



SAMA5D3x

doorroos.realtime

Build Guide Manual

Manual

Document Number: DOOROO-DKBD-BG-2013-xxxx

doorroos@doorroos.org

Version 0.2

Contents

1. Overview	3
1.1. Development Environment.....	3
2. Building Bootloader (JBOOT)	4
2.1. Build sama5d3kek-nandflashboot-uboot-3.5.4.bin	4
2.2. Open the dooroos.realtime visual studio.....	6
2.3. Open the solution	7
2.4. Build J-boot	8
3. Building Image	10
3.1. dooroos.realtime Image	10
3.2. Build Image	11
4. Nand flash fusing with SAM-BA	14
4.1. Run SAM-BA.....	17
4.2. Download the filesystem into nand flash by USB	28

1. Overview

SAMA5D3X_EKBD has a high performance CoretxA5 Core and many Peripheral devices, and supports dooroos.realtime BSP(Board Support Package) and sample images.

This document is a guide for a new SAMA5D3X_EKBD user who use dooroos.realtime BSP. This document describes how to use Bootloader, build dooroos.realtime and download Image files.

1.1. Development Environment

- Windows Desktop
- SAMA5D3X_EKBD Board
- Serial and USB connection between SAMA5D3X_EKBD and Windows Desktop
 - n Your desktop must have serial communication tools(like HyperTerminal).
- dooroos.realtime visual studio, GCC compiler and images (download from dooroos.org)
 - n <http://www.dooroos.org/>



2. Building Bootloader (JBOOT)

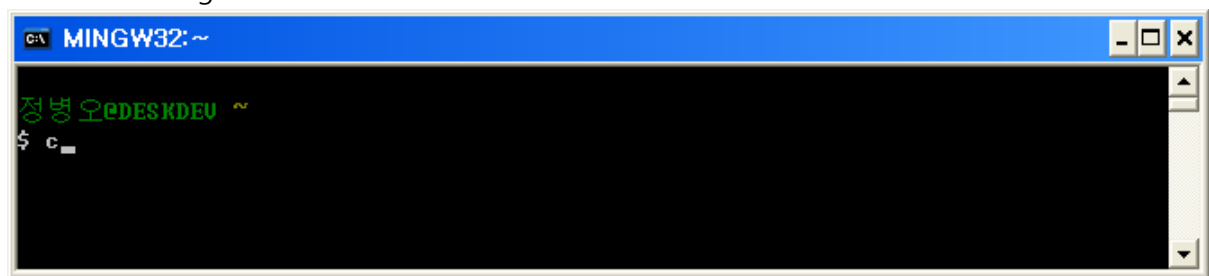
SAMA5D3X_EKBD BSP Bootloader use J-boot as Bootloader. J-boot is the dooroos.realtime bootloader for ARM and stable bootloader. It was widely used, has various functions and has reliability by long time use.

2.1. Build sama5d3xek-nandflashboot-uboot-3.5.4.bin

SAMA5D3X_EKBD needs the file sama5d3xek-nandflashboot-uboot-3.5.4.bin to boot. The file sama5d3xek-nandflashboot-uboot-3.5.4.bin is given from ATMEL. But we have modified to use the special function that is needed in dooroos.realtime.

If you have the above tools, please skip this step and use the given file.

1. run the mingw shell.



```
C:\ MINGW32: ~
정병오@DESKDEV ~
$ c
```

2. move to the directory "at9bootstrap" and build as the followings

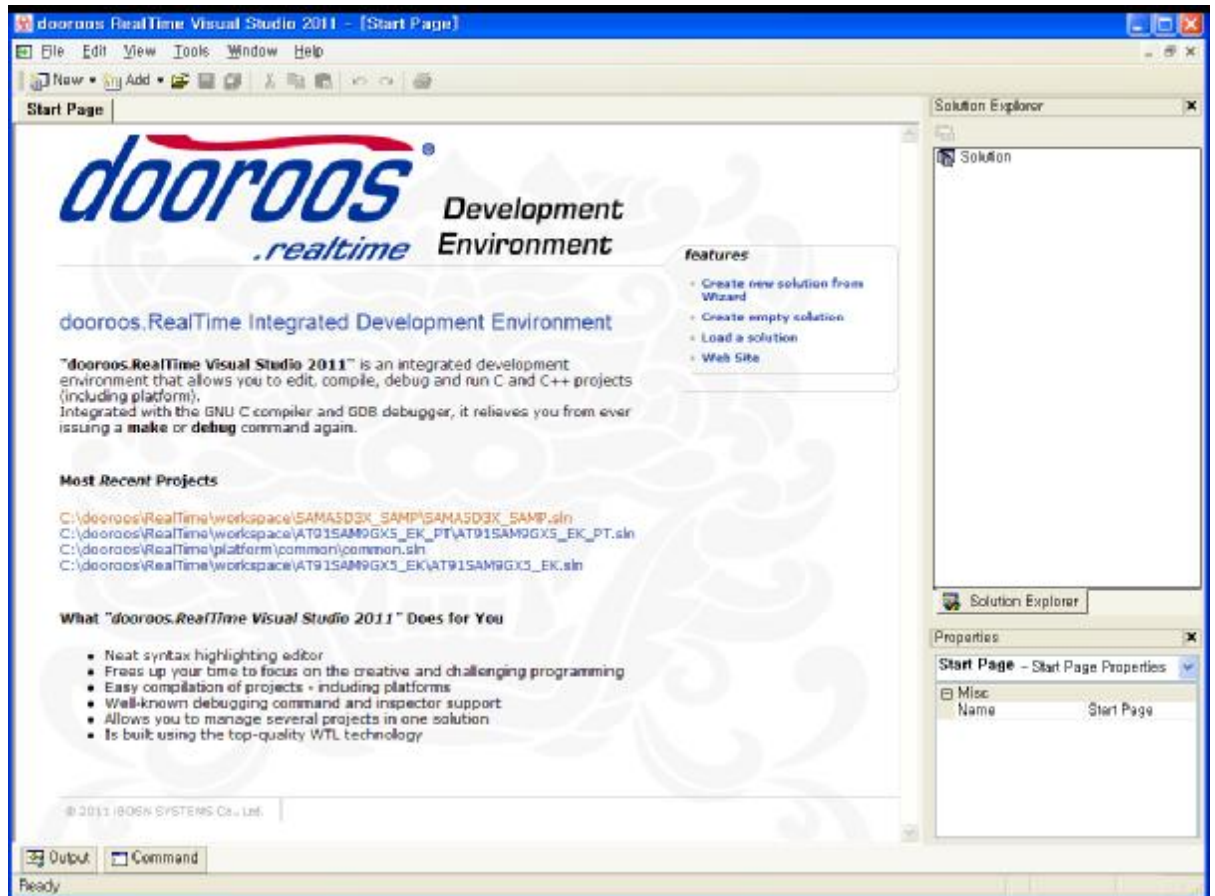


```
C:\ MINGW32: /c/doorroos/RealTime/platform/SAMA5D3X_EK/JBOOT/at9bootstrap
정병오@DESKDEV /c/doorroos/RealTime/platform/SAMA5D3X_EK/JBOOT/at9bootstrap
$ make
```

3. you can get the file sama5d3xek-nandflashboot-uboot-3.5.4.bin

```
C:\ MINGW32:/c/doorroos/RealTime/platform/SAMA5D3X_EK/JBOOT/at91bootstrap
ld FLAGS
=====
-nostartfiles -Map=/c/doorroos/RealTime/platform/SAMA5D3X_EK/JBOOT/at91bootstrap/
binaries/sama5d3xek-nandflashboot-uboot-3.5.4.map --cref -static -T elf32-little
arm.lds --gc-sections -Ttext 0x3000000
/bin/sh: stat: command not found
Size of sama5d3xek-nandflashboot-uboot-3.5.4.bin is  bytes
/bin/sh: line 2: [: : integer expression expected
[Succeeded] It's OK to fit into SRAM area
정병오@DESKDEV /c/doorroos/RealTime/platform/SAMA5D3X_EK/JBOOT/at91bootstrap
$
```

