
SH7216グループ

R01AN0388JJ0101

Rev.1.01

2012.06.15

ユーザプログラムモード使用例

要旨

本アプリケーションノートは、SH7216 ユーザプログラムモードでの内蔵フラッシュ書き換えプログラム動作例について説明しています。内蔵フラッシュに書き込むデータは、SH7216 に接続した外部デバイスが保持しており、シリアルコミュニケーションインターフェースを使用して通信を行います。

本アプリケーションノートで紹介する内蔵フラッシュ書き換えプログラムは、SH7216 のユーザマット上にあるものとしてします。また、内蔵フラッシュの書き換え処理部に関しては、ルネサス エレクトロニクスが提供している SH-2, SH-2A 用シンプルフラッシュ API (以下、標準 API と表記します) を使用しています。

動作確認デバイス

SH7216

目次

1. はじめに.....	2
2. 概要および使用機能説明.....	4
3. 参考プログラムの外部仕様.....	11
4. 参考プログラムの内部仕様.....	19
5. 参考プログラムリスト.....	25
6. 参考ドキュメント.....	31

1. はじめに

1.1 仕様

本応用例では、ユーザプログラムモードを使用して、内蔵フラッシュへの書き込み／消去処理、および読み出し処理を行います。ユーザプログラムモードでは、任意のインタフェースを使用して、これらの処理を制御することができます。本応用例では、ホスト PC とのシリアル通信を使用してこれらの処理を制御します。

ユーザアプリケーション実行中に、ホスト PC からユーザ制御コマンドであるフラッシュ書き込み／消去処理コマンドを受信すると、内蔵フラッシュへの書き込み／消去処理を実行し、フラッシュ読み出し処理コマンドを受信すると、フラッシュの読み出し処理を実行します。

図 1 に本応用例のシステム構成を示します。

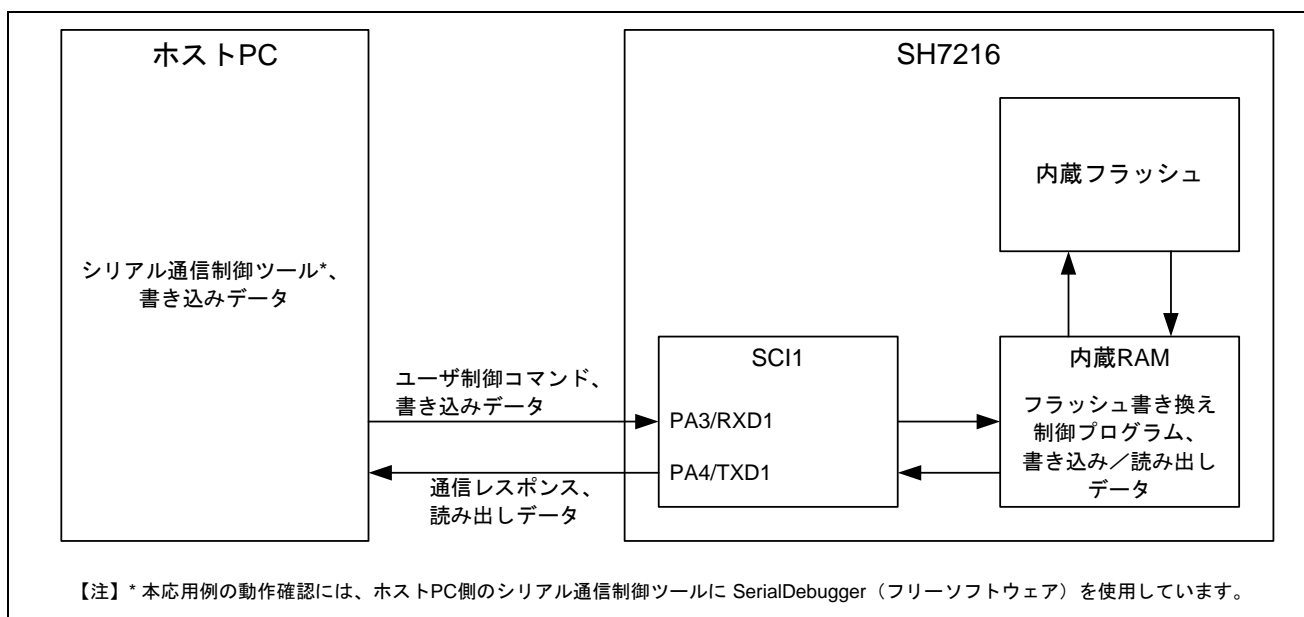


図1 本応用例のシステム構成

1.2 使用機能

- シリアルコミュニケーションインタフェース (SCI)
- 内蔵フラッシュ専用シーケンサ (FCU)

1.3 適用条件

マイコン	SH7216 (R5F72167A : フラッシュ 1 M バイト品)
動作周波数	内部クロック : 200 MHz バスクロック : 50 MHz 周辺クロック : 50 MHz
統合開発環境	ルネサス エレクトロニクス製 High-performance Embedded Workshop Ver.4.07.00
C コンパイラ	ルネサス エレクトロニクス製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.03 Release 00
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 -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 関連アプリケーションノート

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

- SH7216 グループ 初期設定例
- SH ファミリ SH-2, SH-2A 用シンプルフラッシュ API

2. 概要および使用機能説明

本応用例では、SH7216 と外部デバイスの接続にシリアルコミュニケーションインタフェース (SCI) を使用しています。また、SH7216 では、内蔵フラッシュの書き込み/消去に専用のシーケンサ (FCU) を使用します。

2.1 使用機能説明

2.1.1 シリアルコミュニケーションインターフェース (SCI) 機能説明

SCI は、調歩同期式とクロック同期式の 2 方式のシリアル通信が可能なモジュールです。また、全二重通信が可能で、送信部および受信部ともにダブルバッファ構造となっていますので、シリアルデータの連続送信、連続受信が実現できます。

