Flash Protection Schemes and Methods in Renesas NOR Flash Products

This document describes the different types of protections available on most Renesas flash devices. It also details the ways to implement them, as well as what the consequences of the implementations are.

Contents

1. Overview

1.1 Protected Resources

All Renesas flash products implement multiple methods for protecting various on-chip resources from intentional or accidental modification. Examples of protected resources are:

- The memory array or parts of it
- Status registers
- Flash state/mode
- Flash reset

1.1.1 Memory Array Write Protection

Two types of protection methods exist for protecting the memory array. Their objective is to prevent changing the contents of parts of the memory which the user wants to protect. In other words, they block erase and program operations in certain memory sectors or blocks.

1.1.2 Status Register Protection

Protection methods for Status Registers prevent the user from writing to those registers and changing their values. Note that in many cases certain bits of the Status Registers configure the protection state of the memory array; thus, protecting the Status Registers can also indirectly protect the memory array.

1.1.3 Flash State/Mode/Reset Protection

Certain methods exist to protect from accidentally sending commands that change the state of the flash, or otherwise reset the flash.

1.2 Protection Types

1.2.1 Hardware Protection

Hardware protection uses a flash input pin named $\overline{\text{WP}}$ (Write Protect) to signal the flash chip to protect or unprotect certain resources.

1.2.2 Software Protection

Software protection uses flash commands to protect or unprotect certain resources. Examples are:

- Writing to Status Register protection bits that control the protection state of the memory array.
- Protect/unprotect commands (sometimes called lock/unlock) that change the protection state of certain sectors/blocks of the memory array, or the entire memory array.
- Mandatory write-enable command (or reset-enable command in the case of reset) before sending another command that makes a change to a certain resource (memory array, register, flash state). This extra step protects from accidental or unintentional changes.

2. Status Register Protection

All Renesas NOR flash products use the $\overline{\text{WP}}$ pin for hardware protection for the Status Registers. Under a certain configuration, the $\overline{\text{WP}}$ pin can be driven low by the host to block any write operation to the Status Registers. That configuration is determined by one or more bits in the Status Register. In other words, to enable Status Register protection, the host must first write to Status Registers, which makes the $\overline{\text{WP}}$ pin protection effective.

As mentioned above, protecting Status Registers can indirectly protect the memory array. This is because when writes to Status Registers are blocked, the protection state of the memory array cannot be changed.

Note: In products that support quad-SPI interface, the $\overline{\text{WP}}$ pin changes its function to IO_2, one of four parallel I/O pins in quad-SPI communication. In those products, the Quad Enable (QE) bit determines the pin's function. If QE is set to 1, quad-SPI is enabled, the $\overline{\text{WP}}$ becomes IO_2 and should not be used for write-protection.

Figure 1 shows a typical pinout of a NOR flash device. Pin 3 is the $\overline{\text{WP}}$ pin.

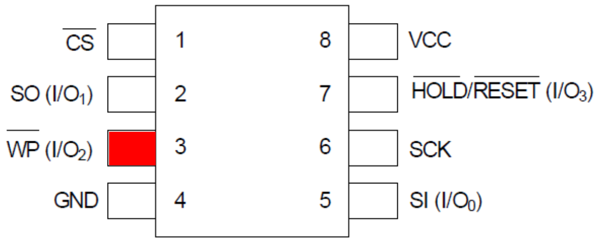

$\overline{\text{CS}}$
1
8
VCC
SO (I/O_1)
2
7
$\overline{\text{HOLD}}/\overline{\text{RESET}}$ (I/O_3)
$\overline{\text{WP}}$ (I/O_2)
3
6
SCK
GND
4
5
SI (I/O_0)

Figure 1. NOR Flash Device Typical Pinout

2.1 Enabling $\overline{\text{WP/}}$ Pin Hardware Protection

To enable the hardware write-protection function of the $\overline{\text{WP}}$ pin, certain bits of the Status Registers must be set.

