RE01 1500KB Group

PMSC Firmware Updater

R01AN4946EJ0100
Rev.1.00
Sep 30, 2019

Overview

This documentation describes PMSC(Peripheral Mass Storage Class) firmware updater using RE MCU.

Target Devices

RE01 1500KB Group

Table of Contents

1. Document Outline

This documentation explains abount the firmware updater(PMSC firmware updater) used the peripheral mass storage class (PMSC).

1.1 Functions

When a board that operates as a PMSC firmware updater is connected to a PC, it is recognized as mass storage and a window is displayed. Copying the BIN format file (user program) to the displayed window will rewrite the FlashROM in the MCU and update the firmware.

1.2 List of Abbreviations and Acronyms

The following lists terms and abbreviations used in this document.

PMSC	:	Peripheral Mass Storage Class
USB	:	Universal Serial Bus

2. Overview of PMSC Firmware Updater

2.1 Overview

When a board operating as MSC (Mass Storage Class) is connected to a PC, a window with the Flash ROM in the MCU as the media area is displayed on the PC. If you drag and drop BIN format file (User program) to this window, the BIN file data will be sent to the board via USB (Mass Storage Class). This updater receives the BIN format file sent from the PC and writes the data to the internal Flash ROM. When the board is reset after the Flash ROM writing is completed, the written BIN file (User program) is executed on the board.

The following shows this updater's data flow.

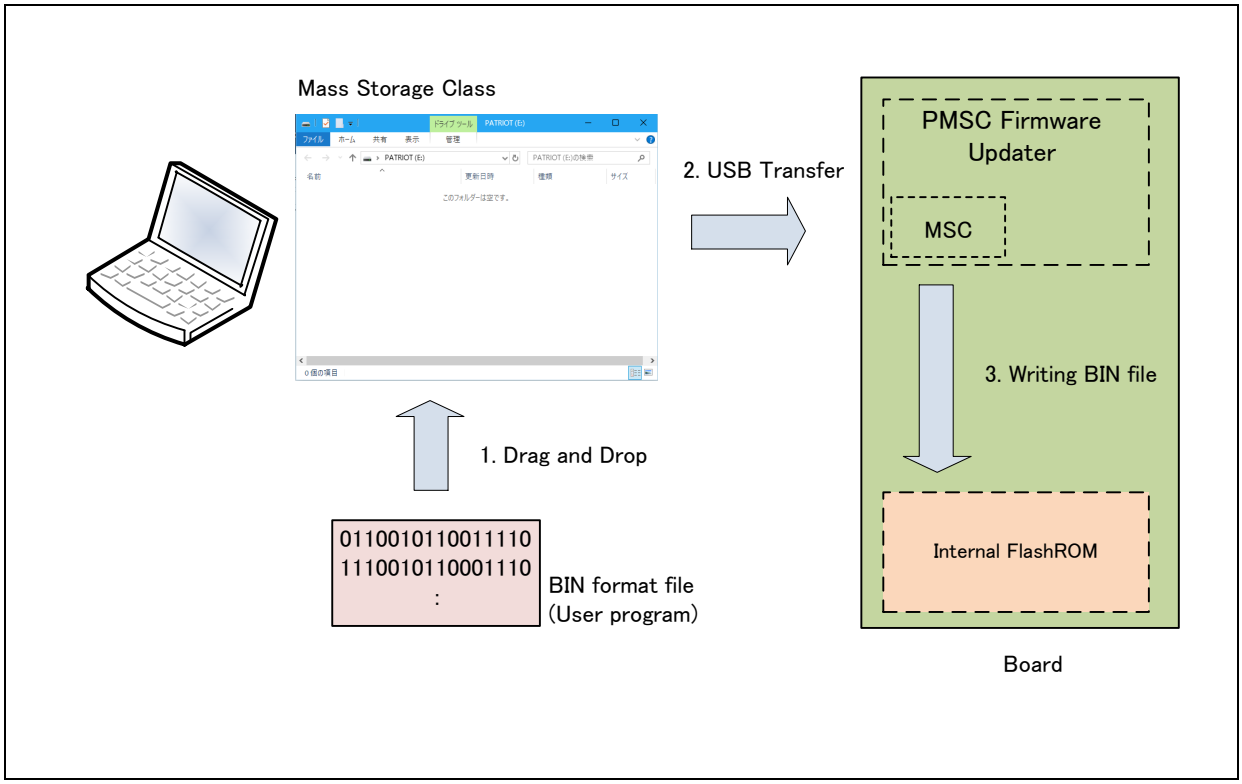

Figure 2-1 PMSC Firmware Updater Flow

2.2 Features

This program has the following features.

