

RX630 グループ

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（スレーブ）

R01AN1251JJ0101
Rev.1.01
2014.12.04

要旨

本アプリケーションノートは、RX630 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（マスタ）」（R01AN1271JJ）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（ユーザマット）の書き込み/消去する処理について説明しています。

調歩同期式シリアル通信で消去ブロック番号、書き込みデータのサイズ、および書き込みデータを送信する処理に関しては、RX630 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（マスタ）」（R01AN1271JJ）をご参考ください。

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。

RX630 の初期設定：

「RX630 グループ 初期設定例」 Rev.1.00 (R01AN1004JJ0100)

内蔵フラッシュメモリの消去 / 書き込み：

「RX600 シリーズ RX600 用のシンプルフラッシュ API」 Rev.2.20 (R01AN0544JU0220)

対象デバイス

- | | | |
|-------------|-------------|--------------------|
| • RX630グループ | 177、176 ピン版 | ROM 容量：768KB ~ 2MB |
| • RX630グループ | 145、144 ピン版 | ROM 容量：768KB ~ 2MB |
| • RX630グループ | 100 ピン版 | ROM 容量：384KB ~ 2MB |
| • RX630グループ | 80 ピン版 | ROM 容量：384KB、512KB |

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	3
2. 動作確認条件	5
3. 関連アプリケーションノート	5
4. ハードウェア説明	6
5. ソフトウェア説明	8
6. 注意事項	39
7. サンプルコード	41
8. 参考ドキュメント	41

1. 仕様

本アプリケーションノートでは、RX630 グループの R5F5630EDDFP を用いて、シングルチップモードでユーザマットの書き込み / 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、ユーザマットの書き込み / 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 P15 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 と EB01 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 や EB01 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、正常にユーザマットの消去 / 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、P15 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 に仕様を示します。

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

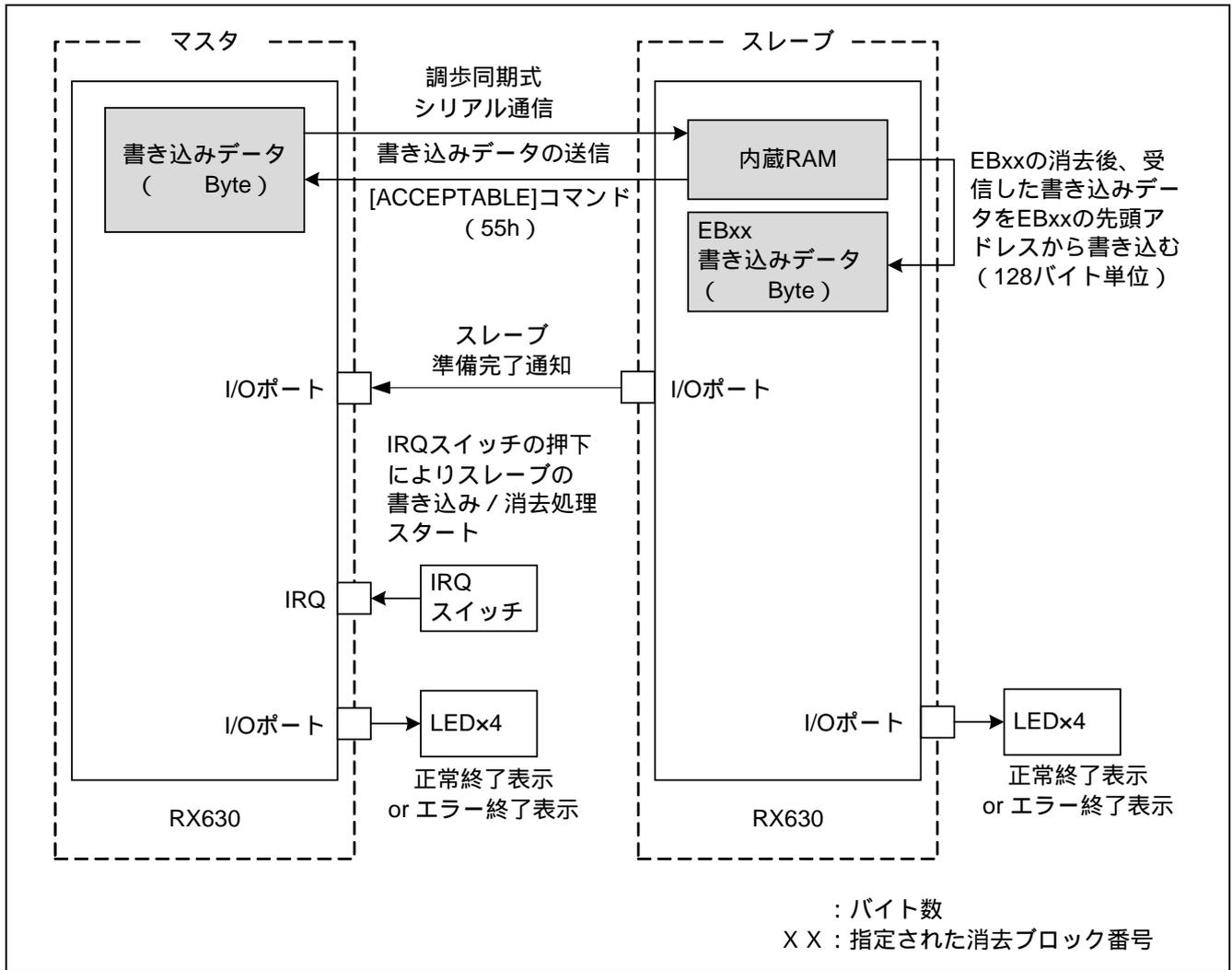


図1.1 仕様

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	R5F5630EDDFP (RX630 グループ)
動作周波数	<ul style="list-style-type: none"> メインクロック: 12.0MHz PLL: 192MHz (メインクロック 1 分周 16 連倍) システムクロック (ICLK): 96MHz (PLL 2 分周) 周辺モジュールクロック B (PCLKB): 48MHz (PLL 4 分周) USB に供給される USB クロック (UCLK): 48MHz (PLL4 分周) FlashIF クロック (FCLK): 48MHz (PLL4 分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.00.007
C コンパイラ	ルネサスエレクトロニクス製 RX Standard Toolchain Version 1.2.1.0 コンパイルオプション -cpu=rx600 -output=obj="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo
iodefine.h のバージョン	1.0A
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
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 RX630 (製品型名: R0K505630C000BE)
最適化リンケージエディタ (rom オプション)*	-rom=D=R,D_1=R_1,D_2=R_2,PFRAM=RPFRAM

【注】 * 「6.5 rom オプションについて」を参照してください。

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。

- RX600 シリーズ RX600 用のシンプルフラッシュ API Rev.2.20 (R01AN0544JU)
- RX630 グループ 初期設定例 Rev.1.00 (R01AN1004JJ)

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。

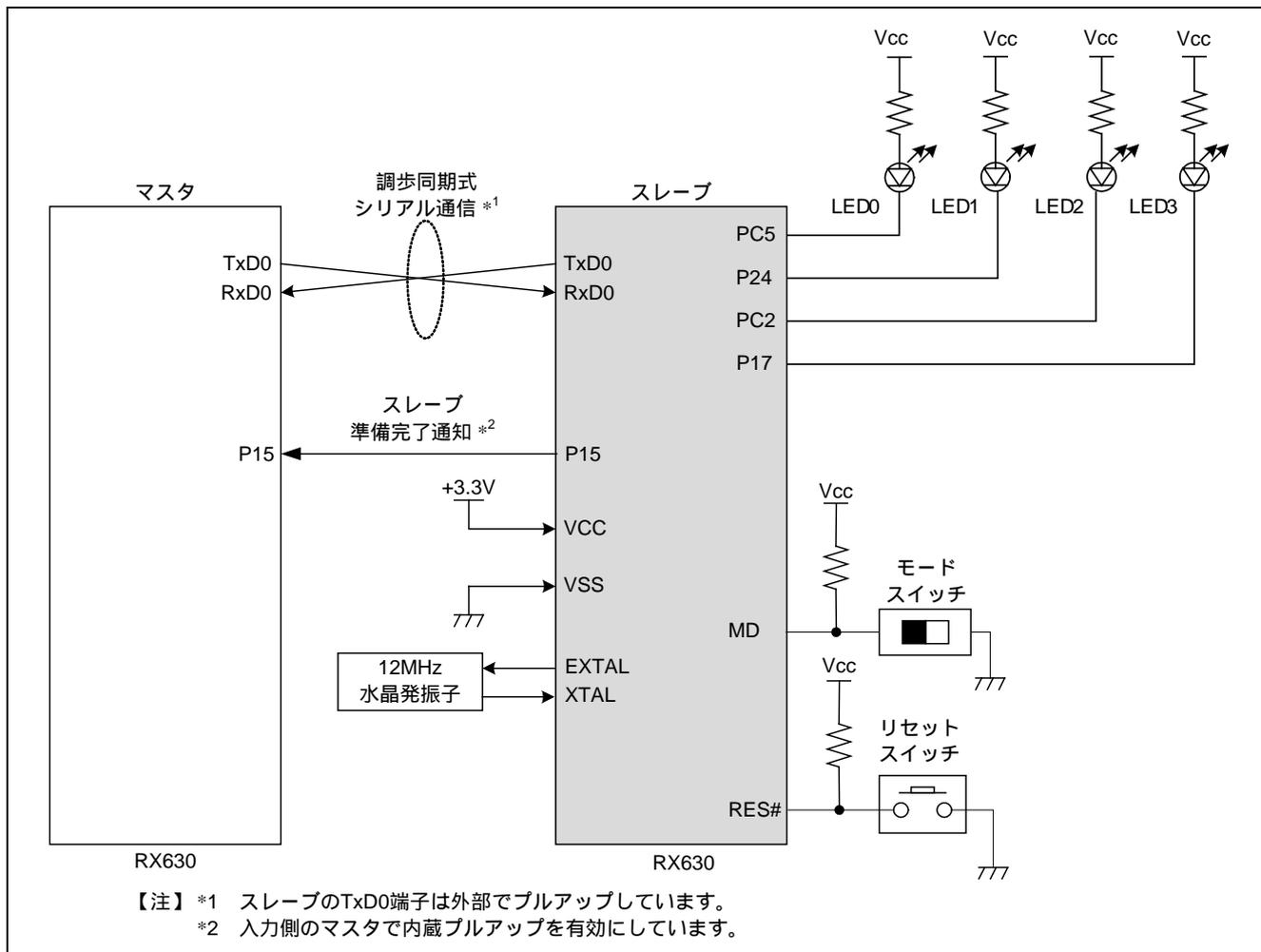


図4.1 スレーブのハードウェア構成図

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
P15	出力	スレーブ準備完了通知用端子
PC5	出力	LED0 接続端子 (“High”出力：消灯、“Low”出力：点灯)
P24	出力	LED1 接続端子 (“High”出力：消灯、“Low”出力：点灯)
PC2	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)
コミュニケーションモード	調歩同期式モード
ビットレート	31250bps (PCLKB=48MHz 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	スレーブのユーザマットの書き込み / 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	スレーブのユーザマットの消去を開始するためのコマンド	マスタ → スレーブ
WRITE	12h	スレーブのユーザマット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	スレーブがマスタへ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 5.4に示します。

