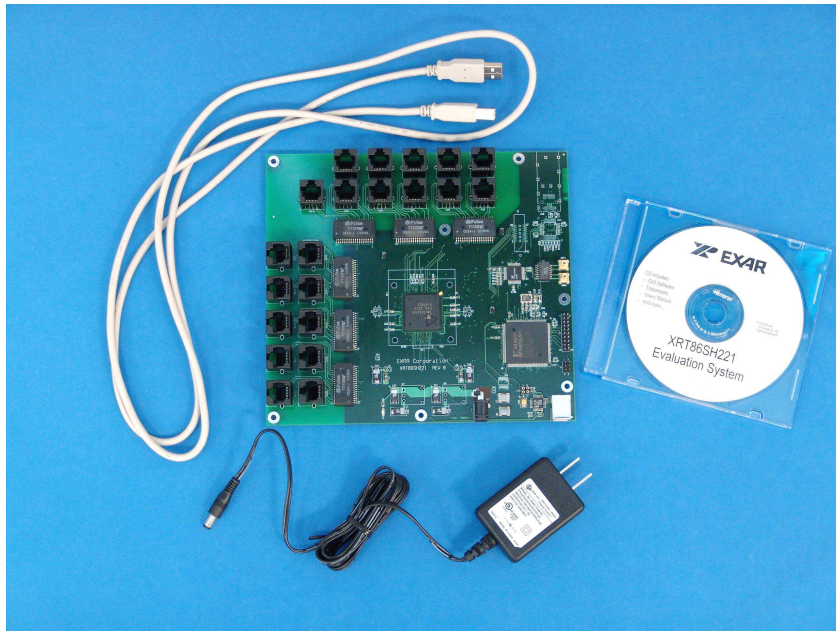


# XRT86SH221ES

## Evaluation System User Manual





## TABLE OF CONTENTS

<b>SECTION 1 Introduction</b> .....	<b>3</b>
<b>SECTION 2 Circuit Overview</b> .....	<b>3</b>
Figure 1. Simplified Block Diagram of the XRT86SH221ES .....	<b>3</b>
<b>Section 3 Schematic Design</b> .....	<b>4</b>
Figure 2. XRT86SH221 Schematic Page 1 .....	<b>4</b>
Figure 3. XRT86SH221 Schematic Page 2.....	<b>5</b>
Figure 4. XRT86SH221 Schematic Page 3.....	<b>6</b>
<b>Section 4 USB Installation</b> .....	<b>7</b>
<b>Section 5 Software Configuration</b> .....	<b>14</b>
<b>Section 6 Python Script Interface</b> .....	<b>18</b>

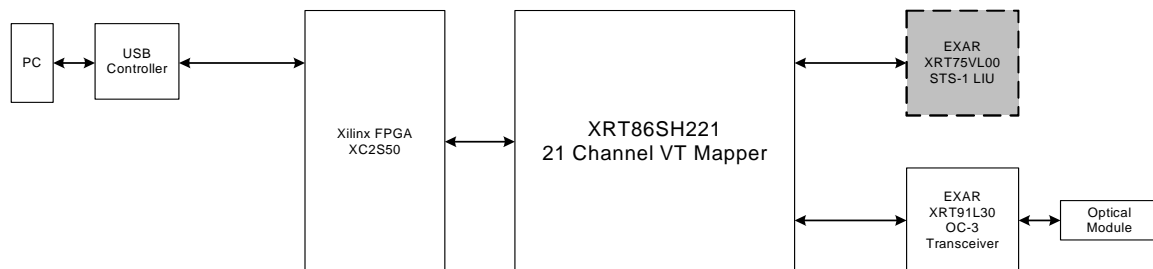
## SECTION 1 Introduction

This XRT86SH221ES is a complete evaluation kit for characterizing the 21 Channel VT Mapper IC. The evaluation kit includes the following:

1. Evaluation Board
2. USB Cable
3. AC Adaptor Power Cord
4. CD Rom
5. Documentation

## SECTION 2 Circuit Overview

This evaluation board is controlled by a graphical user interface (GUI) through a USB port from a typical PC. (Quick Note: The software supports Windows XP and NT down to most versions of Windows 98) Figure 1 shows that there are two separate data paths supported by the evaluation board. It is important to note that only one data path can be selected at any time due to the board architecture (not the DUT). The two data paths are the Telecom Bus (Connected to the XRT91L30 OC-3 Transceiver) or the Serial Interface (Connected to the XRT75VL00 STS-1 LIU). If the part is configured in STM-0, the Serial Interface can be used. If the part is configured in STM-1, then only the Telecom Bus can be used. There are two separate FPGA configuration files ("TelecomBus.hex" and "SerialInterface.hex") that come with this kit.



