

RX ファミリ、RL78 ファミリ、78K0R/Kx3-L R01AN0565JJ0106 Rev.1.06

Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア 2016.03.31

要旨

本アプリケーションノートでは、ルネサス エレクトロニクス製 MCU を使用したルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズの SPI Serial EEPROM 制御方法とサンプルコードの使用方法を説明します。

なお、本サンプルコードは、スレーブデバイスとしての Serial EEPROM を制御するための上位層に位置するソフトウェアです。

別途、マスタデバイスとしての各 MCU 個別の SPI モードを制御するための下位層に位置するソフトウェア（クロック同期式シングルマスタ制御ソフトウェア）を用意していますので、以下の URL から入手してください。なお、クロック同期式シングルマスタ制御ソフトウェアにおいて、新規 MCU が対応可能になった場合でも、本アプリケーションノートの更新が間に合わないことがあります。最新のサポート MCU とその制御ソフトウェアの組み合わせ情報は、以下の URL のページに記載されている「クロック同期式シングルマスタドライバ（下位層ソフトウェア）」を参照してください。

- SPI シリアル EEPROM 制御
— http://japan.renesas.com/driver/spi_serial_eeprom

動作確認デバイス

Serial EEPROM ルネサス エレクトロニクス R1EX25xxx シリーズ SPI Serial EEPROM

動作確認に使用した MCU

RX600 シリーズ	: RX610, RX62N, RX63N, RX63T (SCI を使用) : RX62N, RX63N, RX63T (RSPI を使用)
RX200 シリーズ	: RX210, RX21A, RX220 (SCI を使用) : RX210, RX21A, RX220 (RSPI を使用)
RX100 シリーズ	: RX111 (SCI を使用) : RX111 (RSPI を使用)
78K0R/Kx	: 78K0R/KE3-L (SAU を使用)
RL78/G1x	: RL78/G14, RL78/G1C (SAU を使用)
RL78/L1x	: RL78/L12, RL78/L13C, RL78/L1C (SAU を使用)

本アプリケーションノートを他のマイコンや他メモリへ適用する場合、そのマイコンやメモリの仕様に合わせて変更し、十分評価してください。

目次

1. 仕様	4
2. 動作確認条件	5
2.1 RX ファミリ	5
2.2 RL78 ファミリ、78K0R/Kx3-L	13
3. 関連アプリケーションノート	20
4. ハードウェア説明	21
4.1 使用端子一覧	21
5. ソフトウェア説明	22
5.1 動作概要	22
5.1.1 クロック同期式モードで発生させるタイミング	23
5.1.2 Serial EEPROM の S#端子制御	23
5.1.3 Serial EEPROM 命令コード	24
5.1.4 Serial EEPROM ステータスレジスタ	24
5.2 ソフトウェア制御概要	25
5.2.1 ソフトウェア構成	25
5.2.2 チップセレクト端子初期化 (R_SPI_EEP_Init_Port())	26
5.2.3 チップセレクト端子制御 (EEP_SET_CS())	26
5.2.4 ソフトウェア・ウェイト (mtl_wait_lp())	26
5.2.5 シリアル許可 (R_SIO_Enable())	26
5.2.6 コマンド送信 (R_SPI_EEP_Send_Cmd())	27
5.2.7 データ送信 (R_SIO_Tx_Data ())	27
5.2.8 データ受信 (R_SIO_Rx_Data ())	27
5.2.9 シリアル禁止 (R_SIO_Disable ())	27
5.3 必要メモリサイズ	28
5.3.1 RX ファミリ	28
5.3.2 RL78 ファミリ、78K0R/Kx3-L	36
5.4 ファイル構成	40
5.5 定数一覧	41
5.5.1 戻り値	41
5.5.2 コマンド定義	41
5.5.3 各種定義	42
5.6 構造体／共用体一覧	43
5.7 変数一覧	43
5.8 関数一覧	44
5.9 関数の詳細	45
5.9.1 ドライバ初期化処理	45
5.9.2 ステータス読み出し処理	46
5.9.3 ライトプロテクト設定処理	48
5.9.4 データ読み出し処理	49
5.9.5 データ書き込み処理	50
5.9.6 ポート初期化処理 (内部関数)	51

5.9.7	RAM 初期化処理 (内部関数)	52
5.9.8	コマンド送信処理 (内部関数)	53
5.9.9	読み出しコマンド処理 (内部関数)	54
5.9.10	書き込み許可コマンド処理	55
5.9.11	書き込み禁止コマンド処理	56
5.9.12	ステータスレジスタ読み出しコマンド処理 (内部関数)	57
5.9.13	ステータスレジスタ書き込みコマンド処理 (内部関数)	59
5.9.14	ビジーウェイト処理 (内部関数)	61
5.9.15	ページ書き込みコマンド処理 (内部関数)	62
5.9.16	コマンド設定処理 (内部関数)	64
6.	応用例	65
6.1	Serial EEPROM 制御ソフトウェアの設定	66
6.1.1	R_SPI_EEP.h	66
6.1.2	R_SPI_EEP_sfr.h	67
6.1.3	R_SPI_EEP_io.c	68
7.	使用上の注意事項	69
7.1	組み込み時の注意事項	69
7.2	キャッシュ搭載の MCU を使用する場合	69
7.3	他スレーブデバイスを使用する場合	69

1. 仕様

ルネサス エレクトロニクス製 MCU を使用し、ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズの SPI Serial EEPROM を制御します。

別途、MCU 個別のクロック同期式シングルマスタ制御ソフトウェアが必要です。

表 1-1 に使用する周辺機器と用途を、図 1-1 に使用例を示します。

以下に、機能概略を示します。

- マスタデバイスをルネサス エレクトロニクス製 MCU、スレーブデバイスをルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズの SPI Serial EEPROM としたブロック型デバイスドライバです。
- MCU 内蔵のシリアル通信機能（クロック同期式モード）を使用し、SPI モードで制御します。また、ユーザ設定したシリアル I/O 1 チャンネルの使用が可能です。複数チャンネルの使用は、できません。
- 同一型名、かつ最大 2 個の SPI Serial EEPROM の制御が可能です。
- 通信速度レートは、ユーザ設定可能です。（固定値）
- ビッグエンディアン／リトルエンディアンでの動作が可能です。（使用 MCU 依存）

表 1-1 使用する周辺機器と用途

周辺機器	用途
MCU 内蔵のシリアル通信機能 （クロック同期式モード）	シリアル通信機能（クロック同期式モード）による SPI スレーブデバイスとの通信 1ch（必須）
Port	SPI スレーブデバイスセレクト制御信号用 使用デバイス数分のポートが必要（必須）

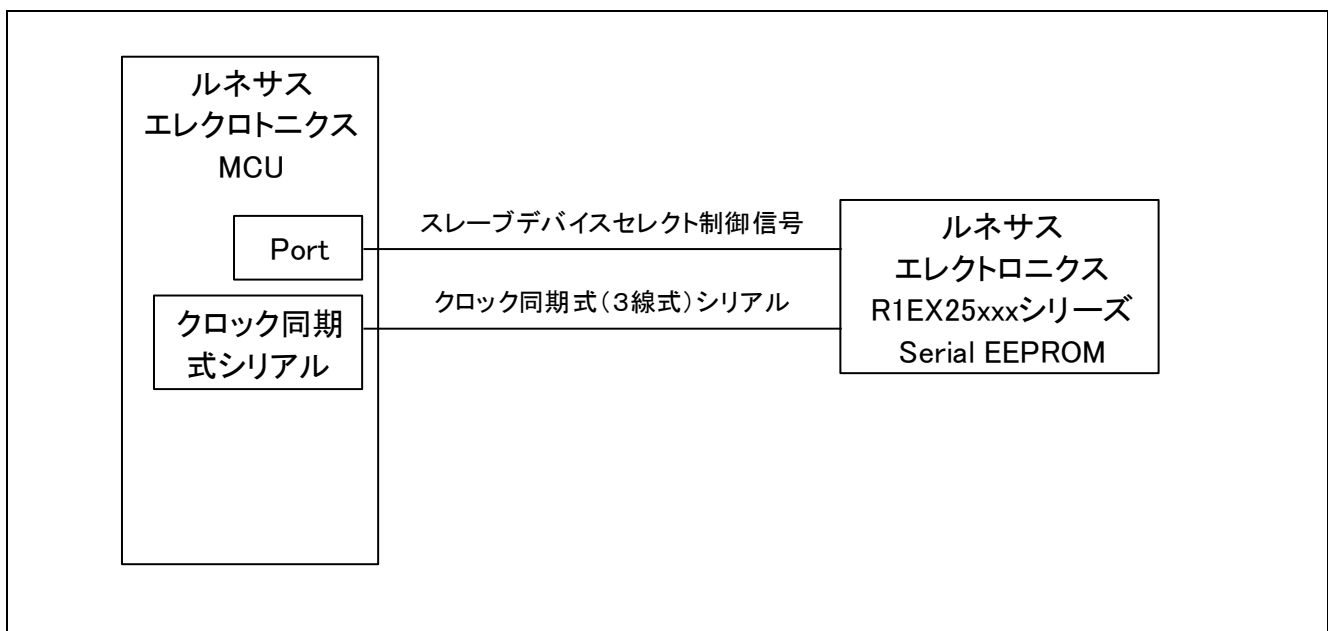


図 1-1 使用例

2. 動作確認条件

本アプリケーションノートのサンプルコードは、以下の動作条件で動作を確認しています。

2.1 RX ファミリ

(1) RX610 SCI の場合

表 2-1 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX610 グループ (プログラム ROM 2MB/RAM 128KB)
動作周波数 (マイコン)	ICLK : 100MHz、PCLK : 50MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 High-performance embedded Workshop Version 4.07.00.007
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.0.0.0) コンパイルオプション 統合開発環境のデフォルト設定を使用しています。
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.00
評価に使用したソフトウェア	RX610 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN0534JJ) Ver.2.00
評価に使用したボード	Renesas Starter Kit for RX610

(2) RX62N RSPI の場合

表 2-2 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX62N グループ (プログラム ROM 512KB RAM 96KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.0.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.01
評価に使用したソフトウェア	RX62N グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN0323JJ) Ver.2.04
評価に使用したボード	Renesas Starter Kit for RX62N

(3) RX62N SCI の場合

表 2-3 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX62N グループ (プログラム ROM 512KB RAM 96KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.0.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX62N グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1088JJ) Ver.2.01
評価に使用したボード	Renesas Starter Kit for RX62N

(4) RX63N RSPI の場合

表 2-4 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX63N グループ (プログラム ROM 1MB RAM 128KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04.R01
評価に使用したボード	Renesas Starter Kit for RX63N

(5) RX63N SCI の場合

表 2-5 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX63N グループ (プログラム ROM 1MB RAM 128KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01
評価に使用したボード	Renesas Starter Kit for RX63N

(6) RX63T RSPI の場合

表 2-6 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX63T グループ (プログラム ROM 512KB RAM 48KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.00.00
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 2.00.00) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04.R01
評価に使用したボード	Renesas Starter Kit for RX63T

(7) RX63T SCI の場合

表 2-7 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX63T グループ (プログラム ROM 512KB RAM 48KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.00.00
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 2.00.00) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01.R01
評価に使用したボード	Renesas Starter Kit for RX63T

(8) RX210 RSPI の場合

表 2-8 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX210 グループ (プログラム ROM 512KB RAM 64KB)
動作周波数	ICLK : 50MHz、PCLK : 25MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04.R01
評価に使用したボード	Renesas Starter Kit for RX210

(9) RX210 SCI の場合

表 2-9 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX210 グループ (プログラム ROM 512KB RAM 64KB)
動作周波数	ICLK : 50MHz、PCLK : 25MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01.R01
評価に使用したボード	Renesas Starter Kit for RX210

(10) RX21A RSPI の場合

表 2-10 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX21A グループ (プログラム ROM 512KB RAM 64KB)
動作周波数	ICLK : 50MHz、PCLK : 25MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04.R01
評価に使用したボード	株式会社北斗電子製 HSBRX21AP-B

(11) RX21A SCI の場合

表 2-11 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX21A グループ (プログラム ROM 512KB RAM 64KB)
動作周波数	ICLK : 50MHz、PCLK : 25MHz
動作電圧	3.3V
統合開発環境	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 ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01.R01
評価に使用したボード	株式会社北斗電子製 HSBRX21AP-B

(12) RX220 RSPI の場合

表 2-12 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX220 グループ (プログラム ROM 256KB RAM 16KB)
動作周波数	ICLK : 32MHz、PCLK : 32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 High-performance embedded Workshop Version 4.09.01.007
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04.R01
評価に使用したボード	Renesas Starter Kit for RX220

(13) RX220 SCI の場合

表 2-13 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX220 グループ (プログラム ROM 256KB RAM 16KB)
動作周波数	ICLK : 32MHz、PCLK : 32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 High-performance embedded Workshop Version 4.09.01.007
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.02
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01.R01
評価に使用したボード	Renesas Starter Kit for RX220

(14) RX111 RSPI の場合

表 2-14 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX111 グループ (プログラム ROM 128KB RAM 16KB)
動作周波数	ICLK : 32MHz、PCLK : 32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 2.01.00) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04 R04
評価に使用したボード	Renesas Starter Kit for RX111

(15) RX111 SCI の場合

表 2-15 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RX111 グループ (プログラム ROM 128KB RAM 16KB)
動作周波数	ICLK : 32MHz、PCLK : 32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
Cコンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 2.01.00) コンパイルオプション 統合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01 R05
評価に使用したボード	Renesas Starter Kit for RX111

2.2 RL78 ファミリ、78K0R/Kx3-L

(1) 78K0R/Kx3-L SAU の場合

表 2-16 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	78K0R/KE3-L (プログラム ROM 64KB/RAM 3KB)
動作周波数	メイン・システム・クロック : 20MHz CPU/周辺ハードウェア・クロック : 20MHz シリアル・クロック : 2.5MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 プロジェクト・マネージャ(PM+ Ver.6.31)
Cコンパイラ、アセンブラ	ルネサス エレクトロニクス製 78K0R 用 C コンパイラ(CC78K0R Ver.2.13) 78K0R 用アセンブラパッケージ(RA78K0R Ver.1.33) コンパイルオプション 統合開発環境のデフォルト設定 ("-a. -zp") を使用しています。
統合デバッガ	ルネサス エレクトロニクス製 Integrated Debugger ID78K0R-QB Ver.3.61
サンプルコードのバージョン	Ver.2.01
評価に使用したソフトウェア	78K0R/Kx3-L シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN0708JJ) Ver.2.00
評価に使用したボード	78K0R/KE3-L ターゲット・ボード(QB-78K0RKE3L-TB)

(2) RL78/G14 SAU 統合開発環境 CS+ for CA,CX の場合 (コンパイラ : CA78K0R)

表 2-17 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/G14 (プログラム ROM 256KB/RAM 24KB)
動作周波数	メイン・システム・クロック : 32MHz CPU/周辺ハードウェア・クロック : 32MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CS+ for CA,CX V3.01.00
C コンパイラ、アセンブラ	ルネサス エレクトロニクス製 RL78,78K0R コンパイラ CA78K0R V1.71 コンパイルオプション 統合開発環境のデフォルト設定 ("-qx2") を使用しています。
サンプルコードのバージョン	Ver.2.04
評価に使用したソフトウェア	RL78/G14、RL78/G1C、RL78/L12、RL78/L13、RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ) Ver.2.05
評価に使用したボード	Renesas Starter Kit for RL78/G14

(3) RL78/G14 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 2-18 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/G14 (プログラム ROM 256KB/RAM 24KB)
動作周波数	メイン・システム・クロック : 32MHz CPU/周辺ハードウェア・クロック : 32MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CS+ for CC V3.03.00
C コンパイラ、アセンブラ	ルネサス エレクトロニクス製 RL78 コンパイラ CC-RL V1.02.00 コンパイルオプション 統合開発環境のデフォルト設定 (既定の最適化を行う(なし)) を使用しています。
サンプルコードのバージョン	Ver.2.04
評価に使用したソフトウェア	RL78/G14、RL78/G1C、RL78/L12、RL78/L13、RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ) Ver.2.05
評価に使用したボード	Renesas Starter Kit for RL78/G14

(4) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench の場合

表 2-19 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/G14 (プログラム ROM 256KB/RAM 24KB)
動作周波数	メイン・システム・クロック : 32MHz CPU/周辺ハードウェア・クロック : 32MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 (Ver.1.30.2)
Cコンパイラ、アセンブラ	IAR Systems 製 IAR Assembler for Renesas RL78 (Ver.1.30.2.50666) IAR C/C++ Compiler for Renesas RL78 (Ver.1.30.2.50666) コンパイルオプション 統合開発環境のデフォルト設定 ("レベル 低") を使用しています。
サンプルコードのバージョン	Ver.2.03
評価に使用したソフトウェア	RL78/G14 シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/G14