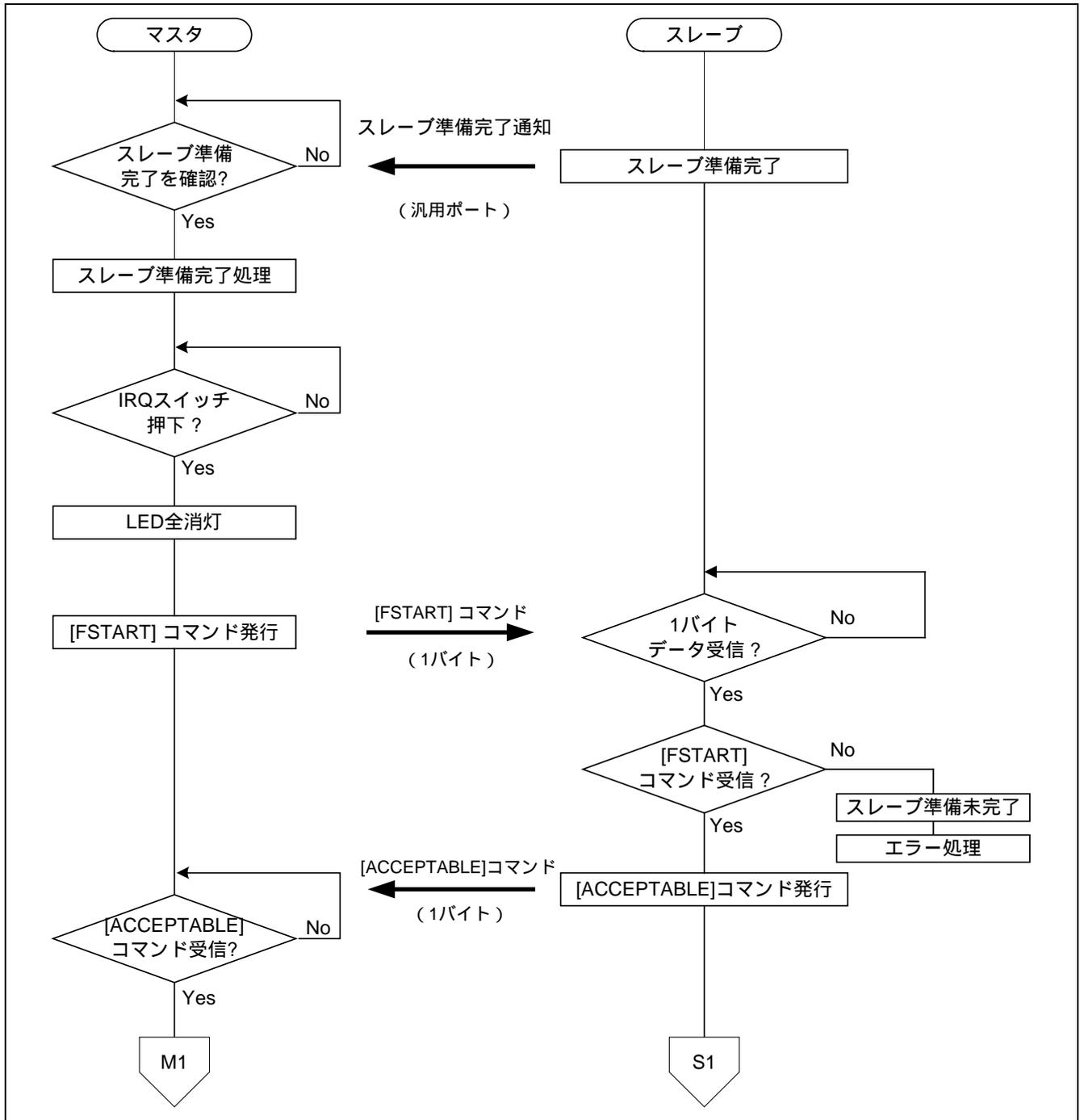


図5.1 通信フロー (1)

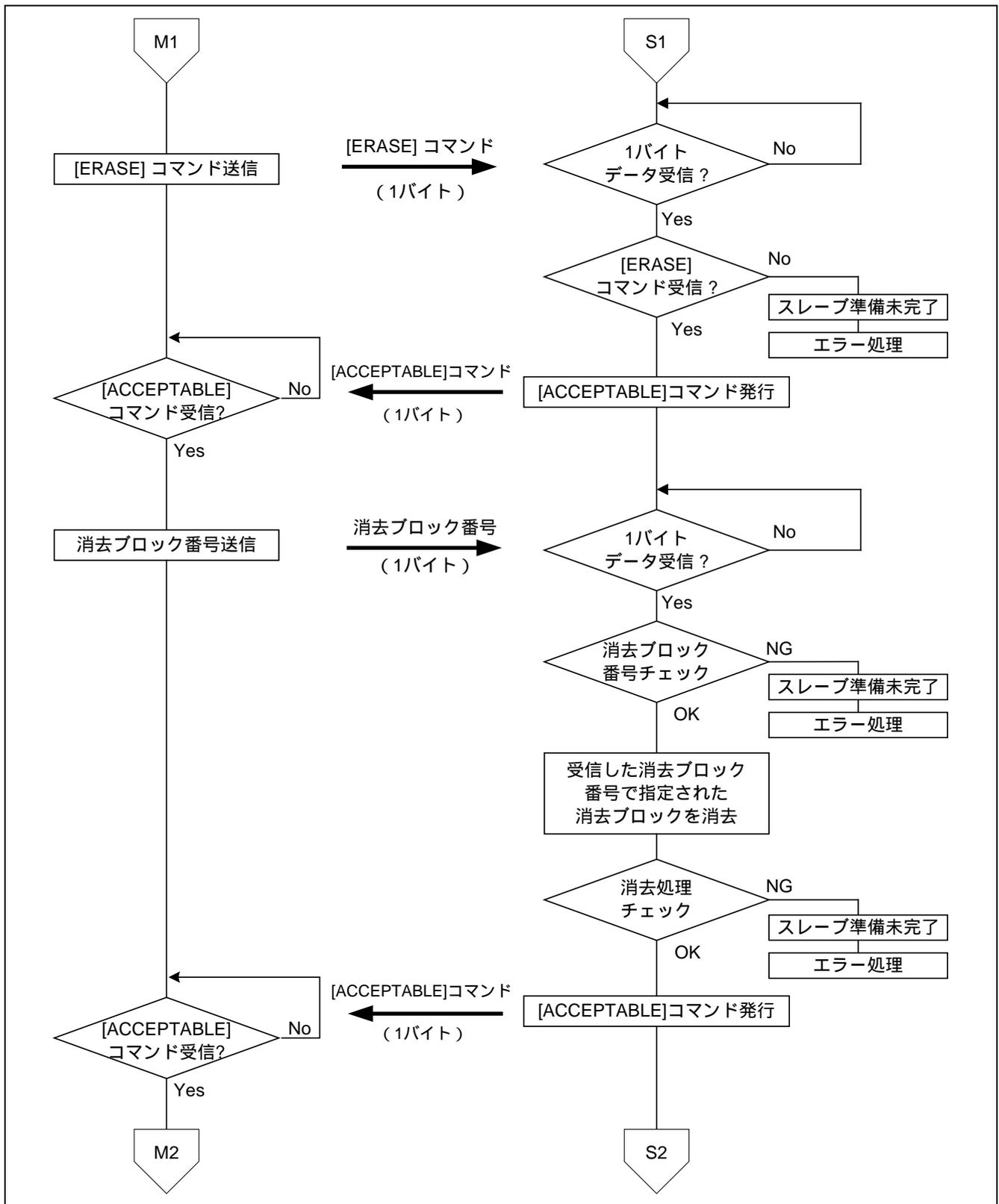


図5.2 通信フロー (2)

