# OmniBus® PCI/cPCI
# USER'S MANUAL

INTERFACE CARD
to AVIONICS DATABUSES

September 14, 2007
Rev. A



**Ballard Technology**
11400 Airport Road, Suite 201
Everett, WA 98204 USA

Phone: (800) 829-1553   425-339-0281   Fax: 425-339-0915
E-mail: *support@ballardtech.com*
Web: *www.ballardtech.com*

MA141-091407

## COPYRIGHT NOTICE



## SAFETY WARNING

*Ballard products are of commercial grade and therefore are neither designed, manufactured, nor tested to standards required for use in critical applications where a failure or deficiency of the product may lead to injury, death, or damage to property. Without prior specific approval in writing by the president of Ballard Technology, Inc., Ballard products are not authorized for use in such critical applications.*

## 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

*CompactPCI® is a registered trademark of PCI Industrial Computer Manufacturers Group, Inc. (PICMG). PowerPC® is a registered trademark of International Business Machines Corporation (IBM). Molex® LFH™ is a trademark of Molex Inc. Windows® is a registered trademark of Microsoft Corporation. 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.*

**Ballard Technology**

Phone: 1-800-829-1553 425-339-0281 Fax: 425-339-0915
E-mail: *support@ballardtech.com* Web: *www.ballardtech.com*

# TABLE OF CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

This page intentionally blank.

# 1. INTRODUCTION

This manual is the user's guide for PCI and cPCI models of Ballard Technology's OmniBus® family of products. This guide introduces the OmniBus, discusses special OmniBus features, describes the installation process, and references programming alternatives.

## 1.1 OmniBus 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 (PCI, cPCI/PXI, VME, 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.

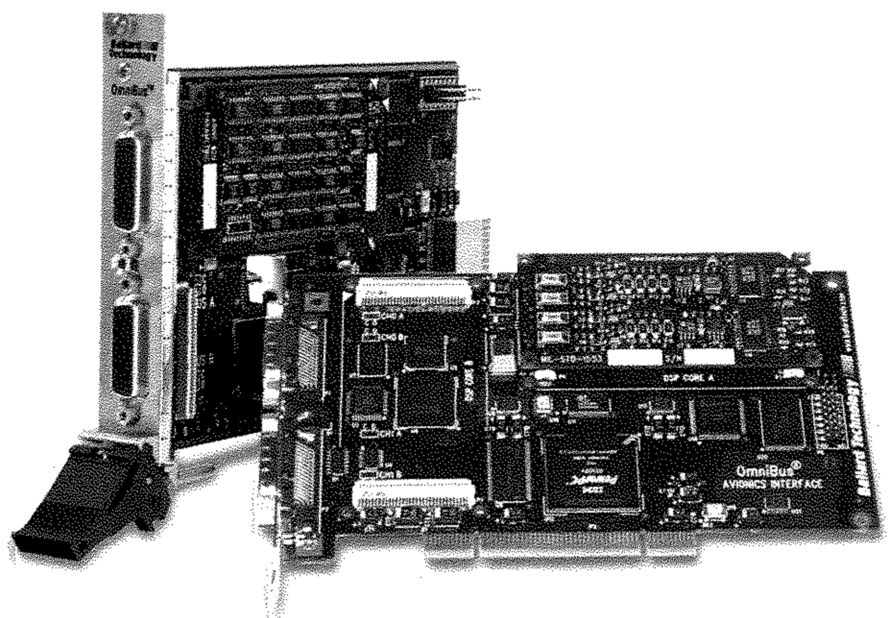

*Figure 1.1—OmniBus cPCI and PCI boards*

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 DSP to handle the channels and protocols attached to it (see Figure 1.2). The high channel count and mixed protocol capabilities can be fully exploited without the risk of overloading the DSP. Figure 1.2 illustrates the modular architecture of the OmniBus.

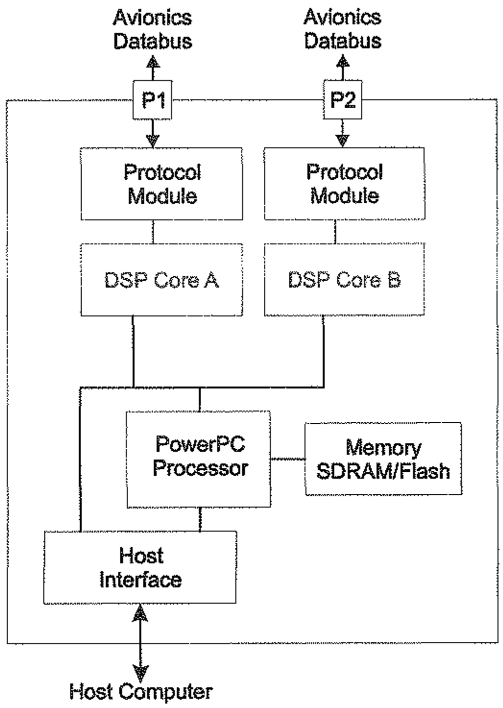


*Figure 1.2—The two-core architecture of OmniBus PCI and 3U cPCI products*

OmniBus products can be purchased with or without the capability to run user code on the PowerPC® processor to offload the host computer. An IRIG circuit allows channels, boards, and computers to be synchronized in time to each other and to external devices.

The easiest way to operate the OmniBus is with CoPilot®, Ballard Technology's graphical interface software. Alternately, software developers can write their own software applications using the included BTIDriver™ API (Application Program Interface).

## 1.2 OmniBus 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 and loading module-specific firmware for the DSPs.

> *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 PCI or cPCI 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 5.

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 Section 2.4.

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. The OmniBus supports 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. Please fill out and return the product registration card included in the front of this manual so that we can keep you informed of updates, customer services, and new product information.

