

---

**Atmel AT01180: Barcode and QR code scanner User Guide**

---

**Atmel 32-bit Microcontroller****Features**

---

- Image processing and decoder for barcode & QR code
- Libdecodeqr and OpenCV based
- IAR Embedded Workbench®
- Support continuous scanning mode
- SAM4S-WPIR\_RD Kit based

**Introduction**

---

The demo software of Atmel® Barcode and QR code scanner is created to showcase a barcode and QR code example solution based on Atmel SAM4S\_WPIR\_RD kit.

Low cost barcode or QR code scanner are using CMOS sensor. The type of application targeted is wide: coffee machine (read the coffee cups bar code or QR code), smart card reader (secure banking exchange by generating a code based on QR code displayed on PC screen), etc.

The demo acts as a barcode and QR code scanner which recognizes valid barcode and QR image, analyzes and decodes image. The demo is developed based on OpenCV-1.0.0 for image processing and Libdecodeqr-0.9.3.

This documentation describes the preprogrammed demo software of Atmel barcode and QR code scanner.

## Table of Contents

1. Reference .....	3
2. Overview .....	4
2.1 Description .....	4
2.2 Hardware resources .....	4
2.3 Hardware files .....	5
3. Barcode .....	6
3.1 Barcode overview .....	6
3.2 How to decode an EAN-13 barcode .....	6
4. QR code .....	8
4.1 QR code overview .....	8
4.2 How to decode .....	8
5. Inside the demo application .....	9
5.1 Overview .....	9
5.2 MCU initialization .....	9
5.2.1 Clock for maximum frequency .....	10
5.3 CMOS Imaging Sensor .....	10
5.3.1 Initialization of PIO Parallel Capture and OV7740 .....	10
5.3.2 Parallel Capture Event Handler .....	10
5.4 OpenCV Library .....	10
5.4.1 Smoothing Images .....	10
5.4.2 Threshold Operations .....	11
5.4.3 Find all contours .....	11
5.5 Libdecodeqr Library .....	11
5.6 BAR Code decoding .....	11
5.7 QR Code decoding .....	12
6. How to use .....	13
6.1 Downloading the Binary .....	13
6.2 Running the application .....	14
7. Revision History .....	15

## 1. Reference

QR code

[http://en.wikipedia.org/wiki/QR\\_Code](http://en.wikipedia.org/wiki/QR_Code).

BAR code

<http://www.gs1us.org/>

Open CV

<http://opencv.org/>

Libdecodeqr

<http://trac.koka-in.org/libdecodeqr>

SAM4S Full datasheet

[http://www.Atmel.com/products/at91/sam4s.asp?family\\_id=605](http://www.Atmel.com/products/at91/sam4s.asp?family_id=605)

More detailed hardware information for this kit can be found and available on the Atmel web page dedicated to this SAM4S\_WPIR\_RD kit: [www.atmel.com/](http://www.atmel.com/).

## 2. Overview

### 2.1 Description

This demo application is developed based on Atmel SAM4S\_WPIR\_RD kit with a demo application, and is intended to provide some guidance to use Atmel SAM4S for QR decode application. Like other examples for SAM4S\_WPIR\_RD, this demo is provided with IAR project and with a similar structure. The OpenCV and Libdecodeqr are provided in a library way.

At power up, demo shows initialization screens. After a while, the application starts with continuous CMOS sensors acquisition task. User is able to put an example QR image like the one attached in this folder in front of CMOS sensor.

After successful detection, the program enters a decode processing and then displays the result and information.

