Using a Real-time Timer with Non-calibrated RC Oscillator as a Real-time Clock

AT91 ARM® Thumb® Microcontrollers

Application Note

1. Introduction

The AT91SAM7S and AT91SAM7X microcontroller series embed a Slow Clock Oscillator and its RC couple. The RC oscillator is the only clock source supplying the Real-time Timer. Using the RC oscillator in the AT91SAM7S and AT91SAM7X series for the Real-time Timer (or other peripherals) results in a frequency drift greater than ±30%. For applications where the user would like to use the Real-time Timer as a Real-time Clock, this error becomes prohibitive.

The following Application Note describes a method to improve the accuracy of a Real-time Timer when it is used as a Real-time Clock. A special hardware process, "Timing Interval Acquisition" (TIA), has been developed and is embedded in AT91SAM7S and AT91SAM7X products.

6148A–ATARM–09-Dec-05

2. Terminology

Table 2-1.

Code	Description	Category
F_{RC-OSC}	Frequency of the embedded RC oscillator	Relative to AT91SAM7S/AT91SAM7X Embedded Oscillator
F_{MO}	Frequency of the embedded 3-20 MHz Oscillator called Main Oscillator	Relative to AT91SAM7S/AT91SAM7X Embedded Oscillator
TIA	Timing Interval Acquisition is the process which allows to count the number of event at Main Oscillator Frequency for 16 RC oscillator periods	Relative to internal process
RTT	Real-time Timer	Relative to internal Peripheral
PMC	Power Management Controller	Relative to internal Peripheral
PLL	Phase Lock Loop	Relative to analog cell
ppm	Part Per Million	Relative to unit
Thumb®	16-bit instruction set versus the 32-bit ARM® Mode	Relative to ARM processor

3. RC Oscillator and Frequency Error Sources

The working frequency spread of the RC oscillator can be equal to ±30% in AT91SAM7S and AT91SAM7X microcontrollers. The parameters affecting the resulting frequency are:

- manufacturing process spread
- supply voltage
- ambient temperature

3.1 Manufacturing Process Spread

Manufacturing process spread is the largest source of error. The inaccuracy due to this spread is equivalent to ±18% of the nominal working frequency. Although it is the largest source of error, it is not the most limiting as it is an intrinsic and static parameter and does not depend on application environment. This error is referred to as ε_{PROC}.

3.2 Frequency Drift and Supply Voltage

The change in frequency due to a change in RC oscillator voltage supply is estimated to be up to ±3% of the nominal working frequency.

There are two classes of drift:

- The absolute supply voltage level puts the RC oscillator at a specific working point. We are calling this error ε_{VOLT}.
- The supply noise adds a jitter component, or modulates the working frequency point around the previous point. In our application, the jitter component is not significant and can be disregarded. Only the average frequency is taken into account.

3.3 Temperature

Most oscillators display some sort of frequency variation with temperature. In the case of the RC oscillator, the drift parameter is estimated up to ±10% of the nominal working frequency.

The importance of temperature stability depends on the application. This is a dynamic parameter and has an effect on the average frequency of the RC oscillator. This error is referred to as ε_{TEMP}.

4. Improving RTT Frequency Accuracy

4.1 RTT and Real-time Clock Function

The functionality of an RTT may approach that of a Real-time Clock (e.g., one second event generation capability, continuous running mode, etc.) and there are software applications that use the RTT to count the elapsed time between a fixed time starting point and the present.

However, the condition for using an RTT as a Real-time Clock is that errors at source clock level must be minimized. To achieve this, most applications use a crystal oscillator due to its low frequency error, low cost and easy implementation.

The RTT in the AT91SAM7S and AT91SAM7X series uses on an on-chip RC oscillator supplying a free-running 32-bit counter via a programmable 16-bit divider as shown in Figure 4-1.

Figure 4-1. Real-time Timer Principle

The use of an RC oscillator in this configuration appears inappropriate since, by adding all stated error sources, it cannot be expected to provide levels of accuracy below ±30% (approximately), i.e., in the targeted application, ±30 seconds after 100 seconds. Generally, this level of accuracy is not acceptable in most applications.

