

# RX ファミリ、RL78 ファミリ、78K0R/Kx3-L R01AN1528JJ0104 Rev.1.04

## Micron Technology 社製 N25Q Serial NOR Flash 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 Micron Technology, Inc. 社製 N25Q 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](http://japan.renesas.com/driver/spi_serial_flash)

### 動作確認デバイス

Serial NOR Flash memory      Micron Technology, Inc. 社製 N25Q Serial NOR Flash 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	14
3.1 RX ファミリ 関連アプリケーションノート一覧	14
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	14
4. ハードウェア説明	15
4.1 ハードウェア構成例	15
4.1.1 Single-SPI 使用時の端子一覧	15
4.1.2 Single-SPI 使用時の接続例	15
4.1.3 Dual-SPI 使用時の端子一覧	16
4.1.4 Dual-SPI 使用時の接続例	16
4.1.5 Quad-SPI 使用時の端子一覧	17
4.1.6 Quad-SPI 使用時の接続例	17
5. ソフトウェア説明	18
5.1 動作概要	18
5.1.1 データバッファと送信/受信データの関係	18
5.1.2 クロック同期式モードで発生させるタイミング	19
5.1.3 Serial NOR Flash memory の S#端子制御	21
5.1.4 Serial NOR Flash memory の命令コード	22
5.2 ソフトウェア構成	23
5.3 必要メモリサイズ	24
5.3.1 RX ファミリ	24
5.3.2 RL78 ファミリ、78K0R/Kx3-L	26
5.4 ファイル構成	30
5.5 定数一覧	31
5.5.1 戻り値	31
5.5.2 コマンド定義	31
5.5.3 各種定義	32
5.6 構造体/共用体一覧	35
5.7 変数一覧	36
5.8 関数一覧	36
5.9 関数仕様	37
5.9.1 ドライバ初期化処理	37
5.9.2 ステータスレジスタ読み出し処理	38
5.9.3 フラグステータスレジスタ読み出し処理	40
5.9.4 ライトプロテクト設定処理	42
5.9.5 クリアフラグステータス処理	45
5.9.6 WRDI コマンド発行処理	46
5.9.7 データ読み出し処理	47

---

5.9.8	データ書き込み処理.....	48
5.9.9	データ書き込み処理（1Page 書き込み用）.....	50
5.9.10	消去処理.....	53
5.9.11	ID 読み出し処理.....	56
5.9.12	ビジーウェイト処理.....	57
5.9.13	アドレスモード設定処理.....	59
6.	応用例.....	60
6.1	Serial NOR Flash memory 制御ソフトウェアの設定.....	61
6.1.1	r_qspi_flash_n25q.h.....	61
6.1.2	r_qspi_flash_n25q_sfr.h.....	63
6.1.3	r_qspi_flash_n25q_sub.h.....	66
6.1.4	r_qspi_flash_n25q_sub.c.....	67
6.1.5	r_qspi_flash_n25q_drvif.c.....	68
6.1.6	r_qspi_flash_n25q_sfr_rl78.c.....	72
7.	使用上の注意事項.....	73
7.1	組み込み時の注意事項.....	73
7.2	キャッシュ搭載の MCU を使用する場合.....	73
7.3	他容量に対応する場合.....	73
7.4	他スレーブデバイスを使用する場合.....	73
7.5	電源投入後の電圧安定待ち時間について.....	74
7.6	N25Q256A83ESF40x と N25Q256A83E1240x の未対応コマンド（ENTER or EXIT QUAD Command）について.....	74

## 1. 仕様

ルネサス エレクトロニクス製 MCU を使用し、Micron Technology, Inc.社製 N25Q 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 Micron Technology, Inc.社製 N25Q 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）

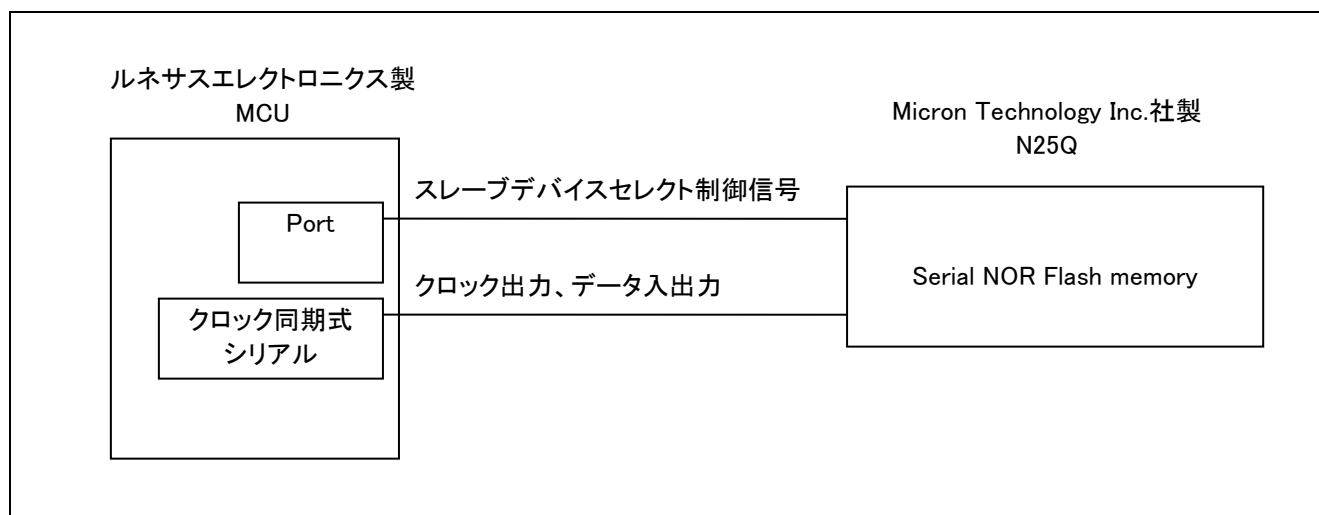


図 1.1 使用例

## 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	Micron Technology, Inc.社製 N25Q Serial NOR Flash 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	Micron Technology, Inc.社製 N25Q 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、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	Micron Technology, Inc.社製 N25Q 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、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 CubeSuite+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	Micron Technology, Inc.社製 N25Q 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+ V1.03.00
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。
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	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 2-5 動作確認条件

項目	内容
評価に使用したメモリ	Micron Technology, Inc.社製 N25Q 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+ 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。
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	Micron Technology, Inc.社製 N25Q 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。
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 2-7 動作確認条件

項目	内容
評価に使用したメモリ	Micron Technology, Inc.社製 N25Q 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。
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 CS+ for CC の場合 (コンパイラ : CC-RL)

表 2-8 動作確認条件

項目	内容
評価に使用したメモリ	Micron Technology, Inc.社製 N25Q 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+ 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。
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

(6) RL78/G1C SAU 統合開発環境 IAR Embedded Workbench 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	Micron Technology, Inc.社製 N25Q 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。
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

## (7) RL78/L12 SAU 統合開発環境 CubeSuite+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	Micron Technology, Inc.社製 N25Q 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。
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/L12 SAU 統合開発環境 IAR Embedded Workbench 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	Micron Technology, Inc.社製 N25Q 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。
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

## (9) RL78/L13 SAU 統合開発環境 CubeSuite+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	Micron Technology, Inc.社製 N25Q 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。
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/L13 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	Micron Technology, Inc.社製 N25Q 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。
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

## (11) RL78/L1C SAU 統合開発環境 CubeSuite+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	Micron Technology, Inc.社製 N25Q 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。
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

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

表 2-15 動作確認条件

項目	内容
評価に使用したメモリ	Micron Technology, Inc.社製 N25Q 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。
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。