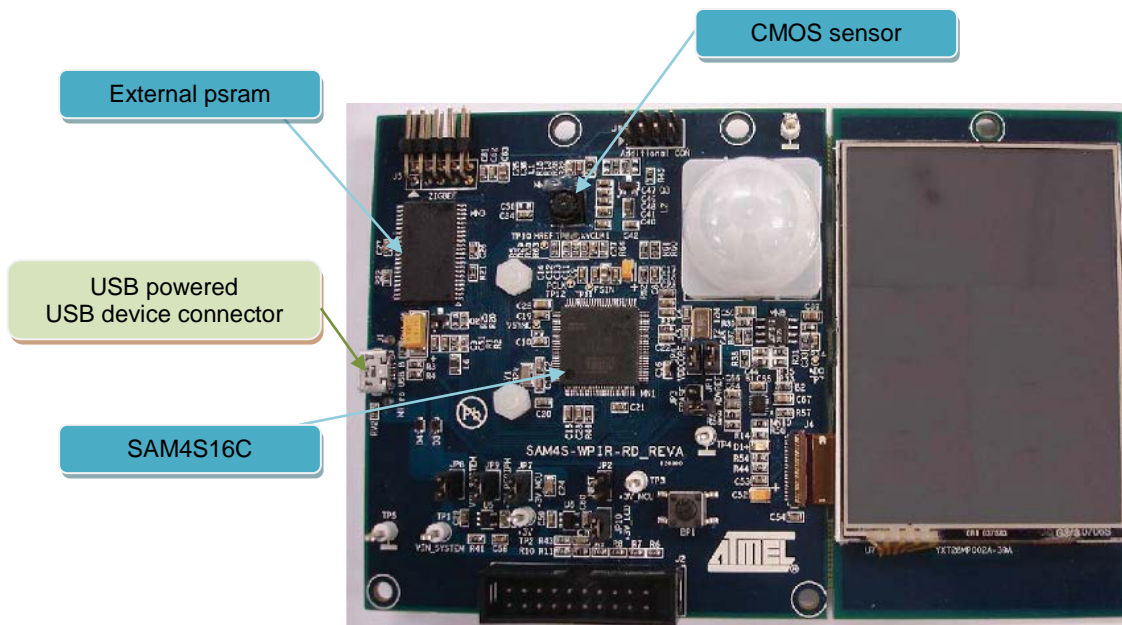
The software includes following tasks:

1. CMOS Sensors image capture task, managing the acquisition of image data.
2. Image processing.
3. QR decoding.
4. LCD display

### 2.2 Hardware resources

The Atmel SAM4S\_WPIR\_RD kit is powered through the USB connector. There is no extra controls needed in the demo and after successful decoding the result will be displayed onto the LCD or error code instead.

Figure 2-9. Hardware resources



## 2.3 Hardware files

Board design files are provided with the reference design. See the “Hardware” folder. Board design and board manufacturing files are provided:

Board Design files:

- SAM4S-WPIR-RD\_RevA.pdf: Schematics, PDF format
- SAM4S- WPIR-RD \_REVA.DSN: Schematics, Cadence® OrCAD® Capture format
- SAM4S- WPIR-RD \_REVA.brd: PCB project, Allegro PCB Design
- SAM4S- WPIR-RD \_REVA.xls: Bill of Materials

Board manufacturing files:

- GERBER files format.

## 3. Barcode

### 3.1 Barcode overview

Barcode is a series of lines with varying width and spacing which contains some information. Although 2D systems using a variety of symbols are generally referred to as barcodes as well, basically the common barcode is parallel and linear lines. Nowadays, barcodes that use EAN/UPC symbol (including the UPC-A, UPC-E, EAN-13 and EAN-8 barcodes) are those allowed for products scanned at retail point of sale. EAN/UPC barcodes ensure that all products are properly identified at any retail point of sale, speed up data collection, and result in more accurate data that can be used by trading partners.

Figure 3-1. Bar code image examples



In this Demo project, EAN-13 barcode is selected to be supported as an example.

An EAN-13 barcode (stands for European Article Number) is a 13 digit (12 data and 1 check) bar coding standard, which is defined by the standards organization GS1. The EAN-13 barcodes are used worldwide for marking products often sold at retail point of sale. The numbers encoded in EAN-13 bar codes are product identification numbers, which are also called Japanese Article Number (JAN) in Japan.

### 3.2 How to decode an EAN-13 barcode

There are several function modules, from left to right followed by left margin, start module, left data digits, center divider, right data digits, check digit, ending module and right margin.

Left guide bar (3) + 6 digits data characters (7\*6) + Center bar (5) + 5 digits data characters (7\*5) + module check character (7) + Right guide bar (3) = 3 + 42 + 5 + 35 + 7 + 3 = 95

Those modules can be divided into following white/black bars.

