

# **LynuxWorks BlueCat Linux on the Avnet V5FX30T Evaluation Board**

February, 2009  
Version 1.0



## Revision History

Version	Description	Date
1.0	Initial draft release	Feb 9, 2009

## Overview

This document describes a PPC440 design implemented and tested on the Avnet Virtex 5FXT Evaluation board. This processor system boots a pre-built LynuxWorks BlueCat Linux kernel image. This kernel image was built using LynuxWorks BlueCat Linux version 5.4.2 for the Xilinx Virtex-5FXT PPC440 processor.

## Objectives

This reference design will demonstrate running BlueCat Linux on the Avnet V5FX30T evaluation board.

## Requirements

This reference design will require the following software and hardware setup.

### Software

The software requirements for this reference design are:

- Windows XP
- Xilinx ISE 10.1 with Service Pack 3
- Xilinx EDK 10.1 with Service Pack 3

### Hardware

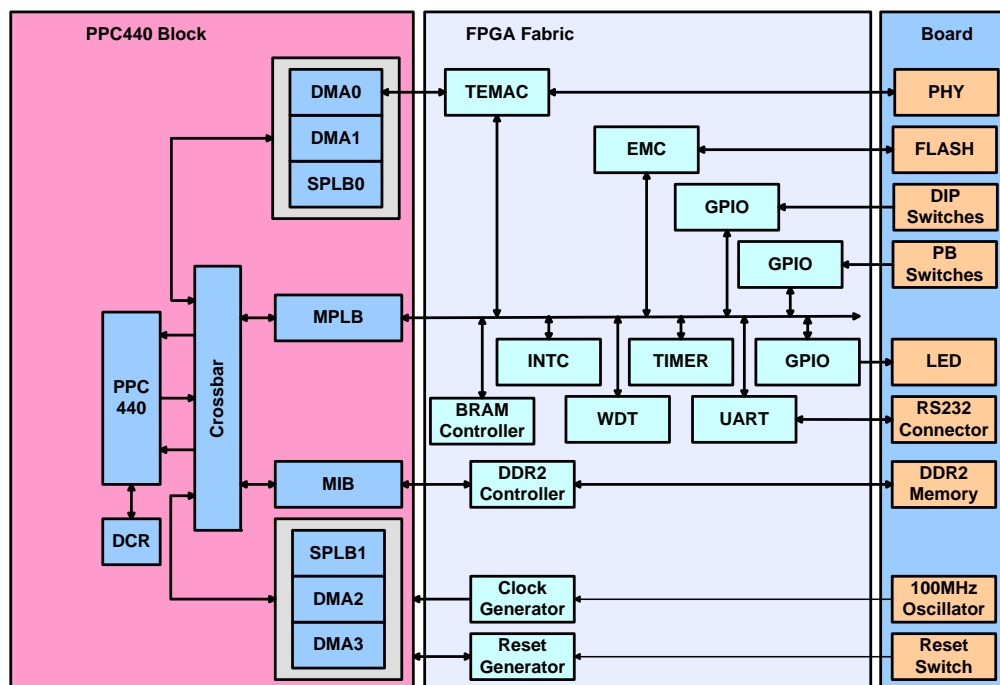
The hardware setup for this reference design is:

- Computer with 1 GB RAM and 1 GB virtual memory (recommended)
- Avnet Virtex-5 FXT evaluation board
- Straight through RS232 cable
- Ethernet cable
- Power supply
- JTAG programming cable (USB or PC4)

## Processor System Block Diagram

The following figure shows a high-level block diagram of the processor system in the FPGA. The design consists of:

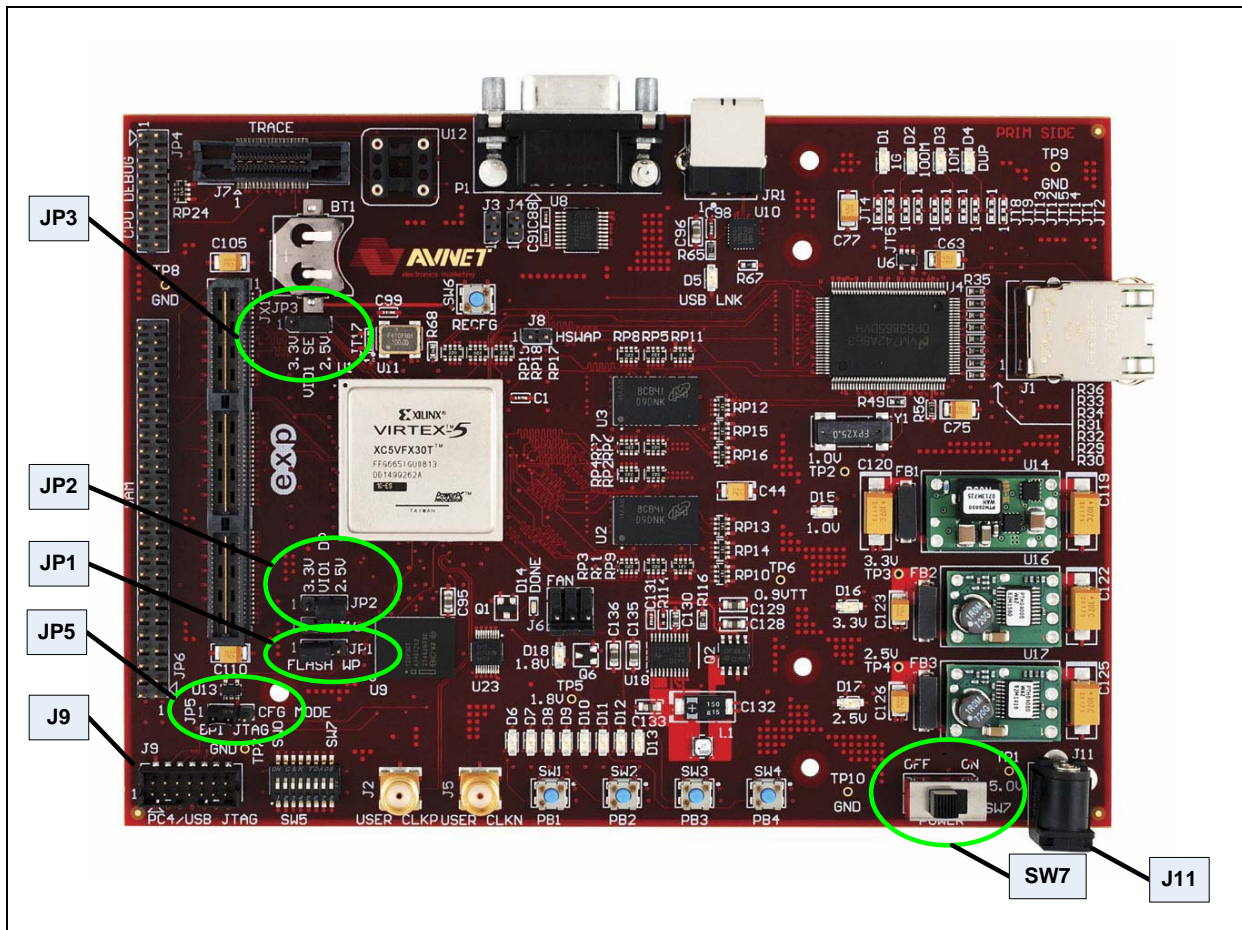
- PowerPC processor
- 64KB of BRAM
- 64MB of DDR2 SDRAM
- 16MB of Flash
- TEMAC Core (implemented with scatter-gather DMA)
- RS232 Port
- 8-position DIP Switch
- Push-button switches
- LEDs
- Timer
- Watchdog timer
- Interrupt Controller



## Board Setup

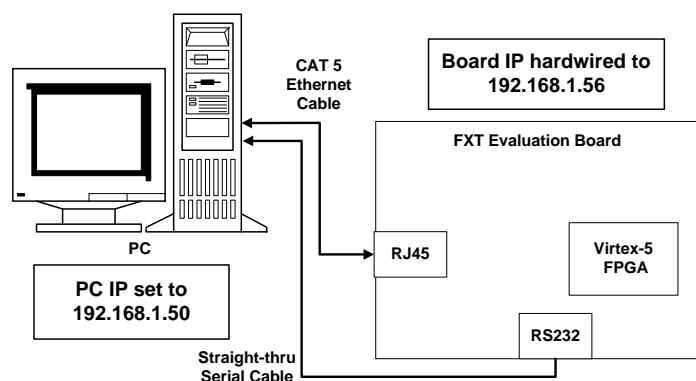
The following figure shows the location of various connectors and jumpers on the Avnet Virtex-5 FXT evaluation board. Perform the following steps to setup the board for running the application software.

