*i*SPAN™ 4538 PMC® T1/E1/J1 Communications Controller Board Installation and Maintenance Manual

Document No. UG04538-000-A2

Release Date: July 2002

NOTE

See Appendix C for Regulatory Statements/Conditions that affect the operation of this product.
The CE Declaration of Conformity can be found at http://www.iphase.com.

Copyright Notice

Printed in the United States of America, 2002.

Interphase Corporation
13800 Senlac
Dallas, Texas 75234

Phone: (214) 654-5000
Fax: (214) 654-5506

Disclaimer

Information in this manual supersedes any preliminary specifications, preliminary data sheets, and prior versions of this manual. While every effort has been made to ensure the accuracy of this manual, Interphase Corporation assumes no liability resulting from omissions, or from the use of information obtained from this manual. Interphase Corporation reserves the right to revise this manual without obligation to notify any person of such revision. Information available after the printing of this manual will be in one or more Read Me First documents. Each product shipment includes all current Read Me First documents. All current Read Me First documents are also available on our web site.

THIS MANUAL IS PROVIDED "AS IS." INTERPHASE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL INTERPHASE BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Trademark Acknowledgments

Interphase® and the Interphase logo are registered trademarks, (*i*)chip™, SynWatch™, FibreView™, ENTIA™, PowerSAN™, SlotOptomizer™, *i*WARE™, *i*NAV™, and *i*SPAN™ are trademarks of Interphase Corporation.

All other trademarks are the property of their respective owners.

Assistance

Product Purchased from Reseller

Contact the reseller or distributor if

- You need ordering, service or any technical assistance.
- You received a damaged, incomplete or incorrect product.

Product Purchased Directly from Interphase Corporation

Contact Interphase Corporation directly for assistance with this, or any other Interphase Corporation product. Please have your purchase order and serial numbers ready.

Customer Support

United States:	Telephone:	(214) 654-5666
	Fax:	(214) 654-5506
	E-Mail:	intouch@iphase.com
Europe:	Telephone:	+ 33 (0) 1 41 15 44 00
	Fax:	+ 33 (0) 1 41 15 12 13

World Wide Web

http://www.iphase.com

END-USER LICENSE AGREEMENT
FOR INTERPHASE CORPORATION SOFTWARE

IMPORTANT NOTICE TO USER–READ CAREFULLY

THIS END-USER LICENSE AGREEMENT FOR INTERPHASE CORPORATION SOFTWARE ("AGREEMENT") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR SINGLE ENTITY) AND INTERPHASE CORPORATION FOR THE SOFTWARE PRODUCTS ENCLOSED HEREIN WHICH INCLUDES COMPUTER SOFTWARE AND PRINTED MATERIALS ("SOFTWARE"). BY INSTALLING, COPYING, OR OTHERWISE USING THE ENCLOSED SOFTWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY RETURN, WITHIN THIRTY DAYS, THE UNUSED SOFTWARE TO THE PLACE FROM WHICH YOU OBTAINED IT FOR A FULL REFUND.

The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold.

Grant of License: You are granted a personal license to install and use the Software on a single computer solely for internal use and to make one copy of the Software in machine readable form solely for backup purposes.

Restrictions on Use: You may not reverse engineer, decompile, or disassemble the Software. You may not distribute copies of the Software to others or electronically transfer the Software from one computer to another over a network. You may not use the Software from multiple locations of a multi-user or networked system at any time. You may not use this software on any product for which it was not intended. You may not use this software on any non-Interphase product. LICENSEE MAY NOT RENT, LEASE, LOAN, OR RESELL THE SOFTWARE OR ANY PART THEREOF.

Ownership of Software: Interphase or its vendors retain all title to the Software, and all copies thereof, and no title to the Software, or any intellectual property in the Software, is being transferred.

Software Transfer: You may permanently transfer all of your rights under this Agreement, provided you retain no copies, you transfer all the Software, and the recipient agrees to the terms of this Agreement.

Limited Warranty: Interphase Corporation ("Seller") warrants that (i) the hardware provided to Buyer ("Products") shall, at the F.O.B. point, be free from defects in materials and workmanship for a period of one (1) year from the date of shipment to Buyer; (ii) the software and/or firmware associated with or embedded in the Products shall comply with the applicable specifications for a period of six (6) months from the date of shipment to Buyer; and (iii) its services will, when performed, be of good quality. Defective and nonconforming Products and software must be held for Seller's inspection and returned at Seller's request, freight prepaid, to the original F.O.B. point.

Upon Buyer's submission of a claim in accordance with Seller's Return and Repair Policy, Seller will, at its option either (i) repair or replace the nonconforming Product; (ii) correct or replace the software/firmware; (iii) rework the nonconforming services; or (iv) refund an equitable portion of the purchase price attributable to such nonconforming Products, software, or services. Seller shall not be liable for the cost of removal or installation of products or any unauthorized warranty work, nor shall Seller be responsible for any transportation costs, unless expressly authorized in writing by Seller. This warranty does not cover damage to the Product resulting from accident, disaster, misuse, negligence, improper maintenance, or modification or repair of the Product other than by Seller. Any Products or software replaced by Seller will become the property of Seller.

REMEDIES AND EXCLUSIONS. THE SOLE LIABILITY OF SELLER AND BUYER'S SOLE REMEDY FOR BREACH OF THESE WARRANTIES SHALL BE LIMITED TO REPAIR OR REPLACEMENT OF THE PRODUCTS OR CORRECTION OF THAT PART OF THE SOFTWARE, WHICH FAILS TO CONFORM TO THESE WARRANTIES. EXCEPT AS EXPRESSLY STATED HEREIN, AND EXCEPT AS TO TITLE, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, IN CONNECTION WITH OR ARISING OUT OF ANY PRODUCT OR SOFTWARE PROVIDED TO BUYER.

IN NO EVENT SHALL SELLER HAVE ANY LIABILITY FOR INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF THESE WARRANTIES, INCLUDING BUT NOT LIMITED TO LOSS OF ANTICIPATED PROFITS, LOSS OF DATA, USE OR GOODWILL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. (IC-199, 1/97)

Limitation of Liability: NEITHER INTERPHASE NOR ITS LICENSORS SHALL BE LIABLE FOR ANY GENERAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR OTHER DAMAGES ARISING OUT OF THIS AGREEMENT EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Confidentiality: The Software is copyrighted and contains proprietary and confidential trade secret information of Interphase and its vendors. Licensee agrees to maintain the Software in confidence and not to disclose the Software to any third party without the express written consent of Interphase. Licensee further agrees to take all reasonable precautions to prevent access to the Software by unauthorized persons.

Termination: Without prejudice to any other rights, Interphase may terminate this Agreement if you fail to comply with any term or condition of the Agreement. In such event you must destroy the Software together with all copies, updates, or modifications thereof.

Export: You agree to comply with all export and re-export restrictions and regulations of the U.S. Department of Commerce or other applicable U.S. agency. You must not transfer the Software to a prohibited country or otherwise violate any such restrictions or regulations.

U.S. Government Restricted Rights: Use, duplication, or disclosure of the Software to or by the U.S. Government is subject to restrictions as set forth in the applicable U.S. federal procurement regulations covering commercial/restricted rights software. You are responsible for complying with the notice requirements contained in such regulations.

General: You acknowledge that you have read and understand this Agreement, and by installing and using the Software you agree to be bound by the terms and conditions herein. You further agree that this is the complete and exclusive Agreement between Interphase and yourself. No variation of the terms of this Agreement or any different terms will be enforceable against Interphase unless agreed to in writing by Interphase and yourself. The validity of this Agreement and the rights, obligations, and relations of the parties hereunder shall be determined under the substantive laws of the State of Texas. If any provision of this Agreement is held invalid, illegal, or unenforceable, the remaining provisions shall in no way be affected or impaired thereby. All rights in the Software not specifically granted in this Agreement are reserved by Interphase.

Contents

APPENDIX A VxWorks® Maintenance Tools

APPENDIX B Solaris® Maintenance Tools

APPENDIX C Regulatory Statements

List of Figures

List of Tables

Using This Manual

Purpose

The 4538 Installation and Maintenance Manual describes the following:

- 4538 communications controller Overview
- 4538 communications controller Installation
- Connection to Network Line
- Commands used to maintain the card

Icon Conventions

Icons draw your attention to especially important information:

NOTE

The Note icon indicates important points of interest related to the current subject.

CAUTION

The Caution icon brings to your attention those items or steps that, if not properly followed, could cause problems in your machine's configuration or operating system.

WARNING

The Warning icon alerts you to steps or procedures that could be hazardous to your health, cause permanent damage to the equipment, or impose unpredictable results on the surrounding environment.

Text Conventions

The following conventions are used in this manual. Computer-generated text is shown in typewriter font. Examples of computer-generated text are: program output (such as the screen display during the software installation procedure), commands, directory names, file names, variables, prompts, and sections of program code.

```
Computer-generated text example
```

Commands to be entered by the user are printed in **`bold Courier`** type. For example:

```
cd /usr/tmp
```

Pressing the return key (↵ **`Return`**) at the end of the command line entry is assumed, when not explicitly shown. For example:

```
/bin/su
```

is the same as:

```
/bin/su  ↵ Return
```

Required user input, when mixed with program output, is printed in **`bold Courier`** type. References to UNIX programs and manual page entries follow the standard UNIX conventions.

When a user command, system prompt, or system response is too long to fit on a single line, it will be shown as

```
Do you want the new kernel moved into
\vmunix?[y]
```

with a backslash at either the beginning of the continued line or at the end of the previous line.

Documentation Updates

The latest documentation (in Adobe® Acrobat® pdf) for our current products are available on our WWW site. Interphase recommends our customers visit the web site to verify that they have the latest version of the documentation.

1. Access the following web page:

   http://www.iphase.com

2. Move the mouse (or other pointer) and click on the `Products` option. A menu will appear on the left side with **Telecom Solutions**, **Enterprise Solutions**, and **Services** options. Choose the appropriate menu item (such as **PowerSAN Fibre Channel HBAs**).

3. A new web page with a list of the currently offered products will appear. Choose your product by clicking on the product number (i.e. 4532, 5540, 4575, etc.).

4. The Product Description page appears for the product selected. At the left side of the page is a list showing additional information web pages for that product. Choose the **User Guides** item.

5. A new web page appears with a list of the latest released user guides available for the product. Click on the document you require.

Driver Updates

Contact our Technical Support Department at swlib@iphase.com to determine if updated drivers are available for your product.

When contacting technical support, please be sure to provide your name, company name and address, phone number, product name, driver version (if applicable), OS and version (if applicable) and serial number. Providing this information will help speed up our response.

