SheevaPlug Development Kit README

Rev. 1.2

SheevaPlug Development Kit README

Introduction

The SheevaPlug development packages are provided to assist developers in developing applications utilizing the SheevaPlug development kit platform for plug computing. This README file provides a brief description of the packages.

1. **Documentation Package:** This package consists of the documents needed to bring up SheevaPlug development kit platform and SheevaPlug reference design guide.
2. **Schematics and Bill of Materials:** This package provides the schematics, layout and BOM necessary to design SheevaPlug development kit reference boards.
3. **U-Boot:** This package consists of the files and source codes needed to build U-Boot, initial bring-up of SheevaPlug development kit reference design as well as upgrading the U-Boot.
4. **Linux Support Package:** This package provides the files and sources needed to build and configure the Linux kernel 2.6.22.18 with Marvell's LSP. It also includes a compiled uImage to bring-up SheevaPlug development kit reference design.
5. **File System:** This package consists of Ubuntu's jaunty filesystem image.
6. **Host Software Support Package:** This package consists of support software for Windows host and Linux host.

    a. **Windows Host:** This package consists of the mini-USB to USB driver that needs to be installed on the Windows host in order to access the debug console for SheevaPlug development kit.
    b. **Linux Host:** This package consists of the GCC crosscompiler that needs to be installed on the Linux Host. The package also consists of the basic root filesystem that needs to be installed on the Linux host to initially boot SheevaPlug development board using NFS.
7. **USB Recovery:** This package consists of the 'flashware.img' image, uImage.sheeva.040309 and u-boot binary needed to bring up the SheevaPlug from USB key attached to the USB port on the SheevaPlug. This image will help in writing the uImage and the Filesystem to the NAND flash by booting up the SheevaPlug from U-Boot.

Flow to use the Software Development Kit packages

Below is the flow to use SheevaPlug's software development kit packages to get started with bringing up SheevaPlug development kit reference design. The flow is based on using the compiled images already provided in the above-mentioned packages.

1. Copy the rootfs.tar.bz2 package from 'Host Software Support Package' to your Linux host and extract all of its contents to a particular directory. This directory path will act as your **rootpath** for NFS boot of SheevaPlug development kit.
2. Copy the LSP package contents to the same folder as above.
3. Install the GCC cross compiler on the Linux host.
4. Install the driver using the package 'CDM 2.04.16_SHEEVA' from 'Host Software Support Package' for the USB debug console on the Windows host.
5. Follow the instructions in step 4 of section '3' in the document '*SheevaPlug - Writing Jaunty Filesystem on the NAND flash-Rev1.2.pdf*' document to flash the U-Boot from the U-Boot package.
6. See section '2' in '*SheevaPlug - Writing Jaunty Filesystem on the NAND flash-Rev1.2.pdf*' for boot from NFS.

7. Follow the instructions of the '*SheevaPlug - Writing Jaunty Filesystem on the NAND flash-Rev1.2.pdf*' document to flash the filesystem on the NAND of SheevaPlug development kit.
8. Alternatively, you can boot the SheevaPlug from USB using the USB recovery mechanism and write the uImage and the filesystem images to the NAND. Follow the document 'SheevaPlug Development Kit - USB Flash Recovery from U-Boot-Rev1.0.pdf' for the procedure.

For more information and regular updates on SheevaPlug development kit, visit the links below:

http://www.marvell.com/products/embedded_processors/developer/kirkwood/sheevaplug.jsp

OR

http://plugcomputer.org/

Appendix A

GCC cross-compiler

Follow the instructions to install the gcc cross-compiler on the Linux host system. Path for the gcc.tar.bz2 package is *../SheevaPlug_Host_SWsupportPackage/ LinuxHost/*

1. Copy gcc.tar.bz2 to the home directory on the host system (/home)

2. Go to the home directory.
   a. [root@localhost ~]# cd /home

3. Untar the gcc package
   a. [root@localhost home]# tar -xjvf gcc.tar.bz2

4. Add the gcc path to the working directory
   a. [root@localhost home]# export PATH=/home/gcc/bin:$PATH

5. Copy “mkimage” binary located in the ***‘SheevaPlug_U-Boot’*** package in the CD (path - /u-boot/tools/mkimage) to HOST Linux machine in the “/usr/bin” directory.

