# SAM Boot Assistant (SAM-BA)

User Guide

# Table of Contents

# Section 1
# Overview

## 1.1 Overview

The SAM Boot Assistant (SAM-BA™) software provides a means of easily programming various Atmel® AT91 ARM® Thumb®-based devices. It runs under Windows® 2000 and XP.

SAM-BA 2.0 is based on AT91Boot_DLL dll.

Communicating with the target device may be done through an RS232, a USB link or a JTAG link.

This guide gives step-by-step instructions for using the software:

- Section 1 provides an overview.
- Section 2 describes software installation.
- Section 3 details the steps necessary to program an AT91 device with SAM-BA.

## 1.2 SAM-BA Features

Key features of the SAM-BA software are:

- Performs in-system programming through JTAG, RS232 or USB interfaces
- May be used via a Graphical User Interface or started from a DOS window
- Runs under Windows 2000 and XP
- Memory and peripheral display content
- Target device memory control: read, write, erase, configure, verify, etc.
- User scripts available
- User scripts executable from SAM-BA Graphical User Interface or a shell

# Section 2

# Installing SAM-BA 2.x

## 2.1 SAM-BA Installation

To install SAM-BA 2.0 (or higher) on a PC:

1. Uninstall the previous version of SAM-BA if already installed.
2. Uninstall the previous version of AT91-ISP if already installed.

SAM-BA 2.0 (or higher) is part of AT91-ISP installation program.

3. Run the **Install AT91-ISP.exe** program.
4. Follow the installation program instructions.
5. When prompted for an installation directory, specify your AT91-ISP installation directory.

SAM-BA 2.0 is installed in the following directory:

```
YOUR_AT91-ISP_INSTALL_DIRECTORY\SAM-BA v2.x
```

# Section 3

# Working with SAM-BA

## 3.1 Running SAM-BA

It is now possible for SAM-BA to execute TCL script files directly from a shell without using the Graphical User Interface.

Connect your board to your communication interface (either the host serial port or the USB device port).

__Warning:__ **The USB cable must NOT be connected to the board for an RS232 use, otherwise the USB interface is chosen by default.**

There are two different ways to start SAM-BA:

1. Click on the different SAM-BA icons

or

2. Type in a shell:

> [*Install Directory*]/SAM-BA.exe [*Communication Interface*] [*Board*] [*Script_File*] [*Script_File Args*]

where:

- [*Communication Interface*]: "\usb\ARM0" for USB, "\jlink\ARM0" for JTAG, "COMx" for RS232 where x is the COM port number
- [*Board*]: the name of the board accessible through the Choose Protocol window (see Figure 3-1)
- [*Script_File*] (Optional): Path to the TCL Script File to execute
- [*Script_File Args*] (Optional): TCL Script File Arguments

For example: > C:\SAM-BA.exe \usb\ARM0 AT91SAM7S64-EK lib/historyCommand.tcl AT91C_DBGU_THR

***Note:*** If you enter fewer than three arguments in the command line, SAM-BA opens the Choose Protocol window.

***Note:*** If you enter bad arguments in the command line or if there are communication problems, SAM-BA is not able to start.

## 3.2 Connecting the Board

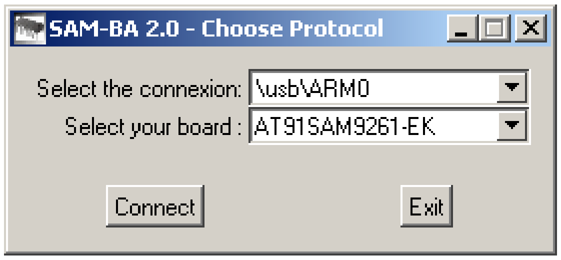
As SAM-BA uses AT91Boot_DLL Scan function to determine all devices connected to the PC, it is recommended to connect the target to the PC before launching SAM-BA.

## 3.3 Selecting a Board

When SAM-BA starts, a pop-up window (see Figure 3-1) appears that allows you to choose the board. Once SAM-BA's main window is displayed, the name of the board is shown at the top of the ***Memory display*** area and on the right side of the status bar (see Figure 3-2).

***Note:*** If you want to change boards, SAM-BA must be restarted.

***Figure 3-1.*** Choose Protocol Window


SAM-BA 2.0 - Choose Protocol

Select the connexion: \usb\ARM0

Select your board : AT91SAM9261-EK

## 3.4 Selecting a Communication Interface

When SAM-BA starts, the communication interface to be used is chosen. All connected devices are listed in the "Select the connection" list. Connections are described by the following strings in the list:

- *"\usb\ARMX"* for USB connected devices
- *"\jlink\ARMX"* for SAM-ICE/JLink connected devices
- *"COMX"* for available COM ports

