
SH7266/SH7267 グループ

R01AN0214JJ0101

Rev. 1.01

2011.03.31

シリアルフラッシュメモリからのブート例

要旨

本アプリケーションノートは、SH7266/SH7267 の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

SH7267

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.....	29
5. ダウンローダの使用法.....	54
6. 参考ドキュメント.....	57

1. はじめに

1.1 仕様

SH7267 はブートモード 1 および 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）。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）

1.3 適用条件

マイコン	SH7266/SH7267
動作周波数	内部クロック：144 MHz バスクロック：72 MHz 周辺クロック：36 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)

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。

- SH7266/SH7267 グループ ルネサスクワッドシリアルペリフェラルインタフェース
シリアルフラッシュメモリ接続例

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	ブート起動用内蔵 ROM プログラムは、ブートモード 1,3 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	ローダプログラムは、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上から RAM に転送し、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うプログラムです。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	アプリケーションプログラムは、お客様がシステムに応じて作成するプログラムです。
ダウンローダ	ダウンローダは、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_BOOT1~0) 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_BOOT1	MD_BOOT0	ブートモード	説明
0	1	ブートモード1	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低速通信でブートします。 低速通信：バスクロック(B ϕ)の1/4の速度で通信
1	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高速通信でブートします。 高速通信：バスクロック(B ϕ)の1/2の速度で通信

ブートモード1または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。

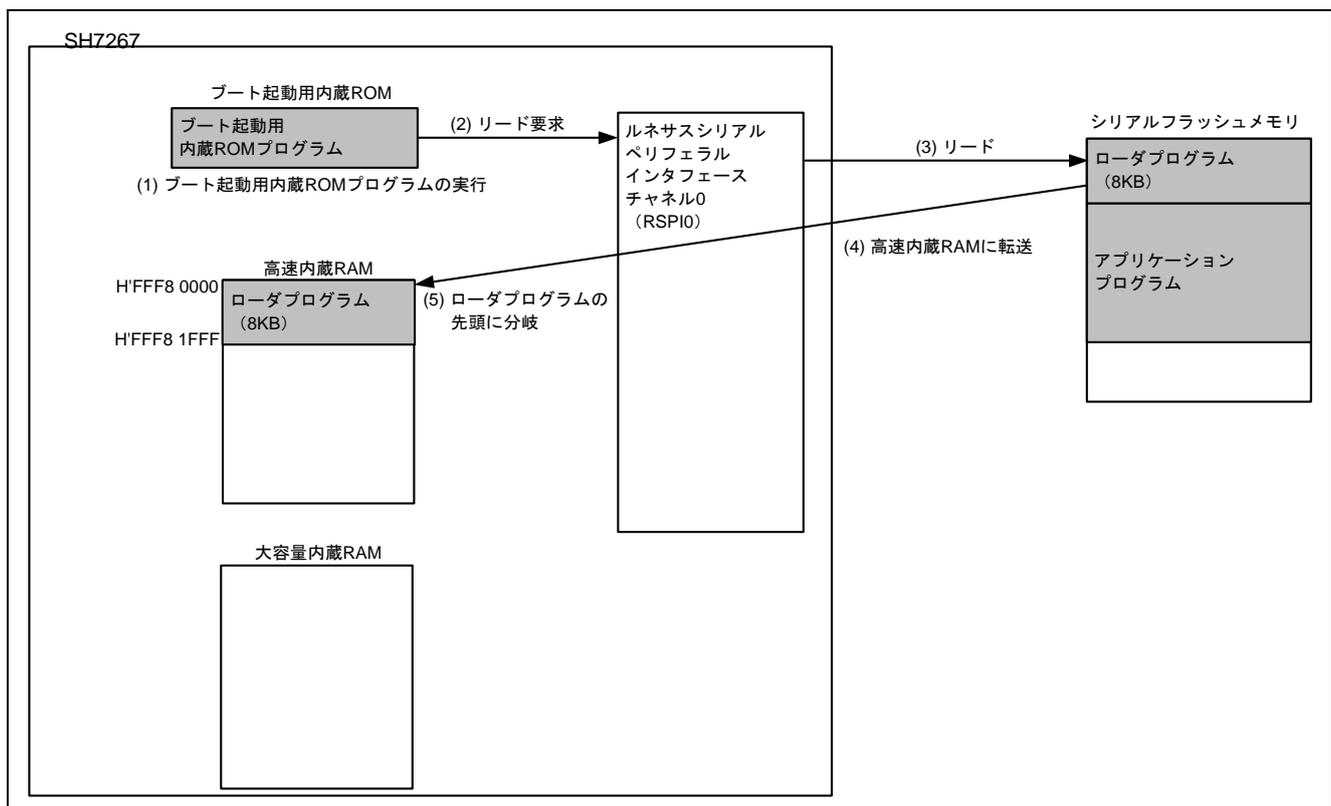


図1 ブート起動用内蔵ROMプログラムの動作イメージ図

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 (RSPI0) に接続されている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大容量内蔵RAMに転送します。転送完了後、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。

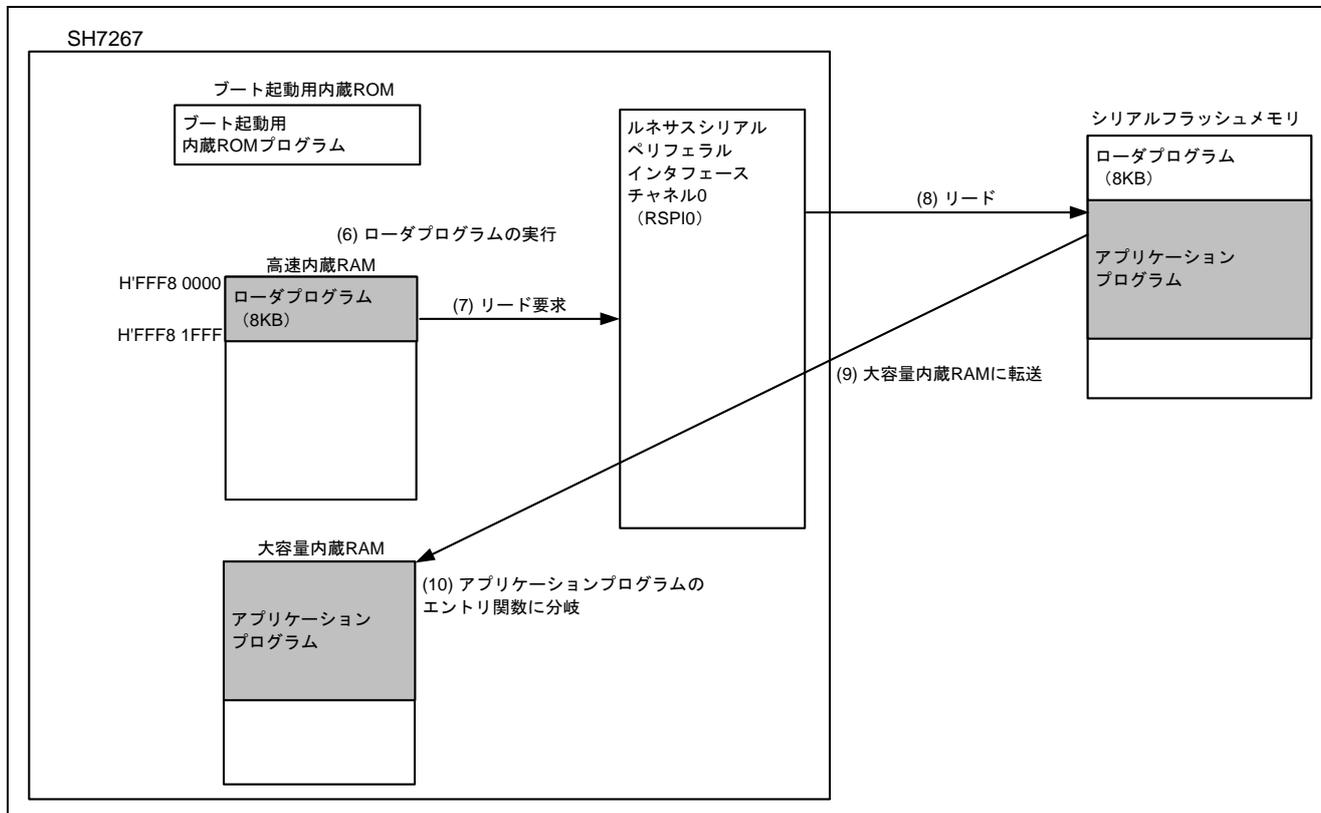


図2 ローダプログラムの動作イメージ図

【注】 ローダプログラムを変更することで、アプリケーションプログラムをSDRAM等の外部RAMに転送することも可能です。

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。

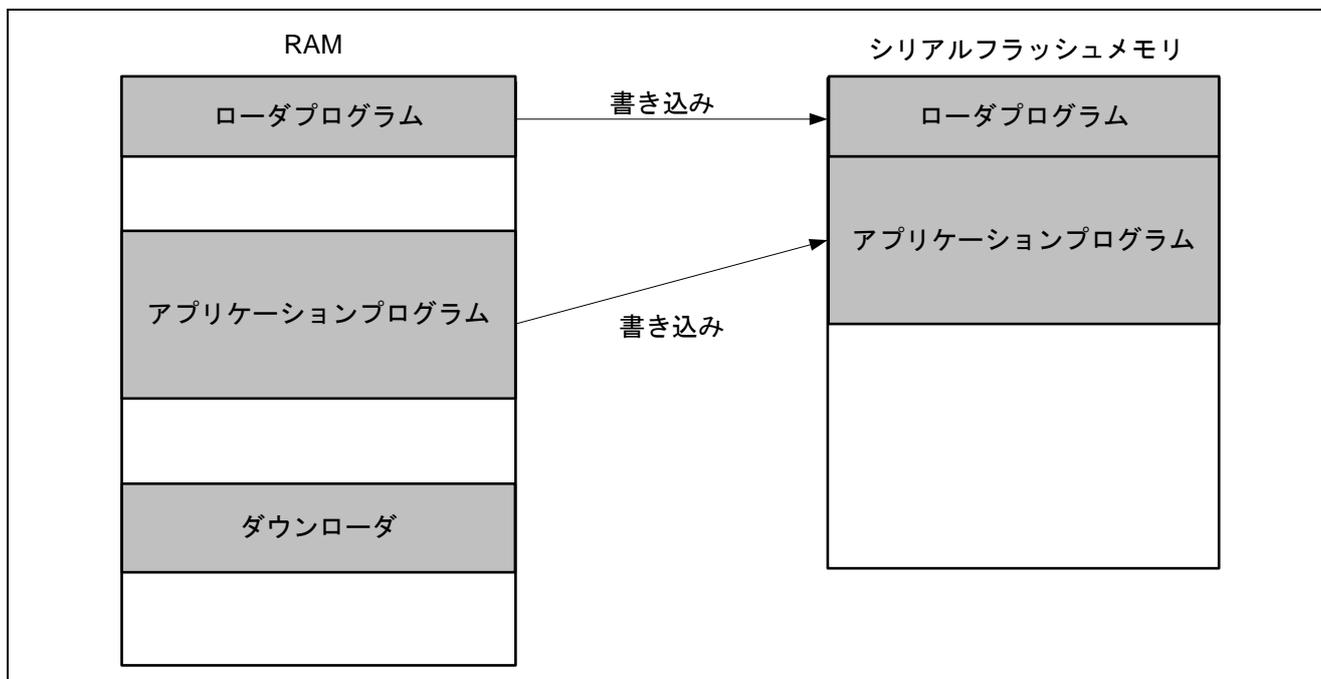


図3 ダウンローダの動作イメージ図

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を使用する場合の接続例を示します。シリアルフラッシュブート機能を使用する場合は、SPI インタフェースのシリアルフラッシュメモリを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) に接続します。