In some products (AT25XE/AT25FF/AT25SF/AT25SL/AT25EU), the SRP0 bit (Status Register 1, bit 7) must be set to 1, and the SRP1 bit (Status Register 2, bit 0) must be set to 0; this enables $\overline{\text{WP}}$ write protection. Note that other settings of these bits typically enable other protections of the Status Registers, such as reset lockdown (blocks write to the Status Registers until the next reset) and permanent lockdown. See the product datasheet for specific details.

In other products (AT25DF/AT25DL) the SPRL bit (Status Register 1 bit 7) must be set to enable Status Register hardware protection by the $\overline{\text{WP}}$ pin. Note that in these products, the $\overline{\text{WP}}$ pin protects not only the Status Registers but also the separate lock bits that are described in Section 3. The datasheets provide more details.

An earlier version of the SPRL bit exists in the AT25DN family, where it is called BPL.

3. Memory Array Protection

In general, the Renesas NOR flash products support one of two major protection schemes for the memory array. Some products support both schemes.

3.1 Individual Block Protection

With this scheme, the host can protect or unprotect individual sectors of the memory array from erase and program operation. The block protection and unprotection actions are referred to in some datasheets as block lock and unlock. There are also provisions for protecting or unprotecting all blocks with a single action (global protect/unprotect or global lock/unlock).

In general, the protection granularity of this scheme is a block of 64 kB, also knows as a “sector.” In certain products such as the AT25XE and AT25FF families, 4K granularity is available in parts of the memory array.

The individual block protection scheme associates each block with a protection bit. When the bit is set, the block is protected; when it is clear, the block is unprotected. Protection bits do not reside in status/control registers. There are dedicated commands to:

- Set a protection bit (protect the associated block).
- Clear a protection bit (unprotect the associated block).
- Read a protection bit (get the associated block’s protection state).

With each of the above commands, the host sends an address to the flash. The action applies to the block that encapsulates that address. In other words, the flash clears the least significant bits of the address to receive the start address of the block to which the action applies. For example, if the host sends an unprotect/unlock command (opcode 39h), and the address field of the command is 0x23800, the flash marks the 64K sector in the address range 0x20000-0x2FFFF as unprotected.

Whenever the individual block protection scheme is used, the reset state of the blocks/sectors is protected. In other words, the reset value of the lock bits is always 1.

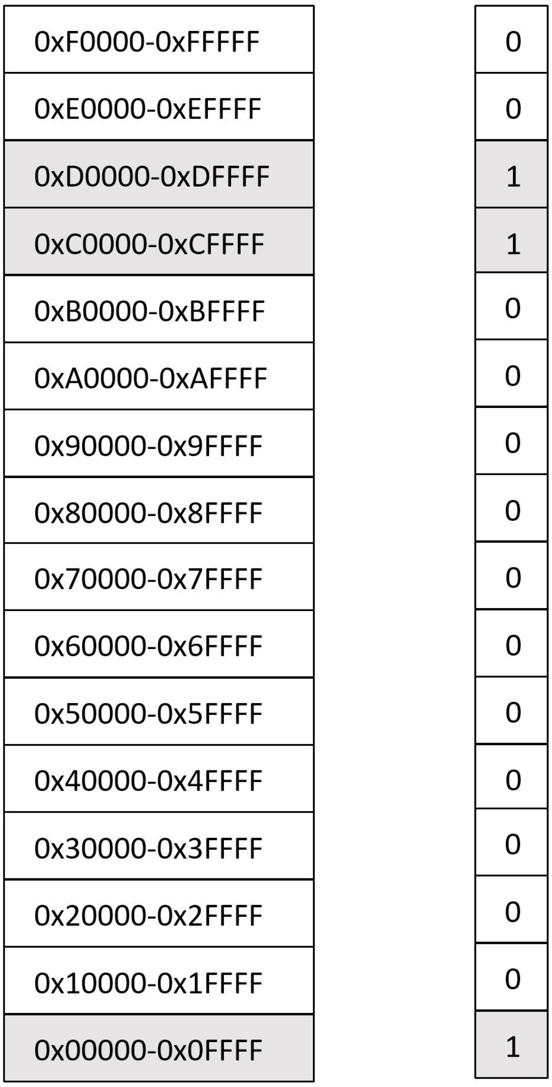
Note that the AT25XE/AT25FF product families have two alternative protection schemes, one of them being the individual block protection scheme. The factory configuration uses the memory edge protection scheme. If the user switches to the individual block protection scheme, all blocks are protected after every reset.

