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 *Abaco Systems / SBS ABI-PMC2-2* at our website: *Click HERE*

Getting Started with the ABI/ASF/ESF-PMC2 Series









1: Introduction 1
2: Before You Begin 7
3: Installing the Software 9
4: Installing the Hardware 13
5: Performing System-Specific Installation 23
6: Running Unit Test 33
7: Connecting to the 1553 Bus 49
8: What to Do Next 51
9: Hardware Specifications 53
10: Card Information 69
11: Using the INTEGRITY API 79
A: Revisions 83

SBS Technologies®
7401 Snaproll NE
Albuquerque, NM 87109
Sales: 800-SBS-1553 or 505-875-0600
Fax: 505-875-0400
Tech Support: 877-832-4727
Tech Support Email: support.government@sbs.com
Documentation Support Email: govdocs@sbs.com
http://www.sbs.com
http://www.resource.sbs.com

Applies to models:
ABI-PMC2, ASF-PMC2, ABI-PMC2-XT,
ASF-PMC2-XT, ABI-PMC2-CC, ASF-PMC2-CC,
ESF-PMC2-CC, ASF-PMC2CC-D

Doc PN: 504-553099-00

SBS Technologies

© 2005 SBS Technologies, Incorporated. All rights reserved.

Getting Started with the ABI/ASF/ESF-PMC2 Series

This document is the intellectual property of SBS Technologies, Inc. (SBS), and contains proprietary and confidential information. Use, disclosure, and reproduction is permitted only under the terms of an SBS software license agreement or explicit written permission of SBS. You are not authorized to use this document or its contents until you have read and agreed to the applicable software license agreement.

THIS DOCUMENT AND ITS CONTENTS ARE PROVIDED AS IS, WITH NO WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM ANY COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

All computer code and software contained in this document is licensed to be used only in connection with an SBS product. Even if this code or software is merged with any other code or software program, it remains subject to the terms and conditions of this license. If you copy, or merge, this code or software, you must reproduce and include all SBS copyright notices and any other proprietary rights notices.

In no event will SBS be liable for any lost revenue or profits or other special, indirect, incidental and consequential damage, even if SBS has been advised of the possibility of such damages, as a result of the usage of this document and the software that this document describes. The entire liability of SBS shall be limited to the amount paid by you for this document and its contents.

SBS shall have no liability with respect to the infringement of copyrights, trade secrets, or any patents by this document of any part thereof. Please see the applicable software license agreement for full disclaimer or warranties and limitations of liability.

RESTRICTED RIGHTS LEGEND

Use, duplication, reproduction, release, performance, display or disclosure by the Government is subject to restrictions set forth in subparagraph (b)(3) of the Rights in Technical Data and Computer Software clause at 48 CFR 252.227-7013.

SBS Technologies, Inc., 7401 Snaproll NE, Albuquerque, NM 87109

This manual, as well as the software described herein, is furnished under license and may only be used or copied in accordance with the terms of such license.

SBS Technologies, Inc., and its logo are trademarks of SBS Technologies, Inc. All other brand names and product names contained herein are trademarks, registered trademarks, or trade names of their respective holders.

1: Introduction

This chapter introduces the *Getting Started with the ABI/ASF/ESF-PMC2 Series* manual. It describes the contents of this manual, and the terminology and conventions used in this manual. The sections are as follows:

- Contents of Getting Started with the ABI/ASF/ESF-PMC2
- Contents of MIL-STD-1553 ABI/ASF User's Manual
- Contents of the Integrated Avionics Library Reference Manual
- Acronyms
- Conventions



Cross Reference: Appendix A gives a brief summary of technical revisions made to this manual.

1.1 Contents of *Getting Started with the ABI/ASF/ESF-PMC2*

This manual applies to the following model numbers (XT indicates extended temperature, CC indicates conduction cooled, and the last digit in the model number indicates the number of channels):

- ABI-PMC2-1
- ABI-PMC2-XT-1
- ABI-PMC2CC-1
- ASF-PMC2-1
- ASF-PMC2-XT-1
- ASF-PMC2CC-1
- ASF-PMC2CC-1-D
- ABI-PMC2-2
- ABI-PMC2-XT-2
- ABI-PMC2CC-2
- ASF-PMC2-2

- ASF-PMC2-XT-2
- ASF-PMC2CC-2
- ASF-PMC2CC-2-D
- ESF-PMC2CC-1
- ESF-PMC2CC-2

The intention of this manual is to assist you in getting the ABI-PMC2, ASF-PMC2, or ESF-PMC2 up and running as quickly as possible. It addresses the following:

- Installing the hardware
- Installing the software
- Configuring the software for your operating system
- Testing
- Basic module operation

This manual assumes that you will be using the SBS Integrated Avionics Library to operate the module. See Section 10.4 for instructions on starting up the module without using the library.

1.2 Contents of *MIL-STD-1553 ABI/ASF User's Manual*

The accompanying *MIL-STD-1553 ABI/ASF User's Manual* contains complete details on module programming and operation, including the following:

- MIL-STD-1553 programming and structures (Chapters 4–15)
- Sample 1553 applications

1.3 Contents of the *Integrated Avionics Library Reference Manual*

The *Integrated Avionics Library Reference Manual* provides information on using the included avionics libraries in your own application.

1.4 Acronyms

Table 1.4.1 defines some of the basic terms used throughout this manual.

Table 1.4.1: Basic Terminology

Term	Meaning
API	Application Programing Interface
BC	Bus controller
BM	Bus monitor
Bus	A single 1553 bus connection (i.e., Bus A or Bus B)
Channel	One complete, dual-redundant 1553 bus interface
Device	A logical entity that corresponds one-for-one with a 1553 channel and a device entry in the *sbs_dev.cfg* configuration file
Dual-redundant	Includes both a primary and a secondary connection (i.e., Bus A and Bus B make up a dual-redundant bus)
Firmware	Program running in the ABI/ASF/ESF digital signal processors that controls all 1553 operations. The firmware must be loaded upon device initialization.
RT	Remote terminal
SA	Subaddress
Word	A 16-bit value; i.e., two bytes

1.5 Conventions

The following conventions appear in this document. These conventions may differ from those used in other SBS publications. The subsections listed below describe each convention in more detail:

- Typographic Conventions
- Words Having Special Meaning
- Compound Keystrokes and Menu Selections
- Symbols

1.5.1 Typographic Conventions

Table 1.5.1 shows the typographic conventions used in this document.

Table 1.5.1: Typographic Conventions

Element	Use in Body Text	Use in Procedures
Italic	➢ Cross references to other SBS publications ➢ Filenames and directory paths ➢ Emphasis	➢ Cross references to other SBS publications ➢ Filenames and directory paths
Bold	➢ (Not used in body text)	➢ Controls, dialogs, menus, and text or numeric fields that appear on the screen ➢ Keys on your keyboard
`Courier Roman`	➢ Code examples	➢ Simulating the appearance of screens
`Courier Bold`	➢ Library function calls and syntax ➢ Emphasizing lines of code	➢ Commands and other information that you type as given
Angle brackets, e.g.,< >	➢ Enclosing variable information that you type (without the brackets) in place of a dummy variable	➢ Enclosing variable information that you type (without the brackets) in place of a dummy variable

The point size of the text varies depending on whether it is used in body text, code examples, notes, screens, or procedures.

1.5.2 Words Having Special Meaning

In procedures, the words “Enter” (or “enter”) and “Type” (or “type”) have special meanings that are indicated in Table 1.5.2.

Table 1.5.2: Words with Special Meaning

Word	Meaning
Enter	Key in the specified text or variable information and press the **Return** key.
Type	Key in the specified text. Do not press **Return**.

1.5.3 Compound Keystrokes and Menu Selections

Compound Keystrokes

Whenever a procedure instructs you to press multiple keys, a double angle bracket "»" separates the names of the keys. Table 1.5.3 shows an example.

Menu Selections

Whenever a procedure instructs you to select an item from a pull-down menu, a double angle bracket "»"separates the menu items. Table 1.5.3 shows an example.

Table 1.5.3: Examples of Notation for Compound Keystrokes and Menu Selections

Instruction	Meaning
Press **Ctrl** » **Alt** » **Delete**.	Press the **Ctrl**, **Alt**, and **Delete** keys simultaneously.
Select **File** » **Open**.	Select **Open** from the **File** menu.

1.5.4 Symbols

The following symbols appear throughout this manual:



Warning: Paragraphs next to this symbol contain information critical to module operation or to your safety.



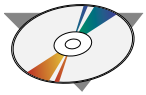
Note: Paragraphs next to this symbol contain information important to module operation.



Tip: Paragraphs next to this symbol contain useful tips.



Cross Reference: Paragraphs next to this symbol contain cross references to other parts of this manual, or to other SBS publications.



Software Cross Reference: Paragraphs next to this symbol contain cross references to software media included with this product.

2: Before You Begin

The sections in this chapter describe what to do after receiving and prior to installing your card. The sections are the following:

- What You Should Have Received
- Unpacking the Card
- What You Will Need

2.1 What You Should Have Received

- ABI-PMC2, ASF-PMC2, or ESF-PMC2 interface module
- Cable assembly
- SBS Resource CD - Contains PDF versions of this manual, the *MIL-STD-1553 ABI/ASF User's Manual,* and the *Integrated Avionics Library Reference Manual*

2.2 Unpacking the Card



Warning: This is an electronic product that is sensitive to electrostatic discharge. Take normal precautions in handling the card to prevent damage.

- Carefully unpack the card and inspect it for physical damage that might have occurred during shipping.
- If you have a damaged card, contact the SBS technical support group that handles maintenance, repairs, and warranties in Albuquerque. When you call us, give us the serial number of your card, and have the card available in case we have questions about its condition.

2.3 What You Will Need

The ABI/ASF/ESF-PMC2 product package includes all items required to operate the card on your chassis except for the following:

- Two MIL-STD-1553 bus terminators, to perform a bus test on the module
- Two single bus couplers or other appropriate transformer coupling devices, to connect to an actual 1553 bus



Cross Reference: See Subsection 9.2.5 for ordering information and part numbers for these items.

3: Installing the Software

The sections in this chapter include descriptions of the provided software disks and instructions on software installation. This chapter contains the following sections:

- Support Software
- Copying the Software to Your Host System

3.1 Support Software

SBS provides support software for its MIL-STD-1553 products as part of the Integrated Avionics Library on the following media:

- SBS Resource CD

SBS Resource CD

The SBS Resource CD contains the following:

- Integrated Avionics Library, including C library source files, DLLs, sample applications, and the console mode version of the Unit Test executable
- Device drivers necessary to support the interface between the libraries and your computer system
- SBS PASS demo software
- Product documentation in PDF format (requires Adobe Acrobat Reader)
- Firmware files that have to be downloaded to the PMC2 card upon initialization

3.2 Copying the Software to Your Host System

Use the following instructions to copy the software to the system in which you are installing the PMC2 card. The following two sections include separate instructions for PC systems and for non-PC systems:

- Windows Operating Systems
- UNIX-Based Operating Systems

3.2.1 Windows Operating Systems

Unless otherwise specified, the installation batch files will place the Integrated Avionics Library on your system. The *Integrated Avionics Library Reference Manual* provides information on using the library in your own application, and operating system dependent software in the *c:\sbs_ver<x.yz>* folder (directory) in your Windows 95/98/ME/2000 or Windows NT environment, where *<x.yz>* is the version number of the current release.

1. Install the library by completing the following steps:
   - Insert the SBS Resource CD into your CD-ROM drive.
   - Start the Explorer and navigate to the CD.
   - Double-click on the Library folder.
   - Double-click **Setup.exe** from the Explorer.
   - Follow the instructions that appear on the screen.

   If you select all of the defaults, the directory structure shown in Figure 3.2.1 is created.

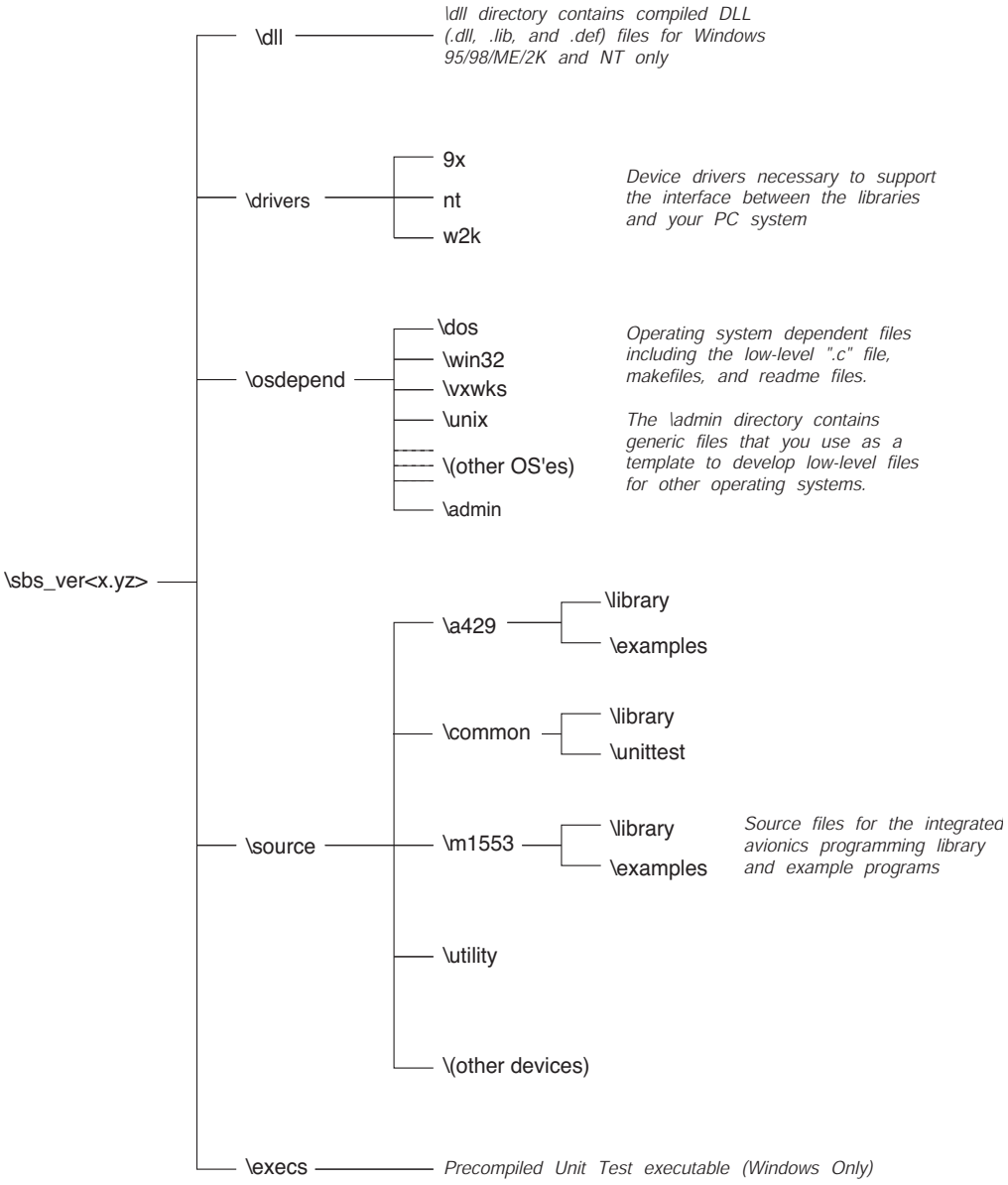


Figure 3.2.1: Default Directory Structure

2. Create the SBS user directory structure shown in Figure 3.2.2 by adding *working* and *firmware* directories under the *sbs_ver<x.yz>* directory, where *<x.yz>* is the version number of the current release.

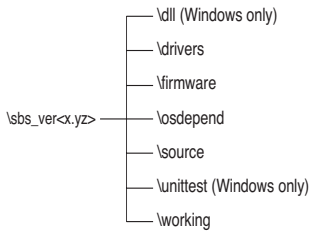


Figure 3.2.2: User Directory Structure

Note: The *\working* directory is a place for you to create and work with files without disturbing other files and directories.

3. Copy the firmware file(s) directly from the Firmware disk to the *\sbs_ver<x.yz>\firmware* directory on your system.

Tip: See the chapter *Performing System-Specific Installation* on page 23 to continue the software installation process.

3.2.2 UNIX-Based Operating Systems

To install the interface libraries and operating system software on your UNIX-based system, you must first copy the software to a PC system on your network, then transfer it via ftp to the UNIX-based system in which you are installing the PMC2 card. The steps below guide you through the process.

