

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

### ezLINX iCoupler Isolated Interface Development Environment

#### **FEATURES**

Sample PC application

- Open source software
- Uses Microsoft .NET Framework, Version 4
- Integrates with the ezLINX hardware platform via an

isolated USB

Allows a complete plug and play evaluation and development experience with 8 isolated communication standards Isolated USB Isolated CAN Isolated RS-485/RS-422 Isolated RS-232 Isolated I<sup>2</sup>C

Isolated SPI Isolated LVDS

#### **APPLICATIONS**

Isolated communication interfaces

#### SOFTWARE PACKAGE CONTENTS

*ez*LINX sample PC application install *ez*LINX USB drivers

#### SOFTWARE REQUIREMENTS

Windows XP, Windows Vista, or Windows 7

#### **GENERAL DESCRIPTION**

The *ez*LINX<sup>™</sup> hardware platform contains an Analog Devices, Inc., ADSP-BF548 processor running the uCLinux<sup>™</sup> kernel and the *ez*LINX embedded software. A sample PC application is also provided, interfacing with the *ez*LINX hardware via an isolated USB. The embedded software is written in C, and the sample PC application is written in Microsoft Visual C#; both use Microsoft .NET Framework, Version 4.0.

The *ez*LINX software and hardware allow:

- Simultaneously transmitting and receiving data on multiple isolated interfaces
- Switching between running interfaces
- Viewing data traffic in real time
- Customizing interfaces to suit various applications
- Easy updating of embedded software, or firmware, via an isolated USB
- Quick saving and loading of an entire configuration for all communication standards
- Hardware routing of signals between interfaces

The open source nature of both the sample PC application and the embedded software allows the user to view and edit the source code of the application to optimize the *ez*LINX hardware system for a given application. The source code and sample PC application can be downloaded from the *ez*LINX *i*Coupler Isolated Interface Development Environment wiki page.



#### ezLINX SAMPLE PC APPLICATION MAIN WINDOW

Figure 1.

# TABLE OF CONTENTS

Features
Applications1
Software Package Contents 1
Software Requirements 1
General Description
ezLINX Sample PC Application Main Window1
Revision History 2
Main Window 3
Main Window Buttons and Options3
Board Configuration Window 4
Setting the Network Features5
Checking/Updating the Firmware5
Enabling/Disabling the Transceivers 6
Isolated RS-232 7
RS-232 Configuration Window7
RS-232 Send/Receive Window 8
Isolated RS-485/RS-42210
RS-485/RS-422 Configuration Window10
RS-485/RS-422 Send/Receive Window11

Isolated I <sup>2</sup> C
I <sup>2</sup> C Configuration Window13
I <sup>2</sup> C Send/Receive Window14
Isolated SPI16
SPI Configuration Window16
SPI Send/Receive Window17
Isolated CAN19
CAN Configuration Window19
CAN Send/Receive Window
Isolated LVDS
LVDS Configuration Window22
LVDS Send/Receive Window
GPIO (LEDs)
GPIO Configuration Window25
Updating the Firmware
Transceivers Status Window
Transceivers Configuration Window
Transceivers Configuration Window Buttons and Options 28

#### **REVISION HISTORY**

12/13—Revision 0: Initial Version

## MAIN WINDOW

Upon starting the *ez*LINX sample PC application, the **Main** window opens (see Figure 2).

This window provides a complete overview of the *ez*LINX hardware system and shows how it connects to the ADSP-BF548 processor. The sample PC application allows simultaneous use and evaluation of multiple communication standards.

#### MAIN WINDOW BUTTONS AND OPTIONS

The application has two sidebars at the left and right of the **Main** window that contain seven active buttons for accessing various transceiver interfaces. These interfaces can be accessed from any window in the application software. The color of each button indicates the status of the corresponding transceiver as follows:

- Light gray indicates that the transceiver is deactivated and disconnected.
- Steel blue indicates that the transceiver is activated and disconnected.
- Royal blue indicates that the transceiver is activated and connected.

The **STATUS** bar along the bottom of the window indicates whether the application is connected to an *ez*LINX hardware platform, the IP address of that hardware platform, and the transceivers on the hardware that are currently enabled. Clicking the word **STATUS** opens the **Transceiver Status** window, where you can view the amount of data sent and received by each interface in the current session. The **Main** window of the application also has three buttons located under the system block diagram: **Connect**, **View Configuration**, and **Configure**.

#### **Connect Button**

Clicking **Connect** establishes a connection with the *ez*LINX hardware board using the current IP address configuration.

#### **View Configuration Button**

Clicking **View Configuration** opens the **Transceivers Configuration** window (see Figure 48), which allows viewing the global configuration of all transceivers and GPIOs on the *ez*LINX hardware platform.

#### **Configure Button**

Clicking **Configure** opens the **Board Configuration** window (see Figure 3 and Figure 4), which allows

- Configuring the IP address of the *ez*LINX hardware platform that the PC application software connects to.
- Changing the IP configuration of the connected *ez*LINX hardware board.
- Applying updates to the embedded software of the *ez*LINX hardware board.



Figure 2. Main Window

## **BOARD CONFIGURATION WINDOW**

There are two ways to open the **Board Configuration** window:

- Click **Configure** (see Figure 3).
- Click the image of the *ez*LINX hardware platform on the bottom left of any window in the application software (see Figure 4).

The **Board Configuration** window is divided into three sections that allow setting the network features, updating the

firmware, and the enabling/disabling the transceivers. The buttons near the bottom of the window depend on which method is used to open the **Board Configuration** window. If the window is accessed by clicking **Configure**, the buttons appear as **Use Changes**, **View Configuration**, and **Cancel Changes**. If the window is accessed by clicking the image of the *ez*LINX hardware platform, the buttons appear as **Connect**, **View Configuration**, and **Configure**.



Figure 3. Board Configuration Window Accessed by Clicking Configure



Figure 4. Board Configuration Window Accessed by Clicking the Image of the ezLINX Hardware Platform

#### SETTING THE NETWORK FEATURES

The upper left section of the **Board Configuration** window (shown in Figure 5) is used to modify the network features of the *ez*LINX hardware platform.

Connect To IP Address	192.168.3.21	
Set New Address To:	192.168.3.21	
New Subnet Mask :	255.255.255.0	
New IP Gateway:	192.168.3.1	

Figure 5. Network Features Box

The boxes in this section can be used as follows:

- **Connect To IP Address** box: Specify the IP address of the *ez*LINX hardware platform that the PC application software connects to when **Connect** is clicked. The default address is 192.168.3.21.
- Set New Address To box: Specify a new IP address to configure the connected *ez*LINX hardware platform. You must select the Set New Address To box and connect an *ez*LINX hardware platform before setting a new address or using the New Subnet Mask and New IP Gateway functions.
- New Subnet Mask box: Specify a new subnet mask to configure the connected *ez*LINX hardware platform.
- New IP Gateway box: Specify a new IP gateway to configure the connected *ez*LINX hardware platform.

#### **CHECKING/UPDATING THE FIRMWARE**

The upper right section of the **Board Configuration** window (shown in Figure 6) is used to easily load newer firmware versions and check the current version of both the firmware (via the **Embedded Version** box in Figure 6) and the PC application software (via the **PC Version** box in Figure 6).

Update Firmwa	are	
192.168.3.21		Send
PC Version :	1.0.3	
Embedded Version :	0.0.0	Check

Figure 6. Firmware Upgrade Box

The boxes in this section can be used as follows:

- Server IP box: Specify the IP address of the server containing the new version of the embedded software. Click Send to load the newer version of the software to the connected *ez*LINX hardware platform. For more information about performing a firmware update, see the Updating the Firmware section of this user guide.
- **PC Version** box: This box shows the current version of the PC application software.
- **Embedded Version** box: This box shows the version of the embedded software on the *ez*LINX hardware platform. Click **Check** to request which version is currently on the hardware platform.