**Figure 1. Simplified Block Diagram of the XRT86SH221ES**

## Section 3 Schematic Design

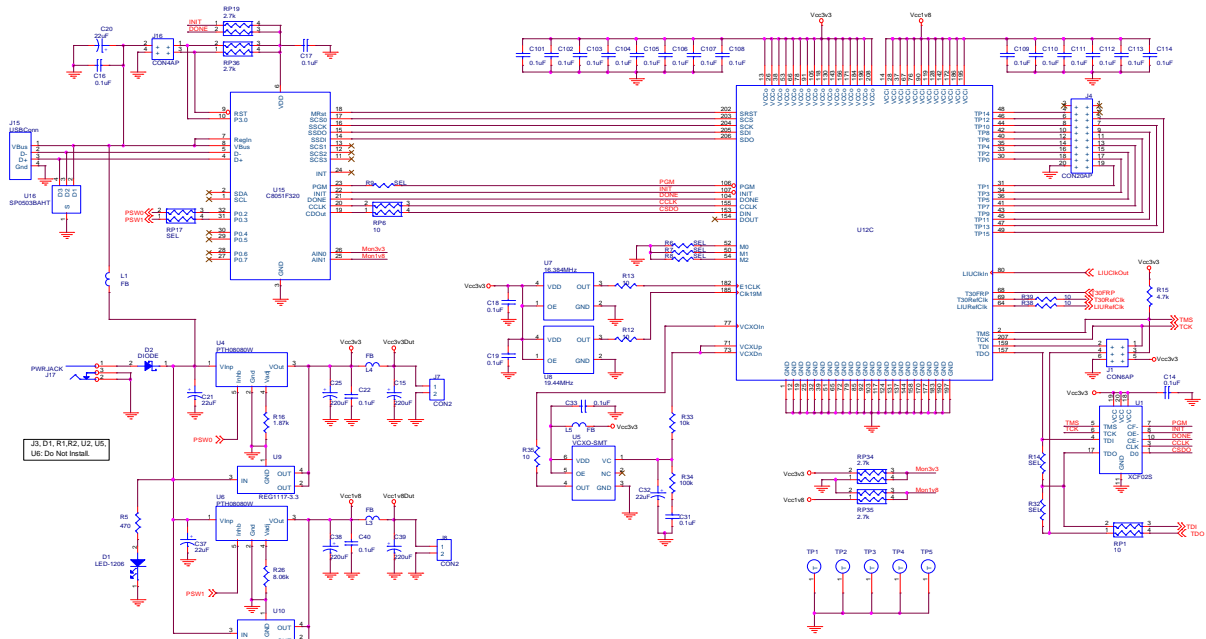


Figure 2. XRT86SH221 Schematic Page 1

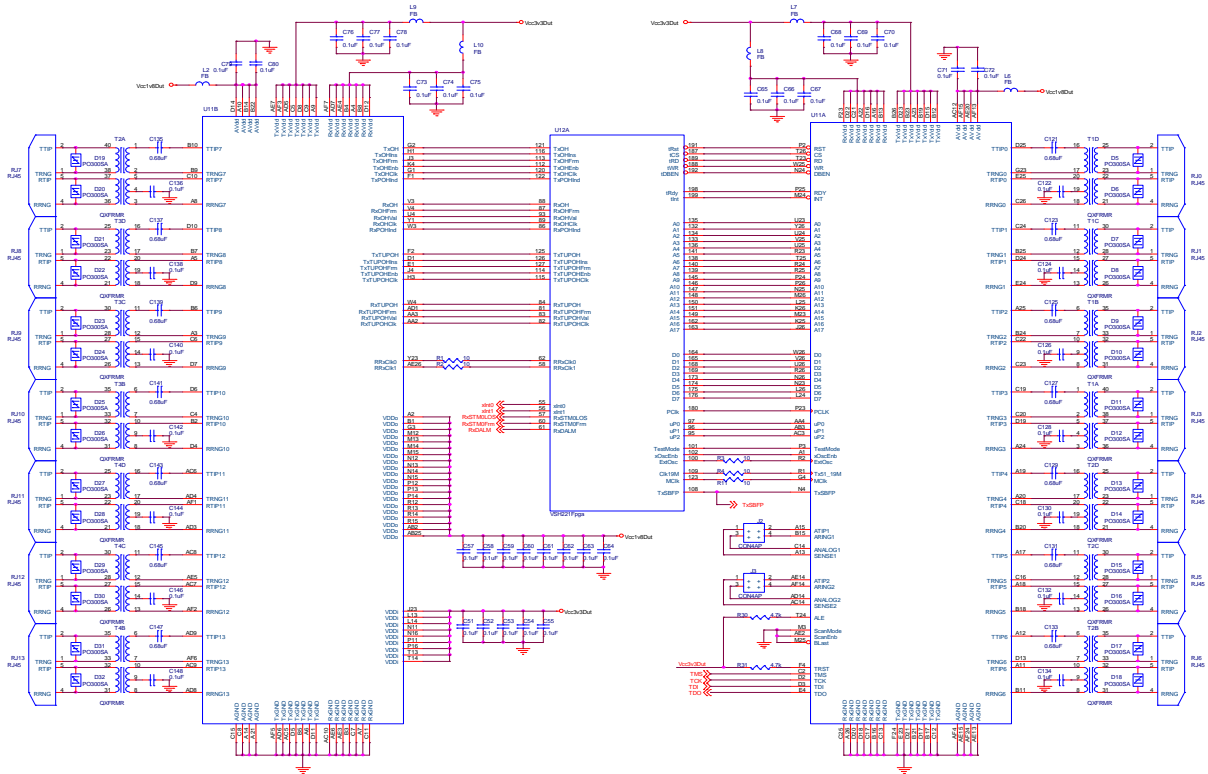


Figure 3. XRT86SH221 Schematic Page 2

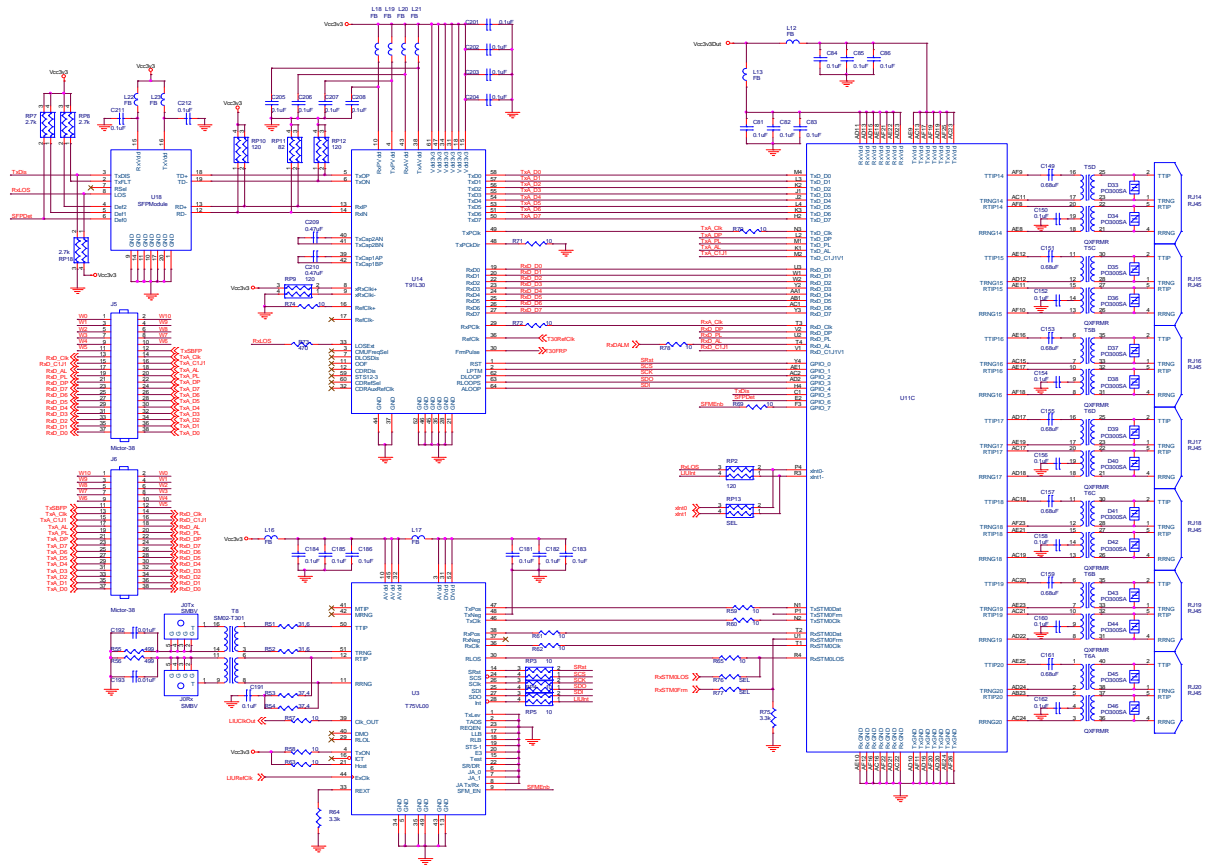
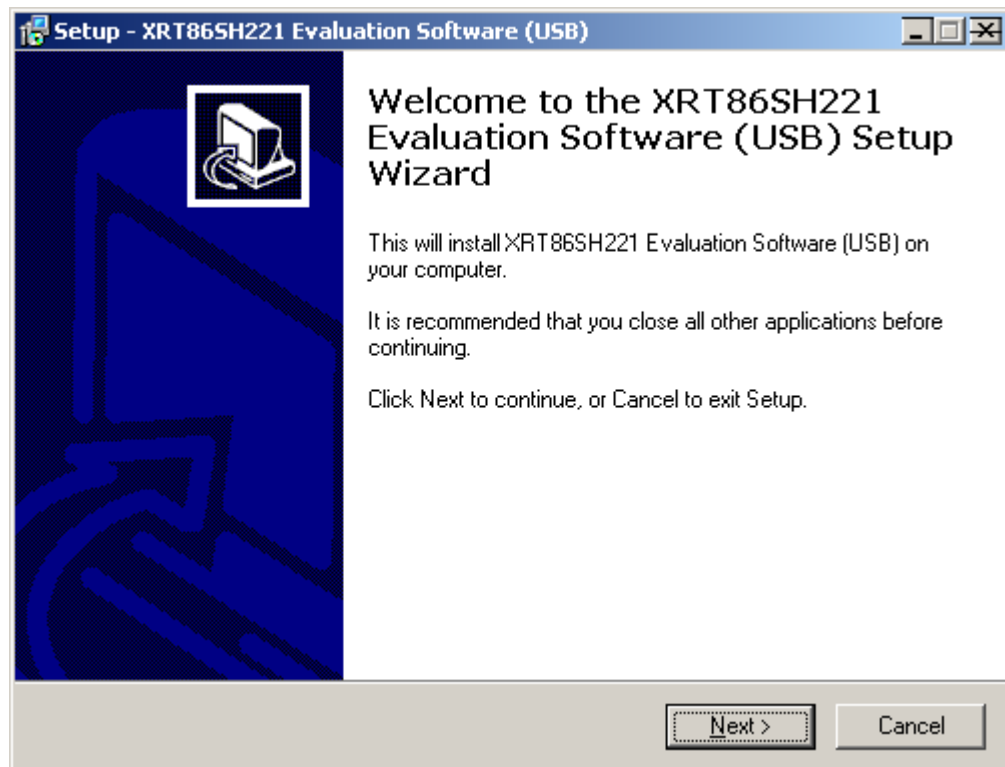


Figure 4. XRT86SH221 Schematic Page 3

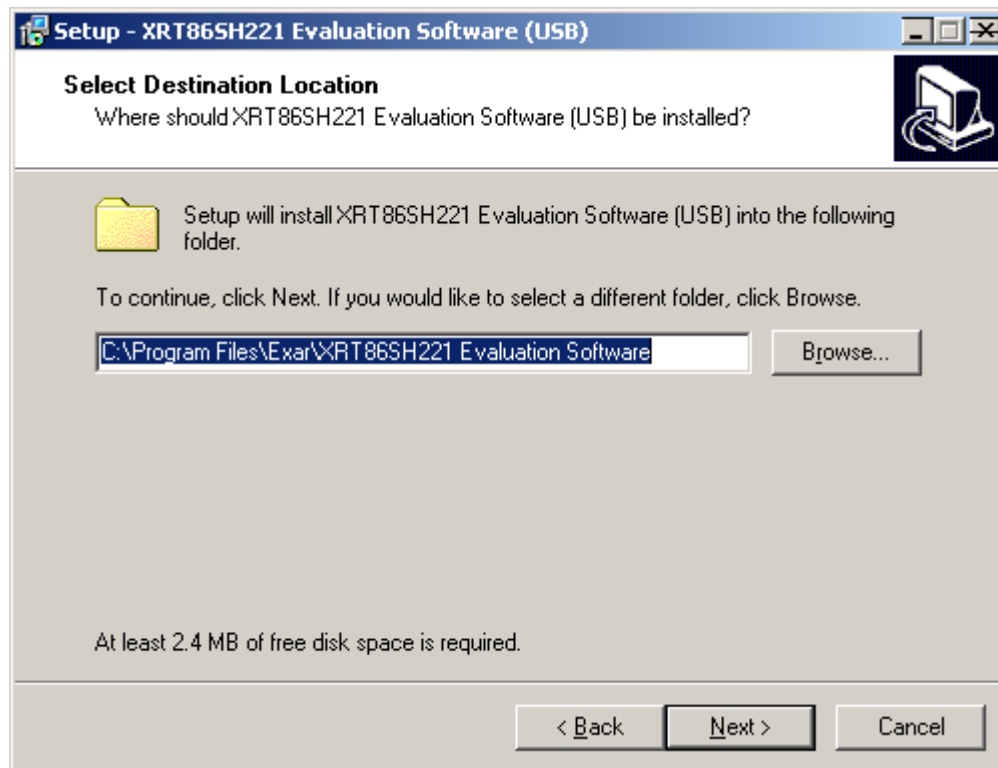
## Section 4 USB Installation

Included in the Evaluation System Kit, there is an Installation file on the CD Rom called "XRT86SH221EvaluationUSB.exe". This file should automatically load the Exar Usb drivers and the board will be up and running. However, if the following steps are performed and the board is NOT recognized by the PC, the last step in this process describes how to manually load the drivers.

