APPLICATION NOTE

Atmel AT03454: SAM-BA for SAM4L

Atmel ATSAM4L

Features

- Allows to program, verify and secure an Atmel® SAM4L device without debugger
- USB connection (CDC device)
- UART connection (RS232)
- Allows the end user to update firmware in application
- Configurable I/O start conditions
- Code security management
- Source code available, can be customized to user's needs (Encrypted loader, custom protocol)
- Miscellaneous topics
- Suggested readings

Introduction

SAM4L devices come pre-loaded with the Atmel SAM Boot Assistant (Atmel SAM-BA®) in flash memory. This bootloader allows In-System Programming (ISP) from USB or UART host without any external programming interface.

This SAM-BA version is compatible with existing SAM-BA software tools but has some differences compared to other SAM devices. These differences are explained in this document.

This application note complements the SAM-BA user guide and explains how the SAM-BA should be used on a SAM4L design.

1. Requirements

1.1 Hardware requirements

The Atmel ATSAM4L SAM-BA Boot Assistant supports serial communication via the UART or USB device port:

Table 1-1. UART mode requirements.

Signal name	Recommended pin connection	Description
VDDIN	≥1.8V	Programming the flash requires the chip to be in power scaling mode 0. The minimum VDDIN voltage for PS0 operation is 1.8V (Refer to the device datasheet for more details).
PA05	Connect to host: (115200 8,N,1)	USART0 RXD pin.
PA07	Connect to host: (115200 8,N,1)	USART0 TXD pin. Note: This pin is not driven until the USART mode is entered by receiving a “#” (sharp) character.

Table 1-2. USB mode requirements.

Signal name	Recommended pin connection	Description
VDDIN	3.3V	USB requires a VDDIN of 3.3V.
D+ (PA26)	Connect to host	USB D+ pin
D- (PA25)	Connect to host	USB D- pin
XIN / XOUT	XTAL or external digital clock input: Supported frequencies: 6MHz ±2500ppm 7.3728MHz ±900ppm 8MHz ±2500ppm 12MHz ±2500ppm 14.7456 MHz ±900ppm 16MHz ±2500ppm 24MHz ±2500ppm	Clock input for PLL used for USB clock source.

Table 1-3. Common optional features.

Signal name	Recommended pin connection	Description
Hardware Bootloader Entry (Any GPIO)	Connect to host or “bootloader access switch”	Optional pin to force bootloader entry on reset. If enabled, the bootloader will check the GPIO on reset to determine if the bootloader monitor shall start. This feature allows the end user to reprogram the device even if the application is corrupted or unable to start the SAM-BA monitor. This GPIO number and its active level are defined in the flash user page. The SAM-BA application can be used to program it [1]. Note: The bootloader won’t pull up or pull down the pin, the level has to set externally.

Notes: 1. Please refer to 2.3.2 Force pin selection.

1.2 Software requirements

1.2.1 Application constrains

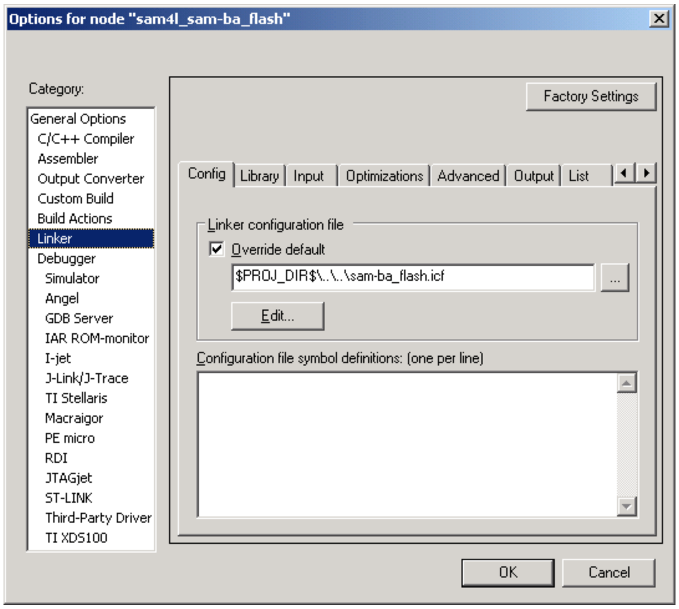
Unlike existing SAM products using a ROM monitor, on the Atmel ATSAM4L SAM-BA is stored in flash memory at [0x0 – 0x4000] and started on reset.

Since SAM-BA is stored in flash memory, if the application requires the entire flash space and does not need the bootloading feature, SAM-BA can be erased using a JTAG/SWD debugger.