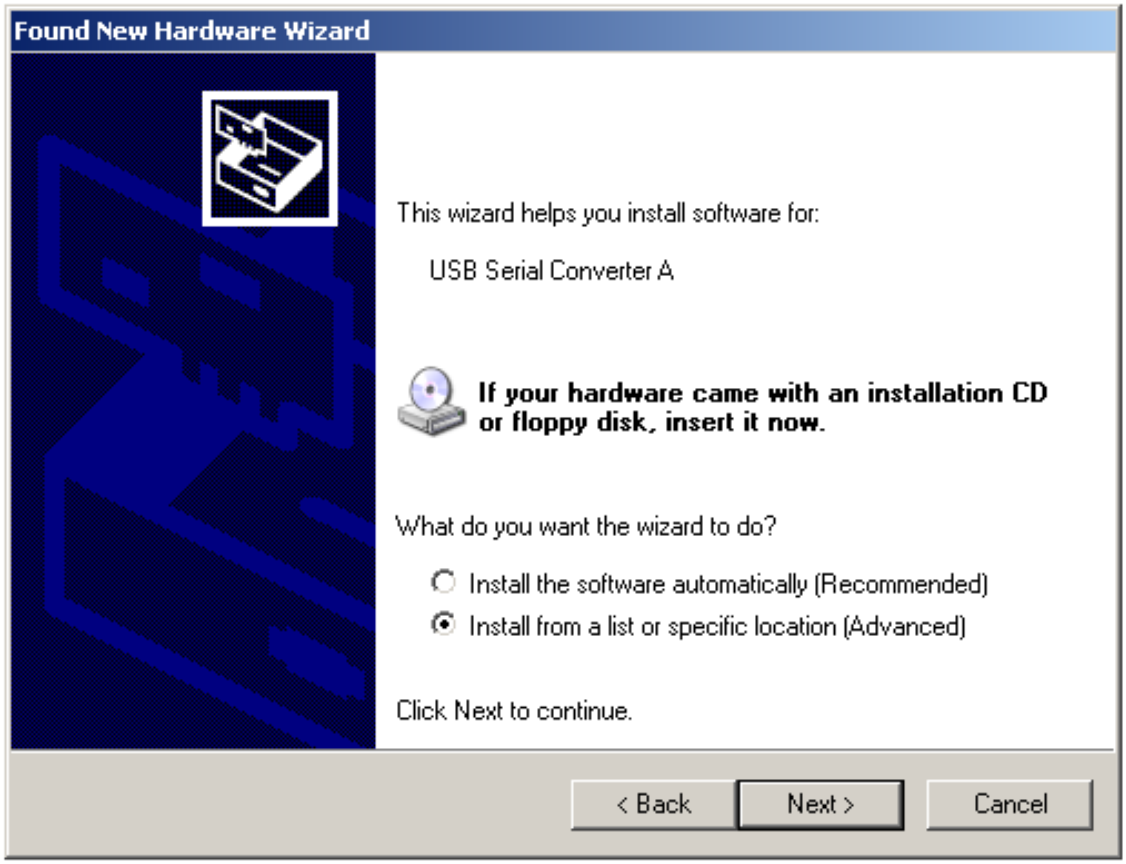
Appendix B

Mini-USB debug driver installation in Windows

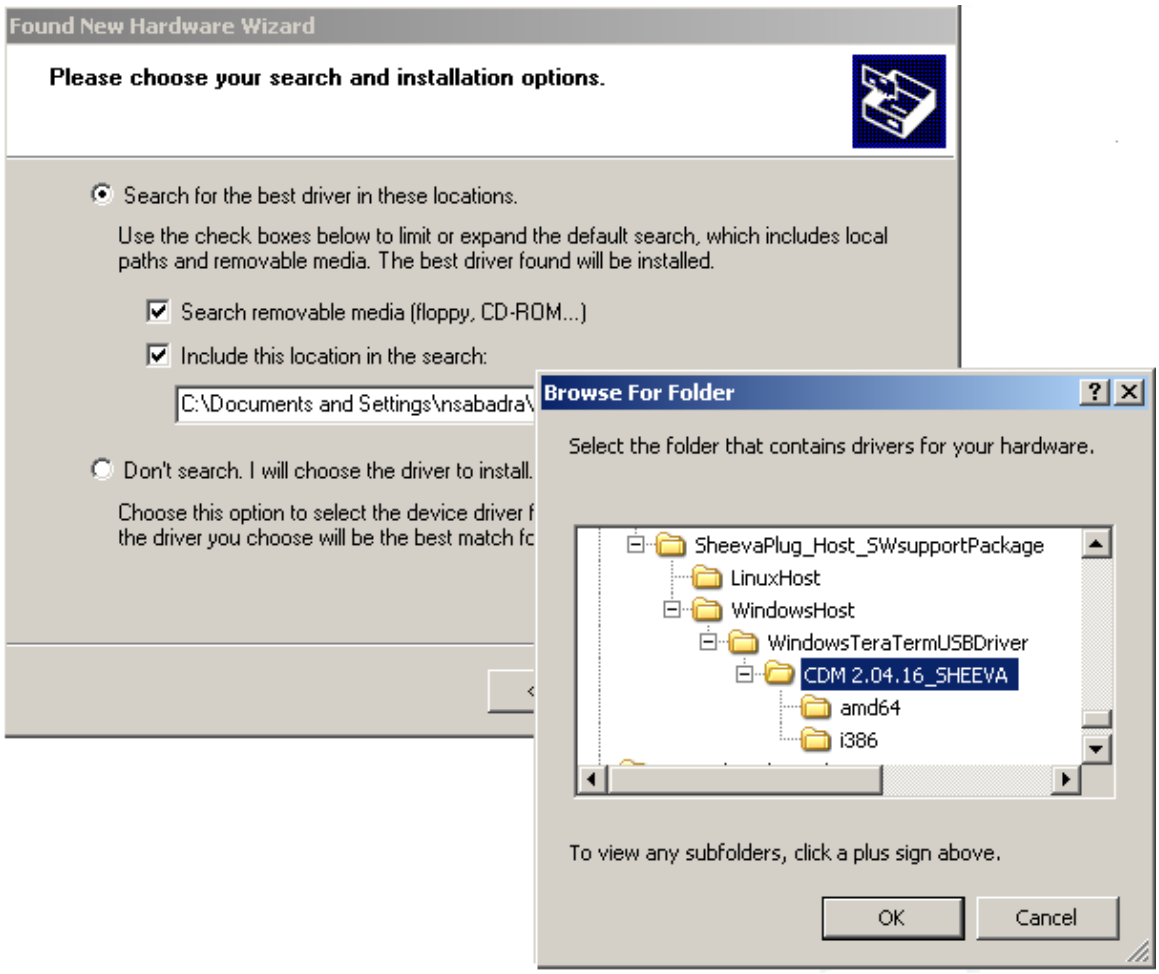
The following steps will help you to install the mini-USB debug driver successfully in order to access the debug console on a Windows host. The procedure below shows gives an example of the installation in Windows XP. The procedure is similar in Vista.

