

# RX Family

R20AN0431EJ0100

Rev.1.00

Nov 30, 2016

## System Timer Module Firmware Integration Technology

### Introduction

This document explains the system timer module using CMT in RX Family MCUs. The system timer counts the year/month/day/hour/minute/seconds. And, this module also has the scheduler function. Users can be executed cyclic function by the system timer to register the function pointer with interval.

### Target Device

RX Family

### Contents

<b>1. Overview .....</b>	<b>2</b>
<b>1.1 File Structure .....</b>	<b>2</b>
<b>1.2 Development Environment.....</b>	<b>4</b>
<b>1.3 Compiler Options .....</b>	<b>4</b>
<b>1.4 ROM/RAM Size.....</b>	<b>4</b>
<b>1.5 Stack Size.....</b>	<b>4</b>
<b>1.6 Sections .....</b>	<b>4</b>
<b>2. API Information.....</b>	<b>5</b>
<b>3. API Functions .....</b>	<b>7</b>
<b>3.1 List of API Functions .....</b>	<b>7</b>
<b>4. Detailed Description of API Functions.....</b>	<b>8</b>

## 1. Overview

### 1.1 File Structure

This application note sample code includes following files.

Table1.1 File Structure 1

File/Directory(Bold) Names		Description
r20an0431ej0100_rx_middle.pdf		System timer module application note. (English)
<b>reference_documents</b>		Folder containing documentation on how to use the FIT module with various integrated development environments
	r01an1826jj0102_rx.pdf	How to use the FIT module with CubeSuite+
	r01an1723ju0111_rx.pdf	How to use the FIT module with e <sup>2</sup> studio
<b>FITModules</b>		FIT Module Folder
	r_sys_time_rx_v.1.00.zip	System timer module
	r_sys_time_rx_v.1.00.xml	System timer module e2 studio XML file for FIT configurator

The folder to which the contents of r\_sys\_timer\_rx\_v.1.00.zip is extracted will contain the files listed in table 1.2 below.

Table 1.2 File Structure

File/Directory(Bold) Names		Description
<b>r_config</b>		System timer config file folder
	r_sys_time_rx_config.h	System timer config file (default settings)
<b>r_sys_time_rx</b>		System timer FIT Module folder
<b>src</b>		System timer source code folder
	r_sys_time_rx.c	System timer source code
	r_sys_time_rx_private.h	System timer header file for internal
<b>doc</b>		System timer document folder
	<b>ja</b>	System timer document folder (Japanese)
	r20an0431jj0100_rx_middle.pdf	System timer application note (Japanese)
	<b>en</b>	System timer document folder (English)
	System timer application note (Japanese)	System timer application note (English)
<b>ref</b>		System timer config file (template) folder
	r_sys_time_rx_config_reference.h	System timer config file (template)
r_sys_time_rx_if.h		System timer header file
readme.txt		readme

## 1.2 Development Environment

System timer module was developed using the environment shown below. When developing your own applications, use the versions of the software indicated below, or newer.

1. Integrated development environment  
e<sup>2</sup> studio V5.2.0.020
2. C compiler  
CC-RX V2.05.00
3. Emulator/debugger  
E1/E20
4. Evaluation Board

Renesas Starter Kit for RX231 : R0K505231S000BE (others are OK)

We have confirmed test on e2 studio. Sample project is also generated by e2 studio.

Project conversion function can convert from e2 studio to CS+, if you detect compile error,  
please contact us.

## 1.3 Compiler Options

System timer module was developed using the following default compiler options:

-debug -nologo -isa=rxv2 -fpu -alias=noansi (exclude -include, -define)

Notice:

System timer module test is executed only little-endian. Please do not set the endian=big for compiler option.

## 1.4 ROM/RAM Size

ROM about 1.2KB, RAM about 0.2KB

## 1.5 Stack Size

108 byte (in executing R\_SYS\_TIME\_SetCurrentTime())

## 1.6 Sections

System timer module uses the default sections (P and C).

---

## 2. API Information

---

### 2.1 Hardware Requirements

---

System timer module is dependent on the CMT functionality built into the RX MCUs.

---

### 2.2 Software Requirements

---

System timer module is dependent on the following module:

- r\_bsp rev3.40 or later (BSP=Board Support Package)
- r\_cmt\_rx rev3.00 or later (CMT=Compare Match Timer)

---

### 2.3 Support Toolchain

---

The operation of the System timer module with the following toolchain has been confirmed.

RX Family C/C++ Compiler Package V2.05.00

---

### 2.4 Header File

---

All API calls and their supported interface definitions are contained in r\_sys\_time\_rx\_if.h.

---

### 2.5 Integer Types

---

This project uses ANSI C99.