# 2. INSTALLATION

This chapter explains the procedures for installing your PCI or cPCI OmniBus product. There are five steps to installation:

1. Print and review the driver installation procedure from the software distribution disk
2. Insert the OmniBus into an empty slot in your computer
3. Install the appropriate software driver
4. Test installation of card and drivers by running BTITST32.EXE
5. Connect the OmniBus to the databus(es)

After the installation steps are completed, the OmniBus is ready to communicate on the databus(es) using either CoPilot or a custom software application (see Chapter 3).

**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 computer.*

## 2.1 Step 1: Review the Driver Installation Procedure

The driver installation procedure varies, depending on the type of board and 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. Having a printed copy will facilitate the driver installation when you get to that step.

The driver installation instructions are in a README.TXT file on the distribution disk in a folder specific to your board and operating system. Follow these steps to locate and print the instructions:

1. Insert the disk in your drive and browse to the folder for your product (e.g., OMNICPCI for an OmniBus cPCI product)
2. Open the DRIVERS subfolder (e.g., OMNICPCI→DRIVERS)

3. Open the subfolder for your operating system (e.g., OMNICPCI→DRIVERS→WIN2K)
4. Print the README.TXT file in the operating system subfolder

Because your system may automatically detect the newly inserted OmniBus card and initiate driver installation, it is a good idea to have the distribution disk in your disk drive and a printed copy of the driver installation procedures in hand before you power up your computer. Aborting the automatic driver installation process before completion can necessitate manual installation of the driver.

## 2.2 Step 2: Insert the Card

OmniBus PCI and cPCI products are plug-and-play devices, so no jumpers or switches are used to configure them. Be sure to follow good ESD (electrostatic discharge) procedures (see the static discharge warning at the beginning of this chapter). To insert the card, do the following:

- Shut down your computer.
- **For PCI cards:** Insert the card(s) into an empty PCI slot and with a screw secure it to the case of your computer.
- **For cPCI cards:** With the injector handle in the down position, insert the card(s) into an empty peripheral slot (marked with a circle) in your CompactPCI (or PXI) system. 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 screw located at the top of the front panel.
- Restart your computer.

## 2.3 Step 3: Install the Driver

To install the driver software, follow the instructions printed from the README.TXT file in Step 1 (Section 2.1). The installation procedure copies several files into the host computer system and modifies the system registry.

If you encounter problems, have installation questions, or cannot find a folder for your operating system, contact Ballard Technology Customer Support at (800) 829-1553.

If you think that the drivers may already be installed, you can run the test program described in Section 2.4. If the OmniBus passes the test program, the drivers are properly installed. Once you have installed the drivers, you can usually remove and reinsert the OmniBus without having to reinstall them.

## 2.4 Step 4: Test the Installation

You can test the installation and functionality of your OmniBus by running the BTITST32.EXE program provided on the installation disk. This program analyzes the OmniBus within several seconds. If the program does not detect any faults with the interface or the OmniBus hardware, it displays a "passed test" message.

BTITST32.EXE also displays important information about your card, such as the assembly part number, configuration, and card number. If you have multiple cards installed, you can differentiate them by turning the LED on and off through the LED option in BTITST32.EXE (right click on the Device icon in the tree).

Card numbers are used by application software to uniquely identify each OmniBus card. They are assigned automatically by the host when the cards are installed. The card numbers may change depending on the position and number of cards used. Changing slots or removing cards when multiple cards are installed may change the card number assigned to a particular OmniBus card. As long as the number of cards is constant and the cards are not moved, the associated card numbers remain constant for all cards.

*Note: At any time you may use BTITST32.EXE or the Windows® Device Manager to determine and/or reassign the card numbers.*

If BTITST32 test detects a fault, it displays relevant fault information. When the fault message displays, follow the instructions on screen. If you need further assistance, call Ballard Technology at (800) 829-1553. A customer support engineer will interpret the fault and guide you through corrective steps.

## 2.5 Step 5: Connect the Databus(es)

OmniBus connections to the databus(es) depend on the module type and the special requirements for each protocol. To find the connector pinout, first determine the part number of the associated module (core), and then look in the appropriate table in Chapter 6 (as described in the following paragraphs).

Look at the labels on your OmniBus card to find its assembly part number. If you can see the individual modules, compare the numbers on each module with the corresponding numbers in the OmniBus assembly number. If there is a mismatch, proper operation cannot be expected. See Section 1.2 and Chapter 5 for more information on the meaning of OmniBus assembly part numbers.

Each core module has a 60-pin LFH connector dedicated to it; they are designated P1 for Core A and P2 for Core B. In Chapter 6, look up the pin assignments in the table associated with each module. Connect the databuses accordingly. Connectors and standard cables are described in Section 6.4. If you are using MIL-STD-1553 or ARINC 708, be sure to follow the coupling and termination guidelines discussed in Appendix A.

This page intentionally blank.

# 3. OPERATION

Software is used to control the OmniBus 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 OmniBus and utilize its powerful interface.

## 3.1 CoPilot

A PC with CoPilot and Ballard's OmniBus makes a powerful, low-cost databus analyzer/simulator. CoPilot interfaces directly with the OmniBus, 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.

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.

