

# AVR32907: AT32UC3C-EK Getting Started Guide

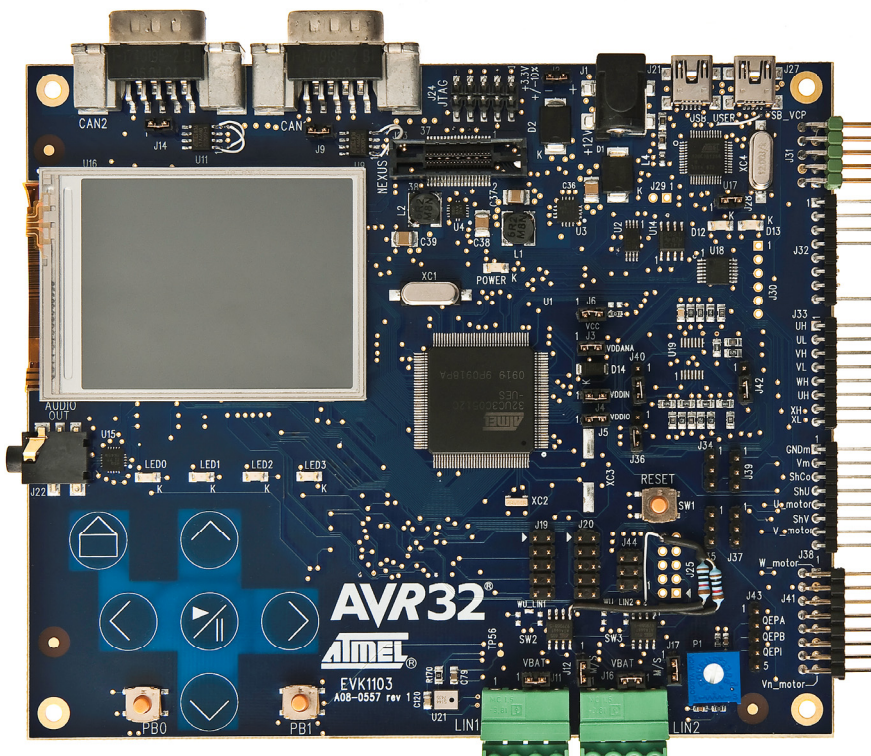
## Features

- Powering up the Board
- Block Diagram
- Using the Preloaded Firmware

## 1 Introduction

The AT32UC3C-EK is a reference design and development system for the 32-bit AVR<sup>®</sup> AT32UC3C0512C microcontroller from Atmel Corporation. The kit is equipped with a rich set of peripherals that make the AT32UC3C-EK a perfect evaluation platform. This guide shows the user how to quickly get started with this kit.

Figure 1-1 AT32UC3C-EK board



32-bit **AVR<sup>®</sup>**  
Microcontrollers

Application Note

Rev. 32137A - AVR32-02/10





## 2 Getting Started

### 2.1 Powering up the Board

The AT32UC3C-EK offers four interfaces to power the board, from the USB Virtual Com Port (USB VCP) interface, from the USB USER interface, or from an external source connected to the J1 or J2 headers. If two power sources are applied at the same time the following priorities are used to select the current valid power source:

1. USB VCP (J27)
2. USB USER (J21)
3. External power supply jack (header J1)
4. External power supply 2-pins header (header J2)

That means that if the board is connected to a USB host it has the highest priority and will therefore serve as current power supply. Other power supplies that are connected to the board at the same time will be ignored.

