
**Quick Start Guide for ATWILC3000
on SAMA5D3-Xplained Linux platform**

AN-XXXX

Prerequisites

- **Hardware Prerequisites**
 - SAMA5D3 Xplained board
 - Atmel WILC3000 evaluation board
 - Micro USB Cable (TypeA / MicroB)
 - USB to Serial Adaptor (for DBGU port)
- **Build Prerequisites**
 - Linux Host PC
 - Linux buildroot-at91

Introduction

The WILC3000 is a single chip IEEE 802.11 b/g/n RF, baseband, MAC, Bluetooth 4.0 and FM receiver optimized for low-power mobile applications. The WILC3000 utilizes highly optimized 802.11 – Bluetooth coexistence protocols. It provides multiple peripheral interfaces including UART, SPI, I2C and SDIO.

This quick start guide describes how to integrate the Atmel ATWILC3000 evaluation board via SDIO for Wi-Fi and USART for Bluetooth in SAMA5D3 Xplained Linux platform. The following links also are available to get more information on Atmel wireless drivers, buildroot-at91 and prebuilt images.

- Atmel Linux For SAM Site: <http://www.at91.com/linux4sam>
- SAMA5D3 Xplained board: <http://www.atmel.com/tools/atsama5d3-xpld.aspx>
- Atmel ATWILC3000 Site: <https://github.com/atwilc3000>

Table of Contents

Prerequisites.....	1
Introduction.....	1
1. ATWILC3000 Linux Software Package.....	3
2. SAMA5D3 Xplained Board Software Package.....	3
2.1 Get sources.....	3
2.2 Build rootfs images	3
3. Flash images.....	3
3.1 Prebuilt images	3
3.2 Engineering images built with source codes.....	4
4. Bring up ATWILC3000	4
4.1 Customizing Buildroot.....	4
4.2 Patch the kernel	5
4.3 Build new rootfs image.....	5
5. Hardware Consideration	6
5.1 Bluetooth.....	6
5.2 WiFi.....	8
6. Running ATWILC3000 for the WLAN.....	9
6.1 Enable the network interface	9
6.2 Scan for networks	9
6.3 Connect to an unprotected network.....	9
6.4 Connect to an WEP encrypted network.....	9
6.5 Connect to an WEP encrypted network with WPA supplicant.....	9
6.6 Connect to WPA2 network with WPA supplicant.....	10
6.7 Assign an IP address	10
6.8 Verify the connection	10
6.9 Disconnect from the network	11
7. Running ATWILC3000 for the Bluetooth	11
7.1 Attach serial device via UART HCI to BlueZ stack	11
7.2 Enable the network interface	11
7.3 Scan for networks	11
7.4 Connect to network	12
8. Conclusion	12
9. Revision history.....	12

1. ATWILC3000 Linux Software Package

The latest WILC3000 is available on the [AtmelSmartConnect](#). Download the latest ATWILC3000 driver and firmware.

```
git clone https://github.com/atwilc3000/driver.git
git clone https://github.com/atwilc3000/firmware.git
```

For more information, visit the GitHub wiki pages: <https://github.com/atwilc3000/driver/wiki>

2. SAMA5D3 Xplained Board Software Package

This chapter describes how to get the Buildroot for SAMA5D3 Xplained board and build the images. The Linux platform for AT91 is maintained at the [Linux4SAM](#). See the following instructions to get the Buildroot-at91 and build images.

2.1 Get sources

```
$ git clone git://github.com/linux4sam/buildroot-at91.git
$ cd buildroot-at91
$ git checkout origin/buildroot-2013.11-at91 -b buildroot-2013.11-at91
```

2.2 Build rootfs images

```
$ make sama5d3_xplained_defconfig
$ make
```

This build gets Linux kernel, for example, the following kernel is available after building is completed.

- o /buildroot-at91/output/build/linux-83a9eb4b2f16d9b388daa473a954fb2a563a7ccb

3. Flash images

This section describes how to flash the image. The Atmel SAM-BA tool is required to flash the images to the target board. Install the SAM-BA if not available. Find more information on the SAM-BA tool from the following link: <http://www.at91.com/linux4sam/bin/view/Linux4SAM/SoftwareTools>

3.1 Prebuilt images

The prebuilt images are provided in the Demo archives section from [Linux4SAM](#). Download the following demo package for SAMA5D3 Xplained board: ftp://www.at91.com/pub/demo/linux4sam_4.3/linux4sam-poky-sama5d3_xplained-4.3.zip.

