

Unit / Module Description:	Installation and first step
Unit / Module Number:	SMT6058
Document Issue Number:	1.0
Issue Date:	30/07/09
Original Author:	F.S

Application Note for SMT6058

Ethernet support for the SMT148- FX standalone carrier board

Sundance Multiprocessor Technology Ltd, Chiltern House,
Waterside, Chesham, Bucks. HP5 1PS.

This document is the property of Sundance and may not be copied
nor communicated to a third party without prior written
permission.

© Sundance Multiprocessor Technology Limited 2009



Certificate Number FM 55022

Revision History

Issue	Changes Made	Date	Initials
1.0	First release	30/07/09	FS

Table of Contents

1	Introduction	5
2	Related Documents	5
2.1	Referenced Documents	5
3	Acronyms, Abbreviations and Definitions	5
3.1	Acronyms and Abbreviations	5
4	Installation	6
4.1	Installation specification	6
4.2	EDK patch.....	6
5	Development procedure	7
5.1	SMT148FX configuration	7
5.2	The IP configuration.....	7
5.3	The Webserver	8
5.3.1	First utilization.....	8
5.3.2	Modify the Web Server	10
5.4	The Ethernet with SMT362 comport loopback	12

Table of Figures

Figure 1 : TCP/IP configuration	7
Figure 2 : image.mfs initialization	8
Figure 3 : SMT148-FX Web Server.....	9
Figure 4 : HyperTerminal.....	10
Figure 5 : image.mfs generation	11
Figure 6 : Web Server modified.....	11
Figure 7 : Virtex 4 comports connections.....	12
Figure 8 : Host application	13
Figure 9 : Debug and 3L server result	13

1 Introduction

This document specifies the SMT6058 installation procedure and the first step to run the design examples.

The SMT6058 product will allow accessing the SMT148-FX60 carrier boards via its Gigabit Ethernet port (onboard RJ45 connector). The SMT6058 includes a software TCP/IP stack for the Gigabit Ethernet interface, a default firmware implementing a PowerPC core for the Virtex-4 FX60 FPGA device, the software functions to access the board resources (flash programming) and on-board modules via Rocket-IO Serial Links and/or Comport link from the gigabit Ethernet port (TCP/IP stack).

2 Related Documents

2.1 Referenced Documents

[SMT6058 : Product specification](#)

[SMT148FX : 4 site stand alone TIM carrier](#)

[SMT362 : Dual 'C6455 DSP Module](#)

3 Acronyms, Abbreviations and Definitions

3.1 Acronyms and Abbreviations

[A list of acronyms etc](#)

4 Installation

4.1 Installation specification

Xilinx EDK does NOT support 'spaces' in the project path.

Copy the SMT6058 folder and make sure that you have no space in your project path. (good path example : C:\SMT6058)

The folder contains two main project:

- The Webserver : “..\SMT6058\Example\Webserver\system.xmp”
- The Ethernet with SMT362 comport loopback :
“..\SMT6058\Hardware\FPGA\FX60\EDK\system.xmp”
This project include a flash and bootloader application

4.2 EDK patch

The EDK patch, in the EDK_Sources_Patch folder, is the replacement files for the IITEMAC (driver for the Ethernet) and the LWiP TCP/IP stack.

- The archive lltemac_v1_00_b should replace the one installed in :
%:\Xilinx\10.1\EDK\sw\XilinxProcessorIPLib\drivers

- The archive lwip_v3_00_a should replace the one installed in :
%:\Xilinx\10.1\EDK\sw\ThirdParty\sw_services

This patch initializes properly the PHY in the correct speed. This is required to adjust chipset's delay as well.

We would not need this if the Xilinx ethernet autonegotiation would be working as expected.

If you haven't yet installed the patch and already use the SMT6058, just clean and rebuild your software application to make sure that you will use the right drivers.

5 Development procedure

5.1 SMT148FX configuration

The SMT148-FX has to be configured to run the SMT6058 applications.

