Freescale Semiconductor
Application Note

Document Number: AN4426
Rev. 0, December 2011

LTIB Quick Start: Targeting the ColdFire MCF54418Tower Board

by: Soledad Godinez and Jaime Hueso
Guadalajara
Mexico

Contents

1 Introduction

The purpose of this document is to indicate step-by-step how to accomplish these tasks:

- Install the BSP on a host development system.
- Run Linux Target Image Builder (LTIB) to build target images.
- Boot Linux on the ColdFire MCF54418 Tower board.

This document is a guide for people who want to properly set up the Freescale Linux Target Image Builder (LTIB) for the ColdFire MCF54418 Tower Board Support Package (BSP).

2 LTIB

2.1 Introduction

Freescale GNU/Linux Target Image Builder (LTIB) is an Open Source tool created by Freescale. You can use LTIB to build Linux target images, composed of a set of packages. It has been released under the terms of the GNU General Public License (GPL). LTIB provides a lightweight command line interface to Perl scripts and LKC-syntax configuration menus to perform the following functions:

- Build kernel, bootloader, and application packages from source.

- Deploy built packages to a root file system (RFS).
- Prepare appropriate kernel or RFS image file ready for network or flash-based use on the embedded target board.
- Package management, including creation of board/architecture-specific Red Hat Package Manager (RPM) for installing a package into the RFS, and patch generation for package source modifications.

2.2 Installing the BSP

In this section, you will learn how to install the LTIB in your host machine.

1. In Linux create a new folder. Figure 1 shows an example where the name of the new folder is M54418Tower:

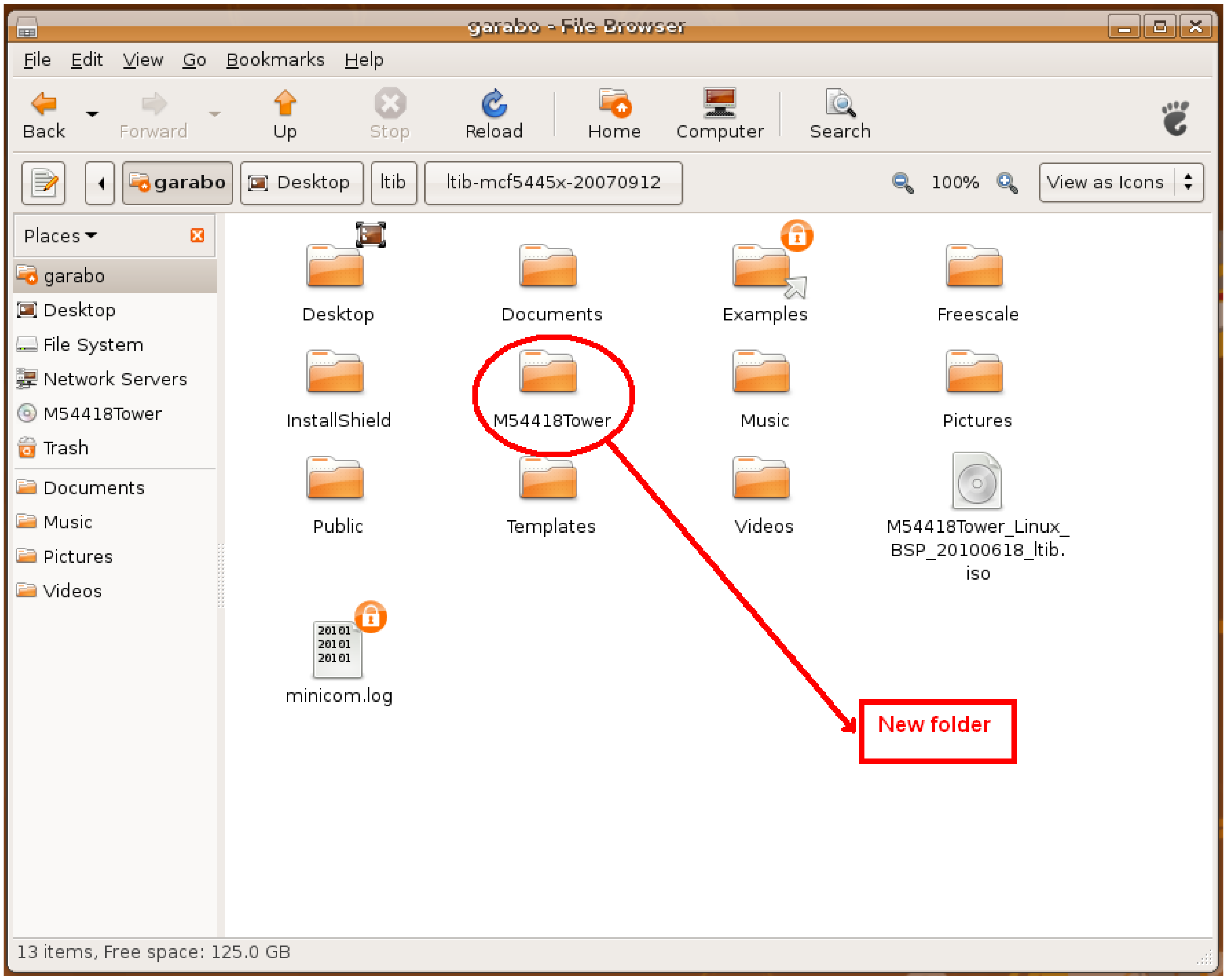


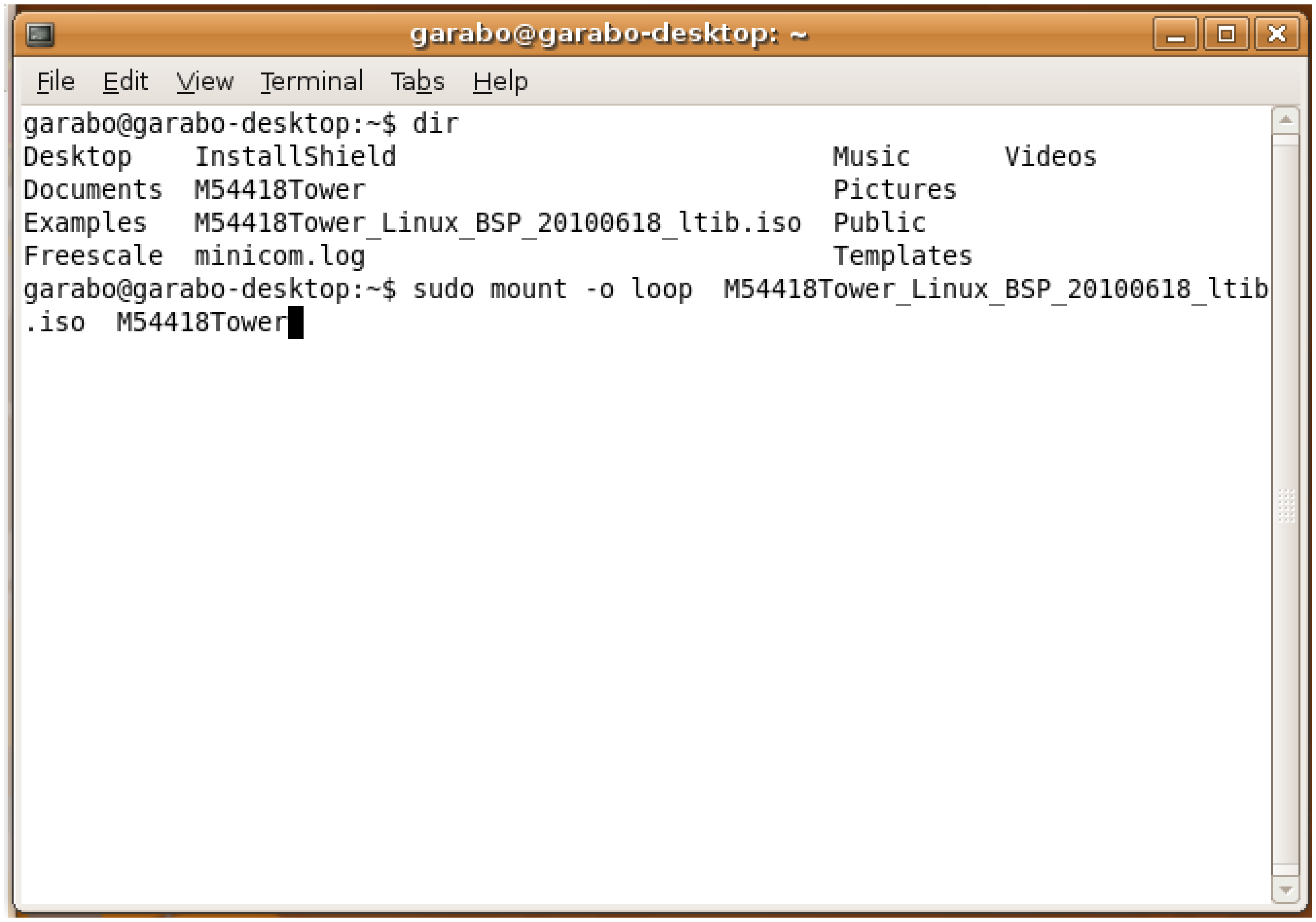
Figure 1. Create a new folder

2. As root, mount the ISO image on your machine:

```
mount -o loop <target-bsp.iso> <mount point>
```

For Ubuntu, use `'sudo'` to execute the command as a root. In this case, you can use `'sudo mount'` to mount the ISO image, as shown in Figure 2.

```
sudo mount -o loop <target-bsp.iso> <mount point>
```

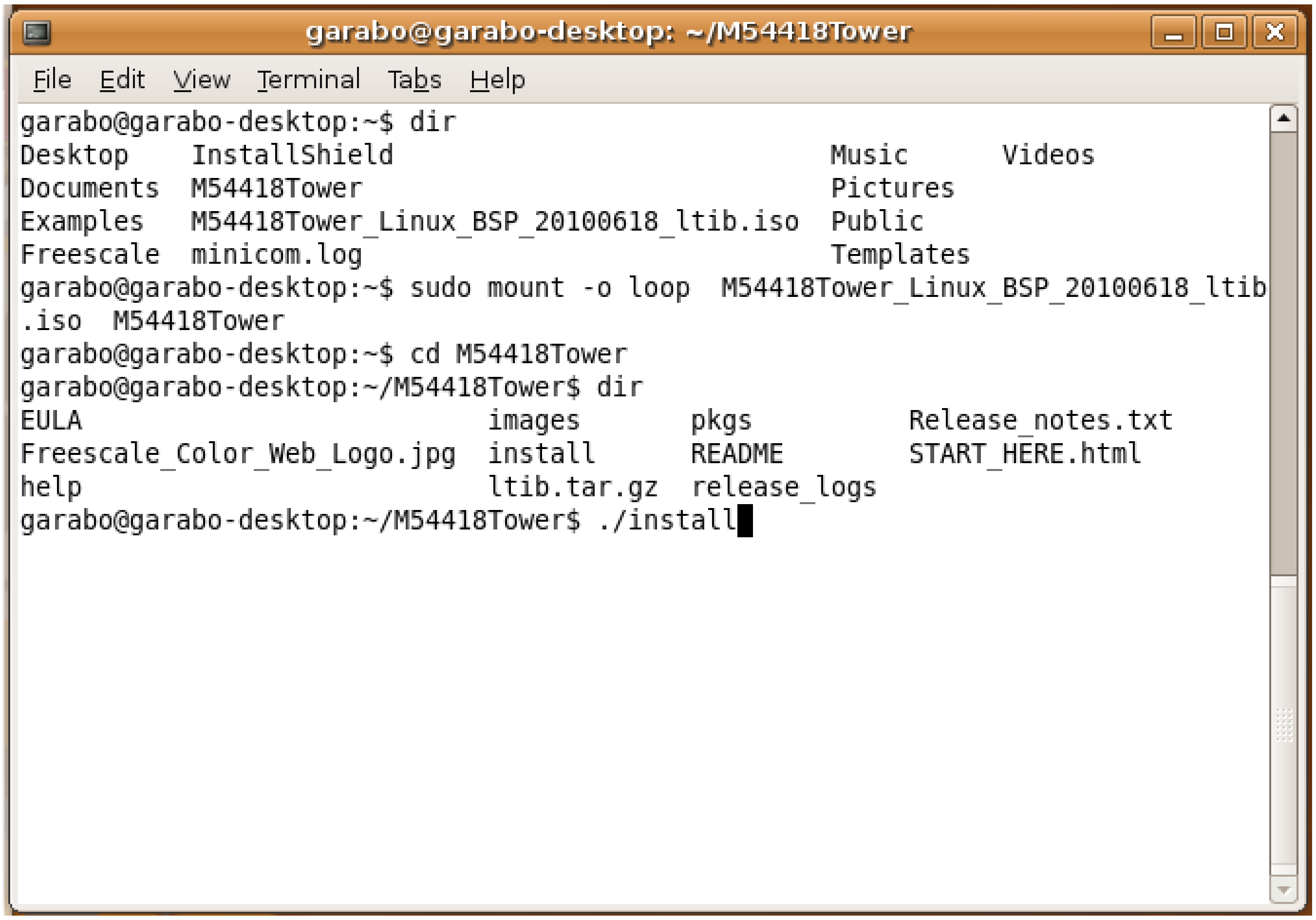


```
garabo@garabo-desktop:~$ dir
Desktop    InstallShield                            Music      Videos
Documents  M54418Tower                              Pictures
Examples   M54418Tower_Linux_BSP_20100618_ltib.iso  Public
Freescale  minicom.log                              Templates
garabo@garabo-desktop:~$ sudo mount -o loop  M54418Tower_Linux_BSP_20100618_ltib
.iso  M54418Tower
```

Figure 2. Mount the ISO

3. As a non-root user, install the LTIB. Figure 3 shows this step:

```
<mount point> ./install
```


garabo@garabo-desktop: ~/M54418Tower
File Edit View Terminal Tabs Help

```
garabo@garabo-desktop:~$ dir
Desktop    InstallShield                           Music     Videos
Documents  M54418Tower                             Pictures
Examples   M54418Tower_Linux_BSP_20100618_ltib.iso Public
Freescale  minicom.log                             Templates
garabo@garabo-desktop:~$ sudo mount -o loop  M54418Tower_Linux_BSP_20100618_ltib
.iso  M54418Tower
garabo@garabo-desktop:~$ cd M54418Tower
garabo@garabo-desktop:~/M54418Tower$ dir
EULA                          images       pkgs          Release_notes.txt
Freescale_Color_Web_Logo.jpg  install      README        START_HERE.html
help                          ltib.tar.gz  release_logs
garabo@garabo-desktop:~/M54418Tower$ ./install
```

Figure 3. Install LTIB

4. The BSP End User License Agreement (EULA) will be presented for acceptance. To continue installing accept the license.
5. Next, input the desired LTIB install path. Be sure the user has the correct permissions to modify the install path.

NOTE

There are no uninstall scripts. To uninstall LTIB you need to remove the `/opt/freescale/pkgs, /opt/freescale/ltib and <install_path>/ltib` directories manually.

For this step you first need to create a new folder in Linux. Figure 4 illustrates the folder for this example.

