

Ethernet Interface Option for the Models
LR-F2812DAQ and LR-F28335DAQ

Technical Reference Manual

July 1, 2008

Document Number: 4-903-40

Copyright © 2008
Link Research
All Rights Reserved

Table of Contents

1	GENERAL INFORMATION	3
2	PREPARING THE EZDSP BOARD (F28335 EZDSP ONLY)	4
3	USES OF THE ETHERNET INTERFACE	4
3.1	USING A TCP/IP CLIENT TO COMMUNICATE WITH THE EZDSP BOARD	4
3.2	WEB BROWSER COMMUNICATING WITH THE EZDSP USING JAVA APPLETS	4
3.3	EZDSP-TO-EZDSP COMMUNICATIONS OVER AN ETHERNET LAN	4
4	PC SOFTWARE INSTALLATION	5
5	CONFIGURING THE XPORT DEVICE	6
5.1	ASSIGNING AN IP ADDRESS AND PORT NUMBER TO THE XPORT	6
5.2	CHANGING THE XPORT'S CONFIGURATION	6
5.3	DOWNLOADING HTML AND JAVA FILES TO THE XPORT'S FLASH MEMORY	7
5.4	REPROGRAMMING THE XPORT DEVICE WITH THE SAMPLE PROGRAMS	7
5.5	DEVELOPING YOUR OWN JAVA APPLETS	7
6	XPORT DRIVER SOFTWARE	8
7	RUNNING THE DEMONSTRATION PROGRAMS	8
8	THE POSSIBILITIES	12
9	VERSION HISTORY	12

XPort is a trademark of Lantronix, Inc.

©2008, Link Research
ALL RIGHTS RESERVED

1 General Information

This technical reference covers the Ethernet Option (option #3) for the Model LR-F2812DAQ and LR-F28335DAQ daughtercards.

The Link Research Ethernet interface utilizes the Lantronix, Inc. XPort™ Device Server module. This module contains the MAC and PHY portions of the Ethernet Interface, a full TCP/IP stack, as well as the RJ45 jack itself. It is IEEE 802.3 compatible. The Lantronix XPort has the following features:

- Simple serial interface to the host processor (the DSP on the eZdsp board), supporting data rates up to 921,600 bps.
- Flash memory to hold HTML files and JAVA applet files.
- User configurable IP address and Port number.
- Email alert system.
- Full TCP/IP stack.

Complete documentation for the Lantronix XPort can be obtained from the Lantronix, Inc. website located at: www.lantronix.com.

The Ethernet interface is available as an option for both the F2812 eZdsp and the F28335 eZdsp. For the F2812, the DSP's serial communications port SCI-B is used to access the XPort. For the F28335, the DSP's serial communications port SCI-C is used. On the F28335 eZdsp, two jumper wires must be installed on the eZdsp board for the Ethernet option to function (see next section).

Two demonstration programs are included on CD with the purchase of the Ethernet option. These programs demonstrate a PC communicating with the eZdsp over the Ethernet. The first demonstration program is fairly simple; it displays the state of the eight digital inputs in a browser window. The second program is more complicated. It displays a graphical image of an oscilloscope within a browser window. The user can control the levels of three of the DAC outputs using sliders, and can view the three DAC signals on the scope.

Each demo program consists of two parts: (1) firmware for the eZdsp, and (2) Java applets and html files which can be stored in the XPort and executed from any browser. As shipped, the XPort has been pre-programmed with these demo applications. See section 8 for more detailed information on using these demo programs.

2 Preparing the eZdsp board (F28335 eZdsp only)

Two jumper wires need to be installed on the F28335 eZdsp board in order for the Ethernet interface to operate. Two signals from the P10 connector must be jumpered to the P2 connector. Specifically:

Connect P10-3 to P2-59
Connect P10-4 to P2-60

Any small insulated wire will work fine for these jumpers. This mod **does not** need to be done on the F2812 eZdsp board.

3 Uses of the Ethernet Interface

The following sections illustrate several interesting ways in which the Ethernet interface can be used.

3.1 Using a TCP/IP Client to communicate with the eZdsp board

In this configuration, A TCP/IP capable client application, such as MATLAB, HyperTerminal, or a user developed TCP/IP capable client application can communicate with an eZdsp physically located anywhere on the LAN or internet. Software on the eZdsp need only be concerned with UART initialization, and receiving and transmitting data to the UART. On the client side, the TCP/IP host application must create (open) a socket to the IP address and port number previously programmed into the XPort . (See section 5.1). Once a valid socket has been established, the host application can use the socket APIs to send and receive data to and from the eZdsp.

