`

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.

Revision History

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

Table of Contents

Table of Figures

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.

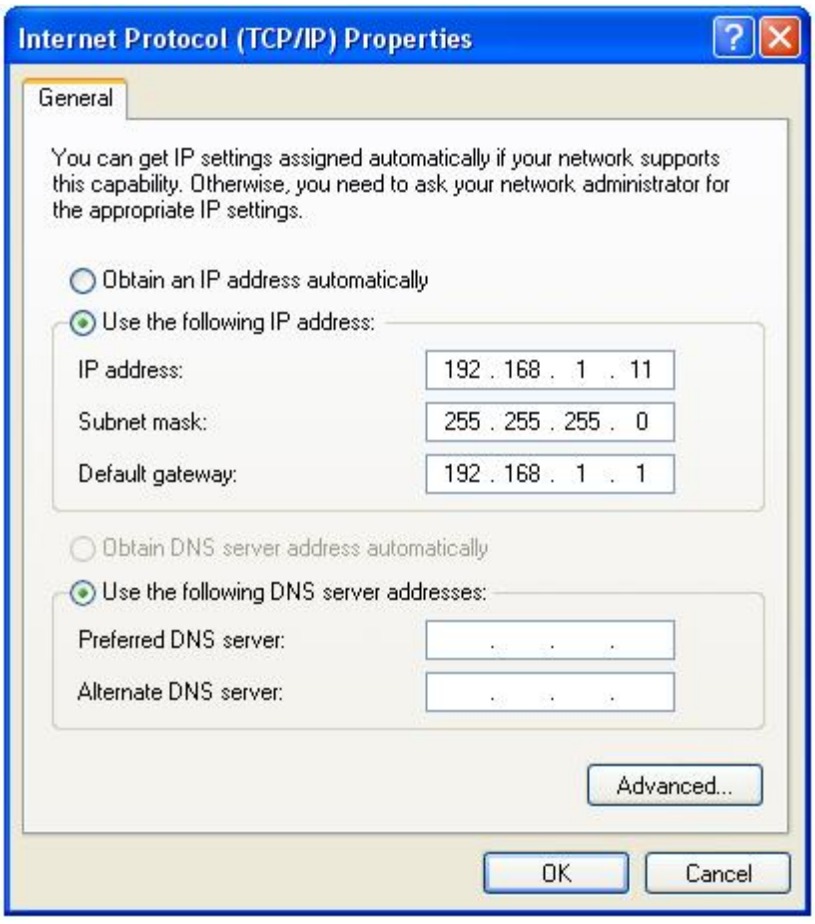

Figure 1 : TCP/IP configuration

5.3 The Webserver

What is in the design?

- PPC running at 300MHz,
- Kernel working on the PowerPC with cache memory,
- Multithreading functionnality,
- 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.

