

RX ファミリ、RL78 ファミリ、78K0R/Kx3-L

R01AN1529JJ0104

Rev.1.04

Spansion 社製 S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御ソフトウェア 2016.03.31

要旨

本アプリケーションノートでは、ルネサス エレクトロニクス製 RX ファミリ、RL78 ファミリ、78K0R/Kx3-L を使用した Spansion Inc.社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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 S25FLxxxS MirrorBit® Flash Non-Volatile 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、マスター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。最新のサポート MCU とそ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
http://japan.renesas.com/driver/spi_serial_flash

動作確認デバイス

シリアルフラッシュメモリ Spansion 社製 128Mbit S25FL128S MirrorBit® Flash Non-Volatile Memory
 Spansion 社製 256Mbit S25FL256S MirrorBit® Flash Non-Volatile Memory

動作確認に使用した MCU

RX600 シリーズ	: RX63N (RSPI を使用)
RX100 シリーズ	: RX111 (SCI を使用) : RX111 (RSPI を使用)
RL78/G1x	: RL78/G14, RL78/G1C (SAU を使用)
RL78/L1x	: RL78/L12, RL78/L13, RL78/L1C (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.....	13
3.1 RX ファミリ 関連アプリケーションノート一覧.....	13
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.....	13
4. ハードウェア説明.....	14
4.1 ハードウェア構成例.....	14
4.1.1 Single-SPI 使用時の端子一覧.....	14
4.1.2 Single-SPI 使用時の接続例.....	14
4.1.3 Dual-SPI 使用時の端子一覧.....	15
4.1.4 Dual-SPI 使用時の接続例.....	15
4.1.5 Quad-SPI 使用時の端子一覧.....	16
4.1.6 Quad-SPI 使用時の接続例.....	16
5. ソフトウェア説明.....	17
5.1 動作概要.....	17
5.1.1 データバッファと送信/受信データの関係.....	17
5.1.2 クロック同期式モードで発生させるタイミング.....	18
5.1.3 S25FLxxxS MirrorBit® Flash Non-Volatile Memory の S#端子制御.....	20
5.1.4 S25FLxxxS MirrorBit® Flash Non-Volatile Memory の命令コード.....	21
5.2 ソフトウェア構成.....	22
5.3 必要メモリサイズ.....	23
5.3.1 RX ファミリ.....	23
5.3.2 RL78 ファミリ、78K0R/Kx3-L.....	25
5.4 ファイル構成.....	28
5.5 定数一覧.....	29
5.5.1 戻り値.....	29
5.5.2 コマンド定義.....	29
5.5.3 各種定義.....	30
5.6 構造体/共用体一覧.....	33
5.7 変数一覧.....	34
5.8 関数一覧.....	34
5.9 関数仕様.....	35
5.9.1 ドライバ初期化処理.....	35
5.9.2 デバイス初期化処理.....	36
5.9.3 ステータスレジスタ 1 読み出し処理.....	37
5.9.4 ステータスレジスタ 2 読み出し処理.....	39
5.9.5 ライトプロテクト設定処理.....	40
5.9.6 コンフィグレーションレジスタ読み出し処理.....	44
5.9.7 コンフィグレーションレジスタ書き込み処理.....	46

5.9.8	クリアステータス処理	49
5.9.9	WRDI コマンド発行処理	50
5.9.10	データ読み出し処理	51
5.9.11	データ書き込み処理	52
5.9.12	データ書き込み処理（1Page 書き込み用）	54
5.9.13	消去処理	57
5.9.14	ID 読み出し処理	60
5.9.15	ビジーウェイト処理	61
6.	応用例	63
6.1	S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御ソフトウェアの設定	64
6.1.1	r_qspi_flash_s25fl.h	64
6.1.2	r_qspi_flash_s25fl_sfr.h	66
6.1.3	r_qspi_flash_s25fl_sub.h	69
6.1.4	r_qspi_flash_s25fl_sub.c	70
6.1.5	r_qspi_flash_s25fl_drvif.c	70
6.1.6	r_qspi_flash_s2fl_sfr_rl78.c	73
7.	使用上の注意事項	74
7.1	組み込み時の注意事項	74
7.2	キャッシュ搭載の MCU を使用する場合	74
7.3	他容量に対応する場合	74
7.4	他スレーブデバイスを使用する場合	74
7.5	制御コマンドの使い分けについて	74
7.6	コンフィグレーションレジスタの Bit3（BPNV）とライトプロテクト	75
7.7	ライトプロテクト領域の選択と注意事項	75
7.8	RESET コマンドの注意事項	75
7.9	電源投入後の電圧安定待ち時間について	75

1. 仕様

ルネサス エレクトロニクス製 MCU を使用し、Spansion 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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 に使用する周辺機能と用途を、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 Spansion 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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、SPI モードで制御します。また、ユーザ設定したシリアル I/O 1 チャンネルの使用が可能です。複数チャンネルの使用は、できません。
- 同一型名、かつ最大 2 個の S25FLxxxS MirrorBit® Flash Non-Volatile 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 使用する周辺機能と用途

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

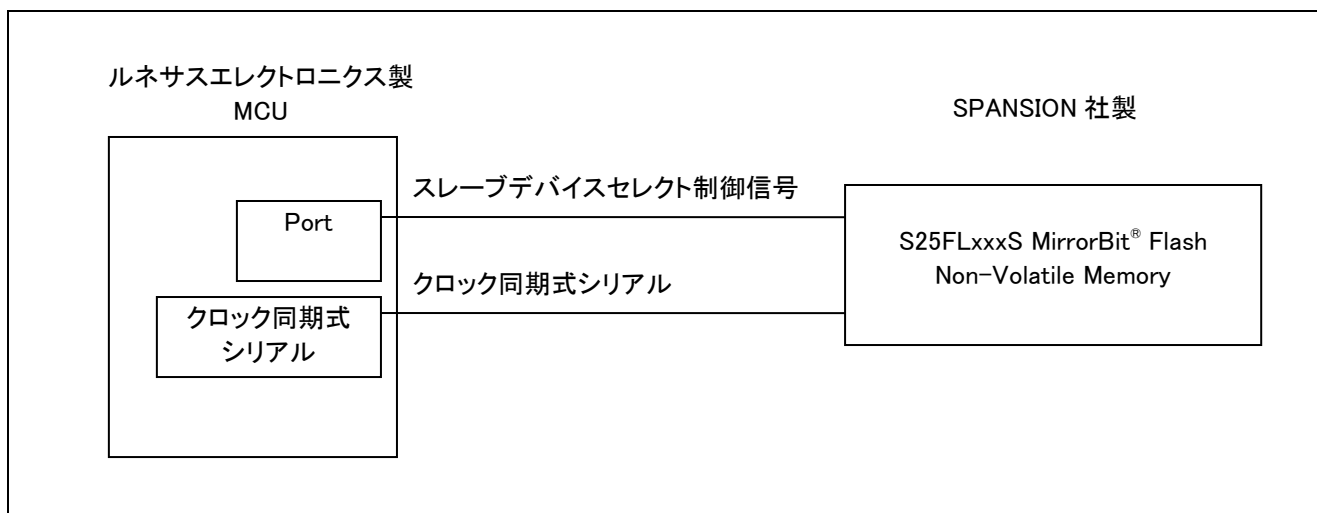


図 1-1 使用例

2. 動作確認条件

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

2.1 RX ファミリ

(1) RX63N 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	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile Memory
評価に使用したマイコン	RX63N グループ (プログラム ROM 1MB/RAM 128KB)
動作周波数	ICLK : 96MHz、PCLK : 48MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 High-performance embedded Workshop Version 4.09.01.007
C コンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 1.2.1.0) コンパイルオプション 統合開発環境のデフォルト設定を使用しています。
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.2.20
評価に使用したソフトウェア	RX63N 用 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
評価に使用した使用ボード	Renesas Starter Kit for RX63N

(2) RX111 RSPI の場合

表 2-2 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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、PCLK : 32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
C コンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 2.01.00) コンパイルオプション 総合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ RSPI を使った クロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) Ver.2.04.R04
評価に使用した使用ボード	Renesas Starter Kit for RX111

(3) RX111 SCI の場合

表 2-3 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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、PCLK : 32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V2.01.00
C コンパイラ	ルネサス エレクトロニクス製 RX ファミリ用 C/C++コンパイラパッケージ (ツールチェーン 2.01.00) コンパイルオプション 総合開発環境のデフォルト設定 (※1) を使用しています。 ※1 : 最適化レベル"2"、最適化方法"サイズ優先"
エンディアン	ビッグエンディアン/リトルエンディアン
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ SCI を使った クロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ) Ver.2.01.R05
評価に使用した使用ボード	Renesas Starter Kit for RX111

2.2 RL78 ファミリ、78K0R/Kx3-L

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

表 2-4 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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 CA,CX V3.01.00
C コンパイラ	ルネサス エレクトロニクス製 RL78,78K0R コンパイラ CA78K0R V1.71 コンパイルオプション 統合開発環境のデフォルト設定 ("-qx2") を使用しています。
サンプルコードのバージョン	Ver.2.22
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

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

表 2-5 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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 V3.03.00
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。
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 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 2-6 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
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

表 2-7 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas RL78/G1C ターゲット・ボード QB-R5F10JGC-TB

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

表 2-8 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas RL78/G1C ターゲット・ボード QB-R5F10JGC-TB

(6) RL78/L12 SAU 統合開発環境 CubeSuite+の場合

表 2-9 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/L12

(7) RL78/L12 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	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/L12

(8) RL78/L13 SAU 統合開発環境 CubeSuite+の場合

表 2-11 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/L13

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

表 2-12 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/L13

(10) RL78/L1C SAU 統合開発環境 CubeSuite+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	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/L1C

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

表 2-14 動作確認条件

項目	内容
評価に使用したメモリ	SPANSION 社製 S25FLxxxS MirrorBit® Flash Non-Volatile 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。
サンプルコードのバージョン	Ver.2.21 R01
評価に使用したソフトウェア	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ0103) Ver.2.03
評価に使用した使用ボード	Renesas Starter Kit for RL78/L1C

3. 関連アプリケーションノート

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

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 グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1196JJ)
- RX210,RX21A,RX220,RX63N,RX63T,RX111 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1229JJ)

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 グループ シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1195JJ)

4. ハードウェア説明

4.1 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。

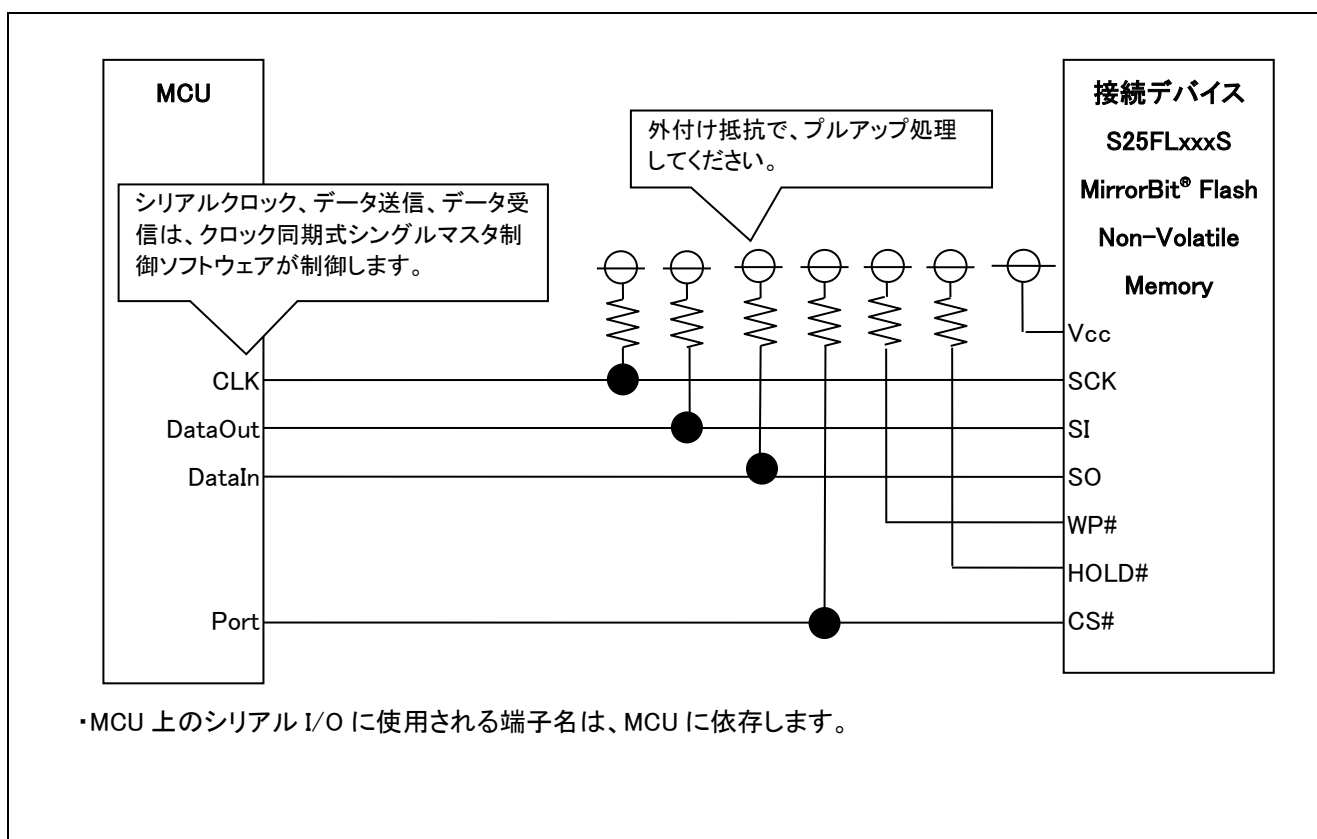


図 4-1 Single-SPI 使用時の MCU と SPI スレーブデバイスの接続例

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 にクラウ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。

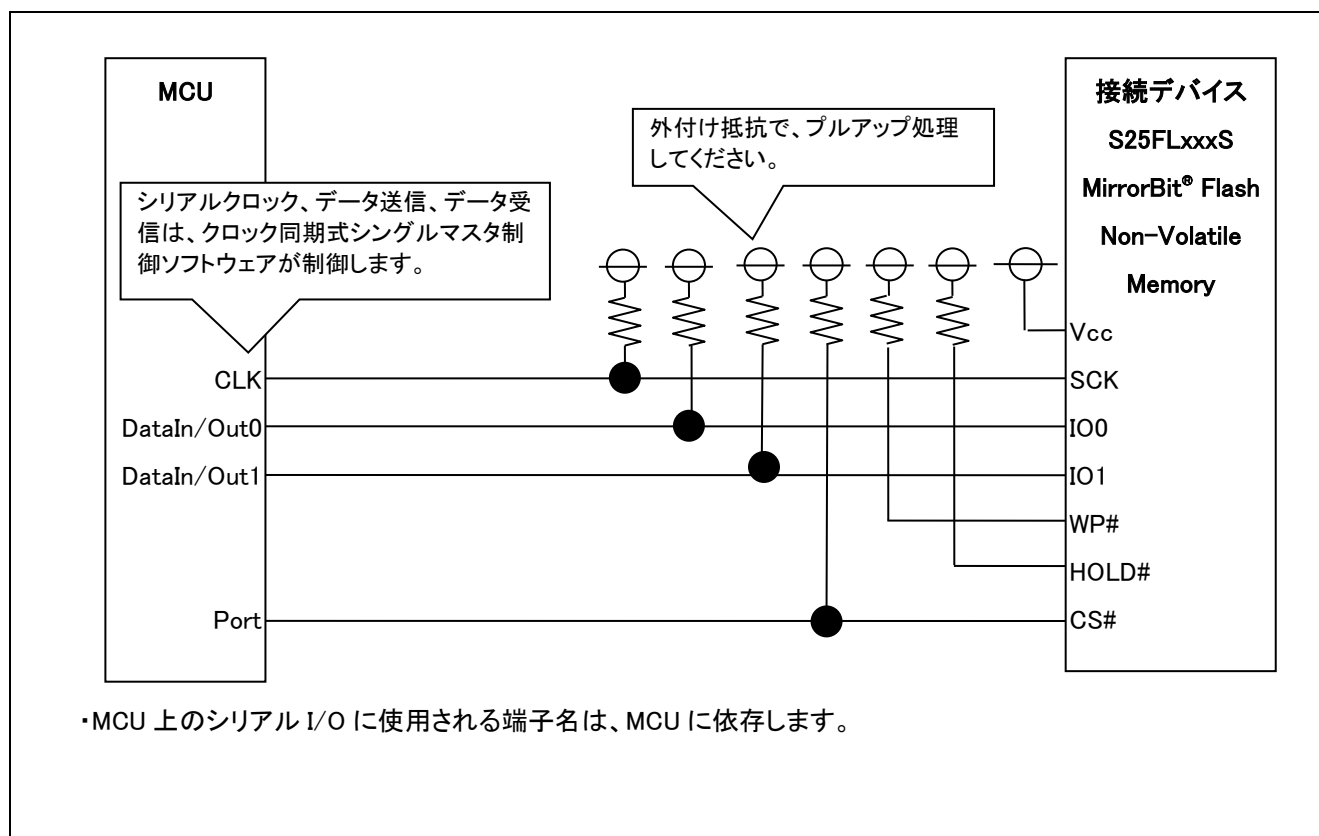


図 4-2 Dual-SPI 使用時の MCU と SPI スレーブデバイスの接続例

4.1.5 Quad-SPI 使用時の端子一覧

表 4-3 に MCU 側の Quad-SPI 使用端子と機能を示します。

なお、Quad-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 4-3 Quad-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
DataIn/Out2	入出力	マスターデータ入出力 2
DataIn/Out3	入出力	マスターデータ入出力 3
Port(CS#)	出力	スレーブデバイスセレクト出力

