AT91 ARM Thumb Microcontrollers

AT91SAM CAN Bootloader

AT91SAM CAN Bootloader User Notes

1. Description

The CAN bootloader SAM-BA™ *Boot4CAN* allows the user to program the different memories and registers of any Atmel AT91SAM product that includes a CAN without removing them from the system and without the need of a preprogrammed application.

The CAN bootloader manages communication with an external device (host) through the CAN network. It implements a CAN protocol which translates serial CAN communication frames into memory and register accesses.

2. Key Features

Key features of the AT91SAM CAN Bootloader are:

- ARM® Thumb® Compatible
- SAM-BA™ *Boot4CAN* Protocol
  - CAN Used as Physical Layer
  - 7 ISP CAN Identifiers
  - Relocatable ISP CAN Identifiers
  - Software Autobaudrate Mode
- In-system Programming Commands
  - Read/Write Byte/Halfword/Word
  - Read/Write Buffers
  - Read/Write Configuration Bytes:
    CAN Node Number
    CAN Relocatable Identifier Segment
    Autobaudrate Parameters Configuration
  - Start Application Command
- Interface with AT91Boot_DLL dll for ISP Usage

6220A–ATARM–06-Feb-06

3. Bootloader Configuration

Table 3-1 lists the different configuration bytes used by the bootloader. Their value can be accessed through a set of commands. See "Special Commands" on page 15..

Table 3-1. Configuration Bytes Description

Mnemonic	Description	Default Value	Flash Address
NNB	Node NumBer	0xFF	0x100_F00
CRIS	CAN Re-locatable Identifier Segment	0xFF	0x100_F04
ABM	AutoBaud Mode	0xFF	0x100_F08
PROPAG	Propagation Segment	0xFF	0x100_F0C
PHASE1	Phase Segment 1	0xFF	0x100_F10
PHASE2	Phase Segment 2	0xFF	0x100_F14
BRP	Baudrate Prescaler	0xFF	0x100_F18

All the configuration bytes are located in the Flash memory starting at address 0x100_F00. Before connecting a CAN node in a point-to-point connection for the first time, the user must take care that the default values are 0xFF (see "Hardware and Software Constraints" on page 17).

4. Device Initialization

Initialization follows the steps described below:

1. Stack setup for ARM supervisor mode
2. Setup the Embedded Flash Controller
3. Main oscillator frequency detection
4. Switch Master Clock on Main Oscillator
5. Copy code into SRAM
6. C variable initialization
7. PLL setup: PLL is initialized to generate a system clock of 24 MHz
8. Disable of the Watchdog and enable of the user reset
9. CAN PIO, Clock and CAN transceiver configuration
10. Jump to SAM-BA *Boot4CAN* sequence (see ”SAM-BA Boot4CAN” on page 4)

5. CAN Autobaudrate

The supported baudrates are 100 Kbits, 125 Kbits, 250 Kbits, 500 Kbits and 1 Mbit/s.

5.1 CAN Autobaudrate Limitation

The CAN autobaud implemented in SAM-BA *Boot4CAN* is efficient only in point-to-point connection because, in this case, the transmit CAN message is repeated until a hardware acknowledge is done by the receiver. This configuration corresponds to a host trying to send CAN frames until the CAN node with SAM-BA Boot4CAN acknowledges one (Figure 6-3 on page 8).

The bootloader can acknowledge an in-coming CAN frame only if a configuration is found.

This functionality is not guaranteed on a network with several CAN nodes.

6. SAM-BA *Boot4CAN*

This section describes how to start the CAN Bootloader and the higher level protocol over the CAN called SAM-BA *Boot4CAN*.

The CAN is used to transmit information and has the following configuration:

- – Standard CAN Frame compliant with CAN Specification 2.0A (Identifier size is 11 bits long)
- – Frame: Data Frame
- – Baudrate: software autobaud performed by the bootloader.

6.1 SAM-BA *Boot4CAN* Initialization

The SAM-BA *Boot4CAN* initialization principle is to:

- – Check for User Configuration bytes.
- – Wait for CAN Connecting messages and execute if necessary the Software AutoBaudrate (see Figure 6-1).

