OMAP35x LogicLoader™ Labs

LogicLoader Documentation (LoLo version 2.4.8)

Logic // Products
Published: June 2008
Last revised: December 2009

Abstract

These informal LogicLoader™ (LoLo) labs provide an introduction to basic LogicLoader commands and usage on the OMAP35x System on Module (SOM).

This document contains valuable proprietary and confidential information and the attached file contains source code, ideas, and techniques that are owned by Logic Product Development Company (collectively "Logic's Proprietary Information"). Logic's Proprietary Information may not be used by or disclosed to any third party except under written license from Logic Product Development Company.

Logic Product Development Company makes no representation or warranties of any nature or kind regarding Logic's Proprietary Information or any products offered by Logic Product Development Company. Logic's Proprietary Information is disclosed herein pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipment. The only warranties made by Logic Product Development Company, if any, with respect to any products described in this document are set forth in such license or agreement. Logic Product Development Company shall have no liability of any kind, express or implied, arising out of the use of the Information in this document, including direct, indirect, special or consequential damages.

Logic Product Development Company may have patents, patent applications, trademarks, copyrights, trade secrets, or other intellectual property rights pertaining to Logic's Proprietary Information and products described in this document (collectively "Logic's Intellectual Property"). Except as expressly provided in any written license or agreement from Logic Product Development Company, this document and the information contained therein does not create any license to Logic's Intellectual Property.

The Information contained herein is subject to change without notice. Revisions may be issued regarding changes and/or additions.

© Copyright 2009, Logic Product Development Company. All Rights Reserved.

Revision History

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	JCA	Release as individual document	JCA	06/09/08
B	RGL	Updates for LoLo v2.4.8; Lab 4 was changed to boot from SD card; Added Lab 8: Loading an NK.BIN	JCA	06/03/09
C	JCA, RGL	Updates for LoLo v2.4.9; Added language to include Torpedo SOM;	JCA	12/18/09

 Logic Product Development Company, All Rights Reserved

Table of Contents

1 LogicLoader Labs 1
1.1 Lab 1: Kit Communications 1
1.1.1 Abstract 1
1.1.2 What you will need 1
1.1.3 Basic instructions 1
1.1.4 Tera Term instructions 1
1.1.5 HyperTerminal instructions 3
1.1.6 Connect the Kit to your workstation 5
1.2 Lab 2: Learning the LogicLoader Help System 6
1.2.1 Abstract 6
1.2.2 What you will need 6
1.2.3 LogicLoader Interface 6
1.2.4 Help for Incorrect Commands 6
1.2.5 Using the ‘help’ command 8
1.3 Lab 3: Using the LogicLoader Command Line Editor 13
1.3.1 Abstract 13
1.3.2 What you will need 13
1.3.3 Instructions 13
1.4 Lab 4: Updating LogicLoader 15
1.4.1 Abstract 15
1.4.2 What you will need 15
1.4.3 Basic instructions 15
1.4.4 Connect the Kit to your workstation 15
1.4.5 Obtain a new version of LogicLoader 16
1.4.6 Download LogicLoader using a Bootable SD card 16
1.5 Lab 5: Network Connection 21
1.5.1 Abstract 21
1.5.2 What you will need 21
1.5.3 Basic instructions 22
1.5.4 Connect the Kit to your workstation 22
1.5.5 Connect the Kit to Ethernet 22
1.5.6 Kit Network Setup for Ethernet Cross-over Cable 22
1.5.7 Kit Network Setup for Standard Ethernet Cable 25
1.5.8 Using the ‘ping’ command 26
1.5.9 Use the ‘ifmac’ command to display and change the MAC address 27
1.6 Lab 6: File Systems 28
1.6.1 Abstract 28
1.6.2 What you will need 28
1.6.3 Basic instructions 29
1.6.4 FATFS (Fat File System) 29
1.6.5 YAFFS (Yet Another Flash File System) 31
1.6.6 Working with YAFFS in LogicLoader 31
1.7 Lab 7: Using Scripts 37
1.7.1 Abstract 37
1.7.2 What you will need 37
1.7.3 Basic instructions 37
1.7.4 Connect the kit to your workstation 38
1.7.5 Basic scripting guidelines 38
1.7.6 Launching a Script from an SD Card 41
1.7.7 Persistent Script Storage 43
1.8 Lab 8: Loading an NK.BIN 47
1.8.1 Abstract 47

 Logic Product Development Company, All Rights Reserved

1.8.2 What you will need 47
1.8.3 Basic instructions 47
1.8.4 Connect the Kit to Your Workstation 48
1.8.5 Load a File from Secure Digital (SD) Card 48
1.8.6 Load a File from CompactFlash 50
1.8.7 Load a File from YAFFS 51
1.8.8 Execute a Windows Embedded CE Image 52

 Logic Product Development Company, All Rights Reserved

1 LogicLoader Labs

These labs provide an informal introduction to the LogicLoader bootloader software. The step-by-step directions act as a means to discover how LogicLoader works. All labs are applicable to both the Zoom OMAP35x Development Kit featuring the OMAP35x SOM-LV and the Zoom OMAP35x Torpedo Development Kit featuring the OMAP35x Torpedo SOM.

1.1 Lab 1: Kit Communications

1.1.1 Abstract

This lab will teach you how to establish a serial connection to your OMAP35x Development Kit. A serial connection is the most basic form of communication between a development workstation computer and the development kit.

1.1.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit
- A development workstation (PC or laptop)
- An available serial port or USB-to-serial adapter
- A terminal emulation program such as HyperTerminal or Tera Term. **Note:** Logic recommends using Tera Term, a free program downloadable from the Internet.

1.1.3 Basic instructions

If you are familiar with serial communications and a terminal emulator program, open your terminal emulator and set the port settings to:

- 115200 BAUD rate
- 8 data-bits
- No parity
- No hardware flow control
- No software flow control

If you are not familiar with serial communications, follow the instructions below for the program you intend to use.

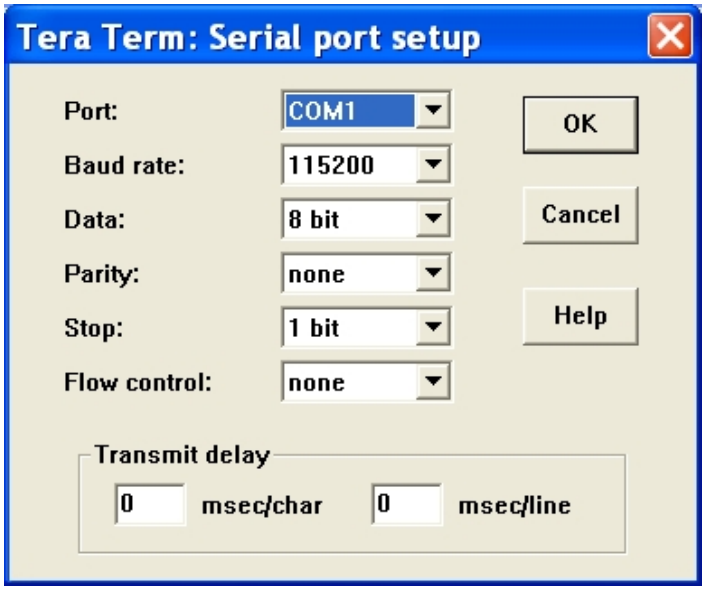
1.1.4 Tera Term instructions

Tera Term is a terminal emulation program available as a free download from the Internet. Logic recommends the use of Tera Term over HyperTerminal as it appears to be a more stable program. All Tera Term settings are controlled by a .ini file that you can modify as needed to make your time in Tera Term as efficient as possible. For example you can preset the port settings.

 Logic Product Development Company, All Rights Reserved

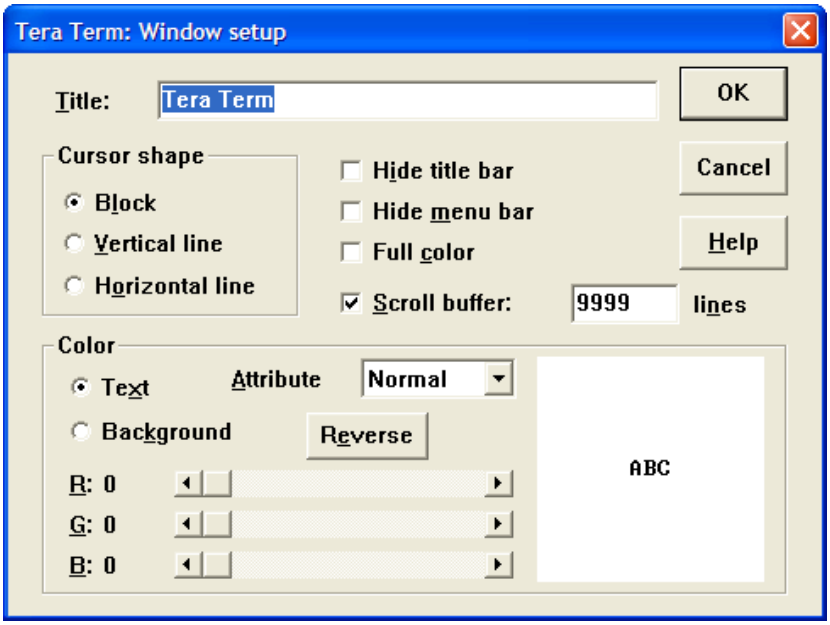
1. Start the Tera Term program

2. Select the ‘Setup -> Serial port…’ menu item
3. Select the appropriate COM port for your workstation
4. Change the Port Settings to:
   - Bits per second: 115200
   - Data bits: 8
   - Parity: None
   - Stop bits: 1
   - Flow control: None



 Logic Product Development Company, All Rights Reserved

5. Click ‘OK’
6. Select the Setup -> Window… menu item
7. Click the box for scroll buffer and make the amount as big as you like, up to 9999.



8. Click ‘OK’

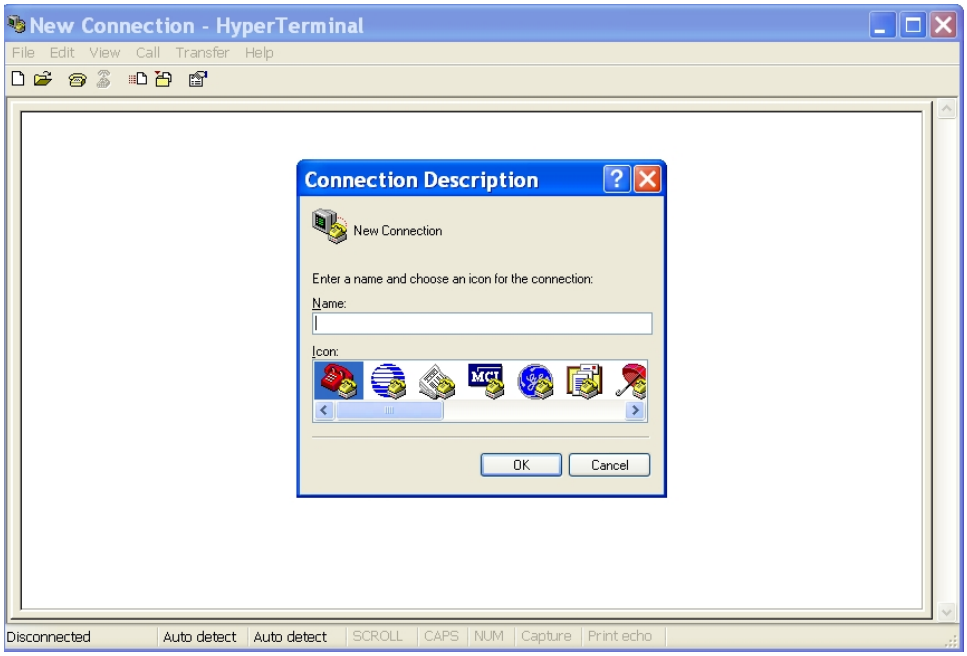
1.1.5 HyperTerminal instructions

HyperTerminal is a standard communications program available on most Microsoft workstations and laptops. You may use HyperTerminal; however, Logic recommends downloading and using the Tera Term program.

1. Start the HyperTerminal program using either of these methods:
   - Find the HyperTerminal icon using the Windows start menu. This is normally found under, ‘All Programs -> Accessories -> Communications -> HyperTerminal.’

 Logic Product Development Company, All Rights Reserved

2. Name your new connection ‘LPD’ and press ‘OK’



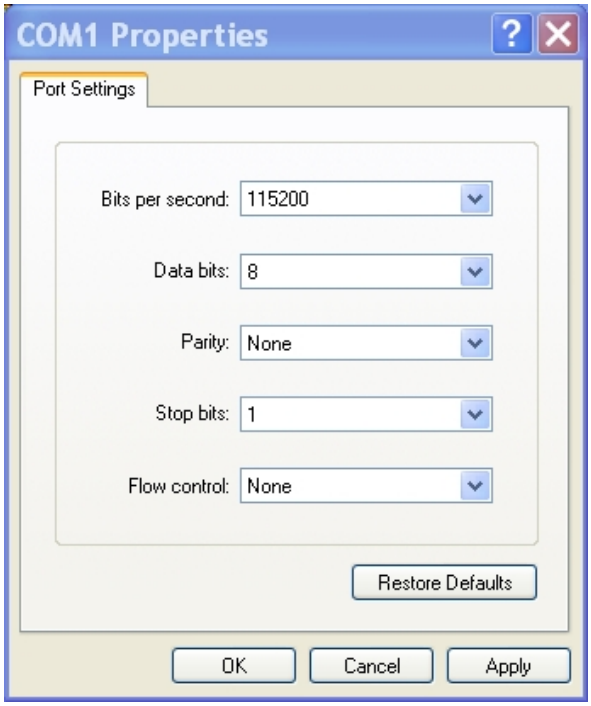
3. Using the drop-down menu, change the ‘Connect using:’ box to ‘COM1’ (or the appropriate COM port for your workstation)

4. Click ‘OK’

5. Change the Port Settings to:

   - Bits per second: 115200
   - Data bits: 8
   - Parity: None
   - Stop bits: 1
   - Flow control: None

 Logic Product Development Company, All Rights Reserved

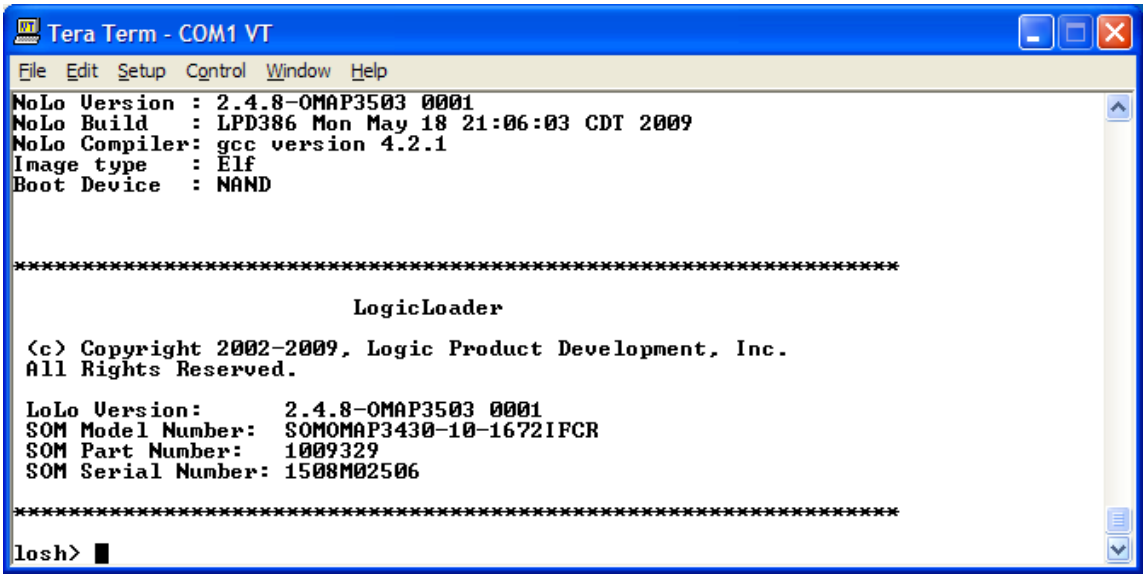


6. Click ‘OK’

1.1.6 Connect the Kit to your workstation

1. Use the gray serial cable marked “LOGIC” to connect the serial port on the kit to the appropriate port on your workstation.

2. Apply power to the kit and you should see the LogicLoader welcome prompt similar to the image below.



1.2 Lab 2: Learning the LogicLoader Help System

1.2.1 Abstract

This lab will teach you the basics of working with the LogicLoader (LoLo) built-in help system.

1.2.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit
- A development workstation (PC or laptop)
- A serial port communications connection between the workstation and the development kit

1.2.3 LogicLoader Interface

1. Start your terminal emulator program (Tera Term or HyperTerminal)
2. Ensure the null-modem serial cable provided with the kit is connected to both the kit and the development workstation
3. Power on the development kit.

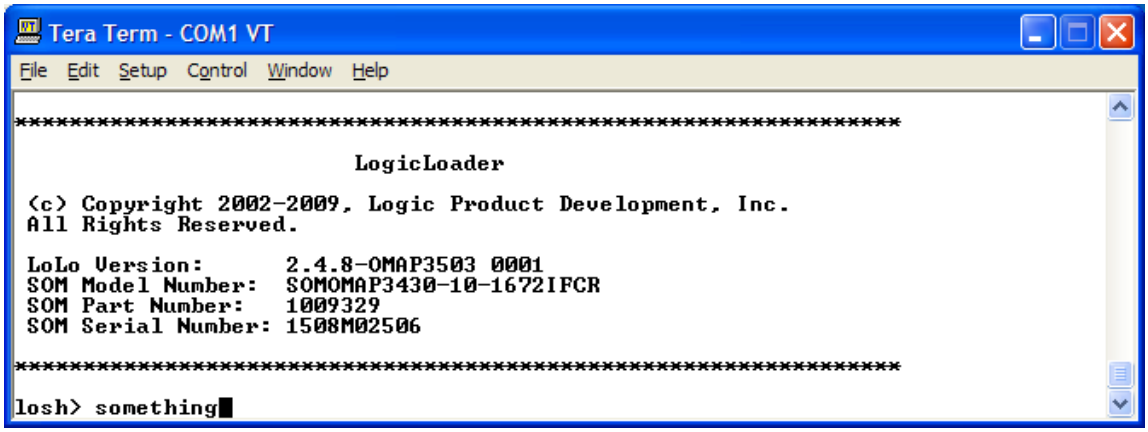
Once the kit powers up, you should see the LogicLoader welcome prompt similar to the image below.

Note the string `Version 2.4.8-OMAP3503 0001` in the image above. This tells us that this kit is running revision 2.4.8 of the LogicLoader program and has been built specifically for the OMAP35x platform. This version of LoLo also provides the SOM Model, Part and Serial Numbers. This allows you to send one screenshot to Technical Support and have all of the information in one place.

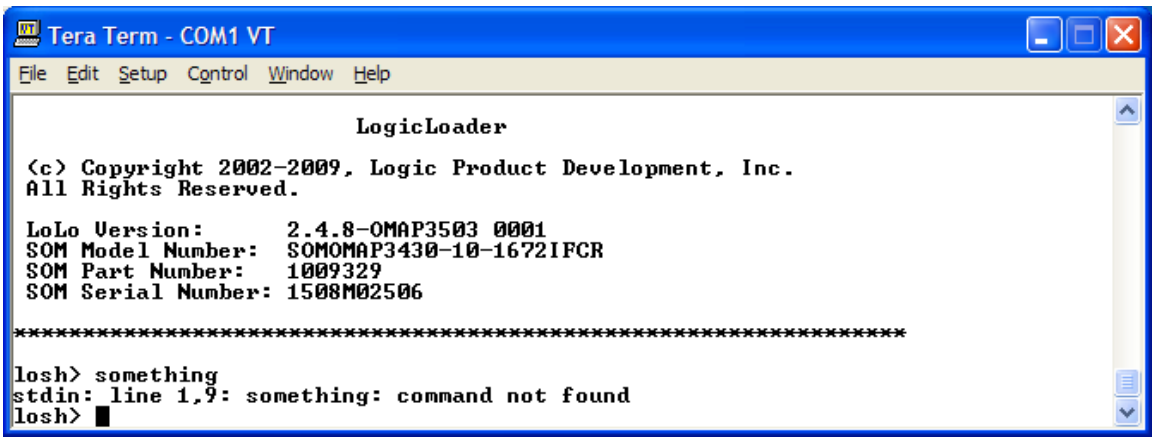
The string `losh>` is the LogicLoader command shell waiting for you to type something in.

1.2.4 Help for Incorrect Commands

Type the word **`something`** at the `losh>` prompt and press the “Enter” key.



You should receive output similar to the image below.



That was a disappointment. Of course, this is because there is no actual command named "something" included with the LogicLoader program. However, let's take a quick look at what was printed out. LoLo displayed the string

```
stdin: line 1,9: something: command not found
```

after we pressed the "Enter" key. There are actually four pieces of information contained in this string, each separated by colons (":").

The first piece of information is `stdin`. This is an abbreviation for the words "standard input." If you've ever spent any time working with a UNIX or Linux computer, you will probably understand what this refers to. If you haven't, don't worry, we will explain it for you. LogicLoader is a program that you get to interact with. The program needs a way to receive information from you so it can try to do what you want it to. The manner in which a program receives input from users has traditionally been called "standard input," or more briefly, "stdin." On a desktop computer, you are probably used to using a keyboard and a mouse to interact with programs. The operating system (e.g., Microsoft Windows) collects the characters you type and information about where you moved and clicked your mouse and passes them to whatever program you are using. In the case of LogicLoader, there isn't a mouse or a keyboard. LoLo collects the characters you send over your serial port via the terminal emulator program (Tera Term or HyperTerminal) and treats it as its "standard input." So the first bit of information in the string

```
stdin: line 1,9: something: command not found
```

 Logic Product Development Company, All Rights Reserved

is telling you that whatever error LoLo encountered occurred when it tried to receive something from the “standard input,” which is the development kit’s serial port.

As a quick aside, “standard input” has a partner in crime named “standard output” or, simply, `stdout`. On your workstation, “standard output” is usually the display or monitor attached to the computer. On a Zoom Development Kit, “standard output” is also the kit’s serial port. So when LogicLoader has anything to say to you, it talks to you on its “standard output.” We will come back to the concept of “standard output” or “stdout” a bit later because sometimes it is useful to tell LoLo to send its information someplace other than “stdout.”

The next piece of information in the string is, `line 1,9`. This is LoLo telling you that whatever error it encountered occurred on the first line of data received on “stdin” between the first and ninth characters. If we look back at what we typed, we’ll notice that the word “something” has nine letters, therefore we can assume that LoLo had a problem with that particular word.

The next piece of information is the actual word `something`. This is LoLo printing out the input that caused the problem.