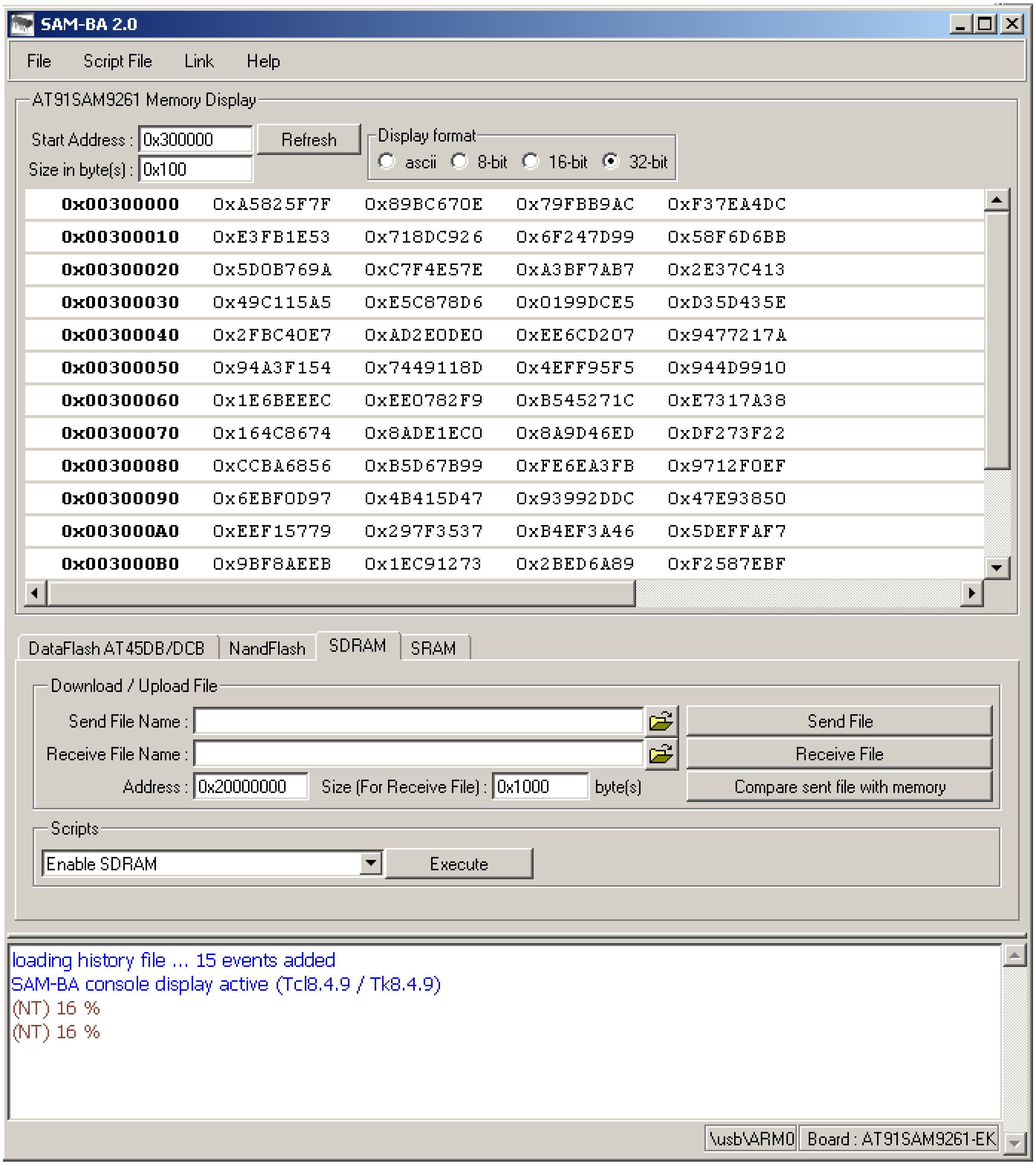
**Warning: The USB cable must NOT be connected to the board for an RS232 use, otherwise the USB interface is chosen by default.**

***Note:*** If the type of connection (RS232/USB/JTAG) is changed, the target must be rebooted and SAM-BA must be restarted.

Click on the Connect button. The main window appears (see Figure 3-2). It contains three different areas. From top to bottom, they are:

- ***Memory display*** area
- ***Memory download*** area
- ***TCL Shell*** area

***Figure 3-2.*** SAM-BA Main Window



AT91SAM9261 Memory Display

Start Address : 0x300000

Size in byte(s) : 0x100

Display format: ascii 8-bit 16-bit [✓ 32-bit]

```
0x00300000   0xA5825F7F   0x89BC670E   0x79FBB9AC   0xF37EA4DC
0x00300010   0xE3FB1E53   0x718DC926   0x6F247D99   0x58F6D6BB
0x00300020   0x5D0B769A   0xC7F4E57E   0xA3BF7AB7   0x2E37C413
0x00300030   0x49C115A5   0xE5C878D6   0x0199DCE5   0xD35D435E
0x00300040   0x2FBC40E7   0xAD2E0DE0   0xEE6CD207   0x9477217A
0x00300050   0x94A3F154   0x7449118D   0x4EFF95F5   0x944D9910
0x00300060   0x1E6BEEEC   0xEE0782F9   0xB545271C   0xE7317A38
0x00300070   0x164C8674   0x8ADE1EC0   0x8A9D46ED   0xDF273F22
0x00300080   0xCCBA6856   0xB5D67B99   0xFE6EA3FB   0x9712F0EF
0x00300090   0x6EBF0D97   0x4B415D47   0x93992DDC   0x47E93850
0x003000A0   0xEEF15779   0x297F3537   0xB4EF3A46   0x5DEFFAF7
0x003000B0   0x9BF8AEEB   0x1EC91273   0x2BED6A89   0xF2587EBF
```

Download / Upload File

Send File Name :

Receive File Name :

Address : 0x20000000 Size (For Receive File) : 0x1000 byte(s)

Scripts

Enable SDRAM

```
loading history file ... 15 events added
SAM-BA console display active (Tcl8.4.9 / Tk8.4.9)
(NT) 16 %
(NT) 16 %
```

