

SH ファミリ

R01AN0285JJ0104

Rev.1.04

2012.06.08

SH-2, SH-2A 用シンプルフラッシュ API

要旨

本アプリケーションプログラムインタフェース (以下、API と称します) は、フラッシュメモリ内蔵の SH-2, SH-2A マイコンデバイスを使用するユーザアプリケーションに対し、ユーザプログラムモードにおけるフラッシュ書き換え機能の組み込みを容易にすることを目的に開発されました。ユーザプログラムモードは、通常運用動作中に内蔵フラッシュメモリの書き換えを可能にするための、ルネサスエレクトロニクス製マイコンデバイスの MCU 動作モードです。本アプリケーションノートでは、API の使用方法、およびユーザアプリケーションへの API 組み込み方法について説明します。

本 API はルネサスエレクトロニクス製 SuperH ファミリ用 C/C++ コンパイラのみにて動作確認しています。

対象デバイス

- SH-2/7125 グループ : SH7125, SH7124
- SH-2/7146 グループ : SH7149, SH7146
- SH-2/7137 グループ : SH7137, SH7136
- SH-2/7080 グループ : SH7086, SH7085, SH7084, SH7083
- SH-2A/7211 グループ : SH7211
- SH-2A/7280 グループ : SH7286, SH7285, SH7243
- SH-2A/7216 グループ : SH7216, SH7214
- SH-2A/7239 グループ : SH7239, SH7237
- SH-2A/7231 グループ : SH7231

目次

1. API ファイル	2
2. API の設定	3
3. 注意 / 制限事項	5
4. RAM への全アプリケーション転送	8
5. API 関数仕様	10
6. API 使用例	16
7. サンプルコード	17
8. 参考ドキュメント	17

1. API ファイル

APIファイルは、同じグループに属するマイコンデバイスごとに区別しています。表 1に、対象デバイス別のAPIファイル一覧を示します。

表1 対象デバイス別のAPI ファイル一覧

デバイス	グループ	API ファイル
SH7125 (SH-2) SH7124 (SH-2)	SH7125 グループ	Flash_API_SH7125.h Flash_API_SH7125.c
SH7149 (SH-2) SH7146 (SH-2)	SH7146 グループ	Flash_API_SH7146.h Flash_API_SH7146.c
SH7137 (SH-2) SH7136 (SH-2)	SH7137 グループ	Flash_API_SH7137.h Flash_API_SH7137.c
SH7086 (SH-2) SH7085 (SH-2) SH7084 (SH-2) SH7083 (SH-2)	SH7080 グループ	Flash_API_SH7080.h Flash_API_SH7080.c
SH7211 (SH-2A)	SH7211 グループ	Flash_API_SH7211.h Flash_API_SH7211.c
SH7286 (SH-2A) SH7285 (SH-2A) SH7243 (SH-2A)	SH7280 グループ	Flash_API_SH7280.h Flash_API_SH7280.c
SH7216 (SH-2A) SH7214 (SH-2A)	SH7216 グループ	Flash_API_SH7216.h Flash_API_SH7216.c
SH7239 (SH-2A) SH7237 (SH-2A)	SH7239 グループ	Flash_API_SH7239.h Flash_API_SH7239.c
SH7231 (SH-2A)	SH7231 グループ	Flash_API_SH7231.h Flash_API_SH7231.c

2. API の設定

API を使用するためには、使用環境に合わせた設定が必要になります。設定は全て、Flash_API_SHxxxx.h ファイル内の #define 定義値を変更することにより行います。

2.1 システムクロック動作速度の設定

【注】 本設定は、SH7216 グループ/SH7239 グループ/SH7231 グループでは必要ありません。

マイコンデバイスのフラッシュ書き換えを行うプログラムコードは CPU により実行されます。そのため、CPU 動作速度を API に設定し、ソフトウェアによる正確なウェイト処理を実現する必要があります。このウェイト処理が不正確な場合、フラッシュ消去/書き込み処理が正常に動作しない要因となります。CPU 動作速度には、フラッシュを制御する期間の CPU 内部クロック ($I\phi$) を設定します。フラッシュ消去/書き込み処理を行わない期間については、異なる速度でシステムを動作させても問題ありません。

具体的には CPU 動作速度を Flash_API_SHxxxx.h ファイル内の OPERATING_FREQUENCY パラメータで設定します。CPU 動作速度は Hz 単位です。例えば、フラッシュ消去/書き込み関数を実行する際に CPU を 25 MHz で動作させる場合、「25000000」を設定します。

```
#define OPERATING_FREQUENCY    25000000
```

【注】 SH マイコンの中には、フラッシュ制御処理を行う際の最大動作速度が規定されているものがあります。最大動作速度の値が規定されているマイコンについては、「OPERATING_FREQUENCY」の値を設定する Flash_API_SHxxxx.h ファイルにその旨記載しています。

