

Multi-Uart Com Port Demo Quickstart Guide

REV A

Publication Date: 2012/10/15
XMOS © 2012, All Rights Reserved.



Table of Contents

1	Multiuart Com Demo: Quick Start Guide	3
1.1	Install Demon Tools on the Host PC	3
1.2	Hardare Setup	3
1.2.1	Setting Up The Multi Uart Slice Card	3
1.2.2	Setting Up The System	5
1.3	Import and Build the Application	5
1.4	Use of Software	6
1.5	Demo Application	6
1.6	Next Steps	7

1 Mult uart Com Demo: Quick Start Guide

IN THIS CHAPTER

- ▶ Install Demon Tools on the Host PC
 - ▶ Hardare Setup
 - ▶ Import and Build the Application
 - ▶ Use of Software
 - ▶ Demo Application
 - ▶ Next Steps
-

We use the XA-SK-UART8 Multi UART Slice Card together with the xSOFTip multi UART software to create a UART, and send data to and from a PC COM port. This application showcases some of the software key features and serves as an example on how to use its API. This demo features UART reconfiguration for various standard baud rates, receives bulk data via file uploads, and perform integrity checks on data before sending it back to COM port.

1.1 Install Demon Tools on the Host PC

The following tools should be installed on the host system in order to run this application

- ▶ For Win 7: Hercules Setup Utility by HW-Group
 - ▶ http://www.hw-group.com/products/hercules/index_en.html
- ▶ For MAC users: SecureCRT7.0
 - ▶ <http://www.vandyke.com/download/securecrt/>

1.2 Hardare Setup

The XP-SKC-L2 SliceKit Core board has four slots with edge connectors: SQUARE, CIRCLE, TRIANGLE and STAR.

To setup up the system refer to the figure and instructions below.

1.2.1 Setting Up The Multi Uart Slice Card

The demo shows the transfer of characters accross all 8 uarts by looping them all back. Data will be received (RX) from the host computer on Uart 0 (pins 0 and 1 of the header on the Slice Card), transmitted through the remaining 7 uarts and then the result of that is sent back to the host via Uart 0 TX.

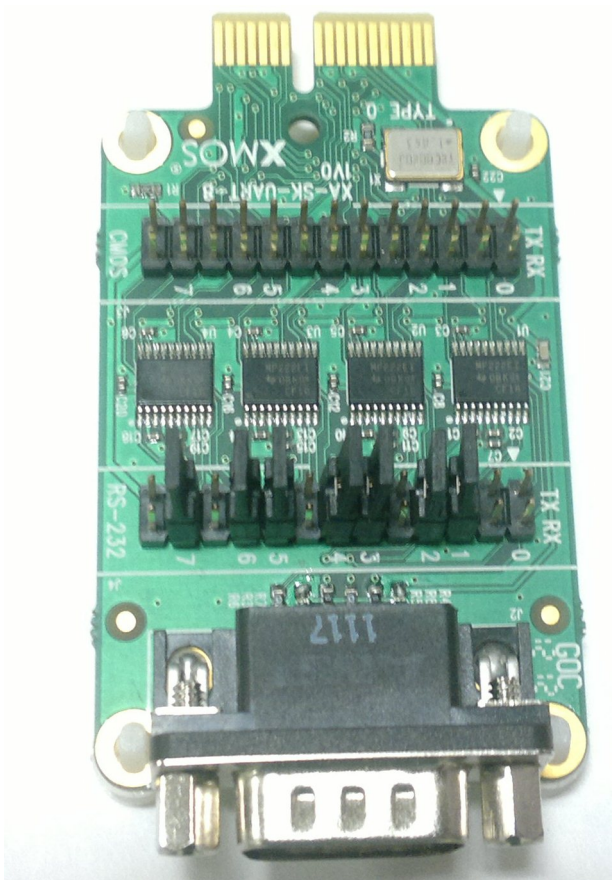


Figure 1:
Setting up
the Loopback
Jumpers

You will need seven 2-pin jumpers, which should be placed on header J4 (the one nearest the DB9 connector) of the Slice Card, on all the rows numbered 1 through 7 on the Slice Card.

