

---

**Atmel AVR1916: USB DFU Boot Loader for XMEGA**

---

**8-bit Atmel Microcontroller****Features**

---

- In-system programming
- USB DFU Atmel® protocol
- Read/write flash and EEPROM on-chip memories
- Read device ID
- Full chip erase
- Start application command

**Introduction**

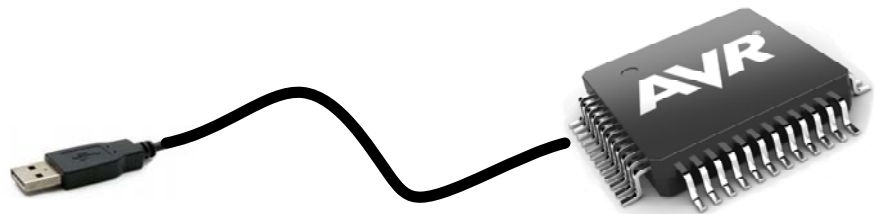
---

Atmel AVR® XMEGA® devices now make easy to implement and use USB. The XMEGA with USB interface devices can be factory configured or reprogrammed with a USB boot loader located in the on-chip flash boot section of the controller.

This USB boot loader allows performing in-system programming (ISP) from a USB host controller without removing the part from the system, and without any external programming interface other than the USB connector.

One boot loader binary is compiled and released for each AVR XMEGA device. This document applies to all XMEGA with USB module.

Atmel provides a USB host loader application called FLIP which works with several Windows® OS. The full USB DFU protocol implemented by Atmel is covered within the [Atmel AVR4023](#) application note.



## Table of Contents

1. Abbreviations .....	3
2. Related parts.....	3
3. Related items .....	3
4. Boot loader environment.....	4
5. Boot loader behavior.....	5
5.1 Part configuration .....	5
5.2 Boot process .....	5
5.3 Security .....	6
6. Using the boot loader.....	7
6.1 Activating the boot loader.....	7
6.1.1 Forcing boot loader mode with XMEGA-A3BU Xplained .....	7
6.1.2 Forcing boot loader mode with XMEGA-B1 Xplained .....	7
6.2 Connecting the boot loader to a USB host .....	8
6.2.1 Drivers installation for Windows XP .....	8
6.2.2 Drivers installation for Windows Vista or Windows 7 .....	9
6.3 Getting started with FLIP.....	10
6.4 Customizing the boot loader .....	11
7. Software information .....	12
7.1 Software revision history .....	12
7.1.1 Version 1.0.2 – 2011/09/01 .....	12
7.1.2 Version 1.0.3 – 2012/03/01 .....	12
7.1.3 Version 1.0.4 – 2012/08/01 .....	12

## 1. Abbreviations

- ISP: In-system programming
- BOD: Brown-out detector
- POR: Power-on reset
- USB: Universal serial bus
- DFU: Device firmware upgrade
- FLIP: Flexible in-system programmer

## 2. Related parts

This documentation applies to the following Atmel AVR XMEGA parts:

- [Atmel ATxmega128A1U](#)
- [Atmel ATxmega64A1U](#)
- [Atmel ATxmega256A3BU](#)
- [Atmel ATxmega256A3U](#)
- [Atmel ATxmega192A3U](#)
- [Atmel ATxmega128A3U](#)
- [Atmel ATxmega64A3U](#)
- [Atmel ATxmega32A4U](#)
- [Atmel ATxmega16A4U](#)
- [Atmel ATxmega128B1](#)
- [Atmel ATxmega64B1](#)
- [Atmel ATxmega128B3](#)
- [Atmel ATxmega64B3](#)
- [Atmel ATxmega384C3](#)
- [Atmel ATxmega256C3](#)
- [Atmel ATxmega128C3](#)
- [Atmel ATxmega64C3](#)
- [Atmel ATxmega32C4](#)
- [Atmel ATxmega16C4](#)