1. Download the *'WindowsTeraTermUSBDriver.zip'* package from the SheevaPlug_Host_SWsupportPackage/WindowsHost.
2. Unzip the package.
3. Connect the USB connector of the debug cable to the Windows machine and the mini-USB connector to the mini-USB slot on the SheevaPlug.
4. Use Windows Found New Hardware Wizard to install the drivers as shown below:
    a. Select the option as shown below and click 'Next'.

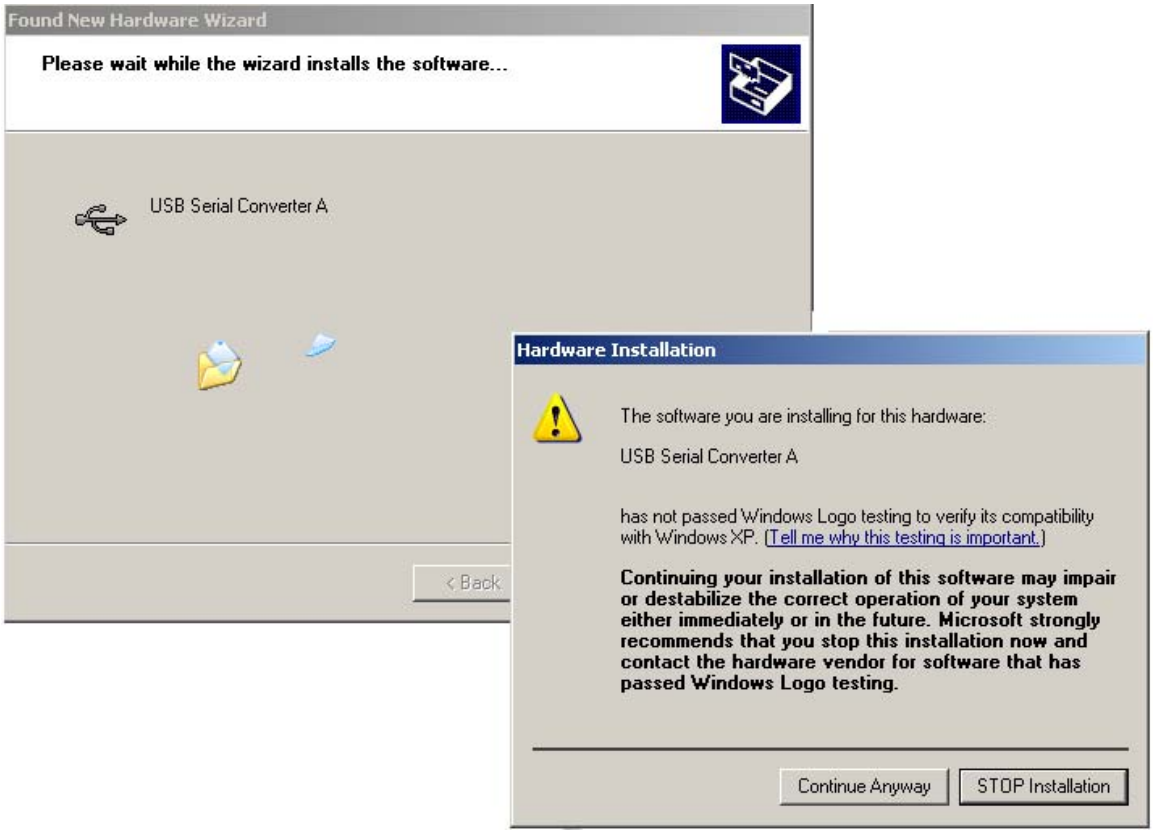
b. Select the option as shown below and click ‘Next’.