First update the CPLD firmware with the firmware:

```
"..\SMT6058\Hardware\CPLD\top.jed"
```

The SMT6058 projects uses the comport of the SMT148-FX Virtex4, the Spartan has to be programmed with the firmware provided in the SMT6058 package.

```
"..\SMT6058\Hardware\FPGA\XCS1500\com.sundance.smt148-  
fx.sc3s1500.fx60toflash\ com.sundance.smt148-fx.sc3s1500.fx60toflash.app"
```

This firmware connect the SMT148-FX Virtex4 to its flash through the comport and to the T1CP4.

5.2 The IP configuration

Make sure that you have no firewall on the port 80 of your network. For a direct link between your computer and the SMT148-FX Ethernet connector configure your IP like the following picture, or make sure that you have the good configuration depending of you project application.

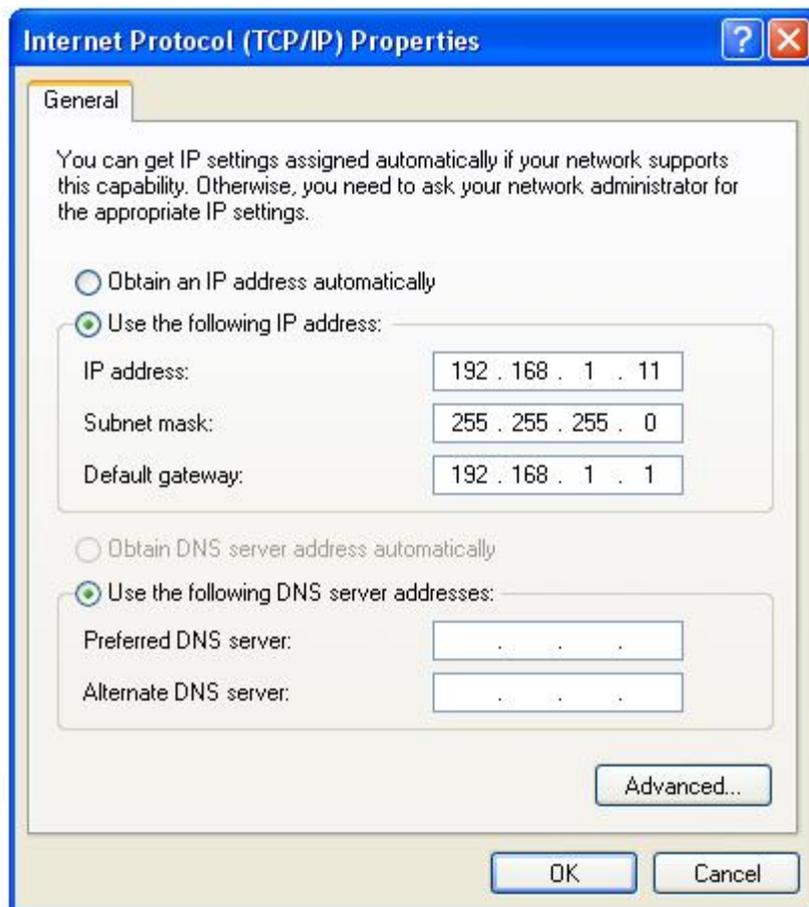


Figure 1 : TCP/IP configuration

5.3 The Webserver

What is in the design?

- PPC running at 300MHz,
- Kernel working on the PowerPC with cache memory,
- Multithreading functionality,
- NO DMA ENGINES
- ZBT RAM implemented
- On-board UART for debug output (if the RS-232 is connected to the board while the design is executed, some feedback from the PPC is sent. This can be observed with an HyperTerminal)
- WebServer (GUI/LED control)
- Ethernet Link connection at 10/100/1000Mbps

5.3.1 First utilization

Open the “..\SMT6058\Example\Webserver\system.xmp” project, Launch Platform Studio SDK, if the Webserver application is not imported under SDK, or if you have any errors (can happen when you will import the application for the first time), delete the Webserver application (by selecting “Also delete contents under ..\SMT6058\Example\Webserver\SDK_projects\Webserver”) and Import this one again with Import, Import XPS Application Projects to SDK.

This project provides an html page to control the SMT148FX LEDs from a web browser. This page is saved in the image.mfs file. This file has to be added during the initialization at address 0x00130000.