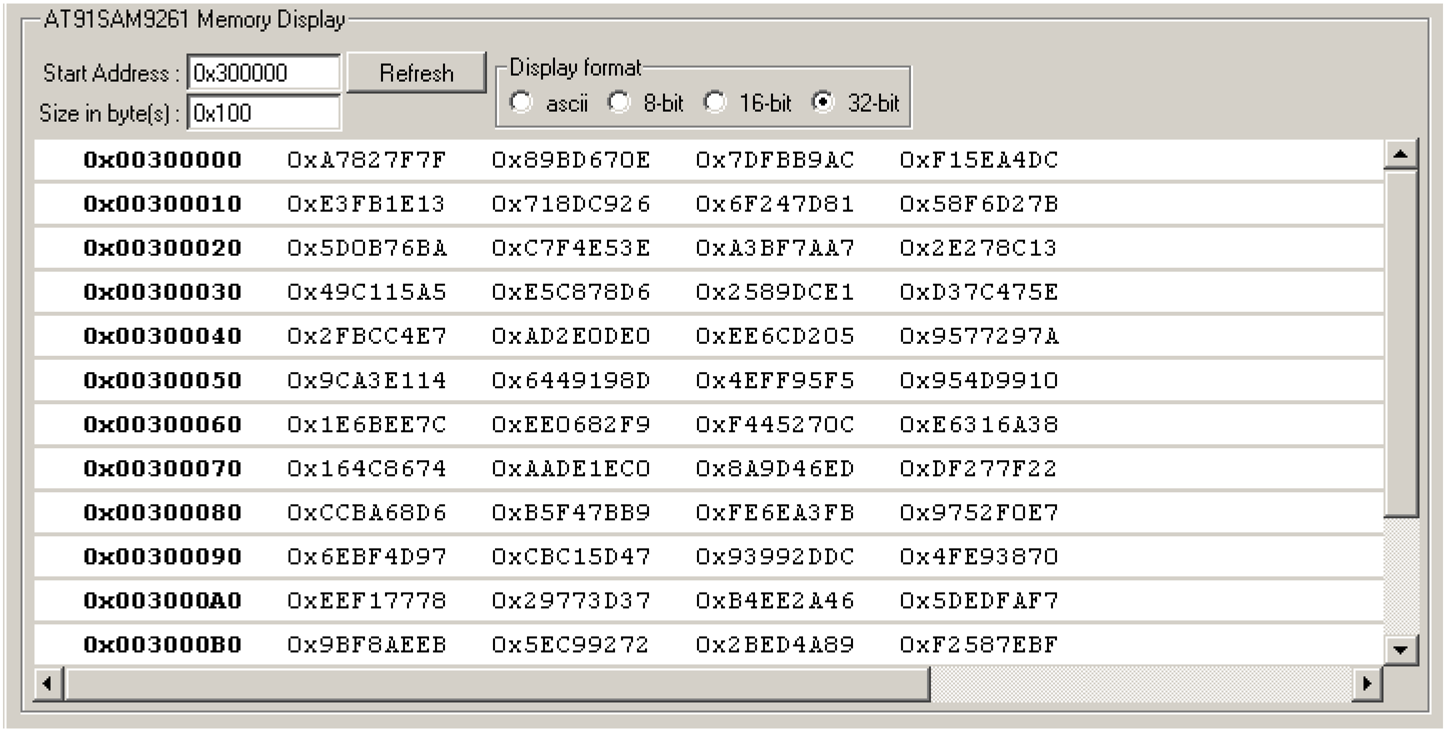
\usb\ARM0 Board : AT91SAM9261-EK

## 3.5 Memory Display Area

In this area you can display a part of the microcontroller memory content. Three different display formats can be used: 32-bit word, 16-bit half-word or 8-bit byte, with a maximum display of 1024-byte long memory area. Values can also be edited by double-clicking on them (see Section 3.5.2).

***Note:*** Only valid memory areas or system/user peripheral areas are displayed. An error message is written in the TCL Shell area (see Section 3.7) if a forbidden address is supplied or if a memory overrun occurs.

***Figure 3-3.*** Memory Display Area



Start Address : 0x300000
Size in byte(s) : 0x100
Display format: ascii 8-bit 16-bit [✓ 32-bit]

```
0x00300000   0xA7827F7F   0x89BD670E   0x7DFBB9AC   0xF15EA4DC
0x00300010   0xE3FB1E13   0x718DC926   0x6F247D81   0x58F6D27B
0x00300020   0x5D0B76BA   0xC7F4E53E   0xA3BF7AA7   0x2E278C13
0x00300030   0x49C115A5   0xE5C878D6   0x2589DCE1   0xD37C475E
0x00300040   0x2FBCC4E7   0xAD2E0DE0   0xEE6CD205   0x9577297A
0x00300050   0x9CA3E114   0x6449198D   0x4EFF95F5   0x954D9910
0x00300060   0x1E6BEE7C   0xEE0682F9   0xF445270C   0xE6316A38
0x00300070   0x164C8674   0xAADE1EC0   0x8A9D46ED   0xDF277F22
0x00300080   0xCCBA68D6   0xB5F47BB9   0xFE6EA3FB   0x9752F0E7
0x00300090   0x6EBF4D97   0xCBC15D47   0x93992DDC   0x4FE93870
0x003000A0   0xEEF17778   0x29773D37   0xB4EE2A46   0x5DEDFAF7
0x003000B0   0x9BF8AEEB   0x5EC99272   0x2BED4A89   0xF2587EBF
```

### 3.5.1 Read Memory Content

1. Enter the address of the area you want to read in the *Starting Address* field.

***Note:*** If a wrong address value is entered, an error message is displayed in the TCL Shell area.

2. Enter the size of the area to display.

***Note:*** If a wrong size is entered, an error message is displayed in the TCL Shell area.

3. Choose display format: 32-bit word, 16-bit half-word or 8-bit byte. This automatically refreshes the memory contents.
4. Press the *Refresh* button.

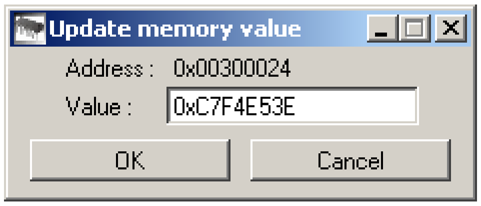
### 3.5.2 Edit Memory Content

Some memories and/or embedded peripherals can be edited:

1. Double-click on the value you want to update. An editable pop-up window appears (see Figure 3-4).

***Note:*** Only some memories can be updated this way, e.g., static RAM or SDRAM (if previously initialized). If you try to write the other memory types, nothing hap-pens.