2.2 入力クロック動作速度の設定

【注】 本設定は、SH7216 グループ/SH7239 グループ/SH7231 グループでのみ必要です。

SH7216/SH7239/SH7231 ではフラッシュの書き換え (消去/書き込み) は、周辺クロック ($P\phi$) で動作するフラッシュ制御ユニット (FCU) により実行されます。フラッシュの書き換え時間制御を行うため、 $P\phi$ を FCU に設定する必要があります。

具体的には入力クロックを逡倍した周波数を Flash_API_SH7216.h、Flash_API_SH7239.h、Flash_API_SH7231.h ファイル内の PLL_FREQUENCY パラメータで設定します。入力クロック動作速度は MHz 単位です。SH7216/SH7239 では入力クロックを 16 逡倍した周波数を、SH7231 では 8 逡倍した周波数を設定します。例えば、SH7216 では入力クロックが 12.5 MHz の場合、16 逡倍され 200 MHz (=12.5 MHz×16) となるので、「200」を設定します。逡倍した後の数値の小数点以下が 0 以外になる場合は、切り上げた値を設定してください。

```
#define PLL_FREQUENCY    200
```

本設定値は、API 内部で $P\phi$ に変換された後、FCU へ設定されます。

2.3 フラッシュ制御用 RAM 領域の選択

【注】 本節は、SH7216 グループ/SH7239 グループ/SH7231 グループには適用されません。

フラッシュメモリの消去/書き込みを行う内蔵プログラムは、RAM 上で実行する必要があります。本アプリケーションノートで対象とする SH-2, SH-2A マイコンでは、その実行形式のバイナリコードをマイコン内の非公開領域に格納しています。本 API は、このバイナリコードを、フラッシュ消去/書き込み時に使用する RAM 領域へダウンロードします。必要な RAM サイズは 3 K バイトで、配置先は Flash_API_SHxxxx.h ファイル内の FTDAR_VALUE パラメータで選択します。RAM の配置先を選択する際、その選択肢は少数に限定しており、マイコンにより異なります。使用可能な領域については、対象デバイスの Flash_API_SHxxxx.h ファイルを参照してください。

【注】 上記の RAM 領域は、フラッシュ消去/書き込み処理を行うまでは、他のユーザアプリケーションでも使用できます。ただし、一旦 R_FlashErase および R_FlashWrite 関数を実行すると、FTDAR_VALUE パラメータで選択した RAM 領域がフラッシュメモリの消去/書き込みを行う内蔵プログラムで上書きされ、そこに格納したユーザアプリケーションのデータは失われます。フラッシュ制御処理を行った後に再度上記の RAM 領域を使用した場合、グローバル変数の「current_firmware」を H'00 にリセットするまで、R_FlashErase および R_FlashWrite 関数は実行しないでください。R_FlashErase および R_FlashWrite 関数を再度実行するために、内部のバイナリコードを再度ダウンロードする必要があります。

3. 注意／制限事項

3.1 High-performance Embedded Workshop によるデバッグ

全ての SH マイコンが、E10A-USB エミュレータを使用したユーザアプリケーションによる内蔵フラッシュの消去および書き換え動作に対応している訳ではありません。E10A-USB エミュレータを使用したユーザアプリケーションによる内蔵フラッシュの消去および書き換え動作に対応していない SH マイコンの場合、E10A-USB エミュレータのユーザーズマニュアルにて、制限事項として「ユーザプログラム実行中に MCU 内蔵フラッシュメモリの内容の書き換えはできません。」と記載しています。デバッグ前に、ご使用になる E10A-USB エミュレータのユーザーズマニュアルの制限事項を確認してください。

また、ユーザシステムと E10A-USB エミュレータとの接続時に、「New ID Code」を選択または前回接続時と異なる ID コードを入力した場合、内蔵フラッシュの全領域が消去されます。そのため、接続前にユーザアプリケーションでフラッシュ書き換えを実施していた場合、その書き換え結果は接続が確立した時点で消去されますので注意してください。

このことは、ルネサス エレクトロニクス製フラッシュ開発ツールキット (FDT) を使用して書き込みを行う場合も同様に当てはまります。FDT ツールを使用する場合は、ブートモードに遷移する際に、内蔵セキュリティ機能によってフラッシュメモリの全領域が自動消去されます。

上記の理由により、本アプリケーションノートのサンプルアプリケーションでは、シリアルポートを使用してフラッシュメモリの内容を出力します。PC のターミナルアプリケーション (Windows の HyperTerminal 等) をシリアルポートに接続することで、容易にかつ信頼性のある方法でフラッシュメモリの「実際の」内容を確認することができます。

SH7216 グループ／SH7239 グループ／SH7231 グループでは、E10A-USB エミュレータ使用中でもモニタリングおよびフラッシュ書き換えが可能な「データフラッシュ」領域を別途保有しています。

