
SH7239グループ

R01AN0057JJ0100

Rev.1.00

2010.11.22

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

要旨

本アプリケーションノートは、SH7239 のルネサスシリアルペリフェラルインタフェース（RSPI）を使用したシリアルフラッシュメモリの接続例について説明します。

動作確認デバイス

SH7239

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムリスト.....	14
4. 参考ドキュメント.....	32

1. はじめに

1.1 仕様

- 1M ビット (256K ビット×4 セクタ)のシリアルフラッシュメモリを SH7239 と接続します。

1.2 使用機能

- ルネサスシリアルペリフェラルインタフェース (RSPI)
- 汎用入出力ポート

1.3 適用条件

マイコン	SH7239 (R5F72395ADFP)
電源電圧	3.3V
動作周波数	内部クロック : 160 MHz バスクロック : 40 MHz 周辺クロック : 40 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 02
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)

2. 応用例の説明

本応用例では、SH7239（マスタ）と SPI 互換のシリアルフラッシュメモリ（スレーブ）を接続して、ルネサスシリアルペリフェラルインタフェース（RSPI）を使用したリード/ライトアクセスを行います。この章では、端子接続例と参考プログラムフローを説明します。

2.1 使用機能の動作概要

SH7239 の RSPI は、MOSI（Master Out Slave In）端子および MISO（Master In Slave Out）端子、SSL（Slave Select）端子、RSPCK（SPI Clock）端子を使用して、SPI 動作で全二重同期式のシリアル通信が可能です。RSPI は、マスタ/スレーブの選択、シリアル転送クロックの極性と位相の変更（SPI モード変更）、転送ビット長の変更（8～16、20、24、32 ビット）が可能のため、多様な SPI 互換デバイスを接続することができます。

2.2 シリアルフラッシュメモリの端子接続例

表 1 に本応用例で使用する SPI 互換シリアルフラッシュメモリ（ST マイクロ社製 M25P10）の仕様を示します。

表1 本応用例で使用するシリアルフラッシュメモリの仕様

項目	仕様
SPI モード	SPI モード 0 およびモード 3 に対応可能
クロック周波数	20MHz(max)
容量	1M ビット
セクタサイズ	256K ビット
ページサイズ	128 バイト
イレースサイズ	Sector Erase (256 K ビット)/Bulk Erase (1M ビット)
プログラムサイズ	Page Program (1~128 バイト)
プロテクト単位	none/Upper quarter (セクタ 3)/Upper half (セクタ 3,2)/Whole memory (全セクタ)

図 1 にシリアルフラッシュメモリ接続回路例を示します。SH7239 の端子機能については、表 2 のマルチプレクス出力端子に従い設定してください。

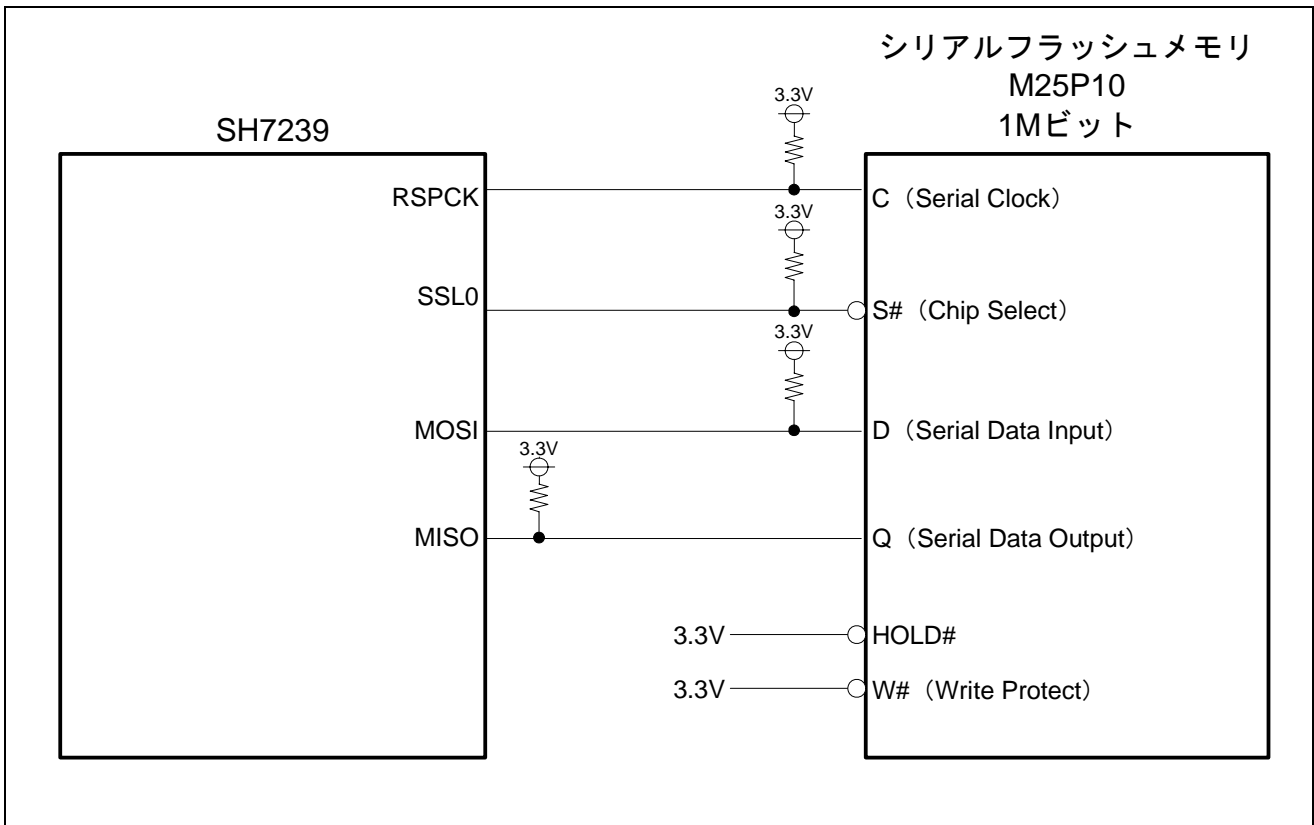


図1 シリアルフラッシュメモリ接続回路例

【注】 制御信号端子の外付け抵抗によるプルアップ/プルダウン処理について
制御信号に対するプルアップ/プルダウン処理は、マイコンの端子状態がハイインピーダンスの場合でも、外部デバイスが誤動作しないように信号線のレベルを決定します。SSL0 端子については外付け抵抗でプルアップ処理を行い、H レベルにしています。RSPCK 端子と MOSI 端子はプルアップまたはプルダウン処理をおこなってください。また MISO 端子は入力のためプルアップまたはプルダウン処理により中間電位になることを防ぎます。

表2 マルチプレクス出力

周辺機能	使用端子名	SH7239 ポートコントロールレジスタ		SH7239 マルチプレクス端子名
		レジスタ名	MD ビット設定値	
RSPi	RSPCK	PACRL2	PA6MD[2:0] = B'101	PA6/IRQ6/TCLKA/ CS6#/RSPCK/SCK1
	MOSI	PACRL2	PA7MD[2:0] = B'101	PA7/IRQ5/TCLKB/ CS5#/MOSI/TXD1
	MISO	PACRL3	PA8MD[2:0] = B'101	PA8/ IRQ4/TCLKC/CS4#/MISO/RXD1
	SSL0	PACRL3	PA9MD[2:0] = B'101	PA9/IRQ3/TCLKD/CS3#/SSL0/SCK0

【注】 SH7239 のマルチプレクス端子について
RSPCK、MOSI、MISO、SSL0 端子はマルチプレクス端子であり、初期状態は汎用入出力ポートになっています。そのためシリアルフラッシュメモリへアクセスする前に、汎用入出力ポートのコントロールレジスタによって RSPi 端子機能に設定する必要があります。

2.3 インタフェースタイミング例

SH7239 とシリアルフラッシュメモリ間のインタフェースタイミング例を示します。スレーブとなるシリアルフラッシュメモリのタイミング条件に合わせて RSPI 設定およびクロック周波数設定を行います。

図 2 にデータ転送タイミング例を示します。本応用例で使用するシリアルフラッシュメモリは、クロックの立ち上がりでデータサンプルを行い、立ち下がりでデータ変化する仕様のため、RSPI コマンドレジスタ (SPCMD) の CPOL ビットと CPHA ビットにはともに 1 を設定します。本設定により、アイドル時の RSPCK は 1 に設定され、RSPI のデータ変化タイミングを奇数エッジ (ここでは立ち下がりエッジ) に設定することができます。表 3 と表 4 に示すタイミング条件を満たすように RSPI を設定してください。

