



# i.MX31 LogicLoader™ Labs

## LogicLoader Documentation (LoLo version 2.4)

Logic Product Development  
Published: April 2008

### Abstract

These informal LogicLoader™ labs provide a step-by-step introduction to basic LogicLoader commands and usage on the i.MX31 System on Module (SOM).

This file contains source code, ideas, techniques, and information (the Information) which are Proprietary and Confidential Information of Logic Product Development, Inc. This information may not be used by or disclosed to any third party except under written license, and shall be subject to the limitations prescribed under license.

No warranties of any nature are extended by this document. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipments. The only warranties made by Logic Product Development, if any, with respect to the products described in this document are set forth in such license or agreement. Logic Product Development cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

Logic Product Development may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except as expressly provided in any written agreement from Logic Product Development, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

© Copyright 2008, Logic Product Development, Inc. All Rights Reserved.

### REVISION HISTORY

| REV | EDITOR          | DESCRIPTION             | APPROVAL | DATE     |
|-----|-----------------|-------------------------|----------|----------|
| A   | Richard Laborde | Release for i.MX31 SOMs | RGL      | 04/23/08 |

This page intentionally left blank.



## LogicLoader Lab #1

### Kit Communications

---

#### 1 Abstract

This lab will teach you how to establish a serial connection to a Logic Zoom i.MX31 LITEKIT. A serial connection is the most basic form of communication between a development workstation computer and the development kit.

#### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- 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.

#### 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.

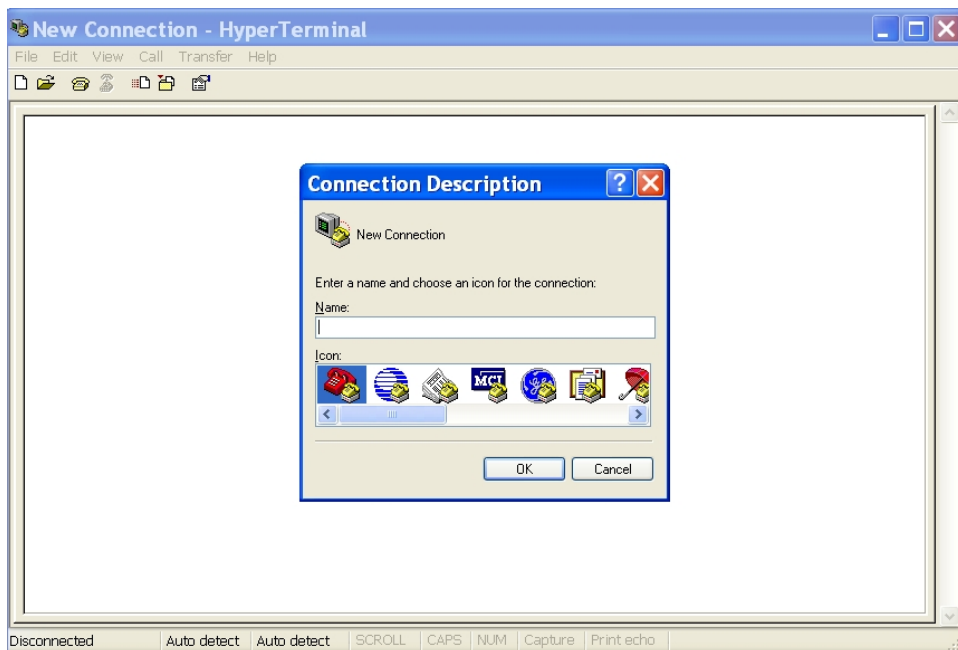
## 4 HyperTerminal instructions

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

### 4.1 Start the HyperTerminal program using either of these methods:

- Click on 'Start -> Run...' from the main Windows Start menu. Then type **hyperterm** and click 'OK.'
- Find the HyperTerminal icon using the Windows start menu. This is normally found under, 'All Programs -> Accessories -> Communications -> HyperTerminal.'

### 4.2 Name your new connection 'LPD' and press 'OK'



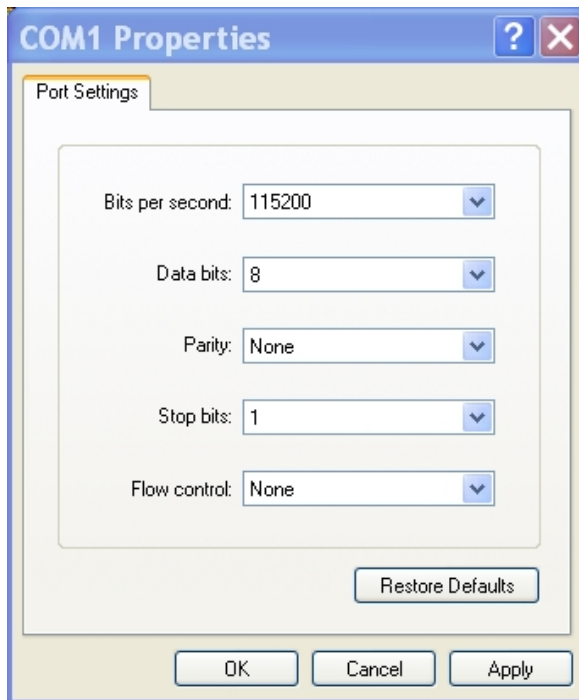
### 4.3 Using the drop-down menu, change the 'Connect using:' box to 'COM1' (or the appropriate COM port for your workstation)



**4.4 Click 'OK'**

**4.5 Change the Port Settings to:**

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

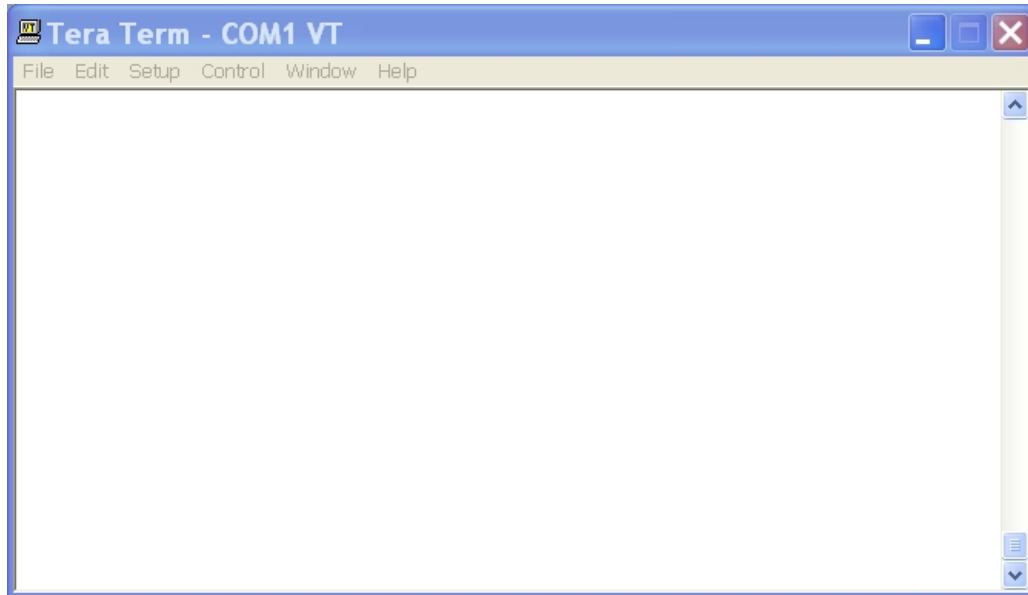


**4.6 Click 'OK'**

## 5 TeraTerm instructions

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

### 5.1 Start the TeraTerm program

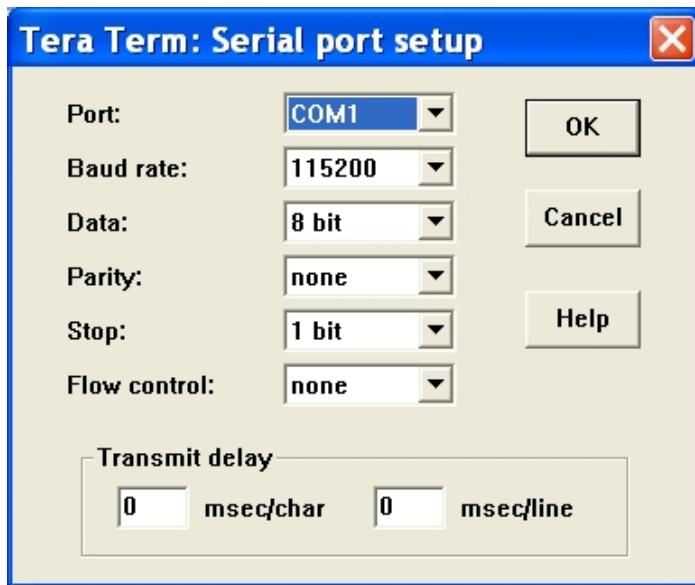


### 5.2 Select the 'Setup -> Serial port...' menu item

### 5.3 Select the appropriate COM port for your workstation

### 5.4 Change the Port Settings to:

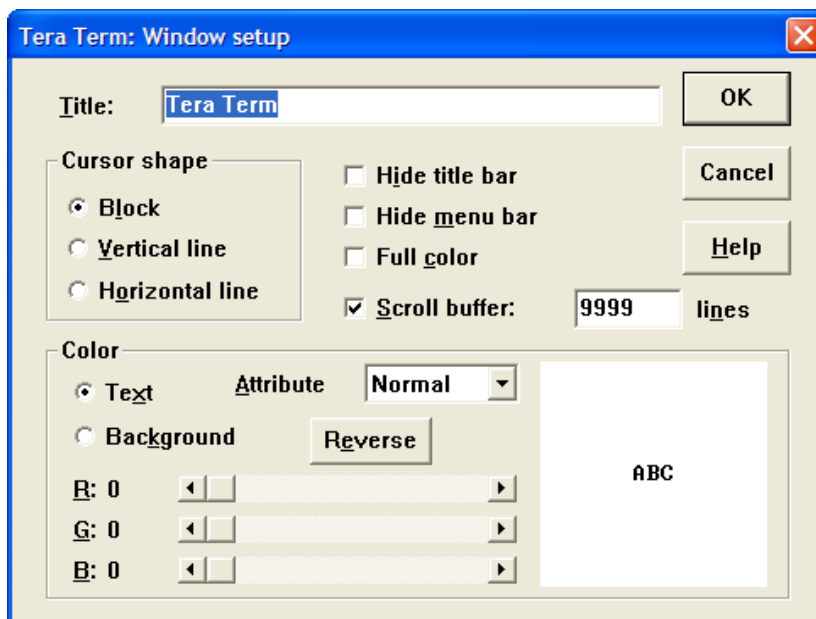
- Bits per second: 115200
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control: None



5.5 Click 'OK'

5.6 Select the Setup -> Window... menu item

5.7 Click the box for scroll buffer and make the amount as big as you like, up to 9999.



5.8 Select 'OK'

## 6 Connect the Kit to your workstation

- 6.1 Use the gray serial cable marked “LOGIC” to connect the serial port on the kit to the appropriate port on your workstation
- 6.2 Apply power to the kit and you should see the LogicLoader™ welcome prompt similar to the image below



The screenshot shows a Tera Term window titled "Tera Term - COM1 VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The main text area displays the following content:

```
*****  
                        LogicLoader  
  
<c> Copyright 2002-2006, Logic Product Development, Inc.  
All Rights Reserved.  
Version 2.4.0-IMX31_10 0001  
*****  
losh> █
```



## LogicLoader Lab #2

# Learning the LogicLoader™ Help System

---

### 1 Abstract

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

### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- A development workstation (PC or laptop)
- A serial port communications connection between the workstation and the development kit

### 3 Instructions

#### 3.1 Start your terminal emulator program (TeraTerm or HyperTerminal)

#### 3.2 Ensure the null-modem serial cable provided with the kit is connected to both the kit and the development workstation

#### 3.3 Power up the development kit

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

A screenshot of a Tera Term terminal window titled 'Tera Term - COM1 VT'. The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The terminal display shows a welcome message for LogicLoader, enclosed in asterisks. The message reads: 'LogicLoader', '<c> Copyright 2002-2006, Logic Product Development, Inc.', 'All Rights Reserved.', 'Version 2.4.0-IMX31\_10 0001'. Below the message is a prompt 'losh>' followed by a cursor. The terminal window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
losh> █
```

Note the string Version 2.3.5-IMX31\_10 0001 in the image above. This tells us that this kit is running revision 2.3.5 of the LogicLoader program and has been built specifically for the i.MX31 platform. If you suspect you are having issues with LogicLoader, one of the first things a Logic Product Development support person will ask you for is the version of LoLo and what platform you are running it on.

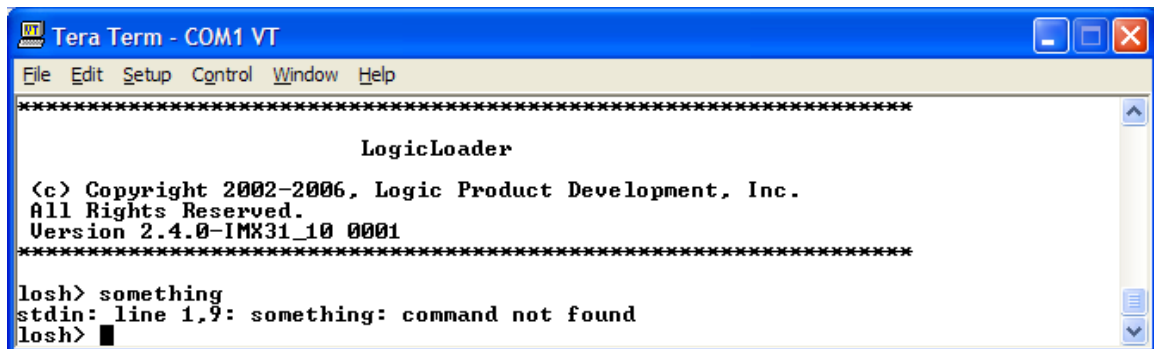
The string `losh>` is the LogicLoader command shell waiting for you to type something in.

### 3.4 Type the word “something” at the “losh>” prompt and press the “Enter” key



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
losh> something
```

You should receive output similar to the image below.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
losh> something
stdin: line 1,9: something: command not found
losh>
```

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 (TeraTerm 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
```

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.

