
SH7266/SH7267 グループ

E10A-USB フラッシュメモリダウンロード機能の応用例 (シリアルフラッシュメモリへのダウンロード)

R01AN1151JJ0100
Rev. 1.00
2012.05.18

要旨

E10A-USB エミュレータにはフラッシュメモリにロードモジュールをダウンロードする機能があります。本機能は、フラッシュメモリにアクセスするために、ダウンロードプログラム（以下、「FMTOOL」と称します）を使用します。

本アプリケーションノートでは、この FMTOOL を応用したシリアルフラッシュメモリへのダウンロード方法について説明します。

対象デバイス

SH7266/SH7267 グループ

以下、総称して「SH7267」として説明します。

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様	3
2. 動作確認条件	4
3. 関連アプリケーションノート	4
4. ハードウェア説明	5
4.1 使用端子一覧	5
4.2 参考回路	6
5. ソフトウェア説明	7
5.1 動作概要	7
5.1.1 バッチファイル	7
5.1.2 イレーズモジュール	7
5.1.3 ライトモジュール	8
5.2 ファイル構成	10
5.3 定数一覧	10
5.4 構造体/共用体一覧	11
5.5 変数一覧	11
5.6 関数一覧	12
5.7 関数仕様	13
5.8 フローチャート	17
5.8.1 イレーズモジュール	17
5.8.2 ライトモジュール	17
5.8.3 FMTOOLの初期化	18
5.8.4 フラッシュメモリ書き込み処理	19
5.9 注意事項	20
5.9.1 ロードモジュールへのダミーデータ追加	20
5.9.2 ロードモジュール間のセクタ共有禁止	21
5.9.3 アドレス変換テーブル内でのセクタ共有禁止	22
6. 応用例	23
6.1 ユーザプログラムダウンロード手順	23
6.1.1 ダウンロード環境の準備	23
6.1.2 バッチファイルの登録	23
6.1.3 Configurationダイアログボックスの設定	24
6.1.4 ダウンロードモジュールの追加	25
6.1.5 ユーザプログラムのダウンロード	25
6.2 シリアルフラッシュブートへの応用	26
6.2.1 ダウンローダをFMTOOLに置き換える際の変更点	26
6.2.2 定数データの保存と読み出し	27
6.3 FMTOOLのカスタマイズ	28
6.3.1 サンプルコードに対応するデバイス仕様	28
6.3.2 カスタマイズの内容	28
7. サンプルコード	29
8. 参考ドキュメント	29

1. 仕様

ロードモジュールをシリアルフラッシュメモリにダウンロードします。ダウンロードには、シリアルフラッシュメモリに対応した FMTOOL を使用します。FMTOOL は、ルネサスシリアルペリフェラルインタフェース (RSPI) を使用して、シリアルフラッシュメモリをアクセスします。

表 1.1に使用する周辺機能と用途を、図 1.1にFMTOOLを使用したダウンロード処理の流れを示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
ルネサスシリアルペリフェラルインタフェース (RSPI) チャンネル 0	シリアルフラッシュメモリへのダウンロード
H-UDI	E10A-USB エミュレータの接続

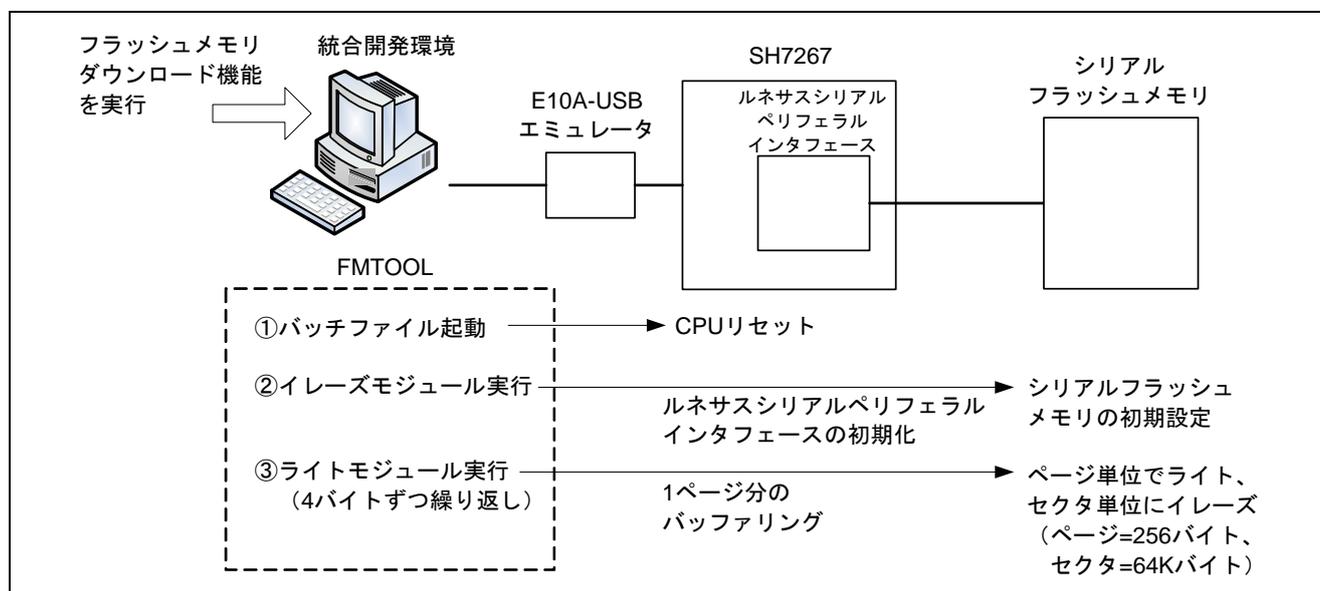


図 1.1 FMTOOL を使用したダウンロード処理の流れ

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	SH7267
使用デバイス	シリアルフラッシュメモリ メーカー : Silicon Storage Technology 社 型名 : SST25VF016B
動作周波数	CPU クロック (I ϕ) : 144MHz バスクロック (B ϕ) : 72MHz 周辺クロック (P ϕ) : 36MHz
動作電圧	電源電圧 (I/O) : 3.3V 電源電圧 (内部) : 1.25V
統合開発環境	ルネサス エレクトロニクス製 High-performance Embedded Workshop Ver.4.07.00
C コンパイラ	ルネサス エレクトロニクス製 SuperH RISC engineファミリ C/C++コンパイラパッケージ Ver.9.03 Release02 コンパイラオプション -cpu=sh2afpu -fpu=single -include="\$(WORKSPDIR)¥inc" -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo
使用ボード	R0K572670C000BR

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

SH7266/SH7267 グループ シリアルフラッシュメモリからのブート例 (R01AN0214JJ)

SH7262/SH7264 グループ ルネサスシリアルペリフェラルインタフェース シリアルフラッシュメモリ接続例 (RJJ06B1001)

E10A-USB エミュレータ用 フラッシュメモリ ダウンロードプログラム アプリケーションノート (R01AN0957JJ)

4.2 参考回路

図 4.1に接続例を示します。

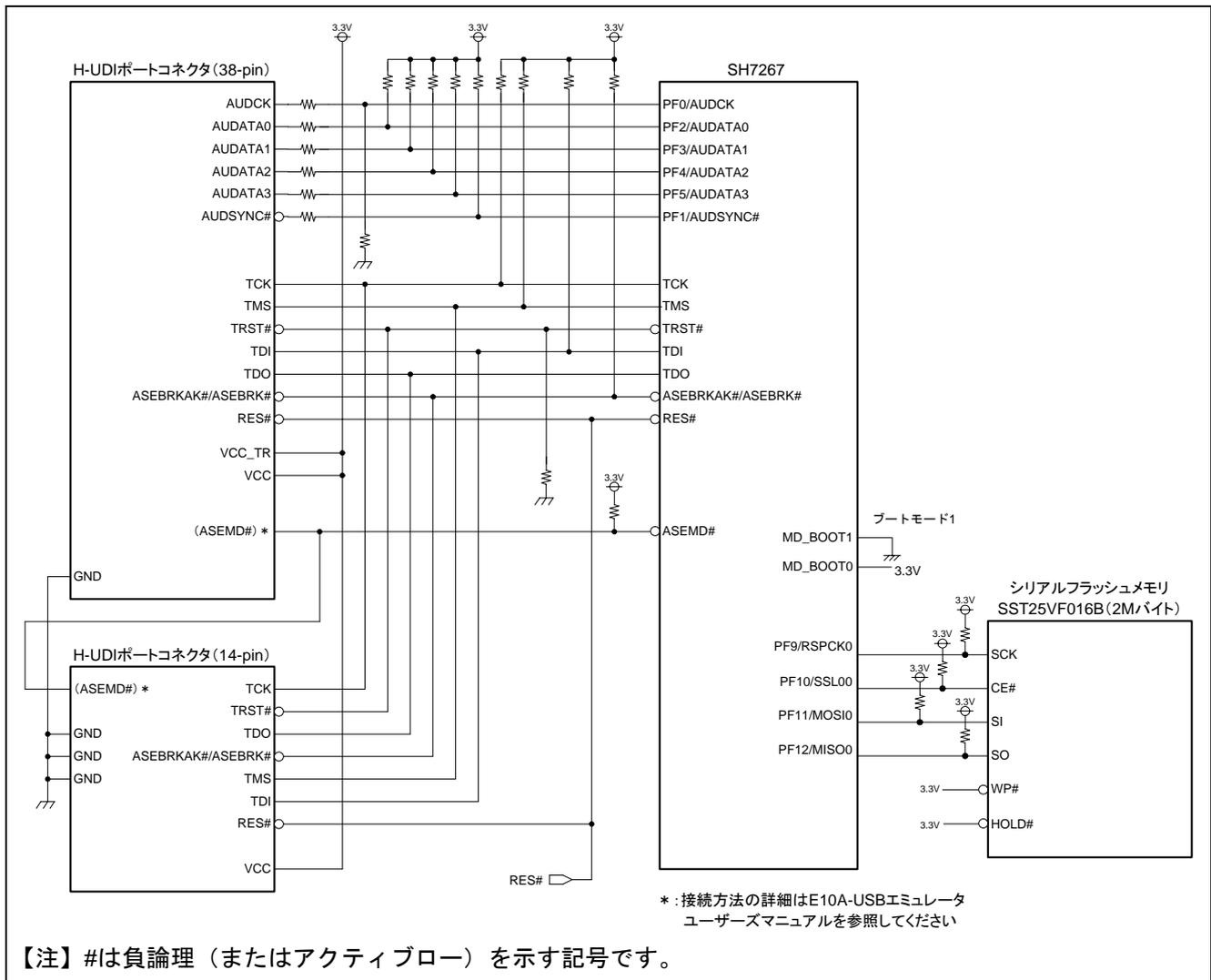


図 4.1 接続例

5. ソフトウェア説明

5.1 動作概要

FMTOOL は、イレーズモジュールとライトモジュールの2つのプログラムから構成されています。E10A-USB エミュレータは、この2つのプログラムを使用して、フラッシュメモリにプログラムデータおよび定数データを書き込みます。イレーズモジュールおよびライトモジュールの詳細仕様については、E10A-USB エミュレータ ユーザーズマニュアル「6.22 フラッシュメモリへのダウンロード機能」を参照してください。

5.1.1 バッチファイル

まず、SH7267 を初期化するためにリセットコマンドを実行します。コマンドの実行には、ロードモジュールのダウンロード前に起動されるバッチファイルを使用します。バッチファイルおよびリセットコマンドについては、統合開発環境のマニュアルを参照してください。

5.1.2 イレーズモジュール

図 5.1にFMTOOLにおけるイレーズモジュールの処理概要を示します。ロードモジュールのダウンロードを開始すると、FMTOOLがSH7267 の高速内蔵RAM上に転送され、イレーズモジュールは転送後一度のみ実行されます。

本来、イレーズモジュールは、フラッシュメモリのチップイレーズ処理を想定した機能ですが、ここではルネサスシリアルペリフェラルインタフェースの初期化と、シリアルフラッシュメモリのプロテクト解除を行います。

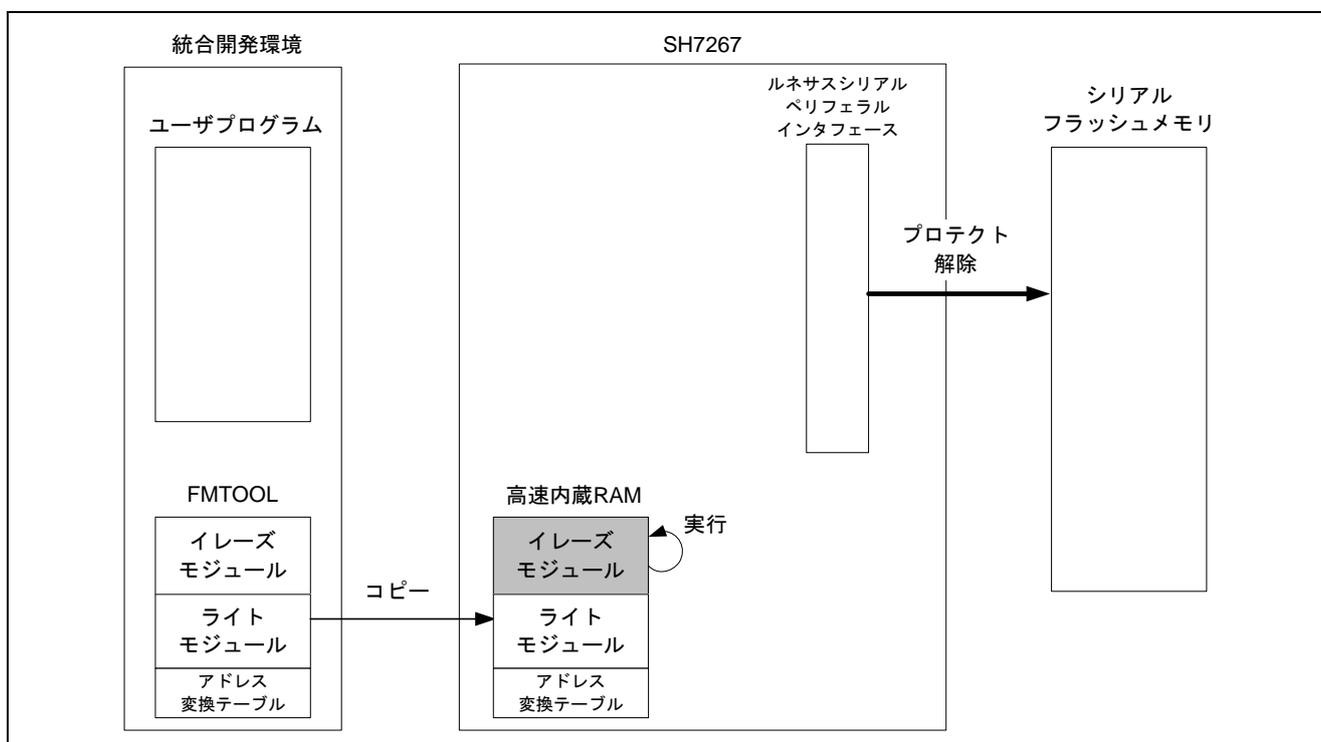


図 5.1 FMTOOL におけるイレーズモジュールの処理概要

5.1.3 ライトモジュール

図 5.2にFMTOOLにおけるライトモジュールの処理概要を示します。ライトモジュールは、ロードモジュールのダウンロード時に、高速内蔵RAM上で繰り返し実行されます。その際、ライトモジュールはアクセスサイズ単位に分割したプログラムデータを引数として受け取り、そのプログラムデータの書き込み先アドレスを計算した後、ページ単位にバッファリングし、シリアルフラッシュメモリに書き込みます。また、書き込み先が未消去セクタの場合は、セクタイレースを行います。

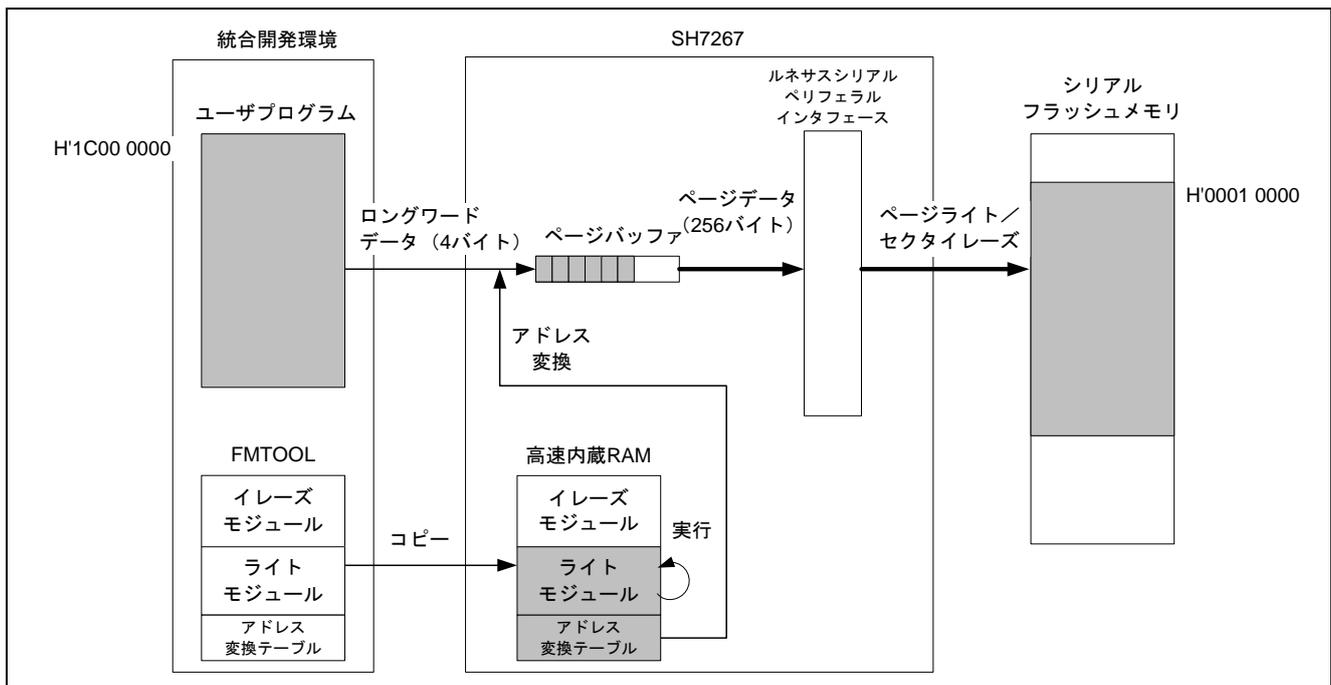


図 5.2 FMTOOL におけるライトモジュールの処理概要

ライトモジュールは、非連続な領域に配置されたプログラム、または定数データを効率良くシリアルフラッシュメモリに格納するために、アドレス変換を行っています。図 5.3にサンプルコードのアドレス変換を示します。変換規則は、「5.4 構造体/共用体一覧に示したアドレス変換テーブル」で定義していますので、必要に応じて変更してください。

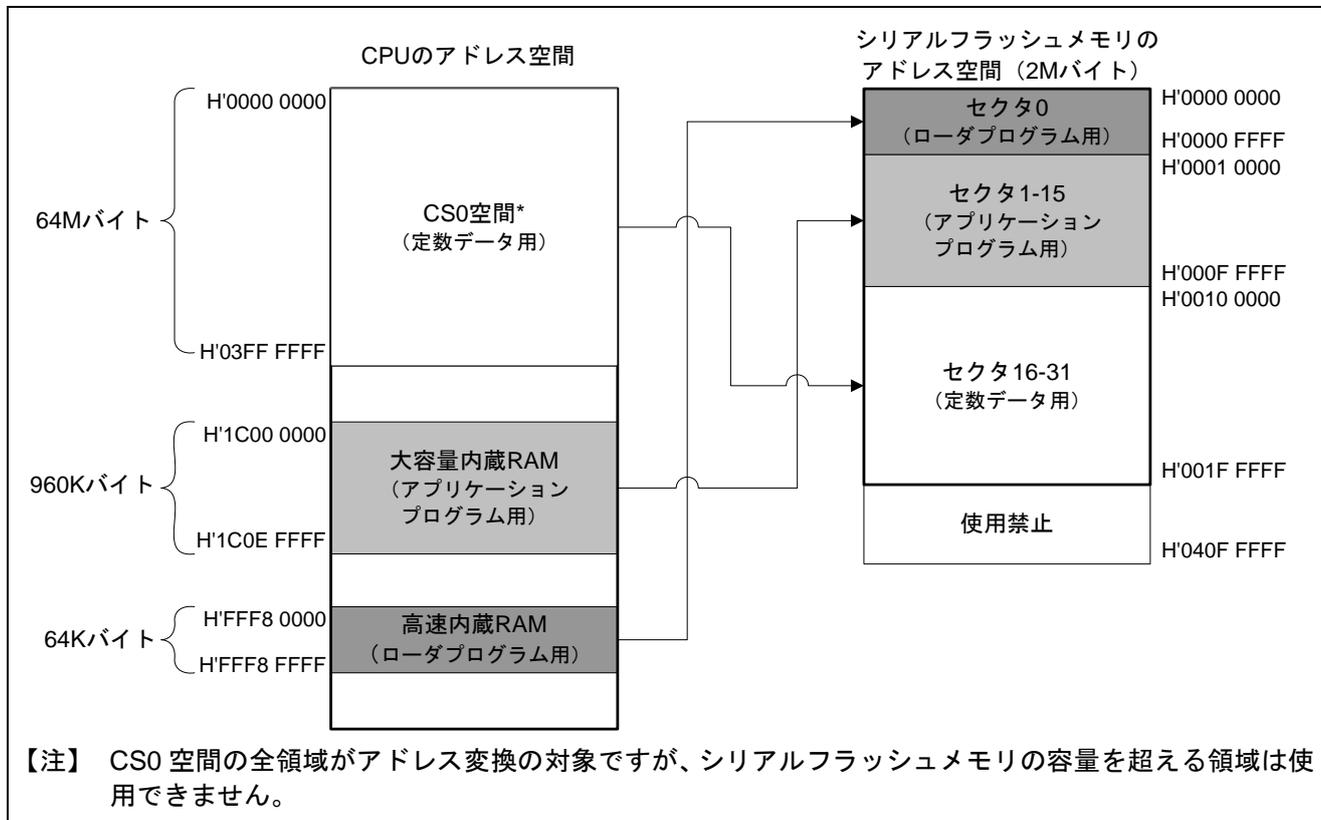


図 5.3 サンプルコードのアドレス変換

5.2 ファイル構成

表 5.1にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表 5.1 ファイル構成

ファイル名	概要	備考
fm_entry.src	FMTOOL のエントリモジュール	イレーズモジュールと ライトモジュールのエントリ
fm_main.c	FMTOOL のメイン処理	エントリから呼び出される関数
fm_cpg.c	CPG の初期化処理	
fm_r_sf_rsapi.c	シリアルフラッシュメモリ操作処理	
fm_io_rsapi.c	ルネサスシリアルペリフェラルインタ フェースの制御処理	
fm_map.c	アドレス変換テーブル	
cpg.h	fm_cpg.c の I/F 定義	
r_sf_rsapi.h	fm_r_sf_rsapi.c の I/F 定義	
io_rsapi.h	fm_io_rsapi.c の I/F 定義	
map.h	fm_map.c の I/F 定義	
sh7267_slash_fmtool.hdc	バッチファイル	ロードモジュール側のプロジェク トで使用
dummy.c	ロードモジュールのダミーデータ定義	ロードモジュール側のプロジェク トで使用

5.3 定数一覧

表 5.2にサンプルコードで使用する定数を示します。

表 5.2 サンプルコードで使用する定数

定数名	設定値	内容
SF_PAGE_SIZE	256	ページサイズ (256 バイト)
PAGE_SIZE	SF_PAGE_SIZE	"
SF_SECTOR_SIZE	(64*1024)	セクタサイズ (64K バイト)
SECTOR_SIZE	SF_SECTOR_SIZE	"
SF_REQ_PROTECT	0	シリアルフラッシュメモリのプロテクト指示
SF_REQ_UNPROTECT	1	シリアルフラッシュメモリのプロテクト解除指示
SR_Init	0x000000F0	ステータスレジスタの初期値
DEFAULT_VALUE	0xFFFFFFFF	FMTOOL で使用する管理データの初期値
TYPE_BYTE	0x4220	ライトモジュールの R5 パラメータ (データアクセスサイズは、バイトサイズ)
TYPE_WORD	0x5720	ライトモジュールの R5 パラメータ (データアクセスサイズは、ワードサイズ)
TYPE_LONG	0x4C20	ライトモジュールの R5 パラメータ (データアクセスサイズは、ロングサイズ)
FM_END_OF_TABLE	0xFFFFFFFF	アドレス変換テーブルの最終要素を示す値

5.4 構造体/共用体一覧

図 5.4にサンプルコードで使用する構造体/共用体を示します。

```

/* アドレス変換テーブル定義用構造体 */
typedef struct
{
    uint32_t src_top;           /* 変換元の先頭アドレス */
    uint32_t src_end;         /* 変換元の最終アドレス+1 */
    uint32_t dest_top;        /* 変換先の先頭アドレス */
} addr_tbl_t;

/* アドレス変換テーブル */
const addr_tbl_t g_fm_addr_tbl[] =
{
    /* src_top, src_end, dest_top */
    {0xFFF80000, 0xFFF90000, 0x00000000}, /* 高速内蔵RAM (ローダプログラム用) */
    {0x1C000000, 0x1C0F0000, 0x00010000}, /* 大容量内蔵RAM (キャッシュ有効空間) */
    {0x00000000, 0x04000000, 0x00100000}, /* CS0空間 (定数データ用) */
    {FM_END_OF_TABLE, 0, 0}
};

```

図 5.4 サンプルコードで使用する構造体/共用体

5.5 変数一覧

表 5.3にグローバル変数を、表 5.4にstatic型変数を示します。

表 5.3 グローバル変数

Type	Variable Name	Contents	Function Used
addr_tbl_t	g_fm_addr_tbl	アドレス変換テーブル	fmtool_write

表 5.4 static 型変数

Type	Variable Name	Contents	Function Used
uint32_t	fmtool_pre_erase_sctno	イレーズ済みセクタの管理情報	fmtool_init, fmtool_write
uint32_t	fmtool_cur_page	バッファリングしているページ の先頭アドレス	fmtool_init, fmtool_write
uint32_t	fmtool_page_buf[PAGE_SIZE / sizeof(uint32_t)]	ページバッファ	fmtool_write

5.6 関数一覧

表 5.5に関数を示します。

表 5.5 関数

関数名	概要
_ERASE_ENTRY	イレーズモジュールのエントリ処理
_WRITE_ENTRY	ライトモジュールのエントリ処理
fmtool_init	イレーズモジュールのメイン処理（初期化処理）
fmtool_write	ライトモジュールのメイン処理（イレーズ/ライト処理）
R_SF_RSPI_Init	シリアルフラッシュメモリ操作関数（ルネサスシリアルペリフェラルインタフェースおよびシリアルフラッシュメモリの初期化）
R_SF_RSPI_CtrlProtect	” （プロテクト制御）
R_SF_RSPI_EraseChip	” （チップイレーズ処理）
R_SF_RSPI_EraseSector	” （セクタイレーズ処理）
R_SF_RSPI_ByteProgram	” （ライト処理）
R_SF_RSPI_ByteRead	” （リード処理）※サンプルコードでは使用していません。
io_set_cpg	クロックパルスジェネレータ（CPG）の初期設定
io_init_rspi	ルネサスシリアルペリフェラルインタフェース（RSPI）の初期設定
io_cmd_exe	RSPI のデータ出力処理
io_cmd_exe_rdmode	RSPI のデータ入力処理

5.7 関数仕様

サンプルコードの関数仕様を示します。

_ERASE_ENTRY

概要	イレーズモジュールのエントリ処理
ヘッダ	なし
宣言	<code>_ERASE_ENTRY:</code>
説明	イレーズモジュールのエントリ部で、H'FFF8 2000 番地に配置します。本モジュールは E10A-USB フラッシュメモリダウンロード機能によって起動されます。本モジュールはスタックポインタの設定後、 <code>fmtool_init</code> 関数を実行します。
引数	R4 レジスタ : アクセスサイズ (バイト= H'4220、ワード= H'5720、ロング= H'4C20)
リターン値	なし
備考	アセンブリ言語で記述されています。

_WRITE_ENTRY

概要	ライトモジュールのエントリ処理
ヘッダ	なし
宣言	<code>_WRITE_ENTRY:</code>
説明	ライトモジュールのエントリ部で、H'FFF8 2100 番地に配置します。本モジュールは E10A-USB フラッシュメモリダウンロード機能によって起動されます。本モジュールはスタックポインタの設定後、 <code>fmtool_write</code> 関数を実行します。
引数	R4 レジスタ : ライトデータが配置されているアドレス R5 レジスタ : アクセスサイズ (バイト= H'4220、ワード= H'5720、ロング= H'4C20) R6 レジスタ : ライトデータ
リターン値	R0 レジスタ=0 の場合 : 正常終了 R0 レジスタ=1 の場合 : エラー終了
備考	アセンブリ言語で記述されています。

fmtool_init

概要	イレーズモジュールのメイン処理 (初期化処理)
ヘッダ	なし
宣言	<code>void fmtool_init(void);</code>
説明	ルネサスシリアルペリフェラルインタフェースとシリアルフラッシュメモリの初期化を行います。本関数は FMTOOL のエントリポイント (<code>_ERASE_ENTRY</code>) から実行します。
引数	なし
リターン値	なし
備考	

fmtreeol_write

概要	ライトモジュールのメイン処理 (イレーズ/ライト処理)
ヘッダ	なし
宣言	int32_t fmtreeol_write(uint32_t addr, int32_t access_size, uint32_t write_data);
説明	シリアルフラッシュメモリのイレーズおよびライト処理を行います。イレーズはセクタ単位、ライトはページ単位でシリアルフラッシュメモリをアクセスします。本関数は FMTTOOL のエントリポイント (WRITE_ENTRY) から実行します。
引数	第一引数 : addr : ライトデータが配置されているアドレス 第二引数 : size : アクセスサイズ (バイト= H'4220、ワード= H'5720、ロング= H'4C20) 第三引数 : write_data : ライトデータ
リターン値	0 の場合 : 正常終了 負の場合 : エラー終了
備考	アクセスサイズはロングワードサイズのみ対応します。

R_SF_RSPI_Init

概要	シリアルフラッシュメモリ操作関数 (ルネサスシリアルペリフェラルインタフェースおよびシリアルフラッシュメモリの初期化)
ヘッダ	"r_sf_rsapi.h"
宣言	void R_SF_RSPI_Init(void);
説明	ルネサスシリアルペリフェラルインタフェースの基本部分の初期設定を行います。また、シリアルフラッシュメモリのプロテクト解除を行います。
引数	なし
リターン値	なし
備考	

R_SF_RSPI_CtrlProtect

概要	シリアルフラッシュメモリ操作関数 (プロテクト制御)
ヘッダ	"r_sf_rsapi.h"
宣言	void R_SF_RSPI_CtrlProtect(en_sf_req_t req);
説明	シリアルフラッシュメモリのプロテクト解除またはセットを行います。
引数	第一引数 : req : プロテクトリクエスト (SF_REQ_PROTECT : プロテクトセット、 SF_REQ_UNPROTECT : プロテクト解除)
リターン値	なし
備考	

R_SF_RSPI_EraseChip

概要	シリアルフラッシュメモリ操作関数 (チップイレーズ処理)
ヘッダ	"r_sf_rsapi.h"
宣言	void R_SF_RSPI_EraseChip(void);
説明	シリアルフラッシュメモリのチップイレーズを行います。
引数	なし
リターン値	なし
備考	