3.2 SH7125 グループ／SH7146 グループ／SH7137 グループ／SH7080 グループ

SH7125 グループ : SH7125, SH7124

SH7146 グループ : SH7149, SH7146

SH7137 グループ : SH7137, SH7136

SH7080 グループ : SH7086, SH7085, SH7084, SH7083

E10A-USB エミュレータを使用したデバッグ中は、フラッシュ制御処理は実行できません。

また、フラッシュメモリ書き込み／消去中において割り込み処理を許可している場合、当該割り込み処理ルーチンの実行中にフラッシュメモリのエラー監視機能の誤動作によりフラッシュメモリの書き込み／消去動作が強制的に中断され、フラッシュメモリの書き込み／消去動作が正常に完了しない場合があります。したがって、これらのマイコンデバイスグループにおけるフラッシュメモリの書き込み／消去処理中は、NMI、IRQ、およびその他すべての割り込みを発生させないでください。

3.3 SH7211 グループ／SH7280 グループ

SH7211 グループ : SH7211

SH7280 グループ : SH7286, SH7285, SH7243

E10A-USB エミュレータを使用したデバッグ中は、フラッシュ制御処理は実行できません。

また、これらのマイコンデバイスグループにおけるフラッシュメモリの書き込み／消去処理中は、NMI、IRQ、およびその他すべての割り込みを発生させないでください。

3.4 SH7216 グループ／SH7239 グループ／SH7231 グループ

SH7216 グループ : SH7216, SH7214

SH7239 グループ : SH7239, SH7237

SH7231 グループ : SH7231

これらのマイコンデバイスグループにおいて本 API を実装する際は、ユーザプログラム領域を書き換える API 関数を格納するためのセクションを、RAM およびフラッシュ上に配置する必要があります。また、リセット後は RAM のセクション領域を初期化する必要があります。ただし、初期化が必要になるのはユーザプログラム領域に対し書き換えを行う場合のみです。データフラッシュ領域のみ書き換える場合、これらの設定は必要ありませんが、その際にはあらかじめ Flash_API_SH7216.h、Flash_API_SH7239.h、Flash_API_SH7231.h ファイル内の「ENABLE_ROM_PROGRAMMING」パラメータの定義値を「0」に変更する必要があります。

以下、セクションの設定手順を Step 1～4 に沿って説明します。

Step 1 : High-performance Embedded Workshop のリンカ設定にて、新規セクション「**RPFRAM**」を RAM 上に追加します。

Step 2 : High-performance Embedded Workshop のリンカ設定にて、新規セクション「**PFRAM**」をフラッシュ上に追加します。

図 1 に、Step 1 および Step 2 のセクション設定例を示します。

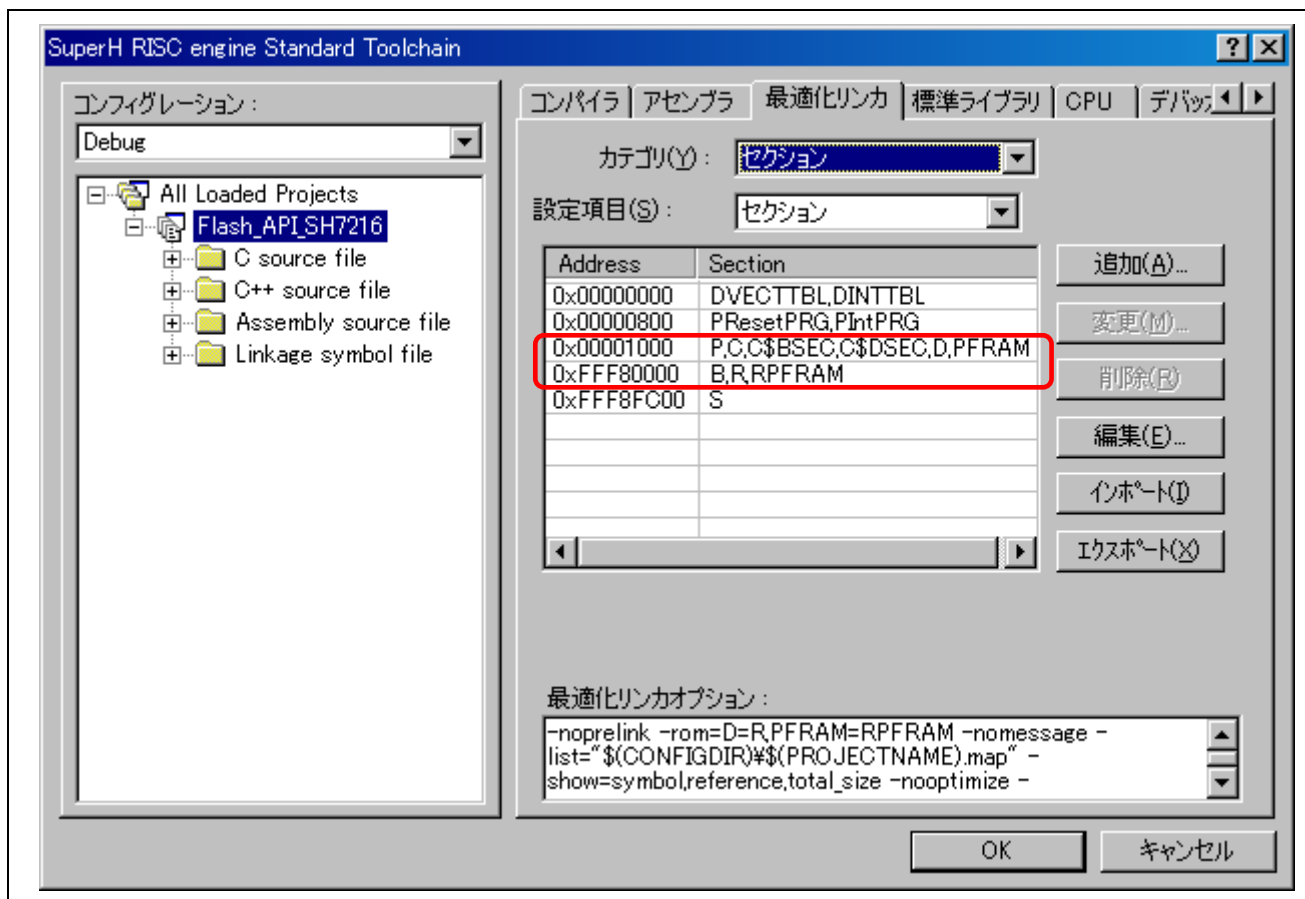


図1 High-performance Embedded Workshop のセクション設定例

Step 3: スタートアップファイルの中の「**dbstc.c**」で、RAMセクションの初期化に必要な設定を行います。「**dbstc.c**」ファイル内で、図2の赤字で示す行を追加してください。

【注】 追加した行の前の行の最後尾には、必ずカンマを追加してください。

```
#pragma section $DSEC
static const struct {
    _UBYTE *rom_s; /* Initial address on ROM of initialization data section */
    _UBYTE *rom_e; /* Final address on ROM of initialization data section */
    _UBYTE *ram_s; /* Initial address on RAM of initialization data section */
} DTBL[] = {
    { __sectop("D"), __secend("D"), __sectop("R") },
    { __sectop("PFRAM"), __secend("PFRAM"), __sectop("RPFRAM") }
};
```

図2 「dbstc.c」ファイルにおけるRAMセクションの初期化設定例

Step 4: High-performance Embedded Workshop のリンカ設定にて、ROMセクション (PFRAM) のアドレスをRAMセクション (RPFRAM) のアドレスにマッピングするためのROM化支援オプションを追加します。

図3に、Step4のROM化支援オプション設定例を示します。

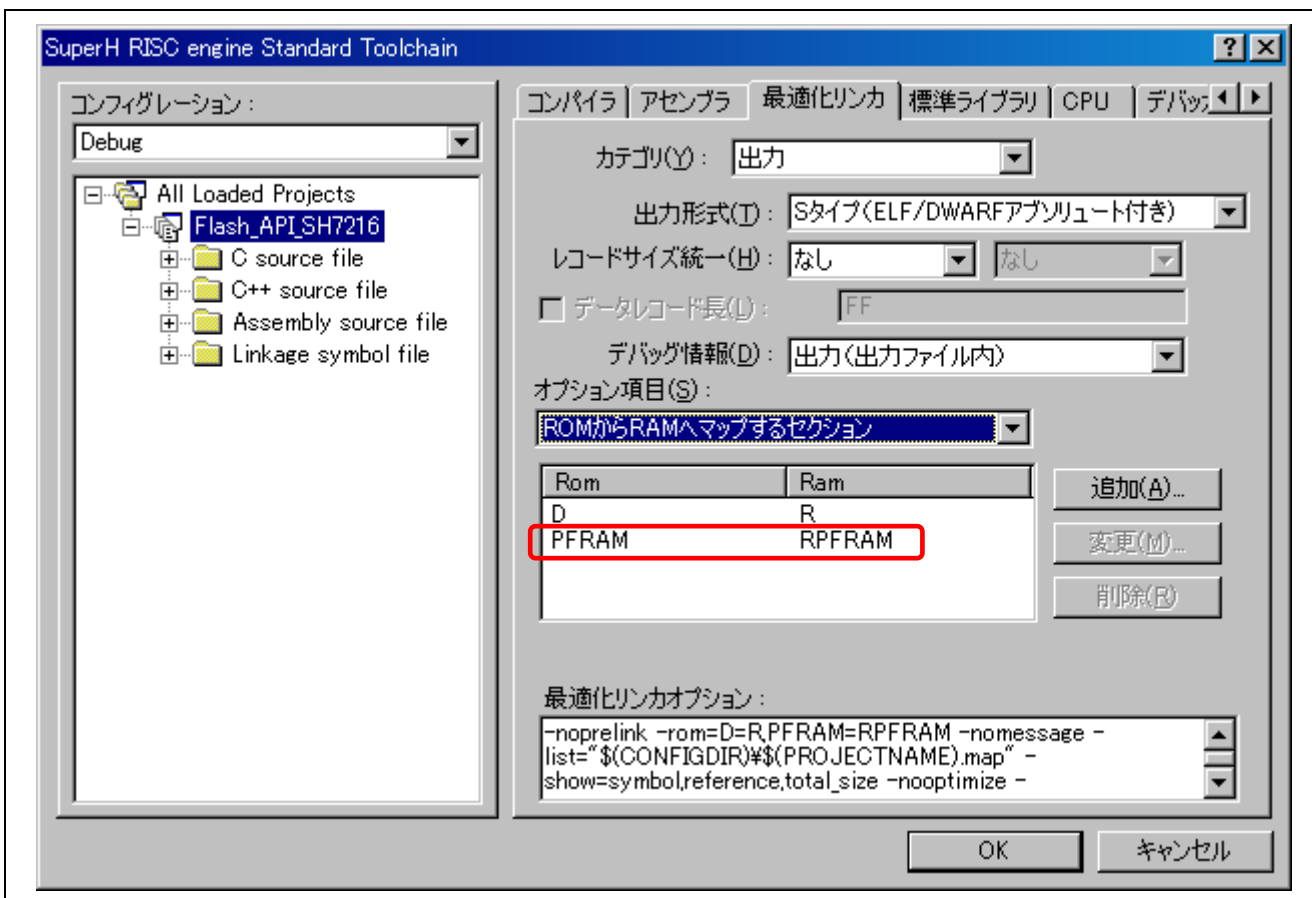


図3 High-performance Embedded Workshop のROM化支援オプション設定例

4. RAM への全アプリケーション転送

マイコンデバイスの全フラッシュメモリ領域に対し消去／書き込みを行うブートローダを開発する場合、実行中のプログラムを含むフラッシュメモリのブロックは消去してはいけないため、このようなブートローダ用のユーザアプリケーションは全て RAM に転送しておく必要があります。以下に、この転送処理を行うための手順および留意事項を示します。

- High-performance Embedded Workshop のリンカ設定にて、ユーザプログラムを実行するためのセクションを RAM 上に割り当てます。このとき、「固定 (Fixed)」領域のセクションであることを明示するため、セクション名の頭文字に「F」を付与しておきます。例えば、MY_APP_RAM というセクション名にする場合、リンカ設定には「FMY_APP_RAM」という名前登録します。
- RAM 上でコードを実行する際、先にダウンロード元および格納元の領域が ROM 上に必要なため、High-performance Embedded Workshop のリンカ設定にてセクションを配置します。このとき、ツールチェインで「プログラム」領域であることを指す「P」を、セクション名の頭文字に付与しておきます。例えば、MY_APP というセクション名にする場合、リンカ設定には「PMY_APP」という名前登録します。
- SH リンカには、ROM 上の関数（およびデータ）のメモリアドレスを RAM に配置するためのオプションがあります。ただし、このオプション設定のみでは実際には RAM に配置されず、物理的には ROM 上に配置されたままになっています。このオプション設定を使用して RAM に配置するためには、関数（およびデータ）が参照される前に、あらかじめユーザアプリケーションによって、当該コードまたはデータ領域を ROM の格納場所から RAM の実行場所へコピーする必要があります。この転送処理の後、オプション設定により、参照される全ての絶対アドレスが RAM 上に配置されます。もし他のソースモジュールがこのコードを参照した場合、参照先は格納元の ROM アドレスではなく、常に RAM アドレスになります。このオプション設定には、リンカの ROM から RAM へのマッピングオプション「-rom=xxxx=yyyy」を使用します。ここで、「xxxx」には ROM に配置したセクション名が入り、「yyyy」には RAM に配置したセクション名が入ります。今回の例では、以下のように表されます：

```
-rom=PMY_APP=FMY_APP_RAM
```

- 図 4 に、ユーザアプリケーション上で対象領域を ROM から RAM へコピーするためのプログラムコード記述例を示します。このコピー処理は、RAM に配置する対象のコードを実行する前に行います。

```
unsigned char *src;
unsigned char *dst;

src = (unsigned char *)__sectop("PMY_APP");
dst = (unsigned char *)__sectop("FMY_APP_RAM");
for( ; src < (unsigned char *)__secend("PMY_APP"); src++, dst++){
    *dst = *src;
}
```

図4 ROM から RAM へコピーするためのプログラムコード記述例

- RAMに配置するプログラムコードを記述する場合、リンカが、ROMセクションの中でどれがRAMに配置する関数なのか判別できるようにセクションを独自に定義する方法があります。セクションを独自に定義するためには、関数の前に「#pragma section MY_APP」と記述します。ここで、セクション名の「MY_APP」の前に頭文字の「P」は付与しないことに注意してください。コンパイラによってプログラム領域と判断されると、セクション名の先頭に「P」が自動的に付与されます。また、ソースファイル内で「#pragma section」指令を使用してセクション名を新たに定義すると、その定義した行からファイルの最後までに宣言された関数やデータは、その新たに定義したセクションに全て配置されることに注意してください。詳細は、SHコンパイラパッケージのユーザーズマニュアルを参照してください。図5に、使用例を示します。

```
#pragma section MY_APP
void function1( void )
{
    {この関数は「PMY_APP」セクションに配置されます}
}
void function2( void )
{
    {この関数は「PMY_APP」セクションに配置されます}
}
```

図5 「#pragma section」指令によるセクション配置例

- 最終的な確認事項として、消去／書き込みを行うための全ての関数が、ROMからRAMに配置する対象セクションに含まれていることをマップファイルなどで確認してください。もし、それらの関数がROM上に配置されていた場合、ユーザプログラムでそれらの関数で消去することになりますので、正常に動作しません。
- また、コンパイラの最適化によって、コードの処理効率を高めるために、実行時ルーチンなどの標準ライブラリ関数がユーザプログラム内から呼び出される場合があります。これは、繰り返し使用する機能をそのまま重複して実装するよりも、その機能を実装したライブラリ関数を1度だけ呼び出した方が、全体的にコードサイズを小さくできるためです。一例として、32ビットの乗算命令をユーザアプリケーションプログラム上で実行する場合があります。ただし、コンパイラが生成する出力結果を確認しない限り標準ライブラリ関数が使われているか見分けられない場合もあります。標準ライブラリ関数は、デフォルトのセクション配置ではROM上に配置されており、ユーザアプリケーションプログラムでRAMに再配置する対象セクションではありません。そのため、ユーザのフラッシュ書き換えプログラムは、いくつかのブロックを消去する際には問題なく実行できますが、ライブラリ関数が配置されたブロックを消去した場合には、ユーザアプリケーションプログラムが正常に動作しなくなりますので注意してください。

5. API 関数仕様

5.1 R_FlashErase

本関数は、フラッシュの指定ブロック全体を消去します。

書式

```
uint8_t R_FlashErase( uint8_t block );
```

引数

block

消去するブロックを指定します。指定する値は `Flash_API_SHxxxx.h` ファイル内で定義しています。ブロックの定義名は、マイコンデバイスのハードウェアマニュアルの記載と同じ内容になっています。例えば、アドレス `H'00000000` のブロックはハードウェアマニュアルではブロック `0` になっており、このブロックを消去対象とする場合は引数に「`BLOCK_0`」を指定します。

戻り値

消去処理の結果を返します。

0 : 正常終了。

0 以外 : エラー発生。エラーコードはマイコンデバイスにより異なります。詳細は、C ソース内の関数ヘッダを参照してください。

実装ファイル

「`Flash_API_SHxxxx.h`」ファイル内でプロトタイプ宣言しています。

「`Flash_API_SHxxxx.c`」ファイル内で実装しています。

解説

フラッシュメモリ内の単一のブロックを消去します。

【注】 実行中のプログラムを含むフラッシュメモリのブロックは消去しないでください。

5.2 R_FlashWrite

本関数は、フラッシュヘータを書き込みます。

書式

```
uint8_t R_FlashWrite( uint32_t   flash_addr,  
                     uint32_t   buffer_addr,  
                     uint16_t   bytes );
```

引数

flash_addr

フラッシュ書き込み対象領域のアドレスを指定します。指定するアドレスは、書き込みを行う対応領域のライン境界（128 または 256 バイト境界）としてください。本引数には重要な制限事項があります。詳細は以下の解説を参照してください。

buffer_addr

フラッシュに書き込むデータバッファのポインタを指定します。指定するアドレスは RAM 領域としてください。

bytes

フラッシュに書き込むデータバッファ `buffer_addr` のバイト数を指定します。指定するバイト数は 128 または 256 の倍数（マイコンにより異なります）にしてください。本引数には重要な制限事項があります。詳細は以下の解説を参照してください。

戻り値

書き込み処理の結果を返します。

0 : 正常終了。

0 以外 : エラー発生。エラーコードはマイコンデバイスにより異なります。詳細は、C ソース内の関数ヘッダを参照してください。

実装ファイル

「Flash_API_SHxxxx.h」ファイル内でプロトタイプ宣言しています。

「Flash_API_SHxxxx.c」ファイル内で実装しています。

解説

フラッシュメモリにデータを書き込みます。

SH-2 および SH-2A マイコンでは、128 または 256 バイトの倍数（マイコンにより異なります）のみにて書き込むことができます。そのため、データバッファには、マイコンの書き込み「ライン」サイズ（128 または 256）の倍数を指定する必要があります。さらに、フラッシュ書き込み対象の先頭アドレスは、ライン境界にそろえる必要があります。例えば、書き込みラインサイズが 128 の場合、指定するフラッシュアドレスのビット 6～ビット 0 は全て「0」でなければなりません。一方、ラインサイズが 256 の場合は、フラッシュアドレスのビット 7～ビット 0 は全て「0」でなければなりません。

SH7216 グループ/SH7239 グループ/SH7231 グループは、プログラム格納用のフラッシュ領域とは別に、データ格納用のフラッシュ(データフラッシュ:FLD)領域を搭載しています。関連する API として、SH7216 グループ/SH7239 グループ/SH7231 グループでは後述の「R_FlashDataAreaAccess」および「R_FlashDataAreaBlankCheck」も参照してください。

表 2に、マイコンデバイスグループ別の書き込みラインサイズを示します。

表2 書き込みラインサイズ

デバイスグループ	書き込みラインサイズ
SH7125 グループ/SH7146 グループ/SH7137 グループ/ SH7080 グループ	128 バイト
SH7211 グループ/SH7280 グループ	256 バイト
SH7216 グループ/SH7239 グループ/SH7231 グループ (プログラム領域)	256 バイト
SH7216 グループ/SH7239 グループ/SH7231 グループ (データ領域)	8 バイトまたは 128 バイト

5.3 R_FlashDataAreaAccess (SH7216 グループ/SH7239 グループ/SH7231 グループのみ)

本関数は、データフラッシュ領域へのアクセス（読み出し/書き換え）を許可/禁止します。

書式

```
void R_FlashDataAreaAccess( uint16_t    read_en_mask,  
                             uint16_t    write_en_mask );
```

引数

read_en_mask

ビット 15~0 のビットフィールド（ただし、SH7216 グループ/SH7231 グループはビット 3~0 のみ）を使用して、それぞれのデータブロックに対し、CPU からの読み出し許可/禁止を設定します。ビット 0~15 は、それぞれデータブロック 0~15 を意味します（ただし、SH7216 グループ/SH7231 グループではデータブロック 0~3 まで）。各ビット値は、「0」に設定すると当該ブロックの読み出し禁止、「1」に設定すると当該ブロックの読み出し許可となります。

write_en_mask

ビット 15~0 のビットフィールド（ただし、SH7216 グループ/SH7231 グループはビット 3~0 のみ）を使用して、それぞれのデータブロックに対し、フラッシュ制御ユニット（FCU）からの書き換え（消去/書き込み）許可/禁止を設定します。ビット 0~15 は、それぞれデータブロック 0~15 を意味します（ただし、SH7216 グループ/SH7231 グループではデータブロック 0~3 まで）。各ビット値は、「0」に設定すると当該ブロックの書き換え禁止、「1」に設定すると当該ブロックの書き換え許可となります。

戻り値

なし。

実装ファイル

「Flash_API_SHxxxx.h」ファイル内でプロトタイプ宣言しています。

「Flash_API_SHxxxx.c」ファイル内で実装しています。

解説

リセット後のデータフラッシュ領域は、CPU からアクセス（読み出し/書き換え）禁止の状態になっています。本関数は、データブロック 0~15 に対するアクセス（読み出し/書き換え）を許可/禁止に設定できます（ただし、SH7216 グループ/SH7231 グループではデータブロック 0~3 まで）。また、本関数は、ユーザアプリケーションの先頭箇所で 1 度実行するだけで問題ありません。

【注】 本関数の適用対象は、SH7216 グループ/SH7239 グループ/SH7231 グループのデータフラッシュ（FLD）のみです。

5.4 R_FlashDataAreaBlankCheck (SH7216 グループ/SH7239 グループ/SH7231 グループのみ)

データフラッシュ領域が消去状態（ブランク）であるかどうかは、単純にメモリの指定場所を読み出すだけでは確認できないため、本関数を使用して指定した領域のブランクチェックを行います。

書式

```
uint8_t R_FlashDataAreaBlankCheck( uint32_t  address,  
                                   uint8_t   size );
```

引数

address

ブランクチェックを行う対象領域のアドレスを指定します。

第2引数 (size) に「BLANK_CHECK_8_BYTE」を指定した場合、本引数には8バイト境界のアドレスを指定してください。

第2引数 (size) に「BLANK_CHECK_ENTIRE_BLOCK」を指定した場合、本引数にはデータブロック番号定義値 (SH7216 グループ/SH7231 グループの場合: 「BLOCK_DB0」～「BLOCK_DB3」、SH7239 グループの場合: 「BLOCK_DB0」～「BLOCK_DB15」) を指定してください。

size

本引数には、ブランクチェックの対象領域サイズを8バイトとするか、または8Kバイトのブロック領域全体とするかを指定します。8バイトの場合は「BLANK_CHECK_8_BYTE」を、8Kバイトのブロック領域全体の場合は「BLANK_CHECK_ENTIRE_BLOCK」を、どちらか一方のみ指定してください。

戻り値

ブランクチェック処理の結果を返します。

H'00: 指定領域は消去状態（ブランク）。

H'01: 指定領域は消去状態ではない（ブランクではない）。

H'02: 処理中にエラーが発生し、ブランクチェックは正常終了しなかった。

実装ファイル

「Flash_API_SHxxxx.h」ファイル内でプロトタイプ宣言しています。

「Flash_API_SHxxxx.c」ファイル内で実装しています。

解説

マイコンの内蔵フラッシュ領域に対し書き込みを行う場合、対象領域はあらかじめ消去した状態（ブランク）である必要があります。SH7216 グループ/SH7239 グループ/SH7231 グループのデータフラッシュ領域では、ユーザプログラム領域とは異なり、データ値が H'FF であっても、それが「ブランク」であることを意味するわけではありません。データフラッシュ領域がブランクであるかどうかは、本関数を使用して確認する必要があります。

SH7216 グループ／SH7239 グループ／SH7231 グループでは、対象領域サイズ別に 2 種類の方法でブランクチェックを実施できます。1 つは 8 バイト単位で確認する方法（データフラッシュ領域への書き込み単位が 8 バイトのため）、もう 1 つはデータフラッシュの 1 ブロック全体を 1 度に確認する方法です。本関数は、対象領域ごとに書き込みを行う場合でも、それぞれの書き込み処理前に毎回呼び出す必要はありません。書き込み用のアプリケーションに対し、1 度実行するだけで問題ありません。

【注】 本関数の適用対象は、SH7216 グループ／SH7239 グループ／SH7231 グループのデータフラッシュ（FLD）のみです。

6. API 使用例

図 6に、SH7137 グループのブロック 2 に対し、消去／書き込みを行う場合のAPI使用例を示します。

```
#include "Flash_API_SH7137.h"

uint8_t write_buffer[256];    // 128 バイト境界から 2 ライン分のデータ領域を保持
void main( void )
{
    uint8_t result;

    /* ブロック 2 を消去 */
    result = R_FlashErase( BLOCK_2 );

    /* ブロック 2 に対しユーザデータバッファを書き込む。
       書き込みは 128 バイト境界から 2 ライン分実施。 */
    result = R_FlashWrite( (uint32_t)0x2000,           // 書き込み先
                           (uint32_t)write_buffer,    // ユーザ書き込みデータ
                           256 );                    // 書き込みサイズ

    while(1);    /* End of DEMO */
}
```

図6 SH7137 グループにおける API 使用例

図 7に、SH7216 グループのデータブロック 0 に対し、消去／書き込みを行う場合のAPI使用例を示します。

```
#include "Flash_API_SH7216.h"

uint8_t write_buffer[8];    // 書き込みデータ格納先
void main( void )
{
    uint8_t result;

    /* 全てのデータフラッシュブロックに対し読み出し／書き込みを許可 */
    R_FlashDataAreaAccess( 0xf, 0xf );

    /* 8 バイト領域に対しブランクチェック */
    R_FlashDataAreaBlankCheck( (uint32_t)0x80100400,
                                BLANK_CHECK_8_BYTE );

    /* データブロック 0 を消去 */
    result = R_FlashErase( BLOCK_DB0 );

    /* フラッシュデータブロック 0 に対しユーザデータバッファを書き込む。 */
    result = R_FlashWrite( (uint32_t)0x80100000,       // 書き込み先
                           (uint32_t)write_buffer,    // ユーザ書き込みデータ
                           8 );                      // 書き込みサイズ

    while(1);    /* End of DEMO */
}
```

図7 SH7216 グループにおける API 使用例

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

ユーザーズマニュアル：ハードウェア

SH7125 グループ、SH7124 グループ ハードウェアマニュアル (RJJ09B0249)

SH7146 グループ ユーザーズマニュアル ハードウェア編 (R01UH0049JJ)

SH7137 グループ ハードウェアマニュアル (RJJ09B0392)

SH7080 グループ ユーザーズマニュアル ハードウェア編 (R01UH0198JJ)

SH7211 グループ ハードウェアマニュアル (RJJ09B0338)

SH7280 グループ、SH7243 グループ ユーザーズマニュアル ハードウェア編 (R01UH0229JJ)

SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編 (R01UH0230JJ)

SH7239 グループ、SH7237 グループ ユーザーズマニュアル ハードウェア編 (R01UH0086JJ)

SH7231 グループ ユーザーズマニュアル ハードウェア編 (R01UH0073JJ)

(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

フラッシュメモリの書き込み/消去中の割り込み処理について (TN-SH7-A828A/J)

(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。)

ユーザーズマニュアル：開発環境

SuperH C/C++コンパイラパッケージ V.9.04 ユーザーズマニュアル (R20UT0704JJ)

SuperH ファミリ用 E10A-USB エミュレータ制限事項 (HS0005KCU01H/HS0005KCU02H 用) (R20UT0913JJ)

(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。)

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/contact/>

改訂記録	SH ファミリ アプリケーションノート SH-2, SH-2A 用シンプル フラッシュ API
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.08.06	—	初版発行
1.01	2010.09.24	—	対象デバイスに SH7125 グループ、SH7211 グループ、および SH7239 グループを追加
1.02	2011.05.19	—	対象デバイスに SH7080 グループ、SH7146 グループ、および SH7231 グループを追加
1.03	2011.06.30	—	SH7216 グループ／SH7239 グループ／SH7231 グループ用の API コー ドを修正
1.04	2012.06.08	1	目次修正
		5	注意／制限事項追記
		17	参考ドキュメント追記
		—	本文修正
		—	SH7216 グループ／SH7239 グループ／SH7231 グループ用の API コー ドを修正

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・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因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に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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て、当社は、一切その責任を負いません。
9. 本資料に記載されている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・システムに使用することはできません。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
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禁じます。

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/contact/>