本応用例では、SH7216 と外部デバイス間でのハンドシェイク通信及び、内蔵フラッシュの書き換えデータ通信に SCI を使用しています。

図 2 に SCI のブロック図を示します。

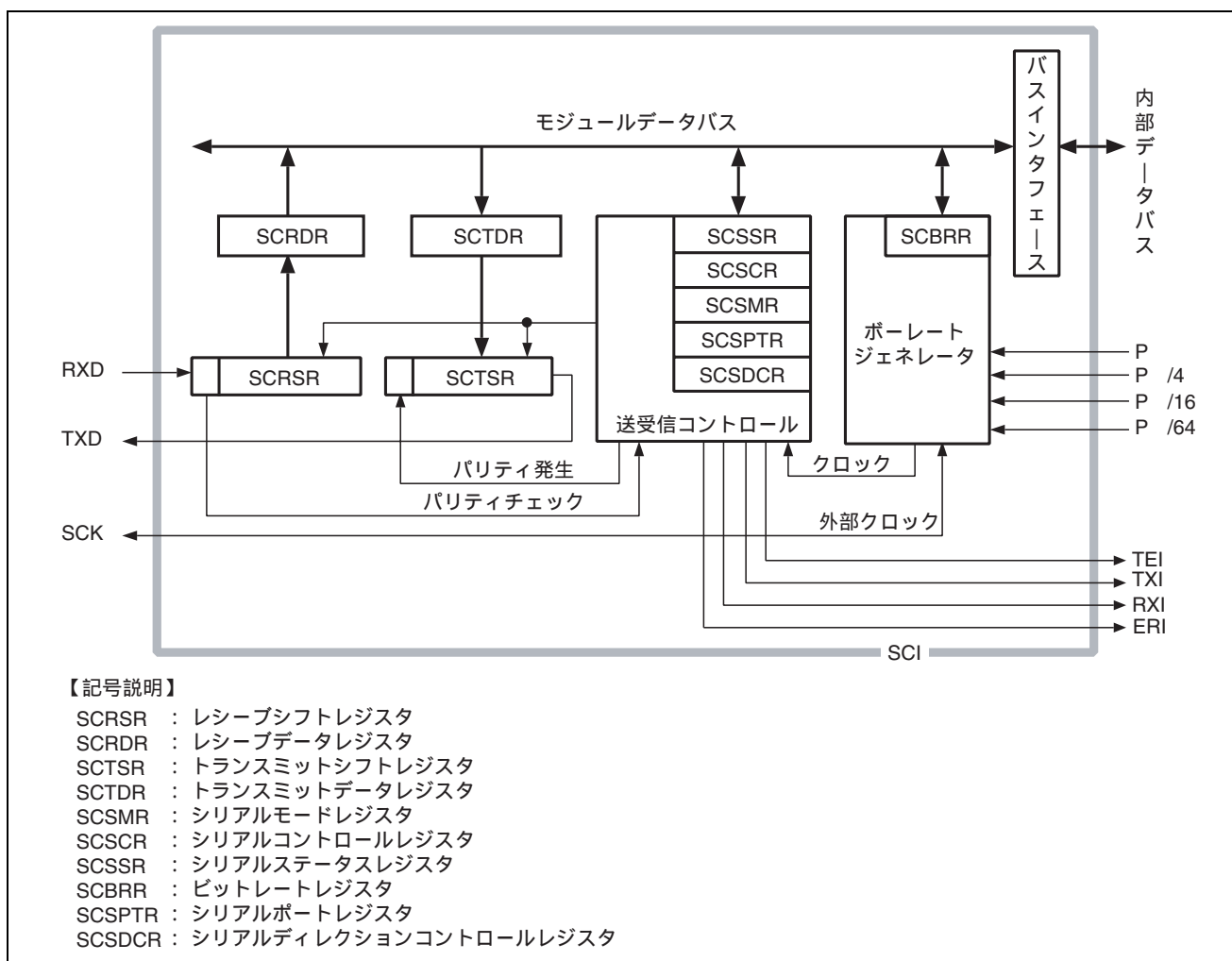


図2 SCI のブロック図

2.1.2 内蔵フラッシュ専用シーケンサ (FCU) 機能説明

SH7216 は、FCU による内蔵フラッシュの書き換えを行います。

図 3に内蔵フラッシュのブロック図を示します。

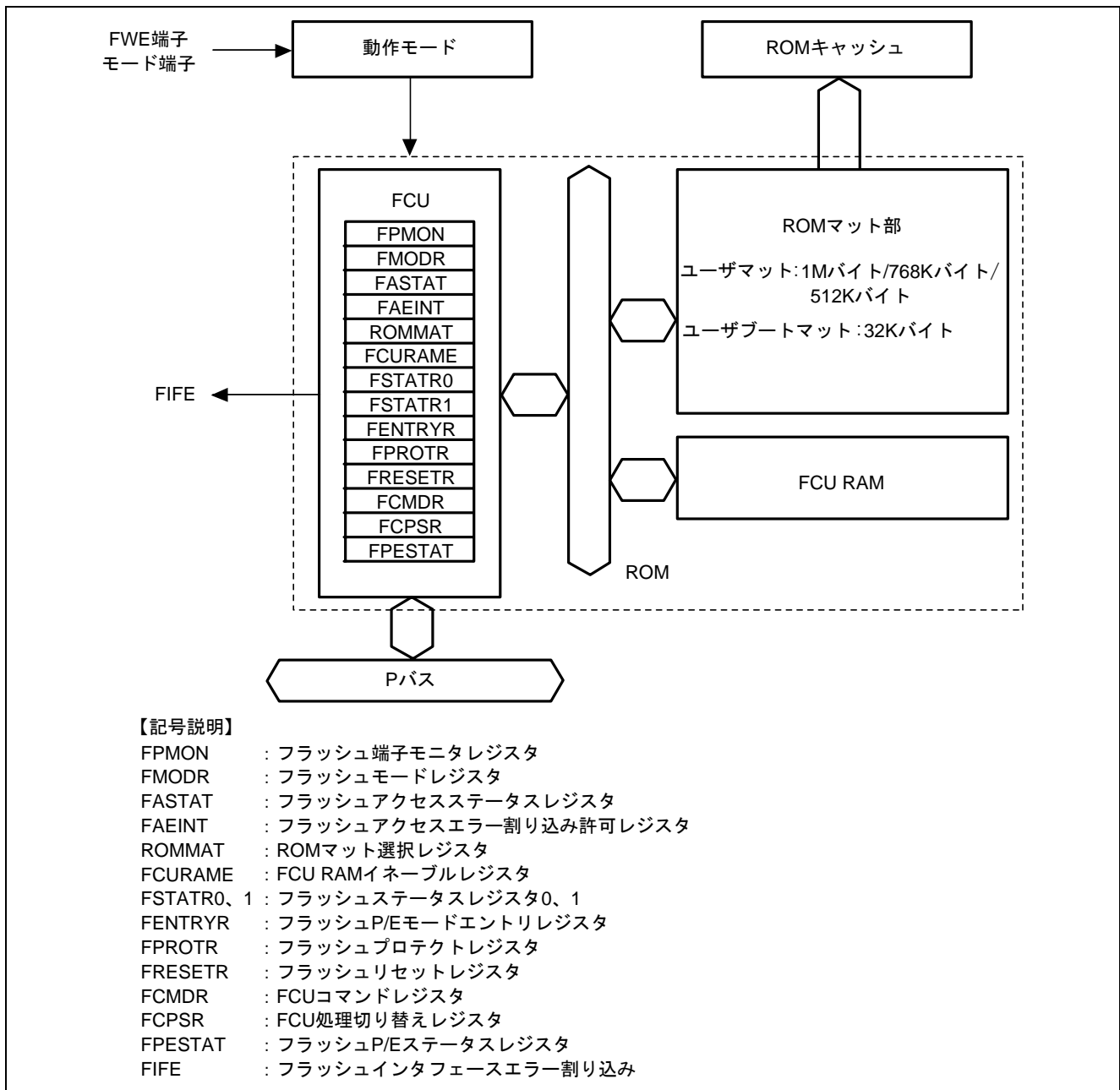


図3 内蔵フラッシュのブロック図

2.2 内蔵フラッシュ書き込み／消去動作説明

SH7216では、FCUを使用して、内蔵フラッシュの書き込み／消去を行います。ここでは、内蔵フラッシュの書き換え手順を説明しています。より詳細な内容は、ハードウェアマニュアルを参照してください。また、本応用例では、内蔵フラッシュの書き込み／消去処理に標準APIを使用しています。標準APIの詳細は、関連アプリケーションノートを参照してください。

2.2.1 内蔵フラッシュ書き込み／消去準備

FCUを使用するためには、FCURAMにFCU用のファームウェア（FCUファーム）を格納する必要があります。FCUファームの転送後、FCUコマンドを発行することで、FCUが内蔵フラッシュの書き込み／消去を行います。

FCUファームは、デバイス内部のFCUファーム領域に格納されており、起動時にFCURAMへ転送する必要があります。また、デバイスの起動時には、FCURAMへのアクセスが禁止となっているため、レジスタ設定でアクセスを許可する必要があります。

図4にFCUコマンドの使用準備フローを示します。

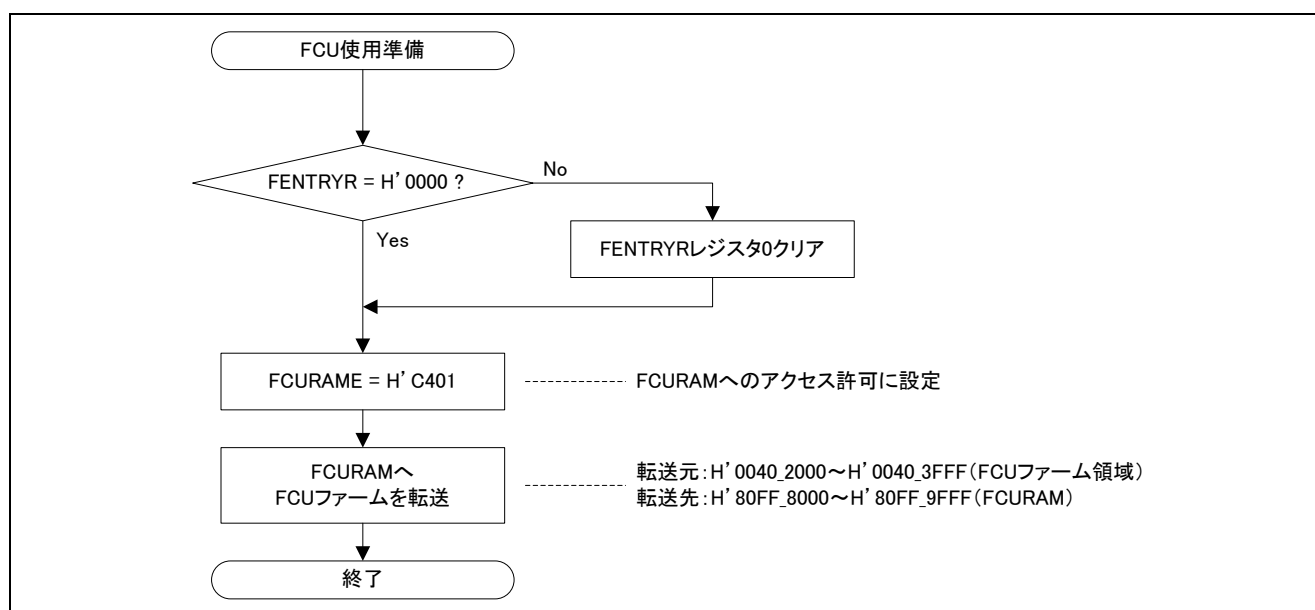


図4 FCU コマンドの使用準備フロー

2.2.2 内蔵フラッシュ消去

本デバイスでは、内蔵フラッシュを複数のブロックに分割しており、ブロック単位で消去を行います。FCUファーム転送後、消去対象ブロックのアドレスへ消去コマンド^{※1}と実行コマンドを書き込むことで、FCUがブロック消去を実行します。

図5にSH7216の消去ブロック割りを、表1に各ブロックとそのアドレス範囲を示します。また、図6に内蔵フラッシュの消去フロー^{※2}を示します。

【注】※1 消去コマンドは、内蔵フラッシュの書き込み／消去アドレスであれば、どこでも受け付けられません。

※2 図6のフローは、標準APIのフローではありません。

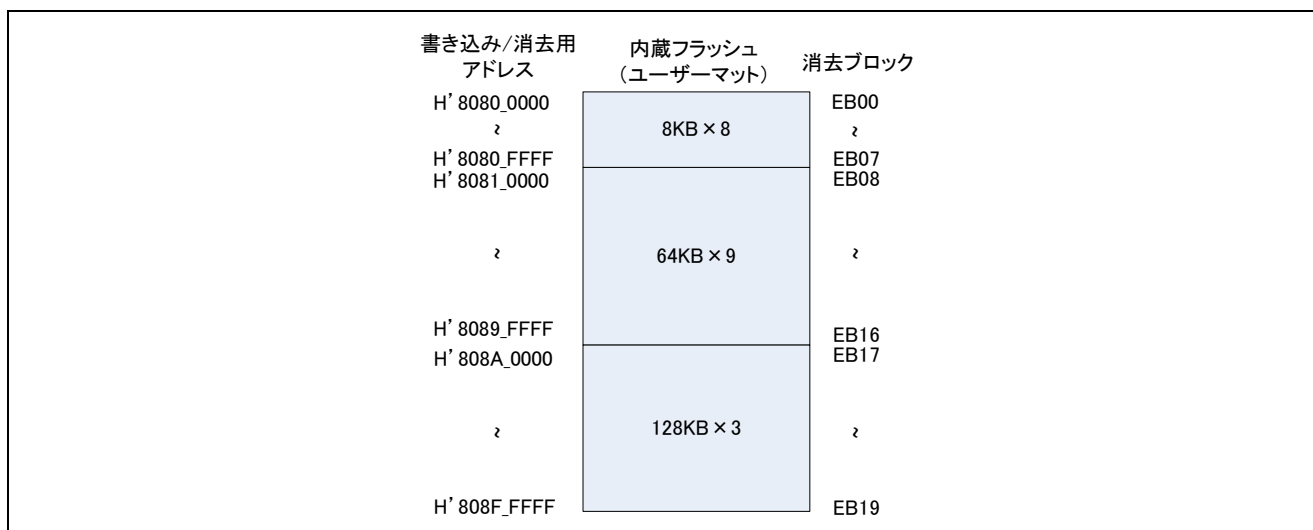


図5 内蔵フラッシュの消去ブロック割

表1 消去ブロックとアドレス

消去ブロック	実アドレス	書き込み/消去用アドレス	単位容量
EB00	H'0000_0000~H'0000_1FFF	H'8080_0000~H'8080_1FFF	8K バイト
EB01	H'0000_2000~H'0000_3FFF	H'8080_2000~H'8080_3FFF	
EB02	H'0000_4000~H'0000_5FFF	H'8080_4000~H'8080_5FFF	
EB03	H'0000_6000~H'0000_7FFF	H'8080_6000~H'8080_7FFF	
EB04	H'0000_8000~H'0000_9FFF	H'8080_8000~H'8080_9FFF	
EB05	H'0000_A000~H'0000_BFFF	H'8080_A000~H'8080_BFFF	
EB06	H'0000_C000~H'0000_DFFF	H'8080_C000~H'8080_DFFF	
EB07	H'0000_E000~H'0000_FFFF	H'8080_E000~H'8080_FFFF	
EB08	H'0001_0000~H'0001_FFFF	H'8081_0000~H'8081_FFFF	64K バイト
EB09	H'0002_0000~H'0002_FFFF	H'8082_0000~H'8082_FFFF	
EB10	H'0003_0000~H'0003_FFFF	H'8083_0000~H'8083_FFFF	
EB11	H'0004_0000~H'0004_FFFF	H'8084_0000~H'8084_FFFF	
EB12	H'0005_0000~H'0005_FFFF	H'8085_0000~H'8085_FFFF	
EB13	H'0006_0000~H'0006_FFFF	H'8086_0000~H'8086_FFFF	
EB14	H'0007_0000~H'0007_FFFF	H'8087_0000~H'8087_FFFF	
EB15	H'0008_0000~H'0008_FFFF	H'8088_0000~H'8088_FFFF	
EB16	H'0009_0000~H'0009_FFFF	H'8089_0000~H'8089_FFFF	128K バイト
EB17	H'000A_0000~H'000B_FFFF	H'808A_0000~H'808B_FFFF	
EB18	H'000C_0000~H'000D_FFFF	H'808C_0000~H'808D_FFFF	
EB19	H'000E_0000~H'000F_FFFF	H'808E_0000~H'808F_FFFF	