- Open JP5 to disable NAND flash memory access
- Press BP2 reset button to boot from on-chip Boot ROM
- For Windows users, verify that the USB connection is well established. AT91 USB to Serial converter should appear in Device Manager. Then, launch the **demo_linux_nandflash.bat** file in the demo archive.
- For Linux users, check /dev/ttyACMx by issuing the following command:

```
$ ls /dev/ttyACM*
```

Then, launch the **demo_linux_nandflash.sh** file. If the dev/ttyACMx that appear is different from /dev/ttyACM0, modify the demo_linux_nandflash.sh before launching the script.

- When the logfile.log appears, check that = Done. = is written at the end of file.
- Press BP2 reset button to boot on the NAND Flash memory and play the demo.

Make sure that the system should know where SAM-BA tool is because demo_linux_nandflash.bat or script file runs SAM-BA with proper parameters. For Linux users, check the PATH and add the SAM-BA in the PATH. For more information visit the [Linux4SAM](#).

```
$ echo $PATH
$ export PATH=$PATH:/home/${SAM-BA_location}
```

3.2 Engineering images built with source codes

The following outputs are generated when Buildroot is completely compiled. All of outputs should be downloaded to the target board.

- /buildroot-at91/output/build/linux-xxxxx/arch/arm/boot/zImage
- /buildroot-at91/output/build/linux-xxxxx/arch/arm/boot/dts/at91-sama5d3_xplained.dtb
- /buildroot-at91/output/images/rootfs.ubi

Copy all of them into the demo archive directory where the demo_linux_nandflash.sh or batch file exists. Then, replace the existing files by new outputs. Modify the followings described in the **demo_linux_nandflash.tcl** file in the demo archive directory like the followings and run the batch or script file as done with prebuilt image.

```
set kernelFile "zImage"
set rootfsFile "rootfs.ubi"
```

4. Bring up ATWILC3000

This section describes how to integrate ATWILC3000 in the SAMA5D3 Xplained board. The ATWILC3000 is connected to SAMA5D3 Xplained board via SDIO interface for WLAN, via USART for Bluetooth.

4.1 Customizing Buildroot

Before building the rootfs image, customizing the Buildroot is required to bring up the ATWILC3000. Add easily packages from the rootfs by issuing the following command.

```
$ make menuconfig
```

Then, add the packages described in the following, for example, go to "Target package", "Networking applications" and add bluez-utils with subordinates.

- Target package → Networking applications → bluez-utils
- Target package → Networking applications → dhcp (ISC)
- Target package → Networking applications → dhcpcd
- Target package → Networking applications → hostapd

Make rootfs images by issuing the make command.

```
$ make
```

4.2 Patch the kernel

This section introduces how to make rootfs image including ATWILC3000 kernel module. Find the latest ATWILC3000 driver source by issuing the following command.

```
$ git clone https://github.com/atwilc3000/driver.git
$ git clone https://github.com/atwilc3000/firmware.git
```

The patch files are also available to download from the following link: <https://github.com/atwilc3000/patch>.

The kernel source is available in the **Buildroot-at91/output/build/linux-xxxxx**. Create the **atmel** directory in the **linux-xxxxx/drivers/net/wireless** if not available.

```
$ mkdir atmel
```

Copy ATWILC3000 driver downloaded from the GitHub to the **atmel** directory.

```
$ git clone https://github.com/atwilc3000/driver.git
$ cd driver
$ cp Kconfig Makefile -r wilc3000/ \
  ~/buildroot-at91/output/build/linux-xxxxx/drivers/net/wireless/atmel
```

Then, modify the **Kconfig** and **Makefile** in the directory, **/drivers/net/wireless** to add the ATWILC3000 driver into the Kbuild system.

- /drivers/net/wireless/Kconfig

```
source "drivers/net/wireless/atmel/Kconfig"
```

- /drivers/net/wireless/Makefile

```
obj-$(CONFIG_ATMEL_SMARTCONNECT) += atmel/
```

Customize the kernel by issuing the following command and add ATWILC3000 driver to support Atmel wireless devices.

```
$ make ARCH=arm menuconfig
```

In the menuconfig, go to “**Device driver → Network device support → Wireless LAN → Atmel SmartConnect Wireless cards Driver**” and set the ATWILC3000 driver as module or built-in. Make sure that select the SDIO interface for WLAN. The ATWILC3000 device will be connected to the SAMA5D3 Xplained board via SDIO for WLAN.

