

RX Family

File driver for FTP server and Web server Module Firmware Integration Technology

R20AN0333EJ0102
Rev.1.02
Oct 01, 2016

Introduction

This Application Note explains about Filesystem interface called from Web server and FTP server for Embedded TCP/IP M3S-T4-Tiny.

Please refer to the following URL to know about details of FIT module.

<https://www.renesas.com/en-us/solutions/rx-applications/fit.html>

The FIT module provided with this application note requires the following FIT modules.

Table 1 FIT modules list

Function	Product	Website
Board Support Package	Board Support Package (BSP) (R01AN1685)	https://www.renesas.com/en-us/software/D3018117.html
TCP/IP (*1)	M3S-T4-Tiny(hereafter T4) (R20AN0051)	http://www.renesas.com/mw/t4
FTP server (*2)	FTP server using the embedded TCP/IP M3S-T4-Tiny Module (R20AN0078)	http://www.renesas.com/mw/t4
Web server (*2)	Web server using the embedded TCP/IP M3S-T4-Tiny Module (R20AN0075)	http://www.renesas.com/mw/t4
Ethernet interface (*1)	Interface conversion module for Ethernet Driver and Embedded system M3S-T4-Tiny (hereafter T4 Interface) (R20AN0311)	http://www.renesas.com/mw/t4
Ethernet driver (*1)	Ethernet Module (R01AN2009)	http://www.renesas.com/mw/t4
Timer (*1)	CMT Module (R01AN1856)	https://www.renesas.com/en-us/software/D6000038.html
File system (*3)	M3S-TFAT-Tiny(hereafter TFAT) (R20AN0038)	http://www.renesas.com/mw/tfat
Memory Driver Interface (*4)	TFAT Memory Driver Interface Module (R20AN0335)	http://www.renesas.com/mw/tfat
USB Driver (*4)	USB Basic Host and Peripheral Driver USB Host Mass Storage Class Driver	http://www.renesas.com/driver/usb

*1 Requires item.

*2 Requires at least 1 module.

*3 Requires when user uses external memory

*4 Please select for your environment.

Target Device

RX Family

Contents

1. Overview	3
1.1 This Application Note	3
1.2 Structure of application note	3
1.3 Structure of Software.....	4
2. API Information.....	5
3. API Functions	8
3.1 Data structure	9
3.2 change_dir	9
3.3 file_close	10
3.4 file_delete.....	10
3.5 file_open.....	11
3.6 file_read.....	11
3.7 file_rename	12
3.8 file_exist.....	12
3.9 file_write.....	13
3.10 get_file_info	13
3.11 get_file_list_info.....	14
3.12 get_file_size	15
3.13 make_dir.....	15
3.14 remove_dir	16
3.15 R_server_file_drv_drive_regist.....	16

1. Overview

1.1 This Application Note

This Application Note explains about Filesystem interface called from Web server and FTP server for Embedded TCP/IP M3S-T4-Tiny.

This Application Note provides the 2 pattern of FIT module. This 2 patterns can be configured.

- Pattern 1 - External memory: Use TFAT Library as filesystem.
- Pattern 2 - Internal memory: Do not use filesystem. Only support file control in current directory. This pattern for using only internal memory, ex internal RAM/ROM. The file list will be initialized in system start phase.

The APIs provided by this module does not need to call by user, because the APIs are called by Web/FTP server. If user would like to change drive number (default: 0), please call section 3.15 function.

When external memory is used, this module calls TFAT Library APIs, this module does not call Memory driver Initialize function.

1.2 Structure of application note

This application note includes files below.

Table 2 Structure of application note

file/folder name		description
r20an0333ej0102-rx-t4.pdf		Application note
reference_documents		
	r01an1723eu0111_rx.pdf	Adding Firmware Integration Technology Modules to e ² studio
	r01an1826ej0102_rx.pdf	Adding Firmware Integration Technology Modules to CS+
FITModules		
	r_t4_file_driver_v1.02.xml	FIT plug-in XML
	r_t4_file_driver_v1.02.zip	FIT plug-in ZIP
	configuration (r_config)	
	r_t4_file_driver_rx_config.h	configuration file(default)
	FIT Module (r_t4_file_driver_rx)	
	document(doc)	
	en	
	r20an0333ej0102-rx-t4.pdf	Application note (English)
	jp	
	r20an0333jj0102-rx-t4.pdf	Application note (Japanese)
	configuration refer reference (ref)	
	r_t4_file_driver_rx_config_reference.h	configuration file(template)
	source code(src)	
	readme (readme.txt)	Readme

