
RX210、RX21A、RX220 グループ

R01AN1423JJ0101

Rev.1.01

2014.07.01

SCI を用いた調歩同期式通信

要旨

本アプリケーションノートでは、RX210、RX21A、RX220 グループのシリアルコミュニケーションインタフェース(以下、SCI)を使用して、調歩同期式のシリアル送受信を行う方法について説明します。

対象デバイス

RX210、RX21A、RX220 グループ

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	4
3. 関連アプリケーションノート	4
4. ハードウェア説明	5
4.1 使用端子一覧	5
5. ソフトウェア説明	6
5.1 動作概要	7
5.1.1 シリアル送信	7
5.1.2 シリアル受信	8
5.2 ファイル構成	9
5.3 オプション設定メモリ	9
5.4 定数一覧	10
5.5 構造体/共用体一覧	16
5.6 変数一覧	16
5.7 関数一覧	17
5.8 関数仕様	17
5.9 フローチャート	23
5.9.1 メイン処理	23
5.9.2 ポート初期設定	24
5.9.3 周辺機能初期設定	24
5.9.4 コールバック関数(SCI 送信完了)	24
5.9.5 コールバック関数(SCI 受信完了)	25
5.9.6 コールバック関数(SCI 受信エラー)	25
5.9.7 ユーザ I/F 関数(SCI 初期設定)	26
5.9.8 ユーザ I/F 関数(SCI 受信開始)	28
5.9.9 ユーザ I/F 関数(SCI 送信開始)	29
5.9.10 ユーザ I/F 関数(SCI 状態取得)	30
5.9.11 送信データエンプティ割り込み	30
5.9.12 送信終了割り込み	31
5.9.13 受信データフル割り込み	32
5.9.14 受信エラー割り込み	33
5.9.15 SCI.ERI 割り込み処理	34
5.9.16 SCI.RXI 割り込み処理	34
5.9.17 SCI.TXI 割り込み処理	35
5.9.18 SCI.TEI 割り込み処理	35
6. RX21A、RX220 グループ 初期設定例 アプリケーションノートとの組み合わせ方	36
7. サンプルコード	37
8. 参考ドキュメント	37

1. 仕様

SCI を使用して、調歩同期式のシリアル送受信を行います。

リセット解除後、送信と受信を 1 度だけ行います。送信は、送信バッファに設定した 12 バイトの文字コード「Hello world!」を送信します。12 バイトの送信が完了すると、LED0 を点灯します。

受信は、12 バイトを受信します。受信したデータは受信バッファに格納し、12 バイトの受信が完了すると、LED1 を点灯します。受信中にエラーが発生すると、受信動作を終了し、LED2 を点灯します。

使用する SCI のチャンネルは、コンフィグレーションファイルで選択します。サンプルコードではチャンネル 0 (SCI0) を選択しています。

- 転送速度 : 57600bps
- データ長 : 8 ビット
- ストップビット : 2 ビット
- パリティ : なし
- ハードウェアフロー制御 : なし

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
SCI (チャンネル 0、1、5、6、8、9、12 より選択)	調歩同期式シリアル送受信
I/O ポート	LED 点灯

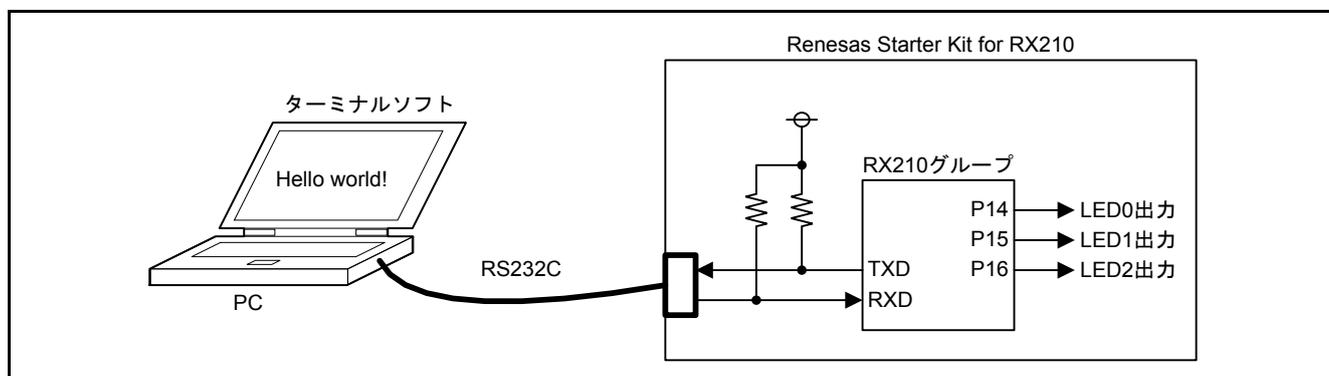


図 1.1 使用例

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"> メインクロック: 20MHz PLL: 100MHz (メインクロック 2 分周 10 通倍) システムクロック (ICLK): 50MHz (PLL 2 分周) 周辺モジュールクロック B (PCLKB): 25MHz (PLL 4 分周)
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)
使用ツール	ターミナルソフト

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。

- RX210グループ 初期設定例 Rev.2.00 (R01AN1002JJ)
- RX21A グループ 初期設定例 Rev.1.10 (R01AN1486JJ)
- RX220 グループ 初期設定例 Rev.1.10 (R01AN1494JJ)

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 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
P14	出力	LED0 出力(SCI 送信完了)
P15	出力	LED1 出力(SCI 受信完了)
P16	出力	LED2 出力(SCI 受信エラー)
P21/RXD0	入力	SCI0 の受信データ入力端子(注 1)
P20/TXD0	出力	SCI0 の送信データ出力端子(注 1)
P15/RXD1	入力	SCI1 の受信データ入力端子(注 1)
P16/TXD1	出力	SCI1 の送信データ出力端子(注 1)
PA3/RXD5	入力	SCI5 の受信データ入力端子(注 1)
PA4/TXD5	出力	SCI5 の送信データ出力端子(注 1)
PB0/RXD6	入力	SCI6 の受信データ入力端子(注 1)
PB1/TXD6	出力	SCI6 の送信データ出力端子(注 1)
PC6/RXD8	入力	SCI8 の受信データ入力端子(注 1)
PC7/TXD8	出力	SCI8 の送信データ出力端子(注 1)
PB6/RXD9	入力	SCI9 の受信データ入力端子(注 1)
PB7/TXD9	出力	SCI9 の送信データ出力端子(注 1)
PE2/RXD12	入力	SCI12 の受信データ入力端子(注 1)
PE1/TXD12	出力	SCI12 の送信データ出力端子(注 1)

注 1. 使用する SCI の端子は、コンフィグレーションファイルで選択する SCI のチャンネルによります。使用しない SCI の端子は、汎用入出力ポートとして使用できます。

5. ソフトウェア説明

リセット解除後、ユーザ I/F 関数(SCI 初期設定)を呼び出し、SCI の初期化を行います。

ユーザ I/F 関数(SCI 受信開始)を呼び出すと、受信動作を許可します。指定バイト数の受信を完了したとき、SCI の受信動作を禁止して、コールバック関数(SCI 受信完了)を呼び出します。コールバック関数(SCI 受信完了)で、LED1 を点灯します。

受信エラーが発生した場合、SCI の受信動作を禁止して、コールバック関数(SCI 受信エラー)を呼び出します。コールバック関数(SCI 受信エラー)で、LED2 を点灯します。

ユーザ I/F 関数(SCI 送信開始)を呼び出すと、送信動作を許可します。指定バイト数の送信を完了したとき、SCI の送信動作を禁止して、コールバック関数(SCI 送信完了)を呼び出します。コールバック関数(SCI 送信完了)で、LED0 を点灯します。

以下に使用する周辺機能の設定を、図 5.1 にソフトウェア構成を示します。

<SCI>

- シリアル通信方式 : 調歩同期式
- 転送速度 : 57600bps
- クロックソース : PCLKB (25MHz)
- データ長 : 8 ビット
- ストップビット : 2 ビット
- パリティ機能 : パリティなし
- 割り込み : 受信エラー割り込み(ERI)を許可
: 受信データフル割り込み(RXI)を許可
: 送信データエンプティ割り込み(TXI)を許可
: 送信終了割り込み(TEI)を許可

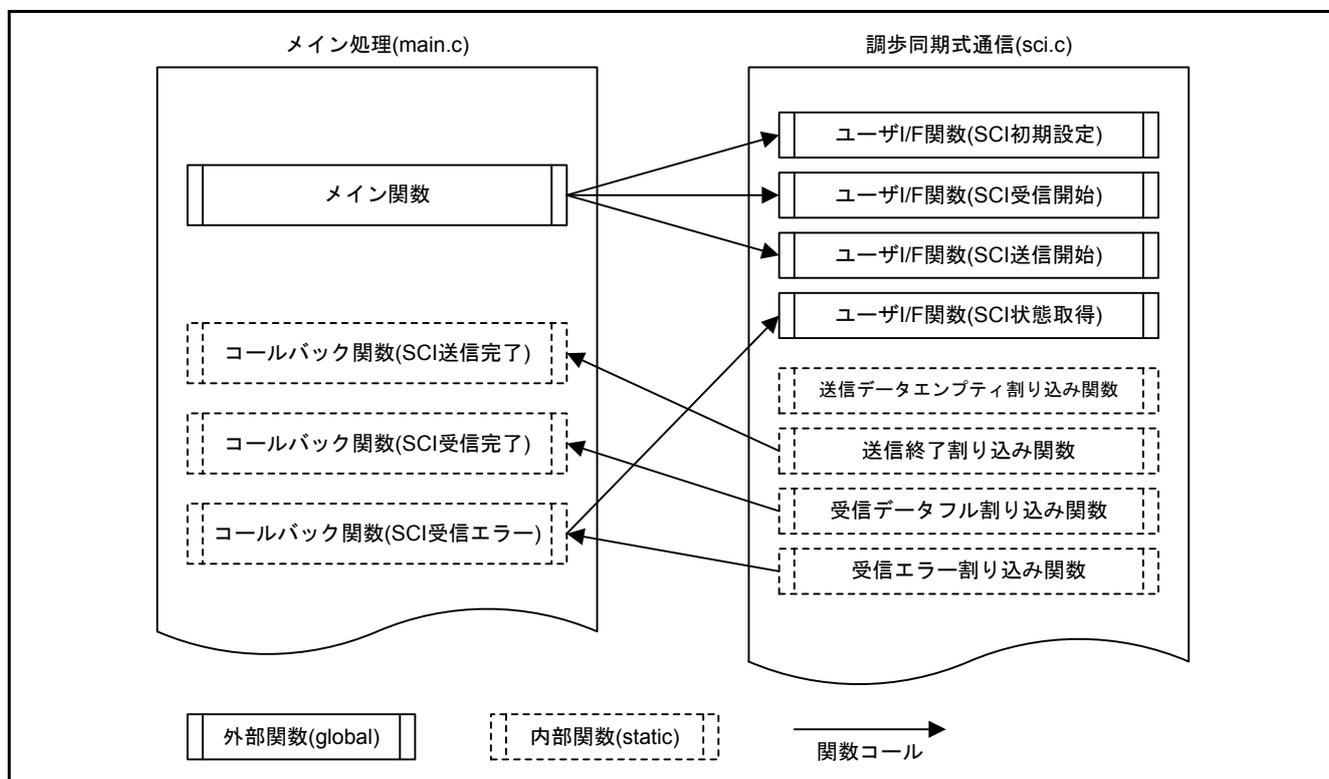


図5.1 ソフトウェア構成

5.1 動作概要

5.1.1 シリアル送信

図 5.2 にシリアル送信のタイミング図を、以下に図中の番号の動作および処理を示します。

- (1) 初期設定
ユーザ I/F 関数(SCI 初期設定)で SCI の初期設定を行い、TXD 端子を “H” 出力にします。
- (2) 送信開始
ユーザ I/F 関数(SCI 送信開始)で変数の送信ビジーフラグを確認します。“1” (送信ビジー)の場合、SCI_BUSY (SCI 送信中)を返します。“0” (送信レディ)の場合、送信ビジーフラグを “1” にして、SCR.TIE ビットを “1” (TXI 割り込み要求を許可)、SCR.TE ビットを “1” (シリアル送信動作を許可)にします。TE ビットを “1” にすると、TXD 端子が使用可能になるので、TXD 端子モード制御ビットを “1” (周辺機能として使用)にして端子機能を TXD 出力に切り替えます。TXI 割り込みの IEN ビットを “1” (割り込み要求許可)にすると、TXI 割り込み要求が発生します。
- (3) データ送信
TXI 割り込み処理で TDR レジスタに送信バッファの値を書きます。最終データを書いたら、TIE ビットを “0” (TXI 割り込み要求を禁止)、SCR.TEIE ビットを “1” (TEI 割り込み要求を許可)にします。
- (4) 送信完了
最終データの送信が完了すると、TEI 割り込み要求が発生します。TEI 割り込み処理で、TXD 端子モード制御ビットを “0” (汎用入出力ポートとして使用)にしてから、TE ビットを “0” (シリアル送信動作を禁止)、TEIE ビットを “0” (TEI 割り込み要求を禁止)にします。送信ビジーフラグを “0” にして、コールバック関数(SCI 送信完了)を呼び出します。