1. Copy the software to a PC system on your network by following procedure steps 1–3 beginning on page 10 in the preceding subsection, *Windows Operating Systems*. Make sure to select the files needed for your target UNIX system, not the PC system.

2. Enter **`ftp`** to access your host system.

3. Create the directory structure shown in Figure 3.2.3 using the ftp **`mkdir`** utility:

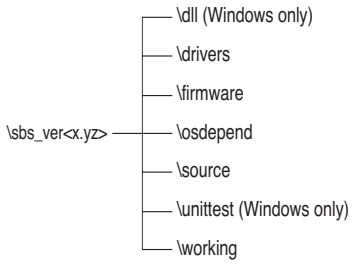


Figure 3.2.3: User Directory Structure

4. Use the ftp **`put`** or **`mput`** utility to transfer the interface library files from the PC environment to the host system (into the above tree structure).

4: Installing the Hardware

The sections in this chapter provide instructions on how to configure and install the ABI/ASF/ESF-PMC2 hardware. The sections are the following:

- Installing the Card
- Making Auxiliary Connections for External Signals
- Attaching the Cable Assembly

4.1 Installing the Card

To install the PMC2, you mount it on a host card. However, the installation procedure depends on the type of card you purchased. SBS manufactures three types; commercial, extended temperature (XT), and conductive-cooled (CC). Except for the components used on the cards themselves, the commercial and extended temperature cards have the same form factor and installation procedure.

The first subsection shows the board layout for each type of card. The remaining two subsections below describe how to install each type of card.

- Board Layouts
- Installing the Commercial/XT PMC2
- Installing the Conductive-Cooled PMC2
- Installing the Conductive-Cooled ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2



Warning: This is an electronic product that is sensitive to electrostatic discharge. Take normal precautions in handling the board to prevent damage.

4.1.1 Board Layouts

The illustrations in this subsection show the board layouts for all PMC2 models.

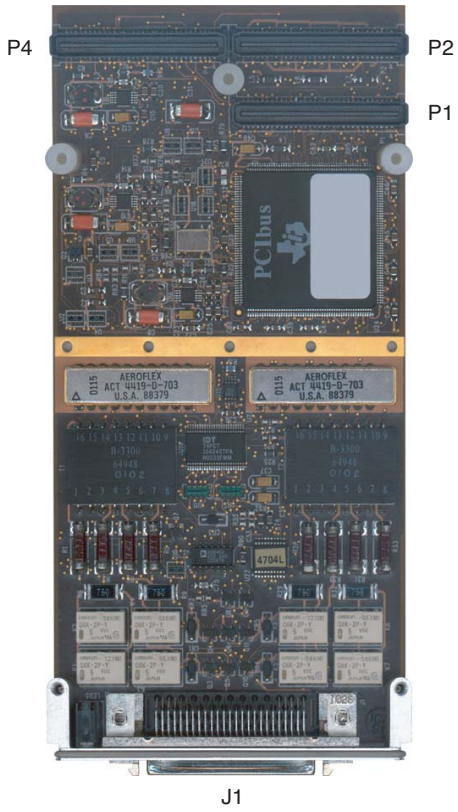


Figure 4.1.1: 1553 PMC2-Commercial and XT Board Layout

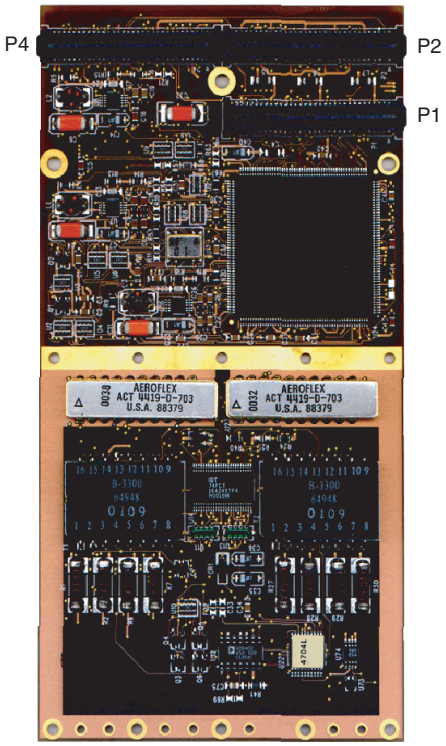


Figure 4.1.2: 1553 PMC2-Conductive Cooled Board Layout

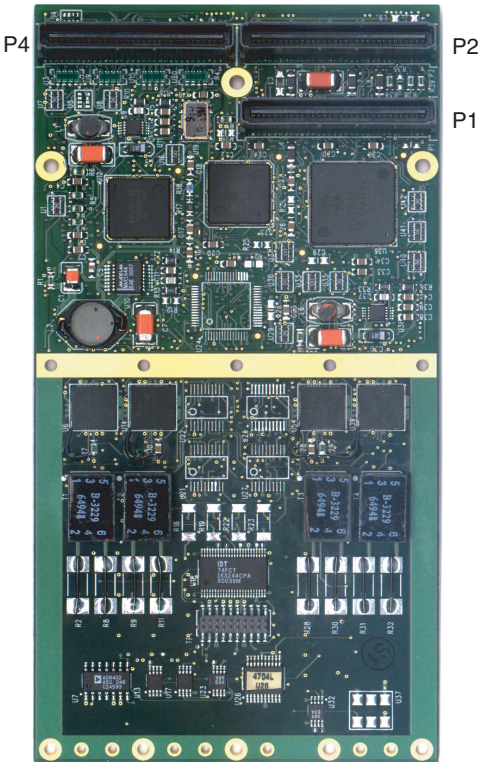


Figure 4.1.3: ASF-PMC2CC-1-D, ASF-PMC2CC-2-D and ESF-PMC2CC-1 Conductive Cooled Board Layout

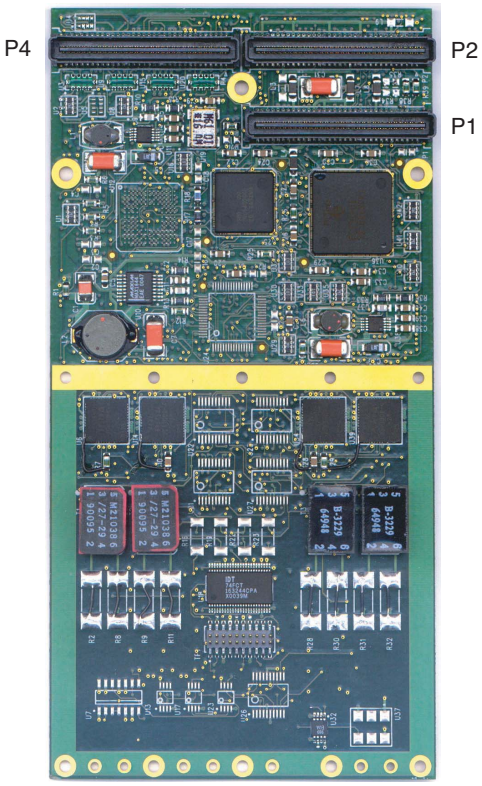


Figure 4.1.4: 1553 ESF-PMC2CC-2 Conductive Cooled Board Layout

4.1.2 Installing the Commercial/XT PMC2

Figure 4.1.5 depicts a typical commercial/XT PMC assembly mounting on a host card. Table 4.1.1 below describes the recommended and alternative methods for mounting the card.

Table 4.1.1: Mounting Methods for Commercial/XT PMC2

Screw Size	Recommended Mounting– Nylon clearance standoff into stainless hex nut	Alternate Mounting– Aluminum tapped standoff free running
2.5mm x 16mm Long	Apply Loctite 222MS, and torque to 30 in-oz.	Apply Loctite 222MS, and torque to 40 in-oz.
2.5mm x 5mm Long	N/A	Apply Loctite 222MS, and torque to 40 in-oz.

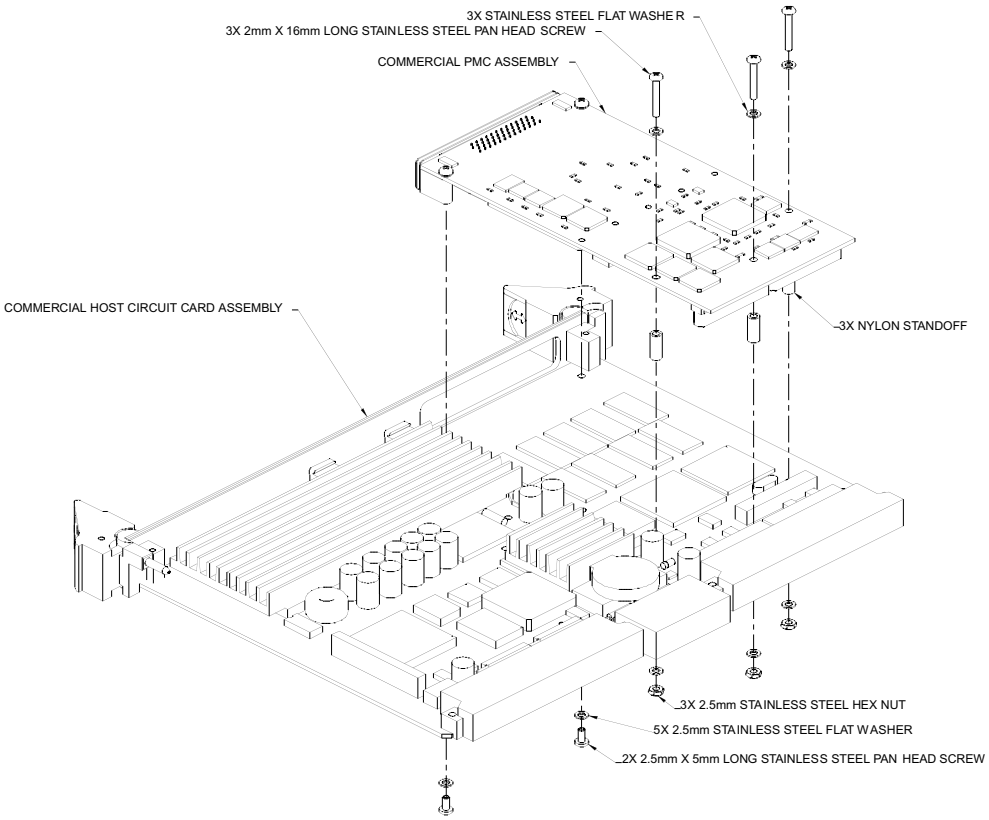


Figure 4.1.5: Typical Commercial/XT PMC2 Mounting to Host Card

Tip: Refer to Figure 4.1.1 for connector locations and to Figure 9.2.5 for connector pinouts.

Table 4.1.2 lists the fastener quantity, vendor part number, description, and vendor for the mounting of a commercial/XT PMC to a host card.

Table 4.1.2: Fasteners Needed for Mounting a Commercial/XT PMC2

Item	Qty	Mfg Part Number	Description	Vendor
1	3	1940-M250-M16-2A21	2.5 mm pan head screw x 16 mm long, stainless steel	Barnhill Bolt
2	2	1940-M250-M5-2A21	2.5 mm pan head screw x 5 mm long, stainless steel	Barnhill Bolt
3	8	1926-M250-A-2	m2.5 stainless steel flat washer	Barnhill Bolt
4	3	1918-M250-A-2	m2.5 stainless steel hex nut	Barnhill Bolt
5	3	M0509 25 N 0	4.5mm round spacer, nylon 10mm long	RAF

4.1.3 Installing the Conductive-Cooled PMC2

Figure 4.1.6 depicts a typical conductive-cooled PMC assembly mounting on a host card. Table 4.1.3 below describes the recommended and alternative methods for mounting the card.

Table 4.1.3: Mounting Methods for Conductive-Cooled PMC2

Screw Size	Recommended Mounting– Free running helical insert	Alternate Mounting 1– Free running tapped aluminum	Alternate Mounting 2– Locking helical insert
2mm	Apply Loctite 222MS, and torque to 30 in-oz.	Apply Loctite 222MS, and torque to 24 in-oz.	Torque additional 5 in-oz. after head is flush
2.5mm	Apply Loctite 222MS, and torque to 60 in-oz.	Apply Loctite 222MS, and torque to 45 in-oz.	Torque additional 8 in-oz. after head is flush

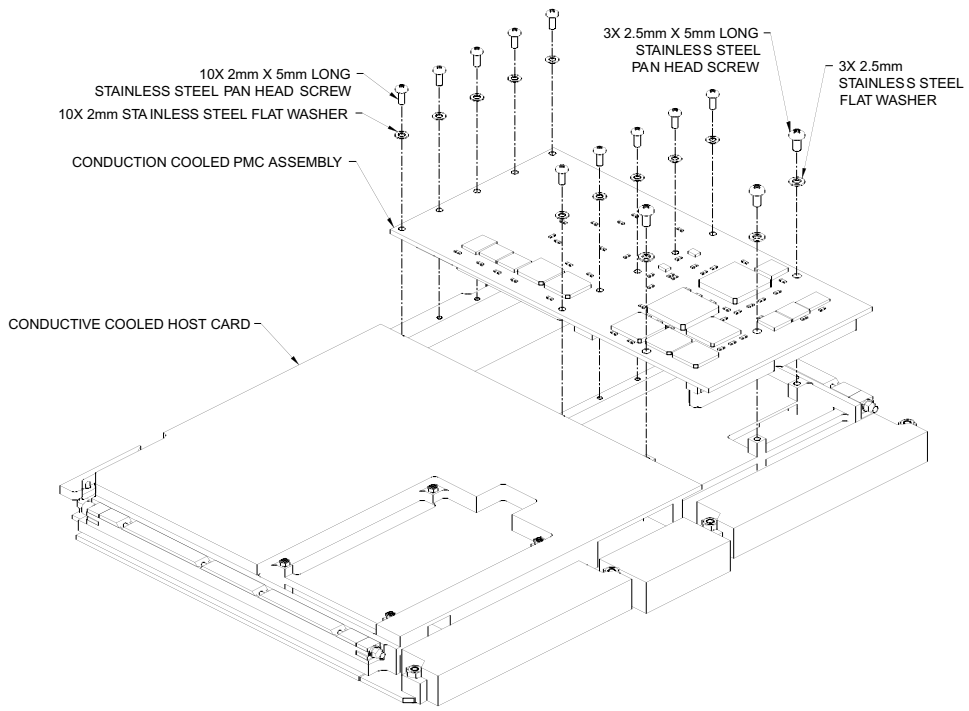


Figure 4.1.6: Typical Conductive-Cooled PMC2 Mounting to Host Card



Tip: Refer to Figure 4.1.2 for connector locations and to Figure 9.2.6 for connector pinouts.

Table 4.1.4 lists the fastener quantity, vendor part number, description, and vendor for the mounting of a conductive-cooled PMC to a host card.

Table 4.1.4: Fasteners Needed for Mounting a Conductive-Cooled PMC2

Item	Qty	Mfg Part Number	Description	Vendor
1	3	1940-M250-M5-2A21	2.5 mm pan head screw x 5 mm long, stainless steel	Barnhill Bolt
2	10	1940-M2-M5-2A21	2 mm pan head screw x 5 mm long, stainless steel	Barnhill Bolt
3	3	1926-M250-A-2	M2.5 stainless steel flat washer	Barnhill Bolt
4	10	1926-M2-A-2	M2 stainless steel flat washer	Barnhill Bolt

4.1.4 Installing the Conductive-Cooled ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2

Figure 4.1.7 depicts the conductive-cooled ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2 assembly mounting on a host card. Table 4.1.5 describes the recommended and alternative methods for mounting the card.