#### **ENABLING/DISABLING THE TRANSCEIVERS**

The lower right section of the **Board Configuration** window (shown in Figure 7) can be used to enable or disable any of the transceivers on the *ez*LINX hardware platform. To enable or disable a transceiver, select or clear the box next to the appropriate interface name.

Enable/Disable Transceiv	vers:
RS-485/UART2 RS-232/UART3 CAN/CAN0 LVDS/SPORT2 12C/TWI0 SPI/SPI1 GPI0/GPI0	

Figure 7. Enable/Disable Transceivers Box

A notification message is displayed when you hover the mouse over a button corresponding to a selected transceiver, indicating that the transceiver is currently enabled (see Figure 8).



Figure 8. Notification Message for Enabled Transceiver

## **ISOLATED RS-232**

The isolated RS-232 interface on the *ez*LINX development platform is implemented using an Analog Devices ADM3251E driver/receiver. For more information about the hardware implementation of the RS-232 interface, see the *ez*LINX Isolated RS-232 Implementation wiki page.

#### **RS-232 CONFIGURATION WINDOW**

To open the RS-232 interface, click **RS232 UART3** from the right sidebar of any window in the application software. This opens the **RS-232 Configuration** window (see Figure 10).

To configure the RS-232 for use, select the **Transceiver Enable** check box. This box is present on all the configuration windows and is used to enable or disable the corresponding transceiver on the *ez*LINX hardware.

The **RS-232 Configuration** window is divided into three sections that control selecting the communication interface, the RS-232 settings, and the RS-232 routing.

**UG-461** 

#### Selecting the Communication Interface

The upper left section of the **RS-232 Configuration** window (shown in Figure 9) is used to select the appropriate communication interface for the RS-232 transceiver. Select **UART3** from the **Interface** drop-down menu.

Interfac	e Settings	
Protocol	RS232 -	
Interface	UART2	
RS-232 Setting	UART2 UART3	00000

Figure 9. Selecting the Communication Interface

	iCoupler® Is	ez LINX™ olated Interface Development Environme	ent ANALOG
RS485 UART2			R\$232 UART3
CAN CAN0	Interface	Settings I Transceiver Enable	LVDS SPORT2
I2C TWI1	Protocol RS2 Interface UAP	RoutingFrom) 32 C Enable 13 LIART3 To LIART3	SPI SPI0
6P10 6P10	RS-232 Settings	110	CAN
	Parity Stop Bits	none  ADM3252E	
	Word Size Flow Control	7 V	
	<c Use Chan</c 	lick To Configure Interface>	PCLinux

Figure 10. RS-232 Configuration Window

#### Configuring the RS-232 Settings

The **RS-232 Settings** box (shown in Figure 11) is used to communicate with the *ez*LINX hardware board through the RS-232 port.

RS-232 Settings				
Baud Rate	110			
Parity	none	•		
Stop Bits	1		ADM3251E	
Word Size	7	•	=	
Flow Control	None	•		

Figure 11. RS-232 Settings Box

Configure the device using the following controls:

- **Baud Rate** box: Select the baud rate for the RS-232 device. Note that the performance of the ADM3251E RS-232 transceiver is not specified for a baud rate above 460,800 bits/sec. From the **Baud Rate** drop-down menu, you can select different transmission rate values, from 110 bits/sec to 1,000,000 bits/sec.
- **Parity** box: Select whether to append an even, odd, or no parity bit to the end of each word transmitted.
- Stop Bits box: Select whether one or two stop bits are used.
- Word Size box: Select whether the application sends 7- or 8-bit words.
- Flow Control box: Select whether to use flow control. When communicating with another *ez*LINX hardware platform, select **None** in this box.

#### **Configuring RS-232 Routing**

The RS-232 interface supports hardware routing to the outputs of other interfaces. To enable hardware routing, select the **Enable** check box in the **Routing(From)** section, located in the upper right of the **RS-232 Configuration** window, and choose the interface to route to using the drop-down menu (see Figure 12). To route RS-232 to RS-485, select **UART2** from the drop-down menu.

То	UART2	•	
	То	To UART2	To UART2 -

Figure 12. Routing(From) Box

Confirm your selection by clicking **Use Changes** near the bottom of the window. If the sample PC application is connected to an

*ez*LINX hardware platform and the **Transceiver Enable** check box is selected, the **Send/Receive** window opens (see Figure 13).

#### **RS-232 SEND/RECEIVE WINDOW**

The **RS-232 Send/Receive** window is shown in Figure 13. The window consists of two main sections: a **Send** section on the left and a **Receive** section on the right.

To send data from the RS-232 port, type the data to be sent into the **Send** box, located just to the left of the **Send Data Format** box. Click **Send Data** to transmit the contents of the text box. Data sent to the RS-232 port automatically appears in the **Receive** section of the window in real time.

The **RS-232 Send/Receive** window also offers a variety of other functions, as follows:

- **On/Off** button: Enables or disables the transceiver, respectively. When **Off** is clicked, the RS-232 transceiver is disconnected and cannot transmit or receive data.
- Send Data button: Transmits the text contained in the Send box.
- **Auto** button: Enables or disables the auto setting. When the auto setting is enabled, data is transmitted automatically as you type it in the **Send** box.
- **Load File** button: Opens a menu to select a .txt file to be loaded. After the file is loaded, the contents of the file are automatically sent through the RS-232 port.
- Clear button: Clears any text that was entered in the Send box.
- Send Data Format box: Allows choosing whether to send the characters in the Send box as hexadecimal or ASCII data.
- Receive Data Format box: Allows choosing whether to display the received data as hexadecimal or ACSII characters.
- Save To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is saved to this file. To stop saving all received data to this file, click Save To File again.
- Log To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is logged to this file with a timestamp. To stop logging all received data to this file, click Log To File again.

To close the **RS-232 Send/Receive** window, turn the transceiver off and click a different interface button, or click **RS232 UART3** from the right sidebar to return to the **RS-232 Configuration** window.

	ez LINX™ iCoupler® Isolated Interface Development Environment	
RS485 UART2	Interface RS232/UART3 V ON OFF	RS232 UART3
CAN CANO	Send Send Data Format ASCII V Clear Receive Data Format ASCII V Clear	LVDS SPORT2
I2C TWI1		SPI SPI0
6P10 6P10		CAN
-	a)	
	Send Data Auto Load File Save To File Log To File	
	< Click To Select Another Interface >	
	Disconnect Configure New Screen	

Figure 13. RS-232 Send/Receive Window

## **ISOLATED RS-485/RS-422**

Both the isolated RS-485 and the isolated RS-422 interfaces on the *ez*LINX development platform are implemented using an Analog Devices ADM2587E transceiver.

For more information about the hardware implementation of the RS-485/RS-422 interfaces, see the *ez*LINX Isolated RS-485/RS-422 Implementation wiki page.

#### **RS-485/RS-422 CONFIGURATION WINDOW**

In the *ez*LINX sample PC application, both the RS-485 and the RS-422 protocols are accessed from the same configuration window. To open the RS-485/RS-422 interface, click **RS485 UART2** from the left sidebar of any window in the application software. This opens the **RS-485/RS-422 Configuration** window (see Figure 15).

To configure the RS-485/RS-422 for use, select the **Transceiver Enable** check box. This box is present on all the configuration windows and is used to enable or disable the corresponding transceiver on the *ez*LINX hardware.

The **RS-485/RS-422 Configuration** window is divided into three sections that control selecting the communication interface, the RS-485/RS-422 settings, and the RS-485/RS-422 routing.