Left margin: white \* 1 = 1

Left guide bar: black \* 2 + white = 3

Left Data characters: (black \* 2 + white \* 2)\*6 = 24

Center divider: black \* 3 + white \* 2 = 5

Right Data characters: (black \* 2 + white \* 2)\*6 = 24

Right guide bar:  $\text{black} * 2 + \text{white} = 3$

Right margin:  $\text{white} * 1 = 1$

$1 + 3 + 24 + 5 + 24 + 3 + 1 = 61$

In another word, there are 61 bars in total.

#### Encoding of the digits

Digit	L-code	G-code	R-code
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

## 4. QR code

### 4.1 QR code overview



QR code (abbreviated from Quick Response Code) is one of matrix barcode (or two-dimensional bar code). Recently, the QR Code system has become more and more popular due to its fast readability and greater storage capacity compared to standard UPC or EAN barcodes. The QR code consists of black modules arranged in a square grid on a white background. The information encoded may be made up of four standardized types ("modes") of data (numeric, alphanumeric, byte / binary, Kanji), or through supported extensions.

The left graphic is the QR code for the URL of ATMEL's official webpage:

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

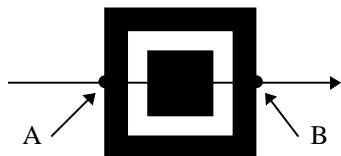
There are several standards in documents covering the physical encoding of QR codes: JIS X 0510 and ISO/IEC18004

### 4.2 How to decode

Generally, QR decoding algorithm is to find meaningful symbols inside the QR image by recognizing black or white square modules. It is similar to barcode. Here is an example on decoding procedure:

1. Calculate the threshold value and convert the image into a binary image
2. Locate the finder patterns. Finder pattern consists of three same patterns located on the corner. The module sequence is fixed to 1 black|1 white|3 black|1 white|1 black.

Figure 4-9. Finder Pattern



3. Analyze the location of finder pattern to identify the rotation.
4. Analyze the location of finder pattern to identify the distance.
5. Calculate the size of module
6. Get the version of QR image
7. Generate the bit map from QR image
8. Read out the correction class and mask pattern
9. Apply XOR operation with mask pattern
10. Analyze the symbol according to the rules in the standard
11. Obtain the bit stream and apply the error correction program
12. Reorganize the bit stream and decode by rules.