4.1.6 Quad-SPI 使用時の接続例

以下に Quad-SPI 使用時の接続例を示します。

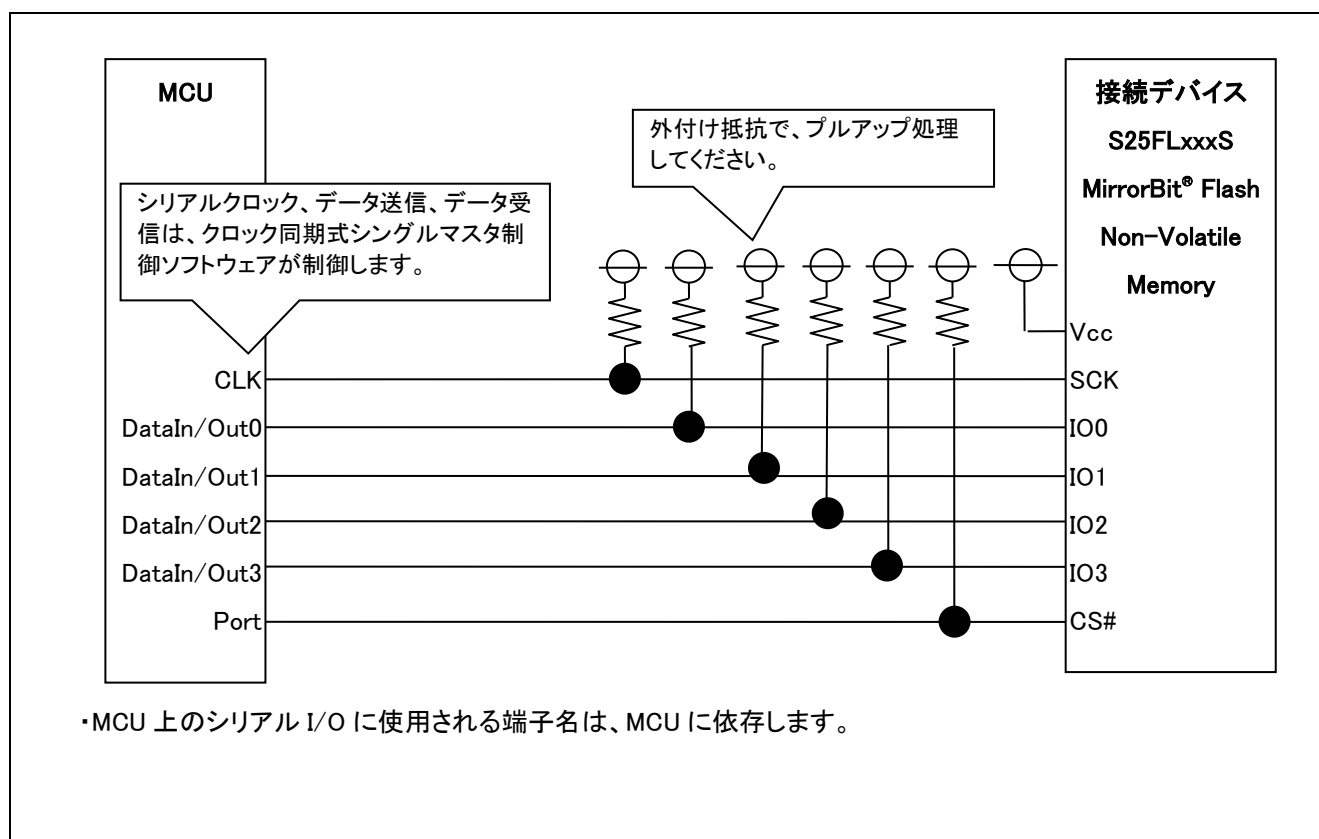


図 4-3 Quad-SPI 使用時の MCU と SPI スレーブデバイスの接続例

5. ソフトウェア説明

5.1 動作概要

MCU のクロック同期式シリアル通信機能を使って、S25FLxxxS MirrorBit® Flash Non-Volatile 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。

- SPI 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。

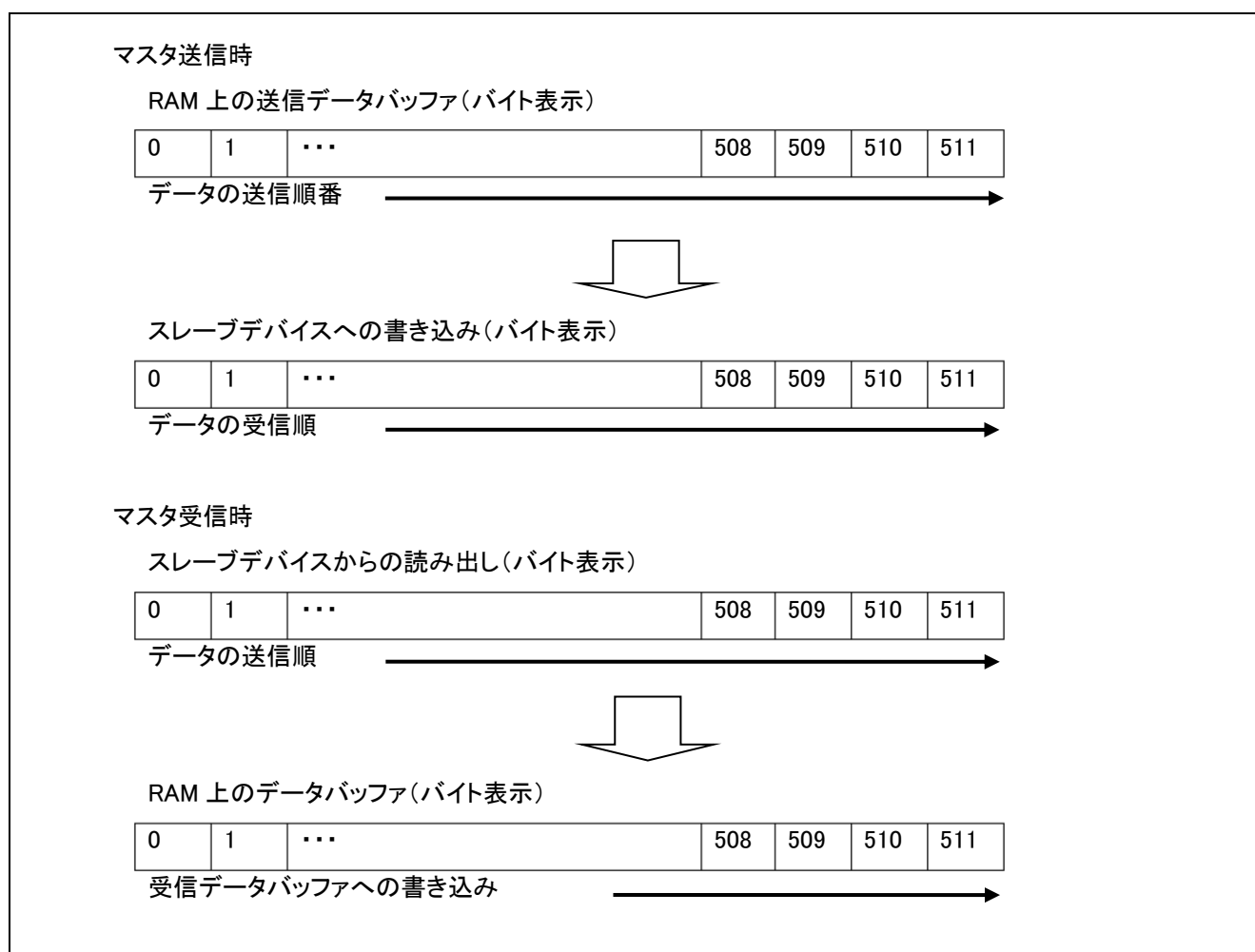


図 5-1 転送データの格納

5.1.2 クロック同期式モードで発生させるタイミング

以下にクロック同期式モードで発生させるタイミングについて示します。

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

(1) Single-SPI 使用時

S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御のため、図 5-2 に示す SPI モード 3 (CPOL=1, CPHA=1) のタイミングを発生させます。

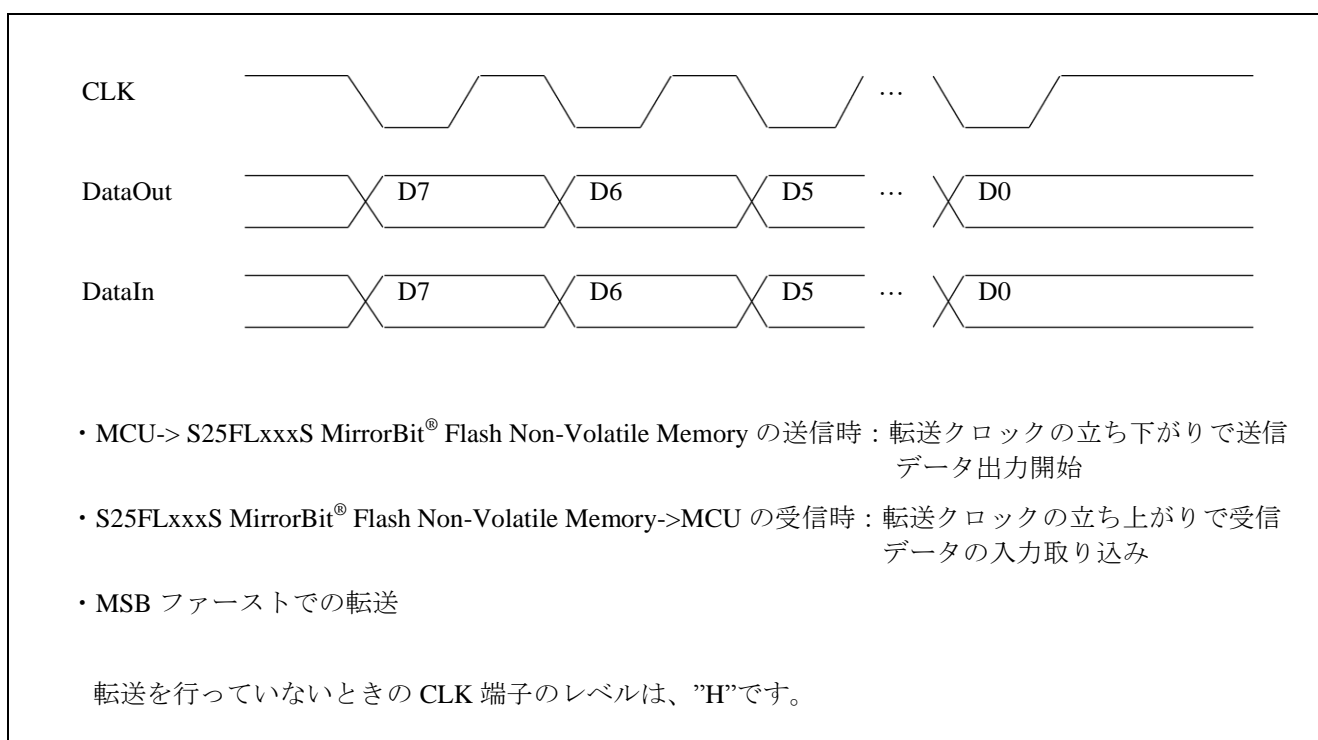


図 5-2 Single-SPI 時のクロック同期式モード タイミング設定

(2) Dual-SPI 使用時

S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御のため、図 5-3 に示す SPI モード 3 (CPOL=1、CPHA=1) のタイミングを発生させます。

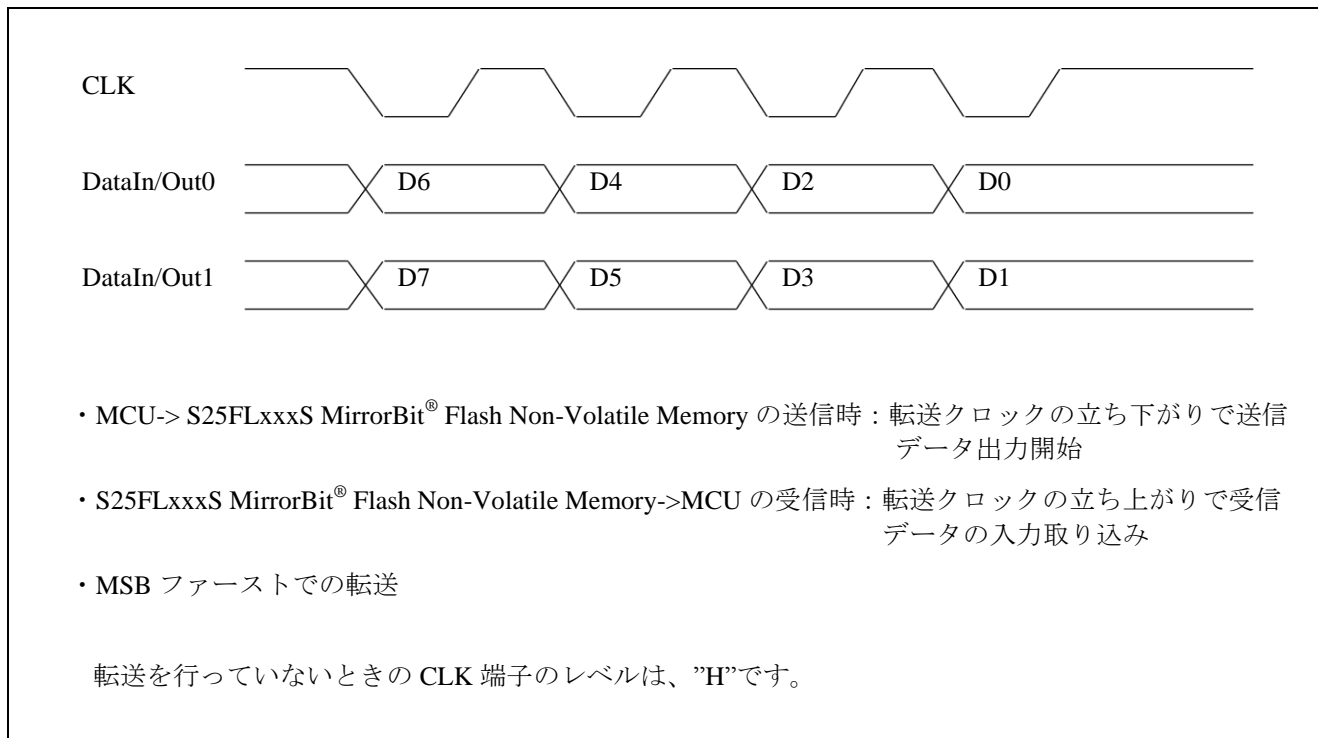


図 5-3 Dual-SPI 時のクロック同期式モード タイミング設定

(3) Quad-SPI 使用時

S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御のため、図 5-4 に示す SPI モード 3 (CPOL=1、CPHA=1) のタイミングを発生させます。

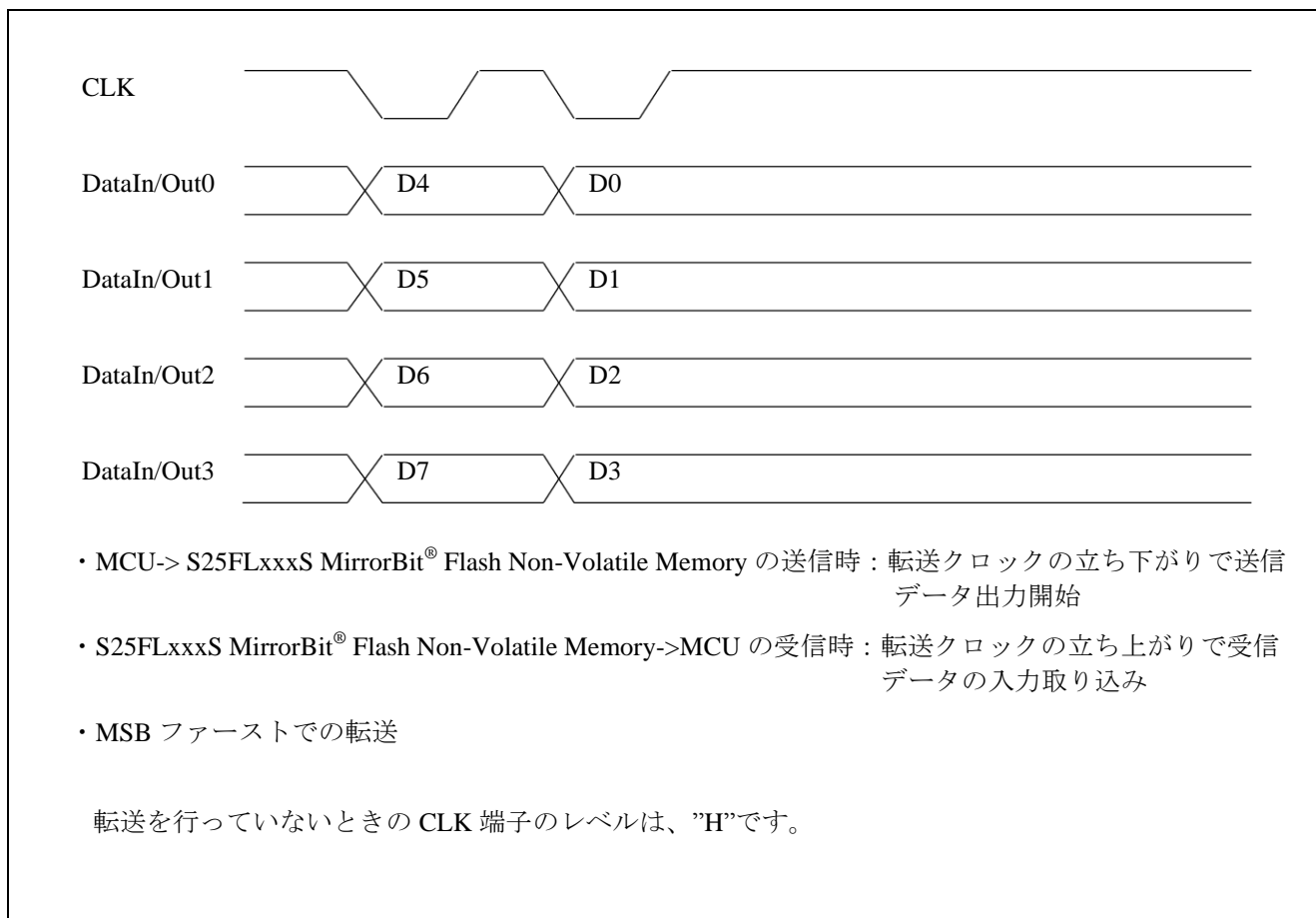


図 5-4 Quad-SPI 時のクロック同期式モード タイミング設定

5.1.3 S25FLxxxS MirrorBit® Flash Non-Volatile Memory の S#端子制御

S25FLxxxS MirrorBit® Flash Non-Volatile 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。