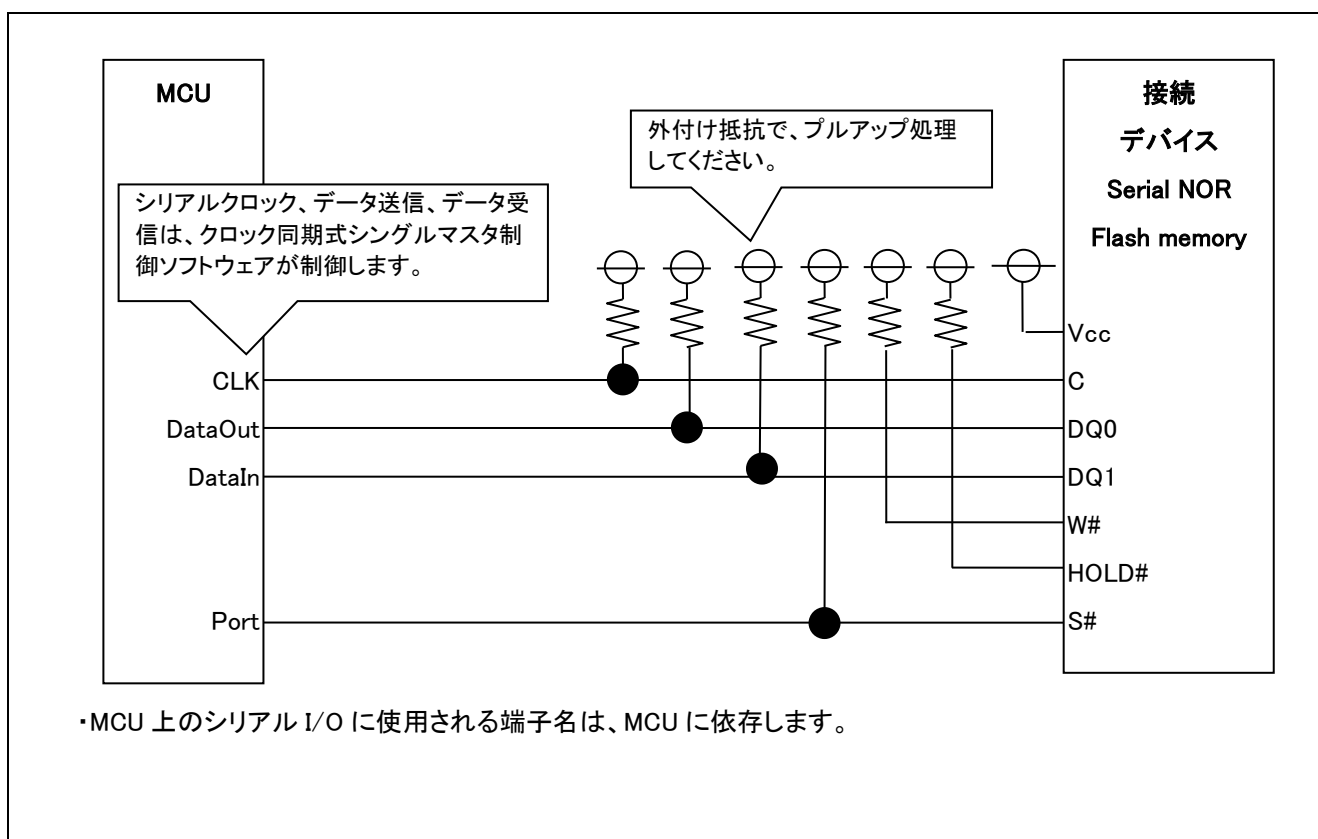


図 4-1 Single-SPI 使用時の MCU と SPI スレーブデバイスの接続例

## 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。

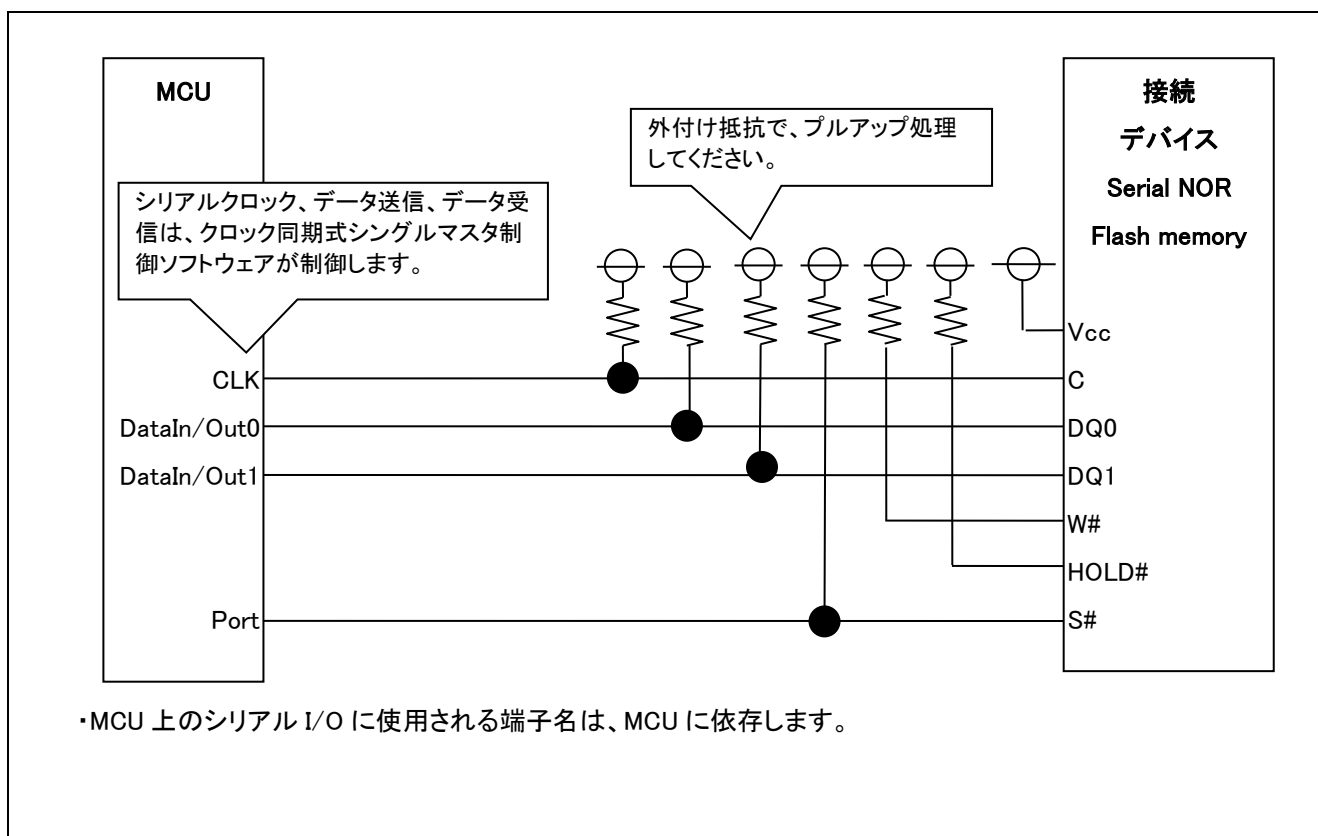


図 4-2 Dual-SPI 使用時の MCU と SPI スレーブデバイスの接続例



## 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。

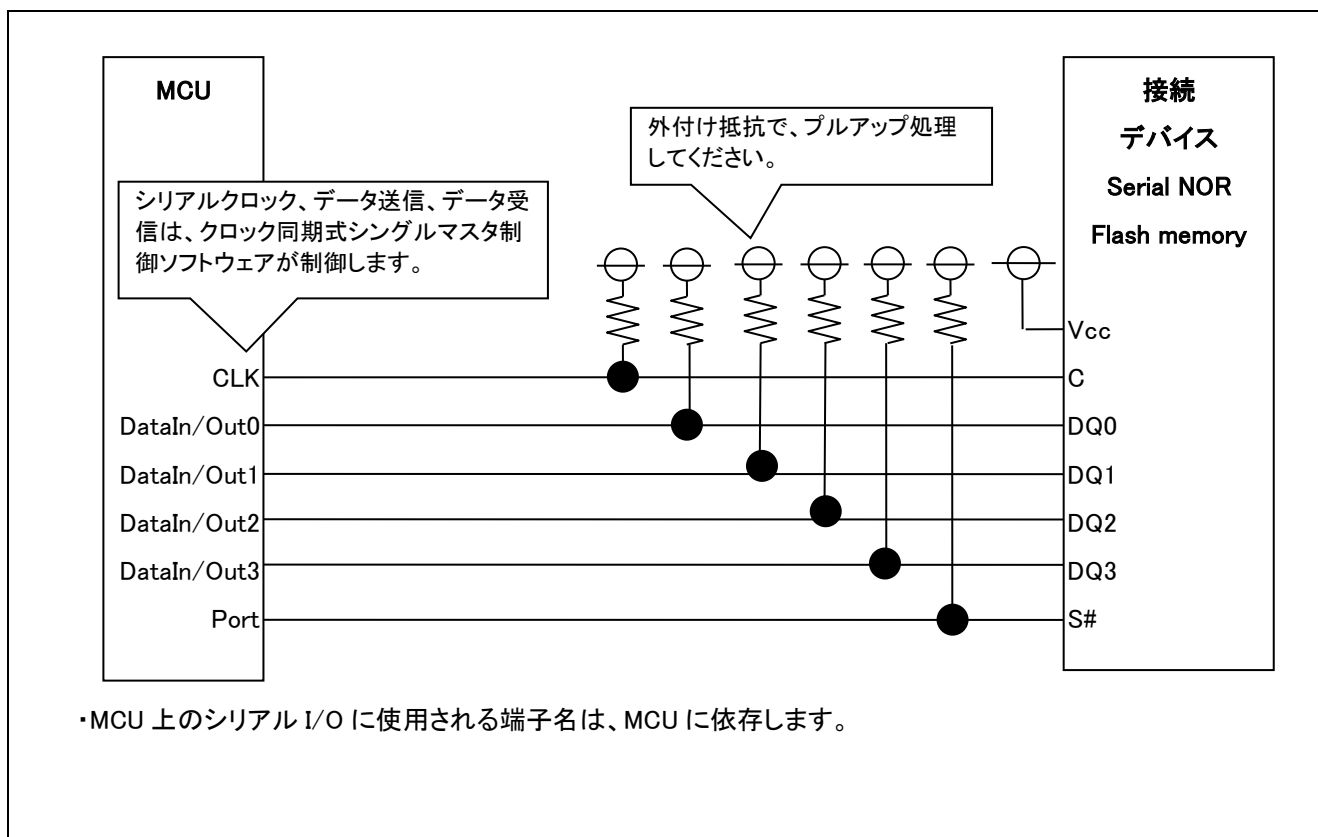


図 4-3 Quad-SPI 使用時の MCU と SPI スレーブデバイスの接続例

## 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、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。

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

#### 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。

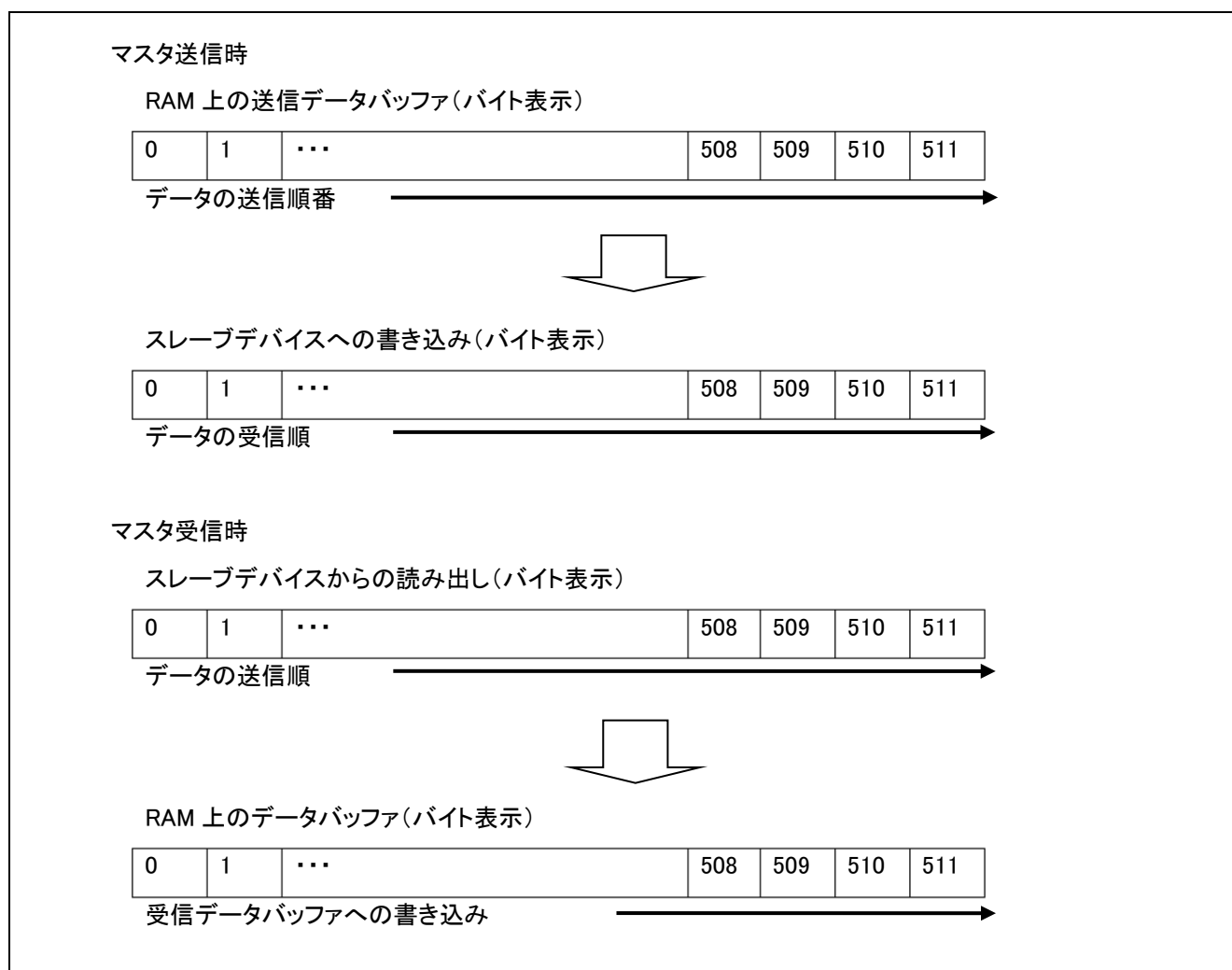


図 5.1 転送データの格納

## 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

Serial NOR Flash 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。

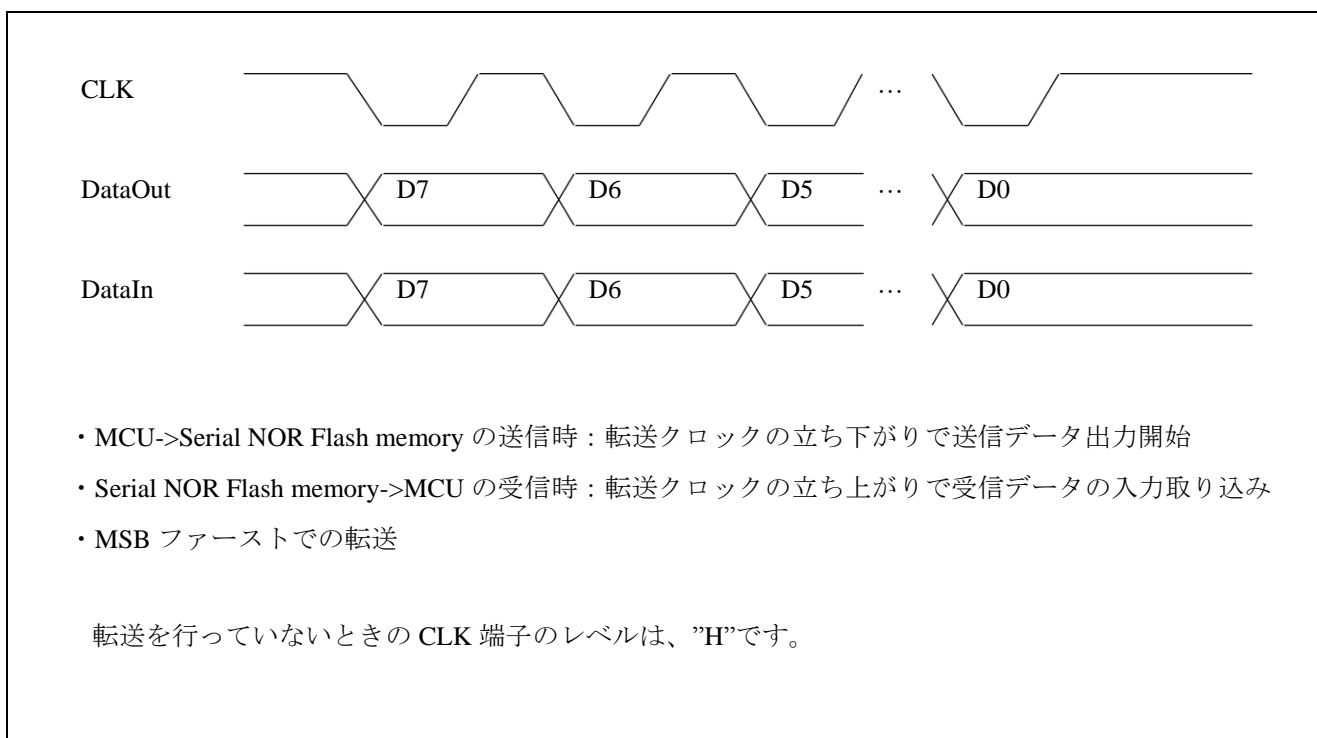


図 5-2 Single-SPI 時のクロック同期式モード タイミング設定

## (2) Dual-SPI 使用時

Serial NOR Flash 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。

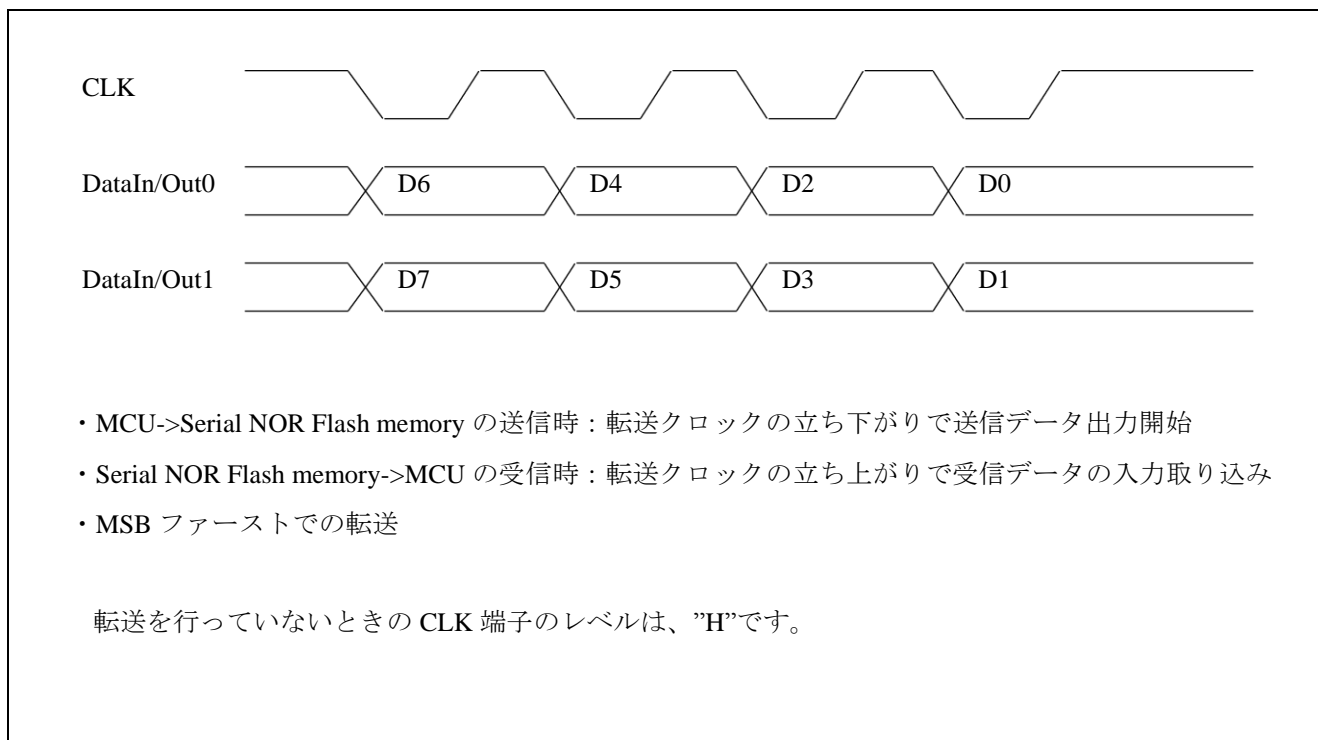


図 5-3 Dual-SPI 時のクロック同期式モード タイミング設定

## (3) Quad-SPI 使用時

Serial NOR Flash 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。

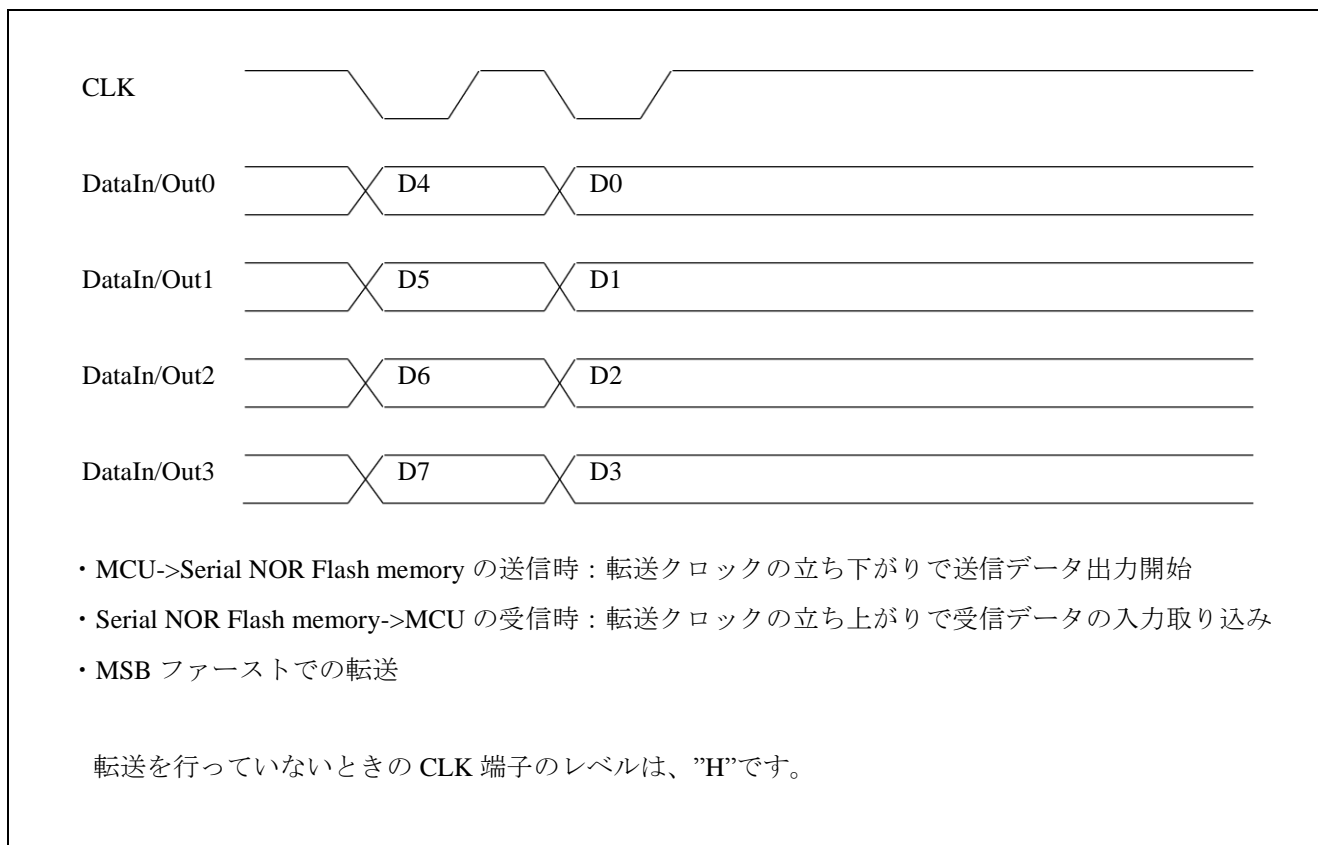


図 5-4 Quad-SPI 時のクロック同期式モード タイミング設定

### 5.1.3 Serial NOR Flash memory の S#端子制御

Serial NOR Flash memory の S#端子を MCU の Port に接続し、MCU 汎用ポート出力で、制御します。

Serial NOR Flash memory の S# (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 S#セットアップ時間待ちのために、ソフトウェア・ウェイトで制御しています。

Serial NOR Flash memory の C (MCU の CLK) 信号の立ち上がりから、Serial NOR Flash memory の S# (MCU の Port(CS#)) 信号の立ち上がりまでの時間は、Serial NOR Flash memory の S#ホールド時間待ちのために、ソフトウェア・ウェイトで制御しています。

Serial 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)
RDFSR	Read Flag Status Register	0111 0000 (70 h)
CLFSR	Clear Flag Status Register	0101 0000 (50 h)
FAST_READ	Read Data at Higher Speed	0000 1011 (0b h)
DOFR	Dual Output Fast Read	0011 1011 (3b h)
QOFR	Quad Output Fast Read	0110 1011 (6b h)
PP	Page Program	0000 0010 (02 h)
DIFP	Dual Input Fast Program	1010 0010 (a2 h)
QIFP	Quad Input Fast Program	0011 0010 (32 h)
SE	Sector Erase	1101 1000 (d8 h)
SSE	Subsector Erase	0010 0000 (20 h)
BE	Bulk Erase	1100 0111 (c7 h)
DE	Die Erase	1100 0100 (c4 h)
RDID	Read Identification	1001 1111 (9f h)
ENTER4	Enter 4-Byte Address Mode	1011 0111 (b7 h)
EXIT4	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、SPI 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。

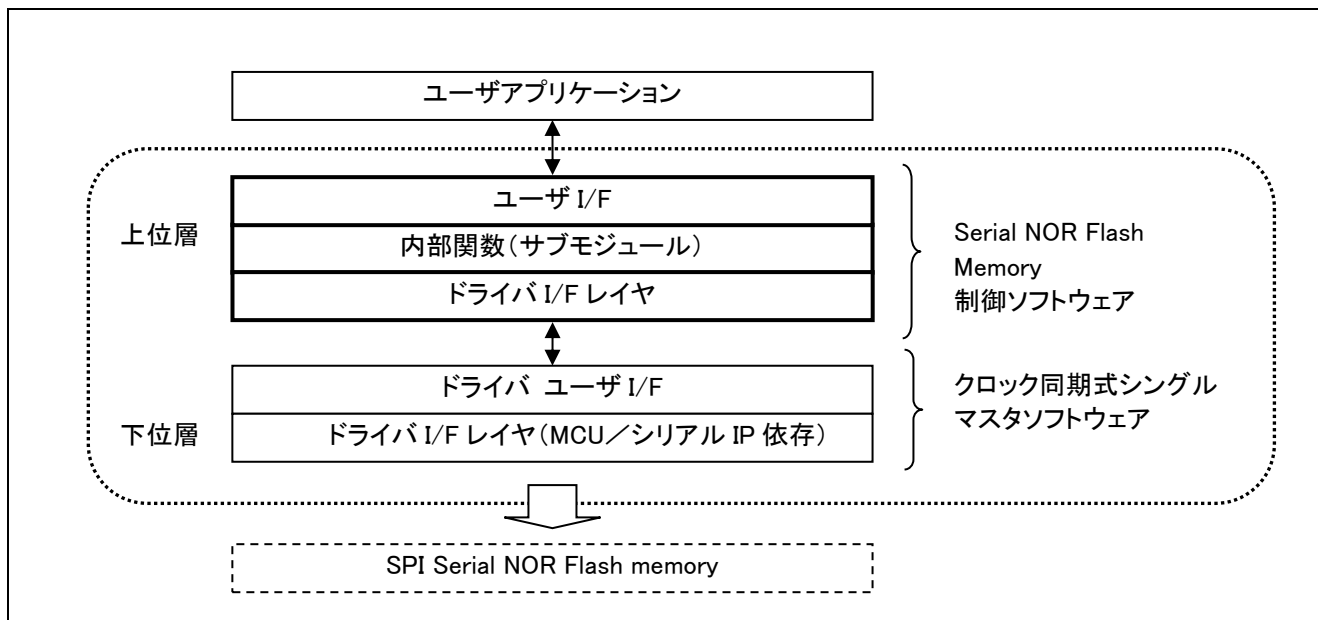


図 5-5 ソフトウェア構成

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

以下に必要なメモリサイズを示します。

#### 5.3.1 RX ファミリ

環境は、「2.動作確認条件」を参照してください。

##### (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	3,760 バイト (リトルエンディアン)	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.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、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	3,792 バイト (リトルエンディアン)	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.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	3,792 バイト (リトルエンディアン)	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c
RAM	6 バイト (リトルエンディアン)	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c
最大使用ユーザスタック	156 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 CubeSuite+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	6,211 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.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 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	5,020 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.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 統合開発環境 IAR Embedded Workbench の場合

表 5-7 必要メモリサイズ (Embedded Workbench の場合)

使用メモリ	サイズ	備考
ROM	6,907 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
最大使用ユーザスタック	154 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 5-8 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	6,309 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.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。

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

表 5-9 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,024 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.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。

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

表 5-10 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	5,398 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
RAM	6 バイト	r_qspi_flash_n25q_usr.c r_qspi_flash_n25q_sub.c r_qspi_flash_n25q_drvif.c r_qspi_flash_n25q_sfr_rl78.c
最大使用ユーザスタック	132 バイト	
最大使用割り込みスタック	—	割り込み未使用

【注】 必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。  
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 5-11 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 5-11 ファイル構成

¥an_r01an1528jj0104_mcu_serial	<DIR>	サンプルコードのフォルダ
r01an1528jj0104_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_n25q	<DIR>	Serial NOR Flash memory 制御ソフトウェア用フォルダ
r_qspi_flash_n25q.h		ヘッダファイル
r_qspi_flash_n25q_drvif.c		ドライバ I/F ソースファイル
r_qspi_flash_n25q_drvif.h		ドライバ I/F ヘッダファイル
r_qspi_flash_n25q_sfr.h.rl78g14		レジスタ用共通定義 (RL78/G14 用)
r_qspi_flash_n25q_sfr.h.rl78g1c		レジスタ用共通定義 (RL78/G1C 用)
r_qspi_flash_n25q_sfr.h.rl78l1c		レジスタ用共通定義 (RL78/L1C 用)
r_qspi_flash_n25q_sfr.h.rl78l12		レジスタ用共通定義 (RL78/L12 用)
r_qspi_flash_n25q_sfr.h.rl78l13		レジスタ用共通定義 (RL78/L13 用)
r_qspi_flash_n25q_sfr.h.rx63n		レジスタ用共通定義 (RX63N 用)