Table 4.1.5: Mounting Methods for Conductive-Cooled PMC2

Screw Size	Recommended Mounting– Free running helical insert	Alternate Mounting 1– Free running tapped aluminum	Alternate Mounting 2– Locking helical insert
2mm	Apply Loctite 222MS, and torque to 30 in-oz.	Apply Loctite 222MS, and torque to 24 in-oz.	Torque additional 5 in-oz. after head is flush
2.5mm	Apply Loctite 222MS, and torque to 60 in-oz.	Apply Loctite 222MS, and torque to 45 in-oz.	Torque additional 8 in-oz. after head is flush

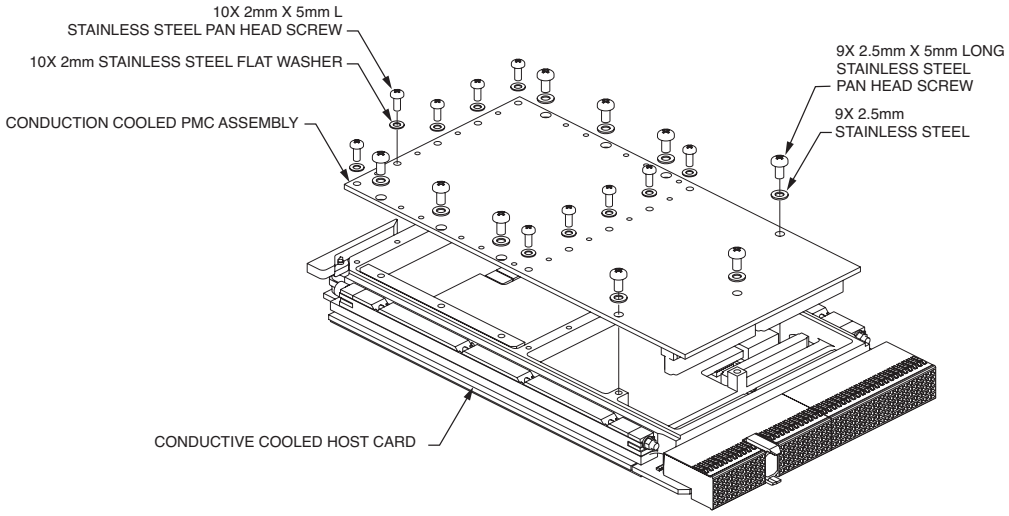


Figure 4.1.7: Conductive-Cooled ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2 Mounting to Host Card



Tip: Refer to Figure 4.1.2 for connector locations and to Figure 9.2.7 for connector pinouts.

Table 4.1.6 lists the fastener quantity, vendor part number, description, and vendor for the mounting of the conductive-cooled PMC2CC-D to a host card.

Table 4.1.6: Fasteners Needed for Mounting ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2

Item	Qty	Mfg Part Number	Description	Vendor
1	9	1940-M250-M5-2A21	2.5 mm pan head screw x 5 mm long, stainless steel	Barnhill Bolt
2	10	1940-M2-M5-2A21	2 mm pan head screw x 5 mm long, stainless steel	Barnhill Bolt
3	9	1926-M250-A-2	M2.5 stainless steel flat washer	Barnhill Bolt
4	10	1926-M2-A-2	M2 stainless steel flat washer	Barnhill Bolt

4.2 Making Auxiliary Connections for External Signals

If you are planning to use external signals, you need to make the proper connections to the J1 front panel connector, mezzanine P4 connector, or the J5 connector on the Auxiliary cable shown in Figure 4.2.1.See Table 9.2.3 on page 66 for a list of the pinouts on the cable. The remaining topics in this section describe making the connections for IRIG signals, external triggers, and external clocks.

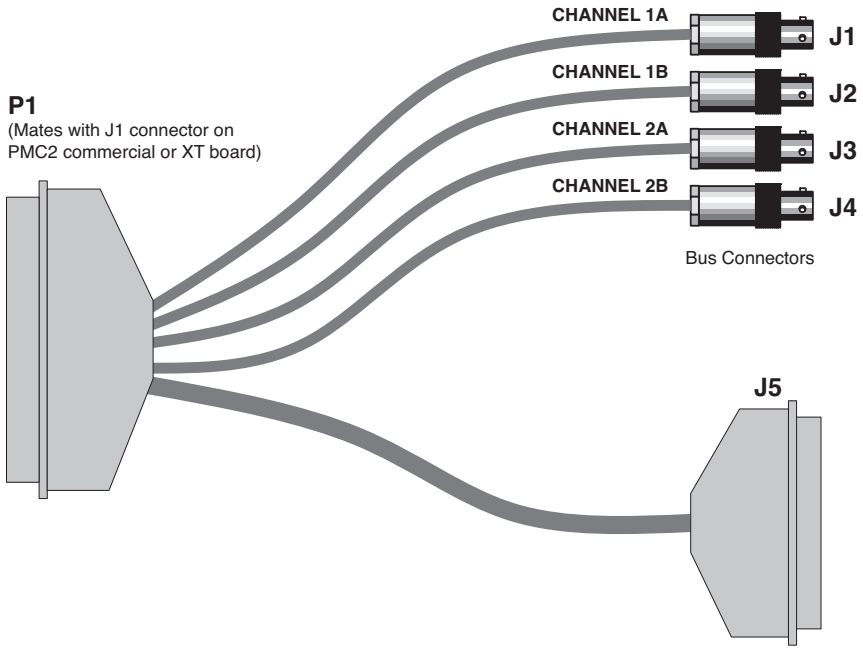


Figure 4.2.1: PMC2 Auxiliary Cable Assembly

IRIG Signal

If you are using an external IRIG signal as your timing source, follow the procedure below to make a connection:

1. Connect the IRIG signal to one of the pins given in Table 4.2.1.

Note: The IRIG input impedance is 10 kΩ.

Table 4.2.1: IRIG Pinouts

Mezzanine P4 Connector	Front Panel J1 Connector	J5 Connector on Auxiliary Cable
Pin 54	Pin 46	Pin 17

2. Connect the ground to any Ground pin on the 44-pin J5 connector.

Cross Reference: For more information on IRIG signals, see the Device Management Programing Chapter of the *ABI/ASF User's Manual*.

External Triggers

The PMC2 has eight external triggers. To connect an external trigger to the PMC2 card, follow the procedure below.



Cross Reference: You can also use External Trigger 0 to input an External Clock signal, see the following External Clock topic for details.



Note: All External Triggers are shared by both devices.

1. Connect the trigger signal to the appropriate pin shown in Table 4.2.2.

Table 4.2.2: External Trigger Pinouts

Trigger	Mezzanine P4 Connector	Front Panel J1 Connector	J5 Connector on Auxiliary Cable
0	58(–), 60(+)‡	43(–), 44(+)‡	43(+), 44(–)‡
1	46(–), 48(+)	41(–), 42(+)	30(–), 29(+)
2	42(–), 44(+)	39(–), 40(+)	14(–), 13(+)
3	14	38	28
4	12	37	27
5	10*	36*	26*
6	8	35	25
7	6	34	24

‡ Also used for External Clock Input
* Also used for External Clock Enable

2. Connect the ground to any ground pin on the P4, J1, or J5 connector, as appropriate. See Figure 9.2.5, Figure 9.2.6, Figure 9.2.7, and Table 9.2.3 for connector pinouts.



Cross Reference: For more information on external triggers, see the Device Management Programing Chapter of the *ABI/ASF User's Manual*.

External Clock

To connect an external clock signal to the PMC2 card, follow the procedure below.



Cross Reference: You can also use External Clock inputs as an eighth External Trigger signal, see the previous External Triggers topic for details.

1. Connect the external clock + and – signals to the appropriate pin described in Table 4.2.2.

2. If applicable, connect the external clock enable signal to the appropriate pin described in Table 4.2.2.

Table 4.2.3: External Clock Pinouts

Function	Mezzanine P4 Connector	Front Panel J1 Connector	J5 Connector on Auxiliary Cable
External Clock Inputs	58(–), 60(+)‡	43(–), 44(+)‡	43(+), 44(–)‡
External Clock Enable	10*	36*	26*

‡ Also used for External Trigger 0
* Also used for External Trigger 5

4.3 Attaching the Cable Assembly

The commercial and XT PMC2 modules require that you use the included cable assembly for both testing the card and for actual 1553 operation. The cable assembly provides leads to attach the PMC2 to the 1553 bus and optionally, connections for external triggers, external clocks, and IRIG.

1. Attach the cable assembly to the J1 connector on the front panel of the commercial or XT PMC2 card.

Before Testing the PMC

2. Attach 1553 bus terminators to each Bus lead on the cable assembly, as illustrated in Figure 4.3.1.



Cross Reference: Before operating the PMC2 on a 1553 Bus, see the instructions on page 49 of this manual for connecting the PMC2 to a 1553 bus.

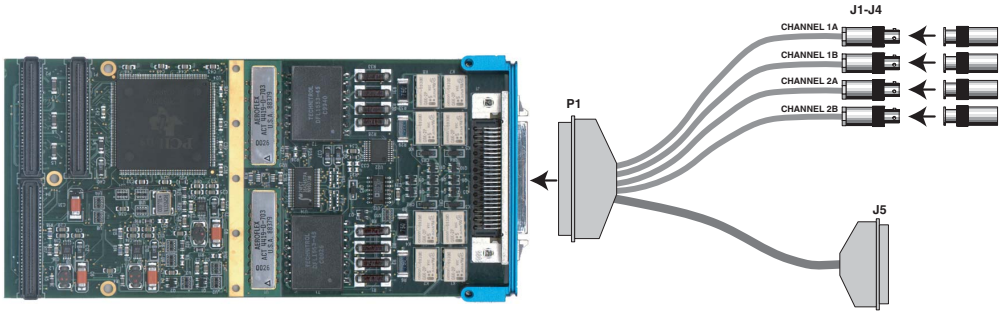


Figure 4.3.1: Attaching the Cable Assembly to the PMC2 (Comm/XT only)

5: Performing System-Specific Installation



Note: These procedures are for use with non-PASS cards. For PASS card installation, refer to the *PASS 3200 User's Manual*.

The sections in this chapter provide instructions on how to install the files specific to your operating system and platform. This chapter provides instructions for the following systems:

- Windows 95/98/ME
- Windows NT
- Windows 2000
- vxWorks
- Installing the INTEGRITY Driver



Note: If you are using a system not covered in this chapter, see Chapter 21 of the *Integrated Avionics Library Reference Manual* for instructions on installing the files for your system.

Note: On ASF-PMC2CC-D cards, the PCI bridge is not used to determine the interrupt vector. Therefore, any function that reads the PCI bridge may return an incorrect interrupt vector.



When setting up interrupts on operating systems that do not automatically configure interrupt vectors, such as VxWorks and Integrity, the computer hardware determines the vector.

To determine the correct vector to the INTA line, refer to the Board Support Package (BSP) for your computer.

For additional information, please contact the SBS Technical Support Department.

5.1 Windows 95/98/ME

Use this procedure with the Integrated Avionics Library, Version 6.3X or later, where PASS-3200 has not been installed.

1. After you have inserted the card into one of the slots in your system and powered up your system, the **New Hardware Found** dialog appears. Select **Driver from disk provided by hardware manufacturer** and click **OK**.
2. Click the **Browse** button.
3. Navigate to the drive where you installed the SBS software.



Note: If you're using Windows 95, the operating system copies the *.vxd* and the *.inf* files automatically.

4. Select *sbs_ver<x.yz>\drivers\sbs.inf*, where *<x.yz>* is the version number of the current release, and click **OK**.
5. If you are prompted to specify the location of the virtual device driver *sbspci.vxd*, click the **Browse** button, select *sbs_ver<x.yz>\drivers\sbspci.vxd*, and click **OK**.



Note: The computer configures parameters such as I/O base, base memory address, and interrupt vector. The *sbs_dev.cfg* (or *dev_cfg.h*) configuration file ignores these parameters.

5.2 Windows NT

Use this procedure with the Integrated Avionics Library, Version 6.3X or later, where PASS-3200 has not been installed.



Warning: In order to run **configmgr**, you must have Microsoft Internet Explorer version 4.0 or higher installed on your system.



Note: You must log on with Administrator privileges before performing these procedures.

1. If you're using the Windows NT operating system, copy the files in the *[drive]:\sbs_ver<x.yz>\drivers\Windows NT* directory, where *<x.yz>* is the version number of the current release, to the *[drive]:\winnt\system32\driver* directory.

2. Execute *[drive]:\sbs_ver<x.yz>\source\utility\configMgr.exe*.



Warning: Do not execute this program from the SBS Resource CD.

3. Select the **ADD PC Board** menu option.

4. Select **PCI Board** from the four board types offered. Once you select the type, follow the appropriate instructions for entering board-specific information.

   For PCI Boards:

   - Select **1553 Dual Device** for the board type from the **Board Type** pull down menu.
   - Select the appropriate board function mode by clicking on the **Mode** radio button under **Board Type**. For 1553 boards, select **ASF** (single function) or **ABI** (multi function).
   - Click the **Apply** button, and proceed to Step 5.

5. Repeat Step 3 and Step 4 for any additional boards.

6. Click on the **Restart driver to update changes** button.

7. When finished, reboot for the changes to take effect.

5.3 Windows 2000

Use this procedure with the Integrated Avionics Library, Version 6.3X or later, where PASS-3200 has not been installed.



Cross Reference: To uninstall the Windows 2000 SBS drivers, see the instructions on page 28 of this manual for Uninstalling SBS Drivers In Windows 2000.



Warning: In order to run **configmgr**, you must have Microsoft Internet Explorer version 4.0 or higher installed on your system.



Note: You must log on with Administrator privileges before performing these procedures.

1. Install your SBS device in your unplugged system.
2. After your SBS device is installed, plug in your system and start Windows 2000.

   The system indicates it found the device and the **Welcome to the Found New Hardware Wizard** window appears.
3. Select **Next**.

   The **Install Hardware Device Drivers** window appears.
4. Select the bottom choice, **Display a list** and click on **Next**.

   The **Hardware Type** window appears.
5. Scroll down to the icon called **Other devices**. This has a (yellow) question mark (?) in front of it. Click **Next**.

   The **Select a Device Driver** window appears.
6. Select the button labeled **Have Disk**.

   The **Install From Disk** window appears.
7. In the lower right portion of the window, select **Browse**.
8. Locate your SBS driver directory for Windows 2000. Select *SBSWDMPC.inf* for the PMC2 card. Select **Open**.

   The **Install From Disk** window reappears.
9. Click the button labeled **OK**.

   The **Select a Device Driver** window appears.
10. Select your device from the list. For example, the PCI version of the WMUX is

the choice **SBS 1553 PCI (PLX 9080)**. Select **Next**.



Note: If you get a warning that it cannot be verified, disregard and close this warning window.

The **Start Device Driver Installation** window appears.

11. Select **Next**.

    The files needed are copied.

    The **Completing the Found New Hardware Wizard** window appears.

12. Select **Finish**.

13. Now, execute *[drive]:\sbs_ver<x.yz>\source\utility\configMgr.exe*, where *<x.yz>* is the version number of the current release of your installed libraries. Click on the 1st blank line, select **Add PC Card** from the menu, then select and click the proper card choice (**PCI Board**).

    Table 5.3.1 lists the windows which can be displayed for the PC card selection choice.

Table 5.3.1: PC Card Selection Choice and Window Displayed

Add PC Card Menu Choice	Window Displayed
ISA Board	PC3 Card Setup
PCI Board	PCI Card Setup
PCMCIA Board	PCMCIA Card Setup
PC104 Board	Setup 1553-PC104

14. In the next window, select **1553 Dual Device** for the proper board type from the drop-down menu and either **ABI** or **ASF**.

15. Restart your computer for the change to take effect.

16. Setup your *SBS_DEV.cfg* file to reflect the device you have. Place this copy, the *unittest.exe* file, and your firmware file (*F066J.DAT* for the WMUX or similar for other cards) in the appropriate directory.

    You should now be able to use your *Unittest.exe* to verify device setup and operation.

Uninstalling SBS Drivers In Windows 2000

Use this procedure with the Integrated Avionics Library, Version 6.3X or later, where PASS-3200 has not been installed.

Note: You must log on with Administrator privileges before performing these procedures.

1. Start Windows 2000.
2. Open the **Control Panel**. Double click the **Add/Remove Hardware** icon.
3. Click **Next** in the 1st window. Once the 2nd window appears, select the bottom choice, **Uninstall/Unplug a device**. Click **Next** to go to the next screen.

   The **Choose a Hardware Device** dialog appears.
4. Scroll down the list until you find the SBS device in the list. Highlight it, then select **Next**.

   The **Completing the Add/Remove Hardware Wizard** window appears.
5. Select **Finish** to complete this stage of the uninstallation.

Note: Windows 2000 might start a hardware troubleshooter help window at this point. This window is not needed; so, close it.

6. In **Explorer**, use browse to find your *Inf* directory. Generally, the location of this directory is *c:\WINNT\Inf*.
7. Look at the files *OEM<X>.inf*, where *<X>* is a number starting at zero (0). There could be 10 or more files. Double click on the file(s) to open them. (If Explorer asks you what to open the file with, use the **Notepad** application. Be sure to check the **Always use this program** checkbox).
8. The top line of the file indicates if it is SBS file or not. Look for the PCI driver which states: *;INF file for SBSWDMPCI32.sys*. After finding *;INF file for SBSWDMPCI32.sys* file, rename or delete this file. Also, rename or delete the associated *OEM<X>.PNF* file.

Note: This file may or may not be in your system, depending on how it was originally installed.