TX	RX
5	6
7	8
11	12
13	14
17	18
19	20
23	24

1.2.2 Setting Up The System

1. Connect XA-SK-MUART Slice Card to the XP-SKC-L2 Slicekit Core board using the connector marked with the SQUARE.
2. To change it to slot STAR add `#define SK_MULTI_UART_SLOT_STAR 1` to `main.xc` and rebuild the binary. Then connect the slice card to STAR slot in the Slice kit core Board.
3. Connect the XTAG Adapter to Slicekit Core board, and connect XTAG-2 to the adapter.
4. Connect the XTAG-2 to host PC. Note that a USB cable is not provided with the Slicekit starter kit.
5. Switch on the power supply to the Slicekit Core board.
6. Connect a null serial cable to DB-9 connector on XA-SK-MUART Slice Card. The cable will need a cross over between the UART RX and TX pins at each end.
7. Connect other end of cable to Host DB-9 connector slot. If the Host does not have an DB-9 Connector slot then use USB-UART cable for the demo. We used the BF-810 USB to Uart adapter (See http://www.bafo.com/products_bf-810_S.asp (Part number : BF-810). Any other usb to uart bridge should do just as well.
8. Identify the serial (COM) port number provided by the Host or the USB to UART adapter and open a suitable terminal software for the selected serial port (refer to the Hercules or SecureCRT documentation above).
9. Configure the host terminal console program as follows: 115200 baud, 8 bit character length, even parity, 1 stop bit, no hardware flow control. The Transmit End-of-Line character should be set to CR (other options presented will probably be LF and CRLF).
10. Connect XA-SK-MUART Slice Card to the XP-SKC-L2 Slicekit Core board.
11. Connect the XTAG Adapter to Slicekit Core board, XA-SK-XTAG2 connector(xtag slice) and connect XTAG-2 to the adapter.
12. Connect the XTAG-2 to host PC.
13. Switch on the power supply to the Slicekit Core board.
14. Open the serial device on the host console program

1.3 Import and Build the Application

1. Open xTimeComposer and check that it is operating in online mode. Open the edit perspective (Window->Open Perspective->XMOS Edit).
2. Locate the 'Slicekit COM Port MUART Demo' item in the xSOFTip pane on the bottom left of the window and drag it into the Project Explorer window in the

xTimeComposer. This will also cause the modules on which this application depends (in this case, `sc_util`) to be imported as well.

3. Click on the Slicekit COM Port MUART Demo item in the Explorer pane then click on the build icon (hammer) in xTimeComposer. Check the console window to verify that the application has built successfully.

For help in using xTimeComposer, try the xTimeComposer tutorial (see Help->Tutorials within the tool).

Note that the Developer Column in the xTimeComposer on the right hand side of your screen provides information on the xSOFTip components you are using.

1.4 Use of Software

1. Open the XDE
2. Choose **Run ► Run Configurations**
3. Double-click *XCore Application* to create a new configuration
4. In the *Project* field, browse for *app_sk_muart_com_demo*
5. In the *C/C++ Application* field, browse for the compiled XE file
6. Ensure the XTAG-2 device is selected in the *Target:* adapter list
7. Click **Run**
8. Wait for the configuration completion messages in XDE console for all UART's before proceeding with the Demo.

1.5 Demo Application

1. Select and Open the configured terminal client application console
2. Press any key on console. A user menu will be displayed
3. Key in `e` to enter echo mode. Type in any character from the key board and application echoes the key pressed. In order to get back to user menu, press `Esc` key.
4. Key in `r` to enter reconfiguration mode. Key in new baud rate value (select one of the values from 115200, 57600, 38400, 19200, 9600, 4800, 600) followed by CR (Enter) key. The UART will be reconfigured (XDE console will display the value entered). The terminal console should be reopened with the new selected baud rate. Press `h` to display user menu.
5. Key in `b` in order to pipe data through UART channels 1-7. Type in the Console window and then press Ctrl+D to send the data through 7 channels and receive it for display. To send a file, select the **File Send** option in Console SW (Right click in Hercules) and select the file. Then press CTRL+D from console to send

file through channels 1-7 and receive it. Timing information is also displayed. Hardware setup for Pipe option should be as shown in the diagram, loop all the Uart channels (like Channel 7 Shown in the figure) except channel 0. If the connection to any of the channel is disconnected you will not see data received back and a message is displayed on the terminal saying that Muart pipe is broken. However, the timing information is not accurate for very small transfer sizes (characters less than 8).