3.2 Web browser communicating with the eZdsp using JAVA applets

For this configuration, no special client-side application is required. Instead, any standard web browser with a Java plug-in may be used to communicate with an eZdsp attached to the LAN. In this case, a standard browser is used to open the socket to the XPort. User written JAVA applets stored on the XPort are then accessed by the web browser, and communications proceeds between the applet and the eZdsp. The XPort contains FLASH memory capable of holding HTML as well as JAVA applet files. Section 5.3 shows the method used to download files to the XPort .

3.3 eZdsp-to-eZdsp communications over an Ethernet LAN

This configuration requires absolutely no TCP/IP programming whatsoever. However, there is slightly more effort involved in programming the 2 (or more) XPort devices. For this setup, one XPort must be configured to be a master, while the other must be configured as a slave. The XPort user's manual available from www.lantronix.com will be helpful in configuring the XPort for this configuration.

4 PC Software Installation

Insert the supplied CD into the CD ROM drive. The installation should start automatically within a few seconds. If this does not happen, then manually start the installation by running SETUP.EXE, located in the root directory of the CD. It is recommended that the files be installed to the "my projects" directory of the TI CCS installation. Once the installation is complete, the CD can be safely removed.

A simple graphical user interface software package is provided on the installation CD to facilitate XPort configuration. This program is called Device Installer, and is made available by Lantronix, Inc. The user is encouraged to visit the Lantronix web site to get the latest version of this useful software program.

Device Installer requires the Microsoft .NET 2.0 foundation software. If this software is not already installed on the system, it must be installed before Device Installer is installed. The Microsoft .NET software can be installed from the supplied CD by following the following steps:

- 1) In the installation directory, locate the **Microsoft** directory in the "Microsoft and Java files" subdirectory.
- 2) Run the file **dotnetfx.exe**.
- 3) Follow the instructions presented during the installation.

Once .NET has been successfully installed, DeviceInstaller can be installed using the following steps:

- 1) In the installation directory, locate the **XPort Device Installer** directory in the "Microsoft and Java files" subdirectory.
- 2) Run the program **DI_4.2.0.0_Web.exe**.
- 3) Follow the instructions presented during installation.

5 Configuring the XPort Device

5.1 Assigning an IP address and Port number to the XPort

There are several ways to assign an IP address and port number to the XPort device. However, the easiest way is to use the DeviceInstaller program and follow these simple steps:

- 1) With a powered-up XPort interface attached to the LAN, run DeviceInstaller on a PC.
- 2) The XPort device should automatically appear in the right frame after a few seconds. If not, click the SEARCH button to locate the XPort device. Don't worry about any error messages that appear at this point.
- 3) Select (i.e., click on) the XPort device in the right frame to select it.
- 4) Click on **ASSIGN IP** in the button bar at the top.
- 5) In the window that opens, select "Assign a specific IP address", then click NEXT.
- 6) In the next window that appears, enter the desired IP address in the standard dot notation (for example: 192.168.0.25). It must be a valid IP address within the network's subnet. The remaining two fields can be left as is. Now click NEXT
- 7) On the final window that appears, click the ASSIGN button.

Note: The default port number is 10001. Normally it is not necessary to change this value.

5.2 Changing the XPort's Configuration

Once the IP of the XPort is set, it's configuration screen can be opened in any web browser by entering the IP address of the XPort device into the URL field of the browser as follows:

http://<IP of XPort>

where <IP of XPort> is the actual IP address of your XPort device.

On this screen, all parameters of the XPort device can be set, although the most important setting is the data rate. This data rate **must** be the same as the data rate used in the **Initialize_Ethernet_Port ()** function of the DSP code.

After changing any of the settings, remember to click on UPDATE SETTINGS, to permanently store the new settings in the XPort's flash.

Note: As shipped, the XPort will have its serial port parameters set to 921.6 kbps, 2 stop bits, and no parity.

5.3 Downloading HTML and JAVA files to the Xport's FLASH memory

The XPort contains FLASH memory that can be used to store user defined web pages (.html files) and JAVA applets (.class files). It is these Java applets that allow an information channel to be created between the eZdsp and the PC.

Once the .html and .class files are created, DeviceInstaller can be used to download the file into the XPort 's FLASH memory. To do this follow these steps:

- 1) Run DeviceInstaller and locate the XPort device in the right frame.
- 2) Select (i.e. click on) the device to highlight it.
- 3) Click on the UPGRADE button.
- 4) In the first window that opens, select "Create a custom installation by specifying individual files". Click NEXT.
- 5) On the next window, simply click NEXT.
- 6) On the third window, select "Install files individually". Click NEXT.
- 7) On the fourth window, click ADD FILES. Select all the .html and .class files you wish to program into the XPort. Click NEXT.
- 8) The next window shows the files you selected, and the partition(s) they will be written to in the flash. Click NEXT, then click NEXT again to begin programming.
- 9) When this process is finished, click CLOSE.