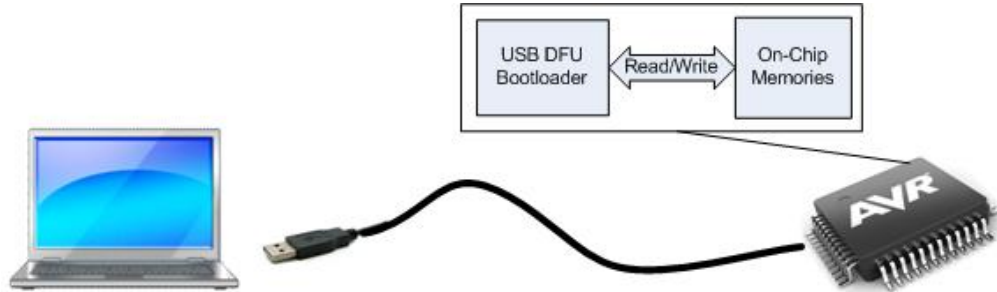
## 3. Related items

- Atmel FLIP:
  - <http://www.atmel.com/tools/FLIP.aspx>
- Atmel AVR Software Framework:
  - <http://www.atmel.com/asf>
- Atmel AVR XMEGA AU Manual
  - <http://www.atmel.com/Images/doc8331.pdf>
- Atmel AVR XMEGA B Manual
  - <http://www.atmel.com/Images/doc8291.pdf>

## 4. Boot loader environment

The boot loader is located in the boot section of the on-chip Atmel AVR XMEGA flash memory; it manages the USB communication protocol and performs read/write operations to the on-chip memories (flash/EEPROM).

Figure 4-1. Physical environment.



## 5. Boot loader behavior

### 5.1 Part configuration

When using parts with pre-programmed DFU boot loader, they are already configured and ready-to-use in DFU mode.

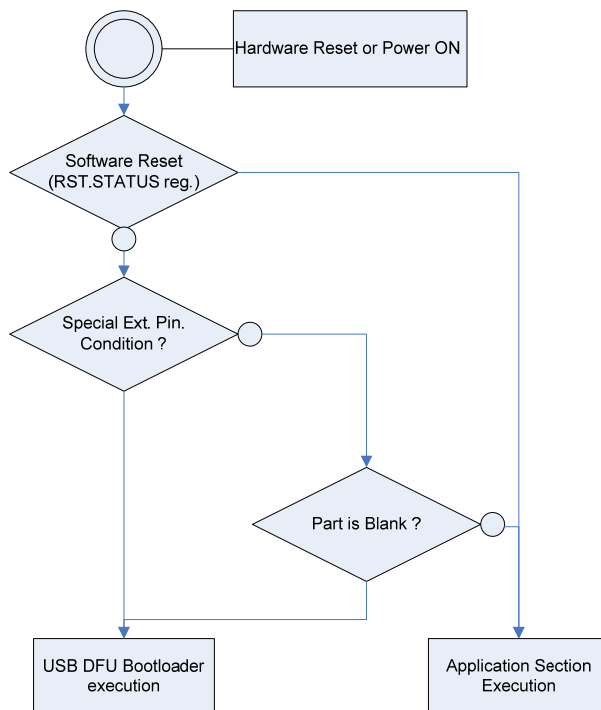
But if the boot loader has been erased or not pre-programmed, to use the on-chip USB DFU boot loader, the parts should be configured as described below:

- Download the on-chip boot loader firmware into the part using regular Atmel AVR tools (Atmel AVR ONE!, STK<sup>®</sup>600, AVRISP mkII, JTAGICE3...). The default boot loader binary for each supported parts can be found in the zip file related to this application note
- The part's fuses should be configured so that the BOOTRST fuse allows the part to reset from the boot loader section of the part
- The part's lock bits should be configured according to the required security level. From more information about boot loader security, please refer to Section 5.3

### 5.2 Boot process

Once the part is configured, the boot loader is executed at each reset/power-on sequence. The boot loader first performs the boot process below (see Figure 5-1) to know whether it should start the USB DFU ISP or the application:

Figure 5-1. On-chip USB DFU boot process.



When the boot loader decides to continue with DFU execution and the part is connected to a USB host, it will enumerate as a USB DFU device. In association with the PC user interface software (FLIP), the on-chip memories can be erased, read or written using the DFU protocol.

Once an application is loaded by DFU, the boot loader execution can be forced at power-on by connecting a specific pin to ground. Table 5-1 gives the default pin configuration used to force boot loader activation:

**Table 5-1. Default pin configuration.**

XMEGA part name	Default I/O pin
ATxmegaxxA3x	PE5
ATxmegaxxC3	
ATxmegaxxA4U	PC3
ATxmegaxxC4	
ATxmegaxxB1	PC6
ATxmegaxxB3	
ATxmegaxxA1U	PF10

### 5.3 Security

When the USB boot loader connection is initiated, the boot loader security state is the same as the product lock bits settings. Production parts with pre-programmed boot loader are shipped with read/write security bits set. This allows protecting the on-chip memories content from read/write access over the USB interface. Thus, the only DFU command allowed after a USB boot loader connection is a “Chip Erase” command (application section). If the DFU boot loader is loaded in a blank part, the lock bits configuration of the part should be set according to the proper security level required by the end user application (meaning the lock bits of the part can not be configured using the DFU boot loader itself).

## 6. Using the boot loader

### 6.1 Activating the boot loader

The DFU session is activated according to the boot process conditions described in Section 5.2. The DFU can be activated in two ways:

- External I/O condition: Tying a specific pin to ground when the part is exiting reset (POR, BOD, External Reset). The default pin can be changed re-compiling the boot loader for a specific hardware target (see Section 6.4 Customizing the boot loader)
- Internal firmware request: The user application can decide to start a DFU session. This can be achieved by jumping to a specific address in the boot loader firmware. The entry point to start a DFU session initiated by a user application firmware jump is `BOOT_SECTION_START + 0x1FC` for all devices

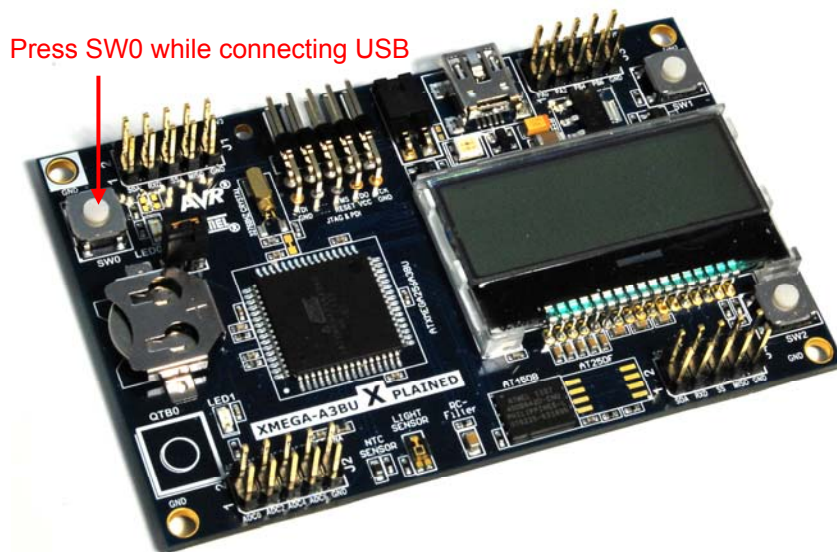
Note: As the DFU firmware is fully interrupt driven, be sure to execute the call outside an interrupt routine execution.

#### 6.1.1 Forcing boot loader mode with XMEGA-A3BU Xplained

The default Atmel ATxmegaA3BU boot loader is configured so that it is possible to activate the DFU boot loader by using the SW0 push button on the Atmel XMEGA-A3BU Xplained kit.

To activate the DFU mode, press SW0 while connecting the kit to the USB.

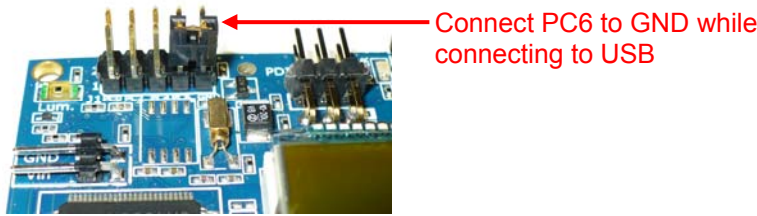
Figure 6-1. Press SW0 while connecting USB.



#### 6.1.2 Forcing boot loader mode with XMEGA-B1 Xplained

The default Atmel ATxmega128B1 boot loader is configured so that it is possible to activate the DFU boot loader using a single jumper on PC6 with the Atmel XMEGA-B1 Xplained kit.

Figure 6-2. Connect PC6 to GND while connecting to USB.



## 6.2 Connecting the boot loader to a USB host

Once the boot loader is properly loaded, activated and connected to a host, it will appear as a new USB device. Depending on the operating system used, the installation procedure is described here after. The drivers required to connect the part to a Windows operating system are delivered within the FLIP application that allows connecting to the DFU boot loader. A FLIP version greater than or equal to 3.4.4 is required to support the Atmel XMEGA DFU boot loaders.

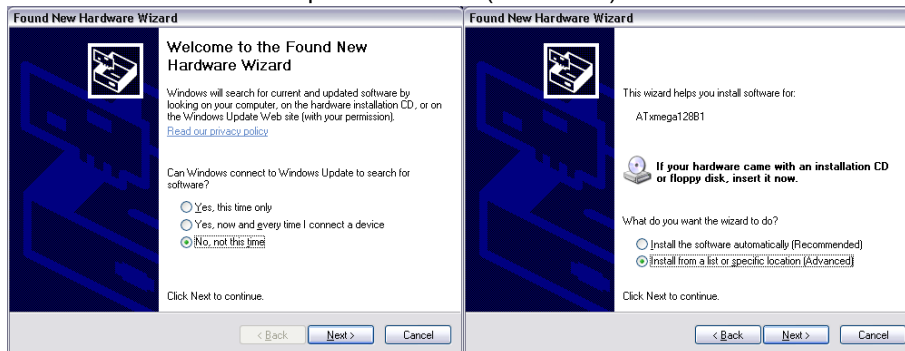
FLIP can be downloaded from:

[http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=3886](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3886)

### 6.2.1 Drivers installation for Windows XP

The procedure here after is valid for Windows XP; for Windows Vista® or Windows 7 the procedure is different and described in Section 6.2.2.

- When prompt with the “New Hardware Wizard”, select “No, not this time”
- And “Install from a list or specific location (Advanced)”



- Browse to the FLIP USB installation folder (“C:\Program Files\Atmel\Flip 3.x\usb”)
- And “Install from a list or specific location”

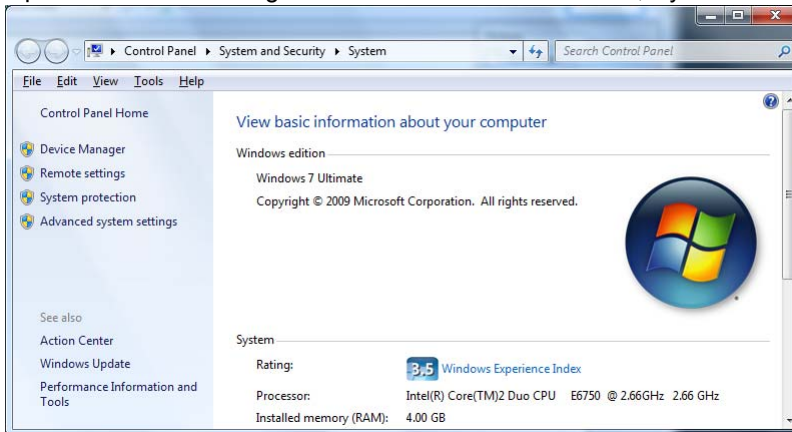




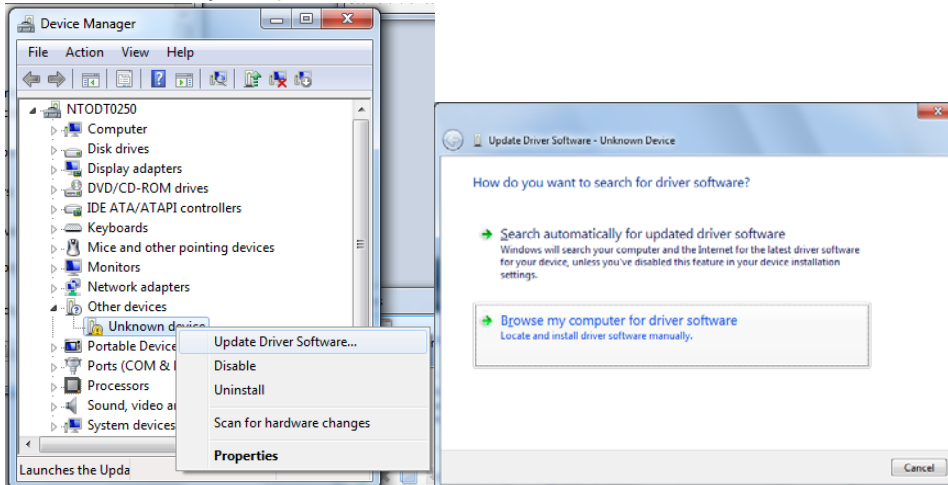
## 6.2.2 Drivers installation for Windows Vista or Windows 7

For Windows Vista or Windows 7 the DFU device installation procedure will look as described below:

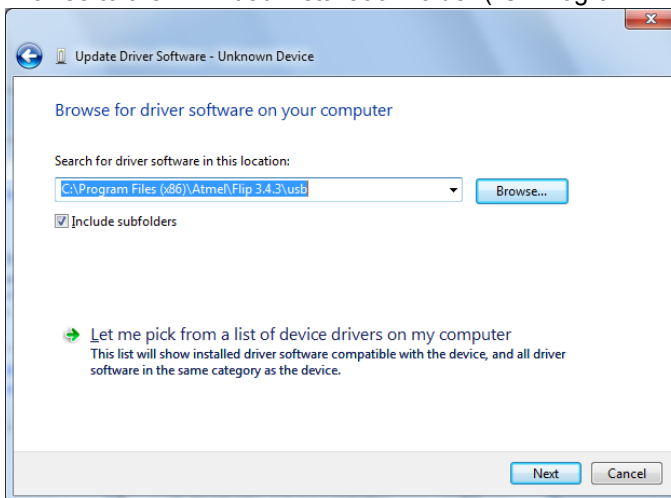
- Open the device manager located in the “Control Panel, System & Security, System”



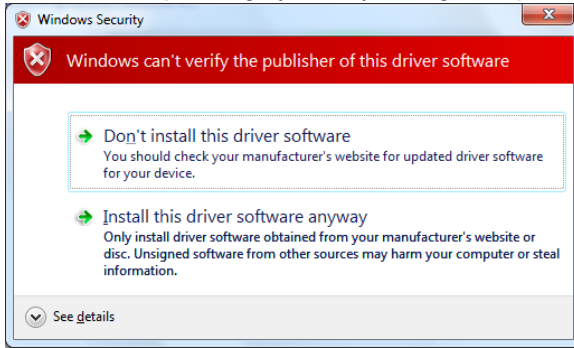
- Locate the new “Unknown device” and select “Update Driver Software”
- Select “Browse my computer for driver software”