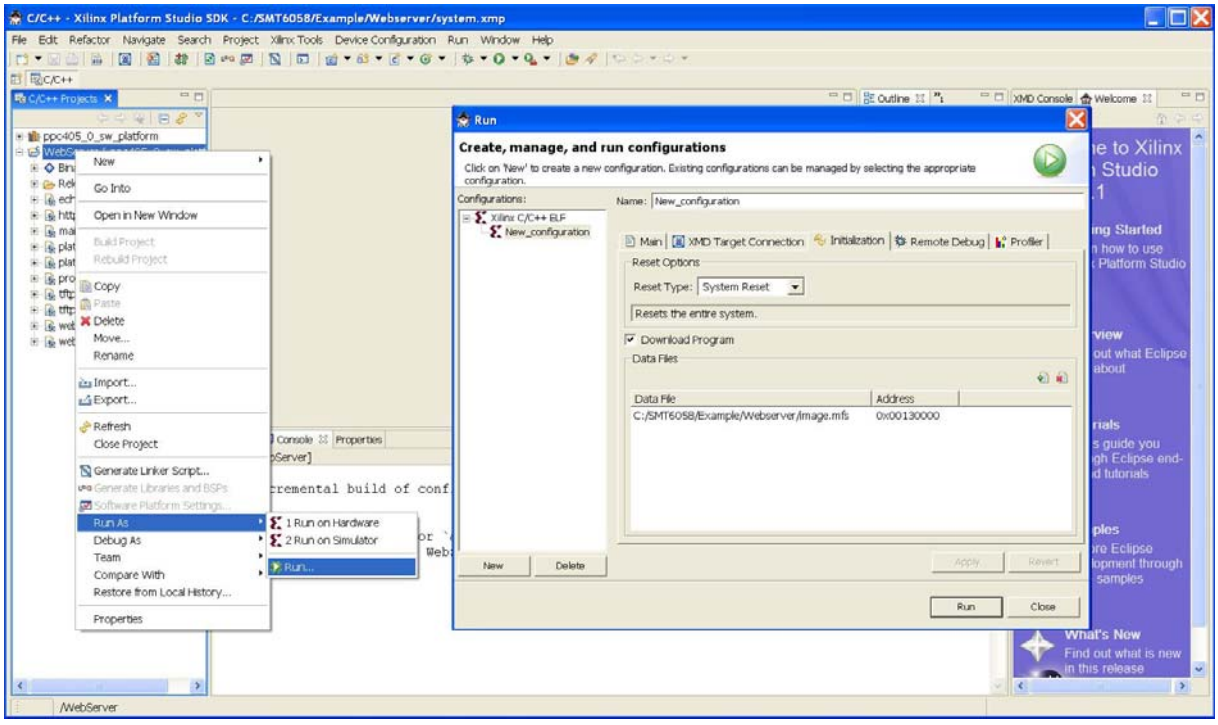

Figure 2 : image.mfs initialization

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.