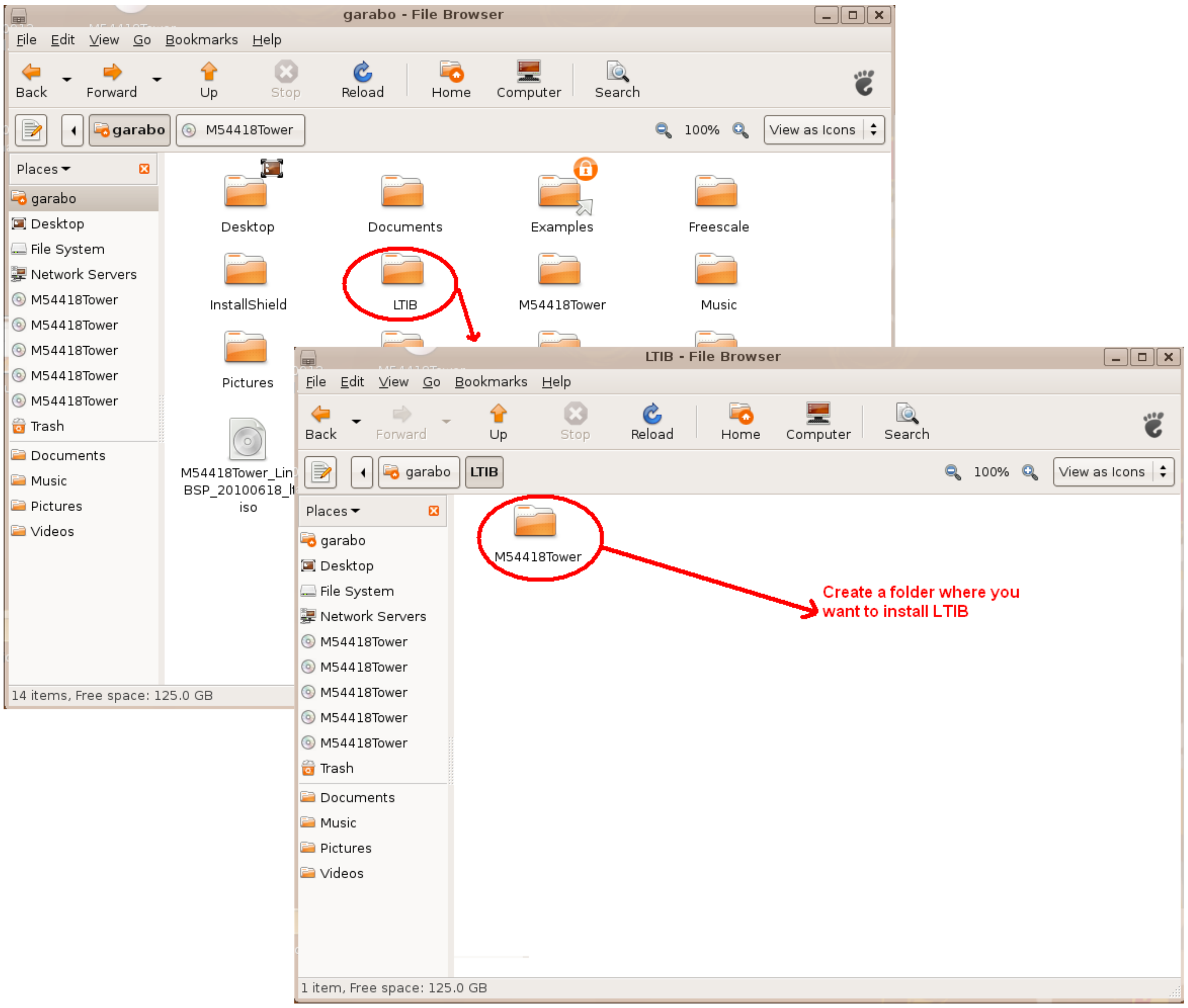
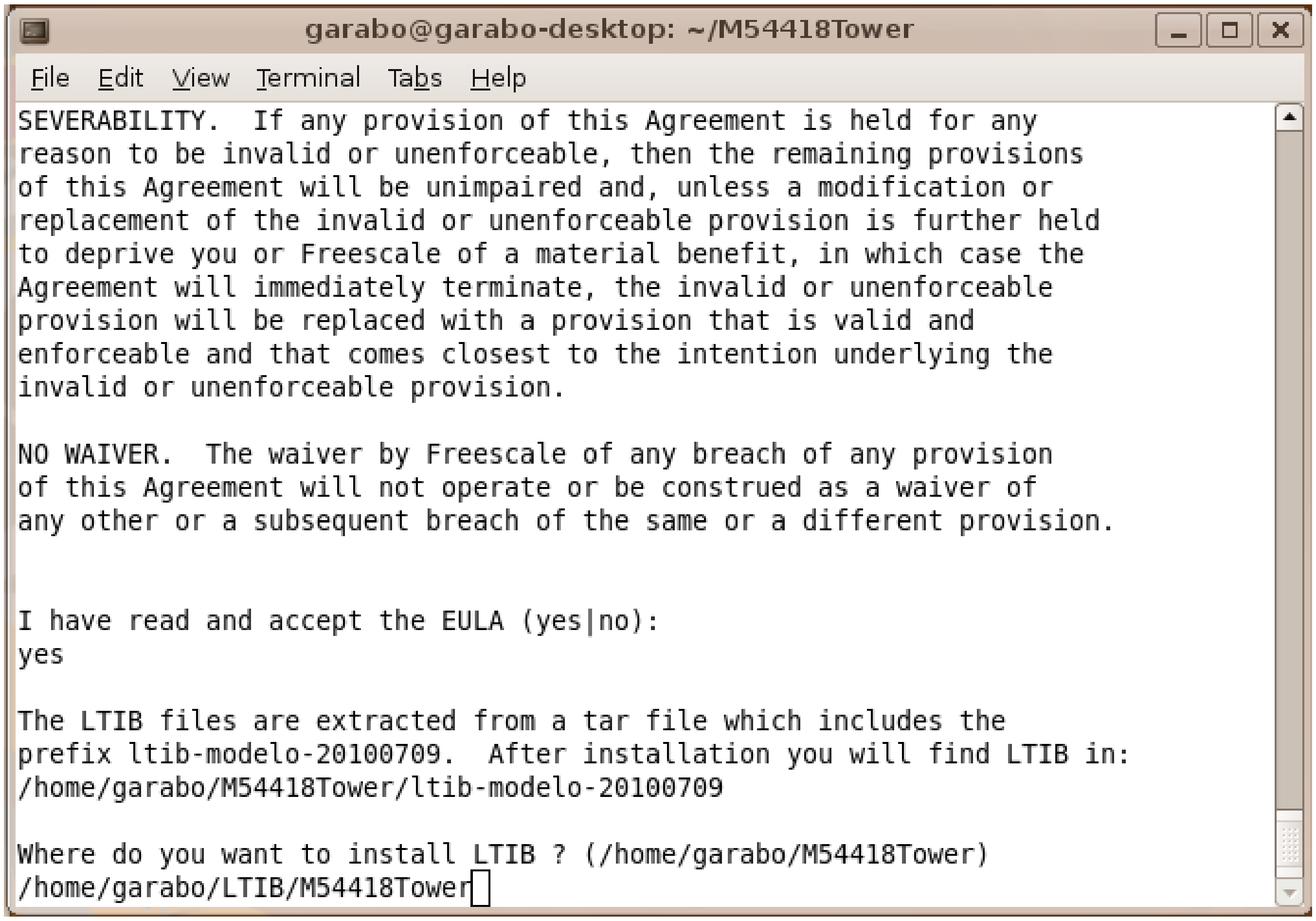

garabo - File Browser
File Edit View Go Bookmarks Help
Back Forward Up Stop Reload Home Computer Search
garabo M54418Tower
100% View as Icons
Places
garabo
Desktop
File System
Network Servers
M54418Tower
Trash
Documents
Music
Pictures
Videos
Desktop Documents Examples Freescale
InstallShield LTIB M54418Tower Music
Pictures
M54418Tower_Lin BSP_20100618_l iso
14 items, Free space: 125.0 GB
LTIB - File Browser
garabo LTIB
M54418Tower
1 item, Free space: 125.0 GB
Create a folder where you want to install LTIB

Figure 4. Folder for the LTIB

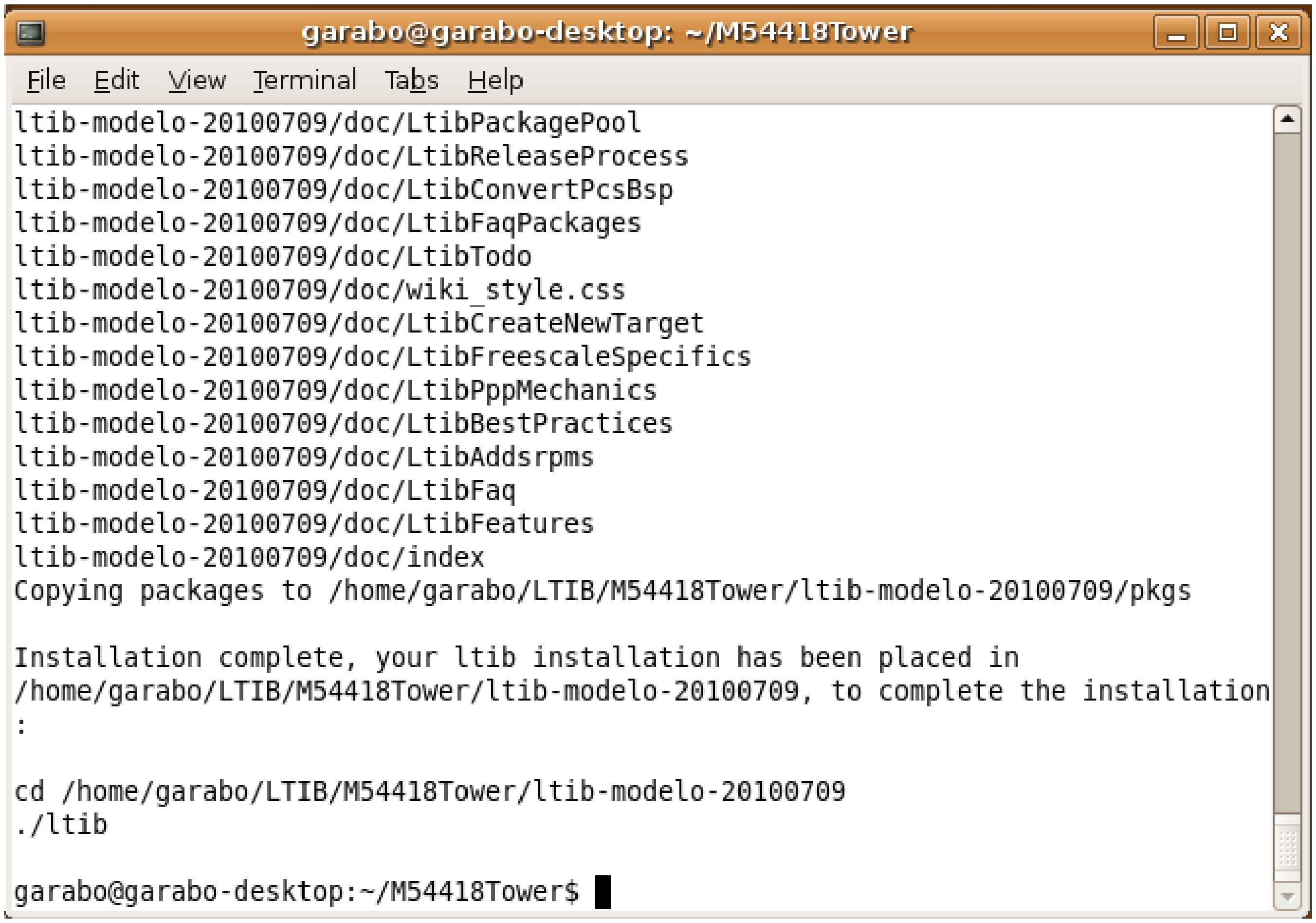
Figure 5 exemplifies the LTIB install path.


garabo@garabo-desktop: ~/M54418Tower
File Edit View Terminal Tabs Help

```
SEVERABILITY.  If any provision of this Agreement is held for any
reason to be invalid or unenforceable, then the remaining provisions
of this Agreement will be unimpaired and, unless a modification or
replacement of the invalid or unenforceable provision is further held
to deprive you or Freescale of a material benefit, in which case the
Agreement will immediately terminate, the invalid or unenforceable
provision will be replaced with a provision that is valid and
enforceable and that comes closest to the intention underlying the
invalid or unenforceable provision.

NO WAIVER.  The waiver by Freescale of any breach of any provision
of this Agreement will not operate or be construed as a waiver of
any other or a subsequent breach of the same or a different provision.


I have read and accept the EULA (yes|no):
yes

The LTIB files are extracted from a tar file which includes the
prefix ltib-modelo-20100709.  After installation you will find LTIB in:
/home/garabo/M54418Tower/ltib-modelo-20100709

Where do you want to install LTIB ? (/home/garabo/M54418Tower)
/home/garabo/LTIB/M54418Tower
```

Figure 5. Path

A large number of text lines will scroll in the terminal until the process finishes, as shown in Figure 6.

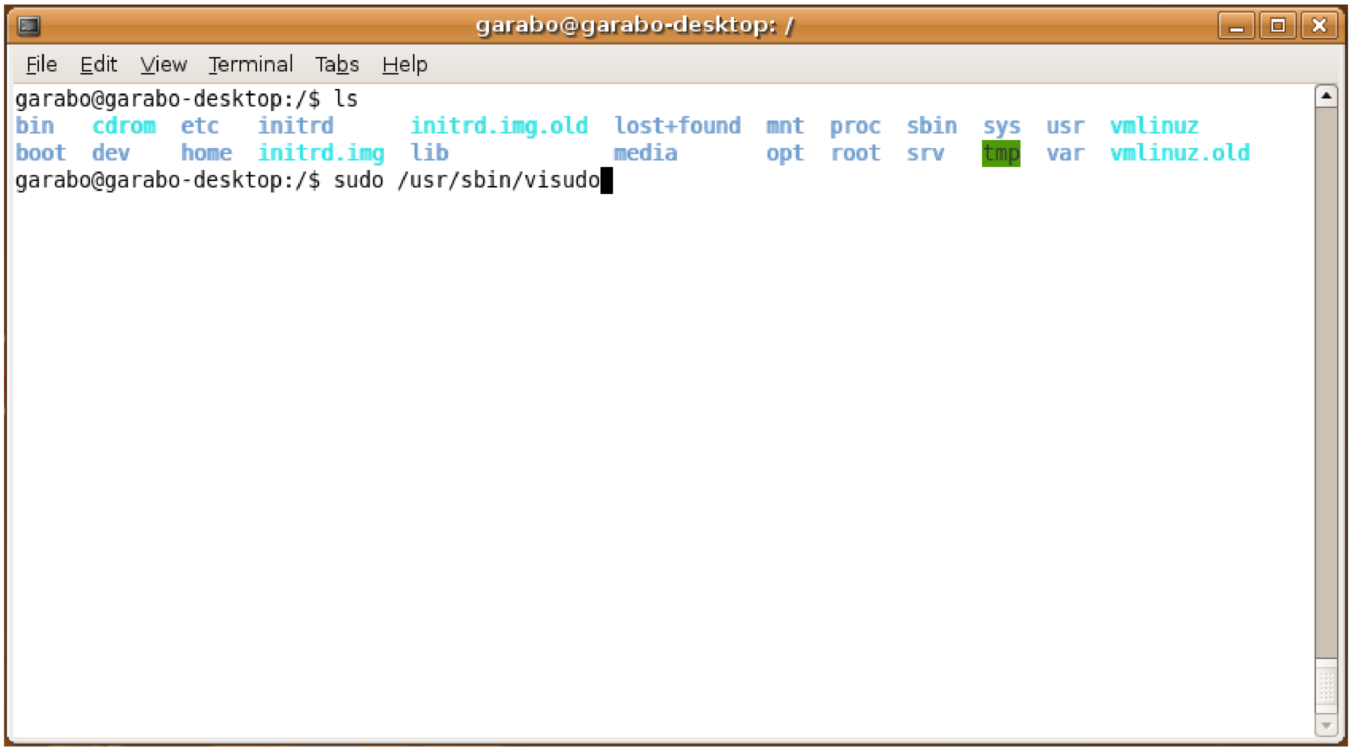

garabo@garabo-desktop: ~/M54418Tower
File Edit View Terminal Tabs Help

```
ltib-modelo-20100709/doc/LtibPackagePool
ltib-modelo-20100709/doc/LtibReleaseProcess
ltib-modelo-20100709/doc/LtibConvertPcsBsp
ltib-modelo-20100709/doc/LtibFaqPackages
ltib-modelo-20100709/doc/LtibTodo
ltib-modelo-20100709/doc/wiki_style.css
ltib-modelo-20100709/doc/LtibCreateNewTarget
ltib-modelo-20100709/doc/LtibFreescaleSpecifics
ltib-modelo-20100709/doc/LtibPppMechanics
ltib-modelo-20100709/doc/LtibBestPractices
ltib-modelo-20100709/doc/LtibAddsrpms
ltib-modelo-20100709/doc/LtibFaq
ltib-modelo-20100709/doc/LtibFeatures
ltib-modelo-20100709/doc/index
Copying packages to /home/garabo/LTIB/M54418Tower/ltib-modelo-20100709/pkgs

Installation complete, your ltib installation has been placed in
/home/garabo/LTIB/M54418Tower/ltib-modelo-20100709, to complete the installation
:

cd /home/garabo/LTIB/M54418Tower/ltib-modelo-20100709
./ltib

garabo@garabo-desktop:~/M54418Tower$ █
```

