

RX610 グループ

R01AN0897JJ0100

Rev.1.00

2012.03.27

RPDL を使用した Serial EEPROM 制御

要旨

本アプリケーションノートでは、RX610 グループの「Renesas Peripheral Driver Library」のシリアルコミュニケーションインタフェース API (SCI クロック同期式シングルマスタ制御) と「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」を使用した Serial EEPROM 制御について説明します。

なお、本アプリケーションノートでは以下のアプリケーションノートおよびライブラリを使用しています。

- Serial EEPROM 制御 :
「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」 Rev.1.03 (R01AN0565JJ0103)
- SCI クロック同期式シングルマスタ制御 :
「Renesas Peripheral Driver Library」 (R20UT0083EE0100)

対象デバイス

対応 MCU RX610 グループ

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

目次

| | |
|----------------------------------|----|
| 1. 仕様 | 2 |
| 2. 動作確認条件 | 3 |
| 3. 関連アプリケーションノートおよび関連ライブラリ | 3 |
| 4. ハードウェア説明 | 4 |
| 5. ソフトウェア説明 | 5 |
| 6. サンプルコード | 26 |
| 7. 参考ドキュメント | 26 |

1. 仕様

本アプリケーションノートのサンプルコードは、図 1.1 の使用例で示した RX610-RSK にシリアルペリフェラルインタフェースの Serial EEPROM (ルネサスエレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ) を接続した状態で動作します。

本アプリケーションノートでは、RX610-RSK に外付けした Serial EEPROM に対して、「Renesas Peripheral Driver Library」(以下 RPDL) の SCI クロック同期式シングルマスタ制御と「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」(以下 Serial EEPROM 制御ミドルウェア) を用いて RX610 のクロック同期式シリアル通信による EEPROM への書き込み/読み出し等を行います。

以下に、通信機能の概略を示します。

- RX610 のクロック同期式 (3 線式) シリアル通信機能 (チャンネルは SCI1 に固定) を使用しています。
- Serial EEPROM のチップセレクトは汎用ポート (P66) で制御しています。
- クロック同期式シリアル通信の転送ビットレートは 3125000bps に設定しています。
- Serial EEPROM への書き込み、読み出しには SCI のクロック同期式の送受信モードを使用し、読み出し時のみ SCI の RXI 割り込みを許可することで読み出し処理を行います。

なお、本アプリケーションノートのサンプルコードはリトルエンディアンでの動作のみ可能です。

表 1.1 使用する周辺機器と用途

| 周辺機器 | 用途 |
|----------------------------|---------------------|
| SCI (シリアルコミュニケーションインタフェース) | クロック同期式 (3 線式) シリアル |

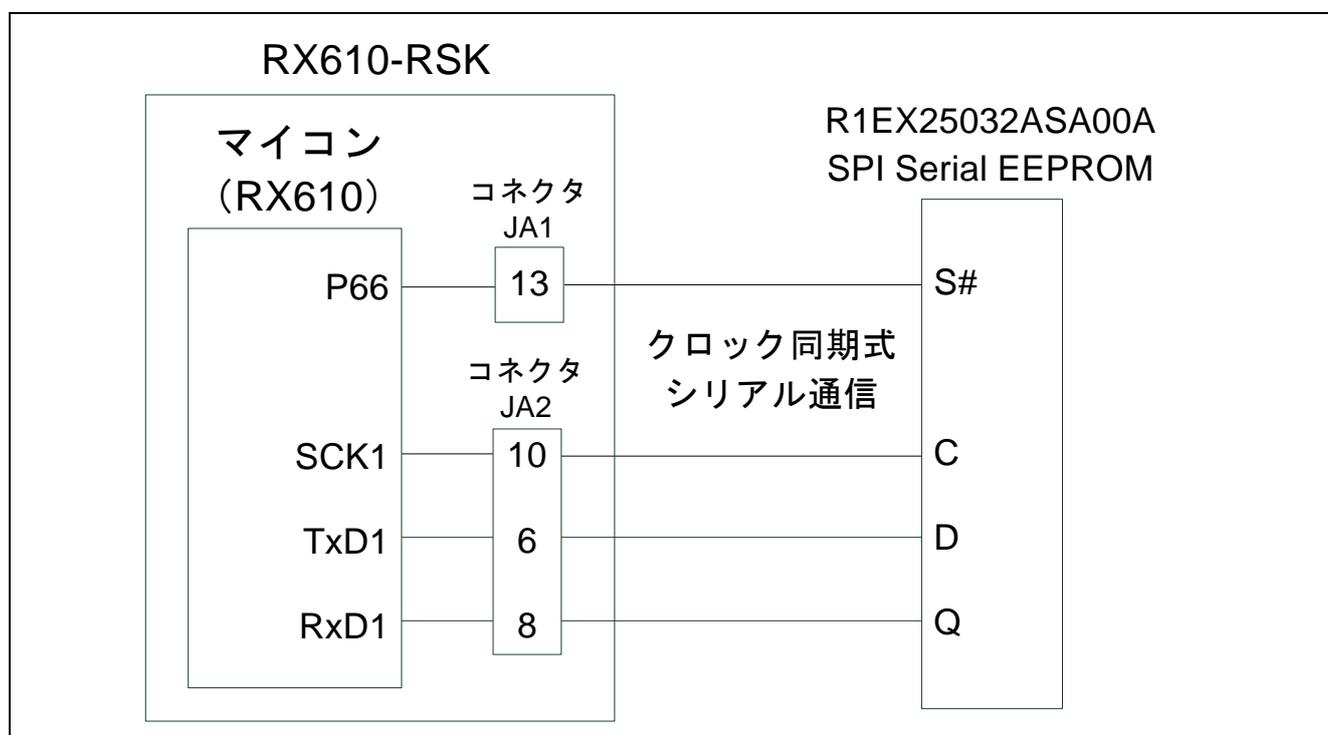


図 1.1 使用例

2. 動作確認条件

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

表2.1 動作確認条件

| 項目 | 内容 |
|---------------|---|
| 使用マイコン | RX610 グループ (R5F56108VNFP) |
| 使用デバイス | ルネサス エレクトロニクス製 R1EX25032ASA00A SPI Serial EEPROM |
| 動作周波数 | <ul style="list-style-type: none"> ● EXTAL : 12.5MHz ● ICLK : 100MHz ● PCLK : 50MHz ● BCLK : 25MHz |
| SCI 動作環境* | <ul style="list-style-type: none"> ● クロック同期式 (3 線式) シリアル ● 使用チャネル : チャネル 1 ● ビットレート : 3125000bps ● MSB ファースト転送 |
| 動作電圧 | 3.3V |
| 総合開発環境 | ルネサス エレクトロニクス製 High-performance embedded Workshop Version 4.09.00.007 |
| C コンパイラ | ルネサス エレクトロニクス製 RX Standard Toolchain Version 1.1.0.0 コンパイルオプション -cpu=rx600 -patch=rx610 -include="\$(PROJDIR)¥RPDL", "\$(HEWDIR)¥." -output=obj="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo |
| エンディアン | リトルエンディアン |
| サンプルコードのバージョン | Ver.1.00 |
| 評価に使用したソフトウェア | ルネサス エレクトロニクス製 R1EX25xxx シリーズの SPI Serial EEPROM 制御ソフトウェア Ver.2.00 |
| 評価に使用したボード | Renesas Starter Kit for RX610 |

【注】 * 設定値については、Serial EEPROM 制御ミドルウェアのデフォルト値を使用しています。

3. 関連アプリケーションノートおよび関連ライブラリ

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およびライブラリのマニュアルを以下に示します。合わせて参照してください。

- Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア Rev.1.03 (R01AN0565JJ)
- RX610 Group Peripheral Driver Library User's Manual(R20UT0083EE0100)

4. ハードウェア説明

4.1 使用端子一覧

表 4.1 に使用端子と機能を示します。

表4.1 使用端子と機能

| 端子名 | 入出力 | 内容 |
|------|-----|----------------|
| SCK1 | 出力 | クロック出力 |
| TxD1 | 出力 | マスタデータ出力 |
| RxD1 | 入力 | マスタデータ入力 |
| P66 | 出力 | スレーブデバイスセレクト出力 |