#### Selecting the Communication Interface

The upper left section of the **RS-485/RS-422 Configuration** window (shown in Figure 14) is used to select the appropriate communication interface for the RS-485/RS-422 transceiver. Select **UART2** from the **Interface** drop-down menu

Interfa	ce Setting	gs
Protocol	RS485	•
Interface	UART2	-
	UART2	
RS-485 Setti		10050

*Figure 14. Selecting the Communication Interface* 

<i>i</i> Coupler® Isolated Interface Development Environment	ANALOG DEVICES
	RS232 UART3
Interface Settings	LVDS SPORT2
Protocol RS485  Interface UART2 Interface UART2 Interface INTERFACE To UART2 INTERFACE	SPI SPI0
RS-465 Settings Boud Rate 110	CAN
Stop Bits 1 ADM2587E	
Flow Control None  Duplex Hall Duplex	
<click configure="" interface="" to=""></click>	μ <sup>CLinux</sup>
	Interface Settings       Transceiver Enable         Protocol       R5485         Interface Settings       Countral Form         Protocol       R5485         Interface Settings       Interface Transceiver Enable         Interface       Interface         Baud Rate       Interface         Baud Rate       Interface         Baud Rate       Interface         Parity       Interface         Stop Bits       Interface         Upplex       Hab Duplex         Click To Configure Interface>

#### Figure 15. RS-485 Configuration Window

#### Selecting the RS-485 Settings

The **RS-485 Settings** box (shown in Figure 16) is used to communicate with the *ez*LINX hardware board through the RS-485 port.

Baud Rate	110	•	
Parity	none	•	
Stop Bits	1	•	ADM2597E
Word Size	7	•	ADW2307E
Flow Control	None	•	
Duplex	Half Duplex	•	

Figure 16. RS-485 Settings Box

Configure the device using the following controls:

- **Baud Rate** box: Select the baud rate for the RS-485/RS-422 device. Note that the performance of the ADM2587E RS-485/ RS-422 transceiver is not specified for a baud rate above 500 kbits/sec. From the **Baud Rate** drop-down menu, you can select different transmission rate values, from 110 bits/sec to 1,000,000 bits/sec.
- **Parity** box: Select whether to append an even, odd, or no parity bit to the end of each word transmitted.
- Stop Bits box: Select between using one or two stop bits.
- Word Size box: Select whether the application sends 7- or 8-bit words.
- Flow Control box: Select whether to use flow control. When the sample PC application is connected to the *ezLINX* hardware platform, select **None** in this box.
- **Duplex** box: Choose between half- and full-duplex operation. If using half-duplex operation, Jumpers JP3, JP4, and JP40 should be connected on the *ez*LINX hardware board.

#### Selecting the RS-485/RS-422 Routing

The RS-485/RS-422 interface supports hardware routing to the outputs of other interfaces. To enable hardware routing, select the **Enable** check box in the **Routing(From)** section, located in the upper right of the **RS-485/RS-422 Configuration** window, and choose the interface to route to using the drop-down menu (see Figure 17). To route RS-485/RS-422 to RS-232, select **UART3** from the drop-down menu.

Routing(From)				
Enable				
UART2	То	UART3	-	59-01
UARTZ	10	Orario		

Figure 17. Routing(From) Box

Confirm your selection by clicking **Use Changes** near the bottom of the window. If you are connected to an *ez*LINX hardware platform and the **Transceiver Enable** check box is selected, the **Send/Receive** window opens (see Figure 18).

#### RS-485/RS-422 SEND/RECEIVE WINDOW

The **RS-485/RS-422 Send/Receive** window is shown in Figure 18. The window is identical to the **RS-232 Send/Receive** window. The left section of the window is used to transmit data by clicking **Send Data**, and the right section of the window is used to receive data in real time.

The **RS-485/RS-422 Send/Receive** window also offers a variety of other functions, as follows:

- **On/Off** button: Enables or disables the transceiver, respecttively. When **Off** is clicked, the RS-485/RS-422 transceiver is disconnected and cannot transmit or receive data.
- Send Data button: Transmits the text contained in the Send box.
- **Auto** button: Enables or disables the auto setting. When the auto setting is enabled, data is transmitted automatically as you type it in the **Send** box.
- Load File button: Opens a menu to select a .txt file to be loaded. After the file is loaded, the contents of the file are automatically sent through the RS-485/RS-422 port.
- Clear button: Clears any text that was entered in the Send box.
- Send Data Format box: Allows choosing whether to send the characters in the Send box as hexadecimal or ASCII data.
- Receive Data Format box: Allows choosing whether to display the received data as hexadecimal or ACSII characters.
- Save To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is saved to this file. To stop saving all received data to this file, click Save To File again.
- **Log To File** button: Opens a window to create a .txt file. After a .txt file is created, all received data is logged to this file with a timestamp. To stop logging all received data to this file, click **Log To File** again.

To close the **RS-485/RS-422 Send/Receive** window, turn the transceiver off and click a different interface button, or click **RS485 UART2** from the left sidebar to return to the **RS-485/RS-422 Configuration** window.

# UG-461

	ez LINX™ iCoupler® Isolated Interface Development Environment	
RS485 UART2	Interface RS485/UART2 V ON OFF	RS232 UART3
CAN CANO	Send Send Data Format ASCII Clear Receive Data Format ASCII Clear	LVDS SPORT2
I2C TWI1		SPI SPIO
GPIO GPIO		CAN
***		
	Send Data Auto Load File Save To File Log To File	
	< Click To Select Another Interface >	
	Disconnect Configure New Screen	pCLinux

Figure 18. RS-485/RS-422 Send/Receive Window

## ISOLATED I<sup>2</sup>C

The isolated inter-integrated circuit (I<sup>2</sup>C) interface on the *ez*LINX development platform is implemented using Analog Devices ADuM1250 and ADuM5000 isolators.

For more information about the I<sup>2</sup>C hardware implementation, see the *ez*LINX Isolated I<sup>2</sup>C Implementation wiki page.

#### I<sup>2</sup>C CONFIGURATION WINDOW

To open the I<sup>2</sup>C interface, click **I2C TWI1** from the left sidebar of any window in the application software. The **I<sup>2</sup>C Configuration** window opens (see Figure 20)

To configure the  $I^2C$  for use, select the **Transceiver Enable** check box. This box is present on all the configuration windows and is used to enable or disable the corresponding transceiver on the *ez*LINX hardware.

The **I**<sup>2</sup>**C Configuration** window is divided into two sections that control selecting the communication interface and the I<sup>2</sup>C settings.

#### Selecting the Communication Interface

The upper left section of the **I**<sup>2</sup>**C Configuration** window (shown in Figure 19) is used to select the appropriate communication interface for the I<sup>2</sup>C transceiver. Select **TWI1** from the **Interface** drop-down menu.

# Interface Settings Protocol I2C Interface

Figure 19. Selecting the Communication Interface

	ez LINX™ iCoupler® Isolated Interface Development Environment	ANALOG
RS485 UART2		RS232 UART3
CAN CANO	Interface Settings	LVDS SPORT2
12C TWI1	Protocol 12C  Interface Tw11	SPI SPI0
GPIO GPIO	I2C Settings Client 10	CAN
	Baud Rate 100 V Mode Master V Write Read Flag Write V	
		····;
	<click configure="" interface="" to=""></click>	
	Use Changes View Configuration Cancel Changes	BLACK

Figure 20. I<sup>2</sup>C Configuration Window

## **Software User Guide**

#### Selecting the I<sup>2</sup>C Settings

The **I<sup>2</sup>C Settings** box (shown in Figure 21) is used to communicate with the *ez*LINX hardware board through I<sup>2</sup>C protocol.



Figure 21. I<sup>2</sup>C Settings Box

Configure the device using the following controls:

- **Client** box: When the ADuM1250 is used as a master device, this box specifies which client to connect to. When the ADuM1250 is used as a slave device, this box specifies the client address of the transceiver.
- **Baud Rate** box: Select either 100 bits/sec or 400 bits/sec as the baud rate.
- **Mode** box: Select whether to configure the connected device as a master or a slave.
- Write Read Flag box: Select whether the device is performing a read or write operation. Confirm your selection by clicking Use Changes near the bottom of the window. If you are connected to an *ezLINX* hardware platform and the Transceiver Enable check box is selected, the I<sup>2</sup>C Send/Receive window opens (see Figure 22).