5.4 Reprogramming the XPort Device with the sample programs

As stated previously, the XPort device is shipped from the Link Research facility with the two demo applications pre-programmed into its flash memory. If the user wishes to re-program the device to factory conditions, simply follow the steps in section 5.3 using all of the files found in the XPORT_FILES directory under the "Microsoft and Java files" subdirectory.

5.5 Developing your own Java applets

Development tools for creating your own Java applets are available from the Sun Microsystems, Inc. website. For insight into how the demo programs were built, locate the batch file located in either of the demo project directories on the included CD. Notice

that the Java compiler is called JAVAC, and compiles .java source files into .class object files. These .class files can be directly downloaded into the XPort device.

The two included sample Java projects are located in the “Microsoft and Java files” directory. One project is located in the “Scope_demo” subdirectory, and the other is located in the “simple_java_demo” subdirectory. To build each project, simply run the .bat batch file located in each directory. Of course, the free Java development system must already be installed on your system. Notice that, after the .java files are compiled, the batch program copies all the .class files and other relevant files to the XPORT_FILES subdirectory. It is these files that must be programmed into the XPort device’s flash memory using the procedure outlined in section 5.3

6 XPort Driver Software

As previously stated, the XPort is connected to the DSP using a serial communications port. For the F2812, this port is SCI-B and for the F28335, the port is SCI-C. As such, a serial port API library is included on the CD to allow easy access to the serial port, and consequently, to the XPort. The API consists of 8 high level functions. These functions are defined in table 1. This software handles all of the low level hardware access functions, including initializing the serial port, configuring the port for a particular data rate, etc. The software also manages transmit and receive circular buffers, whose length is user configurable.

To use the functions in a user application, locate one of the serial port source files, for example, scope_demo_serial_port.c in the Scope Demo project on the included CD. Add this file to your project, renaming it as you wish. The file scope_demo.h will also be needed.

7 Running The Demonstration Programs

The CD which ships with the Link Research Ethernet interface contains two application programs which demonstrate how a web browser can be used to monitor and control eZdsp operation. These applications consist of complete Code Composer Studio projects, as well as HTML and JAVA files that can be downloaded into the XPort. The program demonstrates communication between the browser JAVA applet and the eZdsp. This technique can be extended to allow the eZdsp to communicate with any TCP/IP client application, for example, MATLAB, or a TCP/IP client application of the user’s creation.

The first demonstration program supplied with the Ethernet interface allows a browser to monitor the 8 digital inputs present on the LR-F2812DAQ or LR-F28335DAQ daughtercard. The program running on the eZdsp monitors the 8

inputs, and sends a single byte over the Ethernet to the Java applet running on the PC. For each input that is grounded, a corresponding white square is shown in the browser window.

To run this demo, perform the following steps:

- 1) Open CCS and load the project: **simple_java_demo** from the installation directory.
- 2) Build the project and download the resulting **.out** file into the eZdsp board.
- 3) Run the program.
- 4) Open a browser window and type in the following URL: “http://<xport ip>/simple_java_demo.html”, where <xport ip> should be replaced with your XPort’s actual IP address.
- 5) After a few seconds, a blue background with 8 rectangles will be displayed in the browser window.
- 6) Now, whenever any of the 8 digital inputs on the daughtercard is grounded, the corresponding rectangle will turn white.

The second demonstration program is somewhat more complex. It graphically displays an oscilloscope with a simple trigger setup. It also has 3 slider controls used to control the output amplitude of digital-to-analog (D/A) converter channels 1 thru 3.

The oscilloscope displays the signals of the first 3 D/A channels, so the user can see the amplitudes of these three signals change as he/she moves the sliders. This demo program could be the beginning of a 3-phase power monitoring system.

To run this demo, perform the following steps:

- 1) Make sure that the 60,000 sample sinewave is stored in the DSP’s FLASH according to the procedure outlined in the technical reference manual for the model LR-F2812DAQ or LR-F28335DAQ daughtercard.
- 2) Open CCS and load the project: **scope_demo** from the installation directory.

- 3) Build the project and download the resulting **.out** file into the eZdsp board.
- 4) Run the program.
- 5) Open a browser window and type in the following URL:
"http://<xport ip>/scope_demo.html", where <xport ip> should be replaced with your XPort's actual IP address.
- 6) After a few seconds, the GUI shown in Figure 1 will be displayed.
- 7) Now, any of the 3 sliders on the right can be changed to see the change in the corresponding signal's amplitude.