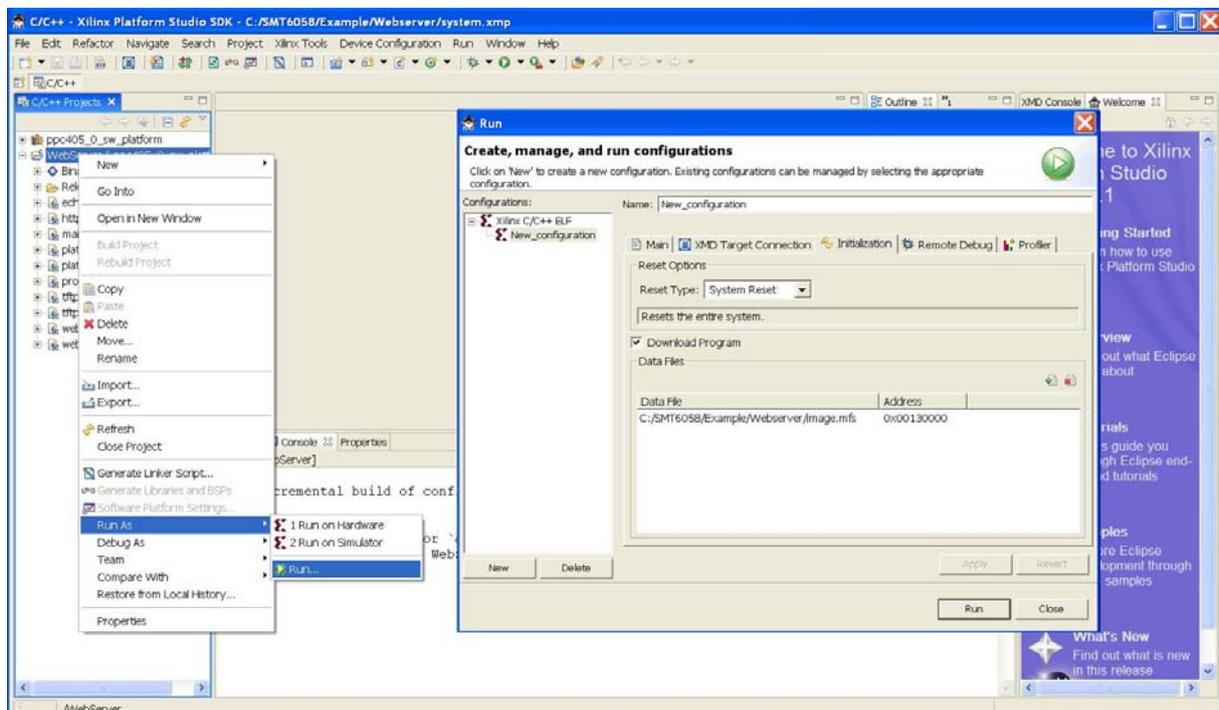


Figure 2 : image.mfs initialization

Before running the application, make sure that you have connected the SMT148-FX60 FPGA via the FPGA JTAG chain using the Xilinx JTAG pod connected to the JP6 header on the SMT148-FX.

Connect the SMT148-FX RJ45 connector to the PC ethernet card via an Ethernet cable. If the RS-232 is connected to the board while the design is executed, some feedback from the PPC is sent. This can be observed with an HyperTerminal.

Now program the FPGA  and after run the Webserver application .

You can test the connection between the PC and the PowerPC (EMAC connections). Open a command prompt window and type the instruction: PING 192.168.1.10

And in a web browser, type: 192.168.1.10

A simple web page is displayed, to demonstrate that the Webserver is running on the PowerPC.

LEDs can be controlled from the web server to demonstrate the communication of system Ethernet-PPC and peripherals.

