Ronetix Application Note AN014: High-Reliability Boot-up for SAMA5D3x-CM

1. Overview,

This article shows you how to build a high reliable boot up for SAMA5D3x-CM. The set of patches adds redundant booting of U-BOOT and Kernel. Additionally 8-bit ECC is used for at91bootstrap, U-BOOT, Kernel image and RootFS. The UBIFS bitflip_threshold is set to 6. This means that if there are 6,7 or 8 bit flips per 512 bytes block, then the NAND page is marked for scrubbing.

2. Prerequisites

- a CPU Module SAMA5D3x-CM
- a Base board SAMA5D3x-MB
- A host machine running Linux for building

3. MTDs partitions

Because of the additionally images, the following MTD partitions are used:

```
0x00000000 0x00040000 At91 bootstrap
0x00040000 0x00080000 U-Boot
0x000C0000 0x00080000 U-Boot Redundant image 1
0x00140000 0x00080000 U-Boot Redundant image 2
0x001C0000 0x000C0000 U-Boot environments
0x001C0000 0x00040000 env
0x00200000 0x00040000 env redundant
0x00240000 0x00040000 reserved for env redundant second
0x00300000 0x00600000 kernel
0x00980000 0x00600000 kernel Redundant
0x00f80000 - rootfs
```

4. Toolchain: download, unpack, set CROSS_COMPILE

The toolchain is this example will be installed in /opt/cross

```
$ cd /opt/
$ sudo mkdir cross
$ cd cross

$ sudo wget https://download.ronetix.at/toolchains/arm/gcc-arm-none-eabi-4_9-2015q1.tar.bz2
$ sudo tar xvfj gcc-arm-none-eabi-4_9-2015q1.tar.bz2
$ export CROSS_COMPILE=/opt/cross/gcc-arm-none-eabi-4_9-2015q1/bin/arm-none-eabi-
```

5. Patches: download, unpack

```
$ mkdir -p ~/work
$ cd ~/work
$ wget https://download.ronetix.at/boards/doc/SAMA5D3x-CM/patches/sama5d3x-redundant-linux_4.9.8
```

```
7-linux4sam_5.8.tar.xz
$ tar xfv sama5d3x-redundant-linux_4.9.87-linux4sam_5.8.tar.xz
```

6. AT91Bootstrap: download, patch, build

```
$ cd ~/work/sama5d3x-redundant/at91bootstrap
$ git clone git://github.com/linux4sam/at91bootstrap.git
$ cd at91bootstrap
$ git checkout -b v3.8.1 v3.8.1
$ git am ../patches/0001-add-support-for-redundant-booting-of-U-BOOT.patch
$ git am ../patches/0002-use-8-bit-PMECC.patch
$ make sama5d3xeknf_uboot_defconfig
$ make
$ cp binaries/sama5d3xek-nandflashboot-uboot-3.8.1.bin ../../bin/boot.bin
```

REF: http://www.at91.com/linux4sam/bin/view/Linux4SAM/AT91Bootstrap

7. U-Boot: download, patch, build, pad to 512KB, add CRC32 at end of file

```
$ cd ~/work/sama5d3x-redundant/u-boot-at91
$ git clone git://github.com/linux4sam/u-boot-at91.git
$ cd u-boot-at91
$ git checkout origin/u-boot-2017.03-at91 -b u-boot-2017.03-at91
$ git am ../patches/0001-Added-support-for-redundant-booting-of-Linux-Kernel.patch
$ git am ../patches/0002-Added-environment-variable-conf_name-used-when-boot-.patch
$ git am ../patches/0003-Use-8-bit-PMECC-instead-of-4-bit.patch
$ make sama5d3xek_nandflash_defconfig
$ make
```

Pad the U-BOOT image with zeros to 512KiB and add at the end 32-bit CRC32:

```
$ ../../bin/padfile u-boot.bin

$ cp u-boot.bin ../../bin/
$ sudo cp tools/mkimage /usr/bin/
```

REF: http://www.at91.com/linux4sam/bin/view/Linux4SAM/U-Boot

8. Linux Kernel: download, patch, build

This set of patches uses the new, FIT (Flattened Image Tree) format which allows more flexibility in handling various images and also enhances integrity protection.

```
$ cd ~/work/sama5d3x-redundant/linux-at91
$ git clone git://github.com/linux4sam/linux-at91.git
$ cd linux-at91
$ git remote add linux4sam git://github.com/linux4sam/linux-at91.git
$ git remote update linux4sam
$ git checkout origin/linux-4.9-at91 -b linux-4.9-at91
$ git am ../patches/0001-Add-sources-for-U-Boot-FIT-images.patch
$ git am ../patches/0002-Modified-NAND-MTD-partitions-because-of-redundant-mu.patch
$ git am ../patches/0003-Use-8-bit-NAND-PMECC.patch
$ git am ../patches/0004-Do-not-use-NFC-SRAM-when-reading-from-NAND-Flash.patch
$ git am ../patches/0005-Add-dts-parameter-for-setting-bitflip_threshold.patch
```

```
$ export ARCH=arm
$ make mrproper
$ make sama5_defconfig
$ make -j 8
$ make uImage
$ mkimage -f ./arch/arm/boot/dts/sama5d3xek.its sama5d3xek.itb
$ cp sama5d3xek.itb ../../bin
```

9. Root File system – Buildroot

```
$ mkdir -p ~/work/sama5d3x-redundant/buildroot
$ cd ~/work/sama5d3x-redundant/buildroot
$ git clone git://github.com/linux4sam/buildroot-at91.git
$ cd buildroot-at91
$ git checkout master
$ make sama5d3_defconfig
$ make
$ cp output/images/rootfs.ubi ../../bin
```

REF:
http://www.at91.com/linux4sam/bin/view/Linux4SAM/BuildRootBuild

10. Program images with SAM-BA

Register to ATMEL site and get SAM-BA v2.15

```
$ cd /opt
$ sudo unzip sam-ba_2.15.zip

$cd ~/work/sama5d3x-redundant/bin
```

Check that all necessary images are here:

```
$ ls
boot.bin ; at91bootstrap bootloader
d3x.sh ; bash script which calls d3x.tcl to program all images
d3x.tcl
padfile ; bash script used to pad u-boot.bin to 512KiB and to add CRC32
rootfs.ubi ; Buildroot RootFS image
sama5d3xek.itb ; Linux kernel FIT image
u-boot.bin ; U-BOOT bootloader
```

Start programming:

```
$ ./d3x.sh
```

11. Program images with PEEDI

PEEDI have to be configured to use the Ronetix configuration file for SAMA5Dx3-CM: atsama5d3.cfg

Note: PEEDI supports 8-bit PMECC programming since firmware v16.x.x.

Copy the images from “~/work/sama5d3x-redundant/bin” to the TFTBOOT directory and modify the paths in atsama5d3.cfg.
When PEEDI successfully connects to the board, then just enter the following command to program bootstrap, two redundant U_BOOT images , two redundant Linux kernel images and root filesystem at once:

```
sama5d3x> run $prog
```

Alternatively, single images can be programmed using:

```
sama5d3x> flash set 0   ; select bootstrap flash section
sama5d3x> flash erase   ; erase the whole NAND Flash
sama5d3x> flash program ; program the bootstrap image using 8-bit ECC and add 52×32-bit header
sama5d3x> flash set 1   ; select U-BOOT flash section
sama5d3x> flash program ; program U-BOOT image using 8-bit ECC
```

Share on LinkedinShare on FacebookShare on Reddit on TwitterShare Via EmailShare on Whatsapp

Share