

# SH7268/SH7269 グループ

R01AN0663JJ0101

Rev. 1.01

2012.02.16

## SPI マルチ I/O バスコントローラを使用した シリアルフラッシュメモリからのブート例

### 要旨

SH7268/SH7269 の SPI マルチ I/O バスコントローラ (SPIBSC) は、シリアルフラッシュメモリ上のプログラムデータを直接フェッチして実行する機能 (外部アドレス空間リードモード) を備えています。本アプリケーションノートは、SH7268/SH7269 の外部アドレス空間リードモードを使用したシリアルフラッシュメモリからのブート例について説明します。

### 動作確認デバイス

SH7268/SH7269

以下、総称して「SH7269」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. はじめに.....	2
2. シリアルフラッシュブート機能概要.....	3
3. 応用例の説明.....	8
4. 参考プログラム例.....	26
5. ダウンローダの使用法.....	52
6. 参考ドキュメント.....	57

## 1. はじめに

### 1.1 仕様

SH7269 はブートモード 3 の場合、シリアルフラッシュメモリからブートします(以下、シリアルフラッシュブートとします)。シリアルフラッシュブートはローダプログラムを高速内蔵 RAM に展開し実行しますが、このとき SPI マルチ I/O バスコントローラ (SPIBSC) の外部アドレス空間リードモードは無効になっています。そのためローダプログラムでこのモードを有効にします。

本アプリケーションノートでは、シリアルフラッシュブートを使用する場合の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作成例について説明します。また、ローダプログラムとアプリケーションプログラムをシリアルフラッシュメモリに書き込むためのダウンローダについても説明します。

### 1.2 使用機能

- ブートモード (シリアルフラッシュブート)
- ルネサスシリアルペリフェラルインタフェース (RSPI)
- SPI マルチ I/O バスコントローラ (SPIBSC)

### 1.3 適用条件

マイコン	SH7268/SH7269
動作周波数	CPU 内部クロック (I $\phi$ ) : 266.67 MHz 内部バスクロック (B $\phi$ ) : 133.33 MHz 周辺クロック 1 (P1 $\phi$ ) : 66.67 MHz 周辺クロック 0 (P0 $\phi$ ) : 33.33 MHz
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
コンパイルオプション	High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -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)
シリアルフラッシュメモリ	Spansion 製 S25FL032P $\times$ 1

### 1.4 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。合わせて参照してください。

- SH7268/SH7269 グループ SPI マルチ I/O バスコントローラ シリアルフラッシュメモリ接続例

### 1.5 "L"アクティブ端子 (信号) の表記について

端子名 (信号名) 末尾の # は "L" アクティブ端子 (信号) であることを示します。

## 2. シリアルフラッシュブート機能概要

この章では、シリアルフラッシュブート機能の概要について説明します。

### 2.1 シリアルフラッシュブートに関連する用語

表 1に本アプリケーションノートで使用するシリアルフラッシュブート関連の用語を示します。

表1 シリアルフラッシュブート関連の用語

用語	説明
ブート起動用内蔵 ROM プログラム	ブートモード3で起動した場合に、シリアルフラッシュメモリの先頭に格納されているローダプログラムを高速内蔵 RAM に転送し、ローダプログラムに分岐する処理を行うプログラムです。CPU 内のブート起動用内蔵 ROM に格納されているため作成する必要はありません。
ローダプログラム	SPI マルチ I/O バスコントローラ (SPIBSC) の外部アドレス空間である SPI マルチ I/O バス空間に配置されたアプリケーションプログラムを実行可能な状態にするプログラムです。SPIBSC の外部アドレス空間リードモードを許可した状態でアプリケーションプログラムのエン트리関数に分岐します。ローダプログラムのプログラムサイズは 8K バイト固定です。システムに応じて作成してください。
アプリケーション プログラム	お客様がシステムに応じて作成するプログラムです。本アプリケーションノートでは、アプリケーションプログラムは SPIBSC の外部アドレス空間である SPI マルチ I/O バス空間に配置されることを想定しています。
ダウンローダ	ローダプログラムとアプリケーションプログラムをシリアルフラッシュメモリへ書き込むためのプログラムです。システムに応じて作成してください。

## 2.2 シリアルフラッシュブートの動作説明

表 2 にブートモードを決定する外部端子 (MD\_BOOT2~0) を示します。

表2 外部端子設定とシリアルフラッシュブートモードの関係

MD_BOOT2	MD_BOOT1	MD_BOOT0	ブートモード	説明
1	0	1	ブートモード 3	ルネサスシリアルペリフェラルインタフェースのチャンネル 0 に接続されたシリアルフラッシュメモリからブートします。

ブートモード 3 の場合パワーオンリセット解除後、ブート起動用内蔵ROMプログラムによりルネサスシリアルペリフェラルインタフェースチャンネル 0 (RSPIO) に接続されているシリアルフラッシュメモリからローダプログラムを高速内蔵RAMに転送します。転送完了後、ローダプログラムの先頭に分岐します。図 1 にブート起動用内蔵ROMプログラムの動作イメージ図を示します。これらの一連の処理は、自動的に行われます。

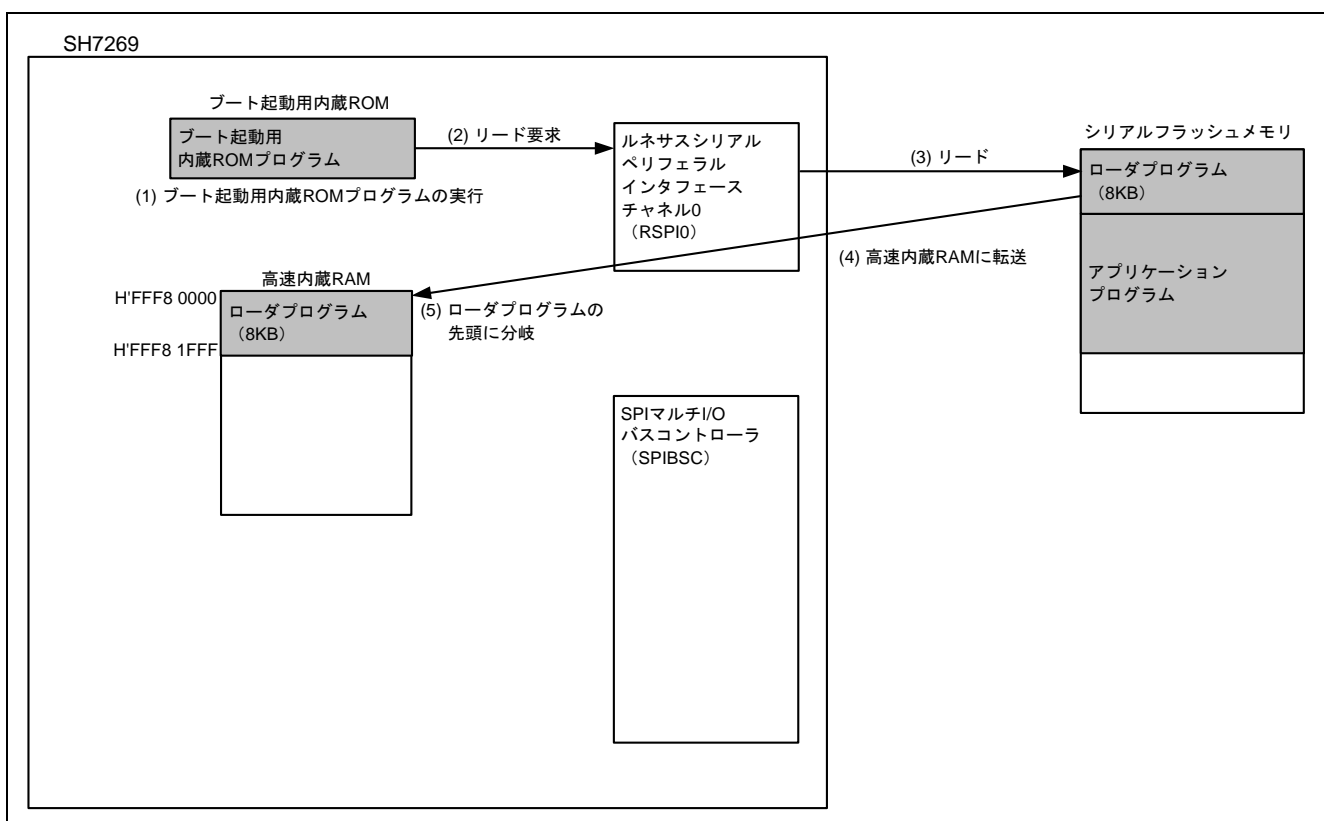


図1 ブート起動用内蔵 ROM プログラムの動作イメージ図

ローダプログラムはSPIBSCの外部アドレス空間リードモードを有効にします。この設定によりシリアルフラッシュメモリに格納されているアプリケーションプログラムを外部アドレス空間上に配置することができます。設定完了後、アプリケーションプログラムのエントリ関数に分岐します。図 2にローダプログラムの動作イメージ図を示します。

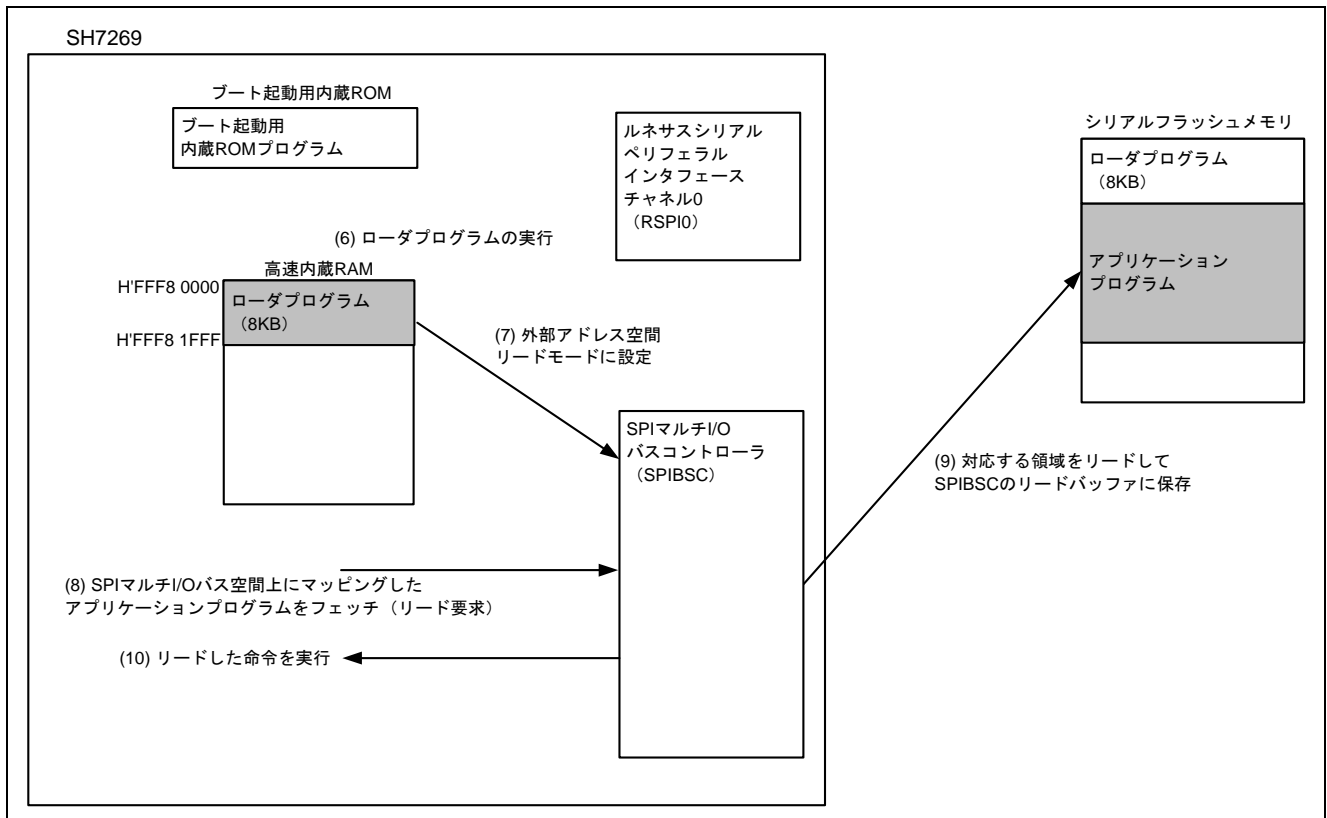


図2 ローダプログラムの動作イメージ図

## 2.3 ダウンローダの動作説明

ダウンローダは、高速内蔵RAM上に配置したローダプログラムとRAM上に配置したアプリケーションプログラムをシリアルフラッシュメモリに書き込むためのプログラムです。図3にダウンローダの動作イメージ図を示します。

詳細は「3.3 ダウンローダ例」を参照してください。

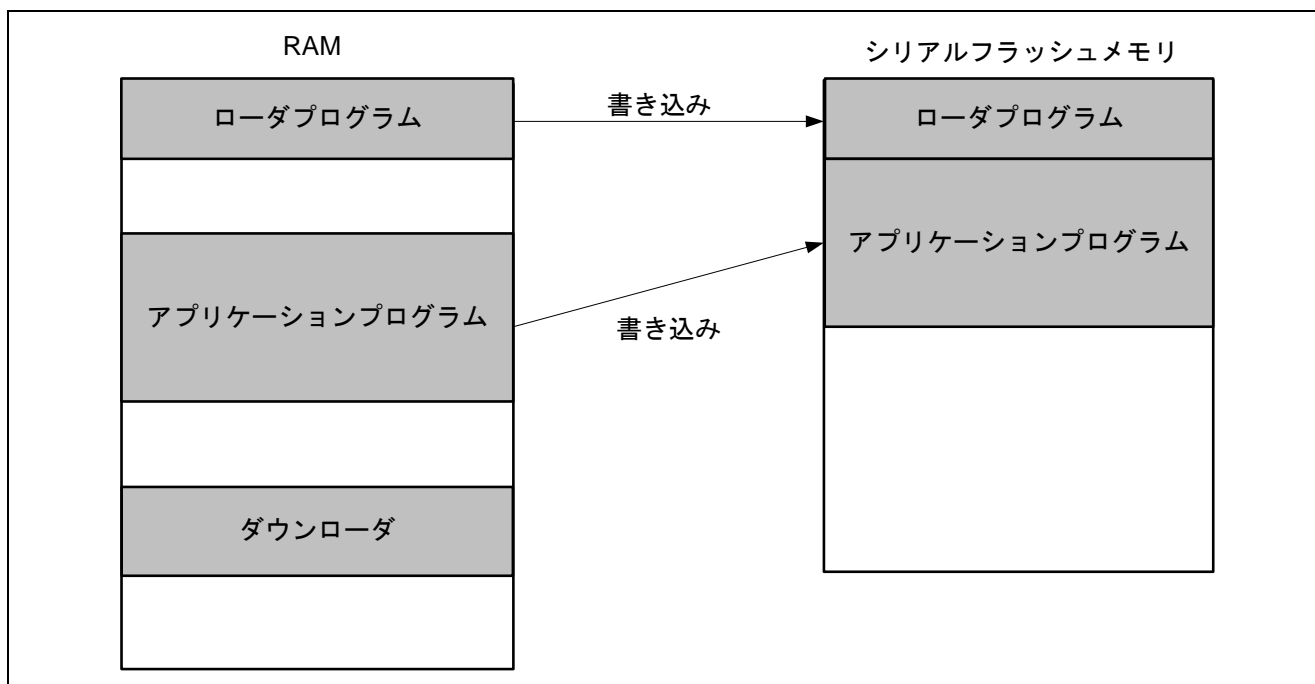


図3 ダウンローダの動作イメージ図

## 2.4 シリアルフラッシュメモリ接続例

図 4にシリアルフラッシュブート機能およびSPIBSCを使用する場合の接続例を示します。ブート起動用内蔵ROMプログラムはルネサスシリアルペリフェラルインタフェースのチャンネル 0 (RSPIO) を使用しますので、シリアルフラッシュメモリはRSPIO に接続します。