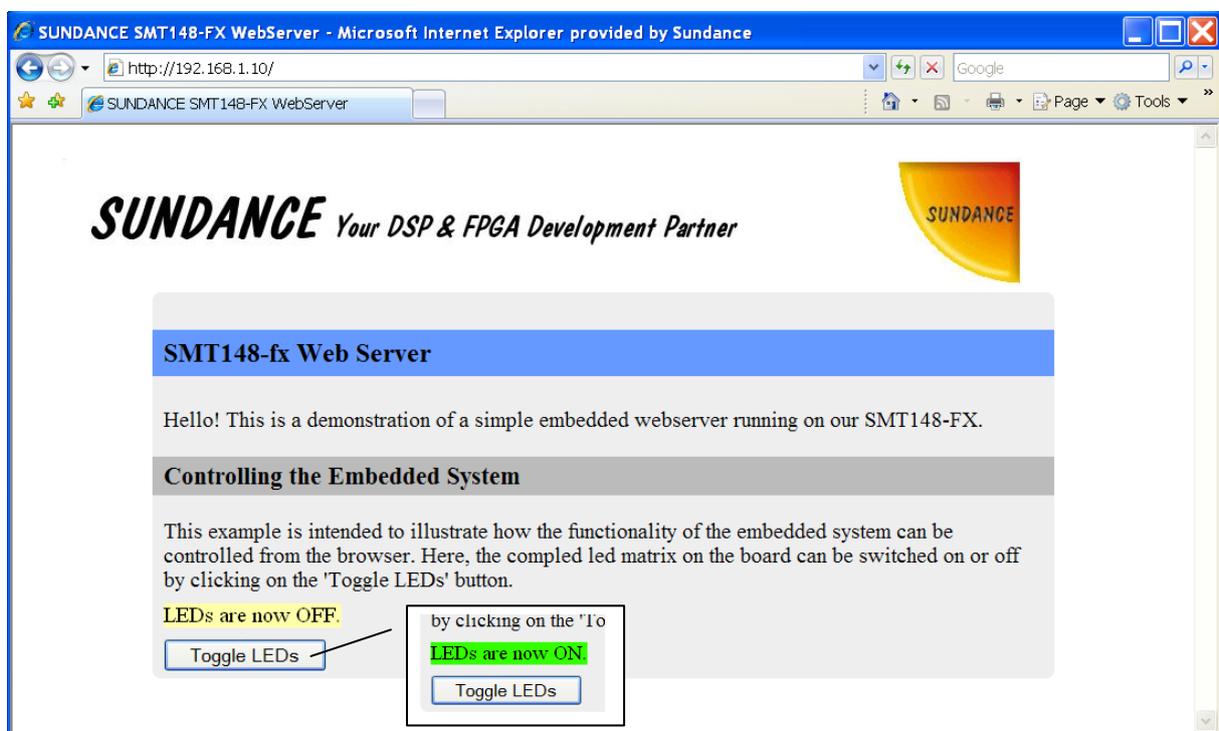


Figure 3 : SMT148- FX Web Server

When you click on the 'Toggle LEDs' button, the LEDs status is shown on the Web Server and you can observe that all the SMT148FX LEDs matrix are ON or OFF on the board.

If you have plugged the RS232, you should have the same result as the following picture.

```
-----lwIP test WebServer -----
Open up your favorite browser and type:
http://192.168.1.10

Board IP: 192.168.1.10
Netmask : 255.255.255.0

Gateway : 192.168.1.1
XL11emac detect_phy: No PHY detected. Assuming a PHY at address 0
auto-negotiated link speed: 1000
PhySetup_Marvell_88e111: Try to set speed of 1000 Mbps
PhySetup_Marvell_88e111: Retries 4
PhySetup_Marvell_88e111: Link is fine

Memory File System initialized
http GET: index.html
http GET: yui/yahoo.js
http GET: yui/dom.js
http GET: yui/event.js
http GET: yui/conn.js
http GET: yui/anim.js
http GET: js/main.js
http GET: css/main.css
http GET: images/sundance.JPG
http POST: switch state: 0
http POST: ledstatus: FFFFFFFF
http POST: ledstatus: 0
http POST: ledstatus: FFFFFFFF
```

Figure 4 : HyperTerminal

5.3.2 Modify the Web Server

The html page is “..\SMT6058\Example\Webserver\memfs\index.html”.

To show how to modify the page and make the image.mfs, start by editing the index.html code.

For a little example we will just centre the Sundance image, the line 17 should become:

```
<center></img></center>
```

Now you need to generate the image.mfs file, under EDK or SDK open the Shell .