6. Key in `h` in order to display user menu. This help is displayed any time during execution by pressing `Esc` key followed by `h`

1.6 Next Steps

1. Refer to the `module_multi_uart` documentation for implementation details of this application and information on further things to try.:wq!
2. Evaluate the full Ethernet to Serial (8 Uart) reference product which can be found at https://github.com/xcore/sw_serial_to_ethernet. This is a fully featured reference product including an embedded webserver, multicast configuration via UDP and a host of other features.
1. Examine the application code. In xTimeComposer navigate to the `src` directory under `app_sk_muart_com_demo` and double click on the `main.xc` file within it. The file will open in the central editor window.
2. This code demonstrates about simple Muart demo application and usage of muart component.

This application uses 8 UART channels. Take a look at the Serial to Ethernet Bridging application which uses Muart Component. Have a look at the documentation for that component and how its API differs from the stand alone Uart.

This application offers many features including dynamic reconfiguration, an embedded webserver and 8 uarts running up to 115KBaud.

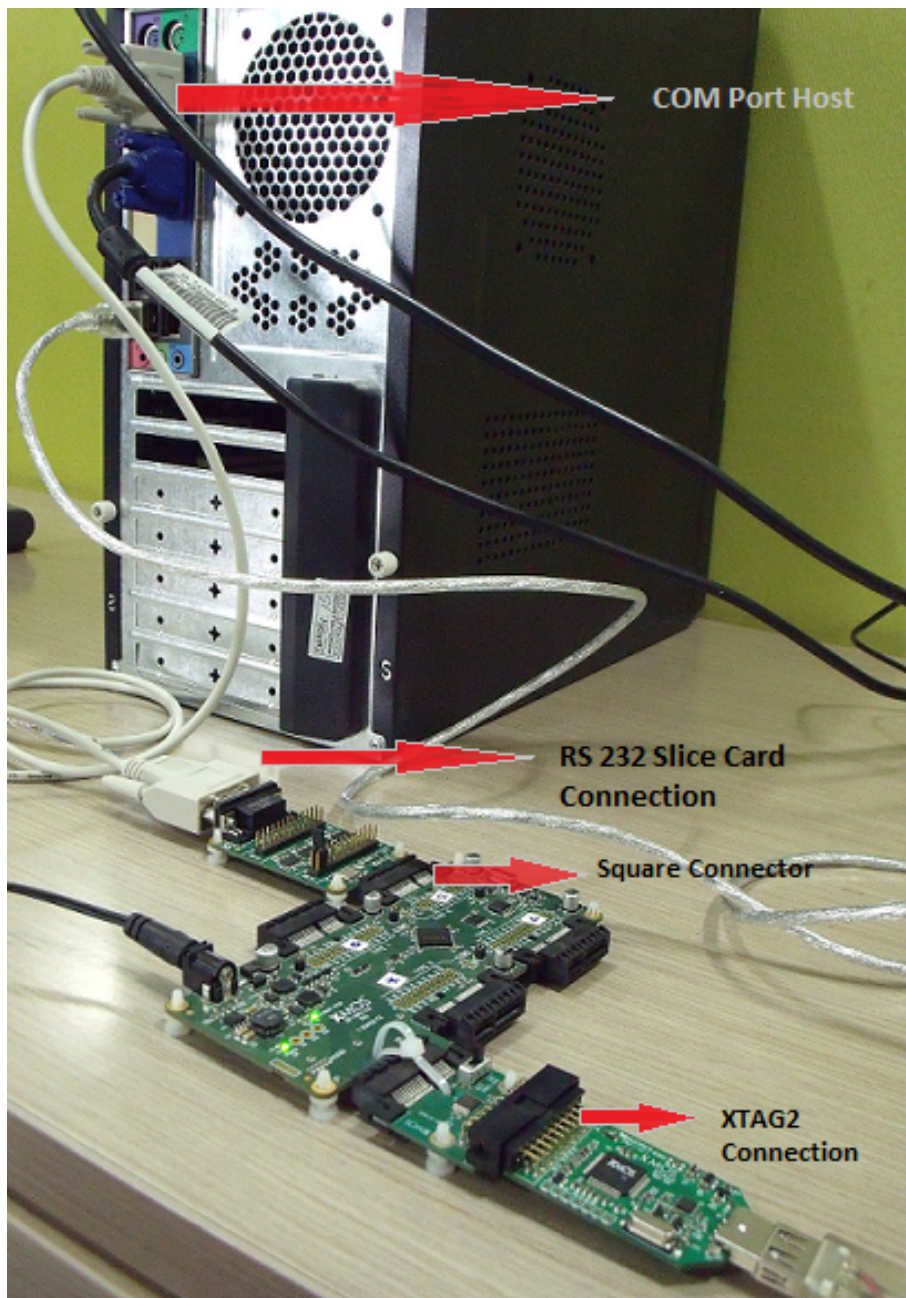


Figure 2:
Full system
setup for
Multi-UART
Demo

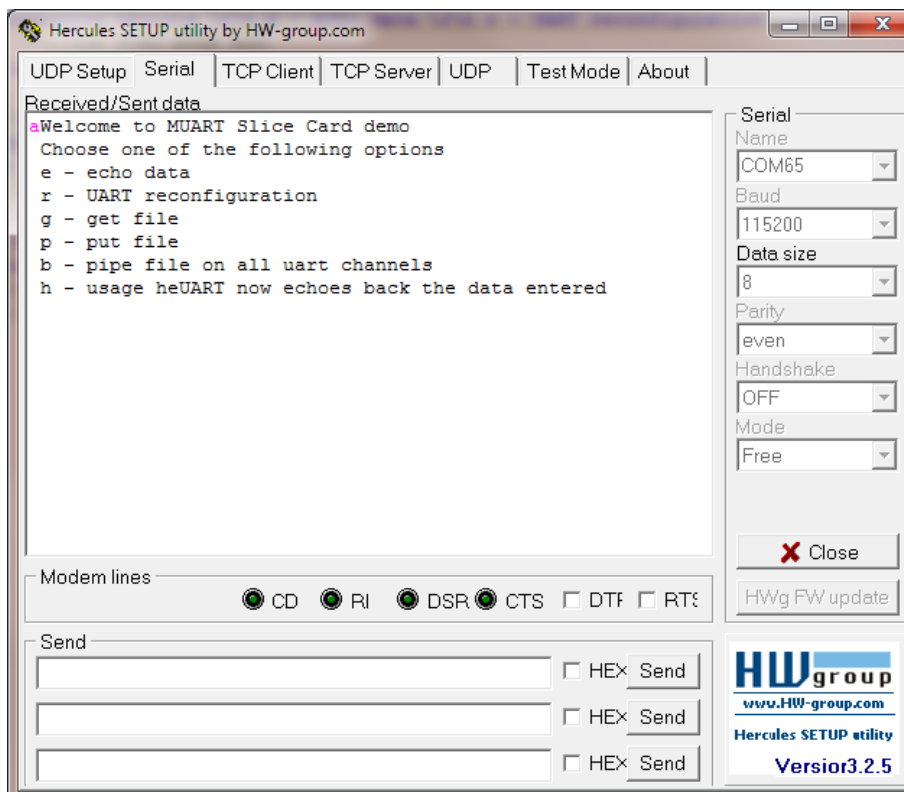


Figure 3:
Screenshot of
Hypertermi-
nal
window



Copyright © 2012, All Rights Reserved.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

XMOS and the XMOS logo are registered trademarks of Xmos Ltd. in the United Kingdom and other countries, and may not be used without written permission. All other trademarks are property of their respective owners. Where those designations appear in this book, and XMOS was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.