1. This program is operated in USB Peripheral (Function) mode.
2. This program supports the following USB specifications:
   (1). USB 2.0 specification, Full-speed transfer
   (2). USB mass storage device class
   (3). SFF-8070i mass storage subclass
3. This program erases the internal flash ROM and writes the user program (mot file) to it.
4. The user program can be allocated to the entire internal flash ROM area, except for the area where this program itself is located.
5. This user program may use all interrupts.

3. Software Structure

Figure 3-1 shows the software configuration of PMSC firmware updater.

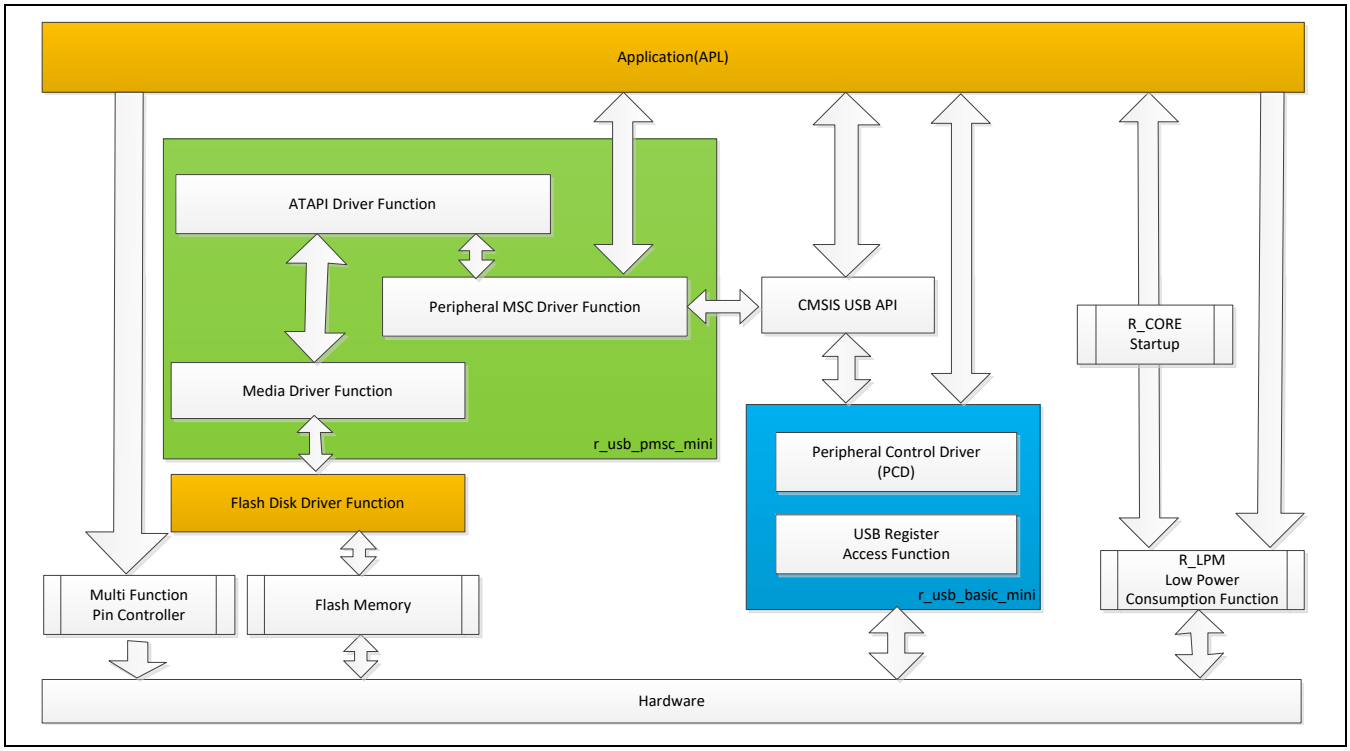

Figure 3-1 Software Structure

4. Operation Confirmation Environment

Operations for this program have been confirmed under the following environment:

1. IAR

Integrated Development Environment (IDE)	:	IAR Embedded Workbench for ARM 8.32.3.20228
Compiler	:	IAR C/C++ Compiler for ARM 8.32.3.193 (8.32.3.193)
Evaluation Board	:	Evaluation Kit RE01 1500KB
Host PC	:	Windows 10
Flash Programming Tool	:	Renesas Flash Programmer (RFP)

2. GCC

Integrated Development Environment (IDE)	:	e2 studio V.7.6.0
Compiler	:	GCC ARM Embeded 6.3.1.20170620
Evaluation Board	:	Evaluation Kit RE01 1500KB
Host PC	:	Windows 10
Flash Programming Tool	:	Renesas Flash Programmer (RFP)

5. Setup

5.1 Project Setup

1. IAR

   (1). The following shows the setup procedure of PMSC firmware updater.

   (2). Start up *EWARM*.

   (3). Select [**File**] → [**Open Workspace**]

   (4). Select the EWW file (usb_downloader.eww) for PMSC firmware updater.

   (5). Generate the binary target program by clicking the [Make] button.

   (6). Check whether the *usb_downloader.mot* file is geneted or not in *Exe* folder. If this *mot* file is generated, the project setup completes.

2. GCC

   (1). The following shows the setup procedure of PMSC firmware updater.

   (2). Start up *e² studio*.

   (3). Select [**File**] → [**Import**].

   (4). Select [**General**] → [**Existing Projects into Workspace**].

   (5). Select "**Select root directly:**".

   (6). Input the folder containing the ".cproject" file.

   (7). Click "Finish" button.

   (8). Generate the binary target program by clicking the "Build" button.

5.2 Board(Evaluation Kit RE01 1500KB) Setup

It is necessary to set the jumper on the board to operate in the peripheral (function) mode. Please refer to the following.

Jumper Setting : J2 Shorted Pin2-3

6. Writing PMSC Firmware Updater to the internal Flash ROM

(1). Hardware setup

The following figures show connection diagrams for writing **PMSC Firmware Updater** to the MCU.

