

*Actel SmartFusion™ MSS RTC Driver
User's Guide*

Version 2.0

Actel Corporation, Mountain View, CA 94043

© 2010 Actel Corporation. All rights reserved.

Printed in the United States of America

Part Number: 50200248-0

Release: March 2010

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Actel.

Actel makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Actel assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Actel Corporation.

Trademarks

Actel, Actel Fusion, IGLOO, Libero, Pigeon Point, ProASIC, SmartFusion and the associated logos are trademarks or registered trademarks of Actel Corporation. All other trademarks and service marks are the property of their respective owners.

Table of Contents

Introduction.....	5
Features	5
Supported Hardware IP	5
Files Provided	7
Documentation	7
Driver Source Code.....	7
Example Code	7
Driver Deployment.....	9
Driver Configuration	11
Application Programming Interface	13
Theory of Operation.....	13
Types.....	14
Constant Values	14
Data Structures	14
Functions.....	14
Product Support.....	21
Customer Service	21
Actel Customer Technical Support Center.....	21
Actel Technical Support.....	21
Website	21
Contacting the Customer Technical Support Center	21

Introduction

The SmartFusion™ microcontroller subsystem (MSS) includes a real time counter (RTC) allowing configurable time delays using a low power 32kHz oscillator to support standby and sleep modes of operation. This software driver provides a set of functions for controlling the MSS RTC block as part of a bare metal system where no operating system is available. This driver can be adapted for use as part of an operating system but the implementation of the adaptation layer between this driver and the operating system's driver model is outside the scope of this driver.

Features

The MSS RTC driver provides the following features:

- Support for initializing and configuring the MSS RTC
- Support for stopping and starting RTC counter
- Support for setting and reading the RTC current value
- Support for setting the RTC match value
- Support for controlling RTC match interrupts

Supported Hardware IP

The MSS RTC bare metal driver can be used with Actel's MSS_RTC IP version 1.0.100 or higher included in the SmartFusion MSS.

Files Provided

The files provided as part of the MSS RTC driver fall into three main categories: documentation, driver source code, and example projects. The driver is distributed via the Actel Firmware Catalog, which provides access to the documentation for the driver, generates the driver's source files into an application project, and generates example projects that illustrate how to use the driver. The Actel Firmware Catalog is available from the Actel web site: www.actel.com/products/software/firmwarecat/default.aspx.

Documentation

The Actel Firmware Catalog provides access to these documents for the driver:

- User's guide (this document)
- A copy of the license agreement for the driver source code
- Release notes

Driver Source Code

The Actel Firmware Catalog generates the driver's source code into the *drivers\mss_rtc* subdirectory of the selected software project directory. The files making up the driver are detailed below.

`mss_rtc.h`

This header file contains the public application programming interface (API) of the MSS RTC software driver. This file should be included in any of your application's C source files that use the MSS RTC software driver.

`mss_rtc.c`

This C source file contains the implementation of the MSS RTC software driver.

Example Code

The Actel Firmware Catalog provides access to example projects illustrating the use of the driver. Each example project is self contained and is targeted at a specific processor and software toolchain combination. The example projects are targeted at the FPGA designs in the hardware development tutorials supplied with Actel's development boards. The tutorial designs may be found on the [Actel Development Kit](http://www.actel.com/products/hardware) web page (www.actel.com/products/hardware).

Driver Deployment

This driver is intended to be deployed from Actel Firmware Catalog into a software project by generating the driver's source files into the project directory. The driver uses the SmartFusion Cortex Microcontroller Software Interface Standard – Peripheral Access Layer (CMSIS-PAL) to access MSS hardware registers. You must ensure that the SmartFusion CMSIS-PAL is either included in the software toolchain used to build your project or is included in your project. The most up-to-date SmartFusion CMSIS-PAL files can be obtained using the Actel Firmware Catalog.