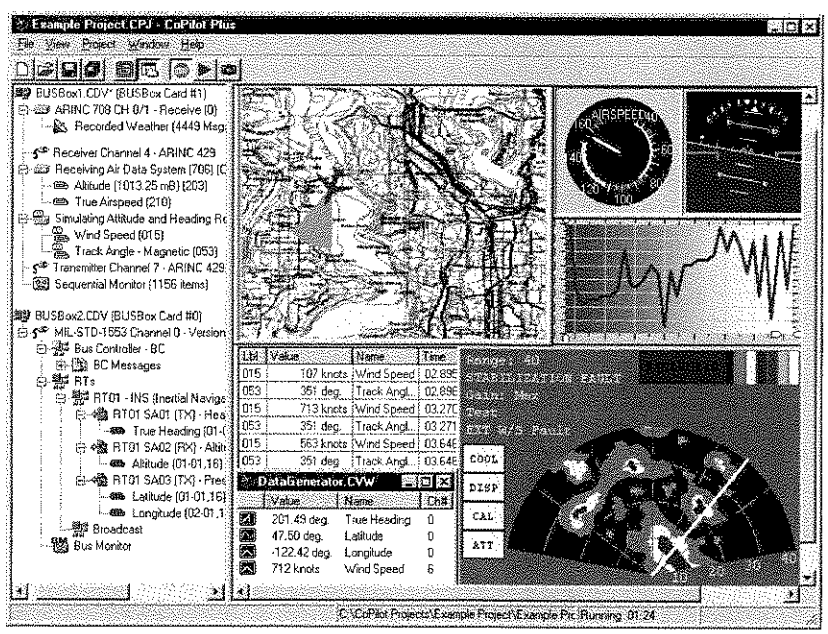

*Figure 3.1—Sample CoPilot screen*

## 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 the OmniBus 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 the OmniBus is found in the separate manuals for each protocol (e.g., *ARINC 429 Programming Manual for BTIDriver-Compliant Devices*).

# 4. OMNIBUS FEATURES

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

## 4.1 PowerPC

OmniBus products can be ordered with the capability to run user code on an IBM 405GP PowerPC® coprocessor. The BTIDriver API functions used in programming the host can also be used by the PowerPC. Thus, a program can be developed to run on the host and later ported to run on the PowerPC. One useful application for the PowerPC is to offload the host by performing computationally intensive manipulations on databus information, such as would occur in a simulator.

Code for the PowerPC can be either loaded and run each time the host application runs, or it can be saved in non-volatile memory so that it runs automatically at power on. A special family of PowerPC BTIDriver API functions is used to load and configure the PowerPC. Code is developed with user-provided software and hardware tools. Please consult Customer Support for a list of recommended compilers and debuggers.

## 4.2 IRIG Time

By default, each OmniBus core creates a 32-bit time-tag from a binary timer in the DSP. Each OmniBus core has an IRIG timer circuit that can be used in place of the binary timer. When a core is configured to use IRIG time, all time records are in IRIG time. The advantage of IRIG time is that it is human-readable in binary coded decimal (BCD) values in days, hours, minutes, seconds, etc. and can be synchronized to date and time of day. This allows timing data from all cores and external devices to be easily correlated. When IRIG time is enabled on a core, all time-tags associated with the core are 64-bit IRIG values that specify days down to microseconds. In time-tag fields the 32-bit binary timer value is replaced by the low 32 bits of IRIG time and an additional 32-bit field is allocated for the high 32 bits of IRIG time.

Each OmniBus core has its own IRIG timer circuit that generates the 64-bit IRIG time value. IRIG timers are free running and individually keep time until either set by software or synchronized to an IRIG signal. To synchronize to an IRIG signal, the IRIG timer must be configured in software as a slave. The source of the IRIG signal is the master, which can be an IRIG timer in another core or an external device. There can be only one master in the system. The IRIG signal from the master resynchronizes the IRIG slaves from one to ten times per second. Note that since the 64-bit IRIG time value has a resolution down to one microsecond and the clocks in the master and slave can vary in accuracy, there can be a slight underflow or overflow in the least significant digits of the IRIG value each time it is resynchronized.

The IRIG circuit in each OmniBus core can be configured as either a master or a slave. An internal IRIG bus connects all the cores within an OmniBus product. This internal IRIG bus goes to a bidirectional buffer that is wired to all the IRIG pins on the LFH connectors. Note that all these external IRIG signals are common (i.e., electrically connected together) across all LFH connectors on the OmniBus product. The IRIG pins on the LFH connectors are driven by the bidirectional buffer only when one of the IRIG core circuits is configured as a master. An IRIG core that is configured as an external slave will expect the IRIG signal to come from an external device (i.e., on the IRIG pin of one of the LFH connectors). Otherwise, if the IRIG core is an internal master or internal slave, it uses the internal bus.

There are a number of formats for IRIG time. OmniBus products use the formats indicated in Table 4.1. The characteristics of the external electrical interface to the IRIG pins are as shown in Table 4.2.

| | | |
|---|---|---|
| **Format** | A | 1000 pps |
| | B | 100 pps |
| **Modulation Frequency** | 0 | Pulse width coded |
| **Frequency/Resolution** | 0 | No carrier/index count interval |
| **Coded Expressions** | 2 | Master: BCD (output) |
| | 0, 1, 2, 3 | Slave: Uses only BCD field (input) |

*Table 4.1—IRIG formats used by OmniBus*

| | |
|---|---|
| **Input impedance (min)** | 12 kΩ |
| **Input level** | 0 to 5 volts |
| **Input level threshold** | 1.5 volts |
| **Output level** | 0 to 3 volts |
| **Output drive capability** | 40 mA |

*Table 4.2—Electrical characteristics of OmniBus IRIG signals*

## 4.3 Discretes

OmniBus PCI/cPCI products have both input and output discretes. There are six discretes per core module and four board discretes. All discrete inputs and outputs for both board and cores are TTL level. The discrete output driver is a 5-volt 74HCT244; the discrete input receiver is a 5-volt tolerant device with a high input impedance (10µA leakage current). Core discretes and board discretes are described briefly in the following sections.

### *4.3.1 Core Discretes, Syncs, and Triggers*

Core discretes can be used as general purpose I/O or as trigger inputs and sync outputs for processes running on the core module. Each core has three input and three output discretes, all wired to the core's associated LFH connector. Core discretes can be accessed through software running either on the host processor or on the PowerPC.