To use SAM-BA together with an application the user needs to link the application starting at 0x4000. (See memory map below).

The linker files for IAR™ and Atmel Studio are provided with this application note.

Figure 1-1. Changing linker script in IAR.



Linker configuration file

☑ Override default

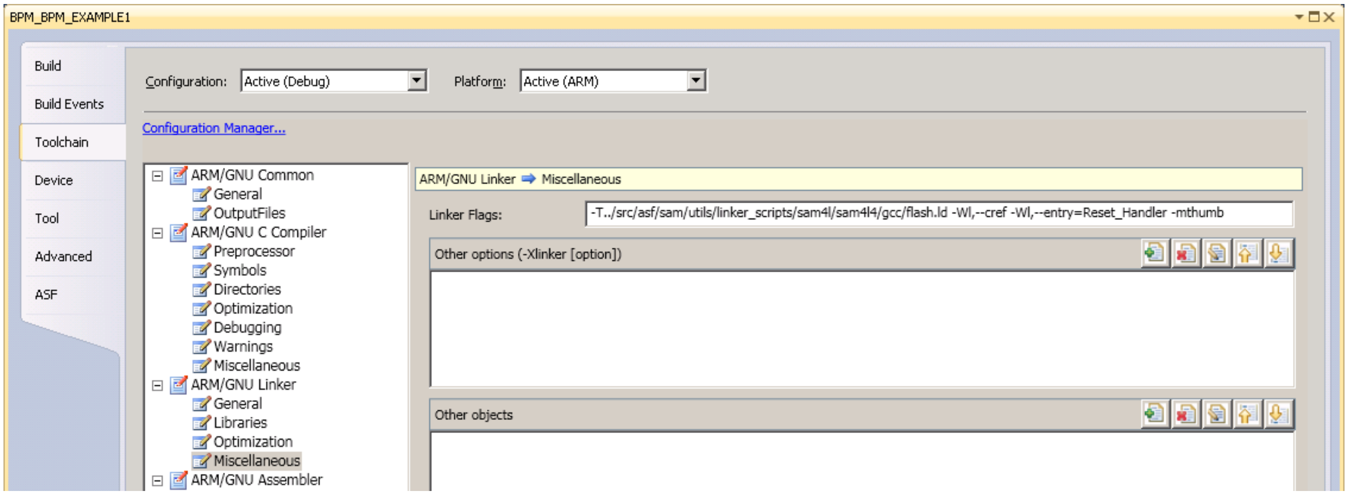
$PROJ_DIR$\..\..\sam-ba_flash.icf

Configuration file symbol definitions: (one per line)

To change the linker script under IAR project:

- Menu Project, Options …
- Go to linker tab
- Check “override default”
- Fill the path to the linker file provided with this application note
- Rebuild the project

Figure 1-2. Changing linker script in Atmel Studio.



Configuration: Active (Debug)

Platform: Active (ARM)

ARM/GNU Linker ➡ Miscellaneous

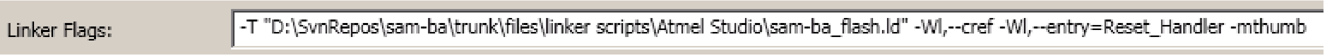
Linker Flags: -T../src/asf/sam/utils/linker_scripts/sam4l/sam4l4/gcc/flash.ld -Wl,--cref -Wl,--entry=Reset_Handler -mthumb

Other options (-Xlinker [option])

Other objects

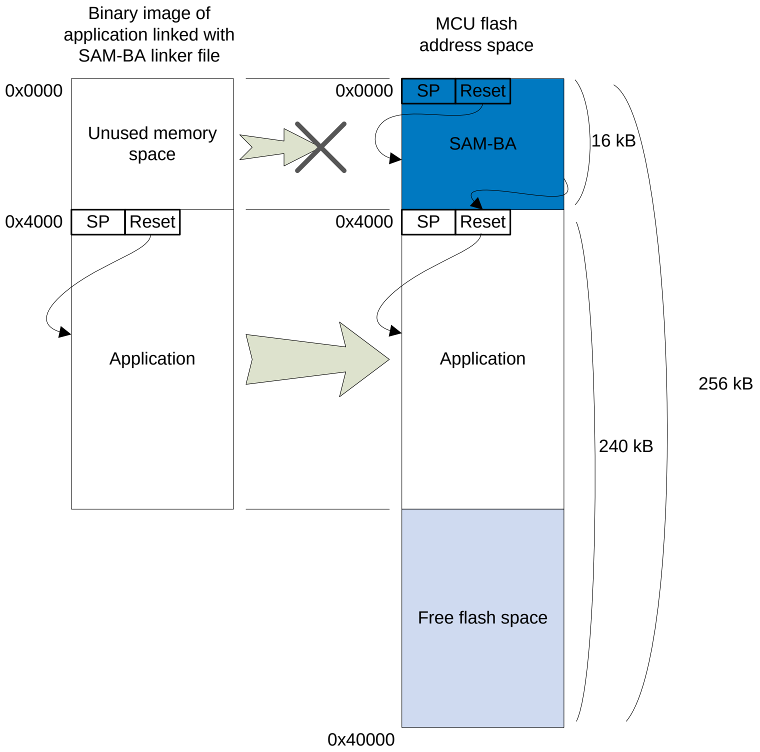
To change the linker script under Atmel Studio:

- Menu Project, Properties,
- Toolchain tab, ARM®/GNU Linker, Miscellaneous
- Fill the path to linker script provided with this application note after the “-T” option:
- (e.g.: -T "D:\SvnRepos\sam-ba\trunk\files\linker scripts\Atmel Studio\sam-ba_flash.ld")



Linker Flags: -T "D:\SvnRepos\sam-ba\trunk\files\linker scripts\Atmel Studio\sam-ba_flash.ld" -Wl,--cref -Wl,--entry=Reset_Handler -mthumb

Figure 1-3. Memory map of an ATSAM4Lx4 device with an application and SAM-BA.



By default, the Atmel SAM-BA flash region [0x0 ... 0x4000] is locked by lock bit 0 on every device shipped.

When loading the binary image into the device, only the upper part of flash starting at 0x4000 address should be programmed.

Any attempt to write to the SAM-BA region using SAM-BA commands will be aborted and will throw an error.

Figure 1-4. Memory map of an ATSAM4Lx8 device with an application and SAM-BA.

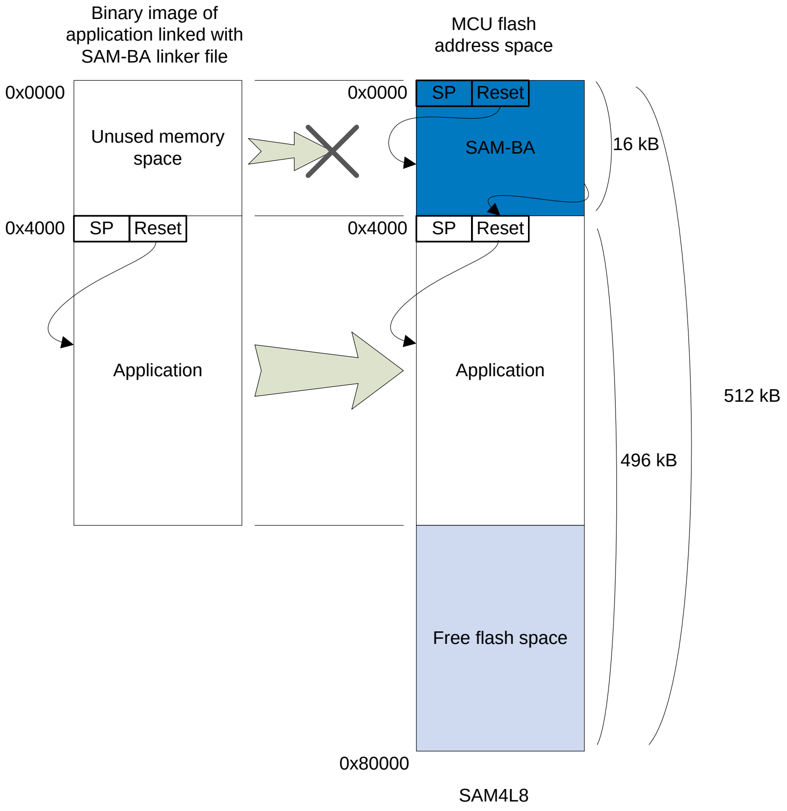
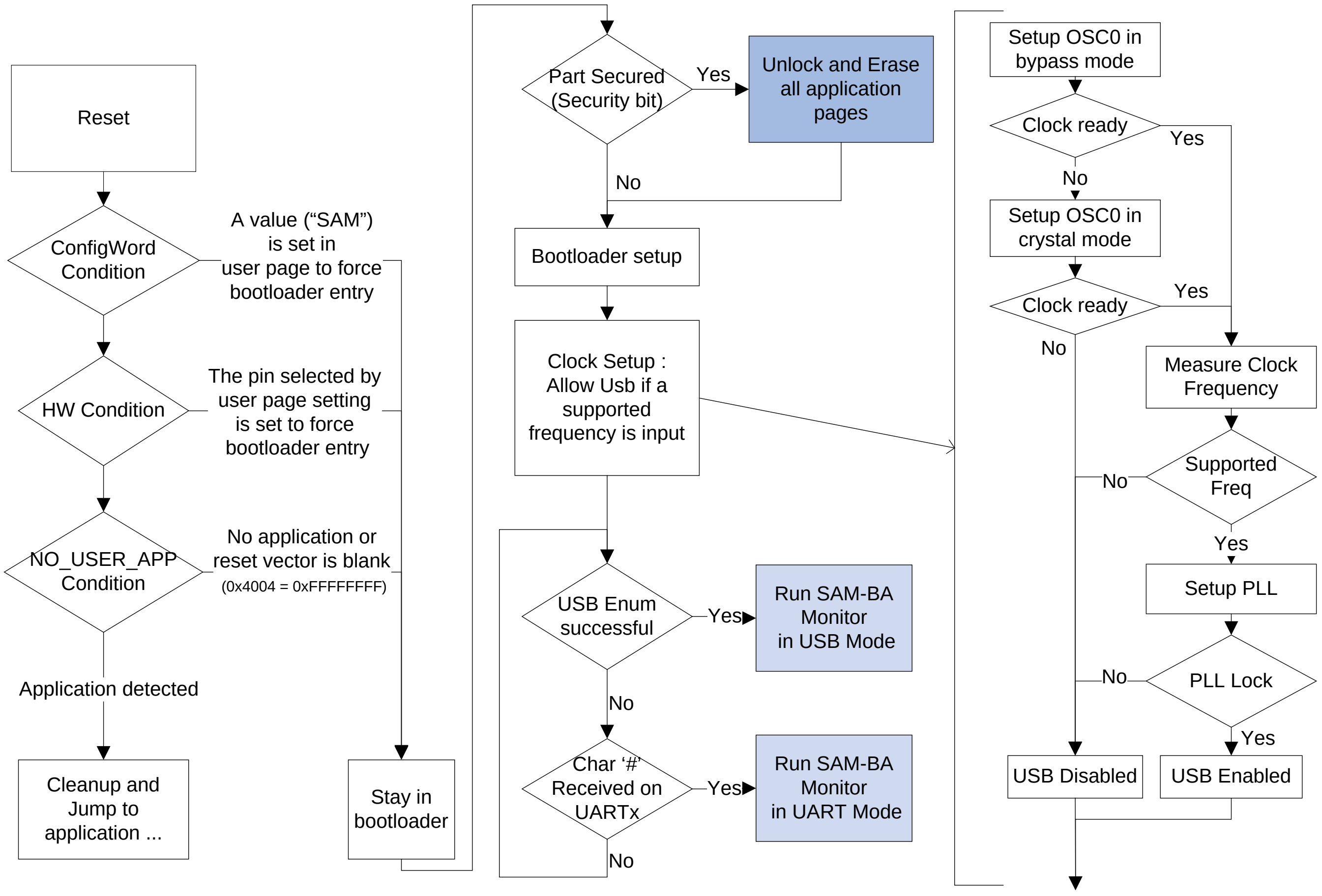


Figure 1-5. Detailed boot process of the Atmel SAM-BA.



Reset
ConfigWord Condition
A value ("SAM") is set in user page to force bootloader entry
HW Condition
The pin selected by user page setting is set to force bootloader entry
NO_USER_APP Condition
No application or reset vector is blank
(0x4004 = 0xFFFFFFFF)
Application detected
Cleanup and Jump to application ...
Stay in bootloader
Part Secured (Security bit)
Yes
Unlock and Erase all application pages
No
Bootloader setup
Clock Setup : Allow Usb if a supported frequency is input
USB Enum successful
Yes
Run SAM-BA Monitor in USB Mode
No
Char '#' Received on UARTx
Yes
Run SAM-BA Monitor in UART Mode
No
Setup OSC0 in bypass mode
Clock ready
Yes
No
Setup OSC0 in crystal mode
Clock ready
Yes
No
Measure Clock Frequency
Supported Freq
No
Yes
Setup PLL
PLL Lock
No
Yes
USB Disabled
USB Enabled

2. Using the bootloader

2.1 Reprogramming the bootloader

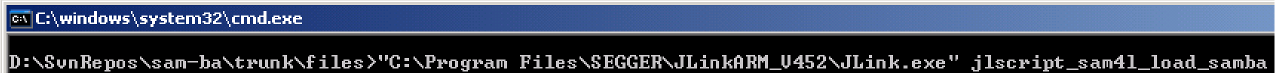
By default, all parts are shipped with the bootloader, so there is no need to program it, except if it has been erased by a jtag/swd debugger or if the user wants to program another version. This step requires a jtag/swd debugger.

2.1.1 Steps to reprogram the bootloader using J-Link

a) Extract the two files included in the application note folder named “jlscript_sam4l_load_samba” and “sam4l_sam-ba_image.bin” in a folder. The script erases the entire chip, loads the Atmel SAM-BA image and locks the SAM-BA flash region.

b) Open a command line box and cd to the folder where the two files were extracted.

c) Call JLink.exe with the script name in argument: “Jlink jlscript_sam4l_load_samba” (if the J-Link folder is not in the path system variable, the complete path to J-Link executable is needed:



```
D:\SvnRepos\sam-ba\trunk\files>"C:\Program Files\SEGGER\JLinkARM_V452\JLink.exe" jlscript_sam4l_load_samba
```

If all went well, the script exits after a few seconds.

2.1.2 Steps to reprogram the bootloader using Atmel Studio

a) Extract the two files “jlscript_sam4l_load_samba” and “sam4l_sam-ba_image.bin”.

b) In Atmel Studio, open “Tools\Device programming”, select debugger and Apply.

c) From the memories tab, launch a chip erase.

d) Fill the path to “sam4l_sam-ba_image.bin” in the Flash box and program.

e) In the Fuses tab, check LOCK0 bit and program.

2.2 Entering the bootloader

Like other Atmel SAM devices, the SAM4L SAM-BA relies on a monitor; this monitor is entered according to the boot process conditions described in Figure 1-5.

SAM-BA monitor activation can be requested in one of the following ways:

- External condition: Reset the part and make sure the configured hardware conditions are true when reset is released. A common usage is to use a pushbutton accessible by the user as a bootloader trigger. The user simply has to hold the push button down when powering up the device
- Internal point of view:
  - The programmed application can launch the monitor by setting a specific value in the config word of the flash user page. The next execution of the reset vector will then systematically launch the monitor. To launch the boot process from the application, the reset vector should be reached by using the watchdog timer reset rather than a software jump or direct call to the monitor. In the later case, unexpected behavior could occur because the MCU reset causes are not updated and MCU peripherals may still be active
  - On erased devices or when the application reset vector (@0x4004) is blank (0xFFFFFFFF)

> Once the monitor is triggered, it checks the security bit. If security bit is set, the monitor will erase the entire application to prevent any unauthorized access to protected code. If a different behavior is needed, a custom bootloader can be created from the SAM-BA bootloader source code found in this application note.

Code security concerns: when monitor is entered, it allows read and write access to the entire memory map of the device. It also allows the host to upload and execute software (applets) on the device.

After these preliminary steps, the monitor will try to identify a suitable clock source for USB operation. If one of the supported clock or XTAL is found on XIN/XOUT, the USB I/Os and device interface are enabled.

Then, the monitor will enter a loop and test whether an USB enumeration has succeeded or a “#” (sharp) character is received on the PA05 (USART0 RX) line. The first satisfied condition will start the monitor in the respective mode: UART or USB.

When UART monitor mode is started, the USART0 TX line (PA07) becomes an output and is driven by the SAM4L.

2.3 Monitor settings (Config Word)

The monitor config word is located at offset 0x10 of the flash user page (Absolute address 0x00800010).

Please note that the flash user page also contains BODs and watchdog settings (refer to “fuse settings” in the FLASHCALW chapter of the device datasheet). To keep the user page settings the user should save the content of the flash user page, modify the content and write it back to the device.

This config word serves three purposes:

2.3.1 Launch monitor on reset

When config word[23..0] equals 0x4D4153 (“SAM” in ASCII), the monitor will be entered when device is reset. This setting takes priority over force pin feature. This can be used to start the monitor using a software initiated reset without any hardware action.

2.3.2 Force pin selection

Using a force pin needs a debounce period at power-on reset to avoid any spurious reading when power supply is not yet stable. On Atmel ATSAM4L SAM-BA, this debounce period is set to 100ms. Consequently, using the force pin feature will delay the boot process by 100ms. This has to be considered when estimating wake up time from a reset or from backup mode.

The format of the config word is the following:

Figure 2-1. Configuration word in flash user page.

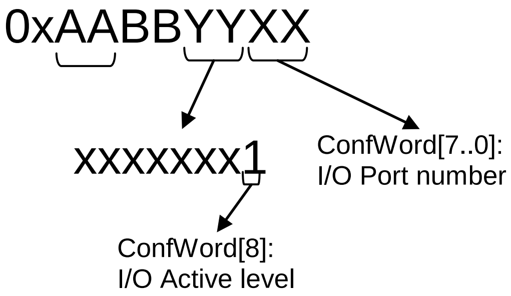


Table 2-1. Config word fields.

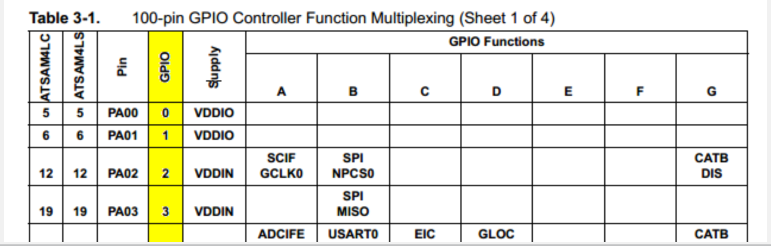
Field	Bits	Description
I/O port number	[7..0]	Selects the I/O port which will force entering the monitor on boot. The pin name mapping to the GPIO number can be found in the device datasheet in the “Peripheral multiplexing on I/O lines” section in the Package and pinout chapter: 
I/O active level[1]	[8]	Selects the I/O active level which will force entering the monitor on boot. BIT8 = 0 selects low level. (i.e. monitor entered when force pin is low) BIT8 = 1 selects high level. (i.e. monitor entered when force pin is low) On reset and after a debounce period of 100ms, Atmel SAM-BA will sample the I/O selected by the I/O port number field. The pin has to be set externally to a stable level (Atmel ATSAM4L does not pull up or pull down this line).

Table 3-1. 100-pin GPIO Controller Function Multiplexing (Sheet 1 of 4)

ATSAM4LC	ATSAM4LS	Pin	GPIO	Supply	GPIO Functions						
					A	B	C	D	E	F	G
5	5	PA00	0	VDDIO							
6	6	PA01	1	VDDIO							
12	12	PA02	2	VDDIN	SCIF GCLK0	SPI NPCS0					CATB DIS
19	19	PA03	3	VDDIN		SPI MISO					
					ADCIFE	USART0	EIC	GLOC			CATB

Notes: 1. On a secured device, entering the monitor erases all application memory. If the force pin is used, user must ensure this pin is stable after the 100ms debounce period.

Examples of config word values:

- Using PA02 as a force pin active low,
  - the config word value will be: 0xFFFFFE02
- Using PC31 as a force pin active high,
  - the config word value will be: 0xFEFFFF5F
- Without the force pin feature
  - the config word value will be: 0xFFFFFFFF
- Monitor always started at boot
  - the config word value will be: 0xFF4D4153

2.3.3 How to set the config word value using SAM-BA application

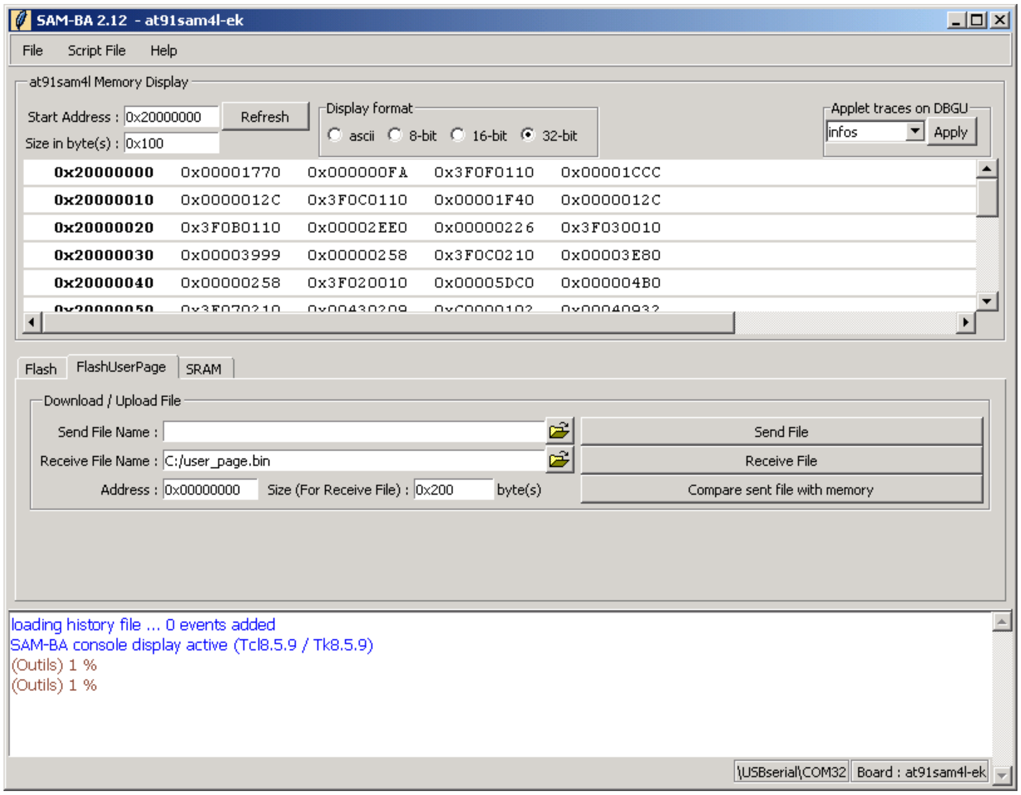
a) Start the monitor and connect to device using SAM-BA