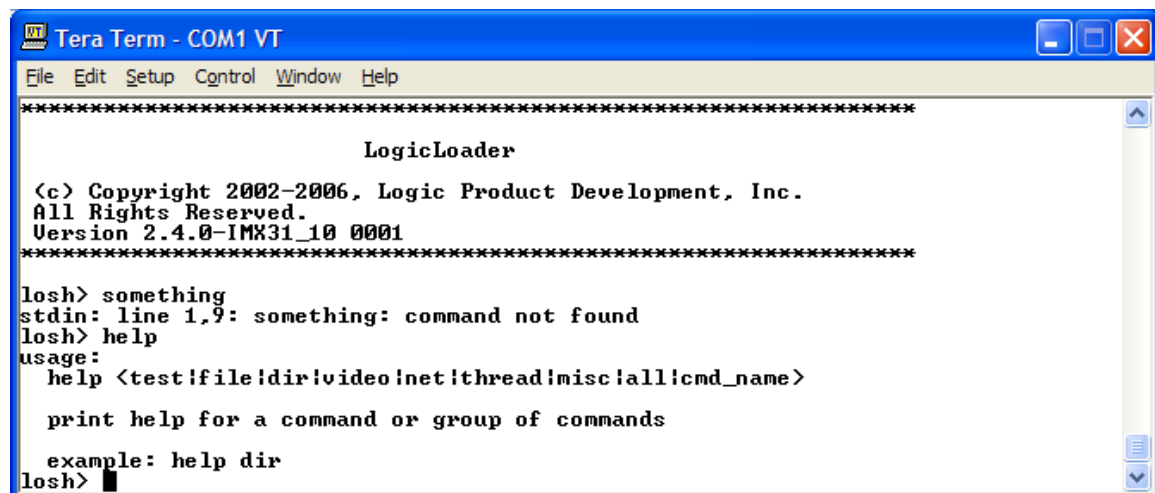
The next piece of information is the actual word `something`. This is LoLo printing out the input that caused the problem.

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?

### 3.5 Type the word “help” at the “`losh>`” prompt and press the “Enter” key



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> something
stdin: line 1,9: something: command not found
losh> help
usage:
  help <test!file!dir!video!net!thread!misc!all!cmd_name>

  print help for a command or group of commands

  example: help dir
losh>
```

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
```

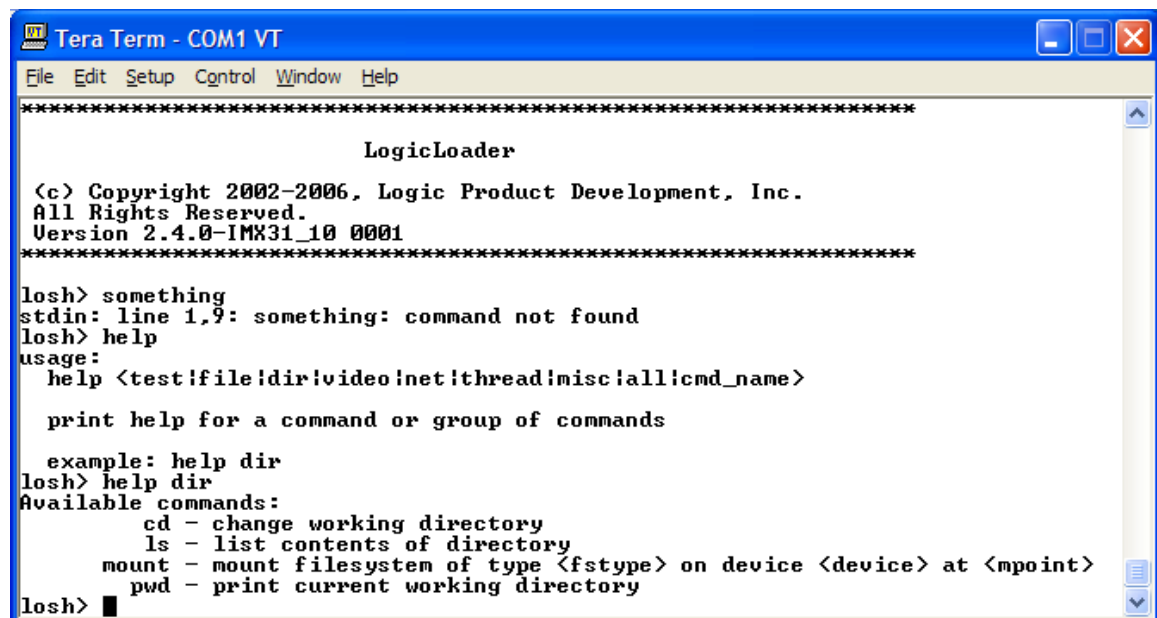
The output above is known as a command's help topic. If you type a valid LoLo command, but don't use it in a valid manner, LoLo will print out the command's help topic to remind you of the command's proper format and use.

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.

### 3.6 Type the words "help dir" at the "losh>" prompt and press the "Enter" key



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

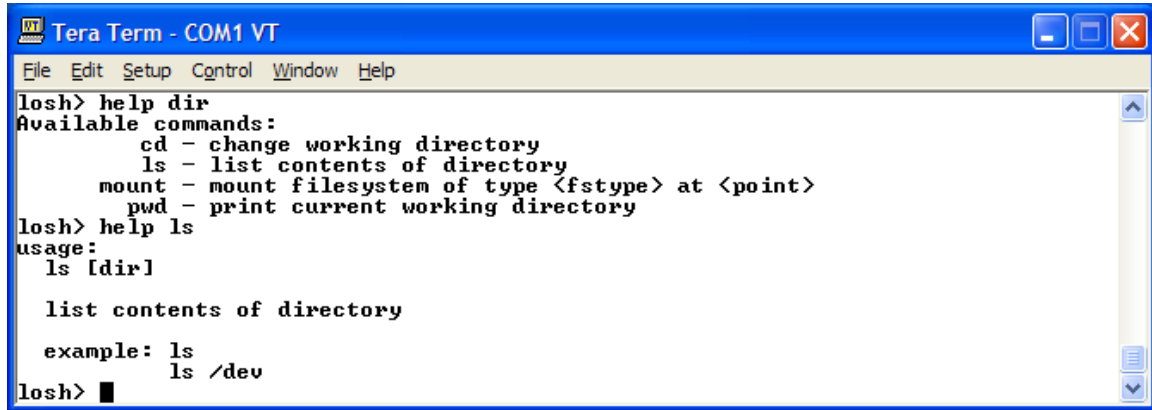
losh> something
stdin: line 1,9: something: command not found
losh> help
usage:
  help <test|file|dir|video|net|thread|misc|all|cmd_name>

  print help for a command or group of commands

  example: help dir
losh> help dir
Available commands:
  cd - change working directory
  ls - list contents of directory
  mount - mount filesystem of type <fstype> on device <device> at <mpoint>
  pwd - print current working directory
losh> █
```

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.

### 3.7 Type the words “help ls” at the “losh>” prompt and press the “Enter” key



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> help dir
Available commands:
    cd - change working directory
    ls - list contents of directory
    mount - mount filesystem of type <fstype> at <point>
    pwd - print current working directory
losh> help ls
usage:
  ls [dir]

  list contents of directory

  example: ls
           ls /dev
losh> █
```

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.

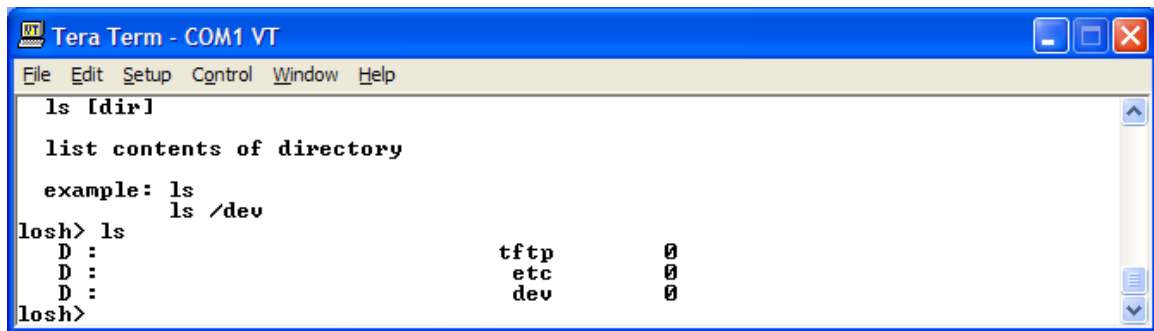
Before we continue, let’s take a closer look at the “ls” command’s help-topic. It looks a little different than what we saw before. Notice the command’s usage string, reprinted below.

```
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.

### 3.8 Type the letters “ls” at the “losh>” prompt and press the “Enter” key



```
Tera Term - COM1 VT
File Edit Setup Control Window Help

ls [dir]

list contents of directory

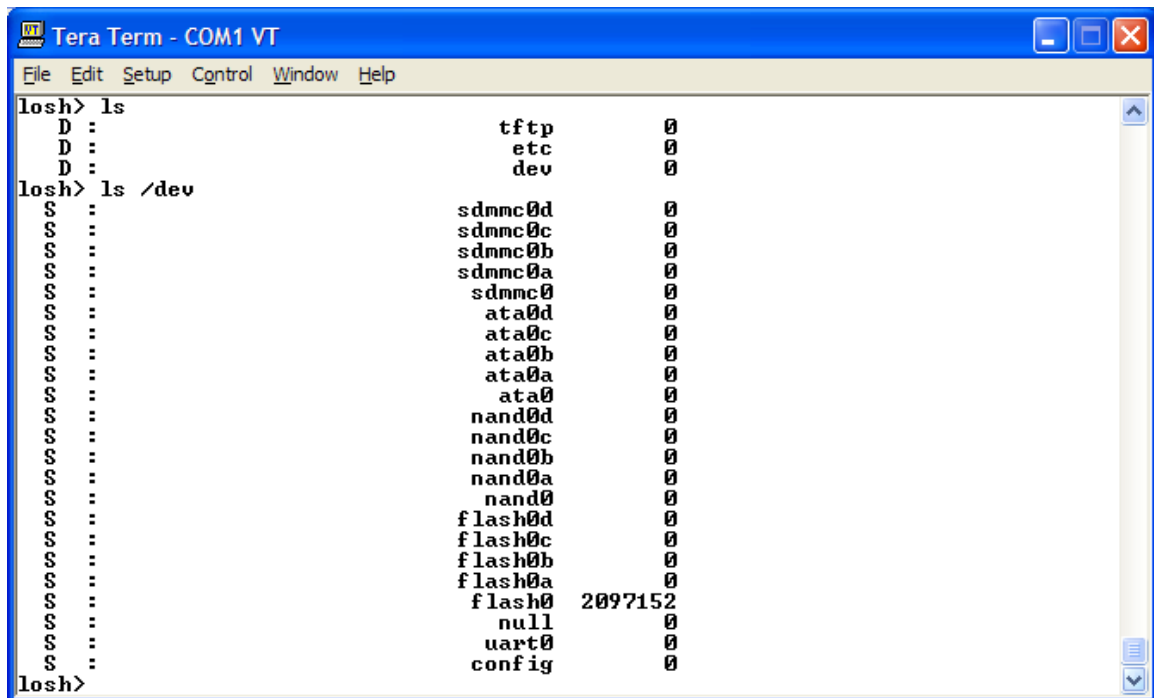
example: ls
          ls /dev

losh> ls
D :      tftp      0
D :      etc       0
D :      dev       0

losh>
```

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

### 3.9 Type the words “ls /dev” at the “losh>” prompt and press the “Enter” key



```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> ls
D :      tftp      0
D :      etc       0
D :      dev       0

losh> ls /dev
S :      sdmmc0d    0
S :      sdmmc0c    0
S :      sdmmc0b    0
S :      sdmmc0a    0
S :      sdmmc0     0
S :      ata0d      0
S :      ata0c      0
S :      ata0b      0
S :      ata0a      0
S :      ata0       0
S :      nand0d     0
S :      nand0c     0
S :      nand0b     0
S :      nand0a     0
S :      nand0      0
S :      flash0d    0
S :      flash0c    0
S :      flash0b    0
S :      flash0a    0
S :      flash0     2097152
S :      null       0
S :      uart0      0
S :      config     0

losh>
```

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.

### 3.10 Type the words “help all” at the “losh>” prompt and press the “Enter” key

```
losh> help all
Available commands:
add-yaffs - Deprecated. Supported in 2.4.x for backward compatibility
bitmap - draws a bitmap on the screen
bootme - starts a 'BOOTME' transfer with Platform Builder
burn - burns the already-loaded image into flash device [device]
cache-off - stop the processor's cache
cache-on - start the processor's cache
cat - print contents of a file to stdout
cd - change working directory
config - saves/displays configuration info in flash
cp - copies file <src> to new file <dest>
cpu-freq - sets the cpu frequency and voltage
date - display the number of seconds since boot
dd - copies from <if> to <of> <count> number of blocks
draw-test - draws framed red, green, blue and stipple test patterns
echo - echo a string to stdout or write to [file]
erase - erase a section of non-volatile memory
exec - disable cache & ints, then jump to a loaded OS, or to [addr]
help - print help for a command or group of commands
hd - print a file to stdout in hex format
ifconfig - configure a network interface
ifmac - program/display the MAC address for a network interface
info - print information from chosen category
jump - jump to a loaded image, or to [addr]
kill - stop thread <thread_id> from executing
load - download a binary image
ls - list contents of directory
md5sum - calculate an MD5 hash on a file or memory chunk
mem-cmp - compare memory from <a> to <b> for <len>
mem-copy - copy memory <src> to <dst> for <count> in sizes of [bhw]
mem-fill - fill memory at <addr> for <count> with <value> in sizes of [bhw]
mount - mount filesystem of type <fstype> on device <device> at <mpoint>
part-add - create partition <a-d> on <device> from <start_blk> with <length>
part-write - Writes the partition table on the <device>
part-rem - Removes the partition <entry> <device>
ping - ping remote host at <ip-address> [reps] times
ps - display the list of currently executing threads
pwd - print current working directory
reset - resets the processor
set - print shell variables or set shell options
setvar - sets <varname> to value or incr/decr
rm - removes a file or files
sleep - sleep for <ms> number of milliseconds
slide-show - displays a slide show of bitmaps on the screen
source - execute a series of losh commands from <file>
tsleep - thread sleep for <ms> number of milliseconds
unset - destroys a variable
update - load and install an update image for this Logic product
video-clear - clear the default video screen
video-close - turns off and disconnects the default video device
video-fb - sets or displays the video frame buffer address
video-init - connect default video device but do not enable the controller
video-off - turns off an initialized display
video-on - turns on an initialized display
video-open - open the default video device
w - write memory [of specified width, w is default] at addr
x - examine memory with [width][format] at an addr for a [len]
losh>
```

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.

## 4 Quiz

1. What communications channel does LogicLoader use for standard input and standard output (stdin, stdout)?
  - a. SPI
  - b. Touch Screen and Video
  - c. Serial Port
  - d. CompactFlash
2. The load command's usage string reads "load <type> [source]", which argument is optional?
  - a. Load
  - b. Type
  - c. source
3. Which of the following are proper uses of the help command?
  - a. help video-open
  - b. help video
  - c. help net
  - d. all of the above
  - e. none of the above



## LogicLoader Lab #3

# Using the LogicLoader™ Command Line Editor

---

### 1 Abstract

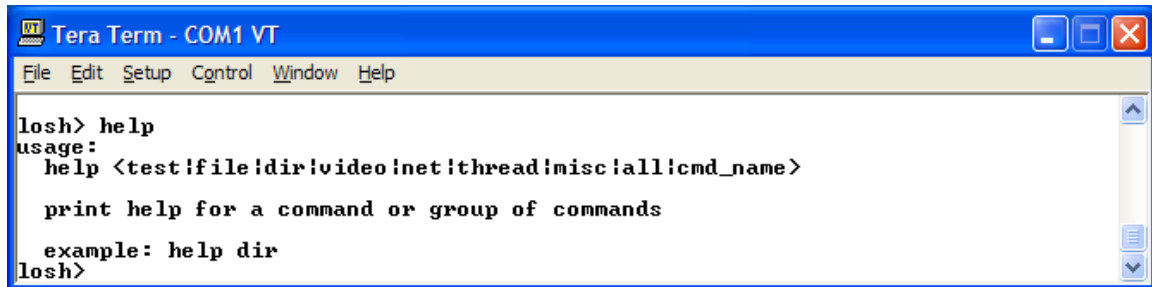
This lab will teach you how to work more efficiently with LogicLoader (LoLo) by learning its command line editing features.

### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- A development workstation (PC or laptop)
- A serial port communications connection between the workstation and the development kit

### 3 Instructions

#### 3.1 At the “losh>” prompt, type the “help” command



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

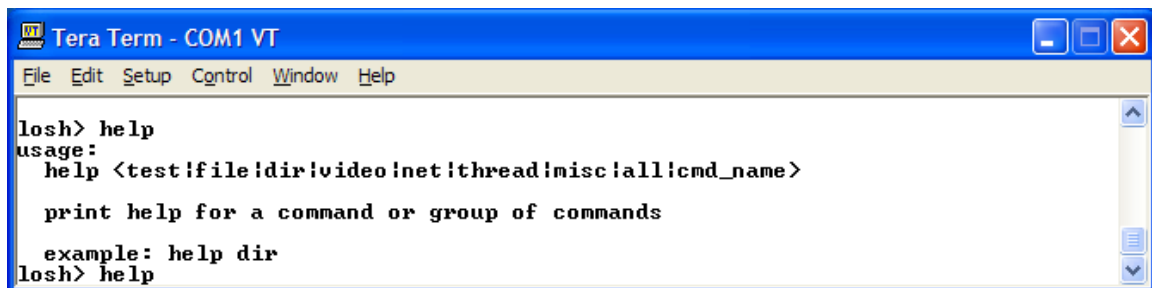
  print help for a command or group of commands

  example: help dir
losh>
```

