SMART ARM-based Microcontroller

AT10840: SAM-BA Bootloader for SAM L Series Devices

APPLICATION NOTE

Introduction

Atmel® SAM Boot Assistant (Atmel SAM-BA)® allows In-system Programming (ISP) using USB or UART host without any external programming interface. In general, Atmel SAM-BA monitor is factory programmed in the available ROM memory, in SAM3 and SAM4 series devices. Since there is no ROM memory in SAM L series devices such as SAM L21 and SAM L22, the SAM-BA monitor is not factory programmed. To support SAM Boot Assistant in SAM L series devices, SAM-BA bootloader can be loaded in Flash memory.

This application note complements the SAM-BA user guide and explains usage of the SAM-BA on SAM L series devices.

Features

- Allows to program, verify, and secure an Atmel SAM L21/SAM L22 device without debugger
- USB and UART connection
- Allows the end user to update application firmware from bootloader
- Configurable I/O start condition
- Customizable Source code available
- Compatible with SAM-BA v2.16 or later

Table of Contents

1. Requirements

1.1. Hardware Requirements

The Atmel SAM L21/L22 SAM Boot Assistant (SAM-BA) supports serial communication through UART or USB device port.

Table 1-1. UART Mode Requirements - ATSAM L22N18A

Signal Name	Recommended Pin Connection	Description
PC25	Connect to host (115200, 8, N, 1)	SERCOM4 PAD3 (UART RXD)
PC24	Connect to host (115200, 8, N, 1)	SERCOM4 PAD2 (UART TXD)

Table 1-2. USB Mode Requirements - ATSAML22N18A

Signal Name	Recommended Pin Connection	Description
PA25	Connect to host	USB D+ pin
PA24	Connect to host	USB D- pin

Table 1-3. Hardware Bootloader Entry - ATSAML22N18A

Signal Name	Recommended Pin Connection	Description
PC01	Connect to host or “Bootloader access switch”	The bootloader will check PC01 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.

Table 1-4. UART Mode Requirements - ATSAML21J18B

Signal Name	Recommended Pin Connection	Description
PA23	Connect to host (115200, 8, N, 1)	SERCOM3 PAD1 (UART RXD)
PA22	Connect to host (115200, 8, N, 1)	SERCOM3 PAD0 (UART TXD)

Table 1-5. USB Mode Requirements - ATSAML21J18B

Signal Name	Recommended Pin Connection	Description
PA25	Connect to host	USB D+ pin
PA24	Connect to host	USB D- pin

Table 1-6. Hardware Bootloader Entry - ATSAML21J18B

Signal Name	Recommended Pin Connection	Description
PA02	Connect to host or “Bootloader access switch”	The bootloader will check PA02 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.

1.2. Software Requirements

1.2.1. Application Constraints

The SAM-BA bootloader in SAM L21/L22 is stored in flash memory at 0x00000000 and started on reset. For SAM-BA with only UART interface enabled, bootloader is stored in flash memory from 0x0000 – 0x2000 (8KB). For SAM-BA with both USB and UART interfaces enabled, bootloader is stored in flash region 0x0000 – 0x6000 (24KB).

SAM-BA bootloader is not factory programmed on SAM L series devices and has to be programmed using an external programmer. 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 external programmer.

To use SAM-BA together with an application, the user must link the application starting at 0x2000 when UART interface is alone enabled and at 0x6000 when both USB and UART interfaces are enabled.

The procedure to modify the start address in an IAR project and an Atmel Studio project is explained in Atmel AT04189: UART Based SAM-BA Bootloader for SAM D20.

