

AT89C5131A Starter Kit

.....
Software User Guide





Table of Contents

Section 1

Introduction	1-1
1.1 Abbreviations	1-1

Section 2

Getting Started.....	2-3
2.1 Hardware Requirements	2-3
2.2 Software Requirements.....	2-3
2.3 Default Hardware Settings	2-3
2.4 FLIP Software	2-4

Section 3

HID Keyboard Demonstration Program	3-7
3.1 Stand-alone Application	3-7
3.2 C51 Generic Board Application.....	3-7

Section 4

Software Description.....	3-9
3.1 Architecture Overview	3-9
3.2 Application Description.....	3-10
3.3 Libraries Description	3-11



Section 1

Introduction

This document describes the AT89C5131A HID keyboard demonstration application, as well as the In-System Programming Tool (FLIP), dedicated to the AT89C5131A microcontroller. This software demonstration is an implementation example of a HID keyboard which is USB Chapter 9 compliant.

1.1 Abbreviations

- USB: Universal Serial Bus
- HID: Human Interface Device
- ISP: In-System Programming
- DFU: Device Firmware Upgrade

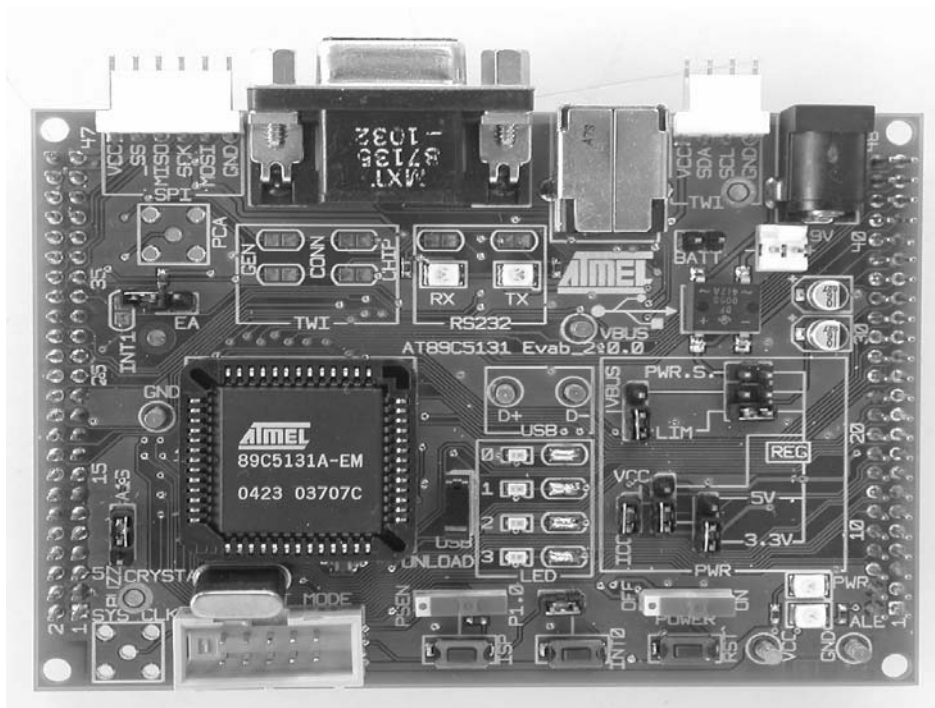


Section 2

Getting Started

-
- 2.1 Hardware Requirements** The demonstration application requires the following hardware:
- AT89C5131A evaluation board
 - AT89C5131A microcontroller (includes a USB bootloader)
 - A-B USB cable
 - PC running Windows® (98, Me, 2000 or XP) or Linux® with a 1.1 or 2.0 USB Host
-
- 2.2 Software Requirements** The following software is necessary to use the demonstration program. The software can be found in the accompanying CD-ROM. Updated FLIP software is available on the Atmel web site.
- Flexible In-System Programming (FLIP) software tool
 - usb_hid_kbd.hex file
-
- 2.3 Default Hardware Settings** The AT89C5131A evaluation board must be configured as follows:
- Power supply: from V_{BUS} through the limiter and the 3.3V regulator.
 - Quartz frequency: 16 MHz
 - EA jumper: Off
 - INT0 jumper: On
- 2.3.1 Setting ISP Mode**
1. When BLJB is set (= 0) (default setting), the ISP mode is enabled. Plug the USB cable into the PC. The bootloader will automatically execute.
 2. When the BLJB is reset (= 1), the ISP is not set. The following procedure must be executed.
 - Press the ISP button (SW4) and plug the USB cable into your PC, or
 - Press both the Reset (SW3) and ISP (SW4) buttons. First release the Reset button and then the ISP button. The device enters ISP mode. The user needs to re-enumerate the device using the USB Unload button (SW1).

Figure 2-1. AT89C5131A Evaluation Board



2.4 FLIP Software

FLIP software runs on Windows® (98/2000, NT, XP). FLIP supports In-System programming of Flash C51 devices through RS-232. The latest version of FLIP software can be found on the Atmel web site. A Linux version of FLIP is also available.

Note: See FLIP User documentation for USB Pilot Installation Procedure.

2.4.1 Flash Programming

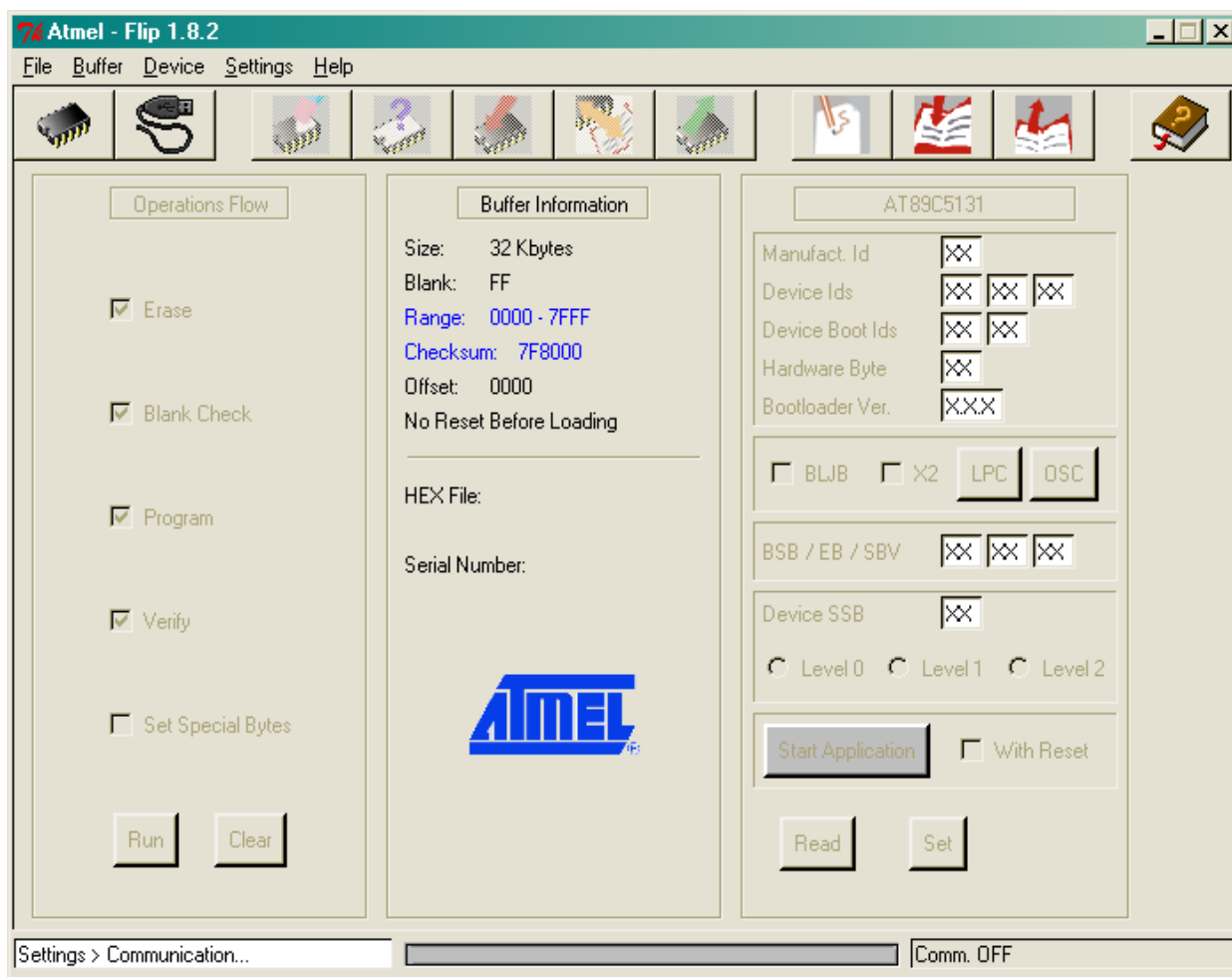
The HID keyboard hex file (usb_hid_kbd.hex) must be programmed in the user Flash memory of the AT89C5131A. There are two different ways to perform the Flash programming:

- Use a programmer which supports the AT89C5131A part
- Use the Atmel FLIP tool (on a PC connected to the evaluation board using a USB cable)

In this section the user will program the AT89C5131A microcontroller via USB using FLIP software. The following procedure will guide you through the programming of the demonstration program.

1. Run FLIP software (see Figure 2-2).
2. From the Device Menu, choose “Select” and select the device (AT89C5131A) that is connected to the evaluation board.

Figure 2-2. FLIP Window

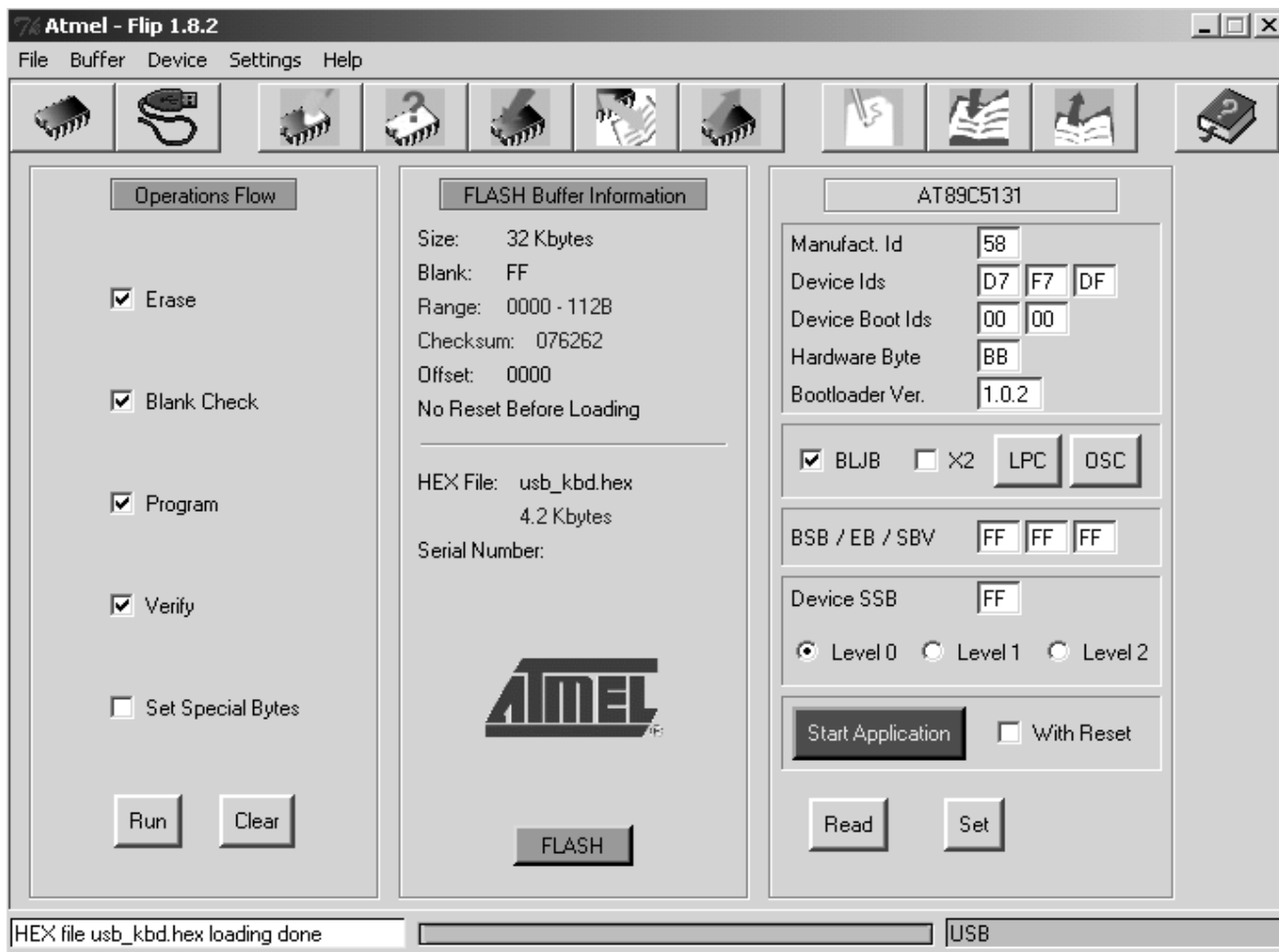


3. Click the “Set Communication” button.



4. Initialize the communication by selecting the “Open” button in the USB Port Connection pop-up window.
If the connection is successful, the FLIP window should look like Figure 2-3.

Figure 2-3. Successful Connection on FLIP



5. In the File menu, select “Load HEX” and choose the demonstration program “usb_kbd.hex”.
6. The message “HEX file usb_kbd.hex loading done” is displayed at the bottom of the FLIP window.
7. Ensure the following check boxes are selected in the Operations Flow section of FLIP:
 - Erase
 - Blank Check
 - Program
 - Verify

These are the operations that will be performed on the microcontroller.

8. Press the “Run” button. Programming is executed. The “Memory Verify Pass” message confirms programming is successful and that the microcontroller has been programmed.
9. Ensure the BLJB box is unchecked. Press “Set”, then “Read” to verify that the BLJB is blank (=1), in order to boot the demonstration program after the next reset.
10. Ensure the “With Reset” box is checked, then press the “Start Application” button.



Section 3

HID Keyboard Demonstration Program

The purpose of the HID Keyboard demonstration program is to send numeric data via the keypad through the AT89C5131A microcontroller to a host PC via USB.

The HID Keyboard demonstration program is used with the AT89C5131A board (Stand-alone Application).

