

SH7734 グループ

R01AN0667JJ0100

Rev.1.00

シリアルフラッシュメモリからのブート例

2012.09.24

要旨

本アプリケーションノートでは、パワーオンリセット時のSH7734に接続したシリアルフラッシュメモリからのブート例について説明します。

本書の位置づけ

本アプリケーションノートは、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」に記載されている初期設定例とそのサンプルコードを流用して、シリアルフラッシュメモリからのブート例を説明しています。SH7734 の初期設定に関する説明は省略していますので、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」のアプリケーションノートも合わせて参照ください。

対象デバイス

SH7734グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様	4
1.1 使用機能	4
1.2 シリアルブートに関連する用語	4
1.3 シリアルブートの動作概要	5
1.3.1 ブート起動用内蔵 ROM プログラムの動作概要	5
1.3.2 ローダプログラムの動作概要	6
1.3.3 アプリケーションプログラムの動作概要	6
1.3.4 ダウンローダの動作概要	7
2. 動作確認条件	8
3. 関連アプリケーションノート	8
4. 周辺機能説明	9
4.1 ブートモード	9
5. ハードウェア説明	10
5.1 ハードウェア構成例	10
5.2 使用端子一覧	11
6. ソフトウェア説明	12
6.1 サンプルコードの構成	12
6.2 各ワークスペースのソフトウェア説明の前に	12
6.2.1 記載について	12
6.2.2 用語について	13
7. ローダプログラム (RSPI) のソフトウェア説明	14
7.1 動作概要	14
7.1.1 メモリマップ	14
7.1.2 セクション配置	15
7.1.3 SPI モード、ビットレート	15
7.2 ファイル構成	16
7.3 定数一覧	17
7.4 関数一覧	19
7.5 関数仕様	20
7.6 フローチャート	23
8. アプリケーションプログラムのソフトウェア説明	26
8.1 動作概要	26
8.1.1 セクション配置	26
8.2 ファイル構成	28
8.3 フローチャート	29
8.4 アプリケーションプログラム転送情報 (appinfo) の生成	30
9. ダウンローダプログラムのソフトウェア説明	32
9.1 ダウンローダプログラムの仕様詳細	32
9.1.1 動作概要	32
9.1.2 セクション配置	33
9.2 ファイル構成	34
9.3 定数一覧	35
9.4 関数一覧	37
9.5 関数仕様	38
9.6 フローチャート	41
9.7 シリアルフラッシュメモリのコマンド	45

10.	応用例	46
10.1	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	46
10.1.1	ダウンロードの使用方法	46
10.1.2	ダウンロードの自動化 (コマンドバッチファイル)	47
10.1.3	ダウンロードモジュールとバッチファイルの登録	48
10.1.4	プログラム書き込み手順	49
10.2	PC ブレーク機能 (BREAKPOINT) 使用の注意事項	51
10.3	Quad-SPI 対応 ロードプログラム (参考)	52
10.3.1	サンプルコードの構成	52
10.3.2	Quad-SPI 対応 ロードプログラムの特徴	52
10.3.3	シリアルフラッシュメモリへの Quad-SPI 対応 ロードプログラムの書き込み方法	54
11.	サンプルコード	55
12.	参考ドキュメント	55

1. 仕様

ブートモードでシリアルブートを選択した場合、シリアルフラッシュメモリからブートします。本アプリケーションノートでは、シリアル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。

【参考】

本アプリケーションノートでは、シリアルフラッシュメモリへのアクセスには、ルネサスシリアルペリフェラルインタフェース (RSPI) を使用したケースの記載になっております。但し、ローダプログラム部分に関しては、転送時間を短縮するために、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 (Quad-SPI) を使用した別サンプルワークスペースを用意しています。詳細は、応用例 10.3 章を参照ください。

1.1 使用機能

表1.1 使用する周辺機能と用途

周辺機能	用途
ルネサスシリアルペリフェラルインタフェース (以下、RSPI)	シリアルフラッシュメモリへのアクセスに使用。 (Single 動作でのシリアル通信)
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 (以下、Quad-SPI)	シリアルフラッシュメモリへのアクセスに使用。 (Single/Dual/Quad-SPI 動作でシリアル通信) 【参考】ローダプログラムのみ別サンプルとして対応。
周辺モジュール用ダイレクトメモリアクセスコントローラ (以下、HPB-DMAC)	HPB バス上の Peripheral (本アプリケーションノートでは、RSPI or Quad-SPI) と DDR2/3-SDRAM 間の DMA 転送に使用します。

1.2 シリアルブートに関連する用語

表 1.2に本アプリケーションノートで使用するシリアルブート関連の用語を示します。

表1.2 シリアルブート関連の用語

用語	説明
ブート起動用内蔵 ROM プログラム	シリアルブートで起動した場合に、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 IL メモリに転送し、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。本プログラムは、あらかじめ CPU 内のブート起動用内蔵 ROM に格納されており、お客様が作成するプログラムではありません。
ローダプログラム	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 DDR2/3-SDRAM に転送し、アプリケーションプログラム本体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 16K バイト固定です。システムに応じて作成してください。
アプリケーションプログラム	お客様がシステムに応じて作成するプログラムです。
ダウンローダ	ローダプログラムとアプリケーションプログラムをシリアルフラッシュメモリへ書き込むためのプログラムです。システムに応じて作成してください。

1.3 シリアルブートの動作概要

シリアルブートは、ブート起動用内蔵 ROM プログラムの動作 ロードプログラム アプリケーションプログラムの順に動作します。以下に、各プログラムの動作概要を説明します。

1.3.1 ブート起動用内蔵 ROM プログラムの動作概要

シリアルブートモードの場合、LSI 起動後、ブート起動用内蔵 ROM プログラムにより、RSPI に接続されているシリアルフラッシュメモリからロードプログラムを IL メモリに転送します。転送完了後、ロードプログラムの先頭に分岐します。図 1.1 にブート起動用内蔵 ROM プログラムの動作概要図を示します。これらの一連の処理は、ハードウェアで自動的に行われます。

【注】ブート起動用内蔵 ROM プログラムは、Quad-SPI には対応していません。

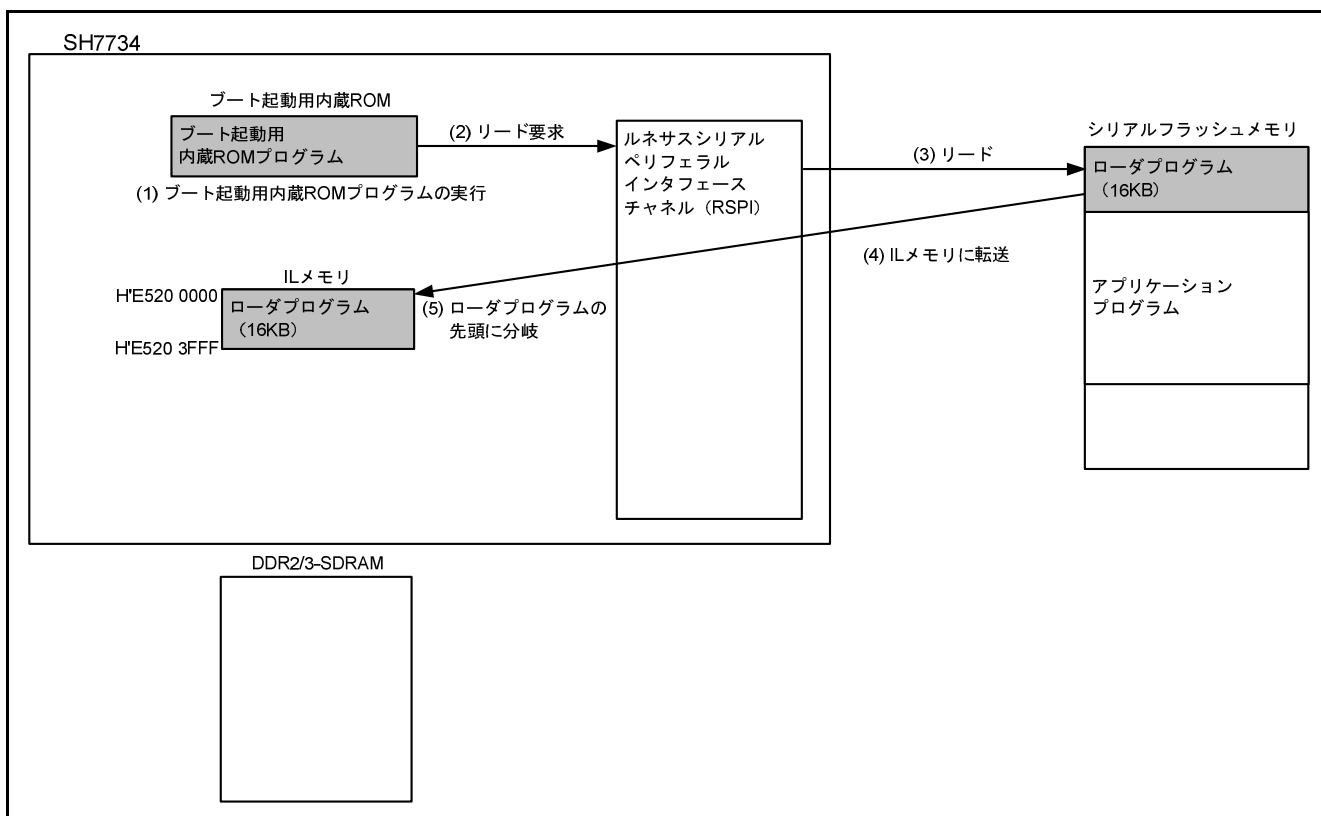


図1.1 ブート起動用内蔵 ROM プログラムの動作概要図

1.3.2 ローダプログラムの動作概要

ローダプログラムにより RSPI 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 DDR2/3-SDRAM に転送します。転送完了後、アプリケーションプログラム本体のエントリ関数に分岐します。図 1.2 にローダプログラムの動作概要図を示します。

詳細は「7章 ローダプログラム (RSPI) のソフトウェア説明」を参照してください。

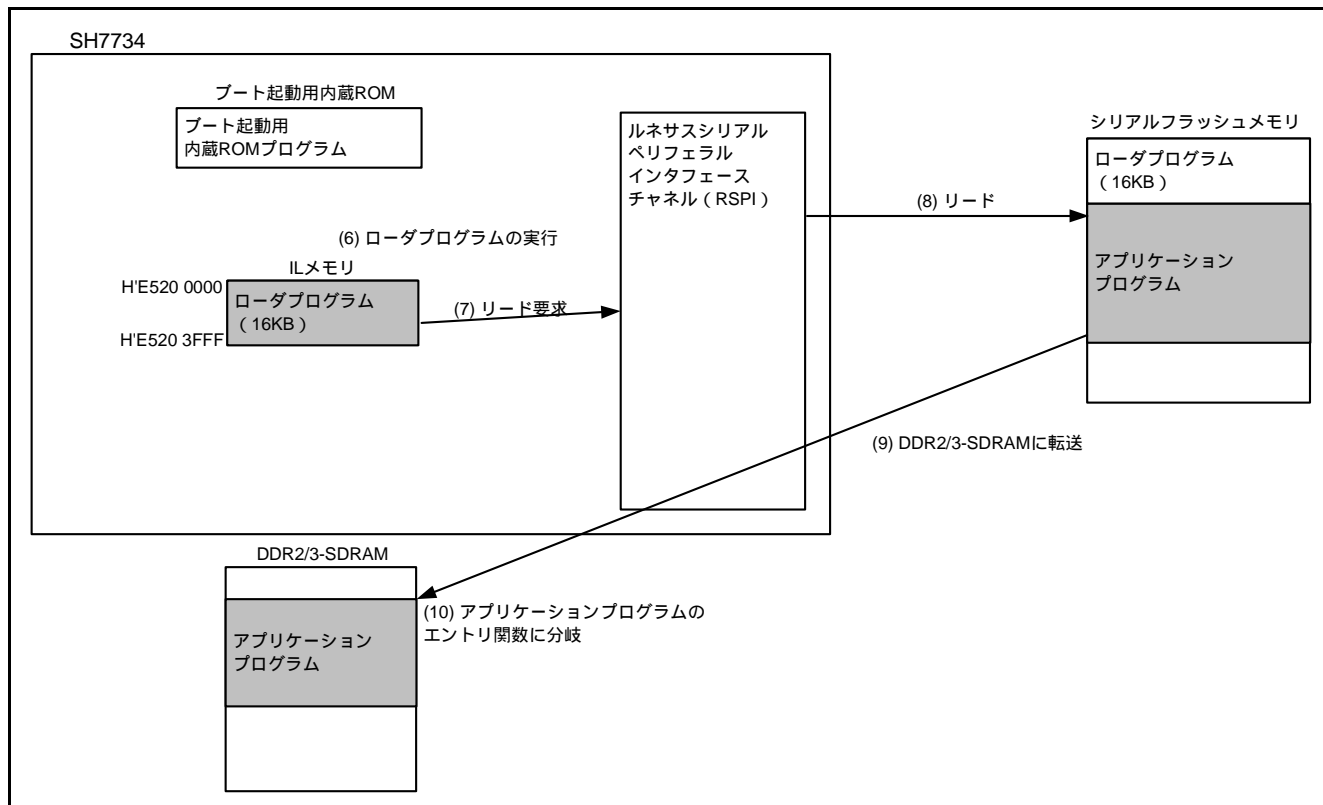


図1.2 ローダプログラムの動作概要図

【参考】

転送時間を短縮するために、Quad-SPI を使用した別サンプルコードを用意しています。詳細は、応用例 10.3章を参照ください。

1.3.3 アプリケーションプログラムの動作概要

お客様がシステムに応じて作成したプログラムが動作します。本アプリケーションノートでは、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」のサンプルコードを流用して、PC 上のターミナルソフトにデバッグ情報を出力する処理を行います。

詳細は「8章 アプリケーションプログラムのソフトウェア説明」を参照ください。

1.3.4 ダウンローダの動作概要

ダウンローダは、ILメモリ上に配置したローダプログラムとDDR2/3-SDRAM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。ダウンローダ実行前に、デバッグを使用して開発環境からOLメモリにダウンローダ、ILメモリにローダプログラム、DDR2-SDRAMにアプリケーションプログラムをそれぞれ転送します。図1.3にダウンローダの動作概要を示します。