(5) RL78/G1C SAU 統合開発環境 CubeSuite+の場合

表 2-20 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/G1C (プログラム ROM 32KB/RAM 5.5KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
Cコンパイラ、アセンブラ	ルネサス エレクトロニクス製 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas RL78/G1C ターゲット・ボード QB-R5F10JGC-TB

(6) RL78/G1C SAU 統合開発環境 IAR Embedded Workbench の場合

表 2-21 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/G1C (プログラム ROM 32KB / RAM 5.5KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C コンパイラ、アセンブラ	IAR Systems 製 IAR Assembler for Renesas RL78 (Ver.1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver.1.30.5.50715) コンパイルオプション 統合開発環境のデフォルト設定 ("レベル 低") を使用しています。
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas RL78/G1C ターゲット・ボード QB-R5F10JGC-TB

(7) RL78/L12 SAU 統合開発環境 CubeSuite+の場合

表 2-22 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/L12 (プログラム ROM 32KB / RAM 1.5KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
C コンパイラ、アセンブラ	ルネサス エレクトロニクス製 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/L12

(8) RL78/L12 SAU 統合開発環境 IAR Embedded Workbench の場合

表 2-23 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/L12 (プログラム ROM 32KB / RAM 1.5KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C コンパイラ、アセンブラ	IAR Systems 製 IAR Assembler for Renesas RL78 (Ver.1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver.1.30.5.50715) コンパイルオプション 統合開発環境のデフォルト設定 ("レベル 低") を使用しています。
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/L12

(9) RL78/L13 SAU 統合開発環境 CubeSuite+の場合

表 2-24 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/L13 (プログラム ROM 128KB / RAM 8KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
C コンパイラ、アセンブラ	ルネサス エレクトロニクス製 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/L13

(10) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合

表 2-25 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/L13 (プログラム ROM 128KB / RAM 8KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
Cコンパイラ、アセンブラ	IAR Systems 製 IAR Assembler for Renesas RL78 (Ver.1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver.1.30.5.50715)) コンパイルオプション 統合開発環境のデフォルト設定 ("レベル 低") を使用しています。
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/L13

(11) RL78/L1C SAU 統合開発環境 CubeSuite+の場合

表 2-26 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/L1C (プログラム ROM 256KB / RAM 16KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
Cコンパイラ、アセンブラ	ルネサス エレクトロニクス製 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/L1C

(12) RL78/L1C SAU 統合開発環境 IAR Embedded Workbench の場合

表 2-27 動作確認条件

項目	内容
評価に使用したメモリ	ルネサス エレクトロニクス製 R1EX25xxx/HN58X25xxx シリーズ SPI Serial EEPROM
評価に使用したマイコン	RL78/L1C (プログラム ROM 256KB / RAM 16KB)
動作周波数	メイン・システム・クロック : 24MHz 周辺ハードウェア・クロック : 24MHz シリアル・クロック : 4MHz
動作電圧	3.3V
統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C コンパイラ、アセンブラ	IAR Systems 製 IAR Assembler for Renesas RL78 (Ver.1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver.1.30.5.50715)) コンパイルオプション 統合開発環境のデフォルト設定 ("レベル 低") を使用しています。
サンプルコードのバージョン	Ver.2.03 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用したボード	Renesas Starter Kit for RL78/L1C

3. 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。併せて参照してください。

- RX610 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN0534JJ)
- RX62N グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN0323JJ)
- 78K0R/Kx3-L シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN0708JJ)
- RX62N グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1088JJ)
- RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ)
- RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ)
- RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ)

4. ハードウェア説明

4.1 使用端子一覧

表 4-1 に、MCU 側の使用端子と機能を示します。

表 4-1 使用する端子名

端子名	入出力	内容
CLK (注 1)	出力	クロック出力
DataOut (注 1)	出力	マスターデータ出力
DataIn (注 1)	入力	マスターデータ入力
Port(CS#) (注 1)	出力	スレーブデバイスセレクト出力

注 1. 本アプリケーションノートでは、サンプルコードの記述に合わせて、CLK 端子、DataIn 端子、DataOut 端子、Port(CS#)端子として、表します。

4.2 参考回路

図 4-1 に、接続例を示します。

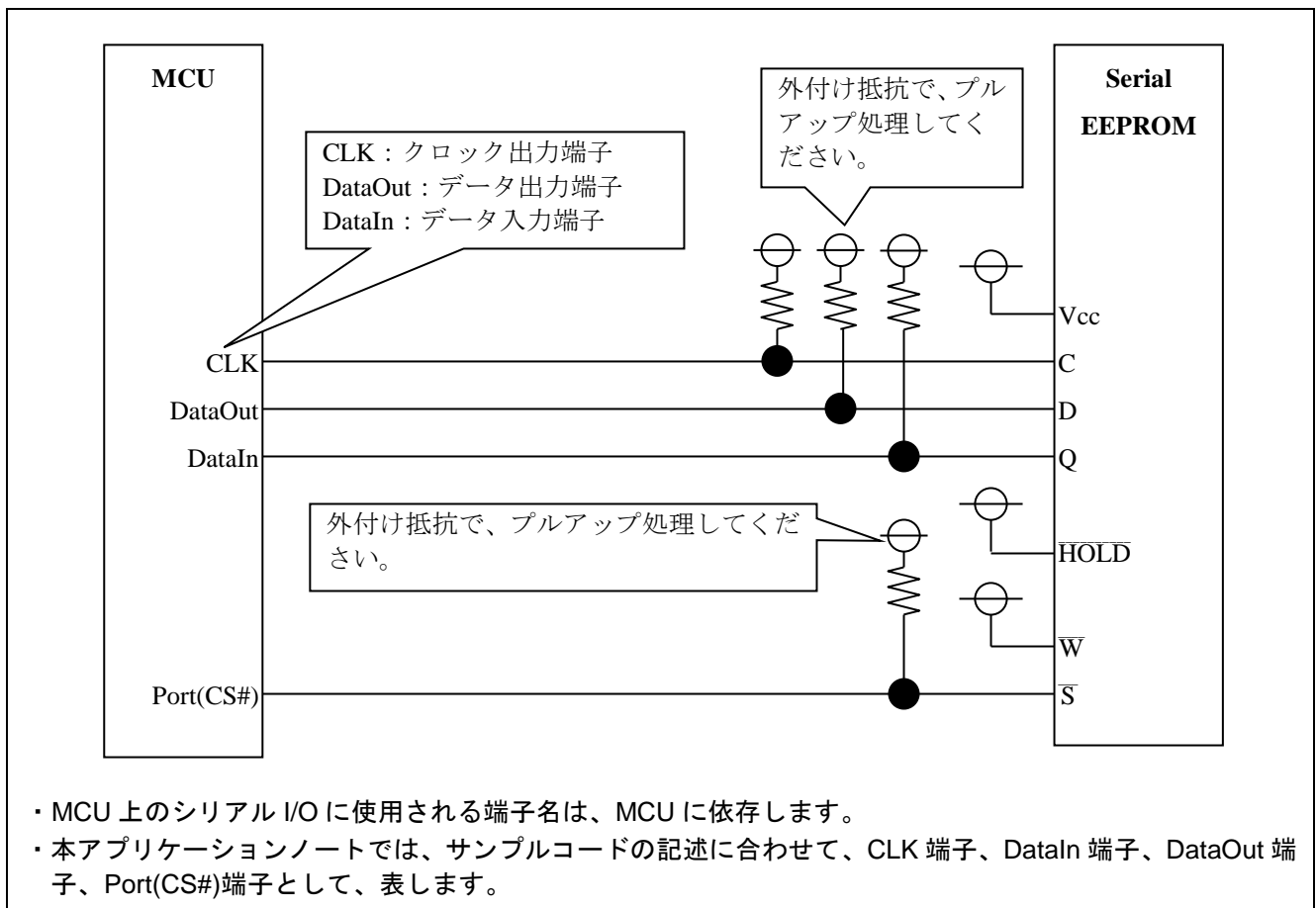


図 4-1 MCU と SPI スレーブデバイスの接続例

5. ソフトウェア説明

5.1 動作概要

MCU のクロック同期式（3 線式）シリアル通信機能を使って、Serial EEPROM 制御を実現します。

本サンプルコードでは、以下の制御を行っています。

- SPI スレーブデバイスの S#端子を MCU の Port に接続し、MCU 汎用ポート出力で、制御する。
（本サンプルコードで制御する。）
- データの入出力を、クロック同期式モード（内部クロック使用）で、制御する。
（本サンプルコードが、MCU 個別のクロック同期式シングルマスタ制御ソフトウェアを使用する。）

本サンプルコードは、以下のように、デバイス上のデータのバイトオフセット値と、転送元/先のメモリのバイトオフセット値が合致するようにしたものです。

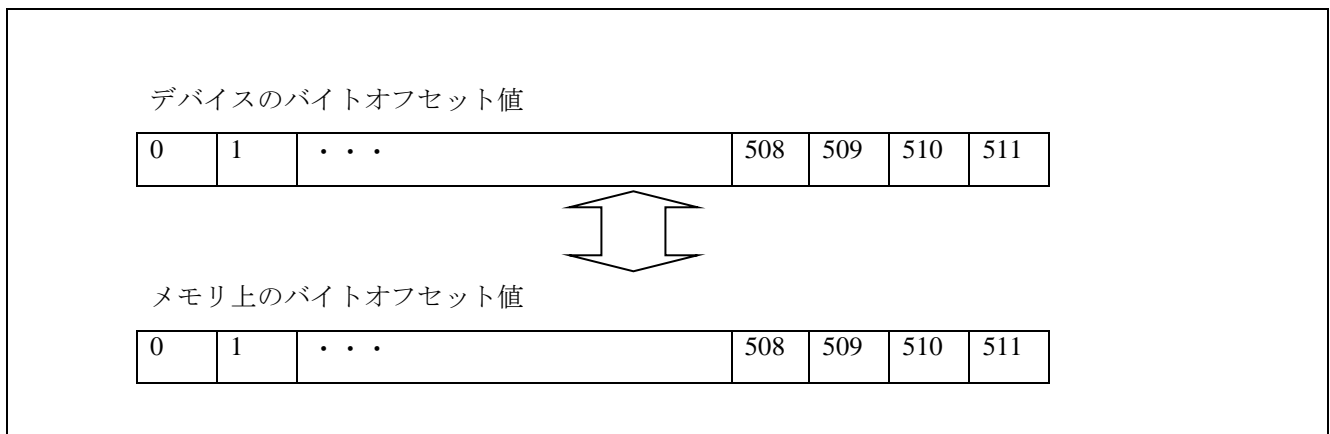


図 5-1 転送データの格納

5.1.1 クロック同期式モードで発生させるタイミング

Serial EEPROM 制御のため、図 5-2 に示す SPI モード 3 (CPOL=1、CPHA=1) のタイミングを発生させます。

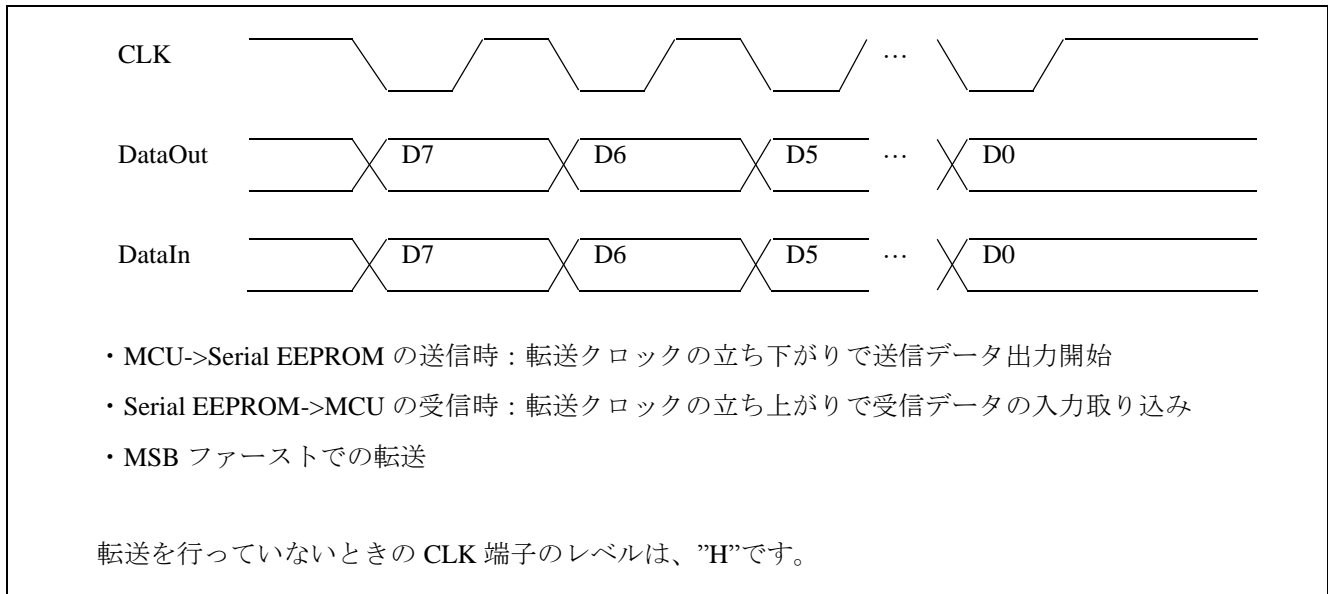


図 5-2 クロック同期式モード タイミング設定

使用可能なシリアル・クロック周波数は、MCU と SPI デバイスの各データシートを参照し、設定してください。

5.1.2 Serial EEPROM の S#端子制御

Serial EEPROM の S#端子を MCU の Port に接続し、MCU 汎用ポート出力で、制御します。

Serial EEPROM の S# (MCU の Port(CS#)) 信号の立ち下がりから、Serial EEPROM の C (MCU の CLK) 信号の立ち下がりまでの時間は、Serial EEPROM の S#セットアップ時間待ちのために、ソフトウェア・ウェイトで制御しています。

Serial EEPROM の C (MCU の CLK) 信号の立ち上がりから、Serial EEPROM の S# (MCU の Port(CS#)) 信号の立ち上がりまでの時間は、Serial EEPROM の S#ホールド時間待ちのために、ソフトウェア・ウェイトで制御しています。

Serial EEPROM のデータシートを確認して、システムに応じたソフトウェア・ウェイト時間を設定してください。

5.1.3 Serial EEPROM 命令コード

Serial EEPROM 制御のための以下の命令(Instruction)コードがあり、このコードを使って、コマンド制御を行います。

表 5-1 Instruction Set

Instruction	Description	Instruction format
WREN	Write Enable	0000 0110
WRDI	Write Disable	0000 0100
RDSR	Read Status Register	0000 0101
WRSR	Write Status Register	0000 0001
READ	Read from Memory Array	0000 0011
WRITE	Write to Memory Array	0000 0010

5.1.4 Serial EEPROM ステータスレジスタ

Serial EEPROM のステータスレジスタは、専用の命令により、読み出し/書き込みが可能です。また、ステータスレジスタの構成は、使用するメモリ容量に依存します。