Figure 1-1. Memory Map of ATSAML21J18B / ATSAML22N18A with an Application and SAM-BA with both USB and UART

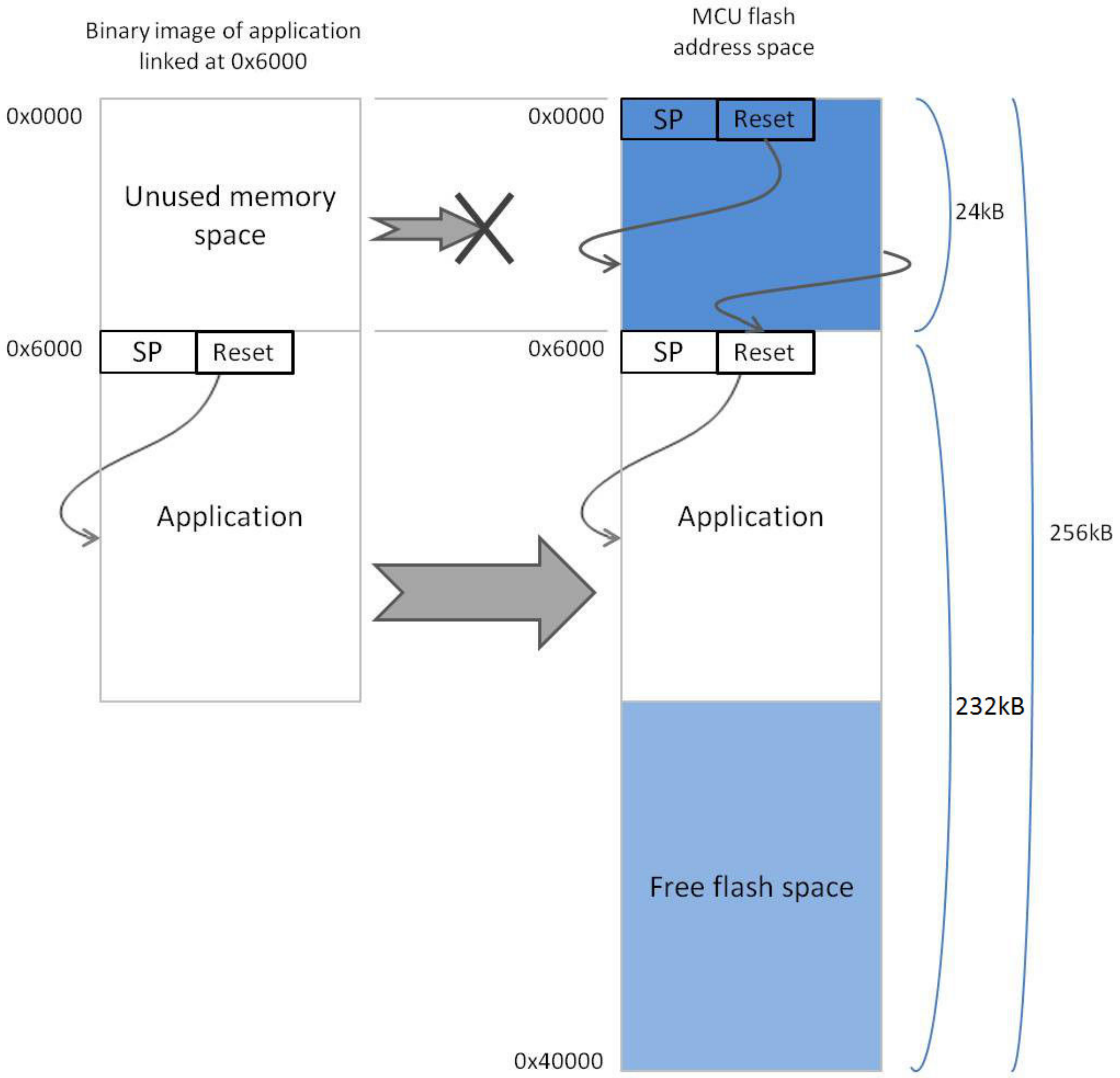
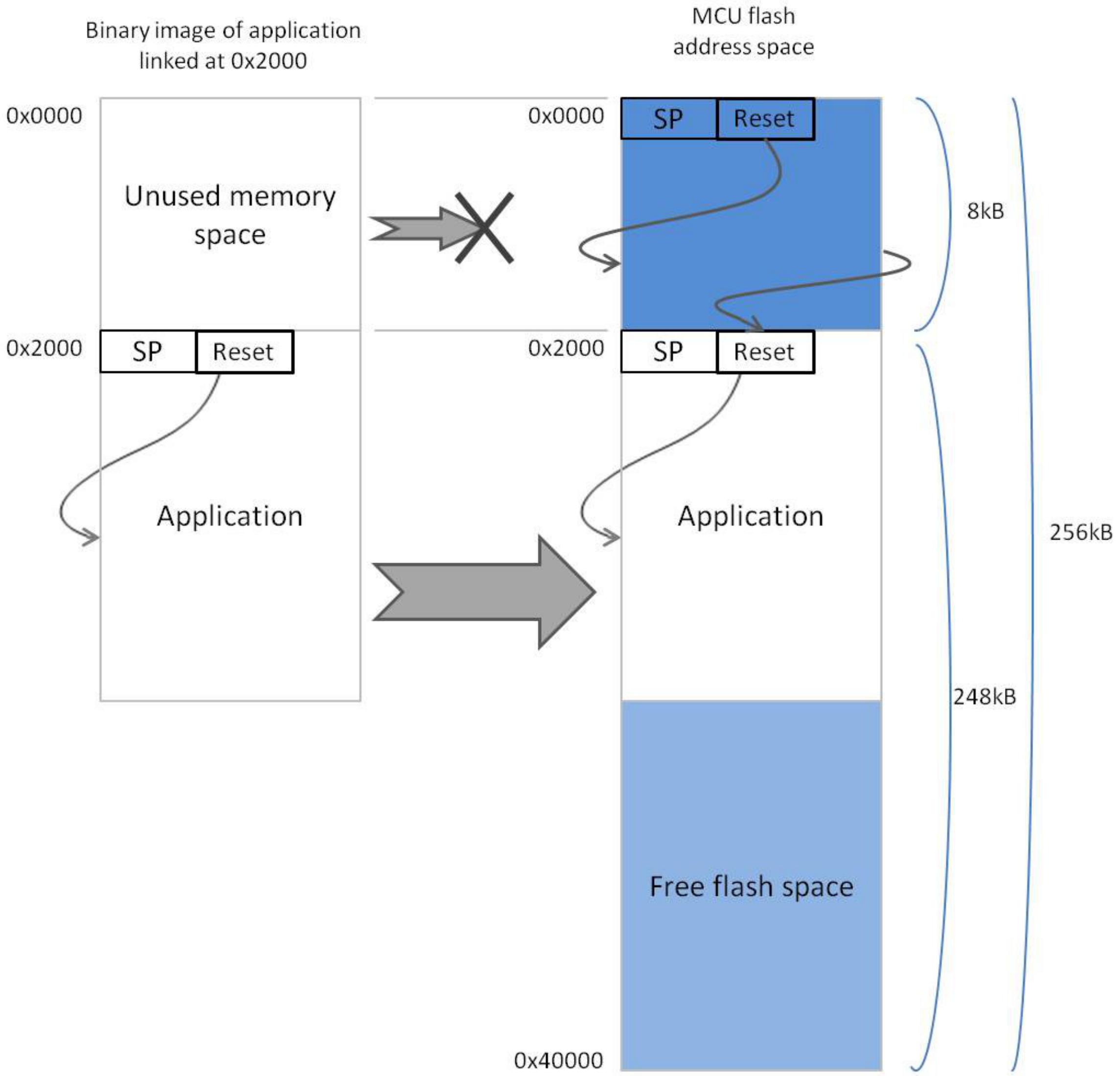