S25FLxxxS MirrorBit® Flash Non-Volatile Memory の CS# (MCU の Port(CS#)) 信号の立ち下がりから、S25FLxxxS MirrorBit® Flash Non-Volatile Memory の SCK (MCU の CLK) 信号の立ち下がりまでの時間は、S25FLxxxS MirrorBit® Flash Non-Volatile 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。

S25FLxxxS MirrorBit® Flash Non-Volatile Memory の SCK (MCU の CLK) 信号の立ち上がりから、S25FLxxxS MirrorBit® Flash Non-Volatile Memory の CS# (MCU の Port(CS#)) 信号の立ち上がりまでの時間は、S25FLxxxS MirrorBit® Flash Non-Volatile 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。

S25FLxxxS MirrorBit® Flash Non-Volatile 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 S25FLxxxS MirrorBit® Flash Non-Volatile Memory の命令コード

S25FLxxxS MirrorBit® Flash Non-Volatile 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)
RDSR1	Read Status Register-1	0000 0101 (05 h)
RDSR2	Read Status Register-2	0000 0111 (07 h)
RDCR	Read Configuration Register-1	0011 0101 (35 h)
WRR	Write Register (Status-1, Configuration-1)	0000 0001 (01 h)
CLSR	Clear Status Register-1 - Erase/Prog. Fail Reset	0011 0000 (30 h)
FAST_READ	Fast Read (3- or 4-byte address)	0000 1011 (0b h)
4FAST_READ	Fast Read (4-byte address)	0000 1100 (0c h)
4DOR	Read Dual Out (4-byte address)	0011 1100 (3c h)
4QOR	Read Quad Out (4-byte address)	0110 1100 (6c h)
PP	Page Program (3- or 4-byte address)	0000 0010 (02 h)
4PP	Page Program (4-byte address)	0001 0010 (12 h)
4QPP	Quad Page Program (4-byte address)	0011 0100 (34 h)
SE	Erase 64 kB or 256 kB (3- or 4-byte address)	1101 1000 (d8 h)
4SE	Erase 64 kB or 256 kB (4-byte address)	1101 1100 (dc h)
P4E	Parameter 4 kB-sector Erase (3- or 4-byte address)	0010 0000 (20 h)
4P4E	Parameter 4 kB-sector Erase (4-byte address)	0010 0001 (21 h)
BE	Bulk Erase	0110 0000 (60 h)
BE	Bulk Erase (alternate command)	1100 0111 (c7 h)
READ_ID (REMS)	Read Electronic Manufacturer Signature	1001 0000 (90 h)
RESET	Software Reset	1111 0000 (f0 h)

5.2 ソフトウェア構成

本サンプルコードは、S25FLxxxS MirrorBit® Flash Non-Volatile 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 S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御ソフトウェア）です。

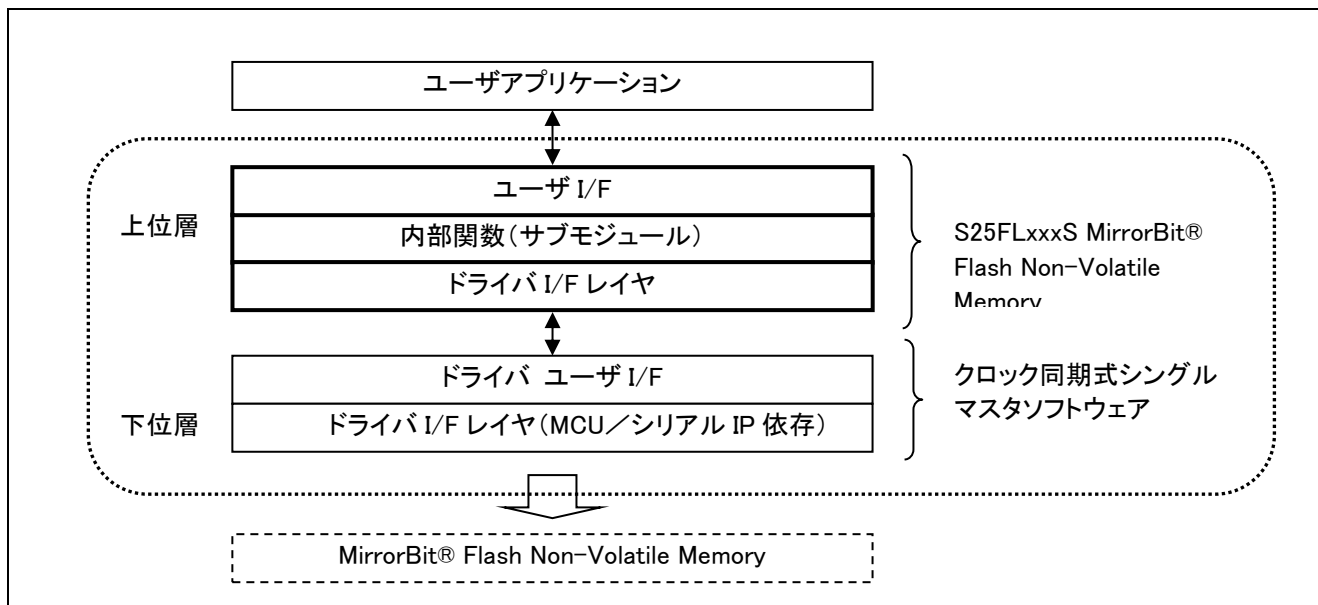


図 5-5 ソフトウェア構成

以下に制御手順を示します。

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

5.3 必要メモリサイズ

表 5-2 に必要メモリサイズを示します。

5.3.1 RX ファミリ

(1) RX63N 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	3,957 バイト (リトルエンディアン)	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c
最大使用ユーザスタック	144 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) RX111 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	3,818 バイト (リトルエンディアン)	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c
最大使用ユーザスタック	152 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。
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) RX111 SCI の場合

表 5-4 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	3,818 バイト (リトルエンディアン)	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c
最大使用ユーザスタック	152 バイト	
最大使用割り込みスタック	—	割り込み未使用

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

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

環境は、「2.動作確認条件」を参照してください。

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

表 5-5 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	7,176 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
最大使用ユーザスタック	122 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 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)

表 5-6 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,131 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
最大使用ユーザスタック	90 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 統合開発環境 IAR Embedded Workbench の場合

表 5-7 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	6,231 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
最大使用ユーザスタック	160 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

表 5-8 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	6,453 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
最大使用ユーザスタック	114 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) 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	5,633 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_s25fl_usr.c r_qspi_flash_s25fl_sub.c r_qspi_flash_s25fl_drvif.c r_qspi_flash_s25fl_sfr_rl78.c
最大使用ユーザスタック	140 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。

5.4 ファイル構成

表 5-10 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 5-10 ファイル構成

¥an_r01an1529jj0104_mcu_serial	<DIR>	サンプルコードのフォルダ
r01an1529jj0104_mcu.pdf		アプリケーションノート
¥source	<DIR>	プログラム格納用フォルダ
¥r_qspi_flash_s25fl	<DIR>	S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御ソフトウェア用フォルダ
r_qspi_flash_s25fl.h		ヘッダファイル
r_qspi_flash_s25fl_drvif.c		ドライバ I/F ソースファイル
r_qspi_flash_s25fl_drvif.h		ドライバ I/F ヘッダファイル
r_qspi_flash_s25fl_sfr.h.rl78g14		レジスタ用共通定義 (RL78/G14 用)
r_qspi_flash_s25fl_sfr.h.rl78g1c		レジスタ用共通定義 (RL78/G1C 用)
r_qspi_flash_s25fl_sfr.h.rl78l1c		レジスタ用共通定義 (RL78/L1C 用)
r_qspi_flash_s25fl_sfr.h.rl78l12		レジスタ用共通定義 (RL78/L12 用)
r_qspi_flash_s25fl_sfr.h.rl78l13		レジスタ用共通定義 (RL78/L13 用)
r_qspi_flash_s25fl_sfr.h.rx63n		レジスタ用共通定義 (RX63N 用)
r_qspi_flash_s25fl_sfr.h.rx111		レジスタ用共通定義 (RX111 用)
r_qspi_flash_s25fl_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_s25fl_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_s25fl_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_s25fl_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_s25fl_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_s25fl_sub.c		内部関数ソースファイル
r_qspi_flash_s25fl_sub.h		内部関数ヘッダファイル
r_qspi_flash_s25fl_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

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

5.5 定数一覧

5.5.1 戻り値

表 5-11 にサンプルコードで使用する戻り値を示します。

表 5-11 戻り値 (r_qspi_flash_s25fl.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)	Block Protection and OTP Locked error
FLASH_ERR_TIMEOUT	(error_t)(-6)	Time out error
FLASH_ERR_OTHER	(error_t)(-7)	Other error

5.5.2 コマンド定義

表 5-12 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 5-12 コマンド定義 (r_qspi_flash_s25fl_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_RDSR1	(uint8_t)(0x05)	Read Status Register-1
FLASH_CMD_RDSR2	(uint8_t)(0x07)	Read Status Register-2
FLASH_CMD_RDCCR	(uint8_t)(0x35)	Read Configuration Register-1
FLASH_CMD_WRR	(uint8_t)(0x01)	Write Register (Status-1, Configuration-1)
FLASH_CMD_CLSR	(uint8_t)(0x30)	Clear Status Register-1
FLASH_CMD_FREAD	(uint8_t)(0x0b)	Fast Read (3-byte address)
FLASH_CMD_4FREAD	(uint8_t)(0x0c)	Fast Read (4-byte address)
FLASH_CMD_4DOR	(uint8_t)(0x3c)	Read Dual Out (4-byte address)
FLASH_CMD_4QOR	(uint8_t)(0x6c)	Read Quad Out (4-byte address)
FLASH_CMD_PP	(uint8_t)(0x02)	Page Program (3-byte address)
FLASH_CMD_4PP	(uint8_t)(0x12)	Page Program (4-byte address)
FLASH_CMD_4QPP	(uint8_t)(0x34)	Quad Page Program (4-byte address)
FLASH_CMD_SE	(uint8_t)(0xd8)	Erase 64 kB or 256 kB (3-byte address)
FLASH_CMD_4SE	(uint8_t)(0xdc)	Erase 64 kB or 256 kB (4-byte address)
FLASH_CMD_P4E	(uint8_t)(0x20)	Parameter 4 kB-sector Erase (3-byte address)
FLASH_CMD_4P4E	(uint8_t)(0x21)	Parameter 4 kB-sector Erase (4-byte address)
FLASH_CMD_BE	(uint8_t)(0x60)	Bulk Erase 本アプリケーションノートでは、128Mbit 品使用時、本コマンドにより Bulk Erase を行います。
FLASH_CMD_BE_ALT	(uint8_t)(0xc7)	Bulk Erase (alternate command) 本アプリケーションノートでは、256Mbit 品使用時、本コマンドにより Bulk Erase を行います。
FLASH_CMD_REMS	(uint8_t)(0x90)	Read Electronic Manufacturer Signature
FLASH_CMD_RESET	(uint8_t)(0xf0)	Software Reset

5.5.3 各種定義

表 5-13 から表 5-17 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 5-13 r_qspi_flash_s25fl.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_B_ERASE	(uint8_t)(1)	Erase Mode : Bulk Erase
FLASH_MODE_S_ERASE	(uint8_t)(2)	Erase Mode : Sector Erase
FLASH_MODE_SS_ERASE	(uint8_t)(3)	Erase Mode : Subsector Erase
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)(0xffff0000)	セクタ消去時のセクタアドレスマスク値 左記は、256Mbit の場合の値です。
FLASH_SSECT_ADDR	(uint32_t)(0xfffff000)	サブセクタ消去時のセクタアドレスマスク値 左記は、256Mbit の場合の値です。
FLASH_PAGE_SIZE	(uint32_t)(256)	ページサイズ (バイト単位) 左記は、256Mbit の場合の値です。
FLASH_ADDR_SIZE	(uint8_t)(4)	アドレスサイズ (バイト単位) 左記は、256Mbit の場合の値です。
FLASH_WP_WHOLE_MEM	(uint8_t)(0x07)	チップ全体のライトプロテクト 左記は、256Mbit の場合の値です。
FLASH_SECTOR_TYPE	FLASH_TYPE_64_SECTOR	Sector Type 左記は、256Mbit, Uniform 64-kB sectors 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_REMSADDR_SIZE	(uint8_t)(3)	REMS コマンドアドレスサイズ
FLASH_STSREG_SIZE	(uint16_t)(2)	Status register サイズ (バイト単位)
FLASH_CFGREG_SIZE	(uint16_t)(2)	Configuration register サイズ (バイト単位)
FLASH_IDDATA_SIZE	(uint16_t)(2)	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、 μ ITRON4.0 を想定しています。

表 5-14 r_qspi_flash_s25fl_sfr.h.rx63n の各種定義値

定数名	設定値	内容
FLASH_DR_CS0	PORTA.PODR.BIT.B0	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	PORTA.PDR.BIT.B0	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 48[MHz]とし、転送レートを 12[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 48[MHz]とし、転送レートを 6[MHz]に設定する場合の値です。

表 5-15 r_qspi_flash_s25fl_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 5-16 r_qspi_flash_s25fl_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 5-17 r_qspi_flash_s25fl_sub.h の各種定義値

定数名	設定値	内容
FLASH_BE_BUSY_WAIT	(uint32_t)(330000)	Bulk Erase Busy Timeout 330000 × 1ms = 330s
FLASH_SE_BUSY_WAIT	(uint32_t)(10400)	Sector Erase Busy Timeout 10400 × 1ms = 10.4s
FLASH_SSE_BUSY_WAIT	(uint32_t)(650)	Subsector Erase Busy Timeout 650 × 1ms = 650ms
FLASH_PBUSY_WAIT	(uint32_t)(750)	Write Ready Timeout 750 × 1us = 750us
FLASH_WBUSY_WAIT	(uint32_t)(500)	Write Ready Timeout 500 × 1ms = 500ms
FLASH_T_PBUSY_WAIT	(uint16_t)MTL_T_1US	Page Program Busy Polling Time
FLASH_T_WBUSY_WAIT	(uint16_t)MTL_T_1MS	Write 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_STSREG_SRWD	(uint8_t)(0x80)	Status Register Write Disable
FLASH_STSREG_P_ERR	(uint8_t)(0x40)	Programming Error Occurred
FLASH_STSREG_E_ERR	(uint8_t)(0x20)	Erase Error Occurred
FLASH_STSREG_BP2	(uint8_t)(0x10)	Block Protection Bit2
FLASH_STSREG_BP1	(uint8_t)(0x08)	Block Protection Bit1
FLASH_STSREG_BP0	(uint8_t)(0x04)	Block Protection Bit0
FLASH_STSREG_WEL	(uint8_t)(0x02)	Write Enable Latch Bit
FLASH_STSREG_WIP	(uint8_t)(0x01)	Write In Progress Bit
FLASH_STSREG	(uint8_t)(0x9c)	Write status fixed data
FLASH_STSREG_BPMASK	(uint8_t)(0x1c)	Block Protect Bit Mask
FLASH_CFGREG_LC1	(uint8_t)(0x80)	Latency Code 1
FLASH_CFGREG_LC0	(uint8_t)(0x40)	Latency Code 0
FLASH_CFGREG_TBPROT	(uint8_t)(0x20)	Configures Start of Block Protection
FLASH_CFGREG_RFU	(uint8_t)(0x10)	Reserved for Future Use
FLASH_CFGREG_BPNV	(uint8_t)(0x08)	Configures BP2-0 in Status Register
FLASH_CFGREG_TBPARAM	(uint8_t)(0x04)	Configures Parameter Sectors location
FLASH_CFGREG_QUAD	(uint8_t)(0x02)	Puts the device into Quad I/O operation
FLASH_CFGREG_FREEZE	(uint8_t)(0x01)	Lock current state of BP2-0 bits
FLASH_CFGREG	(uint8_t)(0xef)	Write configuration fixed data

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_s25fl_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_s25fl.h 参照)

表 5-18 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: S25FLxxxS MirrorBit® Flash Non-Volatile Memory へ書き込むデータの格納元 読み出し時: S25FLxxxS MirrorBit® Flash Non-Volatile 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 5-19 static 型変数 (r_qspi_flash_s25fl_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 5-20 関数

関数名	概要
R_QSPI_FLASH_Init_Driver()	ドライバ初期化処理
R_QSPI_FLASH_Reset_Device()	デバイス初期化処理
R_QSPI_FLASH_Read_Status()	ステータスレジスタ 1 読み出し処理
R_QSPI_FLASH_Read_Status2()	ステータスレジスタ 2 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_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_Clear_Status()	クリアステータス処理
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

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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。