Ballard's BTIDriver API provides functions to read and write the core discretes. The parameter *dionum* in the API functions (BTICard_ExtDIORd and BTICard_ExtDIOWr) specifies which discrete to read or write. When DIO*n* (where *n* is a number) is described in other functions, it is synonymous with *dionum* (i.e., *n = dionum).*

A specific sync or trigger can use one or more of the 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 to 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 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 4.3 below shows the correlation between *dionum,* the output pin, and its hardware reference designator. The last column shows how these discretes are used as trigger inputs and sync outputs in the BTIDriver API functions. The names for core discretes are prefixed by CD (e.g., CDIN2).

| **Hardware Reference** | **LFH Pin P1 or P2** | **API *dionum*** | **API usage** |
|---|---|---|---|
| CDIN0 | 11 | 1 | Trigger A |
| CDIN1 | 21 | 2 | Trigger B |
| CDIN2 | 51 | 3 | Trigger C |
| CDOUT0 | 13 | 5 | Sync A |
| CDOUT1 | 19 | 6 | Sync B |
| CDOUT2 | 49 | 7 | Sync C |

*Table 4.3—Hardware versus software designation of core discretes*

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.

### 4.3.2 Board Discretes

Board discretes may be used only as general-purpose I/O and are only accessible through software running on the PowerPC. Each OmniBus has two input and two output board discretes, one each per LFH connector.

Table 4.4 below shows how the board discretes are distributed between the two core LFH connectors. The names for board discretes are prefixed by BD (e.g., BDOUT1).

| **Hardware Reference** | **LFH Pin** | **Core (number)** |
|---|---|---|
| BDIN0 | P1-41 | A (0) |
| BDIN1 | P2-41 | B (1) |
| BDOUT0 | P1-43 | A (0) |
| BDOUT1 | P2-43 | B (1) |

*Table 4.4—Designators for board discretes*

This page intentionally blank.

# 5. 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 appendix lists the part numbers for PCI/cPCI boards 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.*

## 5.1 OmniBus PCI and cPCI Boards

The PCI (short) and cPCI (3U) OmniBus products can host up to two modules, one per core. OmniBus PCI and cPCI carrier boards are listed in the table below:

| Part No. | Description |
|---|---|
| 111 | PCI w/ 1 core |
| 112 | PCI w/ 2 cores |
| 121 | 3U cPCI w/ 1 core |
| 122 | 3U cPCI w/ 2 cores |

## 5.2 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). The table below illustrates standard single and dual channel 1553 modules. In addition, any mix of levels may be ordered on a dual channel module.

| Part No. | CH0 Level | CH1 Level |
|---|---|---|
| 510 | A | – |
| 520 | B4 | – |
| 530 | B32 | – |
| 540 | C | – |
| 550 | D | – |
| 511 | A | A |
| 522 | B4 | B4 |
| 533 | B32 | B32 |
| 544 | C | C |
| 555 | D | D |

Each MIL-STD-1553 channel is available in five 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 D MIL-STD-1553 modules provide variable transmit amplitude and zero crossing distortion.

| Level Designator→ | A | B4 | B32 | C | D |
|---|---|---|---|---|---|
| Level Number used in PN→ | 1 | 2 | 3 | 4 | 5 |
| Number of Simultaneous Terminals | 1 | 4 | 32 | 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** | | | | | ✔ |

## 5.2.1 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.

## 5.2.2 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 1553A and 7.7 microseconds for 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.

## 5.2.3 Variable Transmit Amplitude

For OmniBus 1553 level D channels, the amplitude of the transmitted databus signal can be varied under software control. The 12-bit amplitude 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. 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.

### 5.2.4 Zero Crossing Distortion

On level D 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.

## 5.3 ARINC 429 Modules

The table below lists the I/O modules available with ARINC 429 channels.

| Part No. | 429 Channels | Parametrics |
|---|---|---|
| 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 | Amp./Freq. |
| 435* | 4R/4T | Amp./Freq. |
| 438 | 8R/8T | Amp./Freq. |

*R = receive and T = transmit*

** 435 also includes ARINC 717 channels*

Module 435 includes ARINC 717 channels (as described in Section 5.5). Note that each ARINC 429 receive channel on the 435 module is only available when the corresponding 717 receive channel is configured for biphase (see the wiring information in Section 6.3 for more details).

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). OmniBus ARINC 429 channels with parametric capability have variable transmit amplitude and user-configurable frequency.

### 5.3.1 Variable Transmit Amplitude

On OmniBus ARINC 429 modules with parametric capability, the transmit amplitude can be individually set in software for each channel. The user specifies the channel and a 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.

### 5.3.2 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 for the appropriate parameters for your module part number and desired frequency.

## 5.4 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*

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.

### 5.4.1 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 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.

### 5.4.2 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.

### 5.4.3 Variable Transmit Amplitude

On ARINC 708 modules with parametric capability, the amplitude of the transmitted databus signal can be varied under software control. 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.

## 5.5 ARINC 717 Modules

The table below lists the I/O modules available with ARINC 717 channels.

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

*R = receive and T = transmit.*

** 435 also includes ARINC 429 channels.*

Module 435 includes ARINC 429 channels (as described in Section 5.3). Note that each ARINC 429 receive channel on the 435 module is only available when the corresponding 717 receive channel is configured for biphase (see the wiring information in Section 6.3 for more details).

Both the 431 and 435 modules have four receivers and four transmitters. 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.

### 5.5.1 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.

# 6. CONNECTOR PINOUTS

The standard connector on OmniBus 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, discretes, and IRIG) 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 module. This chapter provides the information needed to connect to the individual modules through the LFH connector. Should your OmniBus 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.

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

## 6.1 Interface Connector

The user interface connector on OmniBus 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.