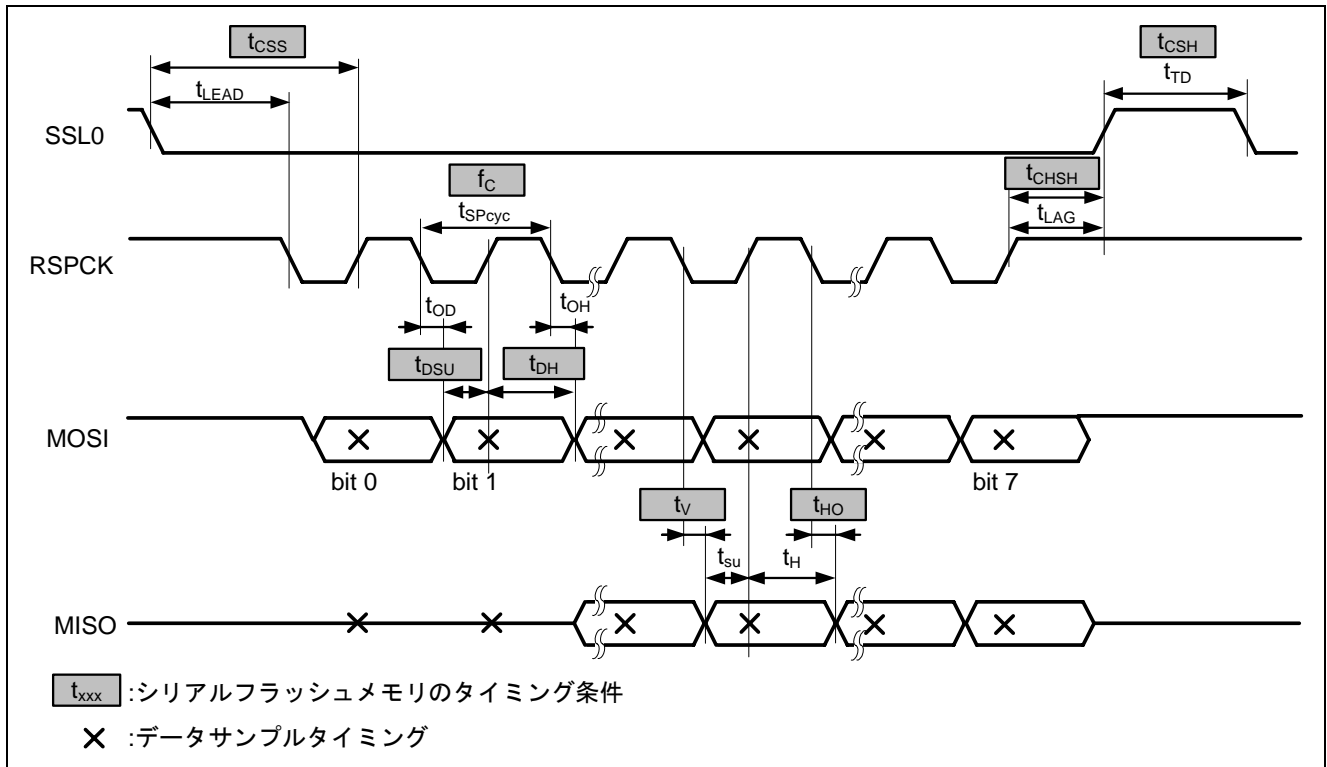


図2 データ転送タイミング例 (CPOL=1, CPHA=1 の場合)

表3 データ転送におけるシリアルフラッシュメモリのタイミング条件

シンボル	項目	説明	関連レジスタ
t _{CSS}	チップセレクト'L' セットアップ時間	SSLのアサートから RSPCK の立ち上がりでスレーブがデータサンプルするまでに必要な時間です。 以下の式を満たす設定を行います。 $t_{LEAD}(=RSPCK \text{ 遅延}) + 1/2 \times t_{SPCyc} \geq t_{CSS} \text{ (min)}$	SPCKD レジスタ SPCMD レジスタ SPBR レジスタ
t _{CSH}	チップセレクト'H' 時間	SSLのネゲート期間として必要な時間です。 以下の式を満たす設定を行います。 $t_{TD}(=次アクセス遅延) \geq t_{CSH} \text{ (min)}$	SPND レジスタ SPCMD レジスタ
f _C	シリアルクロック 周波数	スレーブが対応可能な最大動作周波数です。 以下の式を満たす設定を行います。 $f_C(\text{max}) \geq 1/t_{SPCyc}$	SPBR レジスタ SPCMD レジスタ
t _{CHSH}	チップセレクト'L' ホールド時間	最後の RSPCK の立ち上がりから SSL のネゲートまでに必要なホールド時間です。 以下の式を満たす設定を行います。 $t_{LAG}(=SSL \text{ ネゲート遅延}) \geq t_{CHSH} \text{ (min)}$	SSLND レジスタ SPCMD レジスタ
t _{DSU}	データ入力 セットアップ時間	マスタのデータ出力からデータサンプルまでに必要な時間です。 以下の式を満たす設定を行います。 $1/2 \times t_{SPCyc} - t_{OD}(\text{max}) \geq t_{DSU} \text{ (min)}$	SPBR レジスタ SPCMD レジスタ
t _{DH}	データ入力 ホールド時間	マスタがデータ出力を保持しなければならない時間です。 以下の式を満たす設定を行います。 $t_{OH}(\text{min}) + 1/2 \times t_{SPCyc} \geq t_{DH} \text{ (min)}$	SPBR レジスタ SPCMD レジスタ

表4 データ転送における SH7239 のタイミング条件

シンボル	項目	説明	関連レジスタ
t _{SU}	データ入力 セットアップ時間	スレーブのデータ出力からデータサンプルまでに必要な時間です。 以下の式を満たす設定を行います。 $1/2 \times t_{SPCyc} - t_V(\text{max}) \geq t_{SU}(\text{min})$	SPBR レジスタ SPCMD レジスタ
t _H	データ入力 ホールド時間	スレーブがデータ出力を保持しなければならない時間です。 以下の式を満たす設定を行います。 $t_{HO} \text{ (min)} + 1/2 \times t_{SPCyc} \geq t_H(\text{min})$	SPBR レジスタ SPCMD レジスタ