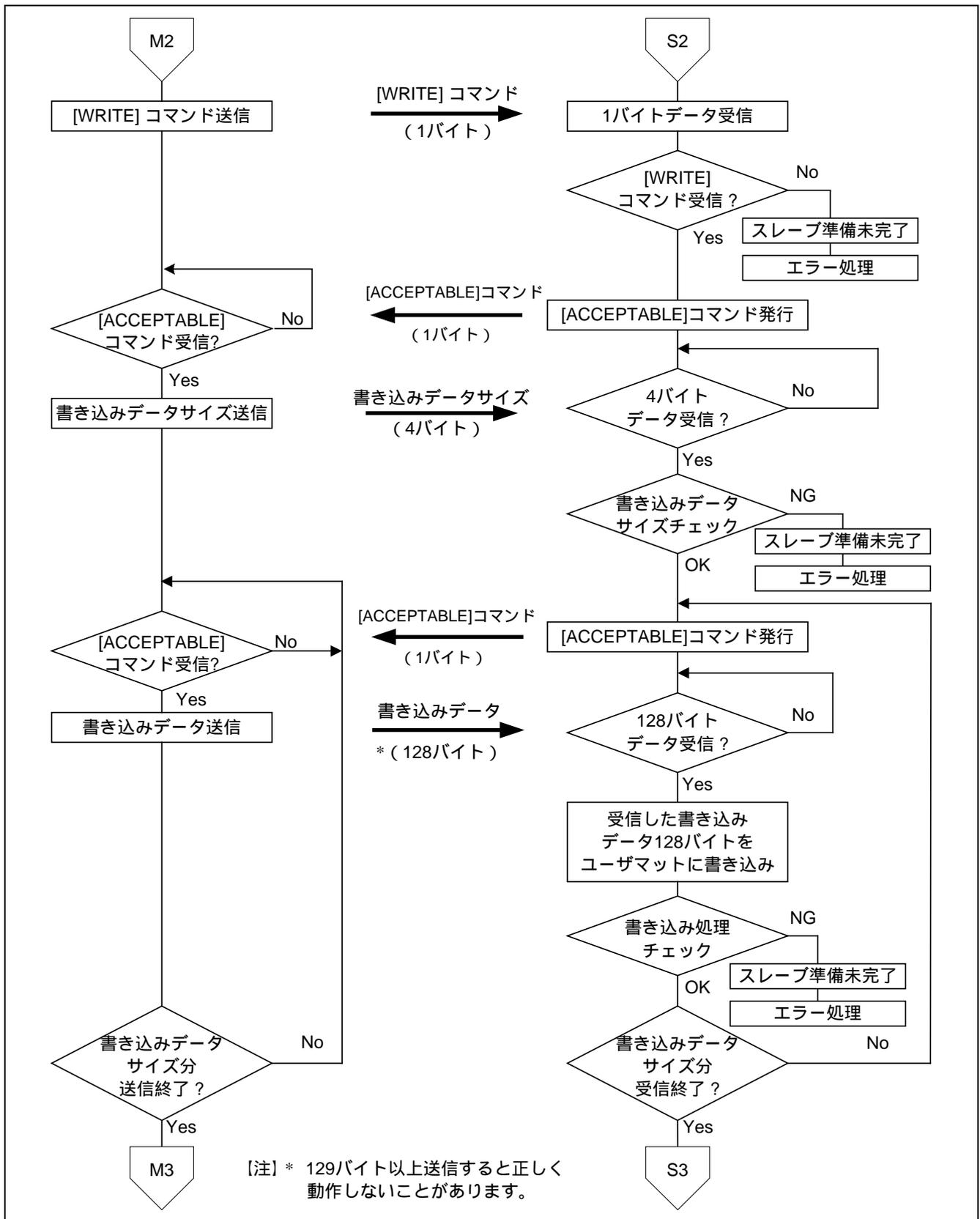


図5.3 通信フロー (3)

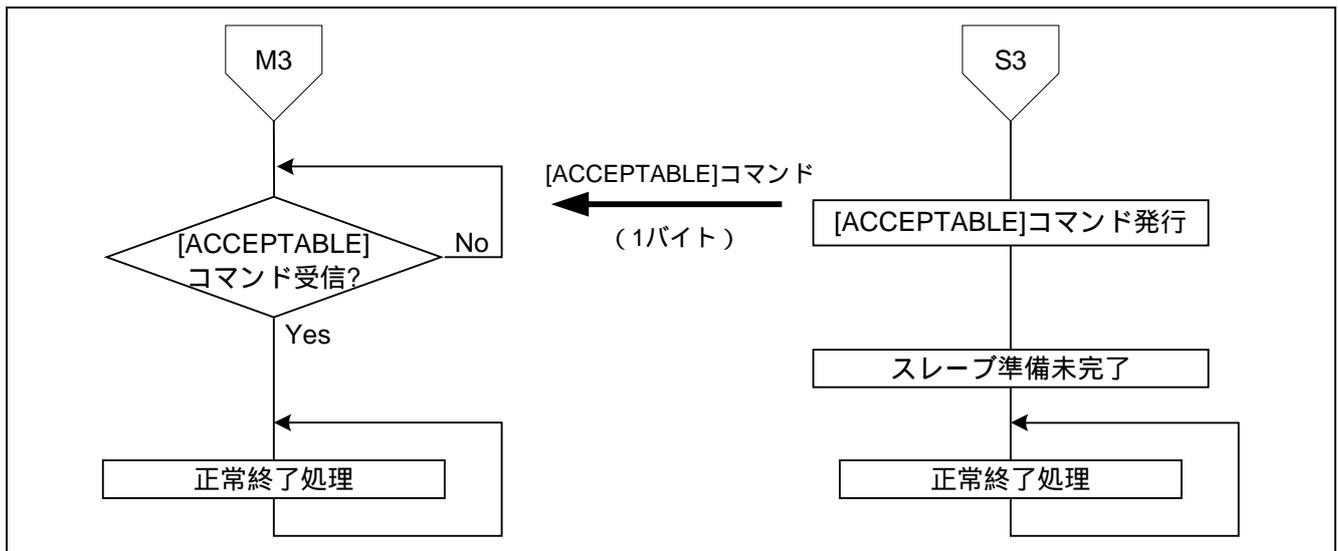


図5.4 通信フロー (4)

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 5.5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 ユーザマットの消去処理」を参照してください。

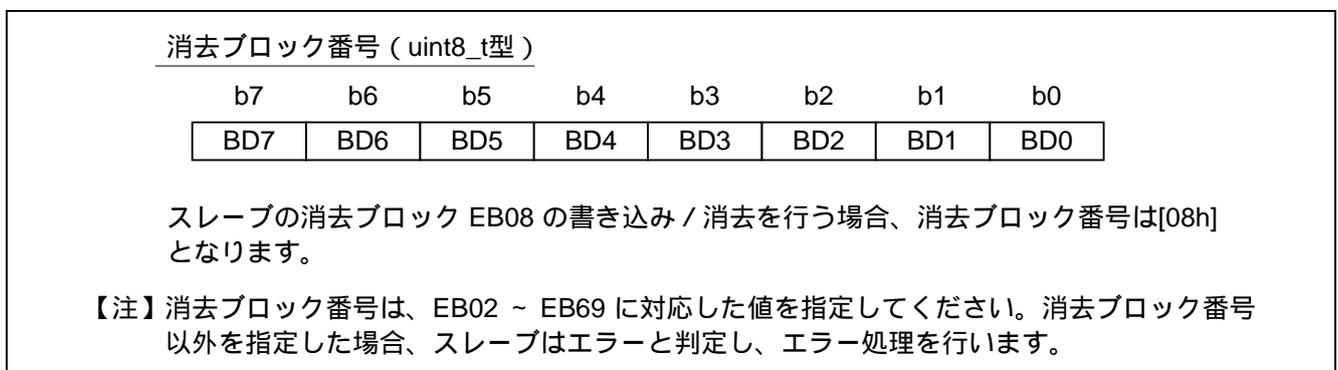


図5.5 消去ブロック番号仕様

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 5.6 に書き込みデータサイズの仕様を示します。

書き込みデータサイズ (uint32_t型)

b31	b30	b29	b28	b27	b26	b25	b24
SZ31	SZ30	SZ29	SZ28	SZ27	SZ26	SZ25	SZ24
b23	b22	b21	b20	b19	b18	b17	b16
SZ23	SZ22	SZ21	SZ20	SZ19	SZ18	SZ17	SZ16
b15	b14	b13	b12	b11	b10	b9	b8
SZ15	SZ14	SZ13	SZ12	SZ11	SZ10	SZ09	SZ08
b7	b6	b5	b4	b3	b2	b1	b0
SZ07	SZ06	SZ05	SZ04	SZ03	SZ02	SZ01	SZ00

書き込みサイズを8Kバイトとした場合、書き込みデータサイズは [0000 2000h] となります。

- 【注】
1. 書き込みデータサイズは、0より大きい値かつ消去ブロック番号で指定した消去ブロックサイズ以下としてください。0の場合もしくは消去ブロック番号で指定した消去ブロックより大きいサイズを指定した場合、スレーブはエラーと判定し、エラー処理を行います。
 2. 書き込みデータの受信は128バイト固定とし、128バイトごとに書き込みデータの受信を行います。書き込みデータサイズが128バイトの倍数でない場合、最後の128バイトに満たない書き込みデータに関してはFFhを追加して、スレーブのユーザマットに書き込みます。

図5.6 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 ユーザマットの書き込み / 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「RX600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」の「RX600用のシンプルフラッシュ API」を参照してください。

5.2.1 ユーザマットの消去処理

ユーザマットの消去処理は「RX600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 ユーザマットの書き込み処理

ユーザマットの書き込みは「RX600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_rx600_config.h”を変更しています。

表 5.3に“r_flash_api_rx600_config.h”の変更点を示します。

表5.3 「r_flash_api_rx600_config.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 IGNORE_LOCK_BITS #define COPY_CODE_BY_API #define FLASH_API_USE_R_BSP </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ませんので、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 5.7 の赤線で囲まれた ~ の処理内容を示します。

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。

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」の「RX600 用のシンプルフラッシュ API」を参照してください。

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;

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

/* 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((uint8_t)target_eb);

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

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

図5.7 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 P15 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。

P15 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、ユーザマットの書き込み / 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 5.8 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5.8 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	
SI		割り込みスタック領域	
PRResetPRG		FFFF E000h	プログラム領域 (PowerON_Reset_PC プログラム)
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 F400h	プログラム領域 (ユーザマットの書き込み / 制御プログラム)
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
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_rx600.c	RX600 シリーズ RX600 用のシンプルフラッシュ API のプログラム	詳細は RX600 シリーズ RX600 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_rx600.h	RX600 シリーズ RX600 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_rx600_private.h		
r_flash_api_rx600_config.h	RX600 シリーズ RX600 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	
mcu_info.h		

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、電圧監視リセット 0 無効 リセット後、HOCO 発振が無効
MDES *	FFFF FF83h - FFFF FF80h	FFFF FFFFh FFFF FFF8h	(シングルチップモード時) リトルエンディアン ビッグエンディアン

【注】 * 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 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	PORTC.PODR.BIT.B5	評価ボード搭載 LED0 の点灯 / 消灯制御
RSK_LED1	PORT2.PODR.BIT.B4	評価ボード搭載 LED1 の点灯 / 消灯制御
RSK_LED2	PORTC.PODR.BIT.B2	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	PORTC.PDR.BIT.B5	評価ボード搭載 LED0 の入出力制御
RSK_LED1_PDR	PORT2.PDR.BIT.B4	評価ボード搭載 LED1 の入出力制御
RSK_LED2_PDR	PORTC.PDR.BIT.B2	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.B5	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.B5	スレーブ準備完了の確認用ポートの入出力制御
WAIT_LED	2000000	スレーブのユーザマットの書き込み / 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	128 *	対象デバイスに応じた、ユーザマットへの書き込み単位が設定されます。 r_flash_api_rx600_private.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
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	