## 6.2 General Pinout

All OmniBus products have the basic pin designations shown in Table 6.1. Note that wiring is done in pairs; 30 pairs total. Especially on databus signals (labeled "*BUSxx*" in Table 6.1), be sure to use twisted pairs to avoid cross talk. The suffix on the designations for databus signals in Table 6.1 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 module, as indicated in Section 6.3.

| 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 | CDIN0A | | 41 | BDIN0/1 * |
| 6 | 12 | GND | 21 | 42 | GND |
| | 13 | CDOUT0A | | 43 | BDOUT0/1 * |
| 7 | 14 | CGND | 22 | 44 | CGND |
| | 15 | NC5VA | | 45 | NC5VA |
| 8 | 16 | GND | 23 | 46 | GND |
| | 17 | IRIG | | 47 | IRIG |
| 9 | 18 | GND | 24 | 48 | GND |
| | 19 | CDOUT1A | | 49 | CDOUT2A |
| 10 | 20 | GND | 25 | 50 | GND |
| | 21 | CDIN1A | | 51 | CDIN2A |
| 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 |

** 0 or 1, depending on core A or B*

*Table 6.1—General pin designations*

## 6.3 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 modules. Listings for the "16036 Pin" give the connector and pin number for the signal when a Ballard 16036 cable is used. See Section 6.4 for more information on cables.

### 6.3.1 MIL-STD-1553

The pin assignments for the MIL-STD-1553 modules are listed in Table 6.2 below. Be sure to follow the coupling and termination guidelines provided in Appendix A.

<table>
<tr><th colspan="2">Used on Modules</th><th rowspan="2">Name</th><th rowspan="2">Description</th><th rowspan="2">LFH Pair #</th><th rowspan="2">LFH Pin</th><th rowspan="2">16036 Pin</th><th rowspan="2">LFH Name</th></tr>
<tr><th>5x0</th><th>5x1 to 5x5</th></tr>
<tr><td rowspan="8">CH0</td><td rowspan="8">CH0</td><td>CH0AD</td><td>BUS A direct coupled (+)</td><td>14</td><td>28</td><td>P2-2</td><td>BUS1P</td></tr>
<tr><td>CH0ADR</td><td>BUS A direct coupled (–)</td><td>14</td><td>29</td><td>P2-14</td><td>BUS1N</td></tr>
<tr><td>CH0AX</td><td>BUS A transformer coupled (+)</td><td>1</td><td>3</td><td>P2-3</td><td>BUS0P</td></tr>
<tr><td>CH0AXR</td><td>BUS A transformer coupled (–)</td><td>1</td><td>2</td><td>P2-15</td><td>BUS0N</td></tr>
<tr><td>CH0BD</td><td>BUS B direct coupled (+)</td><td>2</td><td>5</td><td>P2-4</td><td>BUS2P</td></tr>
<tr><td>CH0BDR</td><td>BUS B direct coupled (–)</td><td>2</td><td>4</td><td>P2-16</td><td>BUS2N</td></tr>
<tr><td>CH0BX</td><td>BUS B transformer coupled (+)</td><td>13</td><td>26</td><td>P2-5</td><td>BUS3P</td></tr>
<tr><td>CH0BXR</td><td>BUS B transformer coupled (–)</td><td>13</td><td>27</td><td>P2-17</td><td>BUS3N</td></tr>
<tr><td rowspan="8">n/a</td><td rowspan="8">CH1</td><td>CH1AD</td><td>BUS A direct coupled (+)</td><td>29</td><td>58</td><td>P3-2</td><td>BUS9P</td></tr>
<tr><td>CH1ADR</td><td>BUS A direct coupled (–)</td><td>29</td><td>59</td><td>P3-14</td><td>BUS9N</td></tr>
<tr><td>CH1AX</td><td>BUS A transformer coupled (+)</td><td>16</td><td>33</td><td>P3-3</td><td>BUS8P</td></tr>
<tr><td>CH1AXR</td><td>BUS A transformer coupled (–)</td><td>16</td><td>32</td><td>P3-15</td><td>BUS8N</td></tr>
<tr><td>CH1BD</td><td>BUS B direct coupled (+)</td><td>17</td><td>35</td><td>P3-4</td><td>BUS10P</td></tr>
<tr><td>CH1BDR</td><td>BUS B direct coupled (–)</td><td>17</td><td>34</td><td>P3-16</td><td>BUS10N</td></tr>
<tr><td>CH1BX</td><td>BUS B transformer coupled (+)</td><td>28</td><td>56</td><td>P3-5</td><td>BUS11P</td></tr>
<tr><td>CH1BXR</td><td>BUS B transformer coupled (–)</td><td>28</td><td>57</td><td>P3-17</td><td>BUS11N</td></tr>
</table>

*Table 6.2—Pinouts for MIL-STD-1553 modules*

### 6.3.2 ARINC 429

The pin assignments for OmniBus ARINC 429 modules are listed in Table 6.3 below.

> *Note: Module 435 in the table below also includes ARINC 717 channels. See Table 6.5 for pinouts of the 717 channels on this module. ARINC 717 bipolar receive channels 4 through 7 and ARINC 429 receive channels 0 through 3 share the same four* ***bipolar*** *receivers, so each ARINC 429 receive channel is only available when the corresponding 717 receive channel is configured for* ***biphase****.*