9. (Optional) Search for and rename or remove all SBS drivers in your *c:\WINNT\SYSTEM32*, *c:\WINNT\SYSTEM32\Drivers*. These drivers are prefixed with SBS. Starting in your c:\WINNT directory, search for *SBS*.VXD* and *SBS*.SYS*. Search should locate all of these files.
10. Restart the system to clean out the rest of the files that may be in use.

Note: If you have a PCI device, completely power down the system for at least 5 seconds.

11. Turn off the system and remove the card unless you are going to reinstall the driver.

5.4 vxWorks

The procedure below describes how to build your Integrated Avionics Library project under vxWorks.

1. Modify the *sbs_sys.h* file as follows.

```
#define BITFIELD_1
#define SWAP_BYTES
#define M1553_APP
#define NO_FILE_SYSTEM
#define NORMAL_SCREEN
#define VXWORKS_PCI
#define VXWORKS
```

2. To prepare your project for compiling, add the necessary files into a working directory, including:

   All 1553 *.c* and *.h* files, depending on your project, and all common library files, i.e. dev_mgmt.c and *dio_mgmt.h*, *sbs_sys.h*, etc.



Cross Reference: For more information about compiling your project, refer to Chapter 2: *Compiling Your Application* in the *Integrated Avionics Library Reference* manual.

3. If doing DIO, include the *dio_mgmt.c* and *dio_mgmt.h* files from the *utility* folder.

4. Include the *ll_vxwks.c* file from the *osdepend* folder.

5.5 Installing the INTEGRITY Driver



Note: This information is for use with the with the CM4 BSP only. You may need to make some modification to the driver and/or the BSP in order for the driver work on other BSPs.

This section provides information for building and using the PMC2CC board driver for INTEGRITY.

5.5.1 INTEGRITY Driver Setup

To build the 1553-PMC2CC board INTEGRITY driver, do the following:

- Link the driver to the kernel.
- Initialize the driver's internal structures.

Link the driver to the kernel

You can achieve this by two methods:

- Include the 1553-PMC2CC driver library project (*cm4_m1553.bld*) into the kernel project.
- Add the *libsbsm1553.a* static library into the kernel project. The 1553-PMC2CC driver requires the static *libsbspci2050.a* and *libsbsmisc.a* libraries to be linked with the kernel project.

See Figure 5.5.1 for an example of a kernel project with static libraries linked into the kernel.

Initialize the driver's internal structures

Call the `M1553IODeviceInit()` routine from kernel's main routine. The *m1553.h* file defines `M1553IODeviceInit()`.

Example

```
#include "m1553.h"

int main()
{
    // Initialize all 1553-PMC2-CC boards
    M1553IODeviceInit();

    Exit(0);
}
```

At this point the driver is built into the kernel and ready to use.

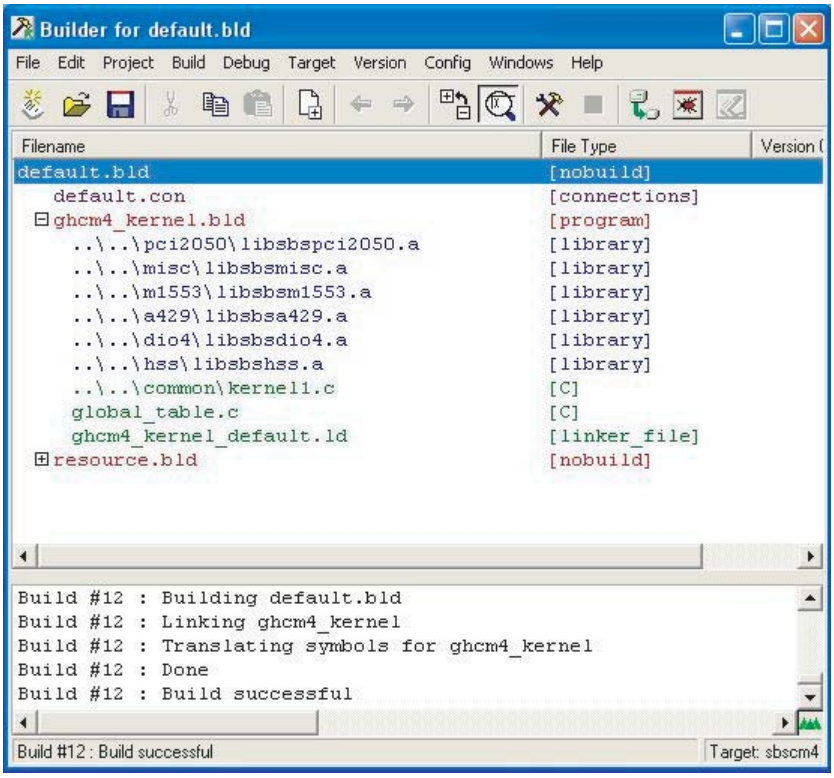


Figure 5.5.1: Example of Kernel Image Project with Static Libraries Linked into the Kernel

6: Running Unit Test

To verify that the PMC2 is properly installed and operational, use the instructions in the following sections to run the Unit Test application:

- Introduction
- Using the Combined (1553, A429, and WMUX) Unit Test Executable
- Setting up the Device Configuration File
- Unit Test Using the Console Mode
- Troubleshooting



Note: You must have an ANSI compatible terminal or driver in order to run Unit Test.

6.1 Introduction

The *\sbs_ver<x.yz>\execs* directory, where *<x.yz>* is the version number of the current release, supplies the executable for Windows for the combined Unit Test application. If you are using one of these operating systems with the PMC2 card and wish to use the combined Unit Test, proceed to the next section. If you do not have access to the distributed Unit Test executables, if there is not a precompiled executable for your operating system, or if you wish to run a stand-alone 1553 Unit Test, use the instructions in the Compiling Your Application Chapter of the *Integrated Avionics Library Reference Manual* to compile a new Unit Test executable.

You can begin using the Unit Test by going to Unit Test Using the Console Mode in Section 6.4.

6.2 Using the Combined (1553, A429, and WMUX) Unit Test Executable

This section describes the procedures for using the combined Unit Test. The topics are as follows:

- Operating Systems with File Systems
- Operating Systems without File Systems

6.2.1 Operating Systems with File Systems

To use the precompiled, combined Unit Test for Windows, complete the following steps:

1. Copy the Unit Test executable from the *\sbs_ver<x.yz>\execs\unittest.exe* directory to the *\sbs_ver<x.yz>\working* directory (where *<x.yz>* is the version number of the current release).
2. Copy the *sbs_dev.cfg* file from the *\sbs_ver<x.yz>\source\common\library* directory to the *\sbs_ver<x.yz>\working* directory.
3. Copy the firmware files from the *\sbs_ver<x.yz>\firmware* directory to the *\sbs_ver<x.yz>\working* directory.
4. Set up the *sbs_dev.cfg* file as described in the next section, *Setting up the Device Configuration File*.

6.2.2 Operating Systems without File Systems

1. Refer to the Compiling Your Application Chapter of the *Integrated Avionics Library Reference Manual* to determine the necessary files and appropriate compiler directives. Copy the appropriate files into the *\sbs_ver<x.yz>\working* directory, where *<x.yz>* is the version number of the current release.
2. Set up the *dev_cfg.h* file as described in the next section, *Setting up the Device Configuration File*.
3. Compile your Unit Test application as described in the Compiling Your Application Chapter of the *Integrated Avionics Library Reference Manual*.

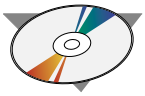
6.3 Setting up the Device Configuration File

You must define the device parameter values for your SBS device(s) in one of the following two files:

- *sbs_dev.cfg*, if your operating system has a file system
- *dev_cfg.h*, if your system does not have a file system

The *sbs_dev.cfg* device configuration file is an ASCII text file containing information that your application uses to initialize one or more SBS devices. The SBS device's initialization procedure calls a parser function (**`sbs_parse_config_file()`**) that reads the information in the configuration file. The parser function must be able to correctly read and verify the information for an SBS device before further library calls can be made. Because the **`sbs_parse_config_file()`** function reads the *sbs_dev.cfg* file at run-time, you can modify *sbs_dev.cfg* without recompiling the application.

If your operating system does not have a file system, you should use the *dev_cfg.h* file instead of the *sbs_dev.cfg* file. The *dev_cfg.h* file is an array of strings with the same format and keywords as the *sbs_dev.cfg* file. It parses in the same manner as the *sbs_dev.cfg* file. However, you must recompile your application each time you edit *dev_cfg.h* for your changes to take effect.



Software Reference: The location of *sbs_dev.cfg* and *dev_cfg.h* configuration files are in the directory in which you installed the Integrated Avionics Library under the *\install\library\source\common\library* directory.

The following sections provide a description and examples of the *sbs_dev.cfg* and *dev_cfg.h* files:

- Format
- Keywords
- sbs_dev.cfg File
- dev_cfg.h File

6.3.1 Format

The format of the *sbs_dev.cfg* and *dev_cfg.h* files resembles an *.ini* file found on most PCs. Values contained in this file vary based on your SBS device(s), system hardware, and operating system. Every device requires a separate set of keyword entries in the configuration file. (Thus, each single-device card requires a single set of keyword entries, and each multi-device card requires multiple sets.) The file is set up as follows:

- An equal sign (=) links each keyword with a value.
- The variable **<num>** represents a numerical value.
- The variable **<name>** represents a case-sensitive character string.
- The parser recognizes the following characters appended to the value string:
  - **b** (binary)
  - **o** (octal)
  - **h** (hex)
  - **d** (decimal; the default if no character is present)
- A semicolon at the beginning of a line denotes a comment line, and the parser ignores the entire line.

6.3.2 Keywords

Table 6.3.1 describes the keywords required in the *sbs_dev.cfg* or *dev_cfg.h* file for configuring the PMC2 card.

Table 6.3.1: Required Configuration File Keywords for the PMC2

Keyword=<Specifier>	Required for:	Description
[DEVICE=<num>]	All operating systems and platforms	This line is required before all other keywords for a device. <num> represents a unique number, starting at 1, that identifies the SBS device.
dd_name=<name>	Windows NT, 95, and 98; some UNIX systems (not required for vxWorks)	<name> specifies the UNIX or Windows device driver name. It contains a text string with the name of the device driver node associated with the SBS device. For UNIX, <name> is the filename of the driver in the */dev* directory. For IRIX 6.4/6.5 systems, <name> is the XIO pathname that points to the memory space of the PCI bus for your card. For Windows NT, this value must correspond exactly to the name of the instantiated driver for the device, which can be found in the device directory using the Device Manager. It is of the form `sbspci320`, `sbspci321`, etc. For Windows 95/98, the driver name is `SBSPCI`.
device_type=<name>	All operating systems and platforms	<name> specifies the type of SBS card being used.

Keyword=<Specifier>	Required for:	Description
`firmware=<name>`	All operating systems and platforms	`<name>` specifies the firmware filename. If you define `NO_FILE_SYSTEM` in *sbs_sys.h*, the firmware will be loaded from a data array found in *firmware.h*. The array is specified by the "`firmware=`" entry in the *dev_cfg.h* configuration file. You can generate the *firmware.h* file using the utility program *setup_fw*.c. Devices that have flash memory still require this field to reprogram the flash memory. **Note:** Verify that the firmware filename listed in the *sbs_dev.cfg* or *dev_cfg.h* file is the same as that of the firmware file provided with the software distribution.
`relative_position=<num>`	Windows 95/98	`<num>` is the sequence number of the card as found in the bus slots. The first board is 0, the second board is 1, etc.

6.3.3 *sbs_dev.cfg* File

Note: If your operating system does not have a file system, you must use the *dev_cfg.h* file instead of *sbs_dev.cfg*. See page 39 for information on *dev_cfg.h*.

If your operating system has a file system, define the device parameter values for each SBS device in the *sbs_dev.cfg* file. Modify *sbs_dev.cfg* as follows:

1. Change directories to the *\sbs_ver<x.yz>\working* directory, where *<x.yz>* is the version number of the current release, if you are not already there.
2. Edit the *sbs_dev.cfg* file and verify that the settings are correct for the device type, firmware file(s), and number of devices you are using.
3. Save the file.

Examples of the parameters used for the *sbs_dev.cfg* file appear on the following page for a dual-device ABI-PMC2 card running under the following systems:

- Windows 95/98/ME
- Windows NT
- vxWorks

These examples show only the required parameters. In the actual *sbs_dev.cfg* file you would remove the comment (semicolon preceding the parameter). All remaining lines are commented out (preceded by a semicolon). If you are using a single-device card, you should uncomment keywords only for device 1. If you are running your card under an operating system other than those in the examples, refer to Table 6.3.1 to determine the keywords that your operating system requires.



Cross Reference: For an example of the complete *sbs_dev.cfg* file, refer to the *Integrated Avionics Library Reference Manual*.



Note: The values required for your system, including the firmware filename, may differ from those shown below.

Windows 95/98/ME

```
[DEVICE=1]
device_type=M1553_PMC2_1
dd_name=sbspci
firmware=f201e.dat
relative_position=0
[DEVICE=2]
device_type=M1553_PMC2_2
dd_name=sbspci
firmware=f201e.dat
relative_position=0
```

Windows NT

```
[DEVICE=1]
device_type=M1553_PMC2_1
dd_name=sbspci320
firmware=f201e.dat
[DEVICE=2]
device_type=M1553_PMC2_2
dd_name=sbspci321
firmware=f201e.dat
```

vxWorks

```
[DEVICE=1]
device_type=M1553_PMC2_1
firmware=f201e.dat
irq_level=0Fh
[DEVICE=2]
device_type=M1553_PMC2_2
firmware=f201e.dat
irq_level=0Fh
```

6.3.4 *dev_cfg.h* File



Note: If your operating system has a file system, you should use the *sbs_dev.cfg* file instead of *dev_cfg.h*. See page 37 for information on *sbs_dev.cfg*.

This file contains the device information to be used in lieu of *sbs_dev.cfg* for embedded systems that do not have a file system. Modify *dev_cfg.h* as follows:

1. Change directories to the *\sbs_ver<x.yz>\working* directory, where *<x.yz>* is the version number of the current release, if you are not already there.
2. Initialize the **`dev_cfg`** string array with the configuration parameters for each card. The keywords in *dev_cfg.h* are identical to those in *sbs_dev.cfg*; see Table 6.3.1 on page 36 for the keyword descriptions.
3. Save the file.
4. Refer to the Sample 1553 Applications Chapter of the *ABI/ASF User's Manual* for information on compiling your application.

An example of the parameters used for the *dev_cfg.h* file appears below for an a dual-device ABI-PMC2 card running under vxWorks.

This example shows only the required parameters. In the actual *dev_cfg.h* file you would remove the comment (semicolon preceding the parameter and following the leading quotation mark). All remaining lines are left commented out (preceded by a semicolon after the leading quotation mark). If you are using a single-device card, you should uncomment keywords only for device 1. If you are running your card under an operating system other than the one in the example, refer to Table 6.3.1 to determine the keywords that your operating system requires.



Cross Reference: For an example of the complete *sbs_cfg.h* file, refer to the *Integrated Avionics Library Reference Manual*.



Note: The values required for your system, including the firmware filename, may differ from those shown below.

```
char *dev_cfg_array[] = {
"[DEVICE=1]",
"device_type=M1553_PMC2_1",
"firmware=f201e.dat",
"[DEVICE=2]",
"device_type=M1553_PMC2_2",
"firmware=f201e.dat"
};
```

6.4 Unit Test Using the Console Mode

This sections gives some of the basic procedures using the Console Mode Unit Test. This section contains the following procedures:

- Starting Unit Test Using the Console Mode
- Opening the 1553 Device
- Running Built-in Tests (BITs)
- Initializing the 1553 Device
- Exiting Unit Test

6.4.1 Starting Unit Test Using the Console Mode



Tips:
Press the carriage return key (**Enter**) to complete menu selections and enter responses to prompts.

Press **Enter** to restore a Unit Test menu following a failure.

1. Execute the Unit Test application.

   As soon as you execute Unit Test, it parses the configuration file (either *sbs_dev.cfg* or *dev_cfg.h*). If your configuration file parses without error, the menu shown in Figure 6.4.1 appears.

```
                    SBS Technologies, Inc.
                M1553 Interface Library Unit Test
                  Version X.YZ    Build MMM DD YYYY

                    1 - Device Management Tools
                    2 - Bus Controller Tools
                    3 - Remote Terminal Tools
                    4 - Bus Monitor Tools

                    q - Quit Unit Test

                Selection ? >
```

Figure 6.4.1: M1553 Interface Library Unit Test Menu



Note: Where shown, X.YZ is the version number of the current release and MMM DD YYYY is the date the current build was compiled.



Note: If you are using the precompiled unit test, select **m** at Integrated Avionics Library Unit Test screen and press return. The *M1553 Interface Library Unit Test Menu as shown in* Figure 6.4.1 appears.