b) Read and save the user page to a file: in the “FlashUserPage” tab, set size to 0x200, set the file path and use “Receive File”.

c) Modify the config word value in the file (a hex editor should be used). Remember that the CPU is little Endian: LSB is stored at the lowest address.

d) Write the modified file to the user page: set the file path, set size to 0x200, and use “Send File”.

Figure 2-2. Reading flash user page.



Start Address : 0x20000000

Size in byte(s) : 0x100

Display format: ○ ascii ○ 8-bit ○ 16-bit ● 32-bit

Applet traces on DBGU: infos

```
0x20000000   0x00001770   0x000000FA   0x3F0F0110   0x00001CCC
0x20000010   0x0000012C   0x3F0C0110   0x00001F40   0x0000012C
0x20000020   0x3F0B0110   0x00002EE0   0x00000226   0x3F030010
0x20000030   0x00003999   0x00000258   0x3F0C0210   0x00003E80
0x20000040   0x00000258   0x3F020010   0x00005DC0   0x000004B0
0x20000050   0x3F070210   0x00430209   0xC0000102   0x00040932
```

Send File Name :

Receive File Name : C:/user_page.bin

Address : 0x00000000 Size (For Receive File) : 0x200 byte(s)

```
loading history file ... 0 events added
SAM-BA console display active (Tcl8.5.9 / Tk8.5.9)
(Outils) 1 %
(Outils) 1 %
```

\USBserial\COM32 Board : at91sam4l-ek

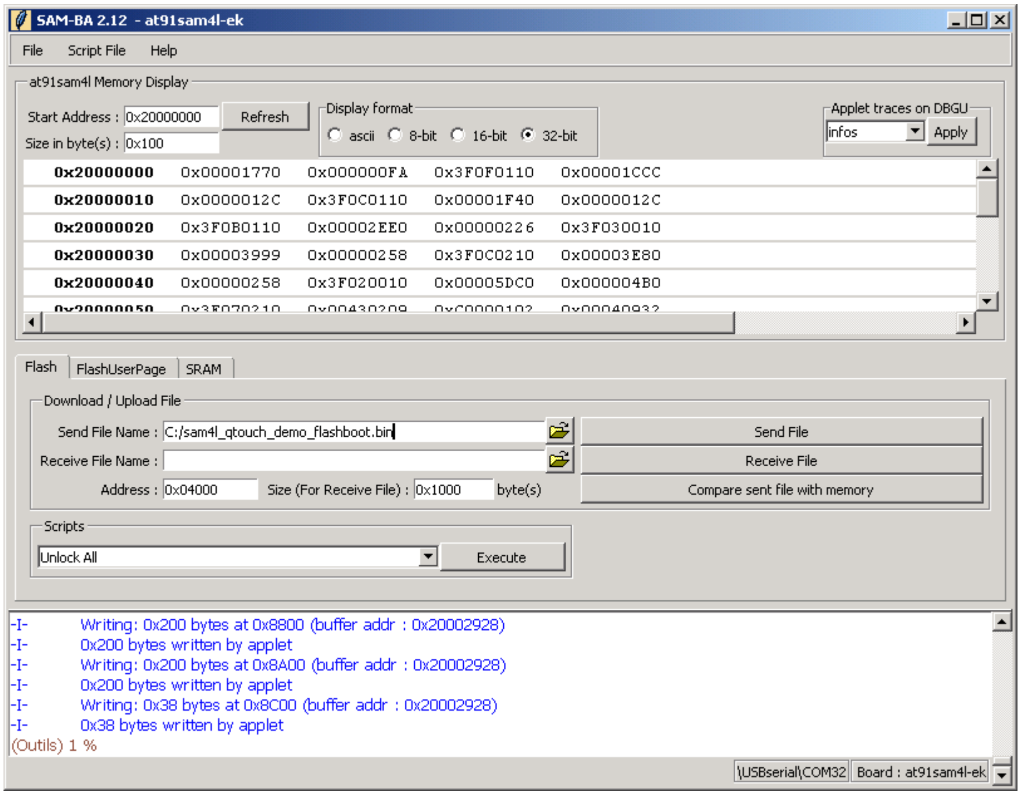
2.4 Atmel SAM-BA commands

2.4.1 Flash loading

Loading the flash content is done by the using the flash tab. Refer to the common SAM-BA user guide for more details.