2.4 参考プログラムの動作

2.4.1 RSPI の初期設定例

図 3および 図 4に本参考プログラムにおけるRSPI初期設定フローを示します。本設定によりマスタモードでのSPI動作が可能となります。

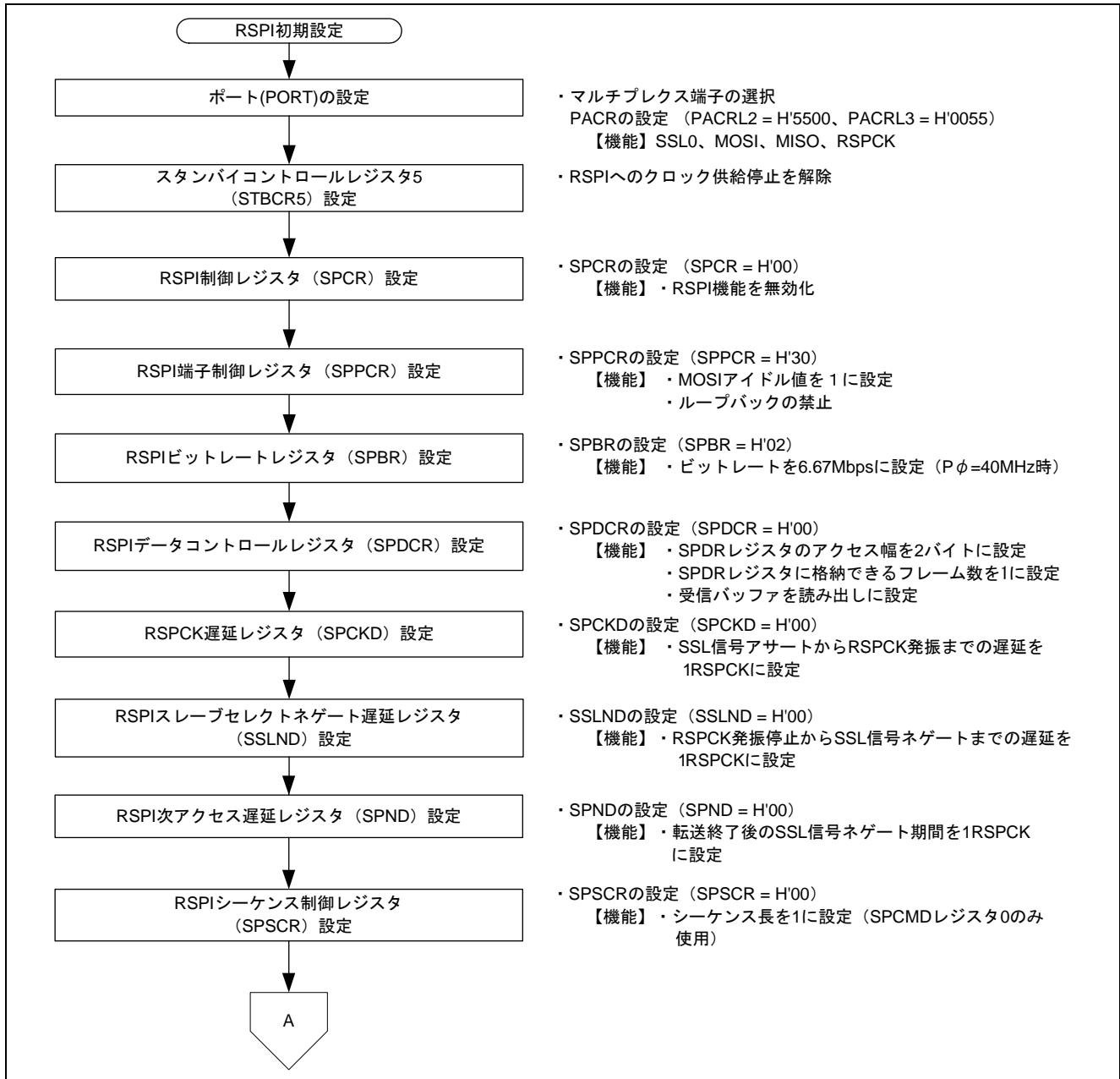


図3 参考プログラムのRSPI 初期設定フロー (1)

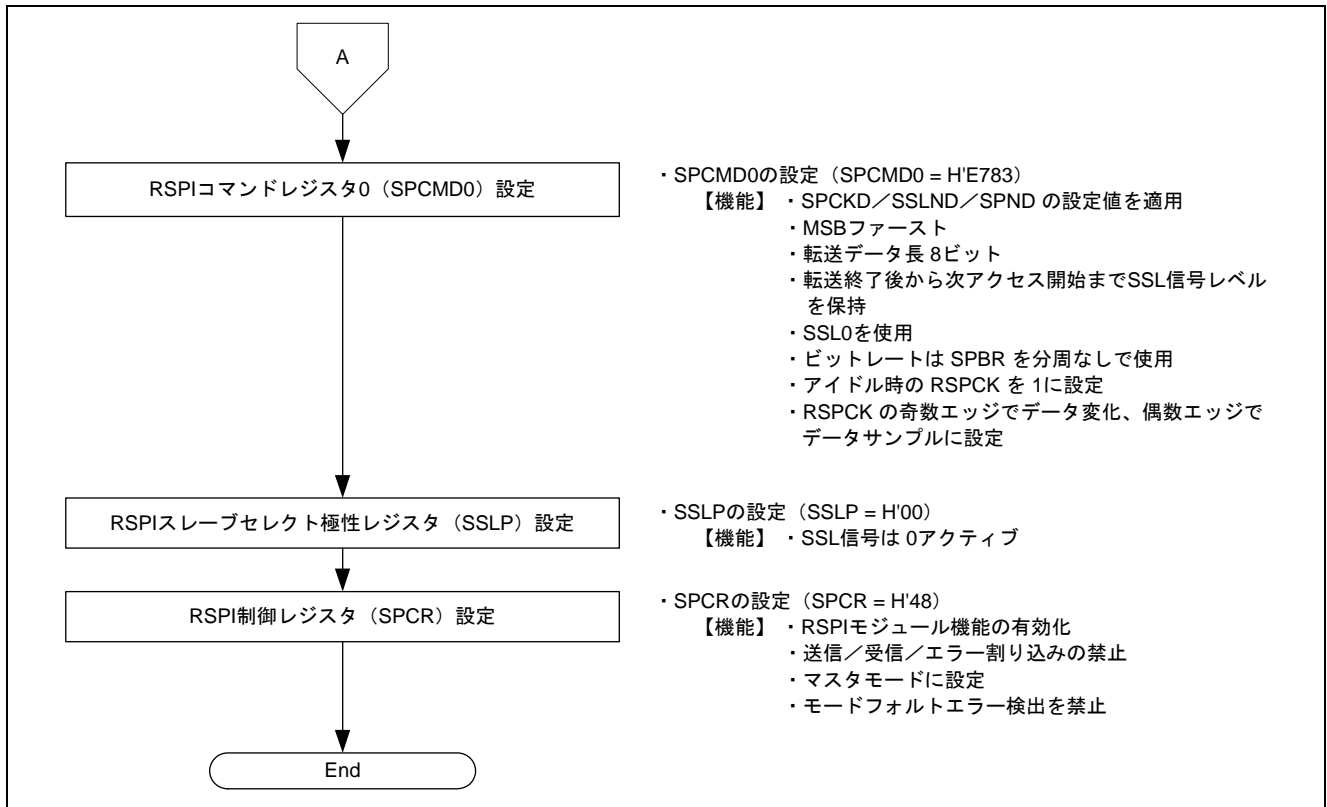


図4 参考プログラムのRSPI 初期設定フロー (2)

2.4.2 コマンド転送例

シリアルフラッシュメモリのアクセスは全てコマンド形式で行います。以下に、主なコマンドおよびコマンドシーケンス例、参考プログラムでの処理フローを示します。

なお、本応用例では ST マイクロ社製 M25P10 のコマンドを参考にしています。コマンドの詳細は使用するデバイスのデータシートを参照してください。

A. 主なコマンド

表 5にM25P10 の主なコマンドを示します。

表5 M25P10 の主なコマンド

コマンド名	オペコード	アドレス バイト数	データ バイト数	機能
Write Enable (WREN)	H'06	0	0	プログラム/イレースコマンドの許可
Write Disable (WRDI)	H'04	0	0	プログラム/イレースコマンドの禁止
Read Status Register (RDSR)	H'05	0	1 以上	ステータスリード
Write Status Register (WRSR)	H'01	0	1	ステータスライト
Read Data Bytes (READ)	H'03	3	1 以上 ^{※1}	データのリード
Page Program (PP)	H'02	3	1~128 ^{※2}	データのライト
Sector Erase (SE)	H'D8	3	0	セクタ (256Kbit) 単位のイレース
Bulk Erase (BE)	H'C7	0	0	全領域のイレース

【注】 ^{※1} 指定アドレスからインクリメントされた領域をリードします。(最終番地を超えた場合は 0 番地に戻ります。)

^{※2} 指定アドレスと同一ページ内で、インクリメントされた領域にライトします。(ページの最終番地を超えた場合はページの先頭に戻ります。)

B. コマンドシーケンス例

図 5にRead Data Bytes(READ) コマンドのシーケンス例を示します。

Read Data Bytes(READ) コマンドは、SSL 信号のアサート後、オペコード (H'03) に続けてアドレス (3 バイト) をマスタから転送します。その後、RSPCK の立ち下がり毎にスレーブから Read データが転送されます。

設定したアクセス幅の転送を繰り返すことでコマンドシーケンスを実現できますが、SSL 信号のレベルに注意が必要です。コマンドの先頭で SSL 信号をアサートしてから、コマンドの最終バイトの転送完了まで SSL 信号をネゲートしないでください。参考プログラムでは SPCMD レジスタの SSLKP ビットを 1 にセットして SSL 信号を保持しています。SSL のネゲートは全データの転送完了後に SPCR レジスタの SPE ビットを 0 クリアすることで行っています。

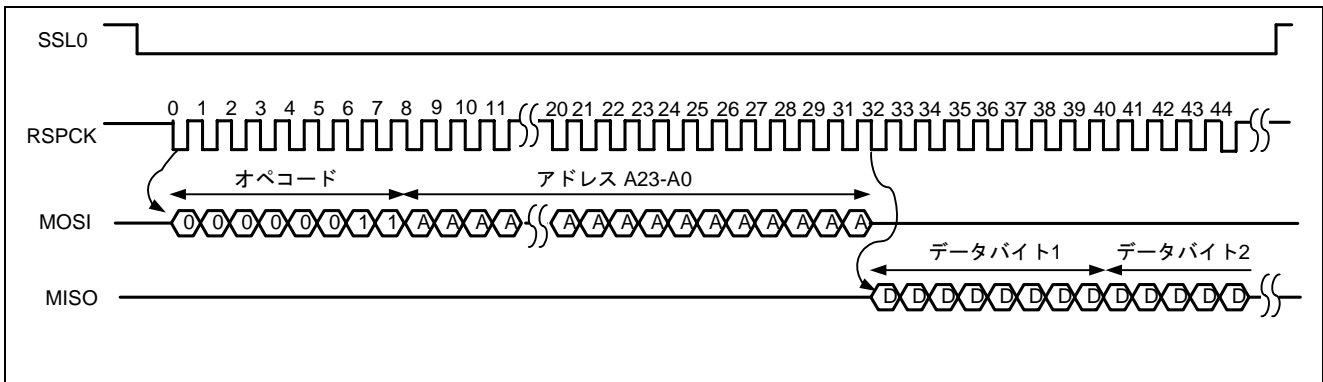


図5 リードコマンドシーケンス (H'03 オペコード)

