Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

AVR32908: EVK1104 Getting Started Guide

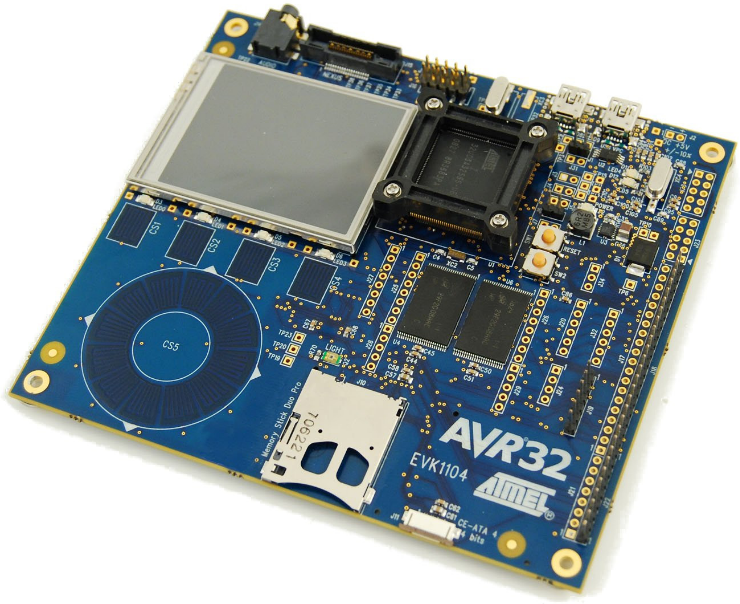
Features

- Powering up the board
- Playing with the DSP application
- Demonstrating the AVR® UC3 DSP capabilities

1 Introduction

The EVK1104 is a reference design and development system for the AT32UC3A3256 microcontroller from Atmel® Corporation. The kit is equipped with a rich set of memories and peripherals that make the EVK1104 a perfect evaluation platform. This guide shows the user how to get quickly started with this kit.

Figure 1-1 EVK1104 board



32-bit AVR® Microcontrollers

EVK1104 Getting Started Guide

Rev. 32138A-AVR32-02/10

2 Powering up the board

The EVK1104 board can be powered from 3 different sources, external power supply, “USB VCP” and “USB USER” connector. If two power sources are applied at the same time the following priorities are used to select the current valid power source:

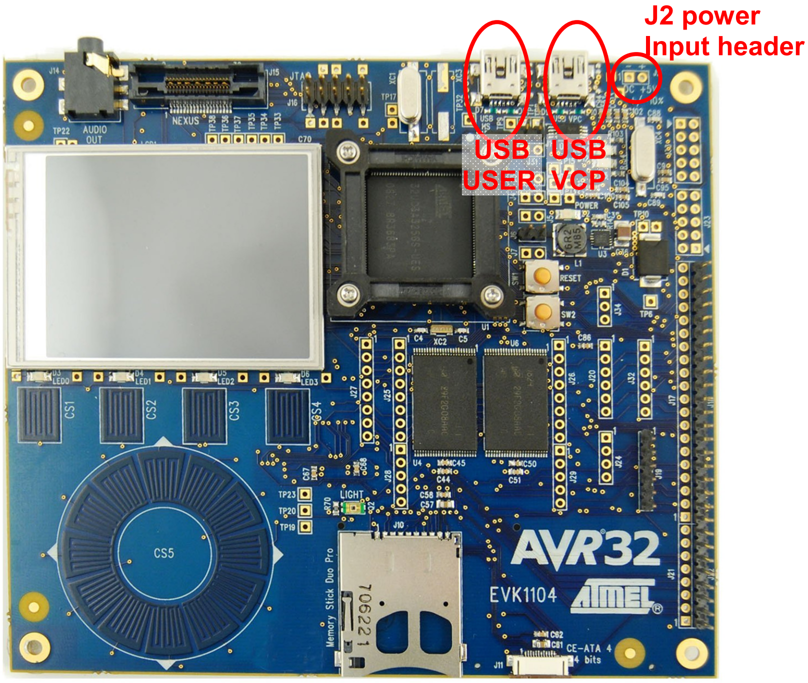
1. External power supply (header J2)
2. USB VCP
3. USB USER

That means that if an external power supply is connected to the board, 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.