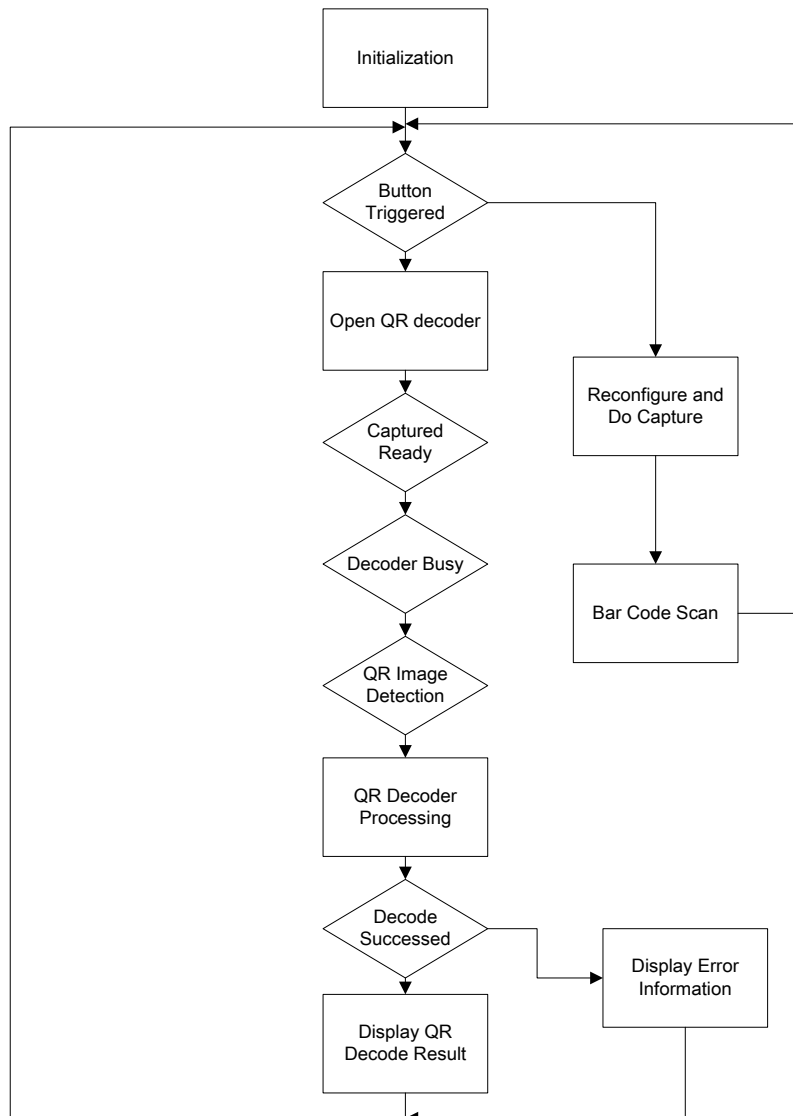
## 5. Inside the demo application

### 5.1 Overview

Basically, similar to other applications, the initialization including clock and peripherals configuration is an essential part and for this demo application CMOS sensor plays as the most important role. The quality of captured images is up to different configurations on CMOS sensor, which is also a critical factor for barcode or QR code decoding.

The functions of demo application could be divided into two parts. One is barcode scanner mode and the other is QR code scanner mode. The mode could be switched by pressing the button BP1.

**Figure 5-1. Flow Chart of Demo project**



### 5.2 MCU initialization

For demos, it is necessary to perform certain initialization tasks to meet various requirements, such as clock configuration, and initialization of some application-specific components.

### 5.2.1 Clock for maximum frequency

The main clock source is the internal fast RC Oscillator. By default, the main clock source is the internal fast RC Oscillator at 4MHz. To work at the maximum frequency, main clock source should be selected as the external crystal at 12 MHz and PLLB as MCK source. To configure microcontroller with these setting, enable the correct define in the Board\_lowlevel.c. In this case, clock is set at 120 MHz

## 5.3 CMOS Imaging Sensor

The SAM4S device is capable of capturing parallel data in synchronization with an external clock, which is used for interfacing to a CMOS image sensor.

### 5.3.1 Initialization of PIO Parallel Capture and OV7740

An 8-bit parallel capture mode is available inside the SAM4S PIO controller. It is used to interface a CMOS digital image sensor. OV7740 is a CMOS image sensor with maximum VGA output, RAW RGB and YUV format, I2C compatible serial interface. For our application, PIO capture mode is used for the clock and data interface and TWI (I2C compatible) for serial controlling interface. For the detailed description of the PIO parallel capture function and settings, refer to the Parallel Input/output Controller section in the SAM4S series datasheet.

### 5.3.2 Parallel Capture Event Handler

The Auto White Balance (AWB) and Auto Exposure (AE) settings of the OV7740 image sensor are set to auto mode by default, so the image sensor gets good quality of images under different environments. But this increases the tuning time and needs frames before it can get good image quality. In the software, because the image sensor is put into low-power mode and wakes up from time to time, if the AWB and AE are set to auto mode, the first frame's quality after wake up will not be perfect. In order to improve the image quality, the following method is used to improve the image quality. After power up, the AWB and AE of the image sensor are set to auto mode. Before the first capture, the software will wait three seconds for AWB and AE tuning, then the parameters of the AWB and AE settings will be saved in the SAM4S backup registers, and AWB and AE will be set to manual mode. Afterwards, each time before capture, the AWB and AE will be configured using these saved (good) parameters from the backup register. Thus after the wake-up, and thanks to the saved parameters, the quality of the first image is improved. Note that this method will make the first capture after power up longer, because there is a three-second waiting time for the good AWB and AE parameters to settle. Also, this method only improves the image quality if the environment does not change greatly compared with the time that AWB and AE perform auto tuning.

## 5.4 OpenCV Library

OpenCV(Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. It is released under a BSD license and it is free for both academic and commercial use. The library has more than 2500 optimized algorithms. Thanks to OpenCV, SAM4S\_WPIR\_RD is able to realize some important image processing function such as threshold, smooth, contour finding and even perspective transform.

### 5.4.1 Smoothing Images

In order to reduce noise, the program adopts several smoothing functions including median filter and Gaussian filter. More details on image smoothing, the documents on OpenCV could be referred to.