1.3 Structure of Software

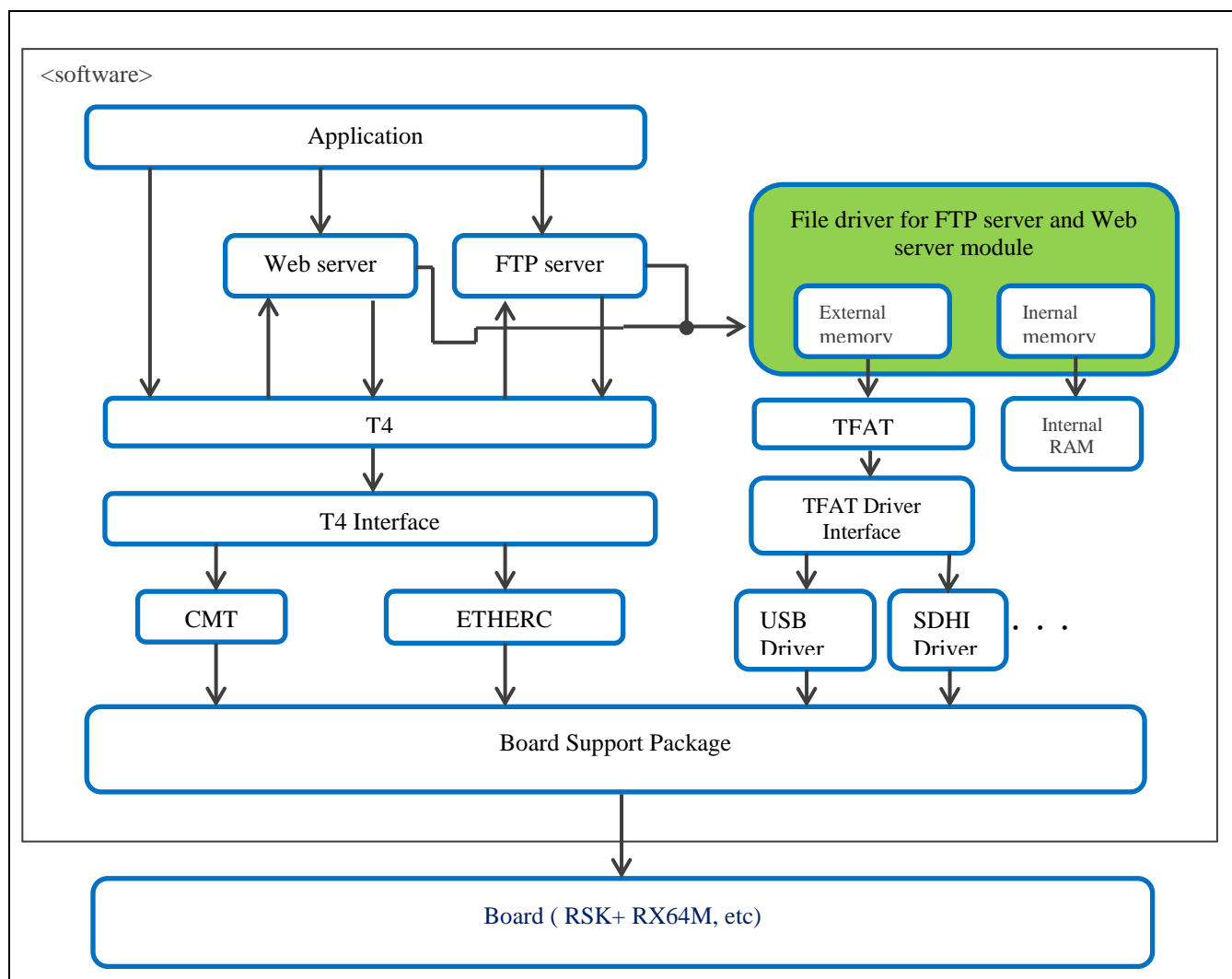


Figure 1 Structure of Software

Table 3 Using FIT Modules version

Product	Version
Board Support Package (BSP)	3.40
M3S-T4-Tiny(T4)	2.05
FTP server using the embedded TCP/IP M3S-T4-Tiny Module	1.04
Web server using the embedded TCP/IP M3S-T4-Tiny Module	1.05
Interface conversion module for Ethernet Driver and Embedded system M3S-T4-Tiny (T4 Interface)	1.05
Ethernet Module	1.11
CMT Module	3.00
M3S-TFAT-Tiny	3.03
TFAT Memory Driver Interface	1.03
USB Driver	
USB Basic Host and Peripheral Driver	1.11
USB Host Mass Storage Class Driver	1.11