Go inside your memfs folder, it should be with :

- cd memfs

And now generate the image.mfs with :

- make

```

/cygdrive/c/SMT6058/Example/Webserver/memfs
EDK Shell
Xilinx EDK 10.1.03 Build EDK_K_SP3.6
Copyright (c) 1995-2008 Xilinx, Inc. All rights reserved.

Analyzing Cygwin versions...
Xilinx EDK detected Cygwin installation v1.5.17(0.129/4/2) on your machine.
This Cygwin (C:\Xilinx\10.1\EDK\cygwin\bin) will be used to run Xilinx EDK
tools.

Fabiens@SudacaXP /cygdrive/c/SMT6058/Example/Webserver
$ cd memfs

Fabiens@SudacaXP /cygdrive/c/SMT6058/Example/Webserver/memfs
$ make
mfsgen -cubfs ../image.mfs 1000 x
mfsgen
Xilinx EDK 10.1.03 EDK_K_SP3.6
Copyright (c) 2004 Xilinx, Inc. All rights reserved.

Makefile 43
css:
main.css 762
images:
sundance.JPG 12497
index.html 1608
js:
main.js 7284
yui:
anim.js 12588
conn.js 11639
dom.js 10862
event.js 14317
yahoo.js 5360
MFS block usage (used / free / total) = 161 / 839 / 1000
Size of memory is 532000 bytes
Block size is 532
mfsgen done!

Fabiens@SudacaXP /cygdrive/c/SMT6058/Example/Webserver/memfs
$ -

```

Figure 5 : image.mfs generation

When the image.mfs is generated, just run the application  under SDK and you should see the difference with the figure 3, the Sundance image is now in the centre of the browser.

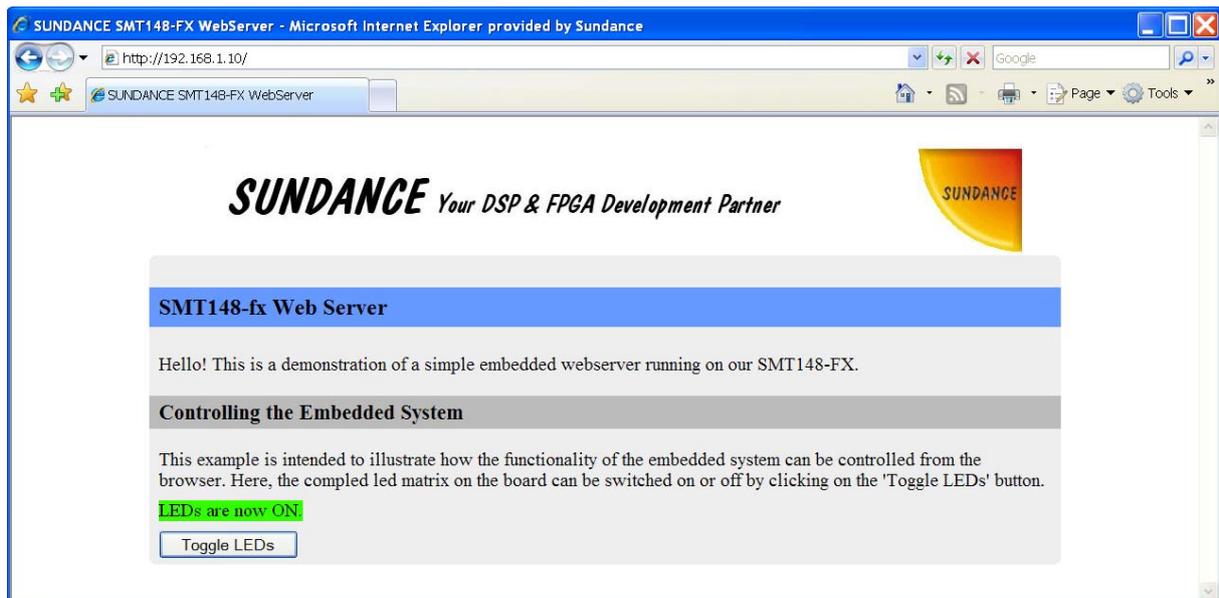


Figure 6 : Web Server modified

5.4 The Ethernet with SMT362 comport loopback

This project provides two comport connections between the PowerPC and the Spartan.