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.

#### 3.2 At the “losh>” prompt, press the up-arrow key on your keyboard



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

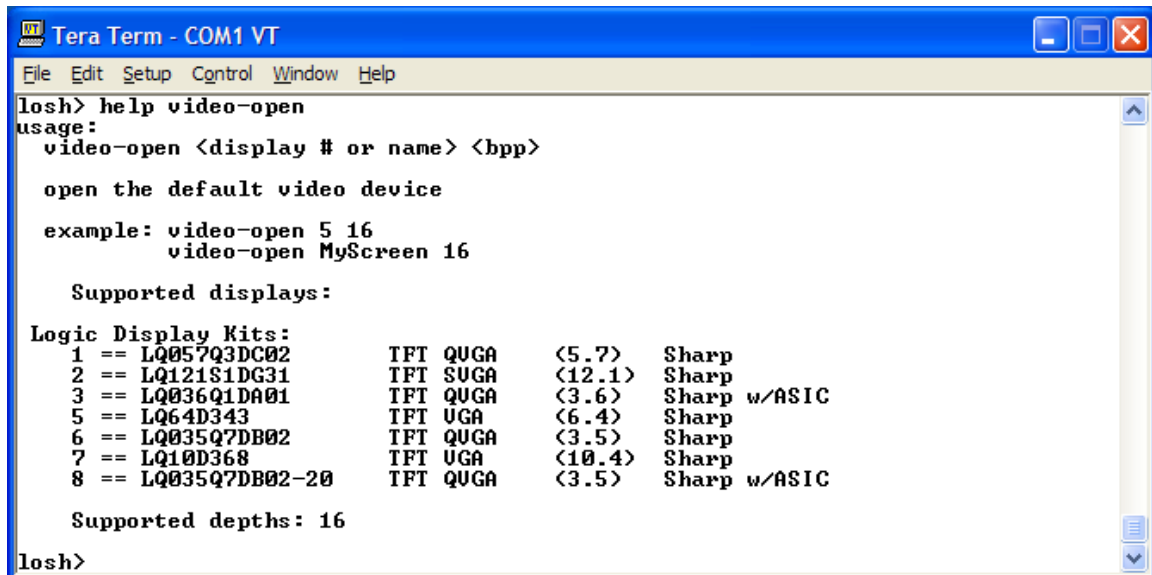
  print help for a command or group of commands

  example: help dir
losh> help
```

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.

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?

**3.3 At the “losh>” prompt type the command “help video-open” followed by the “Enter” key**



```
losh> help video-open
usage:
  video-open <display # or name> <bpp>

  open the default video device

example: video-open 5 16
         video-open MyScreen 16

Supported displays:

Logic Display Kits:
1 == LQ057Q3DC02      TFT QUGA    <5.7>   Sharp
2 == LQ121S1DG31      TFT SUGA    <12.1>  Sharp
3 == LQ036Q1DA01      TFT QUGA    <3.6>   Sharp w/ASIC
5 == LQ64D343         TFT UGA     <6.4>   Sharp
6 == LQ035Q7DB02      TFT QUGA    <3.5>   Sharp
7 == LQ10D368         TFT UGA     <10.4>  Sharp
8 == LQ035Q7DB02-20   TFT QUGA    <3.5>   Sharp w/ASIC

Supported depths: 16

losh>
```

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 1, 2, 3, 5, 6, 7, and 8. We can also see that a display depth of 16-bits per pixel is supported.

**3.4 At the “losh>” prompt type the command “video-open 0 16” followed by the “Enter” key**



```
losh> video-open 0 16
error: video-open: found no suitable display driver
error: video-open: found no suitable display driver
losh>
```

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 cursor 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 8) in place of the zero;
5. Press the “Enter” key.

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.

## 4 Quiz

1. Repeating the last command entered requires how many key presses?
  - a. One
  - b. Two
  
2. How many recent commands does LoLo store?
  - a. Five
  - b. Ten
  - c. Fifteen
  - d. Twenty
  
3. Which character combination moves the cursor to the beginning of a line?
  - a. Control-a
  - b. Control-b
  - c. Control-c
  - d. Control-d

This page intentionally left blank.



## LogicLoader Lab #4

### Upgrading LogicLoader™

---

#### 1 Abstract

This lab will teach you how to download a new version of the LogicLoader (LoLo) to your Zoom i.MX31 LITEKIT using either the 'update' command or the 'load elf' command.

#### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- 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.

#### 3 Basic instructions

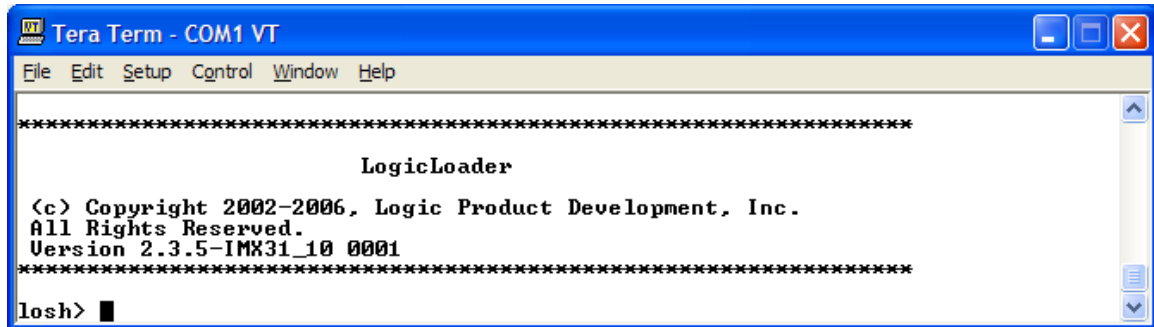
Using the same process you learned in the "Kit Communications" lab, open your terminal emulator and set the port settings to:

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

## 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

The image shows a screenshot of a Tera Term window titled "Tera Term - COM1 VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The main text area displays a welcome message for LogicLoader, enclosed in a box of asterisks. The message reads: "LogicLoader", "<c> Copyright 2002-2006, Logic Product Development, Inc.", "All Rights Reserved.", "Version 2.3.5-IMX31\_10 0001". Below the message, the prompt "losh>" is visible with a cursor.

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

## 5 Obtain a new version of LogicLoader

The first step is to access Logic's download site at <http://www.logicpd.com/auth/> and enter your username and password. This will bring you to your Account Summary page. Click on your product. 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 have to determine which method you can use to update LoLo on your development kit—using the ‘update’ command or using the ‘load elf’ command. Some versions of LogicLoader do not support the ‘update’ command and require using the ‘load elf’ command and the .elf file format of the LoLo download.

To determine which method is right for you, locate the ZIP file containing the latest version of LogicLoader that you just downloaded. Within that file is a “Release\_Notes” text file that will explain which versions require using the ‘load elf’ command. However, it is beneficial for you to know both of these methods, so make sure you read through both of the following sections even though you only need to perform the steps for the method that applies to you.

## 6 Download LogicLoader using the ‘update’ command

### 6.1 Prerequisites

- A recent version of Internet Explorer or Firefox and an internet connection
- TeraTerm installed and running
- Zoom i.MX31 LITEKIT up and running
- Zoom i.MX31 LITEKIT connected to development computer via null modem serial cable

## 6.2 Procedure

1. Now that you have downloaded the version of LoLo that you want, you can open your terminal program and power up your Kit.
2. Type **update**. Your screen should look similar to the figure below.



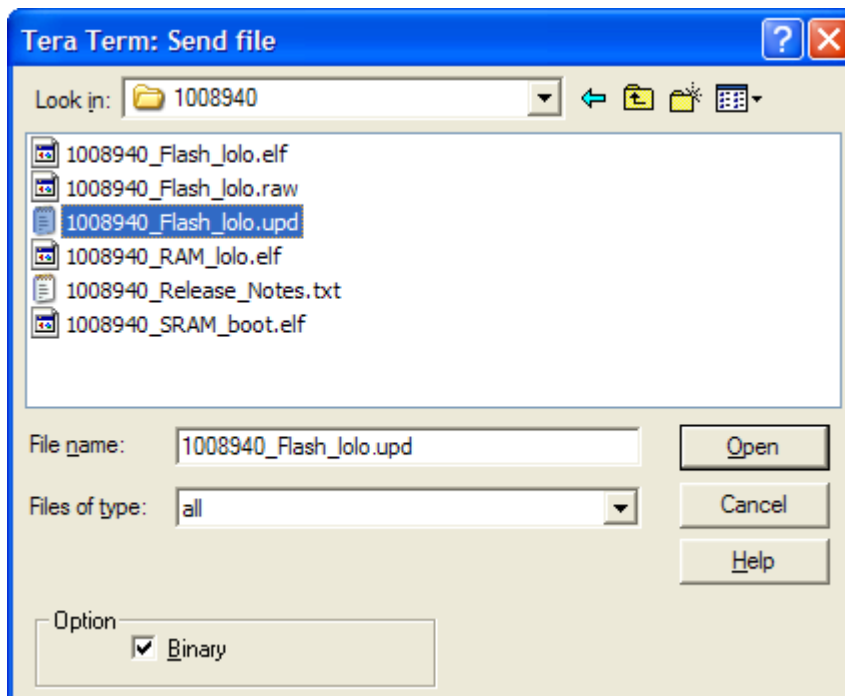
The screenshot shows a terminal window titled "Tera Term - COM1 VT". The menu bar includes File, Edit, Setup, Control, Window, and Help. The terminal content is as follows:

```
*****
                        LogicLoader

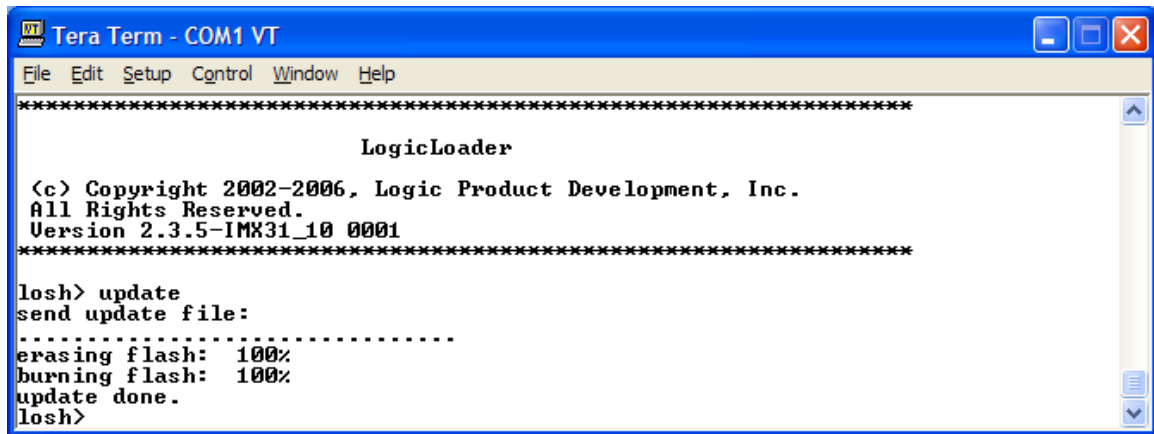
<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.3.5-IMX31_10 0001
*****

losh> update
```

3. LoLo is waiting for you to send the new version over the serial port. Click 'File' and select 'Send file' to send the new LogicLoader to your kit.
4. Locate and select the lolo.upd file you downloaded to your hard drive. Also, verify that the 'Binary' option is selected. Then click 'Open' to download the new LogicLoader.



5. LoLo will erase flash and update you on the progress. Then it will burn the new LoLo into flash. Verify that the download was successful by waiting for an update done message to appear.

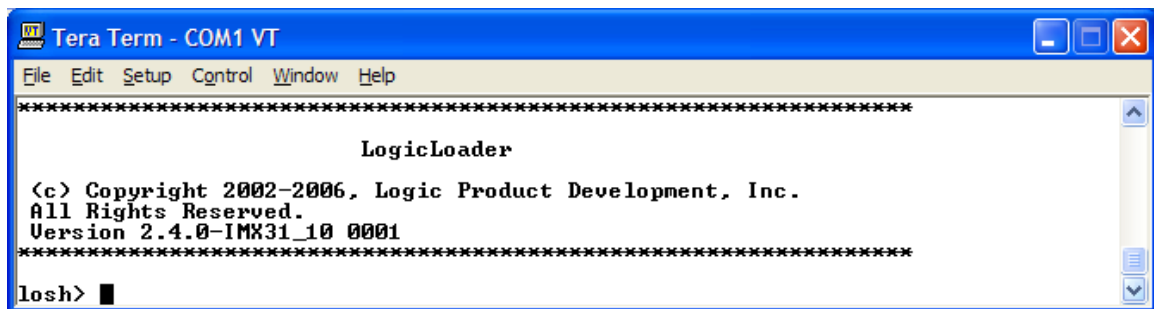