Figure 1-2. Memory Map of ATSAML21J18B / ATSAML22N18A with an Application and SAM-BA with UART



When loading the application's binary image to the device, the part of flash after the SAM-BA must be used for programmed.

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

2. Using the Bootloader

2.1. Programming the Bootloader

Available Bootloader Projects for SAM L Series devices

SAM L21:

Launch Atmel Studio -> **File** -> **New** -> **Example Project...** -> SAM0 SAM-BA Bootloader Example - SAM L21 Xplained Pro (ATSAML21J18B)

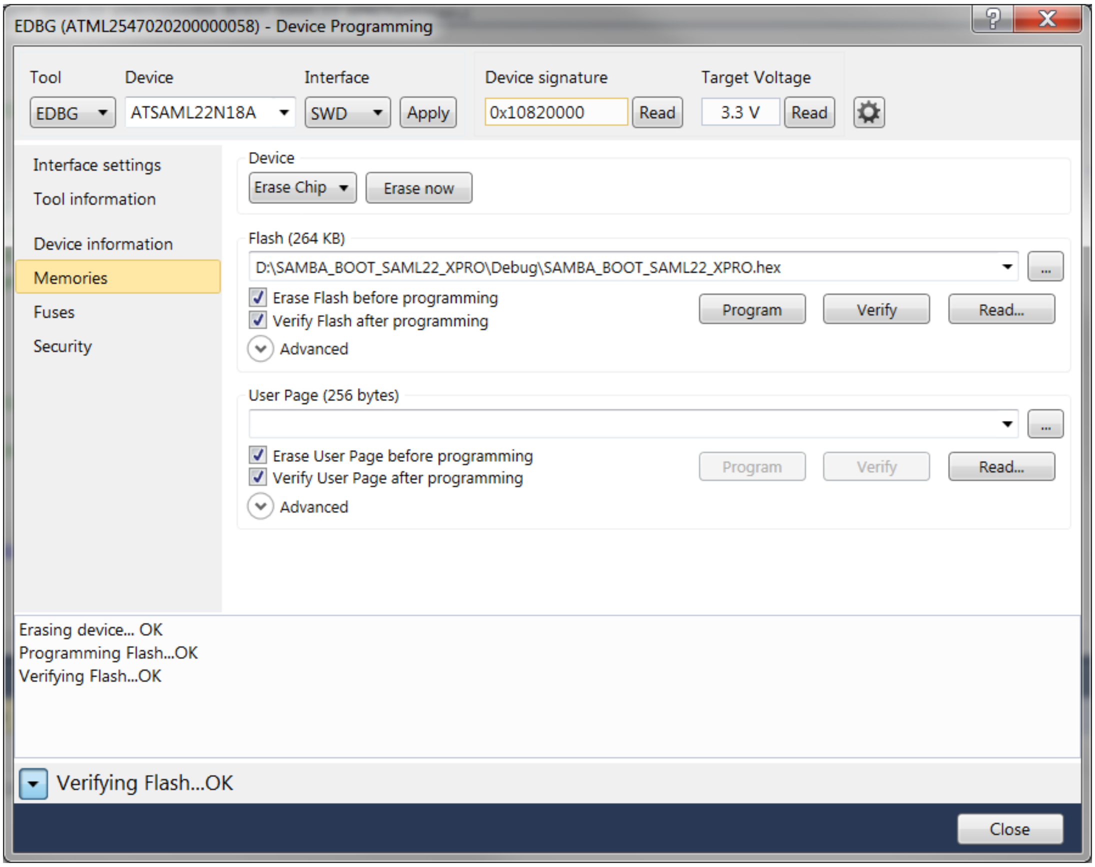
SAM L22:

Launch Atmel Studio -> **File** -> **New** -> **Example Project...** -> SAM0 SAM-BA Bootloader Example - SAM L22 Xplained Pro

Build the Example Project and program the Atmel SAM Boot Assistant using SWD debugger:

1. In Atmel Studio application, open **Tools** > **Device Programming**, select the **Tool**, and click **Apply**.
2. From the **Memories** tab, erase the chip by clicking **Erase now**.
3. Specify the path for SAM-BA image in the **Flash** box and **Program**.

