Note*: The **Motion Feedback Devices** window will display all cards currently installed.*

12) To display the status of the device drivers, select **Driver** from the **Motion Feedback Devices** window. The following Device Driver Information window will be displayed.

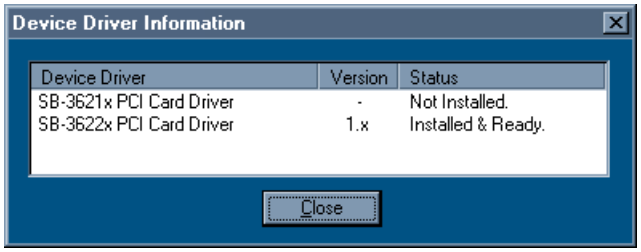


Figure 15. Device Driver Information window

13) Select **Close** to close the Device Driver Information window and return to the Motion Feedback Devices window.

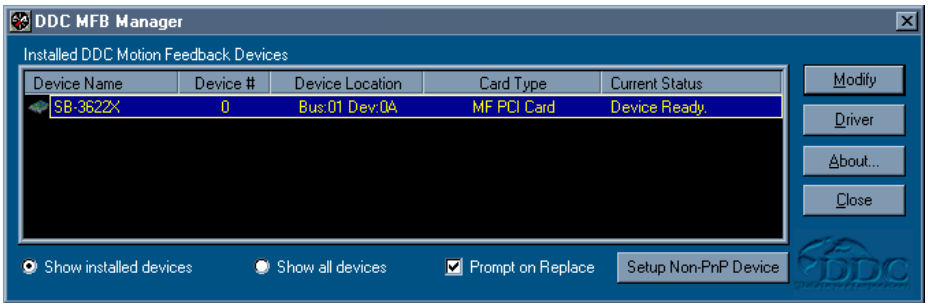


Figure 16. Motion Feedback Devices window (close)

14) Select **Close** to complete the card configuration and close the **Motion Feedback Devices** window. The card is now ready for use.

Graphical User Interface (GUI) Application

The Graphical User Interface allows the user to display and set card parameters in analog, digital and binary formats. Parameters are set and display options are selected using a combination of keyboard and mouse commands according to the chosen format.

1) To access the Graphical User (GUI) Application Interface click on **Start**, **Programs**, **Data Device Corporation**, **SB3622x**, and **SB-3622x Application**, as shown below.

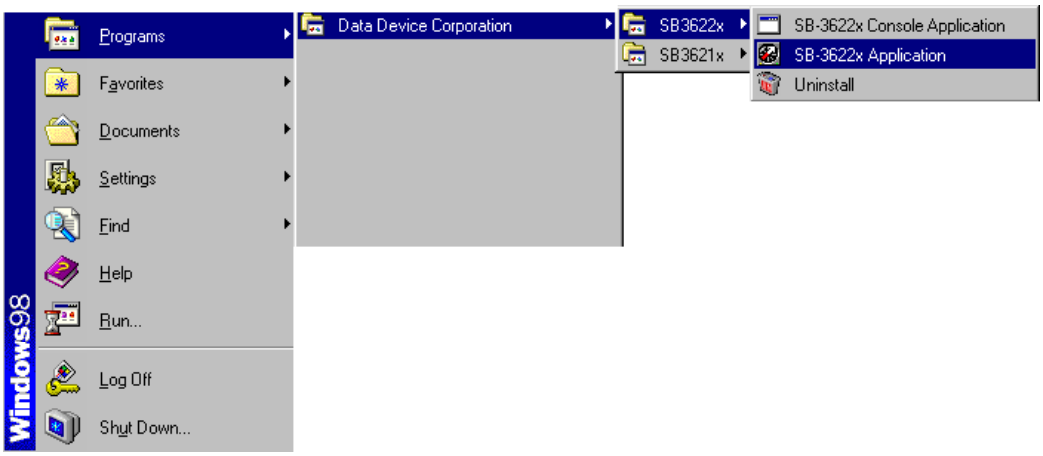


Figure 17. Accessing the GUI Application

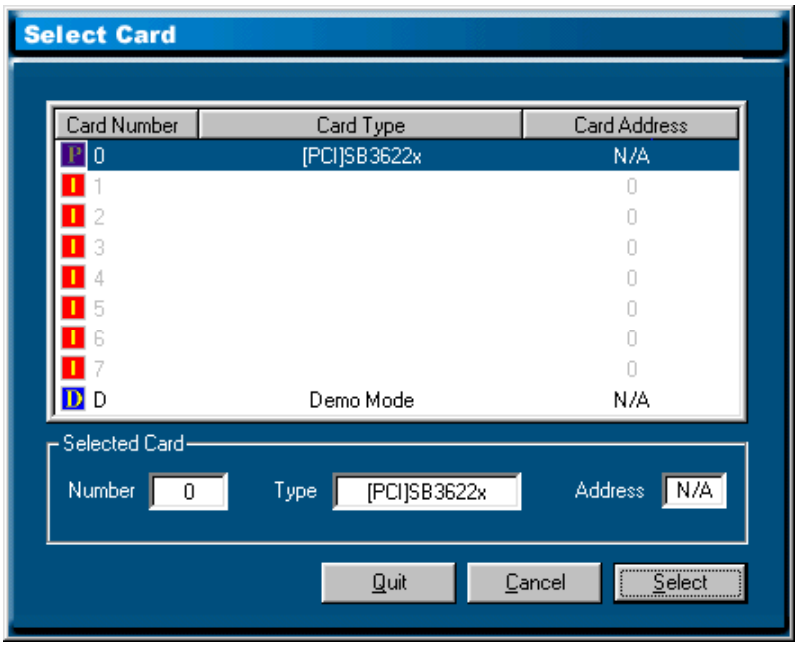


Figure 18. Select Card Screen

2) Select a card from the drop down list, then click **Select**.

Note: *Selecting Demo Mode will open up the following demo application screen where input information can be simulated, even if a card is not installed.*

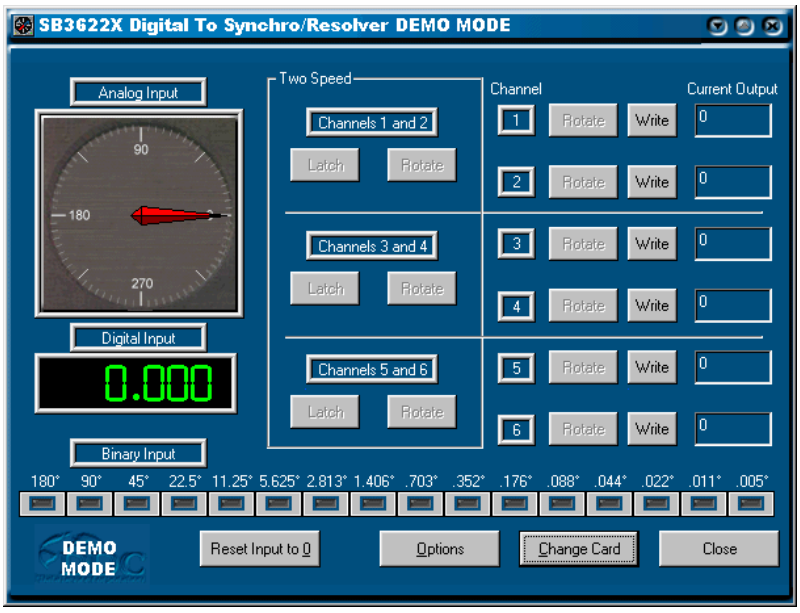


Figure 19. DEMO Application Screen

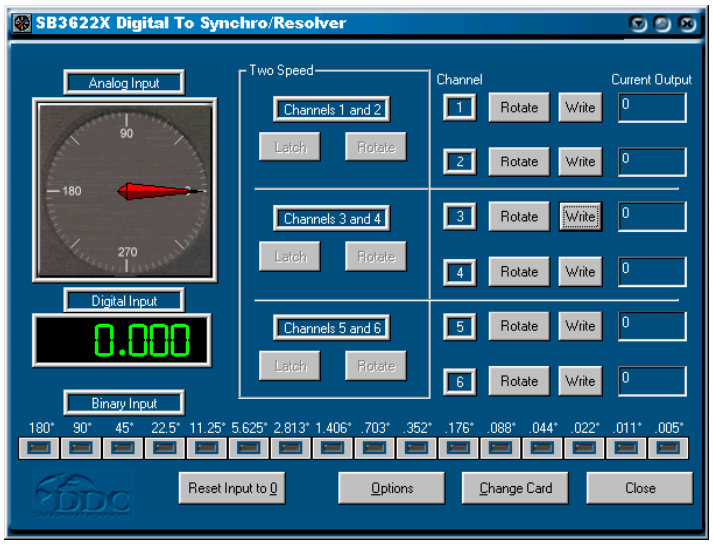


Figure 20. GUI Application Screen

3) To set the oscillator amplitude and frequency click **Options**, this will open the menu shown below. From this menu the display information can be configured along with the hardware configuration.

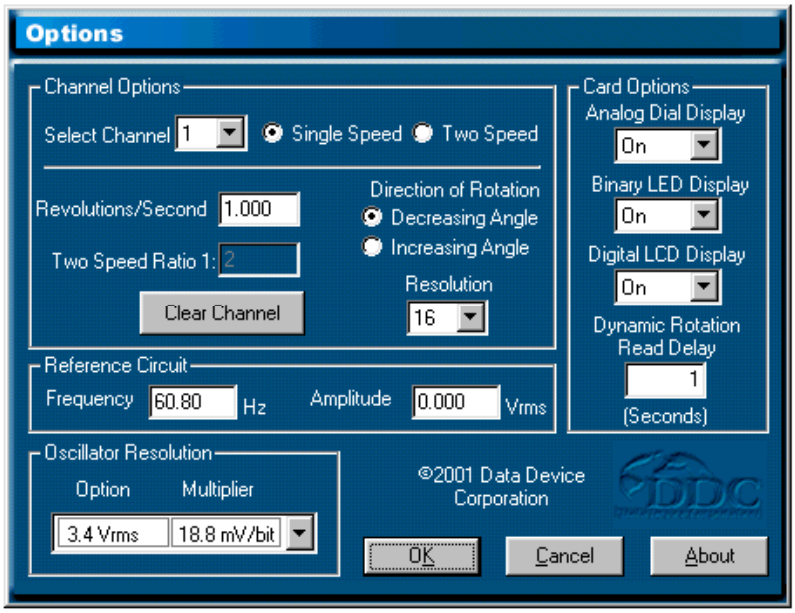


Figure 21. Options Screen – Oscillator Resolution Shown

4) Click on the Oscillator Resolution scroll bar to open up a pull down menu of oscillator ranges. The model number purchased determines the oscillator range. Select the mV/bit of the card installed.