Introduction 1

Overview

The Interphase 4538 PMC T1/E1/J1 Communications Controller is a network interface PCI Mezzanine Card (PMC) equipped with four software-selectable T1/E1/J1 interfaces (two are provided on the front panel). The 4538 board is intended for 2G and 3G wireless networks, Internet access, and Advanced Intelligent Network (AIN) applications.

The 4538 is powered by the Motorola® PowerQUICC II™ MPC8260 32-bit RISC processor running at 200 MHz, combined with a 32-bit, 33 MHz PCI PMC interface.

Figure 1-1 shows the 4538 board.

Figure 1-1. 4538 PMC T1/E1/J1 Communications Controller, Front Access

Hardware Overview

The 4538 Communications Controller contains the following features:

PCI Interface

- 32-bit and 33 MHz (designed for 66 MHz, but currently limited to 57 MHz by the PCI bridge)
- PCI 2.2 master/target bus interface with I²O messaging unit and four linked-list DMAs
- High transfer performance: 64-bit DMA exchanges and 64-bit 60x bus

Processing Power

- PowerQUICC II MPC8260 32-bit RISC processor with core @ 200 MHz and CPM @ 133 MHz

- 64-bit pipelined local 60x type bus @ 66 MHz
- 64 MBytes 64-bit SDRAM memory
- 4 MBytes downloadable 8-bit FLASH EEPROM for boot, Power-On Self Tests (POST), and operational code storage

T1/E1/J1 Line Framing

- One Infineon QuadFALC™ PEB22554 E1/T1/J1 framer
- Two line interfaces accessible from the front panel (software configurable as short or long haul)
- Capability to support four line interfaces (rear access)

Telecom Clock Management

- The line can be configured in LT (clock slave) or NT (clock master) mode
- Synchronization sources are E1/T1/J1 lines or external TDM rhythm on PMC connector P4
- Internal clock availability

TDM Busses

- In Direct mode, TDM busses connect the MPC8260 to the QuadFALC™
- In Switched Mode, TDM busses connect the MPC8260 and the QuadFALC to the TDM busses on P4.
- Pass-through mode available

Other Features

- One Fast-Ethernet interface on the front panel for remote boot or LAN capability with support of 10BaseT/100BaseT auto-negotiation and parallel detection
- One RS232 TTY connector on the front panel for connection to a console for test and development purposes and/or management
- Embedded Power-On Self Tests

4538 Equipment Options :

Available 4538 equipment options are:

- 4538-008: E1/T1/J1 PMC Communications Controller, MPC8260 processor, 2 Port Front Access, Ethernet, TTY, 200 MHz, 64 MB SDRAM
- 4538-009: Developers version of 4538-008 with a JTAG debug connector
- 4538-012: E1/T1/J1 PMC Communications Controller, MPC8260 processor, 2 Port Front Access, Ethernet, TTY, 133 MHz, 32 MB SDRAM
- 4538-013: Developers version of 4538-012 with a JTAG debug connector

Installing the Card 2

Overview

This chapter describes the procedure for physically installing the 4538 card.

Observe all notes, cautions, and warnings in this chapter. Also, refer to the users manuals that came with the card which will accommodate the 4538 and with your system for additional detailed installing instructions.

The 4538 communications controller is designed to be installed on a standard carrier card PCI Mezzanine Card (PMC) site.

NOTE

The 4538 is a universal voltage PMC card and can therefore be used on a mother board that supports 5V or 3.3V PCI signalling.

The 4538 exists in “Front Access” configuration or in “Rear Access” configuration.

In “Front Access” configuration, the 4538 provides direct connections to two T1/E1/J1 lines through two RJ48C connectors placed on its front panel.

In “Rear Access” configuration, the 4538 can support four T1/E1/J1 line, but the line interfaces and the line connectors are not on board, so the 4538 must be used in conjunction with (an) other card(s).

In order to provide the complete E1/T1/J1 functionality in a CompactPCI chassis with the 4538 “Rear Access”, Interphase has developed the 6435 CompactPCI E1/T1/J1 Rear Transition Module (RTM). The complete E1/T1/J1 functionality is provided when the 4538 is placed on a Motorola CPV8540 or Motorola MCPN750 CompactPCI carrier card and combined with the Interphase 6435 RTM.

CAUTION

The 6435 CompactPCI Rear Transition Module is designed to be used with the Motorola CPV8540 CompactPCI carrier card or the Motorola MCPN750 non-system CompactPCI processor board. Using other CompactPCI carrier cards along with the 6435 will probably not work and is not recommended.

As an example of installation, this chapter will describe the installation of the 4538 on a Motorola CPV8540 carrier card and the installation of this carrier card with the 6435 rear transition module into a CompactPCI chassis.

The 4538 “Rear Access” may also be used on other customer-specific environments, based on CompactPCI or not. In this case, other customer-specific information may be needed.

This chapter will describe the installation of a 4538 “Front Access” and the installation of a 4538 “Rear Access”.

The normal tools required are a grounding strap and a screwdriver (type as required for your system). Refer to the instructions that came with your system for any additional tools required.

NOTE

The 4538 card is compliant with the PCI Local Bus Revision 2.2 specifications. As a result, once physically installed into the PCI system, it can be automatically detected and configured by this system during its standard PCI enumeration process. See your host system’s documentation for PCI Device Configuration.

Inspecting the Card

Before installing the card, visually inspect it for any damage that might have occurred during shipment from the factory.

CAUTION

The card is packed in an antistatic bag to protect it during shipment. Keep the card in its protective antistatic bag until you are ready to install it in the host computer. To prevent damage to the card due to electrostatic discharge, wear a grounding strap and handle the card only by its edges. Do not touch its components or any metal parts other than the faceplate.

1. Open the shipping container and carefully remove its contents.
2. Inspect each item for damage. If you find any omissions or damage, contact your supplier and the carrier (for example, UPS or Federal Express) that delivered the package.

Installing the 4538 “Front Access”

The installation procedure for the 4538 “Front Access” describes only the the card mounting onto it’s carrier card. Refer to the carrier card user manual for the rest of the installation into the system.

The 4538 mounts into the carrier board as shown in Figure 2-1 on page 5. With a grounding strap connected to your wrist or ankle, do the following to install the card:

a. Hold the 4538 card at an angle. Insert it through the back of the carrier board's faceplate, and align the dual mating connectors on the carrier with the dual connectors (P1 and P2) on the daughtercard.

b. Carefully press the card into place. Alignment is facilitated via a standoff post, and sometimes also a 5V or 3.3V keying pin on the carrier and corresponding alignment holes on the daughtercard.

c. Fasten the cards together with the four screws

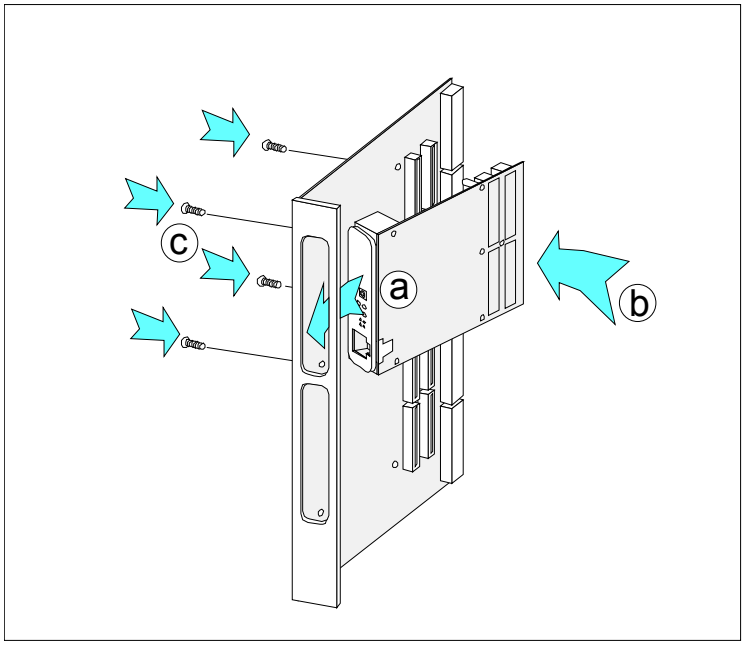


Figure 2-1. Installing the 4538 on the CompactPCI Carrier Card

Installing the 4538 "Rear Access"

As an example of 4538 "Rear Access" installation, this chapter will describe the installation of the 4538 on a Motorola CPV8540 carrier card and the installation of this carrier card with the 6435 CompactPCI Rear Transition Module into a CompactPCI chassis.

CAUTION

The Interphase 6435 CompactPCI Rear Transition Module is designed to be used with the Motorola CPV8540 CompactPCI carrier card or the Motorola MCPN750 non-system CompactPCI processor board. Using an other CompactPCI carrier cards along with the 6435 will probably not work and is not recommended.

Figure 2-2 on page 6 describes the main steps of the physical installation:

1. Determine in the CompactPCI chassis which I/O slot will accommodate the cards,
2. Install the 6435 Rear Transition Module,
3. Install the 4538 PMC board onto the CompactPCI carrier card,
4. Install the CompactPCI carrier card into the CompactPCI chassis.