【注】 * RX630 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
WRITE_ADRS_TOP_64K	0x00E00000	書き込み / 消去用アドレス空間におけるブロックサイズ 64K バイト領域の先頭アドレス
WRITE_ADRS_TOP_32K	0x00F00000	書き込み / 消去用アドレス空間におけるブロックサイズ 32K バイト領域の先頭アドレス
WRITE_ADRS_TOP_16K	0x00F80000	書き込み / 消去用アドレス空間におけるブロックサイズ 16K バイト領域の先頭アドレス
WRITE_ADRS_TOP_4K	0x00FF8000	書き込み / 消去用アドレス空間におけるブロックサイズ 4K バイト領域の先頭アドレス
BLK_SIZE_64K	64×1024	EB54 ~ EB69 の各ブロックサイズ
BLK_SIZE_32K	32×1024	EB38 ~ EB53 の各ブロックサイズ
BLK_SIZE_16K	16×1024	EB08 ~ EB37 の各ブロックサイズ
BLK_SIZE_4K	4×1024	EB00 ~ EB07 の各ブロックサイズ

5.10 構造体 / 共用体一覧

図 5.9 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5.9 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_PROGRA M_SIZE]	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	ユーザマットの書き込み / 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	ユーザマットの書き込み/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
mpc_init	MPC 初期設定関数
pmr_init	PMR 初期設定関数

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_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"> 存在しないポートの初期設定 (100pin 版) クロック設定 (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) LED0~LED3 を接続している I/O ポート (PC5、P24、PC2、および P17) の初期出力設定 モジュールストップ状態解除 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 SCI0 の初期設定
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「RX630グループ 初期設定例 Rev.1.00」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	176 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「RX630グループ 初期設定例 Rev.1.00」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「RX630グループ 初期設定例 Rev.1.00」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、P15 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 上のユーザマツト書き込み / 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	ユーザマツトの書き込み / 消去処理関数
ヘッダ	iodefine.h、mcu_info.h、r_flash_api_rx600.h、r_flash_api_rx600_private.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、ユーザマツトの書き込み / 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"> マスタから FSTART の通信コマンドを受信し、処理を開始 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 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 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、ユーザマツトにデータを書き込む ユーザマツトの書き込み / 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
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、ユーザマットの書き込み / 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 : ユーザマットの書き込み / 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	iodefine.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	iodefine.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

mpc_init	
概要	MPC 初期設定
ヘッダ	iodefine.h
宣言	void mpc_init(void)
説明	MPC を下記の機能に選択します。 <ul style="list-style-type: none">• P20 → TXD0• P21 → RXD0
引数	なし
リターン値	なし

pmr_init	
概要	PMR 初期設定
ヘッダ	iodefine.h
宣言	void pmr_init(void)
説明	PMR の初期設定を行います。 <ul style="list-style-type: none">• P20、P21 を周辺機能として使用
引数	なし
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 5.10に初期設定関数のフローチャートを示します。

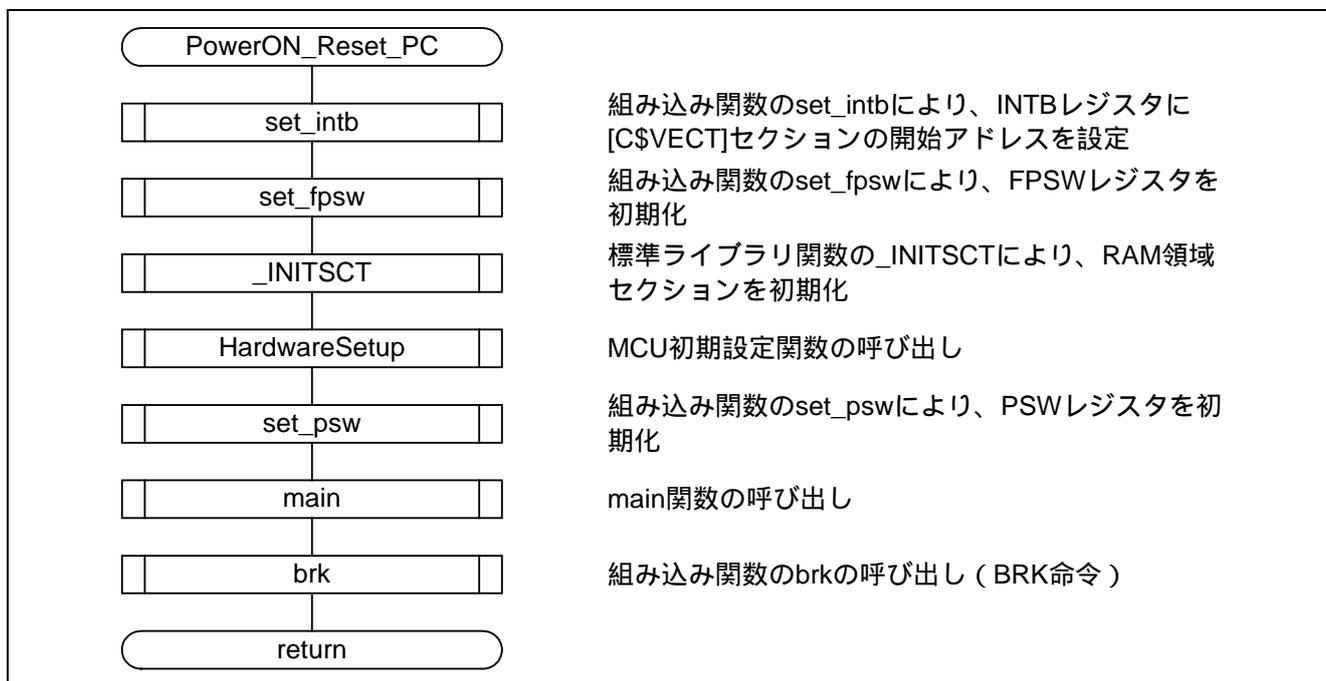


図5.10 初期設定関数

5.14.2 初期化処理

図 5.11に初期化処理のフローチャートを示します。

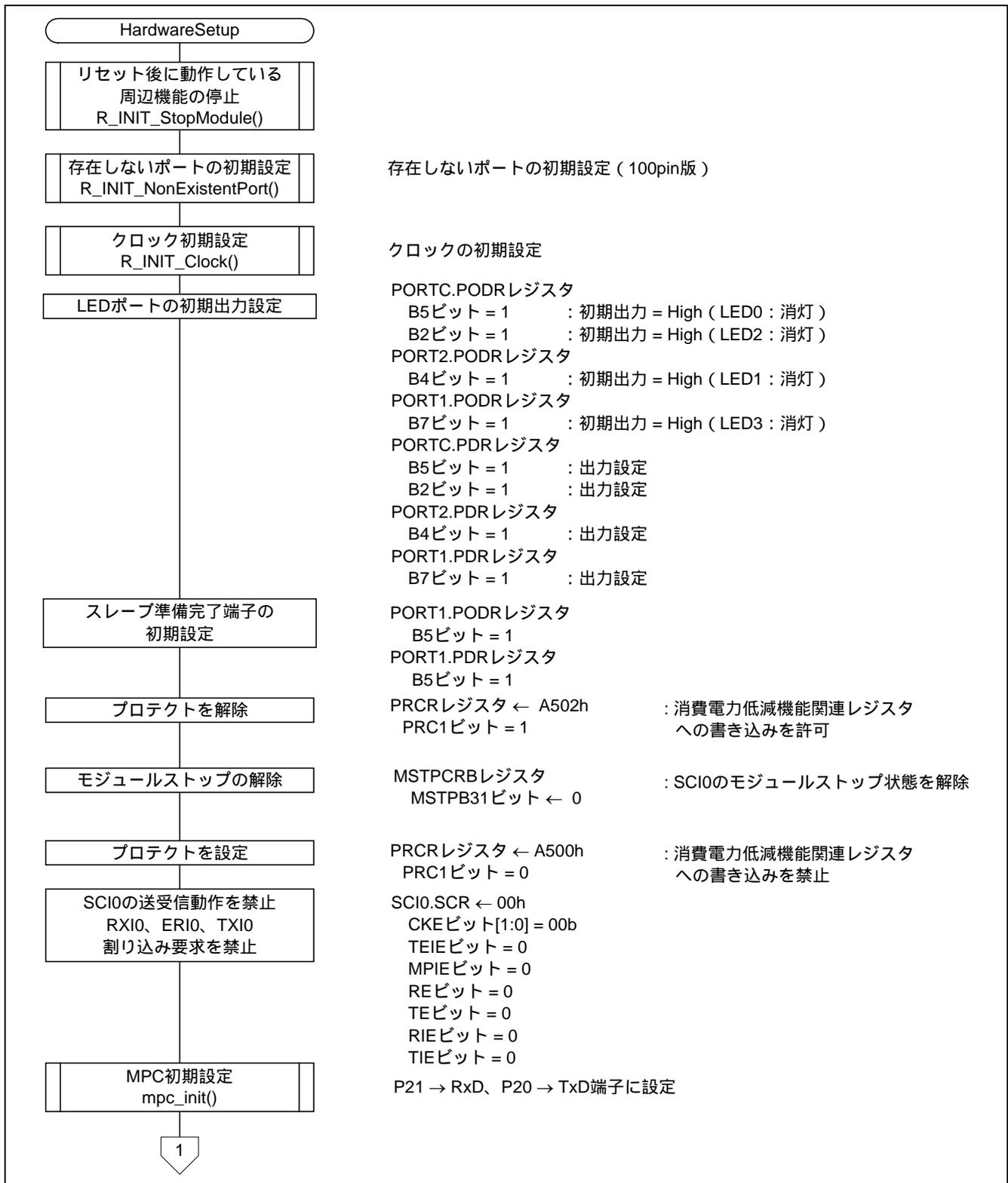


図5.11 初期化処理 (1)