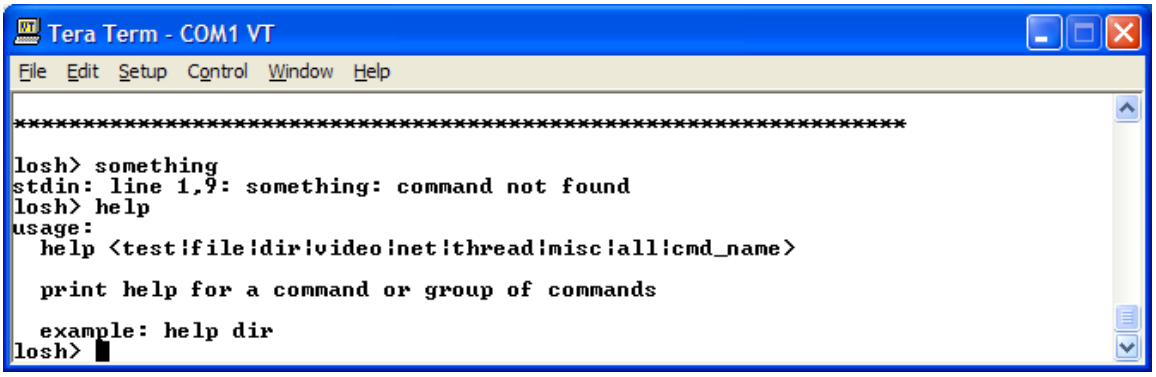
The final thing that the LogicLoader tells us is, `command not found`. This informs us that LoLo couldn’t find a command named “something” to execute.

In general, if you make a mistake, LoLo will try to tell you what you did wrong. Don’t worry about messing up or trying to type every command perfectly the first time. LoLo is a stable and forgiving program and you won’t hurt your board by playing around with it.

Now that we have established that there isn’t a command named “something,” what sorts of commands are included in the LogicLoader?

1.2.5 Using the ‘help’ command

1. Type the word **`help`** at the `losh>` prompt and press the “Enter” key.



Was that helpful? Maybe, maybe not, depending on how you expect a help system to work. Let’s take a look at LoLo’s help system. The first thing you should know is that LoLo interprets the word “help” as a command. The help command actually needs to be typed in conjunction with another word (its argument) to be a valid LoLo command. So by only typing “help,” we typed an invalid LoLo command. We haven’t done much right yet, have we?

It’s okay that we made a mistake though, because LoLo recognized it and is trying to teach us how to use the “help” command. Let’s look at what LoLo printed:

```
usage:
      help <test|file|dir|video|net|thread|misc|all|cmd_name>
      print help for a command or group of commands
      example: help dir
```

 Logic Product Development Company, All Rights Reserved

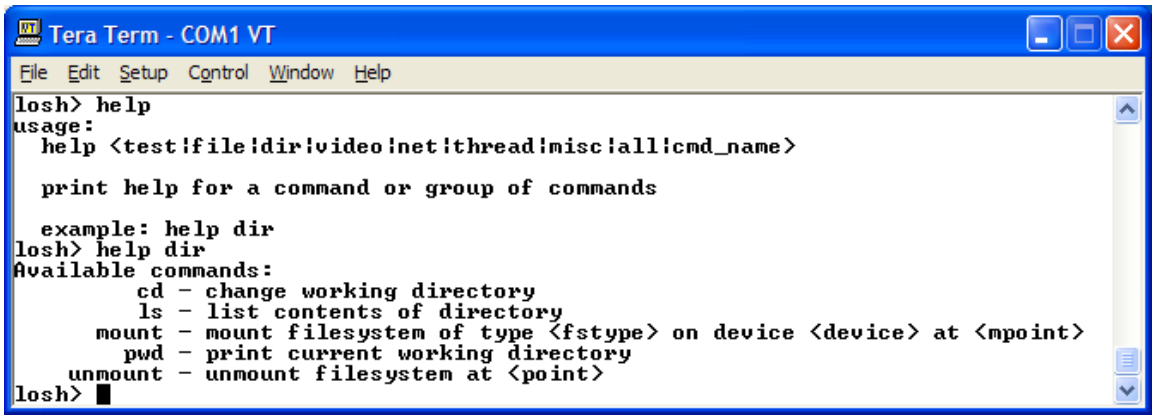
The output above is known as a command's help topic. If you type a valid LoLo command, but don't use it in a valid manner, LoLo will print out the command's help topic to remind you of the command's proper format and use.

The string:

```
usage:
      help <test|file|dir|video|net|thread|misc|all|cmd_name>
```

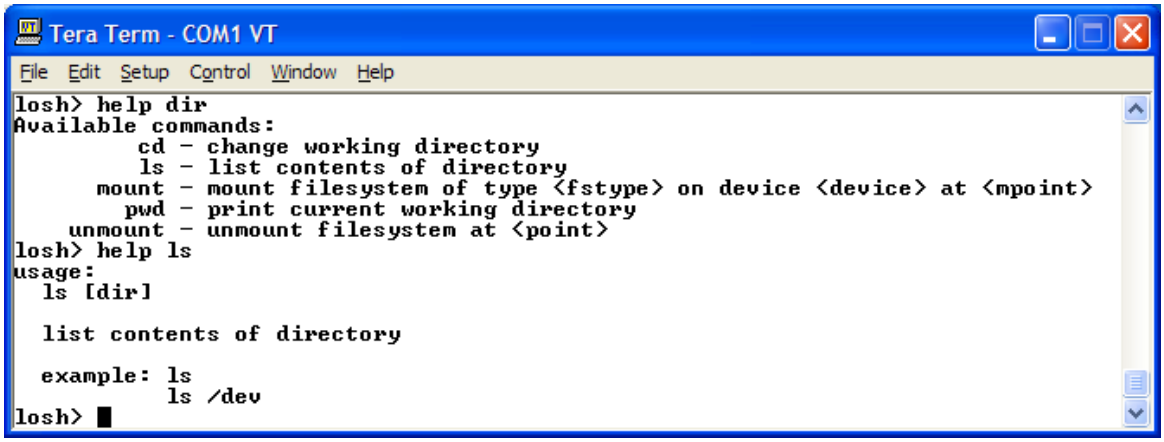
is the command's usage string. It tells us how the command should be typed. Some commands need to be supplied with arguments to work correctly. If this is the case, the necessary arguments will be printed between angle bracket symbols "< >." If more than one option exists for an argument, the possible arguments will be listed and separated by the vertical bar symbol "|." The above usage string is telling us that the help command must be given an argument and that argument can be the word "test," or the word "file," or the word "dir," etc. The rest of the output simply tries to explain what a command does and provide an example of how to use the command. Let's follow LogicLoader's example and see what happens.

2. Type the words **`help dir`** at the `losh>` prompt and press the "Enter" key



Now the help command is working for us. It printed a list of LogicLoader commands relating to directories along with a short summary of what each command does.

3. Type **`help ls`** at the `losh>` prompt and press the "Enter" key



 Logic Product Development Company, All Rights Reserved

By using “ls” as the help command’s argument, we’ve asked the help command to print the help-topic for a specific LogicLoader command.

We’ve just learned three important things about LoLo’s “help” command. First, if you have completely forgotten how to use LoLo, just type “help” at the “losh>” prompt. Typing `help` will print the help-topic for the actual help command.

Second, the help command can print a summary group of commands relating to a specific topic. If you type “help dir” you will get a brief listing of all LoLo commands relating to directories. If you type “help file” you will get a listing of LoLo commands used when working with files.

Third, the help command will display the help-topic for a single command. If you know a command’s name but forgot how to use it, just type “help” followed by the name of the command and you will see the help-topic for that command.

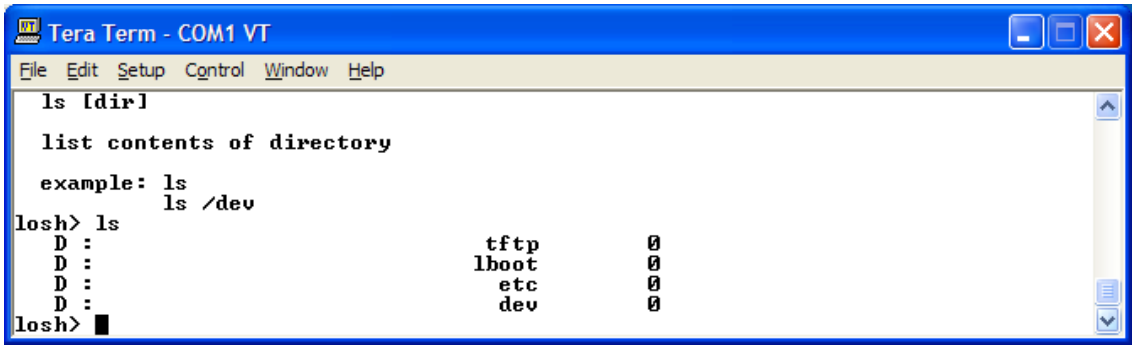
Before we continue, let’s take a closer look at the “ls” command’s help-topic. It looks a little different than what we saw before. Notice the command’s usage string, reprinted below.

```
usage:
      ls [dir]
```

There aren’t any angle bracket symbols (“< >”) on that line, but there is a pair of square brackets (“[ ]”) with the word “dir” in between them. Arguments that *must* be supplied with a given command are listed in angle brackets. Arguments that *may optionally* be used with a command are listed in square brackets. So the “ls” command’s usage string tells you that you can either just type the command alone or optionally type the command with the name of the directory whose contents you want to list.

Try both of the examples as listed by the “ls” command’s help topic.

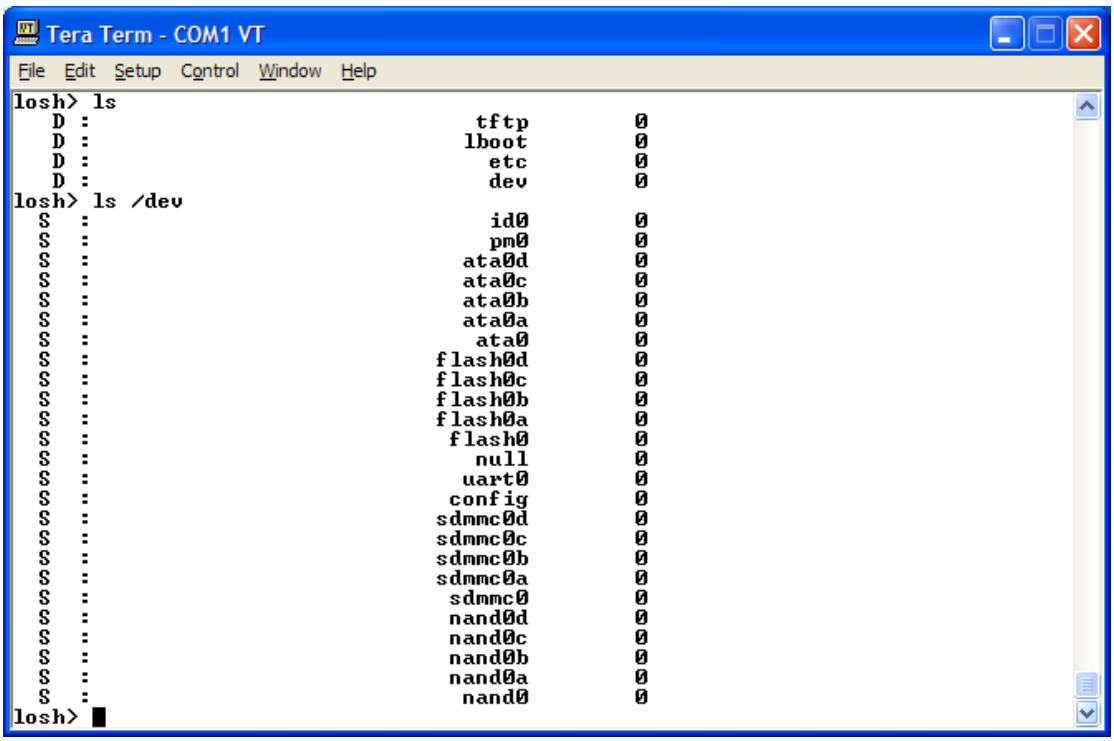
4. Type the letters **`ls`** at the `losh>` prompt and press the “Enter” key



The first invocation of the “ls” command simply listed the contents of the current directory.

 Logic Product Development Company, All Rights Reserved

5. Type **`ls /dev`** at the `losh>` prompt and press the “Enter” key



The second invocation of the “ls” command listed the contents of the “/dev/” directory because we supplied the command with an optional argument. Before we leave the topic of LoLo’s help facility completely, let’s take a look at all of the available commands.

 Logic Product Development Company, All Rights Reserved

6. Type the words **`help all`** at the `losh>` prompt and press the “Enter” key

Supplying the “help” command with the “all” argument lists the name of each command built into the version of LoLo running on your development kit along with a brief description of what the command does. Logic provides a *LogicLoader User's Manual* and a *LogicLoader Command Description Manual* on the downloads page for your specific development kit. You should download these documents now.

 Logic Product Development Company, All Rights Reserved

1.3 Lab 3: Using the LogicLoader Command Line Editor

1.3.1 Abstract

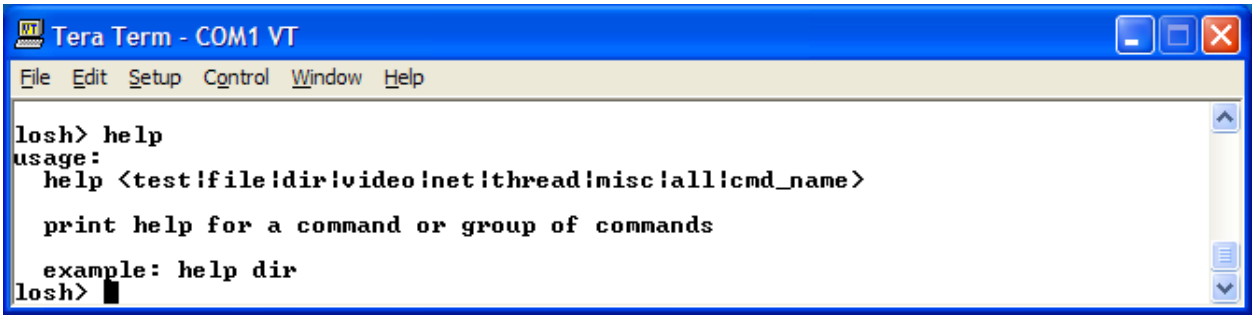
This lab will teach you how to work more efficiently with LogicLoader (LoLo) by learning its command line editing features.

1.3.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit
- A development workstation (PC or laptop)
- A serial port communications connection between the workstation and the development kit

1.3.3 Instructions

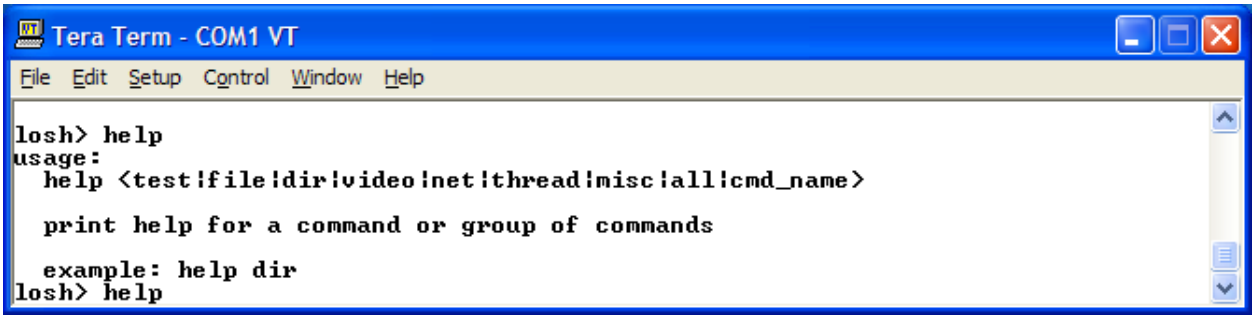
1. At the `losh>` prompt, type the **`help`** command



As discussed in the “LogicLoader Help System” lab, the help command requires an argument. Therefore, just typing “help” prints out the commands usage string to remind us how the command should look.

Now that we’ve been reminded that the help command needs an argument, let’s pretend that we wanted to type “help video” because we wanted to see all of the commands that relate to a display. We *could* just type “help video” at the prompt and press the Enter key. But, there is an easier way.

2. At the `losh>` prompt, press the up-arrow key on your keyboard

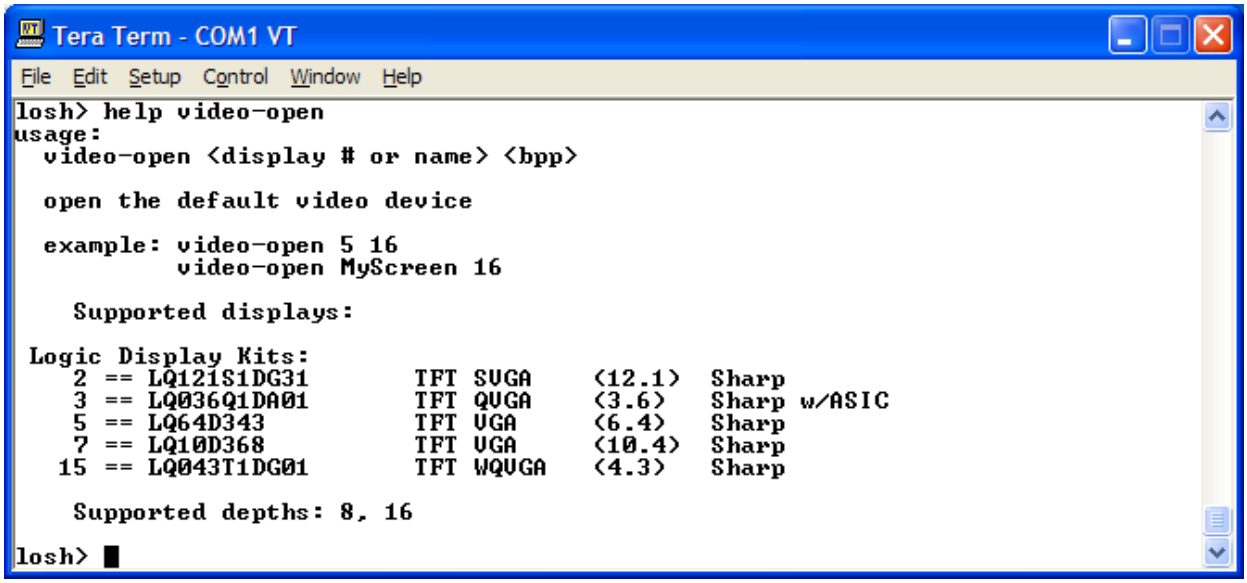


Notice how the help command reappears? The up-arrow cycles through the last several commands that you have typed—fifteen to be exact. Type a few more commands and then use the up-arrow and Enter key to find and repeat them.

 Logic Product Development Company, All Rights Reserved

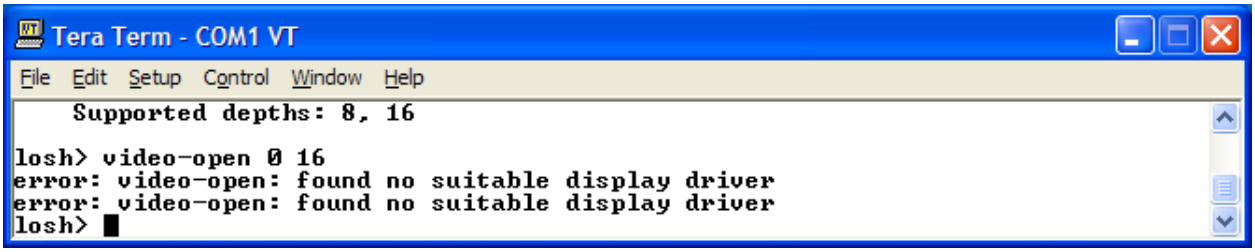
The down-arrow works in the opposite way of the up-arrow. Using the arrows can save you quite a bit of typing if you need to repeat a command often. But what if you typed the command wrong to begin with?

3. At the `losh>` prompt type the command **`help video-open`** followed by the "Enter" key



The "video-open" command is used to initialize LogicLoader's display. We won't go into much detail concerning the specifics of that command during this lab. However, by reading the command's help-topic we can see that it takes two mandatory arguments: a display number and depths. By reading the output further, we can see that the version used during the writing of this lab accepted display numbers 2, 3, 5, 7, and 15. We can also see that a display depth of 8 and 16 bits per pixel is supported.

4. At the `losh>` prompt type the command **`video-open 0 16`** followed by the "Enter" key



Notice that we typed the command wrong and LoLo prompted us. The error was a result of us trying to initialize screen number zero, which our help-topic told us was not supported.

Instead of retyping the entire command line, let's just fix the command we sent to LogicLoader.

1. Press the up-arrow to bring back the command previously entered;
2. Use the left-arrow to bring the curser to the right of the "0" character;
3. Use the "Backspace" key to delete the errant zero;
4. Type the number of a supported display (such as 15) in place of the zero;
5. Press the "Enter" key.

 Logic Product Development Company, All Rights Reserved

Notice how the command is correct now? Editing the command line instead of retyping the entire line cut our key presses in half.

For those of you familiar with traditional terminal emulators or using a program that doesn't properly send Backspace and arrow key presses, there is another series of key combinations that will move you around a command line. These are listed below:

- Control-a - move to beginning of the line
- Control-e - move to the end of the line
- Control-b - move backward (to the left) one character
- Control-f - move forward (to the right) one character
- Control-d - delete (remove) the character under the cursor

Go ahead and take some time to familiarize yourself with the above key combinations and command line editing.

1.4 Lab 4: Updating LogicLoader

1.4.1 Abstract

This lab will teach you how to download a new version of the LogicLoader (LoLo) to your Zoom OMAP35x Development Kit using the 'update' command.

1.4.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit
- A development workstation (PC or laptop)
- An available serial port or USB-to-serial adapter
- A terminal emulation program such as HyperTerminal or Tera Term. **Note:** Logic recommends using Tera Term, a free program downloadable from the Internet.

1.4.3 Basic instructions

Using the same process you learned in Section 1.1 "Lab 1: Kit Communications" open your terminal emulator and set the port settings to:

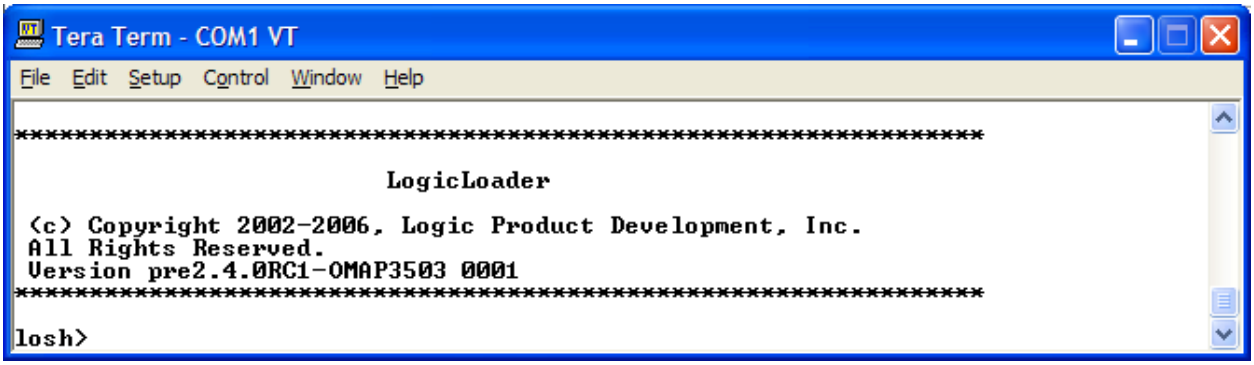
- 115200 BAUD rate
- 8 data-bits
- No parity
- No hardware flow control
- No software flow control

1.4.4 Connect the Kit to your workstation

Use the gray serial cable marked "LOGIC" to connect the serial port on the kit to the appropriate port on your workstation

Apply power to the kit and you should see the LogicLoader welcome prompt similar to the image below

 Logic Product Development Company, All Rights Reserved



Take note of what version number appears in this welcome prompt, as that piece of information will be required later on.