4.2 参考回路

図 4.1 に RX610 SCI1 と R1EX25032ASA00A SPI Serial EEPROM との接続例を示します。

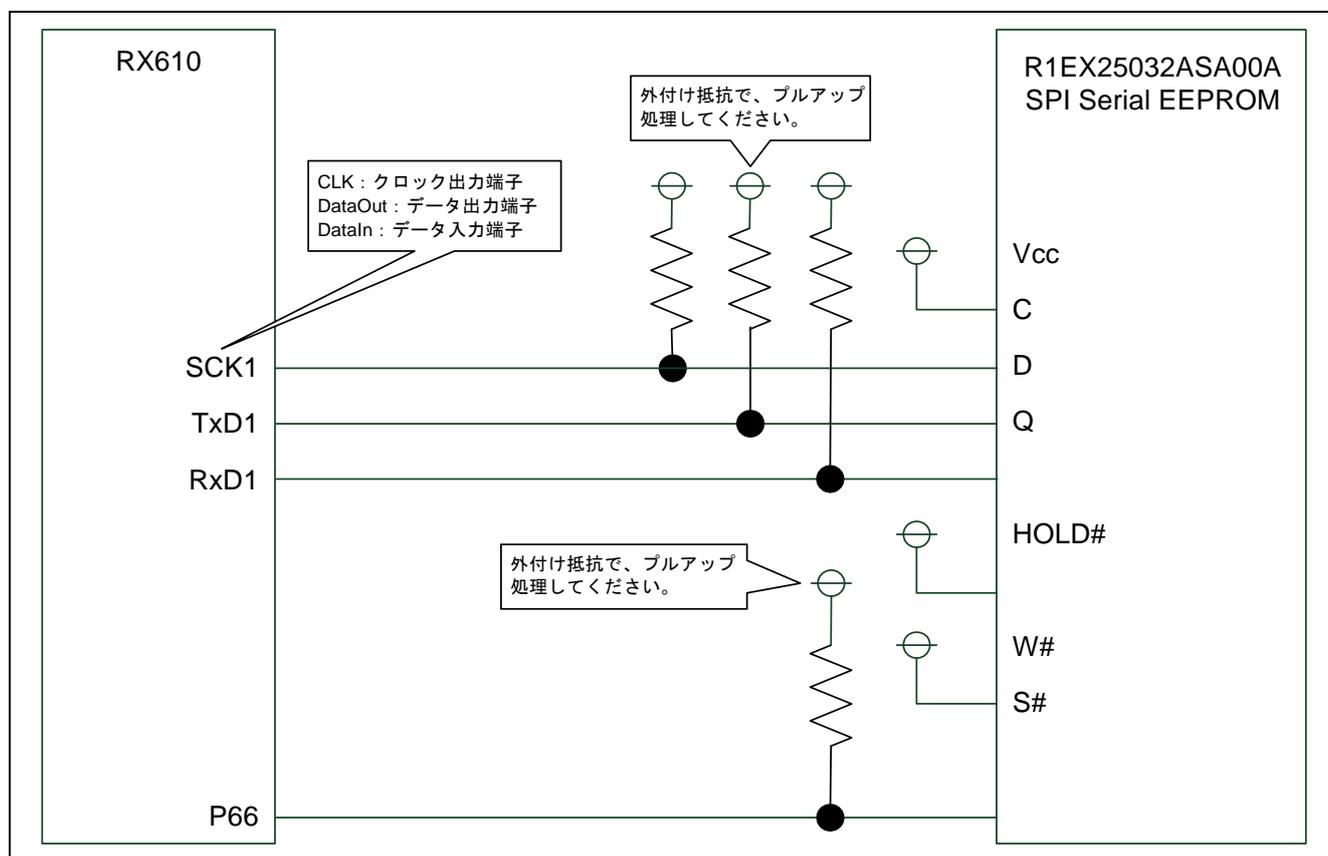


図4.1 RX610 SCI1 と R1EX25032ASA00A SPI Serial EEPROM との接続例

5. ソフトウェア説明

本アプリケーションノートでは、主に RPDL と Serial EEPROM 制御ミドルウェアの結合を説明します。

5.1 動作概要

SCI のクロック同期式（3 線式）シリアル通信機能を使って、シリアルペリフェラルインタフェース Serial EEPROM 制御を実現します。

本アプリケーションノートでは、以下の制御を行っています。

- データの入出力はクロック同期式モード（内部クロック使用）で制御する。
- SCI のクロック同期式（3 線式）シリアル通信は、RPDL のシリアルコミュニケーションインタフェース API を使用して制御する。
- Serial EEPROM の制御は Serial EEPROM 制御ミドルウェアを使用する。

なお、Serial EEPROM 制御ミドルウェア用に、下位層として「クロック同期式シングルマスタ制御ソフトウェア」を準備していますが、本アプリケーションノートでは下位層として RPDL のシリアルコミュニケーションインタフェース API を使用しています。

Serial EEPROM 制御ミドルウェアと RPDL の結合は、Serial EEPROM 制御ミドルウェアが RPDL の関数をコールすることで実現します。詳細については、表 5.10 関数対応表を参照してください。

5.2 必要メモリサイズ

表 5.1 必要メモリサイズに必要メモリサイズを示します。

表5.1 必要メモリサイズ

| 使用メモリ | サイズ | 備考 |
|--------------|----------|--------------------------|
| ROM | 3184 バイト | ROM の合計サイズ |
| RAM | 1352 バイト | RAM の合計サイズ |
| 最大使用ユーザスタック | 192 バイト | R_SPI_EEP_Wait_WBusy 関数 |
| 最大使用割り込みスタック | 36 バイト | Interrupt_SCI1_RXI1 割り込み |

必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。

5.3 ファイル構成

表 5.2に本アプリケーションノートのファイル構成を示します。なお、統合開発環境で自動生成されるファイルおよびRPDLで提供されているファイルは除きます。

RPDLで提供されているファイルの生成方法および使用方法の詳細については、「RX610 Group Peripheral Driver Library User's Manual」の「Using the library within your project」を参照してください。

表5.2 ファイル構成

| ファイル名 | 概要 | 備考 |
|-------------------------------|--|--|
| testmain.c | Serial EEPROM 制御ミドルウェアのモジュールテストプログラム | Serial EEPROM 制御ミドルウェアで提供している動作確認用サンプルプログラムを使用しています。 |
| R_SIO_sci_rx.c | RPDL 制御モジュール | Serial EEPROM 制御ミドルウェアが SCI のクロック同期式シリアル通信を行う際に呼び出す関数と RPDL の制御を行う関数の関数名を一致させています。 |
| R_SIO.h | RPDL 制御モジュールのインクルードヘッダ | R_SIO_sci_rx.c の定数定義およびモジュールのプロトタイプ宣言をします。 |
| mtl_com.c | Serial EEPROM 制御ミドルウェアが使用するデバッグ用モジュール | Serial EEPROM 制御ミドルウェアのデバッグ用エラーログの取得/クリアを行います。 |
| mtl_com.h | Serial EEPROM 制御ミドルウェアの設定選択ヘッダ | Serial EEPROM 制御ミドルウェアの設定選択用定義およびプロトタイプ宣言を行います。 |
| mtl_tim.c | Serial EEPROM 制御ミドルウェアのループタイマ関連モジュール | |
| mtl_tim.h | Serial EEPROM 制御ミドルウェアのループタイマ関連モジュールのインクルードヘッダ | |
| R_SPI_EEP.h ^{*1} | Serial EEPROM 制御ミドルウェアのヘッダ | 詳細は「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |
| R_SPI_EEP_io.c ^{*2} | Serial EEPROM 制御ミドルウェアの I/O モジュール | 詳細は「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |
| R_SPI_EEP_io.h | Serial EEPROM 制御ミドルウェアの I/O モジュール用定義ヘッダ | 詳細は「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |
| R_SPI_EEP_sfr.h ^{*3} | Serial EEPROM 制御ミドルウェアのレジスタ用定義ヘッダ | 詳細は「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |
| R_SPI_EEP_usr.c | Serial EEPROM 制御ミドルウェアのユーザ I/F モジュール | 詳細は「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |

【注】 *1 R_SPI_EEP.h 内に「#include <stdint.h>」を追加しています。

*2 R_SPI_EEP_io.c 内の「#include "iodefine.h"」を追加しています。

また「R_SPI_EEP_Write_Di」関数をコメントにしています。

*3 本アプリケーションノートは、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートで提供されるヘッダファイル「R_SPI_EEP_sfr.h.rx610」をアプリケーションノートの指示に基づき、「R_SPI_EEP_sfr.h」にリネームしています。

また、「R_SPI_EEP_sfr.h」内で定義されている EEP_BR の値を以下に変更しています。

```
変更前：#define EEP_BR          (uint8_t)0x03          /* BRR initial setting */
変更後：#define EEP_BR          (uint32_t)3125000       /* bit rate setting */
```

Serial EEPROM 制御ミドルウェアでは EEP_BR にビットレートレジスタ (BRR) のレジスタへの設定値を定義する仕様となっていますが、本アプリケーションノートでは RPDL の仕様に合わせて、EEP_BR にビットレート値を定義する仕様に変更しました。

5.4 定数一覧

5.4.1 戻り値

表 5.3に、本アプリケーションノートで使用する戻り値を示します。

表5.3 戻り値

| 定数名 | 設定値 | 内容 |
|--------|--------------|--|
| SIO_OK | (error_t)(0) | <ul style="list-style-type: none"> 使用関数 : R_SIO_Disable R_SIO_Enable R_SIO_Tx_Data R_SIO_Rx_Data Serial EEPROM 制御ミドルウェアと作成した関数を組み合わせる上で、SIO_OK のみ必要です。 |

5.4.2 各種定義

表 5.4に、Serial EEPROM制御ミドルウェアで提供しているテストプログラムの定数値を示します。なお、Serial EEPROM制御ミドルウェアで定義している値に関しては、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。

表5.4 Serial EEPROM 制御ミドルウェアで提供しているテストプログラムの定数値

| 定数名 | 設定値 | 内容 |
|--------------|-----|-------------------------|
| EEP_BUF_SIZE | 128 | Cbuf1、Cbuf2 のバッファサイズです。 |

5.5 変数一覧

表 5.5にグローバル変数を、表 5.6にstatic型変数を示します。

なお、Serial EEPROM 制御ミドルウェア内の変数に関しては、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。

表5.5 グローバル変数

| Type | Variable Name | Contents | Function Used |
|-----------|---------------|---------------------------------|---------------------|
| uchar FAR | Cbuf1 | Serial EEPROM への書き込みデータ格納用バッファ | Test200, Test300 |
| uchar FAR | Cbuf2 | Serial EEPROM からの読み出しデータ格納用バッファ | Test300 |
| uchar | gDevNo | Serial EEPROM のデバイス番号* | Test200, Test300 |
| ulong | gAddr | Serial EEPROM への書き込み開始アドレス | Test300 |
| uchar | gStat | Serial EEPROM の読み出しステータス格納バッファ | Test200 |

【注】 * Serial EEPROM 制御ミドルウェアは 2 つのデバイスを選択できる仕様としているため、用意している。本アプリケーションノートでは、1 つのデバイスのみを使用するので、gDevNo は 0 に固定しています。

表5.6 static 型変数

| Type | Variable Name | Contents | Function Used |
|---------------|---------------|------------------|--------------------|
| short | Ret | 戻り値格納用変数 | Test200 Test300 |
| unsigned char | Tr_Dummy | ダミー送信データ格納用バッファ* | R_SIO_Rx_Data |

【注】 * Tr_Dummy は RPDL での受信処理に必要なバッファであり、本アプリケーションノートで追加したバッファです。

5.6 関数構造図

本アプリケーションノートのプログラムの関数構造図を図 5.1～図 5.6 に示します。

Serial EEPROM 制御ミドルウェアが提供している関数は黒字、本アプリケーションノートで作成した関数は赤字、RPDL で提供している関数は青字でそれぞれ示しています。