R_SF_RSPI_EraseSector

概要	シリアルフラッシュメモリ操作関数 (セクタイレーズ処理)
ヘッダ	"r_sf_rsapi.h"
宣言	void R_SF_RSPI_EraseSector(int32_t sector_no);
説明	シリアルフラッシュメモリのセクタイレーズを行います。
引数	第一引数 : sector_no : イレーズするセクタ番号
リターン値	なし
備考	

R_SF_RSPI_ByteProgram

概要	シリアルフラッシュメモリ操作関数 (ライト処理)
ヘッダ	"r_sf_rsapi.h"
宣言	void R_SF_RSPI_ByteProgram(uint32_t addr, uint8_t * buf, int32_t size);
説明	シリアルフラッシュメモリへ引数で指定されたデータのライト処理を行います。バイトプログラムコマンド (H'02) または、自動アドレスインクリメント・ワードプログラムコマンド (H'AD) を使用します。
引数	第一引数 : addr : ライトアドレス (シリアルフラッシュメモリのアドレス) 第二引数 : buf : ライトデータ (バッファの先頭アドレス) 第三引数 : size : データバイト数
リターン値	なし
備考	

R_SF_RSPI_ByteRead

概要	シリアルフラッシュメモリ操作関数 (リード処理)
ヘッダ	"r_sf_rsapi.h"
宣言	void R_SF_RSPI_ByteRead(uint32_t addr, uint8_t * buf, int32_t size);
説明	シリアルフラッシュメモリへ引数で指定された領域のリード処理を行い、バッファに格納します。リードコマンド (H'0B) を使用します。
引数	第一引数 : addr : リードアドレス (シリアルフラッシュメモリのアドレス) 第二引数 : buf : リードバッファの先頭アドレス 第三引数 : size : データバイト数
リターン値	なし
備考	サンプルコードでは使用していません。

io_set_cpg

概要	クロックパルスジェネレータ (CPG) の初期設定
ヘッダ	"cpg.h"
宣言	void io_set_cpg(void);
説明	動作周波数と周辺モジュールへのクロック供給許可を設定します。
引数	なし
リターン値	なし
備考	

io_init_rspi

概要	ルネサスシリアルペリフェラルインタフェース (RSPI) の初期設定
ヘッダ	"io_rspi.h"
宣言	void io_init_rspi(void);
説明	RSPI のチャンネル 0 を初期化します。
引数	なし
リターン値	なし
備考	

io_cmd_exe

概要	RSPI のデータ出力処理
ヘッダ	"io_rspi.h"
宣言	void io_cmd_exe(uint8_t *ope, int32_t ope_sz, uint8_t *data, int32_t data_sz);
説明	引数で指定されたオペコードおよびデータを送信します。オペコードは 8 バイト以下にしてください。
引数	第一引数 : ope : オペコードの先頭アドレス 第二引数 : ope_sz : オペコードのサイズ 第三引数 : data : データの先頭アドレス 第四引数 : data_sz : データのサイズ
リターン値	なし
備考	

io_cmd_exe_rdmode

概要	RSPI のデータ入力処理
ヘッダ	"io_rspi.h"
宣言	void io_cmd_exe_rdmode(uint8_t *ope, int32_t ope_sz, uint8_t *rd, int32_t rd_sz);
説明	引数で指定されたオペコードを送信した後、指定サイズのデータを受信します。オペコードは 8 バイト以下にしてください。
引数	第一引数 : ope : オペコードの先頭アドレス 第二引数 : ope_sz : オペコードのサイズ 第三引数 : rd : データの先頭アドレス 第四引数 : rd_sz : データのサイズ
リターン値	なし
備考	

5.8 フローチャート

サンプルコードで使用する主な関数の処理の流れを示します。シリアルフラッシュメモリの操作関数およびRSPIの制御手順については、アプリケーションノート「SH7262/SH7264 グループ ルネサスシリアルペリフェラルインタフェース シリアルフラッシュメモリ接続例 (RJJ06B1001)」を参考にしてください。