**Step 1** Double Click the "XRT86SH221EvaluationUSB.exe" file to bring up the following screen. Click on the **Next>** button.

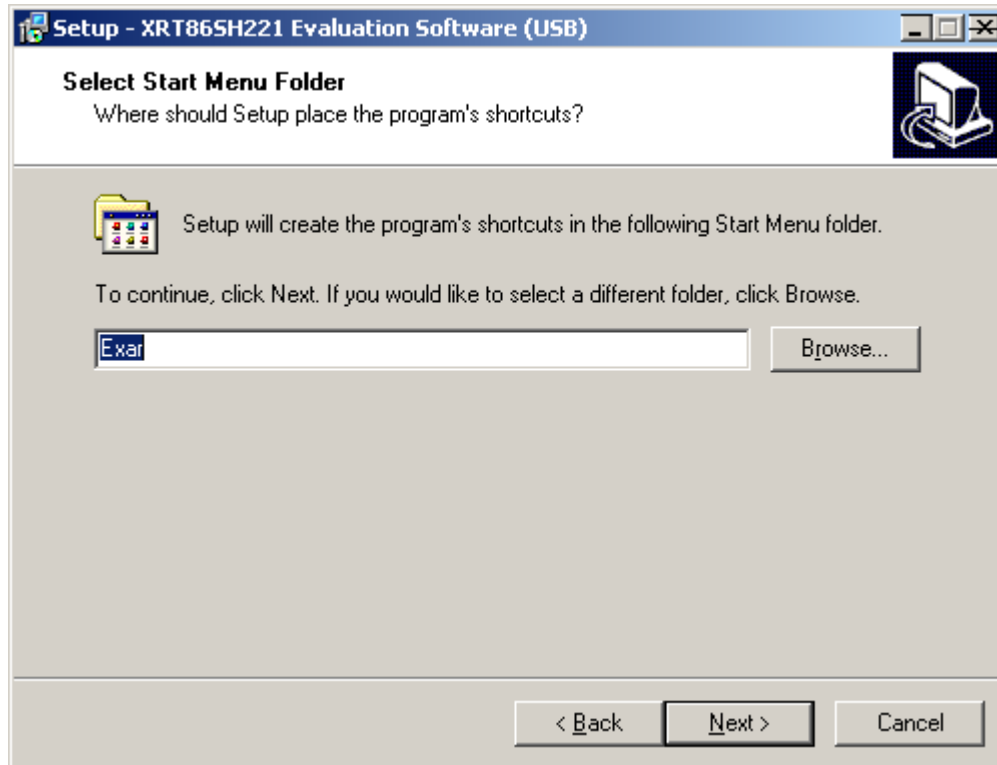


**Step 2** Exar recommends using the default directory, but the files can be installed to any directory that you choose. *Note: Make sure that all files from the CD are properly copied to the unique directory if it is changed.* Click on the **Next>** button to proceed.

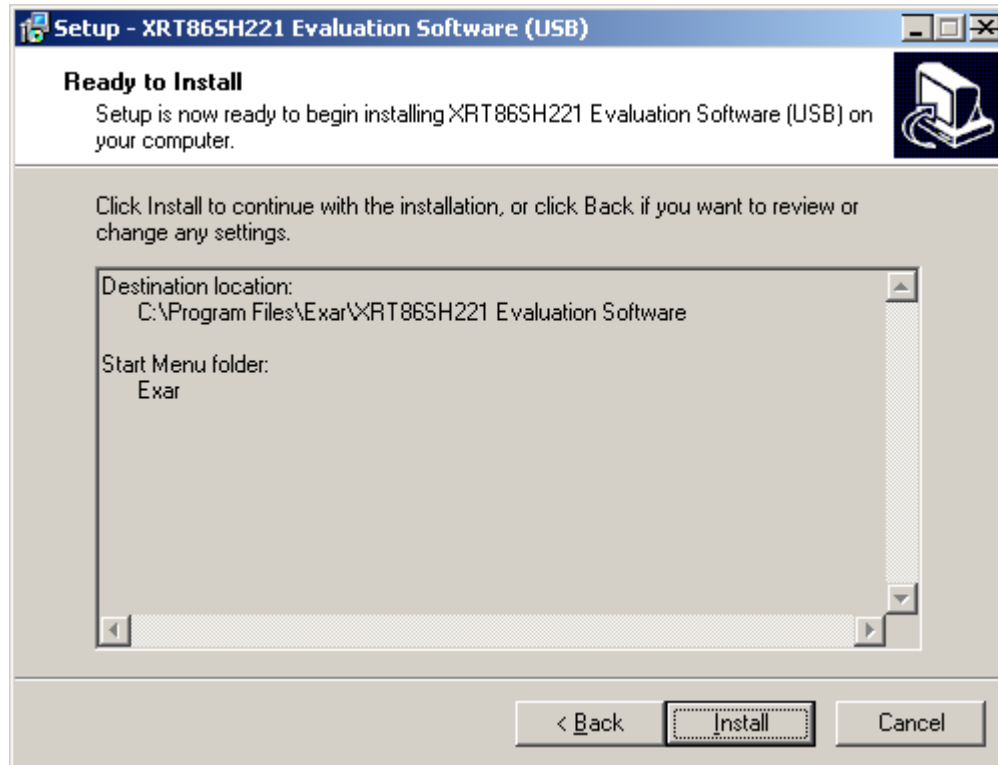




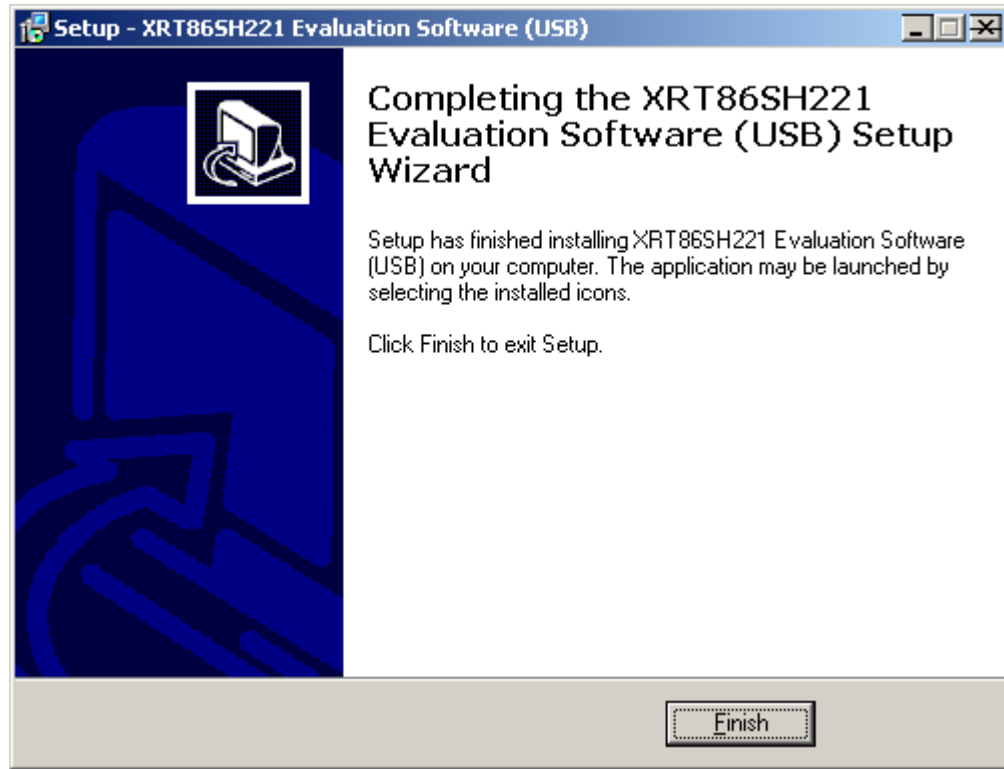
**Step 3** Click the Next> button to continue.