```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.3.5-IMX31_10 0001
*****

losh> update
send update file:
.....
erasing flash: 100%
burning flash: 100%
update done.
losh>
```

6. Now you can reset the Kit and verify the new LogicLoader version number.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> █
```

7. If you get this message—Nice work! You now have the latest and greatest version of LogicLoader installed on your kit.

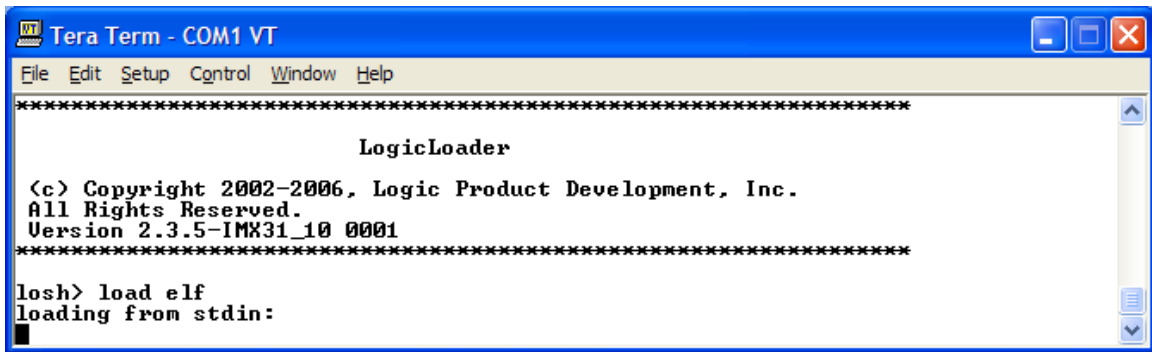
If you don't have a copy of the lolu.upd file but you have flash\_lolu.elf, you can use the method in the next section to update your LoLo.

## 7 Download LogicLoader using the 'load elf' command

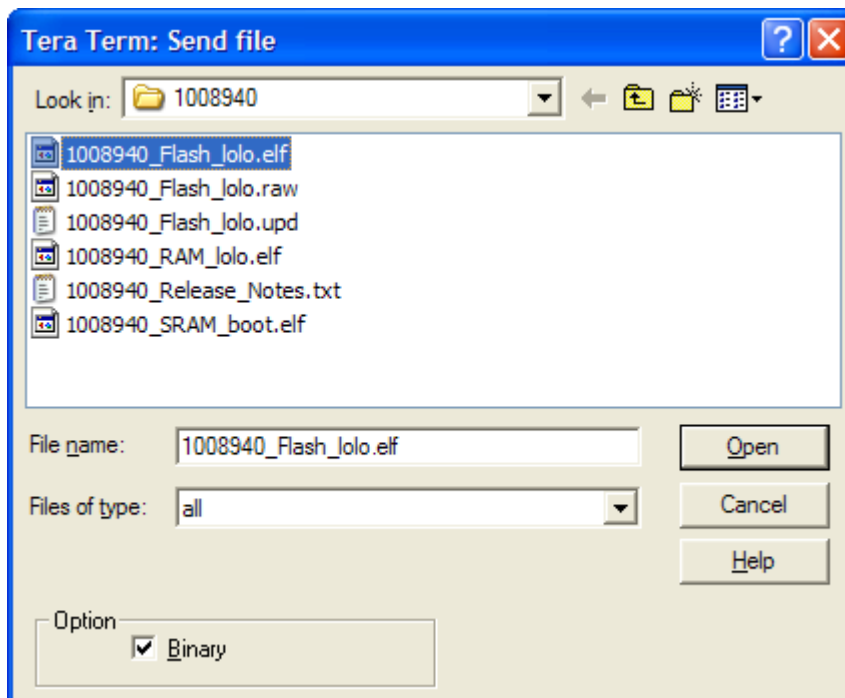
The same prerequisites apply for the 'load elf' command as for the 'update' command above.

### 7.1 Procedure

1. Now that you have downloaded the version of LoLo that you want, you can open your terminal program and power up your Kit.
2. Type **load elf**. You should see a screen that looks like the one shown below.



3. Once again LoLo is waiting for you to send the new version over the serial port. Click 'File' and select 'Send file' to send the new LogicLoader to your development kit.
4. Locate and select the lolo.elf file. Verify that the 'Binary' option is selected. Then click 'Open' to download the new LogicLoader.



5. This method returns a different screen than when using the 'update' command. You can verify that the download was successful by waiting for the file loaded to ram buffer message to appear.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.3.5-IMX31_10 0001
*****

losh> load elf
loading from stdin:
.....
ELF section 0: download address: 0x80200060 load address: 0xa0000000
loaded 261928 @ 0xa0000000 Flash
...done
file loaded to ram buffer, destined for flash addr: 0xa0000000
losh> █

```

6. When you use the 'load elf' command, LoLo puts the file into RAM. If you reset your kit right now, that file will be lost. To finish the update, you need to type **burn** and press the Enter key to burn the updated LoLo to flash.
7. You will be prompted to confirm that you are overwriting the current LogicLoader. Type **y** and **confirm**, then press the Enter key to begin the burn process.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
file loaded to ram buffer, destined for flash addr: 0x0
losh> burn

*****
                        Warning
*****

You are about to overwrite flash block zero.

Flash block zero is the default boot sector for this device.
If programming fails for any reason (power loss, bad image, etc.)
your system may no longer boot.
Even if programming completes successfully, if the new image
does not boot properly your system may become useless.

Confirmation is required before continuing.

*****
*****
Do you wish to continue? (y/n)
Please type "confirm" to confirm: confirm█

```

8. You can verify the burn was successful by waiting for the burn good message to appear.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
*****
Warning

You are about to overwrite flash block zero.

Flash block zero is the default boot sector for this device.
If programming fails for any reason (power loss, bad image, etc.)
your system may no longer boot.
Even if programming completes successfully, if the new image
does not boot properly your system may become useless.

Confirmation is required before continuing.

*****
*****
Do you wish to continue? (y/n)
Please type "confirm" to confirm: confirm

burning 261928 bytes from 80200000 to /dev/flash0
erasing flash: 100%
burning flash: 100%
burn good (261928 bytes)
losh> █
```

9. Now you can reset your development kit and verify the new Logic Loader version number.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help

burning 261928 bytes from 80200000 to /dev/flash0
erasing flash: 100%
burning flash: 100%
burn good (261928 bytes)
losh> █
*****
*****
LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
*****
losh> █
```

## 8 Updating LoLo from another source

### 8.1 Procedure

The load command can also use a filename. If you have an update file on a CompactFlash card for example, you can type **update /cf/filename.upd**

LoLo will find the file and begin the download process. The rest of the procedure works exactly the same. Wait for the update done message to appear and it is safe to reboot your kit.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.3.5-IMX31_10 0001
*****

losh> mount fatfs /cf
timeout resetting cf card
error: mount: failed to open (ATA)
losh> mount fatfs /cf
losh> cd /cf
losh> ls
r:                1008940_Flash_lolo.elf    255848
r:                1008940_Flash_lolo.upd    255632
r:                SPLASH.BMP               230454
r:                SEGGER.SREC              1205082
r:                AUTOEXEC.BAT             137
losh>

```

You can also use the load elf command with a CompactFlash card. Type **load elf /cf/filename.elf** then after the file loaded... message, type **burn**.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> load elf /cf/1007274_lolo.elf
loading from /cf/1007274_lolo.elf:
.....
ELF section 1: download address: 0xc0200080 load address: 0x00000000
loaded 244952 @ 0x00000000 Flash
file loaded to ram buffer, destined for flash addr: 0x0
losh> burn

*****
                        Warning
*****

You are about to overwrite flash block zero.

Flash block zero is the default boot sector for this device.
If programming fails for any reason (power loss, bad image, etc.)
your system may no longer boot.
Even if programming completes successfully, if the new image
does not boot properly your system may become useless.

Confirmation is required before continuing.

*****
Do you wish to continue? (y/n)
Please type "confirm" to confirm: confirm

burning 244952 bytes from c0200080 to /dev/flash0
erasing flash: 100%
burning flash: 100%
burn good (244952 bytes)
losh>

```

## 9 Quiz

1. There is only one way to load a new version of LogicLoader onto your kit?
  - a. True
  - b. False
2. When you use the 'update' command, you need to choose which file to send?
  - a. lolo.elf
  - b. lolo.doc
  - c. lolo.exe
  - d. lolo.upd
3. The 'load elf' command loads your file to what memory?
  - a. RAM
  - b. Flash
  - c. EEPROM
4. You can only update via the Serial Port
  - a. True
  - b. False

This page intentionally left blank.



## LogicLoader Lab #5

### Network Connection

---

#### 1 Abstract

This lab will teach you how to connect your workstation to the Zoom i.MX31 LITEKIT via Ethernet.

#### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- 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.

#### 3 Basic instructions

Using the same process you learned in the “Kit Communication” lab, 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

## 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

The image shows a screenshot of a Tera Term window titled "Tera Term - COM1 VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The main text area displays a welcome message for "LogicLoader" enclosed in asterisks. The message includes copyright information for Logic Product Development, Inc. (2002-2006), version 2.4.0-IMX31\_10 0001, and a prompt "losh>".

