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.

APEX WAVES

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

1-800-915-6216

www.apexwaves.com

sales@apexwaves.com

Request a Quote CLICK HERE *LV-222-511-000*

OMNIBUS II NI PXIe
USER MANUAL

Interface Card
To Avionics Databuses

February 1, 2018
Rev. C.1

Astronics Ballard Technology
www.astronics.com

COPYRIGHT NOTICE

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 user 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, BTIDriver and CoPilot are registered trademarks of Ballard Technology, Inc. All other product names or trademarks are property of their respective owners.

PRODUCT CERTIFICATION

RESPONSIBLE PARTY

Astronics Ballard Technology, Inc.

11400 Airport Road, Suite 201

Everett, WA 98204, USA

Web: www.astronics.com

PRODUCTS

OmniBus II PXIe versions LV-222-xxx-xxx (NI P/N 784xxx-xx) (where x is any numeric character or blank).

FCC

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

1) This device may not cause harmful interference, and
2) This device must accept any interference received, including interference that may cause undesired operation.

CE

This device complies with the European Union Directives and standards for electromagnetic compatibility (EMC) and product safety.

RoHS

This device complies with the European Union Restriction of Hazardous Substances directive.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This manual is the user guide for PXI™ Express (PXIe) models of Astronics Ballard Technology®'s (Ballard) OmniBus® II family of products. Throughout this manual any 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 PXIe card, discusses special features, describes the installation process, and references programming alternatives.

1.1 OmniBus II Overview

OmniBus II (OB2) is a family of products that enable computer systems to communicate with avionics databuses for the purpose of testing, simulation, and/or operation. Each OB2 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, and Discrete I/O. Other protocols (such as ARINC 575, ARINC 573, ARINC 453, etc.) are also supported. Upon request, custom protocols can be implemented.

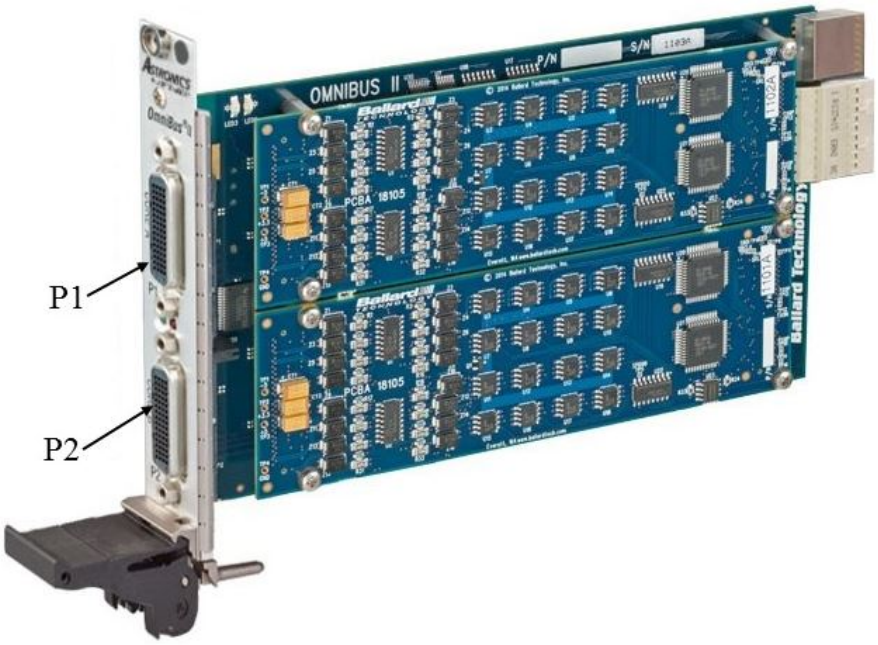


Figure 1.1—OmniBus II PXIe Card

The high-density modular design of the OB2 family provides flexibility that enables the user to select from many protocol, platform, and channel count combinations. Each OB2 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 utilized 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. Though there are similarities, the components of the two generations are not interchangeable. Figure 1.2 illustrates the modular architecture of the OB2 PXIe board.

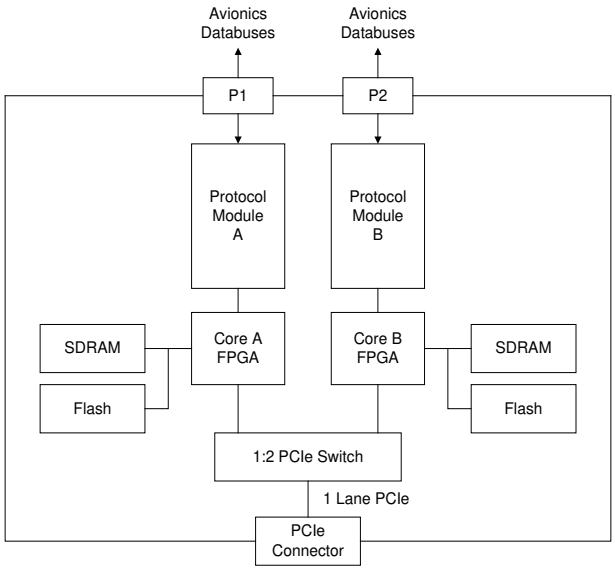


Figure 1.2—The Two-Core Architecture of OmniBus II PXIe Card

The OB2 PXIe product has been certified by National Instruments as "Compatible with LabVIEW." Included with all OB2 models prefixed with "LV-" is the LabVIEW Avionics Instrument Driver - the best way to operate the OB2 PXIe product with LabVIEW Software.

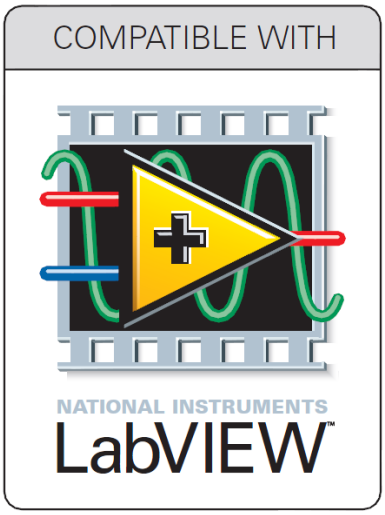


Figure 1.4—Compatible with LabVIEW Logo

Another way to operate OB2 products is with CoPilot®, Ballard'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 II family includes products with many different host platform, protocol, and channel count combinations. A given OB2 part number is produced in the factory by mounting protocol-specific modules on the required host platform and loading module specific firmware on the FPGAs.

Note: The OB2 is not user-configurable. You cannot swap one module type for another one with a different part number. You can exchange a module only with an identical module. You can upgrade the OB2 with additional channels or protocols, but you must do this at the factory.

The assembly part number characterizes the configuration of the OB2. Groups of characters separated by hyphens designate the assembly part number. The first group of characters in the assembly part number is the part number of the main board (e.g. 222 for a two-module PXIe 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. You can find a more detailed description of the individual part numbers in Chapter 6

We have printed the complete assembly part number on the main OB2 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 OB2 product may be determined by running the test program described in Section 4

For future reference, we encourage you to record the assembly part number and serial number of your OB2 product in the space provided below:

Assembly PN: LV-_____ - _____ - _____

(Board PN) (Core A PN) (Core B PN)

OB2 modules are available for many different protocols, including MIL-STD-1553 and ARINC 429/575.

1.3 Avionics Databus Protocols

Avionics databuses interconnect various electronic equipment (navigation, controls, displays, sensors, etc.) on an aircraft. There are a number of military and commercial avionics databus standards. The OB2 family of 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.
- **Discrete** I/Os provide interface capabilities for standard avionics discretes and provide general-purpose digital I/Os for sensing, controlling, and testing a variety of DC circuits

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 OB2 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 OB2 has example programs, drivers, and driver installation instructions for various operating systems (OS), and other information, files, and resources.