### 5.4.2 Threshold Operations

In order to separate out regions of an image corresponding to QR image which we want to analyze and meantime also obtain a proper gray image for next image processing, the function `cvAdaptiveThreshold` is to threshold with mean and Gaussian methods. Also, the OTSU method could be selected as the threshold algorithm by setting `adaptive_th_size` as a negative value.

### 5.4.3 Find all contours

Thanks to `cvFindContours` function, the program is able to find contours in a binary image. As for contour approximation method, the `CV_CHAIN_APPROX_NONE` is applied which stores absolutely all the contour points. That is, any 2 subsequent points  $(x1,y1)$  and  $(x2,y2)$  of the contour will be either horizontal, vertical or diagonal neighbors, that is,  $\max(\text{abs}(x1-x2), \text{abs}(y2-y1))=1$ . For contour retrieval mode, `CV_RETR_LIST` is used to retrieve all of the contours without establishing any hierarchical relationships.

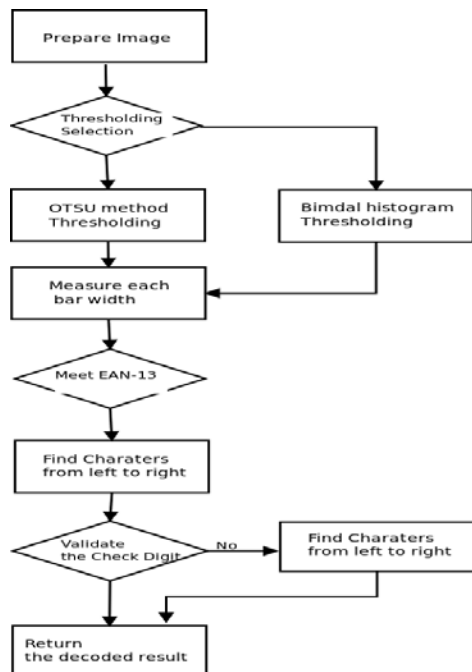
## 5.5 Libdecodeqr Library

"libdecodeqr" is a C/C++ library for decoding QR code based on JIS X 0510 and ISO/IEC18004. The main features of libdecodeqr are completely free and high speed decoding.

## 5.6 BAR Code decoding

In the demo, there are two kinds of methods to convert a captured image into a black and white image. Consequently, the algorithms are different. One is the function `bin_gray_img_ohno()` which is to determine the binarizing threshold directly from bimodal histogram of the image, the other is the function `bin_gray_img_otsu()` which adopts famous OTSU algorithms. After a black and white image is obtained, the bar width is measured and then the number of bars would be checked whether it meets the standard of EAN-13. If so, the program would decode data scanner from left to right and then do another scan from right to left.

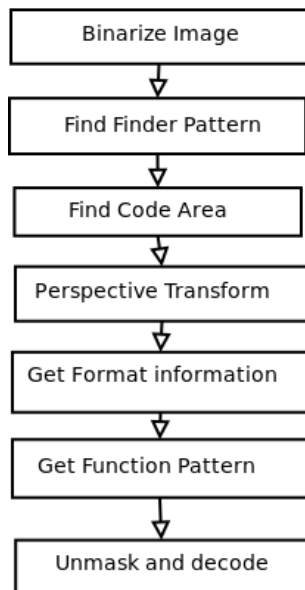
Figure 5-2. Flow chart of Bar code decoding



## 5.7 QR Code decoding

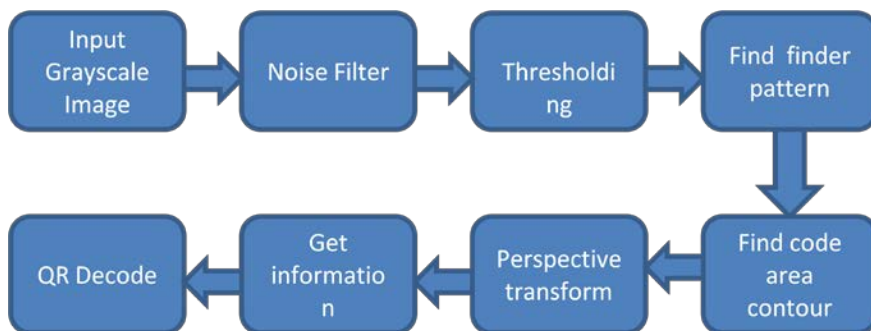
In the demo, the QR code decoding relies on the open source library OpenCV and Libdecodeqr. Supported by image reader of Libdecodeqr, the program is able to complete the image processing task and also for QR code decoding task. Refer to the below flow charts.

