

AVR916: Upgrading the Flash using a U-Disk

Features

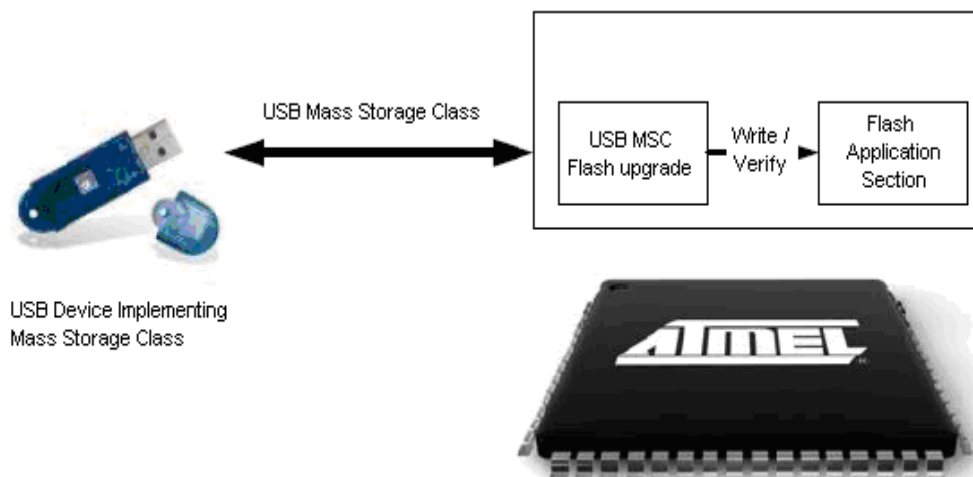
- In-System Programming (ISP) and Field Upgrade
- Upgradable using U-Disk with an Upgrade file
- Supports Intel® hex format
- No computer required
- Confirmation on Flash upgrade
- Supports AVR® USB 7 series
- Integrity of the upgraded file is guaranteed using CRC

1 Introduction

In-System Programming for Atmel® USB 7 series controllers with USB device implementing the Mass Storage class (USB drive), without:

- Removing the part from the system.
- External interface other than USB.

Figure 1-1. Physical Environment



8-bit **AVR**[®]
Microcontrollers

Application Note

Preliminary

Rev. 8255A-AVR-08/09





This application note concentrates over the application where the user would require updating the firmware with the help of a mass storage device. Note that all the 8 bit AVR USB devices are shipped with a pre-programmed USB Device Firmware Upgrade (DFU) bootloader. In the rest of this document, we will suppose that the USB DFU bootloader is always present and located in internal Flash memory. For more information on the USB DFU bootloader, please refer to the USB AVR DFU Bootloader datasheet in www.atmel.com

2 Hardware Requirements

The Upgrading of flash by U-Disk application requires the following hardware:

- ATSTK525 or Other Hardware.
 - AVR USB 7series device:
 - USB mini A to A receptacle adapter
 - USB Type A to mini B cable
 - Mass Storage Device

3 In-System programming and Device Firmware Upgrade

To program the device you can use one of the following methods:

- The JTAG interface using the JTAGICEmkII (with out erasing the DFU bootloader)
- The USB interface thanks to the factory DFU bootloader and FLIP(1) software

Please refer to FLIP(1) help content to see how to install the USB driver and program the device through the USB interface.

Note: 1. FLIP is software provided by Atmel to allow the user to program the Atmel devices through the USB interface (No external hardware required) thanks to the factory DFU bootloader.

Note: With JTAGICE MKII be careful with the box "erase before programming" in AVR Studio®. If checked, the DFU bootloader is deleted before to programming.

4 Related Items

- AT90USB1287 Series Datasheet:
http://www.atmel.com/dyn/products/product_card.asp?part_id=3875
- AVR Studio:
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725
- AVR USB DFU Bootloader Datasheet:
http://www.atmel.com/dyn/resources/prod_documents/doc7618.pdf

5 Application

This application helps the upgradation of the firmware when a U-Disk (mass storage device) is plugged to the AVR USB controllers which could act as an OTG. The U-Disk is plugged to the board, there is a check done for the availability of the upgradable Hex file, if available is the address of the Hex file is valid and if it is valid then is there any error.