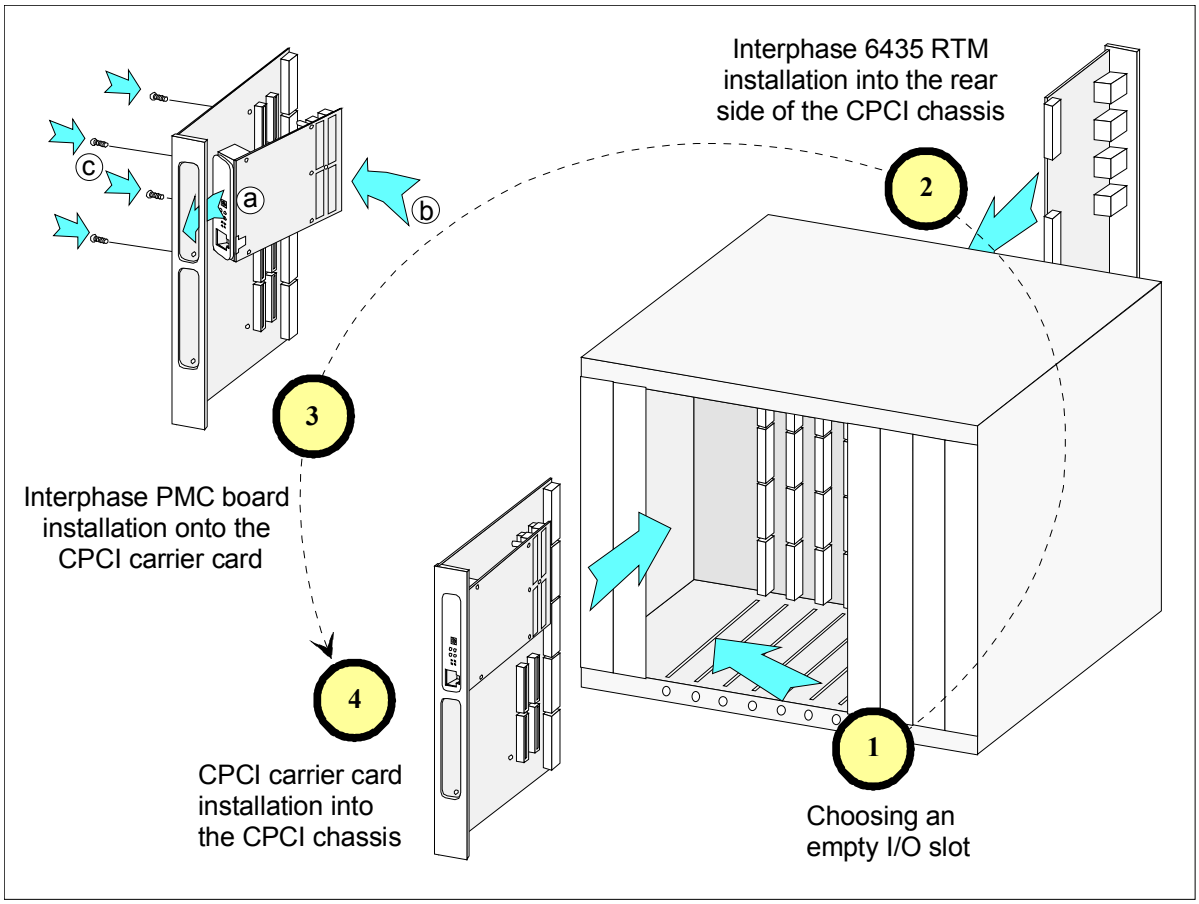


Figure 2-2. Installation Main Steps

Step 1: Choosing a Location in a CompactPCI Chassis

WARNING

Your system operates at voltages that can be lethal. Follow all cautions and warnings in this installation procedure, both to protect yourself and to prevent damage to your system. Use only tools with nonconductive handles, or tools coated with, covered with, or made with nonconductive materials. Nonconductive materials are materials that do not conduct electric current, such as plastic, rubber, and fiberglass.

The 6435 cannot be installed on the rear of a CompactPCI System Slot. A CompactPCI System Slot usually has red plastic guides, and when the chassis provides some markings, there is a triangle.

In the CompactPCI system, choose a non-system location that is free on both sides of the chassis (front and rear). A non-system slot generally has white or black plastic guides, and when the chassis provides some markings, there is a circle.

WARNING

There is a very high risk of damaging your CompactPCI board if it is inserted in a location on which a non-corresponding Rear Transition Module is also inserted. Therefore, it is highly recommended to verify both sides of the chassis.

Step 2: Installing the Rear Transition Module

CAUTION

The corresponding front side CompactPCI slot must be empty (or the CompactPCI chassis power must be off) when installing the Rear Transition Module

With a grounding strap connected to your wrist or ankle, do the following to install the card:

a. Ensure that the corresponding front side slot is empty.

b. Remove the chosen rear slot's expansion cover.

c. Ensure that the 6435 handles are open (pointing away from the top and bottom of the board).

d. Insert the Rear Transition Module into the rear connector slot that will line up directly with the CompactPCI carrier card, until the handles meet the rack's guide bars

WARNING

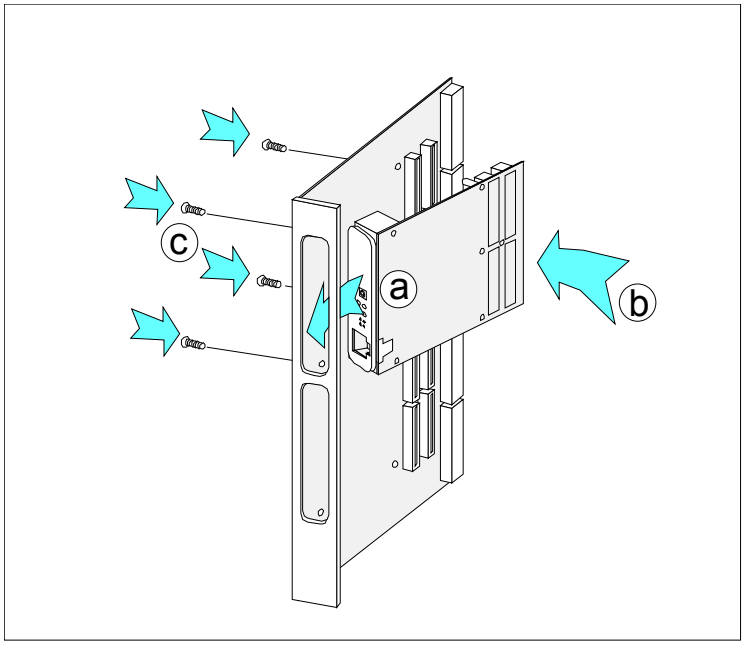
There are situations where none of the backplane connectors effectively used have alignment guides. In this case, there is a very high risk of bad insertion into the backplane, resulting in twisted, crushed or even destroyed pins on the backplane connector. You are encouraged to be very careful.

e. Close the card's handles by pulling them towards each other until they click into place.

f. Secure the adapter in place with two screws located on top and bottom of the faceplate.

Step 3: Installing the 4538 on the Carrier Card

This step is the same as the installation of a 4538 “Front Access” on a carrier card. Refer to *Installing the 4538 “Front Access”* on page 4.



Step 4: Installing the Carrier Card into the CompactPCI Chassis

With a grounding strap connected to your wrist or ankle, do the following to install the card:

a. If the CompactPCI chassis does not support “Hot Swap”, turn off system power.

b. Remove the chosen rear slot’s expansion cover.

c. Ensure that the CompactPCI carrier card handles are open (pointing away from the top and bottom of the board).

d. Place the assembly back in the carrier board PCI slot.

e. Close the card’s handles by pulling them towards each other until they click into place.

f. Secure the adapter in place with two screws located on top and bottom of the faceplate.

g. Turn on system power (if not already on)

Connecting to the Card 3

Overview

Figure 3-1 shows the “Front Access” 4538 front panel. Several connections are made on this panel. the “Rear Access” 4538 front panel is the same, except that connectors J1 and J2 are removed.

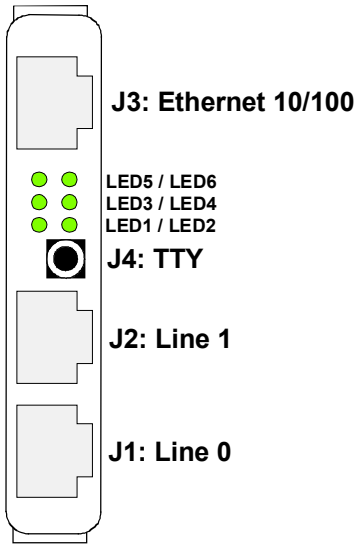


Figure 3-1. 4538 “Front Access” Front Panel

E1/T1/J1 Connectors

The 4538 “Front Access” provides direct connections to two lines through RJ48C connectors.

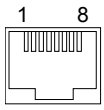
The 4538 “Rear Access” provides four line interfaces, indirectly through PMC connector P4. As a result, the connections to the T1/E1/J1 lines are made on the other card. An example of complete E1/T1/J1 functionality in a CompactPCI chassis with the 4538 “Rear Access” is the 4538 placed on a Motorola CPV8540 or Motorola MCPN750 CompactPCI carrier card and combined with the Interphase 6435 CompactPCI E1/T1/J1 Interface Rear Transition Module.

Customers who want to implement the 4538 on other configurations have to design their own line interfaces. For these customers, Interphase can provide additional information, such as schematics and a bill of material for these line interfaces.

The E1/T1/J1 connectors on the “Front Access” 4538 (J1 and J2) and on the E1/T1/J1 connectors the 6435 Rear Transition Module are unshielded RJ48C connectors. Table 3-1 on page 10 shows the pin-out for these connectors.

.

Table 3-1. RJ48C E1/T1/J1Connectors

	Signal
1	IN1
2	IN2
3	
4	OUT1
5	OUT2
6	
7	
8	

Connecting to the E1/T1/J1 Lines

The method for connecting your system to the lines depends on whether your carrier provides a T1/J1 PRI line or an E1 PRI line. Generally, carriers in North America and Japan provide a T1/J1 PRI line, and carriers in Europe and Australia provide an E1 PRI line. This chapter provides the following information required to connect your card to the line:

- Connection methods and requirements for connecting to a T1/J1 PRI line
- Connection methods and requirements for connecting to an E1 PRI line
- Summary of steps to connect the system to the network

If your carrier provides a T1/J1 line, continue to the next section, T1/J1 PRI Line. If your carrier provides an E1 line, skip to E1 PRI Line.

T1/J1 PRI Line

When you connect the card to a T1/J1 PRI line, first determine the appropriate connection method, cabling requirements, and link requirements.

Connection Method

If the distance to your carrier is less than 655 feet or 200 meters, you can connect the card in a DSX1 (short haul) configuration directly to the T1/J1 PRI line. If the distance is more than 655 feet or 200 meters (up to 6,200 feet or 1.8 kilometers), you must connect the card in a DS1 (long haul) configuration. In the DS1 configuration, a Channel Service Unit (CSU) must translate the short haul DSX1 interface to support the longer distance.

The 4538 card includes an internal CSU, which enables it to connect directly to the T1/J1 PRI line in a DS1 configuration.

Connecting Directly to the Line

Direct connection is subject to constraints imposed by FCC rules and by your partner carrier. To determine if you can connect directly to the interface, and for more information about direct connection constraints, contact your carrier.

WARNING

To avoid harming the network, contact your carrier for approval before connecting the system directly to the line.

CAUTION

In accordance with FCC Rules, Part 68.218 (b), you must notify the telephone company prior to disconnecting the card from the line or turning off power to the card's host system. Without this prior notification, the carrier might temporarily discontinue your T1/PRI service.

Connecting Directly in a DSX1 Configuration

You can connect the card directly to the T1/J1 PRI line without using a CSU if the distance to your carrier is less than 655 feet or 200 meters. The line will be in a DSX1 configuration.