On the PCB revision 1, the external power supply must be 5.5V. In order to be able to supply external USB devices that are connected to the board the minimal power supply specification should be 5.5V, 1A. This ensures that the board will work properly if a USB device is connected that needs the maximum allowed 500mA as specified in the USB specification. It is not possible to draw that amount of current from the host through a normal USB cable but there are cables available that can be connected to two hosts (such a cable is not a kit content).

When playing music from a mass-storage device, it is enough to power the board from the USB VCP 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.

Figure 2-1 Possible power input interfaces on the EVK1104



More information about the different power inputs is available in the hardware user's guide.
When the board is powered the green LED, marked with "POWER", will light up and the default board firmware will start to run.

3 Running the DSP Application

3.1 The AVR UC3 DSP Library

The AVR UC3 family of microcontrollers offers better DSP performance than any other 32-bit microcontroller. With its fixed-point DSP instruction set, dual port SRAM, peripheral DMA controller, and multi-level databus, the AVR UC3 offers superior signal processing capabilities combined with industry leading power consumption and fast communication speeds.

Good software is necessary to unlock the superior AVR UC3 DSP performance. So Atmel is providing the AVR UC3 DSP library, a library of basic DSP functions optimized for performance and power consumption by Atmel's DSP experts and thoroughly tested. The library consists of over 70 popular DSP functions in 7 categories. These basic functions build the basis on which a wide range of signal processing algorithms can be implemented with minimum effort. The library is accompanied with a rich selection of examples and application notes.

AVR UC3 DSP Library is part of the AVR UC3 Software Framework, Atmel's free of charge collection of software examples, drivers and libraries for the AVR UC3 microcontrollers. It works on all AVR UC3 MCUs, and is known for its flexibility, robustness and ease of use.

3.1.1 Fixed point numbers

The real world isn't easy to describe with integers, because their limited dynamic range puts heavy restrictions on which numbers we can store and manipulate. That is why digital signal processing makes use of fixed-point numbers, which allows numbers with a decimal point '.' to be stored and used in arithmetic operations. Fixed-point arithmetic support is a requirement for fast and simple digital signal processing. AVR UC3 is the only 32-bit MCU with support for fixed-point numbers, and the AVR UC3 DSP library makes full use of these instructions to achieve the superior DSP performance.

3.1.2 AVR UC3 DSP Library Categories

AVR UC3 DSP Library organizes functions in 7 categories.

- **Filtering**
  - **FIR, IIR, Interpolation, Least Mean Square**
- **Operators**
  - **Multiply, Divide, Sine, Square Root, Logarithm, Power, Random**
- **Signal Generation**
  - **Sine, Square, Saw tooth, Dirac comb, Ramp, Step, Noise**
- **Transforms**
  - **Fast Fourier Transforms (FFT), Inverse FFT**

- **Vectors**
  - **Add, Subtract, Multiply, Divide, Copy, Negate, Convolution**
- **Windowing**
  - **Rectangular, Gauss, Hamming, Hann, Kaiser, Welch, Bartlett, Blackman**
- **Advanced**
  - **ADPCM encoder/decoder, Frequency resampling**

3.1.3 Vectors

Many DSP functions operate on vectors. A vector is a buffer filled with fixed-point numbers known as elements. Each vector element represents one signal sample or filter coefficient. Vectors can have any length, but some DSP functions may require vectors to have 2*n, 4*n, or 6*n elements. Some functions may also require coefficient vectors to be of a fixed length.

All vectors must be stored word (32-bit) aligned in SRAM because it allows the AVR UC3 CPU to fetch multiple elements more efficiently.