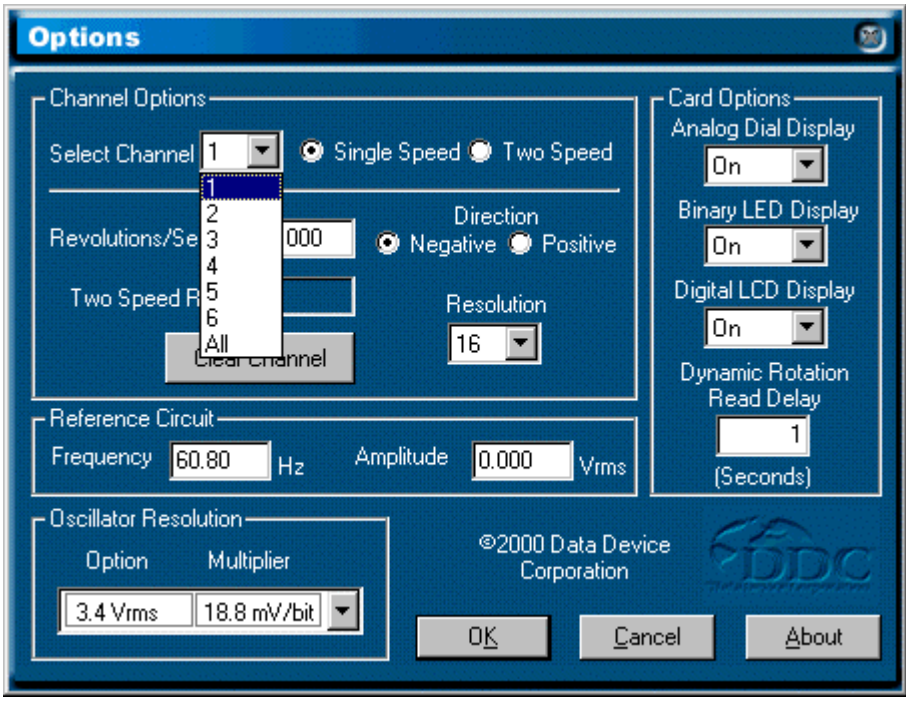


Figure 22. Options Screen – Select Channels Shown

5) Channel option selections are made per channel or for all channels at one time. By selecting a specific channel, options now changed are relevant to that specific channel only. By selecting all channels, changes are relevant to channels 1 through 6. For two-speed applications select the 2 channels, set each channel for two-speed and set each channel's two-speed ratio.

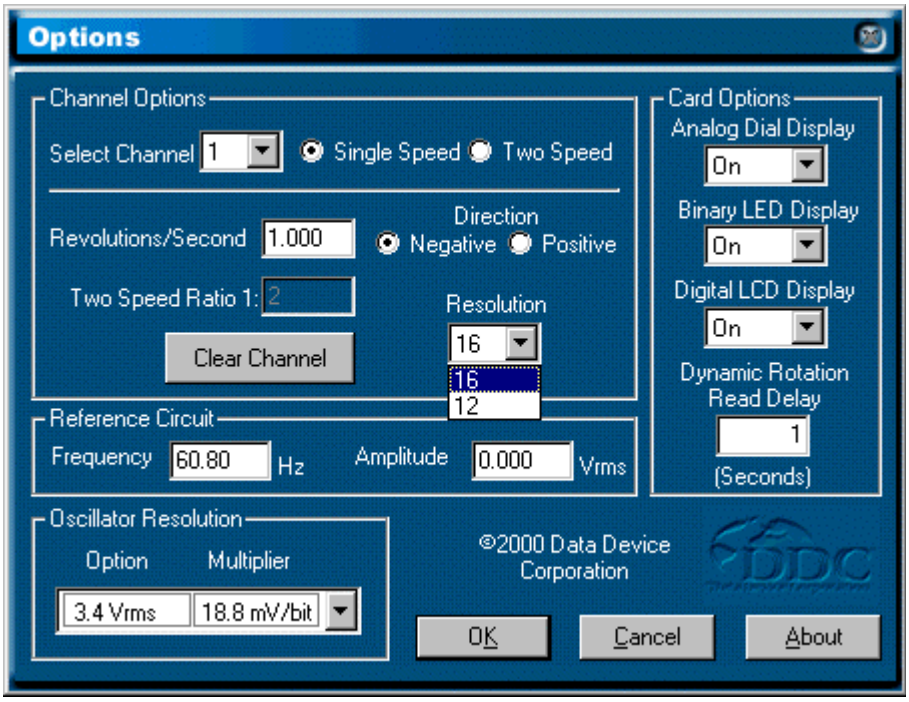


Figure 23. Options Screen – Resolution Shown

6) Click on the Resolution scroll bar to select either 12 or 16 bit resolution.

Note: *The application will automatically recalculate the Revolutions/Second when the resolution is changed.*

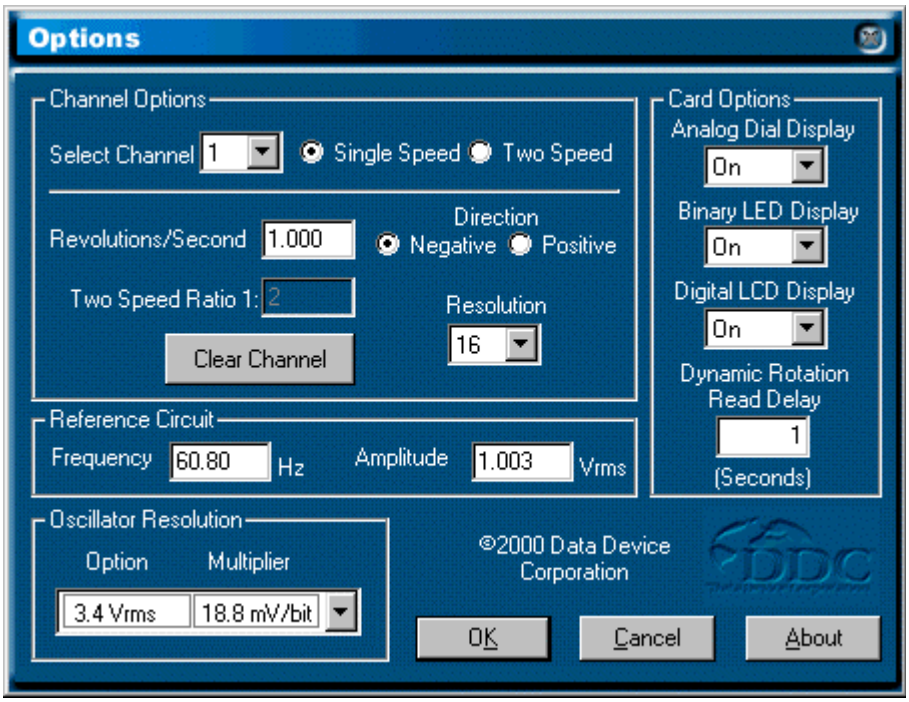


Figure 24. Options Screen – Reference Frequency and Amplitude Shown

7) Once the appropriate oscillator is selected the oscillator amplitude and frequency can be programmed by inputting the values into the Reference Circuit windows. The Amplitude and Frequency displays will update with the closest system values available.

8) The Card Options section enables which type of display is shown in the main screen. The binary display represents the digital angle, (Bit 1 = 180°, Bit 2 = 90°, Bit 3 = 45° etc…)The read delay defines the display update rate.

9) Once the options are selected click on **OK.** The main screen will display angular position.

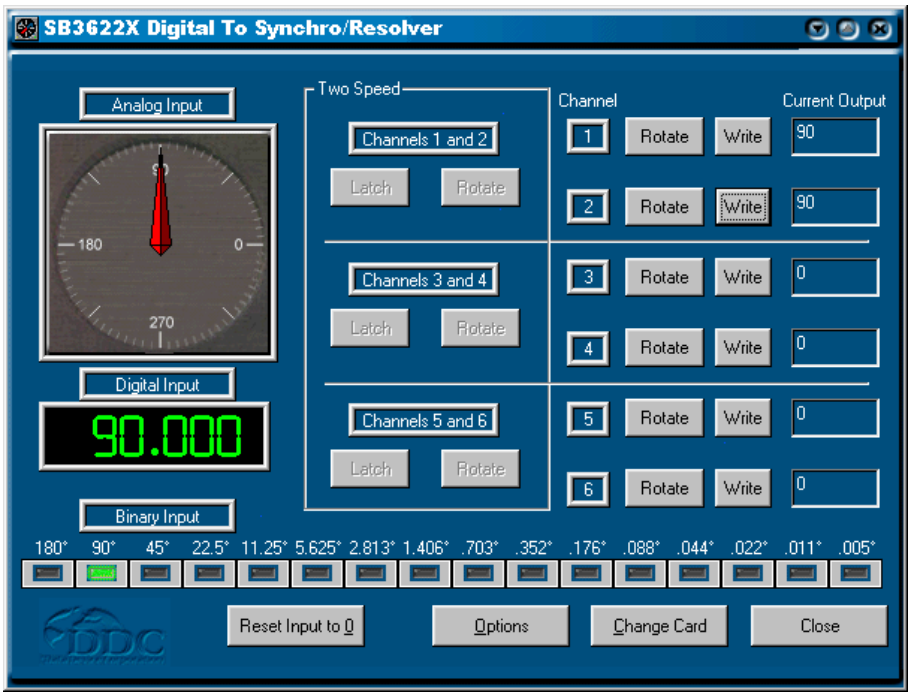


Figure 25. Angular Position Screen – One-Speed

DISPLAY OPTION	EFFECT
Analog Input	Allows the user to set the input angle by using the mouse to move the analog input dial to the desired angle.
Digital Input	Allows the user to set the input angle by typing in a numerical angle in the Digital Input box and pressing enter.
Binary Input	Allows the user to set the input angle by using the mouse to click and toggle on and off the LED Binary Input buttons.
Two Speed Rotate	Starts and stops dynamic rotation for a channel pair (1, 3, 5).
Two Speed Latch	Latches the data for a channel pair when the coarse angle crosses zero.
Channel Rotate	Starts and stops dynamic rotation for an individual channel.
Channel Write	Writes the input angle to the corresponding channel on the card. The written angle is displayed in the Current Angle window.
Reset Input to 0	Resets the input angle to zero.
Current Output	In Single Speed Mode, displays the current angle position of each channel. In Two Speed Mode, displays the Coarse and Fine angles of each channel pair.