***Figure 3-4.*** Update Memory Value Window



Address : 0x00300024
Value : 0xC7F4E53E

2. Press OK to update the value in the ***Memory display*** area. The corresponding TCL command is displayed in the TCL Shell area.

***Note:*** Only the lowest bits of the value are taken into account if the format of the value entered is higher than the display format.

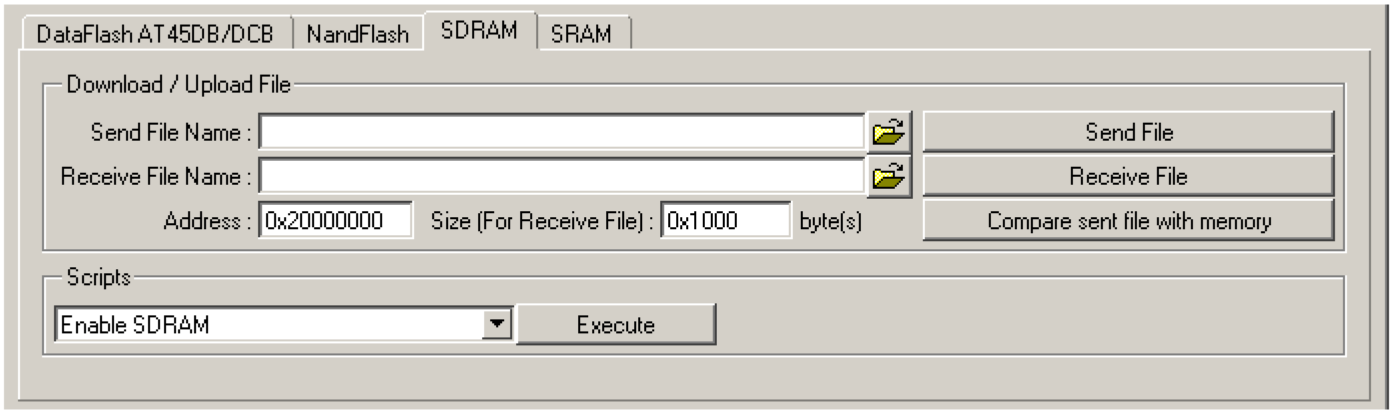
## 3.6 Memory Download Area

The ***Memory download*** area provides a simple way to upload and download data.

For each memory, files can be sent and received and the target's memory content can be compared with a file on your computer (see Figure 3-5).

This area also gives access to some specific scripts for the different embedded memories (SRAM, SDRAM, Flash, DataFlash, etc.).

***Figure 3-5.*** Memory Download Area



Send File Name :
Receive File Name :
Address : 0x20000000 Size (For Receive File) : 0x1000 byte(s)
Enable SDRAM

### 3.6.1 Upload a File

- First, select the memory by clicking on its corresponding tab.
- Enter the file name location in the *Send File name* field or open the file browser by clicking on the Open Folder button and select it.

If you enter a wrong file name, an error message will be displayed in the TCL Shell (see Section 3.7).

- Enter the destination address in the selected memory where the file should be written.

If you enter a forbidden address, or if your file overruns the memory size, an error message is displayed in the TCL Shell.

***Note:*** A forbidden address corresponds to an address outside the selected memory range address.

- Send the file using *the Send File* button. Make sure that the memory is correctly initialized before sending any data.

### 3.6.2 Download Data to a File

- First, select the memory by clicking on its corresponding tab.
- Enter the file name location in the *Receive File name* field or open the file browser by clicking on the Open Folder button and select it.

If you enter a wrong file name, an error message is displayed in the TCL Shell (see Section 3.7).

- Enter the address of the first data to read in the *Address* field.
- Enter the data size to read in the *Size* field.

If you enter a forbidden address, or if your file size overruns the memory size, an error message is displayed in the TCL Shell.
***Note:*** A forbidden address corresponds to an address outside the selected memory range address.

- Get data using *Receive File* button. Make sure that your memory is correctly initialized before getting any data.

### 3.6.3 Compare Memory with a File

Usually, this feature allows to check if a sent file was correctly written into the memory, but you can compare any files with your memory content. The comparison is made on the size of the selected file.

- First, select the concerned memory by clicking on its corresponding tab.
- Enter the file name location in the *Send File name* field or open the file browser by clicking on the Open Folder button and select it.

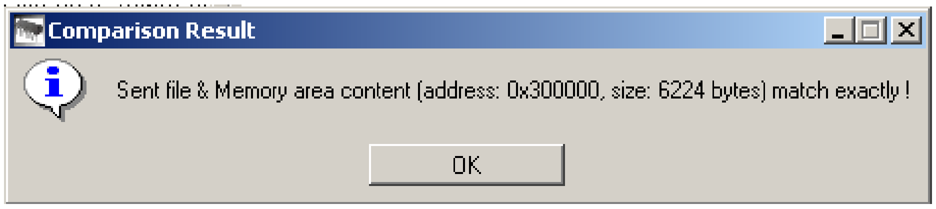
If you enter a wrong file name, an error message will be displayed in the TCL Shell (see Section 3.7).

- Enter the address of the first data to compare with the selected file in the *Address* field.

If you enter a forbidden address, or if your file size overruns the memory size, an error message is displayed in the TCL Shell.
***Note:*** A forbidden address corresponds to an address outside the selected memory range address.

