
AT04189: UART Based SAM-BA Bootloader for SAM D20

Atmel SAM D20

Description

Atmel® SAM Boot Assistant (Atmel SAM-BA®) allows In-System Programming (ISP) from UART host without any external programming interface. Unlike existing SAM products, ROM monitor is not available in Atmel SAM D20 and SAM-BA will be loaded in Flash memory.

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 SAM D20 design.

Features

- Allows to program, verify, and secure a SAM D20 device without debugger
- UART connection (RS232)
- Allows the end user to update firmware in application
- Configurable I/O start condition
- Source code available, can be customized to user's needs
- Compatible with SAM-BA v2.12 with patch 7 or above

1 Requirements

1.1 Hardware Requirements

The Atmel SAM D20 SAM-BA Boot Assistant supports serial communication via the UART interface.

Table 1-1. UART Mode Requirements

Signal name	Recommended pin connection	Description
PA25	Connect to host: (115200 8,N,1)	SERCOM3 PAD3 (UART RXD)
PA24	Connect to host: (115200 8,N,1)	SERCOM3 PAD2 (UART TXD)

Table 1-2. Hardware Bootloader Entry

Signal name	Recommended pin connection	Description
PA15	Connect to host or "Bootloader access switch"	The bootloader will check PA15 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

Unlike existing SAM products using a ROM monitor, on the Atmel SAM D20 SAM-BA is stored in flash memory at [0x0 – 0x800] 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 SWD debugger.

To use SAM-BA together with an application, the user needs to link the application starting at 0x800. The below procedure explains how to modify the start address in an IAR™ project and an Atmel Studio project.

Figure 1-1. Changing .intvec start in IAR

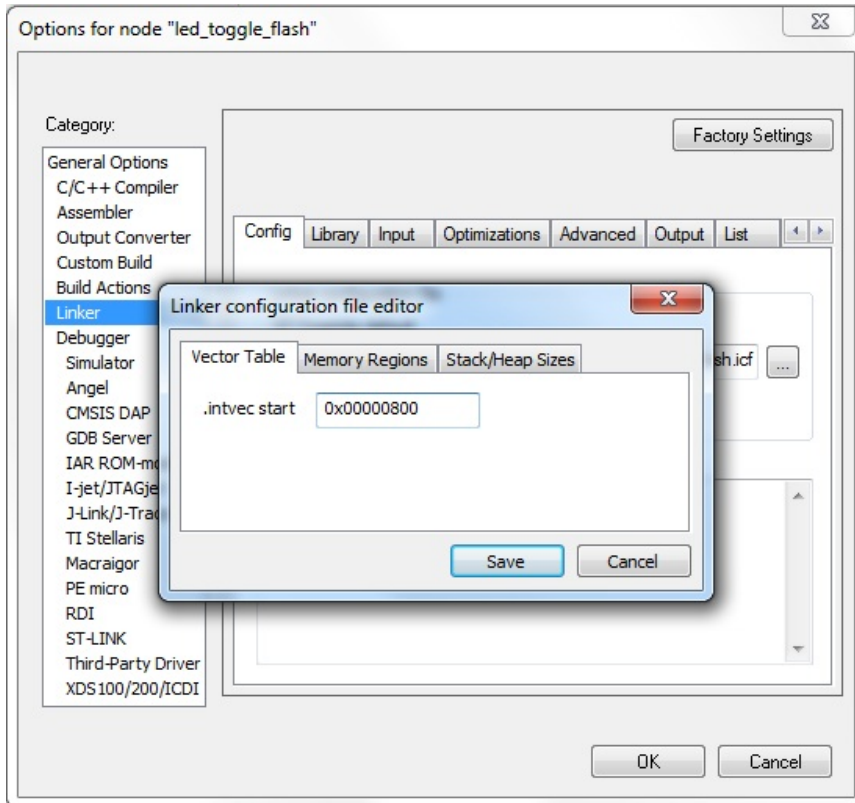
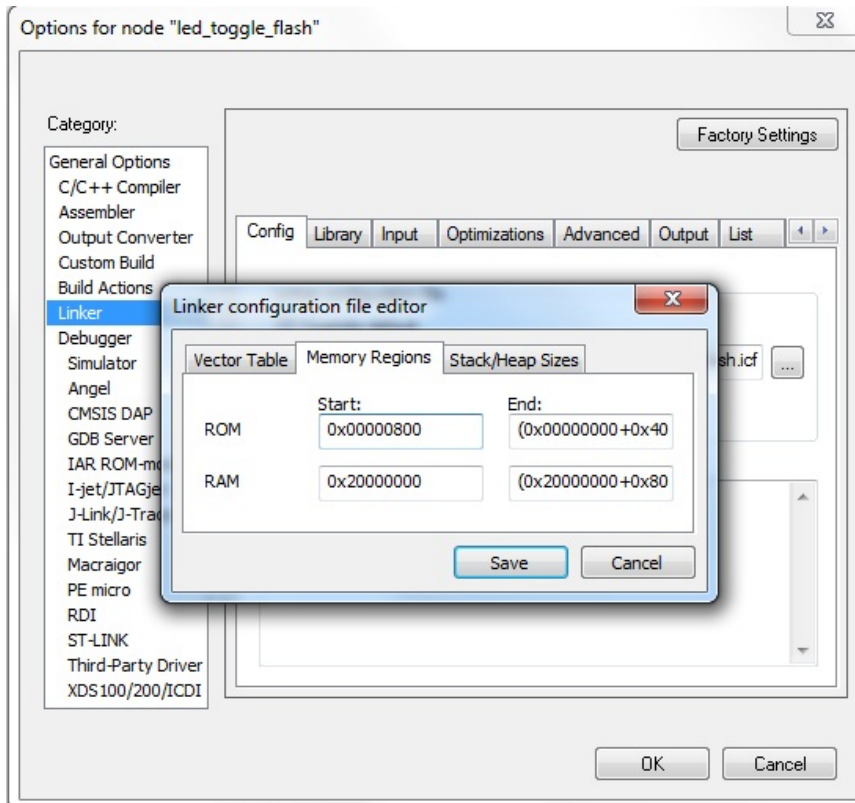