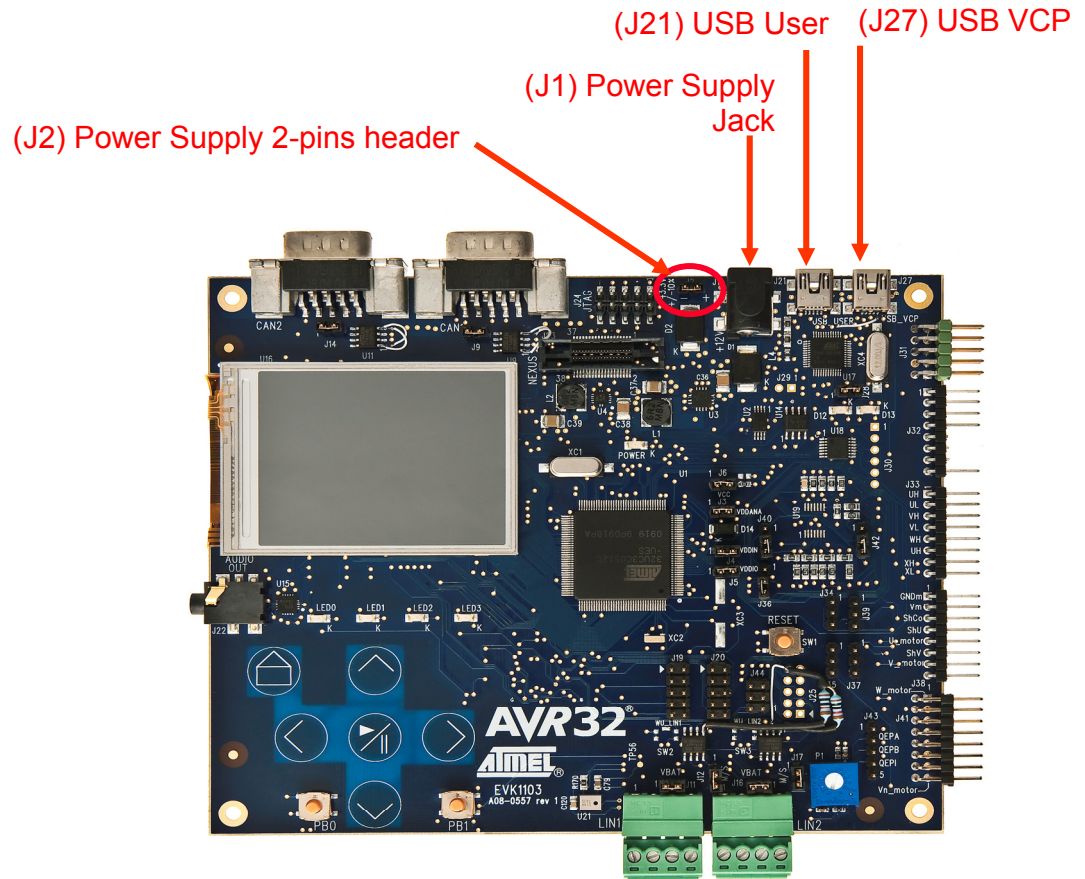
The input range on the external power supply header is from 3.3V to 5V for J2 and from 9V to 12V for J1.

To power the board from the USB VCP or from the USB USER connector, use the "Mini-B plug to std-A plug 1.5m USB2.0" cable that is included in the kit and connect it between the host and the board USB VCP connector.

When the board is powered, the green LED (marked with "POWER") will light up and the default board firmware will start to run.

More information about the different power inputs is available in the hardware user's guide.

Figure 2-1 Possible power input interfaces on the AT32UC3C-EK



## 2.2 Connecting the CAN interface

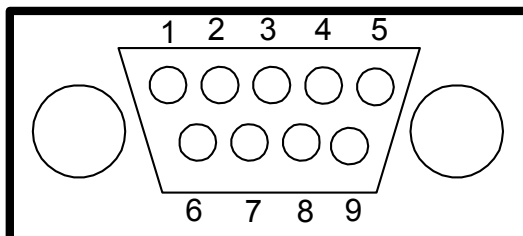
The AT32UC3C-EK offers two CAN interfaces to connect the board to external CAN network:

1. DB9 connector CAN Node 1
2. DB9 connector CAN Node 2

**Table 2-1** CAN connector DB9 Pin-out

Pin #	Signal Names	Signal Description
1	Reserved	
2	CAN_L	Dominant Low
3	CAN_GND	Ground
4	Reserved	
5	CAN_SHLD	Shield, Optional
6	GND	Ground
7	CAN_H	Dominant High
8	Reserved	
9	CAN_V+	Power, Optional