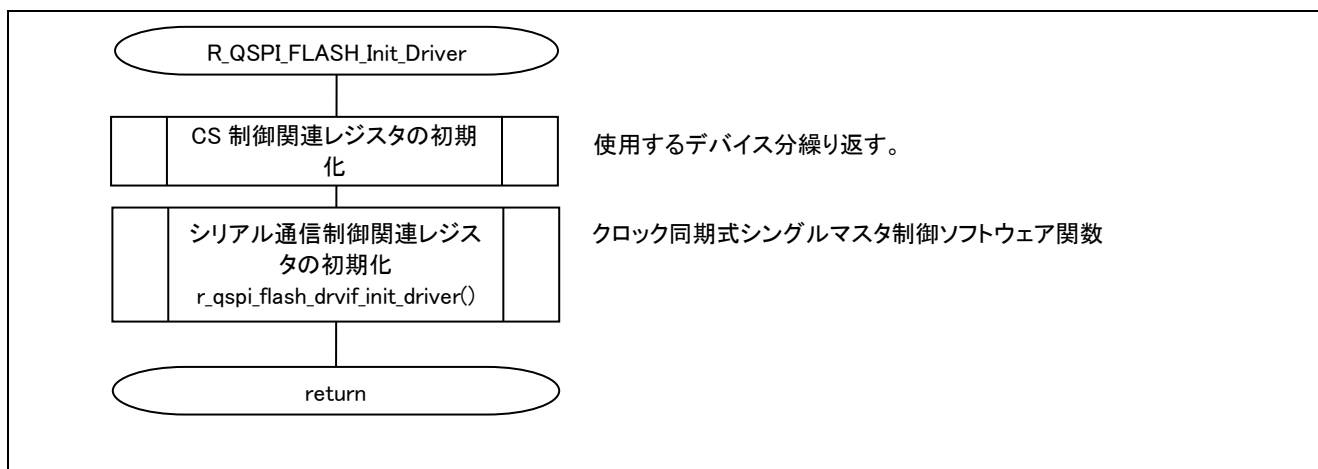


図 5-8 ドライバ初期化処理概要

5.9.2 デバイス初期化処理

R_QSPI_FLASH_Reset_Device	
概要	デバイス初期化処理
ヘッダ	r_qspi_flash_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_drvif.h
宣言	error_t R_QSPI_FLASH_Reset_Device(uint8_t DevNo)
説明	<ul style="list-style-type: none"> ・ Software Reset (RESET)コマンドを発行し、デバイスを電源投入時の状態に戻します。 ・ コンフィグレーションレジスタの不揮発性ビット (TBPROT、TBPARAM、BPNV) は、初期化されずに処理前の値を保持します。 ・ RESET コマンドは強制的に発行することができます。データ書き込みや消去の完了待ち中に発行した場合、処理は中断され、処理対象エリアのデータは不定になります。
引数	uint8_t DevNo ; デバイス番号
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_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

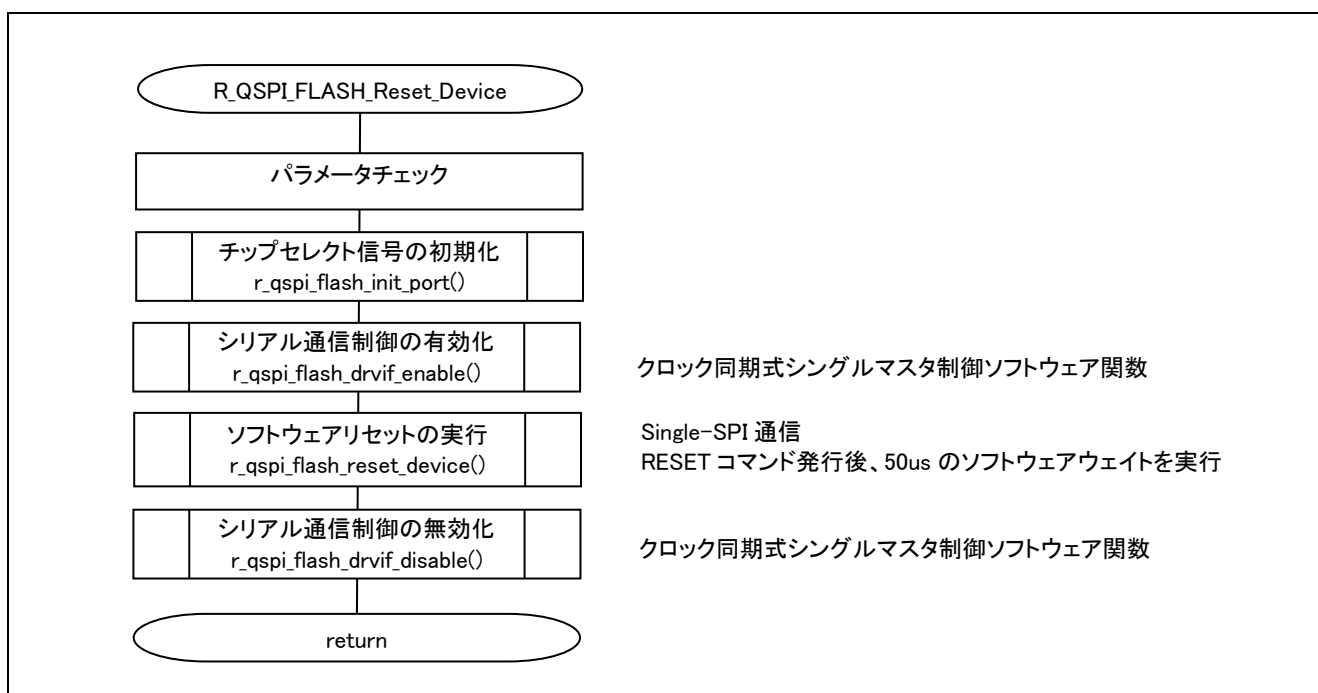


図 5-9 デバイス初期化処理概要

5.9.3 ステータスレジスタ 1 読み出し処理

R_QSPI_FLASH_Read_Status	
概要	ステータスレジスタ 1 読み出し処理
ヘッダ	r_qspi_flash_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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 1 を読み出し、pStatus に格納します。 読み出しバッファとして、2 バイトを設定してください。 ・ 読み出しステータス格納バッファ (pStatus) には下記情報が格納されます。 <ul style="list-style-type: none"> Bit 7: Status register write disable (SRWD) <ul style="list-style-type: none"> 1: BP, and configuration register bits are read-only 0: No protection Bit 6: Programming Error Occurred (P_ERR) <ul style="list-style-type: none"> 1: Programming Error Occurred 0: Programming Error not Occurred Bit 5: Erase Error Occurred (E_ERR) <ul style="list-style-type: none"> 1: Erase Error Occurred 0: Erase Error not Occurred Bits 4 to 2: Block Protection 2 - 0 (BP2 - BP0) 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 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 S25FLxxxS MirrorBit® Flash Non-Volatile 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 ; 読み出しステータス格納バッファポインタ
リターン値	<ul style="list-style-type: none"> ・ ステータスレジスタ 1 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

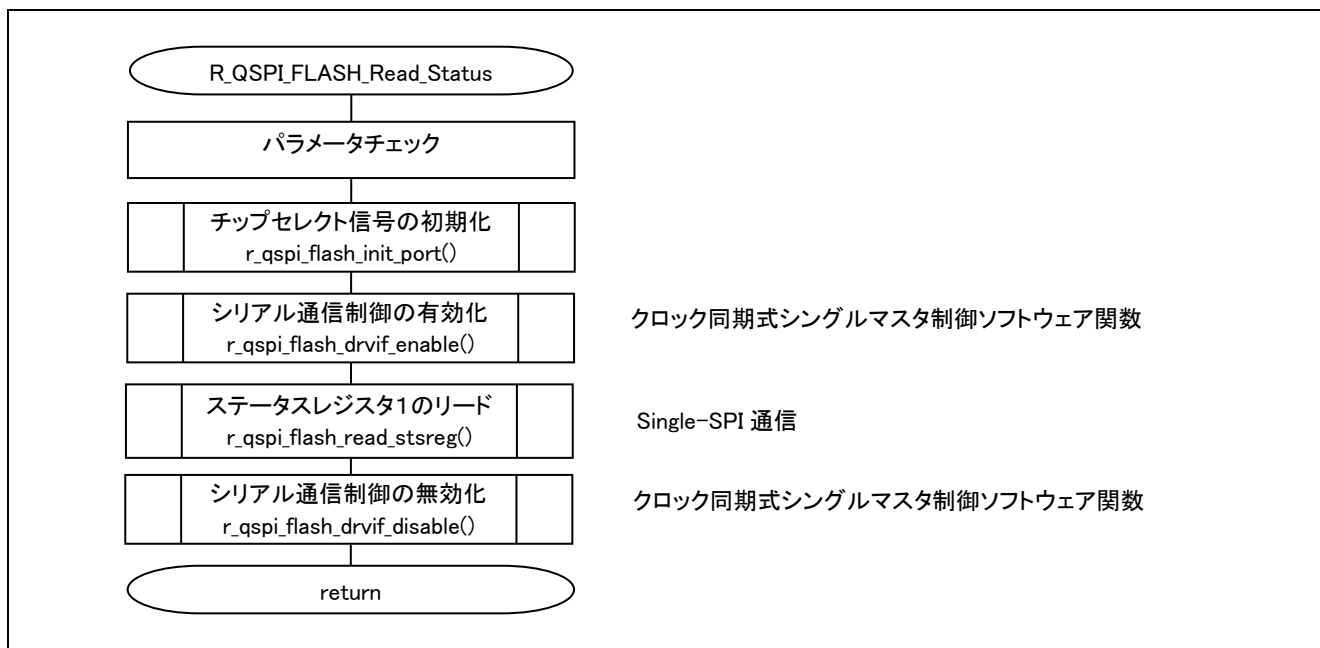


図 5-10 ステータスレジスタ 1 読み出し処理概要

5.9.4 ステータスレジスタ 2 読み出し処理

R_QSPI_FLASH_Read_Status2	
概要	ステータスレジスタ 2 読み出し処理
ヘッダ	r_qspi_flash_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_drvif.h
宣言	error_t R_QSPI_FLASH_Read_Status2(uint8_t DevNo, uint8_t FAR* pStatus)
説明	<ul style="list-style-type: none"> ステータスレジスタ 2 を読み出し、pStatus に格納します。読み出しバッファとして、2 バイトを設定してください。 読み出しステータス格納バッファ (pStatus) には下記情報が格納されます。 <ul style="list-style-type: none"> Bit 7 to 2: Reserved for Future Use Bit 1: Erase Suspend (ES) <ul style="list-style-type: none"> 1: In erase suspend mode 0: Not in erase suspend mode Bit 0: Program Suspend (PS) <ul style="list-style-type: none"> 1: In program suspend mode 0: Not in program suspend mode
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pStatus ; 読み出しステータス格納バッファポインタ
リターン値	<ul style="list-style-type: none"> ステータスレジスタ 2 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

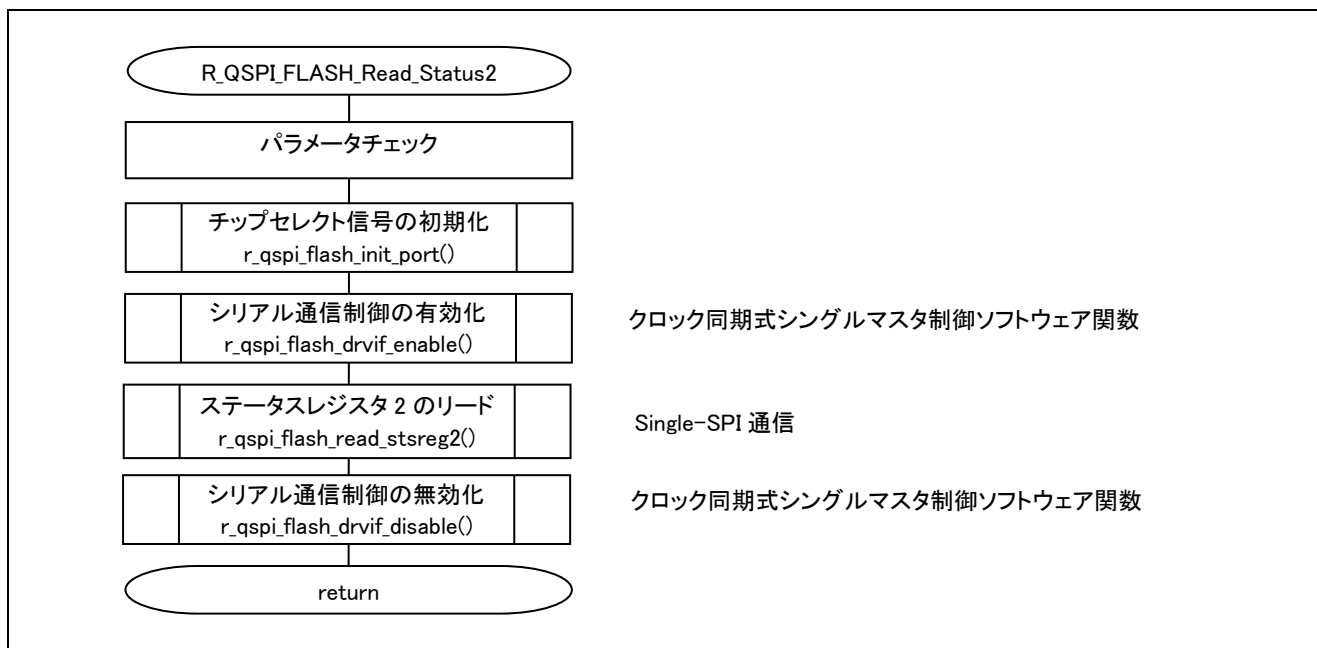


図 5-11 ステータスレジスタ 2 読み出し処理概要

5.9.5 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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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。 ・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。なお、TBPROT ビット (※) の設定により、プロテクトの領域が異なります。

●TBPROT=0 の場合 (上位アドレス領域のプロテクト)

WpSts	BP2	BP1	BP0	Memory Array	128Mbit [kbytes]	256Mbit [kbytes]
0x00	0	0	0	None	0	0
0x01	0	0	1	Upper 64th	256	512
0x02	0	1	0	Upper 32nd	512	1024
0x03	0	1	1	Upper 16th	1024	2048
0x04	1	0	0	Upper 8th	2048	4096
0x05	1	0	1	Upper 4th	4096	8192
0x06	1	1	0	Upper Half	8192	16384
0x07	1	1	1	All Sectors	16384	32768

●TBPROT=1 の場合 (下位アドレス領域のプロテクト)

WpSts	BP2	BP1	BP0	Memory Array	128Mbit [kbytes]	256Mbit [kbytes]
0x00	0	0	0	None	0	0
0x01	0	0	1	Lower 64th	256	512
0x02	0	1	0	Lower 32nd	512	1024
0x03	0	1	1	Lower 16th	1024	2048
0x04	1	0	0	Lower 8th	2048	4096
0x05	1	0	1	Lower 4th	4096	8192
0x06	1	1	0	Lower Half	8192	16384
0x07	1	1	1	All Sectors	16384	32768

※: コンフィグレーションレジスタの TBPROT ビットについて

TBPROT ビットにより、プロテクト領域を選択することができます。デフォルト設定では、上位アドレス領域が選択されています (TBPROT=0)。下位アドレス領域へプロテクトを行う場合は、TBPROT ビットを“1”にセットしてください。ただし、TBPROT は OTP ビットです。一度書き換えると、以降の書き換えは受け付けませんので、ご注意ください。

- ・SRWD は、0 に設定されます。
- ・プロテクト領域とプロテクトビットの関係は、使用する S25FLxxxS MirrorBit® Flash Non-Volatile Memory の仕様を参照してください。BP ビットが割り当てられていない可能性があります。
- ・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。
- ・本ユーザ API で完了待ちを行う場合、r_qspi_flash_s25fl.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_s25fl.h` の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、ビジーウェイト処理 (`R_QSPI_FLASH_Wait()`) を実行してください。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を確認することができます。使用方法は図 5-13 を参照してください。
- ・プロテクト設定が正常に行われたかは、ステータスレジスタ 1 を読み出して確認してください。
- ・リターン値が「FLASH_OK」以外の場合、ステータスレジスタ 1 を読み出し、状態を確認してください。
- ・リターン値が「FLASH_ERR_OTHER」の場合、かつステータスレジスタ 1 の P_ERR もしくは E_ERR ビットがセットされている場合、ライトプロテクト設定が正常に完了していません。次の処理（書き込み／読み出し／消去等）の前にクリアステータス処理 (`R_QSPI_FLASH_Clear_Status()`) を行ってください。また、ステータスレジスタ 1 の WEL ビットがセットされている場合、クリアするために、WRDI コマンド発行処理 (`R_QSPI_FLASH_Write_Di()`) も行ってください。

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 ・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_WP` ; Block Protection and OTP Locked error
 `FLASH_ERR_TIMEOUT` ; Time out error (「FLASH_WAIT_READY」有効時)
 `FLASH_ERR_OTHER` ; Other error