1.5 Support and Service

National Instruments provides support for these products. They have the most familiarity and experience with the LabVIEW platform and will be able to support LabVIEW applications most effectively.

To get support, visit:

support@ni.com

Phone: +1.866.275.6964

www.ni.com/contact-us

Please browse through the locations to find the support office closest to your location, and follow the link to open a support ticket.

1.6 Updates

Software updates for the LabVIEW instrument driver are available on the National Instruments website. Visit the Avionics Interface landing page at http://sine.ni.com/nips/cds/view/p/lang/en/nid/207798 and browse to your product to find updated software as well as the latest documentation.

1.7 Mean Time Between Failure

Product	PN Analyzed	Published MTBFs
PXIe	222-442-442	PXIe (all configurations): **2.8+ million hours** (Calculated, Telcordia SR-332 Issue 3, Ground Benign, 25°C, 60% CL)

This page intentionally blank.

2. INSTALLATION

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

- Shut down the system.
- On the card, select the required clock by sliding the onboard 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:

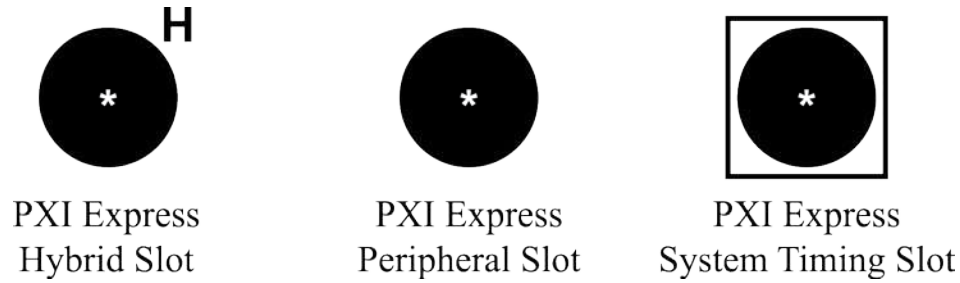


Figure 2.1—PXIe Compatible Chassis Glyphs

- While simultaneously pressing the bottom of the handle against thehorizontal 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, check if the clock select or chassis slot type is incorrectly configured. For more information, refer to Chapter 5.

2.2 Step 2: Install the Driver Software

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

The driver is available for download on the product page at NI.com. The 32-bit installation package only supports 32-bit Windows and 32-bit LabVIEW installations. The 64-bit installation package only supports LabVIEW installations on 64-bit Windows. Both packages include LabVIEW RT support.

To install the driver software:

- Download the appropriate installation package for your system.
- Select RUN or double click the installer after the download is complete.
- The Installer will install the hardware driver, LabVIEW Instrument Driver files, Examples, the Graphical Configuration Editor (BTIXMLEdit) and Help files.
- If LabVIEW is installed, the Instrument Driver VI files will be compiled for the latest supported version of LabVIEW automatically.

If you encounter problems or have installation questions, please reboot the controller prior to contacting Support (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 PXIe card.

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

In Windows, you can use the **BTITST32.EXE/ BTITST64.EXE** test program to assign and manage card numbers and to test the 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 is installed automatically with the Avionics Instrument Driver library. It can be found at the start menu: **Astronics Ballard Technology > LabVIEW Instrument Driver > BTITST32**

Note: You can use the Windows test program at any time to determine or reassign the card number.

If you need further assistance, contact Customer Support (see Section 1.5).

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

Connect the databuses to the PXIe card according to the pin assignment tables in Chapter 7. Follow the coupling and termination guidelines discussed in Appendix A: Coupling and Termination.

Connection of the ground pin(s) to the end system(s) is necessary for proper operation of ARINC 429 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 II products and to manipulate data. Using the Avionics Instrument Driver or Ballard's **BTIDriver** API library, it is easy to operate the PXIe card and to utilize its powerful interface.

3.1 CoPilot

A PC with CoPilot and Ballard's OB2 PXIe card makes a powerful, low-cost databus analyzer/simulator. CoPilot interfaces directly with OB2 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, you can automatically detect bus messages, post them in the hardware tree, and associate them with the appropriate attributes from the database of equipment, message, and engineering unit specifications.

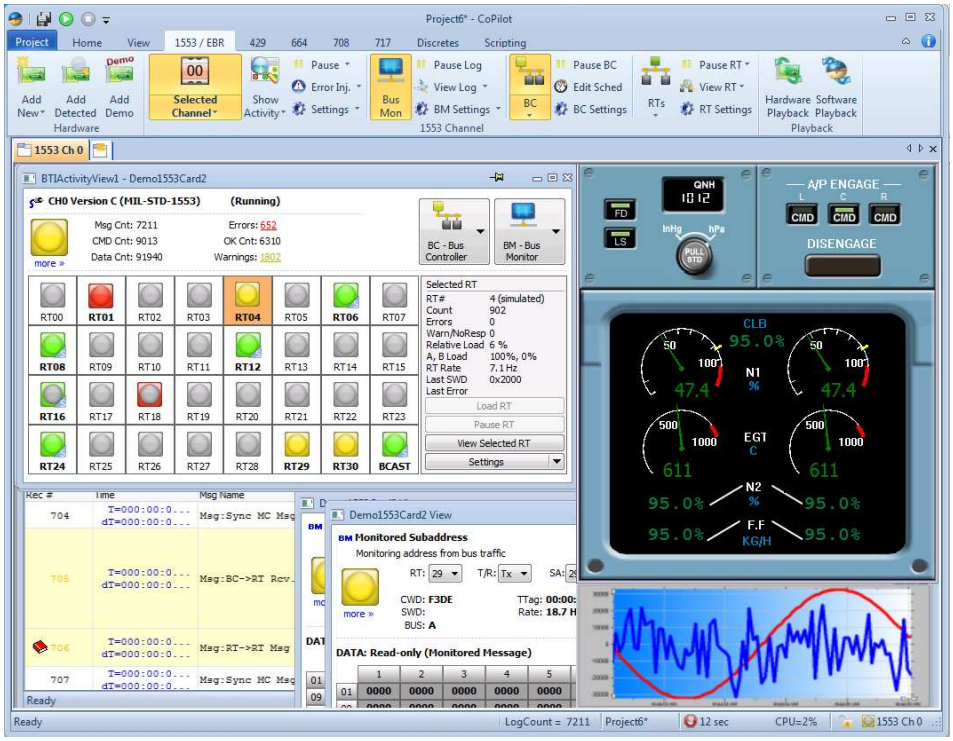


Figure 3.1—Sample CoPilot Screen

CoPilot users can quickly configure, run, and display the activity of multiple databuses in a unified view. You can observe data and change it in engineering units while the bus is running. The Strip View graphically illustrates the history of the selected data values. You can also enter data and view it as virtual instruments (knobs, dials, gauges, etc.). The user creates this data or the user automatically

generates the data 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 OB2 products. You can purchase CoPilot separately or with an OB2 product.

For more information or a free evaluation copy, contact Customer Support (see section 1.5).

3.2 User-Developed Software

All software applications use the Avionics Instrument Driver. A program can operate the OB2 with only a few VI calls 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 VI calls, the comprehensive Avionics Instrument Driver library includes a broad range of functions for specialized needs.

Sample and utility programs are included with the Avionics Instrument Driver on the software distribution package. You can find detailed information about each Avionics Instrument Driver VI and the instructions on programming for the OB2 in context-sensitive VI help for each protocol as well as well-documented and extensive example VIs. Contact Customer Support (see Section 1.5) for additional information.

