

RX ファミリ、RL78 ファミリ、78K0R/Kx3-L

Macronix International 社製 MX25/66L family serial NOR Flash Memory 制御ソフトウェア

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 Macronix International Co., Ltd 社製 MX25/66L serial NOR Flash memory 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 NOR Flash memory 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。最新の制御ソフトウェアの情報は、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/QSPI シリアルフラッシュメモリ制御、QSPI シリアル相変化メモリ制御
[SPI/QSPI シリアルフラッシュメモリ・QSPI シリアル相変化メモリドライバ | Renesas](#)

動作確認デバイス

Serial NOR Flash memory Macronix International Co., Ltd 社製 MX25/66L family serial NOR Flash memory

: MX25R1635F (16Mbit)
 : MX25L3235E (32Mbit) 、MX25L3233F (32Mbit)
 : MX25L25635F (256Mbit) 、MX66L51235F (512Mbit)
 : MX25L51245G (512Mbit) 、MX66L1G45G (1Gbit)

動作確認に使用した MCU

RX100 シリーズ : RX111 (SCI を使用) 、RX111 (RSPI を使用)
 RX600 シリーズ : RX64M (RSPI を使用) 、RX64M (SCI を使用)
 RL78/G1x シリーズ : RL78/G14, RL78/G1C (SAU を使用)
 RL78/L1x シリーズ : RL78/L12, RL78/L13, RL78/L1C (SAU を使用)
 RL78/G2x シリーズ : RL78/G23 (SAU を使用)

上記以外の対象マイコンについては、「3.関連アプリケーションノート」を参照ください。

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、そのマイコンの仕様にあわせて変更し、十分評価してください。

なお、本アプリケーションノートでは、以下の略称を使用します。

Single-SPI (Single SPI mode による通信)

Dual-SPI (Dual SPI mode による通信)

Quad-SPI (Quad SPI mode による通信)

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	7
3. 関連アプリケーションノート	19
3.1 RX ファミリ 関連アプリケーションノート一覧	19
3.2 RL78 ファミリ、78K0R ファミリ 関連アプリケーションノート一覧	19
4. ハードウェア説明	20
4.1 ハードウェア構成例	20
4.1.1 Single-SPI 使用時の端子一覧	20
4.1.2 Single-SPI 使用時の接続例	20
4.1.3 Dual-SPI 使用時の端子一覧	21
4.1.4 Dual-SPI 使用時の接続例	21
4.1.5 Quad-SPI 使用時の端子一覧	22
4.1.6 Quad-SPI 使用時の接続例	22
5. ソフトウェア説明	23
5.1 動作概要	23
5.1.1 データバッファと送信／受信データの関係	23
5.1.2 制御可能なスレーブデバイス	24
5.1.3 Serial NOR Flash memory の CS#端子制御	26
5.1.4 Serial NOR Flash memory の命令コード	27
5.2 ソフトウェア構成	28
5.3 必要メモリサイズ	29
5.3.1 RX ファミリ	29
5.3.2 RL78 ファミリ、78K0R/Kx3-L	30
5.4 ファイル構成	36
5.5 定数一覧	37
5.5.1 戻り値	37
5.5.2 コマンド定義	37
5.5.3 各種定義	38
5.6 構造体/共用体一覧	41
5.7 変数一覧	42
5.8 関数一覧	42
5.9 関数仕様	43
5.9.1 ドライバ初期化処理	43
5.9.2 ステータスレジスタ読み出し処理	44
5.9.3 コンフィグレーションレジスタ読み出し処理	46
5.9.4 コンフィグレーションレジスタ書き込み処理	47
5.9.5 セキュリティレジスタ読み出し処理	49

5.9.6	ライトプロテクト設定処理	51
5.9.7	Quad モード許可設定処理	53
5.9.8	Quad モード禁止設定処理	55
5.9.9	WRDI コマンド発行処理	57
5.9.10	データ読み出し処理	58
5.9.11	データ書き込み処理	59
5.9.12	データ書き込み処理 (1Page 書き込み用)	61
5.9.13	消去処理	64
5.9.14	ID 読み出し処理	67
5.9.15	ビジーウェイト処理	68
5.9.16	4 バイトアドレスモード設定処理	70
6.	応用例.....	71
6.1	Serial NOR Flash memory 制御ソフトウェアの設定	72
6.1.1	r_qspi_flash_mx25l.h	72
6.1.2	r_qspi_flash_mx25l_sfr.h	74
6.1.3	r_qspi_flash_mx25l_sub.h.....	77
6.1.4	r_qspi_flash_mx25l_sub.c.....	80
6.1.5	r_qspi_flash_mx25l_drvif.c.....	81
6.1.6	r_qspi_flash_mx25l_sfr_rl78.c.....	85
7.	使用上の注意事項	86
7.1	組み込み時の注意事項.....	86
7.2	キャッシュ搭載の MCU を使用する場合	86
7.3	他容量に対応する場合.....	86
7.4	他スレーブデバイスを使用する場合	86
7.5	電源投入後の電圧安定待ち時間について.....	87
	改訂記録	89

1. 仕様

ルネサス エレクトロニクス製 MCU を使用し、Macronix International Co., Ltd.社製 MX25/66L serial NOR Flash memory 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 Macronix International Co., Ltd.社製 MX25/66L serial NOR Flash memory とした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、Single-SPI、Dual-SPI、Quad-SPI により制御します。
- ユーザ設定したシリアル通信機能 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 Serial NOR Flash memory 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。
- 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）によるスレーブデバイスとの通信 1 チャンネル（必須）
Port	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）

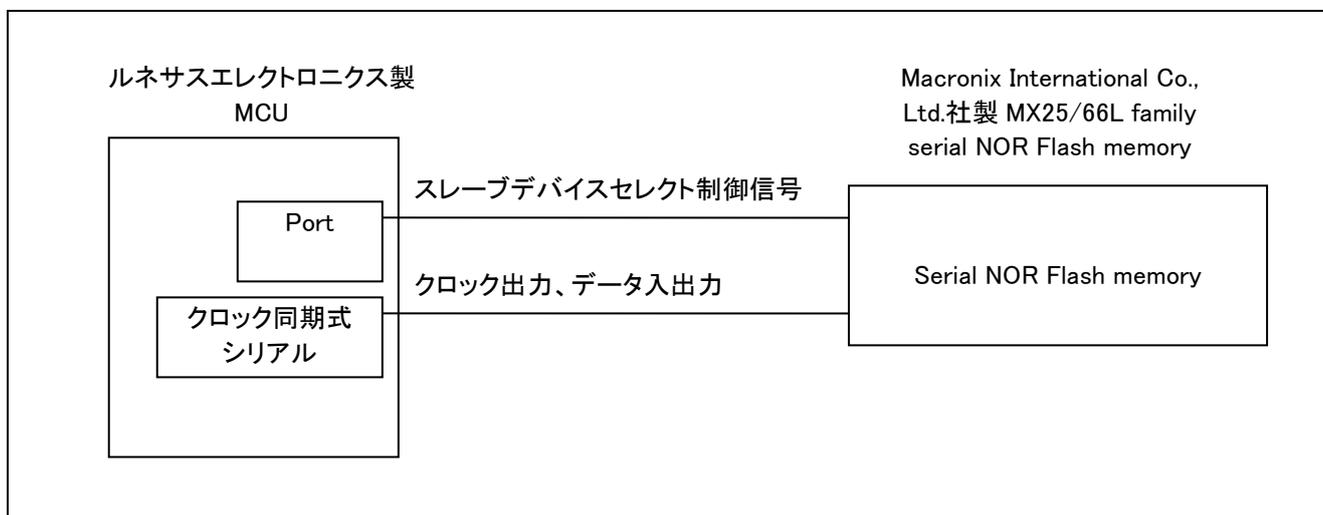


図 1-1 使用例

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、下記の条件で動作を確認しています。

2.1 RX ファミリ

(1) RX111 RSPI 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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、PCLKB : 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.2.21
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 Ver.2.04.R04
評価に使用した使用ボード	Renesas Starter Kit for RX111

(2) RX111 SCI 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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、PCLKB : 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.2.21
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 Ver.2.01.R05
評価に使用した使用ボード	Renesas Starter Kit for RX111

(3) RX64M RSPI 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RX64M グループ (プログラム ROM 4MB/RAM 512KB)
動作周波数	ICLK : 120MHz、PCLKA : 120MHz
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 e ² studio V3.1.3.06
C コンパイラ	ルネサス エレクトロニクス製 C/C++ compiler for RX family V.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に以下のオプションを追加 -lang = c99
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.2.21
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T, RX111, RX64M 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 Ver.2.05
評価に使用した使用ボード	Renesas Starter Kit for RX64M

(4) RX64M SCI 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RX64M グループ (プログラム ROM 4MB/RAM 512KB)
動作周波数	ICLK : 120MHz、PCLKA : 120MHz
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 e ² studio V3.1.3.06
C コンパイラ	ルネサス エレクトロニクス製 C/C++ compiler for RX family V.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に以下のオプションを追加 -lang = c99
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.2.21
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T, RX111, RX64M 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 Ver.2.02
評価に使用した使用ボード	Renesas Starter Kit for RX64M

2.2 RL78 ファミリ、78K0R/Kx3-L

(1) RL78/G14 SAU 統合開発環境 CubeSuite+の場合 (コンパイラ : CA78K0R)

表 2-5 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RL78/G14 (プログラム ROM 256MB/RAM 24KB)
動作周波数	メイン・システム・クロック : 24MHz CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.60 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。
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.2.21
評価に使用したソフトウェア	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 Ver.2.02
評価に使用した使用ボード	Renesas Starter Kit for RL78/G14

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

表 2-6 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RL78/G14 (プログラム ROM 256MB/RAM 24KB)
動作周波数	メイン・システム・クロック : 24MHz CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 V3.02.00
C コンパイラ	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/G14

(3) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 : 24MHz CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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	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。
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.2.21
評価に使用したソフトウェア	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 Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/G14

(4) RL78/G1C SAU 統合開発環境 CubeSuite+の場合 (コンパイラ : CA78K0R)

表 2-8 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 12MHz
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 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70 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。
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.2.21
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
評価に使用した使用ボード	Renesas RL78/G1C ターゲット・ボード QB-R5F10JGC-TB

(5) RL78/G1C SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 2-9 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 12MHz
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 V3.02.00
C コンパイラ	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。
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.2.21
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
評価に使用した使用ボード	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 2-10 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 12MHz
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	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。
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.2.21
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
評価に使用した使用ボード	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 (コンパイラ : CA78K0R)

表 2-11 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70 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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L12

(8) RL78/L12 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 2-12 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 V3.02.00
C コンパイラ	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L12

(9) RL78/L12 SAU 統合開発環境 IAR Embedded Workbench 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L12

(10) RL78/L13 SAU 統合開発環境 CubeSuite+の場合 (コンパイラ : CA78K0R)

表 2-14 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70 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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L13

(11) RL78/L13 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 2-15 動作確認条件

項目	内容
評価に使用したメモリ	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 V3.02.00
C コンパイラ	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L13

(12) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L13

(13) RL78/L1C SAU 統合開発環境 CubeSuite+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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 CubeSuite+ RL78,78K0R コンパイラ CA78K0R V1.70 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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L1C

(14) RL78/L1C 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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 V3.02.00
C コンパイラ	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L1C

(15) RL78/L1C 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	Macronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
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 CPU/周辺ハードウェア・クロック : 24MHz シリアル・クロック : 6MHz
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	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。
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.2.21
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
評価に使用した使用ボード	Renesas Starter Kit for RL78/L1C

(16) RL78/G23 SAU 統合開発環境 CS+ for CC (コンパイラ : CC-RL)

表 2-20 動作確認条件

項目	内容
評価に使用したメモリ	Micronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RL78/G23 (プログラム ROM 128KB / RAM 16KB)
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 : 8MHz
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 V8.05.00
C コンパイラ	ルネサス エレクトロニクス製 RL78 コンパイラ CC-RL V1.10.00 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。
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.2.21
評価に使用したソフトウェア	RL78/G14、RL78/G1C、RL78/L12、RL78/L13、RL78/L1C、RL78/G23 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)
評価に使用した使用ボード	RL78/G23-64p Fast Prototyping Board

(17) RL78/G23 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	Micronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RL78/G23 (プログラム ROM 128KB / RAM 16KB)
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 : 8MHz
動作電圧	3.3V
統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 (Ver.4.21.1.2409)
C コンパイラ	IAR Systems 製 IAR Assembler for Renesas RL78 (Ver.4.21.1.2409) IAR C/C++ Compiler for Renesas RL78 (Ver. 4.21.1.2409) 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。
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.2.21
評価に使用したソフトウェア	RL78/G14、RL78/G1C、RL78/L12、RL78/L13、RL78/L1C、RL78/G23 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)
評価に使用した使用ボード	RL78/G23-64p Fast Prototyping Board

(18) RL78/G23 SAU 統合開発環境 e²studio の場合 (コンパイラ : LLVM)

表 2-22 動作確認条件

項目	内容
評価に使用したメモリ	Micronix International Co., Ltd.社製 MX25/66L family serial NOR Flash memory
評価に使用したマイコン	RL78/G23 (プログラム ROM 128KB / RAM 16KB)
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 : 8MHz
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 e ² studio 22.4.0.R20220331-2313
C コンパイラ	オープンソースコンパイラ LLVM for Renesas RL78 10.0.0.202203 コンパイルオプション 統合開発環境のサイズ設定 ("Optimize size (-Os)") を使用しています。
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.2.22
評価に使用したソフトウェア	RL78/G14、RL78/G1C、RL78/L12、RL78/L13、RL78/L1C、RL78/G23 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)
評価に使用した使用ボード	RL78/G23-64p Fast Prototyping Board

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。

なお、以下の関連アプリケーションノートのうち、動作確認済の MCU については、表紙の“対象デバイス”をご参照ください。

3.1 RX ファミリ 関連アプリケーションノート一覧

- 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)
- 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, RX64M 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, RX64M 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)

3.2 RL78 ファミリ、78K0R ファミリ 関連アプリケーションノート一覧

- 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)
- RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C, RL78/G23 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 ハードウェア構成例

以下にハードウェア構成例を示します。

4.1.1 Single-SPI 使用時の端子一覧

表 4-1 に MCU 側の Single-SPI 使用端子と機能を示します。

表 4-1 Single-SPI 使用端子と機能

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

4.1.2 Single-SPI 使用時の接続例

以下に Single-SPI 使用時の接続例を示します。

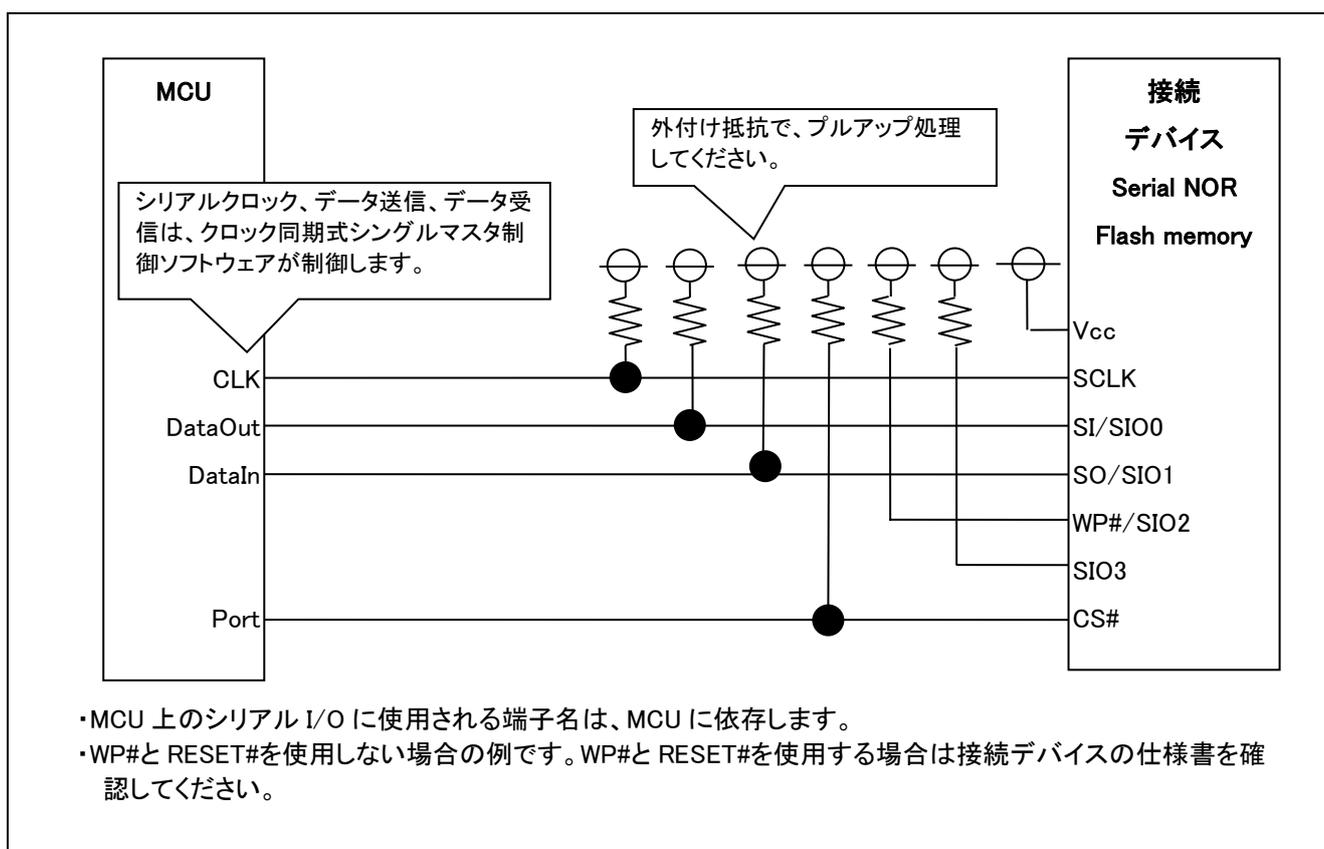


図 4-1 Single-SPI 使用時の MCU とスレーブデバイスの接続例

4.1.3 Dual-SPI 使用時の端子一覧

表 4-2 に MCU 側の Dual-SPI 使用端子と機能を示します。

なお、Dual-SPI を使用するためには、対象 MCU にクワッドシリアルペリフェラルインターフェース機能が搭載されている必要があります。

表 4-2 Dual-SPI 使用端子と機能

MCU 端子名	入出力	内容
CLK	出力	クロック出力
DataIn/Out0	入出力	マスターデータ入出力 0
DataIn/Out1	入出力	マスターデータ入出力 1
Port(CS#)	出力	スレーブデバイスセレクト出力

4.1.4 Dual-SPI 使用時の接続例

以下に Dual-SPI 使用時の接続例を示します。

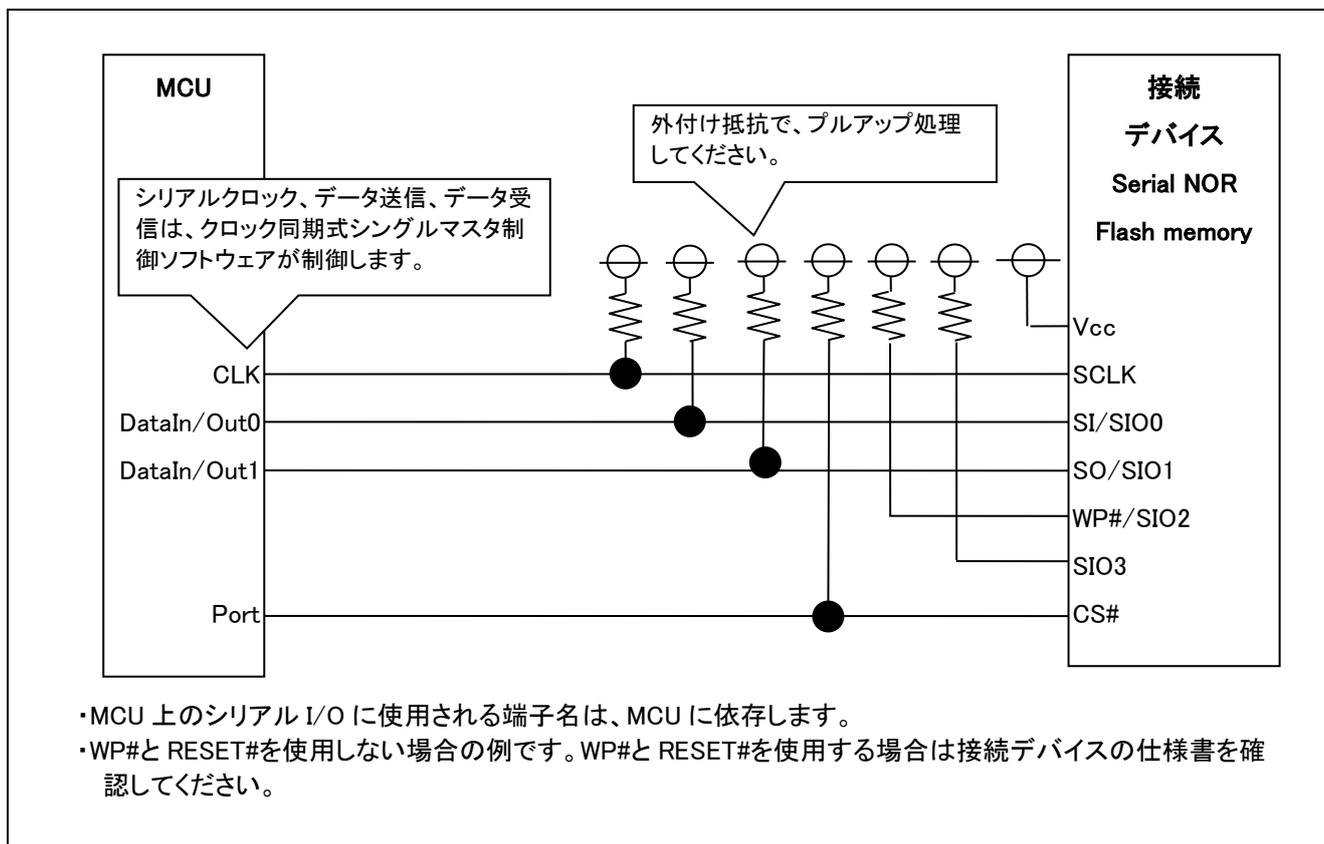


図 4-2 Dual-SPI 使用時の MCU とスレーブデバイスの接続例

5. ソフトウェア説明

5.1 動作概要

MCUのクロック同期式シリアル通信機能を使って、Serial NOR Flash memory 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。

- スレーブデバイスのCS#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。）

5.1.1 データバッファと送信／受信データの関係

本サンプルコードは、ブロック型デバイスドライバであり、送信／受信データポインタを引数として設定します。RAM上のデータバッファのデータ並びと送信／受信順番の関係は、以下のとおりで、エンディアンや使用するシリアル通信機能に関係なく、送信データバッファの並びの順に送信し、また、受信の順に受信データバッファに書き込みます。