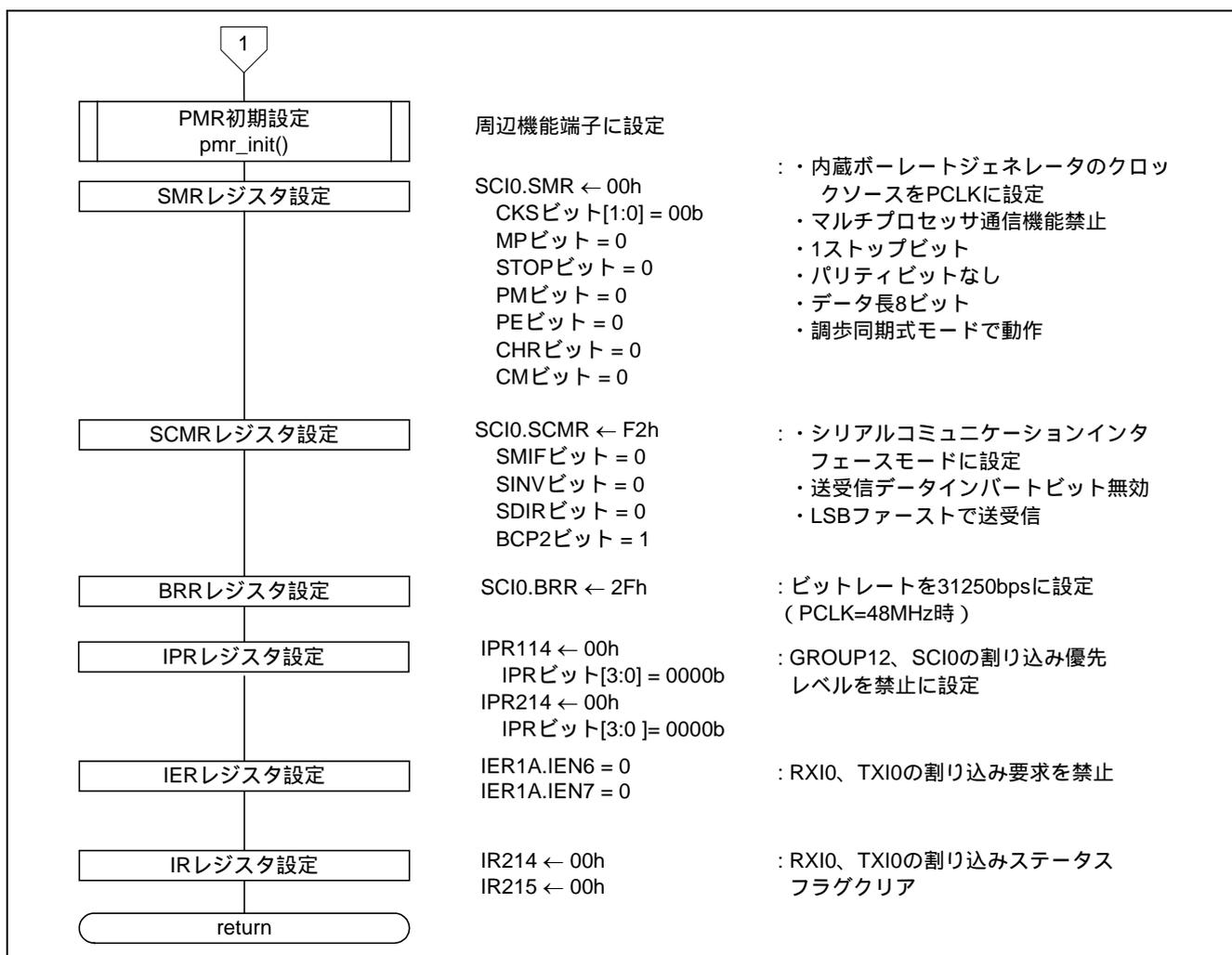


図5.12 初期化处理 (2)

5.14.3 メイン関数

図 5.13にメイン関数のフローチャートを示します。

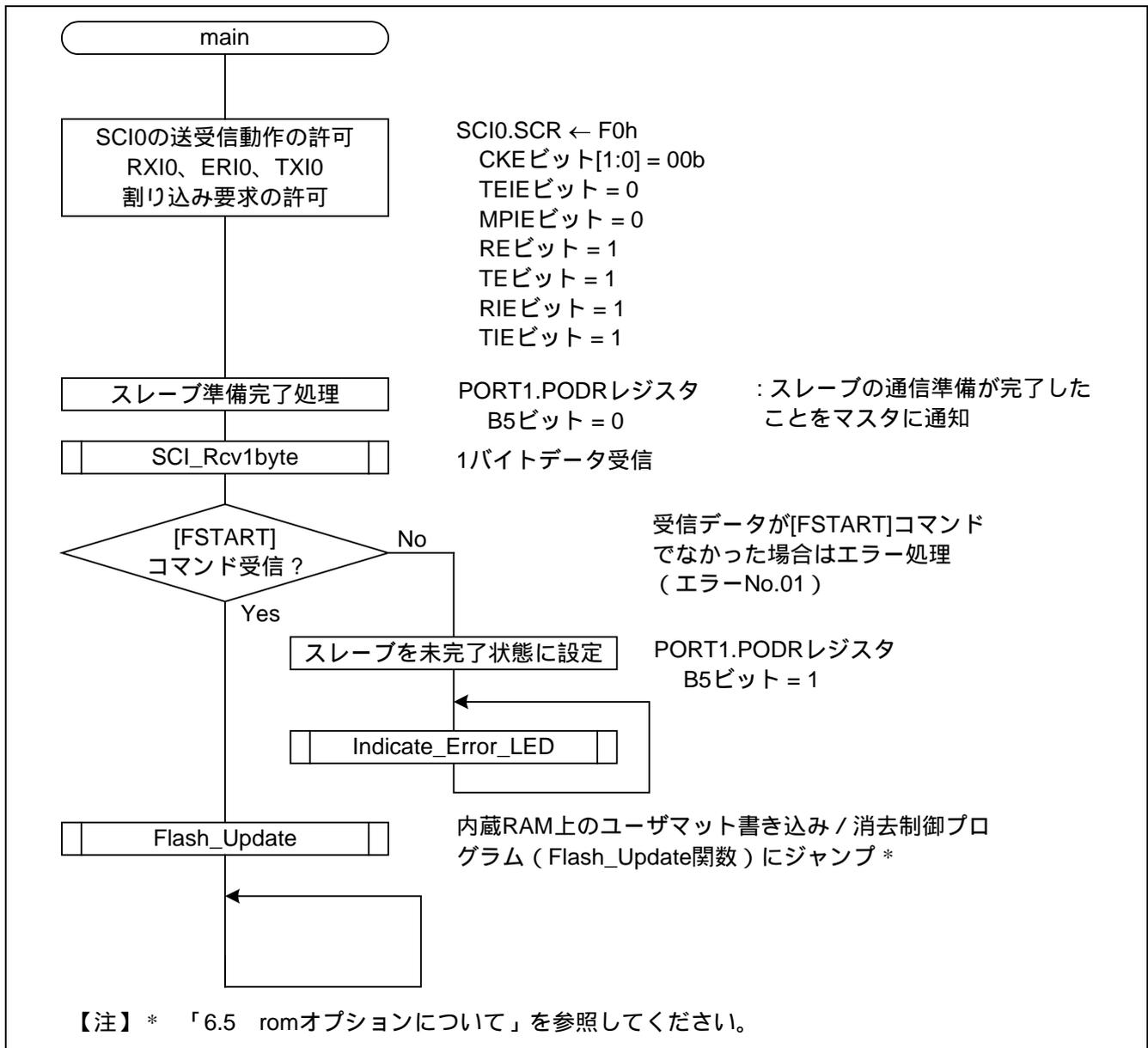


図5.13 メイン関数

5.14.4 書き込み/消去処理

図 5.14 ~ 図 5.16 に書き込み/消去処理のフローチャートを示します。

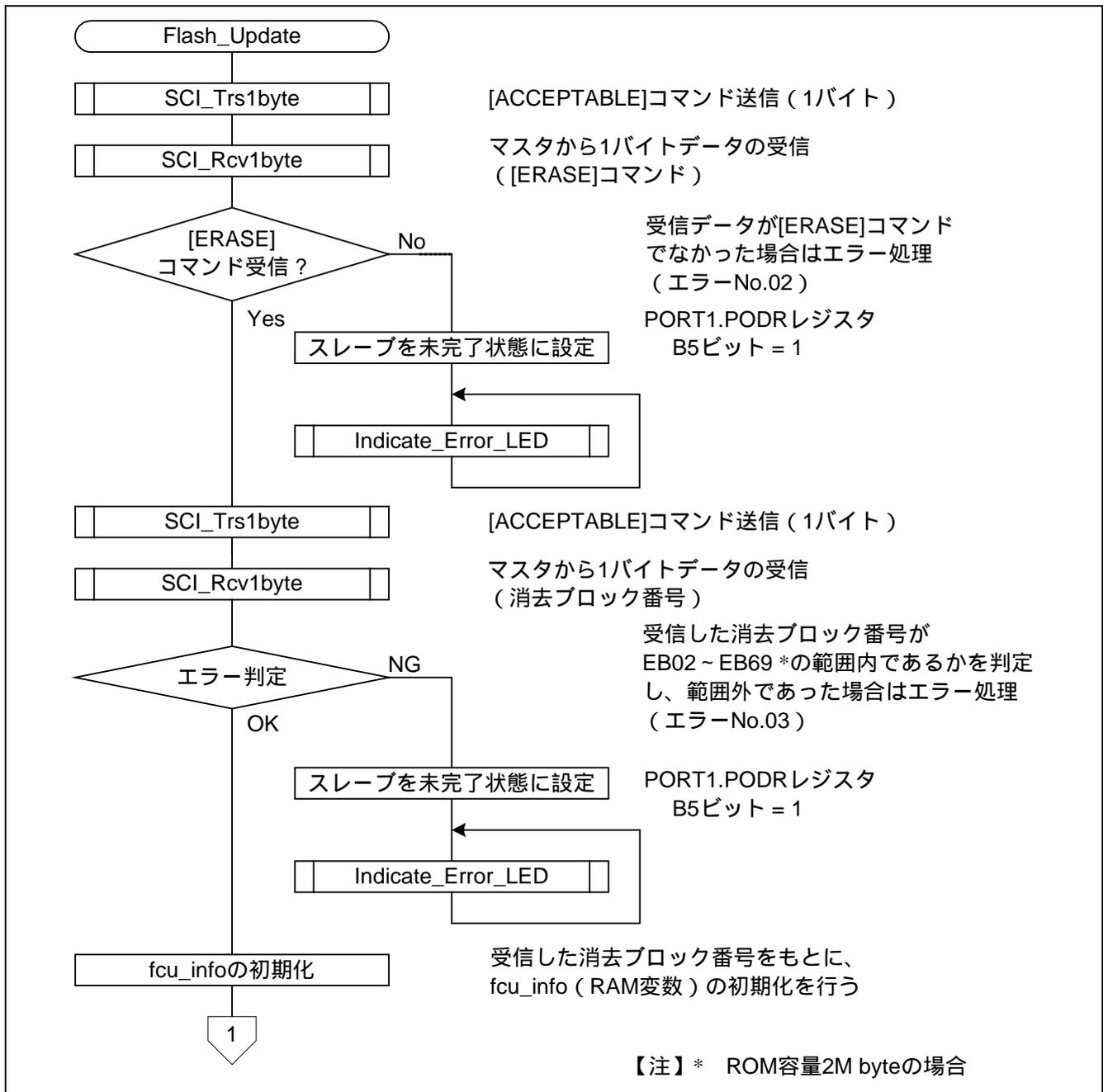


図5.14 書き込み/消去処理 (1)