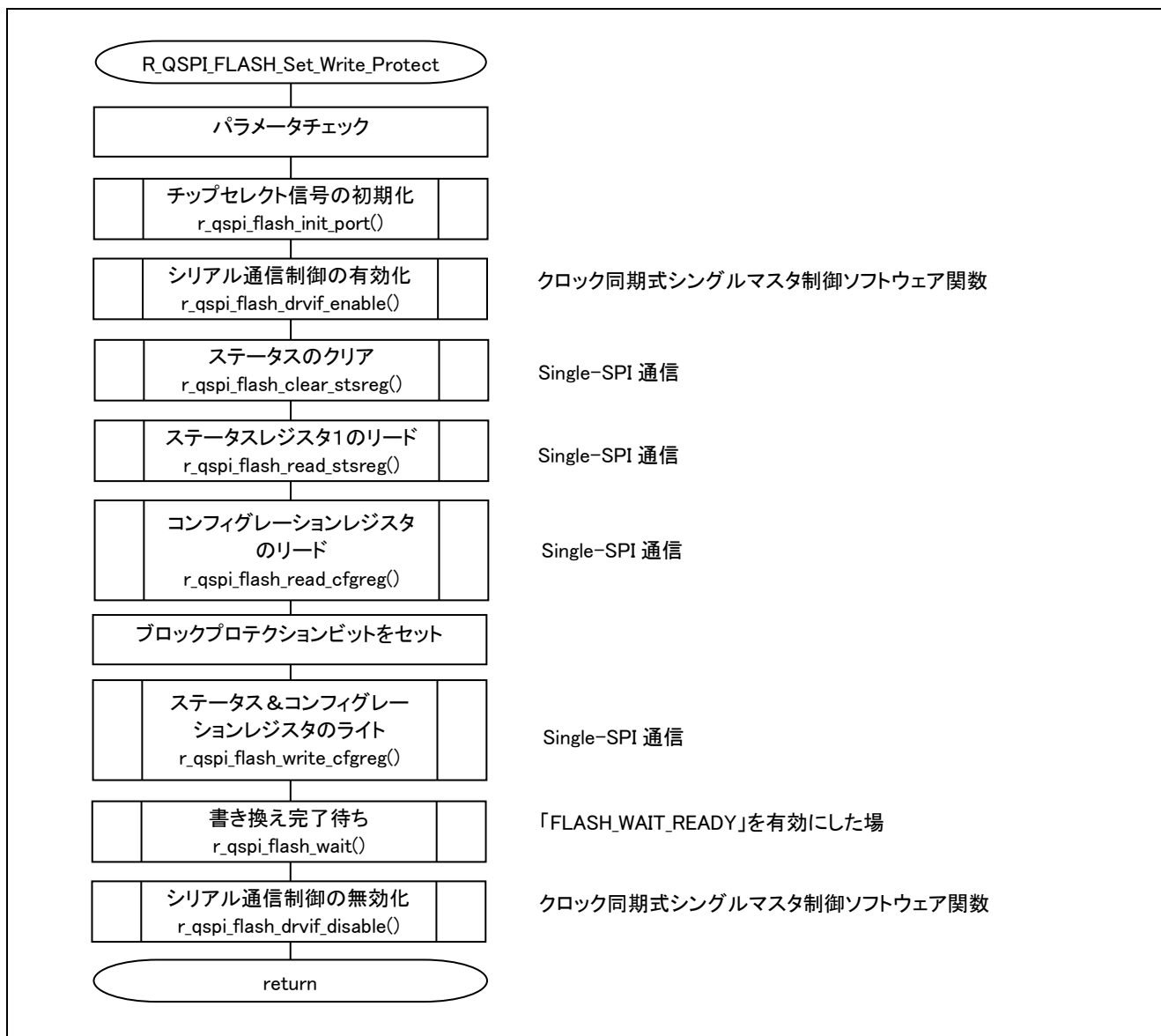


図 5-12 ライトプロテクト設定処理概要

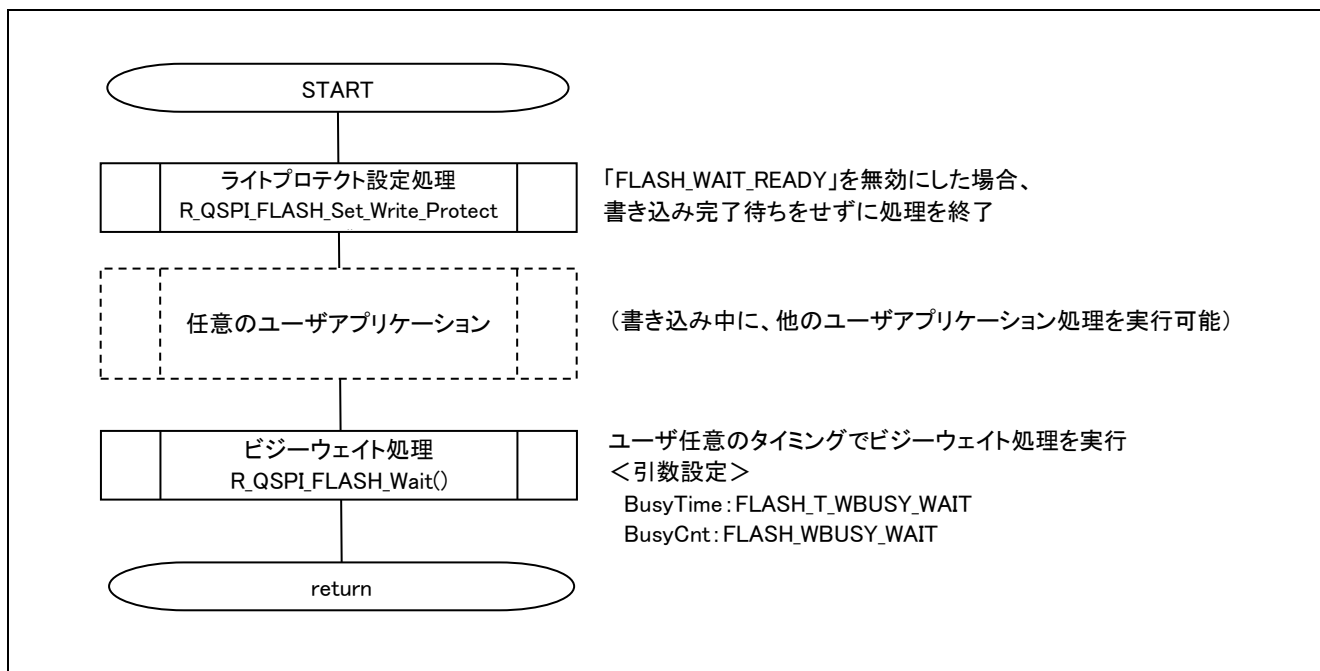


図 5-13 R_QSPI_FLASH_Wait()を使用したライトプロテクト設定完了待ち方法

5.9.6 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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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、2 バイトを設定してください。 ・ 読み出しコンフィグレーション格納バッファ(pConfig)には下記情報が格納されません。 <p>Bit 7 to 6: Latency Code 1 - 0(LC1 - LC0) Selects number of initial read latency cycles</p> <p>Bit 5: Configures Start of Block Protection (TBPROT) 1: BP starts at bottom (Low address) 0: BP starts at top (High address)</p> <p>Bit 4: Reserved for Future Use</p> <p>Bit 3: Configures BP2-0 in Status Register (BPNV) 1: Volatile 0: Non-Volatile</p> <p>Bit 2: Configures Parameter Sectors location (TBPARAM) 1: 4 kB physical sectors at top (high address) 0: 4 kB physical sectors at bottom (Low address) RFU in uniform sector devices</p> <p>Bit 1: Puts the device into Quad I/O operation (QUAD) 1: Quad 0: Dual or Serial</p> <p>Bit 0: Lock current state of BP2-0 bits in Status Register, TBPROT and TBPARAM in Configuration Register, and OTP regions (FREEZE) 1: Block Protection and OTP locked 0: Block Protection and OTP un-locked</p>
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pConfig ; 読み出しコンフィグレーション格納バッファポインタ
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。 <p>FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error</p>

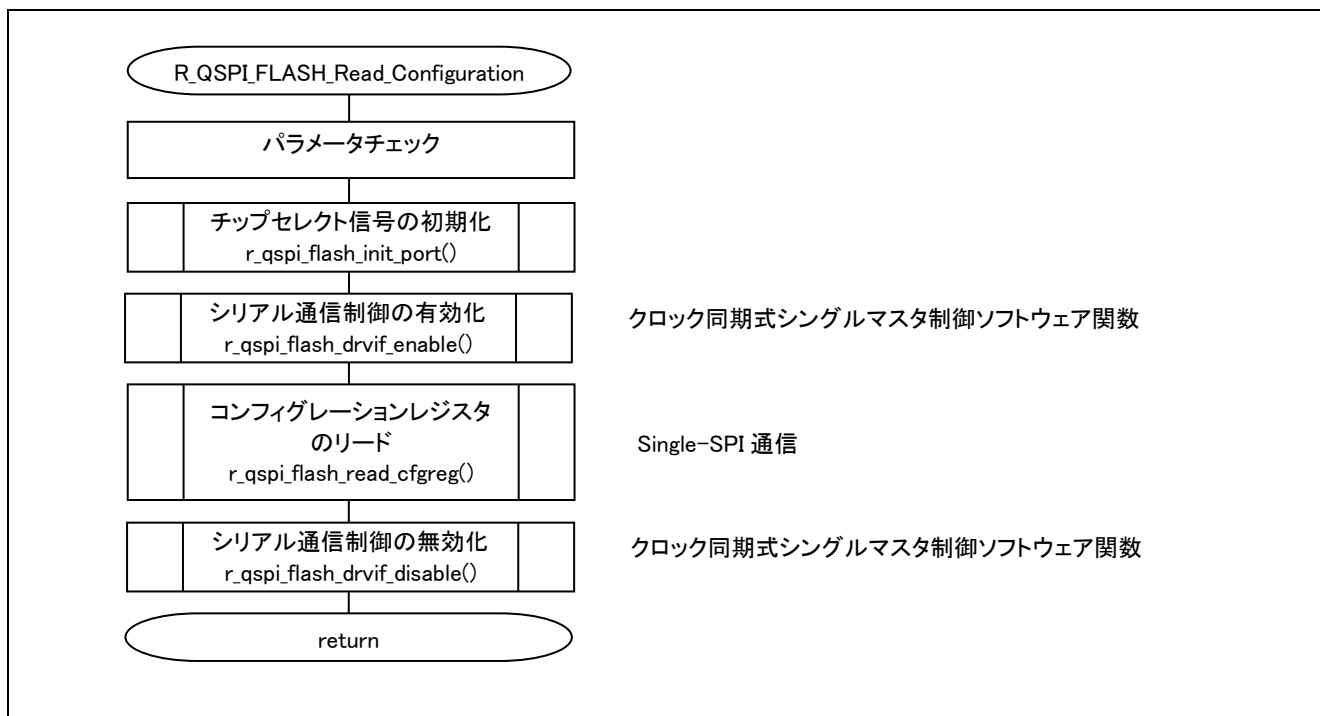


図 5-14 コンフィグレーションレジスタ読み出し処理概要

5.9.7 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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_drvif.h
宣言	error_t R_QSPI_FLASH_Write_Configuration(uint8_t DevNo, uint8_t Config)
説明	<ul style="list-style-type: none"> ・ Config のデータをコンフィグレーションレジスタに書き込みます。 ・ 書き込みコンフィグレーションデータ(Config)には下記情報を格納してください。 <ul style="list-style-type: none"> Bit 7 to 6: Latency Code 1 - 0(LC1 - LC0) Selects number of initial read latency cycles Bit 5: Configures Start of Block Protection (TBPROT) <ul style="list-style-type: none"> 1: BP starts at bottom (Low address) 0: BP starts at top (High address) Bit 4: Reserved for Future Use Bit 3: Configures BP2-0 in Status Register (BPNV) <ul style="list-style-type: none"> 1: Volatile 0: Non-Volatile Bit 2: Configures Parameter Sectors location (TBPARAM) <ul style="list-style-type: none"> 1: 4 kB physical sectors at top (high address) 0: 4 kB physical sectors at bottom (Low address) RFU in uniform sector devices Bit 1: Puts the device into Quad I/O operation (QUAD) <ul style="list-style-type: none"> 1: Quad 0: Dual or Serial Bit 0: Lock current state of BP2-0 bits in Status Register, TBPROT and TBPARAM in Configuration Register, and OTP regions (FREEZE) <ul style="list-style-type: none"> 1: Block Protection and OTP locked 0: Block Protection and OTP un-locked ・ Bit2, 3, 4, 5 は OTP ビットです。一度セットするとデフォルト値に戻すことはできません。コンフィグレーションレジスタを書き換える際は、事前にレジスタの読み出しを行い、OTP ビットの値をセットしてください。 ・ Bit1, 6, 7, は Non-Volatile ビットです。書き換え可能ですが、不揮発性であり、電源を Off しても値は保持されます。 ・ Bit0 は Volatile ビットです。また、電源 ON 中は OTP ビットです。セットした場合、電源を Off するまで値は保持され、書き換えることができません。電源を Off にすると値はデフォルトに戻ります。 ・ 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。 ・ 本ユーザ API で完了待ちを行う場合、r_qspi_flash_s25fl.h の「FLASH_WAIT_READY」を有効にしてください。OTP ビットが"1"にも関わらず"0"を設定し書き込みを行った場合、「FLASH_ERR_TIMEOUT」を返します。 ・ 本ユーザ API で完了待ちを行わない場合、r_qspi_flash_s25fl.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、ビジーウェイト処理 (R_QSPI_FLASH_Wait()) を実行してください。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を確認することができます。使用方法は図 5-16 を参照してください。 ・ コンフィグレーションレジスタへの書き込みが正常に行われたかは、ステータスレジスタ 1 を読み出して確認してください。 ・ リターン値が「FLASH_ERR_OTHER」もしくは「FLASH_ERR_TIMEOUT」の場合

Spansion 社製 S25FLxxxS MirrorBit® Flash Non-Volatile Memory 制御ソフトウェア

合、かつステータスレジスタ 1 の P_ERR もしくは E_ERR ビットがセットされている場合、レジスタ書き込みが正常に完了していません。次の処理（書き込み／読み出し／消去等）の前にクリアステータス処理（R_QSPI_FLASH_Clear_Status()）を行ってください。また、ステータスレジスタ 1 の WEL ビットがセットされている場合、クリアするために、WRDI コマンド発行処理（R_QSPI_FLASH_Write_Di()）も行ってください。

引 数 uint8_t DevNo ; デバイス番号
 uint8_t Config ; 書き込みコンフィグレーションデータ

リターン値 ・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 ; Time out error（「FLASH_WAIT_READY」有効時）
 FLASH_ERR_OTHER ; Other error

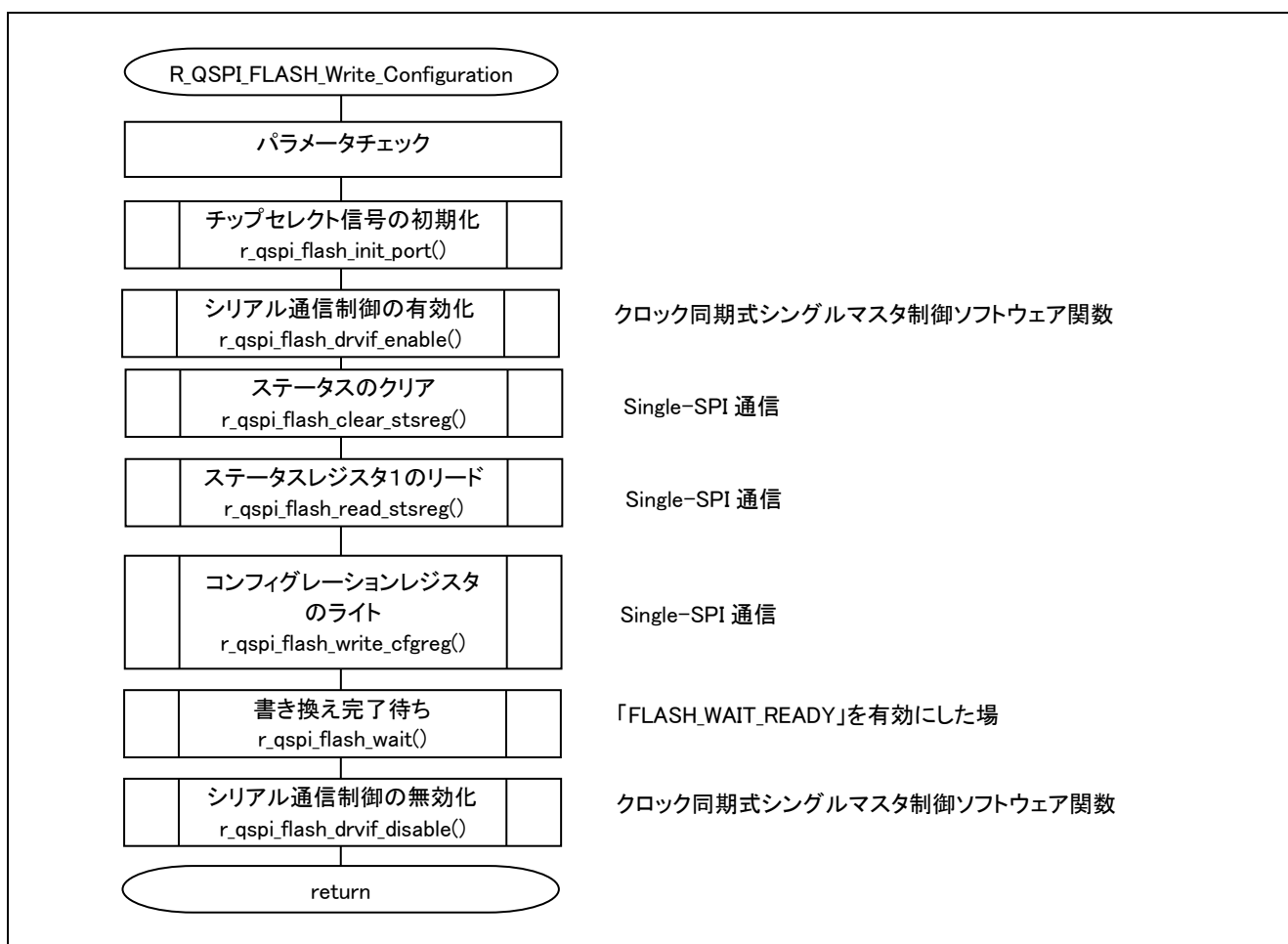


図 5-15 コンフィグレーションレジスタ書き込み処理概要

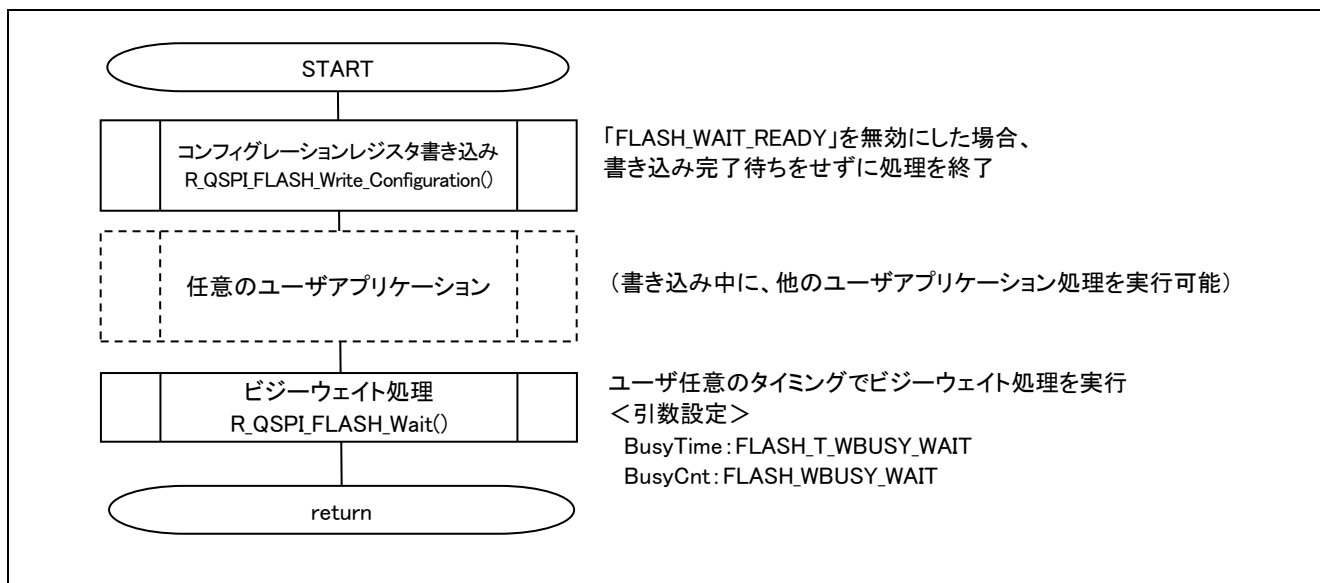


図 5-16 R_QSPI_FLASH_Wait()を使用したライトプロテクト設定完了待ち方法

5.9.8 クリアステータス処理

R_QSPI_FLASH_Clear_Status

概要	クリアステータス処理	
ヘッダ	r_qspi_flash_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_drvif.h	
宣言	error_t R_QSPI_FLASH_Clear_Status(uint8_t DevNo)	
説明	<ul style="list-style-type: none"> ・ステータスレジスタ 1 のエラービットをクリアします。 ・プログラムエラー、消去エラーまたはライトプロテクトエラーが発生した場合は、本関数をコールしてエラービットをクリアする必要があります。 	
引数	uint8_t	DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> ・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