1.4.5 Obtain a new version of LogicLoader

The first step is to access Logic's download site at http://support.logicpd.com/auth/ and enter your username and password. This will bring you to your Account Summary page. Find the "OMAP35x Zoom Development Kit" name and click on the "All Downloads" link. On this page, scroll down to locate the most recent version of LogicLoader (under the "LogicLoader Bootloader/Monitor" heading) and download it to your development computer.

Now that you have downloaded the latest version of LogicLoader to your computer, you can update LoLo on your development kit using a bootable SD card. We have found that the most reliable method for updating LoLo is to boot from an SD card containing the same (new) LoLo version you ultimately want to have on the OMAP35x SOM.

1.4.6 Download LogicLoader using a Bootable SD card

1.4.6.1 Prepare the SD Card

1. The first step is to prepare the SD card to boot. The HP USB Formatting Tool version 2.0.6 appears to be the best tool for formatting an SD card so it can be used to boot the OMAP35x Development Kit into LogicLoader. Using the HP USB Formatting Tool ensures that an extended partition is not created on the SD card (a requirement of the OMAP processor). Using the Windows formatting feature may not be able to achieve this requirement.

   The most recent version of the tool available on HP's site doesn't seem to work as well as version 2.0.6. As such, the tool needs to be downloaded from any number of third-party websites. To find these websites, perform a Google search for: **HP formatting tool 2.0.6**

   NOTE: As with downloading anything from the Internet, use caution in what sites you visit and what you download. Logic has no association with any of these websites and makes no guarantees to their validity or the usability of downloads from their sites.

 Logic Product Development Company, All Rights Reserved

2. Insert an SD card into your USB card reader and open the formatting tool. (Remember that formatting an SD card will erase its entire contents; back-up the SD card's contents before continuing or select a card that doesn't contain vital information.)

3. The file system on your SD card must be FAT32. Make sure FAT32 appears in the File System drop-down menu and click the "Start" button.

4. You will receive a warning that the contents of the card will be erased. Assuming you have backed-up the contents of the SD card, click "Yes".

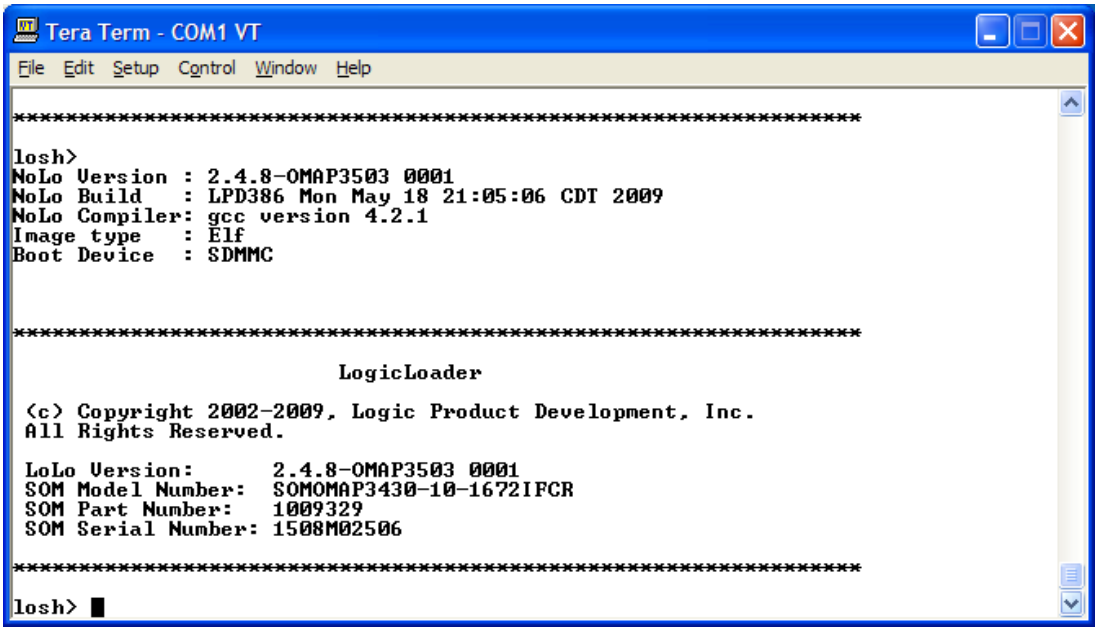
5. When formatting completes, you will receive a confirmation screen with the new file system type and the space allocation on the card. You are now ready to add the LogicLoader files to the SD card.

1.4.6.2 Use the SD Card to Update LoLo

1. Locate the LogicLoader files that you downloaded to your computer. Copy the "MLO" file from that folder to the SD card. This file must be the first added to the SD card for it to boot automatically.

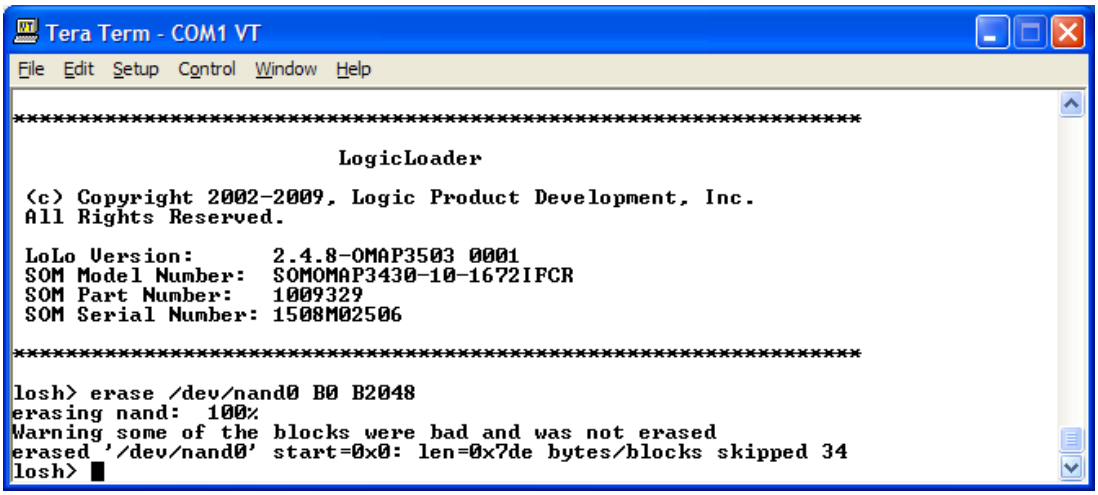
2. Next, add the "lboot.elf" file to the SD card.

3. Remove the SD card from your USB card reader and insert it into the SD card slot on your OMAP35x Development Kit. Press the "Power on/off" button and you should see a screen similar to this:



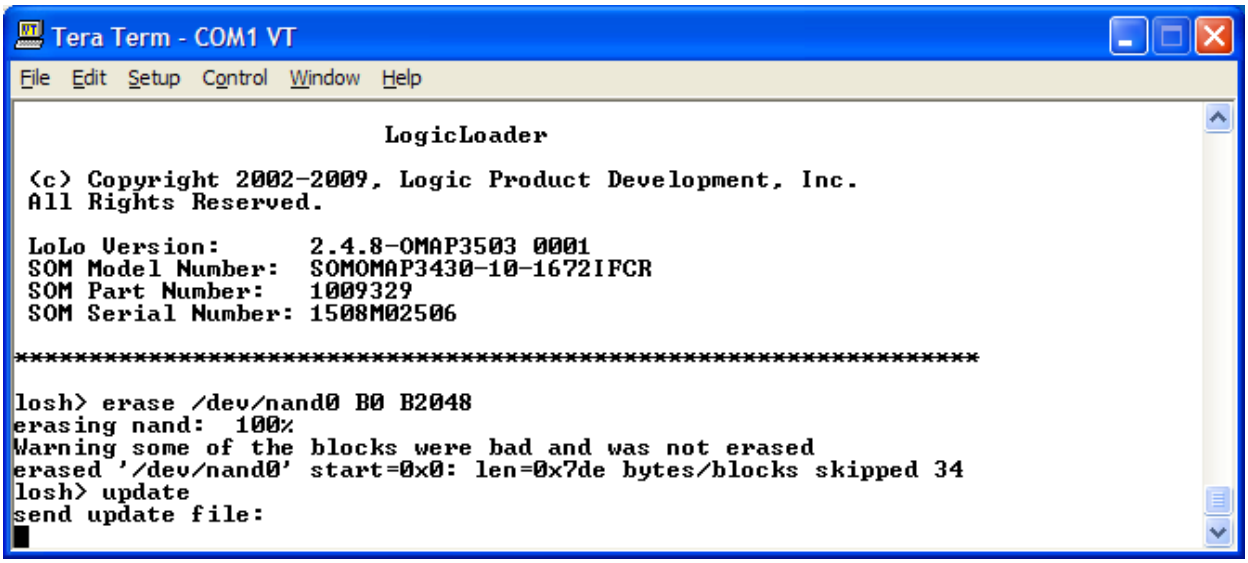
Note that the first section of the banner says the kit booted from an `Image type : Elf` and `Boot Device : SDMMC`.

4. Because NoLo and LoLo reside in NAND, it is necessary to erase the old LogicLoader before updating to a new one. Type **`erase /dev/nand0 B0 B2048`** and press the “Enter” key.

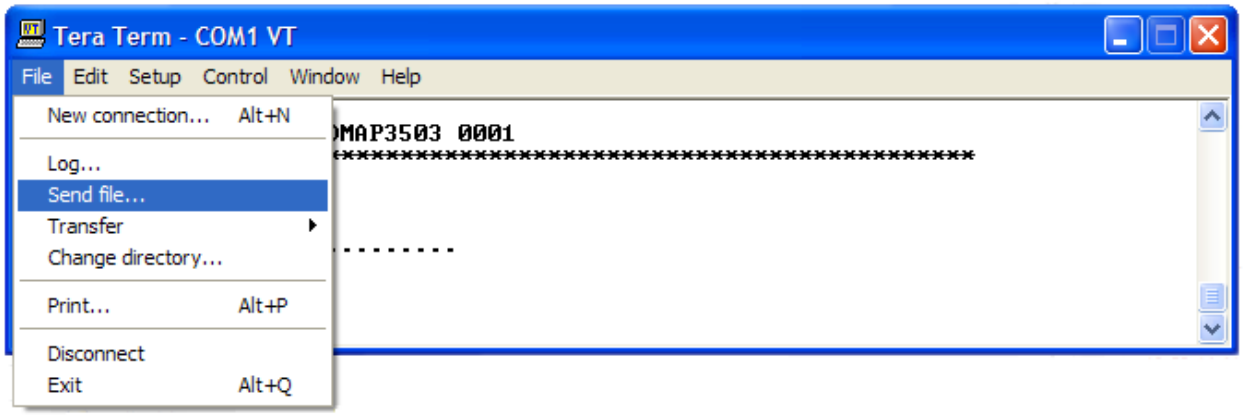


5. Because you booted the kit from SD, you will not be prompted that you are erasing locked areas in NAND. You now have a blank flash device to update to.

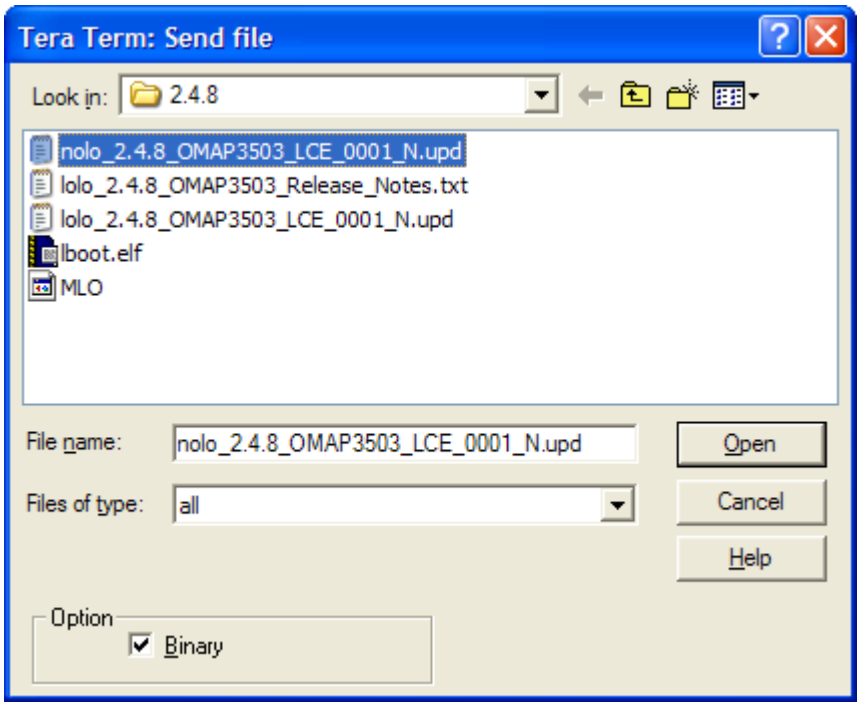
6. Type **`update`** and press the “Enter” key. Your screen should look similar to the figure below.



7. LogicLoader is waiting for you to send the new version over the serial port. Click ‘File’ and select ‘Send file...’ to send the new LogicLoader version to your kit.

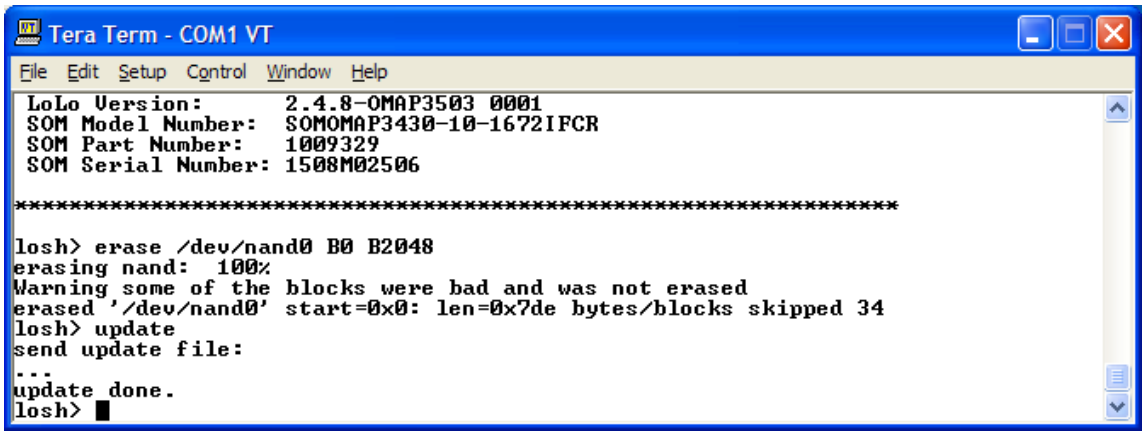


8. Locate and select the “nolo_x.x.x_OMAP....upd” file you downloaded to your computer. Verify that the ‘Binary’ option is selected. Then click ‘Open’.



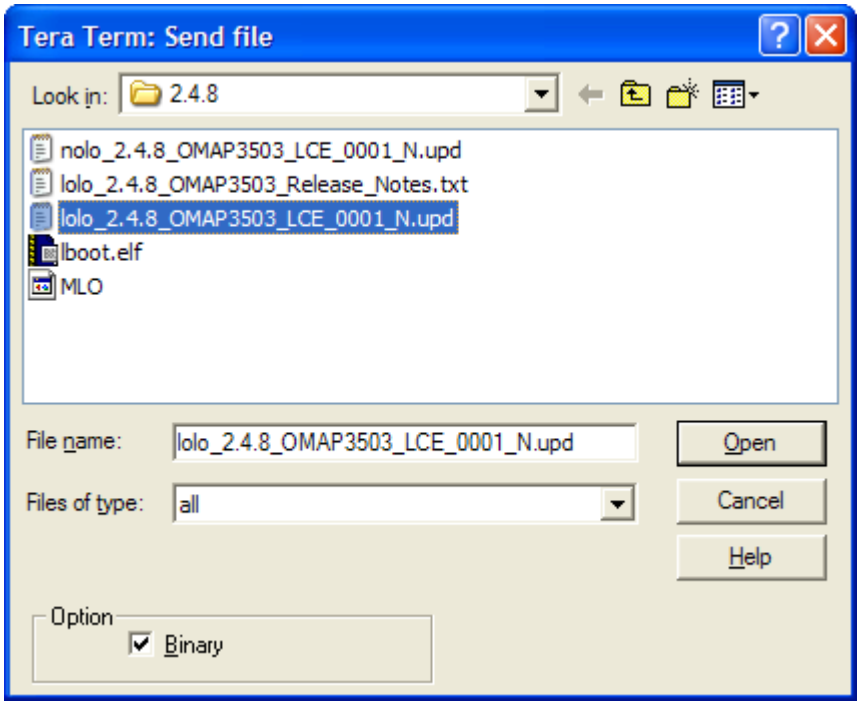
 Logic Product Development Company, All Rights Reserved

9. This will burn the new NoLo into flash. Verify that the download was successful by waiting for an `update done` message to appear.



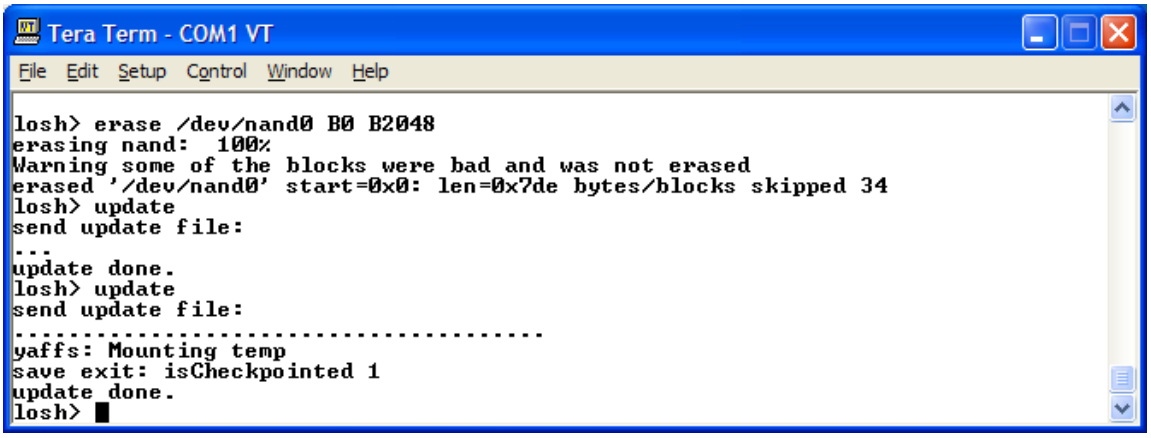
10. This version of LogicLoader has both a LoLo and a NoLo. Logic will update both files together, so you should always update both at the same time.

11. Type **`update`** again (or press the up arrow) and follow the same procedure to send a file, only this time pick the “lolo_x.x.x_OMAP....upd” file. Once again, verify the “Binary” option is checked.

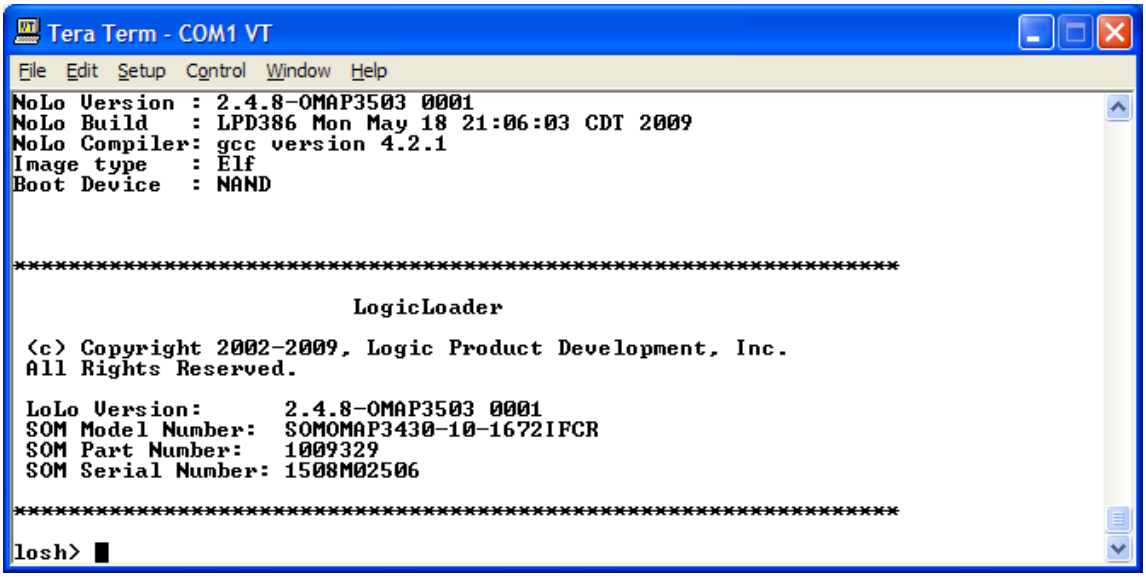


 Logic Product Development Company, All Rights Reserved

12. The update is complete when the `update done` message appears once again.



13. At this point, you can remove the SD card, reset your development kit, and verify the new LogicLoader version number.



14. If the new version number is displayed, you now have the latest and greatest version of LogicLoader installed on your kit.

    Note that the first section of the banner says the kit booted from an `Image type : Elf` and `Boot Device : NAND`.

1.5 Lab 5: Network Connection

1.5.1 Abstract

This lab will teach you how to connect your workstation to the Zoom Development Kit via Ethernet.

1.5.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit

- A development workstation (PC or laptop)
- An available serial port or USB-to-serial adapter
- A terminal emulation program such as HyperTerminal or TeraTerm. **Note:** Logic recommends using TeraTerm, a free program downloadable from the Internet.