| Channel Configurations by Part Number(s)* | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **421** 16R/0T | **422** 12R/4T | **423/438** 8R/8T | **424** 4R/12T | **425** 0R/16T | **426** 8R/0T | **427/434 /435*** 4R/4T | **428** 0R/8T | **Chan-nel** | **Name** | **Polar-ity** | **LFH Pair #** | **LFH Pin** | **16036 Pin** | **LFH Name** |
| R | R | R | R | T | R | R | T | CH0 | CH0P | + | 1 | 3 | P2-3 | BUS0P |
| | | | | | | | | | CH0N | – | 1 | 2 | P2-15 | BUS0N |
| R | R | R | R | T | R | R | T | CH1 | CH1P | + | 14 | 28 | P2-2 | BUS1P |
| | | | | | | | | | CH1N | – | 14 | 29 | P2-14 | BUS1N |
| R | R | R | R | T | R | R | T | CH2 | CH2P | + | 2 | 5 | P2-4 | BUS2P |
| | | | | | | | | | CH2N | – | 2 | 4 | P2-16 | BUS2N |
| R | R | R | R | T | R | R | T | CH3 | CH3P | + | 13 | 26 | P2-5 | BUS3P |
| | | | | | | | | | CH3N | – | 13 | 27 | P2-17 | BUS3N |
| R | R | R | T | T | R | n/a | T | CH4 | CH4P | + | 3 | 7 | P2-6 | BUS4P |
| | | | | | | | | | CH4N | – | 3 | 6 | P2-18 | BUS4N |
| R | R | R | T | T | R | n/a | T | CH5 | CH5P | + | 12 | 24 | P2-8 | BUS5P |
| | | | | | | | | | CH5N | – | 12 | 25 | P2-19 | BUS5N |
| R | R | R | T | T | R | n/a | T | CH6 | CH6P | + | 4 | 9 | P2-9 | BUS6P |
| | | | | | | | | | CH6N | – | 4 | 8 | P2-20 | BUS6N |
| R | R | R | T | T | R | n/a | T | CH7 | CH7P | + | 11 | 22 | P2-10 | BUS7P |
| | | | | | | | | | CH7N | – | 11 | 23 | P2-21 | BUS7N |
| R | R | T | T | T | n/a | T | n/a | CH8 | CH8P | + | 16 | 33 | P3-3 | BUS8P |
| | | | | | | | | | CH8N | – | 16 | 32 | P3-15 | BUS8N |
| R | R | T | T | T | n/a | T | n/a | CH9 | CH9P | + | 29 | 58 | P3-2 | BUS9P |
| | | | | | | | | | CH9N | – | 29 | 59 | P3-14 | BUS9N |
| R | R | T | T | T | n/a | T | n/a | CH10 | CH10P | + | 17 | 35 | P3-4 | BUS10P |
| | | | | | | | | | CH10N | – | 17 | 34 | P3-16 | BUS10N |
| R | R | T | T | T | n/a | T | n/a | CH11 | CH11P | + | 28 | 56 | P3-5 | BUS11P |
| | | | | | | | | | CH11N | – | 28 | 57 | P3-17 | BUS11N |
| R | T | T | T | T | n/a | n/a | n/a | CH12 | CH12P | + | 18 | 37 | P3-6 | BUS12P |
| | | | | | | | | | CH12N | – | 18 | 36 | P3-18 | BUS12N |
| R | T | T | T | T | n/a | n/a | n/a | CH13 | CH13P | + | 27 | 54 | P3-8 | BUS13P |
| | | | | | | | | | CH13N | – | 27 | 55 | P3-19 | BUS13N |
| R | T | T | T | T | n/a | n/a | n/a | CH14 | CH14P | + | 19 | 39 | P3-9 | BUS14P |
| | | | | | | | | | CH14N | – | 19 | 38 | P3-20 | BUS14N |
| R | T | T | T | T | n/a | n/a | n/a | CH15 | CH15P | + | 26 | 52 | P3-10 | BUS15P |
| | | | | | | | | | CH15N | – | 26 | 53 | P3-21 | BUS15N |

*R = receive and T = transmit*

** See Table 6.5 for the ARINC 717 pinouts on PN 435*

*Table 6.3—Pinouts for ARINC 429 modules*

### 6.3.3 ARINC 708

The pin assignments for ARINC 708 modules are listed in Table 6.4 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 810/820 | 811/822 | **Description** | **LFH Pair #** | **LFH Pin** | **16036 Pin** | **LFH Name** |
|---|---|---|---|---|---|---|
| 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 |

*Table 6.4—Pinouts for ARINC 708 modules*

## *6.3.4 ARINC 717*

The pin assignments for the ARINC 717 modules are listed in Table 6.5 below. All ARINC 717 channels can be either biphase or bipolar, but note that 717 receive channels have different pin assignments for biphase and bipolar. ARINC 717 transmit channels use the same pin assignments for both biphase and bipolar.

*Note: Module 435 in the table below also includes ARINC 429 channels. See Table 6.3 for pinouts of the 429 channels on this module. Bipolar ARINC 717 channels 4 through 7 and ARINC 429 receive channels 0 through 3 share the same four* ***bipolar*** *receivers, so each ARINC 429 receive channel is only available when the corresponding 717 receive channel is configured for* ***biphase****.*

| **431** 4R/4T | **435*** 4R/4T | **Channel** | **Polarity** | **LFH Pair #** | **LFH Pin** | **16036 Pin** | **LFH Name** |
|---|---|---|---|---|---|---|---|
| R (biphase) | R (biphase) | CH4 | + | 3 | 7 | P2-6 | BUS4P |
| | | | – | 3 | 6 | P2-18 | BUS4N |
| R (bipolar) | R (bipolar) | CH4 | + | 1 | 3 | P2-3 | BUS0P |
| | | | – | 1 | 2 | P2-15 | BUS0N |
| R (biphase) | R (biphase) | CH5 | + | 12 | 24 | P2-8 | BUS5P |
| | | | – | 12 | 25 | P2-19 | BUS5N |
| R (bipolar) | R (bipolar) | CH5 | + | 14 | 28 | P2-2 | BUS1P |
| | | | – | 14 | 29 | P2-14 | BUS1N |
| R (biphase) | R (biphase) | CH6 | + | 4 | 9 | P2-9 | BUS6P |
| | | | – | 4 | 8 | P2-20 | BUS6N |
| R (bipolar) | R (bipolar) | CH6 | + | 2 | 5 | P2-4 | BUS2P |
| | | | – | 2 | 4 | P2-16 | BUS2N |
| R (biphase) | R (biphase) | CH7 | + | 11 | 22 | P2-10 | BUS7P |
| | | | – | 11 | 23 | P2-21 | BUS7N |
| R (bipolar) | R (bipolar) | CH7 | + | 13 | 26 | P2-5 | BUS3P |
| | | | – | 13 | 27 | P2-17 | BUS3N |
| 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 6.3 for the ARINC 429 pinouts for PN 435*

*Table 6.5—Pinouts for ARINC 717 modules*

## 6.4 Standard Cables

Ballard sells a number of different cables that are useful for wiring to OmniBus 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 16035/10 is a ten-foot-long 16035.

