i.MX53 ARD RevB Linux
User's Guide

Document Number: 924-76373
Rev. 11.05.01
05/2011

freescale™
semiconductor

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 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

Contents

About This Book

This document explains how to build and install the Freescale Linux BSP to the i.MX53 ARD RevB board including board dip switch settings for image download and all kinds of boot mode, the steps to download image through ATK, u-Boot as well as the boot commands for each boot mode.

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.

References

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

Chapter 1
Introduction

The i.MX53 ARD Linux BSP is a collection of binary, source code, and support files that can be used to create a Linux kernel image and a root file system for i.MX53 ARD RevB board. This document is for general Linux platform.

1.1 Boot Loader

The i.MX53 ARD Linux delivery package contains U-Boot bootloader binary.

The default release package `<Version>_images_MX5X/u-boot-mx53-ard-ddr3.bin` in the release package supports NAND and MMC/SD boot for MX53 ARD RevB board.

The default release package `<Version>_images_MX5X/u-boot-mx53-ard.bin` in the release package supports NAND and MMC/SD boot for MX53 ARD RevA board.

1.2 Linux Kernel image

This Freescale i.MX BSP contains the Freescale Linux 2.6.35 ARD kernel, driver source code, and a pre-built kernel image. The i.MX53 ARD kernel image is found at the following location:

`uImage` - uImage is used together with U-Boot.

1.3 Root File System

The root file system package provides busybox, common libraries, and other fundamental elements. The i.MX53 ARD BSP package contains the following rootfs file system:

`<Version>_images_MX5X/rootfs.ext2.gz`

`rootfs.ext2.gz` file system includes Freescale specific libraries and gnome GUI. It can be mounted as NFS or the source of the storage of rootfs.

Chapter 2
Building the Linux Platform

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

2.1 Setting Up the Linux Host

See “ltib_build_host_setup.pdf” to setup Linux host server.

2.2 Installing and Building LTIB

To install and build LTIB, follow the steps below:

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”.

To run LTIB, some host packages are needed. If any error related to a host package is raised, install the host package.

1. Remove all packages on `/opt/freescale/pkgs/` which are installed before.
2. Install the LTIB package not as root:

```
tar zxf <ltib_release>.tar.gz
./<ltib_release>/install
```

This command installs LTIB to your directory.

3. Build LTIB:

```
cd <LTIB directory>
./ltib -m config
```

4. Select platform to **Freescale iMX reference boards** and exit, saving the changes. At the next menu, select platform type as **imx5x** and package profile. Exit and save changes. Please note that only **Min profile** and **FSL gnome release packages** are tested by default.
5. To build U-Boot for MX53 ARD RevB board, Select “Choose your board for u-boot” as “mx53_ard”. Please note this option is only for U-Boot. For kernel image, current default

kernel configuration can build the same images for all i.MX5 parts boards. (mx53_ard is for RevA board)

```
| --- Choose your board                                  | |
 | |     board (mx53_ard_ddr3)  --->
```

6. Run the following command:

   ./ltib

   When this procedure is completed, the kernel image is located at rootfs/boot/uImage.

7. Input the following command to get LTIB command help:

```
./ltib -help
/* Get the source code of one package */
./ltib -m prep -p <package name>
/* Build one package */
./ltib -m scbuild -p <package name>
/* Install one package to rootfs */
./ltib -m scdeploy -p <package name>
```

2.3 Setting rootfs for NFS

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

- Using the ext2 format rootfs package already provided in the distribution
- Using the rootfs that is created after making the build of the kernel

Use the following commands to set the `rootfs` directory for NFS using the `rootfs.ext2.gz` package already included in the distribution (you must be the root user for this operation):

```
mkdir /mnt/rootfs
cp imx5x/rootfs.ext2.gz  /tools
cd /tools
gunzip rootfs.ext2.gz
mount -o loop -t ext2 rootfs.ext2 /mnt/rootfs
cp -r /mnt/rootfs .
export ROOTFS_DIR=/tools/rootfs
```

NOTE

In some Linux distributions (such as Fedora) the user needs to make sure that the contents inside `/tools/rootfs` has the proper permission for user access. Since the mount command is made as root, the content shows as restricted access after the command `cp -r /mnt/rootfs`, this may cause the NFS not been able to get mounted.