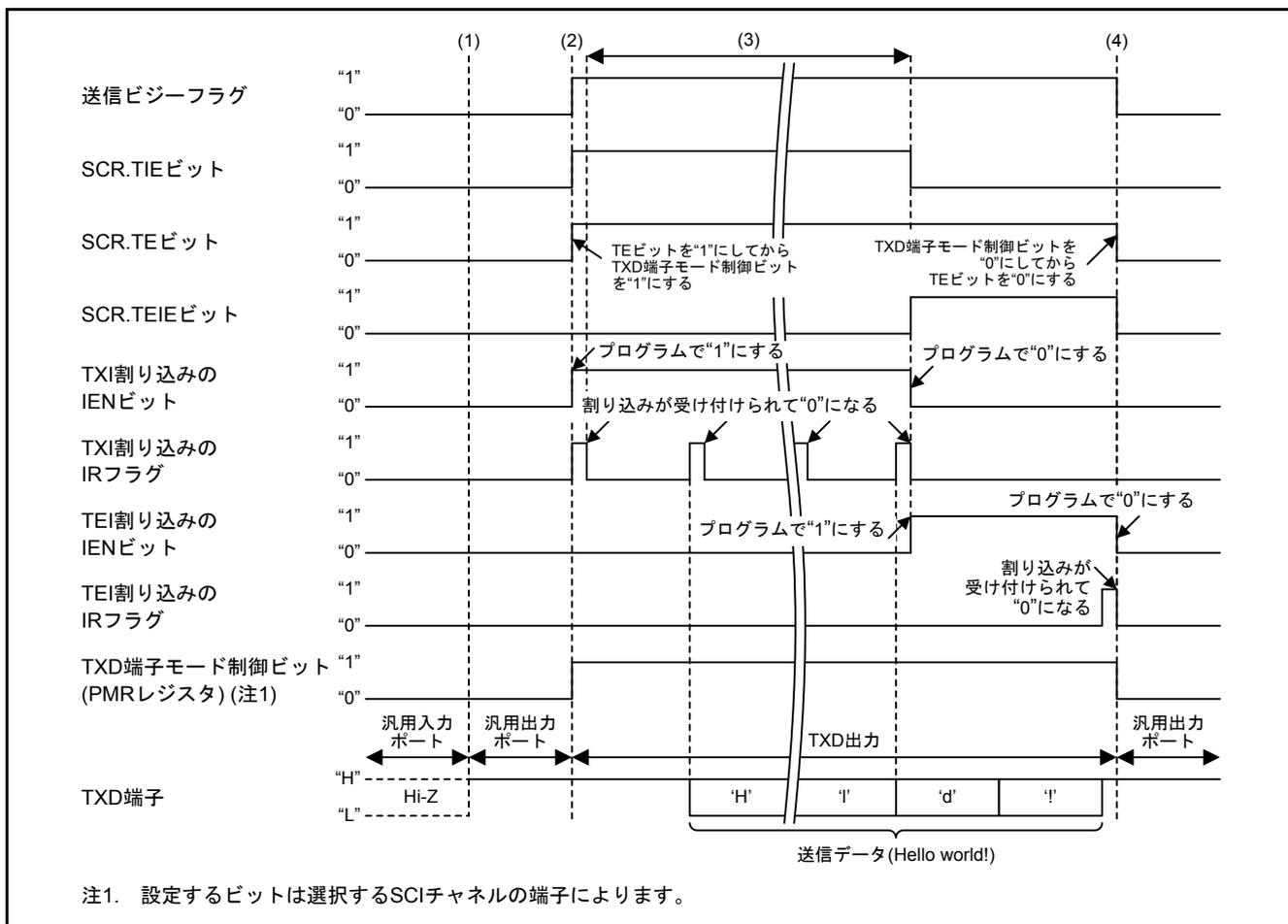


図5.2 シリアル送信のタイミング図

5.1.2 シリアル受信

図 5.3 にシリアル受信のタイミング図を、以下に図中の番号の動作および処理を示します。

- (1) 初期設定
ユーザ I/F 関数(SCI 初期設定)で SCI の初期設定を行い、RXD 端子の端子機能を RXD 入力に切り替えます。
- (2) 受信開始
ユーザ I/F 関数(SCI 受信開始)で変数の受信ビジーフラグを確認します。“1” (受信ビジー)の場合、SCI_BUSY (SCI 受信中)を返します。“0” (受信レディ)の場合、受信ビジーフラグを“1”にして、エラーフラグをクリアします。SCR.RIE ビットを“1” (RXI および ERI 割り込み要求を許可)、SCR.RE ビットを“1” (シリアル受信動作を許可)にして、RXI および ERI 割り込みの IEN ビットを“1” (割り込み要求許可)にします。
- (3) データ受信
データを受信すると、RXI 割り込み要求が発生します。RXI 割り込み処理で RDR レジスタの値を受信バッファに格納します。
受信エラーが発生すると、ERI 割り込み要求が発生します。ERI 割り込み処理で変数のエラーフラグをセットし、RDR レジスタをダミーリードします。RE ビットを“0”にして、SSR レジスタのエラーフラグをクリアします。RIE ビットを“0”、受信ビジーフラグを“0”にして、コールバック関数(SCI 受信エラー)を呼び出します。
- (4) 受信完了
最終データを受信すると、RXI 割り込み処理で RE ビットを“0” (シリアル受信動作を禁止)、RIE ビットを“0” (RXI 割り込み要求を禁止)にします。受信ビジーフラグを“0”にして、コールバック関数(SCI 受信完了)を呼び出します。

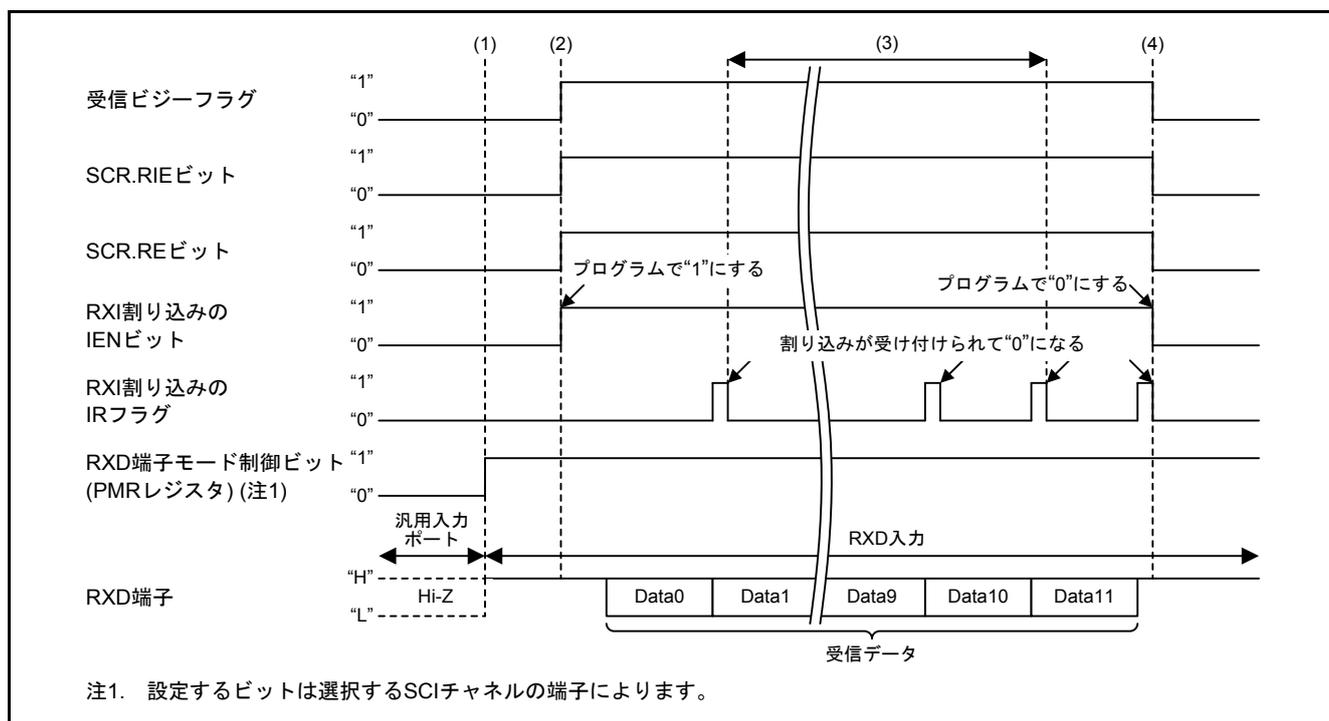


図5.3 シリアル受信のタイミング図

5.2 ファイル構成

表 5.1 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は除きます。

表5.1 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
main.c	メイン処理	
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	
sci.c	調歩同期式通信	
sci.h	sci.c のヘッダファイル	
sci_cfg.h	sci.c のコンフィグレーションヘッダファイル	SCI のチャンネル選択

5.3 オプション設定メモリ

表 5.2 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。

表5.2 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	FFFF FF83h~FFFF FF80h	FFFF FFFFh	リトルエンディアン

5.4 定数一覧

表 5.3～表 5.12にサンプルコードで使用する定数を示します。

表5.3 サンプルコードで使用する定数(main.c)

定数名	設定値	内容
LED0_REG_PODR	PORT1.PODR.BIT.B4	LED0 出力データ格納ビット
LED0_REG_PDR	PORT1.PDR.BIT.B4	LED0 方向制御ビット
LED0_REG_PMR	PORT1.PMR.BIT.B4	LED0 端子モード制御ビット
LED1_REG_PODR	PORT1.PODR.BIT.B5	LED1 出力データ格納ビット
LED1_REG_PDR	PORT1.PDR.BIT.B5	LED1 方向制御ビット
LED1_REG_PMR	PORT1.PMR.BIT.B5	LED1 端子モード制御ビット
LED2_REG_PODR	PORT1.PODR.BIT.B6	LED2 出力データ格納ビット
LED2_REG_PDR	PORT1.PDR.BIT.B6	LED2 方向制御ビット
LED2_REG_PMR	PORT1.PMR.BIT.B6	LED2 端子モード制御ビット
LED_ON	0	LED 出力データ: 点灯
LED_OFF	1	LED 出力データ: 消灯
BUF_SIZE	12	バッファサイズ
NULL_SIZE	1	NULL コードサイズ
SCI_B_TX_BUSY	sci_state.bit.b_tx_busy	送信ビジーフラグ 0: 送信レディ 1: 送信ビジー
SCI_B_RX_BUSY	sci_state.bit.b_rx_busy	受信ビジーフラグ 0: 受信レディ 1: 受信ビジー
SCI_B_RX_ORER	sci_state.bit.b_rx_orer	オーバランエラーフラグ 0: オーバランエラーなし 1: オーバランエラーあり
SCI_B_RX_FER	sci_state.bit.b_rx_fer	フレーミングエラーフラグ 0: フレーミングエラーなし 1: フレーミングエラーあり

表5.4 サンプルコードで使用する定数(sci.c)

定数名	設定値	内容
SSR_ERROR_FLAGS	38h	SCI.SSR レジスタのエラーフラグのビットパターン
B_TX_BUSY	state.bit.b_tx_busy	送信ビジーフラグ 0: 送信レディ 1: 送信ビジー
B_RX_BUSY	state.bit.b_rx_busy	受信ビジーフラグ 0: 受信レディ 1: 受信ビジー
B_RX_ORER	state.bit.b_rx_orer	オーバランエラーフラグ 0: オーバランエラーなし 1: オーバランエラーあり
B_RX_FER	state.bit.b_rx_fer	フレーミングエラーフラグ 0: フレーミングエラーなし 1: フレーミングエラーあり

表5.5 サンプルコードで使用する定数(sci.h)

定数名	設定値	内容
SCI_OK	00h	SCI_StartTransmit 関数、SCI_StartReceive 関数のリターン値: SCI 送信/受信開始
SCI_BUSY	01h	SCI_StartTransmit 関数、SCI_StartReceive 関数のリターン値: SCI 送信/受信済
SCI_NG	02h	SCI_StartTransmit 関数、SCI_StartReceive 関数のリターン値: 引数エラー(送信/受信バイト数が 0)

表5.6 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI0 選択時) (注 1)