The upgrade file should start with “**upgrade**” and files without the filename starting with “**upgrade**” will not be considered for up gradation. The prevention for the wrong data being flashed is taken care by checking for the CRC and the address location where it has to be written.

Once the file which is to be upgraded is error free, the file is updated in the flash section as in the Hex file.

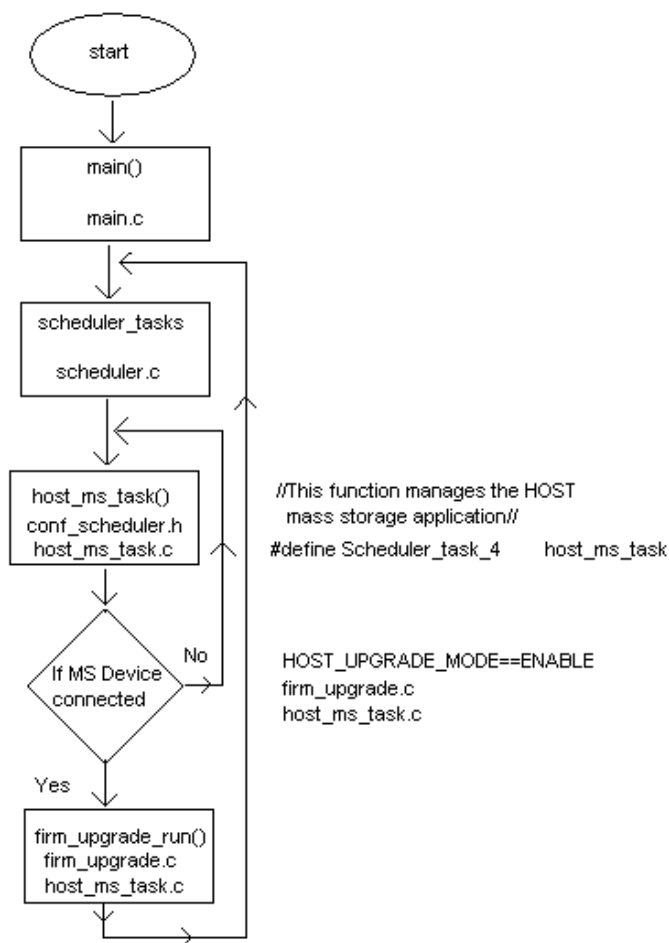
There is a text file which is written back in the U-Disk called "Status.txt". This file will have the status of the operation done. There is a possibility of the following message which could be shown in the Status are

- 1) **!! Error to open upgrade file**
This could happen when there was a possible error in opening the upgrade file.
- 2) **!! Error in HEX file format**
This could happen when there was a possible error in the Hex file format.
- 3) **!! Upgrade zone not authorized**
This could occur when there was a effort for programming in the unauthorized zone of the flash section.
- 4) **!! The bootloader is not loaded**
This will occur if the DFU Bootloader is not programmed.
- 5) **!! Program in flash is BAD**
 !!! Check if the bootloader is loaded in chip.
This is possible if there was error while programming.
- 6) **Upgrade successful**
When the Flash upgrade is successful
- 7) **No upgrade file**
When there is no upgrade file

6 Process Flow

This section basically concentrates over the firmware process flow for the application. In Figure 6.1, the flow chart explains the code process in a flow chart.