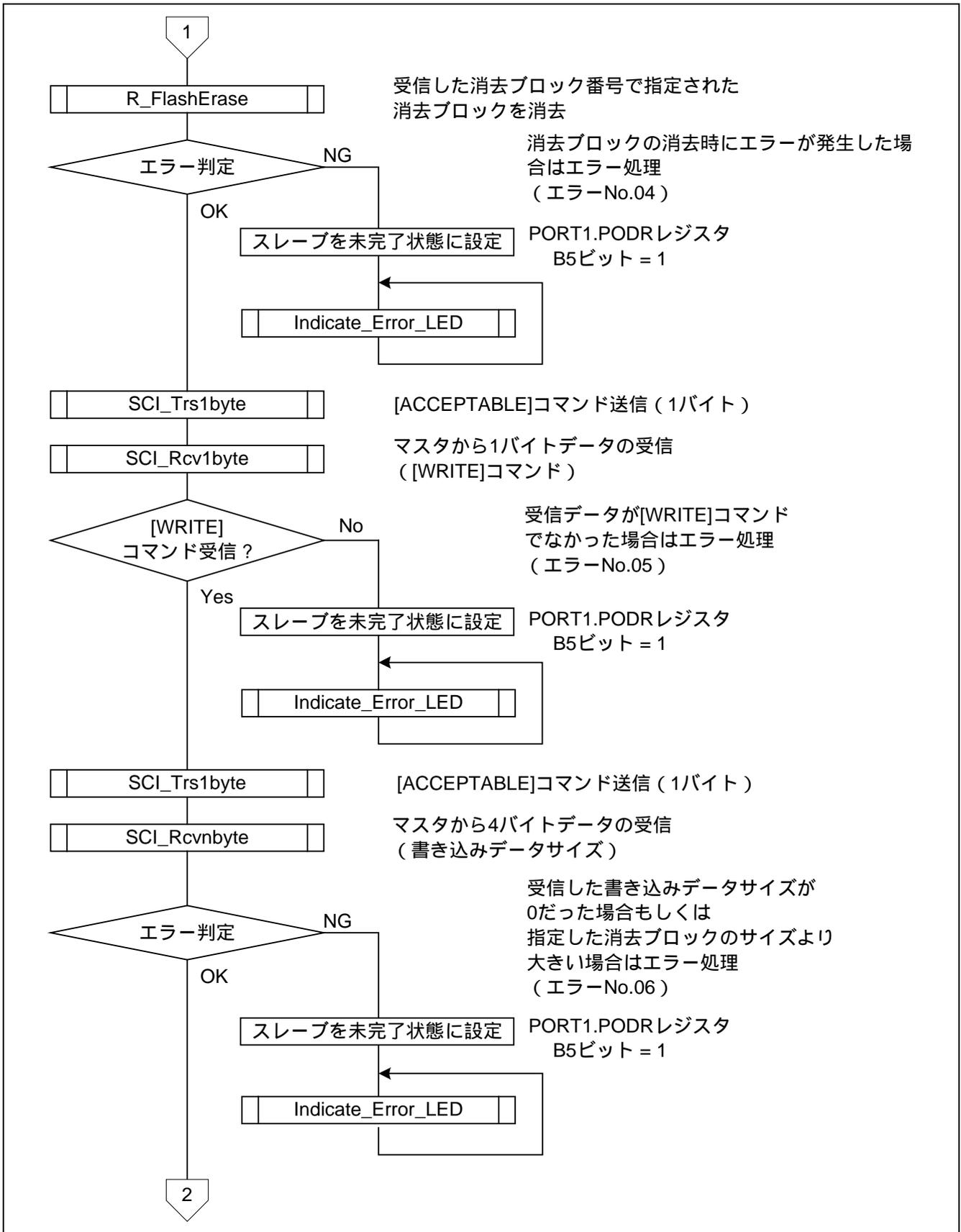


図5.15 書き込み/消去処理 (2)

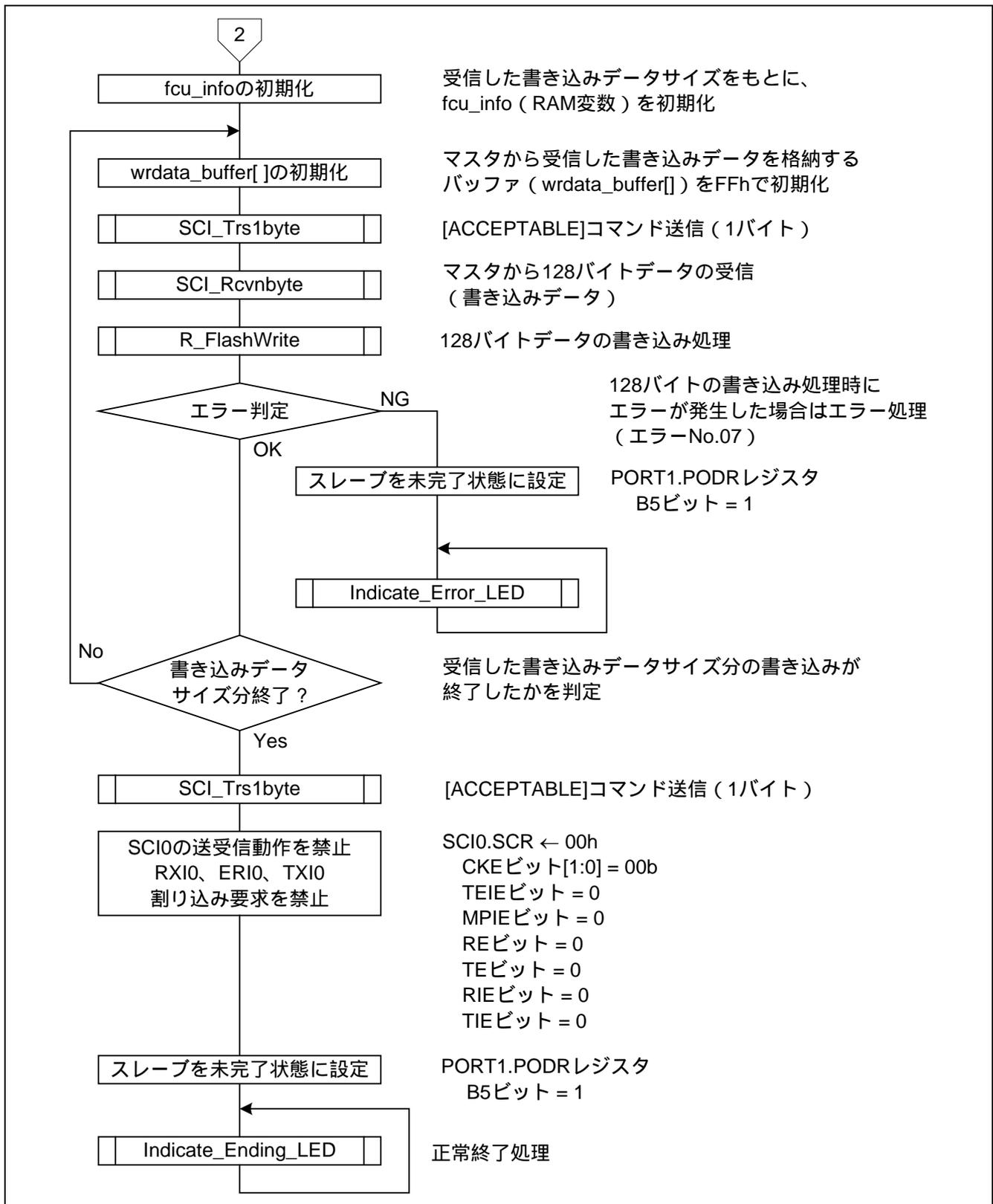


図5.16 書き込み/消去処理 (3)

