Serial to Ethernet (S2E) bridging application quickstart guide

IN THIS DOCUMENT

This application serves as a reference design to demonstrate bridging between Ethernet and serial communication devices. Some features of this application are:

- 10/100 Mbit Ethernet port
- Supports up to 8 serial ports (UARTs) with baud rates up to 115200 at standard UART configuration settings
- Webserver to facilitate dynamic UART configuration
- Telnet server to support data transfer via a telnet socket associated with each UART
- Device discovery and IP configuration management of multiple Serial to Ethernet (S2E) devices in the network
- Flash memory storage and retrieval for device settings such as IP, UART configuration and web pages
- CMOS/TTL level and RS232 level communication for UARTs

1 Host computer setup

A computer with:

- With a spare Ethernet port
- Internet browser (Internet Explorer, Chrome, Firefox, etc...)
- Download and install xTIMEcomposer studio (v13.0.0 or later) from XMOS xTIMEcomposer downloads webpage
- A spare USB port for XTAG debug

Publication Date: 2014/6/6 Document Number: XM-UNKNOWNA

XMOS © 2014, All Rights Reserved

- A spare DB9 port (an additional USB port may be used for serial to USB adapter if DB9 port is not available)

For serial-telnet data communication demo, the following are required in addition to the above:

- A null serial cable to DB-9 connector. The cable will need a cross over between the UART RX and TX pins at each end.
- If the computer does not have a DB-9 connector slot, any USB-UART cable can be used. For the demo, we use BF-810 USB-UART adapter

(http://www.bafo.com/products/accessories/usb-devices/bf-810-usb-to-serial-adapter-d
* A suitable terminal client software. For MAC users, try SecureCRT (http://www.vandyke.com/download/securecrt/) and for Linux users, try cutecom (http://cutecom.sourceforge.net/). We use hercules client (http://www.hw-group.com/products/hercules/index_en.html) on a Windows platform for the demo.

2 Hardware setup

Required sliceKIT units:

- xCORE General Purpose (L-series) sliceKIT core board 1V2 (XP-SKC-L2)
- Ethernet sliceCARD 1V1 (XA-SK-E100)
- Multi UART sliceCARD (XA-SK-UART-8)
- xTAG-2 debug adapter and sliceKIT connector (xTAG-2 and XA-SK-XTAG2)

Setup:

- Connect the XA-SK-XTAG2 adapter to the XP-SKC-L2 sliceKIT core board.
- Ensure the *XMOS Link* switch is at *ON* position on the XA-SK-XTAG2 adapter.
- Connect XTAG2 to XSYS side (J1) of the XA-SK-XTAG2 adapter.
- Connect the XTAG2 to your computer using a USB cable.
- Connect the XA-SK-UART-8 Multi UART sliceCARD to the XP-SKC-L2 core board's SQUARE (indicated by a white colour square) slot.
- Connect the XA-SK-E100 Ethernet sliceCARD to the XP-SKC-L2 core board's TRIANGLE (indicated by a white colour triangle) slot.
- Using an Ethernet cable, connect the other side of XA-SK-E100 Ethernet sliceCARD to your computer's Ethernet port.
- Connect the 12V power supply to the core board and switch it ON.

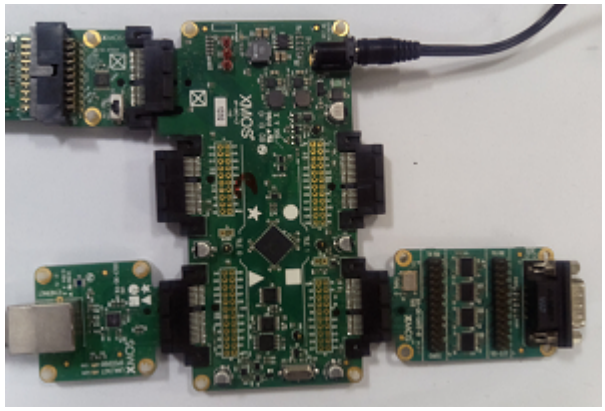

Figure 1: Hardware setup

3 Import and build the application

Importing the `serial to ethernet` reference application:

- Open the xTIMEcomposer studio.
- Open the *Edit* perspective (Window -> Open Perspective -> XMOS Edit).
- Access the *Import* option either by right clicking in the project explorer window or through File ->Import menu
- Click *Import* option (Import -> General -> Existing Projects into Workspace and click Next).
- Choose *Select archive file* option and click *Browse* button.
- Select s2e reference design release package and click *Finish* button
- The application is called as *app_serial_to_ethernet* in the *Project Explorer* window.

Building the `serial to ethernet` application:

- Click on the *app_serial_to_ethernet* item in the *Project Explorer* window.
- Click on the *Build* (indicated by a 'Hammer' picture) icon.
- Check the *Console* window to verify that the application has built successfully.