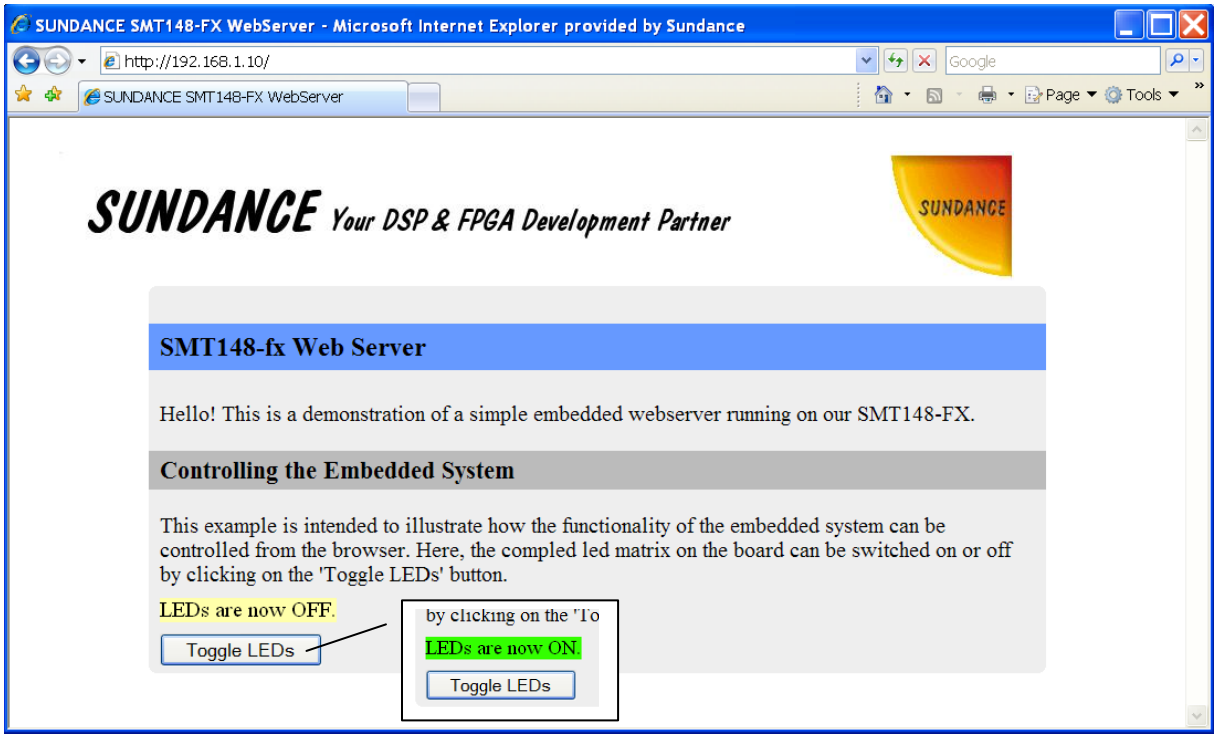

Figure 3 : SMT148-FX Web Server

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.