4. OMNIBUS II FEATURES

This section describes special capabilities and interface signals available on many OB2 products. Some of these features (such as Inter-Range Instrumentation Group (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 for assistance (see Section 1.5).

4.1 PCI Express Bus

OmniBus II PXIe cards interface through a PCIe interface with the following features:

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

4.1.1 PXIe Form Factor

The OmniBus II PXIe card conforms to the PXIe 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 OB2 family includes onboard 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. You can find a summary of each of the tests in the following sections. For more information on the API interaction, please refer to the example programs installed with the Instrument Driver 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 onboard RAM; this verifies memory operation prior to use. In the event of a PBIT error, opening the card (`BTICard_CardOpen`) will fail.

4.2.2 Initiated Built-in Test (IBIT)

The user-developed software may initiate a comprehensive hardware test. 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). You can access the status of these tests through the API via polling or interrupts.

The system monitor polls temperature data from sensors located on the card. You can read present values, as well as historic minimum and maximum values, from the card. The API allows for setting of user temperature limits and enabling notification of exceeded limits. The OB2 will halt card operation, to protect the system, if the temperature-sensor measurements exceed built-in system limits.

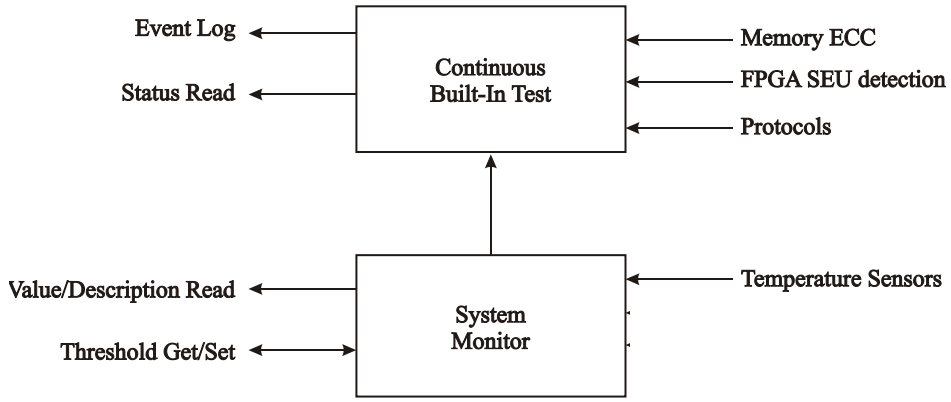


Figure 4.1—Built-in Test and System Monitor Architecture

4.3 System Time Synchronization

Systems with high reliability and fidelity requirements need system-level time synchronization to reduce the effort involved to compare event timing across interconnected devices. The OB2 series family introduces the Timing Synchronization Manager (TSM) to provide the ability to synchronize time between an external reference (IRIG PCM, IRIG AM, PPS) and the protocol cores on the OB2. Additionally, using one of the above references or a 10 MHz input, the TSM can adjust the frequency of the local clock oscillator up to 425 ppm to reduce clock drift between the OB2 and the external reference. Use of the drift control functionality can help increase synchronization between the OB2 core timer and the external reference during the gap between timing samples.

The OB2 can also be the source of the system synchronization references (IRIG PCM, IRIG AM, PPS, 10 MHz). In this case, you can set the core timer and tune the protocol clock frequency using the **BTIDriver** API functions. In some cases, the TSM can synchronize to an input signal and output a different reference signal. Table 4.1 lists the valid input and output combinations. IRIG output and (Pulse-Per-Second) PPS output are not available simultaneously.

Drift Input	Time Input	10 MHz Out	IRIG Out	PPS Out
None	None	Valid	Valid	Valid
None	IRIG	Valid	-	Valid
None	PPS	Valid	-	-
10 MHz	None	-	Valid	Valid
10 MHz	IRIG	-	-	Valid
10 MHz	PPS	-	-	-
IRIG	IRIG	Valid	-	Valid
PPS	None	Valid	Valid	Valid
PPS	IRIG	Valid	-	Valid
PPS	PPS	Valid	-	-

Table 4.1—I/O Combinations for System Synchronization

The TSM logic is activated only in Core A of an OB2 family device and all TSM configuration API calls must be directed to Core A. `BTICARD_TSM<*>` functions are used to configure the TSM, but `BTICard_IRIG<*>` functions are supported for backwards capability. However, `BTICard_IRIG<*>` functions do not allow drift control or IRIG/PPS pin selection. `BTICard IRIGConfig` uses the pins as follows:

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

Table 4.2— Pinout Supported by BTICard_IRIG

4.3.1 Core Timer

The OB2 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 ns, but a precision of 10 ns. You cannot change the resolution from one ns, however,

the current resolution of any **BTIDriver** device (including the OB2) can be read with `BTICard_TimerResolution`.

Each of the cores of the OB2 maintains lockstep time through a broadcast system time message and shared oscillator. The default timer rollover is one year, but is user adjustable to account for leap years.

`BTICard_CardReset` does not reset the core timer. A user must call `BTICard_TSMReset` to reset the TSM configuration and timer. This prevents a call to `BTICard_CardReset` on one core from interfering with the time-stamps used by a different core on the same device.

4.3.2 IRIG

An IRIG time signal contains a human-readable binary coded decimal (BCD) time-value in days, hours, minutes, seconds, etc. Many devices are synchronized using IRIG time signals. Thus, you can correlate timing data from all compatible devices easily.

You can configure the IRIG circuit as either a master or a slave. The device can drive the bidirectional IRIG timer pins only when you configure the IRIG timer as a master. When you configure the IRIG timer as a slave, it will expect the IRIG signal to come from an external device.

Using IRIG drift control in slave mode allows the OB2 to adjust the frequency of the OB2 protocol clock oscillator to match the frequency of the IRIG signal, enabling better system synchronization.

The OB2 family IRIG circuit can operate as a PCM master, PCM slave, AM master, and AM slave.

There are a number of formats for IRIG timing. The OB2 family supports the IRIG formats indicated in Table 4.3. The characteristics of the external electrical interface to the IRIG pins are as shown in Table 4.4 and Table 4.5.

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/0.1 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 4.3—Input and Output IRIG Formats

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

Table 4.4—Electrical Characteristics of the AM IRIG Signals

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

Table 4.5—Electrical Characteristics of the PCM IRIG Signals

The following Table 4.6 lists the protocols supported on each pin. Users can select between the two pins on Core A for PPS and PCM IRIG signals.

Note: The timing pins on the OB2 family are distinct and not internally connected as they were in the OmniBus family.

Timing Protocol	Core/Pin	IRIG TSM Config Constant
PPS/PCM IRIG	Core A Pin 17	`TSMCFG_PWMIRIG0`
PPS/PCM IRIG	Core A Pin 47	`TSMCFG_PWMIRIG1`
AM IRIG	Core B Pin 17	`TSMCFG_AMIRIG`

Table 4.6—IRIG Pinout

4.3.3 PPS

The Pulse-Per-Second (PPS) signal is a time-reference that indicates the start of a new second. You can configure the OB2 PPS circuit as either a master or a slave. The bidirectional buffer drives the PPS only when you configure the PPS circuit as a master. When you configure the PPS circuit as a slave, it will expect the PPS to come from an external device.

In master mode, the PPS signal is asserted high for 100 ms at the beginning of the second. In slave mode, at the rising edge of the input PPS signal, the internal core timer is rounded up or down to the nearest second. We recommend setting system time using the **BTIDriver** API when using PPS in slave mode to keep time precisely and accurately.

Using PPS drift control in slave mode allows the OB2 to adjust the frequency of the OB2 protocol clock oscillator to match the frequency of the PPS signal, enabling better system synchronization.

The OB2 supports PPS in slave and master modes simultaneously.