3.2 Board Setup

Power the EVK1104 by plugging the USB cable into on the USB connectors.

Connect a headphone or speaker to the jack in order to listen to the music.

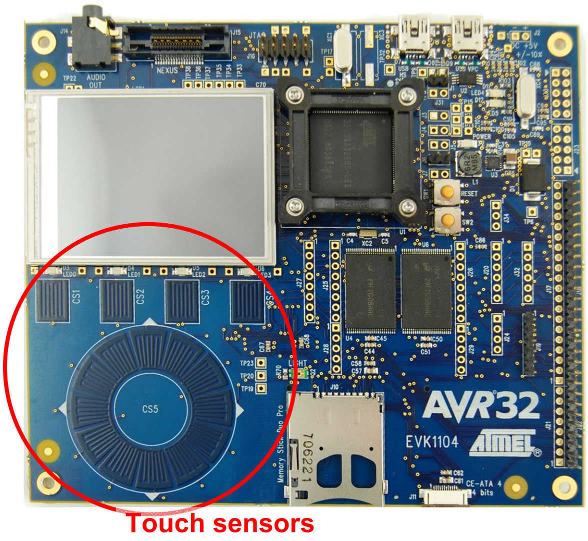
3.3 User Interface and Basic Navigation

The DSP demo is controlled by a graphical user interface (GUI) on the EVK1104 LCD. The touch sensitive buttons and wheel below the screen are used for input. The buttons and the wheel take on different functions depending on the element selected.

When the application is reset, the menu is in idle mode and no function on the screen is selected.

- Turn the wheel and see how you can select different elements on the screen.
- Observe how the instructions in the box at the lower right part of the screen change as you select different elements.
- Push the BACK button to return to the top level menu.

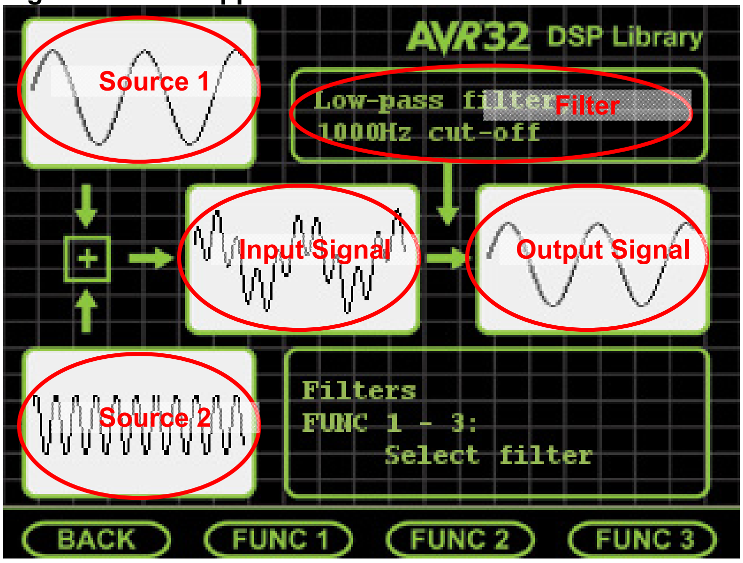
Figure 3-1 Touch sensors for controlling the DSP application



There are five selectable elements on the screen; four signal views and the filter.

- The two signals on the left show the **source signals**. When either of these is selected, the amplitude and frequency of the signal can be changed.
- The signal in the middle is the combined signal of the two sources; this is the **combined input signal** to the filter.
- The **filter selection box** controls which filter is applied to the input signal. By default, no filter is applied.
- The signal on the right is the **output of the filter**. This is the signal output we hear in the speaker.

Figure 3-2 DSP Application Menu



Low-pass filter
1000Hz cut-off

Filters
FUNC 1 - 3:
Select filter

All signals have a zoom function, enabling you to get a better view of the signal.