If a parser error occurs, an error screen may appear instead of the Avionics Interface Library Unit Test Menu. Figure 6.4.2 shows a sample parser error screen. Table 6.5.1 describes the common parser error messages that you may encounter.

```
                    SBS Technologies, Inc.
                Integrated Avionics Library Unit Test
                    Version X.YZ Build XX.YY.ZZ

                  Failure parsing configuration file!
                  Device #1 missing "firmware=" keyword.

                  'Q' to quit, return to reparse.

                  Selection ? >
```

Figure 6.4.2: Sample Parser Error Screen

2. If a parser error occurs, proceed to the troubleshooting procedures in Section 6.5.1.

   If no parser error occurs, the menu shown in Figure 6.4.1 appears.

6.4.2 Opening the 1553 Device

1. To select **Device Management Tools**, enter **1**.

   The menu shown in Figure 6.4.3 appears.

```
              1553 Device Management Tools                |               pg1
                                                          |
  1 - Init Device <Steps 2-7>   a - Get Device Clock      |    n - Next Pg->
  2 -  Open Device              b - Set Device Clock      |
  3 -  Load Firmware                                     |    p - Pick dev #
  4 -  Start Application        d - Display Error Tbl    |
  5 -  Init Chan or V7 Mem      e - Clear   Error Tbl    |    r - r/w Ram
  6 -  Init Interrupt Q                                  |
  7 -  Create BSM Buffers                                |    s - Start i/o
                                                         |    t - sTop  i/o
  9 - Set ASF Mode              i - Execute BIT          |
  0 - Get Device Info           j - Close Device         |    x - eXit <-Pg
                                                         |
--------------------------------------------------------------------------------
Selection? >



Messages: Device # defaulted to 1.
--------------------------------------------------------------------------------

                  Device #1:  M1553_xxx_1  is  CLOSED                fyyy.dat
```

Figure 6.4.3: 1553 Device Management Tools Pg1 Menu



Note: Where shown, xxx is the card type (PCI, cPCI, etc.) of the card installed and fyyy.dat is the firmware being used.

Note: The device defaults to the first device number in the group of devices (in this case device #1). To select an alternate device number, enter **p**. At the "Device Number? >" message, enter the appropriate device number. The message "Device number changed to X. Hit ENTER to continue." appears (X is the number of the selected device). Press Enter. The device number selected appears at the bottom of the 1553 Device Management Tools menu.

2. To select **Open Device**, enter **2**.

   The messages shown in Figure 6.4.4 appear.

```
--------------------------------------------------------------------------------
Selection: 2

          Device #1 opened.
Messages: Hit ENTER to continue.
--------------------------------------------------------------------------------
```

Figure 6.4.4: Open Device Messages

3. To continue, press **Enter**.

   The area between the dashed lines clears. The bottom line displays "Device #X: M1553_xxx_1 is OPEN" where X is the number of the device and xxx is the card type.

6.4.3 Running Built-in Tests (BITs)

1. To select **Execute Built-In Tests**, enter **i**.

   If you are prompted to enter a firmware source, enter the appropriate number to load from flash, from file, or via driver (depending on the type of card and operating system you are using, you may see only one of these options at the prompt).

   The messages and prompt shown in Figure 6.4.5 appear.

```
--------------------------------------------------------------------------------
Selection: i
For the built in test to work properly, a cable assembly with terminators
must be attached to the ABI/ASF card and the bus must be quiet.

Enter q to quit, or ENTER to start test.

Messages:
--------------------------------------------------------------------------------
```

Figure 6.4.5: Execute Built-in Tests Verification Screen

2. Verify that the cable assembly with terminators is attached to the ABI/ASF card (see page 22 for instructions).

3. Press **Enter**.

   The messages and prompt shown in Figure 6.4.6 appear.

```
--------------------------------------------------------------------------------
Selection: i
'0' = [FILE]; '1' = FLASH; '2' = DRIVER
Firmware source? >


Messages:
--------------------------------------------------------------------------------
```

Figure 6.4.6: Firmware Selection Screen

4. Enter the appropriate number to load from file, from flash, or via driver (depending on the type of card and operating system you are using, you may see only one of these options at the prompt) and press **Enter**.

   If the tests are successful, messages and prompts similar to Figure 6.4.7 will appear over a 10- to 12-second period.

```
--------------------------------------------------------------------------------
Selection: i
'0' = [FILE]; '1' = FLASH; '2' = DRIVER
Firmware source? > 1
Please Wait (Takes 6 to 9 seconds)...
          BIT passed!
          Device closed. Initialize before running.
Messages:  Hit ENTER to continue.
--------------------------------------------------------------------------------
```

Figure 6.4.7: Execute Built-in Tests Passed Screen

   If the tests are unsuccessful, error messages appear.

5. To continue (in either case), press **Enter**.

   The area between the dashed lines clears. The bottom line displays "Device #X: M1553_xxx_1 is CLOSED" where X is the number of the device and xxx is the card type.



Tips:
If the tests are unsuccessful, check for proper termination of the bus.

Contact SBS technical support if you are unable to correct the problem.

6.4.4 Initializing the 1553 Device

1. To select **Initialize Device**, enter **1**.

   The messages and prompts shown in Figure 6.4.8 appear one at a time.

```
--------------------------------------------------------------------------------
Selection: 1

Interrupt queue length [4]? >
Seq. monitor length [1000]? >
'0' = [FILE]; '1' = FLASH; '2' = DRIVER
Firmware Source? >

Messages:
--------------------------------------------------------------------------------
```

   Figure 6.4.8: Initialize 1553 Device Prompts

2. To set the queue length to 4 entries, press **Enter** at the Queue Length prompt.
3. To set the sequential monitor length to 1000 words, press **Enter** at the Sequential Monitor Length prompt.
4. If you are prompted to enter a firmware source, enter the appropriate number to load from file, from flash, or via driver (depending on the type of card and operating system you are using, you may see only one of these options at the prompt).

   If the initialization process completes successfully, the message "Device initialized" appears. If this process does not complete successfully, an error appears between the dashed lines.

5. Press **Enter** to continue.

   The area between the dashed lines clears. The bottom line displays "Device #X: M1553_xxx_1 is STOPPED" where X is the number of the device and xxx is the card type.

Troubleshooting If a failure occurs during initialization, proceed to Section 6.5.2.

6.4.5 Exiting Unit Test

1. To return to the M1553 Interface Library Unit Test menu, enter **x**.

   The screen clears, and the M1553 Interface Library Unit Test Menu appears.

2. To exit from the Unit Test application, enter **q**.

   The prompt shown in Figure 6.4.9 appears.

```
Are you sure you want to quit? ([y]/n) >
```

Figure 6.4.9: Prompt to Quit



Note: If you are using the precompiled unit test, the Integrated Avionics Library Unit Test screen appears next. Select **q** at Integrated Avionics Library Unit Test screen and press return. The prompt shown in Figure 6.4.9 appears.

3. Enter **`y`**.

6.5 Troubleshooting

This section discusses troubleshooting if a parser error occurs during the Unit Test using the Console Mode or if a failure occurs during initialization. This section contains the following procedures:

- Parser Error
- Initialization Error

6.5.1 Parser Error

Unit Test Using Console Mode

1. Identify parser error using Table 6.5.1.
2. Correct the error in the configuration file. If you are using *sbs_dev.cfg*, press **Return** from the Unit Test error screen to reparse. If you are using *dev_cfg.h*, exit and recompile your code. Repeat step 1 in Section 6.4.



Note: After the configuration file parses without error (i.e., once you see the menu shown in Figure 6.4.1), you are ready to continue.

Table 6.5.1: Parser Error Messages

Error Message	Diagnosis
Failure parsing configuration file! Error opening <filename> file!	The indicated file could not be opened.
Failure parsing configuration file! <keyword>=<num> for device #xx is invalid.	The parser found a value out of limits in the configuration file. Correct the line containing <keyword>.
Failure parsing configuration file! Application does not support <name> devices!	The application type <name>_APP (where <name> is the avionics bus type) is not defined in the *sbs_sys.h* file.
Failure parsing configuration file! Device #xx missing "<keyword>=" keyword.	The specified <keyword>, which is required for the specified device (Device #xx, where xx is in the range 1 to SBS_MAX_DEV), was not found.
Failure parsing configuration file! No device defined in configuration file!	The Device keyword was not found in the configuration file.
Failure parsing configuration file! "<name>" is not a valid device type.	The <name> specified for the device_type keyword is not valid. Choose a valid device type from the list included in the configuration file.
Failure parsing configuration file! "<filename>" not defined in the firmware.h file.	The <filename> specified for the firmware keyword in *dev_cfg.h* does not match any firmware filename in the *firmware.h* file.
Failure parsing configuration file! "<keyword>=<num>" on line xx: duplicate entry.	A value for the <keyword> specified on line xx has already been parsed for this device number.
Failure parsing configuration file! Line number xx has more than 132 characters.	The number of characters on line xx exceeds the maximum number that can be read.
Failure parsing configuration file! "Device=0" on line xx must be greater than 0.	The Device keyword must have a value greater than zero.
Failure parsing configuration file! "Device=<num>" on line xx exceeds SBS_MAX_DEV.	The value <num> is greater than the SBS_MAX_DEV value specified in the *dev_mgmt.h* file.
Failure parsing configuration file! "Device=<num>" is a noncontiguous <name> 2nd device.	The values of the Device keywords for the first and second devices of the indicated board (<name> = PC3 or PC16) must be consecutive numbers.

6.5.2 Initialization Error

If a failure occurs during initialization, do the following:

1. Turn off the system, and then physically remove and reseat the card in the host computer.
2. Check all cable connections and verify that they are secure.
3. Try initializing the card again.
4. If a failure still occurs, refer to Table 6.5.2.

Table 6.5.2: Initialization Errors and Error Messages

Error	Diagnosis
Computer system locked up	Interrupt request (IRQ) level is in use. Specify a different IRQ level in the *sbs_dev.cfg* or *dev_cfg.h* file.
One of the following error messages appeared:	
Initialize device failed! **sbs_open_device(): The device driver failed to initialize**	The device driver was not started on reboot. Verify that the device is started in Windows NT or that the device driver is selected and loaded in Windows 95/98.
Initialize device failed! **sbs_open_device(): Shared memory fail**	The **base_address** specified in the *sbs_dev.cfg* or *dev_cfg.h* file is invalid or there is a problem with the operating system device driver. On UNIX systems, verify that the correct filename is specified for the **dd_name** keyword in *sbs_dev.cfg* or *dev_cfg.h*.
Initialize device failed! **sbs_load_ram(): Download error**	The **base_address** specified in the *sbs_dev.cfg* or *dev_cfg.h* file is invalid. For MS-DOS, Windows 3.x, and Windows 95/98/NT, verify that the required memory region is excluded in the *config.sys* file.
Initialize device failed! **sbs_load_ram(): File open error**	Your application could not open the firmware file specified in *sbs_dev.cfg* or *dev_cfg.h*. Verify that the firmware filename specified in *sbs_dev.cfg* or *dev_cfg.h* is correct and that the specified firmware file is in your *\working* directory.
Initialize device failed! **sbs_start_firmware(): Start firmware failure**	The firmware did not properly start. Verify that the proper firmware file(s) are specified in *sbs_dev.cfg* or *dev_cfg.h*.
Open device failed! **sbs_open_device(): Device probe fail**	It is possible that the PCI device driver may be conflicting with the resources of another device. Try changing the memory setting for the PCI device in system resources. Conflicts with video display adapters using ROM Shadowing have resulted in resource conflicts.



Cross Reference: For further help with initialization failures, see *Customer Support Services* in the Introduction Chapter of the *MIL-STD-1553ABI/ASF User's Manual*.

7: Connecting to the 1553 Bus

If the Unit Test procedures above completed without error, you may proceed with MIL-STD-1553 operations. However, you must first connect each channel of the card to an actual 1553 bus by completing the following procedure:

1. Remove the terminators from the Bus A and Bus B cables (these cables are marked "Channel 1A" and "Channel 1B," respectively).
2. Attach the Bus A cable to the stub connector on an appropriate transformer-coupled device (see Figure 7.0.1).
3. Attach the Bus B cable to the stub connector on an appropriate transformer-coupled device (see Figure 7.0.1).

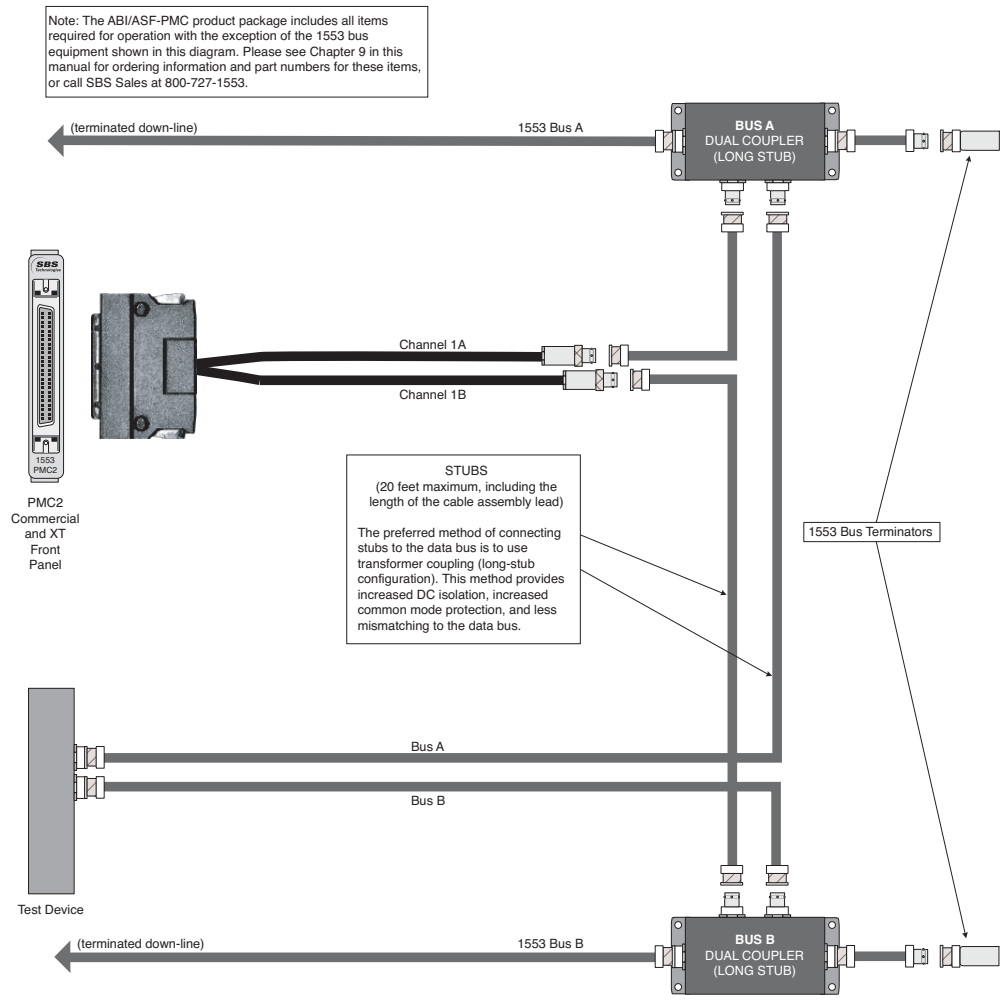


Figure 7.0.1: Connecting the PMC2 to a 1553 Bus

8: What to Do Next

If your card is properly connected to a 1553 bus, you may proceed in one or more of the following ways:

➢ Run sample 1553 applications.



Cross Reference: If you are a first-time user, SBS also recommends that you execute the sample 1553 applications that are included with the product shipment. These applications are detailed in Sample 1553 Applications Chapter 0f the *MIL-STD-1553 ABI/ASF User's Manual*.

➢ Build a custom 1553 application.



Cross Reference: If you are an advanced user, you may wish to begin designing your own 1553 application. See Chapters 4–15 in the *MIL-STD-1553 ABI/ASF User's Manual* for complete details.

9: Hardware Specifications

This chapter presents information about the ABI/ASF/ESF-PMC2 Series of hardware, including:

- General Product Information
- Physical Specifications
- Operational Specifications

The General Product Information section contains general information for the ABI/ASF-PMC2 cards, including the cage code number, extended warranty information, conformance to MIL-STD-1553 electrical specifications, bus equipment part numbers and ordering instructions, and external signal characteristics.

In the Physical Specifications section, you can find information about component locations, dimensions, weight, jumper and DIP switch settings, connectors, pinouts diagrams, and where you can order connecting cables and terminators.