**Step 4** Click on the Install button to continue installation.



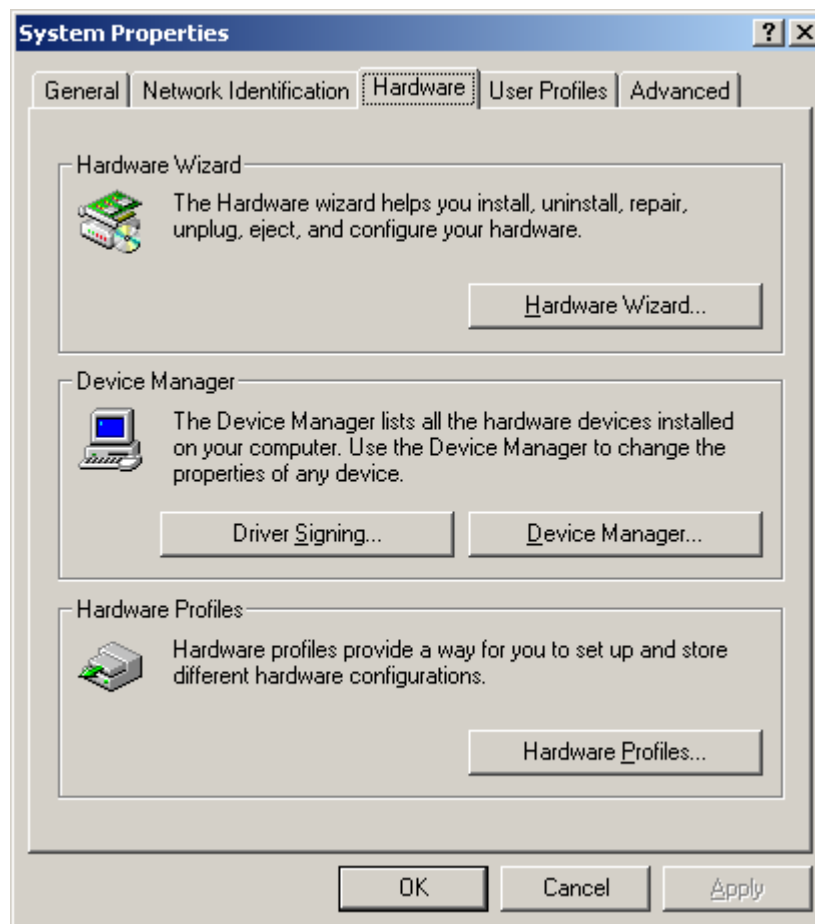
**Step 5** Click on the Finish button to complete installation. The Exar USB drivers should be properly installed and ready to go. If so, continue to the next section that describes the Graphical User Interface. If not, see the last step below to manually load the Exar USB Drivers.



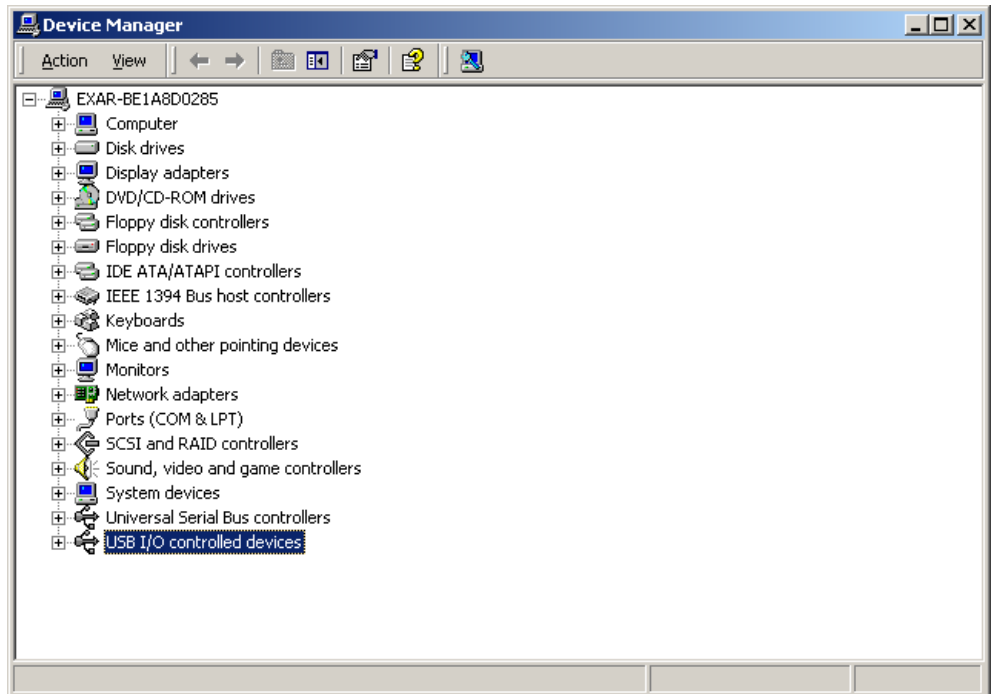
### Manually Loading Exar Drivers (If Installation is NOT Successful)

If the Evaluation Board is NOT recognized and/or the installation process was successful, most likely there is a driver contention or the driver was not installed in the proper directory from the software. This section describes how to manually load the drivers.

Note: Each Windows operating system may access the System Hardware differently. Therefore, it is not practical to describe this procedure for each OS. However, the key component is usually the Control Panel of the PC. From the Control Panel, you can access the System Properties dialog box. From there, choose the Hardware menu item and go to the Device Manager...



The device Manager will bring up the following dialog box. Choose the USB I/O Controlled Devices, select Exar Device, right click, select properties, select Update Driver, and then follow the directions to choose the driver from the CD or the /Driver directory in the Exar working directory (“C:\Program Files\Exar\XRT86SH221 Evaluation Software\Drivers” by default).



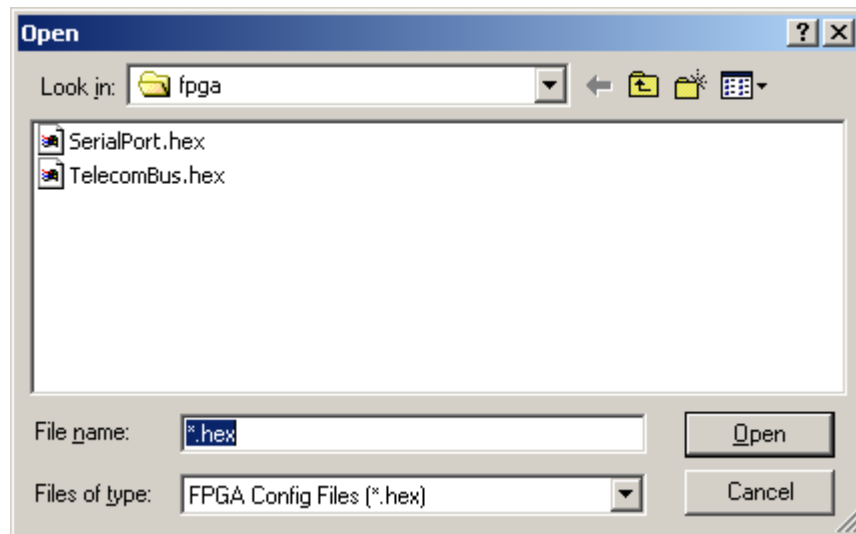
## Section 5 Software Configuration

The GUI is intended to allow easy access to configure basic registers in Voyager Lite. There are too many registers to fit in an easy to follow software package. Therefore, any features NOT supported in the GUI can be executed through a simple Python Scripting Language that can be run directly through this interface. The Python Installer program comes on the Exar CD and can easily be installed by following the on screen directions. In addition, Exar has included common test scripts for Voyager Lite that can be used as a reference. In fact, the script files contain simple function calls that support a majority of the features offered. From there, the script files can easily be modified to accommodate specific requirements.

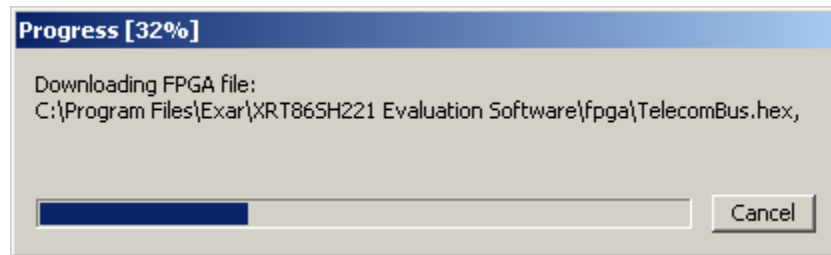
First, this section will run through the basic GUI and describe the modes of operation. Secondly, this section will describe how to run the provided python scripts.