図 5-1 に転送データの格納を示します。

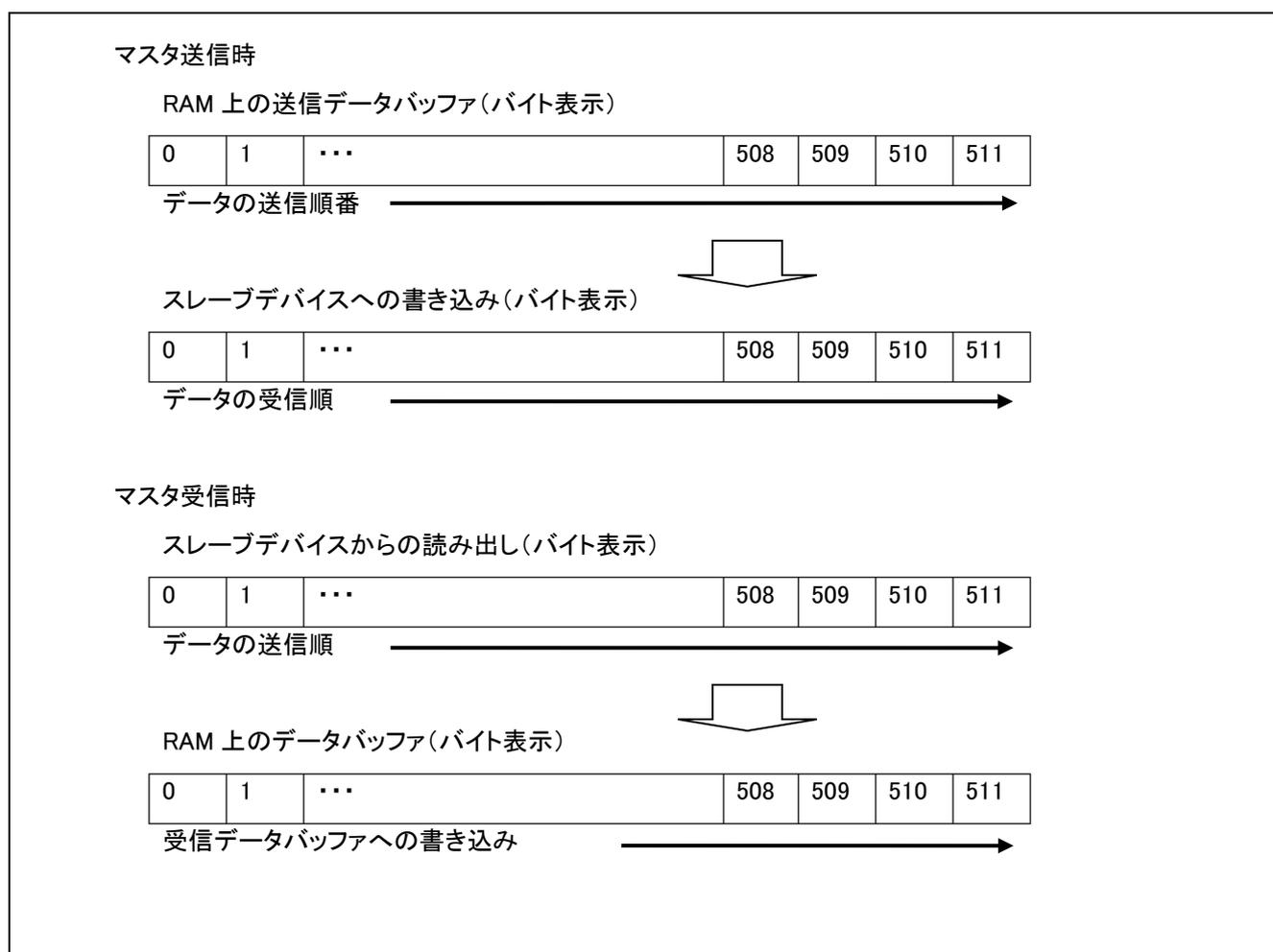


図 5-1 転送データの格納

5.1.2 制御可能なスレーブデバイス

制御可能なスレーブデバイスのタイミングを以下に示します。

(1) Single-SPI 使用時

図 5-2 に示す SPI モード 3 (CPOL=1、CPHA=1) のタイミングで制御します。

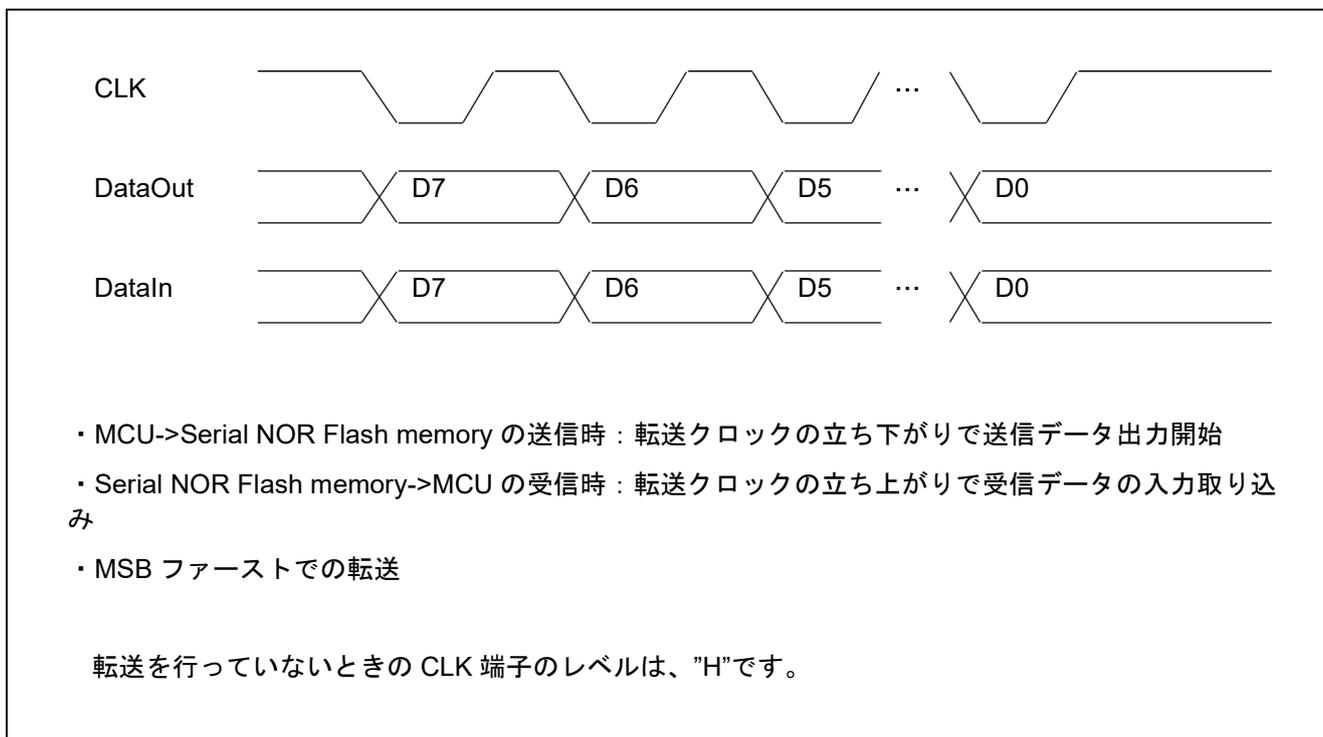


図 5-2 Single-SPI 時のクロック同期式モード タイミング設定

(2) Dual-SPI 使用時

図 5-3 に示す SPI モード 3 (CPOL=1、CPHA=1) のタイミングで制御します。

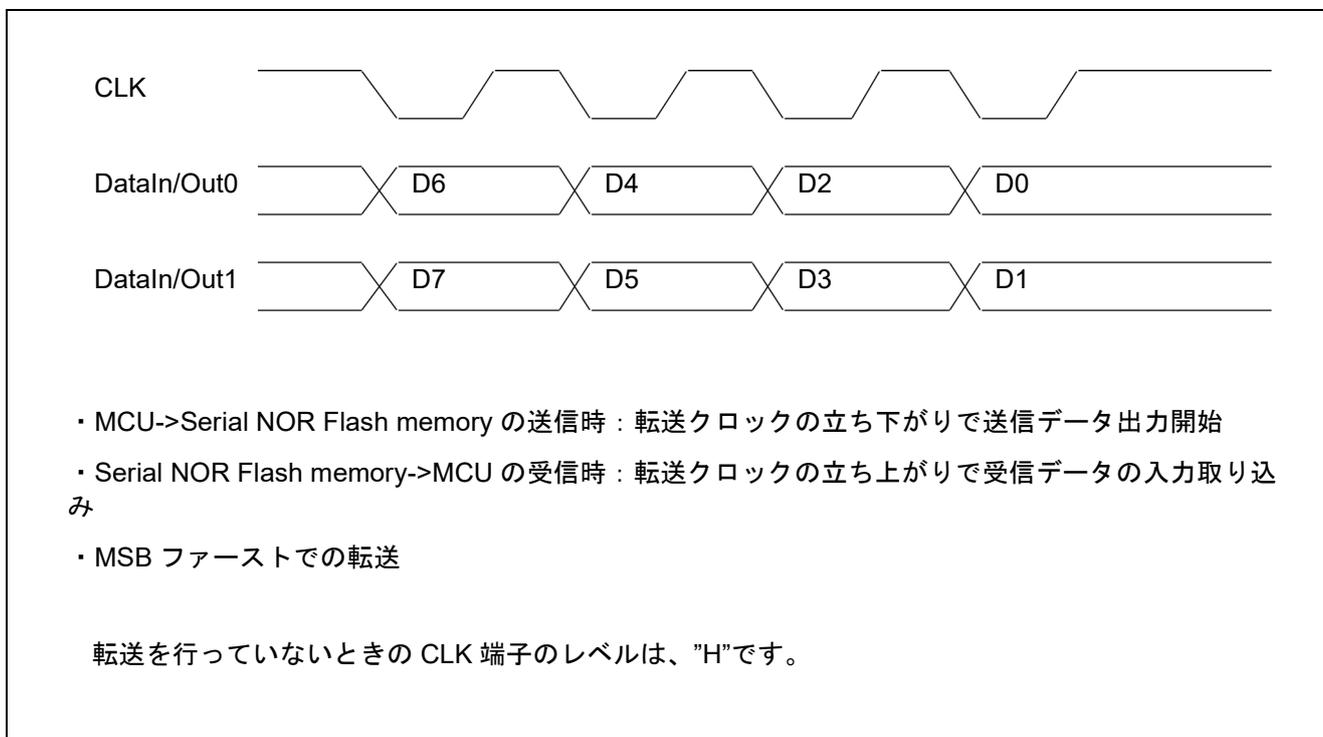


図 5-3 Dual-SPI 時のクロック同期式モード タイミング設定

(3) Quad-SPI 使用時

図 5-4 に示す SPI モード 3 (CPOL=1、CPHA=1) のタイミングで制御します。

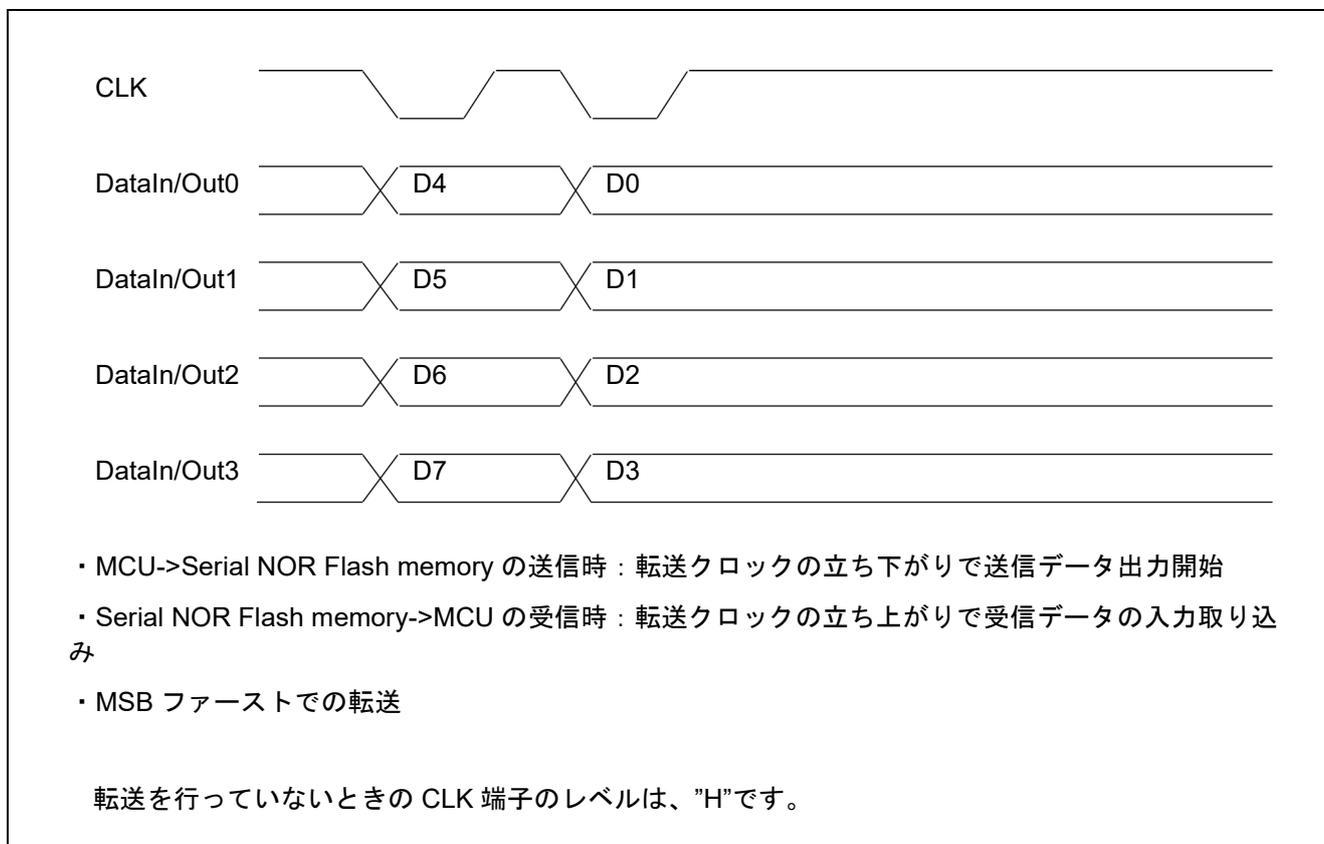


図 5-4 Quad-SPI 時のクロック同期式モード タイミング設定

5.1.3 Serial NOR Flash memory の CS#端子制御

Serial NOR Flash memory の CS#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 NOR Flash memory の CS# (MCU の Port(CS#)) 信号の立ち下がりから、Serial NOR Flash memory の C (MCU の CLK) 信号の立ち下がりまでの時間は、Serial NOR Flash memory の CS#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 NOR Flash memory の C (MCU の CLK) 信号の立ち上がりから、Serial NOR Flash memory の CS# (MCU の Port(CS#)) 信号の立ち上がりまでの時間は、Serial NOR Flash memory の CS#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 NOR Flash memory 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。

5.1.4 Serial NOR Flash memory の命令コード

Serial NOR Flash memory 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 (06 h)
WRDI	Write Disable	0000 0100 (04 h)
RDSR	Read Status Register	0000 0101 (05 h)
WRSR	Write Status Register	0000 0001 (01 h)
RDCR	Read Configuration Register	0001 0101 (15 h)
RDSCUR	Read Security Register	0010 1011 (2b h)
FAST READ	Read Data Bytes at Higher Speed	0000 1011 (0b h)
DREAD	Dual Output Read	0011 1011 (3b h)
QREAD	Quad Read	0110 1011 (6b h)
PP	Page Program	0000 0010 (02 h)
4PP	4 I/O Page Program	0011 1000 (38 h)
SE	Sector Erase (4KB)	0010 0000 (20 h)
BE32K	Block Erase (32KB)	0101 0010 (52 h)
BE	Block Erase (64KB)	1101 1000 (d8 h)
CE	Chip Erase	0110 0000 (60 h)
RDID	Read Identification	1001 1111 (9f h)
EN4B	Enter 4-byte Address Mode	1011 0111 (b7 h)
EXI4B	Exit 4-byte Address Mode	1110 1001 (e9 h)

5.2 ソフトウェア構成

本サンプルコードは、Serial NOR Flash memory 制御のための上位層に位置する制御ソフトウェア（図 5-5 の Serial NOR Flash memory 制御ソフトウェア）です。

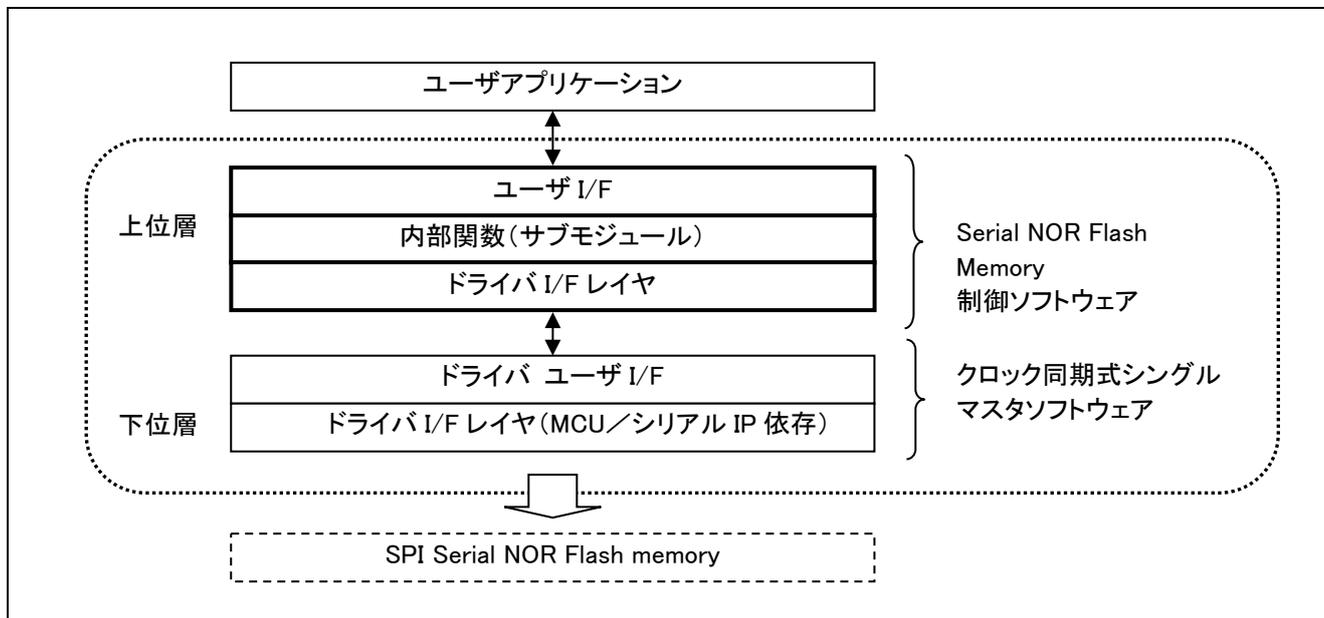


図 5-5 ソフトウェア構成

5.3 必要メモリサイズ

以下に必要メモリサイズを示します。

5.3.1 RX ファミリ

(1) RX111 RSPI の場合

表 5-2 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	4,255 バイト (リトルエンディアン)	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c
最大使用ユーザスタック	164 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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

(2) RX64M RSPI の場合

表 5-3 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	4,251 バイト (リトルエンディアン)	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

5.3.2 RL78 ファミリ、78K0R/Kx3-L

(1) RL78/G14 SAU の場合統合開発環境 CubeSuite+の場合 (コンパイラ : CA78K0R)

表 5-4 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	7,119 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g14.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g14.c
最大使用ユーザスタック	102 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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

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

表 5-5 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,672 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g14.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g14.c
最大使用ユーザスタック	82 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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

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

表 5-6 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	6,620 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g14.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g14.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。

ROM/RAM サイズは下位層の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メモリサイズを含みません。

使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

(4) RL78/L13 SAU 統合開発環境 CubeSuite+の場合 (コンパイラ : CA78K0R)

表 5-7 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	7,218 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78l13.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78l13.c
最大使用ユーザスタック	102 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。

ROM/RAM サイズは下層の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メモリサイズを含みません。

使用する MCU により、上記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、下層のクロック同期式シングルマスタソフトウェアのスタックサイズも含まれます。

(5) RL78/L13 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 5-8 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,676 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78l13.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78l13.c
最大使用ユーザスタック	82 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。

ROM/RAM サイズは下層の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メモリサイズを含みません。

使用する MCU により、上記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、下層のクロック同期式シングルマスタソフトウェアのスタックサイズも含まれます。

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

表 5-9 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,532 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78l13.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78l13.c
最大使用ユーザスタック	126 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。

ROM/RAM サイズは下位層の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メモリサイズを含みません。

使用する MCU により、上記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。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ソフトウェアのスタックサイズも含まれません。

(7) RL78/G23 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 5-10 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,386 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g23.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g23.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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

(8) RL78/G23 SAU 統合開発環境 IAR Embedded Workbench の場合

表 5-11 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	7,527 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g23.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g23.c
最大使用ユーザスタック	126 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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

(9) RL78/G23 SAU 統合開発環境 e²studio の場合 (コンパイラ : LLVM)

表 5-12 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	4,724 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g23.c
RAM	6 バイト	r_qspi_flash_mx25l_usr.c r_qspi_flash_mx25l_sub.c r_qspi_flash_mx25l_drvif.c r_qspi_flash_mx25l_sfr_rl78g23.c
最大使用ユーザスタック	78 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。
ROM/RAM サイズは下位層の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メモリサイズを含みません。
使用する MCU により、上記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ソフトウェアのスタックサイズも含まれます。

5.4 ファイル構成

表 5-12 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を除きます。

表 5-13 ファイル構成

¥r01an1967xx0104-mcu-serial	<DIR>	サンプルコードのフォルダ
r01an1967jj0104-mcu-serial.pdf		アプリケーションノート（本書）
r01an1967ej0104-mcu-serial.pdf		アプリケーションノート英語版
¥source	<DIR>	プログラム格納用フォルダ
¥r_qspi_flash_mx25l	<DIR>	Serial NOR Flash memory 制御ソフトウェア用フォルダ
r_qspi_flash_mx25l.h		ヘッダファイル
r_qspi_flash_mx25l_drvif.c		ドライバ I/F ソースファイル
r_qspi_flash_mx25l_drvif.h		ドライバ I/F ヘッダファイル
r_qspi_flash_mx25l_sfr.h.rl78g1c		レジスタ用共通定義（RL78/G1C 用）
r_qspi_flash_mx25l_sfr.h.rl78g14		レジスタ用共通定義（RL78/G14 用）
r_qspi_flash_mx25l_sfr.h.rl78l1c		レジスタ用共通定義（RL78/L1C 用）
r_qspi_flash_mx25l_sfr.h.rl78l12		レジスタ用共通定義（RL78/L12 用）
r_qspi_flash_mx25l_sfr.h.rl78l13		レジスタ用共通定義（RL78/L13 用）
r_qspi_flash_mx25l_sfr.h.rx63n		レジスタ用共通定義（RX63N 用）
r_qspi_flash_mx25l_sfr.h.rl78g23		レジスタ用共通定義（RL78/G23 用）
r_qspi_flash_mx25l_sfr.h.rx64m		レジスタ用共通定義（RX64M 用）
r_qspi_flash_mx25l_sfr.h.rx111		レジスタ用共通定義（RX111 用）
r_qspi_flash_mx25l_sfr_rl78g1c.c		レジスタ用共通定義ソースファイル（RL78/G1C 用）
r_qspi_flash_mx25l_sfr_rl78g14.c		レジスタ用共通定義ソースファイル（RL78/G14 用）
r_qspi_flash_mx25l_sfr_rl78l1c.c		レジスタ用共通定義ソースファイル（RL78/L1C 用）
r_qspi_flash_mx25l_sfr_rl78l12.c		レジスタ用共通定義ソースファイル（RL78/L12 用）
r_qspi_flash_mx25l_sfr_rl78l13.c		レジスタ用共通定義ソースファイル（RL78/L13 用）
r_qspi_flash_mx25l_sfr_rl78g23.c		レジスタ用共通定義ソースファイル（RL78/G23 用）
r_qspi_flash_mx25l_sub.c		内部関数ソースファイル
r_qspi_flash_mx25l_sub.h		内部関数ヘッダファイル
r_qspi_flash_mx25l_usr.c		ユーザ I/F ソースファイル
¥sample	<DIR>	動作確認プログラム格納用フォルダ
testmain.c		動作確認用のサンプルソースファイル

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 5-13 にサンプルコードで使用する戻り値を示します。

表 5-14 戻り値 (r_qspi_flash_mx25l.h 参照)

定数名	設定値	内容
FLASH_OK	(error_t)(0)	Successful operation
FLASH_ERR_PARAM	(error_t)(-1)	Parameter error
FLASH_ERR_HARD	(error_t)(-2)	Hardware error
FLASH_ERR_WP	(error_t)(-4)	Write-protection error
FLASH_ERR_TIMEOUT	(error_t)(-6)	Timeout error
FLASH_ERR_OTHER	(error_t)(-7)	Other error

5.5.2 コマンド定義

表 5-14 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 5-15 コマンド定義 (r_qspi_flash_mx25l_sub.c 参照)