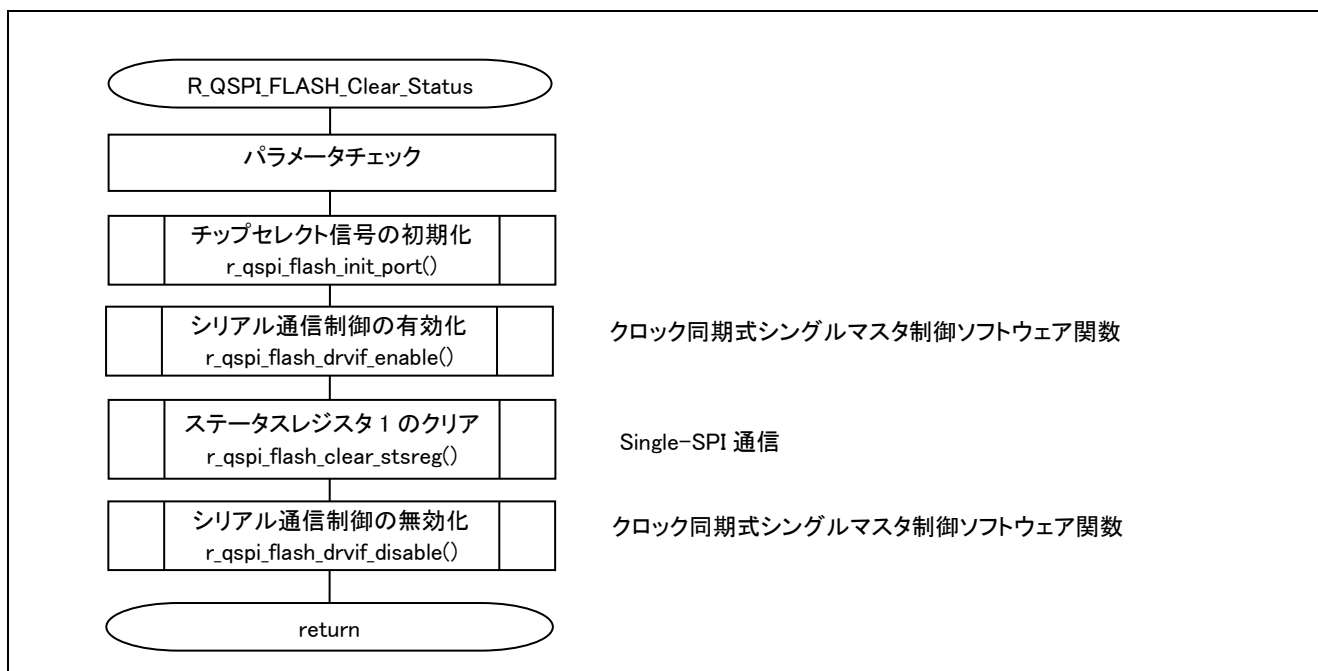


図 5-17 クリアステータス処理概要

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_drvif.h
宣言	error_t R_QSPI_FLASH_Write_Di(uint8_t DevNo)
説明	<ul style="list-style-type: none"> ・ステータスレジスタ 1 の WEL ビットをクリアします。 ・レジスタ書き込みエラー、プログラムエラー、消去エラーまたはライトプロテクトエラーが発生した場合は、WEL ビットがセットされた状態です。本関数をコールして WEL ビットをクリアしてください。
引数	uint8_t DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> ・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

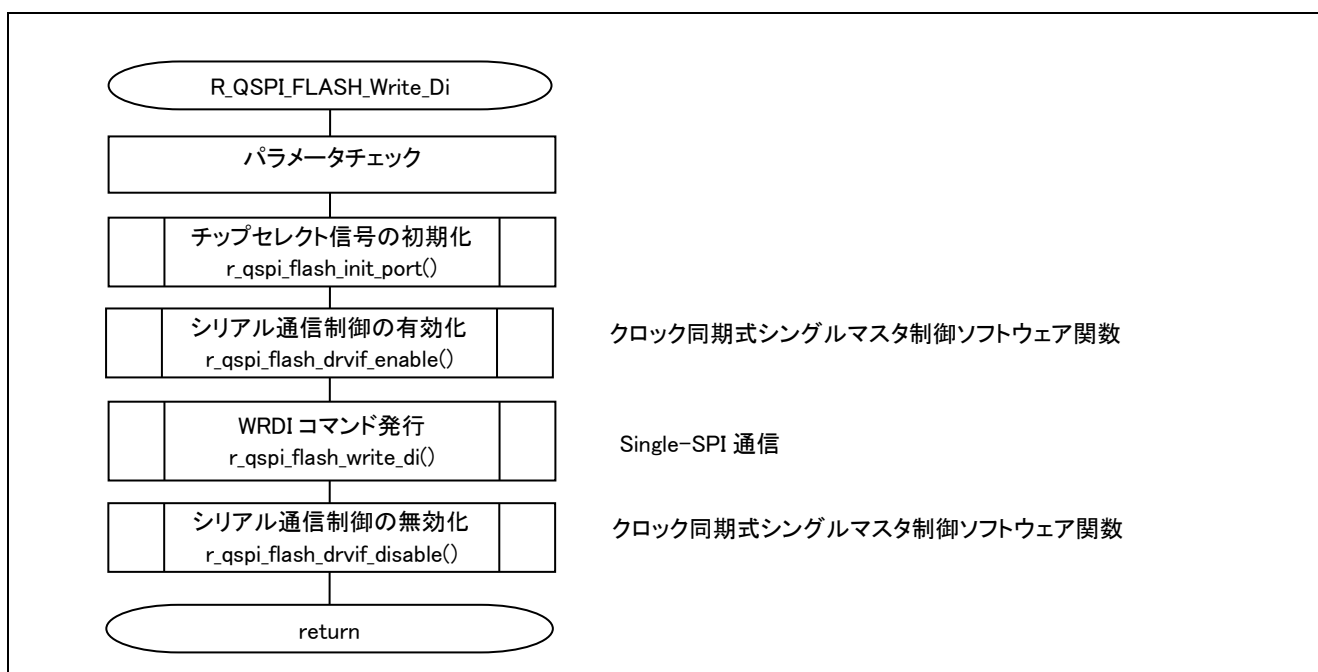


図 5-18 WRDI コマンド発行処理概要

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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"> ・ S25FLxxxS MirrorBit® Flash Non-Volatile Memory の指定アドレスから指定バイト数分、バイト単位でデータを読み出し、pData に格納します。 ・ 最終読み出しアドレスは、S25FLxxxS MirrorBit® Flash Non-Volatile 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																							

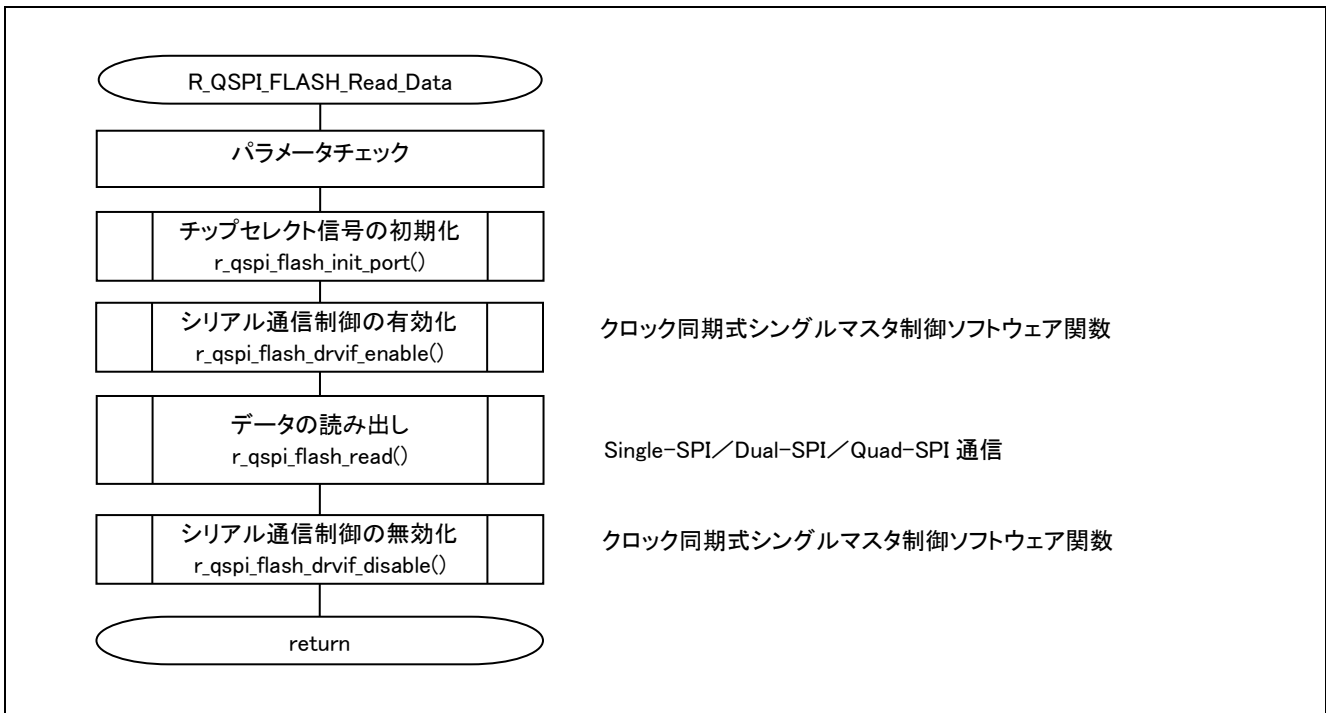


図 5-19 データ読み出し処理概要

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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 S25FLxxxS MirrorBit® Flash Non-Volatile Memory 上の指定アドレスから指定バイト数分書き込みます。 ・ S25FLxxxS MirrorBit® Flash Non-Volatile 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。書き込み完了待ちを実行すると「FLASH_ERR_OTHER」を返します。 ・ 最終書き込みアドレスは、S25FLxxxS MirrorBit® Flash Non-Volatile 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、S25FLxxxS MirrorBit® Flash Non-Volatile Memory 容量の値です。 ・ 本ユーザ API では、r_qspi_flash_s25fl.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。 ・ リターン値が「FLASH_OK」以外の場合、ステータスレジスタ 1 を読み出し、状態を確認してください。 ・ リターン値が「FLASH_ERR_OTHER」の場合、かつステータスレジスタ 1 の P_ERR ビットがセットされている場合、プログラムエラーもしくはプロテクトされた領域への書き込みが発生しています。次の処理（書き込み／読み出し／消去等）の前にクリアステータス処理（R_QSPI_FLASH_Clear_Status()）を行ってください。また、ステータスレジスタ 1 の WEL ビットがセットされている場合、クリアするために、WRDI コマンド発行処理（R_QSPI_FLASH_Write_Di()）も行ってください。 																								
引数	<table> <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"> ・ 書き込み結果を返します。 <table> <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>Time out 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_TIMEOUT	;	Time out error	FLASH_ERR_OTHER	;	Other error									
FLASH_OK	;	Successful operation																							
FLASH_ERR_PARAM	;	Parameter error																							
FLASH_ERR_HARD	;	Hardware error																							
FLASH_ERR_TIMEOUT	;	Time out error																							
FLASH_ERR_OTHER	;	Other error																							

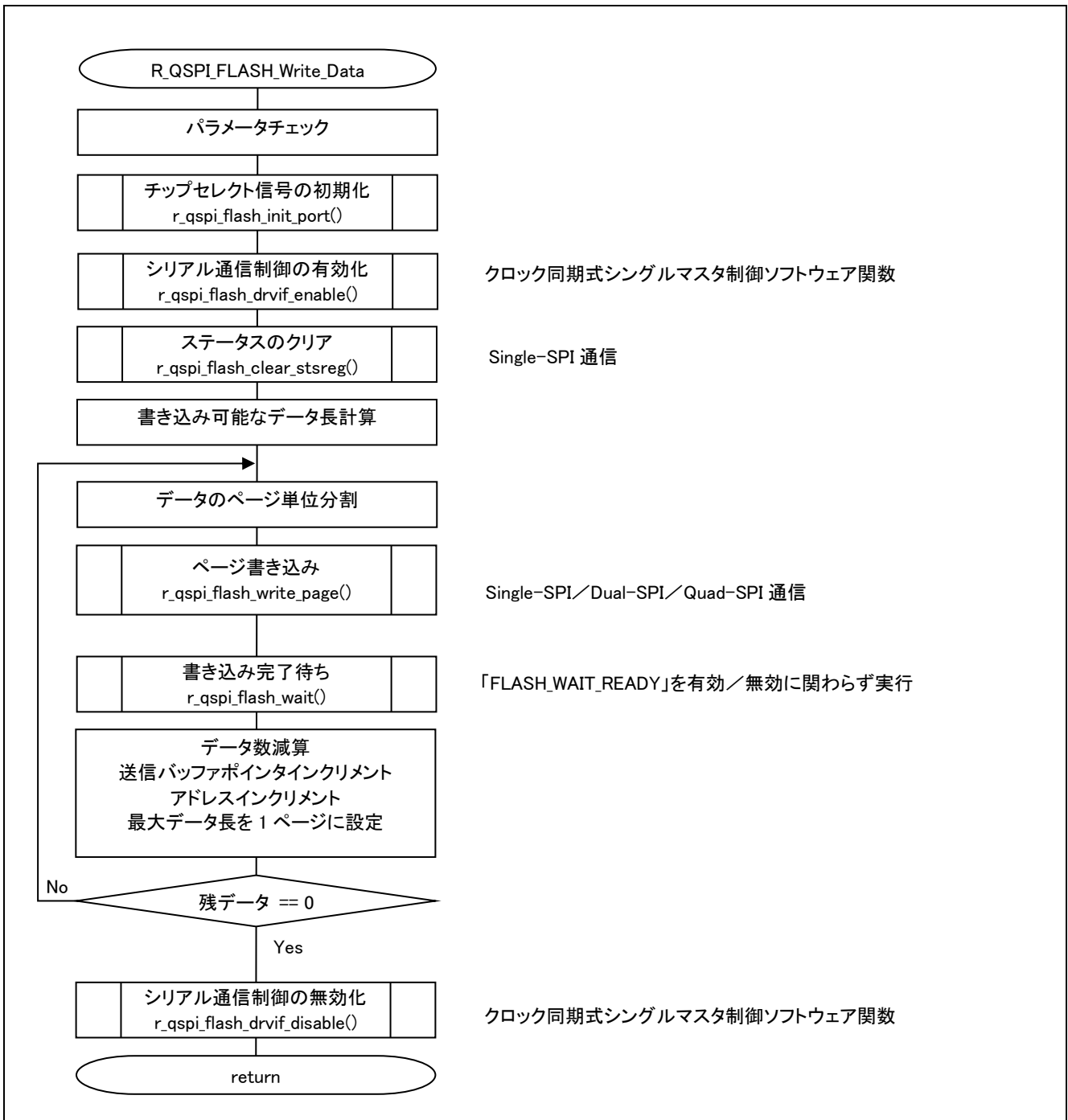


図 5-20 データ書き込み処理概要

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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 S25FLxxxS MirrorBit® Flash Non-Volatile Memory 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。 ・ S25FLxxxS MirrorBit® Flash Non-Volatile 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。書き込み完了待ちを実行すると「FLASH_ERR_OTHER」を返します。 ・ 最終書き込みアドレスは、S25FLxxxS MirrorBit® Flash Non-Volatile 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、S25FLxxxS MirrorBit® Flash Non-Volatile 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_n25q.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_n25q.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、ビジーウェイト処理（R_QSPI_FLASH_Wait()）を実行してください。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を確認することができます。使用方法は図 5-22 を参照してください。 ・ リターン値が「FLASH_OK」以外の場合、ステータスレジスタ 1 を読み出し、状態を確認してください。 ・ リターン値が「FLASH_ERR_OTHER」の場合、かつステータスレジスタ 1 の P_ERR ビットがセットされている場合、プログラムエラーもしくはプロテクトされた領域への書き込みが発生しています。次の処理（書き込み／読み出し／消去等）の前にクリアステータス処理（R_QSPI_FLASH_Clear_Status()）を行ってください。また、ステータスレジスタ 1 の WEL ビットがセットされている場合、クリアするために、WRDI コマンド発行処理（R_QSPI_FLASH_Write_Di()）も行ってください。 																								
引数	<table> <tbody> <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> </tbody> </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"> ・ 書き込み結果を返します。 <table> <tbody> <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> </tbody> </table>	FLASH_OK	;	Successful operation	FLASH_ERR_PARAM	;	Parameter error	FLASH_ERR_HARD	;	Hardware error															
FLASH_OK	;	Successful operation																							
FLASH_ERR_PARAM	;	Parameter error																							
FLASH_ERR_HARD	;	Hardware error																							

FLASH_ERR_TIMEOUT ; Time out error (「FLASH_WAIT_READY」有効時)
 FLASH_ERR_OTHER ; Other error

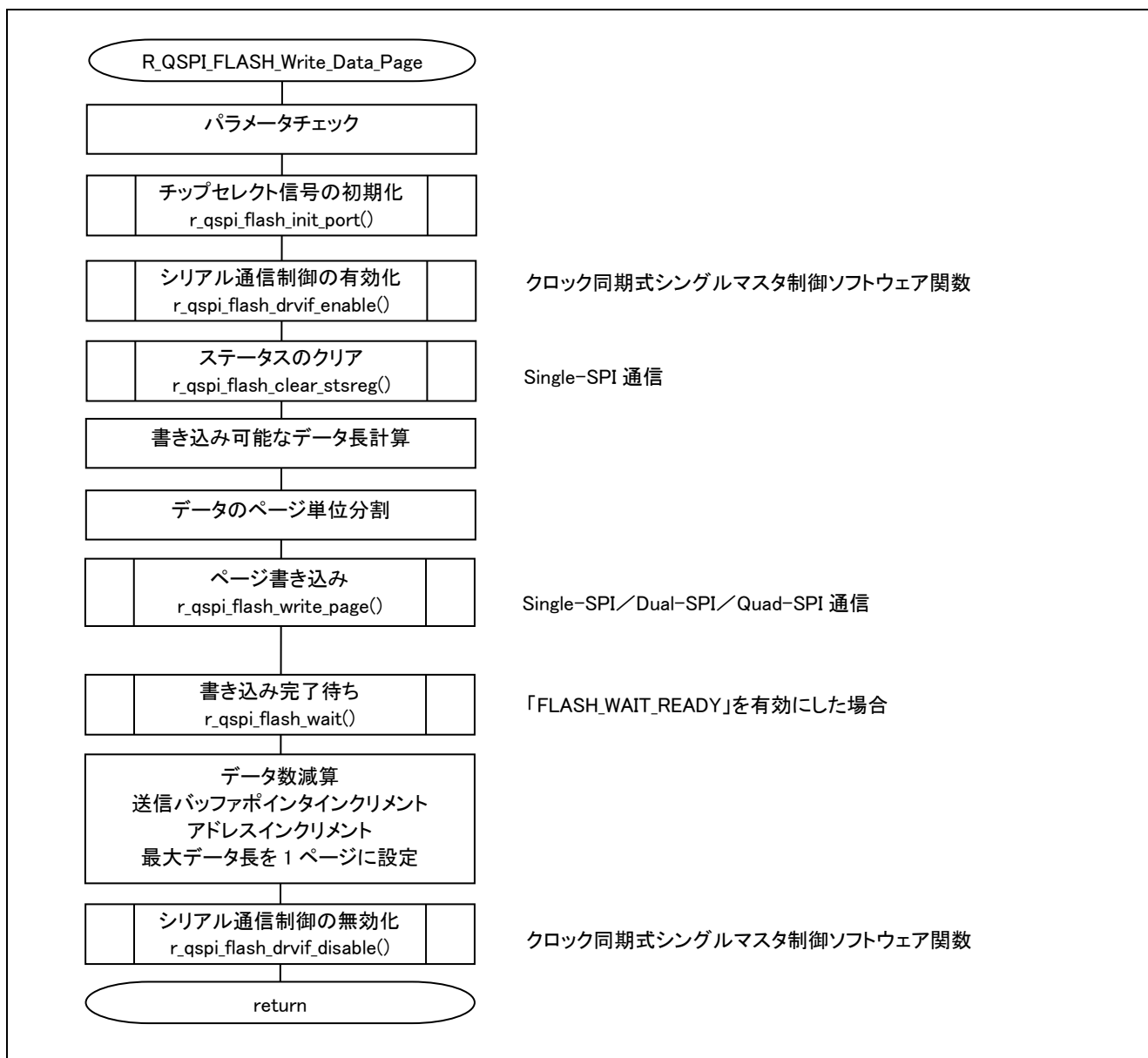


図 5-21 データ書き込み処理 (1Page 書き込み用) 概要

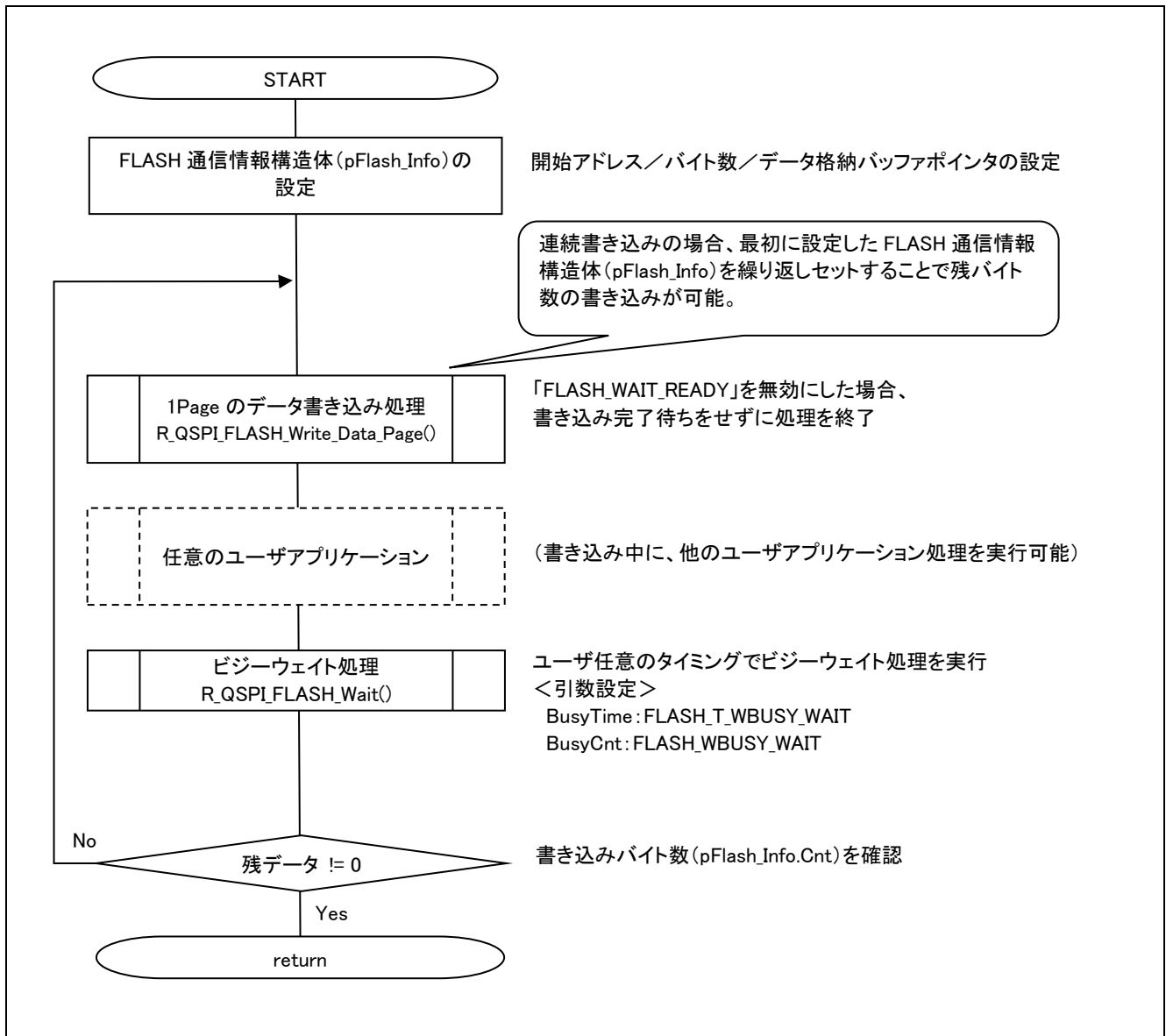


図 5-22 R_QSPI_Flash_Wait()を使用したデータ書き込み処理 (1Page 書き込み用) の完了待ち方法

5.9.13 消去処理

R_QSPI_FLASH_Erase	
概要	消去処理
ヘッダ	r_qspi_flash_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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、メモリの全データ消去 (Bulk Erase)、指定されたセクタの全データ消去 (Sector Erase)、指定されたサブセクタの全データ消去 (SubSector Erase) を行います。 Bulk Erase の場合、Addr には 0x00000000 を設定してください。 S25FLxxxS MirrorBit® Flash Non-Volatile 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領域は、消去はできません。消去完了待ちを実行すると「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_n25q.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_n25q.h の「FLASH_WAIT_READY」を無効にし、本ユーザ API の処理が完了した後、ビジーウェイト処理 (R_QSPI_FLASH_Wait()) を実行してください。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。 Erase の mode により、ビジーウェイト処理 (R_QSPI_FLASH_Wait()) を実行する際の引数 BusyCnt の設定が異なります。 <ul style="list-style-type: none"> Bulk Erase 時 ; BusyCnt = FLASH_BE_BUSY_WAIT Sector Erase 時 ; BusyCnt = FLASH_SE_BUSY_WAIT Subsector Erase 時 ; BusyCnt = FLASH_SSE_BUSY_WAIT リターン値が「FLASH_OK」以外の場合、ステータスレジスタ 1 を読み出し、状態を確認してください。 リターン値が「FLASH_ERR_OTHER」の場合、かつステータスレジスタ 1 の E_ERR ビットがセットされている場合、消去エラーもしくはプロテクトされた領域への消去が発生しています。次の処理 (書き込み/読み出し/消去等) の前にクリアステータス処理 (R_QSPI_FLASH_Clear_Status()) を行ってください。また、ステータスレジスタ 1 の WEL ビットがセットされている場合、クリアするために、WRDI コマンド発行処理 (R_QSPI_FLASH_Write_Di()) も行ってください。
引数	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_B_ERASE ; FLASH_MODE_S_ERASE ; FLASH_MODE_SS_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 ; Time out error (「FLASH_WAIT_READY」有効時) FLASH_ERR_OTHER ; Other error

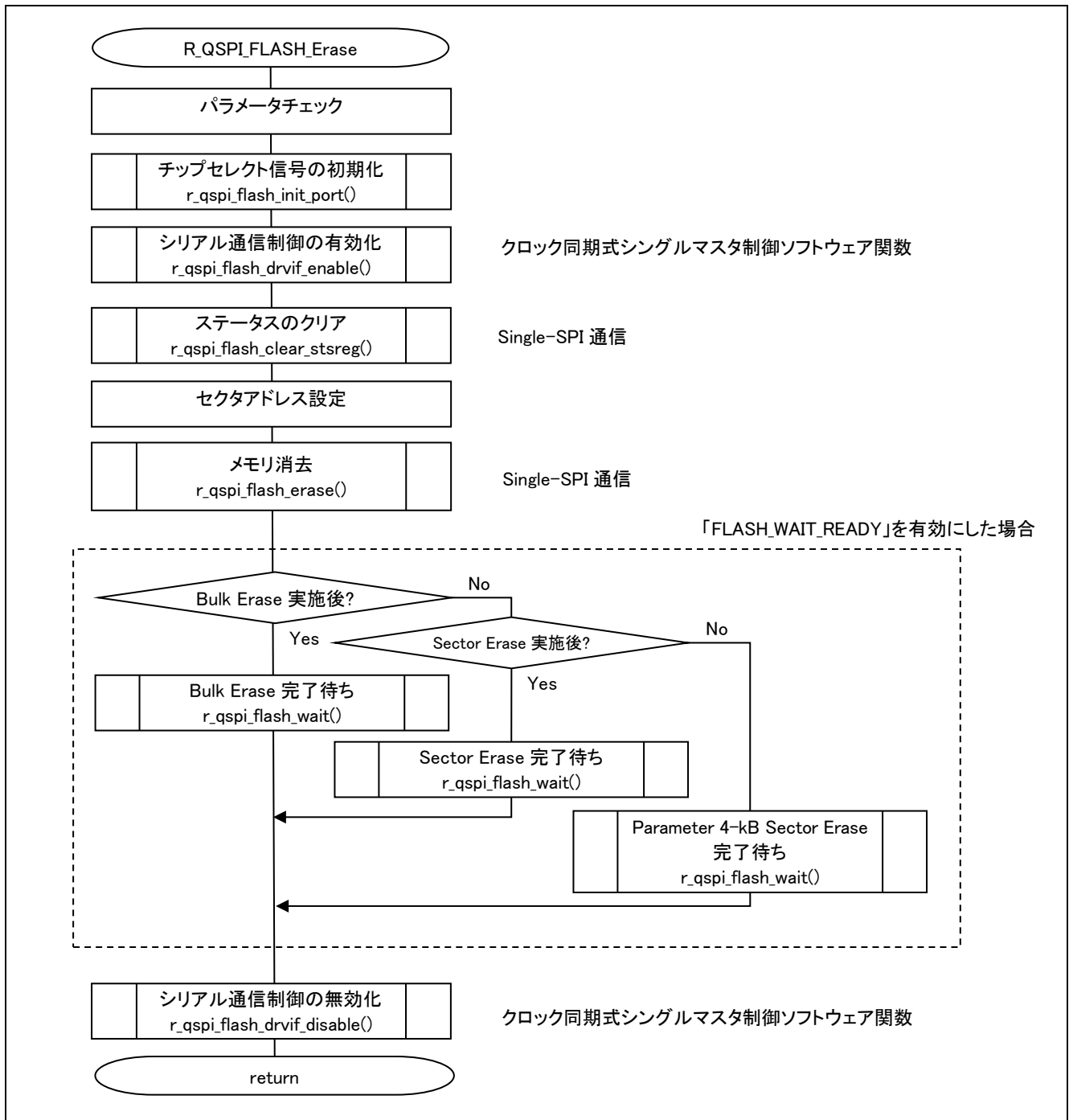


図 5-23 消去処理概要

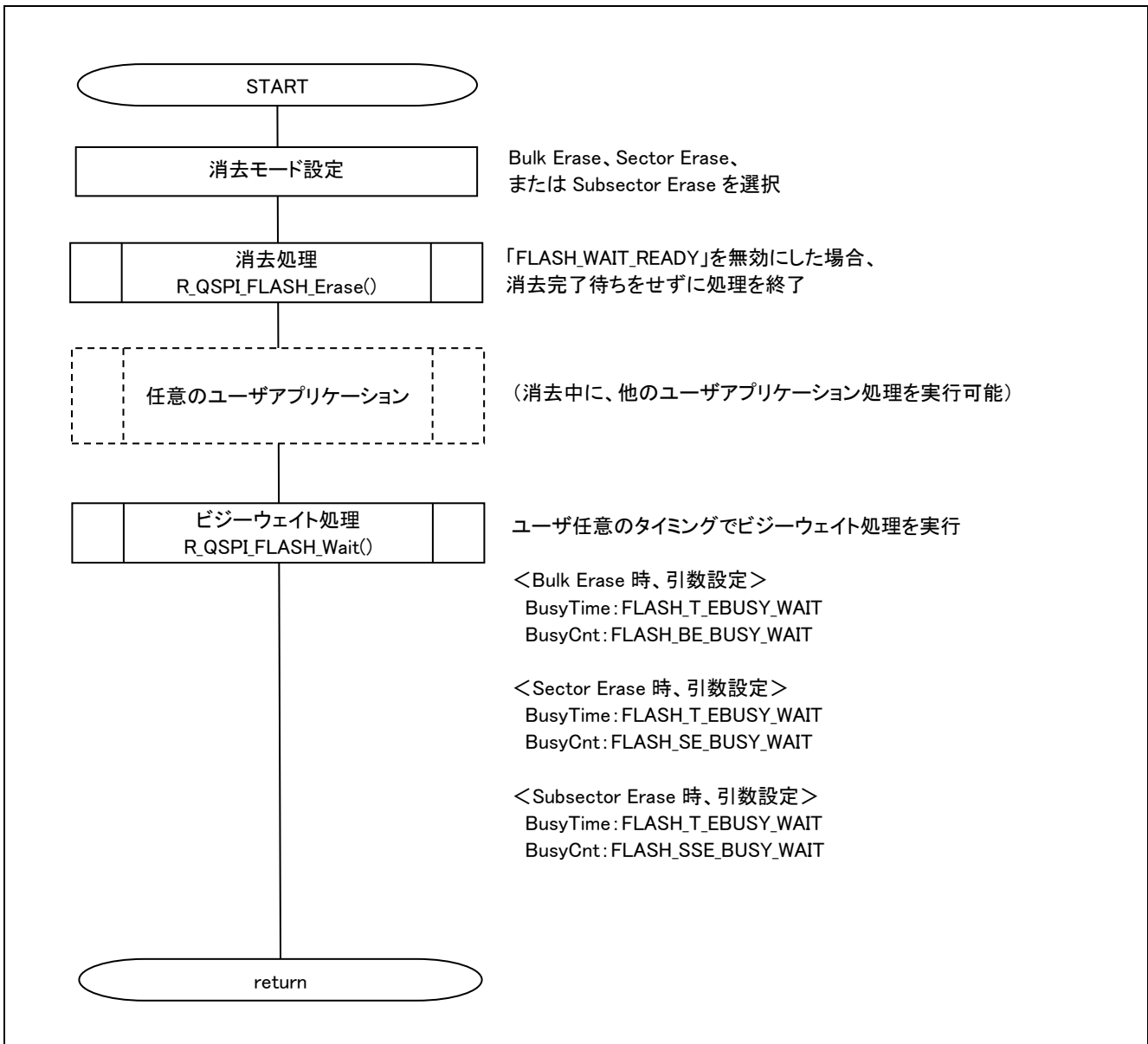


図 5-24 R_QSPI_FLASH_Wait()を使用した消去処理の完了待ち方法

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h, r_qspi_flash_s25fl_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、2 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