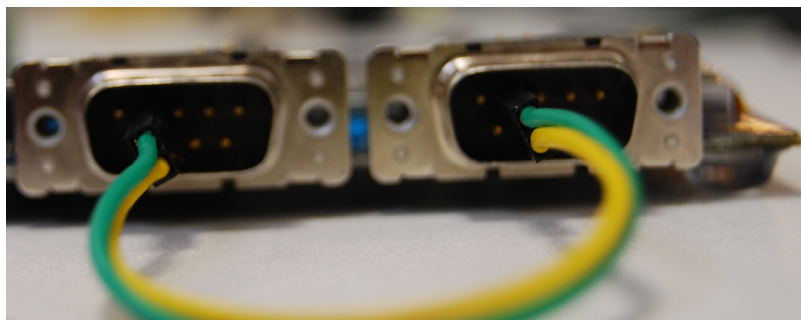
**Figure 2-2** CAN connector DB9 Pin-out



The preloaded firmware requires to do a loopback between CAN Node 1 and CAN Node 2. That means that one of the 2-pins cable delivered in the kit should be used to connect:

CAN Node 1 pin 2 ←-----→ CAN Node 2 pin 2  
 CAN Node 1 pin 7 ←-----→ CAN Node 2 pin 7

**Figures 2-3** CAN Loopback on the AT32UC3C-EK



## 2.3 Connecting the LIN interface

The AT32UC3C-EK offers two LIN interfaces to connect the board to external LIN network:

1. 3-pins header connector LIN Node 1
2. 3-pins header connector LIN Node 2

It follows the pin-out:

**Table 2-2** LIN connector 3-pins header

Pin #	Signal Names	Signal Description
1	GND	Ground
2	LIN	LIN Bus
3	VBAT	Power Supply

The preloaded firmware requires to do a loopback between LIN Node 1 and LIN Node 2. That means that one of the 2-pins cables delivered in the kit should be used to connect:

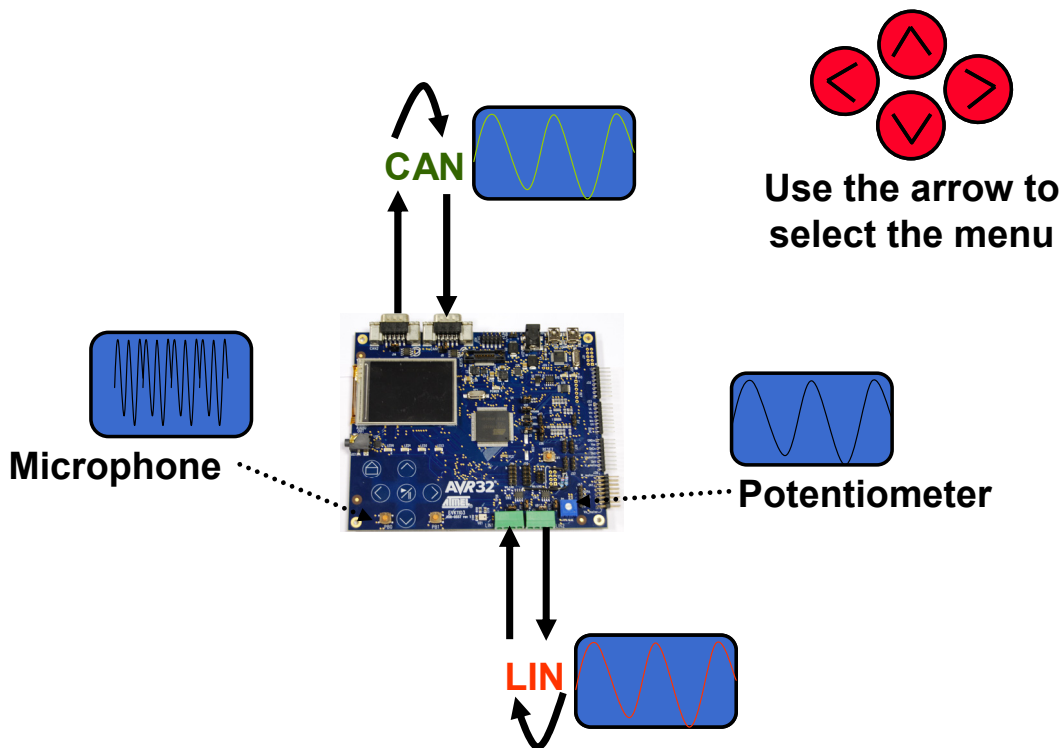
LIN Node 1 pin 1 ←-----→ LIN Node 2 pin 1  
 LIN Node 1 pin 2 ←-----→ LIN Node 2 pin 2