定数名	設定値	内容
FLASH_CMD_WREN	(uint8_t)(0x06)	Write Enable
FLASH_CMD_WRDI	(uint8_t)(0x04)	Write Disable
FLASH_CMD_RDSR	(uint8_t)(0x05)	Read Status Register
FLASH_CMD_WRSR	(uint8_t)(0x01)	Write Status Register
FLASH_CMD_RDCCR	(uint8_t)(0x15)	Read Configuration Register
FLASH_CMD_RDSCUR	(uint8_t)(0x2b)	Read Security Register
FLASH_CMD_FREAD	(uint8_t)(0x0b)	Read Data at Higher Speed
FLASH_CMD_DREAD	(uint8_t)(0x3b)	Dual Read (Single -> Dual Output)
FLASH_CMD_QREAD	(uint8_t)(0x6b)	Quad Read (Single -> Quad Output)
FLASH_CMD_PP	(uint8_t)(0x02)	Page Program (Single -> Single Input)
FLASH_CMD_4PP	(uint8_t)(0x38)	Quad Page Program (Single -> Quad Input)
FLASH_CMD_SE	(uint8_t)(0x20)	Sector Erase (4KB)
FLASH_CMD_BE32K	(uint8_t)(0x52)	Block Erase (32KB)
FLASH_CMD_BE64K	(uint8_t)(0xd8)	Block Erase (64KB)
FLASH_CMD_CE	(uint8_t)(0x60)	Chip Erase
FLASH_CMD_RDID	(uint8_t)(0x9f)	Read Identification
FLASH_CMD_EN4B	(uint8_t)(0xb7)	Enter 4-byte Address Mode
FLASH_CMD_EX4B	(uint8_t)(0xe9)	Exit 4-byte Address Mode

5.5.3 各種定義

表 5-15 から表 5-19 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 5-16 r_qspi_flash_mx25l.h の各種定義値

定数名	設定値	内容
FLASH_DEV_NUM	(1)	接続するデバイスの数
FLASH_DEV0	(0)	デバイス番号 0
FLASH_DEV1	(1)	デバイス番号 1
FLASH_DELAY_TASK	(uint8_t)(1)	ディレイタスクのウェイト時間[単位 : ms](※ 1)
FLASH_LOG_ERR	(1)	Log Type : Error
FLASH_TRUE	(uint8_t)(0x01)	Flag "ON"
FLASH_FALSE	(uint8_t)(0x00)	Flag "OFF"
FLASH_MODE_S_ERASE	(uint8_t)(1)	Erase Mode : Sector Erase (4KB)
FLASH_MODE_BE32K_ERASE	(uint8_t)(2)	Erase Mode : Block Erase (32KB)
FLASH_MODE_BE64K_ERASE	(uint8_t)(3)	Erase Mode : Block Erase (64KB)
FLASH_MODE_C_ERASE	(uint8_t)(4)	Erase Mode : Chip Erase
FLASH_MODE_3BYTE	(uint8_t)(0)	Address Mode : 3-byte Address Mode
FLASH_MODE_4BYTE	(uint8_t)(1)	Address Mode : 4-byte Address Mode
FLASH_MODE_REG_WRITE	(uint8_t)(0)	Wait Mode : Register write mode
FLASH_MODE_PROG_ERASE	(uint8_t)(1)	Wait Mode : Page Program or Erase mode
FLASH_MEM_SIZE	(uint32_t)(33554432)	メモリサイズ (バイト単位) 左記は、256Mbit の場合の値です。
FLASH_SECT_ADDR	(uint32_t)(0xfffff000)	セクタ消去時のセクタアドレスマスク値
FLASH_BE32K_ADDR	(uint32_t)(0xffff8000)	セクタ消去時のセクタアドレスマスク値
FLASH_BE64K_ADDR	(uint32_t)(0xffff0000)	セクタ消去時のセクタアドレスマスク値
FLASH_PAGE_SIZE	(uint32_t)(256)	ページサイズ (バイト単位)
FLASH_ADDR_SIZE	(uint8_t)(4)	アドレスサイズ (バイト単位) 左記は、256Mbit 以上の場合の値です。
FLASH_WP_WHOLE_MEM	(uint8_t)(0x0f)	チップ全体のライトプロテクト
FLASH_FULL_CHIP_ERASE	FLASH_MODE_C_ERASE	サポートする全消去コマンド
FLASH_ADDR_MODE	FLASH_MODE_4BYTE	Addressability Mode 左記は、256Mbit 以上の場合の値です。
FLASH_CMD_SIZE	(uint8_t)1	コマンドサイズ (バイト単位)
FLASH_STSREG_SIZE	(uint16_t)1	ステータスレジスタサイズ (バイト単位)
FLASH_CFGREG_SIZE	(uint16_t)1	コンフィギュレーションレジスタサイズ (バイト単位)
FLASH_SCURREG_SIZE	(uint16_t)1	セキュリティレジスタサイズ (バイト単位)
FLASH_WSTSREG_SIZE	(uint16_t)2	書き込み用ステータスレジスタサイズ (バイト単位)
FLASH_IDDATA_SIZE	(uint16_t)3	ID Data サイズ (バイト単位)

※ 1 : OS 制御時のディレイタスクです。本サンプルコードの OS 制御は、μITRON4.0 を想定しています。

表 5-17 r_qspi_flash_mx25l_sfr.h.rx111 の各種定義値

定数名	設定値	内容
FLASH_DR_CS0	PORT5.PODR.BIT.B5	デバイス番号 0 ポート出力データレジスタ SFR 定義
FLASH_DDR_CS0	PORT5.PDR.BIT.B5	デバイス番号 0 ポート方向レジスタ SFR 定義
FLASH_DR_CS1	-	デバイス番号 1 ポート出力データレジスタ SFR 定義 (2 デバイス制御する場合は設定してください。)
FLASH_DDR_CS1	-	デバイス番号 1 ポート方向レジスタ SFR 定義 (2 デバイス制御する場合は設定してください。)
FLASH_HI	(uint8_t)(0x01)	Port "H"
FLASH_LOW	(uint8_t)(0x00)	Port "L"
FLASH_OUT	(uint8_t)(0x01)	Port Output Setting
FLASH_IN	(uint8_t)(0x00)	Port Input Setting
FLASH_BR	(uint8_t)(0x01)	コマンド送信時の転送レート ※ 1
FLASH_BR_WRITE_DATA	(uint8_t)(0x01)	データ送信時の転送レート ※ 1
FLASH_BR_READ_DATA	(uint8_t)(0x01)	データ受信時の転送レート ※ 1

※ 1 : 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を使用した場合、RSPi ビットレートレジスタ SPBR に設定される値です。周辺モジュールクロックを 32[MHz]とし、転送レートを 16[MHz]に設定する場合の値です。

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を使用した場合、ビットレートレジスタ BRR に設定される値です。周辺モジュールクロックを 32[MHz]とし、転送レートを 4[MHz]に設定する場合の値です。

表 5-18 r_qspi_flash_mx25l_sfr.h.rl78 の各種定義値

定数名	設定値	内容
FLASH_DR_CS0	P8.0	デバイス番号 0 ポート・レジスタ SFR 定義
FLASH_DDR_CS0	PM8.0	デバイス番号 0 ポート・モード・レジスタ SFR 定義
FLASH_DR_CS1	-	デバイス番号 1 ポート・レジスタ SFR 定義 (2 デバイス制御する場合は設定してください。)
FLASH_DDR_CS1	-	デバイス番号 1 ポート・モード・レジスタ SFR 定義 (2 デバイス制御する場合は設定してください。)
FLASH_HI	(uint8_t)(0x01)	Port "H"
FLASH_LOW	(uint8_t)(0x00)	Port "L"
FLASH_OUT	(uint8_t)(0x00)	Port Output Setting
FLASH_IN	(uint8_t)(0x01)	Port Input Setting
FLASH_BR	(uint8_t)(0x01)	コマンド送信時の転送レート ※ 1
FLASH_BR_WRITE_DATA	(uint8_t)(0x01)	データ送信時の転送レート ※ 1
FLASH_BR_READ_DATA	(uint8_t)(0x01)	データ受信時の転送レート ※ 1

※ 1 : 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を使用した場合、シリアル・データ・レジスタ SDR のビット 15-9 に設定される値です。本サンプルコードでは、動作クロックを 24[MHz]とし、転送レートを 6[MHz]に設定する場合の値です。

表 5-19 r_qspi_flash_mx25l_sub.c の各種定義値

定数名	設定値	内容
FLASH_SHORT_SIZE	(uint32_t)(0x00008000)	下位関数での最大転送サイズの設定 (最大 : 32KBytes)

表 5-20 r_qspi_flash_mx25l_sub.h の各種定義値

定数名	設定値	内容
FLASH_SE_BUSY_WAIT	(uint32_t)(200)	Sector Erase (4KB) Busy Timeout 200 × 1ms = 200ms
FLASH_BE32K_BUSY_WAIT	(uint32_t)(1000)	Block Erase (32KB) Busy Timeout 1000 × 1ms = 1s
FLASH_BE64K_BUSY_WAIT	(uint32_t)(2000)	Block Erase (64KB) Busy Timeout 2000 × 1ms = 1s
FLASH_CE_BUSY_WAIT	(uint32_t)(1200000)	Chip Erase Busy Timeout 1200000 × 1ms = 1200s
FLASH_PP_BUSY_WAIT	(uint32_t)(3000)	Page Program Timeout 3000 × 1us = 3ms
FLASH_WR_BUSY_WAIT	(uint32_t)(40000)	Write Register Timeout 40000 × 1us = 40ms
FLASH_T_WBUSY_WAIT	(uint16_t)MTL_T_1US	Write Busy Polling Time
FLASH_T_PBUSY_WAIT	(uint16_t)MTL_T_1MS	Page Program Busy Polling Time
FLASH_T_EBUSY_WAIT	(uint16_t)MTL_T_1MS	Erase Busy Polling Time
FLASH_T_CS_HOLD	(uint16_t)MTL_T_1US	CS Stability Waiting Time
FLASH_T_R_ACCESS	(uint16_t)MTL_T_1US	Reading Start Waiting Time
FLASH_REG_SRWD	(uint8_t)(0x80)	Status Register Write Disable
FLASH_REG_QE	(uint8_t)(0x40)	Quad Enable Bit
FLASH_REG_BP3	(uint8_t)(0x20)	Block Protection Bit3
FLASH_REG_BP2	(uint8_t)(0x10)	Block Protection Bit2
FLASH_REG_BP1	(uint8_t)(0x08)	Block Protection Bit1
FLASH_REG_BP0	(uint8_t)(0x04)	Block Protection Bit0
FLASH_REG_WEL	(uint8_t)(0x02)	Write Enable Latch Bit
FLASH_REG_WIP	(uint8_t)(0x01)	Write In Progress Bit
FLASH_REG_MASK	(uint8_t)(0xfc)	Write status fixed data
FLASH_CNFG_DC1	(uint8_t)(0x80)	Dummy Cycle 1
FLASH_CNFG_DC0	(uint8_t)(0x40)	Dummy Cycle 0
FLASH_CNFG_4BYTE	(uint8_t)(0x20)	4-byte Address Mode
FLASH_CNFG_RSV	(uint8_t)(0x10)	Reserved
FLASH_CNFG_TB	(uint8_t)(0x08)	Top/Bottom
FLASH_CNFG_ODS2	(uint8_t)(0x04)	Output Driver Strength 2
FLASH_CNFG_ODS1	(uint8_t)(0x02)	Output Driver Strength 1
FLASH_CNFG_ODS0	(uint8_t)(0x01)	Output Driver Strength 0
FLASH_SCUR_E_FAIL	(uint8_t)(0x40)	Erase Failed
FLASH_SCUR_P_FAIL	(uint8_t)(0x20)	Program Failed

5.6 構造体/共用体一覧

図 5-6、図 5-7 にサンプルコードで使用する構造体/共用体を示します。

```
typedef union {
    uint32_t    ul;
    uint8_t    uc[4];
} flash_exchg_long_t;                /* total 4bytes */
```

図 5-6 サンプルコードで使用する共用体 (r_qspi_flash_mx25l_sub.c 参照)

```
typedef struct
{
    uint32_t    Addr;                /* Address to issue a command */
    uint32_t    Cnt;                /* Number of bytes to be read/written */
    uint16_t    DataCnt;            /* Temporary counter or Number of bytes to be written in a page */
    uint8_t    rsv[2];              /* Reserved */
    uint8_t FAR* pData;            /* Data storage buffer pointer */
} r_qspi_flash_info_t;
```

図 5-7 サンプルコードで使用する構造体 (r_qspi_flash_mx25l.h 参照)

表 5-21 構造体 “r_qspi_flash_info_t” の説明

構造体メンバ	設定可能範囲	説明
Addr	0000 0000h～ FFFF FFFFh	書き込み/読み出しを開始するアドレス
Cnt	0000 0000h～ FFFF FFFFh	書き込み/読み出しのデータカウンタ (バイト単位)
DataCnt	(設定禁止)	書き込み時：書き込みデータカウンタ Temp. (最大 1Page) 読み出し時：読み出しデータカウンタ Temp. (最大 32KBytes)
rsv[2]	(設定無効)	アライメント調整用
pData	—	データ格納バッファポインタ 書き込み時：Serial NOR Flash memory へ書き込むデータの格納元 読み出し時：Serial NOR Flash memory から読み出すデータの格納先

5.7 変数一覧

表 5-19 に static 型変数を示します。

表 5-22 static 型変数 (r_qspi_flash_mx25l_sub.c 参照)

型	変数名	内容	使用関数
STATIC uint8_t	g_flash_cmdbuf[6]	コマンド用バッファ	r_qspi_flash_send_cmd r_qspi_flash_set_cmd

5.8 関数一覧

表 5-20 に関数を示します。

表 5-23 関数

関数名	概要
R_QSPI_FLASH_Init_Driver()	ドライバ初期化処理
R_QSPI_FLASH_Read_Status()	ステータスレジスタ読み出し処理
R_QSPI_FLASH_Read_Configuration()	コンフィグレーションレジスタ読み出し処理
R_QSPI_FLASH_Write_Configuration()	コンフィグレーションレジスタ書き込み処理
R_QSPI_FLASH_Read_Security()	セキュリティレジスタ読み出し処理
R_QSPI_FLASH_Set_Write_Protect()	ライトプロテクト設定処理
R_QSPI_FLASH_Quad_Enable()	Quad モード許可設定処理
R_QSPI_FLASH_Quad_Disable()	Quad モード禁止設定処理
R_QSPI_FLASH_Write_Di()	WRDI コマンド発行処理
R_QSPI_FLASH_Read_Data()	データ読み出し処理
R_QSPI_FLASH_Write_Data()	データ書き込み処理
R_QSPI_FLASH_Write_Data_Page()	データ書き込み処理 (1Page 書き込み用)
R_QSPI_FLASH_Erase()	消去処理
R_QSPI_FLASH_Read_ID()	ID 読み出し処理
R_QSPI_FLASH_Wait()	ビジーウェイト処理
R_QSPI_FLASH_Set_4byte_Address_Mode()	4 バイトアドレスモード設定処理

【注】 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 関数仕様

サンプルコードの関数仕様を示します。

5.9.1 ドライバ初期化処理

R_QSPI_FLASH_Init_Driver	
概要	ドライバ初期化処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Init_Driver(void)
説明	<ul style="list-style-type: none"> ・ r_qspi_flash_init_port()関数をコールし、CS#端子の初期化を行います。 ・ クロック同期式シングルマスタ制御ソフトウェアの初期化関数をコールし、I/Oポートの初期化を行います。
引数	なし
リターン値	<ul style="list-style-type: none"> ・ 初期化結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_OTHER ; Other error r_qspi_flash_drvif_init_driver()の戻り値を返します。

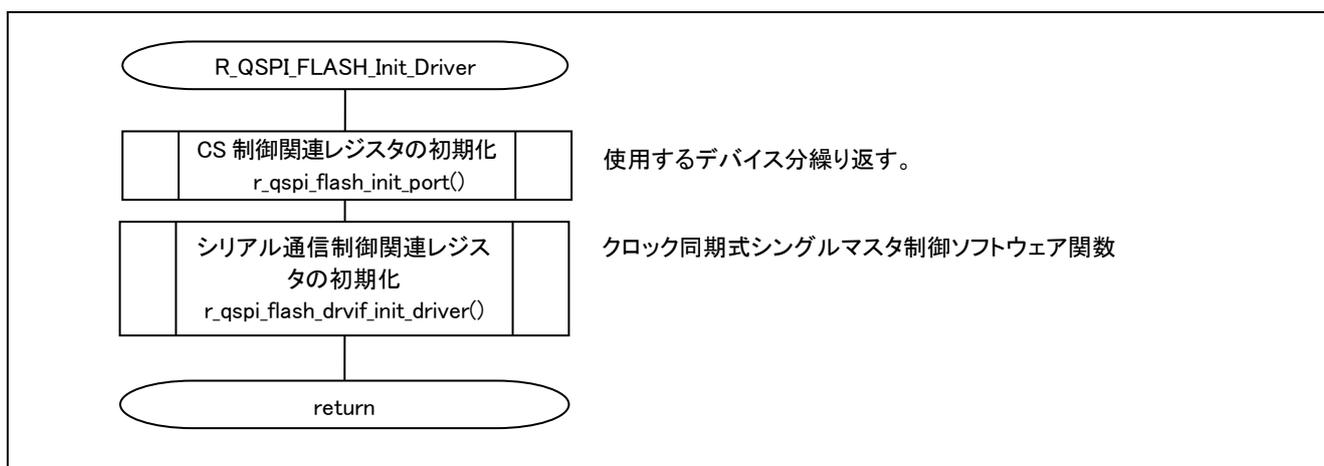


図 5-8 ドライバ初期化処理概要

5.9.2 ステータスレジスタ読み出し処理

R_QSPI_FLASH_Read_Status	
概要	ステータスレジスタ読み出し処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_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。 ・読み出し用バッファ(pStatus)には下記情報が格納されます。 <ul style="list-style-type: none"> Bit 7: Status register write enable/disable (SRWD) <ul style="list-style-type: none"> 1: Disable writing to the status register. 0: Enable writing to the status register. Bit 6: Quad Enable (QE) <ul style="list-style-type: none"> 1: Quad Enable (Performs Quad I/O mode and WP#, RESET# are disabled.) 0: Not Quad Enable (Performs non-Quad I/O mode and WP#, RESET# are enabled.) Bits 5 to 2: Block protect 3 - 0 (BP3 - BP0) <ul style="list-style-type: none"> Set to 1, a designated memory area is protected from PROGRAM and ERASE operations. Bit 1: Write enable latch (WEL) <ul style="list-style-type: none"> 1: Internal Write Enable Latch is set. 0: Internal Write Enable Latch is reset. Bit 0: Write in progress (WIP) <ul style="list-style-type: none"> 1: Program or Erase cycle is in progress. 0: No Program or No Erase cycle is in progress. ・プロテクト領域とプロテクトビットの関係は、使用する Serial NOR Flash memory のデータシートを参照してください。
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pStatus ; 読み出し用バッファポインタ
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。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

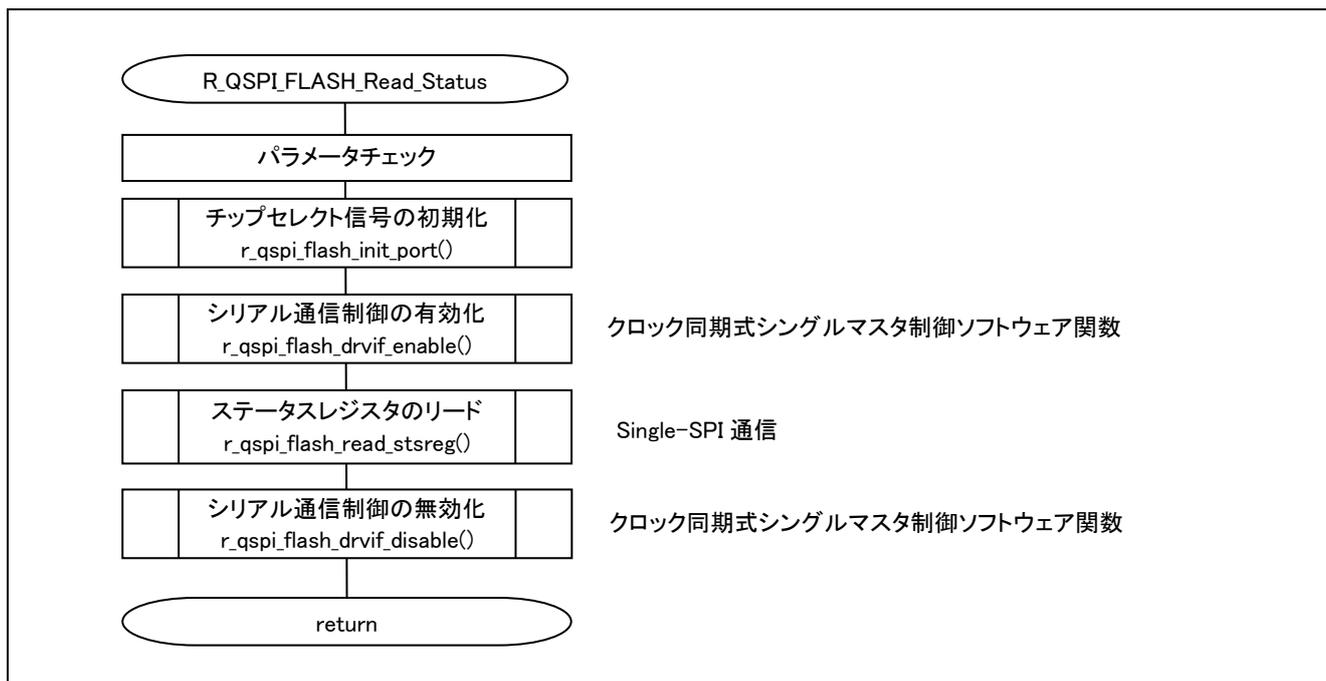


図 5-9 ステータスレジスタ読み出し処理概要

5.9.3 コンフィグレーションレジスタ読み出し処理

R_QSPI_FLASH_Read_Configuration	
概要	コンフィグレーションレジスタ読み出し処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Read_Configuration(uint8_t DevNo, uint8_t FAR* pConfig)
説明	<ul style="list-style-type: none"> ・コンフィグレーションレジスタを読み出し、pConfig 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。 ・読み出し用バッファ(pConfig)には下記情報が格納されます。 <ul style="list-style-type: none"> Bits 7 to 6: DC1-DC0 (Dummy cycle) See the specification of the Flash memory. Bit 5: 4BYTE (4BYTE Indicator) <ul style="list-style-type: none"> 1: 4-byte address mode 0: 3-byte address mode Bit 4: Reserved Bit 3: TB (Top/Bottom) <ul style="list-style-type: none"> 1: Bottom area protect 0: Top area protect Bits 2 to 0: ODS2-ODS0 (Output driver strength) See the specification of the Flash memory.
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pConfig ; 読み出し用バッファポインタ
リターン値	<ul style="list-style-type: none"> ・コンフィグレーションレジスタ取得結果を返します。 <ul style="list-style-type: none"> FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