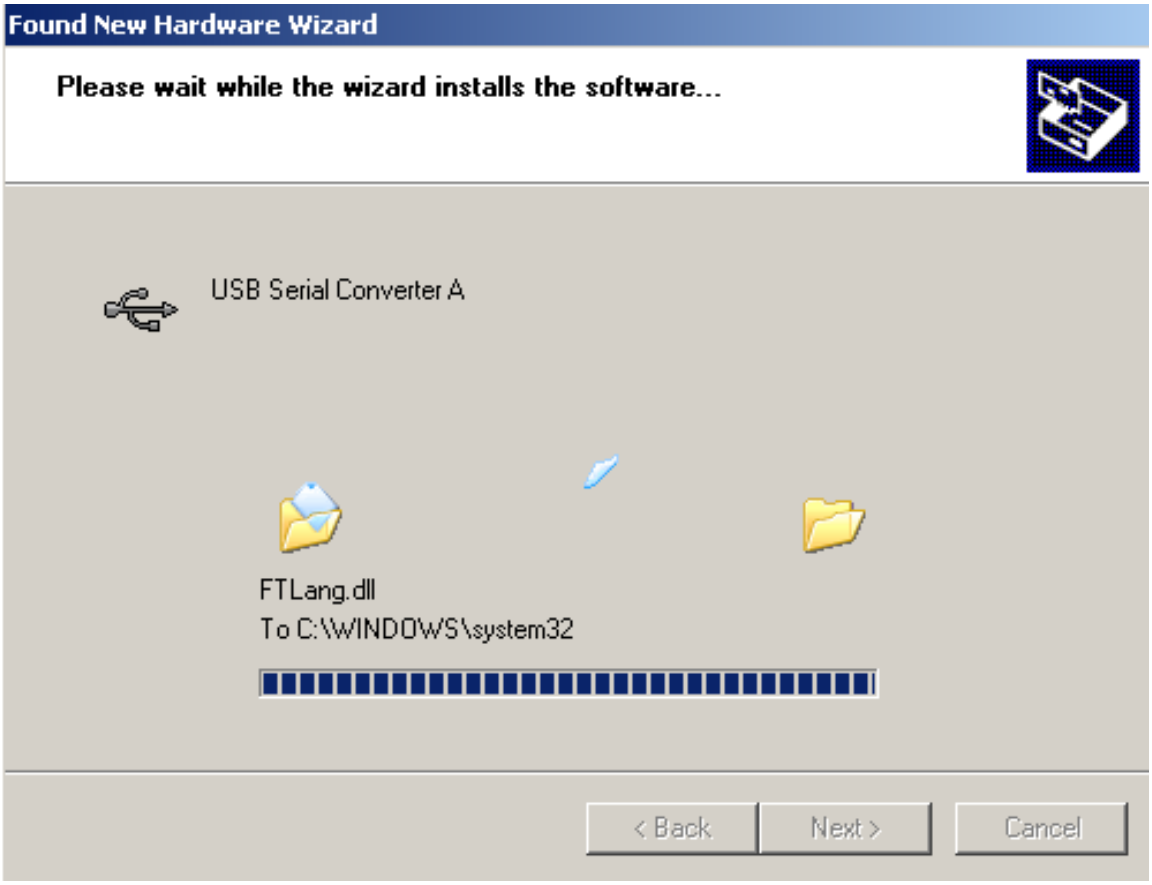
c. Point to the target location where the *WindowsTeraTermUSBDriver.zip* has been downloaded and unzipped.



d. Click on the button ‘Continue Anyway’ to install the driver.