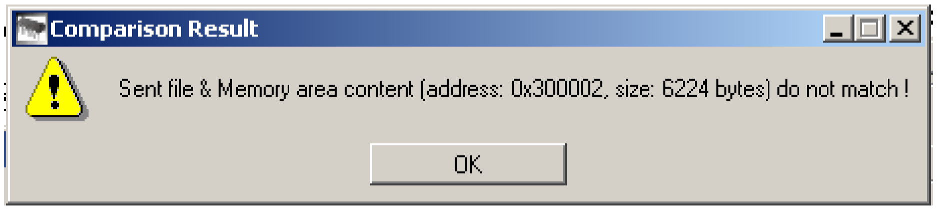
- Compare the selected file with the memory content using the *Compare sent file with memory* button. A message box is displayed if the file matches or not with the memory content of the file size. Make sure that your memory is correctly initialized before comparing any data.

***Figure 3-6.*** Comparison Result Successful



Sent file & Memory area content (address: 0x300000, size: 6224 bytes) match exactly !

***Figure 3-7.*** Comparison Result Failed



Sent file & Memory area content (address: 0x300002, size: 6224 bytes) do not match !

### 3.6.4 Memory Scripts

For each memory some scripts may be supplied. Usually, these scripts allow you to configure and initialize quickly the corresponding memories (SDRAM initialization, erase all Flash, etc).

- To execute a script, select the memory by clicking on its tab.
- Select the script to launch in the list box.
- Click on the *Execute* button.

***Note:*** Messages which inform of the correct execution of the script are displayed in the TCL Shell (see Section 3.7) and/or through a message box.

### 3.6.5 NAND Flash Memory Algorithms

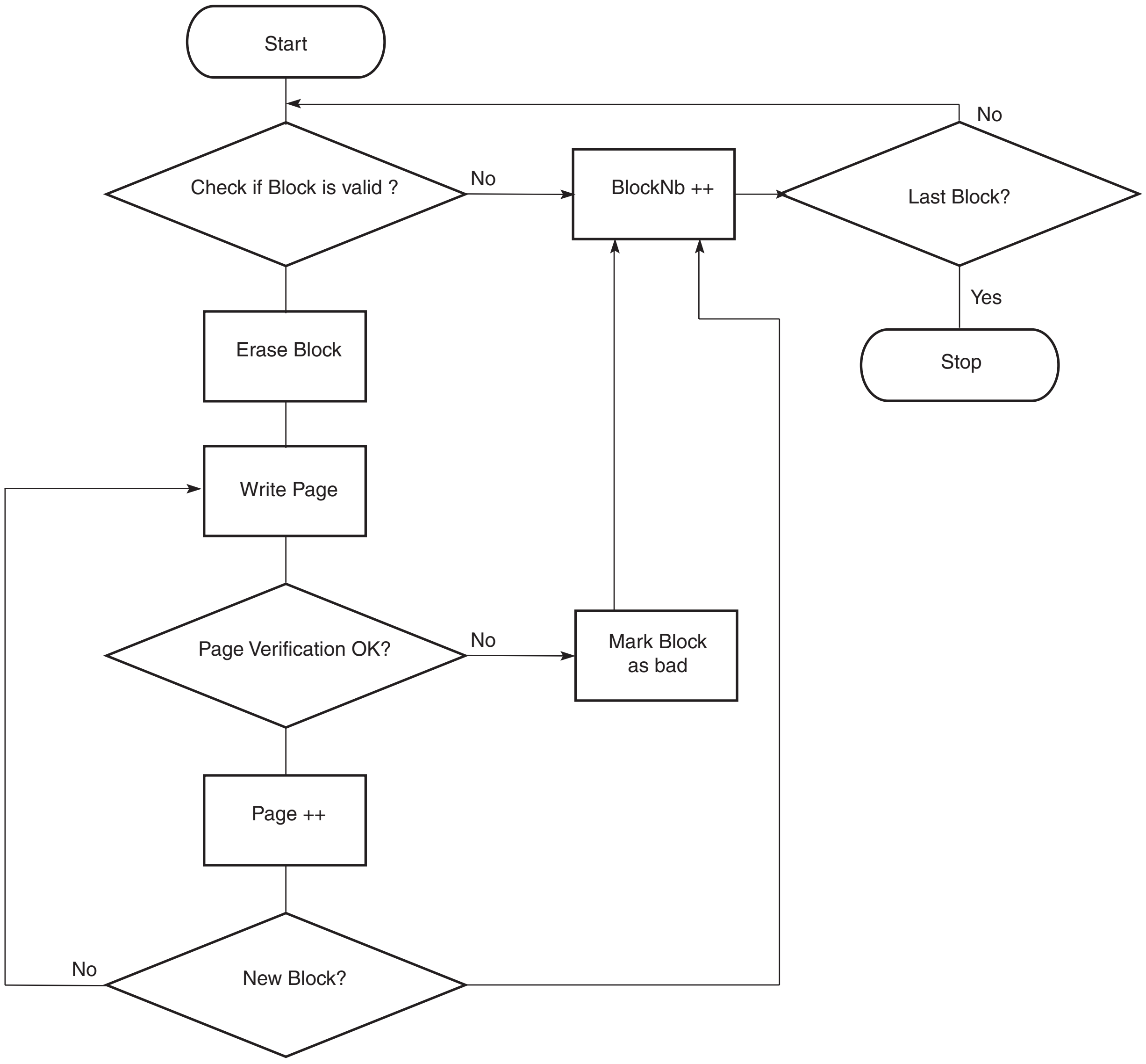
The following information concerns only AT91 parts that contain an External Bus Interface.

As a NAND Flash memory may contain invalid blocks, an Invalid Block Management algorithm has to be implemented.

If an invalid block is detected, it is skipped and data are written in the next block. The same principle has to be implemented in a user application in order to read data from a NAND Flash memory written by SAM-BA application.