Figure 6.1

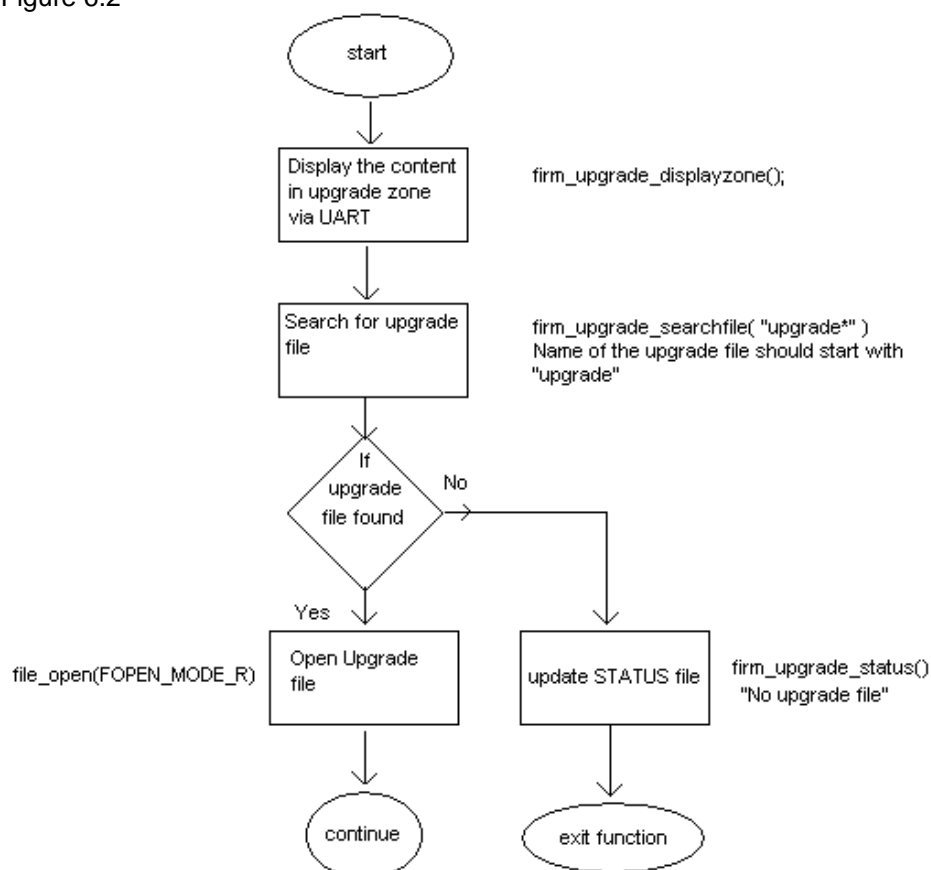


Function which controls the firmware upgrade is **firm_upgrade_run()** located in the file **firm_upgrade.c**.

The process which is involved in the function after a U-Disk is connected is as follows.

- 1) Display the current data in the flash section which is there in the current upgrade zone via UART. Search for valid **upgrade files** in the **U-Disk**, if not available it updates the **status file** with **"No upgrade file"** and exits the function. Figure 6.2 shows the process flow.