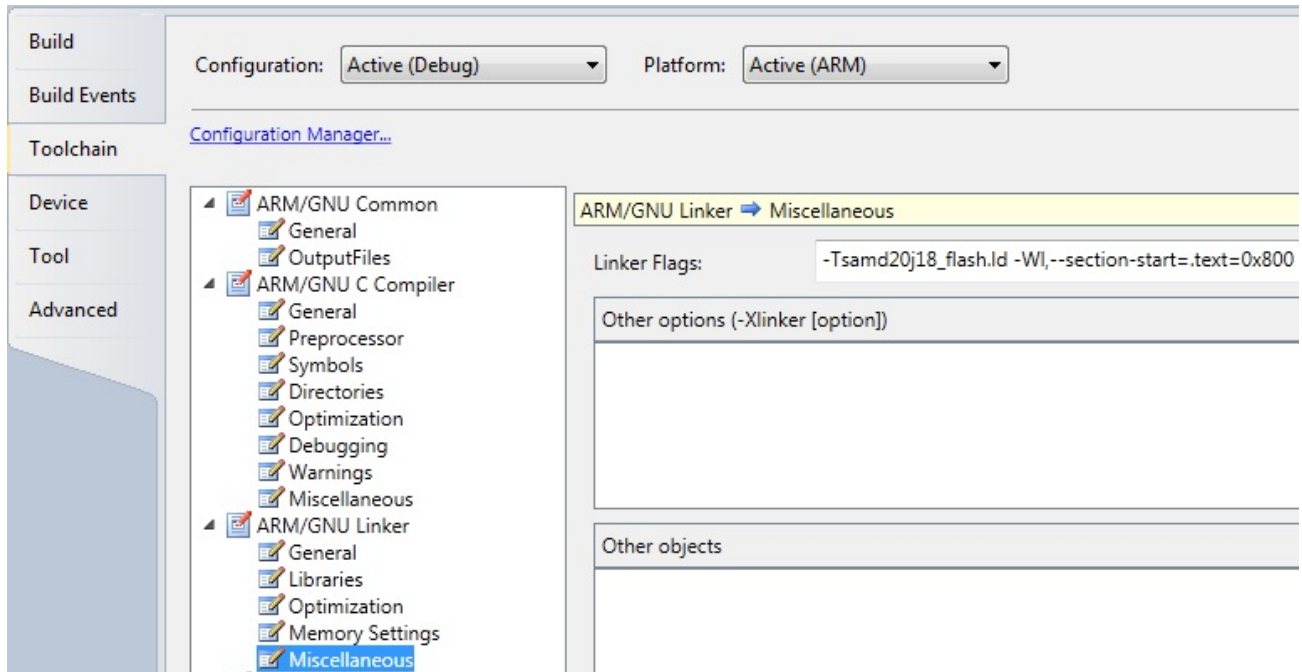
Figure 1-2. Changing ROM Start and Length in IAR



To change the linker script under IAR project:

- Menu Project, Options...
- Go to linker tab -> Config
- Select 'Edit'
- Set .intvect start in 'Vector Table' tab to 0x00000800 as in [Figure 1-1](#)
- Set ROM start in 'Memory Regions' tab to 0x00000800 and end to original ROM length minus 0x800. [Figure 1-2](#) shows modified example for ATSAM20J18
- Rebuild the project

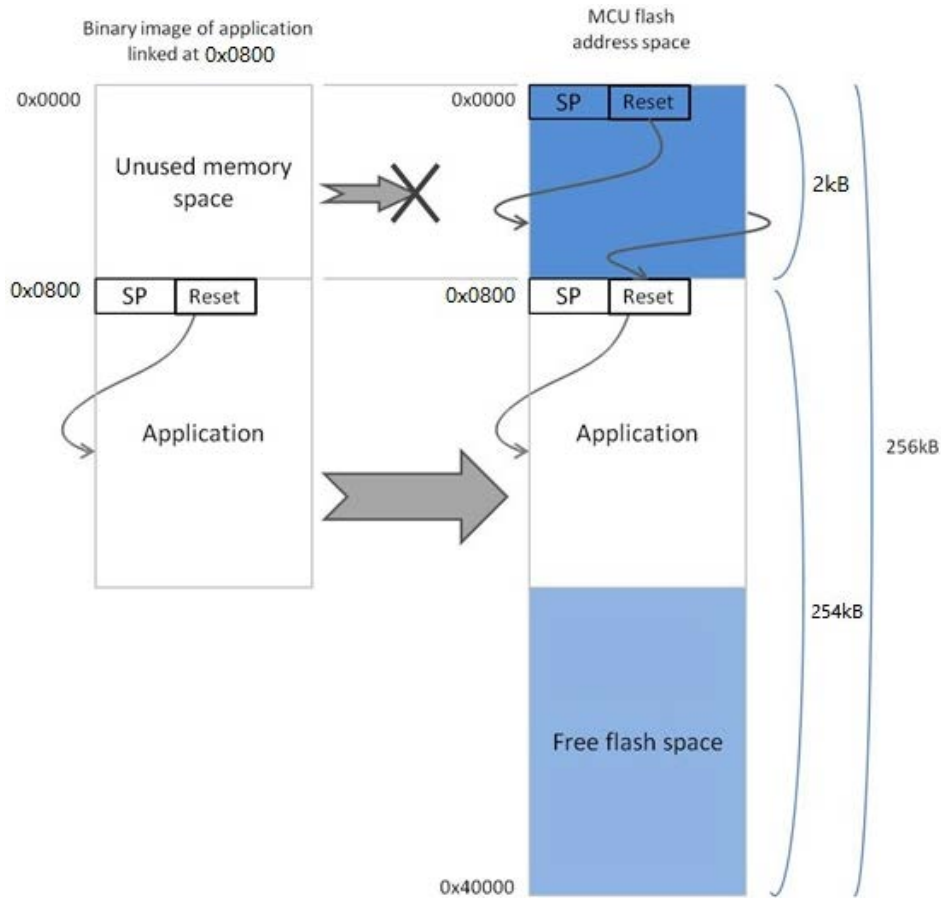
Figure 1-3. Changing Start Address in Atmel Studio



To change the linker script under Atmel Studio:

- Menu Project, Properties...
- Toolchain Tab, ATM/GNU Linker, Miscellaneous
- Include -Wl,--section-start=.text=Desired start address
(For e.g., -Wl,--section-start=.text=0x800)

Figure 1-4. Memory Map of ATSAM20J18 Device with an Application and SAM-BA



When loading the application binary image to the device, only the part of flash starting from 0x800 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.