4.2 Error Compensation using Timing Interval Acquisition (TIA) in the PMC

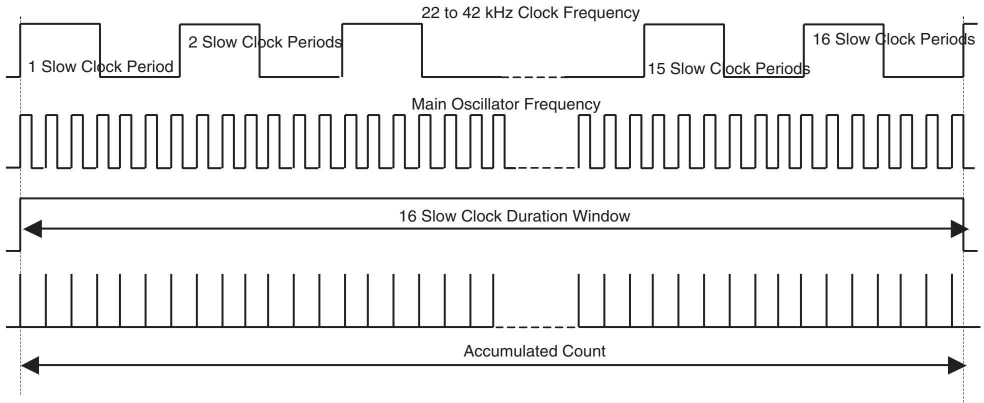
In the AT91SAM7S or AT91SAM7X products, the embedded Power Management Controller (PMC) comprises a specific state machine that directly relates the two microcontroller clock sources: the RC oscillator and the 3 - 20 MHz Main Oscillator.

Initially, with products containing a crystal-based low frequency oscillator and 3 - 20 MHz Main Oscillator, this state machine enables bootstrap software to determine crystal frequency used on Main Oscillator. In our application, a specific process is used to know the RC-Oscillator running frequency vs. known Main Oscillator frequency.

This process, called Timing Interval Acquisition (TIA), is started each time the main oscillator is enabled (refer to the datasheet "Power Management Controller>Main Clock Counter"). Timing Interval Acquisition consists of accumulating events at Main Clock frequency over 16 RC oscillator clock periods.

Using this process, it is possible to evaluate the duration of the window (see Figure 4-2) versus the Main Oscillator frequency and to establish a relationship between the two frequencies and, thus, between the two different time bases with different levels of accuracy.

Figure 4-2. Timing Interval Acquisition Relationship



4.3 Starting the TIA Process

As stated above, starting the process for TIA requires that the Main Oscillator is enabled: first, start the Main Oscillator at Power Management Controller level.

The cornerstone of this application note is based on the TIA result to correct the in front RTT clock divider to correct its value according to its real working frequency. The same argument could be applied to other peripherals that use the Slow Clock as clock source (Timer/Counter, Pulse Width Modulation). In all cases, the user must determine whether it is necessary to start the TIA process once or several times in his application. Two different cases can be established depending on environmental conditions:

- In the case of static temperature and regulated voltage, one TIA measurement is required after a power-on of the system to compensate for ε_{PROC}, ε_{VOLT} and ε_{TEMP}. (The frequency drift can be estimated as a constant static error valid over the entire application time (with stable supply voltage))
- In the case of dynamic temperature, ε_{TEMP} must be compensated for several times in parallel with the working application.

By default, all AT91SAM7S and AT91SAM7X products boot up from the slowest clock source which is, in this case, the RC oscillator. We are sure that the Main Oscillator has started in the application and a TIA result will be available. Therefore, in a static environment (static temperature, constant supply voltage), the TIA result can be used the whole time the application is running.

In other cases with a dynamic environment (temperature, supply voltage variations), the application is forced to restart the Main Oscillator in order to upgrade the TIA result. The environmental parameters that vary over time, thus introducing a new parameter. With this type of behavior, the user needs to make several TIA measurements while the application is running. To carry out a TIA requires clock switching, stopping and starting the Main Oscillator.

Several cases are possible in terms of clock switching to update the TIA. Table 4-1 gives cases according to the three initial master clock possibilities.

