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.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2008. All rights reserved.
# Contents

About This Book ........................................................................................................................................ vi
  Audience ........................................................................................................................................ vi
  Organization ....................................................................................................................................... vi
  References ......................................................................................................................................... vi

Chapter 1 Introduction ................................................................................................................................. 1-1
  1.1 Flash Boot Loader ....................................................................................................................... 1-1
  1.2 Linux Kernel and Driver ............................................................................................................. 1-1
  1.3 Root File System ....................................................................................................................... 1-1

Chapter 2 Building the Linux Platform ........................................................................................................ 2-1
  2.1 Setting Up the Build Environment ............................................................................................. 2-1
  2.1.1 Install Linux OS using Linux Builder .................................................................................... 2-1
  2.1.2 Initial Set-up ....................................................................................................................... 2-1
  2.2 Installing/Building LTIB ............................................................................................................ 2-1
  2.3 Setting rootfs for NFS ............................................................................................................... 2-2

Chapter 3 Downloading to Target ........................................................................................................... 3-1
  3.1 Installing the ATK Tools ............................................................................................................. 3-1
  3.2 Erasing the NAND Flash ............................................................................................................ 3-1
  3.3 Running the Linux SDK using the RedBoot Bootloader ......................................................... 3-3
  3.3.1 Setting up the Host .............................................................................................................. 3-3
  3.3.2 Program RedBoot Bootloader ............................................................................................ 3-4
  3.3.3 Program the Kernel using the RedBoot Bootloader ......................................................... 3-5
  3.3.4 Set Up NFS ....................................................................................................................... 3-5
  3.3.5 Run from Target using the Redboot Bootloader .............................................................. 3-7
  3.3.6 Flash rootfs ....................................................................................................................... 3-8
  3.3.7 Running all from RAM ..................................................................................................... 3-8
Chapter 4 RedBoot Utilities ................................................................. 4-1
  4.1 Configuring RedBoot ........................................................................ 4-1
  4.2 Using RedBoot to Download OS Images ........................................ 4-3
    4.2.1 Serial Download .................................................................. 4-4
    4.2.2 Ethernet Download .............................................................. 4-4
  4.3 Other Helpful RedBoot Commands ................................................ 4-5

Appendix A Running the Linux SDK using the NAND Bootloader .......... 1
  A.1 Program the NAND Bootloader ..................................................... 1
  A.2 Program the Kernel using the NAND Bootloader ......................... 2
  A.3 Set Up NFS ................................................................................. 3
  A.4 Run from Target using the NAND Bootloader .............................. 4
  A.5 Flash rootfs ................................................................................ 4
About This Book

This document explains how to build and install the Freescale Linux platform to the i.MX31 3-Stack board.

For more information about installing the BSP and toolchain for the 3-Stack board, and building a root file system and a FLASH root file system, see the i.MX Family Linux Software Development Kit Reference Manual.

Audience

This document is intended for software, hardware, and system engineers who are planning to use the product and for anyone who wants to understand more about the product.

Organization

This document contains the following chapters.

Chapter 1  Introduces the contents and organization of the SDK Standard Release Package.
Chapter 2  Describes how to build the kernel and driver from source.
Chapter 3  Explains how to download the image to target.
Chapter 4  Provides some RedBoot utilities.
Appendix  Explains how to run the Linux SDK using the NAND bootloader.

References

The following documents were referenced to build this document.

1.  *i.MX Family Linux Software Development Kit Reference Manual*

2.  *Advanced Toolkit Standard User’s Guide*
Chapter 1
Introduction

The i.MX31 3-Stack Platform Linux SDK is a collection of binary, code, and support files that you use to create a Linux kernel image and a root file system image for the i.MX31 3-Stack board.

1.1 Flash Boot Loader

The i.MX31 3-Stack Platform Linux SDK contains Freescale NAND/Redboot Flash bootloader source code and binaries for the 3-Stack board. The NAND/Redboot Flash bootloader will be loaded to memory by i.MX31 from flash when the system is powered on. It configures the i.MX31 board, loads the kernel image from flash, and then jumps to the kernel startup entry.

The following files contain bootloader source code and binaries for the i.MX31 3-Stack board:

<table>
<thead>
<tr>
<th>bootloader\mx31-3ds_nandboot.tar.gz</th>
<th>flash\mx31-3ds_nandboot-iplspl.bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootloader\mx31-3ds_redboot_200740.zip</td>
<td>flash\mx31-3ds_redboot.bin</td>
</tr>
</tbody>
</table>

1.2 Linux Kernel and Driver

The Freescale 3-Stack BSP contains the Freescale Linux 2.6.19 3-Stack kernel, driver source code, and a pre-built kernel image.