---

### 2.6 Configuration

---

None

---

### 2.7 API Data Structure

---

All API data structure definitions are contained in r\_sys\_time\_rx\_if.h.

---

## 2.8 Return Values

---

```
typedef enum e_sys_time_err
{
    SYS_TIME_SUCCESS=0,           /* Normally terminated. */
    SYS_TIME_BAD_CHANNEL,        /* Non-existent channel number. */
    SYS_TIME_BAD_INTERVAL,       /* Bad interval parameter is specified. */
    SYS_TIME_BAD_TIME_OFFSET,    /* Bad time offset is set. */
    SYS_TIME_BAD_FUNCTION_POINTER, /* Bad function pointer is set. */
    SYS_TIME_BAD_SYS_TIME,       /* Bad system timer value is input */
    SYS_TIME_ALREADY_STARTED,    /* System timer is already started. */
    SYS_TIME_NOT_STARTED,        /* System timer is not started. */
    SYS_TIME_FULL_REGISTERED,     /* All register table is used. */
    SYS_TIME_ALREADY_REGISTERED,  /* Specified function pointer has been
                                already registered. */
};

sys_time_err_t;
```

---

## 2.9 Adding the Module

---

It is necessary to add this module to an existing e<sup>2</sup> studio project. If you use the e<sup>2</sup> studio plug-in to add the module, the install file path is updated automatically. For this reason, it is recommend that you use the plug-in to add the module to the project. In case of CS+, it is necessary to set the source code and include path and so on in manually.

For details, see Adding Firmware Integration Technology Modules to Projects  
(r01an1723eu0111\_rx.pdf/r01an1826ej0102\_rx.pdf).

### 3. API Functions

#### 3.1 List of API Functions

table 3-1 API

API	説明
R_SYS_TIME_Open	Open system timer module
R_SYS_TIME_GetCurrentTime	Get system time from system timer
R_SYS_TIME_SetCurrentTime	Set system timer to system timer
R_SYS_TIME_ConvertUnixTimeToSystemTime	Convert Unix time to system time format
R_SYS_TIME_RegisterPeriodicCallback	Register the cyclic function (max: 30)
R_SYS_TIME_UnregisterPeriodicCallback	Unregister the cyclic function
R_SYS_TIME_IsPeriodicCallbackRegistered	Confirm cyclic functions
R_SYS_TIME_Close	Close system timer module
R_SYS_TIME_GetVersion	Get the system timer version information

---

## 4. Detailed Description of API Functions

---

### R\_SYS\_TIME\_Open

---

#### Format

```
#include "r_sys_time_rx_if.h"

sys_time_err_t R_SYS_TIME_Open(void);
```

#### Parameters

なし

#### Return Values

SYS_TIME_SUCCESS	Normal terminated
SYS_TIME_BAD_CHANNEL	No CMT channel is exist
SYS_TIME_ALREADY_STARTED	Already started

#### Description

Open system timer module.

#### Reentrant

Yes



---

**R\_SYS\_TIME\_GetCurrentTime**

---

**Format**

```
#include "r_sys_time_rx_if.h"
```

```
sys_time_err_t R_SYS_TIME_GetCurrentTime(SYS_TIME *sys_time);
```

**Parameters**

sys_time	input/output	システムタイマからシステム時間を取得する領域
----------	--------------	------------------------

**Return Values**

SYS_TIME_SUCCESS	Normal terminated
------------------	-------------------

**Description**

Get the system time from system timer.

**Reentrant**

Yes

---

**R\_SYS\_TIME\_SetCurrentTime**

---

**Format**

```
#include "r_sys_time_rx_if.h"
```

```
sys_time_err_t R_SYS_TIME_SetCurrentTime(SYS_TIME *sys_time);
```

**Parameters**

sys_time	input/output	Area of system time setting to system time
----------	--------------	--

**Return Values**

SYS\_TIME\_SUCCESS

Normal terminated

SYS\_TIME\_BAD\_SYS\_TIME

Illegal system time is set

**Description**

Set the system time to the system timer. And, update Unix time is included into the system timer.

**Reentrant**

Yes

---

**R\_SYS\_TIME\_ConvertUnixTimeToSystemTime**

---

**Format**

```
#include "r_sys_time_rx_if.h"

sys_time_err_t R_SYS_TIME_ConvertUnixTimeToSystemTime(
    uint32_t unix_time, SYS_TIME *sys_time, uint8_t *time_offset);
```

**Parameters**

unix_time	input	Unix time
sys_time	input/output	Area of system time setting to system time
time_offset	input	String to specify the time zone

**Return Values**

SYS_TIME_SUCCESS	Normal terminated
SYS_TIME_BAD_TIME_OFFSET	Illegal time zone is set

**Description**

Set the system time to the system timer using Unix time. To set the string that represents the time zone to time\_offset. The strings that represents the time zone are defined into r\_sys\_time\_rx\_if.h.