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
losh> █
```

## 5 Connect the Kit to Ethernet

### 5.1 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.

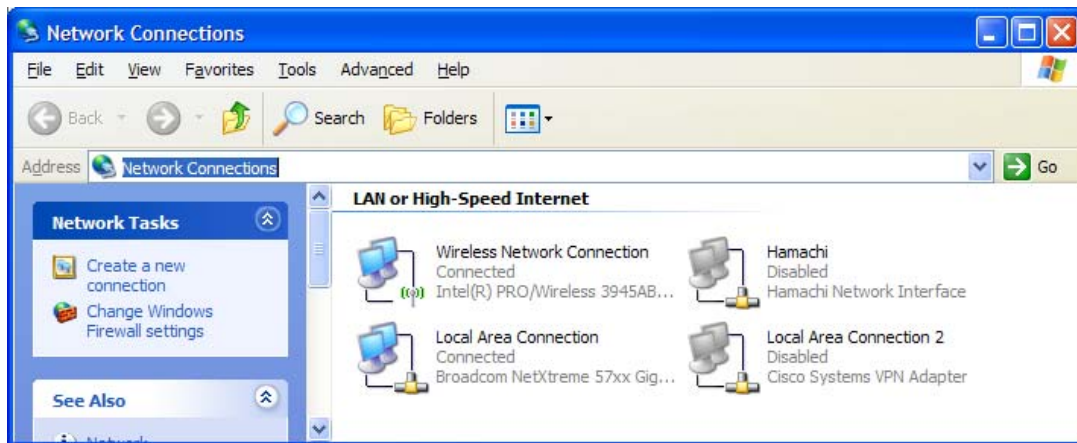
### 5.2 Setup network information on your kit for use with the cross-over Ethernet cable

If you are using the 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.

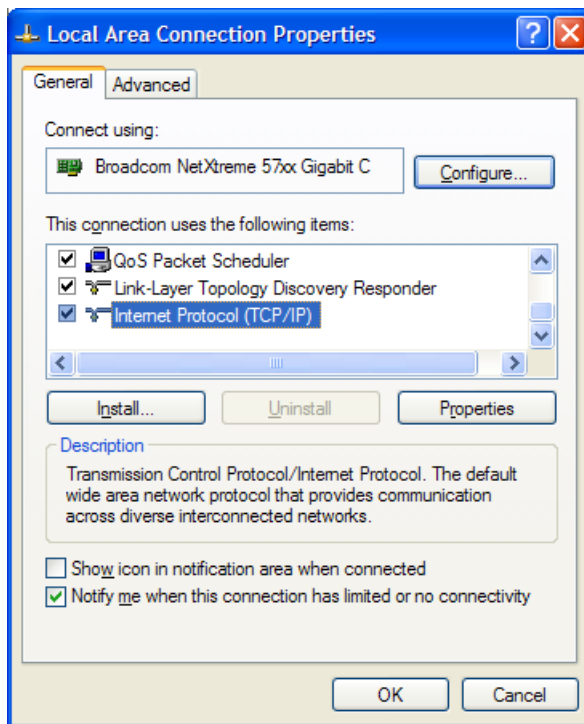
The IP address of the kit must match the IP address of your workstation for as many 255 number sets as there are in the subnet mask. For example, if your subnet mask is 255.255.0.0 then the first two number sets of the IP address must be the same on both devices.

If you would like to set your IP address you can do so by going to

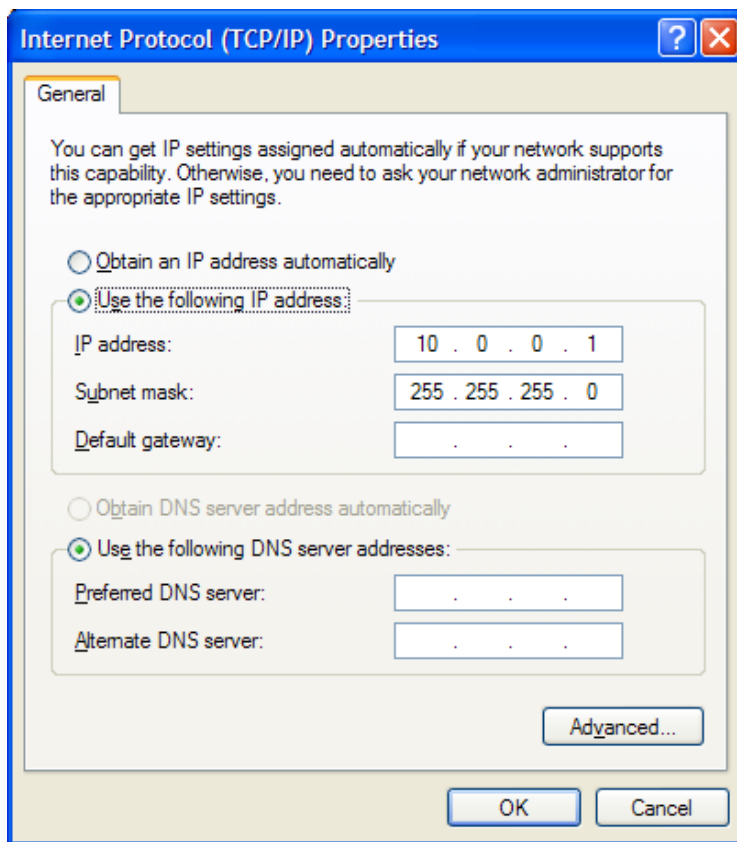
Start -> Network Connections -> Local Area Connection.



Double click on the wired connection and click on Properties. Scroll down to Internet Protocol (TCP/IP).



Click on Properties.



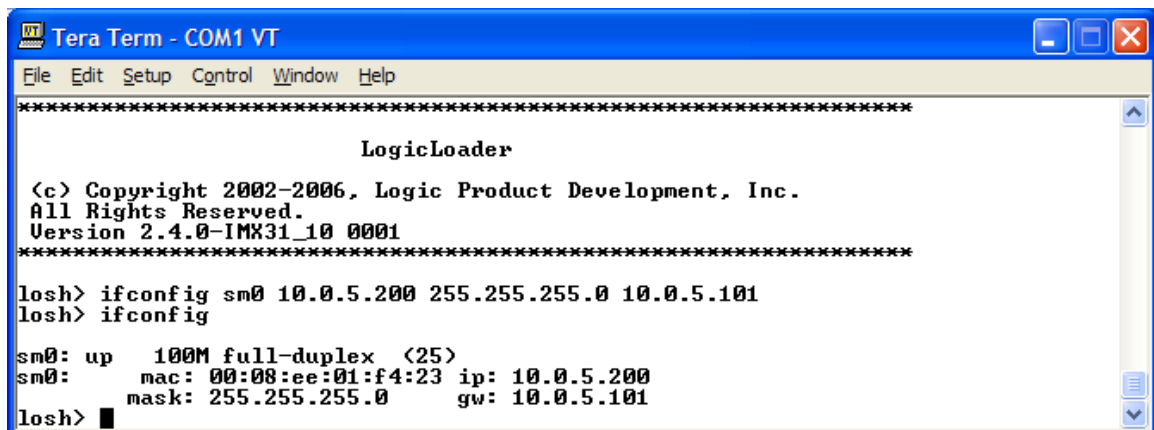
Set your IP address as desired and Click OK.

Once you have the IP address and subnet mask of your workstation you can use

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

to set the networking parameters on the kit.

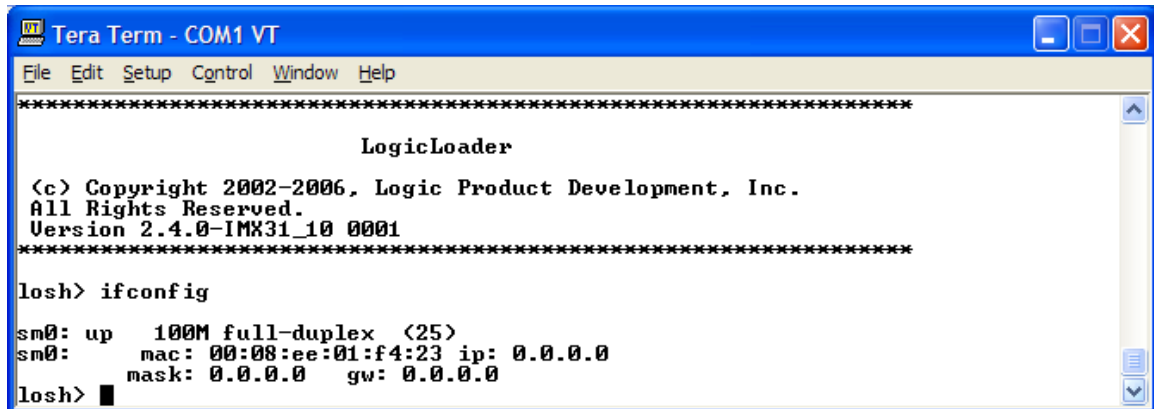
The figure below shows an example of the full 'ifconfig' command. Remember that your numbers will be different.



You now have setup your networking parameters and can skip to section 6.

## 5.3 Setup network information on your kit for use with a standard Ethernet cable

If you have a network connection with DHCP you can use it to quickly set your network parameters. From the `losh>` prompt, type **ifconfig**.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

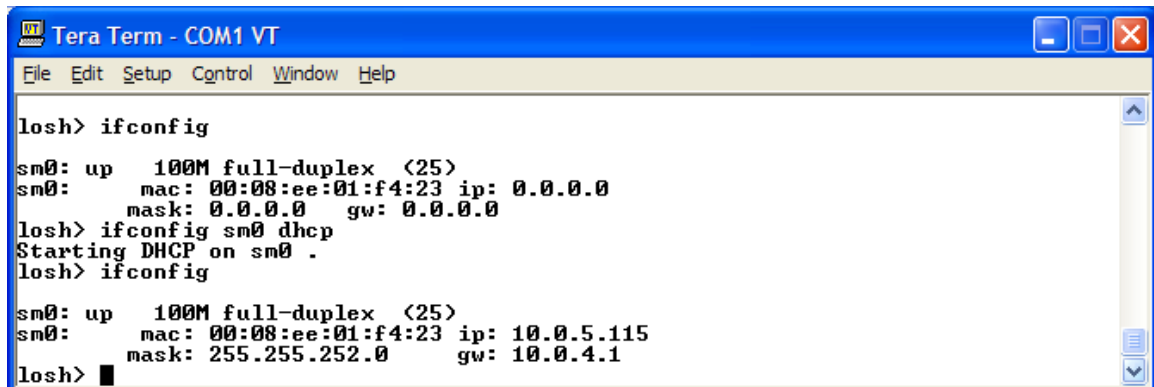
losh> ifconfig

sm0: up    100M full-duplex <25>
sm0:      mac: 00:08:ee:01:f4:23 ip: 0.0.0.0
        mask: 0.0.0.0    gw: 0.0.0.0
losh>
```

You can see the interface information for sm0. The MAC address is 00:08:ee:01:f4:23. The IP address, mask and gateway have not been set.

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.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> ifconfig

sm0: up    100M full-duplex <25>
sm0:      mac: 00:08:ee:01:f4:23 ip: 0.0.0.0
        mask: 0.0.0.0    gw: 0.0.0.0
losh> ifconfig sm0 dhcp
Starting DHCP on sm0 .
losh> ifconfig

sm0: up    100M full-duplex <25>
sm0:      mac: 00:08:ee:01:f4:23 ip: 10.0.5.115
        mask: 255.255.252.0    gw: 10.0.4.1
losh>
```

## 6 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.

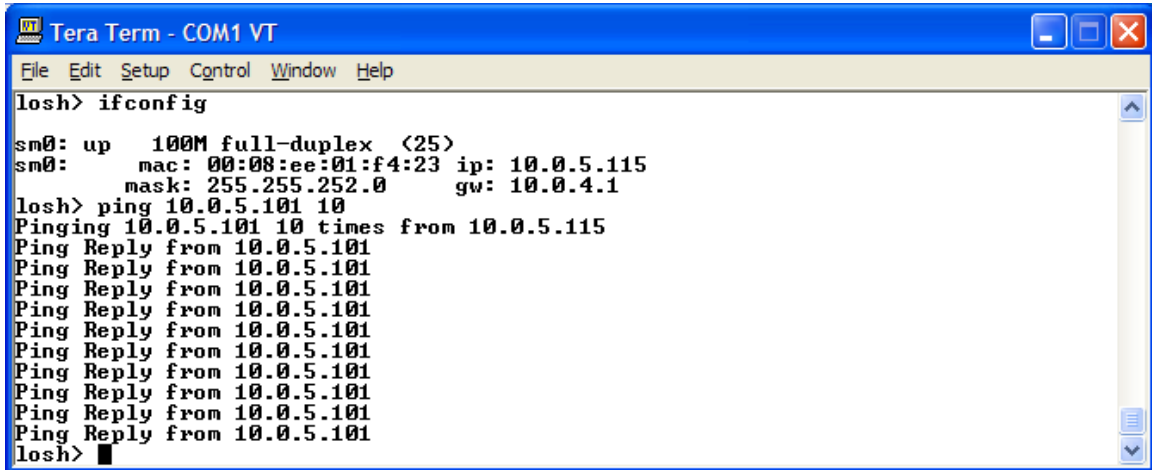
### 6.1 Use the 'ping' command to check your connection

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. An example is shown below. Remember your IP address will be different than that in the example below.

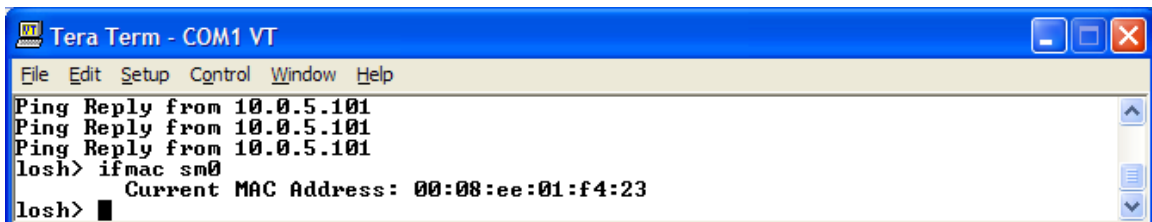


```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> ifconfig
sm0: up 100M full-duplex (25)
sm0: mac: 00:08:ee:01:f4:23 ip: 10.0.5.115
      mask: 255.255.252.0 gw: 10.0.4.1
losh> ping 10.0.5.101 10
Pinging 10.0.5.101 10 times from 10.0.5.115
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
losh>
```

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.

## 7 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.

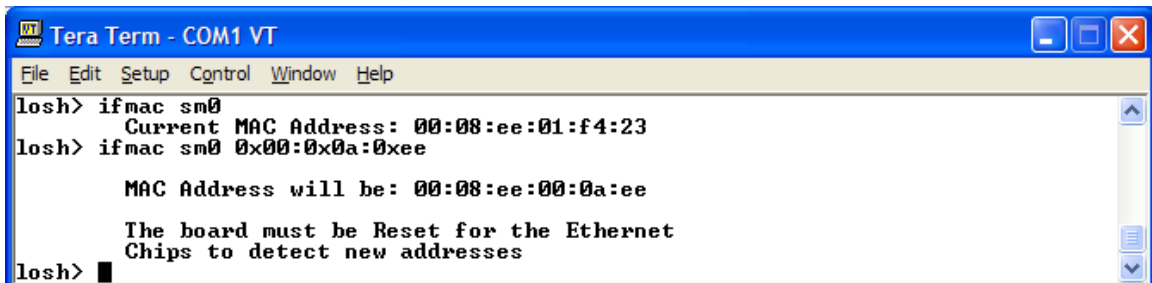


```
Tera Term - COM1 VT
File Edit Setup Control Window Help
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
Ping Reply from 10.0.5.101
losh> ifmac sm0
      Current MAC Address: 00:08:ee:01:f4:23
losh>
```

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.

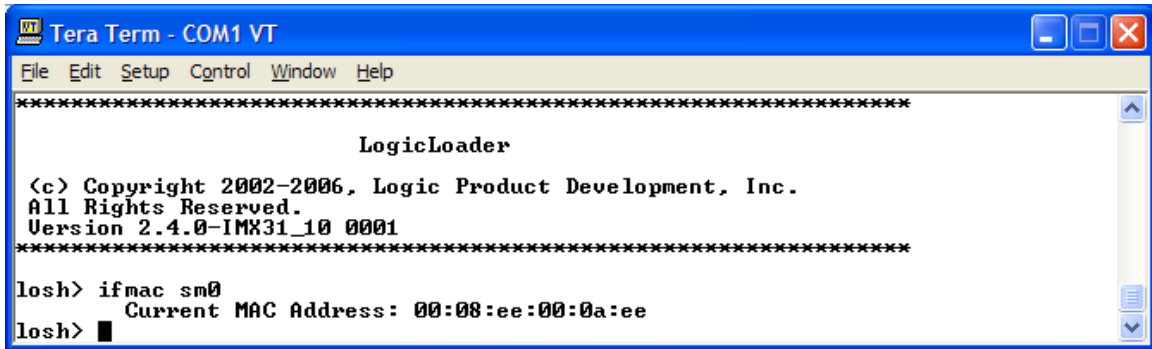
Type **ifmac sm0 0x00:0x0a:0xee**

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



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> ifmac sm0
      Current MAC Address: 00:08:ee:01:f4:23
losh> ifmac sm0 0x00:0x0a:0xee
      MAC Address will be: 00:08:ee:00:0a:ee
      The board must be Reset for the Ethernet
      Chips to detect new addresses
losh>
```

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



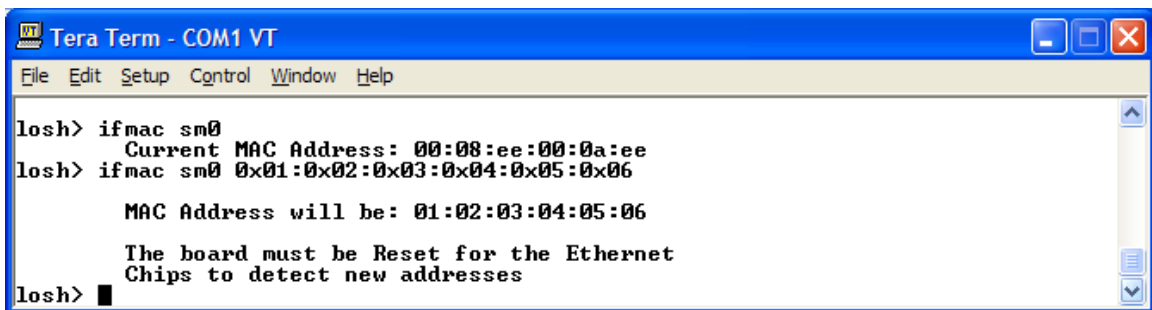
```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> ifmac sm0
      Current MAC Address: 00:08:ee:00:0a:ee
losh> █
```

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

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



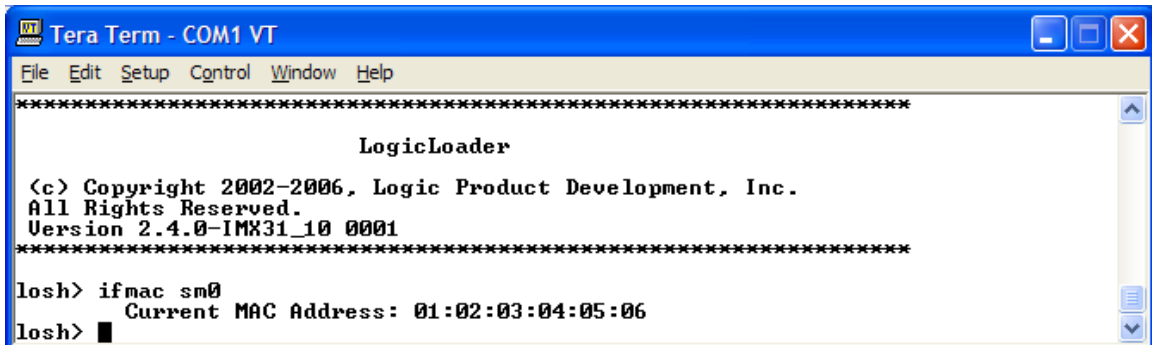
```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> ifmac sm0
      Current MAC Address: 00:08:ee:00:0a:ee
losh> ifmac sm0 0x01:0x02:0x03:0x04:0x05:0x06

      MAC Address will be: 01:02:03:04:05:06

      The board must be Reset for the Ethernet
      Chips to detect new addresses
losh> █
```