To use the root file system created in the LTIB directory after the kernel build, use the command:

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

2.4 Copying images to TFTP server

To use tftp server to download the image, copy the kernel image in the release package or LTIB to the tftp directory. For example:

cp imx5x/uImage /tftpboot

Or:

cp /<LTIB directory>/rootfs/boot/uImage /tftpboot

2.5 How to generate no-padding U-Boot

To generate no-padding U-Boot, run:

```
sudo dd if=u-boot-mx53-ard-ddr3.bin of=u-boot-mx53-ard-ddr3-no-padding.bin bs=512 skip=2
```

2.6 How to generate uImage

In kernel source code, change build target from “zImage” to “uImage”.

If you want to generate uImage from zImage you built, you can generate a “uImage" based on the above zImage as below:

- Build u-boot package to get “mkimage” tool under rpm/BUILD/u-boot-<version>/ tools/mkimage.
- Copy mkimage to /usr/bin/
- Run the below command:

  mkimage -A arm -O linux -T kernel -C none -a 0x70008000 -e 0x70008000 -n "Linux-<kernel_version>" -d zImage uImage

  Note: Replace kernel version for your image. For example, 2.6.35-151-xxxx.

2.7 How to build U-Boot and Kernel in standalone environment

To build U-Boot in the standalone environment, do as the following:

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- distclean

The following configuration is for RevB board

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- mx53_ard_ddr3_config

The following configuration is for RevA board

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- mx53_ard_config

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi-

To build kernel in the standalone environment, do as the following:

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- imx5_defconfig

make ARCH=arm CROSS_COMPILE=/opt/freescale/usr/local/gcc-4.4.4-glibc-2.11.1-multilib-1.0/arm-fsl-linux-gnueabi/bin/arm-none-linux-gnueabi- uImage

2.8 Build Manufacturing Firmware

Please setup LTIB environment and then configure Firmware build profile

```
./ltib --selectype
```

Choose correct item as below:

```
--- Choose the platform type

Selection (imx5x)  --->

--- Choose the packages profile

Selection (mfg firmware profile)--->
```

In “Freescale iMX5x Based Boards” section, choose the board information as the following for RevB board:

```
--- Choose your board for u-boot

        board (mx53_ard_ddr3)  --->
```

After ltib complete build, **initramfs.cpio.gz.uboot** is generated under ltib root folder. **u-boot.bin** and **uImage** for MFG tool are generated under rootfs/boot/.

2.9 NAND LTIB configurations

MX53 ARD uses NAND socket. So the different NAND chips maybe used by the different users. The default JFFS2 rootfs image in the release package is generated for NAND with 4K page +

512K block size. If your NAND configuration is different, you must re-generate JFFS2 file system by changing the following LTIB configurations:

CONFIG_DEPLOYMENT_PAGESIZE_KB="4"

CONFIG_ERASE_BLOCK_SIZE="512"

CONFIG_DEPLOYMENT_ERASE_BLOCK_SIZE="512"

If you want to generate jffs2 manually, the following command can be used as the reference:

```
mkfs.jffs2 -r rootfs -e <block size>  -s <page size> -n -o rootfs.jffs2
```

Chapter 3 Boot configuration switch

The boot modes of the i.MX53 ARD DDR3 board are controlled by the boot configuration DIP switch on the main board. To locate the boot configuration switch refer to the i.MX53 ARD Hardware User's Guide. The following sections just list basic boot setup configurations. For more different combinations for the boot modes, see i.MX53 ARD HW guide.

3.1 Configurations for RevB board

3.1.1 Configuration setup for MFG downloading

MX53 ARD RevB board doesn't provide the dips to select serial download mode directly. The user can make ROM enter serial download mode according to ROM logic. For example, set boot mode as SD boot and don't insert SD card. Then ROM can enter serial download mode. Since NOR boot is not enabled now, the user can set default boot mode as NOR boot to enter serial download mode:

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF

3.1.2 Configuration setup for SD slot1 boot

The following table shows the dip settings for SD slot1 boot (the slot is located on the CPU board):

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0
	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF

3.1.3 Configuration setup for MMC slot1 boot

The following table shows the dip settings for MMC slot1 boot (the slot is located on the CPU board):

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0
	OFF	OFF	ON	OFF	OFF	OFF	OFF	ON	ON	OFF

3.1.4 Configuration setup for NAND boot

Linux SW uses 16-bit ECC.