5.8.1 イレーズモジュール

図 5.5にイレーズモジュールのフローチャートを示します。

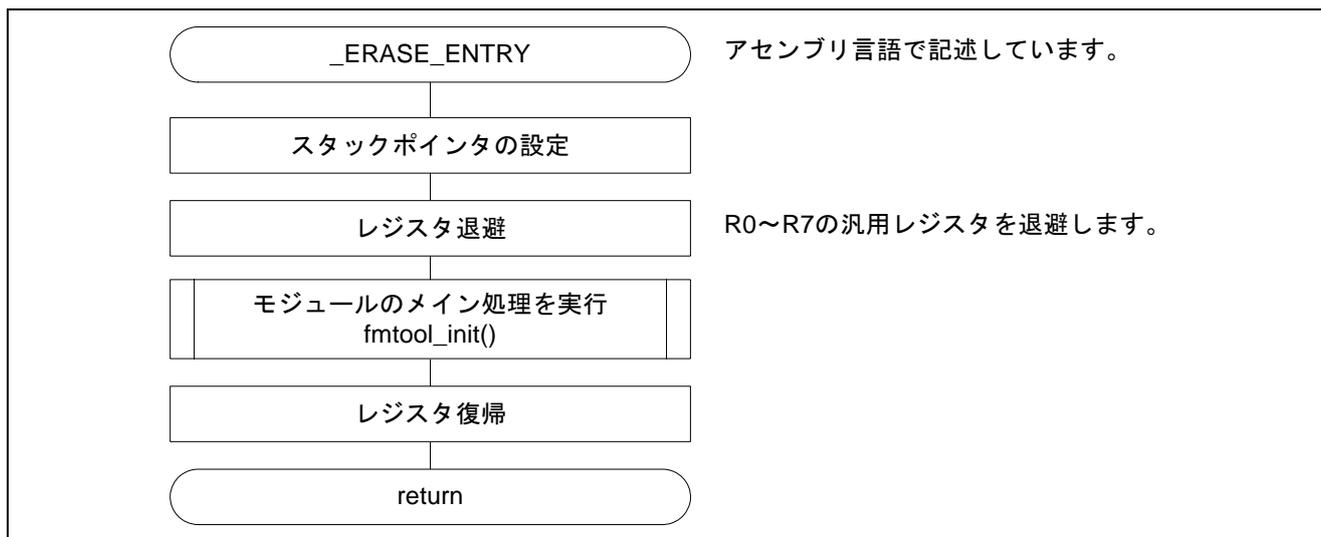


図 5.5 イレーズモジュール

5.8.2 ライトモジュール

図 5.6にライトモジュールのフローチャートを示します。

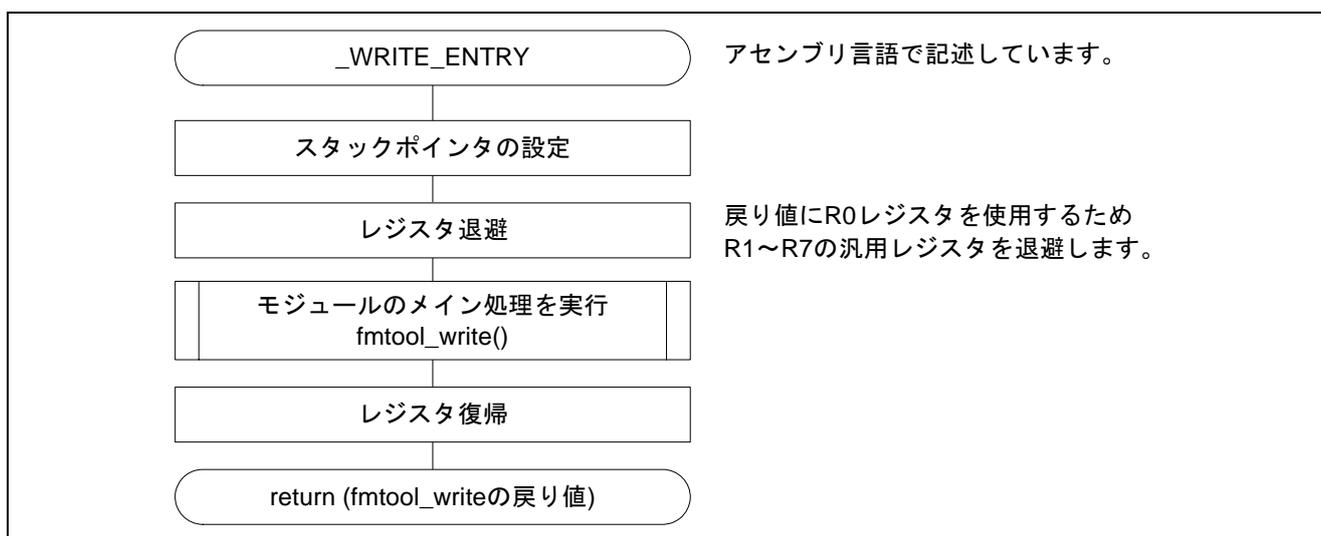


図 5.6 ライトモジュール

5.8.3 FMTOOL の初期化

図 5.7にFMTOOLの初期化フローチャートを示します。

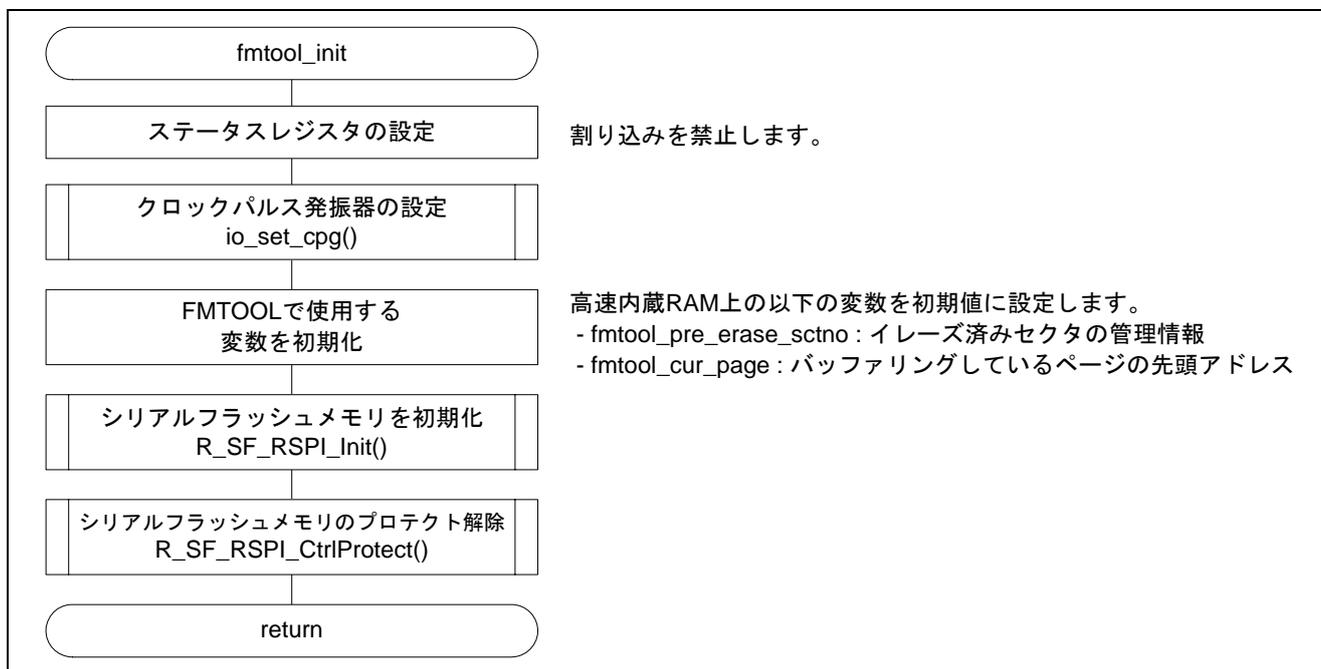


図 5.7 FMTOOL の初期化

5.8.4 フラッシュメモリ書き込み処理

図 5.8にフラッシュメモリ書き込み処理のフローチャートを示します。

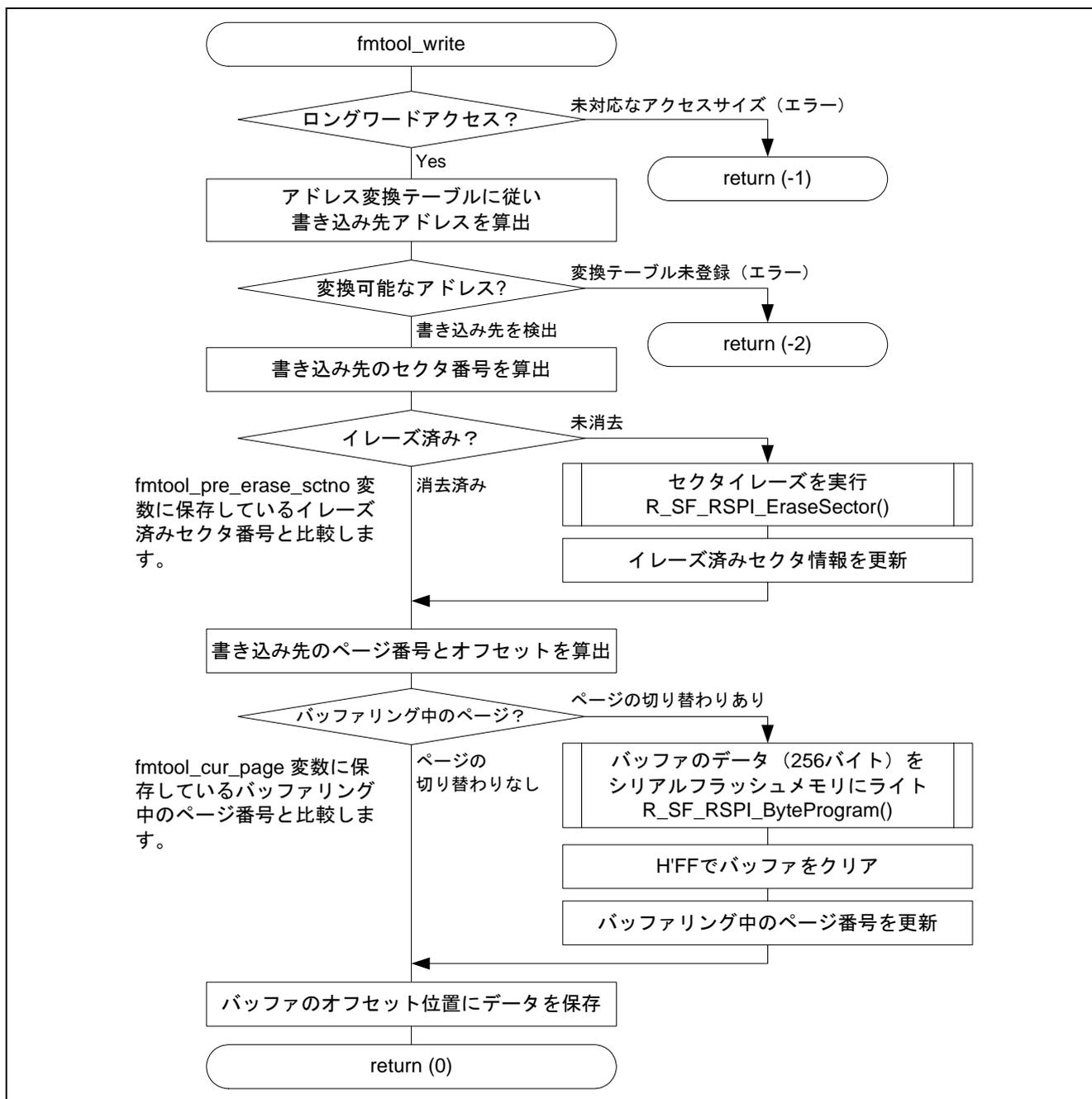


図 5.8 フラッシュメモリ書き込み処理

5.9 注意事項

5.9.1 ロードモジュールへのダミーデータ追加

シリアルフラッシュメモリへの書き込み速度を向上させるため、FMTOOL は書き込みデータをバッファリングしてページ単位に書き込みます。シリアルフラッシュメモリへの書き込みは、バッファリング中のページと異なるページのアドレスが指定されたタイミングで行われるため、最後の1ページ分のデータは、バッファに残ったままシリアルフラッシュメモリに書き込まれない可能性があります。有効データをバッファに残さないために、ロードモジュールの最後の1ページ分のデータにはダミーデータを配置してください。

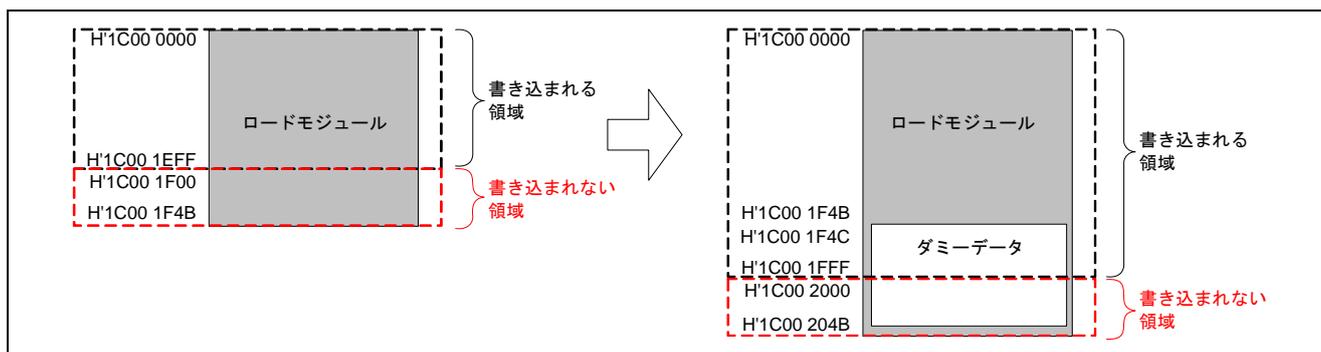


図 5.9 バッファに残って書き込まれない領域

図 5.10にダミーデータの追加例を示します。用意したダミーセクション (CDUMMY_MODULE_END) で 256 バイトの定数データを定義し、ROM領域の最後に配置します。

```
dummy.c
#define SF_PAGE_SIZE 256

#pragma section DUMMY_MODULE_END
const char dummy_area[SF_PAGE_SIZE] = { 0 };
#pragma section
```

セクション設定

Address	Section
0x1C000000	DAPPINFO
	DVECTTBL
	DINTTBL
0x1C000800	PResetPRG
	PIntPRG
0x1C001000	P
	C
	C\$BSEC
	C\$DSEC
	D
	PCACHE
	CDUMMY_MODULE_END
0xFFFF82000	RINTTBL
	B
	RPCACHE
0xFFFF8FC00	S

ROM領域の最後に配置する

図 5.10 ダミーデータの追加例

5.9.2 ロードモジュール間のセクタ共有禁止

図 5.11にロードモジュール間でセクタを共有した場合の動作を示します。FMTOOLで複数のロードモジュールをダウンロードすることも可能ですが、ロードモジュール間でセクタを共有することはできません。セクタを共有した場合、先にダウンロードしたロードモジュールのデータが誤って消去されます。

なお、ここでいうロードモジュールの領域には、5.9.1で説明したダミーデータの領域も含まれます。

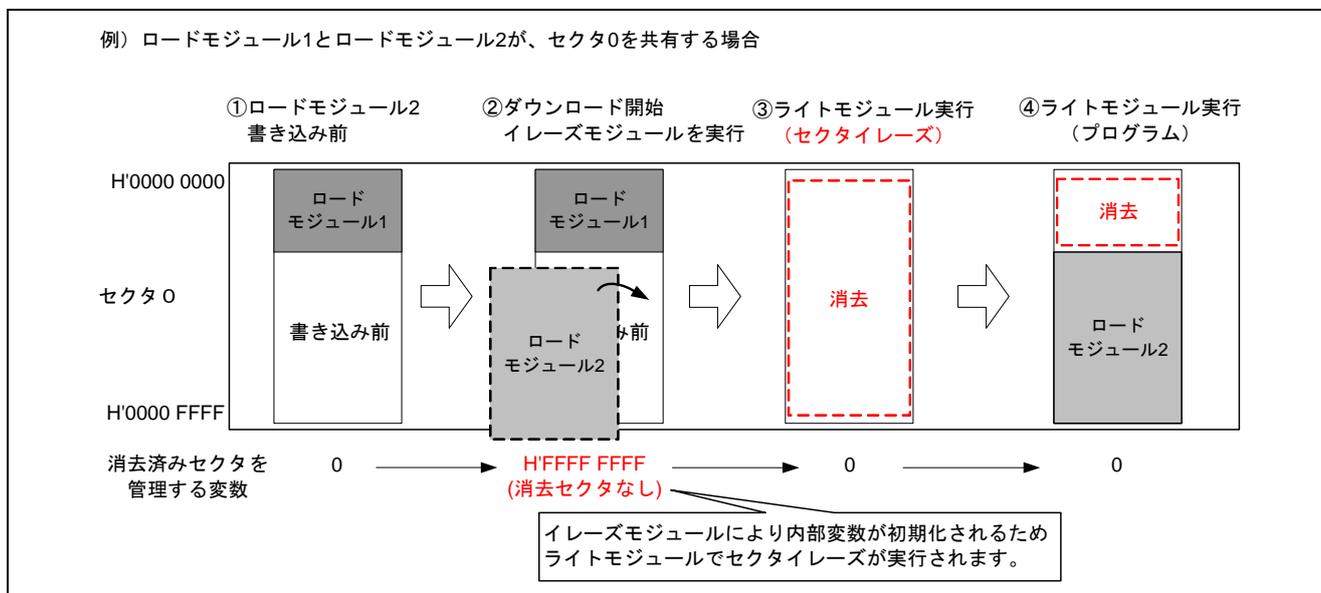


図 5.11 ロードモジュール間でセクタを共有した場合の動作

5.9.3 アドレス変換テーブル内のセクタ共有禁止

図 5.12にアドレス変換テーブル内でセクタを共有した場合の動作を示します。図の例では、アドレス変換テーブルのg_fm_addr_tbl[0]とg_fm_addr_tbl[1]が、セクタ 0 を共有しているため、g_fm_addr_tbl[1]の対象領域をダウンロードした際に、g_fm_addr_tbl[0]で書き込まれたデータが消去されます。