10) The ***SB-3622X*** series cards can also be used to interface with two-speed resolvers and calculate the combined position between the coarse and fine channels based on the two-speed ratio selected.

11) If the two-speed option is selected the Two-Speed section of the main screen will be active (highlighted), and the Single-Speed section will be inactive (unhighlighted). The Current Output windows will display the Coarse and Fine position information of each channel pair. Two-Speed channel pairs are (1 and 2), (3 and 4), and (5 and 6).

Note: *Two-speed allows resolutions greater than 16 bit to be achieved. For Two-speed theory review the following DDC documents. The RD/RDC applications manual system design section, P/N MN-19220XX-001 and the Synchro/Resolver Conversion Handbook. Both of these documents can be downloaded from the DDC website at www.ddc-web.com.*

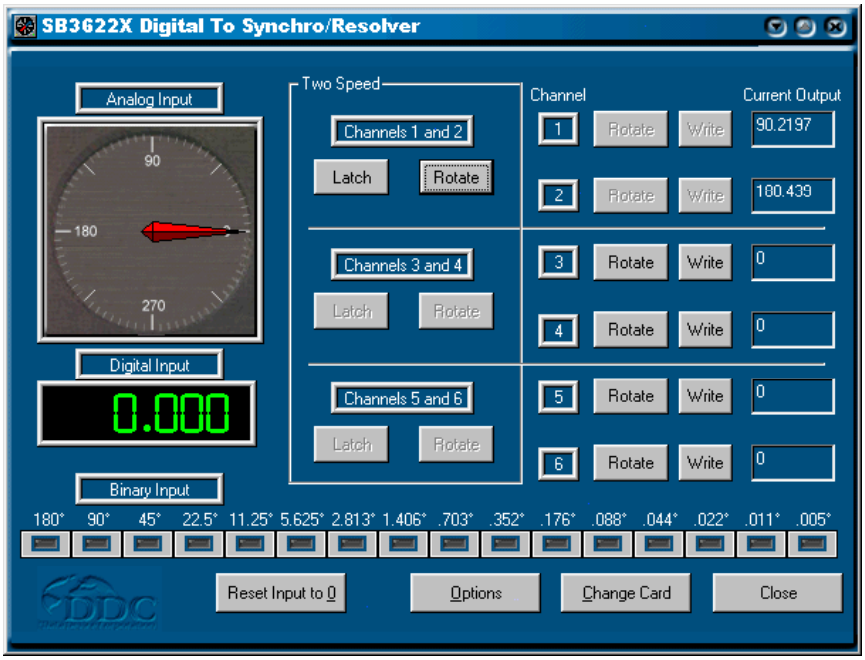


Figure 26. Angular Position Screen – Two-Speed

12) To exit the application screen, select **Close**. The following warning will be displayed. Click on **Yes** to exit.

Note: *Option settings will be saved upon exit from the Application menu. These settings will remain active from one session to the next, provided that the logical device number is not changed or the card removed.*

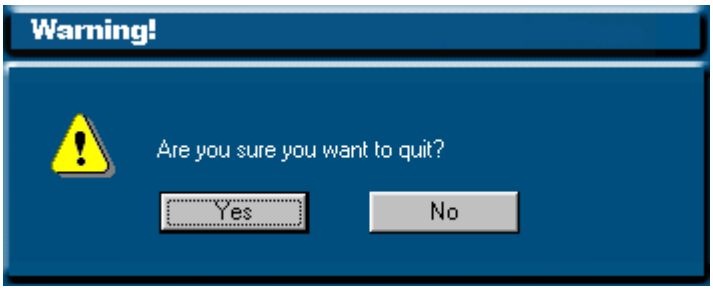


Figure 27. Exit - Warning

Console Application

Note: *For Linux installations refer to the "Readme.txt" supplied with the Linux software.*

The Console Application is an easy to use application that enables engineers and maintenance personnel to test and maintain Synchro/Resolver systems. This program allows easy access to data analysis of Synchro/Resolver outputs. Some features include:

- Self test
- Angle position data
- Inhibit data
- Change bandwidth and resolution
- Reset counters
- Change to encoder mode
- Change oscillator amplitude and frequency

The underlying source code for the Console Application is provided as an example program and is located at C:\Program Files\Data Device Corp\SB-3622x\3622x.cpp .

1) To access the console application click on **Start, Programs, Data Device Corp, SB 3622X** and **SB 3622X Console Application**, as shown below.

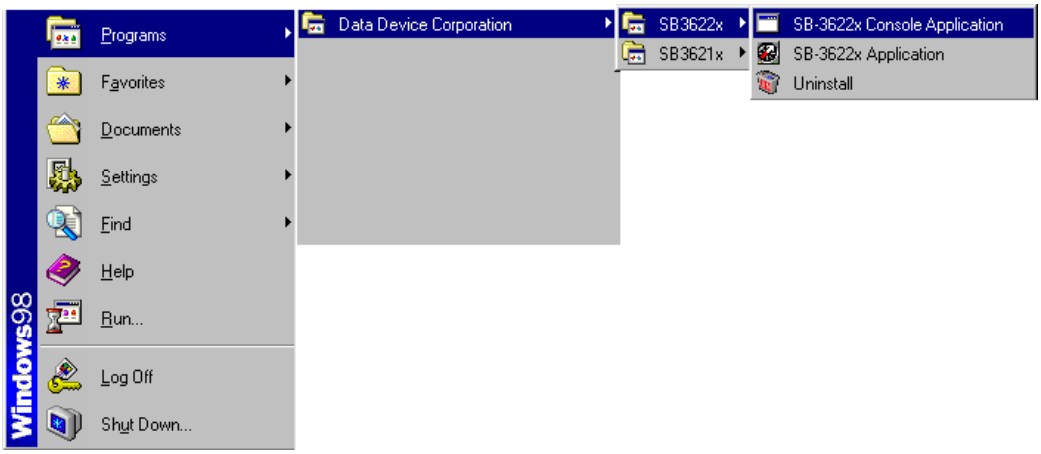


Figure 28. Accessing the Console Application

2) This will launch the console application. The console application provides an easy method to evaluate angle position information.

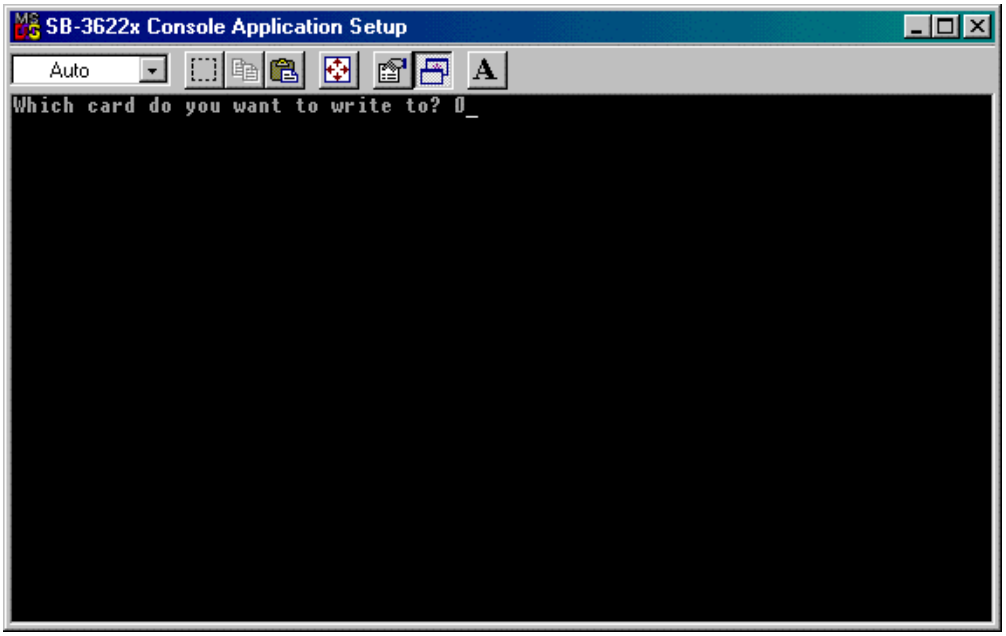


Figure 29. Console Application Setup

3) Enter the card number of the ***SB-3622X*** card and hit Enter. This will open up the following Console Application Menu.