詳細は「9章 ダウンローダプログラムのソフトウェア説明」を参照ください。

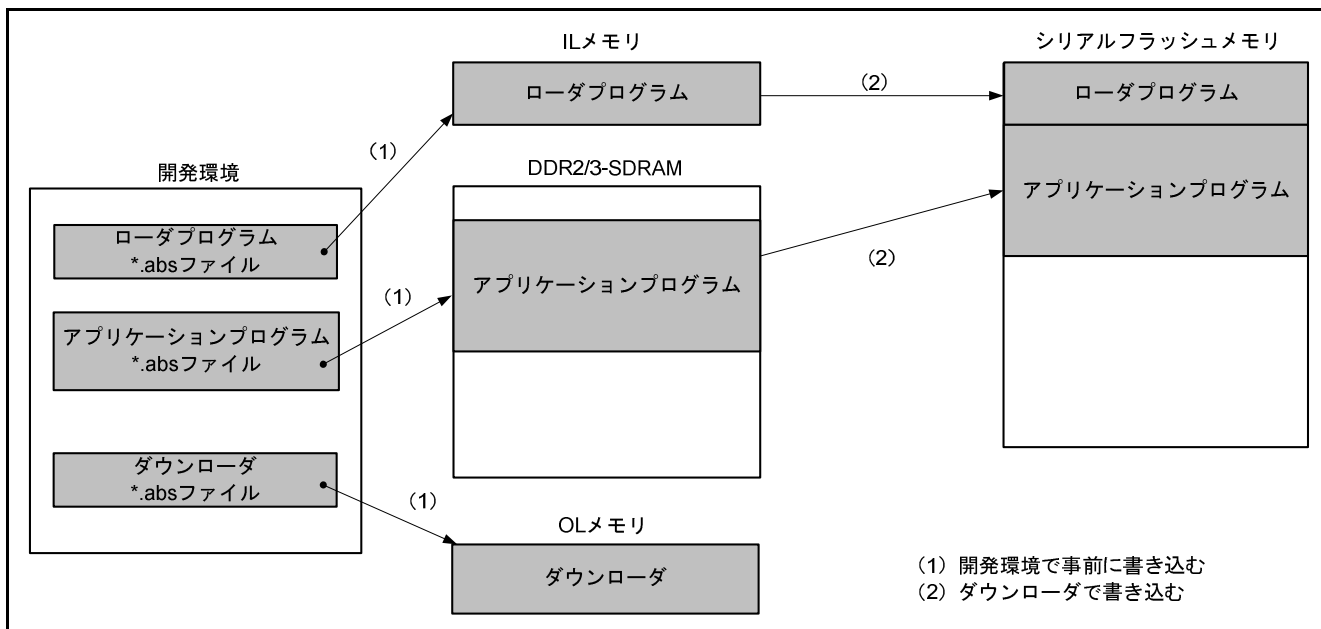


図1.3 ダウンローダの動作概要図

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1 動作確認条件

項目	内容
使用マイコン	SH7734 (R8A77343)
動作周波数	EXTAL 入力周波数: 33.3333MHz CPU クロック (clki) : 400MHz SHwy クロック (clks) : 200MHz SHwy クロック (clks1) : 100MHz DDR クロック (MCK0/MCK0#/MCK1/MCK1#) : 200MHz バスクロック (clk_b) : 50MHz 周辺クロック (clk_p) : 50MHz
動作電圧	IO supply power (3.3V) Core supply power (1.25V)
統合開発環境	ルネサス エレクトロニクス製 High-performance Embedded Workshop (Version 4.09.00.007)
C コンパイラ	ルネサス エレクトロニクス製 C/C++ Compiler Package for SuperH Family (9.4.0.0) コンパイルオプション -cpu=sh4a -endian=little -include="\$(PROJDIR)¥inc" -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
動作モード	シリアルブート (MD19=0,MD14=0,MD18=0,MD17=1,MD16=1)
サンプルコードのバージョン	Ver 1.00
使用ボード	ルネサス エレクトロニクス社製 SH7734 評価用プラットフォーム (R0P7734C00000RZ)
アドレス拡張モード	29 ビット
メモリマネジメントユニット (MMU)	ディスエーブル
使用デバイス	128Mbit On-board QSPI Flash ROM N25Q128A13TSF40F (Numonyx)

【注】シリアルブート時のシリアルフラッシュメモリのコマンドについて

シリアルブート時は、シリアルフラッシュメモリの High-Speed Read コマンド (H'0B) を使用して、シリアルフラッシュメモリからプログラムデータを読み出します。
シリアルブート時は、H'0B コマンドをサポートしているシリアルフラッシュメモリをご使用ください。

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)

4. 周辺機能説明

ブートモードについて補足します。その他、本アプリケーションノートで使用する周辺機能（RSPI、Quad-SPI）の説明は SH7734 ユーザーズマニュアル ハードウェア編（R01UH0233JJ）を参照ください。

4.1 ブートモード

PRESET#=L の状態の時に外部端子を用いてブートモードを決定することができます。ブートモードを決定する外部端子設定は、SH7734 ユーザーズマニュアル ハードウェア編（R01UH0233JJ）を参照ください。

モード設定がシリアルブートのとき、SH7734 は PRESET#=H 後、シリアルフラッシュメモリから先頭 16K バイトを IL メモリにコピーし、IL メモリの先頭番地よりプログラムの実行を開始します。

5. ハードウェア説明

5.1 ハードウェア構成例

図 5.1 に本アプリケーションノートで使用している評価ボードに搭載されている QSPI Flash ROM 使用時の接続例を示します。RSPI と Quad-SPI は、ピンマルチプレクスされており、Quad-SPI に設定を切り換えると Quad-SPI 動作が可能になります。図 5.1 の **太字 斜体** は、RSPI 選択時となります。その他周辺回路の結線等につきましては、SH7734 評価用プラットフォーム (R0P7734C00000RZ) の技術ドキュメントをご参照ください。

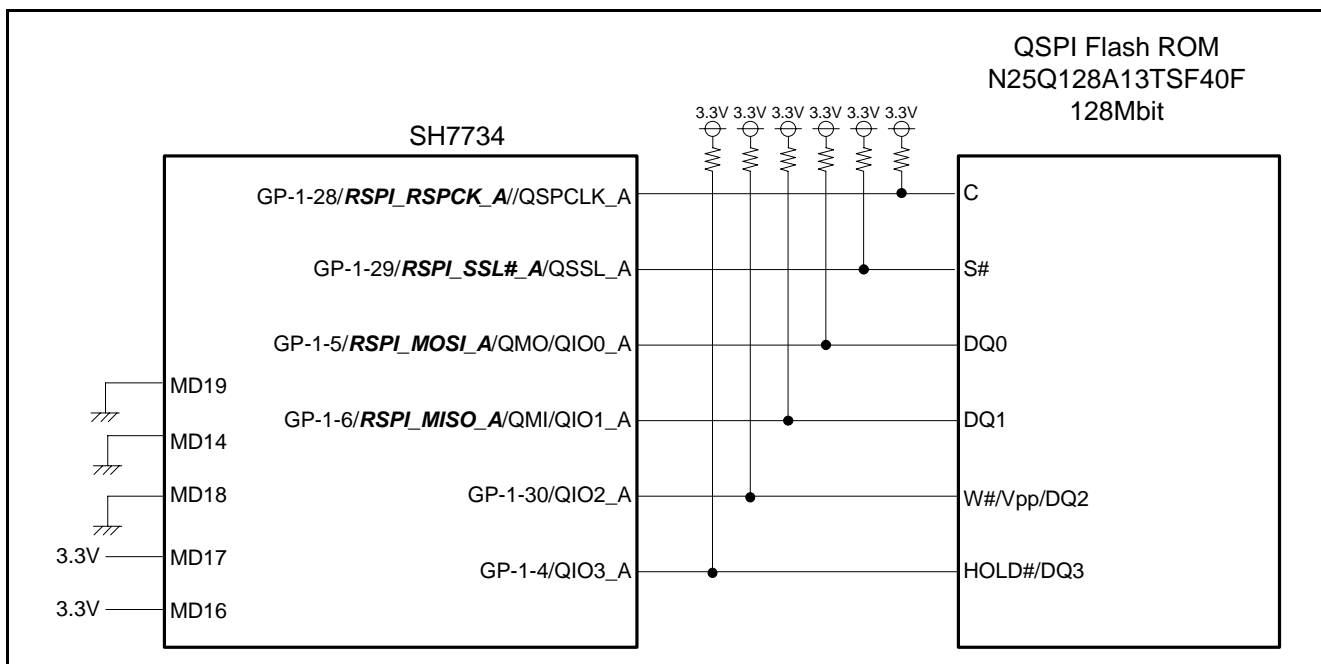


図5.1 接続例

【参考】

本アプリケーションノートで使用している SH7734 評価用プラットフォーム (R0P7734C00000RZ) に搭載している QSPI Flash ROM (N25Q128A13TSF40F) は、Single/Dual/Quad-SPI 動作に対応しています。データ用途として、DQ0 および DQ1 は、RSPI と Quad-SPI で使用されますが、DQ2 および DQ3 は、Quad-SPI でのみ使用されます。

5.2 使用端子一覧

表 5.1に RSPI 使用時の使用端子と機能、表 5.2に Quad-SPI 使用時の使用端子と機能を示します。

表5.1 使用端子と機能 (RSPI 使用時)

端子名	入出力	内容
RSPI_SSL#_A	出力	スレーブセレクト (注 1)
RSPI_RSPCK_A	出力	クロック出力 (注 1)
RSPI_MOSI_A	出力	マスタ送出データ (注 1)
RSPI_MISO_A	入力	マスタ入力データ (注 1)

注1 本アプリケーションノートでは、マスタ動作のみを対象とします。

表5.2 使用端子と機能 (Quad-SPI 使用時)

端子名	入出力	内容
QSSL#_A	出力	スレーブセレクト (注 2)
QSPCLK_A	出力	クロック出力 (注 2)
QMO/QIO0_A (注 3)	入出力	マスタ送出データ/データ 0 (注 2) (注 4)
QMI/QIO1_A (注 3)	入力	マスタ入力データ/データ 1 (注 2) (注 4)
QIO2_A	入力	データ 2 (注 2) (注 4)
QIO3_A	入力	データ 3 (注 2) (注 4)

注2 Quad-SPI は、マスタ動作のみ対応しています。

注3 Single-SPI モード時は QMO/QMI、Dual/Quad-SPI モード時は QIO0/QIO1 となります。

注4 本アプリケーションノートでは、CMD、アドレス、ダミー出力を QMO で行い、データ入力を QIO0_A、QIO1_A、QIO2_A、QIO3_A で行う、リード動作のみを対象とします。

6. ソフトウェア説明

6.1 サンプルコードの構成

本アプリケーションノートでは、ローダプログラム、アプリケーションプログラム、ダウンローダで異なるワークスペースを用意しています。表 6.1のように3つのワークスペースに分かれています。

各ワークスペースのソフトウェア説明は、表 6.1の説明章の章に記載します。

表6.1 サンプルコードの構成

ワークスペース名	説明	説明章
sh7734_sflash_loader_prog (ローダプログラム (RSPI))	このワークスペースのプロジェクトは、ローダプログラムをビルドします。	7章
sh7734_sflash_app (アプリケーションプログラム)	このワークスペースのプロジェクトは、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、[sh7734_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、[sh7734_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。	8章
sh7734_sflash_downloader (ダウンローダ)	このワークスペースのプロジェクトは、ダウンローダをビルドします。	9章

6.2 各ワークスペースのソフトウェア説明の前に

6.2.1 記載について

(1) 記載の省略

7章、8章、9章の以下項目に関して、統合開発環境で自動生成されるもの、および「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」で登場するものについては、記載を省略します。

- ファイル構成
- 定数一覧
- 構造体/共用体一覧
- 変数一覧
- 関数一覧
- 関数仕様

(2) DDR2/3-SDRAM の記載について

本アプリケーションノートで使用している評価ボードは、DDR2-SDRAM を搭載していますので、以降の章では、DDR2-SDRAM と記載します。

6.2.2 用語について

(1) アプリケーションプログラム関連の用語

アプリケーションプログラムの説明する前に、本アプリケーションノートでの、アプリケーションプログラム関連の用語を表 6.2のように定義します。

表6.2 アプリケーションプログラム関連の用語

用語	説明
アプリケーションプログラム転送情報 (appinfo)	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本体を DDR2-SDRAM に転送します。表 7.6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本体のエントリ関数アドレス情報	ローダプログラムが、アプリケーションプログラムを転送後、アプリケーションプログラム本体のエントリ関数のアドレスに分岐する際に使用します。シリアルフラッシュメモリに格納されています。
アプリケーションプログラム本体	アプリケーションプログラム本体を示します。プログラム領域のみでなく、プログラムを動作させるために必要なデータ領域のセクション (C,D 等) も含みます。
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転送情報 (appinfo) + アプリケーションプログラム本体のエントリ関数アドレス情報 + アプリケーションプログラム本体とします。

7. ローダプログラム (RSPI) のソフトウェア説明

7.1 動作概要

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 DDR2-SDRAM に転送し、アプリケーションプログラム本体のエントリ関数に分岐する処理を行います。

7.1.1 メモリマップ

図 7.1にローダプログラムのメモリマップを示します。以下、登場する各セクションについては、表 7.1を参照ください。

1. ローダプログラムは、H'E520 0000 ~ H'E520 3FFF 番地を使用します。
2. ローダプログラムの先頭プログラムを配置している RSTHandler セクションは、H'E520 0000 番地に配置してください。
3. その他、ローダプログラムに必要なセクション (P,C,B 等) や仮の例外処理ハンドラに必要なセクション (PINTHandler、PIntPRG、VECTTBL、INTTBL 等) を、RSTHandler セクション以降の番地に配置します。
4. 本アプリケーションノートでは、ローダプログラムのスタック領域は、H'E520 3C00 ~ H'E520 3FFF 番地を使用します。