## 2.4 Using the Preloaded Firmware

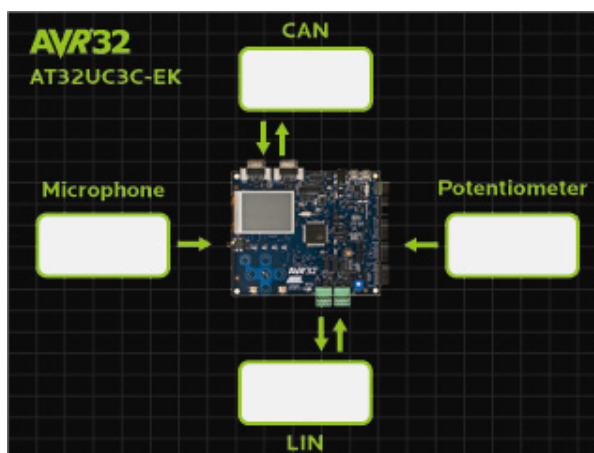
The preloaded application is named “AT32UC3C\_EK\_CAN\_LIN\_LOOPBACKS\_DEMO” located under *APPLICATIONS/AT32UC3C\_EK\_CAN\_LIN\_LOOPBACKS\_DEMO*. The source code of the application can be found in the AVR UC3 Software Framework (see 4.3).

**Figure 2-4** Default application running



This application gathers potentiometer data, displays it and sends it to the CAN and LIN loopback interfaces. The graphical user interface (GUI) also displays CAN and LIN network and microphone activity.

**Figure 2-5** Graphical User Interface



The GUI is composed of four items:

- Potentiometer and microphone

The potentiometer and the microphone are connected to ADC channels. The potentiometer and microphone values are respectively displayed in the potentiometer and microphone boxes.

- CAN

The potentiometer value is sent on the CAN bus through CAN Node 1 and is received by CAN Node 2. The message content is displayed in CAN box.