### 3.6.5.1 SAM-BA NAND Flash Writing Algorithm

***Figure 3-8.*** SAM-BA NandFlash Writing Algorithm



### 3.6.5.2 Invalid Block Management

The SAM-BA application marks a NandFlash block as invalid by setting the “BadBlock Info” byte in the 8-byte Invalid Block Information structure (see Figure 3-9) to a value different from 0xFF. This is performed for the first two pages of each bad block.

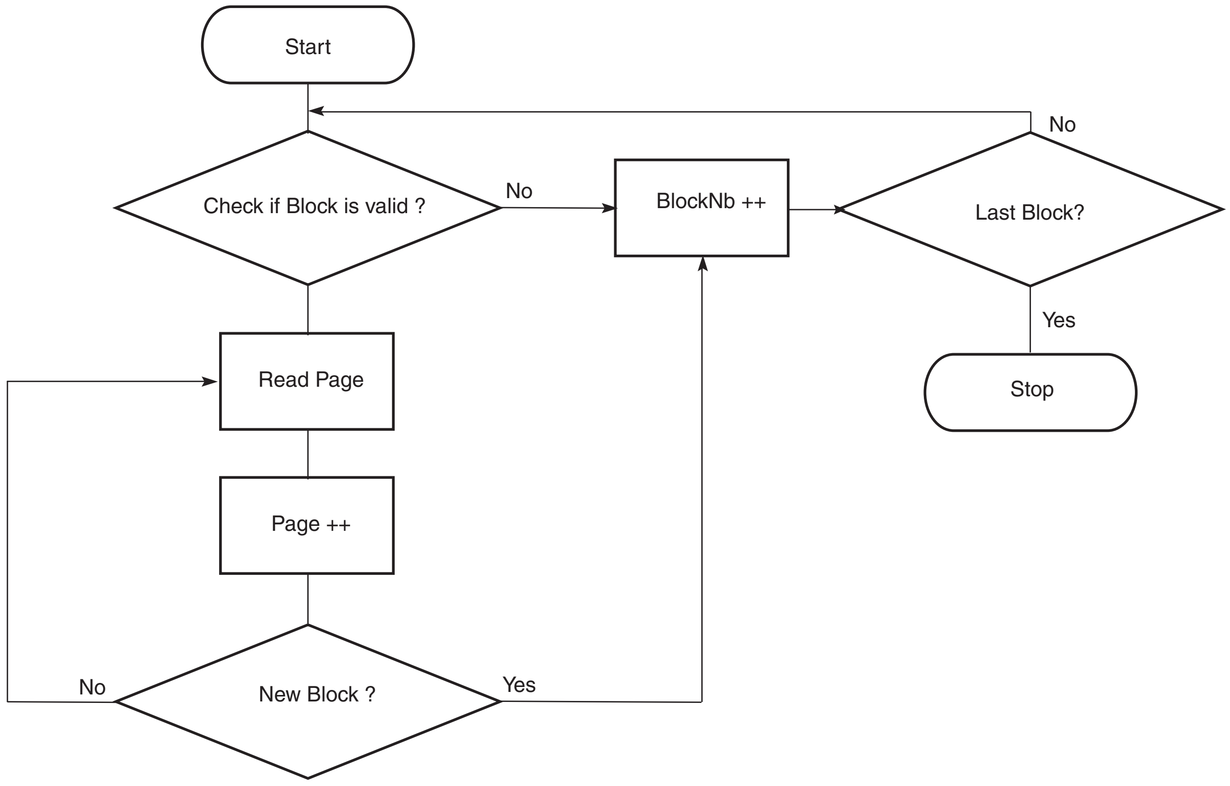
***Figure 3-9.*** 8-byte Invalid Block Information structure located at the beginning of the spare location of the first two pages of a block

| 0 … 3 | 4 | 5 | 6 … 7 |
|---|---|---|---|
| Reserved | Reserved | BadBlock Info | Reserved |

### 3.6.5.3 Reading NAND Flash from a User Application

To read data from a NandFlash memory written by a SAM-BA application, a user application must implement the algorithm represented on Figure 3-10.

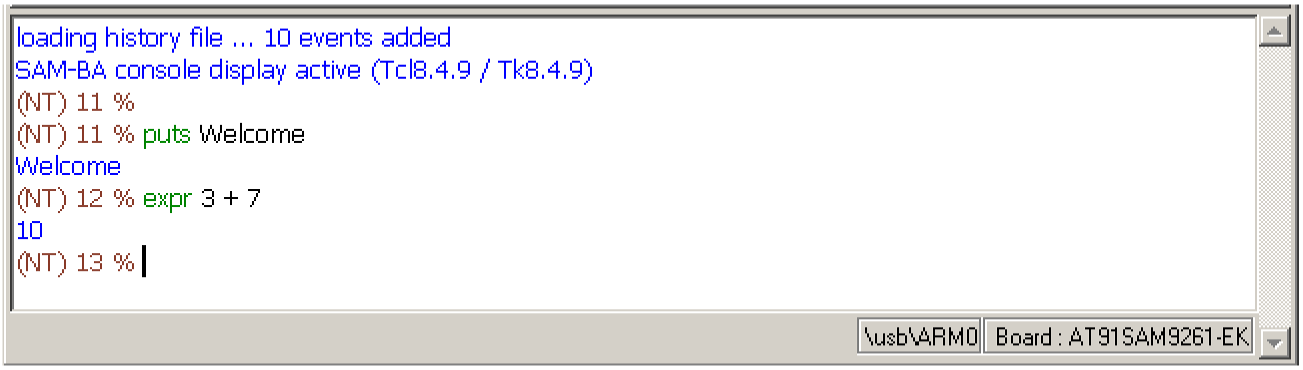
***Figure 3-10.*** SAM-BA NandFlash Writing Algorithm



## 3.7 TCL Shell Area

This is a standard TCL shell. Everything you type in the shell is interpreted by a TCL interpreter. This area gives you access to standard TCL commands. Type "*puts Welcome*" and you will get the result "*Welcome",* type *"expr 3 + 7"* and you will get the result *"10"* (see Figure 3-12).