定数名	設定値	内容
SCIn	SCI0	SCI チャンネル: SCI0
MSTP_SCIn	MSTP(SCI0)	SCI0 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI0,)	SCI0 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI0,ERI0)	SCI0.ERI0 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI0,RXI0)	SCI0.RXI0 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI0,TXI0)	SCI0.TXI0 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI0,TEI0)	SCI0.TEI0 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI0,ERI0)	SCI0.ERI0 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI0,RXI0)	SCI0.RXI0 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI0,TXI0)	SCI0.TXI0 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI0,TEI0)	SCI0.TEI0 割り込み要求許可ビット
RXDn_PDR	PORT2.PDR.BIT.B1	P21 方向制御ビット
RXDn_PMR	PORT2.PMR.BIT.B1	P21 端子モード制御ビット
RXDnPFS	P21PFS	P21 端子機能制御レジスタ
TXDn_PODR	PORT2.PODR.BIT.B0	P20 出力データ格納ビット
TXDn_PDR	PORT2.PDR.BIT.B0	P20 方向制御ビット
TXDn_PMR	PORT2.PMR.BIT.B0	P20 端子モード制御ビット
TXDnPFS	P20PFS	P20 端子機能制御レジスタ
PSEL_SETTING	0Ah	端子機能選択ビット設定値: RXD0、TXD0

注 1. 48 ピン、64 ピンの製品には SCI0 はありません。

表5.7 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI1 選択時)

定数名	設定値	内容
SCIn	SCI1	SCI チャンネル: SCI1
MSTP_SCIn	MSTP(SCI1)	SCI1 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI1,)	SCI1 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI1,ERI1)	SCI1.ERI1 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI1,RXI1)	SCI1.RXI1 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI1,TXI1)	SCI1.TXI1 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI1,TEI1)	SCI1.TEI1 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI1,ERI1)	SCI1.ERI1 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI1,RXI1)	SCI1.RXI1 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI1,TXI1)	SCI1.TXI1 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI1,TEI1)	SCI1.TEI1 割り込み要求許可ビット
RXDn_PDR	PORT1.PDR.BIT.B5	P15 方向制御ビット
RXDn_PMR	PORT1.PMR.BIT.B5	P15 端子モード制御ビット
RXDnPFS	P15PFS	P15 端子機能制御レジスタ
TXDn_PODR	PORT1.PODR.BIT.B6	P16 出力データ格納ビット
TXDn_PDR	PORT1.PDR.BIT.B6	P16 方向制御ビット
TXDn_PMR	PORT1.PMR.BIT.B6	P16 端子モード制御ビット
TXDnPFS	P16PFS	P16 端子機能制御レジスタ
PSEL_SETTING	0Ah	端子機能選択ビット設定値: RXD1、TXD1

表5.8 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI5 選択時)

定数名	設定値	内容
SCIn	SCI5	SCI チャンネル: SCI5
MSTP_SCIn	MSTP(SCI5)	SCI5 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI5,)	SCI5 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI5,ERI5)	SCI5.ERI5 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI5,RXI5)	SCI5.RXI5 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI5,TXI5)	SCI5.TXI5 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI5,TEI5)	SCI5.TEI5 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI5,ERI5)	SCI5.ERI5 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI5,RXI5)	SCI5.RXI5 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI5,TXI5)	SCI5.TXI5 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI5,TEI5)	SCI5.TEI5 割り込み要求許可ビット
RXDn_PDR	PORTA.PDR.BIT.B3	PA3 方向制御ビット
RXDn_PMR	PORTA.PMR.BIT.B3	PA3 端子モード制御ビット
RXDnPFS	PA3PFS	PA3 端子機能制御レジスタ
TXDn_PODR	PORTA.PODR.BIT.B4	PA4 出力データ格納ビット
TXDn_PDR	PORTA.PDR.BIT.B4	PA4 方向制御ビット
TXDn_PMR	PORTA.PMR.BIT.B4	PA4 端子モード制御ビット
TXDnPFS	PA4PFS	PA4 端子機能制御レジスタ
PSEL_SETTING	0Ah	端子機能選択ビット設定値: RXD5、TXD5

表5.9 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI6 選択時)

定数名	設定値	内容
SCIn	SCI6	SCI チャンネル: SCI6
MSTP_SCIn	MSTP(SCI6)	SCI6 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI6,)	SCI6 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI6,ERI6)	SCI6.ERI6 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI6,RXI6)	SCI6.RXI6 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI6,TXI6)	SCI6.TXI6 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI6,TEI6)	SCI6.TEI6 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI6,ERI6)	SCI6.ERI6 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI6,RXI6)	SCI6.RXI6 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI6,TXI6)	SCI6.TXI6 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI6,TEI6)	SCI6.TEI6 割り込み要求許可ビット
RXDn_PDR	PORTB.PDR.BIT.B0	PB0 方向制御ビット
RXDn_PMR	PORTB.PMR.BIT.B0	PB0 端子モード制御ビット
RXDnPFS	PB0PFS	PB0 端子機能制御レジスタ
TXDn_PODR	PORTB.PODR.BIT.B1	PB1 出力データ格納ビット
TXDn_PDR	PORTB.PDR.BIT.B1	PB1 方向制御ビット
TXDn_PMR	PORTB.PMR.BIT.B1	PB1 端子モード制御ビット
TXDnPFS	PB1PFS	PB1 端子機能制御レジスタ
PSEL_SETTING	0Bh	端子機能選択ビット設定値: RXD6、TXD6

表5.10 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI8 選択時)

定数名	設定値	内容
SCIn	SCI8	SCI チャンネル: SCI8
MSTP_SCIn	MSTP(SCI8)	SCI8 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI8,)	SCI8 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI8,ERI8)	SCI8.ERI8 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI8,RXI8)	SCI8.RXI8 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI8,TXI8)	SCI8.TXI8 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI8,TEI8)	SCI8.TEI8 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI8,ERI8)	SCI8.ERI8 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI8,RXI8)	SCI8.RXI8 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI8,TXI8)	SCI8.TXI8 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI8,TEI8)	SCI8.TEI8 割り込み要求許可ビット
RXDn_PDR	PORTC.PDR.BIT.B6	PC6 方向制御ビット
RXDn_PMR	PORTC.PMR.BIT.B6	PC6 端子モード制御ビット
RXDnPFS	PC6PFS	PC6 端子機能制御レジスタ
TXDn_PODR	PORTC.PODR.BIT.B7	PC7 出力データ格納ビット
TXDn_PDR	PORTC.PDR.BIT.B7	PC7 方向制御ビット
TXDn_PMR	PORTC.PMR.BIT.B7	PC7 端子モード制御ビット
TXDnPFS	PC7PFS	PC7 端子機能制御レジスタ
PSEL_SETTING	0Ah	端子機能選択ビット設定値: RXD8、TXD8

表5.11 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI9 選択時) (注 1)

定数名	設定値	内容
SCIn	SCI9	SCI チャンネル: SCI9
MSTP_SCIn	MSTP(SCI9)	SCI9 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI9,)	SCI9 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI9,ERI9)	SCI9.ERI9 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI9,RXI9)	SCI9.RXI9 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI9,TXI9)	SCI9.TXI9 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI9,TEI9)	SCI9.TEI9 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI9,ERI9)	SCI9.ERI9 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI9,RXI9)	SCI9.RXI9 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI9,TXI9)	SCI9.TXI9 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI9,TEI9)	SCI9.TEI9 割り込み要求許可ビット
RXDn_PDR	PORTB.PDR.BIT.B6	PB6 方向制御ビット
RXDn_PMR	PORTB.PMR.BIT.B6	PB6 端子モード制御ビット
RXDnPFS	PB6PFS	PB6 端子機能制御レジスタ
TXDn_PODR	PORTB.PODR.BIT.B7	PB7 出力データ格納ビット
TXDn_PDR	PORTB.PDR.BIT.B7	PB7 方向制御ビット
TXDn_PMR	PORTB.PMR.BIT.B7	PB7 端子モード制御ビット
TXDnPFS	PB7PFS	PB7 端子機能制御レジスタ
PSEL_SETTING	0Ah	端子機能選択ビット設定値: RXD9、TXD9

注 1. 48 ピンの製品には SCI9 はありません。

表5.12 サンプルコードで使用する定数(sci_cfg.h: SELECT_SCI12 選択時)

定数名	設定値	内容
SCIn	SCI12	SCI チャンネル: SCI12
MSTP_SCIn	MSTP(SCI12)	SCI12 モジュールストップ設定ビット
IPR_SCIn	IPR(SCI12,)	SCI12 割り込み優先レベル設定ビット
IR_SCIn_ERIn	IR(SCI12,ERI12)	SCI12.ERI12 割り込みステータスフラグ
IR_SCIn_RXIn	IR(SCI12,RXI12)	SCI12.RXI12 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI12,TXI12)	SCI12.TXI12 割り込みステータスフラグ
IR_SCIn_TEIn	IR(SCI12,TEI12)	SCI12.TEI12 割り込みステータスフラグ
IEN_SCIn_ERIn	IEN(SCI12,ERI12)	SCI12.ERI12 割り込み要求許可ビット
IEN_SCIn_RXIn	IEN(SCI12,RXI12)	SCI12.RXI12 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI12,TXI12)	SCI12.TXI12 割り込み要求許可ビット
IEN_SCIn_TEIn	IEN(SCI12,TEI12)	SCI12.TEI12 割り込み要求許可ビット
RXDn_PDR	PORTE.PDR.BIT.B2	PE2 方向制御ビット
RXDn_PMR	PORTE.PMR.BIT.B2	PE2 端子モード制御ビット
RXDnPFS	PE2PFS	PE2 端子機能制御レジスタ
TXDn_PODR	PORTE.PODR.BIT.B1	PE1 出力データ格納ビット
TXDn_PDR	PORTE.PDR.BIT.B1	PE1 方向制御ビット
TXDn_PMR	PORTE.PMR.BIT.B1	PE1 端子モード制御ビット
TXDnPFS	PE1PFS	PE1 端子機能制御レジスタ
PSEL_SETTING	0Ch	端子機能選択ビット設定値: RXD12、TXD12

5.5 構造体/共用体一覧

図 5.4にサンプルコードで使用する構造体/共用体を示します。

```
#pragma bit_order left /* ビットフィールドの並び順指定: 上位ビット側からメンバを割り付ける */
#pragma unpack /* 構造体メンバの境界調整数指定: メンバの型でアライメントする */
typedef union
{
    uint8_t byte;
    struct
    {
        uint8_t b_tx_busy :1; /* 送信ビジーフラグ 0:送信レディ 1:送信ビジー */
        uint8_t b_rx_busy :1; /* 受信ビジーフラグ 0:受信レディ 1:受信ビジー */
        uint8_t b_rx_orer :1; /* オーバランエラーフラグ 0:エラーなし 1:オーバランエラー */
        uint8_t b_rx_fer :1; /* フレーミングエラーフラグ 0:エラーなし 1:フレーミングエラー */
        uint8_t :4; /* 使用しない */
    } bit;
} sci_state_t;
#pragma packoption /* 構造体メンバの境界調整数指定の終了 */
#pragma bit_order /* ビットフィールドの並び順指定の終了 */
```

図5.4 サンプルコードで使用する構造体/共用体

5.6 変数一覧

表 5.13にstatic 型変数を示します。

表5.13 static 型変数

型	変数名	内容	使用関数
static uint8_t	rx_buf[BUF_SIZE]	受信バッファ	main
static uint8_t	tx_buf[]	送信バッファ	main
static sci_state_t	sci_state	SCI 状態	cb_sci_rx_error
static const uint8_t *	pbuf_tx	送信バッファへのポインタ	SCI_StartTransmit
static uint8_t	tx_cnt	送信カウンタ	sci_txi_isr
static uint8_t *	pbuf_rx	受信バッファへのポインタ	SCI_StartReceive
static uint8_t	rx_cnt	受信カウンタ	sci_rxi_isr
static sci_state_t	state	SCI 状態	SCI_StartReceive SCI_StartTransmit SCI_GetState sci_tei_isr sci_rxi_isr sci_eri_isr

5.7 関数一覧

表 5.14にサンプルコードで使用する関数を示します。

表5.14 サンプルコードで使用する関数

関数名	概要
main	メイン処理
port_init	ポート初期設定
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
peripheral_init	周辺機能初期設定
cb_sci_tx_end	コールバック関数(SCI 送信完了)
cb_sci_rx_end	コールバック関数(SCI 受信完了)
cb_sci_rx_error	コールバック関数(SCI 受信エラー)
SCI_Init	ユーザ I/F 関数(SCI 初期設定)
SCI_StartReceive	ユーザ I/F 関数(SCI 受信開始)
SCI_StartTransmit	ユーザ I/F 関数(SCI 送信開始)
SCI_GetState	ユーザ I/F 関数(SCI 状態取得)
sci_txi_isr	送信データエンプティ割り込み
sci_tei_isr	送信終了割り込み
sci_rxi_isr	受信データフル割り込み
sci_eri_isr	受信エラー割り込み
Excep_SCIn_ERIn	SCI.ERI 割り込み処理
Excep_SCIn_RXIn	SCI.RXI 割り込み処理
Excep_SCIn_TXIn	SCI.TXI 割り込み処理
Excep_SCIn_TEIn	SCI.TEI 割り込み処理