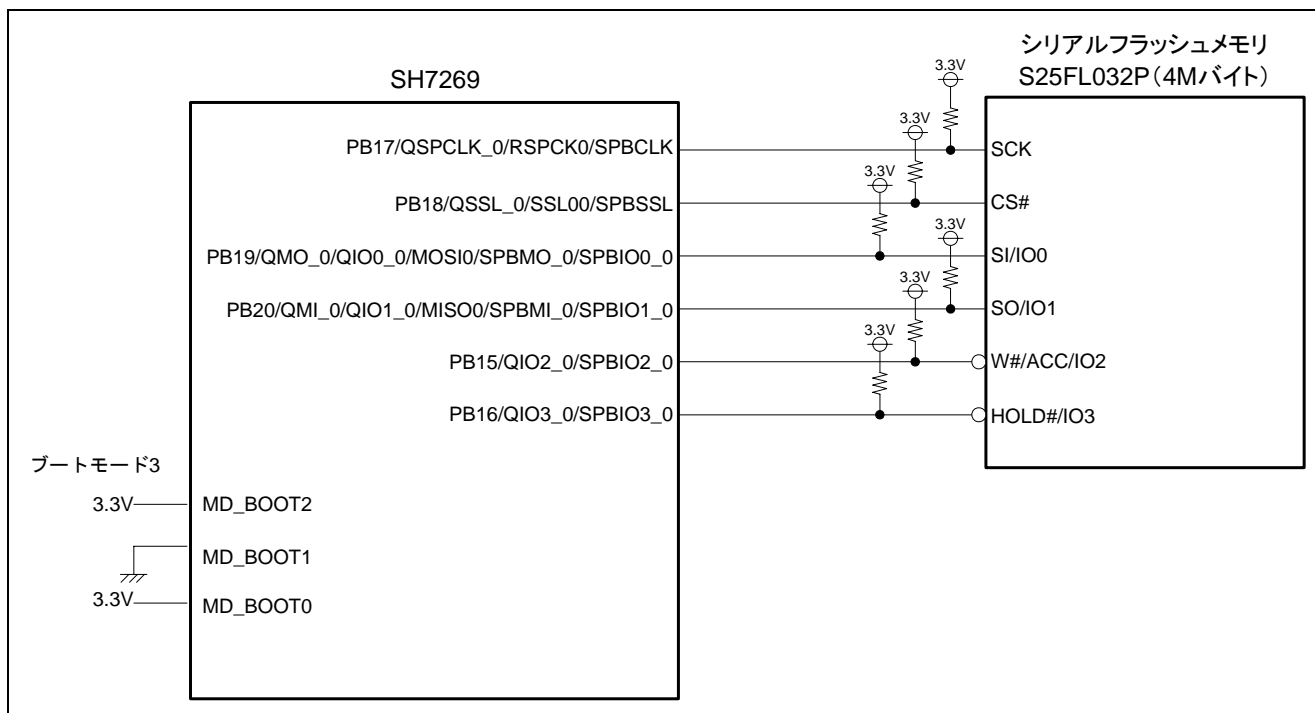


図4 シリアルフラッシュブートを行うための接続例

### 3. 応用例の説明

この章では、ローダプログラム、アプリケーションプログラム、ダウンローダについて説明します。

#### 3.1 ローダプログラムの仕様詳細

ローダプログラムは、SPI マルチ I/O バスコントローラ (SPIBSC) の外部アドレス空間リードモードを許可した後、アプリケーションプログラムのエントリ関数に分岐する処理を行います。

##### 3.1.1 メモリマップ

図 5 にローダプログラムのメモリマップを示します。

ローダプログラムは、ブート起動用内蔵 ROM プログラムによって、シリアルフラッシュメモリから高速内蔵 RAM に転送されます。転送元のシリアルフラッシュメモリの領域は、H'0000 0000 番地～H'0000 1FFF 番地までの 8K バイトの領域です。転送先の領域は、高速内蔵 RAM が割り当てられている H'FFF8 0000 番地～H'FFF8 1FFF 番地です。

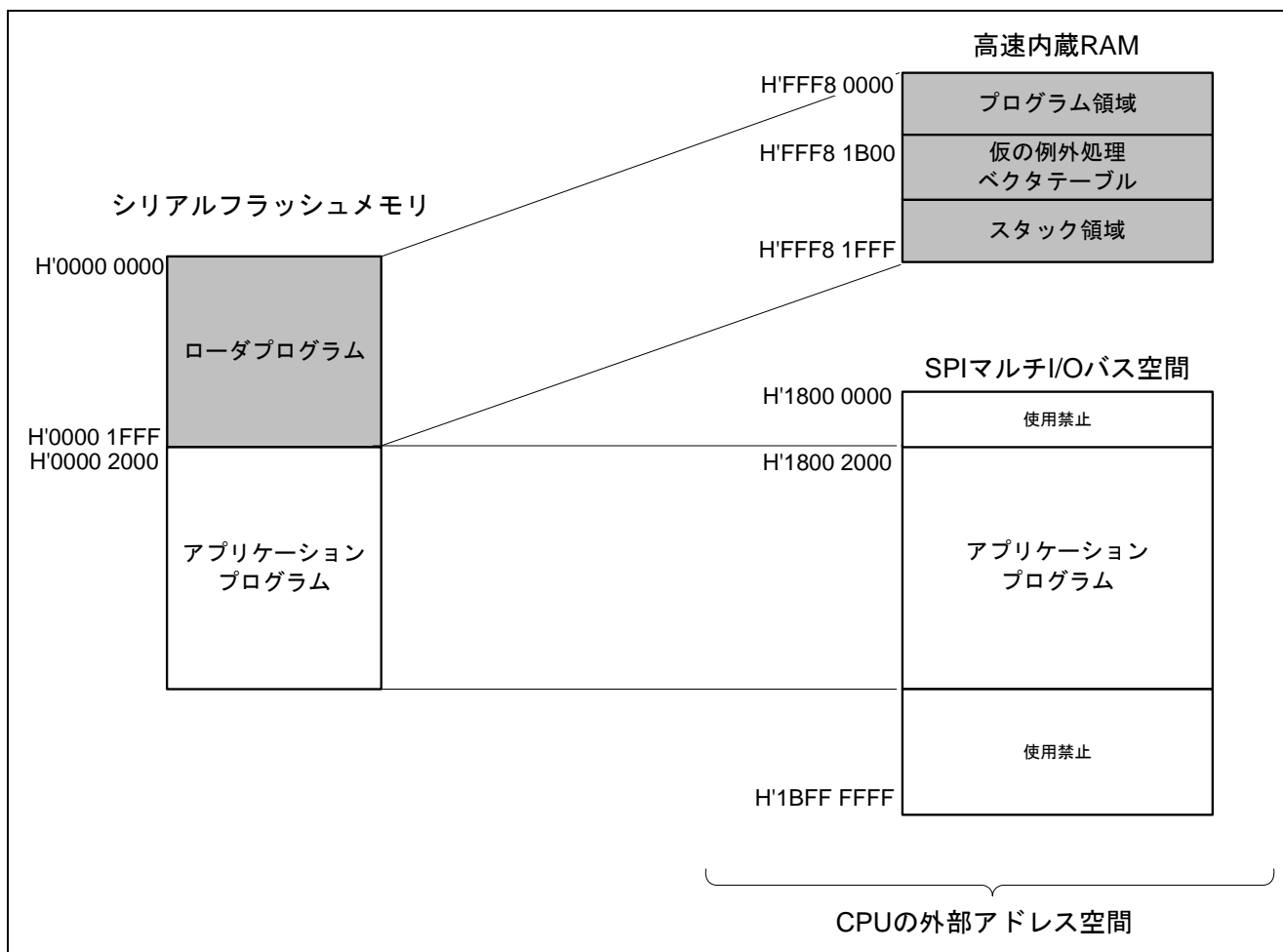


図5 ローダプログラムのメモリマップ



## 3.1.2 フローチャート

図 6にローダプログラムのフローチャートを示します。フローチャート中の各処理の詳細は 3.1.3～3.1.11を参照してください。

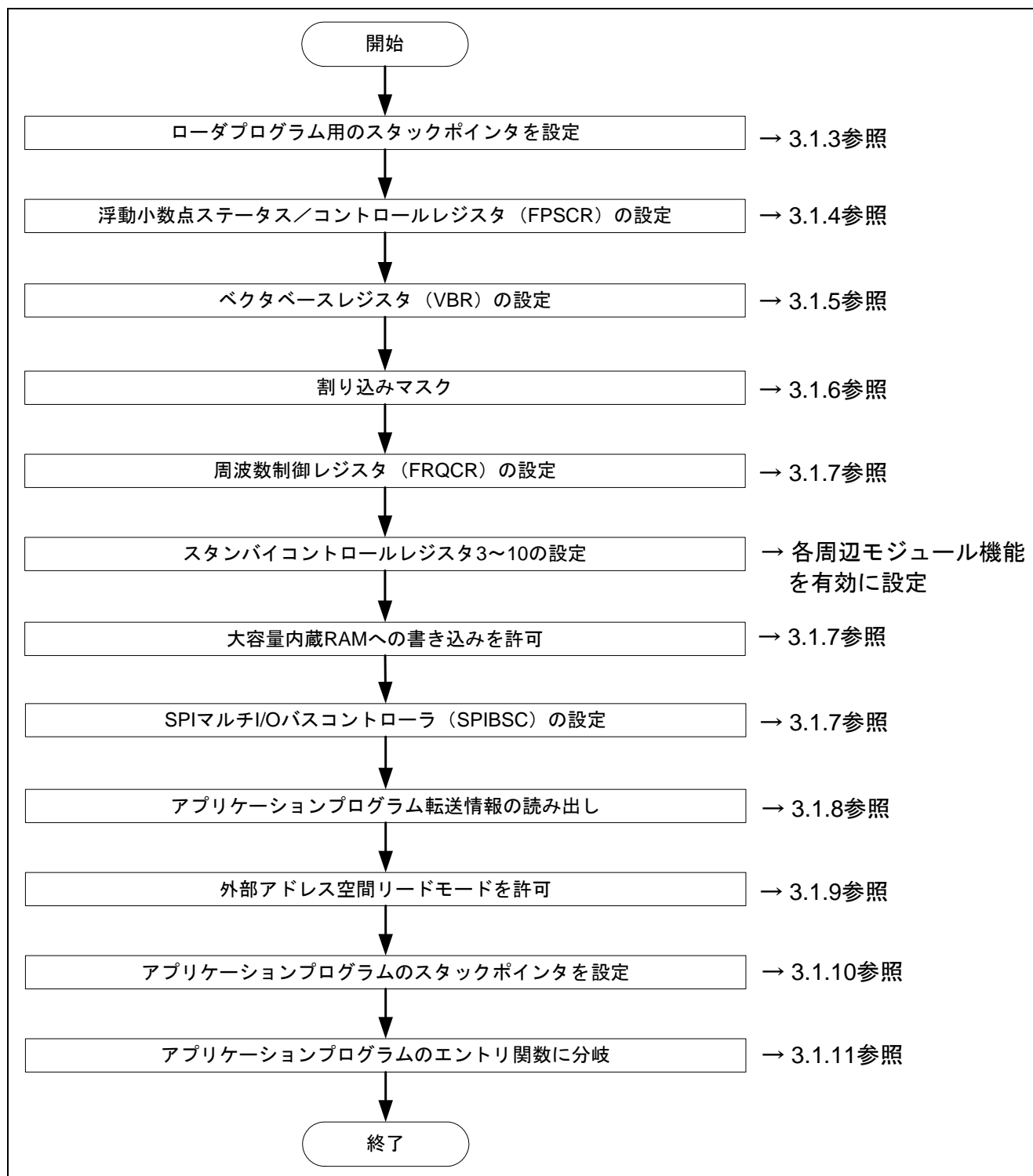


図6 ローダプログラムのフローチャート

### 3.1.3 スタックポインタの設定

スタックポインタ (R15) に H'FFF8 2000 番地を設定します。本処理は H'FFF8 0000 番地に配置し、不定なスタックポインタを使用しないようにアセンブリ言語で記述します。スタックポインタの設定以降は、C 言語による処理の記述が可能になります。スタックポインタ設定後、ローダプログラムのエン트리関数に分岐します。

### 3.1.4 浮動小数点ステータス/コントロールレジスタ (FPSCR) の設定

FPSCR に H'0004 0001 (単精度演算、丸めモード: 0 方向への丸め) を設定します。

### 3.1.5 ベクタベースレジスタ (VBR) の設定

ローダプログラムは、ローダプログラム動作中の例外処理に対応するために「仮の例外処理ベクタテーブル」を VBR に設定しています。VBR を設定するまでは、例外処理ベクタテーブルが不定のため例外や割り込みを発生させないでください。また、ローダプログラムでは割り込みを使用しないため、「仮の例外処理ベクタテーブル」はベクタ番号 0~18 のみ定義しています。ローダプログラム動作中に外部割り込み等の例外処理を行う場合は、「仮の例外処理ベクタテーブル」を拡張してください。

**【注】** 例外処理実行前には、あらかじめ例外処理ベクタテーブルをメモリ上に格納し、CPU がそのメモリにアクセスすることができるようにしておく必要があります。詳細は SH7268 グループ、SH7269 グループ ユーザーズマニュアル ハードウェア編の 6.9.4 「例外処理実行前の注意事項」を参照してください。

### 3.1.6 割り込みマスク

ローダプログラムは動作中の割り込みに対応しないため、ステータスレジスタ (SR) の割り込みマスクレベルビットに B'1111 を設定します。

### 3.1.7 初期設定

シリアルフラッシュメモリからアプリケーションプログラムを読み出すための各周辺機能の初期設定を行います。

### 3.1.8 アプリケーションプログラム転送情報の読み出し

ローダプログラムは、シリアルフラッシュメモリに格納されているアプリケーションプログラム転送情報 (appinfo) を参照してアプリケーションプログラムを割り当てた外部アドレスを取得します。表 3 にアプリケーションプログラム転送情報 (appinfo) の詳細を示します。アプリケーションプログラム転送情報 (appinfo) は、シリアルフラッシュメモリの H'0000 2000 番地に配置します。ローダプログラムは、シリアルフラッシュメモリの H'0000 2000~H'0000 2007 番地をアプリケーションプログラム転送情報として扱います。

なお、ここでのシリアルフラッシュメモリへのアクセスには SPIBSC の SPI モードを使用します。

表3 アプリケーションプログラム転送情報 (appinfo)

項目	アドレス	サイズ
転送先先頭アドレス	H'0000 2000	4
転送先最終アドレス	H'0000 2004	4

### 3.1.9 外部アドレス空間リードモードを許可

SPIBSC の外部アドレス空間リードモードを許可して、SPI マルチ I/O バス空間にアプリケーションプログラムを割り当てます。

### 3.1.10 アプリケーションプログラムのスタックポインタの設定

ローダプログラムは、アプリケーションプログラムの先頭から 12~15 バイト目に格納されているスタックポインタ設定値をスタックポインタ (R15) に設定します。