1.5.3 Basic instructions

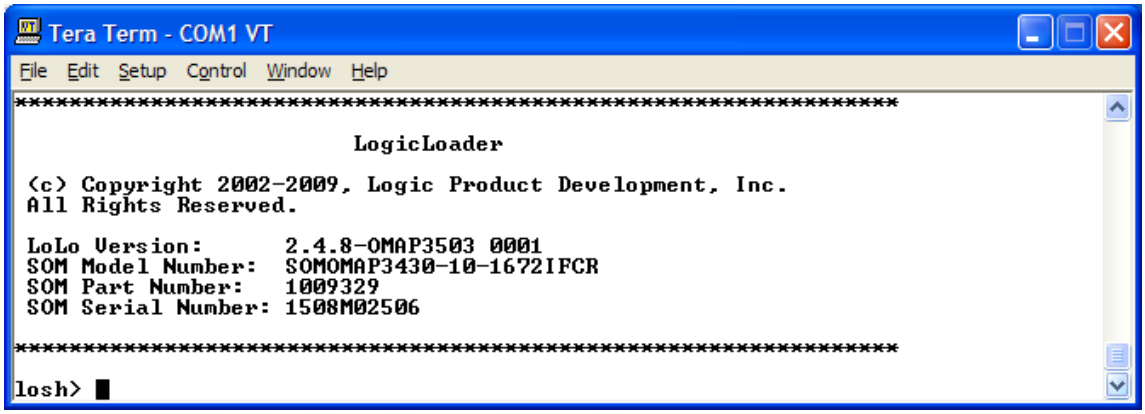
Using the same process you learned in Section 1.1 “Lab 1: Kit Communications” open your terminal emulator and set the port settings to:

- 115200 BAUD rate
- 8 data-bits
- No parity
- No hardware flow control
- No software flow control

1.5.4 Connect the Kit to your workstation

Use the gray serial cable marked “LOGIC” to connect the serial port on the kit to the appropriate port on your workstation

Apply power to the kit and you should see the LogicLoader welcome prompt similar to the image below



1.5.5 Connect the Kit to Ethernet

Use the yellow Ethernet cross-over cable to connect the Ethernet port on the kit to your workstation or a standard cable to connect to a switch

Adding Ethernet connectivity to your kit will make several procedures easier and is essential for connecting to Platform Builder. There are multiple ways to connect your kit to a workstation, we will discuss both a direct connection using a cross-over Ethernet cable and connecting through a network switch.

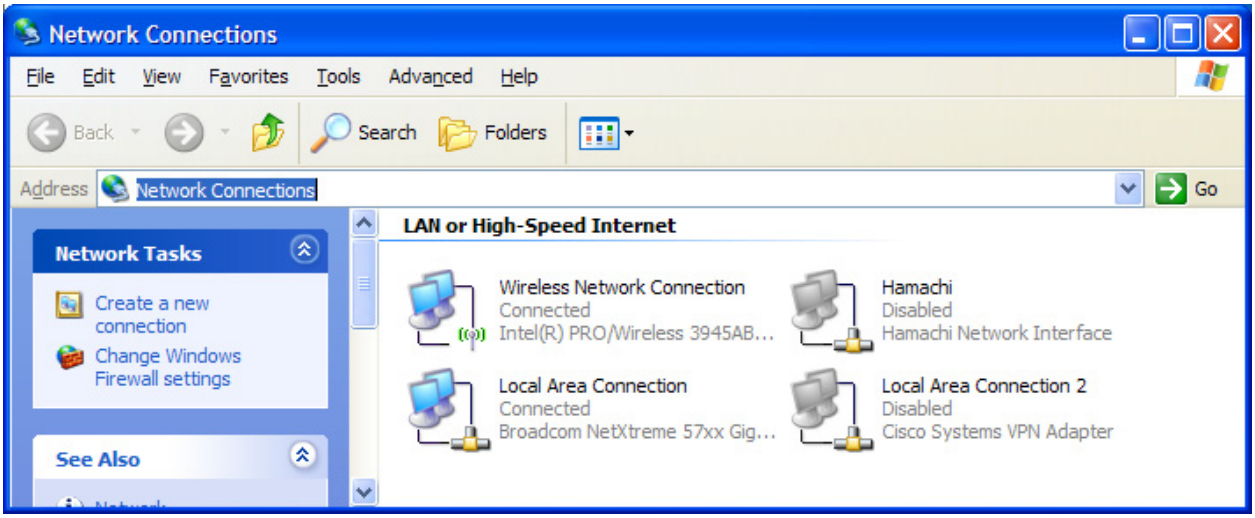
1.5.6 Kit Network Setup for Ethernet Cross-over Cable

If you are using the Ethernet cross-over cable, you will need to setup the kit address using the full ‘ifconfig’ command. If you don’t know the IP address of your workstation, go to Start -> Run and

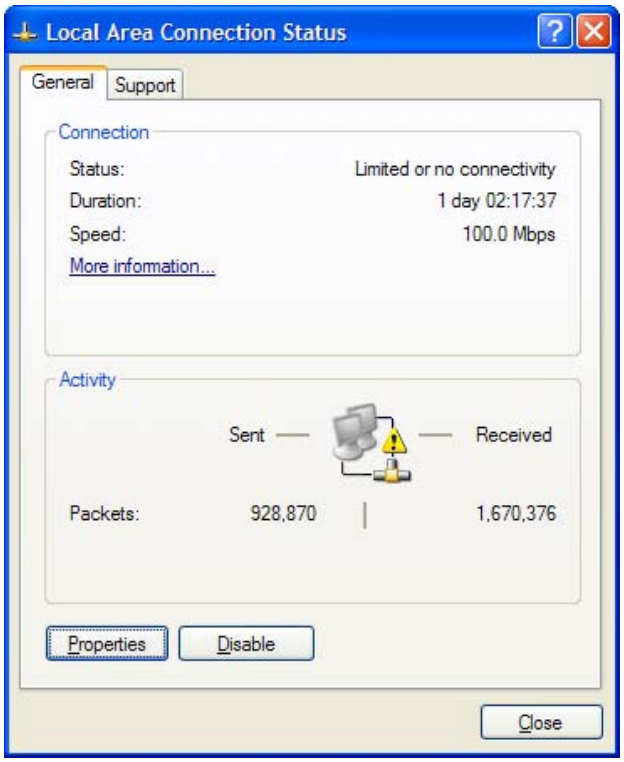
type **cmd**. Type **ipconfig /all** and locate the IP address for the Wired LAN connection that you are using.

The IP address of the kit must match the IP address of your workstation for as many 255 number sets as there are in the subnet mask. For example, if your subnet mask is 255.255.0.0 then the first two number sets of the IP address must be the same on both devices.

1. If you would like to set your IP address you can do so by going to:
   Start -> Network Connections -> Local Area Connection.

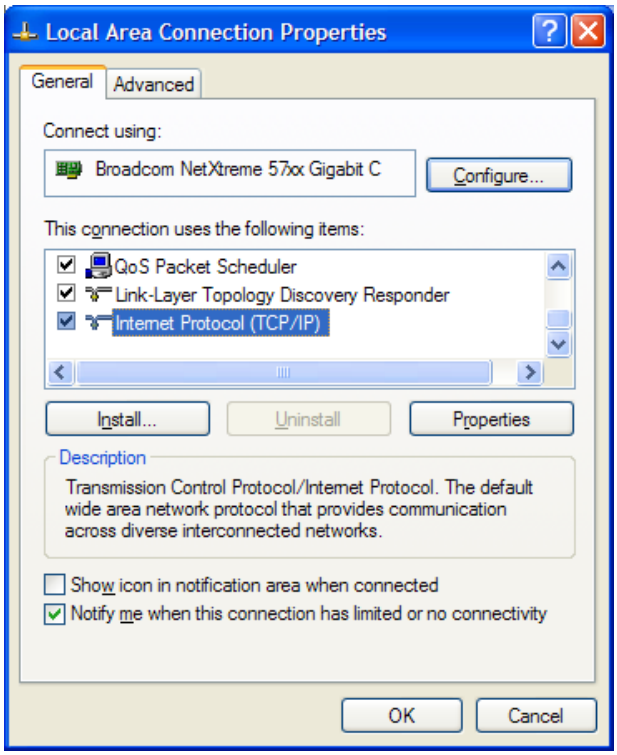


2. Double-click on the icon for your wired connection.



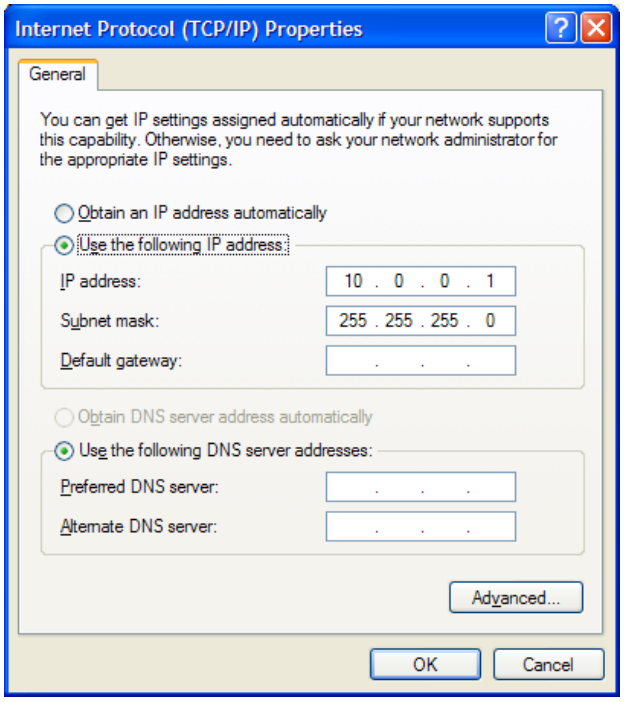
 Logic Product Development Company, All Rights Reserved

3. Click the “Properties” button. Scroll down until you see “Internet Protocol (TCP/IP)”. Click on the name to select it.



4. Click the “Properties” button.

5. Set your IP address and Subnet mask as desired. Then click the “OK” button.

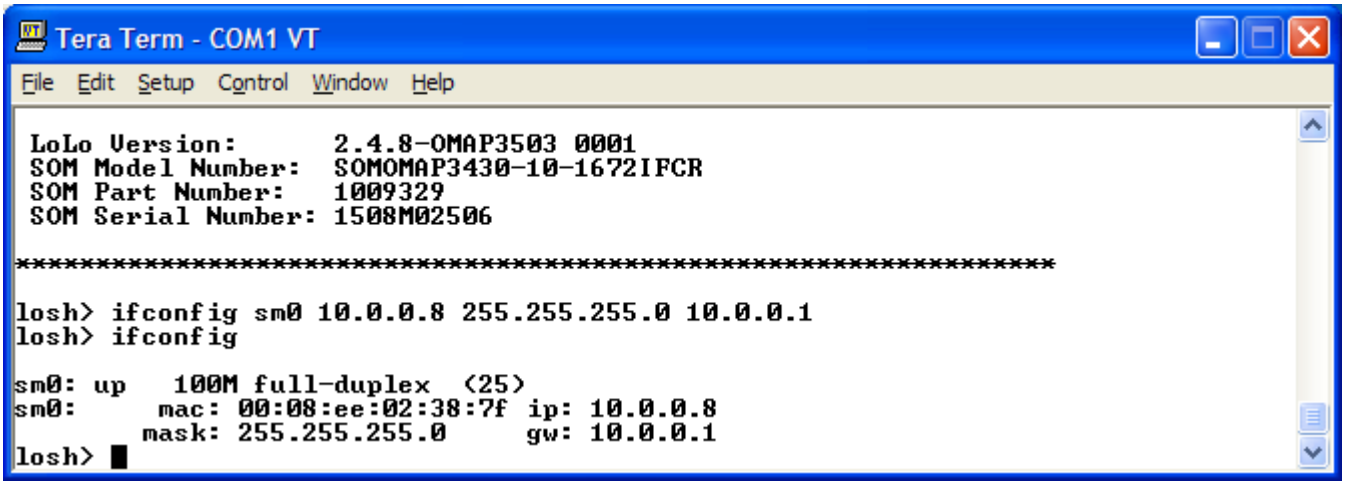


 Logic Product Development Company, All Rights Reserved

6. Once you have the IP address and subnet mask of your workstation, return to your Tera Term window. Once there, you can use the following command line to set the networking parameters on the kit:

`ifconfig sm0 <IP address of kit> <subnet mask> <IP address of workstation>`

The figure below shows an example of the full 'ifconfig' command. Remember that your numbers will be different.

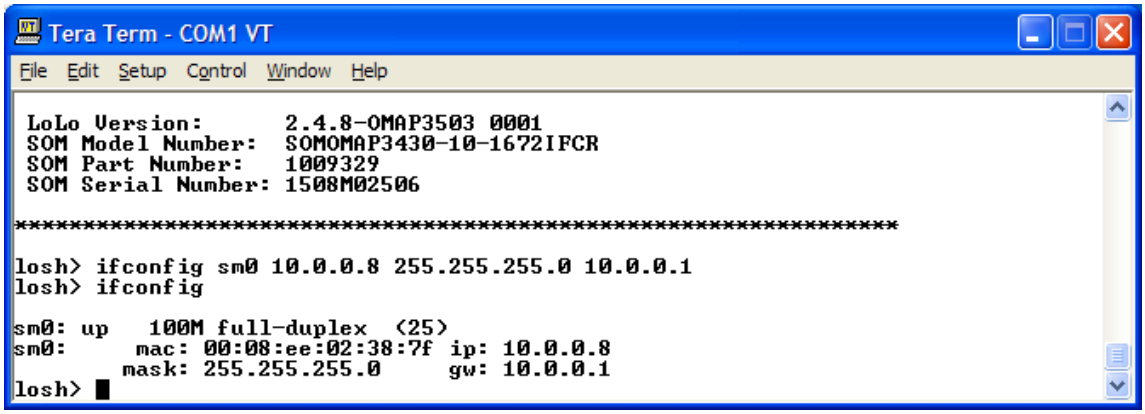


You now have setup your networking parameters and can skip to Section 1.5.8.

1.5.7 Kit Network Setup for Standard Ethernet Cable

If you have a network connection with DHCP you can use it to quickly set your network parameters.

1. From the `losh>` prompt, type **`ifconfig`**.

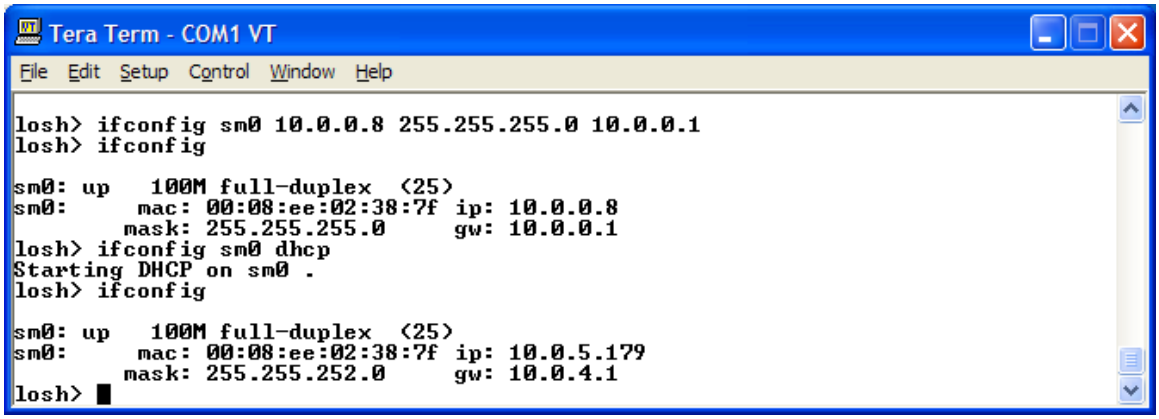


You can see the interface information for sm0. The MAC address is 00:08:ee:02:38:7f. The IP address, mask and gateway have been set.

2. Type **`ifconfig sm0 dhcp`**

LoLo will start DHCP on sm0 and will set the IP address, mask and gateway parameters. To display the new values, type **`ifconfig`** again, as shown below.

 Logic Product Development Company, All Rights Reserved



1.5.8 Using the 'ping' command

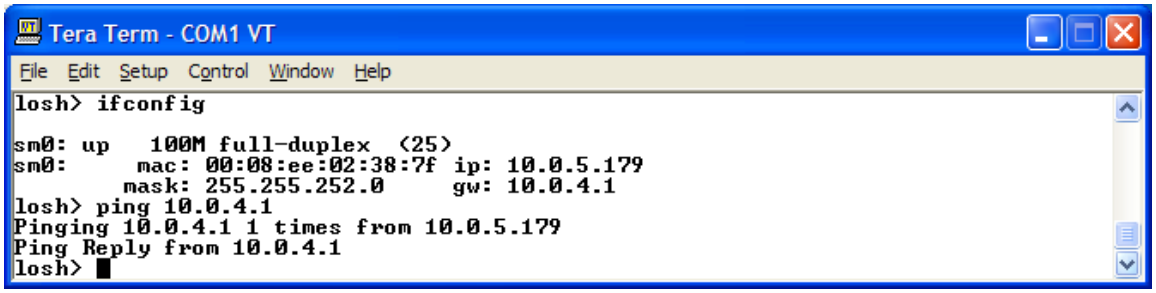
The 'ping' command can be used to check your connection to a specific workstation. You will need to have your kit connected via Ethernet and know the IP address of the target workstation.

1.5.8.1 Use the 'ping' command to check your connection

In the previous sections you used 'ifconfig' to initialize your Ethernet connection. Now you are going to use the 'ping' command to check the connection to a specific workstation. If you don't know the IP address of the target, go to Start -> Run and type **`cmd`**. Type **`ipconfig /all`** and locate the IP address for the connection you are using.

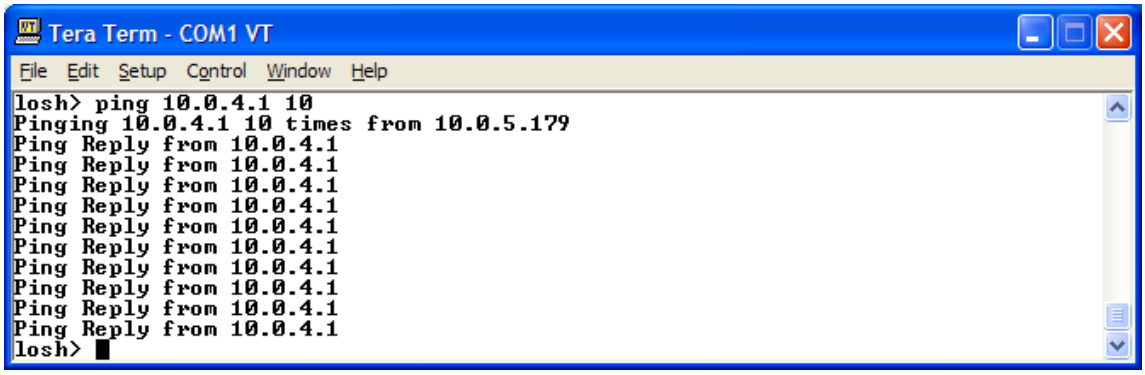
The 'ping' command can be used with just the IP address of the target workstation or with a number of different pings specified.

At the losh> prompt type **`ping XXX.XXX.XXX.XXX`** (fill in the X's with the IP address for your workstation).



This will ping your workstation one time. If you type **`ping XXX.XXX.XXX.XXX 10`** you will ping the target 10 times. Remember you can just hit the up arrow to repeat the last command and then add the 10 at the end. An example is shown below (remember your IP address will be different than that in the example below).

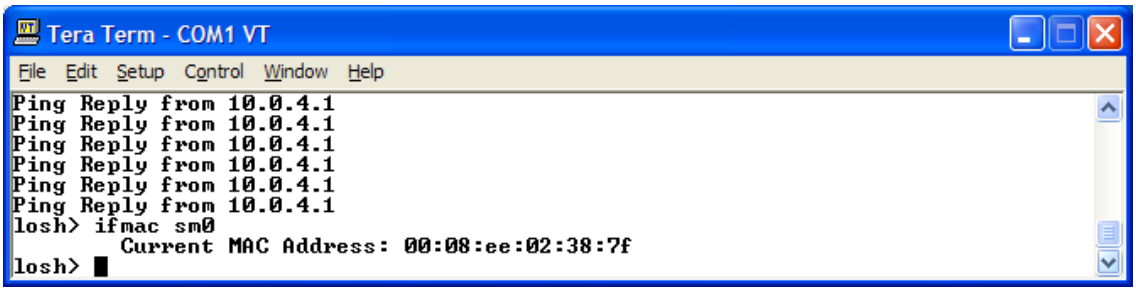
 Logic Product Development Company, All Rights Reserved



Congratulations, you have confirmed the connection. This process can be used any time you need to set up a network connection between a kit and a workstation.

1.5.9 Use the 'ifmac' command to display and change the MAC address

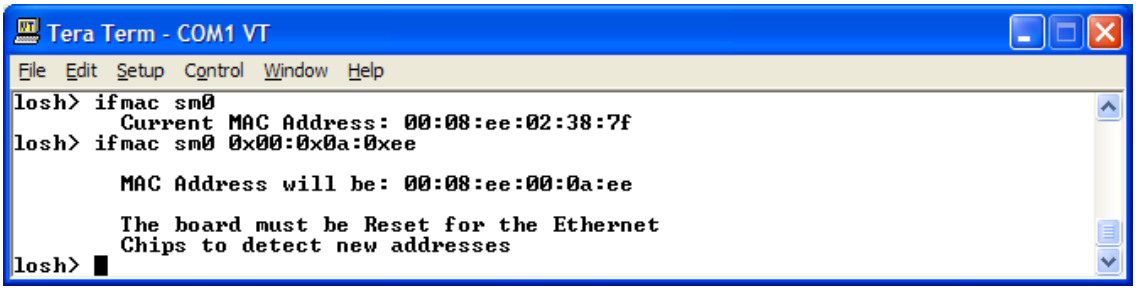
The 'ifmac' command can be used with just the interface name to display the current MAC address of your kit by typing **`ifmac sm0`**. It will take LoLo a few seconds to get the MAC address the first time you use the 'ifmac' command. After the current MAC Address is displayed, the 'ifmac' command will display immediately.



If you need to set your MAC address to something different than the current address, you can do so by adding the new address to the command. To change the address you can send bytes 4, 5, and 6. The first three bytes will default to Logic's 0x00:0x08:0xEE.

Type **`ifmac sm0 0x00:0x0a:0xee`**

LoLo will display the following message to confirm the change has taken place.

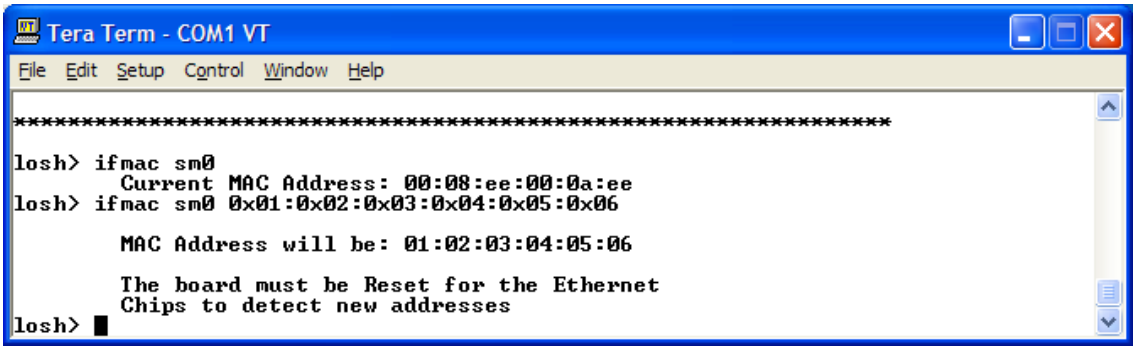


Reset your kit and type **`ifmac sm0`** to verify the new MAC address.

