

COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

 Sell For Cash  Get Credit  Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New**, **New Surplus**, **Refurbished**, and **Reconditioned** NI Hardware.



Bridging the gap between the manufacturer and your legacy test system.

 1-800-915-6216

 www.apexwaves.com

 sales@apexwaves.com

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

Request a Quote

 **CLICK HERE**

LV-222-555-000

OmniBusII PCIe/PXIe USER'S MANUAL

INTERFACE CARD
to AVIONICS DATABUSES

February 6, 2015
Rev. B0

Copyright © 2014-2015
by
Ballard Technology, Inc.

Astronics Ballard Technology

Phone: +1.425.339.0281 800.829.1553

Email: support@ballardtech.com

www.ballardtech.com



COPYRIGHT NOTICE

Copyright © 2014-2015 by Ballard Technology, Inc. Ballard Technology's permission to copy and distribute this manual is for the purchaser's private use only and is conditioned upon purchaser's use and application with the hardware that was shipped with this manual. No commercial resale or outside distribution rights are allowed by this notice. This material remains the property of Ballard Technology. All other rights reserved by Ballard Technology, Inc.

SAFETY WARNING

Ballard products are not intended, warranted or certified for any particular use or application or for use in any application where failure of the products could lead directly to death, personal injury, or damage to property. Customers, licensees, and/or users are responsible for establishing and assuring suitability and proper use of Ballard products for their particular use or application..

INTERFERENCE

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. Operation of this equipment in a residential area is likely to cause interference in which case the user at their own expense will be responsible for taking whatever measures may be required to correct the interference.

TRADEMARKS

PCIe® is a registered trademark of PCI-SIG. PXI™ is a trademark of PXI Systems Alliance. CompactPCI® is a registered trademark of the PCI Industrial Computation Manufacturers Group. Molex® LFH™ is a trademark of Molex Inc. Windows® is a registered trademark of Microsoft Corporation. Ballard Technology®, OmniBus® and CoPilot® are registered trademarks of Ballard Technology, Inc. BTIDriver™ is a trademark of Ballard Technology, Inc. All other product names or trademarks are property of their respective owners.

Astronics Ballard Technology

Phone: +1.425.339.0281 800.829.1553

Email: support@ballardtech.com

www.ballardtech.com

ASTRONICS
BALLARD TECHNOLOGY

TABLE OF CONTENTS

1. INTRODUCTION	1-1
1.1 OmniBus II Overview	1-1
1.2 OmniBus II Configurations	1-3
1.3 Avionics Databus Protocols	1-4
1.4 Other Documentation	1-4
1.5 Support and Service	1-5
1.6 Updates.....	1-5
2. INSTALLATION	2-1
2.1 Step 1: Insert the Card into the System	2-1
2.2 Step 2: Install the Driver Software	2-2
2.3 Step 3: Set the Card Number and Test the Installation.....	2-3
2.4 Step 4: Connect to Databus(s) I/O.....	2-3
3. OPERATION	3-1
3.1 CoPilot.....	3-1
3.2 User-Developed Software	3-2
4. OMNIBUS II FEATURES	4-1
4.1 PCI Express Bus.....	4-1
4.2 Built-in Test	4-1
4.3 IRIG Timer	4-2
4.4 Core Discretes	4-3
4.5 Avionics Discretes.....	4-4
4.5.1 Shunt Inputs	4-5
4.5.2 Shunt Input Considerations.....	4-5
4.5.3 Shunt Outputs	4-5
4.5.4 Shunt Output Considerations	4-6
4.5.5 Shunt Discrete Input/Output Usage	4-7
5. OMNIBUS II PXIE SPECIFIC FEATURES	5-1
5.1 Clock Switch (CLK SEL).....	5-1
5.2 PXIe Trigger Access	5-2
5.3 Protocol Sync and Trigger Support.....	5-2
5.4 PXIe Status	5-4
5.5 Chassis Slot Glyph	5-5
6. MODULE CONFIGURATIONS	6-1
6.1 OmniBus II PCIe Boards.....	6-1
6.2 OmniBus II PXIe Boards	6-1
6.3 MIL-STD-1553 Modules	6-1

6.3.1 Software-Selectable Bus Termination	6-2
6.3.2 Configurable RT Response Time	6-3
6.3.3 Variable Transmit Amplitude	6-3
6.3.4 Zero Crossing Distortion	6-3
6.4 ARINC 429 Modules	6-4
6.4.1 Parametric Waveform.....	6-5
6.4.2 Configurable Frequency	6-5
6.4.3 Output State	6-5
6.5 ARINC 708 Modules	6-6
6.5.1 Software-Selectable Bus Termination	6-6
6.5.2 Variable Bit Length	6-6
6.5.3 Variable Transmit Amplitude	6-6
6.6 ARINC 717 Modules	6-7
6.6.1 Variable Transmit Amplitude	6-7
7. CONNECTOR PINOUTS	7-1
7.1 Interface Connector.....	7-1
7.2 General Pinout	7-1
7.3 Module-Specific Wiring	7-2
7.3.1 MIL-STD-1553	7-3
7.3.2 ARINC 429.....	7-4
7.3.3 ARINC 708.....	7-4
7.3.4 ARINC 717.....	7-6
7.4 Standard Cables	7-7
7.4.1 PN 16035 Cable Assembly: LFH to LFH.....	7-7
7.4.2 PN 16036 Cable Assembly: LFH to Two 25-pin D-sub	7-7
7.4.3 MIL-STD-1553 Cable Assemblies.....	7-9
APPENDIX A COUPLING AND TERMINATION	A-1
A.1 Bus Termination	A-1
A.2 Transformer versus Direct Coupling.....	A-1
APPENDIX B REVISION HISTORY	B-1

LIST OF FIGURES

Figure 1.1—OmniBus II PCIe card	1-1
Figure 1.2—OmniBus II PXIe card	1-2
Figure 1.3—The two-core architecture of OmniBus II PCIe/PXIe card	1-3
Figure 2.1—PXIe Compatible Chassis Glyphs	2-2
Figure 3.1—Sample CoPilot screen	3-1
Figure 4.1—Built-in test and System monitor architecture	4-2
Figure 4.2—OmniBus II Discrete Shunt Input Circuit	4-5
Figure 4.3—OmniBus II Discrete Shunt Output Circuit	4-6
Figure 5.1—PXIe On-Board Clock Switch	5-1
Figure 5.2—PXIe Chassis Slot Glyphs	5-5
Figure A.1—Transformer coupling to a dual-redundant databus	A-2
Figure A.2—Direct connection to a dual-redundant databus	A-3

LIST OF TABLES

Table 4.1—Input and Output IRIG formats	4-3
Table 4.2—Electrical characteristics of the AM IRIG signals	4-3
Table 4.3—Electrical characteristics of the PCM IRIG signals	4-3
Table 4.4—IRIG pinout	4-3
Table 4.5—Hardware versus software designation of core discretes	4-4
Table 4.6—Avionics discrete I/O designations	4-7
Table 5.1—PXIe Trigger Signals	5-2
Table 5.2—PXIe Trigger to Protocol Trigger Mapping	5-3
Table 5.3—PXIe Trigger to Protocol Sync Mapping	5-3
Table 5.4—Transitional Protocol Trigger Parameters	5-4
Table 5.5—PXIe Status Parameters	5-4
Table 6.1—OmniBus II PCIe Host Part Numbering	6-1
Table 6.2—OmniBus II PXIe/cPCIe Host Part Numbering	6-1
Table 6.3—MIL-STD-1553 Module Part Numbering	6-2
Table 6.4—MIL-STD-1553 Level Function Definition	6-2
Table 6.5—MIL-STD-1553 ParamAmplitudeConfig Configval	6-3
Table 6.6—MIL-STD-1553 ParamAmplitudeConfig Output Drive	6-3
Table 6.7—ARINC 429 Module New Applications Part Numbering	6-4
Table 6.8—ARINC 429 Module Deprecated Part Numbering	6-4
Table 6.9—ARINC 429 Parametric Waveform Characteristics	6-5
Table 6.10—MIL-STD-1553 Module Part Numbering	6-6
Table 6.11—ARINC 708 ParamAmplitudeConfig Configval	6-7
Table 6.12—ARINC 717 Module New Applications Part Numbering	6-7
Table 6.13—ARINC 717 Module Deprecated Part Numbering	6-7
Table 7.1—General pin designations	7-2
Table 7.2—Pinouts for MIL-STD-1553 modules	7-3
Table 7.3—Pinouts for ARINC 429 modules	7-4
Table 7.4—Pinouts for ARINC 708 modules	7-5
Table 7.5—Pinouts for ARINC 717 modules	7-6
Table 7.6—Wiring chart for 16036 cable assembly	7-8
Table 7.7—MIL-STD-1553 cable assembly configurations	7-9
Table 7.8—Twinax wiring on MIL-STD-1553 cable assemblies	7-9
Table 7.9—D-sub connector pinout for cable assemblies 16037 and 16039	7-10

This page intentionally blank.

1. INTRODUCTION

This manual is the user’s guide for PCIe® and PXI™ Express models of Ballard Technology’s OmniBus® II family of products. Throughout this manual any reference to the PCIe card applies to the OmniBus II PCIe card, reference to the PXIe card applies to the OmniBus II PXIe card, and references to OmniBus and OmniBus II apply generically to all the products in the designated family. This guide gives the background for the OmniBus II PCIe/PXIe card, discusses special features, describes the installation process, and references programming alternatives.

1.1 OmniBus II Overview

OmniBus is a family of products that enable computer systems to communicate with avionics databuses for the purpose of testing, simulation, and/or operation. Each OmniBus unit can support more than one protocol and a large number of channels. They are available as an interface card for popular computer standards (PCIe, cPCIe/PXIe, etc.) and as a stand-alone bridge to other communications protocols (USB, Ethernet, etc.). All common avionics databus protocols are supported, including MIL-STD-1553, ARINC 429, ARINC 708, and ARINC 717. Other protocols (such as ARINC 575, ARINC 573, ARINC 453, etc.) are also supported. Custom protocols are implemented upon request.

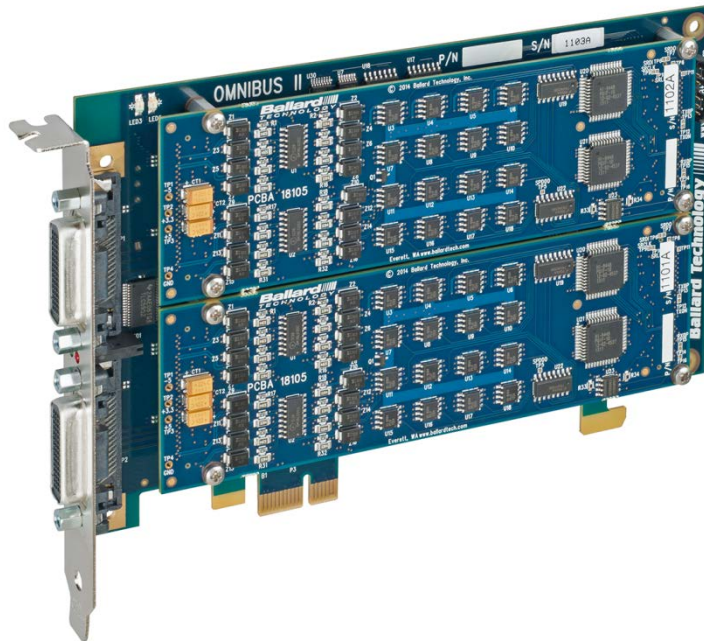


Figure 1.1—OmniBus II PCIe card

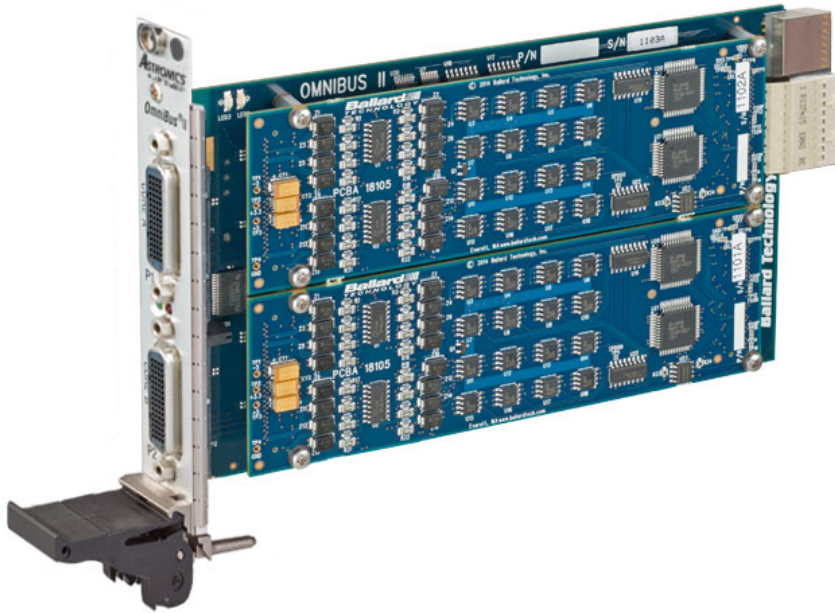


Figure 1.2—OmniBus II PXIe card

The high-density modular design of the OmniBus family provides flexibility that enables the user to select from many protocol, platform, and channel count combinations. Each OmniBus product can have at least two modules, and each module has its own circuitry to handle the channels and protocols associated with it. The high channel count and mixed protocol capabilities can be fully exploited without the risk of overloading the host computer's processor. IRIG and special timing circuits allows channels, boards, and computers to be synchronized in time to each other and to external devices.