1. Verify the Power switch, **SW7**, is in the **OFF** position.
2. Install a jumper on JP3 pins 2-3
3. Install a jumper on JP2 pins 2-3
4. Install a jumper on JP1 pins 1-2
5. Install a jumper on JP5 pins 2-3 (FPGA JTAG mode)
6. Connect the power supply to the J11 connector on the FXT evaluation board and also plug it into the AC outlet.
7. Connect the USB JTAG cable to J9 and the USB port of the PC.
8. Connect a straight through RS232 cable to the board DB-9 connector (P1) and the serial port of the PC. Alternatively, you can use an RS232-USB adapter and connect this adapter to the DB-9 connector and the USB port of the PC. In this case, you must install the RS232-USB driver for the adapter.
9. Slide the power switch to the **ON** position
10. Connect an Ethernet cable to J1 and the Ethernet port of the PC (the Ethernet cable must be connected directly to the PC or via a hub).



## PC Setup

Use the Windows XP Control Panel to configure the host PC with an IP address of **192.168.1.50**, with a subnet mask of **255.255.255.0**



## Running The Demo Files

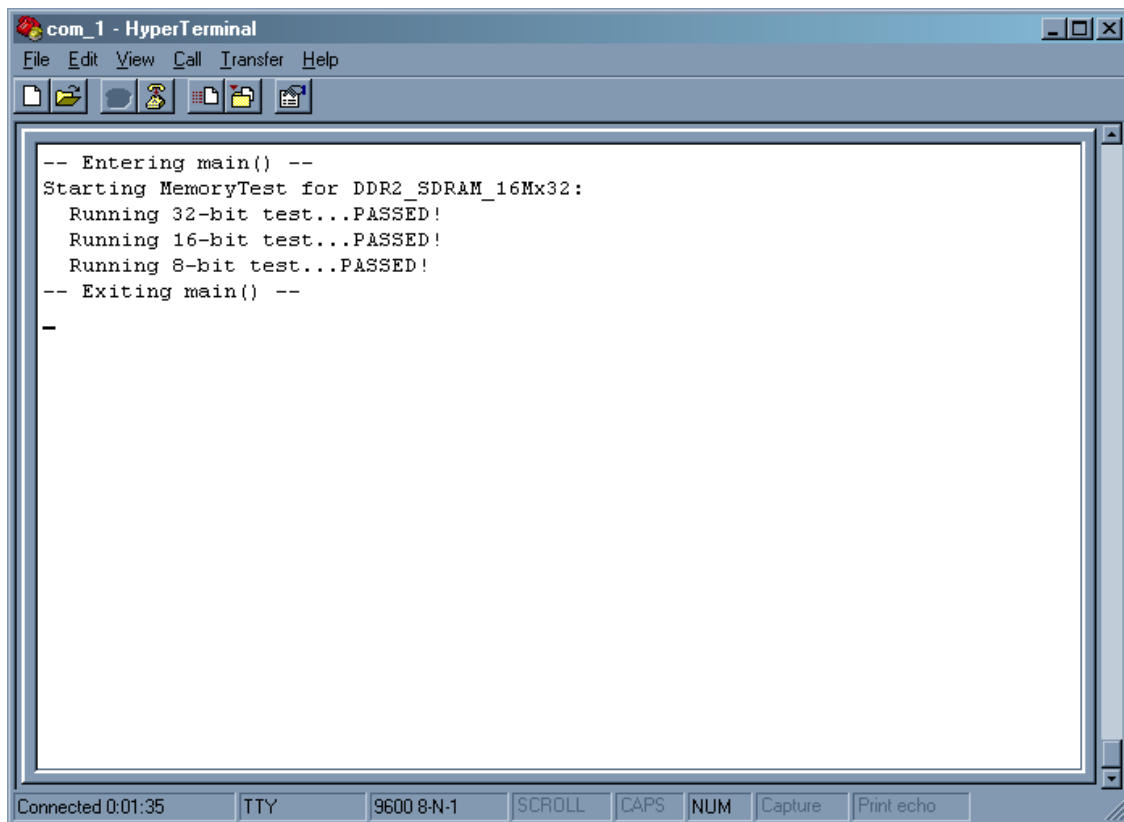
You can load the FPGA and run the test applications without building the design by using the demo script and the pre-built bit and elf files. You must have the Xilinx tools installed on your host, and have the hardware set up and connected as per the previous steps.

1. Start a HyperTerminal session and set the serial port parameters to 9600 baud rate, 8 bits, 1 stop bit, no parity and no flow control.
2. Double-click on **demo\_linux.bat** in the demo\_files directory. This batch file will load the FPGA with the hardware bistream and TestApp\_Memory application and then run XMD to download and boot the BlueCat Linux kernel image on the board. The command line window that opens will close after executing the script.

Note: There is a known issue that the gigabit (Gb) Ethernet operation on the Avnet V5FX30T board does not work. If you have a Gb-capable NIC on the host and are connecting directly to the board, you need to restrict the transmission rate to 100 Mbps Half-duplex. Change this setting through the Windows host Control Panel. Go to **Start → Control Panel → Network Connections → Local Area Connection** and select the **Properties** button. On the **General** tab select the **Configure** button. Now select the **Advanced** tab and look for a setting named something like **Link Speed & Duplex** and select **100Mbps/Half Duplex** among the drop-down options. Alternately you can connect the board and host to a 10/100Mbit router and let auto-negotiate determine the link.

## Implementing The Design

1. Start a HyperTerminal session and set the serial port parameters to 9600 baud rate, 8 bits, 1 stop bit, no parity and no flow control.
2. Open the Linux system design in XPS and select **Software > Build All User Applications** to compile the software.
3. Select **Device Configuration > Update Bitstream** from the XPS GUI to build the design.
4. Select **Device Configuration > Download Bitstream** from the XPS GUI to download the **TestApp\_Memory** application to the board. The memory test program will run on the board and you should see the following on the HyperTerminal:



The screenshot shows a HyperTerminal window titled "com\_1 - HyperTerminal". The window has a menu bar with "File", "Edit", "View", "Call", "Transfer", and "Help". Below the menu bar is a toolbar with icons for file operations and communication. The main text area displays the following output:

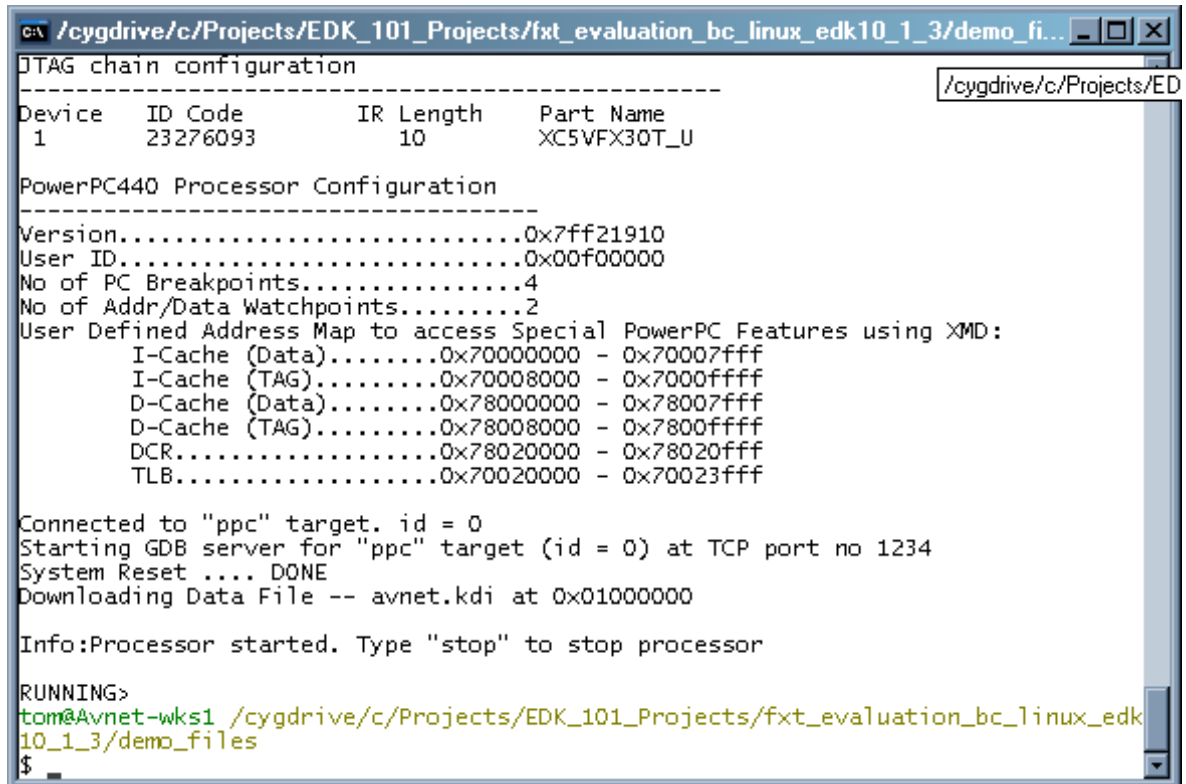
```
-- Entering main() --  
Starting MemoryTest for DDR2_SDRAM_16Mx32:  
  Running 32-bit test...PASSED!  
  Running 16-bit test...PASSED!  
  Running 8-bit test...PASSED!  
-- Exiting main() --  
-
```