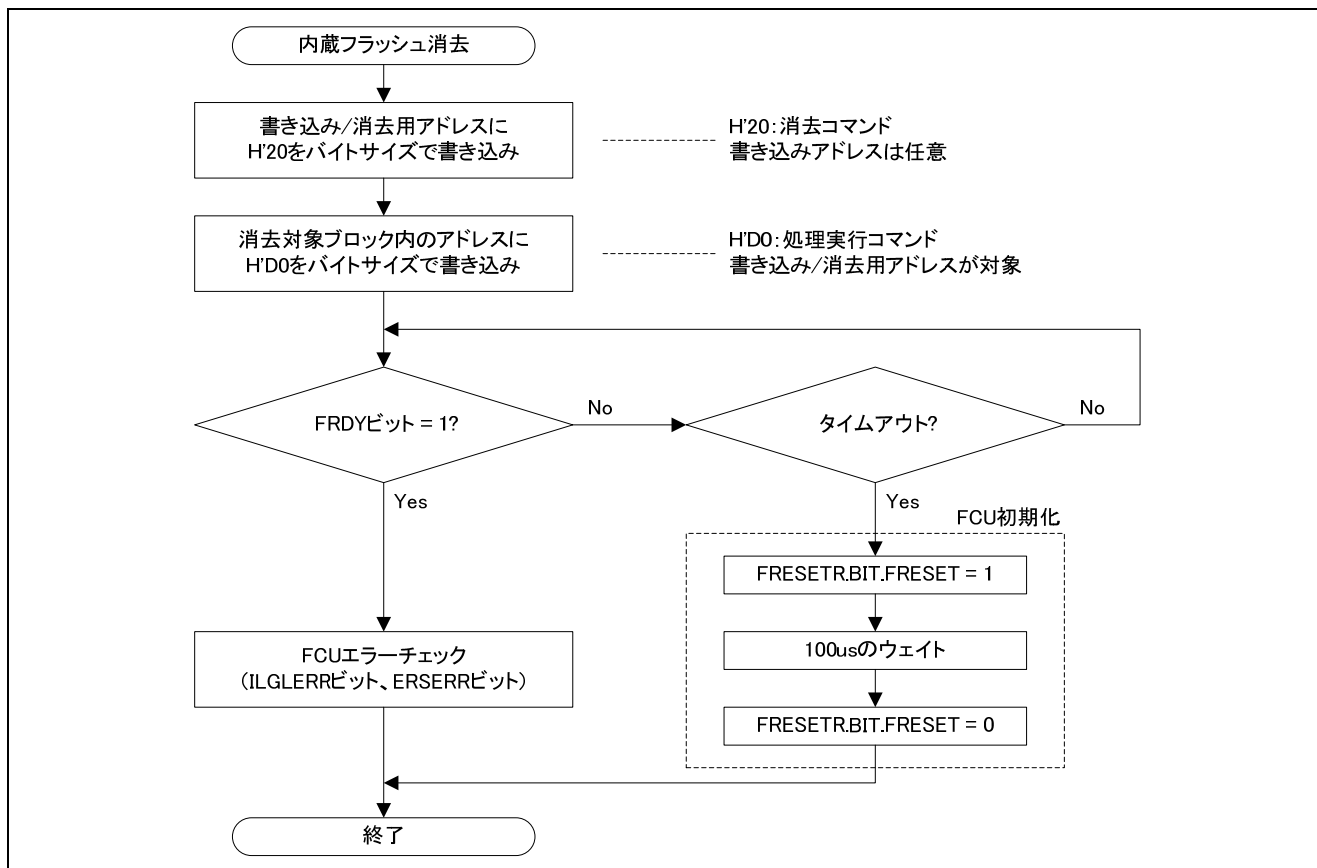


図6 内蔵フラッシュ消去フロー

2.2.3 内蔵フラッシュ書き込み

内蔵フラッシュの書き込みは、消去状態の領域にのみ可能で、ユーザマットへの1回の書き込みは256バイト単位で行います。書き込み手順は、消去同様FCUへのコマンド発行により、FCUが実行します。書き込み/消去用アドレスへ書き込みコマンド、書き込みサイズ^{※1}を発行し、続けて書き込み先アドレス^{※2}へ書き込みデータ(256バイト)をセット^{※3}します。

図7に内蔵フラッシュ書き込みフロー^{※4}を示します。

- 【注】
- ※1 ユーザマット、ユーザブートマットへの書き込みは、256バイト固定です(H'80を発行)。
 - ※2 書き込みアドレスに対してH'8080_0000を足したアドレス(書き込み/消去用アドレス)です。
 - ※3 書き込みデータは、ワードサイズで書き込み/消去用アドレスへ書き込みます。
 - ※4 図7のフローは、標準APIのフローではありません。

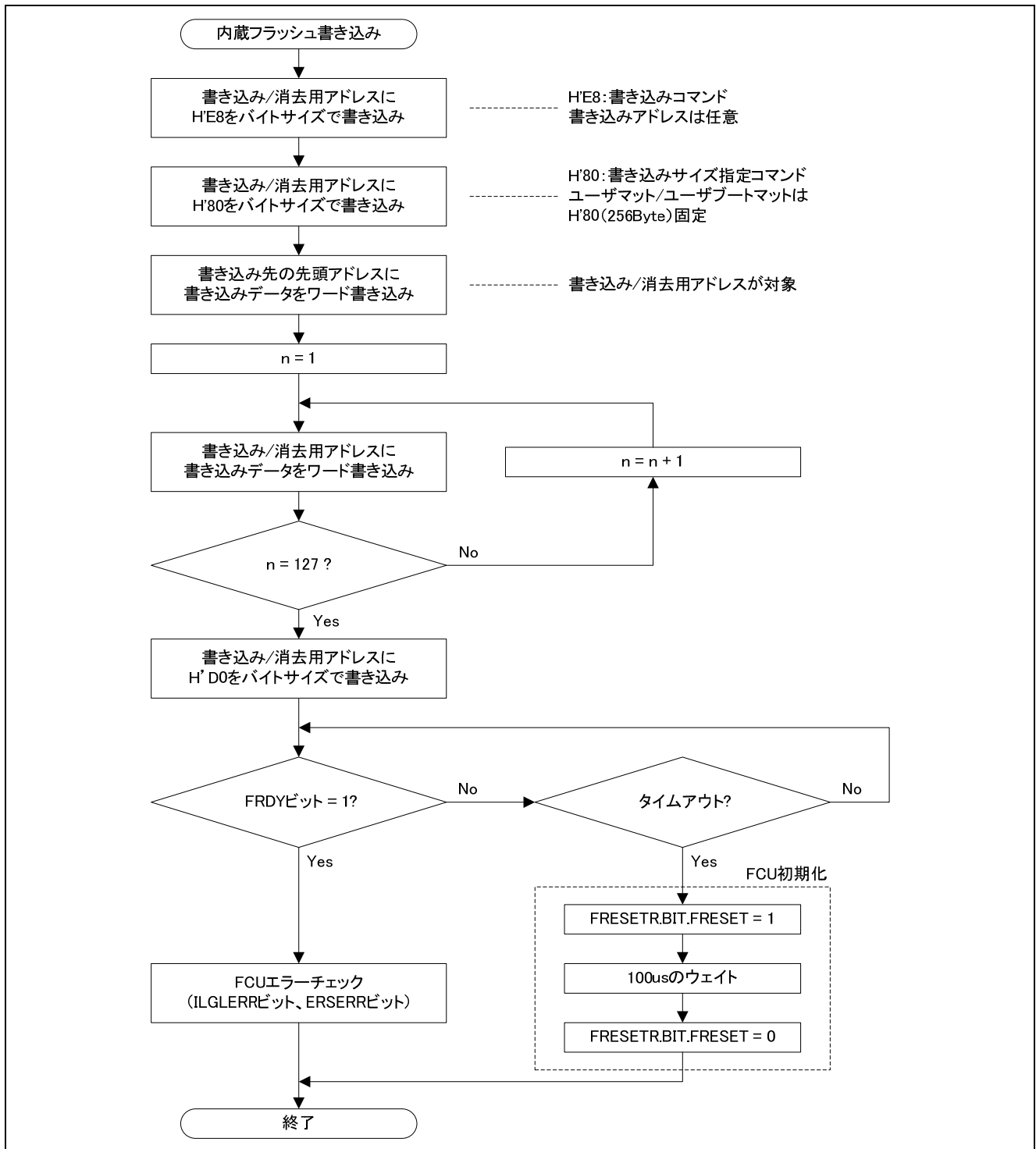


図7 内蔵フラッシュ書き込みフロー

2.3 内蔵フラッシュ書き換え用データバッファ

本応用例では、SH7216の内蔵RAM上に書き込みデータを保持しておくバッファエリアを確保します。バッファエリアの容量は、1回の内蔵フラッシュ書き込みに相当する256バイトとします。

図8に動作イメージを、表2にデータバッファエリアのアドレスを示します*。

【注】※ データバッファエリアはセクションを分けて管理しています。セクションの割付アドレスを変更することで、バッファエリアのアドレスを任意に設定可能です。

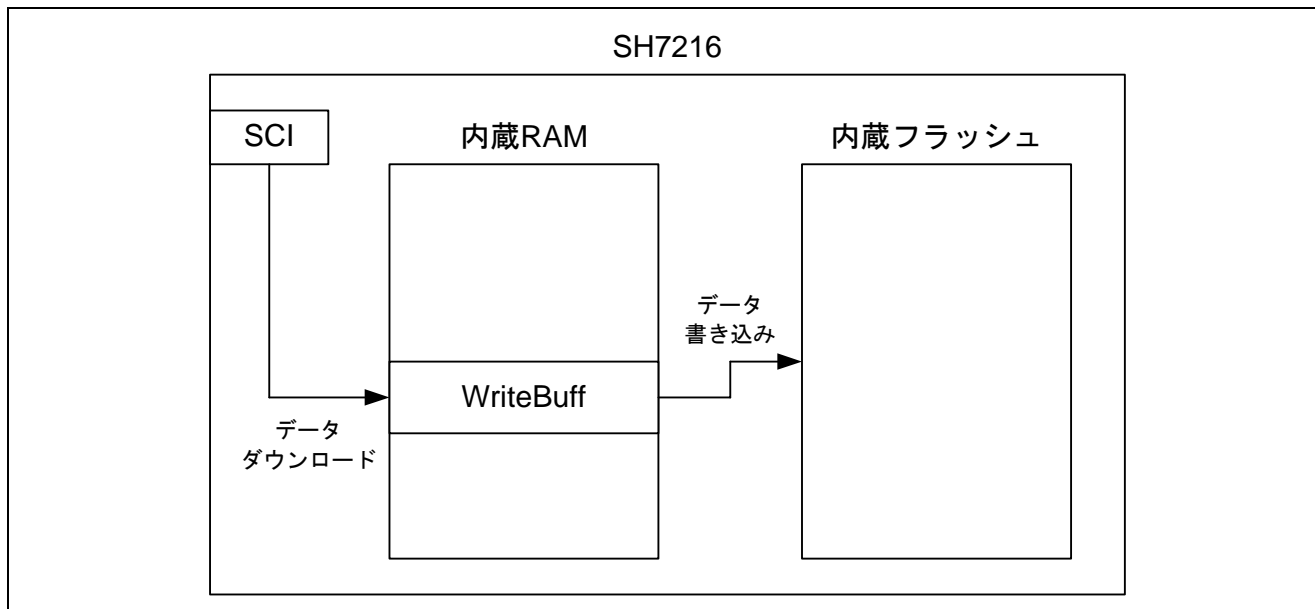


図8 バッファ動作イメージ

表2 データバッファエリア説明

バッファ名	アドレス	サイズ
WriteBuff	H'FFF8_1000~H'FFF8_10FF	256 バイト

3. 参考プログラムの外部仕様

本応用例では、main 関数を含むフラッシュ制御用のプログラム（以下、参考プログラム）をユーザマットの EB00 ブロック（アドレス番地：H'0000 0000 ～ H'0000 1FFF）内に配置しています。参考プログラムは、ユーザアプリケーション（main 関数）、シリアル通信制御プログラム、フラッシュ書き換え制御プログラム、および標準 API で構成されます。