**Figure 5-3. Flow chart of QR code decoding**



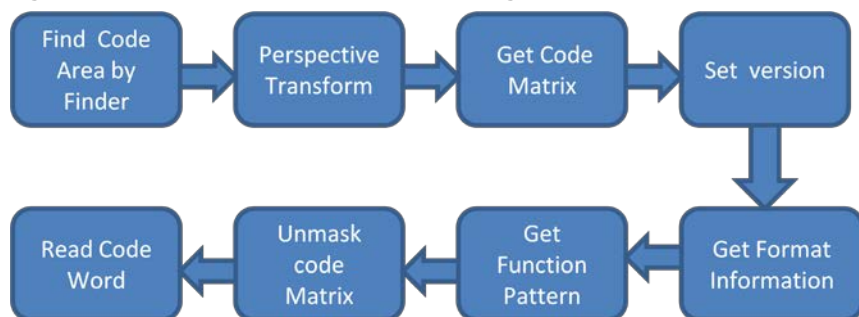
As for the image processing, generally the raw image is needed to be filtered for reducing noise, thresholded to be a black and white one. After using find\_contour() methods, some information contained in the image could be found. Then, the image is required to be transformed according to some acquired information. Finally, a valid bit stream is able to be generated for QR decode.

**Figure 5-4. Flow chart of QR image processing**



As for the QR code decoding part, we follow the generic method to decode the QR image. First of all, the program gets code matrix or QR image and then sets QR version according the width of finder pattern and image. During the first-stage QR decoding, the format information and function pattern could be obtained after analysis. By utilizing the function pattern masked code matrix, the final bit stream could be ready for use.

**Figure 5-5. Flow chart of QR code decoding**



## 6. How to use

The default application programmed on the board is the periodic motion detection demonstration. After power up, the screen will display information about demo, moving in front of the board causes a picture to be taken which is displayed on the LCD. You have to program the barcode and QR code scanner application into the board.

If the demo is programmed into the board, the board acts as a video camera. When a valid barcode or QR image is captured, the information will be displayed after successful decoding. The different modes for barcode scanner and QR code scanner could be switched freely by pressing the button BP1.

### 6.1 Downloading the Binary

All provided binaries were compiled with IAR tool chain and with flash release configuration.

The binary file is as following:

| - QR Scanner\_With\_Ext\_SRAM.bin

The binary files can be downloaded directly into the Flash memory using SAM-BA® v2.11 either with Atmel SAM-ICE or via USB port (Virtual serial port on device manager). SAM-BA v2.11 CDC version must be used for USB port.

- Start SAM-BA
- Choose the connection and the SAM4S16-EK board, and click Connect.
- On the next window shown below, from the Scripts drop-down list choose:
  - Refresh the start Address
  - Enable Flash access script (Scripts drop-down list), click “Execute”
  - Then Boot from Flash (GPNVM1) script, click “Execute”
  - Then click the icon folder close to the “Send File” button, and browse to the desired binary file. Click “Send File” button
  - When asked “Do you want to lock the involved region...” choose No
  - Close SAM-BA and power down then power up the board. The new application starts.

## 6.2 Running the application

After programming the barcode and QR code scanner application, some remind information would show on the LCD screen and then the board enters into the default barcode scanner mode. In barcode mode, there is only one thirds displaying zone available and one small colored square indicator would be found on the top and left. Once the valid barcode image is captured, the decoded information will be displayed. QR code scan mode is very similar to barcode mode, after pressing the button the screen will be displayed will full screen available and the board acts like a camera. Once the valid QR image is captured, the information will be displayed.

## 7. Revision History

Doc. Rev.	Date	Comments
42098A	03/2013	Initial document release

**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

© 2013 Atmel Corporation. All rights reserved. / Rev.: 42098A-ATSAM-03/2013

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. 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.