### *6.4.1 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 6.1. The 16035 is useful for connecting an OmniBus product to a user-provided panel or other assembly.

### *6.4.2 PN 16036 cable assembly: LFH to two 25-pin D-subs*

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 LFH connectors. As can be seen from Table 6.1, 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-subs. The wire pairs on the 16036 are different from those on the 16035. Wiring for the 16036 cable is shown in Table 6.6 below.

| P2 Pair # | From P1 pin | To P2 pin | Name |
|---|---|---|---|
| 1 | 3 | 3 | BUS0P |
| 1 | 2 | 15 | BUS0N |
| 2 | 28 | 2 | BUS1P |
| 2 | 29 | 14 | BUS1N |
| 3 | 5 | 4 | BUS2P |
| 3 | 4 | 16 | BUS2N |
| 4 | 26 | 5 | BUS3P |
| 4 | 27 | 17 | BUS3N |
| 5 | 7 | 6 | BUS4P |
| 5 | 6 | 18 | BUS4N |
| 6 | 24 | 8 | BUS5P |
| 6 | 25 | 19 | BUS5N |
| 7 | 9 | 9 | BUS6P |
| 7 | 8 | 20 | BUS6N |
| 8 | 22 | 10 | BUS7P |
| 8 | 23 | 21 | BUS7N |
| 9 | 11 | 11 | CDIN0 |
| 9 | 10 | 23 | GND |
| 10 | 17 | 12 | IRIG |
| 10 | 19 | 24 | CDOUT1 |
| 11 | 13 | 13 | CDOUT0 |
| 11 | 15 | 25 | NC5V |
| 12 | 21 | 22 | CDIN1 |
| 12 | 20 | 1 | GND |
| 13 | 14 | 7 | CGND |

| P3 Pair # | From P1 pin | To P3 pin | Name |
|---|---|---|---|
| 1 | 33 | 3 | BUS8P |
| 1 | 32 | 15 | BUS8N |
| 2 | 58 | 2 | BUS9P |
| 2 | 59 | 14 | BUS9N |
| 3 | 35 | 4 | BUS10P |
| 3 | 34 | 16 | BUS10N |
| 4 | 56 | 5 | BUS11P |
| 4 | 57 | 17 | BUS11N |
| 5 | 37 | 6 | BUS12P |
| 5 | 36 | 18 | BUS12N |
| 6 | 54 | 8 | BUS13P |
| 6 | 55 | 19 | BUS13N |
| 7 | 39 | 9 | BUS14P |
| 7 | 38 | 20 | BUS14N |
| 8 | 52 | 10 | BUS15P |
| 8 | 53 | 21 | BUS15N |
| 9 | 41 | 11 | BDIN0/1 * |
| 9 | 40 | 23 | GND |
| 10 | 47 | 12 | IRIG |
| 10 | 49 | 24 | CDOUT2 |
| 11 | 43 | 13 | BDOUT0/1 * |
| 11 | 45 | 25 | NC5V |
| 12 | 51 | 22 | CDIN2 |
| 12 | 50 | 1 | GND |
| 13 | 44 | 7 | CGND |

** 0 or 1, depending on core A or B*
*Braids connected shell to shell*

*Table 6.6—Wiring chart for 16036 cable assembly*

## 6.4.3 MIL-STD-1553 cable assemblies

Ballard offers four standard cable assemblies for MIL-STD-1553 (see Table 6.7 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 6.7—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 6.8.

| 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 6.8—Twinax wiring on MIL-STD-1553 cable assemblies*

The 25-pin female D-subminiature connector provides IRIG and discrete signals, as shown in Table 6.9. 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 | CDIN0 | 11 | 1 |
| 1 | GND | 10 | 14 |
| 2 | CDIN1 | 21 | 2 |
| 2 | GND | 20 | 15 |
| 3 | CDIN2 | 51 | 3 |
| 3 | GND | 50 | 16 |
| 4 | BDIN | 41 | 4 |
| 4 | GND | 40 | 17 |
| 5 | BDOUT | 43 | 6 |
| 5 | GND | 42 | 19 |
| 6 | CDOUT0 | 13 | 7 |
| 6 | GND | 12 | 20 |
| 7 | CDOUT1 | 19 | 8 |
| 7 | GND | 18 | 21 |
| 8 | CDOUT2 | 49 | 9 |
| 8 | GND | 48 | 22 |
| 9 | IRIG1 | 17 | 10 |
| 9 | GND | 16 | 23 |
| 10 | IRIG2 | 47 | 11 |
| 10 | GND | 46 | 24 |
| 11 | NC5V | 45 | 12 |
| 11 | CGND | 44 | 13 |

*Braids connected shell to shell*

*Table 6.9—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 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 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 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).