### 3.1.11 アプリケーションプログラムのエントリ関数への分岐

ローダプログラムは、アプリケーションプログラムの先頭から 8~11 バイト目に格納されているエントリ関数アドレスに分岐します。

### 3.1.12 ロータプログラム実行後のレジスタ状態

表 4 にローダプログラム実行後の各レジスタの状態を示します。表 4 に記載していないレジスタは、SH7268 グループ、SH7269 グループ ユーザーズマニュアル ハードウェア編に記載されている初期値と同じ値に設定されています。

表 4 ロータプログラム実行後の各レジスタ状態 (1)

レジスタ名	略名	設定値	備考
汎用レジスタ	R0~R14	不定	
プログラムカウンタ	PC	設定に依存	アプリケーションプログラムのエントリ関数アドレス
スタックポインタ	SP (R15)	設定に依存	アプリケーションプログラムのスタックポインタ設定値
ステータスレジスタ	SR	不定	但し、IMASK ビットは B'1111
ベクタベースレジスタ	VBR	H'FFF8 1B00	
浮動小数点ステータス/ コントロールレジスタ	FPSCR	H'0004 0001	単精度演算 丸めモード : 0 方向への丸め
周波数制御レジスタ	FRQCR	H'1015	
スタンバイコントロールレジスタ 3	STBCR3	H'1A	
スタンバイコントロールレジスタ 4	STBCR4	H'00	
スタンバイコントロールレジスタ 5	STBCR5	H'00	
スタンバイコントロールレジスタ 6	STBCR6	H'00	
スタンバイコントロールレジスタ 7	STBCR7	H'12	
スタンバイコントロールレジスタ 8	STBCR8	H'09	
システムコントロールレジスタ 5	SYSCR5	H'0F	大容量内蔵 RAM への 書き込み許可
共通コントロールレジスタ	CMNCR	H'00FF F320	
ビットレート設定レジスタ	SPBCR	H'0000 0100	
データリードコントロールレジスタ	DRCR	H'0001 0101	
データリードコマンド設定レジスタ	DRCMR	H'00EB 0000	
データリードイネーブル設定レジスタ	DRENCR	H'0222 47E0	
SPI モードコントロールレジスタ	SMCR	H'0000 0004	
SPI モードコマンド設定レジスタ	SMCMR	H'006B 0000	

表5 ローダプログラム実行後の各レジスタ状態 (2)

レジスタ名	略名	設定値	備考
SPI モードアドレス設定レジスタ	SMADR	H'0002 0004	
SPI モードオプション設定レジスタ	SMOPR	H'0003 0000	
SPI モードイネーブル設定レジスタ	SMENR	H'0002 000F	
共通ステータスレジスタ	CMNSR	H'0000 0003	
ポート B コントロールレジスタ 5	PBCR5	H'0006	
ポート B コントロールレジスタ 4	PBCR4	H'6666	
ポート B コントロールレジスタ 3	PBCR3	H'6000	

### 3.2 アプリケーションプログラム例

アプリケーションプログラムは、外部アドレス空間リードモードを使用して読み出すため、SPI マルチ I/O バス空間に配置する必要があります。また、アプリケーションプログラムには、ローダプログラムが参照するアドレス情報を組み込む必要があります。

ここでは、シリアルフラッシュブート固有のアプリケーションプログラム作成方法について説明します。

#### 3.2.1 セクション配置

ここでは、アプリケーションプログラムのセクション配置について説明します。

1. アプリケーションプログラムは、外部アドレス空間リードモードを使用して実行するプログラムです。そのため、本アプリケーションプログラム例では、アプリケーションプログラムのセクションを SPI マルチ I/O バス空間上に配置します。
2. ローダプログラムが参照するアプリケーションプログラム転送情報 (appinfo)、アプリケーションプログラムのエン트리関数アドレスやスタックポインタ設定値は固定アドレスになるようにセクション配置を行います。アプリケーションプログラム転送情報 (appinfo) は DAPPINFO セクションに、アプリケーションプログラムのエン트리関数アドレスおよびスタックポインタ設定値は DVECTTBL セクションに配置します。アプリケーションプログラムの先頭から DAPPINFO セクション、DVECTTBL セクションの順に配置します。
3. SPI マルチ I/O バス空間の H'1800 0000~H'1800 1FFF 番地に対応するシリアルフラッシュメモリの領域はローダプログラムが使用しています。そのため、アプリケーションプログラムのプログラム領域、定数領域、初期化データ領域は H'1800 2000 番地以降の領域に配置してください。
4. DVECTTBL セクションの先頭にリセットベクタテーブル RESET\_Vectors を配置します。

図 7 にセクション配置例を示します。

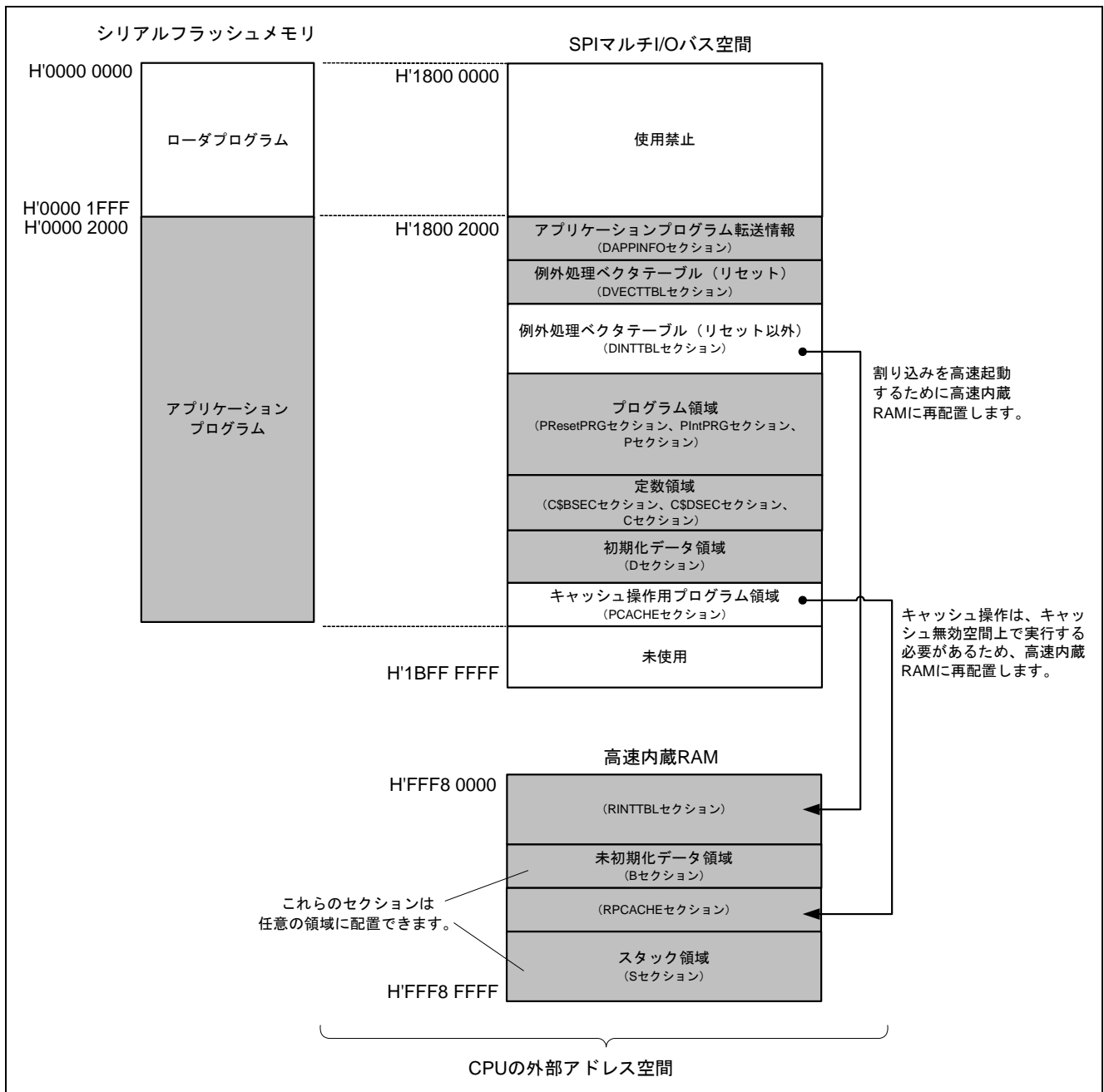


図7 アプリケーションプログラムのセクション配置例

## 3.2.2 フローチャート

本応用例のアプリケーションプログラムでは、FIFO内蔵シリアルコミュニケーションインタフェースのチャンネル2 (SCIF2) へ文字列データを送信した後、コンペアマッチタイマのチャンネル0を使用したLED点滅処理を行っています。図8にアプリケーションプログラムのフローチャートを示します。

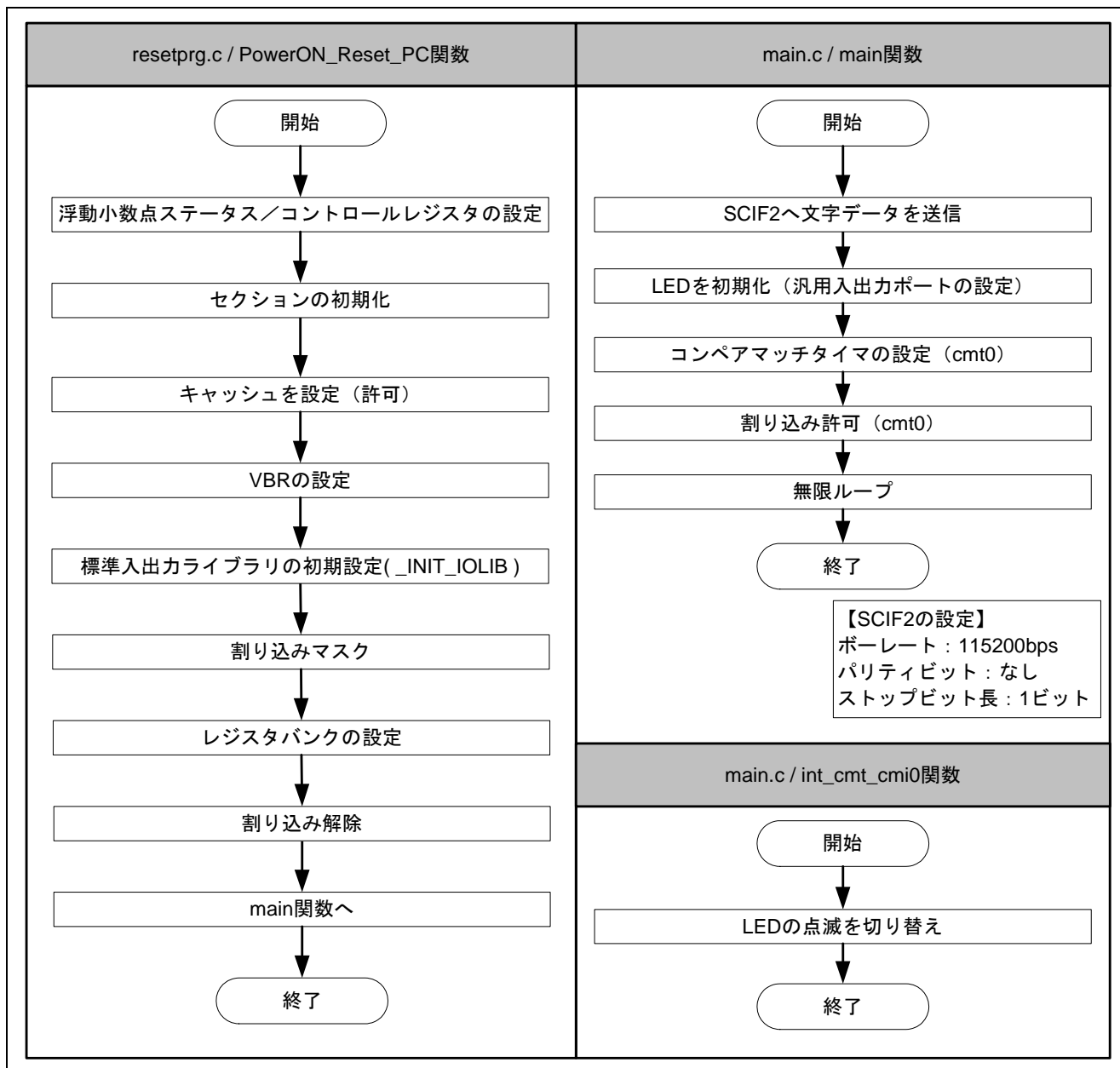


図8 アプリケーションプログラムのフローチャート

### 3.2.3 エントリ関数の設定

アプリケーションプログラムのエントリ関数アドレスは、リセットベクタテーブルRESET\_Vectorsのテーブル番号0に設定します。表6にエントリ関数アドレスの設定を示します。

表6 エントリ関数アドレスの設定

項目	設定
ファイル名	vecttbl.c
配置セクション名	DVECTTBL
テーブル名	RESET_Vectors
テーブル番号	0
初期設定	PowerON_Reset_PC

【注】 PowerON\_Reset\_PCはアプリケーションプログラムのエントリ関数です。

### 3.2.4 スタックポインタの設定

アプリケーションプログラムのスタックポインタは、リセットベクタテーブルRESET\_Vectorsのテーブル番号1に設定します。表7にスタックポインタ設定値を示します。

表7 スタックポインタ設定値の設定

項目	設定
ファイル名	vecttbl.c
配置セクション名	DVECTTBL
テーブル名	RESET_Vectors
テーブル番号	1
初期設定	__secend("S")

### 3.2.5 セクションの初期化

dbsect.cに記述されているセクション初期化用テーブルDTBL、BTBLに格納されている値を使用して、セクションの初期化ルーチン（\_INITSCT関数）を実行することによりセクションの初期化を行います。セクションの初期化ルーチン（\_INITSCT関数）を実行後、キャッシュメモリと大容量内蔵RAMのコヒーレンシを保証するためにキャッシュのライトバック処理を行います。

### 3.2.6 キャッシュの設定（許可）

キャッシュ制御レジスタ1を設定して命令キャッシュおよびオペランドキャッシュを有効にします。

### 3.2.7 ベクタベースレジスタ（VBR）の設定

アプリケーションプログラムの例外処理ベクタテーブルをVBRに設定します。



### 3.2.8 アプリケーションプログラム転送情報 (appinfo) の生成

表 8 にアプリケーションプログラム転送情報 (appinfo) を生成するための構造体を示します。アプリケーションプログラムの先頭アドレスと最終アドレスは、セクションアドレス演算子 (`__sectop`、`__secend`) を使用して取得します。この構造体は DAPPINFO セクションに配置します。`app_top` には、アプリケーションプログラム (プログラム領域、定数領域、初期化データ領域) の先頭アドレスを登録してください。`app_end` には、アプリケーションプログラム (プログラム領域、定数領域、初期化データ領域) の最終アドレスを登録してください。

表 8 アプリケーションプログラム転送情報 (appinfo)

項目	説明		
ファイル名	appinfo.c		
構造体名	appinfo		
構造体メンバ	メンバ名	設定値	説明
	<code>void *app_top</code>	<code>__sectop("DAPPINFO")</code>	アプリケーションプログラムの先頭アドレス
	<code>void *app_end</code>	<code>__secend("PCACHE")</code>	アプリケーションプログラムの最終アドレス+1
配置セクション名	DAPPINFO		

【注】 ロードプログラム (8K バイト) とアプリケーションプログラムのプログラムサイズの合計が、使用するシリアルフラッシュメモリの容量を超えないようにしてください。

図 9 にアプリケーションプログラム転送情報 (appinfo) の生成イメージ図を示します。

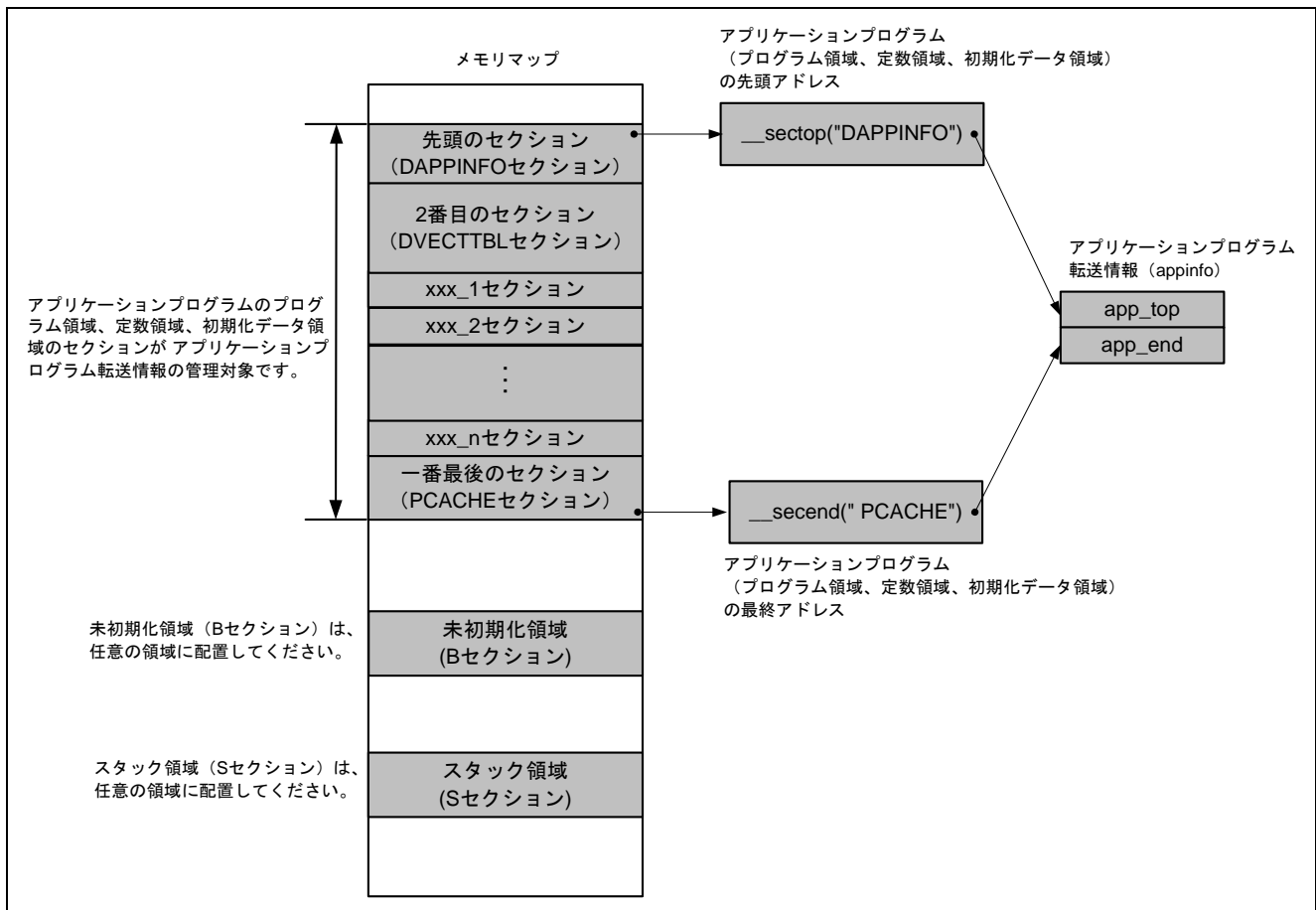


図9 アプリケーションプログラム転送情報 (appinfo) の生成イメージ図

### 3.3 ダウンローダ例

ここでは、本応用例におけるダウンローダについて説明します。

#### 3.3.1 動作概要

ダウンローダ実行前に、デバッガを使用して開発環境からシステム上の高速内蔵RAMにダウンローダ、ローダプログラムを転送し、大容量内蔵RAMにアプリケーションプログラムを転送します。図 10に動作イメージ図を示します。

なお、アプリケーションプログラムは SPI マルチ I/O バス空間上に配置しますが、デバッガを使用して SPI マルチ I/O バス空間にプログラムを転送することはできません。そのため mot ファイルを大容量内蔵 RAM 上にダウンロードし、SPI マルチ I/O バス空間には abs ファイルのデバッグ情報のみをダウンロードします。

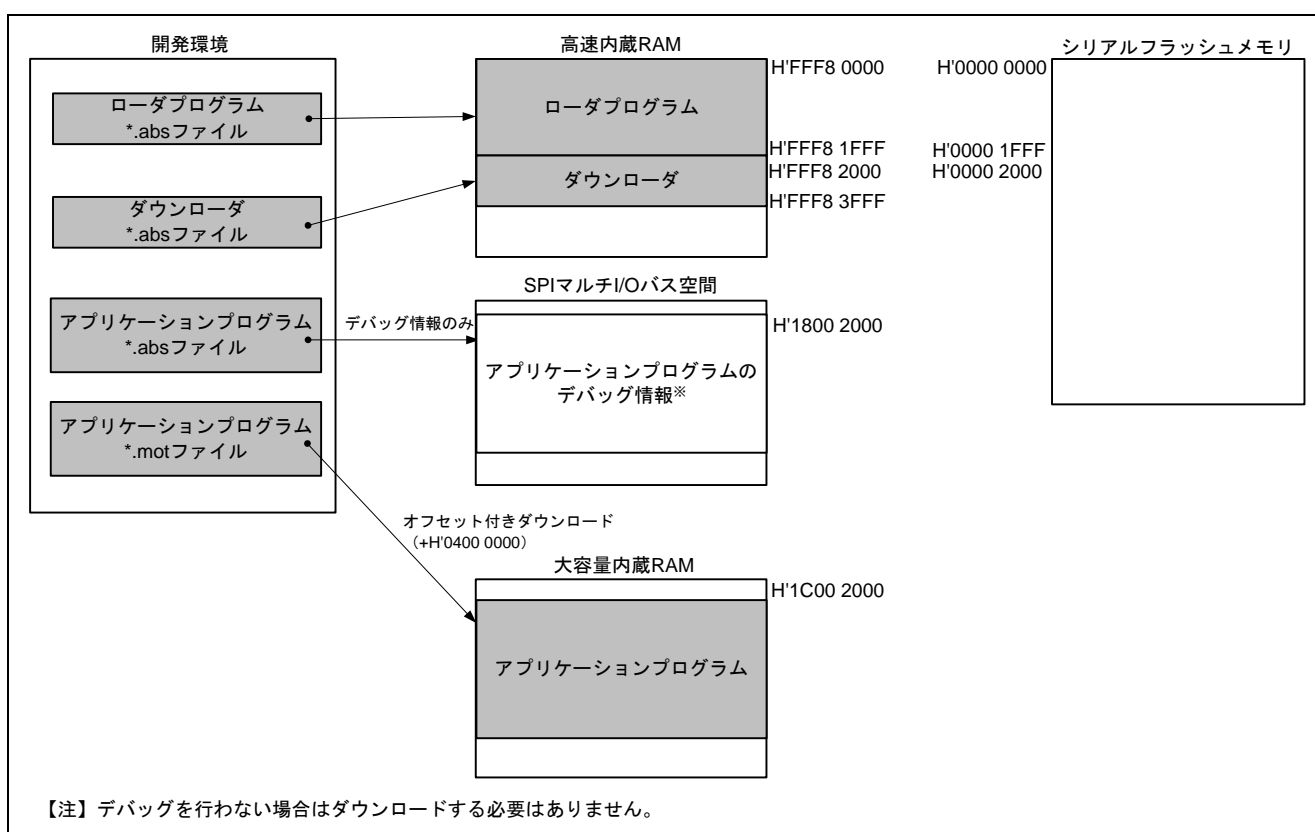


図10 ダウンローダ動作イメージ図 (1)

ダウンローダを実行してローダプログラムおよびアプリケーションプログラムをシリアルフラッシュメモリに書き込みます。ダウンローダは、ローダプログラムをシリアルフラッシュメモリのH'0000 0000～H'0000 1FFF番地に、アプリケーションプログラムをH'0000 2000番地以降に書き込みます。図 11に動作イメージ図を示します。

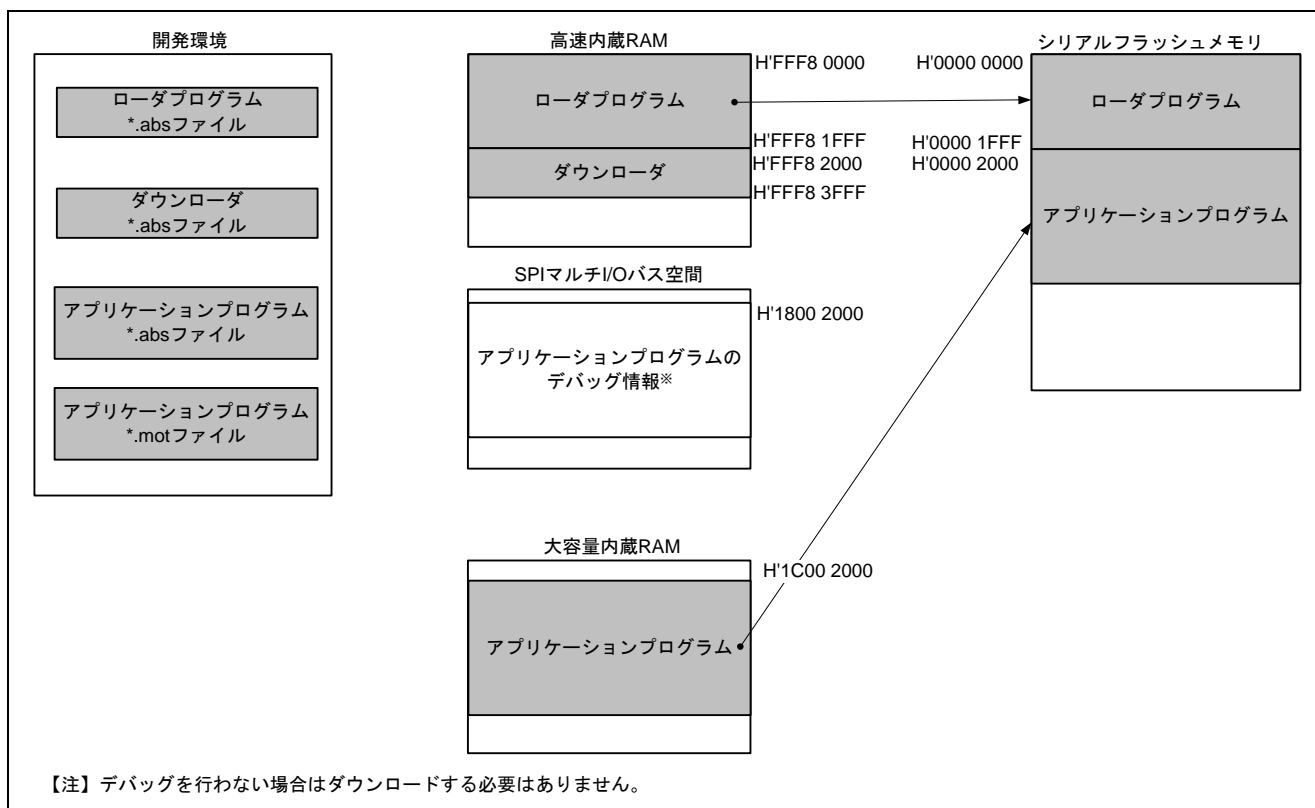


図11 ダウンローダ動作イメージ図 (2)

### 3.3.2 ダウンローダの使用領域

ダウンローダの使用領域は、H'FFF8 2000～H'FFF8 3FFFです。ローダプログラム、アプリケーションプログラムおよびダウンローダのセクションが競合すると、それぞれのプログラムは正常に動作しません。

### 3.3.3 フローチャート

図 12にダウンローダのフローチャートを示します。高速内蔵RAM上に配置したダウンローダを実行し、シリアルフラッシュメモリへの書き込み処理を行います。フローチャート中の各処理の詳細は 3.3.4～3.3.8を参照してください。

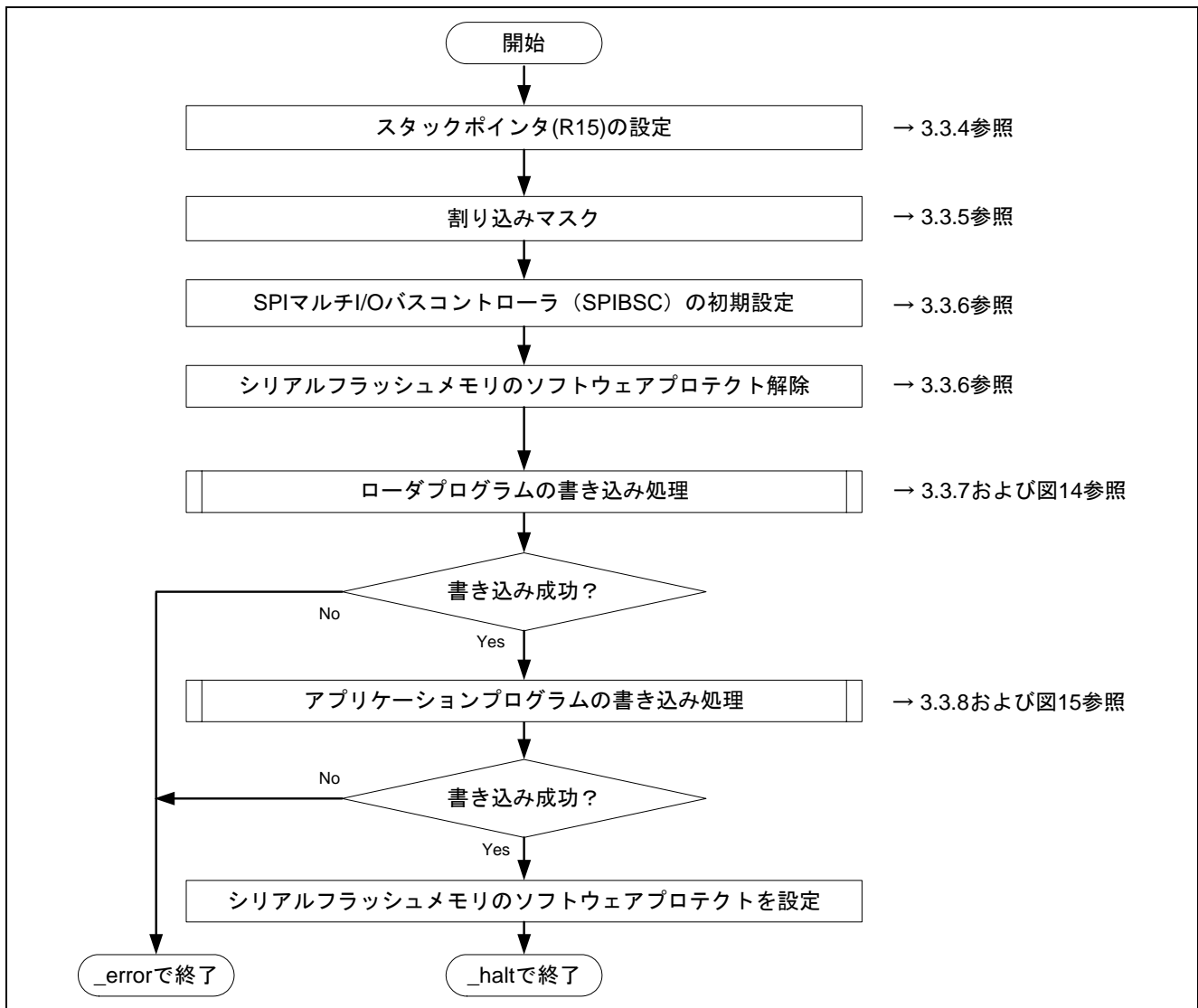


図12 ダウンローダのフローチャート

図 13に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書き込み処理のフローチャートを示します。

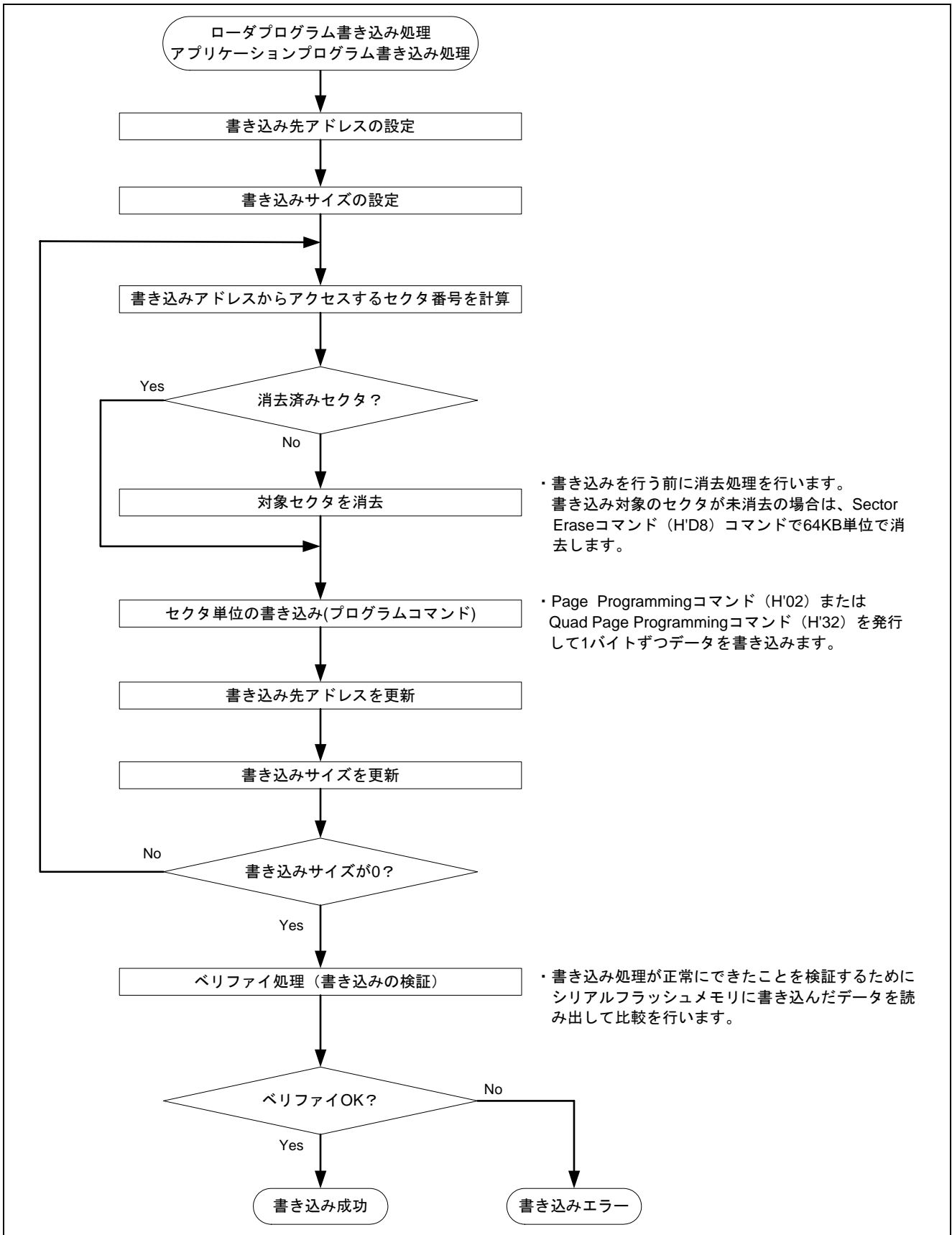


図 13 書き込み処理のフローチャート

### 3.3.4 スタックポインタの設定

スタックポインタ (R15) に H'FFF8 4000 番地を設定します。本処理は H'FFF8 2000 番地に配置し、不定なスタックポインタを使用しないようにアセンブリ言語で記述します。スタックポインタの設定以降は、C 言語による処理の記述が可能になります。スタックポインタ設定後、ダウンローダのエントリ関数に分岐します。

### 3.3.5 割り込みマスク

ダウンローダは動作中の割り込みに対応しないため、ステータスレジスタ (SR) の割り込みマスクレベルビットに B'1111 を設定します。

### 3.3.6 初期設定

シリアルフラッシュメモリにアクセスするために、以下の初期設定を行います。

1. SPIBSC の初期設定を行います。
2. シリアルフラッシュメモリに対して Write Enable コマンドを発行し、ソフトウェアプロテクトを解除します。

### 3.3.7 ローダプログラムの書き込み処理

ダウンローダは、高速内蔵RAMのH'FFF8 0000～H'FFF8 1FFF番地に転送されているローダプログラムをシリアルフラッシュメモリのH'0000 0000～H'0000 1FFF番地へ書き込みます。表 9 にローダプログラムの書き込み処理を示します。

表9 ローダプログラムの書き込み処理

項目	詳細
ローダプログラムの転送元アドレス (高速内蔵 RAM)	H'FFF8 0000 (固定)
ローダプログラムの転送先アドレス (シリアルフラッシュメモリ)	H'0000 0000 (固定)
転送サイズ	H'2000 (固定)
書き込み処理手順	<ol style="list-style-type: none"> <li>1. 書き込み先アドレスが消去済みかどうかを確認します。</li> <li>2. 未消去エリアの場合は消去処理を行います。</li> <li>3. プログラムコマンドを発行し書き込み処理を行います。</li> </ol>

### 3.3.8 アプリケーションプログラムの書き込み処理

ダウンローダは、大容量内蔵RAMに転送されているアプリケーションプログラムをシリアルフラッシュメモリのH'0000 2000番地から書き込みます。表10にアプリケーションプログラムの書き込み処理を示します。

表10 アプリケーションプログラムの書き込み処理

項目	詳細
アプリケーションプログラムの転送元アドレス (大容量内蔵 RAM)	H'1C00 2000 (mot ファイルのダウンロードアドレス)
アプリケーションプログラムの転送先アドレス (シリアルフラッシュメモリ)	H'0000 2000 (固定)
転送サイズ	アプリケーションプログラム内にある appinfo から計算 (アプリケーションプログラムに依存)
書き込み処理手順	1. 書き込み先アドレスが消去済みかどうかを確認します。 2. 未消去エリアの場合は消去処理を行います。 3. プログラムコマンドを発行し書き込み処理を行います。



## 3.3.9 バッチファイル

ローダプログラムとアプリケーションプログラムをシリアルフラッシュメモリに書き込むためには、ローダプログラムおよびダウンローダを高速内蔵 RAM に、アプリケーションプログラムを大容量内蔵 RAM に転送してからダウンローダを実行する必要があります。これらの処理は手動で行うことも可能ですが、本アプリケーションノートでは、一連の処理を自動化するために High-performance Embedded Workshop のコマンドバッチファイルを使用します。

図 14 にコマンドバッチファイルのフローチャートを示します。このバッチファイルにより、高速内蔵 RAM・大容量内蔵 RAM へのプログラム転送とシリアルフラッシュメモリへの書き込み処理を行います。

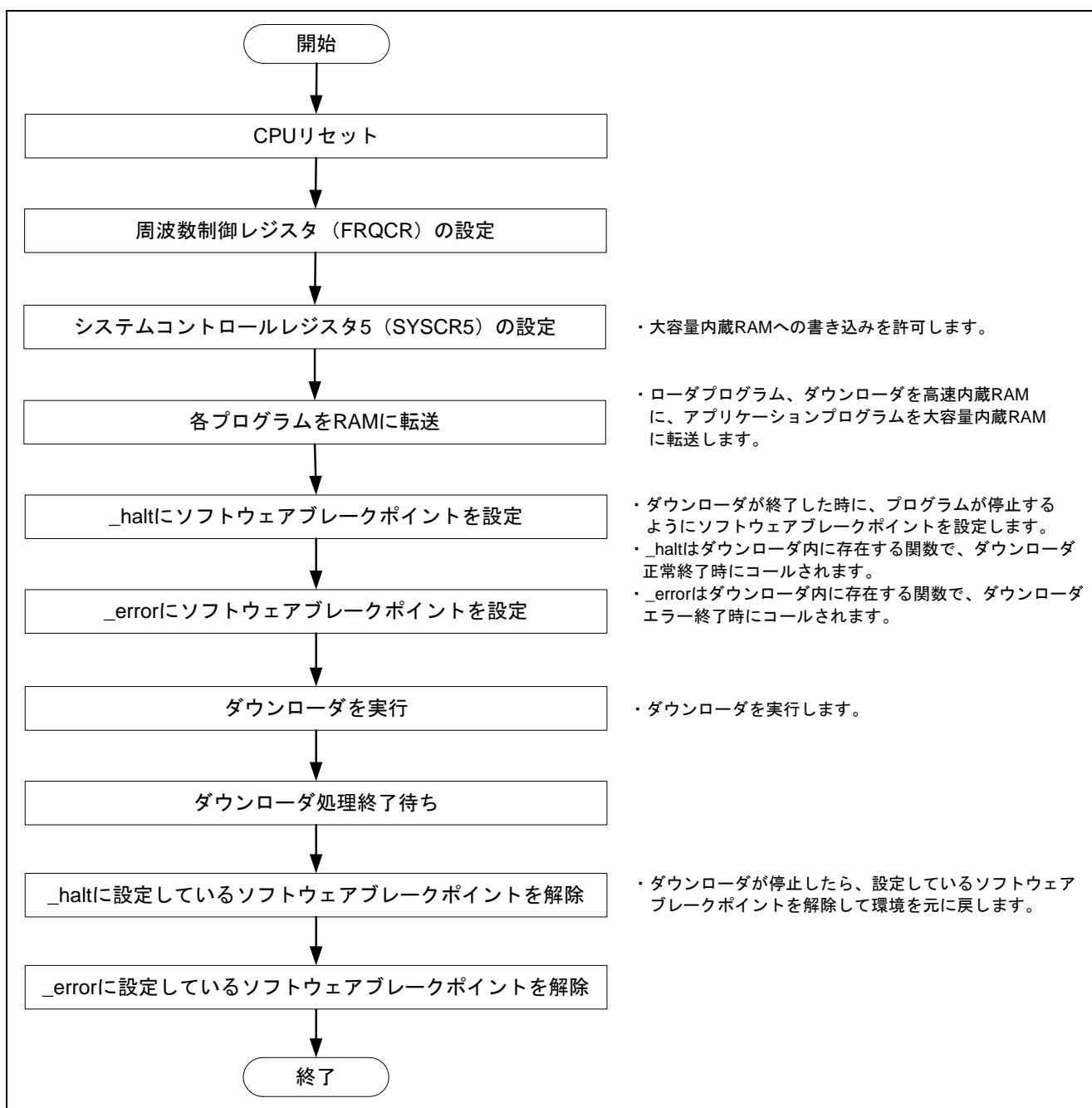


図14 コマンドバッチファイルのフローチャート

## 4. 参考プログラム例

### 4.1 ローダプログラム

#### 4.1.1 ローダプログラムリスト"ld\_loader.src" (1)

```
1 ;/*****
2 ;*  DISCLAIMER
3 ;*
4 ;*  This software is supplied by Renesas Electronics Corporation and is only
5 ;*  intended for use with Renesas products. No other uses are authorized.
6 ;*
7 ;*  This software is owned by Renesas Electronics Corporation and is protected under
8 ;*  all applicable laws, including copyright laws.
9 ;*
10 ;*  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 ;*  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 ;*  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 ;*  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 ;*  DISCLAIMED.
15 ;*
16 ;*  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 ;*  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 ;*  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 ;*  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 ;*  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 ;*
22 ;*  Renesas reserves the right, without notice, to make changes to this
23 ;*  software and to discontinue the availability of this software.
24 ;*  By using this software, you agree to the additional terms and
25 ;*  conditions found by accessing the following link:
26 ;*  http://www.renesas.com/disclaimer
27 ;*****
28 ;*  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29 ;***** Technical reference data *****/
30 ;*  System Name : SH7268/SH7269 Firm Update Sample Program
31 ;*  File Name   : ld_loader.src
32 ;*  Abstract    : Loader program preprocessing/jump processing to the application
33 ;*              : program
34 ;*  Version     : 1.00.00
35 ;*  Device      : SH7268/SH7269
36 ;*  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37 ;*              : C/C++ compiler package for the SuperH RISC engine family
38 ;*              :                               (Ver.9.03Release02).
39 ;*  OS          : None
40 ;*  H/W Platform: R0K57269(CPU board)
41 ;*  Description :
42 ;*****
43 ;*  History     : Jul.06,2011 Ver.1.00.00
44 ;*"FILE COMMENT END"*****/
45     .SECTION LOADER_ENTRY, CODE, ALIGN = 4
46     .IMPORT _main
47     .EXPORT _jmp_app_prog
48
```

## 4.1.2 ロードプログラムリスト"ld\_loader.src" (2)

```

49  _loader_prog:
50  MOV.L L2,R15 ; Sets the stack pointer
51  MOV.L L1,R0  ; Retrieves the entry function of the loader program
52  JMP @R0     ; Jumps to the entry function of the loader program
53  NOP
54
55
56  ;/*"FUNC COMMENT"*****
57  ; * ID      :
58  ; * Outline : Jump to the application program
59  ; *-----
60  ; * Include :
61  ; *-----
62  ; * Declaration : _jmp_app_prog
63  ; *-----
64  ; * Description : 1. Retrieves the stack pointer value stored in the first 12 to
65  ; *               : 15 bytes in the application program.
66  ; *               : 2. Specifies the stack pointer (R15).
67  ; *               : 3. Retrieves the entry function address stored in the first 8 to
68  ; *               : 11 bytes in the application program.
69  ; *               : 4. Jumps to the entry function.
70  ; *-----
71  ; * Argument  : R4 ; I : Start address of the application program
72  ; *-----
73  ; * Return Value: none
74  ; /*"FUNC COMMENT END"*****
75  _jmp_app_prog:
76
77  MOV.L R4,R0 ; Substitutes the start address of the application program for R0
78  ADD #12,R0 ; Calculates the address storing the stack pointer value and
79             ; substitutes the address for R0
80  MOV.L @R0,R15 ; Sets the stack pointer
81
82  MOV.L R4,R0 ; Substitutes the start address of the application program for R0
83  ADD #8,R0 ; Calculates the address storing the entry function of the application
84            ; program and substitutes the address for R0
85  MOV.L @R0,R0 ; Substitutes the entry function address of the application
86            ; program for R0
87  JMP @R0 ; Jumps to the entry function of the application program
88  NOP
89
90
91  .ALIGN 4
92  L1:
93  .DATA.L _main ; Entry function address of the loader program
94
95  L2:
96  .DATA.L H'FFF82000 ; Stack pointer (R15) value of the loader program
97
98  .pool
99  .end
100
101  ;/* End of File */

```

## 4.1.3 ローダプログラムリスト"ld\_main.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 *  System Name : SH7268/SH7269 Firm Update Sample Program
31 *  File Name   : ld_main.c
32 *  Abstract    : loader main
33 *  Version     : 1.00.00
34 *  Device      : SH7268/SH7269
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                      (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57269(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.06,2011 Ver.1.00.00
43 *****/
44 #include <stdio.h>
45 #include <string.h>
46 #include <machine.h>
47 #include "iodefine.h"
48 #include "serial_flash.h"
49
```

## 4.1.4 ローダプログラムリスト "ld\_main.c" (2)

```
50  /* ==== macro defined ==== */
51  #define FPSCR_INIT      0x00040001      /* Value to set in the FPSCR register */
52  #define INT_MASK       0x000000F0      /* Value to set in the SR register
53                                     (for masking the interrupt) */
54
55  #define APROG_TOP_SFLASH 0x00002000     /* Start address of the application program */
56                                     /* (serial flash memory) */
57
58  #define APPINFO_TOP     APROG_TOP_SFLASH /* Address the appinfo.app_top is located */
59  #define APPINFO_END     (APROG_TOP_SFLASH + 4) /* Address the appinfo.app_end is located */
60
61
62  /* ==== prototype declaration ==== */
63  void main(void);
64  void get_appinfo( unsigned long *app_top_addr, unsigned long *app_end_addr);
65  void app_prog_transfer(unsigned long app_top_addr, unsigned long app_end_addr);
66  void system_down(void);
67
68  extern void jmp_app_prog(unsigned long app_top_addr);
69  extern void io_set_cpg(void);
70  extern void sf_byte_read_long(unsigned long addr, unsigned long *buf, int size);
71
72  /* ==== external data ==== */
73  extern unsigned long DUMMY_Vectors;
74
75
```

## 4.1.5 ロードプログラムリスト"ld\_main.c" (3)

```
76  /*****
77  * ID      :
78  * Outline : Loader program main
79  * Include : #include "serial_flash.h"
80  * Declaration : void main(void);
81  * Description : Refers the data in the appinfo to transfer the application program
82  *              : to the large-capacity internal RAM, and jumps to the entry function
83  *              : of the application program.
84  * Argument : void
85  * Return Value: void
86  *****/
87  void main(void)
88  {
89      unsigned long app_top,app_end;
90
91
92      /* Sets the FPSCR */
93      set_fpscr(FPSCR_INIT);
94
95      /* Sets the tentative VBR */
96      set_vbr((void *)&DUMMY_Vectors);
97
98      /* Masks the interrupt */
99      set_cr(INT_MASK);
100
101      /* Sets the CPG */
102      io_set_cpg();
103
104      /* Sets the SPIBSC */
105      sf_init_serial_flash_spibsc();
106
107      /* Retrieves the appinfo */
108      get_appinfo(&app_top,&app_end);
109
110      sf_allocate_exspace_spibsc();
111
112      /* Jumps to the application program */
113      jmp_app_prog(app_top);
114
115      while(1){
116          /* LOOP */
117      }
118  }
119
```

## 4.1.6 ローダプログラムリスト"ld\_main.c" (4)

```
120  /*****
121  * ID      :
122  * Outline : Retrieve the appinfo
123  * Include : #include "serial_flash.h"
124  * Declaration : void get_appinfo (unsigned long *app_top_addr,
125  *      :      unsigned long *app_end_addr);
126  * Description : Retrieves the appinfo.
127  *      : Retrieves the appinfo.top from H'2000 to H'2003 in serial flash
128  *      : memory, and stores it in the address specified by the first
129  *      : argument. This function also retrieves the appinfo.end from
130  *      : H'2004 to H'2007 in serial flash memory, and stores it in the
131  *      : address specified by the second argument.
132  * Argument : unsigned long app_top_addr ; 0 : Start address of the application
133  *      :      program at destination
134  *      : unsigned long app_end_addr ; 0 : End address of the application
135  *      :      program at destination
136  * Return Value: void
137  *****/
138  void get_appinfo( unsigned long *app_top_addr, unsigned long *app_end_addr)
139  {
140
141      /* Retrieves the appinfo.top */
142      sf_byte_read_spibsc(APPINFO_TOP, (unsigned char *)app_top_addr, 4);
143
144      /* Retrieves the appinfo.end */
145      sf_byte_read_spibsc(APPINFO_END, (unsigned char *)app_end_addr, 4);
146
147  }
148
```

## 4.1.7 ローダプログラムリスト"ld\_main.c" (5)

```

149  /*****
150  * ID      :
151  * Outline  : Transfer the application program
152  * Include  : #include "serial_flash.h"
153  * Declaration : void app_prog_transfer(unsigned long app_top_addr,
154  *          :          unsigned long app_end_addr);
155  * Description : Calculates the size of the application program, and transfers
156  *          : the application program from serial flash memory to the
157  *          : large-capacity internal RAM. (Rounds up the allocation size of the
158  *          : application program to multiples of 4 to transfer in longword.)
159  * Argument  : unsigned long app_top_addr ; I : Start address of the application
160  *          :          program at destination
161  *          : unsigned long app_end_addr ; I : End address of the application
162  *          :          at destination
163  * Return Value: void
164  *****/
165  void app_prog_transfer(unsigned long app_top_addr, unsigned long app_end_addr)
166  {
167      unsigned long app_prog_size;
168
169      /* Calculates the size of the application program */
170      app_prog_size = app_end_addr - app_top_addr;
171      if( ( app_prog_size & 0x00000003 ) != 0 ){
172          app_prog_size &= 0xFFFFFFF4;
173          app_prog_size += 4;          /* Rounds up the allocation size of the application
174                                     program to multiples of 4. */
175      }
176
177      /* Loads the application program in the large-capacity internal RAM */
178      sf_byte_read_spibsc(APROG_TOP_SFLASH, (unsigned char *)app_top_addr, app_prog_size);
179
180  }
181
182  /*****
183  * ID      :
184  * Outline  : Terminate the system
185  * Include  :
186  * Declaration : void system_down(void);
187  * Description : This function contains the infinite loop.
188  *          : As this is registered in the DUMMY_Vectors table, this is
189  *          : called when an exception occurs while the loader program
190  *          : is operating.
191  * Argument  : void
192  * Return Value: void
193  *****/
194  void system_down(void)
195  {
196      while(1){
197          /* System error */
198      }
199  }
200
201  /* End of File */

```



## 4.2 アプリケーションプログラム

### 4.2.1 アプリケーションプログラムリスト"main.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 *  System Name : SH7268/SH7269 Sample Program
31 *  File Name   : main.c
32 *  Abstract    : Sample Program Main
33 *  Version     : 1.00.00
34 *  Device      : SH7268/SH7269
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                      (Ver.9.03Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57269(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.06,2011 Ver.1.00.00
43 *****/
44 #include <stdio.h>
45 #include <string.h>
46 #include <machine.h>
47 #include "iodefine.h"
48 #include "r0k57269.h"
49
```

## 4.2.2 アプリケーションプログラムリスト"main.c" (2)

```
50  /* ==== prototype declaration ==== */
51  void main(void);
52  void io_init_cmt0(void);
53  void int_cmt_cmi0(void);
54
55  /* ==== Global variable ==== */
56  int g_led_onoff;      /* LED lighting/turning off */
57
58  /*****
59  * ID          :
60  * Outline     : main
61  *-----
62  * Include     :
63  *-----
64  * Declaration : void main(void);
65  *-----
66  * Description :
67  *-----
68  * Argument    : void
69  *-----
70  * Return Value: void
71  *-----
72  * Note        : None
73  *****/
74  void main(void)
75  {
76      puts("\nSH7269 CPU Board Sample Program. Ver.0.02.00");
77      puts("Copyright (C) 2010(2011) Renesas Electronics Corporation. All rights reserved.");
78      puts("\n");
79
80      /* ==== initial LED port ==== */
81      g_led_onoff = 1;
82      led_init();          /* LED Initialization */
83
84      /* ==== start timer ==== */
85      io_init_cmt0();     /* CMT Initialization */
86
87      /* ==== Setting of interrupt priority level ==== */
88      INTC.IPR12.BIT._CMT0 = 0x1; /* CMI Priority level of interrupt = 1 */
89
90      while(1){
91          /* loop */
92      }
93  }
94
```

## 4.2.3 アプリケーションプログラムリスト"main.c" (3)

```
95  /*****
96  * ID      :
97  * Outline : CMT0 setting
98  *-----
99  * Include : #include "iodefine.h"
100 *-----
101 * Declaration : void io_init_cmt0(void) ;
102 *-----
103 * Description : CMT0 is set as a fixed cycle of about 500ms timer.
104 *-----
105 * Argument  : void
106 *-----
107 * Return Value: void
108 *-----
109 * Note      : None
110 *****/
111 void io_init_cmt0(void)
112 {
113
114     /* ---- STBCR7 setting ---- */
115     CPG.STBCR7.BIT.MSTP72 = 0;      /* Module standby clear */
116
117     /* ==== CMT0 setting ==== */
118     /* ---- CMSTR setting ---- */
119     CMT.CMSTR.BIT.STR0 = 0;        /* Count stop */
120     /* ---- CMCSR0 setting ---- */
121     CMT.CMCSR0.WORD = 0x0043;     /* Pclock/512 */
122
123     /* ---- CMCNT0 setting ---- */
124     CMT.CMCNT0.WORD = 0x0000;     /* Timer counter clear */
125
126     /* ---- CMCOR0 setting ---- */
127     CMT.CMCOR0.WORD = 0x7f08/5;   /* 500/5=100ms */
128
129     /* ---- CMSTR setting ---- */
130     CMT.CMSTR.BIT.STR0 = 0x1;     /* Count start */
131
132 }
133
```

## 4.2.4 アプリケーションプログラムリスト"main.c" (4)

```
134  /*****
135  * ID      :
136  * Outline : CMI interrupt
137  *-----
138  * Include : #include "iodefine.h"
139  *-----
140  * Declaration : void int_cmt_cmi0(void);
141  *-----
142  * Description : The CMF flag is cleared, and the output of
143  *               : LED of each 0.5sec is reversed.
144  *-----
145  * Argument  : void
146  *-----
147  * Return Value: void
148  *-----
149  * Note      : None
150  *****/
151 void int_cmt_cmi0(void)
152 {
153
154     /* ====CMF Clearness of flag ==== */
155     CMT.CMCSR0.BIT.CMF = 0;
156
157     /* ==== PORT Reversing output(LED blinking) ==== */
158     g_led_onoff ^= 1;
159     if(g_led_onoff == 0){
160         led_on(ID_LED1);
161         led_on(ID_LED2);
162     }
163     else{
164         led_off(ID_LED1);
165         led_off(ID_LED2);
166     }
167 }
168
169 /* End of File */
```

## 4.2.5 アプリケーションプログラムリスト"appinfo.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 *  System Name : SH7268/SH7269 Sample Program
31 *  File Name   : appinfo.c
32 *  Abstract    : Generate the application program transfer information (appinfo).
33 *  Version     : 1.00.00
34 *  Device      : SH7268/SH7269
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                      (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57269(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.06,2011 Ver.1.00.00
43 *****/
```

## 4.2.6 アプリケーションプログラムリスト"appinfo.c" (2)

```
44  #include "appinfo.h"
45
46  #pragma section APPINFO
47
48  static APPINFO appinfo = {
49      __sectop("DAPPINFO"), /* Start address in the start section of the application */
50                          /* program (program area, constant area, and initialized */
51                          /* data area). */
52
53      __secend("PCACHE")   /* End address in the end section of the application */
54                          /* program (program area, constant area, and initialized */
55                          /* data area) */
56  };
57
58  /* End of File */
59
```

## 4.2.7 アプリケーションプログラムリスト"appinfo.h"

```

1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 *  System Name : SH7268/SH7269 Sample Program
31 *  File Name   : appinfo.h
32 *  Abstract    : Header file of the application program transfer information (appinfo).
33 *  Version     : 1.00.00
34 *  Device      : SH7268/SH7269
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                               (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57269(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.06,2011 Ver.1.00.00
43 *****/
44 #ifndef __APPINFO_H__
45 #define __APPINFO_H__
46
47 typedef struct appinfo_t {
48     void *app_top;           /* Start address of the application program */
49     void *app_end;          /* End address of the application program */
50 } APPINFO;
51
52 #endif /* __APPINFO_H__ */
53
54 /* End of File */

```

## 4.3 ダウンローダ

### 4.3.1 ダウンローダプログラムリスト"downloader.hdc" (1)

```
1      #/*****
2      #*  DISCLAIMER
3      #*
4      #*  This software is supplied by Renesas Electronics Corporation and is only
5      #*  intended for use with Renesas products. No other uses are authorized.
6      #*
7      #*  This software is owned by Renesas Electronics Corporation and is protected under
8      #*  all applicable laws, including copyright laws.
9      #*
10     #*  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11     #*  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12     #*  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13     #*  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14     #*  DISCLAIMED.
15     #*
16     #*  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17     #*  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18     #*  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19     #*  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20     #*  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21     #*
22     #*  Renesas reserves the right, without notice, to make changes to this
23     #*  software and to discontinue the availability of this software.
24     #*  By using this software, you agree to the additional terms and
25     #*  conditions found by accessing the following link:
26     #*  http://www.renesas.com/disclaimer
27     #*****
28     #*  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29     #***** Technical reference data *****
30     #*  System Name : SH7268/SH7269 Firm Update Sample Program
31     #*  File Name   : downloader.hdc
32     #*  Abstract    : ダウンローダ用パッチファイル
33     #*  Version     : 1.00.00
34     #*  Device      : SH7269/SH7269
35     #*  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36     #*                : C/C++ compiler package for the SuperH RISC engine family
37     #*                :                               (Ver.9.03Release02).
38     #*  OS          : None
39     #*  H/W Platform: R0K57269(CPU board)
40     #*  Description :
41     #*****
42     #*  History     : Jul.06,2011 Ver.1.00.00
43     #*"FILE COMMENT END"*****/
44
45
46     tcl enable
47
48
```



## 4.3.2 ダウンローダプログラムリスト"downloader.hdc" (2)

```
49 #Macro downloader -Start
50 proc init_hardware {} {
51
52     # CPG の設定
53     MF H'FFFE0010 H'FFFE0011 H'1015 WORD
54
55     # SYSCR5 の設定
56     MF H'FFFE0428 H'FFFE0428 H'0F BYTE
57 }
58
59
60 proc downloader {} {
61     # CPU リセット
62     reset
63
64     # init_hardware ルーチンの呼び出し
65     init_hardware
66
67     # High-performance Embedded Workshop に登録されている[Download modules]を全てダウンロード
68     file_load_all
69
70     # ユーザスタックの使用許可 (ソフトウェアブレークポイントを使用するため)
71     sh2a_sbstk enable
72
73     # _halt(main.c 参照)にソフトウェアブレークポイントを設定
74     set_disassembly_soft_break _halt set
75
76     # _error(main.c 参照)にソフトウェアブレークポイントを設定
77     set_disassembly_soft_break _error set
78
79     # _downloader(downloader.src 参照)を実行し、停止するまで待つ
80     go wait _downloader
81
82     # _halt に設定されているソフトウェアブレークポイントを解除
83     set_disassembly_soft_break _halt clear
84
85     # _error に設定されているソフトウェアブレークポイントを解除
86     set_disassembly_soft_break _error clear
87
88 }
89
90 downloader
91 #Macro downloader -End
92
93
94
95 # 【注】 "tcl", "reset", "file_load", "sh2a_sbstk", "set_disassembly_soft_break", "go"は、
96 #High-performance Embedded Workshop および E10A-USB エミュレータのコマンドです。コマンドの詳細は、
97 #マニュアルを参照してください。
98
99 # /* End of File */
```

## 4.3.3 ダウンロードプログラムリスト"dl\_entry.src" (1)

```
1      ;/*****
2      ;*   DISCLAIMER
3      ;*
4      ;*   This software is supplied by Renesas Electronics Corporation and is only
5      ;*   intended for use with Renesas products. No other uses are authorized.
6      ;*
7      ;*   This software is owned by Renesas Electronics Corporation and is protected under
8      ;*   all applicable laws, including copyright laws.
9      ;*
10     ;*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11     ;*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12     ;*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13     ;*   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14     ;*   DISCLAIMED.
15     ;*
16     ;*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17     ;*   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18     ;*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19     ;*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20     ;*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21     ;*
22     ;*   Renesas reserves the right, without notice, to make changes to this
23     ;*   software and to discontinue the availability of this software.
24     ;*   By using this software, you agree to the additional terms and
25     ;*   conditions found by accessing the following link:
26     ;*   http://www.renesas.com/disclaimer
27     ;*****
28     ;*   Copyright (C) 2010(2011) Renesas Electronics Corporation. All rights reserved.
29     ;***** Technical reference data *****
30     ;*   System Name : SH7268/SH7269 Firm Update Sample Program
31     ;*   File Name   : dl_entry.src
32     ;*   Abstract    : downloader start up
33     ;*   Version     : 0.03.00
34     ;*   Device      : SH7268/SH7269
35     ;*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36     ;*                : C/C++ compiler package for the SuperH RISC engine family
37     ;*                :                               (Ver.9.03Release02).
38     ;*   OS          : None
39     ;*   H/W Platform: R0K57269(CPU board)
40     ;*   Description :
41     ;*****
42     ;*   History     : Sep.06,2010 Ver.0.01.00
43     ;*                : Apr.27,2011 Ver.0.02.00 change comment
44     ;*                : May.08,2011 Ver.0.03.00 change comment
45     ;* "FILE COMMENT END" *****/
```

## 4.3.4 ダウンローダプログラムリスト"dl\_entry.src" (2)

```
46     .SECTION ENTRY, CODE, ALIGN = 4
47     .IMPORT _PowerON_Reset_PC
48
49     _downloader:
50     MOV.L STACK_POINTER, R15      ; setting stack pointer
51     MOV.L MAIN_PROGRAM, R0       ; get entry address of downloader
52     JMP @R0                      ; jump to entry address of downloader
53     NOP
54
55     .ALIGN 4
56     MAIN_PROGRAM:
57     .DATA.L _PowerON_Reset_PC   ; entry address of downloader
58
59     STACK_POINTER:
60     .DATA.L H'FFF84000         ; stack pointer of downloader(R15)
61
62     .pool
63     .end
64
65     ;/* End of File */
```

## 4.3.5 ダウンロードプログラムリスト"dl\_main.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 *  System Name : SH7268/SH7269 Firm Update Sample Program
31 *  File Name   : dl_main.c
32 *  Abstract    : downloader main
33 *  Version     : 1.00.00
34 *  Device      : SH7268/SH7269
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                      (Ver.9.03 Release02).
38 *  OS          : None
39 *  H/W Platform: R0K57269(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.06,2011 Ver.1.00.00
43 *****/
44 #include <stdio.h>
45 #include <string.h>
46 #include <machine.h>
47 #include "iodefine.h"
48 #include "serial_flash.h"
49
```

## 4.3.6 ダウンロードプログラムリスト"dl\_main.c" (2)

```
50  /* ==== macro defined ==== */
51  #define SECTOR_SIZE      SF_SECTOR_SIZE      /* Sector size */
52  #define SECTOR_NUM      SF_NUM_OF_SECTOR    /* Total number of sectors in the device */
53  #define DEVICE_SIZE      (SECTOR_SIZE * SECTOR_NUM) /* Device size */
54
55  #if (SFLASH_DUAL == 0)
56  #define L_PROG_SIZE      8192                /* Loader program size */
57  #else
58  #define L_PROG_SIZE      4096                /* Loader program size */
59  #endif
60  #define L_PROG_SRC        0xFFFF8000        /* Source address of the loader program */
61  #define L_PROG_DST        0x00000000        /* Destination address of the loader program */
62
63  #define APROG_TOP_SFLASH  0x00002000        /* Start address of the application program */
64
65  #define APROG_TOP_RAM     0x1C002000        /* Start address of the application program */
66  /* When changing the start section of the */
67  /* application program, change this definition */
68
69  #define APPINFO_TOP       APROG_TOP_RAM      /* Address the appinfo.app_top is located */
70  #define APPINFO_END      ( APROG_TOP_RAM + 4 ) /* Address the appinfo.app_end is located */
71
72  /* ==== prototype declaration ==== */
73  /*** User API ***/
74  void main(void);
75
76  static void halt(void);
77  static void error(void);
78  static void init_erase_flag(void);
79  static int Is_erased_sector(unsigned long sector_no);
80  static void set_erase_flag(unsigned long sector_no);
81  static int write_prog_data(unsigned char *program_data, unsigned long sflash_addr,
82  unsigned long size);
83
84  /*** data ***/
85  static unsigned char sflash_erase_flag[SECTOR_NUM]; /* 0:sector not erased,1:sector erased */
86
```

## 4.3.7 ダウンローダプログラムリスト"dl\_main.c" (3)

```
87  /*****
88  * ID      :
89  * Outline : Downloader main
90  * Include :
91  * Declaration : void main(void);
92  * Description : Writes the loader program and application program in serial
93  *              : flash memory as the following procedures.
94  *              : 1. Mask the interrupt while the downloader is operating.
95  *              : 2. Initialize the RSPI0.
96  *              : 3. Disable the software protection in serial flash memory.
97  *              : 4. Write the loader program in serial flash memory.
98  *              : 5. Write the application program in serial flash memory.
99  * Argument : void
100 * Return Value: void
101 *****/
102 void main(void)
103 {
104     unsigned long app_top_addr,app_end_addr,app_prog_size;
105
106
107     /* Initializes the erase flag */
108     init_erase_flag();
109
110     /* Initializes the SPIBSC */
111     sf_init_serial_flash_spibsc();
112
113     /* Disables the software protection in serial flash memory */
114     sf_protect_ctrl_spibsc(SF_REQ_UNPROTECT);
115
116     /* Writes the loader program */
117     if( write_prog_data( (unsigned char *)L_PROG_SRC, L_PROG_DST, L_PROG_SIZE) < 0 ){
118         error();
119     }
120
121     /* Retrieves the start address and end address from the application program
122        transfer information (appinfo) */
123     app_top_addr = *(volatile unsigned long *)APPINFO_TOP;
124     app_end_addr = *(volatile unsigned long *)APPINFO_END;
125
126     /* Calculates the size of the application program */
127     app_prog_size = app_end_addr - app_top_addr;
128
```

## 4.3.8 ダウンローダプログラムリスト"dl\_main.c" (4)

```
129     /* Writes the application program */
130     if( write_prog_data( (unsigned char *)APROG_TOP_RAM, APROG_TOP_SFLASH, app_prog_size) < 0 ){
131         error();
132     }
133
134     /* Enables the software protection in serial flash memory */
135     sf_protect_ctrl_spibsc(SF_REQ_PROTECT);
136
137     /* Exits the downloader */
138     halt();
139 }
140
```

## 4.3.9 ダウンロードプログラムリスト"dl\_main.c" (5)

```
141  /*****
142  * ID      :
143  * Outline : Write the program data
144  * Include :
145  * Declaration : int write_prog_data(unsigned char *program_data,
146  *                unsigned long sflash_addr, unsigned long size);
147  * Description : Writes the program data as the following procedures.
148  *              : 1. Erase the target sector when it is not erased.
149  *              : 2. Write the program data in serial flash memory.
150  *              : 3. Reads the data in serial flash memory and compare it with the
151  *              :   provided data.
152  * Argument  : unsigned char *program_data ; I : Start address of the program data
153  *            : unsigned long sflash_addr  ; I : Start address at the destination in
154  *            :                          serial flash memory
155  *            : unsigned long size         ; I : Write size
156  * Return Value: Equal or bigger than 0: Success
157  *            : Less than 0: Error
158  *****/
159  int write_prog_data(unsigned char *program_data, unsigned long sflash_addr, unsigned long size)
160  {
161      unsigned long sector_no;
162      unsigned long saddr;
163      unsigned long sz;
164      unsigned char read_data[2];
165      unsigned char *w_p;
166      int wr_size;
167      int rd_size;
168      int bsz;
169
170      bsz = 1;
171
172      /* ==== Copies the value from the argument to the local variable ==== */
173      saddr = sflash_addr;
174      sz = size;
175      w_p = program_data;
176
177      /* ==== Writes data in serial flash memory ==== */
178      while( sz > 0){
179          if(sz > ((256 * bsz) - (saddr % (256 * bsz)))){
180              wr_size = (int)((256 * bsz) - (saddr % (256 * bsz)));
181          }
182          else{
183              wr_size = (int)sz;
184          }
185
186          sector_no = saddr / (SECTOR_SIZE * bsz);
```



## 4.3.10 ダウンロードプログラムリスト"dl\_main.c" (6)

```
187
188     if( Is_erased_sector(sector_no) == 0 ){ /* When it is not erased */
189         sf_sector_erase_spibsc(sector_no); /* Erase */
190         set_erase_flag(sector_no);      /* When it is erased, set the erase flag */
191     }
192
193     sf_byte_program_spibsc(saddr, w_p, wr_size );
194                                     /* Writes data in units of */
195                                     /* single byte */
196     w_p += wr_size;
197     saddr += wr_size;
198     sz -= wr_size;
199 }
200
201 /* ==== Verifies data (serial flash memory is programmed successfully) ==== */
202 saddr = sflash_addr;
203 sz = size;
204 w_p = program_data;
205
206 rd_size = 1;
207 while( sz > 0){
208     sf_byte_read_spibsc(saddr,read_data, rd_size);
209                                     /* Reads the data written in */
210                                     /* serial flash memory */
211
212     if( w_p[0] != read_data[0] ){
213         return -1; /* Returns an error when the data */
214                   /* unmatched */
215     }
216     w_p += rd_size;
217     saddr += rd_size;
218     sz -= rd_size;
219 }
220
221 return 0;
222 }
223
```

## 4.3.11 ダウンロードプログラムリスト"dl\_main.c" (7)

```
224 /*****
225  * ID      :
226  * Outline  : Initialize the Erase Flag
227  * Include  :
228  * Declaration : static void init_erase_flag(void);
229  * Description : Initializes the table sflash_erase_flag[].
230  * Argument   : void
231  * Return Value: void
232  *****/
233 static void init_erase_flag(void)
234 {
235     int i;
236
237     for( i=0; i < SECTOR_NUM ;i++){
238         sflash_erase_flag[i] = 0;
239     }
240 }
241
242 /*****
243  * ID      :
244  * Outline  : Retrieve the Sector Erase Status
245  * Include  :
246  * Declaration : static int Is_erased_sector(unsigned long sector_no);
247  * Description : Returns the information (not erased or eraser) of the
248  *              : sector specified by the sector number.
249  * Argument   : unsigned long sector_no ; I : Sector number
250  * Return Value: 1 : Sector in the specified address is already erased
251  *              : 0 : Sector in the specified address is not erased
252  *****/
253 static int Is_erased_sector(unsigned long sector_no)
254 {
255     return sflash_erase_flag[sector_no];
256 }
257
258 /*****
259  * ID      :
260  * Outline  : Set the Erase Flag
261  * Include  :
262  * Declaration : static void set_erase_flag(unsigned long sector_no);
263  * Description : Sets the erase flag to modify the information of the specified
264  *              : sector as erased.
265  * Argument   : unsigned long sector_no ; I : Sector number
266  * Return Value: void
267  *****/
268 static void set_erase_flag(unsigned long sector_no)
269 {
270     sflash_erase_flag[sector_no] = 1;
271 }
272
```

## 4.3.12 ダウンローダプログラムリスト"dl\_main.c" (8)

```
273  /*****
274  * ID      :
275  * Outline  : Program stops (successful).
276  * Include  :
277  * Declaration : static void halt(void);
278  * Description : When the downloader ends successfully, this function is called
279  *           : to stop the program.
280  * Argument  : void
281  * Return Value: void
282  *****/
283  static void halt(void)
284  {
285      while(1){
286          /* When the downloader ends successfully, this function stops the program. */
287      }
288  }
289
290  /*****
291  * ID      :
292  * Outline  : Program stops (error).
293  * Include  :
294  * Declaration : static void error(void);
295  * Description : When the downloader ends in error, this function is called
296  *           : to stop the program.
297  * Argument  : void
298  * Return Value: void
299  *****/
300  static void error(void)
301  {
302      while(1){
303          /* When the downloader ends in error, this function stops the program */
304      }
305  }
306
307  /* End of File */
308
```

## 5. ダウンローダの使用法

応用例のダウンローダは、High-performance Embedded Workshop と E10A-USB エミュレータの組み合わせで動作します。その他の開発環境を使用する場合は、使用する環境に合わせてプログラムの変更を行ってください。

シリアルフラッシュメモリへのプログラム書き込みは、High-performance Embedded Workshop の[デバッグ]メニュー->ダウンロードからダウンロードモジュールを選択しても行うことはできません。ここでは、応用例のダウンローダを使用して、シリアルフラッシュメモリへのプログラムを書き込む手順について説明します。

### 5.1 参考プログラムの構成

参考プログラムは、表 11 のように 3 つのワークスペースに分かれています。

表 11 参考プログラムの構成

ワークスペース名	説明
sh7269_spibsc_downloader	このワークスペースのプロジェクトでは、ダウンローダをビルドします。
sh7269_spibsc_loader_prog	このワークスペースのプロジェクトでは、ローダプログラムをビルドします。
sh7269_spibsc_app	このワークスペースのプロジェクトでは、アプリケーションプログラムをビルドします。また、このワークスペースのプロジェクトには、[sh7269_spibsc_downloader]ワークスペースで作成したダウンローダ、ダウンローダ起動用のバッチファイル、[sh7269_spibsc_loader_prog]ワークスペースで作成したローダプログラムが登録されています。これらを使用してシリアルフラッシュメモリにローダプログラムとアプリケーションプログラムを書き込みます。

## 5.2 シリアルフラッシュメモリへのプログラムの書き込み方法

ここでは、[sh7269\_spibsc\_app]ワークスペースを使用して、ローダプログラムとアプリケーションプログラムをシリアルフラッシュメモリに書き込む方法について説明します。

### 5.2.1 ダウンロードモジュールとバッチファイルの登録

図 15に[sh7269\_spibsc\_app]ワークスペースのディレクトリ構成を示します。[sh7269\_spibsc\_app]ワークスペースでは、図 15の①、②、③、⑤のダウンロードモジュールと④のバッチファイルをプロジェクトに登録しています。なお①と②のダウンロードモジュールの登録には注意が必要です。SPIマルチI/Oバス空間にはデータを直接ダウンロードできないため、①のabsファイルはデバッグ情報のみダウンロードし、②のmotファイルはオフセット指定を使用して大容量内蔵RAM上にダウンロードしています。

図 16および図 17に参考プログラムで使用するダウンロードモジュールの登録例を示します。

```

¥sh7269_spibsc_app          : ワークスペースディレクトリ
|-sh7269_spibsc_app         : プロジェクトディレクトリ
|  |-debug                  :
|  |  |-sh7269_spibsc_app.abs : アプリケーションプログラムの実行ファイル1-----①
|  |  |-sh7269_spibsc_app.mot : アプリケーションプログラムの実行ファイル2-----②
|  |
|  |
|-inc                       : 共通インクルードファイル格納用ディレクトリ
|-src                       : ソースファイル格納用ディレクトリ
|-sflash_boot              : ダウンローダおよびローダプログラム格納用ディレクトリ
|  |-sh7269_spibsc_downloader.abs : ダウンローダの実行ファイル-----③
|  |-downloader.hdc        : バッチファイル(ダウンローダ起動用)-----④
|  |-sh7269_spibsc_loader_prog.abs : ロードプログラムの実行ファイル-----⑤

```

図15 [sh7269\_spibsc\_app]ワークスペースのディレクトリ構成

#### 1. ダウンロードモジュールの変更方法

プロジェクトに登録されているダウンロードモジュールを変更する場合は、[デバッグの設定]ダイアログボックスで設定を変更します。[デバッグの設定]ダイアログボックスは、High-performance Embedded Workshopの[デバッグ]メニュー→[デバッグの設定]を選択して開きます。

登録方法については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

#### 2. バッチファイルの変更方法

プロジェクトに登録されているバッチファイルを変更する場合は、[バッチファイルを指定]ダイアログボックスで設定を変更します。次の操作を行うと、[バッチファイルを指定]ダイアログボックスが開きます。High-performance Embedded Workshop の[表示]メニュー→[コマンドライン]を選択して[コマンドライン]ウィンドウを表示します。[バッチファイルを指定] ダイアログボックスは、[コマンドライン]ウィンドウのポップアップメニューから[バッチファイル指定]ボタンをクリックして開きます。

登録方法については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

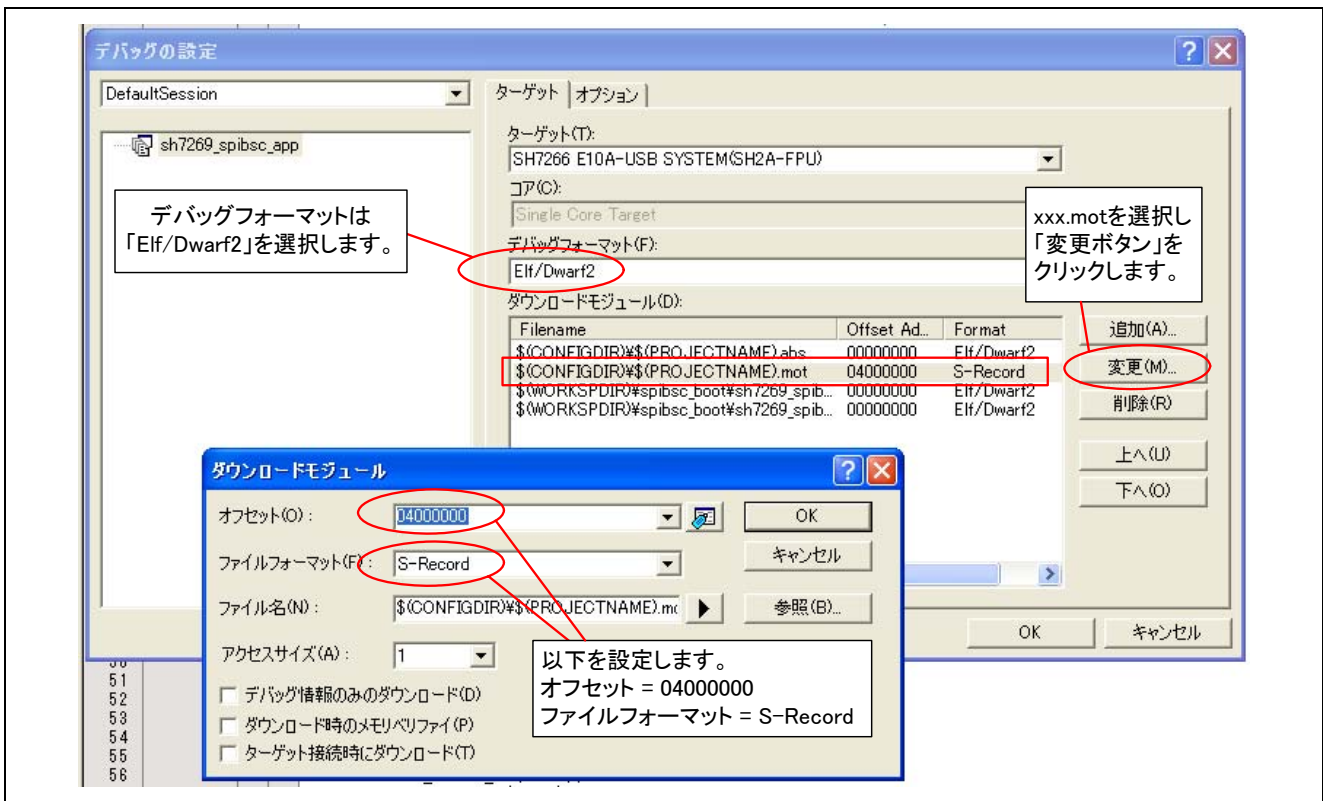


図16 ダウンロードモジュールの登録例 (mot ファイル)

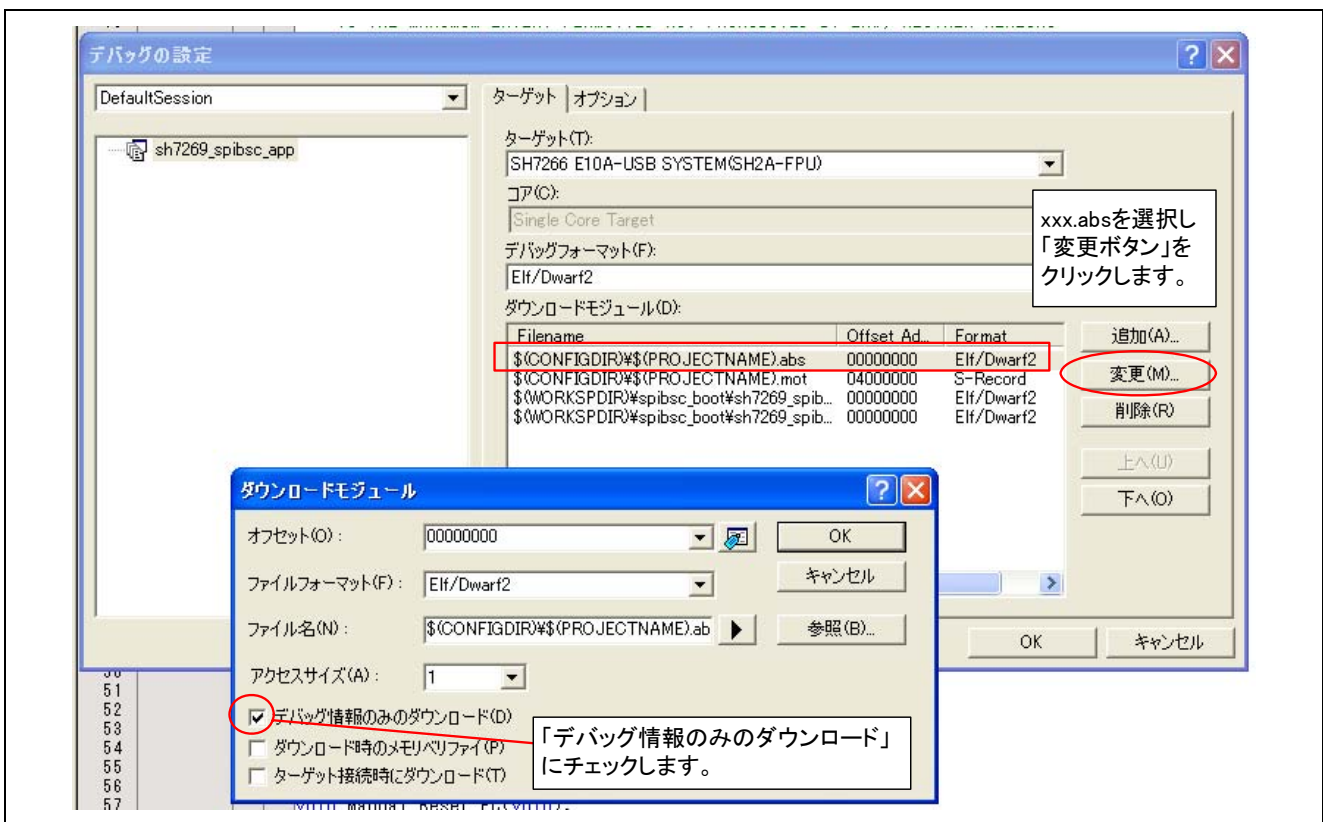


図17 ダウンロードモジュールの登録例 (abs ファイル)

### 5.2.2 プログラム書き込み手順

ここでは、[sh7269\_spibsc\_app]ワークスペースを使用してシリアルフラッシュメモリにローダプログラムとアプリケーションプログラムを書き込む手順を説明します。

1. [sh7269\_spibsc\_app]ワークスペースディレクトリを C:\¥WorkSpace にコピーしてください。
2. ワークスペースディレクトリの中にある[sh7269\_spibsc\_app].hws をダブルクリックしてください。  
High-performance Embedded Workshop が起動します。
3. High-performance Embedded Workshop の[ビルド]メニュー→[すべてをビルド]を選択し、ビルドを行ってください。アプリケーションプログラムが生成されます。
4. High-performance Embedded Workshop の[デバッグ]メニュー→[接続]を選択し、ターゲットとの接続を確立してください。
5. 接続確立後、High-performance Embedded Workshopの[表示]メニュー→[コマンドライン]を選択し、図 18 に示すように[コマンドライン]ウィンドウを表示してください。
6. [コマンドライン]ウィンドウの[バッチファイルの実行]ボタンをクリックして、登録されているバッチファイル[downloader.hdc]を実行してください。

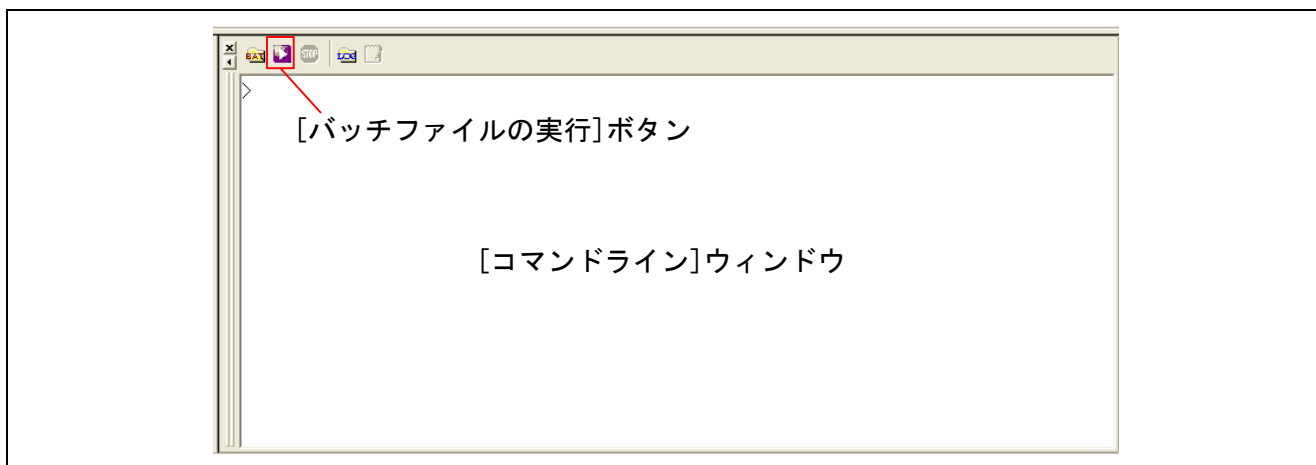


図18 [コマンドライン]ウィンドウと[バッチファイルの実行]ボタン

7. バッチファイル[downloader.hdc]が実行されると、ワークスペースに登録されている全てのダウンロードモジュール（ローダプログラム、アプリケーションプログラム、ダウンローダ）がRAM上に転送され、ダウンローダが実行されます。図 19のようにダウンローダが正常終了すると、プログラムカウンタは\_haltで停止し、書き込みに失敗した場合は\_errorで停止します。[sh7269\_spibsc\_downloader]ワークスペースディレクトリをC:\¥Workspaceにコピーしている場合は、ソースファイルが表示されることがあります。
8. 正常に書き込みが完了した場合は、リセット後実行によ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を実行できます。

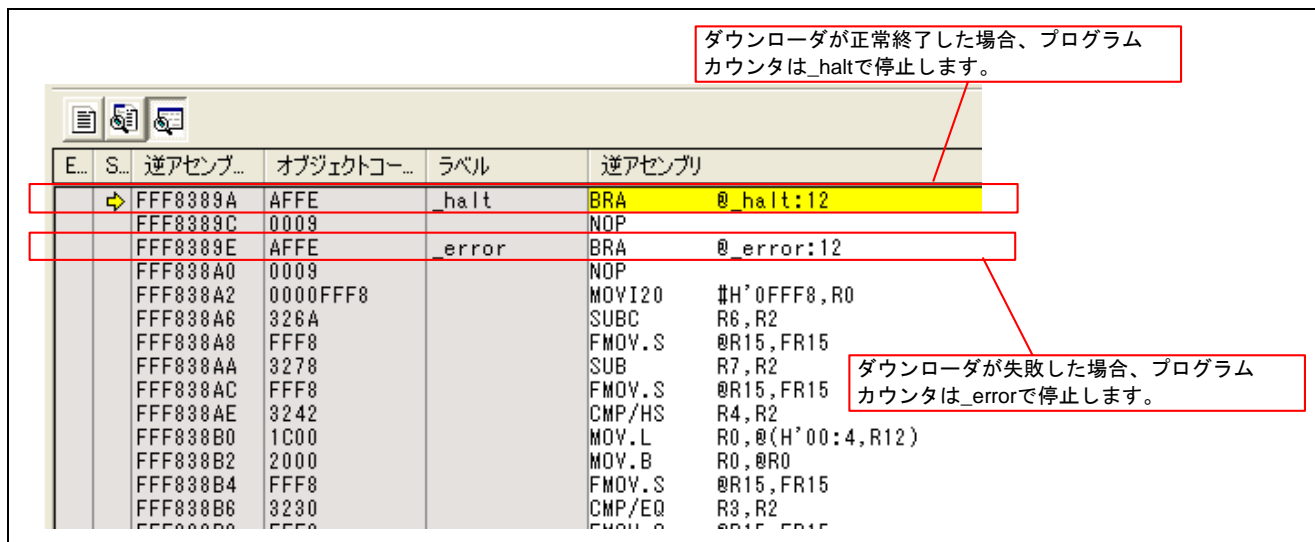


図19 ダウンローダ終了時の High-performance Embedded Workshop の画面



## 6. 参考ドキュメント

- ソフトウェアマニュアル  
SH-2A/SH-2A-FPU ソフトウェアマニュアル Rev.3.00  
(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。)
- ハードウェアマニュアル  
SH7268 グループ、SH7269 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。)

## 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>

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。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2011.07.11	—	初版発行
1.01	2012.02.16	—	SH726B の参考プログラムを追加

## 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っていると、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>