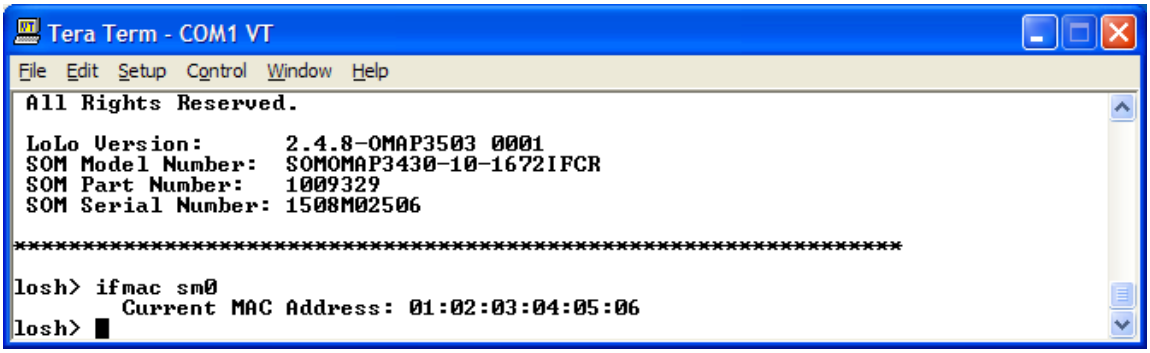
 Logic Product Development Company, All Rights Reserved

If you prefer to set the whole address you can send all 6 bytes.

Type **`ifmac sm0 0x01:0x02:0x03:0x04:0x05:0x06`**



Reset your kit and type **`ifmac sm0`** The new mac address will be displayed.



1.6 Lab 6: File Systems

1.6.1 Abstract

This lab will teach you how to utilize file systems on the Zoom OMAP35x Development Kit.

1.6.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit

- A development workstation (PC or laptop)
- An available serial port or USB to serial adapter
- An SD card
- A terminal emulation program such as HyperTerminal or TeraTerm. **Note:** Logic recommends using TeraTerm, a free program downloadable from the Internet.

1.6.3 Basic instructions

Using the same process you learned in the Section 1.1 “Lab 1: Kit Communications” open your terminal emulation program and set the port settings to:

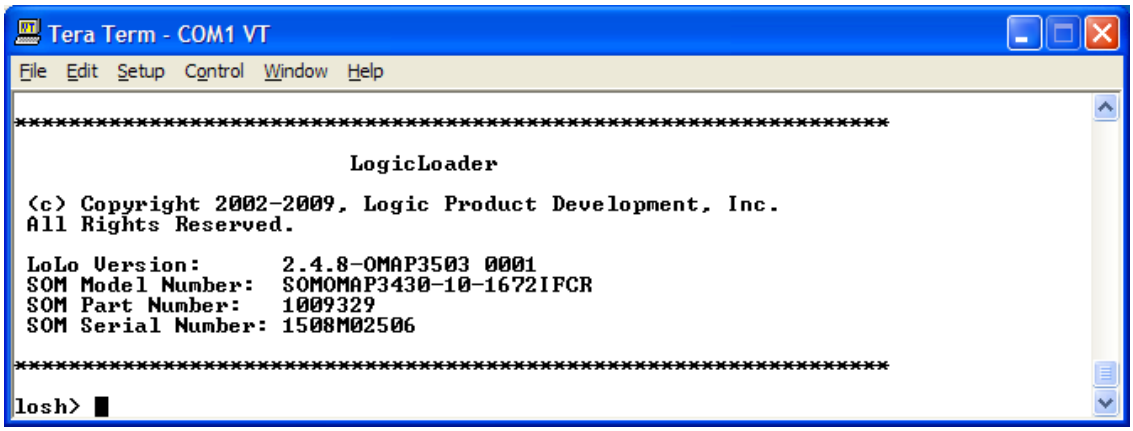
- 115200 BAUD rate
- 8 data-bits
- No parity
- No hardware flow control
- No software flow control

1.6.4 FATFS (Fat File System)

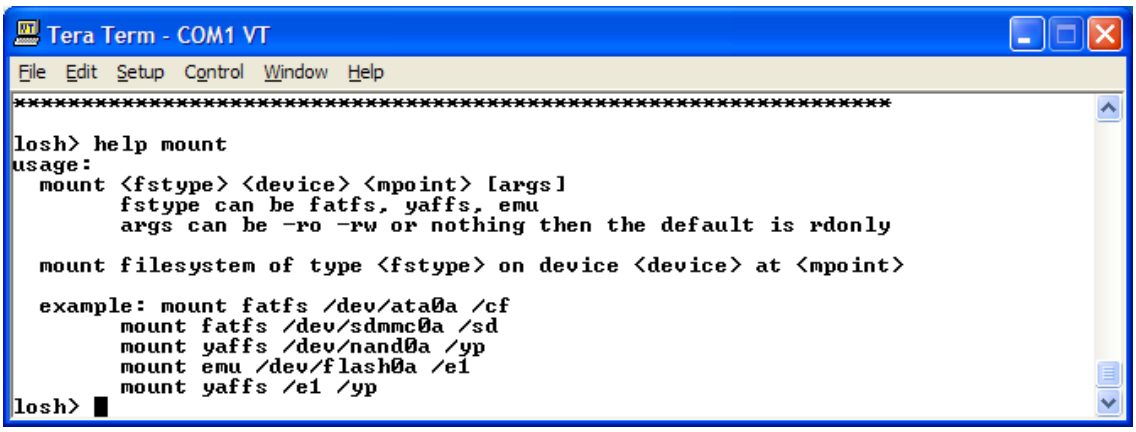
1.6.4.1 Mount an SD Card

The first method of loading a file onto your kit that we will cover in this lab is to use a secure digital (SD) card. This section describes how to mount an SD card in LogicLoader.

1. Store a file on an SD card.
2. Insert the SD card into the Development Kit’s SD card slot.
3. Start a terminal emulator program on the development workstation and connect the serial cable, provided with the Development Kit, from the workstation to the Development Kit.
4. Power up the Development Kit and verify the LogicLoader prompt appears in the terminal emulator window.

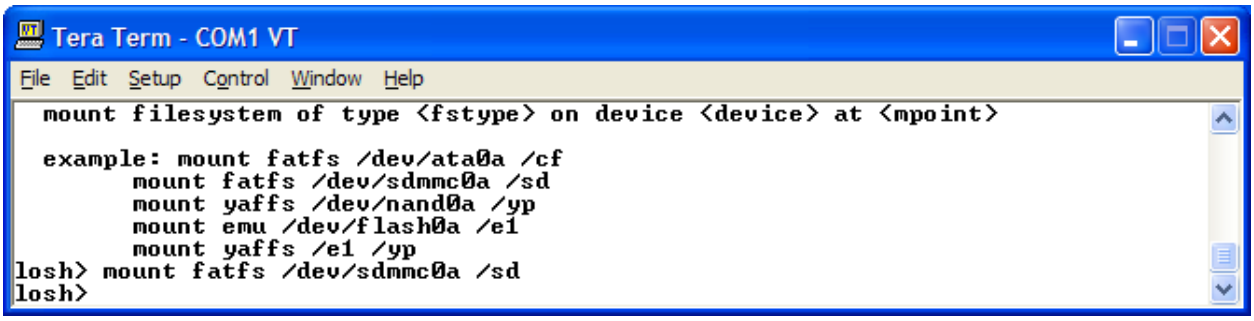


5. First let's look at the help command output for the mount command. Type **`help mount`**.

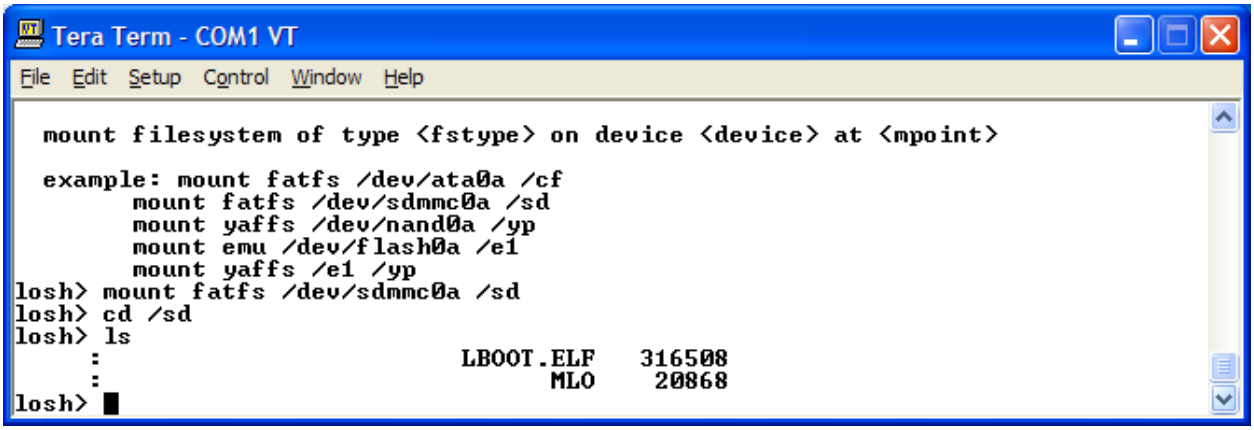


Remember that variables in the <> are required and variables in the [ ] are optional. Help usually gives the most frequently used options in the examples it provides.

6. Type **`mount fatfs /dev/sdmmc0a /sd`**. In TeraTerm you can simply highlight the sample line and right click by the losh prompt. Then hit "Enter". The screen displayed should be similar to the one shown below.



7. Now you can use the **`cd /sd`** command to change directories to the SD card, and then type **`ls`** to confirm that your card has been mounted and your file is on the card.



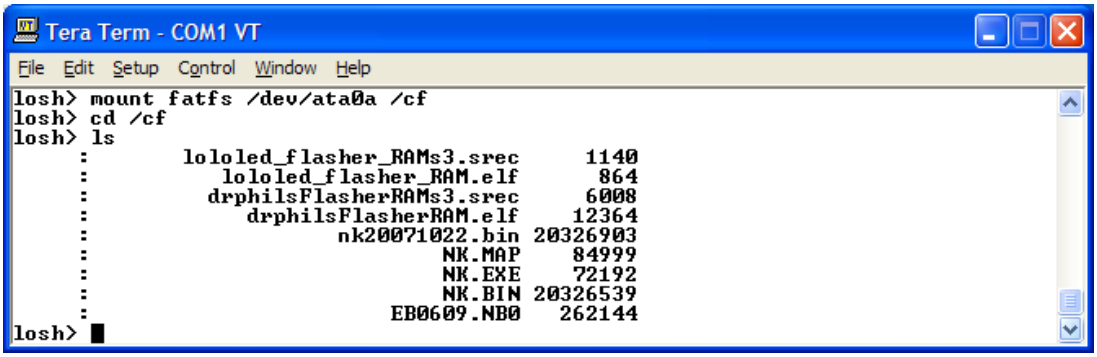
 Logic Product Development Company, All Rights Reserved

8. You now have access to the files on the SD card. If you don't want to use an SD card, you can store the files on a Compact Flash card.

1.6.4.2 Mount a CompactFlash Card

Mounting a CompactFlash card occurs in a similar manner to mounting an SD card.

Type **`mount fatfs /dev/ata0a /cf`** then **`cd`** and **`ls`** to view the files on the CF card.



If you need to store the file directly on the kit, you will have to store it in YAFFS. We will cover one method for doing that in Section 1.6.5.

1.6.5 YAFFS (Yet Another Flash File System)

1.6.5.1 YAFFS Overview

The acronym YAFFS stands for the phrase "Yet Another Flash Filing System." YAFFS was developed by a company named Aleph One Limited and incorporated by Logic Product Development into the LogicLoader software program. The partition entries for YAFFS partitions are not persistent—*they must be restored on each boot*. However, the partitions and data remain persistent.

1.6.6 Working with YAFFS in LogicLoader

1.6.6.1 Developing a Partition Scheme

You can use LogicLoader to mount up to four YAFFS partitions at a time. The following limits are imposed on partitions:

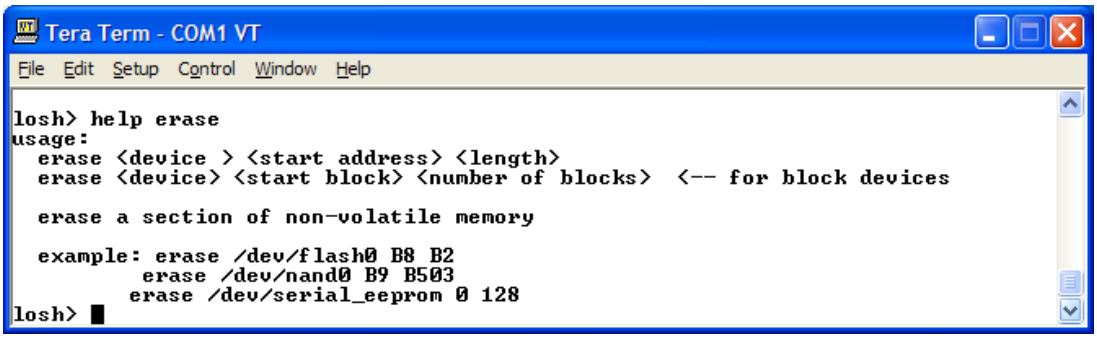
- Each partition must have a unique name.
- Each partition must exist on local flash accessible from LogicLoader's */dev/flashx* device file (where 'x' is an instance index). For example*: /dev/flash0* or */dev/flash1*.
- Each partition must span at least four physical flash blocks.
- A partition **must not** overlap the flash blocks that contain LogicLoader or its configuration block. For information about the location of these items, check the *LogicLoader User's Manual* addendum for your hardware.

1.6.6.2 Formatting YAFFS Partitions

All file systems need to be formatted before they can be mounted. Because YAFFS was designed from the ground up to work with embedded flash technologies, it understands an 'erased' flash device to be both formatted and empty. To prepare your partition for mounting, use LogicLoader's 'erase' command to erase the area of flash where the partition is to be located.

 Logic Product Development Company, All Rights Reserved

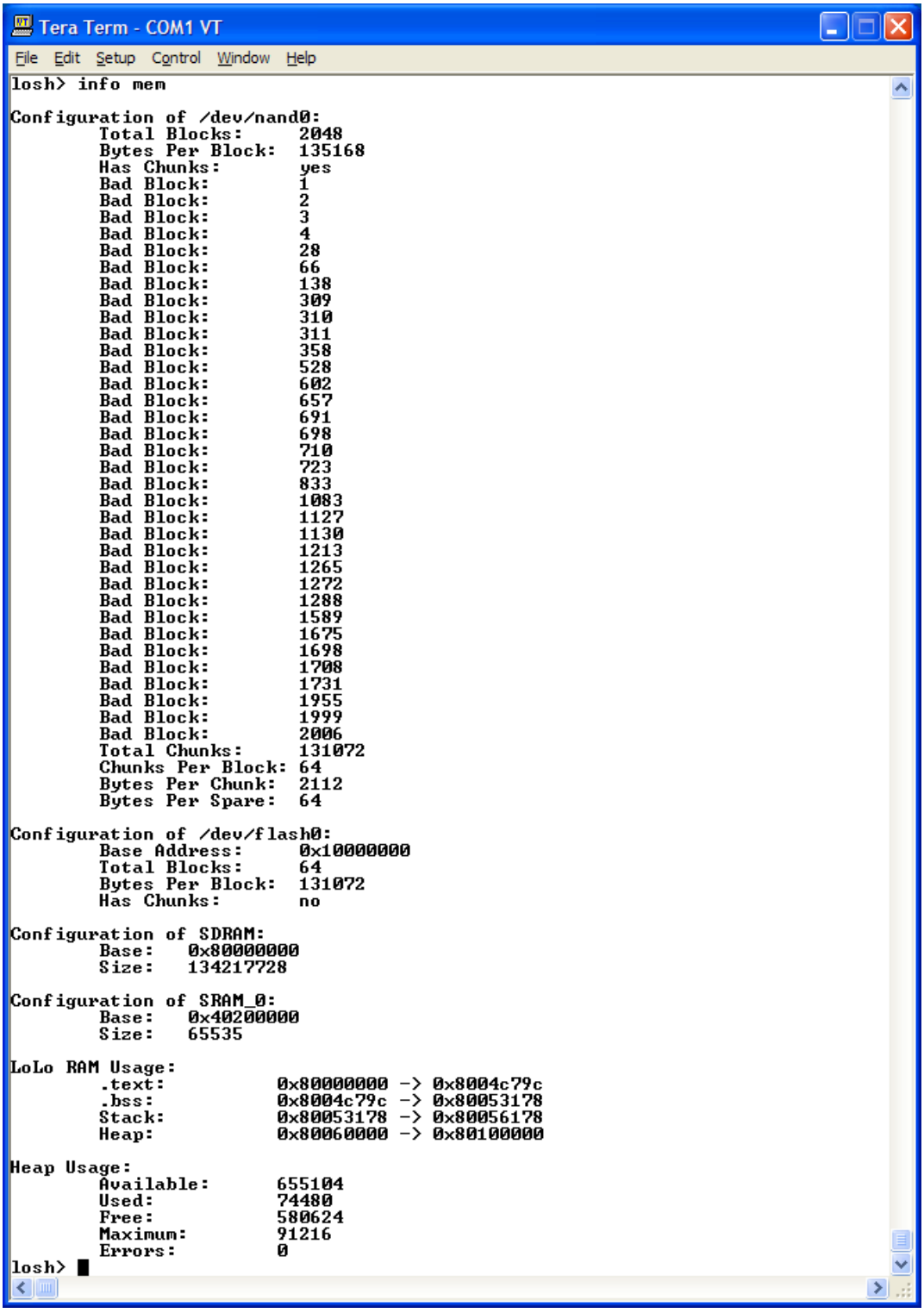
You should first erase the section of flash you are going to use. Several different parameters are available when using the 'erase' command; to see what our options are type **`help erase`**.



Help gives us three examples. The first is a NOR example and the second is a NAND example. The third example is for the serial eeprom, which is only available on legacy SOMs.

Generally, it is a good idea to know what you are erasing before you do it. To find out what is in your kit's memory, type **`info mem`**.

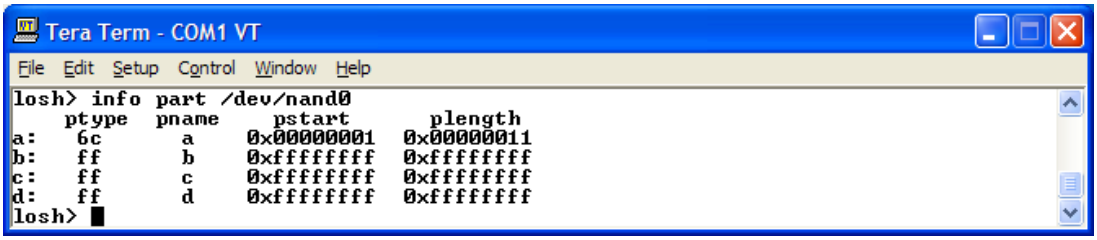
 Logic Product Development Company, All Rights Reserved



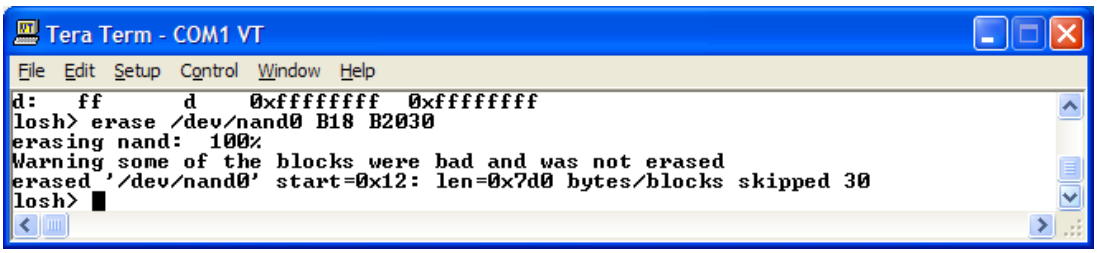
A NAND Example:

NAND can be erased using blocks. If there are bad blocks marked by the manufacturer, you might need to adjust your partition location. Since LoLo and the config block now reside in NAND, let's look at where they are located so we can erase the rest. Type **`info part /dev/nand0`**

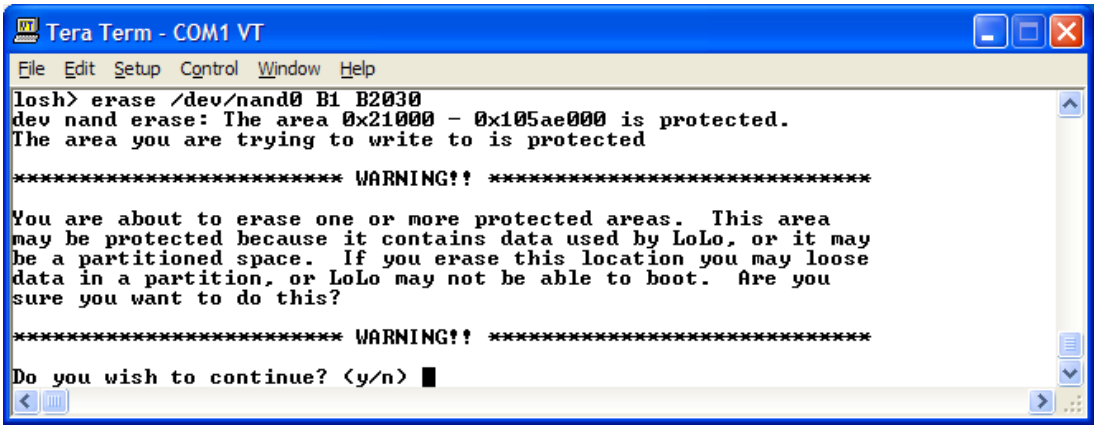
 Logic Product Development Company, All Rights Reserved



On this kit, we already have a partition “a” Lolo resides in B0 – B4. The config block resides in B5 – B17. To erase the rest of NAND for our partition type **`erase /dev/nand0 B18 B2030`**



One word of caution before we leave the ‘erase’ command; if you set your erase parameters incorrectly, and erase part of LogicLoader, your kit will not boot.



LoLo will warn you, so you have the option to say no. If you do erase part of LoLo, you can recover by booting from SD and updating LoLo. Please follow the directions in the NAND Flash Recovery download on our web site or the LoLo Update lab.