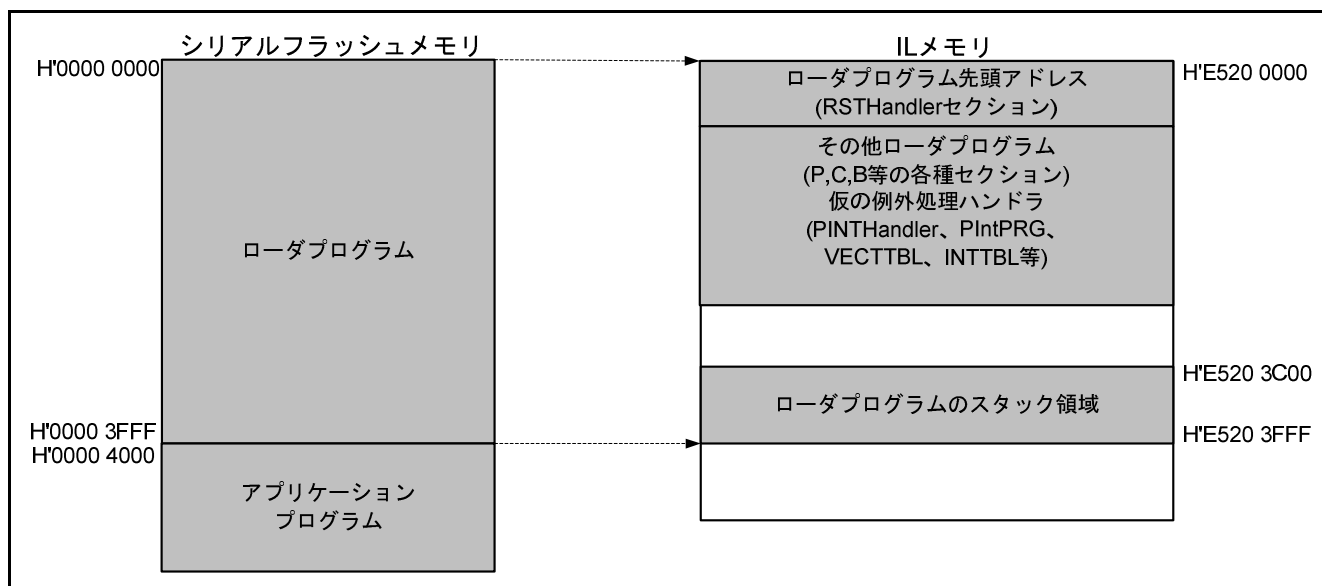


図7.1 ローダプログラム (RSPI) のメモリマップ例

7.1.2 セクション配置

表 7.1に、本アプリケーションノートでのローダプログラムのセクション配置例を示します。以下、登場する各セクションの役割の詳細については、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」を参照ください。

表7.1 ローダプログラムのセクション配置例

配置アドレス	セクション名	セクション用途
H'E520 0000 (ILメモリ)	RSTHandler	リセットハンドラ (ローダプログラムの先頭プログラム)
	PResetPRG	リセットプログラム領域
	P_DBSC3	DBSC3 初期化プログラム領域
	PnonCACHE	キャッシュ操作用プログラム領域
	P\$PSEC	セクション初期化用プログラム領域
	PINTHandler	例外/割り込みハンドラ
	PIntPRG	割り込み関数
	P	プログラム領域
	C	定数領域
	C\$BSEC	未初期化データ領域用アドレス構造体
	C\$DSEC	初期化データ領域用アドレス構造体
	D	初期化データ (初期値)
	VECTTBL	リセットベクタテーブル
	INTTBL	割り込みベクタテーブル
	B	未初期化データ領域
	R	初期化データ領域
H'E520 3C00 (ILメモリ)	S	スタック領域

7.1.3 SPI モード、ビットレート

表 7.2に本アプリケーションノートのローダプログラム (RSPI) の SPI モード、ビットレート仕様を示します。

表7.2 ローダプログラム (RSPI) の SPI モード、ビットレート仕様

項目	内容
SPI モード	SPI モード 3 (CPOL=1, CPHA=1)
ビットレート	12.5MHz

7.2 ファイル構成

表 7.3にローダプログラム (RSPI) を示します。

表7.3 ローダプログラム (RSPI) ファイル構成

ファイル名	概要	備考
sh7734_main_lp.c	ローダプログラム (RSPI) のメイン処理	
dbstc_lp.c	SH7734 初期設定例 (R01AN0665JJ) の dbstc.c から、ローダプログラム (RSPI) で使用するセクション定義に変更	
r_rspl_lp.c	RSPI シリアルフラッシュメモリ制御モジュール群	
r_wdt_lp.c	ウォッチドッグタイマ (WDT) 制御モジュール群	本アプリケーションノートでは、ブート時に予期せぬ状態の回避のため念のため WDT を起動させています。
r_rspl_lp.h	RSPI シリアルフラッシュメモリ制御モジュール群の外部参照用インクルードヘッダ	
r_wdt_lp.h	ウォッチドッグタイマ (WDT) 制御モジュール群の外部参照用インクルードヘッダ	
vhandler_lp.src	SH7734 初期設定例 (R01AN0665JJ) の vhandler.src から、以下を変更。 リセット (RSTHandler)、TLB ミス例外 (VBR+H'400) のハンドラ処理を変更。 アプリケーションプログラムへの分岐プログラム追加。	本アプリケーションノートでは、一般例外 (VBR+H'100)、TLB ミス例外 (VBR+H'400) 処理を想定していません。システムに合わせて変更ください。

【参考】

本アプリケーションノートでは、アプリケーションプログラムやダウンローダプログラムのファイルと区別するため、ローダプログラムのファイル名を "XXX_lp.c" としています。

7.3 定数一覧

表 7.4にローダプログラム (RSPI) サンプルコードで使用する定数を示します。

表7.4 ローダプログラム (RSPI) サンプルコードで使用する定数

定数名	設定値	内容
SF_PAGE_SIZE	256	r_rsipi_lp.h に定義 シリアルフラッシュメモリの 1 ページサイズ (256 B) * ¹
SF_SECTOR_SIZE	H'10000	r_rsipi_lp.h に定義 シリアルフラッシュメモリの 1 セクタサイズ (64 KB) * ¹
SFLASHCMD_CHIP_ERASE	H'C7	r_rsipi_lp.c に定義 Bulk Erase コマンド用 * ¹
SFLASHCMD_SECTOR_ERASE	H'D8	r_rsipi_lp.c に定義 Sector Erase コマンド用 * ¹
SFLASHCMD_BYTE_PROGRAM	H'02	r_rsipi_lp.c に定義 Page Program コマンド用 * ¹
SFLASHCMD_BYTE_READ	H'0B	r_rsipi_lp.c に定義 Read Data Bytes at Higher Speed コマンド用 * ¹
SFLASHCMD_BYTE_READ_LOW	H'03	r_rsipi_lp.c に定義 Read Data Bytes コマンド用 * ¹
SFLASHCMD_WRITE_ENABLE	H'06	r_rsipi_lp.c に定義 Write Enable コマンド用 * ¹
SFLASHCMD_WRITE_DISABLE	H'04	r_rsipi_lp.c に定義 Write Disable コマンド用 * ¹
SFLASHCMD_READ_STATUS	H'05	r_rsipi_lp.c に定義 Read Status Register コマンド用 * ¹
SFLASHCMD_WRITE_STATUS	H'01	r_rsipi_lp.c に定義 Write Status Register コマンド用 * ¹
D_SFLASH_MOD_SEL_SET	H'01000000	r_rsipi_lp.c に定義 MOD_SEL レジスタ設定用 GroupA RSPI 選択用
D_SFLASH_IPSR2_SET_INIT	H'007E07E0	r_rsipi_lp.c に定義 IPSR2 レジスタ設定用 RSPI_RSPCK_A,RSPI_SSL#_A,RSPI_MOSI_A,RSPI_MISO_A 端子初期化用
D_SFLASH_IPSR2_SET_SF_LASH	H'00120120	r_rsipi_lp.c に定義 IPSR2 レジスタ設定用 RSPI_RSPCK_A,RSPI_SSL#_A,RSPI_MOSI_A,RSPI_MISO_A 端子選択用
D_SFLASH_GPSR1_SET_SF_LASH	H'30000060	r_rsipi_lp.c に定義 GPSR1 レジスタ設定用 RSPI_RSPCK_A,RSPI_SSL#_A,RSPI_MOSI_A,RSPI_MISO_A 端子と GPIO 端子切り替え選択用
D_SFLASH_MSTPCR1_SET	H'00000800	r_rsipi_lp.c に定義 MSTPCR1 レジスタ設定用 HPB-DMAC クロック供給用

定数名	設定値	内容
D_SFLASH_DMAC_DPTR27_INIT	H'00000200	r_rsipi_lp.c に定義 DPTR27 レジスタ設定用 RSPI と HPB-DMAC 連携設定用
D_SFLASH_DMAC_DCR27_INIT_LITTLE	H'06A02100	r_rsipi_lp.c に定義 DCR27 レジスタ設定用 リトルエンディアン対象
D_SFLASH_DMAC_DCMDR27_START	H'00000001	r_rsipi_lp.c に定義 DCMDR27 レジスタ設定用 HPB-DMAC 起動開始用
D_SFLASH_DMAC_HSRSTR_SET_RESET	H'00000001	r_rsipi_lp.c に定義 HSRSTR レジスタ設定用 HPB-DMAC モジュールリセット用
SF_USE_DMAC		r_rsipi_lp.c に定義 #ifdef で CPU 転送、HPB-DMAC 転送の切り分けに使用。HPB-DMAC 転送指定時に定義 本アプリケーションノートでは、RSPI と HPB-DMAC 連携動作を行いますので、定義は有効になっています。
D_WDT_WDTBST_INIT	H'55000000	r_wdt_lp.c に定義 WDTBST レジスタ初期値
D_WDT_WDTBST	H'5500C350	r_wdt_lp.c に定義 WDTBST レジスタ設定値 1ms 設定 (clkp=50MHz の場合)
D_WDT_WDTST_INIT	H'5A000000	r_wdt_lp.c に定義 WDTST レジスタ初期値
D_WDT_WDTST	H'5A0003E8	r_wdt_lp.c に定義 WDTST レジスタ設定値 ウォッチドッグタイマ設定値 = WDTBST 設定値 (1ms) × 1000 = 1s
D_WDT_WDTCSR_START	H'A50000C0	r_wdt_lp.c に定義 WDTCSR レジスタ設定値 ウォッチドッグタイマ開始
D_WDT_WDTCSR_STOP	H'A5000000	r_wdt_lp.c に定義 WDTCSR レジスタ設定値 ウォッチドッグタイマ停止
APROG_TOP_SFLASH	H'00004000	sh7734_main_lp.c に定義 アプリケーションプログラムを格納しているシリアルフラッシュメモリの先頭アドレス
APPINFO_TOP	APROG_TOP_SFLASH	sh7734_main_lp.c に定義 アプリケーションプログラム先頭アドレスを格納しているシリアルフラッシュメモリのアドレス
APPINFO_END	APROG_TOP_SFLASH + 4	sh7734_main_lp.c に定義 アプリケーションプログラム最終アドレスを格納しているシリアルフラッシュメモリのアドレス

*1 詳細は、使用するシリアルフラッシュメモリのデータシートを参照ください。

7.4 関数一覧

表 7.5にローダプログラム (RSPI) サンプルコードで使用する関数を示します。

表7.5 ローダプログラム (RSPI) サンプルコードで使用する関数

関数名	概要
main	ローダプログラム (RSPI) メイン処理
get_appinfo	アプリケーションプログラム転送情報 (appinfo) 取得処理
app_prog_transfer	アプリケーションプログラム転送処理
jmp_app_prog	アプリケーションプログラムエントリ関数への分岐処理
R_WDT_Start	ウォッチドッグタイマ起動
R_WDT_RegisterInit1	ウォッチドッグタイマ初期化手順 1
R_WDT_RegisterInit2	ウォッチドッグタイマ初期化手順 2
R_RSPI_SFInitSerialFlash	RSPI 初期化
R_RSPI_SFByteRead	RSPI シリアルフラッシュメモリリード処理 (1 バイト単位)
R_RSPI_SFByteReadLong	RSPI シリアルフラッシュメモリリード処理 (4 バイト単位)

7.5 関数仕様

サンプルコードの関数仕様を示します。

main	
概要	ローダプログラム (RSPI) メイン処理
ヘッダ	r_rspi_lp.h
宣言	void main(void)
説明	<ul style="list-style-type: none"> 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 DDR2-SDRAM に転送し、アプリケーションプログラム本体のエントリ関数に分岐します。 本アプリケーションノートでは、デバッグ用途として、アプリケーションプログラム転送中は LED4 を点灯します。 本アプリケーションノートでは、アプリケーションプログラム転送中に予期せぬエラー状態に陥った場合を考慮し、念のためウォッチドッグタイマを有効にします。
引数	なし
リターン値	なし
get_appinfo	
概要	アプリケーションプログラム転送情報 (appinfo) 取得処理
ヘッダ	r_rspi_lp.h
宣言	static void get_appinfo(uint32_t *app_top_addr, uint32_t *app_end_addr)
説明	<ul style="list-style-type: none"> シリアルフラッシュメモリよりアプリケーションプログラム転送情報 (appinfo) の転送先先頭アドレス、転送先最終アドレスを取得します。アプリケーションプログラム転送情報 (appinfo) の詳細は、表 7.6 を参照ください。
引数	uint32_t *app_top_addr : 転送先先頭アドレス格納先ポインタ uint32_t *app_end_addr : 転送先最終アドレス格納先ポインタ
リターン値	なし
備考	<ul style="list-style-type: none"> 本アプリケーションノートでは、アプリケーションプログラムは、DDR2-SDRAM 上に転送されるため、上記、取得される転送先先頭アドレス、転送先最終アドレスは、DDR2-SDRAM のアドレスとなります。
app_prog_transfer	
概要	アプリケーションプログラム転送処理
ヘッダ	r_rspi_lp.h
宣言	static void app_prog_transfer(const uint32_t app_top_addr, const uint32_t app_end_addr)
説明	<ul style="list-style-type: none"> アプリケーションプログラムを転送します。
引数	uint32_t app_top_addr : 転送先先頭アドレス uint32_t app_end_addr : 転送先最終アドレス
リターン値	なし
備考	<ul style="list-style-type: none"> 本アプリケーションノートでは、アプリケーションプログラムは、DDR2-SDRAM 上に転送されるため、以下引数の転送先先頭アドレス、転送先最終アドレスは、DDR2-SDRAM のアドレスとなります。

jmp_app_prog	
概要	アプリケーションプログラムへの分岐処理
ヘッダ	r_rspi_lp.h
宣言	void jmp_app_prog (uint32_t app_top_addr)
説明	<ul style="list-style-type: none"> アプリケーションプログラム関連情報の転送先先頭アドレスよりアプリケーションプログラム本体のエントリ関数アドレスを算出し、そのエントリ関数へ分岐します。
引数	uint32_t app_top_addr : 転送先先頭アドレス
リターン値	なし
R_WDT_Start	
概要	ウォッチドッグタイマ起動
ヘッダ	r_wdt_lp.h
宣言	void R_WDT_Start(void)
説明	<ul style="list-style-type: none"> 1s 周期のウォッチドッグタイマを起動します。 ウォッチドッグタイマカウンタは、10ms 周期の TMU タイマにより再設定されます。
引数	なし
リターン値	なし
R_WDT_RegisterInit1	
概要	ウォッチドッグタイマ初期化手順 1
ヘッダ	r_wdt_lp.h
宣言	void R_WDT_RegisterInit1(void)
説明	ウォッチドッグタイマの停止を行います。
引数	なし
リターン値	なし
R_WDT_RegisterInit2	
概要	ウォッチドッグタイマ初期化手順 2
ヘッダ	r_wdt_lp.h
宣言	void R_WDT_RegisterInit2(void)
説明	ウォッチドッグタイマカウンタの初期化を行います。
引数	なし
リターン値	なし
R_RSPI_SFInitSerialFlash	
概要	RSPI 初期化
ヘッダ	r_rspi_lp.h
宣言	void R_RSPI_SFInitSerialFlash(void)
説明	<ul style="list-style-type: none"> RSPI へのクロック供給、使用する端子設定、RSPI の初期化を行います。
引数	なし
リターン値	なし