There are two ways to initialize the CAN controller:

- – Use the Software Autobaudrate
- – Use the User Configuration Bytes stored in the CAN Baudrate Register (CAN_BR) of the product.

The user configuration bytes to be used are, respectively:

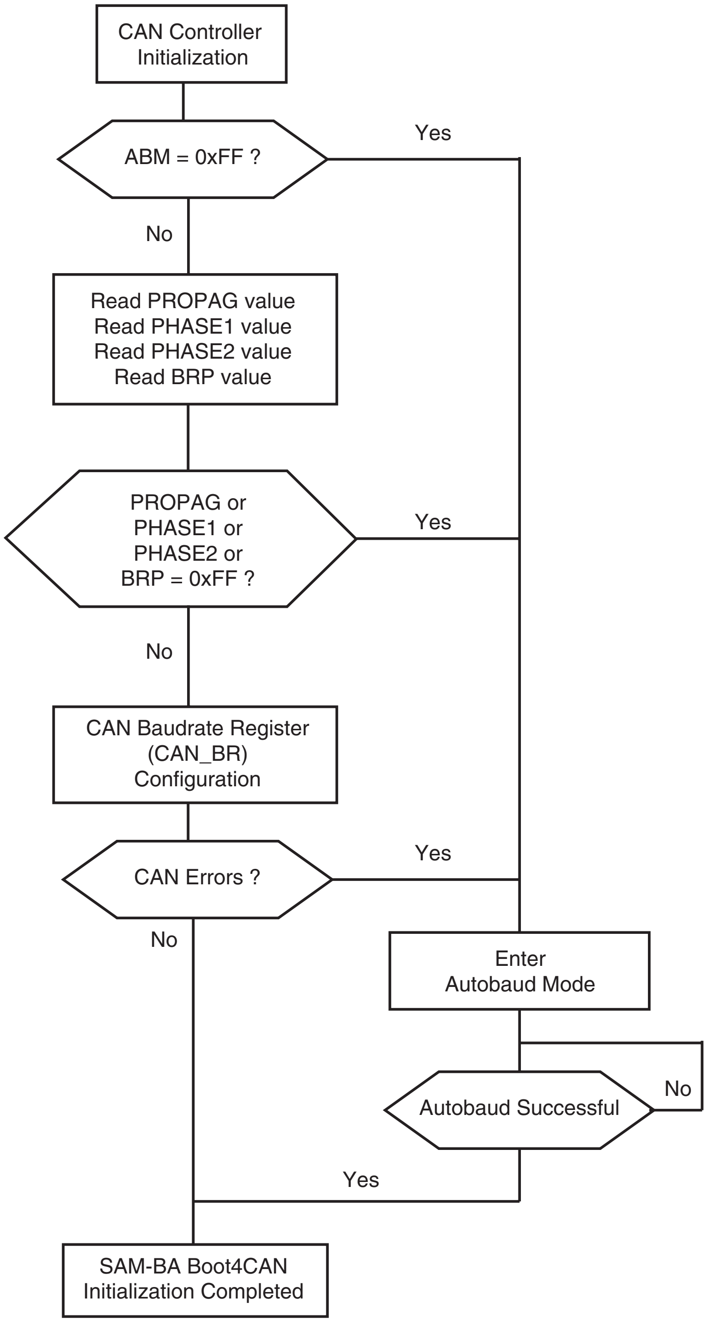
- – PROPAG: Propagation Time Segment: This part of the bit time is used to compensate for the physical delay times within the network.
- – PHASE1 and PHASE2: Phase 1 and Phase 2 Segment: The Phase-Buffer-Segments are used to compensate for edge phase errors. These segments can be lengthened (PHASE1) or shortened (PHASE2) by resynchronization.
- – BRP: Baudrate Prescaler: This field allows user to program the period of the CAN system clock to determine the individual bit timing.

Note: Refer to the full product datasheet for more information on CAN Bit Timing Programmation.

The choice between these two solutions is made with the ABM Configuration Byte:

- – ABM = 0xFF: the Software Autobaudrate is performed.
- – ABM is different from 0xFF: the CAN_BR register configuration is used.