2 Using the Bootloader

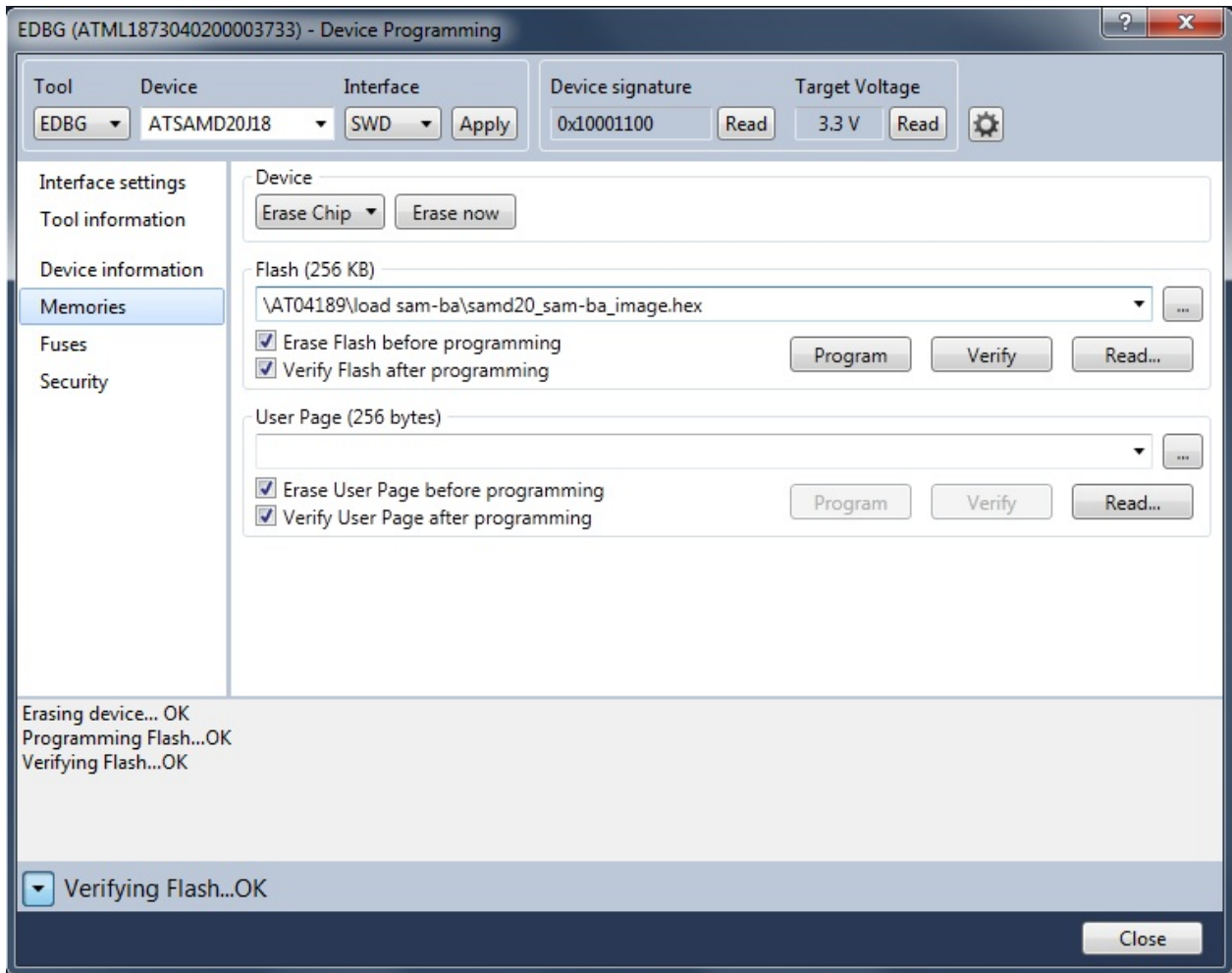
2.1 Programming the Bootloader

Programming the Atmel SAM Boot Assistant can be done using SWD debugger:

- In Atmel Studio, open “Tools\Device Programming”, select debugger and Apply.
- From the memories tab, launch a chip erase.
- Fill the path to “samd20_sam-ba_image.hex” in the Flash box and program.

Note: The `samd20_sam-ba_image.hex` file in folder 'load sam-ba' in AT04189.zip is built for ATSAM20J18 device. For other devices the SAM-BA monitor has to be re-built. Refer Chapter 4 for more details.