本アプリケーションノートで作成した関数および RPDL で提供している関数の詳細については「5.9 関数仕様」を参照してください。

また、Serial EEPROM 制御ミドルウェアが提供している関数に関しては、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。

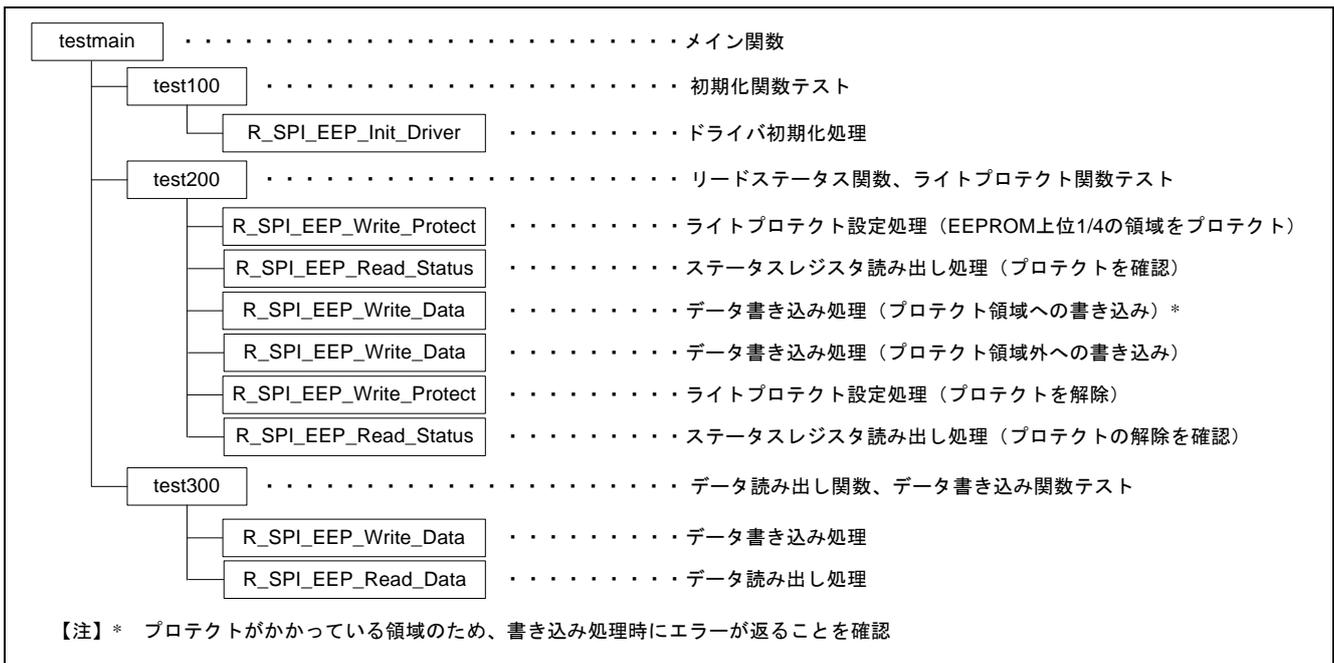


図5.1 testmain 関数

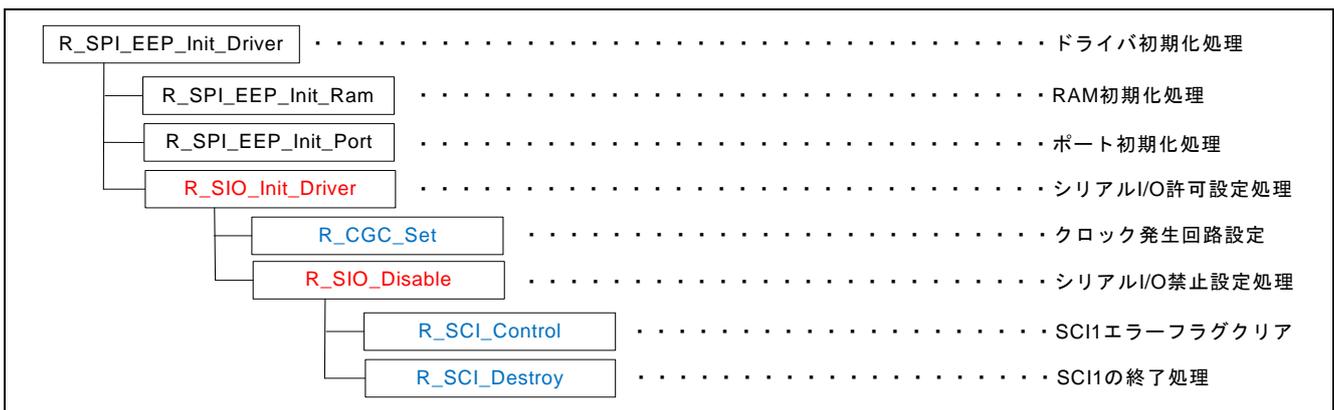


図5.2 R_SPI_EEP_Init_Driver 関数

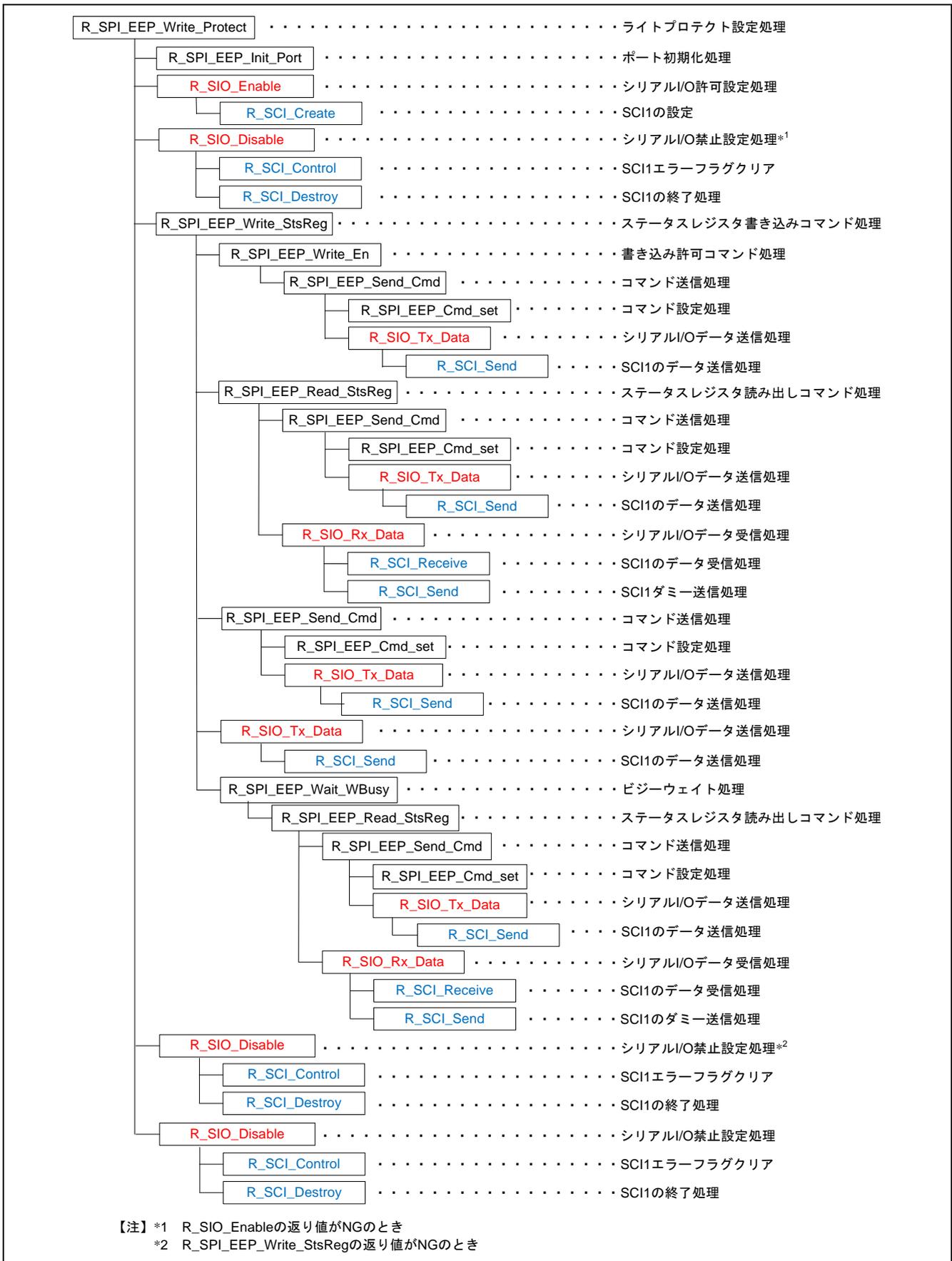