アドレス変換テーブルには、書き込みセクタを共有しない値を設定してください。

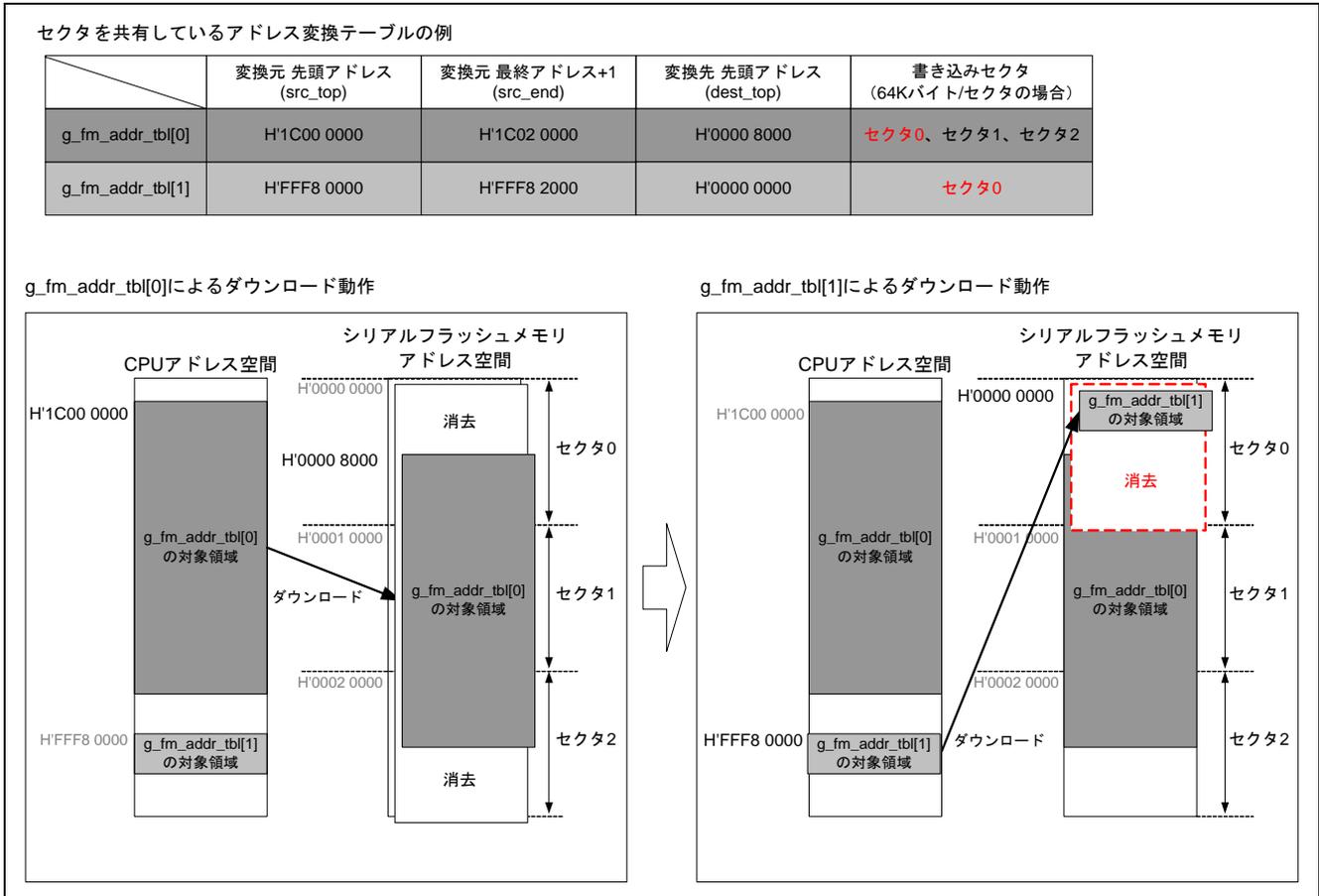


図 5.12 アドレス変換テーブル内でセクタを共有した場合の動作

6. 応用例

6.1 ユーザプログラムダウンロード手順

作成した FMTOOL (sh7267_sflash_fmtool.mot) を使用して、ユーザプログラムをシリアルフラッシュメモリにダウンロードする手順を以下に示します。

6.1.1 ダウンロード環境の準備

- (1) PC に接続した E10A-USB エミュレータとユーザシステムを接続します。
- (2) High-performance Embedded Workshop を起動し、ユーザプログラムのワークスペースを開きます。
- (3) 図 6.1に示すCPU selectダイアログボックスが表示されます。

ご使用の CPU をドロップダウンリストボックスより選択し、OK ボタンを押してください。

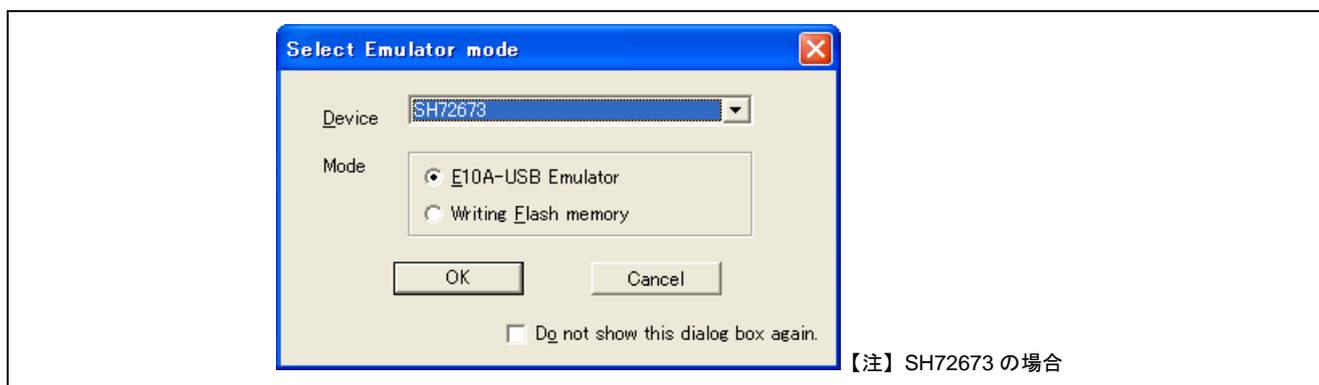


図 6.1 CPU select ダイアログボックス

- (4) Connectingダイアログボックスが表示され、エミュレータの接続を開始します。

図 6.2に示すRESET信号入力要求メッセージのダイアログボックスが表示されます。



図 6.2 RESET 信号入力要求メッセージのダイアログボックス

- (5) ユーザシステムの電源を入れます。

ユーザシステムからRESET信号を入力し、OKボタンをクリックします。

High-performance Embedded Workshop のOutputウィンドウに "Connected" と表示されたら、E10A-USBエミュレータの起動は完了です。

6.1.2 バッチファイルの登録

- (1) メニューの[デバッグ]→[デバッグの設定]を選択します。
- (2) 図 6.3に示すデバッグの設定ウィンドウが開きます。
- (3) "コマンドバッチファイル実行タイミング" のプルダウンメニューから、"Before download modules"を選択します。
- (4) "コマンドバッチファイル実行順序" の追加ボタンをクリックし、バッチファイルを追加します。
- (5) OK ボタンをクリックして登録は完了です。

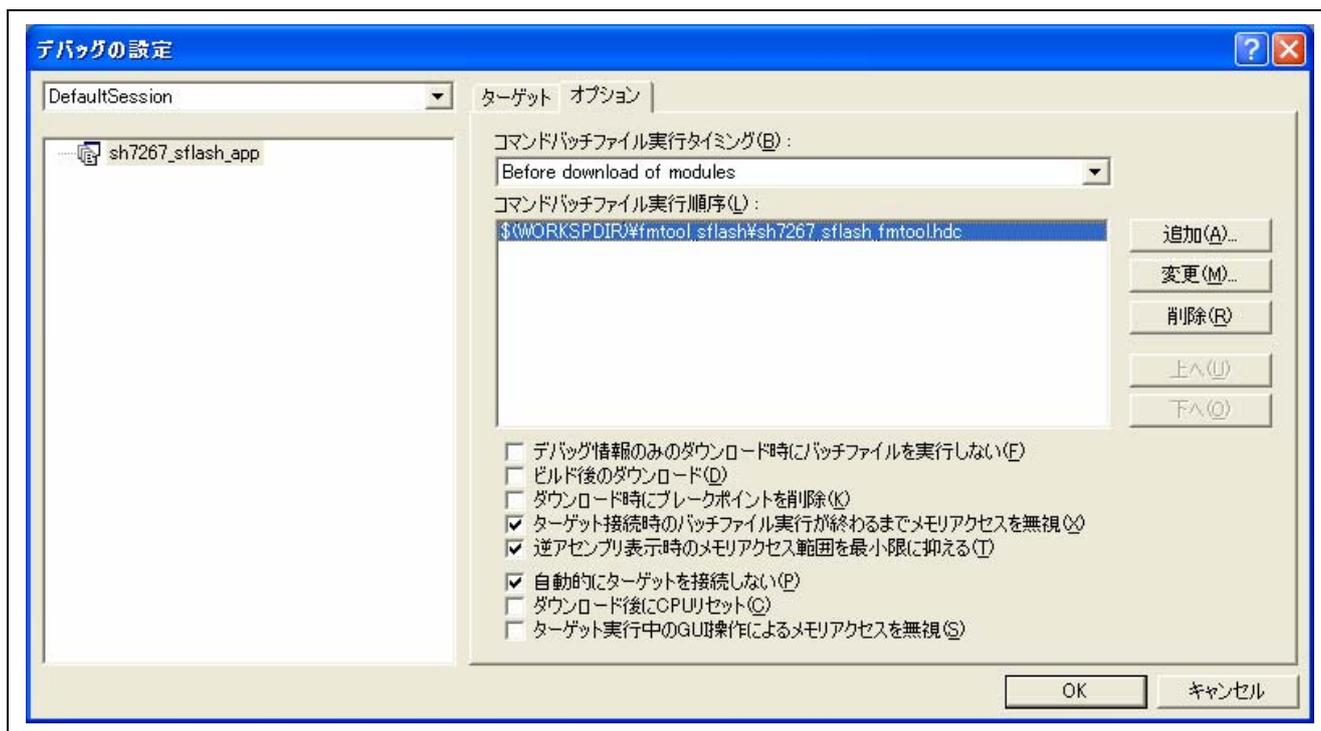


図 6.3 デバッグの設定ウィンドウ

6.1.3 Configuration ダイアログボックスの設定

- (1) メニューの[基本設定]→[エミュレータ]→[システム]を選択します。
- (2) 図 6.4に示すConfigurationダイアログボックス (Loading flash memoryページ) で、E10A-USBエミュレータを使用して外部フラッシュメモリにユーザプログラムをダウンロードするための設定を行います。