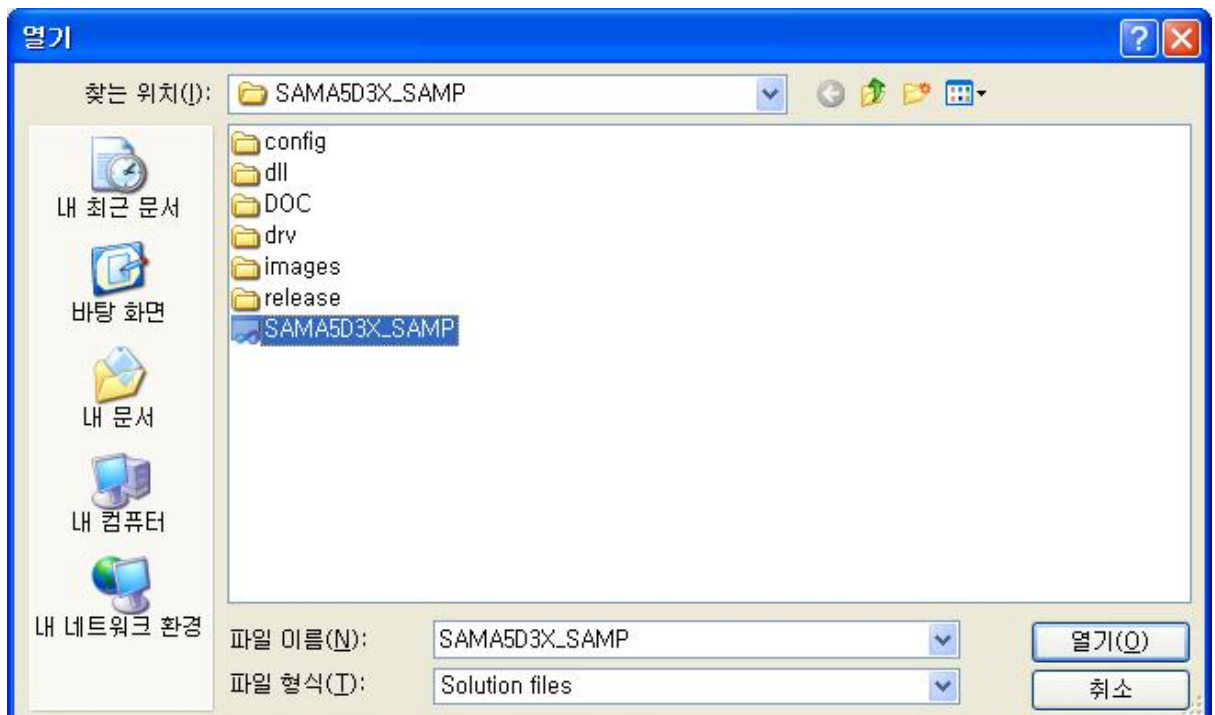
2.2. Open the dooroos.realtime visual studio



2.3. Open the solution

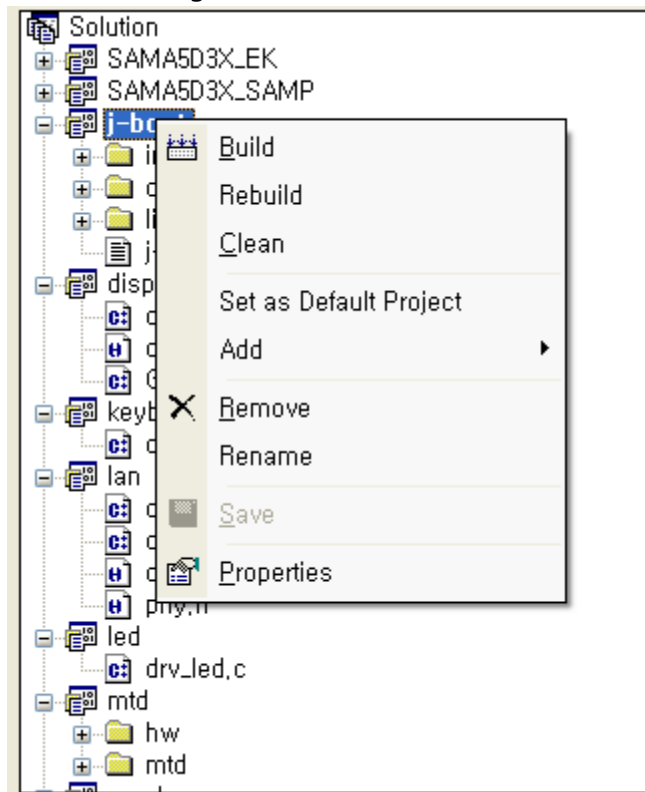
Jboot given is used on the SAMA5D3X_EKBD. The SAMA5D3X_EKBD is a SAMA5D3X development board.

Opent the solution on c:\wdooroos\workspace\sama5d3x_SAMP solution as follows
(File->Open Solution-> Browse)



2.4. Build J-boot

Jboot given is used on the dooroos.realtime SAMA5D3X_EKBD package.
SAMA5D3X_EKBD is a SAMA5D3X development board.
Build it as followings.





```
-
"Compiling .\libWunzip.c"
c:\Program files\yagarto\bin\arm-none-eabi-gcc.exe -march=armv4 -mfloat-abi=soft -Wall -Wstrict-prototypes -O2 -fomit-frame-
pointer -nostdinc -fno-strict-aliasing -pipe -I. -I.\include -I.\include -lc:\doorroos\RealTime\os\include -lc:\doorroos\RealTime
\platform\common\include -lc:\doorroos\RealTime\public\include -c .\libWunzip.c -o .\libWunzip.o
.\libWunzip.c: In function 'malloc':
.\libWunzip.c:134:2: warning: implicit declaration of function 'console_printf' [-Wimplicit-function-declaration]
.\libWunzip.c: In function 'huft_build':
.\libWunzip.c:387:3: warning: implicit declaration of function 'memzero' [-Wimplicit-function-declaration]
.\libWunzip.c: In function 'inflate_codes':
.\libWunzip.c:667:11: warning: implicit declaration of function 'memcpy' [-Wimplicit-function-declaration]
.\libWunzip.c:667:11: warning: incompatible implicit declaration of built-in function 'memcpy' [enabled by default]
.\libWunzip.c: In function 'UncompressKernel':
.\libWunzip.c:1279:14: warning: pointer targets in assignment differ in signedness [-Wpointer-sign]
.\libWunzip.c:1280:8: warning: pointer targets in assignment differ in signedness [-Wpointer-sign]
-
"Linking..."
c:\Program files\yagarto\bin\arm-none-eabi-ld.exe -p -X -Tj-boot.ld -S -Lc:\doorroos\RealTime\os\ARMV4 -Lc:\doorroos
\RealTime\public\ARMV4 --start-group .\init\download.o .\init\main.o .\init\util-comm.o .\init\util-func.o .\init\wrt0.o .\init
\cmd_nand.o .\init\NandFlash.o .\init\ramtest.o .\char\wrzmodem.o .\char\console.o .\char\wxmodem.o .\lib\checksum.o .\lib
\CruntimeLIB.o .\lib\debug_printf.o .\lib\string.o .\lib\Wunzip.o -ldooros_RT_lib -ldooros_RT_interface -lmloder --end-group -o j-
boot.elf -Map j-boot.map
"Generate bin file : j-boot.bin"
c:\Program files\yagarto\bin\arm-none-eabi-objcopy.exe -O binary -R .note -R .comment -S j-boot.elf j-boot.bin
"Done"
C:\doorroos\RealTime\platform\SAMA5D3X_EK\JB00T>cp j-boot.bin c:\doorroos\RealTime\workspace\SAMA5D3X_SAMP
```

You can get the jboot file "j-boot.bin" in the workspace directory "c:\ doorroos\ RealTime\ workspace\ SAMA5D3X_SAMP"

