

## RX210グループ

R01AN1484JJ0100

Rev.1.00

2013.04.01

## シングルチップモードによる UART 経由での 内蔵フラッシュメモリ書き換え

### 要旨

本アプリケーションノートは、マスタから調歩同期式シリアル通信で送信される、消去ブロック番号、書き込みデータのサイズ、および書き込みデータを使用してコード格納用フラッシュメモリ(ユーザ領域)の書き込み/消去する処理について説明しています。

なお、本アプリケーションノートでは以下のアプリケーションノートのサンプルコードを使用しています。

RX210 の初期設定：

「RX210 グループ 初期設定例」 Rev.2.00 (R01AN1002JJ0200)

内蔵フラッシュメモリの消去/書き込み：

「RX200 シリーズ RX200 用のシンプルフラッシュ API」 Rev.1.01 (R01AN0823JU0101)

### 対象デバイス

- ・RX210グループ 100ピン版 ROM容量：128KB～512KB
- ・RX210グループ 80ピン版 ROM容量：128KB～512KB
- ・RX210グループ 64ピン版 ROM容量：128KB～512KB
- ・RX210グループ 48ピン版 ROM容量：128KB～256KB

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。

## 目次

1. 仕様	4
2. 動作確認条件	6
3. 関連アプリケーションノート	6
4. ハードウェア説明	7
4.1 ハードウェア構成例	7
4.2 使用端子一覧	8
5. ソフトウェア説明	9
5.1 動作概要	9
5.1.1 調歩同期式シリアル通信仕様	9
5.1.2 通信コマンド仕様	9
5.1.3 通信フロー	10
5.1.4 消去ブロック番号	13
5.1.5 書き込みデータサイズ	13
5.1.6 オーバランエラー	14
5.1.7 フレーミングエラー	14
5.2 ユーザ領域の書き込み/消去	14
5.2.1 ユーザ領域の消去処理	14
5.2.2 ユーザ領域の書き込み処理	14
5.2.3 シンプルフラッシュAPIの変更点	14
5.2.4 割り込みを使用する場合の注意点	15
5.3 スレーブ準備完了処理	15
5.4 LED表示	16
5.5 ハンドシェイク制御	17
5.6 セクション設定	17
5.7 ファイル構成	18
5.8 オプション設定メモリ	19
5.9 定数一覧	20
5.10 構造体/共用体一覧	21
5.11 変数一覧	21
5.12 関数一覧	22
5.13 関数仕様	23
5.14 フローチャート	27
5.14.1 初期設定関数	27
5.14.2 初期化処理	28
5.14.3 メイン関数	30
5.14.4 書き込み/消去処理	31
5.14.5 正常処理関数	34
5.14.6 エラー終了処理関数	34
5.14.7 1バイトデータ受信関数	35
5.14.8 nバイトデータ受信関数	36
5.14.9 1バイトデータ送信関数	37
6. 注意事項	38
6.1 消去ブロックEB00の書き換え時の注意事項	38
6.2 ROM容量の変更	38
6.3 動作モードの設定	38
6.4 エンディアンの設定	39
6.4.1 リトルエンディアン使用時	39
6.4.2 ビッグエンディアン使用時	39

7. サンプルコード.....	40
8. 参考ドキュメント.....	40

## 1. 仕様

本アプリケーションノートでは、シングルチップモードでユーザ領域の書き込み／消去を行います。

スレーブは、マスタから消去ブロック番号、書き込みデータのサイズ、および書き込みデータを調歩同期式シリアル通信で受信し、ユーザ領域の書き込み／消去を行います。

マスタとスレーブ間の調歩同期式シリアル通信は、SCI チャンネル 0(SCI0)モジュールを使用します。

調歩同期式シリアル通信仕様は、以下の設定とします。

ビットレート : 31250bps

データ長 : 8 ビット

パリティビット : なし

ストップビット : 1 ビット

- ① 本アプリケーションノートでは、スレーブの通信準備が完了したときに P13 端子の出力レベルを Low に切り替え、マスタに通信準備が完了したことを通知します。マスタ側でスレーブの通信準備完了を監視している端子と接続してください。
- ② スレーブとマスタは通信制御するためにハンドシェイクしています。スレーブはマスタから受信したデータに対応する処理を実行後、マスタへ[ACCEPTABLE]コマンド(55h)を送信します。マスタはスレーブから[ACCEPTABLE]コマンドを受信すると次のシリアル送信を開始します。
- ③ スレーブは指定された消去ブロックに対して消去を行い、受信した書き込みデータを指定された消去ブロックの先頭アドレスから書き込みます。  
なお、本アプリケーションノートでは EB00～EB03 にプログラムを配置しているため、消去／書き込みできません。EB00～EB03 の消去ブロック番号を指定された場合はエラーになります。
- ④ スレーブは、正常にユーザ領域の消去／書き込み処理が完了すると、I/O ポートに接続された 4 個の LED で正常終了を知らせ、P13 端子の出力レベルを High(スレーブ準備未完了状態)に戻します。  
なお、マスタとの通信中および消去／書き込み処理中にエラーが発生した場合は、スレーブ準備未完了状態に戻し、LED でそのエラー状態を知らせます。

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。

表1.1 使用する周辺機能と用途

周辺機能	用途
ROM(コード格納用フラッシュメモリ)	ROM P/E モードによる 内蔵フラッシュメモリの書き換え
シリアルコミュニケーションインターフェース	マスタとの通信用の調歩同期式シリアル

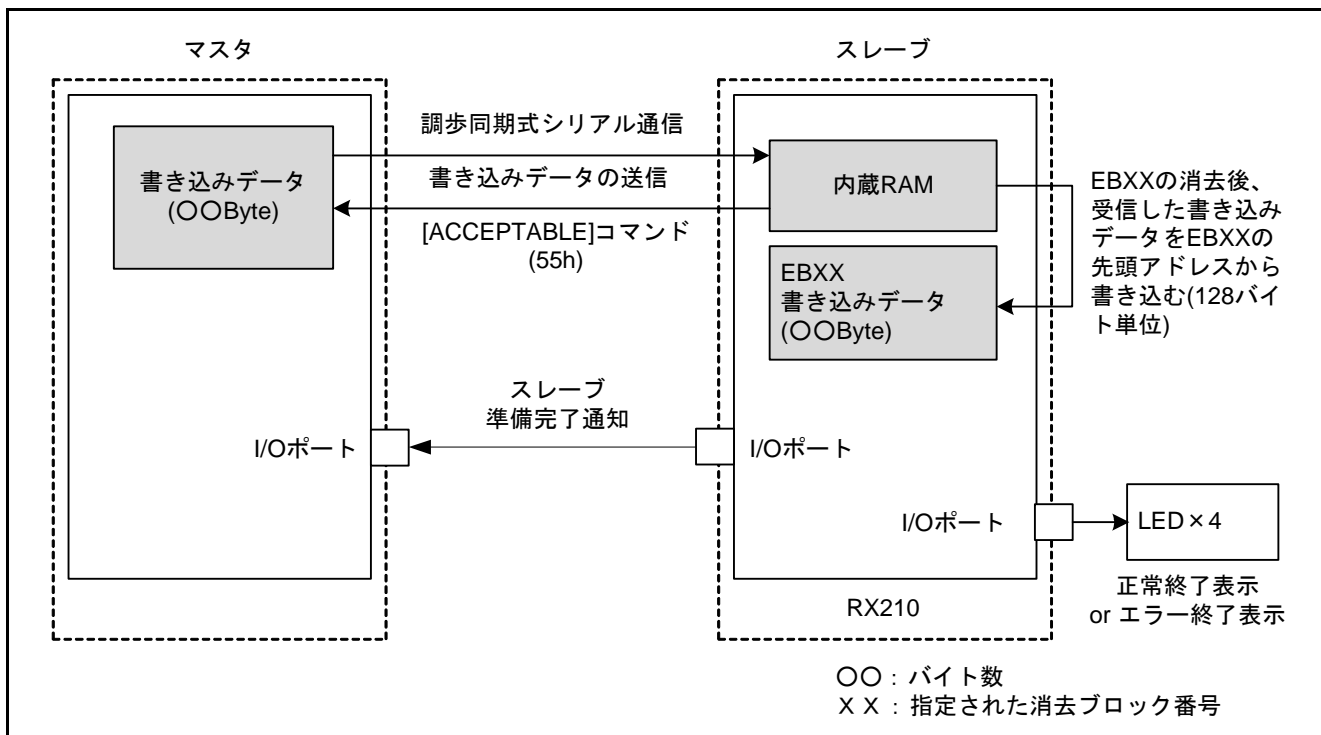


図1.1 使用例

## 2. 動作確認条件

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

表2.1 動作確認条件

項目	内容
使用マイコン	R5F52108ADFP (RX210 グループ)
動作周波数	<ul style="list-style-type: none"> <li>● メインクロック: 20MHz</li> <li>● PLL: 100MHz (メインクロック 2 分周 10 通倍)</li> <li>● システムクロック (ICLK): 50MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック B (PCLKB): 25MHz (PLL 4 分周)</li> <li>● FlashIF クロック (FCLK): 25MHz(PLL 4 分周)</li> </ul>
動作電圧	5.0V
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
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.1.02 Release 01 コンパイルオプション -cpu=rx200 -output=obj="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo (統合開発環境のデフォルト設定を使用しています)
iodefine.h のバージョン	Version 1.2A
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit for RX210 (製品型名: R0K505210C000BE)

## 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。

RX200 シリーズ RX200 用のシンプルフラッシュ API Rev.1.01 (R01AN0823JU0101\_RX200)

RX210 グループ 初期設定例 Rev.2.00 (R01AN1002JJ0200\_RX210)

