

---

## SH7753 グループ

R01AN1621JJ0100

Rev.1.00

## E10A-USB を用いたシリアルフラッシュの書き換えプログラム例

---

2013.7.1

### 要旨

本アプリケーションノートは SPI0 インタフェースを使用したシリアルフラッシュの書き換えプログラム (以下、FMTOOL) について説明します。

### 対象デバイス

SH7753グループ

## 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. ソフトウェア説明.....	5
4.1 動作概要.....	5
4.2 HEWバッチファイル(fmttool.hdc).....	6
4.2.1 ファイル構成.....	6
4.2.2 パラメータの設定.....	6
4.3 シリアルフラッシュ書き換えプログラム .....	7
4.3.1 ファイル構成.....	7
4.3.2 セクション配置 .....	8
4.3.3 関数一覧.....	9
4.3.4 関数仕様.....	10
main.....	11
SPI0_Init .....	13
SPI0_BlkEraseFlash .....	14
SPI0_ReadFlash .....	15
SPI0_WriteFlash .....	16
4.3.5 必要メモリサイズ.....	17
5. FMTTOOLの使用手順.....	18
6. サンプルコード .....	19
7. 参考ドキュメント.....	19

## 1. 仕様

FMTOOL は、シリアルフラッシュ書き換えプログラムと HEW のバッチファイル(以下、fntool.hdc)で構成されています。

表 1.1に開発環境一覧を、図 1.1に FMTOOL の実行環境を示します。

表1.1 環境

開発環境	用途
開発用 PC (Windows 7)	FMTOOL を実行するために使用します。
エミュレータ (E10A-USB)	
統合開発環境 (HEW Version 4.09.01.007)	
C コンパイラ (SHC V.9.04 release01)	
評価ボード (SH7753 評価ボード)	