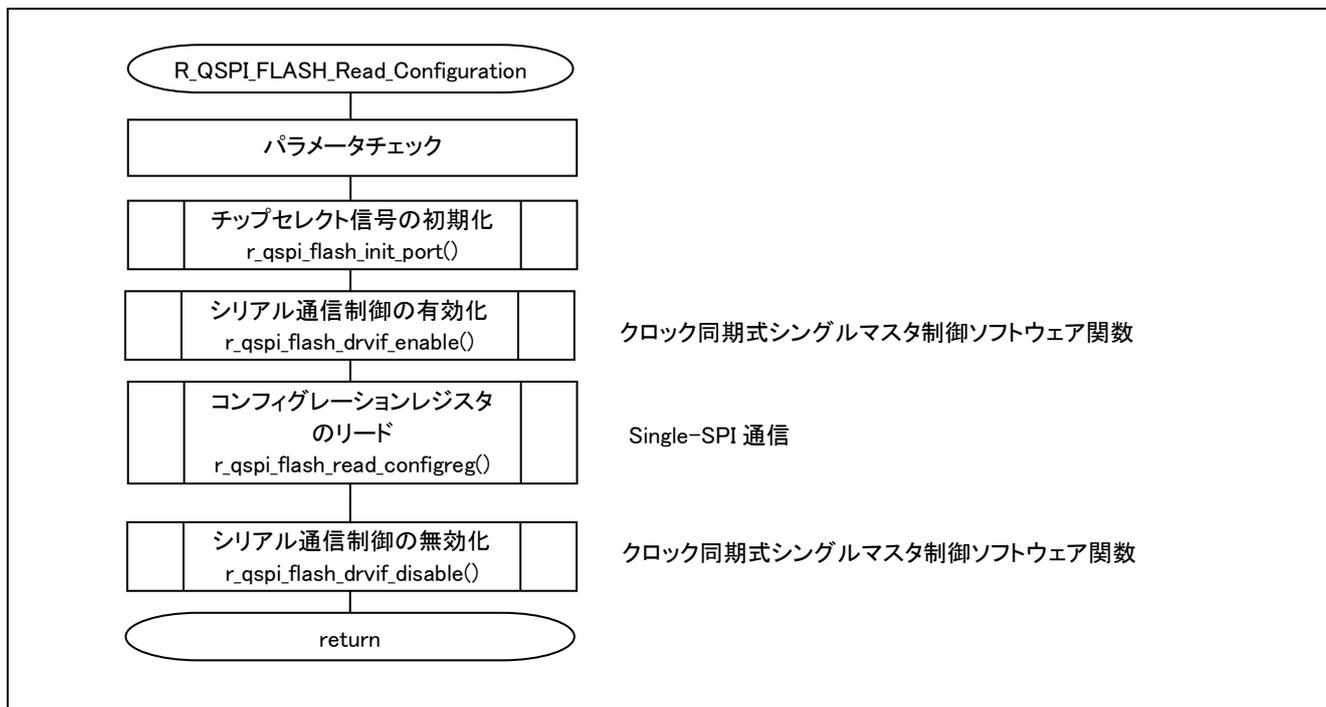


図 5-10 コンフィグレーションレジスタ読み出し処理概要

5.9.4 コンフィグレーションレジスタ書き込み処理

R_QSPI_FLASH_Write_Configuration	
概要	コンフィグレーションレジスタ書き込み処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Write_Configuration(uint8_t DevNo, uint8_t FAR* pConfig)
説明	<ul style="list-style-type: none"> ・ pConfig の値をコンフィグレーション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。 ・ 書き込み用バッファ(pConfig)には下記情報を格納してください。なお、この情報は使用デバイスにより異なります。製品のデータシートをご参照ください。 <ul style="list-style-type: none"> Bits 7 to 6: DC1-DC0 (Dummy cycle) See the specification of the Flash memory. Bit 5: 4BYTE (4BYTE Indicator) <ul style="list-style-type: none"> 1: 4-byte address mode 0: 3-byte address mode Bit 4: PBE (Preamble bit Enable) <ul style="list-style-type: none"> 0: Disable 1: Enable Bit 3: TB (Top/Bottom) <ul style="list-style-type: none"> 1: Bottom area protect 0: Top area protect Bits 2 to 0: ODS2-ODS0 (Output driver strength) See the specification of the Flash memory ・ 書き込み用バッファに値を設定する場合、事前にコンフィグレーションレジスタの値を読み出し、書き換えたい値以外は変更しないでください。 ・ 処理終了後、コンフィグレーションレジスタを読み出して書き込み値を確認してください。 ・ 4BYTE bit は読み出し専用のため、書き込みはできません。設定は無視されます。この bit は、R_QSPI_FLASH_Set_4byte_Address_Mode()を使って変更できます。 ・ 書き込み完了待ちの方法は二種類あります。以降にその方法を示します。なお、次の処理（書き込み／読み出し／消去等）は、書き込み完了を確認した後、実行してください。 <ul style="list-style-type: none"> <本ユーザ API で完了待ちを行う場合> r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を有効にしてください。 <本ユーザ API で完了待ちを行わない場合> r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、R_QSPI_FLASH_Wait()をコールしてください。この処理方法では、ユーザの任意のタイミングで書き込み完了待ちを確認することができます。使用方法は図 5-12 を参照してください。
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pConfig ; 書き込み用バッファポインタ
リターン値	<ul style="list-style-type: none"> ・ コンフィグレーションレジスタ書き込み結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_TIMEOUT ; Timeout error (「FLASH_WAIT_READY」有効時) FLASH_ERR_OTHER ; Other error

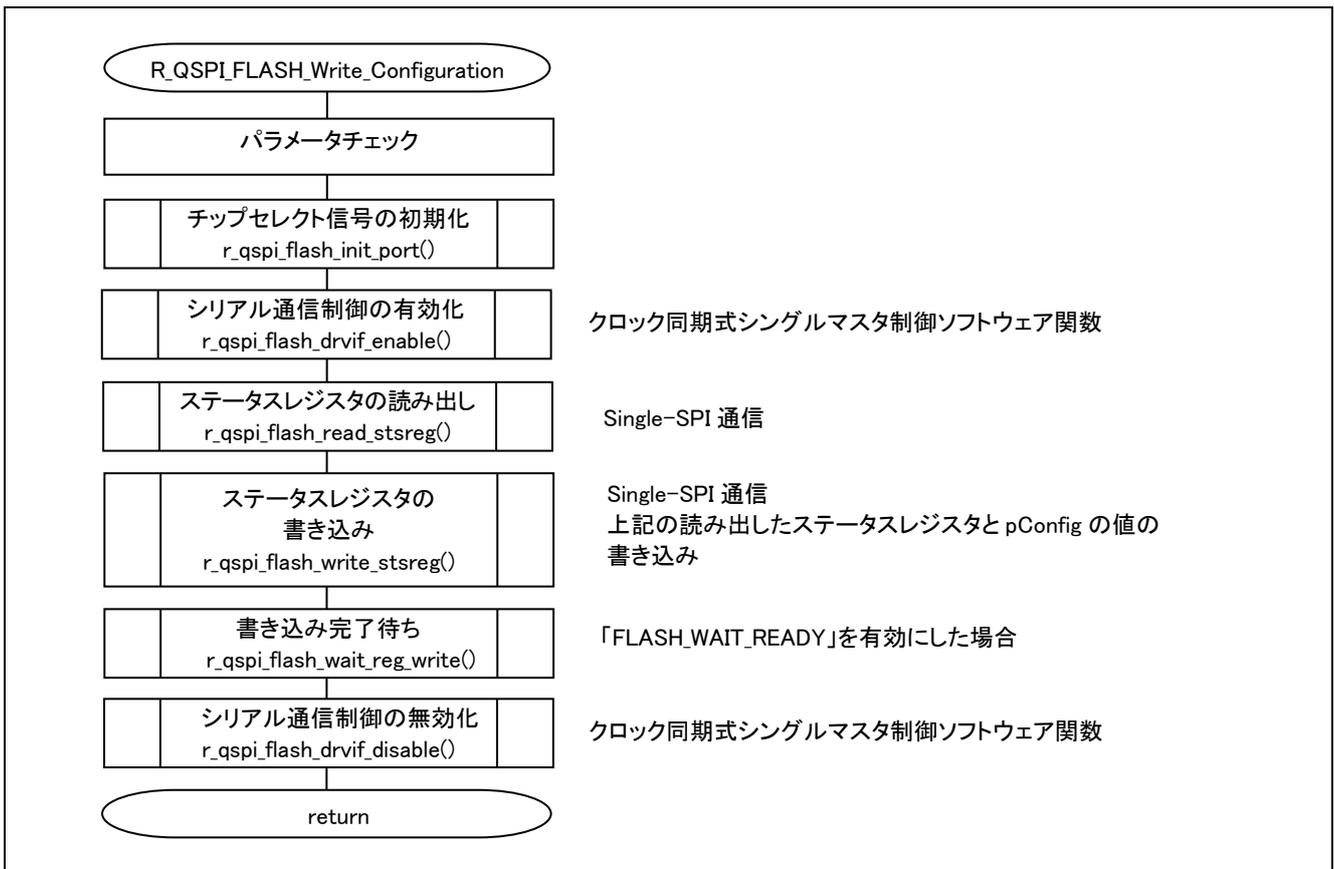


図 5-11 コンフィグレーションレジスタ書き込み処理概要

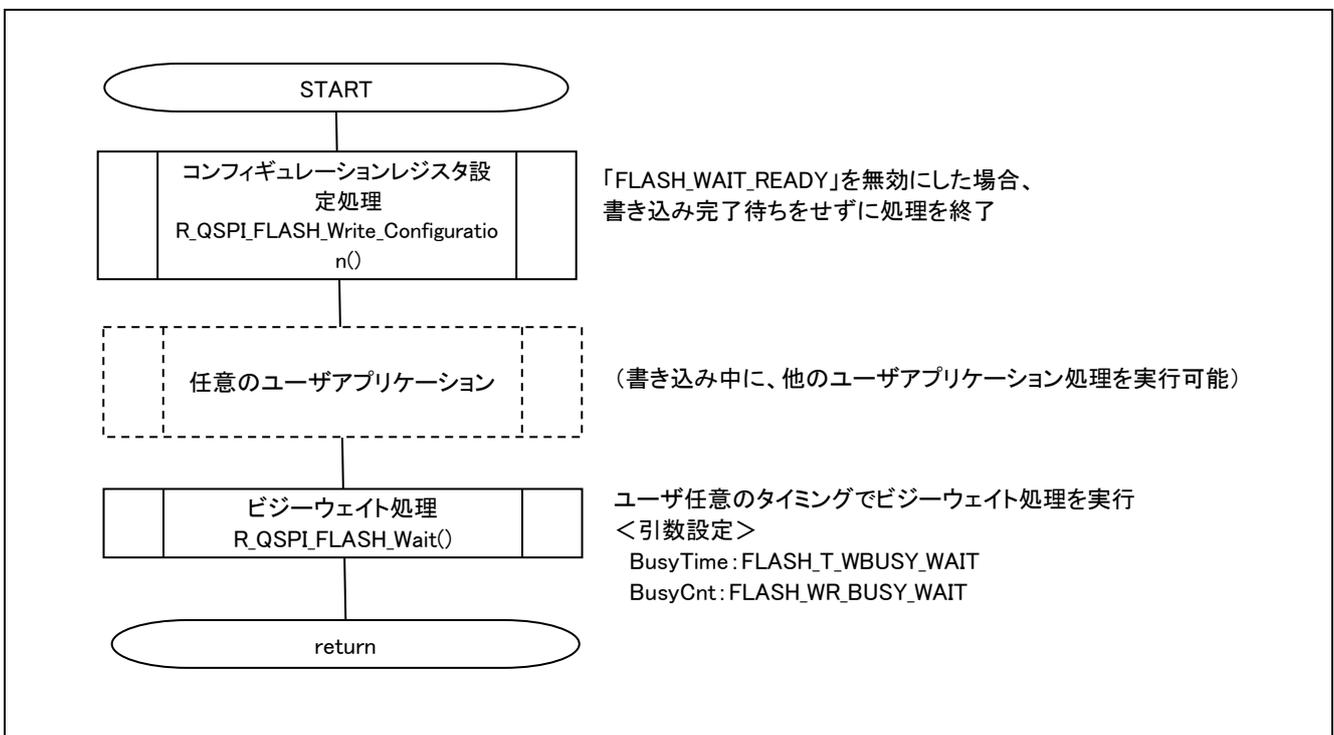


図 5-12 R_QSPI_FLASH_Wait()を使用したコンフィグレーションレジスタ書き込み完了待ち方法

5.9.5 セキュリティレジスタ読み出し処理

R_QSPI_FLASH_Read_Security	
概要	セキュリティレジスタ読み出し処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Read_Security(uint8_t DevNo, uint8_t FAR* pScur)
説明	<ul style="list-style-type: none"> ・セキュリティレジスタを読み出し、pScur 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。 ・読み出し用バッファ(pScur)には下記情報が格納されます。 <ul style="list-style-type: none"> Bit 7: WPSEL <ul style="list-style-type: none"> 1: Individual mode 0: Normal WP mode Bit 6: E_FAIL <ul style="list-style-type: none"> 1: Erase failed 0: Erase succeed Bit 5: P_FAIL <ul style="list-style-type: none"> 1: Program failed 0: Program succeed Bit 4: Reserved Bit 3: ESB (Erase Suspend Bit) <ul style="list-style-type: none"> 1: Erase Suspended 0: Erase is not suspended Bit 2: PSB (Program Suspend Bit) <ul style="list-style-type: none"> 1: Program Suspended 0: Program is not suspended Bit 1: LDSO (Indicate if lock-down) <ul style="list-style-type: none"> 1: Lock-down (Cannot program/erase OTP) 0: Not lock-down Bit 0: Secured OTP indicator <ul style="list-style-type: none"> 1: Factory lock 0: Non-factory lock. ・P_FAIL=1 の場合、次のプログラム処理が成功すると 0 クリアされます。 ・E_FAIL=1 の場合、次の消去処理が成功すると 0 クリアされます。
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pScur ; 読み出し用バッファポインタ
リターン値	<ul style="list-style-type: none"> ・セキュリティレジスタ取得結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

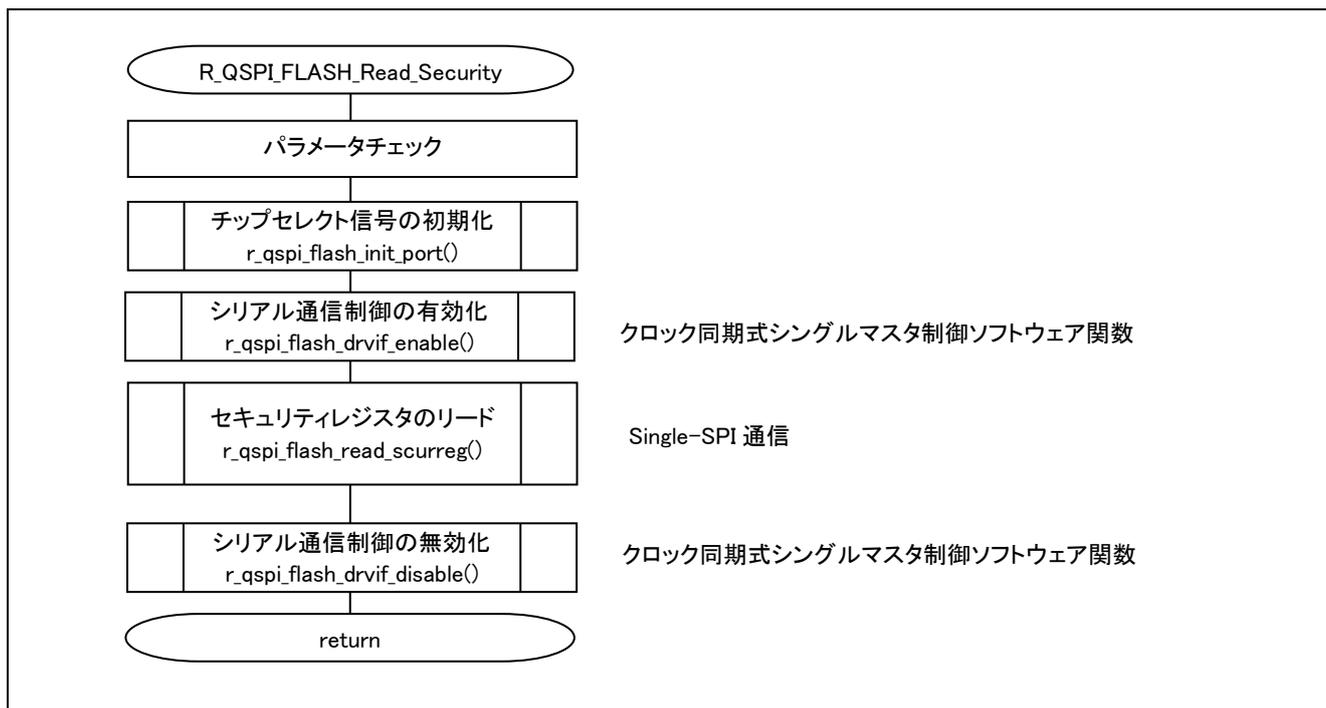


図 5-13 セキュリティレジスタ読み出し処理概要

5.9.6 ライトプロテクト設定処理

R_QSPI_FLASH_Set_Write_Protect

概要	ライトプロテクト設定処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Set_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。SRWD は、0 に設定されます。 ・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。

WpSts	BP3	BP2	BP1	BP0
0x00	0	0	0	0
0x01	0	0	0	1
0x02	0	0	1	0
0x03	0	0	1	1
0x04	0	1	0	0
0x05	0	1	0	1
0x06	0	1	1	0
0x07	0	1	1	1
0x08	1	0	0	0
0x09	1	0	0	1
0x0a	1	0	1	0
0x0b	1	0	1	1
0x0c	1	1	0	0
0x0d	1	1	0	1
0x0e	1	1	1	0
0x0f	1	1	1	1

- ・プロテクト領域とプロテクトビットの関係は、使用する Serial NOR Flash memory のデータシートを参照してください。
- ・処理終了後、ステータスレジスタを読み出して書き込み値を確認してください。
- ・Top/Bottom 設定は、コンフィグレーション書き込み処理で設定してください。
- ・書き込み完了待ちの方法は二種類あります。以降にその方法を示します。なお、次の処理（書き込み／読み出し／消去等）は、書き込み完了を確認した後、実行してください。

<本ユーザ API で完了待ちを行う場合>

r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を有効にしてください。

<本ユーザ API で完了待ちを行わない場合>

r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、R_QSPI_FLASH_Wait()をコールしてください。この処理方法では、ユーザの任意のタイミングで書き込み完了待ちを確認することができます。使用方法は図 5-15 を参照してください。

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。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_TIMEOUT ; Timeout error (「FLASH_WAIT_READY」有効時) FLASH_ERR_OTHER ; Other error

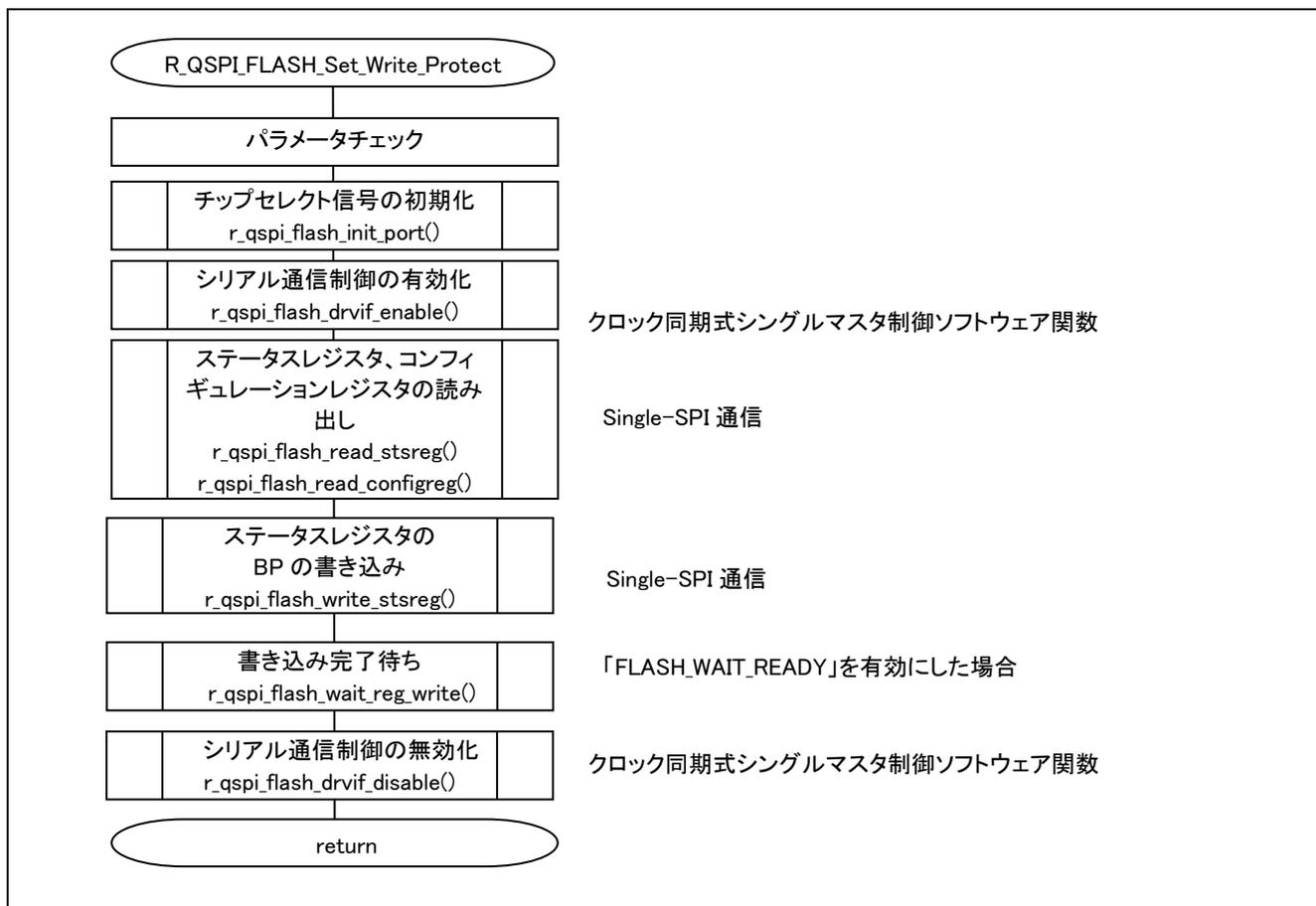


図 5-14 ライトプロテクト設定処理概要

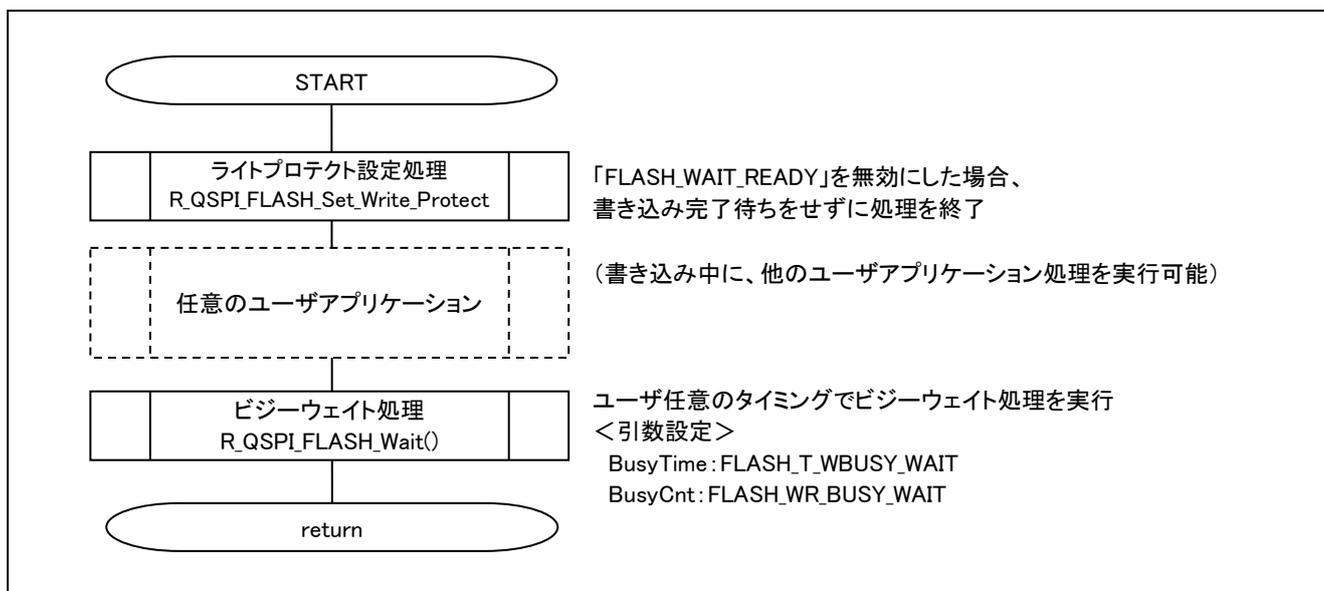


図 5-15 R_QSPI_FLASH_Wait()を使用したライトプロテクト設定完了待ち方法

5.9.7 Quad モード許可設定処理

R_QSPI_FLASH_Quad_Enable	
概 要	Quad モード許可設定処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣 言	error_t R_QSPI_FLASH_Quad_Enable(uint8_t DevNo)
説 明	<ul style="list-style-type: none"> ・ステータスレジスタの Quad Enable (QE) bit を 1 に設定し、Quad モードにします。 ・Quad モードで動作させる場合、事前に本関数をコールしてください。 ・処理終了後、ステータスレジスタを読み出して、QE bit を確認してください。 ・Quad Enable (QE) bit は、Non-volatile bit です。一度 Quad モードに設定した後、Quad モードを禁止にする場合、Quad モード禁止設定処理を実行してください。 ・書き込み完了待ちの方法は二種類あります。以降にその方法を示します。なお、次の処理（書き込み／読み出し／消去等）は、書き込み完了を確認した後、実行してください。 <p>＜本ユーザ API で完了待ちを行う場合＞</p> <p>r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を有効にしてください。</p> <p>＜本ユーザ API で完了待ちを行わない場合＞</p> <p>r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、R_QSPI_FLASH_Wait()をコールしてください。この処理方法では、ユーザの任意のタイミングで書き込み完了待ちを確認することができます。使用方法は図 5-17 を参照してください。</p>
引 数	uint8_t DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> ・設定結果を返します。 <p>FLASH_OK ; Successful operation</p> <p>FLASH_ERR_PARAM ; Parameter error</p> <p>FLASH_ERR_HARD ; Hardware error</p> <p>FLASH_ERR_TIMEOUT ; Timeout error（「FLASH_WAIT_READY」有効時）</p> <p>FLASH_ERR_OTHER ; Other error</p>