Figure 6. Installation complete

2.3 Running LTIB

1. First, configure the sudo permission that allows execution of rpm commands as root without a password. Switch to root and execute the command “/usr/sbin/visudo”. Please refer to Figure 7.



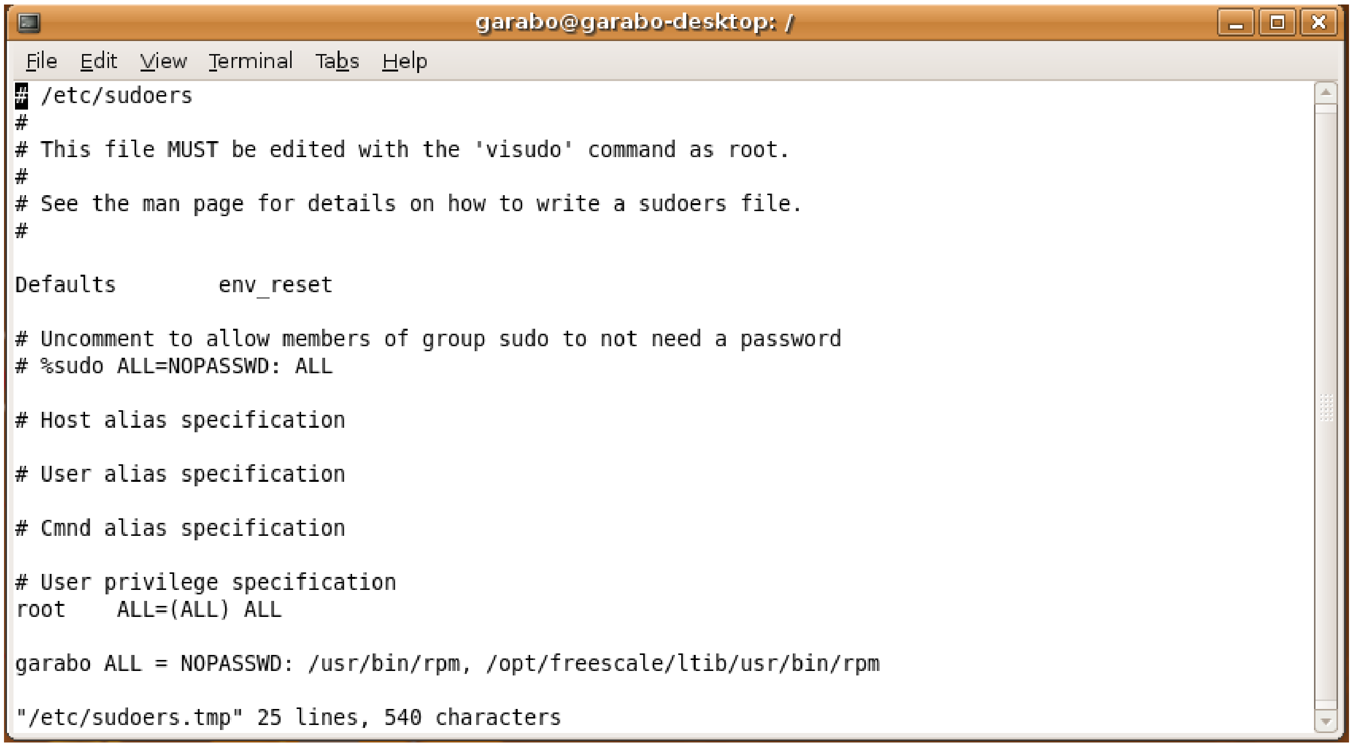
```
garabo@garabo-desktop:/$ ls
bin   cdrom  etc   initrd      initrd.img.old  lost+found  mnt  proc  sbin  sys  usr  vmlinuz
boot  dev    home  initrd.img  lib             media       opt  root  srv   tmp  var  vmlinuz.old
garabo@garabo-desktop:/$ sudo /usr/sbin/visudo
```

Figure 7. Sudo permission

Then, add the following line in the User privilege section (in this case the login user name is “garabo”):

```
garabo ALL = NOPASSWD: /bin/rpm, /opt/freescale/ltib/usr/bin/rpm
```

Figure 8 shows this line.



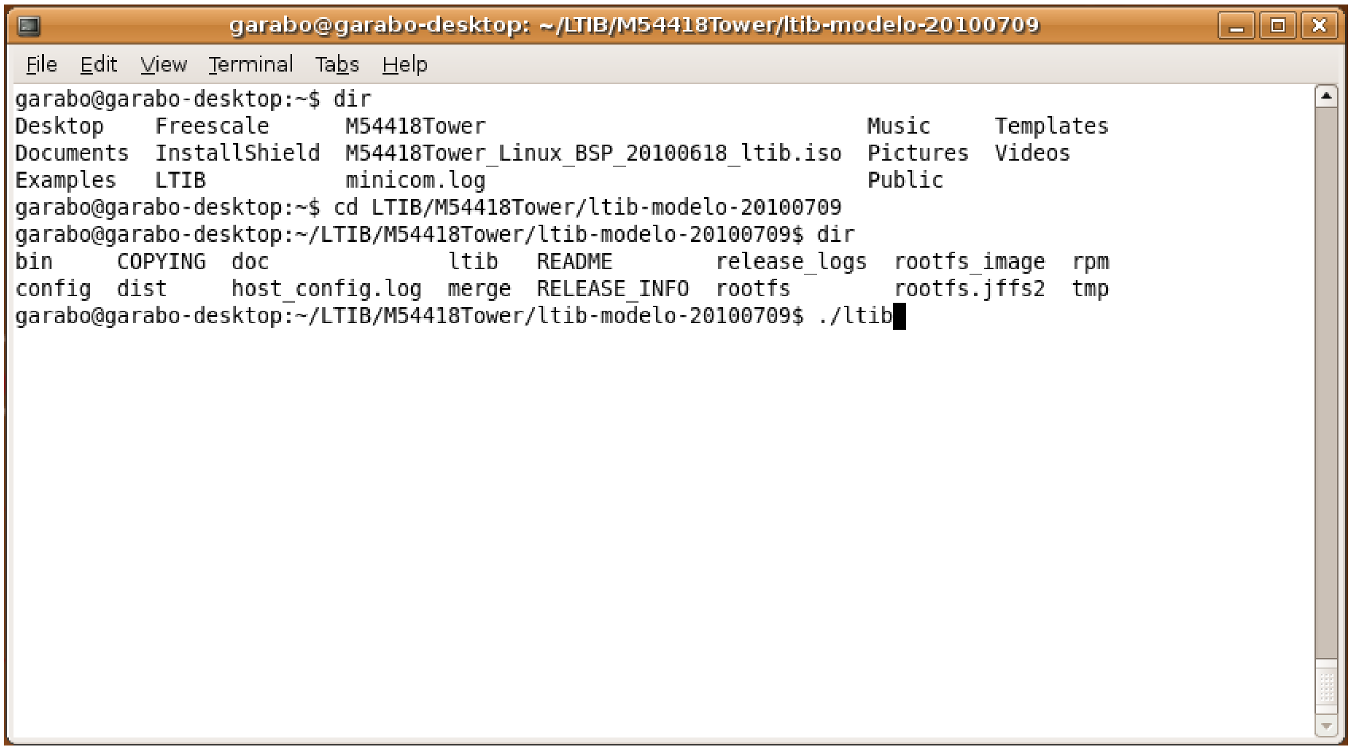
```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset

# Uncomment to allow members of group sudo to not need a password
# %sudo ALL=NOPASSWD: ALL

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

garabo ALL = NOPASSWD: /usr/bin/rpm, /opt/freescale/ltib/usr/bin/rpm

"/etc/sudoers.tmp" 25 lines, 540 characters
```

Figure 8. Line as seen in User privilege section

2. Change to the directory into which you installed ./ltib and run it.

```
cd <install_path>
./ltib
```

Figure 9 shows how to do this step.


garabo@garabo-desktop: ~/LTIB/M54418Tower/ltib-modelo-20100709
File Edit View Terminal Tabs Help

```
garabo@garabo-desktop:~$ dir
Desktop    Freescale     M54418Tower                           Music     Templates
Documents  InstallShield M54418Tower_Linux_BSP_20100618_ltib.iso Pictures Videos
Examples   LTIB          minicom.log                           Public
garabo@garabo-desktop:~$ cd LTIB/M54418Tower/ltib-modelo-20100709
garabo@garabo-desktop:~/LTIB/M54418Tower/ltib-modelo-20100709$ dir
bin     COPYING  doc             ltib   README       release_logs  rootfs_image  rpm
config  dist     host_config.log merge  RELEASE_INFO rootfs        rootfs.jffs2  tmp
garabo@garabo-desktop:~/LTIB/M54418Tower/ltib-modelo-20100709$ ./ltib
```

Figure 9. Run ./ltib

Package	Minimum version	Installed info
glibc-headers	0	not installed
glibc-devel	0	not installed
gcc-c++	2.96	not installed
zlib-devel	0	not installed
rpm	0	not installed
rpm-build	0	not installed
ncurses-devel	0	not installed
m4	0	not installed
bison	0	not installed
patch	0	not installed

The LTIB will show you which packages were not installed in your host. Please install the packages according to the names listed in the table. If the packages are not all successfully installed, then LTIB will not build.

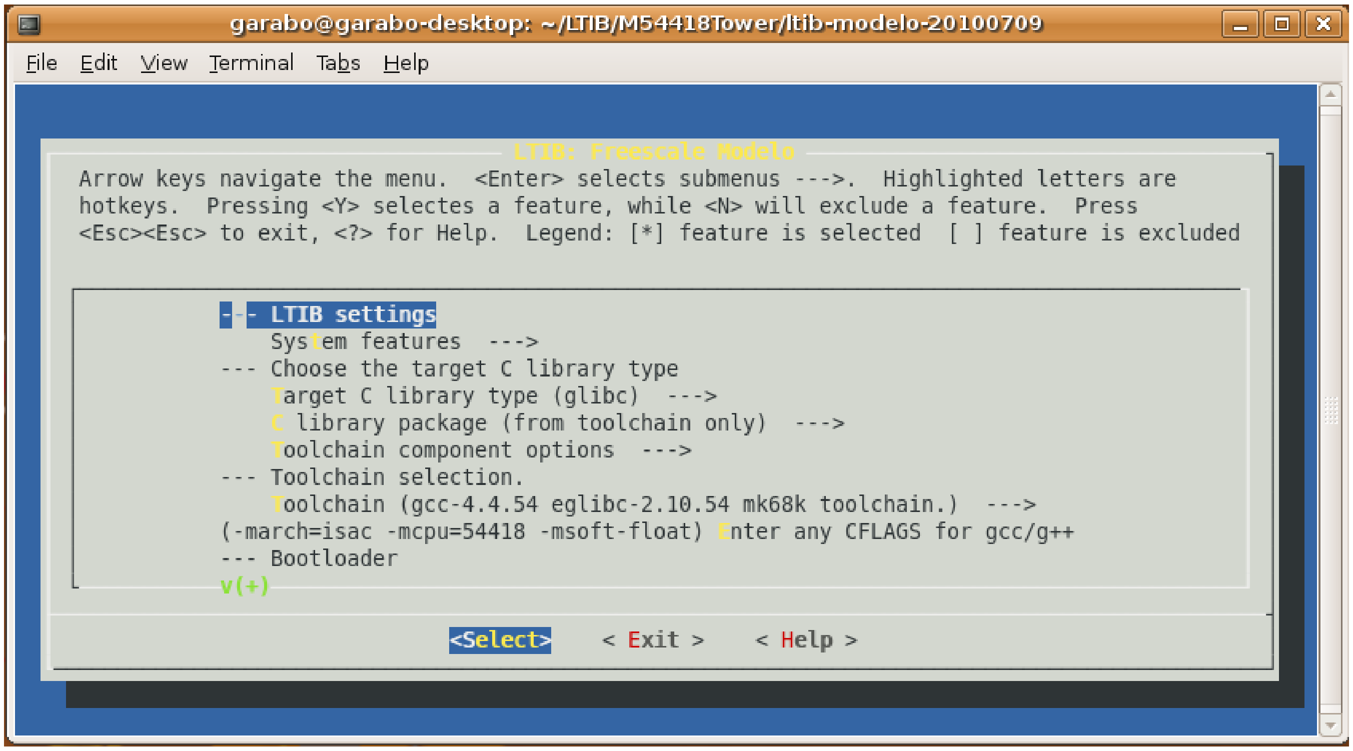
NOTE

In Ubuntu, some packages do not have the same name as the ones listed above. Below is the list of the packages that can be found by Synaptic Package Manager in Ubuntu GUI.

```
glibc-source
g++
rpm
libncurses
bison, m4
patch
```

3. When it executes the first time, LTIB runs a number of host packages that need to be built and installed in order to support LTIB. This may take a few minutes. Then run ./ltib –c to modify the project configuration. This will prompt the platform/board configuration. In the board configuration screen, change settings and select packages as appropriate. When you exit the configuration screen, your target image will compile/build again to be adjusted accordingly. Figure 10 shows the board configuration screen.

```
./ltib -c
```