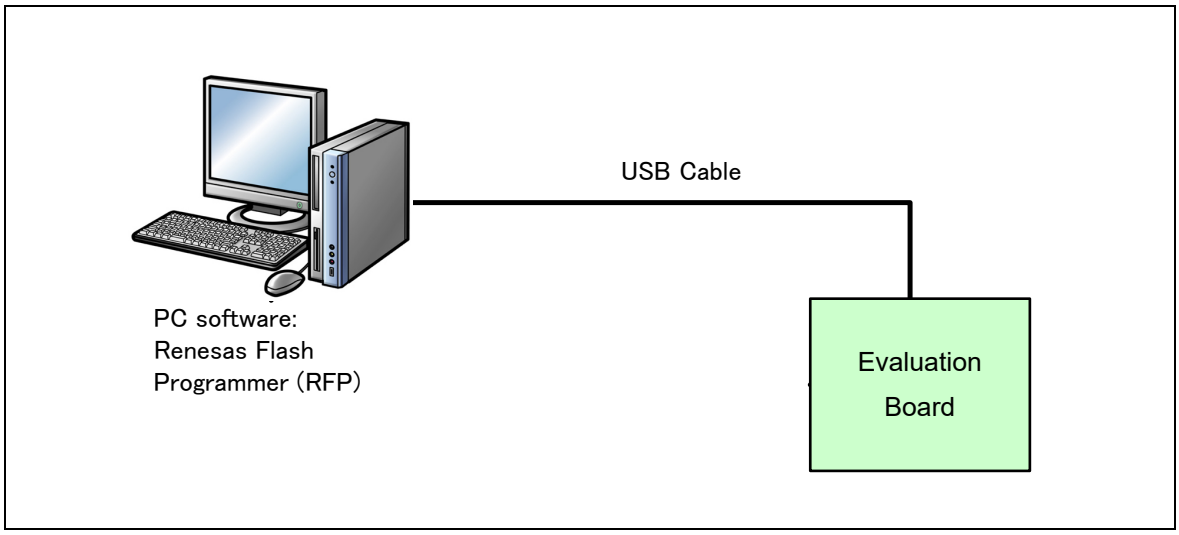

Figure 6-1 Connection Diagram

(2). Writing the **PMSC Firmware Updater.**

Run the Renesas Flash Programmer (RFP) and, using the [**Browse]** for [Program File] button, select PMSC Firmware Updater file to be written from the Workspace/(MCU *name*) folder. Press **Start** to download the program to the evaluation board. The write operation is complete when **OK** is displayed.

Notes:

a) Refer to the following URLs for more details on the Renesas Flash Programmer:
URL:
https://www.renesas.com/en-us/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html

b) Refer to section **8, Allocation of** for more details concering positioning of **PMSC Firmware Updater**.

c) Please make settings to erase all blocks in the internal Flash ROM (program ROM Area).

7. Operation Procedure of PMSC Firmware Updater

The following shows the operation procedure of **PMSC Firmware Updater.**

1. After starting up the board (Evaluation Kit RE01 1500KB), press the RESET button while holding down switch SW3 on the board.
2. Connect the board to the PC (USB Host), the board is recognized as the mass storage and the window is displayed.
3. Drag and Drop the BIN format file (User Program) to the above window.
4. Press RESET button on the board after the above drag and drop is complete. BIN file (User Program) is executed on the board.

8. Allocation of PMSC Firmware Updater

Allocate **PMSC Firmware Updater** in the following area.

Alloation Areas for PMSC Firmware Updater		
0x00000000	-	0x0000FFFF

The following is the memory map. For more details, refer to the user's hardware manual corresponding to the target MCU.