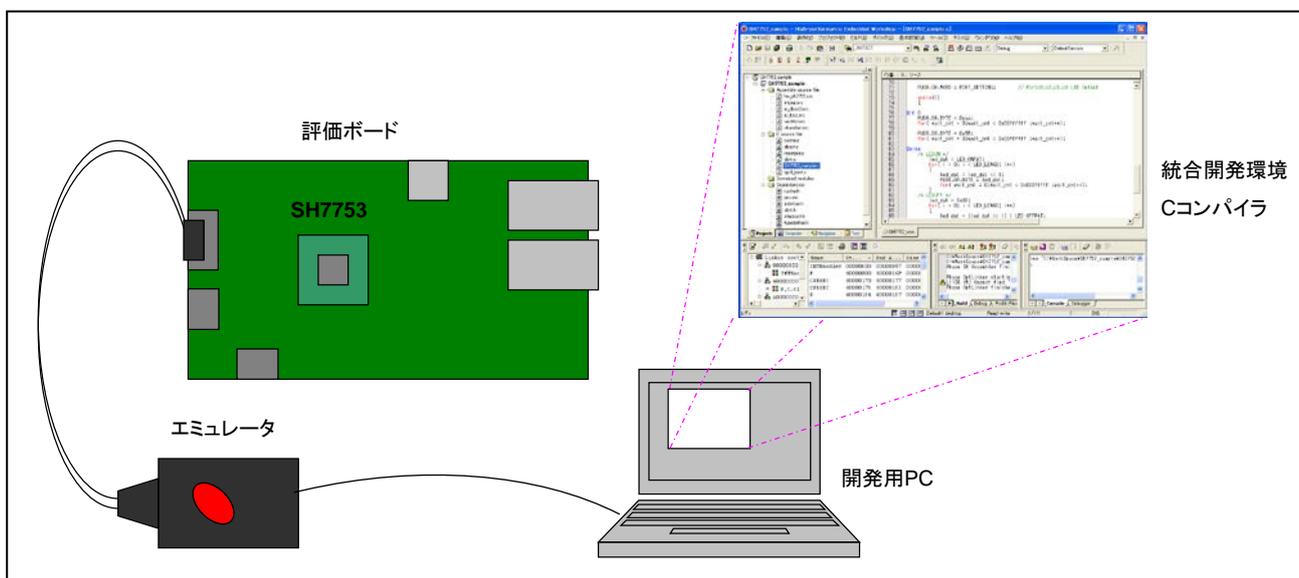


図1.1 FMTOOL の実行環境

## 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	SH-4A/SH7753 (R5S7753)
CPU 周波数	[ クロック動作モード1 ] - CPUクロック : 最大576MHz - ローカルバスクロック : 最大48 MHz - DDR3-SDRAMクロック : 最大 528 MHz - 周辺クロック : 最大48 MHz
統合開発環境	High-Performance Embedded Workshop- Ver 4.09.01.007
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for SuperH Family (V.9.04 release01)
エンディアン	リトルエンディアン
ブートモード	SPI0 ブートモード
アドレス拡張モード	32 ビット
メモリマネジメントユニット (MMU)	ディスエーブル
ウォッチドッグタイマ(WDT)	WDT0/WDT1 : 停止 ABRT:ABR 無効
外部メモリ	DDR3-SDRAM、4Gbit (16 ビットバス幅) - MT41J256M16RE-15E:D / Micron 社製
外部 SPI フラッシュメモリ	- MX25L64 : 8MB SPI フラッシュ (Macronix 社製) - S25FL512S : 64MB SPI フラッシュ (Spansion 社製)
アドレスモード	- 3 バイトアドレスモード : BOOTSZ[2 :0] = 8 MB (MX25L64 使用時) - 4 バイトアドレスモード : BOOTSZ[2 :0] = 64 MB (S25FL512S 使用時)
使用ボード	SH7753 グループ EVB ボード

## 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。

- ・ SH7753 グループ アプリケーションノート  
初期設定例と E10A-USB 使用時のソフトウェアダウンロード方法

## 4. ソフトウェア説明

### 4.1 動作概要

fmtool.hdc にシリアルフラッシュのアドレスやユーザプログラムを設定し、このバッチファイルを実行することで、シリアルフラッシュ書き換えプログラムが動作し、シリアルフラッシュの書き換えを行います。

以下に、FMTOOL の実行手順を示します。

- A) fmtool.hdc をテキストエディタなどで開き、パラメータを設定する
- B) シリアルフラッシュ書き換えプログラムを起動する
- C) HEW のスクリプトウィンドウから fmtool.hdc を実行する
- D) fmtool.hdc を実行すると、シリアルフラッシュ書き換えプログラムとユーザプログラムが SH7753 のメモリ空間へダウンロードされる
- E) シリアルフラッシュ書き換えプログラムが実行され、以下の手順でシリアルフラッシュにユーザプログラムが書き込まれる。
  - ・シリアルフラッシュのイレーズ
  - ・シリアルフラッシュへのユーザプログラムの書き込み
  - ・書き込みデータのベリファイ