OmniBus II is a newer, enhanced generation of the original OmniBus architecture, and though there are similarities, components of the two generations are not interchangeable. Figure 1.3 illustrates the modular architecture of the OmniBus II PCIe/PXIe board.

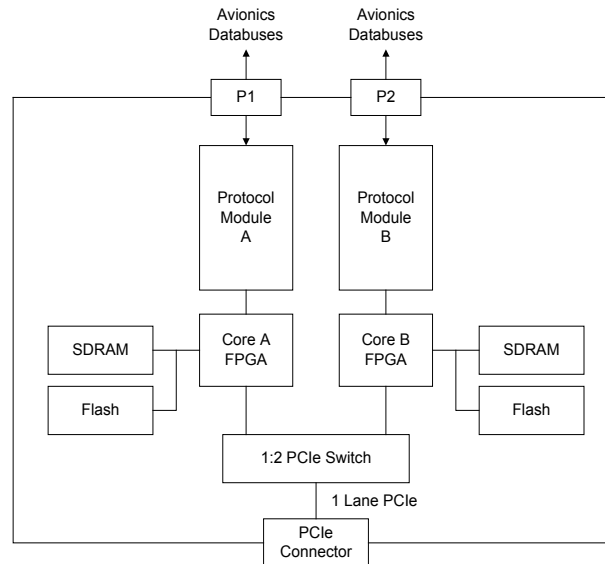


Figure 1.3—The two-core architecture of OmniBus II PCIe/PXIe card

The easiest way to operate OmniBus products is with CoPilot®, Ballard Technology’s databus analyzer and simulation software. Alternately, software developers can write their own software applications using the included BTIDriver™ API (Application Program Interface).

1.2 OmniBus II Configurations

The OmniBus family includes products with many different host platform, protocol, and channel count combinations. A given OmniBus part number is produced in the factory by mounting protocol-specific modules on the required host platform.

Note: OmniBus products are not user-configurable. Do not attempt to swap one module type for another one with a different part number. If a module is exchanged, it must be with an identical module. OmniBus products may be upgraded with additional channels or protocols, but this must be done at the factory.

The assembly part number characterizes the configuration of an OmniBus product. The assembly part number is designated by groups of characters separated by dashes. The first group of characters in the assembly part number is the part number of the main board (eg. 212 for a two module PCIe card), the second group is the part number of the module in the Core A position, and the third group is the part number of the module in the Core B position. A more detailed description of the individual part numbers may be found in Chapter 6.

The complete assembly part number is printed on the main OmniBus board. If the modules are visible, each group of numbers in the assembly configuration that represent modules should match the part numbers printed on the corresponding modules. The configuration of an installed OmniBus product may be determined by running the test program described in Chapter 4.

For future reference, we encourage you to record the assembly part number and serial number of your OmniBus product. You may wish to use the space provided below:

Assembly PN: _____ - _____ - _____
(Board PN) (Core A PN) (Core B PN)

Serial No: _____

1.3 Avionics Databus Protocols

Avionics databuses interconnect various electronic equipment (navigation, controls, displays, sensors, etc.) on an aircraft, much as a local area network (LAN) interconnects computers in an office. Data from one device is passed over the network to other devices that need it. There are a number of military and commercial avionics databus standards. OmniBus products support the most common protocols, which are briefly described below:

- **MIL-STD-1553** is the protocol for military aircraft and other military and commercial applications. It is a digital, command-response, time-division multiplexing databus protocol.
- **ARINC 429**, one of the most prevalent ARINC (Aeronautical Radio INCorporated) standards, defines the transfer of digital data between commercial avionics systems. It uses broadcast bus topology and a label identification method for data words. **ARINC 575** is the specification for a Digital Air Data System (DADS). ARINC 575 includes a databus protocol almost identical to ARINC 429.
- **ARINC 708** defines an airborne pulse Doppler weather radar system for commercial aircraft. The Transmitter/Receiver unit sends data over the 708 display databus to the Control/Display Unit. Data consists of 1600-bit words that are preceded and followed by a sync. The display databus is an adaptation of the proposed, but never approved, **ARINC 453** databus.
- **ARINC 717** includes the databus protocol for interconnecting the Digital Flight Data Acquisition Unit (DFDAU) and the Digital Flight Data Recorder (DFDR). Data words are 12 bits long and are nominally transmitted at 64 or 256 words per second in subframes, frames, and sometimes superframes. **ARINC 573**, an older equipment specification for flight data recorders, uses a databus similar to 717.

These and other standards are not limited to use in aircraft. They are used in many other military and industrial applications such as surface and space vehicles, process control, nuclear research, and oil exploration.

1.4 Other Documentation

Besides this manual, Ballard provides other documentation to facilitate operation of the OmniBus interface. These include protocol manuals, information on the software distribution disk, and CoPilot documentation.

Separate BTIDriver API programming manuals are available for each avionics protocol. These manuals provide information on the specific protocol and include basic and advanced programming instructions for users who intend to write their own software. They also contain a comprehensive reference for each function.

The software distribution disk accompanying the OmniBus has example programs, drivers, and driver installation instructions for various operating systems, and other information, files, and resources.

1.5 Support and Service

Ballard Technology offers technical support before and after purchase. Our hours are 9:00 AM to 5:00 PM Pacific Time, though support and sales engineers are often available outside those hours. We invite your questions and comments on any of our products. You may reach us by telephone at (800) 829-1553 or (425) 339-0281, by fax at (425) 339-0915, on the Web at www.ballardtech.com, or through e-mail at support@ballardtech.com.

1.6 Updates

At Ballard Technology, we take pride in high-quality, reliable products that meet the needs of our customers. Because we are continually improving our products, periodic updates to documentation and software may be issued. For access to the most current information and software, please register for a support account on our website.

This page intentionally blank.

2. INSTALLATION

This chapter explains the procedures for installing an OmniBus II PCIe/PXIe card. There are four steps to installation:

Step 1: Insert the Card into the System

Step 2: Install the Driver Software

Step 3: Set the Card Number and Test the Installation

Step 4: Connect to Databus(s) I/O

After the installation steps are completed, the PCIe/PXIe card is ready to communicate on the databus(es).

WARNING

Static Discharge

As with most electronic devices, static discharge may damage or degrade components on a circuit card. When handling a circuit card, the user should be grounded (e.g., through a wrist strap). Each circuit card is shipped in an anti-static bag, and should be stored in a similar container when not installed in the system.

2.1 Step 1: Insert the Card into the System

In an ESD safe environment, do the following:

For a PCIe System:

- Shut down the system.
- Insert the card into an empty PCIe slot.
- Secure the card with a screw.
- Restart the system.

For a PXIe System:

- Shut down the system.
- On the card, select the required clock by sliding the on-board switch left or right.
- With the injector handle in the down position, insert the card into an empty chassis slot marked with one of the following PXIe Chassis Glyphs:

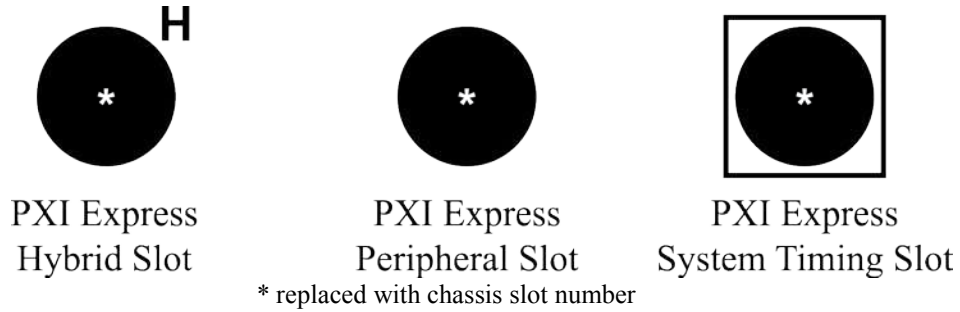


Figure 2.1—PXIe Compatible Chassis Glyphs

- While pressing the bottom of the handle against the horizontal rail of the subrack, move the injector handle up to lock the card in place.
- Secure the card with the screw located at the top of the front panel.
- Restart the system.
- If the system fails to boot, the clock select or chassis slot type may be incorrectly configured. For more information, refer to Chapter 5.

2.2 Step 2: Install the Driver Software

Drivers allow programmatic control of the PCIe/PXIe card(s) from a host computer.

The driver installation procedures vary depending on your computer’s operating system. These procedures are kept on disk so they can be easily updated as operating systems evolve. Before proceeding with the installation, find, print, and review the driver installation procedure for your operating system.

To install the driver software:

- Insert the driver disk in your drive and browse to the folder for your product
- Print the driver installation instructions located in a README file on the software distribution disk in a folder specific to your product
- Follow the instructions from the README file

The installation procedure differs for each operating system, but in most cases, several files are copied to the host computer system and either the system registry is modified or configuration files are created.

If you encounter problems, have installation questions, or cannot find instructions for your operating system, please contact Customer Service (see Section 1.5).

2.3 Step 3: Set the Card Number and Test the Installation

You must set a card number on the controlling computer for software to uniquely identify each PCIe card.

Since many Ballard BTIDriver™-compliant hardware devices (e.g., OmniBus II PCIe/PXIe cards) can be concurrently connected to the same computer, software running on a given computer uses a unique card number to designate which hardware device is being accessed. If you have only one BTIDriver-compliant device connected to the computer, it is recommended that you set it to card number 0 because the example programs included with the driver software assume a card number of 0. After the card number has been set, you can then test the PCIe/PXIe card.

In Windows, the BTITST32.EXE test program can be used to assign and manage card numbers and to test the PCIe/PXIe card (and other BTIDriver-compliant devices). The test program discovers all connected BTIDriver-compliant devices and displays important information about each device such as card number, configuration, serial number, and assembly part number. Running the test sequence verifies both the device hardware and the interface between the device and the computer. If the program does not detect any faults, it displays a “passed test” message.

This program and a README file with instructions are found in the Windows Test subfolder on the software distribution disk:

PCIe: OMNIBUS→DRIVERS→OMNIBUS_II_PCIE→TEST

PXIe: OMNIBUS→DRIVERS→OMNIBUS_II_PXIE→TEST

Note: The Windows test program may be used at any time to determine or reassign the card number.

If you need further assistance, contact Ballard Technology’s Technical Support (see Section 1.5).

2.4 Step 4: Connect to Databus(s) I/O

Connect the databuses to the PCIe/PXIe card according to the pin assignment tables in Chapter 7. Be sure to follow the coupling and termination guidelines discussed in Appendix A.

Connection of the ground pin(s) to the end system(s) is necessary for proper operation of ARINC 429, ARINC 717 and the discrete I/O, and is recommended for MIL-STD-1553. There is no need to terminate unused signals, and do not connect the reserved pin(s).