5.8 関数仕様

サンプルコードの関数仕様を示します。

main	
概要	メイン処理
ヘッダ	なし
宣言	void main(void)
説明	初期設定後、SCI の受信を開始して、送信を開始します。
引数	なし
リターン値	なし

port_init	
概要	ポート初期設定
ヘッダ	なし
宣言	static void port_init(void)
説明	ポートの初期設定を行います。
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、各グループのアプリケーションノート「初期設定例」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、各グループのアプリケーションノート「初期設定例」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、各グループのアプリケーションノート「初期設定例」を参照してください。

peripheral_init	
概要	周辺機能初期設定
ヘッダ	なし
宣言	static void peripheral_init(void)
説明	使用する周辺機能の初期設定を行います。
引数	なし
リターン値	なし

cb_sci_tx_end	
概要	コールバック関数(SCI 送信完了)
ヘッダ	なし
宣言	static void cb_sci_tx_end(void)
説明	SCI 送信完了時に呼び出されます。
引数	なし
リターン値	なし

cb_sci_rx_end	
概要	コールバック関数(SCI 受信完了)
ヘッダ	なし
宣言	static void cb_sci_rx_end(void)
説明	SCI 受信完了時に呼び出されます。
引数	なし
リターン値	なし

cb_sci_rx_error	
概要	コールバック関数(SCI 受信エラー)
ヘッダ	なし
宣言	static void cb_sci_rx_error(void)
説明	SCI 受信エラー発生時に呼び出されます。
引数	なし
リターン値	なし
備考	サンプルコードではエラー処理を行っていません。必要に応じてプログラムを追加してください。

SCI_Init	
概要	ユーザ I/F 関数(SCI 初期設定)
ヘッダ	sci.h
宣言	void SCI_Init(void)
説明	SCI の初期設定を行います。
引数	なし
リターン値	なし

SCI_StartReceive

概要	ユーザ I/F 関数(SCI 受信開始)		
ヘッダ	sci.h		
宣言	uint8_t SCI_StartReceive(uint8_t * pbuf, uint8_t num, CallbackFunc pcb_rx_end, CallbackFunc pcb_rx_error)		
説明	SCI 受信を開始します。		
引数	uint8_t *	pbuf	: 受信データ格納ポインタ
	uint8_t	num	: 受信バイト数
	CallbackFunc	pcb_rx_end	: コールバック関数ポインタ(受信完了)
	CallbackFunc	pcb_rx_error	: コールバック関数ポインタ(受信エラー)
リターン値	SCI_NG	: 引数エラー(受信バイト数が 0)	
	SCI_BUSY	: SCI 受信中	
	SCI_OK	: SCI 受信開始	

SCI_StartTransmit

概要	ユーザ I/F 関数(SCI 送信開始)		
ヘッダ	sci.h		
宣言	uint8_t SCI_StartTransmit(const uint8_t * pbuf, uint8_t num, CallbackFunc pcb_tx_end)		
説明	SCI 送信を開始します。		
引数	const uint8_t *	pbuf	: 送信データ格納ポインタ
	uint8_t	num	: 送信バイト数
	CallbackFunc	pcb_tx_end	: コールバック関数ポインタ(送信完了)
リターン値	SCI_NG	: 引数エラー(送信バイト数が 0)	
	SCI_BUSY	: SCI 送信中	
	SCI_OK	: SCI 送信開始	

SCI_GetState

概要	ユーザ I/F 関数(SCI 状態取得)		
ヘッダ	sci.h		
宣言	sci_state_t SCI_GetState(void)		
説明	SCI 状態を返します。		
引数	なし		
リターン値	sci_state_t.bit.b_tx_busy	: 送信ビジーフラグ 0:送信レディ 1:送信ビジー	
	sci_state_t.bit.b_rx_busy	: 受信ビジーフラグ 0:受信レディ 1:受信ビジー	
	sci_state_t.bit.b_rx_orer	: オーバランエラーフラグ 0:エラーなし 1:オーバランエラー	
	sci_state_t.bit.b_rx_fer	: フレーミングエラーフラグ 0:エラーなし 1:フレーミングエラー	

sci_txi_isr	
概要	送信データエンプティ割り込み
ヘッダ	なし
宣言	static void sci_txi_isr(void)
説明	SCI.TXI 割り込み処理関数で呼び出されます。送信データを書き込みます。最終データを送信すると、TXI 割り込み要求を禁止し、TEI 割り込み要求を許可します。
引数	なし
リターン値	なし

sci_tei_isr	
概要	送信終了割り込み
ヘッダ	なし
宣言	static void sci_tei_isr(void)
説明	SCI.TEI 割り込み処理関数で呼び出されます。シリアル送信を禁止して、コールバック関数(SCI 送信完了)を呼び出します。
引数	なし
リターン値	なし

sci_rxi_isr	
概要	受信データフル割り込み
ヘッダ	なし
宣言	static void sci_rxi_isr(void)
説明	SCI.RXI 割り込み処理関数から呼び出されます。受信データを格納します。最終データを受信すると、シリアル受信を禁止して、コールバック関数(SCI 受信完了)を呼び出します。
引数	なし
リターン値	なし

sci_eri_isr	
概要	受信エラー割り込み
ヘッダ	なし
宣言	static void sci_eri_isr(void)
説明	SCI.ERI 割り込み処理関数から呼び出されます。シリアル受信を禁止して、コールバック関数(SCI 受信エラー)を呼び出します。
引数	なし
リターン値	なし

Excep_SCIn_ERIn	
概要	SCI.ERI 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCIn_ERIn(void)
説明	受信エラー割り込み処理を行います。
引数	なし
リターン値	なし

Excep_SCIn_RXIn

概要	SCI.RXI 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCIn_RXIn(void)
説明	受信データフル割り込み処理を行います。
引数	なし
リターン値	なし

Excep_SCIn_TXIn

概要	SCI.TXI 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCIn_TXIn(void)
説明	送信データエンプティ割り込み処理を行います。
引数	なし
リターン値	なし

Excep_SCIn_TElIn

概要	SCI.TEI 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCIn_TElIn(void)
説明	送信終了割り込み処理を行います。
引数	なし
リターン値	なし

5.9 フローチャート

5.9.1 メイン処理

図 5.5にメイン処理のフローチャートを示します。

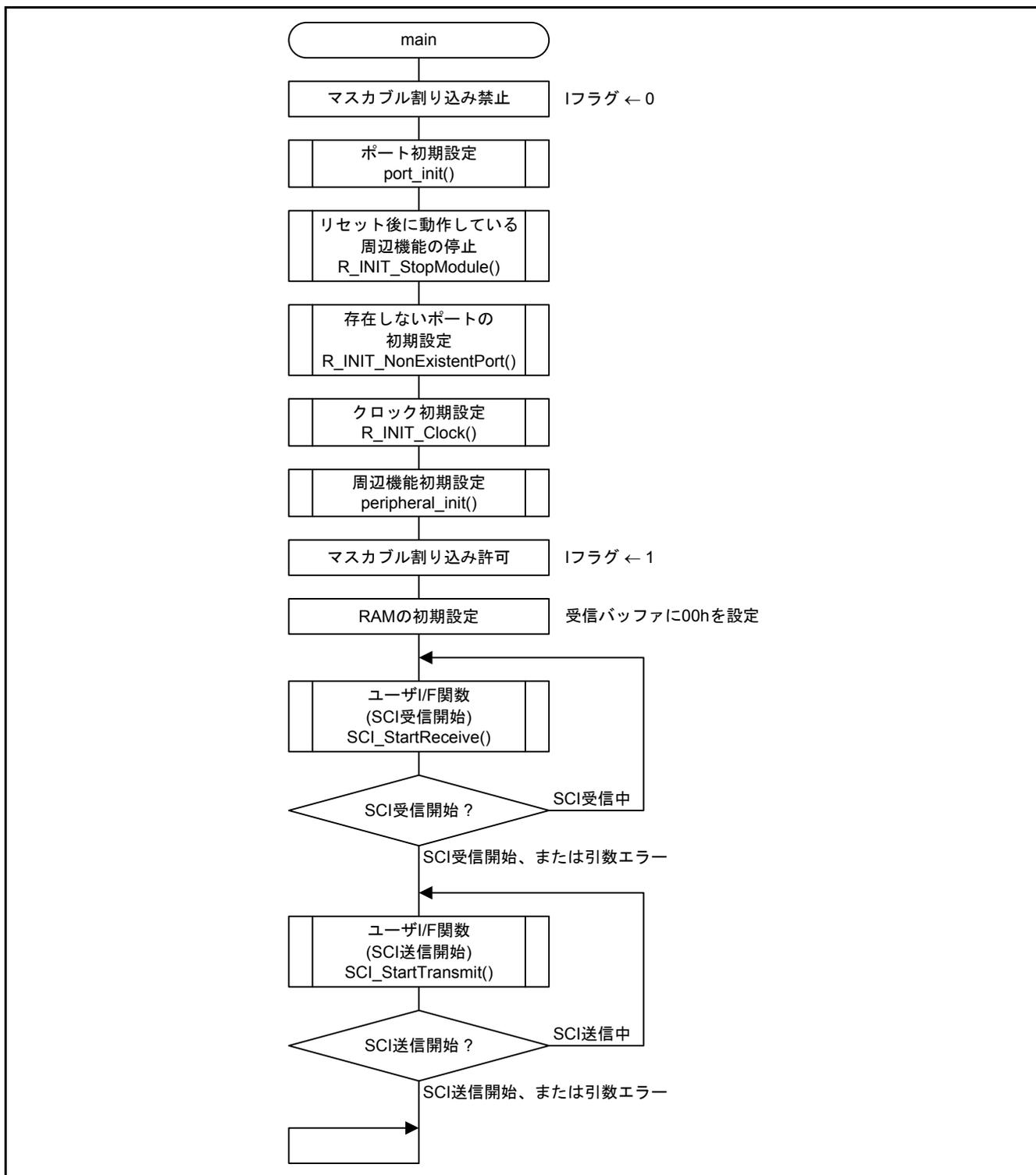


図5.5 メイン処理

5.9.2 ポート初期設定

図 5.6にポート初期設定のフローチャートを示します。

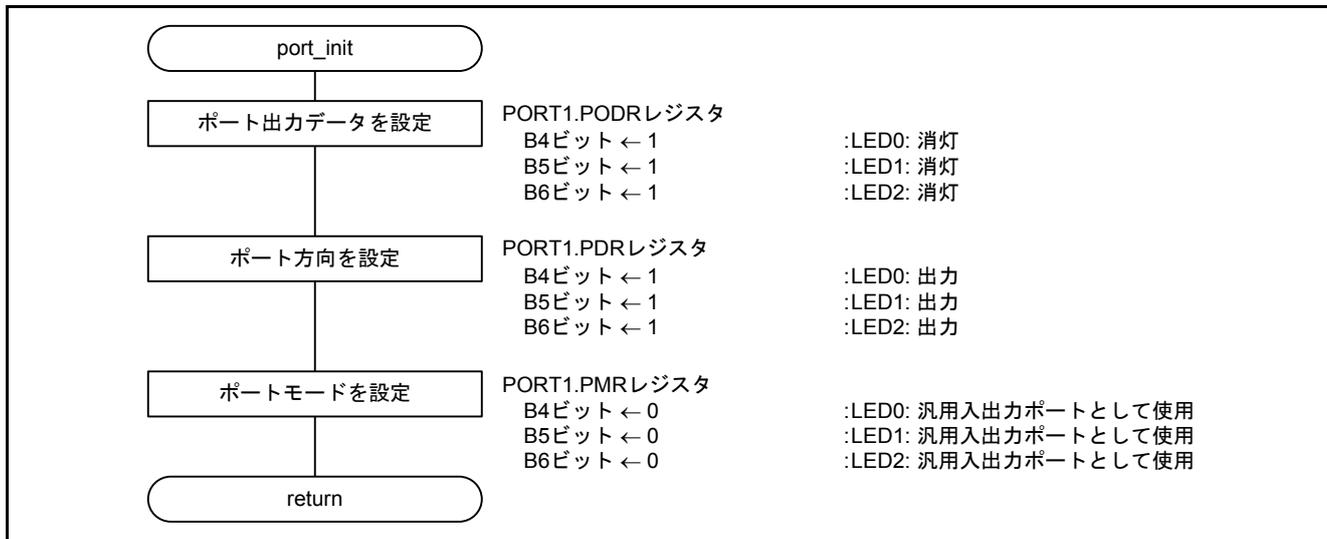


図5.6 ポート初期設定

5.9.3 周辺機能初期設定

図 5.7に周辺機能初期設定のフローチャートを示します。

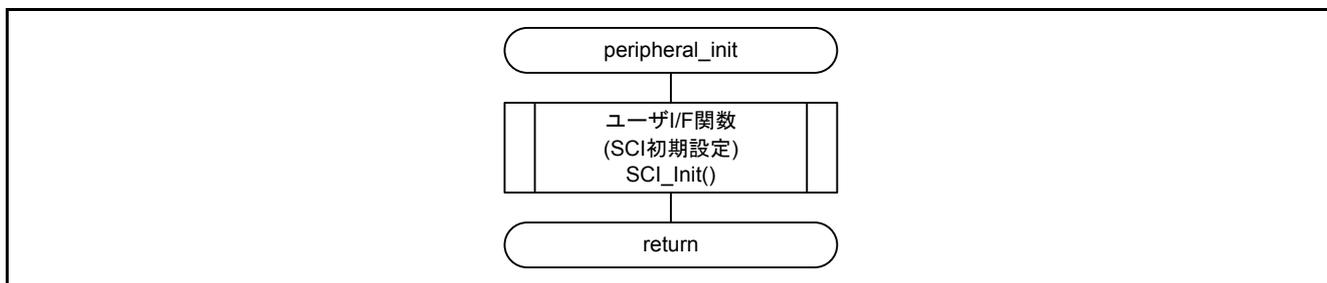


図5.7 周辺機能初期設定

5.9.4 コールバック関数(SCI 送信完了)