#### I<sup>2</sup>C SEND/RECEIVE WINDOW

The I<sup>2</sup>C Send/Receive window is shown in Figure 22. The window is identical to the RS-232 Send/Receive window. The left section of the window is used to transmit data by clicking Send Data, and the right section of the window is used to receive data in real time.

The I<sup>2</sup>C module can only send hexadecimal data. When the *ez*LINX I<sup>2</sup>C transceiver is used to write to a slave device, data must be sent in multiples of two bytes (four hexadecimal digits) for proper operation.

The first byte represents the memory address to write to, and the second byte contains the value to be written to that address. When the I<sup>2</sup>C transceiver is used to read from a slave device, the data must be sent as one or more whole bytes (multiples of two hexadecimal digits). Each byte specifies a memory address to read from. When a byte of data is sent from a register of the slave, the value of that register is received by the master and is displayed in the **Receive** section of the window in real time.

The **I**<sup>2</sup>**C Send/Receive** window also offers a variety of other functions, as follows:

- **On/Off** button: Enables or disables the transceiver, respectively. When **Off** is clicked, the I<sup>2</sup>C transceiver is disconnected and cannot transmit or receive data.
- Send Data button: Transmits the text contained in the Send box.
- **Auto** button: Enables or disables the auto setting. When the auto setting is enabled, data is transmitted automatically as you type it in the **Send** box.
- Load File button: Opens a menu to select a .txt file to be loaded. After the file is loaded, the contents of the file are automatically sent through the I<sup>2</sup>C port. Files must contain hexadecimal characters only.
- Clear button: Clears any text that was entered in the Send box.
- Save To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is saved to this file. To stop saving all received data to this file, click Save To File again.
- Log To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is logged to this file with a timestamp. To stop logging all received data to this file, click Log To File again.

To close the **I**<sup>2</sup>**C Send/Receive** window, turn the transceiver off and click a different interface button, or click **I2C TWI1** from the left sidebar to return to the **I**<sup>2</sup>**C Configuration** window.

	iCoupler® Iso	ez LINX" lated Interface D	evelopment Env	vironment	
RS485 UART2	Interface 12C/Twi1				R\$232 UART3
CAN CANO	Send	Clear	Receive	Clear	LVDS SPORT2
12C TWI1					SPI SPI0
GPID GPID					CAN
***					
	Send Data Aut	Load File	Save To File	Log To File	
	<	Click To Select Ano	ther Interface >		
	Disconnec	Configure	New Scre	en	BLACK

Figure 22. I<sup>2</sup>C Send/Receive Window

# UG-461

## **ISOLATED SPI**

The two isolated serial peripheral interface (SPI) ports on the *ez*LINX development platform are implemented using Analog Devices ADuM3401, ADuM3402, and ADuM5000 signal and power isolators.

For more information about the SPI hardware implementation, see the *ez*LINX Isolated SPI Implementation wiki page.

#### SPI CONFIGURATION WINDOW

To open the SPI interface, click **SPI SPIO** on the right sidebar of any window in the application software. This opens the **SPI Configuration** window (see Figure 24).

To configure the SPI for use, select the **Transceiver Enable** check box. This box is present on all the configuration windows and is used to enable or disable the corresponding transceiver on the *ez*LINX hardware.

The **SPI Configuration** window interface is divided into three sections that control selecting the communication interface, the SPI settings, and the SPI routing.

#### Selecting the Communication Interface

The upper left section of the **SPI Configuration** window (shown in Figure 23) is used to select the appropriate communication interface for the SPI transceiver. Select **SPI0**, **SPI1**, or **SPI2** from the **Interface** drop-down menu.



Figure 23. Selecting the Communication Interface

	ez LINX™ iCoupler® Isolated Interface Development Environment	
RS485 UART2		RS232 UART3
CAN	Interface Settings Transceiver Enable	LVDS SPORT2
I2C TWI1	Protocol SPI   Protocol SPI Interface SPI0  SPI0  To SPI0	SPI SPI0
6P10 6P10	SPI Settings	CAN
	Max Speed         200000         Operation Mode           LSB First         0         ✓         Master Mode         A0w25876           Word Size         a         ✓         Stave Mode         A0w25876	
	<click configure="" interface="" to=""></click>	ucLinux A
	Use Changes View Configuration Cancel Changes	BLACK CO

Figure 24. SPI Configuration Window

#### Selecting the SPI Settings

The **SPI Settings** box (shown in Figure 25) is used to communicate with the *ez*LINX hardware board through SPI protocol.

					B 81
Max Speed	2000000		Operation Mode		E 8
LSB First	0	•	Master Mode Channel		ADUM3403
Word Size	8	•	O Slave Mode	2	E 5

Figure 25. SPI Settings Box

Use the following controls to set the SPI parameters:

- **Max Speed** box: Specify the maximum speed (in bps) for the SPI interface. The maximum data rate that can be used is 32.81 MHz.
- LSB First box: Activate the least significant byte (LSB) first mode. Upon power up, the most significant byte (MSB) first mode is the default. The mode can be changed by programming the configuration register. In LSB first mode, the serial exchange starts with the lowest order bit (LSB) and ends with the highest order bit (MSB). The instruction is 16 bits long, consisting of two bytes. From the **SPI Configuration** window, you can choose the number of bits to be reversed using the LSB First drop-down box.
- Word Size box: Select whether to send data as 8-, 16-, or 32-bit words.
- **Operation Mode** box: Select whether the device is operating as a master or as a slave.
- Channel box: Select which slave to connect to; each channel corresponds to a slave select line. This function is available only when the operation mode selected is master. The *ezLINX* hardware platform has three slave select lines; therefore, a master can be connected to up to three slave devices. Confirm your selection by clicking Use Changes near the bottom of the window. If you are connected to an *ezLINX* hardware platform and the Transceiver Enable check box is selected, the SPI Send/Receive window opens (see Figure 27).

#### Selecting the SPI Routing

The SPI interface supports hardware routing to the outputs of other interfaces. To enable hardware routing, select the **Enable** check box in the **Routing(From)** section, located in the upper right of the **SPI Configuration** window, and choose the interface to route to from the drop-down menu (see Figure 26). To route SPI to RS-232, select **UART3** from the drop-down menu.



Figure 26. Routing(From) Box

Confirm your selection by clicking **Use Changes** near the bottom of the window. If you are connected to an *ez*LINX hardware platform and the **Transceiver Enable** check box is selected, the **SPI Send/Receive** window opens.

#### SPI SEND/RECEIVE WINDOW

The **SPI Send/Receive** window is shown in Figure 27. The window is identical to the **RS-232 Send/Receive** window. The left section of the window is used to transmit data by clicking **Send Data**, and the right section of the window is used to receive data in real time.

The **SPI Send/Receive** window also offers a variety of other functions, as follows:

- **On/Off** button: Enables or disables the transceiver, respectively. When **Off** is clicked, the SPI transceiver is disconnected and cannot transmit or receive data.
- **Auto** button: Enables or disables the auto setting. When the auto setting is enabled, data is transmitted automatically as you type it in the **Send** box.
- **Load File** button: Opens a menu to select a .txt file to be loaded. After the file is loaded, the contents of the file are automatically sent through the SPI port.
- Clear button: Clears any text that was entered in the Send box.
- Send Data Format box: Allows choosing whether to send the characters in the Send box as hexadecimal or ASCII data.
- Receive Data Format box: Allows choosing whether to display the received data as hexadecimal or ACSII characters.
- Save To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is saved to this file. To stop saving all received data to this file, click Save To File again.
- Log To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is logged to this file with a timestamp. To stop logging all received data to this file, click Log To File again.