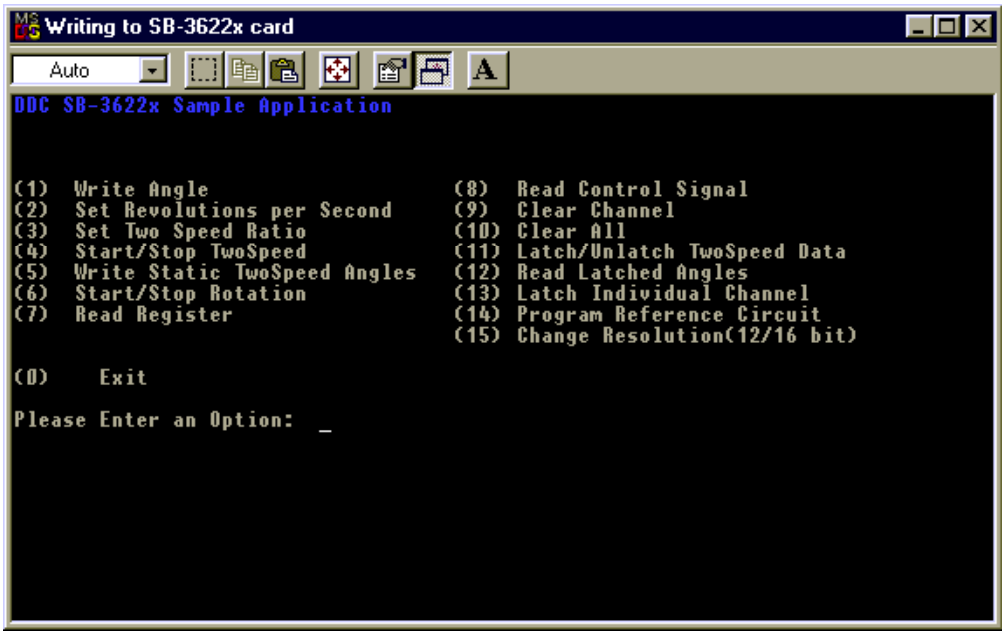


Figure 30. Console Application Menu

CONSOLE APPLICATION MENU OPTIONS

To enter menu options use references below. Channel (1-6) represents channels 1, 2, 3, 4, 5, and 6.

Channel Pair (1-3) represents channel pairs 1 (channels 1 and 2), 2 (channels 3 and 4), and 3 (channels 5 and 6).

Type the code for the desired operation in the **Enter Option:** field and hit **Enter**.

(0) Exit

Exits out of the program.

(1) Write Angle

This routine is called when the user wants to write a single, static angle to the card. The user is asked for a channel to write to (1-6), and the angle to write (0 -359.99). If all parameters are valid, the angle is written to the card, and the angle is displayed on the screen.

(2) Set Revolutions per Second

This function will change the Binary Rate Multiplier for the user-defined channel. The user is asked “Is this a Two Speed Multiplier (1) or Single Speed (0)?”, then which channel (1-6) or pair of channels (1-3) depending upon the previous selection, the number of revolutions per second (0.03-125.0), and a direction (1-up, 0-down). If all parameters are valid, the values are written to the given channel.

(3) Set Two Speed Ratio

Sets the Two Speed Ratio for a user-entered channel pair (1-3). The user is asked for a channel pair, and the ratio (1-128). The routine then writes that data to the card.

(4) Start/Stop Two Speed

This routine will Start or Stop Two Speed rotation, based on the Mode. The card will remain in Two Speed operation, but dynamic rotation will be affected. The user is asked which Channel Pair (1-3) to start/stop. Then the user is asked whether to Start (1) or Stop (0) Two Speed Mode.

(5) Write Static Two Speed Angles

For users who just want to have a static Two Speed display, without using the built-in Two Speed operation mode, the user can enter a coarse angle, and the software will calculate and write the fine angle to the card. This is essentially a static write to the coarse angle, and a static write to the fine angle, using software to calculate what the fine angle should be. The Binary Rate Multiplier, and Two Speed ratio need to be set for this to work. The user is asked which Coarse Channel (1, 3, 5) to write a static Two Speed Angle to. Then the user is asked for an angle (0 - 359.99).

(6) Start/Stop Rotation

This routine will Start or Stop Dynamic Rotation, based on the Mode. The card will remain in Rotation operation, but dynamic rotation will be affected. The user is asked which channel (1-6) to start or stop Dynamic Rotation on. Then the user enters a 1 to start or a 0 to stop Dynamic Rotation.

(7) Read Register

If the user wants to read an individual register, this function will display the data from that register. In this application, the user enters the hardware offset, and the function will shift the data left 16, so that it is in the proper format for the device driver.

(8) Read Control Signal

This is a specialized version of Read Register. It only reads the Control Signal. Since this is the primary register for controlling the functionality of the card, a separate function was created. The user is asked which Channel Pair (1-3) should be read. The Control Register is then outputted to the screen.

(9) Clear Channel

Resets the hardware and software settings for the user-entered channel. Each channel can be cleared independent of the other 5 channels. The user is asked which channel (1-6) should be cleared.

(10) Clear All

Resets the hardware and software settings for the user-entered channel pair (1-3). This will clear both the coarse and fine registers.

(11) Latch/Unlatch Two Speed Data

There is the ability in hardware to latch the Two Speed Data for a given channel pair (1-3). This routine implements that hardware function. When the Two Speed data is latched, as soon as the coarse angle hits zero, the data is latched in a register. The user can then read the values, and verify that the values are at zero degrees.

(12) Read Latched Angles

After the Two Speed data has been latched, this routine will read and display the coarse and fine angles for the user-entered channel pair (1-3). If the data has not been latched yet, the user is alerted. If the data has been latched, the coarse and fine angles are displayed. There is a bit that changes when the data is latched. By reading this bit, the routine assures that the data has been latched, and then proceeds to read the angles.

(13) Latch Individual Channel

This function will latch the data for the user-entered channel (1-6). Since this will effectively stop that channel from dynamically rotating, it is not recommended to use this function unless the user is aware of the results.

(14) Program Reference Circuit

For many configurations of the ***SB-3622X*** cards there is an internal reference circuit. This function allows the user to adjust the frequency (0) and amplitude (1) of that circuitry. In this application, the user enters the hex value (0x00 - 0xFF) to write to the card. There is no conversion routine available.

(15) Change Resolution (12/16 bit)

The ***SB3622X*** card can operate in 12 bit or 16 bit mode. This function allows the user to change the channel's resolution. The user is asked the Channel (1-6) to change the Resolution on. The user may change the Resolution to 16 bit (0) or 12 bit (1).

FUNCTION CALLS

*Software routines for the **SB-3622X** Digital-to-S/R PCI Interface Card.*

The following list contains all of the pertinent Function Calls required by the user. Any Function Calls contained within the Synchro.h file that are not listed here are available from the factory upon request.

void WINAPI Clear_All(int Card, int Channel Pairs)

DESCRIPTION

This routine resets the hardware and clears the Coarse and Fine settings for the given channel pairs.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel Pairs = an integer between 0 and 2 for the channel pairs of interest.

0 = channels 0 and 1
1 = channels 2 and 3
2 = channels 4 and 5

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Clear_All(4,0);`	`Resets the hardware and clears channel 0 (coarse) and channel 1 (fine) for card #4.`

void WINAPI Clear_Channel(int Card, int Channel)

DESCRIPTION

This routine resets the hardware and clears either the Coarse or Fine settings for the given channel.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

0 = channel 0 (Coarse)
1 = channel 1 (Fine)
2 = channel 2 (Coarse)
3 = channel 3 (Fine)
4 = channel 4 (Coarse)
5 = channel 5 (Fine)

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Clear_Channel(4,0);`	`Resets the hardware and clears channel 0 (coarse) for card #4.`

void WINAPI Close(int Card)

DESCRIPTION

This routine closes the device driver and writes default values to the card.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Close(7);`	`Closes the device driver for card #7.`

void WINAPI Enable_Rotation(int Card, int Channel)

DESCRIPTION

This routine puts the card into Dynamic Rotation Mode without starting the rotation.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Enable_Rotation(0,1);`	`Puts card #0, channel 1 into Dynamic Rotation Mode.`

void WINAPI Enable_TwoSpeed(int Card, int Channel)

DESCRIPTION

This routine turns the given channels to Two Speed mode, then starts Two Speed. Two Speed Mode is enabled without starting rotation. The Binary Rate Multiplier and the Speed ratio need to be programmed first. If the Binary Rate Multiplier is not programmed for the channel pair, then Two Speed will not continue.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 2 for the channel of interest.

0 = Channel 0 (Coarse) and Channel 1 (Fine)
1 = Channel 2 (Coarse) and Channel 3 (Fine)
2 = Channel 4 (Coarse) and Channel 5 (Fine)

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Enable_TwoSpeed(0,1);`	`Enables Two Speed mode for card #0, channel 2(Coarse) and channel 3 (Fine).`

char* WINAPI Get_Card_Address(int Card)

DESCRIPTION

Reads the Registry and returns the value of the card address at that location. This routine can be used to display the address of the card for informational purposes.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

RETURNED DATA

This routine returns the card address of the card as a char. The card address returned is the physical PCI slot number that the card is installed in (ie. 10, 11… 17). The PCI slot number is assigned by the computer manufacturer. If the routine returns an Error then the OS is not compatible with the card.

EXAMPLE CODE

Code	Result/Returned Data

`char* CardAddress;`
 	
`CardAddress = Get_Card_Address(4);`
	`Returns the address (as a char) for card #4.`

Int WINAPI Get_Card_Type(int Card)

DESCRIPTION

This routine reads the Registry and returns the value of the card at that location. Can be used to error check the card number a user has chosen.

INPUT DATA

Card = An integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

RETURNED DATA

This routine will return the type of card installed and assigned to Card.
Valid Values:

[PCI]SB3620x
[PCI]SB3621x
[PCI]SB3622x

EXAMPLE CODE

Code	Result/Returned Data
`Get_Card_Type(0)`	`Returns the card type of card 0.`

Int WINAPI Get_DLL_Version(void)

DESCRIPTION

This routine returns the current version of the synchro.dll. Can be displayed in applications, and helps to maintain version control.