図5.3 R_SPI_EEP_Write_Protect関数

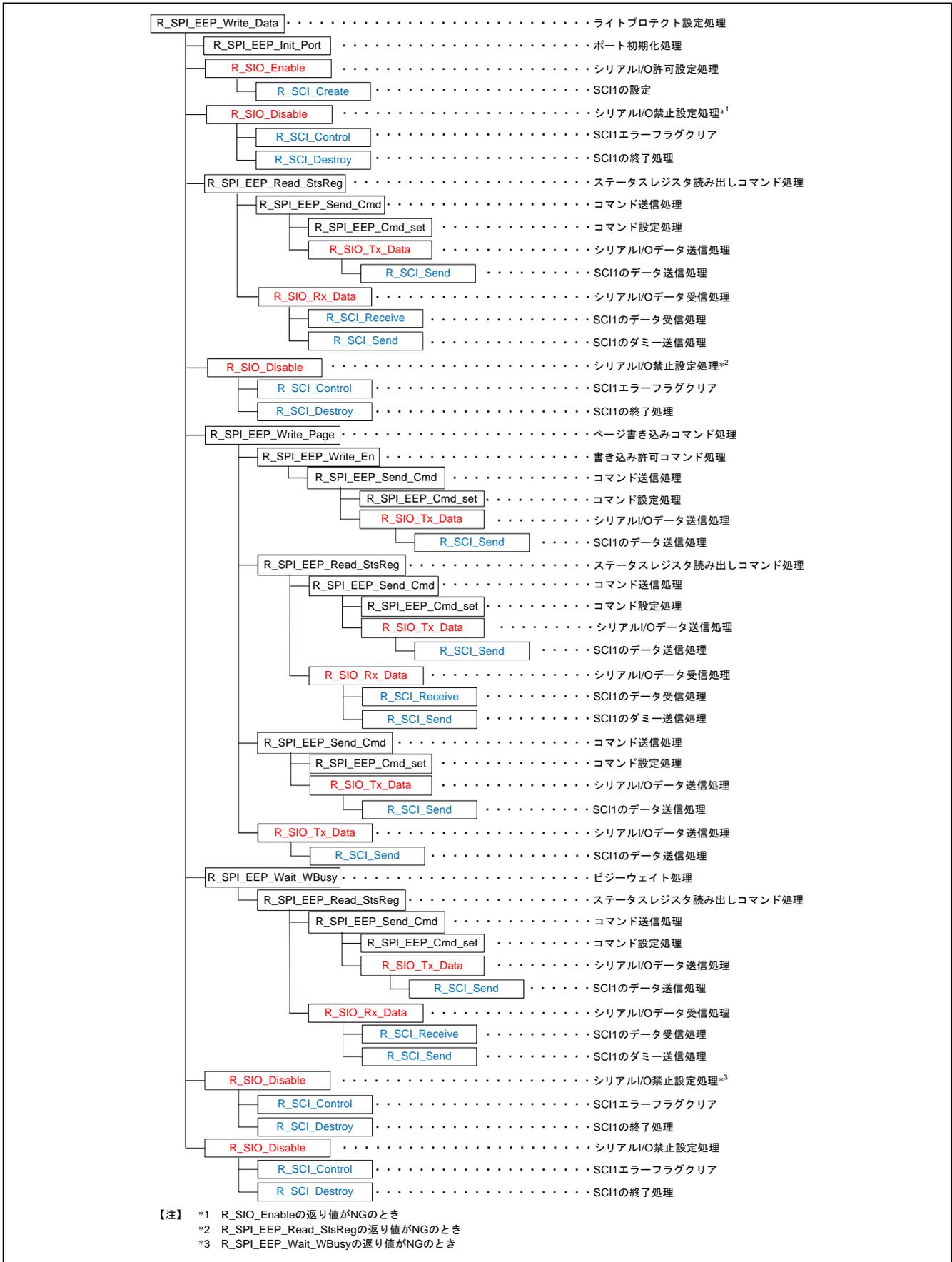


図5.4 R_SPI_EEP_Write_Data 関数

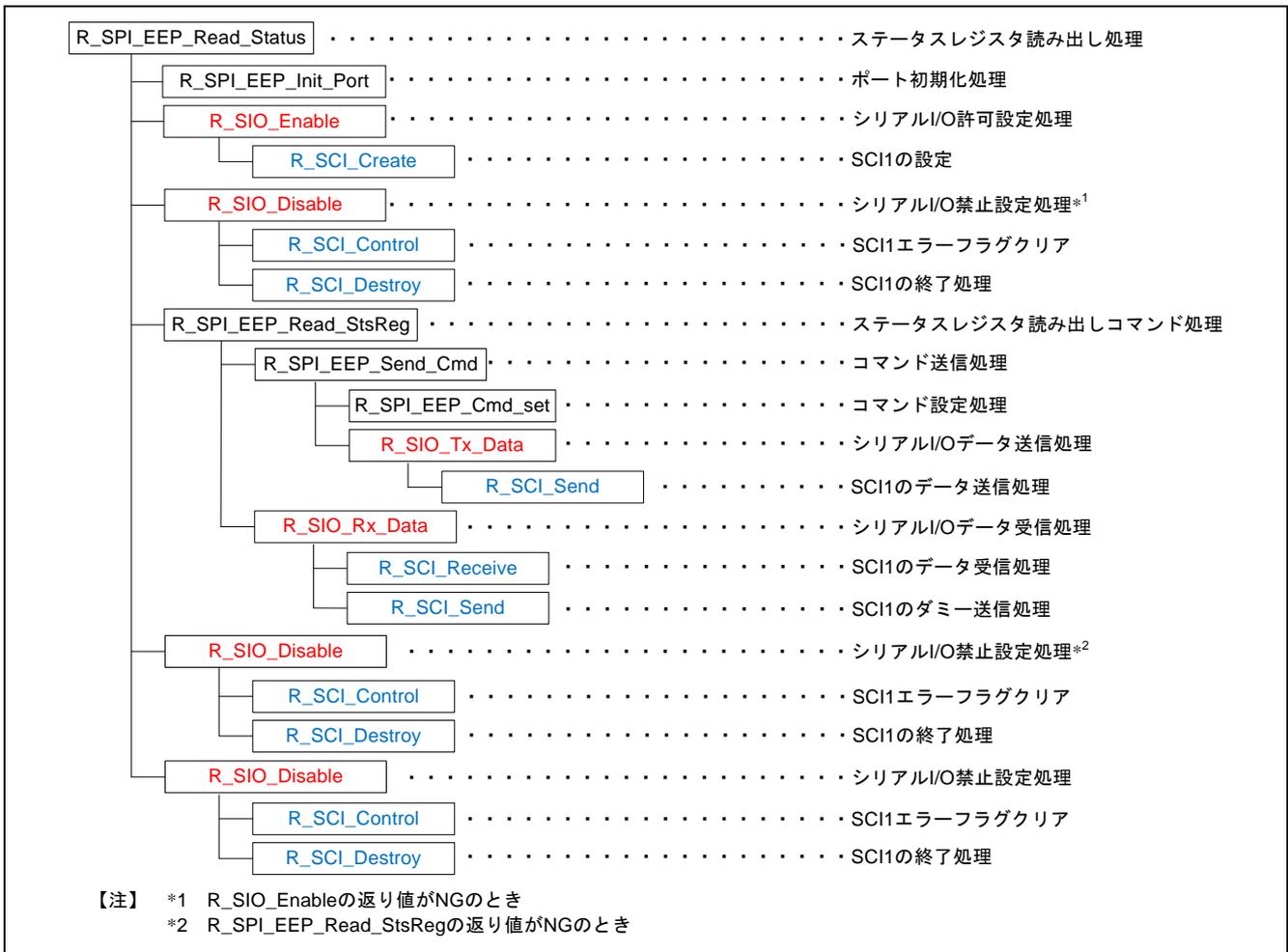


図5.5 R_SPI_EEP_Read_Status 関数

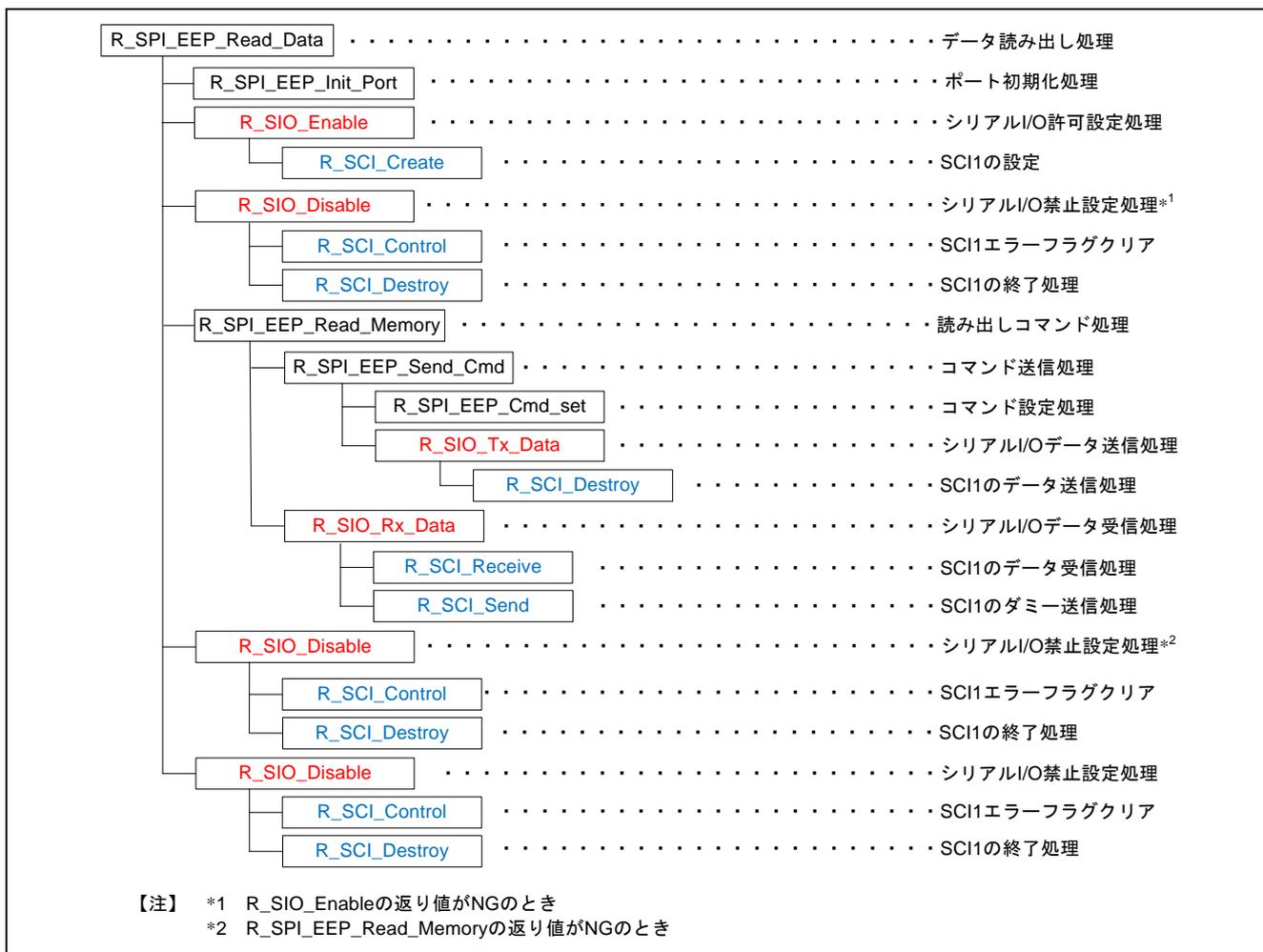


図5.6 R_SPI_EEP_Read_Data 関数

5.7 関数一覧

5.7.1 本アプリケーションノートで作成した関数

表 5.7に本アプリケーションノートで作成した関数を示します。各関数の詳細については「5.9.1 本アプリケーションノートで作成した関数の仕様」を参照してください。