Table 1 provides implementation details of the individual block protection scheme in different product families.

Table 1. Block Protection Implementation Details

Feature	AT25XE/AT25FF	AT25DF/AT25DL
Protect command	36h	36h
Unprotect command	39h	39h
Read protection state	3Ch or 3Dh	3Ch
Global protect command	7Eh	Implemented by writing to the Status Register.
Global unprotect command	98h	Implemented by writing to the Status Register.
Protection granularity	4 kB block for top and bottom 64K, sector (64 kB block) for the rest of the memory array	Sector (64 kB bock)
Reset value of protection bits	1 (protected)	1 (protected)
Notes	Individual protection scheme is not the default protection scheme. To select this scheme, the WPS bit must be set.	Supports individual protection scheme only.

Figure 2 shows a simple example of 1-MB (8-Mbit) flash memory array with individual block protection. Sixteen memory sectors are shown on the left, with three protected sectors highlighted. The respective lock bits are on the right.



0xF0000-0xFFFFF	0
0xE0000-0xEFFFF	0
0xD0000-0xDFFFF	1
0xC0000-0xCFFFF	1
0xB0000-0xBFFFF	0
0xA0000-0xAFFFF	0
0x90000-0x9FFFF	0
0x80000-0x8FFFF	0
0x70000-0x7FFFF	0
0x60000-0x6FFFF	0
0x50000-0x5FFFF	0
0x40000-0x4FFFF	0
0x30000-0x3FFFF	0
0x20000-0x2FFFF	0
0x10000-0x1FFFF	0
0x00000-0x0FFFF	1

Figure 2. Example of Individual Block Protection for 8-Mbit Array

3.2 Memory Edge Protection

The term “memory edge protection” scheme is used here to distinguish it from the individual block protection scheme, but it might not be given that name in datasheets. In fact, in datasheet of products that support both memory array protection schemes, it can be referred to as “standard memory protection.” This scheme is slightly more complicated.

Note: The convention in NOR flash documentation is to refer to the lowest memory address (address 0) as the bottom of the memory and the highest memory address (address = <memory size > - 1) as the top of the memory.

With memory edge protection, a single contiguous region of the memory array is protected. That region either starts at the bottom of the memory or ends at the top of the memory. The size of this protected region can be: 0 (entire memory array is unprotected), equal to the memory array size (entire memory array is unprotected), or a size in between (memory array is partially protected).

The contiguous block is defined by six bits located in Status Registers 1 and 2. Specific details are in the product datasheets. The following is a generic description of what these bits control:

- CMP (or CMPRT) bit (Status Register 2 bit 6) defines whether the contiguous region, whose location is determined by the other bits, is protected or unprotected. If CMP=0, the defined region is protected. If CMP=1, the defined region is unprotected, so its complement region is protected (example: for a 16 MB memory array, if CMP=1 and the defined region is 0x4000-0x1FFFFF, the protected region is 0x0-0x3FFF).
- TB (or BP3) bit (Status Register 1 bit 5) defines whether the contiguous region is aligned with the top or bottom of the memory array. If TB=0, the region is aligned with the top of the memory array. In other words, the end address of the region is the end address of the memory. If TB=0, the region is aligned with bottom of the memory: it starts at address 0.
- BPSIZE (or SEC or BP4) bit (Status Register 1 bit 6) defines the region size granularity. If BPSIZE=0, the granularity is 64 kB. If BPSIZE=1 granularity is 4 kB.
- BP[2:0] (or BP0, BP1, BP2) bits (Status Register 1 bits 2-4) define, in conjunction with the BPSIZE bit, the exact size of the contiguous region.