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

Table 4.7—Electrical Characteristics of the PPS Signal

The following Table 4.8 lists the protocols supported on each pin. Users can select between the two pins on Core A for PPS and PCM IRIG signals.

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 TSM Config Constant
PPS/PCM IRIG	Core A Pin 17	TSMCFG_PPS0
PPS/PCM IRIG	Core A Pin 47	TSMCFG_PPS1

Table 4.8—PPS Pinout

4.3.4 10MHz

A 50% duty cycle, 10 MHz sine wave is a method of system synchronization, usually used to regulate a frequency generator or oscillator. The OB2 can use the 10 MHz signal in master mode (output) or slave mode (input). When using the 10 MHz in slave mode, the OB2 will continuously monitor the phase of the 10 MHz input relative the local clock oscillator and adjust the frequency of the local clock to follow the input 10 MHz signal.

10 MHz Input Characteristics	
Min input impedance (at 1 kHz)	10 kΩ
Max input amplitude (V_{pk-pk})	8 V
10 MHz Output Characteristics	
Output amplitude (V_{pk-pk})	1.4 V
Max output resistive load	45 Ω

Table 4.9—10 MHz Input/Output Characteristics

The following Table 4.10 lists the pin supporting the 10 MHz protocol.

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	TSM Config Constant
10MHz	Core B Pin 47	TSMCFG_10MHZ

Table 4.10—10 MHz Pinout

4.4 Core Discretes

OB2 products have both input and output discrete capability. OB2 has eight bidirectional TTL level discretes per core that can be used as either inputs or outputs. Each discrete output line has a 5-V TTL driver that can source or sink up to 8 mA and has an individual tristate control; the discrete input receiver is a 5 V

tolerant device with high input impedance (10-µA leakage current). When used as an output, you can verify the status of a core-discrete output 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 discretes. The parameter *dionum* in the API functions (`BTICard_ExtDIORd`, `BTICard_ExtDIOWr`, `BTICard_ExtDIOEnWr`, etc.) specifies which discrete to read or write. Table 4.11 shows the mapping between the external hardware pin and *dionum.*

A specific sync or trigger can use one or more of the designated core discretes. 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 discretes and their use as syncs and triggers.

Table 4.11 below shows the correlation between *dionum*, the output pin, and its hardware reference designator. The last column shows which of these discretes may be used as trigger inputs or sync outputs in the **BTIDriver** API functions. The names for core discretes are prefixed by `CDIO` (e.g., `CDIO2`).

Pin Name	**LFH Pin**	**API dionum**	**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 4.11—Hardware Versus Software Designation of Core Discretes

The 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 1553 Avionics Discretes

OB2 modules that have MIL-STD-1553 channels (see Sections 6.2) also have 16 avionics shunt discretes (hereafter referred to as 1553 discretes). While these discrete I/O are simpler than those on the discrete module, they can still signal and detect events, determine status, and drive loads. The 1553 discretes may also be used as general purpose digital I/O and have a wide range of avionics and general-purpose applications. You can configure each 1553 discrete I/O pin as both a shunt input and a shunt output. When used as an output, you can verify the status of a pin 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 OB2 1553 discrete shunt input circuit, illustrated in Figure 4.2, has a 9-kΩ pull-up resistor to a 5-V 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-V source, through the forward-biased diode and the 9-kΩ resistor. A series resistor limits current as a voltage is generated across the load that is compared to a reference voltage produced by an identical configuration. This results in a 3.25-V switching voltage.

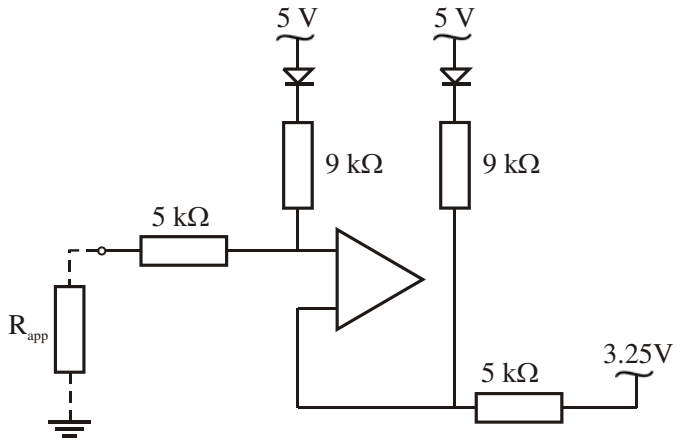


Figure 4.2—OmniBus II Discrete Shunt Input Circuit

4.5.2 Shunt Input Considerations

Limits: The OB2 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 OB2 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 (Zext). A diode protection circuit permits safe switching of inductive loads. Overload detection limits the sink current and shuts the device down in an overtemperature condition. Each output circuit is wired in parallel with an input circuit (not shown) providing self-monitor capabilities.

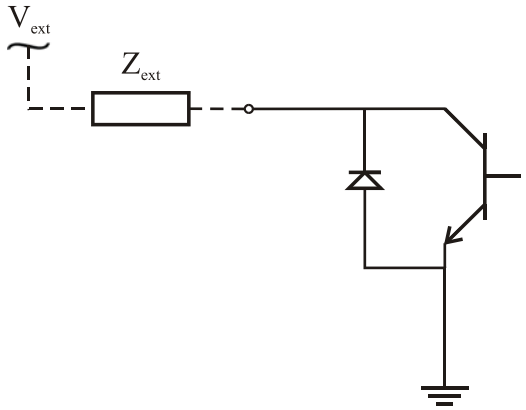


Figure 4.3—OmniBus II Discrete Shunt Output Circuit

4.5.4 Shunt Output Considerations

Limits: The OB2 1553 discrete outputs are open-ground switches capable of sinking up to 200 mA. The discrete outputs can withstand up to 35 VDC and are capable of interfacing with industry standard avionics discrete signals.

Self-Monitor: You can monitor the OB2 1553 discrete shunt output circuits by their corresponding discrete input circuits. Writing to a discrete can drive an enabled output, and reading from that discrete will report the current state of the input. Once the output is driven, there is a finite time-period before you can detect the change of state on the corresponding input. This delay (approximately 30 µs) is due to the latency of the host system and the analog delay of the input and output circuitry.

Over-Load/Fault Reporting: The OB2 1553 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. You can accomplish this 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 condition is cleared and the user drives the output again. Thus, 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 1553 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 OB2's 1553 discretes. The parameter *dionum* in the API functions (`BTICard_ExtDIORd` and `BTICard_ExtDIOWr`) specifies which discrete to read or write.

Table 4.12 below shows the correlation between *dionum* and its hardware reference designator (i.e., `ADIOn`).

Name	Description	LFH Pin#	LFH Pin Name	API dionum
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 4.12—1553 Discrete I/O Designations

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 OB2 PXIe protocol circuitry can use either the onboard 100 MHz oscillator or the PXIe connector's `PXIe_CLK100` input. Selecting the onboard 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.

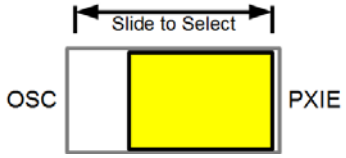


Figure 5.1—PXIe Onboard Clock Switch

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 OB2 PXIe card. All signals are asynchronous to either `CLK10` or `CLK100`. You can access PXIe Trigger signals through the `BTICard_ExtDIORd`, `BTICard_ExtDIOWr`, and `BTICard_ExtDIOEnWr` functions by passing in the associated *dionum* as per Table 5.1. You can also monitor these signals using `BTICard_ExtDIOMonConfig` with *banknum* and *rise_edge/fall_edge* bits as 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 bidirectional (I/O) signals default to tristate mode at power on. They can be tristated, driven high, or driven low with `BTICard_ExtDIOEnWr`. To drive the signal, set *dioen* to **TRUE**. To tristate the signal, set *dioen* to **FALSE**. Note that *dioval* is ignored when tristating with this function.

