

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## 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、海底中継機器、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。

---

# SH7262/SH7264 グループ

## 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、SH7262/SH7264 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

SH7262/SH7264

以下、総称して「SH7264」として説明します。

### 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.....	28
5. ダウンローダの使用法.....	50
6. 参考ドキュメント.....	53

## 1. はじめに

### 1.1 仕様

SH7264 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 1
- 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	SH7262/SH7264
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	ルネサステクノロジ製 High-performance Embedded Workshop Ver.4.04.01
C コンパイラ	ルネサステクノロジ製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.02 Release00
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。

- SH7262/SH7264 グループ 初期設定例
- SH7262/SH7264 グループ ルネサスシリアルペリフェラルインタフェース シリアルフラッシュメモリ 接続例

## 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트리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高速通信でブートします。 高速通信：バスクロック(B $\phi$ )の 1/2 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低速通信でブートします。 低速通信：バスクロック(B $\phi$ )の 1/4 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 (RSPI0) 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。

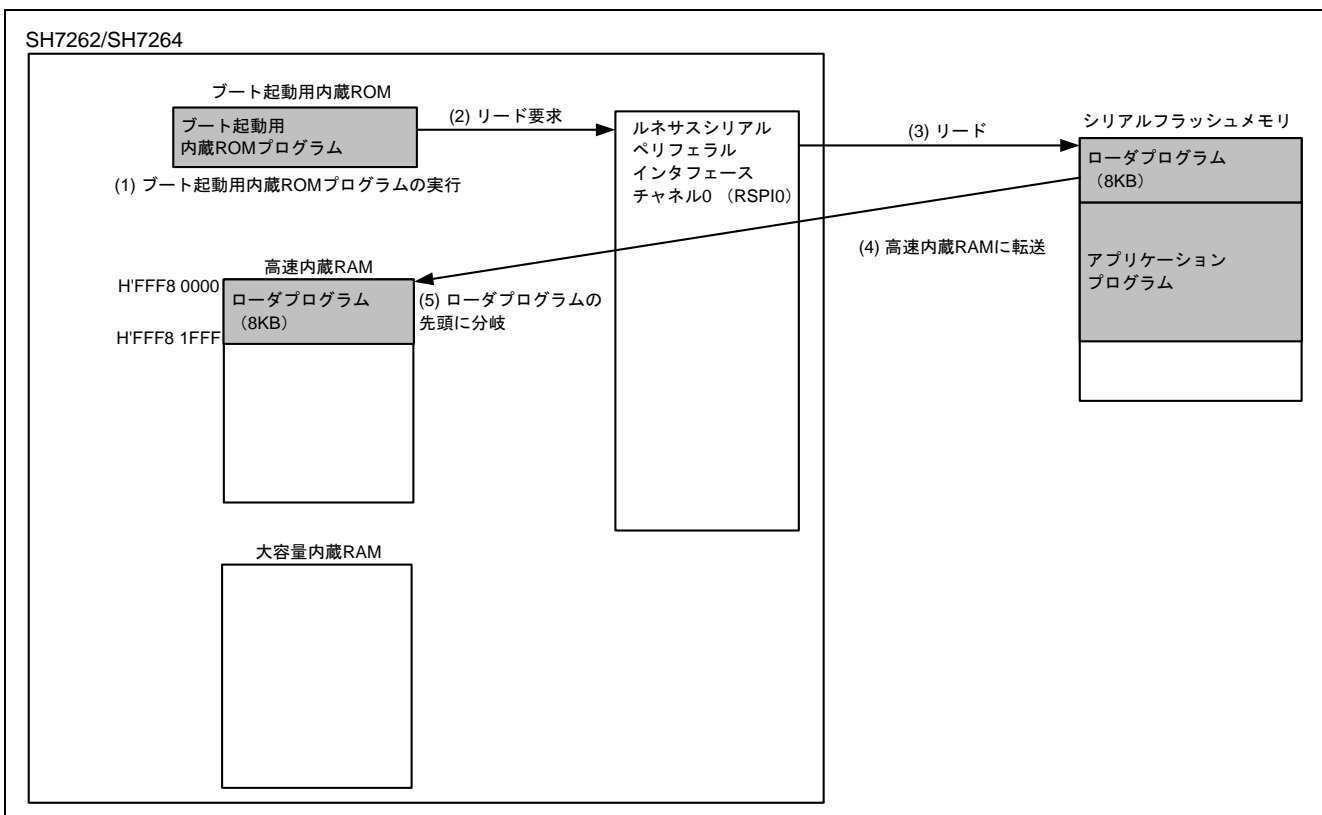


図1 ブート起動用内蔵 ROM プログラムの動作イメージ図

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。

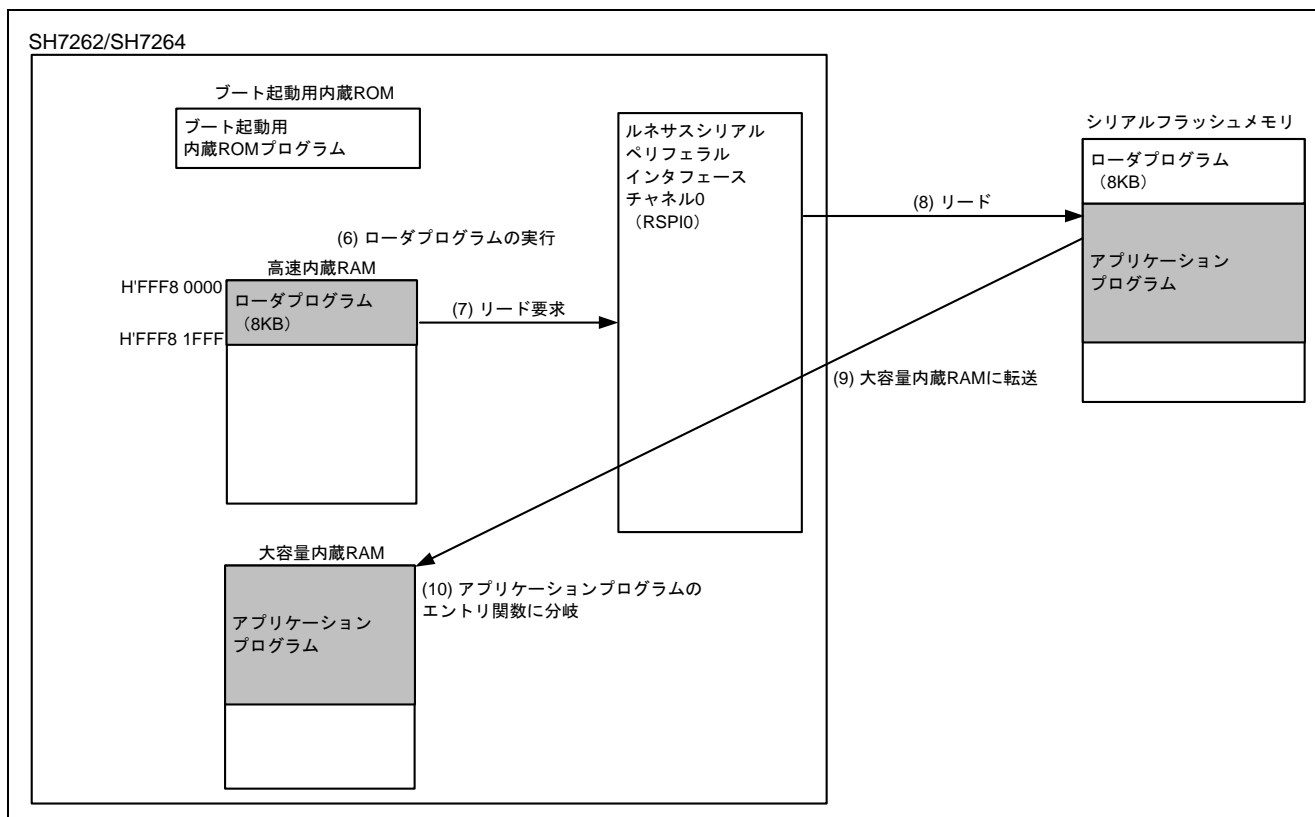


図2 ローダプログラムの動作イメージ図

【注】 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。

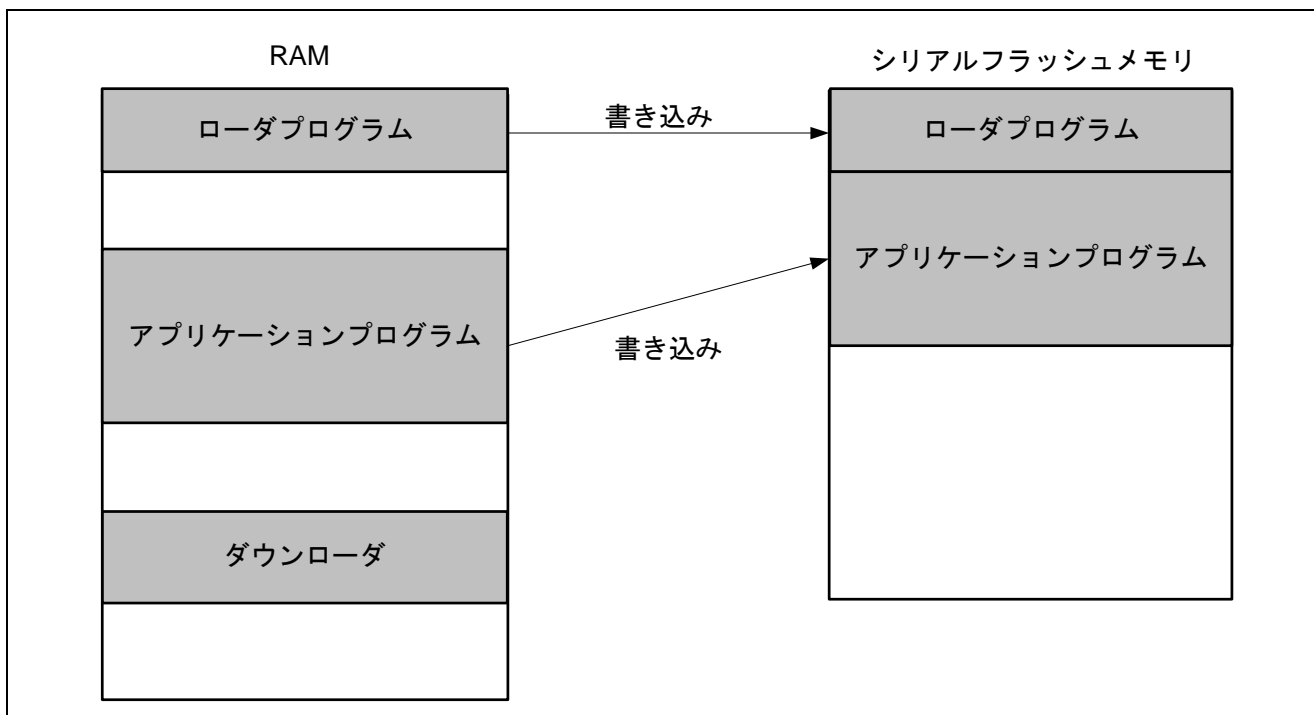


図3 ダウンローダの動作イメージ図



### 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。

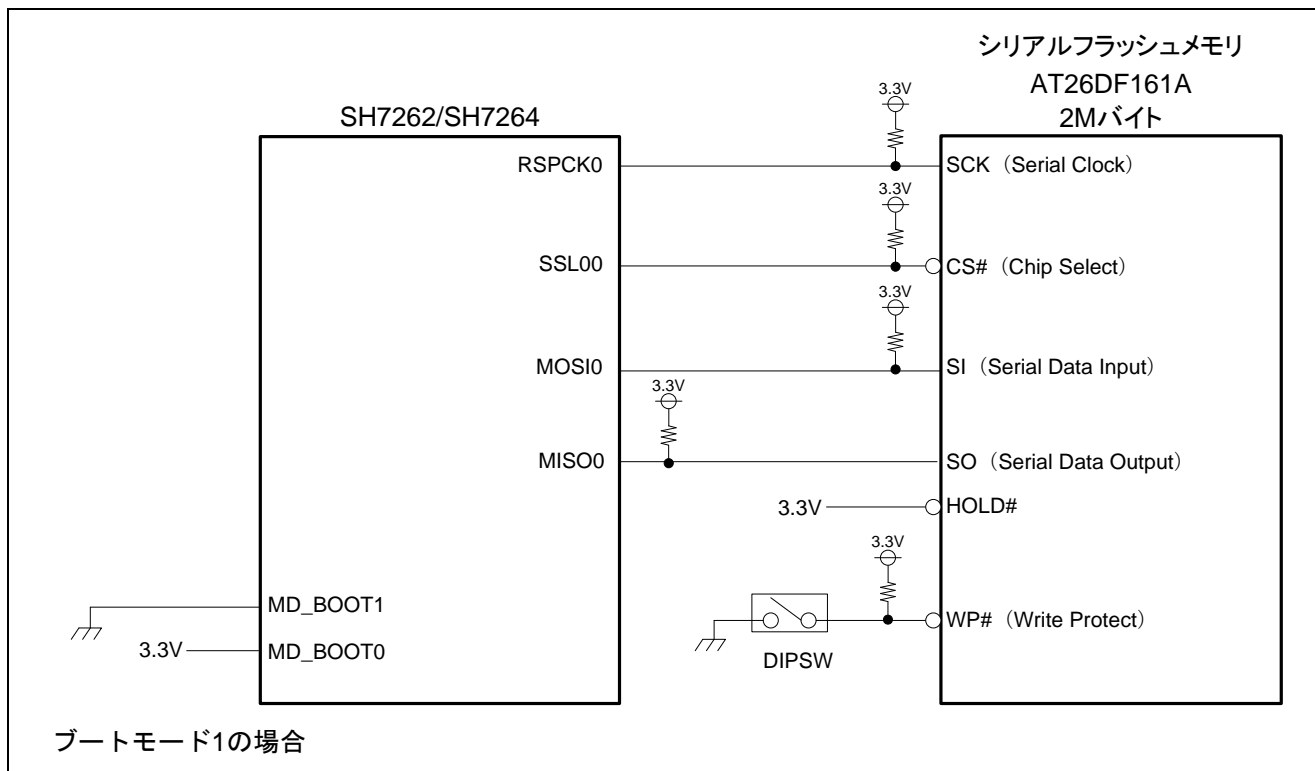


図4 シリアルフラッシュブートを行うための接続例

**【注】** ブートモード1の場合はバスクロック (B $\phi$ ) の2分の1のRSPIクロックを使用し、ブートモード3の場合はB $\phi$ の4分の1のRSPIクロックを使用します。シリアルフラッシュメモリとRSPI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트리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）。

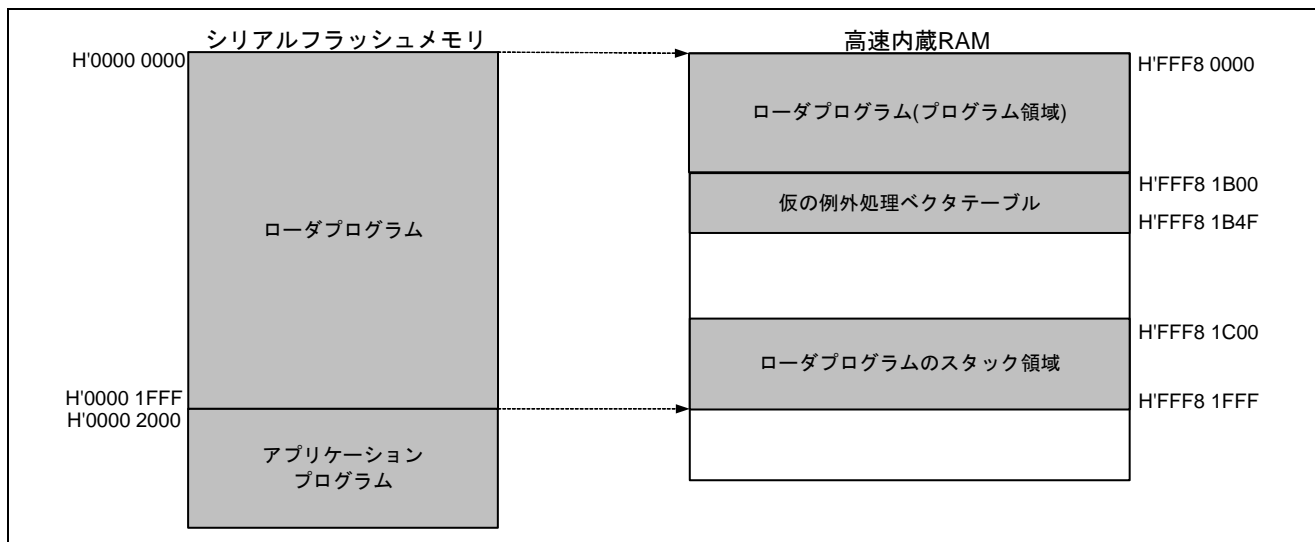


図5 ローダプログラムのメモリマップ

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。

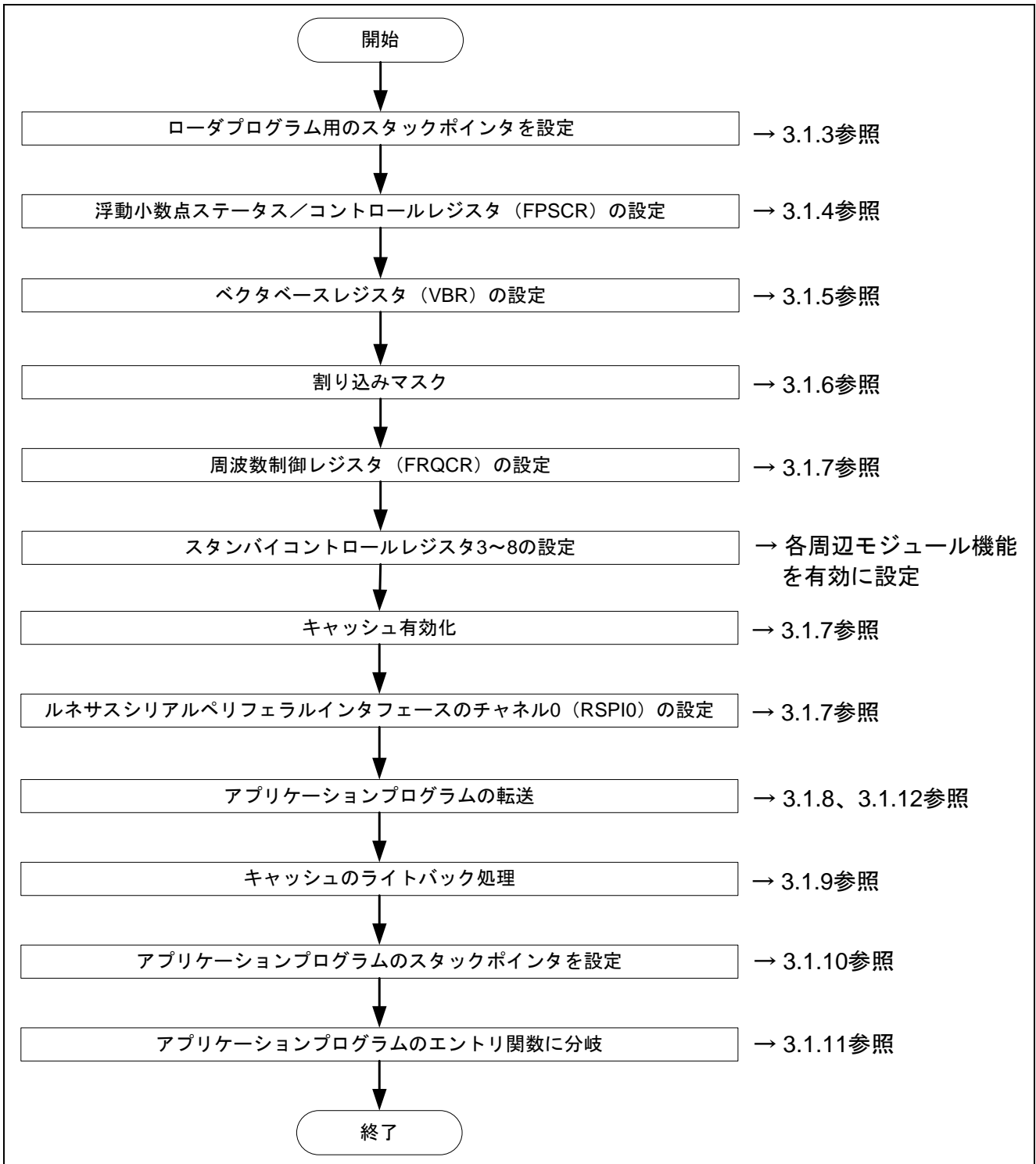


図6 ローダプログラムのフローチャート

### 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 SH7262 グループ、SH7264 グループ ハードウェアマニュアルの 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。

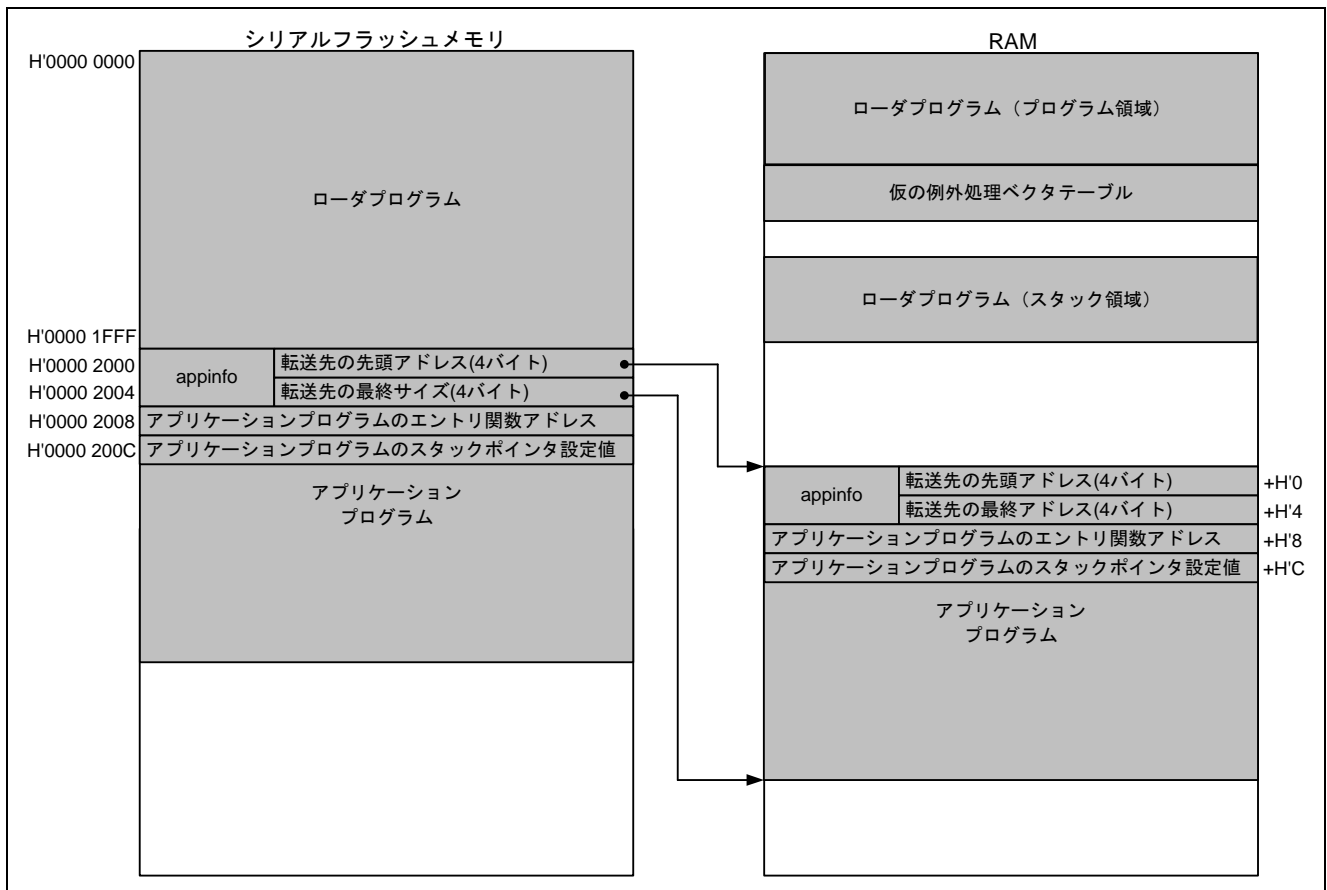


図7 アプリケーションプログラム転送イメージ図

### 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 Read Array 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
Read Array	H'0B	データの読み出し

【注】 本応用例では ATMEL 社製の AT26DF161A 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、SH7262グループ、SH7264 グループ ハードウェアマニュアル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

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'1003	
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	
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'48	
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'60	
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'01	
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'E700	
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'0000	

## 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トリ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トリ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。

図 8 にセクション配置例を示します。



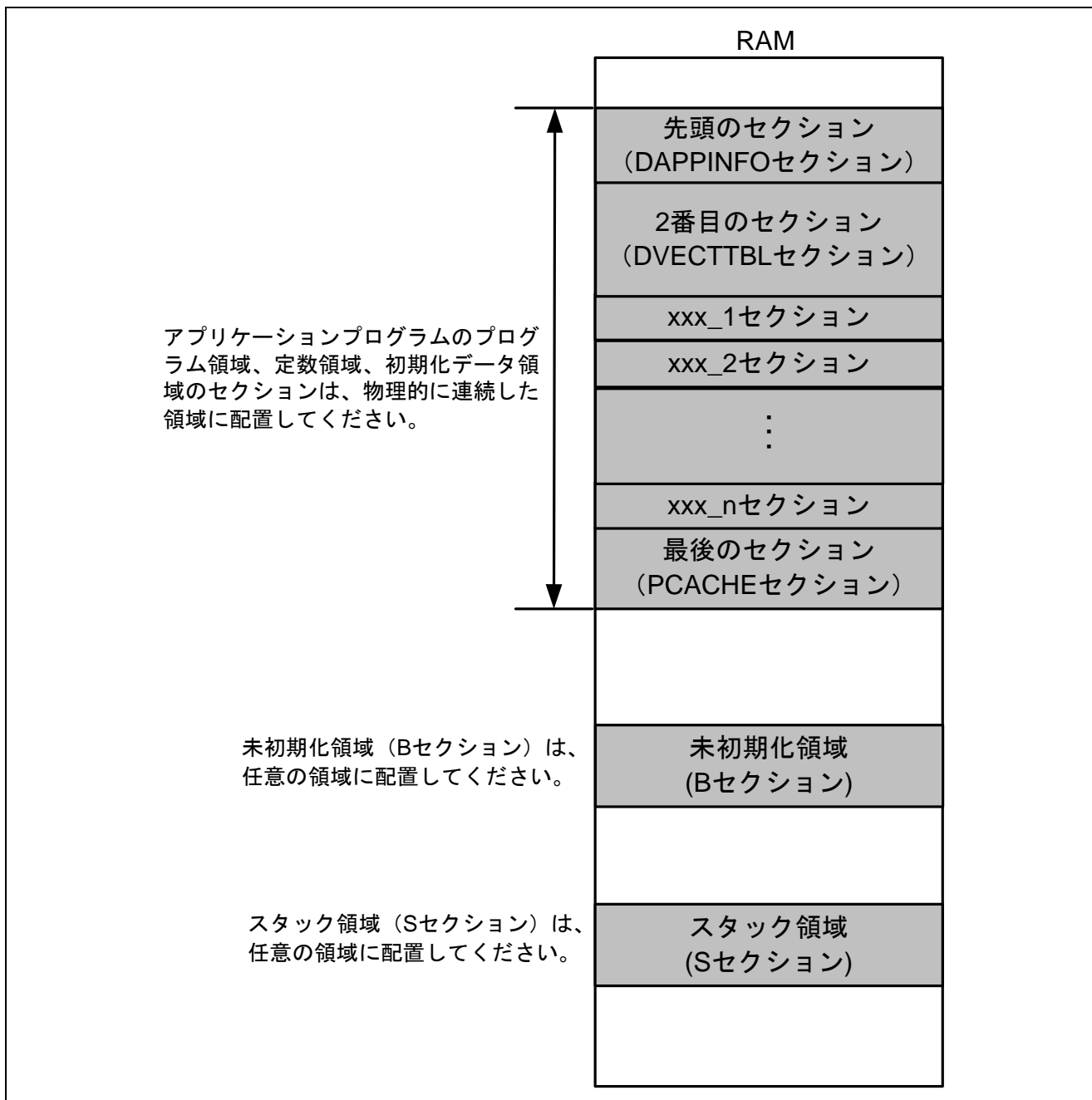


図8 アプリケーションプログラムのセクション配置例

### 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。

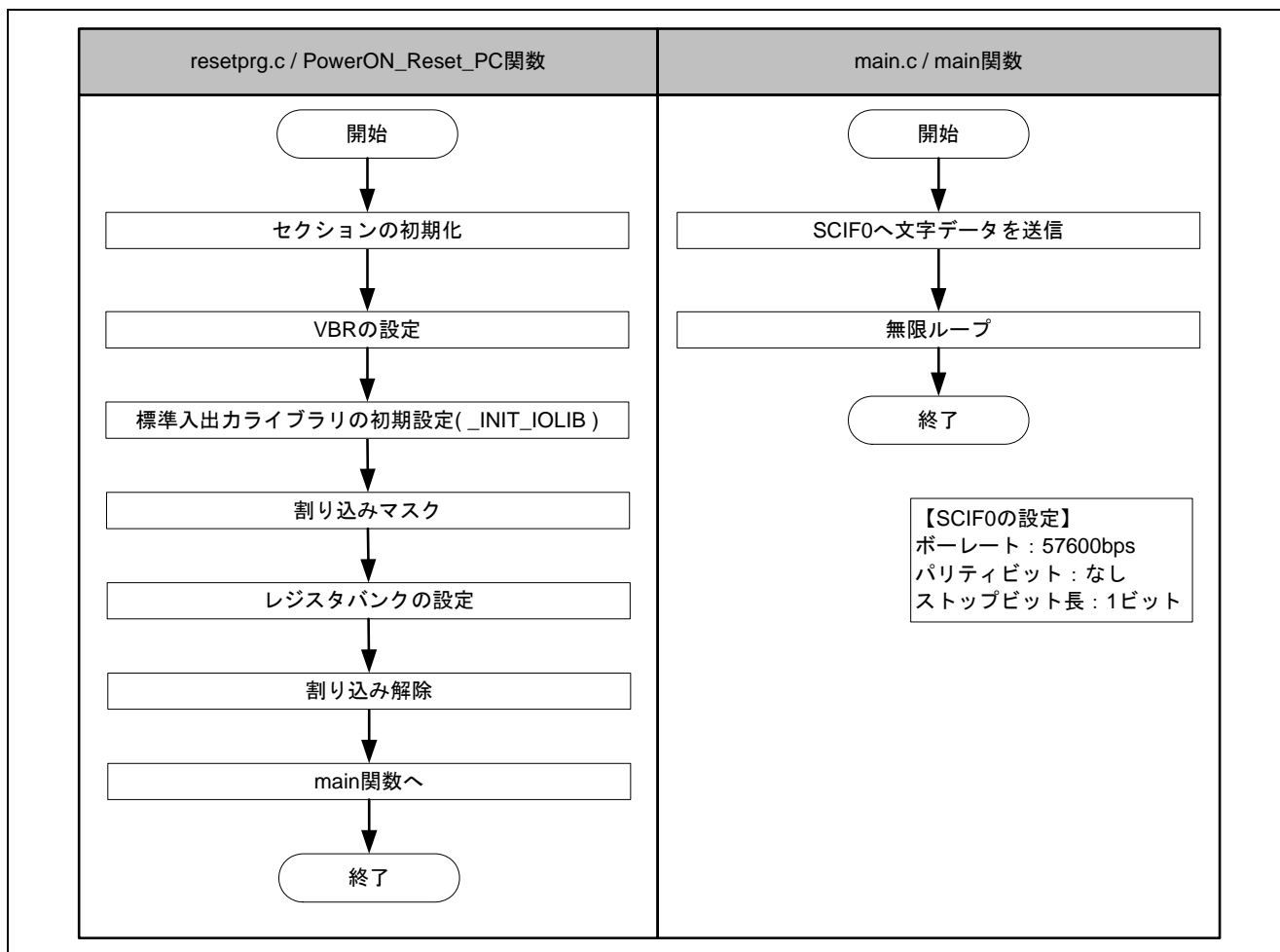


図9 アプリケーションプログラムのフローチャート

### 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。

### 3.2.6 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 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
	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 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。

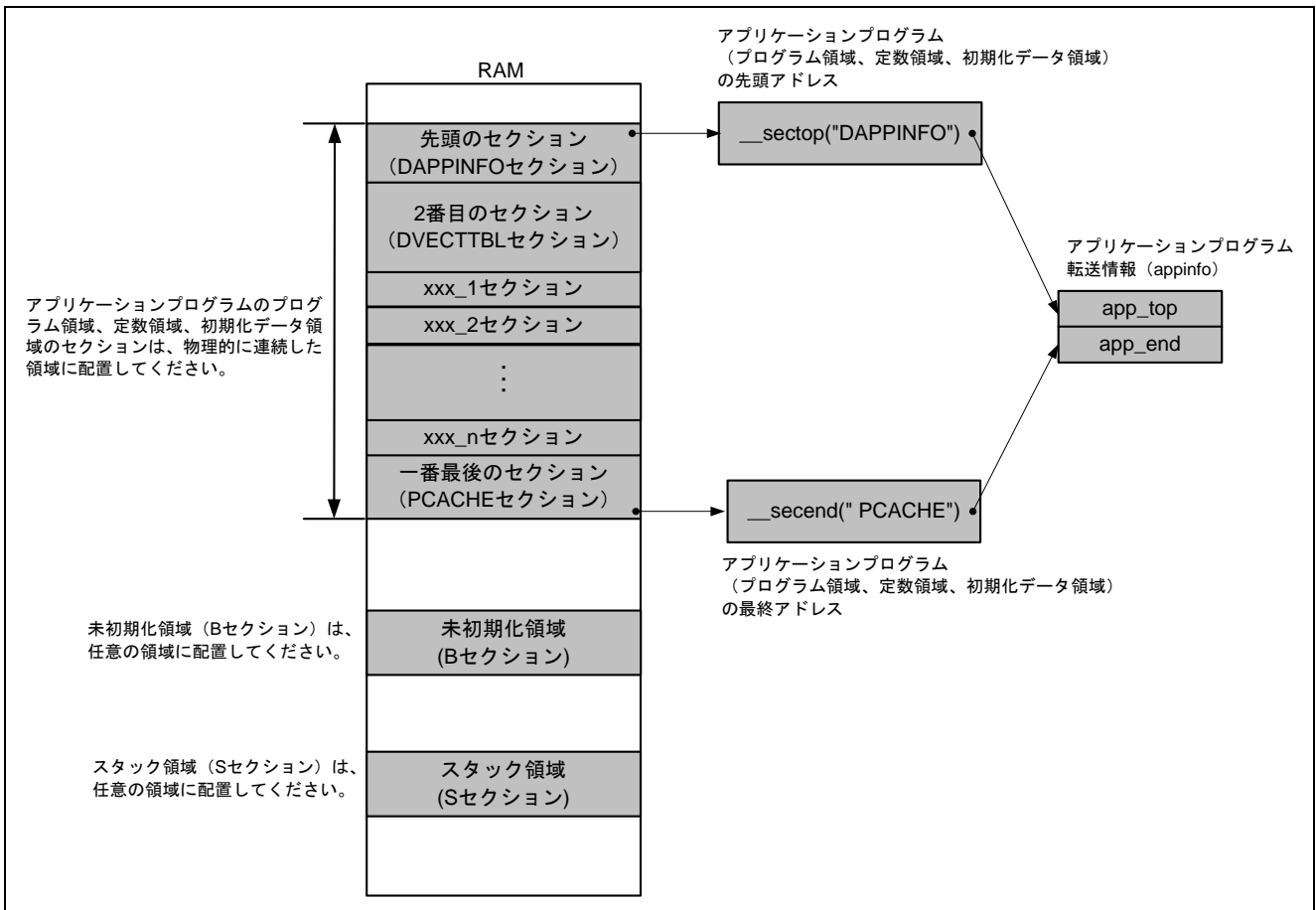


図10 アプリケーションプログラム転送情報 (appinfo) の生成イメージ図

### 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。

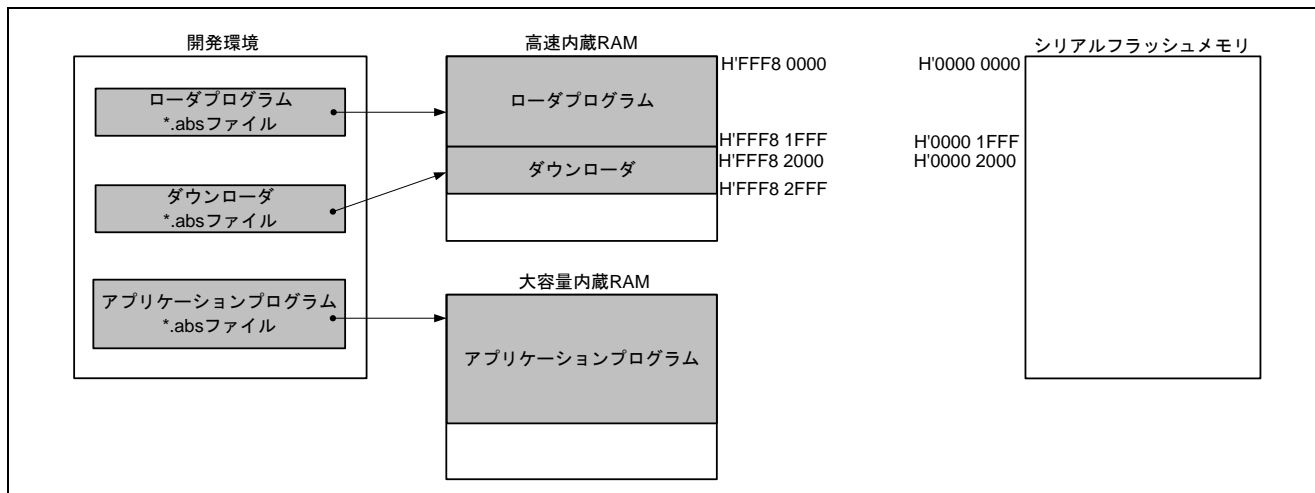


図11 ダウンローダ動作イメージ図 (1)

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。

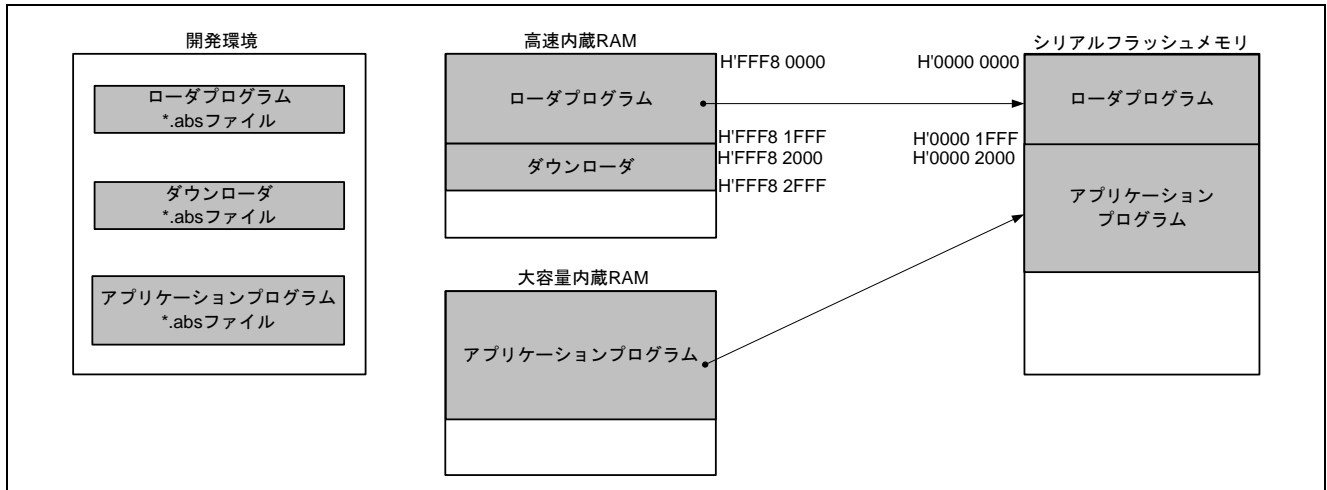


図12 ダウンローダ動作イメージ図 (2)

### 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。

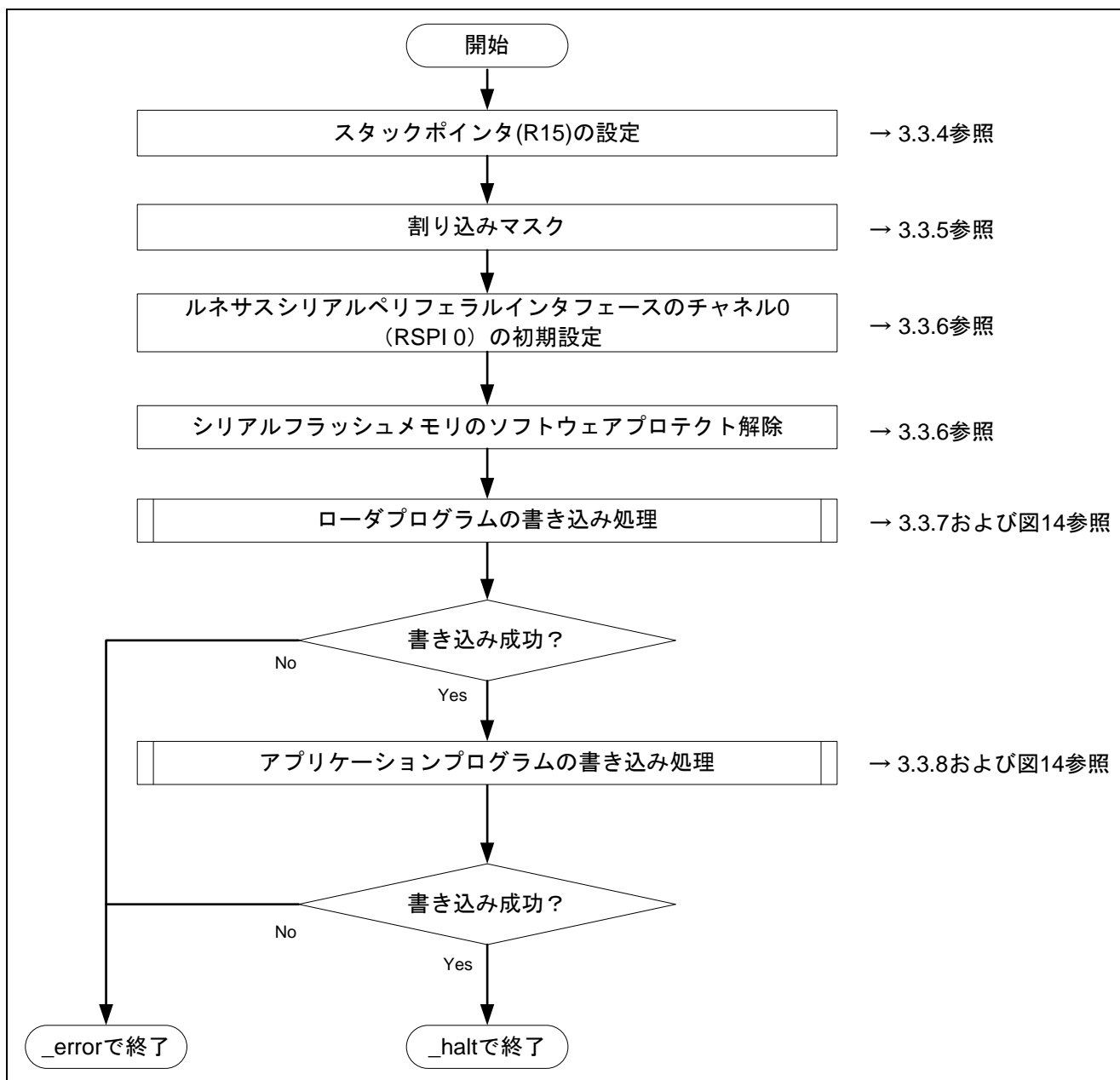


図13 ダウンローダのフローチャート



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。

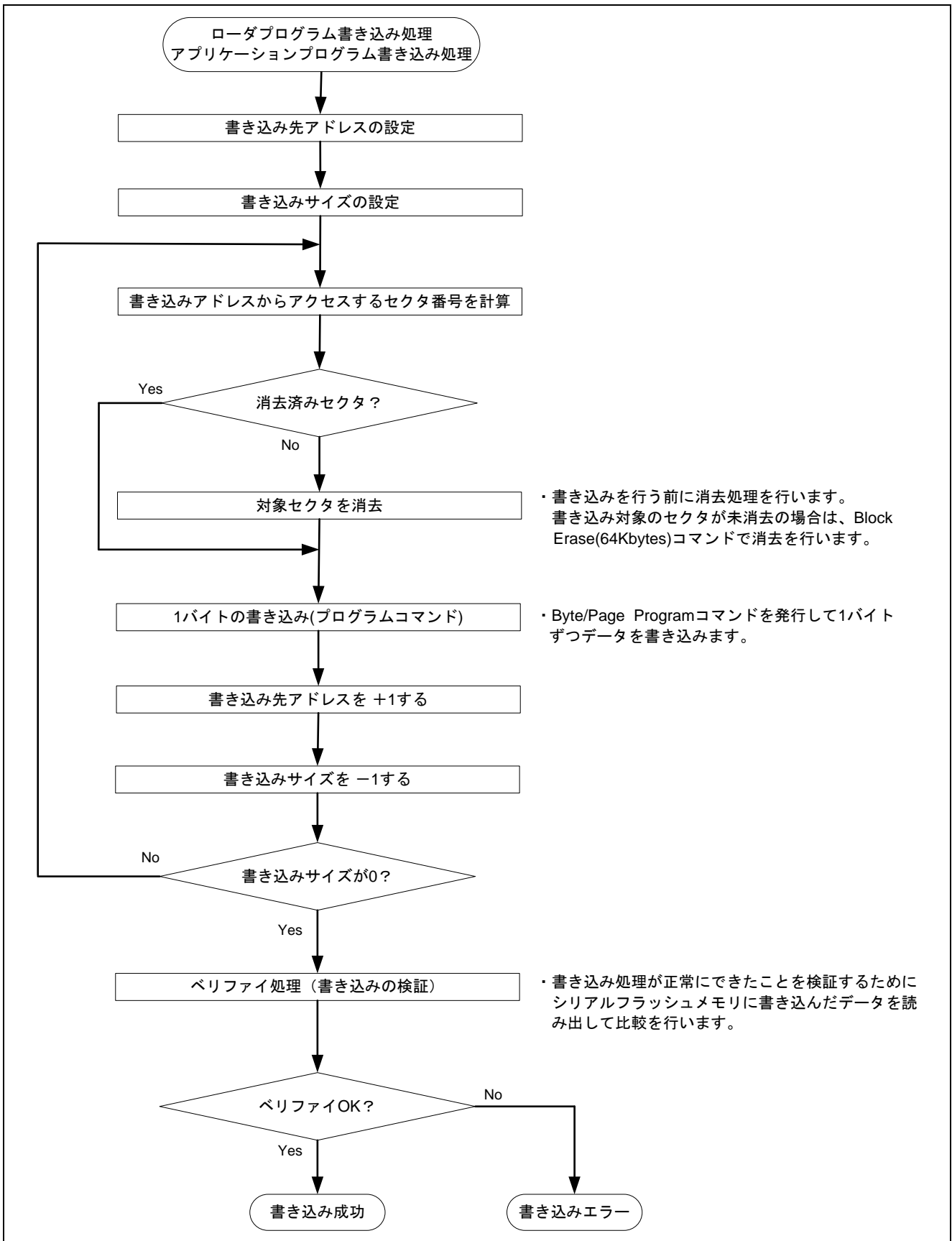


図14 書き込み処理のフローチャート

### 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 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	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)	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"> <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。書き込みは1バイト単位で行っています。</li> </ol>

### 3.3.9 シリアルフラッシュメモリのコマンド

表 11 にダウンロード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 (RSPIO) 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 11 ダウンロード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
Read Array	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/Page Program	H'02	データの書き込み（1～256 バイト）

- 【注】 1. 本応用例では ATMEL 社製の AT26DF161A 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。

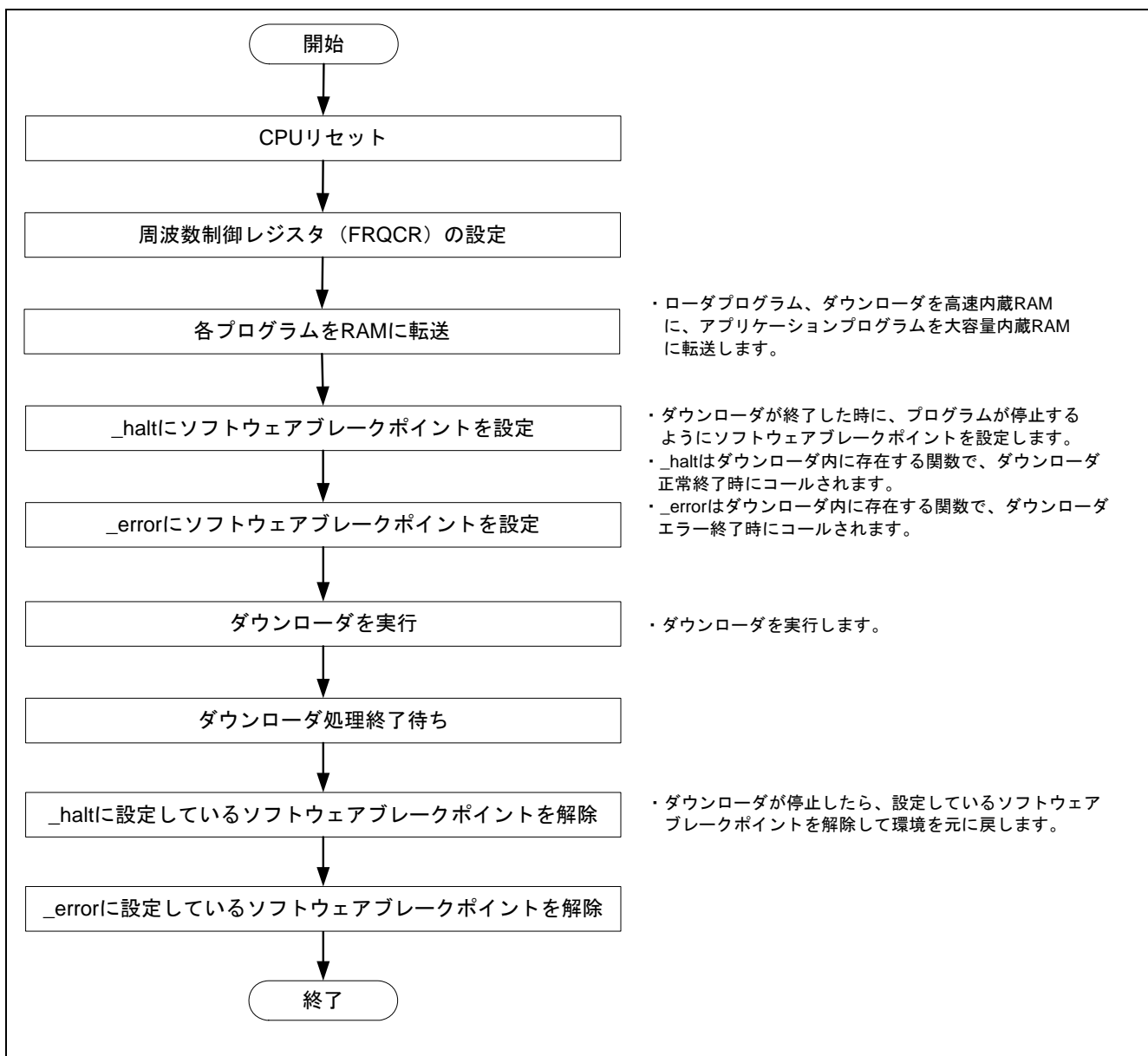


図15 コマンドバッチファイルのフローチャート

## 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      ;/"FILE COMMENT"***** Technical reference data *****
2      ;*
3      ;*   System Name : SH7264 Sample Program
4      ;*   File Name   : loader.src
5      ;*   Abstract   : ローダプログラムの前処理/アプリケーションプログラムへの
6      ;*                   : ジャンプ処理
7      ;*   Version    : 1.00.00
8      ;*   Device     : SH7264/SH7262
9      ;*   Tool-Chain : High-performance Embedded Workshop (Ver.4.04.01).
10     ;*                   : C/C++ compiler package for the SuperH RISC engine family
11     ;*                   :                               (Ver.9.02 Release00).
12     ;*   OS         : None
13     ;*   H/W Platform: M3A-HS64G50(CPU board)
14     ;*   Disclaimer  :
15     ;*               <注意事項>
16     ;*               本サンプルプログラムはすべて参考資料であり、
17     ;*               その動作を保証するものではありません。
18     ;*               本サンプルプログラムはお客様のソフトウェア開発時の
19     ;*               技術参考資料としてご利用ください。
20     ;*
21     ;*   The information described here may contain technical inaccuracies or
22     ;*   typographical errors. Renesas Technology Corporation and Renesas Solutions
23     ;*   assume no responsibility for any damage, liability, or other loss rising
24     ;*   from these inaccuracies or errors.
25     ;*
26     ;*   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
27     ;*   AND Renesas Solutions Corp. All Rights Reserved
28     ;*
29     ;*   History     : Dec.19.2008 Ver.1.00.00
30     ;/"FILE COMMENT END"*****/
31     .SECTION LOADER_ENTRY, CODE, ALIGN = 4
32     .IMPORT _main
33     .EXPORT _jmp_app_prog
34
35     _loader_prog:
36     MOV.L L2,R15 ; スタックポインタの設定
37     MOV.L L1,R0  ; ローダプログラムエントリ関数の取得
38     JMP @R0     ; ローダプログラムエントリ関数へ
39     NOP
40

```

## 4.1.2 ローダプログラムリスト"loader.src" (2)

```

41
42  ;/* "FUNC COMMENT"*****
43  ; * ID      :
44  ; * Outline  : アプリケーションプログラムへのジャンプ
45  ; *-----
46  ; * Include  :
47  ; *-----
48  ; * Declaration : _jmp_app_prog
49  ; *-----
50  ; * Description : 1.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
51  ; *              :   されているスタックポインタ設定値を取得します。
52  ; *              : 2.スタックポインタ (R15) の設定を行います。
53  ; *              : 3.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
54  ; *              :   されているエントリ関数のアドレスを取得します。
55  ; *              : 4.エントリ関数へジャンプします。
56  ; *-----
57  ; * Argument   : R4 ; I : アプリケーションプログラムの先頭アドレス
58  ; *-----
59  ; * Return Value: none
60  ;/* "FUNC COMMENT END"*****
61  _jmp_app_prog:
62
63  MOV.L R4,R0      ; R0 にアプリケーションプログラムの先頭アドレスを代入
64  ADD #12,R0      ; スタックポインタの設定値が格納されているアドレスを計算し R0 に代入
65  MOV.L @R0,R15   ; スタックポインタの設定
66
67  MOV.L R4,R0      ; R0 にアプリケーションプログラムの先頭アドレスを代入
68  ADD #8,R0       ; アプリケーションプログラムのエントリが格納されているアドレスを計算し R0 に代入
69  MOV.L @R0,R0    ; R0 にアプリケーションプログラムのエントリ関数のアドレスを代入
70  JMP @R0         ; アプリケーションプログラムエントリ関数へのジャンプ
71  NOP
72
73
74  .ALIGN 4
75  L1:
76  .DATA.L _main          ; ローダプログラムのエントリ関数アドレス
77
78  L2:
79  .DATA.L H'FFF82000    ; ローダプログラムのスタックポインタ (R15) 設定値
80
81  .pool
82  .end
83
84  ;/* End of File */
85
    
```

## 4.1.3 ローダプログラムリスト"main.c" (1)

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : main.c
5  *   Abstract    : ローダプログラム
6  *   Version     : 1.00.00
7  *   Device      : SH7264/SH7262
8  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  *                 : C/C++ compiler package for the SuperH RISC engine family
10 *                 :                 (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Dec.19.2008 Ver.1.00.00
29 *"FILE COMMENT END"*****/
30 #include <stdio.h>
31 #include <string.h>
32 #include <machine.h>
33 #include "iodefine.h"
34 #include "serial_flash.h"
35
36 /* ==== macro defined ==== */
37 #define FPSCR_INIT    0x00040001    /* FPSCR レジスタへの設定値 */
38 #define INT_MASK      0x000000F0    /* SR レジスタへの設定値(割り込みマスク用) */
39
40 #define APROG_TOP_SFLASH 0x00002000 /* アプリケーションプログラムの先頭アドレス */
41                                     /* (シリアルフラッシュメモリ) */
    
```



#### 4.1.4 ローダプログラムリスト"main.c" (2)

```

42
43 #define APPINFO_TOP    APROG_TOP_SFLASH    /* appinfo.app_top の配置アドレス */
44 #define APPINFO_END    (APROG_TOP_SFLASH + 4) /* appinfo.app_end の配置アドレス */
45
46
47 /* ==== prototype declaration ==== */
48 void main(void);
49 void get_appinfo( unsigned long *app_top_addr, unsigned long *app_end_addr);
50 void app_prog_transfer(unsigned long app_top_addr, unsigned long app_end_addr);
51 void system_down(void);
52
53 extern void jmp_app_prog(unsigned long app_top_addr);
54 extern void io_set_cpg(void);
55 extern void sf_byte_read_long(unsigned long addr, unsigned long *buf, int size);
56
57 /* ==== external data ==== */
58 extern unsigned long DUMMY_Vectors;
59
60
    
```

## 4.1.5 ローダプログラムリスト"main.c" (3)

```

61  /*"FUNC COMMENT"*****
62  * ID      :
63  * Outline   : ローダプログラム main
64  *-----
65  * Include   : #include "serial_flash.h"
66  *-----
67  * Declaration : void main(void);
68  *-----
69  * Description : 本関数は、appinfo のデータを元にアプリケーションプログラムを
70  *               : 大容量内蔵 RAM 上に転送し、アプリケーションプログラムのエントリ関数へ遷移します。
71  *-----
72  * Argument   : void
73  *-----
74  * Return Value: void
75  /*"FUNC COMMENT END"*****/
76 void main(void)
77 {
78     unsigned long app_top,app_end;
79
80     /* FPSCR の設定 */
81     set_fpscr(FPSCR_INIT);
82
83     /* 仮の VBR の設定 */
84     set_vbr((void *)&DUMMY_Vectors);
85
86     /* 割り込みマスク */
87     set_cr(INT_MASK);
88
89     /* CPG の設定 */
90     io_set_cpg();
91
92     /* キャッシュ有効化 */
93     io_init_cache();
94
95     /* RSPIO の設定 */
96     sf_init_serial_flash();
97
98     /* appinfo の取得 */
99     get_appinfo(&app_top,&app_end);
100
101     /* アプリケーションプログラムを大容量内蔵 RAM に転送 */
102     app_prog_transfer(app_top, app_end);
103
104     /* キャッシュのライトバック処理 */
105     io_cache_writeback();
106
107     /* アプリケーションプログラムへジャンプ */
108     jmp_app_prog(app_top);
109
110     while(1){
111         /* LOOP */
112     }
113 }

```

## 4.1.6 ローダプログラムリスト"main.c" (4)

```

114
115 /*"FUNC COMMENT"*****
116 * ID      :
117 * Outline : appinfo の取得
118 *-----
119 * Include : #include "serial_flash.h"
120 *-----
121 * Declaration : void get_appinfo( unsigned long *app_top_addr,
122 *      :      unsigned long *app_end_addr);
123 *-----
124 * Description : 本関数は appinfo の取得を行います。
125 *      : シリアルフラッシュメモリの H'2000~H'2003 番地に格納されている
126 *      : appinfo.top を取得して、第 1 引数で渡されたアドレスに格納します。
127 *      : また、シリアルフラッシュメモリの H'2004~H'2007 番地に格納されて
128 *      : いる appinfo.end を取得して、第 2 引数で渡されたアドレスに格納します。
129 *-----
130 * Argument  : unsigned long app_top_addr ; 0 : アプリケーションプログラムの
131 *      :      転送先の先頭アドレス
132 *      : unsigned long app_end_addr ; 0 : アプリケーションプログラムの
133 *      :      転送先の最終アドレス
134 *-----
135 * Return Value: void
136 /*"FUNC COMMENT END"*****/
137 void get_appinfo( unsigned long *app_top_addr,unsigned long *app_end_addr)
138 {
139
140     /* appinfo.top の取得 */
141     sf_byte_read(APPINFO_TOP, (unsigned char *)app_top_addr, 4);
142
143     /* appinfo.end の取得 */
144     sf_byte_read(APPINFO_END, (unsigned char *)app_end_addr, 4);
145
146 }
147
    
```

## 4.1.7 ローダプログラムリスト"main.c" (5)

```

148  /*"FUNC COMMENT"*****
149  * ID      :
150  * Outline  : アプリケーションプログラムの転送
151  *-----
152  * Include  : #include "serial_flash.h"
153  *-----
154  * Declaration : void app_prog_transfer(unsigned long app_top_addr,
155  *      :                unsigned long app_end_addr);
156  *-----
157  * Description : アプリケーションプログラムのサイズを計算し、シリアルフラッシュ
158  *      : 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大容量内蔵 RAM
159  *      : に転送します(ロングワード転送するためにアプリケーションプログラムの
160  *      : サイズを4の倍数に切り上げます)。
161  *-----
162  * Argument  : unsigned long app_top_addr ; I : アプリケーションプログラムの
163  *      :                転送先の先頭アドレス
164  *      : unsigned long app_end_addr ; I : アプリケーションプログラムの
165  *      :                転送先の最終アドレス
166  *-----
167  * Return Value: void
168  /*"FUNC COMMENT END"*****/
169  void app_prog_transfer(unsigned long app_top_addr,unsigned long app_end_addr)
170  {
171      unsigned long app_prog_size;
172
173      /* アプリケーションプログラムのサイズ計算 */
174      app_prog_size = app_end_addr - app_top_addr;
175      if( ( app_prog_size & 0x00000003 ) != 0 ){
176          app_prog_size &= 0xFFFFF0;
177          app_prog_size += 4;          /* アプリケーションプログラムサイズを4の倍数に切り上げ */
178      }
179
180      /* アプリケーションプログラムを大容量内蔵RAMにロード */
181      sf_byte_read_long(APROG_TOP_SFLASH, (unsigned long *)app_top_addr, app_prog_size);
182
183  }
184
    
```

## 4.1.8 ローダプログラムリスト"main.c" (6)

```

185  /*"FUNC COMMENT"*****
186  * ID      :
187  * Outline   : システム停止
188  *-----
189  * Include   :
190  *-----
191  * Declaration : void system_down(void);
192  *-----
193  * Description : 本関数は無限ループです。
194  *              : 本関数は DUMMY_Vectors テーブルに登録されているため、
195  *              : ロードプログラム動作中に例外が発生した場合は、本関数が
196  *              : コールされます。
197  *-----
198  * Argument   : void
199  *-----
200  * Return Value: void
201  /*"FUNC COMMENT END"*****/
202  void system_down(void)
203  {
204      while(1){
205          /* システムエラー */
206      }
207  }
208
209  /* End of File */
210

```

## 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  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : main.c
5  *   Abstract    : アプリケーションプログラム例
6  *   Version     : 1.00.00
7  *   Device      : SH7264/SH7262
8  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  *               : C/C++ compiler package for the SuperH RISC engine family
10 *               : (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Dec.19.2008 Ver.1.00.00
29 *"FILE COMMENT END"*****/
30 #include <stdio.h>
31
32 /* ==== prototype declaration ==== */
33 void main(void);
34
35
36 /*"FUNC COMMENT"*****
37 * ID          :
38 * Outline     : アプリケーションプログラムの main 関数
39 *-----
40 * Include    :
41 *-----
42 * Declaration : void main(void);
43 *-----
44 * Description : 本関数は、SCIF0 に文字列データを送信します。
45 *               : (ボーレート 57600bps、パリティなし、ストップビット長:1ビット)
46 *-----
47 * Argument   : void
48 *-----
49 * Return Value: void
50 *"FUNC COMMENT END"*****/

```

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

```

51     void main(void)
52     {
53         puts("==== Serial Flash Boot Done. ====");
54         fflush(stdout);
55
56         while(1){
57             /* loop */
58         }
59     }
60
61     /* End of File */
62

```

### 4.2.3 アプリケーションプログラムリスト"appinfo.c"

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : appinfo.c
5  *   Abstract    : アプリケーションプログラム転送情報(appinfo)の生成
6  *   Version     : 1.00.00
7  *   Device      : SH7264/SH7262
8  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  *                : C/C++ compiler package for the SuperH RISC engine family
10 *                :                (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Dec.19.2008 Ver.1.00.00
29 *"FILE COMMENT END"*****/
30 #include "appinfo.h"
31
32 #pragma section APPINFO
33
34 static APPINFO appinfo = {
35     __sectop("DAPPINFO"), /* ここには、アプリケーションプログラム (プログラム領域、定数領域、 */
36                          /* 初期化データ領域) の先頭セクションの先頭アドレスを格納します */
37
38     __secend("PCACHE")   /* ここには、アプリケーションプログラム (プログラム領域、定数領域、 */
39                          /* 初期化データ領域) の最後のセクションの最終アドレスを格納します */
40
41 };
42
43 /* End of File */
44
45

```



#### 4.2.4 アプリケーションプログラムリスト"appinfo.h"

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : appinfo.h
5  *   Abstract    : アプリケーションプログラム転送情報(appinfo)のヘッダファイル
6  *   Version     : 1.00.00
7  *   Device      : SH7264/SH7262
8  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  *                 : C/C++ compiler package for the SuperH RISC engine family
10 *                 :                 (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Dec.19.2008 Ver.1.00.00
29 *"FILE COMMENT END"*****/
30 #ifndef __APPINFO_H__
31 #define __APPINFO_H__
32
33 typedef struct appinfo_t {
34     void *app_top;           /* アプリケーションプログラムの先頭アドレス */
35     void *app_end;          /* アプリケーションプログラムの最終アドレス */
36 } APPINFO;
37
38
39 #endif /* __APPINFO_H__ */
40
41 /* End of File */
42

```

## 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  #/*****"FILE COMMENT"***** Technical reference data *****/
2  /**
3  /**      System Name : SH7264 Sample Program
4  /**      File Name   : downloader.hdc
5  /**      Abstract    : ダウンローダ用バッチファイル
6  /**      Version     : 1.00.00
7  /**      Device      : SH7264/SH7262
8  /**      Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  /**                          : C/C++ compiler package for the SuperH RISC engine family
10 /**                          :                               (Ver.9.02 Release00).
11 /**      OS          : None
12 /**      H/W Platform: M3A-HS64G50(CPU board)
13 /**      Disclaimer  :
14 /**                  <注意事項>
15 /**                  本サンプルプログラムはすべて参考資料であり、
16 /**                  その動作を保証するものではありません。
17 /**                  本サンプルプログラムはお客様のソフトウェア開発時の
18 /**                  技術参考資料としてご利用ください。
19 /**
20 /**      The information described here may contain technical inaccuracies or
21 /**      typographical errors. Renesas Technology Corporation and Renesas Solutions
22 /**      assume no responsibility for any damage, liability, or other loss rising
23 /**      from these inaccuracies or errors.
24 /**
25 /**      Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26 /**      AND Renesas Solutions Corp. All Rights Reserved
27 /**
28 /**      History      : Dec.19.2008 Ver.1.00.00
29 /*******"FILE COMMENT END"*****/
30
31
32  tcl enable
33
34
35  #Macro downloader -Start
36  proc init_hardware {} {
37
38      # CPG の設定
39      # FRQCR I=144MHz/B=72MHz/P=36MHz/CLK MODE2
40      MF H'FFFE0010 H'FFFE0011 H'1003 WORD
41
42  }
43
44
    
```

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

```

45     proc downloader {} {
46         # CPU リセット
47         reset
48
49         # init_hardware ルーチンの呼び出し
50         init_hardware
51
52         # High-performance Embedded Workshop に登録されている[Download modules]を全てダウンロード
53         file_load_all
54
55         # 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)
56         sh2a_sbstk enable
57
58         # _halt(main.c 参照)にソフトウェアブレークポイントを設定
59         set_disassembly_soft_break _halt set
60
61         # _error(main.c 参照)にソフトウェアブレークポイントを設定
62         set_disassembly_soft_break _error set
63
64         # _downloader(downloader.src 参照)を実行し、停止するまで待つ
65         go wait _downloader
66
67         # _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
68         set_disassembly_soft_break _halt clear
69
70         # _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
71         set_disassembly_soft_break _error clear
72
73     }
74
75     downloader
76     #Macro downloader -End
77
78
79
80     # 【注】 "tcl", "reset", "file_load", "sh2a_sbstk", "set_disassembly_soft_break", "go"は、
81     # 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、
82     # マニュアルを参照してください。
83
84     # /* End of File */
85

```

### 4.3.3 ダウンローダプログラムリスト” downloader.src”

```

1      ;/*"FILE COMMENT"***** Technical reference data *****
2      ;*
3      ;*      System Name : SH7264 Sample Program
4      ;*      File Name   : downloader.src
5      ;*      Abstract   : ダウンローダ
6      ;*      Version    : 1.00.00
7      ;*      Device     : SH7264/SH7262
8      ;*      Tool-Chain : High-performance Embedded Workshop (Ver.4.04.01).
9      ;*                      : C/C++ compiler package for the SuperH RISC engine family
10     ;*                      :                               (Ver.9.02 Release00).
11     ;*      OS         : None
12     ;*      H/W Platform: M3A-HS64G50(CPU board)
13     ;*      Disclaimer :
14     ;*                  <注意事項>
15     ;*                  本サンプルプログラムはすべて参考資料であり、
16     ;*                  その動作を保証するものではありません。
17     ;*                  本サンプルプログラムはお客様のソフトウェア開発時の
18     ;*                  技術参考資料としてご利用ください。
19     ;*
20     ;*      The information described here may contain technical inaccuracies or
21     ;*      typographical errors. Renesas Technology Corporation and Renesas Solutions
22     ;*      assume no responsibility for any damage, liability, or other loss rising
23     ;*      from these inaccuracies or errors.
24     ;*
25     ;*      Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26     ;*      AND Renesas Solutions Corp. All Rights Reserved
27     ;*
28     ;*      History     : Dec.19.2008 Ver.1.00.00
29     ;*""FILE COMMENT END"*****
30     .SECTION DOWNLOADER_ENTRY, CODE, ALIGN = 4
31     .IMPORT _main
32
33     _downloader:
34     MOV.L L2,R15 ; スタックポインタの設定
35     MOV.L L1,R0  ; ダウンローダエントリ関数の取得
36     JMP @R0     ; ダウンローダエントリ関数へ
37     NOP
38
39     .ALIGN 4
40     L1:
41     .DATA.L _main ; ダウンローダのエントリ関数アドレス
42
43     L2:
44     .DATA.L H'FFF83000 ; ダウンローダのスタックポインタ(R15)設定値
45
46     .pool
47     .end
48
    
```

## 4.3.4 ダウンローダプログラムリスト"main.c" (1)

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : main.c
5  *   Abstract    : ダウンローダ
6  *   Version     : 1.00.00
7  *   Device      : SH7264/SH7262
8  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9  *                 : C/C++ compiler package for the SuperH RISC engine family
10 *                 :                               (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Dec.19.2008 Ver.1.00.00
29 *"FILE COMMENT END"*****/
30 #include <stdio.h>
31 #include <string.h>
32 #include <machine.h>
33 #include "iodefine.h"
34 #include "serial_flash.h"
35
36 /* ==== macro defined ==== */
37 #define INT_MASK 0x000000F0 /* SR レジスタへの設定値(割り込みマスク用) */
38
39 #define SECTOR_SIZE 0x10000 /* セクタサイズ:64KB */
40 #define SECTOR_NUM 32 /* デバイスの総セクタ数 */
41 #define DEVICE_SIZE (SECTOR_SIZE * SECTOR_NUM) /* デバイスのサイズ */
42
43 #define L_PROG_SIZE 8192 /* ロードプログラムのサイズ */
44 #define L_PROG_SRC 0xFFFF8000 /* ロードプログラムの転送元アドレス */
45 #define L_PROG_DST 0x00000000 /* ロードプログラムの転送先アドレス*/
46
47 #define APROG_TOP_SFLASH 0x00002000 /* アプリケーションプログラムの先頭アドレス */
48 *
    
```

## 4.3.5 ダウンローダプログラムリスト”main.c”（2）

```

49
50 #define APROG_TOP_RAM      0x1C000000 /* アプリケーションプログラムの先頭アドレス      */
51                               /* アプリケーションプログラムの先頭セクションを変更した場合は */
52                               /* この定義も変更してください。      */
53
54 #define APPINFO_TOP      APROG_TOP_RAM      /* appinfo.app_top の配置アドレス*/
55 #define APPINFO_END ( APROG_TOP_RAM + 4 ) /* appinfo.app_end の配置アドレス*/
56
57
58
59 /* ==== prototype declaration ==== */
60 /*** User API *****/
61 void main(void);
62
63 static void halt(void);
64 static void error(void);
65 static void init_erase_flag(void);
66 static int Is_erased_sector(unsigned long sector_no);
67 static void set_erase_flag(unsigned long sector_no);
68 static int write_prog_data(unsigned char *program_data, unsigned long sflash_addr,
69                             unsigned long size);
70
71
72 /*** data ***/
73 static unsigned char sflash_erase_flag[SECTOR_NUM]={0}; /* 0:未消去のセクタ、1:消去済みセクタ */
74
75
76
77 /*"FUNC COMMENT"*****
78 * ID      :
79 * Outline   : ダウンローダ main
80 *-----
81 * Include   :
82 *-----
83 * Declaration : void main(void);
84 *-----
85 * Description : 本関数は、以下の手順でシリアルフラッシュメモリに
86 *               : ロードプログラムおよびアプリケーションプログラムを書き込みます。
87 *               : 1.ダウンローダ動作中の割り込みをマスクします。
88 *               : 2.RSPI0 の初期設定をします。
89 *               : 3.シリアルフラッシュメモリのソフトウェアプロテクトを解除します。
90 *               : 4.ロード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。
91 *               : 5.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
92 *               :   書き込みます。
93 *-----
94 * Argument   : void
95 *-----
96 * Return Value: void
97 *"FUNC COMMENT END"*****
    
```

## 4.3.6 ダウンローダプログラムリスト”main.c”（3）

```

98     void main(void)
99     {
100         unsigned long app_top_addr,app_end_addr,app_prog_size;
101
102         /* 割り込みマスク */
103         set_cr(INT_MASK);
104
105         /* 消去フラグの初期化 */
106         init_erase_flag();
107
108         /* RSPIO の初期設定 */
109         sf_init_serial_flash();
110
111         /* シリアルフラッシュメモリのソフトウェアプロテクト解除 */
112         sf_protect_ctrl(SF_REQ_UNPROTECT);
113
114
115         /* ロードプログラムの書き込み処理 */
116         if( write_prog_data( (unsigned char *)L_PROG_SRC, L_PROG_DST, L_PROG_SIZE) < 0 ){
117             error();
118         }
119
120         /* アプリケーションプログラム転送情報(appinfo)から先頭アドレスと最終アドレスを取得する */
121         app_top_addr = *(volatile unsigned long *)APPINFO_TOP;
122         app_end_addr = *(volatile unsigned long *)APPINFO_END;
123
124         /* アプリケーションプログラムのサイズ計算 */
125         app_prog_size = app_end_addr - app_top_addr;
126
127
128         /* アプリケーションプログラムの書き込み処理 */
129         if( write_prog_data( (unsigned char *)app_top_addr, APROG_TOP_SFLASH, app_prog_size) < 0 ){
130             error();
131         }
132
133         /* シリアルフラッシュメモリのソフトウェアプロテクト有効 */
134         sf_protect_ctrl(SF_REQ_PROTECT);
135
136         /* ダウンローダ終了 */
137         halt();
138     }
139
140     /*"FUNC COMMENT"*****
141     * ID      :
142     * Outline  : プログラムデータの書き込み
143     *-----
144     * Include  :
145     *-----
    
```

## 4.3.7 ダウンローダプログラムリスト”main.c”（4）

```

146 * Declaration : int write_prog_data(unsigned char *program_data,
147 *           :           unsigned long sflash_addr, unsigned long size);
148 *-----
149 * Description : 本関数は、プログラムデータの書き込み処理を以下の手順で行います。
150 *           : 1.書き込み対象のセクタが未消去である場合は、セクタを消去します。
151 *           : 2.シリアルフラッシュメモリにプログラムデータを書き込みます。
152 *           : 3.シリアルフラッシュメモリに書いたデータを読み出し、
153 *           :   書いたデータと読み出したデータを比較します。
154 *-----
155 * Argument   : unsigned char *program_data ; I : プログラムデータの先頭アドレス
156 *           : unsigned long sflash_addr  ; I : フラッシュメモリの書き込み先の先頭アドレス
157 *           : unsigned long size        ; I : 書き込みサイズ
158 *-----
159 * Return Value: 0 以上 : Success
160 *           : 0 未満 : Error
161 * "FUNC COMMENT END"*****/
162 int write_prog_data(unsigned char *program_data, unsigned long sflash_addr, unsigned long size)
163 {
164     unsigned long sector_no;
165     unsigned long saddr;
166     unsigned long sz;
167     unsigned char read_data;
168     unsigned char *w_p;
169
170     /* ==== 引数から値をローカル変数にコピー ==== */
171     saddr = sflash_addr;
172     sz = size;
173     w_p = program_data;
174
175     /* ==== シリアルフラッシュメモリへの書き込み ==== */
176     while( sz > 0){
177         sector_no = saddr / SECTOR_SIZE;
178         if( Is_erased_sector(sector_no) == 0 ){ /* 未消去であれば */ /*
179             sf_sector_erase(sector_no); /* 消去する */ /*
180             set_erase_flag(sector_no); /* 消去したら、消去済みフラグを立てる */ /*
181         }
182
183         sf_byte_program(saddr, w_p, 1 ); /* 1 バイトずつ書き込む */ /*
184         w_p++;
185         saddr++;
186         sz--;
187     }
188
189     /* ==== ベリファイ処理（書き込みの検証） ==== */
190     saddr = sflash_addr;
191     sz = size;
192     w_p = program_data;
193
194     while( sz > 0){
195         sf_byte_read(saddr, &read_data, 1); /* 書き込みしたデータを読み出し */ /*
    
```



## 4.3.8 ダウンロードプログラムリスト"main.c" (5)

```

196
197     if( *w_p != read_data ){
198         return -1;                /* データが一致しない場合エラーとする */
199     }
200
201     w_p++;
202     saddr++;
203     sz--;
204 }
205
206 return 0;
207 }
208
209 /*"FUNC COMMENT"*****
210 * ID      :
211 * Outline  : 消去フラグの初期化
212 *-----
213 * Include  :
214 *-----
215 * Declaration : static void init_erase_flag(void);
216 *-----
217 * Description : 本関数は、テーブル sflash_erase_flag[] を初期化します。
218 *-----
219 * Argument   : void
220 *-----
221 * Return Value: void
222 *"FUNC COMMENT END"*****
223 static void init_erase_flag(void)
224 {
225     int i;
226
227     for( i=0; i < SECTOR_NUM ;i++){
228         sflash_erase_flag[i] = 0;
229     }
230 }
231
232 /*"FUNC COMMENT"*****
233 * ID      :
234 * Outline  : セクタの消去状態取得
235 *-----
236 * Include  :
237 *-----
238 * Declaration : static int Is_erased_sector(unsigned long sector_no);
239 *-----
240 * Description : 本関数は、セクタ番号を渡すと、指定したセクタの管理情報
241 *               : (未消去/消去済み)を返します。
242 *-----
243 * Argument   : unsigned long sector_no ; I : セクタ番号
244 *-----
245 * Return Value: 1 : 指定したアドレスのセクタは消去済み
246 *               : 0 : 指定したアドレスのセクタは未消去
247 *"FUNC COMMENT END"*****
    
```

## 4.3.9 ダウンローダプログラムリスト”main.c”（6）

```

248 static int Is_erased_sector(unsigned long sector_no)
249 {
250     return sflash_erase_flag[sector_no];
251 }
252
253 /*"FUNC COMMENT"*****
254 * ID      :
255 * Outline  :  消去フラグのセット
256 *-----
257 * Include  :
258 *-----
259 * Declaration : static void set_erase_flag(unsigned long sector_no);
260 *-----
261 * Description :  本関数は、指定したセクタの管理情報を消去済み状態にします。
262 *-----
263 * Argument   : unsigned long sector_no ; I : セクタ番号
264 *-----
265 * Return Value: void
266 *"FUNC COMMENT END"*****/
267 static void set_erase_flag(unsigned long sector_no)
268 {
269     sflash_erase_flag[sector_no] = 1;
270 }
271
272 /*"FUNC COMMENT"*****
273 * ID      :
274 * Outline  :  プログラム停止（成功時）
275 *-----
276 * Include  :
277 *-----
278 * Declaration : static void halt(void);
279 *-----
280 * Description :  ダウンローダ成功時、本関数がコールされてプログラムが停止します。
281 *-----
282 * Argument   : void
283 *-----
284 * Return Value: void
285 *"FUNC COMMENT END"*****/
286 static void halt(void)
287 {
288     while(1){
289         /* ダウンローダ成功時、本関数でプログラムが停止します。 */
290     }
291 }
    
```

## 4.3.10 ダウンローダプログラムリスト”main.c”（7）

```

292
293  /*"FUNC COMMENT"*****
294  * ID      :
295  * Outline  : プログラム停止 (エラー時)
296  *-----
297  * Include  :
298  *-----
299  * Declaration : static void error(void);
300  *-----
301  * Description : ダウンローダ失敗時、本関数がコールされてプログラムが停止します。
302  *-----
303  * Argument   : void
304  *-----
305  * Return Value: void
306  /*"FUNC COMMENT END"*****/
307  static void error(void)
308  {
309      while(1){
310          /* ダウンローダ失敗時、本関数でプログラムが停止します。 */
311          }
312      }
313
314  /* End of File */
315
    
```

## 5. ダウンローダの使用方法

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 12のように 3つのワークスペースに分かれています。

表12 参考プログラムの構成

ワークスペース名	説明
sh7264_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。
sh7264_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。
sh7264_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、[sh7264_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、[sh7264_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、[sh7264\_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[sh7264\_sflash\_app]ワークスペースのディレクトリ構成を示します。[sh7264\_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。

```

#sh7264_sflash_app          : ワークスペースディレクトリ
|-sh7264_sflash_app        : プロジェクトディレクトリ
|  |-debug                  :
|  |-sh7264_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
|  |-sh7264_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
|  |-sh7264_sflash_loader_prog.abs : ローダプログラムの実行ファイル-----④

```

図16 [sh7264\_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、[sh7264\_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。

- [sh7264\_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。
- ワークスペースディレクトリの中にある[sh7264\_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。
- 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。
- 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。
- 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。
- [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。

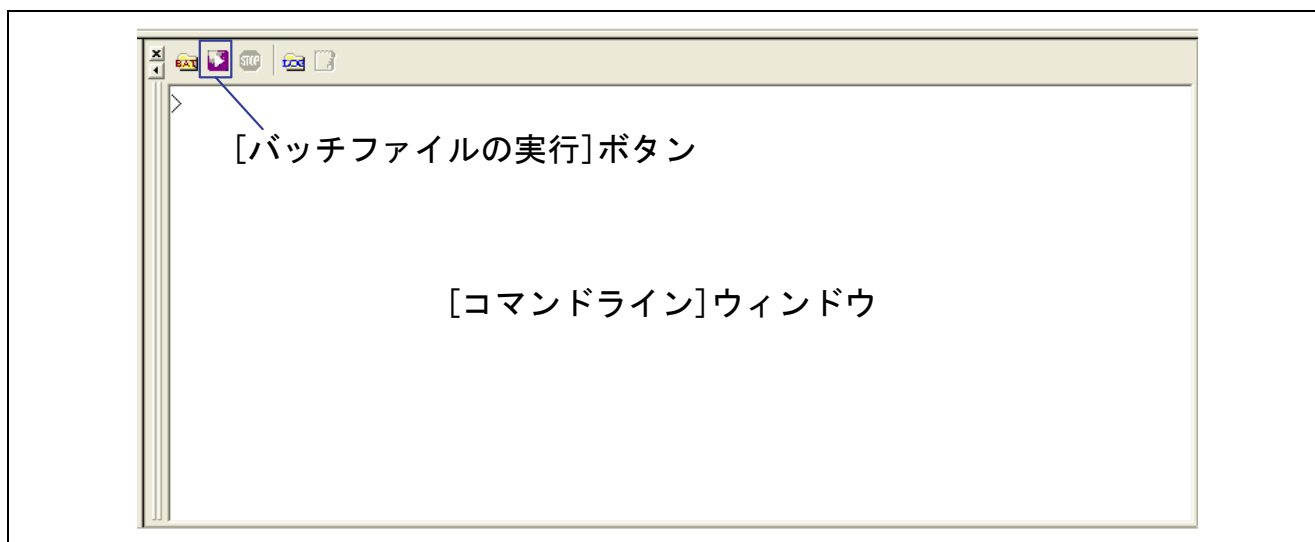


図17 [コマンドライン]ウィンドウと[バッチファイルの実行]ボタン

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。[sh7264\_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。

FFF820EA	AFFE	_halt	BRA	@_halt:12	
FFF820EC	0009		NOP		
FFF820EE	AFFE	_error	BRA	@_error:12	
FFF820F0	0009		NOP		
FFF820F2	0000FFF8		MOV.I20	#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	

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。

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。

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 テクノロジ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  
SH7262 グループ、SH7264 グループ ハードウェアマニュアル Rev.1.00  
(最新版をルネサス テクノロジ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 テクノロジホームページ

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

お問合せ先

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

[csc@renesas.com](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.03.23	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。



**本資料ご利用に際しての留意事項**

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますと、事前に弊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、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 本資料にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなく、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444