RX Family

How to Prevent the On-Chip Flash Memory from Being Accessed by Third Parties or Being Accidentally Programmed by Developers

Summary

This application note describes methods of prohibiting access by third parties to the on-chip flash memory of Renesas MCUs and methods of protecting the on-chip flash memory during self-programming initiated by developers.

As used in this application note, the terms “developer” and “third party” are defined as follows.

Developer: The program developer. The person enabling protection of the on-chip flash memory.

Third party: A person other than the developer.

Target Devices

RX Family

Contents

1. Protection from Third-Party Access

1.1 Device Categories

Devices are categorized into seven groups according to their protection functions to prohibit access by third parties.

The device categories are listed in Table 1.1. For details of each protection function, refer to the User's Manual: Hardware of the device.

Table 1.1 Device Categories

		Protection Function (✓: Supported, —: Not Supported)									
Group	Devices	ID Code Protection	On-Chip Debugger ID Code Protection	ROM Code Protection	Serial Programmer Connection Enable /Disable	Serial Programmer Command Control	On-Chip Debugger Connection Enable /Disable	Trusted Memory	Access Window	FSPR bit	Access Window Protection Command
Device group A	• RX210 • RX610 • RX621 • RX62G • RX62N • RX62T • RX630 • RX631 • RX634 • RX63N • RX63T	✓	✓	✓	✓*1	—	—	—	—	—	—
	• RX230 • RX231 • RX24T • RX24U	✓	✓	✓	✓*1	—	—	—	✓	—	—
Device group B	• RX21A • RX220	✓	✓	—*4	✓*1	—	—	—	—	—	—
	• RX130 • RX13T • RX23E-A • RX23T • RX23W	✓	✓	—*4	✓*1	—	—	—	✓	—	—
	• RX140	✓	✓	—*4	✓*1	—	—	—	✓	—	✓
Device group C	• RX110 • RX111 • RX113	✓	✓	—*4	✓*1	—	—	—	✓	—	—
Device group D	• RX64M • RX71M	✓*2	✓*3	✓	✓	✓	—	✓	—	—	—
Device group E	• RX651 • RX65N • RX66N • RX72M • RX72N	✓	✓*3	✓	✓	—	—	✓	✓	✓	—
Device group F	• RX66T • RX72T	✓	✓*3	✓	✓	✓	—	✓	—	—	—
Device group G	• RX671	✓	✓*3	✓	✓	—	✓	✓	✓	✓	—

Note 1. The ID code protection function is used to make serial programmer connection enable/disable settings.

Note 2. When connected, there is no function for erasing the entire on-chip flash memory area.

Note 3. These do not have a capability to always prohibit connection of an on-chip debugger.

Note 4. The ROM Code Protection is not supported, because those devices cannot use parallel programmer.

1.2 Device Group A Protection Methods

1.2.1 Specifications

Three protection functions are provided to prevent access by third parties to the on-chip flash memory: ID code protection, on-chip debugger ID code protection, and ROM code protection.

The access window is not a function to prevent access from third parties.

An overview of each of these protection functions is shown in Table 1.2

Table 1.2 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode or user boot mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
ROM Code Protection	When a parallel programmer is used, reading, programming, or erasing of the on-chip flash memory by third parties is prevented.
Access Window*1	If the access window is set, the area set outside the access window is prevented programming or erasing. The access window is a function to prevent erroneous rewriting in case a program runs out of control during self-programming.

Note 1. The access window is incorporated into RX230, RX231, RX24T and RX24U.

Note that whether or not ID code protection can be used for USB connections in boot mode differs depending on the device. A breakdown by device of the applicability of ID code protection for USB connections is shown in Table 1.3.

For details on the access window, see Chapter 2, Area Protection.

Table 1.3 List of Applicability of ID Code Protection for USB Connections

Devices	Applicability of ID Code Protection
RX621, RX62N, RX630, RX631, RX63N, RX63T	Not usable*1
RX230, RX231	Usable

Note 1. When a USB connection is established, no ID authentication takes place but the user area and data area are erased, thereby preventing third parties from reading from the on-chip flash memory.

1.2.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection, on-chip debugger ID code protection, and ROM code protection settings, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.1, Table 1.4, and Table 1.5. For details on each protection setting pattern, refer to 1.2.3.1 to 1.2.3.5.

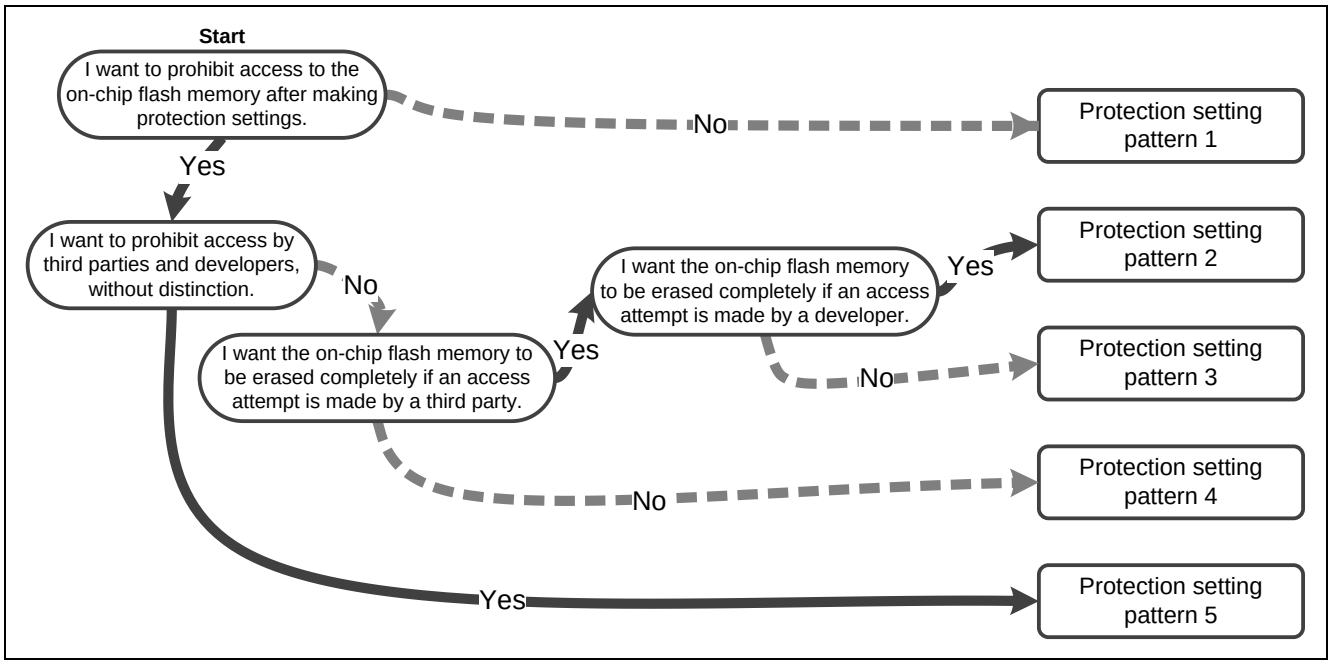


Figure 1.1 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern prevents reading by third parties. When a connection is made in boot mode, the on-chip flash memory is erased completely, without distinction between third parties and developers.
- Protection setting pattern 3
  This protection setting pattern prevents reading by third parties. When a connection is made in boot mode, ID authentication takes place. If ID authentication fails, the on-chip flash memory is erased completely, without distinction between third parties and developers.
- Protection setting pattern 4
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 5
  This protection setting pattern prohibits connections by both developers and third parties. Once this protection setting pattern is applied, the protection cannot be removed, so caution is necessary.

Table 1.4 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection)				Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection)				Connection of parallel programmer (ROM Code Protection)			
	Developer		Third party		Developer		Third party		Developer		Third party	
	R	P/E	R	P/E	R	P/E	R	P/E	R	P/E	R	P/E
1	—	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	—	✓	—	✓	✓	✓	—	—	—	—	—	—
3	✓	✓	—*1	—	✓	✓	—	—	—	—	—	—
4	✓	✓	—	—	✓	✓	—	—	—	—	—	—
5	—	—	—	—	—	—	—	—	—	—	—	✓

R: Read, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. The on-chip flash memory is erased completely if repeated ID code mismatches occur. For details of the scope of “complete erasure,” refer to the User’s Manual: Hardware of the device.

Table 1.5 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection (Protection for Connection of On-Chip Debugger)	ROM Code Protection (Protection for Connection of parallel programmer)
1	Reading is prevented by complete erasure of the on-chip flash memory.*1	Disabled	Disabled
2		Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are prevented always.
3	Reading, programming, and erasing are enabled when the ID code matches. The on-chip flash memory is erased completely*1 if repeated ID code mismatches occur.		
4	Reading, programming, and erasing are enabled when the ID code matches.		
5	Reading, programming, and erasing are prevented always.	Reading, programming, and erasing are prevented always.	

Note 1. For details of the scope of “complete erasure,” refer to the User’s Manual: Hardware of the device.

1.2.3 Description of Protection Setting Patterns

1.2.3.1 Protection Setting Pattern 1

This pattern disables all protection. Note, however that the on-chip flash memory is erased completely when a connection is established in boot mode.

The setting details of protection setting pattern 1 are shown in Table 1.6.

Table 1.6 Protection Setting Pattern 1 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings		ROM Code Protection Settings
Control code (1 byte)	ID code (15 bytes)	ROM code (4 bytes)
FFh	All FFh	Other than (0000 0000h, 0000 0001h)

For the setting method, refer to 1.2.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.7.

Table 1.7 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection	When a connection is made in boot mode, no ID authentication occurs and the on-chip flash memory is erased completely. The device then transitions to a state in which reading, programming, and erasing are possible.	Reading the contents of the on-chip flash memory by a third party is prevented by complete erasure of the on-chip flash memory.
On-Chip Debugger ID Code Protection	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None
ROM Code Protection	Reading, programming, and erasing are possible when a parallel programmer is used.	None

1.2.3.2 Protection Setting Pattern 2

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. It also provides protection by ID authentication when an on-chip debugger is connected. Note that when a connection is made in boot mode, the on-chip flash memory is erased completely, without distinction between third parties and developers.

The setting details of protection setting pattern 2 are shown in Table 1.8.

Table 1.8 Protection Setting Pattern 2 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings		ROM Code Protection Settings
Control code (1 byte)	ID code (15 bytes)	ROM code (4 bytes)
Other than (45h, 52h)	Any value	0000 0000h

For the setting method, refer to 1.2.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.9.

Table 1.9 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection	When a connection is made in boot mode, no ID authentication occurs and the on-chip flash memory is erased completely. The device then transitions to a state in which reading, programming, and erasing are possible.	Reading the contents of the on-chip flash memory by a third party is prevented by complete erasure of the on-chip flash memory.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.2.3.3 Protection Setting Pattern 3

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. It also provides protection by ID authentication when an on-chip debugger is connected. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode. However, if ID authentication fails three times in succession, the on-chip flash memory is erased completely.

The setting details of protection setting pattern 3 are shown in Table 1.10.

Table 1.10 Protection Setting Pattern 3 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings		ROM Code Protection Settings
Control code (1 byte)	ID code (15 bytes)	ROM code (4 bytes)
45h	Any value	0000 0000h

For the setting method, refer to 1.2.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.11.

Table 1.11 Operation of Protection Setting Pattern 3

Protection Type	Operation	Prevented Items
ID Code Protection	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again. However, if ID authentication fails three times in succession, the on-chip flash memory is erased completely.	Reading the contents of the on-chip flash memory by a third party is prevented by complete erasure of the on-chip flash memory. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.2.3.4 Protection Setting Pattern 4

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. It also provides protection by ID authentication when an on-chip debugger is connected. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 4 are shown in Table 1.12.

Table 1.12 Protection Setting Pattern 4 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings		ROM Code Protection Settings
Control code (1 byte)	ID code (15 bytes)	ROM code (4 bytes)
52h	Other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	0000 0000h

For the setting method, refer to 1.2.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.13.

Table 1.13 Operation of Protection Setting Pattern 4

Protection Type	Operation	Prevented Items
ID Code Protection	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.2.3.5 Protection Setting Pattern 5

This pattern prevents reading, programming, or erasing of the on-chip flash memory by using a parallel programmer, when an on-chip debugger is connected, and when a connection is made in boot mode.

Note: After this setting is made and the device is reset, the protection cannot be removed by any method, so caution is necessary.

The setting details of protection setting pattern 5 are shown in Table 1.14.

Table 1.14 Protection Setting Pattern 5 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings		ROM Code Protection Settings
Control code (1 byte)	ID code (15 bytes)	ROM code (4 bytes)
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	0000 0000h

For the setting method, refer to 1.2.4, Protection Setting Examples.

The operation of protection setting pattern 5 is outlined in Table 1.15.

Table 1.15 Operation of Protection Setting Pattern 5

Protection Type	Operation	Prevented Items
ID Code Protection	ID authentication is performed when a connection is made in boot mode, but the ID code is always processed as not matching, and ID authentication is performed again.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection	ID authentication is performed when an on-chip debugger is connected, but the ID code is always processed as not matching, and ID authentication is performed again.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.2.4 Protection Setting Examples

Each protection function is enabled by setting a control code, ID code, and ROM code to addresses in the on-chip flash memory. The control code and ID code should be set to 0xFFFFFFA0, and the ROM code to 0xFFFFFF9C.

Protection setting examples are shown in Figure 1.2 and Figure 1.3.

```
/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE[4] = {0x45010203, 0x04050607, 0x08090A0B, 0x0C0D0E0F};

/* Setup the ROM Code Protection */
#pragma address ROM_CODE = 0xFFFFFF9C
const unsigned long ROM_CODE = 0x00000000;
```

In this example, the control code is 45h, and the ID code is 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh.

Figure 1.2 Protect Setting Pattern 3 Setting Example

```
/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE[4] = {0x5250726F, 0x74656374, 0xFFFFFFFF, 0xFFFFFFFF};

/* Setup the ROM Code Protection */
#pragma address ROM_CODE = 0xFFFFFF9C
const unsigned long ROM_CODE = 0x00000000;
```

In this example, the control code is 52h, and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, FFh.

Figure 1.3 Protect Setting Pattern 5 Setting Example

1.3 Device Group B Protection Methods

1.3.1 Specifications

There are three protection functions available to prohibit access to the on-chip flash memory by third parties: ID code protection, on-chip debugger ID code protection, and Access Window protection command.

The access window is not a function to prevent access from third parties. However, access from third parties can be prohibited by combining the access window protection command and ID code protection.

An overview of each of these protection functions is shown in Table 1.16

Table 1.16 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
Access Window*1	If the access window is set, the area set outside the access window is prevented programming or erasing. The access window is a function to prevent erroneous rewriting in case a program runs out of control during self-programming.
Access Window protection command*2	If the access window protection command is executed, the access window never be set again.*3

Note 1. The access window is incorporated into RX130, RX13T, RX140, RX23E-A, RX23T and RX23W.
Note 2. The access window protection command is incorporated into RX140.
Note 3. Can be set again in boot mode.

For details on the access window, see Chapter 2, Area Protection.

1.3.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection and on-chip debugger ID code protection settings, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.4, Table 1.17, and Table 1.18. For details on each protection setting pattern, refer to 1.3.3.1 to 1.3.3.5.

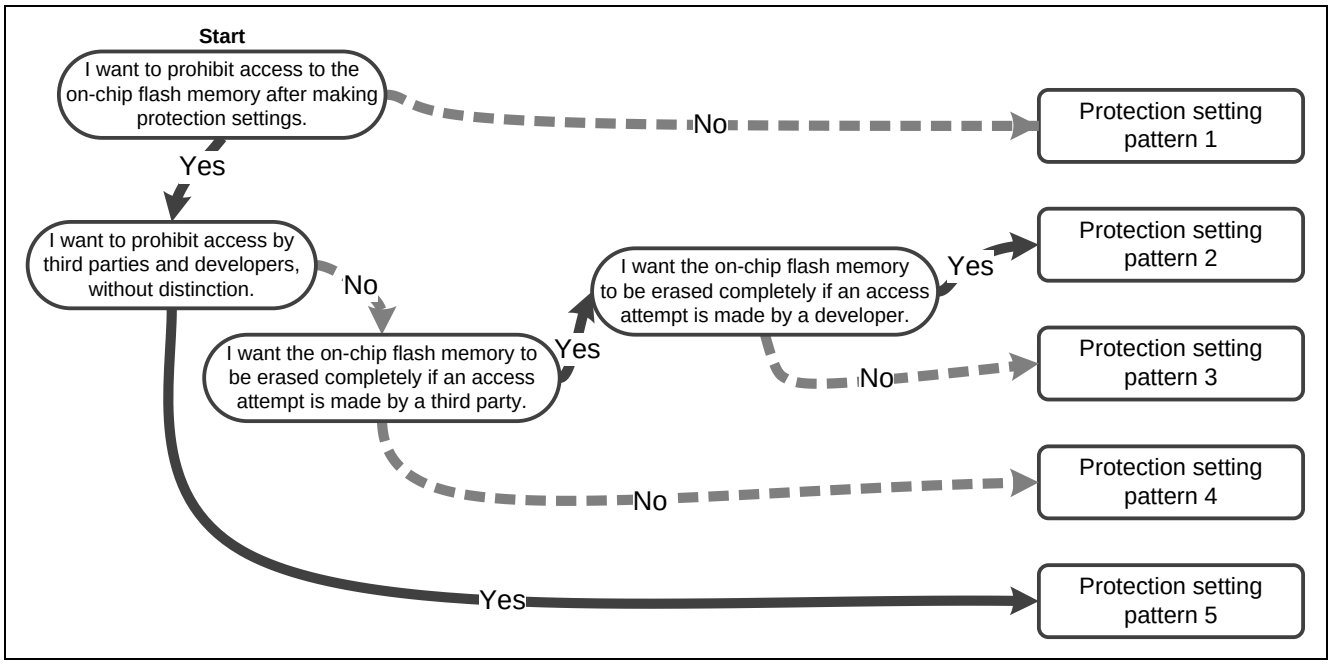


Figure 1.4 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern prevents reading by third parties. When a connection is made in boot mode, the on-chip flash memory is erased completely, without distinction between third parties and developers.
- Protection setting pattern 3
  This protection setting pattern prevents reading by third parties. When a connection is made in boot mode, ID authentication takes place. If ID authentication fails, the on-chip flash memory is erased completely, without distinction between third parties and developers.
- Protection setting pattern 4
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 5
  This protection setting pattern prohibits connections by both developers and third parties. Once this protection setting pattern is applied, the protection cannot be removed, so caution is necessary.

Table 1.17 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection)				Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection)			
	Developer		Third party		Developer		Third party	
	R	P/E	R	P/E	R	P/E	R	P/E
1	—	✓	—	✓	✓	✓	✓	✓
2	—	✓	—	✓	✓	✓	—	—
3	✓	✓	—*1	—	✓	✓	—	—
4	✓	✓	—	—	✓	✓	—	—
5	—	—	—	—	—	—	—	—

R: Read, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. The on-chip flash memory is erased completely if repeated ID code mismatches occur.

Table 1.18 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection (Protection for Connection of On-Chip Debugger)
1	Reading is prevented by complete erasure of the on-chip flash memory.*1	Disabled
2		Reading, programming, and erasing are enabled when the ID code matches.
3	Reading, programming, and erasing are enabled when the ID code matches. The on-chip flash memory is erased completely*1 if repeated ID code mismatches occur.	
4	Reading, programming, and erasing are enabled when the ID code matches.	
5	Reading, programming, and erasing are prevented always.	Reading, programming, and erasing are prevented always.

Note 1. Complete erasure: Erasure of the user area and data area.

1.3.3 Description of Protection Setting Patterns

1.3.3.1 Protection Setting Pattern 1

This pattern disables all protection. Note, however that the on-chip flash memory is erased completely when a connection is established in boot mode.

The setting details of protection setting pattern 1 are shown in Table 1.19.

Table 1.19 Protection Setting Pattern 1 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
FFh	All FFh

For the setting method, refer to 1.3.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.20.

Table 1.20 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection	When a connection is made in boot mode, no ID authentication occurs and the on-chip flash memory is erased completely. The device then transitions to a state in which reading, programming, and erasing are possible.	Reading the contents of the on-chip flash memory by a third party is prevented by complete erasure of the on-chip flash memory.
On-Chip Debugger ID Code Protection	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None

1.3.3.2 Protection Setting Pattern 2

This pattern provides protection by means of ID authentication when an on-chip debugger is connected. Note that when a connection is made in boot mode, the on-chip flash memory is erased completely, without distinction between third parties and developers.

The setting details of protection setting pattern 2 are shown in Table 1.21.

Table 1.21 Protection Setting Pattern 2 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
Other than (45h, 52h)	Any value

For the setting method, refer to 1.3.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.22.

Table 1.22 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection	When a connection is made in boot mode, no ID authentication occurs and the on-chip flash memory is erased completely. The device then transitions to a state in which reading, programming, and erasing are possible.	Reading the contents of the on-chip flash memory by a third party is prevented by complete erasure of the on-chip flash memory.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.

1.3.3.3 Protection Setting Pattern 3

This pattern provides protection by means of ID authentication when an on-chip debugger is connected. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode. However, if ID authentication fails three times in succession, the on-chip flash memory is erased completely.

The setting details of protection setting pattern 3 are shown in Table 1.23.

Table 1.23 Protection Setting Pattern 3 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
45h	Any value

For the setting method, refer to 1.3.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.24.

Table 1.24 Operation of Protection Setting Pattern 3

Protection Type	Operation	Prevented Items
ID Code Protection	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again. However, if ID authentication fails three times in succession, the on-chip flash memory is erased completely.	Reading the contents of the on-chip flash memory by a third party is prevented by complete erasure of the on-chip flash memory. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.

1.3.3.4 Protection Setting Pattern 4

This pattern provides protection by means of ID authentication when an on-chip debugger is connected. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 4 are shown in Table 1.25.

Table 1.25 Protection Setting Pattern 4 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
52h	Other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh

For the setting method, refer to 1.3.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.26.

Table 1.26 Operation of Protection Setting Pattern 4

Protection Type	Operation	Prevented Items
ID Code Protection	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.

1.3.3.5 Protection Setting Pattern 5

This pattern prevents reading, programming, or erasing of the on-chip flash memory by using a when an on-chip debugger is connected and when a connection is made in boot mode.

Note: After this setting is made and the device is reset, the protection cannot be removed by any method, so caution is necessary.

The setting details of protection setting pattern 5 are shown in Table 1.27.

Table 1.27 Protection Setting Pattern 5 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh

For the setting method, refer to 1.3.4, Protection Setting Examples.

The operation of protection setting pattern 5 is outlined in Table 1.28.

Table 1.28 Operation of Protection Setting Pattern 5

Protection Type	Operation	Prevented Items
ID Code Protection	ID authentication is performed when a connection is made in boot mode, but the ID code is always processed as not matching, and ID authentication is performed again.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection	ID authentication is performed when an on-chip debugger is connected, but the ID code is always processed as not matching, and ID authentication is performed again.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.

1.3.4 Protection Setting Examples

Each protection function is enabled by setting a control code and ID code to an address in the on-chip flash memory. The control code and ID code should be set to 0xFFFFFFA0.

Protection setting examples are shown in Figure 1.5 and Figure 1.6.

```
/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE[4] = {0x45010203, 0x04050607, 0x08090A0B, 0x0C0D0E0F};
```

In this example, the control code is 45h, and the ID code is 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh.

Figure 1.5 Protect Setting Pattern 3 Setting Example

```
/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE[4] = {0x5250726F, 0x74656374, 0xFFFFFFFF, 0xFFFFFFFF};
```

In this example, the control code is 52h, and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, FFh.

Figure 1.6 Protect Setting Pattern 5 Setting Example

1.4 Device Group C Protection Methods

1.4.1 Specifications

There are two protection functions available to prohibit access to the on-chip flash memory by third parties: ID code protection and on-chip debugger ID code protection.

The access window is not a function to prevent access from third parties.

An overview of each of these protection functions is shown in Table 1.29

Table 1.29 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
Access Window	If the access window is set, the area set outside the access window is prevented programming or erasing. The access window is a function to prevent erroneous rewriting in case a program runs out of control during self-programming.

For details on the access window, see Chapter 2, Area Protection.

1.4.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection and on-chip debugger ID code protection settings, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.7, Table 1.30, and Table 1.31. For details on each protection setting pattern, refer to 1.4.3.1 to 1.4.3.5.

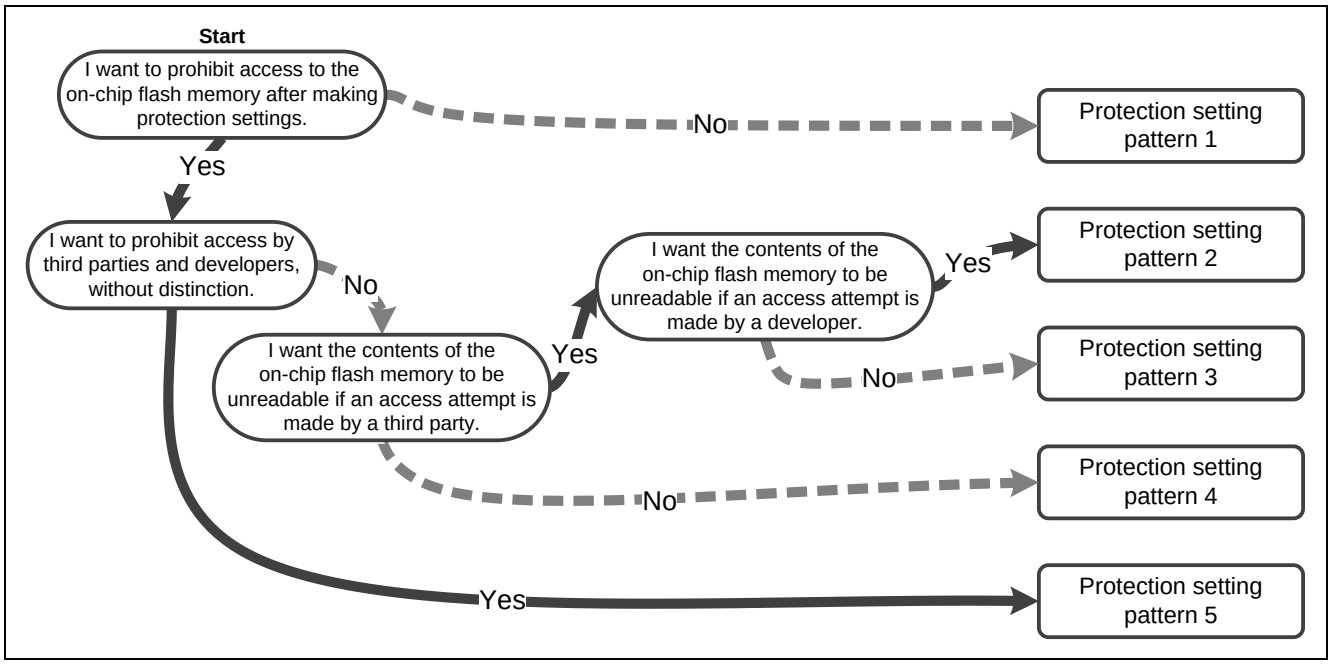


Figure 1.7 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern prevents reading by third parties. When a connection is made in boot mode, the device transitions to the erase-ready*1 state, and reading and programming are prohibited, without distinction between third parties and developers, until all blocks in the user area and data area have been erased.
- Protection setting pattern 3
  This protection setting pattern prevents reading by third parties. When a connection is made in boot mode, ID authentication takes place. If ID authentication fails, the device transitions to the erase-ready*1 state, and reading and programming are prohibited, without distinction between third parties and developers, until all blocks in the user area and data area have been erased.
- Protection setting pattern 4
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 5
  This protection setting pattern prohibits connections by both developers and third parties. Once this protection setting pattern is applied, the protection cannot be removed, so caution is necessary.

Note 1. For details of the erase-ready state, refer to the User's Manual: Hardware of the device.

Table 1.30 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection)				Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection)			
	Developer		Third party		Developer		Third party	
	R	P/E	R	P/E	R	P/E	R	P/E
1	—*1	✓	—*1	✓	✓	✓	✓	✓
2	—*2	✓	—*2	✓	✓	✓	—	—
3	✓	✓	—*3	—	✓	✓	—	—
4	✓	✓	—	—	✓	✓	—	—
5	—	—	—	—	—	—	—	—

R: Read, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. If there is data in the on-chip flash memory when a connection occurs, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased.

Note 2. When a connection occurs, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased.

Note 3. If the ID code does not match during successive attempts, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased.

Table 1.31 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection (Protection for Connection of On-Chip Debugger)
1	If there is data in the on-chip flash memory, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased, thereby preventing reading of the memory contents.	Disabled
2	The device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased, thereby preventing reading of the memory contents.	Reading, programming, and erasing are enabled when the ID code matches.
3	Reading, programming, and erasing are enabled when the ID code matches. If the ID code does not match during successive attempts, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased, thereby preventing reading of the memory contents.	
4	Reading, programming, and erasing are enabled when the ID code matches.	
5	Reading, programming, and erasing are prevented always.	Reading, programming, and erasing are prevented always.

1.4.3 Description of Protection Setting Patterns

1.4.3.1 Protection Setting Pattern 1

This pattern disables all protection. Nevertheless, caution is necessary because if a connection occurs in boot mode and there is data in the on-chip flash memory, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased.

The setting details of protection setting pattern 1 are shown in Table 1.32.

Table 1.32 Protection Setting Pattern 1 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
FFh	All FFh

For the setting method, refer to 1.4.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.33.

Table 1.33 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection	If a connection occurs in boot mode and there is no data in the on-chip flash memory, no ID authentication occurs and the device transitions to a state in which reading, programming, and erasing are possible. If a connection occurs in boot mode and there is data in the on-chip flash memory, no ID authentication occurs and the device transitions to the erase-ready state. After all blocks in the user area and data area have been erased, the device transitions to a state in which reading, programming, and erasing are possible.	Reading of the contents of the on-chip flash memory by third parties is prevented by transitioning to the erase-ready state and preventing reading and programming until all blocks in the user area and data area have been erased.
On-Chip Debugger ID Code Protection	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None

1.4.3.2 Protection Setting Pattern 2

This pattern provides protection by means of ID authentication when an on-chip debugger is connected. When a connection is made in boot mode, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased, without distinction between third parties and developers.

The setting details of protection setting pattern 2 are shown in Table 1.34.

Table 1.34 Protection Setting Pattern 2 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
Other than (45h, 52h)	Any value

For the setting method, refer to 1.4.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.35.

Table 1.35 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection	If a connection occurs in boot mode, no ID authentication occurs and the device transitions to the erase-ready state. After all blocks in the user area and data area have been erased, the device transitions to a state in which reading, programming, and erasing are possible.	Reading of the contents of the on-chip flash memory by third parties is prevented by transitioning to the erase-ready state and preventing reading and programming until all blocks in the user area and data area have been erased.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.

1.4.3.3 Protection Setting Pattern 3

This pattern provides protection by means of ID authentication when an on-chip debugger is connected. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode. Nevertheless, caution is necessary because if ID authentication fails three times in succession, the device transitions to the erase-ready state, and reading and programming are prohibited until all blocks in the user area and data area have been erased.

The setting details of protection setting pattern 3 are shown in Table 1.36.

Table 1.36 Protection Setting Pattern 3 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
45h	Any value

For the setting method, refer to 1.4.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.37.

Table 1.37 Operation of Protection Setting Pattern 3

Protection Type	Operation	Prevented Items
ID Code Protection	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again. However, if ID authentication fails three times in succession, the device transitions to the erase-ready state. After all blocks in the user area and data area have been erased, the device transitions to a state in which reading, programming, and erasing are possible.	Reading of the on-chip flash memory by third parties is prevented by transitioning to the erase-ready state and preventing reading and programming until all blocks in the user area and data area have been erased. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.

1.4.3.4 Protection Setting Pattern 4

This pattern provides protection by means of ID authentication when an on-chip debugger is connected. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 4 are shown in Table 1.38.

Table 1.38 Protection Setting Pattern 4 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
52h	Other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh

For the setting method, refer to 1.4.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.39.

Table 1.39 Operation of Protection Setting Pattern 4

Protection Type	Operation	Prevented Items
ID Code Protection	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.

1.4.3.5 Protection Setting Pattern 5

This pattern prevents reading, programming, or erasing of the on-chip flash memory by using a when an on-chip debugger is connected and when a connection is made in boot mode.

Note: After this setting is made and the device is reset, the protection cannot be removed by any method, so caution is necessary.

The setting details of protection setting pattern 5 are shown in Table 1.40.

Table 1.40 Protection Setting Pattern 5 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	
Control code (1 byte)	ID code (15 bytes)
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh

For the setting method, refer to 1.4.4, Protection Setting Examples.

The operation of protection setting pattern 5 is outlined in Table 1.41.

Table 1.41 Operation of Protection Setting Pattern 5

Protection Type	Operation	Prevented Items
ID Code Protection	ID authentication is performed when a connection is made in boot mode, but the ID code is always processed as not matching, and ID authentication is performed again.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection	ID authentication is performed when an on-chip debugger is connected, but the ID code is always processed as not matching, and ID authentication is performed again.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.

1.4.4 Protection Setting Examples

Each protection function is enabled by setting a control code and ID code to an address in the on-chip flash memory. The control code and ID code should be set to 0xFFFFFFA0.

Protection setting examples are shown in Figure 1.8 and Figure 1.9.

```
/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE[4] = {0x45010203, 0x04050607, 0x08090A0B, 0x0C0D0E0F};
```

In this example, the control code is 45h, and the ID code is 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh.

Figure 1.8 Protect Setting Pattern 3 Setting Example

```
/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE[4] = {0x5250726F, 0x74656374, 0xFFFFFFFF, 0xFFFFFFFF};
```

In this example, the control code is 52h, and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, FFh.

Figure 1.9 Protect Setting Pattern 5 Setting Example

1.5 Device Group D Protection Methods

1.5.1 Specifications

Six protection functions are provided to prohibit third parties from accessing the on-chip flash memory: ID code protection, on-chip debugger ID code protection, ROM code protection, serial programmer connection enable/disable, serial programmer command control, and trusted memory.

An overview of each of these protection functions is shown in Table 1.42

Table 1.42 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode or user boot mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
ROM Code Protection	When a parallel programmer is used, reading, programming, or erasing of the on-chip flash memory by third parties is prevented.
Serial Programmer Connection Enable/Disable	Connections by third parties are prohibited, and reading, programming and erasing of the on-chip flash memory is prevented, by prohibiting connections to a host in boot mode.
Serial Programmer Command Control	Reading, programming, or erasing of the on-chip flash memory after a host is connected in boot mode can be enabled/disabled individually.
Trusted Memory	Reading of the trusted memory area in the on-chip flash memory is prevented.

For instructions on using the trusted memory function, refer to the application note “RX Family: Using the Trusted Memory Function” (R01AN2618).

1.5.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection, on-chip debugger ID code protection, ROM code protection, serial programmer connection enable/disable, and serial programmer command control settings, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.10, Table 1.43, and Table 1.44. For details on each protection setting pattern, refer to 1.5.3.1 to 1.5.3.4.

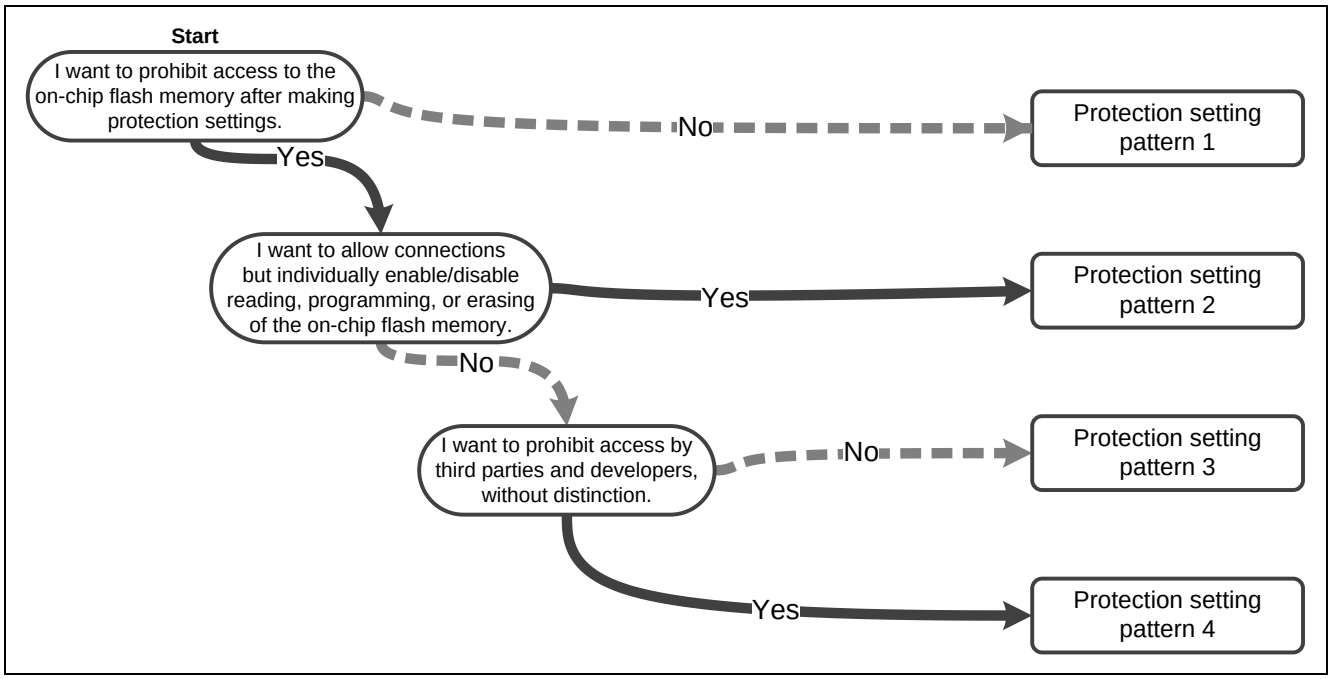


Figure 1.10 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern individually enables/disables reading, programming, or erasing of the on-chip flash memory by third parties and developers, without distinction.
- Protection setting pattern 3
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 4
  This protection setting pattern prohibits connections by both developers and third parties.

Table 1.43 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection, Serial Programmer Connection Enable/Disable, Serial Programmer Command Control)						Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection)				Connection of parallel programmer (ROM Code Protection)			
	Developer			Third party			Developer		Third party		Developer		Third party	
	R	P	E	R	P	E	R	P/E	R	P/E	R	P/E	R	P/E
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓/—*1	✓/—*1	✓/—*1	✓/—*1	✓/—*1	✓/—*1	✓	✓	—	—	—	—	—	—
3	✓	✓	✓	—	—	—	✓	✓	—	—	—	—	—	—
4	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—

R: Read, P: Program, E: Erase, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. The serial programmer command control register (SPCC) is used to individually enable or disable reading, programming, or erasing. For details of the serial programmer command control register (SPCC), refer to the User's Manual: Hardware of the device.

Table 1.44 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection, Serial Programmer Connection Enable/Disable, Serial Programmer Command Control (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection (Protection for Connection of On-Chip Debugger)	ROM Code Protection (Protection for Connection of parallel programmer)
1	Disabled	Disabled	Disabled
2	Individually enables/disables reading, programming, or erasing.	Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are prevented always.
3	Reading, programming, and erasing are enabled when the ID code matches.		
4	Reading, programming, and erasing are prevented always.		

1.5.3 Description of Protection Setting Patterns

1.5.3.1 Protection Setting Pattern 1

This pattern disables all protection.

The setting details of protection setting pattern 1 are shown in Table 1.45 and Table 1.46.

Table 1.45 Protection Setting Pattern 1 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
1	1	1	1	1

Table 1.46 Protection Setting Pattern 1 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
All FFh	Other than (0000 0000h, 0000 0001h)

For the setting method, refer to 1.5.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.47.

Table 1.47 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	If a connection occurs in boot mode, no ID authentication occurs and the device transitions to a state in which reading, programming, and erasing are possible.	None
On-Chip Debugger ID Code Protection	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None
ROM Code Protection	Reading, programming, and erasing are possible when a parallel programmer is used.	None

1.5.3.2 Protection Setting Pattern 2

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. No ID authentication is made when a connection occurs in boot mode, but reading, programming, or erasing of the on-chip flash memory can be enabled/disabled individually.

The setting details of protection setting pattern 2 are shown in Table 1.48 and Table 1.49.

Table 1.48 Protection Setting Pattern 2 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
1	1	0/1*1	0/1*2	0/1*3

Note 1. When the SEPR bit is set to 1 erasing is enabled, and erasing is disabled when it is cleared to 0.
Note 2. When the WRPR bit is set to 1 programming is enabled, and programming is disabled when it is cleared to 0.
Note 3. When the RDPR bit is set to 1 reading is enabled, and reading is disabled when it is cleared to 0.

Table 1.49 Protection Setting Pattern 2 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Other than (All FFh)	0000 0000h

For the setting method, refer to 1.5.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.50.

Table 1.50 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	No ID authentication occurs when a connection is made in boot mode. Enables/disables reading, programming, and erasing individually after the connection is established.	Individually disables enables/disables reading, programming, and erasing, without distinction between third parties and developers.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.5.3.3 Protection Setting Pattern 3

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 3 are shown in Table 1.51 and Table 1.52.

Table 1.51 Protection Setting Pattern 3 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
0	1	0	0	0

Table 1.52 Protection Setting Pattern 3 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Other than (All FFh)	0000 0000h

For the setting method, refer to 1.5.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.53.

Table 1.53 Operation of Protection Setting Pattern 3

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.5.3.4 Protection Setting Pattern 4

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. Connections to a host are prohibited in boot mode.

The setting details of protection setting pattern 4 are shown in Table 1.54 and Table 1.55.

Note that both setting number 1 and setting number 2 prohibit connections to a host in boot mode.

Table 1.54 Protection Setting Pattern 4 Setting Details 1

	Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
	Serial Programmer Command Control Register (SPCC) (4 Bytes)				
Setting No.	ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
1	0	0	0	0	0
2	1	0	Don't care	Don't care	Don't care

Table 1.55 Protection Setting Pattern 4 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	**ROM Code Protection Settings**
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Other than (All FFh)	0000 0000h

For the setting method, refer to 1.5.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.56.

Table 1.56 Operation of Protection Setting Pattern 4

Protection Type	**Operation**	**Prevented Items**
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	Connections to a host are prohibited when connecting in boot mode.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.5.4 Protection Setting Examples

Each protection function is enabled by setting serial programmer connection enable/disable, serial programmer command control, and an ID code to addresses in the option-setting memory, and a ROM code to an address in the on-chip flash memory. Serial programmer connection enable/disable and serial programmer command control should be setting to 0x00120040, the ID code to 0x00120050, and the ROM code to 0xFFFFFF9C.

Also, for instructions on writing data to the option-setting memory, refer to the User's Manual: Hardware of the device.

Protection setting examples are shown in Figure 1.11 and Figure 1.12.

```
/* Setup the Serial programmer command control register */
#pragma address SPCC_REG = 0x00120040
const unsigned long SPCC_REG = 0x1EFFFFFF

/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0x00120050
const unsigned long ID_CODE[4] = {0x04030201, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection */
#pragma address ROM_CODE = 0xFFFFFF9C
const unsigned long ROM_CODE = 0x00000000;
```

In this example, the ID code is 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.11 Protect Setting Pattern 3 Setting Example

```
/* Setup the Serial programmer command control register */
#pragma address SPCC_REG = 0x00120040
const unsigned long SPCC_REG = 0x16FFFFFF

/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0x00120050
const unsigned long ID_CODE[4] = {0x04030201, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection */
#pragma address ROM_CODE = 0xFFFFFF9C
const unsigned long ROM_CODE = 0x00000000;
```

In this example, the ID code is 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.12 Protect Setting Pattern 4 Setting Example

1.6 Device Group E Protection Methods

1.6.1 Specifications

Six protection functions are provided to prohibit third parties from accessing the on-chip flash memory: ID code protection, on-chip debugger ID code protection, ROM code protection, serial programmer connection enable/disable, trusted memory and FSPR bit.

The access window is not a function to prevent access from third parties, but it is possible to prevent access from third parties by setting the FSPR bit.

An overview of each of these protection functions is shown in Table 1.57

Table 1.57 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
ROM Code Protection	When a parallel programmer is used, reading, programming, or erasing of the on-chip flash memory by third parties is prevented.
Serial Programmer Connection Enable/Disable	Connections by third parties are prohibited, and reading, programming and erasing of the on-chip flash memory is prevented, by prohibiting connections to a host in boot mode.
Trusted Memory	Reading of the trusted memory area in the on-chip flash memory is prevented.
Access Window	If the access window is set, the area set outside the access window is prevented programming or erasing. The access window is a function to prevent erroneous rewriting in case a program runs out of control during self-programming.
FSPR bit	If the FSPR bit is set, the access window never be set again. By setting the access window and the FSPR bit at the same time, the area outside the access window can be used as an area that can never be programmed or erased again by both developers and third parties.

For details on the access window, see Chapter 2, Area Protection.

1.6.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection, on-chip debugger ID code protection, ROM code protection, and serial programmer connection enable/disable, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.13, Table 1.58, and Table 1.59. For details on each protection setting pattern, refer to 1.6.3.1 to 1.6.3.4.

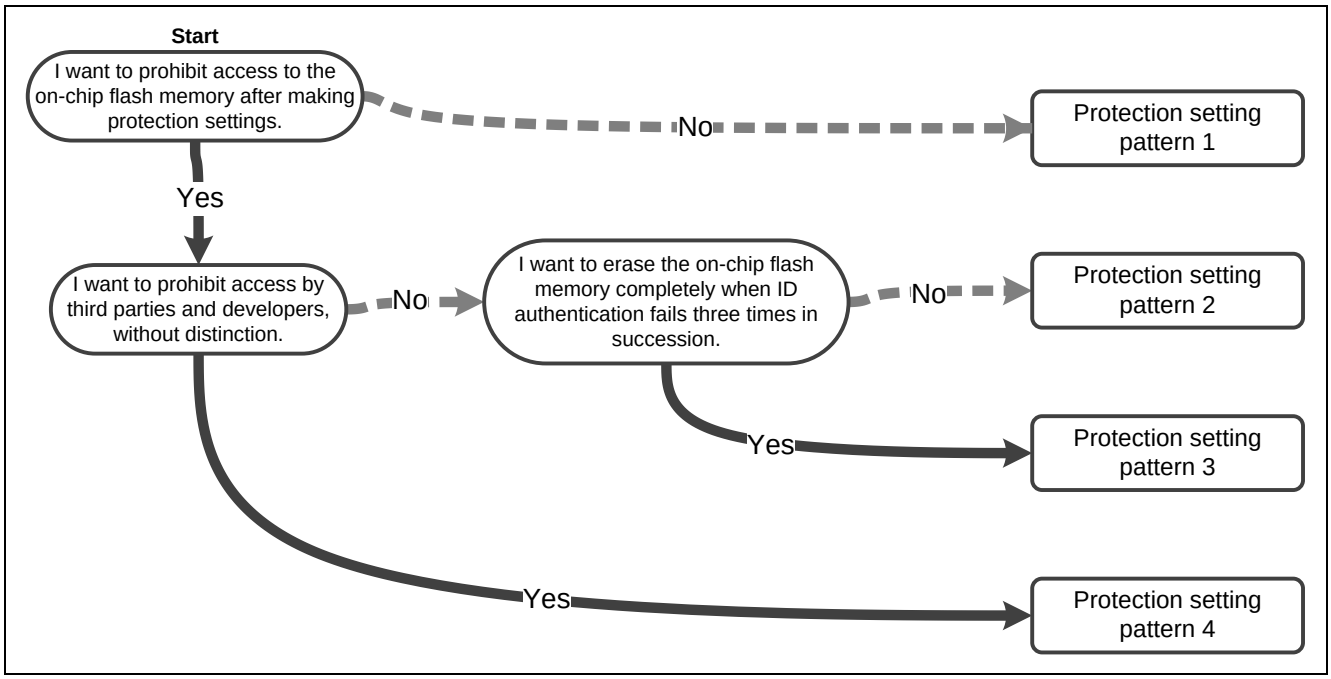


Figure 1.13 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 3
  This protection setting pattern prevents reading, programming, or erasing by third parties. If ID authentication fails three times in succession, the on-chip flash memory is erased completely.
- Protection setting pattern 4
  This protection setting pattern prohibits connections by both developers and third parties.

Table 1.58 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection, Serial Programmer Connection Enable/Disable)				Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection)				Connection of parallel programmer (ROM Code Protection)			
	Developer		Third party		Developer		Third party		Developer		Third party	
	R	P/E	R	P/E	R	P/E	R	P/E	R	P/E	R	P/E
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	—	—	✓	✓	—	—	—	—	—	—
3	✓	✓	—*1	—	✓	✓	—	—	—	—	—	—
4	—	—	—	—	✓	✓	—	—	—	—	—	—

R: Read, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. The on-chip flash memory is erased completely if repeated ID code mismatches occur.

Table 1.59 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection, Serial Programmer Connection Enable/Disable (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection (Protection for Connection of On-Chip Debugger)	ROM Code Protection (Protection for Connection of parallel programmer)
1	Disabled	Disabled	Disabled
2	Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are prevented always.
3	Reading, programming, and erasing are enabled when the ID code matches. If ID authentication fails three times in succession, the on-chip flash memory is erased completely.		
4	Reading, programming, and erasing are prevented always.		

1.6.3 Description of Protection Setting Patterns

1.6.3.1 Protection Setting Pattern 1

This pattern disables all protection.

The setting details of protection setting pattern 1 are shown in Table 1.60.

Table 1.60 Protection Setting Pattern 1 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
All FFh	FFFF FFFFh	Other than (0000 0000h, 0000 0001h)

For the setting method, refer to 1.6.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.61.

Table 1.61 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	If a connection occurs in boot mode, the device transitions to a state in which reading, programming, and erasing are possible by transmitting the ID code all set to FFh.	None
On-Chip Debugger ID Code Protection	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None
ROM Code Protection	Reading, programming, and erasing are possible when a parallel programmer is used.	None

1.6.3.2 Protection Setting Pattern 2

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 2 are shown in Table 1.62.

Table 1.62 Protection Setting Pattern 2 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
Control code / ID code 1: Other than 45h ID code 2 to ID code 16: Any value	FFFF FFFFh	0000 0000h

For the setting method, refer to 1.6.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.63.

Table 1.63 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-chip debugger ID code protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.6.3.3 Protection Setting Pattern 3

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode. If ID authentication fails three times in succession, the on-chip flash memory is erased completely in boot mode.

The setting details of protection setting pattern 3 are shown in Table 1.64.

Table 1.64 Protection Setting Pattern 3 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
Control code / ID code 1: 45h ID code 2 to ID code 16: Any value	FFFF FFFFh	0000 0000h

For the setting method, refer to 1.6.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.65.

Table 1.65 Operation of Protection Setting Pattern 3

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again. If ID authentication fails three times in succession, the on-chip flash memory is erased completely in boot mode.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. If ID authentication fails three times in succession, the on-chip flash memory is erased completely. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-chip debugger ID code protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.6.3.4 Protection Setting Pattern 4

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. Connections to a host are prohibited in boot mode.

The setting details of protection setting pattern 4 are shown in Table 1.66.

Table 1.66 Protection Setting Pattern 4 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
16 bytes are Other than FFh	F7FFFFFFh (SPE bit = 0)	0000 0000h

For the setting method, refer to 1.6.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.67.

Table 1.67 Operation of Protection Setting Pattern 4

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	Connections to a host are prohibited when connecting in boot mode.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.6.4 Protection Setting Examples

Each protection function is enabled by setting serial programmer connection enable/disable, the ID code, and the ROM code to addresses in the option-setting memory. Serial programmer connection enable/disable should be set to 0xFE7F5D40, the ID code to 0xFE7F5D50, and the ROM code to 0xFE7F5D70.

Also, for instructions on writing data to the option-setting memory, refer to the User's Manual: Hardware of the device.

Protection setting examples are shown in Figure 1.14 and Figure 1.15.

```
/* Setup the Serial programmer command control Register */
#pragma address SPCC_REG = 0xFE7F5D40
const unsigned long SPCC_REG = 0xFFFFFFFF;

/* Setup the OCD/Serial Programmer ID Setting Register */
#pragma address OSIS_REG = 0xFE7F5D50
const unsigned long OSIS_REG[4] = {0x04030245, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection Register */
#pragma address ROMCODE_REG = 0xFE7F5D70
const unsigned long ROMCODE_REG = 0x00000000;
```

In this example, the Control code / ID code 1 is 45h, from ID code 2 to ID code 16 are 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.14 Protect Setting Pattern 3 Setting Example

```
/* Setup the Serial programmer command control Register */
#pragma address SPCC_REG = 0xFE7F5D40
const unsigned long SPCC_REG = 0xF7FFFFFF;

/* Setup the OCD/Serial Programmer ID Setting Register */
#pragma address OSIS_REG = 0xFE7F5D50
const unsigned long OSIS_REG[4] = {0x04030201, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection Register */
#pragma address ROMCODE_REG = 0xFE7F5D70
const unsigned long ROMCODE_REG = 0x00000000;
```

In this example, the Control code / ID code 1 is 01h, from ID code 2 to ID code 16 are 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.15 Protect Setting Pattern 4 Setting Example

1.7 Device Group F Protection Methods

1.7.1 Specifications

Six protection functions are provided to prohibit third parties from accessing the on-chip flash memory: ID code protection, on-chip debugger ID code protection, ROM code protection, serial programmer connection enable/disable, serial programmer command control, and trusted memory.

An overview of each of these protection functions is shown in Table 1.68

Table 1.68 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode or user boot mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
ROM Code Protection	When a parallel programmer is used, reading, programming, or erasing of the on-chip flash memory by third parties is prevented.
Serial Programmer Connection Enable/Disable	Connections by third parties are prohibited, and reading, programming and erasing of the on-chip flash memory is prevented, by prohibiting connections to a host in boot mode.
Serial Programmer Command Control	Reading, programming, or erasing of the on-chip flash memory after a host is connected in boot mode can be enabled/disabled individually.
Trusted Memory	Reading of the trusted memory area in the on-chip flash memory is prevented.

For instructions on using the trusted memory function, refer to the application note “RX Family: Using the Trusted Memory Function” (R01AN2618).

1.7.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection, on-chip debugger ID code protection, ROM code protection, serial programmer connection enable/disable, and serial programmer command control settings, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.16, Table 1.69, and Table 1.70. For details on each protection setting pattern, refer to 1.7.3.1 to 1.7.3.5.

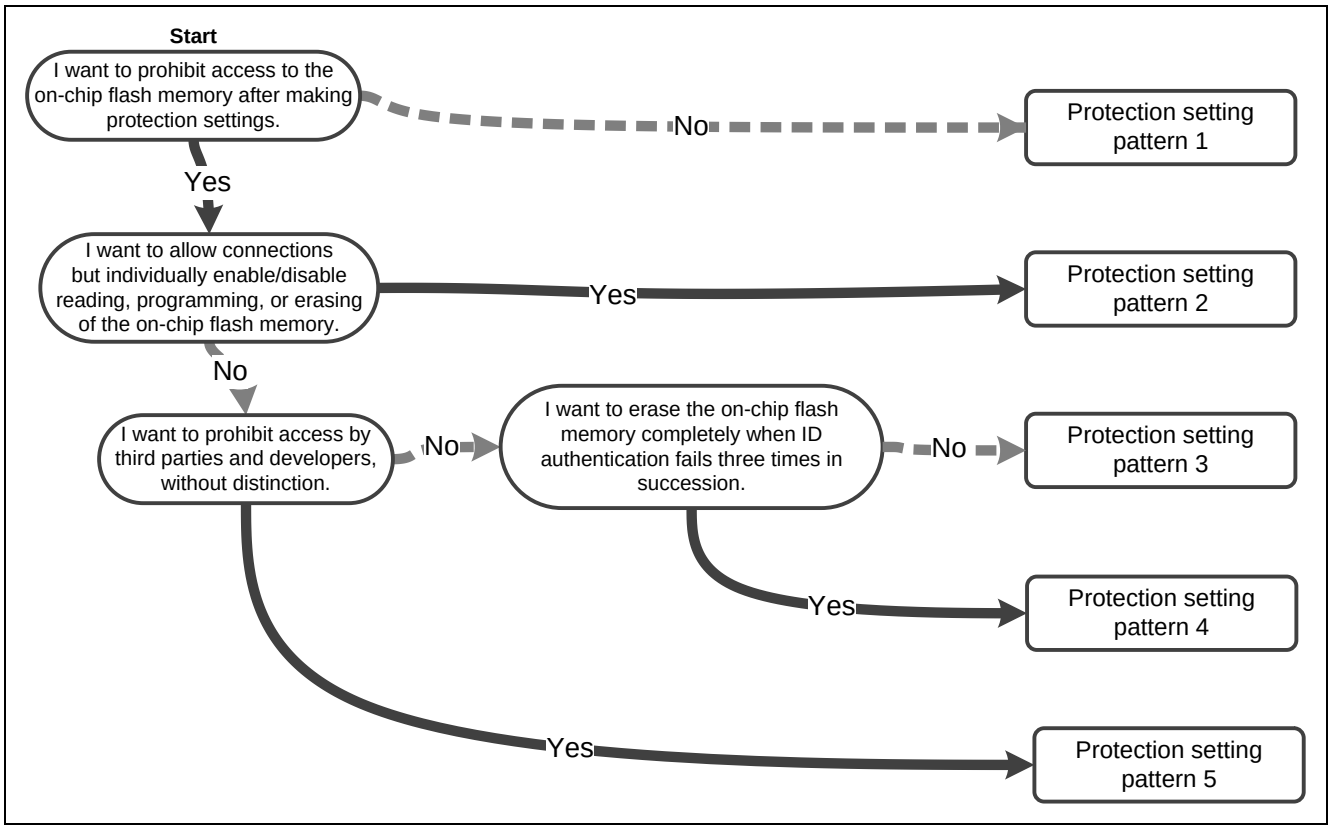


Figure 1.16 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern individually enables/disables reading, programming, or erasing of the on-chip flash memory by third parties and developers, without distinction.
- Protection setting pattern 3
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 4
  This protection setting pattern prevents reading, programming, or erasing by third parties. If ID authentication fails three times in succession, the on-chip flash memory is erased completely.
- Protection setting pattern 5
  This protection setting pattern prohibits connections by both developers and third parties.

Table 1.69 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection, Serial Programmer Connection Enable/Disable, Serial Programmer Command Control)						Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection)				Connection of parallel programmer (ROM Code Protection)			
	Developer			Third party			Developer		Third party		Developer		Third party	
	R	P	E	R	P	E	R	P/E	R	P/E	R	P/E	R	P/E
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓/—*1	✓/—*1	✓/—*1	✓/—*1	✓/—*1	✓/—*1	✓	✓	—	—	—	—	—	—
3	✓	✓	✓	—	—	—	✓	✓	—	—	—	—	—	—
4	✓	✓	✓	—*2	—	—	✓	✓	—	—	—	—	—	—
5	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—

R: Read, P: Program, E: Erase, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. The serial programmer command control register (SPCC) is used to individually enable or disable reading, programming, or erasing. For details of the serial programmer command control register (SPCC), refer to the User's Manual: Hardware of the device.

Note 2. The on-chip flash memory is erased completely if repeated ID code mismatches occur.

Table 1.70 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection, Serial Programmer Connection Enable/Disable, Serial Programmer Command Control (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection (Protection for Connection of On-Chip Debugger)	ROM Code Protection (Protection for Connection of parallel programmer)
1	Disabled	Disabled	Disabled
2	Individually enables/disables reading, programming, or erasing.	Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are prevented always.
3	Reading, programming, and erasing are enabled when the ID code matches.		
4	Reading, programming, and erasing are enabled when the ID code matches. If ID authentication fails three times in succession, the on-chip flash memory is erased completely.		
5	Reading, programming, and erasing are prevented always.		

1.7.3 Description of Protection Setting Patterns

1.7.3.1 Protection Setting Pattern 1

This pattern disables all protection.

The setting details of protection setting pattern 1 are shown in Table 1.71 and Table 1.72.

Table 1.71 Protection Setting Pattern 1 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
1	1	1	1	1

Table 1.72 Protection Setting Pattern 1 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
All FFh	Other than (0000 0000h, 0000 0001h)

For the setting method, refer to 1.7.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.73.

Table 1.73 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	If a connection occurs in boot mode, no ID authentication occurs and the device transitions to a state in which reading, programming, and erasing are possible.	None
On-Chip Debugger ID Code Protection	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None
ROM Code Protection	Reading, programming, and erasing are possible when a parallel programmer is used.	None

1.7.3.2 Protection Setting Pattern 2

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. No ID authentication is made when a connection occurs in boot mode, but reading, programming, or erasing of the on-chip flash memory can be enabled/disabled individually.

The setting details of protection setting pattern 2 are shown in Table 1.74 and Table 1.75.

Table 1.74 Protection Setting Pattern 2 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
1	1	0/1*1	0/1*2	0/1*3

Note 1. When the SEPR bit is set to 1 erasing is enabled, and erasing is disabled when it is cleared to 0.
Note 2. When the WRPR bit is set to 1 programming is enabled, and programming is disabled when it is cleared to 0.
Note 3. When the RDPR bit is set to 1 reading is enabled, and reading is disabled when it is cleared to 0.

Table 1.75 Protection Setting Pattern 2 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Other than (All FFh)	0000 0000h

For the setting method, refer to 1.7.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.76.

Table 1.76 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	No ID authentication occurs when a connection is made in boot mode. Enables/disables reading, programming, and erasing individually after the connection is established.	Individually disables enables/disables reading, programming, and erasing, without distinction between third parties and developers.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.7.3.3 Protection Setting Pattern 3

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 3 are shown in Table 1.77 and Table 1.78.

Table 1.77 Protection Setting Pattern 3 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
0	1	0	0	0

Table 1.78 Protection Setting Pattern 3 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	**ROM Code Protection Settings**
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Control code / ID code 1: Other than 45h ID code 2 to ID code 16: Any value	0000 0000h

For the setting method, refer to 1.7.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.79.

Table 1.79 Operation of Protection Setting Pattern 3

Protection Type	**Operation**	**Prevented Items**
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.7.3.4 Protection Setting Pattern 4

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode. If ID authentication fails three times in succession, the on-chip flash memory is erased completely in boot mode.

The setting details of protection setting pattern 4 are shown in Table 1.80 and Table 1.81.

Table 1.80 Protection Setting Pattern 4 Setting Details 1

Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
Serial Programmer Command Control Register (SPCC) (4 Bytes)				
ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
0	1	0	0	0

Table 1.81 Protection Setting Pattern 4 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	**ROM Code Protection Settings**
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Control code / ID code 1: 45h ID code 2 to ID code 16: Any value	0000 0000h

For the setting method, refer to 1.7.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.82.

Table 1.82 Operation of Protection Setting Pattern 4

Protection Type	**Operation**	**Prevented Items**
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again. If ID authentication fails three times in succession, the on-chip flash memory is erased completely in boot mode.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. If ID authentication fails three times in succession, the on-chip flash memory is erased completely. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.7.3.5 Protection Setting Pattern 5

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. Connections to a host are prohibited in boot mode.

The setting details of protection setting pattern 5 are shown in Table 1.83 and Table 1.84. Note that both setting number 1 and setting number 2 prohibit connections to a host in boot mode.

Table 1.83 Protection Setting Pattern 5 Setting Details 1

	Serial Programmer Connection Enable/Disable and Serial Programmer Command Control Settings				
	Serial Programmer Command Control Register (SPCC) (4 Bytes)				
Setting No.	ID Code Protection Enable bit (IDE) (bit: b24)	Serial Programmer Connection Enable bit (SPE) (bit: b27)	Block Erasure Command Protect bit (SEPR) (bit: b29)	Programming Command Protect bit (WRPR) (bit: b30)	Read Command Protect bit (RDPR) (bit: b31)
1	0	0	0	0	0
2	1	0	Don't care	Don't care	Don't care

Table 1.84 Protection Setting Pattern 5 Setting Details 2

ID Code Protection and On-Chip Debugger ID Code Protection Settings	**ROM Code Protection Settings**
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	ROM code (4 bytes)
Other than (All FFh)	0000 0000h

For the setting method, refer to 1.7.4, Protection Setting Examples.

The operation of protection setting pattern 5 is outlined in Table 1.85.

Table 1.85 Operation of Protection Setting Pattern 5

Protection Type	**Operation**	**Prevented Items**
ID Code Protection Serial Programmer Connection Enable/Disable Serial Programmer Command Control	Connections to a host are prohibited when connecting in boot mode.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.7.4 Protection Setting Examples

Each protection function is enabled by setting serial programmer connection enable/disable, serial programmer command control, and an ID code to addresses in the option-setting memory, and a ROM code to an address in the on-chip flash memory. Serial programmer connection enable/disable and serial programmer command control should be setting to 0x00120040, the ID code to 0x00120050, and the ROM code to 0x0012007C.

Also, for instructions on writing data to the option-setting memory, refer to the User's Manual: Hardware of the device.

Protection setting examples are shown in Figure 1.17 and Figure 1.18.

```
/* Setup the Serial programmer command control register */
#pragma address SPCC_REG = 0x00120040
const unsigned long SPCC_REG = 0x1EFFFFFF

/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0x00120050
const unsigned long ID_CODE[4] = {0x04030245, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection */
#pragma address ROM_CODE = 0x0012007C
const unsigned long ROM_CODE = 0x00000000;
```

In this example, the Control code / ID code 1 is 45h,
from ID code 2 to ID code 16 are 02h, 03h, 04h, 05h,
06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.17 Protect Setting Pattern 4 Setting Example

```
/* Setup the Serial programmer command control register */
#pragma address SPCC_REG = 0x00120040
const unsigned long SPCC_REG = 0x16FFFFFF

/* Setup the ID Code Protection and the ID Code Protection on Connection of the On-Chip Debugger */
#pragma address ID_CODE = 0x00120050
const unsigned long ID_CODE[4] = {0x04030201, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection */
#pragma address ROM_CODE = 0x0012007C
const unsigned long ROM_CODE = 0x00000000;
```

In this example, the Control code / ID code 1 is 01h,
from ID code 2 to ID code 16 are 02h, 03h, 04h, 05h,
06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.18 Protect Setting Pattern 5 Setting Example

1.8 Device Group G Protection Methods

1.8.1 Specifications

Seven protection functions are provided to prohibit third parties from accessing the on-chip flash memory: ID code protection, on-chip debugger ID code protection, ROM code protection, serial programmer connection enable/disable, on-chip debugger connection enable/disable, trusted memory and FSPR bit.

The access window is not a function to prevent access from third parties, but it is possible to prevent access from third parties by setting the FSPR bit.

An overview of each of these protection functions is shown in Table 1.86

Table 1.86 Overview of Protection Functions

Protection Type	Overview of Function
ID Code Protection	After the MCU starts up in boot mode, ID authentication is performed when a host such as a PC is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
On-Chip Debugger ID Code Protection	After the MCU starts up in single-chip mode, ID authentication is performed when an on-chip debugger is connected, prohibiting connection by third parties and preventing reading, programming, or erasing of the on-chip flash memory.
ROM Code Protection	When a parallel programmer is used, reading, programming, or erasing of the on-chip flash memory by third parties is prevented.
Serial Programmer Connection Enable/Disable	Connections by third parties are prohibited, and reading, programming and erasing of the on-chip flash memory is prevented, by prohibiting connections to a host in boot mode.
On-Chip Debugger Connection Enable/Disable	Connections by third parties are prohibited, and reading, programming and erasing of the on-chip flash memory is prevented, by prohibiting connections to an on-chip debugger.
Trusted Memory	Reading of the trusted memory area in the on-chip flash memory is prevented.
Access Window	If the access window is set, the area set outside the access window is prevented programming or erasing. The access window is a function to prevent erroneous rewriting in case a program runs out of control during self-programming.
FSPR bit	If the FSPR bit is set, the access window never be set again. By setting the access window and the FSPR bit at the same time, the area outside the access window can be used as an area that can never be programmed or erased again by both developers and third parties.

For details on the access window, see Chapter 2, Area Protection.

1.8.2 Selecting Protection Settings

The method of access prohibition differs according to the details of the ID code protection, on-chip debugger ID code protection, ROM code protection, and serial programmer connection enable/disable, on-chip debugger connection enable/disable, as well as how they are combined. It is therefore necessary to make protection settings appropriately to match the desired purpose.

Select an optimal protection setting pattern based on the chart in Figure 1.19, Table 1.87, and Table 1.88. For details on each protection setting pattern, refer to 1.8.3.1 to 1.8.3.4.

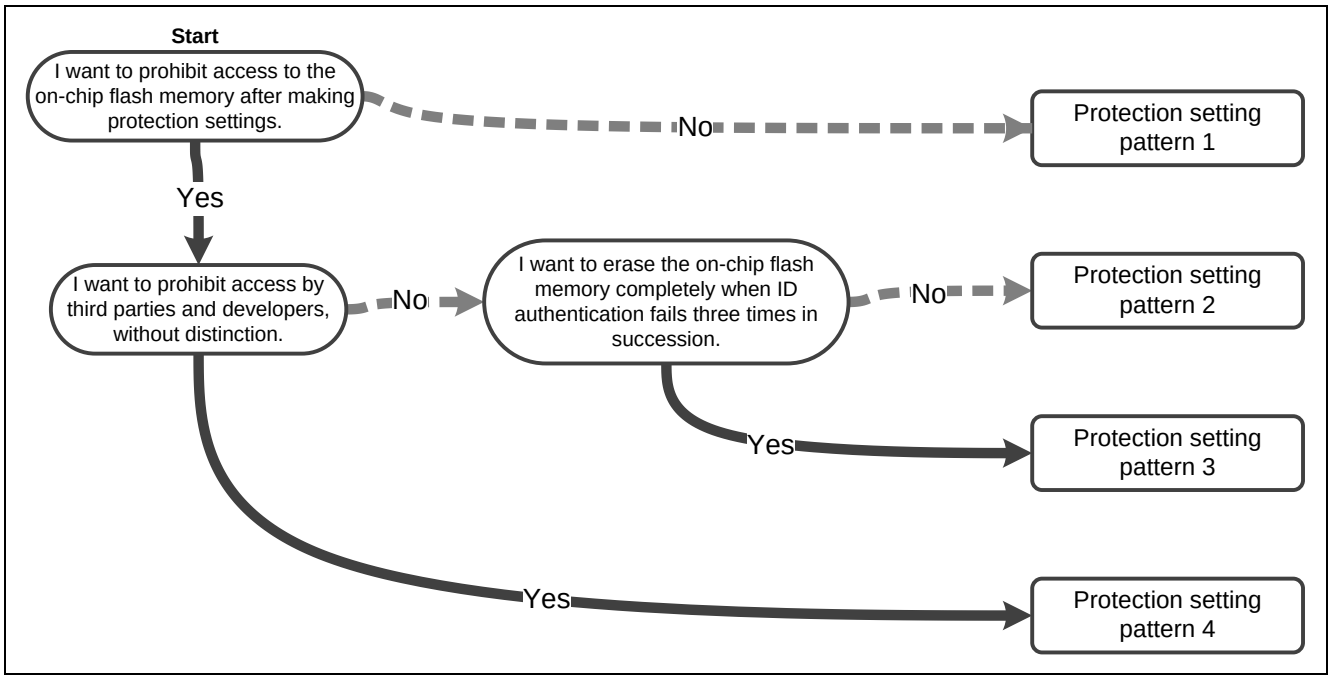


Figure 1.19 Protection Setting Pattern Selection Chart

- Protection setting pattern 1
  All protection against access by developers and third parties is disabled.
- Protection setting pattern 2
  This protection setting pattern prevents reading, programming, or erasing by third parties.
- Protection setting pattern 3
  This protection setting pattern prevents reading, programming, or erasing by third parties. If ID authentication fails three times in succession, the on-chip flash memory is erased completely.
- Protection setting pattern 4
  This protection setting pattern prohibits connections by both developers and third parties. Once this protection setting pattern is applied, the protection cannot be removed, so caution is necessary.

Table 1.87 Comparison of Protection Setting Patterns

Protection Setting Pattern	Connection in Boot Mode (ID Code Protection, Serial Programmer Connection Enable/Disable)				Connection of On-Chip Debugger (On-Chip Debugger ID Code Protection, On-Chip Debugger Connection Enable/Disable)				Connection of parallel programmer (ROM Code Protection)			
	Developer		Third party		Developer		Third party		Developer		Third party	
	R	P/E	R	P/E	R	P/E	R	P/E	R	P/E	R	P/E
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	—	—	✓	✓	—	—	—	—	—	—
3	✓	✓	—*1	—	✓	✓	—	—	—	—	—	—
4	—	—	—	—	—	—	—	—	—	—	—	—

R: Read, P/E: Program/Erase
✓: Allowed, —: Not allowed

Note 1. The on-chip flash memory is erased completely if repeated ID code mismatches occur.

Table 1.88 Functions of Protection Setting Patterns

Protection Setting Pattern	ID Code Protection, Serial Programmer Connection Enable/Disable (Protection for Connection in Boot Mode)	On-Chip Debugger ID Code Protection On-Chip Debugger Connection Enable/Disable (Protection for Connection of On-Chip Debugger)	ROM Code Protection (Protection for Connection of parallel programmer)
1	Disabled	Disabled	Disabled
2	Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are enabled when the ID code matches.	Reading, programming, and erasing are prevented always.
3	Reading, programming, and erasing are enabled when the ID code matches. If ID authentication fails three times in succession, the on-chip flash memory is erased completely.		
4	Reading, programming, and erasing are prevented always.	Reading, programming, and erasing are prevented always.	

1.8.3 Description of Protection Setting Patterns

1.8.3.1 Protection Setting Pattern 1

This pattern disables all protection.

The setting details of protection setting pattern 1 are shown in Table 1.89.

Table 1.89 Protection Setting Pattern 1 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable On-chip debugger Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
All FFh	FFFF FFFFh	Other than (0000 0000h, 0000 0001h)

For the setting method, refer to 1.8.4, Protection Setting Examples.

The operation of protection setting pattern 1 is outlined in Table 1.90.

Table 1.90 Operation of Protection Setting Pattern 1

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	If a connection occurs in boot mode, the device transitions to a state in which reading, programming, and erasing are possible by transmitting the ID code all set to FFh.	None
On-chip debugger ID code protection On-Chip Debugger Connection Enable/Disable	When an on-chip debugger is connected, no ID authentication occurs and the connection with the on-chip debugger is established.	None
ROM Code Protection	Reading, programming, and erasing are possible when a parallel programmer is used.	None

1.8.3.2 Protection Setting Pattern 2

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode.

The setting details of protection setting pattern 2 are shown in Table 1.91.

Table 1.91 Protection Setting Pattern 2 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable On-chip debugger Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
Control code / ID code 1: Other than 45h ID code 2 to ID code 16: Any value	FFFF FFFFh	0000 0000h

For the setting method, refer to 1.8.4, Protection Setting Examples.

The operation of protection setting pattern 2 is outlined in Table 1.92.

Table 1.92 Operation of Protection Setting Pattern 2

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-chip debugger ID code protection On-Chip Debugger Connection Enable/Disable	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.8.3.3 Protection Setting Pattern 3

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, when an on-chip debugger is connected, protection is provided by ID authentication. As is the case when an on-chip debugger is connected, protection is provided by ID authentication when a connection is made in boot mode. If ID authentication fails three times in succession, the on-chip flash memory is erased completely in boot mode.

The setting details of protection setting pattern 3 are shown in Table 1.93.

Table 1.93 Protection Setting Pattern 3 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable On-chip debugger Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
Control code / ID code 1: 45h ID code 2 to ID code 16: Any value	FFFF FFFFh	0000 0000h

For the setting method, refer to 1.8.4, Protection Setting Examples.

The operation of protection setting pattern 3 is outlined in Table 1.94.

Table 1.94 Operation of Protection Setting Pattern 3

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	Protection is provided by ID authentication when a connection is made in boot mode. If the ID code matches, the device transitions to a state in which reading, programming, and erasing are possible. If the ID code does not match, ID authentication is performed again. If ID authentication fails three times in succession, the on-chip flash memory is erased completely in boot mode.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. If ID authentication fails three times in succession, the on-chip flash memory is erased completely. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
On-chip debugger ID code protection On-Chip Debugger Connection Enable/Disable	When a connection is made by an on-chip debugger, ID authentication is performed. If the ID code matches, a connection is established with the on-chip debugger. If the ID code does not match, ID authentication is performed again.	Third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ID authentication. Developers can read, program, or erase the on-chip flash memory by providing the matching ID code.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.8.3.4 Protection Setting Pattern 4

This pattern prevents reading, programming, or erasing of the on-chip flash memory by a parallel programmer. In addition, connections to an on-chip debugger are prohibited when connecting to an on-chip debugger. Connections to a host are prohibited in boot mode.

Note: After this setting is made and the device is reset, the protection cannot be removed by any method, so caution is necessary.

The setting details of protection setting pattern 4 are shown in Table 1.95.

Table 1.95 Protection Setting Pattern 4 Setting Details

ID Code Protection and On-Chip Debugger ID Code Protection Settings	Serial Programmer Connection Enable/Disable On-chip debugger Connection Enable/Disable	ROM Code Protection Settings
OCD/Serial Programmer ID Setting Register (OSIS) (16 bytes)	Serial Programmer Command Control Register (SPCC) (4 Bytes)	ROM Code Protection Register (ROMCODE) (4 bytes)
All FFh	F7FD FFFFh (SPE bit = 0, OCDE bit = 0)	0000 0000h

For the setting method, refer to 1.8.4, Protection Setting Examples.

The operation of protection setting pattern 4 is outlined in Table 1.96.

Table 1.96 Operation of Protection Setting Pattern 4

Protection Type	Operation	Prevented Items
ID Code Protection Serial Programmer Connection Enable/Disable	Connections to a host are prohibited when connecting in boot mode.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
On-Chip Debugger ID Code Protection On-Chip Debugger Connection Enable/Disable	Connections to an on-chip debugger are prohibited when connecting to an on-chip debugger.	Connections are prohibited, preventing both third parties and developers, without distinction, from reading, programming, and erasing.
ROM Code Protection	Reading, programming, and erasing are prohibited when using a parallel programmer.	Both developers and third parties are prevented from reading, programming, or erasing the on-chip flash memory by means of ROM code protection.

1.8.4 Protection Setting Examples

Each protection function is enabled by setting serial programmer connection enable/disable, on-chip debugger connection enable/disable, the ID code, and the ROM code to addresses in the option-setting memory. Serial programmer connection enable/disable and on-chip debugger connection enable/disable should be set to 0xFE7F5D40, the ID code to 0xFE7F5D50, and the ROM code to 0xFE7F5D70.

Also, for instructions on writing data to the option-setting memory, refer to the User's Manual: Hardware of the device.

Protection setting examples are shown in Figure 1.20 and Figure 1.21.

```
/* Setup the Serial programmer command control Register */
#pragma address SPCC_REG = 0xFE7F5D40
const unsigned long SPCC_REG = 0xFFFFFFFF;

/* Setup the OCD/Serial Programmer ID Setting Register */
#pragma address OSIS_REG = 0xFE7F5D50
const unsigned long OSIS_REG[4] = {0x04030245, 0x08070605, 0x0C0B0A09, 0x100F0E0D};

/* Setup the ROM Code Protection Register */
#pragma address ROMCODE_REG = 0xFE7F5D70
const unsigned long ROMCODE_REG = 0x00000000;
```

In this example, the Control code / ID code 1 is 45h, from ID code 2 to ID code 16 are 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h.

Figure 1.20 Protect Setting Pattern 3 Setting Example

```
/* Setup the Serial programmer command control Register */
#pragma address SPCC_REG = 0xFE7F5D40
const unsigned long SPCC_REG = 0xF7FDFFFF;

/* Setup the OCD/Serial Programmer ID Setting Register */
#pragma address OSIS_REG = 0xFE7F5D50
const unsigned long OSIS_REG[4] = {0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF};

/* Setup the ROM Code Protection Register */
#pragma address ROMCODE_REG = 0xFE7F5D70
const unsigned long ROMCODE_REG = 0x00000000;
```

In this example, the Control code / ID code 1, from ID code 2 to ID code 16 are all FFh.

Figure 1.21 Protect Setting Pattern 4 Setting Example

2. Protection during Self-Programming Initiated by Developers

2.1 Device Categories

Devices are categorized into three groups according to their protection functions that work during self-programming initiated by developers.

The device categories are listed in Table 2.1. For details of each protection function, refer to the User's Manual: Hardware of the device.

Table 2.1 Device Categories

		Protection Function (✓: Supported, —: Not Supported)										
Group	Device	Lock Bits	Area Protection	FENTRYR Register	E2 DataFlash Access Disable Mode	FLWE bit	RPDIS bit	DBWE bit	DBRE bit	DFLEN bit	Start-Up Program Protection	Dual Bank Function
Device group A	• RX210 • RX21A • RX220 • RX610 • RX621 • RX62G • RX62N • RX62T • RX630 • RX631 • RX634 • RX63N • RX63T	✓	—	✓	—	✓	—	✓	✓	—	—	—
	• RX64M • RX66T • RX71M • RX72T	✓	—	✓	—	✓	—	—	—	—	—	—
Device group B	• RX111 • RX113 • RX130 • RX13T • RX140 • RX230 • RX231 • RX23E-A • RX23W • RX24T • RX24U	—	✓	✓	✓	—	✓	—	—	✓	✓	—
	• RX110 • RX23T	—	✓	✓	—	—	✓	—	—	—	✓	—
Device group C	• RX651 • RX65N • RX66N • RX671 • RX72M • RX72N	—	✓	✓	—	✓	—	—	—	—	✓	✓

2.2 Device Group A Protection Methods

2.2.1 Protection during Self-Programming

Five protection functions are provided for protection during self-programming: the lock bits, FENTRYR register, FLWE bit, DBWE bit, and DBRE bit.

An overview of these protection functions is shown in Table 2.2.

Table 2.3 indicates the areas to which the protection can be configured. Configure the area for which rewriting is to be prohibited during self-programming.

Table 2.2 Overview of Protection Functions

Protection Type	Purpose	Overview of Function
Lock Bits	Prohibits programming/erasure of the user area in block units.	Each block in the user area includes a lock bit. If you enable lock bit protection, programming/erasure of the block whose lock bit is set to “0”.
FENTRYR Register	Prohibits programming/erasure of the user area and data area by setting the flash memory to read mode.	When the FENTRYR register is set to “0000h”, the flash memory is in read mode. In read mode, programming/erasure of the flash memory is prohibited.
FLWE bit	Prohibits programming/erasure of an area where the P/E mode is set by the FENTRYR register.	If you set the FLWE bit, programming/erasure of the flash memory and lock bits, reading of the lock bits, and blank check are prohibited.
DBWE bit*1	Prohibits programming/erasure of the data area in units of multiple blocks.	If you set the DBWE bit, programming/erasure of the specified data area blocks are prohibited.
DBRE bit*2	Prohibits reading of the data area in units of multiple blocks.	If you set the DBRE bit, reading of the specified data area blocks are prohibited. Programming/erasure cannot be prohibited.

Note 1. The DBWE bit is provided in RX210, RX21A, RX220, RX610, RX621, RX62G, RX62N, RX62T, RX630, RX631, RX634, RX63N, and RX63T.

Note 2. The DBRE bit is provided in RX210, RX21A, RX220, RX610, RX621, RX62G, RX62N, RX62T, RX630, RX631, RX634, RX63N, and RX63T.
For setting details, refer to the User’s Manual: Hardware of the device.

Table 2.3 Area to Which Protection Can be Configured

Protection Function	Flash Memory Reading/Programming/Erasure Prohibition			
	User Area		Data Area	
	Reading	Programming/Erasure	Reading	Programming/Erasure
Lock Bits	—	✓	—	—
FENTRYR Register	—	✓	—	✓
FLWE bit	—	✓	—	✓
DBWE bit	—	—	—	✓
DBRE bit	—	—	✓	—

✓: Can be configured, —: Cannot be configured

2.2.1.1 Description of the Protection Functions

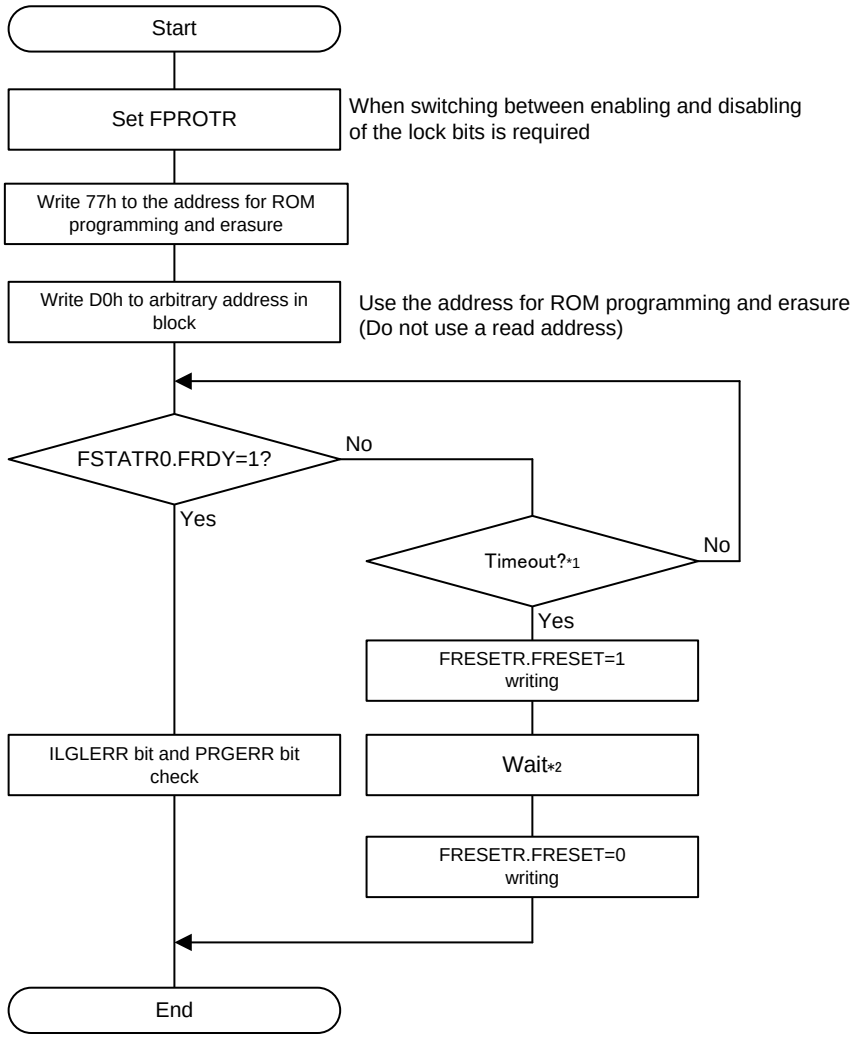
(1) Lock Bits

You can prohibit programming/erasure of the user area in block units.

Each block in the user area includes a lock bit. If the FPROTCN bit of the FPROTR register is set to “0”, programming/erasure of the block whose lock bit is set to “0” is prohibited. The lock bit is set by using the lock bit programming command.

Figure 2.1 shows the flow for issuing the lock bit programming command in RX210, RX21A, RX220, RX610, RX621, RX62G, RX62N, RX62T, RX630, RX631, RX634, RX63N, and RX63T.

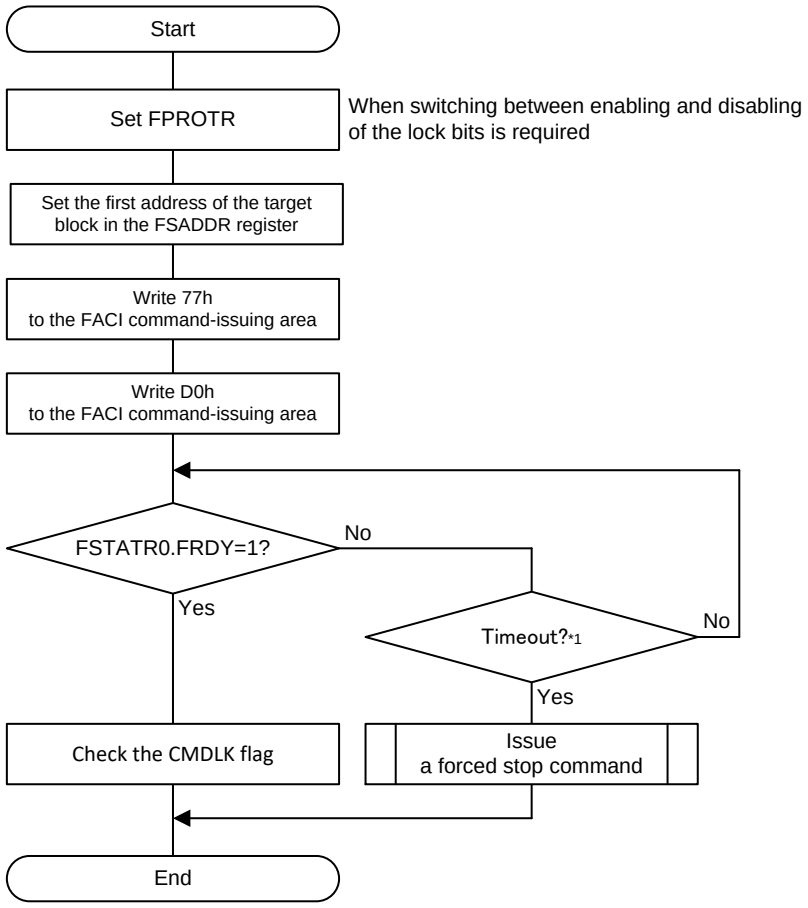
Figure 2.2 shows the flow for issuing the lock bit programming command in RX64M, RX66T, RX71M, and RX72T.



Note 1. For the timeout time (time required for programming data), refer to the “Electrical Characteristics” of your device.

Note 2. For the wait time (reset pulse width during programming/erasure), refer to the “Electrical Characteristics” of your device.

Figure 2.1 Flow for Issuing the Lock Bit Programming Command 1



Note 1. Judgment is based on 25ms (when 20MHz ≤ FCLK ≤ 60MHz) or 50ms (when FCLK = 4MHz).

Figure 2.2 Flow for Issuing the Lock Bit Programming Command 2

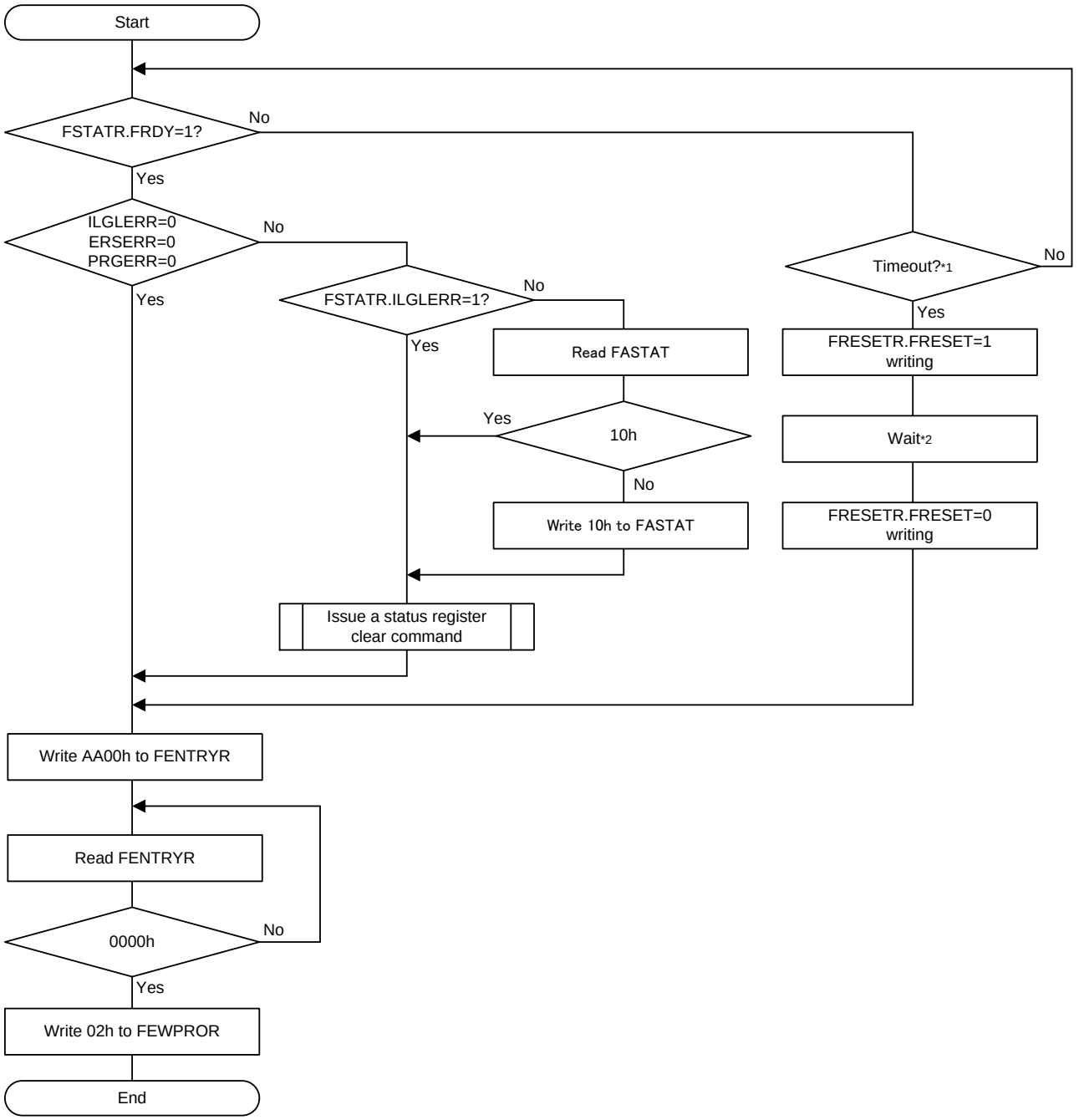
(2) FENTRYR Register

You can prohibit programming/erasure of the user area and data area by setting the flash memory to read mode.

When the FENTRYR register is set to “0000h”, the flash memory is in read mode. In read mode, programming/erasure of the flash memory is prohibited.

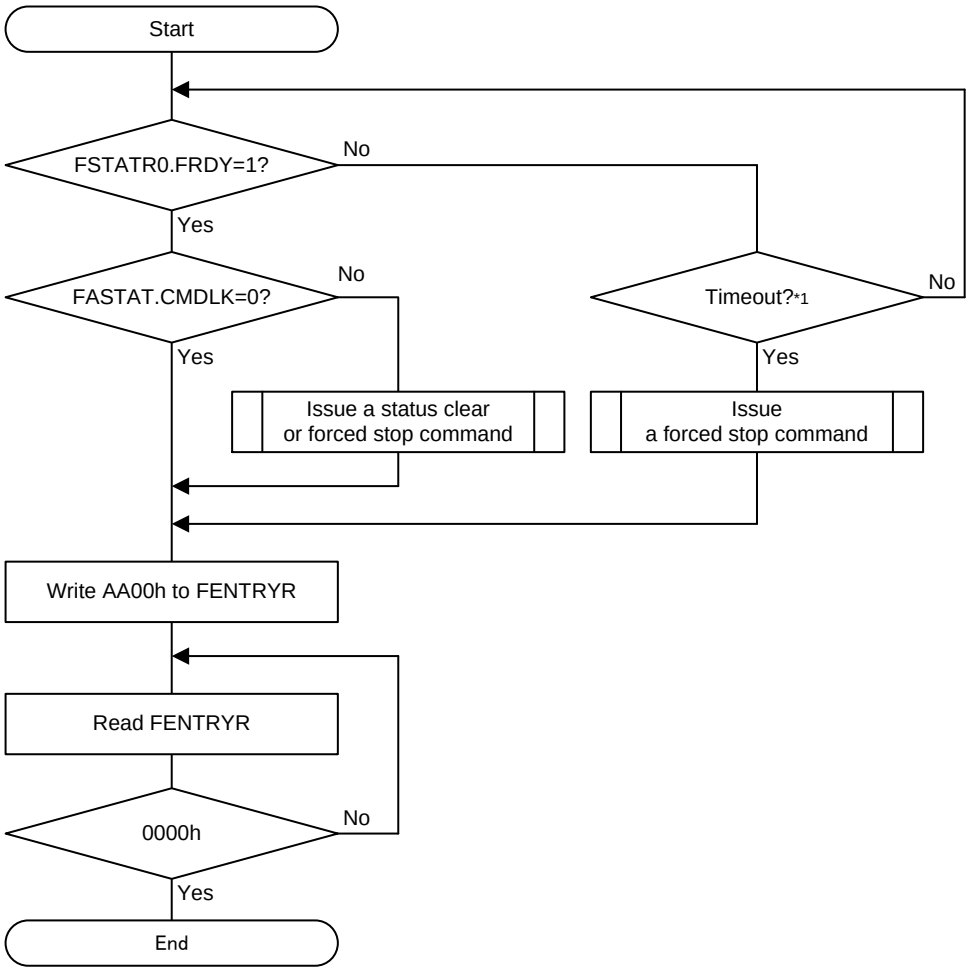
Figure 2.3 shows the flow for transition to read mode in RX210, RX21A, RX220, RX610, RX621, RX62G, RX62N, RX62T, RX630, RX631, RX634, RX63N, and RX63T.

Figure 2.4 shows the flow for transition to read mode in RX64M, RX66T, RX71M, and RX72T.



Note 1. For the timeout time (data erasure time), refer to the “Electrical Characteristics” of your device.
Note 2. For the wait time (reset pulse width during programming/erasure), refer to the “Electrical Characteristics” of your device.

Figure 2.3 Flow for Transition to Read Mode 1



Note 1. For the timeout time (the maximum time for processing of the FACI command that is in progress), refer to the “Electrical Characteristics” of your device.)

Figure 2.4 Flow for Transition to Read Mode 2

(3) FLWE bit

You can prohibit programming/erasure of an area where the P/E mode is set by the FENTRYR register.

If you set the FLWE bit of the FWEPROR register, programming/erasure of the flash memory and lock bits, reading of the lock bits, and blank check are prohibited by software.

Table 2.4 shows the setting of the FWEPROR register.

Table 2.4 FWEPROR Register Setting

Register Name	Flash P/E protect register (FWEPROR)*1
Bit Name	Flash P/E bit (FLWE) (Bit: b0-b1)
Setting Value	00h, 10h (default value), or 11h

Note 1. The FWEPROR register is called with a different name in RX210, RX21A, RX220, RX610, RX621, RX62G, RX62N, and RX62T.

(4) DBWE bit

You can prohibit programming/erasure of the data area in units of multiple blocks.

If you set the DBWE bit of the DFLWE register, programming/erasure of the respective data area blocks is prohibited.

Table 2.5 shows the setting of the DFLWE register.

Table 2.5 DFLWE Register Setting

Register Name	E2 DataFlash P/E enable register (DFLWE)*1	
Bit Name	Block P/E enable bit (DBWEj) (j = Varies by device) (Bit: b0-b7)	Key code (KEY) (Bit: b8-b15)
Setting Value	Set the enable bit corresponding to the block for which programming/erasure is to be prohibited to “0”.	1Eh

Note 1. The DFLWE register is called with a different name in RX210, RX21A, RX220, RX610, RX621, RX62G, RX62N, and RX62T.

2.3 Device Group B Protection Methods

2.3.1 Protection during Self-Programming

Five protection functions are provided for protection during self-programming: area protection, the FENTRYR register, E2 DataFlash access disable mode, RPDIS bit, and DFLEN bit.

An overview of these protection functions is shown in Table 2.6.

Table 2.7 indicates the areas to which the protection can be configured. Configure the area for which rewriting is to be prohibited during self-programming.

Table 2.6 Overview of Protection Functions

Protection Type	Purpose	Overview of Function
Area Protection	Prohibits programming/erasure of the selected blocks in the user area.	Rewriting of the user area other than the selected blocks (access window) is prohibited during self-programming.
FENTRYR Register	Prohibits programming/erasure of the user area and data area by setting the flash memory to read mode.	When the FENTRYR register is set to “0000h”, the flash memory is in read mode. In read mode, programming/erasure of the flash memory is prohibited.
E2 DataFlash Access Disable Mode*1	Prohibits programming/erasure of the user area, and prohibits reading/programming/erasure of the data area.	The E2 DataFlash access disable mode is the mode where access to the data area is prohibited. For the user area, programming/erasure is prohibited.
RPDIS bit	Prohibits programming/erasure of the user area.	If you set the RPDIS bit, programming/erasure of the user area is prohibited.
DFLEN bit*2	Prohibits access to the data area.	If you set the DFLEN bit, access to the data area (i.e. reading/programming/erasure) is prohibited, and access to the extra area in P/E mode (i.e. start-up area information program, access window protection*3, and access window information program) is prohibited.

Note 1. The E2 DataFlash access disable mode is provided in RX111, RX113, RX130, RX13T, RX140, RX230, RX231, RX23E-A, RX23W, RX24T, and RX24U.

Note 2. The DFLEN bit is provided in RX111, RX113, RX130, RX13T, RX140, RX230, RX231, RX23E-A, RX23W, RX24T, and RX24U.

Note 3. Access window protection is provided in RX140.

Table 2.7 Area to Which Protection Can be Configured

Protection Function	Flash Memory Reading/Programming/Erasure Prohibition			
	User Area		Data Area	
	Reading	Programming/Erasure	Reading	Programming/Erasure
Area Protection	—	✓	—	—
FENTRYR Register	—	✓	—	✓
E2 DataFlash Access Disable Mode	—	✓	✓	✓
RPDIS bit	—	✓	—	—
DFLEN bit	—	—	✓	✓

✓: Can be configured, —: Cannot be configured

2.3.1.1 Description of the Protection Functions

(1) Area Protection

You can prohibit programming/erasure of the selected blocks in the user area.

Rewriting of the user area other than the selected blocks (access window) is prohibited during self-programming. The access window is set by using the access window information program command.

Table 2.8 shows the setting of the access window in RX140, RX230, RX231, RX23E-A, RX23T, RX23W, RX24T, and RX24U.

Table 2.9 shows the setting of the access window in RX110, RX111, RX113, RX130, and RX13T.

Table 2.8 Access Window Setting 1

Register Name	Flash write buffer 0 register (FWB0)	Flash write buffer 1 register (FWB1)
Setting Value	From b19 to b10 of the access window start address	From b19 to b10 of the address that follows the access window end address

Table 2.9 Access Window Setting 2

Register Name	Flash write buffer register H (FWBH)	Flash write buffer register L (FWBL)
Setting Value	From b21 to b10 of the access window start address	From b21 to b10 of the address that follows the access window end address

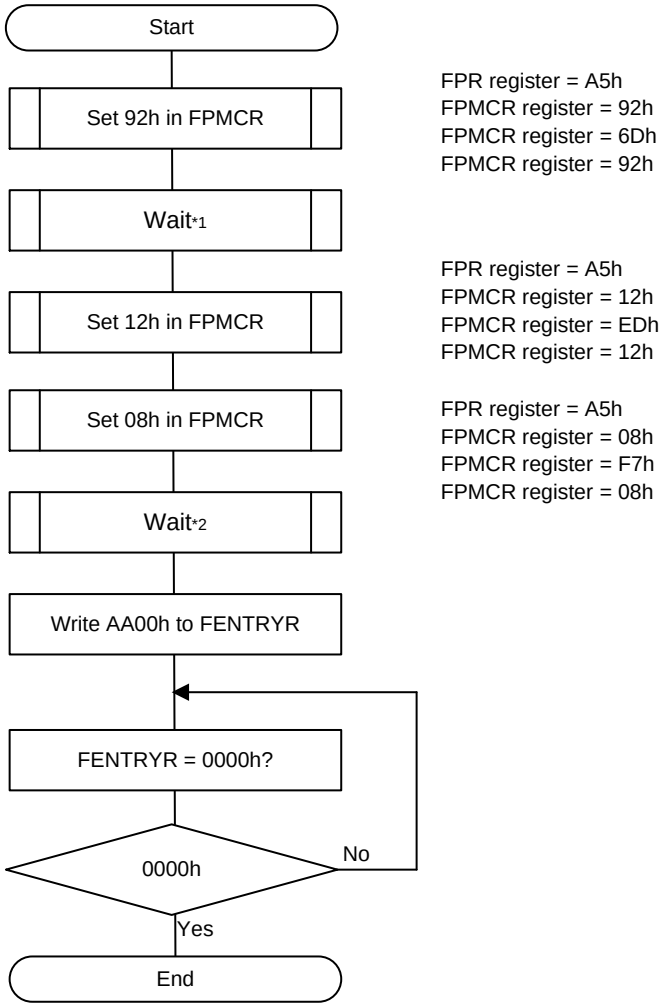
For how to issue the access window information program command, refer to 2.3.2.2, How to Issue the Start-Up Area Information Program Command / Access Window Information Program Command.

(2) FENTRYR Register

You can prohibit programming/erasure of the user area and data area by setting the flash memory to read mode.

When the FENTRYR register is set to “0000h”, the flash memory is in read mode. In read mode, programming/erasure of the flash memory is prohibited.

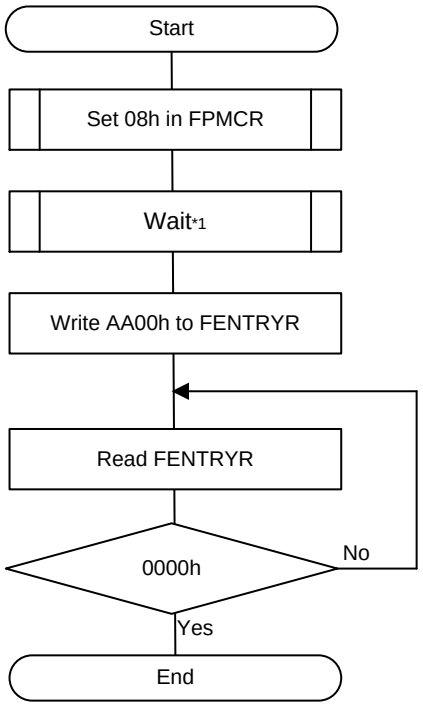
Figure 2.5 and Figure 2.6 show how to set the flash memory to read mode.



Note 1. For the wait time (ROM mode transition wait time 1), refer to the “Electrical Characteristics” of your device.

Note 2. For the wait time (ROM mode transition wait time 2), refer to the “Electrical Characteristics” of your device.

Figure 2.5 Flow for Transition from ROM P/E Mode to Read Mode



Note 1. For the wait time (ROM mode transition wait time 2), refer to the “Electrical Characteristics” of your device.

Figure 2.6 Flow for Transition from E2 DataFlash P/E Mode to Read Mode

(3) E2 DataFlash Access Disable Mode

You can prohibit programming/erasure of the user area and reading/programming/erasure of the data area.

When the FENTRYR register is “0000h” and the DFLCTL register is “00h”, the flash memory is in the E2 DataFlash access disable mode. In the E2 DataFlash access disable mode, programming/erasure of the user area and reading/programming/erasure of the data area are prohibited.

For how to set the FENTRYR register, refer to 2.3.1.1(2), FENTRYR Register.

Table 2.10 shows the setting of the DFLCTL register.

Table 2.10 DFLCTL Register Setting

Register Name	E2 DataFlash control register (DFLCTL)
Bit Name	E2 DataFlash access enable bit (DFLEN) (Bit: b0)
Setting Value	0

(4) RPDIS bit

You can prohibit programming/erasure of the user area.

If you set the RPDIS bit of the FPMCR register, programming/erasure of the user area is prohibited.

Table 2.11 shows the setting of the FPMCR register.

Table 2.11 FPMCR Register Setting

Register Name	Flash P/E mode control register (FPMCR)				
Bit Name	Flash operating mode select 0 bit (FMS0) (Bit: b1)	ROM P/E disable bit (RPDIS) (Bit: b3)	Flash operating mode select 1 bit (FMS1) (Bit: b4)	Low-voltage P/E mode enable bit (LVPE) (Bit: b6)	Flash operating mode select 2 bit (FMS2) (Bit: b7)
Setting Value	Don’t care	1	Don’t care	Don’t care	Don’t care

(5) DFLEN bit

You can prohibit access to the data area.

If you set the DFLEN bit of the DFLCTL register, access to the data area (i.e. reading/programming/erasure) is prohibited, and access to the extra area in P/E mode (i.e. start-up area information program, access window protection, and access window information program) is prohibited.

Table 2.12 shows the setting of the DFLCTL register.

Table 2.12 DFLCTL Register Setting

Register Name	E2 DataFlash control register (DFLCTL)
Bit Name	E2 DataFlash access enable bit (DFLEN) (Bit: b0)
Setting Value	0

2.3.2 Protection during Update

The start-up program protection is provided for protection during update.

An overview of the protection function is shown in Table 2.13.

Table 2.13 Overview of Protection Functions

Protection Type	Overview of Function
Start-up Program Protection	When you update the start-up program, this function allows you to update to a new program without erasing the current start-up program. If the update operation is disrupted for any reason including a reset, the program is still safely updated.

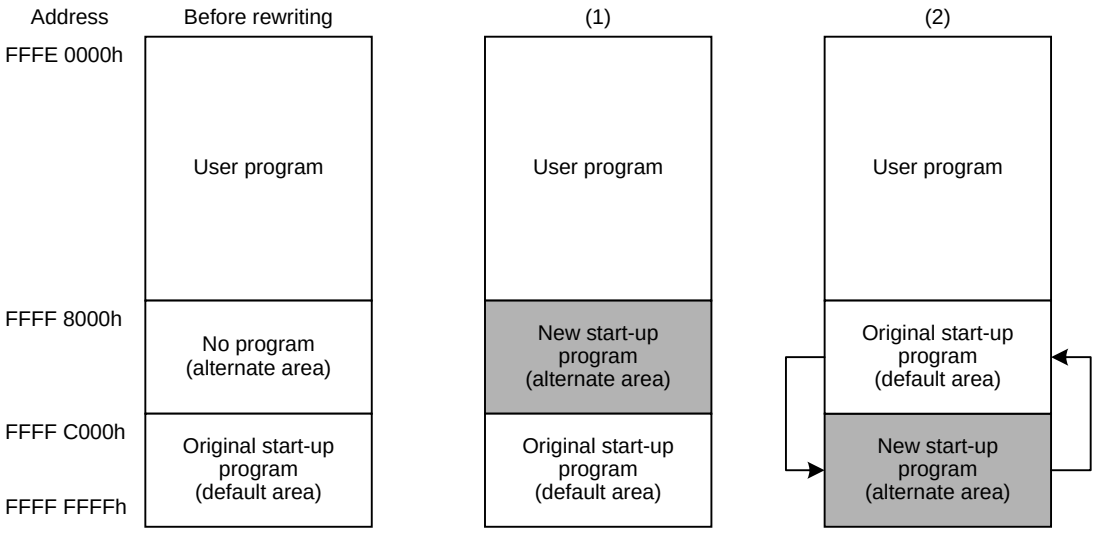
For how to use the start-up program protection, refer to “RX100/RX200 Series Updating Firmware Using Start-Up Program Protection and Serial Communication (R01AN3740)”.

2.3.2.1 Description of the Protection Function

(1) Start-Up Program Protection

When updating the start-up program, the default area (where the current start-up program is stored) is swapped with the alternate area (where the new start-up program is stored) so that the start-up program can be updated without erasing the current start-up program. If the update operation is disrupted for any reason including a reset, the program is still safely updated. The start-up area is swapped by using the start-up area information program command.

Figure 2.7 gives an overview of the start-up program protection.



(1) Program a new start-up program in the alternate area. If the alternate area fails to be rewritten, the new start-up program can be rewritten again after starting up using the default area, because the original start-up program is in the default area.

(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the start-up area information program command. After that, the program in the alternate area starts after a reset.

Figure 2.7 Overview of the Start-Up Program Protection (for RX140)

For how to issue the start-up area program command, refer to 2.3.2.2, How to Issue the Start-Up Area Information Program Command / Access Window Information Program Command.

2.3.2.2 How to Issue the Start-Up Area Information Program Command / Access Window Information Program Command

Figure 2.8 shows the flow for issuing the start-up area information program command and access window information program command.

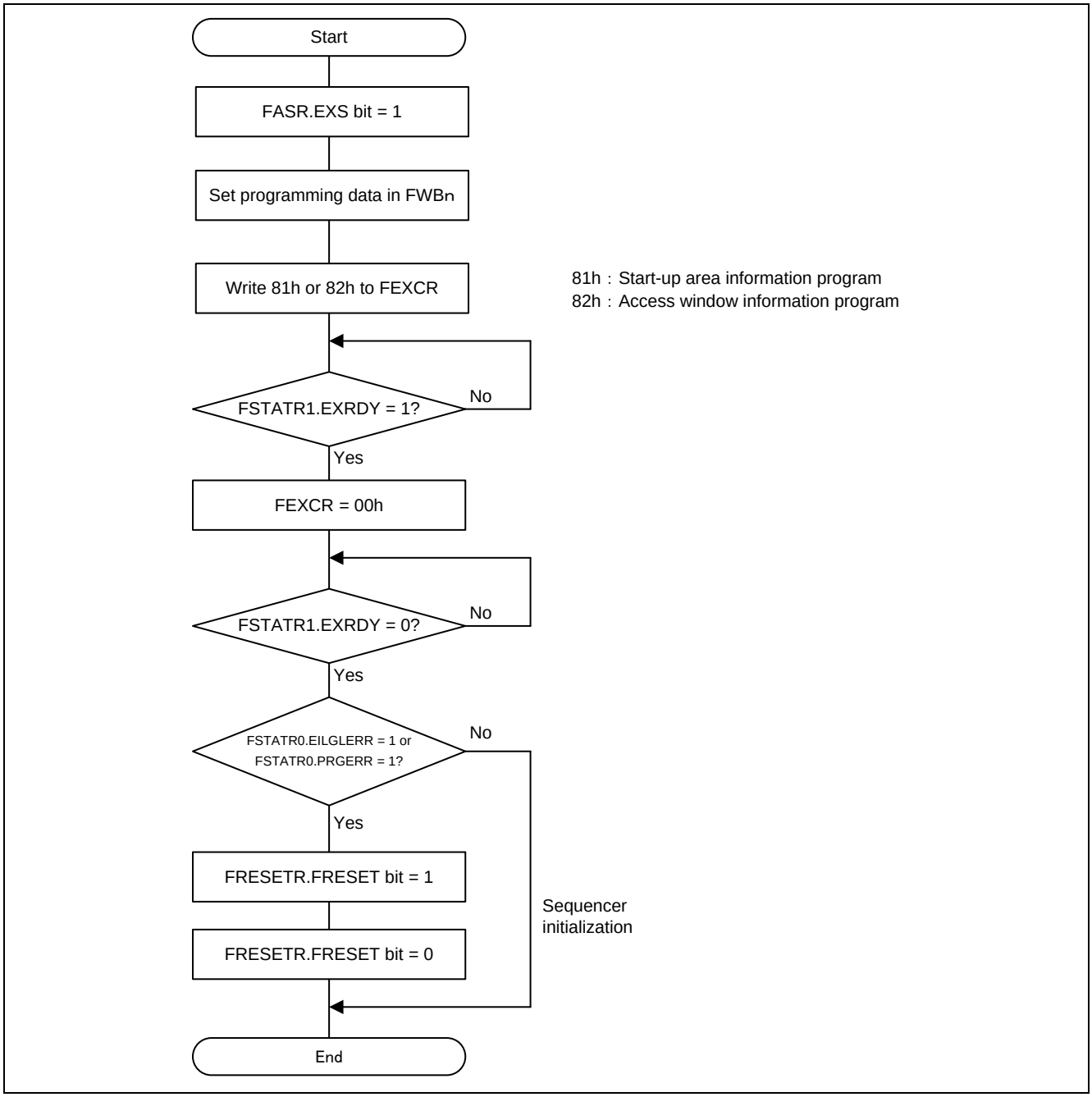


Figure 2.8 Flow for Issuing the Start-Up Area Information Program Command / Access Window Information Program Command

2.4 Device Group C Protection Methods

2.4.1 Protection during Self-Programming

Three protection functions are provided for protection during self-programming: area protection, the FENTRYR register, and FLWE bit.

An overview of these protection functions is shown in Table 2.14.

Table 2.15 indicates the areas to which the protection can be configured. Configure the area for which rewriting is to be prohibited during self-programming.

Table 2.14 Overview of Protection Functions

Protection Type	Purpose	Overview of Function
Area Protection	Prohibits programming/erasure of the selected blocks in the user area.	Rewriting of the user area other than the selected blocks (access window) is prohibited during self-programming.
FENTRYR Register	Prohibits programming/erasure of the user area and data area by setting the flash memory to read mode.	If the FENTRYR register is “0000h”, the flash memory is in read mode. In read mode, programming/erasure of the flash memory is prohibited.
FLWE bit	Prohibits programming/erasure of an area where the P/E mode is set by the FENTRYR register.	If you set the FLWE bit, programming/erasure and blank check of the flash memory are prohibited.

Table 2.15 Area to Which Protection Can be Configured

Protection Function	Flash Memory Reading/Programming/Erasure Prohibition			
	User Area		Data Area	
	Reading	Programming/Erasure	Reading	Programming/Erasure
Area Protection	—	✓	—	—
FENTRYR Register	—	✓	—	✓
FLWE bit	—	✓	—	✓

✓: Can be configured, —: Cannot be configured

2.4.1.1 Description of the Protection Functions

(1) Area Protection

You can prohibit programming/erasure of the selected blocks in the user area.

Rewriting of the user area other than the selected blocks (access window) is prohibited during self-programming. The access window is set by using the FAW register.

If the FSPR bit is set, the access window never be set again. Once the FSPR bit set to “0”, this bit cannot be restored to “1”. Exercise extra caution when handling the FSPR bit.

Table 2.16 shows the setting of the FAW register.

Table 2.16 FAW Register Setting

Register Name	Flash access window setting register (FAW)			
Bit Name	Flash access window start address bit (FAWS) (Bit: b0-b11)	Access Window protection bit (FSPR) (Bit: b15)	Flash access window end address bit (FAWE) (Bit: b16-b27)	Start-up area select bit (BTFLG) (Bit: b31)
Setting Value	From b13 to b23 of the access window start address	“0” when protection is enabled	From b13 to b23 of the address that follows the access window end address	Don’t care

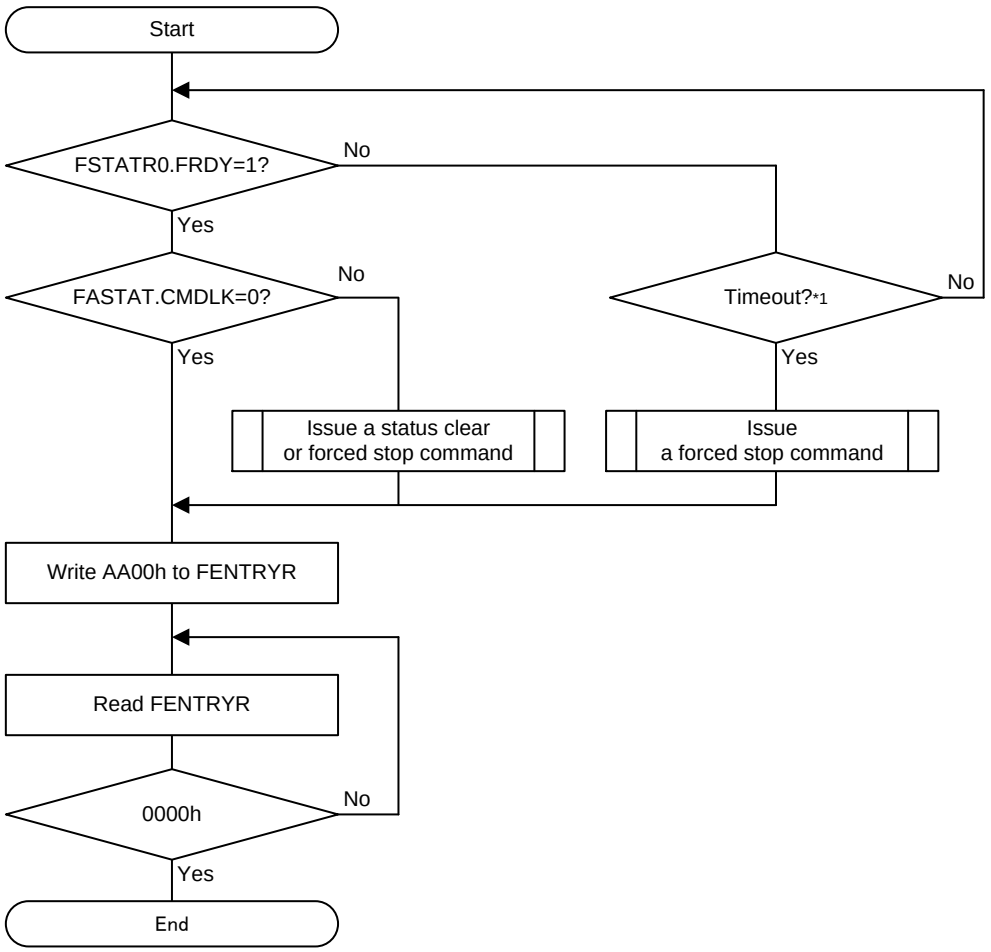
For how to set the FAW register, refer to 2.4.2.2, Option-Setting Memory Setting Example.

(2) FENTRYR Register

You can prohibit programming/erasure of the user area and data area by setting the flash memory to read mode.

When the FENTRYR register is set to “0000h”, the flash memory is in read mode. In read mode, programming/erasure of the flash memory is prohibited.

Figure 2.9 shows how to set the flash memory to read mode.



Note 1. For the timeout time (maximum processing time of the FACI command being executed), refer to the “Electrical Characteristics” of your device.)

Figure 2.9 Flow for Transition to Read Mode

(3) FLWE bit

You can prohibit programming/erasure of an area where the P/E mode is set by the FENTRYR register.

If you set the FLWE bit of the FWEPROR register, programming/erasure and blank check of the flash memory are prohibited by software.

Table 2.17 shows the setting of the FWEPROR register.

Table 2.17 FWEPROR Register Setting

Register Name	Flash P/E protect register (FWEPROR)
Bit Name	Flash P/E bit (FLWE) (Bit: b0-b1)
Setting Value	00h, 10h (default value), or 11h

2.4.2 Protection during Update

Two protection functions are provided for protection during update: the start-up program protection and dual bank function.

An overview of these protection functions is shown in Table 2.18.

Table 2.18 Overview of Protection Functions

Protection Type	Overview of Function
Start-Up Program Protection	When you update the start-up program, this function allows you to update to a new program without erasing the current start-up program. If the update operation is disrupted for any reason including a reset, the program is still safely updated.
Dual Bank Function	You can use the bank mode switching function and selecting the startup bank to update a program in another bank while user programs continue to run. If the update operation is disrupted for any reason including a reset, the program is still safely updated.

For a sample program for the start-up program protection, refer to “RX100/RX200 Series Updating Firmware Using Start-Up Program Protection and Serial Communication (R01AN3740)”.

For a sample program for the dual bank function, refer to “RX Family Firmware Update Module Using Firmware Integration Technology (R01AN5824)”.

2.4.2.1 Description of the Protection Functions

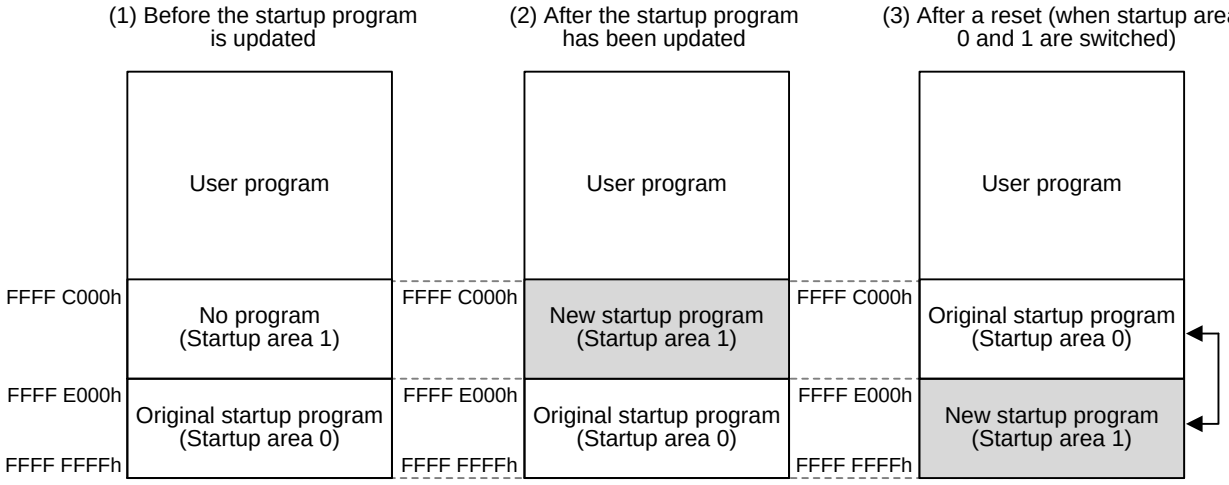
(1) Start-Up Program Protection

When updating the start-up program, an area (where the current start-up program is stored) is swapped with another area (where the new start-up program is stored) so that the start-up program can be updated without erasing the current start-up program. If the update operation is disrupted for any reason including a reset, the program is still safely updated.

The FSPR bit of the FAW register can be used to fix the start-up area selection status. Once the FSPR bit set to “0”, this bit cannot be restored to “1”. Exercise extra caution when handling the FSPR bit.

Note that the start-up program protection cannot be used when dual mode is selected by the bank mode switching function of the dual bank function.

Figure 2.10 gives an overview of the start-up program protection. Figure 2.11 shows the flow for swapping the start-up area.



Note. The addresses are used for reading, programming, and erasure. Startup area 0 is erased when a block erase command is issued for the address FFFF C000h with allocation shown under (3).

(1) Before the startup program is updated
When the FAW.BTFLG bit is 1 and the SAS[1:0] bits are 0xb, the address range from FFFF C000h to FFFF DFFFh is selected as startup area 1 and the address range from FFFF E000h to FFFF FFFFh is selected as startup area 0.

(2) After the startup program has been updated
A new startup program can be written to startup area 1. At this time, startup area 0 will still contain the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.

(3) After a reset (when startup areas 0 and 1 are switched)
When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the FAW.BTFLG bit) with the configuration setting command. Afterwards, the program is started
up from startup area 1 after a reset.

Figure 2.10 Concept of Protection of the Startup Program (for RX72M)

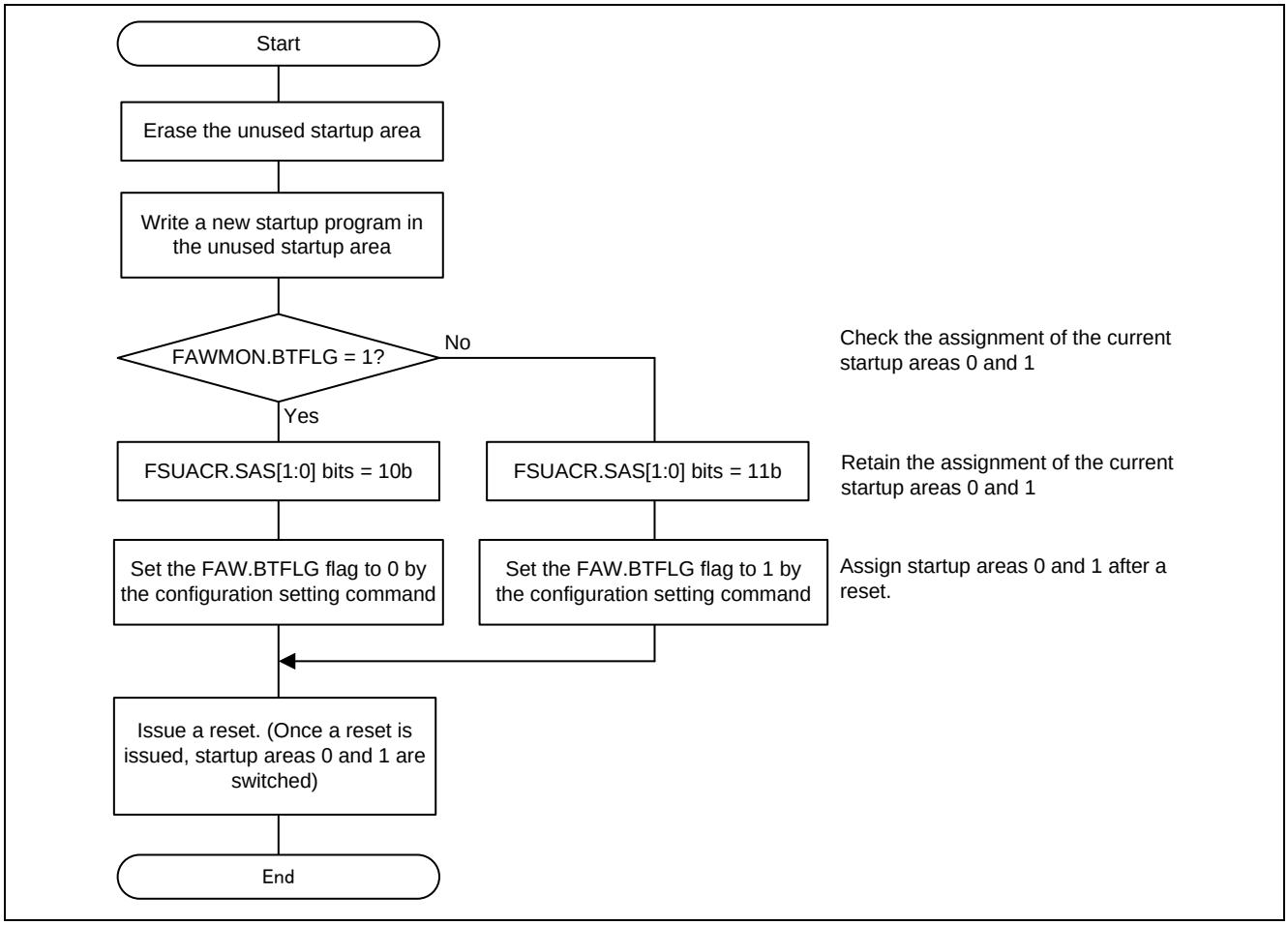


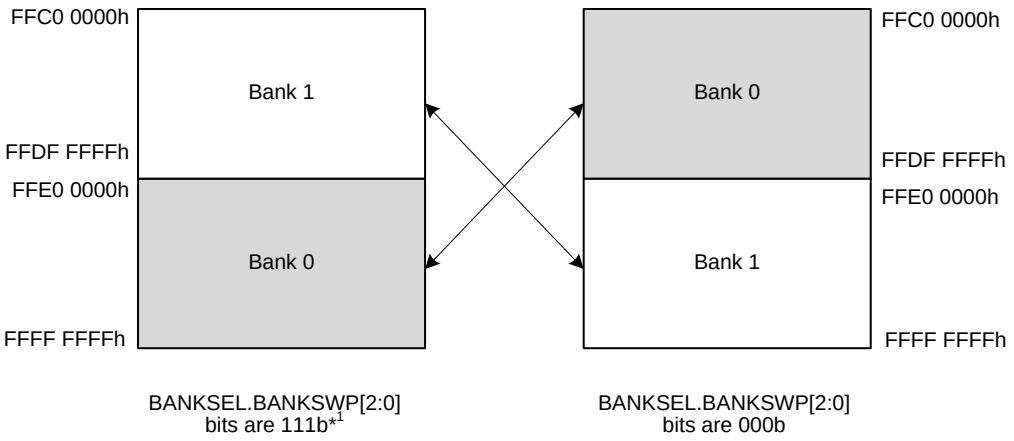
Figure 2.11 Flow for Swapping the Start-Up Area

For how to set the FAW register, refer to 2.4.2.2, Option-Setting Memory Setting Example.

(2) Dual Bank Function

You can use the bank mode switching function and selecting the startup bank to switch the bank address so that a program can be updated in another bank while user programs continue to run. If the update operation is disrupted for any reason including a reset, the program is still safely updated.

Figure 2.12 gives an example start-up bank selection. Figure 2.13 shows the flow for switching the start-up bank.



Note: The addresses of the reset vector are fixed to FFFF FFFCh to FFFF FFFFh.

Note 1. For products without setting of the option-setting memory, the BANKSEL.BANKSWP[2:0] bits are 111b.

Figure 2.12 Example of Start-Up Bank Selection (for RX72M)

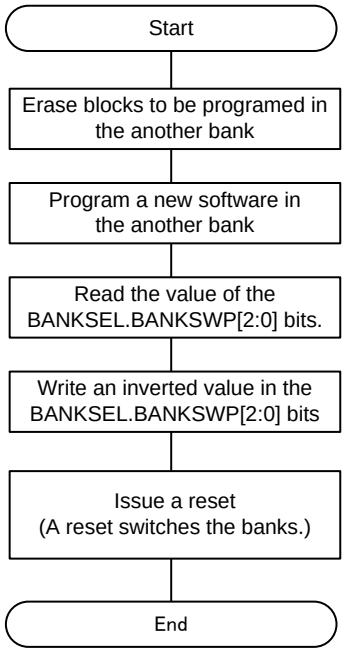


Figure 2.13 Flow for Switching the Start-Up Bank

2.4.2.2 Option-Setting Memory Setting Example

The access window and start-up program protection settings are enabled by writing to 0xFE7F5D64 in the option-setting memory.

Also, for instructions on writing data to the option-setting memory, refer to the User's Manual: Hardware of the device.

Table 2.19 and Table 2.20 gives a setting example for each protection.

Table 2.19 Setting the Access Window

```
/* Setup the Flash access window setting Register */
#pragma address FAW_REG = 0xFE7F5D64
const unsigned long FAW_REG = 0x07FB07F9;
```

In this example,
the access window start address is FFFF 2000h (the FAWS bit is set to 7F9h (b23 to b13 of FFFF 2000h)), the access window end address is set to FFFF 7FFFh (the FAWE bit is set to 7FBh (b23 to b13 of FFFF 7FFFh)), and protection provided by the FSPR bit is enabled.

Table 2.20 Setting the Start-Up Program Protection

```
/* SetUp the Flash access window setting Register */
#pragma address FAW_REG = 0xFE7F5D64
const unsigned long FAW_REG = 0x80000000;
```

In this example, the start-up area is set to use the addresses from FFFF E000h to FFFF FFFFh.

3. Reference Documents

- RX110 Group User's Manual: Hardware (R01UH0421)
- RX111 Group User's Manual: Hardware (R01UH0365)
- RX113 Group User's Manual: Hardware (R01UH0448)
- RX130 Group User's Manual: Hardware (R01UH0560)
- RX13T Group User's Manual: Hardware (R01UH0822)
- RX140 Group User's Manual: Hardware (R01UH0905)
- RX210 Group User's Manual: Hardware (R01UH0037)
- RX21A Group User's Manual: Hardware (R01UH0251)
- RX220 Group User's Manual: Hardware (R01UH0292)
- RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496)
- RX23E-A Group User's Manual: Hardware (R01UH0801)
- RX23T Group User's Manual: Hardware (R01UH0520)
- RX23W Group User's Manual: Hardware (R01UH0823)
- RX24T Group User's Manual: Hardware (R01UH0576)
- RX24U Group User's Manual: Hardware (R01UH0658)
- RX610 Group User's Manual: Hardware (R01UH0032)
- RX62N Group, RX621 Group User's Manual: Hardware (R01UH0033)
- RX62T Group, RX62G Group User's Manual: Hardware (R01UH0034)
- RX630 Group User's Manual: Hardware (R01UH0040)
- RX634 Group User's Manual: Hardware (R01UH0495)
- RX63N Group, RX631 Group User's Manual: Hardware (R01UH0041)
- RX63T Group User's Manual: Hardware (R01UH0238)
- RX64M Group User's Manual: Hardware (R01UH0377)
- RX65N Group, RX651 Group User's Manual: Hardware (R01UH0590)
- RX66N Group User's Manual: Hardware (R01UH0825)
- RX66T Group User's Manual: Hardware (R01UH0749)
- RX671 Group User's Manual: Hardware (R01UH0899)
- RX71M Group User's Manual: Hardware (R01UH0493)
- RX72T Group User's Manual: Hardware (R01UH0803)
- RX72M Group User's Manual: Hardware (R01UH0804)
- RX72N Group User's Manual: Hardware (R01UH0824)
  (Get the latest version from the Renesas Electronics website.)

- Technical Update/Technical News
  (Get the latest information from the Renesas Electronics website.)

- C compiler manual
- C/C++ Compiler Package for RX Family
  (Get the latest version from the Renesas Electronics website.)

- RX100/RX200 Series Updating Firmware Using Start-Up Program Protection and Serial Communication (R01AN3740)
- RX Family Firmware Update Module Using Firmware Integration Technology (R01AN5824)

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Mar.28.12	—	First edition issued
2.00	Sep.30.16	All	Added the following target devices: • RX110 Group • RX111 Group • RX113 Group • RX130 Group • RX210 Group • RX21A Group • RX220 Group • RX231, RX230 Group • RX23T Group • RX24T Group • RX62G Group • RX62T Group • RX634 Group • RX63T Group • RX64M Group • RX71M Group Added “1. Device Categories”. Changed the arrangement of the sections according to the target device categories.
3.00	Jun.01.17	All	Added the following target devices: • RX65N, RX651 Group
4.00	May.13.19	All	Added the following target devices: • RX24U, RX66T, RX72T Group Corrected RX21A Group and RX220 Group from device group A to device group B in Table 1. Corrected Table 4, Table 17 and Table 30. Add descriptions of the access window.
5.00	Nov.07.19	All	Added the following target devices: • RX23E-A Group • RX23W Group • RX13T Group • RX72M Group • RX72N Group • RX66N Group

Rev.	Date	Description	
		Page	Summary
6.00	Sep.10.21	All	Changed Target Device to RX Family. Added RX140 Group and RX671 Group to "Device" in "1. Device Categories". Added On-Chip Debugger Connection Enable/Disable and Access Window Protection Command to "Protection Function" in "1. Device Categories". Added "8. Device Group G Protection Methods". Changed the description of the specifications of the device group that supports the access window.
7.00	Nov.11.21	All	Changed the title to "How to Prevent the On-Chip Flash Memory from Being Accessed by Third Parties or Being Accidentally Programmed by Developers". Reordered the device names in "1. Device Categories". Changed the Note 3 in "1. Device Categories". Corrected RX23W Group from device group A to device group B. Added "2. Protection during Self-Programming Initiated By Developers". Changed the arrangement of the sections to add "Protection during Self-Programming Initiated By Developers". Changed reference to a figure or a table from "below" to the Figure or Table number.

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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

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/.