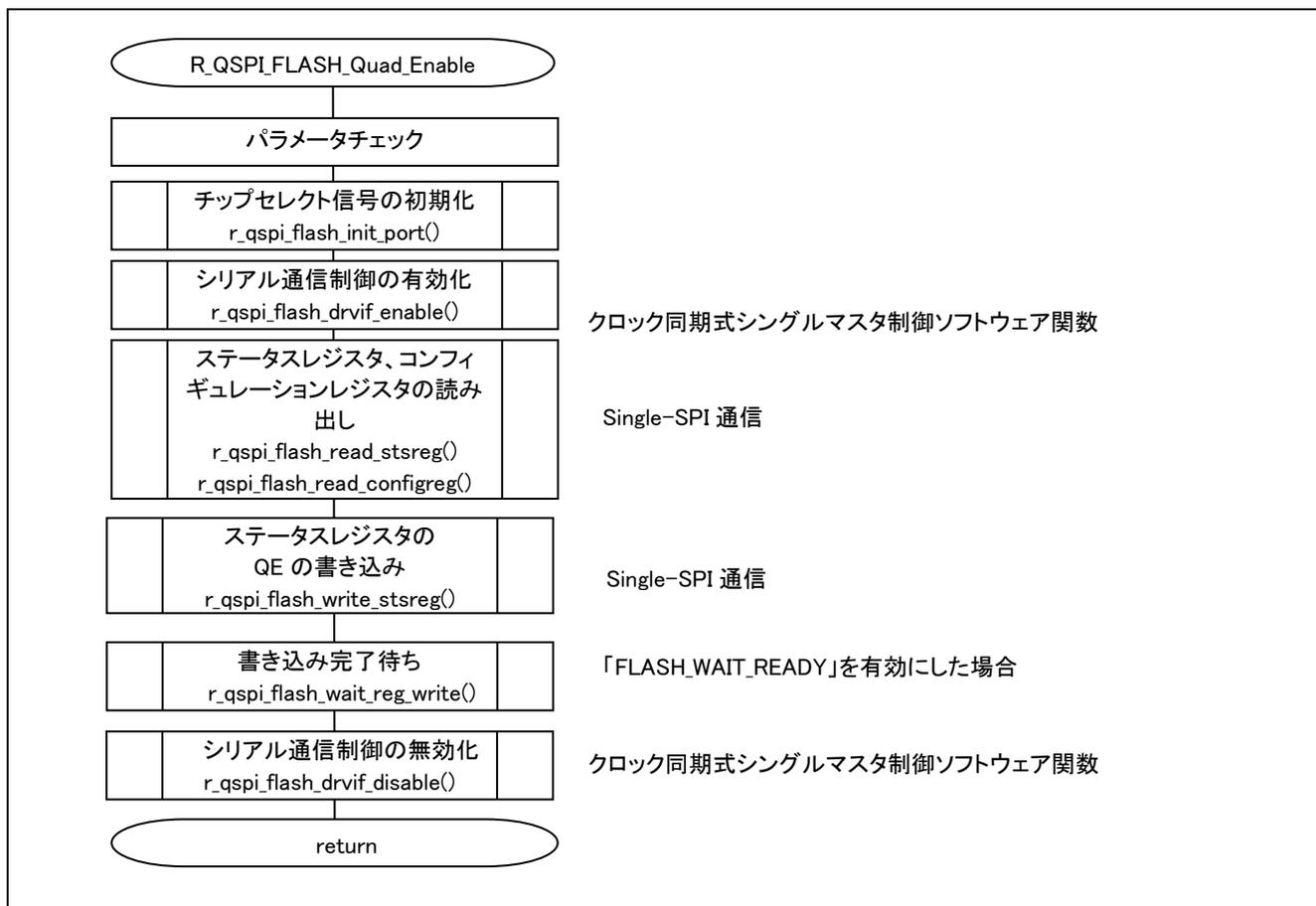


図 5-16 Quadモード許可設定処理概要

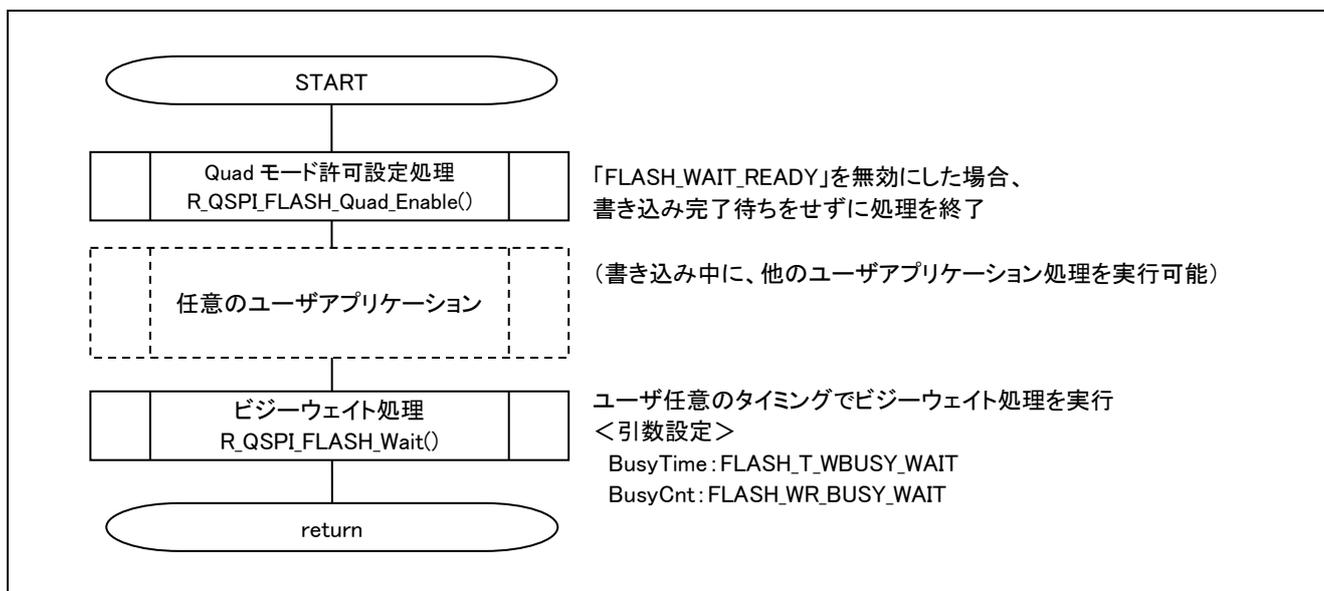


図 5-17 R_QSPI_FLASH_Wait()を使用したQuadモード許可設定完了待ち方法

5.9.8 Quad モード禁止設定処理

R_QSPI_FLASH_Quad_Disable	
概要	Quad モード禁止設定処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Quad_Disable(uint8_t DevNo)
説明	<ul style="list-style-type: none"> ・ステータスレジスタの Quad Enable (QE) bit を 0 に設定し、Quad モードをキャンセルします。 ・処理終了後、ステータスレジスタを読み出して、QE bit を確認してください。 ・Quad Enable (QE) bit は、Non-volatile bit です。一度 Quad モードに設定した後、Quad モードを禁止にする場合、本関数を実行してください。 ・書き込み完了待ちの方法は二種類あります。以降にその方法を示します。なお、次の処理（書き込み／読み出し／消去等）は、書き込み完了を確認した後、実行してください。 <p>＜本ユーザ API で完了待ちを行う場合＞</p> <p>r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を有効にしてください。</p> <p>＜本ユーザ API で完了待ちを行わない場合＞</p> <p>r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、R_QSPI_FLASH_Wait()をコールしてください。この処理方法では、ユーザの任意のタイミングで書き込み完了待ちを確認することができません。使用方法は図 5-19 を参照してください。</p>
引数	uint8_t DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> ・設定結果を返します。 <p>FLASH_OK ; Successful operation</p> <p>FLASH_ERR_PARAM ; Parameter error</p> <p>FLASH_ERR_HARD ; Hardware error</p> <p>FLASH_ERR_TIMEOUT ; Timeout error (「FLASH_WAIT_READY」有効時)</p> <p>FLASH_ERR_OTHER ; Other error</p>

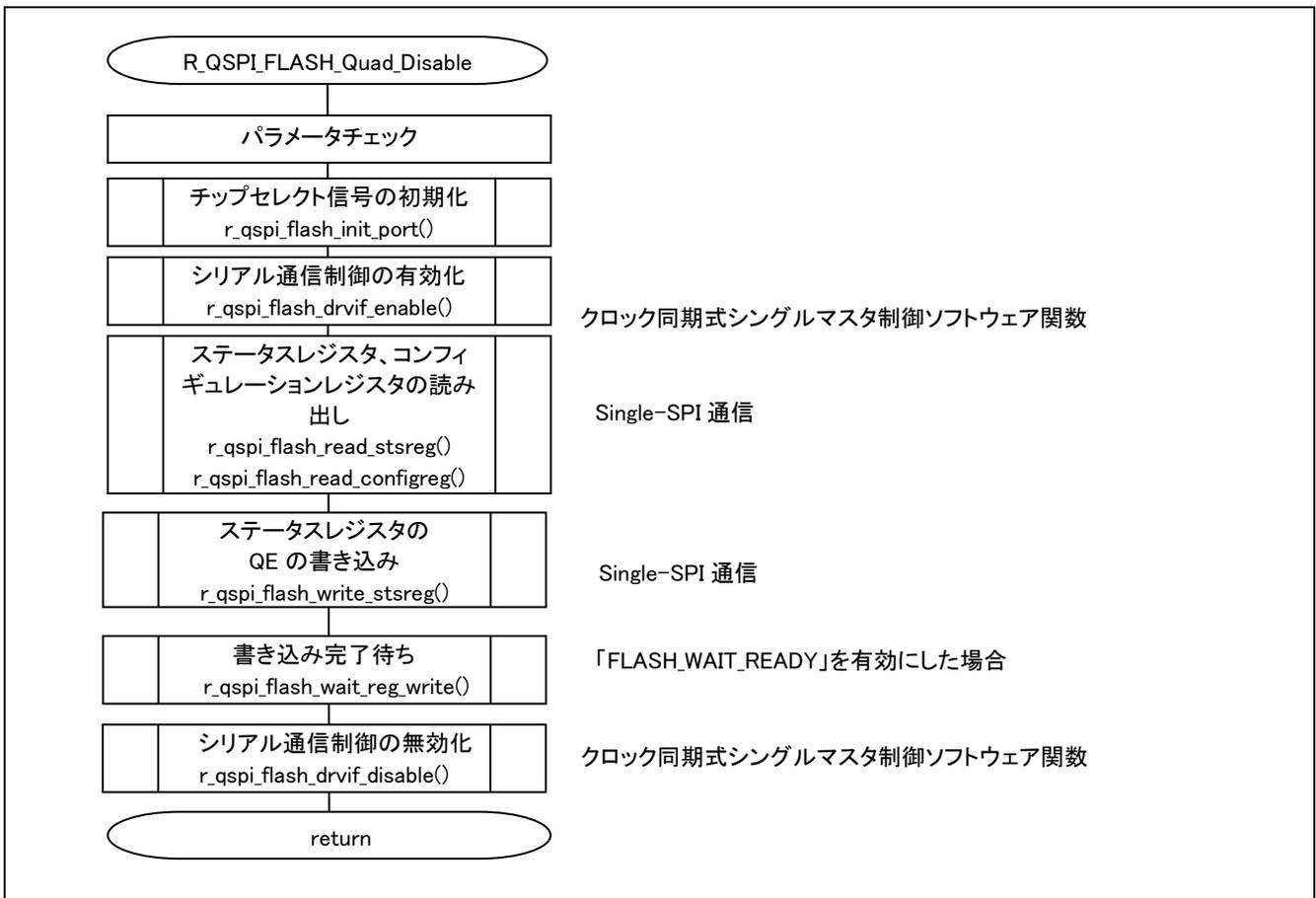


図 5-18 Quadモード禁止設定処理概要

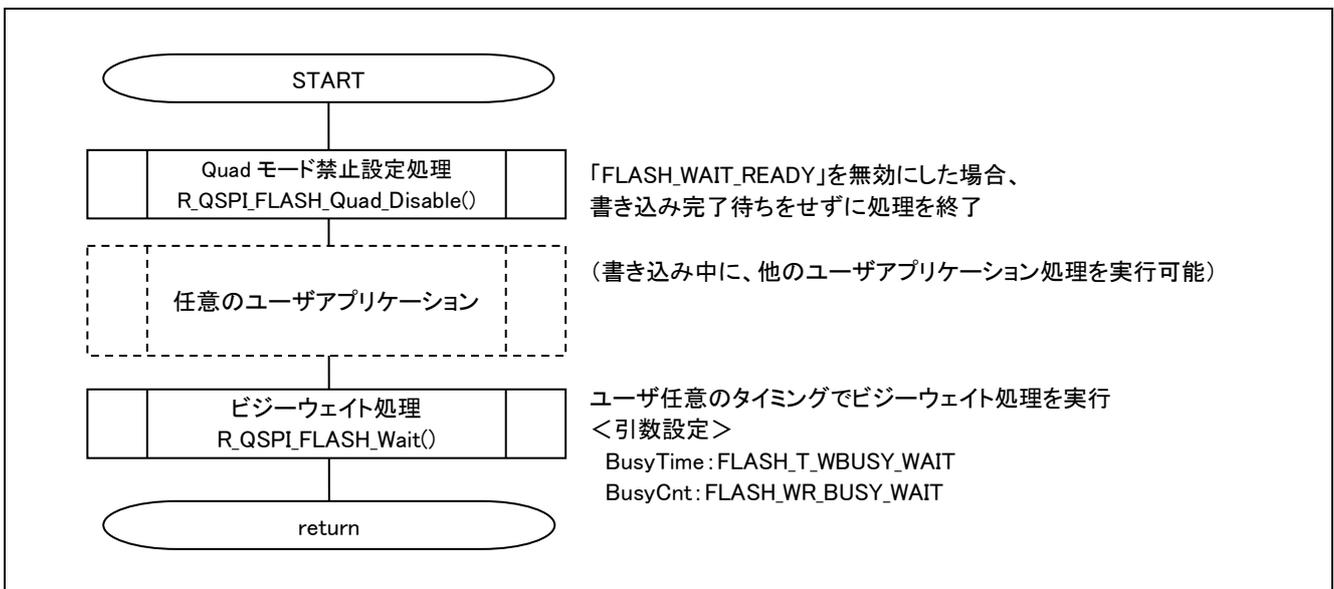


図 5-19 R_QSPI_FLASH_Wait()を使用したQuadモード禁止設定完了待ち方法

5.9.9 WRDI コマンド発行処理

R_QSPI_FLASH_Write_Di	
概要	WRDI コマンド発行処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Write_Di(uint8_t DevNo)
説明	・ステータスレジスタの WEL ビットをクリアします。
引数	uint8_t DevNo ; デバイス番号
リターン値	・クリア結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

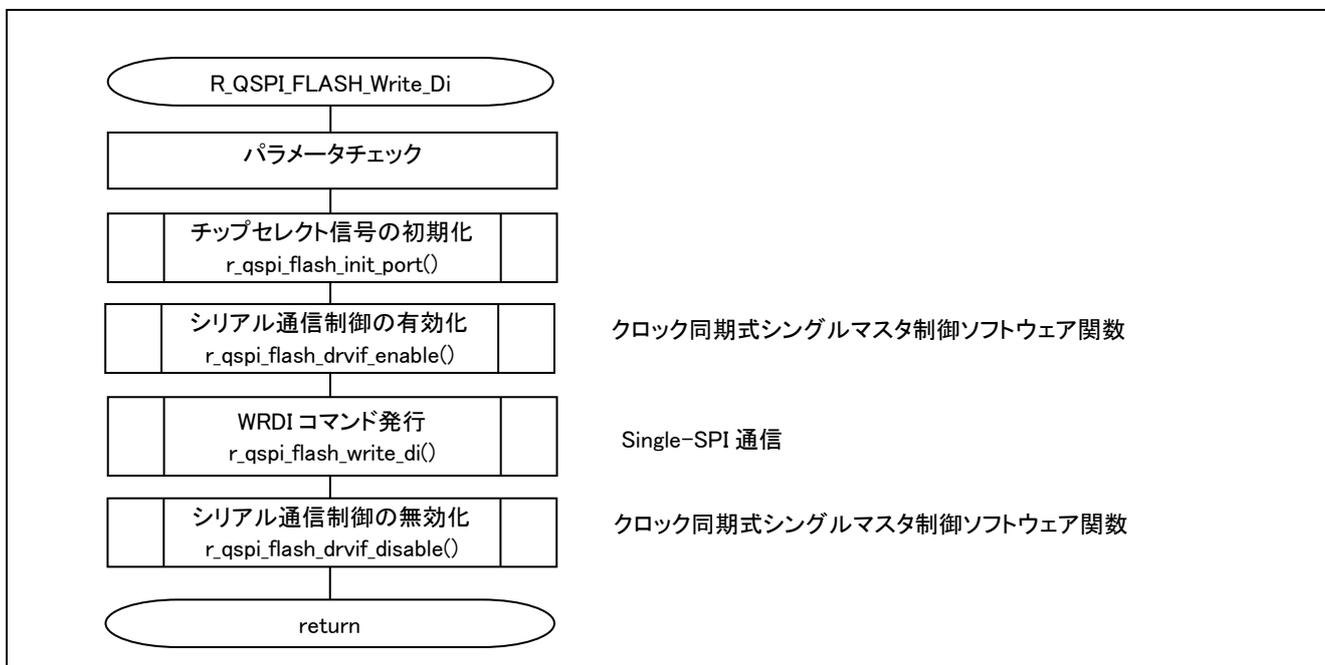


図 5-20 WRDI コマンド発行処理概要

5.9.10 データ読み出し処理

R_QSPI_FLASH_Read_Data																									
概要	データ読み出し処理																								
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h																								
宣言	error_t R_QSPI_FLASH_Read_Data(uint8_t DevNo, r_qspi_flash_info_t FAR* pFlash_Info)																								
説明	<ul style="list-style-type: none"> Serial NOR Flash memory 上の指定アドレスから指定バイト数分、データを読み出し、pData に格納します。 最終読み出しアドレスは、Serial NOR Flash memory 容量-1 です。 ロールオーバーによる読み出しはできません。最終アドレスの読み出し後、一度処理を完了させて、再度アドレスを設定し直してから、本ユーザ API をコールしてください。 																								
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>r_qspi_flash_info_t FAR*</td> <td>pFlash_Info</td> <td>;</td> <td>FLASH 通信情報構造体</td> </tr> <tr> <td>uint32_t</td> <td>Addr</td> <td>;</td> <td>読み出し開始アドレス</td> </tr> <tr> <td>uint32_t</td> <td>Cnt</td> <td>;</td> <td>読み出しバイト数</td> </tr> <tr> <td>uint16_t</td> <td>DataCnt</td> <td>;</td> <td>読み出しバイト Temp. (設定禁止)</td> </tr> <tr> <td>uint8_t FAR*</td> <td>pData</td> <td>;</td> <td>読み出し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	r_qspi_flash_info_t FAR*	pFlash_Info	;	FLASH 通信情報構造体	uint32_t	Addr	;	読み出し開始アドレス	uint32_t	Cnt	;	読み出しバイト数	uint16_t	DataCnt	;	読み出しバイト Temp. (設定禁止)	uint8_t FAR*	pData	;	読み出しデータ格納バッファポインタ
uint8_t	DevNo	;	デバイス番号																						
r_qspi_flash_info_t FAR*	pFlash_Info	;	FLASH 通信情報構造体																						
uint32_t	Addr	;	読み出し開始アドレス																						
uint32_t	Cnt	;	読み出しバイト数																						
uint16_t	DataCnt	;	読み出しバイト Temp. (設定禁止)																						
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	<ul style="list-style-type: none"> 読み出し結果を返します。 <table border="0"> <tr> <td>FLASH_OK</td> <td>;</td> <td>Successful operation</td> </tr> <tr> <td>FLASH_ERR_PARAM</td> <td>;</td> <td>Parameter error</td> </tr> <tr> <td>FLASH_ERR_HARD</td> <td>;</td> <td>Hardware error</td> </tr> <tr> <td>FLASH_ERR_OTHER</td> <td>;</td> <td>Other error</td> </tr> </table>	FLASH_OK	;	Successful operation	FLASH_ERR_PARAM	;	Parameter error	FLASH_ERR_HARD	;	Hardware error	FLASH_ERR_OTHER	;	Other error												
FLASH_OK	;	Successful operation																							
FLASH_ERR_PARAM	;	Parameter error																							
FLASH_ERR_HARD	;	Hardware error																							
FLASH_ERR_OTHER	;	Other error																							