表5.7 本アプリケーションノートで作成した関数

| 関数名 | 概要 |
|-------------------|------------------|
| R_SIO_Init_Driver | ドライバ初期化処理 |
| R_SIO_Disable | シリアル I/O 禁止設定処理 |
| R_SIO_Enable | シリアル I/O 許可設定処理 |
| R_SIO_Tx_Data | シリアル I/O データ送信処理 |
| R_SIO_Rx_Data | シリアル I/O データ受信処理 |
| SCI1RxFunc | 受信処理用コールバック関数 |

5.7.2 Serial EEPROM制御ミドルウェアで提供されている関数

表 5.8にSerial EEPROM制御ミドルウェアで提供されている関数を示します。各関数の詳細に関しては、「Renesas R1EX25xxxシリーズ Serial EEPROM制御ソフトウェア」のアプリケーションノートを参照してください。

表5.8 Serial EEPROM 制御ミドルウェアで提供されている関数

| 関数名 | 概要 |
|-------------------------|--------------------------|
| testmain | テストプログラム |
| test100 | 初期化関数テスト |
| test200 | リードステータス関数、ライトプロテクト関数テスト |
| test300 | データ読み出し関数、データ書き込み関数テスト |
| Trap | 無限ループ処理 |
| R_SPI_EEP_Init_Driver | ドライバ初期化処理 |
| R_SPI_EEP_Write_Protect | ライトプロテクト設定処理 |
| R_SPI_EEP_Read_Status | ステータス読み出し処理 |
| R_SPI_EEP_Read_Data | データ読み出し処理 |
| R_SPI_EEP_Write_Data | データ書き込み処理 |
| R_SPI_EEP_Init_Ram | ポート初期化処理 |
| R_SPI_EEP_Init_Port | ポート初期化処理 |
| R_SPI_EEP_Write_StsReg | ステータスレジスタ書き込みコマンド処理 |
| R_SPI_EEP_Read_StsReg | ステータスレジスタ読み出しコマンド処理 |
| R_SPI_EEP_Write_Page | ページ書き込みコマンド処理 |
| R_SPI_EEP_Wait_WBusy | ビジーウェイト処理 |
| R_SPI_EEP_Read_Memory | 読み出しコマンド処理 |
| R_SPI_EEP_Send_Cmd | コマンド送信処理 |
| R_SPI_EEP_Cmd_set | コマンド設定処理 |
| R_SPI_EEP_Write_En | 書き込み許可コマンド処理 |
| R_SPI_EEP_Write_Di | 書き込み禁止コマンド処理 |

5.7.3 使用しているRPDLの関数

表 5.9に使用しているRPDLの関数を示します。各関数の詳細に関しては、「5.9.2 使用しているRPDLの関数仕様」を参照してください。

表5.9 使用している RPDL の関数

| 関数名 | 概要 |
|---------------|------------------|
| R_SCI_Create | SCI チャンネル設定 |
| R_SCI_Destroy | SCI チャンネル終了 |
| R_SCI_Send | SCI チャンネルでデータを送信 |
| R_SCI_Receive | SCI チャンネルでデータを受信 |
| R_SCI_Control | SCI チャンネルを制御 |

5.8 関数対応表

以下に Serial EEPROM 制御ミドルウェアがコールする関数と呼び出される RPDL の関数および RPDL 関数が行う処理についての関数対応表を示します。

表5.10 関数対応表

| Serial EEPROM制御ミドルウェアで コールされる関数 | 関数内に含まれるRPDLの関数 | RPDLの関数が行う処理 |
|------------------------------------|-----------------|--------------|
| R_SIO_Init_Driver | R_CGC_Set | クロックの設定 |
| R_SIO_Disable | R_SCI_Control | エラーフラグのクリア |
| | R_SCI_Destroy | SCIの終了処理 |
| R_SIO_Enable | R_SCI_Cleate | SCIの設定 |
| R_SIO_Tx_Data | R_SCI_Send | SCIデータ送信処理 |
| R_SIO_Rx_Data | R_SCI_Receive | SCI受信設定 |
| | R_SCI_Send | SCIダミー送信処理 |

5.9 関数仕様

5.9.1 本アプリケーションノートで作成した関数の仕様

以下に本アプリケーションノートで作成した関数の仕様を示します。

| R_SIO_Init_Driver | |
|-------------------|---|
| 概要 | ドライバ初期化処理 |
| ヘッダ | R_SIO.h |
| 宣言 | error_t R_SIO_Init_Driver(void) |
| 説明 | <ul style="list-style-type: none"> • R_CGC_Set 関数 (RPDL 提供) を使用して、クロックの設定を行います。 • R_SIO_Disable 関数を使用して、ドライバの初期化を行います。 |
| 引数 | なし |
| リターン値 | • SIO_OK ; Successful operation |
| 備考 | <ul style="list-style-type: none"> • ドライバ初期化前の SCI 通信状態を含めて、R_SIO_Disable 関数を使用して以下の処理を行います。 • 送信/受信を停止します。 • SSR の PER フラグ/FER フラグ/OERE フラグをクリアします。 |

| R_SIO_Disable | |
|---------------|---|
| 概要 | シリアル I/O 禁止設定処理 |
| ヘッダ | R_SIO.h |
| 宣言 | error_t R_SIO_Disable(void) |
| 説明 | <ul style="list-style-type: none"> • R_SCI_Control 関数 (RPDL 提供) を使用して、SSR の PER フラグ/FER フラグ/OERE フラグをクリアします。 • R_SCI_Destroy 関数 (RPDL 提供) を使用して、SCI の送信/受信を停止します。 |
| 引数 | • なし |
| リターン値 | • SIO_OK ; Successful operation |
| 備考 | • R_SCI_Control 関数および R_SCI_Destroy 関数の詳細については、「5.9.2 使用している RPDL の関数仕様」を参照してください。 |

| R_SIO_Enable | |
|--------------|--|
| 概要 | シリアル I/O 許可設定処理 |
| ヘッダ | R_SIO.h |
| 宣言 | error_t R_SIO_Enable(uint8_t BrgData) |
| 説明 | <ul style="list-style-type: none"> • R_SCI_Create 関数 (RPDL 提供) を使用して、シリアル I/O 機能の有効化、シリアル I/O のモジュールストップの解除およびビットレートを設定します。 |
| 引数 | • uint32_t BrgData ; ビットレート設定値 |
| リターン値 | • SIO_OK ; Successful operation |
| 備考 | • R_SCI_Create 関数の詳細については、「5.9.2 使用している RPDL の関数仕様」を参照してください。 |

| R_SIO_Tx_Data | |
|---------------|---|
| 概要 | シリアル I/O データ送信処理 |
| ヘッダ | R_SIO.h |
| 宣言 | error_t R_SIO_Tx_Data(uint16_t TxCnt, uint8_t FAR* pData) |
| 説明 | <ul style="list-style-type: none"> R_SCI_Send 関数 (RPDL 提供) を使用して、引数が示すデータを指定バイト数分送信します。 |
| 引数 | <ul style="list-style-type: none"> uint16_t TxCnt ;送信バイト数 uint8_t FAR* pData ;送信データ格納バッファポインタ |
| リターン値 | <ul style="list-style-type: none"> SIO_OK ; Successful operation |
| 備考 | <ul style="list-style-type: none"> R_SCI_Send 関数の詳細については、「5.9.2 使用している RPDL の関数仕様」を参照してください。 |

| R_SIO_Rx_Data | |
|---------------|---|
| 概要 | シリアル I/O データ受信処理 |
| ヘッダ | R_SIO.h |
| 宣言 | error_t R_SIO_Rx_Data(uint16_t RxCnt, uint8_t FAR* pData) |
| 説明 | <ul style="list-style-type: none"> R_SCI_Receive 関数 (RPDL 提供) を使用して、受信バイト数および受信データ格納バッファポインタの設定を行います。 R_SCI_Send 関数 (RPDL 提供) を使用して、ダミー送信を行います。 |
| 引数 | <ul style="list-style-type: none"> uint16_t RxCnt ;受信バイト数 uint8_t FAR* pData ;受信データ格納バッファポインタ |
| リターン値 | <ul style="list-style-type: none"> SIO_OK ; Successful operation |
| 備考 | <ul style="list-style-type: none"> 受信処理前に受信回数分、関数内のダミー送信用バッファに最大受信バイト分の 0xFF を格納します。 |

| SCI1RxFunc | |
|------------|---|
| 概要 | 受信処理用コールバック関数 |
| ヘッダ | |
| 宣言 | void SCI1RxFunc(void) |
| 説明 | <ul style="list-style-type: none"> 受信処理時のコールバック関数です。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | <ul style="list-style-type: none"> 全データ受信後にコールバックが実行されます。 関数内で行う処理はありません。 |

5.9.2 使用しているRPDLの関数仕様

以下に、本アプリケーションノートで使用している RPDL の関数仕様を示します。

| R_CGC_Set | |
|-----------|---|
| 概要 | クロック発生回路の設定 |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | bool R_CGC_Set(uint32_t, uint32_t, uint32_t, uint32_t, uint8_t); |
| 説明 | <ul style="list-style-type: none"> RPDL を使用して SCI の設定を行う際、最初に本関数でクロック設定を行う必要があります。 入力周波数、ICLK、PCLK、BCLK の設定およびクロック発生回路関連のオプションの設定を行います。 |
| 引数 | <ul style="list-style-type: none"> 第一引数：12.5E6 ;入力周波数：12.5MHz 第二引数：100E6 ;システムクロック周波数：100MHz 第三引数：50E6 ;周辺モジュールクロック周波数：50MHz 第四引数：25E6 ;バスクロック周波数：25MHz 第五引数： ;設定オプション：BCLK端子を入力端子とする PDL_CGC_BCLK_DISABLE |
| リターン値 | <ul style="list-style-type: none"> 全パラメータが正しくかつ排他的に指定されている場合：True (1) 正しく指定されていない場合：False (0) |
| 備考 | <ul style="list-style-type: none"> RX610 用のRPDLでは、クロック設定用に本関数を用意しています。関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Clock Generation Circuit」で確認してください。 本関数の引数の詳細については、「RX610 Group Peripheral Driver Library User's Manual」の「Clock Generation Circuit」内の「R_CGC_Set」を参照してください。 |

| R_SCI_Control | |
|---------------|--|
| 概要 | SCIの制御（SCIの各エラーフラグのクリア） |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | bool R_SCI_Control(uint8_t data1, uint8_t data2); |
| 説明 | <ul style="list-style-type: none"> SCI の各種受信エラーフラグをクリアします。 |
| 引数 | <ul style="list-style-type: none"> 第一引数：1 ;制御を行うSCIのチャンネル 第二引数： ;制御オプション： PDL_SCI_CLEAR_RECEIVE_ER 受信エラーフラグをクリア ROR_FLAGS |
| リターン値 | <ul style="list-style-type: none"> 全パラメータが正しく指定されている場合：True (1) 正しく指定されていない場合：False (0) |
| 備考 | <ul style="list-style-type: none"> RX610 用のRPDLでは、SCIの設定用に本関数を含めて6つの関数を用意しています。それぞれの関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Serial Communication Interface」で確認してください。 本関数の引数の詳細については、同マニュアル内の「Serial Communication Interface」の「R_SCI_Control」を参照してください。 |

| R_SCI_Destroy | |
|---------------|--|
| 概要 | SCIの終了処理 |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | bool R_SCI_Destroy(uint8_t data); |
| 説明 | <ul style="list-style-type: none"> SCI の転送を禁止し、SCI をモジュールストップの状態に設定します。 【注】 SMR レジスタの初期化を行わないため、SCI 再起動時の転送方式は前回使用した転送方式です。 |
| 引数 | <ul style="list-style-type: none"> 第一引数 : 1 ;転送を禁止するSCIのチャンネル |
| リターン値 | <ul style="list-style-type: none"> 全パラメータが正しくかつ排他的に指定されている場合 : True (1) 正しく指定されていない場合 : False (0) |
| 備考 | <ul style="list-style-type: none"> RX610 用のRPDLでは、SCIの設定用に本関数を含めて6つの関数を用意しています。それぞれの関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Serial Communication Interface」で確認してください。 本関数の引数の詳細については、同マニュアル内の「Serial Communication Interface」の「R_SCI_Destroy」を参照してください。 |

| R_SCI_Create | |
|--------------|---|
| 概要 | SCIの設定 |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | bool R_SCI_Create(uint8_t data1, uint32_t data2, uint32_t data3, uint8_t data4); |
| 説明 | <ul style="list-style-type: none"> SCI を有効にします。 SCI に関する以下の設定を行います。 【転送方式、TxDn 端子、RxDn 端子、データ転送方式、転送ビットレート、SCI 割り込みの優先レベル】 本関数は SCI 転送を行う前に設定します。 本関数の変更は、SCI 転送を禁止した状態で設定します。 |
| 引数 | <ul style="list-style-type: none"> 第一引数 : 1 ;選択するSCIのチャンネル 第二引数 : ;チャンネルの設定 : PDL_SCI_SYNC ;クロック同期式、 PDL_SCI_TX_CONNECTED ;TxDn端子の出力は必要、 PDL_SCI_RX_CONNECTED ;RxDn端子の入力は必要、 PDL_SCI_MSB_FIRST ;MSBファーストで転送、 PDL_SCI_CLK_INT_OUT ;SCKn端子はビットクロックを出力 第三引数 : BrgData ;転送ビットレート : 3.125MHz 第四引数 : 1 ;SCIの割り込み優先レベル |
| リターン値 | <ul style="list-style-type: none"> 全パラメータが正しくかつ排他的に指定されている場合 : True (1) 正しく指定されていない場合 : False (0) |
| 備考 | <ul style="list-style-type: none"> RX610 用のRPDLでは、SCIの設定用に本関数を含めて6つの関数を用意しています。それぞれの関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Serial Communication Interface」で確認してください。 本関数の引数の詳細については、同マニュアル内の「Serial Communication Interface」の「R_SCI_Create」を参照してください。 |

| R_SCI_Send | |
|------------|---|
| 概要 | SCIのデータ送信処理 |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | bool R_SCI_Send(uint8_t data1, uint16_t data2, uint8_t * data3, uint16_t data4, void * func); |
| 説明 | <ul style="list-style-type: none"> • SCI 送信処理を行います。 • 第四引数で設定した送信回数分、第三引数のバッファ内のデータを送信します。 • 第五引数でコールバック関数を設定しない場合は、送信完了をポーリングで転送処理を行います。送信処理が全て完了すると関数から戻ります。 |
| 引数 | <ul style="list-style-type: none"> • 第一引数 : 1 ;送信用SCIチャンネルの選択 • 第二引数 : PDL_NO_DATA ;制御オプション : デフォルト設定 • 第三引数 : pData ;送信データ格納用バッファ • 第四引数 : TxCnt ;総送信回数 • 第五引数 : PDL_NO_FUNC ;コールバック関数 : なし |
| リターン値 | <ul style="list-style-type: none"> • 全パラメータが正しくかつ排他的に指定されている場合 : True (1) • 正しく指定されていない場合 : False (0) |
| 備考 | <ul style="list-style-type: none"> • RX610 用のRPDLでは、SCIの設定用に本関数を含めて6つの関数を用意しています。それぞれの関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Serial Communication Interface」で確認してください。 • 本関数の引数の詳細については、同マニュアル内の「Serial Communication Interface」の「R_SCI_Send」を参照してください。 |

| R_SCI_Receive | |
|---------------|--|
| 概要 | SCIのデータ受信設定 |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | <pre>bool R_SCI_Receive(uint8_t data1, uint16_t data2, uint8_t * data3, uint16_t data4, void * func1, void * func2);</pre> |
| 説明 | <ul style="list-style-type: none"> • SCI 受信設定を行います。 • 第四引数で設定した受信回数分、R_SCI_Send 関数を使用してダミー送信を行うことで、第三引数で用意したバッファに受信データを格納します。 • 受信するタイミングはダミー送信するタイミングとなります。本関数では受信処理は行いません。 • 受信処理（ダミー送信）が全て完了すると、Rxl 割り込み発生後に第五引数のコールバック関数が実行されます。 |
| 引数 | <ul style="list-style-type: none"> • 第一引数：1 ;受信用SCIチャンネルの選択 • 第二引数： ;制御オプション PDL_SCI_DMACH_DTC_TRIGGER データバイト送信時、DMACHとDTCの転送無効_DISABLE, • 第三引数：pData ;受信データ格納用バッファ • 第四引数：RxCnt ;総受信回数 • 第五引数：SCI1RxFunc ;コールバック関数（総受信後）：SCI1RxFunc() • 第六引数：PDL_NO_FUNC ;コールバック関数（エラー処理用）：なし |
| リターン値 | <ul style="list-style-type: none"> • 全パラメータが正しく指定されている場合：True (1) • 正しく指定されていない場合：False (0) |
| 備考 | <ul style="list-style-type: none"> • RX610 用のRPDLでは、SCIの設定用に本関数を含めて6つの関数を用意しています。それぞれの関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Serial Communication Interface」で確認してください。 • 本関数の引数の詳細については、同マニュアル内の「Serial Communication Interface」の「R_SCI_Receive」を参照してください。 |

| R_SCI_Send | |
|------------|---|
| 概要 | 受信処理を行うためのダミー送信処理 |
| ヘッダ | r_pdl_sci.h, r_pdl_definitions.h |
| 宣言 | bool R_SCI_Send(uint8_t data1, uint16_t data2, uint8_t * data3, uint16_t data4, void * func); |
| 説明 | <ul style="list-style-type: none"> ● 本関数は受信処理時のダミー送信用として使用している関数です。 ● 送受信回数の値を第四引数に設定し、受信回数分ダミー送信することで受信処理を行います。 ● 受信を行うタイミングは、ダミー送信を行うタイミングとなります。 |
| 引数 | <ul style="list-style-type: none"> ● 第一引数：1 ;転送用SCIのチャンネル ● 第二引数：PDL_NO_DATA ;制御オプション：デフォルト設定 ● 第三引数：Tr_Dummy ;ダミー送信用データ格納バッファ ● 第四引数：RxCnt ;総受信回数 ● 第五引数：PDL_NO_FUNC ;コールバック関数：なし |
| リターン値 | <ul style="list-style-type: none"> ● 全パラメータが正しくかつ排他的に指定されている場合：True (1) ● 正しく指定されていない場合：False (0) |
| 備考 | <ul style="list-style-type: none"> ● ダミー送信の内容は"0xFF"です。"0xFF"はEEPROM制御用コマンド以外の値となっていますので、"0xFF"以外は設定しないでください。 ● 受信回数分、0xFFをダミー送信用バッファ（Tr_Dummy）に格納し、第三引数にバッファ名を代入する必要があります。 ● RX610用のRPDLでは、SCIの設定用に本関数を含めて6つの関数を用意しています。それぞれの関数については「RX610 Group Peripheral Driver Library User's Manual」内の「API List by Peripheral Function」の「Serial Communication Interface」で確認してください。 ● 本関数の引数の詳細については、同マニュアル内の「Serial Communication Interface」の「R_SCI_Send」を参照してください。 |