**Figure 3-11.** TCL Shell Window



```
loading history file ... 10 events added
SAM-BA console display active (Tcl8.4.9 / Tk8.4.9)
(NT) 11 %
(NT) 11 % puts Welcome
Welcome
(NT) 12 % expr 3 + 7
10
(NT) 13 %
```

### 3.7.1 SAM-BA Built-in Commands

TCL is a commonly used scripting language for automation. This interpreted language offers a standard set of commands which can be extended by application specific commands written in C or other languages. Tutorials and a manual can be downloaded here: http://www.tcl.tk/doc/.

Specific commands have been added to the SAM-BA TCL interpreter to interface with AT91SAM devices. These basic commands can be used to easily build more complex routines. Please refer to the AT91 In-system Programmer (ISP) User Guide, Atmel Literature number 6224, Section 4, "AT91Boot_TCL Interface" for a complete description of all available functions.

When SAM-BA starts, a structure containing board and connection information is set. This global variable name is `target`, and the `global target` statement must be declared in any procedure using an API function.

Target structure contents are:

- handle: identifier of the link used to communicate with the target
- board: a string containing the board name (i.e.: "AT91SAM9261-EK")
- connection: connection type. Can be : "\usb\ARMx" for a USB link, "\jlink\ARMx" for a JTAG link, or "COMx" for a serial link (with x indicating the COM port used)
- comType: 1 for serial; 0 for other (USB or JTAG)

These commands are mainly used to send and receive data from the device:

***Table 3-1.*** Commands Available through the TCL Shell

| **Commands** | **Argument(s)** | **Example** |
|---|---|---|
| TCL_Write_Byte | Handle Value Address err_code | **TCL_Write_Byte** $target(handle) 0xCA 0x200001*err_code* |
| TCL_Write_Short | Handle Value Address err_code | **TCL_Write_Short** $target(handle) 0xCAFE 0x200002 *err_code* |
| TCL_Write_Int | Handle Value Address err_code | **TCL_Write_Int** $target(handle) 0xCAFEDECA 0x200000 *err_code* |
| TCL_Read_Byte | Handle Address err_code | **TCL_Read_Byte** $target(handle) 0x200003 *err_code* |
| TCL_Read_Short | Handle Address err_code | **TCL_Read_Short** $target(handle) 0x200002 *err_code* |
| TCL_Read_Int | Handle Address err_code | **TCL_Read_Int** $target(handle) 0x200000 *err_code* |
| TCL_Compare | “fileName1” “fileName2” | **TCL_Compare** “C:/temp/file1.bin“ “C:/temp/file2.bin“ |
| TCL_Go | Handle Address err_code | **TCL_Go** $target(handle) 0x20008000 *err_code* |
| send_file | Memory Name Address | **send_file** {SDRAM} “C:/temp/file1.bin“ 0x20000000 |
| receive_file | Memory Name Address Size | **receive_file** {SDRAM} “C:/temp/file1.bin“ 0x10000 |
| compare_file | Memory Name Address | **compare_file** {SDRAM} “C:/temp/file1.bin“ 0x20000000 |

Example:

```
# dummy_err needs to be defined !!!
set dummy_err 0
TCL_Go $target(handle) 0x300000 dummy_err
```

- Write commands: Write a byte (**TCL_Write_Byte**), a halfword (**TCL_Write_Short**) or a word (**TCL_Write_Int**) to the target.
  - *Handle:* handler number of the communication link established with the board.
  - *Value*: byte, halfword or word to write in decimal or hexadecimal.
  - *Address*: address in decimal or hexadecimal.
  - *Output*: nothing.
- Read commands: Read a byte (**TCL_Read_Byte**), a halfword (**TCL_Read_Short**) or a word (**TCL_Read_Int**) from the target.
  - *Handle:* handler number of the communication link established with the board.
  - *Address*: address in decimal or hexadecimal.
  - *Output*: the byte, halfword or word read in decimal.

***Notes:*** ***1.*** **TCL_Read_Int** returns a signed integer in decimal. For example, reading with **TCL_Read_Int** $target(handle) *0xFFFFFFFF* command returns *-1* whereas reading *0xFF* with **TCL_Read_Byte** command returns *255*.

***2.*** To obtain the result in hexadecimal format, use the TCL “format“ command:

```
puts [format “%x“ [TCL_Read_Int $target(handle) 0x300000 err_code]]
```

- **send_file** : Send a file in a specified memory.
  - *Memory*: memory tag.
  - *Name*: absolute path file name (in quotes) or relative path from the current directory (in quotes).
  - *Address*: address in decimal or hexadecimal (in quotes).
  - *Output*: information about the corresponding command on the TCL Shell.
- **receive_file**: Receive data into a file from a specified memory.
  - *Memory*: memory tag.
  - *Name*: absolute path file name in quotes or relative path (from the current directory) file name in quotes.
  - *Address*: address in decimal or hexadecimal (in quotes).
  - *Size*: size in decimal or hexadecimal (in quotes).
  - *Output*: information about the corresponding command on the TCL Shell.
- **compare_file** : Compare a file with memory data.
  - *Memory*: memory tag.
  - *Name*: absolute path file name (in quotes) or relative path from the current directory (in quotes).
  - *Address*: address of the first data to compare with the file in decimal or hexadecimal (in quotes).
  - *Output*: information about the command progress on the TCL Shell.
- **TCL_Compare** : Binary comparison of two files.
  - *fileName1*: absolute path file name (in quotes) or relative path from the current directory (in quotes) of the first file to compare.
  - *fileName2*: absolute path file name (in quotes) or relative path from the current directory (in quotes) of the second file to compare with the first.
  - *Output*: return 1 in case of error, 0 if files are identical.
- **TCL_Go** : Jump to a specified address and execute the code.
  - *Handle:* handler number of the communication link established with the board.
  - *Address*: address to jump to in decimal or hexadecimal.