図 5.8にコールバック関数(SCI 送信完了)のフローチャートを示します。

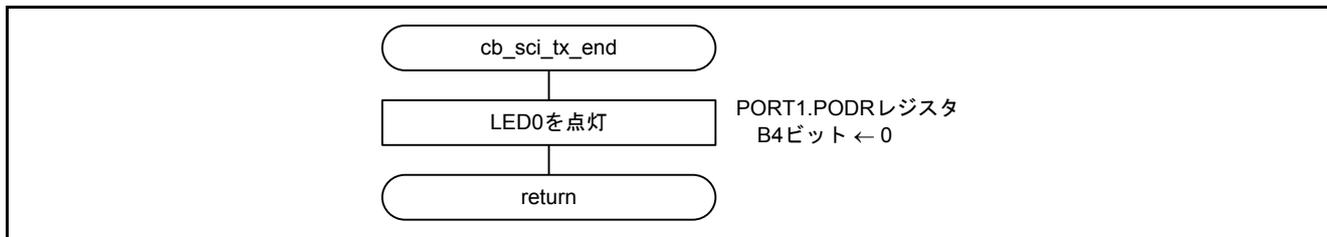


図5.8 コールバック関数(SCI 送信完了)

5.9.5 コールバック関数(SCI 受信完了)

図 5.9にコールバック関数(SCI 受信完了)のフローチャートを示します。

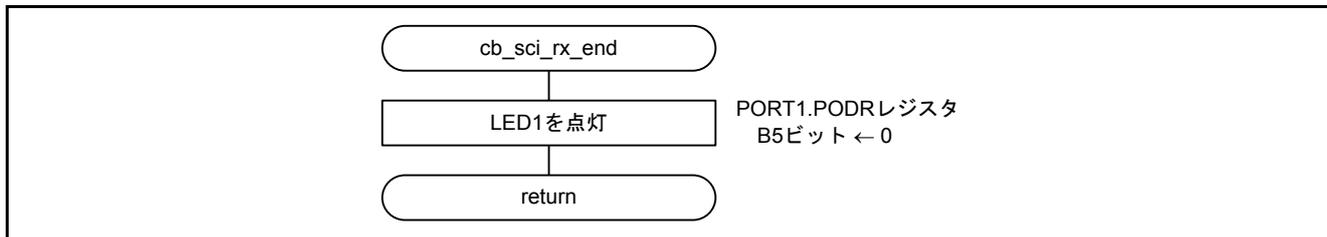


図5.9 コールバック関数(SCI 受信完了)

5.9.6 コールバック関数(SCI 受信エラー)

図 5.10にコールバック関数(SCI 受信エラー)のフローチャートを示します。

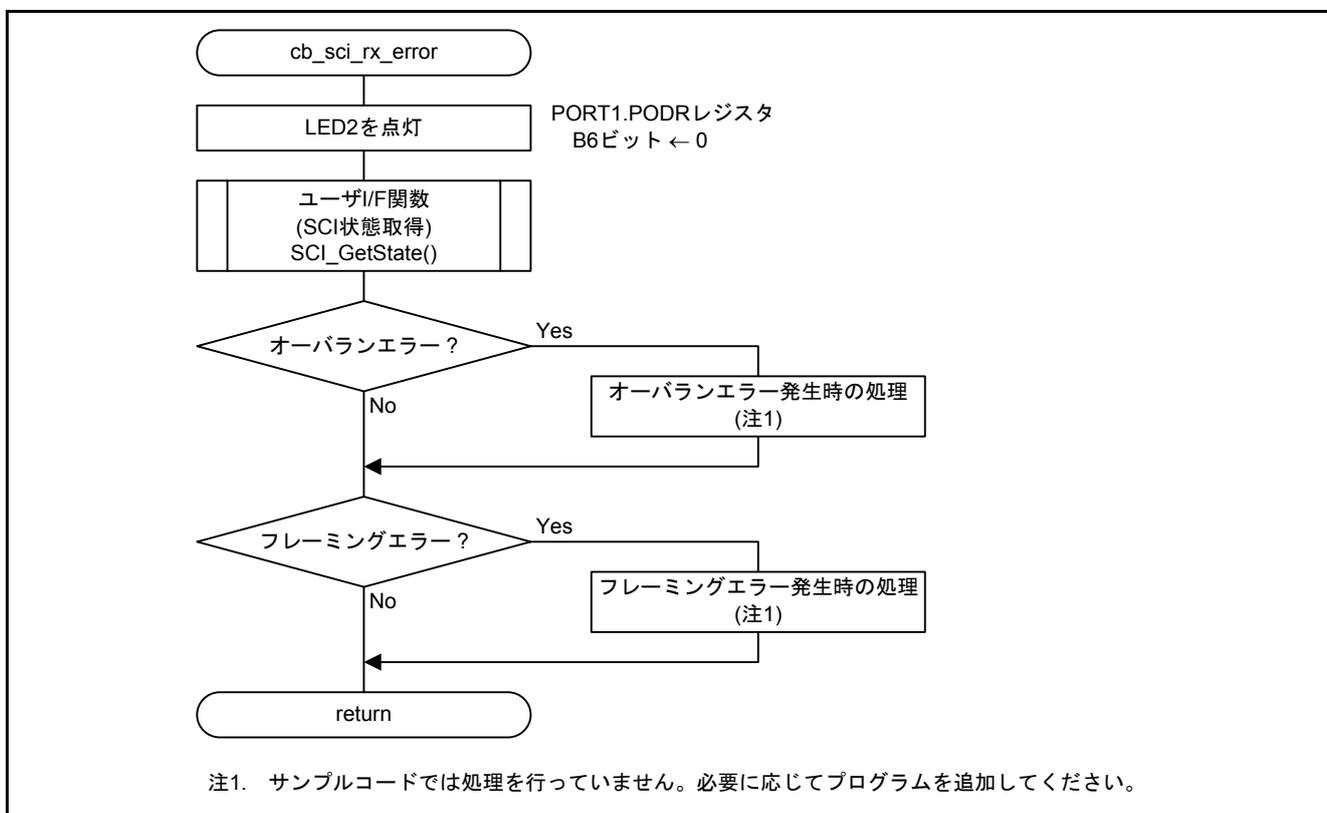


図5.10 コールバック関数(SCI 受信エラー)

5.9.7 ユーザ I/F 関数(SCI 初期設定)

図 5.11、図 5.12にユーザ I/F 関数(SCI 初期設定)のフローチャートを示します。

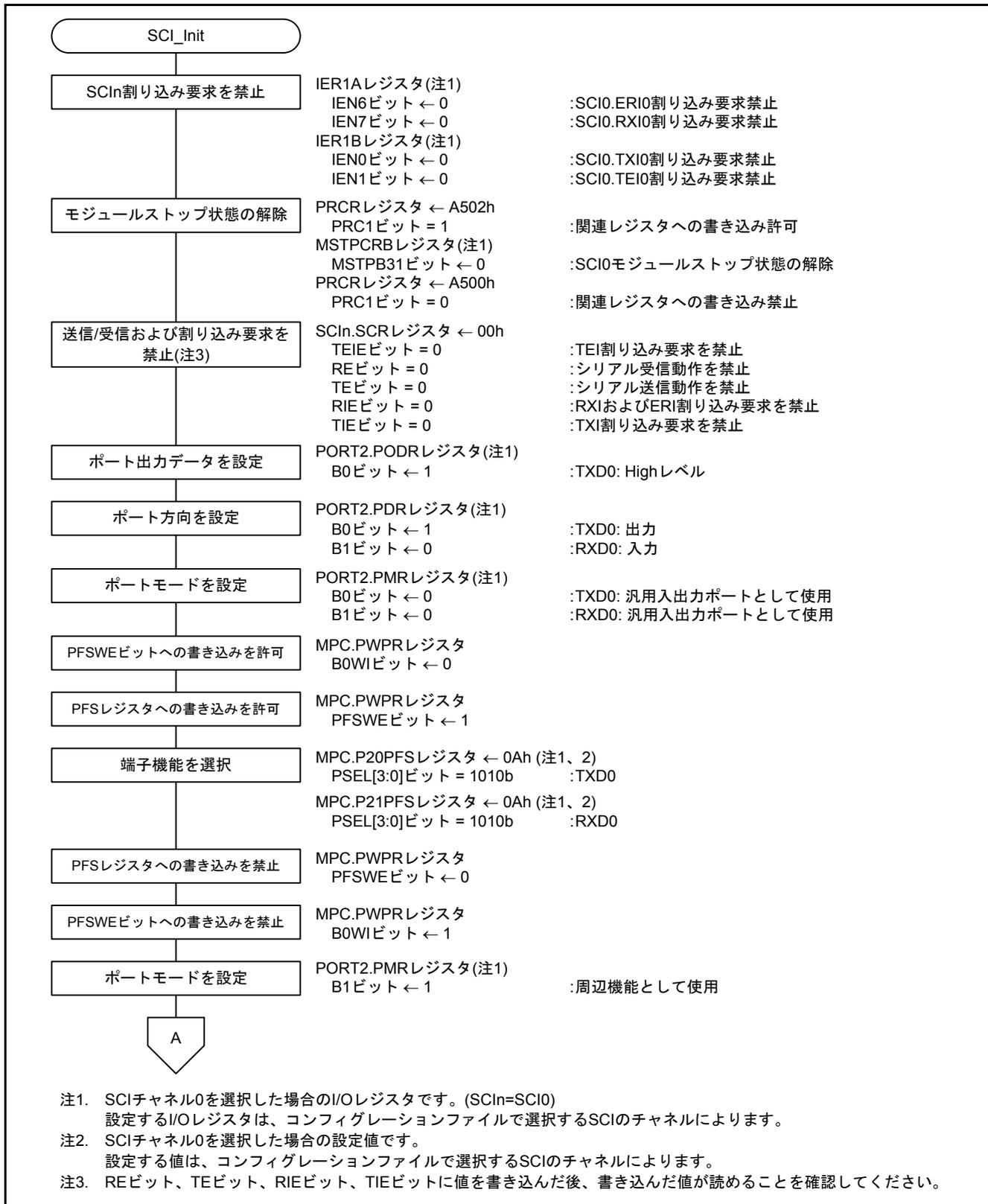


図5.11 ユーザ I/F 関数(SCI 初期設定) (1/2)