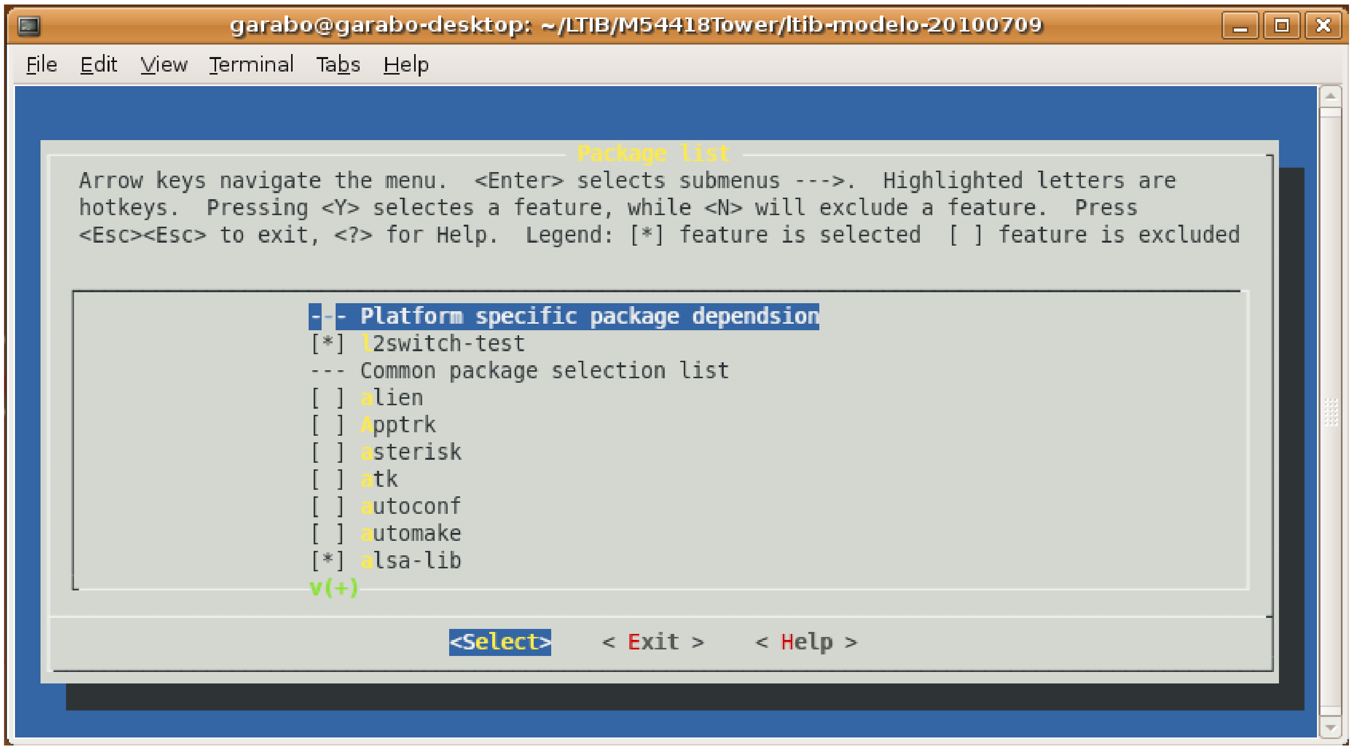
garabo@garabo-desktop: ~/LTIB/M54418Tower/ltib-modelo-20100709
File Edit View Terminal Tabs Help
LTIB: Freescale Modelo

```
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
hotkeys.  Pressing <Y> selectes a feature, while <N> will exclude a feature.  Press
<Esc><Esc> to exit, <?> for Help.  Legend: [*] feature is selected  [ ] feature is excluded

          --- LTIB settings
              System features  --->
          --- Choose the target C library type
              Target C library type (glibc)  --->
              C library package (from toolchain only)  --->
              Toolchain component options  --->
          --- Toolchain selection.
              Toolchain (gcc-4.4.54 eglibc-2.10.54 mk68k toolchain.)  --->
          (-march=isac -mcpu=54418 -msoft-float) Enter any CFLAGS for gcc/g++
          --- Bootloader
          v(+)

                   <Select>    < Exit >    < Help >
```

Figure 10. Board configuration

a. Package list:

To add or remove a package press the space bar. An asterisk shows that the package is selected to be built by LTIB and will be available on the target file system.
A working configuration is loaded by default. For information about the package, you can select Help at the bottom of the page.
On any Linux system, when building, installing, and executing a package, there may be dependencies with other packages, drivers, modules, etc. By selecting a new package, LTIB will try to build it accordingly. But when it detects some sort of dependency to other software that could not be resolved, it may try to access the Internet and download the missing component. If an error occurs when retrieving these components, the build process will fail and you may need to fix the issue yourself. Normally, it can be fixed by installing the missing libraries, modules, or packages to your Linux distribution, or by placing source packages in the /opt/freescale/pkgs directory.

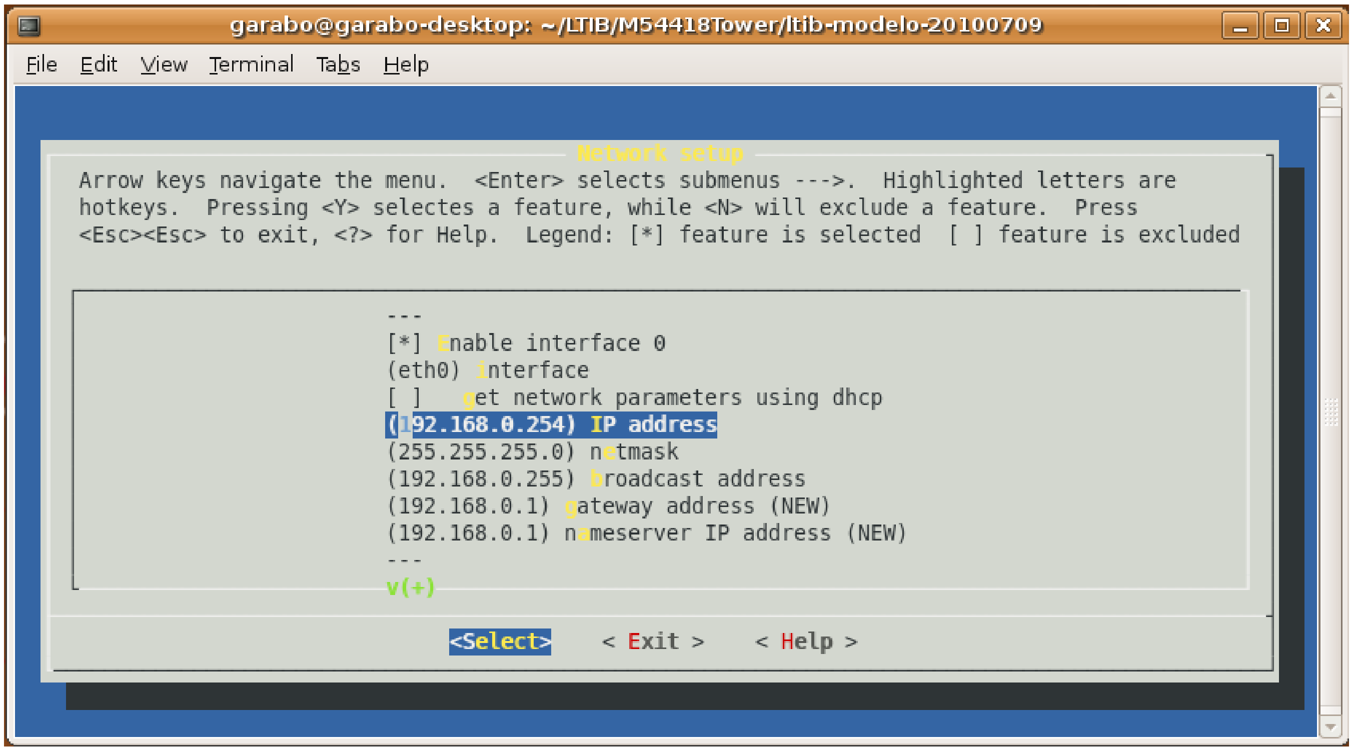

garabo@garabo-desktop: ~/LTIB/M54418Tower/ltib-modelo-20100709
File Edit View Terminal Tabs Help

```
                              Package list
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
  hotkeys.  Pressing <Y> selectes a feature, while <N> will exclude a feature.  Press
  <Esc><Esc> to exit, <?> for Help.  Legend: [*] feature is selected  [ ] feature is excluded

                    --- Platform specific package dependsion
                    [*] l2switch-test
                    --- Common package selection list
                    [ ] alien
                    [ ] Apptrk
                    [ ] asterisk
                    [ ] atk
                    [ ] autoconf
                    [ ] automake
                    [*] alsa-lib
                    v(+)

                 <Select>    < Exit >    < Help >
```

Figure 11. Package list

b. Target system configuration:

Please be sure to set the network environment the first time you build the project. To do this you need to select the board configuration screen:
Target System Configuration ◊ Options ◊ Network Setup ◊
Figure 12 shows the network setup screen.
Please set the correct IP address for the target board. If you connect the target board to your PC directly with a crossover cable, please set the target board IP addresses with the same subnet of the host PC. If you are using a router to connect the target board and the host PC together, you can select “get network parameters using DHCP” to get the IP from the router.

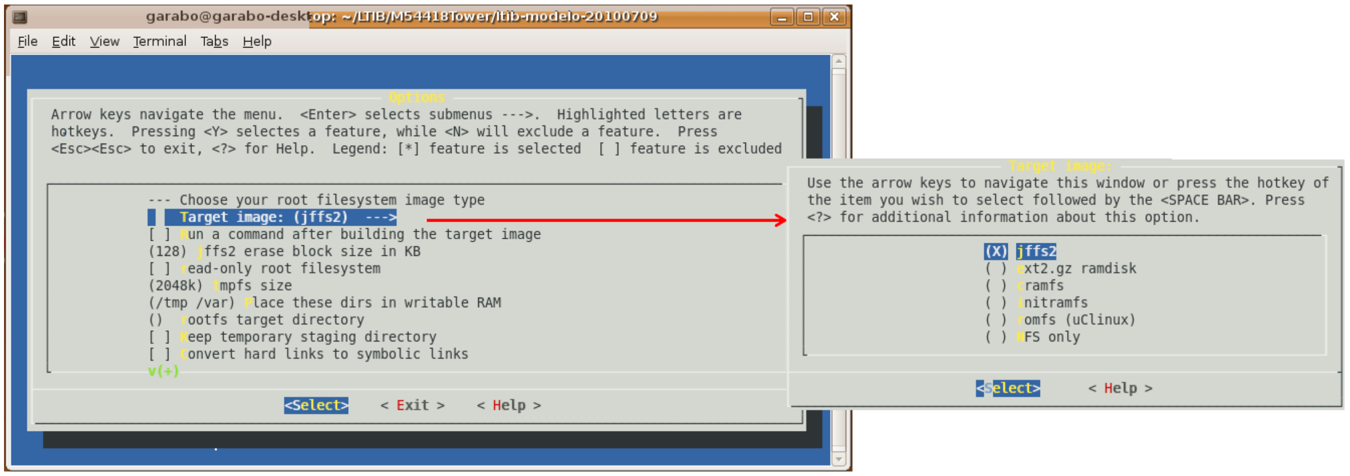


garabo@garabo-desktop: ~/LTIB/M54418Tower/ltib-modelo-20100709
File Edit View Terminal Tabs Help
Network setup

```
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
hotkeys.  Pressing <Y> selectes a feature, while <N> will exclude a feature.  Press
<Esc><Esc> to exit, <?> for Help.  Legend: [*] feature is selected  [ ] feature is excluded

                    ---
                    [*] Enable interface 0
                    (eth0) interface
                    [ ]   get network parameters using dhcp
                    (192.168.0.254) IP address
                    (255.255.255.0) netmask
                    (192.168.0.255) broadcast address
                    (192.168.0.1) gateway address (NEW)
                    (192.168.0.1) nameserver IP address (NEW)
                    ---
                    v(+)

                 <Select>    < Exit >    < Help >
```

Figure 12. Network setup

c. Target image generation:



garabo@garabo-desktop: ~/LTIB/M54418Tower/ltib-modelo-20100709
File Edit View Terminal Tabs Help
Options
Target image:

```
Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are
hotkeys.  Pressing <Y> selectes a feature, while <N> will exclude a feature.  Press
<Esc><Esc> to exit, <?> for Help.  Legend: [*] feature is selected  [ ] feature is excluded

          --- Choose your root filesystem image type
                Target image: (jffs2)  --->
          [ ] Run a command after building the target image
          (128) jffs2 erase block size in KB
          [ ] read-only root filesystem
          (2048k) tmpfs size
          (/tmp /var) Place these dirs in writable RAM
          ()  rootfs target directory
          [ ] Keep temporary staging directory
          [ ] Convert hard links to symbolic links
          v(+)

                 <Select>    < Exit >    < Help >
```

```
Use the arrow keys to navigate this window or press the hotkey of
the item you wish to select followed by the <SPACE BAR>. Press
<?> for additional information about this option.

                  (X) jffs2
                  ( ) ext2.gz ramdisk
                  ( ) cramfs
                  ( ) initramfs
                  ( ) romfs (uClinux)
                  ( ) NFS only

                <Select>      < Help >
```

Figure 13. Target image

You can select the file system type for target image generation. In most cases Network File System (NFS) is appropriate to use in the development stage. Figure 13 shows the Target Image Generation screen.

4. Exit from the top level menu when configuration is done and save the new configuration. LTIB will gather BSP configuration changes from the user. Then you must create and/or update all needed binary RPMs for the target platform, as required by the BSP configuration, by building from original sources and patches. Next, the LTIB creates an RFS tree by installing the binary RPMs of the packages needed for the configuration, and generates all the image files needed for deploying Linux OS to the target board. Once you have built your project you will get the following directory/image files:
   - **rootfs** — Directory, the root file system that will be deployed on your board.

- **rootfs.jffs2** — JFFS2 filesystem file that can be flashed to your board if you selected jffs2 in the target image generation procedure.
- **rootfs/boot/uImage** — Kernel image that can be loaded with CFFlasher. Instructions are provided in Configuring U-boot.

If you want to fully re-configure and re-compile all the packages, you can do the following. (This is generally not necessary.)

a. Clean up all the configure files and objects thoroughly:

```
./ltib -m distclean
```

b. You will be prompted to confirm your choice. Type yes to perform a distclean.
c. Run litb.

```
./ltib
```

More information on LITB can be found in `<install path>/ltib/doc`, or on the web at http://savannah.nongnu.org/projects/ltib.

3 Target deployment

3.1 The Tower kit

The TWR-MCF5441X system is a Freescale Tower compatible system based on ColdFire™ architecture microprocessor MCF5441X.