Table 2 provides some examples.

Table 2. Protection Scheme Examples

Example	CMP	BPSIZE	TB	BP[2:0]	Protection
1	0	0	0	000	With BP[2:0]=000b, the region size is always 0. CMP=0 indicates the defined region is protected. Since the protected region is of size 0, the entire memory is unprotected.
2	0	0	0	001	With BP[2:0]=001b and BPSIZE=0 (64 kB granularity), the region size is 64 kB. TB=0 indicates the region is aligned with the memory top. CMP=0 indicates the defined region is protected. Thus, the last (top) 64 kB of the memory are protected.
3	0	0	1	010	With BP[2:0]=010b and BPSIZE=0 (64K granularity), the region size is 128 kB. TB=1 indicates the region is aligned with the memory bottom. CMP=0 indicates the defined region is protected. Thus, the first (bottom) 128 kB of the memory are protected.
4	0	1	1	001	With BP[2:0]=001b and BPSIZE=1 (4K granularity), the region size is 4 kB. TB=1 indicates the region is aligned with the memory bottom. CMP=0 indicates the defined region is protected. Thus, the first (bottom) 4 kB of the memory are protected.
5	1	1	1	001	With BP[2:0]=001b and BPSIZE=1 (4 kB granularity) the region size is 4 kB. TB=1 indicates the region is aligned with the memory bottom. CMP=1 indicates the defined region is unprotected. Thus, all memory except for the first (bottom) 4 kB is protected.

Figure 3 illustrates the five examples from Table 2. The protected regions are shaded.