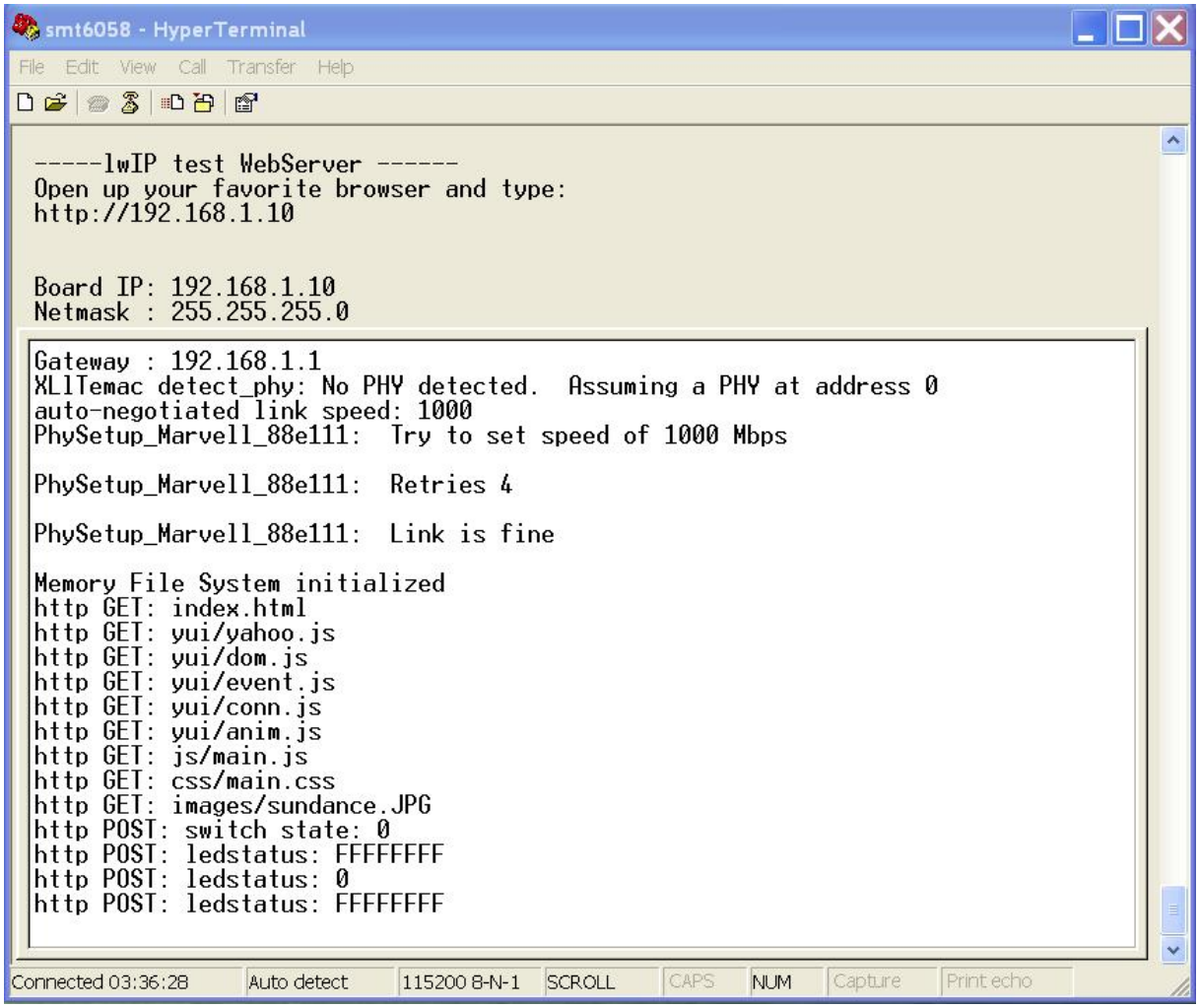

Figure 4 : HyperTerminal

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 src="images/sundance.JPG" style="padding-left:20px"></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

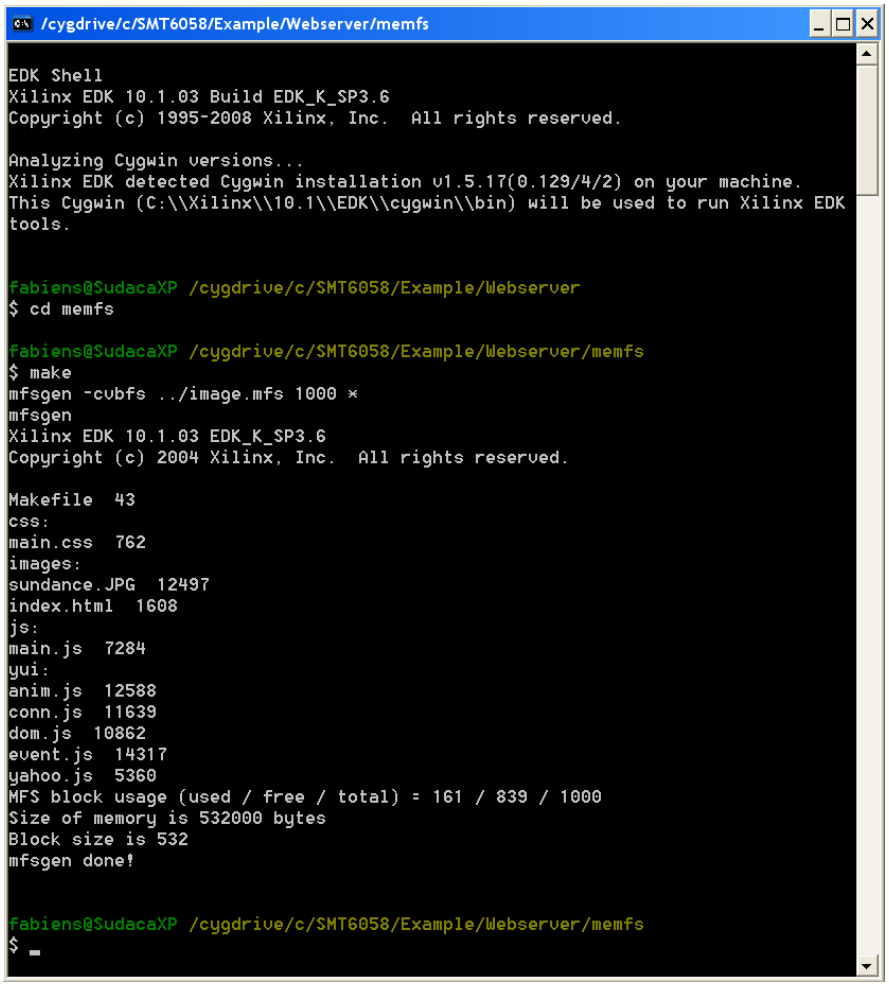

Figure 5 : image.mfs generation

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.