Only one core can drive a particular bidirectional 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]`**,** following which 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 bidirectional 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 10 MHz 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 disable it by calling `BTICard_ExtDIOEnWr` on *dionum* 44 with *dioen* set to **FALSE**. Note that this signal cannot be tristated.

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. In the case of syncs, the protocol logic can drive the sync out to both `PXI_TRIG[2]` and `PXI_STAR` at the same time.

The OB2 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`.

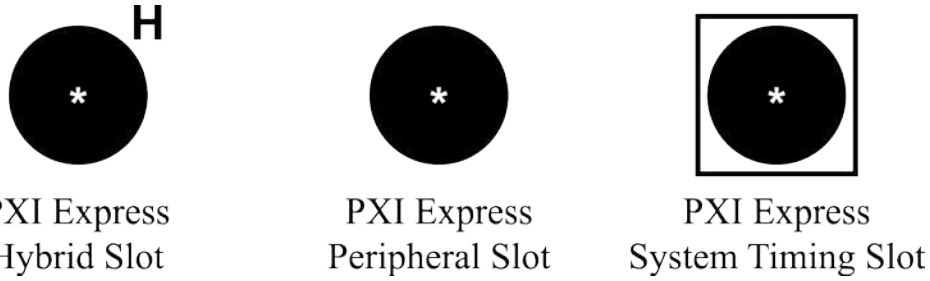
infotype	infoptr
PXITYPE_GEOADDR	Geographical Address
PXITYPE_CLKSEL	PXIe Clock Selection (Refer to Section 5.1) 0 = PXIe_CLK100 1 = Onboard 100 MHz Oscillator
PXITYPE_TRIGVERS	Version of the BTI PXIe trigger engine
PXITYPE_OUTEN	Bitmask of output enables for dionums 33 to 48 (Refer to Section 5.3 for more information)

Table 5.5—PXIe Status Parameters

5.5 Chassis Slot Glyph

You can install the PXIe card 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 Figure 5.2.

Note: 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 II modules are available for many different protocols, including MIL-STD-1553 with Avionics Discrete I/O, and ARINC 429/575. Both module types include core discretes. Other standard and custom modules are available. This chapter lists the part numbers for PXIe cards and for MIL-STD-1553 and ARINC 429 modules and describes the features and functionality of each.

Note: OB2 products are not user-configurable. You cannot swap one module type for another one with a different part number. You can exchange a module only with an identical module. You can upgrade the OB2 products with additional channels or protocols, but you must do this at the factory.

6.1 OmniBus II PXIe Boards

The OB2 PXIe card can host up to two modules, one per core. Table 6.1 below lists OB2 PXIe carrier boards:

Part No.	Description
LV-222	OmniBus II PXIe w/ 2 cores

Table 6.1—OmniBus II PXIe Host Part Numbering

6.2 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 6.2 illustrates standard single and dual channel 1553 modules.

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

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

Table 6.2—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 (RT) and Bus 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	P
Level Number used in Part No.→	**1**	**5**
Number of Simultaneous Terminals	32	32
Concurrent BC and RT Simulation		✓
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 6.3—MIL-STD-1553 Level Function Definition

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

6.2.1 Software-Selectable Bus Termination

When direct-coupled, each databus on the OB2 MIL-STD-1553 module has a 75-Ω termination resistor. This resistor can be switched across the terminals under software control. When the bus is transformer coupled, the direct-coupled termination resistance must be off, and external couplers and terminators are required. See Appendix A: Coupling and Termination for more information about bus termination and transformer versus direct coupling.

6.2.2 Configurable RT Response Time

The RT response time of MIL-STD-1553 OB2 modules may be individually set in software for each 1553 channel. You can measure the response from the mid-bit zero-crossing of the parity bit to the mid-bit zero-crossing of the status word sync. 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 ns up to 819 µs. The error checking process affects the minimum response time and is about 3.7 µs for MIL-STD-1553A and 7.7 µs for MIL-STD-1553B (the default protocol in `BTI1553_RTConfig`). Any value below the minimum yields the minimum. The default RT response time for OB2 modules is approximately 9 µs (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 onboard circuits, and uncertainty due to the 100-ns sampling time.

6.2.3 Variable Transmit Amplitude

For OB2 MIL-STD-1553 level P channels, the amplitude of the transmitted databus signal can be varied under software control. Using `BTI1553_Param-AmplitudeConfig`, the OB2 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 OB2 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.

configval	
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 6.4—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 6.5—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.*

6.2.4 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 +/- 250 ns, 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.

Note: If a zero crossing is delayed, such that the subsequent zero crossing occurs less than 400 ns later, the timing of the subsequent zero crossing may also be distorted.

6.3 ARINC 429 Modules

The Table 6.6 below, lists the preferred ARINC 429 I/O modules for OB2. Each channel of those marked as R/T 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		✓	
442	16 R/T		✓	
458	8R/8T	✓	✓	✓

R = receive and T = transmit

Table 6.6—ARINC 429 Module New Applications Part Numbering

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, the ARINC 429 module may transmit aperiodic words without altering the timing of periodic words. Both receive and transmit channels may be independently set for low or high speed (12.5 or 100 Kbps) and support varying the bit rate for non-standard applications. As shown in the Table 6.7, some ARINC 429 modules provide capability to control the transmitted waveform, frequency, and output state.

Note: When an advanced 429 transmitter is attached to a standard 429 receiver, the standard 429 receivers will clamp the bus voltage below the 200% voltage range that the advanced 429 transmitters can output. This means you can't connect an advanced 429 output to a standard 429 input and get the full amplitude of the advanced 429 module.

6.3.1 Parametric Waveform

OmniBus II ARINC 429 modules with parametric waveform capability provide control over transmit amplitude, offset and null voltages, common-mode bias, 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 6.7.

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%
Common-mode Bias+	-10 V	+10 V	5 mV	0 V (nominal)
Rise/Fall Time (High speed)	1 µs	4 µs	100 ns	Slew rate limit: 40 V/µs
Rise/Fall Time (Low speed)	1 µs	39 µs	100 ns	Slew rate limit: 40 V/µs

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

+ The common-mode bias is available on revision D or higher parametric 429 modules. The common-mode bias voltage cannot be set in a way that causes either side of signal to be outside the -10 V to +10 V range.

Table 6.7—ARINC 429 Parametric Waveform Characteristics

6.3.2 Configurable Frequency

You can operate parametric ARINC 429 modules 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 Customer Support (see Section 1.5) for the appropriate parameters for your module part number and desired frequency.

6.3.3 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

6.4 Part Number Cross Reference

The OmniBus II cards use front panel LFH connectors for databus, discretes and IRIG timing signals. The following break out cables are useful in providing lower density common connectors for test interconnect:

NI Cable Part Number	Ballard Cable Part Number	Photo	Description
784808-01	16036		LFH to (2) DB25 connectors. Mostly used in ARINC 429 applications. This cable brings all LFH signals out including the ARINC signals and IRIG.
784807-01	16037		LFH to (4) PL75 MIL-STD-1553 stub connections and (1) DB25 connector. Two channels of dual-redundant MIL-STD-1553 (Bus A and Bus B) on the PL75's with IRIG and core discretes on the d-sub connector.