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0

	ON	OFF	OFF	ON	ON	ON	OFF	OFF	ON	ON

3.2 Configurations for RevA board

3.2.1 Configuration setup for MFG downloading

MX53 ARD RevA board doesn't provide the dips to select serial download mode directly. The user can make ROM enter serial download mode according to ROM logic. For example, set boot mode as SD boot and don't insert SD card. Then ROM can enter serial download mode. Since NOR boot is not enabled now, the user can set default boot mode as NOR boot to enter serial download mode:

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0
	ON	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF

3.2.2 Configuration setup for SD slot1 boot

The following table shows the dip settings for SD slot1 boot (the slot is located on the CPU board):

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0
	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF

3.2.3 Configuration setup for MMC slot1 boot

The following table shows the dip settings for MMC slot1 boot (the slot is located on the CPU board):

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0
	ON	OFF	ON	OFF	OFF	OFF	OFF	ON	ON	OFF

3.2.4 Configuration setup for NAND boot

Linux SW uses 16-bit ECC. The user must mount R231 for NAND boot and then set dips as the following if using MT29F16G08ABACA. With this HW rework, SD/MMC boot is changed as slot2 (this slot is located on the backside of main board).

Switch	D1	D2	D3	D4	D5	D6	D7	D8	D9	D0
	ON	ON	OFF	ON	ON	ON	OFF	OFF	ON	ON

Chapter 4 Flash memory map

This chapter describes the software layout in MMC/SD cards. It can be useful to understand the later sections about image download.

4.1 MMC/SD flash memory map

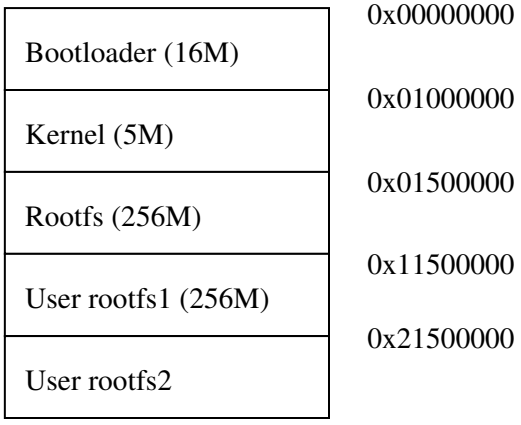
MMC/SD flash scheme is different from NAND and NOR flash which are deployed in the BSP software. MMC/SD flash must keep the first sector (512 bytes) as MBR (Master Boot Record) in order to use MMC/SD as the rootfs.

At boot up, MBR is executed to look up the partition table to determine which partition to use for booting. Bootloader should be at the end of MBR. Kernel Image and rootfs can be put any address after bootloader.

MBR can be generated through `fdisk` command when creating partitions in MMC/SD cards in the Linux Host server.

4.2 NAND flash memory map

NAND flash scheme is configured statically by software. It can be adjusted according to different requirements. The following figure illustrates default NAND flash map on MX53 ARD platform. For more details, see arch/arm/mach-mx5/mx53_ard.c.



Chapter 5 Downloading images using MFG tool

This chapter explains the procedure to use the MFG tool to download images to the different devices.

5.1 Installing the MFG Tools

Unzip Mfgtools-Rel-<version>_MX53_UPDATER.tar.gz

5.2 Usage

Read the MFG tool documentation in the "Document" folder, before using the MFG tool. The MFG tool follows the instructions into "`Profiles\MX53 Linux Update\OS Firmware\ucl.xml`" to execute program operations. The user must read and update ucl.xml to understand the operations before using the MFG tool.

Follow these instructions to use the MX53 SABRE Tablet MFG tool:

- Connect a USB cable from a PC to the USB OTG port (J8) on the CPU board.
- Connect UART to PC for console output. Open a Terminal emulator program.
- Set BOOT dip 4 as on, press SW2 to reset CPU and ensure the USB device is found.
- The default profile of the manufacturing tool assumes your file system to be packed and compressed using `bzip2` algorithm. To create this file, you can run the following commands as a root user. You can also modify profile to support other formats.
    - `>cd your_rootfs_dir`
    - `>tar -cjf rootfs.tar.bz2 *`
- You can specify your images in two ways: The first is by editing "`Profiles\MX53 Linux Update\OS Firmware\ucl.xml`" to modify the file path or flash operations according to your usage. Note that "MX53ARD-DDR3-xxx" lists are the example codes for MX53 ARD RevB boards. "MX53ARD-xxx" lists are the example codes for MX53 ARD RevA board. You can modify them for MX53 ARD RevB board programming. After the modification is completed, save the changes and exit. Another way is by copying your files in "`Profiles\MX53 Linux Update\OS Firmware\files`" directory. You can replace the files inside this folder. Note that you will find `u-boot-mx53-ard-ddr3.bin` and `uImage` binaries in "`Profiles\MX53 Linux Update\OS Firmware`" folder. These files should not be replaced. They are different from your image files and serve another purpose.

- Execute "`MfgTool.exe`". Select the "Options -> Configuration" menu. If this is the first time connecting an i.MX53 board with the MFG tool, install a USB driver under "`Drivers\iMX_BulkIO_Driver`".
- Select the appropriate USB port in the sheet "USB Ports".
- Select the appropriate profile in the sheet "Profiles." Type the item in "Operations." Right-click and select "Edit." (Note that the default profiles are used as examples. They should be modified according to the user's programming operations.)
  - Select "MX53ARD-DDR3-SD" to program images to SD in RevB board
  - Select "MX53ARD-DDR3-NAND(JFFS2)" to program images to NAND in RevB board.
- Insert SD card.
- Start the downloading process by pressing the green, **Start,** button. You will see the progress bar as well as the current task in the notification bar as shown in Figure 5-5. When you see "Update Complete" in the notification bar, press the red, **Stop,** button to finish.
- The manufacturing tool may sometimes report an error message when it is downloading the file system in an SD card. This can be caused by insufficient space in the SD card due to a small partition size. To fix this, unzip the file "`Profiles\MX53 Linux Update\OS Firmware\mksdcard.sh.tar`" and then modify the script to increase the size of the partition and create more partitions according to your file system requirements. After the modification is done, tar the script again.

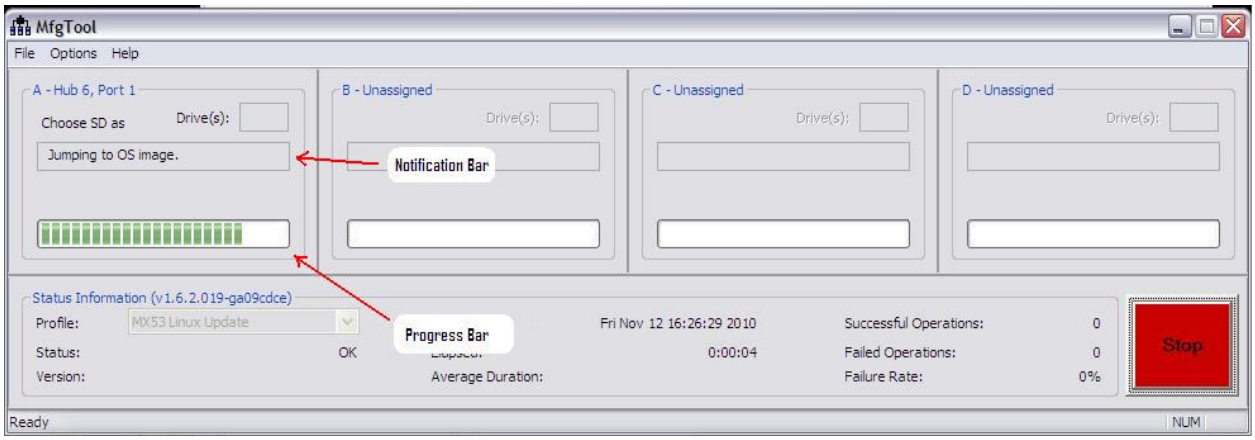


Figure 5-1 Programming SD With Manufacturing Tool.

Chapter 6Download Images by Bootloader or NFS

6.1 Setup Terminal

The i.MX53 ARD board can communicate with a host server (Windows or Linux) using the serial cable. Common serial communication programs such as HyperTerminal, Tera Term or PuTTY can be used. The example below describes the serial terminal setup using HyperTerminal on a Windows host:

1. Connect the target and the Windows PC using a serial cable.
2. Open HyperTerminal on the Windows PC, and select the settings as shown in Figure 6-1.

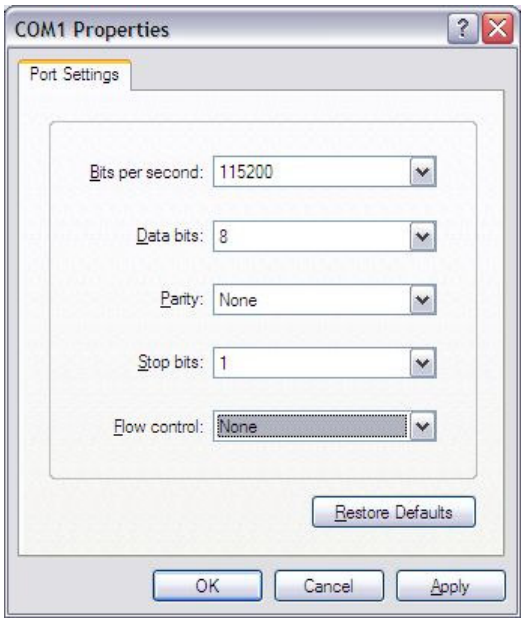


Figure 6 -1 HyperTerminal Settings for Terminal Setup

3. After the bootloader is programmed on SD card, press SW2 key to reset CPU. The bootloader prompt is displayed on the terminal screen.