You can obtain the 3-Stack BSP binary from:

flash\zImage
flash\Image_crc

1.3 Root File System

The root file system package provides busybox, common libraries, and other fundamental elements. The 3-Stack BSP contains the original Root File System package:

flash\rootfs.tgz

NOTE

When building the kernel image a root file system is created as well. You may use this file system rather than using the rootfs.tgz provided in the system package.
Chapter 2
Building the Linux Platform

This chapter explains how to set up the build environment, install and build LTIB, and set rootfs for NFS

2.1 Setting Up the Build Environment

Setting up the build environment includes installing Linux OS and mounting the ISO file.

2.1.1 Install Linux OS using Linux Builder

Install a Linux distribution such as Fedora 4/5 or RedHat on one computer.

2.1.2 Initial Set-up

Mount the ISO file as the root user.

```
mount -o loop ltib-imx31modular-20071228.iso /mnt/cdrom
```

2.2 Installing/Building LTIB

To install and build LTIB, use these steps.

NOTE

In some Linux systems, the following procedure must be done with “root” permissions. However, these instructions are for performing the procedure “not as root”.

1. Install the LTIB package not as root:

   ```
   cd /mnt/cdrom
   ./install
   ```

   This command installs LTIB to your directory.

2. Build LTIB.

   ```
   cd <your LTIB directory>
   ./ltib --preconfig config/platform/imx31modular/defconfig
   ```

   When complete, you may obtain the kernel image from rootfs/boot/zImage.
3. Use the following commands for CRC verification of the kernel image.

```
cd rootfs/boot
cp /mnt/cdrom/flash/crcgen ./crcgen
```

The kernel image `Image_crc` is generated. See Chapter 3 for instructions for downloading the kernel image to the 3-Stack board.

**NOTE**

You must set the network parameters in LTIB in order to boot via NFS:

```
./ltib -c
```

Set the network parameters in the following path:

**Target System Configuration**

- Options--->
  - Network setup
    - IP address
    - netmask
    - broadcast address
    - gateway address
    - nameserver IP address

### 2.3 Setting rootfs for NFS

There are two ways to set the rootfs for NFS on this package.

- Use the rootfs package already provided in the distribution; or
- Use the rootfs that is created after making the build of the kernel

**Method 1:** Using the rootfs package in the distribution

Use the following commands to set the `rootfs` directory for NFS (you must be the root user for this operation):

```
mkdir /tools/rootfs
cp /mnt/cdrom/flash/rootfs.tgz /tools/rootfs
cd /tools/rootfs
tar -zxvf rootfs.tgz
export ROOTFS_DIR=/tools/rootfs/rootfs
```

**Method 2:** Using the rootfs created after the kernel build

Instead of using the `rootfs.tgz`, you could use the root file system in `<your LTIB directory>`. So there is no need to unpack the `rootfs.tgz`.

```
%export ROOTFS_DIR=/<your LTIB directory>/rootfs
```

**Other methods:** For other ways to make a file system image file, see the *i.MX Family Linux Software Development Kit Reference Manual.*
Chapter 3
Downloading to Target

The release package provides two different bootloaders: the RedBoot bootloader and the NAND bootloader. Either can be used to start up the Linux SDK.

- For instructions on flashing and running using RedBoot, see Section 3.3.
- For instructions on flashing and running using NAND, see the Appendix.

Select one of the bootloaders and follow the instructions for that bootloader. You do not need to flash both bootloaders.

3.1 Installing the ATK Tools

These instructions explain how to use the ATK tools to download the bootloader and kernel image to the target. Ensure that you are using ATK version 1.4.

3.2 Erasing the NAND Flash

(This section is optional.)

You need to erase the NAND flash only if:

- You are using the SDK 1.11 release and want to download the SDK 1.2 release
- OR
- You are using the SDK 1.2 release and want to download the SDK 1.11 release

To erase the NAND Flash, use these steps:

1. Run the ATK tools, select the options shown in Figure 3-1, and then click Next.
2. Select Flash Tool, and then click Go.
3. Select the options shown in Figure 3-2, and then click Erase to erase the flash.
4. Change the i.MX31 boot mode to set all bits of SW5 ON (SW5 is on the Debug board).