The pinout of each board and corresponding mating cable are described in Chapter 7 of this manual. The cable part number and correct number of cables typically purchased with each is described in the following table:

Ballard P/N	**NI P/N**	**Cable Core A**	**Cable Core B**
LV-222-441-000	784802-01	784808-01	-
LV-222-442-000	784803-01	784808-01	-
LV-222-442-442	784804-01	784808-01	784808-01
LV-222-510-000	784796-01	784807-01	-
LV-222-511-000	784797-01	784807-01	-
LV-222-511-511	784798-01	784807-01	784807-01
LV-222-550-000	784799-01	784807-01	-
LV-222-555-000	784800-01	784807-01	-
LV-222-555-555	784801-01	784807-01	784807-01
LV-222-511-442	784805-01	784807-01	784808-01
LV-222-555-442	784806-01	784807-01	784808-01

This page intentionally blank.

7. CONNECTOR PINOUTS

The standard connector on the OB2 products is a 60-pin Molex® LFH™ receptacle. Each OB2 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, core discretes, and timing) are common to most modules and protocols. The databus signals are module specific,—their use and meaning depend on the protocol and functionality of the associated OB2 module. This chapter provides the information needed to connect to the individual modules through the LFH connector. Should your OB2 product have a different connector or a module not listed here, please consult other documentation provided with the product or call Customer Support (see Section 1.5).

Ballard offers a number of special cable assemblies to facilitate the use of their OB2 product line.

7.1 Interface Connector

The user interface connector on OB2 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-subminiature 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 OB2 products have the basic pin designations shown in Table 7.1. Note that, you must do wiring in pairs, 30 pairs in total. Especially on databus signals (labeled *BUSx$_1$x$_2$* in Table 7.1), use twisted pairs to avoid cross talk.

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	IRIG**/10MHZ
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

**BUSx$_1$x$_2$ where x$_1$ is the BUS number and x$_2$ is its polarity (Positive or Negative)*

***NOTE: The IRIG pins on cores A and B share common resources. This means that only one can be used at a time. There are additional IRIG configurations that are pin and core specific. Please refer to Table 4.6, Table 4.8, and Table 4.10 for details.*

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 OB2 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

Table 7.2 below lists the pin assignments for the MIL-STD-1553 modules. Be sure to follow the coupling and termination guidelines provided in Appendix A: Coupling and Termination.

Used on Modules				LFH	LFH	16036	LFH
5x0	**5x1 to 5x5**	**Name**	**Description**	**Pair #**	**Pin**	**Pin**	**Name**
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	P2-6	BUS4P
		ADIO1	Avionics DIO 1	3	6	P2-18	BUS4N
		ADIO2	Avionics DIO 2	12	24	P2-8	BUS5P
		ADIO3	Avionics DIO 3	12	25	P2-19	BUS5N
		ADIO4	Avionics DIO 4	4	9	P2-9	BUS6P
		ADIO5	Avionics DIO 5	4	8	P2-20	BUS6N
		ADIO6	Avionics DIO 6	11	22	P2-10	BUS7P
		ADIO7	Avionics DIO 7	11	23	P2-21	BUS7N
		ADIO8	Avionics DIO 8	18	37	P3-6	BUS12P
		ADIO9	Avionics DIO 9	18	36	P3-18	BUS12N
		ADIO10	Avionics DIO 10	27	54	P3-8	BUS13P
		ADIO11	Avionics DIO 11	27	55	P3-19	BUS13N
		ADIO12	Avionics DIO 12	19	39	P3-9	BUS14P
		ADIO13	Avionics DIO 13	19	38	P3-20	BUS14N
		ADIO14	Avionics DIO 14	26	52	P3-10	BUS15P
		ADIO15	Avionics DIO 15	26	53	P3-21	BUS15N

Table 7.2—Pinouts for MIL-STD-1553 Modules

7.3.2 ARINC 429

Table 7.3 lists the pin assignments for OB2 ARINC 429 modules.

*Note: The 441 and 442 models have a transmit and receive channel on each pin pair. You can control the receive channel via the API channel number listed in the **Channel** column. You can control the transmit channel via the receiver's API channel + 16. For example, the receive channel zero shares pins with the transmit channel 16.*

442 16R/T	**458** 8R/8T	**441** 8R/T	**Channel**	**Name**	**Polarity**	**LFH Pair #**	**LFH Pin**	**16036 Pin**	**LFH Name**
R/T	R	R/T	CH0†/ CH16 ††	CH0P	+	1	3	P2-3	BUS0P
				CH0N	–	1	2	P2-15	BUS0N
R/T	R	R/T	CH1 † / CH17 ††	CH1P	+	14	28	P2-2	BUS1P
				CH1N	–	14	29	P2-14	BUS1N
R/T	R	R/T	CH2 † / CH18 ††	CH2P	+	2	5	P2-4	BUS2P
				CH2N	–	2	4	P2-16	BUS2N
R/T	R	R/T	CH3 † /†† CH19 ††	CH3P	+	13	26	P2-5	BUS3P
				CH3N	–	13	27	P2-17	BUS3N
R/T	R	R/T	CH4 † /†† CH20 ††	CH4P	+	3	7	P2-6	BUS4P
				CH4N	–	3	6	P2-18	BUS4N
R/T	R	R/T	CH5 † / CH21 ††	CH5P	+	12	24	P2-8	BUS5P
				CH5N	–	12	25	P2-19	BUS5N
R/T	R	R/T	CH6/ CH22 ††	CH6P	+	4	9	P2-9	BUS6P
				CH6N	–	4	8	P2-20	BUS6N
R/T	R	R/T	CH7 † / CH23 ††	CH7P	+	11	22	P2-10	BUS7P
				CH7N	–	11	23	P2-21	BUS7N
R/T	T	n/a	CH8 † / CH24 ††	CH8P	+	16	33	P3-3	BUS8P
				CH8N	–	16	32	P3-15	BUS8N
R/T	T		CH9 † / CH25 ††	CH9P	+	29	58	P3-2	BUS9P
				CH9N	–	29	59	P3-14	BUS9N
R/T	T		CH10 † / CH26 ††	CH10P	+	17	35	P3-4	BUS10P
				CH10N	–	17	34	P3-16	BUS10N
R/T	T		CH11 † / CH27 ††	CH11P	+	28	56	P3-5	BUS11P
				CH11N	–	28	57	P3-17	BUS11N
R/T	T		CH12 † / CH28 ††	CH12P	+	18	37	P3-6	BUS12P
				CH12N	–	18	36	P3-18	BUS12N
R/T	T		CH13 † / CH29 ††	CH13P	+	27	54	P3-8	BUS13P
				CH13N	–	27	55	P3-19	BUS13N
R/T	T		CH14 † / CH30 ††	CH14P	+	19	39	P3-9	BUS14P
				CH14N	–	19	38	P3-20	BUS14N
R/T	T		CH15 † / CH31 ††	CH15P	+	26	52	P3-10	BUS15P
				CH15N	–	26	53	P3-21	BUS15N

Table 7.3—Pinouts for ARINC 429 modules

7.4 Standard Cables

Ballard sells a number of different cables that are useful for wiring to OB2 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 16036 Cable Assembly: LFH to Two 25-pin D-subs

A three-foot-long Y-cable 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 OB2 LFH connectors. As you can see from Table 7.1, there is symmetry between the upper and lower halves of the LFH connector. On the 16036 cable assembly, you can wire the upper half of the LFH connector to one D-sub and the lower half to the other D-sub, thus giving similar signals on the corresponding pins of both D-subs. The wire pairs on the 16036 are different from those on the 16035. Table 7.4 below shows the wiring for the 16036 cable.

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	IRIG*/10MHZ
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 7.4—Wiring Chart for 16036 Cable Assembly
(See next page for "" note)*

**NOTE: The IRIG pins on cores A and B share common resources. This means that only one can be used at a time. There are additional IRIG configurations that are pin and core specific. Please* refer to *Table 4.6*, *Table 4.8*, and *Table 4.10* for *details*.