INPUT DATA

There is no inputted data for this routine.

RETURNED DATA

This routine will return version number of the synchro.dll.

EXAMPLE CODE

Code	Result/Returned Data
`Get_DLL_Version(void)`	`Returns the card version of the synchro.dll`

Int WINAPI Get_Last_Error(char *ErrorMsg)

DESCRIPTION

This routine returns the last error that occurred, and clears the error code to 0. Can be called frequently, if needed.

INPUT DATA

ErrorMsg A char that will hold the last error message.

RETURNED DATA

ErrorMsg A char that will hold the last error message.

Valid Values:

No Error
Could not open Registry
Card Type is invalid
Card not Initialized
Invalid Direction
Could not open Device
Could not close Device
Invalid Channel
Invalid Option
Invalid Range
Missing Parameters
Invalid Error Code
Card In Use

EXAMPLE CODE

Code	Result/Returned Data
`Get_DLL_Version(char *ErrorMsg)`	`No Error`

char* WINAPI Initialize(int card)

DESCRIPTION

Opens and reads from the REGISTRY to get the card type. Performs a string comparison with the card type to branch to the appropriate card-specific routine. Closes the REGISTRY when finished. Returns the card type from the given card number, for error checking.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

RETURNED DATA

The routine will return a string of characters. The returned data should be: [PCI]SB3622x.

EXAMPLE CODE

Code	Result/Returned Data

`char* CardName;`
 	
`CardName = Initialize(0);`
	`Returns card type for card #0`

void WINAPI Latch_Channel(int Card, int Channel, int Latch)

DESCRIPTION

Either Latches (1) or Unlatches (0) the data for a given channel. This routine will stop any dynamic activity; therefore, it should not be used on a regular basis.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

Latch = an integer that is either a (1) to Latch the given channel, or a (0) to Unlatch the channel.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Latch_Channel(4,0,1);`	`Latches card #4, channel 0.`

void WINAPI Latch_TwoSpeed_Data(int Card, int Channel, int Start)

DESCRIPTION

When this function is activated, the Two Speed Data will latch as soon as the Coarse angle hits zero from either direction. Doing this verifies that the two Coarse and Fine angles are synchronized. The angles will remain latched until the unlatch command is given. Latching this data will not affect the operation of the card.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 2 for the channel pair of interest.

0 = Channel 0 (Coarse) and Channel 1 (Fine)
1 = Channel 2 (Coarse) and Channel 3 (Fine)
2 = Channel 4 (Coarse) and Channel 5 (Fine)

Start= an integer that is either a (1) to Latch or a (0) to Unlatch the Two speed data for the given channel.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Latch_TwoSpeed_Data(1,2,1);`	`Latches the Two Speed Data for channel 4 (Coarse) and channel 5 (fine) on card #1.`

void WINAPI Program_Amplitude(int Card, int Amplitude)

DESCRIPTION

This routine allows the user to program the amplitude of the oscillator. Since there is only one oscillator on the card, the channel does not have to be selected.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Amplitude = an integer between 0 and 255 that corresponds to a bit multiplier. The product of the bit multiplier and the output scaling represents the true amplitude.

To change the Oscillator amplitude, enter the bit multiplier. The bit multiplier, which is an integer, is determined by the following formula.

Amplitude = (bit multiplier) X (mV/bit)

Model Number	Output Scaling (mV/bit)	Examples	Bit Multiplier	Resultant Amplitude	Bit Multiplier Range
I4	32.6 mV/bit	3.26 V = (100) X (.0326)	100	3.26 V	0 - 255
I5	240.0 mV/bit	25.92 V = (108) X (.240)	108	25.92 V	0 - 255
I6	635 mV/bit	114.935 V = (181) X (.635)	181	114.935 V	0 - 210

Example: Amplitude = (Bit Multiplier) X (mV/bit)
Amplitude = (100) X (32.6 mV/bit)
Amplitude = 3.26 Vrms

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Program_Amplitude(7,108);`	Assuming card #7 is a model type I5 (240.0 mv/bit), the oscillator amplitude of card #7 will be set to 25.92 V.

void WINAPI Program_Angle(int Card, int Channel, WORD Angle)

DESCRIPTION

This function takes the angle inputted by the user and writes it to the card. This function checks to make sure that the card is not in Dynamic Mode. Therefore, Dynamic Mode is disabled before the angle is written to the card.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

Angle= a WORD that represents the angle that is to be written to the card. The angle must be in the range of 0 to 0xFFFF. Use the Dec to Bin function to convert from a degree-based number to Synchro Binary (i.e. 90 degrees = 0x4000).

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Program_Angle(0,0,0x4000);`	`Writes an angle of 90 degrees to card #0, channel 0.`

void WINAPI Program_Frequency(int Card, int Frequency)

DESCRIPTION

This routine allows the user to program the frequency of the oscillator. Since there is only one oscillator on the card, the channel does not have to be selected.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Frequency = an integer between 0 and 255 that corresponds to a bit multiplier. The product of the bit multiplier and the output scaling represents the true frequency.

To change the Oscillator frequency, enter the bit multiplier. The bit multiplier, which is an integer, is determined by the following formula.

Frequency = (bit multiplier) X (Hz/bit)

Board Type	Output Scaling (Hz/bit)	Examples	Bit Multiplier	Resultant Frequency	Bit Multiplier Range
60 Hz	7.6 Hz/bit	60.8 Hz = (8) X (7.6)	8	60.8 Hz	8 - 255
		402.8 Hz = (53) X (7.6)	53	402.8 Hz	
400 Hz	57.8 Hz/bit	404.6 Hz = (7) X (57.8)	7	404.6 Hz	6 - 255
		2023.0 Hz = (35) X (57.8)	35	2023.0 Hz	

Note: *Model I6 uses 60 Hz multiplier*

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Program_Frequency(5,53);`	`Assuming card #5 is a 60 Hz type, the oscillator of card #5 will be set to 402.8 Hz.`

float WINAPI Program_Multiplier(int Card, int Channel, float Multiplier, int Direction, int SpeedMode)

DESCRIPTION

This function programs the Binary Rate Multiplier. In order to program the Rotational Frequency for the Coarse channel, the Speed Ratio has to be first set.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

Multiplier = number which is a float.

Direction = an integer that is either a (1) for Up or a (0) for Down.

Speedmode = an integer, (0) for Single Speed mode, and (1) for Two Speed Mode.

RETURNED DATA

This routine returns the actual value of Revs as a float. Since there is some rounding involved the function returns the actual value that will be programmed.

Range:
Single speed / 12 bit mode = .03 to 2014
Single speed / 16 bit mode = .03 to 125
Two speed / 12 and 16 bit mode = .03 to 254

EXAMPLE CODE

Code	Result/Returned Data

`float Multiplier;`
 	
`Multiplier = Program_Multiplier(0,5,5.5,1,0);`
	Programs a positive Rotational speed of 5.501322 rps (closest value to 5.5) for card #0, channel 5. Returns 5.501322 as the value for variable “Multiplier”.

void WINAPI Program_Ratio(int Card, int Channel, WORD Ratio)

DESCRIPTION

Programs the Two Speed Ratio between Coarse and Fine angles. A ratio of 1:2 would be written as 2. A Ratio of 1:32 would be 32.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

Ratio = a WORD that represents the ratio between Coarse and Fine. (range = 2 to 128)

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Program_Ratio(0,0,32);`	`Writes a Two Speed Ratio of 1:32 between channel 0 (Coarse) and channel 1 (fine) on card #0.`

void WINAPI Program_Resolution(int Card, int Channel, int Resolution)

DESCRIPTION

Programs the resolution of a channel to 12 or 16 bits. Error checking is performed, the appropriate bits are masked, and the new value is written to the card.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

Resolution:
12 = 12 bit resolution
16 = 16 bit resolution

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Program_Resolution(0,0,16);`	`Sets the resolution of card #0, channel 0, to 16 bits.`

int WINAPI Read_Control(int Card, int Channel)

DESCRIPTION

This routine is a specialized version of the Read Register function. It reads the Control Register specifically. This way the user does not need to know the offsets for each channel.

INPUT DATA

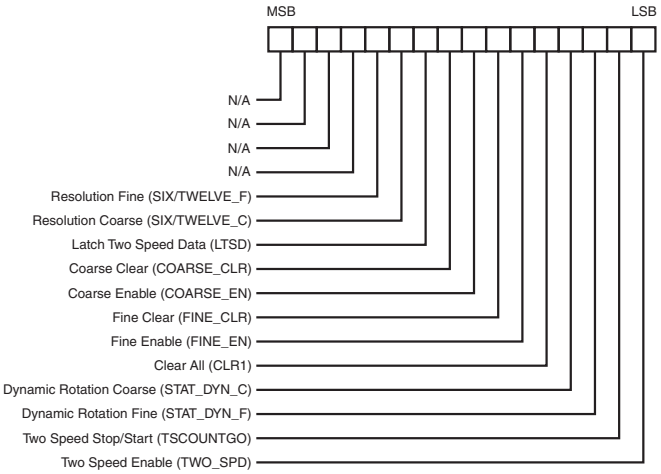
Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

RETURNED DATA

This routine returns the value in the Control Register.