Figure 2-1. Device Programming



Alternatively command-line programming tool 'atprogram' can also be used for programming the 'hex' file.

Example: `atprogram -t edbg -i swd -d atsamd20j18 program -f samd20_sam-ba_image.hex.`

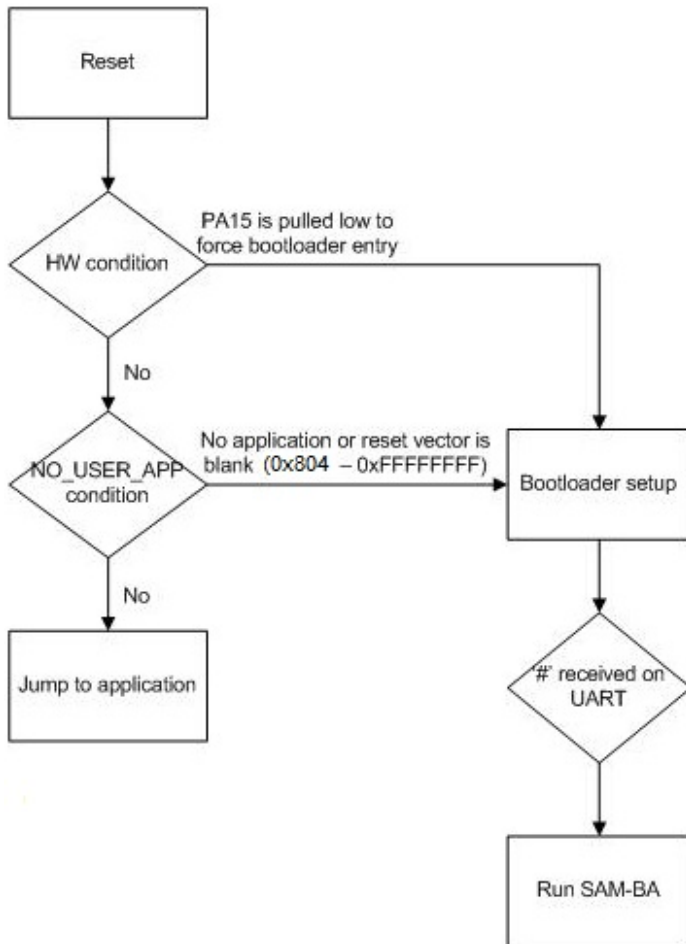
2.2 Entering the Bootloader

Like other Atmel SAM devices, the SAM D20 SAM-BA relies on a monitor; this monitor is entered according to the boot process conditions described in [Figure 2-2](#).

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

- External condition: Reset the part and make sure the Hardware Bootloader Entry pin (PA15) 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 simply has to hold the push button when powering up the device.
- Internal condition: On erased devices or when the application reset vector (@0x804) is blank (0xFFFFFFFF)

Figure 2-2. Boot Process of Atmel SAM-BA



Code security concerns: When SAM-BA 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.

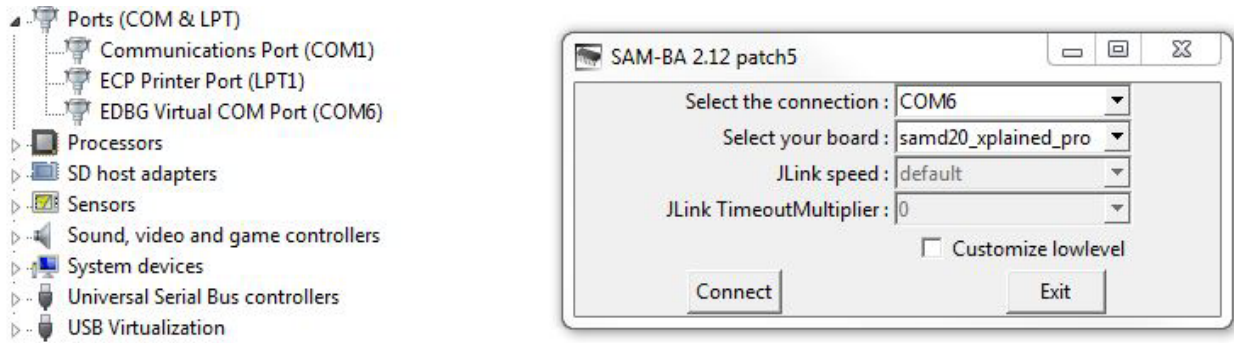
After these preliminary steps, the monitor will enter a loop and test whether a '#' (sharp) character is received on PA25 (SERCOM3 PAD3) line. PA24 (SERCOM3 PAD2) becomes an output driven by the SAM D20 device according to the MUX setting.