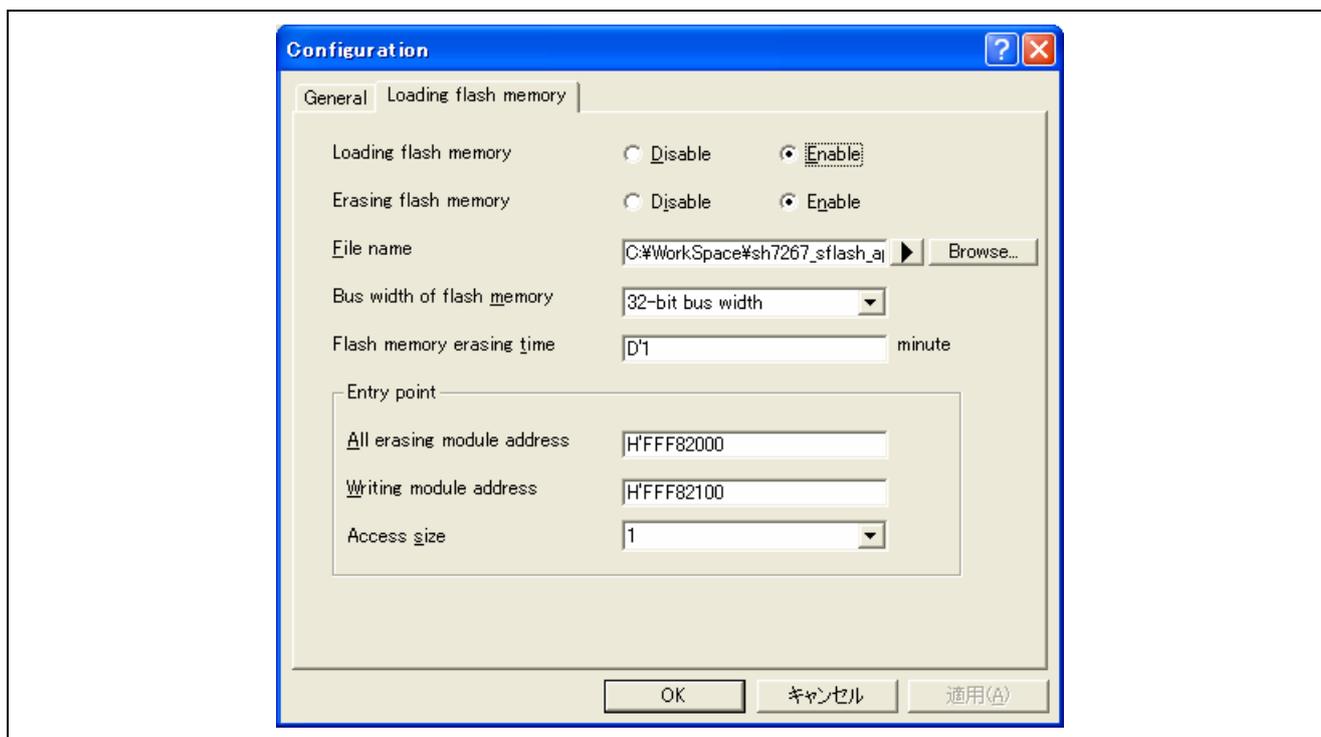


図 6.4 Configuration ダイアログボックス (Loading flash memory ページ)

表 6.1に各項目の設定を示します。設定が完了したらOKボタンをクリックして完了です。

表 6.1 Configuration ダイアログボックスの設定値

項目	設定値
Loading flash memory	Enable
Erasing flash memory	Enable
File Name	(FMTOOL を格納したディレクトリ)¥sh7267_sflash_fmtool.mot
Bus width of flash memory	32-bit bus width
All erasing module address	イレーズモジュールの先頭アドレスを指定 (H'FFF8 2000)
Writing module address	ライトモジュールの先頭アドレスを指定 (H'FFF8 2100)

6.1.4 ダウンロードモジュールの追加

デバッグメニューからデバッグの設定ウィンドウを開き、追加ボタンをクリックしてください。図 6.5に示すダウンロードモジュールウィンドウが表示されますので、ユーザプログラム (シリアルフラッシュメモリにロードしたいプログラム) をダウンロードモジュールに追加します。

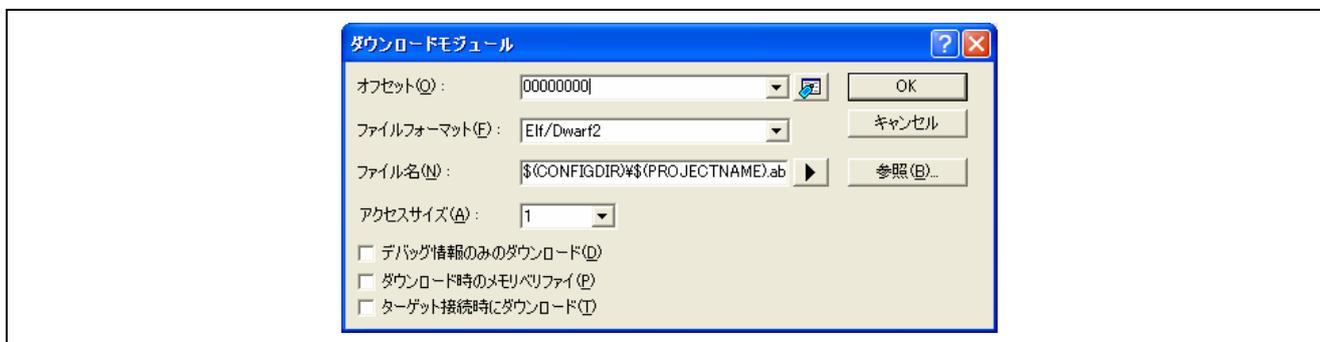


図 6.5 ダウンロードモジュールウィンドウ

6.1.5 ユーザプログラムのダウンロード

図 6.6に示すダウンロード機能にて、ユーザプログラムをダウンロードします。

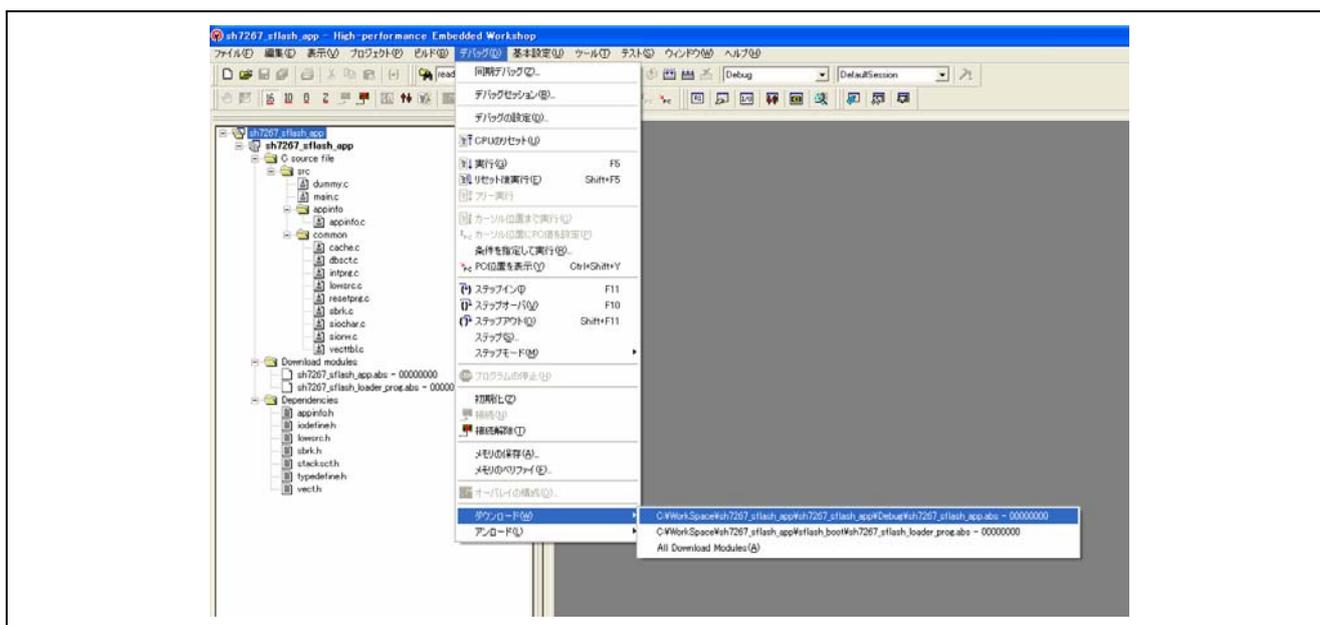


図 6.6 ユーザプログラムのダウンロード

6.2 シリアルフラッシュブートへの応用

シリアルフラッシュメモリからブートする機能を、ここでは、シリアルフラッシュブートと称します。シリアルフラッシュブートの詳細は、アプリケーションノート「SH7266/SH7267 グループ シリアルフラッシュメモリからのブート例 (R01AN0214JJ)」に記載されています。

また、サンプルコードには、本応用例に対応した処理が実装されていますので、併せて参照してください。

6.2.1 ダウンローダを FMTOOL に置き換える際の変更点

ここでは、上記アプリケーションノート(R01AN0214JJ)のフラッシュ書き込みツールであるダウンローダを FMTOOL に置き換える際の変更点について説明します。

(1) アプリケーションプログラムの格納番地変更

図 6.7にシリアルフラッシュブート時のセクション配置とアドレス変換の例を示します。ロードモジュール間でセクタを共有できないため、アプリケーションプログラムは、セクタ 1 以降に格納します。そのため、ローダプログラムで定義している APROG_TOP_SFLASHマクロの値を変更する必要があります。

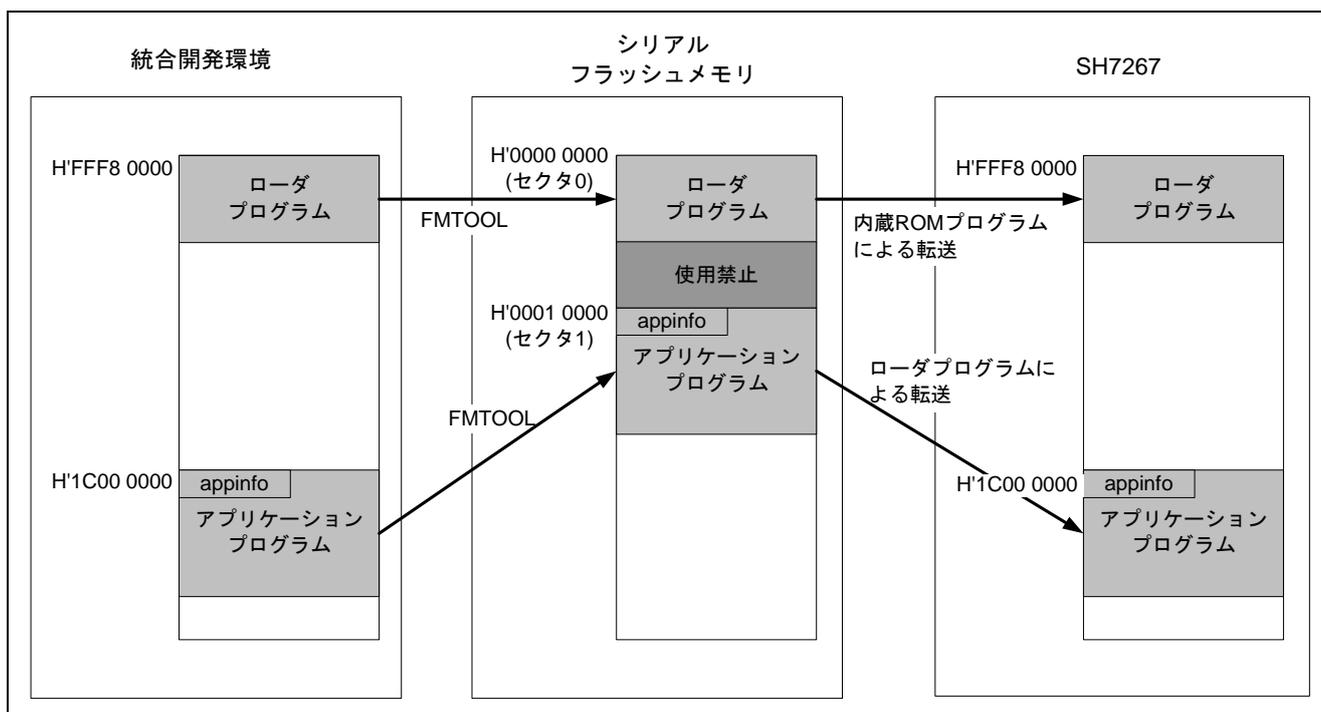


図 6.7 シリアルフラッシュブート時のセクション配置とアドレス変換の例

(2) ダミーデータの追加

「5.9.1 ロードモジュールへのダミーデータ追加」に示したとおり、ローダプログラムおよびアプリケーションプログラムの双方に、ダミーデータを追加する必要があります。

(3) ロードモジュールのダウンロード方法

ロードモジュールをダウンロードする際の統合開発環境の操作方法も変更されます。「6.1 ユーザプログラムダウンロード手順」に示した手順でダウンロードしてください。

6.2.2 定数データの保存と読み出し

ダウンローダは、RAM 上に展開したロードモジュールをシリアルフラッシュメモリにダウンロードしますが、FMTOOL は、RAM を介さずにダウンロードを行います。そのため、RAM 容量を超えるデータでも、シリアルフラッシュメモリに保存することができます。

ここでは、定数データをシリアルフラッシュメモリに保存し、必要に応じて局所的に読み出す方法を説明します。画像データなどの容量の大きい定数データを扱う際に有効です。

図 6.8に定数データの保存と読み出し動作例を示します。アプリケーションプログラムは、以下のように実装してください。

- ・ 未使用空間（CS0 空間など）に定義したセクション内で、const 修飾の定数データを宣言*
- ・ 定数データは、変数名から直接アクセスしない
- ・ RSPI のアクセス関数を実装し、必要に応じてシリアルフラッシュメモリから定数データを読み出す

【注】 セクションを配置する領域は、アドレス変換テーブルに登録されている必要があります。

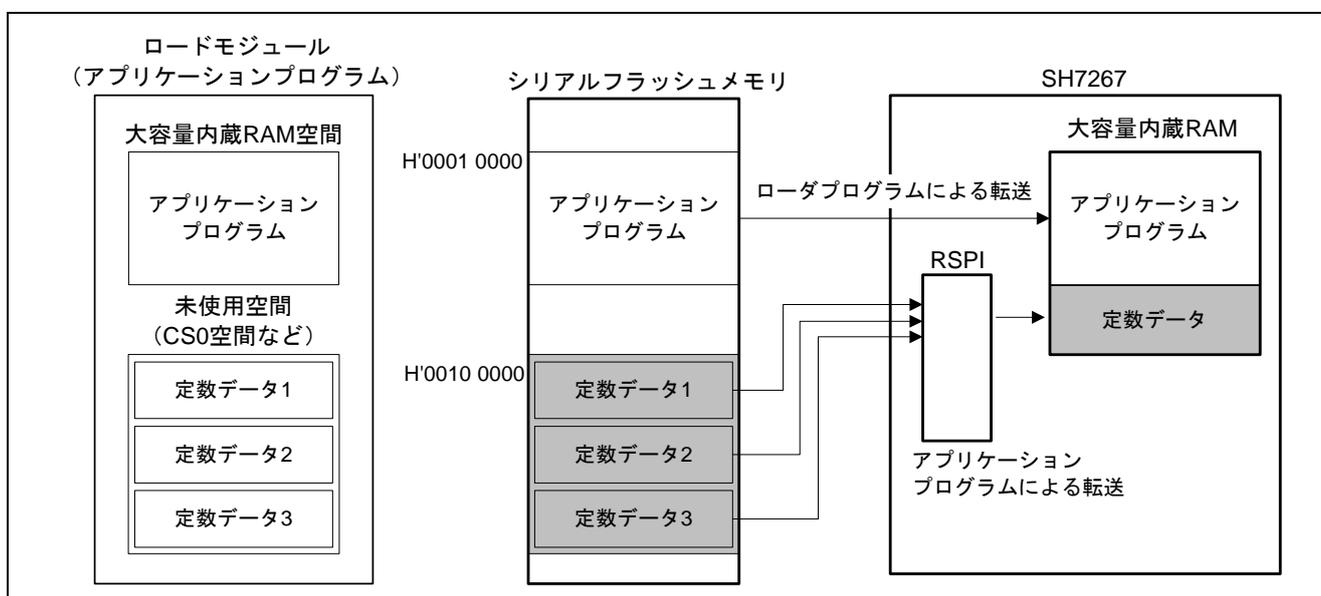


図 6.8 定数データの保存と読み出し動作例

6.3 FMTOOL のカスタマイズ

サンプルコードは、シリアルフラッシュメモリのデバイス仕様に依存するため、デバイス変更時にはプログラムのカスタマイズが必要となる場合があります。

6.3.1 サンプルコードに対応するデバイス仕様

表 6.2と表 6.3に、使用デバイスの詳細仕様およびサンプルコードで使用するコマンドを示します。

表 6.2 使用デバイスの詳細仕様

項目	内容
メーカー	Silicon Storage Technology, Inc.
型名	SST25VF016B
容量	2M バイト
インタフェース	4 線式 SPI バス (マルチ I/O バス非対応)
アクセスタイム	80MHz
セクタ構造	ユニフォーム型
セクタサイズ	64K バイト
ページサイズ	256 バイト

表 6.3 サンプルコードで使用するコマンド

項目	内容
イレーズコマンド	H'D8 (64KB セクタイレーズ)
プログラムコマンド	H'02 (バイトプログラミング) H'AD (自動アドレスインクリメント・ワードプログラミング)

6.3.2 カスタマイズの内容

以下にカスタマイズが必要なケースと、その変更内容を示します。

表 6.4 カスタマイズが必要なケースとカスタマイズ内容

項目	カスタマイズ内容
セクタサイズが異なる場合 (64K バイトのセクタイレーズに対応していない場合)	セクタ構造がユニフォーム型の場合は、SF_SECTOR_SIZE マクロの設定値を新しいセクタサイズに変更し、R_SF_RSPI_EraseChip 関数で使用するセクタイレーズコマンドを新しいセクタサイズに対応したコマンドに変更してください。セクタ構造がトップ型またはボトム型の場合は、fmtreeol_write 関数で行っているセクタ番号の判別アルゴリズムも変更する必要があります。
デバイスの初期設定手順が異なる場合 表 6.3のコマンドが使用できない場合 電気的特性が異なる場合	シリアルフラッシュメモリの操作関数およびルネサスシリアルペリフェラルインタフェースの制御関数のカスタマイズが必要です。詳細はサンプルコードを参照してください。

【注】 FMTOOLはフラッシュメモリの仕様に依存するため、表 6.4が全てのチェック項目を網羅しているわけではありません。使用されるデバイスのデータシートをご確認の上、デバイスの仕様に合わせて FMTOOLを修正してください。

7. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

8. 参考ドキュメント

SH7266/SH7267 グループ ユーザーズマニュアル ハードウェア編 Rev.1.00

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

C コンパイラマニュアル

SuperH RISC engine ファミリ C/C++コンパイラパッケージ V.9.04

C コンパイラユーザーズマニュアル Rev.1.01

(最新版をルネサス エレクトロニクスホームページから入手してください。)

SuperH ファミリ用 E10A-USB エミュレータ ユーザーズマニュアル Rev.9.00

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com>

お問合せ先

<http://japan.renesas.com/inquiry>

改訂記録	SH7266/SH7267 グループ E10A-USB フラッシュメモリダウンロード機能の応用例（シリアルフラッシュメモリへのダウンロード）
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.05.18	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連して発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続きを行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/inquiry>