Zooming:

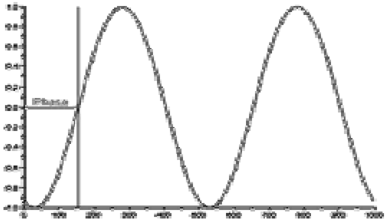
- Select the combined input signal at the middle of the screen.
- Press and hold the FUNC3 button to zoom on the graph.

Reset application:

- Press and hold the BACK button for 3 seconds (or push the RESET button) to reset the source signals and filters and return the application to its initial state.

3.4 Signal generation

To provide a complex input signal to the DSP filters, the application generates two sine signals using the `dsp16_gen_sin()` function found in the Signal Generation category of the DSP library.



This function fills a vector (buffer) with a sinusoidal signal of a specified frequency and sampling rate. This function is used to change the input frequency of our source signals.

Generating different input signals:

- Push and hold the BACK button for 3 seconds to reset the signals and filters.
- Turn the wheel to select Source 1.
- Push FUNC 2 to select frequency setting.
- Turn the wheel and observe how the frequency of the source signal can be modified.

3.5 Basic Filtering

A basic filter function is a function that suppresses (reduces) selected frequencies in a signal. Filters are applied to extract useful information from a signal, either by removing unwanted signals (noise) or extracting useful frequencies that lie within a given frequency range. In audio applications, filters are applied to split the audio signal into bass and treble channels. In RF communication, filters are used to remove the carrier frequency.

A wide range of filters exist, but they are usually categorized into 4 categories:

- **Low pass filters** allow frequencies below the cutoff frequency to pass through, and suppresses frequencies above it.
- **High pass filters** allow frequencies above the cutoff frequency to pass through.
- **Band pass filters** combine a low pass filter and a high pass filter to allow only frequencies in a limited frequency band (range) to pass through.
- **Advanced filters** make use of advanced signal processing theory to extract useful information spread across multiple frequencies.

In this application, we will apply the two basic filters, a high-pass and a low pass filter.

Applying the filters:

- Push and hold the BACK button for 3 seconds to reset the signals and filters.
- Select the filter box and select the high-pass filter by pushing FUNC2.
- Observe how the low frequency component of the output signal is reduced, making the output very similar to signal 2.
- Select the low-pass filter by pushing FUNC3.
- Observe how the high frequency component of the output signal is reduced, making the output very similar to signal 1.

We will now have a closer look at the filters effect on the signal. By adjusting the frequency of the input signals the filters will let more or less of the signal pass through.

With an increase of frequency of a given signal, a low-pass filter will let less of it pass through. By increasing the frequency of signal 2, it will be even less visible in the output signal.

Higher frequency:

- *Select signal source 2*
- *Change the frequency to 5 kHz or more*
- *Observe how the high frequency signal is clearly visible in the combined signal, but not in the output signal*

Lower frequency:

- *Select signal source 2 again*
- *Reduce the frequency to about 1 khz*

At 1 kHz, the remains of signal 2 are clearly visible in the output signal. By comparing this to the known behavior called the frequency response of the filter, we see how our observations match with the theory. At 1 kHz, called the filter’s cutoff frequency, the noise is only reduced by 3 dB, which is equal to 50%.

FUN FACT: If we reduce the frequency of source 2 to approximately 433 Hz, signal 1 and 2 will have almost the same frequency. This will generate a vibrating volume sound effect as signal 2 changes rapidly between amplifying and damping the output signal. Experimenting with the frame rate and frequency of signal 2 will generate a very visible and audible demo of this effect. The frequency of the vibration will be equal to the difference in frequency between the two signals. This is most audible when the difference is 4 Hz.

3.6 Fast Fourier Transform

Fast Fourier transform (FFT) will extract the frequency components from a signal. The theory behind this function is very complicated, but it provides a strikingly simple output that can be very useful when analyzing the behavior of a digital signal processing system.