The Operational Specifications section contains information about the operational aspects of the card, including temperatures, MTBF, and hardware reset.

9.1 General Product Information

The information in this section applies to the ABI/ASF/ESF-PMC2 Series of hardware. This section contains the following topics:

- Cage Code Number
- Extended Warranty Information
- Conformance to MIL-STD-1553 Electrical Specifications
- Bus Equipment Part Numbers and Ordering Instructions
- External Signal Characteristics

9.1.1 Cage Code Number

0BAS8

9.1.2 Extended Warranty Information

SBS offers a comprehensive maintenance service for the ABI/ASF/ESF products. Even though SBS boards rarely fail, these services assure that the end user has thorough coverage and minimal down time in case of a failure.

SBS products include a two-year, parts and labor warranty. You can purchase an extended warranty to extend this service beyond the second year. This provides you with 10-day turn-around for the repair of a module (or provides a replacement module at no cost). Large-quantity repairs may require a longer turn-around time. The cost is minimal and agreements are normally for one-year periods.

For more information or to receive a copy of the maintenance agreement, contact SBS Technologies at one of the numbers listed on the inside cover of this manual and specify “Warranty Support.”

9.1.3 Conformance to MIL-STD-1553 Electrical Specifications

All SBS products conform to the 1553 electrical specifications illustrated in Figure 9.1.1.

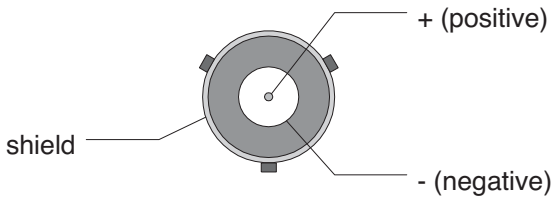


Figure 9.1.1: 1553 Electrical Specifications (as Viewed from End of Connector)

9.1.4 Bus Equipment Part Numbers and Ordering Instructions

You can purchase the bus equipment listed in Table 9.1.1 from SBS Technologies.

Table 9.1.1: OPE 1553 Bus Hardware Available from SBS Technologies

Part Number	Description
BUS-2	Dual-redundant bus with dual-stub couplers: four 15'-cables, four terminators, two 2-stub couplers
BUS-3	Dual-redundant bus with three-stub couplers: six 15'-cables, four terminators, two 3-stub couplers
Single Stub	Single-stub bus coupler
2 Stub	2-stub bus coupler
3 Stub	3-stub bus coupler
4 Stub	4-stub bus coupler
TERM	78-ohm terminator
CAB-COM-ZZ	Commercial-grade cable with PL-75 connectors (ZZ=length in feet)
CAB-MIL-ZZ	MIL17-grade cable with PL-75 connectors (ZZ=length in feet)
BUS-R	Regenerates bus signals for extending a MIL-STD-1553B Notice 2 compliant bus by 100 meters or 330 feet. Two BUS-R products are required for a dual-redundant bus.
BUS-C	Provides an RS-422 interface to SBS ABI modules to extend a MIL-STD-1553B stub from 20 to 300 feet
RS422-C	Cable Option: 300-ft RS-422 Cable for BUS-C

For more information, contact SBS at one of the numbers listed on the inside cover of this manual and specify "Sales Support."

Miscellaneous 1553 components are also available from the following vendor:

MilesTek
1506 Interstate 35 W
Denton, Texas 76207-2402
Attn: Al Stenzel
800-524-7444 or 940-484-9400
FAX: 940-484-9402

9.1.5 External Signal Characteristics

External Trigger

The external trigger feature is standard on the ABI/ASF/ESF-PMC2. The external trigger is a transistor-transistor logic (TTL) signal having the characteristics shown in Figure 9.1.2.

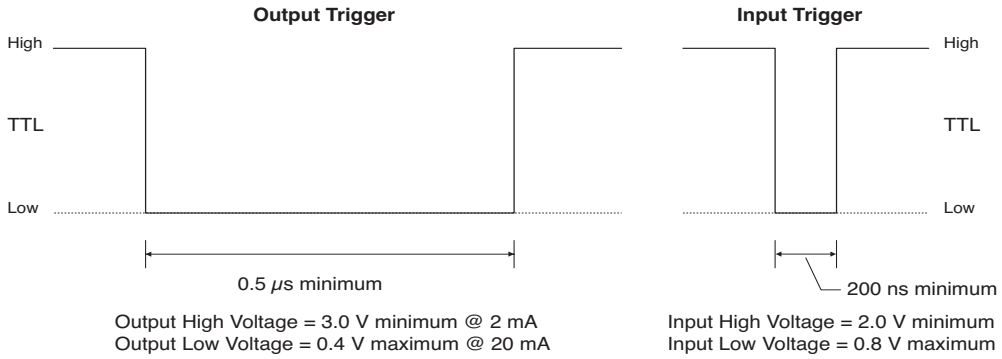


Figure 9.1.2: Characteristics of ABI/ASF External Trigger

Note: When the low or high level is selected for the input trigger, the signal must remain at that level for at least 20 microseconds. Otherwise, the firmware may not delete the input trigger.

External Clock

The ABI/ASF/ESF-PMC2 firmware supports the external clock feature. The external clock requires a differential signal with the specifications listed in Table 9.1.2.

Table 9.1.2: External Clock Differential Signal Specifications

Specification	Value
Recommended Differential Input Voltage (measured from + External Clock pin to – External Clock pin)	+12 V
Recommended Input Voltage (measured at either + pin or – pin)	Minimum: –7 V Maximum: +12 V
Differential Input Threshold	Minimum: –0.2 V Maximum: +0.2 V
Typical Output Voltages	High (minimum): +2.7 V Low (maximum): +0.5 V

IRIG

The IRIG clock feature is an option which must be ordered at the time of purchase. It is available on all ABI/ASF products. ABI/ASF products which include this option will accept an IRIG input signal compatible with the IRIG-B standard.

Note: The IRIG input impedance for all ABI/ASF products is 10 kΩ.

9.2 Physical Specifications

The information in this section describes the physical specifications for the ABI/ASF/ESF-PMC2 cards. This section contains the following topics:

- Board Layout and Mezzanine Connector Pinouts
- Board Dimensions
- Board Weight
- Connector Descriptions and Pinouts
- Part Numbers and Ordering Instructions

9.2.1 Board Layout and Mezzanine Connector Pinouts

The layout of the commercial and XT versions of the ABI/ASF-PMC2 board appears in Figure 9.2.1. Figure 9.2.2 shows the locations of the mezzanine connectors on the ABI/ASF-PMC2 conductive cooled version. Figure 9.2.3 shows the locations of the mezzanine connectors on the ASF-PMC2CC-1-D, ASF-PMC2CC-2-D and ESF-PMC2CC-1 conductive cooled version. Figure 9.2.4 shows the locations of the mezzanine connectors on the ESF-PMC2CC-2 conductive cooled version.

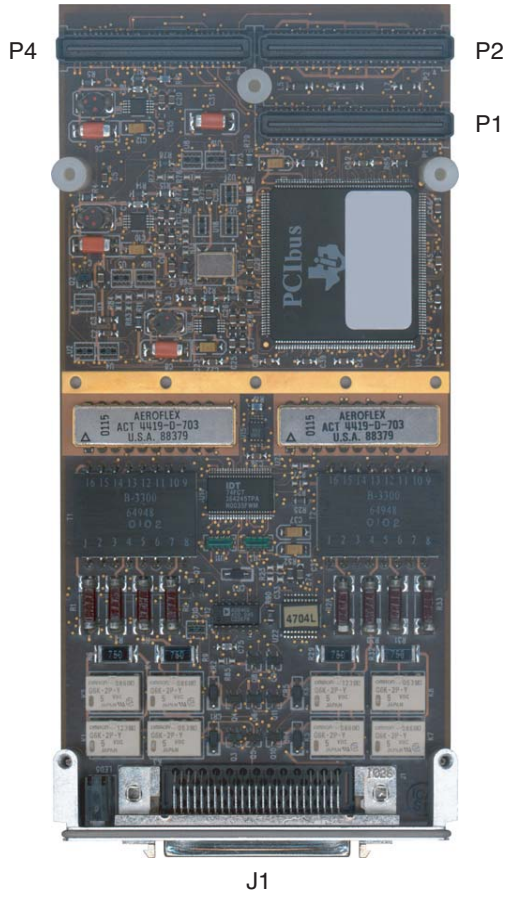


Figure 9.2.1: 1553 PMC2-Commercial and XT Layout and Connector Locations

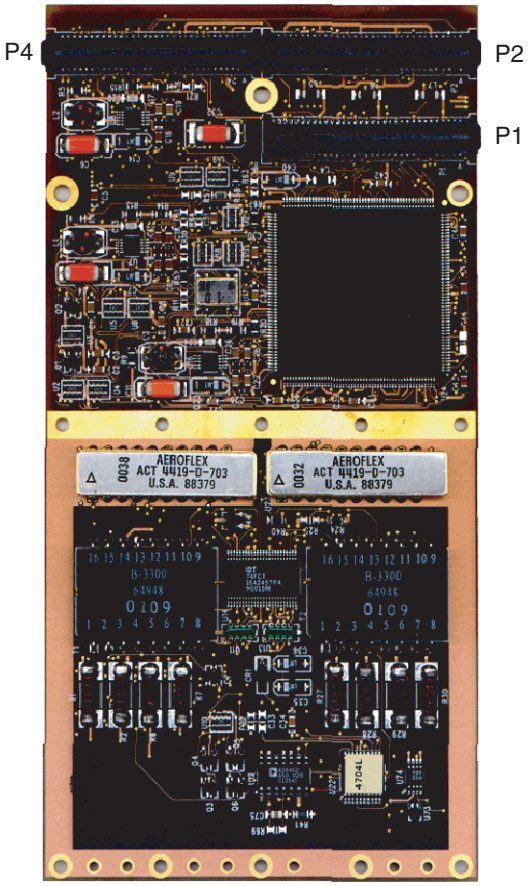


Figure 9.2.2: 1553 PMC2-Conductive Cooled Layout and Connector Locations

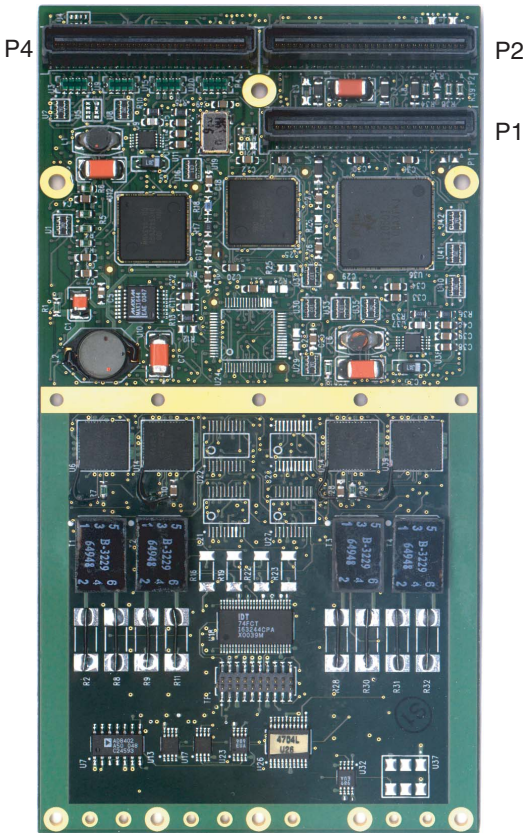


Figure 9.2.3: 1553 ASF-PMC2CC-1-D, ASF-PMC2CC-2-D and ESF-PMC2CC-1 Conductive Cooled Layout and Connector Locations

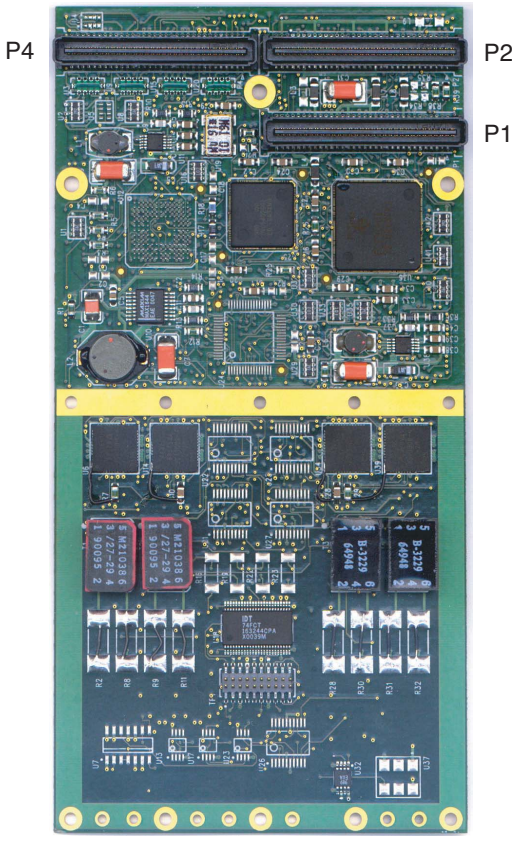


Figure 9.2.4: 1553 ESF-PMC2CC-2 Conductive Cooled Layout and Connector Locations

9.2.2 Board Dimensions

Commercial and XT versions: 149 mm x 74 mm (5.885" x 2.913")

Conductive cooled versions: 143.75 mm x 74 mm (5.660" x 2.913")

9.2.3 Board Weight

ASF-PMC2CC-2-D: 0.0698 kg (0.1539 lb)

9.2.4 Connector Descriptions and Pinouts

Figure 9.2.5 shows the pinouts for the J1 connector. Figure 9.2.6 displays the mezzanine P4 connector pinouts. Figure 9.2.7 shows the mezzanine P4 connector pinouts for the PMC2CC-2-D. Table 9.2.1 compares the pin assignments of the PMC2 with the ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2. Table 9.2.2 gives the pin function descriptions for the ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2.