各 bit の詳細は、Serial EEPROM データシートを参照してください。

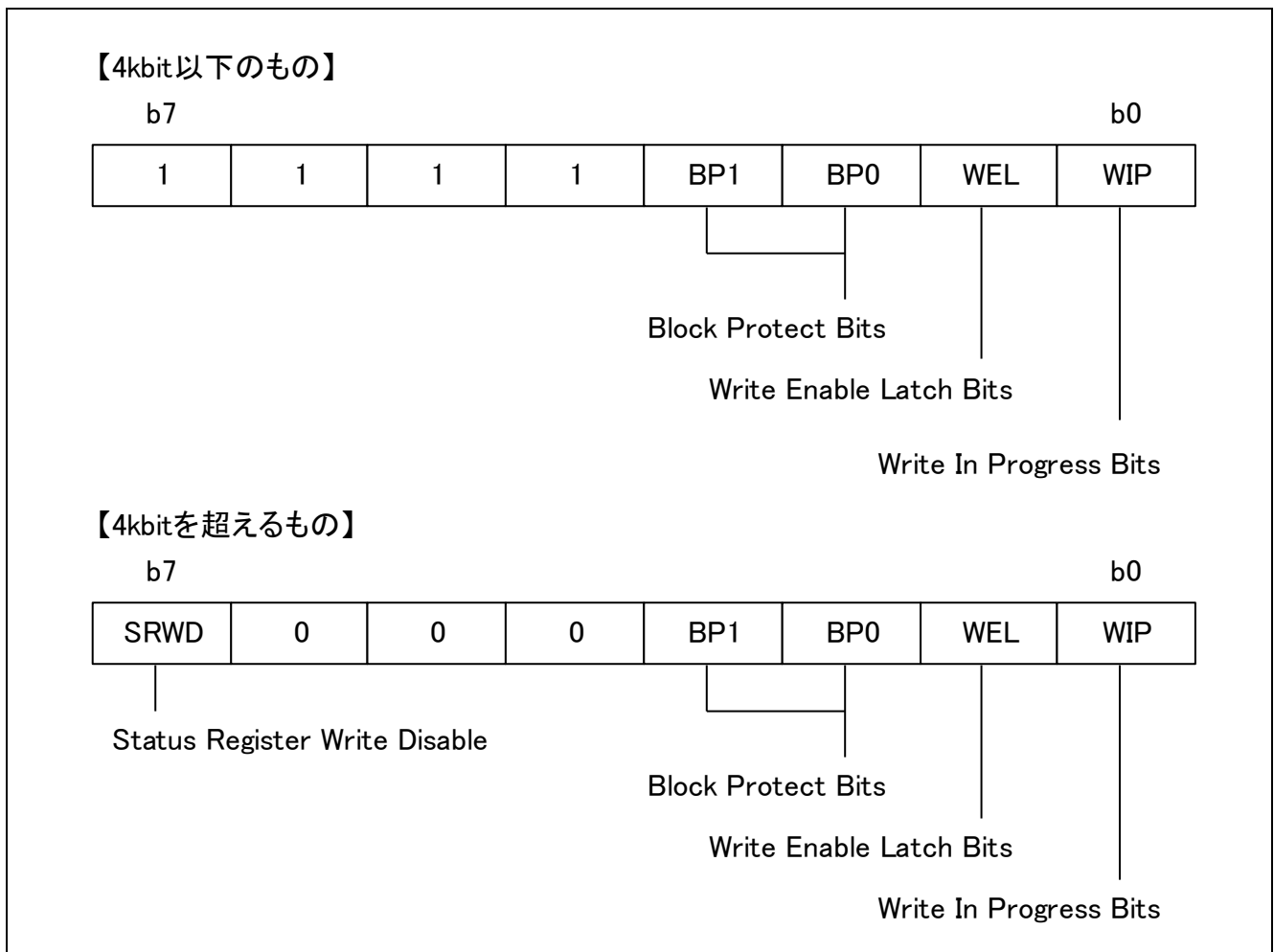


図 5-3 ステータスレジスタの構成図

5.2 ソフトウェア制御概要

5.2.1 ソフトウェア構成

本サンプルコードは、SPI Serial EEPROM 制御のための上位層に位置する制御ソフト（図 5-4 の EEPROM 制御ソフトウェア）です。

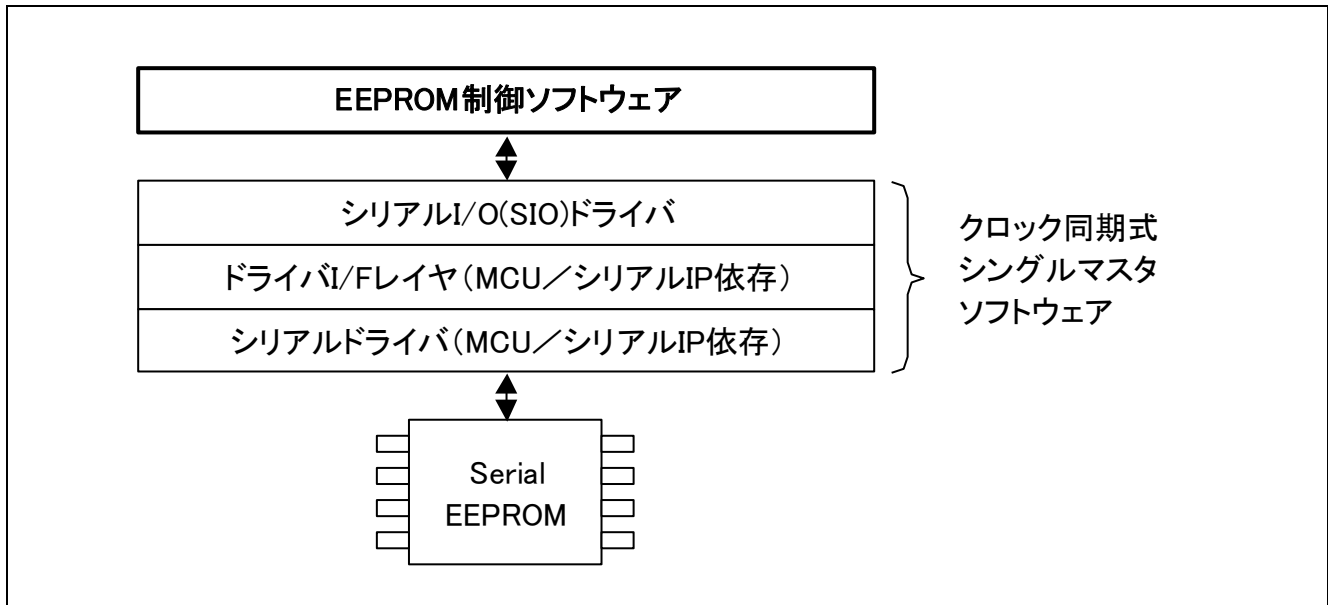


図 5-4 ソフトウェア構成

以下に制御手順を示します。

- ①Port(CS#)信号の立ち下げ
- ②ソフトウェア・ウェイト
- ③クロック同期式シングルマスタソフトウェアを使ったコマンド/データ送受信
- ④ソフトウェア・ウェイト
- ⑤Port(CS#)信号の立ち上げ

本サンプルコードは、以下の 8 つの基本処理で構成しています。

- チップセレクト端子初期化 —Port(CS#)端子の”H”出力設定
- チップセレクト端子制御 —Port(CS#)端子の”H”/”L”出力設定
- ソフトウェア・ウェイト —ソフトウェア・ウェイトによるタイミング調整
- シリアル許可 —DataIn 端子のポート入力化、DataOut 端子と CLK 端子のポート”H”出力
シリアル I/O 有効化、ビットレート設定
- コマンド送信 —Serial EEPROM へのコマンド送信処理
- データ送信 —Serial EEPROM へのデータ送信処理
- データ受信 —Serial EEPROM からのレスポンス/データ受信処理
- シリアル禁止 —シリアル I/O 無効化、DataIn 端子のポート入力化、
DataOut 端子と CLK 端子のポート”H”出力化

5.2.2 チップセレクト端子初期化 (R_SPI_EEP_Init_Port())

スレーブデバイスのチップセレクト信号を”H”出力（動作前に、非アクティブ状態に設定）します。

MCU 毎に制御方法が異なりますが、制御手順は、以下のとおりです。

- ①ポート出力値を”H”出力に設定する。（②のポートを”出力”に切り替えたときに、”H”出力させるため）
- ②ポートを”出力”に設定する。
- ③ポート出力値を”H”出力に設定する。

5.2.3 チップセレクト端子制御 (EEP_SET_CS())

スレーブデバイスのチップセレクト信号を”H”/”L”出力制御します。

5.2.4 ソフトウェア・ウェイト (mtl_wait_lp())

ソフトウェア・ループ処理で、ウェイト制御します。

5.2.5 シリアル許可 (R_SIO_Enable())

シリアルを以下の手順で許可設定します。

- ①シリアル I/O で使用する DataIn 端子をポート入力、DataOut 端子と CLK 端子をポート”H”出力にする。
- ②シリアル I/O 機能を有効にし、DataIn 端子をデータ入力、DataOut 端子をデータ出力、CLK 端子をクロック出力に切り替える。
- ③シリアル I/O で使用する通信速度（ビットレート）を設定する。

R_SIO_Enable()の詳細は、個別のクロック同期式シングルマスタ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。

5.2.6 コマンド送信 (R_SPI_EEP_Send_Cmd())

Serial EEPROM に Instruction コード (コマンド) を送信します。

コマンドに対するレスポンス受信は、R_SIO_Rx_Data ()を使用します。

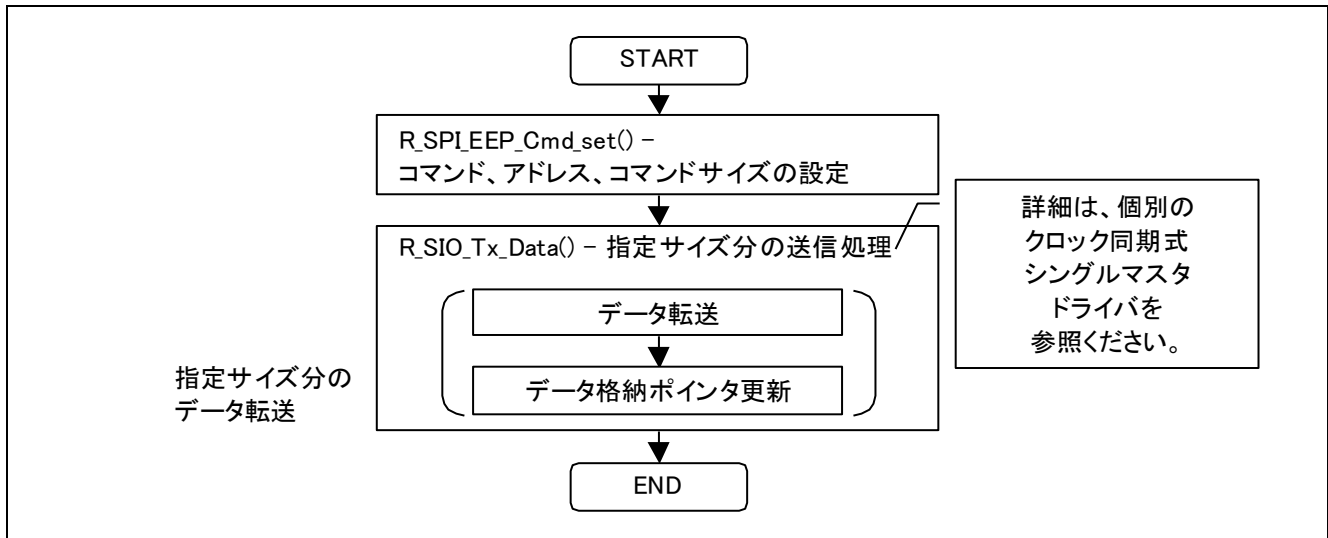


図 5-5 コマンド送信処理概要 (R_SPI_EEP_Send_Cmd())

5.2.7 データ送信 (R_SIO_Tx_Data ())

シリアル I/O を使って、データを送信します。Instruction コード、アドレス情報、書き込みデータ、またはステータスレジスタの値を送信します。

詳細は、個別のクロック同期式シングルマスタ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。

5.2.8 データ受信 (R_SIO_Rx_Data ())

シリアル I/O を使って、データを受信します。読み出しデータまたはステータスレジスタを受信します。

詳細は、個別のクロック同期式シングルマスタ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。

5.2.9 シリアル禁止 (R_SIO_Disable ())

シリアル I/O で使用する端子をポートに切り替えて、DataIn 端子をポート入力、DataOut 端子と CLK 端子をポート”H”出力にします。

詳細は、個別のクロック同期式シングルマスタ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。

5.3 必要メモリサイズ

以下に、必要とするメモリサイズを示します。

5.3.1 RX ファミリ

(1) RX610 SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-2 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	838 バイト (リトルエンディアン) 705 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	152 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(2) RX62N RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-3 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 897 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	168 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(3) RX62N SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-4 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 898 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	148 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(4) RX63N RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-5 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 897 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	168 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(5) RX63N SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-6 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 898 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	148 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(6) RX63T RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-7 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 897 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	168 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(7) RX63T SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-8 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 898 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	148 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(8) RX210 RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-9 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 897 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	168 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(9) RX210 SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-10 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 898 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	148 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(10) RX21A RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-11 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 897 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	168 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(11) RX21A SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-12 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 898 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	148 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(12) RX220 RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-13 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 897 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	168 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(13) RX220 SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-14 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	705 バイト (リトルエンディアン) 898 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト (リトルエンディアン) 4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	148 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(14) RX111 RSPI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-15 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1,613 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	184 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(15) RX111 SCI の場合

環境は、「2.動作確認条件」を参照してください。

表 5-16 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1,613 バイト (リトルエンディアン)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	4+n バイト (リトルエンディアン) (n は、使用デバイス個数を示す。)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	180 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。エンディアンにより、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

5.3.2 RL78 ファミリ、78K0R/Kx3-L

命令の異なる MCU 毎にメモリサイズを示します。使用 MCU の命令を調査し参考にしてください。

(1) 78K0R/Kx3-L SAU の場合

環境は、「2.動作確認条件」を参照してください。

表 5-17 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1211 バイト 1264 バイト	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト 6 バイト (使用デバイス個数 1 または 2 の場合)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	130 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(2) RL78/G14 SAU 統合開発環境 CS+ for CA,CX の場合 (コンパイラ : CA78K0R)

環境は、「2.動作確認条件」を参照してください。

表 5-18 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1254 バイト 1232 バイト	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト 6 バイト (使用デバイス個数 1 または 2 の場合)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	130 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(3) RL78/G14 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

環境は、「2.動作確認条件」を参照してください。

表 5-19 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	879 バイト 864 バイト	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト 6 バイト (使用デバイス個数 1 または 2 の場合)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	98 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(4) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench の場合

環境は、「2.動作確認条件」を参照してください。

表 5-20 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	1353 バイト 1091 バイト	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 バイト 6 バイト (使用デバイス個数 1 または 2 の場合)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	132 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、プロジェクト全体のスタックサイズです。下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(5) RL78/L13 SAU 統合開発環境 CubeSuite+ の場合

環境は、「2.動作確認条件」を参照してください。

表 5-21 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	2,206 バイト	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	6 バイト (使用デバイス個数 1 または 2 の場合)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	120 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

(6) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合

環境は、「2.動作確認条件」を参照してください。

表 5-22 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	2,160 バイト	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	6 バイト (使用デバイス個数 1 または 2 の場合)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
最大使用ユーザスタック	84 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 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。クロック同期式シングルマスタソフトウェアに使用される上記のメモリサイズを含みません。使用する MCU により、上記のメモリサイズは、異なります。最大使用ユーザスタックサイズは、プロジェクト全体のスタックサイズです。下位層のクロック同期式シングルマスタ制御ソフトウェアのスタックも含まれます。

5.4 ファイル構成

表 5-23 に、サンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成するファイルは除きます。

表 5-23 ファイル構成

¥an_r01an0565jj0106_mcu_serial	<DIR>	サンプルコードのフォルダ
r01an0565jj0106_mcu.pdf		アプリケーションノート
¥ source	<DIR>	プログラム格納用フォルダ
¥r_spi_eep	<DIR>	Serial EEPROM 制御ソフトウェア用フォルダ
R_SPI_EEP.h		ヘッダファイル
R_SPI_EEP_io.c		I/O モジュール
R_SPI_EEP_io.h		I/O モジュール用共通定義
R_SPI_EEP_sfr.h.78k0r		レジスタ用共通定義 (78K0R/Kx3-L 用)
R_SPI_EEP_sfr.h.rl78g14		レジスタ用共通定義 (RL78/G14 用)
R_SPI_EEP_sfr.h.rl78g1c		レジスタ用共通定義 (RL78/G1C 用)
R_SPI_EEP_sfr.h.rl78l1c		レジスタ用共通定義 (RL78/L1C 用)
R_SPI_EEP_sfr.h.rl78l12		レジスタ用共通定義 (RL78/L12 用)
R_SPI_EEP_sfr.h.rl78l13		レジスタ用共通定義 (RL78/L13 用)
R_SPI_EEP_sfr.h.rx21a		レジスタ用共通定義 (RX21A 用)
R_SPI_EEP_sfr.h.rx62n		レジスタ用共通定義 (RX62N 用)
R_SPI_EEP_sfr.h.rx63n		レジスタ用共通定義 (RX63N 用)
R_SPI_EEP_sfr.h.rx63t		レジスタ用共通定義 (RX63T 用)
R_SPI_EEP_sfr.h.rx111		レジスタ用共通定義 (RX111 用)
R_SPI_EEP_sfr.h.rx210		レジスタ用共通定義 (RX210 用)
R_SPI_EEP_sfr.h.rx220		レジスタ用共通定義 (RX220 用)
R_SPI_EEP_sfr.h.rx610		レジスタ用共通定義 (RX610 用)
R_SPI_EEP_usr.c		ユーザ I/F モジュール
¥sample	<DIR>	動作確認プログラム格納用フォルダ
testmain.c		動作検証用のサンプルプログラム

【注】 別途、MCU 個別のクロック同期式シングルマスタ制御ソフトウェアが必要です。

5.5 定数一覧

5.5.1 戻り値

表 5-24 に、サンプルコードで使用する戻り値を示します。

表 5-24 戻り値

定数名	設定値	内容
EEP_OK	(error_t)(0)	Successful Operation
EEP_ERR_PARAM	(error_t)(-1)	Parameter Error
EEP_ERR_HARD	(error_t)(-2)	Hardware Error
EEP_ERR_WP	(error_t)(-4)	Write-Protection Error
EEP_ERR_OTHER	(error_t)(-7)	Other Error