This page intentionally blank.

3. OPERATION

Software is used to control OmniBus products and to manipulate data. Whether you use Ballard’s CoPilot® software or develop your own applications using Ballard’s BTIDriver™ API library, it is easy to operate the PCIe/PXIe card and to utilize its powerful interface.

3.1 CoPilot

A PC with CoPilot and Ballard’s OmniBus II PCIe/PXIe card makes a powerful, low-cost databus analyzer/simulator. CoPilot interfaces directly with OmniBus products, eliminating the need to write custom software. CoPilot greatly simplifies such tasks as defining and scheduling bus messages and capturing and analyzing data. CoPilot is a Windows-based program that features a user-friendly GUI (Graphical User Interface) and many timesaving features. For example, bus messages can be automatically detected, posted in the hardware tree, and associated with the appropriate attributes from the database of equipment, message, and engineering unit specifications.

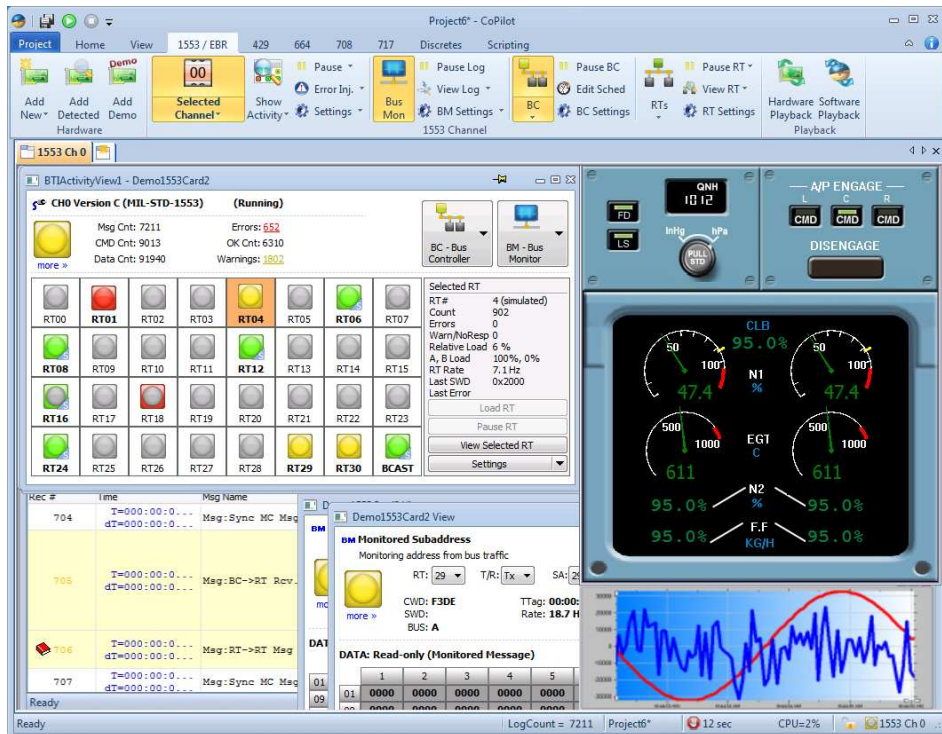


Figure 3.1—Sample CoPilot screen

CoPilot users can quickly configure, run, and display the activity of multiple databuses in a unified view. Data can be observed and changed in engineering units while the bus is running. The Strip View graphically illustrates the history of the selected data values. Data can also be entered and viewed as virtual instruments (knobs, dials, gauges, etc.) that can be created by the user or automatically generated by dragging and dropping an item into the Control View window.

Because CoPilot can host multiple channels and databus protocols in the same project, it is the ideal tool for operating OmniBus products. CoPilot can be purchased separately or with an OmniBus product. For more information or a free evaluation copy, call Ballard at (800) 829-1553. In addition, you can learn more about the latest version of CoPilot at www.ballardtech.com.

3.2 User-Developed Software

Software developers can use the bundled BTIDriver API to create custom applications. With only a few function calls, a program can operate OmniBus products and process messages to and from the avionics databus. Functions include routines for transmitting, receiving, scheduling, recording, data manipulation, and time-tagging bus messages. Although most tasks require only a few API calls, the comprehensive library includes a broad range of functions for specialized needs.

Sample programs and utility programs are included with the API on the software distribution disk. Detailed information about each API function and instructions on programming for OmniBus products is found in separate manuals for each protocol (e.g., *ARINC 429 Programming Manual for BTIDriver-Compliant Devices*).

4. OMNIBUS II FEATURES

This section describes special capabilities and interface signals available on many OmniBus II products. Some of these features (such as IRIG time) are on all models and others (such as avionics discretes) are only on a subset. If you need more information than is presented here, please contact Customer Support at Ballard Technology for assistance (see Section 1.5).

4.1 PCI Express Bus

Both OmniBus II PCIe and OmniBus II PXIe cards interface through a PCI Express interface with the following features:

- PCI Express single-lane (x1) Endpoint
- Full 2.5 Gbps per direction
- PCIe bus mastering

4.1.1 PCIe Form Factor

The OmniBus II PCIe card conforms to the PCI Express add-in card form factor. The card utilizes a single lane (x1) PCIe interface which allows the PCIe card to be installed in any lane-width connector (e.g. x1, x8, x16). All power is supplied through the PCIe interface with no external power source required.

4.1.2 PXIe Form Factor

The OmniBus II PXIe card conforms to the PXI Express Type 2 Peripheral Board form factor. The card utilizes a single lane (x1) PCIe interface through the standard PXIe x8 connector. All power is supplied through the PXIe connectors with no external power source required.

For more PXIe information, refer to Chapter 5.

4.2 Built-in Test

The OmniBus II family includes on-board circuitry to test card operation and provide health information to the user application. Three types of Built-in testing are supported: Power-on, Initiated, and Continuous. Below is a summary of each of the tests, but for more information on the API interaction please refer to the example programs provided on the driver disk or to one of the API manuals (e.g., *MIL-STD-1553 Programming Manual for BTIDriver-Compliant Devices*).

4.2.1 Power-on Built-in Test (PBIT)

Immediately after power-on, hardware verifies the FPGA configuration and writes, reads, and verifies multiple patterns to the entire on-board RAM; this verifies memory operation prior to use. In the event of a PBIT error, opening the card will fail.

4.2.2 Initiated Built-in Test (IBIT)

A comprehensive hardware test may be initiated by user developed software. In addition to the same memory tests as PBIT, IBIT verifies a range of internal

hardware and host communication. Initiated test resets the card and is not intended to be performed while the card is configured or running.

4.2.3 Continuous Built-in Test (CBIT)

During card operation, dedicated hardware constantly monitors multiple internal modules for errors. These sources include error detection/correction circuits for system memory and FPGA, protocol specific tests, and the system monitor (see Figure 4.1). Status of these tests may be accessed through the API via polling or interrupts.

The system monitor polls temperature data from sensors located on the card. Present values, as well as historic minimum and maximum values, can be read from the card. The API allows for setting of user temperature limits and enabling notification of exceeded limits. If temperature sensor measurements exceed built-in system limits, card operation will be halted to protect the system.

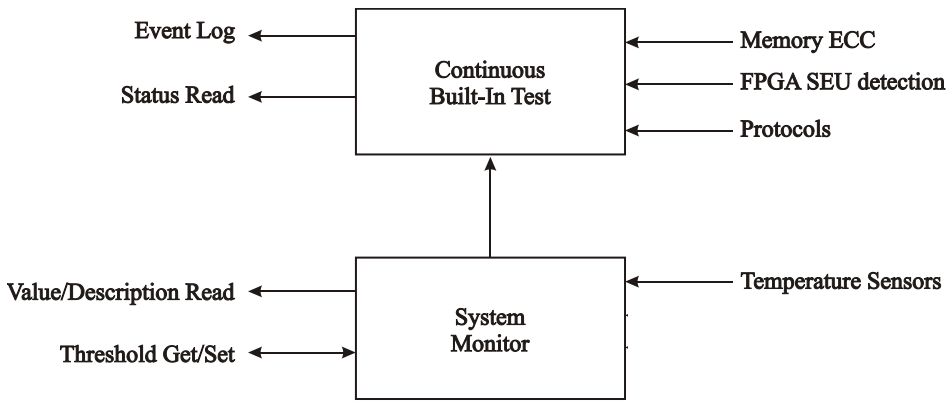


Figure 4.1—Built-in test and System monitor architecture

4.3 IRIG Timer

An IRIG time signal contains a human-readable binary coded decimal (BCD) time value in days, hours, minutes, seconds, etc. and can be used to synchronize many devices. This allows timing data from all compatible devices to be easily correlated.

The IRIG circuit can be configured as either a master or a slave. The IRIG timer pin is driven by the bidirectional buffer only when the IRIG timer is configured as a master. When the IRIG timer is configured as a slave, it will expect the IRIG signal to come from an external device.

The OmniBus II devices internally use a binary system timer that is free running and keeps time until either set by software or synchronized to an IRIG signal when configured as a slave. This system timer is also the source for the IRIG interface when configured as a master. The system timer has a resolution down to one nanosecond.

IRIG data can be encoded using Pulse Code Modulation (PCM), Modified Manchester Modulation, or Amplitude Modulation (AM). The OmniBus II family supports PCM master, PCM slave, AM master, and AM slave modulation modes.

There are a number of formats for IRIG timing. The OmniBus II family uses the IRIG formats indicated in Table 4.1. The characteristics of the external electrical interface to the IRIG pins are as shown in Table 4.2 and Table 4.3.

Format	A		1000 pps	
	B		100 pps	
Modulation Frequency	0	1	Pulse width coded	Amplitude modulated
Frequency/Resolution	0	2 3	No carrier/index count interval	1 kHz/1 ms (B only) 10 kHz/1 ms (A only)
Coded Expressions	0, 1, 2, 3, 4, 5, 6, 7		BCD _{TOY} and BCD _{YEAR} fields are supported System Timestamp only reflects BCD _{TOY} CF can be set to or read from a register SBS are ignored on input and 0 on output	

Table 4.1—Input and Output IRIG formats

AM IRIG Input Characteristics	
Min input impedance (at 1kHz)	10 kΩ
Max input amplitude (V _{pk-pk})	8V
AM IRIG Output Characteristics	
Output mark amplitude (V _{pk-pk})	2.5V to 3.5V
Output space amplitude (V _{pk-pk})	0.75V to 1.25V
Max output resistive load	45Ω

Table 4.2—Electrical characteristics of the AM IRIG signals

Input impedance (min)	12 kΩ
Input voltage (max)	-7.5V to 12.5V volts
Input level threshold	API Controlled (0V-5V)
Output level	0 to 5 volts
Output drive capability	20 mA

Table 4.3—Electrical characteristics of the PCM IRIG signals

The following table lists the protocols supported on each pin. For information on configuring and using the IRIG timer consult the BTIDriver API programming manuals.

Note: The timing pins on the OmniBus II family are distinct and not internally connected as they were in the OmniBus family.

Timing Protocol	Core/Pin
PPS	Core A Pin 17
PCM IRIG	Core A Pin 17
AM IRIG	Core B Pin 17

Table 4.4—IRIG pinout

4.4 Core Discretes

OmniBus products have both input and output discrete capability. OmniBus II has eight bidirectional TTL level discretes per core which can be used as either inputs or outputs. All OmniBus II core discrete inputs and outputs are TTL level. Each discrete output line has a 5-volt TTL driver which can source or sink up to 8 mA and has an individual tristate control; the discrete input receiver is a 5-volt tolerant device with high input impedance (10μA leakage current). When used as

an output, the status of a core discrete output can be verified by reading the input. At power on all core discrete outputs are tristated and are enabled by writing to the output or by explicitly enabling it.

Ballard's BTIDriver API provides functions to read, write, and enable (tristate) the core discrettes. The parameter *dionum* in the API functions (BTICard_ExtDIORd, BTICard_ExtDIOWr, BTICard_ExtDIOEnWr, etc.) specifies which discrete to read or write. Table 4.5 shows the mapping between the external hardware pin and *dionum*.