The control register contains data for the following parameters:



Parameter	Bit = 0	Bit = 1
SIX/TWELVE_F	16 Bit Mode	12 Bit Mode
SIX/TWELVE_C	16 Bit Mode	12 Bit Mode
LTSD	Unlatch Two Speed data	Latch Two Speed data
COARSE_CLR	Resume Coarse	Clear Coarse
COARSE_EN	Stop Coarse	Start Coarse
FINE_CLR	Resume Fine	Clear Fine
FINE_EN	Stop Fine	Start Fine
CLR1	Resumes normal operation	Clears all channels
STAT_DYN_C	Coarse Channel in Dynamic Mode	Coarse Channel in Static Mode
STAT_DYN_F	Fine Channel in Dynamic Mode	Fine Channel in Static Mode
TSCOUNTGO	Rotating in Two Speed Mode	Not Rotating in Two Speed Mode
TWO_SPD	In One Speed Mode	In Two Speed Mode

EXAMPLE CODE

Code	Result/Returned Data

`int Control;`
 	
`Control = Read_Control(1,1);`
	`Reads the Control Register for card #1, channel 1.`

int WINAPI Read_Latched_Angle(int Card, int Channel, int* Coarse, int* Fine)

DESCRIPTION

After the Two Speed Latch bit is set, the latched data can be read to verify operation of the card. A bit will be set when the data has been latched. By reading this bit, the routine will know when to read the angles. The two angles are returned through the Coarse and Fine parameters. When Latched = 0, the latched angles can be read.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 2 for the channel pairs of interest.

0 = Channel 0 (Coarse) and Channel 1 (Fine)
1 = Channel 2 (Coarse) and Channel 3 (Fine)
2 = Channel 4 (Coarse) and Channel 5 (Fine)

Coarse = Current coarse angle data (should be zero), set as a variable.

Fine = Current fine angle data (should be zero), set as a variable.

RETURNED DATA

Returns the Latched status of the selected channel pairs. 0 = channels latched (data valid), 1 = channels unlatched (data invalid).

EXAMPLE CODE

Code	Result/Returned Data

`int Angle;`
`int* Coarse;`
`int* Fine;`
`Angle = Read_Latched_Angle(1,1,Coarse,Fine);`
	Returns the latched status of Two Speed channels 2 (Coarse) and 3 (Fine). Fills variable Coarse with the current angle data of channel 2. Fills variable Fine with the current angle data of channel 3.

int WINAPI Read_Register(int Card, int Register)

DESCRIPTION

This is a generic function to read any register on the Card. It can be used to read angle data, or to check the Control Register, etc. The possible offsets are listed in the synchro.h file.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Register = an integer.

RETURNED DATA

This routine returns an integer that stores the value of the register that was read from the card.

EXAMPLE CODE

Code	Result/Returned Data

`int Register;`
 	
`Register = Read_Register(0,SB3622x_FBRM_CH1);`
	Reads the Fine Binary Rate Multiplier register (SB3622x_FBRM_CH1) for card #0.

Note: *A list of available registers can be found in the Synchro.h file.*

int WINAPI Read_Resolution(int Card, int Channel)

DESCRIPTION

Reads the given channel, and returns the current resolution.
If bits are: 16 Resolution is 16
If bits are: 12 Resolution is 12

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

RETURNED DATA

This routine returns an integer that represents the current resolution of a channel as either 12 or 16 bits. If a 9 is returned than there was an error with the initialization of the card.

EXAMPLE CODE

Code	Result/Returned Data

`int Resolution;`
 	
`Resolution = Read_Resolution(4,3);`
	`Returns the current resolution (as an integer) for card #4, channel 3.`

void WINAPI Start_Rotation(int Card, int Channel, int Start)

DESCRIPTION

This function Starts (Mode = 1) or Stops (Mode = 0) the Dynamic Rotation on a given channel.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 5 for the channel of interest.

Start = an integer that is either a (1) to Start, or a (0) to Stop dynamic rotation.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Start_Rotation(4,0,1);`	`Starts Dynamic Rotation mode on card #4, channel 0.`

void WINAPI Start_TwoSpeed(int Card, int Channel, int Start)

DESCRIPTION

This routine is used to Start (1) or Stop (0) the Two Speed rotation.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channel = an integer between 0 and 2 for the channel pairs of interest.

0 = Channel 0 (Coarse) and Channel 1 (Fine)
1 = Channel 2 (Coarse) and Channel 3 (Fine)
2 = Channel 4 (Coarse) and Channel 5 (Fine)

Start = an integer that is either a (1) to Start, or a (0) to Stop dynamic rotation.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Start_TwoSpeed(1,0,1);`	Starts Two Speed rotation for card #1, channel 0 (Coarse) and channel 1 (Fine).

void WINAPI Static_TwoSpeed(int Card, int Channels, double CoarseAngle)

DESCRIPTION

This routine writes a user-defined Coarse angle to the Coarse channel, and uses the programmed ratio to calculate and write the Fine angle to the Fine channel. The routine will also check to make sure the card is not in Dynamic Mode when writing to the card.

INPUT DATA

Card = an integer between 0 and 7 for the card of interest. The card number is set in the MOTION FEEDBACK BOARD MANAGER.

Channels = an integer 0, 2 or 4.

0 – Channel 0 and Channel 1
2 – Channel 2 and Channel 3
4 – Channel 4 and Channel 5

Coarseangle = a double that represents the new Coarse angle to be written to the card.

RETURNED DATA

This routine does not return any data.

EXAMPLE CODE

Code	Result/Returned Data
`Static_TwoSpeed(0,0,97.2);`	Writes the angle 97.2 to the coarse channel of channel pair #0. Determines and writes the ideal corresponding fine angle to the fine channel of channel pair #0.

Setup for Rotation

Setup Example

To enable rotation of the card you will need to call the following functions:

- Get_Card_Type (int Card)
- Initialize (int Card)
- Program_Amplitude (int Card, int Amplitude)
- Program_Frequency (int Card, int Frequency)
- Program_Multiplier (int Card, int Channel, float Multiplier, int Direction, int SpeedMode) (Note 1)
- Start_Rotation (int Card, int Channel, int Start) (Note 1)
- Close (int Card)

Sample pseudo code:

```
Int main ()
{
      char* CardType = Get_Card_Type (Card) ;
      ***
      Initialize (Card) ;
      ***
      Program_Amplitude (Card, Amplitude) ;
      ***
      Program_Frequency (Card, Frequency) ;
      ***
      float WrittenRate = Program_Multiplier (Card, Channel, Multiplier, Direction,
      SpeedMode) ;
      ***
      Start_Rotation (Card, Channel, Start) ;
      ***
      Close (Card) ;
}
```

Note 1: For Program_Multiplier (…) and Start_Rotation (…): The value for channel is zero based. This means If you want to use channel 1, you would specify channel 0.

APPENDIX A

Signal Input / Output Configuration

Input and output options are created by factory installation of jumpers (TB1 to TB6) on each of the available channels. **Table 1** lists the Jumper Block designations for each channel. **Table** 2 lists the "D" connector reference input pins for each channel. Table 3 lists the jumper installation for each output option. **Figure 31** shows the jumper location and configuration for jumper blocks TB1 to TB6.

Pins 1 through 6 of each TB jumper block (TB1-TB6) determine the card's output configuration. Pins 7 through 10 determine the card's input configuration.

Table 1. Jumper Designations

Jumper Block	Channel
TB1	1
TB2	2
TB3	3
TB4	4
TB5	5
TB6	6

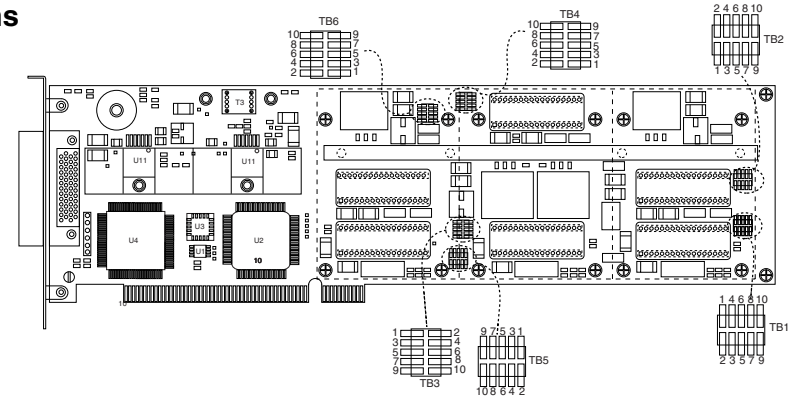


Figure 31. SB-3622X Jumper Location

Note: *The output signal configuration of the hybrids will match the output type option selected for types 5 through 7, (see ordering information), which require a daughter board.*

Card reference inputs accept either 4.4, 26, or 115 Vrms voltages only.

Table 2. "D" Connector Channel Reference Assignments

Channel	Reference High		Reference Low	
	Pin #	Function	Pin #	Function
1	17	RH_EXT_CH1	19	RL_EXT_CH1
2	24	RH_EXT_CH2	7	RL_EXT_CH2
3	6	RH_EXT_CH3	22	RL_EXT_CH3
4	13	RH_EXT_CH4	9	RL_EXT_CH4
5	8	RH_EXT_CH5	15	RL_EXT_CH5
6	14	RH_EXT_CH6	5	RL_EXT_CH6

OUTPUT CONFIGURATION

TBx Jumpers are configured as per **Table 3** to select an appropriate output option.

Table 3. Output TBx Jumper Configuration

(Applies to TB1-6) Jumpers Installed				
Output Option #	**Type**	**Hybrid Type**		
		3	**5**	**8**
1	11.8v Synchro	3-4, 5-6, 7-8, 9-10	3-4, 5-6, 7-8, 9-10	N/A
2	11.8v Resolver	N/A	1-2, 5-6, 7-8, 9-10	7-8, 9-10
3	6.8v Resolver	N/A	1-2, 7-8, 9-10	N/A
4	2v Resolver	N/A	N/A	N/A
5 (Note 3)	11.8v Synchro	N/A	3-4, 5-6, 7-8, 9-10	N/A
6 (Note 3)	11.8v Resolver	N/A	1-2, 5-6, 7-8, 9-10	N/A
7 (Note 3)	90v Synchro	N/A	3-4, 5-6, 7-8, 9-10	N/A
8 (Note 3)	90v Synchro	N/A	3-4, 5-6, 7-8, 9-10	N/A

Notes:
1. N/A = No jumpers installed.
2. 90v output requires a transformer.
3. Transformer coupled output configurations cannot be field reconfigured. (Output options 5, 6, 7, 8).
4. For reference type 0 & 6 when applying a 115v ref input, remove jumpers 7-8 and 9-10.