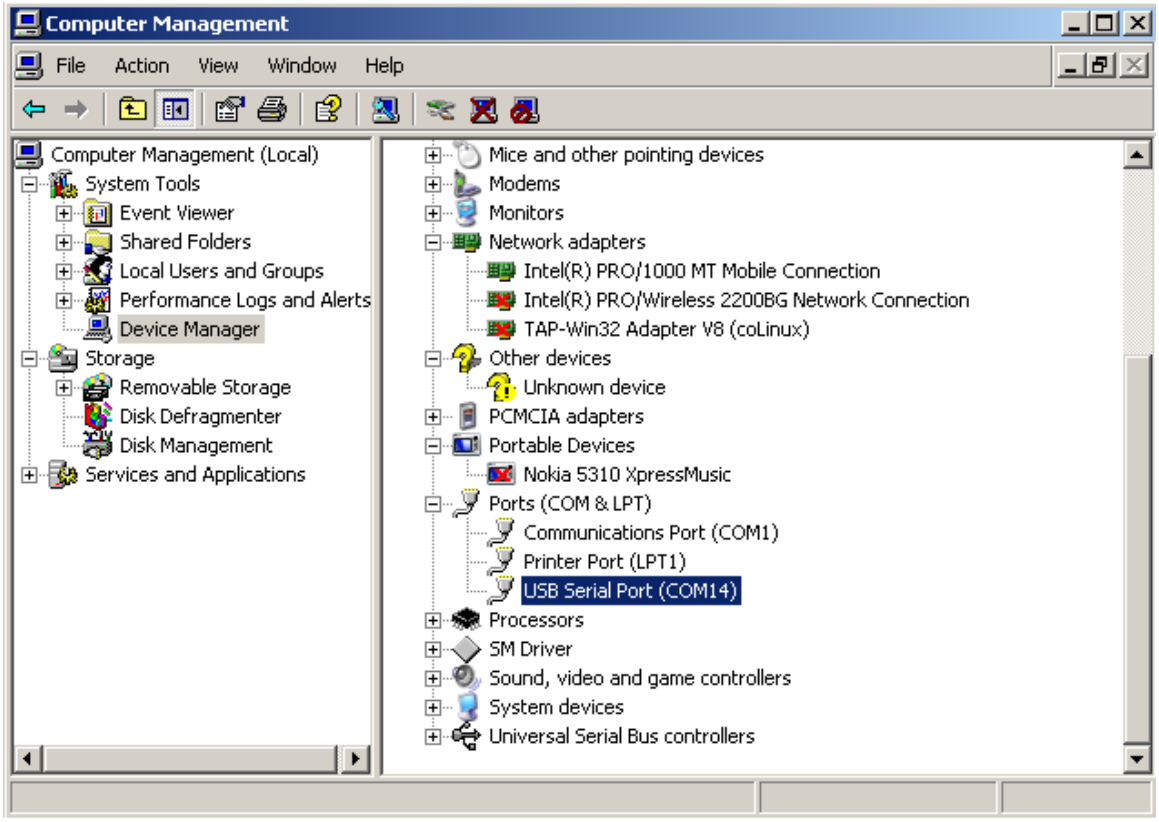
e. You can see that the drivers for ‘USB Serial Converter A’ is being installed.



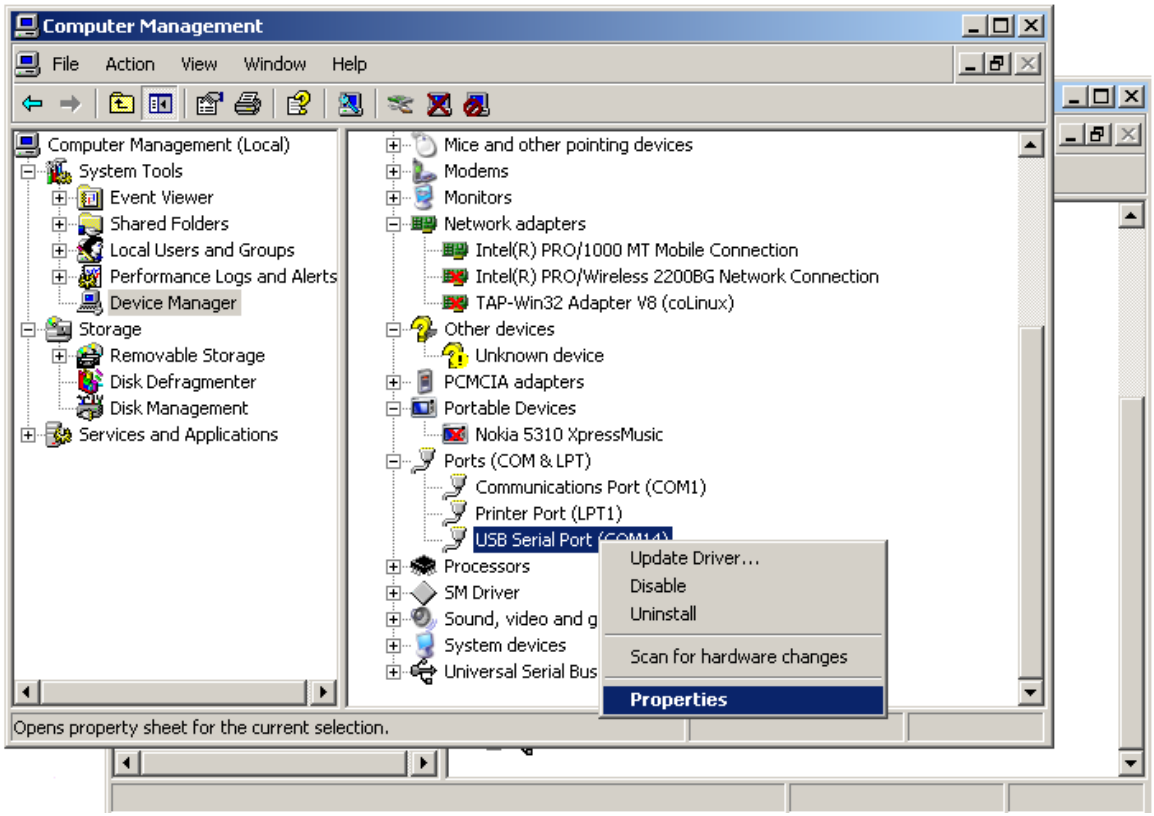
f. Click on ‘Finish’ button.