2. API Information

2.1 Hardware Requirements

None.

2.2 Software Requirements

This FIT Module is dependent on the following packages:

- r_t4_rx
- r_tfat_rx
- r_t4_http_server_rx
- r_t4_ftp_server_rx
-

2.3 Supported Toolchain

This driver is tested and working with the following toolchains:

Renesas RXC Toolchain v.2.04.01

2.4 Header Files

All API calls are accessed by including a single file "r_t4_file_driver_rx_if.h" which is supplied with this software's project code.

Build-time configuration options are selected or defined in the file "r_t4_file_driver_rx_config.h "

2.5 Integer Types

This project uses ANSI C99 "Exact width integer types" in order to make the code clearer and more portable. These types are defined in stdint.h

2.6 Configuration Overview

The configuration options in this module are specified in `r_t4_file_driver_rx_config.h`. The option names and setting values are listed in the table below.

Table 4 Configuration options

Configuration options in <code>r_t4_file_driver_rx_config.h</code>	
<pre>#define SERVER_FILE_DRV_USE_EXTERNAL - Default value = (0)</pre>	When user uses external memory to store the data, please set the value "1".
<pre>#define SERVER_FILE_DRV_USE_INTERNAL - Default value = (0)</pre>	When user uses internal memory to store the data, please set the value "1". SERVER_FILE_DRV_USE_EXTERNAL has high priority than this macro.
<pre>#define SERVER_MAX_OPEN_FILE - Default value = (10)</pre>	The number of file open status. This means the number of file that opening in same time.
<pre>#define SERVER_DEF_CONTENTS - Default value = (100)</pre>	This macro is enable with SERVER_FILE_DRV_USE_INTERNAL. This macro defines number of max file contents.
<pre>#define SERVER_CONTENTS_LIST \ -Default value (none)</pre>	This macro is enable with SERVER_FILE_DRV_USE_INTERNAL. This macro defines the file list.
<pre>#define SERVER_MAX_CONTENTS_LIST - Default value = (0)</pre>	This macro is enable with SERVER_FILE_DRV_USE_INTERNAL. This macro defines max number of file list.
<pre>#define SERVER_CONTENTS_STORE_HEAD_ADR - Default value = (NULL)</pre>	This macro is enable with SERVER_FILE_DRV_USE_INTERNAL. This macro defines head of address to store the data that is requested from upper layer. Please allocate the area for this address.
<pre>#define SERVER_CONTENTS_STORE_MEM_SIZE - Default value = (0)</pre>	This macro is enable with SERVER_FILE_DRV_USE_INTERNAL. This macro defines max size to store the data that is requested from upper layer.

2.7 Arguments

```
/* DATE info.*/
typedef struct date_info_
{
    uint16_t year;                // 2011, 2012, ...
    uint8_t  month[4];            // Jan, Feb, Mar, ...
    uint8_t  day;                 // 1-31
    uint8_t  day_of_the_week[4];  // Sun, Mon, Tus, ...
    uint16_t hour;                // 0-23
    uint16_t min;                 // 0-59
    uint16_t sec;                 // 0-59
}DATE_INFO;

/* file list info.*/
typedef struct file_list_
{
    uint8_t file_name[13];        // file name
    uint32_t file_size;           // file size
    uint32_t file_attr;           // file attribute
    DATE_INFO date_info;          // date info.
}FILE_LIST;

/* contents info. */
typedef struct _contents
{
    uint8_t file_name[256];       // file name
    uint8_t *file_address;        // file data pointer
    int32_t file_size;            // file size
    uint32_t attr;                // date info
}CONTENTS;
```

2.8 Return Values

None

2.9 Adding The FIT Module to Your Project

Please refer to the Adding Firmware Integration Technology Modules to Projects (r01an1723eu0111_rx.pdf, for e2 studio) or the Adding Firmware Integration Technology Modules to CS+ Projects (r01an1826ej0102_rx.pdf).