5.9.3 Serial EEPROM制御ミドルウェアが提供しているテストプログラムの仕様

以下に Serial EEPROM 制御ミドルウェアが提供しているテストプログラムの仕様を示します。

| testmain | |
|----------|---|
| 概要 | メインプログラム |
| ヘッダ | |
| 宣言 | void testmain(void) |
| 説明 | <ul style="list-style-type: none"> ● 本アプリケーションノートのメインプログラムです。 ● test100 関数、test200 関数、test300 関数を実行します。 |
| 引数 | ● なし |
| リターン値 | ● なし |
| 備考 | <ul style="list-style-type: none"> ● 本関数をメインプログラムにしているため、resetprg.c内を変更しています。詳細に関しては、「7.1 resetprg.cの内容変更について」を参照してください。 |

| test100 | |
|---------|--|
| 概要 | 初期化関数テスト |
| ヘッダ | mtl_com.h, R_SPI_EEP.h, R_SPI_EEP_io.h |
| 宣言 | void Test100(void) |
| 説明 | <ul style="list-style-type: none"> R_SPI_EEP_Init_Driver 関数のテストを行います。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | <ul style="list-style-type: none"> R_SPI_EEP_Init_Driver 関数の詳細に関しては、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |

| test200 | |
|---------|--|
| 概要 | リードステータス関数テスト、ライトプロテクト関数テスト |
| ヘッダ | mtl_com.h, R_SPI_EEP.h, R_SPI_EEP_io.h |
| 宣言 | void Test200(void) |
| 説明 | <ul style="list-style-type: none"> R_SPI_EEP_Write_Protect 関数のテストを行います。 R_SPI_EEP_Read_Status 関数のテストを行います。 |
| 引数 | <ul style="list-style-type: none"> なし |
| リターン値 | <ul style="list-style-type: none"> なし |
| 備考 | <ul style="list-style-type: none"> R_SPI_EEP_Write_Protect 関数および R_SPI_EEP_Read_Status 関数の詳細に関しては、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |

| test300 | |
|---------|---|
| 概要 | データ読み出し関数テスト、データ書き込み関数テスト |
| ヘッダ | mtl_com.h, R_SPI_EEP.h, R_SPI_EEP_io.h |
| 宣言 | void Test300(void) |
| 説明 | <ul style="list-style-type: none"> R_SPI_EEP_Write_Data 関数のテストを行います。 R_SPI_EEP_Read_Data 関数のテストを行います。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | <ul style="list-style-type: none"> R_SPI_EEP_Write_Data 関数および R_SPI_EEP_Read_Data 関数の詳細に関しては、「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」のアプリケーションノートを参照してください。 |

| trap | |
|-------|--|
| 概要 | 無限ループ処理 |
| ヘッダ | |
| 宣言 | void trap(void) |
| 説明 | <ul style="list-style-type: none"> 無限ループを行います。 |
| 引数 | なし |
| リターン値 | なし |
| 備考 | <ul style="list-style-type: none"> Test100、Test200、Test300 の関数内における Serial EEPROM 制御ミドルウェアの関数の返り値がエラーの場合、本関数に分岐します。 |