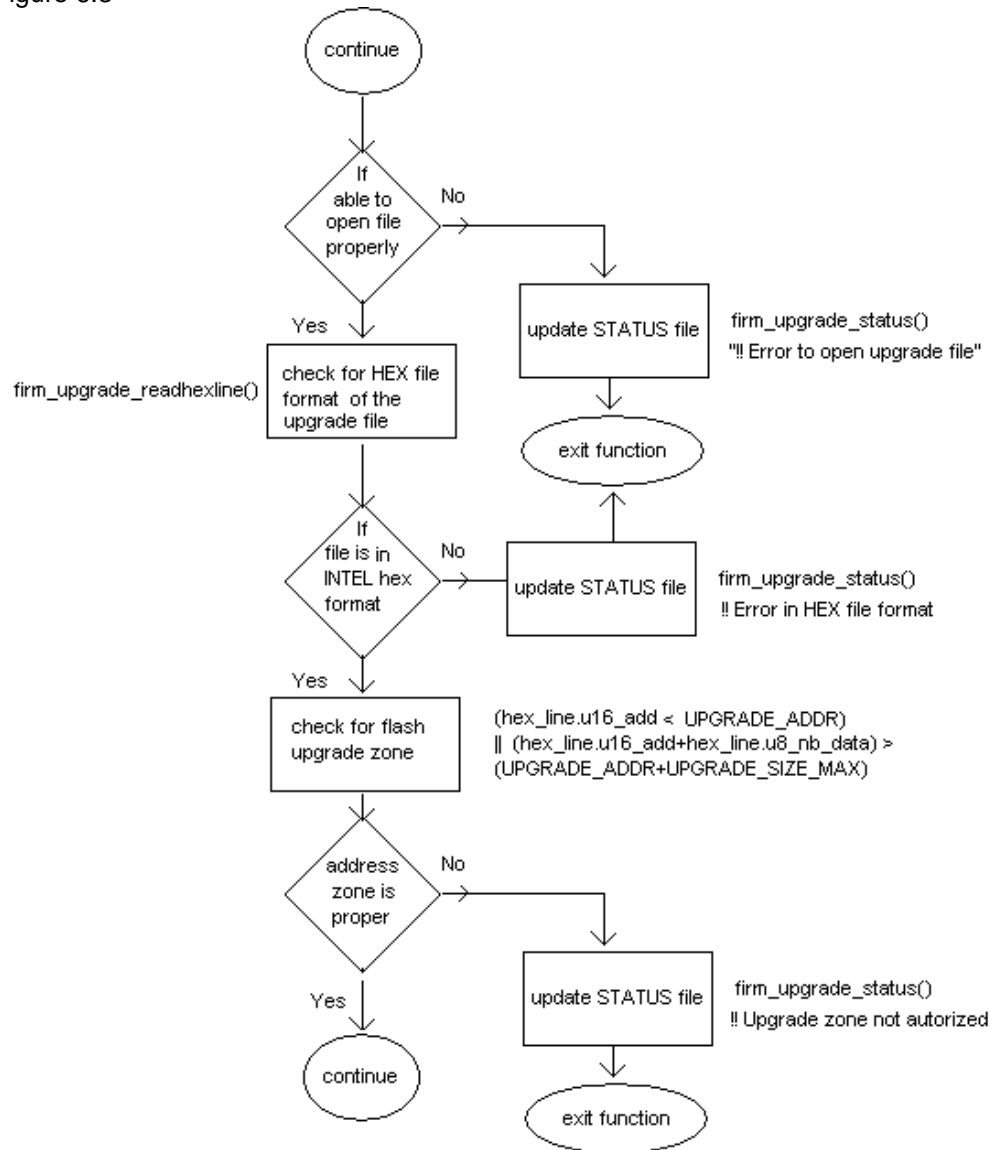
Figure 6.2



- 2) If found, open the file. In case of error in opening the file it updates the **status file** with **"!! Error to open upgrade file"** and exits the function. Check for **Hex file format** and the valid **upgrade zone** in the flash is done. In case of error the **status file** is updated as **"!! Error in HEX file format"** and **"!! Upgrade zone not authorized"** for respective errors. While checking for **Hex file format**, there is also error check which is done for the values in the upgrade file. Apart from the header (INTEL HEX format) of the each line rest all fields including the checksum is added which should yield in LSB value to be '0'. The function being involved is **firm_upgrade_readhexline (St_hex_line* line)**.