To close the **SPI Send/Receive** window, turn the transceiver off and click a different interface button, or click **SPI SPIO** from the right sidebar to return to the **SPI Configuration** window.

# UG-461

	ez LINX™ iCoupler® Isolated Interface Development Environment	
RS485 UART2	Interface SPI/SPI0 V DN OFF	RS232 UART3
CAN CANO	Send Send Data Format ASCII Clear Receive Data Format ASCII Clear	LVDS SPORT2
12C TWI1		SPI SPI0
6P10 6P10		CAN
-		
	~	
	Send Data Auto Load File Save To File Log To File	
	< Click To Select Another Interface >	
Service Services		UCLinux A

Figure 27. SPI Send/Receive Window

The isolated controller area network (CAN) interface on the *ez*LINX development platform is implemented using an Analog Devices ADM3053 transceiver.

For more information about CAN hardware implementation, see the *ez*LINX Isolated CAN Implementation wiki page.

#### **CAN CONFIGURATION WINDOW**

To open the CAN interface, click **CAN CAN0** from the left sidebar of any window in the application software. This opens the **CAN Configuration** window (see Figure 29).

(The **CAN** button in the right sidebar is unimplemented and cannot be selected.)

To configure the CAN for use, select the **Transceiver Enable** check box. This box is present on all the configuration windows and is used to enable or disable the corresponding transceiver on the *ez*LINX hardware.

The **CAN Configuration** window is divided into three sections that control selecting the communication interface, the CAN settings, and the CAN routing.

#### Selecting the Communication Interface

The upper left section of the **CAN Configuration** window (shown in Figure 28) is used to select the appropriate communication interface for the CAN transceiver. Select **CAN0** from the **Interface** drop-down menu.

Protocol	CAN	-
Interface	CANO	-
	CAN0	
CAN Settings	CAN1	

Figure 28. Selecting the Communication Interface

RS485 UART2	Protocol CAN Interface Settings CAN AN Settings Bit Rate: 125000 CAN Error Filter CAN ERR D CAN	Filter 1	Transa Couting(From) CANO Normal	Ceiver Enat	io Io	RS232 UART3 LVDS SP0RT2 SP10
12C TW11 CA 6PI0 6PI0	Protocol CAN Interface CAN0 AN Settings Bit Rate : 125000 Error Filter	Filter 1	CANO	To CAN	0	SPORT2
12C TWT1 CA GPI0 GPI0 	Interface         CAN0           AN Setting:         Bit Rate :           Bit Rate :         125000           Error Filter         CAN.ERR.DX.TIMEOUT	Filter 1	CAN0 Normal	To CAN	0	SPI SPI0
6P10 0	Bit Rate : 125000	Filter 1				
		Mask 1				CAN
	CAN_ERR_LOSTARB	Filter 2 Mask 2				(***
	ADM3053	Filter 3 Mask 3				
	<click td="" to<=""><td>Confi</td><td>gure Interf</td><td>ace&gt;</td><td>10</td><td>pCLinux</td></click>	Confi	gure Interf	ace>	10	pCLinux

Figure 29. CAN Configuration Window

#### Selecting the CAN Settings

The **CAN Settings** box (shown in Figure 30) is used to communicate with the *ez*LINX hardware board through the CAN protocol.

a at ooungo		Normal	Extended
Bit Rate : 125000	Filter 1	0000000000000000	0000000000
Error Filter	Mask 1	0000000000	0000000000
CAN_ERR_TX_TIMEOU1	Filter 2	m 0000000000	0000000000
CAN_ERR_CRTL	Mask 2	0000000000	0000000000
	Filter 3	0000000000	
ADM3053	Mask 3		

Figure 30. CAN Settings Box

Use the following controls to specify the CAN settings:

- **Bit Rate** box: Select the bit rate for the ADM3053. Note that the ADM3053 is not specified at bit rates greater than 1 Mbps.
- Error Filter box: Detects hardware issues on the physical transceiver layer as well as arbitration problems and error frames. The reception of error frames is disabled by default. To enable this function, select the Error Filter box. You can then choose your desired error filter from the following list by selecting the relevant box:
  - **CAN\_ERR\_TX\_TIMEOUT**: transceiver timeout (net device driver)
  - CAN\_ERR\_LOSTARB: lost arbitration
  - CAN\_ERR\_CRTL: controller problems
  - CAN\_ERR\_PROT: protocol violations
  - CAN\_ERR\_TRX: transceiver status
  - CAN\_ERR\_ACK: received no acknowledgement of transmission
  - CAN\_ERR\_BUSOFF: bus off
  - CAN\_ERR\_BUSERROR: bus error
  - CAN\_ERR\_RESTARTED: controller restarted
  - CAN\_ERR\_MASK: omit EFF, RTR, ERR flags
- Filter/Mask boxes: The reception of CAN frames can be controlled by three sets of filters/masks. Each filter/mask can be used for messages with either standard or extended identifiers. Note that you must select the check box next to the appropriate name of the filter/mask before entering your selected value. The available filters and masks are as follows:
  - Normal Filter 1 to Normal Filter 3: filter for standard frame (11-bit identifier).
  - Extended Filter 1 to Extended Filter 3: filter for extended frame (29-bit identifier).
  - Normal Mask 1 to Normal Mask 3: mask for standard frame (11-bit identifier).
  - Extended Mask 1 to Extended Mask 3: mask for extended frame (29-bit identifier).

#### Selecting the CAN Routing

The CAN interface supports hardware routing to the outputs of other interfaces. To enable hardware routing, select the **Enable** check box in the **Routing(From)** section, located in the upper right of the **CAN Configuration** window, and choose the interface to route to from the drop-down menu. To route CAN to RS-485, select **UART2** from the drop-down menu.

То	UART2	•	
	То	To UART2	To UART2 🗸

Figure 31. Routing(From) Box

Confirm your selection by clicking **Use Changes** near the bottom of the window. If you are connected to an *ez*LINX hardware platform and the **Transceiver Enable** check box is selected, the **CAN Send/Receive** window opens.

#### **CAN SEND/RECEIVE WINDOW**

The **CAN Send/Receive** window is shown in Figure 33. The window is identical to the **RS-232 Send/Receive** window. The left section of the window is used to transmit data by clicking **Send Data**, and the right section of the window is used to receive data in real time.

The CAN module can only send hexadecimal data. A colon is used to separate the two parts of each CAN message. Each CAN message has both an identifier and the data. The identifier can be either a standard identifier (SID), ranging from 0x000 to 0x7FF, or an extended ID (EXID), ranging from 0x8000 0000 to 0xFFFF FFFF. The data section of each CAN message must be sent as whole bytes (multiples of two hexadecimal digits). Some examples of CAN messages are shown in Figure 32.



Figure 32. Examples of CAN Messages

The **CAN Send/Receive** window also offers a variety of other functions, as follows:

- **On/Off** button: Enables or disables the transceiver, respectively. When **Off** is clicked, the CAN transceiver is disconnected and cannot transmit or receive data.
- Auto button: Enables or disables the auto setting. When the auto setting is enabled, data is transmitted automatically as you type it in the **Send** box.
- Load File button: Before using this function, you must select the File check box. Clicking Load File opens a menu

to select a .txt file to be loaded. Your choice must be a correctly formatted file containing only hexadecimal values, with the identifier and data sections of each CAN message separated by a colon (no spaces). After the file is loaded, the contents of the file are automatically sent through the CAN port.

- Clear button: Clears any text that was entered in the Send box.
- Save To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is saved to this

file. To stop saving all received data to this file, click **Save To File** again.

• Log To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is logged to this file with a timestamp. To stop logging all received data to this file, click Log To File again.