5.14.5 正常処理関数

図 5.17に正常処理関数のフローチャートを示します。

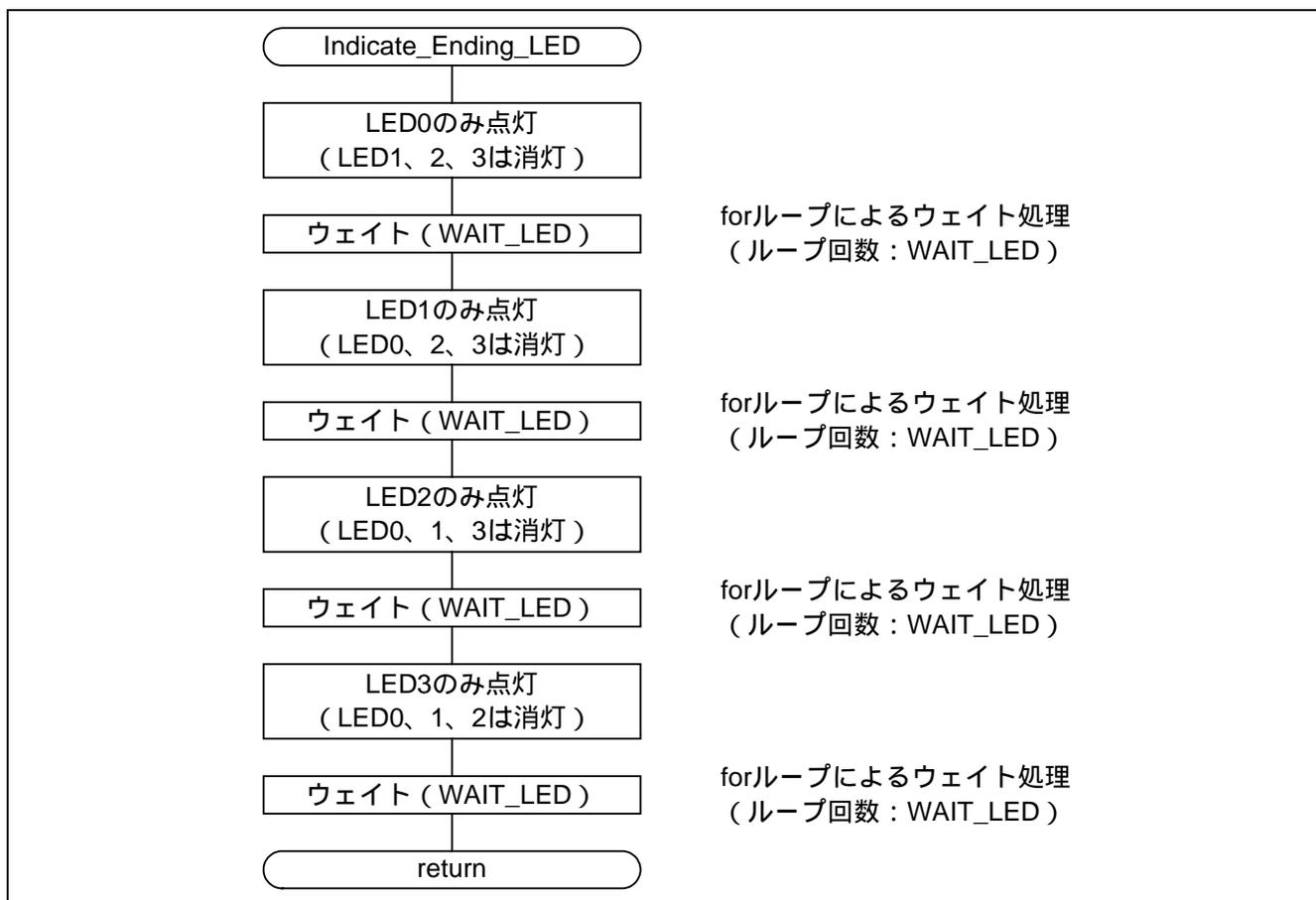


図5.17 正常処理関数

5.14.6 エラー終了処理関数

図 5.18にエラー終了処理関数のフローチャートを示します。

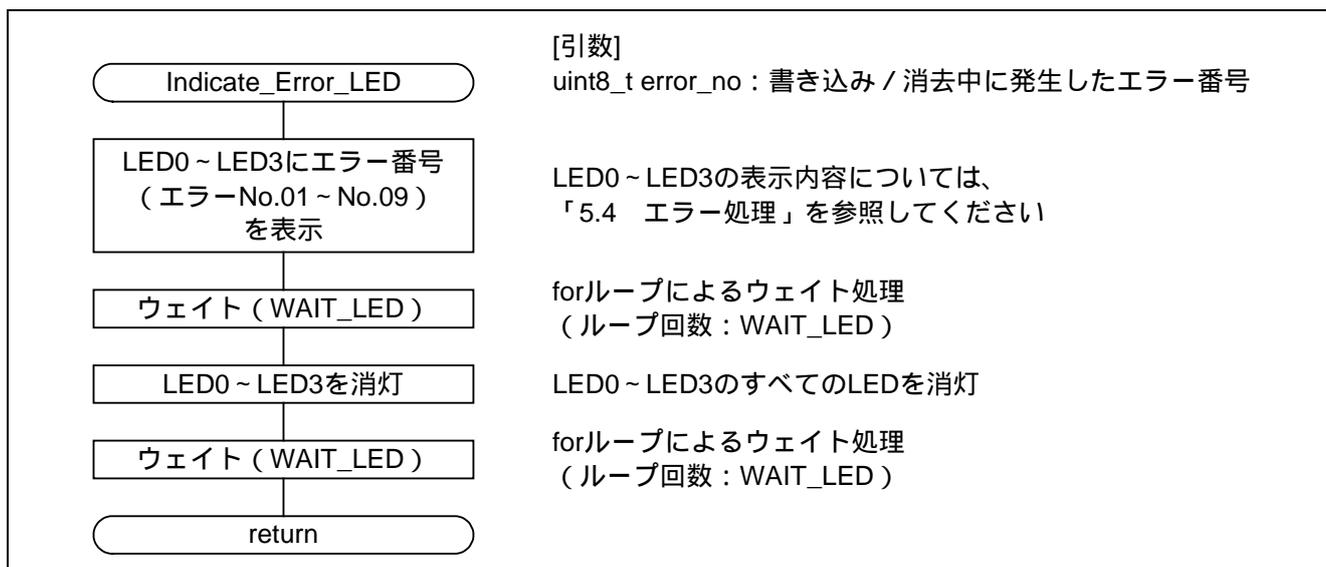


図5.18 エラー終了処理関数

5.14.7 1バイトデータ受信関数

図 5.19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。

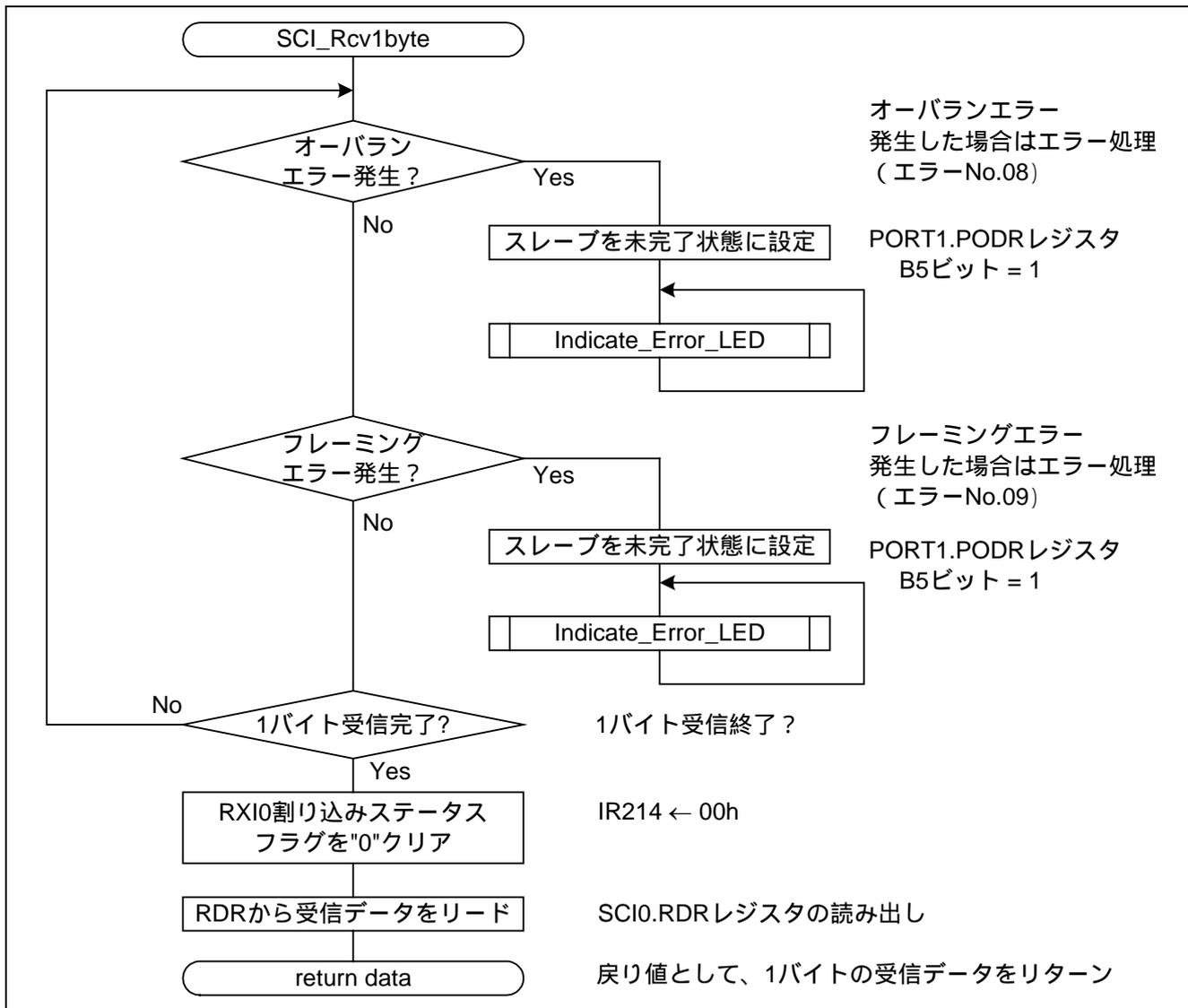


図5.19 1バイトデータ受信関数

5.14.8 n バイトデータ受信関数

図 5.20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。

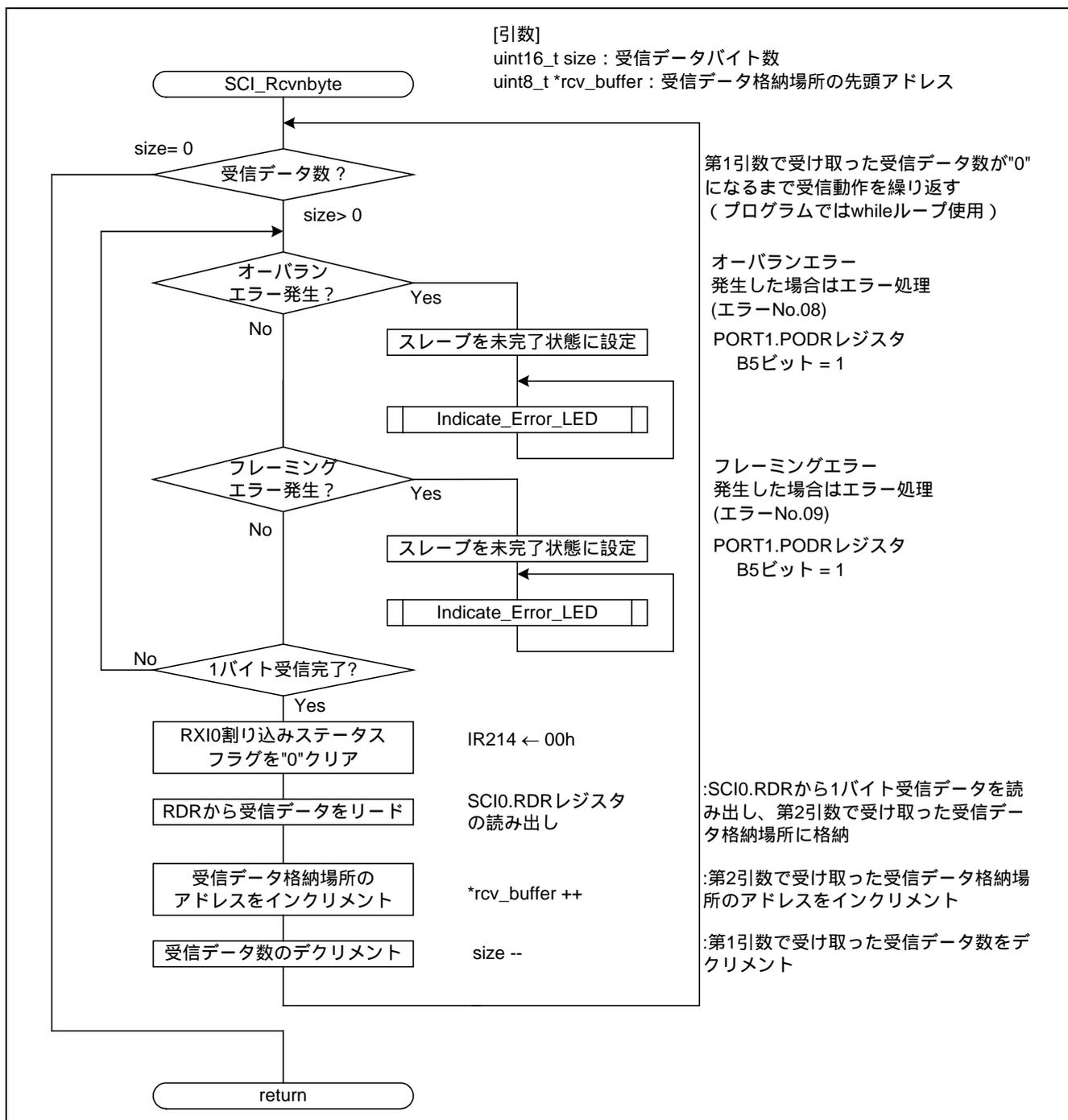


図5.20 n バイトデータ受信関数

5.14.9 1バイトデータ送信関数

図 5.21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。

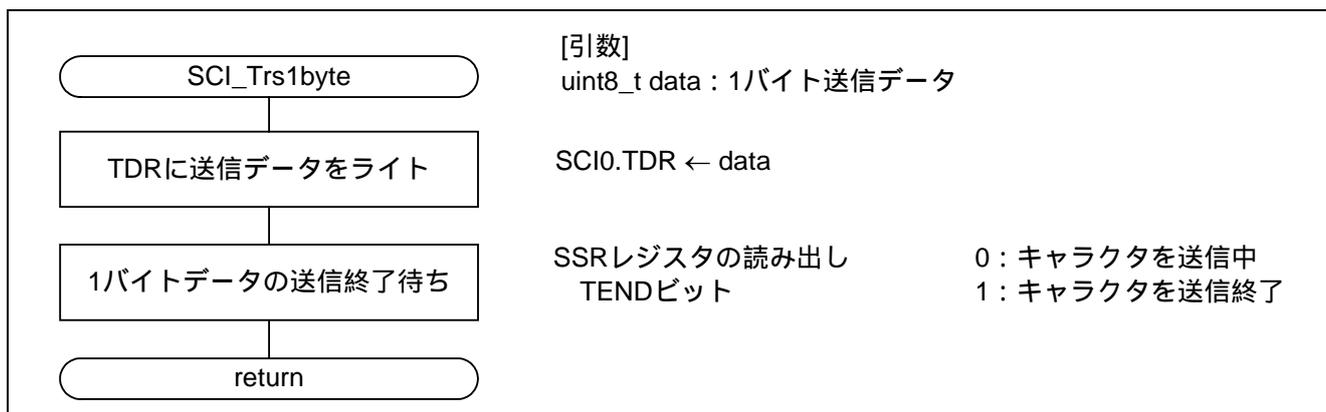


図5.21 1バイトデータ送信関数

5.14.10 MPC 初期設定

図 5.22にMPC 初期設定のフローチャートを示します。

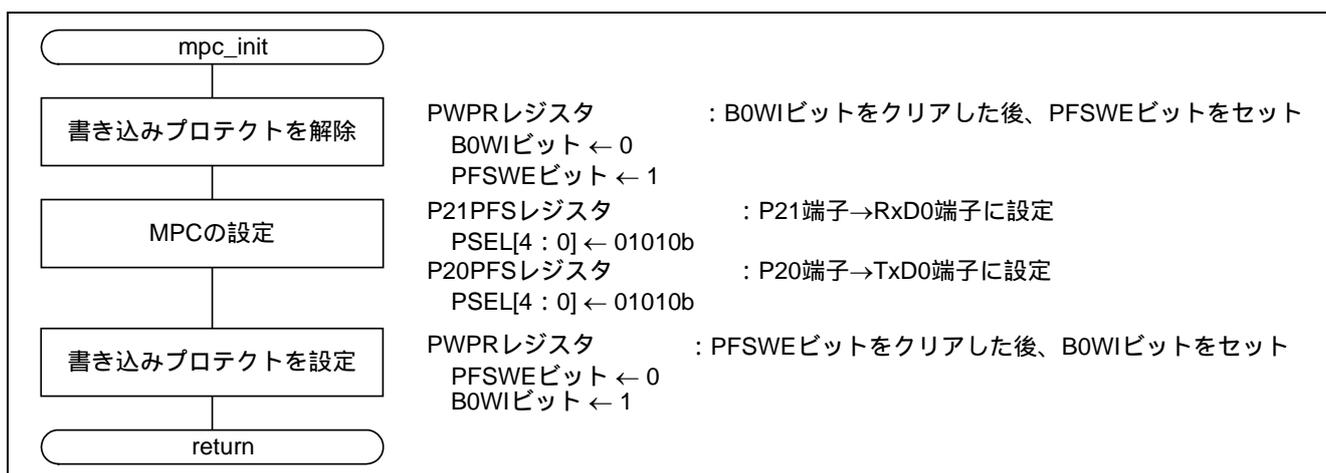


図5.22 MPC 初期設定

5.14.11 PMR 初期設定

図 5.23にPMR 初期設定のフローチャートを示します。

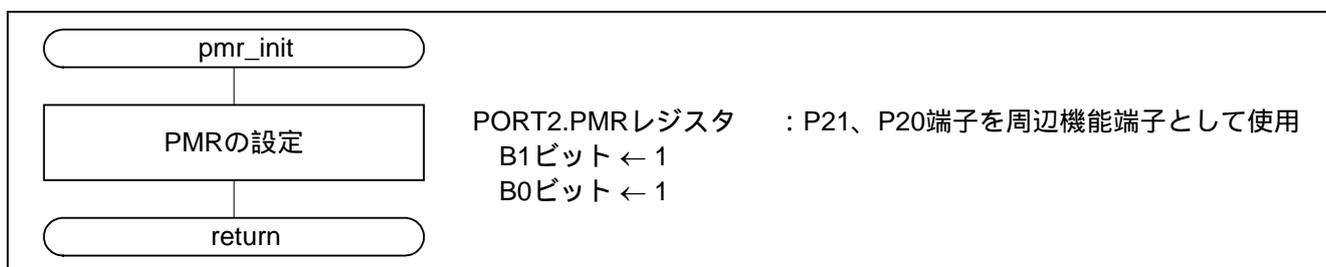


図5.23 PMR 初期設定

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 F000h~00FF FFFF、読み出し用アドレス:FFFF F000h~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」の「ユーザーズマニュアル」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2Mバイトです。ROM容量が1.5Mバイト、1Mバイト、768Kバイト、512Kバイト、または384Kバイトのマイコンを使用する場合は、ファイル“mcu_info.h”内の“ROM_SIZE_BYTES”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1.5Mバイトの場合

```
#define ROM_SIZE_BYTES    (1572864)
```

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
R5F5630E	2M	(2097152)	EB02 ~ EB69
R5F5630D	1.5M	(1572864)	EB02 ~ EB61
R5F5630B	1M	(1048576)	EB02 ~ EB53
R5F5630A	768K	(786432)	EB02 ~ EB45
R5F56308	512K	(524288)	EB02 ~ EB37