Make sure that ATWILC3000 driver depends on the CFG80211 and BT package in the Linux system. Hence, check if the CFG80211 and BT is enabled or not. If disabled, add CFG80211 and BT package in the menuconfig.

For CFG80211, add “**Networking support → Wireless → Cfg80211 – wireless configuration API**”.

For BT, add “**Networking support → Bluetooth subsystem support**” with subordinates. Make sure if the “**Bluetooth subsystem support → Bluetooth device drivers → HCI UART driver**” is selected with subordinates or not.

4.3 Build new rootfs image

This section describes how to make new rootfs image including the ATWILC3000 kernel module after patching the kernel with ATWILC3000 driver. The ATWILC3000 firmware should be placed in the directory, **buildroot-at91/output/target/lib/firmware** before building new rootfs image. Create **firmware** directory in the buildroot-at91/output/target/lib if not available. Then, copy the firmware downloaded

from the GitHub in the firmware directory. To build new rootfs with ATWILC3000 driver in the buildroot source package, remove the following outputs.

- buildroot-at91/output/build/linux-xxxxx/.stamp_built
- buildroot-at91/output/build/linux-xxxxx/.stamp_target_installed
- buildroot-at91/output/build/linux-xxxxx/.stamp_images_installed

Then, build the rootfs by issuing the following command in the buildroot-at91 directory.

```
$ make
```

Check if the **atmel/wilc3000.ko** is generated in the following path:

- /buildroot-at91/output/target/lib/modules/3.10.0/kernel/drivers/net/wireless

5. Hardware Consideration

This section shows how to connect the ATWILC3000 EVB to SAMA5D3 Xplained board for WiFi and Bluetooth. The ATWILC3000 should be connected to SAMA5D3 Xplained via SDIO for WLAN and USART for Bluetooth.

5.1 Bluetooth

The HCI UART transport layer uses the following configurations:

- data length: 8 bits
- parity: no parity
- stop bit: 1stop bit
- flow control: RTS/CTS
- baud rate: vendor specific
- flow-off response time: vendor specific

The local RXD should be connected to the remote TXD and the local RTS should be connected to the remote CTS and vice versa.

For Bluetooth, the USART2 of SAMA5D3 Xplained is connected to the ATWILC3000 EVB. The figure 1 is I/O expansion, J19 on the SAMA5D3 Xplained board. The expansion, J19 has the TXD2 on pin E26, RXD2 on pin E25, RTS2 on pin PE24 and CTS2 on pin PE23.

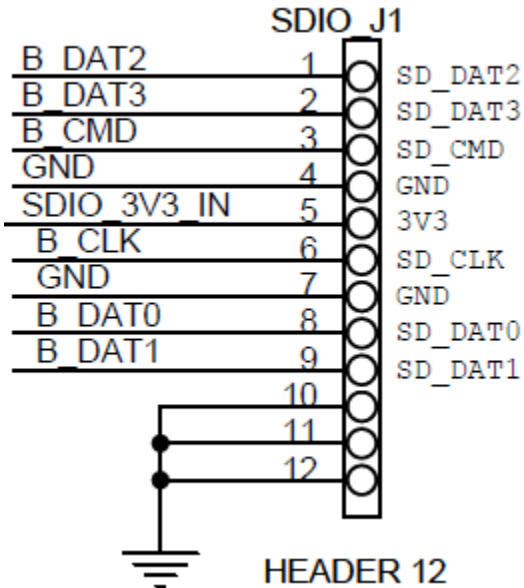
The TXD2 on the SAMA5D3 Xplained should be connected to the RXD on the ATWILC3000 EVB, J216 pin18. In contrast, the RXD2 on the SAMA5D3 Xplained should be connected to the TXD on the ATWILC3000 EVB, J216 pin19. The RTS2 on the SAMA5D3 Xplained should be connected to the CTS on the ATWILC3000 EVB, J216 pin16 and the CTS2 should be connected to the RTS on the ATWILC3000 EVB, J216 pin24. Make sure open J227, J805 and J901 on ATWILC3000 EVB before running the Bluetooth. The figure 2 shows ATWILC3000 EVB J216 connectot.