図 9に参考プログラムの構成図を示します。

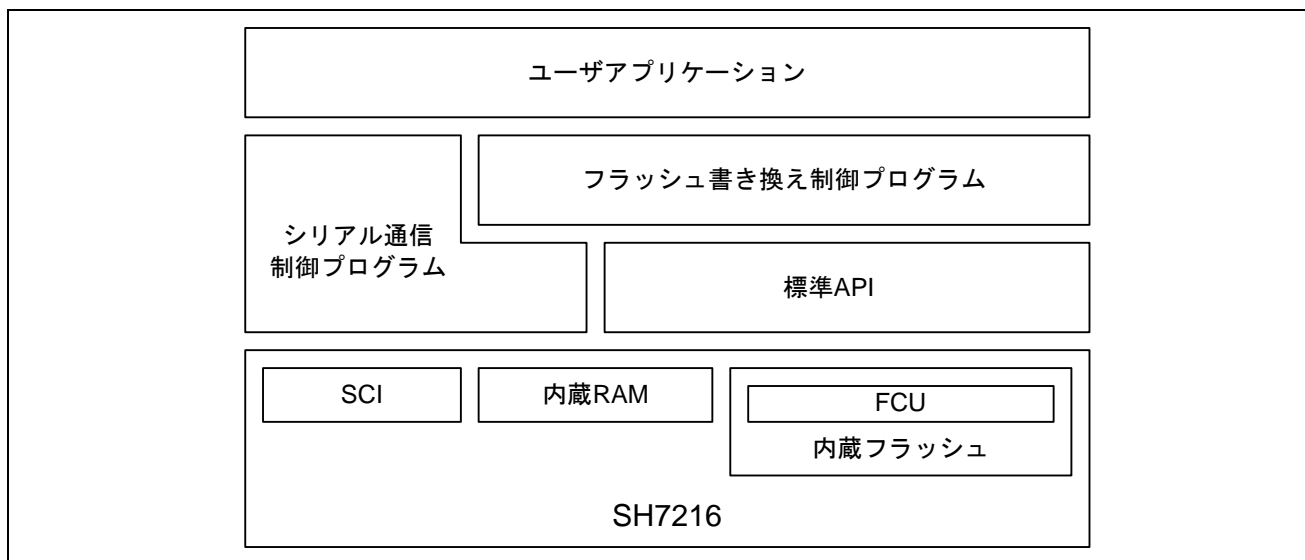


図9 参考プログラムの構成図

3.1 内蔵フラッシュ書き込み／消去の仕様

内蔵フラッシュの書き込み／消去対象領域は、参考プログラムを配置している EB00 ブロックを除くユーザマツト領域 (EB01~EB19 ブロックの書き込み／消去用アドレス番地:H'8080 2000 ~ H'808F FFFF) です。

参考プログラムは、ホスト PC からフラッシュ書き込み／消去処理コマンドを受信すると、上記フラッシュ書き込み／消去対象領域に対しブロック消去処理を実行し、ホスト PC から指定された書き込み先の先頭アドレスより、指定データサイズ分のフラッシュ書き込み処理を実行します。

図 10に参考プログラムの内蔵フラッシュ書き込み／消去処理イメージを示します。

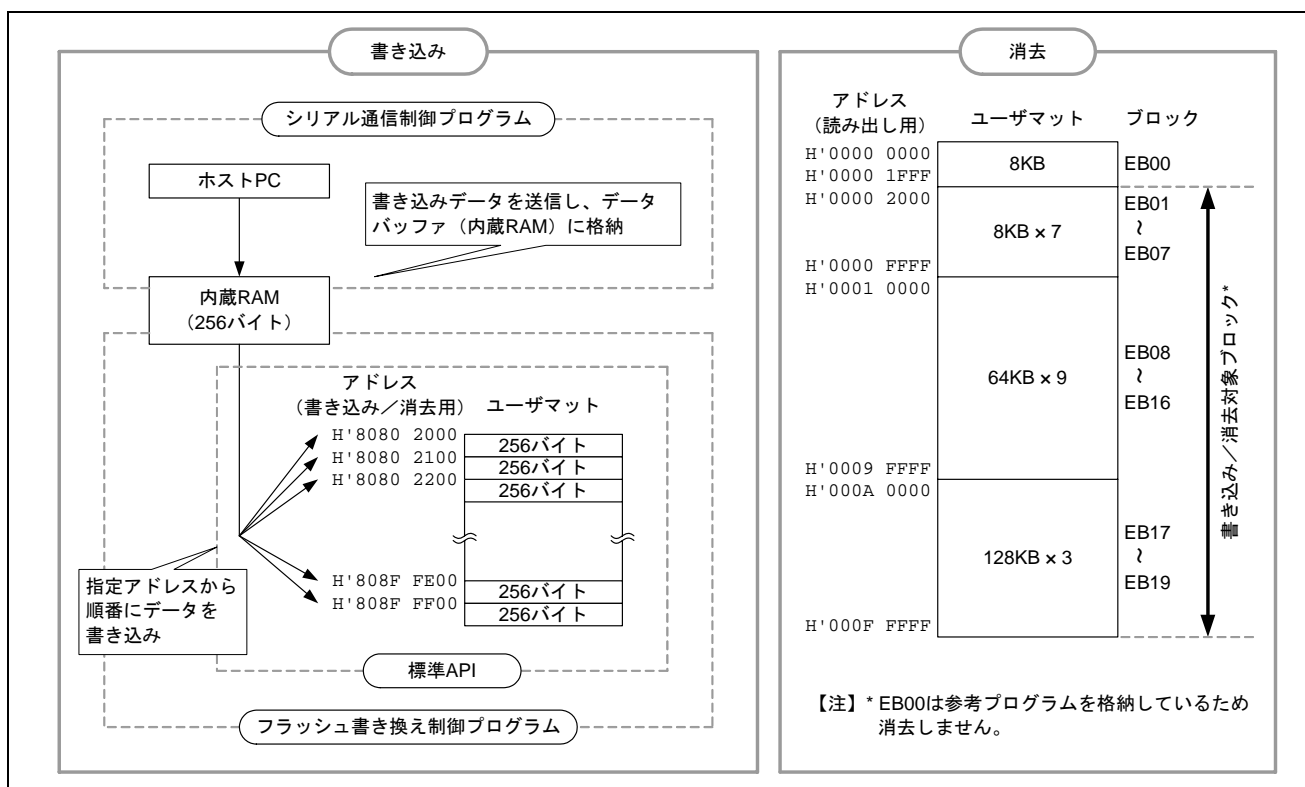


図10 内蔵フラッシュ書き込み／消去処理イメージ

3.2 参考プログラムの動作内容

本応用例では、ホスト PC*からシリアル通信によるコマンド制御により、内蔵フラッシュの書き込み/消去処理、および読み出し処理を行います。シリアル通信には、SH7216のSCIチャンネル1(SCI1)を使用します。また、参考プログラムはすべて内蔵RAM上で実行します。

【注】* ホスト PC 側のシリアル通信制御ツールに SerialDebugger (フリーソフトウェア) を使用して、本応用例の動作確認を行っています。

参考プログラムでは、フラッシュ書き込み/消去の許可状態を確認し、フラッシュ書き込み/消去が許可状態に設定されている場合、参考プログラムはホスト PC に対し通信用ユーザ制御コマンドの発行を要求します。フラッシュ書き込み/消去が禁止状態に設定されている場合、参考プログラムは書き込み/消去が許可状態になるまで FWE ビットをポーリングします。

図 11にメイン処理の概略フローを示します。

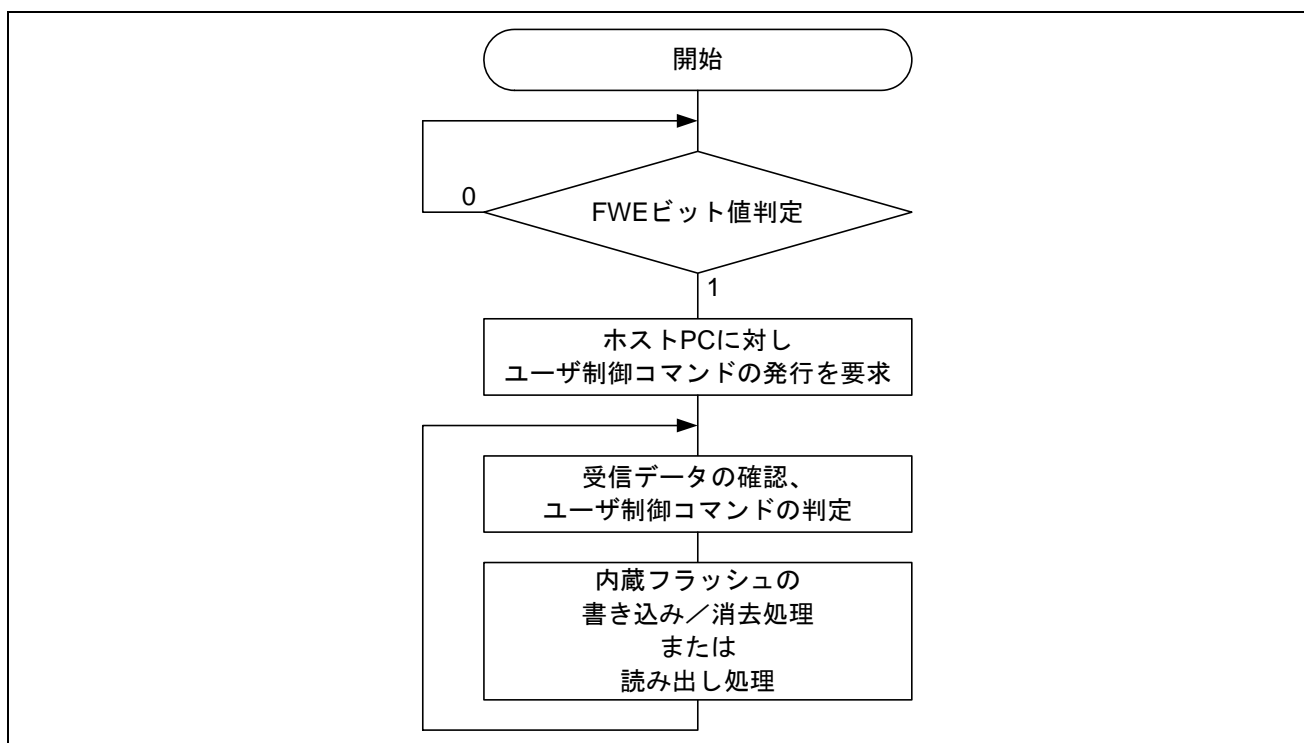


図11 メイン処理の概略フロー

表 3にホストPCからの通信用ユーザ制御コマンド一覧を、表 4にSH7216からの通知一覧を示します。

フラッシュ書き込み／消去処理中に異常が発生した場合、参考プログラムの処理として、ホスト PC に対し異常終了 (RET_NG) を通知し、無限ループとしています。必要に応じてエラー処理を追加してください。

表3 ホスト PC からの通信用ユーザ制御コマンド一覧 (ホスト PC → SH7216)

通信コマンド名	コマンド値	機能
フラッシュ書き込み／消去処理 (CMD_WRITE)	H'50	ブロック消去 (EB01~EB19) を行った後、指定アドレスから指定バイト数分データを書き込み
フラッシュ読み出し処理 (CMD_READ)	H'52	内蔵フラッシュの指定アドレスから指定バイト数分のデータを読み出し

表4 SH7216からの通知一覧 (SH7216 → ホスト PC)

通知名	応答値	機能
正常終了 (RET_OK)	H'00	コマンド処理が正常に終了したことをホスト PC に通知
異常終了 (RET_NG)	H'01	コマンド処理中に異常が発生したことをホスト PC に通知
送信要求 (RET_REQ)	H'11	ユーザ制御コマンドまたは書き込みデータの送信要求状態であることをホスト PC に通知

3.2.1 内蔵フラッシュの書き込み/消去

フラッシュ書き込み/消去処理コマンド (CMD_WRITE) により、参考プログラムはブロック消去と書き込み先の先頭アドレスから指定サイズのデータ書き込みを行います。

参考プログラムは、ホスト PC からフラッシュ書き込み/消去処理コマンド (CMD_WRITE) を受信すると、EB01~EB19 に対しブロック消去処理を実行します。ブロック消去処理終了後、参考プログラムはホスト PC に対し送信要求 (RET_REQ) を通知します。次に、ホスト PC から書き込み先の先頭アドレス (4 バイト形式) と書き込みデータサイズ (4 バイト形式) の計 8 バイトを受信すると、参考プログラムはホスト PC に対し書き込みデータの送信要求 (RET_REQ) を通知し、書き込み処理に移行します。

書き込み先の先頭アドレスには、EB01~EB19 ブロック内の読み出し用アドレス (H'0000 2000 ~ H'000F FFFF) *を、256 バイト境界で指定してください。EB00 ブロック内のアドレス (H'0000 0000 ~ H'0000 1FFF) が指定された場合、参考プログラムはホスト PC に対し異常終了 (RET_NG) を通知し無限ループとなります。また、ユーザマツト領域以外のアドレスが指定された場合のエラーチェックは行っていませんので、範囲外のアドレスは指定しないでください。

【注】 * 標準 API の仕様上、書き込み先の先頭アドレスには読み出し用アドレスを使用してください。標準 API 内部で書き込み/消去用アドレスに変換されます。

書き込み処理に移行すると、参考プログラムは内蔵フラッシュへの書き込み単位である 256 バイトのデータ受信ごとに、書き込みデータの送信要求 (RET_REQ) を通知します。ホスト PC はこの送信要求 (RET_REQ) ごとに 256 バイトのデータを送信してください。

参考プログラムは、256 バイトのデータ受信ごとにフラッシュ書き込み処理を実行します。なお、最終データのデータサイズが 256 バイトに満たない場合、参考プログラムは書き込み単位が 256 バイトになるようにデータを付加してフラッシュ書き込み処理を実行します。付加するデータの値は H'FF です。

フラッシュ書き込み処理により書き込まれたデータサイズが、指定された書き込みデータサイズに達すると、参考プログラムはホスト PC に対し正常終了 (RET_OK) を通知します。

図 12 に参考プログラムにおけるフラッシュ書き込み/消去処理の通信コマンド制御シーケンスを、また 図 13 にフラッシュ書き込み/消去処理の通信コマンド制御例 (SerialDebugger 使用) を示します。

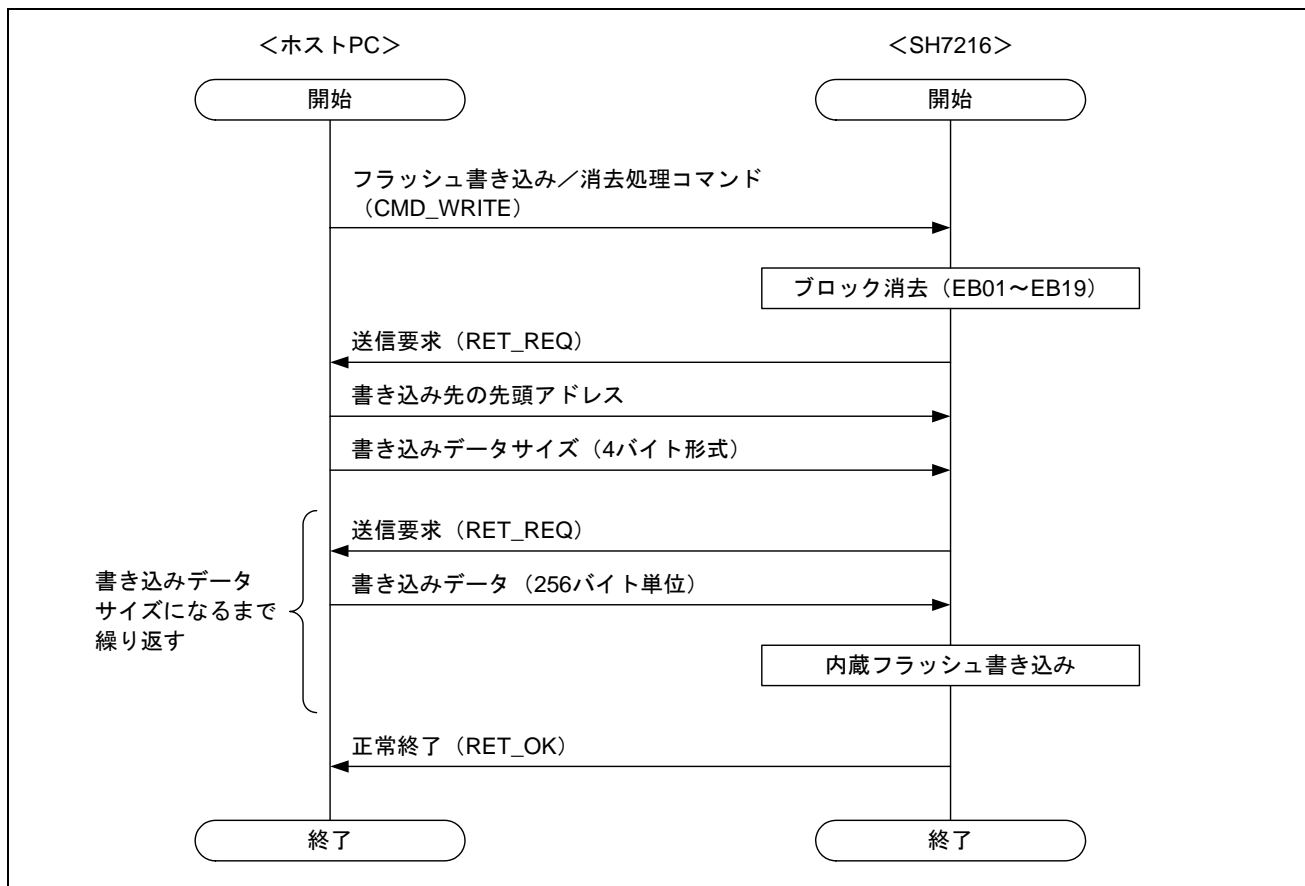


図12 フラッシュ書き込み/消去処理の通信コマンド制御シーケンス

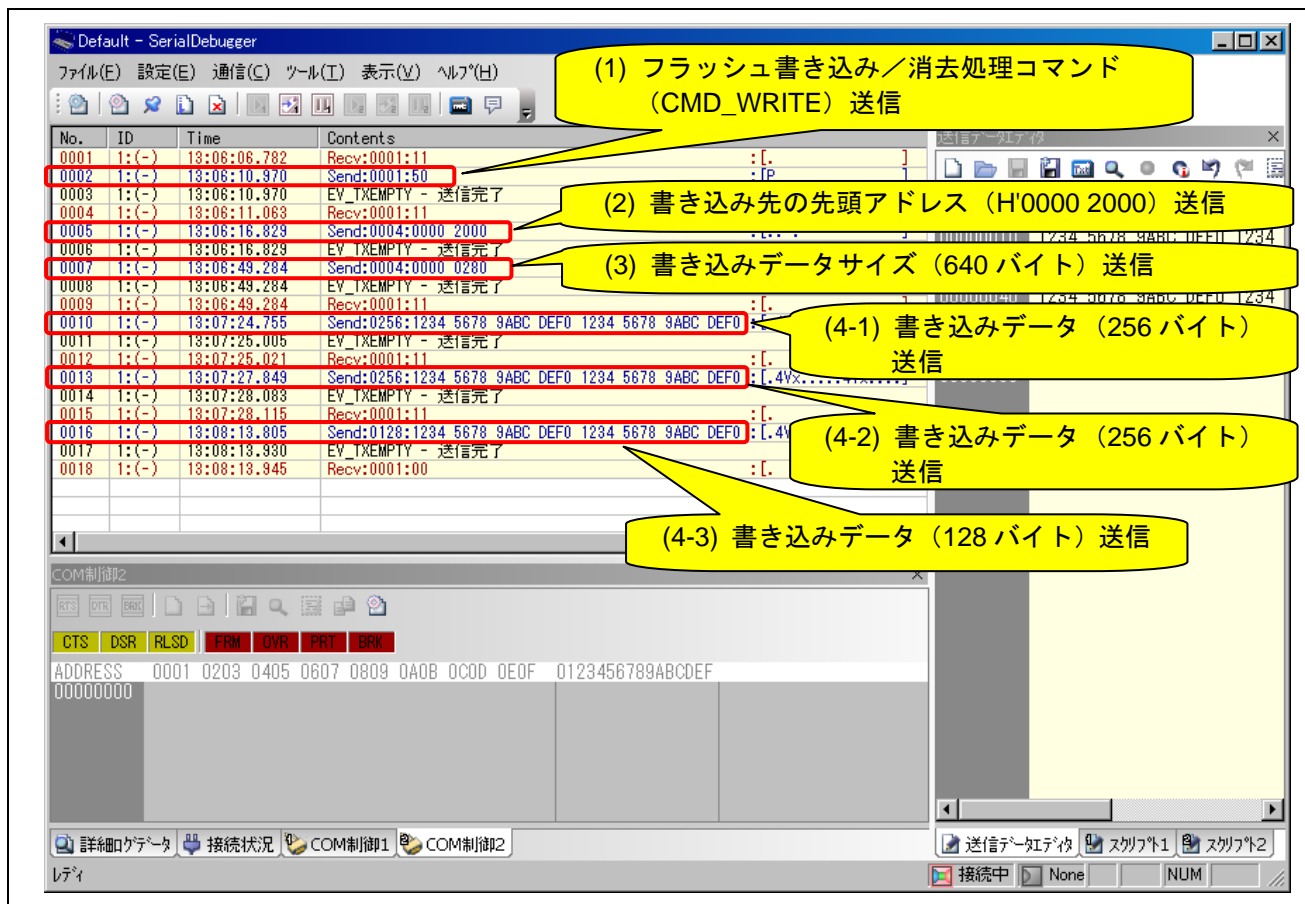


図13 フラッシュ書き込み/消去処理の通信コマンド制御例 (SerialDebugger 使用)

3.2.2 内蔵フラッシュの読み出し

フラッシュ読み出し処理コマンド (CMD_READ) により、参考プログラムは読み出し先の先頭アドレスから指定サイズのデータを読み出してホスト PC へ送信します。

参考プログラムは、フラッシュ読み出し処理コマンド (CMD_READ) を受信すると、ホスト PC に対し送信要求 (RET_REQ) を通知します。次に、ホスト PC から読み出し先の先頭アドレス (4 バイト形式) と読み出しデータサイズ (4 バイト形式) の計 8 バイトを受信すると、読み出し先のアドレスから指定サイズのデータを読み出してホスト PC へ送信します。

読み出し先の先頭アドレスには EB00~EB19 ブロック (ユーザマット) 内の読み出し用アドレス (H'0000 0000 ~ H'000F FFFF) を指定してください。ユーザマット以外のアドレスが指定された場合、参考プログラムは読み出し処理を行わず、ホスト PC に対し異常終了 (RET_NG) を通知し無限ループとなります。また、ユーザマット以外のアドレスが指定された場合のエラーチェックは行っていませんので、範囲外のアドレスは指定しないでください。

図 14 に参考プログラムにおけるフラッシュ読み出し処理の通信コマンド制御シーケンスを、また 図 15 にフラッシュ読み出し処理の通信コマンド制御例 (SerialDebugger 使用) を示します。

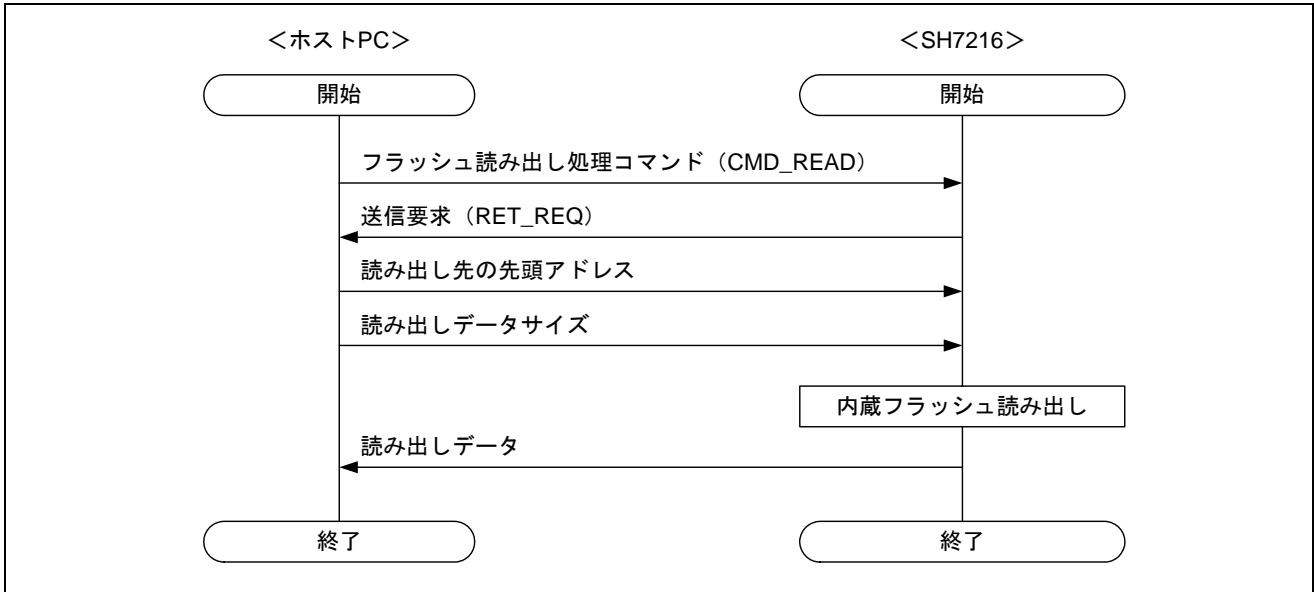


図14 フラッシュ読み出し処理の通信コマンド制御シーケンス

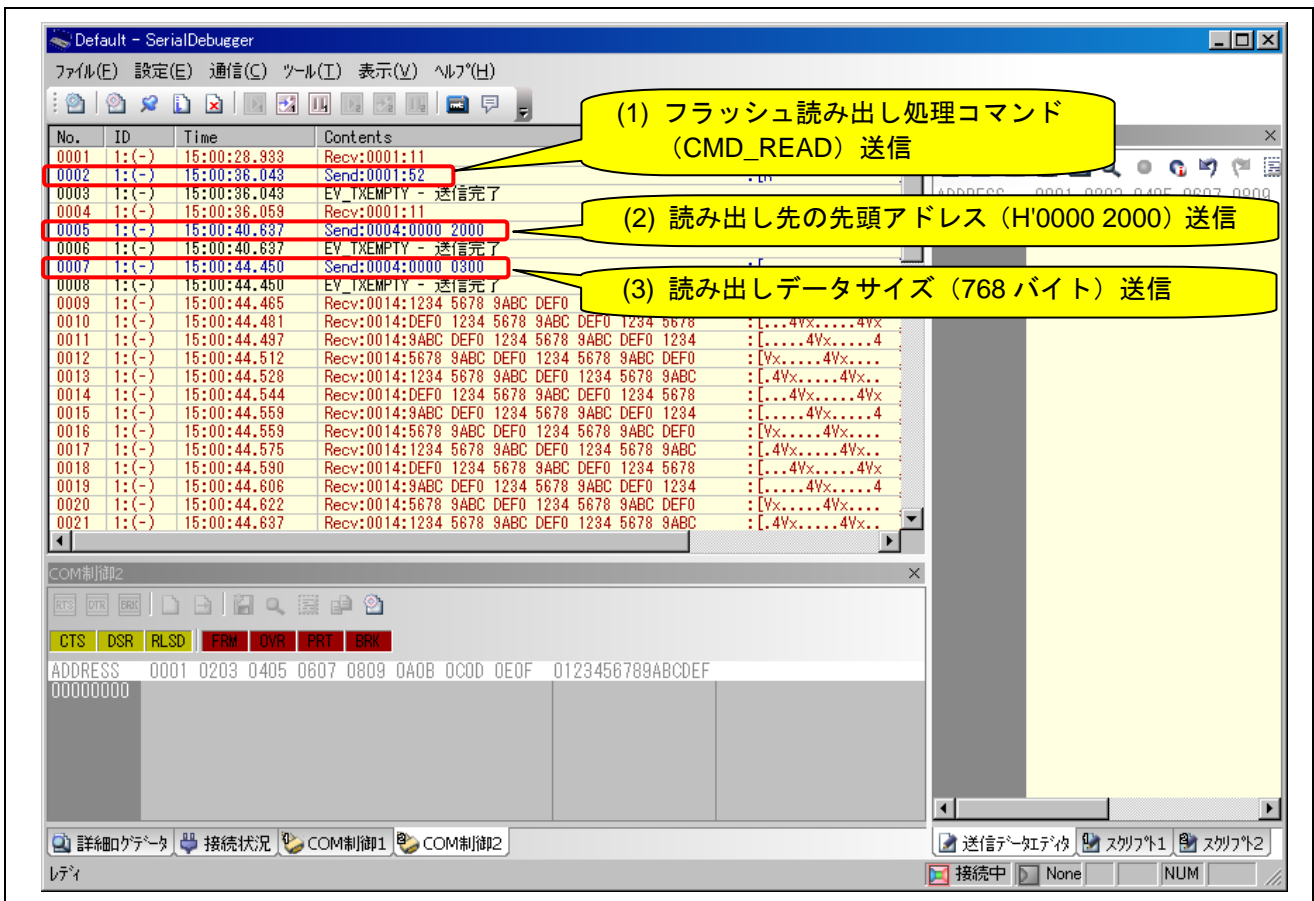


図15 フラッシュ読み出し処理の通信コマンド制御例 (SerialDebugger 使用)

4. 参考プログラムの内部仕様

4.1 モジュール一覧

表 5に参考プログラムのモジュール一覧を示します。

表5 参考プログラムのモジュール一覧

分類	モジュール名	関数名	機能	処理フロー
ユーザアプリケーション	メイン処理	main	ユーザアプリケーションプログラムの実行	図 16参照
フラッシュ書き換え制御	フラッシュ書き込み／消去処理	ocf_write	フラッシュ書き込み／消去処理の実行	図 17, 図 18参照
	フラッシュ読み出し処理	ocf_read	フラッシュ読み出し処理の実行	図 19参照
	フラッシュ書き込み／消去許可状態チェック処理	ocf_pe_chk	フラッシュ書き込み／消去許可状態の確認	図 20参照
シリアル通信制御	SCI 初期化処理	io_sci_init	SCI (チャンネル 1) の初期設定	—
	SCI 受信データ有無チェック処理	io_sci_chk_rcv	受信データが SCRDR レジスタに格納されているか確認	—
	SCI 送信処理	io_sci_snd	1 バイトデータ送信	—
	SCI 受信処理	io_sci_rcv	指定されたバイト数分のデータ受信	—
標準 API	ブロック消去	R_FlashErase	指定ブロック内のデータ消去	—
	フラッシュ書き込み	R_FlashWrite	指定アドレス領域へのデータ書き込み	—

4.2 使用変数

表 6に参考プログラムで使用する変数を示します。

表6 参考プログラムで使用する変数

変数ラベル名	機能	使用モジュール
unsigned char WriteBuff[256]	書き込みデータ格納	ocf_write

4.3 レジスタの設定

表 7に参考プログラムで使用する周辺機能のレジスタ設定内容を示します。

表7 参考プログラムで使用する SCI (チャンネル 1) のレジスタ設定内容

レジスタ	アドレス	設定値	設定内容
シリアルモード レジスタ_1 (SCSMR_1)	H'FFFF 8800	H'00	<ul style="list-style-type: none"> • C/A# = "0" : 調歩同期式モード • CHR = "0" : 8ビットデータ • PE = "0" : パリティビットの付加、およびチェックを禁止 • STOP = "0" : 1ストップビット • MP = "0" : マルチプロセッサモードを禁止 • CKS[1:0] = "B'00" : Pφクロック
ビットレートレジスタ_1 (SCBRR_1)	H'FFFF 8802	D'162	ビットレート=9600 bps (Pφ=50 MHz)
シリアルコントロール レジスタ_1 (SCSCR_1)	H'FFFF 8804	H'30	<ul style="list-style-type: none"> • TE = "1" : 送信動作を許可 • RE = "1" : 受信動作を許可
ポート A コントロール レジスタ L2 (PACRL2)	H'FFFE 3814	H'0006	• PA4MD[2:0] = "B'110" : TXD1 出力 (SCI)
ポート A コントロール レジスタ L1 (PACRL1)	H'FFFE 3816	H'6000	• PA3MD[2:0] = "B'110" : RXD1 入力 (SCI)

4.4 フローチャート

本節では参考プログラムの各モジュール処理フローを示します。

4.4.1 メイン処理フロー

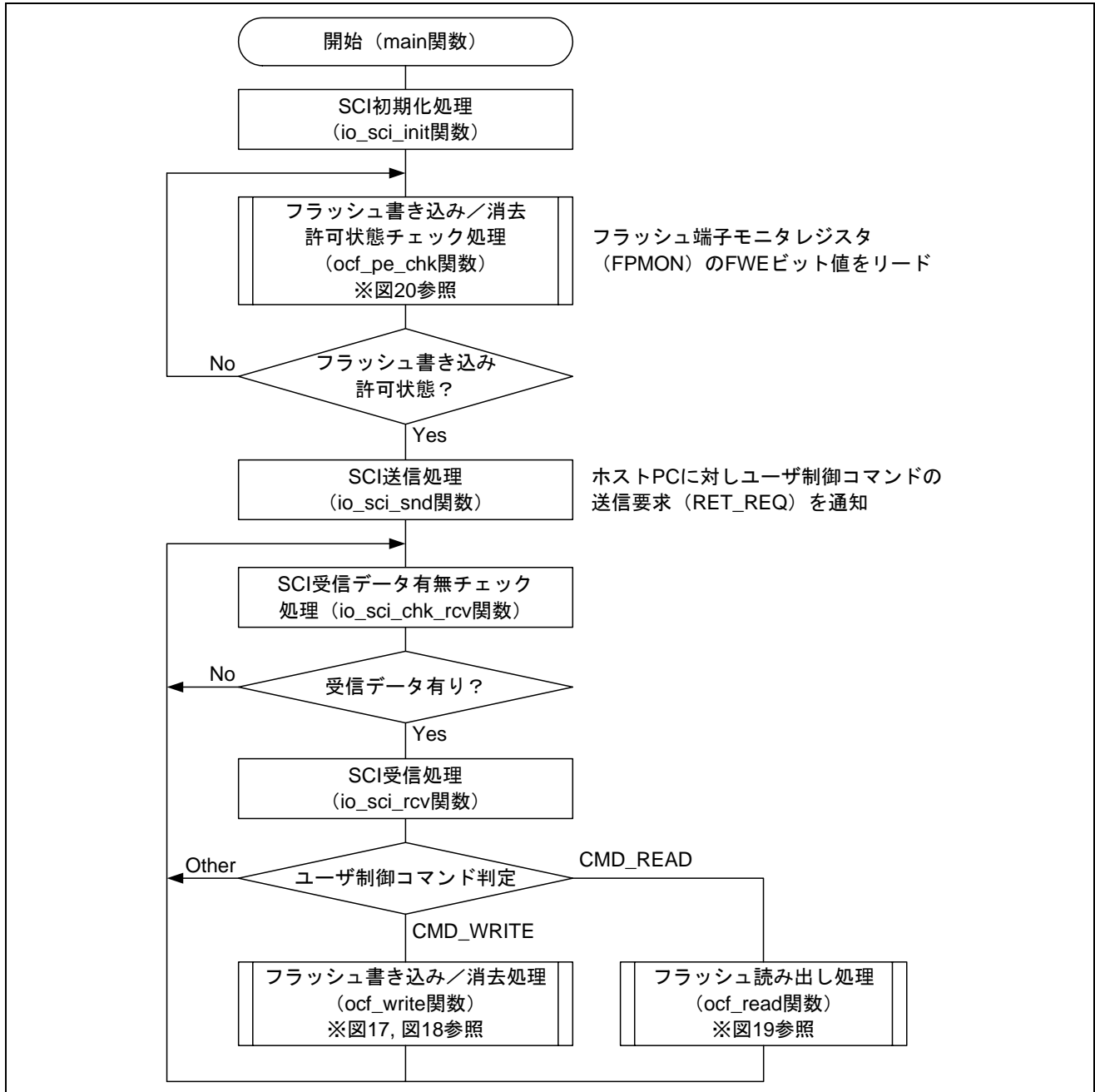


図16 メイン処理フロー

4.4.2 フラッシュ書き込み／消去処理フロー

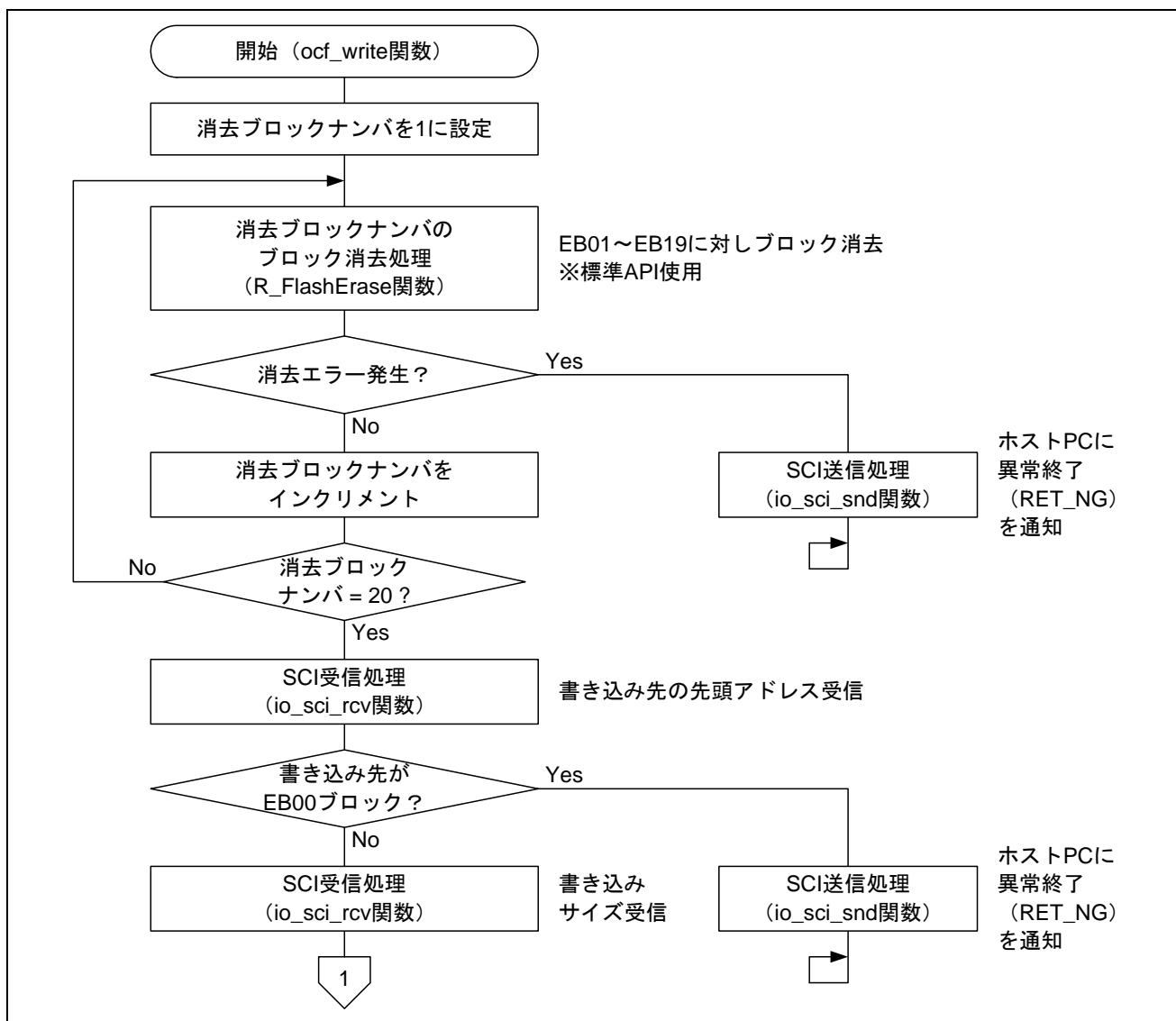


図17 フラッシュ書き込み／消去処理フロー (1)

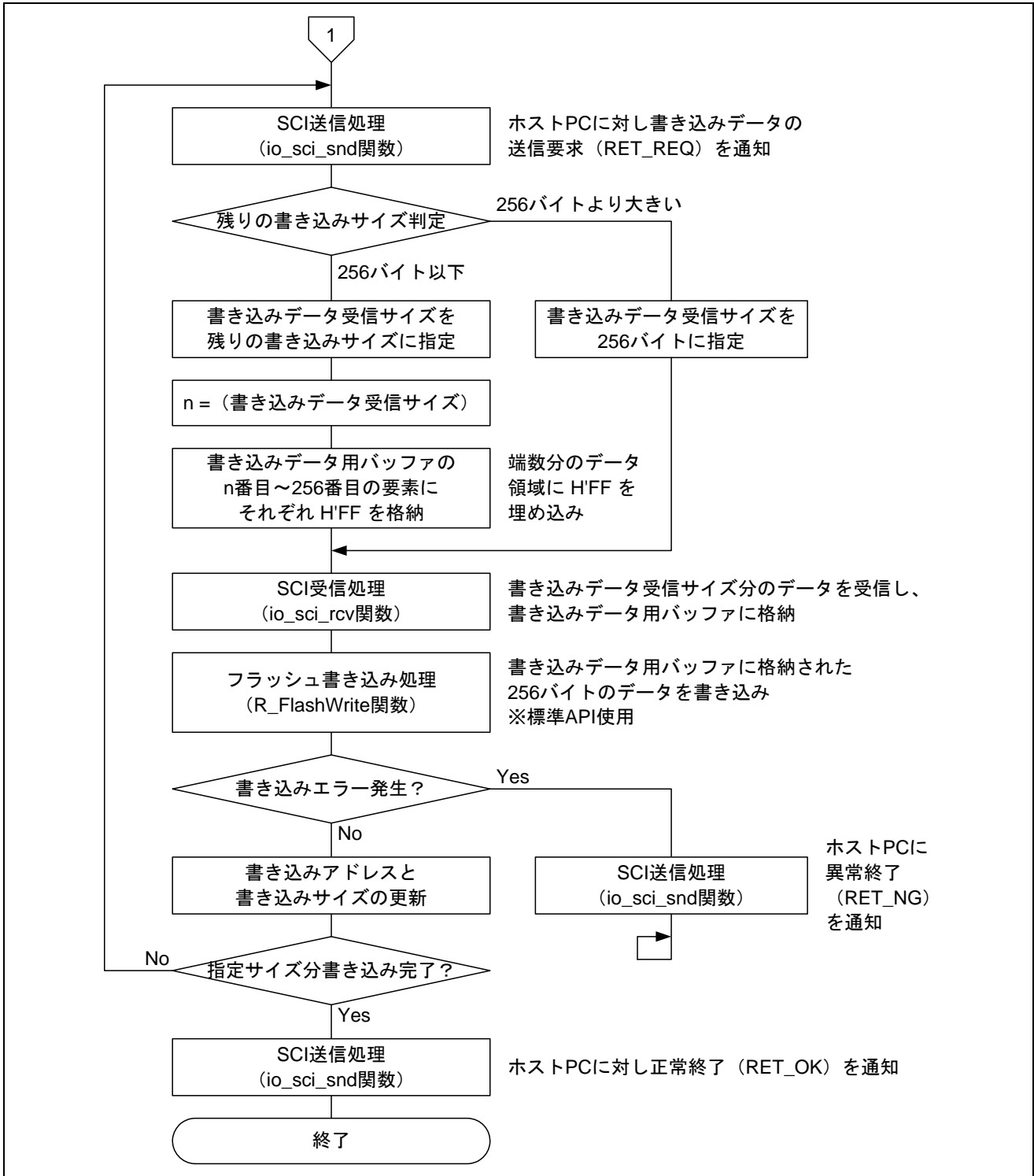


図18 フラッシュ書き込み/消去処理フロー (2)

4.4.3 フラッシュ読み出し処理フロー

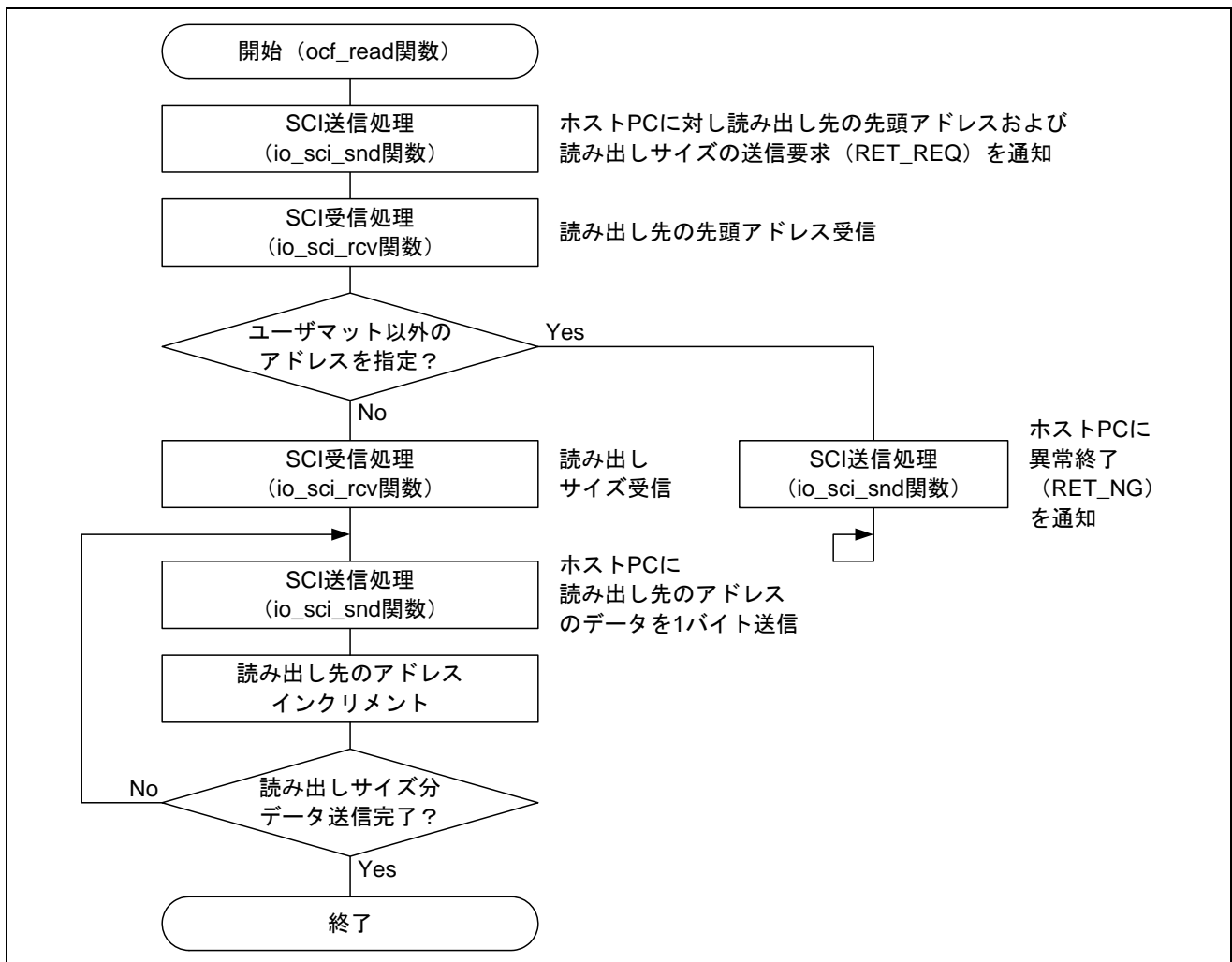


図19 フラッシュ読み出し処理フロー

4.4.4 フラッシュ書き込み／消去許可状態チェック処理フロー

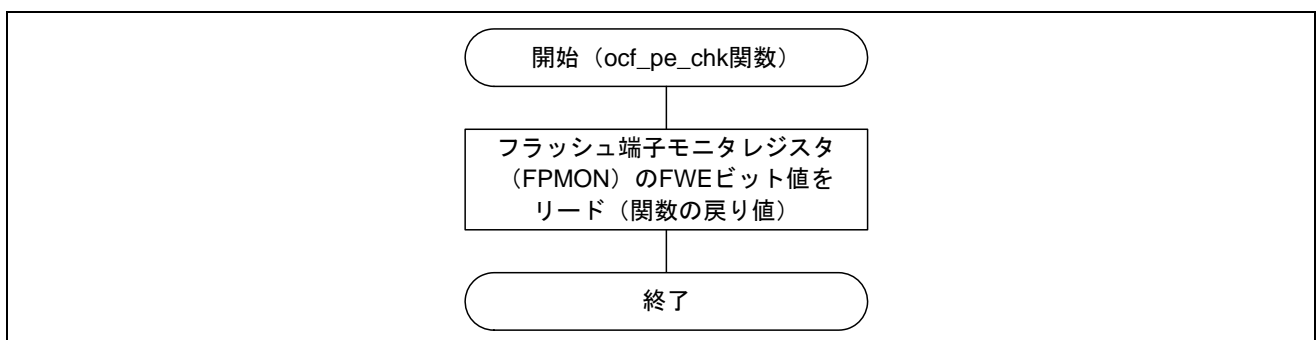


図20 フラッシュ書き込み／消去許可状態チェック処理フロー

5. 参考プログラムリスト

5.1 サンプルプログラムリスト "main.c" (1)

```
1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corp. and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corp. and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 *   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 /*"FILE COMMENT"***** Technical reference data *****
31 *   System Name : SH7216 Sample Program
32 *   File Name   : main.c
33 *   Abstract    : Example of use for the Flash user-program mode
34 *   Version     : 1.00.00
35 *   Device      : SH7216
36 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37 *                : C/C++ compiler package for the SuperH RISC engine family
38 *                :                (Ver.9.03 Release00).
39 *   OS          : None
40 *   H/W Platform: R0K572167 (CPU board)
41 *   Description :
42 *****/
43 *   History     : Aug.20,2010 Ver.1.00.00
44 *"FILE COMMENT END"*****
45 #include "iodefine.h"
46 #include "Flash_API_SH7216.h"
47
```

5.2 サンプルプログラムリスト "main.c" (2)

```

48  /* ==== Macro definition ==== */
49  #define FLASH_PE_ENABLE    1      /* Flash program/erase enabled    */
50  #define FLASH_PE_DISABLE  0      /* Flash program/erase disabled  */
51  #define CMD_WRITE         0x50    /* フラッシュ消去/書き込み処理コマンド */
52  #define CMD_READ          0x52    /* フラッシュ読み出し処理コマンド */
53  #define RET_OK            0x00    /* 正常終了 */
54  #define RET_NG            0x01    /* 異常終了 */
55  #define RET_REQ           0x11    /* 送信要求 */
56
57  /* ==== Prototype declaration ==== */
58  void main(void);
59  int ocf_pe_chk(void);
60  void ocf_write(void);
61  void ocf_read(void);
62  /* ---- External reference ---- */
63  extern void io_sci_init(void);
64  extern int io_sci_chk_rcv(void);
65  extern void io_sci_snd(unsigned char data);
66  extern void io_sci_rcv(unsigned char *data, unsigned long num);
67
68  /* ==== Global variable ==== */
69  #pragma section WriteDATA /* 書き込みデータバッファエリア */
70  unsigned char WriteBuff[PROGRAM_SIZE_ROM];
71
72
73  #pragma section FRAM
74  /*"FUNC COMMENT"*****
75  * ID          :
76  * Outline     : サンプルプログラムメイン
77  *-----
78  * Include     : "iodefine.h" and "flash.h"
79  *-----
80  * Declaration : void main(void);
81  *-----
82  * Function    :
83  *-----
84  * Argument    : void
85  *-----
86  * Return Value : void
87  *-----
88  * Note        : None
89  *"FUNC COMMENT END"*****/
90  void main(void)
91  {
92      unsigned char RcvData;
93      int pe_ok;
94
95      /* ==== SCI 初期化 ==== */
96      io_sci_init();
97

```

5.3 サンプルプログラムリスト "main.c" (3)

```

98     /* ==== フラッシュ書き込み/消去許可チェック ==== */
99     do{
100         pe_ok = ocf_pe_chk();      /* FWE pin = High ? */
101     }while(pe_ok != FLASH_PE_ENABLE);
102
103     /* ==== ホスト PC に対し送信要求を通知 ==== */
104     io_sci_snd(RET_REQ);
105
106     /* ==== フラッシュ書き込み/消去処理 or フラッシュ読み出し処理 ==== */
107     while(1){
108         /* ---- ユーザ制御コマンドチェック ---- */
109         if(io_sci_chk_rcv() != 0){
110             io_sci_rcv(&RcvData, 1);
111             if(RcvData == CMD_WRITE){
112                 ocf_write();      /* フラッシュ消去/書き込み処理 */
113             }
114             else if(RcvData == CMD_READ){
115                 ocf_read();      /* フラッシュ読み出し処理 */
116             }
117         }
118     }
119 }
120
121 /*"FUNC COMMENT"*****
122 * ID          :
123 * Outline    : フラッシュ ROM の書き込み/消去許可状態チェック
124 *-----
125 * Include    : "iodefine.h"
126 *-----
127 * Declaration : int ocf_pe_chk(void);
128 *-----
129 * Description : フラッシュ端子モニタレジスタ(FPMON)の FWE ビット値をリードし、
130 *              : 戻り値として返却します。
131 *-----
132 * Argument   : void
133 *-----
134 * Return Value : ・0 ; フラッシュ ROM 書き込み/消去禁止状態
135 *              : ・1 ; フラッシュ ROM 書き込み/消去許可状態
136 *-----
137 * Note       : None
138 *"FUNC COMMENT END"*****/
139 int ocf_pe_chk(void)
140 {
141     return FLD.FPMON.BIT.FWE;
142 }
143

```

5.4 サンプルプログラムリスト "main.c" (4)

```
144 /*"FUNC COMMENT"*****
145 * ID      :
146 * Outline : フラッシュ書き込み/消去処理
147 *-----
148 * Include : "iodefine.h" and "flash.h"
149 *-----
150 * Declaration : void ocf_write(void);
151 *-----
152 * Description : EB01~EB19 に対しブロック消去を行った後、ホスト PC より指定された
153 *              : 書き込み先の先頭アドレスから指定サイズ分のデータをフラッシュに
154 *              : 書き込みます。
155 *-----
156 * Argument : void
157 *-----
158 * Return Value : void
159 *-----
160 * Note      : None
161 *"FUNC COMMENT END"*****/
162 void ocf_write(void)
163 {
164     unsigned char error;          /* Function return value */
165     unsigned char RcvData;        /* Receive data */
166     unsigned char EraseBlkNum;    /* Erase block number */
167     unsigned long i;             /* Loop counter */
168     unsigned long WriteAddr;     /* Start address to be programmed */
169     unsigned long WriteSize;     /* Data size to be programmed */
170     unsigned long RcvSize;       /* Receiving size for data to be programmed */
171
172     /* ==== ブロック消去処理 ==== */
173     for(EraseBlkNum = BLOCK_1; EraseBlkNum <= BLOCK_19; EraseBlkNum++){
174         /* EB01~EB19 の消去 */
175         error = R_FlashErase((uint8_t)EraseBlkNum);
176         if(error != RET_OK){      /* 消去エラー? */
177             io_sci_snd(RET_NG);  /* エラー送信 */
178             while(1){
179             }
180         }
181     }
182
183     /* ==== 送信要求 ==== */
184     io_sci_snd(RET_REQ);
185
186     /* ==== 書き込み先の先頭アドレス受信 ==== */
187     io_sci_rcv((unsigned char *)&WriteAddr, 4);
188     if( (WriteAddr >= 0x00000000) && (WriteAddr < 0x00002000) ){
189         /* 書き込み先に EB00 を指定? */
190         io_sci_snd(RET_NG);      /* エラー送信 */
191         while(1){
192         }
193     }
194 }
```

5.5 サンプルプログラムリスト "main.c" (5)

```
195     /* ==== 書き込みデータサイズ受信 ==== */
196     io_sci_rcv((unsigned char *)&WriteSize, 4);
197
198     /* ==== 書き込み処理 ==== */
199     while(WriteSize > 0){
200         io_sci_snd(RET_REQ);          /* 送信要求 */
201
202         if(WriteSize > PROGRAM_SIZE_ROM){
203             RcvSize = PROGRAM_SIZE_ROM;
204         }
205         else{
206             RcvSize = WriteSize;
207             for(i = RcvSize; i < PROGRAM_SIZE_ROM; i++){
208                 WriteBuff[i] = 0xff;
209             }
210         }
211
212         /* ---- 書き込みデータ受信 ---- */
213         io_sci_rcv(WriteBuff, RcvSize);
214
215         /* ---- 書き込み ---- */
216         error = R_FlashWrite((uint32_t)WriteAddr, (uint32_t)WriteBuff, PROGRAM_SIZE_ROM);
217         if(error != 0){                /* 書き込みエラー? */
218             io_sci_snd(RET_NG);        /* エラー送信 */
219             while(1){
220                 }
221         }
222
223         WriteAddr += PROGRAM_SIZE_ROM;
224         WriteSize -= RcvSize;
225     }
226
227     io_sci_snd(RET_OK);                /* 書き込み処理(正常)終了 */
228 }
229
```

5.6 サンプルプログラムリスト "main.c" (6)

```
230  /*"FUNC COMMENT"*****  
231  * ID      :  
232  * Outline : フラッシュ読み出し処理  
233  *-----  
234  * Include : "flash.h"  
235  *-----  
236  * Declaration : void ocf_read(void);  
237  *-----  
238  * Description : 読み出し先の先頭アドレスから指定サイズのデータを読み出して  
239  *              : ホスト PC へ送信します。  
240  *-----  
241  * Argument  : void  
242  *-----  
243  * Return Value : void  
244  *-----  
245  * Note      : None  
246  /*"FUNC COMMENT END"*****/  
247 void ocf_read(void)  
248 {  
249     unsigned char *ReadData; /* Pointer for readout data */  
250     unsigned long ReadAddr; /* Start address to be read */  
251     unsigned long ReadSize; /* Reading size */  
252     unsigned long i; /* Loop counter */  
253  
254     /* ==== 送信要求 ==== */  
255     io_sci_snd(RET_REQ);  
256  
257     /* ==== 読み出し開始アドレス受信 ==== */  
258     io_sci_rcv((unsigned char *)&ReadAddr, 4);  
259     if(ReadAddr >= 0x00100000){  
260         /* 読み出し用アドレス以外のアドレスを指定? */  
261         io_sci_snd(RET_NG); /* エラー送信 */  
262         while(1){  
263             }  
264     }  
265  
266     /* ==== 読み出しデータサイズ受信 ==== */  
267     io_sci_rcv((unsigned char *)&ReadSize, 4);  
268  
269     /* ==== ROMから読み出したデータを送信 ==== */  
270     ReadData = (unsigned char *)ReadAddr;  
271     for(i = 0; i < ReadSize; i++){  
272         io_sci_snd(*ReadData++);  
273     }  
274 }  
275  
276 /* End of File */
```

6. 参考ドキュメント

- ハードウェアマニュアル
SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編 Rev.3.00
(最新版をルネサス エレクトロニクス ホームページから入手してください。)
- ソフトウェアマニュアル
SH-2A、SH2A-FPU ユーザーズマニュアル ソフトウェア編 Rev.4.00
(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/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.08.20	—	初版発行
1.01	2012.06.15	—	サンプルコード（シンプルフラッシュ API）更新

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/>