5.5.2 コマンド定義

表 5-25 に、サンプルコードで使用するコマンド定義を示します。

表 5-25 コマンド定義

定数名	設定値	内容
EEP_CMD_WREN	(uint8_t)0x06	Write Enable
EEP_CMD_WRDI	(uint8_t)0x04	Write Disable
EEP_CMD_RDSR	(uint8_t)0x05	Read Status Register
EEP_CMD_WRSR	(uint8_t)0x01	Write Status Register
EEP_CMD_READ	(uint8_t)0x03	Read for Memory Array
EEP_CMD_WRITE	(uint8_t)0x02	Write for Memory Array

5.5.3 各種定義

表 5-26 に、サンプルコードで使用する各種定義した値を示します。

表 5-26 各種定義値

定数名	設定値	内容
EEP_LOG_ERR	1	Log type : Error
EEP_TRUE	(uint8_t)0x01	Flag "ON"
EEP_FALSE	(uint8_t)0x00	Flag "OFF"
EEP_WP_NONE	(uint8_t)0x00	Write-Protection Status : None setting
EEP_WP_UPPER_QUART	(uint8_t)0x01	Write-Protection Status : Upper quarter setting
EEP_WP_UPPER_HALF	(uint8_t)0x02	Write-Protection Status : Upper half setting
EEP_WP_WHOLE_MEM	(uint8_t)0x03	Write-Protection Status : Whole memory setting
EEP_MEM_SIZE	(uint32_t)0x100	メモリサイズ (バイト単位) 左記は、2 Kbit の場合の値です。
EEP_WPAG_SIZE	(uint32_t)16	ページサイズ (バイト単位) 左記は、2 Kbit の場合の値です。
EEP_ADDR_SIZE	(uint8_t)1	アドレスサイズ (バイト単位) 左記は、2 Kbit の場合の値です。
EEP_UPPER_QUART	(EEP_MEM_SIZE – (EEP_MEM_SIZE / 4))	上位 1/4 設定時の開始アドレス
EEP_UPPER_HALF	(EEP_MEM_SIZE / 2)	上位 1/2 設定時の開始アドレス
EEP_CMD_SIZE	(uint8_t)1	コマンドサイズ (バイト単位)
EEP_STSREG_SIZE	(uint16_t)1	ステータスレジスタサイズ (バイト単位)
EEP_SHORT_SIZE	(uint32_t)0x0000fff0	下位関数での最大転送サイズの設定 (0xFFFF 以下を設定)
EEP_HI	(uint8_t)0x01	Port "H"
EEP_LOW	(uint8_t)0x00	Port "L"
EEP_OUT	(uint8_t)0x01	Port Output Setting
EEP_IN	(uint8_t)0x00	Port Input Setting
EEP_READY_WAIT	(uint16_t)8000	Write Busy Waiting Time 8000 × 10us = 80ms
EEP_T_READY_WAIT	(uint16_t)MTL_T_10US	Write Busy Completion Polling Time
EEP_T_CS_HOLD	(uint16_t)MTL_T_1US	CS Stability Waiting Time
EEP_T_R_ACCESS	(uint16_t)MTL_T_1US	Reading Start Waiting Time
EEP_REG_SRWD	(uint8_t)0x80	Status Register Write Disable
EEP_REG_BP1	(uint8_t)0x08	Block Protect Bit1
EEP_REG_BP0	(uint8_t)0x04	Block Protect Bit0
EEP_REG_WEL	(uint8_t)0x02	Write Enable Latch Bit
EEP_REG_WIP	(uint8_t)0x01	Write In Progress Bit
EEP_BYTE_READ	2	1 バイト読み出し実行判断バイト数
EEP_BYTE_WRITE	2	1 バイト書き込み実行判断バイト数

5.6 構造体／共用体一覧

以下に、サンプルコードで使用する構造体を示します。

```

/* uint16_t <-> uint8_t conversion */
typedef union {
    uint32_t ul;
    uint8_t uc[4];
} EEP_EXCHG_LONG; /* total 4byte */

typedef struct {
    uint8_t EEP_Cmd; /* Command */
    uint8_t EEP_Addr[3]; /* Operand */
} EEP_CMD; /* total 4byte */

```

5.7 変数一覧

表 5-27 に、サンプルコードで使用する変数一覧を示します。

表 5-27 変数一覧

型	変数名	内容	使用関数
STATIC EEP_CMD	gEep_CmdBuf	コマンド用バッファ	R_SPI_EEP_Send_Cmd R_SPI_EEP_Cmd_set
uint8_t	gEep_WP[EEP_DEV_NUM]	コマンド用バッファ	R_SPI_EEP_Init_Ram R_SPI_EEP_Write_Protect R_SPI_EEP_Write_Data

5.8 関数一覧

表 5-28 に、関数を示します。

表 5-28 関数一覧

関数名	説明
R_SPI_EEP_Init_Driver	ドライバ初期化処理
R_SPI_EEP_Read_Status	ステータス読み出し処理
R_SPI_EEP_Write_Protect	ライトプロテクト設定処理
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_Port	ポート初期化処理
R_SPI_EEP_Init_Ram	RAM 初期化処理
R_SPI_EEP_Send_Cmd	コマンド送信処理
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
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_StsReg	ステータスレジスタ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_Write_Page	ページ書き込みコマンド処理
R_SPI_EEP_Read_Memory	読み出しコマンド処理
R_SPI_EEP_Cmd_set	コマンド設定処理

キャッシュ搭載の MCU を使用する場合、読み出し／書き込み用データ格納バッファは、非キャッシュ領域を指定してください。

読み出し／書き込み用データ格納バッファアドレスは、下位層の MCU 個別のクロック同期式シングルマスタ制御ソフトウェアに依存し、4 バイト境界アドレスを指定する必要がある場合があります。MCU 個別のクロック同期式シングルマスタ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。

5.9 関数の詳細

5.9.1 ドライバ初期化処理

R_SPI_EEP_Init_Driver	
概要	ドライバ初期化処理
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
宣言	error_t R_SPI_EEP_Init_Driver(void)
説明	<ul style="list-style-type: none"> ・ドライバを初期化します。 以下の処理をデバイス毎に繰り返します。 <ul style="list-style-type: none"> –R_SPI_EEP_Init_Ram()関数をコールし、RAM の初期化を行います。 –R_SPI_EEP_Init_Port()関数をコールし、CS#端子の初期化を行います。 シリアル I/O ドライバの R_SIO_Init_Driver()関数をコールし、I/O ポートの初期化を行います。
引数	void
リターン値	<ul style="list-style-type: none"> ・初期化結果を返します。 <pre>EEP_OK ; Successful operation EEP_ERR_OTHER ; Other error</pre> <p>R_SIO_Init_Driver()の戻り値を返します。</p>
備考	なし

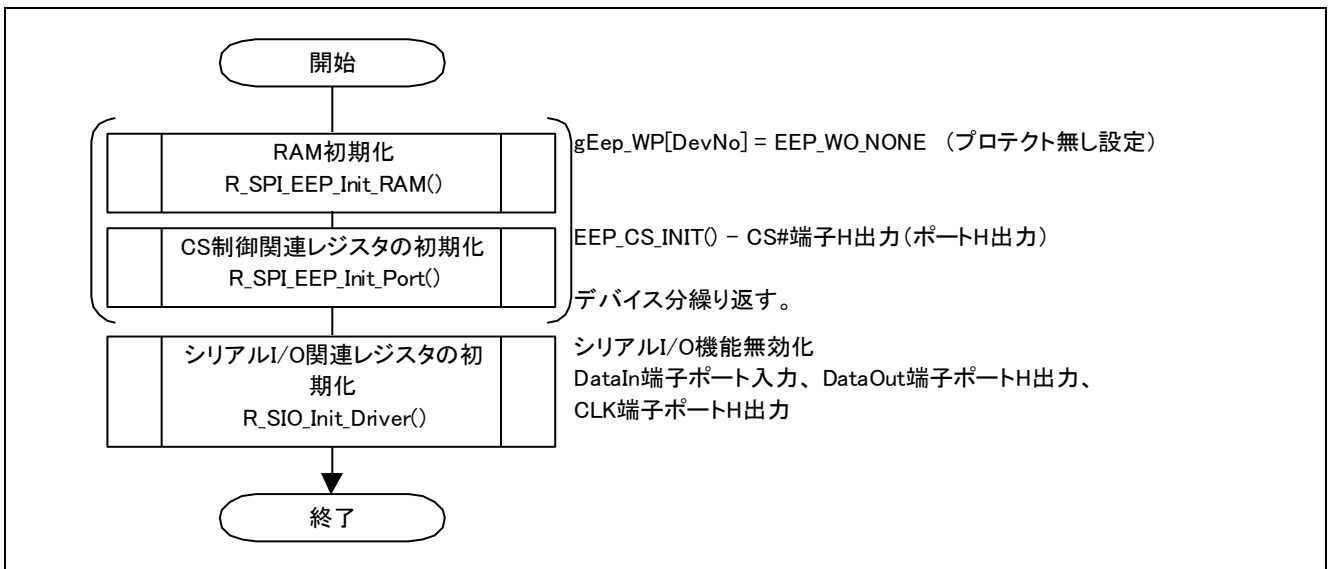


図 5-6 ドライバ初期化処理概要

5.9.2 ステータス読み出し処理

R_SPI_EEP_Read_Status

概要	ステータスレジスタ読み出し処理	
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h	
宣言	error_t R_SPI_EEP_Read_Status(uint8_t DevNo, uint8_t FAR* pStatus)	
説明	<ul style="list-style-type: none"> ・ステータスレジスタを読み出し、pStatus に格納します。 読み出しバッファとして、1バイトを設定してください。 ・読み出しステータス格納バッファ(pStatus)には下記情報が格納されます。 <ul style="list-style-type: none"> ●4 Kbit 以下の場合 <ul style="list-style-type: none"> Bit 7 to 4:Reserved (All "1") Bit 3 to 2:BP1, BP0 00: No protection 01: Upper-quarter protection 10: Upper-half protection 11: Whole memory protection Bit 1:WEL 0: Write disabled 1: Write enabled Bit 0:WIP 1: During write operation ●4 Kbit を超える場合 <ul style="list-style-type: none"> Bit 7:SRWD 0: Status register can be changed. 1: Status register cannot be changed. Bit 6 to 4:Reserved (All "0") Bit3 to 2:BP1, BP0 00: No protection 01: Upper-quarter protection 10: Upper-half protection 11: Whole memory protection Bit 1:WEL 0: Write disabled 1: Write enabled Bit 0:WIP 1: During write operation 	
引数	uint8_t DevNo	; デバイス番号
	uint8_t FAR* pStatus	; 読み出しステータス格納バッファ
リターン値	<ul style="list-style-type: none"> ・ステータスレジスタ取得結果を返します。 EEP_OK ; Successful operation EEP_ERR_PARAM ; Parameter error EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error 	
備考	なし	

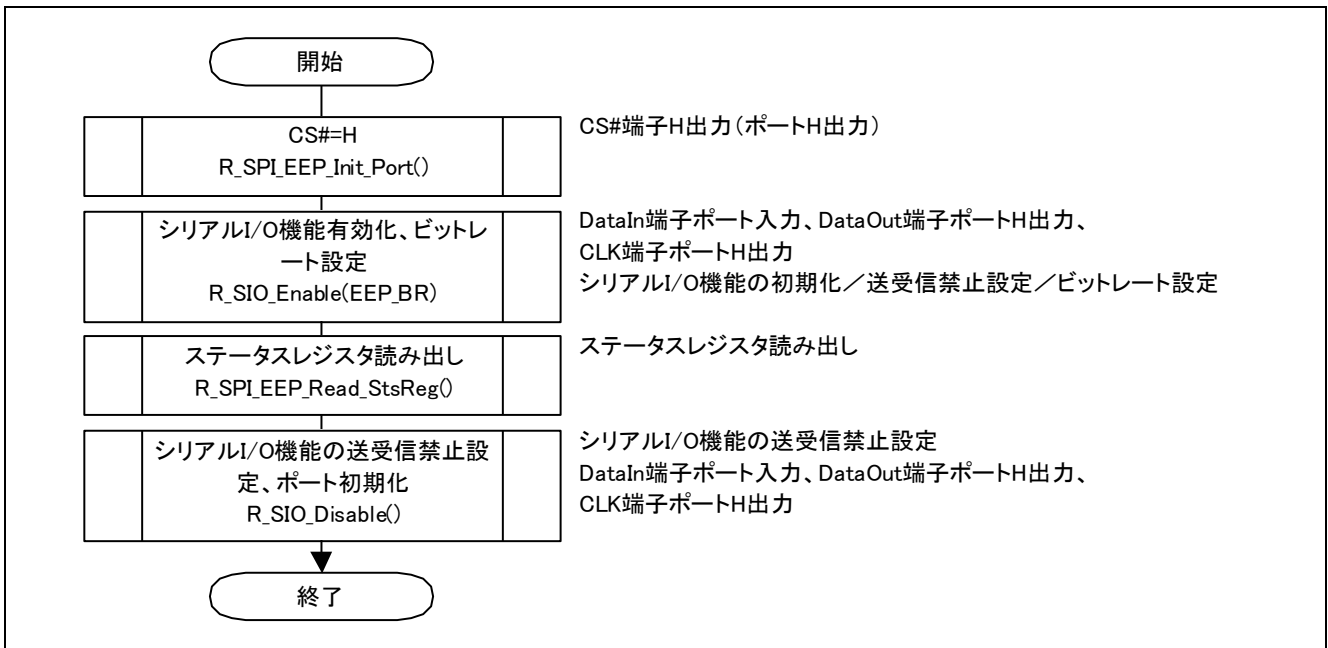


図 5-7 ステータスレジスタ読み出し処理概要

5.9.3 ライトプロテクト設定処理

R_SPI_EEP_Write_Protect

概要	ライトプロテクト設定処理														
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h														
宣言	error_t R_SPI_EEP_Write_Protect(uint8_t DevNo, uint8_t WpSts)														
説明	<ul style="list-style-type: none"> ・ライトプロテクトの設定を行います。 ・ライトプロテクト設定データ(WpSts)は下記のように設定してください。 <table border="0"> <tr> <td>EEP_WP_NONE</td> <td>:</td> <td>No protection</td> </tr> <tr> <td>EEP_WP_UPPER_QUART</td> <td>:</td> <td>Upper-quarter protection setting</td> </tr> <tr> <td>EEP_WP_UPPER_HALF</td> <td>:</td> <td>Upper-half protection setting</td> </tr> <tr> <td>EEP_WP_WHOLE_MEM</td> <td>:</td> <td>Whole memory protection setting</td> </tr> </table> 			EEP_WP_NONE	:	No protection	EEP_WP_UPPER_QUART	:	Upper-quarter protection setting	EEP_WP_UPPER_HALF	:	Upper-half protection setting	EEP_WP_WHOLE_MEM	:	Whole memory protection setting
EEP_WP_NONE	:	No protection													
EEP_WP_UPPER_QUART	:	Upper-quarter protection setting													
EEP_WP_UPPER_HALF	:	Upper-half protection setting													
EEP_WP_WHOLE_MEM	:	Whole memory protection setting													
引数	uint8_t	DevNo	; デバイス番号												
	uint8_t	WpSts	; ライトプロテクト設定データ												
リターン値	<ul style="list-style-type: none"> ・ライトプロテクト設定結果を返します。 <table border="0"> <tr> <td>EEP_OK</td> <td>:</td> <td>Successful operation</td> </tr> <tr> <td>EEP_ERR_PARAM</td> <td>:</td> <td>Parameter error</td> </tr> <tr> <td>EEP_ERR_OTHER</td> <td>:</td> <td>Other error</td> </tr> </table> 			EEP_OK	:	Successful operation	EEP_ERR_PARAM	:	Parameter error	EEP_ERR_OTHER	:	Other error			
EEP_OK	:	Successful operation													
EEP_ERR_PARAM	:	Parameter error													
EEP_ERR_OTHER	:	Other error													
備考	なし														