Connecting Directly in a DS1 Configuration

If the distance to your carrier is up to 6,200 feet (1.8 kilometers), you can connect directly to the T1/J1 PRI line with the 4538 card's internal CSU activated. The line will be in a DS1 configuration.

Connecting Through an External CSU to the Line

If you do not use the card's internal CSU to connect to the T1/J1 PRI line in a DS1 configuration, you can connect the card to an external CSU that connects to the metallic T1/J1 PRI interface. (You might want to use this method if the distance to your carrier exceeds 6,200 feet or 1.8 kilometers but is less than 6,855 feet or 2.0 kilometers.)

WARNING

To avoid harming the network, contact your carrier for approval before connecting the system directly to the line.

CAUTION

In accordance with FCC Rules, Part 68.218 (b), you must notify the telephone company prior to disconnecting the card from the line or turning off the power to the card's host system. Without this prior notification, the carrier might temporarily discontinue your T1/PRI service.

T1/J1 PRI Cabling Requirements

The cable between the card and the carrier or the CSU must meet the following standard T1/J1 attenuation and transmission requirements:

- 100 Ohm for T1, 110 Ohm for J1
- Two twisted pairs, Category 3 or higher
- Maximum length: 655 feet (200 m) without a CSU, or 6,200 feet (1,800 meters) with a CSU
- 26 AWG or larger wire must be used

Measure and record the actual cable length between the system and the carrier or the CSU. You will need this information when you configure the line profile parameters.

The cable must include an RJ48C plug at the end dedicated to the card. It must also provide the appropriate plug or cabling system at the end connected to the carrier or CSU. See relevant carrier or CSU documentation for more information.

If the card connects to a T1/J1 PRI line, the card and CSU must be configured with several common parameters to interoperate. The card's T1/J1 parameters are set as follows:

- Line coding: B8ZS
- Frame format: ESF (Extended Super Frame)
- Line I/O impedance: 100 Ohm ± 5%

E1 PRI Line

When you connect the card to an E1 PRI line, first determine the appropriate connection method, cabling requirements, and link requirements.

Connection Method

Generally, your E1 PRI carrier provides a Network Termination 1 (NT1) device to interface between the system and the metallic interface of the network.

Two types of cables are used for E1: 120 Ohm twisted pairs or 75 Ohm coaxial cables. Interphase provides the 6435, a Rear Transition Module dedicated to 120 Ohm E1 twisted pairs (and also T1/J1).

NOTE

Contact Interphase for applications requiring 75 Ohm E1 coaxial lines

120 Ohm E1 PRI Cabling Requirements

The cable between the card and the NT1 must meet the following standard E1 attenuation and transmission requirements:

- 120 Ohm
- Two symmetrical twisted pairs, Category 3 or higher
- Maximum length: determined in order to have less than 6 dB attenuation at 1024 kHz (G.703). This generally leads to 655 to 1000 feet (200 to 350 m)

The cable must include an RJ48C plug. It must also provide the appropriate plug or cabling system at the end dedicated to the NT1.

E1 Link Requirements

If the system connects to an E1 line, the 4538 and NT1 must be configured with several common parameters to interoperate. The 4538 card's E1 parameters are in accordance with the IUT-T I 431 recommendation, as follows:

- Line coding: HDB3, according to IUT-T G.703
- Frame format: according to IUT-T G.704
- CRC4 to Non-CRC4 operation: according to IUT-T G.706 An.B
- Line I/O impedance: 120 Ohm ± 5%

RS232 TTY Port Connector (J4)

The front panel contains a 2.5 mm stereo jack receptacle used for the RS232 TTY port. Figure 3-2 shows the cable connector. Table 3-2 shows the pin-out for this connector.

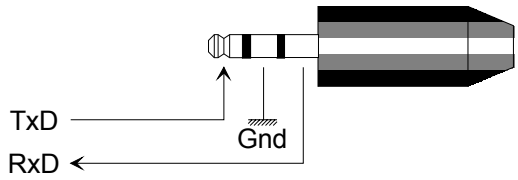


Figure 3-2. TTY Cable Connector: 2.5mm Stereo Jack Plug

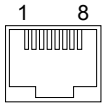
Table 3-2. J2 RS232 TTY Connector

Pin	Signal
Tip	TxD
Ring	Ground
Sleeve	RxD

Front Panel Ethernet Connector (J3)

The front panel also contains one shielded 8-pin modular jack connector for the Ethernet 10/100 interface (twisted pair). Table 3-3 shows the pin-out for this connector.

Table 3-3. J3 RJ45 Ethernet Connector

	Signal
1	OUT+
2	OUT–
3	IN+
4	N/U
5	N/U
6	IN–
7	N/U
8	N/U

Troubleshooting 4

Overview

This chapter provides possible solutions for common problems you might encounter while installing and operating the card. Before proceeding with troubleshooting, make sure you have carefully followed the steps for installing and setting up the hardware and the software, and have rebooted the system.

If the information in this chapter does not solve the problem, contact Interphase Customer Support at one of the locations listed in the *Assistance* section at the beginning of this document.

Interpreting LEDs

The board provides six LEDs, as shown in Figure 4-1.

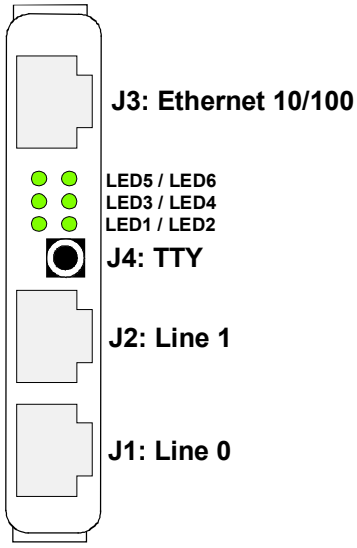


Figure 4-1. LED Arrangement

LED1: Synchronization signal provided by Framer 1 for Line 0

LED2: Synchronization signal provided by Framer 2 for Line 1

LED3: LXT971 LED driver 1

LED4: LXT971 LED driver 2

LED5: LXT971 LED driver 3

LED6: User-programmable LED

Table 4-1 describes the LED indications after VxWorks is running on the 4538 board (i.e., once the Boot Firmware has run 4538 POST).

Table 4-1. Ethernet LED Color Indications

LED	Color	Meaning if the LED is ON
1	Green	Synchronization signal provided by the Framer 1 for Line 0
2	Green	Synchronization signal provided by the Framer 2 for Line 1
3	Green	Ethernet Link status
4	Green	Ethernet Duplex status
5	Green	Ethernet Collision status
6	Green	CPU is active when LED blinks: If it blinks normally (every 500ms) POST succeeded If it blinks quickly (every 125ms) POST failed

Problems and Solutions

Table 4-2. Card Problems and Possible Solutions

Problem	Possible Solution(s)
Computer does not boot up or host communications controller not found	The PCI bus automatically configures the hardware resources used by the card. Therefore, a resource conflict (address or IRQ) is probably not the problem. To solve the problem: **1.** Check to see whether the board is properly seated in its slot. **2.** Try the board in a different slot. **3.** Contact Interphase Customer Support at one of the locations listed in the *Assistance* section at the beginning of this document.
POST fails	CPU LED is active and blinks quickly (every 125 ms). Use either the TTY console or `iphMonitor` PCI Console utility (for VxWorks, see *PCI Console iphMonitor* on page 27, for Solaris, see *PCI Console iphMonitor* on page 44) to run the test again and determine the problem. If the problem cannot be solved, contact Interphase Customer Support at one of the locations listed in the *Assistance* section at the beginning of this document.
POST and Boot Firmware menu are not displayed on the TTY console	**1.** Check TTY cable (see *RS232 TTY Port Connector (J4)* on page 13). **2.** Check COM port parameters: 9600 baud, 8 data bits, 1 stop bit, No flow control.

VxWorks® Maintenance Tools

A

Overview

VxWorks Maintenance tools are provided for Interphase board testing and update purposes. This set of tools has access to the board only through the PCI bus and is compatible with Interphase communication controllers including 4538 based on PowerQUICC II and PowerSpan (PowerPC/PCI bridge) technologies.

- *iphShowDev*: Displays Interphase boards detected on the PCI bus.
- *iphCardInfo*: Displays version information for the detected communications controller(s).
- *iphSetup*: Allows storing code in the Flash EEPROM. This utility is also able to upgrade either the PCI Serial EEPROM (which contains the 4538 power-on configuration), or the Electrically Programmable Logic Device (EPLD).
- *iphPCITest*: Tests PCI DMA transfer and PCI interrupt between the host and communications controller.
- *PCI Console iphMonitor*: Allows monitoring the embedded 4538 boot firmware through the PCI interface. This tool can be used when no TTY console is connected to the 4538 TTY port.
- Other tools are provided for debug purposes (*iphDumpEeprom*, *iphDumpMem*, and *iphDumpReg*).

CAUTION

The Maintenance Tools have been developed for the Motorola MCP750 PCI Interface (BSP Version 1.2/3 by default). PCI Function prototypes and constants may change from one system board to another. To use these tools with another system board, add functions and constants definitions, according to your system board documentation to reflect your system board PCI Interface (functions and constants are located in the `iphSetupTools/c/board.c` file), and select your board by changing the BOARD constant value.

The board-dependent constants and functions needed by the maintenance tools are gathered in the following structure (defined in `iphSetupTools/h/deftypes.h`):

```
typedef struct BoardFuncs
{
   UINT32 pciBaseAddr; /* Host memory space */
   UINT32 hostBaseAddr; /* PCI memory space */
```

PCI_INTRFUNCPTR intrCtrl; /* interrupt control function */

PCI_READFUNCPTR pciConfigInLong; /* PCI configuration read function */

PCI_WRITEFUNCPTR pciConfigOutLong; /* PCI configuration write function */

PCI_FINDFUNCPTR pciFindDevice; /* PCI bus probe function */

PCI_GETINTERRUPTPTR pciGetInterrupt; /* get interrupt line in device PCI configuration space */

} BoardFuncs_t;

BoardFuncs_t iphBoardFuncs;

The global variable `iphBoardFuncs` is initialized in the `board.c` file according to the selected board.

Minimum System Requirements

- A PCI Master system board running VxWorks 5.4 (Tornado® 2.0): These tools must be executed on the PCI Master system board (this is the board that is plugged into the system slot of the CompactPCI system, which is usually a red slot) and not on the 4538 adapter itself.
- VxWorks Board Support Package (BSP) for this system board
  - The BSP must provide PCI bus management to be able to access the 4538 communications controller.