r_qspi_flash_n25q_sfr.h.rx111		レジスタ用共通定義 (RX111 用)
r_qspi_flash_n25q_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_n25q_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_n25q_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_n25q_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_n25q_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_n25q_sub.c		内部関数ソースファイル
r_qspi_flash_n25q_sub.h		内部関数ヘッダファイル
r_qspi_flash_n25q_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 5-12 にサンプルコードで使用する戻り値を示します。

表 5-12 戻り値 (r\_qspi\_flash\_n25q.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)	Time out error
FLASH_ERR_OTHER	(error_t)(-7)	Other error

### 5.5.2 コマンド定義

表 5-13 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 5-13 コマンド定義 (r\_qspi\_flash\_n25q\_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_RDFSR	(uint8_t)(0x70)	Read Flag Status Register
FLASH_CMD_CLFSR	(uint8_t)(0x50)	Clear Flag Status Register
FLASH_CMD_FREAD	(uint8_t)(0x0b)	Read Data at Higher Speed
FLASH_CMD_DOFR	(uint8_t)(0x3b)	Dual Output Fast Read
FLASH_CMD_QOFR	(uint8_t)(0x6b)	Quad Output Fast Read
FLASH_CMD_PP	(uint8_t)(0x02)	Page Program
FLASH_CMD_DIFP	(uint8_t)(0xa2)	Dual Input Fast Program
FLASH_CMD_QIFP	(uint8_t)(0x32)	Quad Input Fast Program
FLASH_CMD_SE	(uint8_t)(0xd8)	Sector Erase
FLASH_CMD_SSE	(uint8_t)(0x20)	Subsector Erase
FLASH_CMD_BE	(uint8_t)(0xc7)	Bulk Erase
FLASH_CMD_DE	(uint8_t)(0xc4)	Die Erase
FLASH_CMD_RDID	(uint8_t)(0x9f)	Read Identification
FLASH_CMD_ENTER4	(uint8_t)(0xb7)	Enter 4-Byte Address Mode
FLASH_CMD_EXIT4	(uint8_t)(0xe9)	Exit 4-Byte Address Mode

## 5.5.3 各種定義

表 5-14 から表 5-18 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 5-14 r\_qspi\_flash\_n25q.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_D_ERASE	(uint8_t)(4)	Erase Mode : Dei Erase
FLASH_MODE_3BYTE	(uint8_t)(0)	Addressability Mode : 3-byte addressability Mode
FLASH_MODE_4BYTE	(uint8_t)(1)	Addressability Mode : 4-byte addressability 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)(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)(0xffff000)	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)(64)	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)(0x1f)	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_FULL_CHIP_ERASE	FLASH_MODE_B_ERASE	サポートする全消去コマンド 左記は、256Mbit の場合の値です。
FLASH_ADDR_MODE	FLASH_MODE_4BYTE	Addressability Mode 左記は、256Mbit 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_IDDATA_SIZE	(uint16_t)20	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、 $\mu$ ITRON4.0 を想定しています。



表 5-15 r\_qspi\_flash\_n25q\_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 5-16 r\_qspi\_flash\_n25q\_sfr.h.rl78 の各種定義値

定数名	設定値	内容
FLASH_DR_CS0	P4.2	デバイス番号 0 ポート・レジスタ SFR 定義
FLASH_DDR_CS0	PM4.2	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 5-17 r\_qspi\_flash\_n25q\_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 5-18 r\_qspi\_flash\_n25q\_sub.h の各種定義値

定数名	設定値	内容
FLASH_BE_BUSY_WAIT	(uint32_t)(480000)	Bulk Erase Busy Timeout 480000 × 1ms = 480s
FLASH_SE_BUSY_WAIT	(uint32_t)(3000)	Sector Erase Busy Timeout 3000 × 1ms = 3s
FLASH_SSE_BUSY_WAIT	(uint32_t)(3000)	Subsector Erase Busy Timeout 3000 × 1ms = 3s
FLASH_DE_BUSY_WAIT	(uint32_t)(480000)	Dei Erase Busy Timeout 480000 × 1ms = 480s
FLASH_WBUSY_WAIT	(uint32_t)(8000)	Write Ready Timeout 8000 × 1us = 8ms
FLASH_T_WBUSY_WAIT	(uint16_t)MTL_T_1US	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_REG_SRWD	(uint8_t)(0x80)	Status Register Write Disable
FLASH_REG_BP3	(uint8_t)(0x40)	Block Protection Bit3
FLASH_REG_TB	(uint8_t)(0x20)	Top/Bottom Bit
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_FSTSREG_READY	(uint8_t)(0x80)	Program or erase controller
FLASH_FSTSREG_ERASE	(uint8_t)(0x20)	Erase (flag status register)
FLASH_FSTSREG_PROG	(uint8_t)(0x10)	Program (flag status register)
FLASH_FSTSREG_VPP	(uint8_t)(0x08)	V <sub>PP</sub> (flag status register)
FLASH_FSTSREG_PROT	(uint8_t)(0x02)	Protection (flag status register)

## 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\_n25q\_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\_n25q.h 参照)

表 5-19 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 5-20 に static 型変数を示します。

表 5-20 static 型変数 (r\_qspi\_flash\_n25q\_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 5-21 に関数を示します。

表 5-21 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_Flag_Status()	フラグステータス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_Clear_Status()	クリアフラグステータス処理
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
R_QSPI_FLASH_Set_Addressability_Mode()	アドレスモード設定処理

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

読み出し/書き込み用データ格納バッファアドレスは、下位層の MCU 個別のクロック同期式シングルマスタ制御ソフトウェアに依存し、4 バイト境界アドレスを指定する必要がある場合があります。MCU 個別のクロック同期式シングルマスタ制御ソフトウェアのアプリケーションノートを参照してください。

## 5.9 関数仕様

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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_drvif.h
宣言	error_t R_QSPI_FLASH_Init_Driver(void)
説明	<ul style="list-style-type: none"> <li>・ r_qspi_flash_init_port()関数をコールし、CS#端子の初期化を行います。</li> <li>・ 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。</li> </ul>
引数	なし
リターン値	<ul style="list-style-type: none"> <li>・ 初期化結果を返します。</li> </ul> FLASH_OK ; Successful operation FLASH_ERR_OTHER ; Other error r_qspi_flash_drvif_init_driver ()の戻り値を返します。

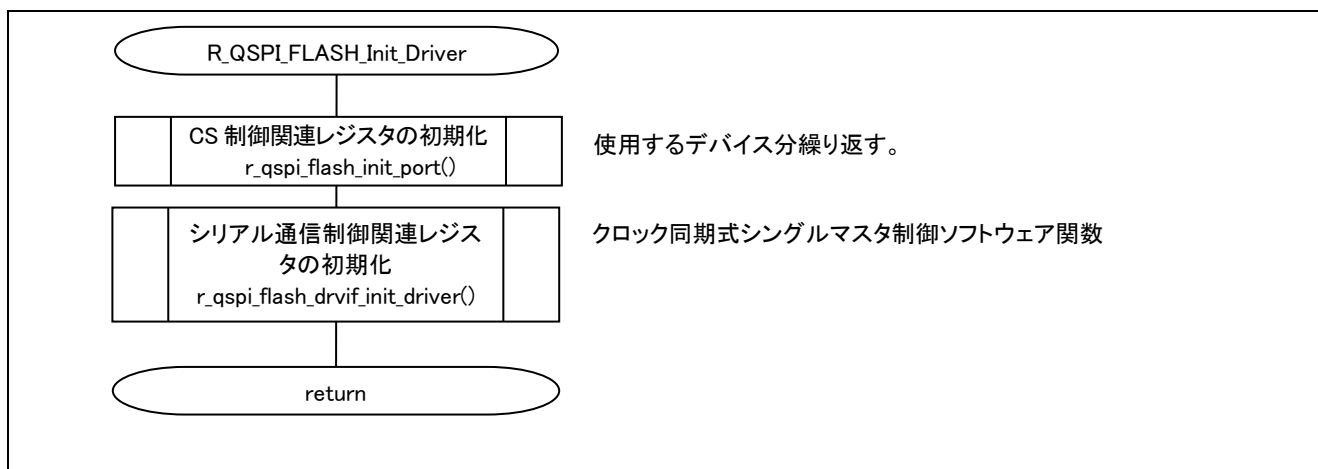


図 5-8 ドライバ初期化処理概要

## 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>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て、1 バイトを設定してください。</li> <li>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"> <li>Bit 7:SRWD <ul style="list-style-type: none"> <li>1: TB, BP3, BP2, BP1, BP0 are read-only bits</li> <li>0: TB, BP3, BP2, BP1, BP0 are read/writable</li> </ul> </li> <li>Bits 6 to 2: BP3, TB, BP2, BP1, BP0</li> <li>Bit 1:WEL <ul style="list-style-type: none"> <li>1: Internal Write Enable Latch is set</li> <li>0: Internal Write Enable Latch is reset</li> </ul> </li> <li>Bit 0:WIP <ul style="list-style-type: none"> <li>1: Program or Erase cycle is in progress</li> <li>0: No Program or Erase cycle is in progress</li> </ul> </li> </ul> </li> <li>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。BP ビットが割り当てられていない可能性があります。</li> </ul>
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"> <li>ステータスレジスタ取得結果を返します。</li> <li>FLASH_OK ; Successful operation</li> <li>FLASH_ERR_PARAM ; Parameter error</li> <li>FLASH_ERR_HARD ; Hardware error</li> <li>FLASH_ERR_OTHER ; Other error</li> </ul>

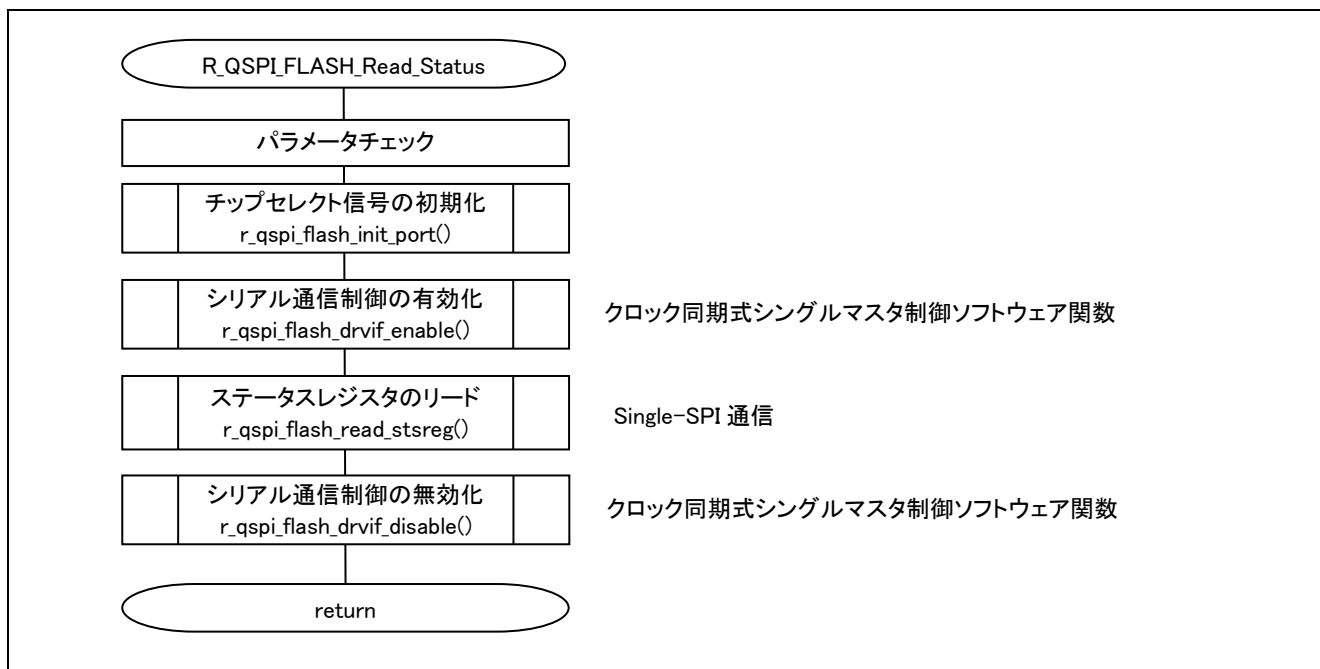


図 5-9 ステータスレジスタ読み出し処理概要

## 5.9.3 フラグステータスレジスタ読み出し処理

R_QSPI_FLASH_Read_Flag_Status	
概要	フラグステータスレジスタ読み出し処理
ヘッダ	r_qspi_flash_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_drvif.h
宣言	error_t R_QSPI_FLASH_Read_Flag_Status(uint8_t DevNo, uint8_t FAR* pFStatus)
説明	<ul style="list-style-type: none"> <li>・フラグステータスレジスタを読み出し、pFStatus に格納します。</li> <li>読み出しバッファとして、1 バイトを設定してください。</li> <li>・読み出しステータス格納バッファ(pFStatus)には下記情報が格納されます。</li> </ul> <p>Bit 7: Program or erase controller</p> <ul style="list-style-type: none"> <li>1: Ready</li> <li>0: Busy</li> </ul> <p>Bit 6: Erase suspend</p> <ul style="list-style-type: none"> <li>1: In effect</li> <li>0: Not in effect</li> </ul> <p>Bit 5: Erase</p> <ul style="list-style-type: none"> <li>1: Failure or protection error</li> <li>0: Clear</li> </ul> <p>Bit 4: Program</p> <ul style="list-style-type: none"> <li>1: Failure or protection error</li> <li>0: Clear</li> </ul> <p>Bit 3: V<sub>PP</sub></p> <ul style="list-style-type: none"> <li>1: Disable</li> <li>0: Enabled</li> </ul> <p>Bit 2: Program suspend</p> <ul style="list-style-type: none"> <li>1: In effect</li> <li>0: Not in effect</li> </ul> <p>Bit 1: Protection</p> <ul style="list-style-type: none"> <li>1: Failure or protection error</li> <li>0: Clear</li> </ul> <p>Bit 0: Reserved</p>
引数	uint8_t DevNo ; デバイス番号 uint8_t FAR* pFStatus ; 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"> <li>・フラグステータスレジスタ取得結果を返します。</li> </ul> <p>FLASH_OK ; Successful operation</p> <p>FLASH_ERR_PARAM ; Parameter error</p> <p>FLASH_ERR_HARD ; Hardware error</p> <p>FLASH_ERR_OTHER ; Other error</p>



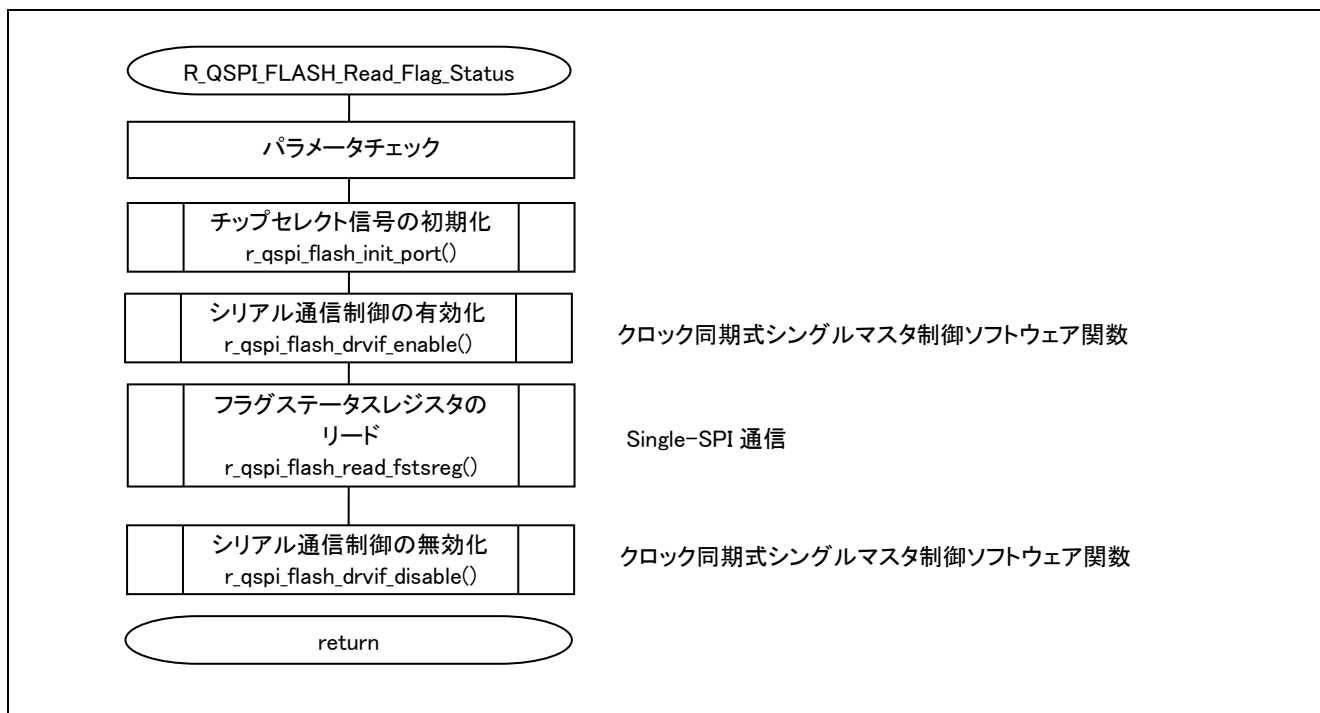


図 5-10 フラグステータスレジスタ読み出し処理概要

## 5.9.4 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>・ライトプロテクトの設定を行います。</li> <li>・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。</li> </ul>

WpSts	BP3	TB1	BP2	BP1	BP0	Protected Area
0x00	0	0	0	0	0	None
0x01	0	0	0	0	1	Sector 2047
0x02	0	0	0	1	0	Sectors (2046 to 2047)
0x03	0	0	0	1	1	Sectors (2044 to 2047)
0x04	0	0	1	0	0	Sectors (2040 to 2047)
0x05	0	0	1	0	1	Sectors (2032 to 2047)
0x06	0	0	1	1	0	Sectors (2016 to 2047)
0x07	0	0	1	1	1	Sectors (1984 to 2047)
0x10	1	0	0	0	0	Sectors (1920 to 2047)
0x11	1	0	0	0	1	Sectors (1792 to 2047)
0x12	1	0	0	1	0	Sectors (1536 to 2047)
0x13	1	0	0	1	1	Sectors (1024 to 2047)
0x14 – 0x17	1	0	1	0/1	0/1	All sectors
0x08	0	1	0	0	0	None
0x09	0	1	0	0	1	Sector 0
0x0a	0	1	0	1	0	Sectors (0 to 1)
0x0b	0	1	0	1	1	Sectors (0 to 3)
0x0c	0	1	1	0	0	Sectors (0 to 7)
0x0d	0	1	1	0	1	Sectors (0 to 15)
0x0e	0	1	1	1	0	Sectors (0 to 31)
0x0f	0	1	1	1	1	Sectors (0 to 63)
0x18	1	1	0	0	0	Sectors (0 to 127)
0x19	1	1	0	0	1	Sectors (0 to 255)
0x1a	1	1	0	1	0	Sectors (0 to 511)
0x1b	1	1	0	1	1	Sectors (0 to 1023)
0x1c – 0x1f	1	1	1	0/1	0/1	All sectors

- ・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 Serial NOR Flash memory のデータシート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\_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、

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。

- ・プロテクトが正常に行われたかは、ステータスレジスタを読み出して確認してください。また、WEL ビットがセットされている場合、WRDI コマンド発行処理 (R\_QSPI\_FLASH\_Write\_Di()) を行ってください。