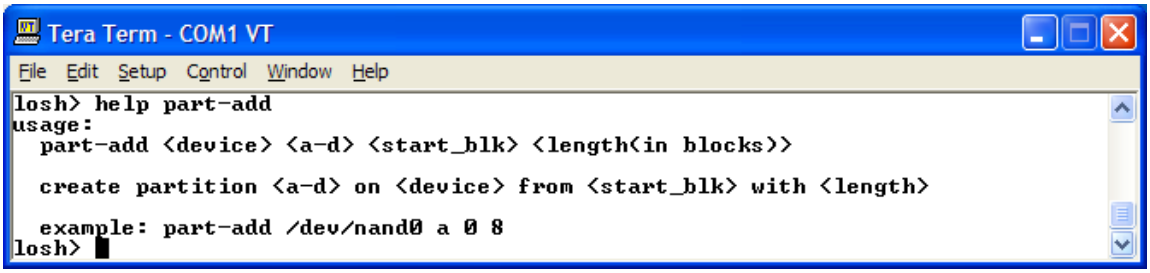
1.6.6.3 Adding YAFFS Type Partitions

LogicLoader maintains a partition table in RAM. Before a YAFFS partition can be mounted, it must be added to the partition table. To do this, the part-add command is used. The part-add command takes the following arguments:

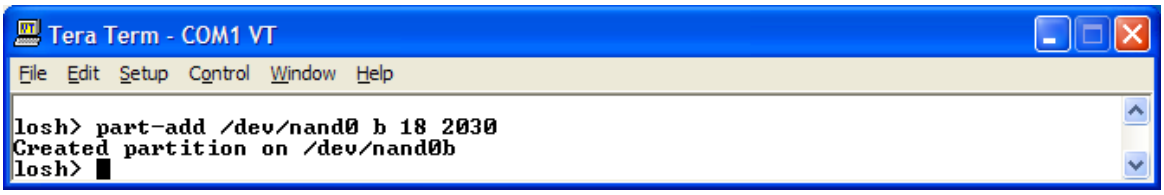
- <device> type of flash device the partition resides on
- <partition> which partition it is (a-d)
- <start> the physical starting address of the partition
- <length> the length (in blocks) of the partition

Type **`help part-add`**.

 Logic Product Development Company, All Rights Reserved



The device we're adding the partition to is nand0. Since LoLo resides in the first partition, this is the second partition and it will be partition b. The partition will start at block 18 and go to block 2048. Type **`part-add /dev/nand0 b 18 2030`**



Why 2030 in the command? Remember that the second parameter is the length, not the end of the partition.

Notice that the partition was added on `/dev/nand0b`, when we go to mount it, we'll need to use that in the mount command.

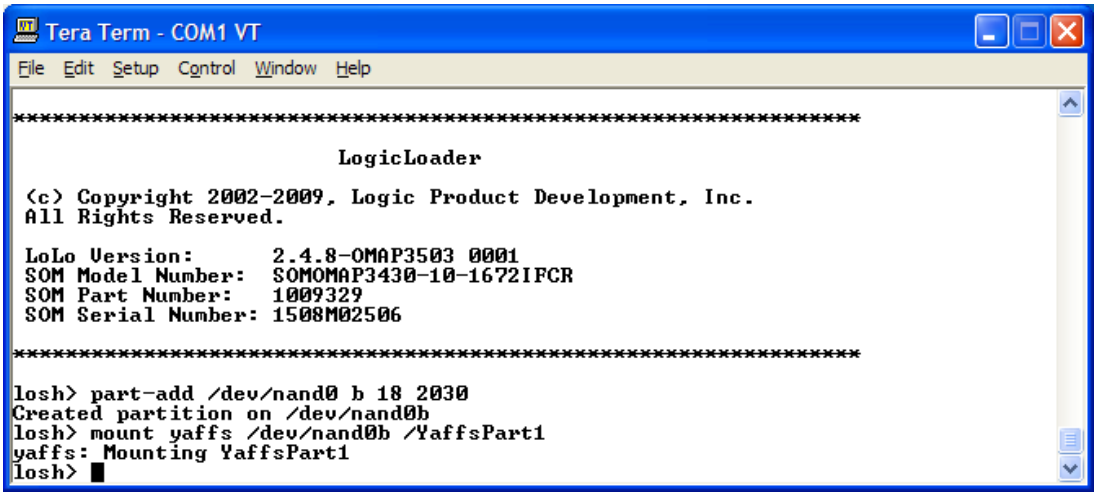
1.6.6.4 Mounting the Partition

To mount a partition in the partition table, the mount command is used. This command takes the following arguments:

- <fstype> the type of file system being mounted
- <point> the name of the YAFFS partition

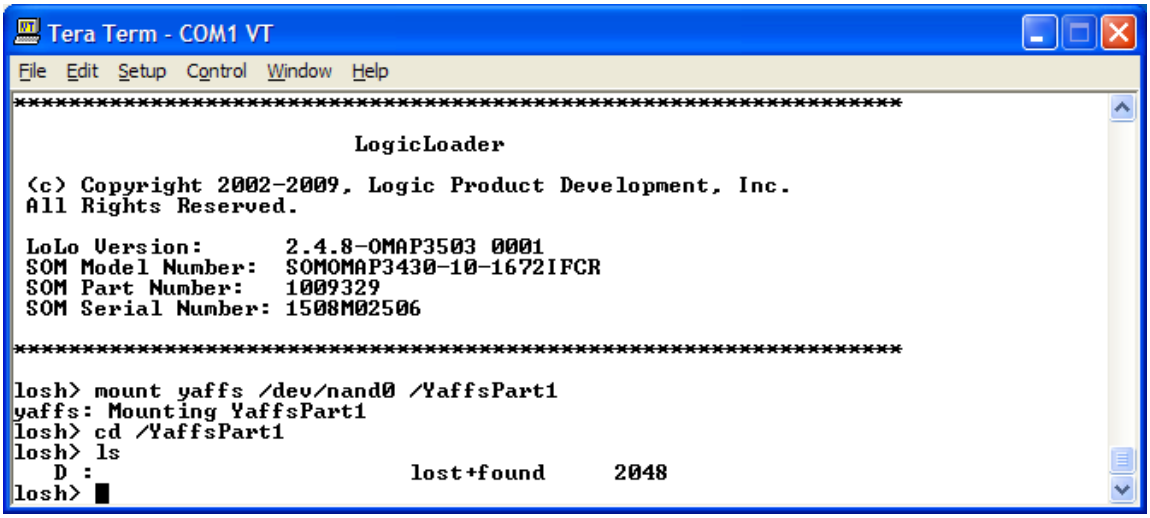
To use the new mount command you'll want to type
`mount yaffs /dev/nand0b /YaffsPart1`

Now change directories and see the contents of your new partition.



You can now store a file in nonvolatile memory so you don't have to download it to the kit every time you boot. Unfortunately, you still have to set up the partition and mount the partition after every power cycle.

LogicLoader also has a shortcut that will allow you to add the rest of flash to a partition and mount it all in one command. Type **`mount yaffs /dev/nand0 /YaffsPart1`**



This is equivalent to typing:

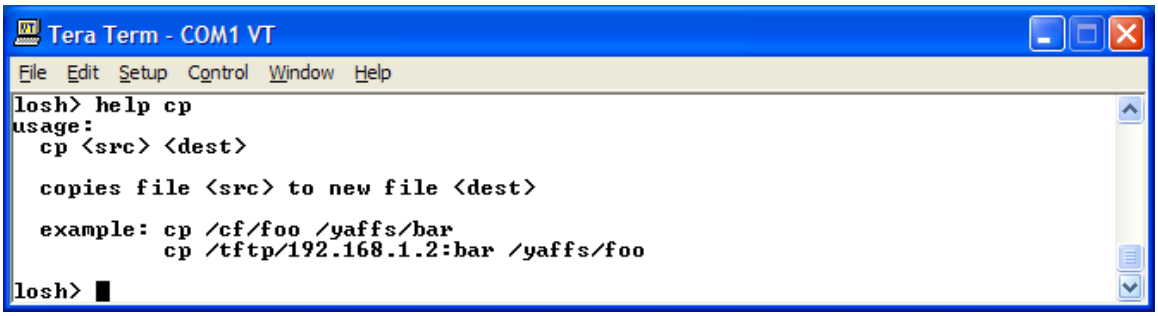
```
part-add /dev/nand0 b 18 2030
```

and

```
mount yaffs /dev/nand0b /YaffsPart1
```

1.6.6.5 Copy a file to the partition

Now that you have the partition, let's put a file in it. Type **`help cp`**



The two examples are to copy from a CompactFlash card and TFTP to a YAFFS partition. If you are going to use TFTP you will need the IP address of your TFTP server and will have to have an Ethernet connection to the server. Review Section 1.5 "Lab 5: Network Connection" if you need a refresher.

Using the example from the help window, we'll copy an NK.BIN from TFTP to our YAFFS partition—remember, your IP address will be different and you have to setup your kit for communications before you are able to copy the file.

In the next lab you will learn how to store scripts that can automate all of these processes and boot an image automatically.

1.7 Lab 7: Using Scripts

1.7.1 Abstract

This lab will teach you how to use scripting to optimize your application on your Zoom OMAP35x Development Kit.

1.7.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit
- A development workstation (PC or laptop)
- An available serial port or USB-to-serial adapter
- An image to load onto your kit and boot with a script
- A terminal emulation program such as HyperTerminal or Tera Term. **Note:** Logic recommends using Tera Term, a free program downloadable from the Internet.

1.7.3 Basic instructions

Using the same process you learned in Section 1.1 “Lab 1: Kit Communications” open your terminal emulator and set the port settings to:

- 115200 BAUD rate
- 8 data-bits
- No parity
- No hardware flow control
- No software flow control

 Logic Product Development Company, All Rights Reserved

1.7.4 Connect the kit to your workstation

1. Use the gray serial cable marked “LOGIC” to connect the serial port on the kit to the appropriate port on your workstation

2. Apply power to the kit and you should see the LogicLoader welcome prompt similar to the image below.

1.7.5 Basic scripting guidelines

Scripts can be used to automate any command or command sequence that can be entered on the command line. Scripts can exist as a single text file containing the commands you want to execute separated by semicolons. They can also be a text file with each command on a separate line.

You can enter scripts using the same syntax you use at the `losh>` prompt. In fact, that is exactly what you are doing. You are setting up a list of LoLo commands to be executed one after the other. By separating those commands by a semicolon or a new line, you are telling LoLo how to handle the script.

Launching scripts

If your script is saved as a text file, you can run a script from Tera Term by using the ‘Send | File’ option when LoLo is waiting at the `losh>` prompt. Let’s make a very simple script called video_test to open your display, clear the display, and run the draw-test.

The text file will look like:

```
video-open 15 16; video-clear; draw-test;
```

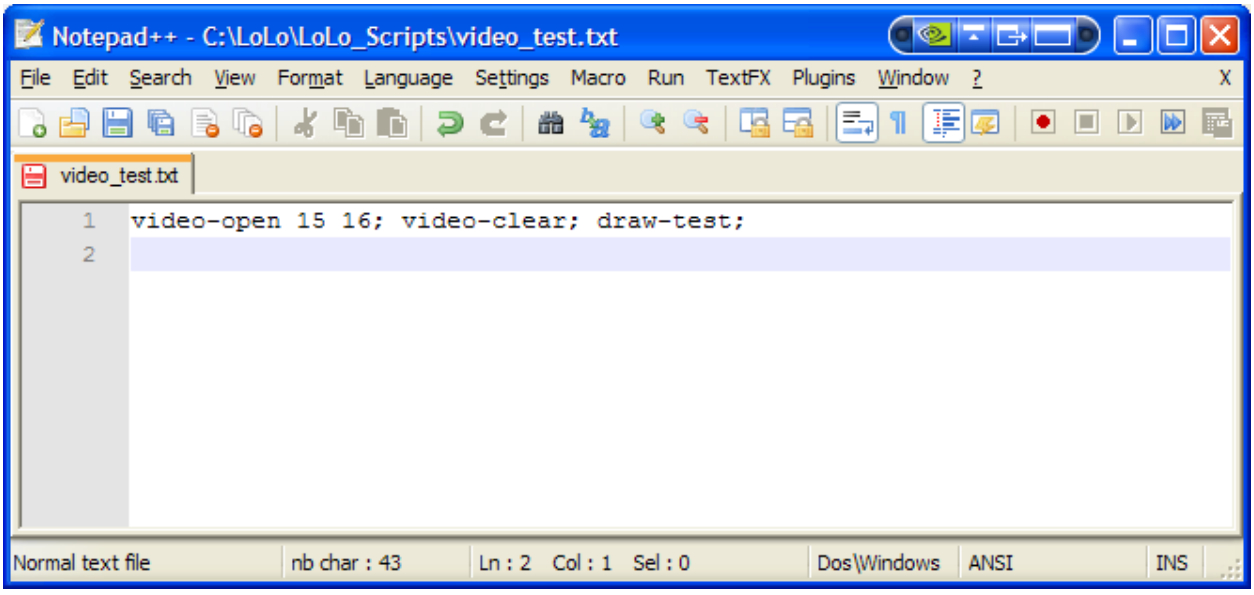
Or it can look like this:

```
video-open 15 16
video-clear
draw-test
```

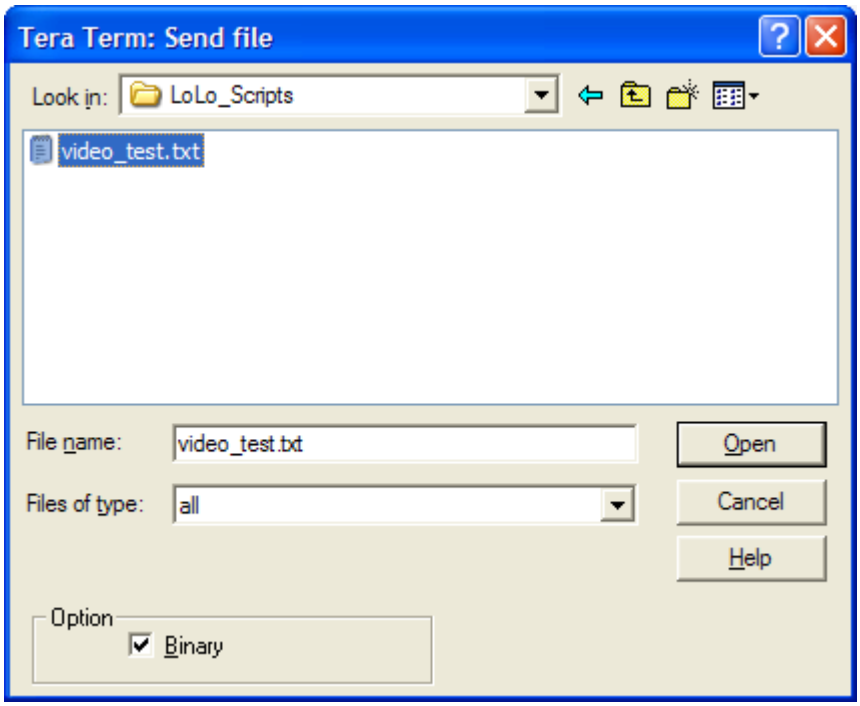
Be sure to place a carriage return after the last command to execute that last command, otherwise you’ll have to press the “Enter” key to see the test pattern.

 Logic Product Development Company, All Rights Reserved

1. Open a text editor program and create a text file using commands above.
2. Save the file as: video_test.txt

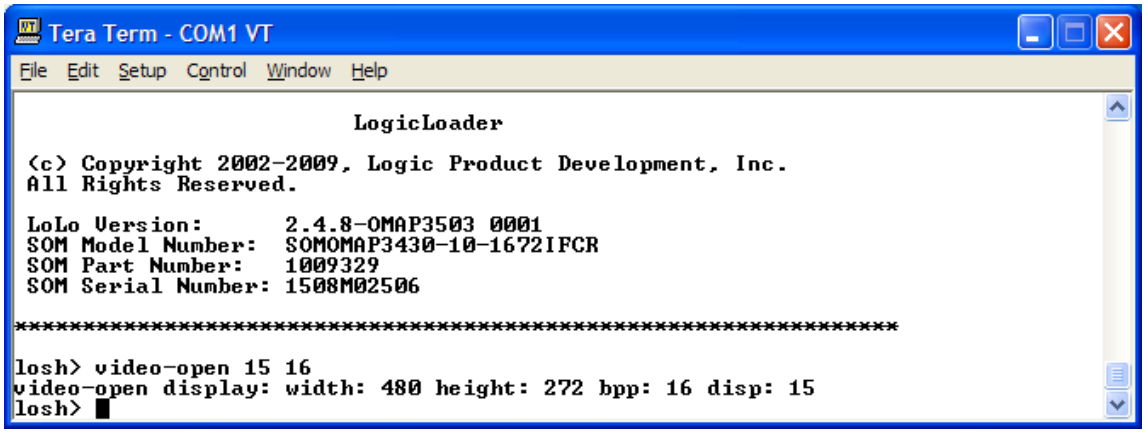


3. In Tera Term, click ‘File’ and select ‘Send’. Then locate the “video_test.txt” file you just created. Click the “Open” button to send the script to your development kit.

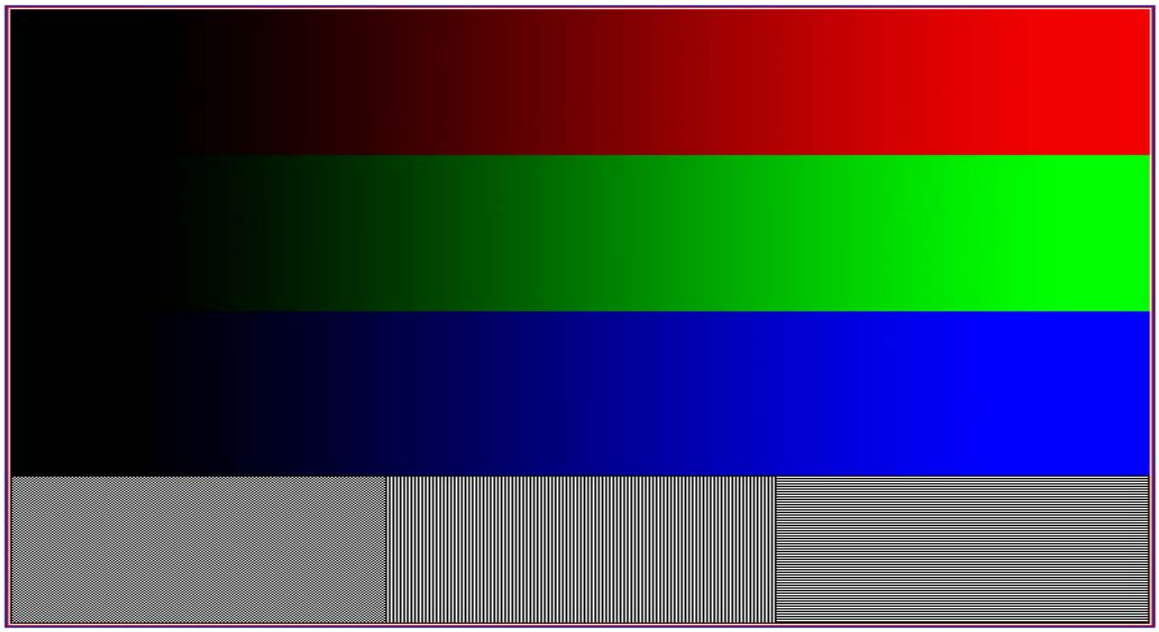


 Logic Product Development Company, All Rights Reserved

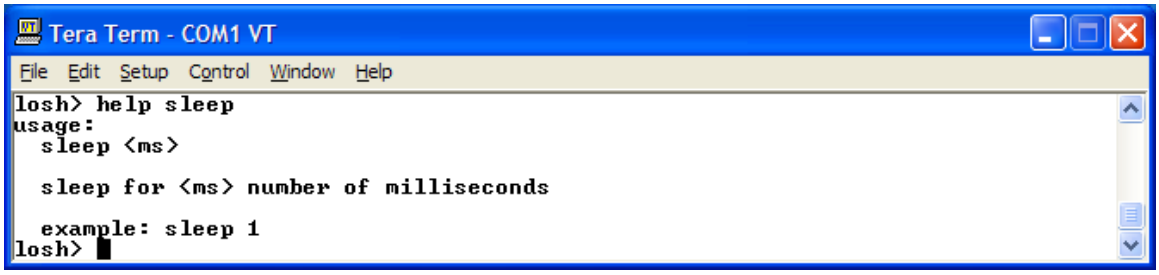
4. The Tera Term window will display output information similar to what you see below.



5. And your LCD display will show the following:



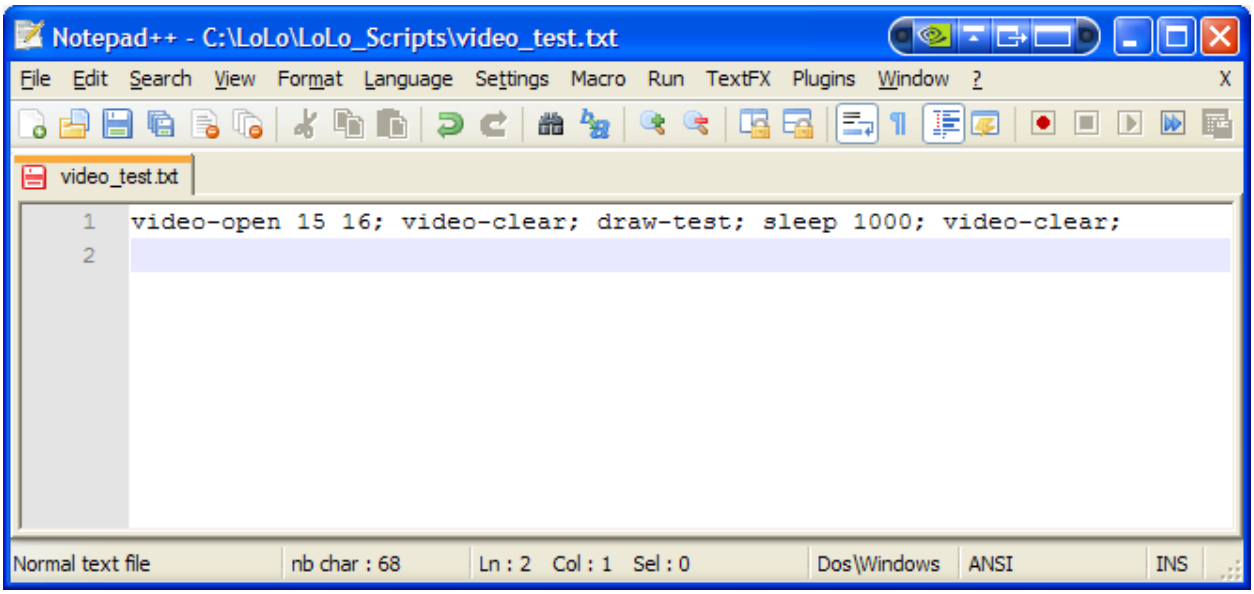
6. If you watched your screen when you sent the script, it should have flashed something and then gone to the test pattern. Let's put some time in between the commands, so we can see what is going on. Type **`help sleep`**



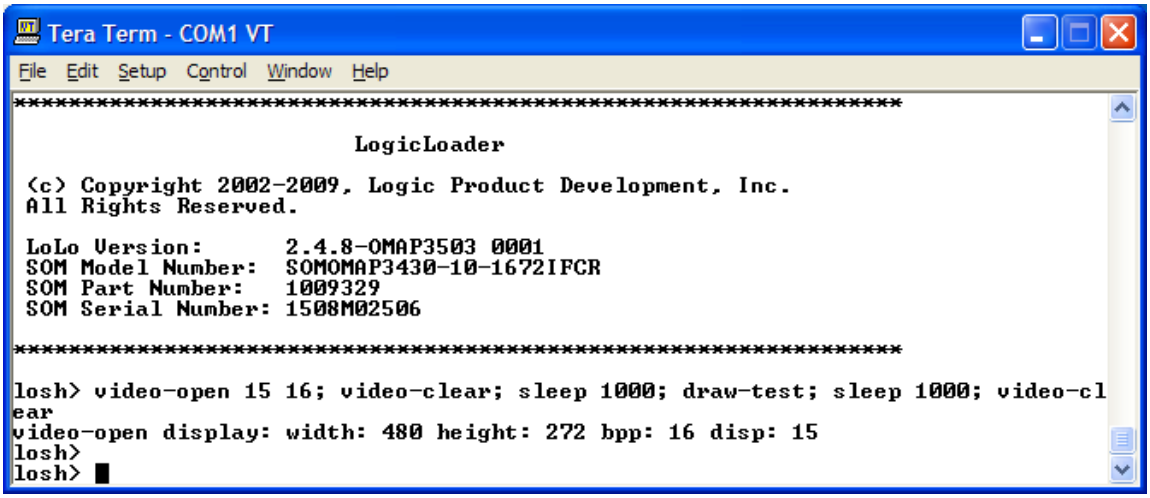
 Logic Product Development Company, All Rights Reserved

7. The help file lets us know we can make LoLo take a little nap by adding the sleep command between commands in our script. After each command, let's add **`sleep 1000;`** that should put 1 second in between each command. Let's also put another **`video-clear;`** at the end of the script so we can see the test pattern go away.