Analyzing a signal by FFT:

- Push and hold the BACK button for 3 seconds to reset the signals and filters.
- Select the output signal and push FUNC 1 to enable FFT view.
- Select signal source 2.
- Change the frequency to 4 kHz.
- Observe how the signal component seen in the FFT view shifts to the right with increasing frequency.
- Reduce the frequency, and see how the signal component moves to the left.
- Zoom in on the output signal to see more details.

Changing amplitude:

- Select signal source 2.
- Change the amplitude to 0 and observe how the signal component disappears.
- Change the amplitude to 0.9999 and observe how the signal component grows

When the amplitude of signal 2 increases above 0.50, some new components may start to appear. This is caused by signal saturation, an effect we will cover in more detail in the next hands-on training.

The FFT function is self scaling. When the amplitude of signal 2 is higher than that of signal 1, the FFT output is scaled. It may appear as if signal 1 decreases in amplitude as signal 2 starts to define the maximum of the display.

3.7 Analyzing filters in the frequency domain

Viewing the frequency domain:

- Push and hold the BACK button for 3 seconds to reset the signals and filters.
- Select the FFT view for both the combined signal and the output signal

Turning on low-pass filtering:

- Select the filter.
- Enable the low-pass filter.
- Observe how the filter allows the 433 Hz of signal 1 to pass though, while the 2 kHz of signal 2 is dampened.

High-pass filtering:

- Select the filter and enable the high-pass filter.
- Observe how this gives the opposite effect of the low-pass filter.

Changing frequency:

- Set the amplitude of both signal sources to a value in the 0.40 – 0.45 range.
- In the FFT view of the combined signal, the two components will have equal power, but in the filter output only the high frequency component is visible.
- Increase the frequency of signal 1.

4 References and further information

What to do next? What is the next step?

The following sections list 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)

4.1.1 More Information

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

- In AVR32 Studio: contains the schematics, a logical block diagram and the errata,
- EVK1104 document materials are available on the Atmel web page dedicated to this kit: www.atmel.com/evk1104/.

The AT32UC3A3 family of devices is specified in the UC3A3 Series datasheet: http://www.atmel.com/dyn/products/product_card.asp?part_id=4544. Always use this document as a reference throughout the development life cycle of an application destined to run on a UC3A3 device.

The AVR32 architecture manual can be found here:

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

4.1.2 FAQ

- The USB Virtual Com Port does not work. Is there a way to fix that? http://support.atmel.no/bin/customer?=&action=viewKbEntry&id=740

4.2 Application Notes Running on this Kit

To investigate further on the AT32UC3A3 and/or the EVK1104, several application notes are supported by this kit. These can be found on the Atmel web page dedicated to this kit: www.atmel.com/evk1104/. 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

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 AVR UC3 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 AVR UC3 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. 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 AVR32 Studio projects: 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 AVR UC3 devices and build binaries for AVR UC3 targets and program an AVR UC3 device, Atmel and its partners provide several tools supported on multiple host targets.

4.4.1 IDE and Compilers: Design Software

- AVR32 Studio: AVR32 Studio is a free Integrated Development Environment (IDE) for 32-bit AVR that enables you to write, build, deploy and debug your C/C++ and assembler code. It also includes the AVR UC3 Software Framework plug-in. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116. This web page also contains all AVR32 Studio-related application notes.
- AVR32 GNU Toolchain: AVR32 GNU Toolchain is a set of standalone command line programs used to create applications for 32-bit AVR® microcontrollers. 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 8-bit AVR and 32-bit AVR with IEEE 1149.1 compliant JTAG interface. 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 8-bit AVR and 32-bit AVR devices. 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

4.5 Support

Atmel has several support channels available:

• Web portal:	http://support.atmel.no/	All Atmel microcontrollers
• Email:	avr@atmel.com	All 8-bit AVR products
• Email:	avr32@atmel.com	All 32-bit AVR 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

Technical Support
avr32@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.