3. API Functions

FTP server and Web server call these functions. These functions call lower layer functions according to configuration.

Table 5 Functions list

Function Name	Function Overview
change_dir()	Changes the working directory
file_close()	Closes a file
file_delete()	Deletes a file
file_open()	Opens a file
file_read()	Reads a file
file_rename()	Renames a file
file_exist()	Verifies that a file exists
file_write()	Writes to a file
get_file_info()	Acquires file information
get_file_list_info()	Acquires a file list
get_file_size()	Acquires a file's size
make_dir()	Creates a directory
remove_dir()	Deletes a directory
R_server_file_drv_drive_regist()	Change the drive number

3.1 Data structure

[Macro Definition]

```
#define FILE_WRITE      (0x10)
#define FILE_READ       (0x01)

/* File attribute bits for FILE_LIST->file_attr */
#define FILE_ATTR_RDO   0x01    /* Read only */
#define FILE_ATTR_HID   0x02    /* Hidden */
#define FILE_ATTR_SYS   0x04    /* System */
#define FILE_ATTR_VOL   0x08    /* Volume label */
#define FILE_ATTR_DIR   0x10    /* Directory */
#define FILE_ATTR_ARC   0x20    /* Archive */
```

3.2 change_dir

Description

This function sets current directory using specified argument. The argument specifies directory path in full path. Information of current directory is managed in each communication endpoint.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t change_dir(uint8_t *dir_path);
```

Parameters

dir_path	input	Pointer to directory path.
----------	-------	----------------------------

Return Value

-1	non-existent directory
0	Normal completion

Remark

This function calls lower layer function according to configuration. The lower layer function is change_dir_sub().

3.3 file_close

Description

This function closes the file corresponding to the ID specified by the argument and discards the file management information.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_close(int32_t file_id);
```

Parameters

file_id	input	ID value of the file to close
---------	-------	-------------------------------

Return Value

-1	Error
0	Normal completion

Remark

This function calls lower layer function according to configuration. The lower layer function is file_close_sub().

3.4 file_delete

Description

This function is to delete a file. The argument is specified in full path from root directory.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_delete(uint8_t *file_path);
```

Parameters

file_path	input	pointer to file path to delete
-----------	-------	--------------------------------

Return Value

-1	Error
0	Normal completion

Remark

This function calls lower layer function according to configuration. The lower layer function is file_delete_sub().

3.5 file_open

Description

This function opens the file specified in its argument in exclusive read mode and saves file management information independently. It also specifies an ID value for this file management information as the return value so that the web server can reference the saved file management information by ID. The saved file management information must be saved until this ID value is passed to the file close function.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_open(uint8_t *file_path, uint8_t mode_flag);
```

Parameters

file_path	input	pointer to file path to open
mode_flag	input	Mode value of file open (FILE_WRITE or FILE_READ)

Return Value

-1	Error
0 and positive integer	The ID value for the opened file

Remark

The file opened state must be maintained until the corresponding ID value is passed to the file close function.

This function calls lower layer function according to configuration. The lower layer function is

file_open_sub().

3.6 file_read

Description

This function reads the file corresponding to the ID value passed as an argument and advances the file pointer by the amount read. The file pointer is recorded in the file management information for each ID value and is maintained until the file close function is called.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_read(int32_t file_id, uint8_t *buf, int32_t read_size);
```

Parameters

file_id	input	ID value of the file to read
buf	output	Storage area for the file data read
read_size	input	Size of file to read

Return Value

-1	Error
0 and positive integer	Data size of receiving

Remark

This function calls lower layer function according to configuration. The lower layer function is file_read_sub().

3.7 file_rename

Description

This function renames the file specified first argument to second argument. These arguments are specified in full path from root directory.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_rename(uint8_t *old_name, uint8_t *new_name);
```

Parameters

old_name	input	pointer to target file name
new_name	input	pointer to after file name

Return Value

-1	Error
0	Normal completion

Remark

This function calls lower layer function according to configuration. The lower layer function is file_rename_sub().