C. 参考プログラムでのコマンド転送例

コマンドにはマスタ出力とスレーブ出力の両方を使用するリード用のコマンドと、マスタ出力のみを使用するライト用のコマンドがあります。図 6にリード/ライト用のコマンド実行処理フローを示します。また 図 7にデータ転送処理フローを示します。

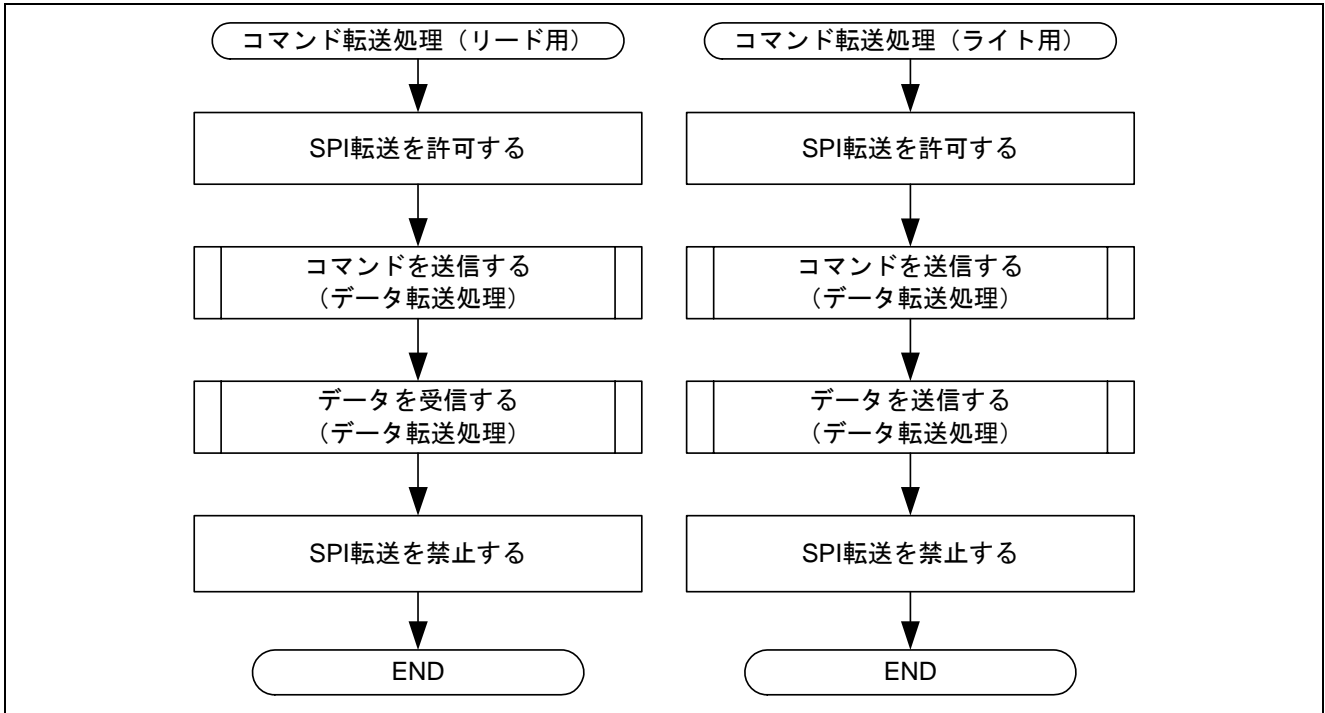


図6 参考プログラムのリード/ライト用コマンド転送フロー

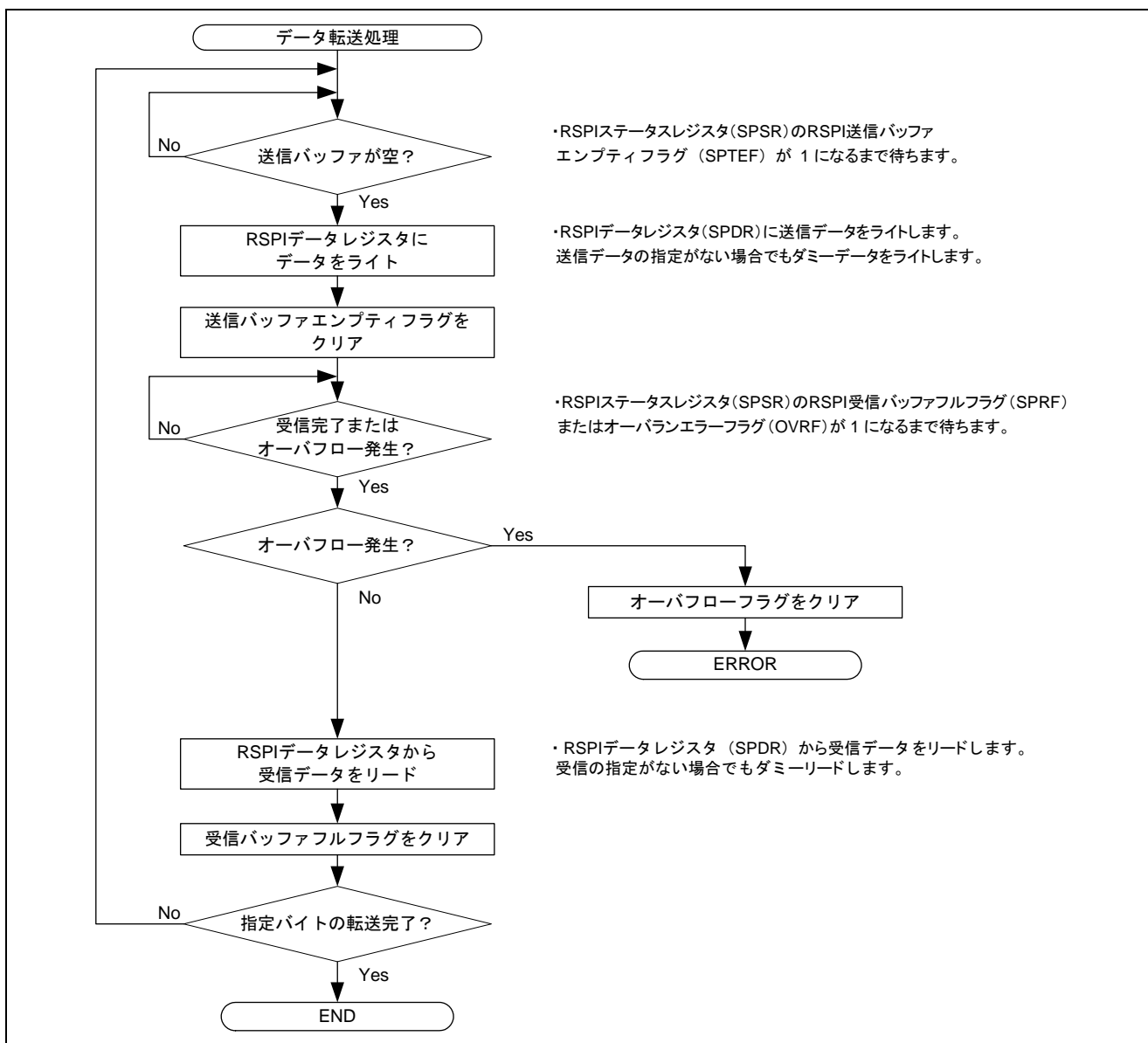
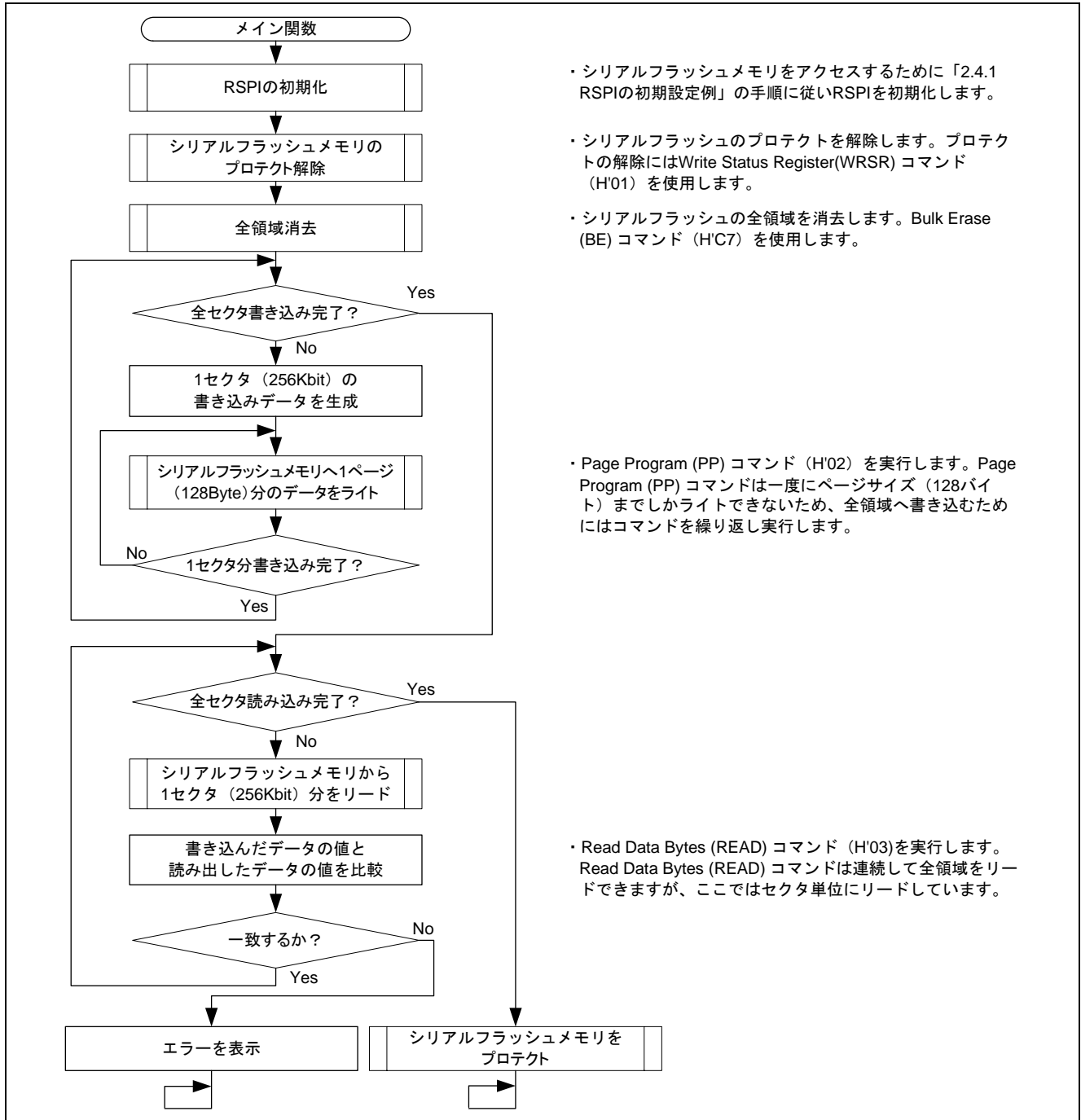


図7 参考プログラムのデータ転送処理フロー

2.4.3 メイン関数

図8に参考プログラムのメイン関数フローを示します。参考プログラムは、シリアルフラッシュメモリの全領域にライトした後、リードした値と等しいかをチェックします。



・シリアルフラッシュメモリをアクセスするために「2.4.1 RSPIの初期設定例」の手順に従いRSPIを初期化します。

・シリアルフラッシュのプロテクトを解除します。プロテクトの解除にはWrite Status Register(WRSR) コマンド (H'01) を使用します。

・シリアルフラッシュの全領域を消去します。Bulk Erase (BE) コマンド (H'C7) を使用します。

・Page Program (PP) コマンド (H'02) を実行します。Page Program (PP) コマンドは一度にページサイズ (128バイト) までしかライトできないため、全領域へ書き込むためにはコマンドを繰り返し実行します。

・Read Data Bytes (READ) コマンド (H'03) を実行します。Read Data Bytes (READ) コマンドは連続して全領域をリードできますが、ここではセクタ単位にリードしています。

図8 参考プログラムのメイン関数フロー

3. 参考プログラムリスト

3.1 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 Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2010 Renesas Electronics Corporation. All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7239 Sample Program
31 *  File Name   : main.c
32 *  Abstract    : ルネサスシリアルペリフェラルインタフェース
33 *              : シリアルフラッシュメモリのリードライト例
34 *  Version     : 1.00.00
35 *  Device      : SH7239
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 Release02).
39 *  OS          : None
40 *  H/W Platform: R0K572390 (CPU board)
41 *  Description : ルネサスシリアルペリフェラルインタフェースを用いて、
42 *              : シリアルフラッシュメモリの接続を行います。
43 *****/
44 *  History     : Aug.17,2010 Ver.1.00.00
45 *  "FILE COMMENT END"*****/

```

3.2 サンプルプログラムリスト"main.c" (2)

```

46 #include <stdio.h>
47 #include "serial_flash.h"
48
49 /* ==== マクロ定義 ==== */
50 #define TOP_ADDRESS    0                /* シリアルフラッシュメモリの先頭アドレス */
51
52 /* ==== 関数プロトタイプ宣言 ==== */
53 void main(void);
54
55 /* ==== 変数定義 ==== */
56 #pragma section DEBUG_BUFFER
57 static unsigned char data[SF_BUFF_SIZE];
58 static unsigned char rbuf[SF_BUFF_SIZE];
59 #pragma section
60
61 /*"FUNC COMMENT"*****
62 * ID          :
63 * Outline     : シリアルフラッシュメモリアクセス メイン処理
64 *-----
65 * Include     :
66 *-----
67 * Declaration : void main(void);
68 *-----
69 * Description : シリアルフラッシュメモリへのイレース、プログラム、リード処理を
70 *               : 行います。RSPI チャンネル0 を初期化後、全領域をイレースした後、先頭から
71 *               : 全領域にデータを書き込みます。結果は読み出して確認します。
72 *-----
73 * Argument    : void
74 *-----
75 * Return Value : void
76 *-----
77 * Note        : None
78 *"FUNC COMMENT END"*****/
79 void main(void)
80 {
81     int i, j, k;
82     static unsigned long addr;
83
84     /* ==== RSPI の初期化 ==== */
85     sf_init_serial_flash();
86
87     /* ==== シリアルフラッシュメモリのプロテクト解除 ==== */
88     sf_protect_ctrl( SF_REQ_UNPROTECT );
89
90     /* ==== チップイレース (1Mbit) ==== */
91     sf_chip_erase();
92

```

3.3 サンプルプログラムリスト"main.c" (3)

```

93  /* ==== データライト (1Mbit) ==== */
94  addr = TOP_ADDRESS;
95  for(i = 0; i < SF_NUM_OF_SECTOR; i++){
96  /* ---- データ初期化 (4KBYTE) ---- */
97  for(j = 0; j < SF_BUFF_SIZE; j++){
98  data[j] = (i + j) % 100;
99  }
100 /* ---- セクタサイズ (32KBYTE) をライト ---- */
101 for(k = 0; k < SF_SECTOR_SIZE/SF_BUFF_SIZE; k++){
102 /* ---- バッファサイズ (4KBYTE) をライト ---- */
103 for(j = 0; j < ( SF_BUFF_SIZE / SF_PAGE_SIZE ); j++){
104 /* ---- ページサイズ (128BYTE) をライト ---- */
105 sf_byte_program( addr, data+(j*SF_PAGE_SIZE), SF_PAGE_SIZE );
106 addr += SF_PAGE_SIZE;          /* 書き込み先アドレス更新 */
107 }
108 }
109 }
110 /* ==== データリード (1Mbit) ==== */
111 addr = TOP_ADDRESS;
112 for(i = 0; i < SF_NUM_OF_SECTOR; i++){
113 /* ---- セクタサイズ (32KBYTE) をリード ---- */
114 for(k = 0; k < SF_SECTOR_SIZE/SF_BUFF_SIZE; k++){
115 /* ---- バッファサイズ (4KBYTE) をリード ---- */
116 sf_byte_read( addr, rbuf, SF_BUFF_SIZE );
117 addr += SF_BUFF_SIZE;          /* 読み込み先アドレス更新 */
118 /* ---- バリファイチェック ---- */
119 for(j = 0; j < SF_BUFF_SIZE; j++){
120 data[j] = (i + j) % 100;      /* 書き込んだデータを再生 */
121 if( data[j] != rbuf[j] ){
122 puts("Error: verify error¥n");
123 fflush(stdout);
124 while(1);
125 }
126 }
127 }
128 }
129 /* ==== シリアルフラッシュメモリのプロテクト ==== */
130 sf_protect_ctrl( SF_REQ_PROTECT );
131
132 while(1){
133 /* loop */
134 }
135 }
136
137 /* End of File */

```


3.4 サンプルプログラムリスト"serial_flash.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2010 Renesas Electronics Corporation. All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7239 Sample Program
31 *  File Name   : serial_flash.c
32 *  Abstract    : ルネサスシリアルペリフェラルインタフェース
33 *              : シリアルフラッシュメモリのリードライト例
34 *  Version     : 1.00.00
35 *  Device      : SH7239
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 Release02).
39 *  OS          : None
40 *  H/W Platform: R0K572390 (CPU board)
41 *  Description : ルネサスシリアルペリフェラルインタフェースを用いて、
42 *              : シリアルフラッシュメモリの接続を行います。
43 *****/
44 *  History     : Aug.17,2010 Ver.1.00.00
45 *  "FILE COMMENT END"*****/
46 #include <stdio.h>
47 #include <machine.h>
48 #include "iodefine.h"
49 #include "serial_flash.h"
50
```

3.5 サンプルプログラムリスト"serial_flash.c" (2)

```

51  /* ==== マクロ定義 ==== */
52  #define SFLASHCMD_CHIP_ERASE  0xc7
53  #define SFLASHCMD_SECTOR_ERASE 0xd8
54  #define SFLASHCMD_BYTE_PROGRAM 0x02
55  #define SFLASHCMD_BYTE_READ    0x0B
56  #define SFLASHCMD_BYTE_READ_LOW 0x03
57  #define SFLASHCMD_WRITE_ENABLE 0x06
58  #define SFLASHCMD_WRITE_DISABLE 0x04
59  #define SFLASHCMD_READ_STATUS  0x05
60  #define SFLASHCMD_WRITE_STATUS 0x01
61  #define UNPROTECT_WR_STATUS    0x00
62  #define PROTECT_WR_STATUS      0x3C
63
64  /* ==== 関数プロトタイプ宣言 ==== */
65  /*** Local function ***/
66  static void write_enable(void);
67  static void write_disable(void);
68  static void busy_wait(void);
69  static unsigned char read_status(void);
70  static void write_status(unsigned char status);
71  static void io_init_rspi(void);
72  static void io_cmd_exe(unsigned char *ope, int ope_sz, unsigned char *data, int data_sz);
73  static void io_cmd_exe_rdmode(unsigned char *ope, int ope_sz, unsigned char *rd, int rd_sz);
74  static int io_rspi_transfer(unsigned char *write_data, unsigned char *read_data, int data_sz);
75
76  /* ==== 変数定義 ==== */
77
78  /*"FUNC COMMENT"*****
79  * ID          :
80  * Outline     : シリアルフラッシュメモリの初期化
81  *-----
82  * Include     :
83  *-----
84  * Declaration : void sf_init_serial_flash(void);
85  *-----
86  * Description : シリアルフラッシュメモリにアクセスするための初期化を行います。
87  *              : ルネサスシリアルペリフェラルインタフェース(RSPI)を初期化します。
88  *-----
89  * Argument    : void
90  *-----
91  * Return Value : void
92  *-----
93  * Note        : None
94  *"FUNC COMMENT END"*****/
95  void sf_init_serial_flash(void)
96  {
97      /* ==== RSPI の初期化 ==== */
98      io_init_rspi();
99  }

```

3.6 サンプルプログラムリスト"serial_flash.c" (3)

```

100 /*"FUNC COMMENT"*****
101 * ID      :
102 * Outline : プロテクト操作
103 *-----
104 * Include :
105 *-----
106 * Declaration : void sf_protect_ctrl(enum sf_req req);
107 *-----
108 * Description : シリアルフラッシュメモリのプロテクト設定または解除を行います。
109 *              : 設定内容は引数 req で指定します。プロテクトの初期値や解除方法は
110 *              : シリアルフラッシュメモリの仕様によって異なります。
111 *-----
112 * Argument   : enum sf_req req ; I : SF_REQ_UNPROTECT -> 全セクタプロテクト解除
113 *              :                   SF_REQ_PROTECT   -> 全セクタプロテクト
114 *-----
115 * Return Value : void
116 *-----
117 * Note        : None
118 /*"FUNC COMMENT END"*****
119 void sf_protect_ctrl(enum sf_req req)
120 {
121     if( req == SF_REQ_UNPROTECT ){
122         write_status( UNPROTECT_WR_STATUS); /* 全領域プロテクト解除 */
123     }
124     else{
125         write_status( PROTECT_WR_STATUS ); /* 全領域プロテクト */
126     }
127 }
128 /*"FUNC COMMENT"*****
129 * ID      :
130 * Outline : チップイレース
131 *-----
132 * Include :
133 *-----
134 * Declaration : void sf_chip_erase(void);
135 *-----
136 * Description : シリアルフラッシュメモリの全ビットをイレースします。
137 *              : イレースまたはプログラムする前にはライトイネーブルコマンドを
138 *              : 発行する必要があります。またイレースまたはプログラム後は
139 *              : シリアルフラッシュメモリのステータスを確認しビジー状態が解除
140 *              : されたことを確認してください。
141 *-----
142 * Argument   : void
143 *-----
144 * Return Value : void
145 *-----
146 * Note        : None
147 /*"FUNC COMMENT END"*****

```

3.7 サンプルプログラムリスト"serial_flash.c" (4)

```

148 void sf_chip_erase(void)
149 {
150     unsigned char cmd[1];
151     cmd[0] = SFLASHCMD_CHIP_ERASE;
152
153     write_enable();
154     io_cmd_exe(cmd, 1, NULL, 0);
155     busy_wait();
156 }
157 /*"FUNC COMMENT"*****
158 * ID          :
159 * Outline     : セクタイレース
160 *-----
161 * Include     :
162 *-----
163 * Declaration : void sf_sector_erase(int sector_no);
164 *-----
165 * Description : シリアルフラッシュメモリの指定セクタをイレースします。
166 *              : イレースまたはプログラムするにはライトイネーブルコマンドを
167 *              : 発行する必要があります。またイレースまたはプログラム後は
168 *              : シリアルフラッシュメモリのステータスを確認しビジー状態が解除
169 *              : されたことを確認してください。
170 *-----
171 * Argument    : int sector_no ; I : セクタ番号
172 *-----
173 * Return Value : void
174 *-----
175 * Note        : None
176 *"FUNC COMMENT END"*****/
177 void sf_sector_erase(int sector_no)
178 {
179     unsigned char cmd[4];
180     unsigned long addr = sector_no * SF_SECTOR_SIZE;
181
182     cmd[0] = SFLASHCMD_SECTOR_ERASE;
183     cmd[1] = (addr >> 16) & 0xff;
184     cmd[2] = (addr >> 8) & 0xff;
185     cmd[3] = addr & 0xff;
186
187     write_enable();
188     io_cmd_exe(cmd, 4, NULL, 0);
189     busy_wait();
190 }

```

3.8 サンプルプログラムリスト"serial_flash.c" (5)

```
191  /*"FUNC COMMENT"*****
192  * ID      :
193  * Outline  : データプログラム
194  *-----
195  * Include  :
196  *-----
197  * Declaration : void sf_byte_program(unsigned long addr, unsigned char *buf, int size);
198  *-----
199  * Description : シリアルフラッシュメモリに指定データをプログラムします。
200  *              : イレースまたはプログラムする前にはライトイネーブルコマンドを
201  *              : 発行する必要があります。またイレースまたはプログラム後は
202  *              : シリアルフラッシュメモリのステータスを確認しビジー状態が解除
203  *              : されたことを確認してください。
204  *              : 最大ライトデータサイズはデバイスによって制限されます。
205  *-----
206  * Argument  : unsigned long addr ; I : ライトするシリアルフラッシュメモリのアドレス
207  *              : unsigned char *buf ; I : ライトデータを格納するバッファのアドレス
208  *              : int size ; I : ライトするバイト数
209  *-----
210  * Return Value : void
211  *-----
212  * Note      : None
213  *"FUNC COMMENT END"*****/
214 void sf_byte_program(unsigned long addr, unsigned char *buf, int size)
215 {
216     unsigned char cmd[4];
217
218     cmd[0] = SFLASHCMD_BYTE_PROGRAM;
219     cmd[1] = (unsigned char)((addr >> 16) & 0xff);
220     cmd[2] = (unsigned char)((addr >> 8) & 0xff);
221     cmd[3] = (unsigned char)(addr & 0xff);
222     write_enable();
223     io_cmd_exe(cmd, 4, buf, size);
224     busy_wait();
225 }
```

3.9 サンプルプログラムリスト"serial_flash.c" (6)

```

226 /*"FUNC COMMENT"*****
227 * ID      :
228 * Outline : データリード
229 *-----
230 * Include :
231 *-----
232 * Declaration : void sf_byte_read(unsigned long addr, unsigned char *buf, int size);
233 *-----
234 * Description : シリアルフラッシュメモリを指定バイト数だけリードします。
235 *-----
236 * Argument   : unsigned long addr ; I : リードするシリアルフラッシュメモリのアドレス
237 *             : unsigned char *buf ; I : リードデータを格納するバッファのアドレス
238 *             : int size           ; I : リードするバイト数
239 *-----
240 * Return Value : void
241 *-----
242 * Note        : None
243 /*"FUNC COMMENT END"*****/
244 void sf_byte_read(unsigned long addr, unsigned char *buf, int size)
245 {
246     unsigned char cmd[4];
247
248     cmd[0] = SFLASHCMD_BYTE_READ_LOW;
249     cmd[1] = (unsigned char)((addr >> 16) & 0xff);
250     cmd[2] = (unsigned char)((addr >> 8) & 0xff);
251     cmd[3] = (unsigned char)(addr & 0xff);
252     io_cmd_exe_rdmode(cmd, 4, buf, size);
253 }
254 /*"FUNC COMMENT"*****
255 * ID      :
256 * Outline : 書き込み許可
257 *-----
258 * Include :
259 *-----
260 * Declaration : static void write_enable(void);
261 *-----
262 * Description : ライトイネーブルコマンドを発行して、シリアルフラッシュメモリへの
263 *             : イレースまたはプログラム動作を許可します。
264 *-----
265 * Argument   : void
266 *-----
267 * Return Value : void
268 *-----
269 * Note        : None
270 /*"FUNC COMMENT END"*****/
271 static void write_enable(void)
272 {
273     unsigned char cmd[1];
274     cmd[0] = SFLASHCMD_WRITE_ENABLE;
275     io_cmd_exe(cmd, 1, NULL, 0);
276 }

```

3.10 サンプルプログラムリスト"serial_flash.c" (7)

```

277 /*"FUNC COMMENT"*****
278 * ID      :
279 * Outline  : 書き込み禁止
280 *-----
281 * Include  :
282 *-----
283 * Declaration : static void write_disable(void);
284 *-----
285 * Description : ライトディスエーブルコマンドを発行して、シリアルフラッシュメモリへの
286 *              : イレースまたはプログラム動作を禁止します。
287 *-----
288 * Argument  : void
289 *-----
290 * Return Value : void
291 *-----
292 * Note      : None
293 *"FUNC COMMENT END"*****/
294 static void write_disable(void)
295 {
296     unsigned char cmd[1];
297     cmd[0] = SFLASHCMD_WRITE_DISABLE;
298     io_cmd_exe(cmd, 1, NULL, 0);
299 }
300 /*"FUNC COMMENT"*****
301 * ID      :
302 * Outline  : ビジー待ち
303 *-----
304 * Include  :
305 *-----
306 * Declaration : static void busy_wait(void);
307 *-----
308 * Description : シリアルフラッシュメモリのステータスがビジー状態の場合は内部で
309 *              : ループします。
310 *-----
311 * Argument  : void
312 *-----
313 * Return Value : void
314 *-----
315 * Note      : None
316 *"FUNC COMMENT END"*****/
317 static void busy_wait(void)
318 {
319     while ((read_status() & 0x01) != 0) {
320         /* serial flash is busy */
321     }
322 }

```

3.11 サンプルプログラムリスト"serial_flash.c" (8)

```

323 /*"FUNC COMMENT"*****
324 * ID      :
325 * Outline  : ステータスリード
326 *-----
327 * Include  :
328 *-----
329 * Declaration : static unsigned char read_status(void);
330 *-----
331 * Description : シリアルフラッシュメモリのステータスをリードします。
332 *-----
333 * Argument   : void
334 *-----
335 * Return Value : ステータスレジスタの値
336 *-----
337 * Note       : None
338 *"FUNC COMMENT END"*****/
339 static unsigned char read_status(void)
340 {
341     unsigned char buf;
342     unsigned char cmd[1];
343
344     cmd[0] = SFLASHCMD_READ_STATUS;
345     io_cmd_exe_rdmode(cmd, 1, &buf, 1);
346     return buf;
347 }
348 /*"FUNC COMMENT"*****
349 * ID      :
350 * Outline  : ステータスライト
351 *-----
352 * Include  :
353 *-----
354 * Declaration : static void write_status(unsigned char status);
355 *-----
356 * Description : シリアルフラッシュメモリのステータスをライトします。
357 *-----
358 * Argument   : unsigned char status ; I : status register value
359 *-----
360 * Return Value: void
361 *-----
362 * Note       : None
363 *"FUNC COMMENT END"*****/
364 static void write_status(unsigned char status)
365 {
366     unsigned char cmd[2];
367
368     cmd[0] = SFLASHCMD_WRITE_STATUS;
369     cmd[1] = status;
370
371     write_enable();
372     io_cmd_exe(cmd, 2, NULL, 0);
373     busy_wait();
374 }

```


3.12 サンプルプログラムリスト"serial_flash.c" (9)

```

375  /*"FUNC COMMENT"*****
376  * ID      :
377  * Outline  : RSPI の初期化
378  *-----
379  * Include  :
380  *-----
381  * Declaration : static void io_init_rsipi(void);
382  *-----
383  * Description : ルネサスシリアルペリフェラルインタフェースを初期化します。
384  *              : マスタモードに設定し、シリアルフラッシュメモリの仕様に合わせた
385  *              : 転送設定を行います。
386  *-----
387  * Argument  : void
388  *-----
389  * Return Value: void
390  *-----
391  * Note      : None
392  *"FUNC COMMENT END"*****
393  static void io_init_rsipi(void)
394  {
395  /* ==== PFC ==== */
396  PFC.PACRL3.BIT.PA9MD = 5; /* SSL0 */
397  PFC.PACRL3.BIT.PA8MD = 5; /* MISO */
398  PFC.PACRL2.BIT.PA7MD = 5; /* MOSI */
399  PFC.PACRL2.BIT.PA6MD = 5; /* RSPCK */
400
401  /* ==== CPG ==== */
402  STB.CR5.BIT._RSPI = 0; /* RSPI active */
403
404  /* ==== RSPI ==== */
405  RSPI.SPCR.BYTE = 0x00; /* RSPI を動作禁止 */
406  RSPI.SPPCR.BYTE = 0x30; /* MOSI アイドル固定値 = 1 */
407  RSPI.SPBR.BYTE = 0x02; /* ベースのビットレートを 6.67MHz に設定 (PΦ=40MHz) */
408  RSPI.SPDCR.BYTE = 0x00; /* SPDR レジスタのアクセス幅: 16 ビット */
409  RSPI.SPCKD.BYTE = 0x00; /* RSPCK 遅延: 1 RSPCK */
410  RSPI.SSLND.BYTE = 0x00; /* SSL ネゲート遅延: 1 RSPCK */
411  RSPI.SPND.BYTE = 0x00; /* 次アクセス遅延: 1 RSPCK */
412  RSPI.SPSCR.BYTE = 0x00; /* シーケンス長: 1 (SPCMD0 のみ使用) */
413  RSPI.SPCMD0.WORD = 0xE783; /* MSB ファースト */
414  /* データ長: 8bit */
415  /* 転送終了後も SSL 信号レベルを保持する */
416  /* ビットレート: ベースビットレートの分周なし */
417  /* アイドル時の RSPCK: 1 */
418  /* 奇数エッジでデータ変化、偶数エッジでデータサンプル */
419  RSPI.SSLP.BYTE = 0x00; /* SSLP = b'0 SSL signal 0-active */
420  RSPI.SPCR.BYTE = 0x48; /* マスタモード */
421  /* 割り込み禁止 */
422  /* RSPI の動作許可 */
423  }

```

3.13 サンプルプログラムリスト"serial_flash.c" (10)

```
424  /*"FUNC COMMENT"*****
425  * ID      :
426  * Outline  : コマンド実行(リードデータなし)
427  *-----
428  * Include  :
429  *-----
430  * Declaration : static int io_cmd_exe(unsigned char *ope, int ope_sz,
431  *      :      unsigned char *data,int data_sz)
432  *-----
433  * Description : 指定されたコマンドを実行します。
434  *      : 引数 ope を送信した後、引数 data を送信します。受信データは破棄します。
435  *      : ope_sz は 0~8 のいずれかの値を設定してください。
436  *      : data_sz は 0~256 のいずれかの値を設定してください。
437  *-----
438  * Argument  : unsigned char *ope ; I : 送信するオペコード部とアドレス部の先頭アドレス
439  *      : int ope_sz      ; I : オペコード部とアドレス部のバイト数
440  *      : unsigned char *data; I : 送信するデータ部の先頭アドレス
441  *      : int data_sz    ; I : データ部のバイト数
442  *-----
443  * Return Value : void
444  *-----
445  * Note      : None
446  *"FUNC COMMENT END"*****/
447  static void io_cmd_exe(unsigned char *ope, int ope_sz, unsigned char *data, int data_sz)
448  {
449  /* ---- SPI 転送許可 ---- */
450  RSPI.SPCR.BIT.SPE = 1;
451
452  /* ---- MOSI ---- */
453  io_rspi_transfer(ope, NULL, ope_sz);
454  io_rspi_transfer(data, NULL, data_sz);
455
456  /* ---- SPI 転送終了 (SSL ネゲート) ---- */
457  RSPI.SPCR.BIT.SPE = 0;
458  }
```

3.14 サンプルプログラムリスト"serial_flash.c" (11)

```

459  /*"FUNC COMMENT"*****
460  * ID      :
461  * Outline  : コマンド実行(リードデータあり)
462  *-----
463  * Include  :
464  *-----
465  * Declaration : static void io_cmd_exe_rdmode(unsigned char *ope, int ope_sz,
466  *          :          unsigned char *rd, int rd_sz)
467  *-----
468  * Description : 指定されたコマンドを実行します。
469  *          : 引数 ope を送信した後、引数 rd にデータを受信します。
470  *          : ope_sz は 0~8 のいずれかの値を設定してください。
471  *          : rd_sz は 0 以上 の値を設定することが可能です。
472  *-----
473  * Argument  : unsigned char *ope ; I : 送信するオペコード部とアドレス部の先頭アドレス
474  *          : int ope_sz          ; I : オペコード部とアドレス部のバイト数
475  *          : unsigned char *rd  ; I : 受信データを格納するバッファアドレス
476  *          : int rd_sz          ; I : データ部のバイト数
477  *-----
478  * Return Value : void
479  *-----
480  * Note       : None
481  *"FUNC COMMENT END"*****/
482  static void io_cmd_exe_rdmode(unsigned char *ope, int ope_sz, unsigned char *rd, int rd_sz)
483  {
484  /* ---- SPI 転送許可 ---- */
485  RSPI.SPCR.BIT.SPE = 1;
486
487  /* ---- MISO ---- */
488  io_rspi_transfer(ope, NULL, ope_sz);
489  io_rspi_transfer(NULL, rd, rd_sz);
490
491  /* ---- SPI 転送終了 (SSL ネゲート) ---- */
492  RSPI.SPCR.BIT.SPE = 0;
493  }

```

3.15 サンプルプログラムリスト"serial_flash.c" (12)

```
494  /*"FUNC COMMENT"*****
495  * ID      :
496  * Outline : RSPI データ転送
497  *-----
498  * Include :
499  *-----
500  * Declaration : int io_rspi_transfer(unsigned char *write_data,
501  *      :      unsigned char *read_data, int data_sz);
502  *-----
503  * Description : コマンド、データをバイト単位で転送します。
504  *      : 引数 write_data からオペコードまたはデータを送信し、
505  *      : 引数 read_data にデータを受信します。
506  *      : 引数 write_data が NULL の場合、ダミーデータ (0xff) の送信を行います。
507  *      : 引数 read_data が NULL の場合、データを受信は行いません。
508  *-----
509  * Argument  : unsigned char *write_data : I : 送信データの先頭アドレス
510  *      :  unsigned char *read_data : O : 受信データを格納するバッファアドレス
511  *      :  int data_sz : I : 送信、受信データのバイト数
512  *-----
513  * Return Value : 0 : 正常転送
514  *      : -1: オーバランエラー発生
515  *-----
516  * Note      : None
517  *"FUNC COMMENT END"*****/
518  static int io_rspi_transfer(unsigned char *write_data, unsigned char *read_data, int data_sz)
519  {
520      unsigned short tmp;
521
522      while(data_sz--){
523          while(RSPI.SPSR.BIT.SPTEF == 0){
524              /* wait */
525          }
526          /* データレジスタへ転送データを書き込み */
527          if(write_data != (unsigned char *)0){
528              tmp = (unsigned short)*write_data++;
529          }
530          else{
531              tmp = 0x00ff; /* ダミーライト用データ */
532          }
533
534          RSPI.SPDR.WORD = 0x00ff & tmp;
535
536          RSPI.SPSR.BIT.SPTEF = 0; /* 送信のための 0 クリア */
537
538      }
```

3.16 サンプルプログラムリスト"serial_flash.c" (13)

```
539     /* 受信完了待ち */
540     while((RSPI.SPSR.BYTE & 0x81) == 0x00){
541         /* 受信バッファがフル状態になるか、オーバーランエラー発生までウェイト */
542     }
543
544     /* オーバランエラー発生? */
545     if(RSPI.SPSR.BIT.OVRF == 1){
546         RSPI.SPSR.BIT.OVRF = 0;
547         return -1; /* オーバランエラー発生 */
548     }
549
550     /* 受信データリード */
551     tmp = RSPI.SPDR.WORD;
552     if(read_data != (unsigned char *)0){
553         *read_data++ = (unsigned char)tmp;
554     }
555     RSPI.SPSR.BIT.SPRF = 0;
556 }
557
558 return 0;
559 }
560
561 /* End of File */
```

3.17 サンプルプログラムリスト"serial_flash.h" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Electronics Corporation and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Electronics Corporation and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2010 Renesas Electronics Corporation. All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7239 Sample Program
31 *  File Name   : serial_flash.h
32 *  Abstract    : ルネサスシリアルペリフェラルインタフェース
33 *              : シリアルフラッシュメモリのリードライト例
34 *  Version    : 1.00.00
35 *  Device     : SH7239
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 Release02).
39 *  OS        : None
40 *  H/W Platform: R0K572390 (CPU board)
41 *  Description: ルネサスシリアルペリフェラルインタフェースを用いて、
42 *              : シリアルフラッシュメモリの接続を行います。
43 *****/
44 *  History    : Aug.17,2010 Ver.1.00.00
45 *  "FILE COMMENT END"*****/
```

3.18 サンプルプログラムリスト"serial_flash.h" (2)

```

46  #ifndef _SERIAL_FLASH_H_
47  #define _SERIAL_FLASH_H_
48
49  /* ==== マクロ定義 ==== */
50  #define SF_PAGE_SIZE      128          /* シリアルフラッシュメモリのページサイズ */
51  #define SF_SECTOR_SIZE   0x8000      /* セクタサイズ = 32KB */
52  #define SF_NUM_OF_SECTOR 4           /* セクタ数 4 */
53  #define SF_BUFF_SIZE     0x1000      /* シリアルフラッシュデータの */
54                                     /* ベリファイ用バッファ領域 = 4KB */
55  enum sf_req{
56      SF_REQ_PROTECT = 0,              /* プロテクト要求 */
57      SF_REQ_UNPROTECT                /* プロテクト解除要求 */
58  };
59  /* ==== 関数プロトタイプ宣言 ==== */
60  void sf_init_serial_flash(void);
61  void sf_protect_ctrl(enum sf_req req);
62  void sf_chip_erase(void);
63  void sf_sector_erase(int sector_no);
64  void sf_byte_program(unsigned long addr, unsigned char *buf, int size);
65  void sf_byte_read(unsigned long addr, unsigned char *buf, int size);
66
67  #endif /* _SERIAL_FLASH_H_ */
68  /* End of File */

```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A, SH2A-FPU ソフトウェアマニュアル Rev.3.00
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- ハードウェアマニュアル
SH7239 グループ、SH7237 グループ ユーザーズマニュアル ハードウェア編 Rev.1.00
(最新版をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問合せ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.11.22	—	初版発行

製品ご使用上の注意事項

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

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

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

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

5. 製品間の相違について

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

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

ご注意書き

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

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

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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