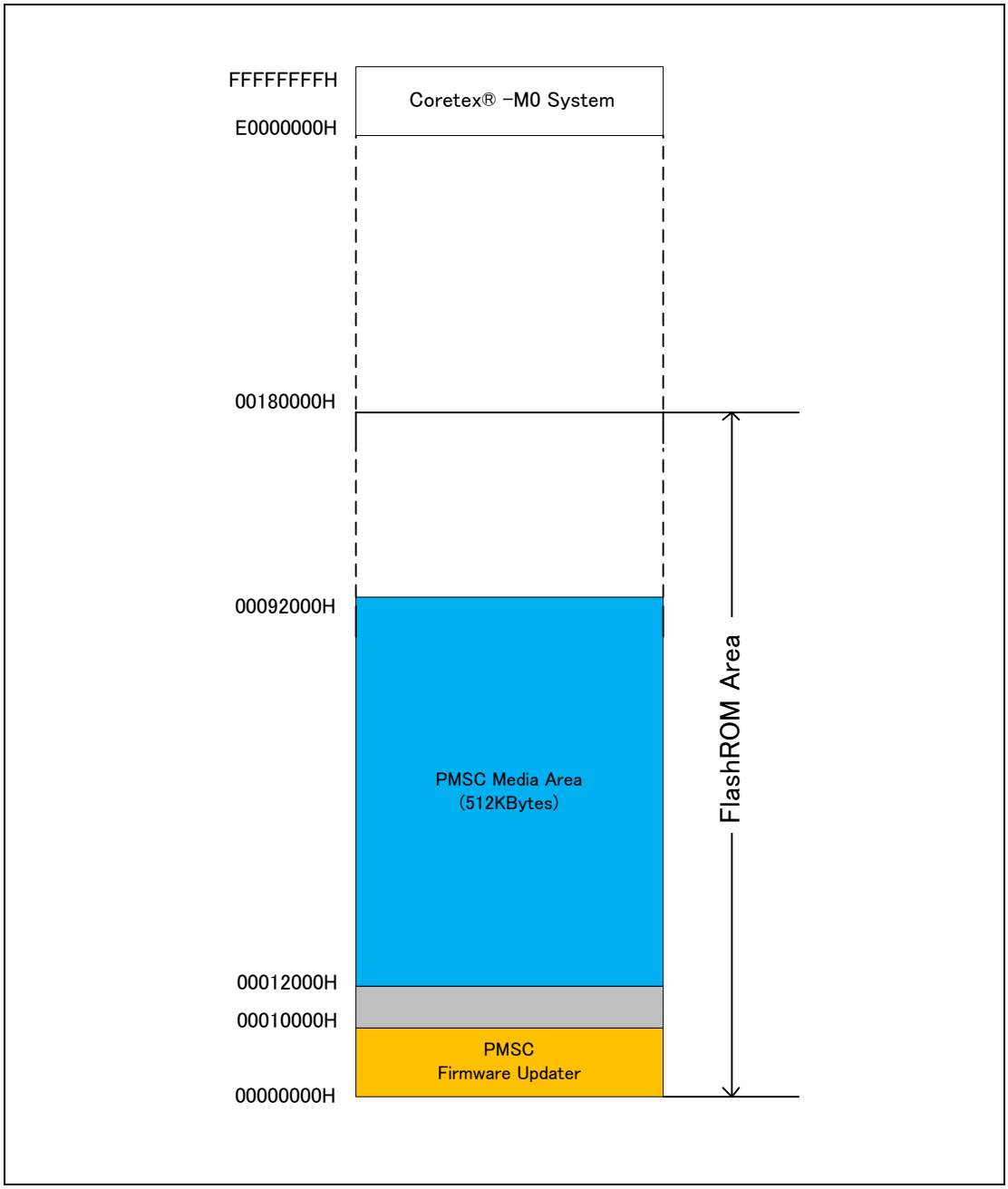

Figure 8-1 Memory Map

Note:

Please set the media area settings to the following definitions in the "*usb_media¥flash_disk¥r_flash_disk.h*" file.

USB_MEDIA_ADDRESS definition	Media Start Address
STRG_MEDIASIZE definition	Media Size

Setting Example)

```
#define     USB_MEDIA_ADDRESS       (0x00012000ul)
#define     STRG_MEDIASIZE          (512 * 1024ul)
```

9. API

The following shows the API supported for **PMSC Firmware Updater.**

API	Description
R_USB_PmscInitialize	PMSC device class initialization.
R_USB_PmscCbwReceive	Reception request of Command Block Wrapper (CBW).
R_USB_PmscCswPipeStall	CSW (Command Status Wrapper) transmission request with STALL set in CSW Status.
R_USB_PmscBotProcess	The following processing is performed. 1. ATAPI commamd analysis 2. PMSC data transfer request 3. PMSC data transfer complete 4. CSW trasnmission request 5. CSW transmission complete 6. CBW reception request

Note:

The APIs other than the above are also used in **PMSC Firmware Updater**. For these APIs, refer to the following URL.

http://arm-software.github.io/CMSIS_5/Driver/html/group__usb__interface__gr.html

10. Power On / Reset Operation Flow

This section explains the operation flow after power on or reset for **PMSC Firmware Updater**.