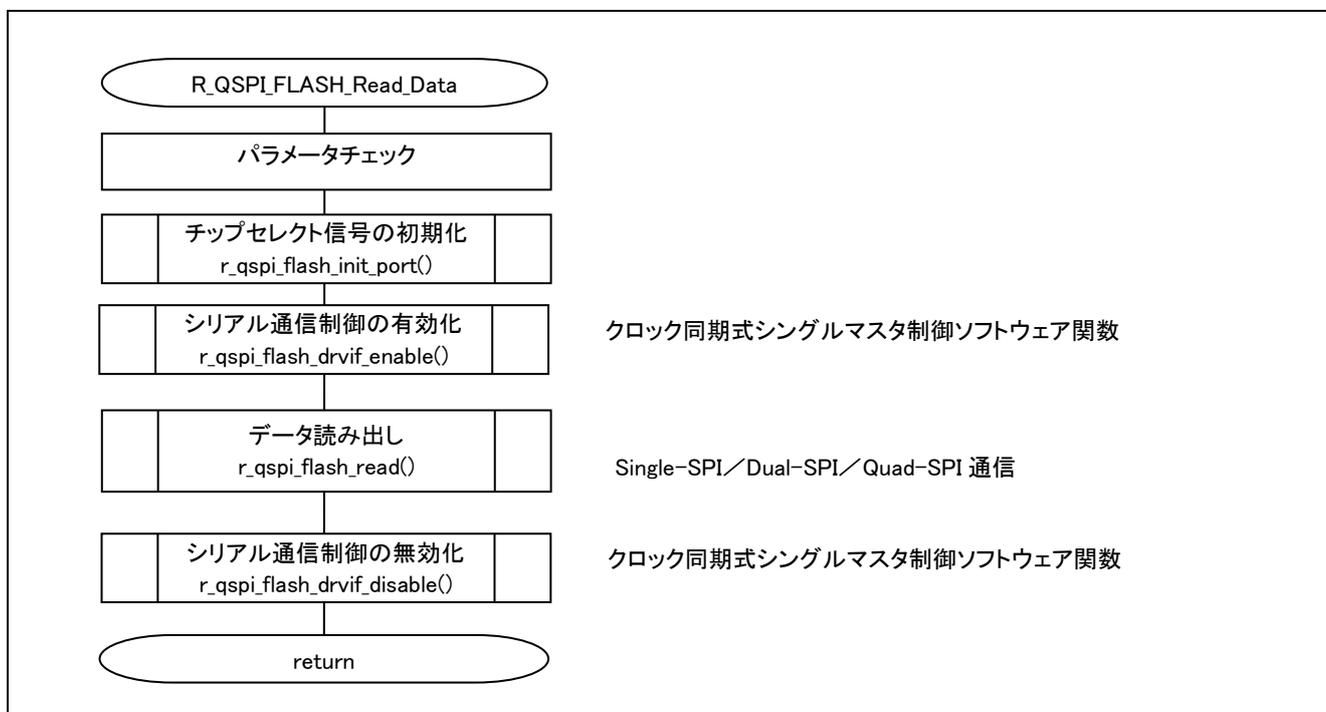


図 5-21 データ読み出し処理概要

5.9.11 データ書き込み処理

R_QSPI_FLASH_Write_Data	
概要	データ書き込み処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Write_Data(uint8_t DevNo, r_qspi_flash_info_t FAR* pFlash_Info)
説明	<ul style="list-style-type: none"> ・pData のデータを Serial NOR Flash memory 上の指定アドレスから指定バイト数分書き込みます。 ・Serial NOR Flash memory への書き込みは、ライトプロテクト解除領域のみ可能です。プロテクトされた領域への書き込みはできません。「FLASH_ERR_OTHER」を返します。 ・最終書き込みアドレスは、Serial NOR Flash memory 容量-1 です。 ・書き込みバイト数 (Cnt) に設定できる最大値は、Serial NOR Flash memory 容量の値です。 ・本ユーザ API では、r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」の設定有無に関わらず、書き込み完了待ちを行います。
引数	uint8_t DevNo ; デバイス番号 r_qspi_flash_info_t FAR* pFlash_Info ; FLASH 通信情報構造体 uint32_t Addr ; 書き込み開始アドレス uint32_t Cnt ; 書き込みバイト数 uint16_t DataCnt ; 書き込みバイト Temp. (設定禁止) 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"> ・読み出し結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_TIMEOUT ; Timeout error FLASH_ERR_OTHER ; Other error

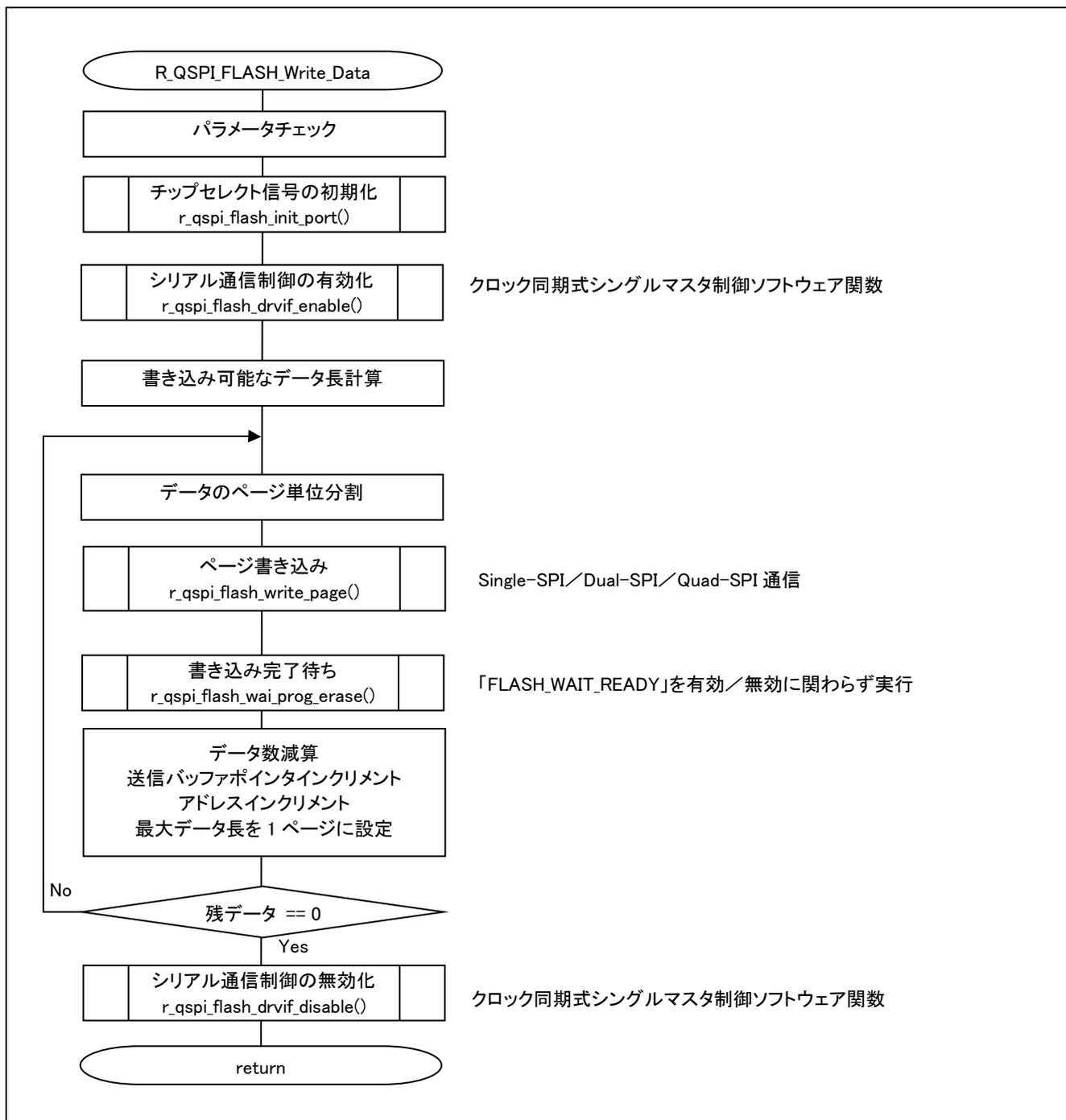


図 5-22 データ書き込み処理概要

5.9.12 データ書き込み処理（1Page 書き込み用）

R_QSPI_FLASH_Write_Data_Page																									
概要	データ書き込み処理（1Page 書き込み用）																								
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h																								
宣言	error_t R_QSPI_FLASH_Write_Data_Page(uint8_t DevNo, r_qspi_flash_info_t FAR* pFlash_Info)																								
説明	<ul style="list-style-type: none"> ・ pData のデータを Serial NOR Flash memory 上の指定アドレスから指定バイト数分（最大：1Page）書き込みます。 ・ 大容量のデータ書き込みの際、Page 単位に通信を分割するため、通信中に他の処理ができなくなることを防ぐことができます。 ・ Serial NOR Flash memory への書き込みは、ライトプロテクト解除領域のみ可能です。プロテクトされた領域への書き込みはできません。「FLASH_ERR_OTHER」を返します。 ・ 最終書き込みアドレスは、Serial NOR Flash memory 容量-1 です。 ・ 書き込みバイト数（Cnt）に設定できる最大値は、Serial NOR Flash memory 容量の値です。 ・ 1Page を超えるバイト数が設定されている場合でも、1Page 書き込み処理完了後、残バイト数と次アドレス情報が FLASH 通信情報構造体（pFlash_Info）に残ります。未変更のまま再びその pFlash_Info をセットすることで残バイト数の書き込みが可能です。 ・ 書き込み完了待ちの方法は二種類あります。以降にその方法を示します。なお、次の処理（書き込み／読み出し／消去等）は、書き込み完了を確認した後、実行してください。 ・ 本ユーザ API で完了待ちを行う場合、r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を有効にしてください。 ・ 本ユーザ API で完了待ちを行わない場合、r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、R_QSPI_FLASH_Wait()をコールしてください。この処理方法では、ユーザの任意のタイミングで書き込み完了待ちを確認することができます。使用方法は図 5-24 を参照してください。 																								
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>r_qspi_flash_info_t FAR*</td> <td>pFlash_Info</td> <td>;</td> <td>FLASH 通信情報構造体</td> </tr> <tr> <td>uint32_t</td> <td>Addr</td> <td>;</td> <td>書き込み開始アドレス</td> </tr> <tr> <td>uint32_t</td> <td>Cnt</td> <td>;</td> <td>書き込みバイト数</td> </tr> <tr> <td>uint16_t</td> <td>DataCnt</td> <td>;</td> <td>書き込みバイト Temp.（設定禁止）</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	r_qspi_flash_info_t FAR*	pFlash_Info	;	FLASH 通信情報構造体	uint32_t	Addr	;	書き込み開始アドレス	uint32_t	Cnt	;	書き込みバイト数	uint16_t	DataCnt	;	書き込みバイト Temp.（設定禁止）	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																						
r_qspi_flash_info_t FAR*	pFlash_Info	;	FLASH 通信情報構造体																						
uint32_t	Addr	;	書き込み開始アドレス																						
uint32_t	Cnt	;	書き込みバイト数																						
uint16_t	DataCnt	;	書き込みバイト Temp.（設定禁止）																						
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"> ・ 読み出し結果を返します。 <table border="0"> <tr> <td>FLASH_OK</td> <td>;</td> <td>Successful operation</td> </tr> <tr> <td>FLASH_ERR_PARAM</td> <td>;</td> <td>Parameter error</td> </tr> <tr> <td>FLASH_ERR_HARD</td> <td>;</td> <td>Hardware error</td> </tr> <tr> <td>FLASH_ERR_TIMEOUT</td> <td>;</td> <td>Timeout error（「FLASH_WAIT_READY」有効時）</td> </tr> <tr> <td>FLASH_ERR_OTHER</td> <td>;</td> <td>Other error</td> </tr> </table>	FLASH_OK	;	Successful operation	FLASH_ERR_PARAM	;	Parameter error	FLASH_ERR_HARD	;	Hardware error	FLASH_ERR_TIMEOUT	;	Timeout error（「FLASH_WAIT_READY」有効時）	FLASH_ERR_OTHER	;	Other error									
FLASH_OK	;	Successful operation																							
FLASH_ERR_PARAM	;	Parameter error																							
FLASH_ERR_HARD	;	Hardware error																							
FLASH_ERR_TIMEOUT	;	Timeout error（「FLASH_WAIT_READY」有効時）																							
FLASH_ERR_OTHER	;	Other error																							

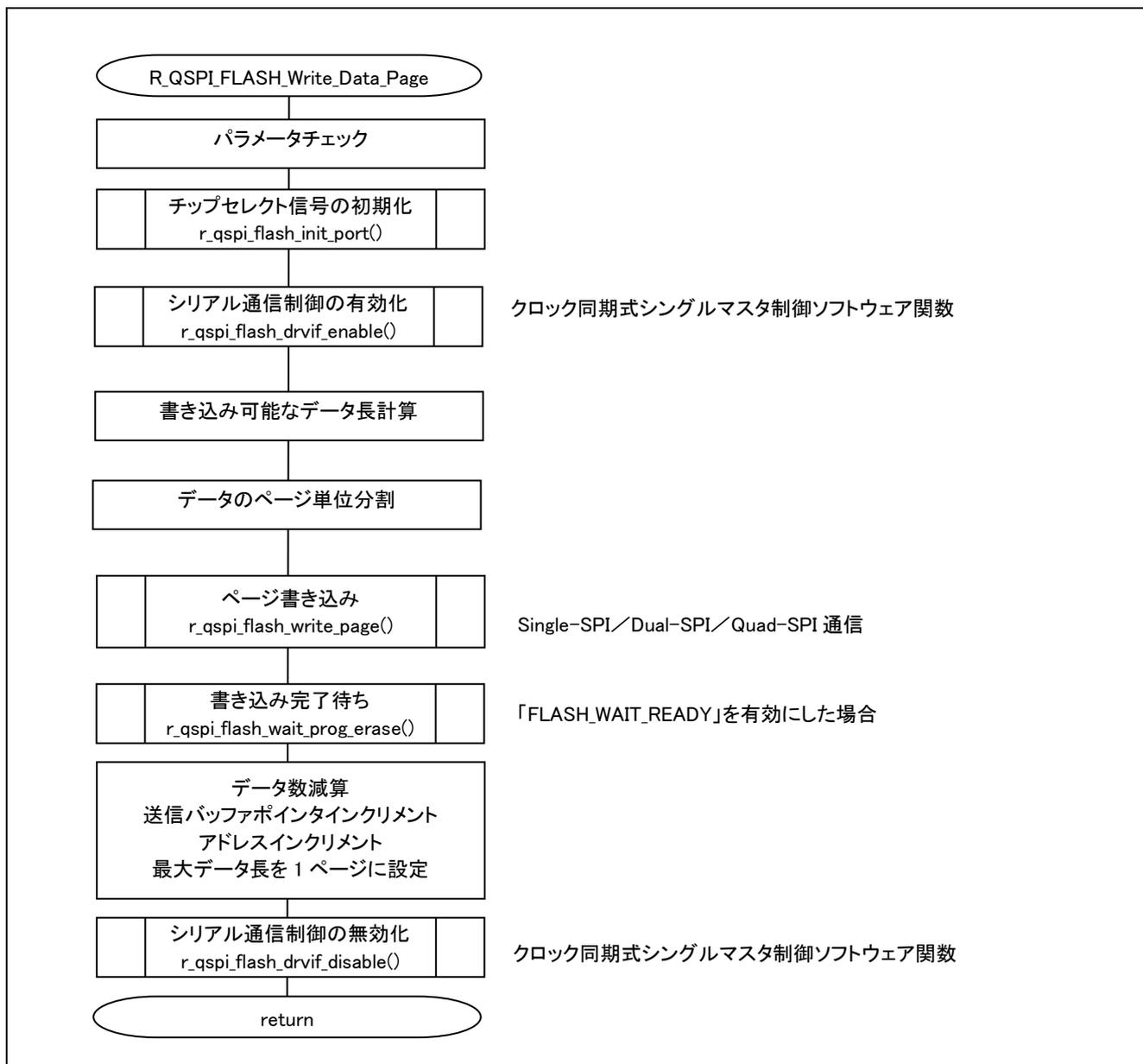


図 5-23 データ書き込み処理（1Page 書き込み用）概要

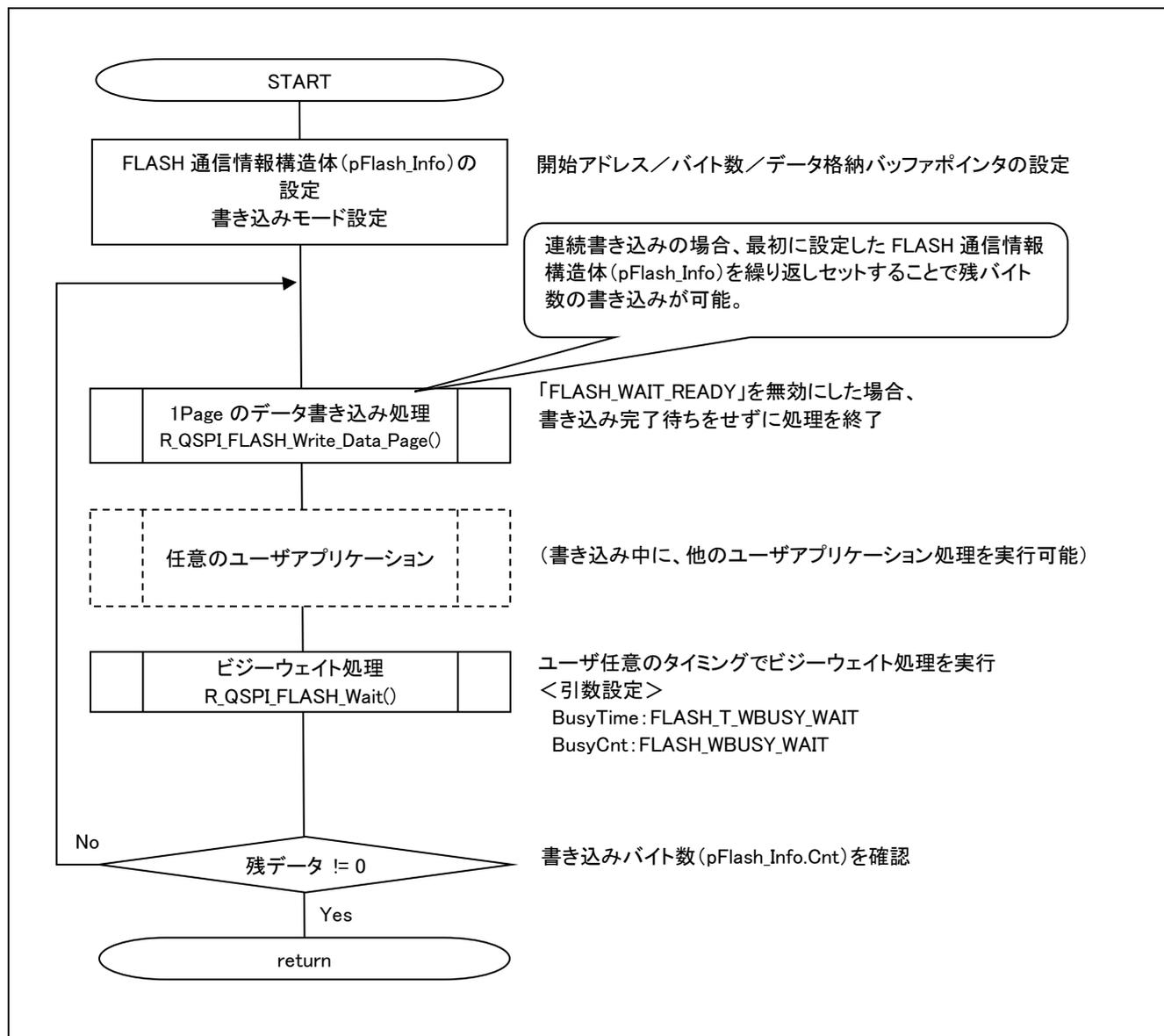


図 5-24 R_QSPI_Flash_Wait()を使用したデータ書き込み処理 (1Page 書き込み用) の完了待ち方法

5.9.13 消去処理

R_QSPI_FLASH_Erase																						
概要	消去処理																					
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h																					
宣言	error_t R_QSPI_FLASH_Erase(uint8_t DevNo, uint32_t Addr, uint8_t Mode)																					
説明	<ul style="list-style-type: none"> Mode の設定により、指定されたセクタの全データ消去（Sector Erase）、指定されたブロックの全データ消去（Block Erase：32KB ブロック、64KB ブロック）、指定されたチップの全データ消去（Chip Erase）を行います。 Sector Erase の場合、Addr にはセクタの先頭アドレスを設定してください。 Block Erase の場合、Addr にはブロックの先頭アドレスを設定してください。 Chip Erase の場合、Addr には 0x00000000 を設定してください。 Serial NOR Flash memory の消去は、ライトプロテクト解除領域のみ可能です。プロテクトされている場合、消去はできません。「FLASH_ERR_OTHER」を返します。 消去完了待ちの方法は二種類あります。以降にその方法を示します。なお、次の処理（書き込み／読み出し／消去等）は、消去完了を確認した後、実行してください。 本ユーザ API で完了待ちを行う場合、r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を有効にしてください。 本ユーザ API で完了待ちを行わない場合、r_qspi_flash_mx25l.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、R_QSPI_FLASH_Wait()をコールしてください。この処理方法では、ユーザの任意のタイミングで書き込み完了待ちを確認することができます。使用方法は図 5-26 を参照してください。 設定された Mode により、R_QSPI_FLASH_Wait()をコールする際の引数 BusyCnt の設定が異なります。 <ul style="list-style-type: none"> Sector Erase (4KB)時 ; BusyCnt = FLASH_SE_BUSY_WAIT Block Erase (32KB)時 ; BusyCnt = FLASH_BE32K_BUSY_WAIT Block Erase (64KB)時 ; BusyCnt = FLASH_BE64K_BUSY_WAIT Chip Erase 時 ; BusyCnt = FLASH_CE_BUSY_WAIT 																					
引数	<table> <tr> <td>uint8_t</td> <td>DevNo</td> <td>; デバイス番号</td> </tr> <tr> <td>uint32_t</td> <td>Addr</td> <td>; 消去アドレス</td> </tr> <tr> <td>uint8_t</td> <td>Mode</td> <td>; 消去モード（下記より 1 つ選択）</td> </tr> <tr> <td></td> <td></td> <td>FLASH_MODE_S_ERASE</td> </tr> <tr> <td></td> <td></td> <td>FLASH_MODE_B32K_ERASE</td> </tr> <tr> <td></td> <td></td> <td>FLASH_MODE_B64K_ERASE</td> </tr> <tr> <td></td> <td></td> <td>FLASH_MODE_C_ERASE</td> </tr> </table>	uint8_t	DevNo	; デバイス番号	uint32_t	Addr	; 消去アドレス	uint8_t	Mode	; 消去モード（下記より 1 つ選択）			FLASH_MODE_S_ERASE			FLASH_MODE_B32K_ERASE			FLASH_MODE_B64K_ERASE			FLASH_MODE_C_ERASE
uint8_t	DevNo	; デバイス番号																				
uint32_t	Addr	; 消去アドレス																				
uint8_t	Mode	; 消去モード（下記より 1 つ選択）																				
		FLASH_MODE_S_ERASE																				
		FLASH_MODE_B32K_ERASE																				
		FLASH_MODE_B64K_ERASE																				
		FLASH_MODE_C_ERASE																				
リターン値	<ul style="list-style-type: none"> 消去結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_TIMEOUT ; Timeout error（「FLASH_WAIT_READY」有効時） FLASH_ERR_OTHER ; Other error 																					