g. Continue with previous steps ***a – f*** to install ‘USB Serial Converter B’ and ‘USB Serial Port’.

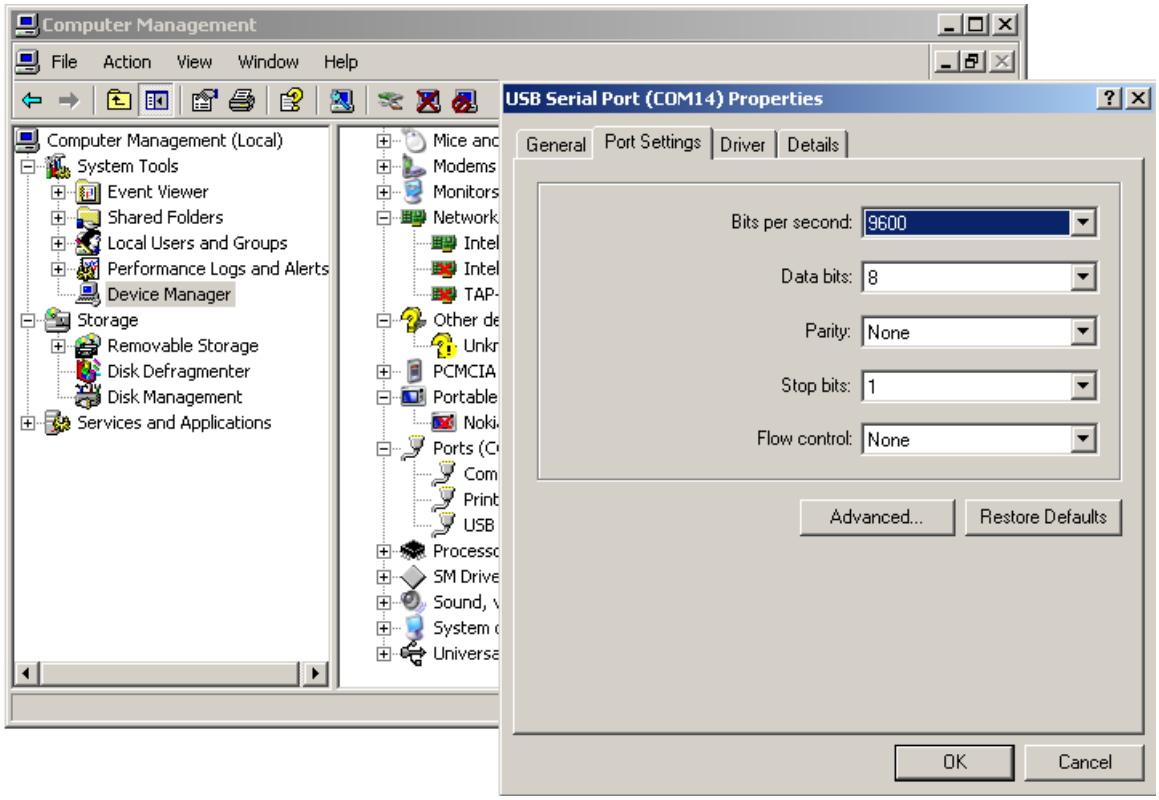
h. After installing the USB Serial Converters A and B and installing the USB Serial Port, go to 'Device Manager' to configure the USB port, as shown below.