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

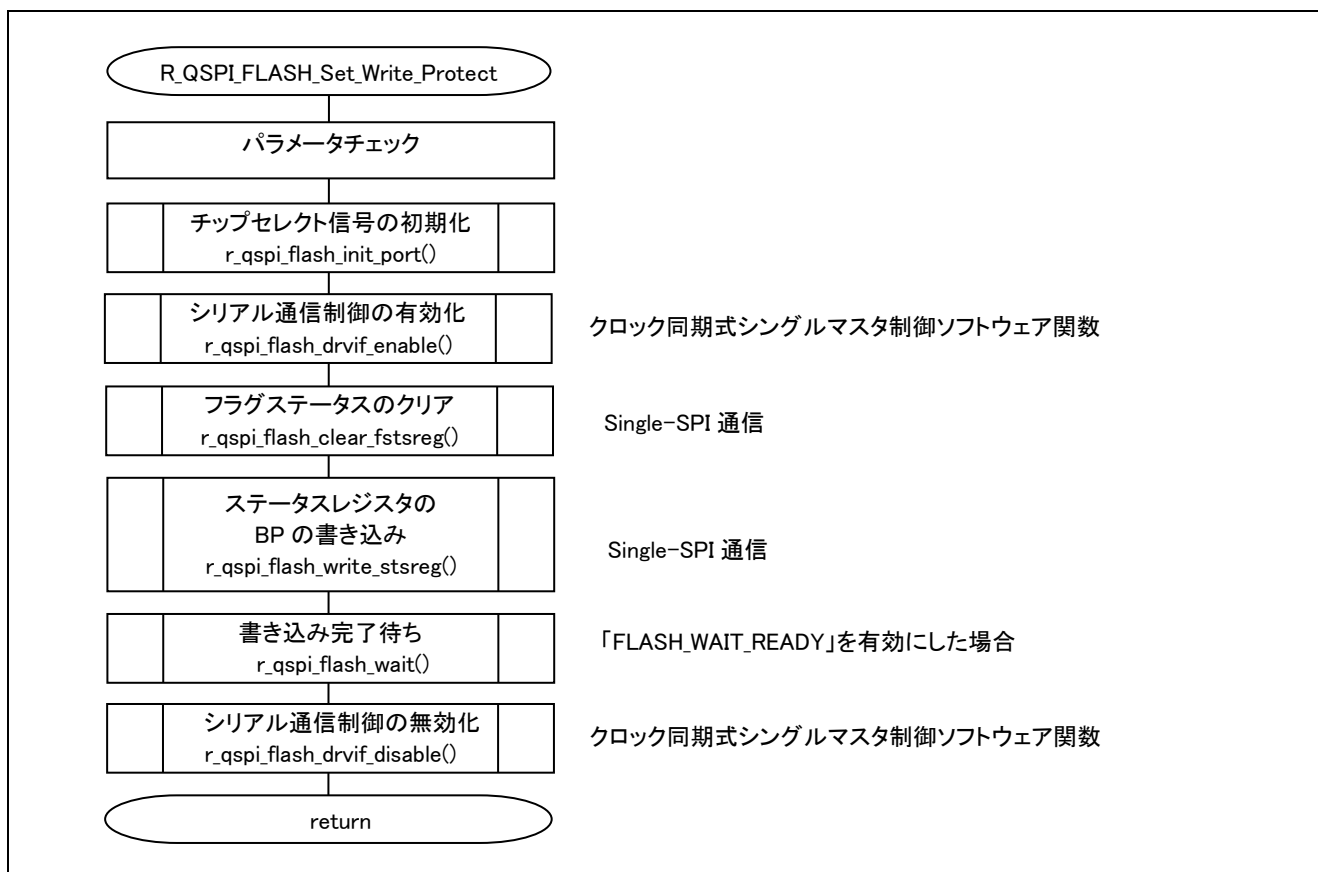


図 5-11 ライトプロテクト設定処理概要

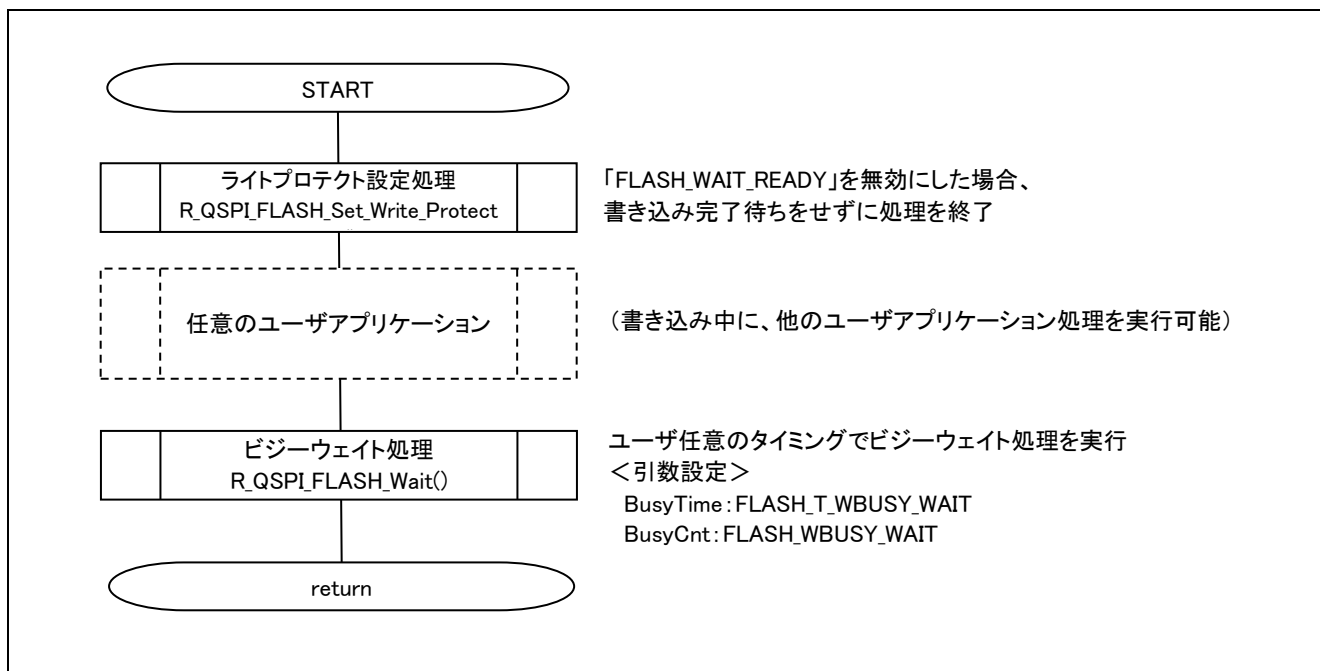


図 5-12 R\_QSPI\_FLASH\_Wait()を使用したライトプロテクト設定完了待ち方法

## 5.9.5 クリアフラグステータス処理

## R\_QSPI\_FLASH\_Clear\_Status

概要	クリアフラグステータス処理	
ヘッダ	r_qspi_flash_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_drvif.h	
宣言	error_t R_QSPI_FLASH_Clear_Status(uint8_t DevNo)	
説明	<ul style="list-style-type: none"> <li>・フラグステータスレジスタのエラービットをクリアします。</li> <li>・プログラムエラー、消去エラー、ライトプロテクトエラー等が発生した場合は、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。</li> </ul>	
引数	uint8_t	DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> <li>・クリア結果を返します。</li> </ul>	
	FLASH_OK	; Successful operation
	FLASH_ERR_PARAM	; Parameter error
	FLASH_ERR_HARD	; Hardware error
	FLASH_ERR_OTHER	; Other error

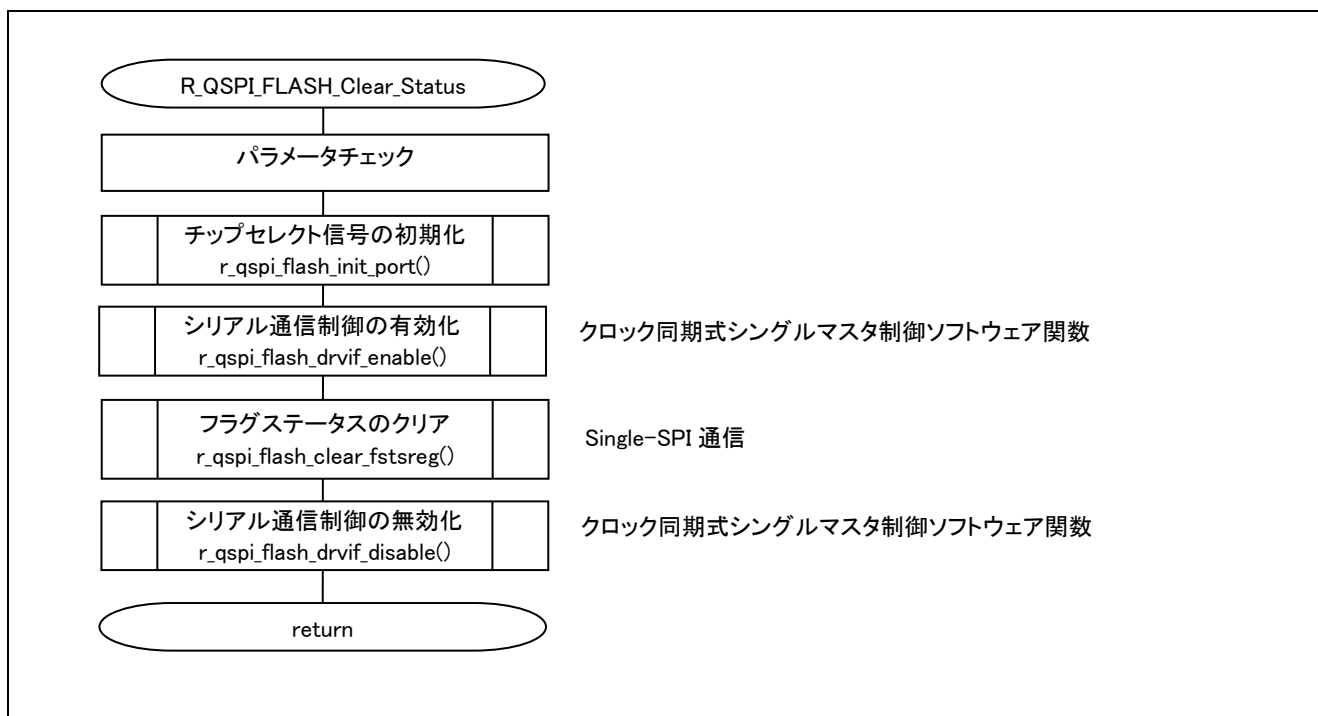


図 5-13 クリアフラグステータス処理概要

## 5.9.6 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_drvif.h	
宣言	error_t R_QSPI_FLASH_Write_Di(uint8_t DevNo)	
説明	<ul style="list-style-type: none"> <li>・ステータスレジスタの WEL ビットをクリアします。</li> <li>・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 WEL ビットをクリアする必要があります。</li> </ul>	
引数	uint8_t	DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> <li>・クリア結果を返します。</li> </ul>	
	FLASH_OK	; Successful operation
	FLASH_ERR_PARAM	; Parameter error
	FLASH_ERR_HARD	; Hardware error
	FLASH_ERR_OTHER	; Other error

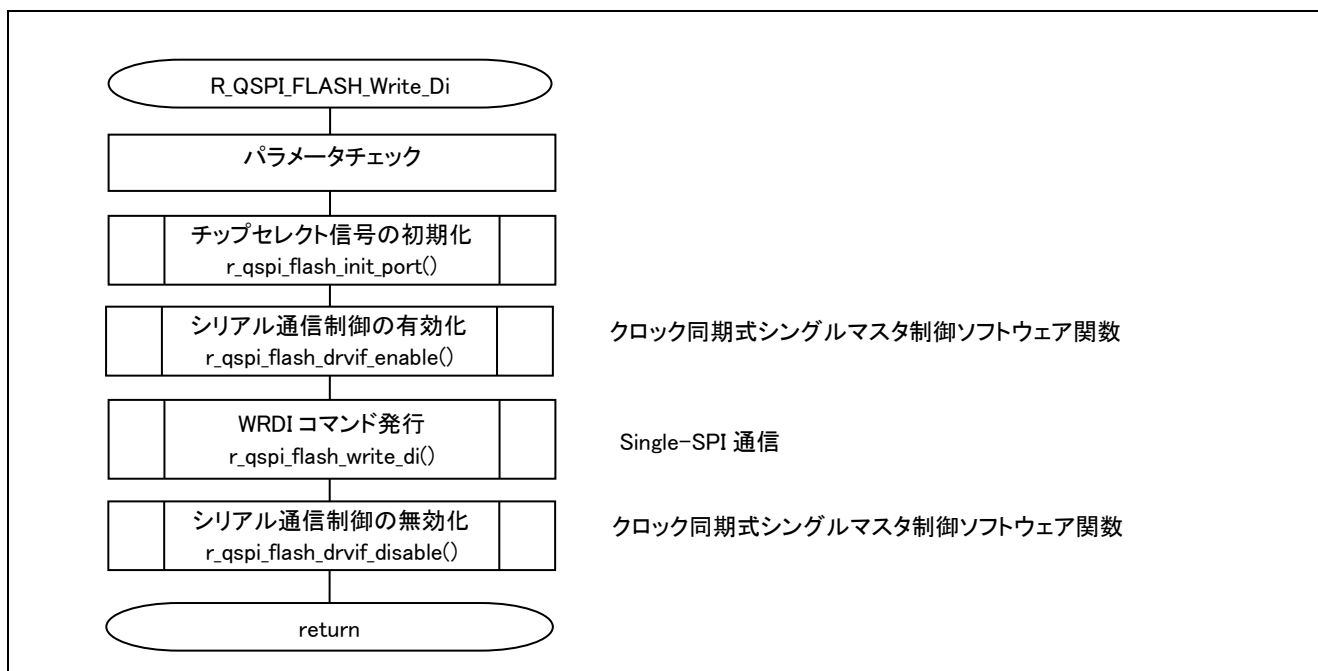


図 5-14 WRDI コマンド発行処理概要

## 5.9.7 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>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。</li> <li>最終読み出しアドレスは、Serial NOR Flash memory 容量-1 です。</li> <li>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。</li> </ul>																								
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"> <li>読み出し結果を返します。</li> </ul> <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																							

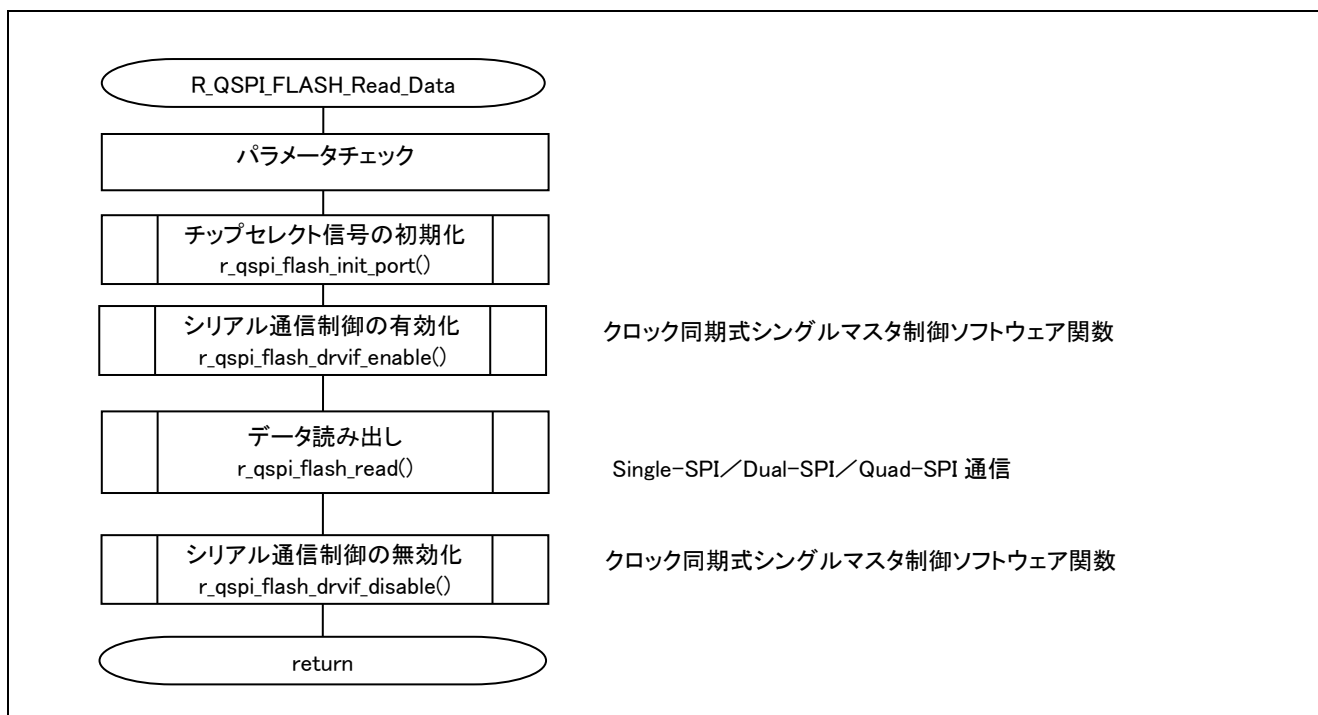


図 5-15 データ読み出し処理概要

## 5.9.8 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>・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。</li> <li>・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。書き込み完了待ちを実行すると「FLASH_ERR_WP」を返します。</li> <li>・最終書き込みアドレスは、Serial NOR Flash memory 容量-1 です。</li> <li>・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。</li> <li>・本ユーザ API では、r_qspi_flash_n25q.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。</li> <li>・リターン値が「FLASH_ERR_WP」または「FLASH_ERR_OTHER」の場合、フラグステータスレジスタのエラービットがセットされているため、クリアフラグステータス処理を行ってください。また、ステータスレジスタの WEL ビットをクリアするため、WRDI コマンド発行処理を行ってください。</li> </ul>																								
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"> <li>・読み出し結果を返します。</li> </ul> <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> <tr> <td>FLASH_ERR_WP</td> <td>;</td> <td>Write-protection 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> </tbody> </table>	FLASH_OK	;	Successful operation	FLASH_ERR_PARAM	;	Parameter error	FLASH_ERR_HARD	;	Hardware error	FLASH_ERR_WP	;	Write-protection 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_WP	;	Write-protection error																							
FLASH_ERR_TIMEOUT	;	Time out error																							
FLASH_ERR_OTHER	;	Other error																							