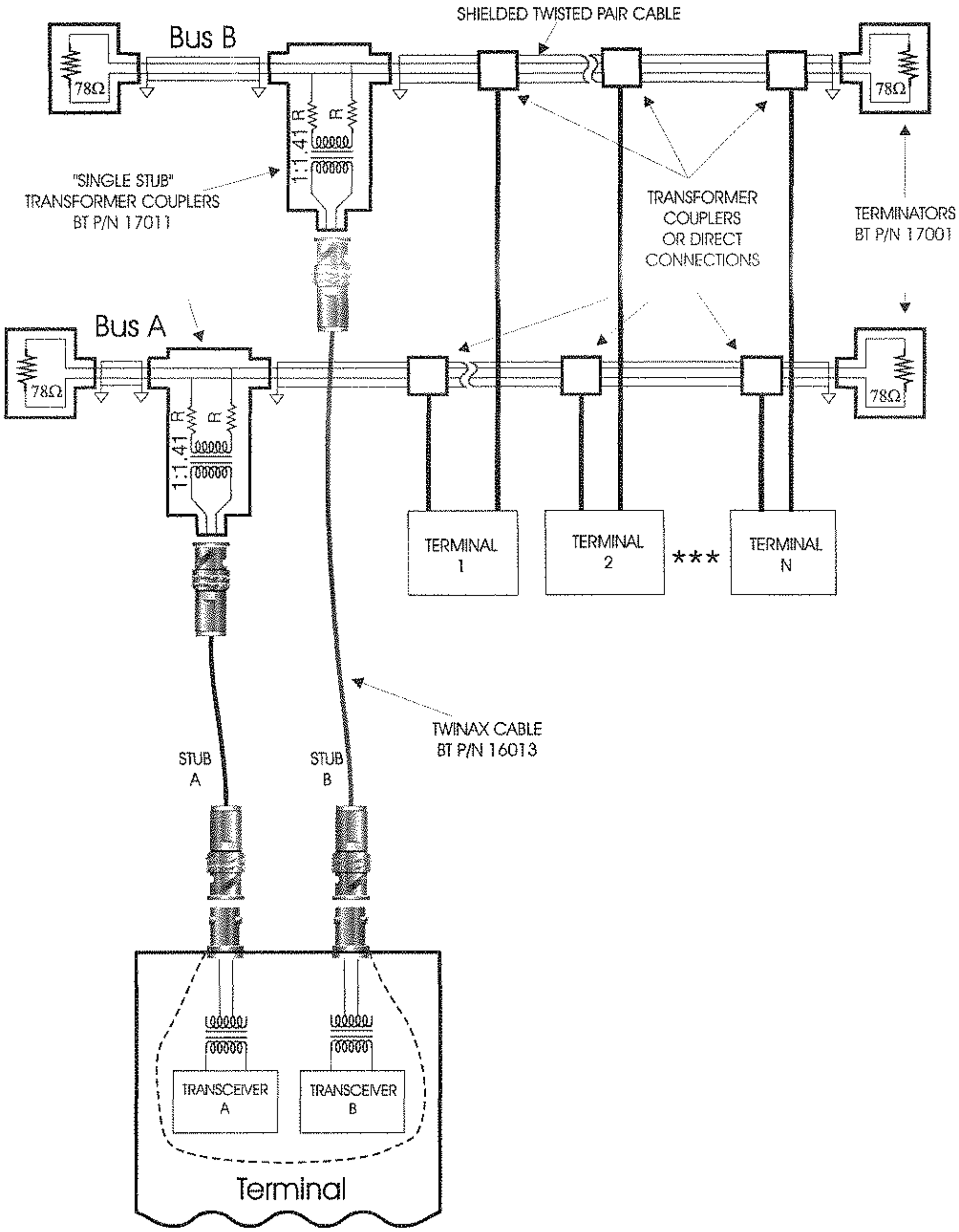


*Figure A.1—Transformer coupling to a dual-redundant databus*

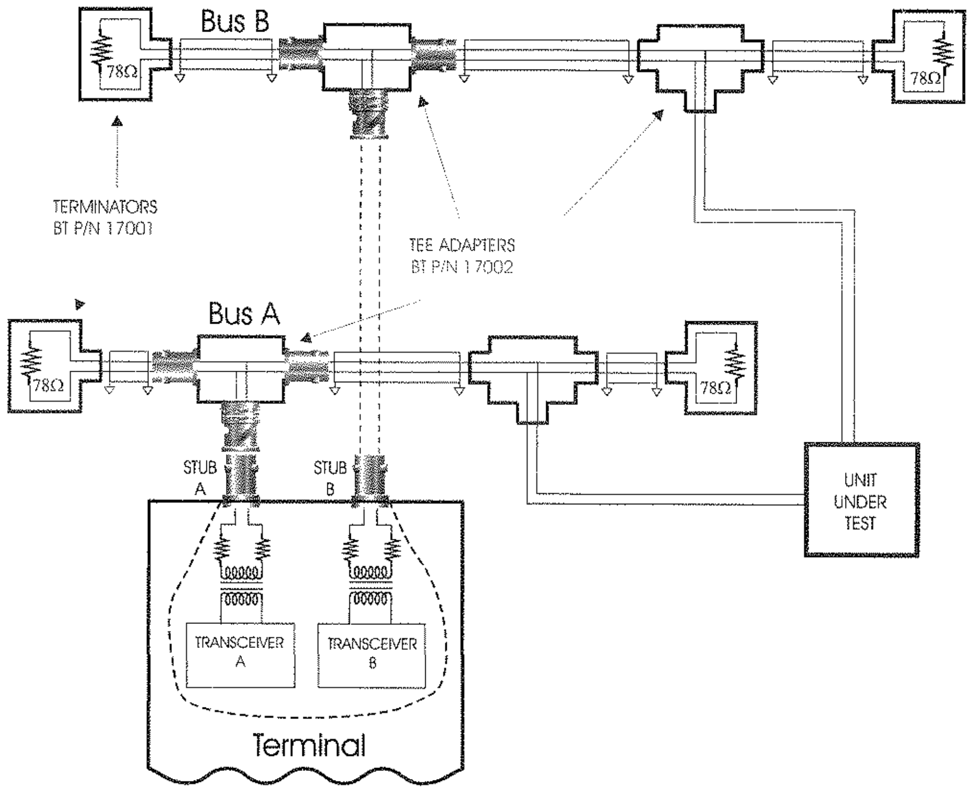


*Figure A.2—Direct connection to a dual-redundant databus*

This page intentionally blank.

# APPENDIX B: REVISION HISTORY

The following revisions have been made to this manual:

### *Preliminary Version 1    Date: January 31, 2003*

Preliminary release of this manual (covering only ARINC 429 and 708 modules).

### *Rev. A    Date: June 12, 2003*

Initial release of this manual.

This page intentionally blank.