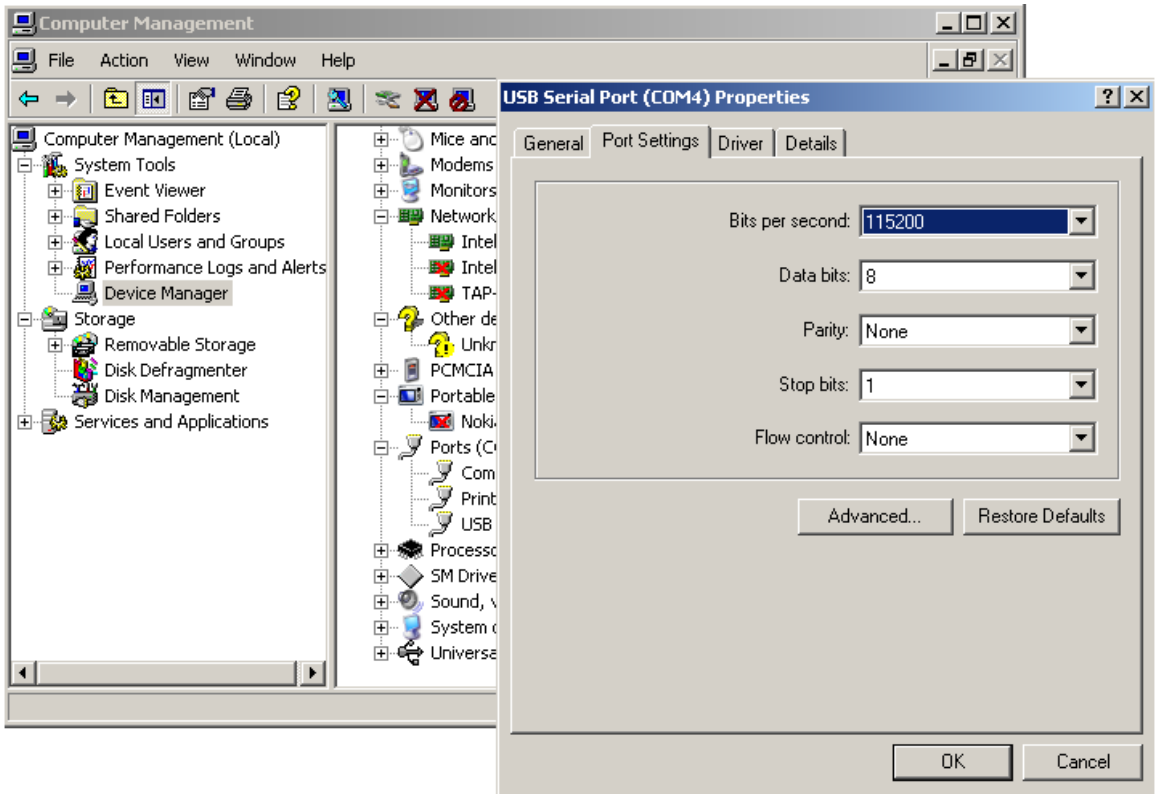
i. Right click on the USB Serial port and Select ‘Properties’.



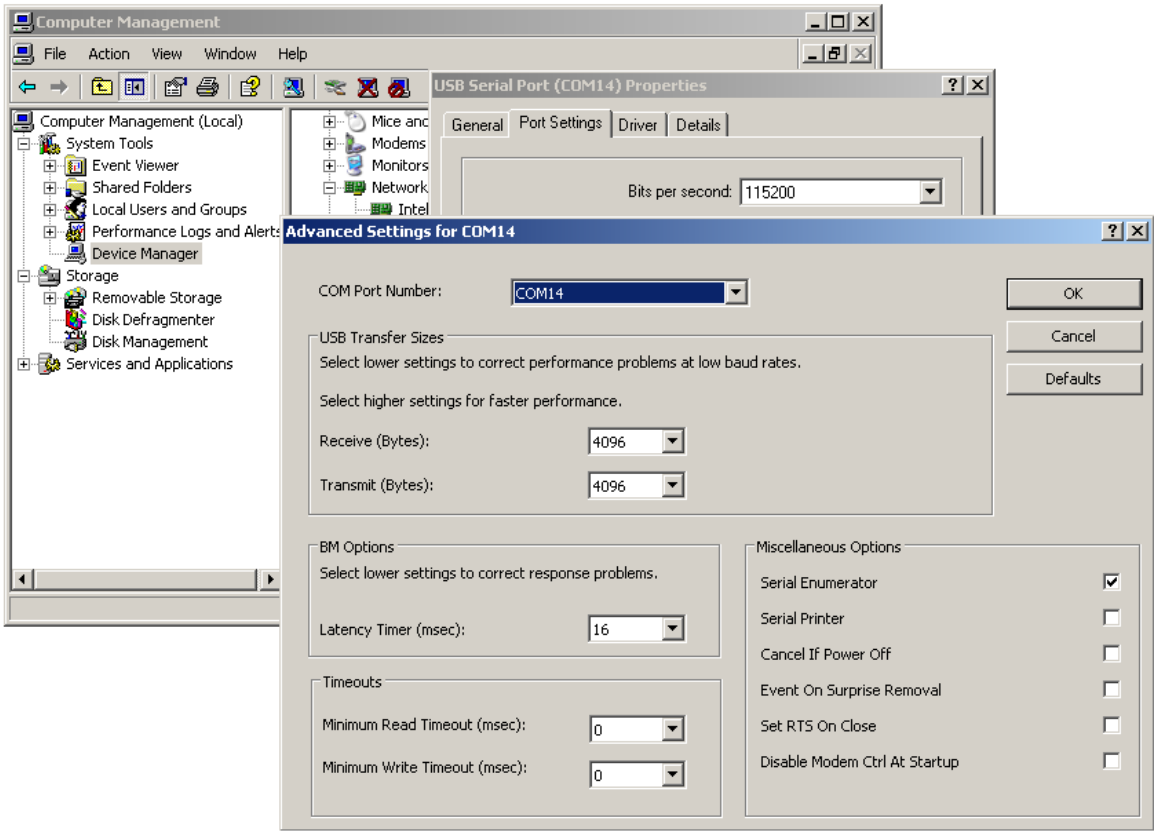
j. In the pop-up window, click on ‘Port Settings’ tab.