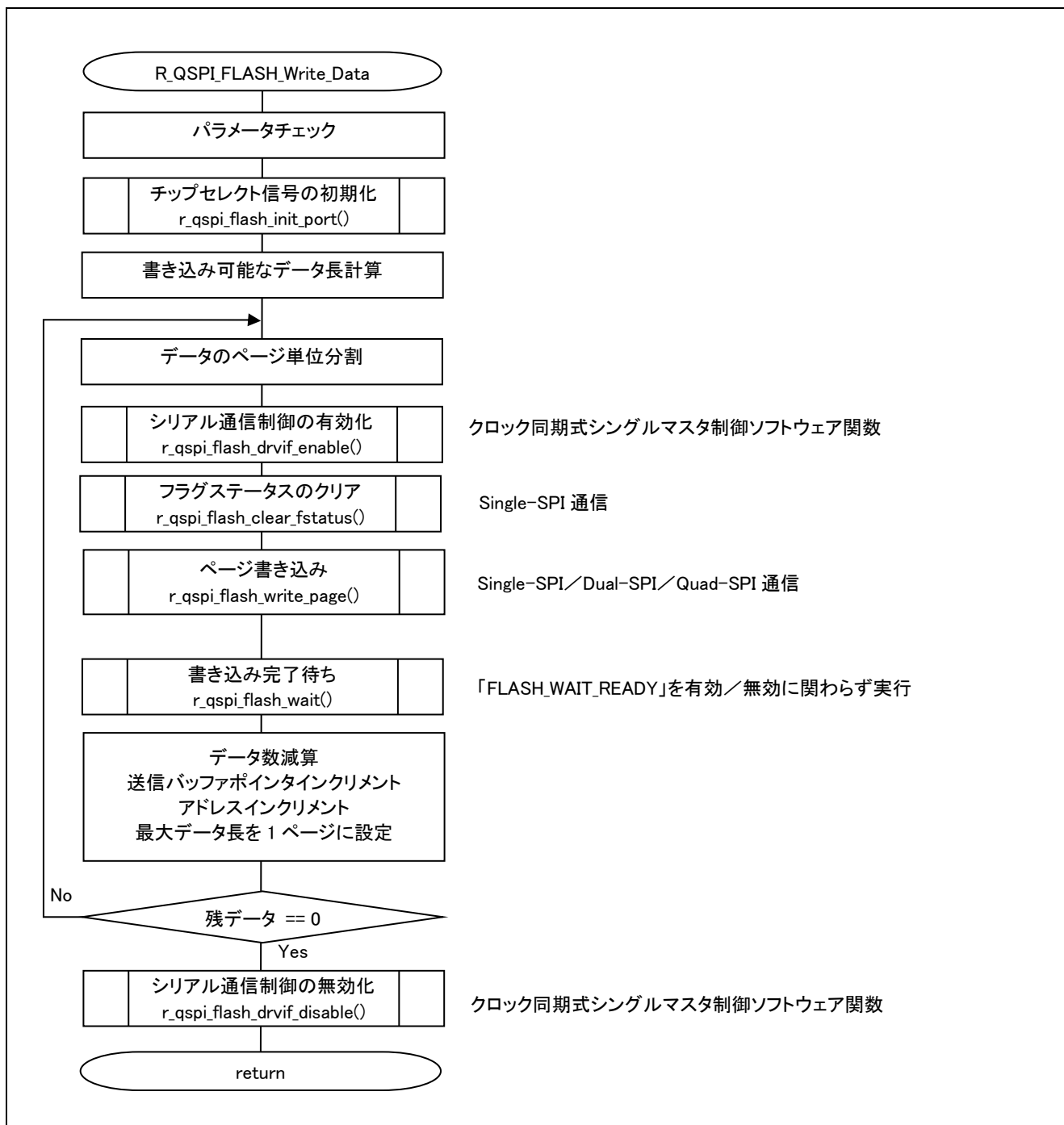


図 5-16 データ書き込み処理概要

## 5.9.9 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>・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。</li> <li>・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。</li> <li>・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。書き込み完了待ちを実行すると「FLASH_ERR_WP」を返します。</li> <li>・最終書き込みアドレスは、Serial NOR Flash memory 容量-1 です。</li> <li>・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。</li> <li>・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。</li> <li>・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。</li> <li>・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。</li> <li>・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、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 5-18 を参照してください。</li> <li>・リターン値が「FLASH_ERR_WP」または「FLASH_ERR_OTHER」の場合、フラグステータスレジスタのエラービットがセットされているため、クリアフラグステータス処理を行ってください。また、ステータスレジスタの WEL ビットをクリアするため、WRDI コマンド発行処理を行ってください。</li> </ul>																								
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"> <li>・読み出し結果を返します。</li> </ul> <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> <tr> <td>FLASH_ERR_WP</td> <td>;</td> <td>Write-protection error（「FLASH_WAIT_READY」有効時）</td> </tr> <tr> <td>FLASH_ERR_TIMEOUT</td> <td>;</td> <td>Time out error（「FLASH_WAIT_READY」有効時）</td> </tr> <tr> <td>FLASH_ERR_OTHER</td> <td>;</td> <td>Other error</td> </tr> </tbody> </table>	FLASH_OK	;	Successful operation	FLASH_ERR_PARAM	;	Parameter error	FLASH_ERR_HARD	;	Hardware error	FLASH_ERR_WP	;	Write-protection error（「FLASH_WAIT_READY」有効時）	FLASH_ERR_TIMEOUT	;	Time out 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_WP	;	Write-protection error（「FLASH_WAIT_READY」有効時）																							
FLASH_ERR_TIMEOUT	;	Time out error（「FLASH_WAIT_READY」有効時）																							
FLASH_ERR_OTHER	;	Other error																							

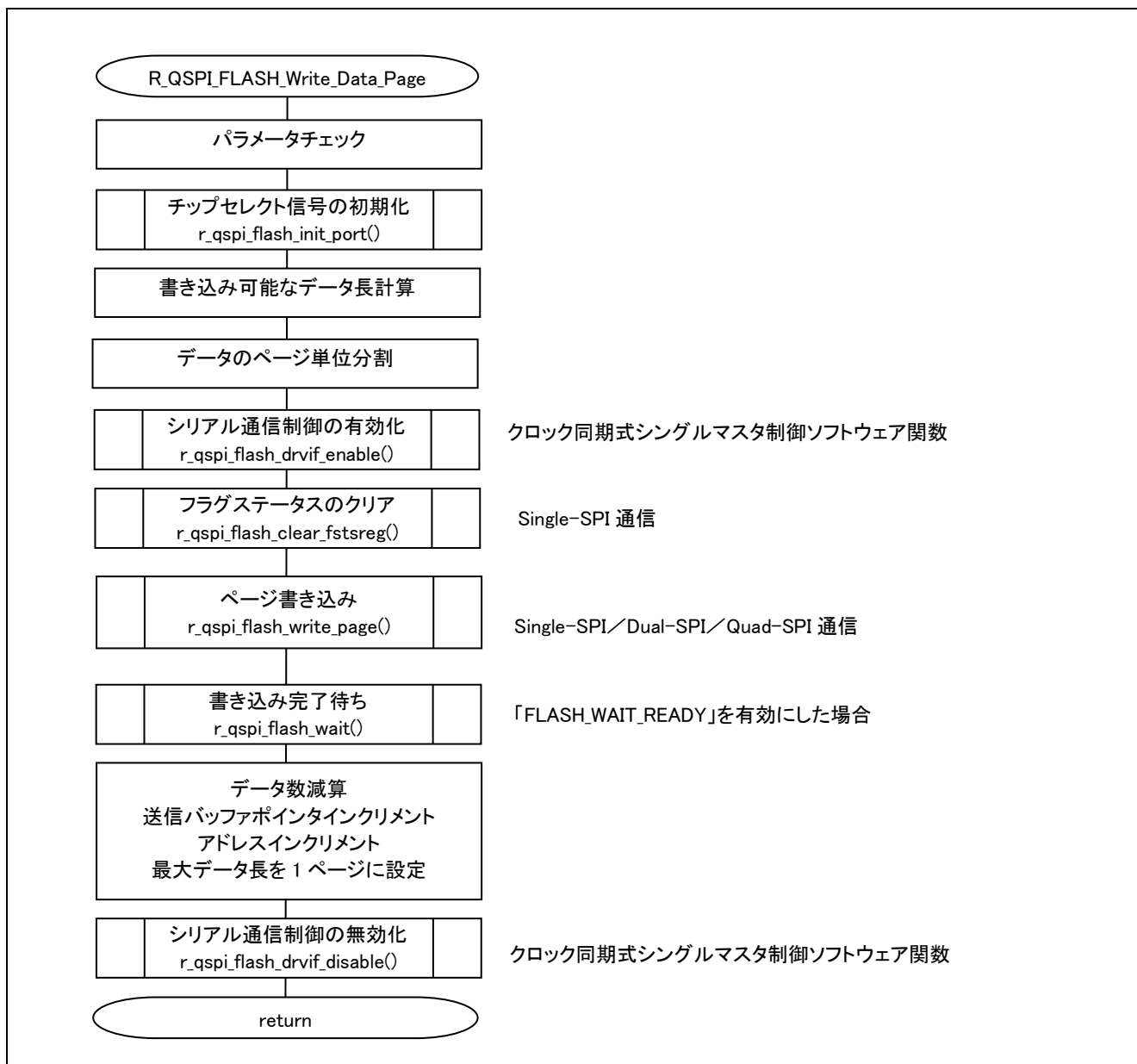


図 5-17 データ書き込み処理（1Page 書き込み用）概要

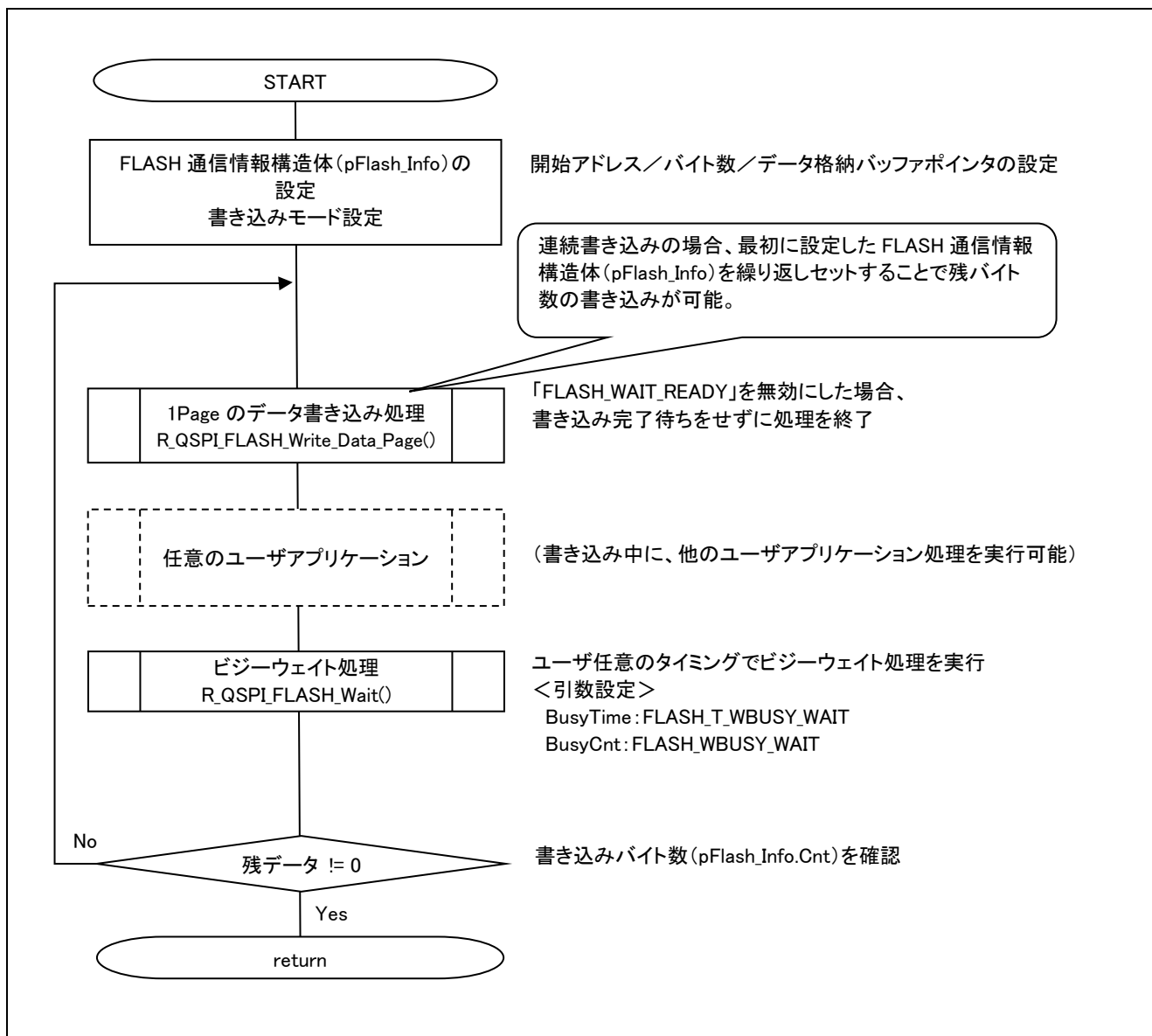


図 5-18 R\_QSPI\_Flash\_Wait()を使用したデータ書き込み処理 (1Page 書き込み用) の完了待ち方法

## 5.9.10 消去処理

## R\_QSPI\_FLASH\_Erase

概要	消去処理	
ヘッダ	r_qspi_flash_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_drvif.h	
宣言 説明	<pre>error_t R_QSPI_FLASH_Erase(uint8_t DevNo, uint32_t Addr, uint8_t Mode)</pre> <ul style="list-style-type: none"> <li>・ 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)、指定されたダイの全データ消去 (Die Erase) を行います。</li> <li>・ Bulk Erase の場合、Addr には 0x00000000 を設定してください。</li> <li>・ Die Erase の場合、Addr には消去したい各ダイの先頭アドレスを設定してください。</li> <li>・ 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。消去完了待ちを実行すると「FLASH_ERR_WP」を返します。</li> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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、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 5-20 を参照してください。</li> <li>・ 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"> <li>Bulk Erase 時 ; BusyCnt = FLASH_BE_BUSY_WAIT</li> <li>Sector Erase 時 ; BusyCnt = FLASH_SE_BUSY_WAIT</li> <li>Subsector Erase 時 ; BusyCnt = FLASH_SSE_BUSY_WAIT</li> <li>Die Erase 時 ; BusyCnt = FLASH_DE_BUSY_WAIT</li> </ul> </li> <li>・ リターン値が「FLASH_ERR_WP」または「FLASH_ERR_OTHER」の場合、フラグステータスレジスタのエラービットがセットされているため、クリアフラグステータス処理を行ってください。また、ステータスレジスタの WEL ビットをクリアするため、WRDI コマンド発行処理を行ってください。</li> </ul>	
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
		; FLASH_MODE_D_ERASE
リターン値	<ul style="list-style-type: none"> <li>・ 消去結果を返します。</li> <li>FLASH_OK ; Successful operation</li> <li>FLASH_ERR_PARAM ; Parameter error</li> <li>FLASH_ERR_HARD ; Hardware error</li> <li>FLASH_ERR_WP ; Write-protection error (「FLASH_WAIT_READY」有効時)</li> <li>FLASH_ERR_TIMEOUT; Time out error (「FLASH_WAIT_READY」有効時)</li> <li>FLASH_ERR_OTHER ; Other error</li> </ul>	