A specific sync or trigger can use one or more of the designated core discrettes. After a core discrete has been allocated as a trigger or sync using the enable and mask parameters in a sync or trigger define API function, the line may no longer be used as a discrete output or input. More than one core discrete, each with an individually specified polarity, may be used in combination to define a sync or trigger state. For instance, a trigger may be defined as a particular state of only one input, or it may be defined as a particular combination of two or three trigger inputs. Other triggers and syncs may use the same or different combinations of these lines. Refer to the BTIDriver software manuals for more information on programming these discrettes and their use as syncs and triggers.

Table 4.5 below shows the correlation between *dionum*, the output pin, and its hardware reference designator. The last column shows which of these discrettes may be used as trigger inputs or sync outputs in the BTIDriver API functions. The names for core discrettes are prefixed by CDIO (e.g., CDIO2).

Pin Name	LFH Pin	API <i>dionum</i>	Trigger/Sync Usage
CDIO0	11	1	Trigger A
CDIO1	21	2	Trigger B
CDIO2	51	3	Trigger C
CDIO3	41	4	-
CDIO4	13	5	Sync A
CDIO5	19	6	Sync B
CDIO6	49	7	Sync C
CDIO7	43	8	-

Table 4.5—Hardware versus software designation of core discrettes

Processes that are configured to be triggered by an external trigger can be triggered through software using the BTICard_CardTriggerEx function. This is useful for software testing and does not require external trigger equipment.

4.5 Avionics Discrettes

The OmniBus II MIL-STD-1553 and ARINC 708 modules (see Sections 6.2 and 6.5) have 16 avionics shunt discrettes which can be used to signal and detect events, determine status, and to drive loads. Each avionics shunt discrete I/O pin is configured as both a shunt input and a shunt output. When used as an output, the status of a pin can be verified by reading the input for the pin.

4.5.1 Shunt Inputs

A shunt input circuit pin is pulled up to a voltage source through a resistor. A load resistance applied between the pin and ground will shunt current from the source and generate a voltage at the pin. The pin voltage is compared with a reference voltage for input state detection. There are two defined states: the “Open” state in which a high impedance is applied to the pin, and the “Ground” state in which a low impedance is applied to the pin.

The OmniBus II discrete shunt input circuit, illustrated in Figure 4.2, has a 9 k Ω pull-up resistor to a 5 volt source. An isolation diode provides protection against over-voltage at the pin. A load resistance connected between the input pin and ground will shunt current from the 5 volt source, through the forward biased diode and the 9 k Ω resistor. A series resistor limits current as a voltage is generated across the load which is compared to a reference voltage produced by an identical configuration. This results in a 3.25 volt switching voltage.

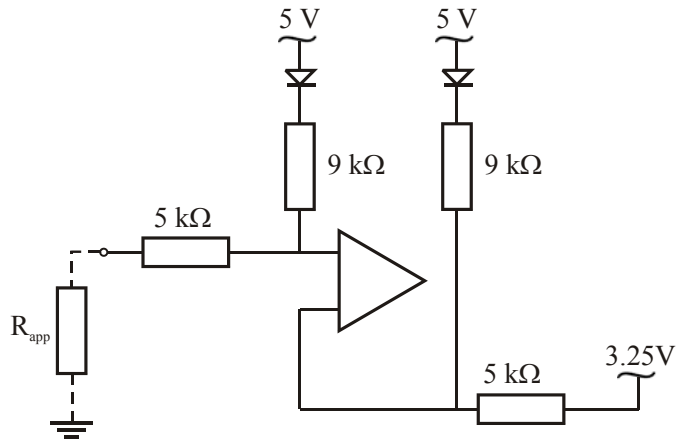


Figure 4.2—OmniBus II Discrete Shunt Input Circuit

4.5.2 Shunt Input Considerations

Limits: The OmniBus II discrete shunt inputs can withstand up to 35 VDC applied to the pin. The discrete inputs are capable of interfacing with industry standard avionics discrete signals.

4.5.3 Shunt Outputs

A shunt output is typically an open-collector circuit and is normally high impedance. When driven, the output sinks current to ground in a low impedance state. Shunt outputs can be used to communicate with an input discrete and/or to energize a load.

The OmniBus II discrete shunt output circuit, illustrated in Figure 4.3, is a low side switch capable of sinking up to 200 mA of current to ground through the external load (Z_{ext}). A diode protection circuit permits safe switching of inductive loads. Over-load detection limits the sink current and shuts the device down in an over-temperature condition. Each output circuit is wired in parallel with an input circuit (not shown) providing self-monitor capabilities.

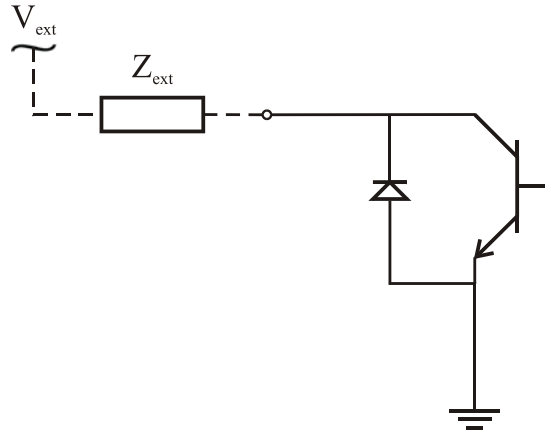


Figure 4.3—OmniBus II Discrete Shunt Output Circuit

4.5.4 Shunt Output Considerations

Limits: The OmniBus II discrete shunt outputs are open-ground switches capable of sinking up to 200mA. The discrete outputs can withstand up to 35 VDC and are capable of interfacing with industry standard avionics discrete signals.

Self-Monitor: The OmniBus II discrete shunt output circuits can be monitored by corresponding OmniBus II discrete input circuits. Writing to a discrete can drive an enabled output, and reading from that discrete, reports the current state of the input. Once the output is driven, there is a finite period of time before the change of state on the corresponding input is detected. This delay (approximately 30 μ s) is due to the latency of the host system and the analog delay of the input and output circuitry.

Over-Load/Fault Reporting: The OmniBus II discrete shunt output circuits contain current limiting and thermal shutdown features. If a user attempts to sink too much current through an output discrete circuit, the output will begin current limiting. This is accomplished by increasing the resistance through the output, which causes the power dissipation and therefore the temperature to increase. The output continues to limit the current until the thermal limit is reached and then the output is automatically shut down. Once an output is shut down due to a fault, the output remains disabled until both the fault is cleared and the user drives the output again. For this reason, it is important that the user corrects fault conditions before attempting to drive the output.

High Current Drive: Each shunt output is capable of sinking up to 200 mA of current. However, the user can wire multiple outputs in parallel to increase the maximum current sinking capability.

Power-On: After power-on, the shunt discrete I/O is in its default state with outputs open (high impedance).

4.5.5 Shunt Discrete Input/Output Usage

Ballard Technology's BTIDriver API provides functions to read and write the discrettes. The parameter *dionum* in the API functions (BTICard_ExtDIORd and BTICard_ExtDIOWr) specifies which discrete to read or write.

Table 4.6 below shows the correlation between *dionum* and its hardware reference designator (i.e., ADIOn).

Name	Description	LFH Pin#	LFH Pin Name	API <i>dionum</i>
ADIO0	Avionics DIO 0	7	BUS4P	17
ADIO1	Avionics DIO 1	6	BUS4N	18
ADIO2	Avionics DIO 2	24	BUS5P	19
ADIO3	Avionics DIO 3	25	BUS5N	20
ADIO4	Avionics DIO 4	9	BUS6P	21
ADIO5	Avionics DIO 5	8	BUS6N	22
ADIO6	Avionics DIO 6	22	BUS7P	23
ADIO7	Avionics DIO 7	23	BUS7N	24
ADIO8	Avionics DIO 8	37	BUS12P	25
ADIO9	Avionics DIO 9	36	BUS12N	26
ADIO10	Avionics DIO 10	54	BUS13P	27
ADIO11	Avionics DIO 11	55	BUS13N	28
ADIO12	Avionics DIO 12	39	BUS14P	29
ADIO13	Avionics DIO 13	38	BUS14N	30
ADIO14	Avionics DIO 14	52	BUS15P	31
ADIO15	Avionics DIO 15	53	BUS15N	32

Table 4.6—Avionics discrete I/O designations

This page intentionally blank.

5. OMNIBUS II PXIE SPECIFIC FEATURES

This chapter describes features available only on OmniBus II PXIe products. OmniBus II PXIe is a Type 2 Compact PCI Express (cPCIe) card with eXtensions for Instrumentation (PXIe). As such, PXIe cards can be used in either cPCIe or PXIe systems.

5.1 Clock Switch (CLK SEL)

The OmniBus II PXIe protocol circuitry can use either the on-board 100MHz oscillator or the PXIe connector's PXIe_CLK100 input. Selecting the on-board oscillator (OSC) allows the PXIe card to be installed in a cPCIe system which does not support the PXIe_CLK100 input. The clock can be selected (prior to applying power) by sliding the yellow switch (SW1) as shown below. The switch setting is sampled once at power-on to guarantee a stable clock selection value.

Note: The yellow switch is not a jumper, thus the switch cannot be removed from the card. Refer to Section 5.4 for more information about reading the clock switch status.

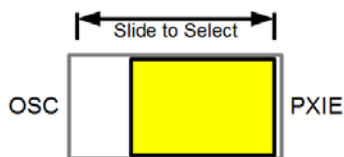


Figure 5.1—PXIe On-Board Clock Switch

If the card does not have a valid clock due an incorrect switch setting, the software will return ERR_NOCLK when accessing the card. Incorrectly setting the clock switch may cause the host computer to freeze when the card is accessed.

5.2 PXIe Trigger Access

The following table shows the PXIe Trigger signals supported by the OmniBus II PXIe card. All signals are asynchronous to either CLK10 or CLK100. PXIe Trigger signals can be accessed through the `BTICard_ExtDIORd`, `BTICard_ExtDIOWr`, and `BTICard_ExtDIOEnWr` functions by passing in the associated *dionum* per Table 5.1. These signals can also be monitored using `BTICard_ExtDIOMonConfig` with *banknum* and *rise_edge/fall_edge* bits per Table 5.1. For more information on these functions, refer to the API manuals (e.g., *MIL-STD-1553 Programming Manual for BTIDriver-Compliant Devices*).

Signal	Direction	API dionum	API banknum	API rise_edge / fall_edge
PXI_TRIG[0]	I/O	33	2	0x0001
PXI_TRIG[1]	I/O	34	2	0x0002
PXI_TRIG[2]	I/O	35	2	0x0004
PXI_TRIG[3]	I/O	36	2	0x0008
PXI_TRIG[4]	I/O	37	2	0x0010
PXI_TRIG[5]	I/O	38	2	0x0020
PXI_TRIG[6]	I/O	39	2	0x0040
PXI_TRIG[7]	I/O	40	2	0x0080
PXIe_DSTARA	In	41	2	0x0100
PXIe_DSTARB	In	42	2	0x0200
PXI_STAR	I/O	43	2	0x0400
PXIe_DSTARC	Out	44	2	0x0800

Table 5.1—PXIe Trigger Signals

All bi-directional (I/O) signals default to tri-state mode at power-on. They can be tri-stated, driven high, or driven low with `BTICard_ExtDIOEnWr`. To drive the signal, set *dioen* to TRUE. To tri-state the signal, set *dioen* to FALSE. Note that *dioval* is ignored when tri-stating with this function.

Only one core can drive a particular bi-directional signal at a time. If both cores are driving the same trigger, the value most-recently written will be driven on the trigger. For example, if Core A is configured to drive `PXI_TRIG[0]`, then Core B is configured to drive the same `PXI_TRIG[0]`, the value written to Core B will be driven on `PXI_TRIG[0]`. Ensure that other cards in the system do not cause a conflict by driving a bi-directional signal at the same time as the PXIe card.