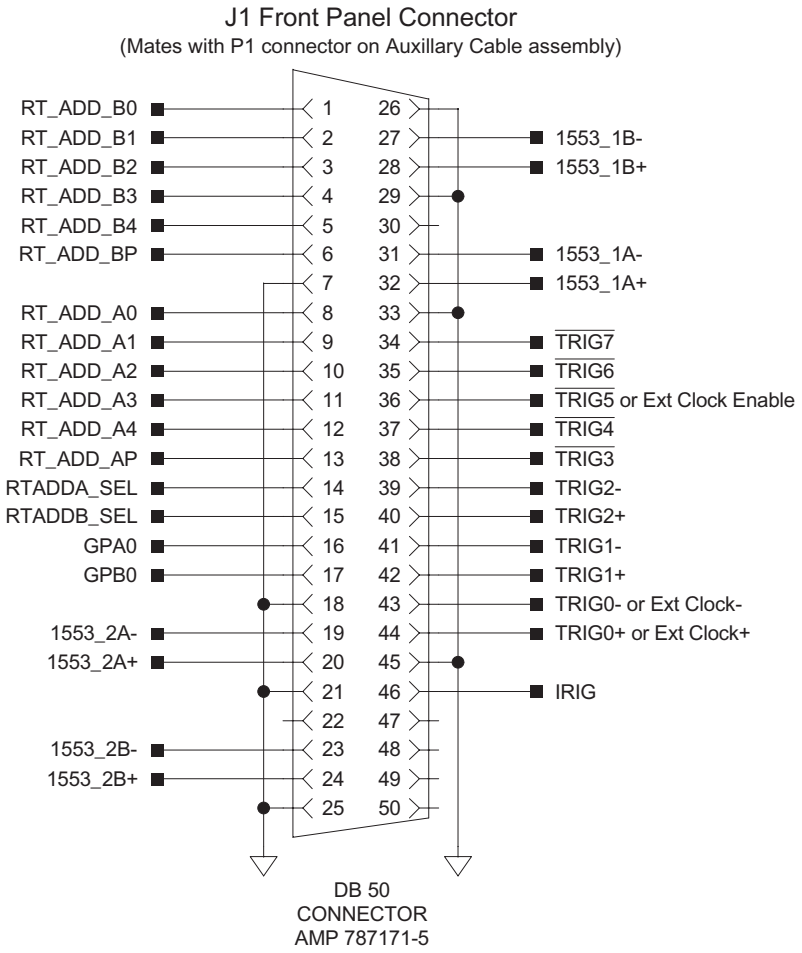


Figure 9.2.5: PMC2 J1 Connector Pinouts

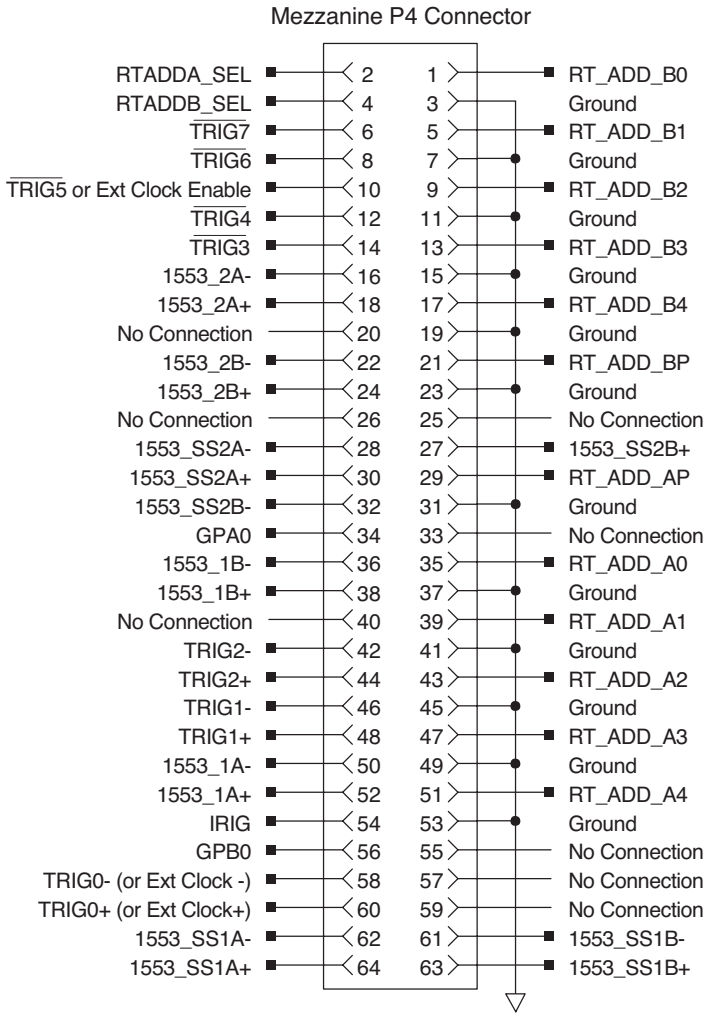


Figure 9.2.6: PMC2 Mezzanine P4 Connector Pinouts

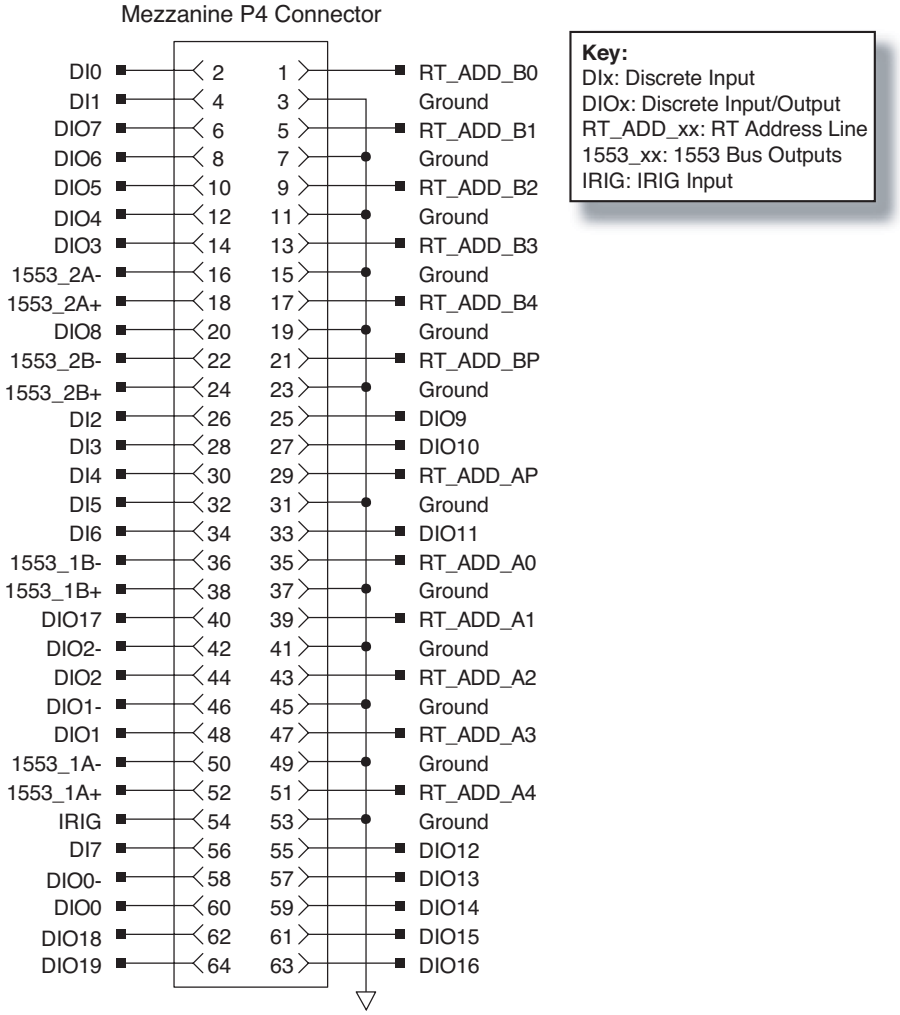


Figure 9.2.7: Mezzanine P4 Connector Pinouts -ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2

Table 9.2.1: P4 Pin Assignments ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2 Compared with PMC2

Pin	Function	Pin	Function
1	RT_ADD_B0	2	DI0
3	GND	4	DI1
5	RT_ADD_B1	6	DIO7 (Same as PMC2 /TRIG7)
7	GND	8	DIO6 (Same as PMC2 /TRIG6)
9	RT_ADD_B2	10	DIO5 (Same as PMC2 /TRIG5)
11	GND	12	DIO4 (Same as PMC2 /TRIG4)
13	RT_ADD_B3	14	DIO3 (Same as PMC2 /TRIG3)
15	GND	16	1553_2A-
17	RT_ADD_B4	18	1553_2A+

Pin	Function	Pin	Function
19	GND	20	DIO8
21	RT_ADD_BP	22	1553_2B-
23	GND	24	1553_2B+
25	DIO9	26	DI2
27	DIO10	28	DI3
29	RT_ADD_AP	30	DI4
31	GND	32	DI5
33	DIO11	34	DI6
35	RT_ADD_A0	36	1553_1B-
37	GND	38	1553_1B+
39	RT_ADD_A1	40	DIO17
41	GND	42	DIO2- (Same as PMC2 TRIG2-)
43	RT_ADD_A2	44	DIO2 (Same as PMC2 TRIG2+)
45	GND	46	DIO1- (Same as PMC2 TRIG1-)
47	RT_ADD_A3	48	DIO1 (Same as PMC2 TRIG1+)
49	GND	50	1553_1A-
51	RT_ADD_A4	52	1553_1A+
53	GND	54	IRIG
55	DIO12	56	DI7
57	DIO13	58	DIO0- (Same as PMC2 TRIG0-)
59	DIO14	60	DIO0 (Same as PMC2 TRIG0+)
61	DIO15	62	DIO18
63	DIO16	64	DIO19

Table 9.2.2: P4 Connector Pin Function Descriptions for ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2

Pin Function	Description
RT_ADD_A[4:0]	External Device 1 RT address 0 - 4
RT_ADD_AP	External Device 1 Parity Bit
RT_ADD_B[4:0}	External Device 2 RT address 0 - 4
RT_ADD_BP	External Device 2 Parity Bit
1553_1A+ & 1553_1A-	1553 Device 1, Bus A
1553_1B+ & 1553_1B-	1553 Device 1, Bus B
1553_2A+ & 1553_2A-	1553 Device 2, Bus A
1553_2B+ & 1553_2B-	1553 Device 2, Bus B
IRIG	IRIG B Time Input
DI[7:0]	Discrete Inputs 0 - 7
DIO[19:0]	Discrete Input/Outputs 0 - 19

9.2.5 Part Numbers and Ordering Instructions

Figure 9.2.8 illustrates the cable assembly provided with the PMC2 and gives part numbers and ordering instructions for the external connectors. If you are planning to use external signals, you need to make the proper connections to the J1 front panel connector, mezzanine P4 connector, or the J5 connector on the Auxiliary cable shown in Figure 9.2.8. See Table 9.2.3 for a description of the pinouts on J5 connector. The remaining topics in this section describe the connections for IRIG signals, external triggers, and external clocks.

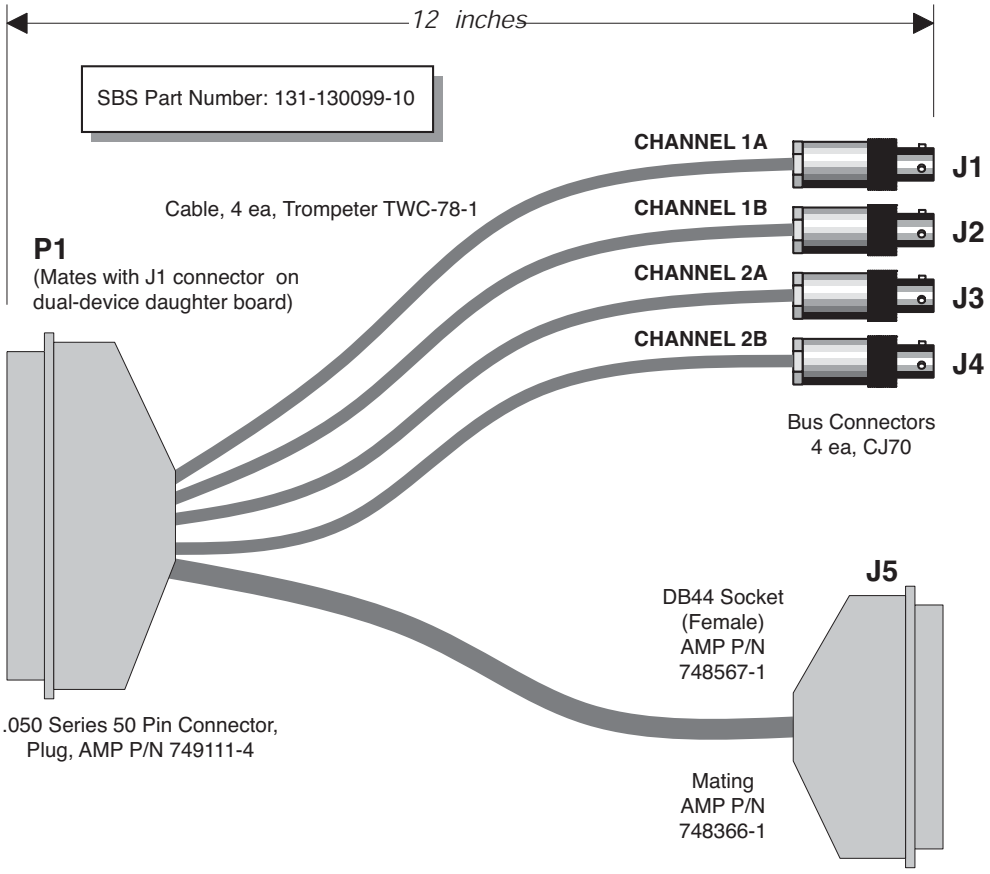


Figure 9.2.8: PMC2 Auxiliary Cable Assembly

Table 9.2.3: J5 Connector Pinouts on Auxiliary Cable Assembly

Pin	Function	Pin	Function
1	RT Address A0	23	–
2	RT Address A1	24	Trigger 7
3	RT Address A2	25	Trigger 6
4	RT Address A3	26	Trigger 5 or External Clock Enable
5	RT Address A4	27	Trigger 4
6	RT Address A Parity	28	Trigger 3
7	Ground	29	Trigger 1+
8	RT Address A Select	30	Trigger 1–
9	GP A0	31	RT Address B0
10	–	32	RT Address B1
11	–	33	RT Address B2
12	–	34	RT Address B3
13	Trigger 2+	35	RT Address B4
14	Trigger 2–	36	RT Address B Parity
15	–	37	Ground
16	–	38	RT Address B Select
17	IRIG Data Input	39	GP B0
18	Ground	40	Ground
19	–	41	–
20	–	42	–
21	–	43	Trigger 0+ or External Clock+
22	–	44	Trigger 0– or External Clock–

9.3 Operational Specifications

The information in this section presents the operational specifications for the PMC2. This section contains the following topics:

- Temperature
- PMC2CC-2-D Specifications
- MTBF
- Hardware Reset

9.3.1 Temperature

Table 9.3.1 lists the PMC2 operating and storage temperature specifications.

Table 9.3.1: PMC2 Temperature Specifications

Model	Operating	Storage
ABI/ASF-PMC2-1,2 (Standard, Commercial Air Cooled)	0° to +70° Celsius ≤95% humidity	–65° to +150° Celsius ≤95% humidity
ABI/ASF-PMC2XT-1,2 (Rugged, Forced Air)	-40° to +85° Celsius ≤95% humidity	–65° to +150° Celsius ≤95% humidity
ABI/ASF-PMC2CC-1,2 (Rugged, Conductive Cooled)	-40° to +85° Celsius ≤95% humidity	–65° to +150° Celsius ≤95% humidity

9.3.2 PMC2CC-2-D Specifications

Table 9.3.2 lists the PMC2CC-2-D specifications.

Table 9.3.2: PMC2CC-2-D Specifications

Function	Description
IRIG:	Input Impedance: approximately 10K ohms
	Coupling: AC
	Input Signal Characteristics: IRIG B
	Input voltage range and mark to space ratio range: With a 3:1 (Data:Clk) input ratio, the Data input range is 0.55 Vpp to 7.5 Vpp With a 6:1 (Data:Clk) input ratio, the Data input range is 1.30 Vpp to 9.3 Vpp
RT_ADD Inputs:	Signals internally pulled up via 3.9K to +5V

Function	Description	
DI[7:0] Inputs:	Vil: -0.5V min, 0.8V max Vih: 1.7V min, 5.75V max Iil: 1.28 ma. @ Vil = 0V, Host Vcc = 5.0 V	
DIO[19:0]	Signals internally pulled up via 2.2K to +5V	
DIO[2:0] Single Ended	As Input	Vil: -0.5V min, 0.8V max Vih: 1.7V min, 5.75V max Iil: 2.27 ma. @ Vil = 0V, Host Vcc = 5.0 V
	As Output	Vol ~0.3V @ Iol = -100 ma. Iol continuous max = 200 ma @ 25C
DIO[2:0] Differential	Uses Maxim MAX1484 Transceiver	
	As Input:	Receiver Diff. Threshold -7V < Vcm < 12V Iih: 0.125 ma. @ Vin = 12V Iil: -0.1 ma. @ Vin = -7V
	As Output	Vol 2.0V min. into 50 ohm load (RS-422) Vol 1.5V min. into 27 ohm load (RS-485)

Note: DIO[2:0]- has an input resistance of 2.8K ohm and is biased at 1.79 V. This allows the positive inputs to be used as a single ended input.

9.3.3 MTBF

At 25˚ C, the mean time between failures is 785,986 hours for the PMC2-2, 997,009 hours for the PMC2-1, and 1,192,311 hours for the ASF-PMC2CC-1&2-D and ESF-PMC2CC-1&2. We calculated the MTBF using the MIL-HDBK-217F, Parts Count Method - Ground Benign Environment.

9.3.4 Hardware Reset

If the hardware reset button on the host system is pressed, the CSR resets and firmware execution halts. At this point, memory above 003Fh is still intact and may be accessed. After a hardware reset, you must restart the firmware and the memory above 003Fh clears at this time.

Cross Reference: See Section 10.2 for a description of the CSR.

10: Card Information



Note: This manual uses the terms channel and device interchangeably.

This chapter provides card information for the ABI/ASF-PMC2 cards on memory organization, hardware control registers, and downloading/starting the firmware. This chapter covers the following topics:

- Memory Organization
- Control/Status Register
- Downloading the Firmware File
- Starting the PMC2 Firmware
- PCI-Based Big Endian Access and Data Word Swapping

10.1 Memory Organization

This section describes the memory organization for the ABI/ASF-PMC2 card. It covers the following topics:

- Single-Channel Boards
- Dual-Channel Boards

10.1.1 Single-Channel Boards

The PMC2 cards reserve one megabyte of memory as shown in Figure 10.1.1.

10.1.2 Dual-Channel Boards

The memory organization for dual-channel boards appears identical to the memory organization for two *single channel* boards. The Software Control Registers and user-defined firmware structures for BC, RT, monitoring, and interrupt handling locations are between word offsets 00000h and 3FFFFh. The Control/Status Register (CSR) location is at 40000h with the region from 40001h to 7FFFFh being reserved.