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



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> ifmac sm0
      Current MAC Address: 01:02:03:04:05:06
losh> █
```

## 8 Quiz

1. If the subnet mask on your kit is 255.255.255.0 how many number sets must match for the IP address on the kit and the target workstation?
  - a. One number set
  - b. Two number sets
  - c. Three number sets
  - d. All number sets
  
2. If you only send 3 bytes with the 'ifmac' command, what will the first 3 bytes of the new MAC address be?
  - a. 00 01 0a
  - b. ee 00 01
  - c. 00 08 ee
  - d. 01 02 03
  
3. Which of the following sets of networking information will work together?
  - a. 192.168.1.101 255.255.0.0 192.168.105.158
  - b. 192.168.1.101 255.255.255.0 192.168.1.102
  - c. 192.168.1.101 255.0.0.0 192.167.2.102
  - d. All of the above



## LogicLoader Lab #6

### File Systems

---

#### 1 Abstract

This lab will teach you how to utilize file systems on the Zoom i.MX31 LITEKIT.

#### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- A development workstation (PC or laptop)
- An available serial port or USB to serial adapter
- A CompactFlash card
- A terminal emulation program such as HyperTerminal or TeraTerm. **Note:** Logic recommends using TeraTerm, a free program downloadable from the Internet.

#### 3 Basic instructions

Using the same process you learned in the “Kit Communications” lab, open your terminal emulator and set the port settings to:

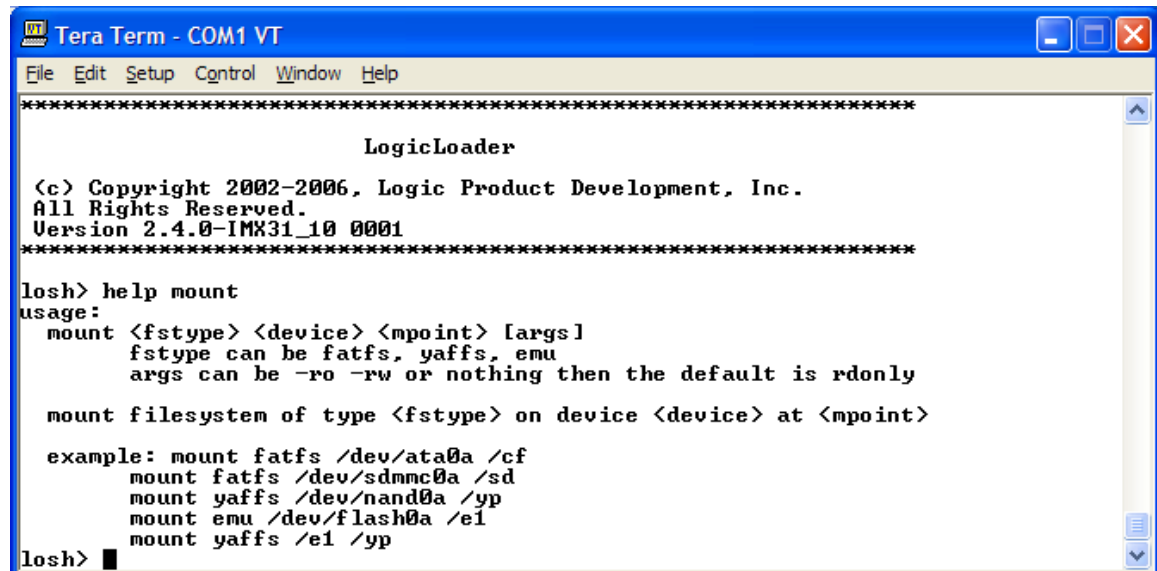
- 115200 BAUD rate
- 8 data-bits
- No parity
- No hardware flow control
- No software flow control

## 4 FATFS (Fat File System)

### 4.1 Mount a CompactFlash Card

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

1. Store a file on a CompactFlash card.
2. Insert the CompactFlash card into the Zoom Development Kit's CompactFlash 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**.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> help mount
usage:
  mount <fstype> <device> <mpoint> [args]
      fstype can be fatfs, yaffs, emu
      args can be -ro -rw or nothing then the default is rdonly

  mount filesystem of type <fstype> on device <device> at <mpoint>

example: mount fatfs /dev/ata0a /cf
         mount fatfs /dev/sdmmc0a /sd
         mount yaffs /dev/nand0a /yp
         mount emu /dev/flash0a /e1
         mount yaffs /e1 /yp

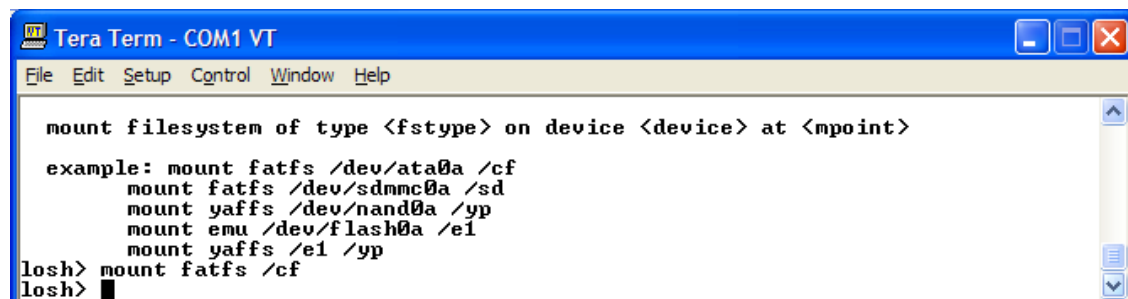
losh>
```

If you are familiar with past versions of LoLo you'll see some changes here.

In this version of LoLo the legacy mount fatfs /cf is still supported, but we now have added SD support as well, so the new syntax is a little longer, but makes it clear if you're mounting CF or SD.

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. Note that the examples list CF, SD, Yaffs and emu. We will just deal with CF in this lab, but SD works the same way. Just remember to use /dev/sdmmc0a /sd.

6. Let's do that. Type **mount fatfs /cf**. The screen displayed should be similar to the one shown below.



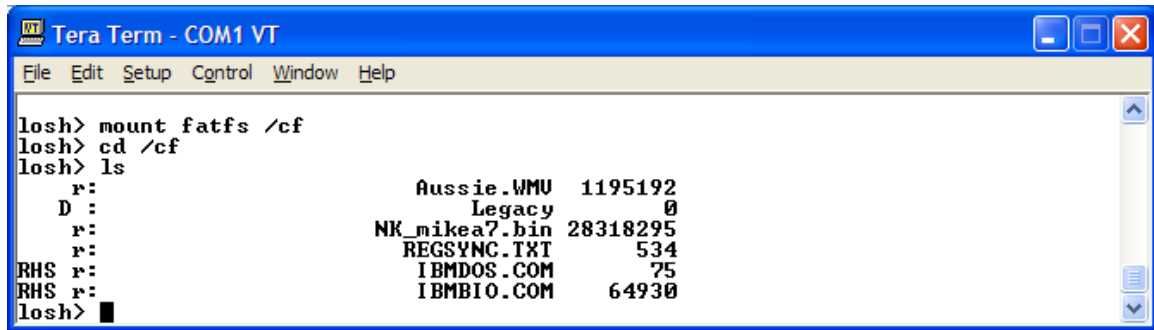
```
Tera Term - COM1 VT
File Edit Setup Control Window Help

  mount filesystem of type <fstype> on device <device> at <mpoint>

example: mount fatfs /dev/ata0a /cf
         mount fatfs /dev/sdmmc0a /sd
         mount yaffs /dev/nand0a /yp
         mount emu /dev/flash0a /e1
         mount yaffs /e1 /yp

losh> mount fatfs /cf
losh>
```

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



```
losh> mount fatfs /cf
losh> cd /cf
losh> ls
r:      Aussie.WMU    1195192
D :      Legacy      0
r:      NK_mikea?.bin 28318295
r:      REGSYNC.TXT   534
RHS r:   IBMDOS.COM    75
RHS r:   IBMBIO.COM    64930
losh>
```

8. You now have access to the files on the CompactFlash card. If you don't want to use a CompactFlash card, you will need to store the file directly on the kit. We will cover one method for doing that in the next section.

## 5 YAFFS (Yet Another Flash File System)

### 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.

### 5.2 Working with YAFFS in LogicLoader

#### 5.2.1 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.
- A partition (in NOR flash) has to be aligned on and exactly span the largest sector boundary found within the address range specified.

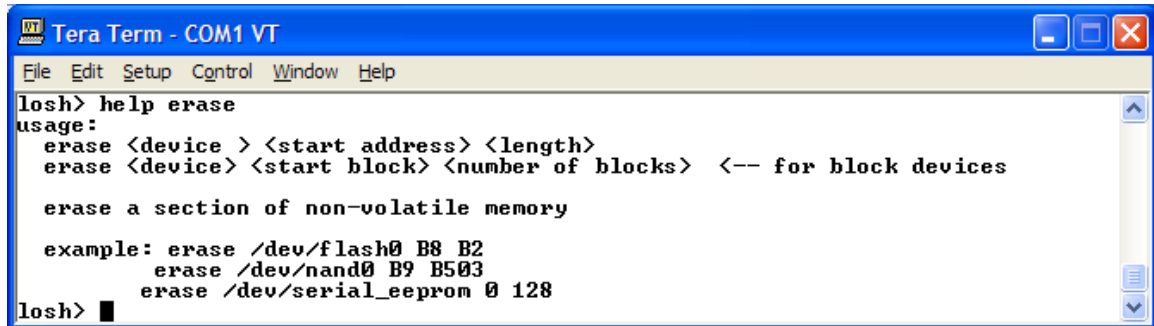
#### 5.2.2 Formatting YAFFS Partitions

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.

You can prepare the partitions for initial use by erasing the regions of the flash device spanned by them. We will cover both NOR and NAND partitions. NOR devices use hex addressing while NAND can use block addresses. Here are two examples for preparing flash for partitioning.

### A NOR example:

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



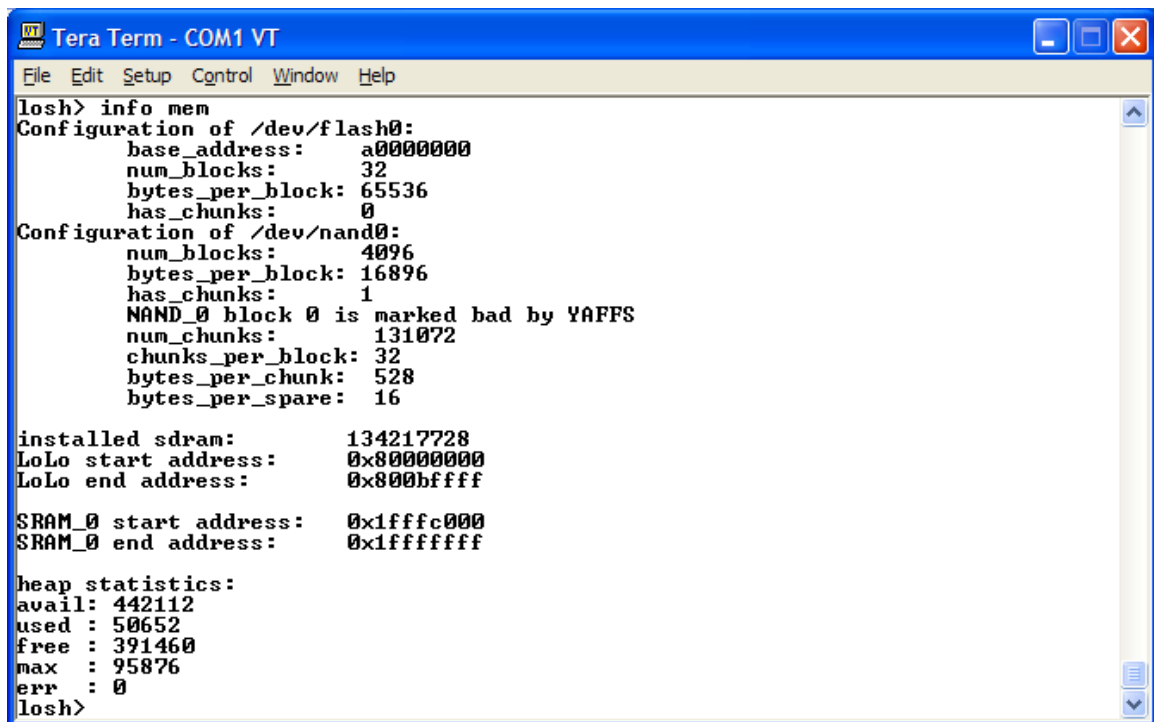
```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> help erase
usage:
  erase <device> <start address> <length>
  erase <device> <start block> <number of blocks>  -- for block devices

  erase a section of non-volatile memory

example: erase /dev/flash0 B8 B2
         erase /dev/nand0 B9 B503
         erase /dev/serial_eeprom 0 128
losh>
```

Help gives us two options that we can use for our examples. The first is a NOR example and the second is a NAND example. The third example is for the serial eeprom, which we will use when we get to scripting in another lab.

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**.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> info mem
Configuration of /dev/flash0:
  base_address:  a0000000
  num_blocks:    32
  bytes_per_block: 65536
  has_chunks:    0
Configuration of /dev/nand0:
  num_blocks:    4096
  bytes_per_block: 16896
  has_chunks:    1
  NAND_0 block 0 is marked bad by YAFFS
  num_chunks:    131072
  chunks_per_block: 32
  bytes_per_chunk: 528
  bytes_per_spare: 16

installed sdram:      134217728
LoLo start address:   0x80000000
LoLo end address:     0x800bffff

SRAM_0 start address: 0x1fffc000
SRAM_0 end address:   0x1fffffff

heap statistics:
avail: 442112
used : 50652
free : 391460
max  : 95876
err  : 0
losh>
```

On the i.MX31 LogicLoader is stored in NOR flash. It is currently limited to 256KB so we could erase from 0xa0040000 (B5) to the end and we shouldn't be overwriting LogicLoader. Even so, we have all of the NAND available, so let's leave the NOR alone and just use the NAND.

### A NAND example:

NAND can be erased using blocks instead of hex addresses. Notice that all of the nand0 options are in blocks. If there are bad blocks marked by the manufacturer, you might need to adjust your partition location. For this NAND example, type **erase B1 B4095 /dev/nand0** and you will get a screen similar to the one shown below.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> erase /dev/nand0 B1 B4095
erasing nand: 100%
erased '/dev/nand0' start=0x1: len=0xff bytes/blocks skipped 0
losh>

```

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. If you make a mistake, LogicLoader will try to protect itself by asking if you really want to do what you have told it to do; just type **n** to stop the erase command.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> erase /dev/flash0 B1 B31
Nor flash erase: The area (or parts of it) 0x10000 - 0x1f0000 is locked.
The area you are trying to write to is locked

***** WARNING!! *****

You are about to erase locked areas.
They might be needed to boot the card engine. If you erase this block the
card engine may not boot. Are you sure you want to do this?