`PXIe_DSTARC` behaves differently than the other PXIe trigger signals. When `PXIe_DSTARC` is enabled, the PXIe card drives a continuous 10MHz signal at 50% duty-cycle on this trigger. The `PXIe_DSTARC` 10MHz output can be enabled by calling `BTICard_ExtDIOEnWr` on *dionum* 44 with *dioen* set to TRUE, and disabled by calling `BTICard_ExtDIOEnWr` on *dionum* 44 with *dioen* set to FALSE. Note that this signal cannot be tri-stated.

5.3 Protocol Sync and Trigger Support

Protocols with Sync and Trigger support (e.g. BTI1553) can interface with the PXIe Trigger signals by using the *SyncDefine* or *TriggerDefine* functions (e.g. `BTI1553_BCSyncDefine`, `BTI429_ChTriggerDefine`, etc). Refer to Table 5.2 for mappings between protocol triggers (Trigger A-C) and PXIe Triggers (PXI*).

Refer to Table 5.3 for mappings between protocol syncs (Sync A-C) and PXIe Triggers (PXI*).

Protocol Trigger	PXIe Signal	Trigger Mask Parameter
A	PXI_TRIG[0]	TRIGMASK_PXITRIGA
	PXIe_DSTARA	TRIGMASK_PXISTARA
B	PXI_TRIG[1]	TRIGMASK_PXITRIGB
	PXIe_DSTARB	TRIGMASK_PXISTARB
C	PXI_TRIG[2]	TRIGMASK_PXITRIGC
	PXI_STAR	TRIGMASK_PXISTARC

Table 5.2—PXIe Trigger to Protocol Trigger Mapping

Protocol Sync	PXIe Signal	Sync Mask Parameter
A	PXI_TRIG[0]	SYNCMASK_PXITRIGA
B	PXI_TRIG[1]	SYNCMASK_PXITRIGB
C	PXI_TRIG[2]	SYNCMASK_PXITRIGC
	PXI_STAR	SYNCMASK_PXISTARC

Table 5.3—PXIe Trigger to Protocol Sync Mapping

Triggers are mutually exclusive; however, syncs are not. For example, only PXI_TRIG[0] or PXIe_DSTARA can be configured at one time to trigger the protocol logic; however, the protocol logic can drive the sync out to both PXI_TRIG[2] and PXI_STAR at the same time.

The OmniBus II PXIe also extends the protocols' *TriggerDefine* function capabilities by adding transitional trigger parameters. These new parameters allow the protocol to be triggered on a high, low, rising, or falling state of the assigned protocol trigger (A-C). To use this feature, pass the respective parameter found in Table 5.4 to the *pinpolarity* argument of the *TriggerDefine* function (e.g. BTI1553_BCTriggerDefine).

Trigger	API pinpolarity	Description
A	TRIGPOL_TRIGAL	Sets active low polarity for trigger line A
	TRIGPOL_TRIGAH	Sets active high polarity for trigger line A
	TRIGPOL_TRIGAF	Sets active on falling edge of trigger line A
	TRIGPOL_TRIGAR	Sets active on rising edge of trigger line A
B	TRIGPOL_TRIGBL	Sets active low polarity for trigger line B
	TRIGPOL_TRIGBH	Sets active high polarity for trigger line B
	TRIGPOL_TRIGBF	Sets active on falling edge of trigger line B
	TRIGPOL_TRIGBR	Sets active on rising edge of trigger line B
C	TRIGPOL_TRIGCL	Sets active low polarity for trigger line C
	TRIGPOL_TRIGCH	Sets active high polarity for trigger line C
	TRIGPOL_TRIGCF	Sets active on falling edge of trigger line C
	TRIGPOL_TRIGCR	Sets active on rising edge of trigger line C

Table 5.4—Transitional Protocol Trigger Parameters

5.4 PXIe Status

The OmniBus II PXIe provides access to some useful PXIe card status information. The desired status is selected by passing in one of the parameters found in Table 5.5 to *infotype* in BTICard_PXIStatus.

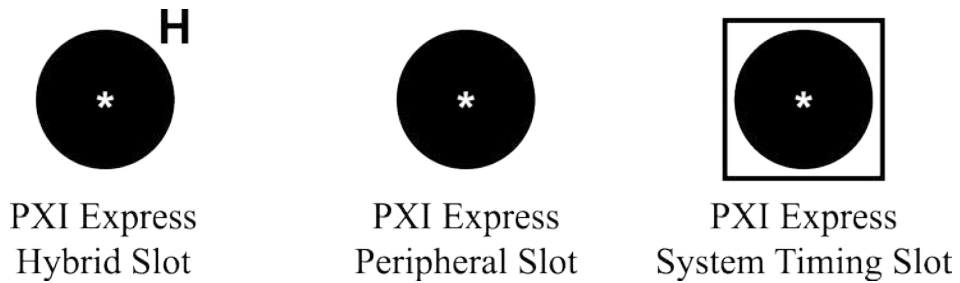
infotype	infoptr
PXITYPE_GEOADDR	Geographical Address
PXITYPE_CLKSEL	PXIe Clock Selection (Refer to Section 5.1) 0 = PXIe_CLK100 1 = On-Board 100MHz Oscillator
PXITYPE_TRIGVERS	Version of the BTI PXIe Trigger Engine
PXITYPE_OUTEN	Bitmask of Output Enables for dionums 33 to 48 (Refer to Section 5.3 for more information)

Table 5.5—PXIe Status Parameters

5.5 Chassis Slot Glyph

The PXIe card can be installed in a “PXI Express Hybrid Slot”, a “PXI Express Peripheral Slot”, or a “PXI Express System Timing Slot”. These slot types will be marked on the system chassis with one of the three Glyphs from * replaced with chassis slot number

Figure 5.2. Note that, while the PXIe card will operate normally in the System Timing Slot, it will not function as a System Timing Module.



* replaced with chassis slot number

Figure 5.2—PXIe Chassis Slot Glyphs

This page intentionally blank.

6. MODULE CONFIGURATIONS

OmniBus modules are available for many different protocols, including MIL-STD-1553, ARINC 429/575, ARINC 708/453, and ARINC 717/573. Other standard and custom modules are available. This chapter lists the part numbers for PCIe/PXie cards and for MIL-STD-1553, ARINC 429/717, and ARINC 708 I/O modules and describes the features and functionality of each.

Note: OmniBus products are not user-configurable. Do not attempt to swap one module type for another one with a different part number. If a module is exchanged, it must be with an identical module. OmniBus products may be upgraded with additional channels or protocols, but this must be done at the factory.

6.1 OmniBus II PCIe Boards

The OmniBus II PCIe card can host up to two modules, one per core. OmniBus II PCIe carrier boards are listed in the table below:

Part No.	Description
211	Reserved
212	OmniBus II PCIe w/ 2 cores

Table 6.1—OmniBus II PCIe Host Part Numbering

6.2 OmniBus II PXie Boards

The OmniBus II PXie card can host up to two modules, one per core. OmniBus II PXie carrier boards are listed in the table below:

Part No.	Description
221	Reserved
222	OmniBus II PXie w/ 2 cores

Table 6.2—OmniBus II PXie/cPCIe Host Part Numbering

6.3 MIL-STD-1553 Modules

MIL-STD-1553 modules can have one or two dual-redundant databuses (channels). The part numbers for 1553 modules start with 5 (e.g., 511). The second digit identifies the level of 1553 channel 0, and the third digit identifies the level of 1553 channel 1 (a zero indicates no second bus). Table 6.3 illustrates standard single and dual channel 1553 modules.

Part No.	CH0 Level	CH1 Level
510	S	–
540	M	–
550	P	–
511	S	S
544	M	M
555	P	P

S = Single function
M = Multifunction
P = Multifunction with parametrics

Table 6.3—MIL-STD-1553 Module Part Numbering

Each MIL-STD-1553 channel is available in three levels of functionality (summarized in the table below). All levels provide at least single terminal Bus Controller, Remote Terminal, and Monitor operation and user-configurable RT response time. Advanced features include multi-terminal simulation (up to 32) with concurrent monitoring and protocol error injection (word, gap, and message errors). Level P MIL-STD-1553 modules provide variable transmit amplitude and zero crossing distortion.

Level Designator→	S	M	P
Level Number used in Part No.→	1	4	5
Number of Simultaneous Terminals	1	32	32
Configurable RT Response Time	✓	✓	✓
Monitor	✓	✓	✓
Filtering for terminal address	✓	✓	✓
Filtering for subaddress		✓	✓
Concurrent terminal monitoring		✓	✓
Protocol Error Injection		✓	✓
Variable Transmit Amplitude			✓
Zero Crossing Distortion			✓

Table 6.4—MIL-STD-1553 Level Function Definition

The MIL-STD-1553 modules also contain Avionics Discrete I/O, see Section 4.5 for more details.

6.3.1 Software-Selectable Bus Termination

Each databus on all OmniBus MIL-STD-1553 modules has a 75-ohm termination resistor that can be switched across the direct-coupled terminals under software control. When transformer coupling is used, the direct-coupled termination resistance must be off, and external couplers and terminators are required. See Appendix A for more information about bus termination and transformer versus direct coupling.

6.3.2 Configurable RT Response Time

The RT response time of MIL-STD-1553 OmniBus modules may be individually set in software for each 1553 channel. The response time is measured from the mid-bit zero crossing of the parity bit to the mid-bit zero crossing of the status word. The RT response time may be set through software using the BTI1553_RTResponseTimeSet function. The response time value is an integer that represents the response time in hundreds of nanoseconds up to 25.5 microseconds. The minimum response time is affected by the error checking process and is about 3.7 microseconds for MIL-STD-1553A and 7.7 microseconds for MIL-STD-1553B (the default protocol in BTI1553_RTConfig). Any value below the minimum yields the minimum. The default RT response time for OmniBus modules is approximately 9 microseconds (a value of 90). The exact response time depends on several factors, such as where on the bus it is measured, analog and digital delays in the on-board circuits, and uncertainty due to the 100-nanosecond sampling time.

6.3.3 Variable Transmit Amplitude

For OmniBus II MIL-STD-1553 level P channels, the amplitude of the transmitted databus signal can be varied under software control. Using BTI1553_ParamAmplitudeConfig, the OmniBus II has extended functionality from the OmniBus Family. The configuration value, configval, now allows for setting a high and a low range. The lower 12 bits of dacval represent the transmit amplitude. The OmniBus II has a resolution of 8 bits, so the least significant 4 bits are “don’t cares.” The full-scale value of FF0h is the default setting.

<i>configval</i>	
Constant	Description
PARAMCFG1553_DEFAULT	Select all default settings (bold below)
PARAMCFG1553_AMPLON	Enables parametric amplitude control
PARAMCFG1553_AMPLHI	Enables parametric amplitude control high range
PARAMCFG1553_AMPLLO	Enables parametric amplitude control low range
PARAMCFG1553_AMPLOFF	Disables parametric amplitude control

Table 6.5—MIL-STD-1553 ParamAmplitudeConfig Configval

Constant	Range*	Output Drive*
PARAMCFG1553_AMPLHI	0V-26V	.101*(dacval >> 4)
PARAMCFG1553_AMPLLO	0V-4.9V	.019*(dacval >> 4)

Table 6.6—MIL-STD-1553 ParamAmplitudeConfig Output Drive

*Since the actual amplitude and linearity depend on both the line driver and load, the user must calibrate with the conditions in use for the degree of accuracy desired. Some line drivers are not capable of putting out very low voltages; be sure to verify the output under your operating conditions.

6.3.4 Zero Crossing Distortion

On level P channels, a zero crossing of the transmitted signal can be shifted from its normal position under software control. This feature allows a specific zero crossing to be shifted up to plus or minus 250 nanoseconds, in increments of 5 ns. A zero crossing shift can be generated on the leading or mid-bit zero crossing of a specified bit position in a specified word.

6.4 ARINC 429 Modules

The preferred ARINC 429 I/O modules for OmniBus II are listed in the table below. Each channel of those marked as “selectable” in the table can be configured as either a receiver or a transmitter.

Part No.	429 Channels	Parametric Waveform	Configurable Frequency	Output State
441	8 R/T selectable		✓	
442	16 R/T selectable		✓	
454	4R/4T	✓	✓	✓
455*	4R/4T	✓	✓	✓
458	8R/8T	✓	✓	✓

R = receive and T = transmit

** Also includes ARINC 717 channels*

Table 6.7—ARINC 429 Module New Applications Part Numbering

Though the above ARINC 429 modules are preferred and recommended for new applications, Ballard does offer OmniBus II modules in the table below in configurations and functionality corresponding to those in the original OmniBus family. Pinout and functionality for these modules are described in the original OmniBus manuals.