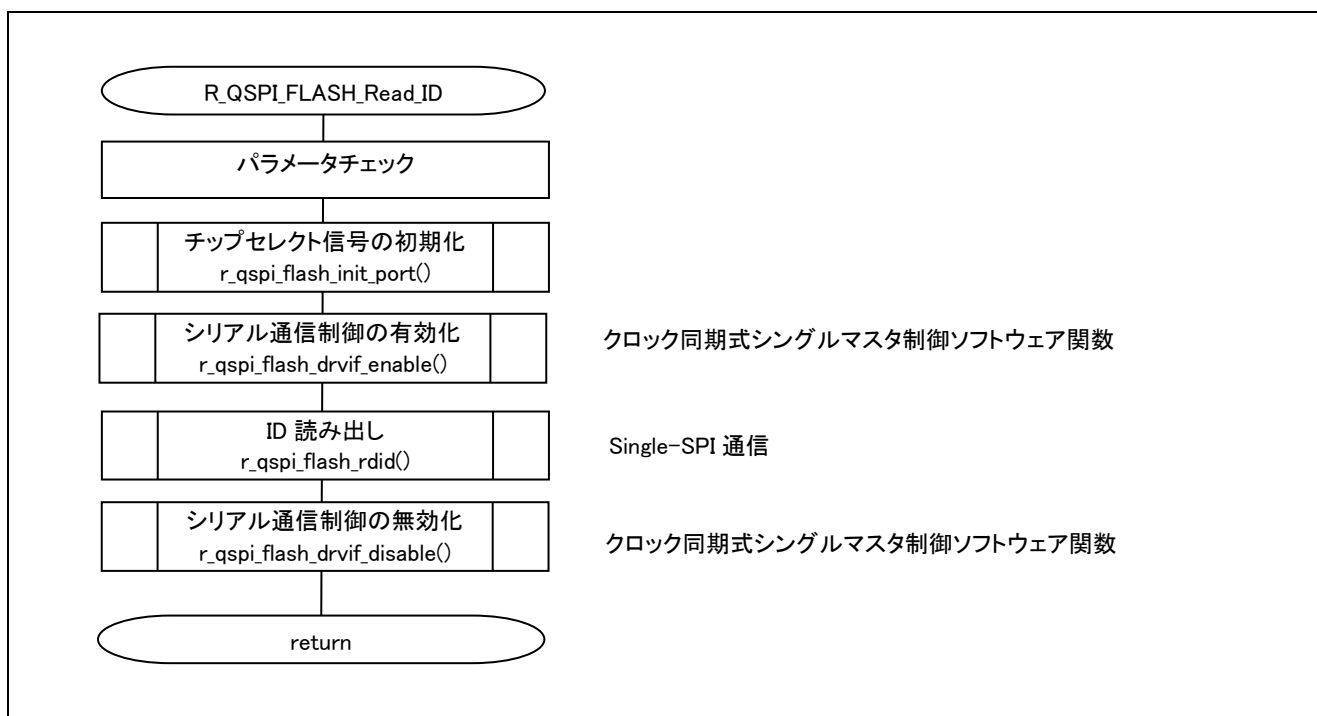


図 5-25 ID 読み出し処理概要

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_s25fl.h, r_qspi_flash_s25fl_sub.h, r_qspi_flash_s25fl_sfr.h,
r_qspi_flash_s25fl_drvif.h

宣言 error_t R_QSPI_FLASH_Wait(uint8_t DevNo, uint16_t BusyTime, uint32_t BusyCnt)

説明

- ・「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。
- ・データ書き込み及び消去モードの場合、レディ判定後ステータスレジスタ 1 のエラービット (P_ERR, E_ERR) を確認します。いずれかのエラービットが 1 の場合、「FLASH_ERR_OTHER」を返します。
- ・コンフィグレーションレジスタの OTP ビットが"1"にも関わらず"0"を設定し書き込みを行った場合、「FLASH_ERR_TIMEOUT」を返します。
- ・リターン値が「FLASH_ERR_OK」以外の場合、各書き込み／消去処理関数の説明にしたがって、リカバリー処理を行ってください。
- ・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処理を行うと、「FLASH_ERR_TIMEOUT」を返す可能性があります。下表にしたがって設定してください。

状態	BusyTime	BusyCnt
デバイス初期化中	FLASH_T_R_ACCESS	FLASH_WBUSY_WAIT
ステータスレジスタ書き込み 中(ライトプロテクトビットセット)	FLASH_T_WBUSY_WAIT	FLASH_WBUSY_WAIT
データ書き込み中	FLASH_T_WBUSY_WAIT	FLASH_PBUSY_WAIT
消去中 (Bulk Erase)	FLASH_T_EBUSY_WAIT	FLASH_BE_BUSY_WAIT
消去中 (Sector Erase)	FLASH_T_EBUSY_WAIT	FLASH_SE_BUSY_WAIT
消去中 (Subsector Erase)	FLASH_T_EBUSY_WAIT	FLASH_SSE_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／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_WBUSY_WAIT: レジスタ／データ書き込み時
; FLASH_BE_BUSY_WAIT: 消去時 (Bulk Erase)
; FLASH_SE_BUSY_WAIT: 消去時 (Sector Erase)
; FLASH_SSE_BUSY_WAIT: 消去時 (Subsector Erase)

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 ; Time out error (BusyCnt≠0 の場合)

FLASH_ERR_OTHER ; Other error

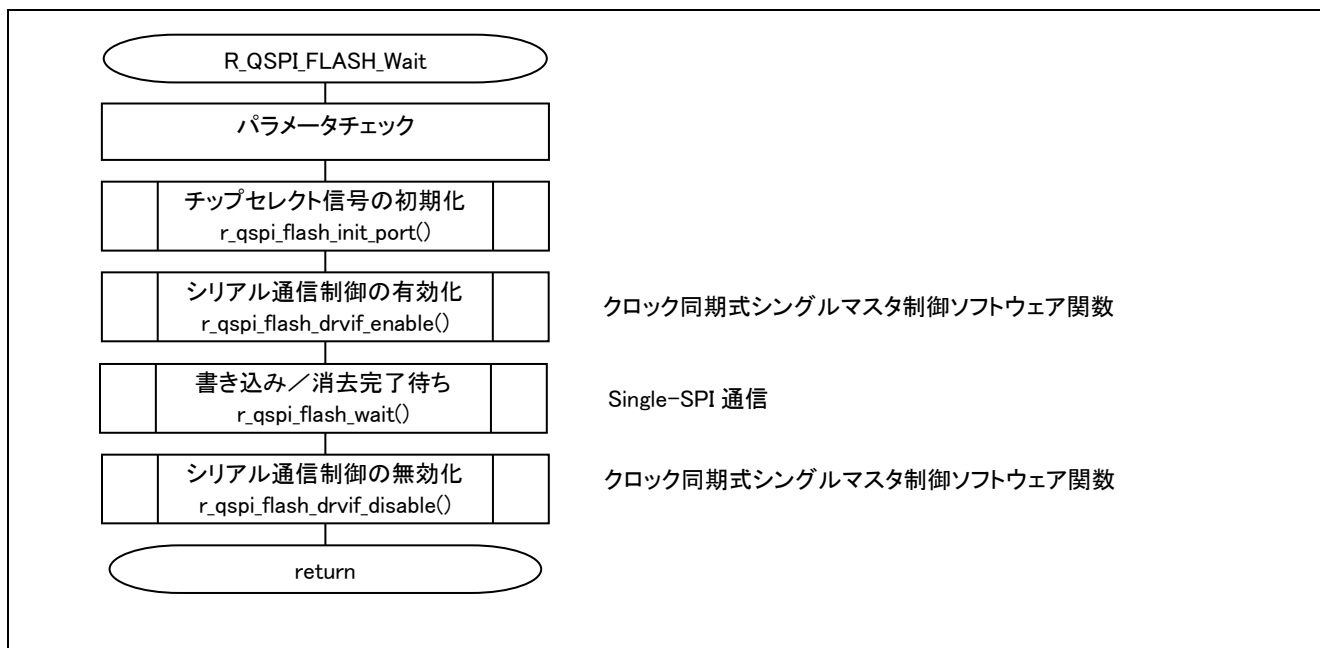


図 5-26 ビジーウェイト処理概要

6. 応用例

S25FLxxxS MirrorBit® Flash Non-Volatile 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 S25FLxxxS MirrorBit® Flash Non-Volatile 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_s25fl.h

本 S25FLxxxS MirrorBit® Flash Non-Volatile 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 serial FLASH devices. (1~N devices) */
/* Define the device number in accordance with the number of serial FLASH devices*/
/* to be connected. */
/*-----*/
/* Define no. 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、セクタサイズが 64kByte の 128Mbit デバイスを使用する場合の例です。

```

/*-----*/
/* Define the serial FLASH device. */
/*-----*/

#define S25FL128SXXXXXX0X      /* 128Mbit (Uniform 64-kB sectors) */
// #define S25FL128SXXXXXX1X   /* 128Mbit (Uniform 256-kB sectors) */
// #define S25FL256SXXXXXX0X   /* 256Mbit (Uniform 64-kB sectors) */
// #define S25FL256SXXXXXX1X   /* 256Mbit (Uniform 256-kB sectors) */

```


(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/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:

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

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())

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_s25fl_sfr.h

r_qspi_flash_s25fl_sfr.h.XXX は、MCU 毎に作成したものです。どれか一つを r_qspi_flash_s25fl_sfr.h にリネームして使用してください。対象 MCU のものが無い場合には、参照して、r_qspi_flash_s25fl_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、RX63N でポート A0 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    PORTA.PODR.BIT.B0  /* FLASH CS0 (Negative-true logic) */
#define FLASH_DDR_CS0  PORTA.PDR.BIT.B0   /* 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 CC-RL 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) 通信速度の設定

通信速度を設定してください。単位は、bit/sec です。

設定値は、使用する MCU とシリアル I/O に依存します。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 通信速度設定

#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、RX63N の RSPI を使用する場合の例です。

```

/* PCLK = 48MHz, n=0 for RX63N RSPI */
#define FLASH_BR          (uint8_t) (0x01) /* SPBR initial setting */
/*          ++----- 12.00MHz */

/* PCLK = 48MHz, n=0 for RX63N RSPI Write Data */
#define FLASH_BR_WRITE_DATA (uint8_t) (0x02) /* SPBR initial setting */
/*          ++----- 12.00MHz */

/* PCLK = 48MHz, n=0 for RX63N RSPI Read Data */
#define FLASH_BR_READ_DATA  (uint8_t) (0x01) /* SPBR initial setting */
/*          ++----- 12.00MHz */

```

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_s25fl_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

メモリの全データ消去 (Bulk Erase) と指定されたセクタの全データ消去 (Sector Erase) と指定されたサブセクタの全データ消去 (SubSector Erase) のタイムアウト時間を設定します。

デバイスにより、消去時間が異なる場合、以下の設定を見直してください。

以下は、Bulk Erase のタイムアウト時間を 330s、Sector Erase のタイムアウト時間を 10.4s、SubSector Erase のタイムアウト時間を 650ms に設定する場合での記述例を示します。

```

/*-----*/
/* Define the software timer value of erase or page program busy waiting. */
/*-----*/

/*----- Definitions of software timer value -----*/
/* Bulk Erase      : 330s (According to the S25FL256S)          */
/* Sector Erase    : 10.4s (According to the 64 kB Top/Bottom) */
/* Parameter 4-kB Sector Erase : 650ms                        */
/* Page Program    : 750us                                       */
/* WRR Write Time  : 500ms                                       */
#define FLASH_BE_BUSY_WAIT (uint32_t) (330000)
/* Bulk Erase busy timeout 330,000*1ms = 330s */
#define FLASH_SE_BUSY_WAIT (uint32_t) (10400)
/* Sector Erase busy timeout 10,400*1ms = 10.4s */
#define FLASH_SSE_BUSY_WAIT (uint32_t) (650)
/* Parameter 4-kB Sector Erase busy timeout */
/* 650*1ms = 650ms */

```

(2) 書き込みタイムアウト時間とデバイス初期化タイムアウト時間設定

デバイスにより、書き込み時間やデバイス初期化時間が異なる場合、以下の設定を見直してください。

以下は、ページプログラム書き込みタイムアウト時間を 750us、レジスタ書き込みタイムアウト時間を 500ms に設定する場合での記述例を示しています。

```

#define FLASH_PBUSY_WAIT (uint32_t) (750)
/* Page Program ready timeout 750*1us = 750us */
#define FLASH_WBUSY_WAIT (uint32_t) (500)
/* Register write ready timeout 500*1ms = 500ms */

```

6.1.4 r_qspi_flash_s25fl_sub.c

本 S25FLxxxS MirrorBit® Flash Non-Volatile 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。

6.1.5 r_qspi_flash_s25fl_drvif.c

本 S25FLxxxS MirrorBit® Flash Non-Volatile 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_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);
}
```

(9) 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);
}
```

(10) 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。主にデータ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_data(uint16_t RxCnt, uint8_t FAR * pData)
{
    return R_SIO_Rx_Data(RxCnt, pData);
}
```


6.1.6 r_qspi_flash_s2fl_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。

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

(1) SFR 領域用デファインの設定

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

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

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

本設定は、SPI スレーブデバイスセレクト制御信号用に使用します。

表 6-2 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	<pre>#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</pre>
	78K0R	要	<pre>#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</pre>
	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_s25fl.h  
r_qspi_flash_s25fl_sub.h  
r_qspi_flash_s25fl_sfr.h  
r_qspi_flash_s25fl_drvif.h
```

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_SSECT_ADDR  
FLASH_PAGE_SIZE  
FLASH_ADDR_SIZE  
FLASH_E_ADDR_SIZE  
FLASH_WP_WHOLE_MEM  
FLASH_SECTOR_TYPE
```

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 制御コマンドの使い分けについて

本サンプルコードでは、128Mbit 品は 3Byte addressing コードコマンド、256Mbit 品は 4Byte addressing コードコマンドにより制御します。

7.6 コンフィグレーションレジスタの Bit3 (BPNV) とライトプロテクト

ステータスレジスタのライトプロテクトビットである Bit 2,3,4 (BP0, BP1, BP2) は、コンフィグレーションレジスタの Bit3 (BPNV) により、Volatile または Non-Volatile を選択することができます。デフォルトでは BPNV=0 (Non-Volatile) が選択されています。

BPNV は OTP ビットです。一度書き換えると、以降の書き換えは受け付けませんので、ご注意ください。

7.7 ライトプロテクト領域の選択と注意事項

ライトプロテクト領域は、ステータスレジスタのライトプロテクトビット Bit 2,3,4 (BP0, BP1, BP2) とコンフィグレーションレジスタの bit5 (TBPROT) により決まります。

TBPROT は、メモリの上位アドレス領域と下位アドレス領域のプロテクトを選択するためのビットです。デフォルトでは、上位アドレス領域 (TBPROT=0) が選択されています。下位アドレス領域へプロテクトを行う場合、TBPROT=1 にセットしてください。

ただし、TBPROT は OTP ビットです。一度書き換えると、以降の書き換えは受け付けませんので、ご注意ください。

7.8 RESET コマンドの注意事項

RESET コマンドは強制的に発行することができます。データ書き込みや消去完了待ち中に発行した場合、処理は中断され、処理対象エリアのデータは不定になります。

7.9 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

Spansion 社製 S25FLxxxS MirrorBit[®] Flash Non-Volatile 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/>

お問い合わせ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.06.14	—	初版発行
1.02	2013.11.29	6	2. 動作確認条件 に、以下を追加 (3) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench の場合
		18	5.3 必要メモリサイズ に、以下を追加 (3) RL78/G14 SAU 統合開発環境 IAR Embedded Workbench の場合
		19	5.4 ファイル構成 アプリケーションノート番号を変更
		57	6.1.2 r_qspi_pcm_p5q_sfr.h 内容を修正した。
		64	6.1.6 r_qspi_pcm_p5q_sfr_rl78.c を新規追加した。
1.03	2014.04.30	1	要旨 にクロック同期式シングルマスタ制御ソフトウェア情報の URL を追加した。
		1	対象デバイス に、RX111、RL78/G1C、RL78/L1C、RL78/L12、RL78/L13 を追加
		5,7	2. 動作確認条件 2.1 RX ファミリと 2.2 RL78 ファミリ、78K0R/Kx3-L を追加した。
		6	2.1 RX ファミリに、以下を追加 (2) RX111 RSPI の場合 (3) RX111 SCI の場合
		8-11	2.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加 (3) RL78/G1C SAU 統合開発環境 CubeSuite+の場合 (4) RL78/G1C SAU 統合開発環境 IAR Embedded Workbench の場合 (5) RL78/L12 SAU 統合開発環境 CubeSuite+の場合 (6) RL78/L12 SAU 統合開発環境 IAR Embedded Workbench の場合 (7) RL78/L13 SAU 統合開発環境 CubeSuite+の場合 (8) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合 (9) RL78/L1C SAU 統合開発環境 CubeSuite+の場合 (10) RL78/L1C SAU 統合開発環境 IAR Embedded Workbench の場合
		12	3.関連アプリケーションノート 以下を更新した。 RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) 元は、RX210, RX21A, RX220, RX63N, RX63T グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ)であった。 RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア (R01AN1229JJ) 元は、RX210, RX21A, RX220, RX63N, RX63T グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1229JJ)であった。 RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C 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) 元は、RL78/G14 シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア

			(R01AN1195JJ)であった。
		22,24	5.3 必要メモリサイズ 5.3.1 RX ファミリと 5.3.2 RL78 ファミリ、78K0R/Kx3-L の章を追加し、ファミリ別に分けた。
		23	5.3.1 RX ファミリに、以下を追加 (2) RX111 RSPI の場合 (3) RX111 SCI の場合
		25	5.3.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加 (3) RL78/L13 SAU 統合開発環境 CubeSuite+ の場合 (4) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合
		26	5.4 ファイル構成 アプリケーションノート番号を変更 ソースのフォルダ名を変更 新規デバイスのレジスタ用共通定義を追加
1.04	2016.03.31	7	2. 動作確認条件 2.2 RL78 ファミリ、78K0R/Kx3-L (1) RL78/G14 SAU 統合開発環境 CS+ for CA,CX の場合（コンパイラ：CA78K0R）で情報を更新した。 (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。
		25	5.3 必要メモリサイズ 5.3.2 RL78 ファミリ、78K0R/Kx3-L (1) RL78/G14 SAU 統合開発環境 CS+ for CA,CX の場合（コンパイラ：CA78K0R）で情報を更新した。 (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。
		28	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。
		66-67	6.1.2 r_qspi_flash_s25fl_sfr.h 記述例を更新した。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

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

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

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



ルネサスエレクトロニクス株式会社

営業お問合せ窓口

<http://www.renesas.com>

営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

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