To close the **CAN Send/Receive** window, turn the transceiver off and click a different interface button, or click **CAN CAN0** from the left sidebar to return to the **CAN Configuration** window.

		ace Development Environment	DEVICES
RS485 UART2	Interface CAN/CANO V ON OFF	)	HS232 UART3
CAN CANO	Send Data Clear	Receive Data	LVDS SPORT2
I2C TWI1	SID EXID : Data Bytes	SID EXID : Data Bytes	SPI SPI0
GPI0 GPI0		_	CAN
		_	***
		2	
	Send Data Auto File Load Fil	e Save To File Log To File	
	< Click To Select A	vnother Interface >	
	Disconnect	Configure New Screen	

Figure 33. CAN Send/Receive Window

## **ISOLATED LVDS**

The isolated low voltage differential signaling (LVDS) interface on the *ez*LINX hardware platform is implemented using the ADuM3442 and ADuM5000 signal and power isolators and the ADN4663 and ADN4664 LVDS receivers and drivers.

For more information about LVDS implementation, see the *ez*LINX Isolated LVDS Implementation wiki page.

#### LVDS CONFIGURATION WINDOW

To open the LVDS interface, click **LVDS SPORT2** from the right sidebar of any window in the application software. The **LVDS Configuration** window opens (see Figure 35).

To configure the LVDS for use, select the **Transceiver Enable** check box. This box is present on all the configuration windows and is used to enable or disable the corresponding transceiver on the *ez*LINX hardware.

The **LVDS Configuration** window is divided into three sections that control selecting the communication interface, the LVDS settings, and the LVDS routing.

#### Selecting the Communication Interface

The upper left section of the **LVDS Configuration** window (shown in Figure 34) is used to select the appropriate communication interface for the LVDS transceiver. Select **SPORT2** from the **Interface** drop-down menu.

Protocol	LVDS	-
Interface	SPORT2	-
	SPORT2	
LVDS Settings	SPORT3	

Figure 34. Selecting the Communication Interface

	ez LINX™ iCoupler® Isolated Interface Development Environment	
RS485 UART2	Interface Settings	RS232 UART3
CAN CANO	Protocol LVDS Enable	LVDS SPORT2
I2C TWI1	Interface SPORT2 SPORT2 To SPORT2 V LVDS Settings	SPI SPI0
GPIO GPIO	Baud Rate Tx 2000000 Baud Rate Rx 2000000	CAN
	Word Size         8         •           Frame Delay         0         •         A0083442           Active Low         0         •         •	
	Internal Clock Tx 0 v Internal Clock Rx 0 v	
	Secondary Channel Enable Click To Configure Interface>	
	Use Changes View Configuration Cancel Changes	PCLinux

Figure 35. LVDS Configuration Window

#### Selecting the LVDS Settings

The **LVDS Settings** section (shown in Figure 36) is used to communicate with the *ez*LINX hardware board through the LVDS protocol.

Baud Rate Tx	2000000		
Baud Rate Rx	2000000		
Word Size	8	•	
Frame Delay	0	•	ADM3053
Active Low	0	-	
Internal Clock Tx	0	•	E E
Internal Clock Rx	0	•	
Secondary Channel Enable	0	-	



Configure the device using the following controls:

- **Baud Rate Tx** box: Select the baud rate for transmitting data via the LVDS interface.
- **Baud Rate Rx** box: Select the baud rate for receiving data via the LVDS interface.
- Word Size box: Select whether to send data as 8- or 16-bit words.
- Frame Delay box: Sets the transfer delay.
- Active Low box: Select either Active Low Enable or Power-Down Input with Pull-Down (3 V TTL/CMOS). If EN is held high, EN enables the drivers when low or open circuit and disables the drivers and powers down the device when high.
- Internal Clock Tx box: Select whether to use the internal clock to drive the LVDS transmitter or to trigger on an external clock. Select 1 for an internal clock or 0 for an external clock. For best results with board-to-board communication, one device should use an internal clock and the other device should trigger on this clock.
- Internal Clock Rx box: Select whether to use the internal clock to drive the LVDS receiver or to trigger on an external clock. Select 1 for an internal clock or 0 for an external clock. For best results with board-to-board communication, one device should use an internal clock and the other device should trigger on this clock.
- Secondary Channel Enable box: Select whether to enable the second LVDS channel (Pin 17 to Pin 32). Select 1 to enable this channel or 0 to disable it.

#### Selecting the LVDS Routing

The LVDS interface supports hardware routing to the outputs of other interfaces. To enable hardware routing, select the **Enable** check box in the **Routing(From)** section, located in the upper right of the **LVDS Configuration** window, and choose the interface to route to from the drop-down menu (see Figure 37). To route LVDS to RS-485, select **UART2** from the drop-down menu.

Routing(From)				1
Enable				
SPORT2	То	UART2	•	9-037
				1095

Figure 37. Routing(From) Box

Confirm your selection by clicking **Use Changes** near the bottom of the window. If you are connected to an *ez*LINX hardware platform and the **Transceiver Enable** check box is selected, the **LVDS Send/Receive** window opens.

#### LVDS SEND/RECEIVE WINDOW

The **LVDS Send/Receive** window for the LVDS protocol is shown in Figure 38. The window is identical to the **RS-232 Send/Receive** window. The left section of the window is used to transmit data by clicking **Send Data**, and the right section of the window is used to receive data in real time.

The **LVDS Send/Receive** window also offers a variety of other functions, as follows:

- **On/Off** button: Enables or disables the transceiver, respectively. When **Off** is clicked, the LVDS transceiver is disconnected and cannot transmit or receive data.
- Auto button: Enables or disables the auto setting. When the auto setting is enabled, data is transmitted automatically as you type it in the **Send** box.
- Load File button: Opens a menu to select a .txt file to be loaded. After the file is loaded, the contents of the file are automatically sent through the LVDS port.
- Clear button: Clears any text that was entered in the Send box.
- Send Data Format box: Allows choosing whether to send the characters in the Send box as hexadecimal or ASCII data.
- Receive Data Format box: Allows choosing whether to display the received data as hexadecimal or ACSII characters.
- Save To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is saved to this file. To stop saving all received data to this file, click **Save To File** again.
- Log To File button: Opens a window to create a .txt file. After a .txt file is created, all received data is logged to this file with a timestamp. To stop logging all received data to this file, click Log To File again.

To close the **LVDS Send/Receive** window, turn the transceiver off and click a different interface button, or click **LVDS SPORT2** from the right sidebar to return to the **LVDS Configuration** window.

# UG-461

# Software User Guide

	iCouple	er® Isola	ez LIN ated Interfac	X™ e Development∣	Environment	
RS485 UART2	Interface	DS/SPORT2	IN OFF			RS232 UART3
CAN CANO	Send Send	Data Format A	SCII Clear	Receive Receive Data F	ormat ASCI 💙 Clear	LVDS SPORT2
12C TW11						SPI SPI0
GPI0 GPI0						CAN
122			<i>.</i>		<i></i>	
	Send Data	Auto	Load File	Save To File	Log To File	
		<cli< td=""><td>ck To Select And</td><td>other Interface&gt;</td><td></td><td></td></cli<>	ck To Select And	other Interface>		
		Disconnect	Confi	gure New	Screen	BLACK DC LINUX

Figure 38. LVDS Send/Receive Window

The sample PC application also has GPIO functionality, through which you can control six LEDs on the *ez*LINX hardware platform.

#### **GPIO CONFIGURATION WINDOW**

To open the GPIO interface, click **GPIO GPIO** on the left sidebar of any window in the application software. The **GPIO Interface Settings** window opens (see Figure 41).

To enable a GPIO pin, click the corresponding check box (GPIO 1 to GPIO 6) in the GPIO Settings section of the window.

The **GPIO Configuration** window is divided into two sections that control selecting the communication interface and the GPIO settings.