4 Flash the web pages and device configuration

To flash the web pages and device configuration using xTIMEcomposer studio:

- In the *Project Explorer* window, locate the *app_serial_to_ethernet.xe* and *web_data.bin* in the (app_serial_to_ethernet -> bin)

- Right click on *app_serial_to_ethernet.xe* and click on (Flash As -> Flash Configurations...).
- In the *Flash Configurations* window, double click the *xCORE Application* to create a new flash configuration.
- Navigate to *XFlash Options* tab and apply the following settings:
  - Check *Boot partition size (bytes):* and its value as 0x10000
  - *Other XFlash Options:* as `--data bin/web_data.bin`
- Click on *Apply* and then *Flash* to the XMOS device.
- Check the *Console* window to verify flashing progress.

5 Run the application

To run the application using xTIMEcomposer studio:

- In the *Project Explorer* window, locate the *app_serial_to_ethernet.xe* in the (app_serial_to_ethernet -> Binaries).
- Right click on *app_serial_to_ethernet.xe* and click on (Run As -> xCORE Application).
- In the *Run Configurations* window, double click the *xCORE Application* to create a new xCORE application launch configuration.
- A *Select Device* window appears.
- Select *XMOS XTAG-2 connected to L1* and click *Apply*.
- Click *Run* and check the *Console* window for any messages.

Demo:

- The following message appears in the *Console* window of the xTIMEcomposer studio:

```
Address: 0.0.0.0
Gateway: 0.0.0.0
Netmask: 0.0.0.0
```

- At this point, the XMOS device is trying to acquire an IP address in the network. Wait for some time (approximately 20 seconds) for the following message to appear in the *Console* window. Note, the IP address may be different based on your network:

```
ipv4ll: 169.254.161.178
```

- Open a web browser (Firefox, etc...) in your host computer and enter the above IP address in the address bar of the browser. It opens a web page as hosted by the webserver running on the XMOS device.

Figure 2: Page hosted by webserver to support UART configuration

- To change the configuration of a UART via web page, click on any UART, say UART 1. It opens a new page for configuring the selected UART 1.
- Observe the *Telnet Port* value for the selected UART. This is the telnet port number on which the UART1 is bridged.
- Alter the *Baud Rate* settings from *115200* to *57600* by choosing this value from the drop box.
- Click on *Set* button and verify the *Response:* value is populated as *Ok*.
- Click *Back to main config page* link to go back to the home page and verify the modified UART settings are intact by clicking on the same UART 1.
- On the main page, click on *Save* button to store any modified UART settings onto the flash.

Serial-Telnet data communication demo:

This demo showcases the data bridging between Ethernet and serial devices. Data from the Serial console (UART) is sent to the corresponding telnet socket associated with the UART and vice versa. In order to run this demo, follow the below instructions.

In addition to the above hardware setup

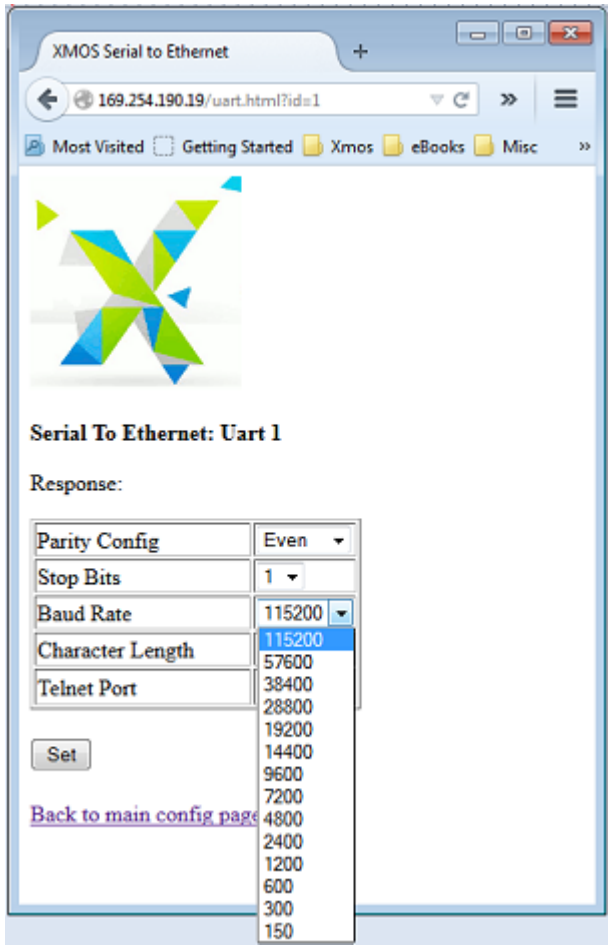

Figure 3: Modifying UART configuration via web page

- Connect a null serial cable to DB-9 connector on Multi UART sliceCARD.
- Connect other end of cable to DB-9 connector slot on the host or USB-UART adapter.
- Identify the serial (COM) port number provided by the Host or *USB to UART* adapter and open a suitable terminal client software for the selected COM port (if required, refer to the documentation of the selected application).
- Configure the host COM port console settings; sample settings while using Hercules client should be as follows:

Parameter	Value
Baud rate	115200
Data size	8
Parity	Even
Handshake	off
Mode	Free

The Transmit End-of-Line character should be set to *CR* (other options presented will probably be *LF* and *CRLF*). In hercules, this setting is achieved by right clicking on *Received/Sent Data* text box, select *Transmit EOL*, select *CR(Mac)* option

If any other terminal console is used, and has any additional settings, following values are used: .. list-table:

```
* - Parameter
  - Value
* - Stop bit
  - 1
* - hardware flow control
  - none
```

- Click on *Open* to open the COM port.
- Now, in order to establish a telnet connection to the above serial connection, open a telnet client application (On Windows, open another instance of the Hercules application, select *TCP Client* tab)
- Configure the telnet client application with ip address as XMOS device address. Key in the port number as *46* in order to connect to the UART0.
- Click *Connect* so that the telnet client connects to the telnet server running on the S2E device. Observe a welcome message *Welcome to serial to ethernet telnet server demo! This server is connected to uart channel 0* appears on the client application console.
- Key in some data from the serial console and observe the data is displayed on the telnet console.
- Now send some data from the telnet console and verify the same data on the serial console.
- Explore the terminal client options to transfer a file in both directions and observe the duplex data transfer in action.

6 Next steps

- Connect two or more USB-UART adapters to the host and Multi UART sliceCARD. Open the terminal client applications for the correct configuration as detailed in the above *Serial-Telnet data communication demo*. Test the data communication between the connected UARTs and their corresponding Telnet sockets.

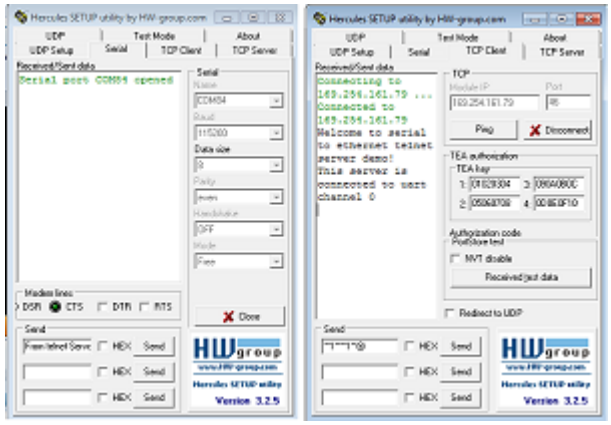

Figure 4: Screenshot of two Hercules application instances for a serial console and a telnet client

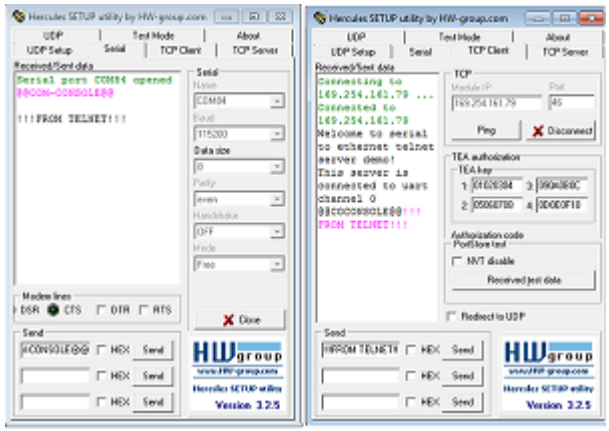

Figure 5: Data communication between a telnet socket and a serial console (UART)

- Detach xTAG-2 debug adapter and sliceKIT connector from xCORE General Purpose (L-series) sliceKIT core board. Connect Ethernet sliceCARD to a spare Ethernet port of the router. If your platform is a MAC or a linux host, navigate to `sw_serial_to_ethernet -> tests -> udp_test_server` and run the udp_server.py python script (python udp_server.py). If you are using a Windows host, download *Serial_to_Ethernet_UDP_test_server* package and extract its contents to a directory. Navigate to (udp_test_server -> windows -> udp_server.exe), right-click on udp-server.exe and run as Administrator. The script displays the selected network adapter on the console. If there are multiple network adapters on your host, ensure the ip address used by the script corresponds to the one used by your network adapter connected to the router. Now, select option `1` to discover the S2E devices available on the network. Look at the S2E device ip address as displayed by the script. Open a web page or test Telnet-UART data communication using ip of the S2E device. Select other choices to change ip configuration settings of the S2E device(s).

- Take a look at the `http://xcore.github.io/sw_serial_to_ethernet` for a more detailed documentation on using various features, design and programming guide for the application.

XMOS®

Copyright © 2014, 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.