Installing the Tools

The following procedure must be done to install the tools:

1. Create an `iphSetupTools` directory in your Tornado project directory.
2. Copy the VxWorks maintenance tools from the Board Development Kit CD-ROM to this directory
   (for example: **`copy e:\tools\vxworks\*.*c:\tornado\proj\iphSetupTools`**).

Loading the Tools

This paragraph describes how to load the maintenance tools for Interphase boards. If a previous version has been loaded, it is strongly recommended that you reboot the PCI master VxWorks target before loading a new version.

To build the tools, open `iphSetupTools project` (**`target/proj/iphSetupTools.wpj`**), check that the build properties are according to the selected system board in the `board.c` file, update dependencies and rebuild the project.

```
ld < target/proj/iphSetupTools/out/iphSetupTools.out
```

This operation will add the following symbols to the symbol table:

- *iphShowDev* on page 21
- *iphCardInfo* on page 22
- *iphSetup* on page 22
- *PCI Console iphMonitor* on page 27
- *iphPCITest* on page 28
- *iphDumpReg* on page 31
- *iphDumpMem* on page 32
- *iphDumpEeprom* on page 32

Using the Tools

This section provides additional detailed information about the tools.

In display examples, "`->`" is the VxWorks prompt, and user commands are in bold characters.

iphShowDev

The `iphShowDev` routine tries to detect all Interphase communication controllers on the PCI bus. Each detected board is given an index from 0 to MAX_INDEX (fixed to 16) displayed with bus number, slot number, function, serial number, device ID, and card name. This index list is the reference for device index parameter used by other tools.

Usage

```
iphShowDev
```

No parameters.

Example

```
-> iphShowDev

Board(s) found with Interphase vendor ID (0x107E):
----- --- ----  -------- ------  ----------------------
Index Bus Slot  Function DevId   SerialNum  Card Name
----- --- ----  -------- ------  ----------------------
0      01  08       00   0x9060   1540120    6535 cPCI 4P/8P T1/E1/J1
1      02  04       00   0x9080   1428129    4532 PMC MM SR SONET 155Mb/s


value = 1 = 0x1
->
```

If there are no Interphase communication controllers on the PCI bus, the following message is displayed:

```
-> iphShowDev

No Interphase device found
value = 1 = 0x1
->
```

iphCardInfo

This tool prints board revision information contained in the serial EEPROM.

Usage

```
iphCardInfo(deviceIndex)
```

`deviceIndex` is an integer representing the index of the device to dump. The `iphShowDev` utility can help the user view all indexes.

Example

```
-> iphCardInfo(0)
------------------------------------------------------------
        Card informations utility for Interphase Communications Controllers
        Version 1.13
        Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------


PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId  = 0x9080
SerialNum = 1428129


Revisions :
   Card Date     : 06/03/2001
   PSPAN EEPROM  : SX00556A00
   EPLD (1rst)   : SX00557A00
   Boot Firmware : SX00558A00
value = 0 = 0x0
->
```

iphSetup

The `iphSetup` tool is mainly used to update the 4538 Flash EEPROM (for example, to flash a VxWorks Bootrom or to upgrade Boot Firmware). This utility is also able to upgrade either the PCI Serial EEPROM (which contains the 4538 power-on configuration), or the EPLD.

CAUTION

Upgrade of the PCI Serial EEPROM or EPLD should only be used under Interphase control, because incorrect use of these two options may put the board out of service.

Usage

```
iphSetup (deviceIndex, filename, flashLoadOffset, compare)
```

Where:

`deviceIndex` is an integer representing the index of the device to set up. The `iphShowDev` utility can help the user view all indexes.

`filename` is a string that is the name, with complete path, of the file to use for the set up. The following types of files can be used:

- With `.isp` extension for EPLD setup: this file is generated by specific tools for the Lattice EPLD and must not be edited. If needed, this file will be provided by Interphase.
- With `.psp` extension for serial EEPROM programing: this file is an ASCII file describing in C-language (bytes array) the first 64 bytes of the EEPROM. If needed, this file will be provided by Interphase.
- With `.bin` extension for flash data loading: this file contains the data to load in flash. It is a binary file with raw data.
- With `.boo` or `.elf` extension for flash loading: this file is the code to load in flash. It is a binary ELF format file for PowerPC.

`flashLoadOffset` is an integer used for flash memory loading and can be expressed either in hexadecimal (0x0123ABCD) or in decimal (12345). `iphSetup` will ignore most significant bits according to Flash EEPROM size (for example, on a 4 MB Flash EEPROM size, only the 22 least significant bits are taken into account, i.e. mask 0x003FFFFF). For firmware and operational codes (`.elf` or `.boo` extension files), the effective address is computed by adding this offset to the address in the ELF format file. For a binary data file (`.bin` extension file), `flashLoadOffset` is considered as an absolute address in the flash memory. For other extension files (`.psp`, `.isp`), this parameter is ignored.

`compare` is a Boolean: when it is equal to 0, the specified file is used to update the board; when it is equal to 1, the file content is compared to the EPLD, EEPROM, or flash memory content according to the file extension.

The communications controller is rebooted without update when `filename` is an empty string.

If `iphSetup` is called without parameters, or if the file name has an invalid extension file, the `iphSetup` usage description is displayed

Source Description

The main function `iphSetup,` prints an about message, checks the file name given in the `filename` parameter, tries to open access to the device according to the specified index, calls the appropriate subroutine according to the extension of the specified file name, and then closes access to the device.

The device access opening function (`OpenDevice`):

- Allocates memory for device descriptor structure
- Checks device presence on the bus
- Does hardware initialization in PCI configuration space header (latency timer and cache line size values)
- Gets two base address registers (BAR0/BAR1 and BAR2) values for PCI chip registers and memory access
- If not already done, installs an interrupt handler according to the interrupt line of the PCI configuration space header
- Prints memory and PCI chip register base addresses, device ID, and serial number
- Reads the PCI-to-local window size from the PowerSpan target image control register
- Enables target image, address translation, and base address register bits in the target image control register
- If not already done, configures the PowerSpan interrupts

The EEPROM update function (`programPspEeprom`):

- Opens the EEPROM data file
- Gets file date and time (or current date and time if file information is not available), and formats them
- Gets the EEPROM checksum, and if wrong, sets default configuration and new checksum
- Reads the EEPROM data file
- Compares file data to or writes file data into the EEPROM
- Closes the file
- Updates card date and SX number in the EEPROM
- Updates checksum

The EPLD update function (`programIspEpld`):

- Opens the EPLD data file
- Gets file date and time (or current date and time if file information is not available), formats them
- Puts the board in reset state
- Asks the user to put a strap on a jumper and hit a key
- Reads EPLD data file, and compares it to or writes it in EPLD
- Closes the file

- Asks the user to remove the strap from the jumper and hit a key
- Runs the board (gets out from reset state)
- Updates the SX number in EEPROM

The card reboot function (`iphReboot`):

- Puts the board in reset state
- Sets the ATN bit (doorbell bit 2 in Interrupt Enable Register 0)
- Runs the board (gets out from reset state)
- Waits for ATN bit reset (doorbell bit 2 in Interrupt Status register) by the Boot Firmware (approximately 20 seconds)

The flash update function (`programFlash`):

- Puts the board in reset state
- Sets flash mode
- Detects the flash type
- Opens the flash data file
- Gets file date and time (or current date and time if file information is not available), formats them
- Determine flash sectors to erase. If there is no sector to erase, it means that the combination of addresses in ELF file and/or the given offset do not correspond to the flash memory area; in this case, after asking the user to verify the specified file and offset, resets flash mode, runs the board, and exits
- If one or more of the first four sectors are to be erased, asks the user to confirm
- Erases sectors
- Loads flash memory with ELF or binary file content and closes the file
- Updates the SX number (or current date if invalid SX number) in EEPROM
- Updates checksum in EEPROM
- If comparison flag is set, compares ELF or binary file data to flash content and closes the file
- Resets flash mode
- Prints board information
- Reboots the card (calls the `iphReboot` function)

Example

Example for flashing Boot Firmware (upgrade of Interphase Boot Firmware):

```
-> iphSetup(0, "bootfirm.elf", 0, 0)

------------------------------------------------------------
          Setup tool for Interphase Communications Controllers
          Version 1.13
          Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------
PSPAN regs base address : 0xC1000000
```

```
Memory base address     : 0xC1200000
DeviceId = 0x9080
SerialNum = 1428129


Program FLASH :
Erasing FLASH : sectors to erase = 0. 48. 49. 50. 51.
  You are going to load in boot firmware area and the current boot firmware
may be lost
  The board will not boot again, if the file bootfirm.elf is not a correct
boot firmware
  Do you really want to erase these flash sectors ('y' or 'n')?
....
  Erase ok
  Loading flash : successfully programmed


Revisions :
  Card Date     : 06/03/2001
  PSPAN EEPROM  : SX556A00
  EPLD (1rst)   : SX557A00
  Boot Firmware : SX558A00
value = 0 = 0x0
->
```

Example for flashing VxWorks bootrom:

```
-> iphSetup(0, "bootrom_uncmp.boo", 0x10000, 0)

------------------------------------------------------------
          Setup tool for Interphase Communications Controllers
          Version 1.13
          Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------
PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId = 0x9080
SerialNum = 1428129


Program FLASH :
Erasing FLASH : sectors to erase = 1. 2. 3. 4. 5. 6. ......
  Erase ok
  Loading flash : successfully programmed


Revisions :
  Card Date     : 06/03/2001
  PSPAN EEPROM  : SX556A00
  EPLD (1rst)   : SX557A00
  Boot Firmware : SX558A00
value = 0 = 0x0
->
```

PCI Console iphMonitor

The `iphMonitor` application starts a dialog session with the Interphase embedded monitor. Within this session, the user can send commands to, and display messages from, the on-board monitor. This tool can be used, for example, when no TTY console is connected to the 4538 TTY port and the user needs to dialog with the 4538 Boot firmware. When using this function in a Tornado shell, the user commands are not echoed.

NOTE

The `iphMonitor` command can only be used when 4538 boot firmware is waiting for a command (either from a TTY or from PCI consoles) and not when boot firmware has already jumped to a VxWorks kernel or any operational firmware.

Overall commands used through `iphMonitor` are the same as described in the BIST and Monitor Manual, except LOADFC. For more information, see the *4538 Built-In Self Test and Monitor Manual* (UG04538-004).

Usage

```
iphMonitor(deviceIndex, reboot)
```

`deviceIndex` is an integer representing the index of the device to monitor. The `iphShowDev` utility can help the user view all indexes.