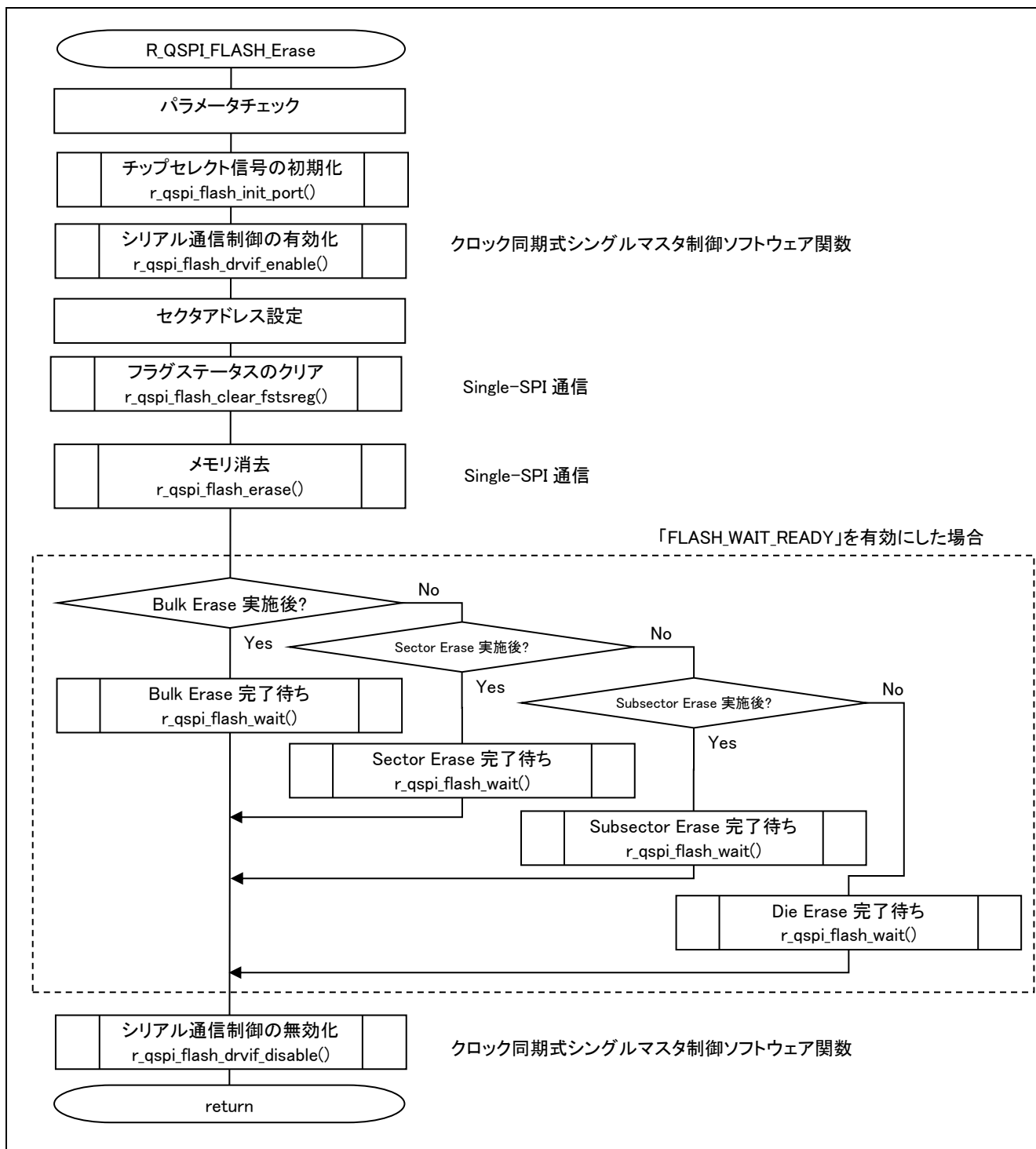


図 5-19 消去処理概要

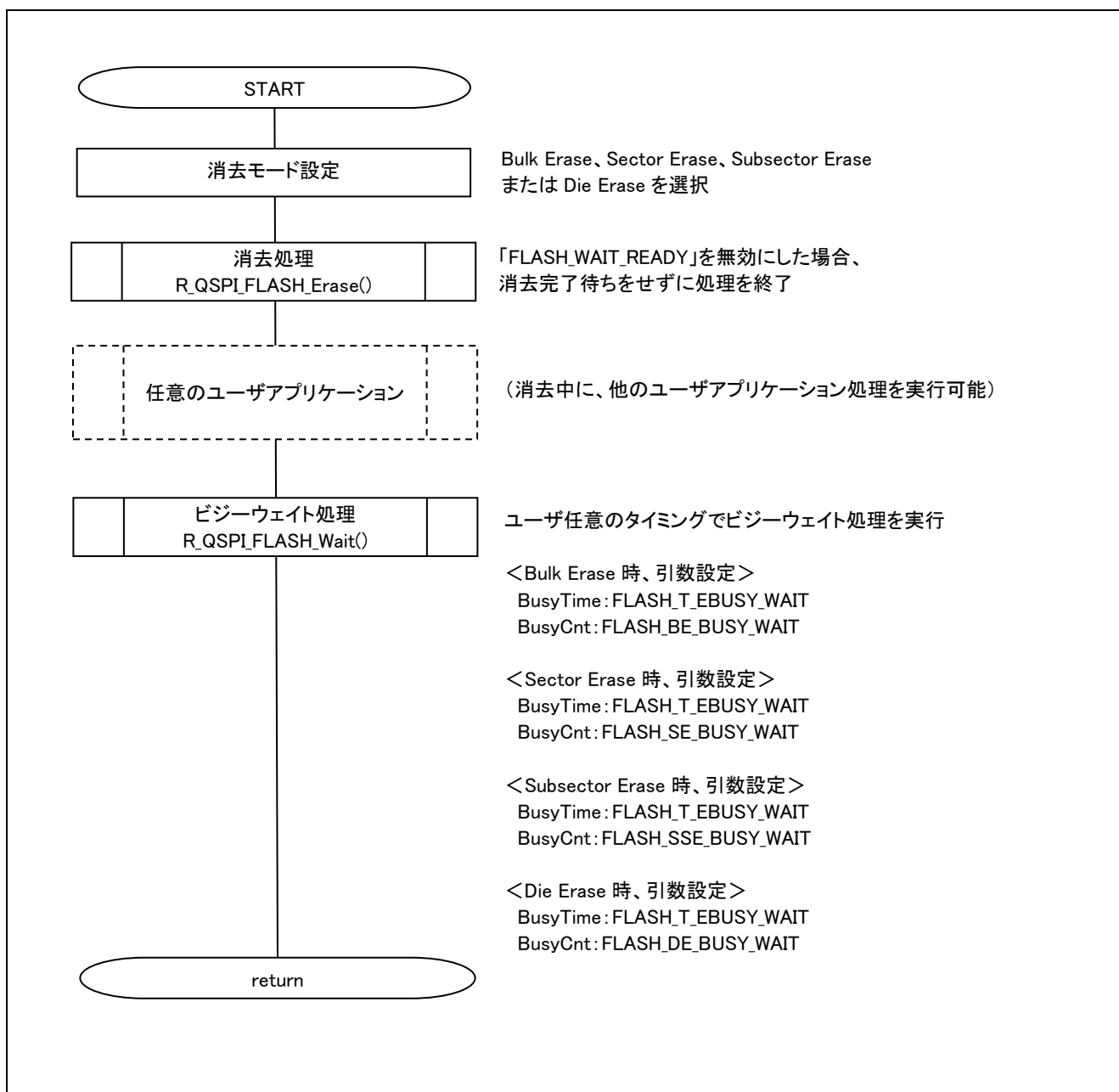


図 5-20 R\_QSPI\_FLASH\_Wait()を使用した消去処理の完了待ち方法

## 5.9.11 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>・ 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、20 バイトを設定してください。</li> <li>・ 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。 <ol style="list-style-type: none"> <li>(1) Manufacturer ID</li> <li>(2) Memory Type</li> <li>(3) Memory Capacity</li> <li>(4) Length of data to follow</li> <li>(5) Extended device ID and device configuration information</li> <li>(6 - 20) Customized factory data</li> </ol> </li> </ul>
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"> <li>・ 読み出し結果を返します。</li> </ul> FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error

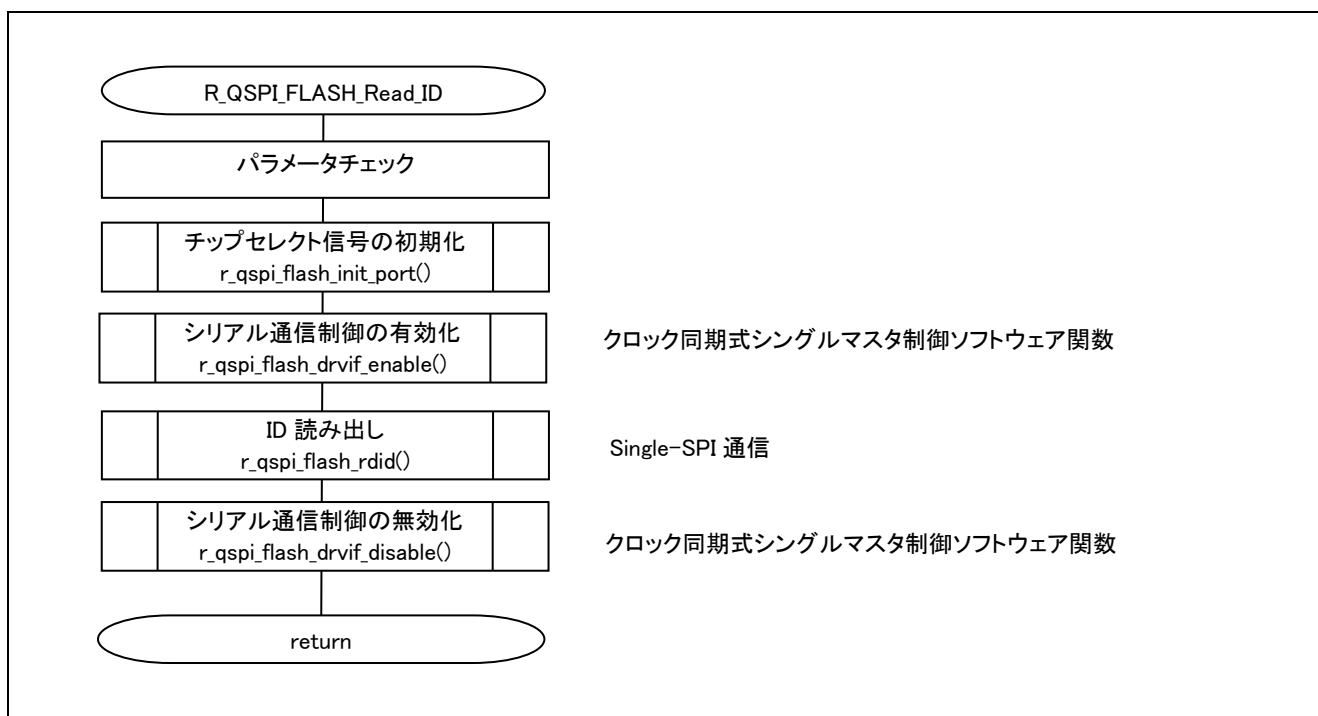


図 5-21 ID 読み出し処理概要



## 5.9.12 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_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_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"> <li>・「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。</li> <li>・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。</li> <li>・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。</li> <li>・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"> <li>FLASH_MODE_REG_WRITE : レジスタ書き込みモード</li> <li>FLASH_MODE_PROG_ERASE : データ書き込み及び消去モード</li> </ul> </li> <li>・レジスタ書き込みモード（256Mbit 以下の小容量品）の場合、RDSR コマンドを発行し、WIP ビットによりレディ／ビジーを判定します。</li> <li>・レジスタ書き込みモード（512Mbit 品, 1Gbit 品）の場合、RDFSR コマンドを発行し、Program or erase controller ビットによりレディ／ビジーを判定します。Program or erase controller ビットが 512Mbit 品では 2 回連続、1Gbit 品では 4 回連続で 1（Ready）になるとレディと判定します。</li> <li>・データ書き込み及び消去モードの場合、最初に WIP ビットによりレディ／ビジーを判定します。WIP ビットがレディになると、RDFSR コマンドを発行し、Program or erase controller ビットによるレディ／ビジー判定とエラービットを確認します。</li> <li>・Protection エラービットが 1 の場合、「FLASH_ERR_WP」を返します。</li> <li>・Protection エラービットが 0 かつ、Erase／Program／V<sub>PP</sub> エラービットのいずれかが 1 の場合、「FLASH_ERR_OTHER」を返します。</li> <li>・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、Time out 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。</li> </ul>

状態	BusyTime	BusyCnt
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_WBUSY_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
消去中（Die Erase）	FLASH_T_EBUSY_WAIT	FLASH_DE_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 : 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 : 書き込み時
;		FLASH_BE_BUSY_WAIT : 消去時（Bulk Erase）	
	;	FLASH_SE_BUSY_WAIT : 消去時（Sector Erase）	

```

uint8_t Mode
; FLASH_SSE_BUSY_WAIT : 消去時 (Subsector Erase)
; FLASH_DE_BUSY_WAIT : 消去時 (Die 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_WP      ; Write-protection error
FLASH_ERR_TIMEOUT ; Time out error (BusyCnt≠0 の場合)
FLASH_ERR_OTHER   ; Other error
    
```