7.4.2 MIL-STD-1553 Cable Assemblies

Ballard offers four standard cable assemblies for MIL-STD-1553 (see Table 7.5 below). The standard length is three feet.

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

Table 7.5—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. Table 7.6 shows the wiring details of the twinax cables.

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 7.6—Twinax Wiring on MIL-STD-1553 Cable Assemblies

The 25-pin female D-subminiature connector provides IRIG and core discrete signals, as shown in Table 7.7. Consequently, the recommended cable assemblies are 16037 for dual-channel and 16039 for single-channel MIL-STD-1553 OB2modules.

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	IRIG/10MHZ	47	11
10	GND	46	24
11	NC/5V	45	12
11	CGND	44	13

Braids connected shell to shell

Table 7.7—D-sub Connector Pinout for Cable Assemblies 16037 and 16039

This page intentionally blank.

APPENDIX A: COUPLING AND TERMINATION

Coupling and termination only apply to OB2 modules for MIL-STD-1553.

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 Ω. Terminate the databus 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 Ω). 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—Transformer Coupling to a Dual-Redundant Databus and Figure A.2—Direct Connection to a Dual-Redundant Databus.

Note: Some Ballard products have onboard termination resistors that you can switch-in manually or under software control. 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.

Both coupling methods 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. In rare cases, 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—Transformer Coupling to a Dual-Redundant Databus).
- The direct-coupled terminal has a higher turns ratio and has isolation resistors that you can connect directly to the main databus. Keep direct-coupled stubs should as short as possible (see Figure A.2—Direct Connection to a Dual-Redundant Databus).

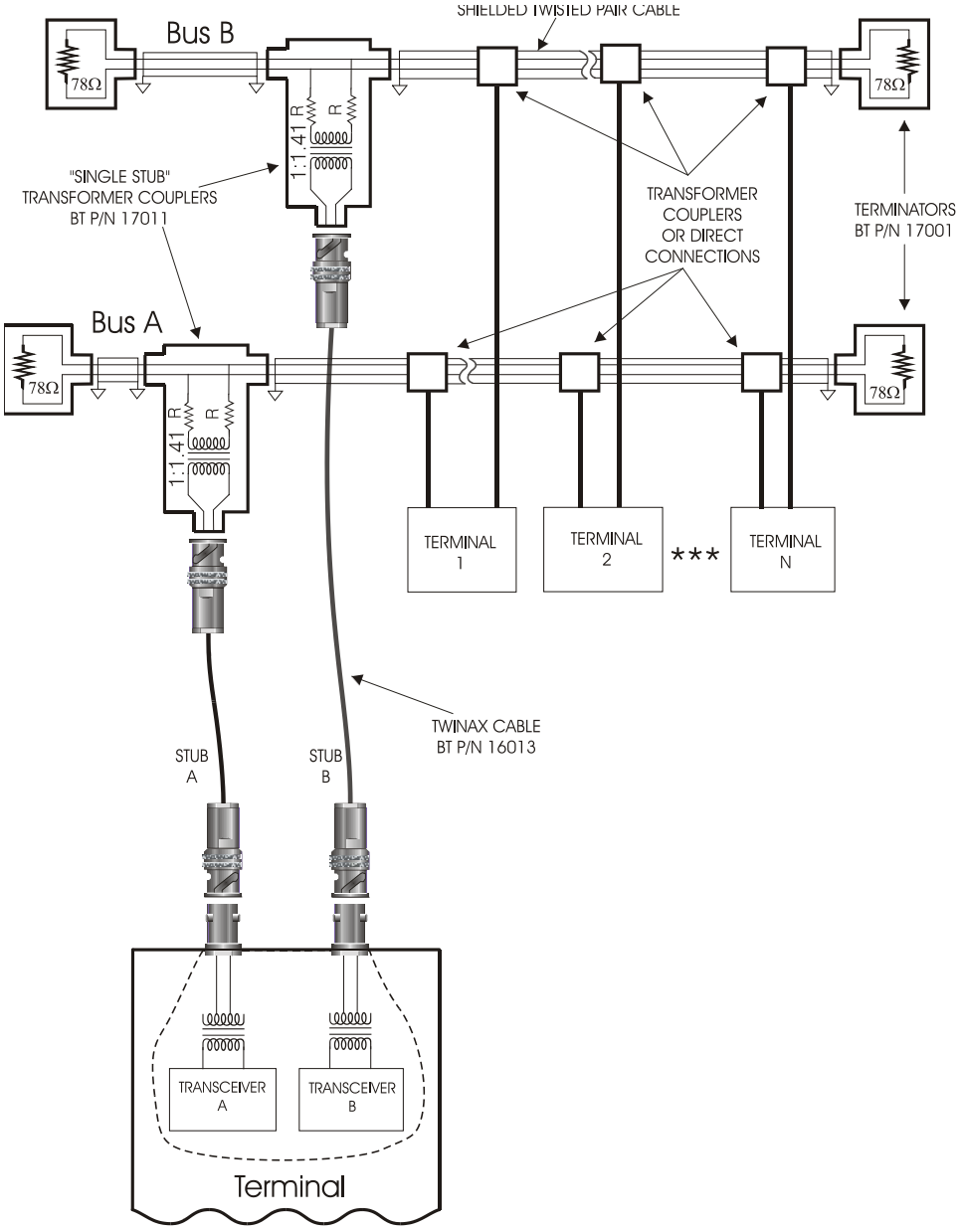


Figure A.1—Transformer Coupling to a Dual-Redundant Databus

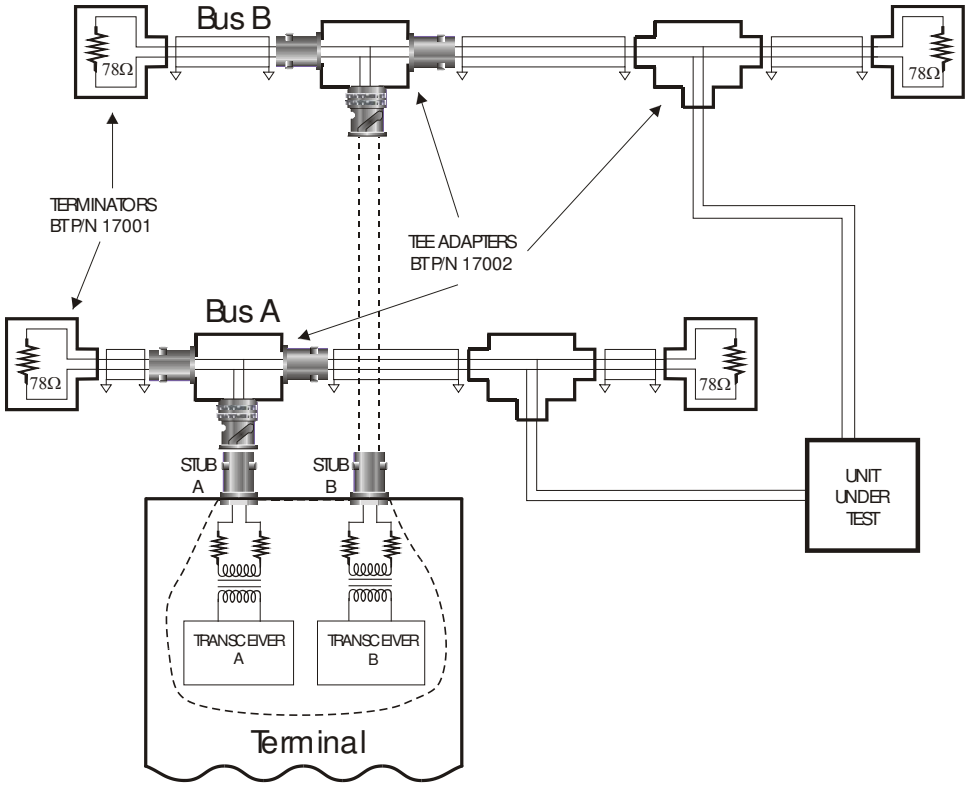


Figure A.2—Direct Connection to a Dual-Redundant Databus

This page intentionally blank.