NOTE
The ON position in SW5 means towards the center of the board, while OFF means towards edge of the board.
3.3 Running the Linux SDK using the RedBoot Bootloader

This section explains how to run and flash the system using RedBoot. If you are using a NAND bootloader, use the instructions in the Appendix.

3.3.1 Setting up the Host

NOTE

These instructions specify using an nfs server. Some Linux systems use nfsserver, rather than nfs. Use these instructions for either server type.

To set up the host system, use the following steps.

1. Turn off the firewall, to enable the tftp to work:
   ```
   iptables -F
   ```
   OR, at the command line, type:
   ```
   setup
   ```

2. Install the tftp server.

3. Install the nfs server.

4. Create the tftboot directory.

   The kernel images and anything that needs to be uploaded by tftp (such as the zImage kernel image) will be stored in this directory:
   ```
   mkdir /tftpboot
   ```

5. After building your project, link rootfs to an exportable directory:
   ```
   ln -s /< your LTIB directory >/rootfs /tftpboot/ltib
   ```

6. Copy over the kernel, bootloader, and flash filesystem images for your deployment (one of the files that is copied with this action is zImage, that is the kernel image from the system) to the /tftpboot directory:
   ```
   cp <your LTIB directory>/rootfs/boot/* /tftpboot
   ```
7. Edit /etc/xinetd.d/tftp to enable tftp as follows:

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

8. Restart the **nfs** and **tftp** servers on your host:

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

Some Linux systems use nfsserver rather than **nfs**.

### 3.3.2 Program RedBoot Bootloader

1. Run the ATK tools, select the options shown in Figure 3-1 (at the beginning of the previous chapter), and then click **Next**.

2. Select **Flash Tool**, click **Go**, and then select the options shown in Figure 4-1.

![Figure 4-1 Programming the Redboot Bootloader](image)

---

**Figure 4-1 Programming the Redboot Bootloader**

---

---
3. Click **Browse** to select the bootloader.
4. Click **Program** to flash.

### 3.3.3 Program the Kernel using the RedBoot Bootloader

To program the kernel using the Redboot bootloader, use these steps:

1. Change the i.MX31 boot mode to **external bootstrap mode**: To do so, set bit4 of SW5 OFF, all others ON. (SW5 is on the Debug board).
2. Power on the 3-Stack target.
3. Download the Kernel Image (zImage) to SDRAM:
   ```
   RedBoot> load -r -b 0x100000 /tftpboot/zImage
   ```
4. Program the downloaded image into Flash:
   ```
   RedBoot> fis create -b 0x100000 -l 0x400000 -f 0xe0040000 -e 0x100000 kernel
   ... Read from 0x07ee0000-0x07eff000 at 0xe0040000: .
   ... Read from 0x07ee0000-0x07eff000 at 0xe0040000: .
   ... Read from 0x07ee0000-0x07eff000 at 0xe0040000: .
   ... Erase from 0xe0040000-0xe0440000: ............................
   ... Program from 0x00100000-0x00500000 at 0xe0040000: .........................
   ... Erase from 0xe0040000-0xe04a0000: .
   ... Program from 0x07ee0000-0x07f00000 at 0xe0040000: .
   RedBoot>
   ```

### 3.3.4 Set Up NFS

To set up **nfs**, you will need two PCs: one Linux PC will be used as the **nfs** server and one Windows PC will be used as the command console. (You could also use one Linux PC as both the **nfs** server and command console, and then use minicom for the command console.)

To set up **nfs**, use these steps:

1. Start the **nfs** service on your Linux host machine.
2. Run the following command on your Linux host machine:
   ```
   vi /etc/exports
   ```
3. Add one of the following lines to the `/exports` file:

   — If you want to use the *rootfs.tgz* package, add the following line:

   `/tools/rootfs *(rw,sync,no_root_squash)`

   — If you want to use the *rootfs* created in the kernel build, add one of the following lines:

   `/<your ltib directory>/rootfs *(rw,sync,no_root_squash)`

   OR

   `/tftpboot/ltib *(rw,sync,no_root_squash)`

   In Section 3.3.1, step 5, a symbolic link was created that attached the location of the *rootfs* folder in the LTIB directory to `/tftpboot/ltib`.

4. Run the following command to remount:

   `exportfs -rv`

5. Connect your target to your local network via the Ethernet port on the Debug board.

6. Connect the target and your Windows PC via a serial cable.
7. Open HyperTerminal on the Windows PC and select the settings shown in Figure 4-2.

![HyperTerminal Settings](image)

Figure 4-2 HyperTerminal Settings

### 3.3.5 Run from Target using the Redboot Bootloader

Use these steps:

1. Power up the target from external bootstrap mode (bit4 of SW5 on DEBUG board is OFF, all other bits ON), and wait until the system enters the RedBoot environment.

2. Under the RedBoot environment, enter the following commands:

   ```bash
   fis load -c kernel
   exec -c "noinitrd console=ttymxc0 root=/dev/nfsroot rootfstype=nfsroot nfsroot=<the IP address of the host machine>/tools/rootfs/rootfs rw ip=dhcp"
   
   Another option may be to use static IPs rather than dhcp. In that case, the command line would be:
   
   ```bash
   fis load -c kernel
   exec -c "noinitrd console=ttymxc0 root=/dev/nfsroot rootfstype=nfsroot nfsroot=<the IP address of the host machine>/tools/rootfs/rootfs rw ip=<targetip>:<serverip>"
   ```

**NOTE**

The usage of static IPs applies in all sections of this document in which dhcp is mentioned.
The system will boot up with nfs mounted.

Note that the target can be automatically run by configuring RedBoot using “fconfig” commands set “true” when prompted with “Run script at boot”, and entering the script that is to be automatically executed after powering on.

### 3.3.6 Flash rootfs

These instructions assume that your rootfs directory is is /home/rootfs on your host PC.

To flash rootfs, use these steps:

1. Copy mkyaffs2image from /mnt/cdrom/flash directory to /home
   
   ```bash
   cp /mnt/cdrom/flash/mkyaffs2image /home
   ```

2. Create the rootfs
   
   ```bash
   cd /home
   mkyaffs2image rootfs rootfs.yaffs
   ```

3. Make sure that you have booted up via nfs according to Section 3.4.4 successfully, and then run the following command:

   ```bash
   flash_eraseall /dev/mtd/2
   nandwrite -a -o /dev/mtd/2 rootfs.yaffs
   ```

   The rootfs is now flashed to nand flash.

### 3.3.7 Running all from RAM

You can use Redboot to run the kernel image and root file system without flashing them. To do so, you must perform some steps from the previous sections.

The changes to this procedure from the previous instructions are these:

- Rather than programming the kernel after downloading it to RAM, the flashing step is not performed.
- The nfs command line stays the same, but the command line from RedBoot changes from:

  ```bash
  fis create –c kernel
  ```

  to

  ```bash
  load –r –b 0x100000 /ttftpboot/ltib
  ```

To run from RAM, use these steps:

1. Program RedBoot as shown in section 3.3.2.
2. Under the RedBoot environment, enter the following commands:

```
load -r -b 0x100000 /tftpboot/zImage
exec -c "noinitrd console=ttymxc0 root=/dev/nfsroot rootfstype=nfsroot nfsroot=<the IP address of the host machine>:/tools/rootfs/rootfs rw ip=dhcp"
```

The system will boot up with nfs mounted.

**NOTE**

You can select to have the target run automatically by configuring RedBoot via “fconfig” commands: set “true” when prompted with “Run script at boot”, and enter the script that is to be auto executed after power on”, as follows:

```
RedBoot> fconfig
Run script at boot: true
Boot script:
Enter script, terminate with empty line
>> load -r -b 0x100000 /tftpboot/zImage
>> exec -b 0x100000 -l 0x200000 -c "noinitrd console=ttymxc0 root=/dev/nfsroot rootfstype=nfsroot nfsroot=<the IP address of the host machine>:/tools/rootfs/rootfs rw ip=dhcp"
>>
Boot script timeout (1000ms resolution): 1
Use BOOTP for network configuration: true
Default server IP address: 10.192.221.167
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Read from 0x07ee0000-0x07eff000 at 0xef80000: .
... Erase from 0xef80000-0xeffa0000: .
... Program from 0x07ee0000-0x07f00000 at 0xef80000: .
RedBoot>
```
Chapter 4
RedBoot Utilities

This chapter explains how to configure RedBoot and use it to download OS images. It also provides some helpful RedBoot commands.

4.1 Configuring RedBoot

By default, RedBoot is configured to display the command prompt and receive serial keyboard input on certain UART ports with 115,200-8-N-1 settings. On the i.MX31 3-Stack board, the CPU_UART1 port is used.

To use this UART port:

- The serial cable should be attached to the UART_DCE con4 connector on the Debug daughter board, and
- The first bit (bit1) of SW4 should be set to ON.

Immediately after programming a new RedBoot image into flash, it may be necessary to reset the board and initialize the flash file system via the command prompt:

```
fis init -f
```

This should only be done ONCE after programming a new RedBoot image into flash.

The system configuration also needs to be set using the `fconfig` command. The following is one example of how to set it up by assigning a static IP address to the board. If the network supports BOOTP/DHCP and you want to use it, set `true` when prompted with Use BOOTP for network configuration in the following example and skip the configurations for Gateway IP address, Local IP address, and Local IP address mask.
Note that the new configuration will not take effect until the board is reset (using the reset button or reset command).