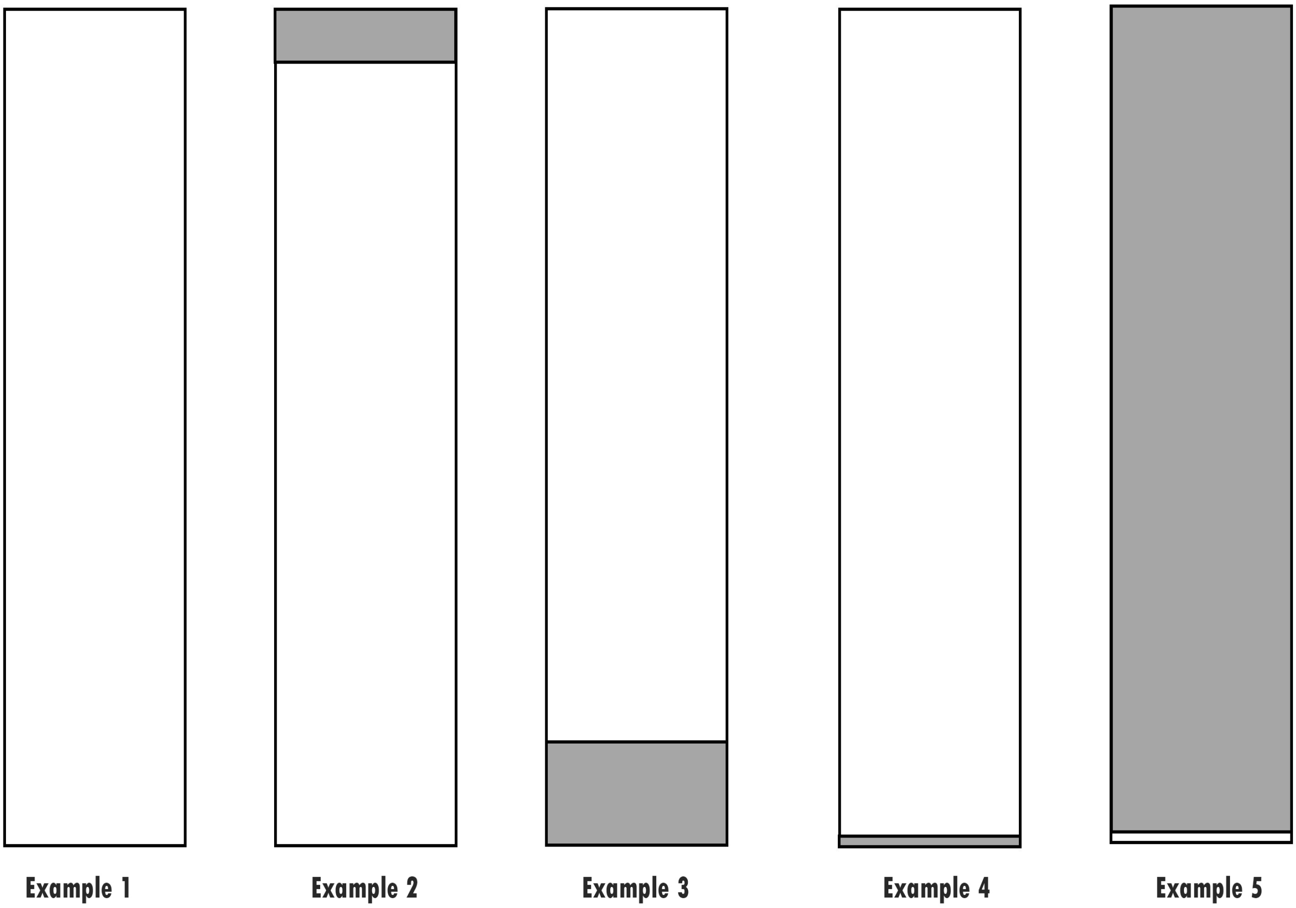


Figure 3. Illustration of Five Examples

Table 3 provides implementation details of edge protection configuration bit naming in different Renesas product families.

Table 3. Bit Naming

Bit(s)	AT25XE/AT25FF	AT25SL/AT25SF	AT25EU
Protection complement bit	CMPRT	CMP	CMP
Top/bottom bit	TB	TB	BP3
Region granularity bit	BPSIZE	SEC	BP4
Region size bits	BP2, BP1, BP0	BP2, BP1, BP0	BP2, BP1, BP0

3.3 Basic Memory Array Protection Schemes

Certain products like the AT25DN family have a basic memory array protection scheme where the array is either entirely protected or entirely unprotected. This is determined by a single bit, called BP0 (Status Register 1 bit 2).

4. Protecting from Accidental Changes with Write-Enable

Write-Enable commands are used to prevent accidental changes, including write operations into Status Registers, erase and program operations on the memory array, and various other operations that make a change to the state of the flash device. For simplicity, the commands used to trigger such operation is referred to as “write commands” in this section.

Unlike the protections described earlier, the Write-Enable protection is not meant to block a write in absolute terms. It merely provides a safeguard for a write operation. Before sending a write command to the flash device, it is required to send a Write-Enable command; otherwise, the write command is ignored and has no effect. This makes the write operation a two-step process, thereby preventing an unintentional change triggered by an accidental or spurious transmission of a write command.

All products support opcode 06h for Write-Enable command. Many products support an additional Write-Enable command with opcode 50h. In contrast to the 06h opcode, which is applicable to all write commands, the 50h opcode is applicable only to register write commands and its semantic is different. The difference between these two Write-Enable commands is explained later in the section.

The Write-Enable command typically sets the WEL (Write-Enable Latch) bit, which most often is bit 1 of Status Register 1. A write command that follows checks the WEL bit and is executed only if it is set to 1. It also causes the WEL bit to be cleared.

Note the following:

- In certain products, a non-write command does not clear the WEL bit. This means that technically the write command can be sent a few commands after Write-Enable, and some non-write commands can be sent in between the two. In general, it is highly recommended to send a write command immediately after Write-Enable.
- A write command that fails may also not clear the WEL bit. For example, a block-erase command that is applied to a protected memory area may not clear the WEL bit.

If, for any reason, it is desired to clear the WEL bit after it was set with the Write-Enable command, a Write-Disable command (opcode 04h) is available.

4.1 Volatile and Non-Volatile Write-Enable

As mentioned above, on some flash products two Write-Enable commands are available for Status Register writes. These are Non-Volatile Write-Enable (opcode 06h) and Volatile Write-Enable (opcode 50h).

Both commands enable writing to Status Registers; however, they have different effects. To understand these effects, the concept of Status Registers volatile copy and non-volatile copy must be explained. In most flash products, Status Registers have two copies:

- A volatile copy that holds the current value of the Status Register. The effective settings of the flash device are based on the volatile copy. This copy does not retain its value across reset events.
- A non-volatile copy that is used only during reset, at which time it is used to initialize the volatile copy. In other words: at reset time, the value of the register’s non-volatile copy is copied to the register’s volatile copy.

Given this Status Register architecture, it is possible to explain the two Write-Enable types.

- When a volatile Write-Enable is used, the following register write command modifies only the Status Register’s volatile copy. This makes the new setting effective immediately. However, this can be regarded as a temporary change, because after the next reset, the Status Register’s volatile copy is re-initialized using the value of the register’s non-volatile copy.
- When a non-volatile Write-Enable is used, the following register write command modifies both the Status Register’s volatile copy and its non-volatile copy. Thus, the new value of the Status Register is retained after the next reset. In other words: the new flash device’s setting, derived from the new value of the register, becomes permanent, at least until a future write to the Status Register.

Figure 4 illustrates the volatile Write-Enable (opcode 50h) vs non-volatile Write-Enable (opcode 06h) on a flash device with four Status Registers (SR1, SR2, SR3 and SR4).

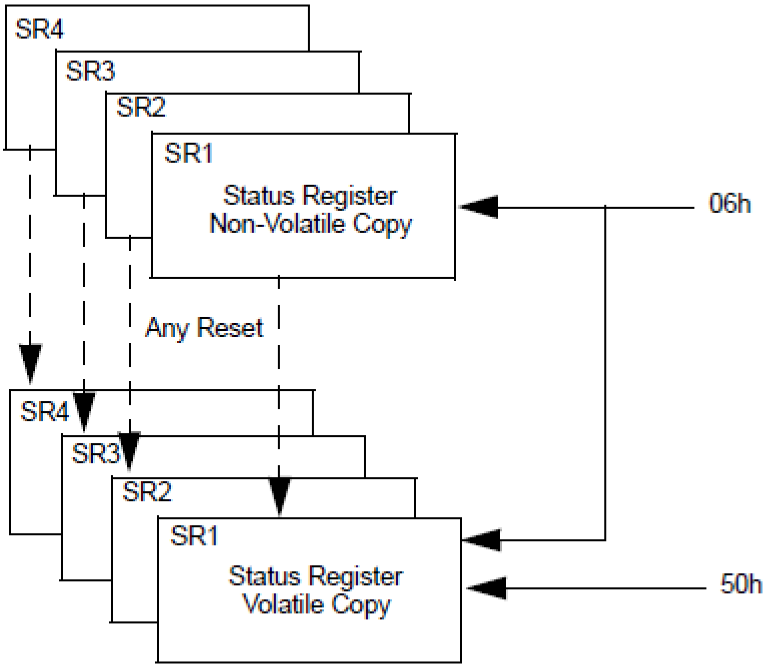


Figure 4. Volatile vs Non-Volatile Write Enable

5. Protecting from an Accidental Software Reset

The Reset-Enable command is similar to the Write-Enable command, but it is dedicated to protecting only a software reset.

Many flash products support a Software-Reset command (opcode 99h); however, this command does not stand on its own. To take effect it must be preceded by a Reset-Enable command (opcode 66h); otherwise, it is ignored. This two-step process is meant to prevent an accidental reset with an unintentional transmission of a Software-Reset command.

6. Revision History

Revision	Date	Description
A0	April, 2022	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.