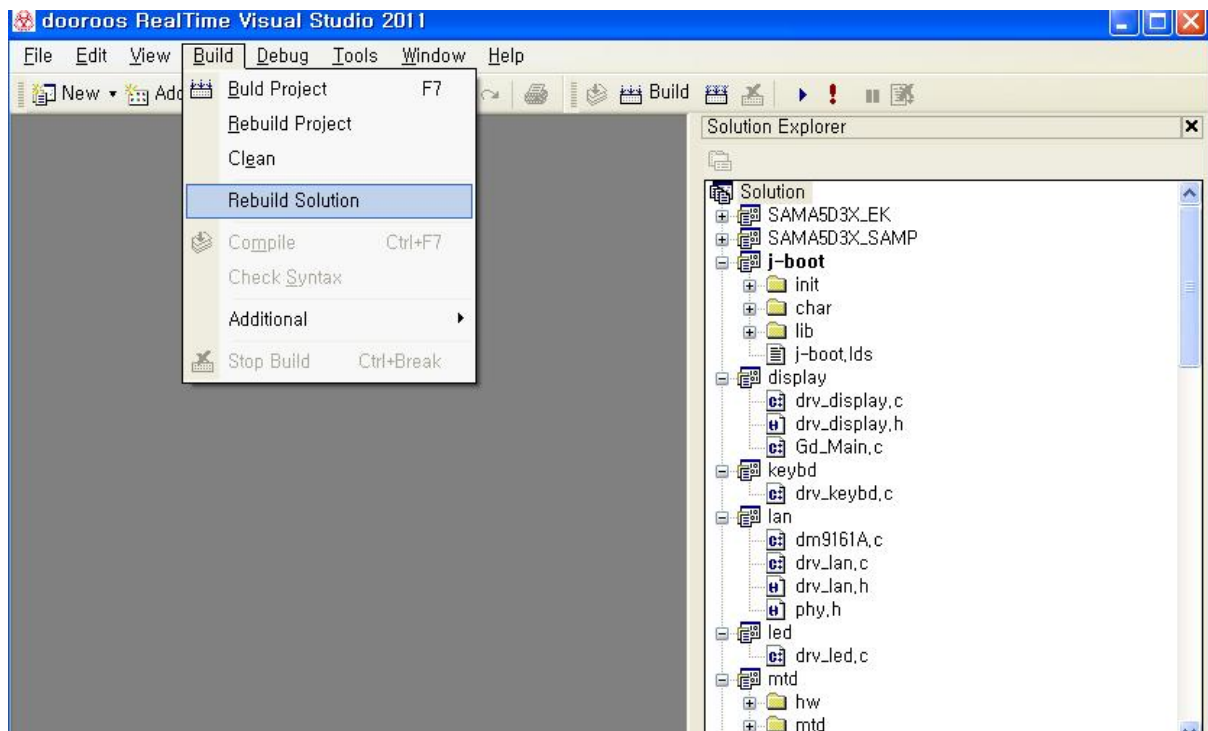
3. Building Image

3.1. dooroos.realtime Image

dooroos.realtime Kernel is given by ".os" files format. See the OS directory (c:\dooroos\os).

[...]	<DIR>	12-06-19 09:02	----
libdooroos_RT_debug	a	11,966 12-07-09 08:54	-a--
libdooroos_RT_hal_debug	a	11,886 12-07-09 08:54	-a--
libdooroos_RT_interface	a	242,178 12-07-09 09:00	-a--
libdooroos_RT_lib	a	1,075,500 12-07-09 08:54	-a--
widgetproc	dll	47,136 12-04-18 17:40	-a--
fs_fat	drv	30,257 12-04-18 17:40	-a--
fs_ram	drv	7,588 12-04-18 17:40	-a--
fs_rom	drv	5,178 12-04-18 17:40	-a--
dooroos_dev	os	6,199 12-07-09 09:00	-a--
dooroos_fs	os	18,356 12-07-09 09:00	-a--
dooroos_loader	os	16,586 12-07-09 09:00	-a--
dooroos_nano	os	20,768 12-07-09 09:00	-a--
dooroos_nw	os	88,595 12-07-09 09:00	-a--
dooroos_sync	os	11,052 12-07-09 09:00	-a--
dooroos_system	os	11,992 12-07-09 09:00	-a--
dooroos_time	os	6,763 12-07-09 09:00	-a--
dooroos_wn	os	44,081 12-07-09 09:00	-a--

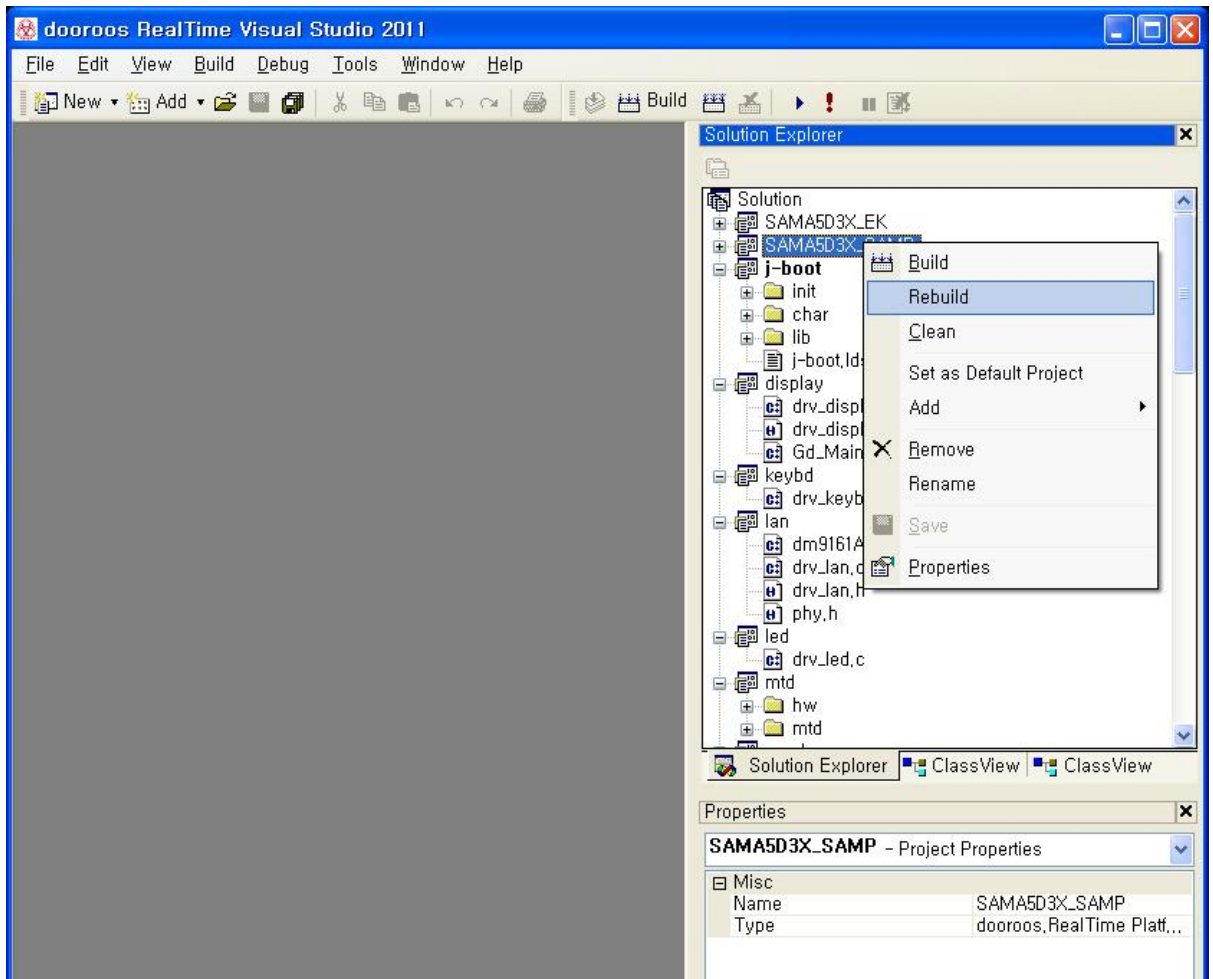
3.2. Build Image



Build projects in the solution as the following procedure:

1. Use rebuild solution menu, then all the projects in the solution is rebuild
 a the drivers and dlls and libraries is generated and copied to the working directory.
2. Finally rebuild the SAMA5D3X_SAMP project again to make download image.
 a The file "dooroos.img" is generated in the working directory.(c:\dooroos\realtime\workspace\sama5d3x_samp\)

3.



```
Output
lc:WdooroosWRealTimeWplatformWcommonWinclude -lc:WdooroosWRealTimeWpublicWinclude -c .WconfigWkernel_main.c -o .Wconfig
Wkernel_main.o
-
"Compiling .W..WplatformWcommonWdriverWblockWblocklibWformat_fat_2048.c"
c:WProgram filesWygartoWbinWarm-none-eabi-gcc.exe -march=armv6 -mfloat-abi=soft -Wall -Wstrict-prototypes -O2 -fomit-frame-
pointer -nostdinc -fno-strict-aliasing -pipe -D_AT91_ -I. -I..W..WplatformWSAMA5D3X_EKWinclude -lc:WdooroosWRealTimeWosWinclude -
lc:WdooroosWRealTimeWplatformWcommonWinclude -lc:WdooroosWRealTimeWpublicWinclude -c .W..WplatformWcommonWdriverWblock
WblocklibWformat_fat_2048.c -o .W..WplatformWcommonWdriverWblockWblocklibWformat_fat_2048.o
-
"Linking..."
c:WProgram filesWygartoWbinWarm-none-eabi-ld.exe -p -X -e stext -T ./config/platform.lds -Lc:WdooroosWRealTimeWosWARMV6 -Lc:
WdooroosWRealTimeWpublicWARMV6 --start-group .WconfigWapp_main.o .WconfigWkernel_main.o .W..WplatformWcommonWdriver
WblockWblocklibWformat_fat_2048.o -ldooros_RT_hal_debug -ldooros_RT_interface -ldooros_RT_lib -ltaskheap .W..Wplatform
WSAMA5D3X_EKWSAMA5D3X_EK.a --end-group -o ./release/platform.elf -Map ./release/platform.map
-
"Generate bin file : platform.bin"
c:WProgram filesWygartoWbinWarm-none-eabi-objcopy.exe -O binary -R .note -R .comment -S ./release/platform.elf
./release/platform.bin
-
Copy the dooroos.RealTime files to the download image directory "images"
-
Copy the file "platform.bin" to the download image directory "images"
-
Make dooroos.RealTime image... : dooroos.img
genromfs -f dooroos.img -d images
"Done"
```