-
- | | | |
|------------|--------------------------------------|--|
| 3.1 | Stand-alone Application | <p>The AT89C5131A board can be used to transmit a message stored in the MCUs Flash memory and display the message on a PC text editor.</p> <p>This demonstrates the AT89C5131A microcontrollers “Plug & Play” and “Hotplug” capability for any USB application.</p> <ol style="list-style-type: none">1. Ensure the USB cable is connected between the AT89C5131A evaluation board and the PC.2. Open the Notepad application or any text editor on the PC.3. Click the INTO button of the AT89C5131A evaluation board. The message “Welcome to the HID keyboard” is displayed on the text editor. |
| 3.2 | C51 Generic Board Application | <ol style="list-style-type: none">1. Plug the USB cable into the USB host socket on one end and to the AT89C5131A evaluation board on the other end.
The message “AT89C5131A demo” appears on the first line of the LCD. The LED 0 of the AT89C5131A evaluation board blinks. This indicates that the USB cable is correctly connected and the SOF messages are correctly sent by the Host controller and understood by the AT89C5131A microcontroller.2. The PC Operating System <i>may</i> ask for a driver, specify the proper directory as indicated in the Section “Note on Windows Drivers” below.
Once the correct driver is loaded, the PC sends a “SET CONFIGURATION” USB message and the “Enumeration Pass” message appears on the third line of the LCD.3. Open a calculator or a text editor. When the buttons of the C51 generic board keypad are keyed, the number or the operation is displayed in the text editor or calculator. |
-

Notes: 1. Ensure the Num Lock key is activated on your PC keyboard number pad. The demonstration program is supporting QWERTY keyboard configuration. All other configuration (such as AZERTY) will result in wrong display of some characters.

3.2.1 Note on Windows Drivers

The HID keyboard example can directly interface with native drivers under Windows 98, Me, 2000 and XP. After initial USB connection, Windows *may* ask for drivers. Indicate the following path:

- <Windows hard drive>:\WINDOWS\inf (for Windows 98 and Me)
- <Windows hard drive>:\WINNT\inf (for Windows 2000 or XP)





Section 4

Software Architecture

4.1 Architecture Overview

The HID keyboard demonstration firmware is based on a scheduler in the free running mode.

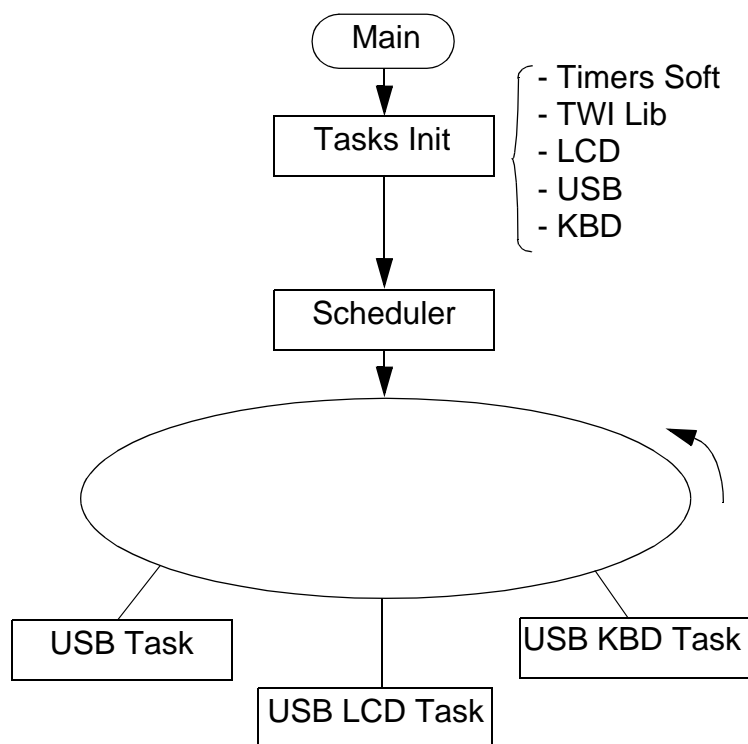
The main program only enables the interrupts and launches the scheduler.

The first process of the scheduler is the initialization of all the peripherals and of the associated variables.

Once the initialization process is complete, the scheduler launches each task one after the other. The first task is the USB task. This task manages the Default Control Endpoint for the enumeration process and the HID keyboard control.

The second task called by the scheduler is the USB LCD task that manages all the LCD display in accordance with the USB bus status (connected or not, enumeration process passed or not, etc.).