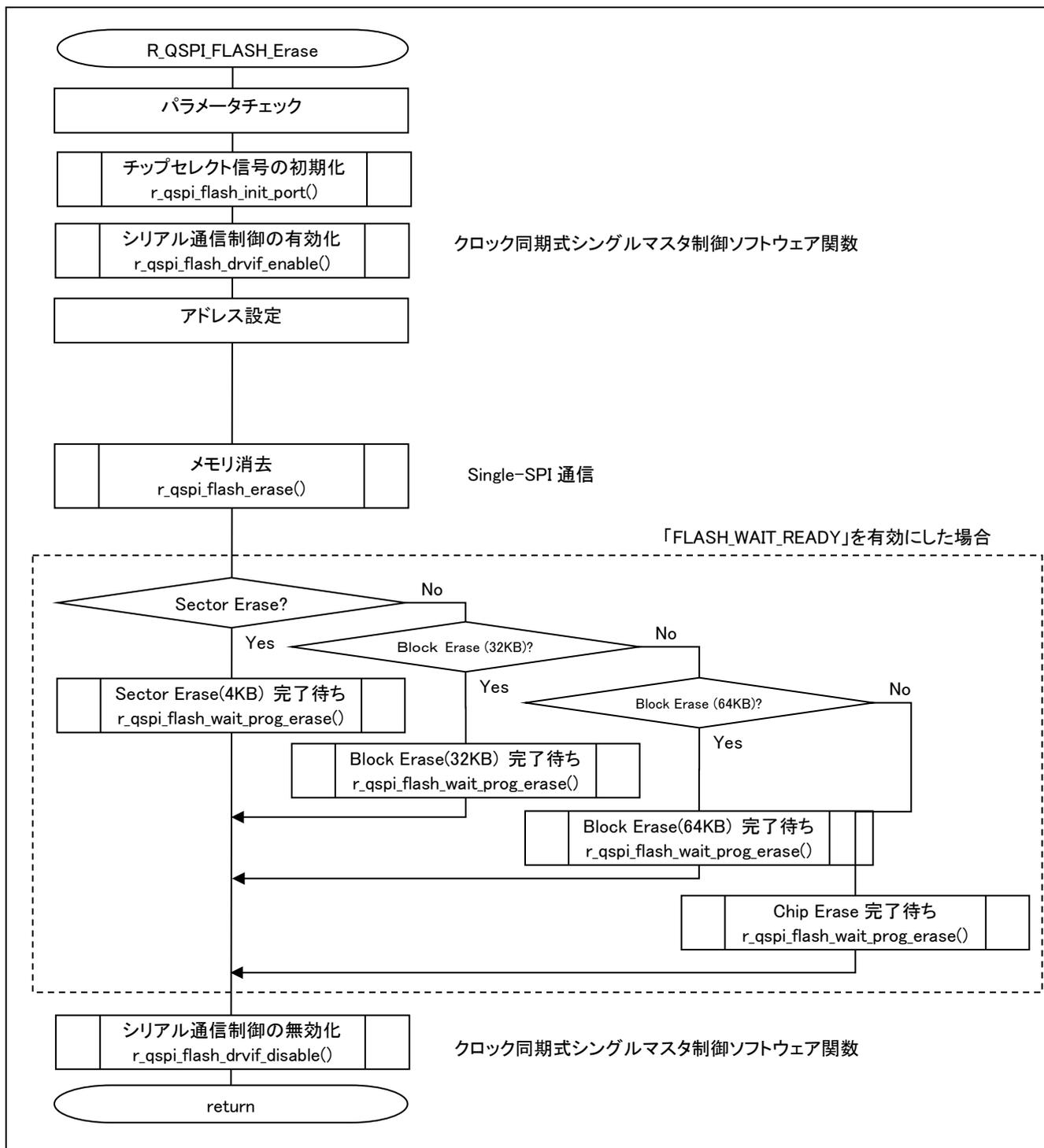


図 5-25 消去処理概要

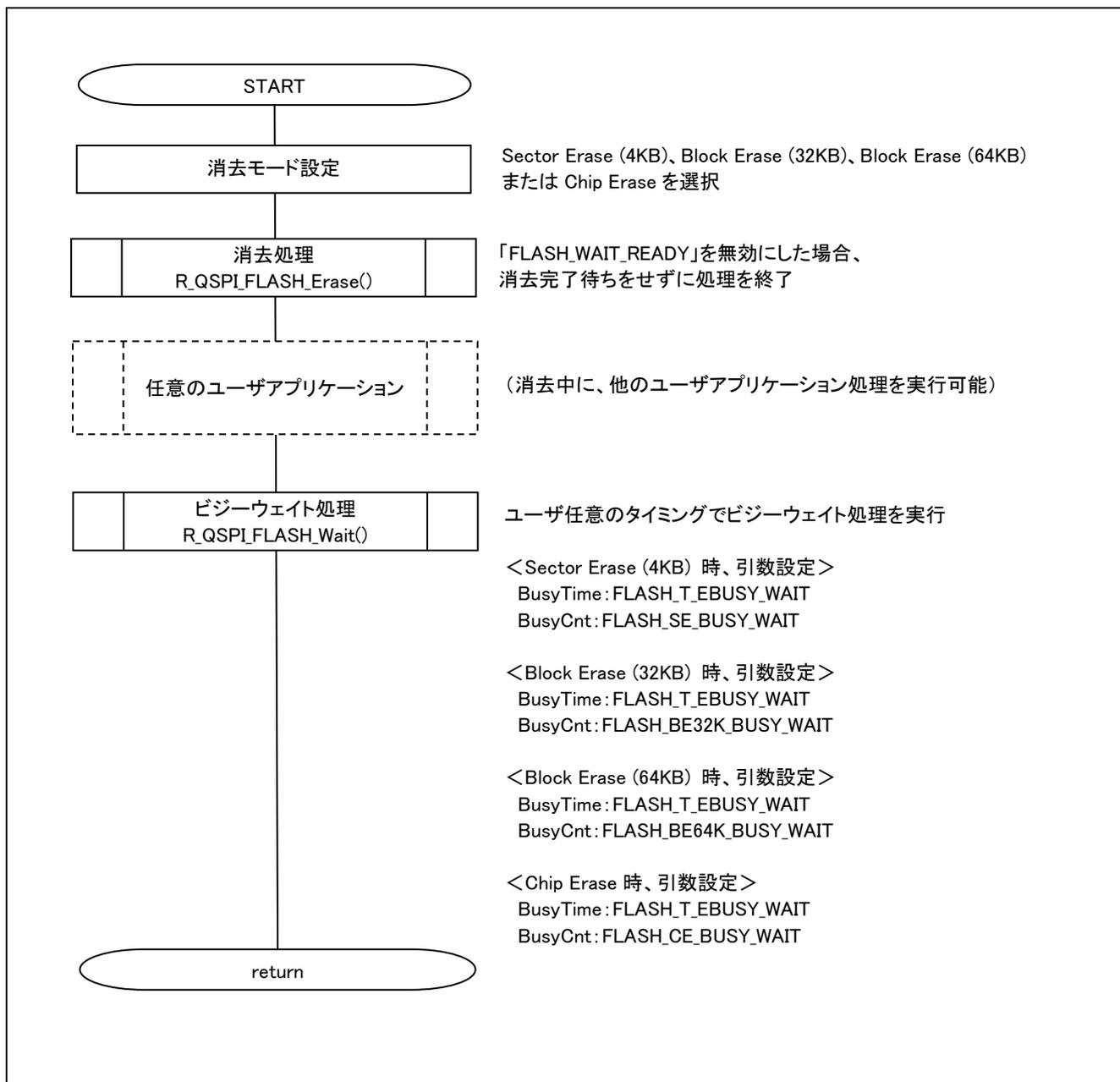


図 5-26 R_QSPI_FLASH_Wait()を使用した消去処理の完了待ち方法

5.9.14 ID 読み出し処理

R_QSPI_FLASH_Read_ID	
概要	ID 読み出し処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Read_ID(uint8_t DevNo, uint8_t FAR* pData)
説明	<ul style="list-style-type: none"> ・ Manufacture ID、及び Device ID を読み出し、pData に格納します。読み出しバッファとして、3 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(pData)には下記情報が格納されます。 <ul style="list-style-type: none"> (1) Manufacturer ID (2) Device ID
引数	uint8_t DevNo ; デバイス番号 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	<ul style="list-style-type: none"> ・ 読み出し結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

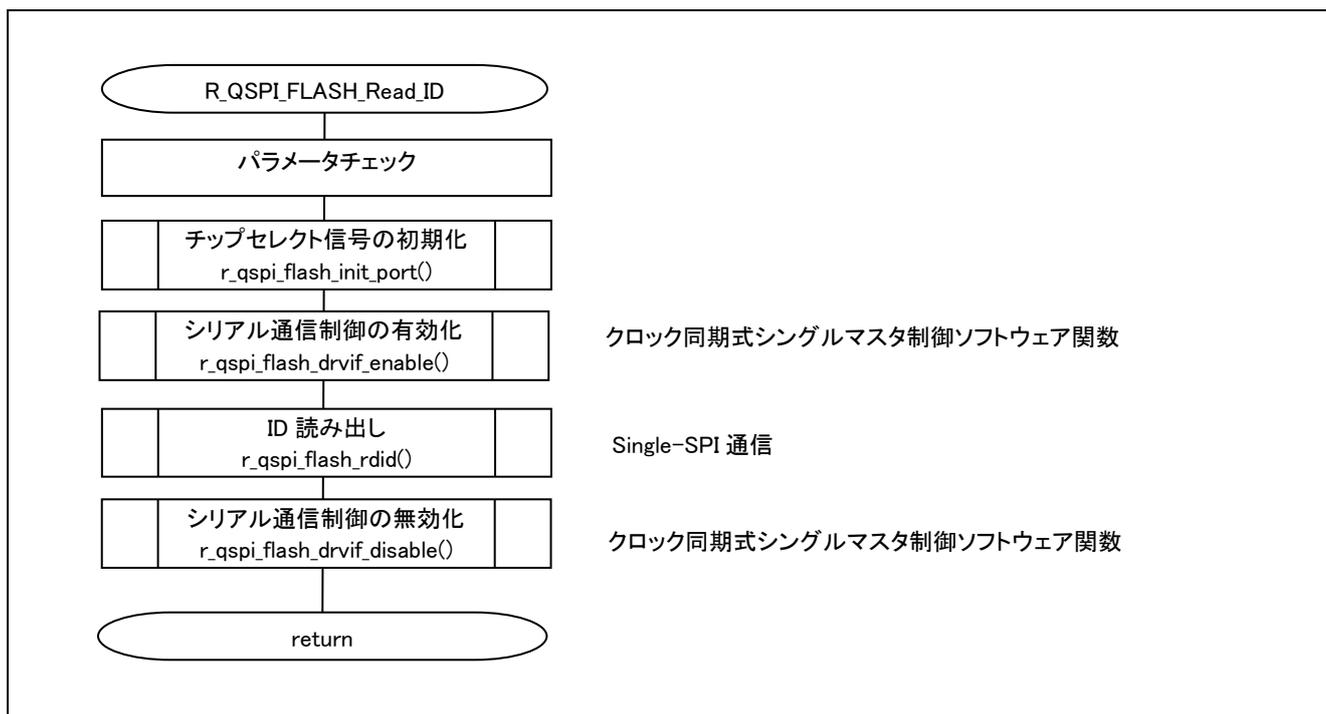


図 5-27 ID 読み出し処理概要

5.9.15 ビジーウェイト処理

R_QSPI_FLASH_Wait																						
概要	ビジーウェイト処理																					
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h																					
宣言	error_t R_QSPI_FLASH_Wait(uint8_t DevNo, uint16_t BusyTime, uint32_t BusyCnt, uint8_t Mode)																					
説明	<ul style="list-style-type: none"> 「FLASH_WAIT_READY」を無効にし、書き込み/消去の完了を確認する場合に使用してください。 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。 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。BusyCnt 回数を超えてもビジーの場合、「FLASH_ERR_TIMEOUT」を返します。 Mode の設定により、レジスタ書き込み待ち、データ書き込み待ち及び消去待ちを行います。 <ul style="list-style-type: none"> FLASH_MODE_REG_WRITE : レジスタ書き込みモード FLASH_MODE_PROG_ERASE : データ書き込み及び消去モード レジスタ書き込みモードの場合、RDSR コマンドを発行し、WIP ビットを使ってレディ/ビジーを判定します。 データ書き込み及び消去モードの場合、最初に WIP ビットを使ってレディ/ビジーを判定します。レディ確認後 RDSCUR コマンドを発行し、E_FAIL/P_FAIL ビットによるエラーを確認します。 書き込み/消去により BusyCnt と BusyTime に設定する値が異なります。意図しない設定でビジーウェイトを行うと、Timeout error が発生する可能性があります。下表にしたがって設定してください。 																					
	<table border="1"> <thead> <tr> <th>状態</th> <th>BusyTime</th> <th>BusyCnt</th> </tr> </thead> <tbody> <tr> <td>レジスタ書き込み中</td> <td>FLASH_T_WBUSY_WAIT</td> <td>FLASH_WR_BUSY_WAIT</td> </tr> <tr> <td>データ書き込み中</td> <td>FLASH_T_PBUSY_WAIT</td> <td>FLASH_PP_BUSY_WAIT</td> </tr> <tr> <td>消去中 (Sector Erase 4KB)</td> <td>FLASH_T_EBUSY_WAIT</td> <td>FLASH_SE_BUSY_WAIT</td> </tr> <tr> <td>消去中 (Block Erase 32KB)</td> <td>FLASH_T_EBUSY_WAIT</td> <td>FLASH_BE32K_BUSY_WAIT</td> </tr> <tr> <td>消去中 (Block Erase 64KB)</td> <td>FLASH_T_EBUSY_WAIT</td> <td>FLASH_BE64K_BUSY_WAIT</td> </tr> <tr> <td>消去中 (Chip Erase)</td> <td>FLASH_T_EBUSY_WAIT</td> <td>FLASH_CE_BUSY_WAIT</td> </tr> </tbody> </table>	状態	BusyTime	BusyCnt	レジスタ書き込み中	FLASH_T_WBUSY_WAIT	FLASH_WR_BUSY_WAIT	データ書き込み中	FLASH_T_PBUSY_WAIT	FLASH_PP_BUSY_WAIT	消去中 (Sector Erase 4KB)	FLASH_T_EBUSY_WAIT	FLASH_SE_BUSY_WAIT	消去中 (Block Erase 32KB)	FLASH_T_EBUSY_WAIT	FLASH_BE32K_BUSY_WAIT	消去中 (Block Erase 64KB)	FLASH_T_EBUSY_WAIT	FLASH_BE64K_BUSY_WAIT	消去中 (Chip Erase)	FLASH_T_EBUSY_WAIT	FLASH_CE_BUSY_WAIT
状態	BusyTime	BusyCnt																				
レジスタ書き込み中	FLASH_T_WBUSY_WAIT	FLASH_WR_BUSY_WAIT																				
データ書き込み中	FLASH_T_PBUSY_WAIT	FLASH_PP_BUSY_WAIT																				
消去中 (Sector Erase 4KB)	FLASH_T_EBUSY_WAIT	FLASH_SE_BUSY_WAIT																				
消去中 (Block Erase 32KB)	FLASH_T_EBUSY_WAIT	FLASH_BE32K_BUSY_WAIT																				
消去中 (Block Erase 64KB)	FLASH_T_EBUSY_WAIT	FLASH_BE64K_BUSY_WAIT																				
消去中 (Chip Erase)	FLASH_T_EBUSY_WAIT	FLASH_CE_BUSY_WAIT																				
引数	uint8_t DevNo ; デバイス番号 uint16_t BusyTime ; ウェイト時間 (以下より1つ選択) FLASH_T_WBUSY_WAIT : レジスタ書き込み時 FLASH_T_PBUSY_WAIT : データ書き込み時 FLASH_T_EBUSY_WAIT : 消去時 uint32_t BusyCnt ; カウンタ (以下より1つ選択) FLASH_WR_BUSY_WAIT : レジスタ書き込み時 FLASH_PP_BUSY_WAIT : データ書き込み時 FLASH_SE_BUSY_WAIT : 消去時 (Sector Erase 4KB) FLASH_BE32K_BUSY_WAIT : 消去時 (Block Erase 32KB) FLASH_BE64K_BUSY_WAIT : 消去時 (Block Erase 64KB) uint8_t Mode ; 消去時 (Block Erase 64KB)																					

FLASH_CE_BUSY_WAIT : 消去時 (Chip Erase)
 ; ウェイトモード (以下より1つ選択)
 FLASH_MODE_REG_WRITE : レジスタ書き込み時
 FLASH_MODE_PROG_ERASE :
 データ書き込み及び消去時

リターン値 ・ウェイト結果を返します。
 FLASH_OK ; Successful operation
 FLASH_ERR_PARAM ; Parameter error
 FLASH_ERR_HARD ; Hardware error
 FLASH_ERR_TIMEOUT ; Timeout error (BusyCnt≠0 の場合)
 FLASH_ERR_OTHER ; Other error

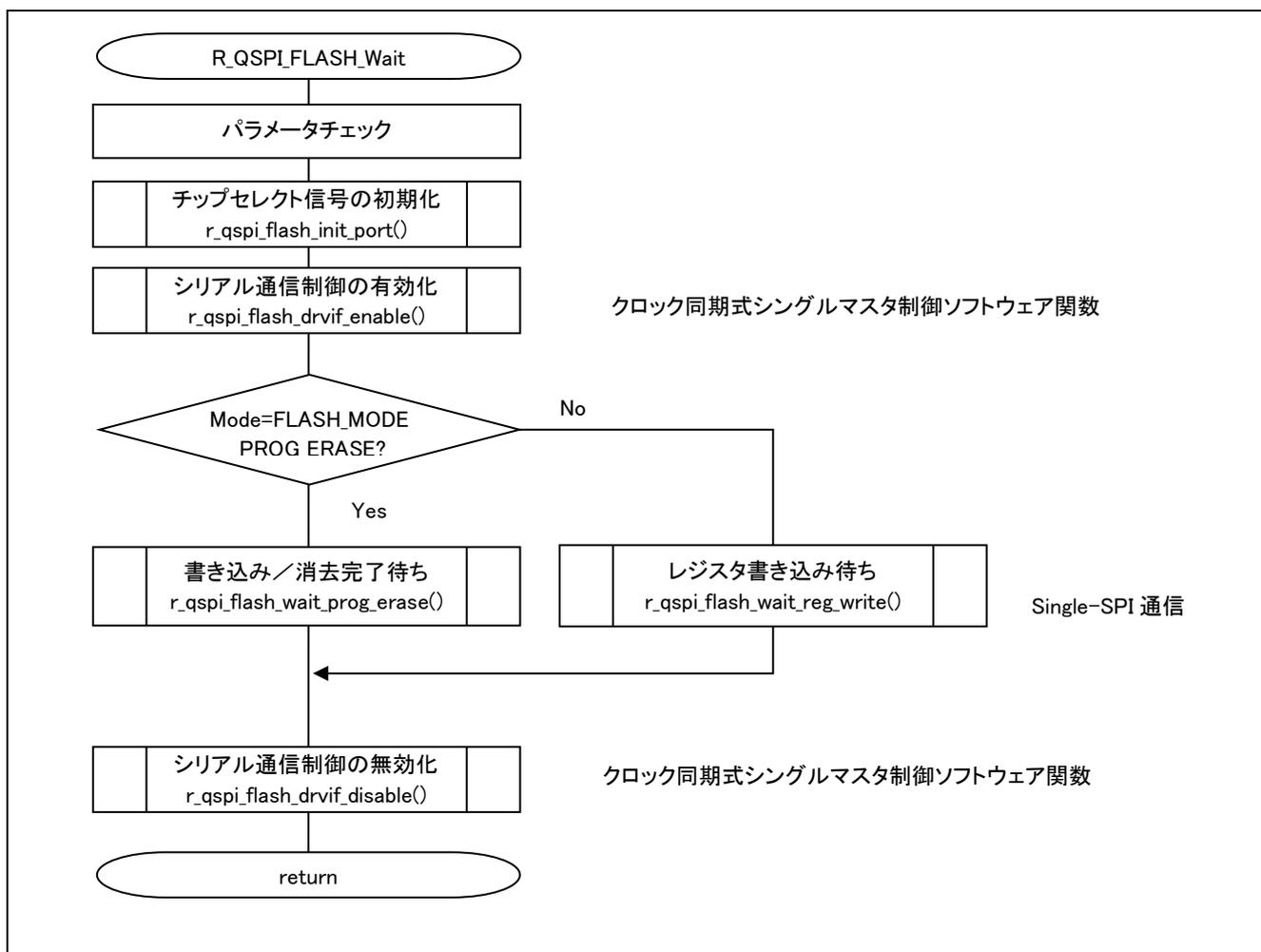


図 5-28 ビジーウェイト処理概要

5.9.16 4 バイトアドレスモード設定処理

R_QSPI_FLASH_Set_4byte_Address_Mode	
概要	4 バイトアドレスモード設定処理
ヘッダ	r_qspi_flash_mx25l.h, r_qspi_flash_mx25l_sub.h, r_qspi_flash_mx25l_sfr.h, r_qspi_flash_mx25l_drvif.h
宣言	error_t R_QSPI_FLASH_Set_4byte_Address_Mode (uint8_t DevNo)
説明	<ul style="list-style-type: none"> ・ r_qspi_flash_enter_4addr()関数をコールし、アドレッシングを 4-byte アドレスモードに設定します。 ・ 処理終了後、コンフィギュレーションレジスタを読み出して、4BYTE bit を確認してください。 ・ システム起動時に一度だけ、R_QSPI_FLASH_Init_Driver()関数をコールした後に呼び出してください。
引数	uint8_t DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> ・ アドレスモード設定結果を返します。 FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

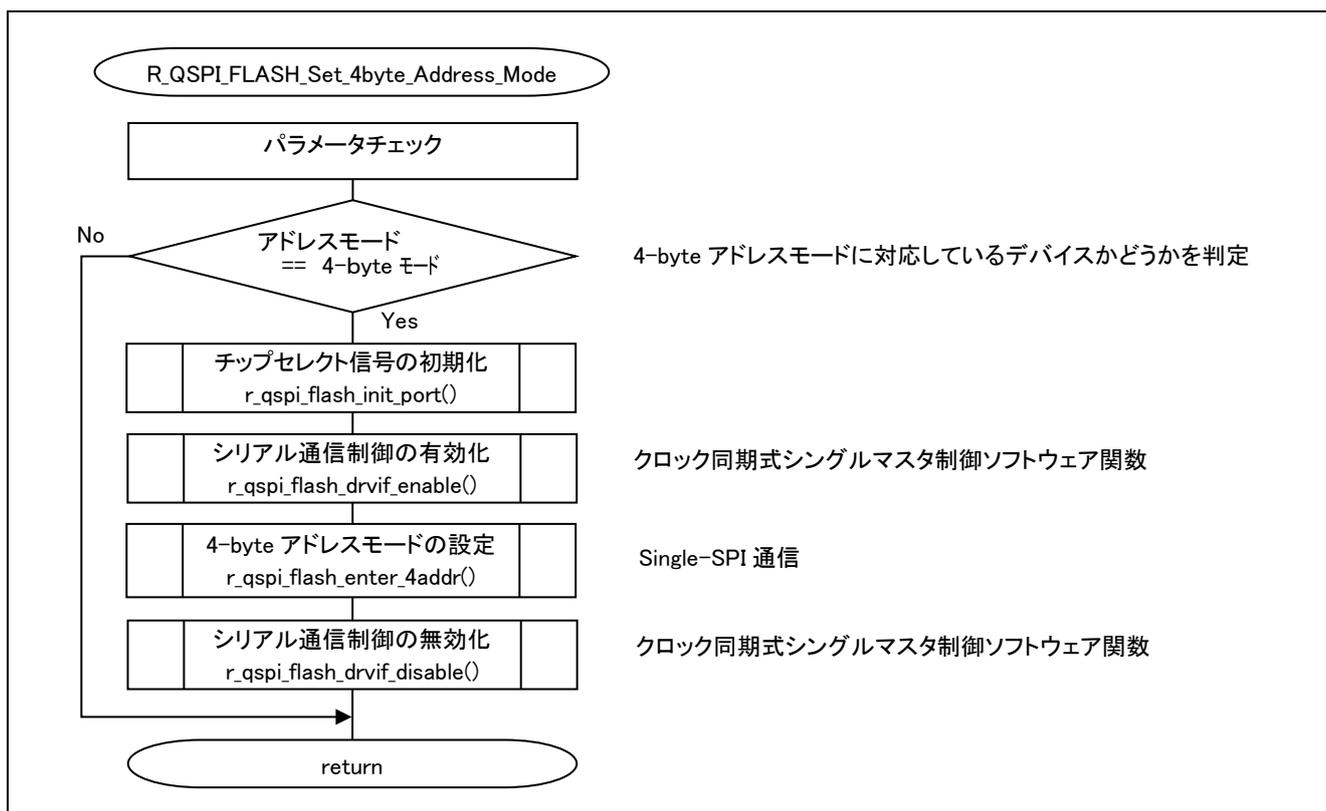


図 5-29 アドレスモード設定処理概要

6. 応用例

Serial NOR Flash memory 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 NOR Flash memory 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_qspi_flash_mx25l.h

本 Serial NOR Flash memory 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 Flash memory. (1~N devices) */
/* Define the device number in accordance with the number of Flash memory */
/* to be connected. */
/*-----*/
/* Define number of devices */
#define FLASH_DEV_NUM          (1)                /* 1device */

/* Define No. of slots */
#define FLASH_DEV0              (0)                /* Device 0 */
/*-----*/
#define FLASH_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、32Mbit デバイスを使用する場合の例です。

```

/*-----*/
/* Define the serial Flash memory. */
/*-----*/

/* #define MX25R1635F */                /* 16Mbit ( 2MByte) */
#define MX25L3235E                    /* 32Mbit ( 4MByte) */
/* #define MX25L3233F */                /* 32Mbit ( 4MByte) */
/* #define MX25L6435E */                /* 64Mbit ( 8MByte) */
/* #define MX25L12835F */               /* 128Mbit ( 16MByte) */
/* #define MX25L25635F */               /* 256Mbit ( 32MByte) */
/* #define MX66L51235F */               /* 512Mbit ( 64MByte) */
/* #define MX25L51245G */               /* 512Mbit ( 64MByte) */
/* #define MX66L1G45G */                /* 1Gbit (128MByte) */

```

(3) Delay task のウェイト時間設定 (OS 制御設定時、有効)

OS 制御 (※) の Delay task のウェイト時間を設定します。単位は ms です。

下記は、1ms を設定する場合の例です。

```
/*----- Definitions of delay task wait time -----*/
#define FLASH_DELAY_TASK    (uint8_t)(1)    /* OS delay task wait time (Unit:ms)
*/
```

