

R7F0C 系列

R01AN1948CC0101

Flash Self-Programming Library S50 应用说明

Rev.1.01
2014.01.15

要点

FSL S50 自编程库是专门针对 R7F0C 系列芯片设计的，通过减少库函数和优化程序结构来压缩自编程库的尺寸，从而减小对用户 RAM 空间的占用。

该应用说明将从资源和 API 函数两方面介绍该自编程库，并通过提供样例程序，为用户展示如何使用该库对 Code Flash 进行操作。

目标器件

R7F0C008/009

目录

1.	概要.....	2
1.1	资源占用.....	2
1.2	资源配置示意图.....	2
2.	库文件及标准接口函数.....	3
2.1	库文件.....	3
2.2	标准接口函数.....	3
3.	自编程库的使用.....	8
3.1	自编程库的工作流程.....	8
3.2	修改 RAM 资源分配.....	9
3.3	程序样例.....	11
4.	注意事项.....	14
	公司主页和咨询窗口.....	15

1. 概要

该自编程库提供了非常易用的接口来操作 code flash 区。目前支持的接口函数包括：erasing, writing, blank check, internal verify, 通过简化接口函数，极大的节省了资源。

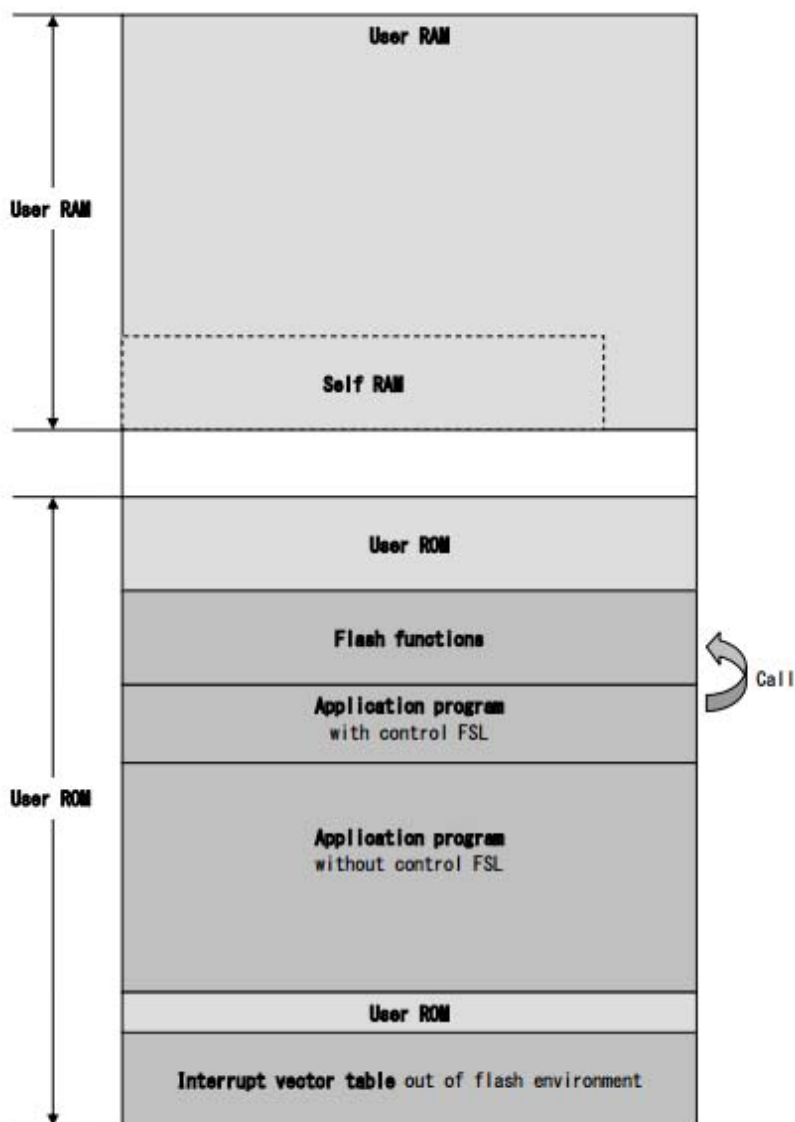
1.1 资源占用

该自编程库使用的资源情况如下表所示。

Resource	Target
使用的 RAM 资源	<= 640 bytes (0xFF900 - 0xFFB7F)
使用的 ROM 资源	<= 470 bytes
使用的堆栈资源	<= 60 bytes

NOTE: RAM 空间 0xFFE20 - 0xFFEDF 不能放置 data buffer，也不能将堆栈区放在该位置范围内。

1.2 资源配置示意图



2. 库文件及标准接口函数

2.1 库文件

2.1.1 fsl.h / fsl.inc

文件对库函数做了声明，同时对错误状态常量做了定义。

2.1.2 fsl_types.h

文件定义了 FSL 库用到的类型和结构体，本文件只用于 C 语言编程环境下。

2.1.3 fsl.lib

本文件是库文件，里面包含了库函数的函数体。

2.2 标准接口函数

2.2.1 初始化函数(FSL_init)

(1) 函数原型

```
fsl_u08 FSL_Init( __far fsl_descriptor_t* descriptor_pstr)
```

(2) 参数

descriptor_pstr: 指向 FSL_Init 函数的设置结构体，该结构体在 fsl_types.h 中定义如下：

C 语言	汇编语言
<pre>/* FSL descriptor type */ typedef struct { fsl_u08 fsl_flash_voltage_u08; fsl_u08 fsl_frequency_u08; fsl_u08 fsl_auto_status_check_u08; } fsl_descriptor_t;</pre>	<pre>fsl_descriptor_str: DB fsl_flash_voltage_u08 DB fsl_frequency_u08 DB fsl_auto_status_check_u08</pre>

fsl_flash_voltage_u0: 设置 flash 电压模式，电压模式选择详情请查看硬件手册。

00: 全速模式

其它: 宽电压模式

fsl_frequency_u08: 自编程时使用的 CPU 时钟频率，例如 fsl_frequency_u08 = 0x14，即 20MHz。

NOTE:

- 如果用户使用小于 4MHz 的时钟频率，可以设置 fsl_frequency_u08 为 1MHz，2MHz，3MHz 整数频率，但是不能使用非整数的时钟频率，如 1.5MHz。
- 如果设置的时钟大于 4MHz，那么可以使用任意频率，但是仍应该将 fsl_frequency_u08 设为整数。（例如用户使用了 4.5MHz 的工作时钟，那么 fsl_frequency_u08 应设为 5MHz）

fsl_auto_status_check_u08: 状态检查设置（本库中未使用）。虽然未使用该参数，但是为了与 FSLRL78 Type01 兼容，建议设置成 01H。

(3) 返回值

Type	Value	Symbol	Description
fsl_u08	0x00	FSL_OK	OK

(4) 错误状态

无

(5) 功能

该函数初始化 FSL 库，同时初始化工作内存空间。其它库函数必须在该函数被执行后才能被调用。

(6) 工作条件

片上调整晶振（H0C0）处于工作状态。

2.2.2 开始函数(FSL_Open)

(1) 函数原型

```
void FSL_Open( void )
```

(2) 参数

无

(3) 返回值

无

(4) 错误状态

无

(5) 功能

打开 flash 环境。

(6) 工作条件

必须在 FSL_Init 后被调用。

2.2.3 准备函数(FSL_PrepareFunctions)

(1) 函数原型

```
void FSL_PrepareFunctions( void )
```

(2) 参数

无

(3) 返回值

无

(4) 错误状态

无

(5) 功能

在该函数之后，FSL_BlankCheck, FSL_Erase, FSL_Write, FSL_IVerify 这些函数才能被调用。

(6) 工作条件

必须在 FSL_Init 后被调用。

2.2.4 块检查函数 (FSL_BlankCheck)

(1) 函数原型

```
fsl_u08 FSL_BlankCheck( fsl_u16 block_u16 )
```

(2) 参数

block_u16: 块号，使用存在块号。

NOTE: 就算用户在使用时指定了错误的参数，也不会返回参数错误，所以请保证参数正确性。

(3) 返回值

Type	Value	Symbol	Description
fsl_u08	0x1B	FSL_ERR_BLANKCHECK	Blank error
	0x00	FSL_OK	OK

(4) 错误状态

如果指定的块没有擦除，返回 FSL_ERR_BLANKCHECK.

(5) 功能

该函数用来检查指定的块是否为空。

(6) 工作条件

必须在 FSL_Init, FSL_Open, FSL_PrepareFunctions 后被调用。

2.2.5 块擦除函数 (FSL_Erase)

(1) 函数原型

```
fsl_u08 FSL_Erase( fsl_u16 block_u16 )
```

(2) 参数

block_u16: 要擦除块的块号，使用存在块号。

NOTE: 就算用户在使用时指定了错误的参数，也不会返回参数错误，所以请保证参数正确性。

(3) 返回值

Type	Value	Symbol	Description
fsl_u08	0x10	FSL_ERR_PROTECTION	Protection error
	0x00	FSL_OK	OK
	0x1A	FSL_ERR_ERASE	Erase error

(4) 错误状态

如果指定的块被保护，返回 FSL_ERR_PROTECTION.

(5) 功能

该函数用来擦除指定的块。

(6) 工作条件

必须在 FSL_Init, FSL_Open, FSL_PrepareFunctions 后被调用

2.2.6 写函数(FSL_Write)

(1) 函数原型

```
fsl_u08 FSL_Write( __near fsl_write_t* write_pstr )
```

(2) 参数

descriptor_pstr: 指向 FSL_Write 函数的设置结构体, 该结构体在 fsl_types.h 中定义如下:

C 语言	汇编语言
<pre>/* FSL descriptor type */ typedef struct { fsl_u08 __near sl_data_buffer_p_u08; fsl_u32 fsl_destination_address_u32; fsl_u08 fsl_word_count_u08; } fsl_write_t;</pre>	<pre>fsl_descriptor_str: DB fsl_flash_voltage_u08 DB fsl_frequency_u08 DB fsl_auto_status_check_u08</pre>

fsl_data_buffer_p_u08: Data buffer 指针地址。

fsl_destination_address_u3: 写数据的开始地址。

fsl_word_count_u08: 写数据的长度, 1 word = 4 bytes。

NOTE:

- a. 就算用户在使用时指定了错误的参数, 也不会返回参数错误, 所以请保证参数的正确性。
- b. 各参数请保证满足如下条件:
 - fsl_destination_address_u32 mod 4 = 0
 - fsl_destination_address_u32 <= 0xFFFFF
 - fsl_destination_address_u32 + fsl_word_count_u08(convert to byte) <= 0xFFFFFFFF
 - fsl_destination_address_u32 + fsl_word_count_u08(convert to byte) <= max. address
 - fsl_word_count_u08 > 0
 - fsl_word_count_u08 <= 64

(3) 返回值

Type	Value	Symbol	Description
fsl_u08	0x00	FSL_OK	OK
	0x10	FSL_ERR_PROTECTION	Protection error
	0x1C	FSL_ERR_WRITE	Write error

(4) 错误状态

如果指定的块被保护, 返回 Protection error.

如果写失败, 返回 Write error.

(5) 功能

向指定地址写指定长度的数据。

(6) 工作条件

必须在 FSL_Init, FSL_Open, FSL_PrepareFunctions 后被调用。

2.2.7 块校验函数 (FSL_IVerify)

(1) 函数原型

```
fsl_u08 FSL_IVerify( fsl_u16 block_u16 )
```

(2) 参数

block_u16: 要校验块的块号, 请使用存在的块号。

NOTE: 就算用户在使用时指定了错误的参数, 也不会返回参数错误, 所以请保证参数的正确性。

(3) 返回值

Type	Value	Symbol	Description
fsl_u08	0x1B	FSL_ERR_IVERIFY	IVerify error
	0x00	FSL_OK	OK

(4) 错误状态

如果指定的块的内部校验出现错误, 返回 FSL_ERR_IVERIFY.

(5) 功能

该函数用来校验 (内部校验) 指定的块。

(6) 工作条件

必须在 FSL_Init, FSL_Open, FSL_PrepareFunctions 后被调用

2.2.8 关闭环境函数 (FSL_Close)

(1) 函数原型

```
void FSL_Close( void )
```

(2) 参数

无

(3) 返回值

无

(4) 错误状态

无

(5) 功能

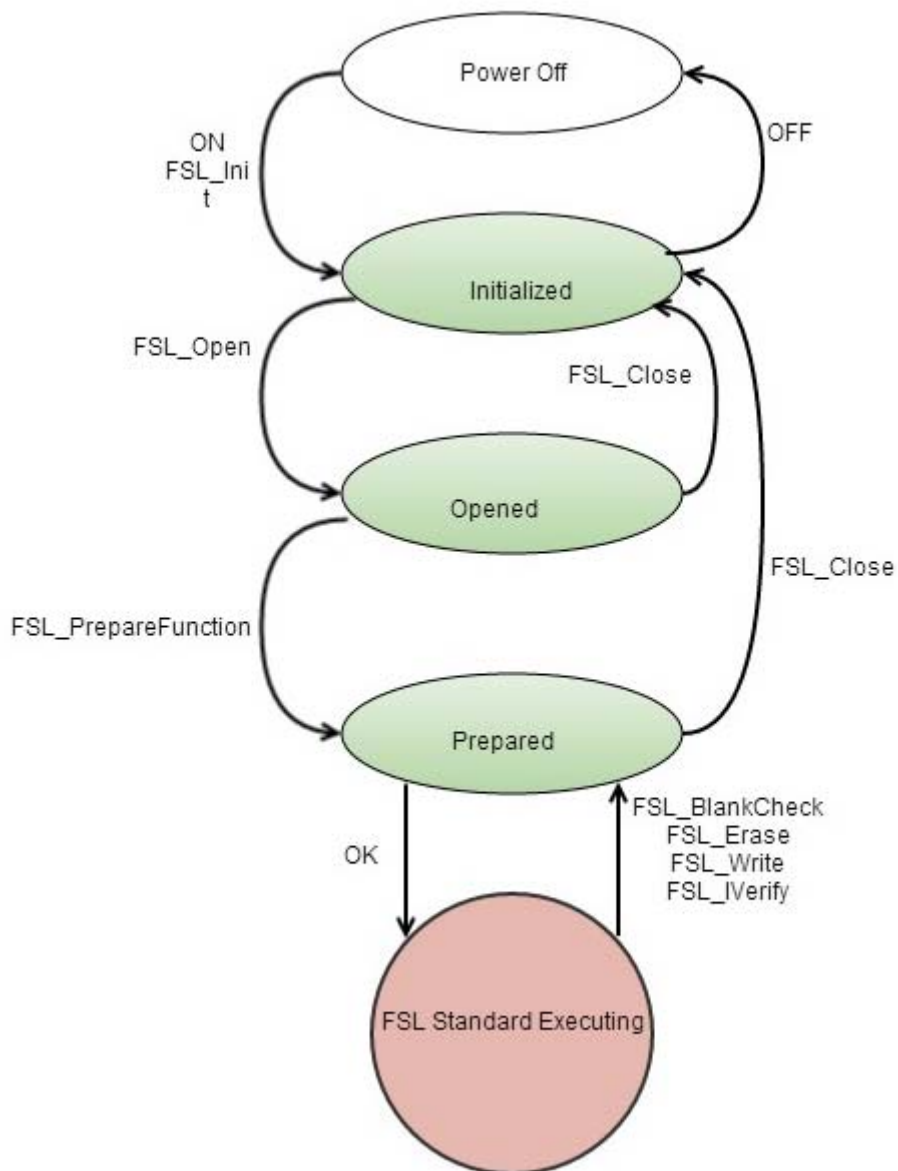
该函数用来关闭自编程环境。

(6) 工作条件

必须在 FSL_Init 后被调用。

3. 自编程库的使用

3.1 自编程库的工作流程

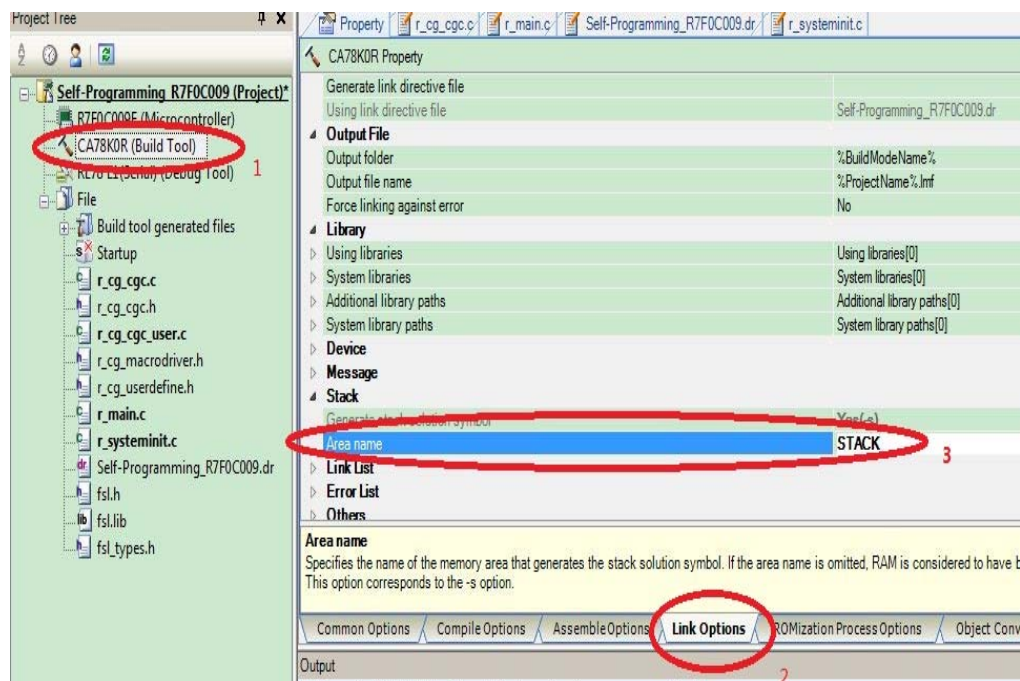


3.2 修改 RAM 资源分配

由于自编程库运行时需要占用 RAM 空间，根据手册，在自编程库运行时 FF900H to FFB7FH 的地址是禁止用户使用的。所以，在使用自编程库时应该手动分配用户 RAM 空间。

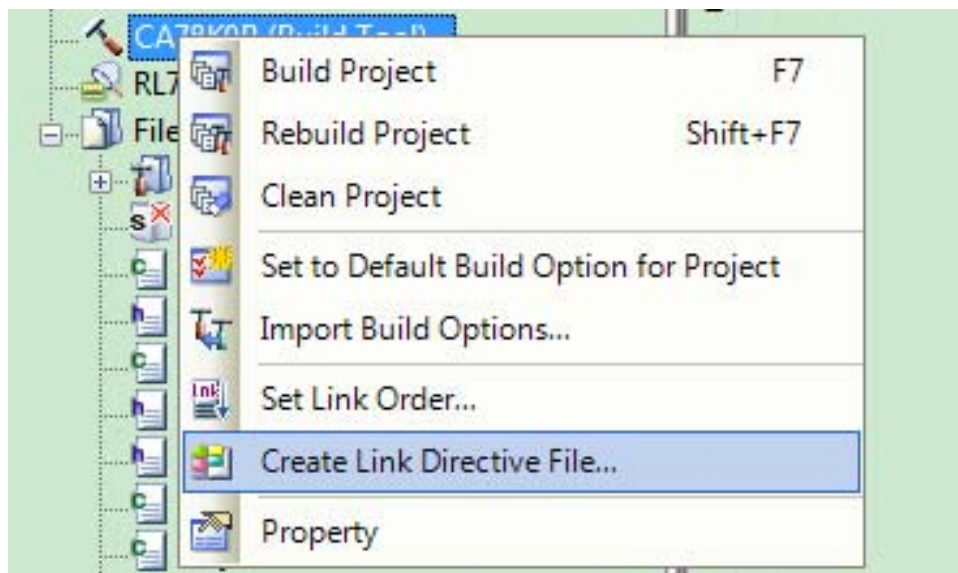
以下流程演示如何修改 RAM 资源分配：

1. 指定堆栈空间。需要首先添加 Stack Area name，方法是单击 CA78K0R，选择 Link Options，在 Stack 选项中，将 Area name 指定为 STACK。如下图所示：



如果用户不指定堆栈区域的话，堆栈会从 RAM 的最大地址向下放置。

2. 右击 CA78K0R (Build Tool) – Create Link Directive File



3. 将 RAM start address 修改为 0xFFB80 (0xFF900 + 0x280)，其中 0x280 为自编程库运行时占用的 640 字节 RAM 空间。同时将堆栈 STACK 的设置 0xFFDBC-0xFFE20，此处设为 64 字节。从 0xFFE20 开始的地址设为 SADDR。设置完成后点击 Generate，这样，就将自编程库用到的 RAM 资源与用户资源分开了。

Memory / Name	Start Address	End Address
REGULAR	0x00000000	0x000fffff
ROM	0x00000000	0x00003fff
RAM	0x000ffb80	0x000fdbba
STACK	0x000fdbbc	0x000ffe1f
SADDR	0x000ffe20	0x000fffff

同时，在工程中会生成一个以工程名字命名的.dr 文件，内容如下：

;MEMORY

MEMORY ROM : (00000000H, 00004000H) / **REGULAR**

MEMORY RAM : (000ffb80H, 0000023bH) / **REGULAR**

MEMORY STACK : (000fdbbcH, 00000064H) / **REGULAR**

MEMORY SADDR : (000ffe20H, 000001e0H) / **REGULAR**

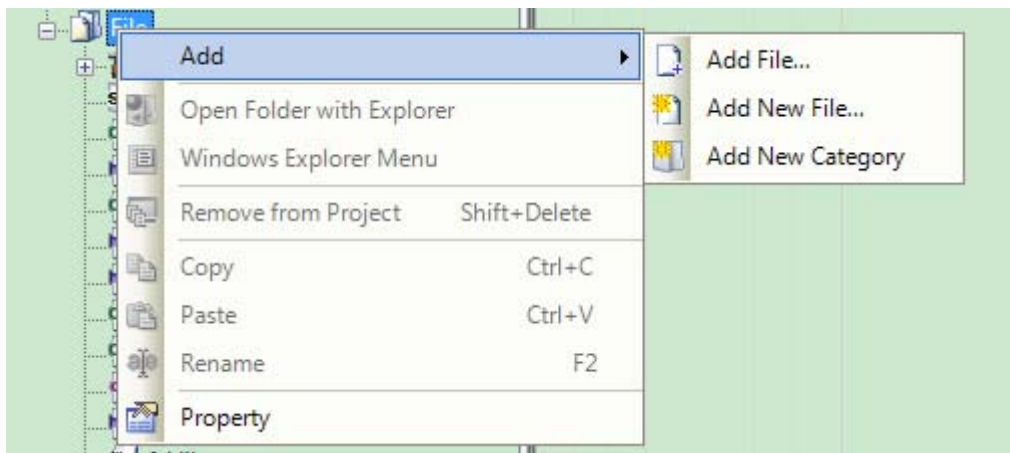
除了通过上述图形化界面来修改 Memory 的资源占用，也可以通过直接修改.dr 文件的方式来完成内存资源分配。需要用到内存指令。内存指令是定义一个内存区间（内存的地址和名字）的指令。定义的内存区间可以被段定位指令通过使用该名字（内存区间的名字）来引用。最多可以定义 100 个内存区间（包含默认定义的内存区间）。以下是语法：

```
MEMORY  memory-area-name  : (  start-address  ,  size  ) [ /  memory-space-
name ]
```

例子见上面的.dr 文件内容。

3.3 程序样例

1. 新建工程，并向工程中导入库文件。右击 file-Add-Add File 将 fsl.h, fsl.lib, fsl_types.h 加入。



2. 将调用相关自编程库函数时用到的头文件包含到源文件中。

```
#include "fsl.h"
#include "fsl_types.h"
```

3. 在调用自编程库函数时，用户不能使用任何中断，所以在调用库函数前使用 DI() 函数。

```
DI();
```

4. 调用 SelfInitiallize() 函数。该函数调用 FSL_init() 函数并判断返回值，如果返回值为 FSL_OK，则调用 FSL_Open() 和 FSL_PrepareFunctions()。

SelfInitiallize() 函数：

```
uint8_t SelfInitiallize(void) {
    __far fsl_descriptor_t InitArg;

    uint8_t          ret1;

    /* ---- Set argument of FSL_Init() ---- */
    InitArg.fsl_flash_voltage_u08 = 0x00;    //模式设置
    InitArg.fsl_frequency_u08     = 0x18;    //频率设置
    InitArg.fsl_auto_status_check_u08 = 0x01; //未使用
    ret1                          = FSL_Init( &InitArg ); /* Flash self library
initialize */

    if(ret1 == FSL_OK) {
        FSL_Open();          /* Flash self programming start */
        FSL_PrepareFunctions();
    }
    return ret1;
}
```

5. 如果 SelfInitiallize() 函数返回值为 FSL_OK, 则调用 uint8_t SelfWriteExecute(uint32_t) 函数, 参数为写操作的目的地址。

SelfWriteExecute(uint32_t) 函数:

```
uint8_t SelfWriteExecute(uint32_t WriteAddr) {

    fsl_write_t WriteArg;
    uint8_t ret;
    uint16_t WriteBlock;

    WriteBlock = ( uint16_t )( WriteAddr / 0x400 );           //计算块号
    ret = FSL_BlankCheck( WriteBlock );                      //对块进行检查
    /* ---- Check blank check result ---- */
    if ( ret == FSL_ERR_BLANKCHECK ) {
        ret = FSL_Erase( WriteBlock );                      //如果块不为空, 则擦除该块
    }
    else {
        /* Do something */
    }

    if(ret == FSL_OK) {
        WriteArg.fsl_data_buffer_p_u08 = &buffer[0];       //要写入的数据
        WriteArg.fsl_destination_address_u32 = WriteAddr;   //要写入的地址
        WriteArg.fsl_word_count_u08 = 0x01;                 //要写入的字数
        ret = FSL_Write( &WriteArg );
    }

    if(ret == FSL_OK) {
        ret = FSL_IVerify(WriteBlock);                      //写入成功, 对块进行校验
    }
    else {
        /* Do something */
    }
    return ret;
}
```

6. 连接仿真器，编译后下载，并仿真，写入结果如下。

设置 WriteArg.fsl_word_count_u08 = 0x01 时的写入结果:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	ASCII
02f70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02f80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02f90	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fa0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fb0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fc0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fd0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fe0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02ff0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03000	86	86	86	86	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030a0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030b0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030c0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030d0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030e0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030f0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???

设置 WriteArg.fsl_word_count_u08 = 0x04;时的写入结果:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	ASCII
02f70	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02f80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02f90	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fa0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fb0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fc0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fd0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02fe0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
02ff0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03000	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86	???
03010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030a0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030b0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030c0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030d0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030e0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
030f0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???
03160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	???

4. 注意事项

NOTE1: 本库不支持参数检查和流程控制检查, 使用时请确保正确的参数和正确的控制流程。

NOTE2: 自编程期间, 不能处理中断服务。

NOTE3: 自编程期间, 看门狗不会关闭。

NOTE4: 在使用自编程库期间, 用户不要擦或修改 RAM。

NOTE5: 在访问 Data Flash 期间不能访问 Code Flash。

NOTE6: 不要将 databuffer (write 函数的参数) 及栈放到 0xFFE20 以后的地址。

NOTE7: 如果在自编程期间使用了 DTC 功能, 不要将 DTC 功能使用的 RAM 资源放到自编程使用的 RAM 区, 同时 0xFFE20 以后的区域也不要使用。

公司主页和咨询窗口

瑞萨电子主页

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

咨询

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

修订记录	R7F0C 系列 应用说明
------	---------------

Rev	发行日	修订内容	
		页	修订处
1.00	2013.11.15	—	初版发行
1.01	2014.01.15	—	文档编号更新

所有商标及注册商标分别归属于其所有者。

产品使用时的注意事项

本文对适用于单片机所有产品的“使用时的注意事项”进行说明。有关个别的使用时的注意事项请参照正文。此外，如果在记载上有与本手册的正文有差异之处，请以正文为准。

1. 未使用的引脚的处理

【注意】将未使用的引脚按照正文的“未使用引脚的处理”进行处理。

CMOS产品的输入引脚的阻抗一般为高阻抗。如果在开路的状态下运行未使用的引脚，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，有可能被误认为是输入信号而引起误动作。

未使用的引脚，请按照正文的“未使用引脚的处理”中的指示进行处理。

2. 通电时的处理

【注意】通电时产品处于不定状态。

通电时，LSI内部电路处于不确定状态，寄存器的设定和各引脚的状态不定。通过外部复位引脚对产品进行复位时，从通电到复位有效之前的期间，不能保证引脚的状态。

同样，使用内部上电复位功能对产品进行复位时，从通电到达到复位产生的一定电压的期间，不能保证引脚的状态。

3. 禁止存取保留地址（保留区）

【注意】禁止存取保留地址（保留区）

在地址区域中，有被分配将来用作功能扩展的保留地址（保留区）。因为无法保证存取这些地址时的运行，所以不能对保留地址（保留区）进行存取。

4. 关于时钟

【注意】复位时，请在时钟稳定后解除复位。

在程序运行中切换时钟时，请在要切换成的时钟稳定之后进行。复位时，在通过使用外部振荡器（或者外部振荡电路）的时钟开始运行的系统中，必须在时钟充分稳定后解除复位。另外，在程序运行中，切换成使用外部振荡器（或者外部振荡电路）的时钟时，在要切换成的时钟充分稳定后再进行切换。

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

即使是同一个群的单片机，如果产品型号不同，由于内部ROM、版本模式等不同，在电特性范围内有时特性值、动作容限、噪声耐量、噪声辐射量等不同。因此，在变更不认同型号的产品时，请对每一个型号的产品进行系统评价测试。

Notice

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

以下"注意事项"为从英语原稿翻译的中文译文，仅作参考译文，英文版的"Notice"具有正式效力。

注意事项

1. 本文件中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文件中的电路、软件 and 相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，瑞萨电子不承担任何责任。
 2. 在准备本文件所记载的信息的过程中，瑞萨电子已尽量做到合理注意，但是，瑞萨电子并不保证这些信息都是准确无误的。用户因本文件中所记载的信息的错误或遗漏而遭受的任何损失，瑞萨电子不承担任何责任。
 3. 对于因使用本文件中的瑞萨电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，瑞萨电子不承担任何责任。本文件所记载的内容不应视为对瑞萨电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
 4. 用户不得更改、修改、复制或其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或其他方式非法使用瑞萨电子产品的行为而遭受的任何损失，瑞萨电子不承担任何责任。
 5. 瑞萨电子产品根据其质量等级分为两个等级：“标准等级”和“高质量等级”。每种瑞萨电子产品的推荐用途均取决于产品的质量等级，如下所示：
标准等级： 计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。
高质量等级： 运输设备（汽车、火车、轮船等）、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。
瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统及可能造成人身伤害的产品或系统（人工生命维持装置或系统、植埋于体内的装置等）中，或者可能造成重大财产损失的产品或系统（核反应堆控制系统、军用设备等）中。在将每种瑞萨电子产品用于某种特定应用之前，用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失，瑞萨电子不承担任何责任。
 6. 使用本文件中记载的瑞萨电子产品时，应在瑞萨电子指定的范围内，特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失，瑞萨电子不承担任何责任。
 7. 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，瑞萨电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。由于难于对微机电软件单独进行评估，所以请用户自行对最终产品或系统进行安全评估。
 8. 关于环境保护方面的详细内容，例如每种瑞萨电子产品的环境兼容性等，请与瑞萨电子的营业部门联系。使用瑞萨电子产品时，请遵守对管制物质的使用或含量进行管理的所有相关法律法规（包括但不限于《欧盟RoHS指令》）。对于因用户未遵守相应法律法规而导致的损害或损失，瑞萨电子不承担任何责任。
 9. 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可将本文件中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的目的（如大规模杀伤性武器的开发等）。在将本文件中记载的瑞萨电子产品或技术进行出口时，应当遵守相应的出口管制法律法规，并按照上述法律法规所规定的程序进行。
 10. 向第三方分销或处分产品或者以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商，有责任事先向上述第三方通知本文件规定的内容和条件；对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失，瑞萨电子不承担任何责任。
 11. 在事先未得到瑞萨电子书面认可的情况下，不得以任何形式部分或全部转载或复制本文件。
 12. 如果对本文件所记载的信息或瑞萨电子产品有任何疑问，或者用户有任何其他疑问，请向瑞萨电子的营业部门咨询。
- (注1) 瑞萨电子：在本文件中指瑞萨电子株式会社及其控股子公司。
(注2) 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

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

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

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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

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

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhichunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0889, Fax: +86-21-2226-0899

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-0888, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9800, Fax: +886-2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #09-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jin Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141