- Browse to the FLIP usb installation folder (“C:\Program Files\Atmel\Fliip 3.x.x\usb”)



- With 64-bits operating system, you might be prompt with this warning:

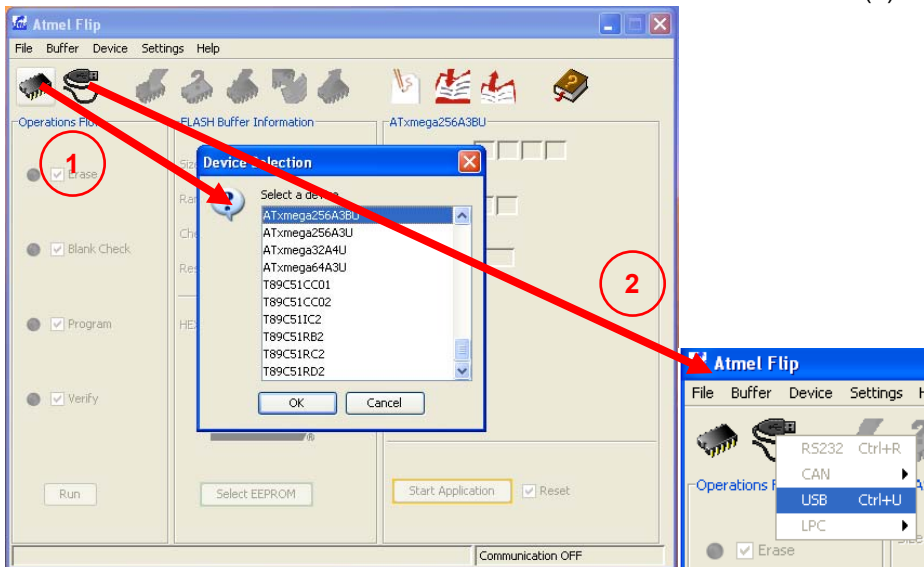


- Choose “Install this driver software anyway”

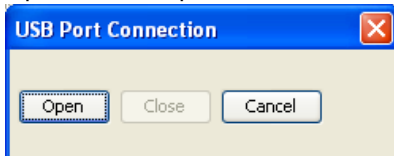
### 6.3 Getting started with FLIP

Once the boot loader is properly recognized and installed on the operating system, the part can be programmed using FLIP. This section gives a short step-by-step introduction about how to use FLIP to program a part. For detailed documentation, please refer to the online help for FLIP.

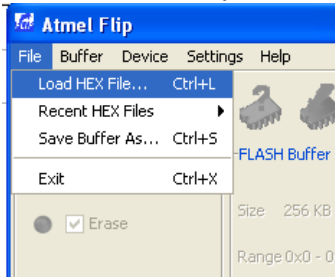
- Start FLIP
- Select the part from the device selection list available with the ‘Device’ icon (1)
- Select “USB” as communication media available with the “USB cable icon” (2)