8. With these changes, your 'video_test.txt' file should look like this:



9. Save the text file and once again use 'Send | File' to send the script. Now you can actually see what the script is doing with each command.

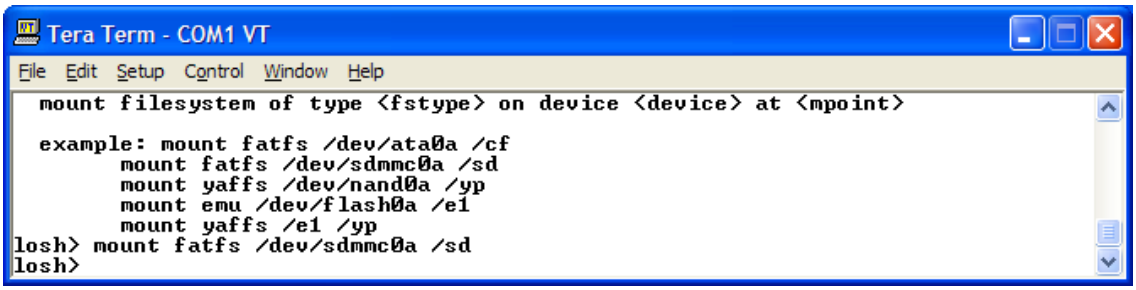


1.7.6 Launching a Script from an SD Card

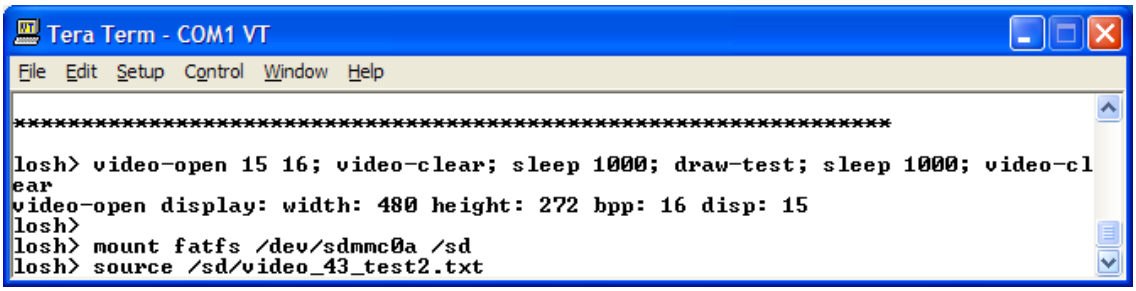
If you have a script saved as a text file on an SD card or on your kit, you can use the source command to run the script. Let's add the necessary commands to mount the SD card and then run the script from the card. We will be using the commands you learned in Section 1.6 "Lab 6: File Systems" to mount the SD card.

1. Using your PC and an SD card reader, save the video_test.txt file that you created to an SD card.

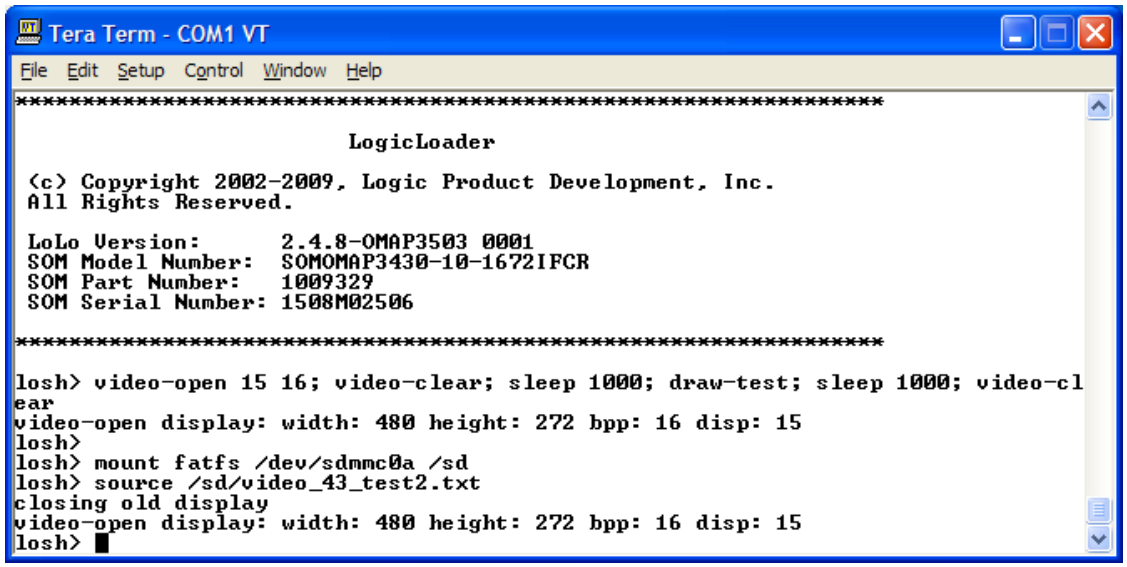
2. Type **`mount fatfs /dev/sdmmc0a /sd`**



3. Then type **`source /sd/video_43_test2.txt`**



4. You should see the screen on your kit flash white, then see the test screen pattern, and then see the screen return to white. Your LoLo output will look similar to the snapshot below.



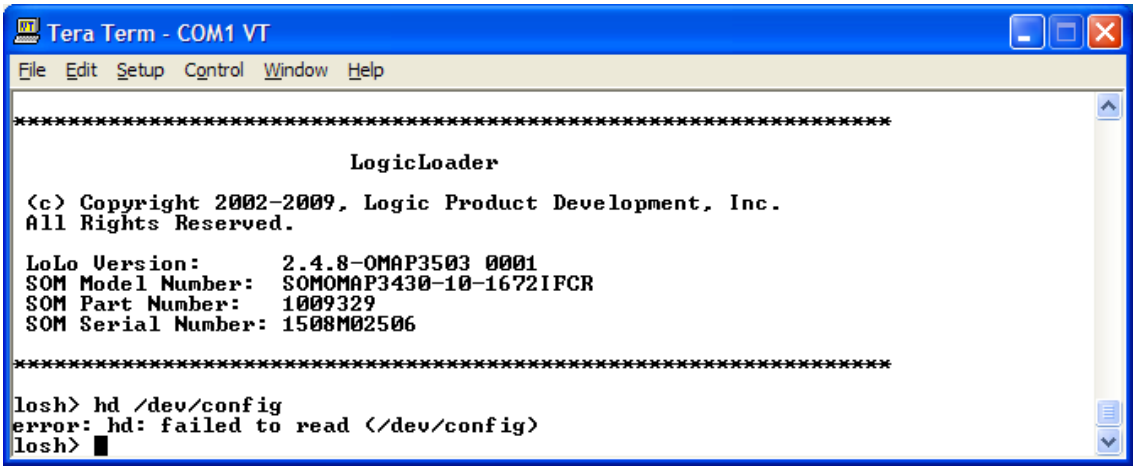
Note that the commands from the script on the SD card did not appear on the screen, but the test pattern did appear for 1 second and then was cleared. Nice work, but what happens when you reboot your kit? Nothing; you have to go through the commands all over again. To overcome this, we'll look at how to store a script so it runs every time you power-up your kit.

1.7.7 Persistent Script Storage

In order for a script to persist across power cycles, the script must be stored to a local, non-volatile memory device on the system. The primary script storage mechanisms supported by LogicLoader on the OMAP35x SOM is the resident flash array (*/dev/config*).

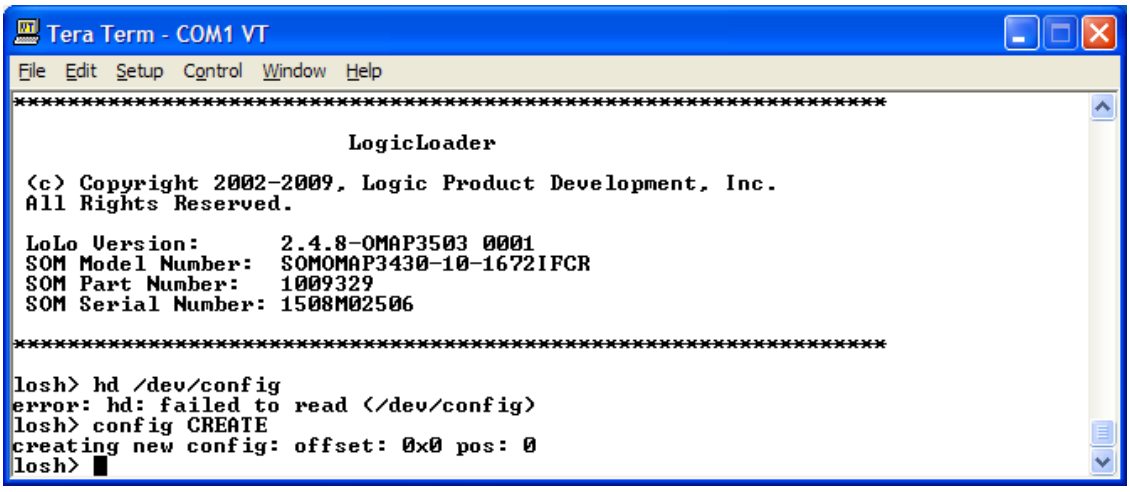
1.7.7.1 Config Block Basics

1. Before we get started, let's look at what is currently in the config block.
   Type **`hd /dev/config`**



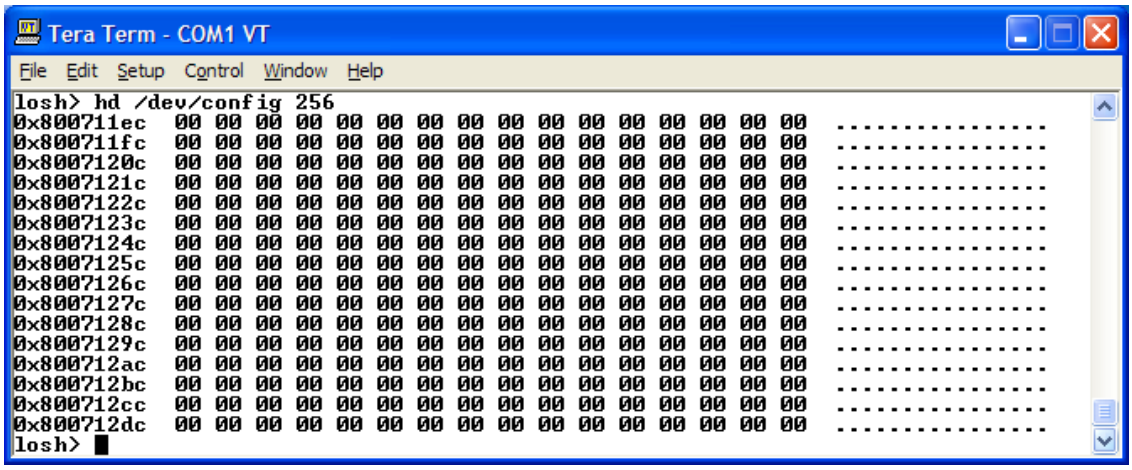
Hmm… not quite what we were looking for. It seems that this kit has never had the config block initialized. That's okay; it is important to learn how to do that too. If you have a brand new kit you must use the 'config CREATE' command to initialize the config block. You can also use it to erase a config block to get back to a clean slate.

2. Type **`config CREATE`** at the `losh>` prompt.



 Logic Product Development Company, All Rights Reserved

3. Now that we've cleared out the config block, let's take a look at it.
   Type **`hd /dev/config 256`**



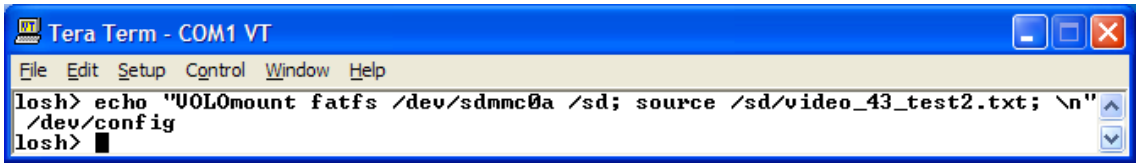
A nice clean slate. Let's put a script in the config block that will run our video test from the SD card.

1.7.7.2 Persisting scripts with the 'echo' command

The 'echo' command can be used to store a script in the config block. To include a new line in the first argument to 'echo', it is necessary to enclose the whole argument in double-quotes "". Remember to end the script by inserting **`\n`** before the closing quotes to instruct the parser to stop parsing the file.

As we stated earlier, scripts are just a series of commands. They can do a whole lot more though with a simple word added to the beginning. If we start a script with the word **`VOLO`** it will execute at boot. Boot scripts can also start with **`LOLO...`** the difference is that a VOLO script talks to the debug serial port and a LOLO script doesn't—the V stands for Verbose.

1. All on one line, type:
   `echo "VOLOmount fatfs /dev/sdmmc0a /sd; source /sd/video_test.txt; \n" /dev/config`



2. If the video test script is on your SD card, you can reboot your kit and it should run the video test.

3. We know that we can write a script onto the kit with the 'echo' command and we know we can erase the script using the **`config CREATE`** command. What happens if we want to keep the script, but need to disable it? We can actually do that by writing a **""** script.

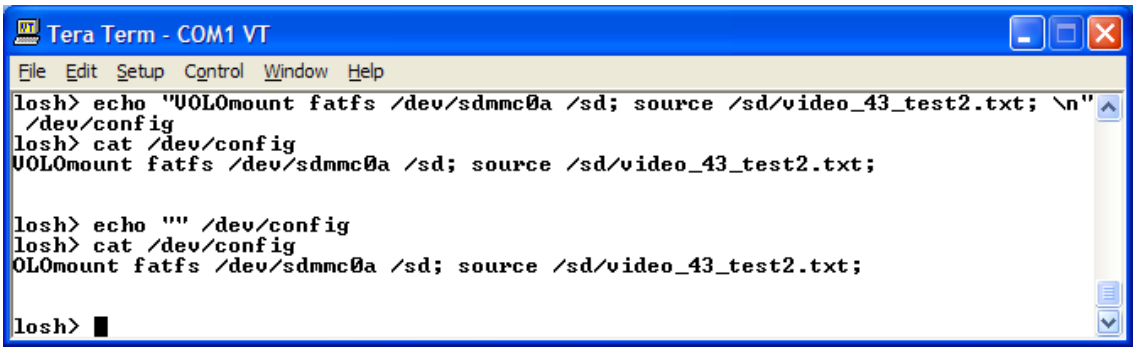
4. Type **`echo "" /dev/config`**

   We just changed the first character in the first command to a space - OLO. Now the kit will not recognize it as a boot script and will not run the script. We can use the 'cat' command to

 Logic Product Development Company, All Rights Reserved

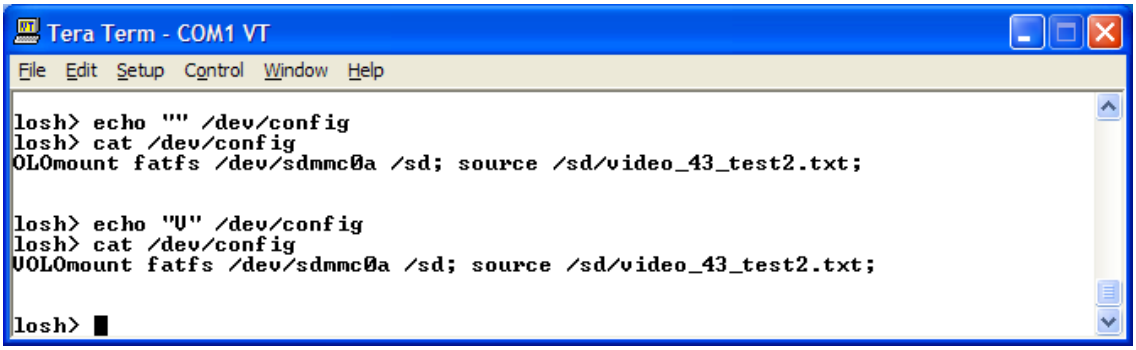
display what is in the config block without having to see all of the hex values in the memory locations.

5. Type **`cat /dev/config`**



It is the same as what was there before, but the V is missing. Since LoLo doesn't recognize OLO as a boot time script, it will just boot to the `losh>` prompt. Let's put the V back in that boot script.

6. Type **`echo "V" /dev/config`** then use **`cat /dev/config`** to display the config block contents.

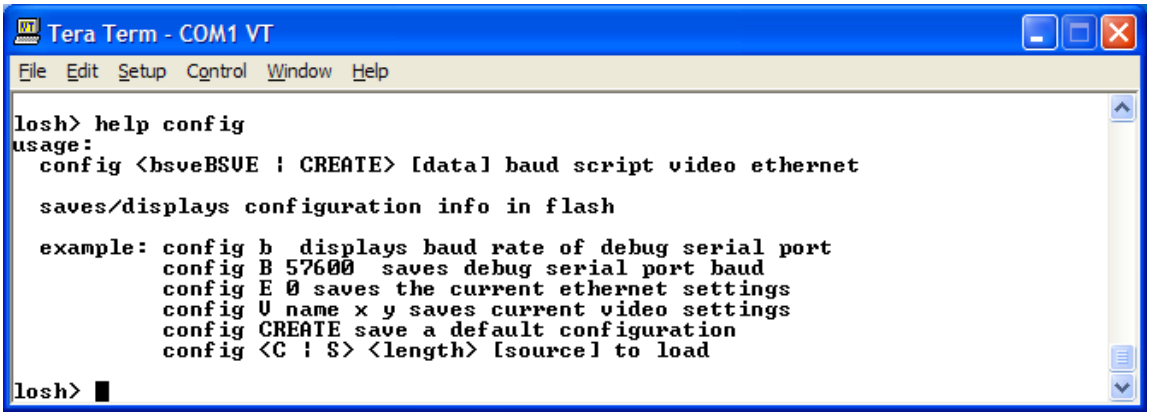


We put the V back into the script and re-enabled it. You can make scripts that do much more than we can cover here.

Using the echo command with long scripts can be a pain, as it is easy to make a simple typing mistake. Often developers will write a long script in a text editor and then use the **`config S`** command to load the file's contents into the config block.

Let's look at the options for storing a script in the config block. Type **`help config`**.

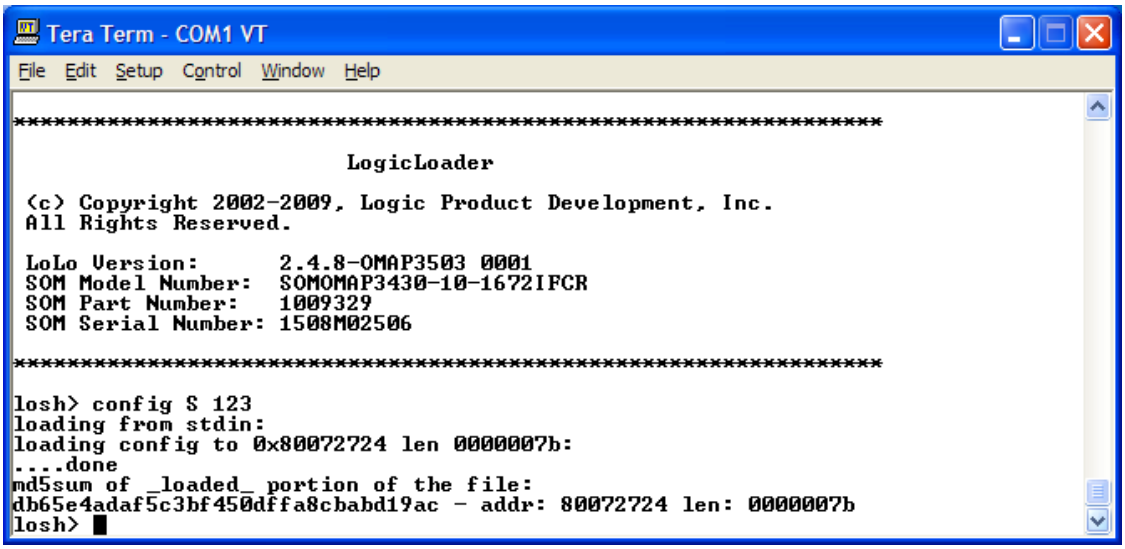
 Logic Product Development Company, All Rights Reserved



Let's use the `config S <The length of the script> [file location]` command.