The third and last task called by the scheduler is the USB KBD task that transmits the keys pressed to the USB controller.

Figure 4-1. USB HID Keyboard Firmware Architecture Overview

4.2 Application Description

- 4.2.1 Configuration** The USB HID keyboard configuration is performed according to the “USB Device Class Definition for Human Interface Device – Firmware Specification” version 1.1 (4/7/99).
- 4.2.2 Implementation**
- 4.2.2.1 usb_task_init()** The `usb_task_init()` function is called during the initialization phase. It enables the USB controller, configures the PLL in order to generate the 48 MHz required by the USB controller, and enables and configures the Endpoint 0. In addition, this function performs a USB Detach/Attach in order to be re-enumerated by the Host. This could be necessary after a Start Application is performed by the bootloader.
- This function also initializes the LED controller of the AT89C5131A part.
- 4.2.2.2 usb_task()** The `usb_task` manages the USB events: Suspend, Resume, USB Reset and Start of frame. When the USB bus is in Suspend state, the LED 3 is turned On. If a USB Resume occurs, the firmware turns Off this LED.
- Each time a Start of Frame occurs on the USB bus, a counter is increased. When this counter reaches 255, this counter is reset and the LED 1 is toggled.

When a SETUP token is detected on the Endpoint 0, the `usb_task` launches the enumeration process routine. Once the Control Transaction has been completed, the enumeration process routine exits.

A Transmit Complete flag (TXCMPL) detection on the Endpoint 1 (IN endpoint for HID keyboard) means that a HID report has been successfully transmitted to the Host. The `usb_task` then clears the Transmit Complete flag in order for the USB keyboard application to send the next HID report.

- 4.2.2.3 `usb_lcd_task()`** The `usb_lcd_task()` displays the status of the USB on the LCD:
- USB Connected
 - USB Suspend
 - Enumeration Process Passed
- 4.2.2.4 `usb_kbd_task()`** This `usb_kbd_task()` determines if a new report has to be sent to the Host in function of the keyboard scan. This function is also in charge of translating the keyboard scan result into HID comprehensive bytes. The key codes sent correspond to the USB HID Usage Tables document for a QWERTY keyboard.
- 4.2.3 Customization**
- 4.2.3.1 Enumeration Customization** The developer can change the USB parameters of the descriptors that are sent during the enumeration process in order to personalize the application. Please refer to the enumeration process application note.
- 4.2.3.2 Keyboard Map** The developer can change the translation between the keyboard map and the HID comprehensive bytes. The values of the `usb_geneb_keys` list of the `usb_task.c` file must be replaced.
- 4.2.3.3 Additional Features** The developer can add other features by adding new tasks in the scheduler.

4.3 Libraries Description

- 4.3.1 USB** The USB management uses two different libraries:
- One for the USB enumeration process
 - One for the low level of the USB controller
- The enumeration process management is contained in the `usb_hid_enum.c` and `usb_hid_enum.h` files. As it is written in the file names, this enumeration process is specific for this application because some HID specific messages require the default control endpoint. However, this enumeration process management can easily be adapted for other applications.
- The low level library gives an easy and comprehensive access to the USB controller. This library allows to manage USB events (USB Reset, Suspend/Resume, Start Of Frame), to configure the USB controller and the endpoints, and to send or receive messages over each endpoint. These drivers are contained in the `usb_drv.c` and `usb_drv.h` files.
- 4.3.2 LED** The `5131_drv.h` file includes the on-chip LED controller driver. This library allows to configure the LED controller, to turn On/Off or to toggle each or every LED.

4.3.3 Keyboard

The keyboard management example has been written for a 4 x 4 keyboard.



Atmel Corporation

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

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

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

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2004. All rights reserved. Atmel®, logo and combinations thereof are registered trademarks, and Everywhere You Are(SM) are the trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.



Printed on recycled paper.