以下に FMTOOL の処理手順を示します。

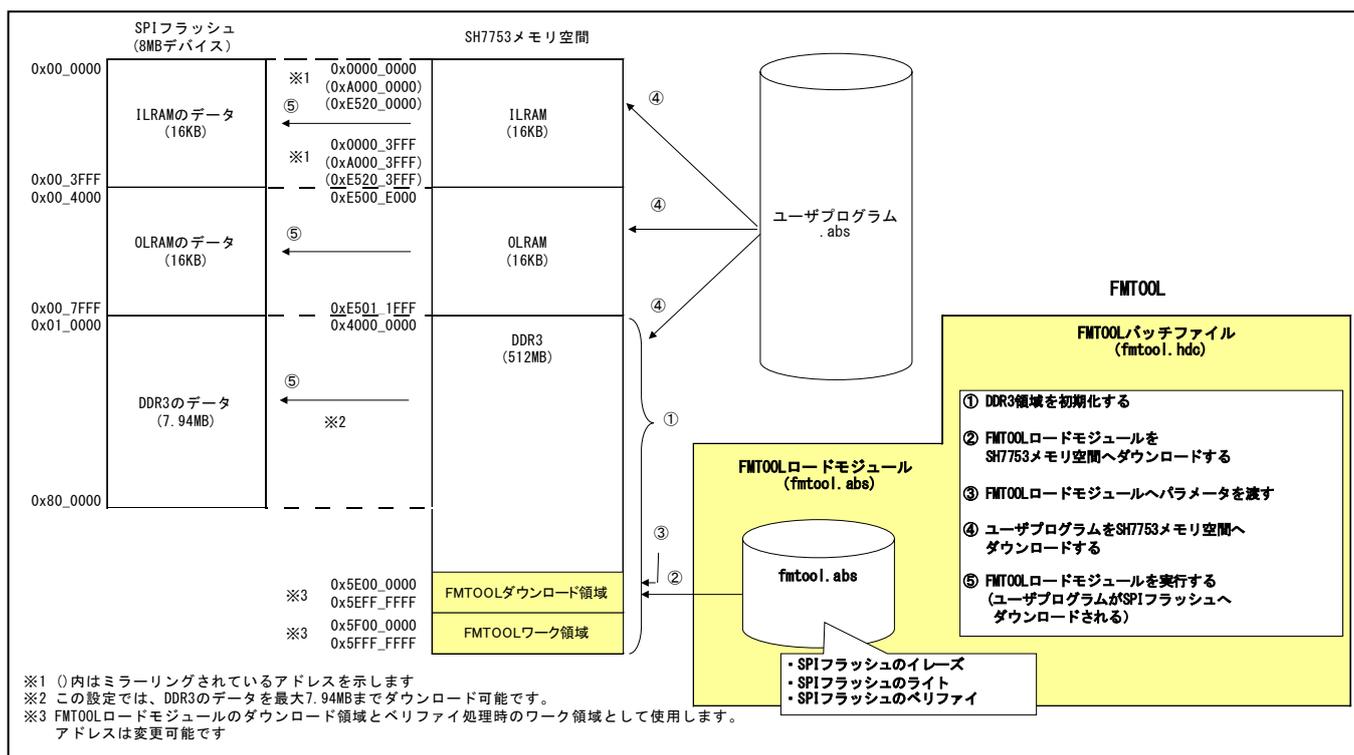


図 4.1 FMTOOL の処理手順

## 4.2 HEW バッチファイル(fmtool.hdc)

### 4.2.1 ファイル構成

表 4.1に HEW バッチファイルのファイル構成を示します。

表 4.1 HEW バッチファイルのファイル構成

ファイル名	概要
fmtool.hdc	シリアルフラッシュ書き換えプログラムとユーザプログラムを SH7753 のメモリ空間へダウンロードし、シリアルフラッシュ書き換えプログラムを実行する HEW バッチファイルです。

### 4.2.2 パラメータの設定

図 4.2に fmtool.hdc のパラメータ設定を示します。このパラメータをシリアルフラッシュ書き換えプログラムが参照し、シリアルフラッシュへプログラムを書き込みます。

fmtool.hdc		
ユーザ定義パラメータ		
! -----		
! - User Define Parameter		
! - Please set following parameters.		
! -----		
! [User's Program File]	①ユーザプログラムへのパスを指定します	
set User_Program	"C:¥¥workspace¥¥Initialize_Base¥¥Initialize_Base¥¥Debug¥¥Initialize_Base.abs"	
! [Address of User's Program]	} ②ユーザプログラムが使用する領域のアドレスを指定します。(サンプルコードは <b>ILRAM</b> 、 <b>OLRAM</b> 、 <b>DDR3#1</b> 、 <b>DDR3#2</b> の 4 つ領域を指定しています)	
set SH7753_Addr_B0		0x00000800
set SH7753_Addr_B1		0xE500E000
set SH7753_Addr_B2		0x40000000
set SH7753_Addr_B3	0x41000000	
! [Address of Work Area]	} ③ワーク領域のアドレスを指定します。 ワーク領域はユーザプログラムと SPI フラッシュの内容をベリファイするために使用します。 ④で指定するサイズ以上の領域を確保してください。	
set SH7753_Addr_WORK		0x5F000000
! [Size of User's Program]	} ④ユーザプログラムが使用する領域のサイズを指定します。サイズは、余裕を持って指定することが可能ですが、あまり大きなサイズを指定するとダウンロード時間が長くなります。 使用しない領域のサイズは 0 としてください	
set Write_Size_B0		0x00000C15
set Write_Size_B1		0x00000000
set Write_Size_B2		0x0000037F
set Write_Size_B3	0x00018000	
! [Address of SPI Flash]	} ⑤ユーザプログラムをダウンロードする SPI フラッシュのアドレスを指定します	
set SPI0_Addr_B0		0x00000800
set SPI0_Addr_B1		0x00004000
set SPI0_Addr_B2		0x00010000
set SPI0_Addr_B3	0x00100000	

図 4.2 FMTOOL バッチファイルの設定箇所

### 4.3 シリアルフラッシュ書き換えプログラム

#### 4.3.1 ファイル構成

表 4.3にシリアルフラッシュ書き換えプログラムで使用するファイルを示します。

表 4.3 シリアルフラッシュ書き換えプログラムで使用するファイル

ファイル名	概要
intprg.src vecttbl.src dbsct.c sbrk.c env.inc vect.inc iodefine.h sbrk.h stacksct.h typedefine.h	統合開発環境で自動生成されるファイルです。
vhandler.src	統合開発環境で自動生成されるファイルです。 サンプルコードは SH7753 レジスタの初期化関数と DBSC3 の初期化関数をコールするように変更しています。
resetprg.c	統合開発環境で自動生成されるファイルです。 サンプルコードは SR レジスタへ設定する値を変更しています。
io_SH7753.src	SH7753 のレジスタ初期設定が定義されたファイルです。
io_dbsc3.src	DBSC3 の初期設定が定義されたファイルです。
fmttool.c	main()が定義されたファイルです。 SPI フラッシュの初期化、イレーズ、ライト、ベリファイを実行します。 SPI フラッシュヘダウンロードするパラメータが定義されています。
spi0.c spi0.h	SPI フラッシュの初期化、イレーズ、ライト、リード関数が定義されたファイルです。

## 4.3.2 セクション配置

表 4.4にシリアルフラッシュ書き換えプログラムのセクション配置を示します。

表 4.4 シリアルフラッシュ書き換えプログラムのセクション配置

セクション名	セクション用途	配置アドレス（仮想アドレス）	
P	プログラム領域（指定なしの場合）	H'5E000000	P0 領域（DDR3）  キャッシング可能、 MMU アドレス変換可能  ※ユーザプログラムが使用する領域と重なる場合は、配置アドレスを変更してください。
C	定数領域		
C\$BSEC	未初期化データ領域用アドレス構造体		
C\$DSEC	初期化データ領域用アドレス構造体		
D	初期化データ（初期値）		
B	未初期化データ領域		
R	初期化データ領域		
RSTHandler	リセットハンドラ	H'A0000000	P2 領域（ILRAM）  キャッシング不可、 MMU アドレス変換不可  ※FMTOOLは起動時にのみこの領域を使用しますので、ユーザプログラムをこの領域にダウンロードしても問題ありません。
INIT_HW	H/W 初期化関数		
INIT_DBSC	DDR3 初期化関数		
INTHandler	例外/割り込みハンドラ		
VECTTBL	リセットベクタテーブル		
INTTBL	割り込みベクタテーブル 割り込みマスクテーブル		
IntPRG	割り込み関数		
PRResetPRG	リセットプログラム		

## 4.3.3 関数一覧

表 4.5にシリアルフラッシュ書き換えプログラムの主な関数を示します。

表 4.5 シリアルフラッシュ書き換えプログラムの主な関数

分類	関数名	機能概要	ファイル名	スタック サイズ
メイン関数	main	下記の初期化、イレーズ、リード、ライト関数を使用し、SH7753 メモリ空間に格納されたユーザプログラムを SPI フラッシュへダウンロードする。	fmttool.c	24 バイト
初期化関数	SPI0_Init	SPI0 のレジスタ、FIFO などをクリアする。	spi0.c	4 バイト
イレーズ関数	SPI0_BlkEraseFlash	シリアルフラッシュをバルクイレーズする。		16 バイト
リード関数	SPI0_ReadFlash	シリアルフラッシュをリードする。		32 バイト
ライト関数	SPI0_WriteFlash	シリアルフラッシュをプログラムする。		40 バイト

#### 4.3.4 関数仕様

シリアルフラッシュ書き換えプログラムの関数仕様を示します。関数仕様の読み方は以下のとおりです。

関数名		分類
機能概要		
書式	関数の呼び出し形式を示します。#include “ヘッダファイル”で示すヘッダファイルは、この関数の実行に必要な標準ヘッダファイルであるため、必ずインクルードしてください。	
引数	I/O は、引数がそれぞれ入力データ、出力データであることを意味します。	
戻り値	関数の戻り値を示します。	
解説	関数の仕様について説明します。	
注意事項	注意事項があればここに示します。	

## main

メイン関数

## FMTOOL のメイン処理

書式

```
#include "iodefine.h"
#include "spi0.h"
void main(void);
```

引数

なし

戻り値

なし

解説

本関数は、SPI0 の初期化関数、イレーズ関数、ライト関数、リード関数をコールして、SH7753 メモリ空間に格納されたユーザプログラムをシリアルフラッシュに書き込みます。また、書き込みが完了後、シリアルフラッシュの内容を DDR3 上のワーク領域に読み出し、ベリファイを行います。

シリアルフラッシュの書き換えに必要な情報は fmtreeol.hdc に設定したパラメータを参照します。Fmtreeol.hdc のパラメータを変更する方法は、4.2.2章を参照してください。

以下に、本関数が参照するパラメータを示します。

## 【パラメータ変数 1/2】

型	変数名	初期値	内容
unsigned long	B0_ADDR_SH7753	0x00000800	ユーザプログラムが使用する領域#0 のアドレス
unsigned long	B1_ADDR_SH7753	0xE500E000	ユーザプログラムが使用する領域#1 のアドレス
unsigned long	B2_ADDR_SH7753	0x40000000	ユーザプログラムが使用する領域#2 のアドレス
unsigned long	B3_ADDR_SH7753	0x41000000	ユーザプログラムが使用する領域#3 のアドレス
unsigned long	B0_SIZE	0x00000000	ユーザプログラムが使用する領域#0 のサイズ
unsigned long	B1_SIZE	0x00000000	ユーザプログラムが使用する領域#1 のサイズ
unsigned long	B2_SIZE	0x00000000	ユーザプログラムが使用する領域#2 のサイズ
unsigned long	B3_SIZE	0x00000000	ユーザプログラムが使用する領域#3 のサイズ

## 【パラメータ変数 2/2】

型	変数名	初期値	内容
unsigned long	B0_ADDR_SPI0	0x00000800	ユーザプログラムが使用する領域#0 をダウンロードする SPI フラッシュのアドレス
unsigned long	B1_ADDR_SPI0	0x00004000	ユーザプログラムが使用する領域#1 をダウンロードする SPI フラッシュのアドレス
unsigned long	B2_ADDR_SPI0	0x00010000	ユーザプログラムが使用する領域#2 をダウンロードする SPI フラッシュのアドレス
unsigned long	B3_ADDR_SPI0	0x00010000	ユーザプログラムが使用する領域#3 をダウンロードする SPI フラッシュのアドレス
unsigned long	WK_ADDR_SH7753	0x5F000000	ワーク領域のアドレス

**注意事項** 本関数は、SPI0 の初期化、イレーズ、ライト、リードの進行状況と、ベリファイ結果を GPIO のポート U に出力します。SH7753 評価ボードはポート U に LED[3:9] が接続されているため、LED[3:9] の点灯パターンからステータス情報を確認できます。

LED[3:9] の点灯パターンは以下のとおりです。

## 【ステータス情報】 LED[3:9] が示す内容 (○ : 点灯 - : 消灯)

LED3	LED4	LED5	LED6	LED7	LED8	LED9	内容
-	-	-	-	-	-	-	初期化中 または、FMTOOL が実行されていない。
-	-	-	-	-	-	○	イレーズ中 ※1~2分かかります
-	-	-	-	-	○	-	ライト中
-	-	-	-	○	-	-	ベリファイ中
○	○	○	○	○	○	○	ベリファイ成功。正常終了
○	-	○	-	○	-	○	ベリファイ失敗。異常終了

シリアルフラッシュの書き換え結果は HEW のソースウィンドウで確認することもできます。確認方法は「5.FMTOOL の使用手順」を参照してください。

---

## SPI0\_Init

初期化関数

SPI0 の初期化処理

---

書式

```
#include "iodefine.h"
#include "spi0.h"
void SPI0_Init(void);
```

引数

なし

戻り値

なし

解説

本関数は、SPI0 のレジスタを初期化し、FIFO をリセットします。

注意事項

SPI0 のイレーズ、ライト、リード関数をコールする前に、必ず本関数を実行してください。

---

## SPI0\_BlkEraseFlash

イレーズ関数

SPI0 のイレーズ処理

---

書式

```
#include "iodefine.h"
#include "spi0.h"
void SPI0_BlkEraseFlash(char ss);
```

引数

char ss	I	SPI0 フラッシュのチップセレクト
		0: SP0-SS0 端子を選択
		1: SP0-SS1 端子を選択
		2: SP0-SS2 端子を選択
		3: SP0-SS3 端子を選択

戻り値

なし

解説

本関数は、SPI0 フラッシュのチップセレクトで指定したシリアルフラッシュをイレーズします。

注意事項

SPI0 のライト関数をコールする前に、必ず本関数を実行してください。  
8M バイトのシリアルフラッシュをイレーズする場合、イレーズ時間は1~2分程度必要です。

---

## SPI0\_ReadFlash

リード関数

SPI0 のリード処理

---

書式	<pre>#include "iodefine.h" #include "spi0.h" void SPI0_ReadFlash (char ss, unsigned long addr, unsigned long len, char* buff);</pre>								
引数	<table><tr><td>char ss</td><td>I SPI0 フラッシュのチップセレクト 0: SP0-SS0 端子を選択 1: SP0-SS1 端子を選択 2: SP0-SS2 端子を選択 3: SP0-SS3 端子を選択</td></tr><tr><td>unsigned long addr</td><td>I シリアルフラッシュのリード開始アドレス (シリアルフラッシュのアドレス)</td></tr><tr><td>unsigned long len</td><td>I リードサイズ</td></tr><tr><td>char* buff</td><td>O リードするデータを格納する領域へのポインタ (SH7753 のアドレス)</td></tr></table>	char ss	I SPI0 フラッシュのチップセレクト 0: SP0-SS0 端子を選択 1: SP0-SS1 端子を選択 2: SP0-SS2 端子を選択 3: SP0-SS3 端子を選択	unsigned long addr	I シリアルフラッシュのリード開始アドレス (シリアルフラッシュのアドレス)	unsigned long len	I リードサイズ	char* buff	O リードするデータを格納する領域へのポインタ (SH7753 のアドレス)
char ss	I SPI0 フラッシュのチップセレクト 0: SP0-SS0 端子を選択 1: SP0-SS1 端子を選択 2: SP0-SS2 端子を選択 3: SP0-SS3 端子を選択								
unsigned long addr	I シリアルフラッシュのリード開始アドレス (シリアルフラッシュのアドレス)								
unsigned long len	I リードサイズ								
char* buff	O リードするデータを格納する領域へのポインタ (SH7753 のアドレス)								
戻り値	なし								
解説	本関数は、SPI0 フラッシュのチップセレクトで指定したシリアルフラッシュをリードします。								
注意事項	なし								

## SPI0\_WriteFlash

ライト関数

SPI0 のプログラム処理

書式	<pre>#include "iodefine.h" #include "spi0.h" void SPI0_WriteFlash(char ss, unsigned long addr, unsigned long len, char* buff);</pre>								
引数	<table><tr><td>char ss</td><td>I SPI0 フラッシュのチップセレクト 0: SP0-SS0 端子を選択 1: SP0-SS1 端子を選択 2: SP0-SS2 端子を選択 3: SP0-SS3 端子を選択</td></tr><tr><td>unsigned long addr</td><td>I シリアルフラッシュのプログラム開始アドレス (シリアルフラッシュのアドレス)</td></tr><tr><td>unsigned long len</td><td>I プログラムサイズ</td></tr><tr><td>char* buff</td><td>O プログラムするデータが格納されている領域へのポインタ(SH7753 のアドレス)</td></tr></table>	char ss	I SPI0 フラッシュのチップセレクト 0: SP0-SS0 端子を選択 1: SP0-SS1 端子を選択 2: SP0-SS2 端子を選択 3: SP0-SS3 端子を選択	unsigned long addr	I シリアルフラッシュのプログラム開始アドレス (シリアルフラッシュのアドレス)	unsigned long len	I プログラムサイズ	char* buff	O プログラムするデータが格納されている領域へのポインタ(SH7753 のアドレス)
char ss	I SPI0 フラッシュのチップセレクト 0: SP0-SS0 端子を選択 1: SP0-SS1 端子を選択 2: SP0-SS2 端子を選択 3: SP0-SS3 端子を選択								
unsigned long addr	I シリアルフラッシュのプログラム開始アドレス (シリアルフラッシュのアドレス)								
unsigned long len	I プログラムサイズ								
char* buff	O プログラムするデータが格納されている領域へのポインタ(SH7753 のアドレス)								
戻り値	なし								
解説	本関数は、SPI0 フラッシュのチップセレクトで指定したシリアルフラッシュをプログラムします。								
注意事項	本関数をコールする前に、必ず、SPI0 のイレーズ関数を実行してください。								

#### 4.3.5 必要メモリサイズ

表 4.6にシリアルフラッシュ書き換えプログラムが使用するメモリサイズを示します。

表4.6 シリアルフラッシュ書き換えプログラムが使用するメモリサイズ

使用メモリ	サイズ	備考
ROM	7397 バイト	-
RAM	1604 バイト	-
最大使用ユーザスタック	96 バイト	-
最大使用割り込みスタック	0 バイト	-

【注】 必要メモリサイズはCコンパイラのバージョンやコンパイルオプションにより異なります。

## 5. FMTOOL の使用手順

図 5.1に FMTOOL の使用手順を示します。

■ **fmtool.hdc の編集**

C:\WorkSpace\fmtool\fmtool\fmtool.hdc をエディタで開く。

ユーザプログラムへのパス、ユーザプログラムが使用する領域のアドレス、ユーザプログラムが使用する領域のサイズ、SPI フラッシュのアドレスなどを編集する。

```

13 !
14 ! - User Define Parameter
15 ! - Please set following parameters.
16 !
17 ! [User's Program File]
18 set User_Program "C:\WorkSpace\fmtool\fmtool\fmtool.hdc"
19 ! [Address of User's Program]
20 set SH7753_Addr_B0 0x00000800
21 set SH7753_Addr_B1 0xE500E000
22 set SH7753_Addr_B2 0x40000000
23 set SH7753_Addr_B3 0x41000000
24
25 ! [Address of Work Area]
26 set SH7753_Addr_WORK 0x5F000000
27
28 ! [Size of User's Program]
29 set Write_Size_B0 0x00000C15
30 set Write_Size_B1 0x00000000
31 set Write_Size_B2 0x0000037F
32 set Write_Size_B3 0x00018000
33
34 ! [Address of SPI Flash]
35 set SPI0_Addr_B0 0x00000800
36 set SPI0_Addr_B1 0x00004000
37 set SPI0_Addr_B2 0x00010000
38 set SPI0_Addr_B3 0x00100000
39

```

■ **FMTOOL ロードモジュールのプロジェクト実行**

C:\WorkSpace\fmtool\fmtool.hws をダブルクリックする。

CPU Select ウィンドウにて[SH7753]を選択する。

■ **シリアルフラッシュ書き換えプログラムのビルド**

HEW のメニューより「ビルド→すべてをビルド」を実行する。

■ **fmtool.hdc の実行**

HEW のスクリプトウィンドウで fmtool.hdc を選択し、実行する。



■ **シリアルフラッシュの書き換え結果の確認**

HEW のソースウィンドウでプログラムの停止位置を確認する。

```

56 5E000000 void normal_end ( void )
57 5E000000     while(0);
58 5E000010
59 5E000010
60
61 5E000014 void error_end ( void )
62
63 5E000014     while(0);
64 5E000024
65
66 5E000028 void main(void)

```

図5.1 FMTOOL の使用手順

## 6. 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。

## 7. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

SH7753 グループ ユーザーズマニュアル ハードウェア編

(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

(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 Family 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

(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	E10A-USB を用いたシリアルフラッシュの書き換えプログラム例
------	-----------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.7.01	—	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、本ドキュメントおよびテクニカルアップデートを参照してください。

### 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（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）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（予約領域）がありません。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、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品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、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いません。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用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が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)

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/>