***** WARNING!! *****

Do you wish to continue? (y/n)

```

### 5.2.3 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 'add-yaffs' command is used. The 'add-yaffs' command takes the following arguments:

- <name> a unique string which identifies the partition
- <type> type of flash device the partition resides on
- <start> the physical starting address of the partition
- <length> the length (in bytes) of the partition

Continuing with the example partitions above, LogicLoader can be instructed to add the partitions by executing the commands as shown below.

Type **help add-yaffs**.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> help add-yaffs
usage:
  add-yaffs <name> nor <start address> <byte length>
  add-yaffs <name> nand <start block> <number of blocks>

  Deprecated. Supported in 2.4.x for backward compatibility

  example: add-yaffs boot nor 0xC0000 0x800000
           add-yaffs store nand B200 B800
losh> █
```

Uh, deprecated? Like the mount command, changes were made in LoLo 2.4.0 to the way we handle YAFFS. We added the part-add command. For those of you that know the old versions of LoLo you'll still be able to use existing scripts with add-yaffs the same as in previous versions, but we recommend using the new syntax with the part-add command. First we'll cover the legacy command then the new one.

Let's prepare the flash for yaffs and then add the partition.

Type **erase /dev/nand0 B1 B4095**

Then **add-yaffs YaffsPart1 nand B1 B4095**

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> erase /dev/nand0 B1 B4095
erasing nand: 100%
erased '/dev/nand0' start=0x1: len=0xfff bytes/blocks skipped 0
losh> add-yaffs YaffsPart1 nand B1 B4095
Created partition on /dev/nand0a
losh> █
```

You can name your Yaffs partition anything you want, WindowsCE will call the partition YaffsPart1, so that's what is used here. The add-yaffs step must be performed every time LogicLoader boots. Because LogicLoader keeps the partition table in RAM, the existence and locations of YAFFS partitions does not remain after resets or power cycles.

Now type help part-add.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> help part-add
usage:
  part-add <device> <a-d> <start_blk> <length(in blocks)>

  create partition <a-d> on <device> from <start_blk> with <length>

  example: part-add /dev/nand0 a 0 8
losh> █
```

This is a little cryptic, but the example is what we need to see to add a Yaffs partition.

The device we're adding the partition to is nand0, since this is the first partition, it will be a and the partition will start at block 0 and go to block 8. Let's use the same location and size as the first example. Type **part-add /dev/nand0 a 1 4095**



```
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> part-add /dev/nand0 a 1 4095
Created partition on /dev/nand0a
losh> █
```

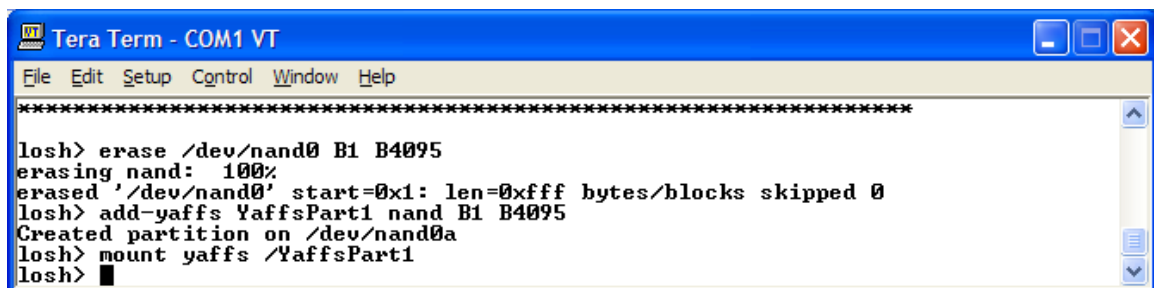
Notice that the partition was added on /dev/nand0a, when we go to mount it, we'll need to use that in the mount command.

#### 5.2.4 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 ('yaffs' here)
- <point> the name of the YAFFS partition

For a legacy example type **mount yaffs /YaffsPart1** and LogicLoader will display the following screen.



```
*****

losh> erase /dev/nand0 B1 B4095
erasing nand: 100%
erased '/dev/nand0' start=0x1: len=0xffff bytes/blocks skipped 0
losh> add-yaffs YaffsPart1 nand B1 B4095
Created partition on /dev/nand0a
losh> mount yaffs /YaffsPart1
losh> █
```

Notice that the <point> argument needs to correspond to the name of the partition (as defined by the 'add-yaffs' command) preceded by a forward slash. LogicLoader needs to mount all YAFFS partitions at the root-directory level. Thus, a partition added using:

**add-yaffs YaffsPart1 ...**

will be mounted using:

**mount ... /YaffsPart1.**

Note the absence of the '/' character during the 'add-yaffs' command and its presence during the 'mount' command.

To use the new mount command you'll want to type

**mount yaffs /dev/nand0a /YaffsPart1**

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> part-add /dev/nand0 a 1 4095
Created partition on /dev/nand0a
losh> mount yaffs /dev/nand0a /YaffsPart1
losh> █
```

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

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> part-add /dev/nand0 a 1 4095
Created partition on /dev/nand0a
losh> mount yaffs /dev/nand0a /YaffsPart1
losh> cd /YaffsPart1
losh> ls
D :                               lost-found      512
losh> █
```

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.

### 5.2.5 Copy a file to the partition

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

```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> help cp
usage:
  cp <src> <dest>

  copies file <src> to new file <dest>

example: cp /cf/foo /yaffs/bar
         cp /tftp/192.168.1.2:bar /yaffs/foo

losh> █
```

The two examples are to copy from a CF 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 Lab #5 Networking if you need a refresher.

Let's copy a CE image from the CF card to yaffs. You'll need the CF card mounted and the yaffs partition mounted as well. Type `cp /cf/NK.BIN /boot/NK.BIN`

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
copies file <src> to new file <dest>
example: cp /cf/foo /yaffs/bar
         cp /tftp/192.168.1.2:bar /yaffs/foo
losh> mount fatfs /cf
losh> cp /cf/NK.BIN /boot/NK.BIN
cp /cf/NK.BIN -> /boot/NK.BIN done
losh> █
```

You can also use tftp. Here is an example – remember, your ip address will be different and you have to setup your kit for communications before you try to copy the file.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> cp /tftp/192.168.1.149:PX0270_video_setup2.txt /boot/video_setup.txt
cp /tftp/192.168.1.149:PX0270_video_setup2.txt -> /boot/video_setup.txt done
```

```
Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> cd /boot
losh> ls
:                               video_setup.txt      43
:                               NK.bin 15111887
D :                               lost-found      512
losh>
```

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

## 6 Quiz

1. How many partitions can be used at one time?
  - a. 1
  - b. 2
  - c. 3
  - d. 4
2. Put the following steps to use the YAFFS file system within LogicLoader in the proper order.
  - a. Add the partitions in LogicLoader
  - b. Decide on a partitioning scheme
  - c. Mount the partitions using
  - d. Format the partitions by erasing the associated flash blocks
3. When must you erase the flash?
  - a. Every time you reboot
  - b. Only the first time you setup the partition
  - c. Whenever you want to change the partition
  - d. Never
4. When must you use the 'add-yaffs' or 'part-add' command?
  - a. Every time you reboot
  - b. Only the first time you setup the partition
  - c. Whenever you want to change the partition
  - d. The first Monday of the month



## LogicLoader Lab #7

### Using Scripts

---

#### 1 Abstract

This lab will teach you how to use scripting to optimize your application on your Zoom i.MX31 LITEKIT.

#### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- 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 TeraTerm. **Note:** Logic recommends using TeraTerm, a free program downloadable from the Internet.

#### 3 Basic instructions

Using the same process you learned in “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

#### 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.



## 5 Using scripts

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.

### 5.1 Basic scripting guidelines

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 TeraTerm 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 3 16; video-clear; draw-test;`

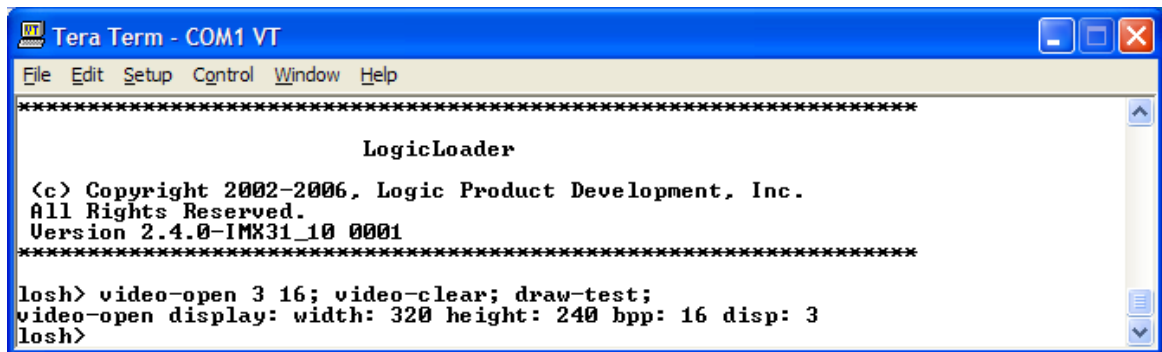
Or it can look like this:            `video-open 3 16`

`video-clear`

`draw-test`

Be sure to place a return after the last command to execute that last command, otherwise you'll have to hit return to see the test pattern.

Use File | Send File to send the video\_test script to your kit.

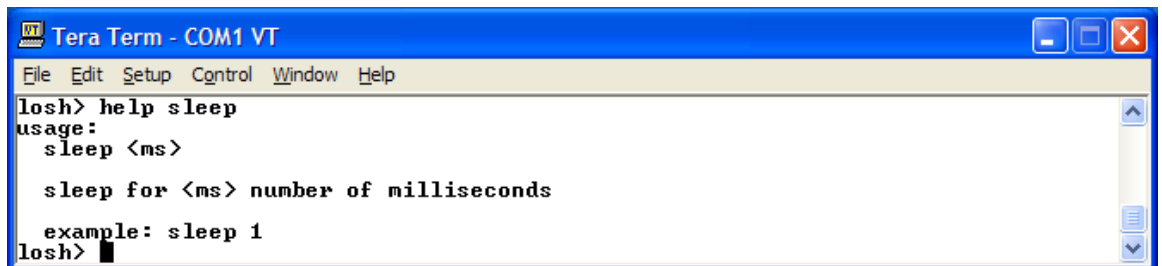


```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> video-open 3 16; video-clear; draw-test;
video-open display: width: 320 height: 240 bpp: 16 disp: 3
losh>
```

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**.



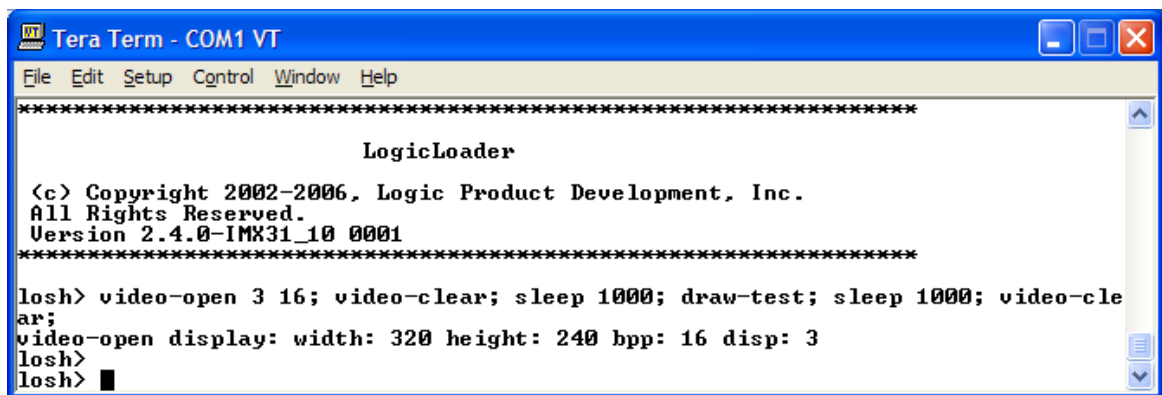
```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> help sleep
usage:
  sleep <ms>

  sleep for <ms> number of milliseconds

example: sleep 1
losh>
```

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.



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader

(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> video-open 3 16; video-clear; sleep 1000; draw-test; sleep 1000; video-clear;
video-open display: width: 320 height: 240 bpp: 16 disp: 3
losh>
losh>
```

Now you can actually see what the script is doing.

If you have a script saved as a text file on a CompactFlash card or on your kit, you can use the source command to run the script. Let's add the necessary commands to mount the CompactFlash card and then run the script from the card. Use the commands you learned in "Lab #6 File Systems" to mount the CompactFlash card. Then type **source /cf/video\_test.txt**

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

```

Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> mount fatfs /cf
losh> cd /cf
losh> ls
  r:                video_test.txt          91
  r:                NK.BIN 11165267
  r:                1007274_lolo.elf 245264
  r:                1007274_lolo.upd 245016
  D:                mpeg_files          0
  D:                Digikey Training     0
losh> source /cf/video_test.txt
video-open display: width: 320 height: 240 bpp: 16 disp: 3
losh>

```

Note that the commands didn't 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 do it all over again. Next we'll look at how to store a script so it runs every time you power-up your kit.

## 6 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 i.MX is the resident flash array (`/dev/config`).

### Config Block Basics

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

```

Tera Term - COM1 VT
File Edit Setup Control Window Help

*****
                        LogicLoader

<c> Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****

losh> hd /dev/config
error: hd: failed to read </dev/config>
losh> █

```

Hmm... not quite what we were looking for. It seems that this kit has never had the config block initialized. That's okay, you need 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.

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

```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> hd /dev/config
error: hd: failed to read (/dev/config)
losh> config CREATE
creating new config: offset: 0x0 pos: 0
total size: 0x10000, address 0x40000
Nor flash erase: The area (or parts of it) 0x40000 - 0x10000 is locked.