Part No.	429 Channels	Special Functionality
421	16R/0T	–
422	12R/4T	–
423	8R/8T	–
424	4R/12T	–
425	0R/16T	–
426	8R/0T	–
427	4R/4T	–
428	0R/8T	–
434	4R/4T	Amplitude & Frequency
435*	4R/4T	Amplitude & Frequency
438	8R/8T	Amplitude & Frequency

R = receive and T = transmit

** Also includes ARINC 717 channels*

Table 6.8—ARINC 429 Module Deprecated Part Numbering

The 435 and 455 modules include ARINC 717 channels (as described in Section 6.6). Note that for 435 modules each ARINC 429 receive channel is only available when the corresponding 717 receive channel is not configured for bipolar.

ARINC 429 modules are available in many combinations of receive/transmit channels and features. All ARINC 429 receive channels feature automatic speed detection and independent label and SDI filtering. Each transmit channel automatically maintains accurate label repetition rates. To support data transfer protocols, aperiodic words may be transmitted without altering the timing of periodic words. Both receive and transmit channels may be independently set for standard low or high speed (12.5 or 100 Kbps). As shown in the table, some ARINC 429 modules provide capability to control the transmitted waveform, frequency, and output state.

6.4.1 Parametric Waveform

OmniBus II ARINC 429 modules with parametric waveform capability provide control over transmit amplitude, offset and null voltages, rise time, and fall time. The amplitude, offset and null voltages are controlled by specifying the high, null, and low voltages of the differential waveform. These parameters can be individually set in software for each channel as shown in the following table.

Parameter	Min	Max	Resolution	Comment
Waveform High*	-20 V	+20 V	10 mV	10 V (nominal) Range = 0 to 200%
Waveform Null**	-20 V	+20 V	10 mV	0 V (nominal)
Waveform Low*	-20 V	+20 V	10 mV	-10 V (nominal) Range = 0 to 200%
Rise/Fall Time (High speed)	1 μ sec	4 μ sec	100 nsec	Slew rate limit: 40V/ μ sec
Rise/Fall Time (Low speed)	1 μ sec	39 μ sec	100 nsec	Slew rate limit: 40V/ μ sec

* Differential.

** Differential and must be between the waveform high and low.

Table 6.9—ARINC 429 Parametric Waveform Characteristics

6.4.2 Configurable Frequency

Parametric ARINC 429 modules can be operated at non-standard speeds. This configurable frequency can be set in software for each transmit and receive channel. Thus, 429 channels may be used with equipment that varies from the ARINC 429 standard (such as some implementations of ARINC 575).

Use a bit rate configuration function to get a non-standard frequency. Contact Ballard Technology (see Section 1.5) for the appropriate parameters for your module part number and desired frequency.

6.4.3 Output State

ARINC 429 modules with output state functionality have the capability under software control to:

- Open either output leg of a transmit channel
- Short either output leg of a transmit channel to ground

6.5 ARINC 708 Modules

The table below lists the I/O modules available with ARINC 708 channels

Part No.	708 Channels	Parametrics
810	1R/1T	–
811	2R/2T	–
820	1R/1T	Amplitude
822	2R/2T	Amplitude

R = receive and T = transmit

Table 6.10—MIL-STD-1553 Module Part Numbering

ARINC 708 modules are available with one receiver and one transmitter or two receivers and two transmitters. ARINC 708 channels communicate on the airborne pulse Doppler weather radar display databus. Each channel can be independently switched to operate on either of two buses. All channels can be configured for variable bit length. Parametric versions of ARINC 708 modules have variable transmit amplitude.

The ARINC 708 modules also contain Avionics Discrete I/O, see Section 4.5 for more details.

6.5.1 Software-Selectable Bus Termination

Each databus on all OmniBus ARINC 708 modules has a 75-ohm termination resistor that can be switched across the direct-coupled terminals under software control. Though direct coupling is standard for ARINC 708, transformer coupling may be used. When the bus is transformer coupled, the direct-coupled termination resistance must be off, and external couplers and terminators are required. See Appendix A for more information about bus termination and transformer versus direct coupling.

6.5.2 Variable Bit Length

All OmniBus ARINC 708 modules can support messages with user-defined number of bits. Variable bit length mode is software-selected at the channel level. Special functions are provided in the BTIDriver API to read and write messages with a bit count of 1 to 1865 (116 x 16). This allows ARINC 708 channels to be used with other transfer protocols that vary from the standard (1600-bit word) display databus.

6.5.3 Variable Transmit Amplitude

For OmniBus II ARINC 708 modules with parametric capability, the amplitude of the transmitted databus signal can be varied under software control. Using `BTI708_ParamAmplitudeConfig`, the OmniBus II has extended functionality from the OmniBus Family. The configuration value, `configval`, now allows for setting a high and a low range. The lower 12 bits of `dacval` represent the transmit amplitude. The OmniBus II has a resolution of 8 bits, so the least significant 4 bits are “don’t cares.” The full-scale value of `FF0h` is the default setting.

<i>configval</i>	
Constant	Description
PARAMCFG708_DEFAULT	Select all default settings (bold below)
PARAMCFG708_AMPLON	Enables parametric amplitude control
PARAMCFG708_AMPLHI	Enables parametric amplitude control high range
PARAMCFG708_AMPLLO	Enables parametric amplitude control low range
PARAMCFG708_AMPLOFF	Disables parametric amplitude control

Table 6.11—ARINC 708 ParamAmplitudeConfig Configval

Since the actual amplitude and linearity depend on both the line driver and load, the user must calibrate with the conditions in use for the degree of accuracy desired. Some line drivers are not capable of putting out very low voltages; be sure to verify the output under your operating conditions.

6.6 ARINC 717 Modules

The preferred ARINC 717 I/O modules for OmniBus II are listed in the table below.

Part No.	717 Channels	717 Parametrics
451	4R/4T	Amplitude
455*	4R/4T	Amplitude

R = receive and T = transmit.

** Also includes ARINC 429 channels.*

Table 6.12—ARINC 717 Module New Applications Part Numbering

Both the 451 and 455 modules have four receivers and four transmitters. Module 455 also includes ARINC 429 channels (as described in Section 6.4). All OmniBus ARINC 717 channels are capable of operating at 64, 128, 256, 512, 1024, 2048, 4096, and 8192 wps and may be software-configured as biphase or bipolar. All transmit channels have variable amplitude capability.

Though the above ARINC 717 modules are preferred and recommended for new applications, Ballard does offer OmniBus II modules in the table below in configurations and functionality corresponding to those in the original OmniBus family.

Part No.	717 Channels	717 Parametrics
431	4R/4T	Amplitude
435*	4R/4T	Amplitude

R = receive and T = transmit.

** Also includes ARINC 429 channels.*

Table 6.13—ARINC 717 Module Deprecated Part Numbering

6.6.1 Variable Transmit Amplitude

To use variable transmit amplitude (available for all 717 channels), the user specifies the channel and a relative 12-bit amplitude. The full-scale value of FF0h is the default setting. The 12-bit amplitude has a resolution of 8 bits, so the least significant 4 bits are “don’t cares.” Since the actual amplitude and linearity depend on both the line driver and load, the user must calibrate with the conditions in use for the degree of accuracy desired. Some line drivers are not capable of putting out very low voltages; be sure to verify the output under your operating conditions.

This page intentionally blank.

7. CONNECTOR PINOUTS

The standard connector on OmniBus II products is a 60-pin Molex® LFH™ receptacle. Each OmniBus module (core) has an LFH connector dedicated to it. Signals on the LFH connector are either general-purpose or module-specific. General-purpose signals (including triggers, syncs, discretets, and timing) are common to most modules and protocols. The databus signals are module-specific in that their use and meaning depend on the protocol and functionality of the associated OmniBus II module. This chapter provides the information needed to connect to the individual modules through the LFH connector. Should your OmniBus II product have a different connector or a module not listed here, please consult other documentation provided with the product or call Customer Support at Ballard Technology (see Section 1.5).

Ballard offers a number of special cable assemblies to facilitate the use of their OmniBus product line.

7.1 Interface Connector

The user interface connector on OmniBus II products is a Molex 60-pin LFH receptacle (Molex PN 70928-2000). The recommended mating connector is a cable plug assembly consisting of a frame subassembly (Molex PN 70929-2000) and four terminal strips (Molex PN 51-24-2022). For more information, consult www.molex.com. Appropriate shields, strain-reliefs, and backshells are also required. The LFH is a high-density connector about the size of a 15-pin D-sub-miniature connector. For proper clearance from adjacent connectors, the overall length of each LFH connector (including any backshell molding) must not exceed 1.64 inches.

7.2 General Pinout

All OmniBus II products have the basic pin designations shown in Table 7.1. Note that wiring is done in pairs; 30 pairs total. Especially on databus signals (labeled “*BUSxx*” in Table 7.1), be sure to use twisted pairs to avoid cross talk. The suffix on the designations for databus signals in Table 7.1 represents the polarity (P for positive and N for negative). The use and meaning of databus signals depends on the protocol and functionality of the associated OmniBus II module, as indicated in Section 7.3.

Pair #	LFH Pin	Name	Pair #	LFH Pin	Name
1	2	BUS0N	16	32	BUS8N
	3	BUS0P		33	BUS8P
2	4	BUS2N	17	34	BUS10N
	5	BUS2P		35	BUS10P
3	6	BUS4N	18	36	BUS12N
	7	BUS4P		37	BUS12P
4	8	BUS6N	19	38	BUS14N
	9	BUS6P		39	BUS14P
5	10	GND	20	40	GND
	11	CDIO0		41	CDIO3
6	12	GND	21	42	GND
	13	CDIO4		43	CDIO7
7	14	CGND	22	44	CGND
	15	NC/5VA		45	NC/5VA
8	16	GND	23	46	GND
	17	IRIG		47	RSVD
9	18	GND	24	48	GND
	19	CDIO5		49	CDIO6
10	20	GND	25	50	GND
	21	CDIO1		51	CDIO2
11	22	BUS7P	26	52	BUS15P
	23	BUS7N		53	BUS15N
12	24	BUS5P	27	54	BUS13P
	25	BUS5N		55	BUS13N
13	26	BUS3P	28	56	BUS11P
	27	BUS3N		57	BUS11N
14	28	BUS1P	29	58	BUS9P
	29	BUS1N		59	BUS9N
15	30	GND	30	60	GND
	1	GND		31	GND

Table 7.1—General pin designations

7.3 Module-Specific Wiring

The meaning and use of the databus signals on the LFH connector depends on the protocol and functionality of the associated module. This section provides channel definitions and connector pinouts for the more common OmniBus II modules. Listings for the “16036 Pin” give the connector and pin number for the signal when a Ballard 16036 cable is used. See Section 7.4 for more information on cables.

7.3.1 MIL-STD-1553

The pin assignments for the MIL-STD-1553 modules are listed in Table 7.2 below. Be sure to follow the coupling and termination guidelines provided in Appendix A.