Figure 6-1. SAM-BA *Boot4CAN* Initialization


CAN Controller Initialization
ABM = 0xFF ?
Yes
No
Read PROPAG value
Read PHASE1 value
Read PHASE2 value
Read BRP value
PROPAG or PHASE1 or PHASE2 or BRP = 0xFF ?
Yes
No
CAN Baudrate Register (CAN_BR) Configuration
CAN Errors ?
Yes
No
Enter Autobaud Mode
Autobaud Successful
No
Yes
SAM-BA Boot4CAN Initialization Completed

6.2 SAM-BA *Boot4CAN* Protocol

This protocol is used to:

- Initiate the communication
- Read/Write Byte/Half-word/Word
- Read/Write Data Buffers
- Read/Write Configuration Bytes
- Jump to a specified address

6.2.1 Generic CAN Data Frame Description

Identifier Field	Control Field	Data Field
11-bit	1 byte	8 bytes max

- Identifier: identifies the frame. Only the standard mode (11-bit) is used.
- Control: contains the DLC information in 4 bits (number of data in Data field).
- Data: Data field consists of zero to eight bytes. The interpretation within the frame depends on the Identifier field.

6.2.2 Command Description

Once SAM-BA *Boot4CAN* initialization is complete, the application runs in an infinite loop waiting for different commands (Table 6-1)
Table 6-1 lists the CAN message identifiers defined to manage this protocol:

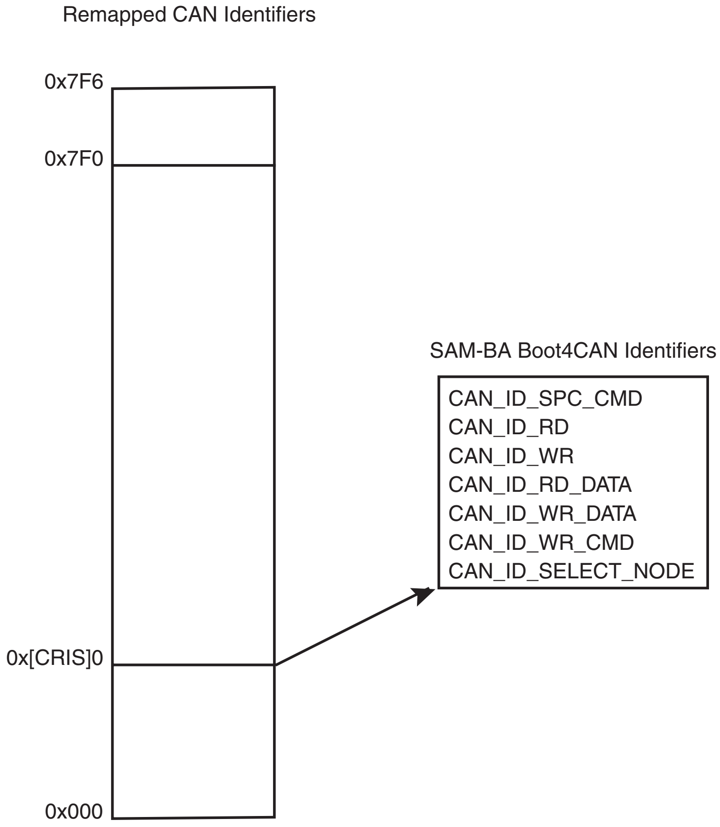
Table 6-1. Commands Available through the SAM-BA *Boot4CAN*

Identifier	Action	Value	Initial Value
CAN_ID_SELECT_NODE	Open/Close a communication with a node	0x[CRIS]0	0x0
CAN_ID_WR_CMD	Initiate a Write Buffer Command	0x[CRIS]1	0x1
CAN_ID_WR_DATA	Write a Buffer	0x[CRIS]2	0x2
CAN_ID_RD_DATA	Read a Buffer	0x[CRIS]3	0x3
CAN_ID_RD	Read a Byte/Half-word/Word	0x[CRIS]4	0x4
CAN_ID_WR	Write a Byte/Half-word/Word	0x[CRIS]5	0x5
CAN_ID_SPC_CMD	Special Commands	0x[CRIS]6	0x6

It is possible to allocate a new value for CAN ISP identifiers by writing the byte CRIS with the base value for the group of identifier s(Figure 6-2).