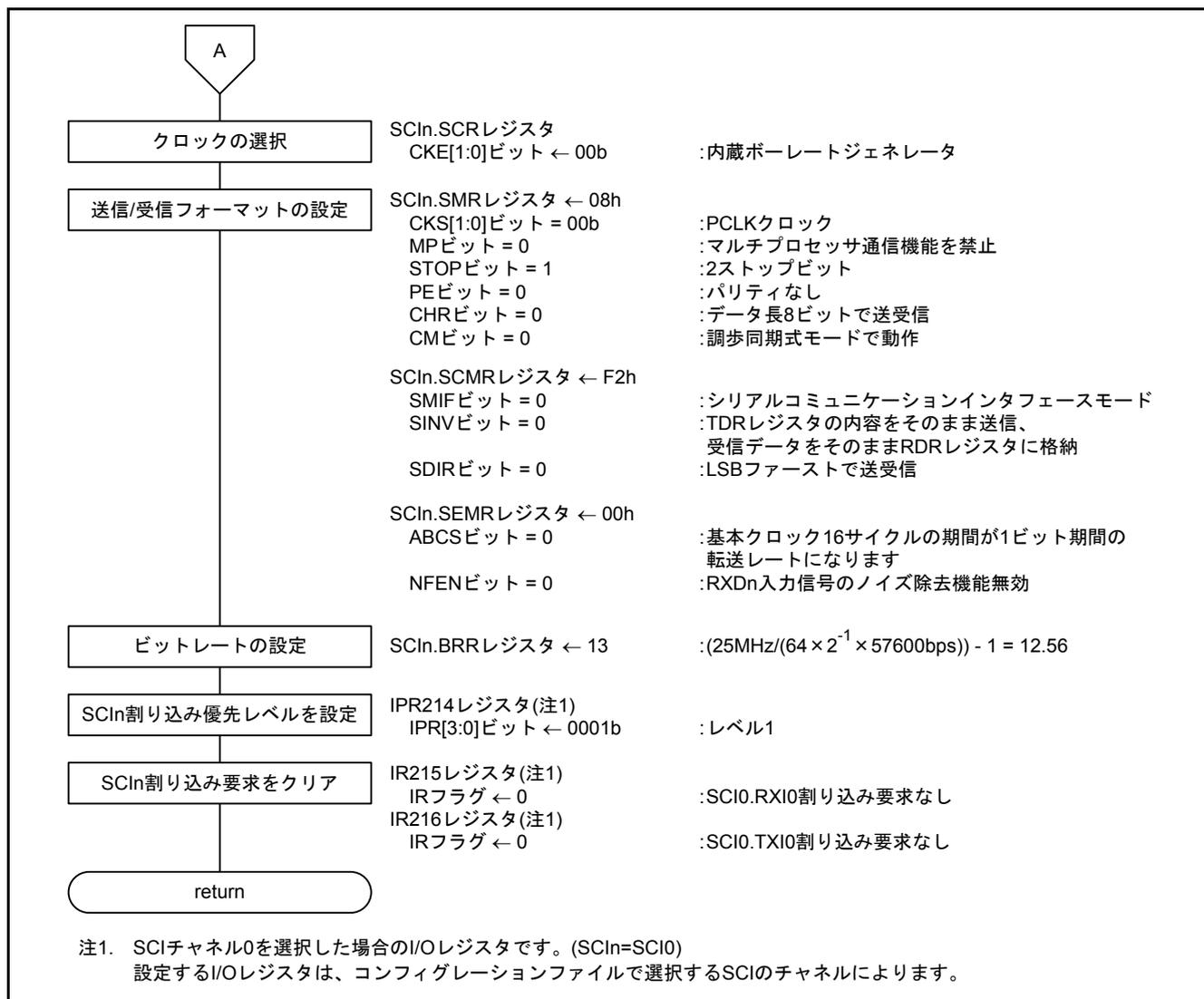


図5.12 ユーザ I/F 関数(SCI 初期設定) (2/2)

5.9.8 ユーザ I/F 関数(SCI 受信開始)

図 5.13にユーザ I/F 関数(SCI 受信開始)のフローチャートを示します。

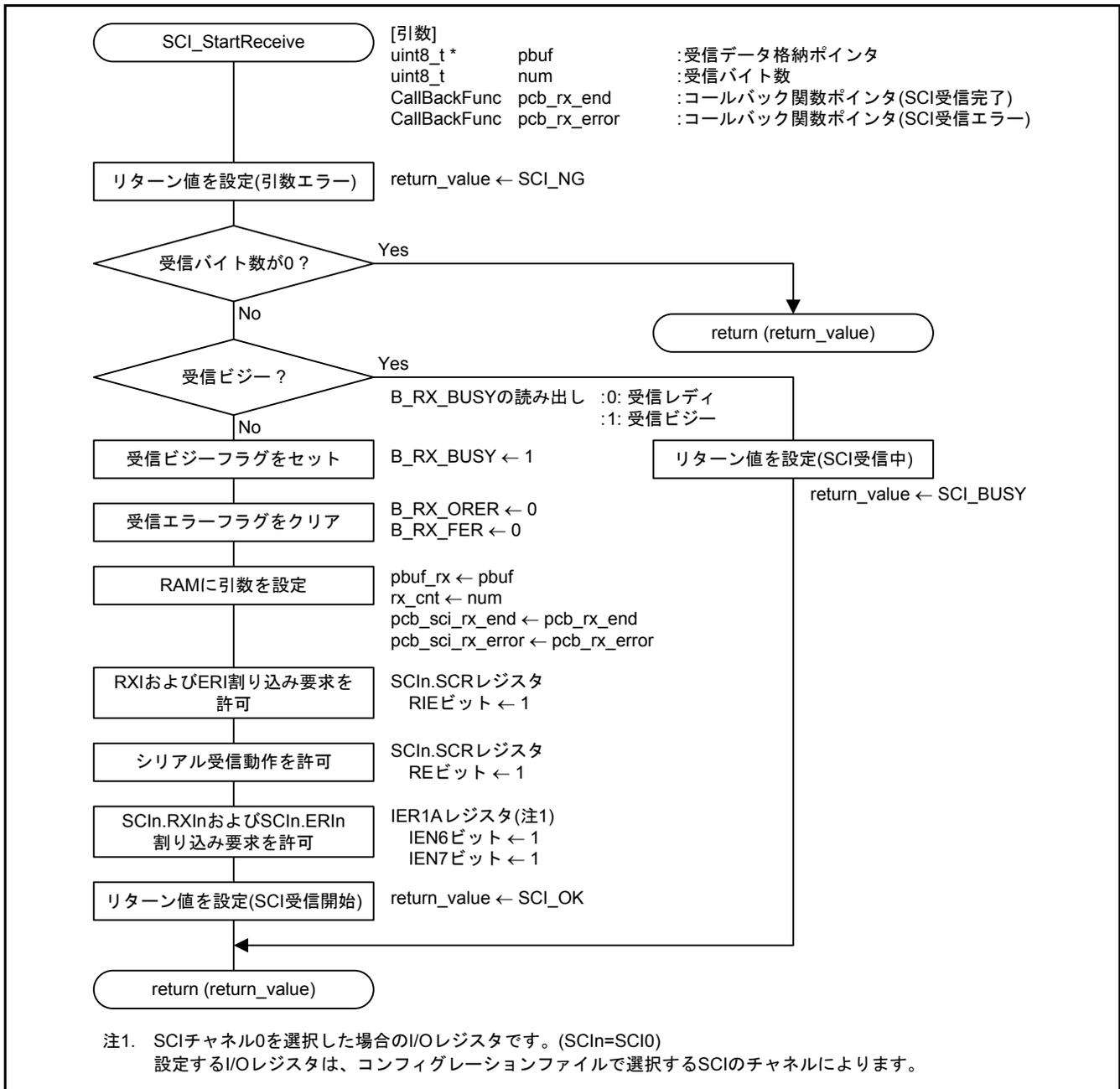


図5.13 ユーザ I/F 関数(SCI 受信開始)

5.9.9 ユーザ I/F 関数(SCI 送信開始)

図 5.14にユーザ I/F 関数(SCI 送信開始)のフローチャートを示します。

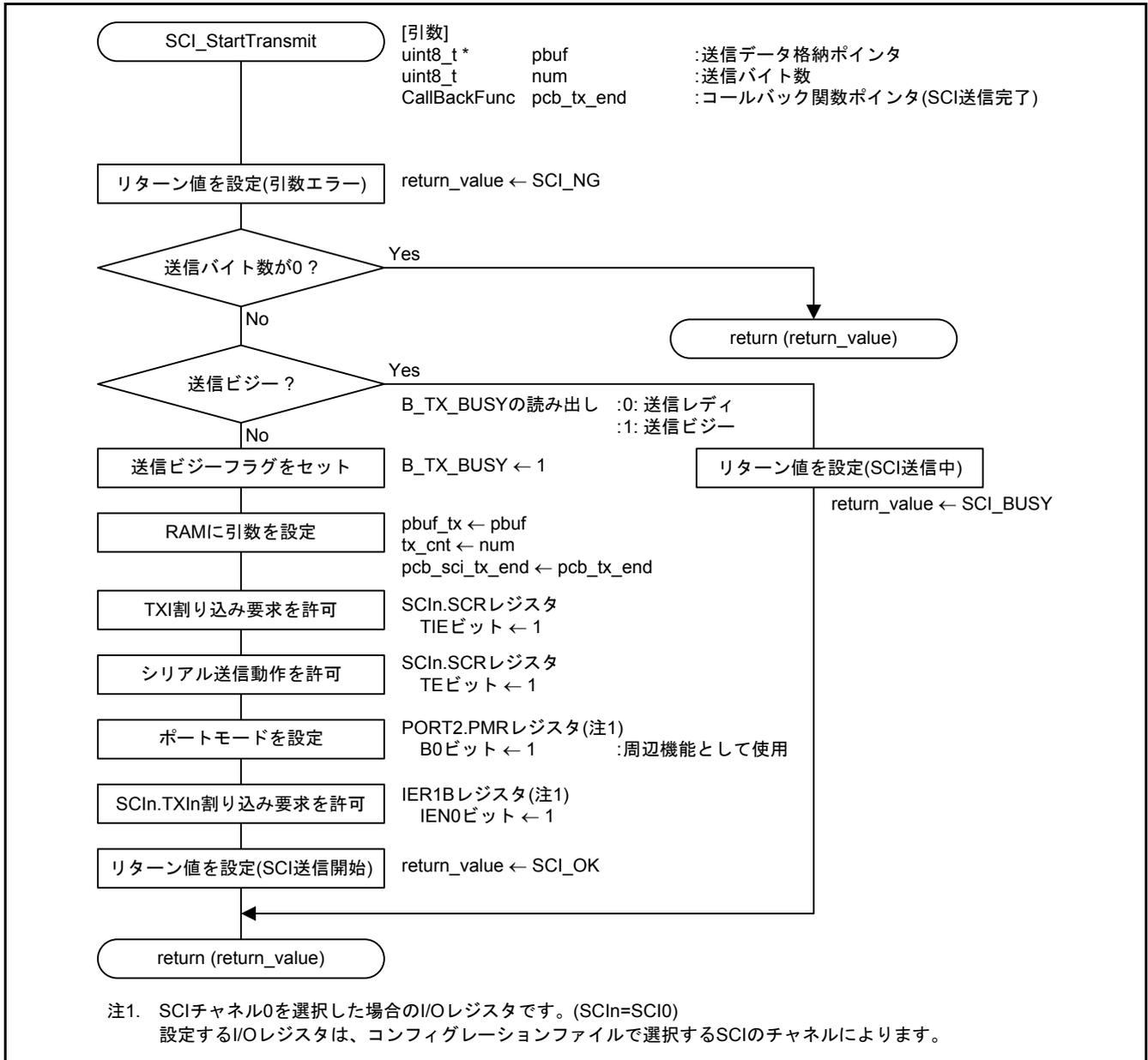


図5.14 ユーザ I/F 関数(SCI 送信開始)

5.9.10 ユーザ I/F 関数(SCI 状態取得)

図 5.15にユーザ I/F 関数(SCI 状態取得)のフローチャートを示します。

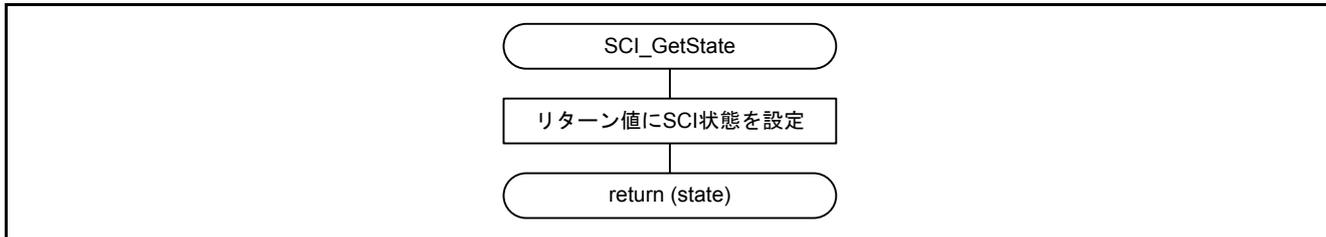


図5.15 ユーザ I/F 関数(SCI 状態取得)