***** WARNING!! *****

You are about to erase the config block.
Do you wish to continue? (y/n) █
```

Ah, the config block is protected. You'll have to type **y** to continue.

```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> config CREATE
creating new config: offset: 0x0 pos: 0
total size: 0x10000, address 0x40000
Nor flash erase: The area (or parts of it) 0x40000 - 0x10000 is locked.

***** WARNING!! *****

You are about to erase the config block.
Do you wish to continue? (y/n)
erasing flash: 100%
burning flash: 100%
dev_config burn good (65536 bytes)
losh> █
```

In LoLo 2.4 all config CREATE instances are met with the warning screen. Since you are always warned, you won't know what is in the config block unless you look at it. If you aren't sure what is in config, you should probably answer **n** and use **cat /dev/config** to view the config block contents. In either case, LoLo will inform you what it is doing. Go ahead and use the config CREATE command to erase/initialize your config block.

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

```
Tera Term - COM1 VT
File Edit Setup Control Window Help

losh> hd /dev/config 256
0x8005ad00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ad90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ada0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005adb0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005adc0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005add0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005ade0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x8005adf0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
losh> █
```

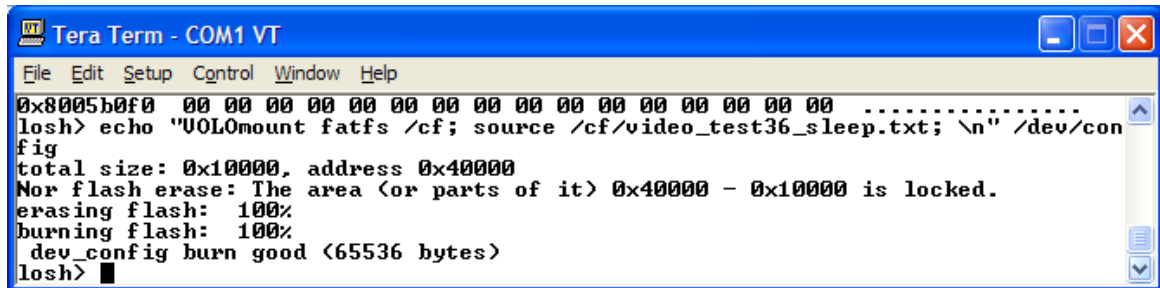
A nice clean slate. Let's put a script in the config block that will run our video test from the CF card

## 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 LoLo doesn't. The V is for Verbose.

Type `echo "VOL0mount fatfs /cf; source /cf/video_test36_sleep.txt; \n" /dev/config`



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
0x8005b0f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
losh> echo "VOL0mount fatfs /cf; source /cf/video_test36_sleep.txt; \n" /dev/config
total size: 0x10000, address 0x40000
Nor flash erase: The area (or parts of it) 0x40000 - 0x10000 is locked.
erasing flash: 100%
burning flash: 100%
dev_config burn good (65536 bytes)
losh> █
```

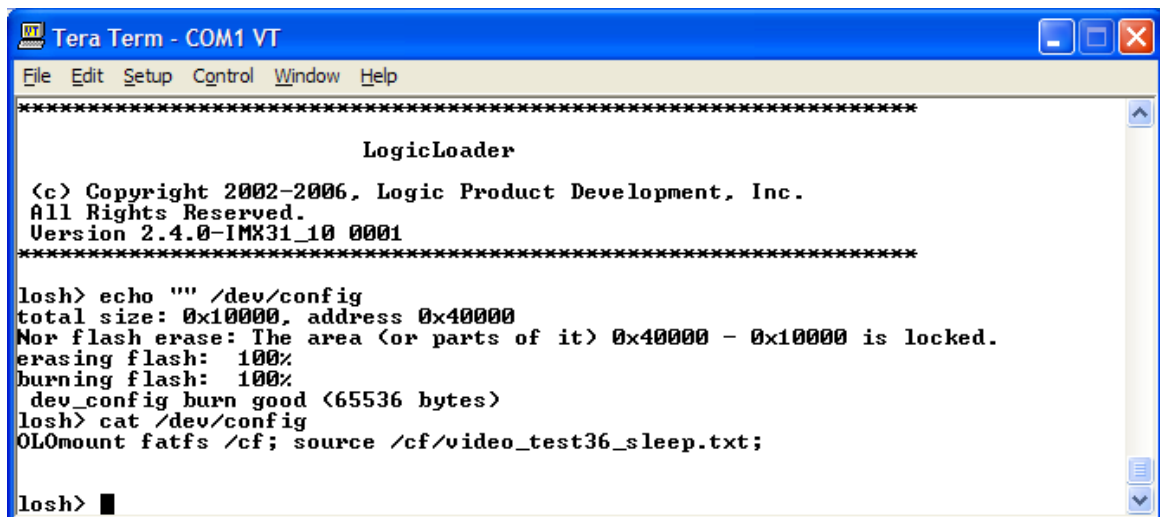
If the video test script is on your CF card, you can reboot your kit and it should run the video test.

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.

Type `echo "" /dev/config`

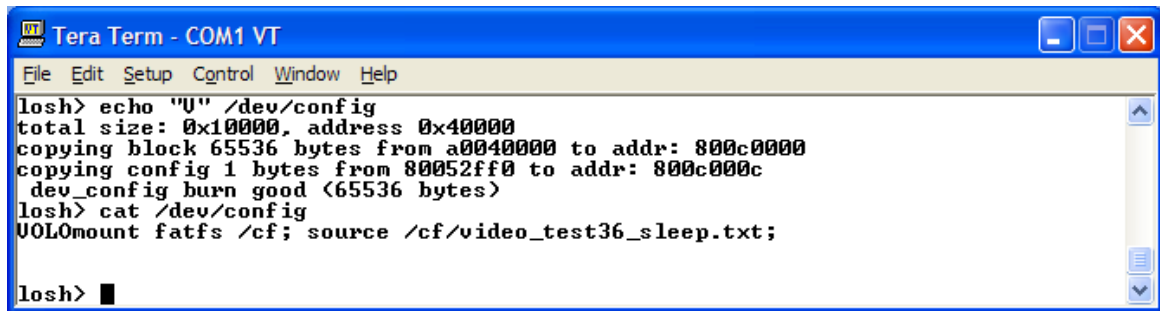
We just changed the first character in the first command to a space - OLO. Now the kit will not recognize it as a boot script and will not run the script. We can use the 'cat' command to display what is in the config block without having to see all of the hex values in the memory locations.

Type `cat /dev/config`



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
losh> echo "" /dev/config
total size: 0x10000, address 0x40000
Nor flash erase: The area (or parts of it) 0x40000 - 0x10000 is locked.
erasing flash: 100%
burning flash: 100%
dev_config burn good (65536 bytes)
losh> cat /dev/config
OLOmount fatfs /cf; source /cf/video_test36_sleep.txt;
losh> █
```

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. Type **echo "V" /dev/config** then use 'cat' to display the Config Block contents.



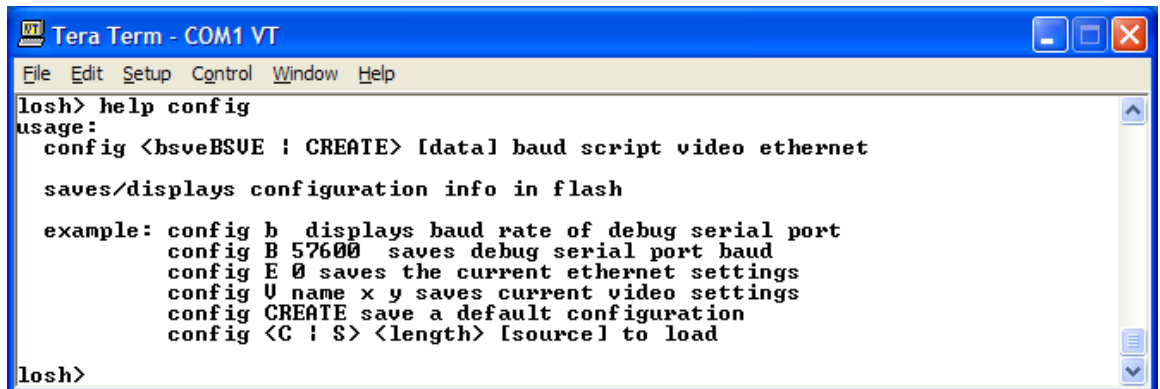
```
losh> echo "U" /dev/config
total size: 0x10000, address 0x40000
copying block 65536 bytes from a0040000 to addr: 800c0000
copying config 1 bytes from 80052ff0 to addr: 800c000c
dev_config burn good (65536 bytes)
losh> cat /dev/config
UOL0mount fatfs /cf; source /cf/video_test36_sleep.txt;

losh> █
```

We put the V back into the script and re-enabled it.

Using the echo command with long scripts can be a pain. Chances are you messed up on one of those "echos" and had to do it again. 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**.



```
losh> help config
usage:
  config <bsveBSUE ! CREATE> [data] baud script video ethernet

  saves/displays configuration info in flash

  example: config b displays baud rate of debug serial port
            config B 57600 saves debug serial port baud
            config E 0 saves the current ethernet settings
            config U name x y saves current video settings
            config CREATE save a default configuration
            config <C ! S> <length> [source] to load

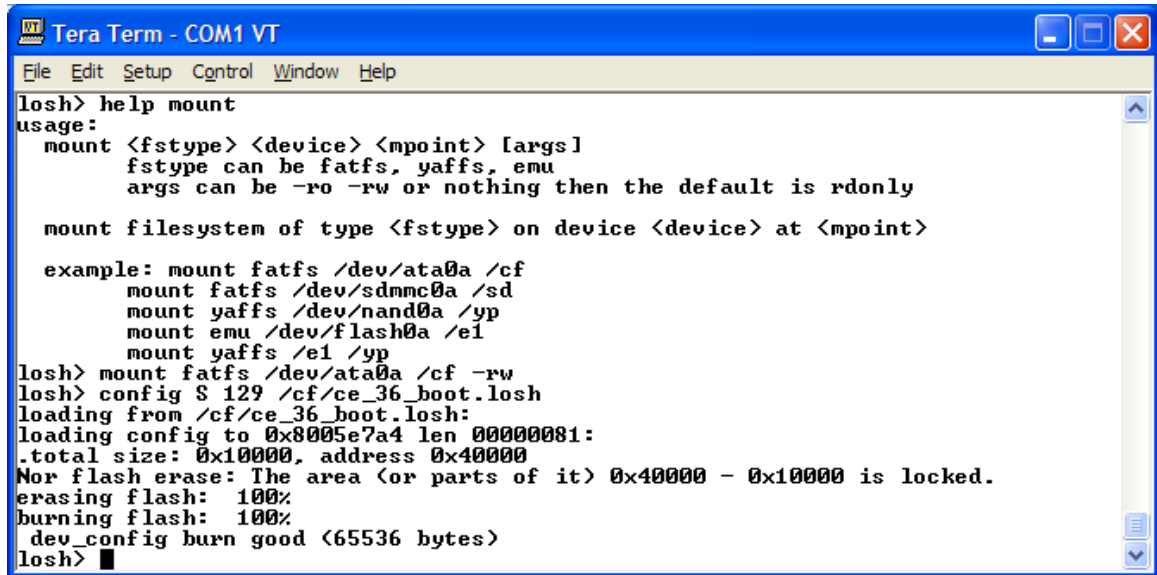
losh>
```

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

Please note, you'll have to use the new mount syntax to mount the CF card if that is where you have the script. This will have to be done before you use the config S command.

Type **mount fatfs /dev/ata0a /cf -rw**

Then type **config S 129 /cf/ce\_36\_boot.losh**



```
losh> help mount
usage:
  mount <fstype> <device> <mpoint> [args]
  fstype can be fatfs, yaffs, emu
  args can be -ro -rw or nothing then the default is rdonly

  mount filesystem of type <fstype> on device <device> at <mpoint>

  example: mount fatfs /dev/ata0a /cf
           mount fatfs /dev/sdmmc0a /sd
           mount yaffs /dev/nand0a /yp
           mount emu /dev/flash0a /e1
           mount yaffs /e1 /yp
losh> mount fatfs /dev/ata0a /cf -rw
losh> config $ 129 /cf/ce_36_boot.losh
loading from /cf/ce_36_boot.losh:
loading config to 0x8005e7a4 len 00000081:
total size: 0x10000, address 0x40000
Nor flash erase: The area (or parts of it) 0x40000 - 0x10000 is locked.
erasing flash: 100%
burning flash: 100%
dev_config burn good (65536 bytes)
losh>
```

Reboot your kit and it will run your script.

You can also send the script over the serial port by leaving the location off the command and using File |Send File to send the script. Create a script as a text file on your workstation and try this method too.

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 simply 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.

If you did everything correctly, you should now have a CE image running on your kit. In the next lab we'll cover the individual steps necessary to load and boot a CE image.



## LogicLoader Lab #8

### Loading an NK.BIN

---

#### 1 Abstract

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

#### 2 What you will need

- A Logic Zoom i.MX31 LITEKIT
- A development workstation (PC or laptop)
- An available serial port or USB-to-serial adapter
- A CompactFlash Card
- An Ethernet connection between the workstation and the kit
- A terminal emulation program such as HyperTerminal or TeraTerm. **Note:** Logic recommends using TeraTerm, a free program downloadable from the Internet.

#### 3 Basic instructions

Using the same process you learned in “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

#### 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.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
*****
                        LogicLoader
(c) Copyright 2002-2006, Logic Product Development, Inc.
All Rights Reserved.
Version 2.4.0-IMX31_10 0001
*****
losh> █

```

## 5 Load a file from CompactFlash

In previous labs you learned how to mount a CompactFlash card. We will now go a step further to load an NK.BIN WinCE image from the CompactFlash card onto your kit. First let's look at the help for the 'load' command. Type **help load**

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> help load
usage:
  load <type> [source]
      <type> /tftp/<server:filename>
      raw <dest addr> <length> [source]
      <type> -dhcp

      <type> can be 'elf', 'bin' (wince img), 'raw' or 'srec'
      when no [source] is given, stdin is read
      -dhcp looks in the boot packet for a filename

  download a binary image

  example: load elf
losh> █

```

You can see from the output that there are several ways to use the 'load' command. We are interested in loading a WinCE image, so we will use **load bin**. Use the commands you learned in "Lab #6 File Systems" to mount the CompactFlash card. Now type **ls** to view the contents of the card.

```

Tera Term - COM1 VT
File Edit Setup Control Window Help
losh> mount fatfs /cf
losh> cd /cf
losh> ls
r:                                     NK.BIN 15776631
r:          video_test_2.txt          78
r:          video_test.txt            91
r:    1007274_lolo.elf                245264
r:    1007274_lolo.upd                245016
D :          Digikey Training          0
losh> █

```

You can see the video-test script from the scripting lab and a file called NK.BIN; that is the WinCE 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.



