

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

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

Visit our website - *Click HERE*

LP1553-3 and LC1553-3
USER'S MANUAL

PCI / cPCI to MIL-STD-1553
INTERFACE BOARD AND SOFTWARE

January 14, 2000
Rev. C.1

Copyright © 1998, 2000
by

Ballard Technology
3229A Pine Street
Everett, WA 98201 USA

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

MA113-011400

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS

1. INTRODUCTION 1-1

1.1 MIL-STD-1553 1-1

1.2 The LP1553-3 and LC1553-3 1-1
1.2.1 Hardware 1-1
1.2.2 Software 1-2

1.3 About This User's Manual 1-3

1.4 Updates to Software and Documentation 1-4

1.5 Technical Support 1-4

2. INSTALLATION 2-1

2.1 Card Insertion 2-2

2.2 Driver Installation 2-2
2.2.1 Windows 95 and Windows 98 Automatic Driver Installation 2-2
2.2.2 Windows 95 and Windows 98 Manual Driver Installation 2-3
2.2.3 Windows NT Driver Installation 2-4

2.3 Testing the Lx1553-3 Installation 2-4

2.4 Multiple Card Installation 2-4

2.5 Connecting to the MIL-STD-1553 Databus 2-6
2.5.1 Transformer Versus Direct Coupling 2-6
2.5.2 Bus Termination 2-9

3. DEVELOPING SOFTWARE FOR THE LX1553-3 3-1

3.1 Skeleton Lx1553-3 Application 3-1

3.2 Lx1553-3 Configuration Overview 3-2
3.2.1 BC Configuration 3-3
3.2.2 RT Configuration 3-4
3.2.3 Monitor Configuration 3-4

3.3 Lx1553-3 Runtime Overview 3-5

4. BASIC SINGLE TERMINAL SIMULATION 4-1

4.1 Simulating the BC: Unscheduled Messages 4-1

4.2 Simulating the BC: Scheduled Messages 4-2

4.3 Simulating an RT 4-5

4.4 Monitoring the Bus 4-6

4.5 Summary 4-7

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. MULTI-TERMINAL SIMULATION 5-1

5.1 Simultaneous BC, RT and Monitor Simulation 5-1

5.2 Broadcast Data and RT Address 31 5-2

6. ERROR INJECTION 6-1

6.1 Error Injection Overview 6-1
6.1.1 Defining the Error 6-1
6.1.2 Tagging Messages 6-2
6.1.3 Enabling Error Injection 6-2

6.2 Errors in BC Transmissions: Unscheduled Messages 6-3

6.3 Errors in BC Transmissions: Scheduled Messages 6-4

6.4 Errors in RT Transmissions 6-5

7. ADVANCED FEATURES 7-1

7.1 True Dual Port Ram 7-1

7.2 List Buffering 7-1
7.2.1 List Buffer Modes 7-1
7.2.2 List Buffer Operation 7-2

APPENDIX A: FUNCTION REFERENCE

APPENDIX B: D-SUB CONNECTORS

APPENDIX C: EXTERNAL SIGNALS

APPENDIX D: SPECIFICATIONS

APPENDIX E: UPDATES

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LIST OF FIGURES

Fig. 1-1 Photographs of the LP1553-3 and LC1553-3 1-1
Fig. 1-2 A sample CoPilot 1553 screen 1-3
Fig. 2-1 Coupling selection jumpers 2-6
Fig. 2-2 Schematic of transformer coupling 2-7
Fig. 2-3 Cable assembly for transformer coupling of Lx1553-3 to unit under test 2-7
Fig. 2-4 Schematic of direct coupling 2-8
Fig. 2-5 Cable assembly for direct coupling of Lx1553-3 to unit under test 2-8
Fig. 2-6 Termination resistor selection jumpers for direct coupling 2-9
Fig. 3-1 Skeleton Lx1553-3 application 3-1
Fig. 3-2 Bus controller terminal configuration 3-3
Fig. 3-3 Remote terminal configuration 3-4
Fig. 3-4 Control of data transfer while Lx1553-3 is active 3-5
Fig. 4-1 Example 1: Unscheduled message transmission 4-1
Fig. 4-2 Typical BC message schedule 4-3
Fig. 4-3 Example 2: Scheduled message transmission 4-4
Fig. 4-4 Example 3: RT simulation 4-5
Fig. 4-5 Example 4: Bus monitoring 4-7
Fig. 5-1 Example 5: Multi-terminal simulation 5-1
Fig. 6-1 Example 6: Errors in unscheduled BC transmissions 6-3
Fig. 6-2 Example 7: Errors in scheduled BC transmissions 6-4
Fig. 6-3 Example 8: Errors in RT transmissions 6-5
Fig. B-1 Pinouts of LC1553-3 D-sub connector B-1
Fig. C-1 Pinouts of the external signals header C-1

LIST OF TABLES

Table 1-1 Features of different Lx1553-3 versions 1-2
Table 2-1 Multiple Lx1553-3 card installation 2-5
Table 6-1 L13_ErrorDefine parameters according to error type 6-1
Table A-1 Function summary A-2
Table A-2 Errors summary A-4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COPYRIGHT NOTICE

Copyright ©1998, 2000 by Ballard Technology, Inc. Ballard Technology's permission to copy and distribute this manual is for the purchaser's private use only and is conditioned upon purchaser's use and application with the LP1553-3 or LC1553-3 hardware that was shipped with this manual. No commercial resale or outside distribution rights are allowed by this notice. This material remains the property of Ballard Technology, Inc. All other rights reserved by Ballard Technology, Inc.

LIMITED WARRANTY

Ballard Technology (the "Manufacturer") warrants its product to be free from defects in materials and workmanship under normal use for a period of one year from delivery. Should the product fail to be in good working order during the one year period the Manufacturer's sole obligation shall be at its option, to repair, correct or replace the product, or, at Manufacturer's sole discretion, to refund amounts paid for such product upon its return to a location designated by Manufacturer in the United States. Manufacturer makes no other warranties, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Manufacturer's entire liability and Customer's exclusive remedy for damages from any cause whatsoever, including, but not limited to, nonperformance or misrepresentation, and regardless of the form of action, shall be limited to the purchase price for the specific product that caused the damages or that are the subject matter of, or directly related to, the cause of action. In no event will Manufacturer be liable for damages caused by Customer's negligence, or for any lost profits, lost savings or other incidental or consequential damages, even if Manufacturer has been advised of the possibility of such damages, or for any other claim against Customer by any other party. This limited warranty does not cover products damaged by accident, disaster, misuse, abuse or unauthorized modifications.

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

Windows™, Windows 95™, Windows 98™, Windows NT™ and MS-DOS™ are trademarks of Microsoft Corporation. CompactPCI™ is a trademark of PCI Industrial Computers Manufacturers Group (PICMG). All other product names or trademarks are property of their respective owners.

Ballard Technology//////// 3229A Pine Street, Everett, WA 98201 USA
Phone: (800) 829-1553 425-339-0281 Fax: 425-339-0915
E-mail: support@ballardtech.com Web: *www.ballardtech.com*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1. INTRODUCTION

This manual covers both the LP1553-3 and the LC1553-3. Electrically they are nearly identical. Software written for one will work with the other. Throughout this manual, they are referred to together as the Lx1553-3. Wherever explicit differences apply each product is referred to individually as either the LP1553-3 or the LC1553-3.

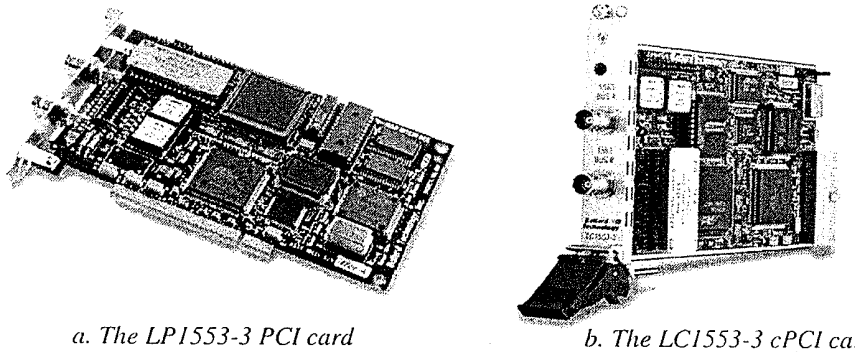
1.1 MIL-STD-1553

MIL-STD-1553 is a standard defining a local area network (LAN) originally developed for and widely used on military aircraft. This digital, command-response, time-division multiplexing network protocol is also used in many other military and commercial applications where fast, positive control is required. The standard defines the handshaking, data formats and timing requirements of the protocol as well as the electrical characteristics of the bus and the terminals' interface electronics. Familiarity with the MIL-STD-1553 databus protocol is essential to understanding the operation of the Lx1553-3, and it is assumed throughout this manual that the user is acquainted with this protocol.

1.2 The LP1553-3 and LC1553-3

1.2.1 Hardware

The Lx1553-3 family of MIL-STD-1553 intelligent interface cards share a common DSP-based core design. The LP1553-3 interfaces a computer's *PCI* bus to a dual-redundant MIL-STD-1553 databus. The LC1553-3 is identical to the LP1553-3 except the LC1553-3 card is in the *CompactPCI* form factor. The standard LC1553-3 is a 3U-sized card that may also be used in a 6U chassis with an optional 6U front panel adapter. Custom software written for one card will run both LP1553-3 and LC1553-3 cards without any code modifications (provided similar card configurations).



a. The LP1553-3 PCI card *b. The LC1553-3 cPCI card*

Fig. 1-1 Photographs of the LP1553-3 and LC1553-3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The primary capabilities of the four currently available Lx1553-3 versions are summarized in Table 1-1.

	Versions			
	A	B4	B32	C
Number of Simultaneous Terminals	1	4	32	32
Monitor	✔	✔	✔	✔
Filtering on TA, SA		✔	✔	✔
Concurrent terminal monitoring				✔
Error Injection				✔
On-Board Static RAM	256KB	256KB	256KB	1MB
True Dual-Port RAM	64KB	64KB	64KB	64KB
Bundled Software	✔	✔	✔	✔
CoPilot 1553 compatible	✔	✔	✔	✔

Table 1-1 Features of different Lx1553-3 versions

1.2.2 Software

Ballard Technology has a variety of software support for the Lx1553-3. This includes drivers, API libraries and a Graphical User Interface (GUI) for a variety of operating systems. Windows 95, Windows 98 and Windows NT drivers and API libraries are provided with the Lx1553-3 hardware. The API library of the LP1553-3 is identical to the LC1553-3, so code written for the LP1553-3 can run both LP1553-3 and LC1553-3 families of cards without any code modifications (provided similar card configurations). Software is continually in development, so call Ballard Technology for the latest information on software availability.

Available separately is a powerful, easy-to-use graphical software application called CoPilot 1553. Although some specialized applications may require custom software, CoPilot 1553 greatly simplifies the simulation and testing of a 1553 databus. Using CoPilot 1553 and your Lx1553-3, you can set up and run BC schedules, simulate RTs or engage the bus monitor with a few clicks of the mouse. You can also observe and change data while the bus is running. The tools and filters built into the monitor mode of CoPilot 1553 assist you in locating and analyzing bus activity. With CoPilot 1553, there is little need to code custom software to access the capabilities of the new generation of 1553 boards, including the Lx1553-3. The best way to discover how CoPilot 1553 can increase your productivity is to try it. For a free evaluation copy of CoPilot 1553, please contact customer support at (800) 829-1553.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

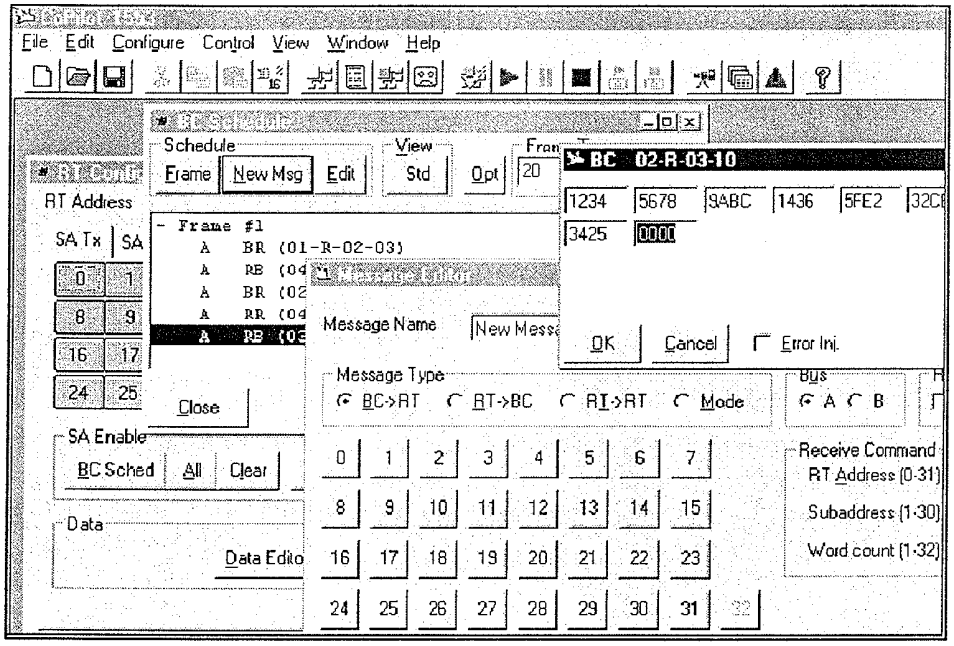


Fig. 1-2 A sample CoPilot 1553 screen

1.3 About This User's Manual

This manual shows how to develop WIN32 applications for the Lx1553-3 A, B and C versions using the C-language Lx1553-3 API (Application Program Interface). The API is a library of functions your program calls to configure and control the Lx1553-3. This simplifies software development by relieving you of hardware-level details. Source code for sample programs is also included on disk with the API.

With minor exceptions, the content of this manual also applies to other languages and operating systems. This manual assumes that you are familiar with the essentials of compiling, linking and running programs in C. It is also assumed that you are familiar with the MIL-STD-1553 protocol.

The following conventions are observed throughout this manual:

- Function names in the Lx1553-3 API are printed in a **bold courier** font and are all prefixed by "L13_" (e.g., **L13_RTConfig**).
- A small "h" suffix indicates hexadecimal values (e.g., F01Ch).
- Constants defined in the Lx1553-3 header file are written in all-capital letters (e.g., ERRDEF_PAR).
- The symbol "??" is used in function names in this manual to indicate a category of functions with similar uses or attributes. These characters should be replaced by a category prefix or suffix in an actual function call (e.g., **L13_MsgData??** to represent **L13_MsgDataRd** and **L13_MsgDataWr**).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.4 Updates to Software and Documentation

At Ballard Technology we take pride in a high-quality, reliable product that meets the needs of its users, and improvement is a continuing process. Periodically, updates to documentation (in the form of Application Notes) may be issued as well as updates to software. Please fill out and return the Product Registration Card included in the front of this manual so that we can keep you informed of documentation and software updates, customer services, and new product information. Contact Ballard Technology Technical Support for the latest software and documentation updates.

1.5 Technical Support

Ballard Technology offers free technical support before and after the sale. Our hours are 9:00 AM to 5:00 PM Pacific Time, though support and sales engineers are often available outside those hours. We can be reached at (800) 829-1553, (425) 339-0281 or on the Web at *www.ballardtech.com*. You can also send e-mail to *support@ballardtech.com*. We invite your questions and comments on our products.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. INSTALLATION

This chapter explains the procedures for installing your Lx1553-3. There are four phases to installing the Lx1553-3:

1. Insert the Lx1553-3 into an empty slot in your Computer
2. Install drivers for your Lx1553-3
3. Test installation of card and drivers by running L13TST32.EXE
4. Connect the Lx1553-3 to your MIL-STD-1553 databus

The following system configuration is the minimum required for Lx1553-3 installation:

- Microsoft Windows 95, Windows 98, or Windows NT operating system
- PCI or CompactPCI system with at least one peripheral slot free
- One 3.5-inch high-density disk drive (or other media device)
- 1 MB of available hard-disk space

When you have completed the installation steps in this chapter, your Lx1553-3 is ready to communicate on the 1553 databus using CoPilot 1553 or your custom software application. Chapter 3 outlines the tasks an application program performs to configure and control the Lx1553-3, and Chapter 4 presents sample single terminal programs. If you plan to use CoPilot 1553, please consult the documentation provided with that program.

WARNING

Static Discharge

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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.1 Card Insertion

The Lx1553-3 is a Plug-and-Play device, so no jumpers or switches are used to configure the PCI interface. However, some jumpers on headers are used for 1553 bus termination and coupling selection as described in Section 2.5. Installation of the Lx1553-3 is simple:

1. Verify the jumper settings for bus termination selection before continuing. See Section 2.5 for more information.
2. Shut down your computer.
3. **For LP1553-3:** Insert the LP1553-3 card(s) into an empty PCI slot and secure the card bracket to the case of your computer.
   For LC1553-3: Insert the LC1553-3 card(s) with the injector handle down into an empty peripheral slot (marked with a circle) in your CompactPCI system. When the bottom of the handle is pressed against the subrack's horizontal rail, move the injector handle up to complete the insertion of the card. Secure the LC1553-3 front panel collar screw (located at the top of the front panel) to the chassis.
4. Restart your computer.

Your operating system is now prepared to have the Lx1553-3 drivers installed. The installation process may begin automatically or may require manual installation as mentioned in the next section.

2.2 Driver Installation

This section explains the procedures for installing drivers for single or multiple cards into a computer running Windows 95, Windows 98 or Windows NT (4.0 and older). The Lx1553-3 requires the following files from the Lx1553-3 driver installation disk:

File Name	Windows 95	Windows 98	Windows NT
BTIKRNL.DLL	✓	✓	✓
BTIUNIV.SYS			✓
BTIUNIV.VXD	✓	✓	
L13W32.DLL	✓	✓	✓
L13WNT.REG			✓

Windows 95 and Windows 98 installation of the Lx1553-3 card is described in Sections 2.2.1 and 2.2.2. Windows NT driver installation of the Lx1553-3 card is described in Section 2.2.3. For multiple card installation, refer to Section 2.4 after completing the steps in this section. Note, if you are installing from media other than a floppy disk, refer to the README.TXT file in the root directory for the location of driver files.

2.2.1 Windows 95 and Windows 98 Automatic Driver Installation

Since the Lx1553-3 is a Plug-and-Play device, the first time Windows 95 or Windows 98 is started with your Lx1553-3 installed, the Lx1553-3 card is detected and the driver installation process should begin automatically. If you are not prompted to install drivers for the Lx1553-3 before Windows finishes load-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ing and the drivers are not yet installed, skip the rest of this section and go to Section 2.2.2. When Windows detects the Lx1553-3 and prompts you to install the drivers, follow the appropriate steps for either Windows 95 or Windows 98. When driver installation is complete, skip to section 2.3.

Windows 95

1. Select the **Driver Provided by Manufacturer** option.
2. Click **Next**.
3. Insert the Lx1553-3 installation disk into drive A (or other media location).
4. Type the path containing the driver files for the Lx1553-3 (**A:\WIN95** for floppy disk installation) in the **Copy manufacturer's files from** field.
5. Click **OK** to complete the installation process.
6. Skip to Section 2.3 to test installation.

Windows 98 (or Windows 95 OSR2)

1. Click **Next** when the **Add New Hardware Wizard** starts.
2. Select the **Display a list of all the drivers in a specific location** option and click **Next**.
3. When prompted to select the device type, ignore this step and press **Next**.
4. Click **Have Disk** and insert the media containing the Lx1553-3 installation files (if the data is not located on the hard disk).
5. Type the path containing the driver files for the Lx1553-3 (**A:\WIN95** for floppy disk installation) and press **OK**.
6. Select the Lx1553-3 and press **Next**.
7. Press **Next** again to search the specified location for the driver files.
8. Click **Finish** to complete the installation process.
9. Skip to Section 2.3 to test installation.

2.2.2 Windows 95 and Windows 98 Manual Driver Installation

Manual installation of drivers is only required if the Windows 95 or 98 installation procedure did not automatically begin, the installation was unsuccessful, or drivers need reinstallation.

Perform the following steps to manually install the drivers:

1. Click the **Start** button, point to **Settings** and click the **Control Panel** icon.
2. Double-click the **Add New Hardware** icon.
3. When the first **Add New Hardware Wizard** window opens, click **Next**.
4. For Windows 98 installations only, click **Next** to automatically search for Plug and Play hardware.
5. Select **No** so Windows does not search for new hardware and click **Next**.
6. Select **Other devices** from the **Hardware types** list and click **Next**.
7. Click **Have Disk** and insert the media containing the Lx1553-3 installation files (if the data is not located on the hard disk).
8. Type the path containing the driver files for the Lx1553-3 (**A:\WIN95** for floppy disk installation) in the **Copy manufacturer's files from** field of the **Install From Disk** window.
9. Click **OK** to return to the **Add New Hardware Wizard**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10. **Select** the Lx1553-3 card that you are installing from the **Models** list and click **Next**.
11. Click **Next** to install drivers.
12. Click **Finish** to complete driver installation.
13. Skip to Section 2.3 to test installation.