Table 4-1. TIA Measurement and Clock Switching

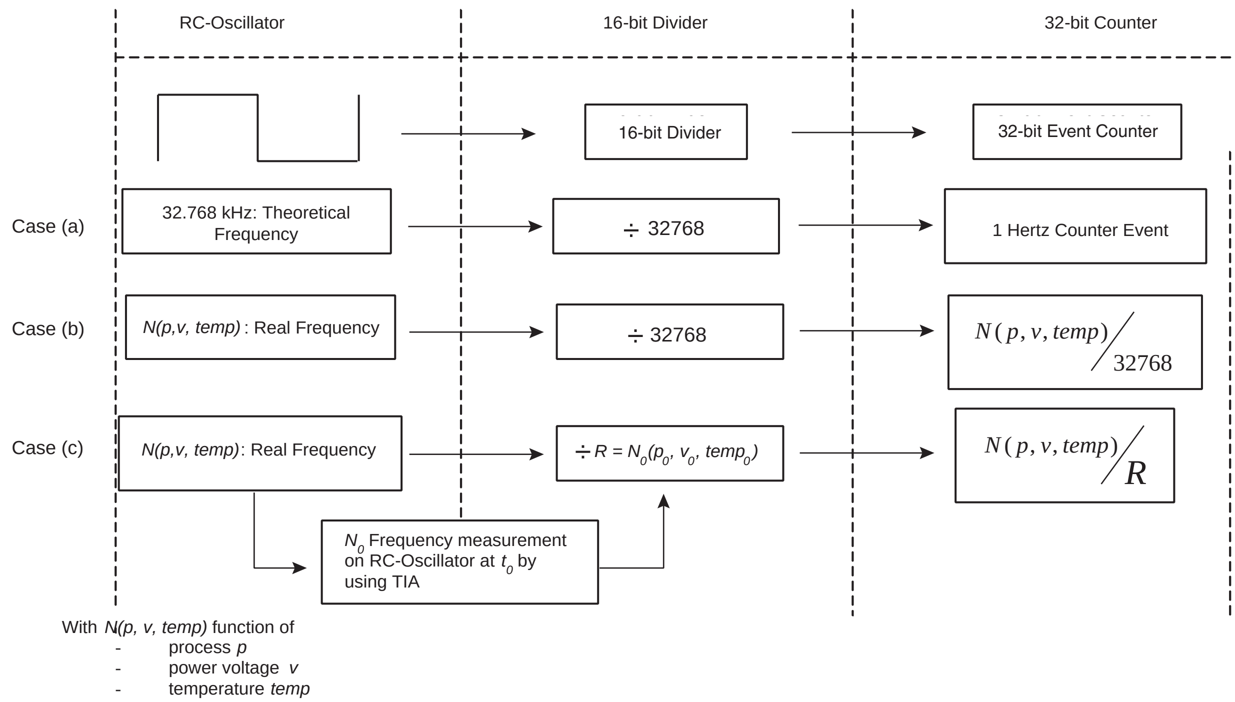
Current Master Clock Source	**Required Clock Switching Tasks for TIA Update**					**Time and Consumption Considerations**
RC oscillator	Main Oscillator Start/Stop					Time constraint: No lost time. Starting the Main Oscillator can be performed in parallel with the application Consumption: Main Oscillator current consumption
Main Oscillator	-> Switching to	RC oscillator	Main Oscillator Stop/Start	-> Switching to	Main Oscillator	Time Constraint: Clock switching, starting time of the Main Oscillator Consumption: No more than the application
PLL	-> Switching to	RC oscillator	Main Oscillator Stop/Start	-> Switching to	PLL Source	Time Constraint: Clock switching, starting time of the Main Oscillator and PLL Consumption: No more than the application

5. Compensation Process

5.1 From the RTT to the RTC

The Real-time Timer uses the RC oscillator clock as shown in Figure 4-1 on page 4. The TIA gives the number of Main Oscillator events during 16 RC oscillator cycles. In order to obtain 1 Hz after the 16-bit divider of the RTT, we need an R divider value equal to the RC oscillator frequency as shown Figure 5-1.

Figure 5-1. R Divider Setting



Case (a)

It is not realistic and represents the theoretical principle. In this case, the RC oscillator is running at nominal frequency 32,768 Hz. If its working frequency is known, it becomes easy to have a known and accurate time base via the RTT 16-bit divider.

Case (b)

This represents the actual setting without any compensation process. In this case, the result cannot be equal to 1 Hertz (or other known frequency) while N is unknown. With a constant denominator, the result depends on intrinsic and environment parameters since N is a function of these parameters.