The following example shows the intended directory structure for a SoftConsole ARM® Cortex™-M3 project targeted at the SmartFusion MSS. This project uses the MSS RTC and MSS Watchdog drivers. Both of these drivers rely on SmartFusion CMSIS-PAL for accessing the hardware. The contents of the *drivers* directory result from generating the source files for each driver into the project. The contents of the *CMSIS* directory result from generating the source files for the SmartFusion CMSIS-PAL into the project.

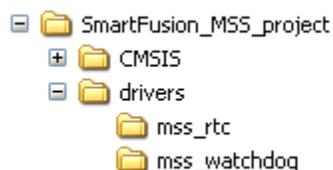


Figure 1 · SmartFusion MSS Project Example

Driver Configuration

The configuration of all features of the MSS RTC is covered by this driver.

The SmartFusion MSS configurator in the hardware design must be configured as follows, when using the RTC for SmartFusion sleep mode control:

- The SmartFusion analog block must be powered up by enabling the *ACE* block
- The SmartFusion internal voltage regulator must be powered up by configuring the *Reset Management* block to have the *Voltage Regulator output at power up* set to *On*.

When the SmartFusion device is put into sleep mode, an RTC match event can be configured to switch the SmartFusion internal voltage regulator back on. This powers up to the SmartFusion device and brings it back out of sleep mode.

The base address, register addresses and interrupt number assignment for the MSS RTC are defined as constants in the SmartFusion CMSIS-PAL. You must ensure that the SmartFusion CMSIS-PAL is either included in the software tool chain used to build your project or is included in your project.

Application Programming Interface

This section describes the driver's API. The functions and related data structures described in this section are used by the application programmer to control the MSS RTC peripheral from the user's application.

Theory of Operation

The MSS RTC driver functions are grouped into the following categories:

- Initialization and configuration
- Counter operations and match value control
- Interrupt Control

Initialization and Configuration

The MSS RTC driver is initialized through a call to the *MSS_RTC_init()* function. The *MSS_RTC_init()* function must be called before any other RTC driver functions can be called. The *MSS_RTC_init()* function does not reset the RTC hardware so that the RTC counter value is preserved when the system is initialized after SmartFusion resets as a result of coming out of sleep mode.

The SmartFusion RTC is configured through a call to the *MSS_RTC_configure()* function. This function is used to select:

- Whether or not the RTC counter value should be reset when a RTC match event occurs.
- Whether or not the SmartFusion internal voltage regulator should be powered back up when a RTC match event occurs.

Counter Operations and Match Value Control

The RTC counter can be started and stopped using the following functions:

- *MSS_RTC_start()*
- *MSS_RTC_stop()*

The RTC current value can be set or read using the following functions:

- *MSS_RTC_set_rtc_count()*
- *MSS_RTC_get_rtc_count()*
- *MSS_RTC_set_seconds_count()*
- *MSS_RTC_get_seconds_count()*

The RTC match value can be set using the following function:

- *MSS_RTC_set_rtc_match()*

Interrupt Control

The RTC interrupts are controlled using the following functions:

- *MSS_RTC_enable_irq()*
- *MSS_RTC_disable_irq()*
- *MSS_RTC_clear_irq()*

Types

There are no MSS RTC driver specific types.

Constant Values

MSS_RTC_NO_COUNTER_RESET

The `MSS_RTC_NO_COUNTER_RESET` constant is used to specify the *configuration* parameter to the `MSS_RTC_configure()` function. It is used to specify that the RTC count value must not be reset when an RTC match occurs.

MSS_RTC_RESET_COUNTER_ON_MATCH

The `MSS_RTC_RESET_COUNTER_ON_MATCH` constant is used to specify the *configuration* parameter to the `MSS_RTC_configure()` function. It is used to specify that the RTC count value must be reset when an RTC match occurs.

MSS_RTC_ENABLE_VOLTAGE_REGULATOR_ON_MATCH

The `MSS_RTC_ENABLE_VOLTAGE_REGULATOR_ON_MATCH` constant is used to specify the *configuration* parameter to the `MSS_RTC_configure()` function. It is used to specify that the SmartFusion voltage regulator must be enabled when an RTC match occurs. This is typically used to wake the Smartfusion device from sleep mode after a delay timed using the RTC.