```
RedBoot> fconfig
Run script at boot: false
Use BOOTP for network configuration: false
Gateway IP address: 10.192.221.254
Local IP address: 10.192.221.247
Local IP address mask: 255.255.254.0
Default server IP address: 10.192.221.174
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Read from 0x07ee0000-0x07eff000 at 0xeff80000: .
... Erase from 0xeff80000-0xeффa0000: .
... Program from 0x07ee0000-0x07f00000 at 0xeфф80000: .
RedBoot>
```

You will need to modify each of the parameters for the specific network and usage. That is, the Gateway IP address in the 3-Stack RedBoot should be set to the same value as your PC, which is used as **tftp server**. It is very important that the **board specifics** parameters be set correctly. These parameters are not used by the RedBoot, but are used by some operating systems.

**NOTE**

Some DHCP servers are not configured to support BOOTP requests, and as a result, a static configuration will be required.

The **board specifics** parameter is a bit mask of board options. At this time, there are no options, and **board specifics** should be set to 0.

The configuration can be listed as follows:

```
RedBoot> fconfig -l
Run script at boot: false
Use BOOTP for network configuration: false
Gateway IP address: 10.192.221.254
Local IP address: 10.192.221.247
Local IP address mask: 255.255.254.0
Default server IP address: 10.192.221.174
Board specifics: 0
Console baud rate: 115200
Set eth0 network hardware address [MAC]: false
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
RedBoot>
```
Parameter settings can be listed along with their short names:

```
RedBoot> fconfig -l -n
boot_script: false
boot: false
bootp: false
bootp_my_gateway_ip: 10.192.221.254
bootp_my_ip: 10.192.221.247
bootp_my_ip_mask: 255.255.254.0
bootp_server_ip: 10.192.221.174
brd_specs: 0
console_baud_rate: 115200
eth0_esa: false
gdb_port: 9000
info_console_force: false
net_debug: false
RedBoot>
```

The names can be used to change individual parameters. For instance, the following will change the BOOTP server IP address to 10.192.221.167:

```
RedBoot> fconfig bootp_server_ip 10.192.221.167
bootp_server_ip: Setting to 10.192.221.167
Update RedBoot non-volatile configuration - continue (y/n)? y
... Read from 0x07ee0000-0x07eff000 at 0xeff80000: .
... Erase from 0xeff80000-0xeffa0000: .
... Program from 0x07ee0000-0x07f00000 at 0xeff80000: .
RedBoot>
```

The network configuration can be tested using the ping command:

```
ping -n 3 -h <ip address> -r 10
```

### 4.2 Using RedBoot to Download OS Images

You can download OS images using RedBoot via Ethernet or the serial port. The downloaded images can be immediately decompressed into SDRAM and executed, or they can be stored into flash and decompressed into SDRAM at a later time.

The instructions that follow assume that RedBoot has been installed onto the 3-Stack board and configured.
4.2.1  Serial Download

To download an image using a serial download, issue the following command under the RedBoot prompt:

```
load -r -b 0x100000 -m xmodem
```

RedBoot now is ready to receive data and is printing out the character “C” continuously.

To send a file, use these steps:

1. Click Transfer -> Send File -> Xmodem (under Protocol) -> Browse.
2. Select the file to download and then click **Send**.

**NOTE**

Ymodem can also be chosen for the download.

4.2.2  Ethernet Download

TFTP is not a reliable protocol and downloads may sometimes fail (with no error message) on busy networks. Always check the byte count of the download to make sure the download is complete.

Some 3-Stack boards do not have the correct MAC address programmed in the EEPROM that is used by the Ethernet controller. In that case, try using the “setmac” command under Redboot to program a valid MAC address.

This MAC address is used when using **config** and **false** for the following:

```
Set eth0 network hardware address [MAC]: false
```

If the above is set to **true**, it means that the MAC address from the flash will be used.

To download a file via the Ethernet, use these steps:

1. Set up and configure a TFTP server.
2. Configure the server to load files from the appropriate directories, or copy the desired image to the server directory.
3. Use the following command to test the network configuration:

```
ping -n 3 -h <TFTP server IP address> -r 10
```
4. Use the following command to download the image:

```
load -r -b 0x00100000 <file name>
```
4.3 Other Helpful RedBoot Commands

To delete an image:

```
fis delete <image name>
```

To list images:

```
fis list
```

To see available commands, type “help”.

Note that it is possible to configure RedBoot to execute a script at boot. This script could load and run an image from flash, or it could even load an image over Ethernet and execute it. For more information on RedBoot, refer to http://sources.redhat.com/redboot/.
Appendix A
Running the Linux SDK using the NAND Bootloader

This appendix explains how to flash and run the Linux SDK using the NAND bootloader. If you are using a RedBoot bootloader, read Section 3.3 instead.

A.1 Program the NAND Bootloader

To program the NAND bootloader, use these steps:

1. Change the i.MX31 boot mode to **internal bootstrap mode**. To do so, set switch SW5 all on (SW5 is on the Debug board, **ON** is the bottom position and **OFF** is the top position).

2. Power on the 3-Stack target and run **ADSToolkit.exe** on the PC.

3. Run the ATK tools on the PC and select the options in Figure A-1.

![Figure A-1 Flash Tool Settings](image-url)
4. Click **Browse** to select the bootloader file, and then click **Program** to download the bootloader.

### A.2 Program the Kernel using the NAND Bootloader

To program the kernel, use these steps:

1. Change the i.MX31 boot mode to **internal bootstrap mode**: To do so, set switch SW5 all on (SW5 is on the Debug board, **ON** is the bottom position and **OFF** is the top position).
2. Power on.
3. Run **ADSToolkit.exe** on the PC.
   — Run the ATK tools, select the options shown in Figure 3-1, and then click **Next**.
   — Select **Flash Tool** and click **Go**.
   — Select the options shown in Figure A-2.

![Figure A-2 Programming the Kernel](image-url)
4. Click **Program** to download the kernel image.

## A.3 Set Up NFS

To set up NFS, you will need two PCs: one Linux PC will be used as the **nfs** server and one Windows PC will be used as the command console. (You could also use one Linux PC as both **nfs** server and command console, and then use minicom for the command console.)

To set up **nfs**, use these steps:

1. **Start an **nfs** service on your Linux host machine.**
2. **Run the following command on your Linux host machine:**
   ```bash
   vi /etc/exports
   ```
3. **Add the following line in this file:**
   ```bash
   /tools/rootfs *(rw,sync,no_root_squash)
   ```
4. **Run the following command to remount:**
   ```bash
   exportfs -rv
   ```
5. **Connect your target to your local network via the Ethernet port on the Debug board.**
6. **Connect the target and your Windows PC via a serial cable.**
7. **Open HyperTerminal on the Windows PC and select the settings shown in Figure A-3.**

![Figure A-3 HyperTerminal Settings](image.png)
A.4 Run from Target using the NAND Bootloader

Use these steps:

1. Power up the target (make sure the target is booted from external bootstrap mode).

   The following information is displayed on your HyperTerminal:

   Linux 2.6 Freescale MXC processor

   Choose an option from below:

   1. Load kernel to RAM and then boot from [0x80008000]
   2. Change kernel loading address [0x80008000]
   3. Enter command line option for kernel

2. Select option 3.

3. Enter the following command:

   noninitrd console=ttymxc0 root=/dev/nfsroot rootfstype=nfsroot nfsroot=<the IP address of the host machine>/tools/rootfs/rootfs rw ip=dhcp

   The following information is displayed on your HyperTerminal:

   Choose an option from below:

   1. Load kernel to RAM and then boot from [0x80008000]
   2. Change kernel loading address [0x80008000]
   3. Enter command line option for kernel

4. Select option 1.

   The system will boot up with nfs mounted.

A.5 Flash rootfs

The following instructions assume that your rootfs directory is /home/rootfs on your host PC.

To flash rootfs, use these steps:

1. Copy mkyaffs2image from /mnt/cdrom/flash directory to /home.

2. Run the following commands to create the rootfs.

   cd /home
   mkyaffs2image rootfs rootfs.yaffs

3. To make sure you have booted up via nfs (using section 3.3.4) successfully, run the following commands.

   flash_eraseall /dev/mtd/2
   nandwrite -a -o /dev/mtd/2 rootfs.yaffs

   The rootfs is now flashed to NAND flash.