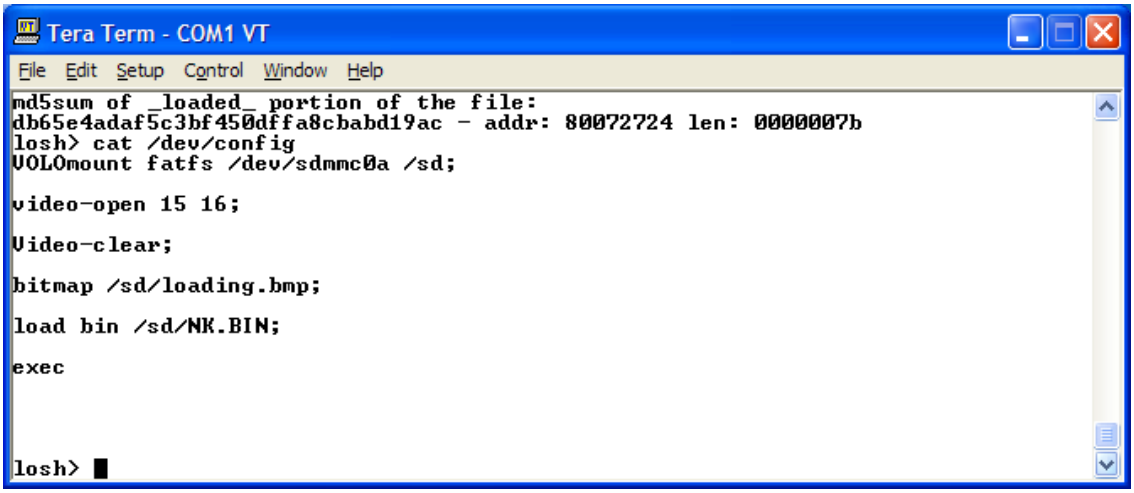
Create a text file called script.txt and save it on your PC. It doesn't matter what it does, it can be an extended version of the video test script, or it can be a boot script with VOLO or LOLO as the first command. For this example, we'll have a boot script that mounts the SD card, initializes the screen, clears it, and places a bitmap image on the screen while the CE image loads. It happens to be 123 Bytes long.

To save it to the config block, type **`config S 123`** then click on 'File' and 'Send File' and send the file over the serial port.



You can check to see if your script was saved properly with the cat command we used earlier.

 Logic Product Development Company, All Rights Reserved



Reboot your kit and it will run your script.

Before we leave this lab, you need to know one last thing about scripts.

If you store a boot script that doesn't return to LoLo—like a CE image boot script—you won't be able to get back to the `losh>` prompt by rebooting; the kit will just run the script again. To get out of this situation you can hold down the "q" key when you boot. LoLo checks the keyboard first and will not run the script if the "q" key is detected. When you return to the `losh>` prompt, you can disable your script if you need to.

1.8 Lab 8: Loading an NK.BIN

1.8.1 Abstract

This lab will teach you how to use scripting to load an NK.BIN file and boot it on your kit.

1.8.2 What you will need

- A Zoom OMAP35x Development Kit or Torpedo Development Kit
- A development workstation (PC or laptop)
- An available serial port or USB-to-serial adapter
- An SD Card
- An Ethernet connection between the workstation and the kit
- A terminal emulation program such as HyperTerminal or Tera Term. **Note:** Logic recommends using Tera Term, a free program downloadable from the Internet.

1.8.3 Basic instructions

Using the same process you learned in Section 1.1 "Lab 1: Kit Communications" open your terminal emulator and set the port settings to:

- 115200 BAUD rate
- 8 data-bits
- No parity

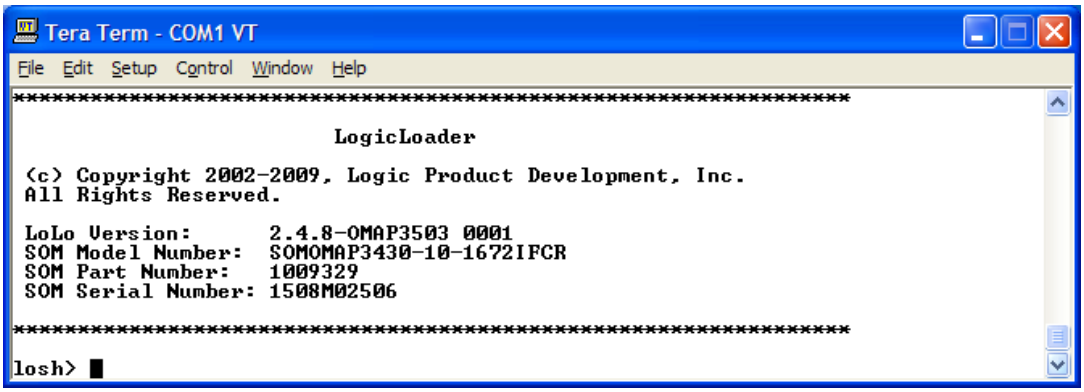
 Logic Product Development Company, All Rights Reserved

- No hardware flow control
- No software flow control

1.8.4 Connect the Kit to Your Workstation

Use the gray serial cable marked “LOGIC” to connect the serial port on the kit to the appropriate port on your workstation.

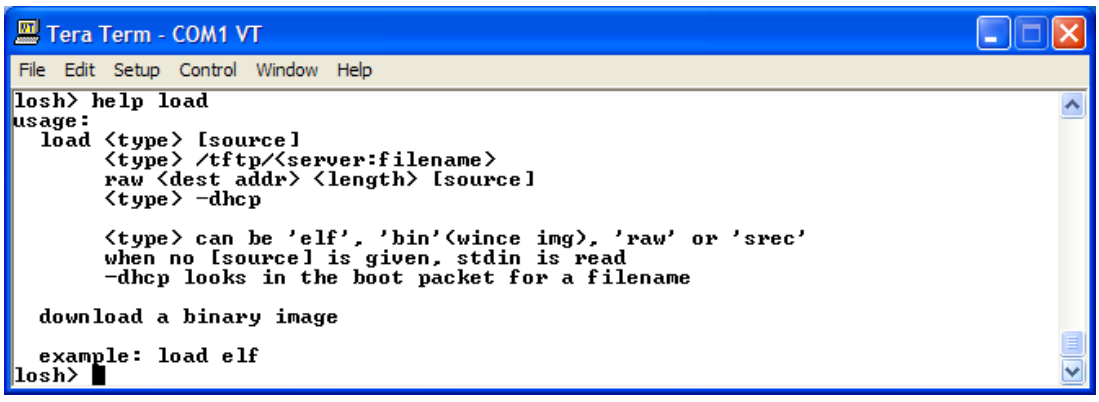
Apply power to the kit and you should see the LogicLoader welcome prompt similar to the image below.



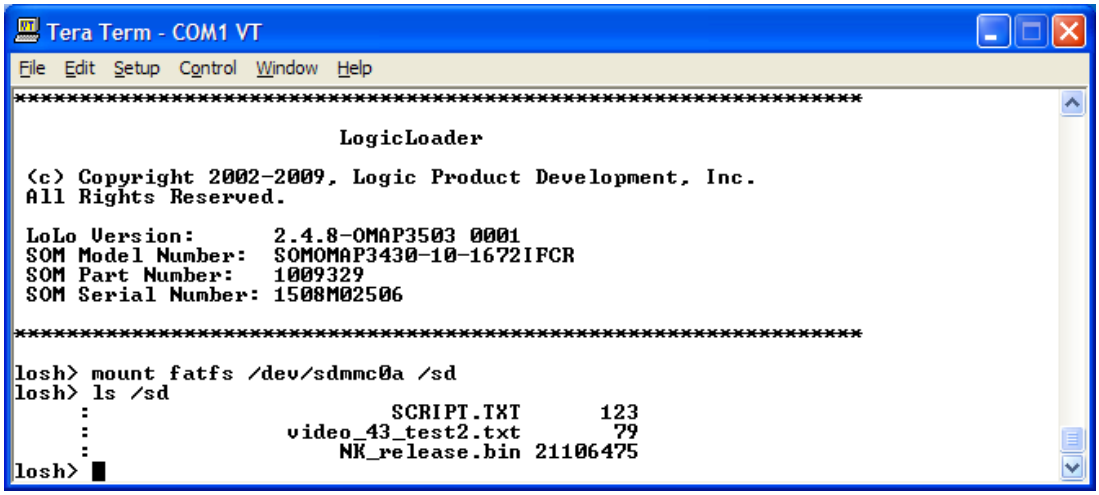
1.8.5 Load a File from Secure Digital (SD) Card

The OMAP processor has the ability to boot directly from an SD card. This ability is what allows Logic to ship SD cards that boot directly to Windows Embedded CE and Linux with the kit. If for some reason you would like to boot from NAND and then execute an OS image, you can do that by following the process explained here. The process for CompactFlash cards and YAFFS are very similar—mount the file system, load the image, execute the image. They will be covered briefly in later sections.

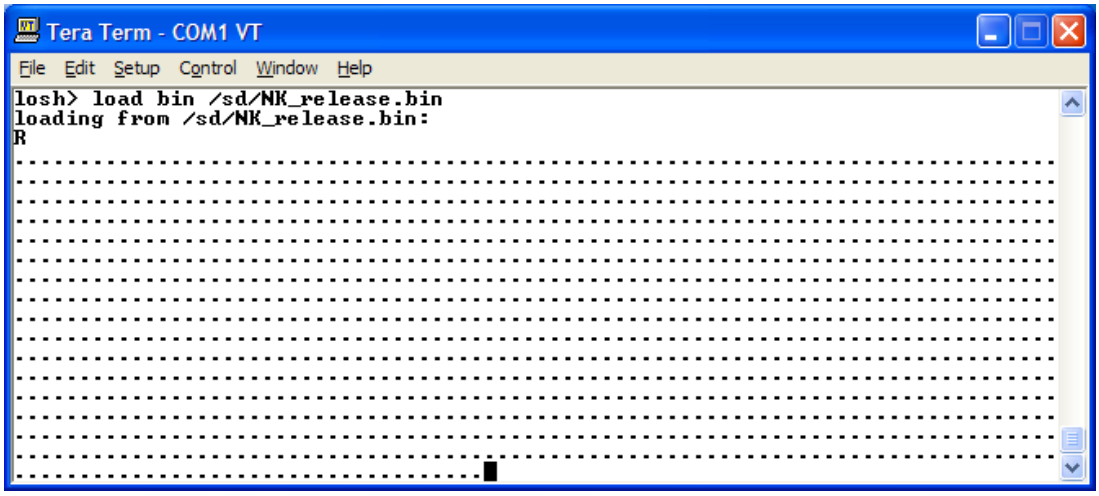
In previous labs you learned how to mount an SD card. We will now go a step further to load an NK.BIN Windows Embedded CE image from the SD card onto your kit. First let’s look at the help for the ‘load’ command. Type **`help load`**

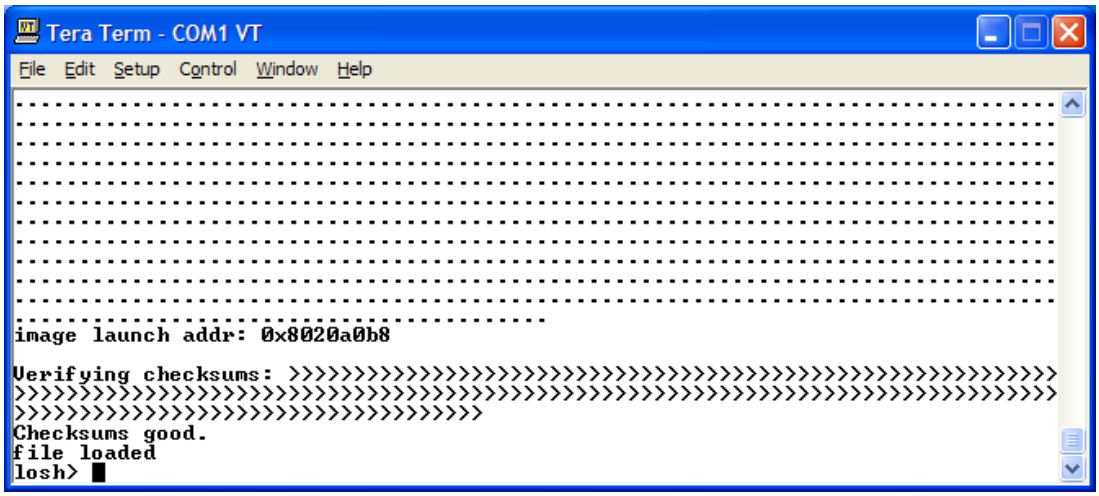


You can see from the output that there are several ways to use the ‘load’ command. We are interested in loading a Windows Embedded CE image, so we will use load bin. Use the commands you learned in Section 1.6 “Lab 6: File Systems” to mount the SD card. Now type **`ls /sd`** to view the contents of the card.



You can see the video-test script from the scripting lab and a file called “NK_release.bin”; that is the Windows Embedded CE image we are going to load. Now that we know the name and location of the file, we can use the ‘load bin’ command to load our image. Type **`load bin /sd/NK_release.bin`**

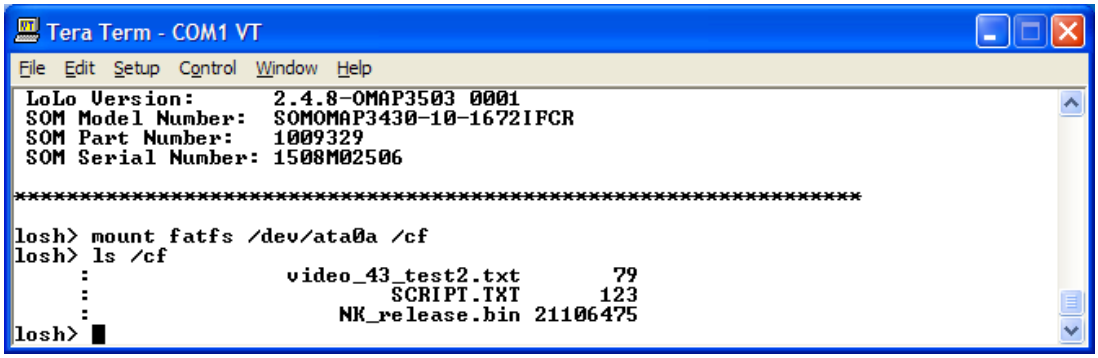




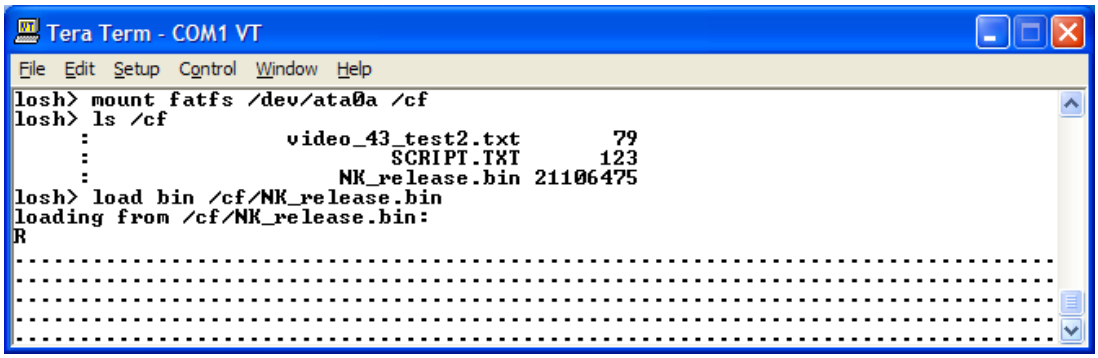
Congratulations! You have loaded an NK.BIN from SD. Next we're going to cover loading a file from CompcatFlash and YAFFS.

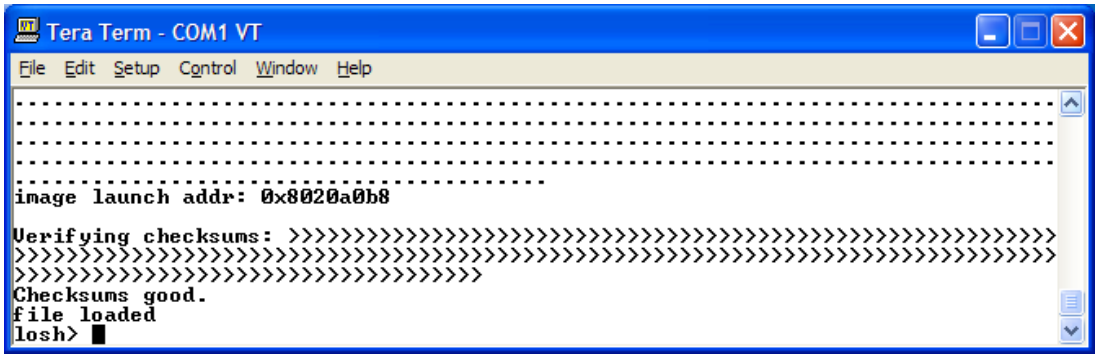
1.8.6 Load a File from CompactFlash

In previous labs you learned how to mount a CompactFlash card. Use the commands you learned in Section 1.6 "Lab 6: File Systems" to mount the CompactFlash card. Now type **`ls /cf`** to view the contents of the card.



In this case we made the CompactFlash card match the SD card, so the same files are visible on the CompactFlash card. Now type **`load bin /cf/NK_release.bin`** to load the image.





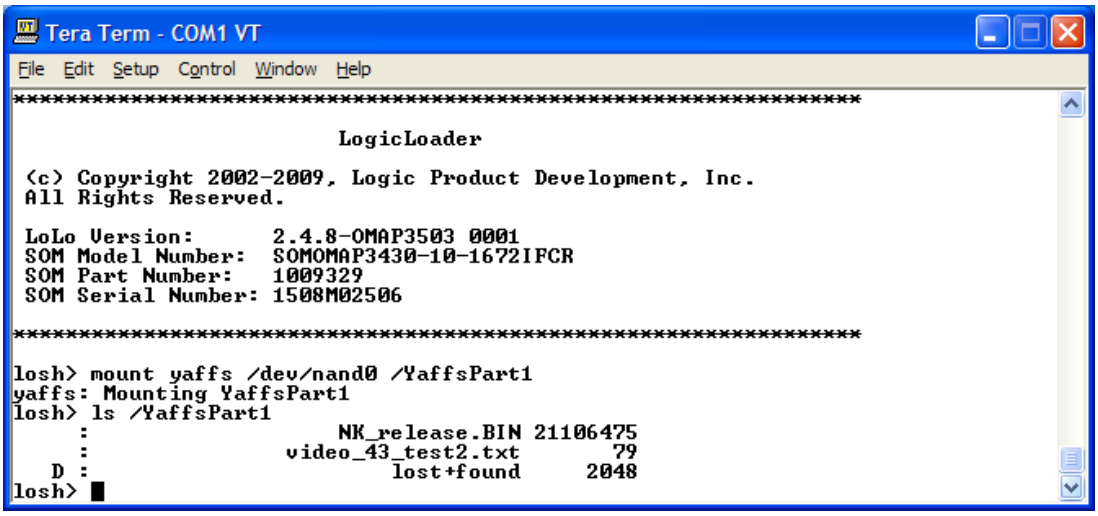
Congratulations! You have loaded an NK.BIN from CompactFlash. Last we'll cover loading a file from YAFFS.

 Logic Product Development Company, All Rights Reserved

1.8.7 Load a File from YAFFS

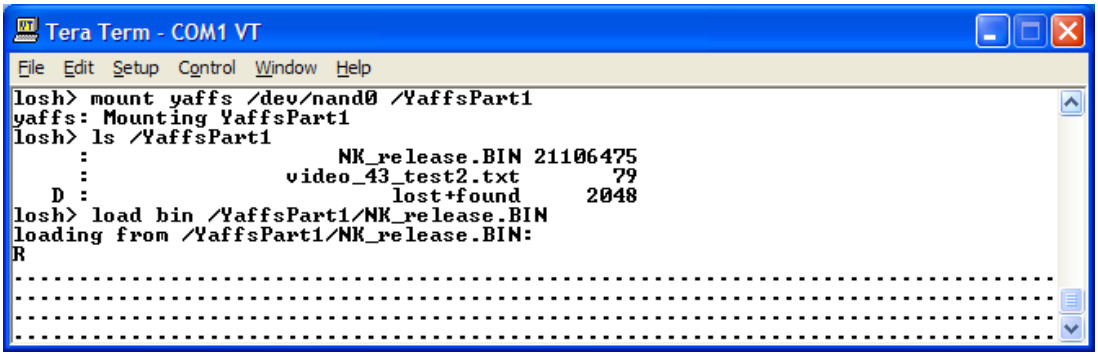
To load a file from YAFFS you must first have a YAFFS partition added to your device, you must have it mounted and you have to have a file saved in YAFFS to load. Please review Section 1.6.5 “YAFFS (Yet Another Flash File System)” for the basics on YAFFS.

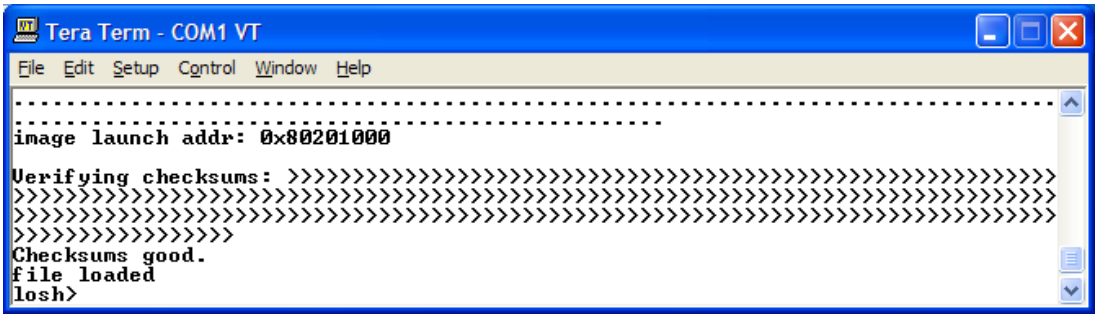
For now we’ll assume you have all these things and are ready to go. After the YAFFS partition is mounted, type **`ls /YaffsPart1`** to look at the files.



You can now use the load command to load your NK.BIN.

Type **`load bin /YaffsPart1/NK_release.BIN`**

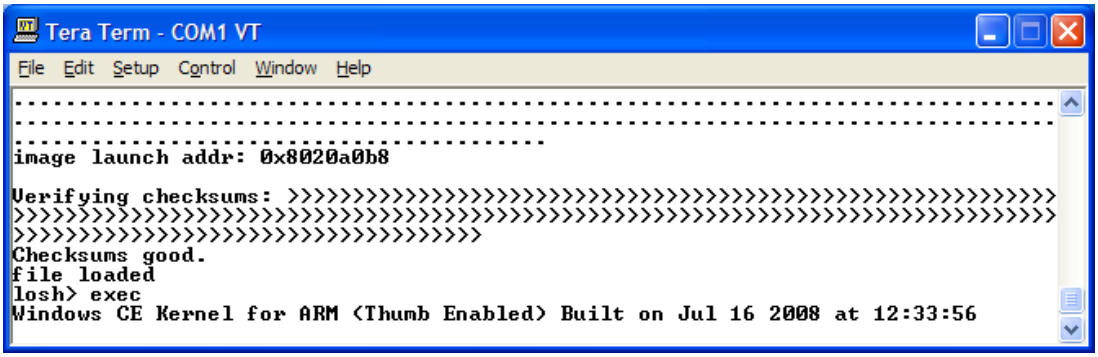




Congratulations! You now have an NK.BIN sitting in RAM waiting for you to do something. So now what?

1.8.8 Execute a Windows Embedded CE Image

Once you have an image loaded, you need to execute it by typing **exec** at the losh> prompt.



After this line, there will be a long debug dialog and then you will see your program running.

Congratulations, you now have a Windows Embedded CE image running on your Kit.

 Logic Product Development Company, All Rights Reserved