`reboot` indicates if the board must be rebooted before the dialog session or not. This parameter is useful when the board is configured to jump to an operational firmware after the POST, and there is no TTY interface.

If reboot = 0, the dialog session is opened directly.

If reboot =1, the card is rebooted and at the end of the POST, a carriage return is sent in the board-host PCI exchange area. This operation forces the boot firmware to start a dialog session, even if it was configured to jump to an operational firmware.

Source Description

The `iphMonitor` function:

- Opens device access
- Checks the card type to determine if it is possible to dialog with
- If reboot = 1, reboots the card and simulates a carriage return entry to avoid jumping to an operational firmware after the POST
- Opens a monitor dialog session and enters main loop:
  - If there is data in the board-to-host exchange area, gets and prints it
  - Gets user's command and puts it in the host-to-board exchange area
  - If the user command is **`exit`**, gets out from the loop and closes the monitor dialog session

- Closes device access

Example

```
-> iphMonitor(0,1)

------------------------------------------------------------
          PCI monitor for Interphase Communications Controllers
          Version 1.13
          Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------
PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId = 0x9080
SerialNum = 1428129


Rebooting card...ok
Sending<CR>...

Dialog with on-board monitor (type 'exit' to end)
------------------------------------------------


-----------------------------------
Boot Firmware Version 1.24
Copyright (c) 2000 Interphase Corp.
-----------------------------------


Commands' categories:
I   : Information
C   : Configuration
T   : Tests
U   : Utilities
To show all commands on a category, type corresponding letter
>
>value = 0 = 0x0
->
```

iphPCITest

The `iphPCITest` application launches DMA transfer tests between host memory and Interphase board local memory.

Usage

`iphPCITest(deviceIndex)`

`deviceIndex` is an integer representing the index of the device to test. The `iphShowDev` utility can help the user view all indexes.

Source Description

The `iphPCITest` function:

- Opens device access
- Tests each DMA channel by performing board local memory to host memory transfers in direct mode DMA transfer
- Tests PowerSpan linked-list mode DMA
- Tests board-to-host interrupt servicing (interrupt is generated by DMA transfer end)
- Closes device access

Direct Mode DMA Test

- Allocates a buffer in host memory and gets its physical address.
- Allocates a buffer in board local memory. The lower three bits of the buffer address are taken from the lower three bits of the host buffer address to enforce 8-byte alignment of starting source and destination addresses for DMA transfers.
- Disables DMA interrupts.
- Initializes local memory buffer with a pattern.
- Initializes host memory buffer with a different pattern from local memory.
- Ensures that the DMA channel to use is not active.
- Programs source and destination port, byte count, and endian conversion if needed.
- Programs source and destination addresses.
- Clears status bits and sets `go` bit.
- Tests activity bit to verify DMA launching.
- Polls end of DMA transfer (by testing activity bit reset) with a time out.
- Checks DMA status (done without error).
- Checks transfer by comparing local and host memory buffers, they must be identical.

Linked List Mode DMA Test

- Allocates a buffer in host memory and gets its physical address.
- Allocates a buffer in board local memory. The lower three bits of the buffer address are taken from the lower three bits of the host buffer address to enforce 8-byte alignment of starting source and destination addresses for DMA transfer.
- Disables DMA interrupts.
- Initializes local memory buffer with a pattern.
- Initializes host memory buffer with a pattern different from local memory.
- Allocates two 32-byte aligned buffers for command packets and gets physical address.
- Initializes the two command packets, with their source and destination addresses, source and destination ports, byte count and endian conversion, and the next packet.
- Ensures that the DMA channel to use is not active.

- Sets up the command packet port.
- Sets up the DMA channel to point to first command packet address.
- Ensures that the byte count is 0 for the DMA channel.
- Clears status bits and sets go bit.
- Tests activity bit to verify DMA launching.
- Polls end of DMA transfer (by testing activity bit reset) with a time out.
- Checks DMA status (done without error).
- Checks transfer by comparing local and host memory buffers, they must be identical.

DMA Interrupt Test

The interrupt test is a direct-mode DMA test, but DMA end polling is done on a global variable set in the interrupt handler. The DMA done interrupt must be enabled.

Example

```
-> iphPCITest(1)

------------------------------------------------------------
          PCI tests utility for Interphase Communications Controllers
          Version 1.13
          Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------


PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId = 0x9080
SerialNum = 1428129


DMA transfers and interrupt tests
---------------------------------


DMA channels test :
   DMA0 local -> PCI   : OK
   DMA1 PCI   -> local : OK
   DMA2 local -> PCI   : OK
   DMA3 PCI   -> local : OK

Linked list test (2 command packets) :
   Local to PCI : OK
   PCI to local : OK

DMA0 Interrupt test (local -> PCI) : OK
DMA1 Interrupt test (local -> PCI) : OK
DMA2 Interrupt test (local -> PCI) : OK
DMA3 Interrupt test (local -> PCI) : OK
value = 0 = 0x0
->
```

If the DMA transfer test does not succeed, the five first 4-byte words of each buffer (local and PCI host) and total number of errors are displayed, as follows:

```
. Channel0 :
  Offset 0x0000, PCI = 0x00000000, Loc = 0x324BB4AA.
  Offset 0x0004, PCI = 0x00000001, Loc = 0xDCF65F55.
  Offset 0x0008, PCI = 0x00000002, Loc = 0x213AA399.
  Offset 0x000C, PCI = 0x00000003, Loc = 0xEE077066.
  Offset 0x0010, PCI = 0x00000004, Loc = 0x24FF9425.
  ...  0x400 errors
```

If the interrupt test does not succeed, a time-out error is displayed, as follows:

```
DMA0 Interrupt test (local -> PCI) : timeout error, DMA end not detected
```

iphDumpReg

This tool dumps and prints all PowerSpan registers. The display shows register name, offset, and value.

Usage

```
iphDumpReg(deviceIndex)
```

`deviceIndex` is an integer representing the index of the device to dump. The `iphShowDev` utility can help the user view all indexes.

Example

```
-> iphDumpReg(0)

---------------------------------------------------------------
       Registers dump utility for Interphase Communications Controllers
       Version 1.13
       Copyright (c) 2001 - Interphase Corp.
---------------------------------------------------------------


PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId = 0x9080
SerialNum = 1428129


      Name      Offset      Value
  -------------- -------   ----------
  P1_ID          (0x000) = 0x9080107E
  P1_CSR         (0x004) = 0x0AB00002
  P1_CLASS       (0x008) = 0x02800002
  P1_MISC0       (0x00C) = 0x00004000
  P1_BSI2O       (0x010) = 0x02100008
  ........
  HOST_OIO       (0x540) = 0x00000000
  HOST_OIA       (0x544) = 0x00000000
  IOP_OI         (0x548) = 0x00000000
```

```
   IOP_OI_INC     (0x54C) = 0x00000000
value = 0 = 0x0
->
```

iphDumpMem

This tool dumps and prints board local memory. The display shows the offset from the start address, memory content, and ASCII interpretation.

Usage

```
iphDumpMem(deviceIndex, start_address, size)
```

`deviceIndex` is an integer representing the index of the device to dump. The `iphShowDev` utility can help the user view all indexes.

`start_address` is the memory address to start display.

`size` is the memory size in bytes to display.

Example

```
-> iphDumpMem(0,0,64)


------------------------------------------------------------
        Memory dump utility for Interphase Communications Controllers
        Version 1.13
        Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------
PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId = 0x9080
SerialNum = 1428129


  SDRAM read (start address = 0x00000000) :
Offset
  000  00 01 D9 26 55 55 55 55 99 99 99 99 66 66 66 66   ..Ú&UUUU™™™™ffff
  010  9D 5E 8A 25 48 09 34 D0 8C 4D 79 14 59 1A 45 E1   ”^Š%H.4ÐŒMy.Y.Eá
  020  90 12 69 A0 3A BD 14 4B 7F 01 58 8F 4B CE 25 5C   .i:½.K-.XKÎ%\
  030  82 C6 49 1B 2D 70 F3 C6 71 B5 38 0A 3E 82 04 D7   ‚ÆI.-póÆqµ8.>‚.×
value = 0 = 0x0
->
```

iphDumpEeprom

This tool dumps and prints board serial EEPROM. The display shows the offset from the start address, memory content, and ASCII interpretation.

Usage

```
iphDumpEeprom(deviceIndex)
```

`deviceIndex` is an integer representing the index of the device to dump. The `iphShowDev` utility can help the user view all indexes.

Example

```
-> iphDumpEeprom(0)

------------------------------------------------------------
       Serial eeprom dump utility for Interphase Communications Controllers
       Version 1.13
       Copyright (c) 2001 - Interphase Corp.
------------------------------------------------------------


PSPAN regs base address : 0xC1000000
Memory base address     : 0xC1200000
DeviceId  = 0x9080
SerialNum = 1428129


  Serial EEPROM content :
Offset
 0000  02 00 00 00 00 0C 00 90 80 10 7E 02 1C 53 00 00   .......`~..S..
 0010  80 5F 20 00 00 00 00 00 00 00 00 00 00 00 00 00   __..............
 0020  90 80 10 7E 02 80 00 00 00 00 00 00 00 00 C0 00   _`.~._........À.
 0030  00 F0 02 00 00 00 00 00 00 00 00 00 00 00 00 00   .ð..............
 0040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 0050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 0060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 0070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 0080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 0090  FF FF 66 2A FF FF FF FF FF FF 56 05 57 05 58 05 ÿÿf*ÿÿÿÿÿÿV.W.X
 00A0  18 80 17 00 FF FF FF FF FF FF 00 00 00 00 24 00   ._..ÿÿÿÿÿÿ....$.
 00B0  00 00 77 97 80 18 FF FF FF FF FF FF FF FF 00 80   ..w--_.ÿÿÿÿÿÿÿÿ.
 00C0  FF FF 40 80 FF 81 01 00 FF BF 00 00 FF FF FF FF   ÿÿ@_ÿ_..ÿ¿..ÿÿÿÿ
 00D0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 00E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
 00F0  FF FF FF FF FF FF FF FF FF FF FF FF 01 02 04 1E ÿÿÿÿÿÿÿÿÿÿÿÿ....
value = 0 = 0x0
->
```

Source File Organization

All the source files necessary to build the tools are contained in the `$(WIND_BASE)/target/proj/iphSetupTools` file:

- The C source files are located in the `c` subdirectory:
  - `board.c`: System board dependent
  - `eeprom.c`: Serial EEPROM manipulation functions (read, write, and checksum calculation)
  - `elf.c`: `elf` file manipulation functions (analyze and load in target memory)
  - `lattice.c`: EPLD functions (this file is provided according to the EPLD and must not be modified)

  - psppci.c: Functions to dialog with the on-board monitor via the PCI bus
  - readh.c: Functions to read the EEPROM file
  - utils.c: Miscellaneous utility functions (buffer display, delay functions)
  - flash.c: Flash memory manipulation functions (for flash-type detection, byte programming, sector or all flash erase, get file data and write it in flash, compare file data to flash content)
  - pspdma.c: PowerSpan DMA test functions (initiate direct-mode or linked-list mode DMA transfer, check DMA status, compare areas of board local memory and host memory, configure and test DMA interrupt)
  - plxdma.c: PLX DMA test functions (initiate direct-mode DMA transfer, check DMA status, compare areas of board local memory and host memory, configure and test DMA interrupt)
  - iph_ithandle.c: Interrupt handler routines (manage only DMA interrupts)
  - common.c: Miscellaneous routines for hardware access (registers and memory)
  - iphsetup.c: Tools main functions
- The include files are located in the h subdirectory:
  - iphsetup.h: Header file for iphSetup.c
  - eeprom.h: Header file for eeprom.c
  - elfdefs.h, elfppc.h, elftypes.h: Header files for elf.c
  - deftypes.h: Global header file
  - lattice.h: Header file for lattice.c
  - psppci.h: Header file for psppci.c
  - readh.h: Header file for readh.c
  - utils.h: Header file for utils.c
  - flash.h: Header file for flash.c
  - pspdma.h: Header file for pspdma.c
  - plxdma.h: Header file for plxdma.c
  - ithandle.h: Header file for iph_ithandle.c
  - common.h: Header file for common.c
  - pspdef.h: PowerSpan definitions
  - vxport.h: Definitions for VxWorks portability
  - toolsVersion.h: file for tool version update
- The assembly files are located in the s subdirectory:
  - iph_iol_vx.s: Low level input/output functions

The project directory contains a makefile and a Tornado 2.0 project file (*.wpj) to build a downloadable object module. It also contains a readme.txt containing a short description of the project.

Troubleshooting

MCP750 BSP Version 1.2/0

If you are using the MCP750 BSP version 1.2/0, you need to do some modifications prior to building the Maintenance Tools:

- In `board.c` file replace `PCI_MSTR_MEMIO_LOCAL` with `CPU_PCI_ISA_MEM_ADRS`
- In project build properties (C/C++ compiler options), replace `C:/TORNADO/target/config/mcpn750` with `C:/TORNADO/target/config/mcp750`

PCI Console Monitor

If you see the `Embedded monitor is busy` message (or no response from the board) when you call the `iphMonitor` function, it means the board is not running the boot firmware, so enter **`exit`** to quit the `iphMonitor` function. To force the boot firmware to start a dialog session, call `iphMonitor` with 1 as the second parameter; the card is rebooted and at the end of the POST, a carriage return is sent in the board-host PCI exchange area.

MCP750 with PowerSpan

The PowerSpan does not react correctly with some MCP750's BSP version (1.2/3 for example), which twice detects the same card on different slots of the PCI bus. To solve this problem, patch the code of the `pciAutoFuncConfig` function in the `pciAutoConfig.c` file, rebuild the bootrom and the kernel of MCP750 BSP, and update the board with these new versions. The code modification is shown in the following extract:

```
/*****************************************************************************
*
* pciAutoFuncConfig - Assign memory and/or I/O space to single function.
*
* This routine allocates and assigns memory and/or I/O space to a
* single PCI function. Allocations are made for each implemented
* base address register (BAR) in the PCI configuration header.
*
* RETURNS: N/A.
*/


LOCAL void pciAutoFuncConfig
(
PCI_SYSTEM * pSystem,
PCI_LOC * pPciFunc  /* input: "Include list" pointer to function */
```

```
)
{
   UINT baMax;   /* Total number of base addresses    */
   UINT baI;  /* Base address register index     */
   UINT baseAddr;   /* PCI Offset of base address     */
   UINT readVar; /* Contents of base address register */
   UINT addrInfo;   /* PCI address type information     */
   UINT sizeMask;   /* LSbit for size calculation     */
   UCHAR headerType;   /* Read from PCI config header     */
   UINT dev_vend;
   UINT i, addOnes;

   /* If there is a function, then consult the exclusion routine */

   if ( (pSystem->includeRtn) != NULL )
   {
      pciConfigInLong (pPciFunc->bus, pPciFunc->device, pPciFunc->function,
                       PCI_CFG_VENDOR_ID, &dev_vend);
      if ( ((pSystem->includeRtn) (pSystem, pPciFunc, dev_vend)) == ERROR )
      {
         if ((pPciFunc->attribute & PCI_AUTO_ATTR_BUS_PCI) == 0)
         {
            pPciFunc->attribute |= PCI_AUTO_ATTR_DEV_EXCLUDE;
            PCI_AUTO_DEBUG_MSG("pciAutoFuncConfig: exc [%d,%d,%d,0x%02x]\n",
                               pPciFunc->bus, pPciFunc->device, pPciFunc->function,
                               pPciFunc->attribute,0,0);
            return;
         }
      }
   }

   /* Disable the function */

   pciAutoFuncDisable (pPciFunc);

   /* Determine the number of base address registers present */
```

```
pciConfigInByte (pPciFunc->bus, pPciFunc->device, pPciFunc->function,
                 PCI_CFG_HEADER_TYPE, &headerType);
headerType &= 0x7f;

switch (headerType)
{
   case PCI_HEADER_TYPE0:
      baMax = 6;
      break;

   case PCI_HEADER_PCI_PCI:
      baMax = 2;
      break;

   default:
      baMax = 0;
      break;
}

/* Allocate Memory or I/O space for each implemented base addr register */

for (baI = 0; baI < baMax; baI++)
{
   /* Get the base address register contents */

   baseAddr = PCI_CFG_BASE_ADDRESS_0 + (baI * 4);

   pciConfigOutLong (pPciFunc->bus, pPciFunc->device, pPciFunc->function,
                     baseAddr, 0xFFFFFFFF);

   pciConfigInLong (pPciFunc->bus, pPciFunc->device, pPciFunc->function,
                    baseAddr, &readVar);

   /* Go to the next BAR when an unimplemented one (BAR==0) is found */

   if (readVar == 0)
   {
```

```
            continue;
         }

         /* Mask off all but space, memory type, and prefetchable bits */

         addrInfo = readVar & PCI_BAR_ALL_MASK;

         /* Check for type, setup mask variables (based on type) */

         if ((addrInfo & PCI_BAR_SPACE_MASK) == PCI_BAR_SPACE_IO)
         {
            PCI_AUTO_DEBUG_MSG("pciAutoFuncConfig: IO Space found at BAR[%d]\n",
                               baI, 0, 0, 0, 0, 0);
            sizeMask = (1 << 2);
         }
         else
         {
            PCI_AUTO_DEBUG_MSG("pciAutoFuncConfig: MemSpace found at BAR[%d]\n",
                               baI, 0, 0, 0, 0, 0);
            sizeMask = (1 << 4);
         }

         /* Loop until we find a bit set or until we run out of bits */

         for (; sizeMask; sizeMask <<= 1)
         {
            /* is this bit set? if not, keep looking */

            if (readVar & sizeMask)
            {
      /* Start of Interphase patch on BARx value to avoid powerspan configuration
      initialization problems */
               if (((dev_vend & 0x0000ffff)==0x107E) &&
                   ((addrInfo & PCI_BAR_SPACE_MASK) != PCI_BAR_SPACE_IO))
               {
                  sizeMask<<=1; /* theoretical mem size multiplied by 2 */
                 baI += pciAutoRegConfig (pSystem, pPciFunc, baseAddr, sizeMask,addrInfo);
```

```
                /*read barx configured*/
                pciConfigInLong (pPciFunc->bus, pPciFunc->device,
                                pPciFunc->function, baseAddr, &readVar);
                readVar &= PCI_MEMBASE_MASK;

                addOnes=0;
                for (i = 0; i < 32; i++)
                {
                   addOnes += ((readVar >>i) & 0x01);
                }
                if (addOnes==1) /* address is one bit code*/
                {
                   readVar +=0x10000;
                   pciConfigOutLong (pPciFunc->bus, pPciFunc->device,
                                   pPciFunc->function, baseAddr, readVar);
                }
             }
             /*end of Interphase patch on BARx value*/
             else
             {

                baI += pciAutoRegConfig (pSystem, pPciFunc, baseAddr, sizeMask,
                                         addrInfo);
             }
             break;
          }
       }
    }
}
```

Solaris® Maintenance Tools

B

Overview

Solaris Maintenance tools are provided for Interphase board testing and update purposes. This set of tools has access to the board only through the PCI bus and is compatible with Interphase boards based on PowerQUICC I I and PowerSpan (PowerPC/PCI bridge) technologies.

- *iphSetup*: Allows storing code in the Flash EEPROM. This utility is also able to upgrade either the PCI Serial EEPROM (which contains the 4538 power-on configuration), or the Electrically Programmable Logic Device (EPLD).

CAUTION

Upgrade of the PCI Serial EEPROM or EPLD should only be used under Interphase control, because incorrect use of these two options may put the board out of service.

- *iphMonitor* Allows monitoring the embedded 4538 boot firmware through the PCI interface. This tool can be used when no TTY console is connected to the 4538 TTY port.

Minimum System Requirements

The system must be equipped with a PCI Master system board running Solaris 7 or Solaris 8.

Package Description

The 4538 Solaris software package is a standard package for the Solaris operating system. It can be installed using the **`pkgadd`** command or the software installation program `Admintool`. Installing the package creates the following files and directories:

(cont)

Table 4-1. Installation Directories and Files

File Location and Files	Description
`/usr/kernel/drv`	Drivers
`/usr/kernel/drv/iphcdrv` /usr/kernel/drv/sparcv9/iphcdrv	32-bit base driver object 64-bit base driver object (Solaris 7 or Solaris 8 only)

Table 4-1. Installation Directories and Files

File Location and Files	Description
/opt/iphdrvtoo	Package location
/opt/iphdrvtoo/tools/iphSetup	Tool to flash one of the device's read-only components
/opt/iphdrvtoo/tools/iphMonitor	PCI console to dialog with boot firmware
/opt/iphdrvtoo/script/start_iphcdrv	Script sample to load the driver
/opt/iphdrvtoo/script/stop_iphcdrv	Script sample to unload the driver
/opt/iphdrvtoo/man/man1m/iphMonitor.1m	iphMonitor man image
/opt/iphdrvtoo/man/man1m/iphSetup.1m	iphSetup man image
/opt/iphdrvtoo/man/man1m/start_iphcdrv.1m	start_iphcdrv man image
/opt/iphdrvtoo/man/man1m/stop_iphcdrv.1m	stop_iphcdrv man image

Package Installation

NOTE

You must have root permission to install the software package

Copy the Solaris package file `iphdrvtoo.tar` from the `/tools/Solaris` directory of the CD-ROM to the target Solaris machine. Then untar the package file by executing the following command:

```
tar xvf iphdrvtoo.tar
```

To install this software package, use the **`pkgadd`** command or the software installation program `Admintool`. For both programs, you will have to specify the complete path name where the package can be found, that is the directory name where you copied the package file to your local disk. The following line gives an example of use of the **`pkgadd`** command:

```
pkgadd -d <pathname> iphdrvtoo
```

Where `<pathname>` is the full path name of the directory where the package is located.