2.2.3 Windows NT Driver Installation

To install drivers for Windows NT, you will need the Ballard Technology "L13 Windows NT Driver" disk. Perform the following steps to manually install the drivers for your Windows NT operating system:

1. Install the required registry keys by opening the WINNT40 folder on the driver disk containing the L13WNT.REG file and double click on it. Windows will update the required registry keys.
2. Copy the BTIKRNL.DLL and L13W32.DLL files from the WINNT40 folder on the driver disk into the C:\WINNT\SYSTEM32 folder.
3. Copy the BTIUNIV.SYS file from the WINNT40 folder on the driver disk into the C:\WINNT\SYSTEM32\DRIVERS folder.
4. Restart the computer.

After completing the driver installation, the card may be tested by running the program L13TST32.EXE located on the driver disk. A complete description of testing the Lx1553-3 is described in Section 2.3.

2.3 Testing the Lx1553-3 Installation

You can test the installation and functionality of your Lx1553-3 by double-clicking the L13TST32.EXE program provided on the installation disk. This program completes execution within several seconds. If the program does not detect any faults with the interface or the Lx1553-3 hardware, it displays the Card Number, Slot Number and a "passed test" message. The Lx1553-3 is now ready to connect to the databus as described in Section 2.5.

If the L13TST32 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.

2.4 Multiple Card Installation

The installation of multiple Lx1553-3 cards into your computer requires a few more steps than a single card installation. These extra steps are needed to differentiate between the installed cards by determining their Slot and Card Numbers.

Slot Numbers are logical numbers assigned by the operating system. They are constant for a given system and identify each physical PCI (cPCI) slot. Note, the logical Slot Number is usually not equal to the physical PCI (cPCI) chassis slot. Card Numbers are used by application software to uniquely identify each

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Lx1553-3. The Card Numbers may change depending on the position and number of cards used. **If only one Lx1553-3 is used, it is Card Number 0**.

The association of the Card Number to each physical card depends on the Slot Number occupied by a particular card. Changing slots or removing cards when multiple cards are installed may change the Card Number assigned to a particular Lx1553-3 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.

Use one of the following procedures to determine the Slot Numbers of the computer and to relate these Slot Numbers to the Card Numbers. Table 2-1 may be useful when installing multiple Lx1553-3s. Make a copy and fill in the information as needed.

For multiple LP1553-3 PCI cards:

1. Install a single LP1553-3 as described in Sections 2.1 and 2.2.
2. Run the L13TST32 test utility and note the Slot Number reported by L13TST32.
3. Write this Slot Number in Table 2-1 and on the case of the computer by the PCI slot containing the card.
4. Shut down the computer and move the LP1553-3 to the next open PCI slot.
5. Restart the computer.
6. Repeat steps 2 through 5 for each slot to be used by installed LP1553-3s.
7. Once Slot Numbers have been determined, insert and secure all LP1553-3s into the computer.
8. Run the L13TST32 test utility a final time and record the Card Number associated with each Slot Number.

For multiple LC1553-3 cPCI cards:

1. Insert all LC1553-3 cards in the computer as described in Section 2.1.
2. Install the drivers as described in Section 2.2.
3. Run L13TST32.EXE with the -LED option (i.e., **A:\L13TST32 –LED**).
4. Follow the on-screen instructions. L13TST32 will sequentially blink the LEDs to identify each LC1553-3. Record the Card Number and Slot Number displayed on screen.

Physical Slot # (may be marked on chasis)	1	2	3	4	5	6	7	8
System Slot #								
Card #								
Lx1553-3 model (A, B32, C, etc.)								

Table 2-1 Multiple Lx1553-3 card installation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5 Connecting to the MIL-STD-1553 Databus

This section describes the details of connecting your Lx1553-3 to the MIL-STD-1553 databus, including:

- Selecting the coupling method
- Correctly terminating the bus

The standard Lx1553-3 is provided with BJ77 twinax connectors. A 25-pin D-subminiature connector is a non-standard option; see Appendix B for more information. Cables with PL75 plugs can mate with the BJ77 jack on the Lx1553-3. A variety of 1553 cables, couplers and accessories are available from Ballard Technology. Please call Ballard Technology with your custom cabling requirements.

2.5.1 Transformer Versus Direct Coupling

Per MIL-STD-1553 the Lx1553-3 may be either transformer coupled or direct coupled to the databus.

If your Lx1553-3 has standard twinax connectors, a coupling method is selected with the P3 jumpers shown in Fig. 2-1. If your Lx1553-3 has a non-standard D-sub connector, these jumpers have no effect.

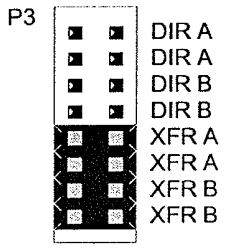


Fig. 2-1 Coupling selection jumpers

Each channel requires two jumpers. Both jumpers for a given channel must be of the same coupling type. XFR indicates transformer coupling. DIR indicates direct coupling.

Fig. 2-2 and Fig. 2-3 show the electrical schematic for the transformer coupling and the corresponding cable assembly using standard bus coupling components.

Fig. 2-4 and Fig. 2-5 show the electrical schematic and cable assembly for direct coupling.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

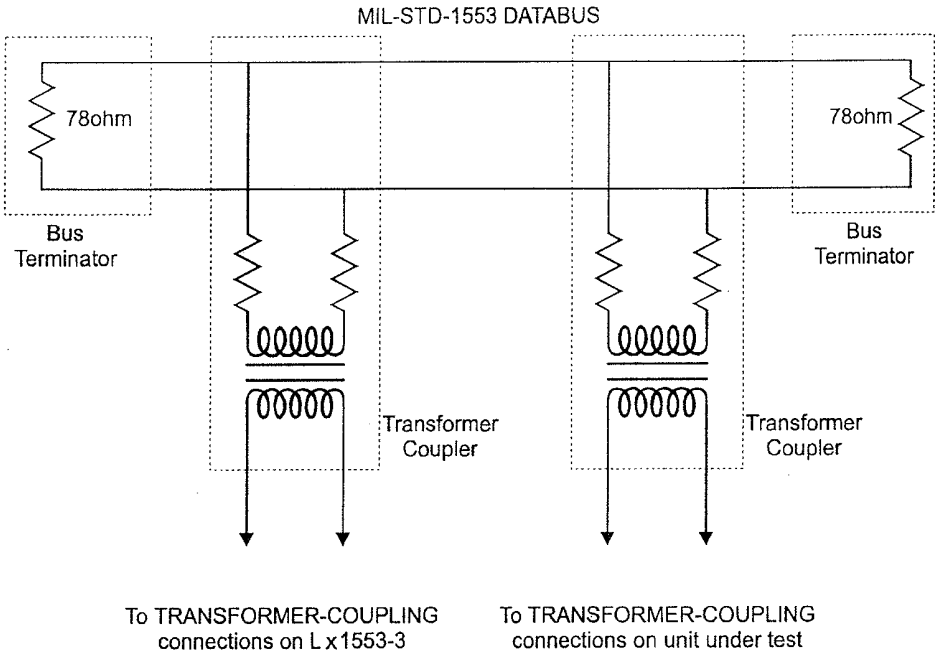


Fig. 2-2 Schematic of transformer coupling

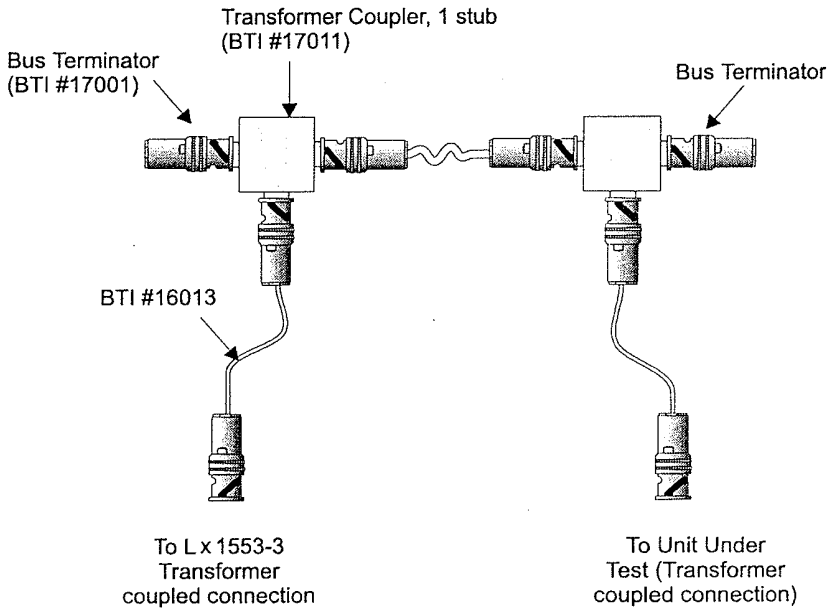


Fig. 2-3 Cable assembly for transformer coupling of Lx1553-3 to unit under test

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

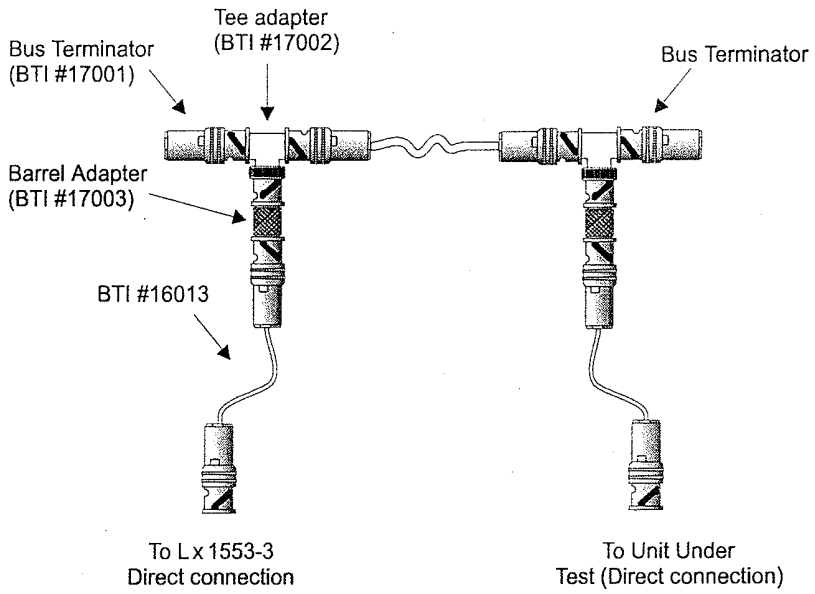


Fig. 2-4 Schematic of direct coupling

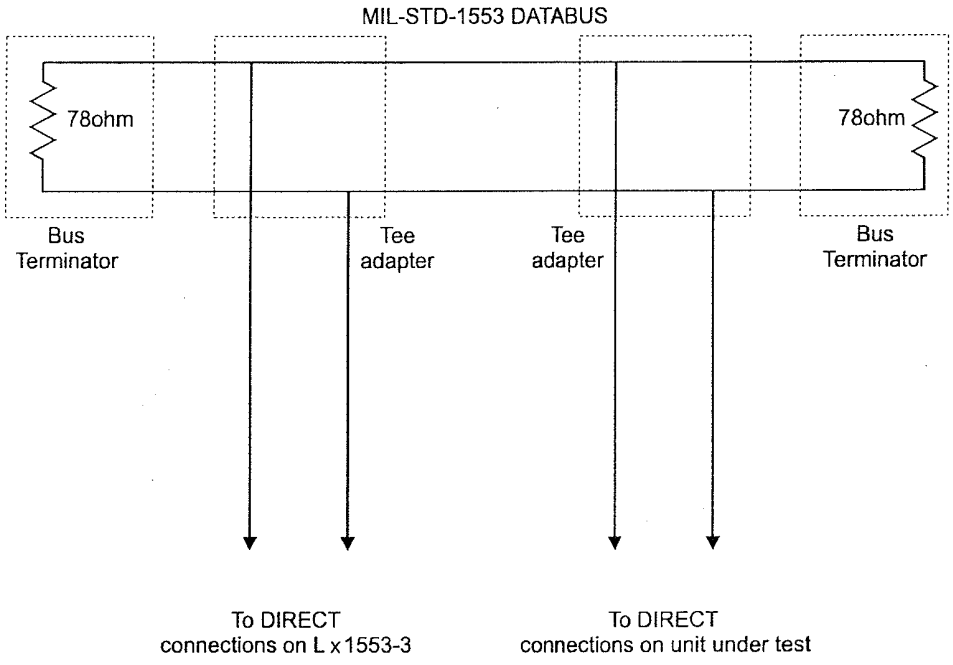


Fig. 2-5 Cable assembly for direct coupling of Lx1553-3 to unit under test

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5.2 Bus Termination

A MIL-STD-1553 bus must be terminated at both ends by 70-85 ohm resistors. These match the characteristic impedance of the cable and are essential to provide proper loading and minimize signal reflection and degradation on the bus. The most common problem on new systems is an improperly terminated databus.

The Lx1553-3 contains on-board 75 ohm terminating resistors that may be used with direct coupling in place of one of the bus terminators shown in Fig. 2-5. A second bus terminator must still be used at the other end of the bus. Having the jumpers in place, the termination resistors are selected. Fig. 2-6 shows the terminating resistor jumpers. The on-board terminating resistors should not be selected with transformer coupling. The header is labeled P1 on the LP1553-3, and P2 on the LC1553-3.

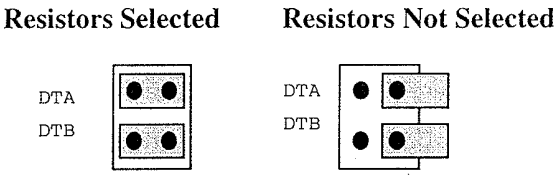


Fig. 2-6 Termination resistor selection jumpers for direct coupling

In some applications (e.g., lab-bench assemblies), the bus may be extremely short or non-existent. However, the bus must still be terminated regardless of the coupling method used.

Your Lx1553-3 should now be ready to communicate on the 1553 databus under the control of an application program such as CoPilot 1553. You can also run the example programs included on the Lx1553-3 installation disk. If you are planning to write your own programs, you should proceed to Chapter 3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page intentionally blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. DEVELOPING SOFTWARE FOR THE LX1553-3

Use of the Lx1553-3 requires software running on the host computer. The easiest way to get started is to use the Graphical User Interface (GUI) application called CoPilot 1553 (available separately from Ballard Technology). With CoPilot 1553, you can interactively operate the Lx1553-3 without having to do any software programming. Contact Ballard Technology for more information about obtaining an evaluation version of CoPilot 1553.

Specialized applications, however, may require development of custom software. An API (application program interface) is included with the Lx1553-3 to simplify software development. The API is a library of functions your program calls to configure and control the Lx1553-3. The remainder of this manual explains how to develop software for the Lx1553-3 using the API.

This chapter explains the general requirements of Lx1553-3 software, independent of the application. Also covered are the basic structure of an Lx1553-3 application program and the main functionality of the API. Chapter 4 introduces sample source code for single terminal simulation. Multi-terminal simulation is described in Chapter 5.

3.1 Skeleton Lx1553-3 Application

The API was designed to be simple to use. Therefore, most programs can be modeled on the skeleton application shown in Fig. 3-1.

```
Card = L13_CardOpen(CARDNUM);

    L13_CardReset(Card);

    //  Configure the Lx1553-3 as a BC, RT and/or monitor

    L13_??Config(..., Card)

    //  Create and initialize message structure(s) for the terminal(s)

    Msg = L13_??CreateMsg(..., Card)

    L13_CardStart(Card);

        // Mode and application dependent handling of data

        L13_MsgDataRd(..., Msg, Card)
        L13_MsgDataWr(..., Msg, Card)


    L13_CardStop(Card);

L13_CardClose (Card);
```

Fig. 3-1 Skeleton Lx1553-3 application

The ellipses (...) in Fig. 3-1 indicate parameters not shown, and the "??" prefixes should be replaced by either BC, RT or Mon.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

As Fig. 3-1 shows, using the API to control the card involves eight steps:

1. Open the Lx1553-3
2. Reset the Lx1553-3 (optional)
3. Configure terminal type(s)
4. Configure and initialize message structure(s)
5. Start the Lx1553-3
6. Handle message data according to application requirements
7. Stop the Lx1553-3 (optional)
8. Close the Lx1553-3

There can be some variation to this layout of an Lx1553-3 application. For example, no message creation (**L13_??CreateMsg**) is required to run the Lx1553-3 as a monitor. Furthermore, between **L13_CardOpen** and **L13_CardClose** the Lx1553-3 may be started, stopped, reset and/or reconfigured any number of times. However, nearly all programs use the following functions:

- **L13_CardOpen**
- **L13_CardReset**
- **L13_??Config** (where ?? is either BC, RT or Mon)
- **L13_CardStart**
- **L13_CardStop**
- **L13_CardClose**

3.2 Lx1553-3 Configuration Overview

A Lx1553-3 can be configured to simulate a MIL-STD-1553 bus controller, remote terminal and/or bus monitor. A set of internal data structures constitutes a terminal configuration. The API functions set up and fill in these data structures. The Lx1553-3 version A can be loaded with one terminal configuration. Other Lx1553-3 versions can be loaded with multiple configurations simultaneously. The **L13_??Config** functions initialize terminal configurations. On multi-terminal simulation versions, **L13_CardReset** should precede **L13_??Config** to clear the existing terminal configurations.

As described in the following two sections, the primary goal of configuring the Lx1553-3 for operation is creating message structures in which to store message parameters and data. Both BC and RT configurations require message structures. Message structures are not shared between terminal configurations. Message structures may contain:

- data words
- command words
- status words
- time-tags
- error flags
- control flags

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The content of a message structure is accessed and manipulated using the various **L13_Msg??** functions.

3.2.1 BC Configuration

As shown in Fig. 3-2, a BC terminal configuration consists of two main components: the schedule and a set of message structures.

Message structures for a BC configuration must be created with the **L13_BCCreateMsg** function before they can be scheduled. These message structures contain space for command, status and data words, as well as time tags and other information.

The schedule is a program composed of an array of Command Blocks executed by the on-board DSP. Each Command Block contains an opcode. The opcodes control timing and transmission of messages. After the BC configuration is initialized with **L13_BCConfig**, a schedule can be constructed with the scheduling functions (all of which are prefixed by **L13_BCSched**). Each of the scheduling functions appends one or more Command Blocks to the current end of the schedule.

Messages can also be transmitted singly (without a schedule) using the **L13_BCTransmitMsg** function. Messages transmitted in this way do not need to be created with **L13_BCCreateMsg**.

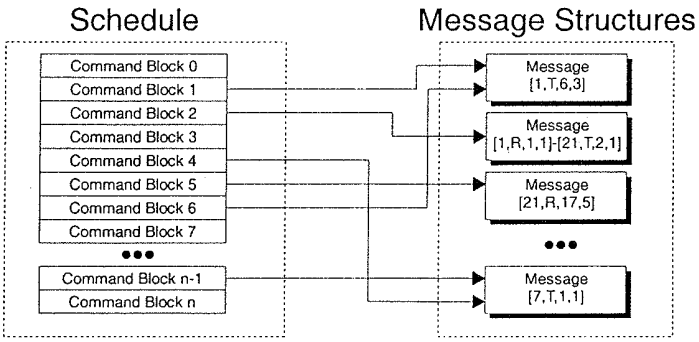


Fig. 3-2 Bus controller terminal configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2.2 RT Configuration

As shown in Fig. 3-3, a remote terminal configuration consists of a collection of message structures. Each message structure in an RT configuration is associated with a particular T/R bit and subaddress/mode-code combination. The **`L13_RTConfig`** function can automatically create message structures for every T/R bit and subaddress/mode-code combination. Alternatively, RT message structures may be created individually with **`L13_RTCreateMsg`**. Command words for which no message structures have been created for a simulated RT are considered illegalized for that RT.