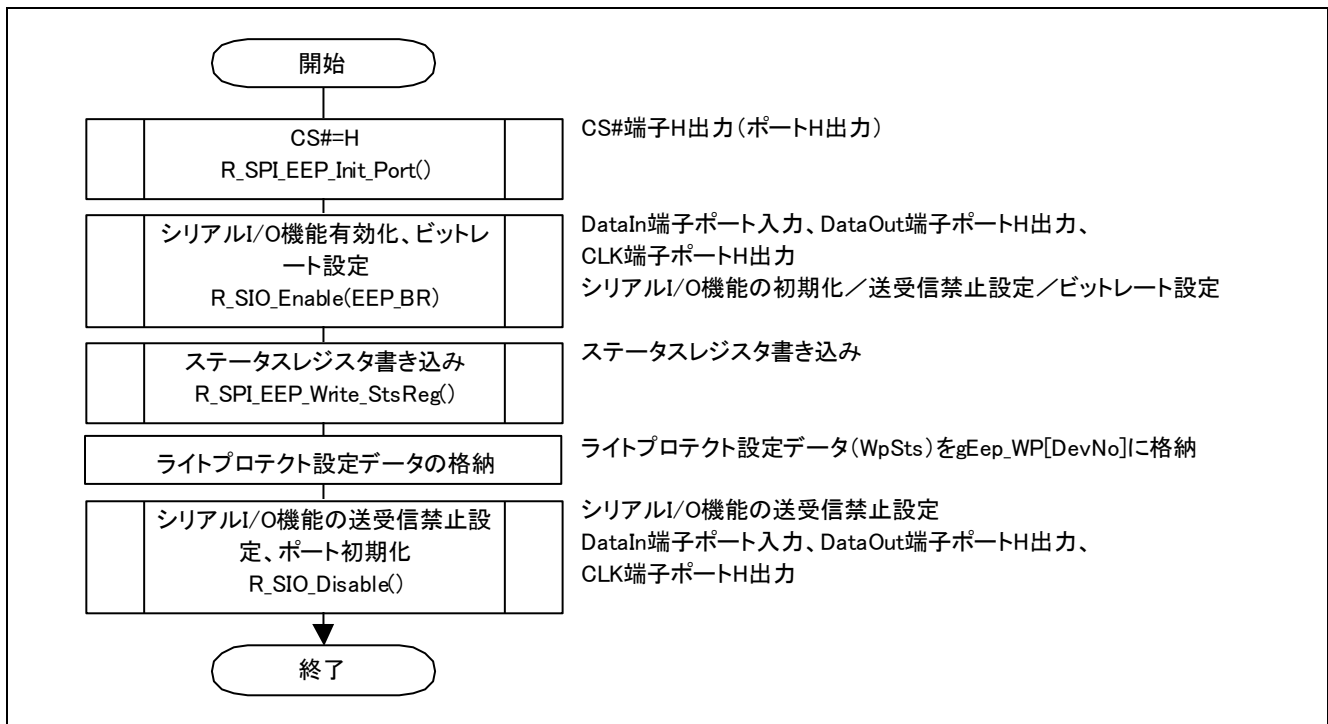


図 5-8 ライトプロテクト処理概要

5.9.4 データ読み出し処理

R_SPI_EEP_Read_Data	
概要	データ読み出し処理
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
宣言	error_t R_SPI_EEP_Read_Data(uint8_t DevNo, uint32_t RAddr, uint32_t RCnt, uint8_t FAR* pData)
説明	・EEPROM から、指定アドレスから指定バイト数分、バイト単位でデータを読み出し、pData に格納します。
引数	uint8_t DevNo ; デバイス番号 uint32_t RAddr ; 読み出し開始アドレス uint32_t RCnt ; 読み出しバイト数 uint8_t FAR* pData ; 読み出しデータ格納バッファポインタ
リターン値	・読み出し結果を返します。 EEP_OK ; Successful operation EEP_ERR_PARAM ; Parameter error EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
備考	・最大読み出しアドレスは、EEPROM 容量-1 です。

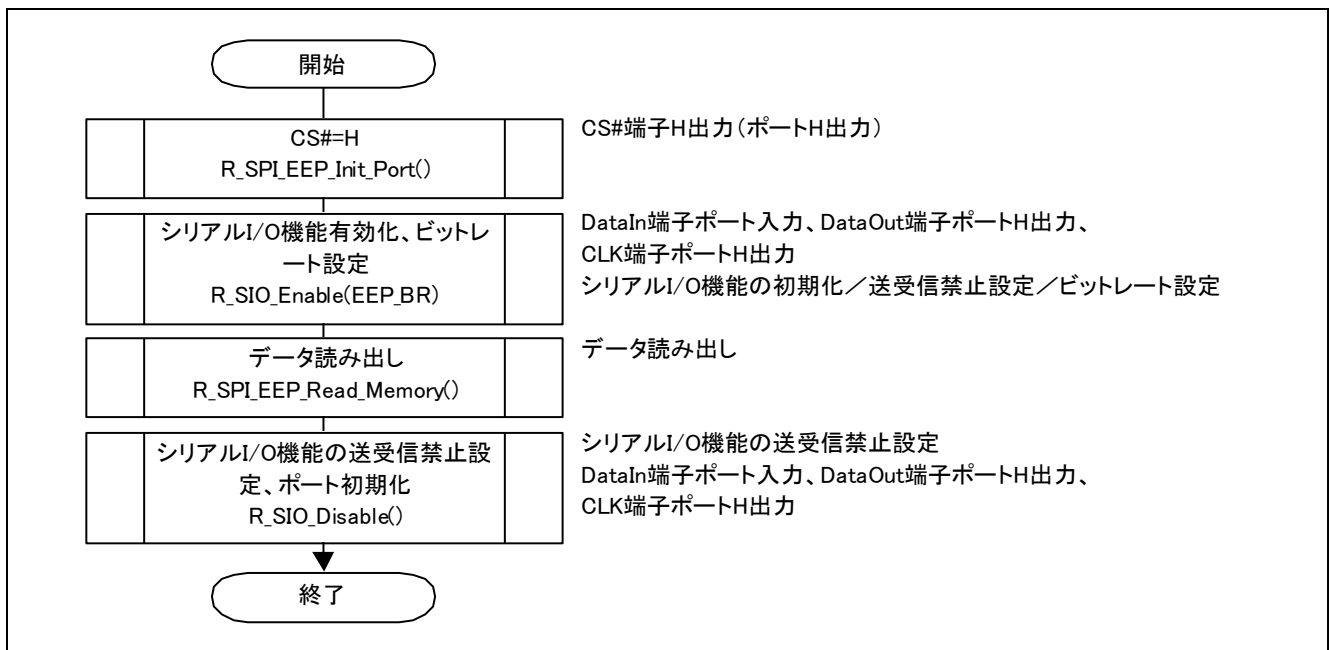


図 5-9 データ読み出し処理概要

5.9.5 データ書き込み処理

R_SPI_EEP_Write_Data

概要	データ書き込み処理		
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h		
宣言	error_t R_SPI_EEP_Write_Data(uint8_t DevNo, uint32_t WAddr, uint32_t WCnt, uint8_t FAR* pData)		
説明	・pData データを EEPROM 上の指定アドレスから指定バイト数分書き込みます。		
引数	uint8_t	DevNo	; デバイス番号
	uint32_t	WAddr	; 書き込み開始アドレス
	uint32_t	WCnt	; 書き込みバイト数
	uint8_t FAR*	pData	; 書き込みデータ格納バッファポインタ
リターン値	・書き込み結果を返します。		
	EEP_OK		; Successful operation
	EEP_ERR_PARAM		; Parameter error
	EEP_ERR_HARD		; Hardware error
	EEP_ERR_WP		; Write-Protection error
	EEP_ERR_OTHER		; Other error
備考	・EEPROM への書き込みは、ライトプロテクト解除状態の場合のみ可能です。 ・プロテクトされたページへの書き込みはできません。エラーを返します。 ・最大書き込みアドレスは、EEPROM 容量-1 です。		

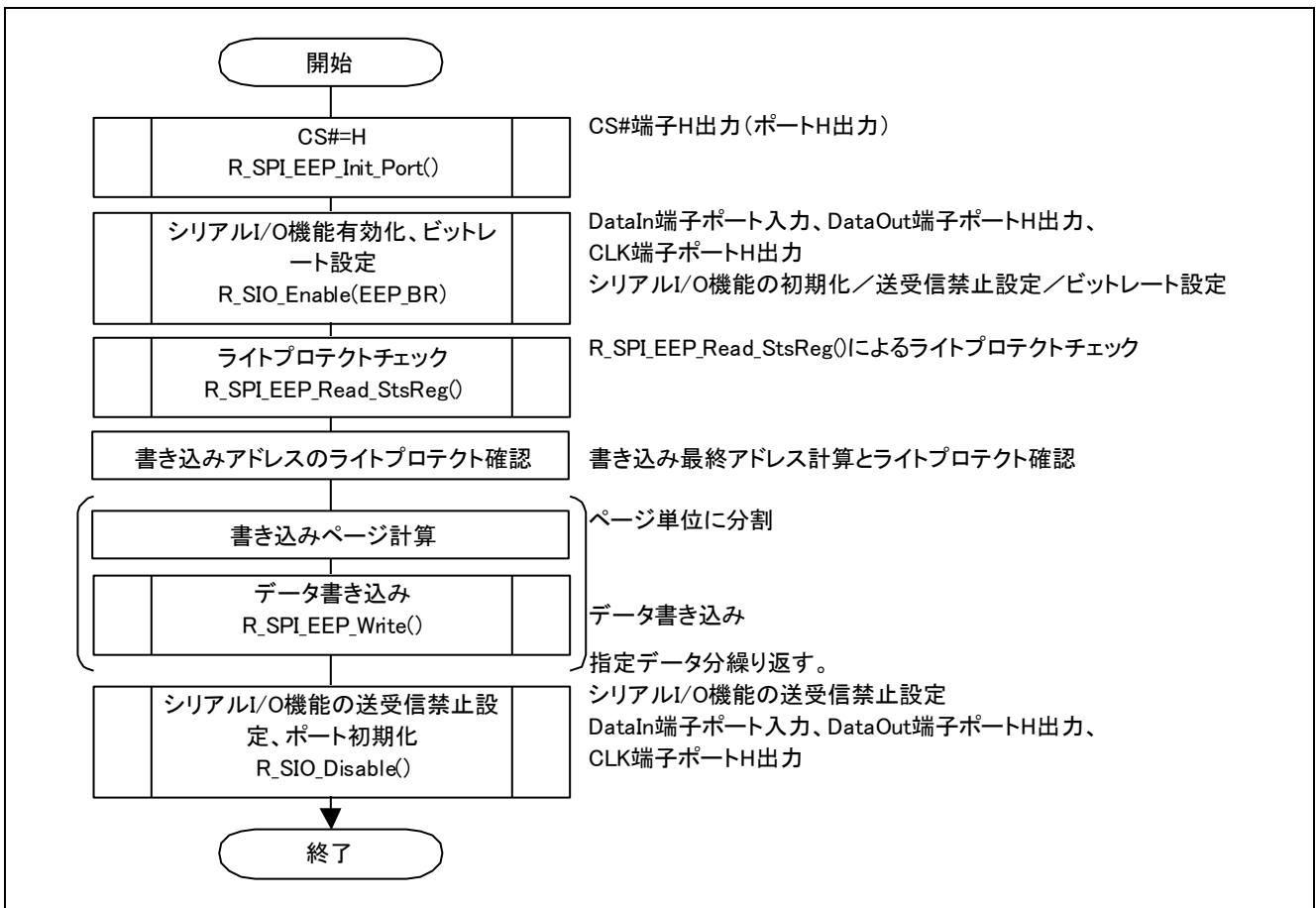


図 5-10 データ書き込み処理概要

5.9.6 ポート初期化処理（内部関数）

R_SPI_EEP_Init_Port

概要	ポート初期化処理（内部関数）	
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h	
宣言	void R_SPI_EEP_Init_Port(uint8_t DevNo)	
説明	・ 指定デバイスのポート（CS#）を初期化し、H 出力します。	
引数	uint8_t	DevNo ; デバイス番号
リターン値	void	
備考	なし	

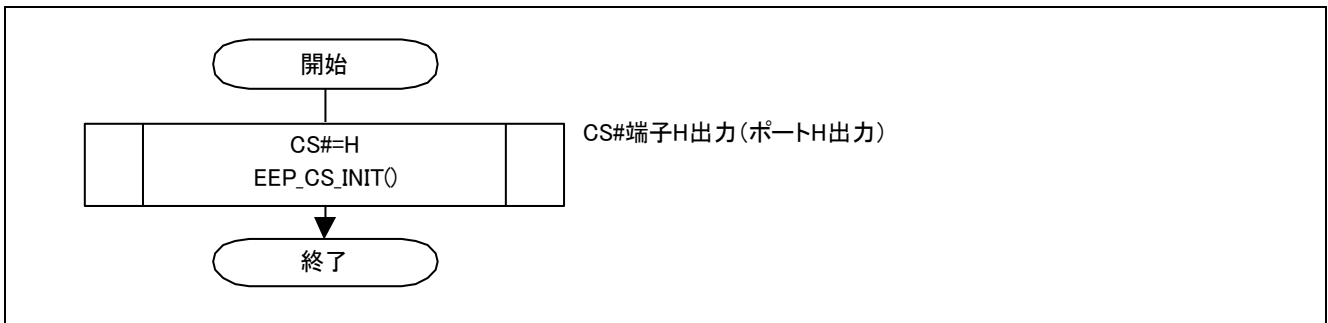


図 5-11 ポート初期化処理概要

5.9.7 RAM 初期化処理 (内部関数)

R_SPI_EEP_Init_Ram

概要	ポート初期化処理 (内部関数)		
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h		
宣言	void R_SPI_EEP_Init_Ram(uint8_t DevNo)		
説明	・ 指定デバイスに関連する RAM を初期化します。		
引数	uint8_t	DevNo	; デバイス番号
リターン値	void		
備考	なし		

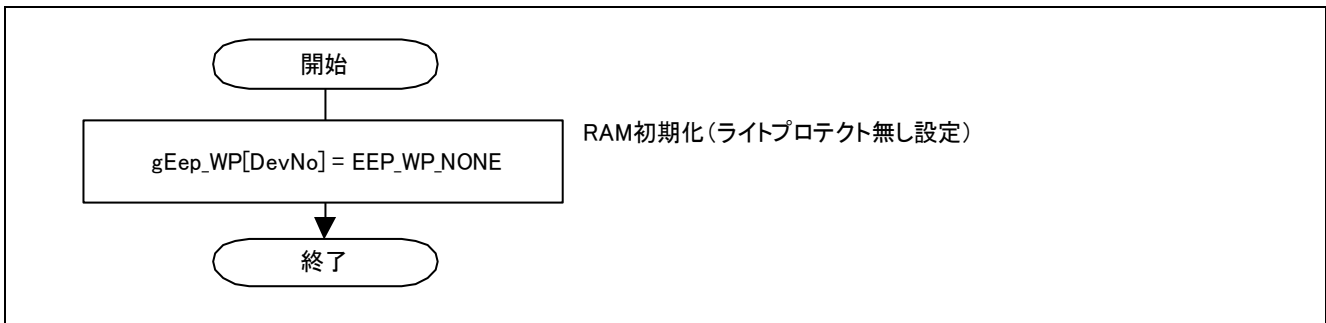


図 5-12 RAM 初期化処理概要

5.9.8 コマンド送信処理（内部関数）

R_SPI_EEP_Send_Cmd	
概要	コマンド送信処理（内部関数）
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
宣言	STATIC error_t R_SPI_EEP_Send_Cmd(uint8_t Cmd, uint32_t Addr, uint8_t CmdSize)
説明	・指定されたコマンドを送信します。
引数	uint8_t Cmd ; コマンドコード uint32_t Addr ; アドレス uint8_t CmdSize ; コマンドサイズ
リターン値	・コマンド送信結果を返します。 EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
備考	なし

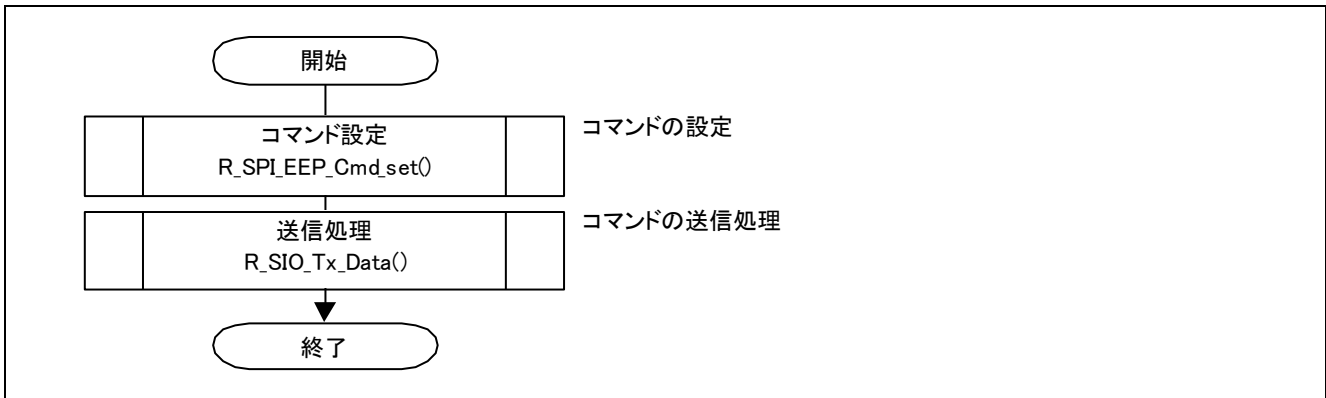


図 5-13 コマンド送信処理概要

5.9.9 読み出しコマンド処理 (内部関数)

R_SPI_EEP_Read_Memory	
概要	読み出しコマンド処理 (内部関数)
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
宣言	error_t R_SPI_EEP_Read_Memory(uint8_t DevNo, uint32_t RAddr, uint32_t RCnt, uint8_t FAR* pData)
説明	・ READ コマンドを使って、EEPROM から、指定アドレスから指定バイト数分、バイト単位でデータを読み出し、pData に格納します。
引数	uint8_t DevNo ; デバイス番号 uint32_t RAddr ; 読み出し開始アドレス uint32_t RCnt ; 読み出しバイト数 uint8_t FAR* pData ; 読み出しデータ格納バッファポインタ
リターン値	・ コマンド処理結果を返します。 EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
備考	・ 読み出し最大アドレスは、EEPROM の容量-1 です。