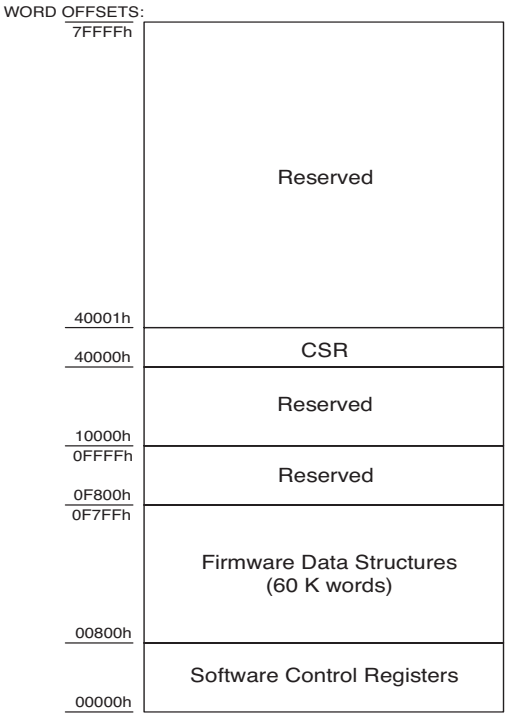


Figure 10.1.1: ABI/ASF-PMC2 Memory Map

10.2 Control/Status Register

The Control/Status Register (CSR) is a key register for proper initialization and operation of the PMC2. Table 10.2.1 describes the bits in the Control/Status Register for a Channel of the card (word address 40000h, byte address 80000h). These registers provide the following module functions to the host:

- Module reset and operation control
- PMC2 interrupt control

Table 10.2.1: PMC2 CSR Bit Descriptions

Bit	Function	Description
0	Run	0 = Firmware Run 1 = Firmware Stop
1	Flash Select	0 = Select RAM 1 = Select Flash
2	Reserved	–
3	PCI Interrupt Enable	0 = PCI Hardware interrupts disabled 1 = PCI Hardware interrupts enabled
4	Altera Reload	0 = No reload? 1 = Reload
5-6	Reserved	–
7	Interrupt Pending (ro)	0 = No interrupt pending 1 = Interrupt is pending
	Interrupt Clear (wo)	0 = No function 1 = Clears pending interrupt
8-10	LED Ctrl	**Bit 10 / Bit 9 / Bit 8 / LED Status** 0 / 0 / 1 / LED off 0 / 1 / 1 / LED green 1 / 0 / 1 / LED red
11-12	Reserved	–
13	Byte Swap	When Bit 15=1, 0 = Disabled, 1 = Enabled
14	Word Swap	When Bit 15=1, 0 = Disabled, 1 = Enabled
15	Byte/Word Swap Control	0 = Disable setup of Byte/Word Swapping 1 = Enable setup of Byte/Word Swapping

Note: Reserved bits should be set to zero when writing to the CSR.

10.3 Downloading the Firmware File

The PMC2 uses flash memory to store the firmware code. Prior to shipping the board, SBS loads the latest firmware into flash. Normally, the firmware loads automatically when power is applied to the board.

- SBS provides a download file (in ASCII format) that contains the firmware code.
- The data in the download file has a 16-bit word structure. The first sixteen words makeup the file header, containing product and version information.

If you need to load new firmware into flash memory, follow the procedure below.

1. Write a value of 0003h to the CSR then wait for approximately 10 milliseconds.

   This resets the DSP and selects host flash access.

2. Read the first 16 words from the firmware file. The first 16 words are header information and can be skipped.

3. Read the next 2 words from the firmware file, convert the two 16 bit words to a 32 bit word count (N). The first word is the 16 MSB and the second word is the 16 LSB. The 32 bit word count are the number of words (N) in the DSP section of the firmware file.

4. Read the next N number of words from the firmware file AND write each word to flash starting at offset 00000h. This is the DSP portion of the firmware file and should be written to flash sector 0. Each sector in flash is 8000h words long and sector 0 must be cleared before the DSP portion of the firmware file can be written.



Cross Reference: For more information on clearing a sector in flash, see "Clearing a PMC2 Flash Sector" on page 73.

5. Write N number of words from the firmware file to flash.



Cross Reference: For more information on writing to flash memory, see "Writing a Word to PMC2 Flash Memory" on page 74.

6. Read the next 2 words from the firmware file, convert the two 16 bit words to a 32 bit word count (M). The first word is the 16 MSB and the second word is the 16 LSB. The 32 bit word count are the number of words (M) in the Altera section of the firmware file.

7. Read the next M number of words from the firmware file writing each word to flash starting at offset 08000h. This is the Altera portion of the firmware file. Each sector in flash is 8000h words long and sectors 1-3 must be cleared before the Altera portion of the firmware can be file written.



Cross Reference: For more information on clearing a sector in flash, see "Clearing a PMC2 Flash Sector" on page 73.

8. Write N number of words from the firmware file to flash.



Cross Reference: For more information on writing to flash memory, see "Writing a Word to PMC2 Flash Memory" on page 74.

9. Now, clear Bit 1 in the CSR to access DSP memory once again by writing a 1 to the CSR.

10. Verify Bit 1 has been cleared by reading the CSR.

11. Once the new firmware is loaded into flash, there are two options for restarting the PMC2. The table below describes each option.

Option 1	Option 2
Power off the computer and restart it. The new Altera will be loaded when the computer is powered-up.	Use the procedure for "Starting the PMC2 Firmware" on page 75 and use the setting bit 4 in the CSR in Step 3.

10.3.1 Clearing a PMC2 Flash Sector

To clear a sector in flash, follow the procedure below.

1. Determine the Sector Offset.

- Flash sector 0 is for the DSP portion of the download/firmware file.
- Flash sectors 1-3 are for the Altera portion of the download/firmware file.
- The Sector Offset is computed by multiplying the desired sector (0-3) by 8000h as follows:
  - For sector 0, 0 * 8000h = 0000h
  - For sector 1, 1 * 8000h = 8000h
  - For sector 2, 2 * 8000h = 10000h
  - For sector 3, 3 * 8000h = 18000h

2. To reset flash:

  - Write F0h to offset 0h.
  - Write AAh to offset 555h.
  - Write 55h to offset 2AAh.
  - Write 80h to offset 555h.
  - Write AAh to offset 555h.
  - Write 55h to offset 2AAh.
  - Write 30h to the sector offset (computed above).

3. To verify the sector clear has completed:

   - ➢ Read the sector offset location.
   - ➢ If bit 7 is set, then the clear operation succeeded.
   - ➢ If bit 7 is not set, then check bit 5.
   - ➢ If bit 5 is not set, then repeat this Step.
   - ➢ If bit 5 is set, then read the sector offset location.
   - ➢ If bit 7 is set, then the clear operation succeeded.
   - ➢ If bit 7 is not set, then the clear operation failed.



Note: If the clear operation failed, please contact SBS Technical Support at 1-877-TECHSBS.

4. Finally, verify that the sector was indeed cleared (all locations set to FFFFh) by reading each location using the sector offset computed above and accessing locations 0000h-7FFFh within the sector.

10.3.2 Writing a Word to PMC2 Flash Memory

To write data to flash memory, follow the procedure below.

1. Write F0h to offset 0h to reset flash.

   - ➢ Write AAh to 555h.
   - ➢ Write 55h to 2AAh.
   - ➢ Write A0h to 555h.

2. Write the word to flash using the Sector Offset computed in the previous topic, Clearing a PMC2 Flash Sector and the write location (0000h-7FFFh) within the sector.

```
write offset = sector offset + write location
```



Note: If the clear operation failed, please contact SBS Technical Support at 1-877-TECHSBS.

3. To verify the write operation has completed:

   - ➢ Read the sector offset location.
   - ➢ If bit 7 is equal to bit 7 of the word written, then the clear operation succeeded.
   - ➢ If bit 7 is not equal to bit 7 of the word written, then check bit 5.
   - ➢ If bit 5 is not set, then repeat this Step.
   - ➢ If bit 5 is set, then read the sector offset location.
   - ➢ If bit 7 is equal to bit 7 of the word written, then the clear operation succeeded.
   - ➢ If bit 7 is not equal to bit 7 of the word written, then the clear operation failed.



Note: If the clear operation failed, please contact SBS Technical Support at 1-877-TECHSBS.

4. Finally, verify that the word value was indeed written to the write offset location by reading write offset using the sector offset computed above and the write location within the sector.

5. Repeat steps 1-4 above to write additional words to flash incrementing the write location within the sector for each write.

10.4 Starting the PMC2 Firmware

To start the firmware, follow the procedure below.

1. Write a value of 00001h to the CSR

   This resets the DSP.

2. Write a value of 0FFFFh to offset 0003Bh. Offset 0003Bh is used as a handshake between the DSP code and the host application.

3. Write a value of 00000h to the CSR. If bit 4 is set in this step, the Altera is reloaded. Note that the plug-and-play/PCI information will be lost if this is done.



Note: The host must restore this information before RAM, flash, or the CSR can be accessed. For additional information, please contact SBS Technical Support.

   The DSP starts and downloads the DSP section of the firmware from flash.

4. Read offset 0003Bh and wait for the value to equal 00000h.

5. The firmware is initialized and waiting for a BIT control register (Offset 0003Ah) command.

6. To start the application code, write a 00000Dh to the BIT control register.

10.5 PCI-Based Big Endian Access and Data Word Swapping

This section presents the PCI big endian access as used with the PMC2 cards. It covers the following topics:

- Background on Big and Little Endian Data Buses
- PMC2 Card Memory Architecture
- Data Word Swapping (PMC2 only)
- Big Endian Accesses
- Data Word Swapping and Big Endian

10.5.1 Background on Big and Little Endian Data Buses

You can define the PMC2 bus as a Little Endian bus. On a Little Endian bus, data is long-word aligned to the lowermost byte lane. Byte 0 (address 0) appears in AD[07:00], Byte 1 appears in AD[15:08], Byte 2 appears in AD[23:16], Byte 3 appears in AD[31:24]. The example below shows Little Endian byte ordering using the data pattern 87654321:

Byte	3	2	1	0
Data	87	65	43	21

The Big Endian bus presents the bytes in reverse order. The same data pattern presented in Big Endian byte ordering would look as follows:

Byte	0	1	2	3
Data	21	43	65	87

10.5.2 PMC2 Card Memory Architecture

The PMC2 cards utilize a 16-bit DSP processor for 1553 processing. Since a 16-bit DSP is used, the memory architecture also uses a 16-bit base. To facilitate getting data to and from the host at the fastest rate possible, the PMC2 card, however, provides a 32-bit (long-word) interface between the host and the card's memory. The transition between 16-bit memory and the 32-bit PCI interface raises issues in data byte and word ordering. Table 10.5.1 shows how the data is stored in the card's memory. When the host performs word-based accesses, it reads data in this format.

Table 10.5.1: PMC2 Memory Data Structure

Word Address	Data
0	4321
1	8765
2	CBA9
3	0FED

When the host performs long-word based accesses, it reads data as shown in Table 10.5.2.

Table 10.5.2: Little Endian Long-Word Access

Long-Word Address	Data
0	87654321
1	0FEDCBA9

While this is desirable in Intel and other Little Endian buses, it may not be desirable in Big Endian systems. In some types of Big Endian systems, it may be desirable to read the data in Big Endian fashion with the bytes reversed as shown in Table 10.5.3.

Table 10.5.3: Big Endian Long-Word Access

Long-Word Address	Data
0	21436587
1	A9CBED0F

In word-based Big Endian systems, it may even be desirable to read data with the words swapped as shown in Table 10.5.4.

Table 10.5.4: Word Swapped Long-Word Access

Long-Word Address	Data
0	43218765
1	CBA90FED

10.5.3 Data Word Swapping (PMC2 only)

Memory data word swapping, added as a new hardware feature to the PMC2 card, has the capability to be enabled on the fly by setting bit 14 of the CSR to active. When CSR bit 14 sets, memory accesses to the PMC card automatically swap even and odd data words. When CSR bit 14 clears, long-word memory accesses behave normally.

This data word swapping feature affects only memory accesses. Accesses to PCI configuration registers are unaffected. The CSR is memory mapped, thus it is also affected. The byte/word swapping mode is reset to the default setting when the firmware halts (Run bit =1).



Note: Data word swapping is currently available only on PMC2 cards.

10.5.4 Big Endian Accesses

The PMC2 also provides Big Endian accesses with the bytes accessed in reverse order. Setting bits in the CSR activates this capability. When activated, all accesses to the PMC2 card memory are Big Endian.



Note: Big Endian access is standard on PMC2 cards.



Interesting Side-Note: The term "endian" is derived from a passage in Jonathan Swift's *Gulliver's Travels*.

10.5.5 Data Word Swapping and Big Endian

You can enable both Data Word Swapping and Big Endian features simultaneously on PMC2 cards.

11: Using the INTEGRITY API



Note: The INTEGRITY IODevice API calls that are supported by the 1553-PMC2CC driver are described in this chapter.

The 1553-PMC2CC INTEGRITY driver doesn't have a specialized Application Programing Interface (API). Instead, it facilitates various INTEGRITY IODevice API calls. The following sections describe the IODevice API calls that are supported by the 1553-PMC2CC INTEGRITY driver and can be used in a user application. See the following list:

- `RequestResource()`
- `CloseIODevice()`
- `ReadIODeviceRegister()`

11.1 RequestResource

Description The `RequestResource` call opens a given 1553 device. A zero-based index identifies each device. Concatenating strings `SBS1553_NAME_ROOT` and `<device index>` create the device full name. For example, `SBS1553_NAME_ROOT "1"` creates a name for the second 1553 device. The *m1553.h* file defines `SBS1553_NAME_ROOT`.

Syntax

```
Error  RequestResource(Object      *TheObject,
                       const char  *name,
                       const char  *password);
```

Parameters The `TheObject` parameter is the 1553 device's IODevice object to be created.

The `name` parameter is the name of the 1553 device registered with the Resource Manager.

The `password` parameter uses the default Resource Manager password `"!password"`.

Comments None

Example

```
#include "m1553.h"

Error e;
IODevice ioDev;

// open first device (0)
e = RequestResource((Object*)&ioDev, SBS1553_NAME_ROOT "0", "!systempass-
word);
```

See Also None

11.2 CloseIODevice

Description `CloseIODevice` closes the given device. The device must be successfully opened by a previous call to `RequestResource`.

Syntax

```
Error  CloseIODevice(IODevice  TheIODevice)
```

Parameters The `TheIODevice` parameter is the 1553 device's IODevice object created by `RequestResource`.

Comments None

Example

```
#include "m1553.h"

Error e;
IODevice ioDev;

// open device here
...

// done, close the device
e = CloseIODevice(ioDev);
```

See Also `RequestResource()`

11.3 ReadIODeviceRegister

Description `ReadIODeviceRegister` has only one register numbers defined as follows:

- `M1553_BASE_ADDR_REG` - This reads the PCI memory base address assigned to the device's onboard registers

The *m1553.h* file defines the status number.

Syntax

```
Error  ReadIODeviceRegister(IODevice   TheIODevice,
                            Value      RegisterNumber,
                            Value      *RegisterValue)
```

Parameters The `TheIODevice` parameter is the 1553 device's IODevice object created by `RequestResource`.

The `RegisterNumber` parameter is the register number to be read from.

The `RegisterValue` parameter is the register value read.

Comments None

Example

```
#include "m1553.h"

IODevice ioDev;
Error e;
Value bar;

// open device here
...

// get BAR register from PCI configuration space
e = ReadIODeviceRegister(ioDev, M1553_BASE_ADDR_REG, &bar);
```

See Also `RequestResource()`

A: Revisions

The table in this appendix gives a brief summary of any technical revisions made to this manual. When reading this manual online, you can jump to the first citation of a revision by clicking the links in blue.



Note: Only technical revisions appear in the table. Most even numbered pages contain a date stamp in the footer. If the footer date is more recent then the latest revision date given in the table, then the newest revision of this manual contains only *non-technical* revisions.

Revision Number	Revision Date	Description
2	12 Jul 2004	Added this Revisions chapter. This revision also incorporates all previous revisions.
2.01	17 Aug 2004	Revised Figure 9.2.1 (Connector locations and names were correct, previous version was missing front panel.)
2.02	20 Oct 2004	Added Uninstalling SBS Drivers In Windows 2000 to Chapter 5. Changed the title of the manual from "*Getting Started with the ABI/ASF/ESF-PMC2 Family*" to the new name "Getting Started with the ABI/ASF/ESF- PMC2 Series" beginning at the Front Cover.

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