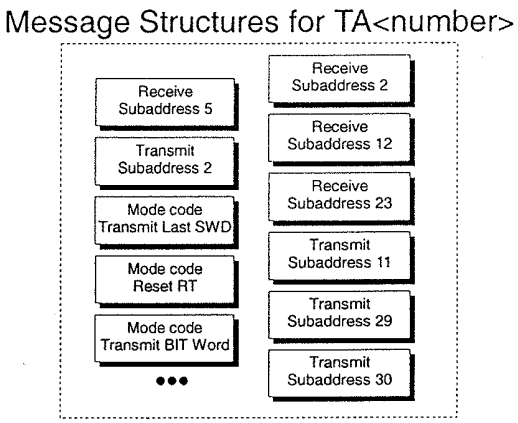


Fig. 3-3 Remote terminal configuration

3.2.3 Monitor Configuration

Configuring the Lx1553-3 to be a monitor simply means reserving memory for the monitor buffer and setting filtering options. The Lx1553-3 will fill this memory at run time with variable-length message records subject to constraints specified with the **`L13_MonConfig`** and **`L13_MonFilter`**?? functions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 Lx1553-3 Runtime Overview

The Lx1553-3 does not interact in any way with the MIL-STD-1553 databus until **`L13_CardStart`** activates the card. **`L13_CardStart`** activates all terminal configurations simultaneously. Once activated, the Lx1553-3 responds to the databus independently of the host computer until **`L13_CardStop`** halts it. A simulated BC processes its schedule (if present); simulated RTs respond automatically to commands from the BC.

As seen in Fig. 3-4, the Lx1553-3 transmits data from and stores received data in its random access memory (RAM). The various driver functions that transfer message data access message structures in this RAM. None of these functions directly affects the MIL-STD-1553 databus. For example, updating data to be transmitted with the **`L13_MsgDataWr`** function updates a specific message structure in Lx1553-3 memory. The BC schedule or bus activity, not the **`L13_MsgDataWr`** function, determines when this data is actually transmitted on the databus.

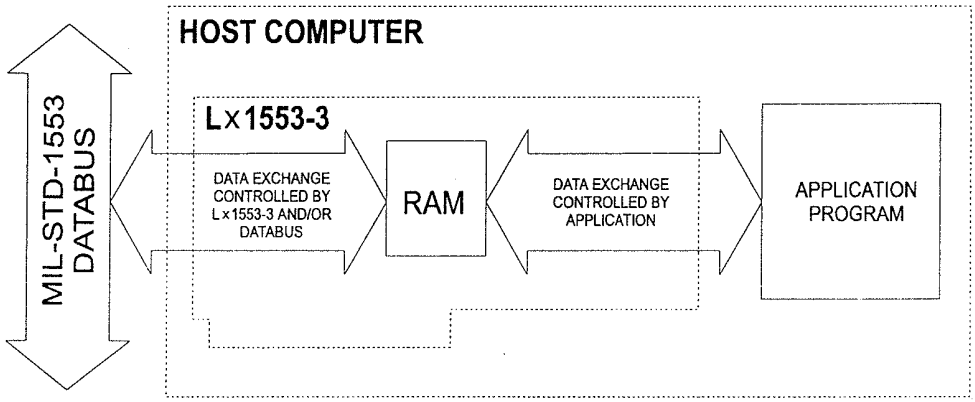


Fig. 3-4 Control of data transfer while Lx1553-3 is active

An application program can coordinate its accesses to the Lx1553-3 with the operation of the Lx1553-3 by either:

- polling status information on the Lx1553-3 or
- responding to interrupts from the Lx1553-3

This may be useful for timing data exchanges between the application program and the Lx1553-3 or controlling error injection in messages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page intentionally blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. BASIC SINGLE TERMINAL SIMULATION

Using sample programs, this chapter illustrates basic single-terminal simulation that applies to all Lx1553-3 versions. This chapter demonstrates how to:

- configure the Lx1553-3 as a BC, an RT or a monitor
- initialize and access data in each mode

After finishing this chapter, you will be familiar with the essentials of an Lx1553-3 application program, and you will be able to recognize the important functions. Chapters 5 and 6 cover multi-terminal simulation and error injection, respectively, and only apply to Lx1553-3 B and C versions.

Examples are presented as sequences of C function calls, not complete programs. To simplify and focus the presentation on the Lx1553-3 API the examples omit standard program elements (such as data declarations) except where they are crucial to understanding the example. Complete source code and executable versions of all sample code are provided on the installation disk with the API library. These samples can serve as starting points for custom applications. Some of these samples may be useful as they are.

4.1 Simulating the BC: Unscheduled Messages

The simplest BC application transmits a single 1553 message, waits for its completion and then processes and/or displays the status word and/or data response. Fig. 4-1 shows all the function calls necessary to accomplish this with the Lx1553-3.

```
XMITFIELDS Msg;

Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);
L13_BCConfig(BCCFG_DEFAULT, Card);

Msg.ctrlflags = MSGCRT_BCRT;
Msg.cwd1      = 0x0843;
Msg.data[0]   = some_value0;
Msg.data[1]   = some_value1;
Msg.data[2]   = some_value2;

L13_BCTransmitMsg(&Msg, Card);

if (Msg.errflags & (MSGERR_NORESP | MSGERR_ANYERR))
{
        //  The RT did not respond or had an error in its response.
}
else
{
        //  The message was completed successfully.
        //  Check Msg.status1 for the RT's status word response.
}
L13_CardClose(Card)
```

Fig. 4-1 Example 1: Unscheduled message transmission

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

L13_CardOpen is always the first function called since it obtains the card handle needed by subsequent functions. **L13_CardReset** clears Lx1553-3 memory, removing any existing configuration data. **L13_BCConfig** configures the Lx1553-3 to be a bus controller.

L13_BCTransmitMsg commands the Lx1553-3 BC to transmit the message defined in the *Msg* structure. The *Msg* structure (declared in the header file) contains many fields in which to pass message parameters to **L13_BCTransmitMsg. L13_BCTransmitMsg** also passes information back to your program in this structure. The code initializes only the fields relevant to this example before passing the structure to **L13_BCTransmitMsg**. Here the message type, command word and three data words are initialized.

If the transmission was successful, the function returns the status word in the *Msg.swd1* field. If the message transmitted was an RT-to-BC or RT-to-RT transfer, **L13_BCTransmitMsg** would return data values in the *Msg.data[]* array. In contrast to the skeleton example in the previous chapter, no **L13_CardStart** or **L13_CardStop** calls are required. Both are implicit in **L13_BCTransmitMsg**.

To transmit messages at given intervals, **L13_BCTransmitMsg** could be placed in a timed loop and passed different XMITFIELDS structures. However, this is unnecessary work for the host computer since the Lx1553-3 can time message transmission internally as demonstrated in the next section.

4.2 Simulating the BC: Scheduled Messages

Messages on a MIL-STD-1553 databus are typically transmitted according to periodic schedules. These schedules are usually composed of timed groups of messages called frames. Frames are sequenced so that each message is transmitted at the desired frequency.

Suppose, for example, that we would like to transmit three messages at the following rates:

msg0 every 10ms

msg1 every 20ms

msg2 every 40ms

Fig. 4-2 shows a schedule to achieve these rates. Notice in Fig. 4-2 that by repeating a message in different frames and possibly repeating frames in the cycle we are able to meet the timing requirements for each message.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

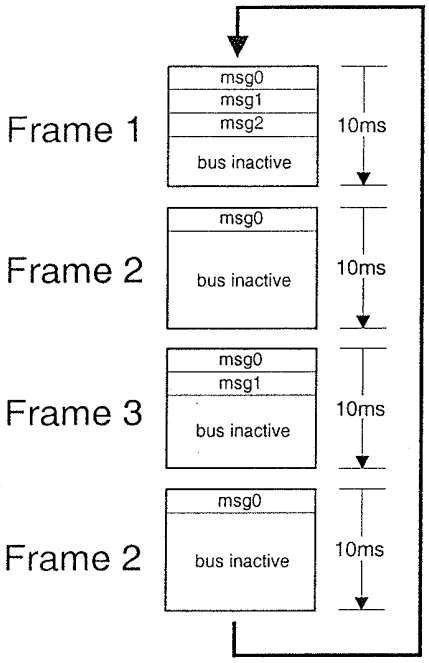


Fig. 4-2 Typical BC message schedule

The Lx1553-3 can execute such a schedule autonomously, maintaining all associated timing on-board. The code in Fig. 4-3 demonstrates how to configure the Lx1553-3 with the schedule shown above.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
MSGADDR BCMsgs[3];

Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

L13_BCConfig(BCCFG_DEFAULT,Card);

BCMsgs[0] = L13_BCCreateMsg(MSGCRT_DEFAULT, 0x843, 0, &InitalData, Card);
BCMsgs[1] = L13_BCCreateMsg(MSGCRT_DEFAULT, CMD1, 0, NULL, Card);
BCMsgs[2] = L13_BCCreateMsg(MSGCRT_RTRT, RCV_CMD2, XMT_CMD2, NULL, Card);

L13_BCSchedFrame(1000, Card);
L13_BCSchedMsg(BCMsgs[0], Card);
L13_BCSchedMsg(BCMsgs[1], Card);
L13_BCSchedMsg(BCMsgs[2], Card);

L13_BCSchedFrame(1000, Card);
L13_BCSchedMsg(BCMsgs[0], Card);

L13_BCSchedFrame(1000, Card);
L13_BCSchedMsg(BCMsgs[0], Card);
L13_BCSchedMsg(BCMsgs[1], Card);

L13_BCSchedFrame(1000, Card);
L13_BCSchedMsg(BCMsgs[0], Card);

L13_CardStart(Card);

while ( /* AS REQUIRED BY APPLICATION */ )
{
    error_field = L13_MsgFieldRd(FIELD_ERROR, BCMsgs[0], Card);

    if (error_field & MSGERR_HIT)
    {
        // The current data values have been transmitted, so update them.

        L13_MsgDataWr(&NewData, 3, BCMsgs[0], Card);

        //Clear the "hit" bit

        L13_MsgFieldWr(0, FIELD_ERROR, BCMsgs[0], Card);
    }
}

L13_CardStop(Card);

L13_CardClose(Card);
```

Fig. 4-3 Example 2: Scheduled message transmission

After obtaining the card's handle with **L13_CardOpen** and resetting the Lx1553-3 with **L13_CardReset, L13_BCConfig** configures it as a bus controller. If **L13_CardReset** is omitted, any previous terminal configuration remains intact and may corrupt the current configuration when the card is started. Flags can replace BCCFG_DEFAULT to enable or disable options such as the interrupt log list or mode codes. See the Function Reference (Appendix A).

L13_BCCreateMsg creates a message structure and initializes it with the command words and, for BC-to-RT messages, the data to be transmitted. This function returns the address of the message structure. Other functions such as **L13_MsgDataWr** require this address to access the message. The *BCMsgs* array holds the returned message addresses.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The first **L13_BCCreateMsg** function creates a BC-to-RT message. The 0x0843 parameter is the 1553 command word which commands terminal address 1 to receive 3 data words into its receive subaddress 2. Notice that the third **L13_BCCreateMsg** function creates an RT-to-RT message. The MSGCRT_DEFAULT parameter is replaced by MSGCRT_RTRT, and a second command word is provided.

After the messages are created, they can be scheduled. Notice how the blocks of **L13_BCSched**?? functions exactly correspond with the frames illustrated in Fig. 4-2. **L13_BCSchedFrame** marks the start of a frame and includes the frame time (in microseconds) as one of its parameters. Frames can be up to 1.3 seconds in length. **L13_BCSchedMsg** inserts the specified message into the schedule. The schedule contains references to the messages structures, not copies of structures. This means message data exists in only one place, so updating message data updates all instances of it in the schedule.

The scheduling functions (all of which are prefixed by **L13_BCSched**) actually create a program in Lx1553-3 memory. After **L13_CardStart**, the Lx1553-3 functions as a BC by executing this program. The program is cyclic by default. That is, the sequence of frames described in the code are transmitted repeatedly according to the schedule.

Your program can read or update message data at any time with the **L13_MsgDataRd** or **L13_MsgDataWr** functions. The code in this example only deals with the BC-to-RT message. The **L13_MsgField**?? functions access message parameters other than data. They are used in this example to check and reset a bit in the errors field of the message structure that indicates that the message has been transmitted.

4.3 Simulating an RT

The code in Fig. 4-4 complements the preceding example. It configures the Lx1553-3 as a remote terminal with terminal address 1 to receive the BC-to-RT message transmitted by the BC in the preceding example. It polls receive subaddress 2 where it expects to be receiving 3 data words.

```
Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

L13_RTConfig(RTCFG_DEFAULT, 1, Card);

RTMsg = L13_RTGetMsg(SUBADDRESS, 1, RCV, 2, Card);

L13_CardStart(Card);

while( /* AS REQUIRED BY THE APPLICATION */ )
{
    //  Read the message from the BC

    L13_MsgDataRd(&Data, 3, RTMsg, Card);
}
L13_CardStop(Card);
L13_CardClose(Card);
```

Fig. 4-4 Example 3: RT simulation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The most significant configuration parameter, its terminal address, is assigned in the **L13_RTConfig** function. The RTCFG_DEFAULT parameter tells the Lx1553-3 to set up message structures for all subaddresses and mode codes. Since message access requires a message address, you must follow **L13_RTConfig** with a call to **L13_RTGetMsg** for each message of interest. In this example, **L13_RTGetMsg** gets the message address associated with the 0843h command.

Since this example only anticipates receiving data from the BC, the code does not initialize any of the Lx1553-3 RT transmit message structures. You could use **L13_MsgDataWr** after **L13_RTGetMsg** to initialize data for transmission. Within the *while* loop **L13_MsgDataRd** reads received message data using the address returned by **L13_RTGetMsg**.

Since **L13_RTConfig** was called with RTCFG_DEFAULT, all subaddresses and mode codes are legal. After **L13_CardStart** activates the Lx1553-3, it responds to any valid command to terminal address 1, although this example only anticipates command word 0843h.

Alternatively, you could call **L13_RTConfig** with RTCFG_NOBUILD in place of RTCFG_DEFAULT and then call **L13_RTCreateMsg** for each message of interest. In this case, only subaddresses and mode codes specified in a **L13_RTCreateMsg** function are mapped into memory; others are illegalized. This conserves Lx1553-3 memory for multi-terminal simulation.

4.4 Monitoring the Bus

In monitor mode the Lx1553-3 records messages sequentially in its memory in the order they are received from the bus. Each message record contains:

- command word(s)
- data word(s)
- status word(s)
- time-tag
- error flags

The code in Fig. 4-5 demonstrates the use of the Lx1553-3 as a bus monitor to capture a "snapshot" of bus traffic.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

L13_MonConfig(MONCFG_DEFAULT, Card);

L13_CardStart(Card);

while(L13_MonIsRunning(Card))
{
    //  Wait for buffer to fill
}

while(words_copied = L13_MonRd(&MsgBuffer, Card))
{
    //  Store, process or display the message in MsgBuffer.
}

L13_CardStop(Card);
L13_CardClose(Card);
```

Fig. 4-5 Example 4: Bus monitoring

After opening and resetting the card, the **L13_MonConfig** function sets up monitor mode. A 16K-word buffer is allocated by default. See **L13_MonConfig** in Appendix A for other buffer size options.

By default, monitoring halts when the buffer is full. The first *while* loop polls the Lx1553-3, waiting for its buffer to fill. After it is full, the following *while* loop empties the Lx1553-3 buffer into the program one message at a time. **L13_MonRd** fills the buffer pointed to by *MsgBuffer* with the oldest unread message, increments an internal pointer and returns a value indicating the number of words it copied. **L13_MonRd** returns 0 when the buffer is empty. See **L13_MonRd** in Appendix A for a description of the structure of *MsgBuffer*.

It is not necessary to wait for the buffer to fill before reading. The loops in this example could be nested, so that **L13_MonRd** empties the buffer as it fills. Furthermore, you can use the MONCFG_CONTINUOUS flag in **L13_MonConfig** to make the buffer circular. In this mode the internal buffer pointers are automatically reset when the buffer fills and the Lx1553-3 starts overwriting old data without pause.

Lx1553-3 versions B and above can filter the monitor record. Filtering is defined with the **L13_MonFilterTA** and **L13_MonFilterSA** functions.

4.5 Summary

The preceding examples illustrated the most important API functions. Your programs for the Lx1553-3 can be modeled on the code in the preceding examples. Detailed function descriptions are found in Appendix A, the Function Reference. Check the README.TXT files on the installation disk for updated function information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page intentionally blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. MULTI-TERMINAL SIMULATION

The Lx1553-3 B and C versions can simultaneously simulate a bus controller and/or up to 32 remote terminals. The C version adds concurrent monitoring capability and error injection. This chapter demonstrates how to configure the Lx1553-3 to simulate multiple terminals.

5.1 Simultaneous BC, RT and Monitor Simulation

To simulate more than one terminal simultaneously with the Lx1553-3 you simply concatenate the configuration code for each terminal. The code in Fig. 5-1 illustrates this, combining terminal configurations similar to Examples 2, 3 and 4 from Chapter 4. This example assumes a Lx1553-3 version C. If a B version is used the monitor function is not available simultaneously with RT and BC simulation.

```
MSGADDR BCMsg, RTMsg;

Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

//  Bus controller configuration from Example 2 (Fig. 4-3)

L13_BCConfig(BCCFG_DEFAULT, Card);

BCMsg = L13_BCCreateMsg(MSGCRT_DEFAULT, 0x0843, 0, &InitialData, Card);

L13_BCSchedFrame(10000, Card);
L13_BCSchedMsg(BCMsg, Card);

//  RT configuration from Example 3 (Fig. 4-4)

L13_RTConfig(RTCFG_DEFAULT, 1, Card);
RTMsg = L13_RTGetMsg(SUBADDRESS, 1, RCV, 2, Card);

//  Monitor configuration from Example 4 (Fig. 4-5)

L13_MonConfig(MONCFG_DEFAULT, Card);
L13_MonFilterTA(0x00000002, Card);

L13_CardStart(Card);

while( /* AS REQUIRED BY THE APPLICATION */ )
{
    //  Handle data as in Examples 2 through 4 in Chapter 4.
}

L13_CardStop(Card);
L13_CardClose(Card);
```

Fig. 5-1 Example 5: Multi-terminal simulation

L13_CardOpen and **L13_CardReset** only need to be called once. Note that **L13_CardReset** is especially important in programs for multi-terminal versions of the Lx1553-3. If **L13_CardReset** is omitted, terminal configurations from a previous program may be intact on the Lx1553-3. This could result in simulation of unintended terminals.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This example, like the single-terminal examples, follows the program model discussed in Section 3.1. In this example, the message configuration functions for a given terminal immediately follow the corresponding **L13_??Config** function. This grouping is arbitrary. **L13_??Config** functions could be grouped and all other configuration follow. The order in which the **L13_??Config** functions are called is also arbitrary.

The **L13_BCSchedFrame** and **L13_BCSchedMsg** functions create a schedule containing only one frame. This frame contains only the 0843h command, so this one message is transmitted every 10 milliseconds.

There is one additional function not found in the previous examples: **L13_MonFilterTA**. This function is available for Lx1553-3 versions B and above. It takes a 32-bit bitmask parameter that specifies which terminal addresses to monitor. Messages to or from other terminals are omitted from the monitor record. The 00000002h parameter in the example specifies that only remote terminal address 1 will be monitored.

As described in Section 3.3, the **L13_CardStart** function activates all configured terminals. Message data and parameters would be accessed as in previous examples with the **L13_MsgData**?? and **L13_MsgField**?? functions.

5.2 Broadcast Data and RT Address 31

An RT configuration can be configured to ignore or accept broadcast message data. If it is configured to accept broadcast data, the Lx1553-3 always follows Notice II to MIL-STD-1553 regarding the storage of this data. That is, broadcast data are always stored separately from non-broadcast data.

If any RT address is configured to accept broadcast data, the Lx1553-3 automatically creates a shadow terminal address 31, in which all broadcast data is stored. Access to this data is obtained by calling **L13_RTGetMsg** with terminal address 31.

You may also enable RT address 31 as a genuine RT by calling **L13_RTConfig**(..., 31, ...). This completely disables use of broadcast messages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6. ERROR INJECTION

This chapter applies only to the Lx1553-3 version C, which can generate specific errors in its transmissions. This capability is useful for testing the response of hardware or systems to bus faults. This chapter explains how to use the Lx1553-3 API to control generation of errors. Three examples demonstrate how to specify the type, location and frequency of an error.

6.1 Error Injection Overview

Error injection is the act of transmitting a signal that does not comply with the MIL-STD-1553 specification. Any of the following types of errors may be injected:

1. Parity: inverts the parity bit
2. Sync error: inverts the sync pulse
3. Bit Count: transmits a word with fewer or greater than the normal 20 bit times.
4. Manchester error: inverts either the first or second half bit in a given bit position
5. Word count: transmits a specified number of data words, ignoring the word count field of the command word
6. Gap errors: insert a gap of specified length preceding a word

Error injection is a three-step process. An application program:

1. Defines an error in the Lx1553-3.
2. Tags the message or messages that are candidates to contain the error. This step is optional.
3. Enables error injection.

The following subsections describe these steps in detail.

6.1.1 Defining the Error

Errors are defined using the **`L13_ErrorDefine`** function. Only one error may be defined on the Lx1553-3 at any given time, and that error is used in all subsequent error injections. The error may be changed by repeated calls to **`L13_ErrorDefine`**. Table 6-1 shows the parameters required for complete definition of different types of errors.

Error Type	Word position	Value	Count	Bit position
Parity	✔			
Sync	✔			
Manchester	✔			✔
Word count			✔	
Bit count	✔	✔	✔	
Gap	✔	✔		

Table 6-1 L13_ErrorDefine parameters according to error type

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.1.2 Tagging Messages

Error injection is most useful when the error is associated with a specific message(s). For example, when simulating the BC, you may want to inject errors only in messages to a specific RT. In this case, you would tag those messages in the bus controller schedule for that RT. Then when error injection is enabled, only those messages are candidates to contain errors.

The Lx1553-3 can inject errors into the transmissions of both simulated RTs and a simulated BC. Since different parts of a MIL-STD-1553 message are transmitted by different terminals, you must tag the message in the terminal that transmits the part of the message with the error. For example, if the error definition specifies an error in the status word of a message, the message must be tagged in the simulated RT (see Section 6.4).

Messages can be tagged in two ways. They can be tagged when they are created by using the MSGCRT_ERR flag in the **`L13_??CreateMsg`** function. Also, they can be tagged and untagged thereafter by calling one of the two **`L13_ErrorTag??`** functions.

Note that tagging a message does not guarantee that it will contain an error in its next transmission. A tagged message is only a candidate. Two factors could prevent a tagged message from containing an error.

1. The error definition does not make sense in the given message. For example, the error definition specifies the sixth data word in a message, but the message tagged only transmits three data words.
2. Several messages have been tagged, but the Lx1553-3 is configured to only transmit one error after being enabled (see Section 6.1.3 below).

6.1.3 Enabling Error Injection

After an error has been defined and messages have been tagged, error injection can be enabled with the **`L13_ErrorCtrl`** function. This function arms the Lx1553-3 error-injection mechanism. Depending on the options selected by its flag parameter, one of four things happens following a call to **`L13_ErrorCtrl`**:

1. OFF: Error injection is completely disabled.
2. ON: Every applicable message transmission contains the error until the condition is changed by another **`L13_ErrorCtrl`** call.
3. ONCE: The next applicable message transmission contains an error. After one instance of the error is transmitted, error injection is automatically disabled.
4. EXTERNAL: Error injection is under control of an external hardware signal located on header P4, *EXTTRIG (see Appendix C).

By default, only tagged messages are candidates to contain errors, however, this may be overridden by **`L13_ErrorCtrl`**. If the ERRCTRL_ANYMSG flag is used in this function, any transmitted message can contain the error. Tagging is ignored.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

As the following examples show, the above steps can be performed at any time, including while the Lx1553-3 is running. For example, you could inject intermittent, random errors in bus controller transmissions without stopping the BC schedule.

The following examples demonstrate how to define errors, tag messages and enable error injection. Note that the examples in this chapter are similar to those in earlier chapters, except that they ignore data to focus on the process of error injection.

6.2 Errors in BC Transmissions: Unscheduled Messages

The following code modifies Example 1 in Section 4.1 so that the message transmitted has a parity error in its command word.

```
XMITFIELDS Msg;

Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

L13_BCConfig(BCCFG_DEFAULT, Card);

Msg.ctrlflags  = MSGCRT_BCRT;
Msg.cwd1       = 0x0843;
Msg.data[0]    = some_value0;
Msg.data[1]    = some_value1;
Msg.data[2]    = some_value2;

L13_ErrorDefine(ERRDEF_PAR, 0, 0, ERRDEF_CWD1, 0, Card);
L13_ErrorCtrl(ERRCTRL_ONCE | ERRCTRL_ANYMSG, Card);

L13_BCTransmitMsg(&Msg, Card);

if(Msg.errflags & (MSGERR_NORESP | MSGERR_ANYERR))
{
    //  The RT did not respond or had an error in its response.
}
else
{
    //  The message was completed successfully.
    //  Check Msg.status1 for the RT's status word response.
}

L13_CardClose(Card);
```

Fig. 6-1 Example 6: Errors in unscheduled BC transmissions

Only the **L13_ErrorDefine** and **L13_ErrorCtrl** functions are added to Example 1 (Fig. 4-1). **L13_ErrorDefine** specifies that the first command word in a message should contain a parity error. **L13_ErrorCtrl** arms the Lx1553-3 error-injection mechanism. This example can safely use the ERRCTRL_ANYMSG flag since only one message is to be transmitted.

When **L13_BCTransmitMsg** is executed, the transmitted message contains incorrect parity in its command word as specified by the parameters of **L13_ErrorDefine**. According to MIL-STD-1553, an RT should never respond to invalid commands. Thus, in this example, no response is expected from the RT, and the first block of the *if-else* statement should be executed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3 Errors in BC Transmissions: Scheduled Messages

The example in Fig. 6-2 demonstrates how to:

- inject a single type of error into a bus controller's schedule,
- control which message contains an error

The code creates a schedule with two messages, defines a bit count error and alternates which message contains the error while the schedule is running.

```
MSGADDR     BCMsgs[2];
int         ErrorCount = 0;

Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

L13_BCConfig(BCCFG_DEFAULT, Card);

//  Create the messages

BCMsgs[0] = L13_BCCreateMsg(MSGCRT_DEFAULT, 0x0841, 0, &InitialData, Card);
BCMsgs[1] = L13_BCCreateMsg(MSGCRT_DEFAULT, 0x1041, 0, &InitialData, Card);

//  Create the schedule

L13_BCSchedFrame(10000, Card);
L13_BCSchedMsg(BCMsgs[0], Card);
L13_BCSchedMsg(BCMsgs[1], Card);

//  Initialize error injection controls

L13_ErrorDefine(ERRDEF_CNTBIT, 3, 2, 1, 0, Card);
L13_ErrorTagBC(TRUE, BCMsgs[0], Card);
L13_ErrorCtrl(ERRCTRL_ONCE | ERRCTRL_TAGMSG, Card);

L13_CardStart(Card);

while( /* AS REQUIRED BY THE APPLICATION */ )
{
    if( L13_ErrorSent(Card) && /* ANOTHER ERROR DESIRED */ )
    {
        if(++ErrorCount & 1)
        {
            L13_ErrorTagBC(FALSE, BCMsgs[0], Card);
            L13_ErrorTagBC(TRUE, BCMsgs[1], Card);
        }
        else
        {
            L13_ErrorTagBC(TRUE, BCMsg[0], Card);
            L13_ErrorTagBC(FALSE, BCMsg[1], Card);
        }

        //  Reenable injection of one error

        L13_ErrorCtrl(ERRCTRL_ONCE | ERRCTRL_TAGMSG, Card);
    }
}

L13_CardStop(Card);
L13_CardClose(Card);
```

Fig. 6-2 Example 7: Errors in scheduled BC transmissions

This example first creates a simple message schedule consisting of a single 10ms frame containing two messages. Then **L13_ErrorDefine** specifies a bit-count error in which the first data word contains 2 extra bits, and the value of

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

those bits is 3 ("11" binary). **L13_ErrorTagBC** marks the 0841h message as a candidate to receive the error. Since **L13_ErrorCtrl** is called before **L13_CardStart**, the very first transmission of 0841h will contain the error. Since one occurrence was specified (by ERRCTRL_ONCE), subsequent transmissions will be error free until the body of the *if* statement is executed.

The "if(**L13_ErrorSent**...)" conditional guarantees that the currently tagged message has been transmitted (at least once) with an error before the code reassigns the error to the other message. The code counts the number of errors that have been transmitted. The **L13_ErrorTagBC** functions untag one of the two messages and tag the other according to the parity of *ErrorCount*. Error injection is then reenabled for a single occurrence by **L13_ErrorCtrl**.

6.4 Errors in RT Transmissions

The following example configures the Lx1553-3 to simulate a single RT and inject a variety of errors.

```
Card = L13_CardOpen(CARDNUM);

L13_CardReset(Card);

L13_RTConfig(RTCFG_DEFAULT, 1, Card);

L13_ErrorDefine(   ERRDEF_GAP,     // a gap of...
                   200,            // 20 microseconds
                   0,              // don't care
                   ERRDEF_SWD1,    // ...preceding the status word
                   0,              // don't care
                   Card);

L13_ErrorTagRT(TRUE, SUBADDRESS, 1, XMT, 2, Card);
L13_ErrorCtrl(ERRCTRL_ONCE | ERRCTRL_TAGMSG, Card);

L13_CardStart(Card);

while( /* AS REQUIRED BY THE APPLICATION */ )
{
    if (L13_ErrorSent(Card) && /* ANOTHER ERROR DESIRED */ )
    {
        L13_ErrorDefine(   new_error_type,
                           new_value,
                           new_count,
                           new_word_position,
                           new_bit_position,
                           Card);

        L13_ErrorCtrl(ERRCTRL_ONCE | ERRCTRL_TAGMSG, Card);
    }
}

L13_CardStop(Card);
L13_CardClose(Card);
```

Fig. 6-3 Example 8: Errors in RT transmissions

After opening the card and configuring an RT with terminal address 1, **L13_ErrorDefine** defines a gap error of 20µs to precede the status word in the RT's response. (The value parameter of **L13_ErrorDefine** for gap errors is in units of 0.1µs.) **L13_ErrorTagRT** specifies that the error can occur when

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the RT is commanded to transmit from its second subaddress. **L13_ErrorCtrl** enables one occurrence of the defined error.

As in the preceding example, **L13_ErrorSent** is used to guarantee that the currently defined error is sent at least once before the code changes it. Inside the *if* statement **L13_ErrorDefine** changes the error definition. This function could merely change the length of the inserted gap error, or it could change the type of error altogether.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7. ADVANCED FEATURES

The Lx1553-3 has two advanced features, dual-port RAM and list buffering, that are useful in certain applications. These features allow users to optimize their system for speed and create custom message structures.

7.1 True Dual Port Ram

The Lx1553-3 uses 64KB (32K x 16) of true dual-port RAM (DPRAM). This DPRAM is in addition to the standard RAM, which has dual-port capability but requires more time to resolve contention. The DPRAM allows simultaneous access by both ports: the internal bus of the Lx1553-3 and the host computer. By minimizing the time used in the contention circuit to increase data throughput, the DPRAM is useful for time-critical applications. To further enhance the speed of the Lx1553-3, a small portion of DPRAM is reserved for internal variables and the communication process. The remainder is available to the user.

In some applications, the dual-port RAM is automatically used, so specifying the DPRAM may not show any improvement over the default setting. For example, the Lx1553-3 uses the DPRAM by default when configured as a bus controller or remote terminal. When the DPRAM is full, any additional memory requirements are allocated from the standard RAM cache.

The **`L13_BCCreateMsg`**, **`L13_RTCreateMsg`** and **`L13_MonConfig`** functions contain control flags that allow the user to specify whether the DPRAM is used, and how. Information on these functions is found in the Function Reference of Appendix A.

7.2 List Buffering

Normally, the Lx1553-3 uses a single buffer per message. Alternatively, list buffering allocates multiple memory locations for each message and increases the versatility of the Lx1553-3. These multiple memory locations operate in a variety of ways, depending on the list buffer mode.

List buffers are created using functions that specify the mode and size of the list. These functions are similar to the functions used for creating single messages and automatically keep track of the message locations in the list.

7.2.1 List Buffer Modes

The three different modes of list buffering outlined below control the behavior and configuration of the memory blocks. The circular mode can be used only for transmission, while the other two list buffering modes can be used for either transmission or reception. In all modes, the indexes are automatically maintained by the functions that do the list buffer reads and writes. As with single-buffered messages, it is a good idea to initialize the values before transmission.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Circular List Buffer

The circular list buffer is used for repeatedly transmitting patterns of data. The circular list only operates as a transmit buffer. When this mode is specified, each time a message is transmitted onto the databus, the Lx1553-3 automatically advances the index to the next list entry. When the end of the list is reached, the index returns to the beginning of the list and continues. The circular list buffer could be used to repeatedly transmit a pattern of data, like a ramp or sine wave.

FIFO List Buffer

The "first-in first-out" (FIFO) list buffer is used for transmitting or receiving a sequence of data, as in a file transfer. This type of list buffer operates like a queue on a first-in first-out basis. Operation of the FIFO is the same for both BC and RT configurations, but differs between transmission and reception.

When the transmitting FIFO buffer is full and the host tries to write more data to it, the list write function returns an error. If the transmitting FIFO is empty, the last value is used for transmissions until the host writes a new value into the FIFO.

When receiving, if the FIFO becomes full, newly received messages are written to the FIFO and the oldest messages are lost. If the receiving FIFO becomes empty and the host tries to read the FIFO, the list read function returns a zero to indicate that there was no data to be read. This process may be used for polling the status of the FIFO buffer.

Ping-Pong List Buffer

The ping-pong list buffer guaranties data integrity by preventing a problem that can occur when only a single buffer is used. If the host computer and the Lx1553-3 processor simultaneously access the same message in memory, the data being read by the host or transmitted on the bus could contain part of the old message and part of the new message. The ping-pong list buffer uses multiple memory locations, so the host always reads the most recent, complete copy of a received message. In addition, the most recent, complete copy of a message is automatically used for transmissions.

7.2.2 List Buffer Operation

When a list buffer is needed in place of a single buffer, the **`L13_BCCreateList`** or **`L13_RTCreateList`** functions are used in place of the **`L13_BCCreateMsg`** or **`L13_RTCreateMsg`** functions. The ??**`CreateList`** functions have two additional parameters, which specify the mode of the list buffer and the number of records. For a ping-pong list buffer, the number of records parameter is ignored. List buffer messages are transferred to and from the host computer using **`L13_ListDataWr`** and **`L13_ListDataRd`** functions, much like the single-buffer accesses using **`L13_MsgDataWr`** and **`L13_MsgDataRd`**.

Examples of the Lx1553-3 list buffer modes are found on the Lx1553-3 installation disk, as explained in the README.TXT file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX A: FUNCTION REFERENCE

This appendix provides detailed information on the primary MIL-STD-1553 driver functions of the LP1553-3 and LC1553-3 (together referred to as Lx1553-3). The descriptions and examples discussed here are intended for use with programs written in the C language. Users of other languages should contact Ballard Technology for assistance.

Overview of the Lx1553-3 API

The API is divided into several functional categories. Some of these categories are based on the MIL-STD-1553 mode for which the function is used. Other categories are mode-independent. All function names contain a prefix (BC, RT, Mon, Msg, etc.) indicating the category to which they belong.

A few functions only apply to specific Lx1553-3 versions. For example, the error injection functions only apply to the Lx1553-3 version C. The acceptable parameters for some functions also may vary with the Lx1553-3 version. However, all functions apply to all Lx1553-3 versions unless otherwise noted.

Several aspects of the API deserve special mention.

"hCard" parameters

Nearly all functions require a card "handle" parameter. **`L13_CardOpen`**, which is always the first Lx1553-3 function called, returns this handle. The handle uniquely identifies a card when more than one Lx1553-3 is used in a single computer. The handle is always the last parameter in any function that requires it.

"ctrlflags" parameters

Many functions have a *ctrlflags* parameter. Each bit controls an option in this bitmask parameter. Constants are defined in the header (.H) file for these parameters. The name of a constant reflects the function in which it is used (e.g., RTCFG_DEFAULT is used in the **`L13_RTConfig`** function). Option parameters are always first in the parameter list of a function which accepts them. The default options can always be selected by using the ??_DEFAULT constant where ?? depends on the function in which it is used (e.g., BCCFG_DEFAULT). When multiple options are selected, the constants should be bitwise OR-ed together. The default options are shown in bold in this appendix. Since the default constants are *#defined* to 0, only non-default constants actually need to be included in the OR-ing. The constants defined in the header file should be used by name (not value) in your code since the values are subject to change.

"message" parameters

Message data and related information such as the time-tag are stored in individual message structures in the Lx1553-3. All of the message manipulation func-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

tions (e.g., **L13_MsgDataRd**) require a message address parameter that uniquely identifies a message structure. The **L13_??CreateMsg** functions create these structures and return their addresses. The message address parameter always immediately precedes the card handle in Lx1553-3 function calls.

Schedule indices (SCHNDX)

All of the scheduling functions (**L13_BCSched??**) return a value of type SCHNDX (schedule index). This is the index of the Command Block that the function appended to the schedule. This index is a parameter of some of the advanced scheduling functions (e.g., **L13_BCSchedCall**, **L13_BCSchedBranch**).

Table A-1 is a summary of the driver functions. As illustrated in the body of this manual, most applications can be implemented using just the bolded functions.

CARD Functions	
Function	**Description**
L13_CardClose	Disables access to a Lx1553-3 card and releases its hardware resources
L13_CardGetInfo	Gets model and version information from a Lx1553-3
L13_CardOpen	Enables access to a Lx1553-3 card and secure hardware resources
L13_CardReset	Resets the Lx1553-3 hardware; destroys all existing configuration data
L13_CardResume	Starts the card without resetting the BC schedule
L13_CardStart	Starts operation of the Lx1553-3 card
L13_CardStop	Stops operation of the Lx1553-3 card
L13_CardTest	Performs a hardware test on the Lx1553-3
BUS CONTROLLER Functions	
Function	**Description**
L13_BCConfig	Initializes a BC terminal configuration
L13_BCCreateList	Creates and initializes a message list for the BC.
L13_BCCreateMsg	Creates and initializes a message structure for the BC
L13_BCSchedBranch	Appends a conditional branch to the schedule
L13_BCSchedBuild	Automatically sequences messages and frames according to given frequencies
L13_BCSchedFrame	Appends the start of a new frame in the schedule
L13_BCSchedInt	Appends an interrupt (to the host PC) to the schedule
L13_BCSchedMsg	Appends a message transmission to the schedule
L13_BCSchedRetry	Appends a conditional retry of the last message transmission
L13_BCSetTimeout	Sets the maximum time the Lx1553-3 will wait before declaring an RT non-responsive.
L13_BCTransmitMsg	Transmits a single message without the need for a schedule.
REMOTE TERMINAL Functions	
Function	**Description**
L13_RTConfig	Initializes an RT terminal configuration
L13_RTCreateMsg	Creates and initializes a message structure for an RT
L13_RTCreateList	Creates an RT message list
L13_RTGetMsg	Gets the address of an existing message structure
L13_RTReset	Performs a logical reset of the RT (similar to Reset RT mode code)
L13_RTSWDWr	Sets the status word for the RT

Table A-1 Function summary

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MONITOR Functions	
Function	**Description**
L13_MonConfig	Initializes a monitor terminal configuration
L13_MonFilterTA	Configures terminal address filtering
L13_MonFilterSA	Configures subaddress filtering
L13_MonIsRunning	Determines whether the monitor is in the process of recording messages
L13_MonRd	Reads the next message record from the monitor list
L13_MonResume	Starts the Lx1553-3 monitor without resetting the ` pointers
L13_MonStart	Starts the Lx1553-3 monitor independently of other terminals
L13_MonStop	Stops the Lx1553-3 monitor independently of other terminals
MESSAGE Functions	
Function	**Description**
L13_MsgBlockRd	Reads an entire message structure from the Lx1553-3
L13_MsgBlockWr	Writes an entire message structure on the Lx1553-3
L13_MsgCommRd	Reads a message structure guaranteeing data integrity (slower than MsgBlockRd)
L13_MsgCommWr	Writes a message structure guaranteeing data integrity (slower than MsgBlockWr)
L13_MsgDataRd	Reads the data associated with a message
L13_MsgDataWr	Writes the data associated with a message
L13_MsgFieldWr	Writes a field to a message structure
L13_MsgFieldRd	Reads a field from a message structure
LIST Functions	
Function	**Description**
L13_ListBlockRd	Reads an entire message structure from the next entry in a list
L13_ListBlockWr	Writes an entire message structure to the next entry in a list
L13_ListDataRd	Reads the next data associated with a list
L13_ListDataWr	Writes the next data associated with a list
INTERRUPT Functions	
Function	**Description**
L13_IntConfig	Enables interrupts and initializes the interrupt log list
L13_IntInstall	Installs an interrupt handler (OS-dependent)
L13_IntRd	Reads an entry from the interrupt log list
L13_IntUninstall	Removes an interrupt handler (OS-dependent)
DATA CONVERSION/UTILITY Functions	
Function	**Description**
L13_ValFromAscii	Converts a string representation of an integer to a numeric value (specified base)
L13_ValGetBits	Extracts the specified bit field from an integer value.
L13_ValPackCWD	Creates a MIL-STD-1553 command word from its four fields
L13_ValPutBits	Inserts a bit field into an integer value.
L13_ValToAscii	Creates a string representation of an integer in a specified base
L13_ValUnpackCWD	Extracts the four bit fields from a MIL-STD-1553 command word
TIMER Functions	
Function	**Description**
L13_TimerClear	Clears the time-tag timer
L13_TimerRd	Reads the current value of the time-tag timer
L13_TimerResolution	Selects 1024us, 16us, or 1us time-tag timer resolution
ERROR INJECTION Functions	
Function	**Description**
L13_ErrorCtrl	Enables error injection
L13_ErrorDefine	Defines the error to be injected
L13_ErrorSent	Determines whether or not an error has been sent since the last L13_ErrorCtrl call
L13_ErrorTagBC	Tags the specified message as a candidate to receive an error (BC side)
L13_ErrorTagRT	Tags the specified message as a candidate to receive an error (RT side)

Table A-1 Function summary (cont'd)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Error values

Functions that return an address return a zero if an error is encountered. Functions that return a value other than an address return a negative value if an error is encountered. The negative error values are listed in Table A-2 as well as the predefined constants that can be used to test for them. See the function description to determine if a particular function can return these error values.

ERROR Values		
Name	**Value**	**Description**
ERR_NONE	0	No error
ERR_UNKNOWN	-1	An unexpected error occurred
ERR_BADVER	-2	Running under an untested version of Windows
ERR_BADPARAMS	-11	L13_CardOpen called with bad parameters
ERR_NOHANDLES	-12	L13_CardOpen already has allocated too many handles
ERR_NOCARD	-13	L13_CardOpen could not find a Lx1553 card at the specified address
ERR_NOIO	-14	L13_CardOpen could not find the I/O ports
ERR_NOMEM	-15	L13_CardOpen could not find the memory
ERR_BAD16BIT	-16	Card is conflicting with another 16-bit card
ERR_WRONGMODEL	-17	Card does not support this feature
ERR_NOSEL	-18	L13_CardOpen could not allocate a memory selector
ERR_LOCK	-19	The communication process is locked up
ERR_TOOMANY	-20	Too many terminals have been configured
ERR_BADHANDLE	-21	A bad handle was specified
ERR_BADTA	-22	A bad terminal address was specified
ERR_BADFRAME	-23	A bad frame time was specified
ERR_NOTBC	-24	The BC has not been configured
ERR_NOTRT	-25	The RT has not been configured
ERR_NOTMON	-26	The monitor has not been configured
ERR_ALLOC	-27	There is not enough memory to allocate
ERR_VXD	-28	Use WIN32 API GetLastError() to retrieve WIN32 system error from VXD
ERR_BADMSG	-31	The specified Command Block is not a message block
ERR_BADSCHNDX	-32	Specified command index is out of range
ERR_BUFSIZE	-33	Insufficient space in user buffer
ERR_NOCONFIG	-34	No configuration data is present; cannot start the card
ERR_BOOTFULL	-41	No space to add boot code
ERR_BOOTNUM	-42	There is no boot code with the specified number
ERR_ACCESS	-43	Unable to write to access register
ERR_ROMVERIFY	-44	Unable to verify the value written to the ROM
ERR_COUNT	-45	An invalid count was specified
ERR_CRC	-46	There was a bad checksum in the HEX file
ERR_FNAME	-47	Bad filenames were specified
ERR_FRDWR	-48	There was an error reading or writing the HEX file
ERR_HEX	-49	There was a bad hex character in the HEX file
ERR_INDEX	-51	The Command Block index was invalid or the schedule is full
ERR_NOMSGS	-52	No messages specified
ERR_RESPONSE	-53	A bad response or no response was received from an RT
ERR_TYPE	-54	There was a bad type value in the HEX file
ERR_ZEROLEN	-55	Zero length was specified
ERR_DMABAD	-61	Bad DMA channel was specified
ERR_DMAODD	-62	An odd DMA address or count was specified
ERR_DMAPAGE	-63	DMA buffer specified crosses a page boundary
ERR_SELFIOFAIL	-71	I/O selftest failed
ERR_SELFMEMFAIL	-72	Memory selftest failed
ERR_SELFCOMMFAIL	-73	Communication selftest failed
ERR_SELFXMTFAIL	-74	Transmit selftest failed

Table A-2 Errors summary

The following pages contain descriptions of the primary Lx1553-3 driver functions. The constants in bold in the tables on the following pages are the default options. Note that the "L13_" prefix has been omitted from the headings for easier reading, but all Lx1553-3 functions must begin with "L13_" in source code.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCConfig..... Initializes a BC terminal configuration.

SYNOPSIS: ERRVAL L13_BCConfig(ctrlflags, hCard)
ULONG ctrlflags Selects configuration options
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, otherwise zero.

DESCRIPTION: Configures the specified Lx1553-3 to simulate a BC. This function allocates Lx1553-3 memory for a schedule of 512 Command Blocks and initializes internal variables associated with the schedule. Therefore, any previously created schedules are cleared by calling this function.

ctrlflags	
Constant	**Description**
BCCFG_DEFAULT	Select all default settings (**bold** below)
BCCFG_DISABLE	Disable bus controller
BCCFG_ENABLE	Enable bus controller
BCCFG_TRIGNONE	No external trigger
BCCFG_TRIGEXT	Trigger external
BCCFG_TRIGSTART	Trigger start
BCCFG_NOINTHALT	Do not generate interrupt on halt
BCCFG_INTHALT	Generate interrupt on halt
BCCFG_NOINTPAUSE	Do not generate interrupt on pause
BCCFG_INTPAUSE	Generate interrupt on pause
BCCFG_SYNCALL	Sync on all messages
BCCFG_SYNCSEL	Selective sync
BCCFG_NOLOOPMAX	Disable maximum loop count
BCCFG_LOOPMAX	Enable maximum loop count
BCCFG_MCNONE	Disable mode codes
BCCFG_MC0	Select SA=00000 for mode codes
BCCFG_MC1	Select SA=11111 for mode codes
BCCFG_MC01	Select SA=00000 or SA=11111 for mode codes
BCCFG_NOSTEP	Disables single-stepping the schedule
BCCFG_STEP	Enables single-stepping schedule

WARNINGS: None.

SEE ALSO: L13_RTConfig, L13_MonConfig

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCCreateList ..... Creates and initializes a message list for the BC.

SYNOPSIS: LISTADDR L13_BCCreateList(listctrlflags, count, msgctrlflags, cwd1, cwd2, data, hCard)

ULONG listctrlflags	Selects list options
ULONG count	Number of entries in list
ULONG msgctrlflags	Selects message options
USHORT cwd1	Command word 1 to transmit
USHORT cwd2	Command word 2 (RT-RT only)
LPWORD data	Buffer containing data words
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The Lx1553-3 address of the list if successful, otherwise zero.

DESCRIPTION: Similar to L13_BCCreateMsg except creates a list. Function allocates memory for a list of message structures and initializes each entry with the command and data words provided. If the *data* pointer is NULL, data initialization is skipped.

listctrlflags	
Constant	**Description**
LISTCRT_DEFAULT	Select all default settings (**bold** below)
LISTCRT_PINGPONG	Selects ping-pong mode
LISTCRT_FIFO	Selects FIFO mode
LISTCRT_CIRCULAR	Selects circular mode
LISTCRT_TRBIT	The host will read or write data depending upon the T/R bit of *cwd1* if a BC-RT transfer, or read data if an RT-RT transfer
LISTCRT_READ	The host will read receive data from the list
LISTCRT_WRITE	The host will write transmit data to the list
LISTCRT_MODEL	List is created in static RAM
LISTCRT_RAM	List is created in static RAM
LISTCRT_DPRAM	List is created in dual-port RAM

Continued on following page...

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

msgctrlflags	
Constant	**Description**
MSGCRT_DEFAULT	Select all default settings (**bold** below)
MSGCRT_BUSA	Message will be transmitted on bus A
MSGCRT_BUSB	Message will be transmitted on bus B
MSGCRT_BCRT	Message is a BC-RT transfer
MSGCRT_RTRT	Message is an RT-RT transfer
MSGCRT_NOINT	Message will not generate interrupt
MSGCRT_INT	Message will generate interrupt
MSGCRT_NOERR	This message is not a candidate for errors
MSGCRT_ERR	This message is a candidate for errors
MSGCRT_NOTIMETAG	Message will not record time-tag
MSGCRT_TIMETAG	Message will record time-tag
MSGCRT_NOELAPSE	Message will not record elapsed time
MSGCRT_ELAPSE	Message will record elapsed time
MSGCRT_NOHIT	Message will not record hit count
MSGCRT_HIT	Message will record hit count
MSGCRT_NOMON	Message will not be monitored
MSGCRT_MON	Message will be monitored
MSGCRT_NOMIN	Message will not record min time
MSGCRT_MIN	Message will record min time
MSGCRT_NOMAX	Message will not record max time
MSGCRT_MAX	Message will record max time
MSGCRT_NOSYNC	No sync will be generated for message
MSGCRT_SYNC	Sync will be generated for message
MSGCRT_WIPE0	Zeros the message structure on creation
MSGCRT_NOWIPE	Do not zero the message on creation
MSGCRT_WIPE123	Initialize the data words with incrementing values
MSGCRT_WIPECWD	Initialize the data words with command word values

WARNINGS: None.

SEE ALSO: L13_RTCreateList, L13_ListDataWr, L13_ListDataWr
L13_BCCreateMsg, L13_BCSchedMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCCreateMsg..... Creates and initializes a message structure for the BC.

SYNOPSIS: MSGADDR L13_BCCreateMsg(ctrlflags, cwd1, cwd2, data, hCard)

ULONG ctrlflags	Selects message options
USHORT cwd1	Command word 1 to transmit
USHORT cwd2	Command word 2 (RT-RT only)
LPWORD data	Buffer containing data words
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The Lx1553-3 address of the message structure if successful, otherwise zero.

DESCRIPTION: Allocates memory for a message structure and initializes the structure with the command and data words provided. If the *data* pointer is NULL, data initialization is skipped.

ctrlflags	
Constant	**Description**
MSGCRT_DEFAULT	Select all default settings (**bold** below)
MSGCRT_BUSA	Message will be transmitted on bus A
MSGCRT_BUSB	Message will be transmitted on bus B
MSGCRT_BCRT	Message is a BC-RT transfer
MSGCRT_RTRT	Message is an RT-RT transfer
MSGCRT_NOINT	Message will not generate interrupt
MSGCRT_INT	Message will generate interrupt
MSGCRT_NOERR	This message is not a candidate for errors
MSGCRT_ERR	This message is a candidate for errors
MSGCRT_NOTIMETAG	Message will not record time-tag
MSGCRT_TIMETAG	Message will record time-tag
MSGCRT_NOELAPSE	Message will not record elapsed time
MSGCRT_ELAPSE	Message will record elapsed time
MSGCRT_NOHIT	Message will not record hit count
MSGCRT_HIT	Message will record hit count
MSGCRT_NOMON	Message will not be monitored
MSGCRT_MON	Message will be monitored
MSGCRT_NOMIN	Message will not record min time
MSGCRT_MIN	Message will record min time
MSGCRT_NOMAX	Message will not record max time
MSGCRT_MAX	Message will record max time
MSGCRT_NOSYNC	No sync will be generated for message
MSGCRT_SYNC	Sync will be generated for message
MSGCRT_WIPE0	Zeros the message structure on creation
MSGCRT_NOWIPE	Do not zero the message on creation
MSGCRT_WIPE123	Initialize the data words with incre-menting values
MSGCRT_WIPECWD	Initialize the data words with command word values
MSGCRT_MODEL	Messages are created in dual-port RAM on Lx1553-3 version A, otherwise in static RAM
MSGCRT_RAM	Messages are created in static RAM
MSGCRT_DPRAM	Messages are created in dual-port RAM

WARNINGS: None.

SEE ALSO: L13_RTCreateMsg, L13_BCCreateList

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSchedBranch ..... Appends a conditional branch to the schedule.

SYNOPSIS: SCHNDX L13_BCSchedBranch(condition, destindex, hcard)

USHORT condition	Condition for branch
SCHNDX destindex	Destination index for branch
HCARD hcard	Handle of Lx1553-3 card

RETURNS: The index of the appended block if successful, or a negative value if an error occurs.

DESCRIPTION: Appends a conditional branch command block to the current end of the schedule. A conditional branch command block causes the Lx1553-3 to branch to the index in the schedule specified by *destindex* if *condition* evaluates as TRUE.

condition specifies one or more conditions that can cause a branch. The conditions are applied to the last message that was transmitted in the schedule.

The branch is taken if one or more of the conditions is TRUE. The conditions flags listed below may be ORed together.

level	
Constant	**Condition**
COND_FAIL	When all retries have failed
COND_SRQ	Service Request bit set in status word
COND_INS	Instrumentation bit set in status word
COND_SSF	Subsystem Fail bit set in status word
COND_TF	Terminal Flag bit set in status word
COND_BUSY	Busy bit set in status word
COND_ME	Message Error bit set in status word
COND_RESPERR	Error in the status word response
COND_NORESP	No status word response received

WARNINGS: A call to L13_BCConfig must precede this function.

SEE ALSO: L13_BCSchedFrame, L13_BCSchedMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSchedBuild..... Sequences messages and frames by given frequencies.

SYNOPSIS: ERRVAL L13_BCSchedBuild(nummsgs, msgs, freqs, hCard)

USHORT nummsgs	Number of messages to schedule
LPMSGADDR msgs[]	Array of message addresses on card
LPUSHORT freqs[]	Array of message frequencies
HCARD hCard	Handle of Lx1553-3 card

RETURNS: Zero if the schedule was successfully built, or a negative value if an error occurred.

DESCRIPTION: Clears any BC schedule from the specified card and creates a new BC schedule. The new BC schedule will consist of *nummsgs* messages, each transmitted at a specific frequency. *msgs* points to an array of message addresses, each previously generated by a call to L13_BCCreateMsg. *freqs* points to an array of frequencies in units of Hz.

The function schedules messages and frames to generate the specified frequencies. If the schedule cannot be generated, an error is returned.

WARNINGS: A call to L13_BCConfig must precede this function as well as a call to L13_BCCreateMsg for each message to be scheduled.

SEE ALSO: L13_BCSchedFrame, L13_BCSchedMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSchedFrame ..... Appends the start of a new frame to the schedule.

SYNOPSIS: SCHNDX L13_BCSchedFrame(timeval, hCard)

ULONG timeval	Length of frame in microseconds
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The index of the appended Command Block if successful, or a negative value if an error occurs.

DESCRIPTION: Appends a Frame Command Block to the current end of the schedule. A Frame Command Block marks the start of a new frame. When the Lx1553-3 encounters a Frame Command Block it pauses in its processing of the schedule until the timer set by a *previous* Frame Command Block reaches zero. It then resets the timer to the *timeval* specified by the current Frame Command Block and resumes processing of the schedule.

Internally, the frame time is a 16-bit multiple of 20µs, so the *timeval* parameter is rounded to the nearest multiple of 20. This gives a range of 0–1,310,720µs.

WARNINGS: A call to L13_BCConfig must precede this function.

SEE ALSO: L13_SchedMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSchedInt .....Appends an interrupt (to the host PC) to the schedule.

SYNOPSIS: SCHNDX L13_BCSchedInt(tagval, hCard)
USHORT tagval Interrupt tag value
HCARD hCard Handle of Lx1553-3 card

RETURNS: The index of the appended block if successful, or a negative value if an error occurs.

DESCRIPTION: Schedules an interrupt to the host PC. When the interrupt is encountered in the schedule, an interrupt type value of INTTYPE_OPCODE and the user-specified value *tagval* are written to the interrupt log list. Both values can then be read from the interrupt log list using the L13_IntRd function.

WARNINGS: A call to L13_BCConfig must precede this function.

SEE ALSO: L13_IntRd

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSchedMsg ..... Appends a message transmission to the schedule.

SYNOPSIS: SCHNDX L13_BCSchedMsg(message, hCard)

MSGADDR message	Address of message or list
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The index of the appended Command Block if successful, or a negative value if an error occurs.

DESCRIPTION: Appends a Message Command Block to the current end of the schedule. A Message Command Block triggers transmission of the message or list specified by *message*.

Note: Execution of this function does NOT transmit the message. The message is transmitted only when the resulting schedule is executed after L13_CardStart starts the Lx1553-3.

WARNINGS: A call to L13_BCConfig must precede this function. In addition, the message must have been created with L13_BCCreateMsg or L13_BCCreateList.

SEE ALSO: L13_BCCreateMsg, L13_BCCreateList, L13_BCSchedFrame, L13_BCSchedRetry

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSchedRetry ..... Appends a conditional retry of the last transmission.

SYNOPSIS: SCHNDX L13_BCSchedRetry(ctrlflags, retries, hCard)
USHORT ctrlflags Conditions on which to perform retries
USHORT retries Number of retries (0–255)
HCARD hCard Handle of Lx1553-3 card

RETURNS: The index of the appended Command Block if successful, or a negative value if an error occurs.

DESCRIPTION: Appends a Retry Command Block to the current end of the schedule. A Retry Command Block causes the Lx1553-3 to retry the transmission of the most recently transmitted message subject to the conditions specified by *ctrlflags*. The conditions flags listed below may be OR-ed together. The Lx1553-3 retransmits the message up to *retries* times until transmission succeeds.

ctrlflags	
Constant	**Condition**
COND_SRQ	Service Request bit set in status word
COND_INS	Instrumentation bit set in status word
COND_SSF	Subsystem Fail bit set in status word
COND_TF	Terminal Flag bit set in status word
COND_BUSY	Busy bit set in status word
COND_ME	Message Error bit set in status word
COND_RESPERR	An error in the status word response
COND_NORESP	No status word response received
COND_ALTBUS	Retries are performed on alternate bus

WARNINGS: A call to L13_BCConfig must precede this function.

SEE ALSO: L13_BCSchedFrame, L13_BCSchedMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCSetTimeout ..... Sets the maximum BC timeout.

SYNOPSIS: ERRVAL L13_BCSetTimeout(timeoutval, hCard)

USHORT timeoutval	Timeout value (in units of 0.1µs)
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: When simulating the BC, sets the maximum time the Lx1553-3 will wait before declaring an RT non-responsive. *timeoutval* is the no-response time-out value plus 3.0 µs. The default value for *timeoutval* is 190, making the default no-response time-out 16 µs as measured by MIL-STD-1553B.

WARNINGS: This function should not be called while the card is running. L13_BCSetTimeout only operates on Lx1553-3 version C cards.

SEE ALSO: L13_BCConfig

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BCTransmitMsg ..... Transmit a single message.

SYNOPSIS: ERRVAL L13_BCTransmitMsg(msg, hCard)

LPXMITFIELDS msg	Pointer to structure containing message parameters
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, otherwise zero.

DESCRIPTION: Transmits the message specified in the *msg* structure immediately, bypassing the scheduling mechanism. The first call to this function creates a special message structure. Subsequent calls reuse this same structure, so it is not necessary to create the message with L13_BCCreateMsg.

This function does not return until transmission is complete or a no-response timeout occurs. This function does not affect or rely on the BC schedule.

The XMITFIELDS structure is declared in the header file and shown below. All of its members, except for *ctrlflags*, correspond with those of internal message structures created by L13_BCCreateMsg. The *ctrlflags* member is a bit field, the bits of which correspond to the MSGCRT_ constants passed to L13_BCCreateMsg.

```
typedef struct
{
        ULONG  ctrlflags;      //User writes message configuration options
        USHORT flag1;          //Card writes message flag 1
        USHORT flag2;          //Card writes message flag 2
        USHORT errflags;       //Card writes error flags
        USHORT actflags;       //Card writes activity flags
        USHORT resptime1;      //Card writes response time 1 (0.1µs units)
        USHORT resptime2;      //Card writes response time 2 (RT-RT only)
        USHORT datacount;      //Card writes data count
        USHORT resv7;          //Reserved
        ULONG  timetag;        //Card writes time tag
        ULONG  elapsetime;     //Card writes elapsed time
        USHORT resv12;         //Reserved
        USHORT resv13;         //Reserved
        USHORT resv14;         //Reserved
        USHORT resv15;         //Reserved
        USHORT resv16;         //Reserved
        USHORT resv17;         //Reserved
        USHORT resv18;         //Reserved
        USHORT resv19;         //Reserved
        USHORT cwd1;           //User writes command word 1
        USHORT cwd2;           //User writes command word 2 (RT-RT only)
        USHORT swd1;           //Card writes status word 1
        USHORT swd2;           //Card writes status word 2 (RT-RT only)
        USHORT data[32];       //User or card writes data words
        USHORT extra[8];       //User writes extra words (error injection)
} XMITFIELDS;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

WARNINGS: A call to L13_BCConfig must precede this function.

Note: Bus errors (in the message transmission) are returned in the *errflags* field of the *msg* structure. Internal Lx1553-3 errors (e.g., ERR_NOTBC) are returned in the function's return value.

SEE ALSO: L13_BCCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CardClose ..... Disables access to a Lx1553-3 card.

SYNOPSIS: ERRVAL L13_CardClose(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Disables access to the specified Lx1553-3 card and releases the associated device hardware resources. This function does not stop the Lx1553-3 from operating.

WARNINGS: This function MUST be called before a program terminates to prevent the associated hardware resources from being lost. This is especially important in Microsoft Windows operating systems.

SEE ALSO: L13_CardOpen

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CardGetInfo.....Get model and version information from a Lx1553-3.

SYNOPSIS: ULONG L13_CardGetInfo(infotype, hCard)
USHORT infotype Type of information to be returned
HCARD hCard Handle of Lx1553-3 card

RETURNS: The requested information about the specified card.

DESCRIPTION: Used to determine information about the specified Lx1553-3. Model, version number and supported features can be determined. The *infotype* constants listed below may be used to specify the requested information.

Infotype	
Constant	**Description**
INFOTYPE_PLAT	Returns the Lx1553-3 platform type ("LP"=0x00004C50, "LC"=0x00004C43)
INFOTYPE_PROD	Returns the Lx1553-3 product type (1553)
INFOTYPE_DASH	Returns the Lx1553-3 dash number (3)
INFOTYPE_VERSION	Returns the Lx1553-3 version number (0x0000MMmm – MM = Major version, mm = minor version)
INFOTYPE_DATE	Returns the Lx1553-3 version date (0xYYYYMMDD – YYYY = year, MM = month, DD = day)
INFOTYPE_MODEL	Returns the Lx1553-3 model number (1 = A, 2 = B, 3 = C)
INFOTYPE_MAXCOUNT	Returns the number of terminals the Lx1553-3 can simulate (1,4, 32)
INFOTYPE_MULTIMODE	Returns whether Lx1553-3 can simulate multiple modes (BC, RT, and/or MON). (Only TRUE for B & C cards)
INFOTYPE_ERRORGEN	Returns whether Lx1553-3 can generate errors (Only TRUE for version C cards)
INFOTYPE_CONCURMON	Returns whether Lx1553-3 has a concurrent monitor (Only TRUE for version C cards)
INFOTYPE_MONFILTER	Returns whether Lx1553-3 has a monitor filter

Features of different Lx1553-3 versions				
	Versions			
	A	B4	B32	C
Number of Simultaneous Terminals	1	4	32	32
Monitor	✔	✔	✔	✔
Filtering on TA, SA		✔	✔	✔
Concurrent terminal monitoring				✔
Error Injection				✔
Bundled Software	✔	✔	✔	✔
CoPilot 1553 compatible	✔	✔	✔	✔

WARNINGS: None.

SEE ALSO: None.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CardResume.....Starts the card without resetting the BC schedule.

SYNOPSIS: ERRVAL L13_CardResume(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an unable to resume, otherwise zero.

DESCRIPTION: Reactivates the specified Lx1553-3 from the point at which it was stopped using L13_CardStop. The following list compares the difference between calling L13_CardResume and L13_CardStart.

Feature	When CardStart is called....	When CardResume is called....
BC Schedule	Execution starts at the start of the BC schedule.	Execution resumes at the point the BC schedule was stopped.
Interrupt Log List	Any unread entries in the interrupt log list are cleared before the card is started.	Any unread records in the interrupt log list are pre-served before the card is started.
Monitor	Any unread records in the sequential monitor are cleared before the card is started.	Any unread records in the sequential monitor are pre-served before the card is resumed.

WARNINGS: A call to L13_CardStop must precede this function.

SEE ALSO: L13_CardStart, L13_CardStop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CardStart..... Starts operation of the specified Lx1553-3.

SYNOPSIS: ERRVAL L13_CardStart(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: Returns a negative value if unable to start, otherwise zero.

DESCRIPTION: Activates the specified Lx1553-3 according to its configuration:
BC: Starts processing its schedule.
RT(s): Start responding to commands.
Monitor: Starts recording bus traffic.

On versions capable of multi-terminal simulation, all terminals that the specified Lx1553-3 is configured to simulate become active simultaneously.

WARNINGS: The Lx1553-3 continues operating even after an application program ends unless L13_CardStop halts it.

SEE ALSO: L13_CardStop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CardStop ..... Stops operation of the specified Lx1553-3.

SYNOPSIS: BOOL L13_CardStop(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: TRUE if the card was active.

DESCRIPTION: Stops operation of the specified Lx1553-3. On versions capable of multi-terminal simulation all simulated terminals (BC, RT and monitor) are halted. If a 1553 message is being processed, the processing is allowed to finish before the card is halted.

WARNINGS: None.

SEE ALSO: L13_CardStart

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CardTest..... Performs a hardware test on the Lx1553-3.

SYNOPSIS: ERRVAL L13_CardTest(level, hCard)
USHORT level Level of tests to perform
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Executes the test specified by the *level* constant. When the test completes, the card is left in the same state as after a L13_CardReset.

Level	
Constant	**Description**
TEST_LEVEL_0	Tests the port I/O interface of the Lx1553-3. The test reads and writes each I/O port with a walking-bit pattern.
TEST_LEVEL_1	In addition to testing Level 0 tests, this level tests the memory interface of the Lx1553-3. The test performs a pattern test of the RAM.
TEST_LEVEL_2	In addition to testing Level 1 tests, this level tests the communication process of the Lx1553-3. The test performs a pattern test of the RAM using the communication process.
TEST_LEVEL_3	In addition to testing Level 2 tests, this level tests the bus transceivers of the Lx1553-3. The card transmits isolated data words on bus A and bus B and verifies they are received. (This utilizes a wrap through the transceiver on the Lx1553-3.)

WARNINGS: This function disrupts normal operation of the card. Care should be taken when this function is called for a card on an active databus.

SEE ALSO: L13_CardReset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ErrorCtrl ..... Enables error injection.

SYNOPSIS: ERRVAL L13_ErrorCtrl(ctrlflags, hCard)

MSGADDR ctrlflags	Select control options
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: This function is used to enable error injection after an error has been defined. The *ctrlflags* parameter controls both:

- how errors are triggered.
- whether all or only tagged messages can contain the error

Typically either _ON, _ONCE or _EXTERNAL is bitwise OR-ed with either _TAGMSG or _ANYMSG to produce the *ctrlflags* value.

ctrlflags	
Constant	**Description**
ERRCTRL_OFF	Completely disables error injection
ERRCTRL_ON	Errors are injected in all candidate messages until error injection is disabled.
ERRCTRL_ONCE	An error is injected only in the next candidate message.
ERRCTRL_EXTERNAL	An error is injected in the first candidate message following an external trigger pulse.
ERRCTRL_TAGMSG	Only messages which have been tagged with L13_ErrorTag?? are candidates
ERRCTRL_ANYMSG	All messages are candidates

WARNINGS: The CTRL_ANYMSG flag should be used with extra care especially when multiple terminals are being simulated. When CTRL_ANYMSG is specified, any transmission from any simulated terminal (BC or RT) may contain the error.

This and other L13_Error?? functions only operate on Lx1553-3 version C.

SEE ALSO: L13_ErrorTagBC, L13_ErrorTagRT, L13_ErrorDefine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ErrorDefine..... Defines an error to be injected.

SYNOPSIS: ERRVAL L13_ErrorDefine(ctrlflags, value, count, wordpos, bitpos, hCard)

USHORT ctrlflags	Selects the type of error
USHORT value	See below
INT count	See below
USHORT wordpos	See below
USHORT bitpos	See below
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Defines the error to be injected into any subsequent message transmission. Since only one error may be defined at a time, the *ctrlflags* constants in the following table may NOT be bitwise OR-ed together.

ctrlflags	
Constant	**Description**
ERRDEF_NONE	Disable error generation (default)
ERRDEF_GAP	Generate a gap error preceding a word
ERRDEF_CNTWRDABS	Generate an absolute word count error
ERRDEF_CNTWRDREL	Generate a word count error interpreting count as relative to the word count in the command word
ERRDEF_CNTBIT	Generate a bit count error
ERRDEF_MAN1	Generate Manchester error on first half of a specified bit
ERRDEF_MAN2	Generate Manchester error on second half of a specified bit
ERRDEF_SYNC	Generate a sync error (invert the sync)
ERRDEF_PAR	Generate a parity error (invert the parity)

For any given type of error some of the parameters of this function are unused. The following table shows which parameters apply to each error type.

PARAMETERS USED				
ctrlflags	***wordpos***	***bitpos***	***count***	***value***
ERRDEF_PAR	4			
ERRDEF_SYNC	4			
ERRDEF_MAN1 or _MAN2	4	4		
ERRDEF_CNTWRDABS or _CNTWRDREL			4	
ERRDEF_CNTBIT	4		4	4
ERRDEF_GAP	4			4

The parameters are described on the following page.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

value Specifies a time for gap errors or the values of any extra bits in a bit count error. The *value* parameter may be 0-1024 for gap errors, and the units of time are 0.1µs. The least significant bits of *value* determine the values of extra bits in a bit count error.

count Specifies the size of a bit or word count error. For bit count errors, *count* may be in the range of -2 to +2. For absolute word-count errors, *count* may be from 0 to 40. For relative word count errors, the sum of *count* and the word count field in the command word must be less than or equal to 40.

wordpos Specifies the location within a message of a word error. The *wordpos* parameter may be a value from 0 to 31 indicating a data word, or it may be one of the constants from the following table.

***wordpos* PARAMETER**	
Constant	**Description**
ERRDEF_CWD1	Specifies the first command word.
ERRDEF_CWD2	Specifies the second command word.
ERRDEF_SWD1	Specifies the first status word.
ERRDEF_SWD2	Specifies the second status word.

bitpos The *bitpos* specifies the bit containing a Manchester error. It may range from 0 to 19.

WARNINGS: This and other L13_Error?? functions only operate on Lx1553-3 version C.

SEE ALSO: L13_ErrorCtrl, L13_ErrorSent, L13_ErrorTagBC, L13_ErrorTagRT

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ErrorSent.....Determines whether an error has been sent.

SYNOPSIS: BOOL L13_ErrorSent(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: TRUE if an error has been sent since the last call to L13_ErrorCtrl, FALSE otherwise.

DESCRIPTION: Checks an internal flag that indicates whether at least one message with an error has been transmitted. This flag is set when an error is sent. It remains set until cleared by L13_ErrorCtrl. This function is used to verify that the currently defined error has been sent before redefining it.

WARNINGS: This and other L13_Error?? functions only operate on Lx1553-3 version C.

This function cannot detect the occurrence of errors triggered by the external trigger signal.

SEE ALSO: L13_ErrorCtrl, L13_ErrorDefine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ErrorTagBC.... Tags the specified message as a candidate to receive an error.

SYNOPSIS: ERRVAL L13_ErrorTagBC(tagval, message, hCard)

BOOL tagval	Tag if TRUE, untag if FALSE
MSGADDR message	Message to receive error
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Makes the specified message a candidate to contain errors when error injection is enabled.

WARNINGS: This and other L13_Error?? functions only operate on Lx1553-3 version C.

SEE ALSO: L13_ErrorTagRT, L13_ErrorDefine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ErrorTagRT.... Tags the specified message as a candidate to receive an error.

SYNOPSIS: ERRVAL L13_ErrorTagRT(tagval, mcflag, taval, trval, saval, hCard)

BOOL tagval	Tag if TRUE, untag if FALSE
BOOL mcflag	TRUE if mode code, FALSE if subaddress
INT taval	Terminal address
BOOL trval	T/R bit
INT saval	Subaddress or mode-code number
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Makes the specified message a candidate to contain errors when error injection is enabled.

WARNINGS: This and other L13_Error?? functions only operate on Lx1553-3 version C.

SEE ALSO: L13_ErrorTagBC, L13_ErrorDefine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IntConfig ..... Configures the interrupt log list.

SYNOPSIS: ERRVAL L13_IntConfig(ctrlflags, count, hCard)

USHORT ctrlflags	Selects the configuration options
USHORT count	Number of entries in the interrupt log list
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Enables the interrupt log list of the Lx1553-3 card. *count* sets the number of entries in the card's interrupt log list. *count* must be a positive value specified by the user. *ctrlflags* can be one of the following constants.

ctrlflags	
Constant	**Condition**
INTCFG_DEFAULT	Select all default settings
INTCFG_ENABLE	Enable interrupt log list (default)
INTCFG_DISABLE	Disable interrupt log list

WARNINGS: This function does not enable interrupts to the host. The interrupt log list can be polled by an application without installing an interrupt handler.

SEE ALSO: L13_IntInstall

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IntInstall (WIN32) ..... Installs an interrupt handler under WIN32.

SYNOPSIS: ERRVAL L13_IntInstall(hEvent, hCard)
LPVOID hEvent Handle of a WIN32 event object
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Associates a WIN32 event object with interrupts from the card specified by *hCard*. If the function is successful, any interrupt issued from the Lx1553-3 causes the event object specified by *hEvent* to be set to the signaled state.

The user's application must ensure that the event object is set to the unsignaled state before the Lx1553-3 issues the first interrupt. This can be done when creating the event object with the WIN32 API function *CreateEvent*.

Note that event objects are never polled. Create a worker thread which immediately goes to sleep by calling a WIN32 API wait function like *WaitForSingleObject*. When the Lx1553-3 issues an interrupt, the event object is signaled, and the worker thread wakes up to respond to the interrupt.

It is the user's responsibility to clear the interrupt from the Lx1553-3 by calling L13_IntReset. The user must also set the event object to the unsignaled state.

WARNINGS: If this function is used, L13_IntUninstall MUST be called before the user's program terminates. It removes the association between the Lx1553-3 and the event object.

SEE ALSO: L13_IntInstall (DOS), L13_IntRd, L13_IntUninstall

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IntRd ..... Reads the next entry from the interrupt log list.

SYNOPSIS: ULONG L13_IntRd(typeval, infoval, hCard)

LPUSHORT typeval	Pointer to variable to receive type value
LPUSHORT infoval	Pointer to variable to receive info value
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The address of the entry in the interrupt log list, or zero if it is empty and there are no entries to read.

DESCRIPTION: Reads a single entry from the interrupt log list. Each entry contains two values – the type of interrupt that generated the entry, and an information word associated with the interrupt. The type of interrupt is passed through *typeval*, and the information word is passed through *infoval*.

typeval is a 16-bit value with the following two fields:

Reserved								Type Value							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The value returned in *infoval* depends upon the *typeval* values listed below.

typeval	Description	*infoval*
INTTYPE_MSG	The interrupt option was enabled on a message that was processed	Value of the command word
INTTYPE_OPCODE	An interrupt was encountered in a BC schedule	Tag value passed in L13_SchedInt()
INTTYPE_HALT	The BC schedule encountered a halt opcode	None
INTTYPE_PAUSE	The BC schedule encountered a pause opcode	None
INTTYPE_SEQ	The sequential record is full	None
INTTYPE_LIST	A list buffer was full or empty.	Address of the list buffer

Note: *typeval* and *infoval* must each be large enough to store at least one 16-bit word. If *typeval* or *infoval* is a NULL pointer, that parameter is not returned.

WARNINGS: This function should be preceded by a call to L13_IntConfig. To use this function, it is not necessary to install an interrupt handler.

SEE ALSO: L13_IntConfig, L13_IntInstall

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IntUninstall ..... Removes an interrupt handler.

SYNOPSIS: ERRVAL L13_IntUninstall(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Removes an interrupt handler installed by the L13_IntInstall function. The Lx1553-3's interrupt log list is remains unchanged.

WARNINGS: This function must be called before the user's application terminates if L13_IntInstall has been called.

SEE ALSO: L13_IntConfig, L13_IntInstall

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ListDataRd ..... Reads the next data associated with a list.

SYNOPSIS: INT L13_ListDataRd(buf, count, list, hCard)

LPWORD buf	Pointer to destination
INT count	Number of words to read
LISTADDR list	List from which to read data
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The number of data words read from the *list* or zero if an error occurred or unable to read from the list.

DESCRIPTION: Similar to L13_MsgDataRd except reads from a list. This function copies *count* number of data words to *buf* from the message structure in the list specified by the *list* parameter. The *list* parameter is the value returned when the list was created using L13_BCCreateList or L13_RTCreateList. The position of the message to be read is determined by the mode of the list as follows:

CIRCULAR MODE: This function should not be used in this mode.

FIFO MODE: Reads the oldest complete message received.

PINGPONG MODE: Reads the newest complete message received.

WARNINGS: *list* must be configured as a read (receive) list.

SEE ALSO: L13_MsgDataRd, L13_ListDataWr, L13_BCCreateList, L13_RTCreateList

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ListDataWr ..... Writes the next data associated with a list.

SYNOPSIS: INT L13_ListDataWr(buf, count, list, hCard)

LPWORD buf	Pointer to data words
INT count	Number of words to write
LISTADDR list	List to receive new data
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The number of data words written to the *list* or zero if an error occurred or unable to write to the list.

DESCRIPTION: Similar to L13_MsgDataWr except writes to a list. This function copies *count* data words from the *buf* to the message structure in the list specified by the *list* parameter. The *list* parameter is the value returned when the list was created using L13_BCCreateList or L13_RTCreateList. The position to which the message is written in the list is determined by the mode of the list as follows:

CIRCULAR MODE: If the number of writes exceeds the number of entries in the list, data is overwritten starting at the top of the list.

FIFO MODE: Data is written to one end of the list and is transmitted and removed from the other end of list. This function returns a zero if the list is full.

PINGPONG MODE: When writing is complete, the data will be used for the next message transmission.

WARNINGS: *list* must be configured as a write (transmit) list.

SEE ALSO: L13_MsgDataWr, L13_ListDataRd, L13_BCCreateList, L13_RTCreateList

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonConfig ..... Configures the monitor of the card.

SYNOPSIS: ERRVAL L13_MonConfig(ctrlflags, hCard)
ULONG ctrlflags Selects configuration options
HCARD hCard Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, otherwise zero.

DESCRIPTION: Configures the Lx1553-3 to monitor bus traffic, allocates an on-board buffer to hold message records and initializes internal pointers associated with the buffer.

ctrlflags	
Constant	**Description**
MONCFG_DEFAULT	Select all default settings (**bold** below)
MONCFG_DISABLE	Disable monitor
MONCFG_FILLHALT	Enable monitor in fill and halt mode
MONCFG_CONTINUOUS	Enable monitor in continuous mode
MONCFG_16K	Allocate a 16K monitor buffer
MONCFG_ALLAVAIL	Allocate all available memory to monitor buffer
MONCFG_32K	Allocate a 32K monitor buffer
MONCFG_64K	Allocate a 64K monitor buffer
MONCFG_128K	Allocate a 128K monitor buffer
MONCFG_DPRAM	Allocate all available dual-port RAM to monitor buffer
MONCFG_NOINTFULL	Do not generate interrupt when monitor full
MONCFG_INTFULL	Generate interrupt when monitor full

WARNINGS: Only the Lx1553-3 version C supports monitoring concurrent with BC and/or RT simulation.

Note: If the MONCFG_ALLAVAIL flag is used, L13_MonConfig should be the last of the L13_??Config and L13_??CreateMsg functions called before L13_CardStart.

SEE ALSO: L13_MonFilterTA, L13_MonFilterSA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonFilterSA.....Configures subaddress filtering.

SYNOPSIS: ERRVAL L13_MonFilterSA(ta, rsa, tsa, rmc, tmc, hCard)

USHORT ta	Terminal address to which to apply subaddress filtering
ULONG rsa	Bitmask of receive subaddresses
ULONG tsa	Bitmask of transmit subaddresses
ULONG rmc	Bitmask of receive mode codes
ULONG tmc	Bitmask of transmit mode codes
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, otherwise zero.

DESCRIPTION: Specifies which messages to monitor for the terminal address specified by *ta*. If this function is not used, all messages to the terminal address are monitored.

In each bitmask parameter, the least significant bit corresponds to subaddress or mode code number zero. The most significant bit corresponds to subaddress or mode code number 31.

WARNINGS: L13_MonFilter?? functions only operate on Lx1553-3 versions B and above.

SEE ALSO: L13_MonFilterTA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonFilterTA ..... Configures terminal address filtering.

SYNOPSIS: ERRVAL L13_MonFilterTA(termaddresses, hCard)

ULONG termaddresses	Bitmask specifying terminal addresses to monitor
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, otherwise zero.

DESCRIPTION: Specifies which terminal addresses to monitor. If this function is not used, all terminal addresses are monitored.

The least significant bit in *termaddresses* corresponds to terminal address 0. The most significant bit corresponds to terminal address 31.

WARNINGS: L13_MonFilter?? functions only operate on Lx1553-3 versions B and above.

SEE ALSO: L13_MonFilterSA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonIsRunning..... Determines whether the monitor is recording messages.

SYNOPSIS: BOOL L13_MonIsRunning(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: TRUE if the monitor is still active, otherwise FALSE.

DESCRIPTION: This function is typically used when the monitor has been configured with the MONCFG_FILLHALT flag. In this case, the monitor disables itself after filling its internal buffer. Thus, L13_MonIsRunning effectively indicates whether the buffer is full.

WARNINGS: None.

SEE ALSO: L13_MonConfig

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonRd ..... Reads the next message record from the monitor list.

SYNOPSIS: USHORT L13_MonRd(buf, hCard)
LPWORD buf Buffer to receive message record
HCARD hCard Handle of Lx1553-3 card

RETURNS: The number of (16-bit) words read from the monitor buffer, or zero if there was no new record to read.

DESCRIPTION: Copies the next entry from the monitor buffer on the Lx1553-3 to the user-supplied *buf*. The record lengths are variable, so the *buf* must point to an array large enough to store MONRD_MAX_COUNT words.

Contents of buffer[] after L13_MonRd	
Word	**Description**
0 (lo-byte)	Total length (in 16-bit words) of the message record
0 (hi-byte)	Lx1553-3 version code
1	Least significant 16 bits of 32-bit time-tag
2	Most significant 16 bits of 32-bit time-tag
3	Activity (see below)
4	Reserved
5	Command word 1
6	Command word 2 (for RT-RT transfers only)
7	Status word 1
8	Status word 2 (for RT-RT transfers only)
9 to (length-1)	Data words

Bits in the Activity word	
Bit	**Description**
0	Bus A or B (0 indicates A, 1 indicates B)
4	Parity error
5	Manchester error
6	Data words were received
8	Status word 2 is valid
9	Status word 1 is valid
10	Command word 2 is valid
11	Command word 1 is valid
14	An error was detected
15	The RT did not respond

WARNINGS: None.

SEE ALSO: L13_MonFilterSA, L13_MonFilterTA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonResume..... Starts the Lx1553-3 monitor without resetting.

SYNOPSIS: BOOL L13_MonResume(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: FALSE if an error occurs, or TRUE if successful.

DESCRIPTION: Starts the Lx1553-3 monitor without resetting the buffer pointers. Monitor operation resumes at the point where L13_MonStop was called.

WARNINGS: None.

SEE ALSO: L13_MonConfig, L13_MonStop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonStart ..... Starts the Lx1553-3 monitor.

SYNOPSIS: BOOL L13_MonStart(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: FALSE if an error occurs, or TRUE if successful.

DESCRIPTION: Starts the Lx1553-3 monitor independently of other terminals.

WARNINGS: Calls to L13_MonConfig and L13_CardStart must precede this function. This function is only intended for starting the monitor of a multi-terminal card independent of other BC or RT configurations.

SEE ALSO: L13_MonConfig, L13_MonStop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MonStop .....Stops the Lx1553-3 monitor independent of other terminals.

SYNOPSIS: BOOL L13_MonStop(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: FALSE if an error occurs, or TRUE if successful.

DESCRIPTION: Stops the Lx1553-3 monitor independent of other terminals. Messages are no longer added to the sequential record while the monitor is stopped.

WARNINGS: None.

SEE ALSO: L13_MonConfig, L13_MonStart

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgBlockRd ..... Reads an entire message structure from the Lx1553-3.

SYNOPSIS: MSGADDR L13_MsgBlockRd(msgbuf, message, hCard)

LPUSHORT msgbuf	Pointer to array of 64 16-bit words
MSGADDR message	Message from which to read
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The address of the message structure that was read.

DESCRIPTION: Reads an entire message structure from the Lx1553-3.

Structure for msgbuf	
Word	**Value**
0-1	Reserved
2	Error field - Errors associated with this message (see below)
3	Activity field - Bits denoting the bus activity associated with this message (see below)
4	Response time 1 + 3.0 us (in units of 0.1 us)
5	Response time 2 + 3.0 us (in units of 0.1 us)
6	Data count field - Number of valid data words associated with this message
7	Reserved
8 - 9	Time-tag field - Value of time-tag
10 - 11	Elapse time field - Value of elapse time
12 - 19	Reserved
20	Command Word 1
21	Command Word 2 (RT-RT only)
22	Status Word 1
23	Status Word 2 (RT-RT only)
24 - 55	Data Words
56 - 63	Additional data when generating a word count error

The error and activity fields always contain the result of the last transmission or reception of the message. These fields may be tested by ANDing the values returned in word 2 or word 3 of the message structure with constants from the following tables.

Error field	
Constant	**Description**
MSGERR_NORESP	No response was received form the RT (BC only)
MSGERR_ANYERR	Set if any other error bits are set
MSGERR_PROTOCOL	A protocol error occurred
MSGERR_SYNC	Wrong polarity sync pulse
MSGERR_DATACOUNT	Too many/too few data words
MSGERR_MANCH	Manchester error
MSGERR_PARITY	Parity error
MSGERR_WORD	Reserved
MSGERR_RETRY	All attempts to retry transmission of this message failed (BC only)
MSGERR_SYSTEM	Internal Lx1553-3 error
MSGERR_HIT	Indicates that this message was transmitted or received since this bit was last cleared

Continued on following page...

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Activity field	
Constant	**Description**
MSGACT_XMTCWD1	The first command word was transmitted.
MSGACT_ XMTCWD2	The second command word was transmitted.
MSGACT_ XMTSWD1	The first status word was transmitted.
MSGACT_ XMTSWD2	The second status word was transmitted.
MSGACT_ RCVCWD1	The first command word was received.
MSGACT_ RCVCWD2	The second command word was received.
MSGACT_ RCVSWD1	The first status word was received.
MSGACT_ RCVSWD2	The second status word was received.
MSGACT_ XMTDWD	Data words were transmitted.
MSGACT_ RCVDWD	Data words were received.

WARNINGS: None.

SEE ALSO: L13_MsgDataRd

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgBlockWr..... Writes an entire message structure on the Lx1553-3.

SYNOPSIS: MSGADDR L13_MsgBlockWr(msgbuf, message, hCard)

LPUSHORT msgbuf	Pointer to array of 64 16-bit words
MSGADDR message	Message to receive new data
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The address of the message structure that was written.

DESCRIPTION: Writes an entire message structure on the Lx1553-3.

Structure for msgbuf	
Word	**Value**
0-1	Reserved
2	Error field - Errors associated with this message (see below)
3	Activity field - Bits denoting the bus activity associated with this message (see below)
4	Response time 1 + 3.0 us (in units of 0.1 us)
5	Response time 2 + 3.0 us (in units of 0.1 us)
6	Data count field - Number of valid data words associated with this message
7	Reserved
8 - 9	Time-tag field - Value of time-tag
10 - 11	Elapse time field - Value of elapse time
12 - 19	Reserved
20	Command Word 1
21	Command Word 2 (RT-RT only)
22	Status Word 1
23	Status Word 2 (RT-RT only)
24 - 55	Data Words
56 - 63	Additional data when generating a word count error

The error and activity fields always contain the result of the last transmission or reception of the message. These fields may be tested by ANDing the values returned in word 2 or word 3 of the message structure with constants from the following tables.

Error field	
Constant	**Description**
MSGERR_NORESP	No response was received form the RT (BC only)
MSGERR_ANYERR	Set if any other error bits are set
MSGERR_PROTOCOL	A protocol error occurred
MSGERR_SYNC	Wrong polarity sync pulse
MSGERR_DATACOUNT	Too many/too few data words
MSGERR_MANCH	Manchester error
MSGERR_PARITY	Parity error
MSGERR_WORD	Reserved
MSGERR_RETRY	All attempts to retry transmission of this message failed (BC only)
MSGERR_SYSTEM	Internal Lx1553-3 error
MSGERR_HIT	Indicates that this message was transmitted or received since this bit was last cleared

Continued on following page...

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Activity field	
Constant	**Description**
MSGACT_XMTCWD1	The first command word was transmitted.
MSGACT_ XMTCWD2	The second command word was transmitted.
MSGACT_ XMTSWD1	The first status word was transmitted.
MSGACT_ XMTSWD2	The second status word was transmitted.
MSGACT_ RCVCWD1	The first command word was received.
MSGACT_ RCVCWD2	The second command word was received.
MSGACT_ RCVSWD1	The first status word was received.
MSGACT_ RCVSWD2	The second status word was received.
MSGACT_ XMTDWD	Data words were transmitted.
MSGACT_ RCVDWD	Data words were received.

WARNINGS: None.

SEE ALSO: L13_MsgDataWr

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgCommRd.....Reads a message structure guaranteeing data integrity.

SYNOPSIS: MSGADDR L13_MsgCommRd(msgbuf, message, hCard)

LPUSHORT msgbuf	Pointer to array of 64 16-bit words
MSGADDR message	Message from which to read
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The address of the message structure that was read.

DESCRIPTION: Reads an entire message structure guaranteeing data integrity by using the communications buffer on the Lx1553-3. This function is slower than L13_MsgBlockRd.

Structure for msgbuf	
Word	**Value**
0-1	Reserved
2	Error field - Errors associated with this message (see below)
3	Activity field - Bits denoting the bus activity associated with this message (see below)
4	Response time 1 + 3.0 us (in units of 0.1 us)
5	Response time 2 + 3.0 us (in units of 0.1 us)
6	Data count field - Number of valid data words associated with this message
7	Reserved
8 - 9	Time-tag field - Value of time-tag
10 - 11	Elapse time field - Value of elapse time
12 - 19	Reserved
20	Command Word 1
21	Command Word 2 (RT-RT only)
22	Status Word 1
23	Status Word 2 (RT-RT only)
24 - 55	Data Words
56 - 63	Additional data when generating a word count error

The error and activity fields always contain the result of the last transmission or reception of the message. These fields may be tested by ANDing the values returned in word 2 or word 3 of the message structure with constants from the following tables.

Error field	
Constant	**Description**
MSGERR_NORESP	No response was received form the RT (BC only)
MSGERR_ANYERR	Set if any other error bits are set
MSGERR_PROTOCOL	A protocol error occurred
MSGERR_SYNC	Wrong polarity sync pulse
MSGERR_DATACOUNT	Too many/too few data words
MSGERR_MANCH	Manchester error
MSGERR_PARITY	Parity error
MSGERR_WORD	Reserved
MSGERR_RETRY	All attempts to retry transmission of this message failed (BC only)
MSGERR_SYSTEM	Internal Lx1553-3 error
MSGERR_HIT	Indicates that this message was transmitted or received since this bit was last cleared

Continued on following page...

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Activity field	
Constant	**Description**
MSGACT_XMTCWD1	The first command word was transmitted.
MSGACT_ XMTCWD2	The second command word was transmitted.
MSGACT_ XMTSWD1	The first status word was transmitted.
MSGACT_ XMTSWD2	The second status word was transmitted.
MSGACT_ RCVCWD1	The first command word was received.
MSGACT_ RCVCWD2	The second command word was received.
MSGACT_ RCVSWD1	The first status word was received.
MSGACT_ RCVSWD2	The second status word was received.
MSGACT_ XMTDWD	Data words were transmitted.
MSGACT_ RCVDWD	Data words were received.

WARNINGS: None.

SEE ALSO: L13_MsgDataRd, L13_MsgBlockRd

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgCommWr..... Writes a message structure guaranteeing data integrity.

SYNOPSIS: MSGADDR L13_MsgCommWr(msgbuf, message, hCard)

LPUSHORT msgbuf	Pointer to array of 64 16-bit words
MSGADDR message	Message to receive new data
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The address of the message structure that was written.

DESCRIPTION: Writes an entire message structure guaranteeing data integrity by using the communications buffer on the Lx1553-3. This function is slower than L13_MsgBlockWr.

Structure for msgbuf	
Word	**Value**
0-1	Reserved
2	Error field - Errors associated with this message (see below)
3	Activity field - Bits denoting the bus activity associated with this message (see below)
4	Response time 1 + 3.0 us (in units of 0.1 us)
5	Response time 2 + 3.0 us (in units of 0.1 us)
6	Data count field - Number of valid data words associated with this message
7	Reserved
8 - 9	Time-tag field - Value of time-tag
10 - 11	Elapse time field - Value of elapse time
12 - 19	Reserved
20	Command Word 1
21	Command Word 2 (RT-RT only)
22	Status Word 1
23	Status Word 2 (RT-RT only)
24 - 55	Data Words
56 - 63	Additional data when generating a word count error

The error and activity fields always contain the result of the last transmission or reception of the message. These fields may be tested by ANDing the values returned in word 2 or word 3 of the message structure with constants from the following tables.

Error field	
Constant	**Description**
MSGERR_NORESP	No response was received form the RT (BC only)
MSGERR_ANYERR	Set if any other error bits are set
MSGERR_PROTOCOL	A protocol error occurred
MSGERR_SYNC	Wrong polarity sync pulse
MSGERR_DATACOUNT	Too many/too few data words
MSGERR_MANCH	Manchester error
MSGERR_PARITY	Parity error
MSGERR_WORD	Reserved
MSGERR_RETRY	All attempts to retry transmission of this message failed (BC only)
MSGERR_SYSTEM	Internal Lx1553-3 error
MSGERR_HIT	Indicates that this message was transmitted or received since this bit was last cleared

Continued on following page...

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Activity field	
Constant	**Description**
MSGACT_XMTCWD1	The first command word was transmitted.
MSGACT_ XMTCWD2	The second command word was transmitted.
MSGACT_ XMTSWD1	The first status word was transmitted.
MSGACT_ XMTSWD2	The second status word was transmitted.
MSGACT_ RCVCWD1	The first command word was received.
MSGACT_ RCVCWD2	The second command word was received.
MSGACT_ RCVSWD1	The first status word was received.
MSGACT_ RCVSWD2	The second status word was received.
MSGACT_ XMTDWD	Data words were transmitted.
MSGACT_ RCVDWD	Data words were received.

WARNINGS: None.

SEE ALSO: L13_MsgDataWr, L13_MsgBlockWr

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgDataRd.....Reads the data associated with a message.

SYNOPSIS: VOID L13_MsgDataRd(buf, count, message, hCard)

LPWORD buf	Pointer to data words
USHORT count	Number of words to read
MSGADDR message	Message from which to read
HCARD hCard	Handle of Lx1553-3 card

RETURNS: None.

DESCRIPTION: Copies *count* data words to the *buf* from the message structure on the Lx1553-3 specified by *message*.

WARNINGS: None.

SEE ALSO: L13_MsgDataWr, L13_BCCreateMsg, L13_RTCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgDataWr.....Writes the data associated with a message.

SYNOPSIS: VOID L13_MsgDataWr(buf, count, message, hCard)

LPWORD buf	Pointer to data words
USHORT count	Number of words to write
MSGADDR message	Message to receive new data
HCARD hCard	Handle of Lx1553-3 card

RETURNS: None.

DESCRIPTION: Copies *count* data words from the *buf* to the message structure on the Lx1553-3 specified by *message*.

WARNINGS: None.

SEE ALSO: L13_MsgDataRd, L13_BCCreateMsg, L13_RTCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgFieldRd..... Reads a field from a message structure.

SYNOPSIS: ULONG L13_MsgFieldRd(ctrlflags, message, hCard)

USHORT ctrlflags	Field to read
MSGADDR message	Message to read
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The value of the specified field in the specified message structure.

DESCRIPTION: Reads the value of the field specified by *ctrlflags* from the message structure at address *message*. It is typically used to read the status words and error fields in a BC message structure after message transmission.

ctrlflags	
Constant	**Description**
FIELD_CWD1	Command word 1 field
FIELD_CWD2	Command word 2 field (RT-RT only)
FIELD_SWD1	Status word 1 field
FIELD_SWD2	Status word 2 field (RT-RT only)
FIELD_TTAG	Time-tag field
FIELD_ELAPSE	Elapsed time field
FIELD_ERROR	Error field (see below)
FIELD_ACT	Activity field (see below)
FIELD_RESP1	Response time 1 field
FIELD_RESP2	Response time 2 field (RT-RT only)
FIELD_COUNT	Data count field
FIELD_FLAG1	Flags field (see below)
FIELD_FLAG2	Flags field (see below)

All of the fields are simple values except for _ERROR, _ACT, _FLAG1 and _FLAG2, which are bit fields. See the header file for #defined constants which correspond to these bits. The constants for the _ERROR and _ACT fields are shown on the following page.

The _FLAGn fields contain bits that correspond to L13_??CreateMsg options. Other than bit 0 of FIELD_FLAG1, these rarely need to be explicitly read and should not be explicitly written.

FIELD_FLAG1	
Bit	**Description**
0	In BC message structures, indicates on which bus (A/B) to transmit. In RT message structures, indicates on which bus command was received.
1–15	Correspond to L13_??CreateMsg options. See the header file for #defined constants for each bit.

FIELD_FLAG2	
Bit	**Description**
0–15	Correspond to L13_??CreateMsg options. See the header file for #defined constants for each bit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The error and activity fields always contain the results of the last transmission or reception of the message. These fields may be tested by AND-ing the values returned by L13_MsgFieldRd with constants from the following tables.

The _ERROR field indicates errors that were detected during the last transmission or reception of the message.

FIELD_ERROR	
Constant	**Description**
MSGERR_HIT	Indicates that this message was transmitted or received since this bit was last cleared
MSGERR_SYSTEM	Internal Lx1553-3 error
MSGERR_RETRY	All attempts to retry transmission of this message failed (BC only)
MSGERR_WORD	Reserved
MSGERR_PARITY	Parity error
MSGERR_MANCH	Manchester error
MSGERR_DATACOUNT	Too many/too few data words
MSGERR_SYNC	Wrong polarity sync pulse
MSGERR_PROTOCOL	A protocol error occurred
MSGERR_ANYERR	Set if any other error bits are set
MSGERR_NORESP	No response was received from the RT (BC only)

The Lx1553-3 always sets bit 0 (MSGERR_HIT) in the _ERROR field after transmission or reception of the message. The user should explicitly clear this bit to detect subsequent transmissions/receptions of the message. See Example 2 in Section 4.2.

FIELD_ACT	
Constant	**Description**
MSGACT_RCVDWD	Data words were received.
MSGACT_XMTDWD	Data words were transmitted.
MSGACT_RCVSWD2	The second status word was received.
MSGACT_RCVSWD1	The first status word was received.
MSGACT_RCVCWD2	The second command word was received.
MSGACT_RCVCWD1	The first command word was received.
MSGACT_XMTSWD2	The second status word was transmitted.
MSGACT_XMTSWD1	The first status word was transmitted.
MSGACT_XMTCWD2	The second command word was transmitted.
MSGACT_XMTCWD1	The first command word was transmitted.

WARNINGS: The _ERROR field indicates *detected* errors. It does NOT relate to error *injection* on Lx1553-3 version C cards.

SEE ALSO: L13_MsgFieldWr , L13_BCCreateMsg, L13_RTCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MsgFieldWr ..... Writes a field to a message structure.