The maximum value for CRIS is 0x7F. All values superior to 0x7F will be considered as a 0x00 value.

Figure 6-2. CAN Identifier Remapping



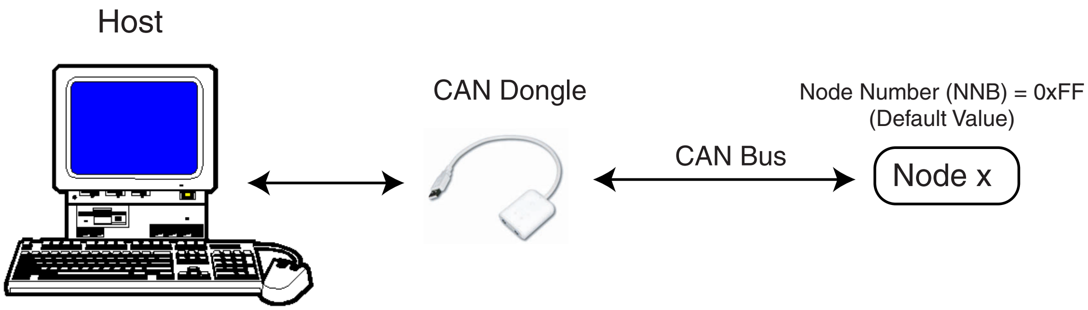
6.2.3 Open and Close Communication

The communication with a device (CAN node) must be opened prior to initiation of any communication.

To open communication with the device, the host must send a “connecting” CAN message (CAN_ID_SELECT_NODE) with the node number (NNB) passed in parameter.

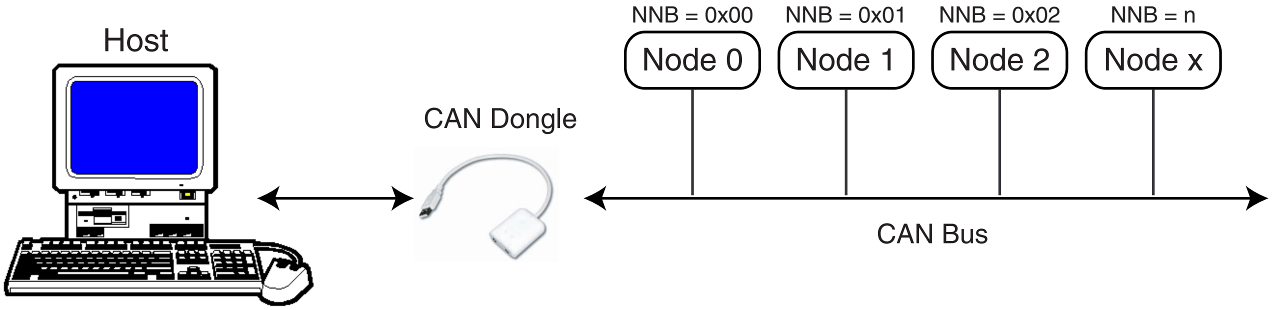
If the node number passed is equal to 0xFF then the CAN bootloader accepts the communication (see Figure 6-3).

Figure 6-3. Point To Point Connection



Otherwise the node number passed in parameter must be equal to the local Node Number (NNB) (see Figure 6-4).

Figure 6-4. Network Connection



Before opening a new communication with another device, the current device communication must be closed with its connecting CAN message (CAN_ID_SELECT_NODE).

6.2.3.1 Request from Host

Identifier	Length	data[0]
CAN_ID_SELECT_NODE	1	Node Number (NNB)

Note: NNB is the Node Number Byte to which the host wants to talk.

6.2.3.2 Answer from Bootloader

Identifier	Length	data[0]	data[1]	Comment
CAN_ID_SELECT_NODE	2	Boot Version	0x00	The communication is closed
			0x01	The communication is opened

Note: data[0] contains the bootloader version.

If the communication is closed, then no other messages are managed by the bootloader.

