

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

SH7262/SH7264 グループ

ルネサスシリアルペリフェラルインタフェース

EEPROM リードライト例

要旨

本アプリケーションノートは、SH7262/SH7264 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 EEPROM のリードライト例について説明します。

動作確認デバイス

SH7262/SH7264

以下、総称して「SH7264」として説明します。

目次

| | |
|--------------------|---|
| 1. はじめに..... | 2 |
| 2. 応用例の説明..... | 3 |
| 3. 参考プログラムリスト..... | 6 |
| 4. 参考ドキュメント..... | 6 |

1. はじめに

1.1 仕様

- 16K バイト (128K ビット) の EEPROM を SH7264 と接続します。
- EEPROM へのアクセスには、RSPI のチャンネル 0 を使用します。

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 | SH7262/SH7264 |
| 動作周波数 | 内部クロック : 144 MHz バスクロック : 72 MHz 周辺クロック : 36 MHz |
| 統合開発環境 | ルネサステクノロジ製 High-performance Embedded Workshop Ver.4.04.01 |
| C コンパイラ | ルネサステクノロジ製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.02 Release00 |
| コンパイルオプション | High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chginclpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo) |

1.4 関連アプリケーションノート

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

- SH7262/SH7264 グループ 初期設定例

2. 応用例の説明

本応用例では、SH7264（マスタ）と SPI 互換の EEPROM（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 RSPI の動作概要

SH7264 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、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。

RSPI ではチャンネル 0 とチャンネル 1 の両方を使用できますが、本応用例ではチャンネル 0 を使用しています。

2.2 EEPROM の端子接続例

表 1 に本応用例で使用する SPI 互換 EEPROM（ルネサステクノロジ社製 N58X25128I）の仕様を示します。

表 1 本応用例で使用する EEPROM の仕様

| 項目 | 仕様 |
|---------|-----------------------------------|
| 電源 | 単一電源 1.8V~5.5V |
| 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 | 5MHz (2.5V~5.5V)、3MHz (1.8V~5.5V) |
| 容量 | 16K バイト (128K ビット) |
| ページ書き換え | ページサイズ 64 バイト |
| 書き換え時間 | 5ms (2.5V 以上) / 8ms (1.8V 以上) |
| 書き換え回数 | 100 万回 |

図 1 に EEPROM 接続回路例を示します。SH7264 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。

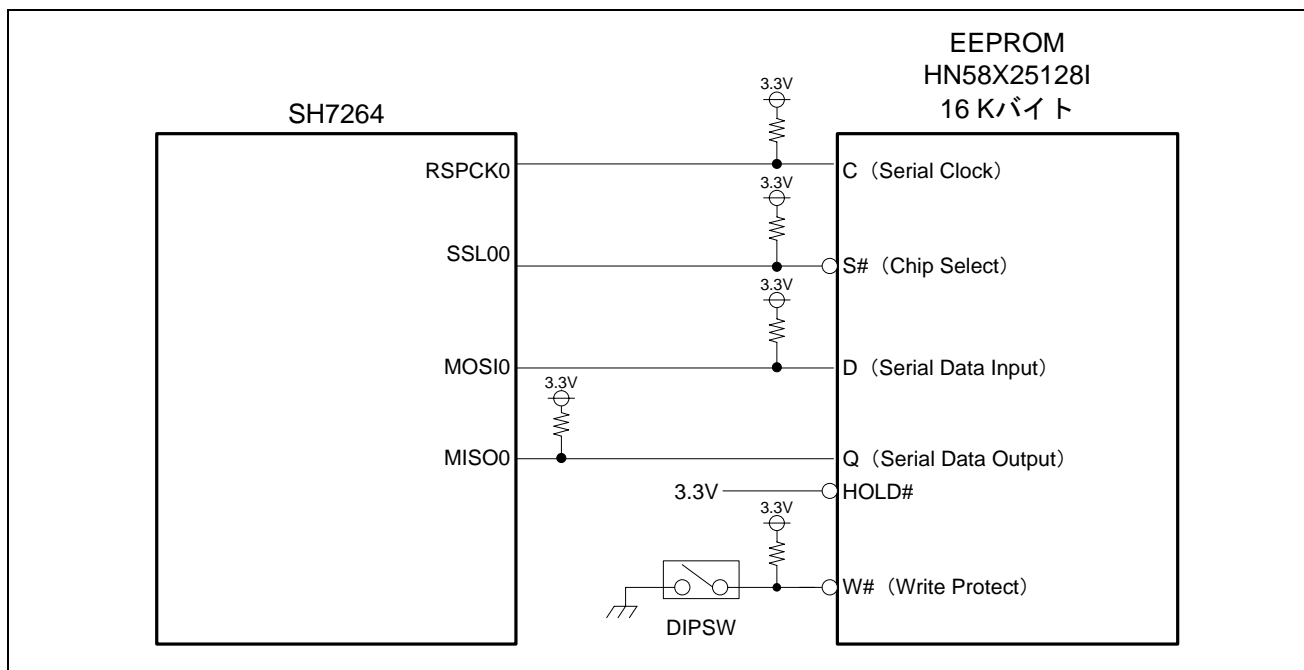


図 1 EEPROM 接続回路例

【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高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。SSL00 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。RSPCK0 端子と MOSI0 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 MISO0 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 | SH7264 ポートコントロールレジスタ | | SH7264 マルチプレクス端子名 |
|------|--------|----------------------|---------------------|---|
| | | レジスタ名 | MD ビット設定値 | |
| RSPI | MISO0 | PFCR3 | PF12MD[2:0] = B'011 | PF12/BS#/MISO0/TIOC3D/SPDIF_OUT |
| | MOSI0 | PFCR2 | PF11MD[2:0] = B'011 | PF11/A25/SSIDATA3/MOSI0/TIOC3C/SPDIF_IN |
| | SSL00 | PFCR2 | PF10MD[2:0] = B'011 | PF10/A24/SSIWS3/SSL00/TIOC3B/FCE# |
| | RSPCK0 | PFCR2 | PF9MD[2:0] = B'011 | PF9/A23/SSISCK3/RSPCK0/TIOC3A/FRB |

【注】 SH7264 のマルチプレクス端子について

MISO0、MOSI0、SSL00、RSPCK0 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 EEPROM 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

SH7264 と EEPROM 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 EEPROM 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 EEPROM 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、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。

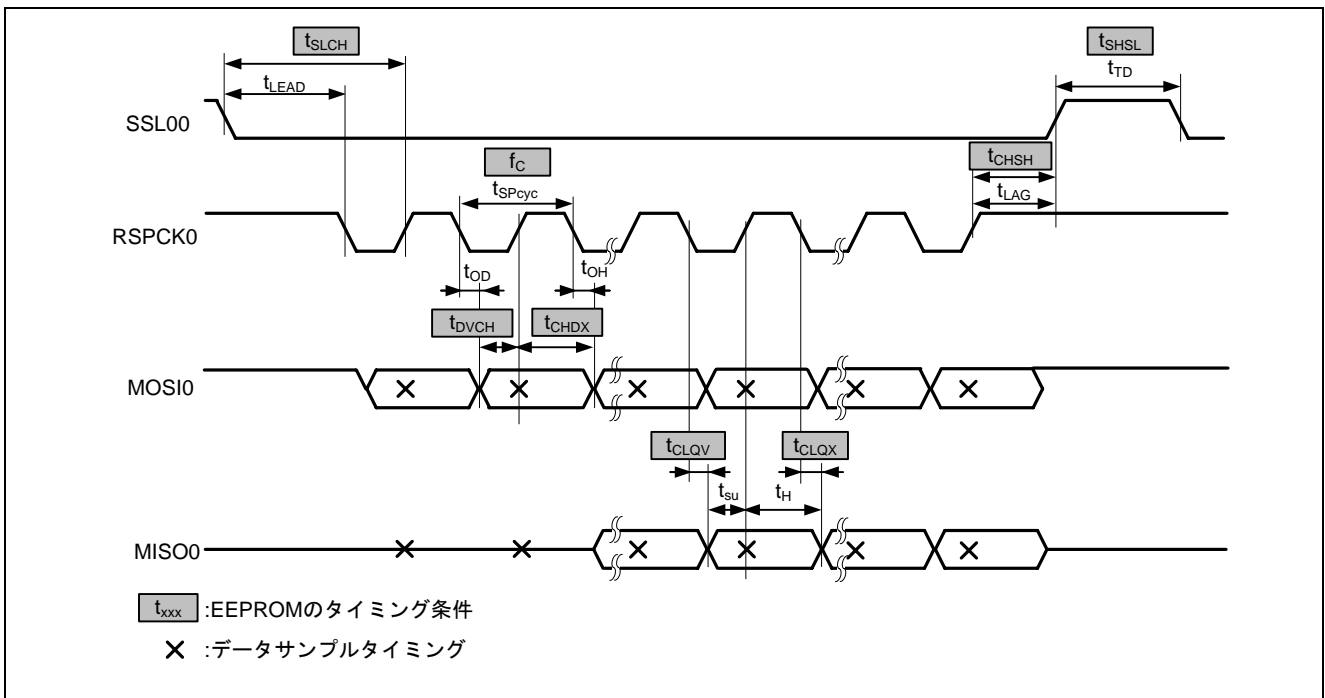


図2 データ転送タイミング例 (CPOL=1, CPHA=1 の場合)

表3 データ転送における EEPROM のタイミング条件

| シンボル | 項目 | 説明 | 関連レジスタ |
|------------|--------------------|--|--------------------------|
| t_{SLCH} | 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。 $t_{LEAD}(=RSPCK \text{ 遅延}) + 1/2 \times t_{SPcyc} > t_{SLCH} \text{ (min)}$ | SPCKD レジスタ SPCMD レジスタ |
| t_{SHSL} | 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。 $t_{TD}(=2 \times B\phi + \text{次アクセス遅延}) > t_{SHSL} \text{ (min)}$ | 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。 $f_C(\text{max}) > 1/t_{SPcyc}$ | 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。 $t_{LAG}(=SSL \text{ ネゲート遅延}) > t_{CHSH} \text{ (min)}$ | SSLND レジスタ SPCMD レジスタ |
| t_{DVCH} | 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。 $1/2 \times t_{SPcyc} - t_{OD}(\text{max}) > t_{DVCH} \text{ (min)}$ | |
| t_{CHDX} | データ入力ホールド時間 | データサンプルからマスタのデータ出力の停止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。 $t_{OH}(\text{min}) + 1/2 \times t_{SPcyc} > t_{CHDX} \text{ (min)}$ | |

表4 データ転送における SH7264 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。 $1/2 \times t_{SPcyc} - t_{CLQV}(\text{max}) > t_{SU}(\text{min})$ | |
| t_H | データ入力ホールド時間 | データサンプルからスレーブのデータ出力の停止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。 $t_{CLQX}(\text{min}) + 1/2 \times t_{SPcyc} > t_H(\text{min})$ | |

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ーモードでの SPI 動作が可能となります。

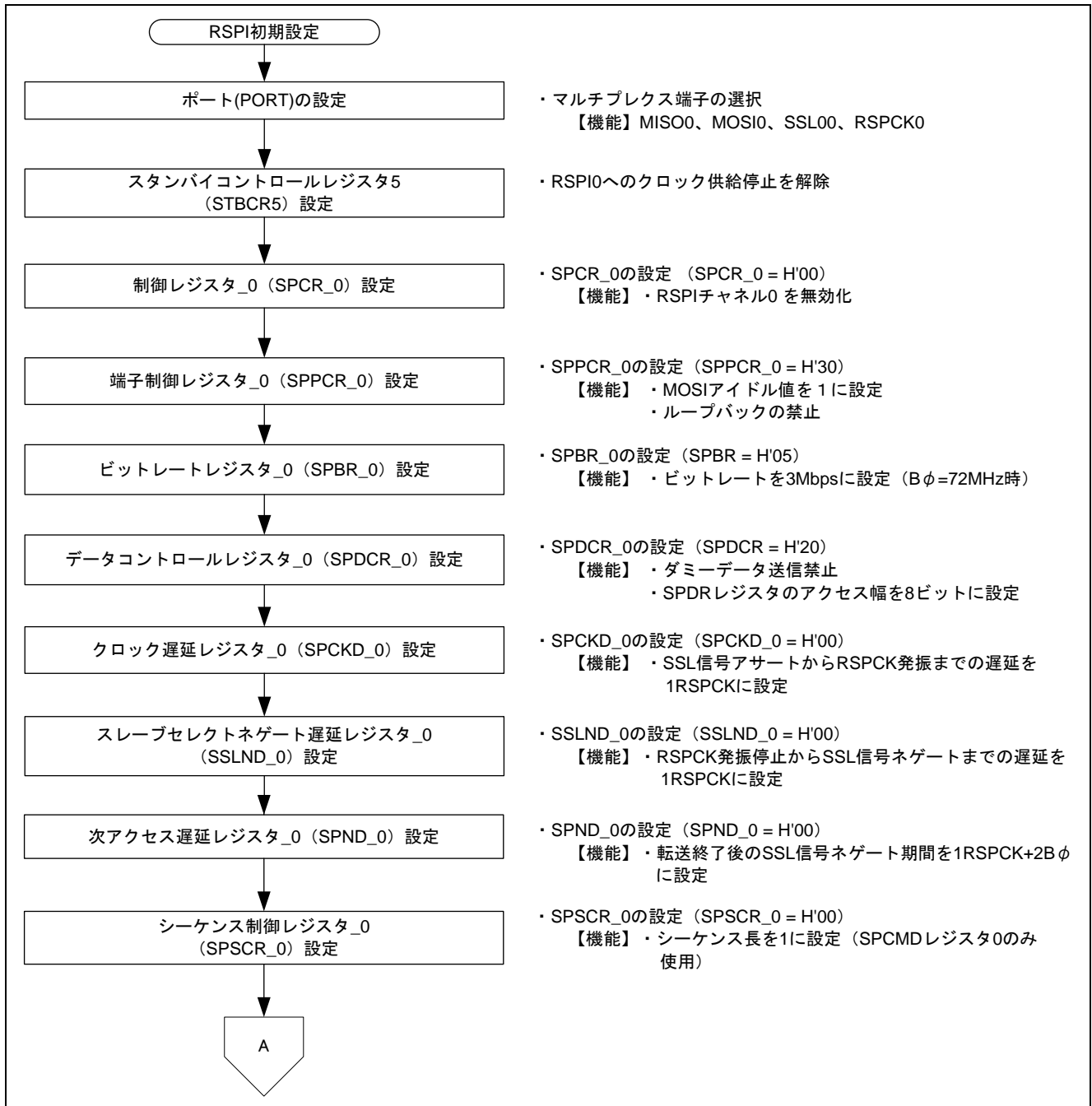


図3 参考プログラムの RSPI 初期設定フロー (1)

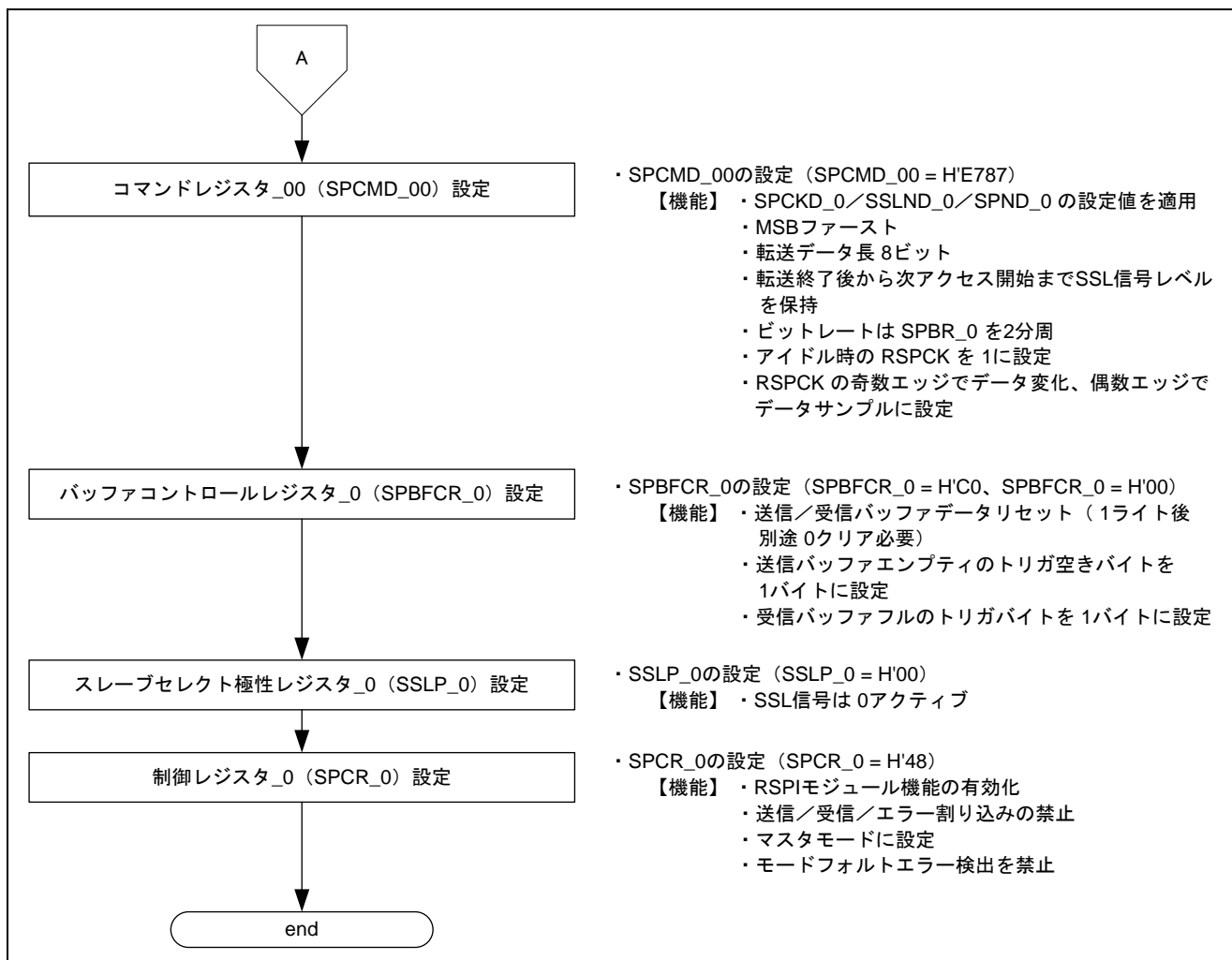


図4 参考プログラムのRSPI初期設定フロー (2)

2.4.2 コマンド転送例

EEPROM 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、本応用例ではルネサステクノロジ社製の HN58X25128I 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 HN58X25128I の主なコマンドを示します。

表5 HN58X25128I 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 |
|-------|-------|--------------|--------------------|----------------|
| WREN | H'06 | 0 | 0 | ライトイネーブル |
| WRDI | H'04 | 0 | 0 | ライトディセーブル |
| RDSR | H'05 | 0 | 1 | ステータスレジスタのリード |
| WRSR | H'01 | 0 | 1 | ステータスレジスタへのライト |
| READ | H'03 | 2 | 1 以上 ^{※1} | メモリリード |
| WRITE | H'02 | 2 | 1~64 ^{※2} | メモリライト |

【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 コマンドのシーケンス例を示します。

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 (2 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 信号をネゲートしてはいけません。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。

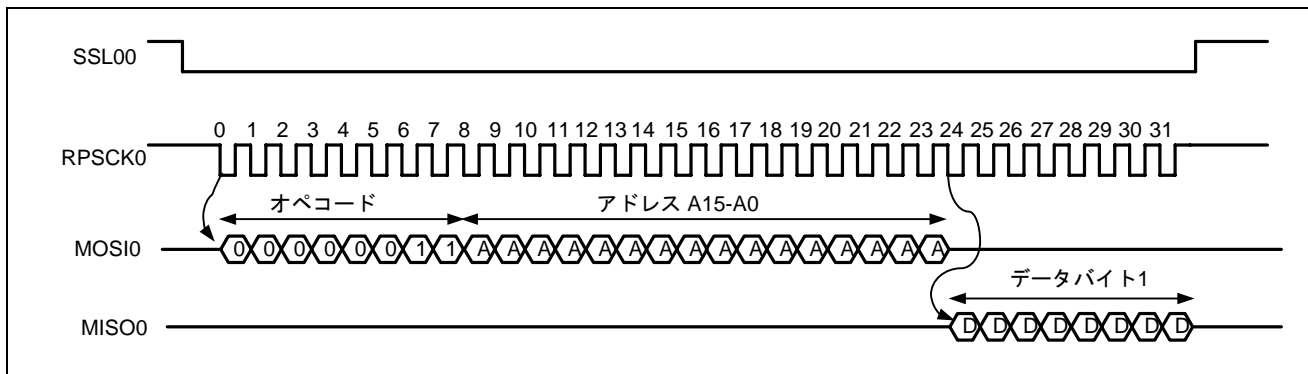


図5 リードコマンドシーケンス (H'03 オペコード)

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 にリード用のコマンド転送処理フローを示します。READ コマンドはこの処理フローに従います。図 7 にライト用のコマンド転送処理フローを示します。

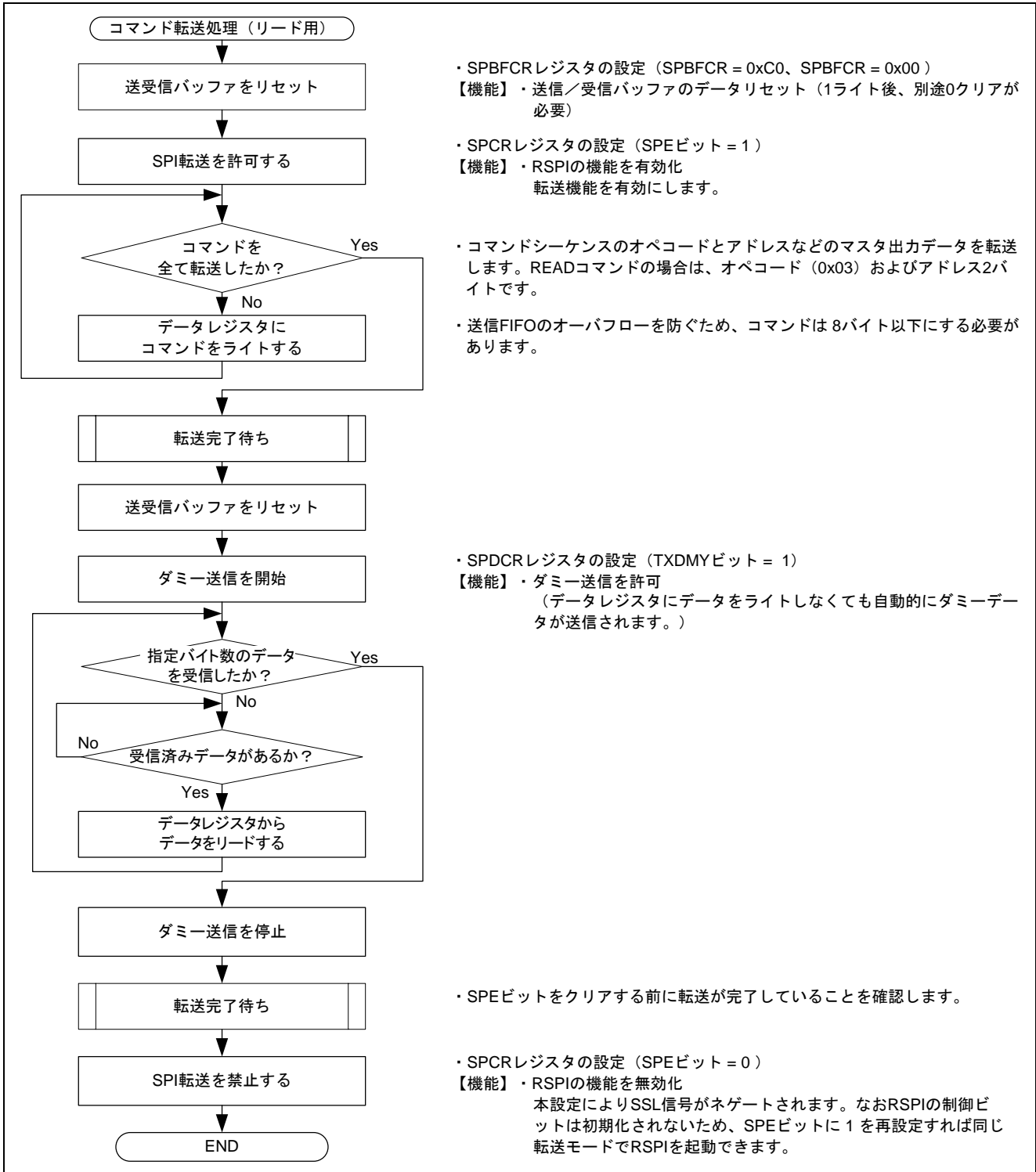
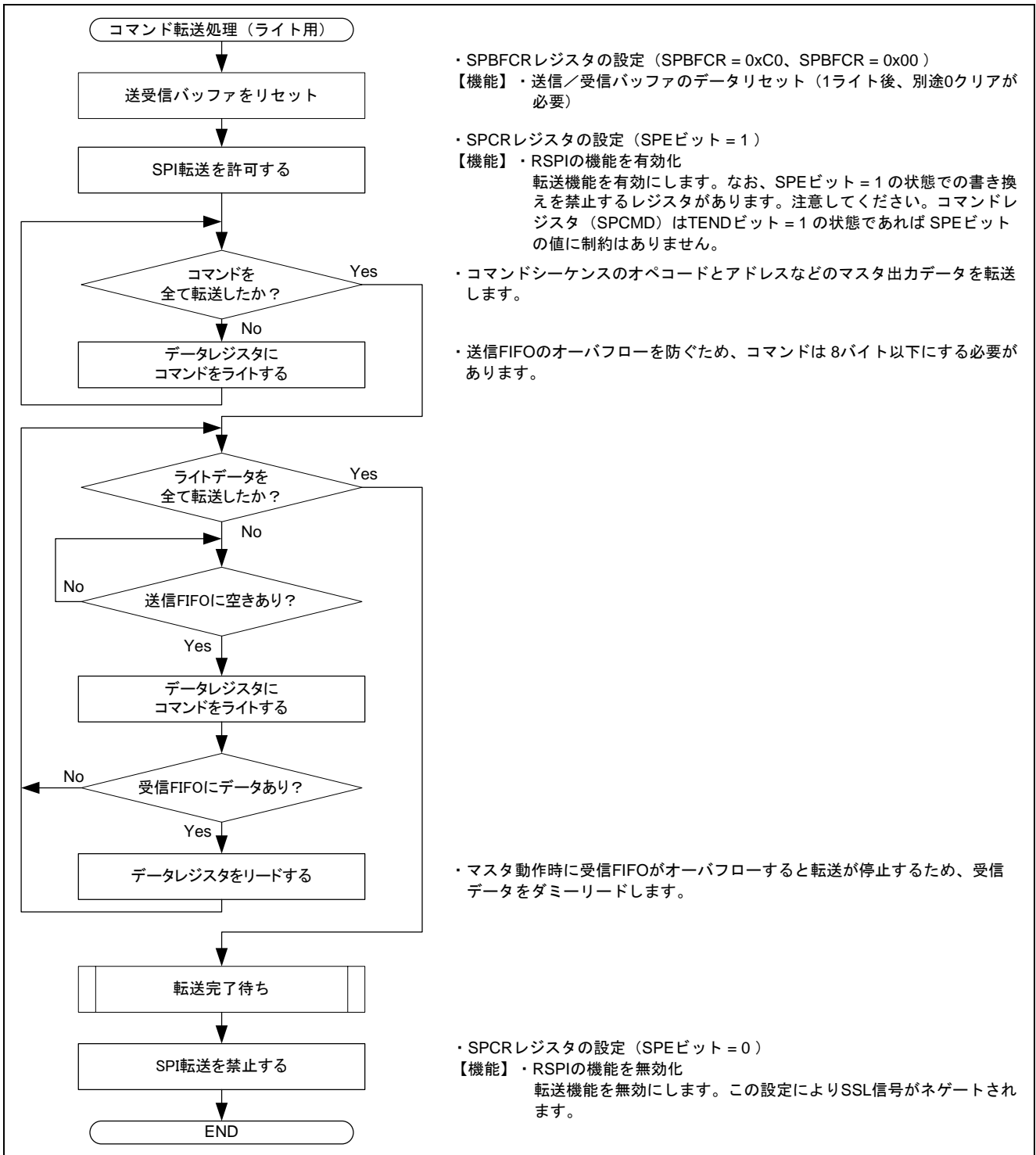


図6 参考プログラムのリード用コマンド転送フロー



- ・ SPBFCRレジスタの設定 (SPBFCR = 0xC0, SPBFCR = 0x00)
【機能】・送信/受信バッファのデータリセット (1ライト後、別途0クリアが必要)

- ・ SPCRレジスタの設定 (SPEビット = 1)
【機能】・RSPIの機能を有効化
転送機能を有効にします。なお、SPEビット = 1の状態での書き換えを禁止するレジスタがあります。注意してください。コマンドレジスタ (SPCMD) はTENDビット = 1の状態であればSPEビットの値に制約はありません。

- ・ コマンドシーケンスのオペコードとアドレスなどのマスタ出力データを転送します。

- ・ 送信FIFOのオーバーフローを防ぐため、コマンドは8バイト以下にする必要があります。

- ・ マスタ動作時に受信FIFOがオーバーフローすると転送が停止するため、受信データをダミーリードします。

- ・ SPCRレジスタの設定 (SPEビット = 0)
【機能】・RSPIの機能を無効化
転送機能を無効にします。この設定によりSSL信号がネゲートされます。

図7 参考プログラムのライト用コマンド転送フロー

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、EEPROM 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。

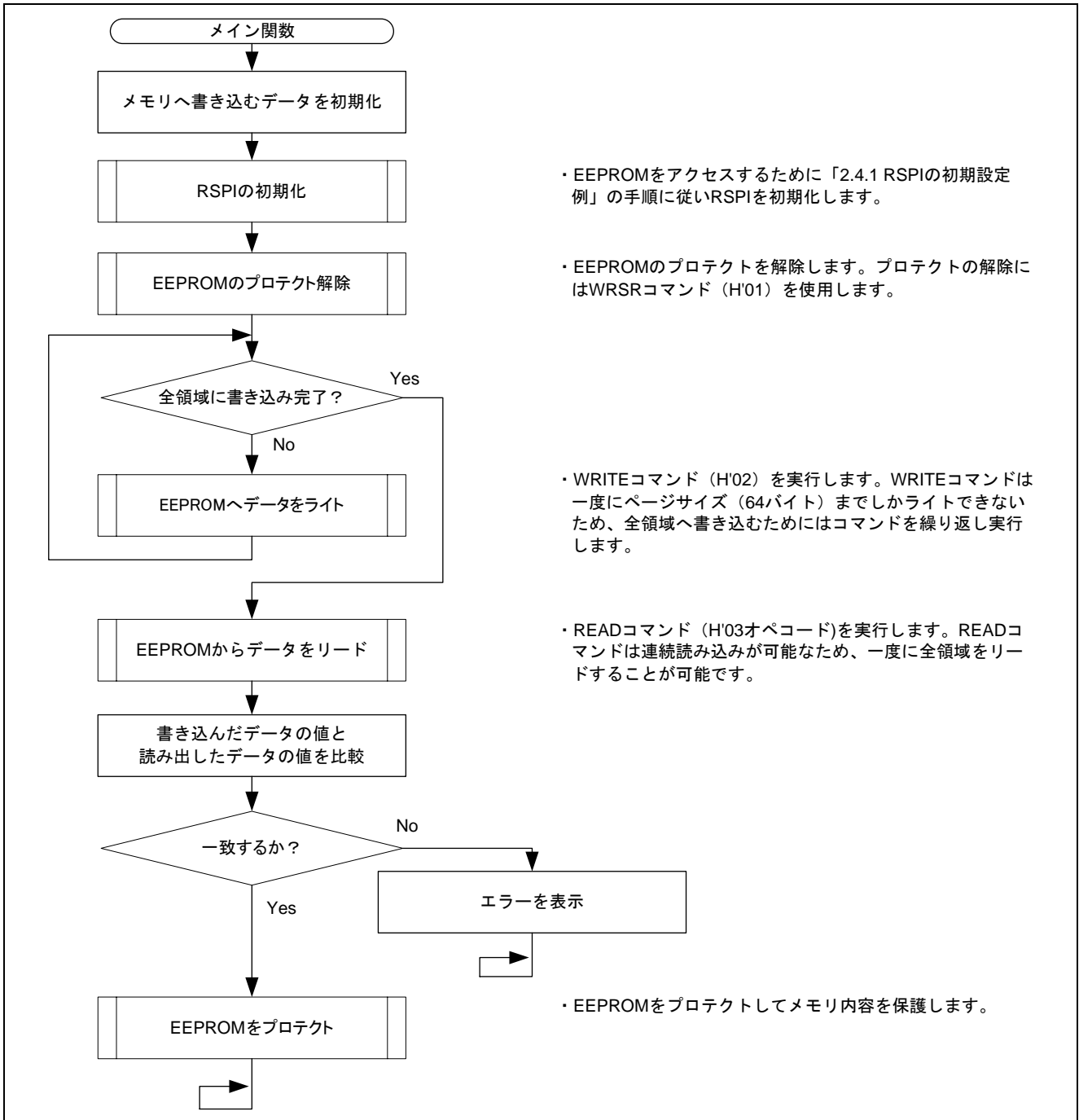


図8 参考プログラムのメイン関数フロー

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  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : main.c
5  *   Abstract    : ルネサスシリアルペリフェラルインタフェース EEPROM リードライト例
6  *   Version     : 1.00.00
7  *   Device      : SH7262/SH7264
8  *   Tool-Chain  : High-performance Embedded Workshop (Version 4.04.01)
9  *                : C/C++ compiler package for the SuperH RISC engine family
10 *                :                               (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Feb.20,2009 Ver.1.00.00
29 * "FILE COMMENT END"*****
30 #include <stdio.h>
31 #include "eeprom.h"
32
33 /* ==== マクロ定義 ==== */
34 #define TOP_ADDRESS    0                /* EEPROM の先頭アドレス */
35
36 /* ==== 関数プロトタイプ宣言 ==== */
37 void main(void);
38
39 /* ==== 変数定義 ==== */
40 #pragma section DEBUG_32K_BYTES
41 static unsigned char data[EEP_MEM_SIZE];
42 static unsigned char rbuf[EEP_MEM_SIZE];
43 #pragma section
44

```

3.2 サンプルプログラムリスト"main.c" (2)

```

45  /*"FUNC COMMENT"*****
46  * ID      :
47  * Outline : EEPROM アクセス メイン処理
48  *-----
49  * Include :
50  *-----
51  * Declaration : void main(void);
52  *-----
53  * Function  : EEPROM へのライト処理とリード処理を行います。
54  *           : RSPI チャンネル0 を初期化後、先頭から全領域にデータを書き込みます。
55  *           : 結果は読み出して確認します。
56  *-----
57  * Argument  : void
58  *-----
59  * Return Value: void
60  *"FUNC COMMENT END"*****/
61 void main(void)
62 {
63     int i;
64     static unsigned long addr;
65     /* ==== データ初期化 ==== */
66     for(i = 0; i < EEP_MEM_SIZE; i++){
67         data[i] = i % 100;
68         rbuf[i] = 0;
69     }
70     /* ==== RSPI の初期化 ==== */
71     eep_init_eeeprom();
72
73     /* ==== EEPROM のプロテクト解除 ==== */
74     eep_protect_ctrl( EEP_REQ_UNPROTECT );
75
76     /* ==== データライト ==== */
77     for(addr = TOP_ADDRESS; addr < (TOP_ADDRESS + EEP_MEM_SIZE); addr += EEP_PAGE_SIZE ){
78         eep_byte_write( addr, data+addr, EEP_PAGE_SIZE );
79     }
80     /* ==== データリード ==== */
81     eep_byte_read( TOP_ADDRESS, rbuf, EEP_MEM_SIZE );
82
83     /* ==== ベリファイチェック ==== */
84     for(i = 0; i < EEP_MEM_SIZE; i++){
85         if( data[i] != rbuf[i] ){
86             puts("Error: verify error\n");
87             fflush(stdout);
88             while(1);
89         }
90     }
91     /* ==== EEPROM のプロテクト ==== */
92     eep_protect_ctrl( EEP_REQ_PROTECT );
93     while(1){
94         /* loop */
95     }
96 }
97 /* End of File */

```


3.3 サンプルプログラムリスト"eeprom.c" (1)

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : eeprom.c
5  *   Abstract    : ルネサスシリアルペリフェラルインタフェース EEPROM リードライト例
6  *   Version     : 1.00.00
7  *   Device      : SH7262/SH7264
8  *   Tool-Chain  : High-performance Embedded Workshop (Version 4.04.01)
9  *                : C/C++ compiler package for the SuperH RISC engine family
10 *                :
11 *                : (Ver.9.02 Release00).
12 *   OS          : None
13 *   H/W Platform: M3A-HS64G50(CPU board)
14 *   Disclaimer  :
15 *               <注意事項>
16 *               本サンプルプログラムはすべて参考資料であり、
17 *               その動作を保証するものではありません。
18 *               本サンプルプログラムはお客様のソフトウェア開発時の
19 *               技術参考資料としてご利用ください。
20 *
21 *   The information described here may contain technical inaccuracies or
22 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
23 *   assume no responsibility for any damage, liability, or other loss rising
24 *   from these inaccuracies or errors.
25 *
26 *   Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
27 *   AND Renesas Solutions Corp. All Rights Reserved
28 *
29 *   History     : Feb.20,2009 Ver.1.00.00
30 * "FILE COMMENT END"***** /
31 #include <stdio.h>
32 #include <machine.h>
33 #include "iodefine.h"
34 #include "eeprom.h"
35
36 /* ==== マクロ定義 ==== */
37 #define EEPROMCMD_WRITE_ENABLE    0x06
38 #define EEPROMCMD_WRITE_DISABLE  0x04
39 #define EEPROMCMD_READ_STATUS     0x05
40 #define EEPROMCMD_WRITE_STATUS    0x01
41 #define EEPROMCMD_READ_ARRAY      0x03
42 #define EEPROMCMD_WRITE_ARRAY     0x02
43 #define UNPROTECT_WR_STATUS       0x00
44 #define PROTECT_WR_STATUS         0x0C
45 #define EEP_BUSY_BIT              0x01

```

3.4 サンプルプログラムリスト"eeprom.c" (2)

```

46  /* ==== 関数プロトタイプ宣言 ==== */
47  /*** Local function ***/
48  static void write_enable(void);
49  static void write_disable(void);
50  static void busy_wait(void);
51  static unsigned char read_status(void);
52  static void write_status(unsigned char status);
53  static void io_init_rsipi(void);
54  static void io_cmd_exe(unsigned char *ope, int ope_sz, unsigned char *data, int data_sz);
55  static void io_cmd_exe_rdmode(unsigned char *ope, int ope_sz, unsigned char *rd, int rd_sz);
56  static void io_wait_tx_end(void);
57
58  /* ==== 変数定義 ==== */
59
60  /*"FUNC COMMENT"*****
61  * ID      :
62  * Outline : EEPROMの初期化
63  *-----
64  * Include :
65  *-----
66  * Declaration : void eep_init_eeprom(void);
67  *-----
68  * Function    : EEPROMにアクセスするための初期化を行います。
69  *              : ルネサスシリアルペリフェラルインタフェース(RSPI)のチャンネル0を初期化
70  *              : します。
71  *-----
72  * Argument   : void
73  *-----
74  * Return Value: void
75  *"FUNC COMMENT END"*****/
76  void eep_init_eeprom(void)
77  {
78      /* ==== RSPI0の初期化 ==== */
79      io_init_rsipi();
80  }
81

```

3.5 サンプルプログラムリスト"eeprom.c" (3)

```

82  /*"FUNC COMMENT"*****
83  * ID      :
84  * Outline : プロテクト操作
85  *-----
86  * Include :
87  *-----
88  * Declaration : void eep_protect_ctrl(enum eep_req req);
89  *-----
90  * Function   : EEPROM のプロテクト設定または解除を行います。
91  *             : 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
92  *             : EEPROM の仕様によって異なります。
93  *-----
94  * Argument   : enum eep_req req ; I : EEP_REQ_UNPROTECT -> 全領域プロテクト解除
95  *             :                               EEP_REQ_PROTECT  -> 全領域プロテクト
96  *-----
97  * Return Value: void
98  *"FUNC COMMENT END"*****/
99  void eep_protect_ctrl(enum eep_req req)
100 {
101     if( req == EEP_REQ_UNPROTECT ){
102         write_status( UNPROTECT_WR_STATUS); /* 全領域プロテクト解除 */
103     }
104     else{
105         write_status( PROTECT_WR_STATUS ); /* 全領域プロテクト */
106     }
107 }
108

```

3.6 サンプルプログラムリスト"eeprom.c" (4)

```

109  /*"FUNC COMMENT"*****
110  * ID      :
111  * Outline : データライト
112  *-----
113  * Include :
114  *-----
115  * Declaration : void eep_byte_write(unsigned long addr, unsigned char *buf, int size);
116  *-----
117  * Function   : EEPROMに指定データを書き込みます。
118  *             : 書き込む前にはライトイネーブルコマンドを発行する必要があります。
119  *             : また書き込み後はEEPROMのステータスを確認しビジー状態が解除
120  *             : されたことを確認してください。
121  *             : 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。
122  *-----
123  * Argument   : unsigned long addr ; I : ライトするEEPROMのアドレス
124  *             : unsigned char *buf ; I : ライトデータを格納するバッファのアドレス
125  *             : int size           ; I : ライトするバイト数
126  *-----
127  * Return Value: void
128  *"FUNC COMMENT END"*****/
129  void eep_byte_write(unsigned long addr, unsigned char *buf, int size)
130  {
131      unsigned char cmd[3];
132
133      cmd[0] = EEPROMCMD_WRITE_ARRAY;
134      cmd[1] = (unsigned char)((addr >> 8) & 0xff);
135      cmd[2] = (unsigned char)(addr      & 0xff);
136      write_enable();
137      io_cmd_exe(cmd, 3, buf, size);
138      busy_wait();
139  }
140

```

3.7 サンプルプログラムリスト"eeprom.c" (5)

```

141  /*"FUNC COMMENT"*****
142  * ID      :
143  * Outline : データリード
144  *-----
145  * Include :
146  *-----
147  * Declaration : void eep_byte_read(unsigned long addr, unsigned char *buf, int size);
148  *-----
149  * Function    : EEPROMを指定バイト数だけリードします。
150  *-----
151  * Argument    : unsigned long addr ; I : リードする EEPROM のアドレス
152  *              : unsigned char *buf ; I : リードデータを格納するバッファのアドレス
153  *              : int size           ; I : リードするバイト数
154  *-----
155  * Return Value: void
156  *"FUNC COMMENT END"*****/
157 void eep_byte_read(unsigned long addr, unsigned char *buf, int size)
158 {
159     unsigned char cmd[3];
160
161     cmd[0] = EEPROMCMD_READ_ARRAY;
162     cmd[1] = (unsigned char)((addr >> 8) & 0xff);
163     cmd[2] = (unsigned char)(addr & 0xff);
164     io_cmd_exe_rdmode(cmd, 3, buf, size);
165 }
166 /*"FUNC COMMENT"*****
167 * ID      :
168 * Outline : 書き込み許可
169 *-----
170 * Include :
171 *-----
172 * Declaration : static void write_enable(void);
173 *-----
174 * Function    : ライトイネーブルコマンドを発行して、EEPROMへの
175 *              : ライト動作を許可します。
176 *-----
177 * Argument    : void
178 *-----
179 * Return Value: void
180 *"FUNC COMMENT END"*****/
181 static void write_enable(void)
182 {
183     unsigned char cmd[1];
184     cmd[0] = EEPROMCMD_WRITE_ENABLE;
185     io_cmd_exe(cmd, 1, NULL, 0);
186 }

```

3.8 サンプルプログラムリスト"eeprom.c" (6)

```

187  /*"FUNC COMMENT"*****
188  * ID      :
189  * Outline  : 書き込み禁止
190  *-----
191  * Include  :
192  *-----
193  * Declaration : static void write_disable(void);
194  *-----
195  * Function   : ライトディスエーブルコマンドを発行して、EEPROM への
196  *             : ライト動作を禁止します。
197  *-----
198  * Argument   : void
199  *-----
200  * Return Value: void
201  *"FUNC COMMENT END"*****/
202  static void write_disable(void)
203  {
204      unsigned char cmd[1];
205      cmd[0] = EEPROMCMD_WRITE_DISABLE;
206      io_cmd_exe(cmd, 1, NULL, 0);
207  }
208
209  /*"FUNC COMMENT"*****
210  * ID      :
211  * Outline  : ビジー待ち
212  *-----
213  * Include  :
214  *-----
215  * Declaration : static void busy_wait(void);
216  *-----
217  * Function   : EEPROM のステータスがビジー状態の場合は内部で
218  *             : ループします。
219  *-----
220  * Argument   : void
221  *-----
222  * Return Value: void
223  *"FUNC COMMENT END"*****/
224  static void busy_wait(void)
225  {
226      while ((read_status() & EEP_BUSY_BIT) != 0) { /* RDY/BSY */
227          /* serial flash is busy */
228      }
229  }
230

```

3.9 サンプルプログラムリスト"eeprom.c" (7)

```

231  /*"FUNC COMMENT"*****
232  * ID      :
233  * Outline : ステータスリード
234  *-----
235  * Include :
236  *-----
237  * Declaration : static unsigned char read_status(void);
238  *-----
239  * Function   : EEPROMのステータスをリードします。
240  *-----
241  * Argument   : void
242  *-----
243  * Return Value: ステータスレジスタの値
244  *"FUNC COMMENT END"*****/
245  static unsigned char read_status(void)
246  {
247      unsigned char buf;
248      unsigned char cmd[1];
249
250      cmd[0] = EEPROMCMD_READ_STATUS;
251      io_cmd_exe_rdmode(cmd, 1, &buf, 1);
252      return buf;
253  }
254
255  /*"FUNC COMMENT"*****
256  * ID      :
257  * Outline : ステータスライト
258  *-----
259  * Include :
260  *-----
261  * Declaration : static void write_status(unsigned char status);
262  *-----
263  * Function   : EEPROMのステータスをライトします。
264  *-----
265  * Argument   : unsigned char status ; I : status register value
266  *-----
267  * Return Value: void
268  *"FUNC COMMENT END"*****/
269  static void write_status(unsigned char status)
270  {
271      unsigned char cmd[2];
272
273      cmd[0] = EEPROMCMD_WRITE_STATUS;
274      cmd[1] = status;
275
276      write_enable();
277      io_cmd_exe(cmd, 2, NULL, 0);
278      busy_wait();
279  }

```

3.10 サンプルプログラムリスト"eeprom.c" (8)

```

280  /*"FUNC COMMENT"*****
281  * ID      :
282  * Outline : RSPI の初期化
283  *-----
284  * Include :
285  *-----
286  * Declaration : static void io_init_rsipi(void);
287  *-----
288  * Function   : ルネサスシリアルペリフェラルインタフェースのチャンネル0を初期化します。
289  *             : マスタモードに設定し、EEPROMの仕様に合わせた転送設定を行います。
290  *-----
291  * Argument   : void
292  *-----
293  * Return Value: void
294  *"FUNC COMMENT END"*****/
295  static void io_init_rsipi(void)
296  {
297      /* ==== PORT ==== */
298      PORT.PFCR3.BIT.PF12MD = 3; /* PF12:MISO0 */
299      PORT.PFCR2.BIT.PF11MD = 3; /* PF11:MOSI0 */
300      PORT.PFCR2.BIT.PF10MD = 3; /* PF10:SSL00 */
301      PORT.PFCR2.BIT.PF9MD  = 3; /* PF9:RSPCK0 */
302
303      /* ==== CPG ==== */
304      CPG.STBCR5.BIT.MSTP51 = 0; /* RSPI0 active */
305
306      /* ==== RSPI ==== */
307      RSPI0.SPCR.BYTE = 0x00; /* RSPIチャンネル0を動作禁止 */
308      RSPI0.SPPCR.BYTE = 0x30; /* MOSIアイドル固定値 = 1 */
309      RSPI0.SPBR.BYTE = 0x05; /* ベースのビットレートを3MHzに設定(BΦ=72MHz) */
310      RSPI0.SPDCR.BYTE = 0x20; /* ダミーデータ送信禁止 */
311                                     /* SPDRレジスタのアクセス幅:8ビット */
312      RSPI0.SPCKD.BYTE = 0x00; /* RSPCK遅延:1 RSPCK */
313      RSPI0.SSLND.BYTE = 0x00; /* SSLネゲート遅延:1 RSPCK */
314      RSPI0.SPND.BYTE = 0x00; /* 次アクセス遅延:1 RSPCK + 2 BΦ */
315      RSPI0.SPSCR.BYTE = 0x00; /* シーケンス長:1 (SPCMD0のみ使用) */
316      RSPI0.SPCMD0.WORD = 0xE787; /* MSBファースト */
317                                     /* データ長:8bit */
318                                     /* 転送終了後もSSL信号レベルを保持する */
319                                     /* ビットレート:ベースビットレートの2分周 */
320                                     /* アイドル時のRSPCK:1 */
321                                     /* 奇数エッジでデータ変化、偶数エッジでデータサンプル */
322      RSPI0.SPBFCR.BYTE = 0xC0; /* 送受信バッファのデータリセット許可 */
323      RSPI0.SPBFCR.BYTE = 0x00; /* 送受信バッファのデータリセット禁止 */
324                                     /* 送信バッファのトリガ:1バイト以上の空き */
325                                     /* 受信バッファのトリガ:1バイト以上の受信 */
326      RSPI0.SSLP.BYTE = 0x00; /* SSLP = b'0 SSL signal 0-active */
327      RSPI0.SPCR.BYTE = 0x48; /* マスタモード */
328                                     /* 割り込み禁止 */
329                                     /* RSPIチャンネル0の動作許可 */
330  }

```


3.11 サンプルプログラムリスト"eeprom.c" (9)

```

331  /*"FUNC COMMENT"*****
332  * ID      :
333  * Outline : コマンド実行(リードデータなし)
334  *-----
335  * Include :
336  *-----
337  * Declaration : static void io_cmd_exe(unsigned char *ope, int ope_sz,
338  *      :      unsigned char *data,int data_sz)
339  *-----
340  * Function  : 指定されたコマンドを実行します。
341  *      :      : 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。
342  *      :      : ope_sz は 0~8 のいずれかの値を設定してください。
343  *      :      : data_sz は 0~256 のいずれかの値を設定してください。
344  *-----
345  * 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
346  *      :      : int ope_sz      ; I : オペコード部とアドレス部のバイト数
347  *      :      : unsigned char *data; I : 送信するデータ部の先頭アドレス
348  *      :      : int data_sz    ; I : データ部のバイト数
349  *-----
350  * Return Value: void
351  *"FUNC COMMENT END"*****/
352  static void io_cmd_exe(unsigned char *ope, int ope_sz, unsigned char *data, int data_sz)
353  {
354      unsigned char tmp;
355
356      /* ==== バッファリセット ==== */
357      RSP10.SPBFCR.BYTE = 0xC0u;
358      RSP10.SPBFCR.BYTE = 0x00u;
359
360      /* ---- SPI 転送許可 ---- */
361      RSP10.SPCR.BIT.SPE = 1;
362
363      /* ==== MOSI(コマンド、アドレス、ライトデータ) ==== */
364      while(ope_sz--){
365          RSP10.SPDR.BYTE = *ope++;      /* コマンドは 8 バイト以下とする */
366      }
367      while(data_sz--){
368          while( RSP10.SPSR.BIT.SPTEF == 0 ){
369              /* wait */
370          }
371          RSP10.SPDR.BYTE = *data++;
372          if( RSP10.SPSR.BIT.SPRF == 1 ){
373              tmp = RSP10.SPDR.BYTE;      /* オーバフロー防止のためのダミーリード */
374          }
375      }
376      io_wait_tx_end();      /* 完了待ち */
377
378      /* ---- SPI 転送終了 (SSL ネゲート) ---- */
379      RSP10.SPCR.BIT.SPE = 0;
380  }
381

```


3.13 サンプルプログラムリスト"eeprom.c" (11)

```

435  /*"FUNC COMMENT"*****
436  * ID      :
437  * Outline : 送信完了待ち
438  *-----
439  * Include :
440  *-----
441  * Declaration : static void io_wait_tx_end(void);
442  *-----
443  * Function   : 送信完了が確認できるまで内部でループします。
444  *-----
445  * Argument   : void
446  *-----
447  * Return Value: void
448  /*"FUNC COMMENT END"*****/
449  static void io_wait_tx_end(void)
450  {
451      while(RSPI0.SPSR.BIT.TEND == 0){
452          /* wait */
453      }
454  }
455
456  /* End of File */

```

3.14 サンプルプログラムリスト"eeprom.h" (1)

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  *
3  *   System Name : SH7264 Sample Program
4  *   File Name   : eeprom.h
5  *   Abstract    : ルネサスシリアルペリフェラルインタフェース EEPROM リードライト例
6  *   Version     : 1.00.00
7  *   Device      : SH7262/SH7264
8  *   Tool-Chain  : High-performance Embedded Workshop (Version 4.04.01)
9  *               : C/C++ compiler package for the SuperH RISC engine family
10 *               :                               (Ver.9.02 Release00).
11 *   OS          : None
12 *   H/W Platform: M3A-HS64G50(CPU board)
13 *   Disclaimer  :
14 *               <注意事項>
15 *               本サンプルプログラムはすべて参考資料であり、
16 *               その動作を保証するものではありません。
17 *               本サンプルプログラムはお客様のソフトウェア開発時の
18 *               技術参考資料としてご利用ください。
19 *
20 *   The information described here may contain technical inaccuracies or
21 *   typographical errors. Renesas Technology Corporation and Renesas Solutions
22 *   assume no responsibility for any damage, liability, or other loss rising
23 *   from these inaccuracies or errors.
24 *
25 *   Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
26 *   AND Renesas Solutions Corp. All Rights Reserved
27 *
28 *   History     : Feb.20,2009 Ver.1.00.00
29 *"FILE COMMENT END"*****
30
31 #ifndef _EEPROM_H_
32 #define _EEPROM_H_
33
34 /* ==== マクロ定義 ==== */
35 #define EEP_PAGE_SIZE      64          /* EEPROM のページサイズ */
36 #define EEP_MEM_SIZE      0x4000     /* EEPROM のサイズ(16KB) */
37 enum eep_req{
38     EEP_REQ_PROTECT = 0,             /* プロテクト要求 */
39     EEP_REQ_UNPROTECT          /* プロテクト解除要求 */
40 };
41 /* ==== 関数プロトタイプ宣言 ==== */
42 void eep_init_serial_flash(void);
43 void eep_protect_ctrl(enum eep_req req);
44 void eep_byte_write(unsigned long addr, unsigned char *buf, int size);
45 void eep_byte_read(unsigned long addr, unsigned char *buf, int size);
46
47 /* ==== 変数定義 ==== */
48
49 #endif /* _EEPROM_H_ */
50 /* End of File */
51

```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A/SH-2A-FPU ソフトウェアマニュアル Rev.3.00
(最新版をルネサス テクノロジホームページから入手してください。)
- ハードウェアマニュアル
SH7262 グループ、SH7264 グループ ハードウェアマニュアル Rev.1.00
(最新版をルネサス テクノロジホームページから入手してください。)

ホームページとサポート窓口

ルネサステクノロジホームページ

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

お問合せ先

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

csc@renesas.com

改訂記録

| Rev. | 発行日 | 改訂内容 | |
|------|------------|------|------|
| | | ページ | ポイント |
| 1.00 | 2009.03.10 | | 初版発行 |

すべての商標および登録商標は、それぞれの所有者に帰属します。

本資料ご利用に際しての留意事項

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

D039444