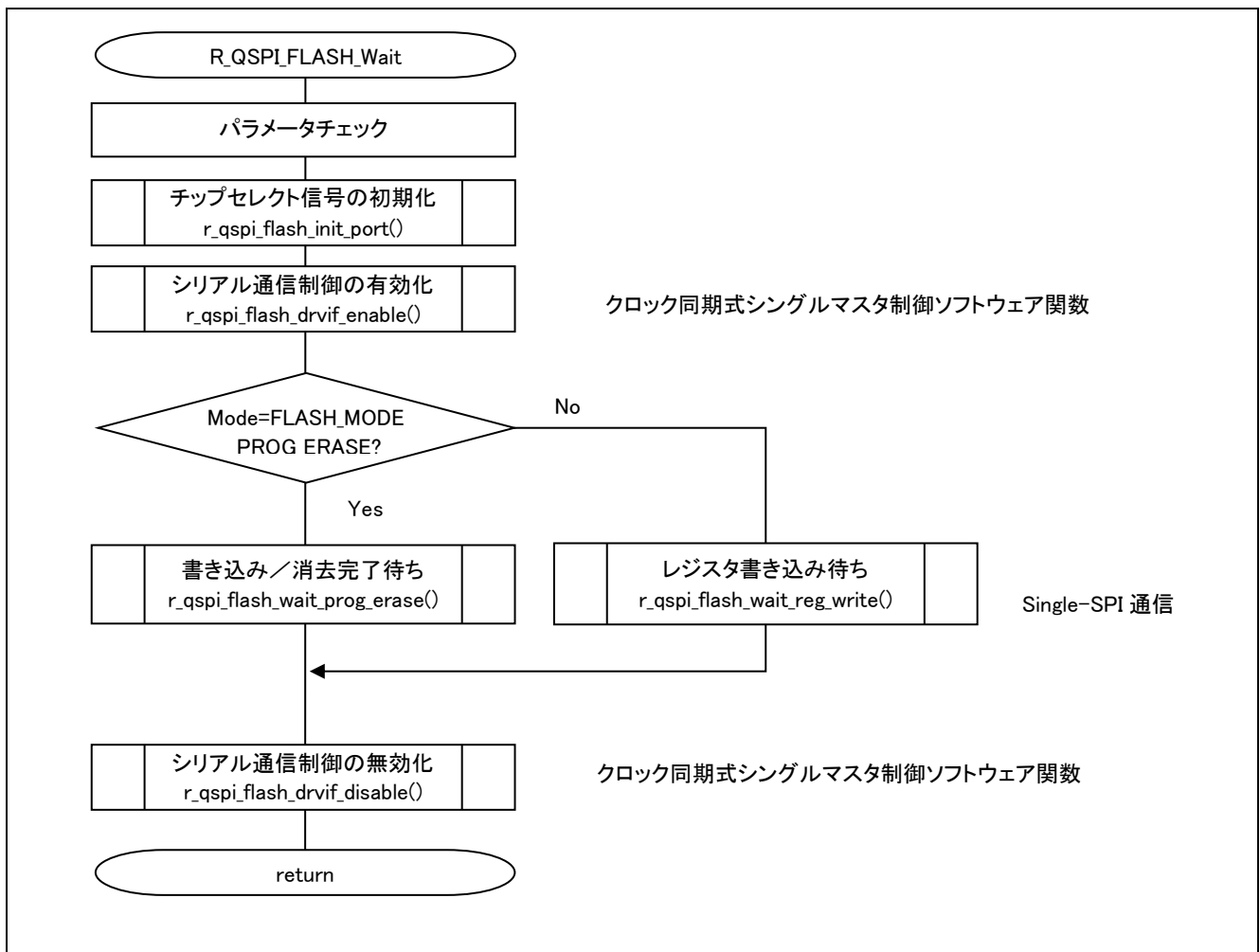


図 5-22 ビジーウェイト処理概要

5.9.13 アドレスモード設定処理

R\_QSPI\_FLASH\_Set\_Addressability\_Mode

概要	アドレスモード設定処理
ヘッダ	r_qspi_flash_n25q.h, r_qspi_flash_n25q_sub.h, r_qspi_flash_n25q_sfr.h, r_qspi_flash_n25q_drvif.h
宣言	error_t R_QSPI_FLASH_Set_Addressability_Mode (uint8_t DevNo)
説明	<ul style="list-style-type: none"> <li>・ r_qspi_flash_enter_4addr()関数をコールし、アドレッシングを 4Byte アドレスモードに設定します。</li> <li>・ 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。</li> </ul>
引数	uint8_t DevNo ; デバイス番号
リターン値	<ul style="list-style-type: none"> <li>・ アドレスモード設定結果を返します。</li> <li>FLASH_OK ; Successful operation</li> <li>FLASH_ERR_PARAM ; Parameter error</li> <li>FLASH_ERR_HARD ; Hardware error</li> <li>FLASH_ERR_OTHER ; Other error</li> </ul>

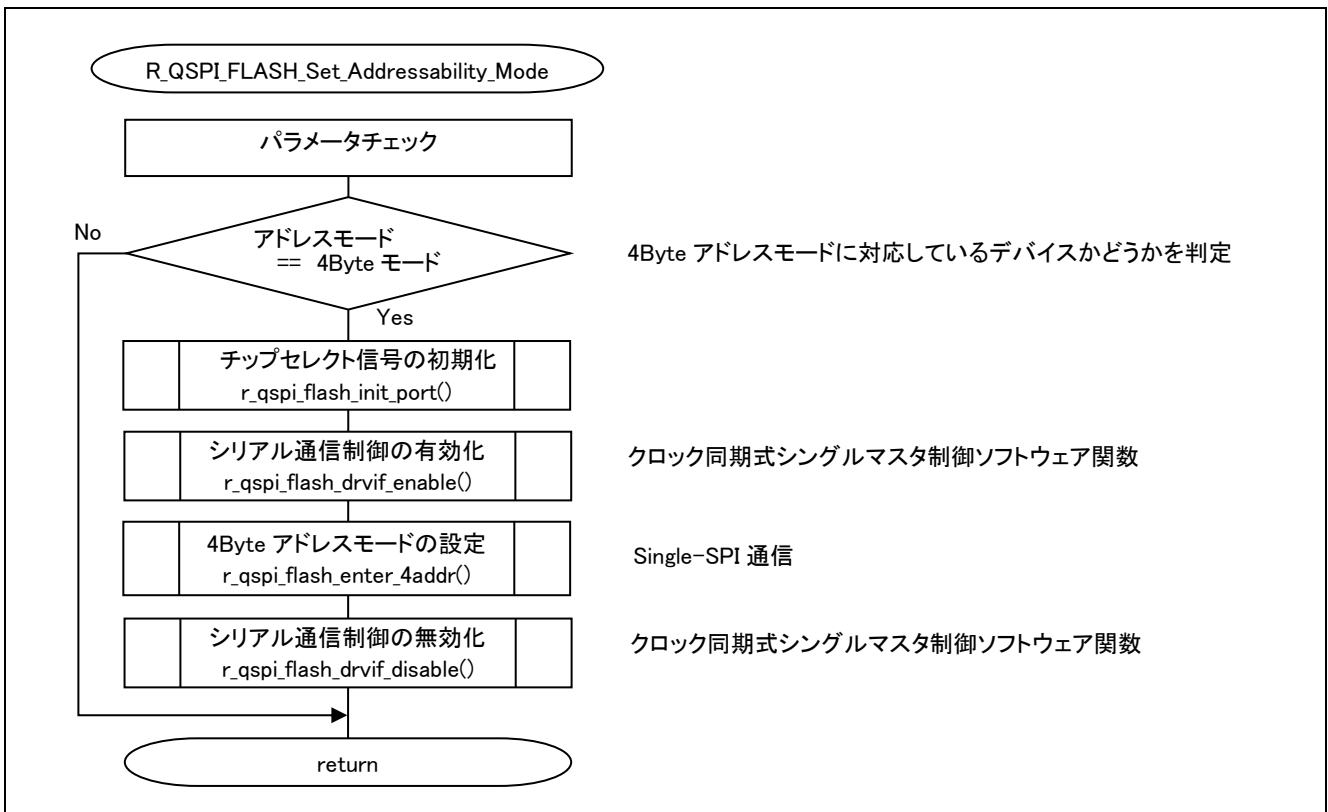


図 5-23 アドレスモード設定処理概要

## 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\_n25q.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 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、256Mbit デバイスを使用する場合の例です。

```

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

// #define N25Q32M                /* 32Mbit ( 4MByte) */
// #define N25Q64M                /* 64Mbit ( 8MByte) */
// #define N25Q128M               /* 128Mbit (16MByte) */
#define N25Q256M                 /* 256Mbit (32MByte) */
// #define N25Q512M               /* 512Mbit (64MByte) */
// #define N25Q1G                 /* 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 (Uint:ms) */
```

※: 本サンプルコードの OS 制御は、 $\mu$ 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:

ライトプロテクト設定処理 (R\_QSPI\_FLASH\_Set\_Write\_Protect())

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\_n25q\_sfr.h

r\_qspi\_flash\_n25q\_sfr.h.XXX は、MCU 毎に作成したものです。どれか一つを r\_qspi\_flash\_n25q\_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\_n25q\_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 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\_n25q\_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)、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)、指定されたダイの全データ消去 (Die 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 と Die Erase のタイムアウト時間を 480s、Sector Erase と Subsector Erase のタイムアウト時間を 3s 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 : 480s */
/* Sector Erase : 3s */
/* Subsector Erase : 3s */
/* Die Erase : 480s */
/* Page (256 bytes) Program: 8ms */
#define FLASH_BE_BUSY_WAIT (uint32_t) (480000)
/* Bulk Erase busy timeout 480,000*1ms = 480s */
#define FLASH_SE_BUSY_WAIT (uint32_t) (3000)
/* Sector Erase busy timeout 3,000*1ms = 3s */
#define FLASH_SSE_BUSY_WAIT (uint32_t) (3000)
/* Subsector Erase busy timeout 3,000*1ms = 3s */
#define FLASH_DE_BUSY_WAIT (uint32_t) (480000)
/* Die Erase busy timeout 480,000*1ms = 480s */

```

## (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。

以下は、書き込みタイムアウト時間を 8ms に設定する場合での記述例を示しています。

```

#define FLASH_WBUSY_WAIT (uint32_t) (8000)
/* Write ready timeout 8,000*1us = 8ms */

```

## 6.1.4 r\_qspi\_flash\_n25q\_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( <b>FLASH_CMD_FREAD</b> ,(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( <b>FLASH_CMD_DOFRR</b> ,(uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE+1)	DUAL OUTPUT FAST READ	Dual-SPI 読み出し (高速)
3	r_qspi_flash_send_cmd( <b>FLASH_CMD_QOFR</b> ,(uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE+1)	QUAD OUTPUT FAST 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( <b>FLASH_CMD_PP</b> , (uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE)	PAGE PROGRAM	Single-SPI 書き込み
2	r_qspi_flash_send_cmd( <b>FLASH_CMD_DIPP</b> , (uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE)	DUAL INPUT FAST PROGRAM	Dual-SPI 書き込み
3	r_qspi_flash_send_cmd( <b>FLASH_CMD_QIPP</b> , (uint32_t)Addr,FLASH_CMD_SIZE+FLASH_ADDR_SIZE)	QUAD INPUT FAST PROGRAM	Quad-SPI 書き込み

### 6.1.5 r\_qspi\_flash\_n25q\_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\_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。

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\_n25q\_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 6-4 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)

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_n25q.h  
r_qspi_flash_n25q_sub.h  
r_qspi_flash_n25q_sfr.h  
r_qspi_flash_n25q_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_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。

なお、Serial NOR Flash memory は、電源を投入し S#端子が  $V_{ccmin}$  以上の規定電圧に達した後、Status Register の write in progress bit (WIP) が 1 になります。その後、VTW ( $V_{CC,min}$  to device fully accessible) の時間が経過すると、WIP がリセットされて 0 になります。WIP が 1 にセットされてから 0 になるまでの間、READ STATUS REGISTER と READ FLAG STATUS REGISTER を除く、全てのコマンドを受け付けません。WRITE/PROGRAM/ERASE コマンドの発行は、WIP が 0 になってから行ってください。

## 7.6 N25Q256A83ESF40x と N25Q256A83E1240x の未対応コマンド (ENTER or EXIT QUAD Command) について

N25Q256A83ESF40x と N25Q256A83E1240x にて 4Byte アドレスモードに変更するためには、ENTER or EXIT QUAD Command を発行する必要があります。ただし、本サンプルコードでは本コマンドに対応しておりません。コマンドの発行は、ユーザ独自で作成し、処理するようにしてください。

RX ファミリ、RL78 ファミリ、78K0R/Kx3-L

Micron Technology 社製 N25Q 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
		52	6.1.2 r_qspi_pcm_p5q_sfr.h 内容を修正した。
		60	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 の章番号をを追加し、ファミリ別に分けた。
		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
		7	2.2 RL78 ファミリ、78K0R/Kx3-L (1) RL78/G14 SAU 統合開発環境 CubeSuite+の場合 「エンディアン リトルエンディアン」を削除した。
		7	2.2 RL78 ファミリ、78K0R/Kx3-L (2) RL78/G1C SAU 統合開発環境 IAR Embedded Workbench の場合 「エンディアン リトルエンディアン」を削除した。
		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 マイコングループ追加によりタイトルを更新した。 RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ) 元は、RX210, RX21A, RX220, RX63N, RX63T, グループ RSPI を使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1196JJ)であった。 RX210, RX21A, RX220, RX63N, RX63T, RX111 グループ SCI を使ったクロック同期式シングルマスタ制御ソフトウェア

			<p>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。</p> <p>RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ) 元は、RL78/G14 シリアル・アレイ・ユニットの CSI モードを使ったクロック同期式シングルマスタ制御ソフトウェア(R01AN1195JJ)であった。</p>
		22,24	<p>5.3 必要メモリサイズ</p> <p>5.3.1 RX ファミリと</p> <p>5.3.2 RL78 ファミリ、78K0R/Kx3-L を追加した。</p>
		22	<p>5.3.1 RX ファミリ に、以下を追加した。</p> <p>環境は、「2. 動作確認条件」を参照してください。</p>
		25	<p>5.3.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加</p> <p>(3) RL78/L13 SAU 統合開発環境 CubeSuite+の場合</p> <p>(4) RL78/L13 SAU 統合開発環境 IAR Embedded Workbench の場合</p>
		26	<p>5.4 ファイル構成</p> <p>アプリケーションノート番号を変更</p> <p>ソースのフォルダ名を変更</p> <p>新規デバイスのレジスタ用共通定義を追加</p>
1.06	2016.03.31	7,9	<p>2.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加</p> <p>(2) RL78/G14 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)</p> <p>(5) RL78/G1C SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)</p>
		26,28	<p>5.3.2 RL78 ファミリ、78K0R/Kx3-L に、以下を追加</p> <p>(2) RL78/G14 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)</p> <p>(5) RL78/L13 SAU 統合開発環境 CS+ for CC の場合 (コンパイラ : CC-RL)</p>
		30	<p>5.4 ファイル構成</p> <p>サンプルコードのフォルダ名を変更</p> <p>アプリケーションノート番号を変更</p>
		64	<p>6.1.2 r_qspi_flash_n25q_sfr.h</p> <p>記述例を更新した。</p>

## 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/>