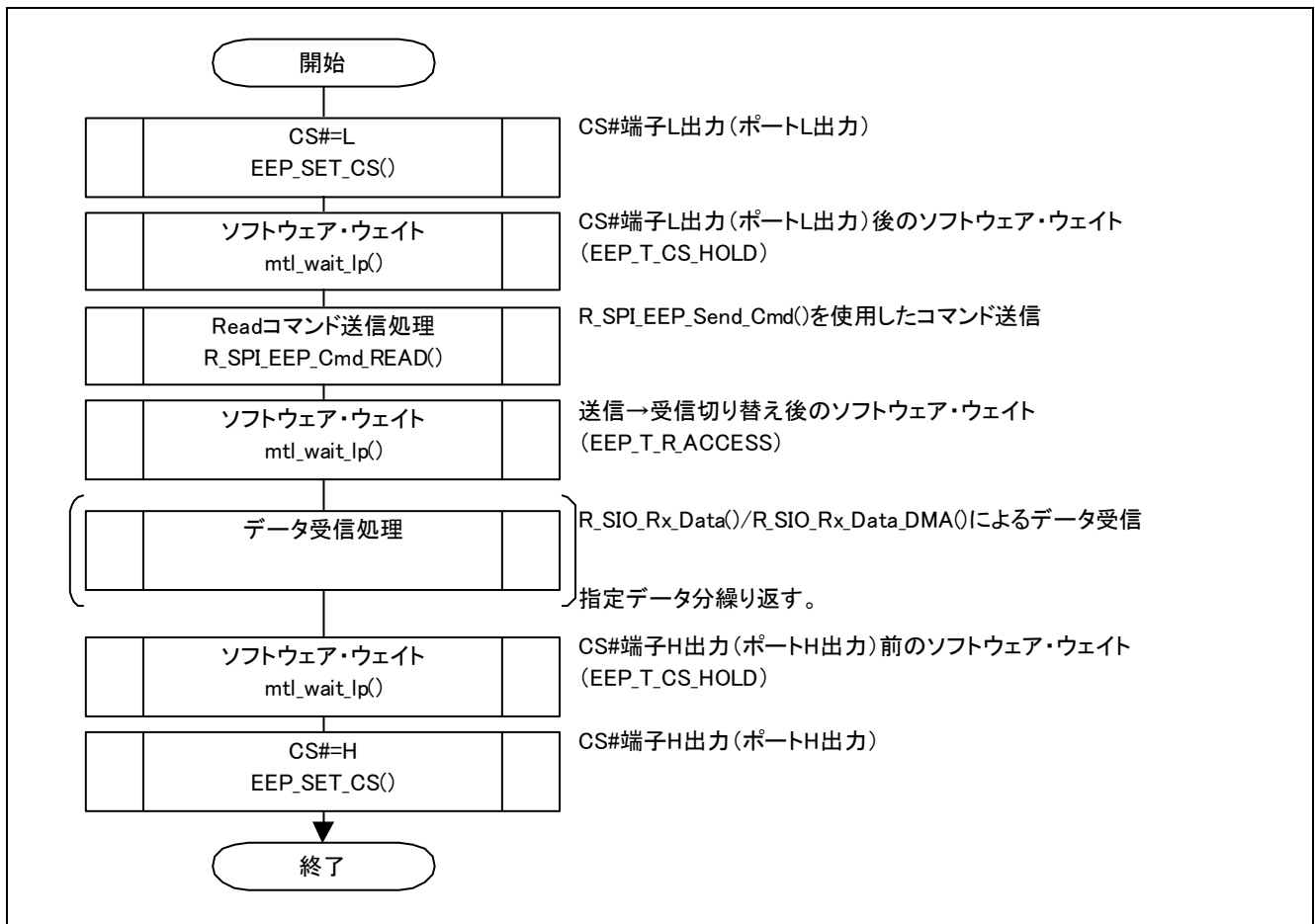


図 5-14 読み出しコマンド処理概要

5.9.10 書き込み許可コマンド処理

R_SPI_EEP_Write_En

概要	書き込み許可コマンド処理	
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h	
宣言	STATIC error_t R_SPI_EEP_Write_En(uint8_t DevNo)	
説明	・WREN コマンドを送信し、指定されたデバイスを書き込み許可設定（WEL ビットセット）します。	
引数	uint8_t DevNo ;	デバイス番号
リターン値	・コマンド処理結果を返します。	
	EEP_OK ;	Successful operation
	EEP_ERR_HARD ;	Hardware error
	EEP_ERR_OTHER ;	Other error
備考	なし	

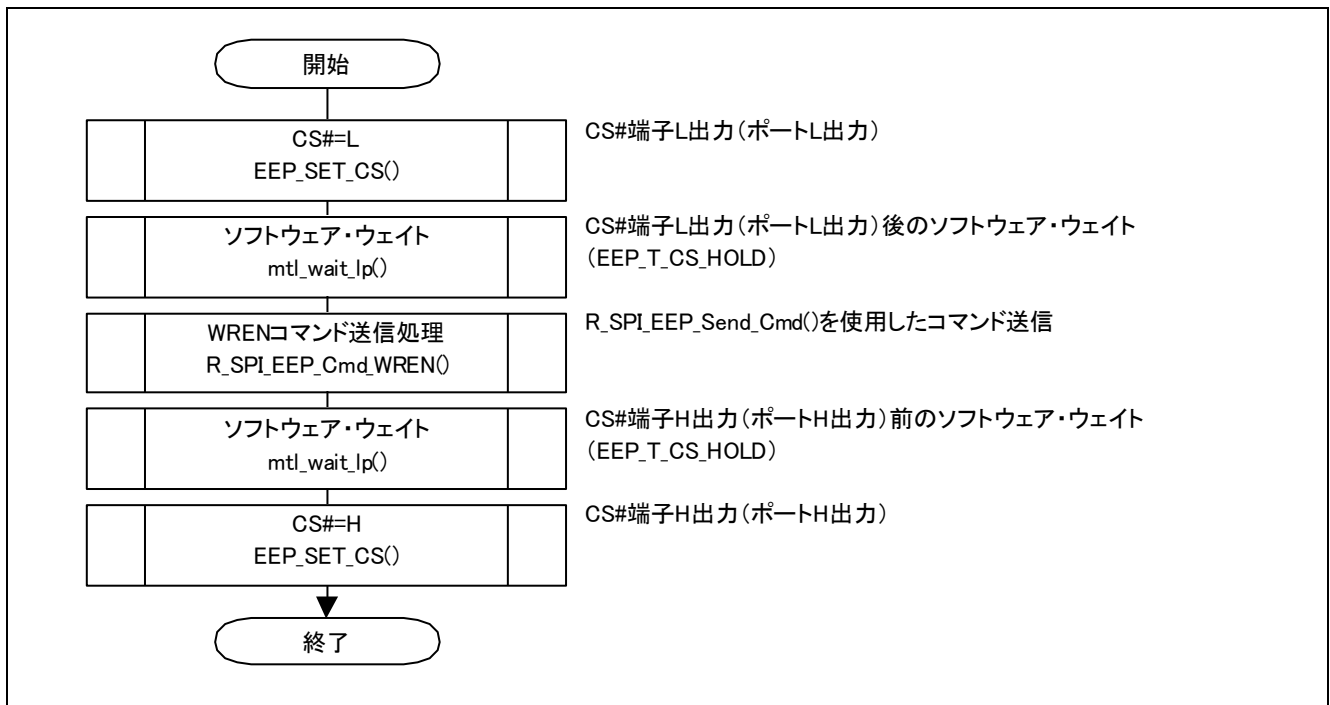


図 5-15 書き込み許可コマンド処理概要

5.9.11 書き込み禁止コマンド処理

R_SPI_EEP_Write_Di

概要	書き込み禁止コマンド処理		
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h		
宣言	STATIC error_t R_SPI_EEP_Write_Di(uint8_t DevNo)		
説明	・WRDI コマンドを送信し、指定されたデバイスを書き込み禁止設定（WEL ビットクリア）します。		
引数	uint8_t	DevNo	; デバイス番号
リターン値	・コマンド処理結果を返します。		
	EEP_OK		; Successful operation
	EEP_ERR_HARD		; Hardware error
	EEP_ERR_OTHER		; Other error
備考	なし		

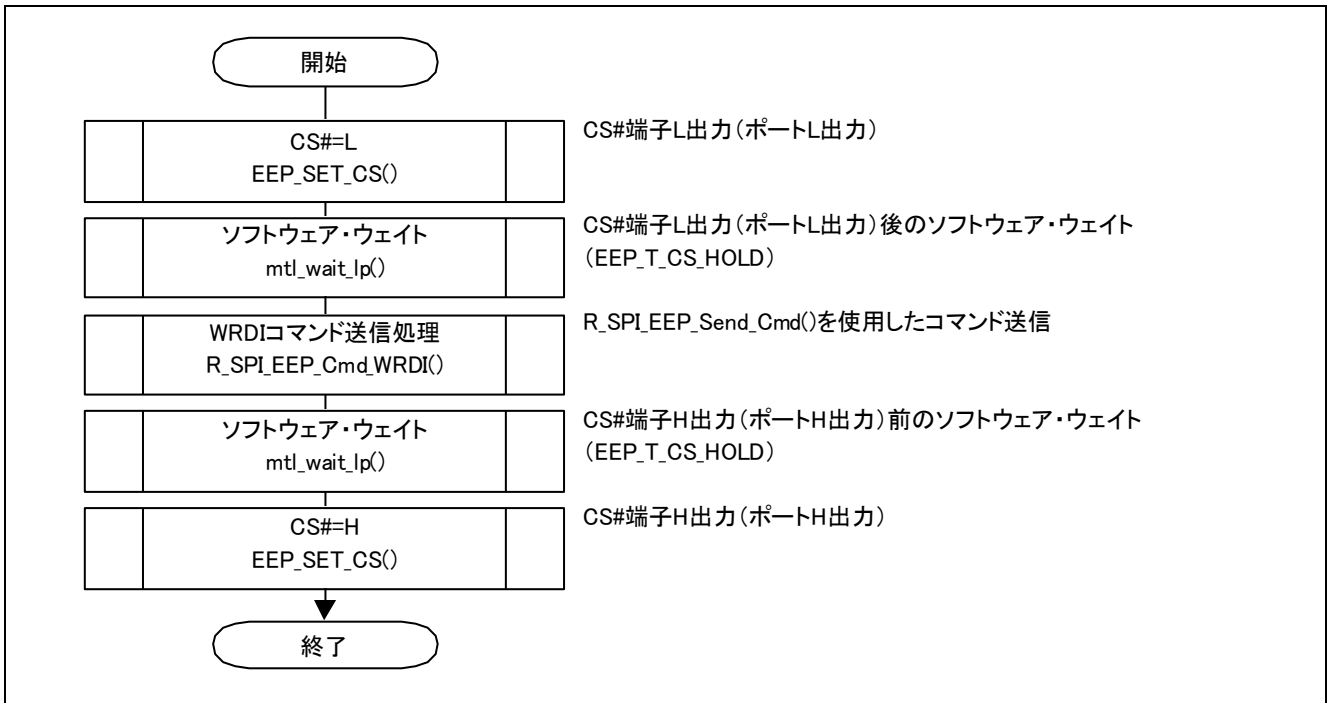


図 5-16 書き込み禁止コマンド処理概要

5.9.12 ステータスレジスタ読み出しコマンド処理 (内部関数)

R_SPI_EEP_Read_StsReg	
概要	ステータスレジスタ読み出しコマンド処理 (内部関数)
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
宣言	error_t R_SPI_EEP_Read_StsReg(uint8_t DevNo, uint8_t FAR* pStsReg)
説明	<ul style="list-style-type: none"> ・ RDSR コマンド送信し、ステータスレジスタを読み出し、pStsReg に格納します。読み出しバッファとして、1 バイトを設定してください。 ・ 読み出しステータス格納バッファ(pStatus)には下記情報が格納されます。 <ul style="list-style-type: none"> ●4 Kbit 以下の場合 <ul style="list-style-type: none"> Bit 7 to 4:Reserved (All "1") Bit 3 to 2:BP1, BP0 00: No protection 01: Upper-quarter protection 10: Upper-half protection 11: Whole memory protection Bit 1:WEL 0: Write disabled 1: Write enabled Bit 0:WIP 1: During write operation ●4 Kbit を超える場合 <ul style="list-style-type: none"> Bit 7:SRWD 0: Status register can be changed. 1: Status register cannot be changed. Bit 6 to 4:Reserved (All "0") Bit3 to 2:BP1, BP0 00: No protection 01: Upper-quarter protection 10: Upper-half protection 11: Whole memory protection Bit 1:WEL 0: Write disabled 1: Write enabled Bit 0:WIP 1: During write operation
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pStsReg ; 読み出しステータス格納バッファポインタ
リターン値	<ul style="list-style-type: none"> ・ コマンド処理結果を返します。 EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
備考	なし

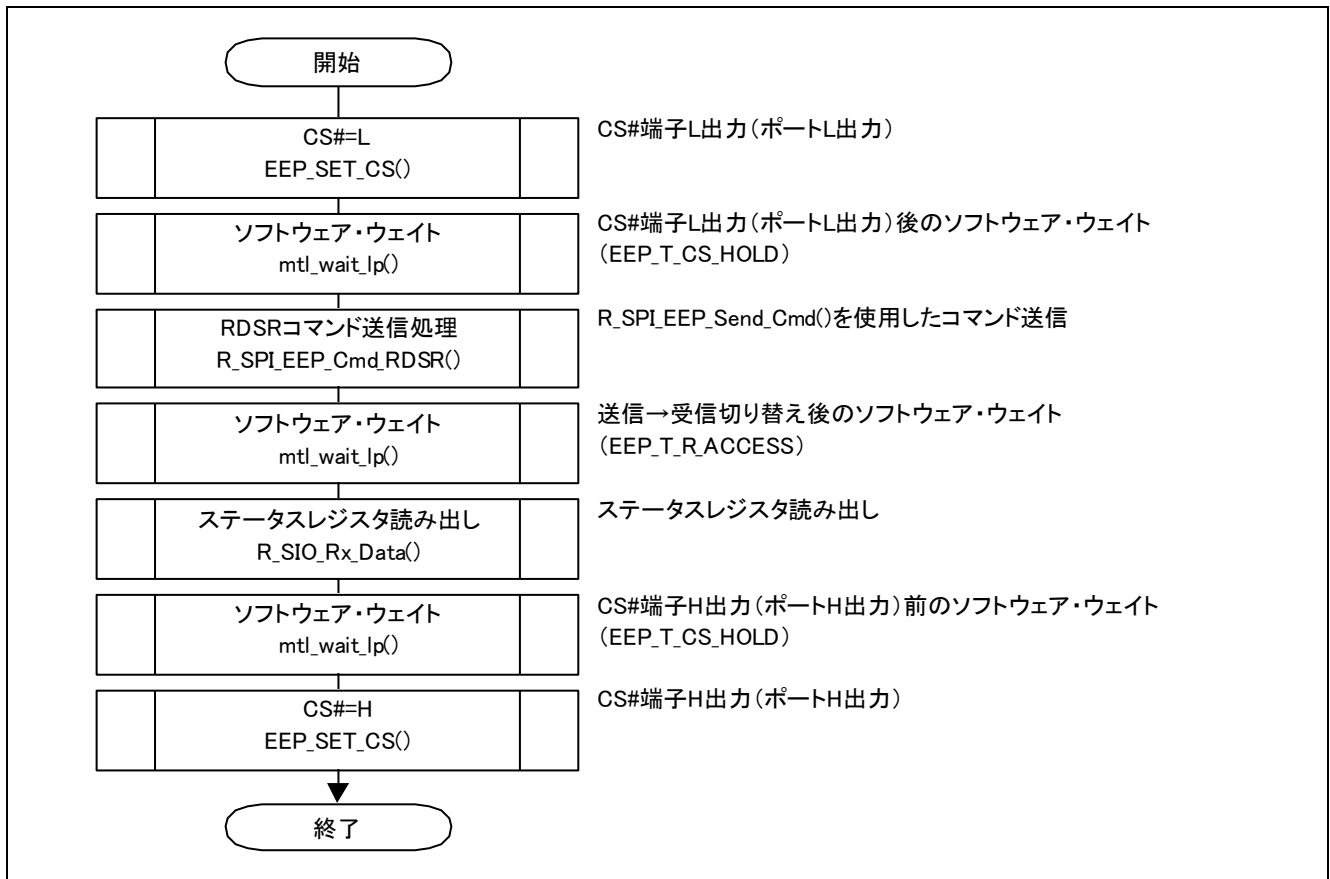


図 5-17 ステータスレジスタ読み出しコマンド処理概要

5.9.13 ステータスレジスタ書き込みコマンド処理（内部関数）

R_SPI_EEP_Write_StsReg									
概要	ステータスレジスタ書き込みコマンド処理（内部関数）								
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h								
宣言	error_t R_SPI_EEP_Write_StsReg(uint8_t DevNo, uint8_t FAR* pStsReg)								
説明	<ul style="list-style-type: none"> ・書き込み許可コマンド処理（WREN コマンド発行処理）を実行し、WRSR コマンド送信し、pStsReg のデータをステータスレジスタに書き込みます。書き込みバッファとして、1 バイトを設定してください。 ・書き込みステータス格納バッファ(pStatus)には下記情報を設定してください。 <ul style="list-style-type: none"> ●4 Kbit 以下の場合 <ul style="list-style-type: none"> Bit 7 to 4:Reserved (All "1") Bits 3 to 2:BP1, BP0 00: No protection 01: Upper-quarter protection 10: Upper-half protection 11: Whole memory protection Bit 1 to 0:Read-only (All "0") ●4 Kbit を超えるの場合 <ul style="list-style-type: none"> Bit 7:SRWD 0: Status register can be changed. 1: Status register cannot be changed. Bit 6 to 4:Reserved (All "0") Bits 3 to 2:BP1, BP0 00: No protection 01: Upper-quarter protection 10: Upper-half protection 11: Whole memory protection Bit 1 to 0:Read-only (All "0") 								
引数	<table style="width: 100%; border: none;"> <tr> <td style="width: 15%;">uint8_t</td> <td style="width: 35%;">DevNo</td> <td style="width: 5%;">;</td> <td style="width: 45%;">デバイス番号</td> </tr> <tr> <td>uint8_t FAR*</td> <td>pStsReg</td> <td>;</td> <td>書き込みステータス格納バッファポインタ</td> </tr> </table>	uint8_t	DevNo	;	デバイス番号	uint8_t FAR*	pStsReg	;	書き込みステータス格納バッファポインタ
uint8_t	DevNo	;	デバイス番号						
uint8_t FAR*	pStsReg	;	書き込みステータス格納バッファポインタ						
リターン値	<ul style="list-style-type: none"> ・コマンド処理結果を返します。 EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error 								
備考	<ul style="list-style-type: none"> ・ステータスレジスタの Reserved のビットに対しては、読み出し値を同じ値を設定しています。 								