上記アプリケーションノートの初期設定関数を、本アプリケーションノートのサンプルコードで使用しています。Rev は本アプリケーションノート作成時点のものです。

最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

## 4. ハードウェア説明

### 4.1 ハードウェア構成例

図 4.1に本アプリケーションノートにおけるスレーブのハードウェア構成図を示します。

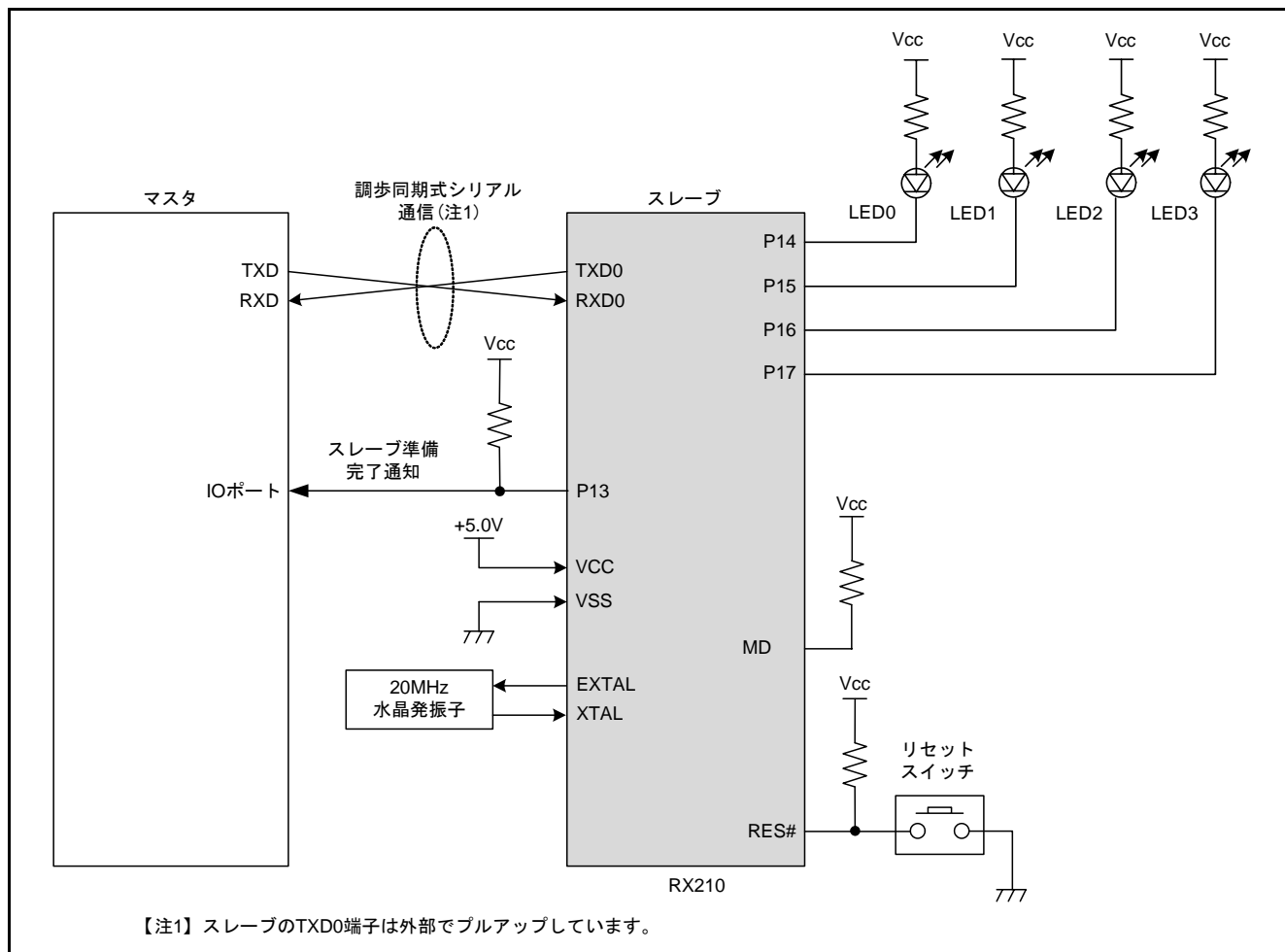


図4.1 スレーブのハードウェア構成図

## 4.2 使用端子一覧

表 4.1に使用端子と機能を示します。

使用端子は 100 ピン版の製品を想定しています。100 ピン版未満の製品を使用する場合は、使用する製品に合わせて端子を選択してください。

表4.1 使用端子と機能

端子名	入出力	内容
P20/TXD0	出力	マスタ通信用シリアル送信端子
P21/RXD0	入力	マスタ通信用シリアル受信端子
P13	出力	スレーブ準備完了通知用端子
P14	出力	LED0 接続端子("High"出力：消灯、"Low"出力：点灯)
P15	出力	LED1 接続端子("High"出力：消灯、"Low"出力：点灯)
P16	出力	LED2 接続端子("High"出力：消灯、"Low"出力：点灯)
P17	出力	LED3 接続端子("High"出力：消灯、"Low"出力：点灯)



## 5. ソフトウェア説明

### 5.1 動作概要

#### 5.1.1 調歩同期式シリアル通信仕様

本アプリケーションノートでは、マスタとスレーブ間で調歩同期式シリアル通信により、通信コマンド ([FSTART]、[ERASE]、[WRITE])、消去ブロック番号、書き込みデータサイズ、書き込みデータの受信を行います。また、スレーブからはハンドシェイク用のステータスコマンドとして[ACCEPTABLE]コマンド(55h)を送信します。

表 5.1に調歩同期式シリアル通信仕様を示します。

表 5.1 調歩同期式シリアル通信仕様

項目	仕様
チャンネル	SCI チャンネル 0(SCI0)
シリアル通信方式	調歩同期式モード
転送速度	31250bps(PCLKB=25MHz 時)
データ長	8 ビット
パリティビット	なし
ストップビット	1 ビット
エラー	オーバランエラー、フレーミングエラー

#### 5.1.2 通信コマンド仕様

マスタとスレーブ間の通信コマンドの仕様を表 5.2に示します。

表 5.2 通信コマンド仕様

コマンド	値	説明	通信方向
FSTART	10h	スレーブのユーザ領域の書き込み／消去処理を開始するためのコマンド	マスタ → スレーブ
ERASE	11h	スレーブのユーザ領域の消去を開始するためのコマンド	マスタ → スレーブ
WRITE	12h	スレーブのユーザ領域の書き込みを開始するためのコマンド	マスタ → スレーブ
ACCEPTABLE	55h	スレーブがマスタヘデータ受信可能であることを通知するためのステータスコマンド	スレーブ → マスタ

5.1.3 通信フロー

マスタとスレーブ間の通信フローを図 5.1～図 5.3に示します。

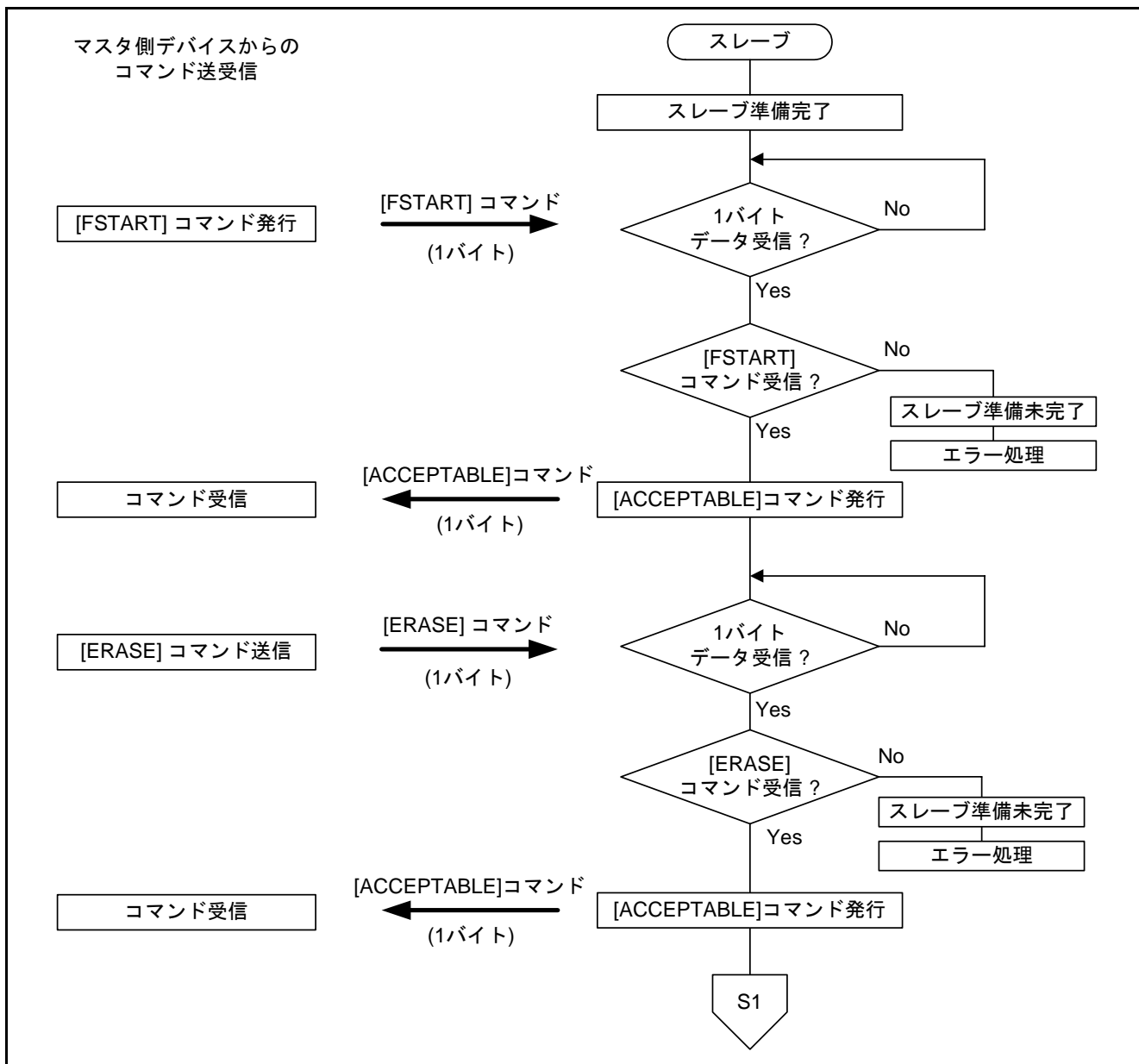


図 5.1 通信フロー(1)

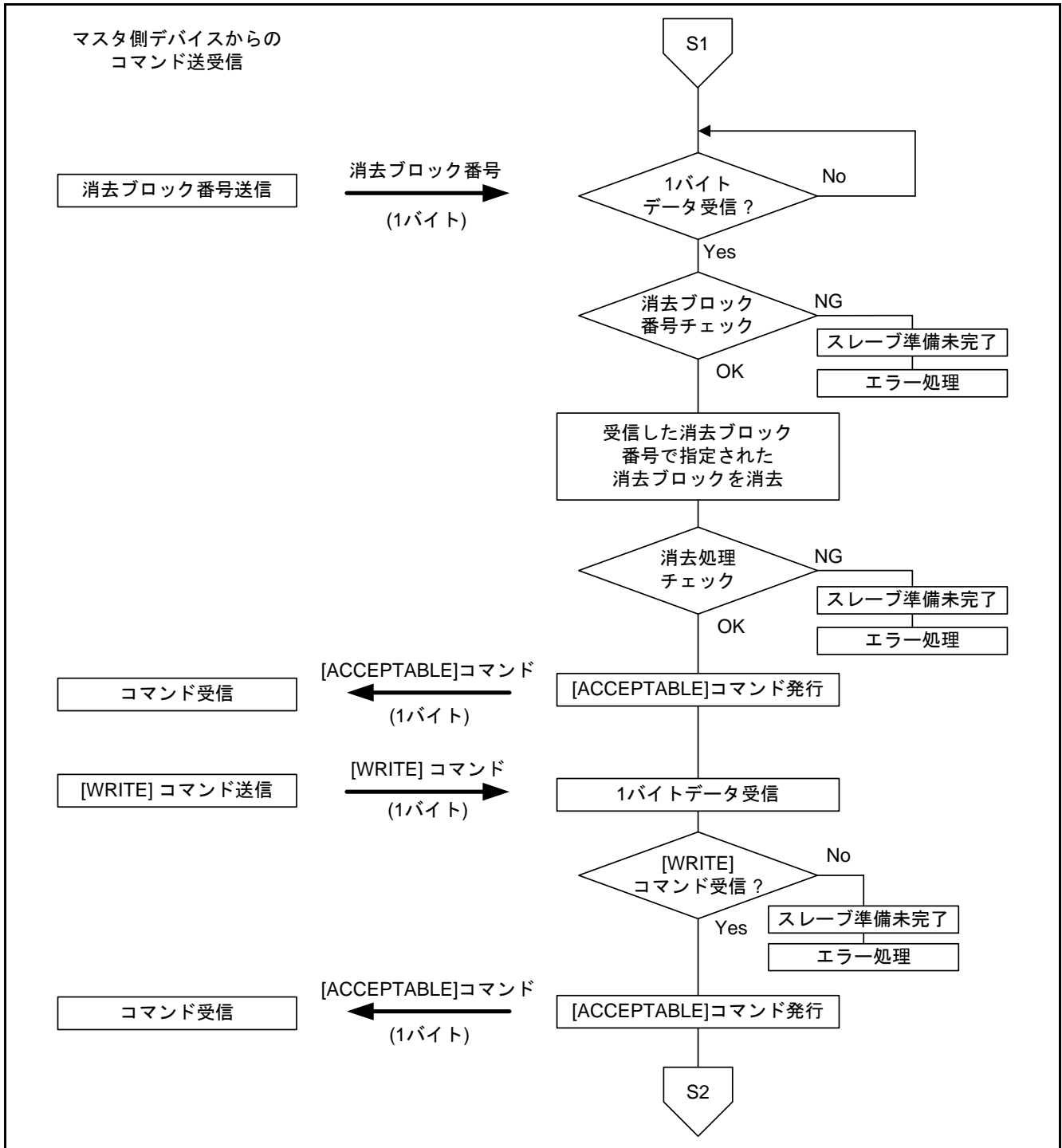


図 5.2 通信フロー(2)

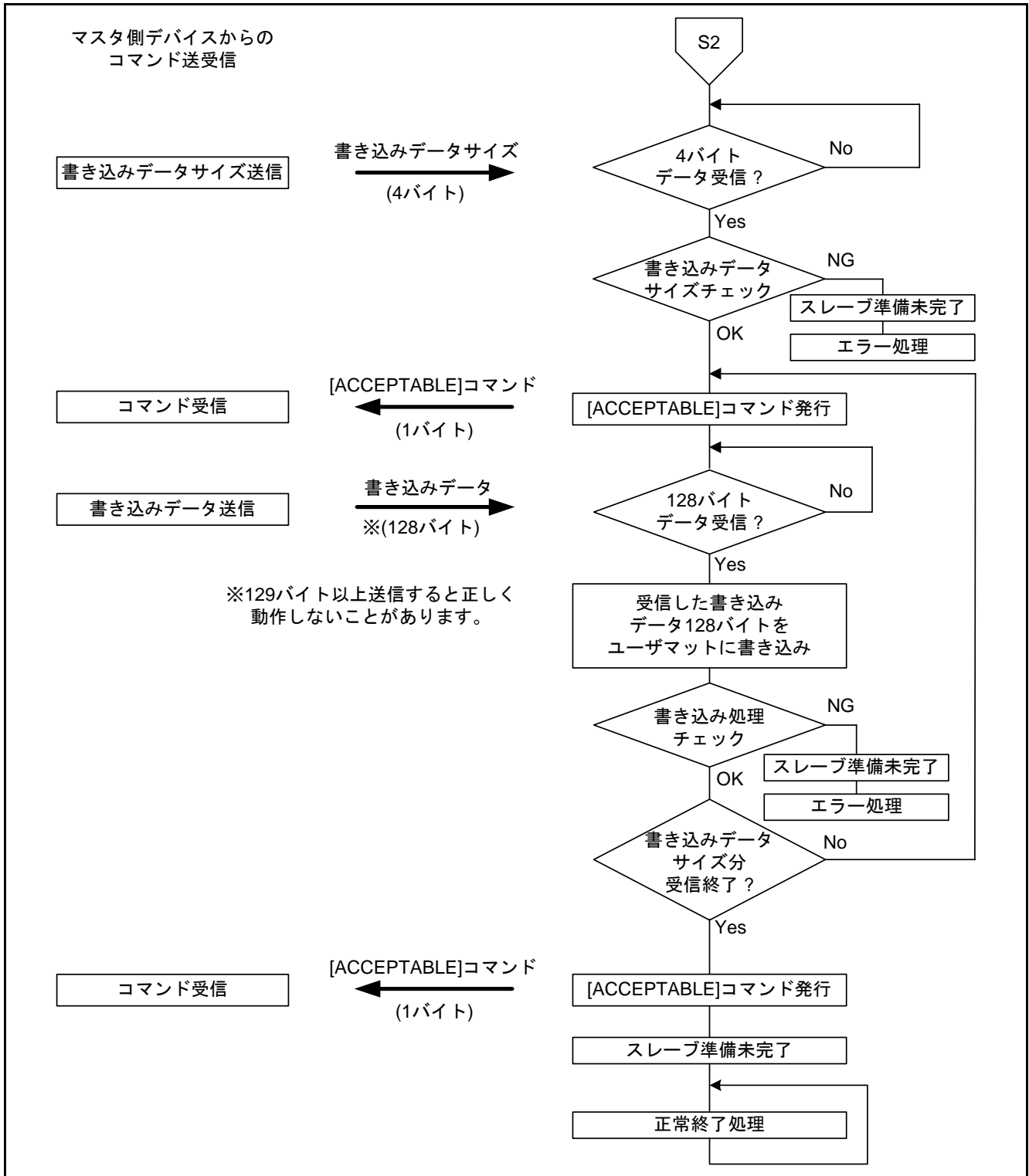


図 5.3 通信フロー(3)

#### 5.1.4 消去ブロック番号

スレーブは、マスタから[ERASE]コマンド受信後に1バイトの消去ブロック番号(定義された1バイトのデータ)を受信します。

図 5.4に消去ブロック番号の仕様を示します。また、消去ブロック番号の詳細に関しては「5.2.1 ユーザ領域の消去処理」を参照してください。

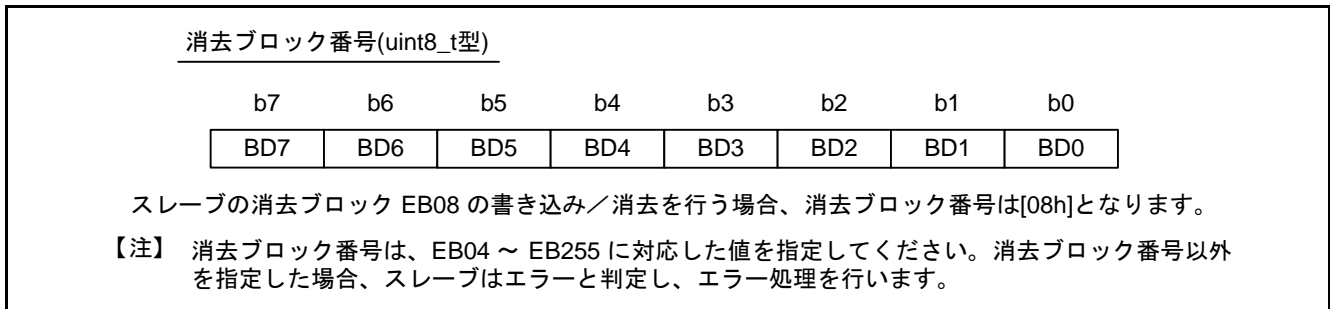


図 5.4 消去ブロック番号仕様

#### 5.1.5 書き込みデータサイズ

スレーブは、マスタから[WRITE]コマンド受信後に4バイトの書き込みデータサイズを受信します。図 5.5に書き込みデータサイズの仕様を示します。



図 5.5 書き込みデータサイズ仕様

### 5.1.6 オーバランエラー

本アプリケーションノートでは、スレーブの調歩同期式シリアル通信の受信時にオーバランエラーが発生 (SCIO.SSR.ORER ビットが 1 にセット) した場合には、エラー処理します。

### 5.1.7 フレーミングエラー

本アプリケーションノートでは、スレーブの調歩同期式シリアル通信の受信時にフレーミングエラーが発生 (SCIO.SSR.FER ビットが 1 にセット) した場合には、エラー処理します。

## 5.2 ユーザ領域の書き込み/消去

本アプリケーションノートの書き込みおよび消去で使用している「RX200 用のシンプルフラッシュAPI」については、8. 参考ドキュメント「RX200 用のシンプルフラッシュAPI」を参照してください。

### 5.2.1 ユーザ領域の消去処理

ユーザ領域の消去処理は「RX200 用のシンプルフラッシュ API」で提供される R\_FlashErase 関数を使用しています。したがって、消去するブロック番号も R\_FlashErase 関数に指定する値と共通にしています。消去エラーは R\_FlashErase 関数から戻る値となります。

### 5.2.2 ユーザ領域の書き込み処理

ユーザ領域の書き込みは「RX200 用のシンプルフラッシュ API」で提供される R\_FlashWrite 関数を使用しています。書き込みエラーは R\_FlashWrite 関数から戻る値となります。

### 5.2.3 シンプルフラッシュ API の変更点

シンプルフラッシュ API から”r\_Flash\_API\_RX200\_UserConfig.h”を変更しています。

表 5.3に「r\_Flash\_API\_RX200\_UserConfig.h」の変更点を示します。

表 5.3 「r\_Flash\_API\_RX200\_UserConfig.h」の変更点

変更項目	変更箇所
シンプルフラッシュ API の設定変更	<pre>#define ROM_SIZE          524288 //#define ROM_SIZE        131072 #define FLASH_CLOCK_FREQUENCY 25 //#define DATA_FLASH_BGO</pre>

### 5.2.4 割り込みを使用する場合の注意点

FLASH ROM の書き込み、および消去中は ROM へのアクセスはできません。したがって、ROM の書き込み、および消去中に発生する割り込みによる ROM へのアクセスを抑止する必要があります。本アプリケーションノートでは割り込みを使用していないので、割り込みの抑止は行っていません。割り込みをご使用になる場合、サンプルコードに『#define FOR\_INTERRUPTION』の定義を追加することにより、赤線で囲まれた箇所が有効となり、割り込みの抑止になります(サンプルコードは『#define FOR\_INTERRUPTION』を定義していないため、赤線で囲まれた箇所は無効となっています)。

以下に図 5.6の赤線で囲まれた①～③の処理内容を示します。

- ① 変数名 flash\_pipl に現在の IPL(プロセッサ割り込み優先レベル)の優先レベルの値を保存します。
- ② 書き込み/消去の処理の開始前にIPLの優先レベルを『FLASH\_READY\_IPL』に変更します。なお、サンプルコードでは『FLASH\_READY\_IPL』の値を5と定義しており、IPRの設定が5以下の割り込みは禁止となります。『FLASH\_READY\_IPL』の詳細は8. 参考ドキュメント「RX200用のシンプルフラッシュAPI」を参照してください。
- ③ 書き込み/消去の処理の終了後に IPL の優先レベルを元の値に戻します。

上記処理により、IPL の優先レベルを一時的に変更することで、割り込みによる ROM へのアクセスを抑止しています。

```

#ifdef FOR_INTERRUPTION
/* Holds IPL of processor before flash operation */
static unsigned char flash_pipl;

① → /* Save current processor IPL */
flash_pipl = get_ipi();           /* If your system is using the interrupt, Enable this line. */

② → /* Set the processor IPL so that interrupts that access ROM will not occur during ROM program/erase operations. */
set_ipi(FLASH_READY_IPL);       /* If your system is using the interrupt, Enable this line. */
#endif

/* Erasure process by using "simple API" */
fcu_status = R_FlashErase((uint16_t)target_eb);

/* Programming process by using "simple API" */
fcu_status = R_FlashWrite((uint32_t)fcu_info.p_write_adrs_now,
(uint32_t)wtrdata_buffer,ROM_PROGRAM_SIZE); /* Program 128 byte data to the target EB by using "simpleAPI"
* (call of R_FlashWrite function in "simpleAPI") */

③ → #ifdef FOR_INTERRUPTION
/* Restore processor IPL */
set_ipi(flash_pipl);           /* If your system is using the interrupt, Enable this line. */
#endif

```

図 5.6 割り込みの抑止例

### 5.3 スレーブ準備完了処理

スレーブ側の通信準備が完了後、スレーブ準備完了通知の端子である P13 から Low レベルを出力します。マスタ側でスレーブの通信準備完了を監視している端子と接続してください。

P13 から High レベルが出力している時はスレーブ準備未完了状態を示し、ユーザ領域の書き込み/消去を受け付けません。

## 5.4 LED 表示

本サンプルコードの動作状態におけるLED表示を図 5.7に示します。

エラー番号	動作状態	LED表示				順序
		LED3	LED2	LED1	LED0	
	正常終了時 (一定間隔でシフト表示)	●	●	●	○	
		●	●	○	●	
		●	○	●	●	
		○	●	●	●	
エラーNo.01	[FSTART]コマンドエラー発生時	●	●	●	⊗	
エラーNo.02	[ERASE]コマンドエラー発生時	●	●	⊗	●	
エラーNo.03	消去ブロック番号エラー発生時	●	●	⊗	⊗	
エラーNo.04	消去処理エラー発生時	●	⊗	●	●	
エラーNo.05	[WRITE]コマンドエラー発生時	●	⊗	●	⊗	
エラーNo.06	書き込みデータサイズ エラー発生時	●	⊗	⊗	●	
エラーNo.07	書き込み処理エラー発生時	●	⊗	⊗	⊗	
エラーNo.08	オーバランエラー発生時	⊗	●	●	●	
エラーNo.09	フレーミングエラー発生時	⊗	●	●	⊗	

点灯： ○      点滅： ⊗      消灯： ●

図 5.7 LED 表示



## 5.5 ハンドシェイク制御

スレーブは通信制御するためにマスタとハンドシェイクしています。

ハンドシェイク制御としてスレーブはマスタからのシリアル通信受信後に、受信データに対する処理を実行し、次のシリアル通信が受信可能になってから[ACCEPTABLE]コマンド(55h)をマスタへ返信します。マスタはスレーブから[ACCEPTABLE]コマンドを受信すると次のシリアル送信を開始します。

## 5.6 セクション設定

スレーブのセクション設定を表 5.4に示します。

表5.4 スレーブのセクション設定

セクション名	開始アドレス	説明
RPFRAM	0000 0000h	[PFRAM]セクションを ROM 化支援オプションにより RAM 上にマップした領域
B_1	0000 1000h	未初期化データ領域(ALIGN = 1)
R_1		[D_1]セクションを ROM 化支援オプションにより RAM 上にマップした領域
B_2		未初期化データ領域(ALIGN = 2)
R_2		[D_2]セクションを ROM 化支援オプションにより RAM 上にマップした領域
B		未初期化データ領域(ALIGN = 4)
R		[D]セクションを ROM 化支援オプションにより RAM 上にマップした領域
SU		ユーザスタック領域
SI		割り込みスタック領域
PResetPRG		FFFF E000h
P	プログラム領域	
PIntPRG	プログラム領域(割り込みプログラム)	
C_1	FFFF EE00h	定数領域(ALIGN = 1)
C_2		定数領域(ALIGN = 2)
C		定数領域(ALIGN = 4)
C\$*		未初期化データ領域のセクション初期化用テーブル, 可変ベクタ領域
D_1		初期化データ領域(ALIGN = 1)
D_2		初期化データ領域(ALIGN = 2)
D		初期化データ領域(ALIGN = 4)
W*		switch 文分岐テーブル領域
PFRAM		FFFF F800h
FIXEDVECT	FFFF FFD0h	固定ベクタ領域

## 5.7 ファイル構成

表 5.5 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されるファイルのうち、内容を変更していないファイルは除きます。

表 5.5 サンプルコードで使用するファイル

ファイル名	概要	備考
r_init_stop_module.c	リセット後に動作している周辺機能の停止	
r_init_stop_module.h	r_init_stop_module.c のヘッダファイル	
r_init_non_existent_port.c	存在しないポートの初期設定	
r_init_non_existent_port.h	r_init_non_existent_port.c のヘッダファイル	
r_init_clock.c	クロック初期設定	
r_init_clock.h	r_init_clock.c のヘッダファイル	
resetprg.c	初期設定	PowerON_Reset_PC 関数内の HardwareSetup 関数の呼び出しのコメントアウトを解除して、main.c ファイル内の HardwareSetup 関数を PowerON_Reset_PC 関数から呼び出すように変更
dbsect.c	RPFRAM 領域の初期化	{ __sectop("PFRAM"), __sectend("PFRAM"), __sectop("RPFRAM") } を追加
main.c	メイン処理、マスタとの調歩同期式シリアル通信による通信コマンドの送受信制御、消去ブロック番号、書き込みデータサイズ、および書き込みデータの受信制御、正常終了時、およびエラー発生時の LED の表示制御	
r_Flash_API_RX200.c	RX200 シリーズ RX200 用のシンプルフラッシュ API のプログラム	詳細は RX200 シリーズ RX200 用のシンプルフラッシュ API のアプリケーションノートを参照してください。
r_Flash_API_RX200.h	RX200 シリーズ RX200 用のシンプルフラッシュ API のプログラムの外部参照用インクルードヘッダ	
r_Flash_API_RX200_User Config.h	RX200 シリーズ RX200 用のシンプルフラッシュ API のパラメータ設定用インクルードヘッダ	

## 5.8 オプション設定メモリ

表 5.6にサンプルコードで使用するオプション設定メモリの状態を示します。必要に応じて、お客様のシステムに最適な値を設定してください。なお、本サンプルコードではマスタ側とスレーブ側でエンディアンを合わせてご使用ください。

表5.6 サンプルコードで使用するオプション設定メモリ

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh~FFFF FF8Ch	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh~FFFF FF88h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効 リセット後、HOCO 発振が無効
MDES(*1)	FFFF FF83h~FFFF FF80h	FFFF FFFFh	リトルエンディアン

(\*1): 本サンプルコードの設定はリトルエンディアンです。エンディアンの切り替えは6.4 エンディアンの設定を参照ください。

## 5.9 定数一覧

表 5.7、表 5.8にサンプルコードで使用する定数を示します。

表5.7 サンプルコードで使用する定数(1)

定数名	設定値	内容
FSTART	0x10	書き込み／消去開始コマンド
ERASE	0x11	消去開始コマンド
WRITE	0x12	書き込み開始コマンド
ACCEPTABLE	0x55	スレーブから送信されるステータスコマンド
LED_ON	0	LED 点灯時の設定値
LED_OFF	1	LED 消灯時の設定値
RSK_LED0	PORT1.PODR.BIT.B4	評価ボード搭載 LED0 の点灯／消灯制御
RSK_LED1	PORT1.PODR.BIT.B5	評価ボード搭載 LED1 の点灯／消灯制御
RSK_LED2	PORT1.PODR.BIT.B6	評価ボード搭載 LED2 の点灯／消灯制御
RSK_LED3	PORT1.PODR.BIT.B7	評価ボード搭載 LED3 の点灯／消灯制御
RSK_LED0_PDR	PORT1.PDR.BIT.B4	評価ボード搭載 LED0 の入出力制御
RSK_LED1_PDR	PORT1.PDR.BIT.B5	評価ボード搭載 LED1 の入出力制御
RSK_LED2_PDR	PORT1.PDR.BIT.B6	評価ボード搭載 LED2 の入出力制御
RSK_LED3_PDR	PORT1.PDR.BIT.B7	評価ボード搭載 LED3 の入出力制御
NOT_READY	1	スレーブ準備未完了を示す出力レベル
READY	0	スレーブ準備完了を示す出力レベル
SLAVE_READY_PODR	PORT1.PODR.BIT.B3	スレーブ準備完了の確認用ポートの完了／未完了制御
SLAVE_READY_PDR	PORT1.PDR.BIT.B3	スレーブ準備完了の確認用ポートの入出力制御
WAIT_LED	2000000	スレーブのユーザ領域の書き込み／消去が正常に終了した際に表示するLEDの点灯間隔の時間データ
ROM_PROGRAM_SIZE_LARGE	128(*1)	対象デバイスに応じた、ユーザ領域への書き込み単位が設定されます。r_Flash_API_RX200.h に記述されており、インクルードすることで参照するようにしています
RXD0_PMR	PORT2.PMR.BIT.B1	RXD0 のポートモードレジスタ設定
TXD0_PMR	PORT2.PMR.BIT.B0	TXD0 のポートモードレジスタ設定
RXD0_PDR	PORT2.PDR.BIT.B1	RXD0 のポート方向レジスタ設定
TXD0_PODR	PORT2.PODR.BIT.B0	TXD0 のポート出力データレジスタ設定
TXD0_PDR	PORT2.PDR.BIT.B0	TXD0 のポート方向レジスタ設定

(\*1) : RX210 グループを対象デバイスにした場合の値になります。

表5.8 サンプルコードで使用する定数(2)

定数名	設定値	内容
ERROR_NO_01	1	エラー状態を示すデータ
ERROR_NO_02	2	
ERROR_NO_03	3	
ERROR_NO_04	4	
ERROR_NO_05	5	
ERROR_NO_06	6	
ERROR_NO_07	7	
ERROR_NO_08	8	
ERROR_NO_09	9	
WRITE_ADRS_TOP_2K	ROM_PE_ADDR	書き込み／消去用アドレス空間におけるブロックサイズ 2K バイト領域の先頭アドレス
BLK_SIZE_2K	2×1024	EB00～EB255 の各ブロックサイズ

## 5.10 構造体/共用体一覧

図 5.8にサンプルコードで使用する構造体を示します。

```
typedef struct
{
    uint8_t *p_write_adrs_top; /* 書き込み時の対象消去ブロックの先頭アドレス */
    uint8_t *p_write_adrs_end; /* 書き込み時の対象消去ブロックの終了アドレス */
    uint8_t *p_write_adrs_now; /* 書き込み時の書き込み先アドレス */
    uint32_t eb_block_size; /* 対象消去ブロックのブロックサイズ */
}st_fcu_info_t;
```

図 5.8 サンプルコードで使用する構造体

## 5.11 変数一覧

表 5.9にグローバル変数を示します。

表5.9 グローバル変数

型	変数名	内容	使用関数
uint8_t	wrdata_buffer [ROM_PROGRAM_SIZE_LARGE]	マスタから受信した 128 バイトの書き込みデータを格納する配列(128 バイト)	Flash_Update
st_fcu_info_t	fcu_info	ユーザ領域の書き込み／消去時に使用する FCU 関連のアドレス情報を格納する構造体(16 バイト)	Flash_Update

## 5.12 関数一覧

表 5.10に関数を示します。ただし、シンプルフラッシュAPIで使用しているものは除きます。

表5.10 関数

関数名	概要
HardwareSetup	MCU 初期設定関数
R_INIT_StopModule	リセット後に動作している周辺機能の停止
R_INIT_NonExistentPort	存在しないポートの初期設定
R_INIT_Clock	クロック初期設定
main	メイン処理
Flash_Update	ユーザ領域の書き込み/消去処理関数
Indicate_Ending_LED	正常終了処理関数
Indicate_Error_LED	エラー終了処理関数
SCI_Rcv1byte	1バイトデータ受信関数
SCI_Rcvnbyte	nバイトデータ受信関数
SCI_Trns1byte	1バイトデータ送信関数

## 5.13 関数仕様

サンプルコードの関数仕様を示します。

HardwareSetup	
概要	MCU 初期設定関数
ヘッダ	iodef.h、r_init_stop_module.h、r_init_clock.h、r_init_non_existent_port.h
宣言	void HardwareSetup (void)
説明	HardwareSetup 関数は MCU の初期設定を行います。 <ul style="list-style-type: none"> <li>リセット後に動作している周辺機能の停止</li> <li>存在しないポートの初期設定 (100pin 版)</li> <li>クロック初期設定 (システムクロック (ICLK) および周辺モジュールクロック B (PCLKB) などの設定)</li> <li>LED0~LED3 を接続している I/O ポート (P14、P15、P16、および P17) の初期出力設定</li> <li>モジュールストップ状態解除</li> <li>マルチファンクションピンコントローラ (MPC)、ポートモードレジスタ (PMR) の設定</li> <li>SCI0 の初期設定</li> </ul>
引数	なし
リターン値	なし

R_INIT_StopModule	
概要	リセット後に動作している周辺機能の停止
ヘッダ	r_init_stop_module.h
宣言	void R_INIT_StopModule(void)
説明	モジュールストップ状態へ遷移する設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、モジュールストップ状態への遷移は行っていません。 本関数の詳細は、アプリケーションノート「RX210グループ 初期設定例 Rev.2.00」を参照してください。

R_INIT_NonExistentPort	
概要	存在しないポートの初期設定
ヘッダ	r_init_non_existent_port.h
宣言	void R_INIT_NonExistentPort(void)
説明	100 ピン未満の製品に対して、存在しないポートの端子に対応するポート方向レジスタの初期設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、100 ピン版 (PIN_SIZE=100) に設定しています。 本関数をコールした後に、存在しないポートを含む PDR、PODR レジスタへバイト単位で書き込む場合、存在しないポートの方向制御ビットには“1”、ポート出力データ格納ビットには“0”を設定してください。 本関数の詳細は、アプリケーションノート「RX210グループ 初期設定例 Rev.2.00」を参照してください。

---

R_INIT_Clock	
概要	クロック初期設定
ヘッダ	r_init_clock.h
宣言	void R_INIT_Clock(void)
説明	クロックの初期設定を行います。
引数	なし
リターン値	なし
備考	サンプルコードでは、システムクロックを PLL とし、サブクロックを使用しない処理を選択しています。 本関数の詳細は、アプリケーションノート「RX210グループ 初期設定例 Rev.2.00」を参照してください。

---

main	
概要	メイン処理
ヘッダ	iodefine.h
宣言	void main (void)
説明	main 関数は、P13 から Low レベルを出力することで、準備完了したことを通知します。その後、マスタからの 1 バイトデータの受信制御、エラー発生時に Indicate_Error_LED 関数の呼び出し、内蔵 RAM 上のユーザ領域書き込み／制御プログラム(Flash_Update 関数)の呼び出しを行います。
引数	なし
リターン値	なし

---



Flash_Update	
概要	ユーザ領域の書き込み／消去処理関数
ヘッダ	iodefine.h、r_Flash_API_RX200.h、r_Flash_API_RX200_UserConfig.h
宣言	void Flash_Update (void)
説明	Flash_Update 関数は、マスタから消去ブロック番号、書き込みデータサイズ、書き込みデータを調歩同期式シリアル通信で受信し、ユーザ領域の書き込み／消去を行います。 正常終了時には Indicate_Ending_LED 関数、エラー終了時には Indicate_Error_LED 関数を呼び出します。 <ul style="list-style-type: none"> <li>マスタから FSTART の通信コマンドを受信し、処理を開始</li> <li>マスタから ERASE の通信コマンドを受信後に 1 バイトの消去ブロック番号を受信し、R_FlashErase 関数を呼び出し、ブロックの消去を行う</li> <li>マスタから WRITE の通信コマンドを受信後に、4 バイトの書き込みデータサイズを受信</li> <li>マスタから 128 バイトのデータを受信し、書き込みデータサイズ分受信するまで R_FlashWrite 関数を呼び出し、ユーザ領域にデータを書き込む</li> <li>ユーザ領域の書き込み／消去が正常に終了した場合、スレーブ準備未完了状態を通知した後、Indicate_Ending_LED 関数を呼び出して正常終了を示す LED が点灯</li> </ul>
引数	なし
リターン値	なし

Indicate_Ending_LED	
概要	正常終了処理関数
ヘッダ	なし
宣言	void Indicate_Ending_LED (void)
説明	Indicate_Ending_LED 関数は、書き込み／消去が正常に終了した場合に、LED0～LED3 に正常終了を示す表示を行います。LED0～LED3 を順番に 1 つずつ点灯させます。
引数	なし
リターン値	なし

SCI_Rcv1byte	
概要	1 バイトデータ受信関数
ヘッダ	iodefine.h
宣言	uint8_t SCI_Rcv1byte (void)
説明	SCI_Rcv1byte 関数は、SCI0 の調歩同期式シリアル通信による 1 バイトデータの受信制御を行います。
引数	なし
リターン値	SCI0 の調歩同期式シリアル通信による 1 バイト受信データ
備考	なし

## Indicate\_Error\_LED

概要	エラー終了処理関数
ヘッダ	なし
宣言	void Indicate_Error_LED(uint8_t error_no)
説明	Indicate_Error_LED 関数は、ユーザ領域の書き込み／消去の処理中にエラーが発生した場合に、LED0～LED3 に発生したエラー番号の表示を行います。表示はエラー番号の表示と全消灯を繰り返します。
引数	uint8_t error_no : ユーザ領域の書き込み／消去中に発生したエラー番号 (*1)
リターン値	なし
備考	なし

【注】 \*1 エラー番号は、「5.4 LED表示」を参照してください。

## SCI\_Rcvnbyte

概要	n バイトデータ受信関数
ヘッダ	iodef.h
宣言	void SCI_Rcvnbyte(uint16_t size, uint8_t *rcv_buffer)
説明	SCI_Rcvnbyte 関数は、SCI0 の調歩同期式シリアル通信による n バイトデータ(n は uint16_t 型の第 1 引数)の受信制御を行います。
引数	uint16_t size : SCI0 の調歩同期式シリアル通信による受信データバイト数 uint8_t *rcv_buffer : 受信データ格納場所の先頭アドレス
リターン値	なし

## SCI\_Trns1byte

概要	1 バイトデータ送信関数
ヘッダ	iodef.h
宣言	void SCI_Trns1byte(uint8_t data)
説明	SCI_Trns1byte 関数は、SCI0 の調歩同期式シリアル通信による 1 バイトデータの送信制御を行います。
引数	uint8_t data : SCI0 の調歩同期式シリアル通信による 1 バイト送信データ
リターン値	なし

## 5.14 フローチャート

### 5.14.1 初期設定関数

図 5.9に初期設定関数のフローチャートを示します。

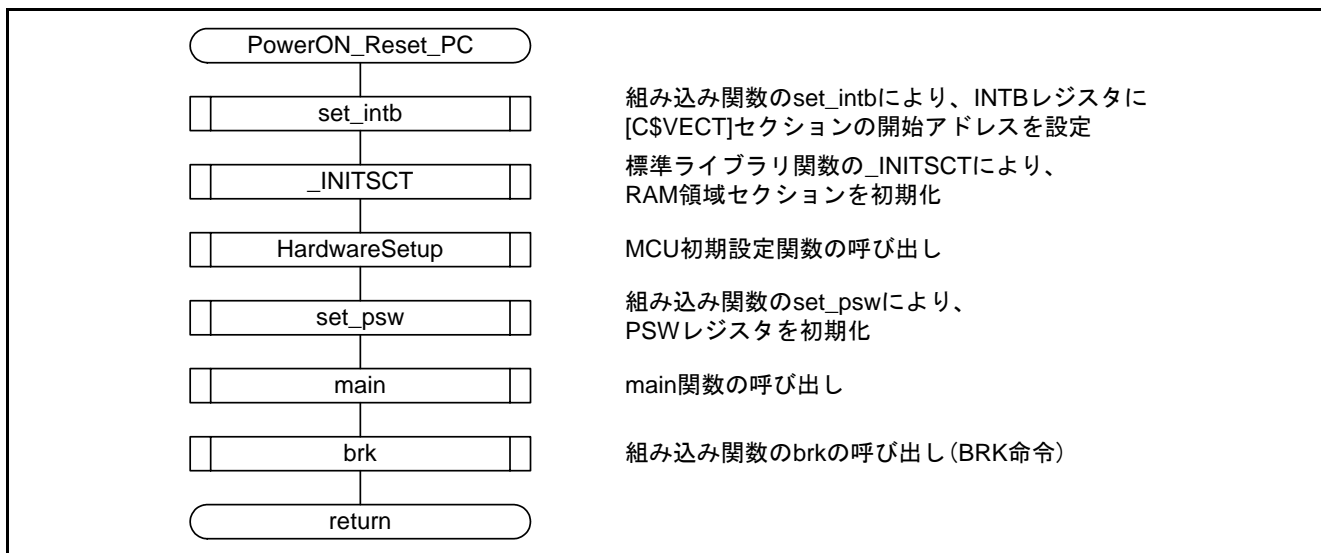


図 5.9 初期設定関数

## 5.14.2 初期化処理

図 5.10に初期化処理のフローチャートを示します。

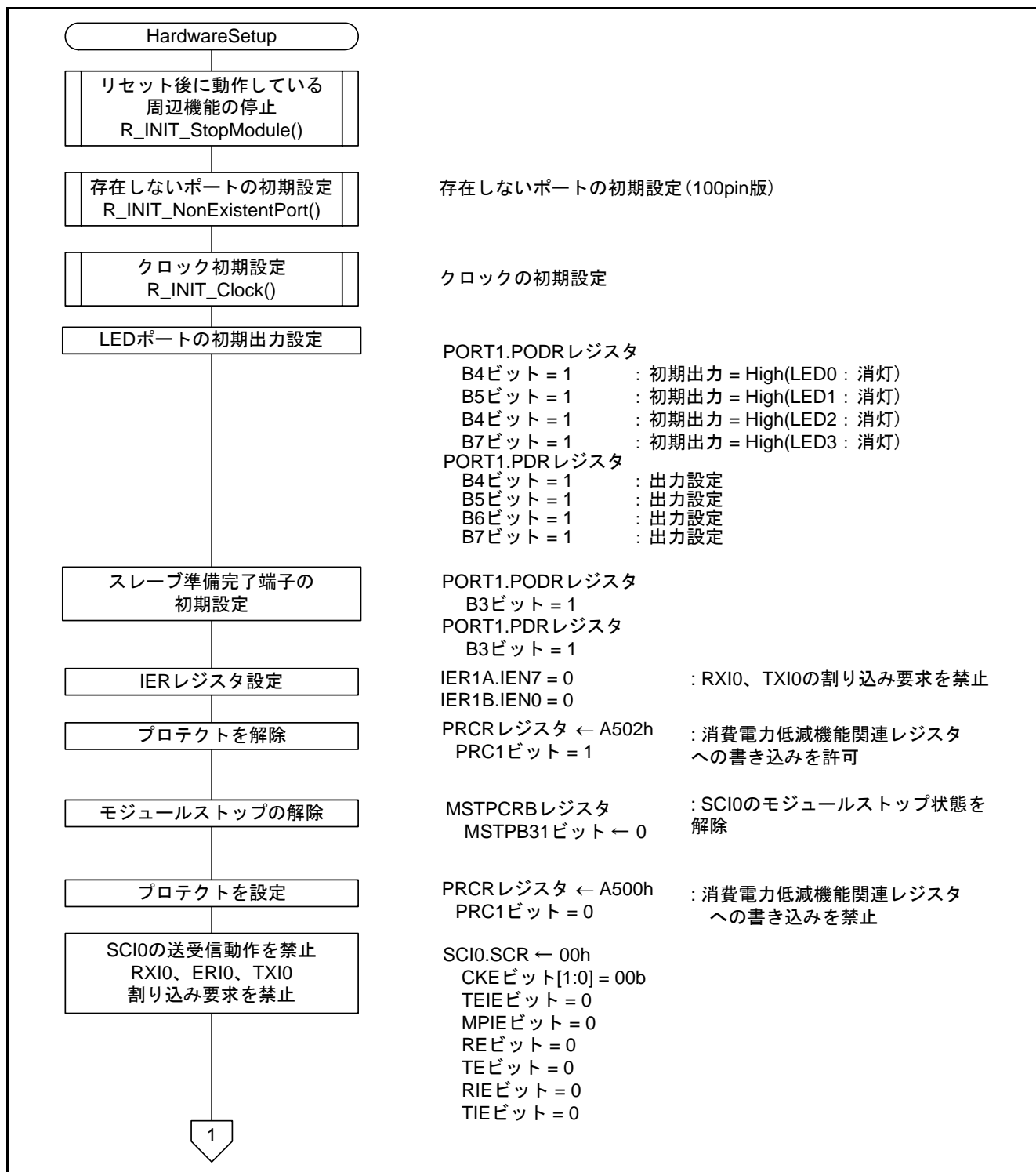


図 5.10 初期化処理(1)



図 5.11 初期化処理(2)

5.14.3 メイン関数

図 5.12にメイン関数のフローチャートを示します。

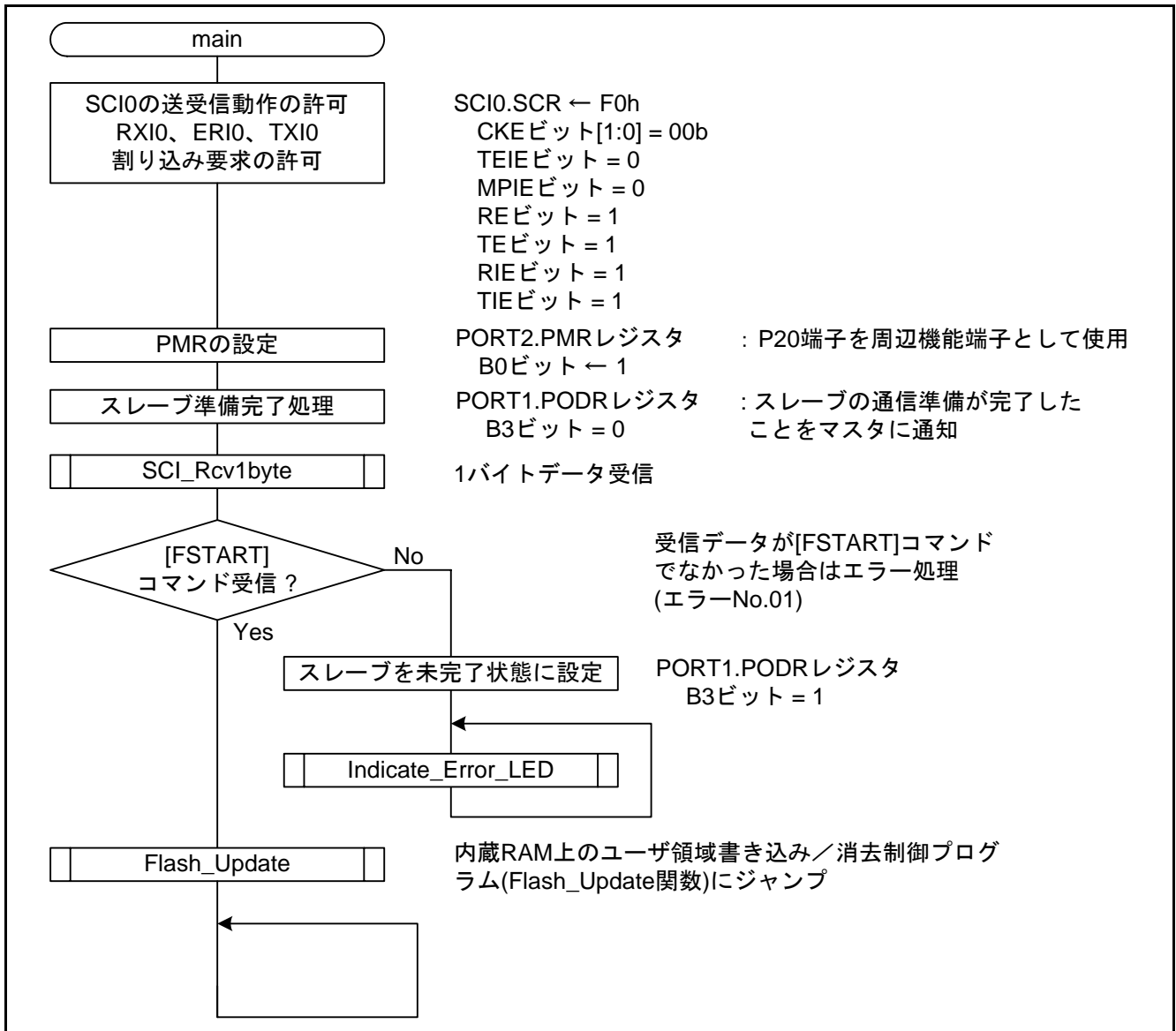


図 5.12 メイン関数

5.14.4 書き込み/消去処理

図 5.13～図 5.15に書き込み/消去処理のフローチャートを示します。

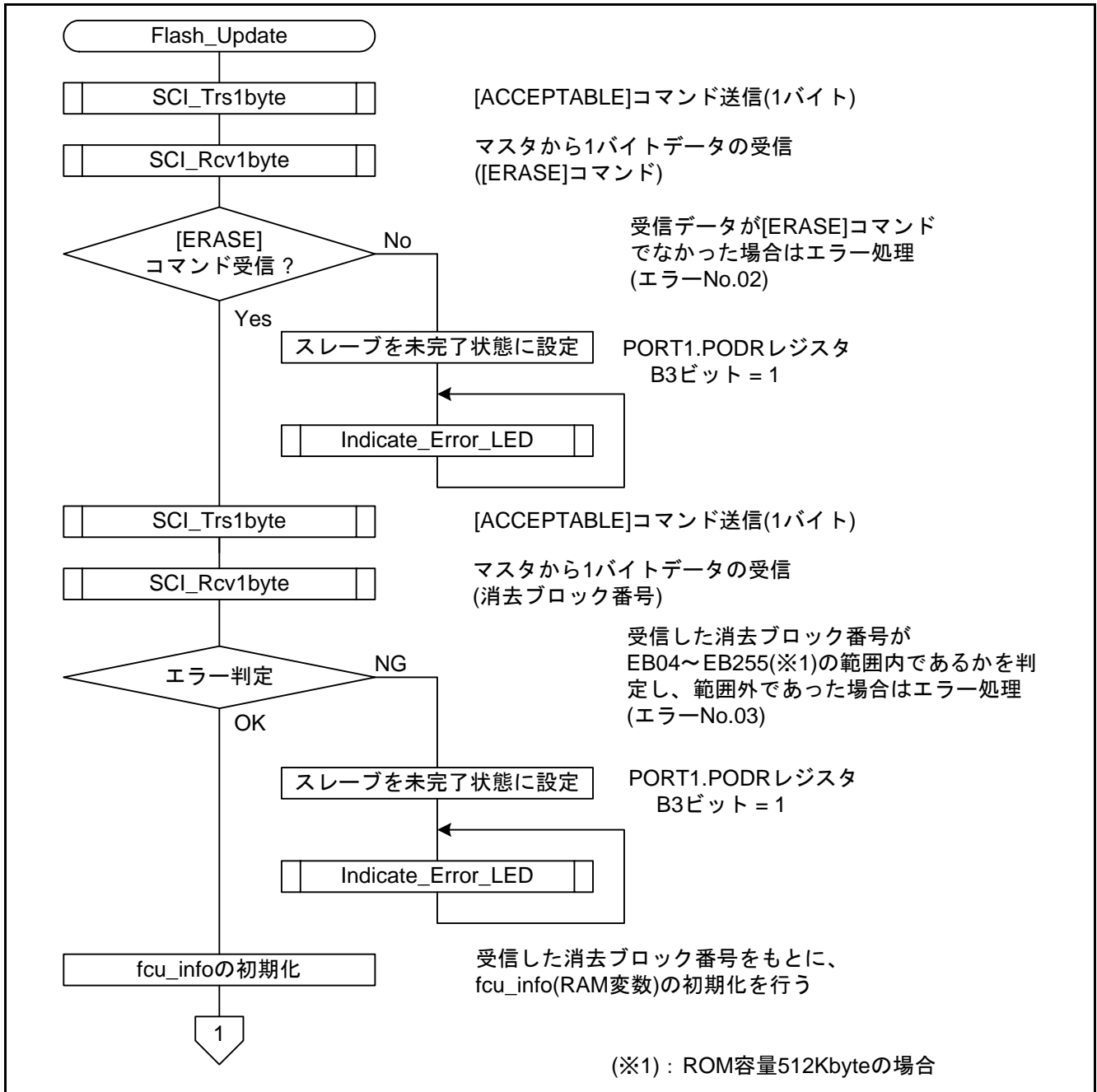


図 5.13 書き込み/消去処理(1)

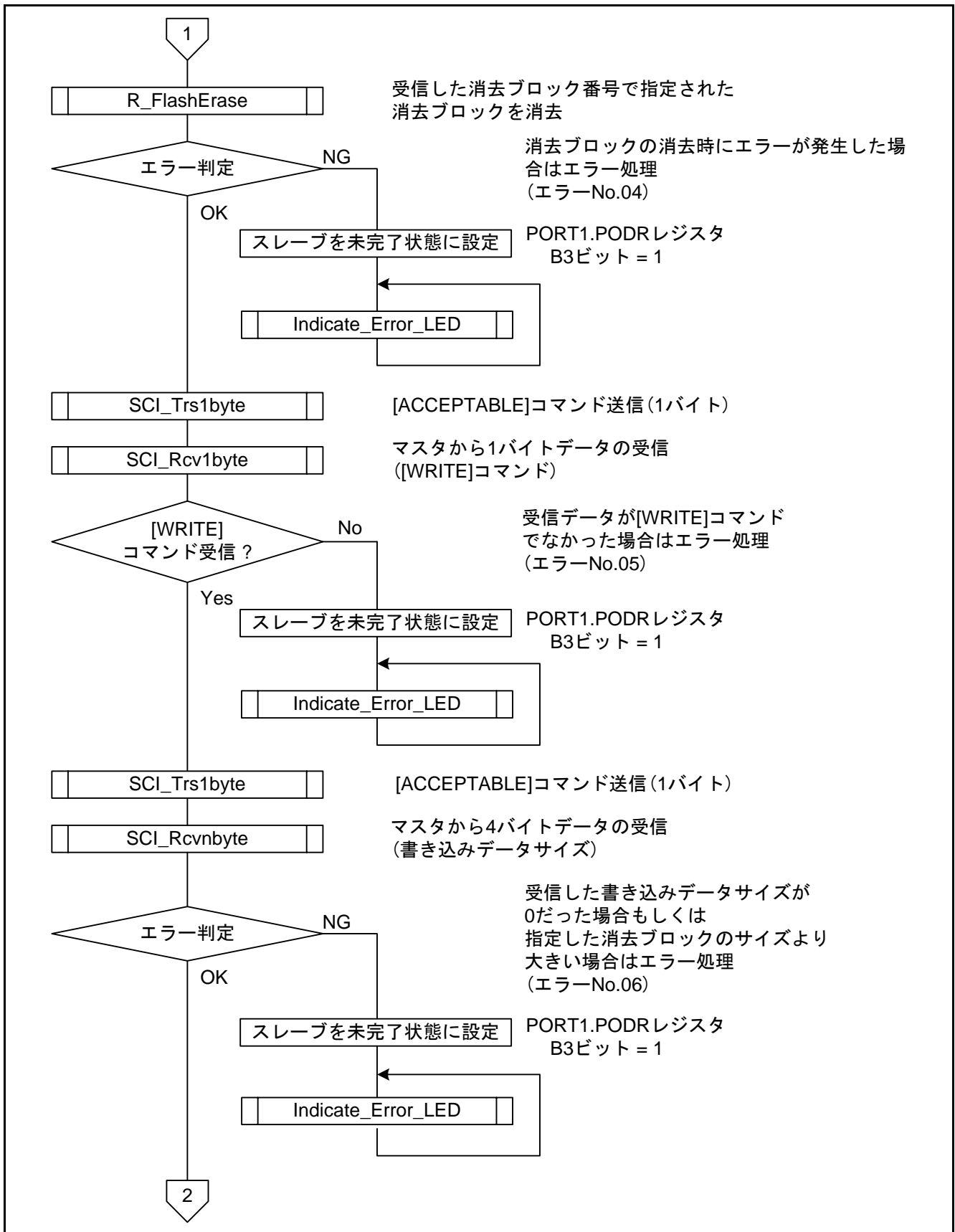


図 5.14 書き込み/消去処理(2)



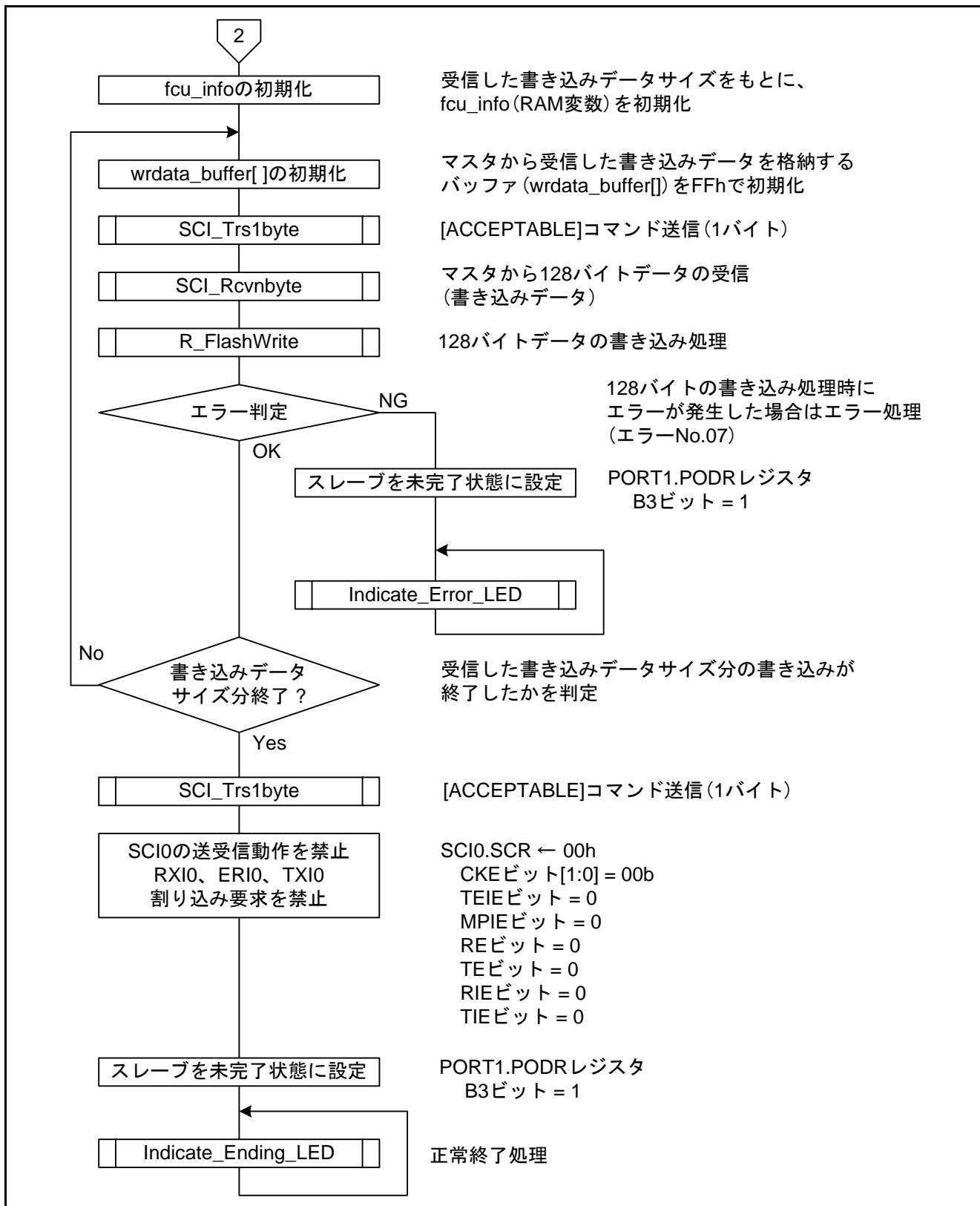


図 5.15 書き込み/消去処理(3)