### Initialization of the GUI

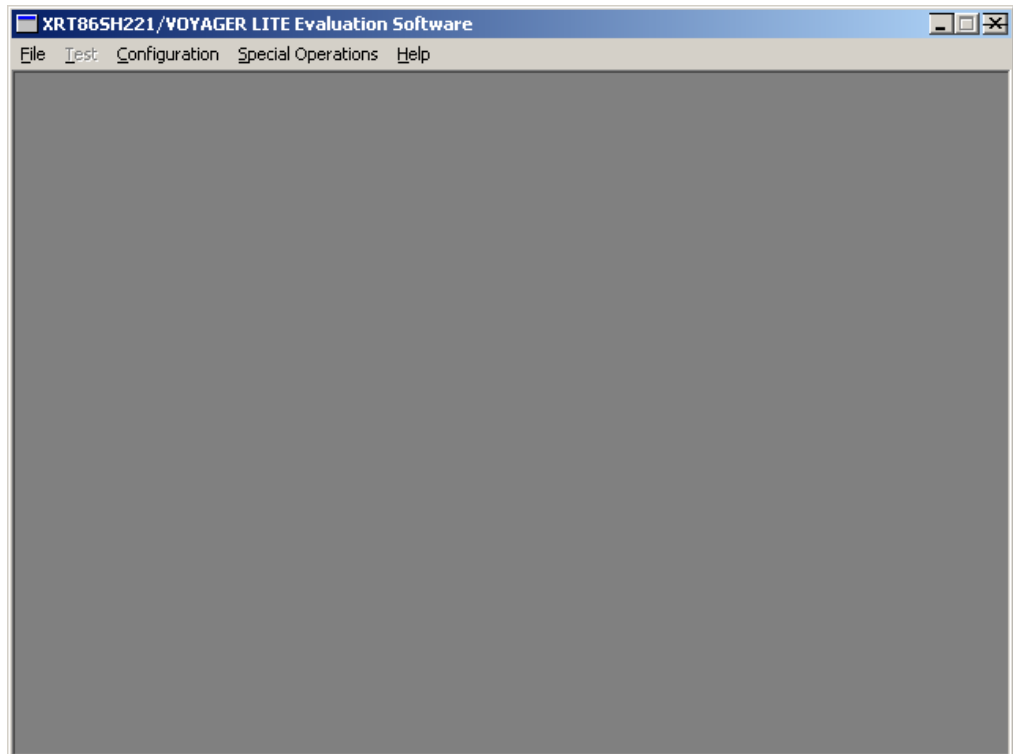
Once the USB Installer program is completed, an executable file called "xrt86sh221gui.exe" will be available in the working directory. Double Click this file to initiate the GUI. The following dialog box will appear. As mentioned earlier, there are two main configurations. If your Evaluation Board has the XRT91L30 installed, you must choose the "TelecomBus.hex" FPGA file. If your Evaluation Board has the XRT75VL00 installed, you must choose the "SerialPort.hex" file. Please select the appropriate file to continue.



After clicking on the Open button, the following dialog box will appear to notify the user that the FPGA file is being downloaded to the on board FPGA.



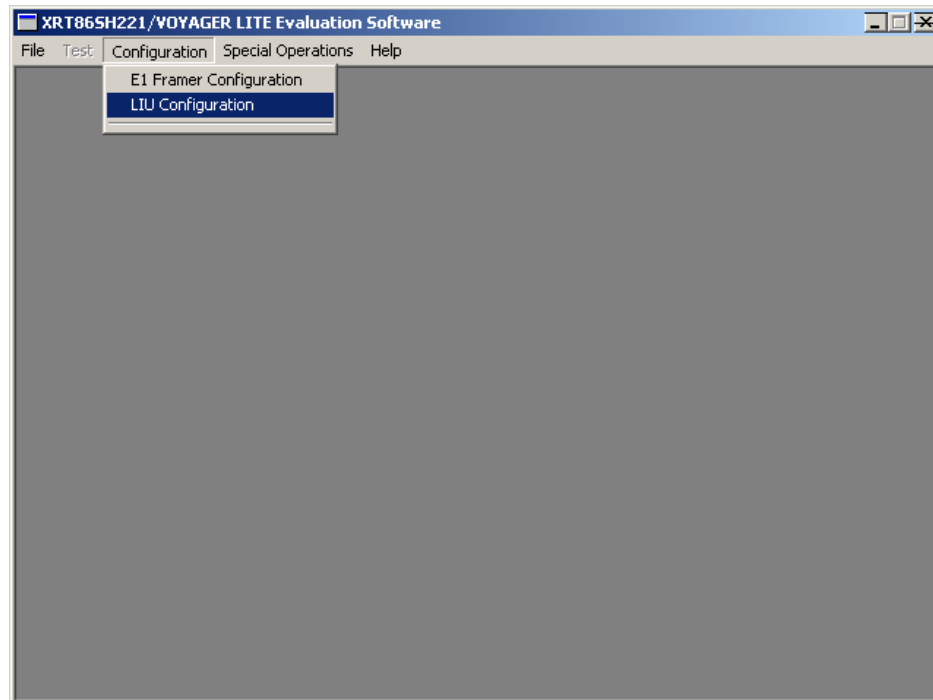
If there is an error loading the FPGA file, refer to the section that describes "Manually Loading Exar Drivers". If the FPGA has been successfully loaded, the following dialog box will appear.



There are two main selection items on the tool bar.

1. Configuration
2. Special Operations

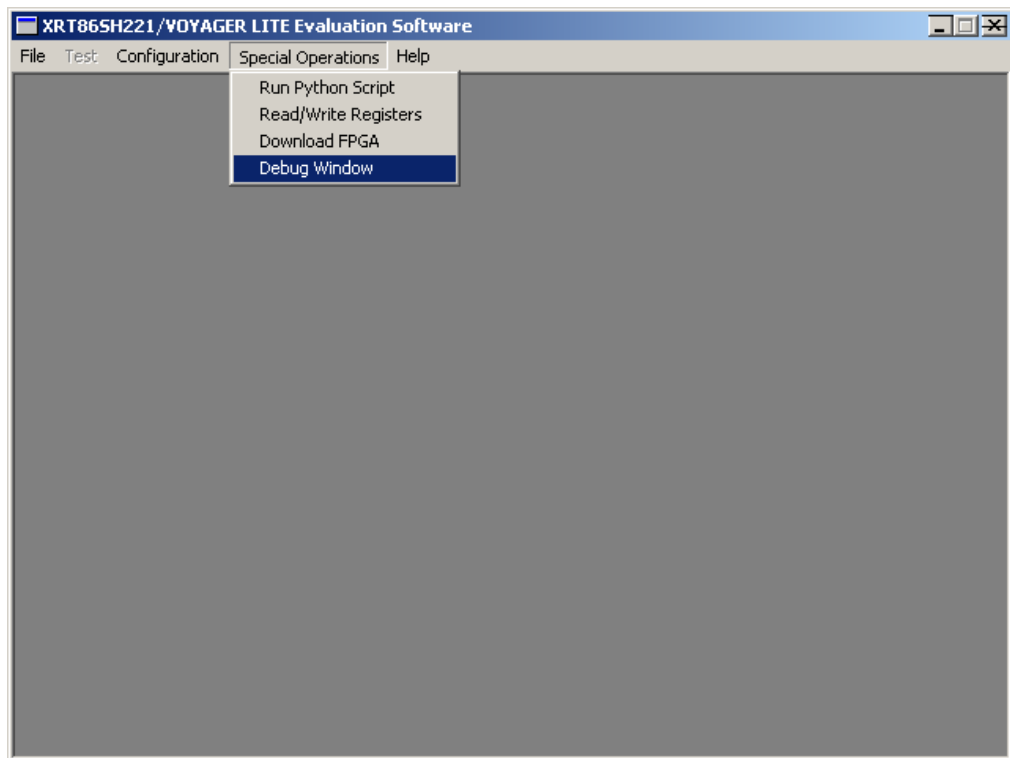
Under the Configuration menu, the following items will appear. You'll have the option to program the E1 Framer and the E1 LIU block registers.





Under the Special Operations menu, the following items will appear. You'll have the option to do one of the following:

1. Run Python Script (Script Files are Automatically Loaded)
2. Read/Write Registers (Allows Access to a Manual Read or Write)
3. Download FPGA
4. Debug Window (Interactive Tool to Monitor Register Activity)



## Section 6 Python Script Interface

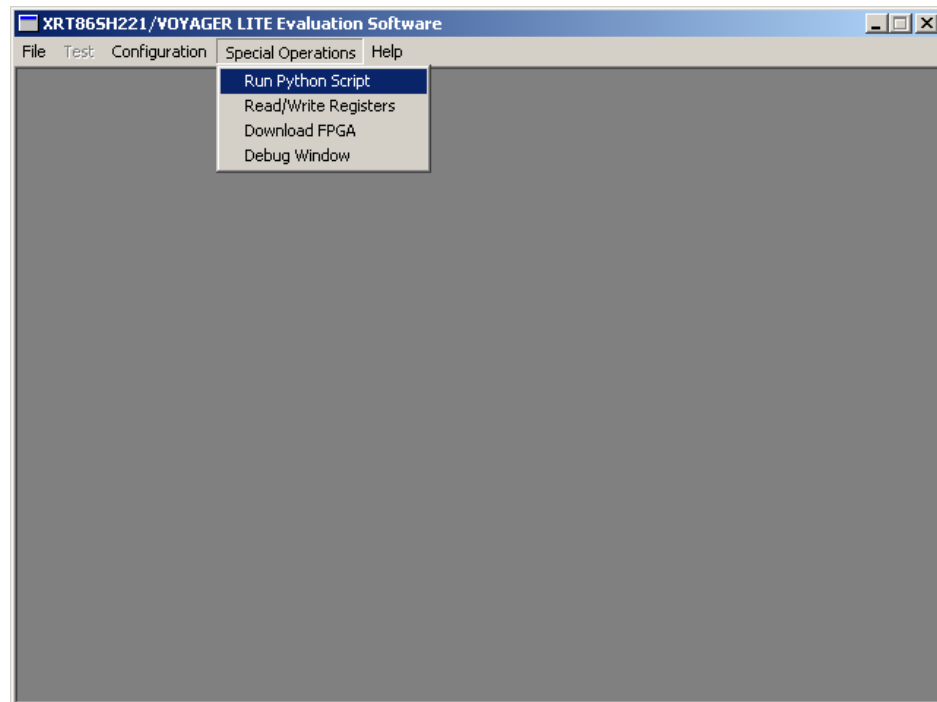
The Python Installer program is included on the Exar Evaluation System CD. When selected, the following dialog box will appear. Just follow the on screen instructions, and it will automatically be installed.