- LIN

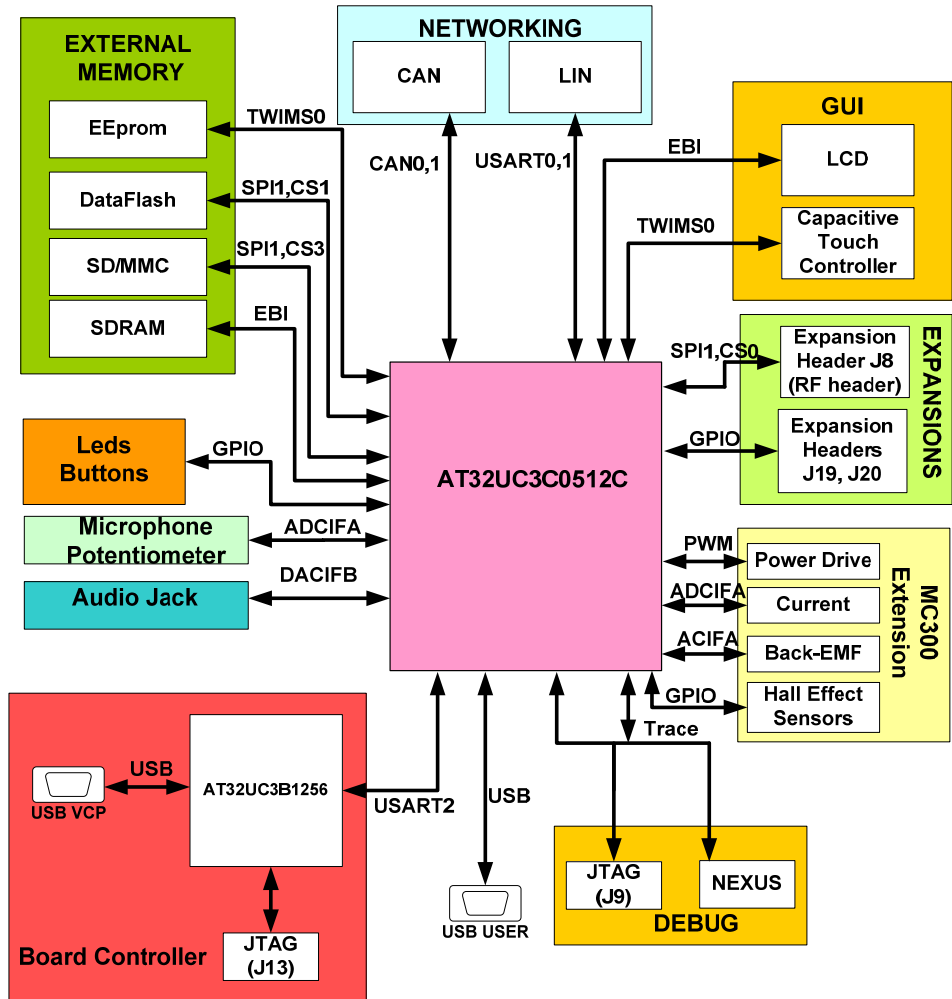
The LIN message built from CAN Node 2 is sent on the LIN bus through LIN Node 1 and received by LIN Node 2. The content of this LIN message received by LIN Node 2 is displayed in LIN box.

Touching:

- the RIGHT (>) pad highlights the potentiometer item,
- the LEFT (<) pad highlights the microphone item ,
- the UP (^) pad highlights the CAN item,
- the DOWN (v) pad highlights the LIN item,
- the PLAY/PAUSE (>/||) focus the item highlighted.

### 3 AT32UC3C-EK Block Diagram

Figure 6-1 AT32UC3C-EK Block Diagram





## 4 References and Further Information

What to do next? What is the next step?

If you want more detailed information concerning the hardware of this kit, if you want to check out all existing application notes and software running on this kit, if you want to start a development whether for evaluation or rapid prototyping, the following sections lists the main available resources to actually get beyond getting started:

- Schematics and BOM,
- Application notes,
- Software Framework,
- Tools,
- Support.

### 4.1 Detailed Hardware References (and Associated Errata)

More detailed hardware information for this kit can be found in two places:

- in 32-bit AVR Studio®: contains the schematics, a logical block diagram and the errata,
- the file AT32UC3C-EK\_Hardware-References.zip available on the Atmel web page dedicated to this kit: [www.atmel.com/](http://www.atmel.com/)

The AT32UC3C family of devices is specified in the UC3C Series datasheet. Always use this document as a reference throughout the development life cycle of an application destined to run on a UC3C device.

The AVR32 architecture manual can be found here:

[http://www.atmel.com/dyn/resources/prod\\_documents/doc32000.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc32000.pdf)

The AVR32 UC technical reference guide can be found here:

[http://www.atmel.com/dyn/resources/prod\\_documents/doc32002.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc32002.pdf)

### 4.2 Application Notes Running on this Kit

To investigate further on the AT32UC3C and/or the AT32UC3C-EK, several application notes are supported by this kit. These can be found on the ATMEL web page dedicated to this kit: [www.atmel.com/](http://www.atmel.com/). All the 32-bit AVR application notes can be found on this Atmel web page: [http://www.atmel.com/dyn/products/app\\_notes.asp?family\\_id=682](http://www.atmel.com/dyn/products/app_notes.asp?family_id=682)

Note that some application notes are bundled with a stand-alone software package. To be able to build the software example and program the binary and step through the code in debug mode, tools are necessary: references to the tools are mentioned in all application notes but are also given here below.