Flow chart is shown in figure 6.3

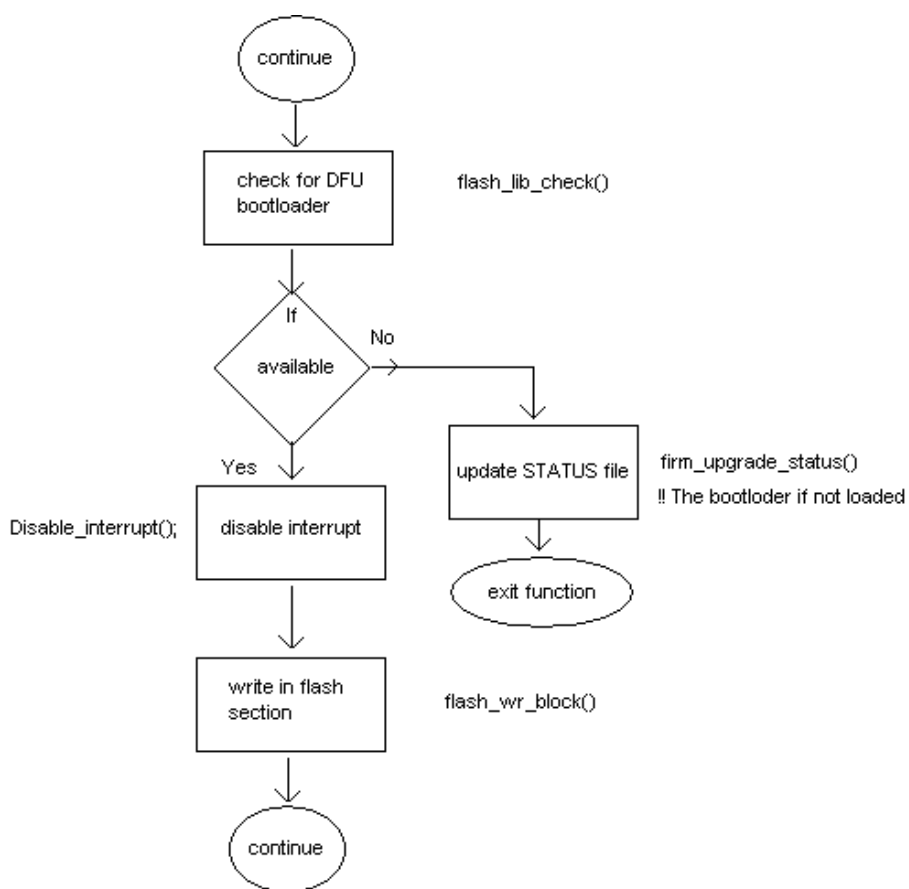
Figure 6.3



When all the check which is done is through without errors then the interrupts are disabled and **flash_wr_block(hex_line.datas, hex_line.u16_add, hex_line.u8_nb_data)** is called which is in turn responsible for the updating the defined flash section. The updated flash section is read back to verify using **flash_rd_byte((U8 farcode*)hex_line.u16_add)**, when found error free the **status file** is updated with **"Upgrade successful"** and if error found the **status file** has **"!! Program in flash is BAD \n\r!!! Check if the bootloader is loaded in chip."**

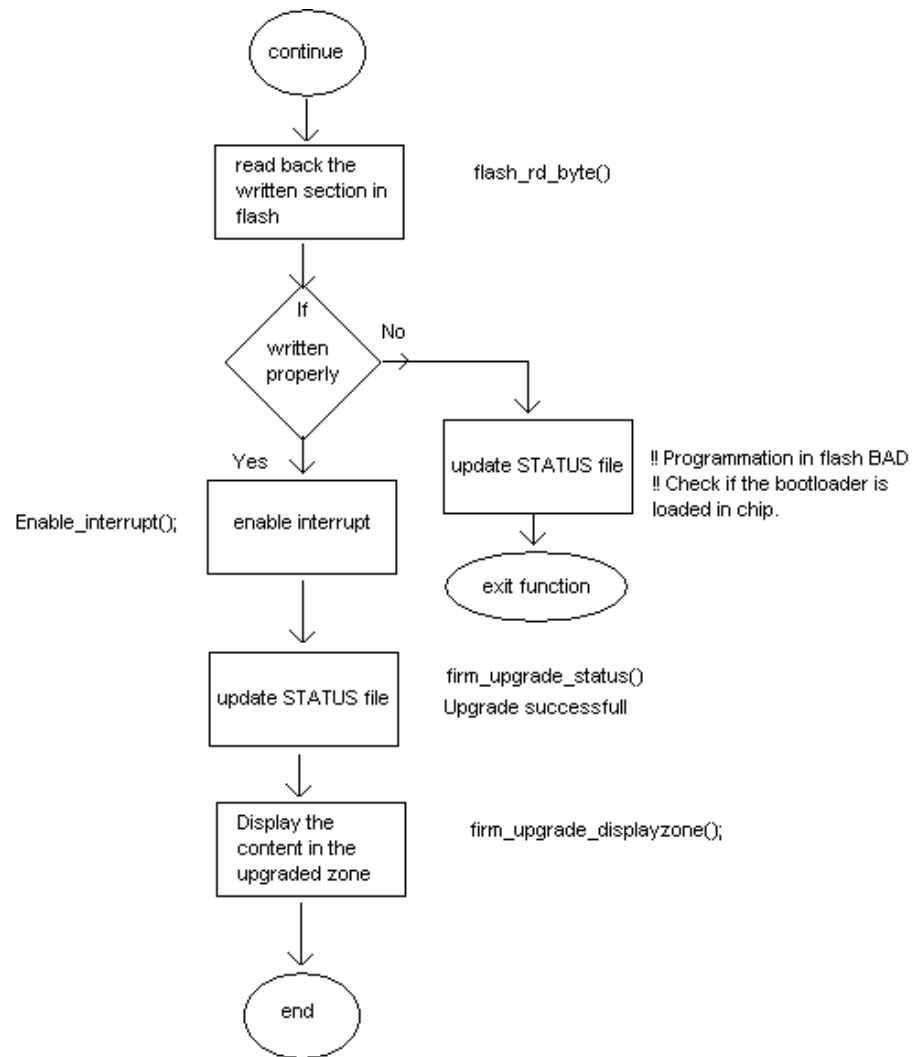
Figure 6.4 and Figure 6.5 shows in the flow chart regarding the check for availability of the DFU bootloader and flash upgrade process.