※ : 本サンプルコードの OS 制御は、 μ ITRON4.0 を想定しています。

(4) 書き込み待ち/消去完了待ち処理の組み込み設定

以下の関数内でコマンド実行後の完了を待つ設定が可能です。完了を待つ設定にする場合、設定を有効にしてください。

対象関数 :

コンフィグレーションレジスタ書き込み処理 (R_QSPI_FLASH_Write_Configuration())

ライトプロテクト設定処理 (R_QSPI_FLASH_Set_Write_Protect())

Quad モード許可設定処理 (R_QSPI_FLASH_Quad_Enable())

Quad モード禁止設定処理 (R_QSPI_FLASH_Quad_Disable())

データ書き込み処理 (1Page 書き込み用) (R_QSPI_FLASH_Write_Data_Page())

消去処理 (R_QSPI_FLASH_Erase())

下記は、完了待ち設定をする場合の例です。

```
/*----- Definitions of using wait -----*/
/* When you wait completion a Flash memory writing or erasing, please define
it.*/
#define FLASH_WAIT_READY
```

6.1.2 r_qspi_flash_mx25l_sfr.h

r_qspi_flash_mx25l_sfr.h.XXX は、MCU 毎に作成したものです。どれか一つを r_qspi_flash_mx25l_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、以下の情報を参照して、r_qspi_flash_mx25l_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の SFR 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、RX111 でポート 55 を使用する場合での記述例を示しています。

```

/*-----*/
/*  Define the CS port.                               */
/*-----*/
#define FLASH_DR_CS0    PORT5.PODR.BIT.B5    /* FLASH CS0 (Negative-true
logic)*/
#define FLASH_DDR_CS0   PORT5.PDR.BIT.B5     /* FLASH CS0 (Negative-true
logic)*/

#if (FLASH_DEV_NUM > 1)
#define FLASH_DR_CS1    /* FLASH CS1 (Negative-true logic)*/
#define FLASH_DDR_CS1   /* FLASH CS1 (Negative-true logic)*/
#endif /* #if (FLASH_DEV_NUM > 1) */

```

以下は、RL78/G14 でポート 80 を使用する場合での記述例を示しています。

```

/*-----*/
/* Define the CS port. */
/*-----*/
#ifdef __CA78K0R__ /* Renesas RL78 Compiler */
#define FLASH_DR_CS0 P8.0 /* FLASH CS0 (Negative-true logic) */
#define FLASH_DDR_CS0 PM8.0 /* FLASH CS0 (Negative-true logic) */

#if (FLASH_DEV_NUM > 1)
#define FLASH_DR_CS1 /* FLASH CS1 (Negative-true logic) */
#define FLASH_DDR_CS1 /* FLASH CS1 (Negative-true logic) */
#endif /* #if (FLASH_DEV_NUM > 1) */
#endif /* __CA78K0R__ */
#ifdef __CCRL__ /* Renesas CCRL Compiler */
#define FLASH_DR_CS0 P8_bit.no0 /* FLASH CS0 (Negative-true logic) */
#define FLASH_DDR_CS0 PM8_bit.no0 /* FLASH CS0 (Negative-true logic) */

#if (FLASH_DEV_NUM > 1)
#define FLASH_DR_CS1 /* FLASH CS1 (Negative-true logic) */
#define FLASH_DDR_CS1 /* FLASH CS1 (Negative-true logic) */
#endif /* #if (FLASH_DEV_NUM > 1) */
#endif /* __CCRL__ */
#ifdef __ICCRL78__ /* IAR RL78 Compiler */
#define FLASH_DR_CS0 P8_bit.no0 /* FLASH CS0 (Negative-true logic) */
#define FLASH_DDR_CS0 PM8_bit.no0 /* FLASH CS0 (Negative-true logic) */

#if (FLASH_DEV_NUM > 1)
#define FLASH_DR_CS1 /* FLASH CS1 (Negative-true logic) */
#define FLASH_DDR_CS1 /* FLASH CS1 (Negative-true logic) */
#endif /* #if (FLASH_DEV_NUM > 1) */
#endif /* __ICCRL78__ */

```

(2) 通信クロック周波数の設定

通信クロック周波数を設定してください。

設定値は、使用する MCU とシリアル I/O に依存します。通信する用途により設定を分けています。詳しくは表 6-1 を参照ください。

表 6-1 通信クロック周波数設定

#define 定義	用途
FLASH_BR	下記 2 項以外の通信処理（コマンド送信等）
FLASH_BR_WRITE_DATA	データ書き込み処理
FLASH_BR_READ_DATA	データ読み出し処理

以下は、RX111 の RSPI を使用する場合の例です。

```

    /* PCLK = 32MHz, n=0 for RX111 RSPI */
#define FLASH_BR          (uint8_t)(0x00) /* SPBR initial setting
*/
                                /*      ++----- 16.00MHz
*/

    /* PCLK = 32MHz, n=0 for RX111 RSPI Write Data */
#define FLASH_BR_WRITE_DATA (uint8_t)(0x00) /* SPBR initial setting
*/
                                /*      ++----- 16.00MHz
*/

    /* PCLK = 32MHz, n=0 for RX111 RSPI Read Data */
#define FLASH_BR_READ_DATA  (uint8_t)(0x00) /* SPBR initial setting
*/
                                /*      ++----- 16.00MHz
*/

```

以下は、RL78/G14 の CSI を使用する場合の例です。

```

    /* fMCK = 24MHz for RL78 CSI */
#define FLASH_BR          (uint8_t)(0x01) /* SDR[15:9] initial setting */
                                /*      ++----- 6.00MHz
*/

    /* fMCK = 24MHz for RL78 CSI Write Data */
#define FLASH_BR_WRITE_DATA (uint8_t)(0x01) /* SDR[15:9] initial setting */
                                /*      ++----- 6.00MHz
*/

    /* fMCK = 24MHz for RL78 CSI Read Data */
#define FLASH_BR_READ_DATA  (uint8_t)(0x01) /* SDR[15:9] initial setting */
                                /*      ++----- 6.00MHz
*/

```

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_qspi_flash_mx25l_sub.h

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) 消去タイムアウト時間設定

指定されたセクタの全データ消去 (Sector Erase)、指定されたブロックの全データ消去 (Block Erase : 32KB ブロック、64KB ブロック)、指定されたチップの全データ消去 (Chip Erase) のタイムアウト時間を設定します。

デバイスにより、最大時間が異なる場合、以下の設定を見直してください。

```

/*----- Definitions of software timer value -----*/
/* Write Status/Configuration Register
   : 30ms - MX25R1635F
   */
/*
   : 40ms - MX25L3233F, MX25L3235E, MX25L6435E,
*/
/*
   : 40ms - MX25L12835F, MX25L25635F, MX66L51235F,
*/
/*
   : 40ms - MX25L51245G, MX66L1G45G
*/
/* Sector Erase (4KB) : 240ms - MX25R1635F */
/*
   : 200ms - MX25L3235E, MX25L3233F */
/*
   : 300ms - MX25L6435E
*/
/*
   : 120ms - MX25L12835F, MX25L25635F, MX66L51235F
*/
/*
   : 400ms - MX25L51245G, MX66L1G45G
*/
/* Block Erase (32KB) : 3s - MX25R1635F
*/
/*
   : 1.6s - MX25L3235E
*/
/*
   : 0.6s - MX25L3233F
*/
/*
   : 2s - MX25L6435E
*/
/*
   : 0.65s - MX25L12835F, MX25L25635F, MX66L51235F
*/
/*
   : 1s - MX25L51245G, MX66L1G45G
*/
/* Block Erase (64KB) : 3.5s - MX25R1635F
*/
/*
   : 2s - MX25L3235E, MX25L6435E
*/
/*
   : 1s - MX25L3233F
*/
/*
   : 0.65s - MX25L12835F, MX25L25635F, MX66L51235F
*/
/*
   : 2s - MX25L51245G, MX66L1G45G
*/
/* Chip Erase
   : 60s - MX25R1635F
*/
/*
   : 50s - MX25L3235E
*/
/*
   : 30s - MX25L3233F
*/
/*
   : 80s - MX25L6435E, MX25L12835F
*/

```

```

/*          : 150s - MX25L25635F
*/
/*          : 300s - MX66L51235F
*/
/*          : 200s - MX25L51245G
*/
/*          : 600s - MX66L1G45G
*/
/* Page (256 bytes) Program : 10ms - MX25R1635F
*/
/*          : 3ms - MX25L3235E
*/
/*          : 1.2ms - MX25L3233F
*/
/*          : 5ms - MX25L6435E
*/
/*          : 1.5ms - MX25L12835F, MX25L25635F, MX66L51235F
*/
/*          : 0.75ms - MX25L51245G
*/
/*          : 3ms - MX66L1G45G
*/

#define FLASH_SE_BUSY_WAIT      (uint32_t) (200)
/* Sector Erase busy timeout      200*1ms = 0.2s
*/
#define FLASH_BE32K_BUSY_WAIT   (uint32_t) (1000)
/* Block Erase (32KB) busy timeout 1,000*1ms = 1s
*/
#define FLASH_BE64K_BUSY_WAIT   (uint32_t) (2000)
/* Block Erase (64KB) busy timeout 2,000*1ms = 2s
*/
#define FLASH_CE_BUSY_WAIT      (uint32_t) (1200000)
/* Chip Erase busy timeout      1,200,000*1ms =1200s
*/
#define FLASH_PP_BUSY_WAIT      (uint32_t) (3000)
/* Page Program timeout        3,000*1us = 3ms
*/
#define FLASH_WR_BUSY_WAIT      (uint32_t) (40000)
/* Write Register timeout      40,000*1us = 40ms
*/

```

(2) 書き込みタイムアウト時間設定

デバイスにより、書き込み時間が異なる場合、以下の設定を見直してください。

```
#define FLASH_PP_BUSY_WAIT      (uint32_t) (3000)
                                /* Page Program timeout          3000*1us = 3ms */
#define FLASH_WR_BUSY_WAIT     (uint32_t) (40000)
                                /* Write Register timeout       40000*1us = 40ms */
```

6.1.4 r_qspi_flash_mx25l_sub.c

本 Serial NOR Flash memory 用の内部関数ソースファイルです。

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) マクロ関数 R_QSPI_FLASH_CMD_READ()の定義

読み出し処理の動作コマンドを設定します。下表より1つ定義してください。

表 6-2 マクロ関数 R_QSPI_FLASH_CMD_READ()定義

No.	#define 定義値	データシート上の命令コード	処理内容
1	r_qspi_flash_send_cmd(FLASH_CMD_FREAD ,(uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE+1)	FAST READ	Single-SPI 読み出し (高速)
2	r_qspi_flash_send_cmd(FLASH_CMD_DREAD ,(uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE+1)	DUAL OUTPUT READ	Dual-SPI 読み出し (高速)
3	r_qspi_flash_send_cmd(FLASH_CMD_QREAD ,(uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE+1)	QUAD OUTPUT READ	Quad-SPI 読み出し (高速)

(2) マクロ関数 R_QSPI_FLASH_CMD_PP()の設定

書き込み処理の動作コマンドを設定します。下表より1つ定義してください。

表 6-3 マクロ関数 R_QSPI_FLASH_CMD_PP()定義

No.	#define 定義値	データシート上の命令コード	処理内容
1	r_qspi_flash_send_cmd(FLASH_CMD_PP , (uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE)	PAGE PROGRAM	Single-SPI 書き込み
2	r_qspi_flash_send_cmd(FLASH_CMD_4PP , (uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE)	4 x I/O PAGE PROGRAM	Quad-SPI 書き込み

6.1.5 r_qspi_flash_mx25l_drvif.c

本 Serial NOR Flash memory 用のクロック同期式シングル制御ソフトウェア I/F ソースファイルです。

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) r_qspi_flash_drvif_init_driver()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのドライバ初期化処理を設定します。

対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_init_driver(void)
{
    return R_SIO_Init_Driver();
}
```

(2) r_qspi_flash_drvif_disable()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO 禁止設定処理を設定します。

対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_disable(void)
{
    return R_SIO_Disable();
}
```

(3) r_qspi_flash_drvif_enable()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO 許可設定処理を設定します。

引数 BrgData は、ビットレートレジスタに設定する値です。

対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_enable(uint8_t BrgData)
{
    return R_SIO_Enable(BrgData);
}
```

(4) r_qspi_flash_drvif_enable_tx_data()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのデータ書き込み専用のシリアル IO 許可設定処理を設定します。

引数 BrgData は、ビットレートレジスタに設定する値です。

対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_enable_tx_data(uint8_t BrgData)
{
    return R_SIO_Enable(BrgData);
}
```

(5) r_qspi_flash_drvif_enable_rx_data()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのデータ読み出し専用のシリアル IO 許可設定処理を設定します。

引数 BrgData は、ビットレートレジスタに設定する値です。

対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_enable_rx_data(uint8_t BrgData)
{
    return R_SIO_Enable(BrgData);
}
```

(6) r_qspi_flash_drvif_open_port()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO 開放設定処理を設定します。

対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_open_port(void)
{
    return R_SIO_Open_Port();
}
```

(7) r_qspi_flash_drvif_tx()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO データ送信処理を設定します。主にコマンド送信やステータスレジスタへの書き込みに使用します。

引数 TxCnt は送信データサイズ（バイト）、引数 pData は送信データの格納先バッファアドレスです。対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_tx(uint16_t TxCnt, uint8_t FAR * pData)
{
    return R_SIO_Tx_Data(TxCnt, pData);
}
```

(8) r_qspi_flash_drvif_tx_add()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO データ送信処理を設定します。

r_qspi_flash_drvif_tx()と別に設定が必要な場合を想定して追加したものです。

本 Serial NOR Flash memory の場合、アドレス送信に使用します。

引数 TxCnt は送信データサイズ（バイト）、引数 pData は送信データの格納先バッファアドレスです。対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_tx_add(uint16_t TxCnt, uint8_t FAR * pData)
{
    return R_SIO_Tx_Data(TxCnt, pData);
}
```

(9) r_qspi_flash_drvif_tx_data()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのデータ書き込み専用のデータ送信シリアル IO データ送信処理を設定します。主にデータ書き込みに使用します。

引数 TxCnt は送信データサイズ（バイト）、引数 pData は送信データの格納先バッファアドレスです。対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_tx_data(uint16_t TxCnt, uint8_t FAR * pData)
{
    return R_SIO_Tx_Data(TxCnt, pData);
}
```

(10) r_qspi_flash_drvif_rx()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO データ受信処理を設定します。主にステータスレジスタの読み出しに使用します。

引数 RxCnt は受信データサイズ (バイト)、引数 pData は受信データの格納先バッファアドレスです。対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_rx(uint16_t RxCnt, uint8_t FAR * pData)
{
    return R_SIO_Rx_Data(RxCnt, pData);
}
```

(11) r_qspi_flash_drvif_rx_add()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのシリアル IO データ受信処理を設定します。r_qspi_flash_drvif_rx()と別に設定が必要な場合を想定して追加したものです。

本 Serial NOR Flash memory の場合は、使用していません。

引数 RxCnt は受信データサイズ (バイト)、引数 pData は受信データの格納先バッファアドレスです。対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_rx_add(uint16_t RxCnt, uint8_t FAR * pData)
{
    return R_SIO_Rx_Data(RxCnt, pData);
}
```

(12) r_qspi_flash_drvif_rx_data()の設定

使用するクロック同期式シングルマスタ制御ソフトウェアのデータ読み出し専用のデータ送信シリアル IO データ送信処理を設定します。

引数 RxCnt は受信データサイズ (バイト)、引数 pData は受信データの格納先バッファアドレスです。対応のものが無い場合、追加してください。

```
error_t r_qspi_flash_drvif_rx_data(uint16_t RxCnt, uint8_t FAR * pData)
{
    return R_SIO_Rx_Data(RxCnt, pData);
}
```

6.1.6 r_qspi_flash_mx25l_sfr_rl78.c

本プログラムは RL78 用の SFR モジュールファイルです。

r_qspi_flash_mx25l_sfr_rl78XXX.c は、MCU 毎に作成したものです。どれか一つをビルド対象としてください。対象 MCU のものが無い場合には、以下の情報を参照して、r_qspi_flash_mx25l_sfr_rl78.c 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) 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、スレーブデバイスセレクト制御信号用に使用します。

表 6-4 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)
e ² studio	RL78	不要	不要
	78K0R	(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)
	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、以下のヘッダファイルをインクルードしてください。

```
r_qspi_flash_mx25l.h  
r_qspi_flash_mx25l_sub.h  
r_qspi_flash_mx25l_sfr.h  
r_qspi_flash_mx25l_drvif.h
```

e² studio (CC-RL コンパイラ) でスマート・コンフィグレータを使用する場合は、iodefine.h をインクルードしてください。RL78/G23 の場合、パスは以下になります。

```
"/src/smc_gen/r_bsp/mcu/rl78_g23/register_access/ccrl"
```

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 他容量に対応する場合

他容量に対応する場合、以下の定義の見直しが必要です。

```
FLASH_MEM_SIZE  
FLASH_SECT_ADDR  
FLASH_B32K_ADDR  
FLASH_B64K_ADDR  
FLASH_PAGE_SIZE  
FLASH_ADDR_SIZE  
FLASH_WP_WHOLE_MEM  
FLASH_FULL_CHIP_ERASE  
FLASH_ADDR_MODE
```

上記以外の定義の見直しが必要になる可能性がありますので、メモリのデータシートを入手し、定義を見直してください。

7.4 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。

7.5 電源投入後の電圧安定待ち時間について

電源投入後、電圧が安定するまでの間、十分に時間を待ってから初期化関数をコールしてください。

電源投入後の電圧安定待ち時間については、スレーブデバイスのデータシートをご確認ください。

RX ファミリ、RL78 ファミリ、78K0R/Kx3-L

Macronix International 社製 MX25/66L family serial NOR Flash Memory 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/>

お問合せ先

<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	2014.7.29	—	新規作成
1.01	2015.2.28	1	動作確認デバイス RX600 シリーズ において、RX64M (SCI を使用) 元は、RX64M (QSPI を使用) であった。
		6	2.1 RX ファミリ (3) RX64M RSPI の場合 表 2-3 動作確認条件 の評価に使用したソフトウェアの内容欄を更新した。
		6	2.1 RX ファミリ (4) RX64M QSPI の場合 を削除した。
		6	2.1 RX ファミリ (4) RX64M SCI の場合 を追加した。
		12	3.1 RX ファミリ関連アプリケーションノート一覧 において 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) “RX64M” グループを追加した。
		12	3.1 RX ファミリ関連アプリケーションノート一覧 において 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) “RX64M” グループを追加した。
		12	3.1 RX ファミリ関連アプリケーションノート一覧 において “RX ファミリ RSPI クロック同期式シングルマスタ制御モジュール Firmware Integration Technology (R01AN1914JJ)” を削除した。
		12	3.1 RX ファミリ関連アプリケーションノート一覧 において “RX ファミリ QSPI クロック同期式シングルマスタ制御モジュール Firmware Integration Technology (R01AN1940JJ)” を削除した。
		22	5.3.1 RX ファミリ (2) RX64M RSPI の場合 表 5-3 必要メモリサイズ欄を更新した。
		23	5.3.2 RL78 ファミリ、78K0R/Kx3-L (2) RL78/G14 SAU の場合統合開発環境 IAR Embedded Workbench の場合 の表外にて、「プロジェクト全体のスタックサイズです。」を追加した。(誤記修正)
		23	5.3.2 RL78 ファミリ、78K0R/Kx3-L (2) RL78/G14 SAU の場合統合開発環境 IAR Embedded Workbench の場合 の表外にて、「プロジェクト全体のスタックサイズです。」を追加した。(誤記修正)
		24	5.3.2 RL78 ファミリ、78K0R/Kx3-L (4) RL78/L13 SAU の場合統合開発環境 IAR Embedded Workbench の場合 の表外にて、「プロジェクト全体のスタックサイズです。」を追加した。(誤記修正)
		25	5.4 ファイル構成 にて、サンプルコードのフォルダの名前とアプリケーションノート番号を更新した。
		29	5.5.3 各種定義 の表 5-15 qspi_flash_mx25l_sub.h の各種定義値 の FLASH_CNFG_4BYTE の内容 元は、「4Byte Address Mode」であった。(記載方法の変更)
		35	5.9.3 コンフィグレーションレジスタ読み出し処理 の説明欄の Bit 5 元は、「4 BYTE」であった。(記載方法の変更)

		36	5.9.4 コンフィグレーションレジスタ書き込み処理 の説明欄の Bit 5 元は、「4 BYTE」であった。(記載方法の変更) 「4BYTE bit は読み出し専用のため、書き込みはできません。」元は、「4 BYTE bit は読み出し専用のため、書き込みはできません。」であった。(記載方法の変更)
		42	5.9.7 Quad モード許可設定処理 の説明欄の「Quad モードを禁止にする場合、Quad モード禁止設定処理を実行してください。」元は、「Quad モードを禁止にする場合、Quad モード許可設定処理を実行してください。」であった。(誤記修正)
		57	5.9.15 ビジーウェイト処理 の説明欄の「レジスタ書き込みモードの場合」元は、「レジスタ書き込みモード (256Mbit 以下の小容量品) の場合」であった。(誤記修正)
		59	5.9.16 4 バイトアドレスモード設定処理 の図 5-29 アドレスモード設定処理概要 にて、「4-byte モード」元は、「4Byte モード」であった。(記載方法の変更)
		61	6.1.1 r_qspi_flash_mx25l.h の(2)使用するデバイスの容量の定義 にて、「32Mbit デバイス」元は、「256Mbit デバイス」であった。
1.02	2016.3.31	7-15	2. 動作確認条件 2.2 RL78 ファミリ、78K0R/Kx3-L (2) 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。 (5) RL78/G1C 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。 (8) RL78/L12 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。 (11) RL78/L13 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。 (14) RL78/L1C 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。
		28-30	5.3 必要メモリサイズ 5.3.2 RL78 ファミリ、78K0R/Kx3-L (2) 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。 (5) RL78/L13 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。
		32	5.4 ファイル構成 アプリケーションノート番号を更新した。
		71	6.1.2 r_qspi_flash_mx25l_sfr.h 記述例を更新した。
		80	6.1.6 r_qspi_flash_mx25l_sfr_rl78.c 表 6-4 を更新した。
1.03	2021.7.14	1	文書のリンクを最新に変更 動作確認デバイスに MX25R1635F、MX25L3233F を追加 動作確認に使用した MCU に RL78/G23 を追加
		16	2.動作確認条件 に、RL78/G23 の動作確認条件を追加

		17	3.2 RL78 ファミリ、78K0R ファミリ 関連アプリケーション ノート一覧 に記載のタイトルを最新のタイトルに変更
		31	5.3.2 RL78 ファミリ、78K0R/Kx3-L に、RL78/G23 のメモリ サイズを追加
		32	5.4 ファイル構成 に、RL78/G23 のファイルを追加
		68	6.1.1 r_qspi_flash_mx25l.h の (2)使用するデバイスの用量の 定義 に MX25R1635F、MX25L3233F の定義を追加
		73	6.1.3 r_qspi_flash_mx25l_sub.h の (1)消去タイムアウト時間 設定 に MX25R1635F、MX25L3233F の定義を追加
1.04	2022.8.4	18	2.2 RL78 ファミリ、78K0R/Kx3-L (18) RL78/G23 SAU 統合開発環境 e ² studio の場合 (コンパイラ : LLVM) を追加
		35	5.3.2 RL78 ファミリ、78K0R/Kx3-L (9) RL78/G23 SAU 統合開発環境 e ² studio の場合 (コンパイラ : LLVM) を追加
		36	5.4 ファイル構成 サンプルコードのフォルダ名を変更 アプリケーションノート番号を更新
		85	6.1.6 r_qspi_flash_mx25l_sfr_rl78.c (1) SFR 領域用デファインの設定 に e ² studio を追加
		86	7.1 組み込み時の注意事項 e ² studio (CC-RL コンパイラ) でスマート・コンフィグレータを使用する場合の注意事項を追加

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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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くと、フラッシュメモリ、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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、変更、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、変更、複製、リバースエンジニアリング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。
6. 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）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。
10. 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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
13. 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。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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直接的、間接的に支配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。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。