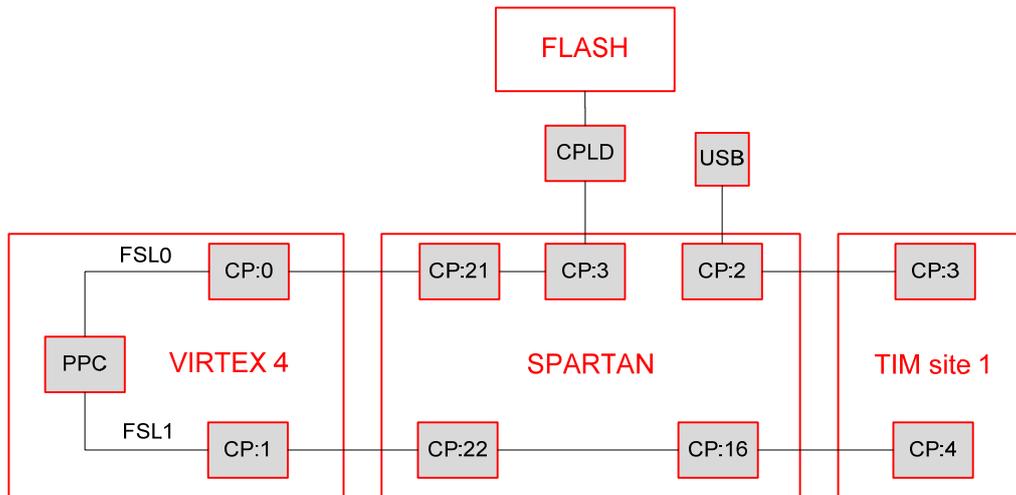


Figure 7 : Virtex 4 comport connections

The comport CP0 is connected to have the flash access from the Virtex4 and the CP1 is connected to the T1CP4.

First you need to run the 3L SMT362 loopback application with the TIM on site 1.

“\SMT6058\Hardware\DSP\SMT362\CpLoopback\output\CpLoopback.app”

Open the system.xmp project, Launch Platform Studio SDK, if the TestApp_Ethernet application is not imported under SDK Import this one with Import, Import XPS Application Projects to SDK.

To get the same Debug result of the following picture, uncomment the Line 19 in the eth2cp.c file.

```
#define _UART_DBG
```

Before running the application, make sure that you have connected the SMT148-FX60 FPGA via the FPGA JTAG chain using the Xilinx JTAG pod connected to the JP6 header on the SMT148-FX.

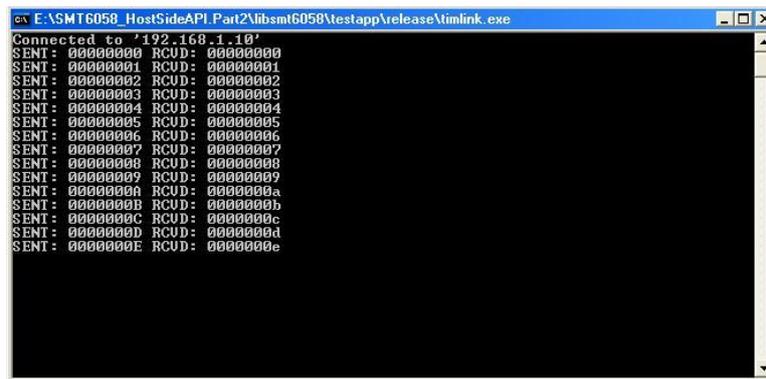
Connect the SMT148-FX RJ45 connector to the PC ethernet card via an Ethernet cable. If the RS-232 is connected to the board while the design is executed, some feedback from the PPC is sent. This can be observed with an HyperTerminal.

Now program the FPGA  and after run the TestApp_Ethernet application .

Then execute the host application:

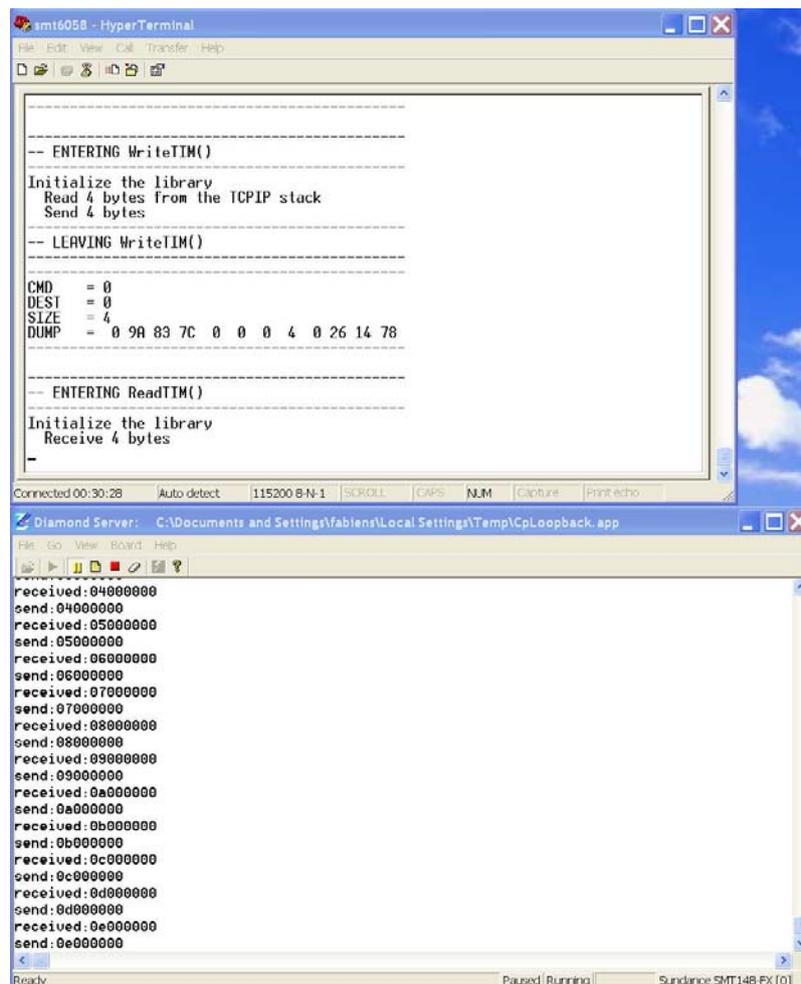
```
“..\SMT6058\Host\testapp\release\timlink.exe”
```

You should get the following result.



```
Connected to '192.168.1.10'  
SENT: 00000000 RCUD: 00000000  
SENT: 00000001 RCUD: 00000001  
SENT: 00000002 RCUD: 00000002  
SENT: 00000003 RCUD: 00000003  
SENT: 00000004 RCUD: 00000004  
SENT: 00000005 RCUD: 00000005  
SENT: 00000006 RCUD: 00000006  
SENT: 00000007 RCUD: 00000007  
SENT: 00000008 RCUD: 00000008  
SENT: 00000009 RCUD: 00000009  
SENT: 0000000a RCUD: 0000000a  
SENT: 0000000b RCUD: 0000000b  
SENT: 0000000c RCUD: 0000000c  
SENT: 0000000d RCUD: 0000000d  
SENT: 0000000e RCUD: 0000000e
```

Figure 8 : Host application



```
-- ENTERING WriteTIM()  
-----  
Initialize the library  
Read 4 bytes from the TCP/IP stack  
Send 4 bytes  
-----  
-- LEAVING WriteTIM()  
-----  
CMD = 0  
DEST = 0  
SIZE = 4  
DUMP = 0 9A 83 7C 0 0 0 4 0 26 14 78  
-----  
-- ENTERING ReadTIM()  
-----  
Initialize the library  
Receive 4 bytes  
-----  
-----  
received:04000000  
send:04000000  
received:05000000  
send:05000000  
received:06000000  
send:06000000  
received:07000000  
send:07000000  
received:08000000  
send:08000000  
received:09000000  
send:09000000  
received:0a000000  
send:0a000000  
received:0b000000  
send:0b000000  
received:0c000000  
send:0c000000  
received:0d000000  
send:0d000000  
received:0e000000  
send:0e000000
```

Figure 9 : Debug and 3L server result

To restart the PPC software you need first to reset the system due to the direction of the comports which is changed.

The PPC FSL interfaces to a comport peripheral that always starts as a transmitter t reset.

If the comport is in receiver mode when you re-launch the software, the system will hang. One way of doing it is to use the Diamond server and start the Diamond app again, as a reset is issued by default by the server before loading the app.