6.2 Download by U-Boot

6.2.1 MMC/SD in SDHC1

- To clean up the environments stored on MMC/SD, do as the following in U-Boot console:

```
MX53-ARD-DDR3 U-Boot > mmc write 0 0x70100000 0x600 0x10
```

- Set BOOT dip 3, 9 on for RevB board. Power up the board and set the U-Boot environment variables as needed. For example,

```
MX53-ARD-DDR3 U-Boot > setenv bootargs console=ttymxc0,115200
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000
MX53-ARD-DDR3 U-Boot > setenv serverip 10.192.225.216
MX53-ARD-DDR3 U-Boot > setenv bootfile uImage
MX53-ARD-DDR3 U-Boot > saveenv
```

- Copy uImage to `tftp` server. Then download it to RAM:

```
MX53-ARD-DDR3 U-Boot > dhcp
```

- Query the information about MMC/SD card.

```
MX53-ARD-DDR3 U-Boot > mmcinfo 0
Device: FSL_ESDHC
Manufacturer ID: 3
OEM: 5344
Name: SD04G
Tran Speed: 25000000
Rd Block Len: 512
SD version 2.0
High Capacity: Yes
Capacity: 3965190144
Bus Width: 4-bit
Boot Partition Size: No boot partition available
```

- Check the usage of "`mmc`" command. The "`blk#`" is equal to "`<the offset of read/write>/<block length of the card>`". The "`cnt`" is equal to "`<the size of read/write>/<block length of the card>`".

```
MX53-ARD-DDR3 U-Boot > help mmc
mmc - MMC sub system

Usage:
mmc mmc read <device num> addr blk# cnt [partition]
mmc write <device num> addr blk# cnt [partition]
mmc rescan <device num>
mmc list - lists available devices
```

- Program the kernel uImage into MMC/SD. For example, the below command writes the image with the size 0x300000 from `${loadaddr}` to the offset 0x100000 of the MMC/SD card. Here 0x800 =0x100000/512, 0x1800=0x300000/512. The block size of

this card is 512. This example assumes the kernel image is less than 0x300000 bytes. If the kernel image exceeds 0x300000, enlarge the image length.

```
MX53-ARD-DDR3 U-Boot > mmc write 0 ${loadaddr} 0x800 0x1800

MMC write: dev # 0, block # 2048, count 6144, partition # 0 ...
6144 blocks written: OK
```

- Boot up the system through RFS in SD card via HannStar LVDS0 in the CPU board:

```
MX53-ARD-DDR3 U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs}
              console=tty1 root=/dev/mmcblk0p1 rootwait rw
              video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0 ip=dhcp'
MX53-ARD-DDR3 U-Boot > setenv bootcmd_mmc 'run bootargs_base
              bootargs_mmc;mmc read 0 ${loadaddr} 0x800 0x1800;bootm'
MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_mmc'
MX53-ARD-DDR3 U-Boot > saveenv
```

- Boot up the system through RFS in SD card via HannStar LVDS1 in the main board:

```
MX53-ARD-DDR3 U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs}
              console=tty1 root=/dev/mmcblk0p1 rootwait rw
              video=mxcdi1fb:RGB666,XGA di1_primary ldb=di1 ip=dhcp'
MX53-ARD-DDR3 U-Boot > setenv bootcmd_mmc 'run bootargs_base
              bootargs_mmc;mmc read 0 ${loadaddr} 0x800 0x1800;bootm'
MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_mmc'
MX53-ARD-DDR3 U-Boot > saveenv
```

- Boot up the system through RFS in SD card via VGA (Please note: Short PIN 1-2 for J14 and J16 on CPU board):

```
MX53-SMD U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs}
              console=tty1 root=/dev/mmcblk0p1 rootwait rw
              video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga ldb=off ard-
              vga ip=dhcp'
```

- To program the rootfs to MMC/SD, See section "Using a Linux Host to Set Up an SD/MMC card".

6.2.2 NAND

1. Insert NAND chip into NAND socket.
2. The default environment setting for MX53 ARD board is the SD card. To store the environment on the NAND, do as the followings:
   a. Modify U-Boot source code and apply the following patch:

```
diff --git a/include/configs/mx53_ard_ddr3.h
b/include/configs/mx53_ard_ddr3.h

index 3b391da..049a557 100644

--- a/include/configs/mx53_ard_ddr3.h

+++ b/include/configs/mx53_ard_ddr3.h
```

```
@@ -262,7 +262,8 @@
 #define CONFIG_SYS_NO_FLASH


 /* Monitor at beginning of flash */
-#define CONFIG_FSL_ENV_IN_MMC
+/* #define CONFIG_FSL_ENV_IN_MMC */
+#define CONFIG_FSL_ENV_IN_NAND
 /* #define CONFIG_FSL_ENV_IN_SATA */


#define CONFIG_ENV_SECT_SIZE    (128 * 1024)
```

b. Rebuild U-Boot and flash new bootloader to NAND device via kobs-ng tool.

3. Set BOOT PIN 1, 4,5,6,9, 10 ON, Others are OFF. Power up the board and set the U-Boot environment variables as needed. The following examples assume the environment variables are stored into SATA device.

```
MX53-ARD-DDR3 U-Boot > setenv bootargs console=ttymxc0,115200
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000
MX53-ARD-DDR3 U-Boot > setenv serverip 10.192.225.216
MX53-ARD-DDR3 U-Boot > setenv bootfile uImage
MX53-ARD-DDR3 U-Boot > saveenv
```

4. Copy uImage to tftp server. Then download it to RAM:

```
MX53-ARD-DDR3 U-Boot > dhcp
```

5. Program the kernel uImage into NAND.

```
MX53-ARD-DDR3 U-Boot > nand erase 0x1000000 0x300000

MX53-ARD-DDR3 U-Boot > nand write ${loadaddr} 0x1000000 0x300000



NAND write: device 0 offset 0x1000000, size 0x300000

 3145728 bytes written: OK
```

6. Boot up the system through RFS in NAND via LVDS0:

```
MX53-ARD-DDR3 U-Boot > setenv bootargs_nand 'setenv bootargs ${bootargs}
             console=tty1 root=/dev/mtdblock2 rootfstype=jffs2
             rootwait rw video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0
             ip=dhcp'
MX53-ARD-DDR3 U-Boot > setenv bootcmd_nand 'run bootargs_base
             bootargs_nand;nand read ${loadaddr} 0x1000000
             0x300000;bootm'
MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_nand'
```

6.2.3 SATA

1. Connect a SATA device to J20.
2. Power up the board from SD and set the U-Boot environment variables as needed. The following examples assume the environment variables are stored into SATA device.

```
MX53-ARD-DDR3 U-Boot > setenv bootargs console=ttymxc0,115200\
MX53-ARD-DDR3 U-Boot > setenv bootargs console=ttymxc0,115200
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000
MX53-ARD-DDR3 U-Boot > setenv serverip 10.192.225.216
MX53-ARD-DDR3 U-Boot > setenv bootfile uImage
MX53-ARD-DDR3 U-Boot > saveenv
```

3. Copy uImage to tftp server. Then download it to RAM:

```
MX53-ARD-DDR3 U-Boot > dhcp
```

4. Query SATA information:

```
MX53-ARD-DDR3 U-Boot > sata info

AHCI 0001.0100 32 slots 1 ports 3 Gbps 0x1 impl SATA mode

flags: ncq stag pm led clo only pmp pio slum part

SATA Device Info:

S/N: 104550300093

Product model number: SanDisk SSD P4 32GB

Firmware version: SSD 8.00

Capacity: 62533296 sectors


SATA device 0: Model: SanDisk SSD P4 32GB Firm: SSD 8.00 Ser#:
104550300093

            Type: Hard Disk

            Supports 48-bit addressing

Capacity: 30533.8 MB = 29.8 GB (62533296 x 512)
```

5. Program the kernel uImage into SATA. For example, the below command writes the image with the size 0x300000 from ${loadaddr} to the offset 0x100000 of the SATA. Here 0x800 =0x100000/512, 0x1800=0x300000/512. The block size of this card is 512. This example assumes the kernel image is less than 0x300000 bytes. If the kernel image exceeds 0x300000, enlarge the image length.

```
MX53-ARD-DDR3 U-Boot > sata write ${loadaddr} 0x800 0x1800


   SATA write: device 0 block # 2048, count 6144 ... 6144 blocks written:
OK
```

6. Boot up the system through RFS in SATA via LVDS0:

```
MX53-ARD-DDR3 U-Boot > setenv bootargs_sata 'setenv bootargs ${bootargs}
              console=tty1 root=/dev/sda1 rootwait rw
              video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0 ip=dhcp'
MX53-ARD-DDR3 U-Boot > setenv bootcmd_sata 'run bootargs_base
              bootargs_sata;sata read  ${loadaddr} 0x800 0x1800;bootm'
MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_sata'
MX53-ARD-DDR3 U-Boot > saveenv
```

6.2.4 U-Boot Configurations

The U-Boot “print” command can be used to check environment variable values. The “setenv” command can be used to set environment variable values. See the U-Boot user guide for details.

Chapter 7
Running the Image on the Target

This chapter explains how to run an image on the target from downloaded flash and NFS. These instructions assume that you have downloaded the kernel image using the instructions in Chapter 5 or Chapter 6.

7.1 Run the image from NFS

To boot from NFS, do as the follows (Please pay attention to the items marked as blue color. You need to modify them per your environment or HW information):

1. Set boot configuration switch as indicated in Chapter "Boot configuration switch". Power on the board.
2. Enter the following commands in the U-Boot prompt:

```
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000

MX53-ARD-DDR3 U-Boot > setenv serverip 10.192.225.216

MX53-ARD-DDR3 U-Boot > setenv bootfile uImage

MX53-ARD-DDR3 U-Boot > setenv nfsroot 10.192.225.216:/nfs/rootfs

MX53-ARD-DDR3 U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc0,115200'

### for LVDS0 on the CPU board

MX53-ARD-DDR3 U-Boot > setenv bootargs_nfs 'setenv bootargs ${bootargs} root=/dev/nfs ip=dhcp
nfsroot=${nfsroot},v3,tcp video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0'

MX53-ARD-DDR3 U-Boot > setenv bootcmd_net 'run bootargs_base bootargs_nfs;bootm'

MX53-ARD-DDR3 U-Boot > setenv bootcmd 'dhcp; run bootcmd_net'

MX53-ARD-DDR3 U-Boot > saveenv

MX53-ARD-DDR3 U-Boot > run bootcmd
```

7.2 Run the image from MMC/SD flash

To boot the system from MMC/SD flash follow the steps bellow:

1. Set boot configuration switch as indicated in Chapter "Boot configuration switch". Power on the board.

2. Assume the kernel image start from the address 0x100000 byte (the block start address is 0x800). The kernel image size is less than 0x300000 byte. The rootfs is located into /dev/mmcblk0p1 partition. Enter the following commands in the U-Boot prompt:

```
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000

MX53-ARD-DDR3 U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc0,115200'

### For VGA output, Short PIN 1-2 for J14 and J16 on CPU board

MX53-ARD-DDR3 U-Boot > setenv bootargs_mmc 'setenv bootargs ${bootargs} console=tty1
root=/dev/mmcblk0p1 rootwait rw video=mxcdi1fb:GBR24,VGA-XGA di1_primary vga ldb=off ard-vga
ip=dhcp'

MX53-ARD-DDR3 U-Boot > setenv bootcmd_mmc 'run bootargs_base bootargs_mmc;mmc read 0
${loadaddr} 0x800 0x1800;bootm'

MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_mmc'

MX53-ARD-DDR3 U-Boot > saveenv

MX53-ARD-DDR3 U-Boot > run bootcmd
```

7.3 Run the image from NAND

To boot the system from NAND flash follow the steps bellow:

1. Set boot configuration switch as indicated in Chapter "Boot configuration switch". Power on the board.

2. Enter the following commands in the U-Boot prompt:

```
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000
MX53-ARD-DDR3 U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc0,115200'
### For LVDS1 on the main board
MX53-ARD-DDR3 U-Boot > setenv bootargs_nand 'setenv bootargs ${bootargs} console=tty1
root=/dev/mtdblock2 rootfstype=jffs2 rootwait rw video=mxcdi1fb:RGB666,XGA di1_primary ldb=di1
ip=dhcp'
MX53-ARD-DDR3 U-Boot > setenv bootcmd_nand 'run bootargs_base bootargs_nand;nand read
${loadaddr} 0x1000000 0x300000;bootm'
MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_nand'
MX53-ARD-DDR3 U-Boot > saveenv
MX53-ARD-DDR3 U-Boot > run bootcmd
```

7.4 Run the image from SATA

To boot the system from SATA device follow the steps bellow:

3. Set boot configuration switch as indicated in Chapter "Boot configuration switch". Power on the board.

4. Enter the following commands in the U-Boot prompt:

```
MX53-ARD-DDR3 U-Boot > setenv loadaddr 0x70800000
MX53-ARD-DDR3 U-Boot > setenv bootargs_base 'setenv bootargs console=ttymxc0,115200'
### For LVDS0 on the CPU board
MX53-ARD-DDR3 U-Boot > setenv bootargs_sata 'setenv bootargs ${bootargs} console=tty1
                       root=/dev/sda1 rootwait rw video=mxcdi0fb:RGB666,XGA di0_primary ldb=di0
                       ip=dhcp'
MX53-ARD-DDR3 U-Boot > setenv bootcmd_sata 'run bootargs_base bootargs_sata;sata read  ${loadaddr}
                       0x800 0x1800;bootm'
MX53-ARD-DDR3 U-Boot > setenv bootcmd 'run bootcmd_sata'
MX53-ARD-DDR3 U-Boot > saveenv
MX53-ARD-DDR3 U-Boot > run bootcmd
```

Chapter 8
Using a Linux Host to set up an SD/MMC card

This chapter describes the steps to prepare an SD/MMC card to boot off an i.MX53 ARD.

8.1.1 Requirements

An SD/MMC card reader, like a USB card reader, is required. It will be used to transfer the boot loader and kernel images, to initialize the partition table and copy the root file system. To simplify the instructions, it is assumed that a 4GB SD/MMC card is used.

Any Linux distribution can be used for the following procedure. It is recommended to use a Linux distribution that LTIB has been tested against (like Fedora, Ubuntu, etc).

The Linux kernel running on the Linux host will assign a device node to the SD/MMC card reader. The kernel might decide the device node name or udev rules might be used. In the following instructions, it is assumed that udev is not used.

To identify the device node assigned to the SD/MMC card, enter the command:

```
$ cat /proc/partitions

major minor  #blocks  name

   8     0   78125000 sda

   8     1   75095811 sda1

   8     2          1 sda2

   8     5    3028221 sda5

   8    32  488386584 sdc

   8    33  488386552 sdc1

   8    16    3921920 sdb

   8    18    3905535 sdb1
```

In this example, the device node assigned is `/dev/sdb` (a block is 1kB large)

8.1.2 Copying the boot loader image

Enter the following command to copy the U-Boot image to the SD/MMC card (please note that this operation will delete the partition table present on the media):

```
$ sudo dd if=u-boot-mx53-ard-ddr3.bin of=/dev/sdb bs=512 && sync && sync
```

To update U-Boot to another version, please run the following command instead:

```
$ sudo dd if=u-boot-mx53-ard-ddr3-no-padding.bin of=/dev/sdb bs=512 seek=2 && sync && sync
```

The first 1kB, that includes the partition table, will be preserved.

8.1.3 Copying the kernel image

The following command will copy the kernel image to the SD/MMC card

```
$ sudo dd if= uImage of=/dev/sdb bs=512 seek=2048 && sync && sync
```

This will copy the `uImage` to the media at offset 1MB.

8.1.4 Copying the file system (rootfs)

A partition table must be first created. If a partition already exists and it is big enough for the file system you want to deploy, then you can skip this step.

To create a partition, at offset 8192 (in sectors of 512 bytes) enter the following command:

```
$ sudo fdisk /dev/sdb
```

Type the following parameters (each followed by **<ENTER>**):

u	[switch the unit to sectors instead of cylinders]
d	[repeat this until no partition is reported by the ‘p’ command]
n	[create a new partition]
p	[create a primary partition]
1	[the first partition]
8192	[starting at offset sector #8192, i.e. 4MB, which leaves enough space for the kernel, the boot loader and its configuration data]
<enter>	[using the default value will create a partition that spans to the last sector of the medium]
w	[this writes the partition table to the medium and fdisk exits]

The file system format ext3 or ext4 is a good option for removable media due to the built-in journaling. Run the following command to format the partition:

```
$ sudo mkfs.ext3 /dev/sdb1
Or
$ sudo mkfs.ext4 /dev/sdb1
```

Copy the target file system to the partition:

```
$ mkdir /home/user/mountpoint
```

```
$ sudo mount /dev/sdb1 /home/user/mountpoint
```

Let's assume the root file system files are located in `/home/user/rootfs`:

```
$ cd /home/user/rootfs
$ sudo cp –rpa [A-z]* /home/user/mountpoint
$ sudo umount /home/user/mountpoint
```

The file system content is now on the media.