At the bottom of the window, there is a status bar with the following information: "Connected 0:01:35", "TTY", "9600 8-N-1", "SCROLL", "CAPS", "NUM", "Capture", and "Print echo".

- Run the provided XMD debugger script file to download the Xilinx kernel image to the V5FX30T board. Downloading the kernel image to the board will take a few minutes. Select **Project > Launch EDK Shell** and enter the following command to run the script file:

**xmd**

The XMD commands will run and should look similar to the window in the following figure:



```

C:\ /cygdrive/c/Projects/EDK_101_Projects/fxt_evaluation_bc_linux_edk10_1_3/demo_files
DTAG chain configuration
-----
Device      ID Code      IR Length  Part Name
  1         23276093      10        XC5VFX30T_U

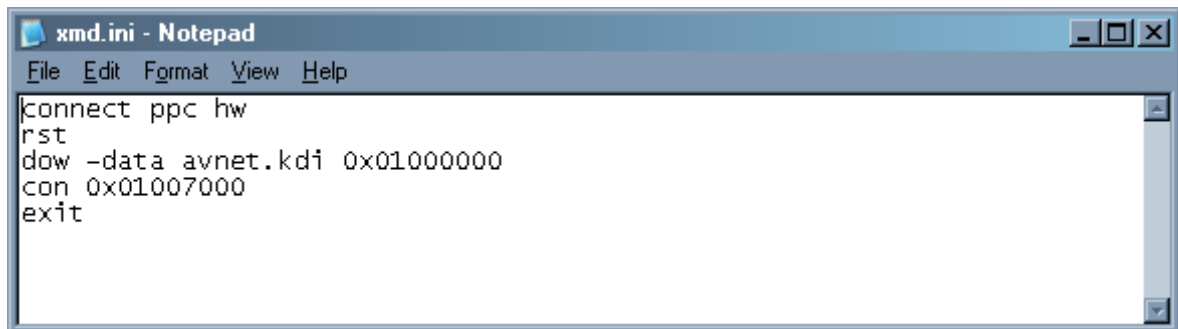
PowerPC440 Processor Configuration
-----
Version.....0x7ff21910
User ID.....0x00f00000
No of PC Breakpoints.....4
No of Addr/Data Watchpoints.....2
User Defined Address Map to access Special PowerPC Features using XMD:
    I-Cache (Data).....0x70000000 - 0x70007fff
    I-Cache (TAG).....0x70008000 - 0x7000ffff
    D-Cache (Data).....0x78000000 - 0x78007fff
    D-Cache (TAG).....0x78008000 - 0x7800ffff
    DCR.....0x78020000 - 0x78020fff
    TLB.....0x70020000 - 0x70023fff

Connected to "ppc" target. id = 0
Starting GDB server for "ppc" target (id = 0) at TCP port no 1234
System Reset .... DONE
Downloading Data File -- avnet.kdi at 0x01000000

Info:Processor started. Type "stop" to stop processor

RUNNING>
tom@Avnet-wks1 /cygdrive/c/Projects/EDK_101_Projects/fxt_evaluation_bc_linux_edk
10_1_3/demo_files
$
  
```

When XMD was launched it automatically ran the commands in the xmd.ini file in the project directory. The xmd.ini file is a simple text file as shown here:

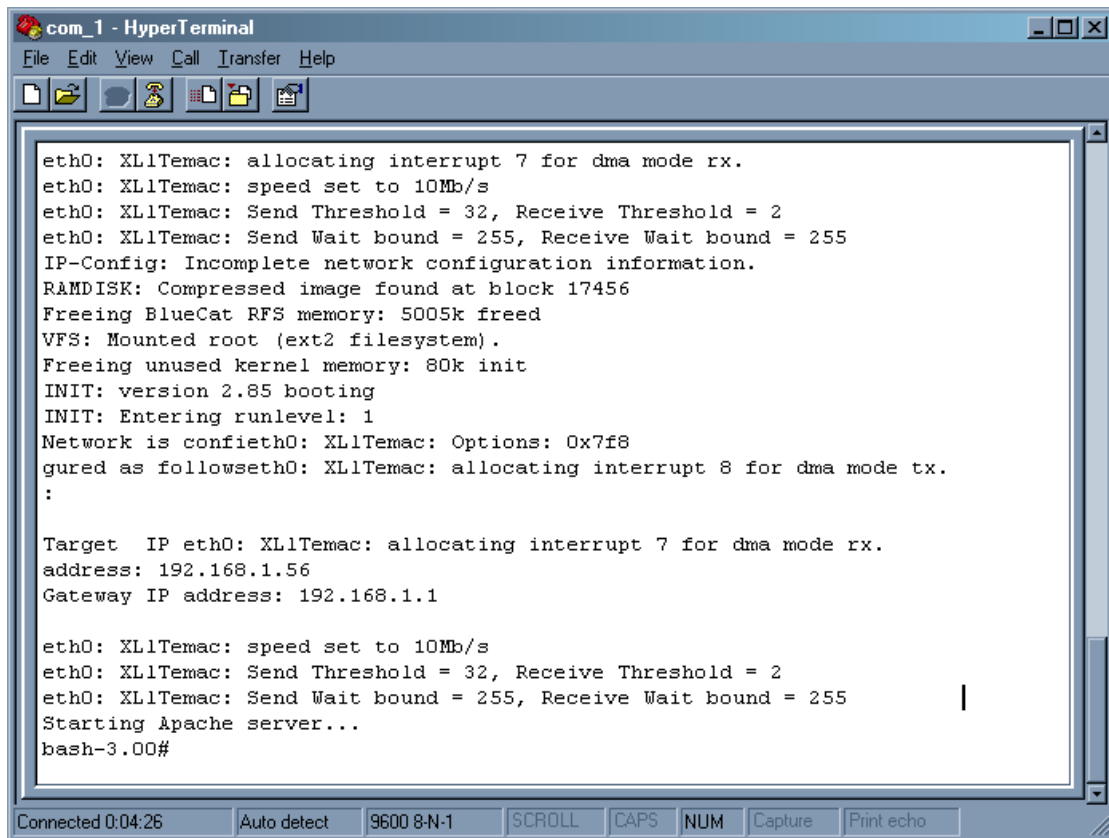


```

xmd.ini - Notepad
File Edit Format View Help
connect ppc hw
rst
dow -data avnet.kdi 0x01000000
con 0x01007000
exit
  
```



6. Once the Linux kernel is done booting, you should see the following on the HyperTerminal:



```
com_1 - HyperTerminal
File Edit View Call Transfer Help

eth0: XL1Temac: allocating interrupt 7 for dma mode rx.
eth0: XL1Temac: speed set to 10Mb/s
eth0: XL1Temac: Send Threshold = 32, Receive Threshold = 2
eth0: XL1Temac: Send Wait bound = 255, Receive Wait bound = 255
IP-Config: Incomplete network configuration information.
RAMDISK: Compressed image found at block 17456
Freeing BlueCat RFS memory: 5005k freed
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 80k init
INIT: version 2.85 booting
INIT: Entering runlevel: 1
Network is configured as follows
eth0: XL1Temac: Options: 0x7f8
eth0: XL1Temac: allocating interrupt 8 for dma mode tx.
:

Target IP eth0: XL1Temac: allocating interrupt 7 for dma mode rx.
address: 192.168.1.56
Gateway IP address: 192.168.1.1

eth0: XL1Temac: speed set to 10Mb/s
eth0: XL1Temac: Send Threshold = 32, Receive Threshold = 2
eth0: XL1Temac: Send Wait bound = 255, Receive Wait bound = 255
Starting Apache server...
bash-3.00#
```

Connected 0:04:26 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

7. The Linux kernel now running on the board mounts a RAMdisk that contains the root file system and all available applications. You can verify that the OS is running by inspecting all the running process by typing `ps axl` at the prompt. Feel free to poke around and try out a few Linux commands. The kernel is running the Apache web server, so you can point your browser to <http://192.168.1.56> to see a simple LynuxWorks webpage. The ftp and telnet daemons are also active in the running kernel (login=root), so you can use those applications to interact with the kernel.