BSP Targeting the Freescale ColdFire MCF54418Tower

User Manual

Rev. 0.2

03/2011

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

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 Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Learn More: For more information about Freescale products, please visit www.freescale.com.

Content

About This Manual

This user manual provides the information on the basic features supported by the BSP and the instructions about how to accomplish these tasks:

- Install the BSP on a host development system.
- Run Linux Target Image Builder (LTIB) to build target images.
- Deploy built images to the ColdFire MCF54418Tower board.
- Boot Linux on the ColdFire MCF54418Tower board.

Audience

This document is addressed to developers who want to take advantage of the Freescale Linux Target Image Builder (LTIB) for the ColdFire MCF54418Tower Board Support Package (BSP).

Organization

This document is organized into 4 chapters.

Chapter 1	Provides an introduction to the ColdFire MCF54418Tower BSP.
Chapter 2	Provides basic information on LTIB
Chapter 3	Provides important target set-up information
Chapter 4	Provides host and target-specific build and deployment information

Conventions

This document uses the following notational conventions:

- Courier monospaced type indicates commands, command parameters, code examples, expressions, data types, and directives.
- Italic type indicates replaceable command parameters.
- All source code examples are in C.

Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

BSP	Board Support Package
EVB	Evaluation Board
LTIB	Linux Target Image Builder
NFS	Network File System
TFTP	Trivial File Transfer Protocol

Chapter 1
LTIB Basics

1.1 LTIB Overview

The Linux Target Image Builder (LTIB) is a framework tool used to manage, configure, extend and create Linux software elements to easily build a Linux target image, a root filesystem. LTIB runs on an x86 PC with the Linux OS.

This Board Support Package (BSP) operates with LTIB running on a host development system with the following:

- Ethernet card
- Serial port
- 1 GB of free disk space required
- NFS Server
- TFTP Server
- rsync

The following packages are required on the host for LTIB:

Around 1GB of free disk space is recommended for full BSP builds.

package	version	comment
perl	>= 5.6.1	to run the LTIB script
glibc	>= 2.2.x	to build/run host packages
glibc-headers	>= 2.2.x	to build/run host packages
glibc-devel	>= 2.2.x	to build/run host packages
binutils	>= 2.11.93	to build host packages
libstdc++	any	to build rpm-fs host package
libstdc++-devel	any	to build rpm-fs host package
gcc	>= 2.96	to build host packages
gcc-c++	>= 2.26	to build rpm-fs host package
sudo	any	to run the 'rpm install' phase on each package

zlib	any	to build rpm-fs and mtd-utils host packages
zlib-devel	any	to build rpm-fs and mtd-utils host packages
rpm	any	to build initial rpm-fs host package
rpm-build	any	to build initial rpm-fs host package
wget	any	to download packages/patches on demand
ncurses	>= 5.1	to build lkc (config language) host package
ncurses-devel	>= 5.1	to build lkc (config language) host package
m4	any	may be needed by bison
bison	any	to build lkc (config language) host package
flex	any	Not required we install: for host lkc
texinfo	any	to build genext2fs host package (requires ncurses-deve)
gettext	any	genext2fs target package
autoconf	>= 2.54	Not required we install: automake target package
libtool	>= 1.4.2	Not required we install: libusb target package

NOTE: Be aware that some host side packages may not function properly on every Linux distribution.

The following platforms were tested with LTIB.

- Redhat: 7.3, 8.0, 9.0, RHEL 4, 5.3 Enterprise Server
- Fedora Core: 1, 2, 3, 4, 5, 8, 10
- Debian: 3.1r0, 4.0
- OpenSuse: 8.2, 9.1, 9.2, 10.0, 10.2, 10.3
- Ubuntu 6.10, 7.04, 7.10, 8.04, 8.10, 9.04, 9.10, 10.04(32bit)

The released BSP contains 1 DVD-ROM files:

- DVD –M54418Tower_Linux_BSP_20110422_ltib.iso: It contains the LTIB based BSP to generate U-Boot, kernel.

1.2 BSP Overview

The ColdFire MCF54418Tower BSP is designed for the use with LTIB. Once the BSP is installed and running with its basic configuration, you can use LTIB to customize your project.

The BSP components provide the tools, device drivers, and additional features needed for your embedded Linux project.

Linux 2.6.29 kernel

- Linux-2.6.29 kernel with ColdFire (MMU) support
- 10 UART support
- 2 FEC port support
- L2 switch configuration command support
- L2 switch RSTP configuration commands support
- IEEE1588 driver support
- LCD fb driver support
- DSPI support
- I2C support
- USB(host and OTG) support
- eSDHC support
- eDMA support
- NAND MTD support
- MRAM MTD support
- RTC support
- Watchdog support
- CAU crypto support
- DMA timer support
- FlexCAN driver support
- MMC over SPI support

U-Boot bootloader – v2009.08 + Freescale patches.

- Booting from NAND support
- Booting from MRAM support
- Booting from SPI flash support
- UART support
- NAND support
- DDR2 support
- 2 FEC support
- Robust RTC support

GNU gcc 4.4.1, glibc 2.11.217, binutils-2.19.51, and elf2flt

Chapter 2
LTIB Basics

2.1 Installing the BSP

Please follow the steps below to install LTIB on your host machine.

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

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

2. As a non-root user, install the LTIB:

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

You will be prompted to input the desired LTIB install path. Be sure the user has the correct permissions for the install path.

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.

2.2 Running LTIB

To run LTIB, change to the directory into which you have installed it and run `./ltib`.

```
cd <install_path>/ltib

./ltib
```

The first time LTIB runs on your machine, a number of host packages are built and installed that support LTIB. This may take a few minutes.

To modify the project configuration, simply run:

`./ltib --configure` (or `-c`; type `--help` to see configuration options)

This will re-prompt you for the platform/board configuration. In the board configuration screens, change settings and select packages as appropriate. When you exit the configuration screen, your target image will be adjusted accordingly.

Once you build your project, you will get the following directory/image files:

- **rootfs/** – directory, the root file system that will be deployed to your evaluation board.

- **rootfs/boot/vmlinux.bin** – kernel image for loading into target memory
- **rootfs/boot/uImage** – kernel for target deploy option selected in Target Image Generation.
- **rootfs.jffs2** – if jffs2 option selected in Target Image Generation.

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

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

   ```
   ./ltib –m distclean
   ```

2. You will be prompted to confirm your choice. Type `yes` to perform a `distclean`.

3. 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

Chapter 3
Target Configuration

3.1 Supported Target Revisions

The target system is the ColdFire MCF54418 Tower system. The initial BSP configuration is for a jffs2 mounted root filesystem.

This BSP is known to work on the following board: TWR-M54418 MPU module; TWR-SER2 module; TWR-MEM module, TWR-SER1 module and the elevator cards.

3.2 Constructing the Tower Kit

Please refer to the M5441x Kit Quick Start Guide on http://cache.freescale.com/files/netcomm/doc/user_guide/TWRMCF5441XQSG.pdf?fpsp=1 to set up the hardware environment and the board settings.

Please note that different hardware settings need different software (U-Boot, kernel and ltib) configurations. Please try to deselect the functions which are not included in your hardware system to avoid any possible system exception.

For example, When TWR-M54418 MPU module operates as an individual card with onboard 25MHz reference clock, please select the “NAND boot-25M clocksource” for U-Boot image building in LTIB, and disable the devices drivers which are not included on the TWR-M54418 MPU module.

If the external 50M clock source is used, please select corresponding “-50M clocksource” U-Boot options in LTIB to build the U-Boot image.

3.3 Target System Memory Map

After system startup, the boot loader 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
```

Chapter 4
Target Deployment

4.1 Host Set-up

Host setup is critical for your BSP to function. The host must be running TFTP and NFS in order for deployment to work. The following instructions are generic. Your system may be different and the commands should be adjusted accordingly.

1. Turn off firewall for TFTP and NFS to work. `iptables -F` or type “`setup`” at the command line.

2. Install tftp-server.

3. Install nfs-server.

4. create the tftpboot directory: `makdir /tftpboot`

5. Link rootfs to an exportable directory once you have built your project.

```
ln -s <install_path>/ltib/rootfs /tftpboot/ltib
```

6. Copy kernel, bootloader, and flash filesystem image to the /tftpboot directory.

```
cp <install_path>/ltib/rootfs/boot/uImage /tftpboot

cp <install_path>/ltib/<flashfs>/tftpboot

cp <install_path>/ltib/u-boot.bin /tftpboot
```

7. Edit /etc/exports and add the following line:

```
      /tftpboot/ltib/ <target board IP> (rw, no_root_squash, async)
```

8. Edit /etc/xinetd.d/tftp to enable tftp like this:

```
      {
                  disable = no
                  socket_type = dgram
                  protocol = udp
                  wait = yes
                  user = root
                  server = /usr/sbin/in.tftpd
                  server_args = /tftpboot
      }
```

9. Restart the NFS and TFTP servers on your host.

```
      /etc/init.d/xinetd restart
```

```
/etc/init.d/nfsserver restart
```

10. Connect board to the network

11. Connect the target to the host via a serial connection

12. Start `minicom` and set it up to talk to the evaluation board

    - Serial Setup:

        - Select correct serial device

        - Hardware & Software Flow control = No

        - Bps = 115200

        - Parity = None

        - Data bits = 8

        - Stop bits = 1

    - Modem & dialing: Delete text for the following: Init String, Reset String, Hang-up String, No flow control

13. Power on board and see the console prompt

4.2 Programming U-Boot

U-Boot is the default bootloader for 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.

This BSP provides three versions of loading location support - NAND Flash (TWR-M54418 MPU module), Atmel Serial Flash (TWR-MEM module), and MRAM (TWR-MEM module). Please refer to M5441x Kit Quick Start Guide on http://cache.freescale.com/files/netcomm/doc/user_guide/TWRMCF5441XQSG.pdf?fpsp=1 for the hardware settings for different board booting locations.

If booting from MRAM is supported, the NAND boot image and serial flash boot image can be updated through MRAM boot.

This BSP also provides tools for programming MRAM, NAND and serial flash separately.

1.1.1 Upgrading U-Boot from MRAM

1.1.1.1 Upgrading U-Boot in MRAM

Copy the U-Boot image for MRAM boot from <ltib>/rootfs/boot/u-boot.bin to your TFTP server directory and set correct U-Boot environment then:

```
      -> set u-boot u-boot.bin
      -> run upd
```

1.1.1.2 Upgrading U-Boot in NAND

Copy the U-Boot image for NAND boot from <ltib>/ rootfs/boot/u-boot.bin to your TFTP server directory, set correct U-Boot environment and then:

```
      -> tftp 41000000 u-boot.bin
      -> nand erase 0 40000
      -> nb_update 41000000 ${filesize}
      -> save
```

If there is already U-Boot image in NAND and want to upgrade the U-Boot version, please also refer to the above commands to update the U-Boot image in NAND.

1.1.1.3 Upgrading U-Boot in serial flash

For booting from serial flash, please make sure to connect 1-2 for J4 and 2-3 for J14 on TWR-MEM module.

Copy the U-Boot image for serial flash boot from <ltib>/ rootfs/boot/u-boot.bin and the SBF header file from the BSP image folder to your TFTP server directory, set the correct U-Boot environment and then:

```
      -> tftp 41000000 sbfhdr.54418
      -> tftp 41000007 u-boot.bin
      -> sf probe 0:1 1000000 3
      -> sf erase 0 100000
      -> sf write 41000000 0 ${filesize}
      -> save
```

If there is already U-Boot image in serial flash and want to upgrade the U-Boot version, please also refer to the above commands to update the U-Boot image in serial flash.

After programming the image to serial flash, disconnect 3-4 for J5 on TWR-M54418 MPU module, reset the system, then the board will boot from serial flash.

1.1.2 Programming U-Boot by CFFlasher tools

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

To reinstall the U-Boot in MRAM and SBF, use the CFFlasher-sbf-mram.zip tools in this BSP ISO (/help/software/CFFlasher). For MRAM program, select target configuration M5418TWR_mem,For SBF (serial boot flash), select target configuration M5418TWR_SBF.

For serial flash programming, set the board as non-SBF mode (connect 3-4 for J5 on TWR-M54418 MPU module). Please also make sure to connect 1-2 for J4 and 2-3 for J14 on TWR-MEM module.

The u-boot.bin built by LTIB cannot be used to program to serial flash directly, a 7-bytes header needs to be added to the beginner of the u-boot.bin. Please refer to following commands to add the header to u-boot.bin, the header file can be found in this BSP ISO (images/sbfhdr.54418).

```
cat sbfhdr.54418 > u-boot.sbf.bin
cat u-boot.bin >> u-boot.sbf.bin
```

To reinstall the U-Boot in NAND flash, use the CFFlasher-nand.zip tools in this BSP ISO (/help/software /CFFlasher). This is a DOS version tool, so refer to the readme in the tools to program the U-Boot to the NAND Flash. Please make sure to set SW1.1 to ON for TWR-M54418 MPU module.

4.3 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.

   ```
   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
   ipaddr=192.168.1.2
   serverip=192.168.1.1
   gatewayip=192.168.1.1
   netmask=255.255.255.0
   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

   Environment size: 567/131068 bytes
   ```

2. 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.

4.4 Development (NFS/Network) Deploy

1. Copy the kernel image from `<install_path>/ltib/rootfs/boot/uImage` to the `/tftpboot` directory.

2. Using the `printenv` command to verify all the network settings. See the "Configuring U-Boot" section for more information.

3. Set the kernel command line for an NFS deploy. (This can also be set with a kernel configuration option "kernel hacking/compiled in kernel boot parameters" and then rebuild the kernel, but it is recommended to use the U-Boot setting.)

```
-> set bootargs root=/dev/nfs rw nfsroot=<Host
IP>:/tftpboot/ltib/rootfs ip=<Target IP>:<Host IP>:<Gateway IP>:
<Netmask>::eth0:off
-> save
```

4. Boot the kernel

```
-> tftp 0x41000000 uImage
-> bootm 0x41000000
```

4.5 Production (Jffs2/Flash) Deploy

1. Select the jffs2 option under Target Image Generation in LTIB. (Make sure the "read-only root filesystem" option is not selected under "Target Image Generation", the block size is 128 and the page size is 2.) Copy the rootfs.jffs2 file to `/tftpboot/`.

```
# cp rootfs.jffs2 /tftpboot/
```

2. Copy the uImage created kernel image from <install_path>/ltib/rootfs/boot/uImage to the /tftpboot directory.

3. Erase the NAND flash block and flash kernel image.

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

4. Erase the NAND flash block and flash rootfs image.

```
-> nand erase 800000 F800000
-> tftp 41000000 rootfs.jffs2
-> nand write 41000000 800000 <rootfs.jffs2_size>
(rootfs.jffs2_size need be aligned to 2K)
```

5. Set the kernel command line for a flash/jffs2 deploy. (This can also be set with a kernel configuration option "kernel hacking/compiled in kernel parameters" and then rebuild the kernel, but it is recommended to use the U-Boot setting.)

```
-> set bootargs root=/dev/mtdblock2 rw rootfstype=jffs2
mtdparts=NAND:1M(u-boot)ro,7M(kernel)ro,-(jffs2)
console=ttyS0,115200
```

6. Boot kernel

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