The package installation program copies all the component files to their destination directory and dynamically loads the device driver. It does not modify any system files.

After installing the package, set the `PATH` and `MANPATH` variables to access the tools, scripts, and man pages.

- For C shell, open the `~/.cshrc` file, and add the following at the end of the file:

```
set path=(/opt/iphdrvtoo/tools /opt/iphdrvtoo/script $path)
setenv MANPATH /opt/iphdrvtoo/man/:$MANPATH
```

- For Bourne shell, open the `~/.profile` file, and add the following at the end of the file:

  ```
  PATH=/opt/iphdrvtoo/tools:/opt/iphdrvtoo/script:$PATH
  MANPATH=/opt/iphdrvtoo/man:$MANPATH
  ```

- For Korn shell, open the `~/.profile` file, and add the following at end of the file:

  ```
  export PATH=/opt/iphdrvtoo/tools:/opt/iphdrvtoo/script:$PATH
  export MANPATH=/opt/iphdrvtoo/man:$MANPATH
  ```

To reread the shell's initialization files, enter the following:

- For the C shell:

  `source ~/.cshrc`

- For the Bourne shell and Korne shell:

  `. ~/.profile`

Package Uninstallation

NOTE

You must have root permission to uninstall the software package.

To uninstall the package, use the **`pkgrm`** command or the software installation program `Admintool`. The following line gives an example of use of the **`pkgrm`** command:

```
pkgrm iphdrvtoo
```

This command removes the driver from memory, and removes all the files created by the package installation.

NOTE

If you changed the `PATH` and `MANPATH` variables after installing the package, you must manually update the shell's initialization files.

Driver Load/Unload

After the package installation, the driver is automatically loaded. However two scripts are provided to manage the driver load and the driver unload.

Location

These two scripts are located in the `/opt/iphdrvtoo/script` directory.

Driver Reload

The driver can be dynamically reloaded into memory at any time.

The `start_iphcdrv` script first unloads the driver if it is already loaded, removes the device nodes, and then dynamically restarts the driver.

To run the script, enter the following command:

`/opt/iphdrvtoo/script/start_iphcdrv`

Driver Unload

The driver can be dynamically removed from memory at any time.

The `stop_iphcdrv` script dynamically removes the driver from memory and also removes the device nodes that were dynamically created by the driver when loaded. The driver can be removed only if no applications are currently using the device nodes.

To run the script, enter the following command:

`/opt/iphdrvtoo/script/stop_iphcdrv`

Using the Tools

PCI Console iphMonitor

`iphMonitor` is a utility to dialog with the Boot Firmware and display all the results you could do if you connected a monitor to the TTY port.

Location

The `iphMonitor` utility is located in the `/opt/iphdrvtoo/tools` directory.

Usage

```
iphMonitor DeviceNode [-r] [-f LogFile] [-c CmdFile]
```

`DeviceNode` is the name of the device node used to access the card (format `/dev/iph_wan_x` where `x` represents a card index).

The `-r` option can be specified to reboot the card.

The `-f` option specifies logging the display into the `LogFile` file.

The `-c` option switches `iphMonitor` to an automatic mode where all the input commands are specified in the `CmdFile file`.

To end the dialog with the Boot Firmware and close the application in interactive mode, use the **`exit`** command.

Command File Syntax

The command file is an ASCII file. It contains a list of commands that are sequentially executed by `iphMonitor`. This option is especially useful for automating hardware testing.

Each line of the command file is any command that can be interactively executed (when **`-c`** option is not specified). The only unsupported commands are all the on-line help commands (**`h`** or **`help`**).

For interactive commands that require the operator to interrupt it, a dedicated delay command is supported. The delay command specifies that the execution of a command should stop after a certain amount of time. The delay command should be specified on the line that follows immediately after the command to interrupt. It requires one argument, which is the delay value in seconds.

Example:

```
RUN
FPHY
delay 10
```

Specifies that the `iphMonitor` should execute the **`RUN`** command and then **`FPHY`** but `iphMonitor` should stop the execution of FPHY after 10 seconds.

Flash Utility iphSetup

The `iphSetup` application is used to update the serial EEPROM, the flash memory (with new Boot Firmware, operational firmware, or binary file), and the EPLD of an Interphase board.

CAUTION

Use extreme caution when using this tool, because incorrect use can put the board out of service.

Location

The flash utility is located in the `/opt/iphdrvtoo/tools` directory.

Usage

```
iphSetup devicenode [options] [filename]
```

`devicenode` is the name of the device node used to access the card to be configured (format `/dev/iph_wan_x` where `x` represents a card index).

Options:

- `-v` to verify instead of program. When a filename is given, its content is compared to the EPLD, EEPROM, or flash memory content according to the file extension.

- `-n` to disable the echo mode.
- `-r` to reboot the card without update.
- `-o` offset to specify the offset from the beginning of the memory space, where to load the code. It is an integer used for flash memory update. It can be in hexadecimal (0x0123ABCD) or decimal format (12345).

`filename` is the name of the file to load. If not specified, the **`iphSetup`** command displays information about the boot firmware that is currently flashed.

Accepted file extensions:

- `.boo` or `.elf` is for Boot Firmware or WAN Firmware to load in a device's Flash EEPROM. It is a binary ELF format file.
- `.bin` is for a non-ELF binary code to load in a device's Flash EEPROM.
- `.psp` is for code to load in the PowerSpan EEPROM. This is an ASCII file describing, in C-language (bytes array), the first 64 bytes of the EEPROM. No file for PowerSpan EEPROM update is included in the present package and this operation must be done **only under Interphase control**.
- `.isp` is for code to in -situ program the Lattice EPLD. No file for EPLD update is included in the present package and this operation must be done **only under Interphase control**

Flash EEPROM Location

The Flash EEPROM is mapped at offset 0xFF800000 in the device memory.

Examples

`iphSetup /dev/iph_wan_0 firmware.elf`

Store the WAN Firmware contained in the `firmware.elf` binary file in the Flash EEPROM of the card identified by the `/dev/iph_wan_0` device node. You do not need to specify the offset where to flash the code since `iphSetup` gets the address directly from the ELF file.

`iphSetup /dev/iph_wan_0 -o 0xFF8B0000 file.bin`

Store the binary code contained in the `.bin` binary file in the Flash EEPROM of the card identified by the `/dev/iph_wan_0` device node at offset `0xFF8B0000`.

`iphSetup /dev/iph_wan_0 -r`

Reboot the card identified by the `/dev/iph_wan_0` device node. This command can be used to reboot a card after the process for an automatic configuration is completed.

`iphSetup /dev/iph_wan_0`

Display the revision numbers for the card identified by the `/dev/iph_wan_0` device node

Regulatory Statements C

FCC

4538 Communications Controller

FCC Part 15 Regulatory Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This equipment complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This equipment may not cause harmful interference, and

(2) This equipment must accept any interference received, including interference that may cause undesired operation.

FCC Part 68 Regulatory Compliance

This T1 Communications Controller complies with Part 68 of the FCC Rules. The card includes a label that contains, among other information, the FCC registration number for this equipment. If requested, this information must be provided to the telephone company.

The FCC Digital Interface Code of this equipment is 04DU9-1SN.
The FCC Service Order Code is 6.0N.
The USOC jack for this equipment is RJ48C.

An FCC compliant telephone cable and modular plug are provided with this equipment. This equipment is designed to be connected to the telephone network or premises wiring using a compatible modular jack, which is Part 68 compliant.

This equipment cannot be used on telephone company-provided coin service. Connection to Party Line Service is subject to state tariffs.

If this equipment causes harm to the telephone network, the telephone company will notify you in advance that temporary discontinuance of service may be required. If advance notice is not practical, the telephone company will notify the customer as soon as possible. Also, you will be advised of your right to file a complaint with the FCC if you believe it is necessary.

The telephone company may make changes in its facilities, equipment, operations, or procedures that could affect the operation of the equipment. If this happens, the telephone company will provide advance notice so you can make necessary modifications to maintain uninterrupted service.

If you experience trouble with this equipment, please contact Interphase Corporation at 214-654-5000 for warranty information. If the trouble causes harm to the telephone network, the telephone company may request the removal of the equipment from the network until the problem is resolved.

No repairs may be performed by the customer. Repairs are to be made only by Interphase Corporation or its licensees. Unauthorized repairs void registration and warranty.

It is recommended that the customer install an AC surge arrester in the AC outlet to which this device is connected. This is to avoid damaging the equipment caused by local lightening strikes and other electrical surges.

EN60950–IEC950 Safety Standard

This equipment complies with the EN60950–IEC950 safety standard.

- The 4538 card must be used only in a data terminal equipment (DTE)—for example, a CompactPCI chassis. Because unsafe voltages (TNV) exist on the card, disconnect the card from the telephone lines before removing the card.
- The card must be installed in the DTE in a way that ensures at least 2 mm of air space between the PCB card and any other components in the DTE.
- Care should be exercised when installing/removing cards in adjacent slots, because unsafe voltages (TNV) also exists on those cards.

These cards are for use only with NRTL Certified personal computers that have instructions detailing installation of accessories by the user.

Canadian

Tested to Comply with Canadian Standards

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

NOTICE: This equipment meets the applicable Industry Canada Terminal Equipment Technical Specifications. This is confirmed by the registration number. The abbreviation, IC, before the registration number signifies that registration was performed based on a Declaration of Conformity indicating that Industry Canada technical specifications were met. It does not imply that Industry Canada approved the equipment.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be coordinated by a representative designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Caution: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

- Standard Connecting Arrangement: CA-48C
- The Ringer Equivalence Number: Not Applicable

European

Regulatory Information for Europe

This equipment displays the CE mark to show that it has been found to be in full compliance with the requirements of the RTTE/R&TTE Directive 1999/5/EC.