4. Nand flash fusing with SAM-BA

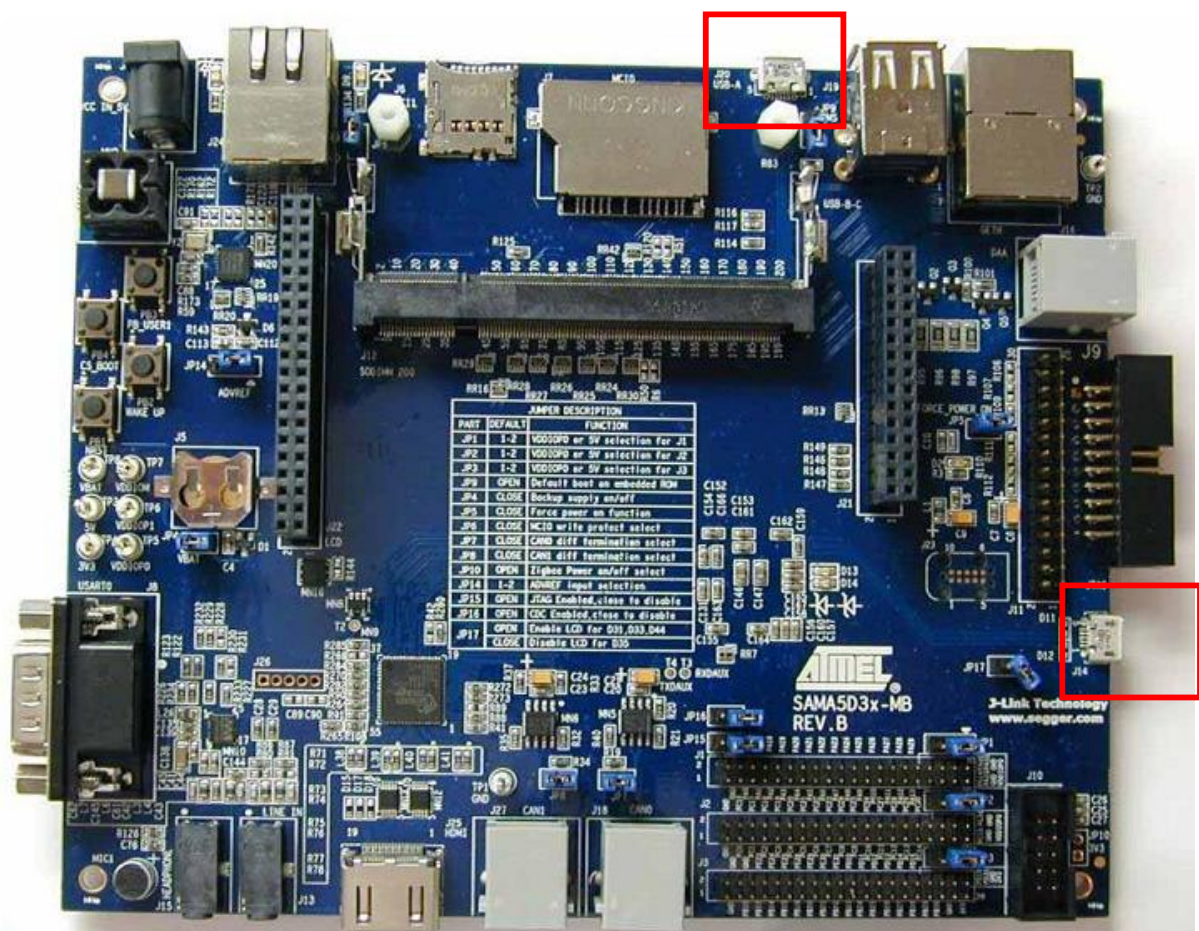
To write the boot file "sama5d3xek-nandflashboot-uboot-3.5.4.bin", the boot-loader file "j-boot.bin" and the doorooms image file "doorooms.img" to the nandflash, you have to use the SAM-BA tools of the ATMEL.

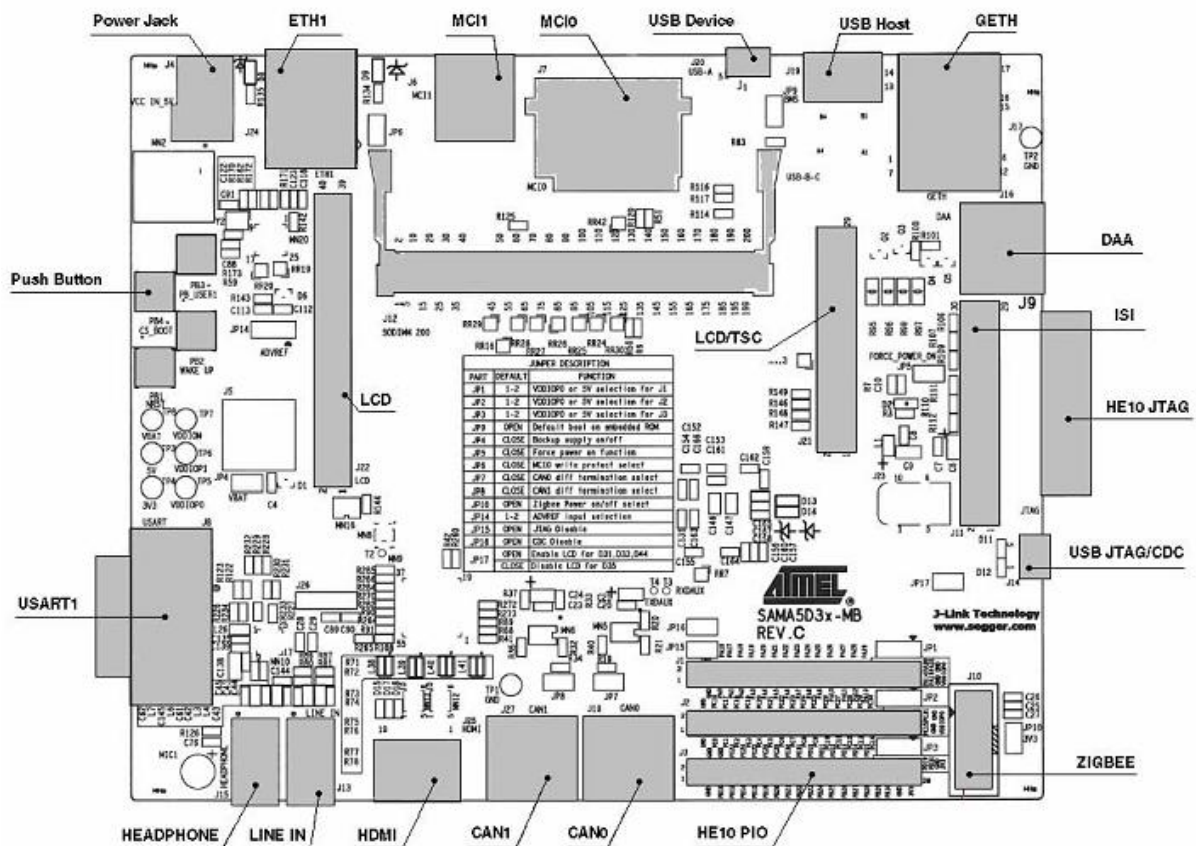
Please download the SAM-BA from the atmel homepage(www.atmel.com)

First you should install the SAM-BA usb driver on your system.

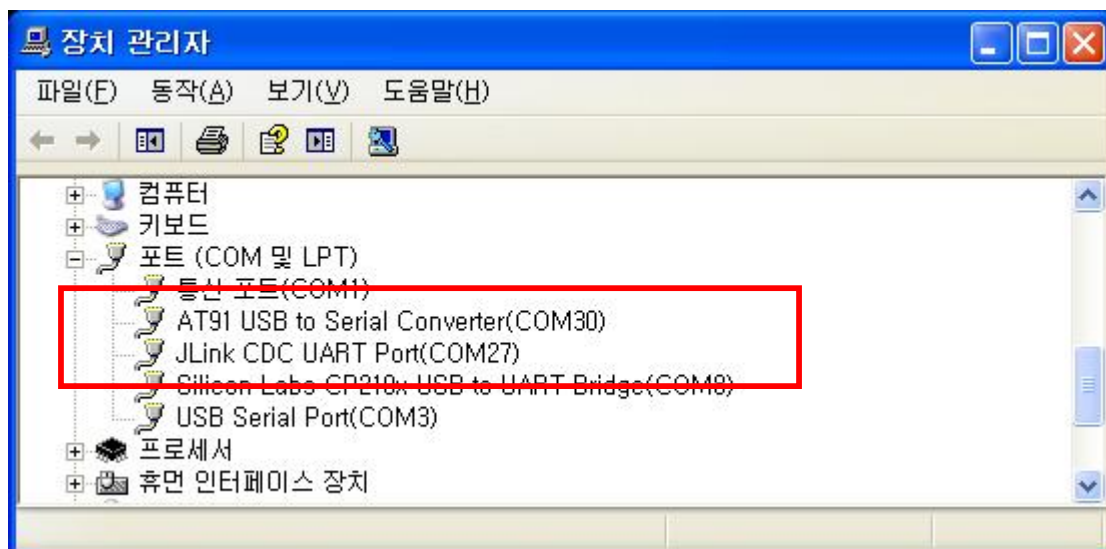
Second, you should connect the debug serial cable(115200. no parity, 1 stop bit,, 8bit), and usb device cable with PC on J14 by USB cable (USB JTAG / CDC).

The connector J20 (USB-A) is the SAMA5D3X_EKBD usb device port.



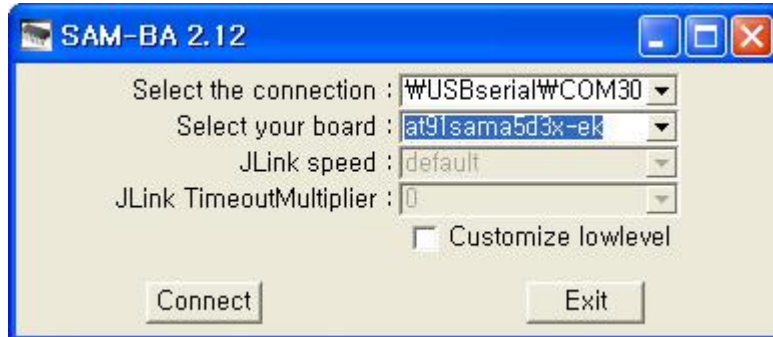


CPU board has the some setting jumper pins. To boot on usb mode, the button BP4 is kept pressed and reset(press the button BP1). You can see the following message in the device manager in the window, if you use SAM-BA V2.12.

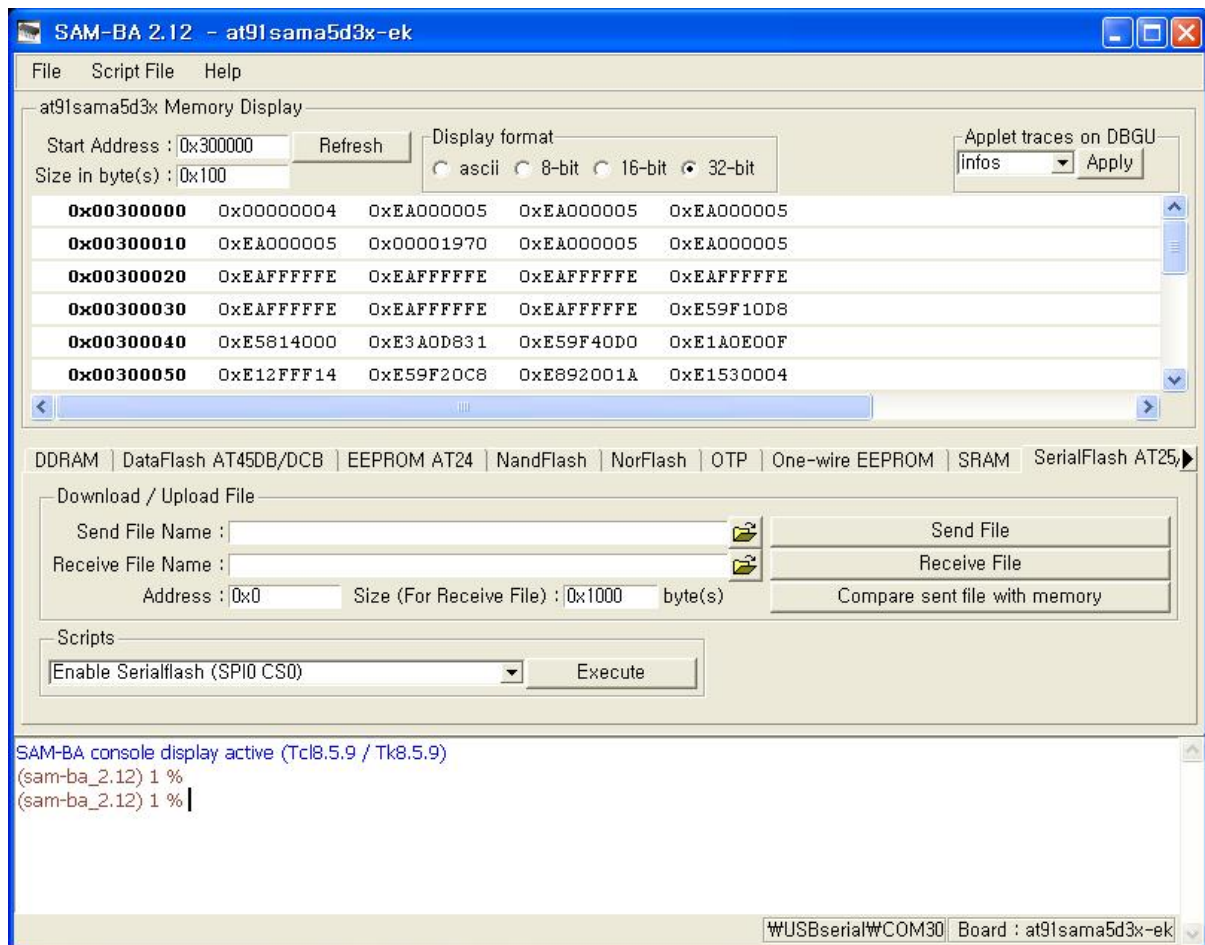


If you get the usb connection, then release the botton BP4 to write the images on the nand-flash.

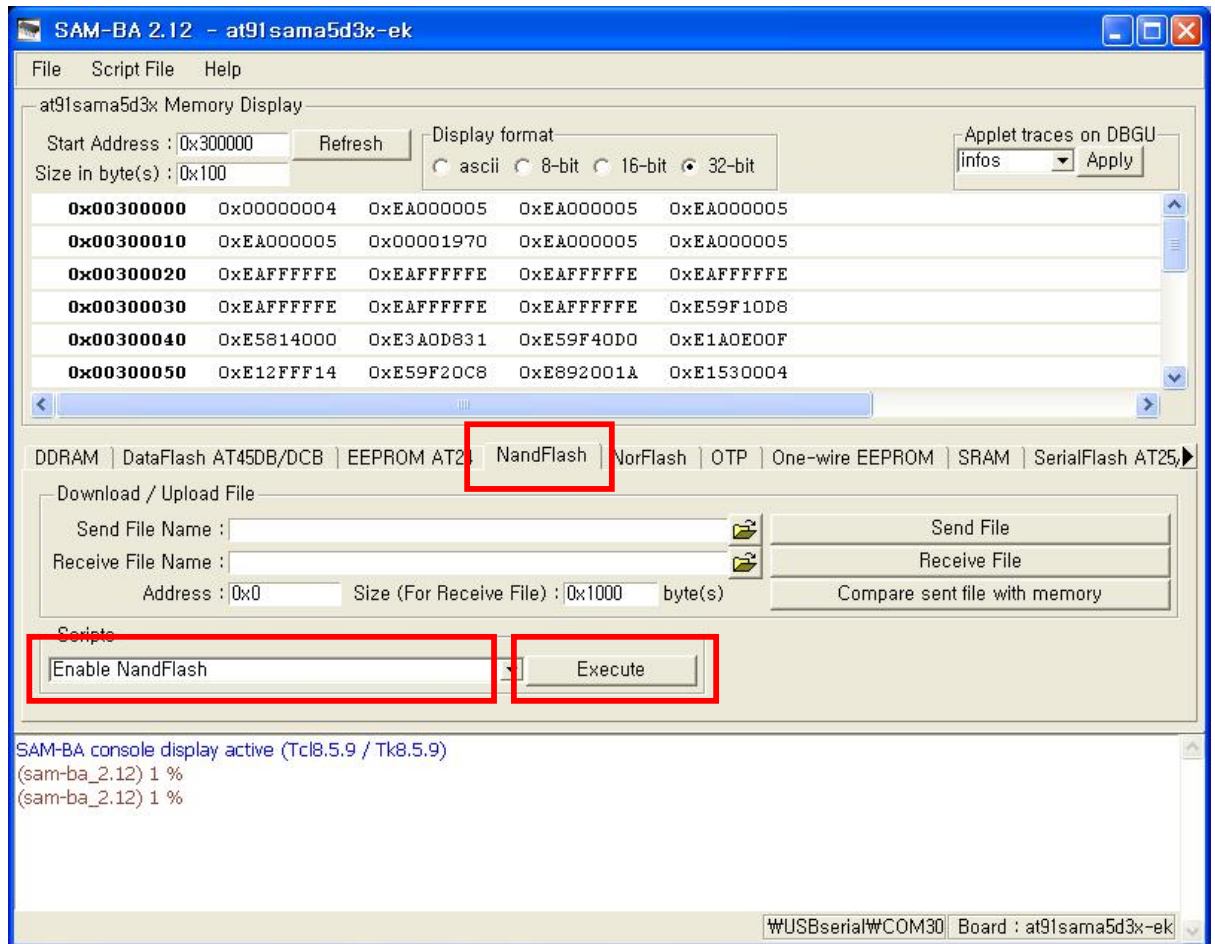
4.1. Run SAM-BA

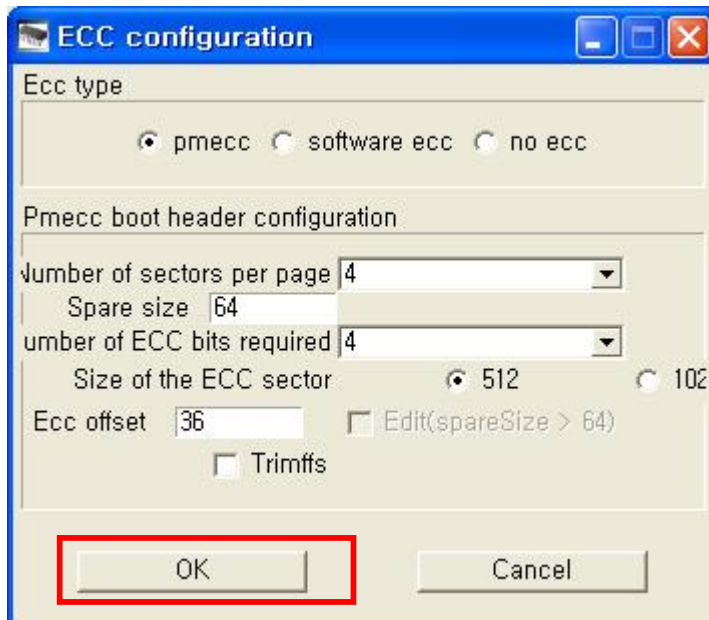
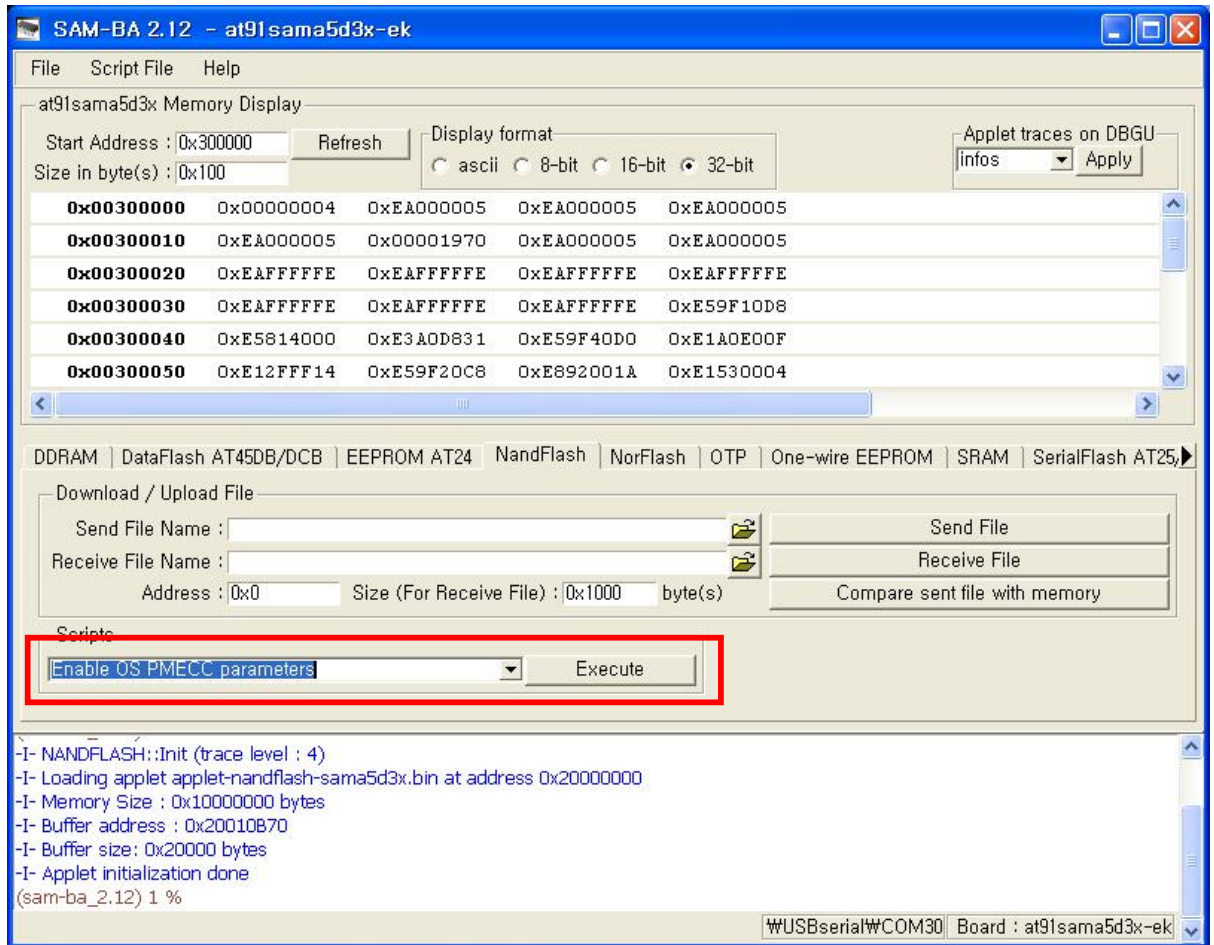


1. Run the sam-ba_2.12.exe
2. Select the usb port and board as the above
3. Press connect.
4. You can get the following window..

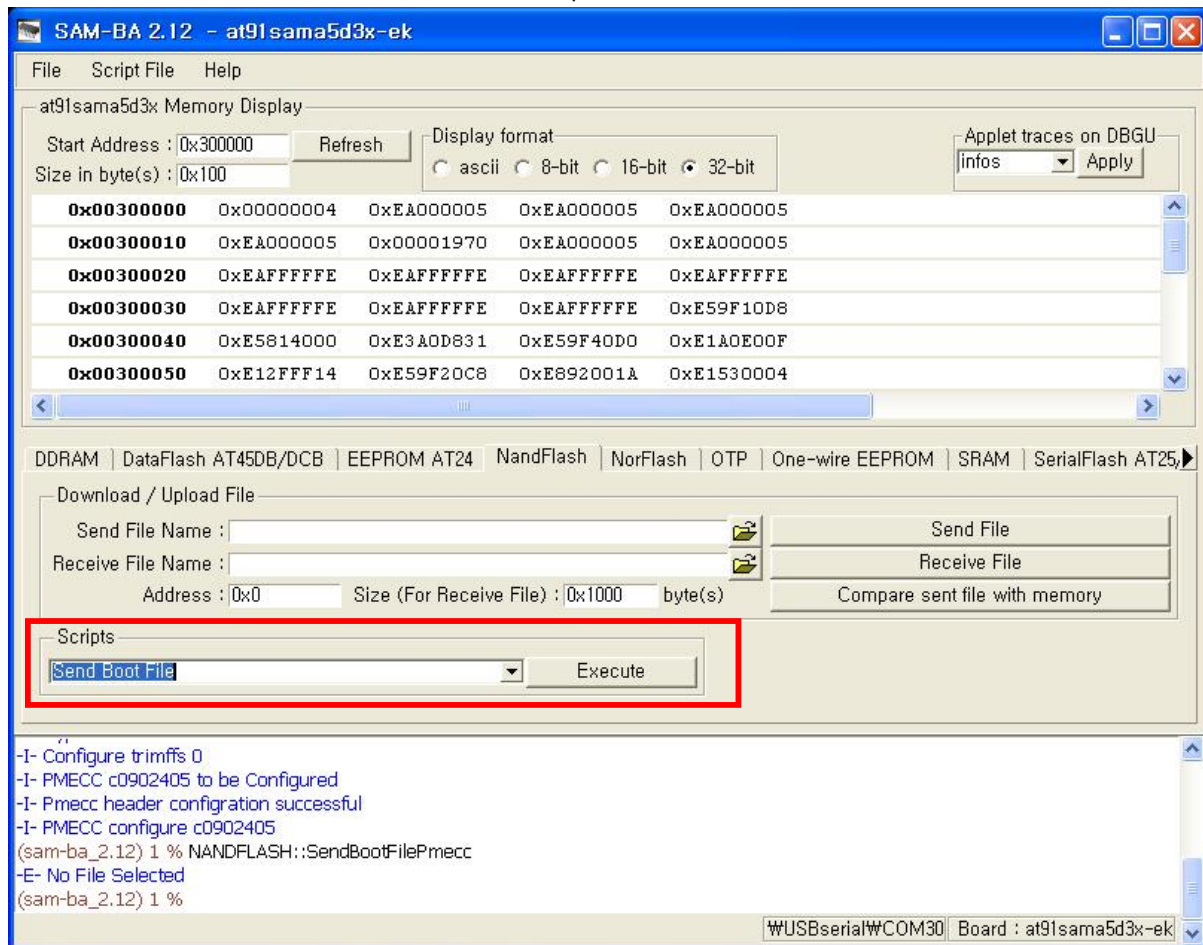


5. Select the Nandflash tab and initialize the nand flash device
 - select the tab NandFlash
 - Press the "button Execute"

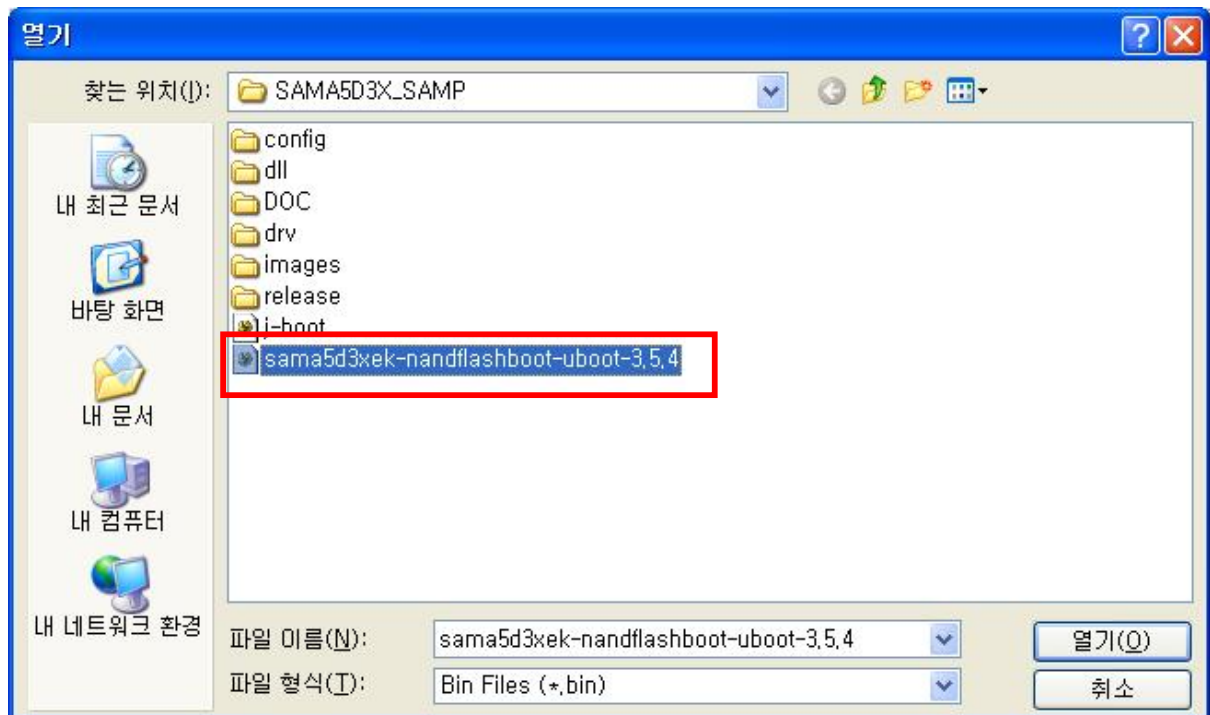




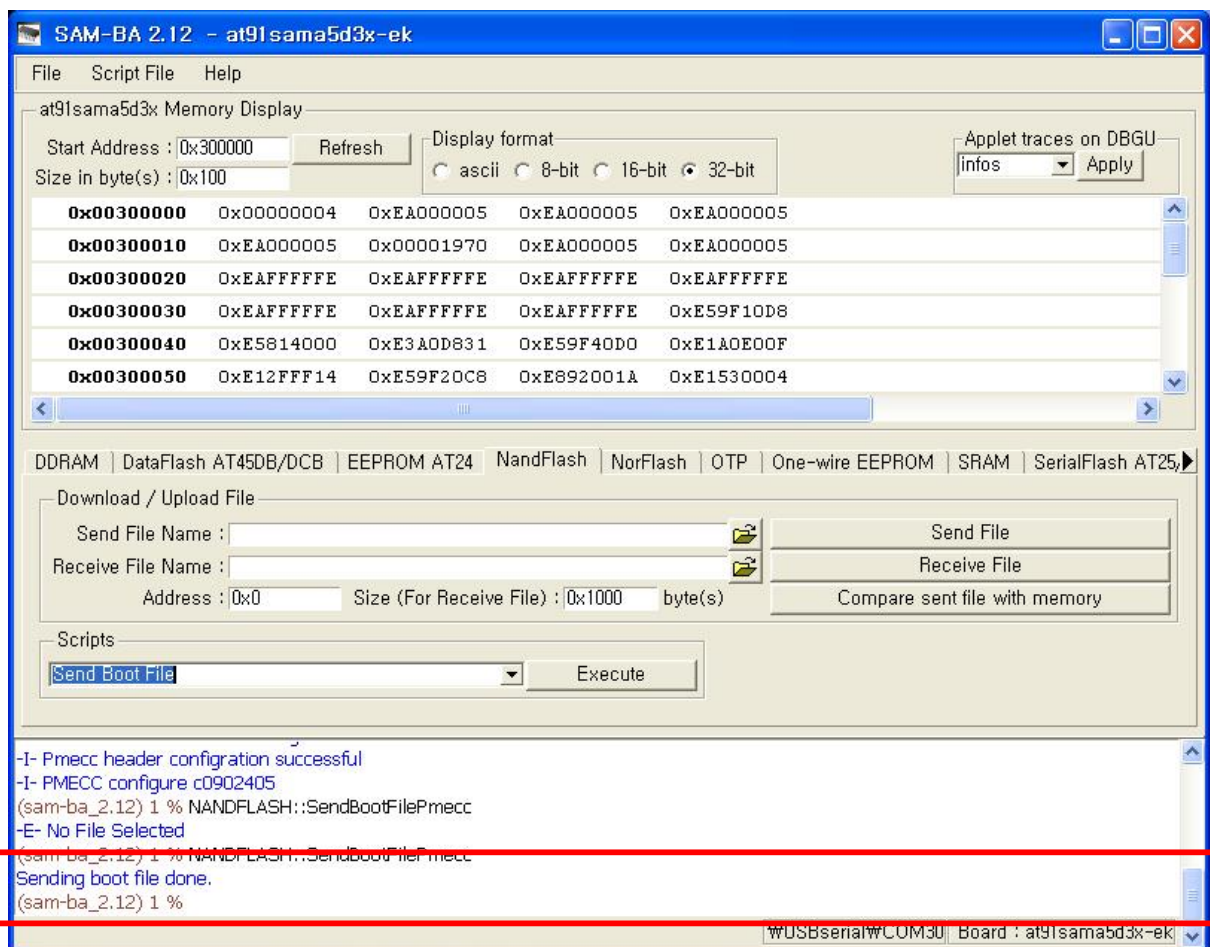
6. Let's write the file "sama5d3xek-nandflashboot-uboot-3.5.4.bin"
 - Select the Send Boot file and press the Execute button



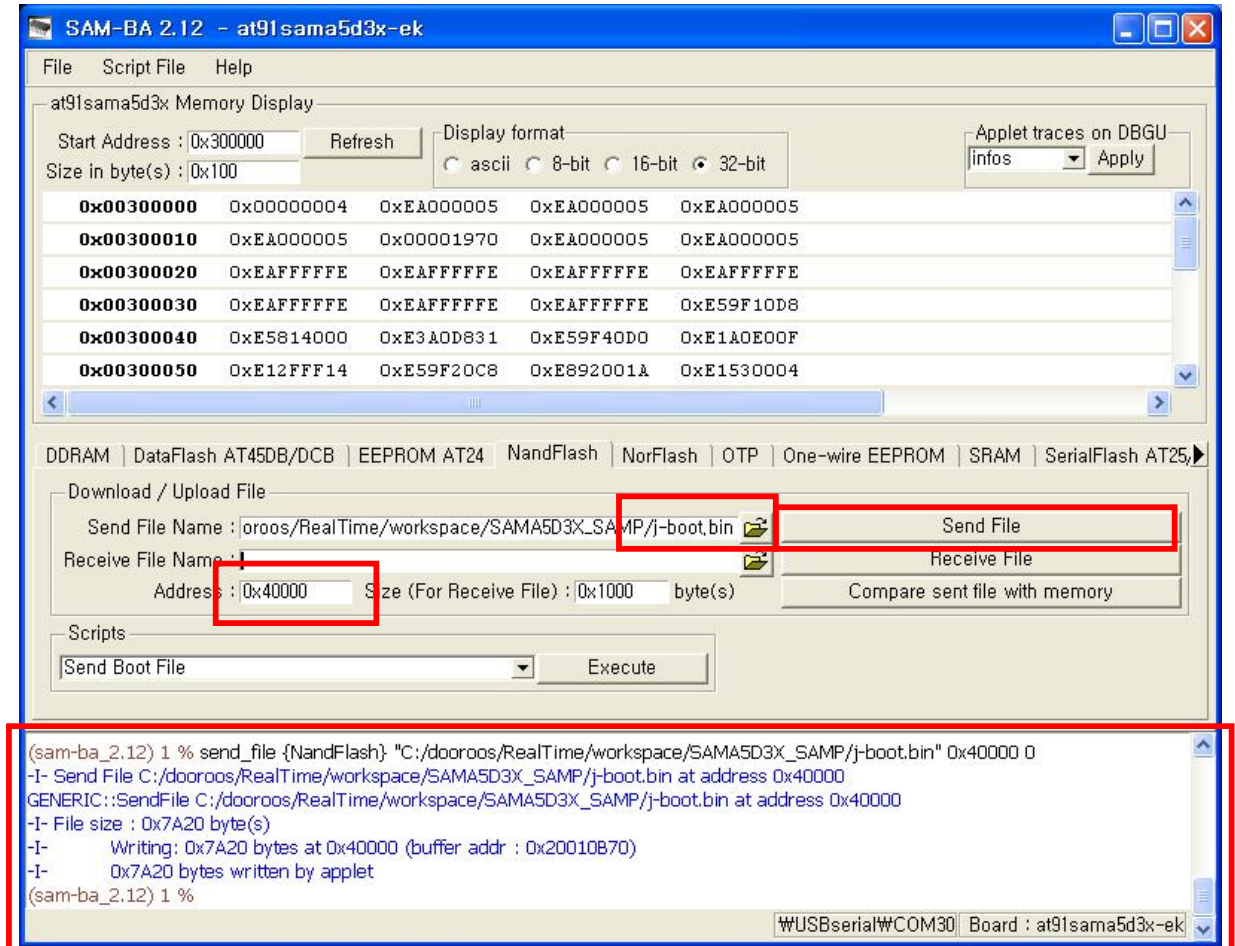
7. The new dialog is opened and select the file "sama5d3xek-nandflashboot-uboot-3.5.4.bin"



8. Then the file is written.

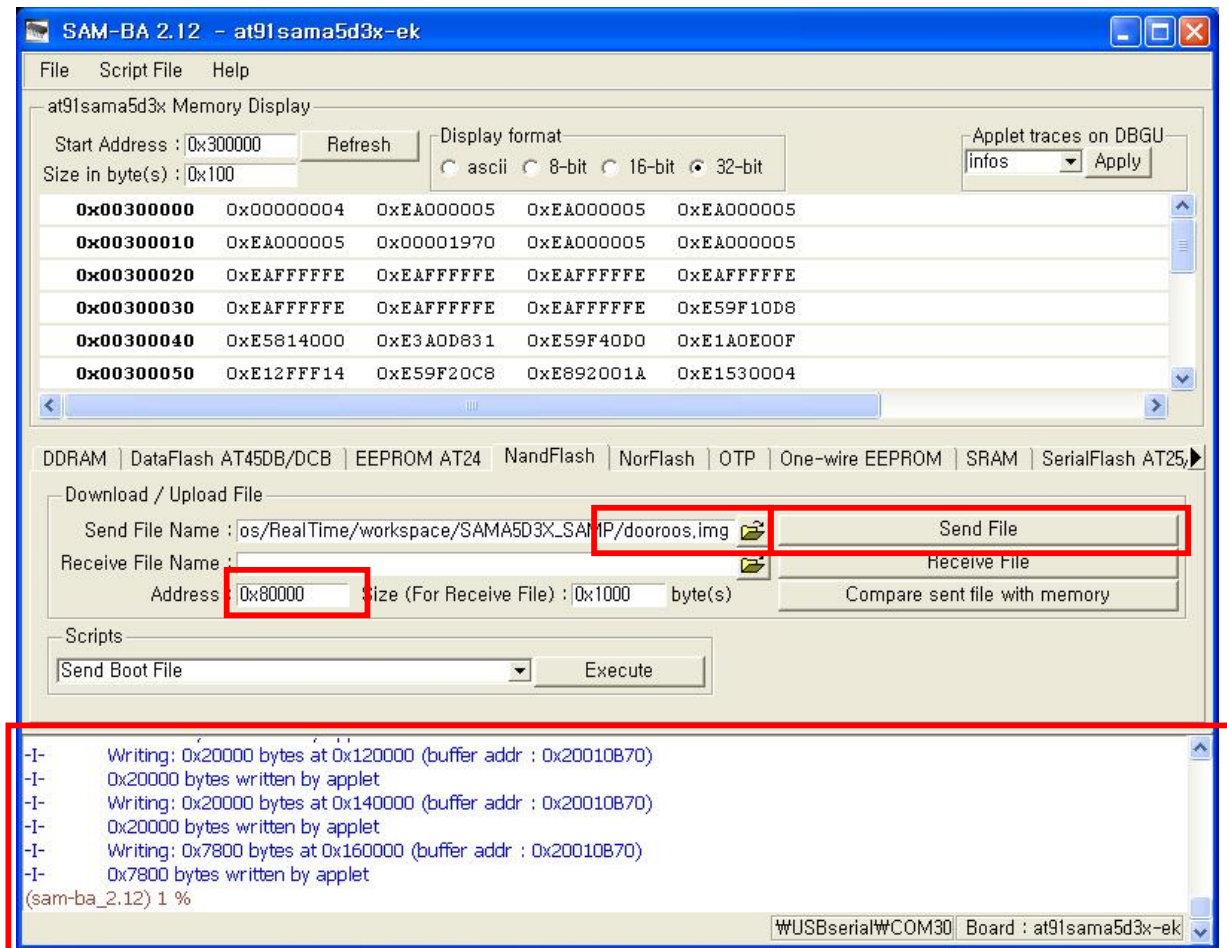


9. Now Let's write the file "j-boot.bin"
 - write the address 0x40000 on address filed
 - select the j-boot.bin
 - press the button "Send File"
 - Then you can get the output message in the output window



10. Let's write the dooroos.realtim image file "dooroos.img".

- write the 0x80000 on address filed
- select the file "dooroos.img"
- press the button "Send File"
- you can get the following output on the output window




```
download the platform.bin : bRet = 80e4
-> SUCCESS
```

[illegible]

[illegible]

th_AppLoader (173): 935f8
RegisterDrivers DSK1
th_AppLoader (178): 938f0
th_AppLoader (204): 953f8
th_AppLoader (207): 954a0
th_AppLoader (211): 1
th_AppLoader (219): 955f0
th_AppLoader (222): 95698
th_AppLoader (228): 93be8
RegisterDrivers USB_DEVICE Class driver for Mass Storage

4.2. Download the filesystem into nand flash by USB

1. now the SAMA5D3X_EKBD board is booting.
2. But if it is your first boot of dooroos.realtime, please wait a while.
At the first boot, the nand flash should be formatted.
3. Now you can see the usb connection message on the window PC. The new usb disk drive is attached automatically.
4. Format the new drive(for ex. d:) with FAT filesystem.
5. Copy all the directories (c:\dooroos\realtime\workspace\sama5d3_samp\images_disk\) into the drive(d:)
6. Copy all the directories and files ("c:\dooroos\realtime\workspace\EMULATOR_SAMP\images\nand) into the drive(d:) except the directories "dll", "drv"

Reboot.....

dooroos.realtime on your LCD
please touch them.