### 4.3 Software Framework

To start a development whether for evaluation or rapid prototyping, the AVR32 Software Framework is a complete and useful source of existing source code a





developer can start from. It has been designed to help develop and glue together the different components of a software design, and to be easily integrable into an operating system (OS) as well as to operate in a stand-alone way.

The 32-bit AVR Software Framework is available in two formats:

- As a stand-alone zip package containing IAR™ projects and GNU Makefile projects, downloadable from this Atmel web page: [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=4192](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4192). This web page also contains all Software Framework-related application notes.
- As a plug-in in AVR32 Studio that allows the user to create pre-defined example AVR32 Studio projects and to add pre-written drivers to existing 32-bit AVR Studio® projects: [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=4116](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116). This web page also contains all AVR32 Studio-related application notes.

To be able to build the software example and program the binary and step through the code in debug mode, other tools are necessary: references to these tools are provided here below.

## 4.4 Tools

To be able to develop applications for 32-bit AVR devices and build binaries for AVR32 targets and program an 32-bit AVR device, Atmel and its partners provide several tools supported on multiple host targets.

### 4.4.1 IDE and Compilers: Design Software

- 32-bit AVR Studio: 32-bit AVR Studio is a free Integrated Development Environment (IDE) for AVR32 that enables you to write, build, deploy and debug your C/C++ and assembler code. It also includes the AVR32 Software Framework plug-in. [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=4116](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116). This web page also contains all AVR32 Studio-related application notes.
- 32-bit AVR GNU Toolchain: AVR32 GNU Toolchain is a set of standalone command line programs used to create applications for AVR32 microcontrollers. [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=4118](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4118).
- IAR Embedded Workbench®: IAR Embedded Workbench with its optimizing C and C++ compiler provides full support and generates very compact and efficient code for 32-bit AVR devices of the AP7000 and UC3 families. <http://www.iar.se/website1/1.0.1.0/124/1/>

### 4.4.2 Programming and On-Chip Debugging Tools : Debug Tools

- AVR JTAGICE mkII: The AVR JTAGICE mkII from Atmel® is a powerful development tool for On-chip Debugging of all AVR 8-bit RISC MCUs and AVR32 32-bit DSP/MCUs with IEEE 1149.1 compliant JTAG interface. [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=3353](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3353)
- AVR ONE! : The AVR ONE! is a powerful development tool for on-chip debugging and programming of all AVR32 and AVR XMEGA™ devices. [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=4279](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4279)

- AVR Dragon: The AVR Dragon sets a new standard for low cost development tools. AVR Dragon supports all programming modes for the AVR device family. [http://www.atmel.com/dyn/Products/tools\\_card.asp?tool\\_id=3891](http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=3891)

## 4.5 Support

Atmel has several support channels available:

- Web portal: <http://support.atmel.no/> All Atmel microcontrollers
- Email: [avr@atmel.com](mailto:avr@atmel.com) All AVR products
- Email: [avr32@atmel.com](mailto:avr32@atmel.com) All AVR32 products

Please register on the web portal to gain access to the following services:

- Access to a rich FAQ database
- Easy submission of technical support requests
- History of all your past support requests
- Register to receive Atmel microcontrollers' newsletters





## 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](http://www.atmel.com)

**Technical Support**  
[avr32@atmel.com](mailto:avr32@atmel.com)

**Sales Contact**  
[www.atmel.com/contacts](http://www.atmel.com/contacts)

**Literature Request**  
[www.atmel.com/literature](http://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.

© 2010 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, AVR®, AVR® logo, AVR Studio®, DataFlash® and others, are the registered trademarks, XMEGA™ and others are trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others..