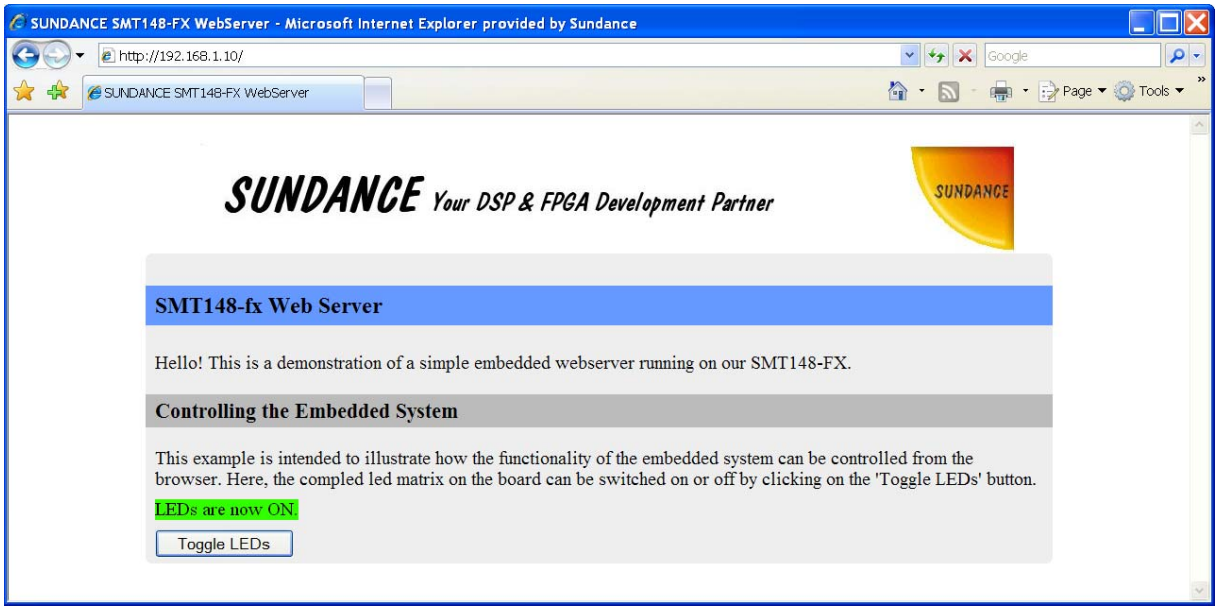

Figure 6 : Web Server modified

5.4 The Ethernet with SMT362 comport loopback

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

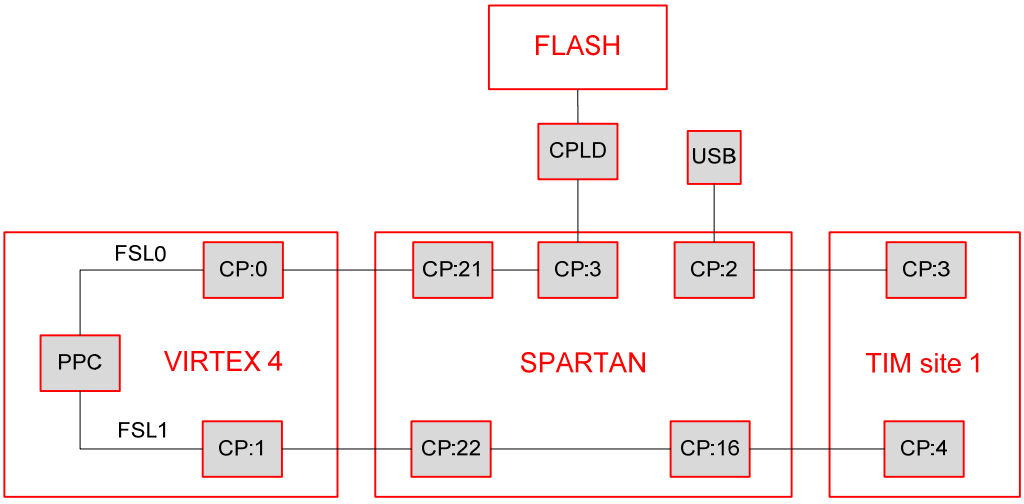

Figure 7 : Virtex 4 comports connections

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.