Diagram of a 26-pin J216 connector. The connector is a vertical strip with 26 pins. Pins 1 through 24 are numbered on both sides. Pins 25 and 26 are at the bottom and are connected to a common ground labeled "HEADER 13X2". The diagram shows the internal wiring of the connector, with lines connecting the pins to the header.

5.2 WiFi

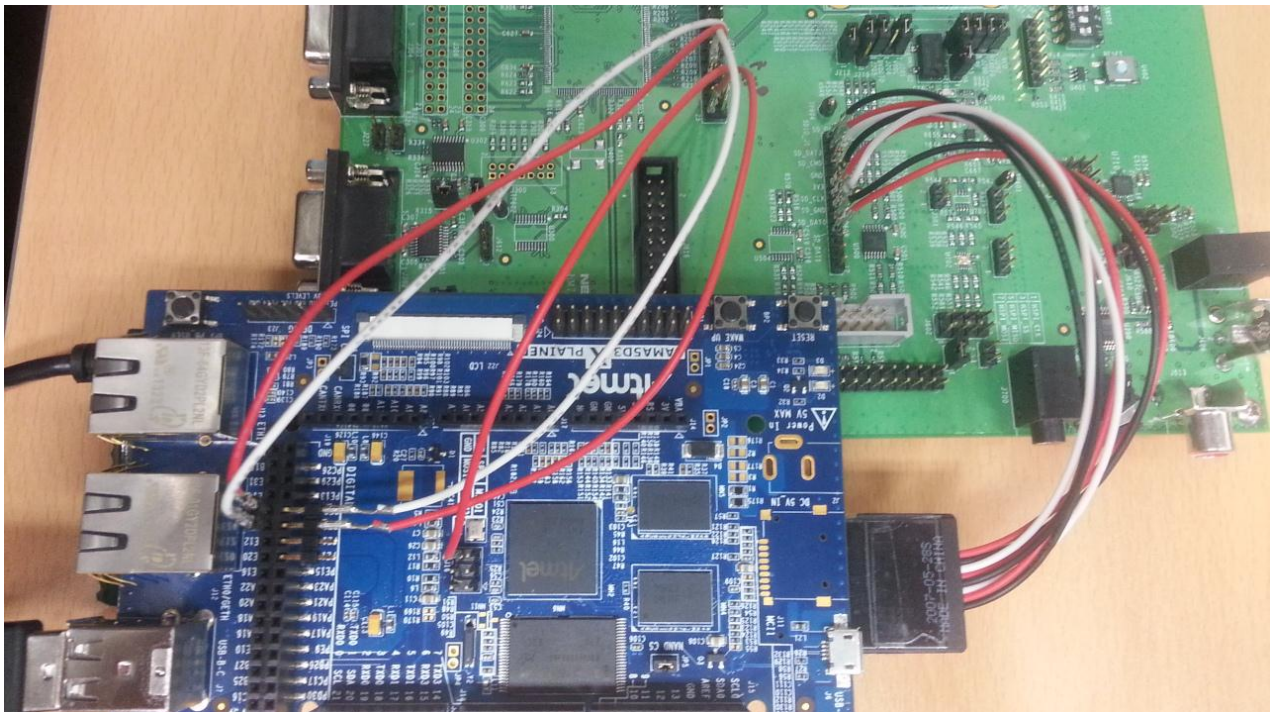
For the WLAN, the MCI0 SD/MMC card slot J10 at the bottom on the SAMA5D3 Xplained should be connected to the SDIO_J1 on the ATWILC3000 EVB. The figure 3 is the SDIO_J1 pin on the ATWILC3000.

Figure 3 ATWILC3000 SDIO_J1 Pin



The SD_DAT0 to SD_DATA3 should be connected to support SDIO 4-bit mode. In addition, SD_CMD, SD_CLK and GND should be connected to the SAMA5D3 Xplained board. The following is real picture showing how to connect ATWILC3000 EVB to SAMA5D3 Xplained board for SDIO interface. The SD/MMC card slot is recommended like the following picture.