5.9.11 送信データエンプティ割り込み

図 5.16に送信データエンプティ割り込みのフローチャートを示します。

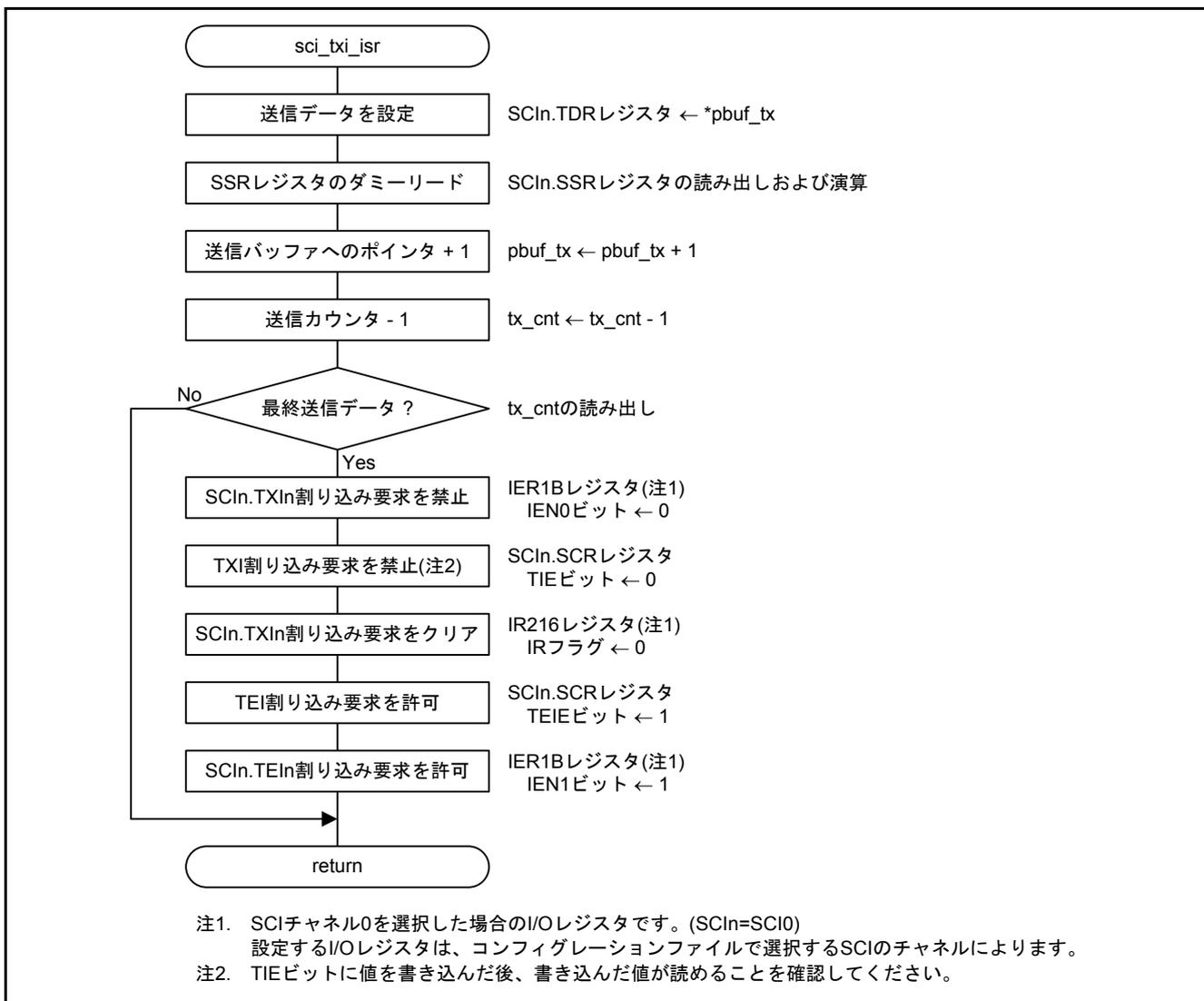


図5.16 送信データエンプティ割り込み

5.9.12 送信終了割り込み

図 5.17に送信終了割り込みのフローチャートを示します。

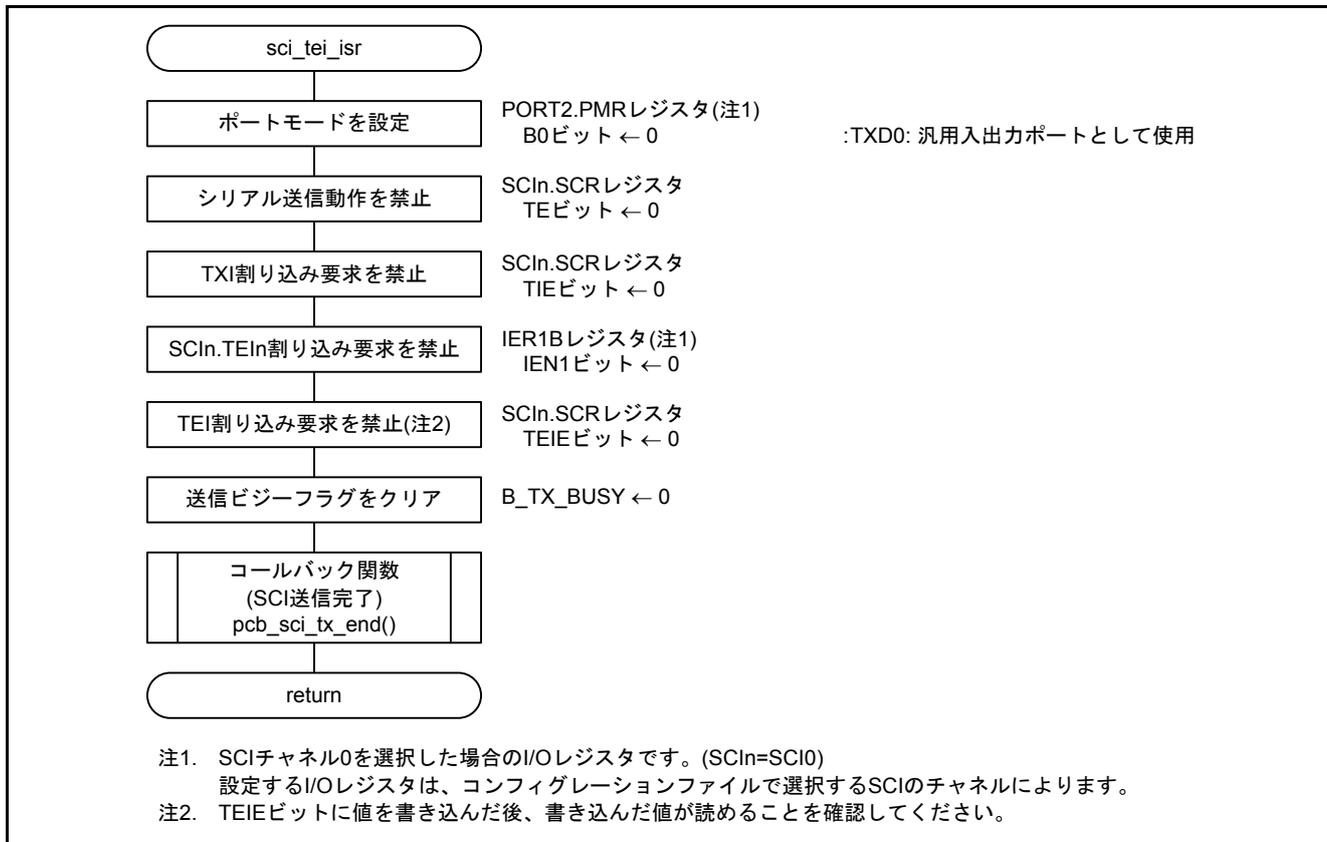


図5.17 送信終了割り込み

5.9.13 受信データフル割り込み

図 5.18に受信データフル割り込みのフローチャートを示します。

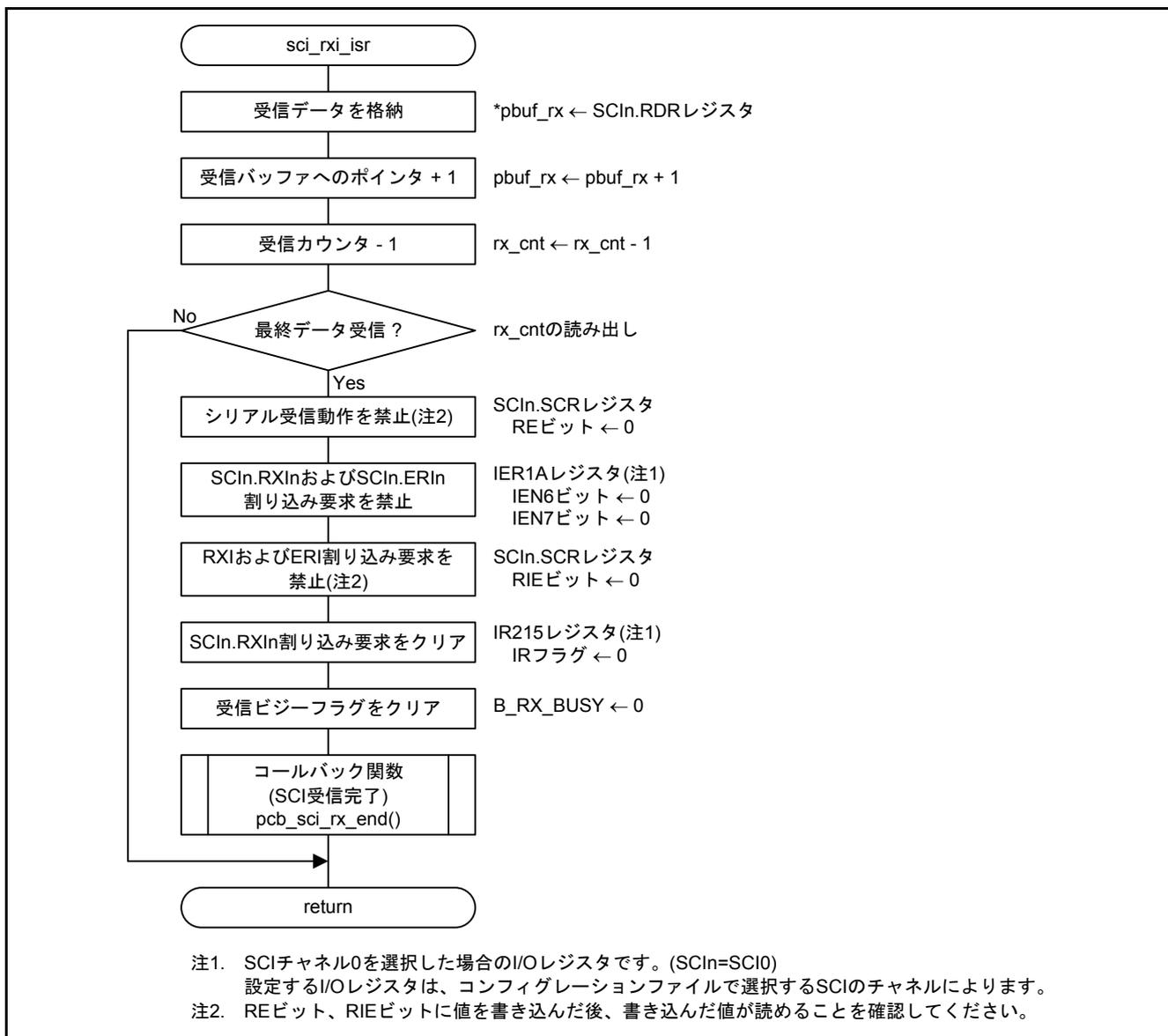


図5.18 受信データフル割り込み

5.9.14 受信エラー割り込み

図 5.19に受信エラー割り込みのフローチャートを示します。

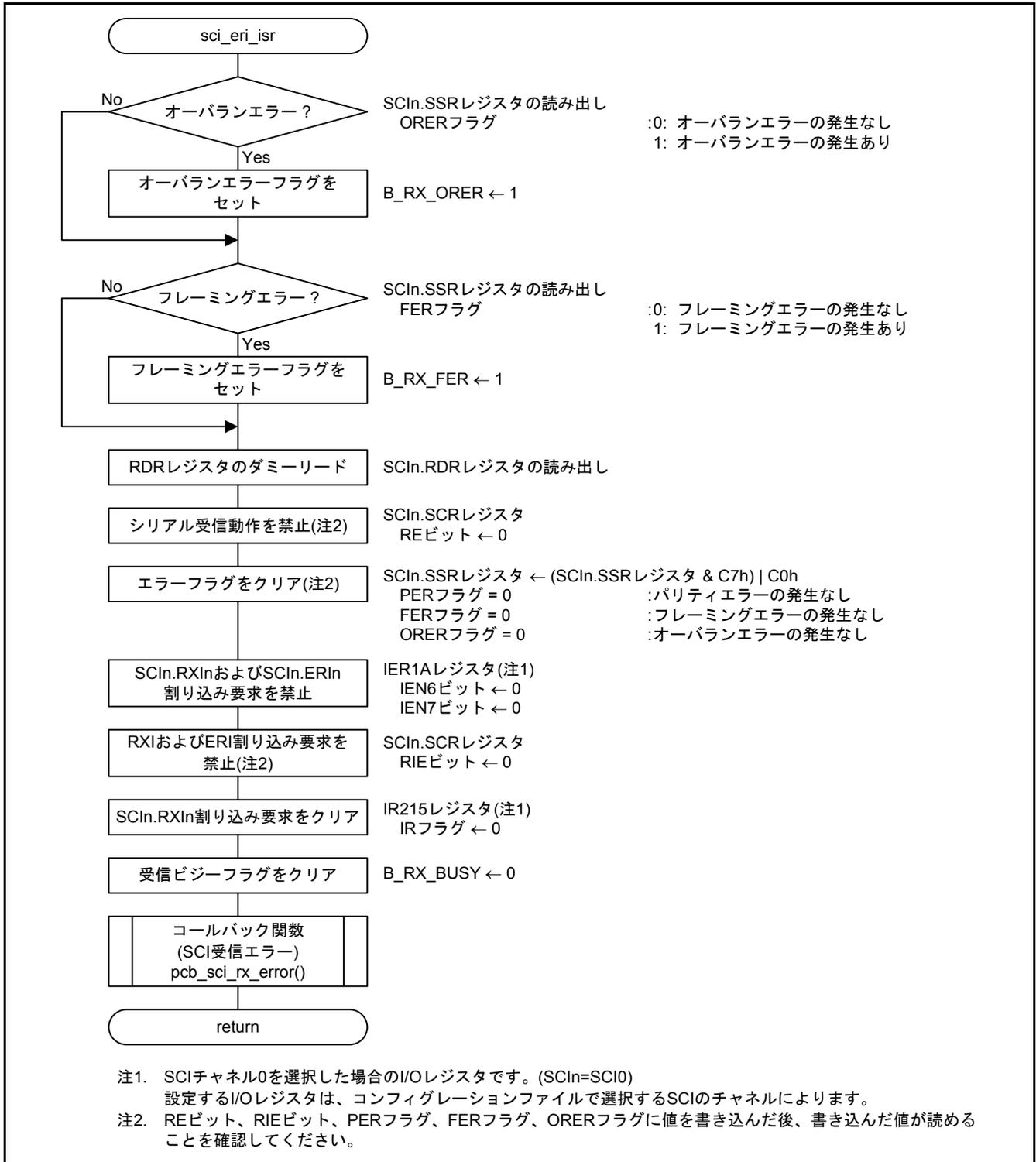


図5.19 受信エラー割り込み

5.9.15 SCI.ERI 割り込み処理

図 5.20にSCI.ERI 割り込み処理のフローチャートを示します。

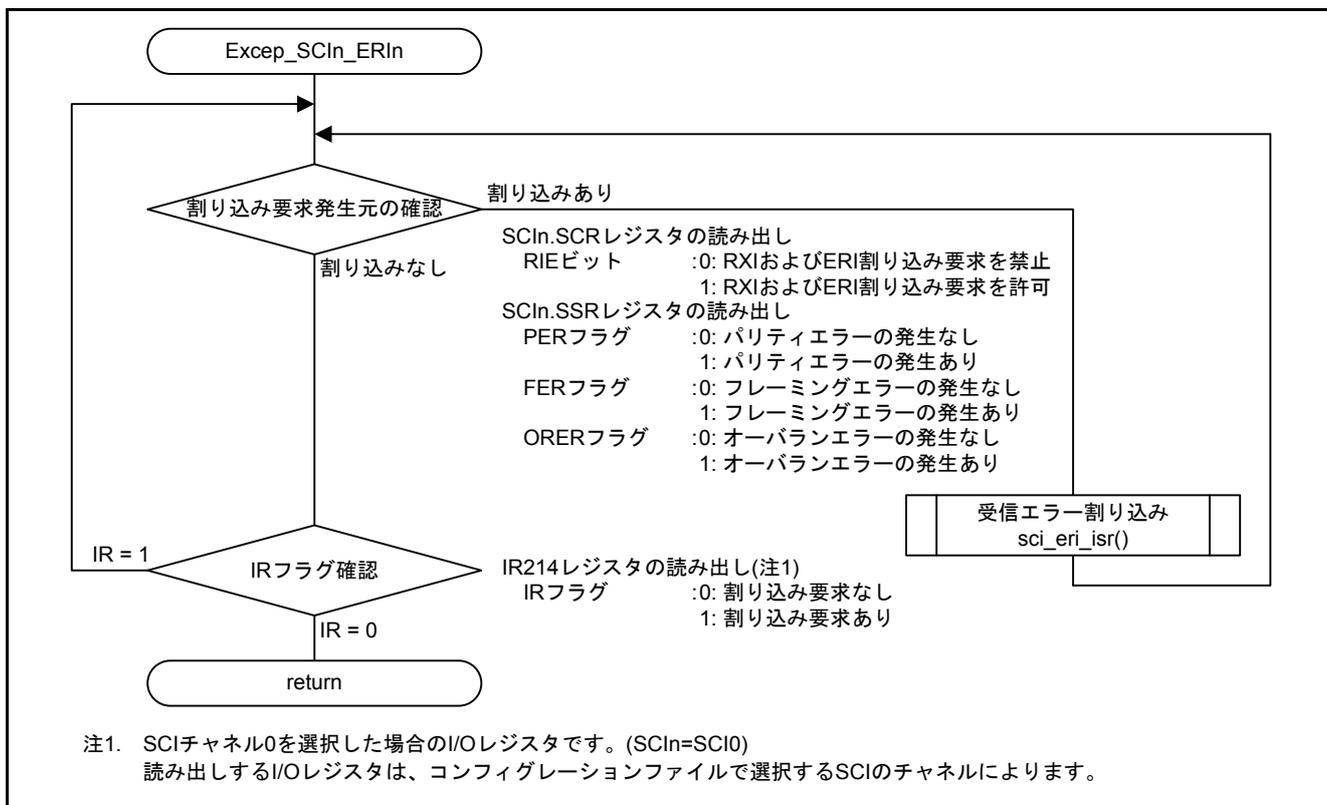


図5.20 SCI.ERI 割り込み処理

5.9.16 SCI.RXI 割り込み処理

図 5.21にSCI.RXI 割り込み処理のフローチャートを示します。

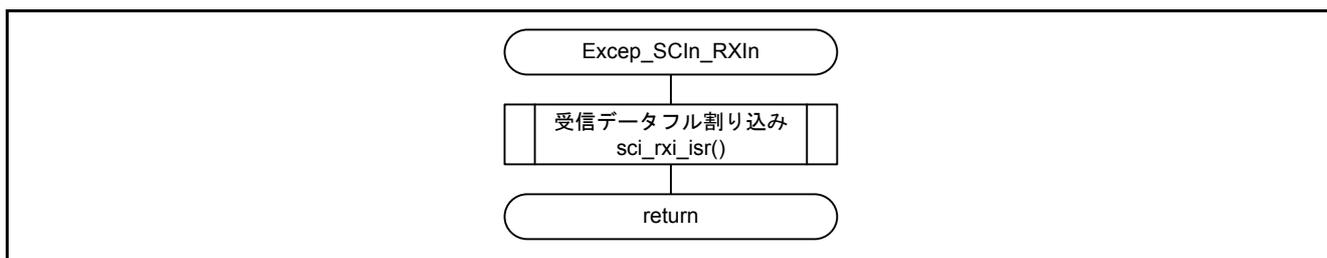


図5.21 SCI.RXI 割り込み処理

5.9.17 SCI.TXI 割り込み処理

図 5.22にSCI.TXI 割り込み処理のフローチャートを示します。

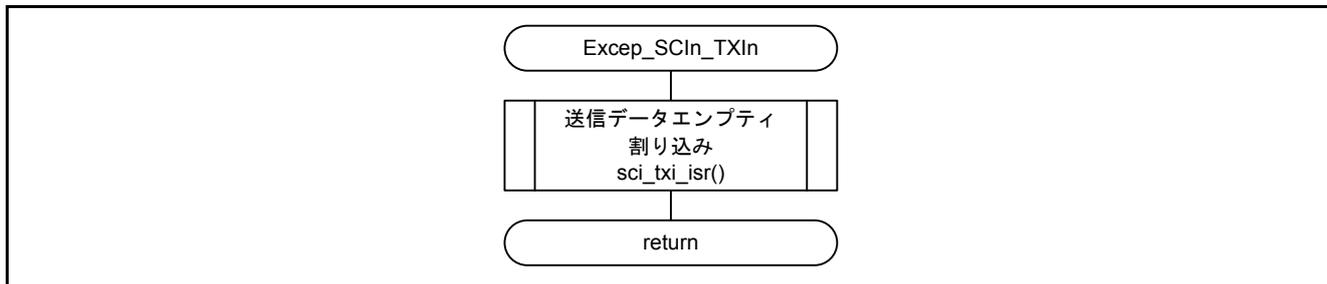


図5.22 SCI.TXI 割り込み処理

5.9.18 SCI.TEI 割り込み処理

図 5.23にSCI.TEI 割り込み処理のフローチャートを示します。

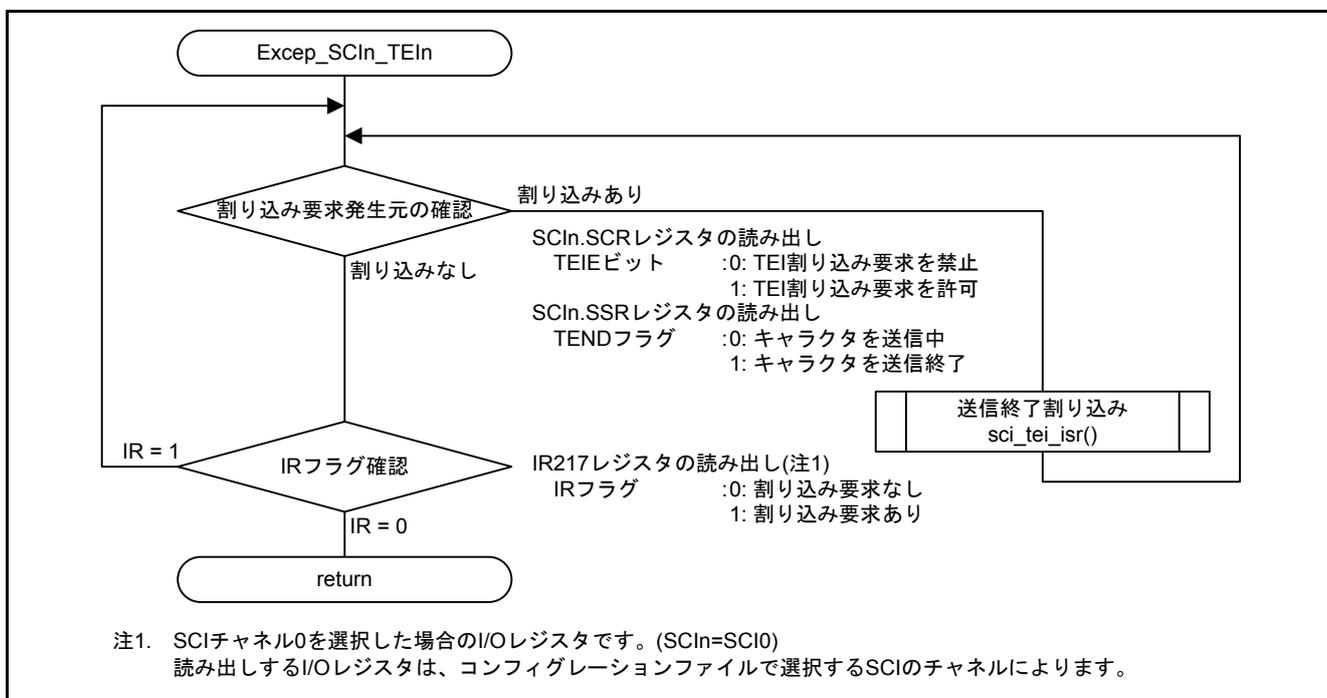


図5.23 SCI.TEI 割り込み処理

6. RX21A、RX220 グループ 初期設定例 アプリケーションノートとの組み合わせ方

本アプリケーションノートのサンプルコードは、RX210 グループで動作することを確認しています。RX21A グループや RX220 グループで動作させるには、それぞれの初期設定例のアプリケーションノートと組み合わせてください。