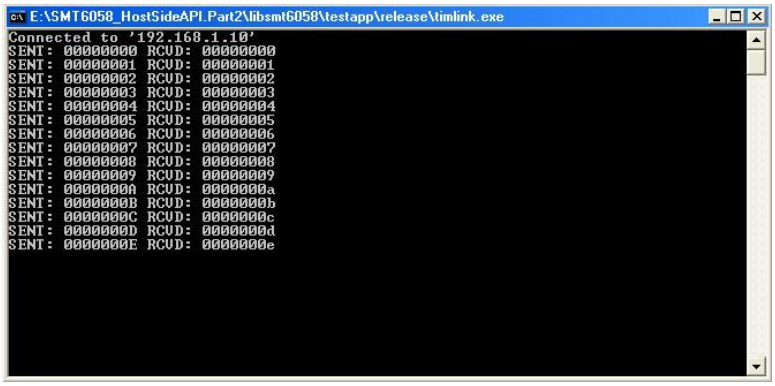

Figure 8 : Host application

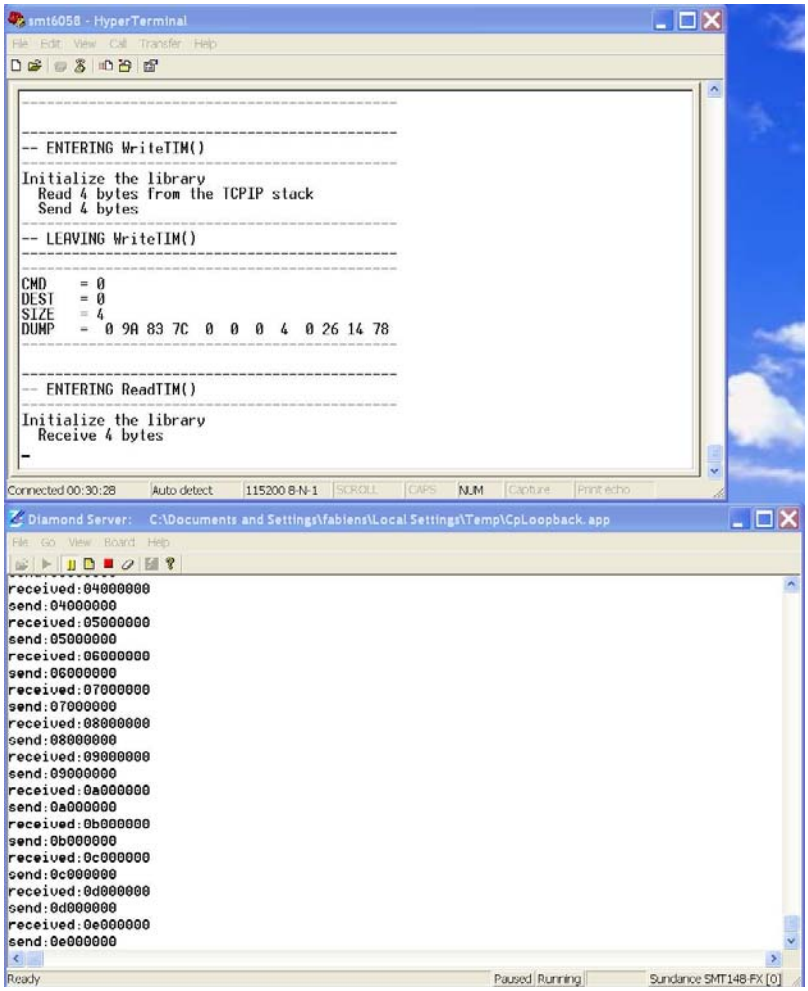

Figure 9 : Debug and 3L server result

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.