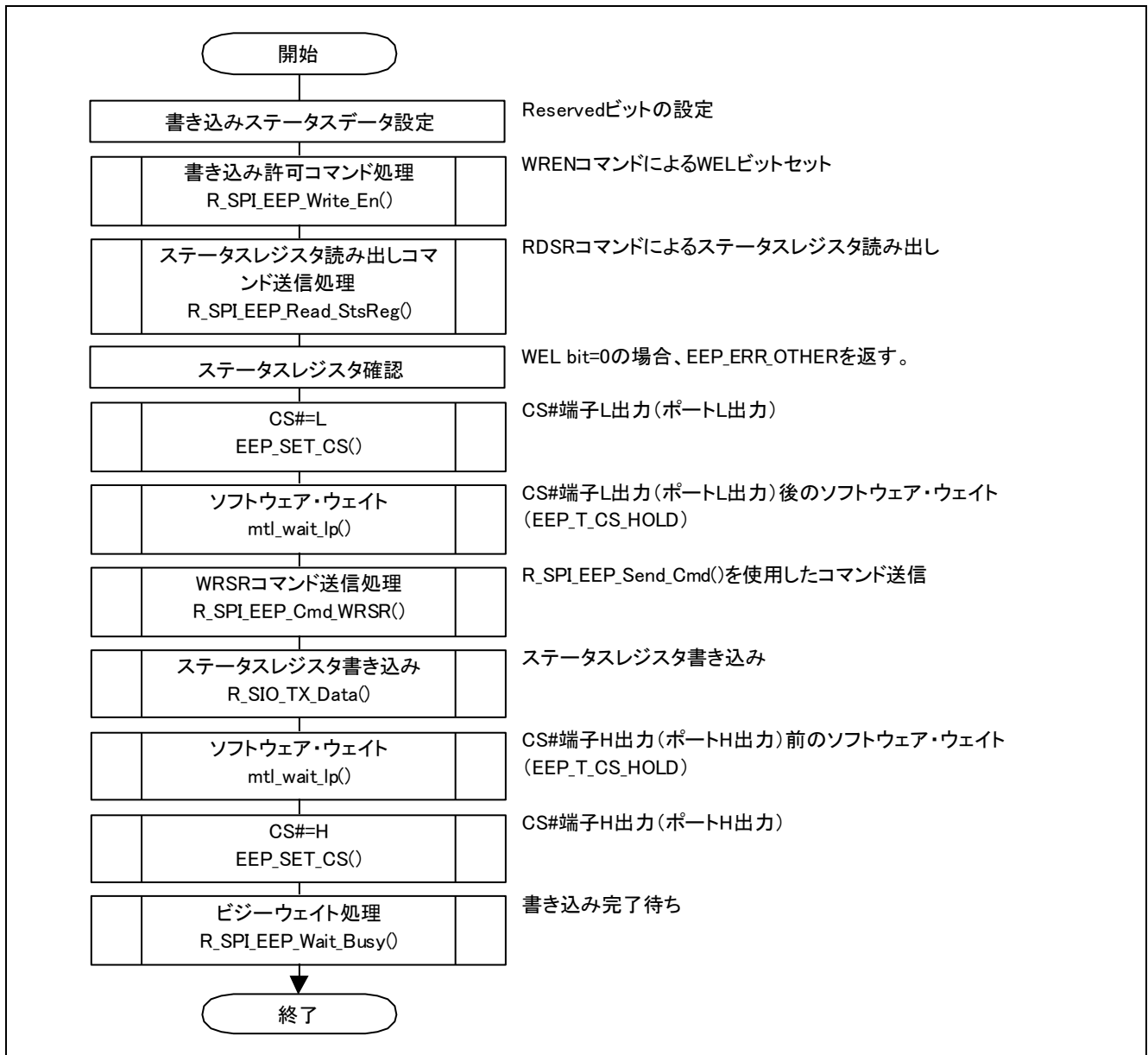


図 5-18 ステータスレジスタ書き込みコマンド処理概要

5.9.14 ビジーウェイト処理（内部関数）

R_SPI_EEP_Wait_WBusy													
概要	ビジーウェイト処理（内部関数）												
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h												
宣言	STATIC error_t R_SPI_EEP_Wait_WBusy(uint8_t DevNo, uint16_t BusyTime, uint16_t BusyCnt)												
説明	<ul style="list-style-type: none"> ・ステータスレジスタ読み出しコマンド処理を使って、BusyCnt=0 の場合、BusyTime 時間間隔で、ビジーの期間をウェイトします。 ・ステータスレジスタ読み出しコマンド処理を使って、BusyCnt≠0 の場合、BusyTime 時間間隔で、BusyCnt 回数分、ビジーの期間をウェイトします。 												
引数	<table border="0"> <tr> <td>uint8_t</td> <td>DevNo</td> <td>;</td> <td>デバイス番号</td> </tr> <tr> <td>uint16_t</td> <td>BusyTime</td> <td>;</td> <td>ステータス確認用ウェイト時間</td> </tr> <tr> <td>uint16_t</td> <td>BusyCnt</td> <td>;</td> <td>カウンタ</td> </tr> </table>	uint8_t	DevNo	;	デバイス番号	uint16_t	BusyTime	;	ステータス確認用ウェイト時間	uint16_t	BusyCnt	;	カウンタ
uint8_t	DevNo	;	デバイス番号										
uint16_t	BusyTime	;	ステータス確認用ウェイト時間										
uint16_t	BusyCnt	;	カウンタ										
リターン値	<ul style="list-style-type: none"> ・結果を返します。 <table border="0"> <tr> <td>EEP_OK</td> <td>;</td> <td>Successful operation</td> </tr> <tr> <td>EEP_ERR_HARD</td> <td>;</td> <td>Hardware error</td> </tr> <tr> <td>EEP_ERR_OTHER</td> <td>;</td> <td>Other error</td> </tr> </table>	EEP_OK	;	Successful operation	EEP_ERR_HARD	;	Hardware error	EEP_ERR_OTHER	;	Other error			
EEP_OK	;	Successful operation											
EEP_ERR_HARD	;	Hardware error											
EEP_ERR_OTHER	;	Other error											
備考	なし												

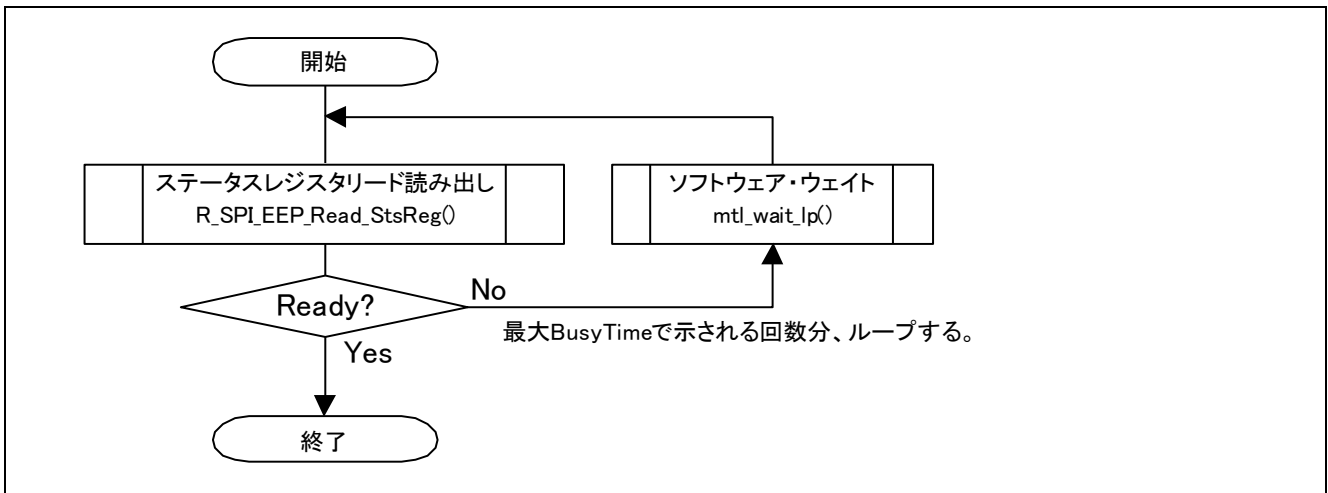


図 5-19 ビジーウェイト処理概要

5.9.15 ページ書き込みコマンド処理（内部関数）

R_SPI_EEP_Write_Page																	
概要	ページ書き込みコマンド処理（内部関数）																
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h																
宣言	error_t R_SPI_EEP_Write_Page(uint8_t DevNo, uint32_t WAddr, uint32_t WCnt, uint8_t FAR* pData)																
説明	<ul style="list-style-type: none"> ・WRITE コマンドを使って、pData のデータを EEPROM 上の指定アドレスから指定バイト数分書き込みます。 																
引数	<table> <tr> <td>uint8_t</td> <td>DevNo</td> <td>;</td> <td>デバイス番号</td> </tr> <tr> <td>uint32_t</td> <td>WAddr</td> <td>;</td> <td>書き込み開始アドレス</td> </tr> <tr> <td>uint32_t</td> <td>WCnt</td> <td>;</td> <td>書き込みバイト数</td> </tr> <tr> <td>uint8_t FAR*</td> <td>pData</td> <td>;</td> <td>書き込みデータ格納バッファポインタ</td> </tr> </table>	uint8_t	DevNo	;	デバイス番号	uint32_t	WAddr	;	書き込み開始アドレス	uint32_t	WCnt	;	書き込みバイト数	uint8_t FAR*	pData	;	書き込みデータ格納バッファポインタ
uint8_t	DevNo	;	デバイス番号														
uint32_t	WAddr	;	書き込み開始アドレス														
uint32_t	WCnt	;	書き込みバイト数														
uint8_t FAR*	pData	;	書き込みデータ格納バッファポインタ														
リターン値	<ul style="list-style-type: none"> ・書き込み結果を返します。 <table> <tr> <td>EEP_OK</td> <td>;</td> <td>Successful operation</td> </tr> <tr> <td>EEP_ERR_HARD</td> <td>;</td> <td>Hardware error</td> </tr> <tr> <td>EEP_ERR_OTHER</td> <td>;</td> <td>Other error</td> </tr> </table>	EEP_OK	;	Successful operation	EEP_ERR_HARD	;	Hardware error	EEP_ERR_OTHER	;	Other error							
EEP_OK	;	Successful operation															
EEP_ERR_HARD	;	Hardware error															
EEP_ERR_OTHER	;	Other error															
備考	<ul style="list-style-type: none"> ・ページを超える書き込みは、許可されていません。 ・プロテクトされたページには、書き込みができません。エラーを返しません。 																

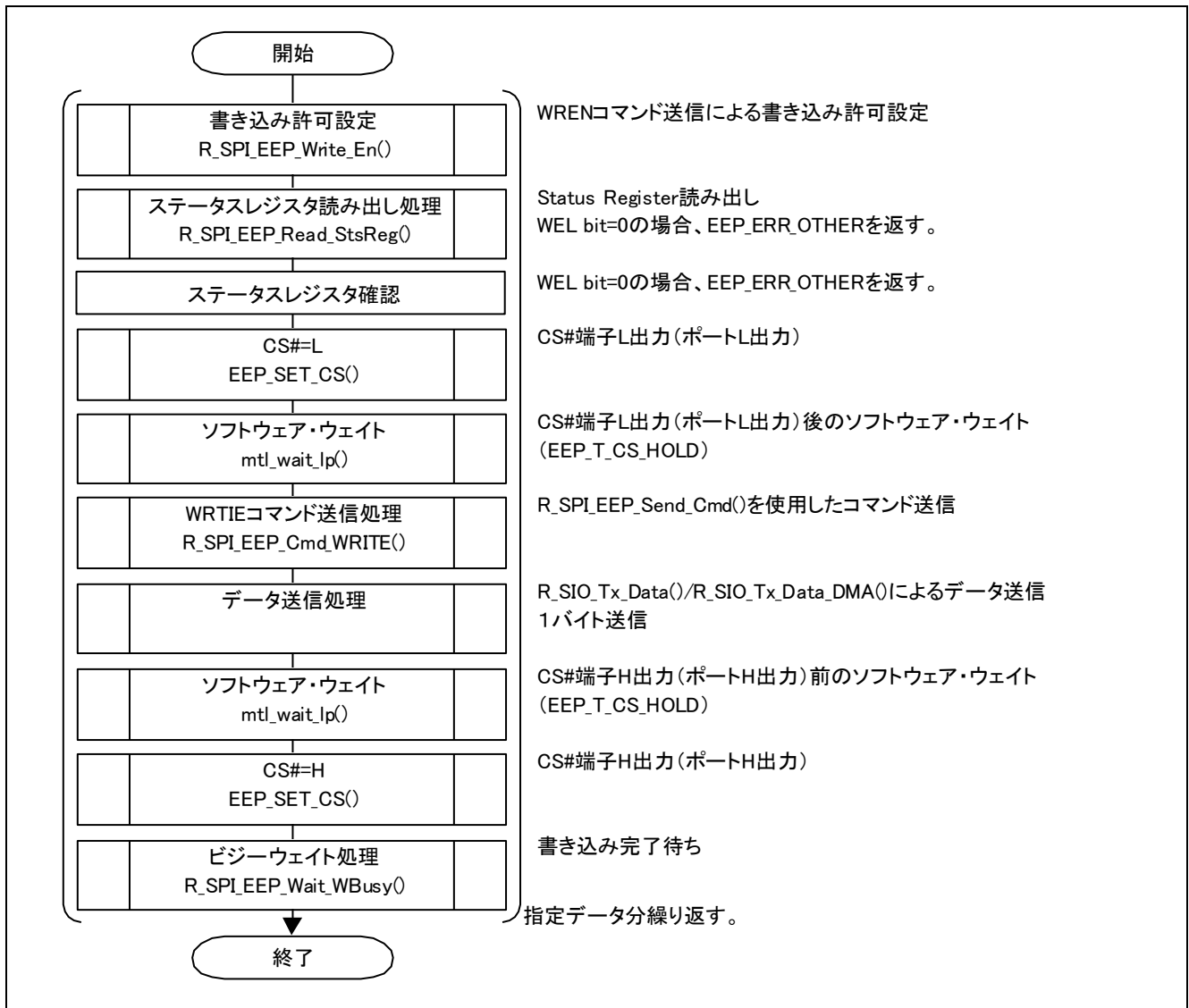


図 5-20 ページ書き込みコマンド処理概要

5.9.16 コマンド設定処理（内部関数）

R_SPI_EEP_Cmd_set

概要	コマンド設定処理（内部関数）		
ヘッダ	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h		
宣言	STATIC void R_SPI_EEP_Cmd_set(uint8_t Cmd, uint32_t Addr, uint8_t CmdSize)		
説明	・ コマンドとアドレスを設定します。エンディアンに応じて、変換します。		
引数	uint8_t	Cmd	; コマンド（命令コード）
	uint32_t	Addr	; アドレス情報
	uint8_t	CmdSize	; コマンドサイズ
リターン値	なし		
備考	・ エンディアンは、MTL_MCU_LITTLE (mtl_com.h)にて、指定してください。		