3.8 file_exist

Description

This function verifies the file or directory existing. The argument is specified in full path from root directory.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_exist(uint8_t *file_path);
```

Parameters

file_path	input	Pointer to file or directory path
-----------	-------	-----------------------------------

Return Value

-1	Not exist
0	Exist

Remark

This function calls lower layer function according to configuration. The lower layer function is file_exist_sub().

3.9 file_write

Description

This function writes the file corresponding to the ID value passed as an argument and advances the file pointer by the amount write. The file pointer is recorded in the file management information for each ID value and is maintained until the file close function is called.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_write(int32_t file_id, uint8_t *buf, int32_t write_size);
```

Parameters

file_id	input	ID value of the file to write
buf	input	Storage area for the file data write
write_size	input	Size of the file to write

Return Value

-1	Error
0	Normal completion

Remark

This function calls lower layer function according to configuration. The lower layer function is file_write_sub().

3.10 get_file_info

Description

This function reads the file management information for the file corresponding to the ID value specified as an argument and writes the file date information to a date information structure.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t get_file_info(int32_t file_id, DATE_INFO *date_info);
```

Parameters

file_id	input	ID value of the file to read
date_info	output	pointer to information of date structure to store.

Return Value

-1	Error
0	Normal completion

Remark

This function calls lower layer function according to configuration. The lower layer function is get_file_info_sub().

3.11 get_file_list_info

Description

This function writes the file list stored at the directory path specified as an argument to a file list structure.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t get_file_list_info(uint8_t *dir_path, FILE_LIST *file_list, uint32_t num_file_list, int32_t read_index);
```

Parameters

dir_path	input	pointer to directory path to read
file_list	output	pointer to file list to store. This function stores '\0' to end of structure
num_file_list	input	Max number of file list to read at one time
read_index	input	Index of read starting

Return Value

-1	Error
0 and positive integer	Number of file

Remark

In case return value is smaller than num_file_list, it's the end of file list. In case return value is same value num_file_list, there is the data continuing. When this function needs continuing data, this function is called with 0 and positive integer with in read_index.

There are two cases. The argument "dir_path" has '/' termination and does not have. Please adjust for user file system.

This function calls lower layer function according to configuration. The lower layer function is get_file_info_sub().

3.12 get_file_size

Description

This function reads the file management information for the file corresponding to the ID value specified as an argument and returns the file size.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t get_file_size(int32_t file_id);
```

Parameters

file_id	input	ID value of the file to read
---------	-------	------------------------------

Return Value

-1	Error
0 and positive integer	File size

Remark

This function calls lower layer function according to configuration. The lower layer function is get_file_size_sub().

3.13 make_dir

Description

This function makes the directory. The argument is specified in full path from root directory.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t make_dir(uint8_t *dir_path);
```

Parameters

dir_path	input	pointer to file path to make
----------	-------	------------------------------

Return Value

-1	Error
0	Normal completion

Remark

There are two cases. The argument "dir_path" has '/' termination and does not have. Please adjust for user file system.

This function calls lower layer function according to configuration. The lower layer function is make_dir_sub().

3.14 remove_dir

Description

This function removes the directory. The argument is specified in full path from root directory.

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t remove_dir(uint8_t *dir_path);
```

Parameters

dir_path	input	pointer to file path to remove.
----------	-------	---------------------------------

Return Value

-1	Error
0	Normal completion

Remark

There are two cases. The argument "dir_path" has '/' termination and does not have. Please adjust for user file system.

This function calls lower layer function according to configuration. The lower layer function is remove_dir_sub().

3.15 R_server_file_drv_drive_regist

Description

This function controls filesystem using specified argument.

Usage

```
#include <stdint.h>
int32_t R_server_file_drv_drive_regist(uint8_t drive);
```

Parameters

drive	input	drive number
-------	-------	--------------

Return Value

-1	Error
0	Normal completion

Remark

Drive number initial value is "0". Please call this function to change the Drive number.

Please initialize changed drive for the TFAT Library.

Please do not call this API during other APIs processing.

This function calls lower layer function according to configuration. The lower layer function is

R_server_file_drv_drive_regist_sub ().

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
1.02	Oct 01, 2016	-	Updated the xml file for FIT
1.01	Jan 01, 2015	-	Added Support MCUs.
1.00	Dec 01, 2014	-	First edition issued

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