When uploading a program to flash memory, the start address needs to be above 0x4000, otherwise the transfer will abort.

Figure 2-3. Flash programming.



Start Address : 0x20000000

Size in byte(s) : 0x100

○ ascii ○ 8-bit ○ 16-bit ● 32-bit

Applet traces on DBGU: infos

```
0x20000000   0x00001770   0x000000FA   0x3F0F0110   0x00001CCC
0x20000010   0x0000012C   0x3F0C0110   0x00001F40   0x0000012C
0x20000020   0x3F0B0110   0x00002EE0   0x00000226   0x3F030010
0x20000030   0x00003999   0x00000258   0x3F0C0210   0x00003E80
0x20000040   0x00000258   0x3F020010   0x00005DC0   0x000004B0
0x20000050   0x3F070210   0x00430209   0xC0000102   0x00040932
```

Send File Name : C:/sam4l_qtouch_demo_flashboot.bin

Receive File Name :

Address : 0x04000 Size (For Receive File) : 0x1000 byte(s)

Scripts: Unlock All

```
-I-     Writing: 0x200 bytes at 0x8800 (buffer addr : 0x20002928)
-I-     0x200 bytes written by applet
-I-     Writing: 0x200 bytes at 0x8A00 (buffer addr : 0x20002928)
-I-     0x200 bytes written by applet
-I-     Writing: 0x38 bytes at 0x8C00 (buffer addr : 0x20002928)
-I-     0x38 bytes written by applet
(Outils) 1 %
```

\USBserial\COM32 Board : at91sam4l-ek

2.4.1.2 Access to first flash region of SAM4Lx8

Every SAM4L devices are shipped with flash region 0 locked, on SAM4Lx8 (512kB) devices, this region also includes the start of the applicative area (0x4000 – 0x8000).

When a write or erase of this particular area is requested, SAM-BA application will temporarily unlock the region 0, access the area and finally relock the region.

2.4.2 Scripts

Atmel SAM-BA application comes with the following pre-defined scripts:

Script name	Description
Enable Security Bit	Set the security bit to secure the device (refer to the device datasheet in the FLASHCALW chapter for more information)
Read Security Bit	Give the current security state
Erase application area	Erase all application code (SAM-BA part won’t be erased)
Invalidate application	Erase first page of application
Read Fuses	Returns the values of the flash fuses
Read Locks	Returns the current lock bits value
Read Unique Serial Number	Returns the unique serial number of the Atmel ATSAM4L device (refer to the device datasheet in the FLASHCALW chapter for more information)
Set Lock Bit [0..15]	Set the specified lock bit to prevent any erasure of a flash memory region. (refer to the device datasheet in the FLASHCALW chapter for more information)
Unlock All	Unlock every flash memory sections

3. Suggested reading

3.1 Device datasheet

The device datasheet contains block diagrams of the peripherals and details about implementing firmware for the device. It also contains the electrical specifications and expected characteristics of the device.

The datasheet is available on http://www.atmel.com/ in the Datasheets section of the Atmel ATSAM4L product page.

3.2 Atmel SAM-BA user guide

The SAM-BA user guide contains detailed reference information on How to use and SAM-BA.

This user guide comes with the SAM-BA package found on http://www.atmel.com/.

3.3 USB specification

The Universal Serial Bus specification is available from http://www.usb.org.

3.4 ARM Documentation on Cortex-M4 core

- Cortex™-M4 Devices Generic User Guide for revision r0p1
- Cortex-M4 Technical Reference Manual for revision r0p1

These documents are available at http://www.arm.com/ in the info center section.

4. Revision history

Doc. Rev.	Date	Comments
42051A	01/2013	Initial document release
42051B	05/2013	Update for SAM4Lx8 devices and changed the number of the application note to from AVR®10005 to AT03454

Atmel® | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Building
1-6-4 Osaki
Shinagawa-ku, Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370

Atmel®, Atmel logo and combinations thereof, AVR®, Enabling Unlimited Possibilities®, SAM-BA®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. ARM®, Cortex® are registered trademarks or trademarks of ARM Ltd. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.