Figure 4 ATWILC3000 to SAMA5D3 Xplained via SDIO



6. Running ATWILC3000 for the WLAN

This section briefly introduces how to run ATWILC3000 only as STATION mode for WLAN with the wireless tools provided from Buildroot package. For more information on wireless tools to run the ATWILC3000 in the Linux platform, refer to the GitHub [wiki](#) or **ATWILC3000-User-Guide-For-Linux.pdf** in the release package.

6.1 Enable the network interface

Make sure the wilc3000 module already is loaded in the kernel by issuing the following command.

```
$ insmod /lib/modules/3.10.0/kernel/drivers/net/wireless/Atmel/wilc3000/wilc3000.ko
```

This will enable the wilc3000 device and firmware will be downloaded.

```
$ ifconfig wlan0 up
```

6.2 Scan for networks

Scan for neighboring networks. A list of networks with information on e.g. ssid, WPS and signal levels will be shown when the scan is completed.

```
$ iw wlan0 scan
```

6.3 Connect to an unprotected network

Connect to an unprotected network that was found during the scan. In this case we will choose the ssid *DEMO_AP*.

```
$ iw wlan0 connect DEMO_AP
```

We can check the connection by invoking the following command.

```
$ iw wlan0 link

Connected to 00:26:66:23:05:a4 (on wlan0)
    SSID: DEMO_AP
    freq: 2452
    signal: -61 dBm
    tx bitrate: 36.0 MBit/s
```

6.4 Connect to an WEP encrypted network

The ATWILC3000 on the Linux platform can connect to the access point that uses WEP 64-bit encryption, shared key authentication and with key index 0 set to 1122334455 for example.

```
$ iw wlan0 connect -w DEMO_AP key 0:1122334455

wlan0 (phy #0): connected to 00:26:66:23:05:a4
```

6.5 Connect to an WEP encrypted network with WPA supplicant

The WPA supplicant can be better choice to connect to the WEP encrypted network instead of *iw* command. For more information on the WPA supplicant, refer to the <https://github.com/atwilc3000/driver/wiki/wifi>. Make sure the WPA supplicant daemon runs in background before using WPA supplicant.

```
$ wpa_supplicant -b -Dnl80211 -iwlan0 -c /etc/wpa_supplicant.conf
```

The configuration file is usually located in the **/etc/wpa_supplicant.conf**. The following is simple WPA supplicant configuration. If not available, make it by referring the following.

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
```

The following shows how to connect the access point secured with open WEP 64 bit security. Assume the SSID is con_system with WEP key, 1122334455.

```
$ wpa_cli -iwlan0 add_network
$ wpa_cli -iwlan0 set_network 0 mode 0
$ wpa_cli -iwlan0 set_network 0 auth_alg OPEN
$ wpa_cli -iwlan0 set_network 0 key_mgmt NONE
$ wpa_cli -iwlan0 set_network 0 wep_key0 1122334455
$ wpa_cli -iwlan0 set_network 0 ssid "con_system"
$ wpa_cli -iwlan0 enable_network 0
```

Check the link status by issuing the following command.

```
$ wpa_cli 0iwlan0 status
```

6.6 Connect to WPA2 network with WPA supplicant

The following is sample commands to connect to WPA2 network. Assume that the SSID is con_system and the key is 1345678.

```
$ wpa_cli -iwlan0 add_network
$ wpa_cli -iwlan0 set_network 0 mode 0
$ wpa_cli -iwlan0 set_network 0 auth_alg OPEN
$ wpa_cli -iwlan0 set_network 0 key_mgmt WPA-PSK
$ wpa_cli -iwlan0 set_network 0 proto RSN
$ wpa_cli -iwlan0 set_network 0 psk "12345678"
$ wpa_cli -iwlan0 set_network 0 ssid "con_system"
$ wpa_cli -iwlan0 enable_network 0
```

6.7 Assign an IP address

If a static IP address is used, set it using *ifconfig* command. For example, refer to the following command.

```
$ ifconfig wlan0 192.168.0.127
```

If DHCP is used, invoke the DHCP client available on the platform, for example, refer to the following command.

```
$ udhcpc -i wlan0
```

6.8 Verify the connection

Ping the access point to verify the connection by issuing the following command for instance.

```
$ ping atmec.com
```

6.9 Disconnect from the network

```
$ iw wlan0 disconnect
```

We can check the disconnection by invoking the following command.