5.10 フローチャート

5.10.1 ドライバ初期化処理

図 5.7 にドライバ初期化処理のフローチャートを示します。

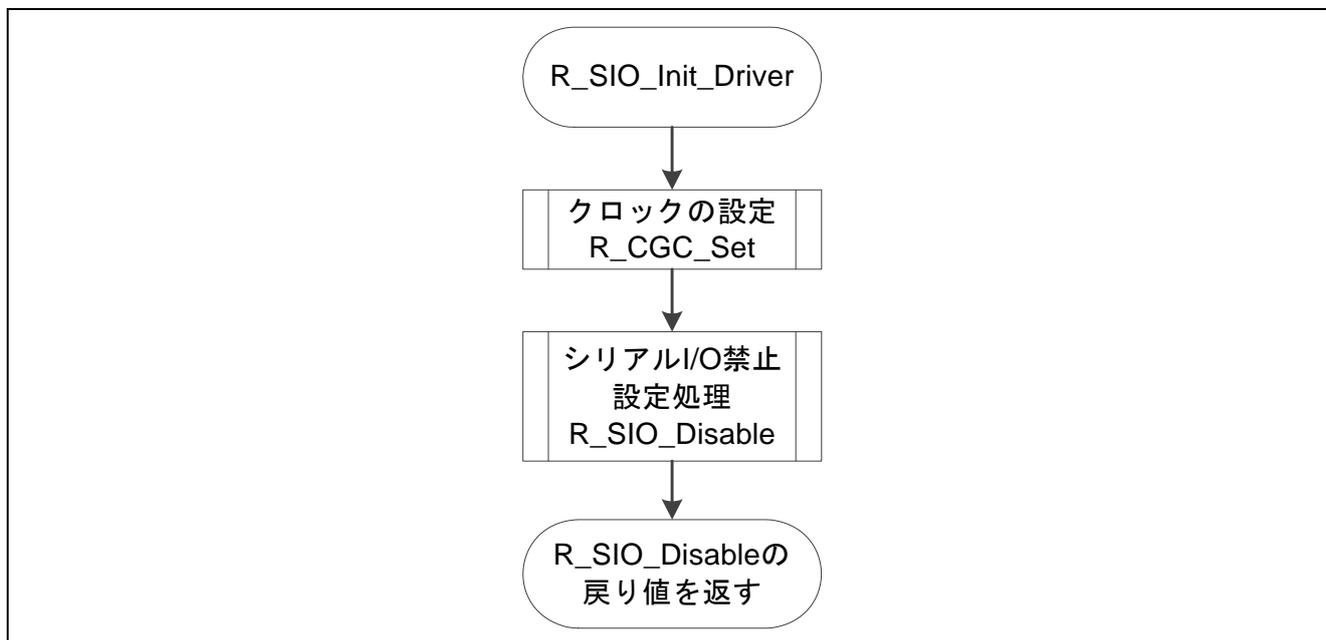


図5.7 ドライバ初期化処理

5.10.2 シリアルI/O禁止設定処理

図 5.8 にシリアルI/O禁止設定処理のフローチャートを示します。

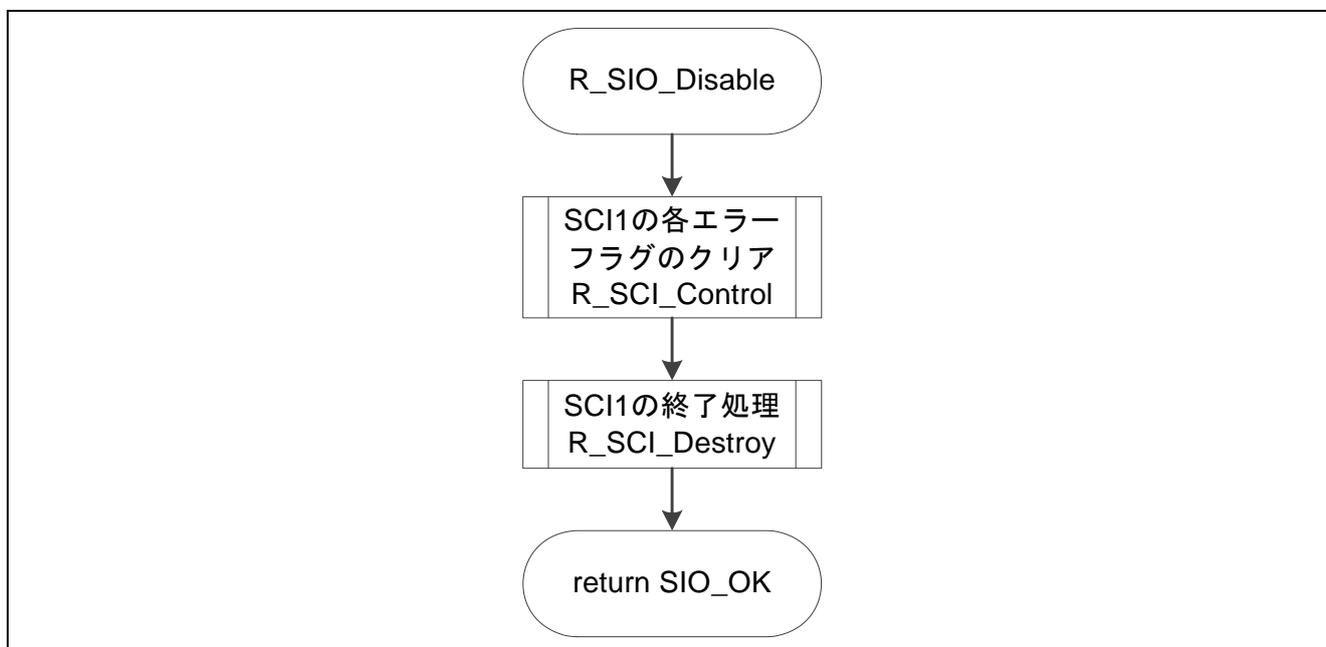


図5.8 シリアル I/O 禁止設定処理

5.10.3 シリアルI/O許可設定処理

図 5.9にシリアルI/O許可設定処理のフローチャートを示します。

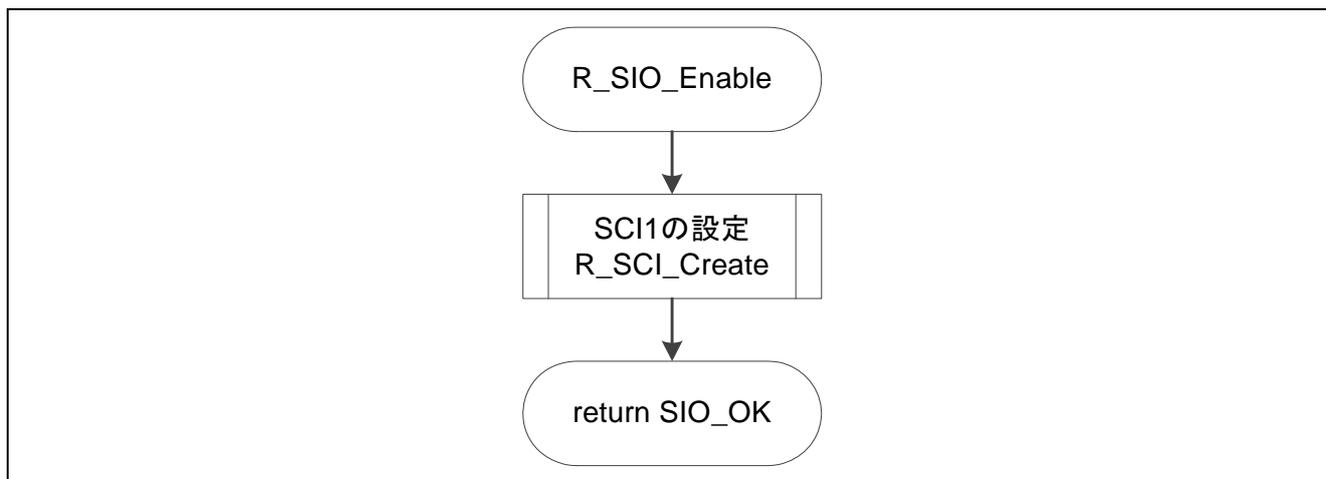


図5.9 シリアル I/O 許可設定処理

5.10.4 シリアルI/Oデータ送信処理

図 5.10にシリアルI/Oデータ送信処理のフローチャートを示します。

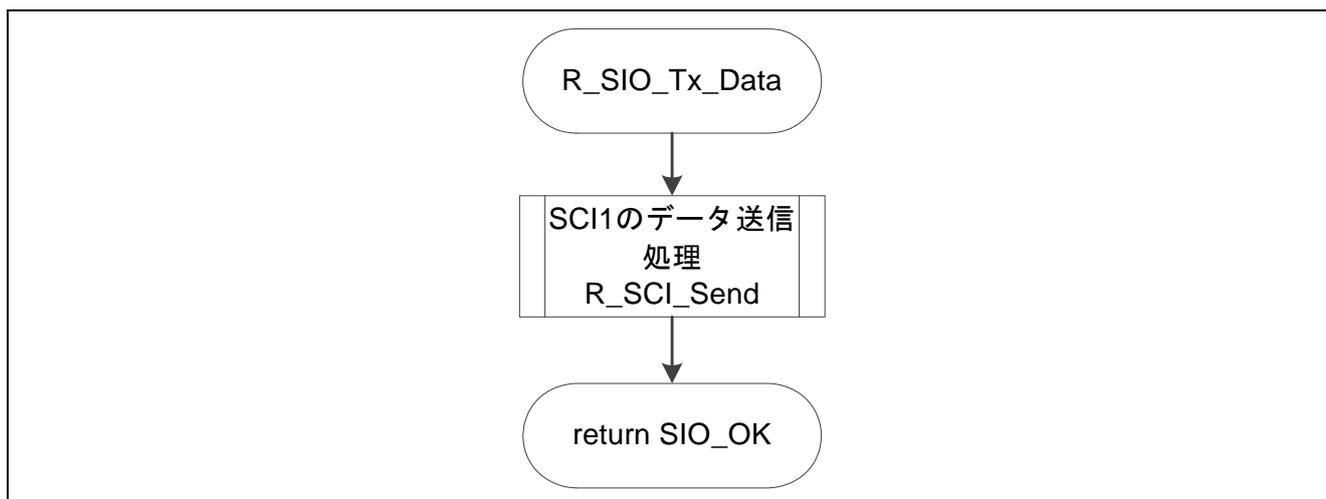


図5.10 シリアル I/O データ送信処理

5.10.5 シリアルI/Oデータ受信処理

図 5.11にシリアルI/Oデータ受診処理のフローチャートを示します。

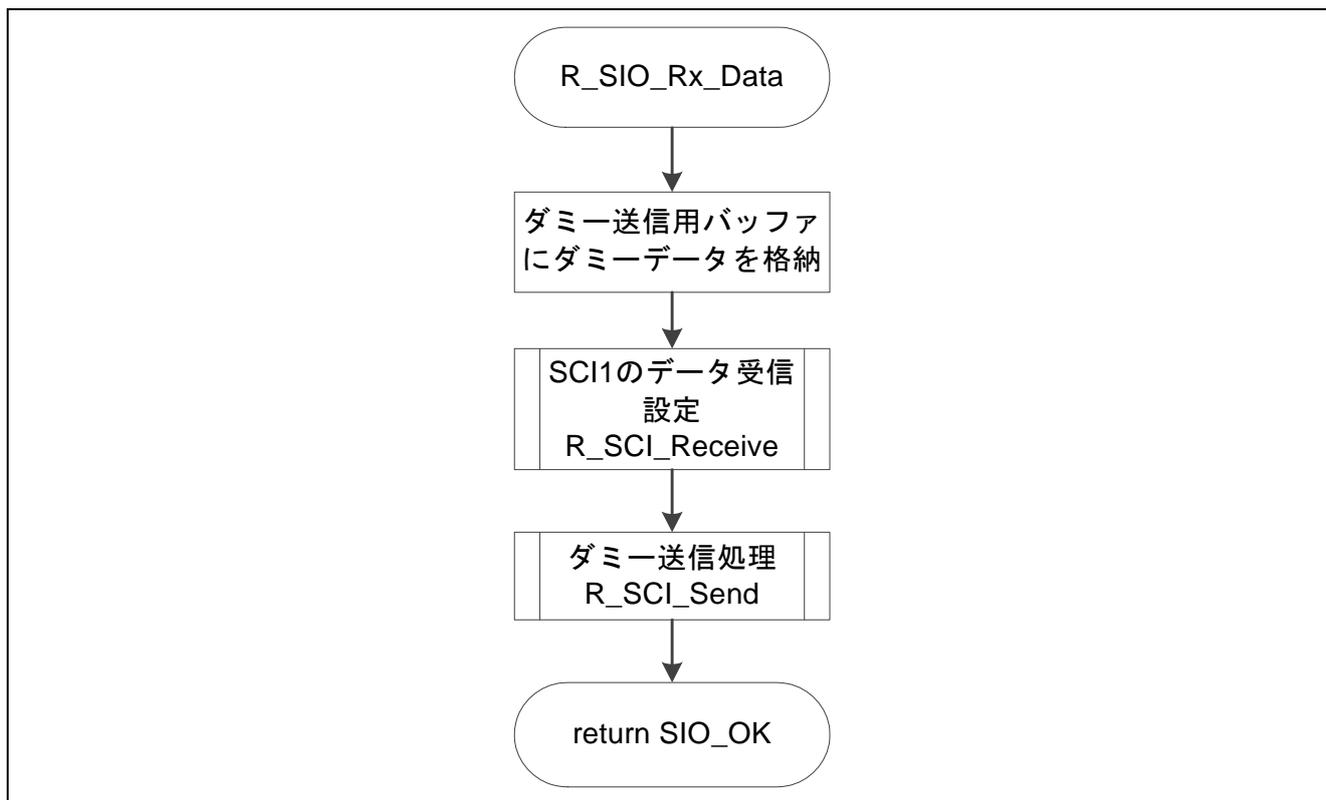


図5.11 シリアル I/O データ受信処理

5.10.6 受信処理用コールバック関数

図 5.12に受診処理用コールバック関数のフローチャートを示します。

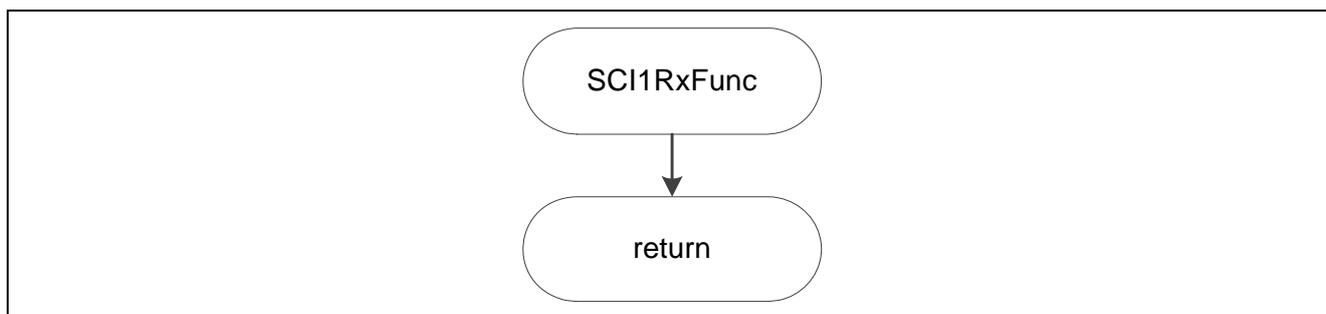


図5.12 受信処理用コールバック関数

6. サンプルコード

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

7. 参考ドキュメント

- RX610 グループ ユーザーズマニュアルハードウェア編 Rev.1.11
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- テクニカルアップデート/テクニカルニュース
(最新の情報をルネサス エレクトロニクスホームページから入手してください。)
- C コンパイラマニュアル
RX ファミリ用 C/C++コンパイラパッケージ V.1.01 Release00
RX ファミリ C/C++コンパイラパッケージ V1.01 ユーザーズマニュアル Rev.1.00
(最新版をルネサス エレクトロニクスホームページから入手してください。)
- アプリケーションノート「Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア」 Rev.1.03 (R01AN0565JJ)
(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。)
- ライブラリ「Renesas Peripheral Driver Library」(R20UT0083EE0100)
(最新版をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
<http://japan.renesas.com/>
- お問い合わせ先
<http://japan.renesas.com/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 | 2012.03.27 | — | 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、本ドキュメントおよびテクニカルアップデートを参照してください。

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（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）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（予約領域）がありません。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、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品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情報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が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。
総お問合せ窓口：<http://japan.renesas.com/contact/>