6.2.3.3 Example

Open The Communication:

	Identifier	Length	Data
HOST	CAN_ID_SELECT_NODE	01	FF
BOOTLOADER	CAN_ID_SELECT_NODE	02	01 01

6.2.4 Read Byte/Half-word/Word Command

This command allows the user to read bytes, half-words or words at a specified address.

This operation can be executed only with a device previously opened in communication.

6.2.4.1 Request from Host

Identifier	Length	data[0]	data[1]	data[2]	data[3]	data[4]	Description
CAN_ID_RD	5	0x05	Address				Read a byte
		0x06					Read a half word
		0x08					Read a word

6.2.4.2 Answer from Bootloader

Identifier	Length	data[0]
CAN_ID_RD	1	Byte Value
	2	Half Word Value
	4	Word Value

6.2.4.3 Example

`Read A Byte at address 0x202000:`

	Identifier	Length	Data
`HOST`	CAN_ID_RD	`05`	`05 00 20 20 00`
`BOOTLOADER`	CAN_ID_RD	`01`	`69`

6.2.5 Write Byte/Half-word/Word Command

This command allows the user to write bytes, half-words or words at a specified address.

This operation can be executed only with a device previously opened in communication.

6.2.5.1 Request from Host

Identifier	Length	data[0:3]	data[4]	data[5]	data[6]	data[7]	Description
CAN_ID_WR	5	Address	Value	-			Write a byte
	6		Value		-		Write a half word
	8		Value				Write a word

6.2.5.2 Answer from Bootloader

Identifier	Length	data[0]	Description
CAN_ID_WR	1	0x00	Command OK

6.2.5.3 Example

<u>`Write A Word at address 0x202030 (value = 0xCAFEDECA):`</u>

	Identifier	Length	Data
`HOST`	CAN_ID_WR	`08`	`00 20 20 30 CA FE DE CA`
`BOOTLOADER`	CAN_ID_WR	`01`	`00`

6.2.6 Read Data Buffer Command

This command allows the user to read a buffer.

This operation can be executed only with a device previously opened in communication.

To start the read operation, the Host sends a “Read Data Buffer” CAN message (CAN_ID_RD_DATA) with the start address and the end address passed in parameter.

If the number of data to read is greater than 8 bytes, the device splits them into blocks of 8 bytes to be transferred to the host.

6.2.6.1 Request from Host

Identifier	Length	data[0:3]	data[4:7]	Description
CAN_ID_RD_DATA	8	Start Address	End Address	Read bytes from start address to end address

6.2.6.2 Answer from Bootloader

Identifier	Length	data[n]
CAN_ID_RD_DATA	n	Values

6.2.6.3 Example

`Read Data from 0x200000 to 0x200009:`

	Identifier	Length	Data
HOST	CAN_ID_RD_DATA	08	00 20 00 00 00 20 00 09
BOOTLOADER	CAN_ID_RD_DATA	08	01 23 45 67 89 AB CD EF
BOOTLOADER	CAN_ID_RD_DATA	02	EA FF

6.2.7 Write Data Buffer Command

This command allows the user to write a buffer in the device.

This operation can be executed only with a device previously opened in communication.

1. The first step is to indicate the range address to write.
2. The second step is to transmit the data.

The host must take care to transmit 8 data bytes in a CAN message when possible.

To start the write operation, the host sends a “Write Command” CAN message (CAN_ID_WR_CMD) with the start address and the end address passed in parameter.

6.2.7.1 Request from Host

Identifier	Length	data[0:3]	data[4:7]	Description
CAN_ID_WR_CMD	Up to 8	Start Address	End Address	Initiate Write Data Buffer Command

6.2.7.2 Answer from Bootloader

Identifier	Length	data[0]	Description
CAN_ID_WR_CMD	1	0x00	Command OK

The second step is to send data to write.

The host must send a “Write Data“ CAN message (CAN_ID_WR_DATA) with up to 8 data per message and must wait for the device answer before sending the next data.

6.2.7.3 Request from Host

Identifier	Length	data[0]	...	data[7]	Description
CAN_ID_WR_DATA	up to 8	x	...	x	Data to be written

6.2.7.4 Answer from Bootloader

The device has three possible answers:

– If the device is ready to receive new data, it sends a “Write Data” CAN message (CAN_ID_WR_DATA) with the result “Wait for New Command” passed in parameter.