Once Python has been installed, return to the GUI to launch one of the python scripts that will allow you to perform function calls. Below is a brief description on this process.

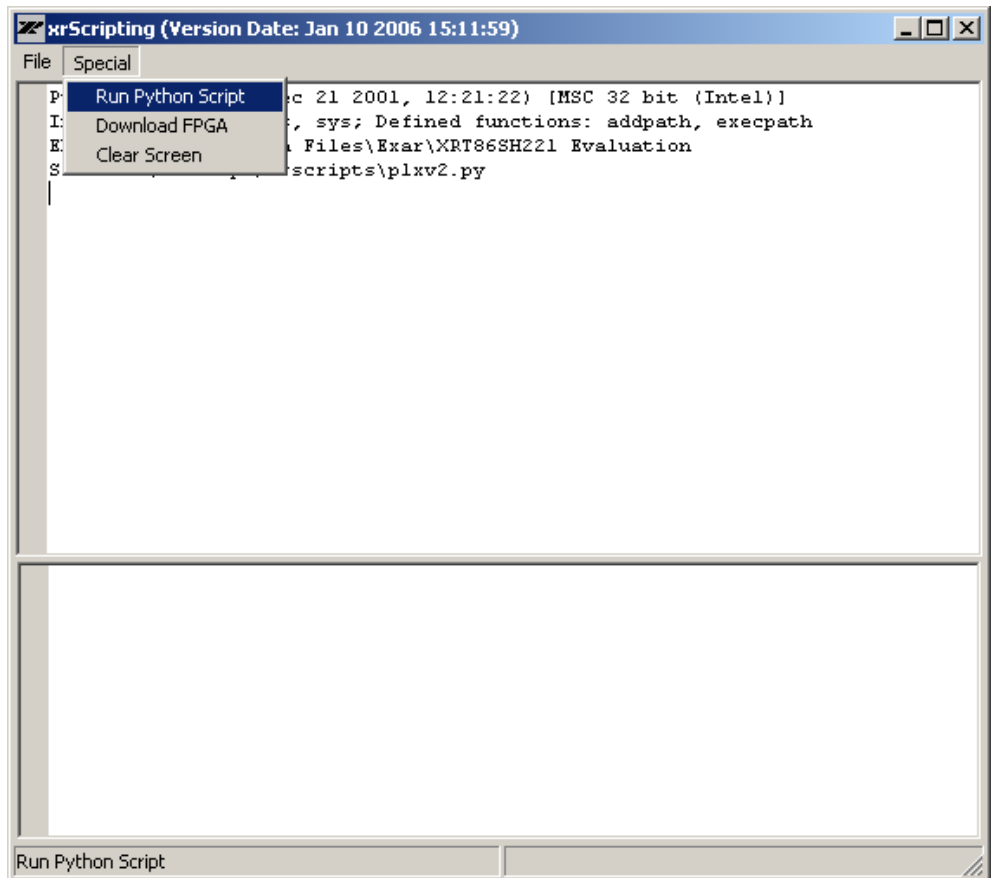
### Running a Python Script From the GUI

From the main GUI, select the Special Operations menu shown below and click on the “Run Python Script” to continue.

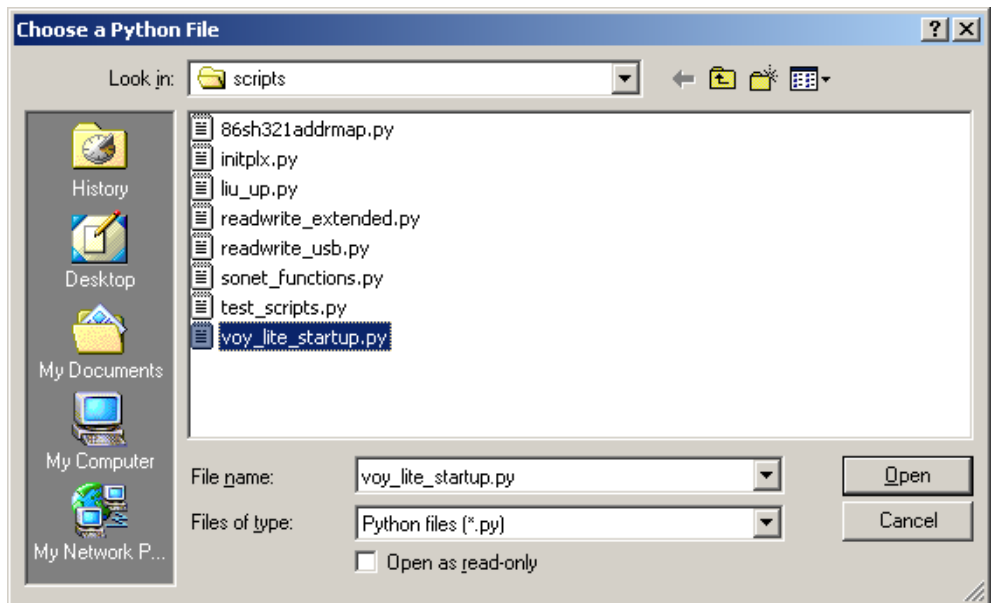


Once you select the “Run Python Script” option from the Special Operations menu from the GUI, the “xrScripting Interface” will pop-up as shown below. At this point, the xrScripting Interface is independent of the GUI and any script files that are run will automatically overwrite the GUI settings if there is an overlap with the registers. Therefore, caution must be used to keep track of the register configurations if one wants to toggle back and forth from the GUI to the xrScripting Interface. The next step is to use the “Special” menu on the tool bar to run the actual test script file.

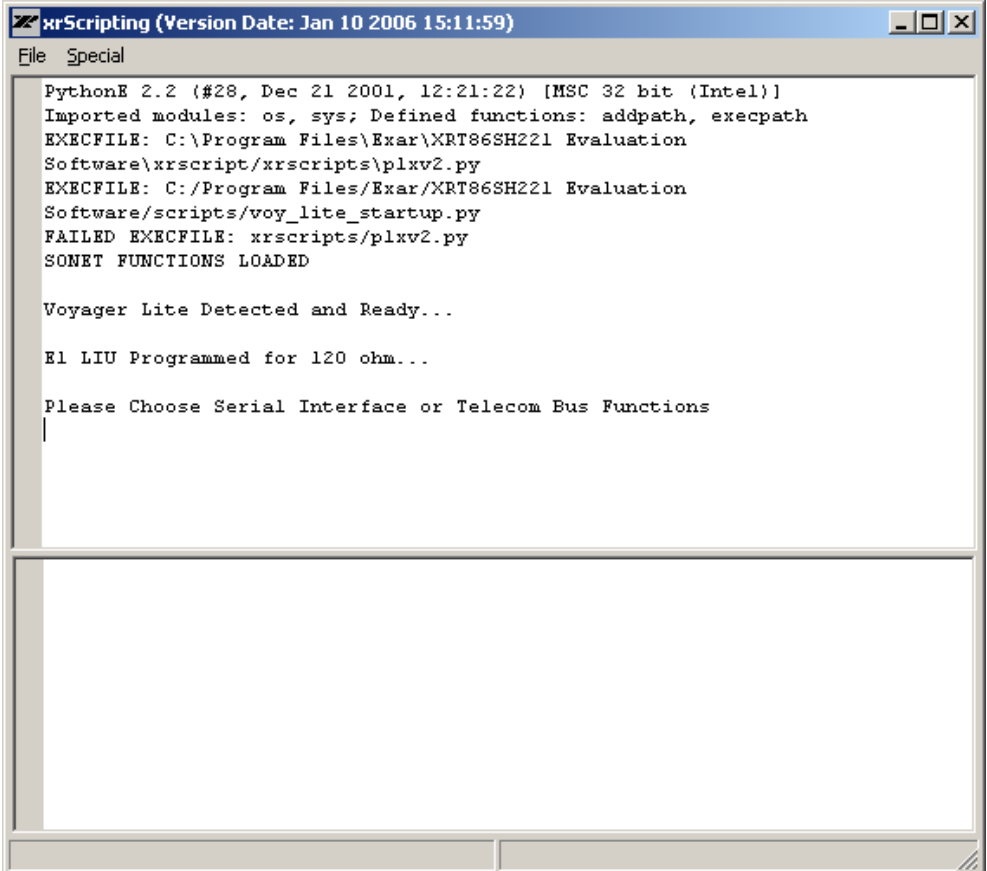
*Note: If you would like to view a particular test script in a plain text editor mode, use the “File” menu on the tool bar and select “Open”.*



Once you select the “Run Python Script” option from the “xrScripting Interface”, the following dialog box will appear. The first python script that should be run is the “voy\_lite\_startup.py” which will automatically run all the other script files for you. This way, you do NOT have to run each one individually. Once this file runs, you can open up any one of these files for reference. Select the “voy\_lite\_startup.py” script file and click Open.