#### Selecting the Communication Interface

The upper left section of the **GPIO Configuration** window (shown in Figure 39) is used to select the appropriate communication interface for the GPIO transceiver. Select **GPIO** from the **Interface** drop-down menu.

Interfa	ce Setti	ngs	
Protocol	GPIO	•	
Interface	GPIO	•	959-039
			2

Figure 39. Selecting the Communication Interface

#### Selecting the GPIO Settings

The **GPIO Settings** box (shown in Figure 40) is used to configure the GPIO pins.

GPIO 1	Direction:	0	•	GPIO PIN :	PD6	•	Value :	0	•
GPIO 2	Direction:	0	•	GPIO PIN :	PD6	•	Value :	0	•]
GPIO 3	Direction:	0	•	GPIO PIN :	PD6	•	Value :	0	•
GPIO 4	Direction:	0	•	GPIO PIN :	PD6	•	Value :	0	•
GPIO 5	Direction:	0	•	GPIO PIN :	PD6	•	Value :	0	•
GPIO 6	Direction:	0	•	GPIO PIN :	PD6	•	Value :	0	•

Figure 40. GPIO Settings Box

Configure the device using the following controls:

- **Direction** boxes: Select whether to configure each GPIO pin as an input or an output. Select 0 for input or 1 for output.
- GPIO PIN boxes: Select which GPIO pins to use.
- Value boxes: Select the binary value for each GPIO pin. By enabling multiple GPIO pins (via the GPIO 1 to GPIO 6 check boxes), you can use multiple GPIO pins together. Confirm your settings by clicking Use Changes near the bottom of the window.

										RS232 UART3
Interface S	Setting	gs								LVDS SPORT2
Protocol GPI0		*								SPI
Intendce GPIO		~								SPIO
GPI0 Settings GPI0 1 Direct	on: 0	~	GPIO PIN :	PD6	v	Value :	0	~		CAN
GPI0 2 Direct	on: 0	~	GPIO PIN	PD6	~	Value :	0	~		
GPI0 3 Direct	on: 0	¥	GPIO PIN :	PD6	*	Value :	0	~		
GPI0 4 Direct	on: 0	~	GPIO PIN :	PD6	~	Value :	0	~		
GPI0 5 Direct	on: 0	~	GPIO PIN :	PD6	~	Value :	0	~		
GPI0 6 Direct	on: 0	*	GPIO PIN :	PD6	~	Value :	0	*		
	<c< td=""><td>lick 1</td><td>To Cor</td><td>figur</td><td>e Int</td><td>terfac</td><td>e&gt;</td><td></td><td></td><td></td></c<>	lick 1	To Cor	figur	e Int	terfac	e>			
	Interface S Protocol GPI0 Interface GPI0 GPI0 Settings GPI0 1 Directi GPI0 3 Directi GPI0 4 Directi GPI0 5 Directi GPI0 6 Directi	Interface Setting Protocol GPI0 Interface GPI0 GPI0 Settings GPI0 1 Direction: 0 GPI0 2 Direction: 0 GPI0 4 Direction: 0 GPI0 5 Direction: 0 GPI0 5 Direction: 0 GPI0 6 Direction: 0 GPI0 6 Direction: 0	Interface Settings Protocol GPI0  Interface GPI0  GPI0 I Direction: 0  GPI0 2 Direction: 0  GPI0 3 Direction: 0  GPI0 4 Direction: 0  GPI0 5 Direction: 0  GPI0 6 Direction: 0  CClick	Interface Settings Protocol GPI0  Interface GPI0 Interface GPI0 GPI0 Settings GPI0 1 Direction: 0 GPI0 PIN GPI0 3 Direction: 0 GPI0 PIN GPI0 4 Direction: 0 GPI0 PIN GPI0 5 Direction: 0 GPI0 PIN GPI0 6 Direction: 0 GPI0 PIN CPI0 6 Direction: 0 GPI0 PIN CPI0 6 Direction: 0 GPI0 PIN	Interface Settings Protocol GPI0  Interface GPI0  GPI0 Settings GPI0 1 Direction: 0  GPI0 PIN: PD6 GPI0 3 Direction: 0  GPI0 PIN: PD6 GPI0 4 Direction: 0  GPI0 PIN: PD6 GPI0 5 Direction: 0  GPI0 PIN: PD6 GPI0 6 Direction: 0  GPI0 PIN: PD6 GPI0 6 Direction: 0  GPI0 PIN: PD6	Interface Settings Protocol GPI0 V Interface GPI0 V Interface GPI0 V GPI0 Settings GPI0 1 Direction: 0 V GPI0 PIN P06 V GPI0 3 Direction: 0 V GPI0 PIN P06 V GPI0 4 Direction: 0 V GPI0 PIN P06 V GPI0 5 Direction: 0 V GPI0 PIN P06 V GPI0 6 Direction: 0 V GPI0 PIN P06 V Click To Configure Interview	Interface Settings Protocol GPI0 V Interface GPI0 V Interface GPI0 V GPI0 Settings GPI0 1 Direction: 0 GPI0 PIN: P06 Value: GPI0 3 Direction: 0 GPI0 PIN: P06 Value: GPI0 4 Direction: 0 GPI0 PIN: P06 Value: GPI0 5 Direction: 0 GPI0 PIN: P06 Value: GPI0 6 Direction: 0 GPI0 PIN: P06 Value: GPI0 6 Direction: 0 GPI0 PIN: P06 Value: GPI0 6 Direction: 0 GPI0 PIN: P06 Value:	Interface Settings Protocol GPI0  Interface GPI0  GPI0 Setting: GPI0 1 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 3 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 4 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 PIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0 GPI0 5 Direction: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 FIN: PD5 Value: 0  GPI0 5 Direction: 0  GPI0 FIN: PD5 Value: 0  GPI0 FIN: PD5 Va	Interface Settings	Interface Settings Protocol GPI0 V Interface GPI0 V GPI0 Settings GPI0 1 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 3 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 4 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 5 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 5 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 5 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 6 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 6 Direction: 0 GPI0 PIN: PD6 Value: 0 V GPI0 6 Direction: 0 GPI0 PIN: PD6 Value: 0 V Click To Configure Interface>

Figure 41. GPIO Configuration Window

## **UPDATING THE FIRMWARE**

To update the embedded software version on the evaluation board,

- Configure Windows Firewall to allow the use of FTPServer.exe on your computer.
  - a. If using Windows XP or Windows Vista, click Start > Control Panel > Windows Firewall > Exceptions > Add Program and select FTPServer.exe from the list of programs and services (see Figure 42).
  - b. If using Windows 7, click Start > Control Panel (All Control Panel Items) > Windows Firewall > Allow a program or feature through Windows Firewall > Change settings. A dialog box opens asking if you want to allow Windows Firewall to make changes to the computer. Click Yes, and then click Allow another program and select FTPServer.exe from the list of programs and services (see Figure 42).





 Configure the application to connect to the correct IP address. From the IP address drop-down box in the PASV Settings section (see Figure 43), select the IP address of the adapter used to connect the board to the PC.

General Tasks 🛞	Configuration -			
🞝 Show Server Log	General			
Show User Accounts Show Configuration	General ETP port	21	Connection (imeout (in minutes)	5
👩 Show Online Users	Max. gonnections:	10	Mag. connections per IP.	3
Configuration Tasks	Welcome message:	Welcome to Quick 'n E	any FTP Server	× 15
🕼 Show Configuration Help 🕼 Frequently Adled Questions	Goodbye message:	Вуе		
	Startup settings Launch FTP Ser Automatically act	ver at Windows startup tivale server at startup d in systemitray	Logging Loglevet	Епог 💌
	PASV settings IP address: Port range:	1921(S0210) Default Get IP from remote car		Configure .