R_RSPI_SFByteRead

概要	RSPI シリアルフラッシュメモリリード処理 (1 バイト単位)		
ヘッダ	r_rsipi_lp.h		
宣言	void R_RSPI_SFByteRead(const uint32_t addr, uint8_t *buf, const uint32_t size)		
説明	<ul style="list-style-type: none">パラメータ指定されたシリアルフラッシュメモリアドレスから指定サイズ分のデータを 1 バイト単位でリードします。SPCMD レジスタの転送データ長 (SPB[0:3]ビット) に 8 ビット長指定します。		
引数	uint32_t addr	:	リード対象のシリアルフラッシュメモリアドレス
	uint8_t *buf	:	リードデータ格納用ポインタ
	uint32_t size	:	リードサイズ (1 バイト単位)
リターン値	なし		

R_RSPI_SFByteReadLong

概要	RSPI シリアルフラッシュメモリリード処理 (4 バイト単位)		
ヘッダ	r_rsipi_lp.h		
宣言	void R_RSPI_SFByteReadLong(const uint32_t addr, uint32_t *buf, const uint32_t size)		
説明	<ul style="list-style-type: none">パラメータ指定されたシリアルフラッシュメモリアドレスから指定サイズ分のデータを 4 バイト単位でリードします。SPCMD レジスタの転送データ長 (SPB[0:3]ビット) に 32 ビット長指定します。パラメータ size は、1 バイト単位のサイズを指定しますが、4 の倍数で指定ください。例) 16 バイトをリードする場合は、16 を指定ください。		
引数	uint32_t addr	:	リード対象のシリアルフラッシュメモリアドレス
	uint32_t *buf	:	リードデータ格納用ポインタ
	uint32_t size	:	リードサイズ (1 バイト単位)
リターン値	なし		

7.6 フローチャート

図 7.2にローダプログラム (RSPI) 処理のフローチャートを示します。

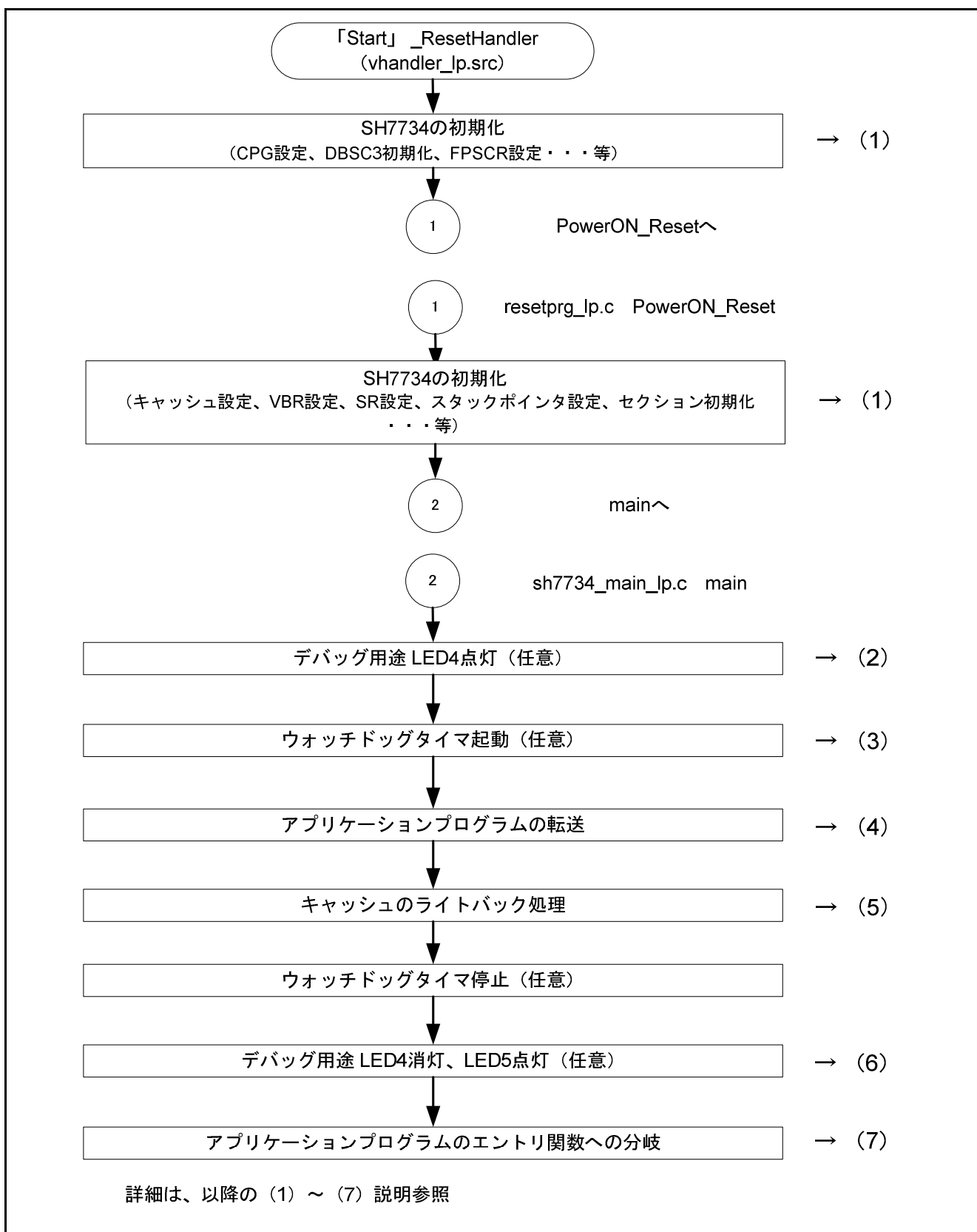


図7.2 ローダプログラム (RSPI) 処理 フローチャート

(1) SH7734 の初期化処理

SH7734 を動作させるための最低限の処理 (CPG 設定、DBSC3 初期化、キャッシュ設定、FPSCR 設定、VBR 設定、SR 設定、スタックポインタ設定、セクション初期化・・・等) を行います。

本アプリケーションノートでは、SH7734 の初期化処理について、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」を流用していますので、そちらを参照ください。

(2) LED4 点灯 (任意)

本アプリケーションノートでは、デバッグ用途として、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ロードしている間、LED4 を点灯します。

(3) ウォッチドッグタイマ起動 (任意)

本アプリケーションノートでは、アプリケーションプログラム転送中に予期せぬエラー状態に陥った場合を考慮し、念のためウォッチドッグタイマ (1s 周期) を有効にします。合わせて、ウォッチドッグタイマカウンタを再設定する 10ms 周期の TMU タイマを起動します。TMU はタイムアウト時、割り込みを発生させます。

(4) 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 DDR2-SDRAM に転送します。アプリケーションプログラム転送情報 (appinfo) は、表 7.6 のようにシリアルフラッシュメモリの H'0000 4000 ~ H'0000 4007 番地に配置します。アプリケーションプログラム本体のエントリ関数アドレス情報は、表 7.7 のようにシリアルフラッシュメモリの H'0000 4008 ~ H'0000 400B 番地から取得します。この領域にエントリ関数のアドレス情報を配置してください。

表7.6 アプリケーションプログラム転送情報 (appinfo)

項目	セクション	シリアルフラッシュメモリアドレス	内容	サイズ
転送先先頭アドレス	DAPPINFO	H'0000 4000	H'0C000000	4
転送先最終アドレス		H'0000 4004	H'0C000000 + アプリケーションプログラムのサイズ	4

表7.7 アプリケーションプログラム本体のエントリ関数アドレス情報

項目	セクション	シリアルフラッシュメモリアドレス	内容	サイズ
アプリケーションプログラム本体のエントリ関数アドレス情報	VECTTBL	H'0000 4008	アプリケーションプログラムのエントリ関数 (PowerON_Reset()) アドレス	4

図 7.3 にアプリケーションプログラム転送情報 (appinfo) 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 8.4 章を参照ください。

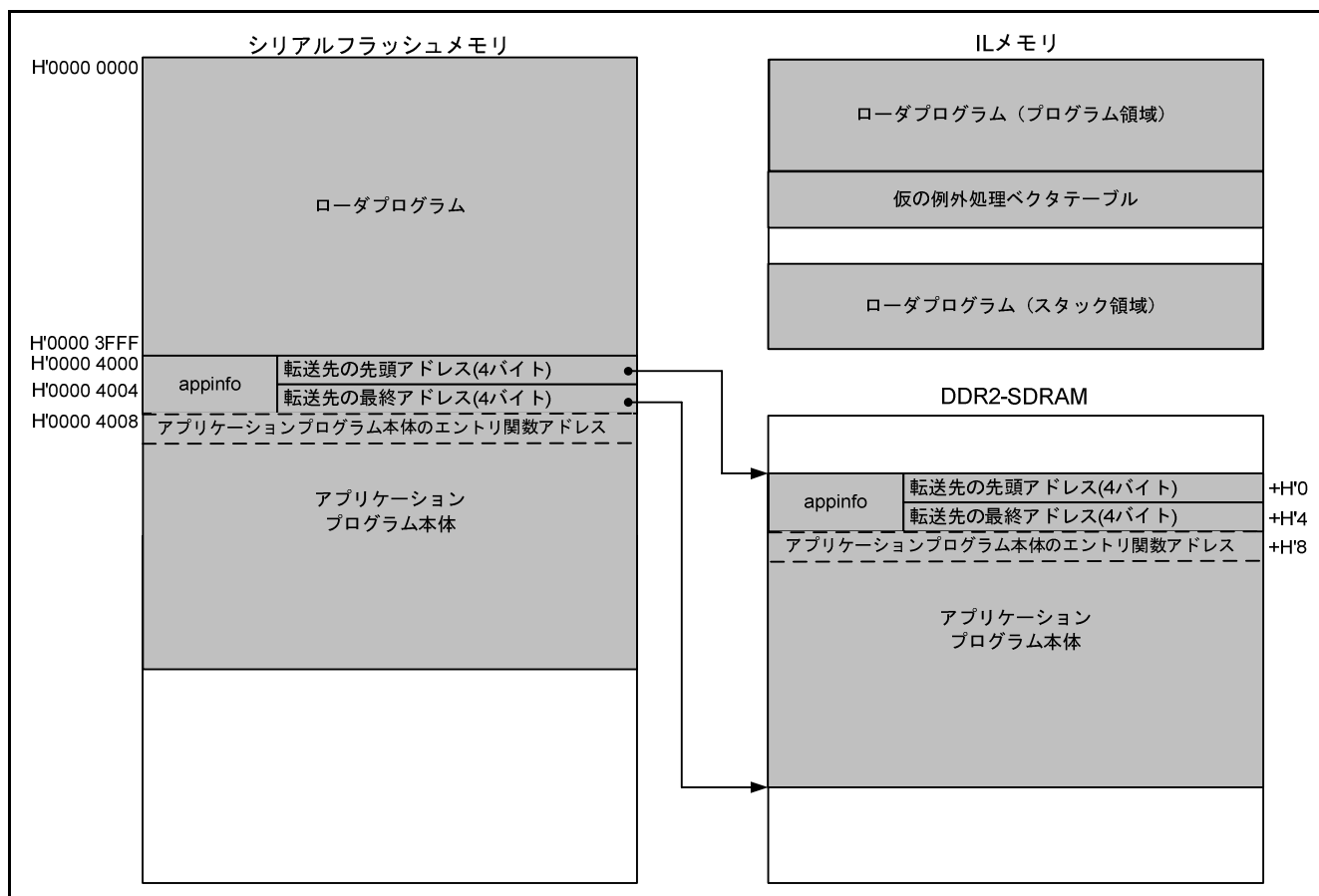


図7.3 アプリケーションプログラム転送イメージ図

(5) 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 DDR2-SDRAM に CPU で転送した場合、キャッシュメモリと DDR2-SDRAM 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。HPB-DMAC で転送した場合については、本アプリケーションノートでは、特に何も行いません。

(6) LED4 消灯、LED5 点灯 (任意)

本アプリケーションノートでは、デバッグ用途として、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正常ロード完了時に、LED4 を消灯、LED5 点灯します。

(7) アプリケーションプログラム本体のエントリ関数への分岐

ローダプログラムは、表 7.7 アプリケーションプログラム本体のエントリ関数アドレス情報に格納されているアプリケーションプログラムのエントリ関数 (PowerON_Reset()) に分岐します。アプリケーションプログラム本体のエントリ関数アドレス情報の設定方法は、8.1.1 セクション配置を参照ください。

8. アプリケーションプログラムのソフトウェア説明

8.1 動作概要

アプリケーションプログラムは、ローダプログラムが読み込むメモリに配置する必要があります。また、アプリケーションプログラムには、アプリケーションプログラム転送情報（appinfo）とアプリケーションプログラム本体のエントリ関数アドレス情報を組み込む必要があります。

8.1.1 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。以下、登場する各セクションの役割の詳細については、「SH7734 グループ SH7734 初期設定例（R01AN0665JJ）」を参照ください。

表8.1に本アプリケーションノートでのセクション配置例を示します。

表8.1 アプリケーションプログラムのセクション配置例

配置アドレス	セクション名	セクション用途
H'0C00 0000 (DDR2-SDRAM) *1	DAPPINFO	アプリケーションプログラム転送情報（appinfo）*3
H'0C00 0008 (DDR2-SDRAM) *1	VECTTBL	アプリケーションプログラム本体のエントリ関数アドレス情報*4 リセットベクタテーブル*7
	INTTBL*5	割り込みベクタテーブル*7
	PResetPRG	リセットプログラム
	PINTHandler*5	例外/割り込みハンドラ*6
	PIntPRG*5	割り込み関数*6
	P\$PSEC	セクション初期化用プログラム領域
	P	プログラム領域
	C	定数領域
	C\$BSEC	未初期化データ領域用アドレス構造体
	C\$DSEC	初期化データ領域用アドレス構造体
	D*5	初期化データ（初期値）
	PnonCACHE*5	キャッシュ操作プログラム領域*8
	H'0F00 0000 (DDR2-SDRAM) *2	B
R*5		初期化データ領域
H'0FFF F9F0 (DDR2-SDRAM) *2	S	スタック領域
H'8FFF FDF0 (DDR2-SDRAM) *2	SP_S	TLB ミスハンドラ専用スタック領域
H'E501 0000 (OLメモリ)	INTTBL_OL*5	割り込みマスクテーブルコピー領域*7
H'E520 0000 (ILメモリ)	PINTHandler_IL*5	例外/割り込みハンドラコピー領域*6
	PIntPRG_IL*5	割り込み関数コピー領域*6
	PnonCACHE_IL*5	キャッシュ操作プログラムコピー領域*8

- *1
本アプリケーションノートでは、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 DDR2-SDRAM 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。先頭アドレスと最終アドレスの情報の指定方法は、8.4章を参照ください。
- *2
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) 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。
- *4
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本体のエントリ関数アドレス情報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。
VECTTBL セクションが、アプリケーションプログラム本体のエントリ関数 (PowerON_Reset()) を先頭に定義しているセクションとなりますので、そのセクションを定義しています。アプリケーションプログラムの先頭から DAPPINFO セクション、VECTTBL セクションの順に配置します。
- *5
本アプリケーションノートでは、High-performance Embedded Workshop の「ROM から RAM へマップするセクション」の設定を表 8.2 のように行います。アプリケーションプログラムより実行されるセクションの初期化ルーチン (InitSct()) で、表 8.2 の ROM* 上にあるセクションを RAM 上にコピーします。
- *6
IL メモリは、命令の格納に適しています。本アプリケーションノートでは、割り込み処理を高速起動するために、割り込み処理に関するプログラム関連セクションを IL メモリにコピーしてから実行します。
- *7
OL メモリは、データの格納に適しています。本アプリケーションノートでは、割り込み処理を高速起動するために、割り込み処理に関するデータ関連セクションを OL メモリにコピーしてから実行します。
- *8
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 PnonCACHE セクションは IL メモリ上の PnonCACHE_IL セクションにコピーしてから実行します。

表8.2 ROM から RAM へマップするセクション

ROM *	RAM
D	R
INTTBL	INTTBL_OL
PIntPRG	PIntPRG_IL
PINTHandler	PINTHandler_IL
PnonCACHE	PnonCACHE_IL

【注】

「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」で紹介している NOR フラッシュ (ROM) から DDR2-SDRAM/IL メモリ/OL メモリ (RAM) へマップするケースが、本アプリケーションノートでは、ローダプログラムが、シリアルフラッシュ上のデータを一度 DDR2-SDRAM に展開後、IL メモリへコピーしますので、ここでは、ROM * に相当するものが、DDR2-SDRAM 上のセクション、RAM に相当するものが、IL メモリ/OL メモリになります。

8.2 ファイル構成

表 8.3にアプリケーションプログラム ファイル構成を示します。

表8.3 アプリケーションプログラム ファイル構成

ファイル名	概要	備考
sh7734_main.c	アプリケーションプログラムのメイン処理	SH7734 初期設定例 (R01AN0665JJ) から、シリアルログのみ変更。
dbstc.c	SH7734 初期設定例 (R01AN0665JJ) から、アプリケーションプログラムで使用するセクション定義に変更	
vhandler.src	SH7734 初期設定例(R01AN0665JJ)から、リセットハンドラの処理を削除。	ローダプログラムにより、リセット時の初期化処理は実行されているため。

8.3 フローチャート

本アプリケーションノートでは、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」を流用して、PC上のターミナルソフトにデバッグ情報の出力処理を行っています。

図 8.1 にアプリケーションプログラムのフローチャートを示します。

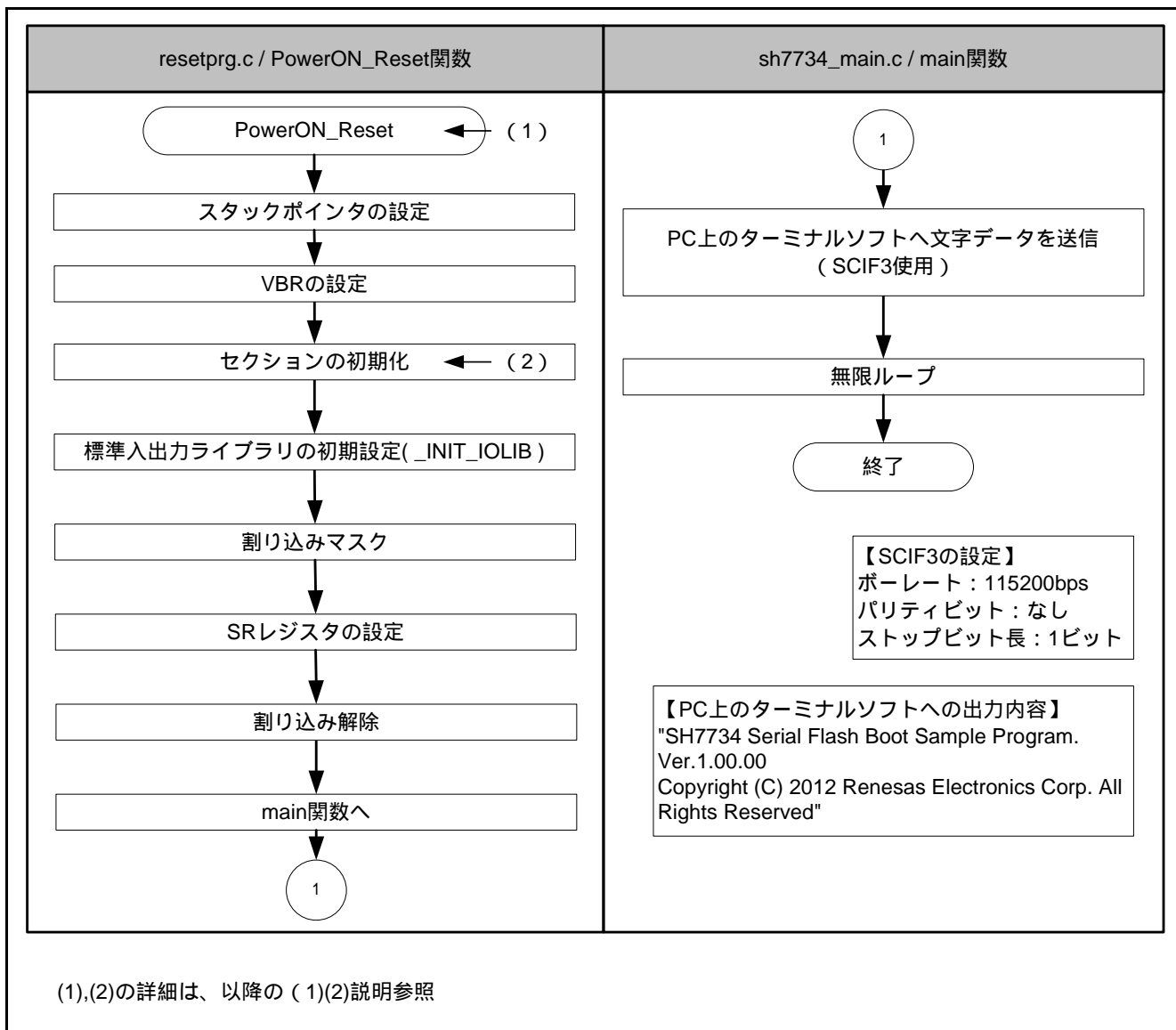


図8.1 アプリケーションプログラム フローチャート

(1) エントリ関数 (PowerON_Reset) の設定

アプリケーションプログラム本体のエントリ関数 (PowerON_Reset) アドレスは、リセットベクタテーブル `_RESET_Vectors` の先頭に設定します。表 8.4にエントリ関数アドレスの設定を示します。

表8.4 エントリ関数アドレスの設定

関数名	概要
ファイル名	vecttbl.src
配置セクション名	VECTTBL
テーブル名	_RESET_Vectors
設定関数名	PowerON_Reset (アプリケーションプログラム本体のエントリ関数)

(2) セクションの初期化

dbsect.c に定義されている、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。詳細は、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」を参照ください。

ローダプログラム実行時に、既にキャッシュが有効になっているため、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 DDR2-SDRAM のコヒーレンシが保証されない可能性があります。

本アプリケーションノートで実装されているサンプルプログラムでは特に問題となりませんが、main プログラム等で、キャッシュメモリと DDR2-SDRAM のコヒーレンシを保証しないと問題となるプログラムが実装される場合を考慮し、キャッシュのライトバック処理を行います。

8.4 アプリケーションプログラム転送情報 (appinfo) の生成

表 8.5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の最終アドレスを登録してください。

表8.5 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("PnonCACHE ")	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 (16K 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 8.2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。

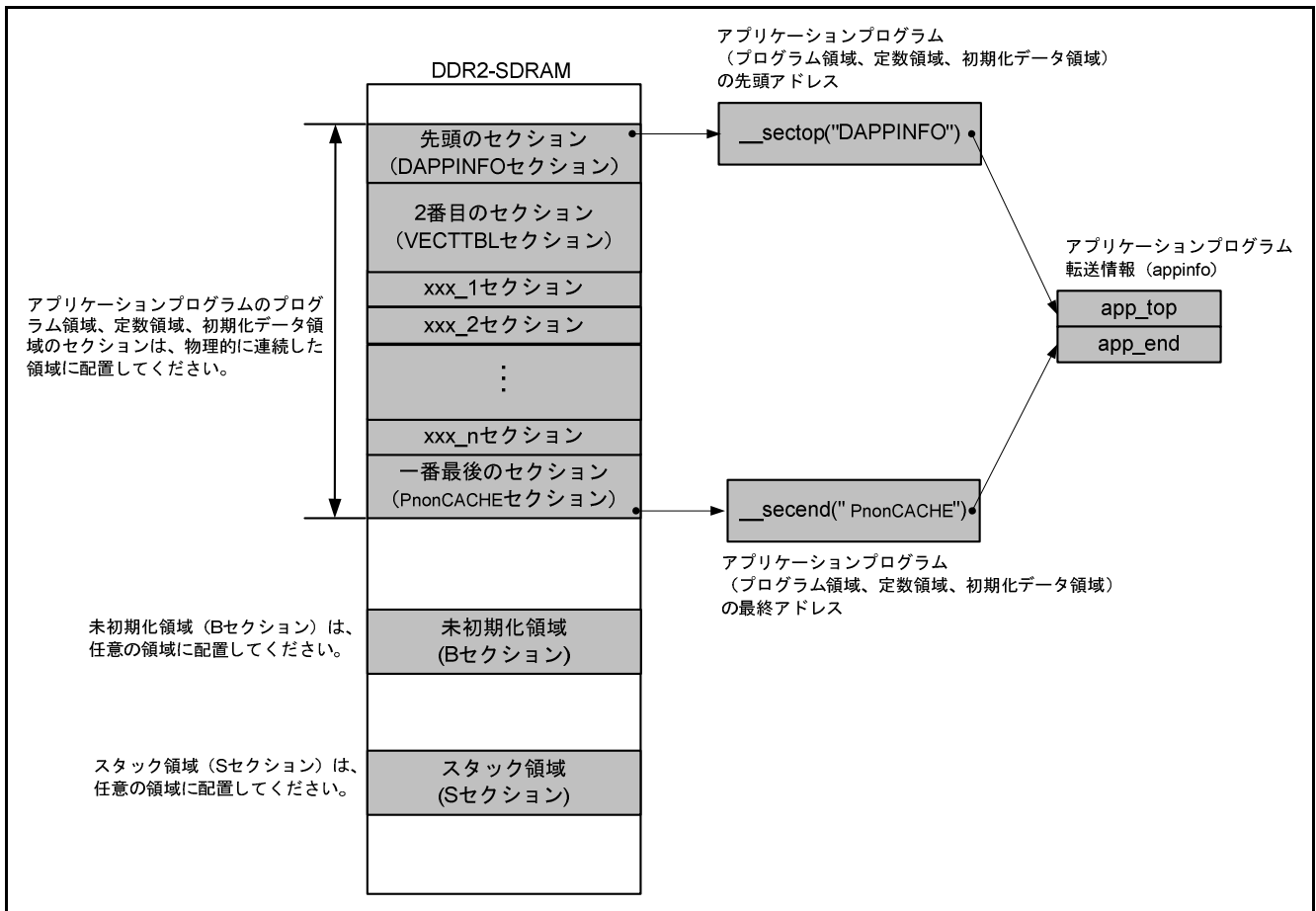


図8.2 アプリケーションプログラム転送情報 (appinfo) の生成イメージ図

9. ダウンローダプログラムのソフトウェア説明

9.1 ダウンローダプログラムの仕様詳細

9.1.1 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 OL メモリにダウンローダ、IL メモリにローダプログラム、DDR2-SDRAM にアプリケーションプログラムをそれぞれ転送します。図 9.1 に動作イメージ図を示します。

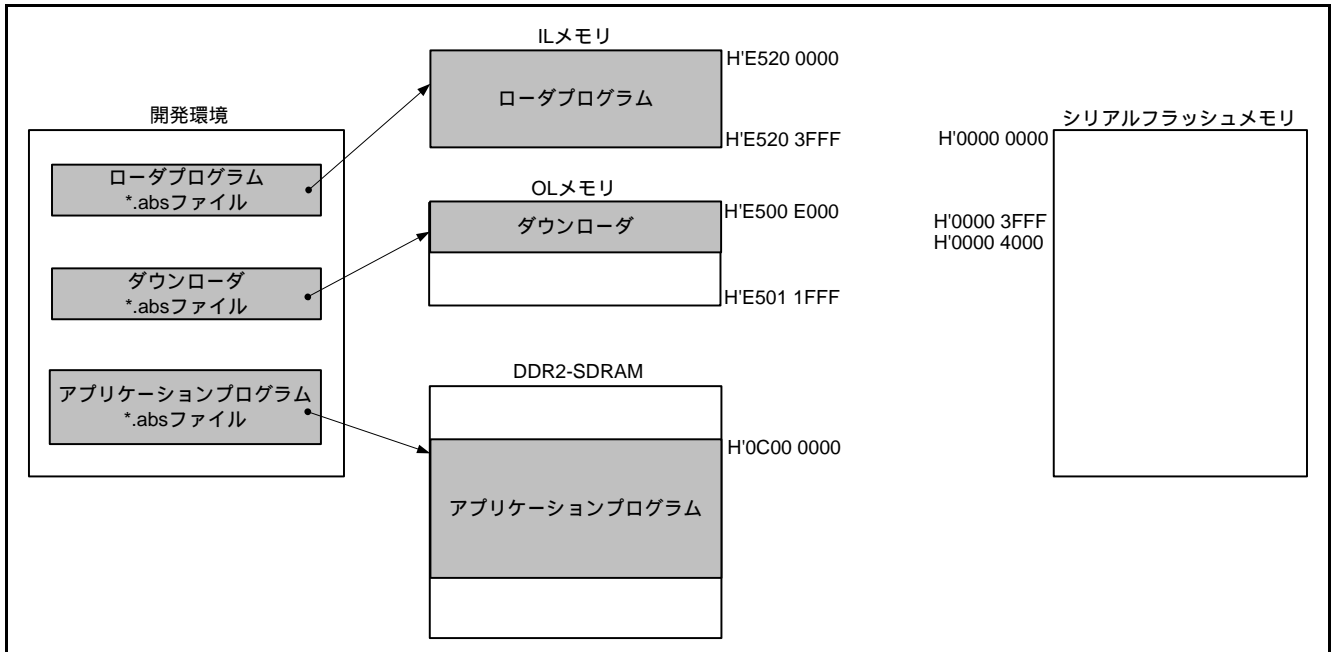


図9.1 ダウンローダ動作イメージ図(1)

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 3FFF 番地に、アプリケーションプログラムを H'0000 4000 番地以降に書き込みます。図 9.2 に動作イメージ図を示します。

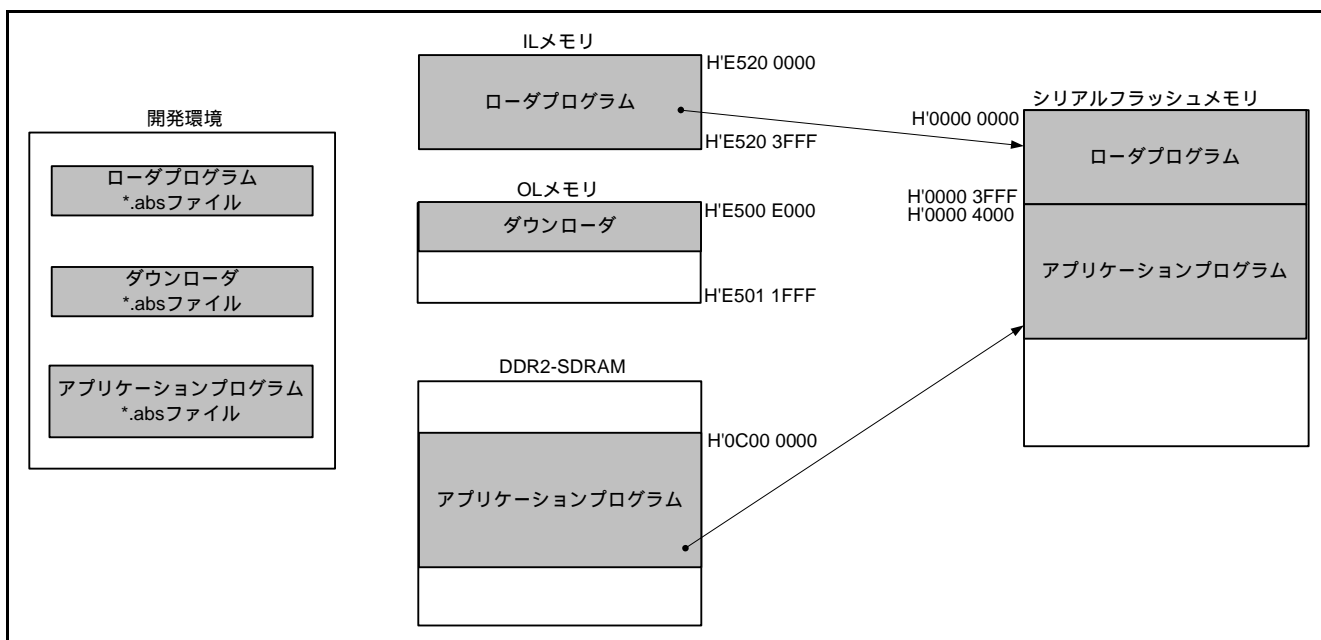


図9.2 ダウンローダ動作イメージ図(2)

【参考】本アプリケーションノートでは、ダウンローダをOLメモリ、ローダプログラムをILメモリ、アプリケーションプログラムをDDR2-SDRAMに配置することにより、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の競合を回避しています。全てのプログラムを、競合に注意してDDR2-SDRAM上に配置しても問題ありません。

9.1.2 セクション配置

表 9.1 に、本アプリケーションノートでのダウンローダのセクション配置例を示します。以下、登場する各セクションの役割の詳細については、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」を参照ください。

表9.1 ダウンローダのセクション配置例

配置アドレス	セクション名	セクション用途
H'E500 E000 (OLメモリ)	PResetPRG	リセットプログラム領域
	PnonCACHE	キャッシュ操作プログラム領域
	P_DBSC3	DBSC3 初期化プログラム領域
	P\$PSEC	セクション初期化用プログラム領域
	P	プログラム領域
	C\$BSEC	未初期化データ領域用アドレス構造体
	C\$DSEC	初期化データ領域用アドレス構造体
	C	定数領域
	D	初期化データ (初期値)
	B	未初期化データ領域
R	初期化データ領域	
H'E501 1C00 (OLメモリ)	S	スタック領域

9.2 ファイル構成

表 9.2にダウンローダを示します。

表9.2 ダウンローダ ファイル構成

ファイル名	概要	備考
sh7734_main_dl.c	ダウンローダのメイン処理	
dbstc_dl.c	SH7734 初期設定例(R01AN0665JJ)の dbstc.c から、ダウンローダで使用するセクション定義に変更	
r_rspi_dl.c	RSPI シリアルフラッシュメモリ制御モジュール群	表 7.3ローダプログラム (RSPI) と同じ内容です。
resetprg_dl.c	ダウンローダ用にリセット時の処理を変更。	シリアルフラッシュメモリへの書き込み参考プログラムであるため、リセット時の初期化処理を必要最低限に簡素化しています。(割り込み制御なし等)
r_rspi_dl.h	RSPI シリアルフラッシュメモリ制御モジュール群の外部参照用インクルードヘッダ	表 7.3ローダプログラム (RSPI) と同じ内容です。

【参考】

本アプリケーションノートでは、アプリケーションプログラムやローダプログラムのファイルと区別するため、ダウンローダのファイル名を“XXX_dl.c”としています。

9.3 定数一覧

表 9.3 にダウンローダ サンプルコードで使用する定数を示します。また、7.3 章 ローダプログラム (RSPI) サンプルコードと同じファイルを使用している場合は、記載を省略します。

表9.3 ダウンローダ サンプルコードで使用する定数

定数名	設定値	内容
SF_REQ_PROTECT	0	r_rsapi_dl.h に定義 シリアルフラッシュメモリソフトウェアプロテクト 設定要求用
SF_REQ_UNPROTECT	1	r_rsapi_dl.h に定義 シリアルフラッシュメモリソフトウェアプロテクト 解除要求用
UNPROTECT_WR_STA TUS	H'00	r_rsapi_dl.c に定義 シリアルフラッシュメモリソフトウェアプロテクト モード解除用 * ¹
PROTECT_WR_STATUS	H'5C	r_rsapi_dl.c に定義 シリアルフラッシュメモリソフトウェアプロテクト モード設定用 * ¹
SF_SECTOR_SIZE	H'10000	sh7734_main_dl.c に定義 シリアルフラッシュメモリの 1 セクタサイズ (64 KB) * ¹
SECTOR_NUM	256	sh7734_main_dl.c に定義 シリアルフラッシュメモリの全セクタ数 * ¹
DEVICE_SIZE	SECTOR_SIZE × SECTOR_NUM	sh7734_main_dl.c に定義 シリアルフラッシュメモリの全メモリサイズ
L_PROG_SIZE	H'4000	sh7734_main_dl.c に定義 ローダプログラムのサイズ
L_PROG_SRC	H'E520 0000	sh7734_main_dl.c に定義 ローダプログラム書き込み用データ格納元先頭アド レス (IL メモリのアドレス) 本アプリケーションノートでは、IL メモリ上に ローダプログラムをあらかじめ書き込みます。
L_PROG_DST	0	sh7734_main_dl.c に定義 ローダプログラム書き込み用データ格納先アドレス (シリアルフラッシュメモリのアドレス) SH7734 シリアルブートの仕様としてシリアルフ ラッシュメモリの先頭番地より読み出しを行うた め、シリアルフラッシュメモリの先頭番地にローダ プログラムを書き込む必要があります。
APROG_TOP_RAM	H'AC00 0000	sh7734_main_dl.c に定義 アプリケーションプログラムの先頭アドレス 本アプリケーションノートでは、P2 領域 (キャッ シング不可、MMU アドレス変換不可) にアプリケーシ ョンプログラムを格納しておきます。
APROG_TOP_SFLASH	H'4000	sh7734_main_dl.c に定義 シリアルフラッシュメモリのアプリケーションプロ グラムを格納している先頭アドレス

定数名	設定値	内容
APPINFO_TOP	APROG_TOP_RAM	sh7734_main_dl.c に定義 アプリケーションプログラム先頭アドレスを格納している DDR2-SDRAM アドレス
APPINFO_END	APROG_TOP_RAM + 4	sh7734_main_dl.c に定義 アプリケーションプログラム最終アドレスを格納している DDR2-SDRAM アドレス

*1 詳細は、使用するシリアルフラッシュメモリのデータシートを参照ください。

9.4 関数一覧

表 9.4にダウンローダ サンプルコードで使用する関数を示します。また、7.4章 ローダプログラム (RSPI) サンプルコードと同じ関数は、記載を省略します。

表9.4 ダウンローダ サンプルコードで使用する関数

関数名	概要
main	ダウンローダ メイン処理
write_prog_data	ローダプログラム/アプリケーションプログラムライト処理
init_erase_flag	イレース済みセクタ管理テーブルの初期化
is_erased_sector	対象セクタ イレース済み確認処理
set_erase_flag	対象セクタ イレース済み設定処理
halt	ダウンローダプログラム正常終了時コール
error	ダウンローダプログラム Error 終了時コール
R_RSPI_SFProtectCtrl	シリアルフラッシュメモリソフトウェアプロテクト設定/解除
R_RSPI_SFSectorErase	シリアルフラッシュメモリ対象セクタイレース処理
R_RSPI_SFByteProgram	RSPI シリアルフラッシュメモリライト処理 (1 バイト単位)

9.5 関数仕様

サンプルコードの関数仕様を示します。また、7.5章 ローダプログラム (RSPI) サンプルコードと同じ関数は、記載を省略します。

main	
概要	ダウンローダ メイン処理
ヘッダ	r_rspi_dl.h
宣言	void main(void)
説明	<ul style="list-style-type: none"> 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処理を行います。 本アプリケーションノートでは、デバッグ用途として、ダウンローダ処理中はLED4を点灯します。 書き込み時にローダプログラム、アプリケーションプログラムを格納するアドレスと実行時に配置されるアドレスが同じであるため、ブート起動用内蔵ROMプログラムまたは、ローダプログラムが正しくプログラムをロードできなくても、書き込み時に格納したプログラムが動作して、正しく動作していると誤認識する可能性があります。このため、本アプリケーションノートでは、シリアルフラッシュメモリへの書き込みの際に使用したILメモリ上のローダプログラムを格納していた領域とDDR2-SDRAM上のアプリケーションプログラムを格納していた領域を消去します。(任意)
引数	なし
リターン値	なし
write_prog_data	
概要	ローダプログラム/アプリケーションプログラムライト処理
ヘッダ	r_rspi_dl.h
宣言	static int32_t write_prog_data(uint8_t *program_data, const uint32_t sflash_addr, const uint32_t size)
説明	<ul style="list-style-type: none"> シリアルフラッシュメモリへライトする前にライト対象のセクタをイレースします。 ローダプログラム/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ライトします。 シリアルフラッシュメモリへライト完了後、Verify処理を行います。
引数	uint8_t *program_data : ライトデータ格納用ポインタ const uint32_t sflash_addr : ライト対象のシリアルフラッシュメモリアドレス const uint32_t size : ライトデータサイズ(1バイト単位)
リターン値	0 : ライト成功 0以外 : ライト失敗
init_erase_flag	
概要	イレース済みセクタ管理テーブルの初期化
ヘッダ	
宣言	static void init_erase_flag(void)
説明	<ul style="list-style-type: none"> イレース済みセクタ管理テーブルの初期化を行います。 イレース済みセクタ管理テーブルは、既にイレース済みのセクタであれば、イレース処理をスキップするために使用されます。
引数	なし
リターン値	なし

is_erased_sector	
概要	対象セクタ イレース済み確認処理
ヘッダ	
宣言	static volatile uint8_t is_erased_sector(const uint32_t sector_no)
説明	<ul style="list-style-type: none"> パラメータ指定されたセクタが既にイレース済みであるかどうか確認します。
引数	uint32_t sector_no : イレース済み確認対象セクタ
リターン値	1 : イレース済み 0 : イレースしていない
set_erase_flag	
概要	対象セクタ イレース済み設定処理
ヘッダ	
宣言	static void set_erase_flag(const uint32_t sector_no)
説明	<ul style="list-style-type: none"> パラメータ指定されたセクタに対してイレース済みセクタ管理テーブルをイレース済みに更新します。
引数	uint32_t sector_no : イレース済み対象セクタ
リターン値	なし
halt	
概要	ダウンロードプログラム正常終了時コール
ヘッダ	
宣言	static void halt(void)
説明	<ul style="list-style-type: none"> ダウンロードプログラム正常終了時にコールされます。 内部で無限ループするのみです。
引数	なし
リターン値	なし
error	
概要	ダウンロードプログラム Error 終了時コール
ヘッダ	
宣言	static void error(void)
説明	<ul style="list-style-type: none"> ダウンロードプログラム Error 終了時にコールされます。 内部で無限ループするのみです。
引数	なし
リターン値	なし
R_RSPI_SFProtectCtrl	
概要	シリアルフラッシュメモリソフトウェアプロテクト設定/解除
ヘッダ	r_rspi_dl.h
宣言	void R_RSPI_SFProtectCtrl(const enum sf_req req)
説明	<ul style="list-style-type: none"> パラメータに従い、シリアルフラッシュメモリソフトウェアプロテクトを設定/解除します。
引数	enum sf_req req UNPROTECT_WR_STATUS : プロテクト解除 PROTECT_WR_STATUS : プロテクト設定
リターン値	なし

R_RSPI_SFSectorErase

概要	シリアルフラッシュメモリ対象セクタ イレース処理
ヘッダ	r_rspi_dl.h
宣言	void R_RSPI_SFSectorErase(const uint32_t sector_no)
説明	● パラメータ指定されたシリアルフラッシュメモリのセクタをイレースします。
引数	uint32_t sector_no : イレース対象セクタ
リターン値	なし

R_RSPI_SFByteProgram

概要	RSPI シリアルフラッシュメモリライト処理 (1 バイト単位)
ヘッダ	r_rspi_dl.h
宣言	void R_RSPI_SFByteProgram(const uint32_t addr, const uint8_t *buf, const uint32_t size)
説明	● パラメータ指定されたシリアルフラッシュメモリアドレスへ指定サイズ分の指定データを 1 バイト単位でライトします。 ● SPCMD レジスタの転送データ長 (SPB[0:3]ビット) に 8 ビット長指定します。
引数	uint32_t addr : ライト対象のシリアルフラッシュメモリアドレス uint8_t *buf : ライトデータ格納用ポインタ uint32_t size : ライトサイズ (1 バイト単位)
リターン値	なし

9.6 フローチャート

図 9.3にダウンローダ メイン処理のフローチャートを示します。OL メモリ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。

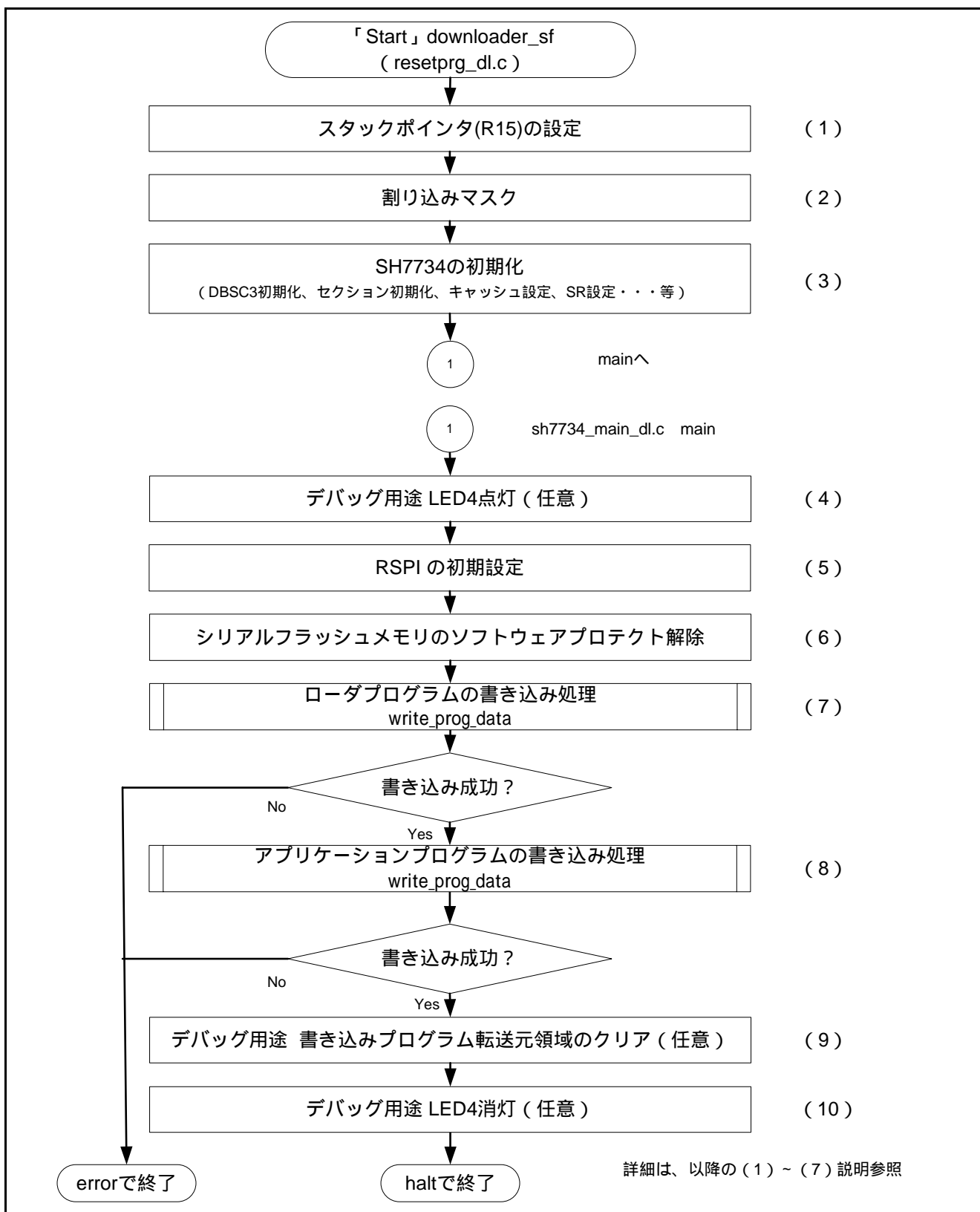


図9.3 ダウンローダ処理 フローチャート

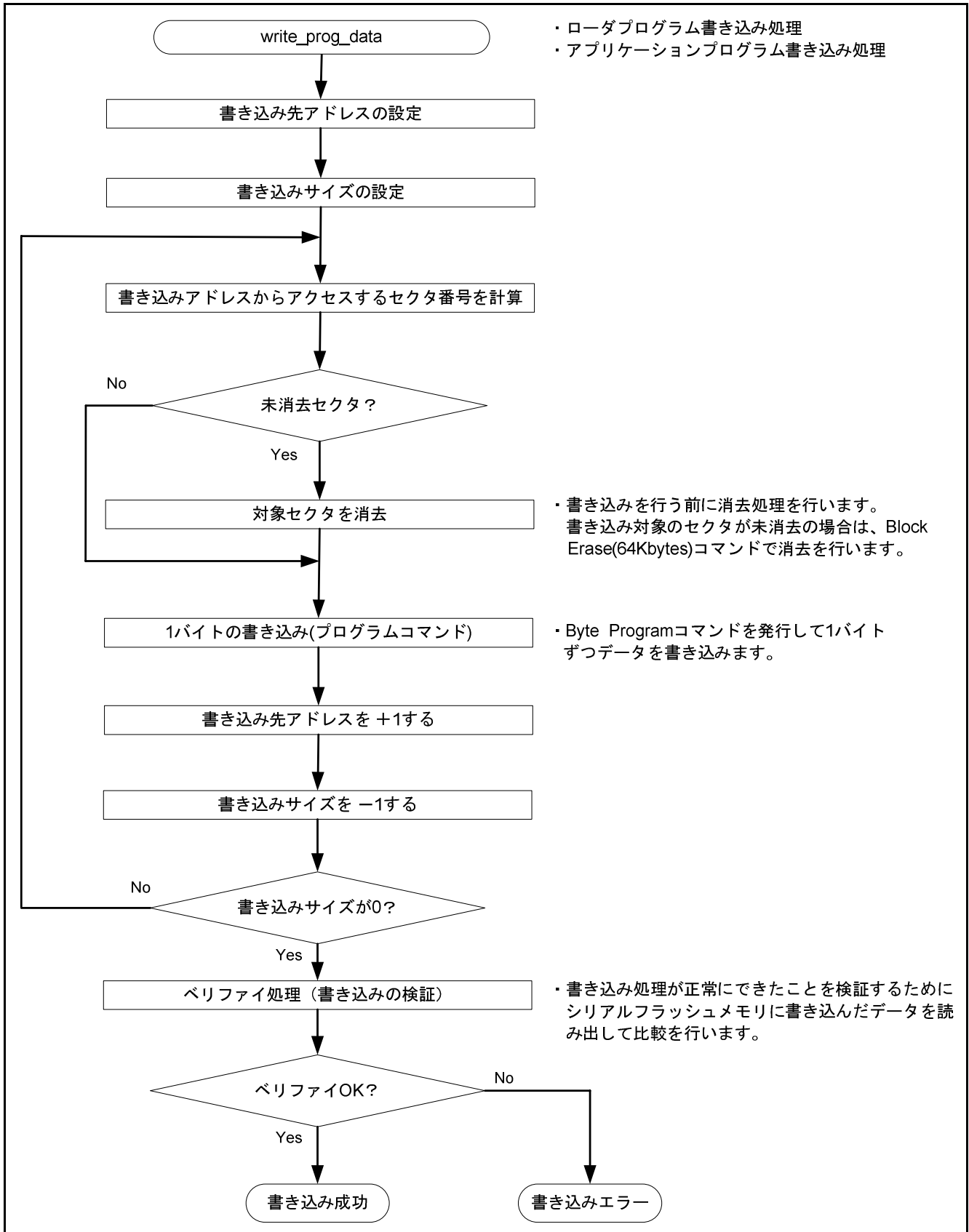


図9.4 シリアルフラッシュへの書き込み処理 (write_prog_data()) フローチャート

(1) スタックポインタの設定

本アプリケーションノートでは、スタックポインタ (R15) に H'E501 2000 番地 (OL メモリ) を設定します。図 9.5 のように #pragma entry で指定します。

```
#pragma entry downloader_sf (sp=0xE5012000)
```

図9.5 ダウンローダプログラム スタックポインタ設定

(2) 割り込みマスク

本アプリケーションノートでは、ダウンローダは割り込み処理を行いません。SR レジスタは、パワーオンリセット後の初期値の状態 (BL=1) とします。

(3) SH7734 の初期化処理

ダウンローダを動作させるための最低限の処理 (DBSC3 初期化、セクション初期化、キャッシュ設定、SR レジスタ設定・・・等) を行います。

本アプリケーションノートでは、SH7734 の初期化処理について、「SH7734 グループ SH7734 初期設定例 (R01AN0665JJ)」を流用していますので、詳細は、そちらを参照ください。

(4) LED4 点灯 (任意)

本アプリケーションノートでは、デバッグ用途として、シリアルフラッシュへの書き込み処理をしている間、LED4 を点灯します。

(5) RSPI 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、RSPI の初期設定を行います。

(6) ソフトウェアプロテクトの解除

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。

(7) ローダプログラムの書き込み処理

ダウンロードは、IL メモリの H'E520 0000 ~ H'E520 3FFF 番地に配置済みの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 3FFF 番地へ書き込みます。表 9.5 にローダプログラムの書き込み処理を示します。

表9.5 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 (IL メモリ アドレス)	H'E520 0000
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 アドレス)	H'0000 0000 (固定)
転送サイズ	H'4000 (固定)
書き込み処理手順	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。

(8) アプリケーションプログラムの書き込み処理

ダウンロードは、DDR2-SDRAM に配置済みの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 4000 番地から書き込みます。表 9.6 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9.6 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 (DDR2-SDRAM)	アプリケーションプログラム内にある appinfo 配置アドレス downloader ワークスペース sh7734_main.c #define APROG_TOP_RAM で定義されます。
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 4000 (固定)
転送サイズ	アプリケーションプログラム内にある appinfo から計算 (アプリケーションプログラムに依存)
書き込み処理手順	1. 書き込み先アドレスが消去済みかどうかを確認します。 2. 未消去エリアの場合は消去処理を行います。 3. プログラムコマンドを発行し書き込み処理を行います。書き込みは 1 バイト単位で行っています。

(9) 書き込みプログラム転送元領域のクリア (任意)

本アプリケーションノートでは、ブート時に、確実にシリアルフラッシュメモリから、ローダプログラム、アプリケーションプログラムをロードすることを確認するため、念のため、シリアルフラッシュメモリへの書き込みの際に使用した IL メモリ上のローダプログラムを格納していたデータ領域と DDR2-SDRAM 上のアプリケーションプログラムを格納していたデータ領域を消去します。

(10) LED4 消灯 (任意)

本アプリケーションノートでは、デバッグ用途として、シリアルフラッシュへの書き込み処理が完了した際に、LED4 を消灯します。

9.7 シリアルフラッシュメモリのコマンド

表 9.7 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。RSPI 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9.7 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(ソフトウェアプロテクトの解除)
Sector Erase (64Kbytes)	H'D8	セクタ消去（64KB）
Page Program	H'02	データの書き込み（1バイト）

10. 応用例

10.1 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、本アプリケーションノートの[sh7734_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。

10.1.1 ダウンローダの使用方法

本アプリケーションノートのダウンローダは、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、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。

10.1.2 ダウンローダの自動化 (コマンドバッチファイル)

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、ローダプログラムを IL メモリ、ダウンローダを OL メモリに、アプリケーションプログラムを DDR2-SDRAM 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。

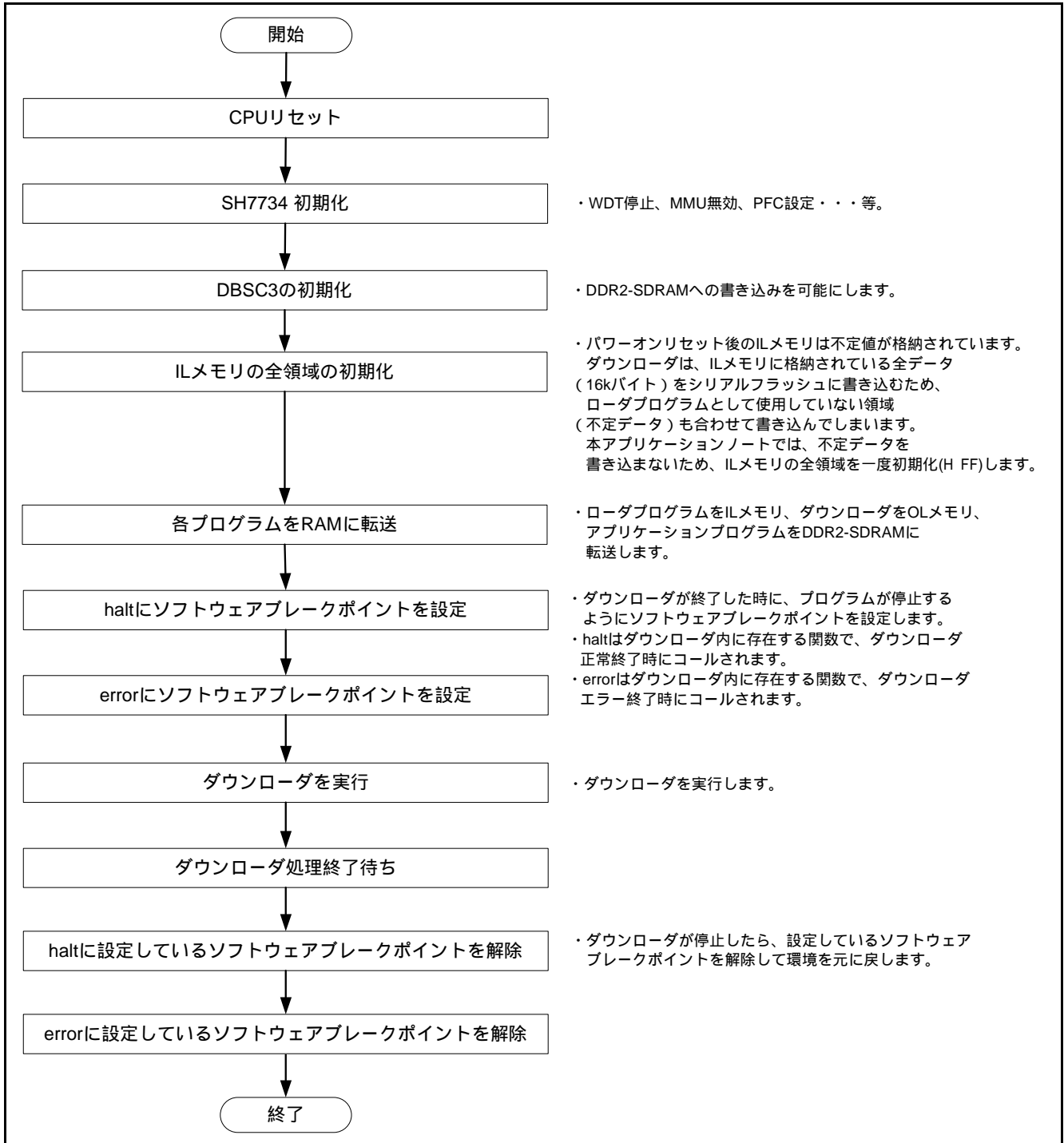


図10.1 コマンドバッチファイルのフローチャート

10.1.3 ダウンロードモジュールとバッチファイルの登録

図 10.2に[sh7734_sflash_app]ワークスペースのディレクトリ構成を示します。[sh7734_sflash_app]ワークスペースでは、図 10.2の 、 、 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 10.2の のバッチファイルをプロジェクトに登録しています。

```

¥sh7734_sflash_app      : ワークスペースディレクトリ
|-sh7734_sflash_app    : プロジェクトディレクトリ
|  |-debug              :
|  |  |-sh7734_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
|  |  |-sh7734_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
|  |  |-sh7734_sflash_loader_prog.abs : ローダプログラムの実行ファイル-----④

```

図10.2 [sh7734_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。

10.1.4 プログラム書き込み手順

ここでは、[sh7734_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. [sh7734_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[sh7734_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 10.3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。

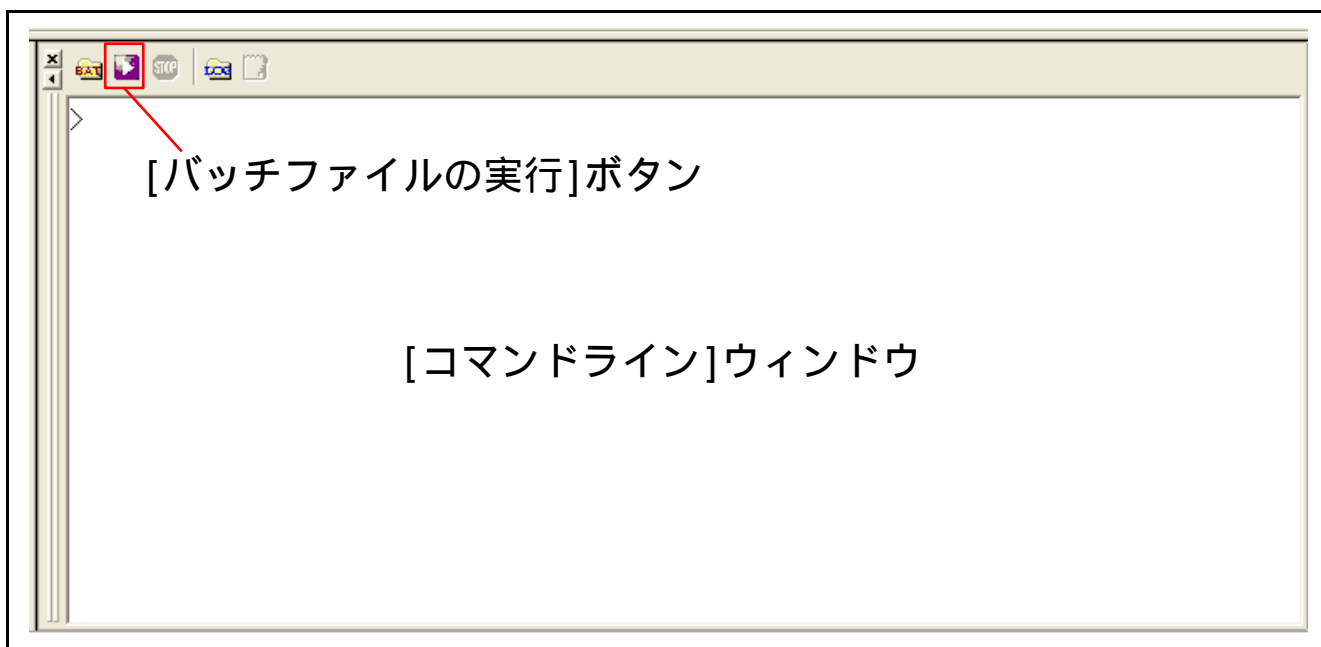


図10.3 [コマンドライン]ウィンドウと[バッチファイルの実行]ボタン

- 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 10.4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。
- 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。

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。

⇒ E500F172	E101	_halt	MOV	#H'01,R1
E500F174	8013		MOV	R1,R0
E500F176	8800		CMP/EQ	#H'00,R0
E500F178	8903		BT	@H'E500F182:8
E500F17A	E101		MOV	#H'01,R1
E500F17C	8013		MOV	R1,R0
E500F17E	8800		CMP/EQ	#H'00,R0
E500F180	8BFB		BF	@H'E500F17A:8
E500F182	000B		RTS	
E500F184	0009		NOP	
E500F186	E101	_error	MOV	#H'01,R1
E500F188	8013		MOV	R1,R0
E500F18A	8800		CMP/EQ	#H'00,R0
E500F18C	8903		BT	@H'E500F196:8
E500F18E	E101		MOV	#H'01,R1
E500F190	8013		MOV	R1,R0
E500F192	8800		CMP/EQ	#H'00,R0
E500F194	8BFB		BF	@H'E500F18E:8
E500F196	000B		RTS	
E500F198	0009		NOP	

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

10.2 PC ブレーク機能 (BREAKPOINT) 使用の注意事項

PC ブレーク機能 (BREAKPOINT) を使用する際の注意事項を記載します。

PC ブレーク機能は、指定アドレスの命令を専用命令に置きかえることでブレークする機能です。

ローダプログラムやアプリケーションプログラムを起動前に設定した PC ブレーク機能(専用命令に置きかえ)は、ブート過程において通常の命令で上書きされるため、ブレークすることができません。

このため、ローダプログラムやアプリケーションプログラムのブレークは、ハードウェアブレーク機能を使用してください。

PC ブレーク機能を使用する場合は、それぞれのプログラム実行後、1度、ハードウェアブレーク機能でブレークした後、PC ブレーク機能を設定することによりブレークさせることができます。

10.3 Quad-SPI 対応 ロードプログラム (参考)

ロードプログラム部分について、転送時間を短縮するために、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 (Quad-SPI) を使用した別ワークスペースを用意しています。

10.3.1 サンプルコードの構成

本アプリケーションノートでは、Quad-SPI 対応のロードプログラム、アプリケーションプログラム、ダウンロードで異なるワークスペースを用意しています。表 10.1 のように 3 つのワークスペースに分かれています。

表10.1 サンプルコードの構成

ワークスペース名	説明
sh7734_qsfash_loader_prog (ロードプログラム (Quad-SPI))	このワークスペースのプロジェクトは、Quad-SPI 対応ロードプログラムをビルドします。
sh7734_sflash_app (アプリケーションプログラム)	RSPI と同じものを使用します。
sh7734_sflash_downloader (ダウンロード)	RSPI と同じものを使用します。 【注】本アプリケーションノートでは、Quad-SPI 動作でのシリアルフラッシュメモリへの書き込みは対応していません。

10.3.2 Quad-SPI 対応 ロードプログラムの特徴

(1) 使用する周辺機能

シリアルフラッシュメモリへのアクセスに RSPI でなく Quad-SPI を使用します。

(2) ロードプログラムの動作概要

シリアルフラッシュメモリへのアクセスに RSPI でなく Quad-SPI を使用する以外は、1.3.2章ロードプログラムの動作概要と同じになります。

(3) 転送効率

1 バイトのデータを転送する場合は、RSPI の場合は、図 10.5 のように 8 クロックを要しますが、Quad-SPI (Quad-SPI モード) の場合は、2 クロックでの転送が可能となります。同じビットレートであれば、Quad-SPI の方が、4 倍転送効率が良くなります。

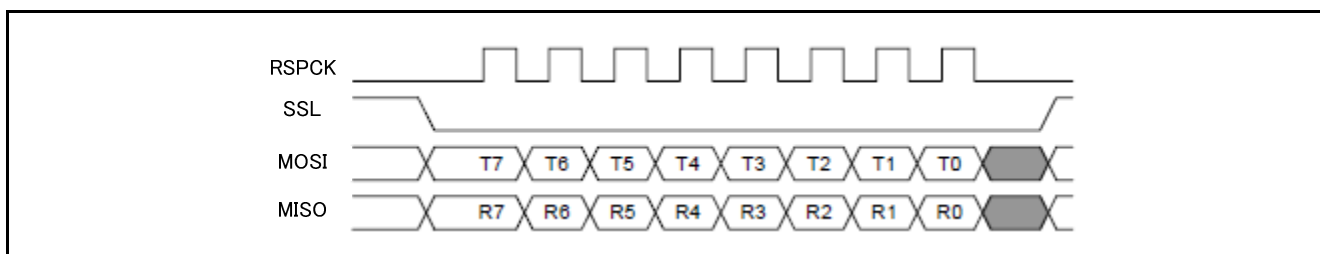


図10.5 RSPI 転送フォーマット

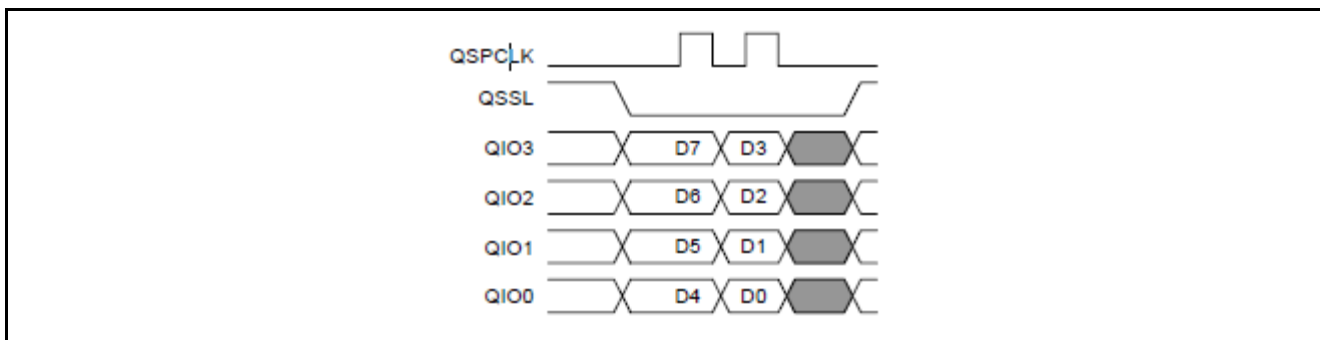


図10.6 Quad-SPI (Quad-SPI モード) 転送フォーマット

(4) シリアルフラッシュ、IL メモリのメモリマップ

RSPI と同じです。詳細は、7.1.1章参照ください。

(5) SPI モード、ビットレート仕様

表10.2 ローダプログラム (Quad-SPI) の SPI モード、ビットレート仕様

項目	内容
SPI モード	SPI モード 3 (CPOL=1, CPHA=1)
ビットレート	12.5MHz

(6) ファイル構成

ローダプログラム (RSPI) との差分のみ記載を示します。

表10.3 ローダプログラム (Quad-SPI) で使用するファイル

ファイル名	概要	備考
sh7734_main_lp.c	ローダプログラム (Quad-SPI) のメイン処理	
r_dmac_lp.c	Quad-SPI HPB-DMAC 制御モジュール群	ローダプログラム (RSPI) では、r_rspl_lp.c で実装していたが、別ファイル化した。
r_rqspi_lp.c	Quad-SPI 制御モジュール群	
r_rqspi_qserial_flash_lp.c	Quad-SPI シリアルフラッシュメモリ制御モジュール群	ローダプログラム (RSPI) では、r_rspl_lp.c で実装していたが、シリアルフラッシュメモリ制御に関するユーザ IF 部分を別ファイル化した。
r_rqspi_lp.h	Quad-SPI 制御モジュール群の外部参照用インクルードヘッダ	
r_rqspi_qserial_flash_lp.h	Quad-SPI シリアルフラッシュメモリ制御モジュール群の外部参照用インクルードヘッダ	

(7) 定数、関数一覧

RSPI と異なります。詳細は、sh7734_qsflash_loader_prog ワークスペースのサンプルコードを参照ください。

(8) フローチャート

アプリケーションプログラムの転送に RSPI でなく Quad-SPI を使用する以外は、7.6章フローチャートと同じになります。

10.3.3 シリアルフラッシュメモリへの Quad-SPI 対応 ロードプログラムの書き込み方法

High-performance Embedded Workshop の Session 設定を DefaultSession から qsflashSession に変更し、以下の部分に sh7734_qsflash_loader_prog ワークスペースで作成した.abs ファイルを格納ください。

それ以外に関しては、10.1章の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と基本的には同じです。

```

¥sh7734_sflash_app      : ワークスペースディレクトリ
|-sh7734_sflash_app    : プロジェクトディレクトリ
|  |-debug              :
|  |-sh7734_sflash_app.abs : アプリケーションプログラムの実行ファイル-----RSPIと同じ
|  :
|-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ドプログラム格納用ディレクトリ
|-sh7734_sflash_downloader.abs : ダウンローダの実行ファイル-----RSPIと同じ
|-downloader.hdc      : バッチファイル(ダウンローダ起動用)-----RSPIと同じ
|-qsflash              : ロードプログラムの実行ファイル格納フォルダ
|-sh7734_sflash_loader_prog.abs : ロードプログラムの実行ファイル-----①

```

図10.7 Quad-SPI 対応時の[sh7734_sflash_app]ワークスペースのディレクトリ構成

11. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

12. 参考ドキュメント

- ユーザーズマニュアル：ハードウェア
SH7734 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。)
- テクニカルアップデート/テクニカルニュース
(最新の情報をルネサス エレクトロニクスホームページから入手してください。)
- ユーザーズマニュアル：開発環境
SuperH C/C++コンパイラパッケージ V.9.04 ユーザーズマニュアル 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/contact/>

改訂記録	SH7734 グループ アプリケーションノート シリアルフラッシュ メモリからのブート例
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.09.24	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

製品ご使用上の注意事項

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

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

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

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

5. 製品間の相違について

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

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

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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