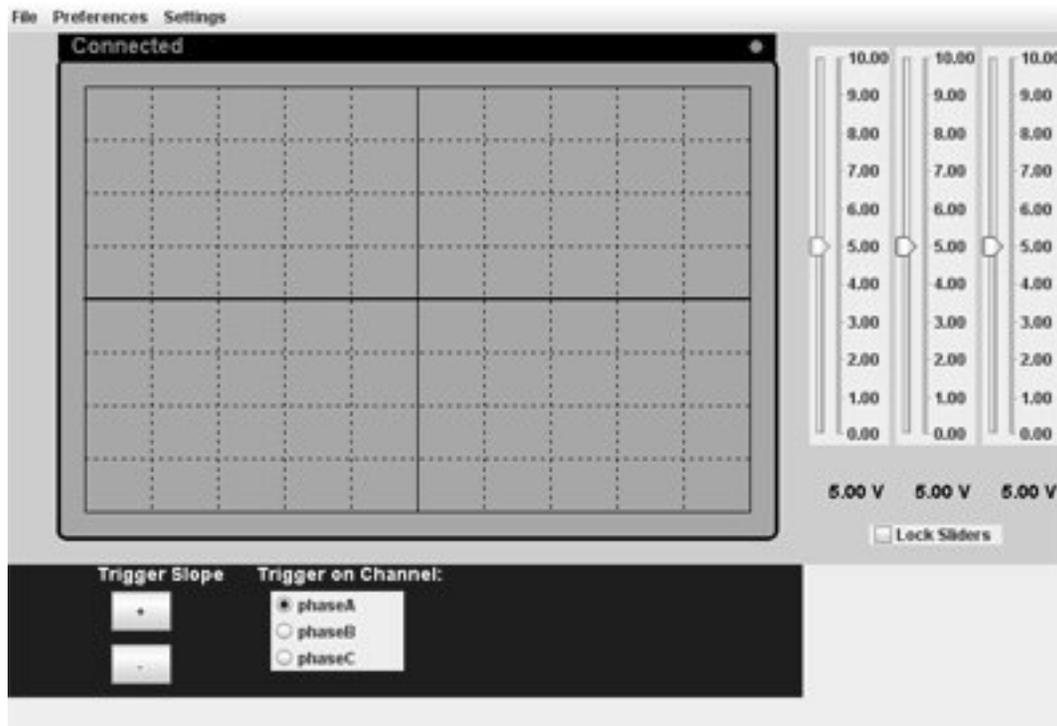


Figure 1

TABLE 1	
Function	Description
<code>void initialize_UART (int, unsigned char);</code>	Initializes the UART, first argument specifies channel number (1,2,3,or 4), second argument specifies the data rate. See the "mainxxx.h" file for the definitions for the data rate constants.
<code>int UART_send_byte (int, unsigned char);</code>	Sends a single 8 bit character to a UART channel. First argument specifies the channel, second argument specifies the character.
<code>int UART_rcv_byte (int, unsigned char*);</code>	Receives a single 8 bit character from a UART channel. First argument specifies the channel, second argument specifies a pointer to the character.
<code>int UART_send_block (int, unsigned char*,unsigned int);</code>	Sends a block of 8 bit characters to a UART channel. First argument specifies the channel, second argument specifies a pointer to the character array, third channel specifies the number of characters to send.
<code>int UART_rcv_block (int, unsigned char*,unsigned int);</code>	Reads a block of 8 bit characters from a UART channel. First argument specifies the channel, second argument specifies a pointer to the character array, third channel specifies the number of characters to read.
<code>unsigned int UART_rcv_count(int);</code>	Returns the number of characters available to be read. Argument specifies the UART channel to query.
<code>unsigned int UART_xmit_count(int);</code>	Returns the number of characters in the transmit output buffer. Can be used to determine if there is enough room in the transmit buffer for more data. Argument specifies the UART channel to query.
<code>void service_serial_port(void);</code>	This function handles all of the low level polling for the serial port. It should be called in the application's main loop often enough to prevent receiver underrun conditions.

8 The Possibilities

The simple_java_demo demonstration program demonstrates two way communication between the eZdsp and a Java applet running in a browser window on the PC. Using this as a starting point, many interesting applications can be created. Here are some examples:

- 1) Data from the eZdsp can be sent to the PC and stored on disk.
- 2) A GUI (graphical user interface) can be created as a Java applet, and used to control various parameters on the eZdsp.
- 3) A eZdsp application can be controlled and monitored from any location having LAN access, and for that matter, anywhere having internet access.
- 4) DSP/BIOS instrumentation variables can be sent to any PC anywhere on the internet, with little performance impact on the DSP application.

9 Version History

Date	Version	Description
1-July-2008	1.0	Original
16-July-2008	1.1	Various changes
21-May-2009	1.2	Correct part number – was 9-403-62