2.3 Using the Bootloader with SAM-BA GUI

2.3.1 Connecting from SAM-BA PC Application

Connect SAM D20 Xplained Pro to the PC to USB DEBUG port using the micro-USB cable as the UART pins are connected to it.

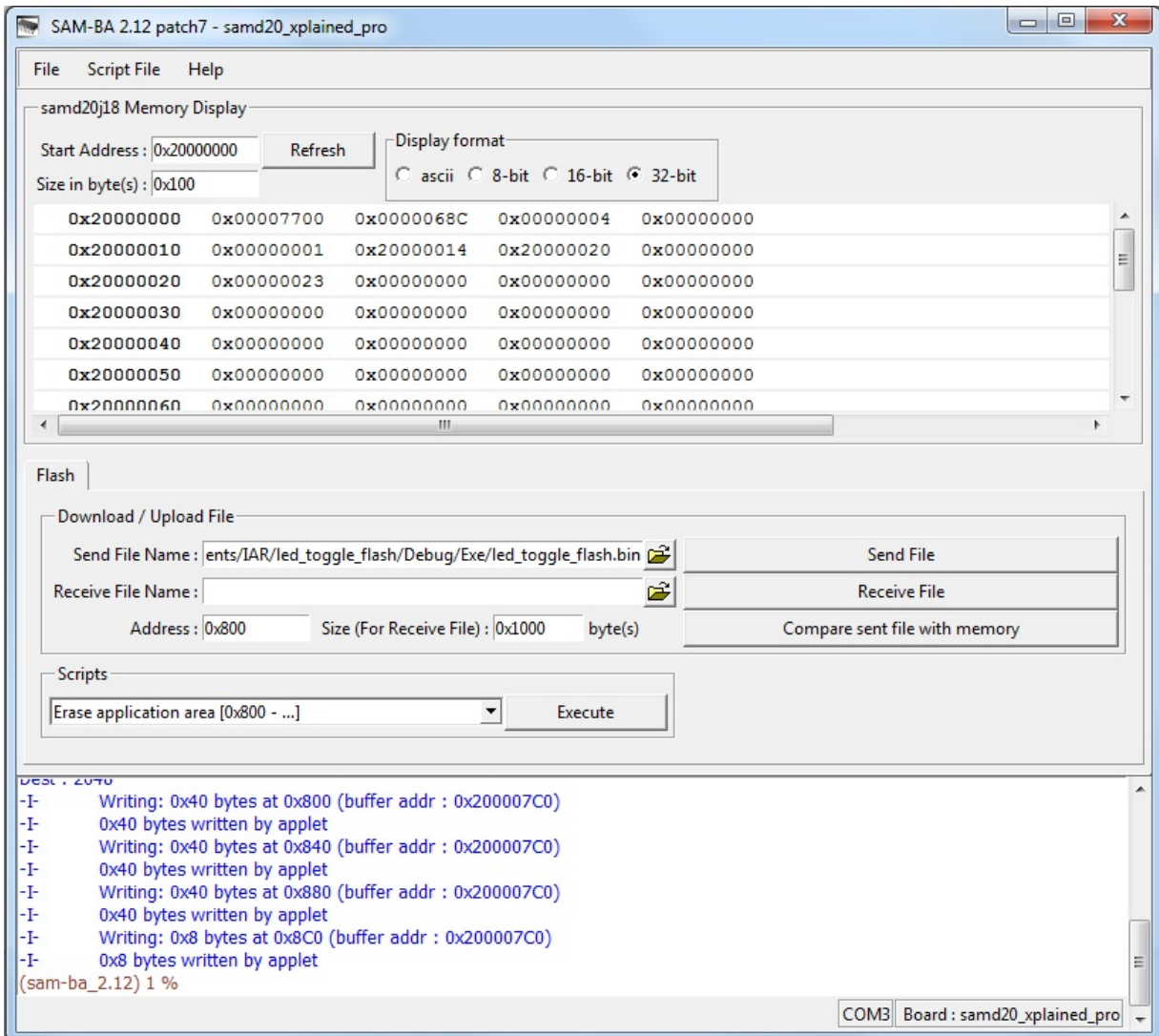
Figure 2-3. Connecting from SAM-BA PC Application



2.3.2 Flash Loading

Loading the flash content is done by using the Flash tab. When uploading a program to flash memory, the start address needs to be above 0x800, otherwise the transfer will abort.

Figure 2-4. Flash Programming



2.3.3 Scripts

Atmel SAM-BA application comes with the following predefined scripts (see [Table 2-1](#)):

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 part won't 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 NVMCTRL chapter in the device datasheet for more information)
Unlock all	Unlock all flash memory regions

3 Software Package Contents

The software package with this application note contains the following:

1. Source code of SAM D20 SAM-BA monitor and the associated IAR project.
2. Example binary image of an application which starts at 0x800 – The application continuously toggles LED0 on SAM D20 Xplained Pro.
3. SAM-BA monitor image in 'hex' format.

Note: To ensure minimal size for the bootloader, the source code as well as the project file is available only for the IAR compiler.

4 Re-building SAM-BA Monitor

The SAM-BA monitor IAR project in AT04189.zip uses ATSAM D20J18 device by default. To use the SAM-BA monitor with other SAM D20 devices, the monitor IAR project has to be re-built with following changes to be done in the project settings.

1. Open SAM-BA monitor IAR project and go to Project -> Options -> General Options -> Target tab -> Processor Variant -> select the required SAM D20 device.
2. Now go to C/C++ Compiler -> Preprocessor tab -> under 'Defined Symbols' box, edit the device name macro. For example, for ATSAM D20G18 device, change the macro to `__SAM D20G18__` (Note: Double underscores should be used).
3. Now go to Linker -> Config tab -> Linker Configuration File -> change the file name to required device. For example, for ATSAM D20G18 device, use `$PROJ_DIR$linker\samd20g18_flash.icf`.

5 Supported Devices

Following devices in SAM D20 family are supported by the SAM-BA version 2.12 patch 7 or above.

1. ATSAMD20E15
2. ATSAMD20E16
3. ATSAMD20E17
4. ATSAMD20E18
5. ATSAMD20G15
6. ATSAMD20G16
7. ATSAMD20G17
8. ATSAMD20G18
9. ATSAMD20J15
10. ATSAMD20J16
11. ATSAMD20J17
12. ATSAMD20J18

ATSAMD20E14/G14/J14 devices are not supported by SAM-BA.

6 References

6.1 Device Datasheet

The device datasheet contains the 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.

Datasheet is available on www.atmel.com in the Documents section of Atmel SAM D20 product page.

6.2 Atmel SAM-BA User Guide

The SAM-BA User Guide contains detailed reference information on how to use SAM-BA.

This user guide comes with the SAM-BA package available on www.atmel.com.

6.3 ARM Documentation on Cortex-M0+ Core

- Cortex[®]-M0+ Devices Generic User Guide revision r0p1
- Cortex-M0+ Technical Reference Manual revision r0p1

6.4 Atmel SAM-BA In-system Programmer

Latest Atmel SAM-BA In-system Programmer and related patches can be downloaded from <http://www.atmel.com/tools/ATMELSAM-BAIN-SYSTEMPROGRAMMER.aspx>.

6.5 Atmel Studio

The latest version of Atmel Studio can be downloaded from <http://www.atmel.com/tools/atmelstudio.aspx>.

7 Revision History

Doc Rev.	Date	Comments
42238B	06/2014	<ol style="list-style-type: none">1. Changed all entries of application start address from '0x2000' to '0x800'2. Modified Figures 1-1, 1-2, 1-4, 2-1, 2-2, and 2-43. Section 2-1: 'elf' has been changed to 'hex'4. Chapter 4 'Re-building SAM-BA Monitor' and Chapter 5 'Supported Devices' are added5. New template.
42238A	01/2014	Initial document release.



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

© 2014 Atmel Corporation. / Rev.:Atmel-42238B-UART-based-SAM-BA-Bootloader-for-SAM-D20_ApplicationNote_AT04189_062014.

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, SAM-BA®, and others are registered trademarks or trademarks of Atmel Corporation in U.S. and other countries. ARM®, ARM Connected® logo, Cortex®, and others are the 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.

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.