Figure 6.4



During the process of upgrading the flash, the status of the microcontroller is displayed in the U-Shell. The Upgraded flash section is shown in the UART.

Figure 6.5



7 Defining Upgrade Zone

The file **firm_upgrade.c** has the macro's `UPGRADE_SIZE_MAX` and `UPGRADE_ADDR` which is used to define the flash section length and the flash address.

```
#define UPGRADE_SIZE_MAX
#define UPGRADE_ADDR
```

These macros can be used to change the respective field depending on the end requirement.

8 Using U-Shell

The demo comes with an UART Interface, which could be used for seeing the status getting the live demo with oneself. When using with STK[®]525 starter kit, and connected with USB mini A to A receptacle adapter, without the Mass storage device, the UART terminal in the PC will get the message as shown

```
-----
      ATMEL AVR uShell
-----
Memory interface available:
a: "On board data flash"
$>
```

After the mass storage device is connected (device without Upgrade file) the following message could be seen. There is a display of the previous section and a search is done for the upgrade file. The result is declared at the end.

```
-----
      ATMEL AVR uShell
-----
Memory interface available:
a: "On board data flash"
$>
Display upgrade zone BEFORE upgrade:
"Hello word!
  It is an update of firmware."
Search upgrade file
...
No upgrade file
```

Mass storage device is connected (device with upgrade file which erases the section with "0") the following message could be seen. There is a display of the previous section and a search is done for the upgrade file. The result is declared at the end,

```
Display upgrade zone BEFORE upgrade:
"Hello word!
  It is an update of firmware."
Search upgrade file
...
Open upgrade file "upgrade-erasespace.hex"
...
Check upgrade file
...
Program FLASH
...
Upgrade successful
Display upgrade zone AFTER upgrade:
```



""

Now the corresponding flash section is erased and we will program it with "upgrade-helloworld.hex "

```
Display upgrade zone BEFORE upgrade:
""
Search upgrade file
...
Open upgrade file "upgrade-helloworld.hex"
...
Check upgrade file
...
Program FLASH
...
Upgrade successful
Display upgrade zone AFTER upgrade:
"Hello word!
  It is an update of firmware."
```

The demo also comes with a folder "UpgradesFiles" which has four files upgrade-ADDERR.hex, upgrade-CRCERR.hex, upgrade-erasespace.hex, upgrade-helloworld.hex.

upgrade-ADDERR.hex describes the status when there is an address error.

upgrade-CRCERR.hex describes the status when there is an in the checksum in the file.

upgrade-erasespace.hex describes the status after erasing the respective flash section.

upgrade-helloworld.hex is a proper file which could be programmed for the demo.

Ushell also comes with lots of other features which could be seen through the command "help" in depth of all the commands is out of scope of this document.

9 Limitations

- 1) The location of this demo other than address 0x00 needs to be taken care in the interrupt routine of the user application.
- 2) USB DFU bootloader presence is a must in the boot section for upgrading the file using this demo.
- 3) This demo can be used from IAR™ only.



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

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 ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, 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 OF 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 product 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's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2009 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, AVR®, AVR® logo, AVR Studio®, STK® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.