Figure 43. Update Firmware

- 3. Click **Show User Accounts** and change the path to the directory containing the uImage file to be downloaded to the *ez*LINX hardware platform.
- 4. Click **Start** near the top left of the application to begin running the FTP service.
- 5. In the **IP address** box of the **Quick 'n Easy FTP Server** window, enter the IP address of the server that contains the needed uImage, and then click **Configure**.
- 6. The software application then prompts you to restart the application (see Figure 44). Click **OK**.

		×
Please re	start the P	C Application
ſ	ОК	

Figure 44. Restart Application to Update the Embedded Software Version

- 7. Close the application and wait approximately 2 minutes to 3 minutes for the application to erase the hardware platform and download and extract the new firmware version. (Warning: Do not reset the *ez*LINX hardware until the transfer is complete, which is indicated when the data sent counter near the bottom right of the **Quick 'n Easy FTP Server** application window equals the size of the update file.)
- 8. Reset the board and restart the PC application.
- Confirm that the firmware update was downloaded correctly by opening the Board Configuration window and clicking Check, located next to the Embedded Version box (see Figure 3 and Figure 4). The new version should be displayed.

## TRANSCEIVERS STATUS WINDOW

The **Transceivers Status** window (see Figure 45) allows monitoring the status of each interface as it transmits data. To access the **Transceivers Status** window, click the word **STATUS** at the bottom left of the application window. The enabled transceivers are indicated by the check boxes in the **Transceivers Enable** section.

The amount of data sent and received through each interface during the current session is also shown; this information is displayed in the **Transfer status** section of the window. Each **TX** box displays the amount of kilobytes of data that has been sent through the corresponding transceiver, and each **RX** box displays the amount of kilobytes of data that has been received. These values do not update in real time; therefore, to refresh the values, you must close and reopen the **Transceivers Status** window.

Transceivers Enable	Transfer status				
RS485	RS485	TX:	Kbytes	RX:	Kbyte
RS232	RS232	TX:	Kbytes	RX:	Kbyte
CAN 0	CAN 0	TX:	Kbytes	RX:	Kbytes
LVDS	LVDS	TX:	Kbytes	RX:	Kbytes
🗌 I2C	12C	TX:	Kbytes	RX:	Kbytes
SPI SPI	SPI	TX:	Kbytes	RX:	Kbytes
GPIO	CAN1	TX:	Kbytes	RX:	Kbytes
CAN1					

Figure 45. Transceiver Status Window

# UG-461

## TRANSCEIVERS CONFIGURATION WINDOW

To access the **Transceivers Configuration** window (see Figure 48), click **View Configuration** near the bottom of any window in the application software. The **Transceivers Configuration** window allows viewing the global configuration of all transceivers and GPIOs on the *ez*LINX hardware platform. From this window, you can see which transceivers are enabled, what hardware routing is active, and the settings of each individual interface. The configuration cannot be modified from this window; however, the application supports loading and saving a configuration as an .xml file.

# TRANSCEIVERS CONFIGURATION WINDOW BUTTONS AND OPTIONS

To save the current configuration, click **Save** and choose a name. To load a previously saved configuration, click **Load** and select the desired .xml configuration file.

The **Transceivers Configuration** window has three buttons (**Load**, **Save**, and **Close**) at the bottom left of the window that enable the functionalities described in this section.

#### Load Button

Click **Load** to load an existing configuration from an .xml file. The **LoadInterface** dialog box appears (see Figure 46).

In the **LoadInterface** box,

- Clicking **Ok** loads the configuration.
- Clicking **Cancel** cancels the load, and the **LoadInterface** window closes.

Note that before a configuration is loaded, you must exit the **Transceivers Configuration** window by clicking **Close** from

the **File** menu. Exiting the window by clicking the **Close** button (represented by a red box with an X) in the upper right corner of the window does not allow a configuration to be loaded.

🖳 LoadInte	face 🖂	• 🔀
I		Browse
Ok	Cancel	

Figure 46. Loading an Existing Configuration to the Board

#### Save Button

Click Save to save the current configuration to a new .xml file.

#### **Close Button**

Click **Close** to close the **Transceivers Configuration** window. A confirmation message appears, asking whether you want to save the current configuration.

		ß
Do you want to	save the configur	ation or no?
	Ves	No

Figure 47. Asking Whether to Save the Current Configuration Before Closing the Transceivers Configuration Window

# Software User Guide

RS485 Transceiver Configuration						RS232 Tra	RS232 Transceiver Configuration						
Protocol	RS485	Transceiver Enable				Protocol	Pertocol RS232			Transceiver Enable			
Interface	RS-485 Settings					Interface	HAR	12	RS-232 Settings				
Intellace OPATT2		Paul Data 110		Word Size	7	Pauline/Con	-	12	Baud Rate	110	Word Size	7	
Houting(Ho	m)	bauu nate	None	11010 3020	None	Fouting(Pror	n)			none	Flow Contr	nol None	
LIADTO	- HART2	Parity	-	How Control		L Diable	11.125	UADTO	Panty	1	-		
UARIZ	To WARTZ	Stop Bits	1	Duplex	Half Duplex	UART2	To	UMATZ	Stop Bits	1 m			
I2C Transc	eiver Configuration	,				SPI Transe	ceiver (	Configuratio	on	-			
Transcoluter Enable Proto		al 12C		Client	10	Protocol	J SPI		CPI Settions				
- Honoce	Inter Engliste Friede	TWI1	1	David Date	100	Interface	SPID	_	SFI Settings	2000000	Operatio	on Mode	
	intena	00		baug hate	Clause	Deution (Fee			Max Speed	200000	🔿 Mas	ter Mode	
				Mode	Jove	Finable			LOD FIRS	U	Q Qau	e Mode	
				Write Read Flag	Write	CPID	Te	010	Word Size	8	Char	2	
						Lot to		5110	U.		Cridit		
Interface CAN Settin Bit Rate Error CAN_EF CAN_EF CAN_EF CAN_EF CAN_EF	CAND CAND TI25000 Filter RR_TX_TIMEOU RR_CRTL RR_PROT RR_PROT RR_TRX RR_ACK	Filter 1 Mask 1 Filter 2 Mask 2 Filter 3 Mask 3 Mask 3 Mask 3	Normal 0 0 0 0 0 0		CAN0	Protocol Interface LVDS Set Baud Baud Word Frame	LVD SPC ttings Rate Tx Rate Tx Rate Rx Size Delay	IS DRT2 2000000 2000000 8 0		SPOR Active Low Internal Clock Internal Clock Secondary C	T2 To Tx k Rx hannel Enable	SPORT2 0 0 0 0	
GPIO GPIO1		GPIO2		GPIO3		GPIO4	_	G	PIO5		GPIOS		
GPIO 1		GPIO 2		GPIO 3		GPIO 4	GPIO 4		GPIO 5		GPIO 6		
Direction:	0	Direction:	0	Direction:	0	Direction:	0		Direction: 0	_	Direction:	0	
	PD6	GPIO PIN :	PD6	GPIO PIN :	PD6	GPIO PIN :	PD6	G	PIO PIN : PD	6	GPIO PIN :	PD6	
GPIO PIN :							-	100					
GPIO PIN : Value :	0	Value :	0	Value :	0	Value :	0	V	alue: n		Value :	0	

Figure 48. Transceivers Configuration Window

# UG-461

# NOTES

# NOTES

## NOTES

I<sup>2</sup>C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

#### Legal Terms and Conditions

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners. Information contained within this document is subject to change without notice. Software or hardware provided by Analog Devices may not be disassembled, decompiled or reverse engineered. Analog Devices' standard terms and conditions for products purchased from Analog Devices can be found at: http://www.analog.com/en/content/analog\_devices\_terms\_and\_conditions/fca.html.

©2013 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners. UG10959-0-12/13(0)



www.analog.com

Rev. 0 | Page 32 of 32