CUSTOM DIGITAL-TO-SYNCHRO/RESOLVER OUTPUT VOLTAGES

For custom digital-to-synchro/resolver output voltages, the reference input voltage must be set as follows:

$$\text{Reference Input Voltage} = \frac{\text{Desired Output Voltage}}{X}$$

where X is a custom ratio multiplier – refer to Table 4.

Table 4. Reference Multiplier (for non-standard D-S/R output voltages)

OUTPUT OPTION	OUTPUT TYPE	NOMINAL REFERENCE INPUT (VRMS)	MULTIPLIER X
1	11.8V L-L Synchro	26	0.45
2	11.8V L-L Resolver	26	0.45
3	6.8V L-L Resolver (single-ended)	26	0.26
4	2V Resolver (single-ended)	4.4	0.455
5	11.8V L-L Synchro / 400 Hz, Transformer coupled	26	0.45
6	11.8V L-L Resolver / 400 Hz, Transformer coupled	26	0.45
7	90V L-L Synchro / 400 Hz, Transformer coupled	115	0.78

For example:

Desired output = 9V L-L

Multiplier X = 0.45 (for option #1)

$$\text{Reference Input Voltage} = \frac{9\ V_{LL}}{0.45}$$

$$\text{Reference Input Voltage} = 20V$$

APPENDIX B

Pinouts

CARD PINOUTS

This section describes the pinouts for the card. The card has one connector, a 50 pin mini D connector. The pinouts for this mating connector are shown in **Figure 32** and **Table 5** on the following pages.

Note: *All ground returns are connected to the chassis ground.*

Table 5. SB-3622X Pinouts of P1

Pin	Function	Pin	Function
1	-5V EXT * TP OUTPUT	26	S1_CH 6 OUTPUT
2	+15V EXT * TP OUTPUT	27	S3_CH 5 OUTPUT
3	GND	28	S1_CH 5 OUTPUT
4	RL_INT OUTPUT	29	S3_CH 6 OUTPUT
5	RL_EXT_CH 6 INPUT	30	S2_CH 6 OUTPUT
6	RH_EXT_CH 3 INPUT	31	S4_CH 6 OUTPUT
7	RL_EXT_CH 2 INPUT	32	S2_CH 5 OUTPUT
8	RH_EXT_CH 5 INPUT	33	S4_CH 5 OUTPUT
9	RL_EXT_CH 4 INPUT	34	S3_CH 4 OUTPUT
10	GND 1	35	S1_CH 4 OUTPUT
11	+5 EXT * TP OUTPUT	36	S3_CH 3 OUTPUT
12	-15 EXT * TP OUTPUT	37	S4_CH 4 OUTPUT
13	RH_EXT_CH 4 INPUT	38	RH_115_INT OUTPUT
14	RH_EXT_CH 6 INPUT	39	S1_CH 3 OUTPUT
15	RL_EXT_CH 5 INPUT	40	S4_CH 3 OUTPUT
16	GND 5	41	S2_CH 4 OUTPUT
17	RH_EXT_CH 1 INPUT	42	S2_CH 3 OUTPUT
18	GND 2	43	S1_CH 2 OUTPUT
19	RL_EXT_CH 1 INPUT	44	S1_CH 1 OUTPUT
20	GND 6	45	S3_CH 2 OUTPUT
21	GND 4	46	S2_CH 2 OUTPUT
22	RL_EXT_CH 3 INPUT	47	S4_CH 2 OUTPUT
23	GND 3	48	S2_CH 1 OUTPUT
24	RH_EXT_CH 2 INPUT	49	S3_CH 1 OUTPUT
25	RH_26_INT OUTPUT	50	S4_CH 1 OUTPUT

***Note:** No connection required, these are test points for the internal power supplies.*

SIGNAL CONNECTIONS

- Synchro Mode Connect S1, S2, S3
  - S1 = X
  - S2 = Z
  - S3 = Y
- Resolver Mode Connect
  - S3 = +SIN
  - S1 = -SIN
  - S2 = +COS
  - S4 = -COS
- Single Ended Mode Connections
  - When using 2v single ended configurations, S1 and S4 on card connector are no connect. Use associated analog ground per output channel for S1 and S4 resolver outputs used.

MATING CONNECTOR

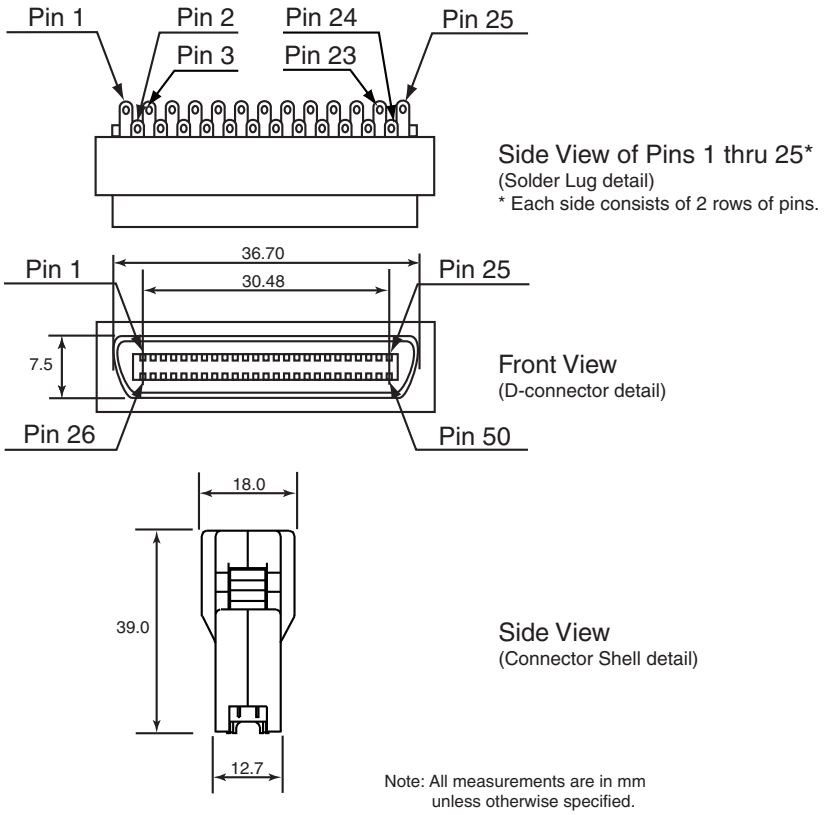


Figure 32. Mating Connector

PN: *50 pin connector solder plug (3M 10150-3000VE)*
50 pin connector Junction shell (3M 10350-52FO-008)