SYNOPSIS: ULONG L13_MsgFieldWr(fieldvalue, ctrlflags, message, hCard)

ULONG ctrlflags	Field to rewrite
ULONG fieldvalue	New field value
MSGADDR message	Message to rewrite
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The old value of the field.

DESCRIPTION: Writes *fieldvalue* to the specified field of the specified message structure. This function can be used to reconfigure an existing message.

Although all fields are writable, this function is typically only used to clear the "Hit" bit in the _ERROR field. This bit indicates that the message has been transmitted or received. See Example 2 in Section 4.2.

See L13_MsgFieldRd for tables describing the fields.

WARNINGS: None.

SEE ALSO: L13_MsgFieldRd , L13_BCCreateMsg, L13_RTCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RTConfig ..... Configures the card for remote terminal mode.

SYNOPSIS: ERRVAL L13_RTConfig(ctrlflags, taval, hCard)

ULONG ctrlflags	Selects configuration options
USHORT taval	Terminal address
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, otherwise zero.

DESCRIPTION: Configures the specified Lx1553-3 to simulate a remote terminal with the indicated terminal address. On Lx1553-3 versions B and higher, the function can be called repeatedly with different terminal addresses to configure the Lx1553-3 to simulate multiple terminals.

By default, L13_RTConfig creates message structures for all subaddresses and mode codes. These message structures are allocated using the default L13_RTCreateMsg options. The addresses of these structures are obtained as needed by calling L13_RTGetMsg.

If L13_RTConfig is called with the RTCFG_NOBUILD flag, all command words are illegalized until message structures are created with L13_RTCreateMsg.

ctrlflags	
Constant	**Description**
RTCFG_DEFAULT	Select all default settings (**bold** below)
RTCFG_DISABLE	Disable RT
RTCFG_SIMULATE	Enable RT simulation
RTCFG_MONITOR	Enable RT monitor
RTCFG_MCNONE	Disable mode codes
RTCFG_MC0	Select SA=00000 for mode codes
RTCFG_MC1	Select SA=11111 for mode codes
RTCFG_MC01	00000 & 11111 indicates mode codes
RTCFG_NOBCAST	Disable broadcast
RTCFG_BCAST	Enable broadcast receptions
RTCFG_NOAUTOBUSY	Disable auto busy
RTCFG_AUTOBUSY	Set Busy bit after the next command
RTCFG_BUILD	All SAs and MCs initially enabled
RTCFG_NOBUILD	All SAs and MCs initially disabled
RTCFG_NODYNBC	Ignore dynamic BC mode code
RTCFG_DYNBC	Respond to dynamic BC mode code
RTCFG_NOIMMCLR	Do not clear status word bits
RTCFG_IMMCLR	Clear status word bits after transmitted
RTCFG_WIPE0	Zero the message structures on creation
RTCFG_NOWIPE	Do not zero the message on creation
RTCFG_WIPE123	Initialize the data words with incrementing values
RTCFG_WIPECWD	Initialize the data words with command word values
RTCFG_MODEL	Messages are created in dual-port RAM on Lx1553-3 version A, otherwise in static RAM
RTCFG_RAM	Messages are created in static RAM
RTCFG_DPRAM	Messages are created in dual-port RAM

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

WARNINGS: None.

SEE ALSO: L13_RTGetMsg, L13_RTCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RTCreateList.....Creates an RT message list.

SYNOPSIS: LISTADDR L13_RTCreateList(listctrlflags, count, msgctrlflags, mcflag, taval, trval, saval, hCard)

ULONG listctrlflags	Selects list options
ULONG count	Number of entries in list
ULONG msgctrlflags	Selects configuration options.
BOOL mcflag	TRUE if mode code, FALSE if subaddress
INT taval	Terminal address containing message
BOOL trval	Transmit/receive bit
INT saval	Subaddress or mode-code number
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The Lx1553-3 address of the list if successful, otherwise zero.

DESCRIPTION: Similar to L13_RTCreateMsg except creates a list. Function allocates memory for a list of message structures. The flags SUBADDRESS or MODECODE may be used for the *mcflag* parameter. XMT or RCV may be used for the *trval* parameter.

listctrlflags	
Constant	**Description**
LISTCRT_DEFAULT	Select all default settings (**bold** below)
LISTCRT_PINGPONG	Selects ping-pong mode
LISTCRT_FIFO	Selects FIFO mode
LISTCRT_CIRCULAR	Selects circular mode
LISTCRT_TRBIT	The host will read or write data depending upon the T/R bit of *cwd1* if a BC-RT transfer, or read data if an RT-RT transfer
LISTCRT_READ	The host will read receive data from the list
LISTCRT_WRITE	The host will write transmit data to the list
LISTCRT_MODEL	List is created in static RAM
LISTCRT_RAM	List is created in static RAM
LISTCRT_DPRAM	List is created in dual-port RAM

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Msgctrlflags	
Constant	**Description**
MSGCRT_DEFAULT	Select all default settings (**bold** below)
MSGCRT_ENABLE	Enable subaddress (RT only)
MSGCRT_DISABLE	Disable subaddress (RT only)
MSGCRT_NOWRAP	Disable data wrap (RT only)
MSGCRT_WRAP	Enable data wrap (RT only)
MSGCRT_NOINT	Message not will generate interrupt
MSGCRT_INT	Message will generate interrupt
MSGCRT_ERR	This message is a candidate for errors
MSGCRT_NOERR	This message is not a candidate for errors
MSGCRT_NOTIMETAG	Message will not record time-tag
MSGCRT_TIMETAG	Message will record time-tag
MSGCRT_NOELAPSE	Message will not record elapsed time
MSGCRT_ELAPSE	Message will record elapsed time
MSGCRT_NOHIT	Message will not record hit count
MSGCRT_HIT	Message will record hit count
MSGCRT_NOMON	Message will not be monitored
MSGCRT_MON	Message will be monitored
MSGCRT_NOMIN	Message will not record min time
MSGCRT_MIN	Message will record min time
MSGCRT_NOMAX	Message will not record max time
MSGCRT_MAX	Message will record max time
MSGCRT_NOSYNC	No sync will be generated for message
MSGCRT_SYNC	Sync will be generated for message
MSGCRT_WIPE	Zeros the message structure on creation
MSGCRT_NOWIPE	Does not zero the message on creation

WARNINGS: A call to L13_RTConfig must precede the use of this function for a given terminal address.

SEE ALSO: L13_BCCreateList, L13_ListDataRd, L13_ListDataWr, L13_RTCreateMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RTCreateMsg ..... Creates or reconfigures an RT message structure.

SYNOPSIS: MSGADDR L13_RTCreateMsg(ctrlflags, mcflag, taval, trval, saval, hCard)

ULONG ctrlflags	Selects configuration options.
BOOL mcflag	TRUE if mode code, FALSE if subaddress
INT taval	Terminal address containing message
BOOL trval	transmit/receive bit
INT saval	Subaddress or mode-code number
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The Lx1553-3 address of the message structure if successful, otherwise zero.

DESCRIPTION: If a message structure for the indicated command word in the specified terminal address does not already exist, this function creates it. If a message structure already exists for the specified parameters, L13_RTCreateMsg reconfigures it with the new *ctrlflags*. This function is usually only needed if the RTCFG_NOBUILD flag was used in L13_RTConfig. The flags SUBADDRESS or MODECODE may be used for the *mcflag* parameter. XMT or RCV may be used for the *trval* parameter.

Continued on following page...

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ctrlflags	
Constant	**Description**
MSGCRT_DEFAULT	Select all default settings (**bold** below)
MSGCRT_ENABLE	Enable subaddress (RT only)
MSGCRT_DISABLE	Disable subaddress (RT only)
MSGCRT_NOWRAP	Disable data wrap (RT only)
MSGCRT_WRAP	Enable data wrap (RT only)
MSGCRT_NOINT	Message not will generate interrupt
MSGCRT_INT	Message will generate interrupt
MSGCRT_ERR	This message is a candidate for errors
MSGCRT_NOERR	This message is not a candidate for errors
MSGCRT_NOTIMETAG	Message will not record time-tag
MSGCRT_TIMETAG	Message will record time-tag
MSGCRT_NOELAPSE	Message will not record elapsed time
MSGCRT_ELAPSE	Message will record elapsed time
MSGCRT_NOHIT	Message will not record hit count
MSGCRT_HIT	Message will record hit count
MSGCRT_NOMON	Message will not be monitored
MSGCRT_MON	Message will be monitored
MSGCRT_NOMIN	Message will not record min time
MSGCRT_MIN	Message will record min time
MSGCRT_NOMAX	Message will not record max time
MSGCRT_MAX	Message will record max time
MSGCRT_NOSYNC	No sync will be generated for message
MSGCRT_SYNC	Sync will be generated for message
MSGCRT_WIPE0	Zero the message structure on creation
MSGCRT_NOWIPE	Do not zero the message on creation
MSGCRT_WIPE123	Initialize the data words with incrementing values
MSGCRT_WIPECWD	Initialize the data words with command word values
MSGCRT_MODEL	Messages are created in dual-port RAM on Lx1553-3 version A, otherwise in static RAM
MSGCRT_RAM	Messages are created in static RAM
MSGCRT_DPRAM	Messages are created in dual-port RAM

WARNINGS: A call to L13_RTConfig must precede the use of this function for a given terminal address.

SEE ALSO: L13_RTCreateList, L13_BCCreateMsg , L13_RTConfig, L13_RTGetMsg

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RTGetMsg ..... Gets the address of an existing message structure.

SYNOPSIS: MSGADDR L13_RTGetMsg(mcflag, taval, trval, saval, hCard)

BOOL mcflag	TRUE if mode code, FALSE if subaddress
INT taval	Terminal address containing message
BOOL trval	transmit/receive bit
INT saval	Subaddress or mode-code number
HCARD hCard	Handle of Lx1553-3 card

RETURNS: The Lx1553-3 address of the message structure if successful, otherwise zero.

DESCRIPTION: This function obtains the message address for an existing message structure. It is typically used when L13_RTConfig is allowed to automatically create all possible message structures for the specified terminal address instead of explicitly creating them with L13_RTCreateMsg.

The #defined constants SUBADDRESS or MODECODE may be used for the *mcflag* parameter. The #defined constants XMT or RCV may be used for the *trval* parameter.

WARNINGS: None.

SEE ALSO: L13_RTCreateMsg, L13_RTConfig, L13_MsgFieldRd, L13_MsgFieldWr, L13_MsgDataRd, L13_MsgDataWr

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RTReset.....Performs a logical reset of the RT.

SYNOPSIS: ERRVAL L13_RTReset(taval, hCard)

INT taval	Terminal address
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Perform a logical reset of the RT with address *taval*. It is similar to receiving the Reset RT mode code. It clears all bits of the status word (except the terminal address) and enables both bus A and bus B transmitters. It disables override of the T/F bit of the status word.

WARNINGS: None.

SEE ALSO: L13_RTConfig

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RTSWDWr ..... Sets the status word for the RT.

SYNOPSIS: ERRVAL L13_RTSWDWr(swdval, taval, hCard)

USHORT swdval	MIL-STD-1553 status word
USHORT taval	Terminal address of simulated RT
HCARD hCard	Handle of Lx1553-3 card

RETURNS: Zero if successful, otherwise a negative value.

DESCRIPTION: Sets the status word with which the simulated RT specified by *taval* responds to any subsequent commands from the bus controller. This function completely determines the status word to be transmitted with one exception: the Lx1553-3 always automatically sets the Broadcast and Message Error bits of the status word according to 1553 protocol requirements. These bits can also be set in *swdval*, but they cannot be overridden (cleared) by *swdval*. The Broadcast and Message Error bits in *swdval* are bitwise OR-ed with internal values.

L13_RTConfig initializes the terminal address field of the terminal's status word, however L13_RTSWDWr overwrites it. Unless an invalid status word is desired, the user must make sure that the terminal address specified in *swdval* matches *taval* since this function overwrites the initialization performed by L13_RTConfig.

Setting the Busy bit affects Lx1553-3 command processing. The first command received after the Busy bit is set is processed normally, and the Busy bit is set in the status word. The Lx1553-3 subsequently refuses commands to receive or transmit data until this bit is explicitly cleared by the user.

Note: This function does not write to the status word field in any message structure. The status word field in a message structure contains the status word actually transmitted on the last command to that subaddress, not the status word that will be transmitted.

WARNINGS: Since this function overwrites a terminal's preinitialized status word it should be used with extra care to avoid unintended results.

The specified terminal address must have already been configured by L13_RTConfig.

SEE ALSO: L13_RTConfig

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TimerClear..... Clears the time-tag timer.

SYNOPSIS: VOID L13_TimerClear(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: None.

DESCRIPTION: Clears the time-tag timer to zero on the specified Lx1553-3 card.

WARNINGS: None.

SEE ALSO: L13_TimerRd, L13_TimerResolution

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TimerRd ..... Reads the time-tag timer.

SYNOPSIS: ULONG L13_TimerRd(hCard)
HCARD hCard Handle of Lx1553-3 card

RETURNS: The current time-tag timer value.

DESCRIPTION: Reads the current value of the time-tag timer on the specified Lx1553-3 card.

WARNINGS: None.

SEE ALSO: L13_TimerClear, L13_TimerResolution

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TimerResolution.....Selects the time-tag timer resolution.

SYNOPSIS: INT L13_TimerResolution(timerresol, hCard)

INT timerresol	Selects the timer resolution
HCARD hCard	Handle of Lx1553-3 card

RETURNS: A negative value if an error occurs, or zero if successful.

DESCRIPTION: Selects the resolution for the time-tag timer on the specified Lx1553-3. *timerresol* must be one of the following predefined constants.

Timerresol		
Constant	**Resolution**	**Range (h:m:s)**
TIMERRESOL_1US	1us	1:11:34
TIMERRESOL_16US	16us	19:05:19
TIMERRESOL_1024US	1024us	50 days

WARNINGS: After changing the resolution, a call to L13_TimerClear should be made to clear the timer.

SEE ALSO: L13_TimerClear

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ValFromAscii..... Converts a string to a numeric value.

SYNOPSIS: ULONG L13_ValFromAscii(asciistr, radixval)
LPCSTR asciistr ASCII string to convert
INT radixval Radix of string

RETURNS: The converted integer numeric value.

DESCRIPTION: Converts a string representation of a 16-bit value with the specified radix to an integer. Processing stops at the first null-terminator.

This is a utility function and does not access the Lx1553-3 hardware.

WARNINGS: No check is made for invalid characters.

SEE ALSO: L13_ValToAscii

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ValGetBits.....Extract the specified bit field from an integer value.

SYNOPSIS: USHORT L13_ValGetBits(oldvalue, startbit, endbit)

USHORT oldvalue	The old value
INT startbit	Position of starting bit of field
INT endbit	Position of ending bit of field

RETURNS: The value of the extracted bit field.

DESCRIPTION: Extracts the specified bit field from the 16-bit integer *oldvalue*. The result is obtained by masking the field and shifting the *endbit* to the LSB of the return value.

This is a utility function and does not access the Lx1553-3 hardware.

WARNINGS: None.

SEE ALSO: L13_ValPutBits

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ValPackCWD .... Creates a MIL-STD-1553 command word from its fields.

SYNOPSIS: USHORT L13_ValPackCWD(TAval, TRval, SAval, WCval)

INT TAval	Terminal address
INT TRval	T/R bit
INT SAval	Subaddress
INT WCval	Word count/mode code

RETURNS: A MIL-STD-1553 command word with its bit fields set to the specified values.

DESCRIPTION: Creates a 16-bit MIL-STD-1553 command word from the values of its four bit fields. Only the least significant bit of *TRval* and the 5 least significant bits of *TAval*, *SAval* and *WCval* are used.

This is a utility function that does not access Lx1553-3 hardware.

WARNINGS: None.

SEE ALSO: L13_ValUnpackCWD

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ValPutBits ..... Inserts a bit field into an integer value.

SYNOPSIS: USHORT L13_ValPutBits(oldval, newfld, startbit, endbit)

USHORT oldval	The old value
USHORT newfld	The value of the new field
INT startbit	Position of starting bit of field
INT endbit	Position of ending bit of field

RETURNS: The integer value with the inserted bit field.

DESCRIPTION: Inserts a bit field into an 16-bit integer value. The *oldval* is masked and ORed with the shifted value of *newfld*.

This is a utility function and does not access the Lx1553-3 hardware.

WARNINGS: None.

SEE ALSO: L13_ValGetBits

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ValToAscii ..... Creates a string representation of an integer.

SYNOPSIS: LPSTR L13_ValToAscii(value, asciistr, numbits, radixval)

ULONG value	The value to be converted
LPSTR asciistr	A string to receive the results.
INT numbits	The number of significant bits
INT radixval	The radix value

RETURNS: A string representing the integer.

DESCRIPTION: Creates a string representation of an integer in a specified radix. The string is copied to *asciistr* and is also returned. The string is always null-terminated. *asciistr* is assumed to be large enough to hold the resulting string.

The length of the string is determined by *numbits* and *radixval* and is padded by leading zeros. *radixval* can be any positive integer but is commonly 16 for hexadecimal, 8 for octal, or 10 for decimal. For example, a string representation of a value with 16 significant bits and a radix of 16 will always be 4 characters long followed by a null-terminator.

This is a utility function and does not access the Lx1553-3 hardware.

WARNINGS: None.

SEE ALSO: L13_ValFromAscii

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ValUnpackCWD ...Extracts fields from a MIL-STD-1553 command word.

SYNOPSIS: VOID L13_ValUnpackCWD(CWDval, TAval, TRval, SAval, WCval)

USHORT CWDval	MIL-STD-1553 command word
LPINT TAval	Pointer to terminal address
LPINT TRval	Pointer to T/R bit
LPINT SAval	Pointer to subaddress
LPINT WCval	Pointer to word count

RETURNS: None.

DESCRIPTION: Extracts the four bit fields from a MIL-STD-1553 command word and returns their (right-shifted) values in the pointed-to integers. If a pointer is NULL, that value is skipped.

This is a utility function that does not access Lx1553-3 hardware.

WARNINGS: None.

SEE ALSO: L13_ValPackCWD

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX B: D-SUB CONNECTORS

In place of the standard BJ77 twinax connectors, the Lx1553-3 may be ordered with a 25-pin female D-subminiature connector wired as shown in Fig. B-1.

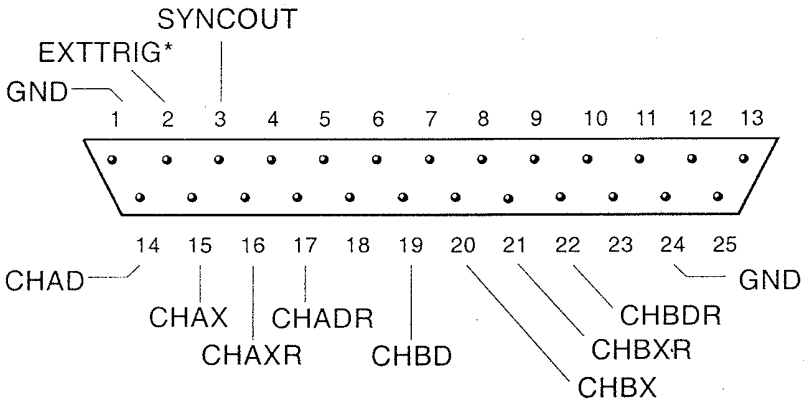


Fig. B-1 Pinouts of Lx1553-3 D-sub connector

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page intentionally blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX C: EXTERNAL SIGNALS

The Lx1553-3 provides several general purpose discrete I/O signals. Possible uses for these signals include:

- triggering BC message transmission
- triggering error injection
- transmitting a superseding command word on the second channel of a dual redundant bus (requires a second Lx1553-3)
- changing frame time resolution for bus controller message scheduling
- allowing on-board events to trigger external processes

If your Lx1553-3 has a D-sub connector, some external signals are available there (see Appendix B). Signals available on the I/O header P4 are shown in Fig. C-1. Contact Ballard Technology for more information on the use of these signals.

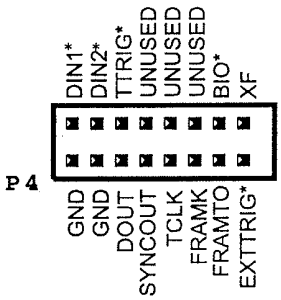


Fig. C-1 Pinouts of the external signals header

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page intentionally blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX D: SPECIFICATIONS

AVAILABLE MODELS

- Lx1553-3A
  Single terminal
- Lx1553-3B4
  Multiple terminals (up to 4)
- Lx1553-3B32
  Multiple terminals (up to 32)
- Lx1553-3C
  Multiple terminals (up to 32)
  Concurrent monitor
  Subaddress filtering
  Error injection
  1MB Static RAM

GENERAL

- Plug-and-Play PCI interface
- MIL-STD-1553 dual redundant
- Bus Controller (BC), Remote Terminal (RT), Monitor (MON)
- DSP Core—80 MHz
- Buffering: single, ping-pong, FIFO, or circular
- Interrupts or polling

BUS CONTROLLER MODE

- Message scheduling
- Executable architecture
- Automatic retry option
- Exception processing

REMOTE TERMINAL MODE

- MIL-STD-1553 A or B protocol
- Buffering by mode code or subaddress
- Selectable illegalization: by mode code or subaddress
- Simulator or passive shadow monitor

MONITOR TERMINAL MODE

- Time tagged sequential record
- Monitor all or selected terminals
- Configurable buffer size up to all available memory
- Continuous or fill and halt
- Detailed message information

INTERRUPTS

- Configurable interrupt log list
- Interrupt identification word

MISCELLANEOUS

- 256 KB On-Board Static RAM (1MB for C cards)
- 64KB True Dual-Port RAM
- Time tagging: 1us/16us/1024us resolution
- Digital input/output
- External trigger
- Sync output

LP1553-3 PHYSICAL

- PCI short card
- 6.9 × 4.2 in (17.5 x 10.7 cm)
- Weight: 6.0 oz (170 g)

LC1553-3 PHYSICAL

- CompactPCI interface card
- IEEE 1101.10 compliant
- Size:
  3U x 4HP (single slot, standard)
  6.3 × 3.94 in (16.0 x 10.0 cm)
  6U x 4HP (available)
- Weight: 6.6 oz (188 g)

POWER REQUIREMENTS

- Idle: 0.6A max at 5V
  0.05A max at 12V
- Transmitting:
  0.6A max at 5V
  0.25A max at 12V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ENVIRONMENTAL

- Temperature: 0 to 55 °C (ambient)
- Operating humidity: 5 to 95% (non-condensing)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX E: UPDATES

Revisions

The following revisions have been made to this manual.

Rev A:

Original release.

Rev B:

Modified text regarding CoPilot 1553 in section 1.2.

Added extended functions in Appendix A.

Added specifications to Appendix D.

Performed minor editing of manual without significant change to content.

Rev C:

Included LC1553-3 in LP1553-3 manual to make Lx1553-3 manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page intentionally blank.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP