

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。

# SH7147

## Synchronous Serial Communication Unit マスタ送受信 (SPI バス EEPROM のライト/リード)

### 1. 要約

本アプリケーションノートは、Synchronous Serial Communication Unit モジュールを使用した 4 線式シリアルインタフェースのマスタ送受信動作について述べており、ユーザソフトウェア設計の際のご参考として役立ててください。

尚、本アプリケーションノートに記載されているプログラムの動作確認は行っておりますが、実際にご使用になる場合は、必ず動作確認の上ご使用くださいますようお願いいたします。

【注】 本アプリケーションノートで使用している EEPROM (HN58X2564I) は民生用途に使用されることを意図した製品です。

### 2. はじめに

#### 2.1 仕様

- Synchronous Serial Communication Unit マスタ送信によって、1 バイトのデータを SPI バス上の EEPROM へ書き込みます。
- Synchronous Serial Communication Unit マスタ受信によって、1 バイトのデータを SPI バス上の EEPROM から読み出します。

#### 2.2 使用機能

- Synchronous Serial Communication Unit

#### 2.3 適用条件

- マイコン : SH7147 (R5F71474AK64FPV)
- 動作周波数 : 内部クロック 64MHz  
: バスクロック 32MHz  
: 周辺クロック 32MHz  
: MTU2S クロック 64MHz  
: MTU2 クロック 32MHz
- C コンパイラ : ルネサステクノロジ製 Ver. V.9.1.0.0  
SuperH RISC engine ファミリ C/C++ コンパイラパッケージ
- コンパイルオプション : HEW でのデフォルト設定

### 3. 応用例の説明

- SH7147 の Synchronous Serial Communication Unit モジュールを使用して、4 線式シリアル方式の EEPROM (HN58X2564I、64k bit、8k word × 8bit) へ 1 バイトのデータの書き込み、および読み出しを行います。
- 接続は SH7147 をマスタデバイスとした、シングルマスタ構成とします。
- データ転送クロックは、2MHz とします。
- 図 1 に SH7147 と EEPROM の接続を示し、表 1 に Synchronous Serial Communication Unit および EEPROM の端子機能を示します。
- 表 2 に Synchronous Serial Communication Unit の設定概要を示します。
- 表 3 に本応用例で使用する EEPROM の命令コードを示します。
- 表 4 に本応用例で使用する EEPROM のステータスレジスタの内容を示します。

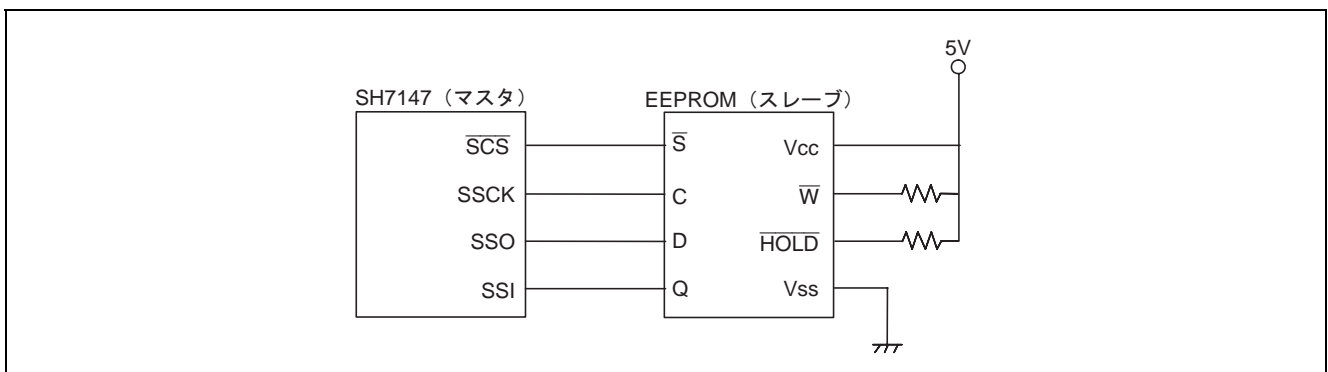


図1 SH7147 と EEPROM 接続図

表1 端子機能表

端子名	端子機能	Synchronous Serial Communication Unit/EEPROM
SCS	Synchronous Serial Communication Unit チップセレクト入出力端子 (本応用例では、出力設定)	Synchronous Serial Communication Unit 端子
SSCK	Synchronous Serial Communication Unit クロック入出力端子 (本応用例では、出力設定)	
SSI	Synchronous Serial Communication Unit データ入出力端子 (本応用例では、入力設定)	
SSO	Synchronous Serial Communication Unit データ入出力端子 (本応用例では、出力設定)	
S-bar	EEPROM チップセレクト入力端子	EEPROM 端子
C	EEPROM クロック入力端子	
D	EEPROM データ入力端子	
Q	EEPROM データ出力端子	
Vcc	EEPROM 電源端子	
W-bar	EEPROM ライトプロテクト端子 (本応用例では、High 固定)	
HOLD	EEPROM ホールド端子 (本応用例では、High 固定)	
Vss	EEPROM グランド端子	

表2 Synchronous Serial Communication Unit 設定概要

設定項目	設定内容
動作モード	マスタモード
データ入力端子	標準モード (データ入力端子とデータ出力端子の2端子を使用)
転送クロック	2MHz (Pφ=32MHz)
データのビット数	8ビット (EEPROM への書き込みコードの送信時) 16ビット (EEPROM からのステータスフラグの読み出し時) 32ビット (EEPROM へのデータ書き込み時) 32ビット (EEPROM からデータ読み込み時)
MSB/LSB ファースト	MSB ファースト
TENDビットのセットタイミング	最後尾ビット送信後

表3 EEPROM の命令コード

コード名	動作内容	コードフォーマット
WREN	EEPROM を書き込み許可状態へ設定	0000 0110
WRDI	EEPROM を書き込み禁止状態へ設定	0000 0100
RDSR	EEPROM のステータスレジスタの読み出し	0000 0101
WRSR	EEPROM のステータスレジスタへの書き込み	0000 0001
READ	EEPROM の格納データの読み出し	0000 0011
WRITE	EEPROM へ格納データの書き込み	0000 0010

表4 EEPROM ステータスレジスタ

レジスタ名	ビット名	ビット	機能
EEPROM ステータスレジスタ	SRWD	7	ライトプロテクト ( $\bar{W}$ ) 信号と関連して動作します。 $\bar{W}$ 信号との組み合わせで、ハードウェアプロテクト状態にセットできます。 ただし、本応用例では、 $\bar{W}$ 信号は High 固定なので、ハードウェアプロテクト状態にはなりません。 (詳細は EEPROM のマニュアルを参照してください。)
	リザーブ	6	
	リザーブ	5	
	リザーブ	4	
	BP[1:0]	3-2	ソフトウェアプロテクトする領域を設定します。 00: ソフトウェアプロテクトはしない 01: ソフトウェアプロテクト領域は H'1800 ~ H'1FFF 10: ソフトウェアプロテクト領域は H'1000 ~ H'1FFF 11: ソフトウェアプロテクト領域は H'0000 ~ H'1FFF
	WEL	1	0: データの書き込み・ステータスレジスタへの書き込みは許可されていません。 1: データの書き込み・ステータスレジスタへの書き込みが許可されています。
	WIP	0	0: 書き込み動作終了 1: 書き込み動作中

3.1 使用機能の動作概要

Synchronous Serial Communication Unit には、マスタモード(本 LSI からクロック出力)とスレーブモード(外部デバイスからクロック入力)があります。また、クロック極性とクロック位相の異なるデバイス間との同期シリアル通信が可能です。図 2 に Synchronous Serial Communication Unit モジュールのブロック図を示し、表 5 に各レジスタの説明を示します。

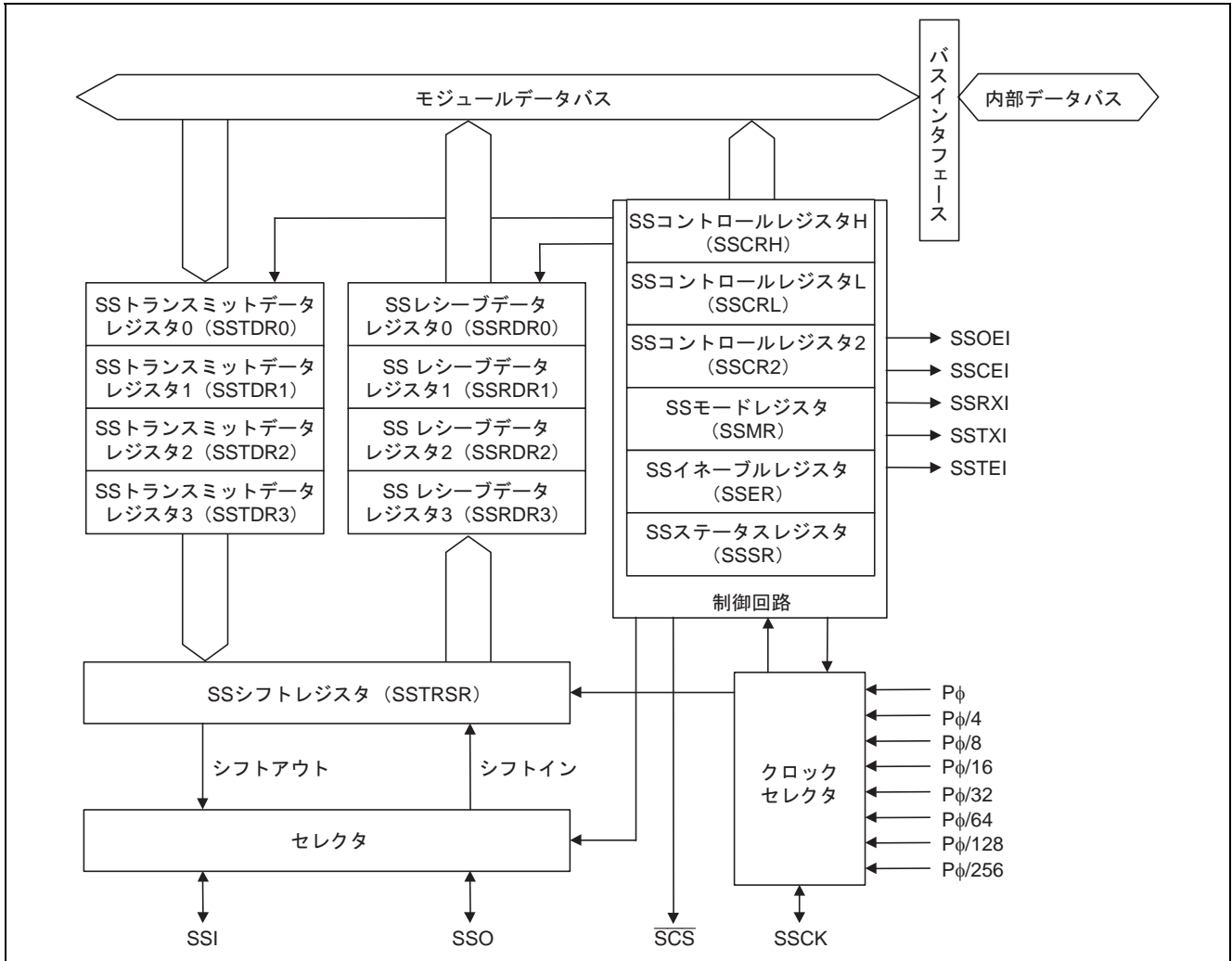


図2 Synchronous Serial Communication Unit のブロック図

表5 Synchronous Serial Communication Unit レジスタ構成

レジスタ名	略称	機能
SS コントロールレジスタ H	SSCRH	マスタ/スレーブモードの選択、入出力端子モードの選択、SSO 端子の出力値選択、SCS 端子機能選択を行います。
SS コントロールレジスタ L	SSCRL	フラグクリアモード、動作モード、ソフトウェアリセット、送受信データのデータ長を選択します。
SS モードレジスタ	SSMR	MSB/LSB ファースト、クロック極性、クロック位相、転送クロックレートを選択します。
SS イネーブルレジスタ	SSER	トランスミットイネーブル、レシーブイネーブル、割り込み要求イネーブルを設定します。
SS ステータスレジスタ	SSSR	各種割り込みのステータスフラグレジスタです。
SS コントロールレジスタ 2	SSCR2	SCS 端子のアサートタイミング、SSO 端子のデータ出力タイミング、TEND ビットのセットタイミングを設定します。
SS トランスミットデータレジスタ 0~3	SSTDR0~3	送信データを格納するための 8 ビットレジスタです。
SS レシーブデータレジスタ 0~3	SSRDR0~3	受信データを格納するための 8 ビットレジスタです。
SS シフトレジスタ	SSTRSR	シリアルデータを送受信するためのシフトレジスタです。

【注】 各レジスタの詳細機能については、SH7147 ハードウェアマニュアルの「第 14 章 シンクロナスシリアルコミュニケーションユニット」を参照してください。

### 3.2 動作説明

#### (1) EEPROM へのデータ書き込み

手順 書き込み許可状態へのセット / 書き込み許可状態の確認

- EEPROM へ WREN (書き込み許可) コードを送信し、EEPROM ステータスレジスタの WEL ビットを 1 にセットし、EEPROM を書き込み許可状態にします。
- EEPROM へ RDSR (ステータスレジスタ読み出し) コードを送信し、ステータスレジスタの WEL ビットが 1 にセットされていることを確認し、EEPROM が書き込み許可状態になっていることを確認します。
- 図 3 に通信内容と動作内容を示します。また、表 6 にソフトウェアおよびハードウェア処理内容を示します。

手順 データ書き込み

- EEPROM へ WRITE (書き込み) コード、書き込みアドレス (上位 8 ビット、下位 8 ビット)、書き込みデータを送信し、EEPROM へデータを書き込みます。
- 図 4 に通信内容と動作内容を示します。また、表 7 にソフトウェアおよびハードウェア処理内容を示します。

手順 データ書き込みの終了確認

- EEPROM へ RDSR (ステータスレジスタ読み出し) コードを送信し、ステータスレジスタの WIP ビットが 0 にクリアされたことを確認し、書き込み処理が終了したことを確認します。
- 図 5 に通信内容と動作内容を示します。また、表 8 にソフトウェアおよびハードウェア処理内容を示します。



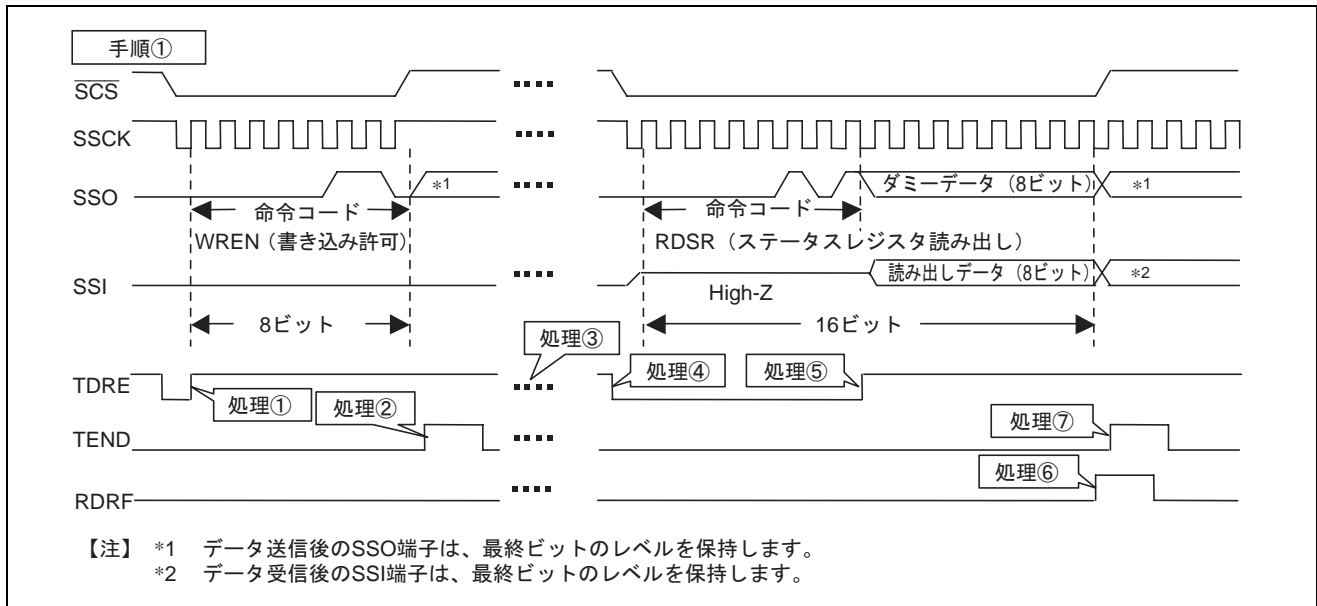


図3 通信内容と動作内容（書き込み許可状態へのセット / 書き込み許可状態の確認）

表6 ソフトウェアおよびハードウェア処理内容（書き込み許可状態へのセット / 書き込み許可状態の確認）

	ソフトウェア処理	ハードウェア処理
処理	<ul style="list-style-type: none"> <li>TE ビットを 1 にセットし、送信を許可</li> <li>TDRE ビットが 1 にセットされていることを確認</li> <li>SSTDR0 レジスタに送信データ（書き込み許可コード）をセット</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR0 レジスタへ送信データをセットすると、TDRE ビットを 0 クリア</li> <li>SSTDR0 レジスタから SSTRSR レジスタへデータを転送後、TDRE ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>TEND が 1 にセットされている（送信完了）ことを確認し、TEND を 0 クリア</li> <li>TE ビットを 0 クリアし、送信を禁止</li> </ul>	<ul style="list-style-type: none"> <li>TDRE = 1 の状態で、最後尾ビットの送信後、TEND ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>DATS ビットを 1 にセットし、送信データ長を 16 ビットに設定</li> <li>TEND ビットが 0 にセットされていることを確認</li> <li>1 ビット期間経過するまで待機</li> <li>TE ビットを 1、RE ビットを 1 にセットし、送受信を許可</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> </ul>
処理	<ul style="list-style-type: none"> <li>TDRE ビットが 1 にセットされていることを確認</li> <li>SSTDR0 レジスタに送信データ（ステータス読み出しコード）をセット</li> <li>SSTDR1 レジスタに送信データ（ダミーデータ）をセット</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR0~1 レジスタへ送信データをセットすると、TDRE ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>なし</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR1 レジスタから SSTRSR レジスタへデータを転送後、TDRE ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>RDRF = 1、ORER 1 を確認後、SSRDR0~1 レジスタの値を RAM へ転送</li> </ul>	<ul style="list-style-type: none"> <li>SSTRSR レジスタから SSRDR1 レジスタに受信データを転送後 RDRF ビットを 1 にセット</li> <li>SSRDR0~1 レジスタの値を CPU でリードすると、RDRF ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>TEND ビットが 1 にセットされている（送信完了）ことを確認し、TEND ビットを 0 クリア</li> <li>1 ビット期間経過するまで待機</li> <li>SSRDR1 レジスタの値の 1 ビット（EEPROMWEL ビット）が 1 にセットされていること（書き込み許可状態になっている）を確認</li> <li>TE ビットを 0、RE ビットを 0 にクリアし、送受信を禁止</li> </ul>	<ul style="list-style-type: none"> <li>TDRE = 1 の状態で、最後尾ビットの送信後、TEND ビットを 1 にセット</li> </ul>

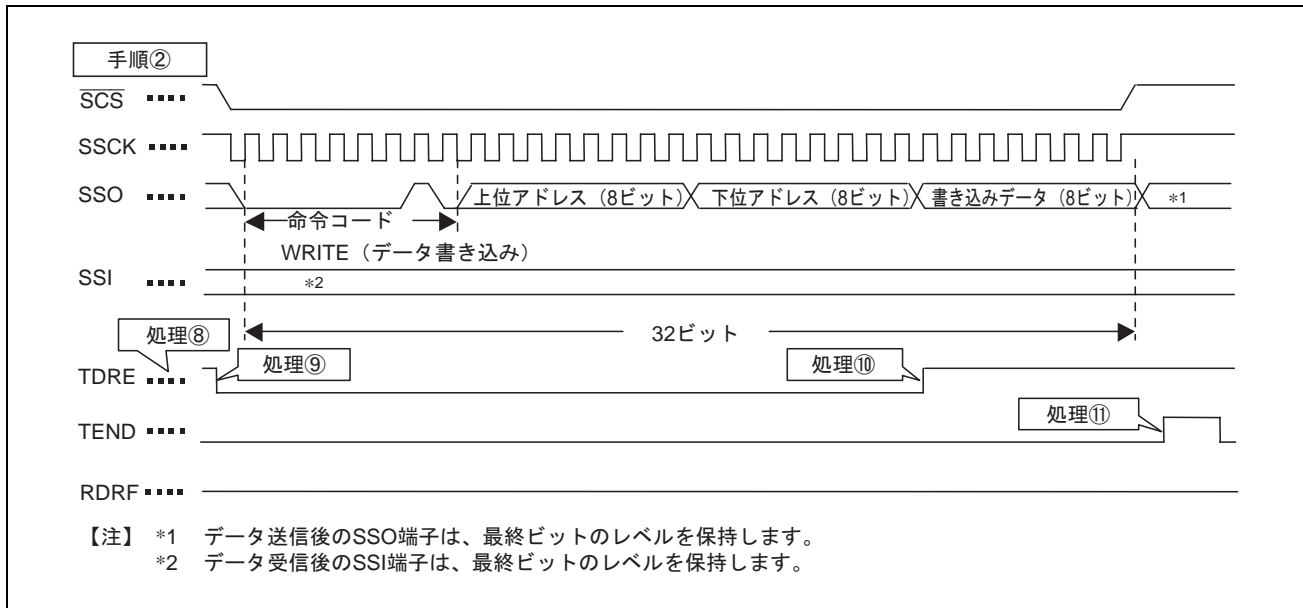


図4 通信内容と動作内容 (データ書き込み)

表7 ソフトウェアおよびハードウェア処理内容 (データ書き込み)

	ソフトウェア処理	ハードウェア処理
処理	<ul style="list-style-type: none"> <li>DATS ビットを 2 にセットし、送信データ長を 32 ビットに設定</li> <li>TE ビットを 1 にセットし、送信を許可</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> </ul>
処理	<ul style="list-style-type: none"> <li>TDRE ビットが 1 にセットされていることを確認</li> <li>SSTDR0 レジスタに送信データ (書き込みコード) をセット</li> <li>SSTDR1 レジスタに送信データ (上位アドレス) をセット</li> <li>SSTDR2 レジスタに送信データ (下位アドレス) をセット</li> <li>SSTDR3 レジスタに送信データ (書き込みデータ) をセット</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR0~3 レジスタへ送信データをセットすると、TDRE ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>なし</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR3 レジスタから SSTRSR レジスタへデータを転送後、TDRE ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>TEND ビットが 1 にセットされている (送信完了) ことを確認し、TEND ビットを 0 クリア</li> <li>TEND が 0 クリアされたことを確認</li> <li>1 ビット期間経過するまで待機</li> <li>TE ビットを 0 にクリアし、送信を禁止</li> </ul>	<ul style="list-style-type: none"> <li>TDRE = 1 の状態で、最後尾ビットの送信後、TEND ビットを 1 にセット</li> </ul>

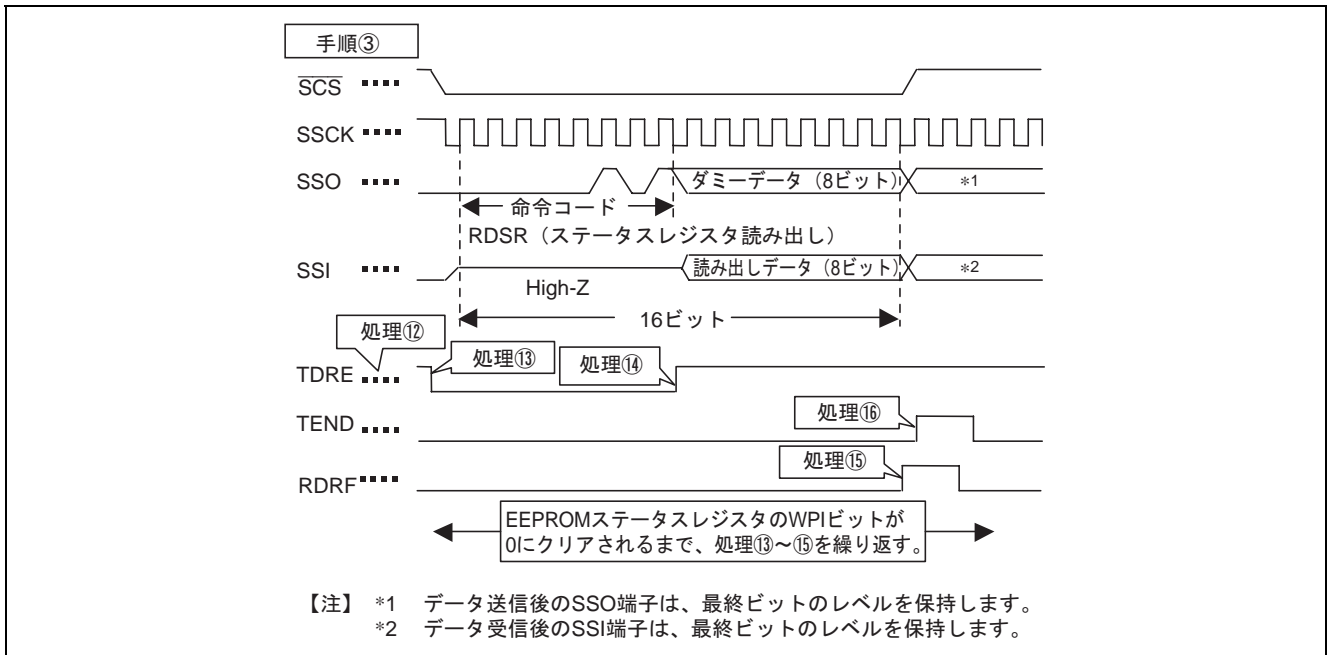


図5 通信内容と動作内容 (データ書き込みの終了確認)

表8 ソフトウェアおよびハードウェア処理内容 (データ書き込みの終了確認)

	ソフトウェア処理	ハードウェア処理
処理	<ul style="list-style-type: none"> <li>DATS ビットを 1 にセットし、送信データ長を 16 ビットに設定</li> <li>TE ビットを 1、RE ビットを 1 にセットし、送受信を許可</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> </ul>
処理	<ul style="list-style-type: none"> <li>TDRE ビットが 1 にセットされていることを確認</li> <li>SSTDR0 レジスタに送信データ (ステータス読み出しコード) をセット</li> <li>SSTDR1 レジスタに送信データ (ダミーデータ) をセット</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR0 ~ 1 レジスタへ送信データをセットすると、TDRE ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>なし</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR1 レジスタから SSTRSR レジスタへデータを転送後、TDRE ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>RDRF = 1、ORER 1 を確認後、SSRDR0 ~ 1 レジスタの値を RAM へ転送</li> </ul>	<ul style="list-style-type: none"> <li>SSTRSR レジスタから SSRDR1 レジスタに受信データを転送後 RDRF ビットを 1 にセット</li> <li>SSRDR0 ~ 1 レジスタの値を CPU でリードすると、RDRF ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>TEND ビットが 1 にセットされている (送信完了) ことを確認し、TEND ビットを 0 クリア</li> <li>1 ビット期間経過するまで待機</li> <li>SSRDR2 レジスタの値の 1 ビット (EEPROM/WEL ビット) を確認し、0 にクリア (書き込みが終了) されていれば、TE ビットを 0、RE ビットを 0 にクリアし、送受信を禁止。0 にクリア (書き込みが終了) されていなければ、処理 に戻る。</li> </ul>	<ul style="list-style-type: none"> <li>TDRE = 1 の状態で、最後尾ビットの送信後、TEND ビットを 1 にセット</li> </ul>

(2) EEPROM からのデータ読み出し

手順 データ書き込みの終了確認

- EEPROM へ RDSR (ステータスレジスタ読み出し) コードを送信し、ステータスレジスタの WIP ビットが 0 にクリアされたことを確認し、書き込み処理が終了したことを確認します。
- 図 5 に通信内容と動作内容を示します。また、表 9 にソフトウェアおよびハードウェア処理内容を示します。

手順 データ読み出し

- EEPROM へ WREN (書き込み許可) コードを送信し、EEPROM からデータを読み出します。そして、読み出したデータを RAM へ転送します。
- 図 6 に通信内容と動作内容を示します。また、表 10 にソフトウェアおよびハードウェア処理内容を示します。

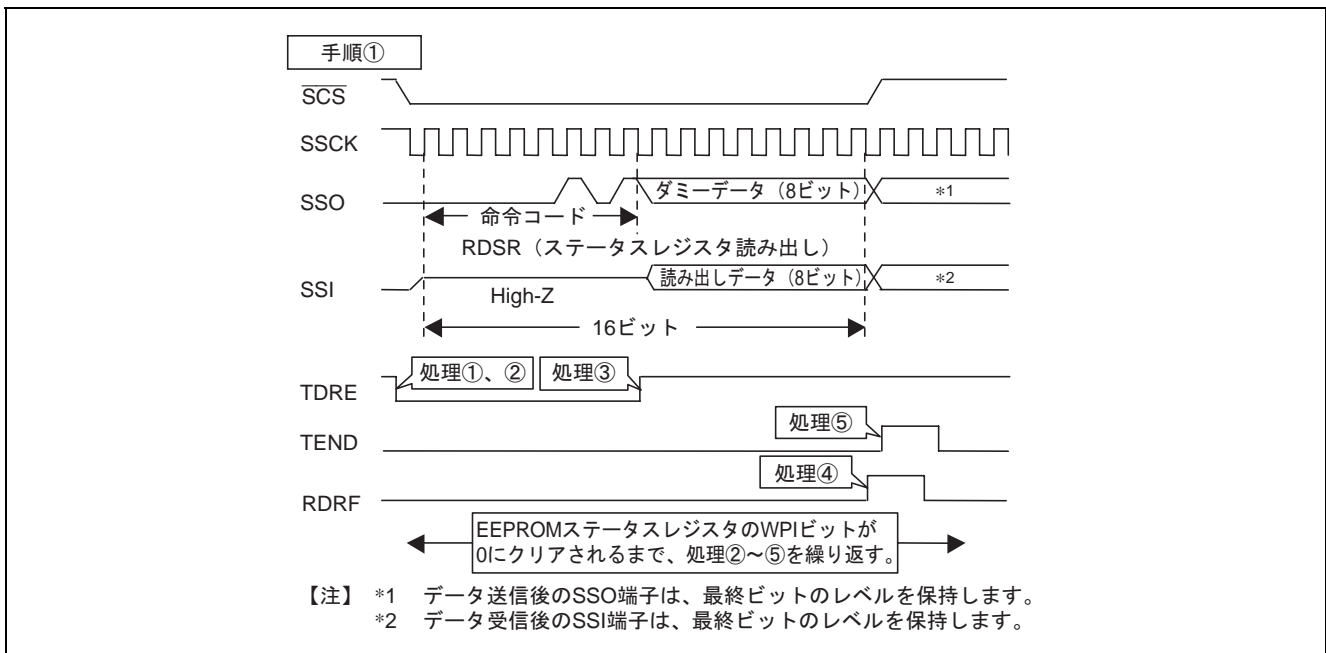


図6 通信内容と動作内容 (データ書き込みの終了確認)

表9 ソフトウェアおよびハードウェア処理内容 (データ書き込みの終了確認)

	ソフトウェア処理	ハードウェア処理
処理	<ul style="list-style-type: none"> <li>• DATS ビットを 1 にセットし、送信データ長を 16 ビットに設定</li> <li>• TE ビットを 1、RE ビットを 1 にセットし、送受信を許可</li> </ul>	<ul style="list-style-type: none"> <li>• なし</li> </ul>
処理	<ul style="list-style-type: none"> <li>• TDRE ビットが 1 にセットされていることを確認</li> <li>• SSTDR0 レジスタに送信データ (ステータス呼び出しコード) をセット</li> <li>• SSTDR1 レジスタに送信データ (ダミーデータ) をセット</li> </ul>	<ul style="list-style-type: none"> <li>• SSTDR0~1 レジスタへ送信データをセットすると、TDRE ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>• なし</li> </ul>	<ul style="list-style-type: none"> <li>• SSTDR1 レジスタから SSTRSR レジスタへデータを転送後、TDRE ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>• RDRF = 1、ORER 1 を確認後、SSRDR0~1 レジスタの値を RAM へ転送</li> <li>• RDRF = 0 を確認</li> </ul>	<ul style="list-style-type: none"> <li>• SSTRSR レジスタから SSRDR1 レジスタに受信データを転送後 RDRF ビットを 1 にセット</li> <li>• SSRDR0~1 レジスタの値を CPU でリードすると、RDRF ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>• TDRE ビットと TEND ビットが 1 にセットされている (送信完了) ことを確認し、TEND ビットを 0 クリア</li> <li>• 1 ビット期間経過するまで待機</li> <li>• SSRDR2 レジスタの値の 0 ビット (EEPROM/WIP ビット) を確認し、0 にクリア (書き込みが終了) されていれば、TE ビットを 0、RE ビットを 0 にクリアし、送受信を禁止。0 にクリア (書き込みが終了) されていない場合は、処理に戻る。</li> </ul>	<ul style="list-style-type: none"> <li>• TDRE = 1 の状態で、最後尾ビットの送信後、TEND ビットを 1 にセット</li> </ul>

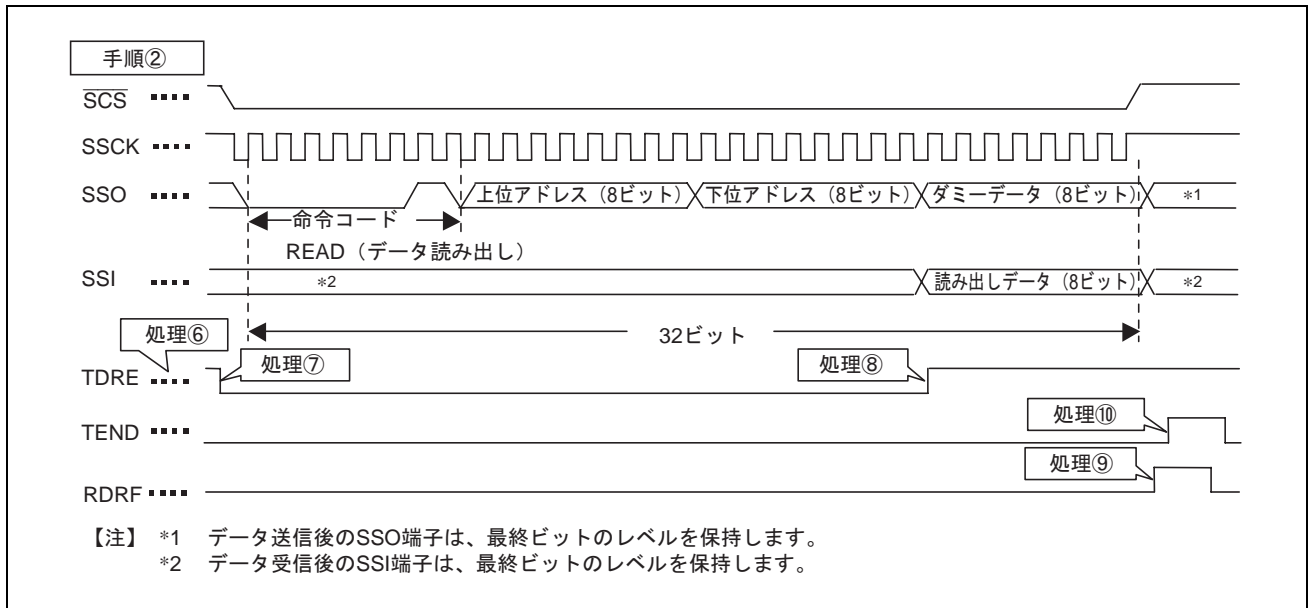


図7 通信内容と動作内容（データ読み出し）

表10 ソフトウェアおよびハードウェア処理内容（データ読み出し）

	ソフトウェア処理	ハードウェア処理
処理	<ul style="list-style-type: none"> <li>DATS ビットを 2 にセットし、送信データ長を 32 ビットに設定</li> <li>TE ビットを 1、RE ビットを 1 にセットし、送受信を許可</li> </ul>	<ul style="list-style-type: none"> <li>なし</li> </ul>
処理	<ul style="list-style-type: none"> <li>TDRE ビットが 1 にセットされていることを確認</li> <li>SSTDR0 レジスタに送信データ（読み出しコード）をセット</li> <li>SSTDR1 レジスタに送信データ（上位アドレス）をセット</li> <li>SSTDR2 レジスタに送信データ（下位アドレス）をセット</li> <li>SSTDR3 レジスタに送信データ（ダミーデータ）をセット</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR0～3 レジスタへ送信データをセットすると、TDRE ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>なし</li> </ul>	<ul style="list-style-type: none"> <li>SSTDR3 レジスタから SSTRSR レジスタへデータを転送後、TDRE ビットを 1 にセット</li> </ul>
処理	<ul style="list-style-type: none"> <li>RDRF = 1、ORER 1 を確認後、SSRDR0～3 レジスタの値を RAM へ転送</li> </ul>	<ul style="list-style-type: none"> <li>SSTRSR レジスタから SSRDR3 レジスタに受信データを転送後 RDRF ビットを 1 にセット</li> <li>SSRDR0～3 レジスタの値を CPU でリードすると、RDRF ビットを 0 クリア</li> </ul>
処理	<ul style="list-style-type: none"> <li>TEND が 1 にセットされている（送信完了）ことを確認し、TEND を 0 クリア</li> <li>1 ビット期間経過するまで待機</li> <li>TE ビットを 0、RE ビットを 0 にクリアし、送受信を禁止</li> </ul>	<ul style="list-style-type: none"> <li>TDRE = 1 の状態で、最後尾ビットの送信後、TEND ビットを 1 にセット</li> </ul>

### 3.3 参考プログラムの処理手順

(1) 本応用例 (EEPROM へのデータ書き込み時 / EEPROM からのデータ読み出し時) で使用するレジスタ設定を示します。

表 11 にクロック発振器 (CPG)、低消費電力モード、PFC のレジスタ設定を示します。

表 12 に Synchronous Serial Communication Unit のレジスタ設定を示します。

表11 使用レジスタ設定 (CPG、低消費電力モード、PFC)

モジュール	レジスタ名	ビット名	ビット	機能	設定値
CPG	周波数制御レジスタ (FRQCR)	リザーブ	15		0
		IFC[2:0]	14-12	内部クロック (I $\phi$ ) 周波数の分周率 0: $\times 1$ 倍、入力クロック 8MHz のとき I $\phi$ : 64MHz	0
		BFC[2:0]	11-9	バスクロック (B $\phi$ ) 周波数の分周率 1: $\times 1/2$ 倍、入力クロック 8MHz のとき B $\phi$ : 32MHz	1
		PFC[2:0]	8-6	周辺クロック (P $\phi$ ) 周波数の分周率 1: $\times 1/2$ 倍、入力クロック 8MHz のとき P $\phi$ : 32MHz	1
		MIFC[2:0]	5-3	MTU2S クロック (MI $\phi$ ) 周波数の分周率 0: $\times 1$ 倍、入力クロック 8MHz のとき MI $\phi$ : 64MHz	0
		MPFC[2:0]	2-0	MTU2S クロック (I $\phi$ ) 周波数の分周率 1: $\times 1/2$ 倍、入力クロック 8MHz のとき MP $\phi$ : 32MHz	1
低消費電力	スタンバイ コントロールレジスタ (STBCR3)	リザーブ	7-6		1
		MSTP13	5	1: SCI_2 へのクロック供給停止 (SCI_2 は動作しない)	1
		MSTP12	4	1: SCI_1 へのクロック供給停止 (SCI_1 は動作しない)	1
		MSTP11	3	1: SCI_0 へのクロック供給停止 (SCI_0 は動作しない)	1
		MSTP10	2	0: SSU は動作	0
		MSTP8	0	1: RCAN-ET へのクロック供給停止 (RCAN-ET は動作しない)	1
PFC	ポート D コントロール レジスタ L3 (PDCRL3)	リザーブ	15-11		0
		PD10MD[2:0]	10-8	101: 端子機能を SSO 入出力 (SSU)	101
		リザーブ	7		0
		PD9MD[2:0]	6-4	101: 端子機能を SSI 入出力 (SSU)	101
		リザーブ	3		0
	ポート D コントロール レジスタ L2 (PDCRL2)	PD8MD[2:0]	2-0	101: 端子機能を SCK 入出力 (SSU)	101
		リザーブ	15		0
		PD7MD[2:0]	14-12	101: 端子機能を_SCS 入出力 (SSU)	101
		リザーブ	11		0
		PD6MD[2:0]	10-8	0: 端子機能を PD6 入出力 (汎用ポート)	0
		リザーブ	7		0
		PD5MD[2:0]	6-4	0: 端子機能を PD5 入出力 (汎用ポート)	0
		リザーブ	3		0
	ポート D・IO レジスタ L (PDIORL)	PD4MD[2:0]	2-0	0: 端子機能を PD4 入出力 (汎用ポート)	0
		リザーブ	15-11		0
		PD10IOR	10	1: PD10 端子は出力	1
		PD9IOR	9	1: PD9 端子は入力	0
		PD8IOR	8	1: PD8 端子は出力	1
		PD7IOR	7	1: PD7 端子は出力	1
		PD6IOR	6	1: PD6 端子は入力	0
		PD5IOR	5	1: PD5 端子は入力	0
PD4IOR		4	1: PD4 端子は入力	0	
PD3IOR		3	1: PD3 端子は入力	0	
PD2IOR		2	1: PD2 端子は入力	0	
PD1IOR	1	1: PD1 端子は入力	0		
PD0IOR	0	1: PD0 端子は入力	0		

表12 使用レジスタ設定 ( Synchronous Serial Communication Unit )

モジュール	レジスタ名	ビット名	ビット	機能	設定値
Synchronous Serial Communication Unit	SS コントロール レジスタ H ( SSCRH )	MSS	7	1 : マスタモード	1
		BIDE	6	0 : 標準モード	0
		リザーブ	5		0
		SOL	4	0 : シリアルデータの出力を Low レベルに変更	0
		SOLP	3	1 : SOL ビットの値によって、出力レベルを変更	1
		リザーブ	2		1
		CSS[1:0]	1-0	11 : $\overline{SCS}$ 端子を自動出力にセット	11
	SS コントロール レジスタ L ( SSCRL )	FCLRM	7	1 : レジスタアクセス時に割り込みフラグをクリア	1
		SSUMS	6	0 : SSU モード	0
		SRES	5	1 にセットすると、SSU 内部シーケンサをリセット	0
		リザーブ	4-2		0
		DATS[1:0]	1-0	00 : 8 ビットデータ長 01 : 16 ビットデータ長 10 : 32 ビットデータ長 11 : 設定禁止	00 01 10
	SS モードレジスタ ( SSMR )	MLS	7	1 : MSB ファーストで転送	1
		CPOS	6	0 : クロックの極性をアイドル時に H 出力、 アクティブ時に L 出力	0
		CPHS	5	0 : クロックの最初のエッジでデータ変化	0
		リザーブ	4-3		0
		CKS[2:0]	2-0	011 : 転送クロックレート P $\phi$ /16	011
	SS イネーブルレジスタ ( SSER )	TE	7	0 : 送信動作禁止 1 : 送信動作許可	0 1
		RE	6	0 : 受信動作禁止 1 : 受信動作許可	0 1
		リザーブ	5-4		0
		TEIE	3	0 : SSTEI ( 送信完了 ) 割り込み禁止	0
		TIE	2	0 : SSTXI ( 送信データエンプティ ) 割り込み禁止	0
		RIE	1	0 : SSRXI ( 受信データフル ) 割り込み、および SSOEI ( オーバーランエラー ) 割り込み禁止	0
		CEIE	0	0 : SSCEI ( コンフリクトエラー ) 割り込み禁止	0



モジュール	レジスタ名	ビット名	ビット	機能	設定値	
Synchronous Serial Communication Unit	SS ステータスレジスタ (SSSR)	リザーブ	7		0	
		ORER	6	1: オーバーランエラーが発生 [セット条件] RDRF=1 で、次の 1 バイト受信が完了 [クリア条件] ORER=1 をリードした後、0 をライト	0	
		リザーブ	5-4		0	
		TEND	3	1: 送信完了 [セット条件] • TENDSTS = 0 のとき、TDRE=1 で、送信データの最後尾ビットの送信時 • TENDSTS = 1 のとき、TDRE=1 で、送信データの最後尾ビットの送信後 [クリア条件] • TEND=1 をリードした後、0 をライト • SSTDR ヘデータをライト	0 1	
		TDRE	2	1: SSTDR 内のデータ無し [セット条件] • TE=0 • SSTDR から SSTRSR にデータが転送され、SSTDR にデータライトが可能になったとき [クリア条件] • TDRE=1 をリードした後、TDRE フラグに 0 をライト • TE=1 で、SSTDR ヘライト • SSTXI 割り込みにより DTC が起動され、DTC の DISEL ビットが 0 のとき、SSTDR に転送データをライト	0 1	
		RDRF	1	1: SSRDR 内のデータ有り [セット条件] シリアル受信が正常終了し、SSTRSR から SSRDR へ受信データが転送されたとき [クリア条件] • RDRF=1 をリードした後、RDRF フラグに 0 をライト • SSRDR から受信データをリード • SSRXI 割り込みにより DTC が起動され、DTC の DISEL ビットが 0 のとき、SSRDR に受信データをリード	0 1	
		CE	0	1: コンフリクトエラー/インコンプリートエラーが発生	0	
	SS コントロールレジスタ 2 (SSCR2)	リザーブ	7-5		0	
		TENDSTS	4	1: 最後尾ビット送信後に TEND ビットをセット	1	
		SCSATS	3	SCS 端子のアサートタイミングを選択 0: $t_{LEAD}$ , $t_{LAG}$ の出力期間の Min. を $1/2 \times t_{SUcyc}$ 【注】 $t_{LEAD}$ , $t_{LAG}$ , $t_{SUcyc}$ についてはハードウェアマニュアルの AC 特性を参照してください。	0	
		SSODTS	2	SSO 端子のデータ出力タイミングを選択 0: BIDE0, MSS=1, TE=1, または BIDE=1, TE=1, RE=0 のとき SSO 端子はデータを出力	0	
		リザーブ	1-0		0	
	SS トランスミットデータレジスタ 0~3 (SSTDR0~3)			7-0	送信データを格納するための 8 ビットレジスタ	H'00 (初期値)
	SS レシーブデータレジスタ 0~3 (SSRDR0~3)			7-0	受信データを格納するための 8 ビットレジスタ	H'00 (初期値)

(2) 書き込みフロー

図 8～図 13 に EEPROM への書き込み処理フローを示します。

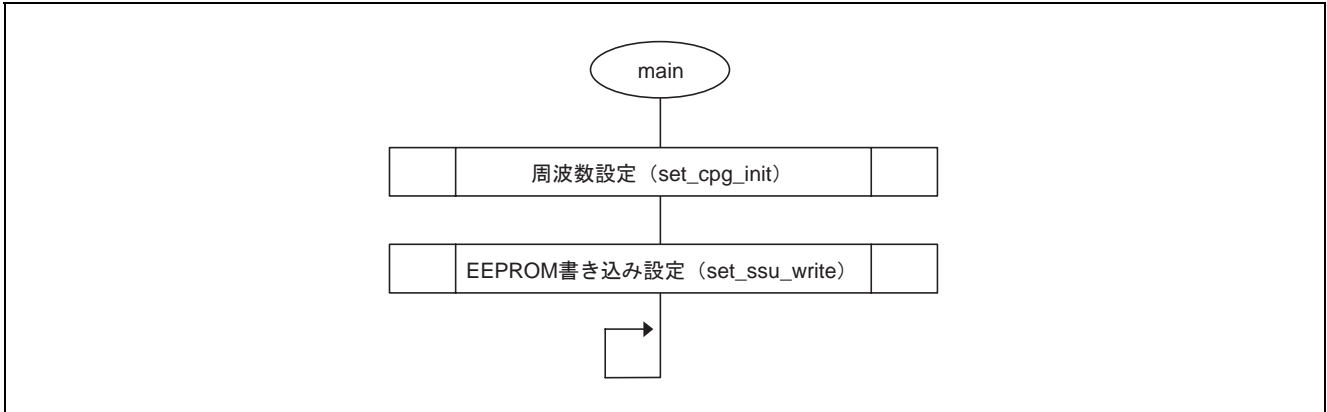


図8 メインフロー (EEPROM へのデータ書き込み時)

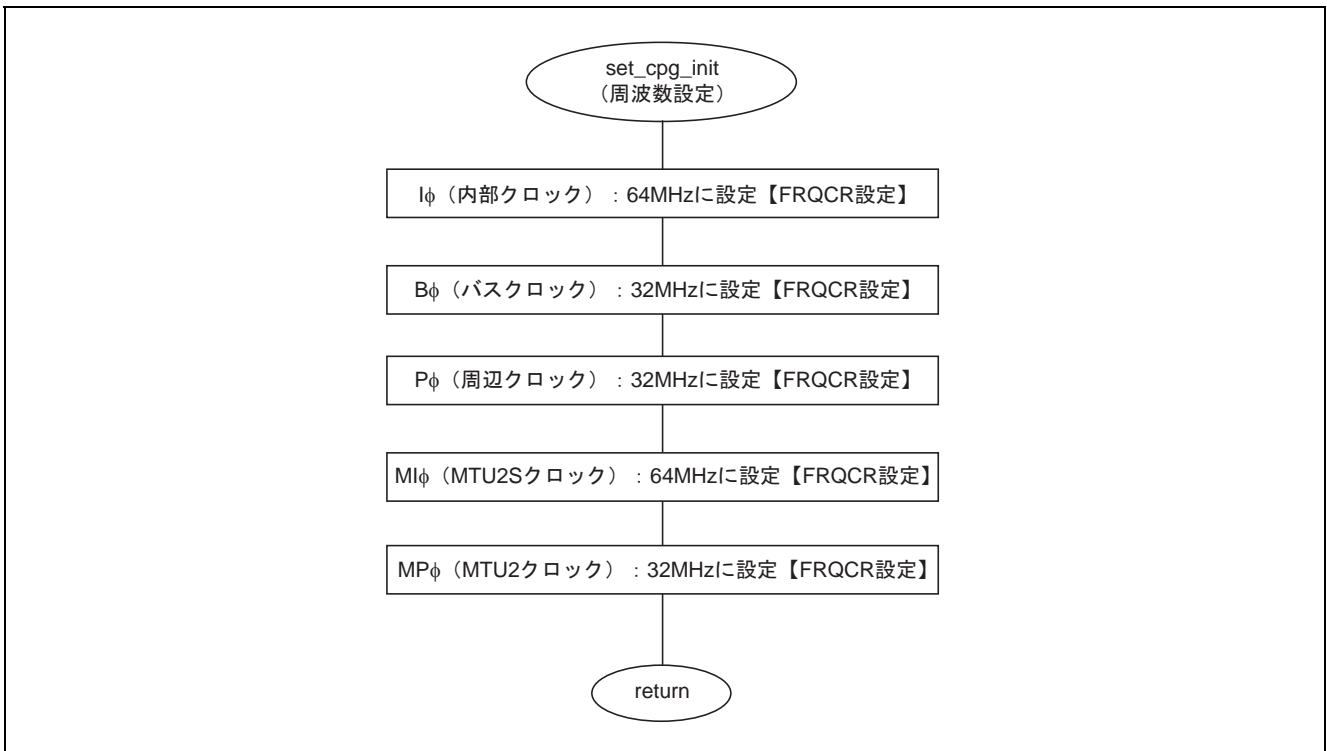


図9 各周波数設定フロー

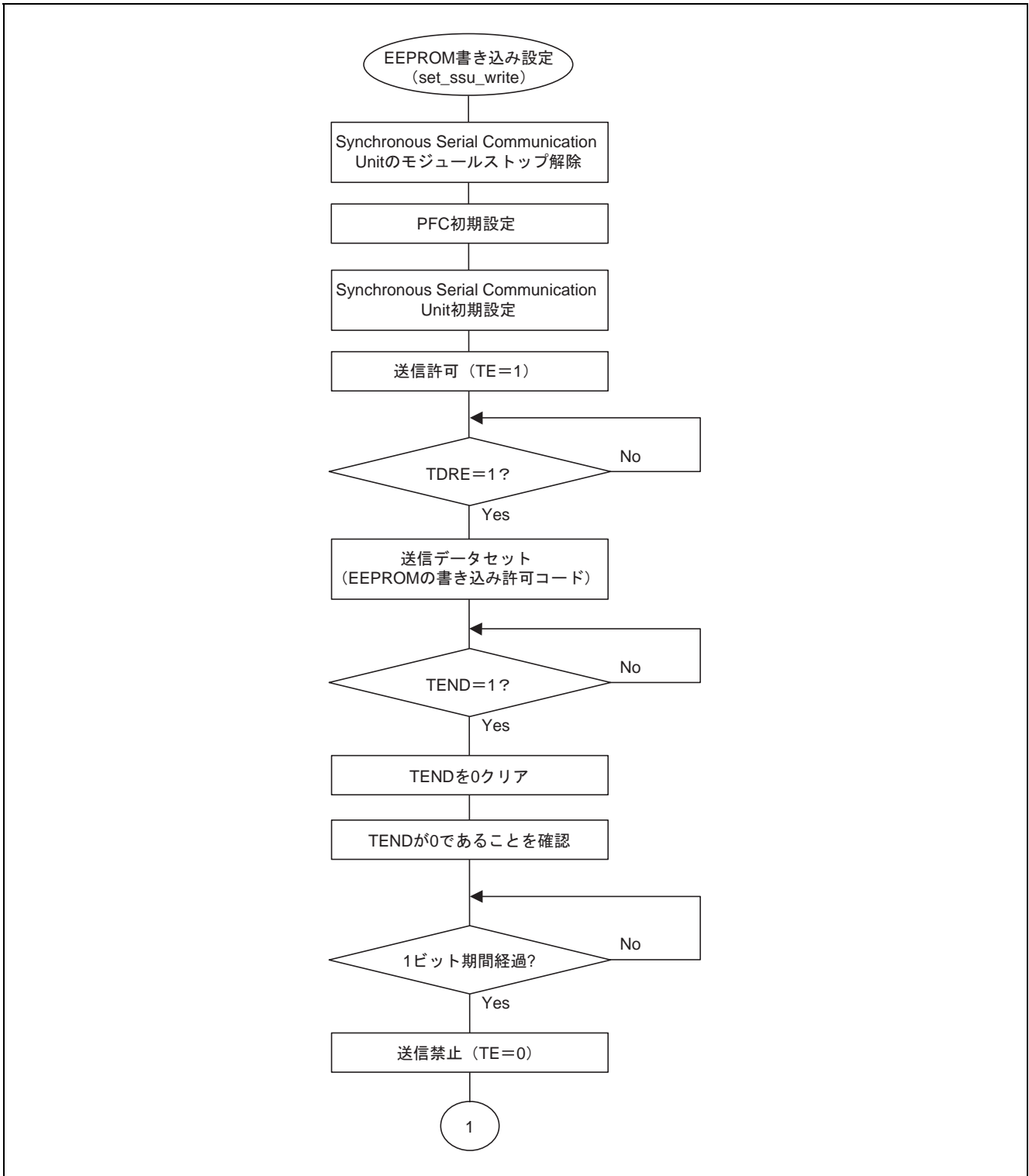


図10 データ書き込みフロー

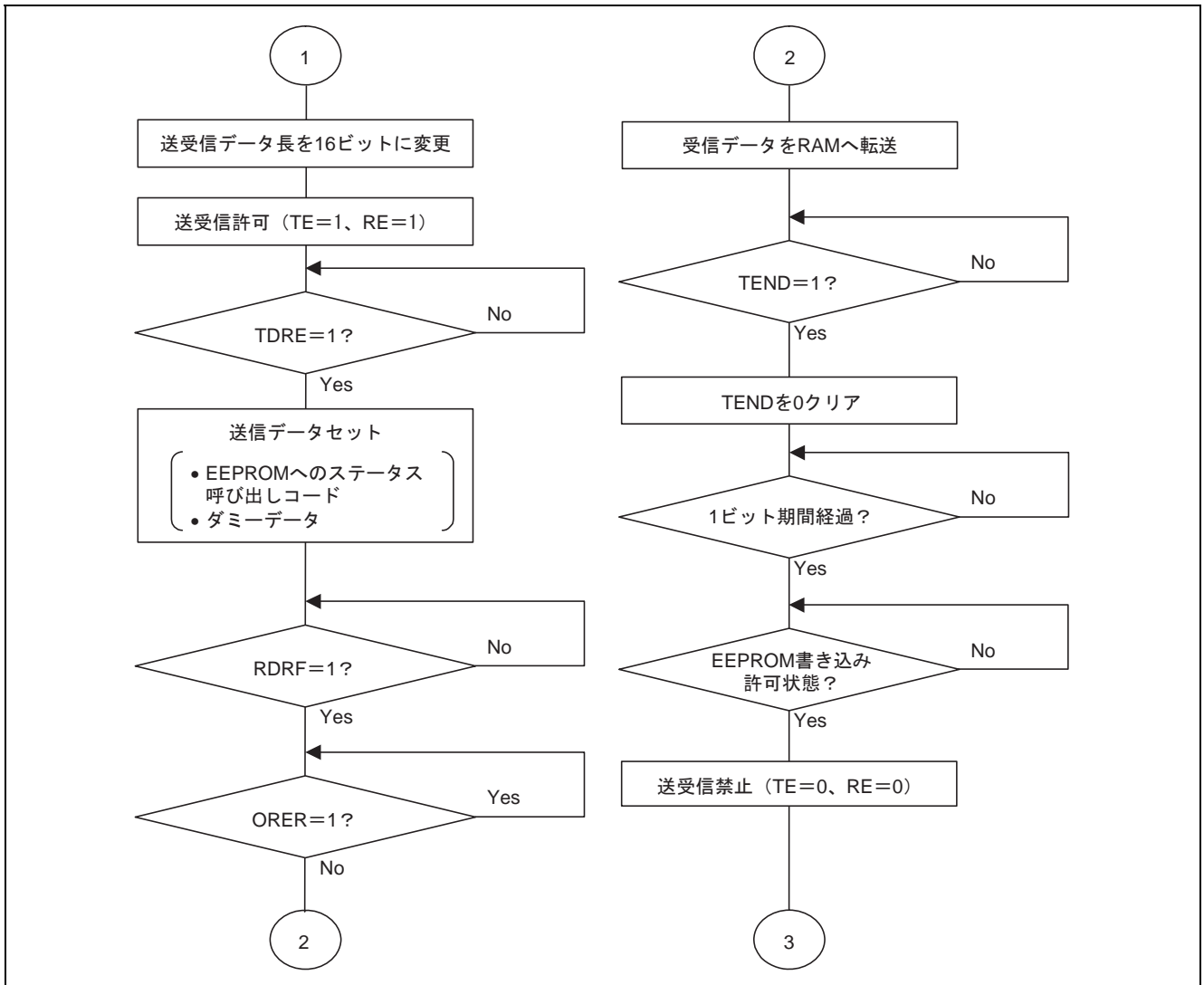


図11 データ書き込みフロー

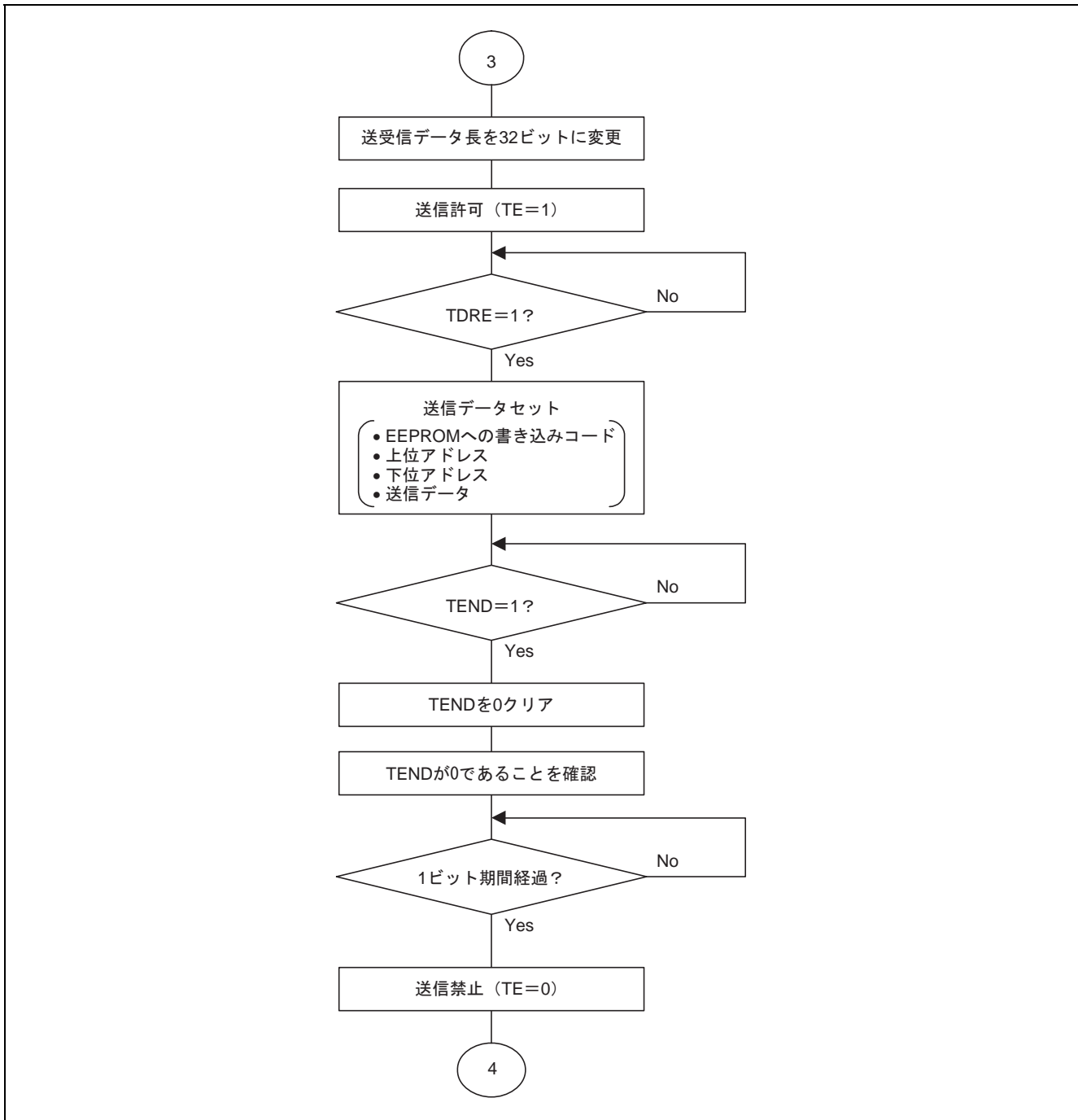


図12 データ書き込みフロー

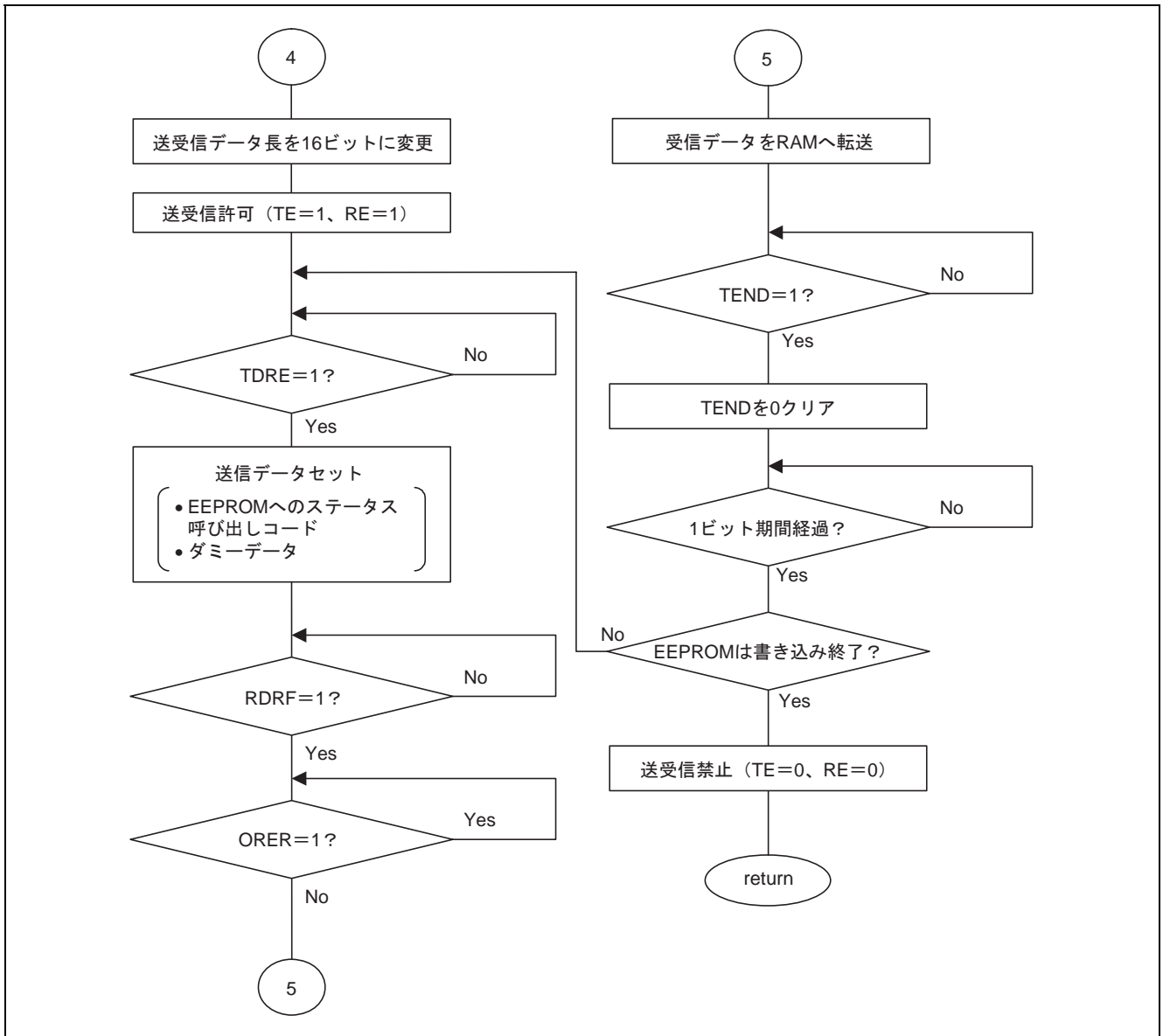


図13 データ書き込みフロー

(3) 読み出しフロー

図 14～図 17 に EEPROM からの読み出し処理フローを示します。

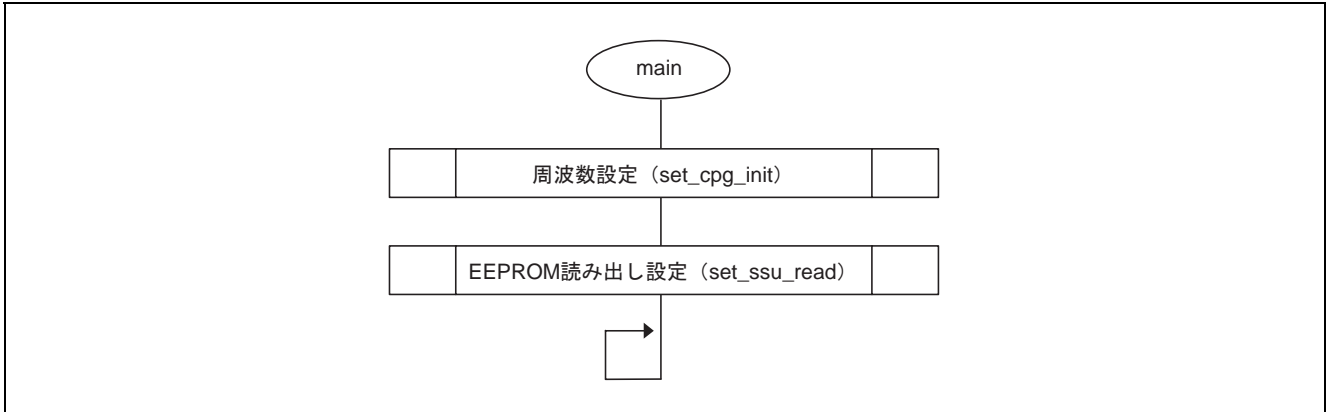


図14 メインフロー (EEPROM へのデータ読み出し時)

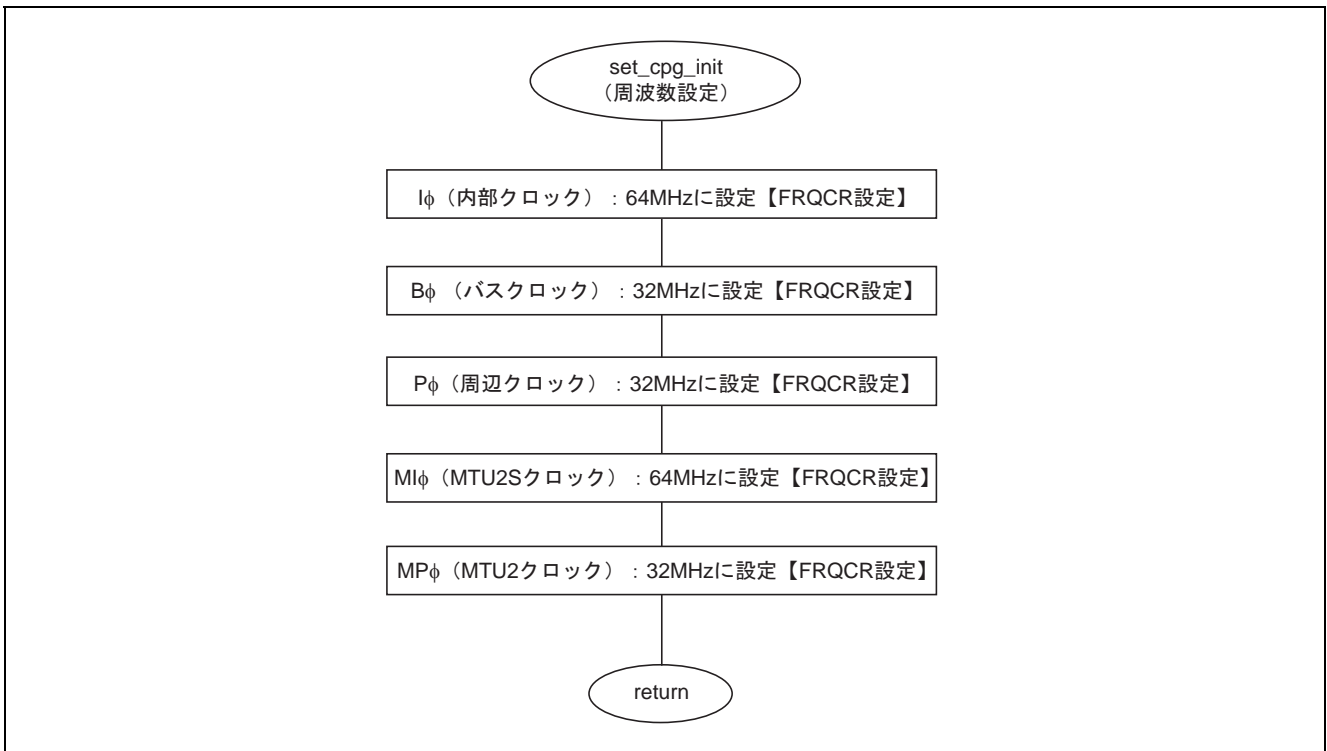


図15 各周波数設定フロー

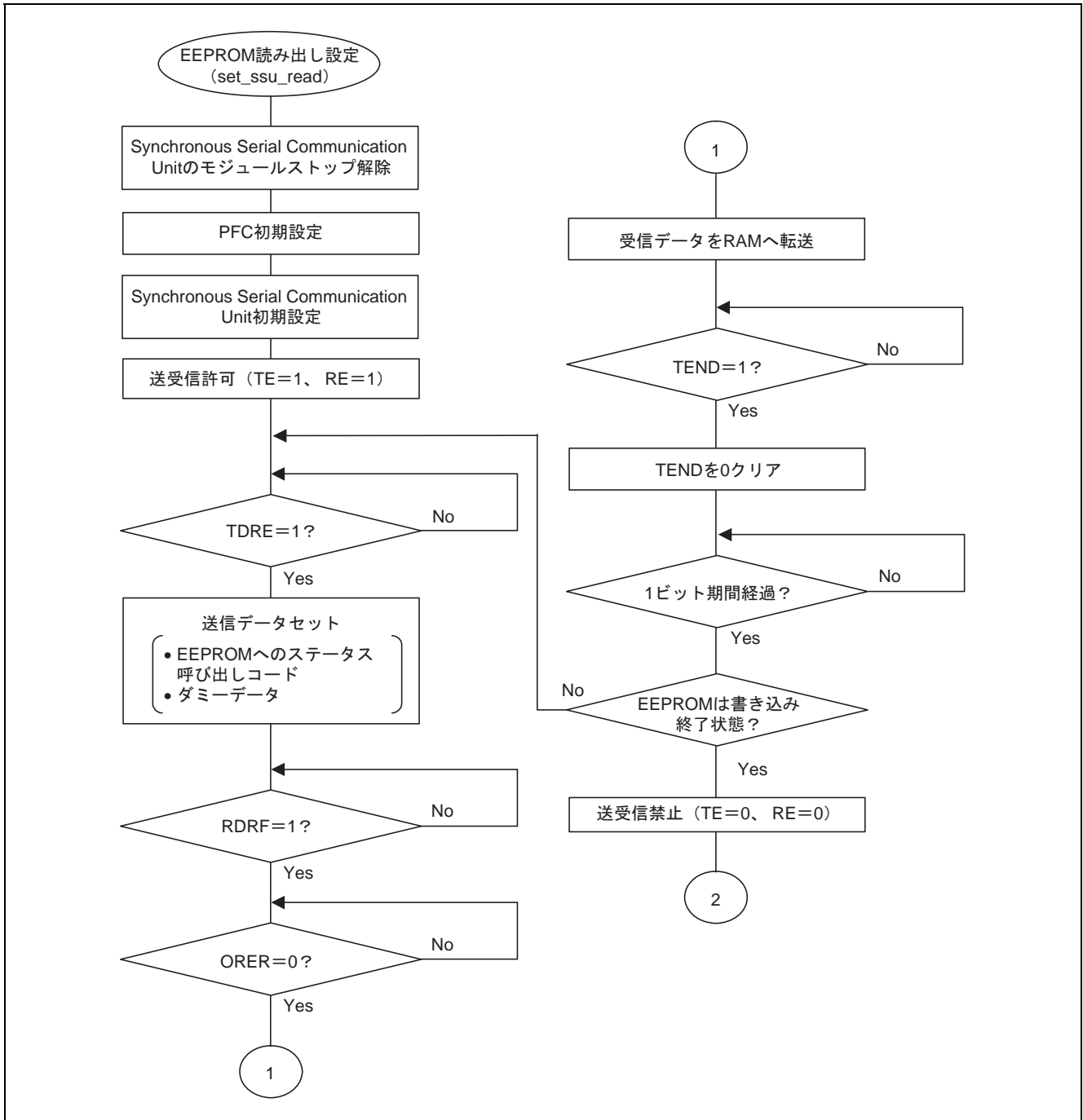


図16 データ読み出しフロー



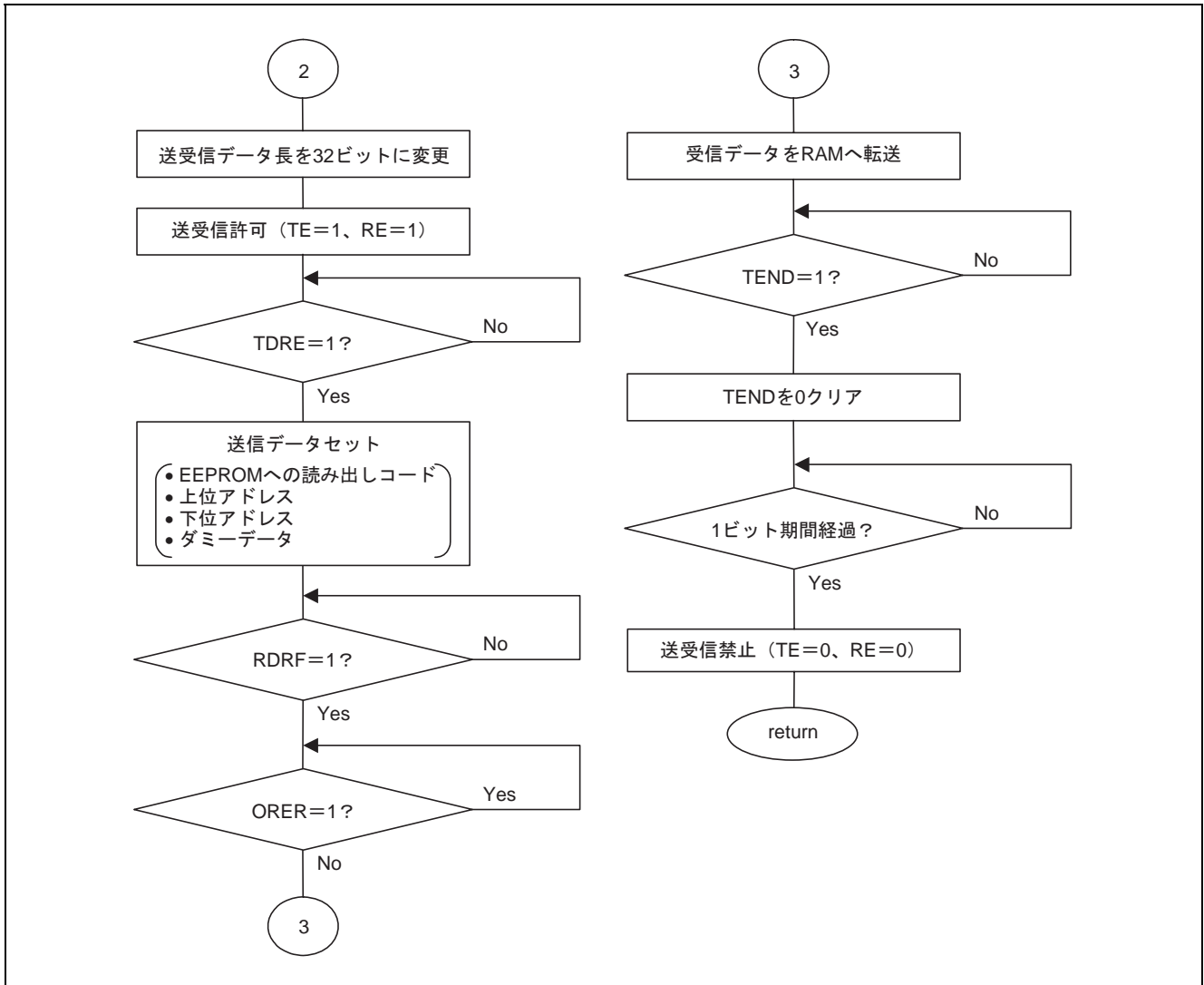


図17 データ読み出しフロー

## 4. 参考プログラム例

### (1) Synchronous Serial Communication Unit 通信による EEPROM へのデータ書き込み

```

/*****/
/*  SH7147                                     */
/*  Synchronous Serial Communication Unit 通信による EEPROM 書き込み  */
/*****/

/*----- Include File-----*/
#include <machine.h>
#include "iodefine.h"

/*****/
/*      関数宣言                                     */
/*****/
void main(void);          /* メインルーチン */
void set_cpg_init(void); /* 各クロック設定 */
void set_ssu_write(void); /* EEPROM 書き込み設定 */
void wait_1bit(void);    /* 1 bit 期間経過 */

/*****/
/*      メインルーチン                                     */
/*****/
void main(void){

    set_cpg_init();          /* 各クロックの設定 */
    set_ssu_write();        /* EEPROM 書き込み設定 */

    while(1);
}

/*****/
/*      変数定義                                     */
/*****/

unsigned char write_data; /* 書き込みデータ */
unsigned char upper_address; /* 書き込みデータ上位アドレス */
unsigned char under_address; /* 書き込みアドレス下位アドレス */
unsigned char dummy_data; /* 受信データ(ダミーデータ)格納 RAM */
    
```

```

unsigned char status_flag1; /* 受信データ (ステータスレジスタ) 格納 RAM */
unsigned char status_flag2; /* 受信データ (ステータスレジスタ) 格納 RAM */

/*****
/*      各クロックの設定                                */
/*      Iφ:Bφ:Pφ:MIφ:MPφ = 64MHz:32MHz:32MHz:64MHz:32MHz    */
*****/

void set_cpg_init(void){

    CPG.FRQCR.BIT.IFC = 0;      /* Iφ=64MHz */
    CPG.FRQCR.BIT.BFC = 1;      /* Bφ=32MHz */
    CPG.FRQCR.BIT._PFC = 1;     /* Pφ=32MHz */
    CPG.FRQCR.BIT.MIFC = 0;     /* MIφ=64MHz */
    CPG.FRQCR.BIT.MPFC = 1;     /* MPφ=32MHz */
}

/*****
/*      1bit ( 500ns ) 期間経過設定                        */
*****/

void wait_1bit(void){

    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
}

```

```

    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    }

/*****/
/*      Synchronous Serial Communication Unit 書き込み設定ルーチン      */
/*****/
void set_ssu_write(void){

    /* RAM 初期化 */
    dummy_data = 0x00;
    status_flag1 = 0x00;
    status_flag2 = 0xFF;

    /* 書き込みデータの設定 */
    write_data = 0x55;

    /* 書き込みアドレス設定 */
    upper_address = 0x1A;
    under_address = 0xAA;

    /* モジュールスタンバイの解除 */
    STB.CR3.BIT._SSU = 0;                /* 0:active 1:standby */

    /* PFC 設定 */
    PFC.PDCRL3.BIT.PD10MD = 5;          /* SSO に設定 */

```

```

PFC.PDCRL3.BIT.PD9MD = 5;          /* SSI に設定 */
PFC.PDCRL3.BIT.PD8MD = 5;          /* SSCK に設定 */
PFC.PDCRL2.BIT.PD7MD = 5;          /* SCS に設定 */
PFC.PDIORL.BIT.B10 = 1;           /* PD10(SSO)を出力 */
PFC.PDIORL.BIT.B9 = 0;            /* PD9(SSI)を入力 */
PFC.PDIORL.BIT.B8 = 1;            /* PD8(SSCK)を出力 */
PFC.PDIORL.BIT.B7 = 1;            /* PD7(SCS)を出力 */

/* Synchronous Serial Communication Unit モジュールの初期設定 */
SSU.SSER.BYTE = 0x00;              /* 送受信を禁止 */
SSU.SSCRH.BYTE = 0x8F;             /* マスタモード、標準モード、Low レベル出力、自動出力 */
SSU.SSCRL.BYTE = 0x80;             /* レジスタアクセス時にフラグクリア、SSU モード、
                                     データ長：8 ビット */
SSU.SSMR.BYTE = 0x83;              /* MSB ファースト、転送クロック:  $\Phi(32\text{MHz})/16 = 2\text{MHz}$  */
SSU.SSCR2.BYTE = 0x10;             /* 最後尾ビットの送信後に TEND ビットをセット*/

/* EEPROM 書き込み許可コード送信 */
SSU.SSER.BIT.TE = 1;               /* 送信許可 */
while(SSU.SSSR.BIT.TDRE !=1);      /* TDRE=1 の確認 */
SSU.SSTDR0 = 0x06;                 /* 送信データ (書き込み許可コード) セット */
while(SSU.SSSR.BIT.TEND !=1);      /* TEND=1 の確認 */
SSU.SSSR.BIT.TEND = 0;              /* TEND クリア */
while(SSU.SSSR.BIT.TEND !=0);      /* TEND=0 の確認 */
wait_lbit();                        /* 1bit 期間経過 */
SSU.SSER.BIT.TE = 0;               /* 送信禁止 */

/* EEPROM ステータスフラグチェック (書き込み許可状態であることを確認) */
SSU.SSCRL.BIT.DATS = 1;            /* データ長:16 ビット */
SSU.SSER.BYTE = 0xC0;              /* 送受信を許可 TE=1.RE=1 */
while(SSU.SSSR.BIT.TDRE !=1);      /* TDRE=1 の確認 */
SSU.SSTDR0 = 0x05;                 /* 送信データ (ステータス読み出しコード) セット */
SSU.SSTDR1 = 0x00;                 /* 送信データ (ダミーデータ) セット */
while(SSU.SSSR.BIT.RDRF !=1);      /* RDRF=1 の確認 */
while(SSU.SSSR.BIT.ORER ==1);      /* ORER≠1 の確認 */
dummy_data = SSU.SSRDR0;           /* ダミーデータ */
status_flag1 = SSU.SSRDR1;         /* 受信データ (ステータスレジスタ) */
while(SSU.SSSR.BIT.TEND !=1);      /* TEND=1 の確認 */
SSU.SSSR.BIT.TEND = 0;              /* TEND クリア */
    
```

```

while(SSU.SSSR.BIT.TEND !=0);          /* TEND=0 の確認 */
wait_lbit();                            /* 1bit 期間経過 */
while((status_flag1 & 0x02) != 0x02); /* EEPROM 書き込み許可状態? */
SSU.SSER.BYTE = 0x00;                  /* 送受信を禁止 */

/* EEPROM 書き込みデータ送信 */
SSU.SSCTL.BIT.DATS = 2;                /* データ長:32 ビット */
SSU.SSER.BIT.TE = 1;                  /* 送信許可 */
while((SSU.SSSR.BIT.TDRE & 1) !=1); /* TDRE=1 の確認 */
SSU.SSTDR0 = 0x02;                    /* 送信データ (書き込みコード) セット */
SSU.SSTDR1 = upper_address;           /* 送信データ (上位アドレス) セット */
SSU.SSTDR2 = under_address;          /* 送信データ (下位アドレス) セット */
SSU.SSTDR3 = write_data;             /* 送信データ (書き込みデータ) セット */
while(SSU.SSSR.BIT.TEND !=1);        /* TEND=1 の確認 */
SSU.SSSR.BIT.TEND = 0;                /* TEND クリア */
while(SSU.SSSR.BIT.TEND !=0);        /* TEND=0 の確認 */
wait_lbit();                          /* 1bit 期間経過 */
SSU.SSER.BIT.TE = 0;                  /* 送信禁止 */

/* EEPROM ステータスフラグチェック (書き込み動作が終了したことを確認) */
SSU.SSCTL.BIT.DATS = 1;                /* データ長:16 ビット */
SSU.SSER.BYTE = 0xC0;                 /* 送受信を許可 TE=1,RE=1 */
do {
    while(SSU.SSSR.BIT.TDRE !=1);      /* TDRE=1 の確認 */
    SSU.SSTDR0 = 0x05;                 /* 送信データ (ステータス読み出しコード) セット */
    SSU.SSTDR1 = 0x00;                 /* 送信データ (ダミーデータ) セット */
    while(SSU.SSSR.BIT.RDRF !=1);      /* RDRF=1 の確認 */
    while(SSU.SSSR.BIT.ORER ==1);      /* ORER≠1 の確認 */
    dummy_data = SSU.SSRDR0;           /* ダミーデータ */
    status_flag2 = SSU.SSRDR1;         /* 受信データ (ステータスレジスタ) */
    while(SSU.SSSR.BIT.TEND !=1);      /* TEND=1 の確認 */
    SSU.SSSR.BIT.TEND = 0;             /* TEND クリア */
    while(SSU.SSSR.BIT.TEND !=0);      /* TEND=0 の確認 */
    wait_lbit();                       /* 1bit 期間経過 */
} while ((status_flag2 & 0x01) != 0x00);
SSU.SSER.BYTE = 0x00;                 /* 送受信を禁止 */
}
    
```

(2) Synchronous Serial Communication Unit 通信による EEPROM からのデータ読み出し

```

/*****/
/* SH7147 */
/* Synchronous Serial Communication Unit 通信による EEPROM 読み出し */
/*****/

/*----- Include File-----*/
#include <machine.h>
#include "iodefine.h"

/*****/
/* 関数宣言 */
/*****/

void main(void); /* メインルーチン */
void set_cpg_init(void); /* 各クロック設定 */
void set_ssu_read(void); /* EEPROM 読み出し設定 */
void wait_1bit(void); /* 1bit 期間経過 */

/*****/
/* 変数定義 */
/*****/

unsigned char upper_address; /* 読み出し時の上位アドレス */
unsigned char under_address; /* 読み出し時の下位アドレス */
unsigned char dummy_data; /* 受信データ (ダミーデータ) 格納用 RAM */
unsigned char status_flag3; /* 受信データ (ステータスレジスタ) 格納用 RAM */
unsigned char ssu_rx_data0; /* 受信データ (ダミーデータ) 格納用 RAM */
unsigned char ssu_rx_data1; /* 受信データ (ダミーデータ) 格納用 RAM */
unsigned char ssu_rx_data2; /* 受信データ (ダミーデータ) 格納用 RAM */
unsigned char ssu_rx_data3; /* 受信データ (読み出しデータ) 格納用 RAM */

/*****/
/* メインルーチン */
/*****/

void main(void){

    set_cpg_init(); /* 各クロック設定 */
    set_ssu_read(); /* EEPROM 読み出し設定 */

```

```

while(1);
}

/*****
/*      各クロックの設定
/*      Iφ:Bφ:Pφ:MIφ:MPφ = 64MHz:32MHz:32MHz:64MHz:32MHz
*****/

void set_cpg_init(void){

    CPG.FRQCR.BIT.IFC = 0;      /* Iφ=64MHz */
    CPG.FRQCR.BIT.BFC = 1;      /* Bφ=32MHz */
    CPG.FRQCR.BIT._PFC = 1;     /* Pφ=32MHz */
    CPG.FRQCR.BIT.MIFC = 0;     /* MIφ=64MHz */
    CPG.FRQCR.BIT.MPFC = 1;     /* MPφ=32MHz */
}

/*****
/*      1bit (500ns) 期間経過設定
*****/

void wait_1bit(void){

    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
}

```



```

    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    nop();
    }

/*****/
/*      Synchronous Serial Communication Unit 読み出し設定ルーチン      */
/*****/
void set_ssu_read(void){

    /* RAM初期化 */
    dummy_data = 0x00;      /* 受信データ(デミーデータ)格納用RAM */
    status_flag3 = 0xFF;   /* 受信データ(ステータスレジスタ)格納用RAM */
    ssu_rx_data0 = 0;      /* 受信データ(デミーデータ)格納用RAM */
    ssu_rx_data1 = 0;      /* 受信データ(デミーデータ)格納用RAM */
    ssu_rx_data2 = 0;      /* 受信データ(デミーデータ)格納用RAM */
    ssu_rx_data3 = 0;      /* 受信データ(読み出しデータ)格納用RAM */

    /* 書き込みアドレス設定 */
    upper_address = 0x1A;
    under_address = 0xAA;

    /* モジュールスタンバイの解除 */
    STB.CR3.BIT._SSU = 0;      /* 0:active 1:standby */

    /* PFC設定 */
    PFC.PDCRL3.BIT.PD10MD = 5;      /* SSOに設定 */

```

```

PFC.PDCRL3.BIT.PD9MD = 5;          /* SSI に設定 */
PFC.PDCRL3.BIT.PD8MD = 5;          /* SSCK に設定 */
PFC.PDCRL2.BIT.PD7MD = 5;          /* SCS に設定 */
PFC.PDIORL.BIT.B10 = 1;           /* PD10(SSO)を出力 */
PFC.PDIORL.BIT.B9 = 0;            /* PD9(SSI)を入力 */
PFC.PDIORL.BIT.B8 = 1;            /* PD8(SSCK)を出力 */
PFC.PDIORL.BIT.B7 = 1;            /* PD7(SCS)を出力 */

/* Synchronous Serial Communication Unit モジュールの初期設定 */
SSU.SSER.BYTE = 0x00;              /* 送受信を禁止 */
SSU.SSCRH.BYTE = 0x8F;             /* マスタモード、標準モード、Low レベル出力、自動出力 */
SSU.SSCRL.BYTE = 0x81;             /* レジスタアクセス時にフラグクリア、SSU モード、
                                     データ長：16 ビット */
SSU.SSMR.BYTE = 0x83;              /* MSB ファースト、転送クロック:Pφ(32MHz)/16 = 2MHz */
SSU.SSCR2.BYTE = 0x10;             /* 最後尾ビットの送信後に TEND ビットをセット */

/* EEPROM ステータスフラグチェック */
SSU.SSER.BYTE = 0xC0;              /* 送受信を許可 TE=1.RE=1 */
do {
    while(SSU.SSSR.BIT.TDRE !=1);  /* TDRE=1 の確認 */
    SSU.SSTDR0 = 0x05;              /* 送信データ (ステータス読み出しコード) セット */
    SSU.SSTDR1 = 0x00;              /* 送信データ (ダミーデータ) セット */
    while(SSU.SSSR.BIT.RDRF !=1);  /* RDRF=1 の確認 */
    while(SSU.SSSR.BIT.ORER ==1);  /* ORE≠1 の確認 */
    dummy_data = SSU.SSRDR0;        /* ダミーデータ */
    status_flag3 = SSU.SSRDR1;      /* 受信データ (ステータスレジスタ) */
    while(SSU.SSSR.BIT.TEND !=1);  /* TEND=1 の確認 */
    SSU.SSSR.BIT.TEND = 0;          /* TEND クリア */
    wait_lbit();                    /* 1bit 期間経過 */
} while (status_flag3 != 0x00);    /* EEPROM 書き込み動作は終了? */
SSU.SSER.BYTE = 0x00;              /* 送受信を禁止 */

/* EEPROM データ読み出し */
SSU.SSCRL.BIT.DATS = 2;            /* データ長:32 ビット */
SSU.SSER.BYTE = 0xC0;              /* 送受信を許可 TE=1.RE=1 */
while(SSU.SSSR.BIT.TDRE !=1);    /* TDRE=1 の確認 */
SSU.SSTDR0 = 0x03;                /* 送信データ (読み出しコード) セット */
SSU.SSTDR1 = upper_address;        /* 送信データ (上位アドレス) セット */
    
```

```

SSU.SSTDR2 = under_address;          /* 送信データ(下位アドレス)セット */
SSU.SSTDR3 = 0x00;                   /* 送信データ(ダミーデータ)セット */
while(SSU.SSSR.BIT.RDRF !=1);        /* RDRF=1の確認 */
while(SSU.SSSR.BIT.ORER ==1);        /* ORER≠1の確認 */
ssu_rx_data0 = SSU.SSRDR0;           /* ダミーデータ */
ssu_rx_data1 = SSU.SSRDR1;           /* ダミーデータ */
ssu_rx_data2 = SSU.SSRDR2;           /* ダミーデータ */
ssu_rx_data3 = SSU.SSRDR3;           /* EEPROMからの読み出しデータ */
while(SSU.SSSR.BIT.TEND !=1);        /* TEND=1の確認 */
SSU.SSSR.BIT.TEND = 0;               /* TENDクリア */
wait_1bit();                          /* 1bit 期間経過 */
SSU.SSER.BYTE = 0x00;                /* 送受信を禁止 */
}
    
```

## 5. 参考ドキュメント

ハードウェアマニュアル

SH7147 ハードウェアマニュアル Rev.2.00

(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。)

## 6. 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://www.renesas.com/>

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2006.10.12	—	初版発行
2.00	2007.09.18	—	第 2 版発行

### 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事務の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」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頂きますとともに、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下さい。
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下さい。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下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 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下さい。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）を行なうもの。
  - 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致します。
  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致します。
  13. 本資料に関する詳細についてのお問い合わせ、その他お気づきの点等がございましたら弊社営業窓口までご照会下さい。