手順は、初期設定例のアプリケーションノート 「5. RX210 グループのアプリケーションノートを RX21A グループに適用する方法」、 「4. RX210 グループのアプリケーションノートを RX220 グループに適用する方法」を参照ください。

注：RX21A では、SCI0、12 は存在しません。SCI1、5、6、8、9 を使用してください。
RX220 では、SCI0、8、12 は存在しません。SCI1、5、6、9 を使用してください。

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

RX210グループ ユーザーズマニュアル ハードウェア編 Rev.1.50 (R01UH0037JJ)

RX21A グループ ユーザーズマニュアル ハードウェア編 Rev.1.00 (R01UH0251JJ)

RX220 グループ ユーザーズマニュアル ハードウェア編 Rev.1.10 (R01UH0292JJ)

(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。)

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、RX21A、RX220 グループ SCI を用いた調歩同期式通信
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.03.01	—	初版発行
1.01	2014.07.01	1	対象デバイスに RX21A、RX220 グループを追加
		4	関連アプリケーションノートに RX21A、RX220 グループ 初期設定例のアプリケーションノートを追加
		18	参照するアプリケーションノートを各グループのアプリケーションノート初期設定例に変更
		36	RX21A、RX220 グループ 初期設定例と組み合わせる方法の参照先を追加
		37	参考ドキュメントに RX21A、RX220 グループのユーザーズマニュアルを追加

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うと、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い合わせください。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/>