R5F56307	384K	(393216)	EB02 ~ EB29

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おりません。

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。

6.5 rom オプションについて

本アプリケーションノートでは、最適化リンカージェディタの rom オプションを使用して、PFRAM (ROM セクション) 内の定義シンボルを RPFram (RAM セクション) 内のアドレスにリロケーションしています。これにより、Flash_Update 関数コール後の実行アドレスはリロケーション先である RAM 上のアドレスとなります。

rom オプションについての詳細は、「8. 参考ドキュメント」の「ユーザーズマニュアル：開発環境 RX ファミリ C/C++ コンパイラパッケージ」を参照してください。

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

RX630グループ ユーザーズマニュアル ハードウェア編 Rev.1.50

(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 (V.1.02 添付資料含む)

(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

RX630グループ

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 (マスタ) Rev.1.00

(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。)

RX600 シリーズ RX600 用のシンプルフラッシュ API Rev.2.20

(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。)

RX630 グループ 初期設定例 Rev.1.00

(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/>

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	2013.04.05	—	初版発行
1.01	2014.12.04	5	表 2.1 最適化リンケージエディタ 追加
		5	表 2.1 注* 追加
		30	図 5.13 注* 追加
		40	6.5 rom オプションについて 追加

製品ご使用上の注意事項

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

1. 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に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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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 〒100-0004 千代田区大手町2-6-2 (日本ビル)

■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/>