Case (c)

The R divider value depends on N at a given time (t_0 in the example). As described at the beginning of the Application Note, process and voltage can be considered as static parameters. On the other hand, temperature is often a physical parameter depending on time. In this case, the user must adapt the sampling rate parameter R depending on N (temperature) variation vs. time.

5.2 Relationship between TIA Result and the RTT 16-bit Divider

In order to obtain 1 Hz frequency, the R divider value must be equal to the entry frequency supplied from the RC oscillator. The TIA measurement result allows us to compute this value.

The number of events E_{TIA} is equal to:

$$E_{TIA} = F_{MO} \times 16 \times T_{RC-OSC} \quad [1]$$

where:

- F_{MO} is the Main Oscillator frequency
- T_{RC-OSC} is the RC oscillator period

By replacing T_{RC-OSC} by $1/F_{RC-OSC}$ we deduce:

$$F_{RC-OSC} = \frac{F_{MO} \times 16}{E_{TIA}} = R \quad [2]$$

where F_{RC-OSC} is the RC oscillator frequency value.

The formula given as [2] provides the required value to set in the RTT frequency divider. The software requires only one 16-bit division. The user knows the Main Oscillator frequency and can perform F_{MO} x 16 by the pre-compiler in a define expression for example.

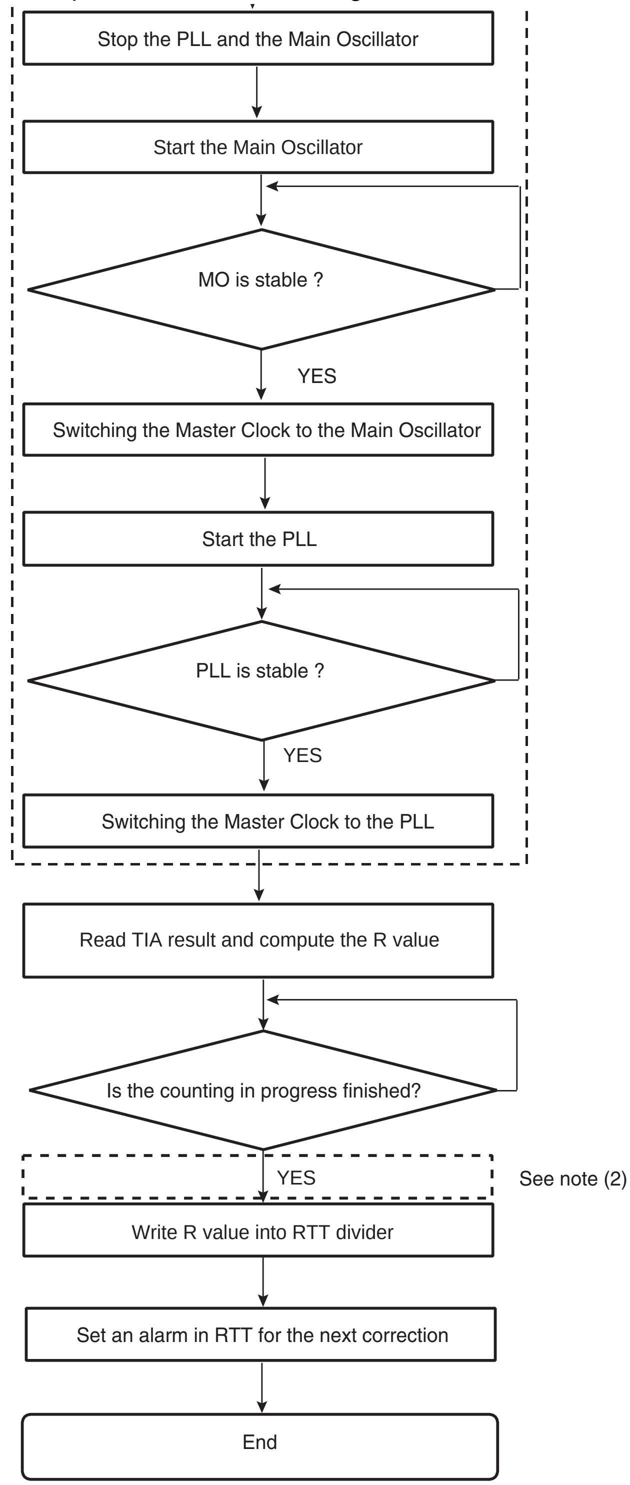
5.3 Compensation Procedure Algorithm

The full procedure requires three steps in the application:

- Start a TIA measurement
- Modify the divider value
- Set an alarm for the next compensation task

The first step is described in Section 4.3. Then the software must modify the RTT divider value using formula [2] above. At this level, the user is not free to modify the divider at any time, but must do so at the end of counting RTT cycles. We are going to show how modifying the RTT divider without taking certain precautions can generate other errors. The final step and the most difficult is to set the next TIA measurement depending on the RC temperature drift curve and maximum ambient temperature variation, whether measured or estimated.

Figure 5-2. Compensation Procedure Algorithm with PLL as Master Clock Source



Notes: 1. This step depends on application conditions. The switching shown in this flow chart is an example where the application is running from the PLL clock source. In other cases, refer to Table 4-1 on page 6.

2. This transition time has an impact on final accuracy error. The user must start this update at full speed and avoid modifying the divider value during a counting in progress.

The resulting compiled C function uses 88 Thumb® Instructions. From the code size, it is possible to obtain the necessary time required by this compensation task. Table 5-1 gives all cases and switching times.

Table 5-1. TIA Time Constraints

<table>
<tr><th>Current Master Clock Source</th><th colspan="5">Required Clock Switching Tasks for TIA Update</th><th>Time and Consumption Considerations</th></tr>
<tr><td>1. RC Oscillator</td><td colspan="2">Main Oscillator starting time depends on connected crystal (approx. 14.5 ms for a 3 MHz crystal)</td><td colspan="3">TIA Measurement
16 Slow Clock Cycles at 22 kHz in worst case (approx. 728 µs)</td><td>Time constraint: The TIA measurement is not preponderant but only the Main Oscillator start-up. Result: approx. 14.5 ms + 16 Slow Clock Cycles</td></tr>
<tr><td rowspan="2">2. Main Oscillator</td><td rowspan="2">Switching to RC oscillator 4 x Slow Clock</td><td rowspan="2">Stop Main Oscillator, restart Main Oscillator. Starting time depends on connected crystal (approx. 14.5 ms for a 3 MHz crystal)</td><td colspan="3">TIA Measurement
16 Slow Clock Cycles at 22 kHz in worst case (approx. 728 µs)</td><td rowspan="2">Time Constraint: As above, Main Oscillator starting time can be taken into account. Result: approx. 14.5 ms + 20 Slow Clock Cycles</td></tr>
<tr><td colspan="3">Switching to the Main Oscillator
4 x Slow Clock
TIA Measurement
16 Slow Clock Cycle at 22 kHz in worst case (approx. 728 µs)</td></tr>
<tr><td rowspan="2">3. PLL</td><td rowspan="2">Switching to RC oscillator 5 x Slow Clock</td><td rowspan="2">Stop Main Oscillator, restart Main Oscillator. Starting time depends on connected crystal (approx. 14.5 ms for a 3 MHz crystal)</td><td colspan="2">TIA Measurement
16 Slow Clock Cycles at 22 kHz in worst case (approx. 728 µs)
PLL Source</td><td rowspan="2">Switching to PLL 4 x Slow Clock</td><td rowspan="2">Time Constraint: Clock switching, Starting time of the Main oscillator, Result: approx. 14.5 ms + [13 + PLL Settling Time] Slow Clock Cycles</td></tr>
<tr><td>Switching to the Main Oscillator
4 x Slow Clock
TIA Measurement
16 Slow Clock Cycles at 22 kHz in worst case (approx. 728 µs)</td><td>PLL Start-up and Settling time waiting</td></tr>
</table>

5.4 Resulting Error

The resulting error of this method needs to be considered.

The goal is to use the accuracy of the Main Crystal Oscillator in order to evaluate the relative working frequency of the RC oscillator. By establishing a relationship between the RC oscillator and Main Oscillator, we can establish the RC working point for a given ε_{PROC}, ε_{VOLT}, and ε_{TEMP} after each TIA measurement.