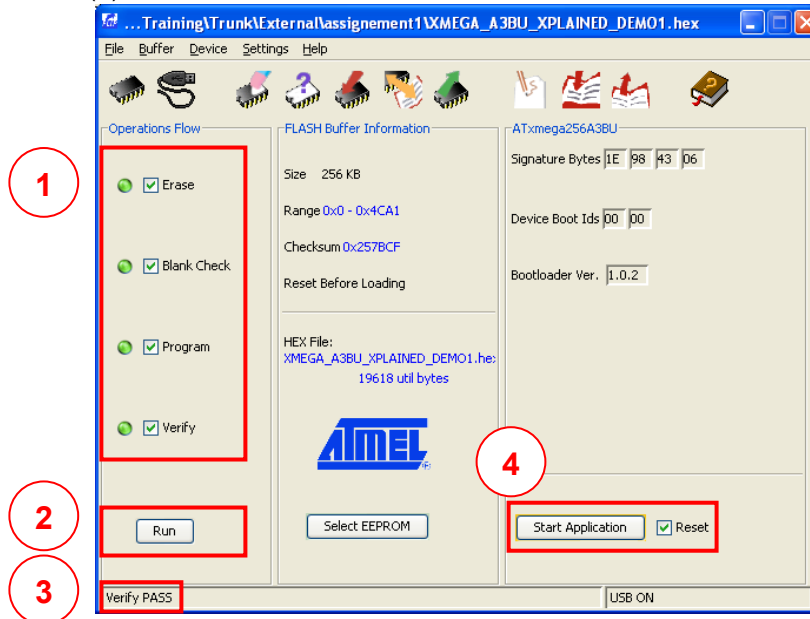
- Open the USB port to connect to the target



- Select the .hex file you want to download using “File”-> “Load HEX File” menu



- Select the ‘Erase’, ‘Blank check’, ‘Program’ and ‘Verify’ operation check boxes (1) and press the ‘Run’ button (2)



You should get confirmation in the bottom section of the window (3) that the FLASH was successfully programmed and verified.

- Press ‘Start Application’ button (4) to execute the loaded application

## 6.4 Customizing the boot loader

The boot loader source code is available within the ASF so that it is possible to customize its behavior and especially the default I/O pin used to force DFU session. This might be needed when the default I/O pin is not available or compatible on the target hardware.

The default I/O pin is referenced in the “conf\_isp.h” file located in the

“/common/services/usb/class/dfu\_flip/device/bootloader/xmega/conf” of the ASF. The I/O pin selection is performed thanks to the following define (the example here under is for the Atmel ATxmega256A3U):

```
#define ISP_PORT_DIR      PORTE_DIR
#define ISP_PORT_PINCTRL PORTE_PIN5CTRL
#define ISP_PORT_IN      PORTE_IN
#define ISP_PORT_PIN      5
```

Note: To ensure the boot loader binary fits the available boot section of the Atmel AVR XMEGA, the source code as well as project files are available within the standalone package for the IAR™ compiler of the ASF.

## 7. Software information

### 7.1 Software revision history

#### 7.1.1 Version 1.0.2 – 2011/09/01

Initial boot loaders release for the following Atmel parts:

ATxmega256A3BU, ATxmega256A3U, ATxmega192A3U, ATxmega128A3U, ATxmega64A3U, ATxmega32A4U, ATxmega16A4U, ATxmega128B1, ATxmega64B1, ATxmega128B3, ATxmega64B3.

#### 7.1.2 Version 1.0.3 – 2012/03/01

Initial boot loaders release for the following Atmel parts:

ATxmega128A1U, ATxmega64A1U.

Updated version for all other supported parts:

- Fixed system clock configuration to load factory 48MHz RC calibration value from factory raw

#### 7.1.3 Version 1.0.4 – 2012/08/01

Initial boot loaders release for the following Atmel parts:

ATxmega384C3, ATxmega256C3, ATxmega128C3, ATxmega64C3, ATxmega32C4, ATxmega16C4.

Updated version for all other supported parts:

- Reduced initial assumption about system clock initialization to ease bootloader activation from user application call

**Atmel Corporation**

2325 Orchard Parkway  
San Jose, CA 95131  
USA

**Tel:** (+1)(408) 441-0311

**Fax:** (+1)(408) 487-2600

[www.atmel.com](http://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  
Parking 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

© 2012 Atmel Corporation. All rights reserved. / Rev.: 8429B-AVR-08/2012

Atmel®, logo and combinations thereof, AVR®, STK®, XMEGA®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® and others are registered trademarks or trademarks of Microsoft Corporation in U.S. and or other countries. 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.