## 5.14.5 正常処理関数

図 5.16に正常処理関数のフローチャートを示します。

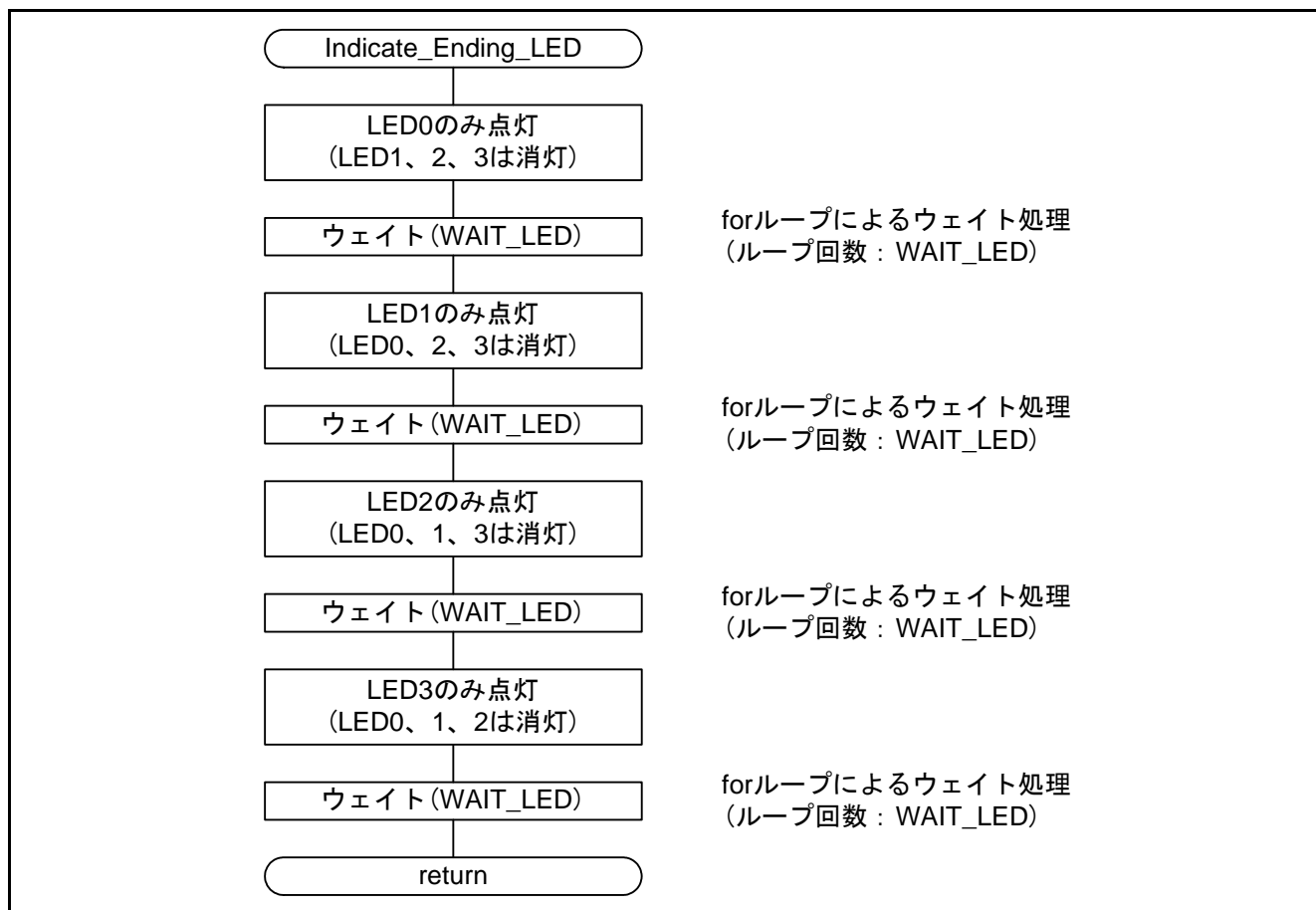


図 5.16 正常処理関数

## 5.14.6 エラー終了処理関数

図 5.17にエラー終了処理関数のフローチャートを示します。

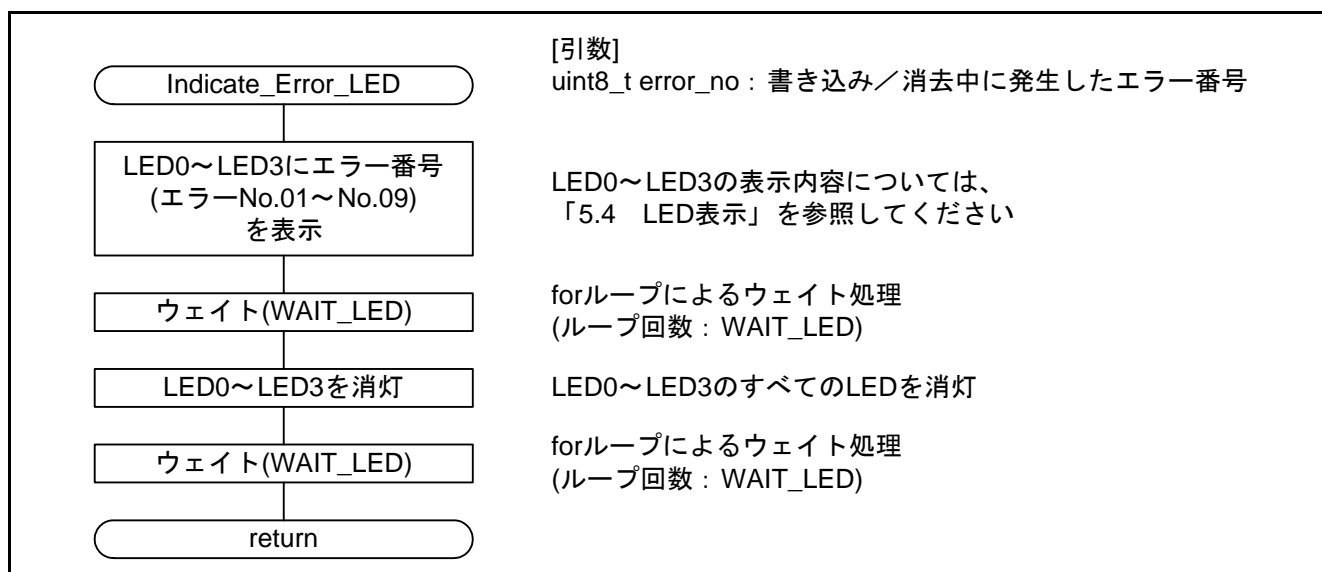


図 5.17 エラー終了処理関数

5.14.7 1バイトデータ受信関数

図 5.18に1バイトデータ受信関数のフローチャートを示します。

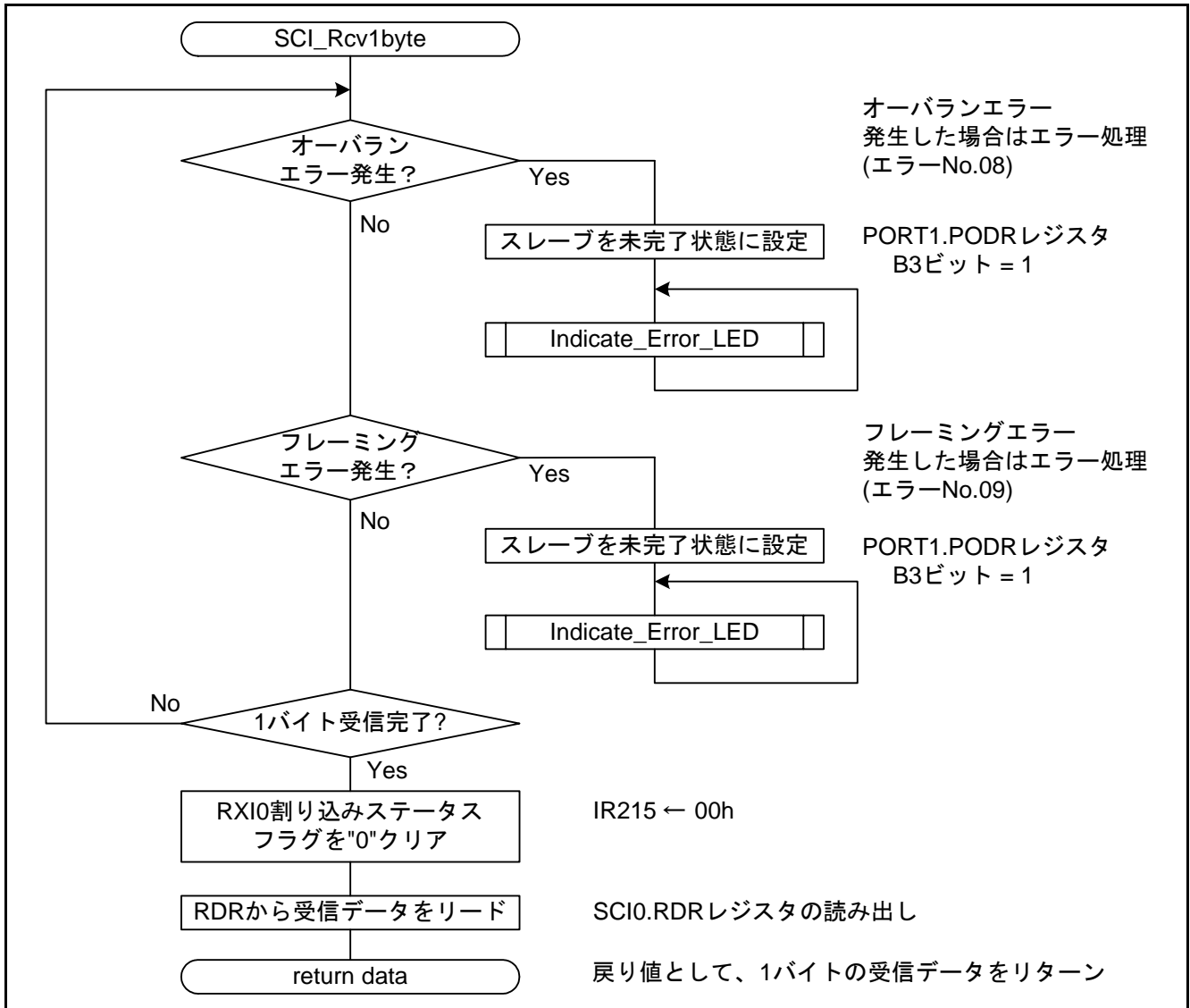


図 5.18 1バイトデータ受信関数

5.14.8 n バイトデータ受信関数

図 5.19に nバイトデータ受信関数のフローチャートを示します。

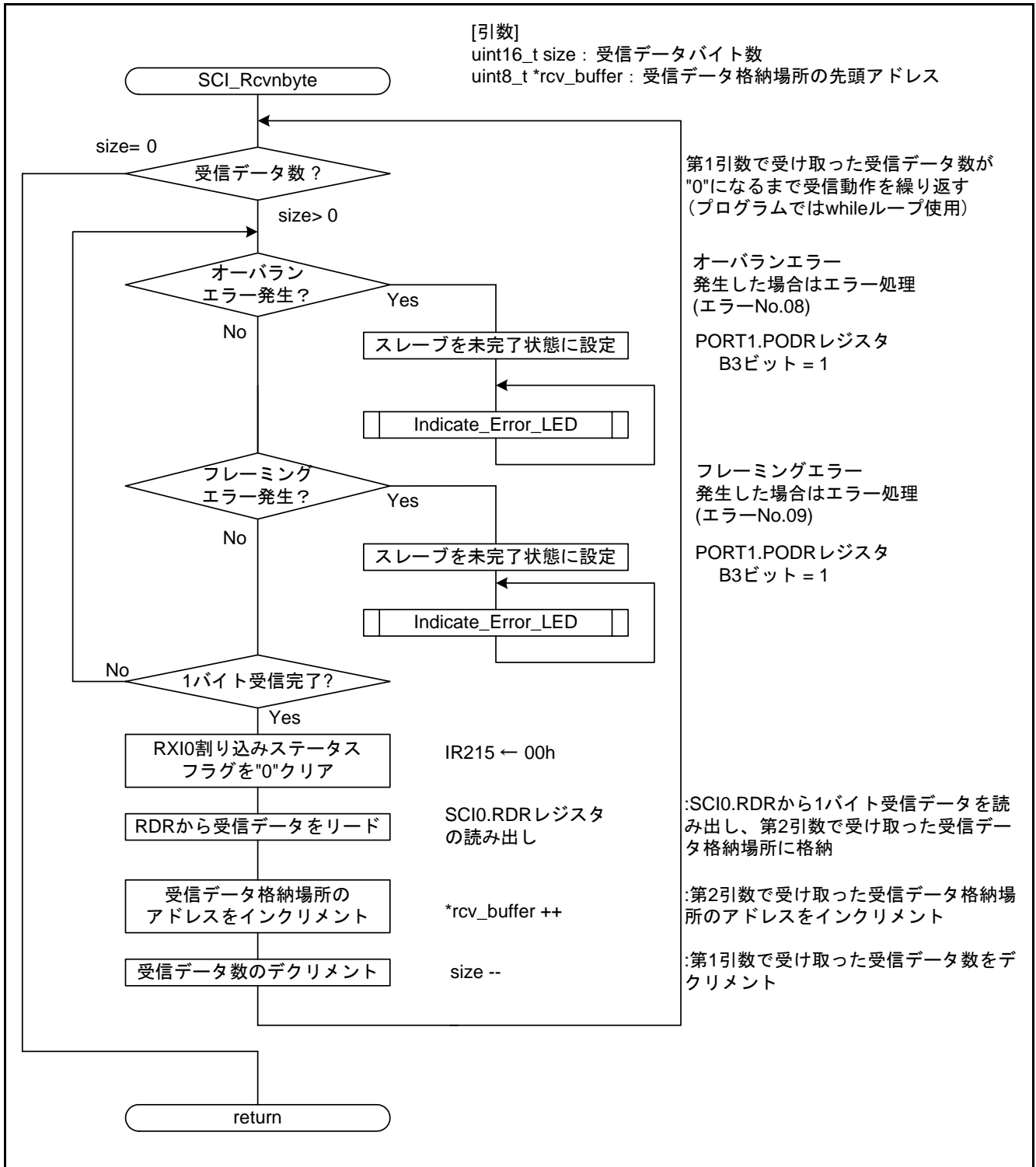


図 5.19 n バイトデータ受信関数

## 5.14.9 1バイトデータ送信関数

図 5.20に1バイトデータ送信関数のフローチャートを示します。

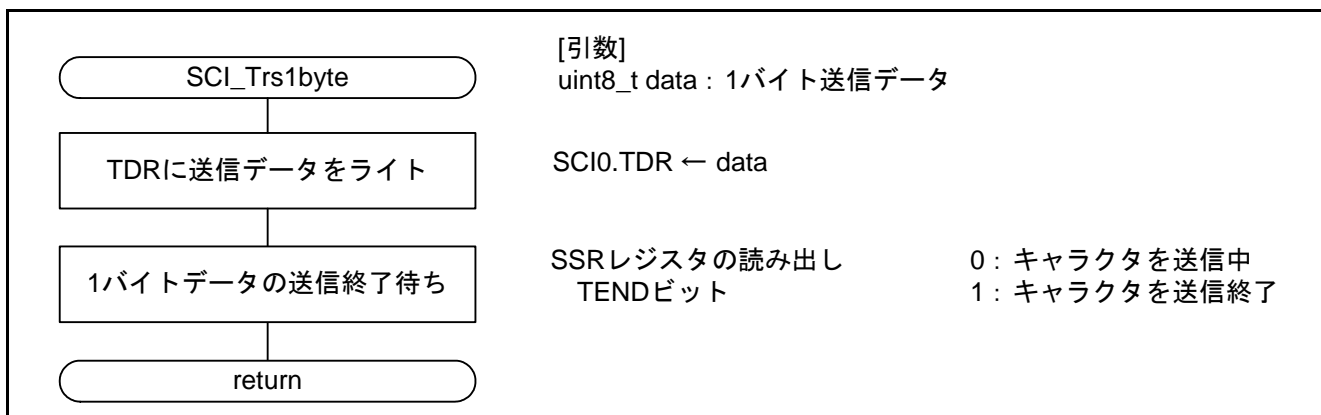


図 5.20 1バイトデータ送信関数

## 6. 注意事項

### 6.1 消去ブロック EB00 の書き換え時の注意事項

消去ブロック EB00(書き込み/消去用アドレス:00FF F800h~00FF FFFF、読み出し用アドレス:FFFF F800h~FFFF FFFFh)には、固定ベクタ(FFFF FF80h~FFFF FFFFh)、ID コードプロテクト(FFFF FFA0h~FFFF FFAFh)などが配置されています。

EB00 の書き込み/消去を行うと、上記の固定ベクタ、ID コードプロテクトは、一度消去されてしまいます。したがって、EB00 の消去後に再度、固定ベクタ、ID コードプロテクトの設定を行う必要がありますのでご注意ください。

IDコードプロテクトは、ホストからの読み出し/書き込み/消去を禁止するための機能で、ROM上に書かれている制御コードおよびIDコードを使い、IDコードプロテクトの判定を行います。IDコードプロテクトの詳細については、8. 参考ドキュメント「ユーザズマニュアル」を参照してください。

### 6.2 ROM 容量の変更

サンプルコードが使用しているマイコンのROM容量は 512Kバイトです。ROM容量が 512Kバイト、384Kバイト、256Kバイト、または 128Kバイトのマイコンを使用する場合は、ファイル“r\_Flash\_API\_RX200\_UserConfig.h”内の“ROM\_SIZE”定義を、使用する容量に合わせて変更してください。表 6.1にROM容量一覧を示します。

例：ROM 容量が 512K バイトの場合  
#define ROM\_SIZE (524288)

表 6.1 ROM 容量一覧

製品型名	ROM 容量	定義値	使用可能な消去ブロック番号
R5F52108	512K	(524288)	EB04~EB255
R5F52107	384K	(393216)	EB04~EB191
R5F52106	256K	(262144)	EB04~EB127
R5F52105	128K	(131072)	EB04~EB63

### 6.3 動作モードの設定

本アプリケーションノートでは、モード端子を MD=High に設定し動作モードをシングルチップモードに、システムコントロールレジスタ 0(SYSCR0)の ROME ビットを 1 に設定し内蔵 ROM を有効にそれぞれ設定しています。

本アプリケーションノートにおける動作モードの設定を表 6.2に示します。

表 6.2 動作モードの設定

モード設定端子	SYSCR0 レジスタ	動作モード	内蔵 ROM
MD	ROME		
High	1	シングルチップモード	有効

【注】 SYSCR0 レジスタの ROME ビットの初期値は、SYSCR0.ROME=1 のため、プログラム中で SYSCR0 レジスタの設定は行っていません。

## 6.4 エンディアンの設定

本アプリケーションノートは、ビッグエンディアン／リトルエンディアンの両方に対応しています。なお、本サンプルコードではマスタ側とスレーブ側でエンディアンを合わせてご使用ください。

### 6.4.1 リトルエンディアン使用時

リトルエンディアンで動作する場合は以下の設定を行ってください。

コンパイラオプションのエンディアンの設定で“Little-endianデータ”を指定してください。5.8 オプション設定メモリのMDESはリトルエンディアンの値になります。

### 6.4.2 ビッグエンディアン使用時

ビッグエンディアンで動作する場合は以下の設定を行ってください。

コンパイラオプションのエンディアンの設定で“Big-endianデータ”を指定してください。5.8 オプション設定メモリのMDESはビッグエンディアンの値になります。

## 7. サンプルコード

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

## 8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX210グループ ユーザーズマニュアル ハードウェア編 Rev.1.20 (R01UH0037JJ)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：開発環境

RX ファミリー C/C++コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00 (R20UT0570JJ)  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

アプリケーションノート

RX200 シリーズ RX200 用のシンプルフラッシュ API Rev.1.01  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

RX210 グループ 初期設定例 Rev.2.00

(最新版をルネサス エレクトロニクスホームページから入手してください。)

## 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/>



改訂記録	RX210 グループ アプリケーションノート シングルチップモード による UART 経由での内蔵フラッシュメモリ書き換え
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.04.01	—	初版発行

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。

## 製品ご使用上の注意事項

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

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

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

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

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

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

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

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

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

### 5. 製品間の相違について

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

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

## ご注意書き

1. 本資料に記載された回路、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らず、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、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て、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事情報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・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町 2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>