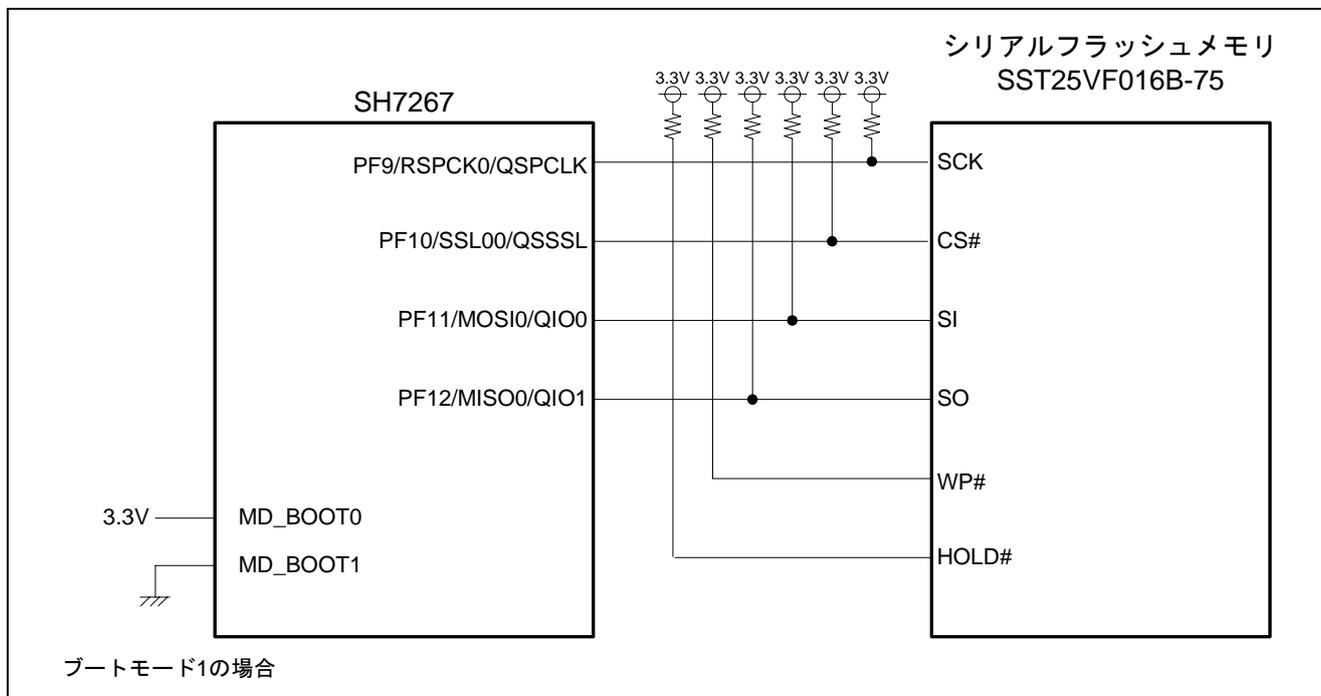


図4 シリアルフラッシュブートを行うための接続例

【注】 ブートモード1の場合はバスクロック (B ϕ) の2分の1のRSPIOクロックを使用し、ブートモード3の場合はB ϕ の4分の1のRSPIOクロックを使用します。シリアルフラッシュメモリとRSPIOのAC特性を満たすようにブートモードを選択してください。

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、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から大容量内蔵 RAM に転送し、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。

1. ローダプログラム（プログラム領域）は、H'FFF8 0000～H'FFF8 1AFF 番地を使用します。
2. 仮の例外処理ベクタテーブルは、H'FFF8 1B00～H'FFF8 1B4F 番地を使用します（詳細は3.1.5を参照）。
3. ローダプログラムのスタック領域は、H'FFF8 1C00～H'FFF8 1FFF 番地を使用します（詳細は3.1.3を参照）。

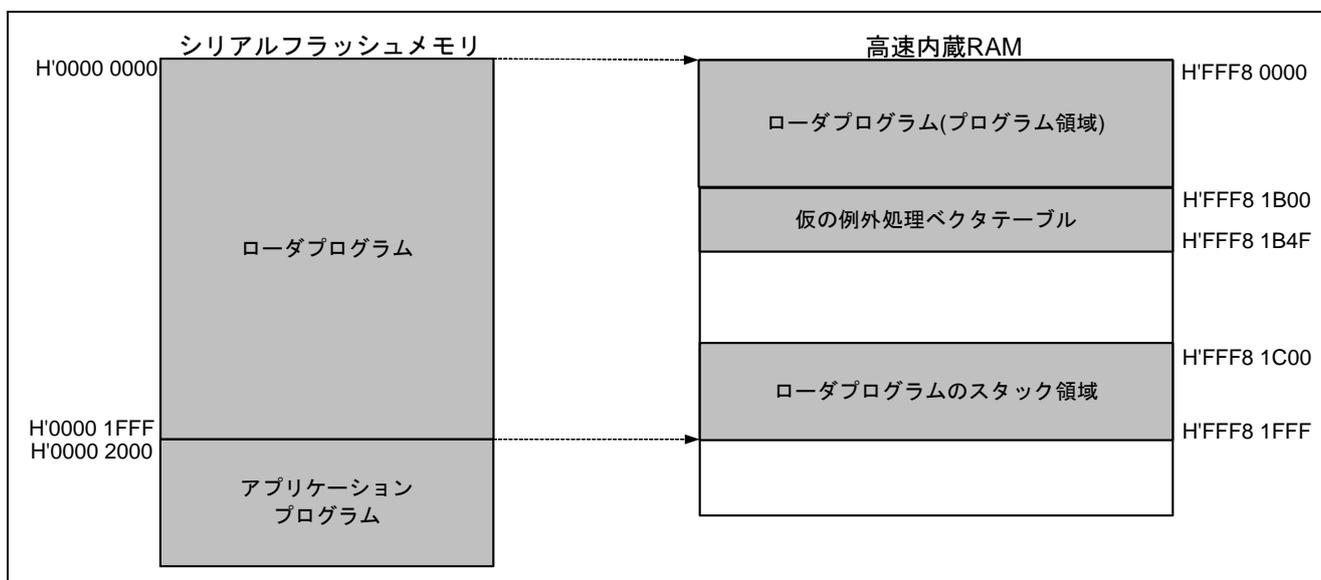


図5 ローダプログラムのメモリマップ

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。

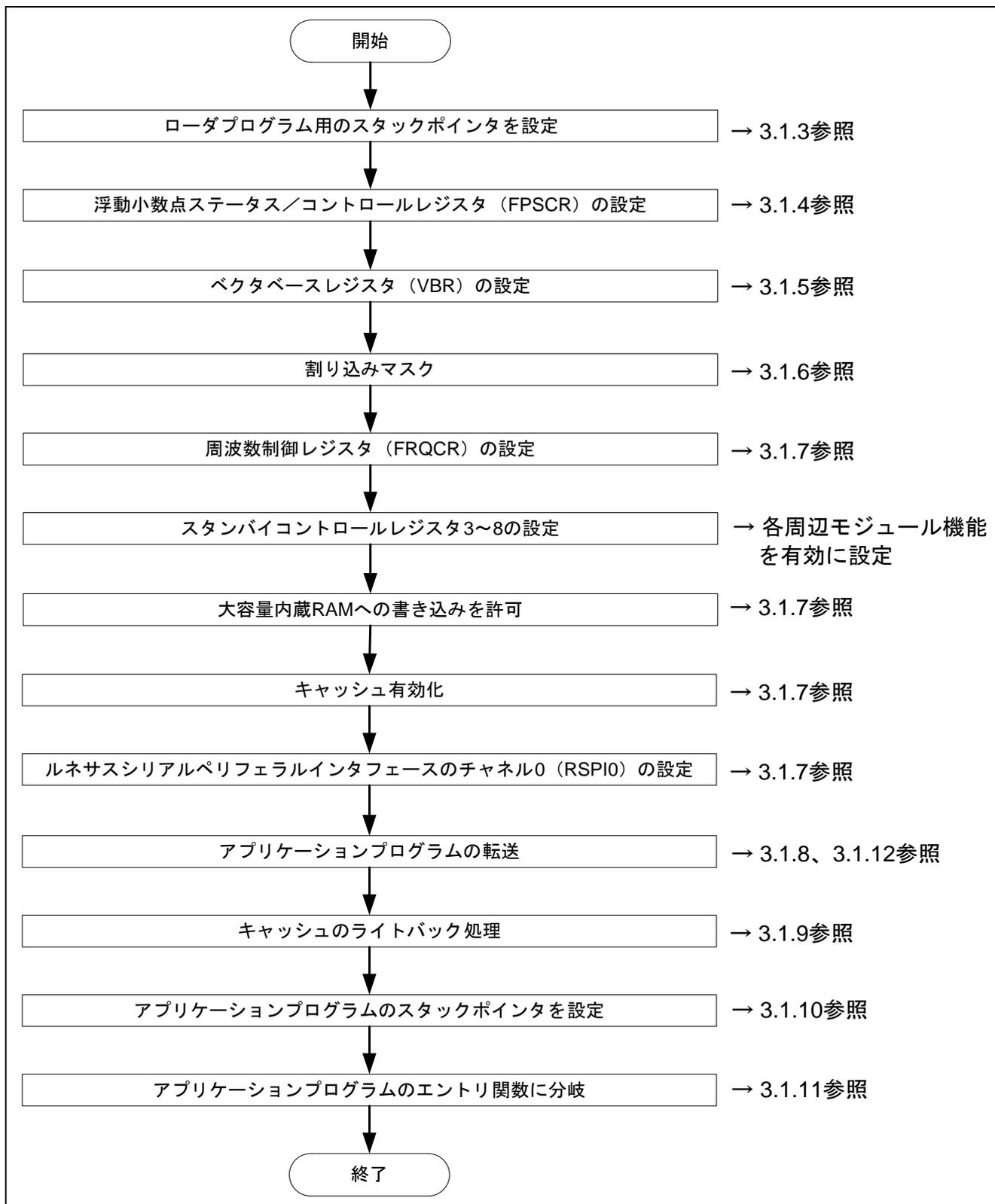


図6 ローダプログラムのフローチャート

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 SH7266 グループ、SH7267 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の転送

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大容量内蔵 RAM に転送します。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。

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

図 7 にアプリケーションプログラム転送情報を用いた転送イメージ図を示します。アプリケーションプログラム転送情報 (appinfo) の生成方法については3.2.7を参照してください。

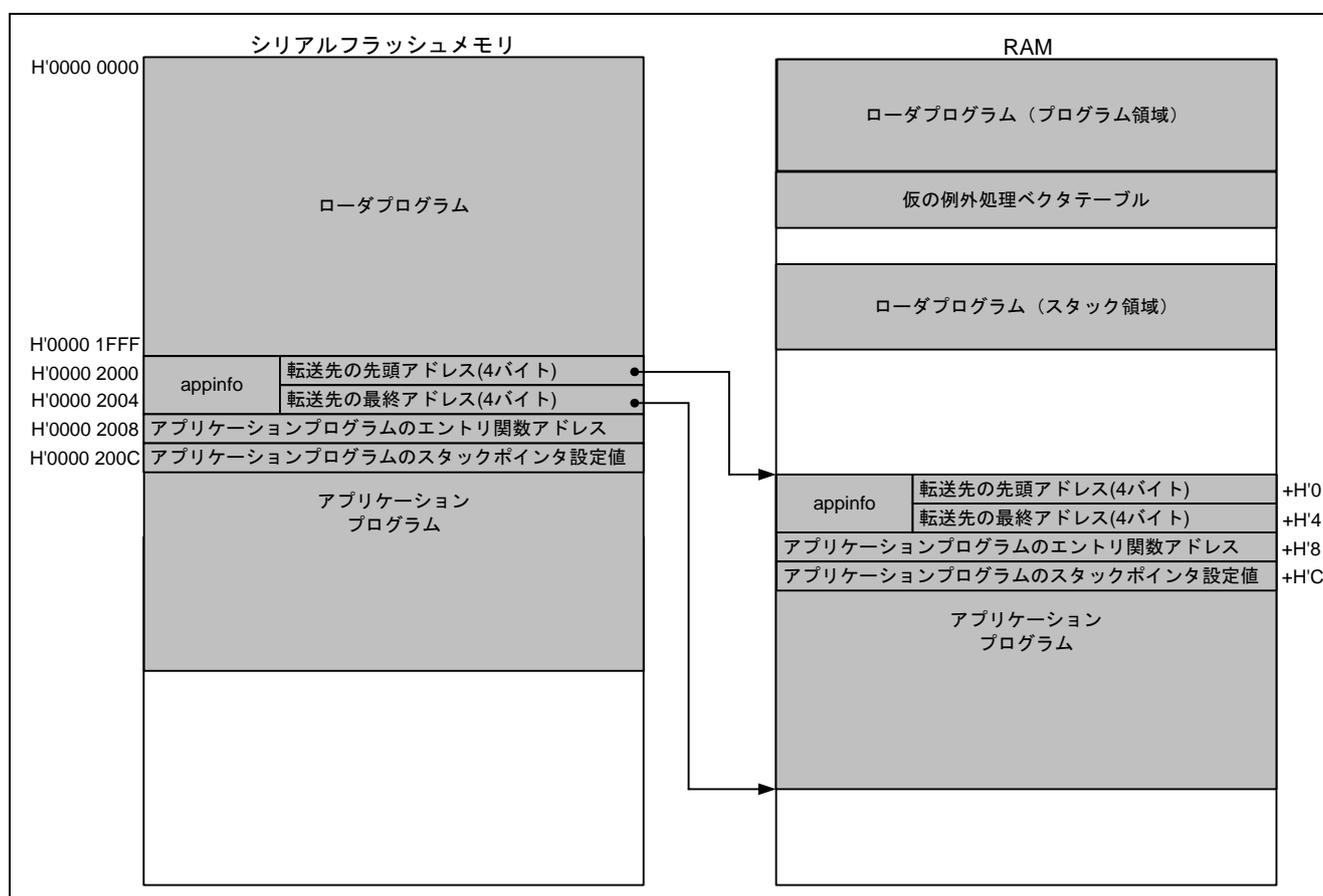


図7 アプリケーションプログラム転送イメージ図

3.1.9 キャッシュのライトバック処理

ローダプログラムは、アプリケーションプログラムを大容量内蔵 RAM に転送後、キャッシュメモリと大容量内蔵 RAM とのコヒーレンス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。

3.1.10 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 シリアルフラッシュメモリのコマンド

シリアルフラッシュメモリへのアクセスは全てコマンド形式で行います。ローダプログラムは、シリアルフラッシュメモリの High-Speed Read コマンドを使用して、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し、大容量内蔵 RAM に転送します。表 4 にローダプログラムで使用するシリアルフラッシュメモリのコマンドを示します。

表4 ローダプログラムで使用するシリアルフラッシュメモリのコマンド

コマンド名	オペコード	機能
High-Speed Read	H'0B	データの読み出し

【注】 本応用例では SST 社製の SST25VF016B のコマンドを参考にしていますが、シリアルフラッシュメモリのコマンド仕様は、デバイスメーカーによって異なります。使用するデバイスのデータシートを参照してください。

3.1.13 ローダプログラム実行後のレジスタ状態

表 5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 5に記載していないレジスタは、SH7266 グループ、SH7267 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5 ローダ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'1103	
スタンバイコントロールレジスタ 3	STBCR3	H'02	
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'10	
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'2A	
スタンバイコントロールレジスタ 8	STBCR8	H'7E	
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
キャッシュ制御レジスタ 1	CCR1	H'0000 0101	命令キャッシュ有効 オペランドキャッシュ有効
制御レジスタ_0	SPCR_0	H'88	
スレーブセレクト極性レジスタ_0	SSLP_0	H'00	
端子制御レジスタ_0	SPPCR_0	H'30	
ステータスレジスタ_0	SPSR_0	H'E0	
データレジスタ_0	SPDR_0	不定	
シーケンス制御レジスタ_0	SPSCR_0	H'00	
シーケンスステータスレジスタ_0	SPSSR_0	H'00	
ビットレートレジスタ_0	SPBR_0	H'00	
データコントロールレジスタ_0	SPDCR_0	H'20	
クロック遅延レジスタ_0	SPCKD_0	H'00	
スレーブセレクトネゲート遅延 レジスタ_0	SSLND_0	H'00	
コマンドレジスタ_00	SPCMD_00	H'E781	
コマンドレジスタ_01	SPCMD_01	H'070D	
コマンドレジスタ_02	SPCMD_02	H'070D	
コマンドレジスタ_03	SPCMD_03	H'070D	
バッファコントロールレジスタ_0	SPBFCR_0	H'00	
バッファデータカウントセット レジスタ_0	SPBFDR_0	H'0002	

表6 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
DMA ソースアドレスレジスタ_0	SAR_0	H'FFFF8004	
DMA デスティネーションアドレス レジスタ_0	DAR_0	H'1C001ED0	
DMA チャンネルコントロールレジスタ_0	CHCR_0	H'00080000	
DMA オペレーションレジスタ	DMAOR	H'0001	
DMA 拡張リソースセクタ 0	DMARS0	H'0052	
ポート F コントロールレジスタ 3	PFCR3	H'0003	QMI/QIO1 端子
ポート F コントロールレジスタ 2	PFCR2	H'3330	QMO/QIO0 端子、 QSSL 端子、QSPCLK 端子

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、ローダプログラムによりシリアルフラッシュメモリから大容量内蔵 RAM へ転送されるため、ローダプログラムが読み込むことが可能なメモリ配置に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、RAM 上で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大容量内蔵 RAM 上に配置します。
2. 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アドレスと最終アドレスの情報を用いてシリアルフラッシュメモリから大容量内蔵 RAM へ転送を行います。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物理的に連続した領域に配置してください。未初期化データ領域およびスタック領域は任意のアドレスに配置することが可能です。
3. 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。
4. 高速内蔵 RAM の H'FFF8 0000~H'FFF8 1FFF 番地はローダプログラムが使用するため、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、初期化データ領域を配置することはできません。
5. 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。
6. キャッシュ操作プログラムは、キャッシュ無効空間上で実行します。そのため PCACHE セクションは高速内蔵 RAM 上の RPCACHE セクションに再配置してから実行します。

図 8 にセクション配置例を示します。

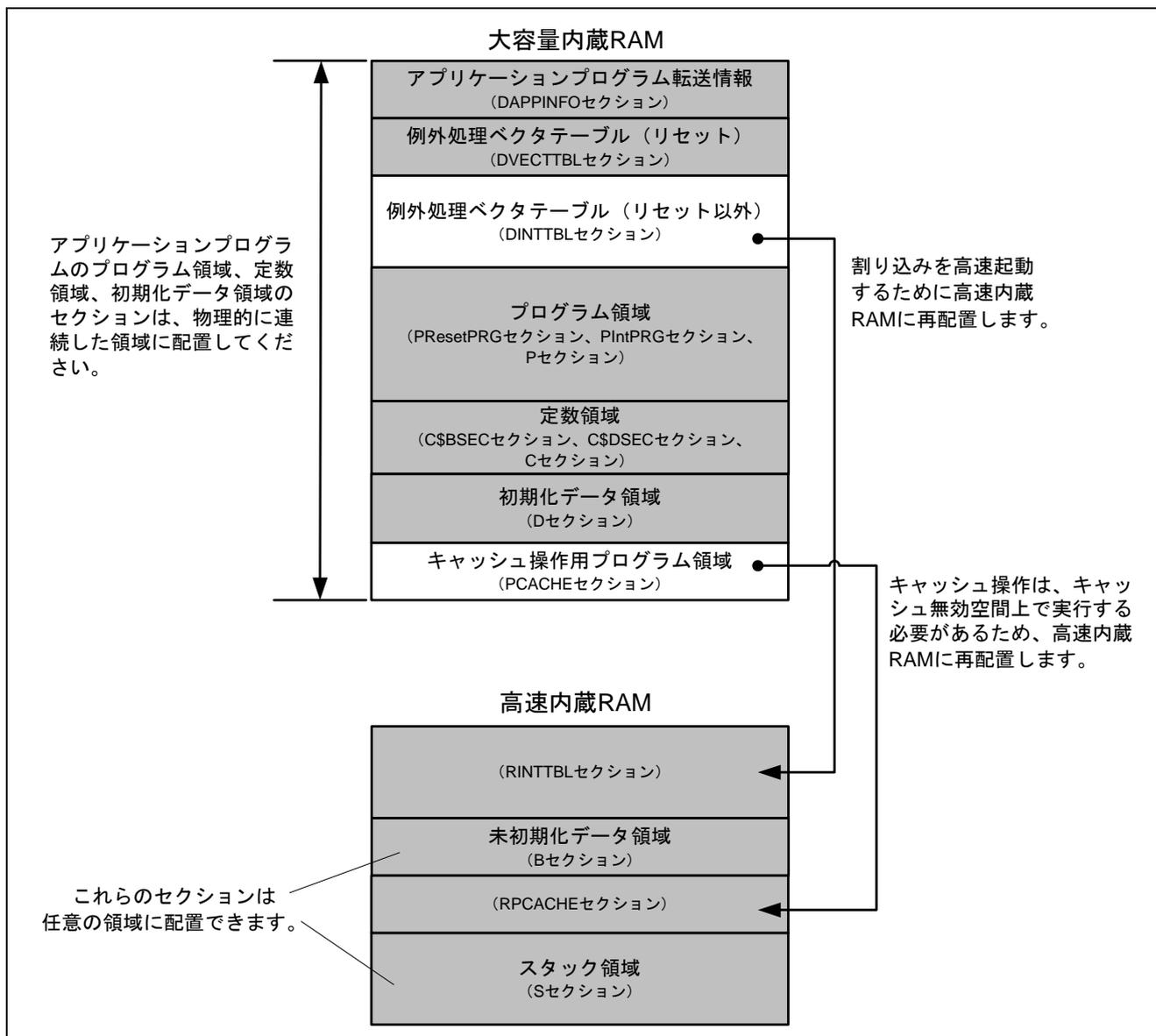


図8 アプリケーションプログラムのセクション配置例

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 0 (SCIF0) へ文字列データを送信する処理を行っています。図 9 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。

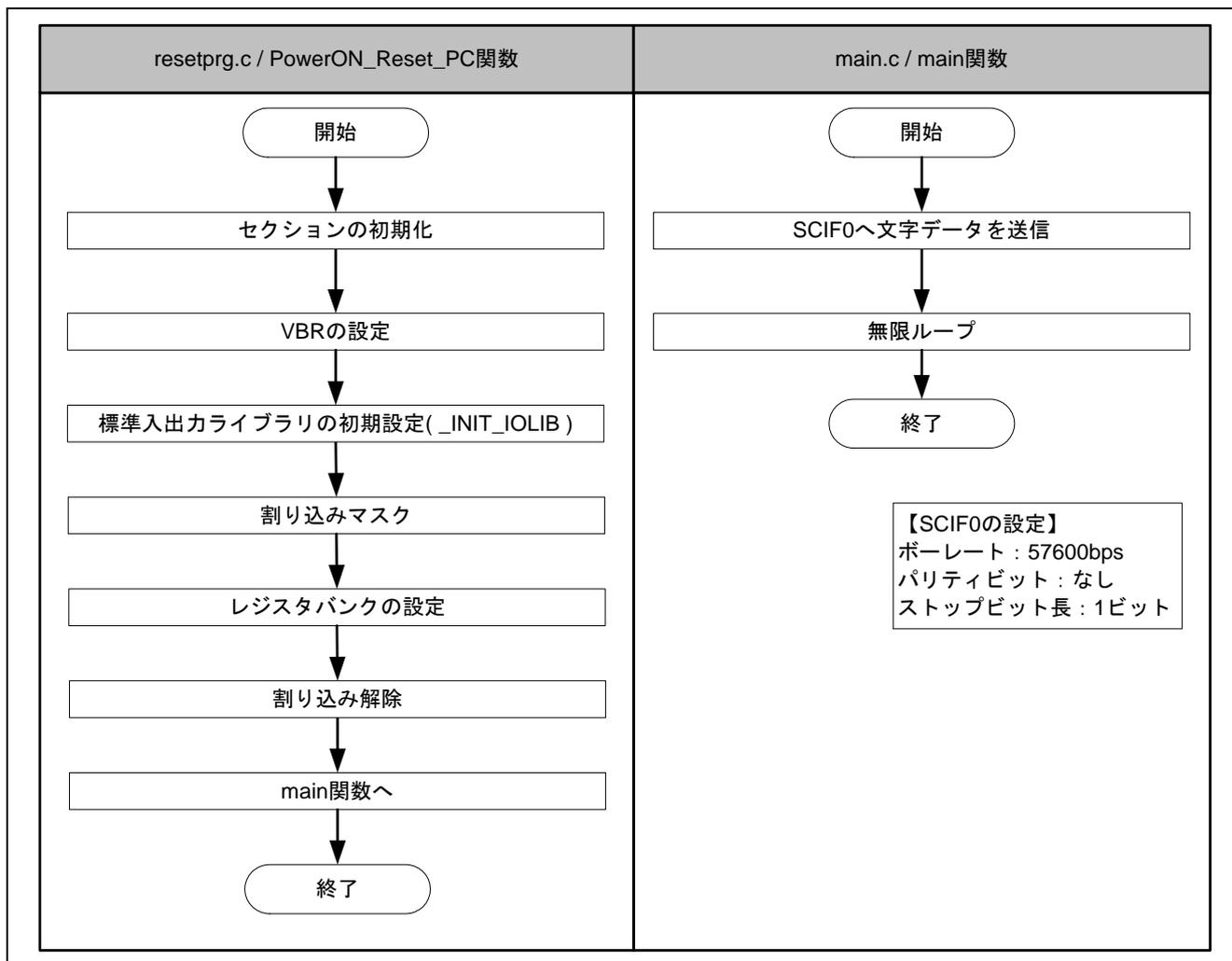


図9 アプリケーションプログラムのフローチャート

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 7 にエントリ関数アドレスの設定を示します。

表7 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 8 にスタックポインタ設定値を示します。

表8 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 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。

3.2.7 アプリケーションプログラム転送情報（appinfo）の生成

表 9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9 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
	void *app_top	__sectop("DAPPINFO")	アプリケーションプログラムの先頭アドレス
	void *app_end	__secend("PCACHE")	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】 ローダプログラム（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 10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。

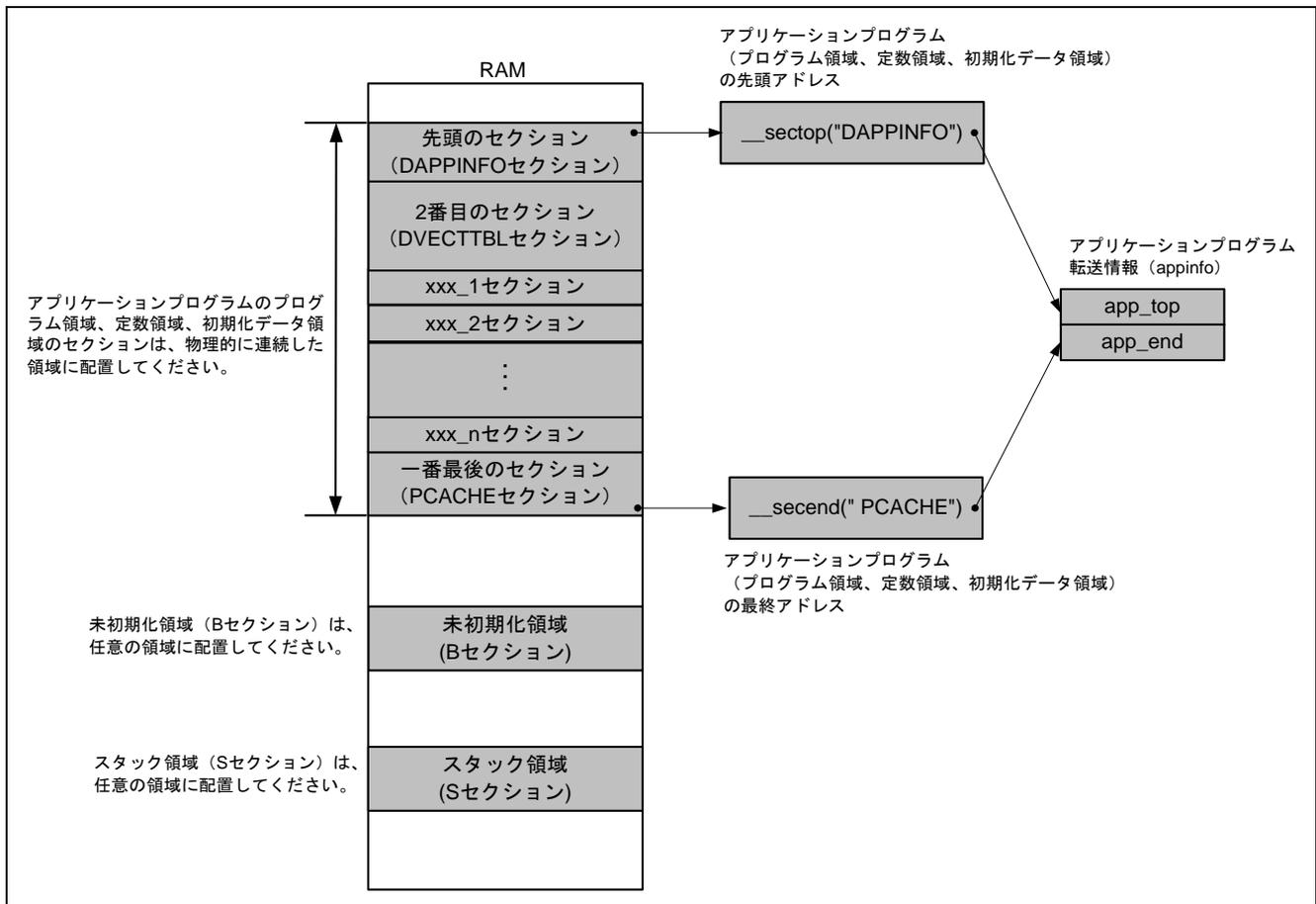


図10 アプリケーションプログラム転送情報 (appinfo) の生成イメージ図

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 11 に動作イメージ図を示します。

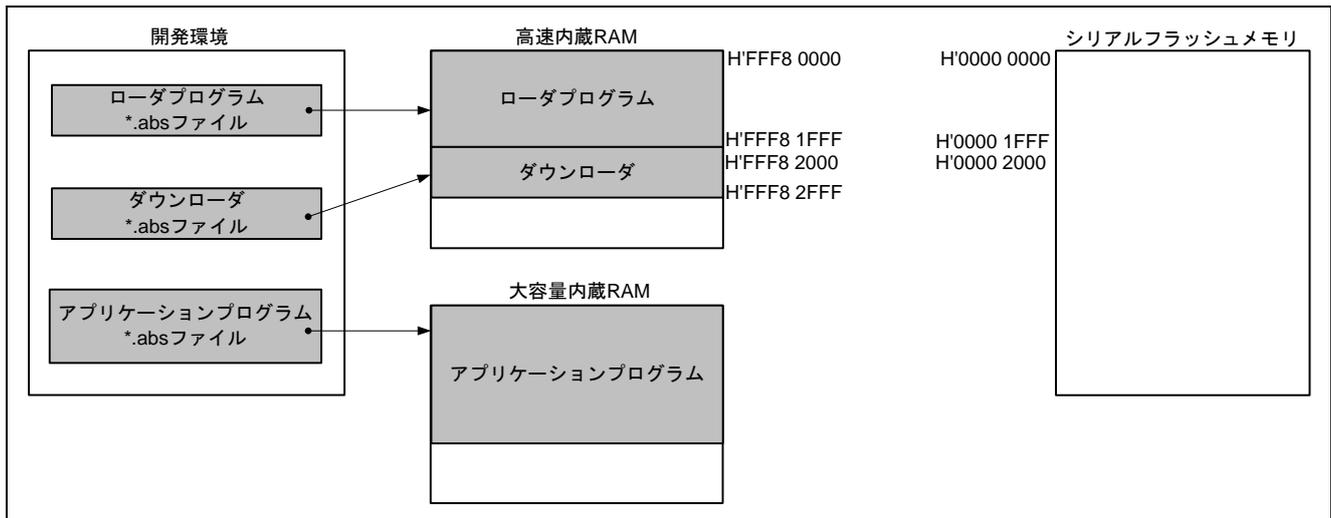


図11 ダウンローダ動作イメージ図 (1)

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 12に動作イメージ図を示します。

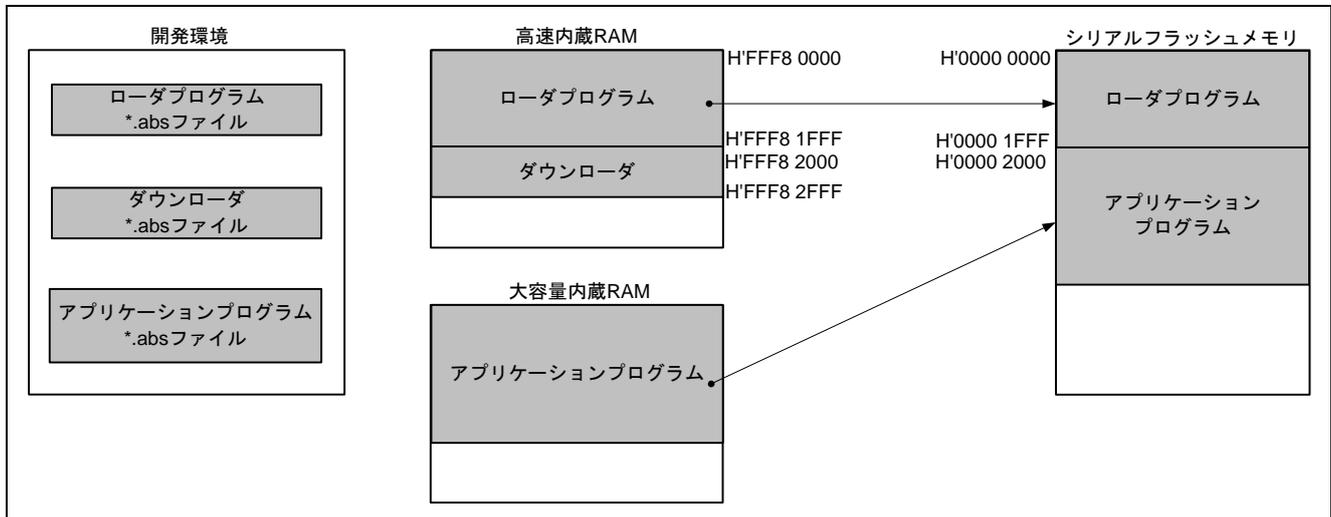


図12 ダウンローダ動作イメージ図 (2)

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 2FFF 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 13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。

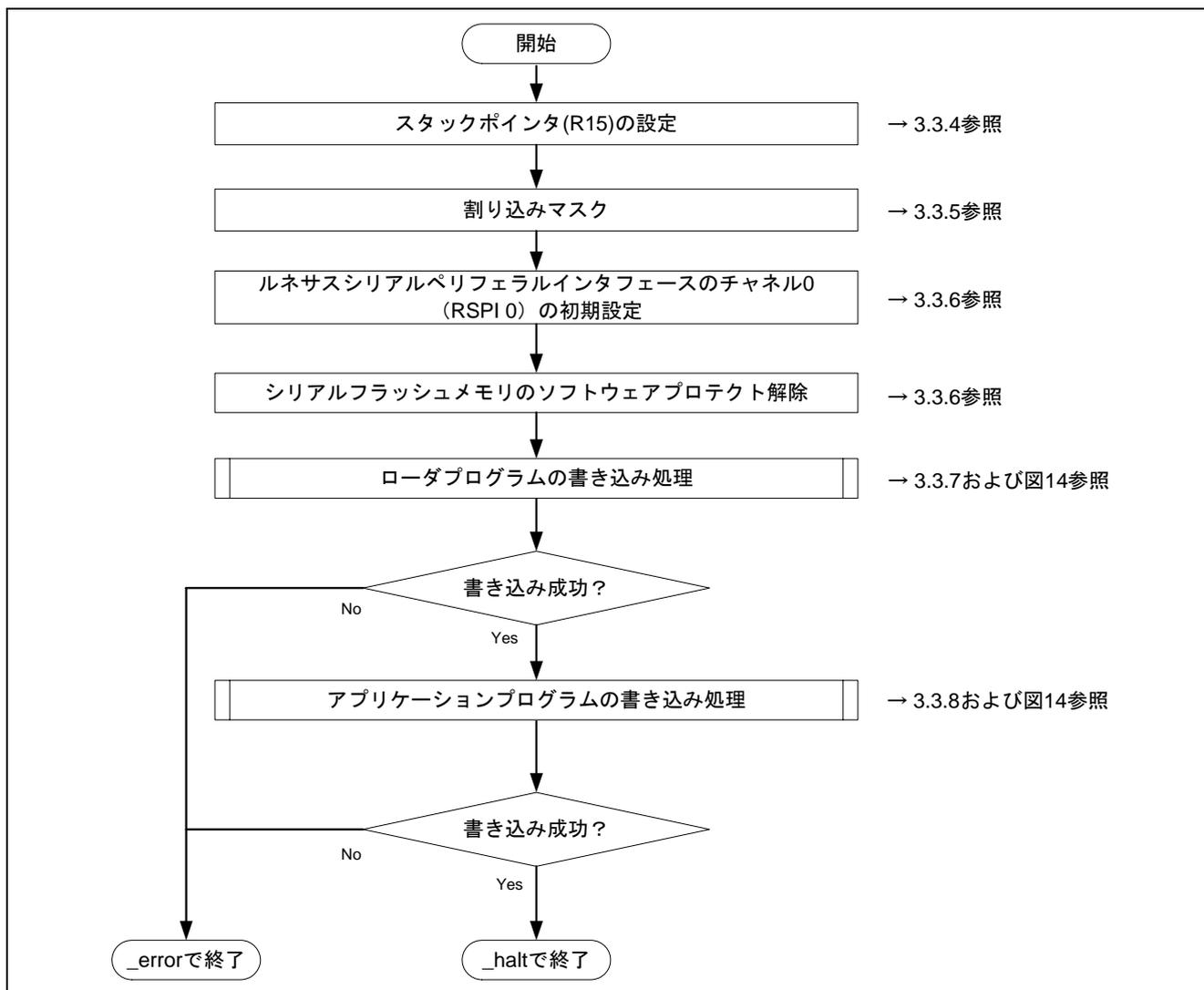


図13 ダウンローダのフローチャート

図 14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。

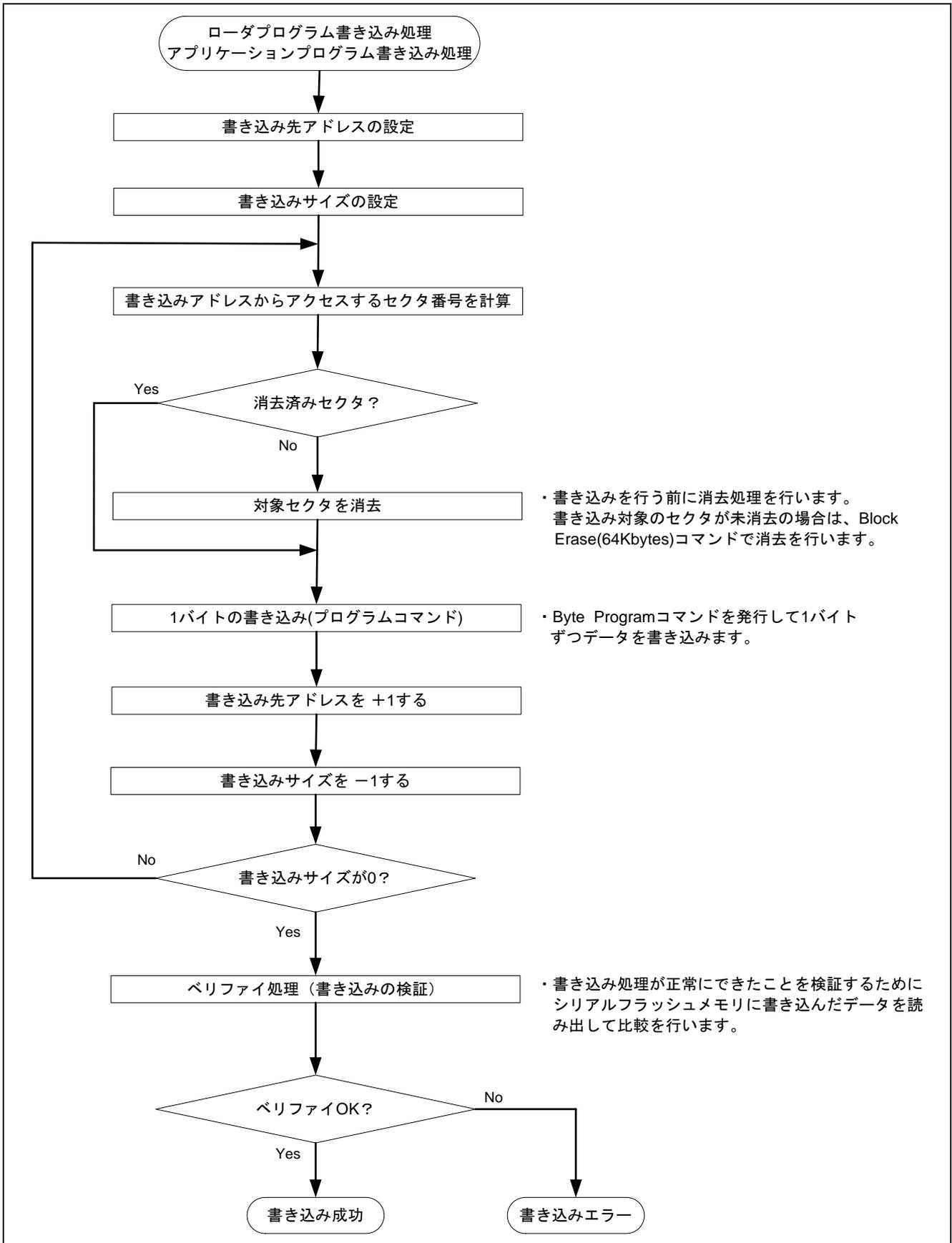


図14 書き込み処理のフローチャート

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 3000 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. RSPIO 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 Status Register 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 10 にローダプログラムの書き込み処理を示します。

表10 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"> 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。書き込みは 1 バイト単位で行っています。

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 11 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11 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)	アプリケーションプログラム内にある appinfo から取得 (アプリケーションプログラムに依存)
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	<ol style="list-style-type: none">書き込み先アドレスが消去済みかどうかを確認します。未消去エリアの場合は消去処理を行います。プログラムコマンドを発行し書き込み処理を行います。書き込みは 1 バイト単位で行っています。

3.3.9 シリアルフラッシュメモリのコマンド

表 12にダウンローダで使用するシリアルフラッシュメモリのコマンドを示します。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 (RSPI0) 経由でこれらのコマンドを発行し、シリアルフラッシュメモリを操作します。

表12 ダウンローダで使用するシリアルフラッシュメモリのコマンド

コマンド名	オペコード	機能
High-Speed Read	H'0B	データの読み出し
Write Enable	H'06	プログラム（書き込み）／イレース／ライトステータスレジスタコマンドの許可
Write Disable	H'04	プログラム（書き込み）／イレース／ライトステータスレジスタコマンド等の禁止
Read Status Register	H'05	ステータスレジスタの読み出し
Write Status Register	H'01	ステータスレジスタの書き込み（プロテクトの解除）
Block Erase (64Kbytes)	H'D8	ブロック消去（64KB）
Byte Program	H'02	データの書き込み（1バイト）

- 【注】 1. 本応用例では SST 社製の SST25VF016B のコマンドを参考にしていますが、シリアルフラッシュメモリのコマンド仕様は、デバイスメーカーによって異なります。使用するデバイスのデータシートを参照してください。
2. シリアルフラッシュメモリにデータを書き込む前に、書き込み対象の領域にあるデータを消去してください。

3.3.10 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、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、これらの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 15 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。

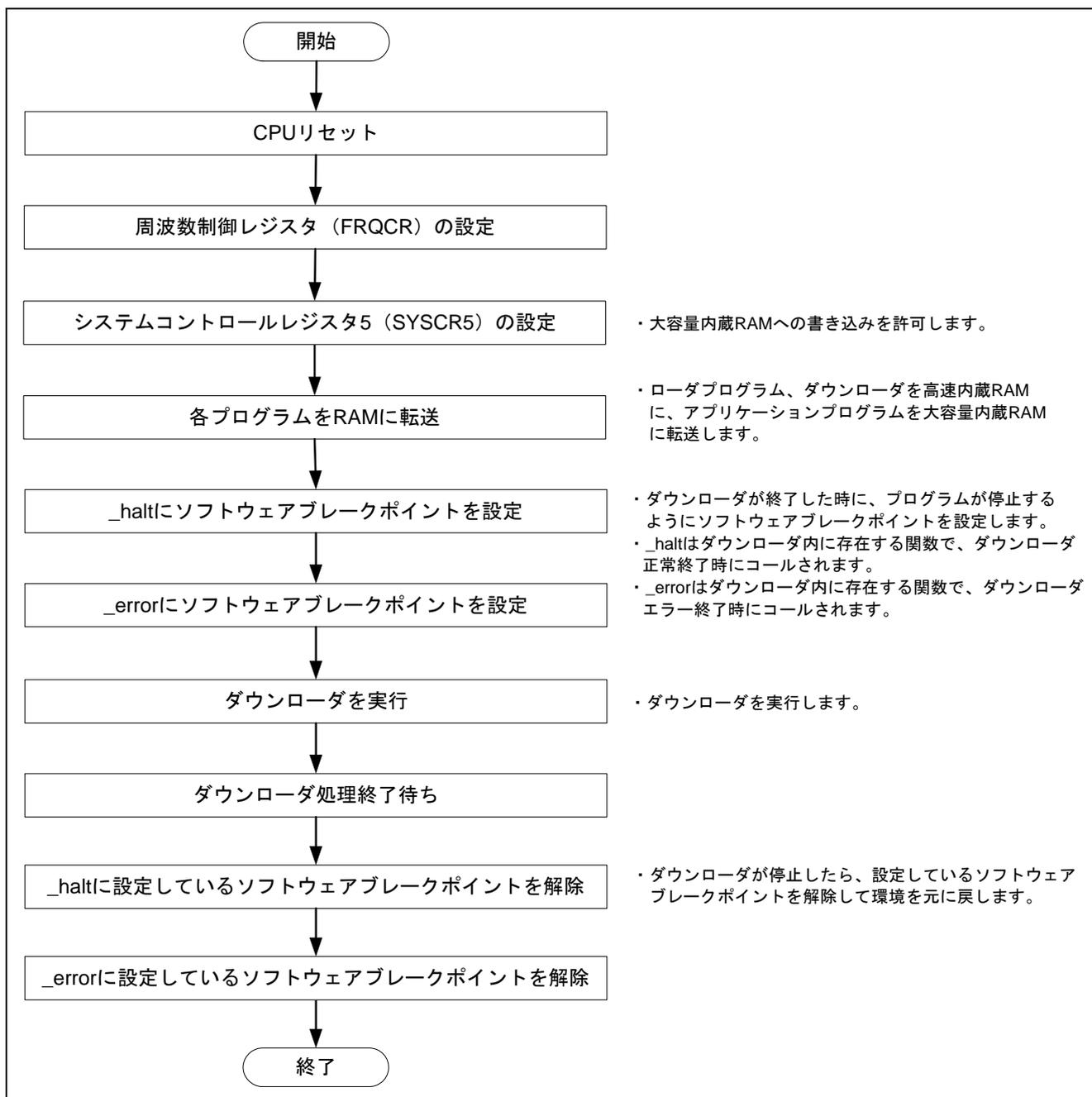


図 15 コマンドバッチファイルのフローチャート

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"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) 2010 Renesas Electronics Corporation. All rights reserved.
29     ;*"FILE COMMENT"***** Technical reference data *****
30     ;*  System Name : SH7266/SH7267 Sample Program
31     ;*  File Name   : loader.src
32     ;*  Abstract    : Loader program preprocessing/jump processing to the application
33     ;*  Version     : 1.00.00
34     ;*  Device      : SH7266/SH7267
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: R0K57267(CPU board)
40     ;*  Description :
41     ;*****
42     ;*  History     : Aug.17,2010 Ver.1.00.00 First Release
43     ;*"FILE COMMENT END"*****/
44     .SECTION LOADER_ENTRY,CODE,ALIGN = 4
45     .IMPORT _main
46     .EXPORT _jmp_app_prog
47
```

4.1.2 ローダプログラムリスト"loader.src" (2)

```

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

```

4.1.3 ローダプログラムリスト"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) 2010 Renesas Electronics Corporation. All rights reserved.
29 *   "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7266/SH7267 Sample Program
31 *   File Name   : main.c
32 *   Abstract    : Loader program
33 *   Version     : 1.00.00
34 *   Device      : SH7266/SH7267
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: R0K57267(CPU board)
40 *   Description :
41 *****/
42 *   History     : Aug.17,2010 Ver.1.00.00 First Release
43 *   "FILE COMMENT END"*****/
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"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 io_init_cache(void);
71  extern int io_cache_writeback(void);
72  extern void sf_byte_read_long(unsigned long addr, unsigned long *buf, int size);
73
74  /* ==== external data ==== */
75  extern unsigned long DUMMY_Vectors;
76
77
```

4.1.5 ローダプログラムリスト"main.c" (3)

```

78  /*"FUNC COMMENT"*****
79  * ID      :
80  * Outline : Loader program main
81  *-----
82  * Include : #include "serial_flash.h"
83  *-----
84  * Declaration : void main(void);
85  *-----
86  * Description : Refers the data in the appinfo to transfer the application program
87  *              : to the large-capacity internal RAM, and jumps to the entry function
88  *              : of the application program.
89  *-----
90  * Argument  : void
91  *-----
92  * Return Value: void
93  /*"FUNC COMMENT END"*****/
94  void main(void)
95  {
96      unsigned long app_top,app_end;
97
98      /* Sets the FPSCR */
99      set_fpscr(FPSCR_INIT);
100
101      /* Sets the tentative VBR */
102      set_vbr((void *)&DUMMY_Vectors);
103
104      /* Masks the interrupt */
105      set_cr(INT_MASK);
106
107      /* Sets the CPG */
108      io_set_cpg();
109
110      /* Enables the cache */
111      io_init_cache();
112
113      /* Sets the RSPI0 */
114      sf_init_serial_flash();
115
116      /* Retrieves the appinfo */
117      get_appinfo(&app_top,&app_end);
118
119      /* Transfers the application program to the large-capacity internal RAM */
120      app_prog_transfer(app_top, app_end);
121
122      /* Writes back the cache */
123      io_cache_writeback();
124
125      /* Jumps to the application program */
126      jmp_app_prog(app_top);
127
128      while(1){
129          /* LOOP */
130      }
131  }

```

4.1.6 ローダプログラムリスト"main.c" (4)

```
132
133 /*"FUNC COMMENT"*****
134 * ID      :
135 * Outline : Retrieve the appinfo
136 *-----
137 * Include : #include "serial_flash.h"
138 *-----
139 * Declaration : void get_appinfo (unsigned long *app_top_addr,
140 *      :      unsigned long *app_end_addr);
141 *-----
142 * Description : Retrieves the appinfo.
143 *      : Retrieves the appinfo.top from H'2000 to H'2003 in serial flash
144 *      : memory, and stores it in the address specified by the first
145 *      : argument. This function also retrieves the appinfo.end from
146 *      : H'2004 to H'2007 in serial flash memory, and stores it in the
147 *      : address specified by the second argument.
148 *-----
149 * Argument  : unsigned long app_top_addr ; 0 : Start address of the application
150 *      :      program at destination
151 *      : unsigned long app_end_addr ; 0 : End address of the application
152 *      :      program at destination
153 *-----
154 * Return Value: void
155 *"FUNC COMMENT END"*****/
156 void get_appinfo( unsigned long *app_top_addr,unsigned long *app_end_addr)
157 {
158     /* Retrieves the appinfo.top */
159     sf_byte_read(APPINFO_TOP, (unsigned char *)app_top_addr, 4);
160
161     /* Retrieves the appinfo.end */
162     sf_byte_read(APPINFO_END, (unsigned char *)app_end_addr, 4);
163 }
164
```

4.1.7 ロードプログラムリスト"main.c" (5)

```
165  /*"FUNC COMMENT"*****
166  * ID      :
167  * Outline : Transfer the application program
168  *-----
169  * Include : #include "serial_flash.h"
170  *-----
171  * Declaration : void app_prog_transfer(unsigned long app_top_addr,
172  *           :           unsigned long app_end_addr);
173  *-----
174  * Description : Calculates the size of the application program, and transfers
175  *           : the application program from serial flash memory to the
176  *           : large-capacity internal RAM. (Rounds up the allocation size of the
177  *           : application program to multiples of 4 to transfer in longword.)
178  *-----
179  * Argument  : unsigned long app_top_addr ; I : Start address of the application
180  *           :           program at destination
181  *           : unsigned long app_end_addr ; I : End address of the application
182  *           :           at destination
183  *-----
184  * Return Value: void
185  *"FUNC COMMENT END"*****/
186  void app_prog_transfer(unsigned long app_top_addr,unsigned long app_end_addr)
187  {
188      unsigned long app_prog_size;
189
190      /* Calculates the size of the application program */
191      app_prog_size = app_end_addr - app_top_addr;
192      if( ( app_prog_size & 0x00000003 ) != 0 ){
193          app_prog_size &= 0xFFFFFFF4;
194          app_prog_size += 4;          /* Rounds up the allocation size of the application
195                                     program to multiples of 4. */
196      }
197
198      /* Loads the application program in the large-capacity internal RAM */
199      sf_byte_read_long(APROG_TOP_SFLASH, (unsigned long *)app_top_addr, app_prog_size);
200  }
201
```

4.1.8 ローダプログラムリスト"main.c" (6)

```
202  /*"FUNC COMMENT"*****  
203  * ID      :  
204  * Outline : Terminate the system  
205  *-----  
206  * Include :  
207  *-----  
208  * Declaration : void system_down(void);  
209  *-----  
210  * Description : This function contains the infinite loop.  
211  *             : As this is registered in the DUMMY_Vectors table, this is  
212  *             : called when an exception occurs while the loader program  
213  *             : is operating.  
214  *-----  
215  * Argument  : void  
216  *-----  
217  * Return Value: void  
218  *"FUNC COMMENT END"*****/  
219  void system_down(void)  
220  {  
221      while(1){  
222          /* System error */  
223      }  
224  }  
225  
226  /* End of File */  
227
```

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) 2010 Renesas Electronics Corporation. All rights reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7266/SH7267 Sample Program
31 *  File Name   : main.c
32 *  Abstract    : Application program example
33 *  Version     : 1.00.00
34 *  Device      : SH7266/SH7267
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: R0K57267(CPU board)
40 *  Description :
41 *****/
42 *  History     : Aug.17,2010 Ver.1.00.00 First Release
43 *  "FILE COMMENT END"*****
44 #include <stdio.h>
45
46 /* ==== prototype declaration ==== */
47 void main(void);
48
```

4.2.2 アプリケーションプログラムリスト"main.c" (2)

```
49  /*"FUNC COMMENT"*****
50  * ID      :
51  * Outline : Application program main function
52  *-----
53  * Include :
54  *-----
55  * Declaration : void main(void);
56  *-----
57  * Description : Transmits the strings of characters to the SCIF0.
58  *              : (Baud rate: 57600 bps, no parity, stop bit length: 1).
59  *-----
60  * Argument  : void
61  *-----
62  * Return Value: void
63  *"FUNC COMMENT END"*****/
64  void main(void)
65  {
66      puts("\nSH7267 CPU Board Sample Program. Ver.1.00.00");
67      puts("Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.\n");
68      puts("SH7266/SH7267 Serial-flash boot done.\n");
69      fflush(stdout);
70
71      while(1){
72          /* loop */
73      }
74  }
75  /* End of File */
```

4.2.3 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) 2010 Renesas Electronics Corporation. All rights reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7266/SH7267 Sample Program
31 *  File Name   : appinfo.c
32 *  Abstract    : Generate the application program transfer information (appinfo).
33 *  Version     : 1.00.00
34 *  Device      : SH7266/SH7267
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: R0K57267(CPU board)
40 *  Description :
41 *****/
42 *  History     : Aug.17,2010 Ver.1.00.00 First Release
43 *  "FILE COMMENT END"*****/
```

4.2.4 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
```

4.2.5 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) 2010 Renesas Electronics Corporation. All rights reserved.
29 *   "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7266/SH7267 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      : SH7266/SH7267
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: R0K57267(CPU board)
40 *   Description :
41 *****/
42 *   History     : Aug.17,2010 Ver.1.00.00 First Release
43 *   "FILE COMMENT END"*****
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) 2010 Renesas Electronics Corporation. All rights reserved.
29     #*"FILE COMMENT"***** Technical reference data *****
30     #*   System Name : SH7266/SH7267 Sample Program
31     #*   File Name   : downloader.hdc
32     #*   Abstract    : Batch File for the Downloader
33     #*   Version     : 1.00.00
34     #*   Device      : SH7266/SH7267
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: R0K57267(CPU board)
40     #*   Description :
41     #*****
42     #*   History     : Aug.17,2010 Ver.1.00.00 First Release
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     # Set the CPG
53     # FRQCR I=144MHz/B=72MHz/P=36MHz/CLK MODE0
54     MF H'FFFE0010 H'FFFE0011 H'1103 WORD
55
56     # On-Chip Large-Capacity RAM write enable
57     # CPG.SYSCR5.BYTE = 0x0fu;
58     MF H'FFFE0428 H'FFFE0428 H'0F BYTE
59 }
60
61
62 proc downloader {} {
63     # Reset CPU
64     reset
65
66     # Calls the init_hardware routine
67     init_hardware
68
69     # Downloads all modules registered in the High-performance Embedded Workshop
70     file_load_all
71
72     # Enables the user stack (to use the software breakpoint)
73     sh2a_sbstk enable
74
75     # Inserts a software breakpoint at the _halt (refer to main.c)
76     set_disassembly_soft_break _halt set
77
78     # Inserts a software breakpoint at the _error (refer to main.c)
79     set_disassembly_soft_break _error set
80
81     # Executes the _downloader (refer to downloader.src) to wait until it terminates
82     go wait _downloader
83
84     # Removes a software breakpoint at the _halt
85     set_disassembly_soft_break _halt clear
86
87     # Removes a software breakpoint at the _error
88     set_disassembly_soft_break _error clear
89
90 }
91
92 downloader
93 #Macro downloader -End
94
95
96
97 # Note: "tcl", "reset", "file_load", "sh2a_sbstk", "set_disassembly_soft_break",
98 # and "go" are commands used in the High-performance Embedded Workshop and the
99 # E10A-USB emulator. For details, refer to manuals.
100
101 # /* End of File */
```

4.3.3 ダウンロードプログラムリスト"downloader.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 Renesas Electronics Corporation. All rights reserved.
29     ;*"FILE COMMENT"***** Technical reference data *****
30     ;*   System Name : SH7266/SH7267 Sample Program
31     ;*   File Name   : downloader.src
32     ;*   Abstract    : Downloader
33     ;*   Version     : 1.00.00
34     ;*   Device      : SH7266/SH7267
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: R0K57267(CPU board)
40     ;*   Description :
41     ;*****
42     ;*   History     : Aug.17,2010 Ver.1.00.00 First Release
43     ;*"FILE COMMENT END"*****
```

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

```
44     .SECTION DOWNLOADER_ENTRY, CODE, ALIGN = 4
45     .IMPORT _main
46
47     _downloader:
48     MOV.L L2,R15      ; Sets the stack pointer
49     MOV.L L1,R0      ; Retrieves the entry function of the downloader
50     JMP @R0         ; Jumps to the entry function of the downloader
51     NOP
52
53     .ALIGN 4
54     L1:
55     .DATA.L _main    ; Entry function address of the downloader
56
57     L2:
58     .DATA.L H'FFF83000 ; Stack pointer (R15) value of the downloader
59
60     .pool
61     .end
```

4.3.5 ダウンローダプログラムリスト"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) 2010 Renesas Electronics Corporation. All rights reserved.
29 *   "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7266/SH7267 Sample Program
31 *   File Name   : main.c
32 *   Abstract    : Downloader
33 *   Version     : 1.00.00
34 *   Device      : SH7266/SH7267
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: R0K57267(CPU board)
40 *   Description :
41 *****/
42 *   History     : Aug.17,2010 Ver.1.00.00 First Release
43 *   "FILE COMMENT END"*****/
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 ダウンローダプログラムリスト"main.c" (2)

```
50  /* ==== macro defined ==== */
51  #define INT_MASK    0x000000F0          /* Value to set in the SR register
52                                         (for masking the interrupt) */
53
54  #define SECTOR_SIZE    0x10000         /* Sector size: 64 KB */
55  #define SECTOR_NUM    32              /* Total number of sectors in the device */
56  #define DEVICE_SIZE    (SECTOR_SIZE * SECTOR_NUM) /* Device size */
57
58  #define L_PROG_SIZE    8192           /* Loader program size */
59  #define L_PROG_SRC    0xFFFF80000     /* Source address of the loader program */
60  #define L_PROG_DST    0x00000000     /* Destination address of the loader program */
61
62  #define APROG_TOP_SFLASH  0x00002000  /* Start address of the application program */
63  #define APROG_TOP_RAM    0x1C000000   /* Start address of the application program */
64                                         /* When changing the start section of the */
65                                         /* application program, change this definition */
66
67  #define APPINFO_TOP      APROG_TOP_RAM /* Address the appinfo.app_top is located */
68  #define APPINFO_END      (APROG_TOP_RAM + 4) /* Address the appinfo.app_end is located */
69
70  /* ==== prototype declaration ==== */
71  /*** User API ***/
72  void main(void);
73
74  static void halt(void);
75  static void error(void);
76  static void init_erase_flag(void);
77  static int Is_erased_sector(unsigned long sector_no);
78  static void set_erase_flag(unsigned long sector_no);
79  static int write_prog_data(unsigned char *program_data, unsigned long sflash_addr, unsigned
80                             long size);
81
82  /*** data ***/
83  static unsigned char sflash_erase_flag[SECTOR_NUM]={0}; /* 0: sector not erased, 1: sector
erased */
```

4.3.7 ダウンローダプログラムリスト"main.c" (3)

```
84  /*"FUNC COMMENT"*****
85  * ID      :
86  * Outline : Downloader main
87  *-----
88  * Include :
89  *-----
90  * Declaration : void main(void);
91  *-----
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  *-----
100 * Argument  : void
101 *-----
102 * Return Value: void
103 *"FUNC COMMENT END"*****/
104 void main(void)
105 {
106     unsigned long app_top_addr,app_end_addr,app_prog_size;
107
108     /* Masks the interrupt */
109     set_cr(INT_MASK);
110
111     /* Initializes the erase flag */
112     init_erase_flag();
113
114     /* Initializes the RSPI0 */
115     sf_init_serial_flash();
116
117     /* Disables the software protection in serial flash memory */
118     sf_protect_ctrl(SF_REQ_UNPROTECT);
119
120     /* Writes the loader program */
121     if( write_prog_data( (unsigned char *)L_PROG_SRC, L_PROG_DST, L_PROG_SIZE) < 0 ){
122         error();
123     }
124
125     /* Retrieves the start address and end address from the application program
126        transfer information (appinfo) */
127     app_top_addr = *(volatile unsigned long *)APPINFO_TOP;
128     app_end_addr = *(volatile unsigned long *)APPINFO_END;
129
130     /* Calculates the size of the application program */
131     app_prog_size = app_end_addr - app_top_addr;
132
```

4.3.8 ダウンローダプログラムリスト"main.c" (4)

```
133     /* Writes the application program */
134     if( write_prog_data( (unsigned char *)app_top_addr, APROG_TOP_SFLASH, app_prog_size) < 0 ){
135         error();
136     }
137
138     /* Enables the software protection in serial flash memory */
139     sf_protect_ctrl(SF_REQ_PROTECT);
140
141     /* Exits the downloader */
142     halt();
143 }
144
145 /*"FUNC COMMENT"*****
146 * ID      :
147 * Outline : Write the program data
148 *-----
149 * Include :
150 *-----
151 * Declaration : int write_prog_data(unsigned char *program_data,
152 *                               :               unsigned long sflash_addr, unsigned long size);
153 *-----
154 * Description : Writes the program data as the following procedures.
155 *               : 1. Erase the target sector when it is not erased.
156 *               : 2. Write the program data in serial flash memory.
157 *               : 3. Reads the data in serial flash memory and compare it with the
158 *               :   provided data.
159 *-----
160 * Argument  : unsigned char *program_data ; I : Start address of the program data
161 *            : unsigned long sflash_addr   ; I : Start address at the destination in
162 *            :                               serial flash memory
163 *            : unsigned long size         ; I : Write size
164 *-----
165 * Return Value: Equal or bigger than 0: Success
166 *            : Less than 0: Error
167 *"FUNC COMMENT END"*****
```

4.3.9 ダウンローダプログラムリスト"main.c" (5)

```
168 int write_prog_data(unsigned char *program_data, unsigned long sflash_addr, unsigned long size)
169 {
170     unsigned long sector_no;
171     unsigned long saddr;
172     unsigned long sz;
173     unsigned char read_data;
174     unsigned char *w_p;
175
176     /* ==== Copies the value from the argument to the local variable ==== */
177     saddr = sflash_addr;
178     sz = size;
179     w_p = program_data;
180
181     /* ==== Writes data in serial flash memory ==== */
182     while( sz > 0){
183         sector_no = saddr / SECTOR_SIZE;
184         if( Is_erased_sector(sector_no) == 0 ){ /* When it is not erased */
185             sf_sector_erase(sector_no); /* Erase */
186             set_erase_flag(sector_no); /* When it is erased, set the erase flag */
187         }
188
189         sf_byte_program(saddr, w_p, 1); /* Writes data in units of single byte */
190         w_p++;
191         saddr++;
192         sz--;
193     }
194
195     /* ==== Verifies data (serial flash memory is programmed successfully) ==== */
196     saddr = sflash_addr;
197     sz = size;
198     w_p = program_data;
199
200     while( sz > 0){
201         sf_byte_read(saddr,&read_data, 1);/* Reads the data written in serial flash memory */
202
203         if( *w_p != read_data ){
204             return -1; /* Returns an error when the data unmatched */
205         }
206
207         w_p++;
208         saddr++;
209         sz--;
210     }
211
212     return 0;
213 }
214
```

4.3.10 ダウンローダプログラムリスト"main.c" (6)

```
215  /*"FUNC COMMENT"*****
216  * ID      :
217  * Outline  : Initialize the Erase Flag
218  *-----
219  * Include  :
220  *-----
221  * Declaration : static void init_erase_flag(void);
222  *-----
223  * Description : Initializes the table sflash_erase_flag[].
224  *-----
225  * Argument   : void
226  *-----
227  * Return Value: void
228  *"FUNC COMMENT END"*****/
229  static void init_erase_flag(void)
230  {
231      int i;
232
233      for( i=0; i < SECTOR_NUM ;i++){
234          sflash_erase_flag[i] = 0;
235      }
236  }
237
238  /*"FUNC COMMENT"*****
239  * ID      :
240  * Outline  : Retrieve the Sector Erase Status
241  *-----
242  * Include  :
243  *-----
244  * Declaration : static int Is_erased_sector(unsigned long sector_no);
245  *-----
246  * Description : Returns the information (not erased or eraser) of the
247  *               : sector specified by the sector number.
248  *-----
249  * Argument   : unsigned long sector_no ; I : Sector number
250  *-----
251  * Return Value: 1 : Sector in the specified address is already erased
252  *               : 0 : Sector in the specified address is not erased
253  *"FUNC COMMENT END"*****/
254  static int Is_erased_sector(unsigned long sector_no)
255  {
256      return sflash_erase_flag[sector_no];
257  }
258
```

4.3.11 ダウンローダプログラムリスト"main.c" (7)

```
259  /*"FUNC COMMENT"*****
260  * ID      :
261  * Outline  : Set the Erase Flag
262  *-----
263  * Include  :
264  *-----
265  * Declaration : static void set_erase_flag(unsigned long sector_no);
266  *-----
267  * Description : Sets the erase flag to modify the information of the specified
268  *              : sector as erased.
269  *-----
270  * Argument   : unsigned long sector_no ; I : Sector number
271  *-----
272  * Return Value: void
273  /*"FUNC COMMENT END"*****/
274  static void set_erase_flag(unsigned long sector_no)
275  {
276      sflash_erase_flag[sector_no] = 1;
277  }
278
279  /*"FUNC COMMENT"*****
280  * ID      :
281  * Outline  : Program stops (successful).
282  *-----
283  * Include  :
284  *-----
285  * Declaration : static void halt(void);
286  *-----
287  * Description : When the downloader ends successfully, this function is called
288  *              : to stop the program.
289  *-----
290  * Argument   : void
291  *-----
292  * Return Value: void
293  /*"FUNC COMMENT END"*****/
294  static void halt(void)
295  {
296      while(1){
297          /* When the downloader ends successfully, this function stops the program. */
298      }
299  }
300
```

4.3.12 ダウンローダプログラムリスト"main.c" (8)

```
301  /*"FUNC COMMENT"*****
302  * ID      :
303  * Outline : Program stops (error).
304  *-----
305  * Include :
306  *-----
307  * Declaration : static void error(void);
308  *-----
309  * Description : When the downloader ends in error, this function is called
310  *              : to stop the program.
311  *-----
312  * Argument  : void
313  *-----
314  * Return Value: void
315  *"FUNC COMMENT END"*****/
316  static void error(void)
317  {
318      while(1){
319          /* When the downloader ends in error, this function stops the program */
320          }
321      }
322
323  /* End of File */
324
```

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 13 のように 3 つのワークスペースに分かれています。

表13 参考プログラムの構成

ワークスペース名	説明
sh7267_sflash_downloader	このワークスペースのプロジェクトには、ダウンローダをビルドします。
sh7267_sflash_loader_prog	このワークスペースのプロジェクトには、ローダプログラムをビルドします。
sh7267_sflash_app	このワークスペースのプロジェクトには、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、[sh7267_sflash_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、[sh7267_sflash_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、[sh7267_sflash_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 16 に [sh7267_sflash_app] ワークスペースのディレクトリ構成を示します。[sh7267_sflash_app] ワークスペースでは、図 16 の①、②、④のダウンロードモジュールをプロジェクトに登録しています。また、図 16 の③のバッチファイルをプロジェクトに登録しています。

```

¥sh7267_sflash_app      : ワークスペースディレクトリ
|-sh7267_sflash_app    : プロジェクトディレクトリ
|  |-debug              :
|  |-sh7267_sflash_app.abs : アプリケーションプログラムの実行ファイル-----①
|
|-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
|-sh7267_sflash_downloader.abs : ダウンローダの実行ファイル-----②
|-downloader.hdc       : バッチファイル(ダウンローダ起動用)-----③
|-sh7267_sflash_loader_prog.abs : ローダプログラムの実行ファイル-----④

```

図16 [sh7267_sflash_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。まず、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。

5.2.2 プログラム書き込み手順

ここでは、[sh7267_sflash_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. [sh7267_sflash_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[sh7267_sflash_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 17 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。

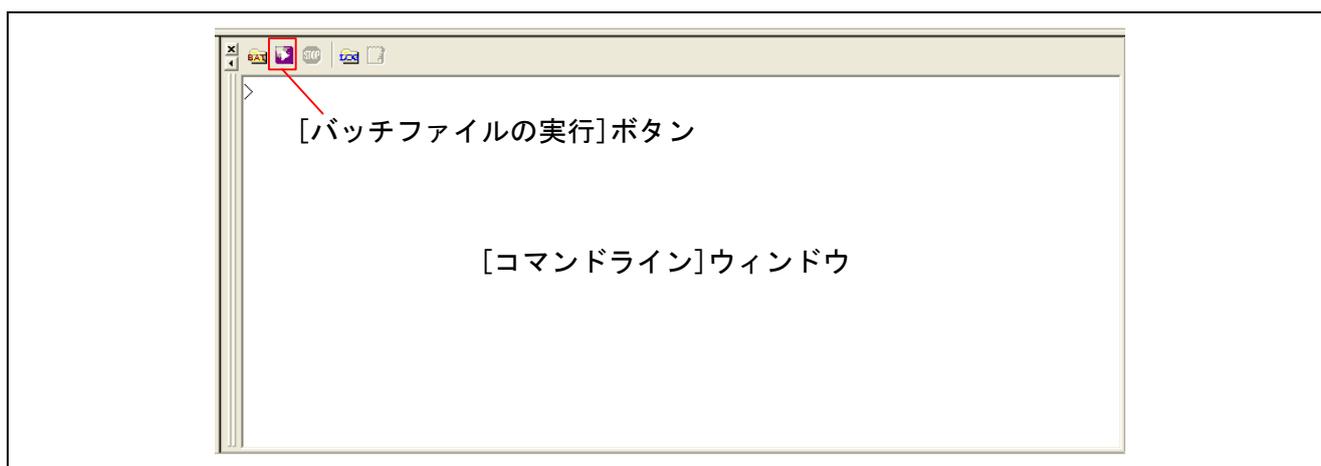


図17 [コマンドライン]ウィンドウと[バッチファイルの実行]ボタン

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 18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。[sh7267_sflash_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。

FFF820EA	AFFE	_halt	BRA	@_halt:12	ダウンローダが正常終了した場合、プログラムカウンタは_haltで停止します。
FFF820EC	0009		NOP		
FFF820EE	AFFE	_error	BRA	@_error:12	ダウンローダが失敗した場合、プログラムカウンタは_errorで停止します。
FFF820F0	0009		NOP		
FFF820F2	0000FFF8		MOVI20	#H'0FFF8,R0	
FFF820F6	210C		CMP/STR	R0,R1	
FFF820F8	FFF8		FMOV.S	@R15,FR15	
FFF820FA	2110		MOV.B	R1,@R1	
FFF820FC	FFF8		FMOV.S	@R15,FR15	
FFF820FE	2140		MOV.B	R4,@R1	
FFF82100	FFF8		FMOV.S	@R15,FR15	
FFF82102	2174		MOV.B	R7,@-R1	
FFF82104	FFF8		FMOV.S	@R15,FR15	
FFF82106	21AA		XOR	R10,R1	

図18 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
SH7266 グループ、SH7267 グループ ユーザーズマニュアル ハードウェア編 Rev.1.00
(最新版をルネサスエレクトロニクスホームページから入手してください。)

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/>

お問い合わせ先

<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	2010.11.12	—	初版発行
1.01	2011.03.31	4	ブートモードの説明を修正

製品ご使用上の注意事項

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

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

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

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

5. 製品間の相違について

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

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

ご注意書き

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

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

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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