Figure 2-1. Device Programming - SAML22 device



EDBG (ATML2547020200000058) - Device Programming

Tool: EDBG
Device: ATSAML22N18A
Interface: SWD
Device signature: 0x10820000
Target Voltage: 3.3 V

Device: Erase Chip

Flash (264 KB): D:\SAMBA_BOOT_SAML22_XPRO\Debug\SAMBA_BOOT_SAML22_XPRO.hex
☑ Erase Flash before programming
☑ Verify Flash after programming

User Page (256 bytes)
☑ Erase User Page before programming
☑ Verify User Page after programming

Erasing device... OK
Programming Flash...OK
Verifying Flash...OK

Verifying Flash...OK

Alternately, command-line programming tool *atprogram* can also be used for programming the hex file.

Example Command: *atprogram –t edbg –i swd –d atsaml22n18a program –f SAMBA_BOOT_SAML22_XPRO.hex*

Note: The same steps must be followed to program SAM-BA Bootloader for SAML21 series devices

2.2. Entering the Bootloader

The SAM L21/ L22 SAM-BA relies on a SAM-BA bootloader which is stored in flash memory. This monitor will be executed according conditions specified in the boot process.

SAM-BA bootloader activation can be requested using one of the following ways:

- **External condition:** Reset the part and make sure the Hardware Bootloader Entry pin (PC01 for SAM L22 & PA02 for SAM L21) is pulled low, when reset is released. A common usage is to use a push button accessible by the user as a bootloader trigger. The user must simply hold the push button when powering up the device.
- **Internal condition:** On erased devices or when the application reset vector (@Application start address + 4) is blank (0xFFFFFFFF).

2.3. Selecting the Communication Interface

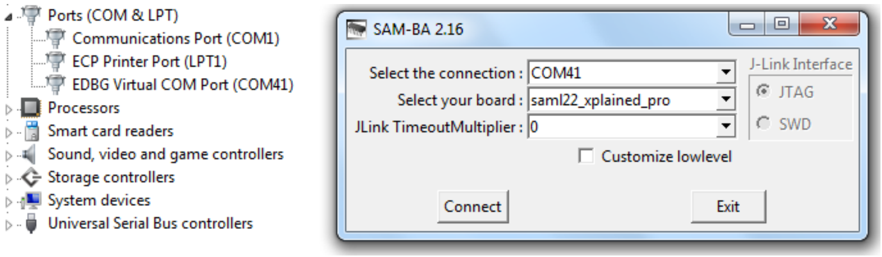
The SAM-BA bootloader can be compiled by commenting #define `CONF_USB_CDC_INTERFACE_SUPPORT` in `conf_board.h` file in the Atmel Studio project. By default, project supports both interfaces (USB, UART)

2.4. Using the Bootloader with SAM-BA GUI

2.4.1. Connecting from SAM-BA PC Application

For using SAM-BA monitor with UART interface, connect SAM L22 Xplained Pro to the PC via DEBUG USB port using the micro-USB cable as the UART pins are connected to it.

Figure 2-2. Connecting from SAM-BA PC Application using UART



- Ports (COM & LPT)
  - Communications Port (COM1)
  - ECP Printer Port (LPT1)
  - EDBG Virtual COM Port (COM41)
- Processors
- Smart card readers
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

SAM-BA 2.16

Select the connection : COM41

Select your board : saml22_xplained_pro

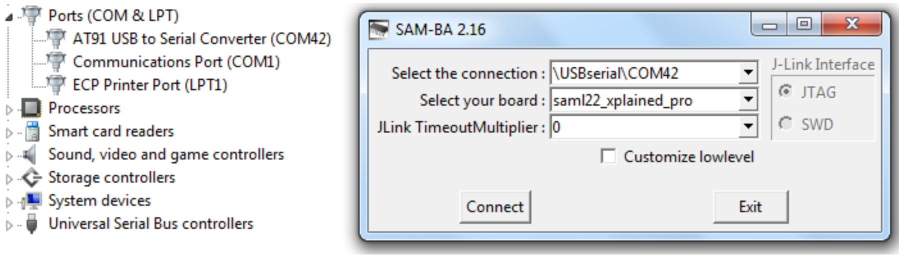
JLink TimeoutMultiplier : 0

□ Customize lowlevel

J-Link Interface: ● JTAG ○ SWD

For using SAM-BA monitor with USB interface, connect SAM L22 Xplained Pro to the PC via TARGET USB port using the micro-USB cable.