**Note: If you do not want to exit SAM-BA, the code must include lines that enable a return to the SAM-BA application. Otherwise, SAM-BA does not recover correctly.**

The memory tag (in curly brackets) is the name of the memory module defined in the memoryAlgo array in the board description file. For example: `{DataFlash AT45DB/DCB}`

**Warning:** If you leave SAM-BA, be sure to reboot your board before launching it the next time.

## 3.8 Script File Functionality

SAM-BA allows you to create, edit and execute script files. A script file configures your device easily or automatically runs significant scripts.

The *Script File* menu supplies commands to start and stop recording, to execute, reset, edit and save the recording file.

The name of the generate file is *historyCommand.tcl*.

### 3.8.1 Start / Stop / Reset Recording

In the *Script File* menu, select *Start Recording* to begin the record. Now, all the commands that are to be executed in the different blocks of the software are recorded in a specific file called ***historyCommand.tcl***. This file is located in the *usr* directory.

***Note:*** Only this file can be written through the Start / Stop / Reset Recording commands.

If the recording file is not reset, the new recorded commands are added at the end of the ***historyCommand.tcl*** file.

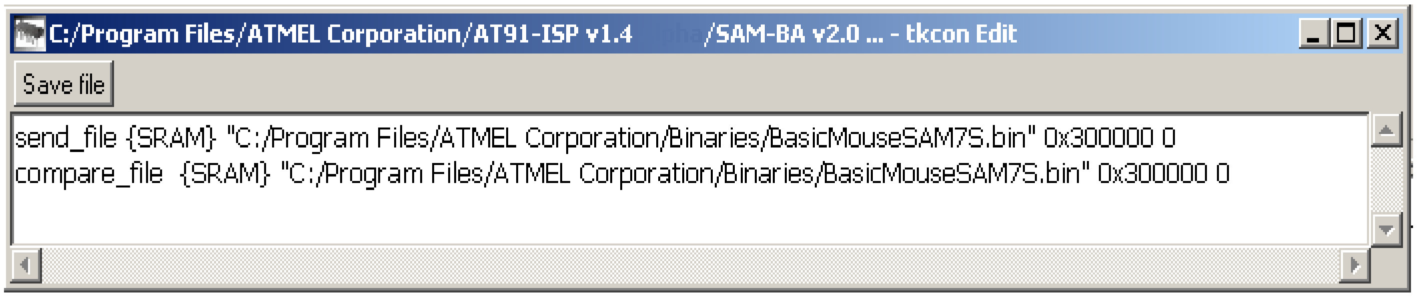
When you want to stop recording, select *Stop Recording* in the menu.

If you want to erase the ***historyCommand.tcl*** content, select *Reset Recording*.

### 3.8.2 Editing the Script File

You have the possibility to edit the ***historyCommand.tcl*** recording file. Use the command *Edit Script File* in the *Script File* menu. A new window appears in which you can edit and save the content. You can save your modified script in another file through the Save file button. Thus several configuration scripts for specific use are available.

***Figure 3-12.*** Script File Edition View


C:/Program Files/ATMEL Corporation/AT91-ISP v1.4 /SAM-BA v2.0 ... - tkcon Edit
Save file

```
send_file {SRAM} "C:/Program Files/ATMEL Corporation/Binaries/BasicMouseSAM7S.bin" 0x300000 0
compare_file  {SRAM} "C:/Program Files/ATMEL Corporation/Binaries/BasicMouseSAM7S.bin" 0x300000 0
```

### 3.8.3 Execute the Script File

As it is now possible for SAM-BA to execute TCL script files directly from a shell without using the GUI. There are two possibilities to execute a script file.

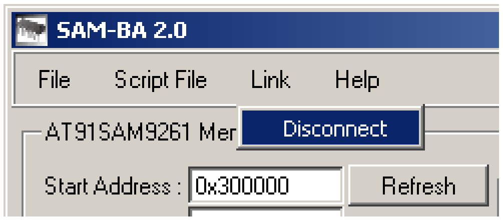
- See Section 3.1 "Running SAM-BA" for more information on how to execute TCL script files from a shell.
- Use the command *Execute Script File* in the GUI *Script File* menu and enter the TCL file to execute. Messages that inform of the correct execution of the script are displayed in the TCL Shell and/or through message boxes.

***Note:*** All TCL commands can be executed through script files.

## 3.9 Deconnection

It is possible to close the current connection if necessary by clicking on the corresponding *Link/Disconnect* menu (see Figure 3-13).

***Figure 3-13.*** Link Menu



# Section 4

# Revision History

## 4.1 Revision History

| Document Ref. | Comments | Change Request Ref. |
|---|---|---|
| 6132A | First issue | |
| 6132B | New feature: TCL scripts now executable directly from a shell: Section 1.2 ”SAM-BA Features”, Section 3.1 ”Running SAM-BA”, Section 3.8.3 ”Execute the Script File”. New sections on NandFlash memory management: Section 3.6.5 ”NAND Flash Memory Algorithms”, Section 3.6.5.1 ”SAM-BA NAND Flash Writing Algorithm”, Section 3.6.5.2 ”Invalid Block Management” and Section 3.6.5.3 ”Reading NAND Flash from a User Application”. | |
| 6132C | Major update to document, SAM-BA 2.0 provides new functions. All sections updated. | |

## Atmel Corporation

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

*Microcontrollers*
2325 Orchard Parkway
San Jose, CA 95131, USA
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, USA
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, USA
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

*Literature Requests*
www.atmel.com/literature