In a perfect world (static temperature), the resulting error is attributed to the main oscillator frequency and TIA measurement errors:

- Absolute Main Oscillator error
- TIA absolute measurement error value can be equal to 2 missing Main Clock periods. The relative generated error depends on the working frequency of the Main and the RC Clock.
- The relationship between two frequencies is going to generate a rounded error due to the use of integer values instead of real number.

- The error due to the counting discontinuity gives rise to the RTT divider update shown on Figure 5-2 on page 9 and qualified in Note [(2)].

5.4.1 Main Oscillator Error

Main oscillator frequency depends on crystal specification depending on the hardware around it (fitted load capacitance versus specified, ambient temperature). The user must refer to the crystal manufacturer's specification. In general, this error is equal to ±50 ppm.

5.4.2 Timing Interval Acquisition

The TIA process is a state machine which samples main oscillator events for 16 RC oscillator cycles and asynchronously. At counting limits (at the beginning and the end), two master clock events can be missed in worst case. The absolute error depends on two parameters: the Main Oscillator frequency and RC frequency.

The number of main oscillator events for 16 slow clock periods can be expressed by:

$$E_{TIA} = 16 \times F_{MO} / F_{RC-OSC} \quad (3)$$

where:

$F_{RC\text{-}OSC}$ is the RC-Oscillator frequency value,

F_{MO} is the Main Oscillator Frequency,

E_{TIA} the number of Main Oscillator events.

The absolute error is equal to:

$$\varepsilon = 2 \times 10^6 / E_{TIA} \text{ in ppm} \quad (4)$$

The smaller E_{TIA}, the larger the absolute error. Thus a hardware constraint has been introduced into error calculation. To minimize this error, the user must implement the higher working frequency at main oscillator level. Table 5-2 gives some levels depending on the RC Oscillator frequency range and three main oscillator frequencies.

Table 5-2.

	$F_{RC\text{-}osc}$= 22 kHz	$F_{RC\text{-}osc}$= 30 kHz	$F_{RC\text{-}osc}$= 42 kHz	Unit
F_{MO} = 3 MHz	-916.6667	-1250	-1750	ppm
F_{MO} = 18.432 MHz	-149.197	-203.4505	-284.8307	
F_{MO} = 20 MHz	-137.5	-187.5	-262.5	

The measurement error is going to reduce the denominator of equation [2] on page 8, thus introducing an error by having a period higher than one second centered around ε/2.

5.4.3 Computing Error

The calculation of each new RTT divider may add a rounded one RTT step fewer errors. The calculation of the 16-bit real divider (see equation [2] on page 8) and the use of integer value introduces a rounded error when results are not integer. The maximum relative error is equal to one RTT period vs. one theoretical second.

Table 5-3.

$F_{RC\text{-}osc}$= 22 kHz	$F_{RC\text{-}osc}$= 30 kHz	$F_{RC\text{-}osc}$= 42 kHz	Unit
45.45	33.33	23.8	ppm

This error is centered close to half of maximum error.

5.4.4 Software Error

In Section 5.4.2, we showed how the TIA measurement error is dependent on main oscillator and RC oscillator frequencies and, as a result, introduces a hardware constraint in error calculation

There is also the case of the software that has to modify the RTT divider in the compensation process. If no precautions are taken at this level, errors can occur during the correction process: Updating the RTT divider before counting has completed can result in the maximum error equal to one second, and, in this case, resulting in an error more than the RC frequency inaccuracy equal to 100% error. Thus, the user must avoid modifying the divider value when the RTT counting is in progress.

Modifying the RTT divider in progress clears it and restarts counting.

For example, if the application is updating the RTT divider every 10 minutes and introducing a discontinuity when counting is equal to middle state (0.5 second), the introduced error is approximately equal to 800 ppm.

Note (2) on page 9 specifies that the application must wait for a new event of the RTT counter before performing the divider update and must have a minimum delay between the event detection and the RTT divider update in order to obtain the smallest discontinuity.

Thus, there are two constraints regarding the implementation of this process in an application:

- An interruption running mode instead of polling mode is required.
- The application must be running at main oscillator frequency at least, or, better, from the PLL output to minimize discontinuity counting for the RTT divider update.

6. Revision History

Table 6-1. Revision History

Doc. Rev.	Comments	Change Request Ref.
6148A	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

6148A–ATARM–09-Dec-05