**Reentrant**

Yes

---

## R\_SYS\_TIME\_RegisterPeriodicCallback

---

**Format**

```
#include "r_sys_time_rx_if.h"

sys_time_err_t R_SYS_TIME_RegisterPeriodicCallback(
    callback_from_sys_time_t function_pointer, uint32_t interval)
```

**Parameters**

function_pointer	input	Function pointer
interval	input	Cyclic interval (unit=10ms)

**Return Values**

SYS_TIME_SUCCESS	Normal terminated
SYS_TIME_BAD_FUNCTION_POINTER	Illegal function pointer is set
SYS_TIME_BAD_INTERVAL	Illegal interval is set
SYS_TIME_FULL_REGISTERED	Reach the upper limit of registration
SYS_TIME_ALREADY_REGISTERED	Known function pointer is specified

**Description**

Users can be executed cyclic function by the system timer to register the function pointer with interval. Max 30 functions can be registered. The function pointer is executed into CMT interrupt. This CMT interrupt enables interrupt. Please execute process that needs realtime into the interrupt function that has higher priority than CMT interrupt priority (can be set at r\_cmt\_rx\_config.h).

**Reentrant**

Yes

---

**R\_SYS\_TIME\_UnregisterPeriodicCallback**

---

**Format**

```
#include "r_sys_time_rx_if.h"
```

```
sys_time_err_t R_SYS_TIME_UnregisterPeriodicCallback(callback_from_sys_time_t function_pointer);
```

**Parameters**

function_pointer	input	Function pointer
------------------	-------	------------------

**Return Values**

SYS_TIME_SUCCESS	Normal terminated
SYS_TIME_BAD_FUNCTION_POINTER	Illegal function pointer is set

**Description**

Unregister the cyclic function.

**Reentrant**

Yes

---

**R\_SYS\_TIME\_IsPeriodicCallbackRegistered**

---

**Format**

```
#include "r_sys_time_rx_if.h"
```

```
bool R_SYS_TIME_IsPeriodicCallbackRegistered(callback_from_sys_time_t function_pointer);
```

**Parameters**

function_pointer	input	Function pointer
------------------	-------	------------------

**Return Values**

true	Already registered
false	Not registered yet

**Description**

Confirm the registration of cyclic function

**Reentrant**

Yes

---

**R\_SYS\_TIME\_Close**

---

**Format**

```
#include "r_sys_time_rx_if.h"

sys_time_err_t R_SYS_TIME_Close(void);
```

**Parameters**

なし

**Return Values**

SYS_TIME_SUCCESS	Normal terminated
SYS_TIME_BAD_CHANNEL	Failed CMT chanel close
SYS_TIME_NOT_STARTED	Not executed system timer yet

**Description**

Stop the system timer

**Reentrant**

Yes

**R\_SYS\_TIME\_GetVersion**

---

**Format**

```
#include "r_sys_time_rx_if.h"

uint32_t R_SYS_TIME_GetVersion(void);
```

**Parameters**

なし

**Return Values**

Version information about this module.

**Description**

The function returns the version of this module. The version number is encoded such that the top two bytes are the major version number and the bottom two bytes are the minor version number.

**Reentrant**

Yes



## Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/contact/>

All trademarks and registered trademarks are the property of their respective owners.

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	Nov 30, 2016	-	First Release.

## General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

### 1. Handling of Unused Pins

Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.  
In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.  
In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

### 5. Differences between Products

Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

- The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
  2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
  3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
  4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
  5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.  
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
  6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
  7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
  8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
  9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
  10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
  11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



### SALES OFFICES

### Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

#### **Renesas Electronics America Inc.**

2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

#### **Renesas Electronics Canada Limited**

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3  
Tel: +1-905-237-2004

#### **Renesas Electronics Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

#### **Renesas Electronics Europe GmbH**

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

#### **Renesas Electronics (China) Co., Ltd.**

Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### **Renesas Electronics (Shanghai) Co., Ltd.**

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

#### **Renesas Electronics Hong Kong Limited**

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2886-9022

#### **Renesas Electronics Taiwan Co., Ltd.**

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

#### **Renesas Electronics Singapore Pte. Ltd.**

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300

#### **Renesas Electronics Malaysia Sdn.Bhd.**

Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

#### **Renesas Electronics India Pvt. Ltd.**

No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India  
Tel: +91-80-67208700, Fax: +91-80-67208777

#### **Renesas Electronics Korea Co., Ltd.**

12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141