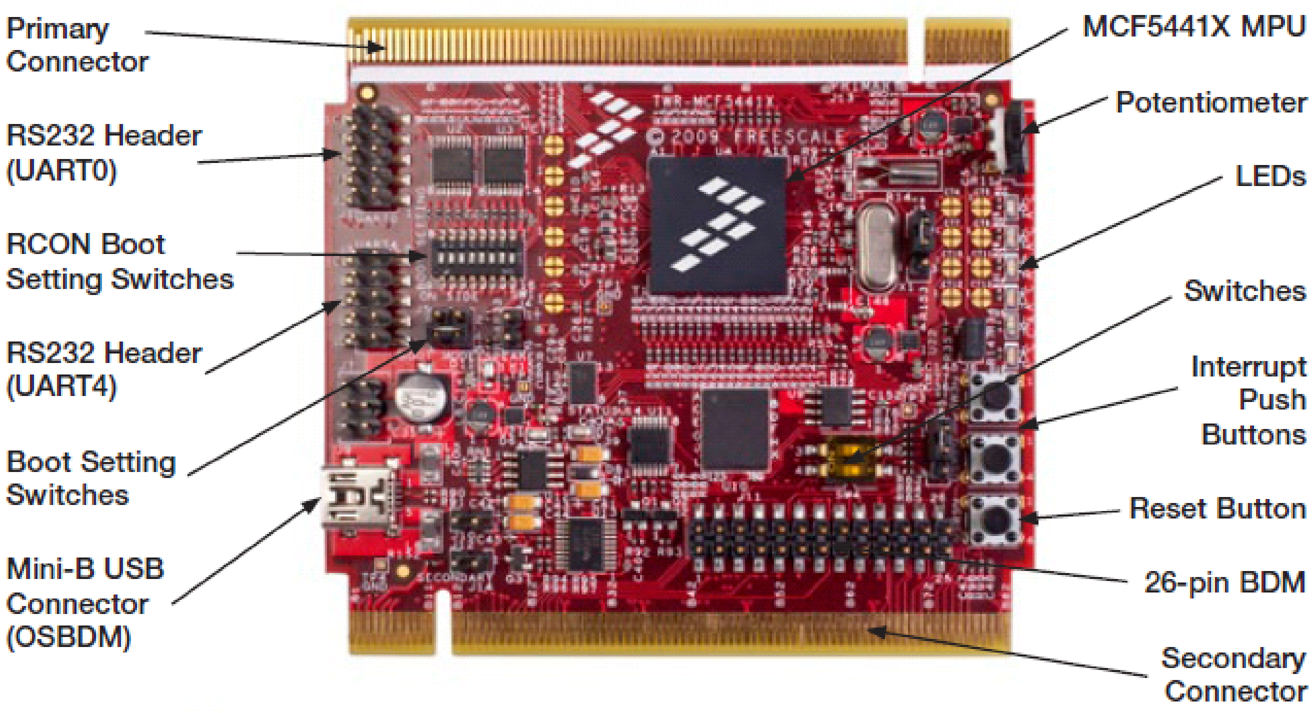


Figure 14. TWR-MCF5441X module

The TWR-MCF5441X module is part of the Freescale Tower System, a modular development platform that enables rapid prototyping and tool re-use through reconfigurable hardware.

The TWR-MCF5441X has three boot mode options:

- Boot with default configuration constants specified in the RCON register
- Boot with NAND/NOR with configuration data specified by the Flexbus FB_AD[7:0] pins
- Boot with configuration data obtained from an external SPI memory through the serial boot facility

The boot modes are determined by the jumper configuration of J5 at reset. Placing a jumper on pins 1–2 of J5 causes BOOTMOD[0] to be low (0). Placing a jumper on pins 3–4 of J5 causes BOOTMOD[1] to be low (0). In Table 2 “ON” implies that the respective jumper is shunted.

Table 1. J5 headers

BOOTMOD[1:0]	J5 3–4	J5 1–2	Description
00	On	On	Boot from Flexbus with default (RCON).
01	On	Off	Override default via data bus (FB_AD[7:0]) — NAND/ FlexBus. This the typical boot mode for the TWR-MCF5441X.
10	Off	On	Override default and boot from serial boot facility with load configuration and optional booting from internal SRAM. If not booting from internal SRAM, serial RCON configuration will decide the boot source at address 0 either from FlexBus or NAND flash memory.

This application uses BOOTMOD 01, which is the typical boot mode for the TWR-MCF5441X.

If the BOOTMOD pins are 01 during reset, the MCF5441x configuration after reset is determined according to the levels driven onto the FB_AD[7:0] pins. On the TWR-MCF5441X, the FB_AD[7:0] pins are actively driven by two 4-bit buffers enabled when the MCF5441x RSTOUT signal is asserted. The values driven by the buffer are set by the SW1 DIP switch settings. For SW1, a value of 0 implies that the DIP is switched “On.”

Table 2. SW1 8-way DIP switch

Override Pins in Reset	Function
SW1–DIP 1	**Boot memory**
0 (Default)	NAND flash
1	FlexBus
SW1–2	**PLL mode**
0	Disabled
1 (Default)	Enabled
SW1–3	**Oscillator mode**
0 (Default)	Crystal oscillator mode
1	Oscillator bypass mode
SW1–4	**FB_ALE select**

Table continues on the next page...

Table 2. SW1 8-way DIP switch (continued)

Override Pins in Reset	Function
0	FB_TS_B
1 (Default)	FB_ALE
SW1–[6:5]	**BOOT port size**
00	32-bit (32-bit muxed address)
01	8-bit (24-bit non-muxed address)
10 (Default)	16-bit (16-bit non-muxed address)
11	16-bit (16-bit non-muxed address)
SW1–[8:7]	**PLL multiplier**
00 (Default)	Fvco = 10 × Fref
01	Fvco = 15 × Fref
10	Fvco = 16 × Fref
11	Fvco = 20 × Fref

For this application you need to use SW1[1:8]={0,1,0,1,1,0,0,0}; this is the default configuration.

3.2 Constructing the Tower kit

Please use these steps to build the Tower:

1. Locate and identify the elevator modules. Each elevator module is identifiable by its four card edge connectors. Each elevator module is either Primary or Secondary. They are identifiable by these characteristics:
   - Primary elevator: written on inside top and denoted by white card edge connectors.
   - Secondary elevator: written on inside bottom.
2. Additional modules. The Linux demonstration requires the use of the TWR-SER2. Please assemble the TWR-MCF5441X and the TWR-SER2 together using the elevator modules.
3. Identify the primary and secondary card edges. For each module, the words Primary and Secondary are written along the card edges. The primary edge is also denoted by a white stripe. Match the white stripe on the edge of each module to any available connector on the primary elevator and plug it in. With the TWR-MCF5441X and the TWR-SER2 modules connected to the primary elevator, carefully attach the secondary elevator onto the secondary card edges of each module.

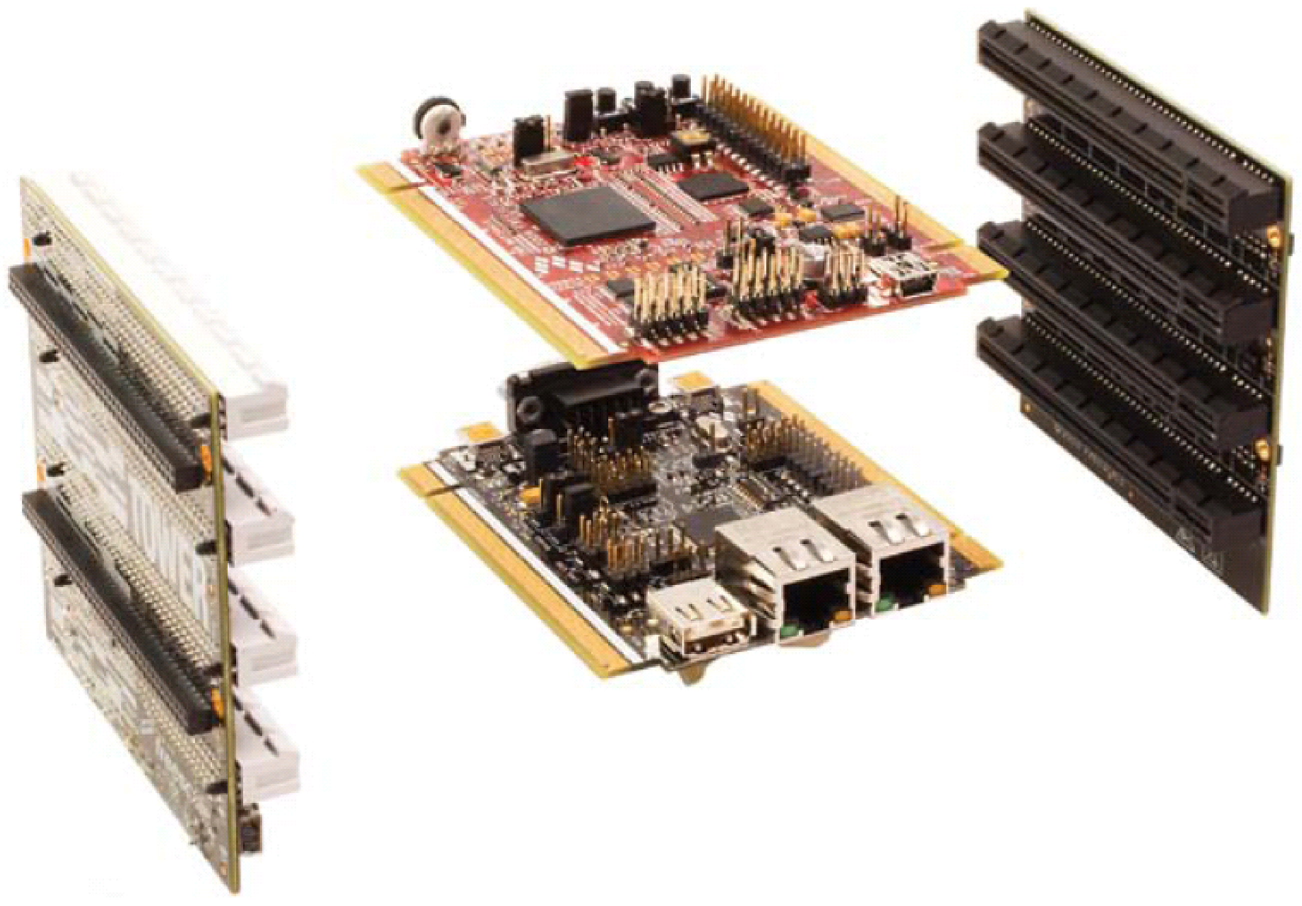


Figure 15. Tower kit

4. Connect the serial cable. Attach the header end of the provided DB9 adapter cable to UART0 (J1) of the TWR-MCF5441X. Attach the DB9 end to a serial cable connected to the host computer. Open a terminal window on the host computer. Use the appropriate COM port to establish a serial connection with the TWR-MCF5441X. Use these connection settings:
    - Bits per second: 115200
    - Data bits: 8
    - Parity: None
    - Stop bit: 1
    - Flow control: None
5. Connect the USB cable. Attach the provided USB A-to-Mini-B cable between the host computer (or a USB wall adapter) and the Mini-B (J9) receptacle on the TWR-MCF5441X. The USB cable will supply power to the TWR-MCF5441X and additional Tower System modules via the TWR-ELEV.

3.3 Programming U-boot, kernel, and root file system

U-Boot is the default bootloader for the board and should be pre-installed. (Don't reinstall U-Boot unless necessary.) LTIB can build several versions of U-Boot from source and the executable built image can be found in `<ltib>/rootfs/boot/u-boot.bin`.

The BSP provides two versions of CFFlasher to reinstall the U-Boot bootloader for MRAM/SBF and NAND flash memory. NAND flash memory is the version for this application.

To reinstall the U-Boot in NAND flash memory, use the CFFlasher-nand.zip tools located in the BSP ISO under the folder (/help/software/CFFlasher). Please make sure to copy all the files of CFFlasher-nand folder to C:\>, as shown in Figure 16 and Figure 17.

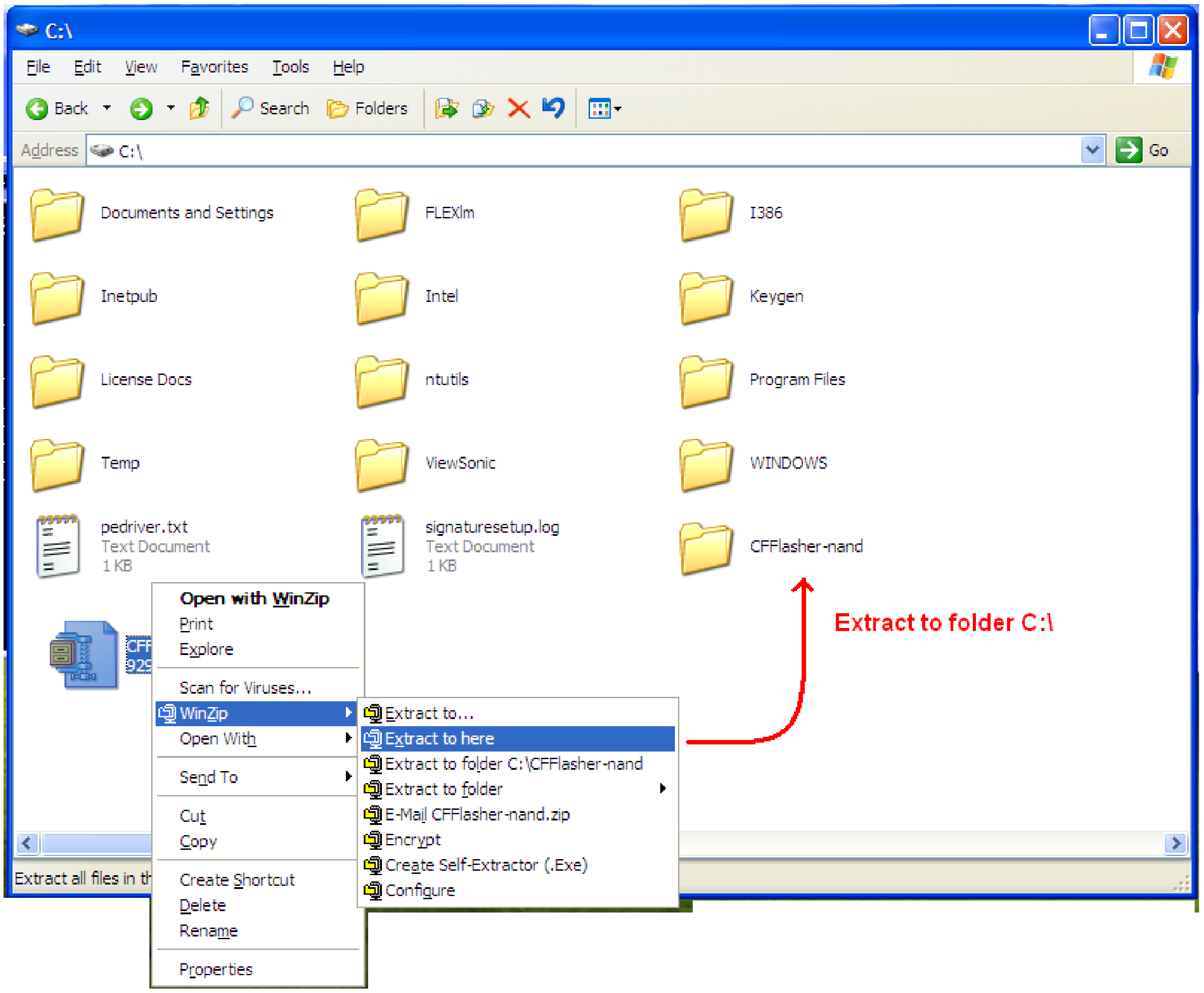


Figure 16. Extract to folder C

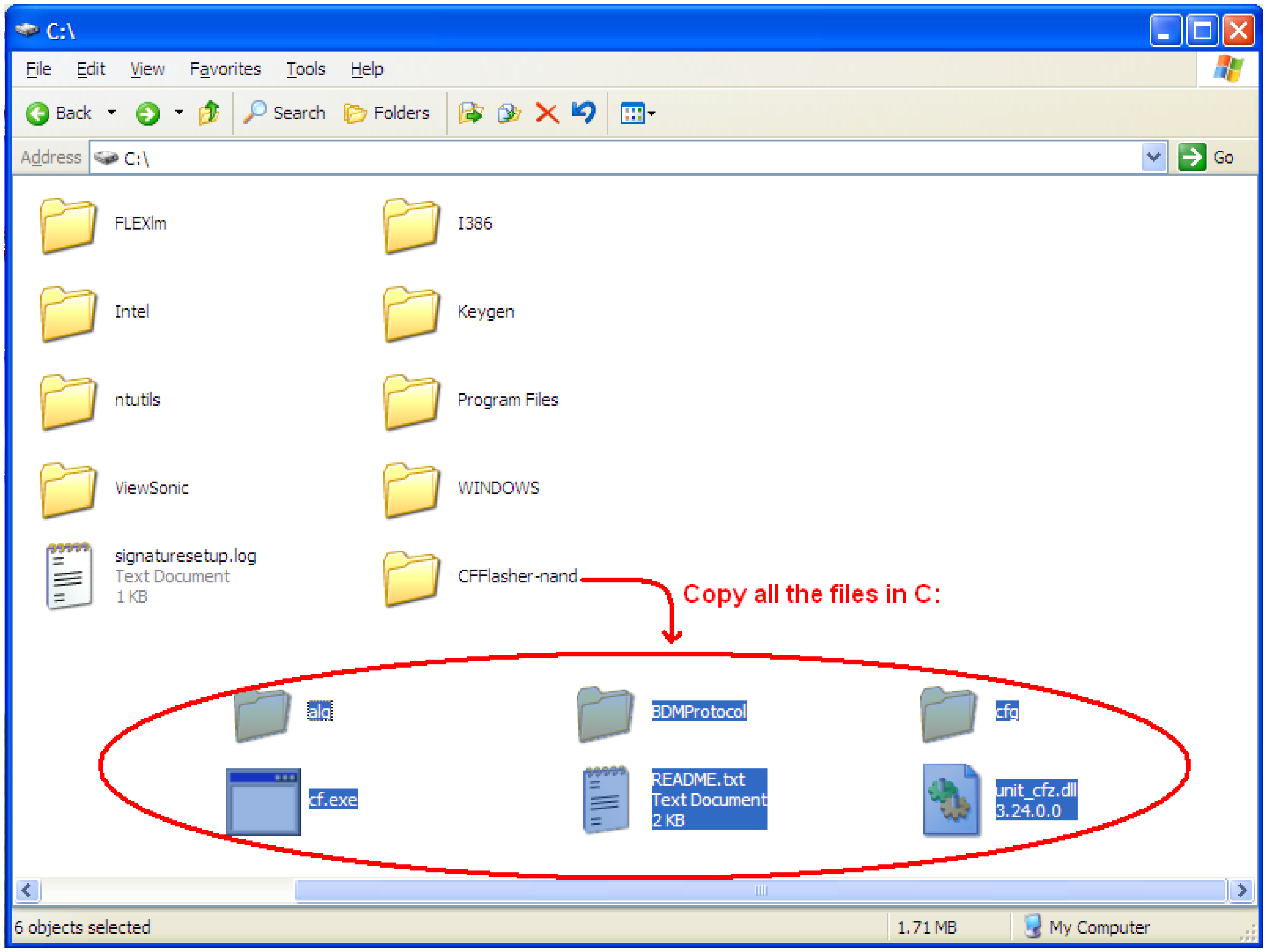


Figure 17. CFFlasher nand files

In addition you need to copy the U-boot.bin file to C:\> (see Figure 18).

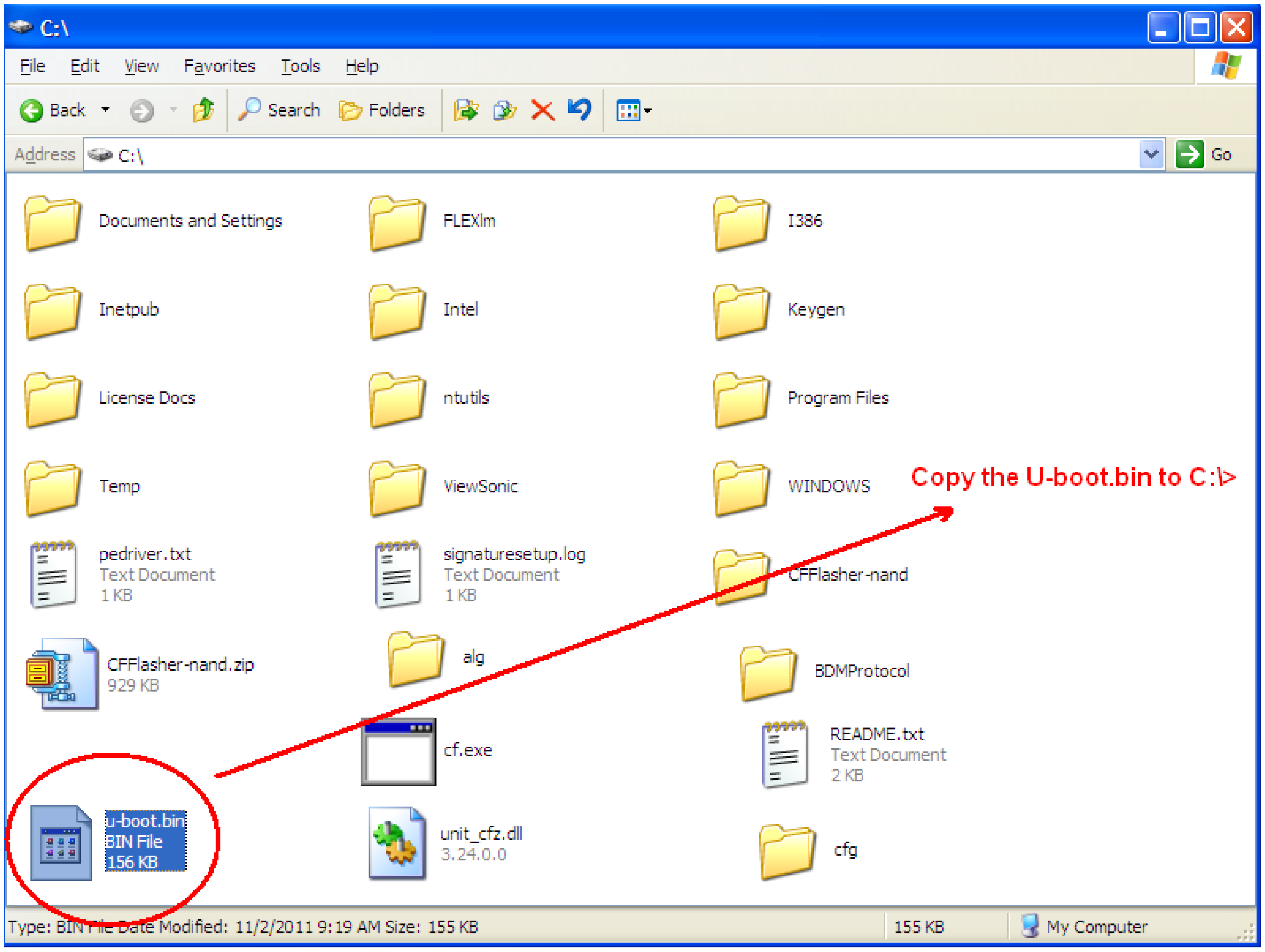

C:\
File Edit View Favorites Tools Help
Back Search Folders
Address C:\ Go
Documents and Settings
FLEXlm
I386
Inetpub
Intel
Keygen
License Docs
ntutils
Program Files
Temp
ViewSonic
WINDOWS
Copy the U-boot.bin to C:\>
pedriver.txt Text Document 1 KB
signaturesetup.log Text Document 1 KB
CFFlasher-nand
CFFlasher-nand.zip 929 KB
alg
BDMProtocol
cf.exe
README.txt Text Document 2 KB
u-boot.bin BIN File 156 KB
unit_cfz.dll 3.24.0.0
cfg
Type: BIN File Date Modified: 11/2/2011 9:19 AM Size: 155 KB
155 KB
My Computer

Figure 18. U-boot

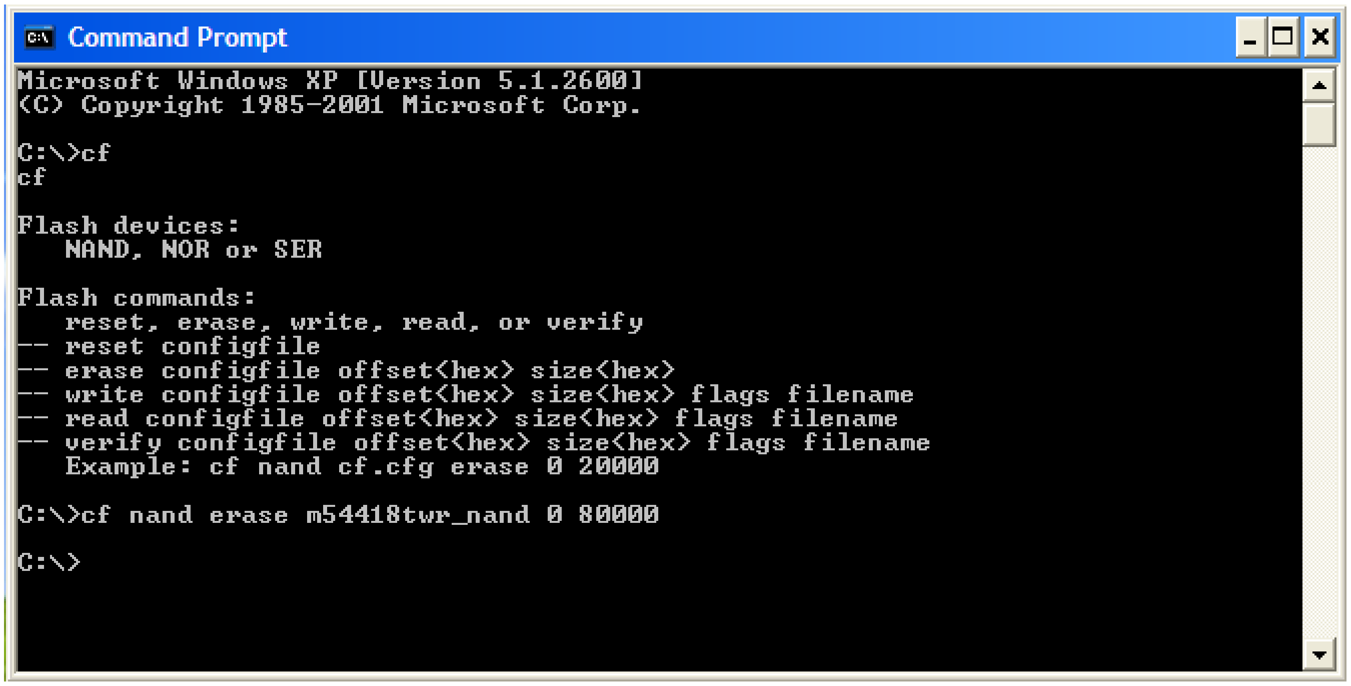
CFFlasher-nand is a DOS version tool. This is a utility to program a NAND flash memory device via BDM interface. The utility program currently only supports NAND interface and it works only under Command prompt. The sequence for programming the NAND is:

1. Erase.
2. Write/verify.

First, the erase feature will erase block(s) and detect bad block(s) at the same time, and store the bad block(s) offset at M54418 internal SRAM. Then, the bad block(s) will be referred by write/verify features. If the first erase offset and size are given, it cannot be used at a later time for write/verify at different offset and size.

To erase at NAND offset 0 and four blocks, size is set to 0x80000. (0x20000 block aligned) Please refer to Figure 19.

```
C:\> cf nand erase m54418twr_nand 0 80000
```



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>cf
cf

Flash devices:
   NAND, NOR or SER

Flash commands:
   reset, erase, write, read, or verify
-- reset configfile
-- erase configfile offset<hex> size<hex>
-- write configfile offset<hex> size<hex> flags filename
-- read configfile offset<hex> size<hex> flags filename
-- verify configfile offset<hex> size<hex> flags filename
   Example: cf nand cf.cfg erase 0 20000

C:\>cf nand erase m54418twr_nand 0 80000

C:\>
```

Figure 19. Erase m54418twr_nand

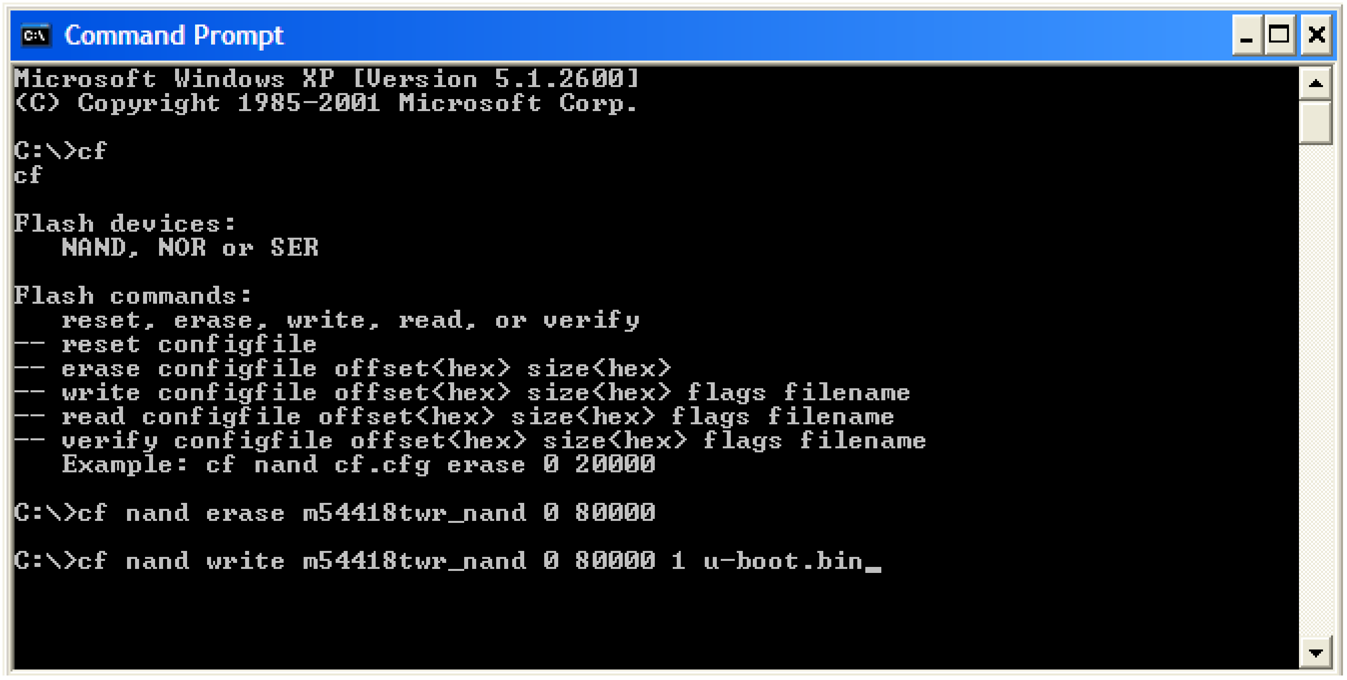
When the actual file size is smaller than a given size (0x80000), the actual file size will be used. If a given size is smaller than the actual file size, the given size will be used. The flag is set to one to indicate to the programmer to program the boot pages (8-bit width bus for the four boot pages). If it is zero, the four pages will become regular 16-bit width bus pages

The bootloader maps system memory as shown below.

```
System memory map:
0x0000_0000 0x0004_0000 MRAM
0x4000_0000 0x5000_0000 SDRAM
0xE000_0000 0xFFFF_FFFF Peripheral bus
NAND Flash map:
U-Boot image 0x00000000 - 0x0007FFFF
U-Boot env 0x00080000 - 0x000807FF
Kernel image 0x00400000 - 0x007FFFFF
Jffs2 image 0x00800000 - 0x10000000
MRAM map:
U-Boot image 0x00000000 - 0x0003EFFF
U-Boot env 0x0003F000 - 0x0003FFFF
SPI serial flash map:
U-Boot image 0x00000000 - 0x0003ffff
U-Boot env 0x40000 - 0x0004ffff
```

Please follow the steps below to write/verify u-boot, kernel, and root file system.

1. Write at NAND the u-boot.bin (see Figure 20).

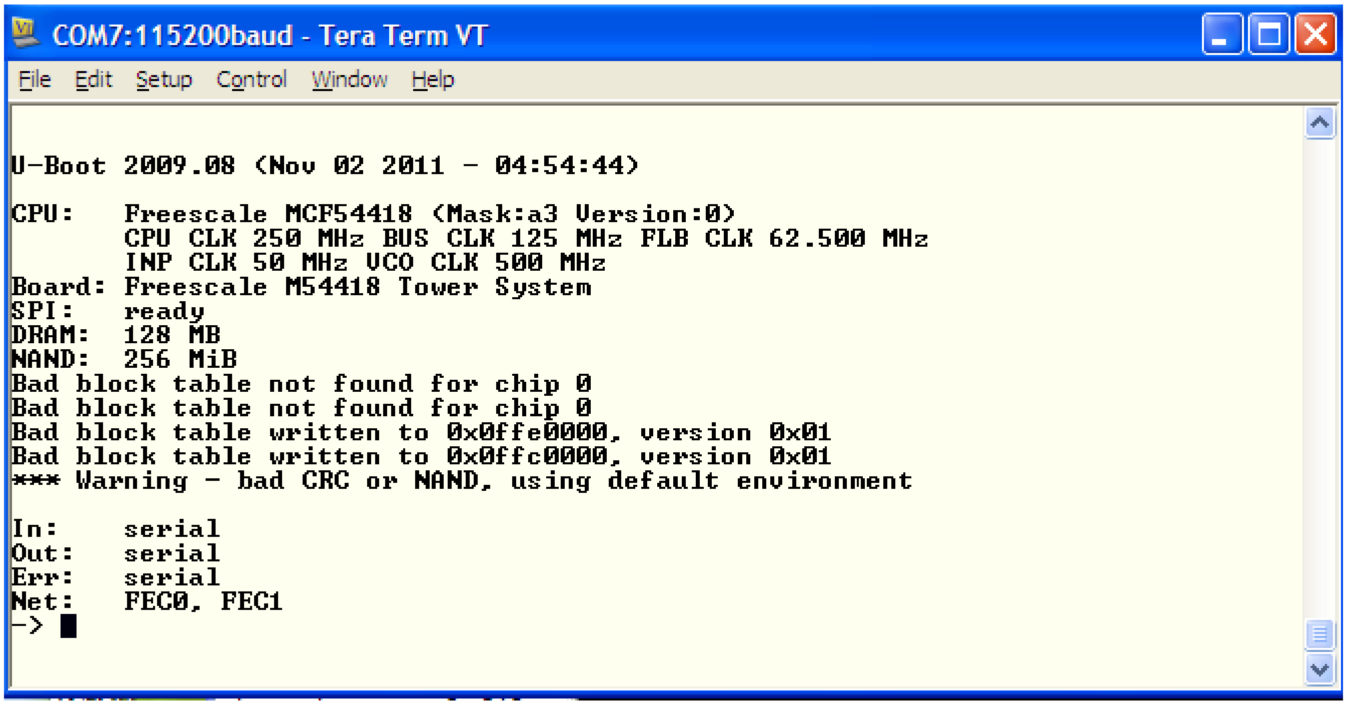


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>cf
cf

Flash devices:
   NAND, NOR or SER

Flash commands:
   reset, erase, write, read, or verify
-- reset configfile
-- erase configfile offset<hex> size<hex>
-- write configfile offset<hex> size<hex> flags filename
-- read configfile offset<hex> size<hex> flags filename
-- verify configfile offset<hex> size<hex> flags filename
   Example: cf nand cf.cfg erase 0 20000

C:\>cf nand erase m54418twr_nand 0 80000

C:\>cf nand write m54418twr_nand 0 80000 1 u-boot.bin_
```

Figure 20. Write the u-boot

2. Open a terminal window on the host computer. Use the appropriate COM port to establish a serial connection with the TWR-MCF5441X . Remember to use these connection settings:
   - Bits per second: 115200
   - Data bits: 8
   - Parity: None
   - Stop bit: 1
   - Flow control: None

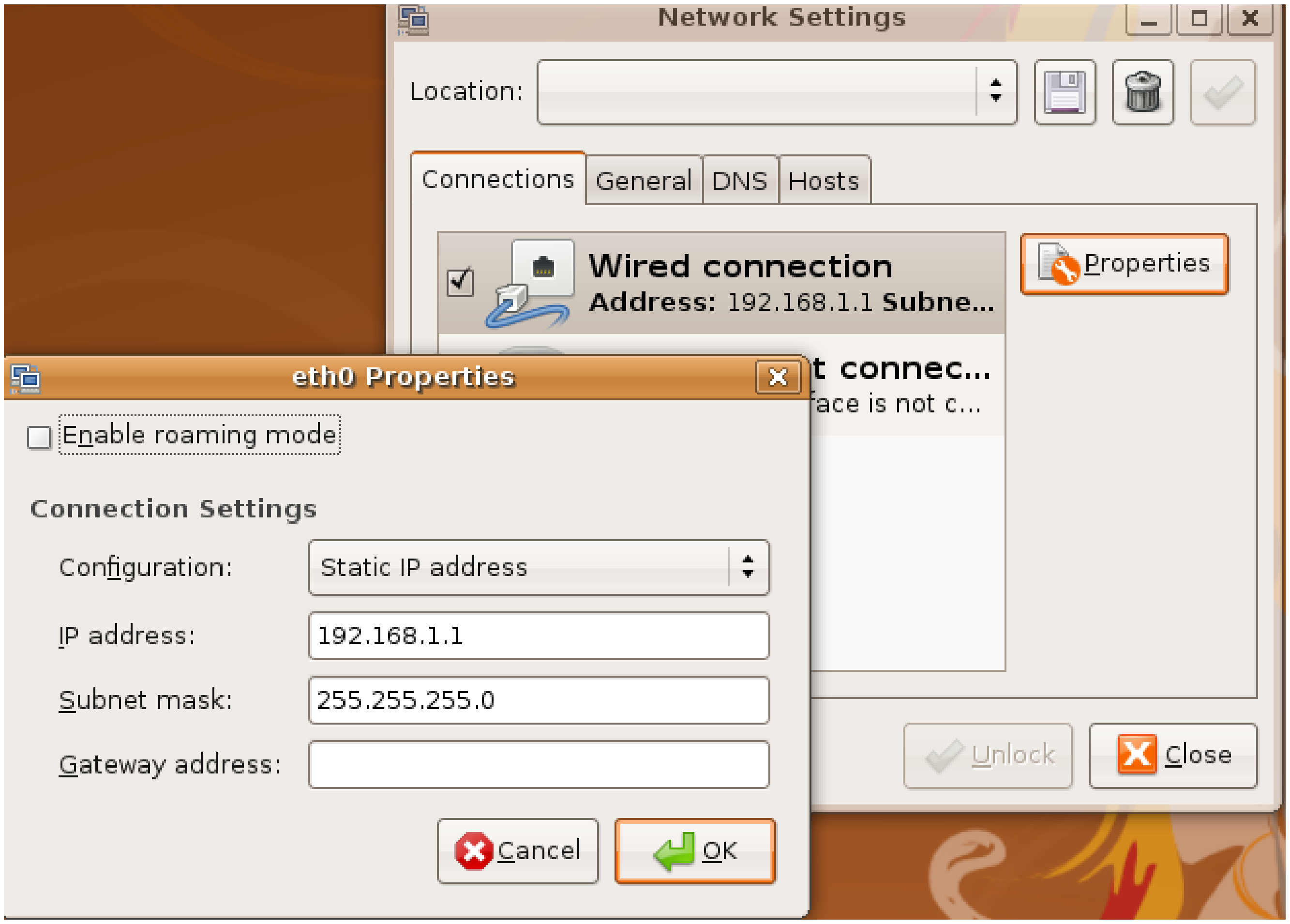
   Press the reset button (SW3) on the TWR-MCF5441X to restart U-Boot. Figure 21 shows the terminal window.


COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
U-Boot 2009.08 (Nov 02 2011 - 04:54:44)

CPU:    Freescale MCF54418 (Mask:a3 Version:0)
        CPU CLK 250 MHz BUS CLK 125 MHz FLB CLK 62.500 MHz
        INP CLK 50 MHz VCO CLK 500 MHz
Board:  Freescale M54418 Tower System
SPI:    ready
DRAM:   128 MB
NAND:   256 MiB
Bad block table not found for chip 0
Bad block table not found for chip 0
Bad block table written to 0x0ffe0000, version 0x01
Bad block table written to 0x0ffc0000, version 0x01
*** Warning - bad CRC or NAND, using default environment

In:     serial
Out:    serial
Err:    serial
Net:    FEC0, FEC1
-> █
```

Figure 21. Terminal window

3. Connect the Ethernet cable to the Linux host computer which had previously run the LTIB. Configure the network settings according to the U-boot settings. In Figure 22 you can see the network properties for this example. Please refer to Configuring U-boot in order to check the U-boot settings in your system.



Enable roaming mode

Connection Settings

Configuration: Static IP address

IP address: 192.168.1.1

Subnet mask: 255.255.255.0

Gateway address:

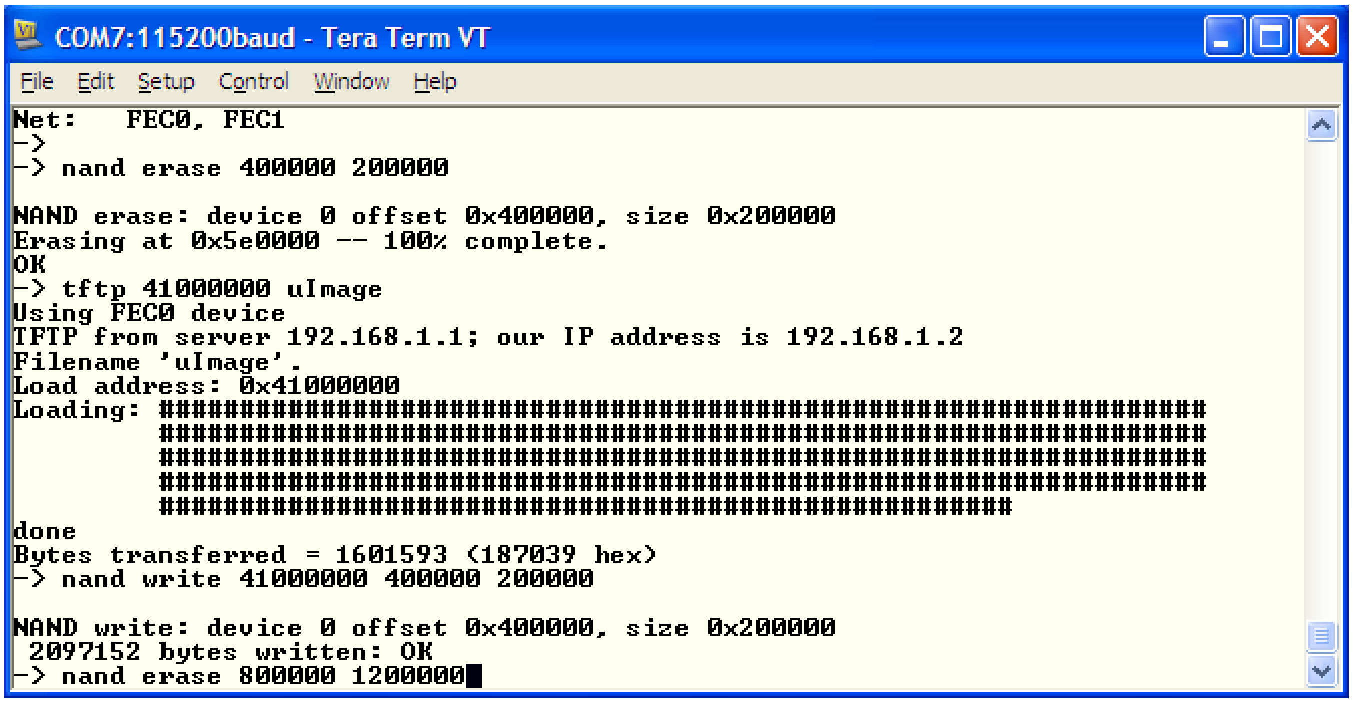
Figure 22. Network settings

4. In the terminal window, erase the NAND flash block and flash kernel image. For an example, see Figure 23.

```
-> nand erase 400000 200000
-> tftp 41000000 uImage
-> nand write 41000000 400000 <uImage_size>  (uImage_size is aligned to 2K
```

For example:

```
-> nand erase 400000 200000
-> tftp 41000000 uImage
-> nand write 41000000 400000 200000
```


COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Net:   FEC0, FEC1
->
-> nand erase 400000 200000

NAND erase: device 0 offset 0x400000, size 0x200000
Erasing at 0x5e0000 -- 100% complete.
OK
-> tftp 41000000 uImage
Using FEC0 device
TFTP from server 192.168.1.1; our IP address is 192.168.1.2
Filename 'uImage'.
Load address: 0x41000000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #####################################################
done
Bytes transferred = 1601593 (187039 hex)
-> nand write 41000000 400000 200000

NAND write: device 0 offset 0x400000, size 0x200000
 2097152 bytes written: OK
-> nand erase 800000 1200000
```

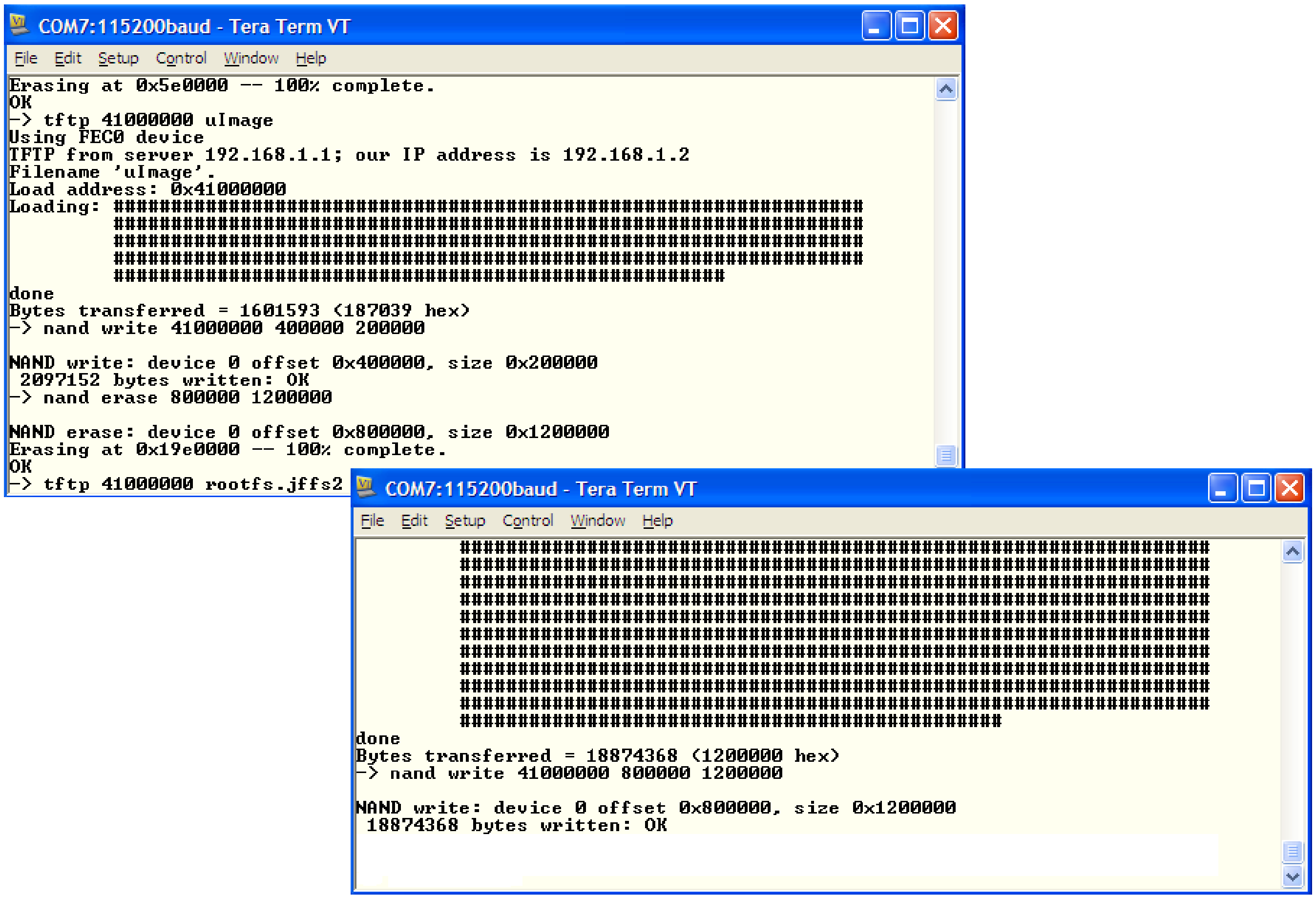
Figure 23. Erase the NAND flash block and flash uImage

5. In the terminal window, erase the NAND flash block and flash root file system image, as seen in Figure 24.

```
-> nand erase 800000 1200000
-> tftp 41000000 rootfs.jffs2
-> nand write 41000000 800000 <rootfs.jffs2_size>
(rootfs.jffs2_size need be aligned to 2K)
```

   For example:

```
-> nand erase 800000 1200000
-> tftp 41000000 rootfs.jffs2
-> nand write 41000000 800000 1200000
```


COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
Erasing at 0x5e0000 -- 100% complete.
OK
-> tftp 41000000 uImage
Using FEC0 device
TFTP from server 192.168.1.1; our IP address is 192.168.1.2
Filename 'uImage'.
Load address: 0x41000000
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #####################################################
done
Bytes transferred = 1601593 (187039 hex)
-> nand write 41000000 400000 200000

NAND write: device 0 offset 0x400000, size 0x200000
 2097152 bytes written: OK
-> nand erase 800000 1200000

NAND erase: device 0 offset 0x800000, size 0x1200000
Erasing at 0x19e0000 -- 100% complete.
OK
-> tftp 41000000 rootfs.jffs2
```

COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         ###############################################
done
Bytes transferred = 18874368 (1200000 hex)
-> nand write 41000000 800000 1200000

NAND write: device 0 offset 0x800000, size 0x1200000
 18874368 bytes written: OK
```

Figure 24. Erase the NAND flash block and flash rootfs

3.4 Configuring U-boot

1. At the U-Boot "→" prompt, use the printenv command to see the current U-Boot configuration. This example shows settings for a rootfs.jffs2 root filesystem deploy.

```
printenv
bootargs=root=/dev/mtdblock2 rw rootfstype=jffs2 mtdparts=NAND:1M(u-boot)ro,
7M(kernel)ro,-(jffs2) console=ttyS0,115200
bootdelay=2
baudrate=115200
ethaddr=00:e0:0c:bc:e5:60
eth1addr=00:e0:0c:bc:e5:61
hostname=M54418TWR
netdev=eth0
inpclk=50000000
loadaddr=0x40010000
u-boot=u-boot.bin
load=tftp ${loadaddr} ${u-boot};
upd=run load; run prog
prog=nand device 0;nand erase 0 40000;nb_update ${loadaddr} ${filesize};save
stdin=serial
stdout=serial
stderr=serial
ethact=FEC0
mem=129024k
filesize=1200000
fileaddr=41000000
```

```
gatewayip=192.168.1.1
netmask=255.255.255.0
ipaddr=192.168.1.2
serverip=192.168.1.1
bootcmd=nand read 0x42000000 0x400000 200000;bootm 42000000

Environment size: 662/131068 bytes
```

2. You can customize U-Boot for your system. For example, to change the hostname type:

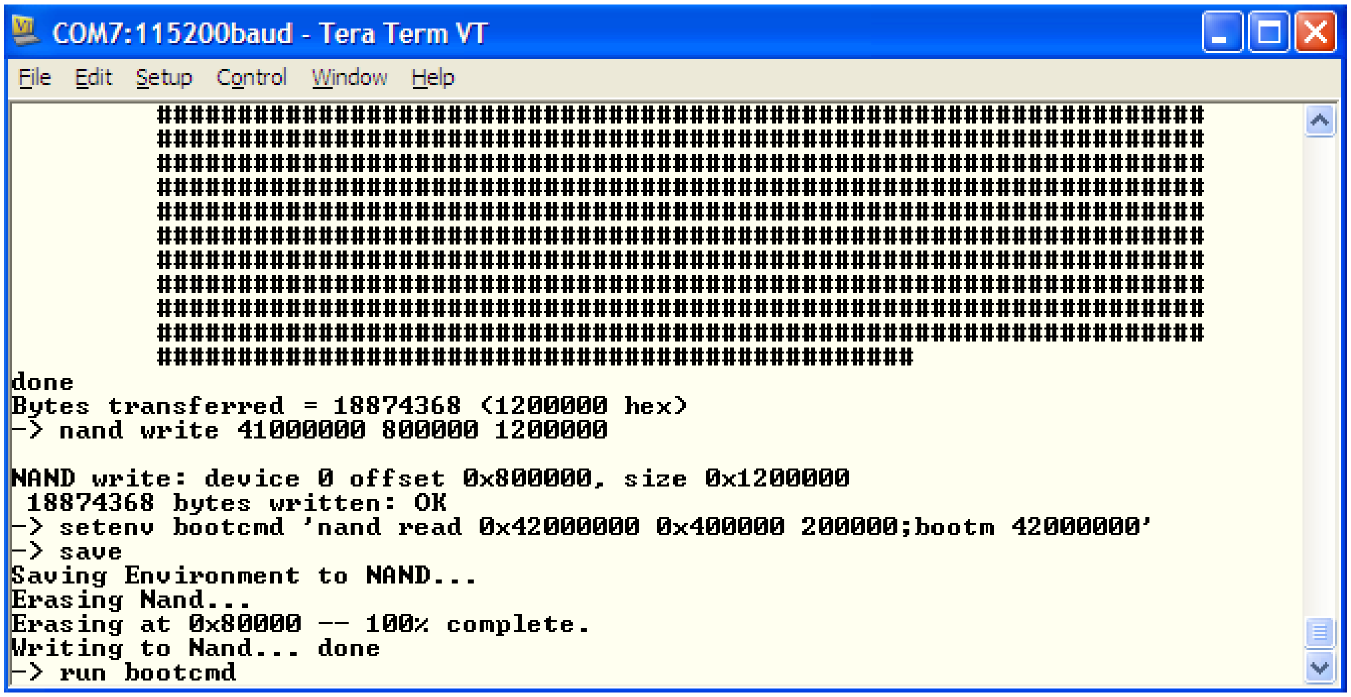
```
setenv hostname <value>
save
```

3. You'll probably need to set the bootargs variable the first time you start your target.

3.5 Running

Once the U-boot, uImage, and rootfs are programmed to boot the kernel, write the next lines in the command window (Figure 25).

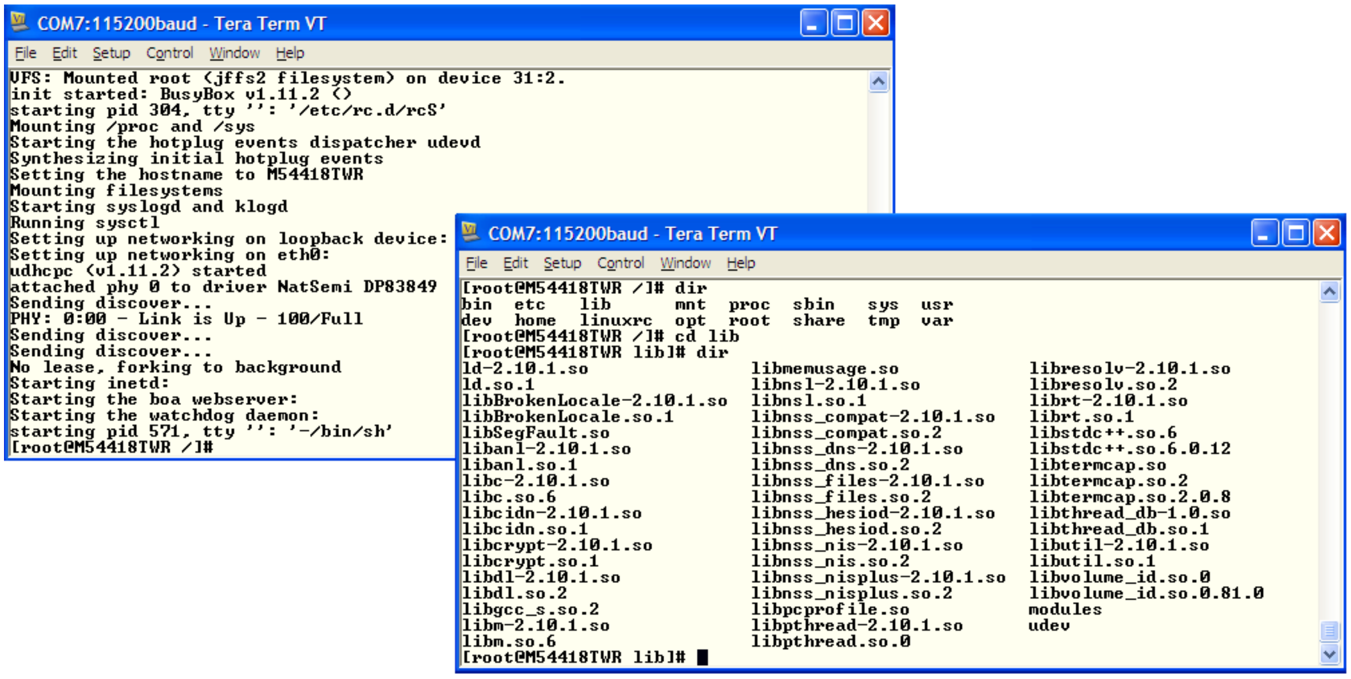
```
->setenv bootcmd 'nand read 0x42000000 0x400000 200000;bootm
42000000'
-> save
-> run bootcmd
```


COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          #######################################################################
          ###################################################
done
Bytes transferred = 18874368 (1200000 hex)
-> nand write 41000000 800000 1200000

NAND write: device 0 offset 0x800000, size 0x1200000
 18874368 bytes written: OK
-> setenv bootcmd 'nand read 0x42000000 0x400000 200000;bootm 42000000'
-> save
Saving Environment to NAND...
Erasing Nand...
Erasing at 0x80000 -- 100% complete.
Writing to Nand... done
-> run bootcmd
```

Figure 25. Boot the kernel

Several lines will be displayed in the terminal. Once the Linux boot has completed, the user is presented with a Linux prompt. At the prompt use the “help” command to explore available commands.


COM7:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
VFS: Mounted root (jffs2 filesystem) on device 31:2.
init started: BusyBox v1.11.2 ()
starting pid 304, tty '': '/etc/rc.d/rcS'
Mounting /proc and /sys
Starting the hotplug events dispatcher udevd
Synthesizing initial hotplug events
Setting the hostname to M54418TWR
Mounting filesystems
Starting syslogd and klogd
Running sysctl
Setting up networking on loopback device:
Setting up networking on eth0:
udhcpc (v1.11.2) started
attached phy 0 to driver NatSemi DP83849
Sending discover...
PHY: 0:00 - Link is Up - 100/Full
Sending discover...
Sending discover...
No lease, forking to background
Starting inetd:
Starting the boa webserver:
Starting the watchdog daemon:
starting pid 571, tty '': '-/bin/sh'
[root@M54418TWR /]#
```

```
[root@M54418TWR /]# dir
bin  etc   lib      mnt  proc  sbin   sys  usr
dev  home  linuxrc  opt  root  share  tmp  var
[root@M54418TWR /]# cd lib
[root@M54418TWR lib]# dir
ld-2.10.1.so               libmemusage.so            libresolv-2.10.1.so
ld.so.1                    libnsl-2.10.1.so          libresolv.so.2
libBrokenLocale-2.10.1.so  libnsl.so.1               librt-2.10.1.so
libBrokenLocale.so.1       libnss_compat-2.10.1.so   librt.so.1
libSegFault.so             libnss_compat.so.2        libstdc++.so.6
libanl-2.10.1.so           libnss_dns-2.10.1.so      libstdc++.so.6.0.12
libanl.so.1                libnss_dns.so.2           libtermcap.so
libc-2.10.1.so             libnss_files-2.10.1.so    libtermcap.so.2
libc.so.6                  libnss_files.so.2         libtermcap.so.2.0.8
libcidn-2.10.1.so          libnss_hesiod-2.10.1.so   libthread_db-1.0.so
libcidn.so.1               libnss_hesiod.so.2        libthread_db.so.1
libcrypt-2.10.1.so         libnss_nis-2.10.1.so      libutil-2.10.1.so
libcrypt.so.1              libnss_nis.so.2           libutil.so.1
libdl-2.10.1.so            libnss_nisplus-2.10.1.so  libvolume_id.so.0
libdl.so.2                 libnss_nisplus.so.2       libvolume_id.so.0.81.0
libgcc_s.so.2              libpcprofile.so           modules
libm-2.10.1.so             libpthread-2.10.1.so      udev
libm.so.6                  libpthread.so.0
[root@M54418TWR lib]# █
```

Figure 26. Linux prompt

4 Conclusions

This application note shows that by installing the right packages and executing the right commands, LTIB can build/compile the image without errors, and set up the NFS and TFTP to start the development of Linux on the ColdFire MCF54418Tower board.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN4426
Rev. 0, December 2011