After running the “voy\_lite\_startup.py” script and a brief pause, the following dialog box will appear. This is the main Python Window that consists of two main windows. The window on top is the Display Monitor, which as can be seen starts with the message “Voyager Lite is Detected and Ready...”. The bottom Window is the Command Console where you can perform manual Read/Write to registers and perform function calls. See the examples on the following pages.



```
PythonE 2.2 (#28, Dec 21 2001, 12:21:22) [MSC 32 bit (Intel)]
Imported modules: os, sys; Defined functions: addpath, execpath
EXECFILE: C:\Program Files\Exar\XRT86SH221 Evaluation
Software\xrscript\xrscripts\plxv2.py
EXECFILE: C:/Program Files/Exar/XRT86SH221 Evaluation
Software/scripts/voy_lite_startup.py
FAILED EXECFILE: xrscripts/plxv2.py
SONET FUNCTIONS LOADED

Voyager Lite Detected and Ready...

E1 LIU Programmed for 120 ohm...

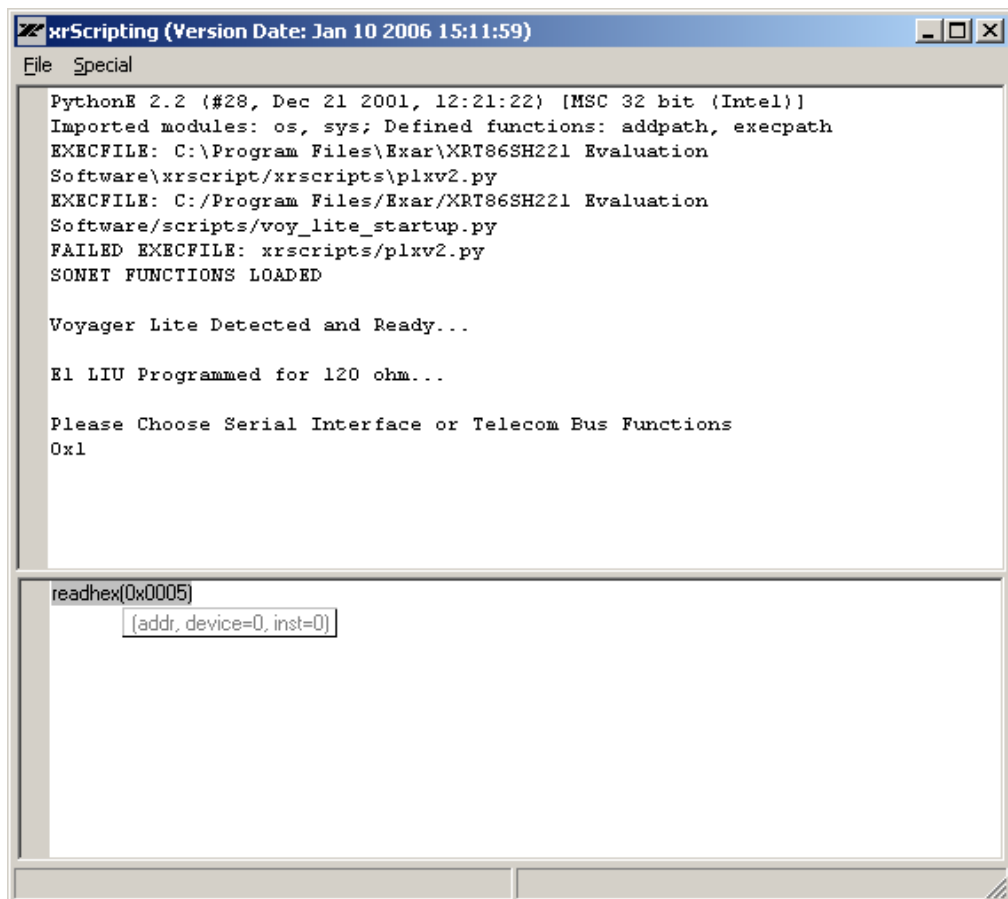
Please Choose Serial Interface or Telecom Bus Functions
|
```

### Example 1: Read the Revision ID Register

Looking at the dialog box below, you can see that the Command Console (lower window) was used to type the following command: `readhex(0x0005)`, which is the Revision ID register. You can also see that the value of `0x1` was returned in the Display Monitor (upper window), which corresponds to Revision A silicon.

### **Note 1: Critical: You must press Ctrl + Enter to execute a command from the Command Console!**

*Note 2: in the lower window, the command is highlighted in Grey. The Grey highlight notifies the user that the command has been executed. If the Grey highlight does not appear, the command (or function) is still being processed. Please wait for the Grey highlight before continuing.*



The screenshot shows a window titled "xrScripting (Version Date: Jan 10 2006 15:11:59)". The window is divided into two main sections. The upper section is a text area containing the following output:

```
PythonE 2.2 (#28, Dec 21 2001, 12:21:22) [MSC 32 bit (Intel)]
Imported modules: os, sys; Defined functions: addpath, execpath
EXECFILE: C:\Program Files\Exar\XRT86SH221 Evaluation
Software\xrscript\xrscripts\plxv2.py
EXECFILE: C:/Program Files/Exar/XRT86SH221 Evaluation
Software/scripts/voy_lite_startup.py
FAILED EXECFILE: xrscripts/plxv2.py
SONET FUNCTIONS LOADED

Voyager Lite Detected and Ready...

E1 LIU Programmed for 120 ohm...

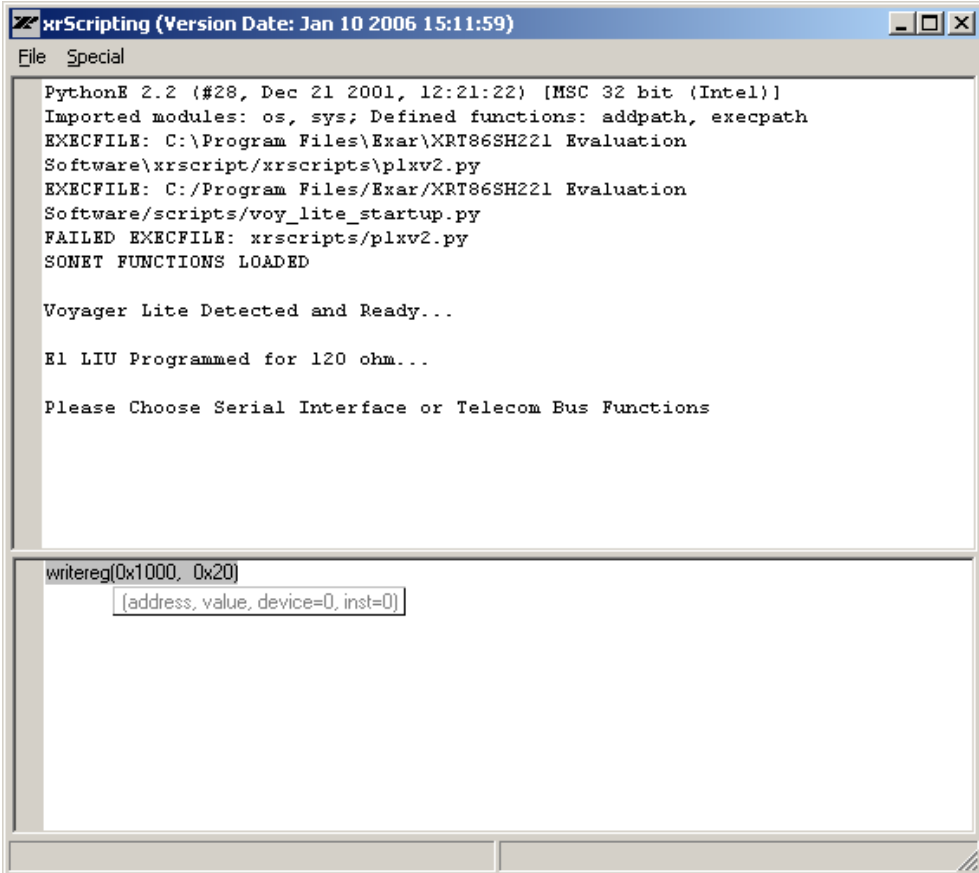
Please Choose Serial Interface or Telecom Bus Functions
0x1
```

The lower section is a command console with a text input field containing the command `readhex(0x0005)`. Below the input field, the text `[addr, device=0, inst=0]` is displayed, indicating the command's parameters.

### Example 2: Write to Specific Register

Looking at the dialog box below, you can see that the Command Console (lower window) was used to type the following command: `writereg(0x1000, 0x20)`, which will turn the E1 LIU Receiver On. Nothing will show up in the Display Monitor for Write functions. However, you can perform a Read function to verify that the value was written if desired.