– If the device has finished the programming, it sends a “Write Data” CAN message with the result “Command OK” passed in parameter.

– If the device is not supposed to write data, it sends a “Write Data” CAN message with the result “Command Not OK” passed in parameter..

Identifier	Length	data[0]	Description
CAN_ID_WR_DATA	1	0x00	Command OK
		0x01	Command Not OK
		0x02	Wait for New Command

6.2.7.5 Example

Write ‘0x55’ from 0x200000 to 0x200008:

	Identifier	Length	Data
// Initiate Data Buffer Write			
HOST	CAN_ID_WR_CMD	08	00 20 00 00 00 20 00 08
BOOTLOADER	CAN_ID_WR_CMD	01	00
// Write Data Buffer			
HOST	CAN_ID_WR_DATA	08	55 55 55 55 55 55 55 55
BOOTLOADER	CAN_ID_WR_DATA	01	02 // Wait For New Command
HOST	CAN_ID_WR_DATA	01	55
BOOTLOADER	CAN_ID_WR_DATA	01	00 // Command OK
// Try to make a unwanted transfer			
HOST	CAN_ID_WR_DATA	01	55
BOOTLOADER	CAN_ID_WR_DATA	01	01 // Command NOK

6.2.8 Special Commands

This command allows the user to read or write configuration bytes. A jump to a specified address is also available.

This operation can be executed only with a device previously opened in communication.

6.2.8.1 Prerequisite

Embedded Flash Controller Flash Mode Register (EFC_FMR) must be programmed correctly before using one of these commands (except for the Jump to a specified address command).

6.2.8.2 Request from Host

Identifier	Length	data[0]	data[1]	data[2]	data[3]	data[4]	Description
CAN_ID_SPC_CMD	2	0x00	0x00	-			Read NNB
			0x01				Read CRIS
			0x02				Read ABM
			0x03				Read PROPAG
			0x04				Read PHASE1
			0x05				Read PHASE2
			0x06				Read BRP
	3	0x1	0x00	Value	-		Write NNB
			0x01				Write CRIS
			0x02				Write ABM
			0x03				Write PROPAG
			0x04				Write PHASE1
			0x05				Write PHASE2
			0x06				Write BRP
	5	0x2	Address				Jump to a specified address

6.2.8.3 Answer from Bootloader

Answer to a Special Read Command:

Identifier	Length	data[0]	Description
CAN_ID_SPC_CMD	1	Value	Read the Configuration Byte

Answer to a Special Write Command:

Identifier	Length	data[0]	Description
CAN_ID_SPC_CMD	1	0x00	Command OK

Note: No answer is returned by the bootloader for a Jump Command.

6.2.8.4 Example

Read the CAN Re-locatable Identifier Segment (CRIS = 0x7F):

	Identifier	Length	data
HOST	CAN_ID_SPC_CMD	02	00 01
BOOTLOADER	CAN_ID_SPC_CMD	01	7F

Write the Node Number Value (NNB) value to 0xCA:

	Identifier	Length	data
HOST	CAN_ID_SPC_CMD	03	01 00 CA
BOOTLOADER	CAN_ID_SPC_CMD	01	00

Jump at address 0x100000:

	Identifier	Length	data
HOST	CAN_ID_SPC_CMD	05	02 00 10 00 00

7. Hardware and Software Constraints

- The first 4 Kbytes in the Flash memory are reserved for the CAN Bootloader Application (0x100_000 to 0x100_FFF).
- All the configuration bytes are located in the Flash memory starting at address 0x100_F00. Before connecting a CAN node in a point-to-point connection for the first time, the user must take care that their default values are 0xFF (See "Bootloader Configuration" on page 2.).
- SAM-BA *Boot4CAN* copies itself in the SRAM and uses a block of internal SRAM for variables and stacks. In consequence, the user area starts at address 0x202_000.
- Embedded Flash Controller Flash Mode Register (EFC_FMR) must be programmed correctly prior to using Special Read or Write Commands.

8. Revision History

Table 8-1. Revision History

Doc. Rev.	Comments	Change Request Ref.
6220A	First issue.	

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

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.

Printed on recycled paper.