APPENDIX B: SPECIFICATIONS

B.1 General

- 2 Core I/O sites
- 8 bidirectional TTL discrete I/O per core
- 2 user controlled LED indicators per core
- 64 MB memory per core (ECC)
- RoHS

B.2 Interfaces (Model Dependent)

MIL-STD-1553

- Up to 2 dual-redundant channels per core
- BC/RT/MON (single or multi-function)
- Hardware controlled transmit scheduling
- CH/TA/SA message filtering
- Sequential monitor and time stamping
- Error injection including MBZC shifting
- Playback with errors
- Amplitude controls
- 16 Open/GND avionics discrete I/O

ARINC 429 (Standard Module)

- Up to 8 Tx/8 Rx configurable channels per core
- Periodic and asynchronous messages
- Hardware controlled transmit schedule
- Hardware playback mode
- Sequential monitor and time stamping
- Programmable bit rate frequency
- Error detection and injection
  - Parity bit inversion
  - +/- bit count (8-33 bits)
  - Intermessage gap error

ARINC 429 (Advanced Module)

All features of the standard module plus:

- Control of each bus leg as open, ground, or normal
- Parametric waveform control (0-40 V differential pk-pk)

Advanced Timing (IRIG)

- 64-bit hardware time-tag (1 ns resolution)
- IRIG A/B input and output (AM, PWM)
  - Generate or synchronize timer
  - Synchronize hardware time-tags
- 10 MHz and PPS
  - Frame synchronization
  - Synchronize hardware time-tags

Software

- CoPilot available (Windows software)
- BTIDriver API available (for VB, C/C++, C#, LabVIEW)
- OS drivers: Windows included—call for availability of Linux, VxWorks®, or others

B.3 Environmental/Physical

Physical

- Size: PXIe: Standard 3U (100 x 160 mm)
- Weight: 3.5 lbs. (1.6 kg)

Physical Dimensions

See the following diagrams for the dimensions of the OB2 PXIe Card.

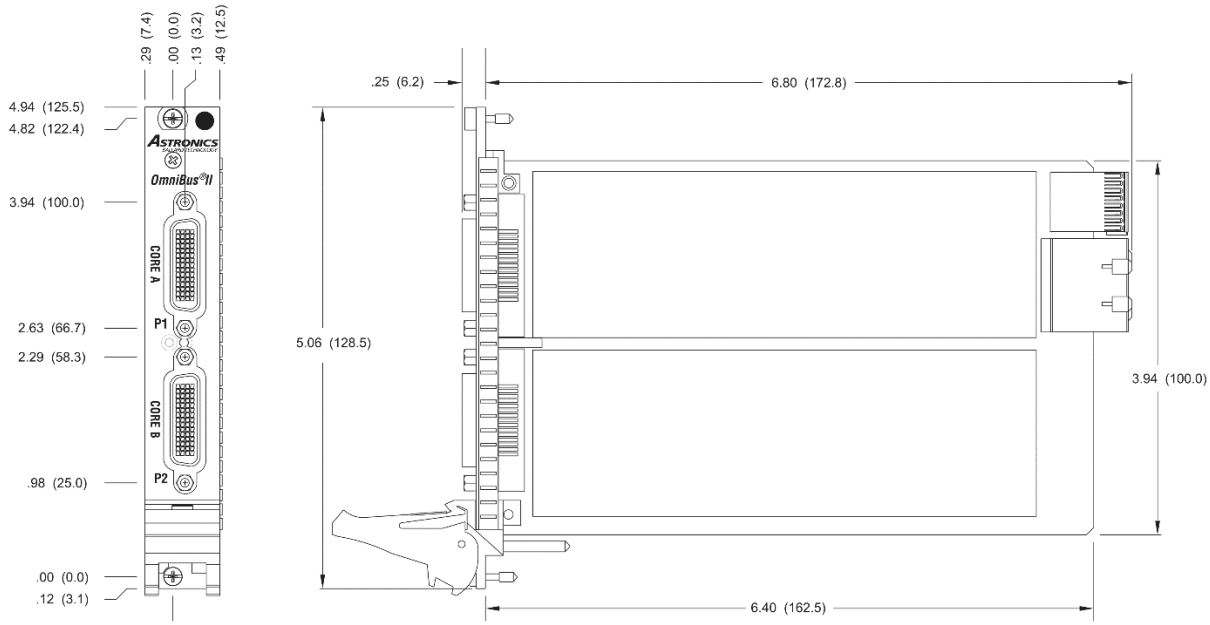


Figure B.1—Dimensions of OmniBus II PXIe Card

Environmental

- Component temperature: -40 to 85° C
- Storage temperature: -55 to 100° C
- Humidity: 0 to 95% (non-condensing)

Electrical

- Power: +3.3 and +12 VDC
- PCIe bus: x1 lane, bus mastering

Power

PXIe ARINC 429 (Model 222-442-442)

	Typical			Maximum		
Core A	Idle	Active	Active	Idle	Active	Active
Core B	Idle	Idle	Active	Idle	Idle	Active
3.3V	5.6 W	5.7 W	5.7 W	5.6 W	5.7 W	5.7 W
12V	3.8 W	5.1 W	6.3 W	3.8 W	10.6 W	18.0 W
Total	9.4 W	10.8 W	12.0 W	9.4 W	16.3 W	23.7 W

PXIe MIL-STD-1553 (Model 222-544-544)

	Typical			Maximum		
Core A	Idle	Active	Active	Idle	Active	Active
Core B	Idle	Idle	Active	Idle	Idle	Active
3.3V	5.8 W	5.8 W	5.9 W	5.8 W	5.9 W	5.9 W
12V	3.0 W	5.3 W	7.7 W	3.0 W	7.5 W	12.3 W
Total	8.8 W	11.1 W	13.6 W	8.8 W	13.4 W	18.2 W

Note: Technical specifications are subject to change without notice.

APPENDIX C: ERRATA SHEET

This errata sheet describes the known functional problems and any deviations from the technical specifications known at the release date of this manual.

Deviations

Errata Number	Section	Description	Applies to:		Status
			Hardware Revision	Software Revision	
1	6.3	Standard ARINC 429 Modules (42x/44x) present a low impedance on the 429 databus when power is removed from the OB2. This can result in an attenuated bus signal for other powered devices on the 429 databus. This errata does not apply to Advanced ARINC 429 Modules (45x).	42x/44x Modules Rev B and earlier *	N/A	Closed

**All OB2 Modules are labeled with the Core Model, Serial Number, and Hardware Revision (e.g. "442 1234C"). Additionally, BTITST32 may be used to programmatically determine the OB2 Module Revision by assigning a Card Number, selecting either "Core A" or "Core B", and locating the Hardware Revision in the "I/O module production information" section (e.g. "Revision C").*

This page intentionally blank.

APPENDIX D: REVISION HISTORY

The following revisions have been made to this manual:

Rev. A.0 Date: February 22, 2017

Initial release of this manual.

Rev. B.0 Date: September 15, 2017

Added Field Definitions. Corrected Installation Procedure. Removed references to 708, 717, Serial. Minor copy update.

Rev. C.0 Date: December 13, 2017

Added Section 1.7 Mean Time Between Failure, added Appendix B Specifications including power tables, and added Appendix C Errata Sheet with information regarding performance of 42x/44x Modules Rev B and earlier.

Rev. C.1 Date: February 1, 2018

Corrected cable cross reference errors in both tables in Section 6.4.

This page intentionally blank.