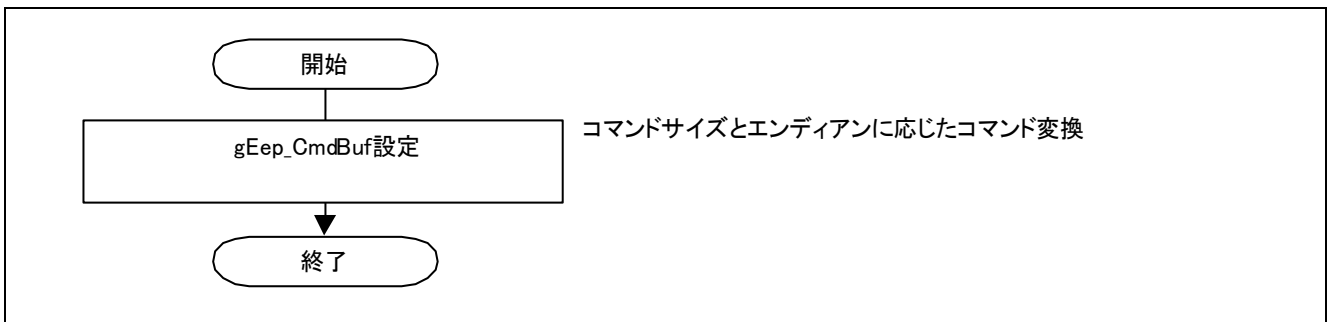


図 5-21 コマンド設定処理概要

6. 応用例

Serial EEPROM 制御部分（シリアル I/O 制御部分は、対象外）の設定例を示します。

シリアル I/O 制御部分は、MCU 個別のクロック同期式シングルマスタ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。

なお、通信速度は、スレーブデバイス個別の設定が必要なため、本サンプルコードにて設定します。

設定箇所は、各ファイル中の「`/** SET **/`」というコメントの部分です。

また、共通で使用される関数（`mtl_wait_lp()`等）は、MCU 個別のクロック同期式シングルマスタ制御ソフトウェアに含まれているものを使用してください。

6.1 Serial EEPROM 制御ソフトウェアの設定

設定箇所は、各ファイル中の「/** SET **/」というコメントの部分です。

6.1.1 R_SPI_EEP.h

本 Serial EEPROM 用の定義ファイルです。

設定箇所は、各ファイル中の「/** SET **/」というコメントの部分です。

(1) 使用するデバイスの個数とデバイス番号の定義

使用するデバイスの個数を指定し、各デバイスに番号を割り当ててください。

下記は、デバイスを 1 個使用し、デバイス番号を 0 に割り当てる場合の例です。

最大 2 個までの制御が可能です。

```

/*-----*/
/* Define number of required serial EEPROM devices.(1~N devices)          */
/* Define the device number in accordance with the number of serial EEPROM devices */
/* to be connected.                                                         */
/*-----*/
/* Define no. of devices */
#define EEP_DEV_NUM          1          /* 1devices          */

/* Define no. of slots */
#define EEP_DEV0             0          /* Device 0          */
#define EEP_DEV1             1          /* Device 1          */

```

(2) 使用するデバイスの容量の定義

使用するデバイスの容量を指定してください。

下記は、4 Kbit デバイスを使用する場合の例です。

```

/*-----*/
/* Define the serial EEPROM device.                                         */
/*-----*/
/*#define EEP_SIZE_002K          */ /* 2kbit   (256 Byte)          */
#define EEP_SIZE_004K          /* 4kbit   (512 Byte)          */
/*#define EEP_SIZE_008K          */ /* 8kbit   ( 1kByte)          */
/*#define EEP_SIZE_016K          */ /* 16kbit  ( 2kByte)          */
/*#define EEP_SIZE_032K          */ /* 32kbit  ( 4kByte)          */
/*#define EEP_SIZE_064K          */ /* 64kbit  ( 8kByte)          */
/*#define EEP_SIZE_128K          */ /* 128kbit (16kByte)          */
/*#define EEP_SIZE_256K          */ /* 256kbit (32kByte)          */
/*#define EEP_SIZE_512K          */ /* 512kbit (64kByte)          */

```

6.1.2 R_SPI_EEP_sfr.h

R_SPI_EEP_sfr.h.XXX は、各 MCU に評価目的で作成したものです。どれか一つを R_SPI_EEP_sfr.h にリネームして使用してください。対象 MCU のものが無い場合には、参照して、R_SPI_EEP_sfr.h を作成してください。

設定箇所は、各ファイル中の「**** SET ****」というコメントの部分です。

(1) チップセレクト信号設定

使用するチップセレクト信号のポートを定義してください。

2 つ目のデバイスを接続する場合、2 つ目のポートも定義してください。

以下は、ポート 66 を使用する場合での記述例を示しています。

```

/*-----*/
/* Define the CS port.                                     */
/*-----*/
#define EEP_DR_CS0    PORT6.DR.BIT.B6    /* EEPROM CS0  (Negative-true logic) */
#define EEP_DDR_CS0  PORT6.DDR.BIT.B6   /* EEPROM CS1  (Negative-true logic) */

#if (EEP_DEV_NUM > 1)
#define EEP_DR_CS1    /* EEPROM CS1  (Negative-true logic) */
#define EEP_DDR_CS1  /* EEPROM CS1  (Negative-true logic) */
#endif /* #if (EEP_DEV_NUM > 1) */

```

(2) 通信速度の設定

通信速度を設定してください。単位は、bit/sec です。

設定値は、使用する MCU とシリアル I/O に依存します。

以下は、RX610 の SCI 3.125Mbps(bit/sec)の例です。

```

/*-----*/
/* Define the value of the bit rate register according to a communication baud
rate.          */
/* The possible maximum transfer frequency of CLK is depends on hardware circuit
*/
/* and MCU conditions.                                     */
/* Refer to MCU hardware manual/memory card specifications and specify the buad
rate.          */

/* PCLK = 50MHz, n=0 for RX610 SCI */
#define EEP_BR          (uint8_t)0x03          /* BRR initial setting          */
/*          +----- 3.125MHz          */

```

設定値は、MCU のハードウェアマニュアルを参考に設定してください。

6.1.3 R_SPI_EEP_io.c

本 Serial EEPROM 用の I/O モジュールファイルです。

設定箇所は、各ファイル中の「/** SET **/」というコメントの部分です。

(1) SFR 領域用デファインの設定

RL78 ファミリ、もしくは 78K0R を使用する場合、使用する C コンパイラには、定義済プリプロセッサシンボルがあります。この定義済プリプロセッサシンボルを使用し、プログラムを記述済です。

また、使用する MCU が RL78 ファミリ、もしくは 78K0R であり、かつ、IAR Systems 製の統合開発環境を使用する場合には、使用する MCU の SFR が定義されているヘッダファイルを設定する必要があります。

MCU 個別のクロック同期式シングルマスタ制御ソフトウェアも参照してください。

本設定は、SPI スレーブデバイスセレクト制御信号用に使用します。

表 6-1 MCU と SFR 領域用デファインの設定

統合開発環境	MCU	SFR 設定の要・不要	設定方法
CubeSuite+ CS+	RL78	不要	不要
	78K0R	不要	不要
	RX	不要	不要
IAR Embedded Workbench	RL78	要	#ifdef __ICCRL78__ #include <ior5f104pj.h> ←MCU に合わせて変更 #include <ior5f104pj_ext.h> ←MCU に合わせて変更 #endif
	78K0R	要	#ifdef __ICC78K__ #include <io78f1009_64.h> ←MCU に合わせて変更 #include <io78f1009_64_ext.h> ←MCU に合わせて変更 #endif
	RX	(本ソフトウェアは 未サポート)	(本ソフトウェアは未サポート)

以下は、RL78/G14 100pin を使用する場合の例です。

```
#ifdef __ICCRL78__                /* IAR RL78 Compiler          */
#include <ior5f104pj.h>            /* for RL78/G14 100pin (R5F104PJ) */
#include <ior5f104pj_ext.h>       /* for RL78/G14 100pin (R5F104PJ) */
#endif /* __ICCRL78__ */
```

7. 使用上の注意事項

7.1 組み込み時の注意事項

本サンプルコードを組み込む場合は、R_SPI_EEP.h をインクルードしてください。

7.2 キャッシュ搭載の MCU を使用する場合

読み出し／書き込み用データ格納バッファは、非キャッシュ領域を指定してください。

7.3 他スレーブデバイスを使用する場合

同一 SPI バス上で、他スレーブデバイスを制御することが可能です。

スレーブデバイス制御ソフトウェアを作成する場合は、本サンプルコードを参考にしてください。

また、スレーブデバイス制御ソフトウェア毎に、通信速度設定が可能です。

RX ファミリ、RL78 ファミリ、78K0R/Kx3-L

Renesas R1EX25xxx シリーズ Serial EEPROM 制御ソフトウェア

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問い合わせ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2011.06.30	—	初版発行
1.01	2011.08.31	全頁	ヘッダの対象マイコンを MCU に変更
		1	動作確認に使用した MCU に、78K0R/Kx3-L を追加
		3	2. 動作確認条件 に、(2) 78K0R/Kx3-L の場合 を追加
		4	3. 関連アプリケーションノート に、78K0R/Kx3-L 用アプリケーションノート(R01AN0708JJ) を追加
		12	5.3 必要メモリサイズ (1) RX610 の場合 に、最大使用ユーザスタックサイズ条件を追加
		12	5.3 必要メモリサイズ に、78K0R/Kx3-L の場合 を追加
		13	5.4 ファイル構成 にて、アプリケーションノートのファイル名を更新
		13	5.4 ファイル構成 に、R_SPI_EEP_sfr.h.78k0r を追加
		41	6.1.3 R_SPI_EEP_io.c を追加
1.02	2012.03.31	1	対象デバイス に、RX62N グループ を追加
		3-4	2. 動作確認条件 の「(1)RX610 の場合」を「(1)RX610 SCI の場合」に変更。評価に使用したソフトウェア欄の内容を変更
		3-4	2. 動作確認条件 の「(2)78K0R/Kx3-L の場合」を「(2)78K0R/Kx3-L SAU の場合」に変更。評価に使用したソフトウェア欄の内容を変更
		4	2. 動作確認条件 に、(3)RX62N RSPI の場合 を追加
		4	3. 関連アプリケーションノート に、アプリケーションノート R01AN0534JJ の対象グループ名を変更
		4	3. 関連アプリケーションノート に、アプリケーションノート R01AN0323JJ の対象グループ名を変更
		13	5.3 必要メモリサイズ の「(1)RX610 の場合」を「(1)RX610 SCI の場合」に変更。
		13	5.3 必要メモリサイズ の「(2)78K0R/Kx3-L の場合」を「(2)78K0R/Kx3-L SAU の場合」に変更。
		14	5.3 必要メモリサイズ に、(3)RX62N RSPI の場合 を追加
		15	5.4 ファイル構成 に、アプリケーションノート番号を変更
		15	5.4 ファイル構成 に、R_SPI_EEP_sfr.h.rx62x を追加
		1.03	2012.05.15
4	2. 動作確認条件 に、(4)RX62N SCI の場合 を追加		
5	3. 関連アプリケーションノート に、RX62N グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1088JJ) を追加		
14	5.3 必要メモリサイズ に、(4)RX62N SCI の場合 を追加		
15	5.4 ファイル構成 に、アプリケーションノート番号を変更		
1.04	2013.11.29	1	対象デバイス に、RX63N、RX63T、RX210、RX21A、RX220、RL78/G14 を追加
		7-12	2. 動作確認条件 に、以下を追加 (5) RX63N RSPI の場合 (6) RX63N SCI の場合 (7) RX63T RSPI の場合 (8) RX63T SCI の場合 (9) RX210 RSPI の場合 (10) RX210 SCI の場合

			(11) RX21A RSPI の場合 (12) RX21ASCI の場合 (13) RX220 RSPI の場合 (14) RX220 SCI の場合 (15) RL78/G14 SAU 統合開発環境 CubeSuite+の場合 (16) RL78/G14 SAU 統合開発環境 IAR の場合
		13	3. 関連アプリケーションノート に、以下を追加 RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1196JJ) RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1229JJ) RL78/G14 シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1195JJ0102)
		23-28	5.3 必要メモリサイズ に、以下を追加 (5) RX63N RSPI の場合 (6) RX63N SCI の場合 (7) RX63T RSPI の場合 (8) RX63T SCI の場合 (9) RX210 RSPI の場合 (10) RX210 SCI の場合 (11) RX21A RSPI の場合 (12) RX21ASCI の場合 (13) RX220 RSPI の場合 (14) RX220 SCI の場合 (15) RL78/G14 SAU 統合開発環境 CubeSuite+の場合 (16) RL78/G14 SAU 統合開発環境 IAR の場合
		29	5.4 ファイル構成 アプリケーションノート番号を変更 ソースのフォルダ名を変更 新規デバイスのレジスタ用共通定義を追加
		56	6.1.2 R_SPI_EEP_sfr.h 内容を修正した。
		57	6.1.3 R_SPI_EEP_io.c 内容を修正した。
1.05	2014.04.30	1	要旨 にクロック同期式シングルマスタ制御ソフトウェア情報の URL を追加した。
		1	対象デバイス に、RX111、RL78/G1C、RL78/L1C、RL78/L12、RL78/L13 を追加
		4,12	2. 動作確認条件 2.1 RX ファミリと 2.2 RL78 ファミリ、78K0R/Kx3-L を追加した。
		11	2.1 RX ファミリに、以下を追加 (14) RX111 RSPI の場合 (15) RX111 SCI の場合
		12	2.2 RL78 ファミリ、78K0R/Kx3-L (1) 78K0R/Kx3-LSAU の場合 にて、「エミュレータ Minicube2」を削除した
		13	2.2 RL78 ファミリ、78K0R/Kx3-L (2) RL78/G14 SAU 統合開発環境 CubeSuite+の場合 にて、「エミュレータ Minicube2」を削除した
		13	2.2 RL78 ファミリ、78K0R/Kx3-L (3) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench の場合 にて、

			「エミュレータ Minicube2」を削除した
		14-17	2.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加 (4) RL78/G1C SAU 統合開発環境 CubeSuite+の場合 (5) RL78/G1C SAU 統合開発環境 IAR Embedded Workbench の場合 (6) RL78/L12 SAU 統合開発環境 CubeSuite+の場合 (7) RL78/L12 SAU 統合開発環境 IAR Embedded Workbench の場合 (8) RL78/L13 SAU 統合開発環境 CubeSuite+の場合 (9) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合 (10) RL78/L1C SAU 統合開発環境 CubeSuite+の場合 (11) RL78/L1C SAU 統合開発環境 IAR Embedded Workbench の場合
		18	3.関連アプリケーションノート マイコングループ追加によりタイトルを更新した。 RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) 元は、RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ)であった。 RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) 元は、RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ)であった。 RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ) 元は、RL78/G14 シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ)であった。
		26,34	5.3 必要メモリサイズ 5.3.1 RX ファミリと 5.3.2 RL78 ファミリ、78K0R/Kx3-L の章番号を追加し、ファミリ別に分けた。
		33	5.3.1 RX ファミリに、以下を追加 (14) RX111 RSPI の場合 (15) RX111 SCI の場合
		35	5.3.2 RL78 ファミリ、78K0R/Kx3-L (3) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench の場合 ROM サイズ 元は、1408 バイトと 1233 バイトであった。 最大使用ユーザスタックサイズ 元は、120 バイトであった。
		36	5.3.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加 (4) RL78/L13 SAU 統合開発環境 CubeSuite+の場合 (5) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合
		37	5.4 ファイル構成 アプリケーションノート番号を変更 ソースのフォルダ名を変更 新規デバイスのレジスタ用共通定義を追加
1.06	2016.03.31	14	2.2RL78 ファミリ、78K0R/Kx3-L (2) RL78/G14 SAU 統合開発環境 CS+ for CA,CX の場合

			(コンパイラ : CA78K0R) を更新した。 (3) RL78/G14 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL) を追加した。
		37	5.3.2 RL78 ファミリ、78K0R/Kx3-L (2) RL78/G14 SAU 統合開発環境 CS+ for CA,CX の場合 (コンパイラ : CA78K0R) を更新した。 (3) RL78/G14 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL) を追加した。
		40	5.4 ファイル構成のアプリケーションノート番号を変更した。

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）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、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、電氣的特性の範囲で、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。型名が違う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. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しており、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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

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



ルネサスエレクトロニクス株式会社

営業お問合せ窓口

<http://www.renesas.com>

営業お問合せ窓口の住所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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

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