Figure 2-3. Connecting from SAM-BA PC Application using USB CDC



- Ports (COM & LPT)
  - AT91 USB to Serial Converter (COM42)
  - Communications Port (COM1)
  - ECP Printer Port (LPT1)
- Processors
- Smart card readers
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

SAM-BA 2.16

Select the connection : \USBserial\COM42

Select your board : saml22_xplained_pro

JLink TimeoutMultiplier : 0

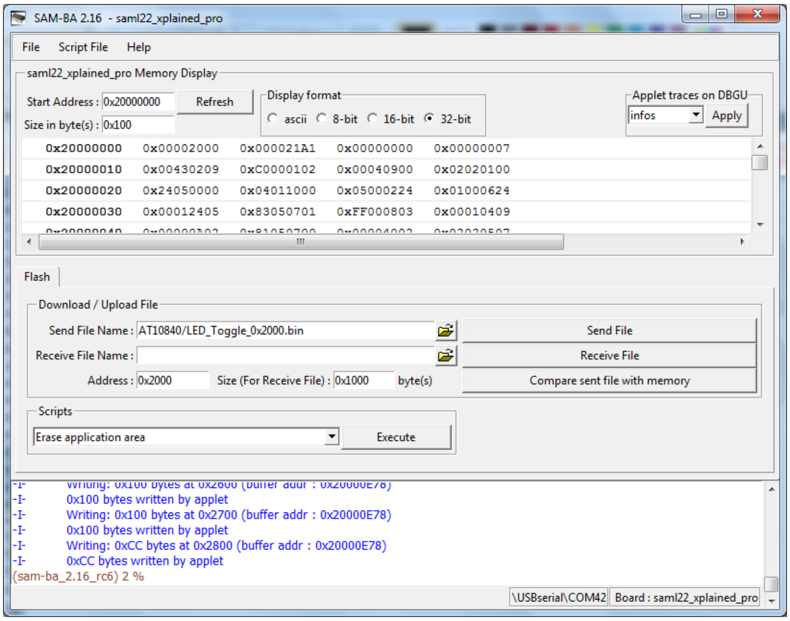
□ Customize lowlevel

J-Link Interface: ● JTAG ○ SWD

2.4.2. Flash Loading

The contents of the Flash are loaded using the **Flash** tab. While uploading a program to flash memory, the start address must be 0x2000 when using UART and 0x6000 when using both USB and UART, otherwise the transfer process will be aborted.

Figure 2-4. Flash Programming



SAM-BA 2.16 - saml22_xplained_pro
File
Script File
Help
saml22_xplained_pro Memory Display
Refresh
Display format
○ ascii ○ 8-bit ○ 16-bit ● 32-bit
Applet traces on DBGU
Apply
Flash
Download / Upload File
Send File
Receive File
Compare sent file with memory
Scripts
Execute

Start Address : 0x20000000
Size in byte(s) : 0x100
Applet traces on DBGU: infos

```
0x20000000   0x00002000   0x000021A1   0x00000000   0x00000007
0x20000010   0x00430209   0xC0000102   0x00040900   0x02020100
0x20000020   0x24050000   0x04011000   0x05000224   0x01000624
0x20000030   0x00012405   0x83050701   0xFF000803   0x00010409
```

Send File Name : AT10840/LED_Toggle_0x2000.bin
Receive File Name :
Address : 0x2000   Size (For Receive File) : 0x1000 byte(s)
Scripts: Erase application area

```
-I-     Writing: 0x100 bytes at 0x2600 (buffer addr : 0x20000E78)
-I-     0x100 bytes written by applet
-I-     Writing: 0x100 bytes at 0x2700 (buffer addr : 0x20000E78)
-I-     0x100 bytes written by applet
-I-     Writing: 0xCC bytes at 0x2800 (buffer addr : 0x20000E78)
-I-     0xCC bytes written by applet
(sam-ba_2.16_rc6) 2 %
```

\USBserial\COM42   Board : saml22_xplained_pro

Note: The same steps can be followed for SAM L21 series devices.