Used on Modules		Name	Description	LFH Pair #	LFH Pin	16036 Pin	LFH Name
5x0	5x1 to 5x5						
CH0	CH0	CH0AD	BUS A direct coupled (+)	14	28	P2-2	BUS1P
		CH0ADR	BUS A direct coupled (-)	14	29	P2-14	BUS1N
		CH0AX	BUS A transformer coupled (+)	1	3	P2-3	BUS0P
		CH0AXR	BUS A transformer coupled (-)	1	2	P2-15	BUS0N
		CH0BD	BUS B direct coupled (+)	2	5	P2-4	BUS2P
		CH0BDR	BUS B direct coupled (-)	2	4	P2-16	BUS2N
		CH0BX	BUS B transformer coupled (+)	13	26	P2-5	BUS3P
		CH0BXR	BUS B transformer coupled (-)	13	27	P2-17	BUS3N
n/a	CH1	CH1AD	BUS A direct coupled (+)	29	58	P3-2	BUS9P
		CH1ADR	BUS A direct coupled (-)	29	59	P3-14	BUS9N
		CH1AX	BUS A transformer coupled (+)	16	33	P3-3	BUS8P
		CH1AXR	BUS A transformer coupled (-)	16	32	P3-15	BUS8N
		CH1BD	BUS B direct coupled (+)	17	35	P3-4	BUS10P
		CH1BDR	BUS B direct coupled (-)	17	34	P3-16	BUS10N
		CH1BX	BUS B transformer coupled (+)	28	56	P3-5	BUS11P
		CH1BXR	BUS B transformer coupled (-)	28	57	P3-17	BUS11N
Avionics DIOs		ADIO0	Avionics DIO 0	3	7		BUS4P
		ADIO1	Avionics DIO 1	3	6		BUS4N
		ADIO2	Avionics DIO 2	12	24		BUS5P
		ADIO3	Avionics DIO 3	12	25		BUS5N
		ADIO4	Avionics DIO 4	4	9		BUS6P
		ADIO5	Avionics DIO 5	4	8		BUS6N
		ADIO6	Avionics DIO 6	11	22		BUS7P
		ADIO7	Avionics DIO 7	11	23		BUS7N
		ADIO8	Avionics DIO 8	18	37		BUS12P
		ADIO9	Avionics DIO 9	18	36		BUS12N
		ADIO10	Avionics DIO 10	27	54		BUS13P
		ADIO11	Avionics DIO 11	27	55		BUS13N
		ADIO12	Avionics DIO 12	19	39		BUS14P
		ADIO13	Avionics DIO 13	19	38		BUS14N
		ADIO14	Avionics DIO 14	26	52		BUS15P
	ADIO15	Avionics DIO 15	26	53		BUS15N	

Table 7.2—Pinouts for MIL-STD-1553 modules

7.3.2 ARINC 429

The pin assignments for OmniBus ARINC 429 modules are listed in Table 7.3 below.

Note: Module 455 in the table below also includes ARINC 717 channels. See 7.3.4 for pinouts of the 717 channels on this module.

442 16R/T	458 8R/8T	441 8R/T	454/455* 4R/4T	Channel	Name	Polarity	LFH Pair #	LFH Pin	16036 Pin	LFH Name	
R/T	R	R/T	R	CH0	CH0P	+	1	3	P2-3	BUS0P	
					CH0N	-	1	2	P2-15	BUS0N	
R/T	R	R/T	R	CH1	CH1P	+	14	28	P2-2	BUS1P	
					CH1N	-	14	29	P2-14	BUS1N	
R/T	R	R/T	R	CH2	CH2P	+	2	5	P2-4	BUS2P	
					CH2N	-	2	4	P2-16	BUS2N	
R/T	R	R/T	R	CH3	CH3P	+	13	26	P2-5	BUS3P	
					CH3N	-	13	27	P2-17	BUS3N	
R/T	R	R/T	n/a	CH4	CH4P	+	3	7	P2-6	BUS4P	
					CH4N	-	3	6	P2-18	BUS4N	
R/T	R	R/T		CH5	CH5P	+	12	24	P2-8	BUS5P	
					CH5N	-	12	25	P2-19	BUS5N	
R/T	R	R/T		CH6	CH6P	+	4	9	P2-9	BUS6P	
					CH6N	-	4	8	P2-20	BUS6N	
R/T	R	R/T		CH7	CH7P	+	11	22	P2-10	BUS7P	
					CH7N	-	11	23	P2-21	BUS7N	
R/T	T	n/a		T	CH8	CH8P	+	16	33	P3-3	BUS8P
						CH8N	-	16	32	P3-15	BUS8N
R/T	T			T	CH9	CH9P	+	29	58	P3-2	BUS9P
						CH9N	-	29	59	P3-14	BUS9N
R/T	T		T	CH10	CH10P	+	17	35	P3-4	BUS10P	
					CH10N	-	17	34	P3-16	BUS10N	
R/T	T		T	CH11	CH11P	+	28	56	P3-5	BUS11P	
					CH11N	-	28	57	P3-17	BUS11N	
R/T	T		n/a	CH12	CH12P	+	18	37	P3-6	BUS12P	
					CH12N	-	18	36	P3-18	BUS12N	
R/T	T			CH13	CH13P	+	27	54	P3-8	BUS13P	
					CH13N	-	27	55	P3-19	BUS13N	
R/T	T	CH14		CH14P	+	19	39	P3-9	BUS14P		
				CH14N	-	19	38	P3-20	BUS14N		
R/T	T	CH15		CH15P	+	26	52	P3-10	BUS15P		
				CH15N	-	26	53	P3-21	BUS15N		

R = receive and T = transmit

* See Table 7.5 for the ARINC 717 pinouts on PN 455

Table 7.3—Pinouts for ARINC 429 modules

7.3.3 ARINC 708

The pin assignments for ARINC 708 modules are listed in Table 7.4 below. Each channel can use either of two buses, which are shared between adjacent receive and transmit channels. Thus, receive channel 0 can listen to either bus, one of which could have the transmissions from channel 1. Be sure to follow the

coupling and termination guidelines provided in Appendix A. Direct coupling is standard for ARINC 708, but transformer coupling is possible.

Used on Modules		Description	LFH Pair #	LFH Pin	16036 Pin	LFH Name
810/820	811/822					
CH0 (R)	CH0 (R)	BUS A direct coupled (+)	14	28	P2-2	BUS1P
		BUS A direct coupled (-)	14	29	P2-14	BUS1N
		BUS A transformer coupled (+)	1	3	P2-3	BUS0P
		BUS A transformer coupled (-)	1	2	P2-15	BUS0N
		BUS B direct coupled (+)	2	5	P2-4	BUS2P
		BUS B direct coupled (-)	2	4	P2-16	BUS2N
		BUS B transformer coupled (+)	13	26	P2-5	BUS3P
		BUS B transformer coupled (-)	13	27	P2-17	BUS3N
CH1 (T)	CH1 (T)	BUS A direct coupled (+)	14	28	P2-2	BUS1P
		BUS A direct coupled (-)	14	29	P2-14	BUS1N
		BUS A transformer coupled (+)	1	3	P2-3	BUS0P
		BUS A transformer coupled (-)	1	2	P2-15	BUS0N
		BUS B direct coupled (+)	2	5	P2-4	BUS2P
		BUS B direct coupled (-)	2	4	P2-16	BUS2N
		BUS B transformer coupled (+)	13	26	P2-5	BUS3P
		BUS B transformer coupled (-)	13	27	P2-17	BUS3N
n/a	CH2 (R)	BUS A direct coupled (+)	29	58	P3-2	BUS9P
		BUS A direct coupled (-)	29	59	P3-14	BUS9N
		BUS A transformer coupled (+)	16	33	P3-3	BUS8P
		BUS A transformer coupled (-)	16	32	P3-15	BUS8N
		BUS B direct coupled (+)	17	35	P3-4	BUS10P
		BUS B direct coupled (-)	17	34	P3-16	BUS10N
		BUS B transformer coupled (+)	28	56	P3-5	BUS11P
		BUS B transformer coupled (-)	28	57	P3-17	BUS11N
	CH3 (T)	BUS A direct coupled (+)	29	58	P3-2	BUS9P
		BUS A direct coupled (-)	29	59	P3-14	BUS9N
		BUS A transformer coupled (+)	16	33	P3-3	BUS8P
		BUS A transformer coupled (-)	16	32	P3-15	BUS8N
		BUS B direct coupled (+)	17	35	P3-4	BUS10P
		BUS B direct coupled (-)	17	34	P3-16	BUS10N
		BUS B transformer coupled (+)	28	56	P3-5	BUS11P
		BUS B transformer coupled (-)	28	57	P3-17	BUS11N
Avionics DIOs	Avionics DIO 0	3	7		BUS4P	
	Avionics DIO 1	3	6		BUS4N	
	Avionics DIO 2	12	24		BUS5P	
	Avionics DIO 3	12	25		BUS5N	
	Avionics DIO 4	4	9		BUS6P	
	Avionics DIO 5	4	8		BUS6N	
	Avionics DIO 6	11	22		BUS7P	
	Avionics DIO 7	11	23		BUS7N	
	Avionics DIO 8	18	37		BUS12P	
	Avionics DIO 9	18	36		BUS12N	
	Avionics DIO 10	27	54		BUS13P	
	Avionics DIO 11	27	55		BUS13N	
	Avionics DIO 12	19	39		BUS14P	
	Avionics DIO 13	19	38		BUS14N	
	Avionics DIO 14	26	52		BUS15P	
	Avionics DIO 15	26	53		BUS15N	

Table 7.4—Pinouts for ARINC 708 modules

7.3.4 ARINC 717

The pin assignments for the ARINC 717 modules are listed in Table 7.5 below. All ARINC 717 channels can be either biphasic or bipolar.

Note: Module 455 in the table below also includes ARINC 429 channels. See Table 7.3 for pinouts of the 429 channels on this module.

451 4R/4T	455* 4R/4T	Channel	Polarity	LFH Pair #	LFH Pin	16036 Pin	LFH Name
R (biphase/ bipolar)	R (biphase/ bipolar)	CH4	+	3	7	P2-6	BUS4P
			-	3	6	P2-18	BUS4N
R (biphase/ bipolar)	R (biphase/ bipolar)	CH5	+	12	24	P2-8	BUS5P
			-	12	25	P2-19	BUS5N
R (biphase/ bipolar)	R (biphase/ bipolar)	CH6	+	4	9	P2-9	BUS6P
			-	4	8	P2-20	BUS6N
R (biphase/ bipolar)	R (biphase/ bipolar)	CH7	+	11	22	P2-10	BUS7P
			-	11	23	P2-21	BUS7N
T (biphase/ bipolar)	T (biphase/ bipolar)	CH12	+	18	37	P3-6	BUS12P
			-	18	36	P3-18	BUS12N
T (biphase/ bipolar)	T (biphase/ bipolar)	CH13	+	27	54	P3-8	BUS13P
			-	27	55	P3-19	BUS13N
T (biphase/ bipolar)	T (biphase/ bipolar)	CH14	+	19	39	P3-9	BUS14P
			-	19	38	P3-20	BUS14N
T (biphase/ bipolar)	T (biphase/ bipolar)	CH15	+	26	52	P3-10	BUS15P
			-	26	53	P3-21	BUS15N

R = receive and T = transmit

** See Table 7.3 for the ARINC 429 pinouts for PN 455*

Table 7.5—Pinouts for ARINC 717 modules

7.4 Standard Cables

Ballard sells a number of different cables that are useful for wiring to OmniBus II products. Each cable has a standard length. Non-standard lengths may be specified by adding a /xx suffix after the part number, where xx is the length in feet. For example, a 16036/10 is a ten-foot-long 16036.

7.4.1 PN 16035 Cable Assembly: LFH to LFH

This is a three-foot-long straight-through cable with 60-pin male LFH plugs on both ends. It is wired pin for pin and pair for pair as shown in Table 7.1. The 16035 is useful for connecting an OmniBus II product to a user-provided panel or other assembly.

7.4.2 PN 16036 Cable Assembly: LFH to Two 25-pin D-sub

This is a three-foot-long Y-cable that adapts a 60-pin male LFH plug (labeled P1) to two 25-pin male D-subminiature connectors (P2 and P3). Because of the size and popularity of D-sub connectors, some users may find it easier to interface to them than to the OmniBus II LFH connectors. As can be seen from Table 7.1, there is symmetry between the upper and lower halves of the LFH connector. On the 16036 cable assembly, the upper half of the LFH connector is wired to one D-sub and the lower half is wired to the other D-sub, thus giving similar signals on the corresponding pins of both D-sub. The wire pairs on the 16036 are different from those on the 16035. Wiring for the 16036 cable is shown in Table 7.6 below.