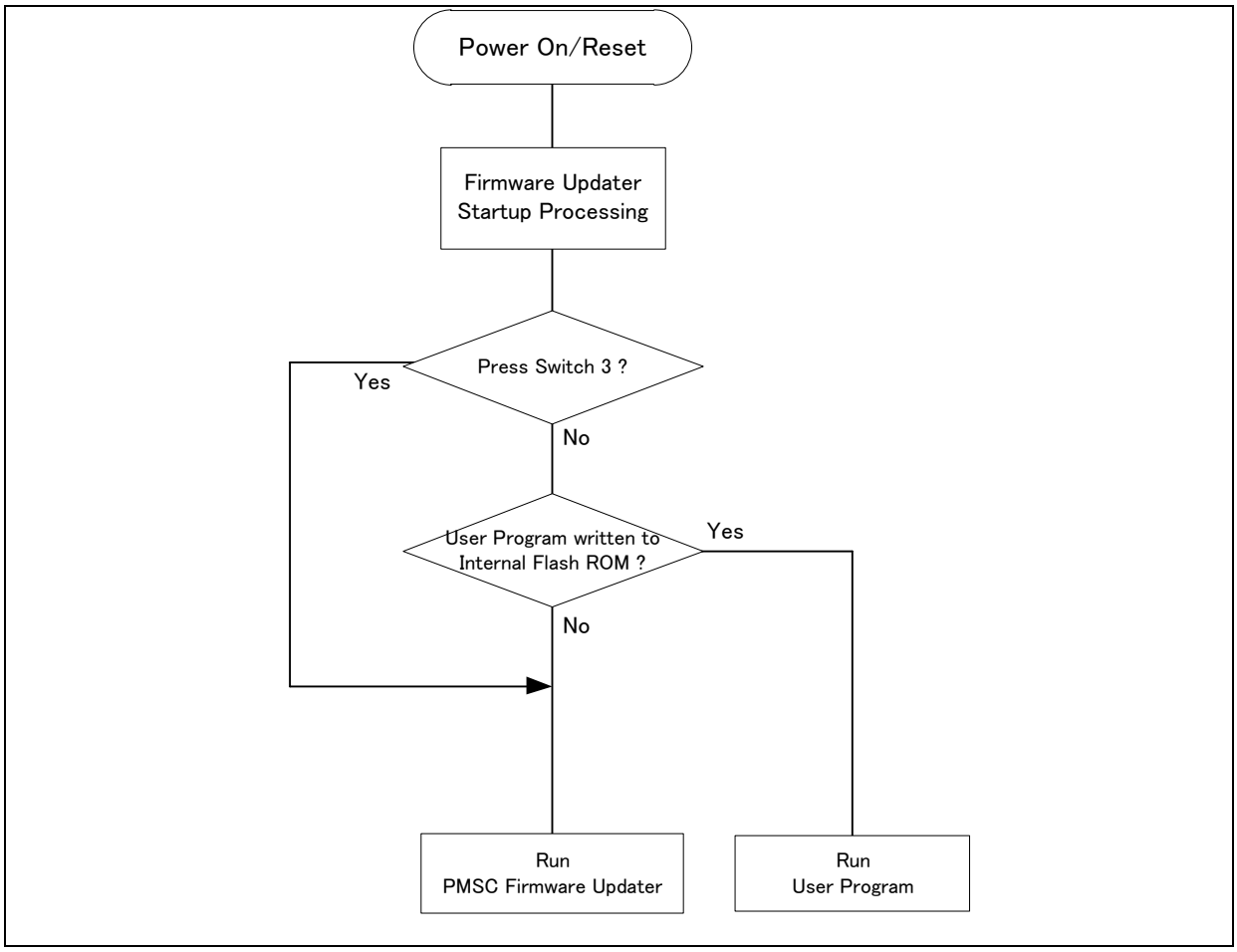

Figure 10-1 Power On / Reset Operation Flow

Note:

When running the **PMSC Firmware Updater** when the user program (BIN file) is written to the internal flash ROM, press the RESET button while pressing switch 3 on the board (Evaluation Kit RE01 1500KB).

11. Cautions Regarding Creating BIN File (User Program)

1. Do not make any settings to the option setting memory in BIN file (User program).
2. Do not include fixed vector area in BIN file (User program).
3. Allocate the startup program to the start area of the BIN file (user program).

12. Limitation

1. Do not copy more than one BIN file (User program) to the PMSC media area.
2. If the evaluation board starting as **PMSC Firmware Updater** is reconnected after copying the BIN file to PMSC media area and running User program, the BIN file is not displayed on the Window on PC.
3. The BIN format is the only executable file format that can be copied to the PMSC media area. The other file format is not supported.
4. The DMA and DTC transfer are not supported.

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sep 30, 2019	-	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.