2.4.3. Scripts

The following predefined scripts are available with the Atmel SAM-BA application.

Table 2-1. Predefined Scripts

Script Name	Description
Set Security Bit	Set the security bit to secure the device. Refer to NVMCTRL chapter in the device datasheet for more information.
Read Security Bit	Read the current security state
Erase application area	Erase all application code (SAM-BA monitor region will not be erased)
Invalidate application	Erase first page of application
Read Fuses	Returns the values of fuse settings. Refer to NVM user row mapping section in the device datasheet for more information.
Read Lock Fuses	Read the current lock settings
Read DeviceID	Read the Device Identification register
Set Lock Bit [0:15]	Set the specified lock bit to prevent any erasure of flash memory region. Refer to the NVMCTRL chapter in the device datasheet for more information.
Unlock all	Unlock all flash memory regions

3. Re-building SAM-BA Monitor

The following pins are used by the SAM-BA bootloader for ATSAML22N18A (100 pin count) device.

Table 3-1. Pins Used in ATSAML22N18A

Pins Used	Description
PA25	USB D+
PA24	USB D-
PC25	UART RxD (SERCOM4 PAD3)
PC24	UART TxD (SERCOM4 PAD2)
PC01	General Purpose Input (Hardware Bootloader Entry pin)

The USB pins (PA24, PA25) are same in all the SAM L22 device variants.

While building the project for other SAM L22 devices, the general purpose input pin, SERCOM pins (UART RxD, UART TxD) must be changed based on the respective SAM L22 device's pinout.

To change the pin mapping, edit the `conf_bootloader.h` file in the SAM-BA bootloader project.

For other SAM L22 device variants, the project has to be recompiled by changing the device.

The following pins are used by the SAM-BA bootloader for ATSAML21J18B (64 pin count) device.

Table 3-2. Pins Used in ATSAML21J18B

Pins Used	Description
PA25	USB D+
PA24	USB D-
PA23	UART RxD (SERCOM3 PAD1)
PA22	UART TxD (SERCOM3 PAD0)
PA02	General Purpose Input (Hardware Bootloader Entry pin)

While building the project for other SAM L21 devices, the general purpose input pin, SERCOM pins (UART RxD, UART TxD) must be changed based on the respective SAM L21 device's pinout.

To change the pin mapping, edit the `conf_bootloader.h` file in the SAM-BA bootloader project.

For other SAM L21 device variants, the project has to be recompiled by changing the device.

Note:
1. For SAML21 series devices: SAM-BA bootloader is supported only from Rev B or later version of devices.
2. For SAM L21 series devices: There is no USB interface supported in ASF3.30. This support will be added in the subsequent ASF release.

4. References

1. Device Datasheet: http://www.atmel.com/Images/Atmel-42402-SAM-L22_Datasheet.pdf.
2. Device Datasheet: http://www.atmel.com/Images/Atmel-42385-SAM-L21_Datasheet.pdf.
3. Atmel SAM-BA User Guide in SAM-BA installation folder in the PC.
4. ARM® documentation on Cortex®-M0+ Core:
   - Cortex-M0+ Devices Generic User Guide revision r0p1
   - Cortex-M0+ Technical Reference Manual revision r0p1
5. Atmel SAM-BA In-system Programmer: http://www.atmel.com/tools/atmelsam-bain-systemprogrammer.aspx.
6. Atmel Studio: http://www.atmel.com/tools/ATMELSTUDIO.aspx.

5. Revision History

Doc. Rev.	Date	Comments
42650A	04/2016	Initial document release.

Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

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.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death ("Safety-Critical Applications") without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.