P2 Pair #	From P1 pin	To P2 pin	Name	P3 Pair #	From P1 pin	To P3 pin	Name
1	3	3	BUS0P	1	33	3	BUS8P
1	2	15	BUS0N	1	32	15	BUS8N
2	28	2	BUS1P	2	58	2	BUS9P
2	29	14	BUS1N	2	59	14	BUS9N
3	5	4	BUS2P	3	35	4	BUS10P
3	4	16	BUS2N	3	34	16	BUS10N
4	26	5	BUS3P	4	56	5	BUS11P
4	27	17	BUS3N	4	57	17	BUS11N
5	7	6	BUS4P	5	37	6	BUS12P
5	6	18	BUS4N	5	36	18	BUS12N
6	24	8	BUS5P	6	54	8	BUS13P
6	25	19	BUS5N	6	55	19	BUS13N
7	9	9	BUS6P	7	39	9	BUS14P
7	8	20	BUS6N	7	38	20	BUS14N
8	22	10	BUS7P	8	52	10	BUS15P
8	23	21	BUS7N	8	53	21	BUS15N
9	11	11	CDIO0	9	41	11	CDIO3
9	10	23	GND	9	40	23	GND
10	17	12	IRIG	10	47	12	RSVD
10	19	24	CDIO5	10	49	24	CDIO6
11	13	13	CDIO4	11	43	13	CDIO7
11	15	25	NC/5V	11	45	25	NC/5V
12	21	22	CDIO1	12	51	22	CDIO2
12	20	1	GND	12	50	1	GND
13	14	7	CGND	13	44	7	CGND

Braids connected shell to shell

Table 7.6—Wiring chart for 16036 cable assembly

7.4.3 MIL-STD-1553 Cable Assemblies

Ballard offers four standard cable assemblies for MIL-STD-1553 (see Table 7.7 below). The standard length is three feet.

Cable Assy. No.	No. of Ch.	CH0	CH1	D-Sub
16037	2	✓	✓	✓
16038	2	✓	✓	–
16039	1	✓	–	✓
16041	1	✓	–	–

Table 7.7—MIL-STD-1553 cable assembly configurations

These four cables are available for single or dual channel modules and with or without a D-sub connector. All of them provide a twinax cable from the LFH connector to a PL-75 for each of the transformer coupled MIL-STD-1553 buses. All channels are dual redundant, so there are either two or four twinax cables with PL-75s on each assembly. Twinax cables are wired as shown in Table 7.8.

Cable Name	Wire Name	From LFH Pin	To PL-75	LFH Name
CH0	CH0AX	3	Center	BUS0P
BUS A	CH0AXR	2	Outer	BUS0N
CH0	CH0BX	26	Center	BUS3P
BUS B	CH0BXR	27	Outer	BUS3N
CH1	CH1AX	33	Center	BUS8P
BUS A	CH1AXR	32	Outer	BUS8N
CH1	CH1BX	56	Center	BUS11P
BUS B	CH1BXR	57	Outer	BUS11N

Braids connected between the LFH shell and the PL-75 shell

Table 7.8—Twinax wiring on MIL-STD-1553 cable assemblies

The 25-pin female D-subminiature connector provides IRIG and discrete signals, as shown in Table 7.9. Consequently, the recommended cable assemblies are 16037 for dual-channel and 16039 for single-channel MIL-STD-1553 OmniBus modules.

Pair	Name	From LFH Pin	To DB25S Pin
1	CDIO0	11	1
1	GND	10	14
2	CDIO1	21	2
2	GND	20	15
3	CDIO2	51	3
3	GND	50	16
4	CDIO3	41	4
4	GND	40	17
5	CDIO7	43	6
5	GND	42	19
6	CDIO4	13	7
6	GND	12	20
7	CDIO5	19	8
7	GND	18	21
8	CDIO6	49	9
8	GND	48	22
9	IRIG	17	10
9	GND	16	23
10	RSVD	47	11
10	GND	46	24
11	NC/5V	45	12
11	CGND	44	13

Braids connected shell to shell

Table 7.9—D-sub connector pinout for cable assemblies 16037 and 16039

APPENDIX A COUPLING AND TERMINATION

Coupling and termination only apply to OmniBus modules for MIL-STD-1553 and ARINC 708. Electrically, these databuses have similar characteristics. Except where a protocol is specified, the following discussion applies to both.

A.1 Bus Termination

The main databus consists of a pair of twisted, shielded wires with a characteristic impedance in the range of 70 to 85 ohms. The databus must be terminated at both ends with a resistor to provide proper loading and to minimize signal reflection and degradation on the bus. The resistor value should be close to the characteristic impedance of the databus. The resulting total load on the databus is the two terminating resistors in parallel (about 39 ohms). Even with a very short databus, the load from the terminating resistors is still required. Notice how the resistors terminate the databuses in Figure A.1 and Figure A.2. Note that some Ballard products have on-board termination resistors that can be switched in manually or under software control.

Note: The most common problem in a new system is an improperly terminated databus.

A.2 Transformer versus Direct Coupling

MIL-STD-1553 can be either direct or transformer coupled. Most military MIL-STD-1553 systems are transformer coupled. ARINC 708, however, is normally direct coupled.

Both protocols have a transformer as part of the terminal's interface, but MIL-STD-1553 transformer coupling has an additional external transformer coupler that isolates the stub from the main databus and reduces signal reflections. The signal level on the main bus is the same for both direct and transformer coupling. Though it is rarely done, systems can mix the use of direct and transformer coupling.

A terminal must be properly configured for either direct or transformer coupling. There is a difference between the terminal's internal interface circuit for direct and transformer coupling:

- The transformer-coupled terminal has a lower turns ratio and no isolation resistors, but this is made up for in the external coupler, which has a step-up transformer and isolation resistors (see Figure A.1).
- The direct-coupled terminal has a higher turns ratio and has isolation resistors that are connected directly to the main databus. Direct coupled stubs should be kept as short as possible (see Figure A.2).

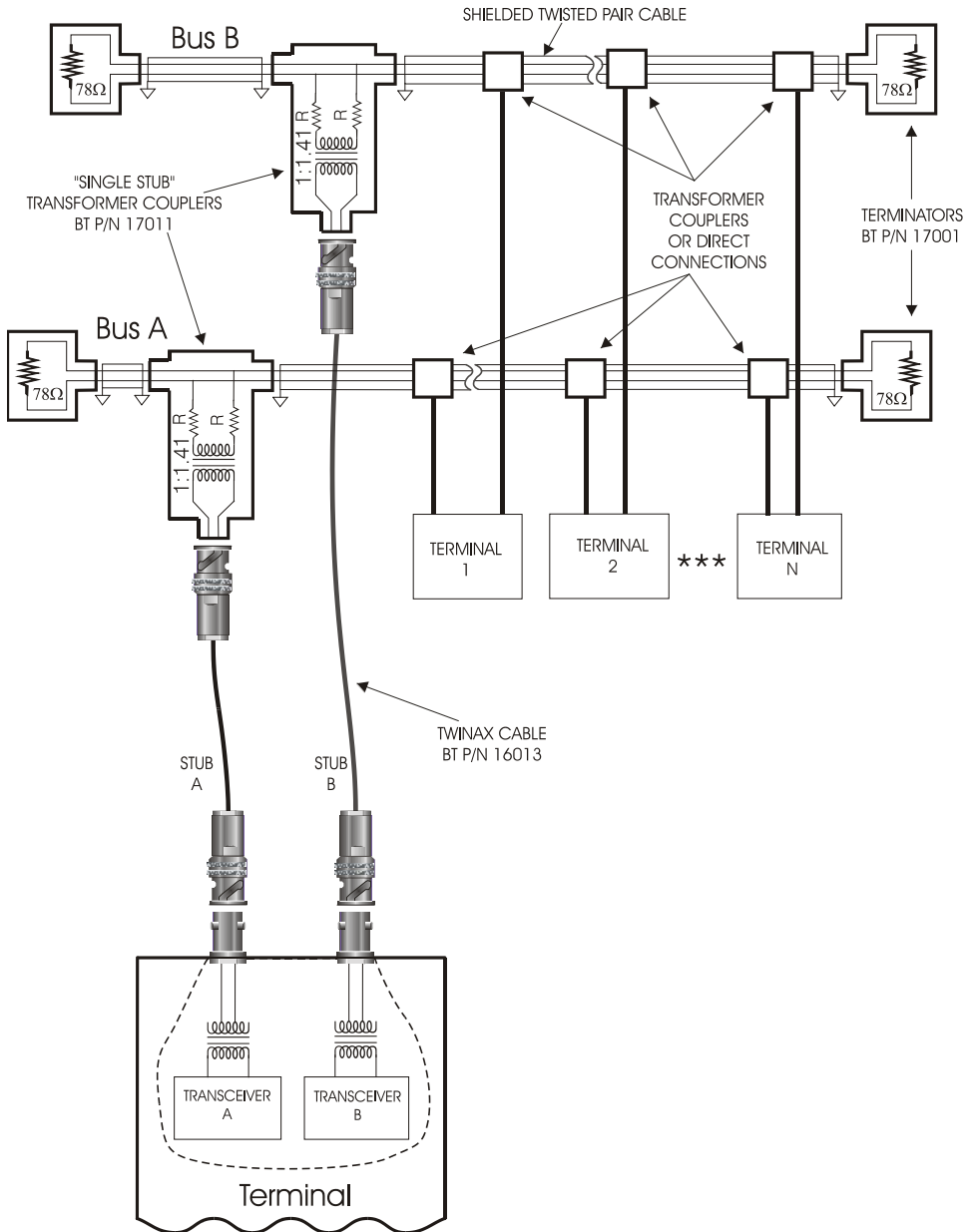


Figure A.1—Transformer coupling to a dual-redundant databus

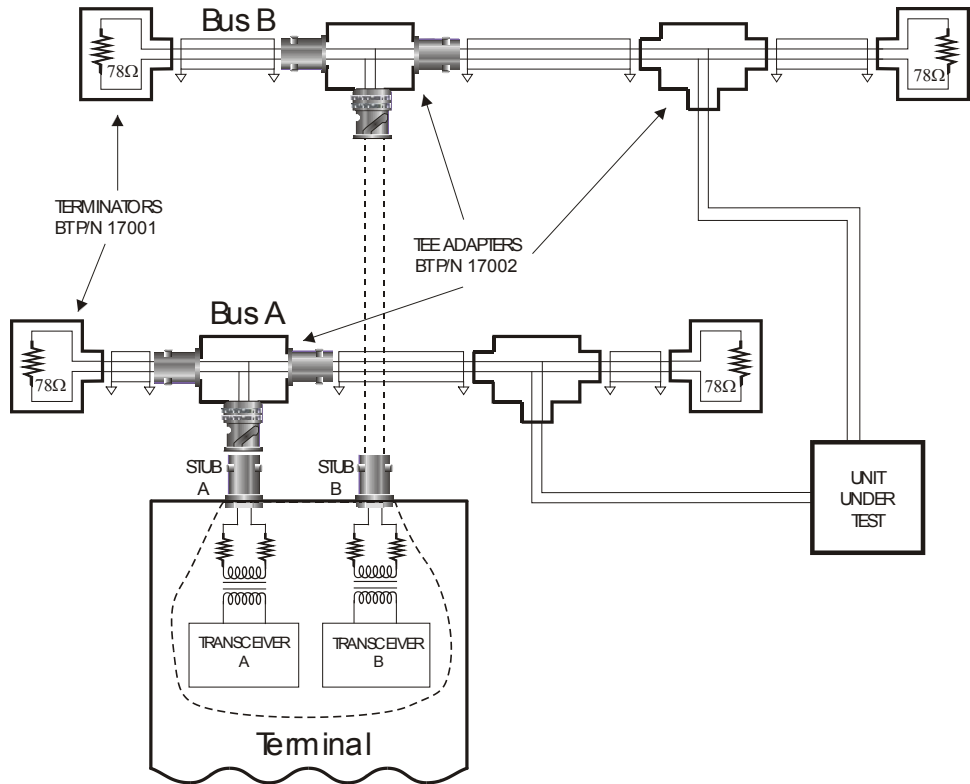


Figure A.2—Direct connection to a dual-redundant databus

This page intentionally blank.

APPENDIX B REVISION HISTORY

The following revisions have been made to this manual:

Rev. A *Date: March 27, 2014*

Initial release of this manual.

Rev. B0 *Date: February 6, 2015*

Added updates for OmniBus II PXIe

This page intentionally blank.