```
$ iw wlan0 link  
Not connected.
```

7. Running ATWILC3000 for the Bluetooth

This section provides how to run the ATWILC3000 for Bluetooth in brief. This section also introduces the BLueZ tools. Make sure the Bluetooth firmware should be downloaded before attaching the Bluetooth device in your target platform. You can start the BT firmware by the following procedures.

- *insmod wilc3000.ko* on the target platform
- Check the interface of wlan0 by issuing *ifconfig -a*
- Bring up the wlan0 by *ifconfig wlan0 up*

The firmware should be located in **/lib/firmware** on your target platform. Then, it's possible to attach the wilc3000 device to the target platform.

7.1 Attach serial device via UART HCI to BlueZ stack

The wilc3000 Bluetooth driver provides the UART interface right now so the following command should be issued to attach the device. Make sure the **/dev/ttyS3** exists on the target platform. The BT firmware has the baud rate, 1375000 and flow control.

```
# hciattach ttyS3 any 1500000 flow
```

Make sure the HCI interface is created.

```
# hciconfig -a  
hci0: Type: BR/EDR Bus: UART  
BD Address: AB:89:67:45:23:01 ACL MTU: 1021:9 SCO MTU: 255:4  
DOWN  
RX bytes:574 acl:0 sco:0 events:27 errors:0  
TX bytes:411 acl:0 sco:0 commands:27 errors:0  
Features: 0xff 0xff 0xcd 0xfe 0xdb 0xff 0x7b 0x87  
Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3  
Link policy: RSWITCH HOLD SNIFF PARK  
Link mode: SLAVE ACCEPT
```

7.2 Enable the network interface

Bring up the wilc3000 Bluetooth HCI interface.

```
# hciconfig hci0 up
```

7.3 Scan for networks

Scan for neighboring networks. A list of networks with information on BD_ADDR and name will be shown when the scan is completed.

```
# hcitool scan
Scanning ...
    60:6C:66:A4:29:63    D247-PC
    60:03:08:89:93:E7    damiank-mbp1
    94:63:D1:06:52:B5    Jude_android
    48:D2:24:63:5B:C3    n/a.
    E0:06:E6:BE:A8:FA    APDN194
    D8:57:EF:C7:20:4D    SHV-E210K
    78:DD:08:B2:91:C9    ALEX-PC
```

7.4 Connect to network

You can use DBUS interface so that you connect to a network that was found during the scan. In this case, you have to choose the MAC address. Make sure Bluetooth daemon should first run.

```
# bluetoothd &

# export BT_ADAPTER=`dbus-send --system --dest=org.bluez --print-reply /
org.bluez.Manager.DefaultAdapter | tail -n 1 | sed 's/^.*"(.*)"".*$/1/'`

# dbus-send --system --print-reply --dest=org.bluez $BT_ADAPTER
org.bluez.Adapter.CreatePairedDevice string: D8:57:EF:C7:20:4D
objpath:/org/bluez/agent string:NoInputNoOutput
```

You can list up the paired device.

```
# dbus-send --system --print-reply --dest=org.bluez $BT_ADAPTER
org.bluez.Adapter.ListDevices
method return sender=:1.2 -> dest=:1.11 reply_serial=2
array [
    object path "/org/bluez/1348/hci0/dev_D8_57_EF_C7_20_4D"
]
```

You can remove the paired device by issuing the following command:

```
# dbus-send --system --print-reply --dest=org.bluez $BT_ADAPTER
org.bluez.Adapter.RemoveDevice objpath:$BT_ADAPTER/dev_D8_57_EF_C7_20_4D
```

8. Conclusion

This Quick Start Guide described how to integrate the Atmel ATWILC3000 Combo driver in the Linux platform with SAMA5D3 Xplained board. This document also introduced how to run the ATWILC3000 on Linux platform with wireless tool commands in brief.

9. Revision history

Doc. Rev.	Date	Comments
XXXXXA	11/2014	Initial document release
	11/2014	Add wilc3000 Bluetooth
	2/2015	Update the contents

Atmel Corporation

1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
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
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.

16F Shin-Osaki Kangyo
Bldg.
1-6-4 Osaki, Shinagawa-
ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-
0300
Fax: (+81)(3) 6417-
0370

© 2012 Atmel Corporation. All rights reserved. / Rev.: XXXXA-10/14

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.