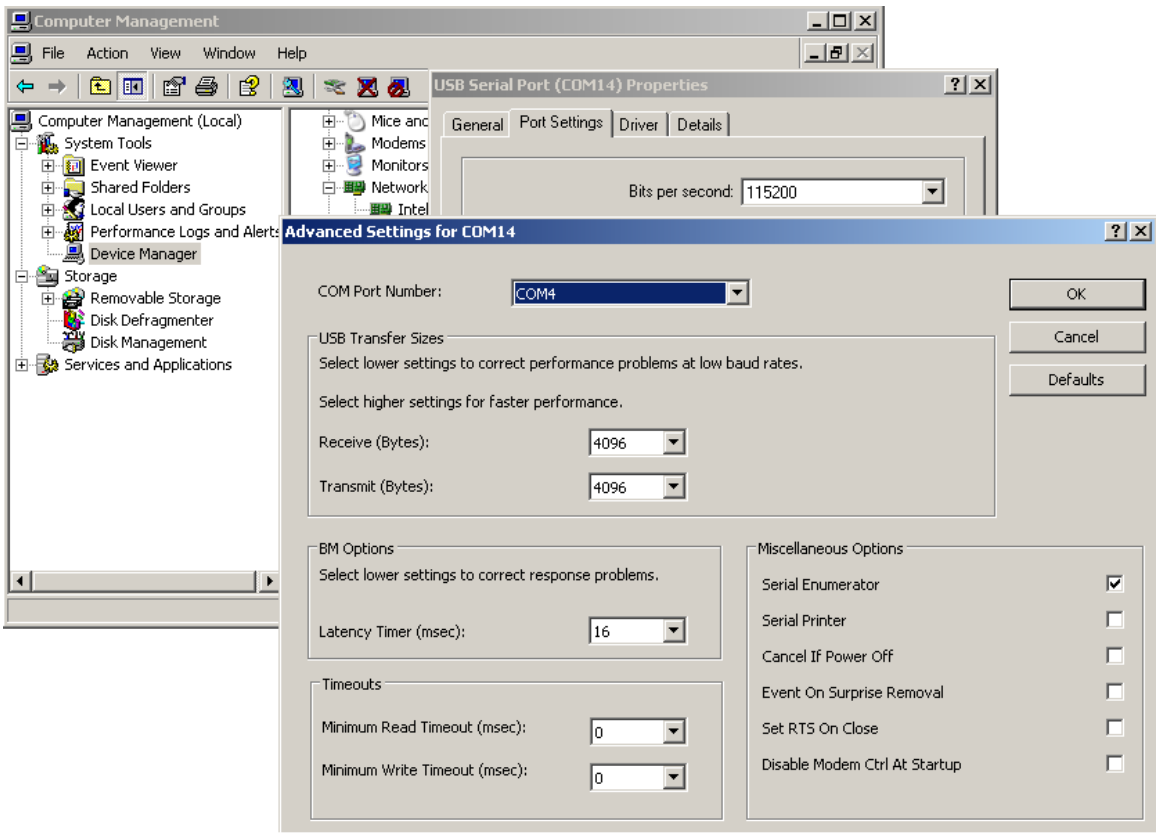
k. Change the value of the field ‘Bits per Second’ from 9600 to 115200.



I. Click on ‘Advanced’ button. A new window will pop-up as shown below.



m. Change the 'COM Port Number' field to a COM port number between 1 – 4.
Note: Please note that this change in COM port number is only required if TeraTerm application is used to access the debug console and the value of the parameter 'MaxComPort' has not been changed.



n. Click on 'OK' buttons on the opened pop-up and exit out of the properties window.

o. On refreshing the 'Device Manager' tab, the new COM port number will assigned to the 'USB Serial Port'.