APPENDIX C

Assembly Drawings

MECHANICAL OUTLINE

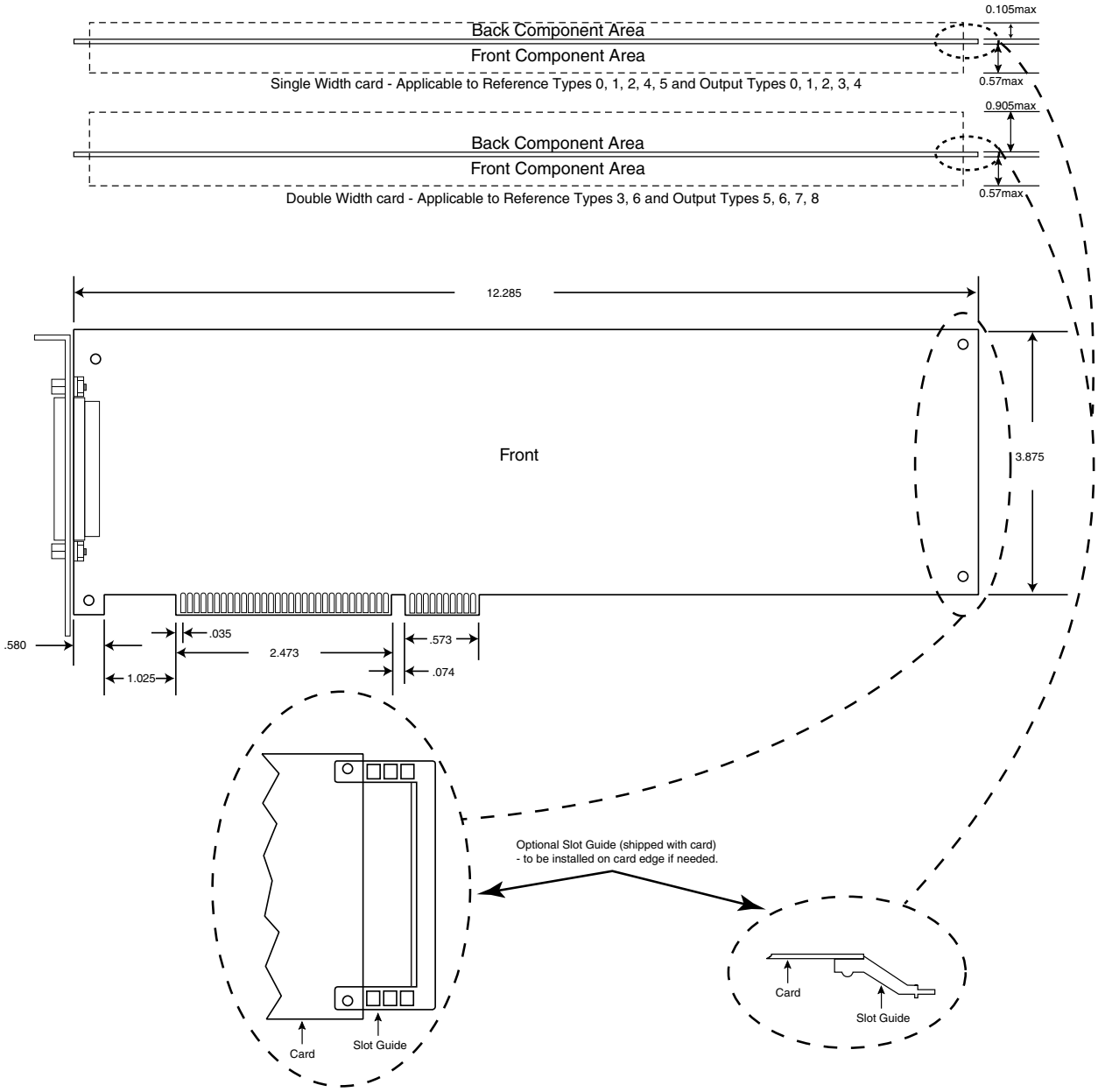


Figure 33. Mechanical Outline

Note: *Oscillator option IB requires two slots.*

ASSEMBLY DRAWING

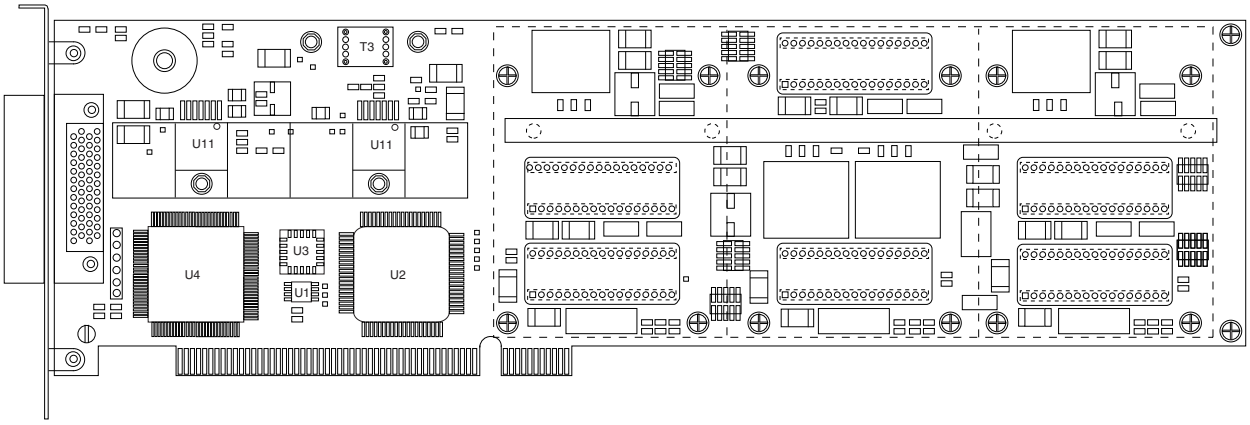


Figure 34. SB-3622X Card Assembly

CARD PHOTOS

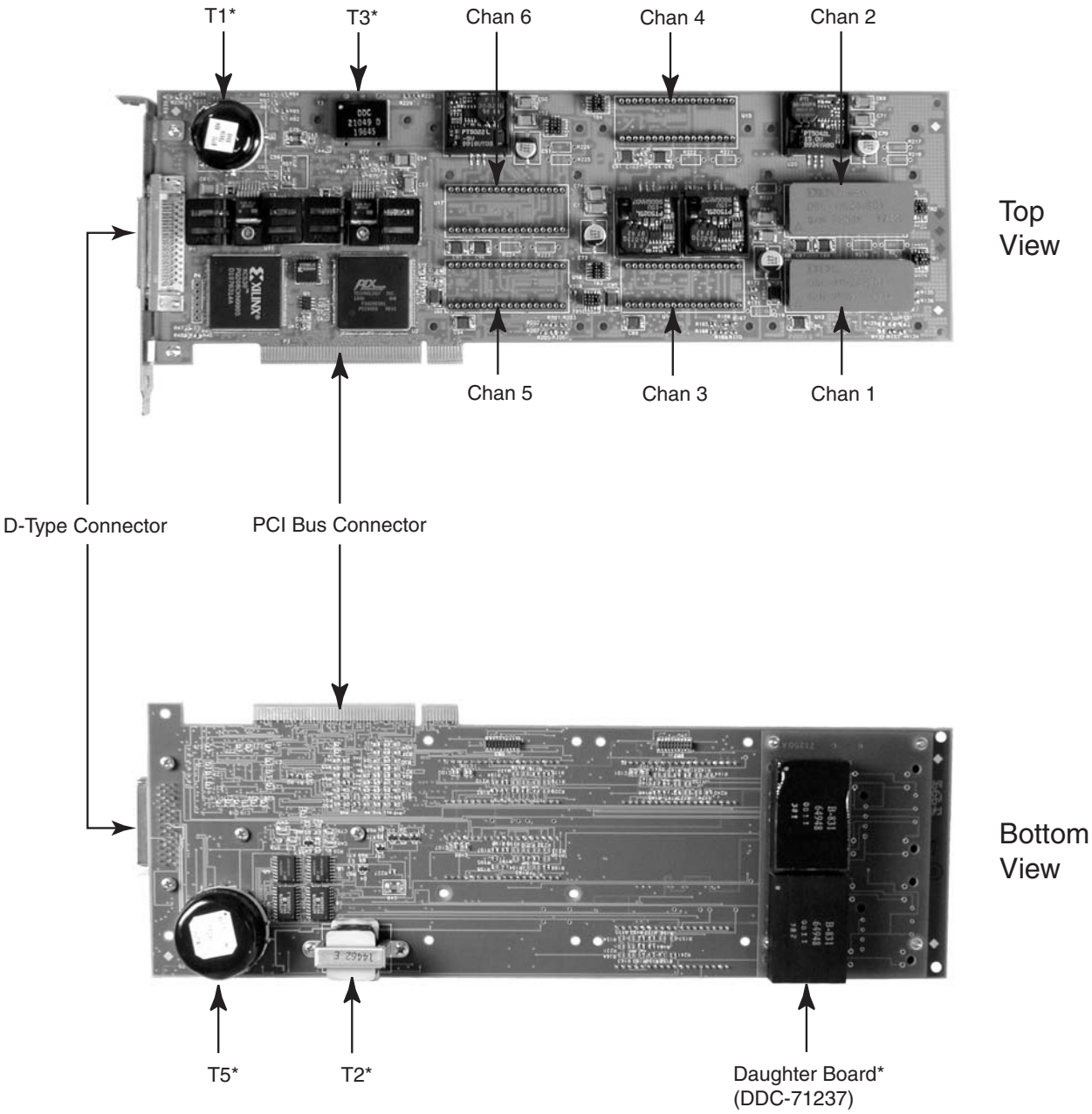


Figure 35. SB-3622X Card Photos

(2 Channel version shown)

***Note:** T1, T2, T3, T5 and Daughter Board options shown installed. Actual configuration depends upon card Reference and Output type.*

APPENDIX D

Reference and Output Options

Table 6. Reference Type

Reference Option	Reference Type	T1	T2	T3	T5
0	External, Solid State Isolated Input				
1	External, 26V/400Hz Transformer Isolated Input			X	
2	External, 115V/400Hz Transformer Isolated Input			X	
3	External, 115V/60Hz Transformer Isolated Input		X		
4	Internal, 3.4V High Current (300ma, Differential) Solid State Oscillator Output				
5	Internal, 26V High Current (1.5VA) Transformer Isolated Oscillator Output	X			
6	Internal, 115V High Current (1.5VA) Transformer Isolated Oscillator Output				X

Table 7. Output Type

Output Option	Output Type	Reference Input Required (Vrms)	Daughter Board (Installed)
1	11.8V L-L Synchro	26	
2	11.8V L-L Resolver	26	
3	6.8V Resolver	26	
4	2V Resolver	3.4	
5	11.8V L-L Synchro/400Hz, Transformer Coupled	26	X
6	11.8V L-L Resolver/400Hz, Transformer Coupled	26	X
7	90V L-L Synchro/400Hz, Transformer Coupled	115	X

GLOSSARY

Accuracy**.....***A measurement deviation from a calibrated standard.*

AGND**.....***Analog Ground.*

BW (Bandwidth) **.....***The frequency band in which a system can process information. Also known as the 3dB point.*

BIT (Built-in test) **.....***Digital output that will indicate if a fault condition exists.*

Carrier Frequency **.....***Reference or excitation frequency required by the synchro or resolver device and the converter.*

Coarse Angle**.....***In a two-speed system, the reading from the synchro or resolver with a 1:1 ratio.*

Differential**.....***Input signal that is with respect to another signal (S3-S1), but not to ground.*

Direct Input (sin/cos)**.....***Input signal that is with respect to ground.*

Fine Angle**.....***In a two-speed system, the reading from the resolver with a 1:N ratio, where N >1.*

GND **.....***Digital Ground.*

J-lead**.....***The lead of a device that is J- or hooked-shaped.*

Line-to-Line Voltage**.....***The measurement (in Vrms) between two signals.*

Measured Angle **.....***The analog synchro or resolver signals representing a position that is converted to a digital word or angle.*

Reference **.....***Excitation for synchro or resolver device and converter.*

Resolution **.....***Number of bits available for measurement or simulation.*

Resolver**.....***Type of transducer whose four-wire output has a direct relationship to the angular position of the shaft. A resolver contains two output windings 90° out of phase.*

RPS(Revolutions per second) **.....***A unit of measure for velocity or tracking rate.*

Simulated Angle**.....***The digital word or angle to be converted into analog signals representing a synchro or resolver positioned at that angle.*

Speed Ratio **.....***The ratio of coarse to fine in a two-speed system.*

Synchro **.....***Type of transducer with a three-wire output having a direct relationship to the angular position of the shaft. A synchro contains three output windings 120° out of phase.*

Tracking rate**.....***A velocity or speed at which the converter can monitor the changes of the signal output.*

Type II Tracking**.....***A type of servo control loop.*

Velocity**.....***The rate of a change in position for a certain time interval.*

Vrms**.....***Root-mean-square, or effective voltage (Vp x* $1/\sqrt{2}$ *).*

INDEX