**Note: Critical: You must press Ctrl + Enter to execute a command from the Command Console!**



```
PythonE 2.2 (#28, Dec 21 2001, 12:21:22) [MSC 32 bit (Intel)]
Imported modules: os, sys; Defined functions: addpath, execpath
EXECFILE: C:\Program Files\Exar\XRT86SH221 Evaluation
Software\xrscript\xrscripts\plxv2.py
EXECFILE: C:/Program Files/Exar/XRT86SH221 Evaluation
Software/scripts/voy_lite_startup.py
FAILED EXECFILE: xrscripts/plxv2.py
SONET FUNCTIONS LOADED

Voyager Lite Detected and Ready...

E1 LIU Programmed for 120 ohm...

Please Choose Serial Interface or Telecom Bus Functions

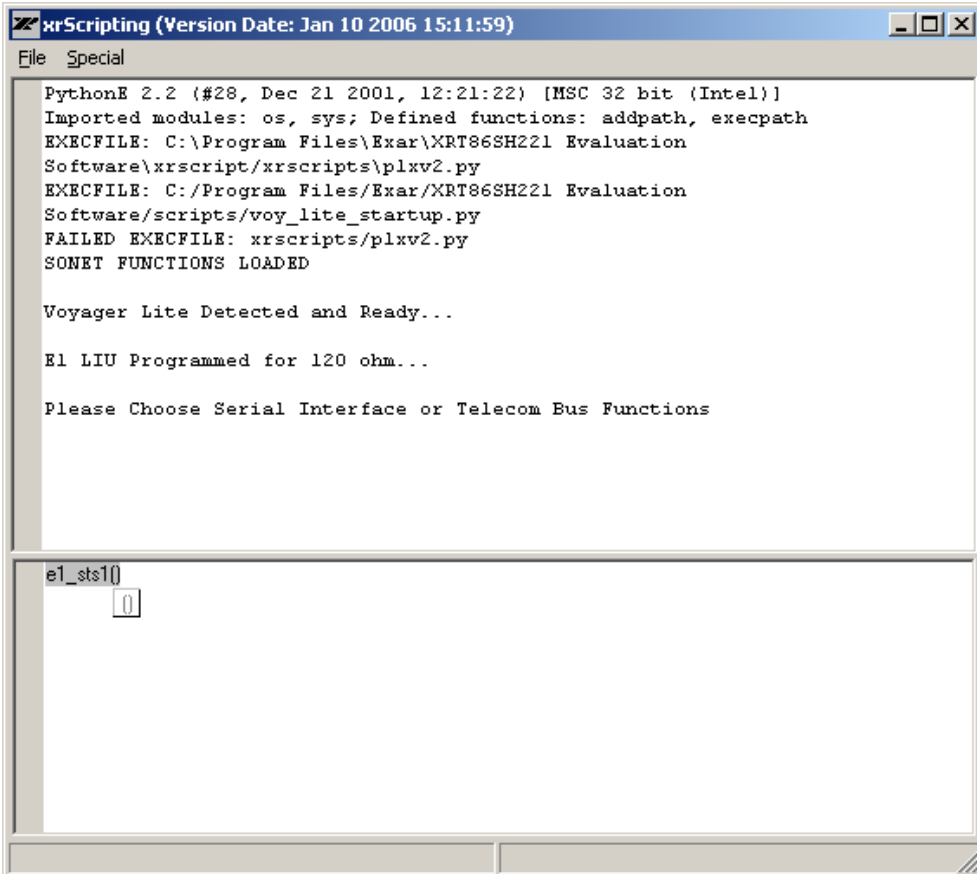
writereg(0x1000, 0x20)
(address, value, device=0, inst=0)
```



### Example 3: Run a Function Call for STM-0 Serial Interface

The following function “e1\_sts1()” (which is located in the “test\_scripts.py” script file) will set a general provision for running 21 channels of E1 traffic, VT Mapped up to STM-0 and output to the Serial Interface. On the Evaluation Board, a jumper cable can be used to externally route the output of the XRT75VL00 to the inputs. Once this is done, there should be no bit errors on the Test Equipment. From this point, the manual Read/Write access or a customized script can be used to fit a particular application.

**Note: Critical: You must press Ctrl + Enter to execute a command from the Command Console!**



```
xrScripting (Version Date: Jan 10 2006 15:11:59)
File Special
PythonE 2.2 (#28, Dec 21 2001, 12:21:22) [MSC 32 bit (Intel)]
Imported modules: os, sys; Defined functions: addpath, execpath
EXECFILE: C:\Program Files\Exar\XRT86SH221 Evaluation
Software\xrscript\xrscripts\plxv2.py
EXECFILE: C:/Program Files/Exar/XRT86SH221 Evaluation
Software/scripts/voy_lite_startup.py
FAILED EXECFILE: xrscripts/plxv2.py
SONET FUNCTIONS LOADED

Voyager Lite Detected and Ready...

E1 LIU Programmed for 120 ohm...

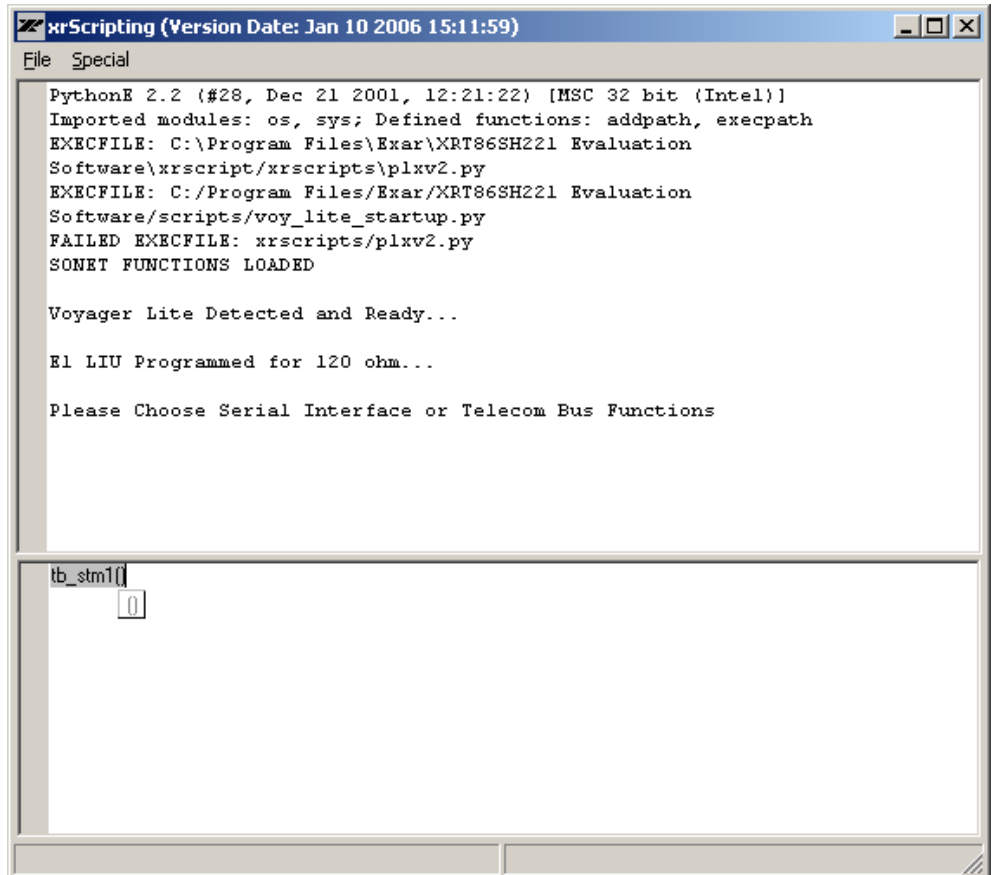
Please Choose Serial Interface or Telecom Bus Functions

e1_sts1()
0
```

#### Example 4: Run a Function Call for STM-0 Telecom Bus Interface

The following function “tb\_stm1()” (which is located in the “test\_scripts.py” script file) will set a general provision for running 21 channels of E1 traffic, VT Mapped up to STM-1 and output to the Telecom Bus. From this point, the manual Read/Write access or a customized script can be used to fit a particular application.

**Note: Critical: You must press Ctrl + Enter to execute a command from the Command Console!**



```
xrScripting (Version Date: Jan 10 2006 15:11:59)
File Special
PythonE 2.2 (#28, Dec 21 2001, 12:21:22) [MSC 32 bit (Intel)]
Imported modules: os, sys; Defined functions: addpath, execpath
EXECFILE: C:\Program Files\Exar\XRT86SH221 Evaluation
Software\xrscript\xrscripts\plxv2.py
EXECFILE: C:/Program Files/Exar/XRT86SH221 Evaluation
Software/scripts/voy_lite_startup.py
FAILED EXECFILE: xrscripts/plxv2.py
SONENT FUNCTIONS LOADED

Voyager Lite Detected and Ready...

E1 LIU Programmed for 120 ohm...

Please Choose Serial Interface or Telecom Bus Functions

tb_stm1()
```