Data Structures

There are no MSS RTC driver specific data structures.

Functions

MSS_RTC_init

Prototype

```
void MSS_RTC_init (void);
```

Description

The `MSS_RTC_init()` function initializes the RTC driver without resetting the RTC hardware. This function clears any pended RTC match interrupts. The `MSS_RTC_init()` function does not reset the RTC hardware so that the RTC counter value is preserved when the system is initialized after SmartFusion resets as a result of coming out of sleep mode.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

MSS_RTC_configure

Prototype

```
void MSS_RTC_configure  
(  
    uint8_t configuration  
);
```

Description

The *MSS_RTC_configure()* function configures the real time counter behavior. It is used to select whether the voltage regulator should be enabled or not when an RTC match occurs and whether the RTC counter should reset to zero when a RTC match occurs.

Parameters

configuration

The *configuration* parameter specifies the configuration to be applied to the RTC. It is a logical OR of the counter value behavior when a RTC match occurs and the effect of an RTC match on the voltage regulator.

The RTC count value can be reset or not when a RTC match occurs by using one of the following defined constants:

- MSS_RTC_NO_COUNTER_RESET
- MSS_RTC_RESET_COUNTER_ON_MATCH

The SmartFusion voltage regulator can be enabled when a RTC match occurs. This is typically used to wake the Smartfusion device from sleep mode after a delay timed using the RTC. The following defined constant can be used as parameter to enable this behavior:

- MSS_RTC_ENABLE_VOLTAGE_REGULATOR_ON_MATCH

Return Value

This function does not return a value.

Example

The following call to *MSS_RTC_configure()* will cause the RTC count to keep incrementing when an RTC match occurs. It will also cause the voltage regulator to become enabled causing SmartFusion to be powered back up if it was in sleep mode.

```
MSS_RTC_configure(MSS_RTC_NO_COUNTER_RESET | MSS_RTC_ENABLE_VOLTAGE_REGULATOR_ON_MATCH);
```

MSS_RTC_start

Prototype

```
void MSS_RTC_start (void);
```

Description

The *MSS_RTC_start()* function starts the RTC incrementing.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

MSS_RTC_stop

Prototype

```
void MSS_RTC_stop (void);
```

Description

The *MSS_RTC_stop()* function stops the RTC from incrementing.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

MSS_RTC_set_rtc_count

Prototype

```
void MSS_RTC_set_rtc_count  
(  
    uint64_t new_rtc_value  
);
```

Description

The *MSS_RTC_set_rtc_count()* function sets the current value of the RTC counter.

Parameters

new_rtc_value

The *new_rtc_value* parameter contains the 40 bit value that will be used as the new RTC counter value.

Return Value

This function does not return a value.

MSS_RTC_get_rtc_count

Prototype

```
uint64_t MSS_RTC_get_rtc_count (void);
```

Description

The *MSS_RTC_get_rtc_count()* function returns the current value of the RTC count.

Parameters

This function takes no parameters.

Return Value

This function returns the 40 bit current value of the RTC counter.

MSS_RTC_set_seconds_count

Prototype

```
void MSS_RTC_set_seconds_count  
(  
    uint32_t new_seconds_count  
);
```

Description

The *MSS_RTC_set_seconds_count()* function sets the current value of the RTC counter to the number of seconds passed as parameter.

Note: This function assumes a 32.768KHz external crystal.

Parameters

new_seconds_count

The *new_seconds_count* parameter contains the number of seconds that will be used to set the new current value of the RTC counter.

Return Value

This function does not return a value.

MSS_RTC_get_seconds_count

Prototype

```
uint32_t MSS_RTC_get_seconds_count (void);
```

Description

The *MSS_RTC_get_seconds_count()* function returns the number of seconds equivalent to the current RTC counter value.

Note: This function assumes a 32.768KHz external crystal.

Parameters

This function takes no parameters.

Return Value

This function returns the number of seconds equivalent to the RTC current value.

MSS_RTC_set_rtc_match

Prototype

```
void MSS_RTC_set_rtc_match  
(  
    uint64_t match_value  
);
```

Description

The *MSS_RTC_set_rtc_match()* function sets the match value of the RTC. The RTC match value is the value the RTC count must reach for the RTC to generate a RTC match event. The RTC event can be used to generate an RTC match interrupt or turn on the voltage regulator.

Note: You should ensure that the match value is greater than the current count value. Otherwise you may trigger an unintended RTC match event, as the new match value is written by 5 individual match register writes in the order MSByte to LSByte of the new match value.

Parameters

match_value

The *match_value* parameter contains the 40 bit value that will be used as the new RTC counter match value.

Return Value

This function does not return a value.

MSS_RTC_enable_irq

Prototype

```
void MSS_RTC_enable_irq (void);
```

Description

The *MSS_RTC_enable_irq()* function enables the RTC match interrupt (RTC_Match_IRQn) in the Cortex-M3 interrupt controller. The *RTC_Match_IRQHandler()* function will be called when an RTC match interrupt occurs.

Note: A *RTC_Match_IRQHandler()* default implementation is defined, with weak linkage, in the SmartFusion CMSIS-PAL. You must provide your own implementation of the *RTC_Match_IRQHandler()* function, that will override the default implementation, to suit your application.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

MSS_RTC_disable_irq

Prototype

```
void MSS_RTC_disable_irq (void);
```

Description

The *MSS_RTC_disable_irq()* function disables the RTC match interrupt (RTC_Match_IRQn) in the Cortex-M3 interrupt controller.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

MSS_RTC_clear_irq

Prototype

```
void MSS_RTC_clear_irq (void);
```

Description

The *MSS_RTC_clear_irq()* function clears pending RTC match interrupts. This function also clears the interrupt in the Cortex-M3 interrupt controller.

Note: You must call the *MSS_RTC_clear_irq()* function as part of your implementation of the *RTC_Match_IRQHandler()* RTC match interrupt service routine (ISR) in order to prevent the same interrupt event retriggering a call to the ISR.

Parameters

This function takes no parameters.

Return Value

This function does not return a value.

Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From Northeast and North Central U.S.A., call **650.318.4480**

From Southeast and Southwest U.S.A., call **650.318.4480**

From South Central U.S.A., call **650.318.4434**

From Northwest U.S.A., call **650.318.4434**

From Canada, call **650.318.4480**

From Europe, call **650.318.4252** or **+44 (0) 1276 401 500**

From Japan, call **650.318.4743**

From the rest of the world, call **650.318.4743**

Fax, from anywhere in the world **650.318.8044**

Actel Customer Technical Support Center

Actel staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Actel Technical Support

Visit the [Actel Customer Support website](http://www.actel.com/support/search/default.aspx) (<http://www.actel.com/support/search/default.aspx>) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website

You can browse a variety of technical and non-technical information on Actel's [home page](http://www.actel.com/), at <http://www.actel.com/>.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is tech@actel.com.

Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650 . 318 . 4460

800 . 262 . 1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. [Sales office listings](#) can be found at www.actel.com/company/contact/default.aspx.



Actel is the leader in low-power FPGAs and mixed-signal FPGAs and offers the most comprehensive portfolio of system and power management solutions. Power Matters. Learn more at <http://www.actel.com> .

Actel Corporation • 2061 Stierlin Court • Mountain View, CA 94043 • USA
Phone 650.318.4200 • Fax 650.318.4600 • Customer Service: 650.318.1010 • Customer Applications Center:
800.262.1060

Actel Europe Ltd. • River Court, Meadows Business Park • Station Approach, Blackwater • Camberley Surrey GU17
9AB • United Kingdom
Phone +44 (0) 1276 609 300 • Fax +44 (0) 1276 607 540

Actel Japan • EXOS Ebisu Building 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • <http://jp.actel.com>

Actel Hong Kong • Room 2107, China Resources Building • 26 Harbour Road • Wanchai • Hong Kong
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn