RX64M、RX71Mグループ
SCIg ビットレートモジュレーション機能の使い方

R01AN2085JJ0110
Rev.1.10
2020.12.21

要旨

本アプリケーションノートでは、RX64M、RX71Mグループのビットレートモジュレーション機能を使用した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。

ビットレートモジュレーション機能は、内蔵ボーレートジェネレータにより生成されるビットレートを補正し、誤差を低減します。

対象デバイス

- RX64M グループ
  - ・RX64M グループ　177、176 ピン版　　ROM 容量：2MB～4MB
  - ・RX64M グループ　145、144 ピン版　　ROM 容量：2MB～4MB
  - ・RX64M グループ　100 ピン版　　ROM 容量：2MB～4MB

- RX71M グループ
  - ・RX71M グループ　177、176 ピン版　　ROM 容量：2MB～4MB
  - ・RX71M グループ　145、144 ピン版　　ROM 容量：2MB～4MB
  - ・RX71M グループ　100 ピン版　　ROM 容量：2MB～4MB

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

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います。この送受信ではビットレートモジュレーション機能を使用します。

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。

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 (チャネル 7)	調歩同期式シリアル送受信
I/O ポート	LED 点灯

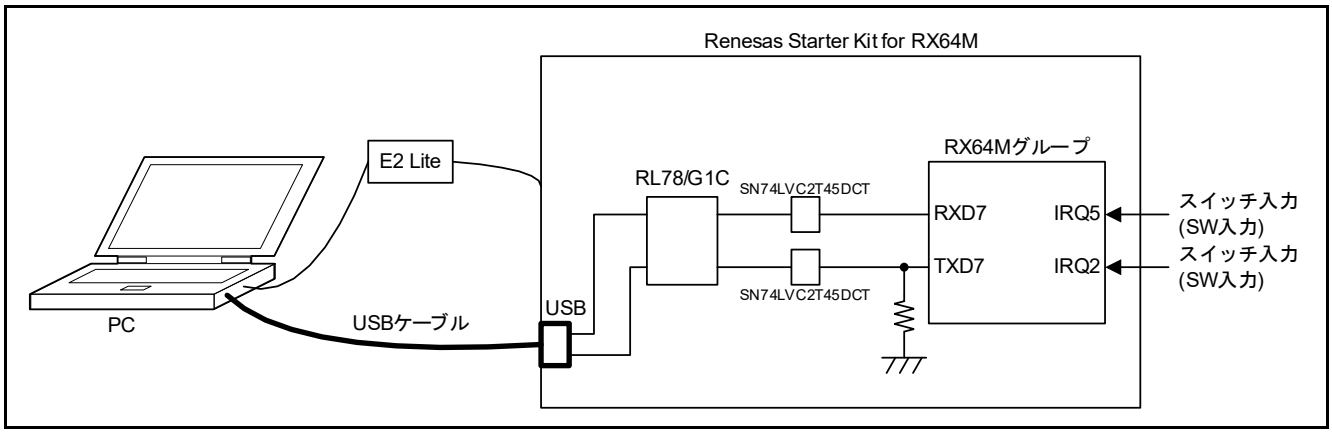


図1.1　使用例

1.1　USB シリアル変換

Renesas Starter Kit+ for RX64M は製品出荷時、RX64M マイクロコントローラのシリアルポート SCI7 が RL78/G1C マイクロコントローラのシリアルポートに接続されており、仮想 COM ポートとして使用できます。本アプリケーションノートでは、この仮想 COM ポートを使用して、PC との通信を行っています。

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	R5F564MLCDFC(RX64M グループ)
動作周波数	メインクロック:24MHz PLL:240MHz(メインクロック 1 分周 10 逓倍) システムクロック(ICLK):120MHz(PLL2 分周) （注 1） 周辺モジュールクロック B(PCLKB):60MHz(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 e² studio Version: 2020-10
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 V3.02.00（注 2）
	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
iodefine.h のバージョン	0.9
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	Version1.10
使用ボード	Renesas Starter Kit+ for RX64M(製品型名:R0K50564MSxxxBE)

注1. RX71M において ICLK を 120MHz より速くする場合は、MEMWAIT レジスタの変更が必要となります。詳細は、RX71M グループ ユーザーズマニュアル ハードウェア編を参照してください。

注2. 元のプロジェクトで指定するツールチェーン(C コンパイラ）と同一のバージョンがインポートする先にない場合は、ツールチェーンが選択されない状態になり、エラーが発生します。プロジェクトの設定画面でツールチェーンの選択状態を確認してください。

選択方法は、FAQ 3000404 を参照してください。

FAQ 3000404 :インポートしたプロジェクトをビルドすると「PATH でプログラム "make" が見つかりません」エラーになる(e² studio)

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。

RX64M グループ　初期設定例　Rev.1.00(R01AN1918JJ0100_RX64M)

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。

最新版がある場合は、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。

表4.1 使用端子と機能

端子名	入出力	内容
P03	出力	LED0 出力(SCI 送信完了)
P05	出力	LED1 出力(SCI 受信完了)
P26	出力	LED2 出力(SCI 受信エラー)
P92/RXD7	入力	SCI7 の受信データ入力端子
P90/TXD7	出力	SCI7 の送信データ出力端子

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、送信動作と受信動作を許可します。

ユーザ I/F 関数(SCI 送信開始)を呼び出すと、送信データエンプティ割り込み要求を許可します。指定バイト数の送信を完了したとき、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。

ユーザ I/F 関数(SCI 受信開始)を呼び出すと、受信データフル割り込み要求と受信エラー割り込み要求を許可します。指定バイト数の受信を完了したとき、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 の送信動作と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。

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それぞれ示します。

<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 (60MHz)
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

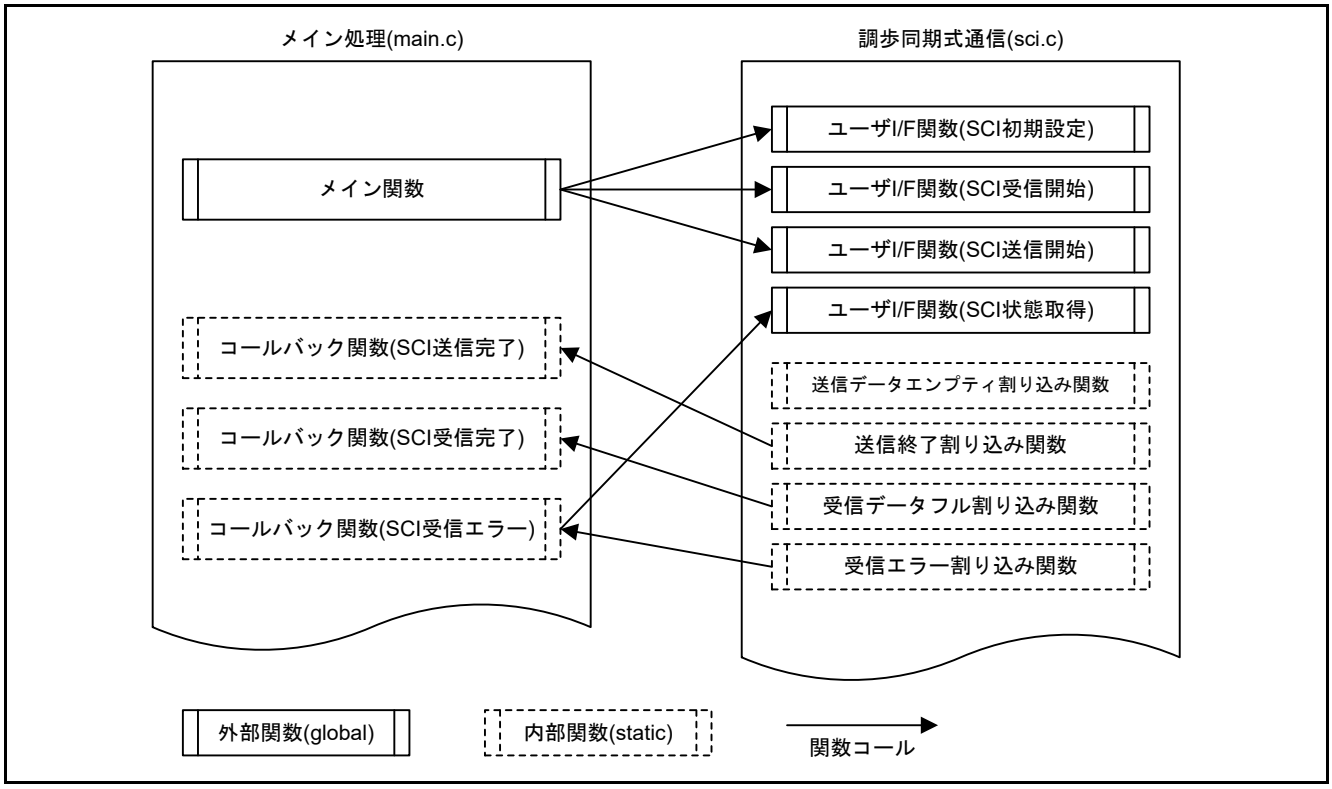


図 5.1　ソフトウェア構成

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、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。（本サンプルコードでは、SCR.TE ビットを“1”にすると同時に、SCR.RE ビットも“1”に設定しています。）

(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”にした後、TXI 割り込みの IERm.IENj ビットを“1”(割り込み要求許可)にします。このとき、(1)の処理によって既に TXI 割り込みの IRi.IR フラグは“1” (割り込み要求あり)になっていますので、TXI 割り込みが発生します。

(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。TDR レジスタから TSR レジスタに送信バッファの値が転送されると、TXI 割り込みの IRi.IR フラグが“1” (割り込み要求あり)になり、割り込みが発生します。最終データを書くまでこの処理を繰り返します。最終データを書いたら、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、TEIE ビットを“0”(TEI 割り込み要求を禁止)にします。その後、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。

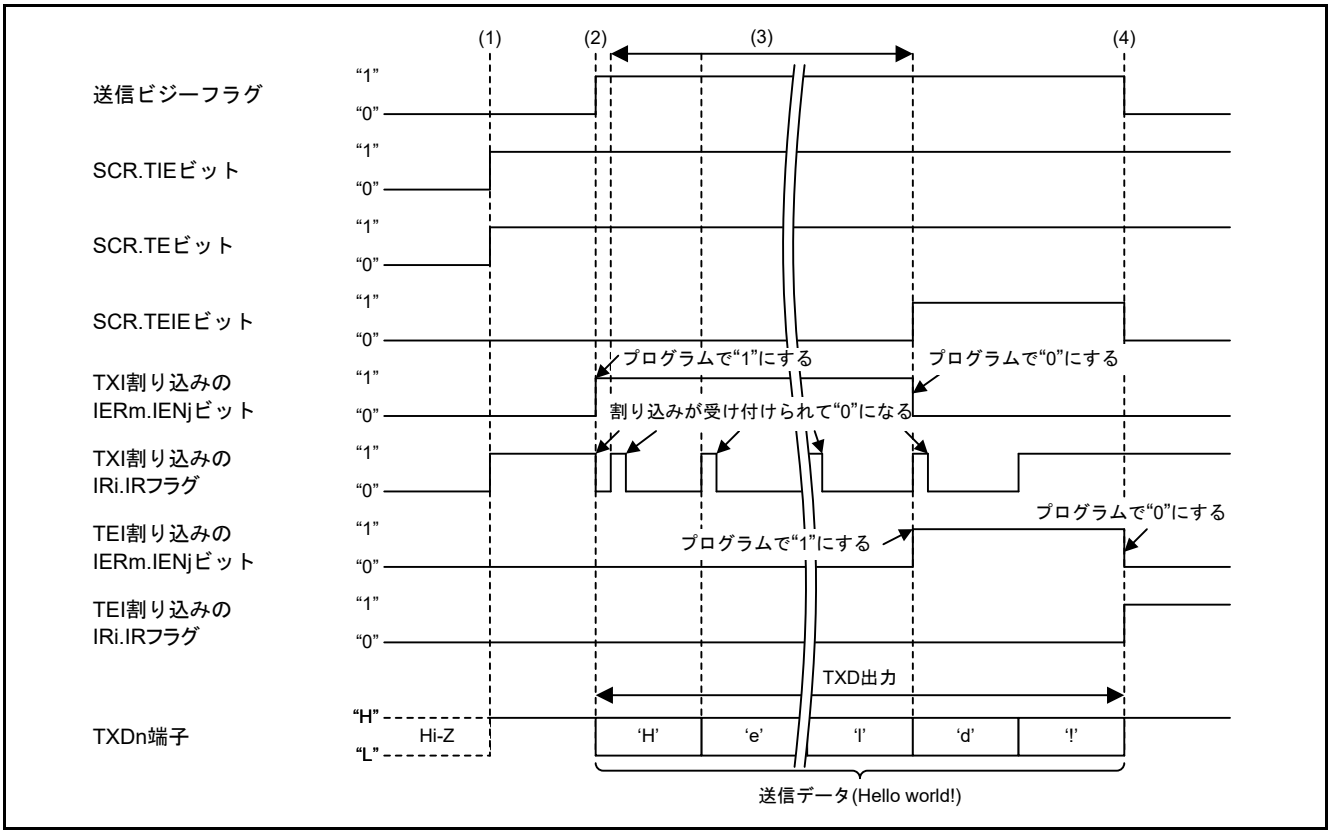


図 5.2　シリアル送信のタイミング図

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、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)にします。（本サンプルコードでは、SCR.RE ビットを“1”にすると同時に、SCR.TE ビットも“1”に設定しています。）

(2)　受信開始(注 1)

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。RXI および ERI 割り込みの IERm.IENj 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(注 2)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”、TE 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”、TIE ビットを“0”、TEIE 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、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。

注 1. ユーザ I/F 関数(SCI 初期設定)を呼び出した後、かつユーザ I/F 関数(SCI 受信開始)を呼び出す前に、2 バイト以上データを受信すると、オーバランエラーが発生します。

注 2. 受信エラー発生後に、再度シリアル送受信を開始する場合は、ユーザ I/F 関数(SCI 初期設定)を呼び出した後に、ユーザ I/F 関数(SCI 受信開始、SCI 送信開始)を呼び出してください。

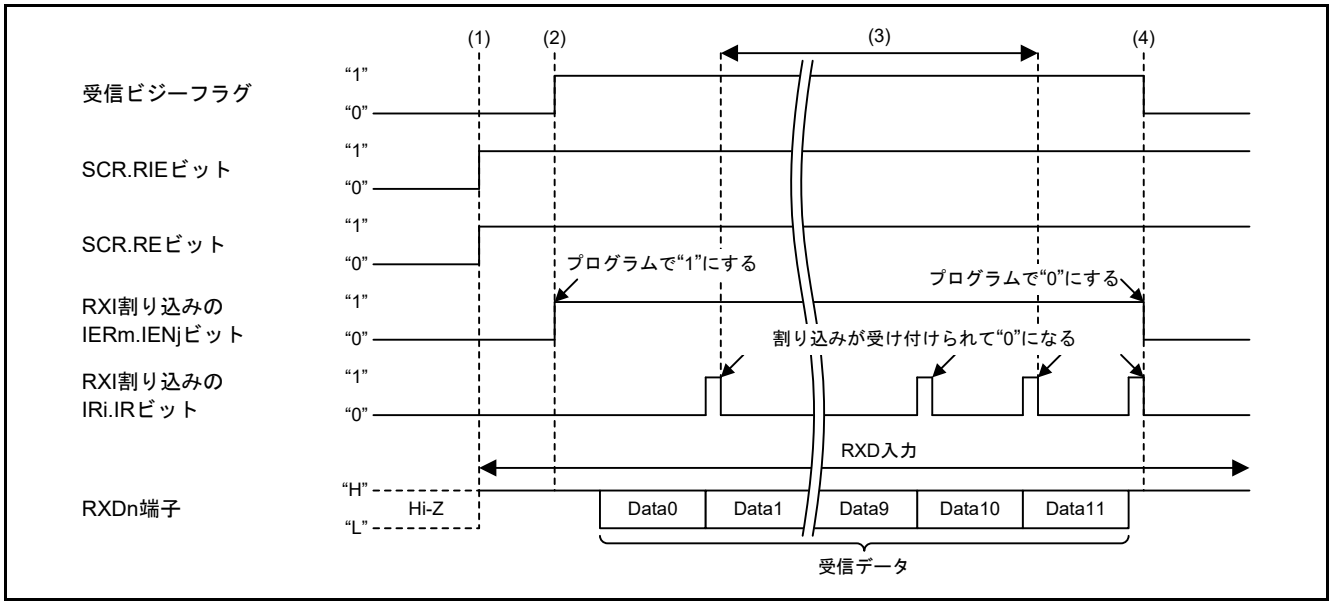


図 5.3　シリアル受信のタイミング図

5.2　ビットレートモジュレーション機能

5.2.1　ビットレートモジュレーション機能について

ビットレートモジュレーション機能(注)は、SMR.CKS[1:0]ビットで指定された内部クロックを、MDDR レジスタで指定した値によって平均的にイネーブルし、ビットレートを補正します。

MDDR = 160 に設定した場合、平均的に 256 クロック中 160 クロックを有効にします（96 クロックは無効）。

注.　ビットレートモジュレーション機能に関する詳細は、RX64M グループ ユーザーズマニュアル ハードウェア編 Rev.1.00「40.9 ビットレートモジュレーション機能」に記載されています。

5.2.2　ビットレートモジュレーション機能の使用方法

SEMR.BRME ビットを 1 に設定すると、ビットレートモジュレーション機能が有効になります。その後、BRR レジスタ及び MDDR レジスタを設定します。

BRR レジスタ及び MDDR レジスタには、ビットレートの誤差がシリアル通信可能な範囲に収まるように値を設定してください。表5.1から誤差を求めることができます。(注)

表5.1　ビットレートモジュレーション機能使用時のビットレートの誤差計算式

モード	SEMR レジスタの設定		誤差
	BGDM ビット	ABCS ビット	
調歩同期式通信	0	0	誤差(%) = $\left\{\frac{PCLKB\times10^6}{B\times64\times2^{2n-1}\times(256/M)\times(N+1)}-1\right\}\times100$
	0	1	誤差(%) = $\left\{\frac{PCLKB\times10^6}{B\times32\times2^{2n-1}\times(256/M)\times(N+1)}-1\right\}\times100$
	1	0	
	1	1	誤差(%) = $\left\{\frac{PCLKB\times10^6}{B\times16\times2^{2n-1}\times(256/M)\times(N+1)}-1\right\}\times100$

B: ビットレート(bps)

M: MDDR レジスタの設定値 (128≦M≦255)

N: ボーレートジェネレータの BRR の設定値 (0≦N≦255)

PCLKB: 動作周波数(MHz)

n: SMR.CKS[1:0]ビットの設定値 (0≦n≦3)

注.　ビットレートモジュレーション機能使用時の誤差計算式の詳細は、RX64M グループ ユーザーズマニュアル ハードウェア編 Rev.1.00「40.2.12 モジュレーションデューティレジスタ」に記載されています。

5.2.3　ビットレートモジュレーション機能使用時と未使用時の比較

表5.2に転送速度 57600bps を使用する時の誤差が最小になる設定を以下に示します。

表5.2　シリアル送受信の設定及び誤差が最小になる N、M

	ビットレートモジュレーション機能	
	使用時	未使用時
転送速度[bps] (B)	57600	
動作周波数[MHz] (PCLKB)	60	
SEMR.BGDM ビットの設定	0	0
SEMR.ABCS ビットの設定	0	0
SMR.CKS[1:0]ビットの設定(n)	0	0
BRR レジスタの設定(N)	21	32
MDDR レジスタの設定(M)	173	

(a)ビットレートモジュレーション機能使用時の誤差計算

SEMR.BGDM ビット = 0、SEMR.ABCS ビット = 0 から使用する式は

$$誤差(\%) = \left\{\frac{PCLKB\times10^{6}}{B\times64\times2^{2n-1}\times(256/M)\times(N+1)}-1\right\}\times100 \quad ・・・①$$

になります。

式①の PCLKB、M、N、n にそれぞれ値を代入します。

$$誤差(\%) = \left\{\frac{60\times10^{6}}{57600\times64\times2^{2\times0-1}\times(256/173)\times(21+1)}-1\right\}\times100 \quad ・・・②$$

式②より、誤差は

$$\left\{\frac{60\times10^{6}}{57600\times64\times2^{-1}\times(256/173)\times22}-1\right\}\times100 = -0.008692886\ [\%]$$

(b)ビットレートモジュレーション機能未使用時の誤差計算

SEMR.BGDM ビット = 0、SEMR.ABCS ビット = 0 から使用する式(注)は

$$誤差(\%) = \left\{\frac{PCLKB\times10^{6}}{B\times64\times2^{2n-1}\times(N+1)}-1\right\}\times100 \quad ・・・①$$

になります。

式①の PCLKB、N、n にそれぞれ値を代入します。

$$誤差(\%) = \left\{\frac{60\times10^{6}}{57600\times64\times2^{2\times0-1}\times(32+1)}-1\right\}\times100 \quad ・・・②$$

式②より、誤差は

$$\left\{\frac{60\times10^{6}}{57600\times64\times2^{-1}\times33}-1\right\}\times100 = -1.357323232\ [\%]$$

(a)、(b)の結果より、ビットレートモジュレーション機能を使用することで誤差を低減できます。

注.　ビットレートモジュレーション機能未使用時の誤差計算式の詳細は、RX64M グループ ユーザーズマニュアル ハードウェア編 Rev.1.00「40.2.11 ビットレートレジスタ」に記載されています。

本アプリケーションノートの設定以外でのビットレート誤差の計算結果を表5.3、表5.4に示します。

表5.3　ビットレート誤差(ビットレートモジュレーション機能使用時)

ビットレート (bps)	動作周波数 PCLKB(MHz)											
	10				30				60			
	n	N	M	誤差(%)	n	N	M	誤差(%)	n	N	M	誤差(%)
1200	0	176	174	0.0011	1	176	232	0.0011	1	205	135	-0.0031
2400	0	117	232	0.0011	0	205	135	-0.0031	1	176	232	0.0011
4800	0	58	232	0.0011	0	176	232	0.0011	0	205	135	-0.0031
9600	0	21	173	-0.0087	0	73	194	0.0069	0	176	232	0.0011
14400	0	14	177	0.0298	0	58	232	0.0011	0	117	232	0.0011
19200	0	10	173	-0.0087	0	36	194	0.0069	0	73	194	0.0069
38400	0	6	220	-0.0913	0	22	241	-0.0715	0	36	194	0.0069
57600	0	4	236	0.0298	0	10	173	-0.0087	0	21	173	-0.0087

表5.4　ビットレート誤差(ビットレートモジュレーション機能未使用時)

ビットレート (bps)	動作周波数 PCLKB(MHz)								
	10			30			60		
	n	N	誤差(%)	n	N	誤差(%)	n	N	誤差(%)
1200	1	64	0.1603	1	194	0.1603	2	97	-0.3508
2400	0	129	0.1603	1	97	-0.3508	1	194	0.1603
4800	0	64	0.1603	0	194	0.1603	1	97	-0.3508
9600	0	32	-1.3573	0	97	-0.3508	0	194	0.1603
14400	0	21	-1.3573	0	64	0.1603	0	129	0.1603
19200	0	15	1.7252	0	48	-0.3508	0	97	-0.3508
38400	0	7	1.7252	0	23	1.7252	0	48	-0.3508
57600	0	4	8.5069	0	15	1.7252	0	32	-1.357

5.3　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
main.c	メイン処理	
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	

5.4　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。

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	0012 0068h～0012 006Bh	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	0012 006Ch～0012 006Fh	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
MDE	0012 0064h～0012 0067h	FFFF FFFFh	リトルエンディアン

5.5　定数一覧

表5.7～表5.10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(main.c)

定数名	設定値	内容
LED0_REG_PODR	PORT0.PODR.BIT.B3	LED0 出力データ格納ビット
LED0_REG_PDR	PORT0.PDR.BIT.B3	LED0 方向制御ビット
LED0_REG_PMR	PORT0.PMR.BIT.B3	LED0 端子モード制御ビット
LED1_REG_PODR	PORT0.PODR.BIT.B5	LED1 出力データ格納ビット
LED1_REG_PDR	PORT0.PDR.BIT.B5	LED1 方向制御ビット
LED1_REG_PMR	PORT0.PMR.BIT.B5	LED1 端子モード制御ビット
LED2_REG_PODR	PORT2.PODR.BIT.B6	LED2 出力データ格納ビット
LED2_REG_PDR	PORT2.PDR.BIT.B6	LED2 方向制御ビット
LED2_REG_PMR	PORT2.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5.8　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5.9　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/受信中
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)
SCIn	SCI7	SCI チャネル: SCI7
GROUPBLn	GROUPBL0	グループ BL0 割り込み
MSTP_SCIn	MSTP(SCI7)	SCI7 モジュールストップ設定ビット
IPR_SCIn_RXIn	IPR(SCI7,RXI7)	SCI7.RXI7 割り込み優先レベル設定ビット
IPR_SCIn_TXIn	IPR(SCI7,TXI7)	SCI7.TXI7 割り込み優先レベル設定ビット
IPR_SCIn_GROUPBLn	IPR(ICU,GROUPBL0)	SCI7.GROUPBL0 割り込み優先レベル設定ビット
IR_SCIn_RXIn	IR(SCI7,RXI7)	SCI7.RXI7 割り込みステータスフラグ
IR_SCIn_TXIn	IR(SCI7,TXI7)	SCI7.TXI7 割り込みステータスフラグ
IR_SCIn_GROUPBLn	IR(ICU,GROUPBL0)	SCI7.GROUPBL0 割り込みステータスフラグ
IS_SCIn_ERIn	ICU.GRPBL0.BIT.IS15	SCI7.ERI7 割り込みステータスフラグ
IS_SCIn_TEIn	ICU.GRPBL0.BIT.IS14	SCI7.TEI7 割り込みステータスフラグ
IEN_SCIn_RXIn	IEN(SCI7,RXI7)	SCI7.RXI7 割り込み要求許可ビット
IEN_SCIn_TXIn	IEN(SCI7,TXI7)	SCI7.TXI7 割り込み要求許可ビット
IEN_SCIn_GROUPBLn	IEN(ICU,GROUPBL0)	SCI7.GROUPBL0 割り込み要求許可ビット
EN_SCIn_ERIn	ICU.GENBL0.BIT.EN15	SCI7.ERI7 割り込み要求許可ビット
EN_SCIn_TEIn	ICU.GENBL0.BIT.EN14	SCI7.TEI7 割り込み要求許可ビット
RXDn_PDR	PORT9.PDR.BIT.B2	P92 方向制御ビット
RXDn_PMR	PORT9.PMR.BIT.B2	P92 端子モード制御ビット
RXDnPFS	P92PFS	P92 端子機能制御ビット
TXDn_PODR	PORT9.PODR.BIT.B0	P90 出力データ格納ビット
TXDn_PDR	PORT9.PDR.BIT.B0	P90 方向制御ビット
TXDn_PMR	PORT9.PMR.BIT.B0	P90 端子モード制御ビット
TXDnPFS	P90PFS	P90 端子機能制御ビット
PSEL_SETTING	0x0Ah	端子機能選択ビット設定値: RXD7、TXD7

表5.10　サンプルコードで使用する定数(sci_cfg.h)

定数名	設定値	内容
ENABLE_BIT_RATE_MODULATION	—	ビットレートモジュレーション機能有効
DISABLE_BIT_RATE_MODULATION	—	ビットレートモジュレーション機能無効
SELECT_SCI7	—	SCI7 チャネル選択

5.6　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

5.7　変数一覧

表5.11にstatic型変数を示します。

表5.11　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 sci_txi_isr
static uint8_t	tx_cnt	送信カウンタ	
static uint8_t *	pbuf_rx	受信バッファへのポインタ	SCI_StartReceive sci_rxi_isr
static uint8_t	rx_cnt	受信カウンタ	
static sci_state_t	state	SCI 状態	SCI_StartReceive SCI_StartTransmit SCI_GetState sci_tei_isr sci_rxi_isr sci_eri_isr

5.8　関数一覧

表5.12に関数を示します。

表5.12　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
Excep_ICU_GROUPBLn	グループBL0割り込み処理

5.9　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、アプリケーションノート「RX64M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	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、176 ピン版(PIN_SIZE=176)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「RX64M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「RX64Mグループ 初期設定例 Rev.1.00」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。TEI 割り込み要求を禁止して、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、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。シリアル受信とシリアル送信を禁止して、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_TEIn	
概　要	SCI.TEI割り込み処理
ヘッダ	なし
宣　言	static void Excep_SCIn_TEIn(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

Excep_ICU_GROUPBLn	
概　要	グループBL0割り込み処理
ヘッダ	なし
宣　言	static void Excep_ICU_GROUPBL0(void)
説　明	グループ BL0 の割り込み処理を行います。
引　数	なし
リターン値	なし

5.10　フローチャート

5.10.1　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。

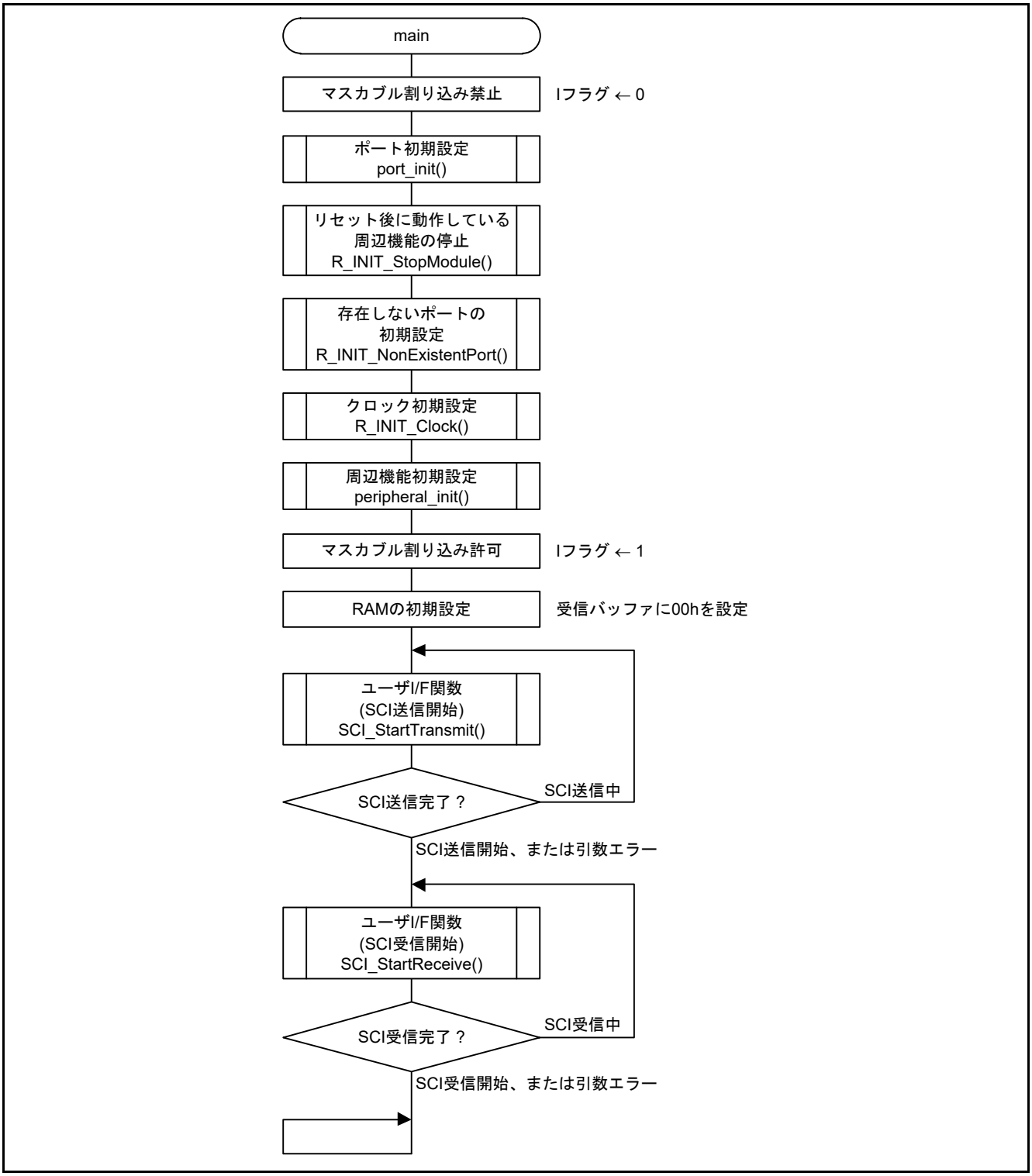


図5.5　メイン処理

5.10.2　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。

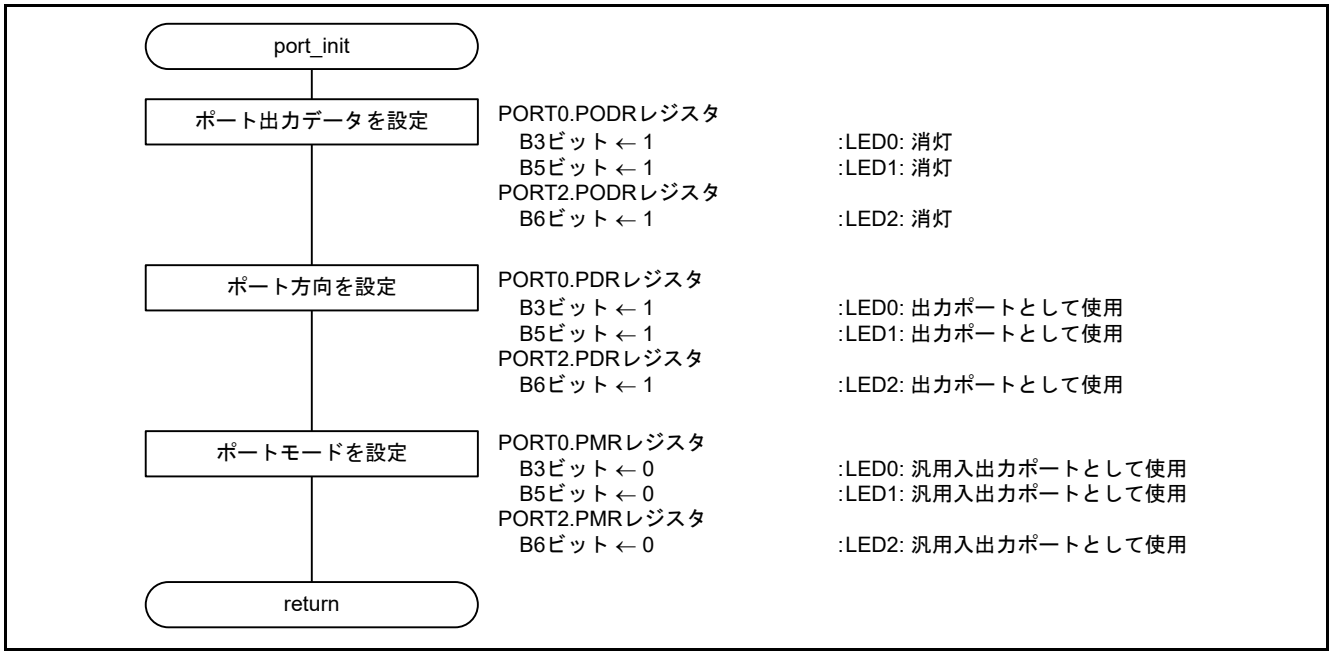


図5.6　ポート初期設定

5.10.3　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。

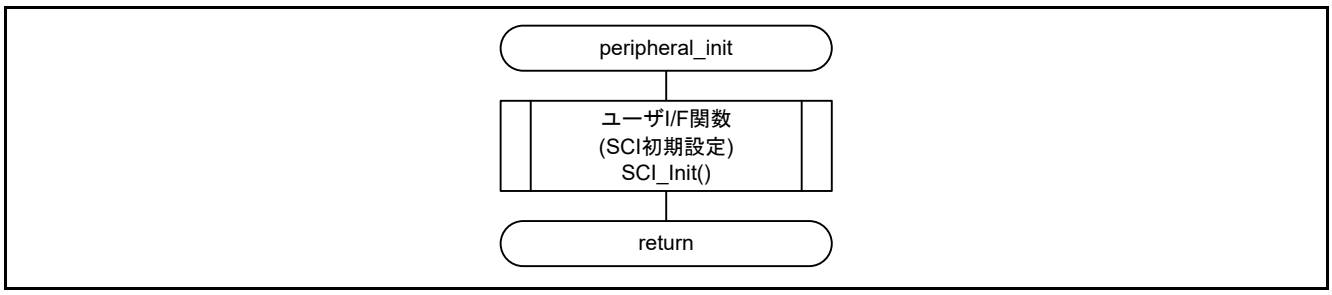


図5.7　周辺機能初期設定

5.10.4　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。

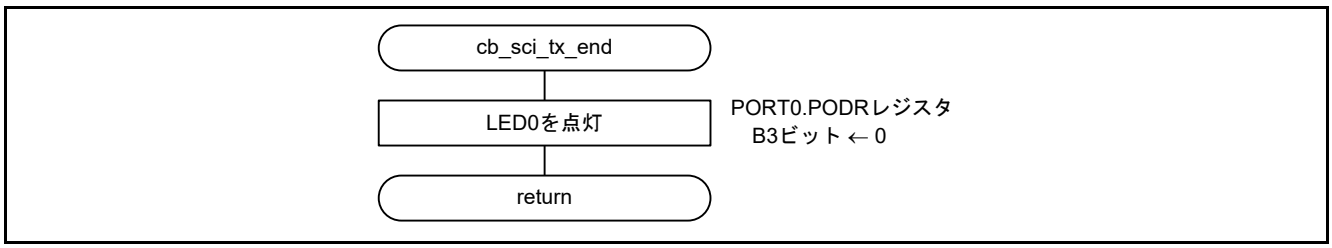


図5.8　コールバック関数(SCI 送信完了)

5.10.5　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。

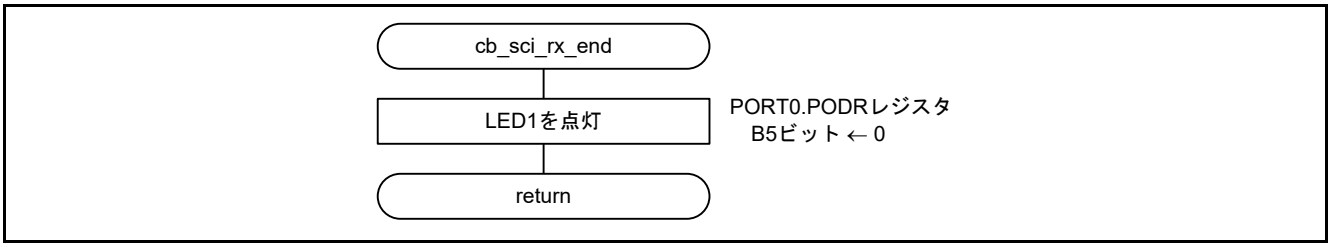
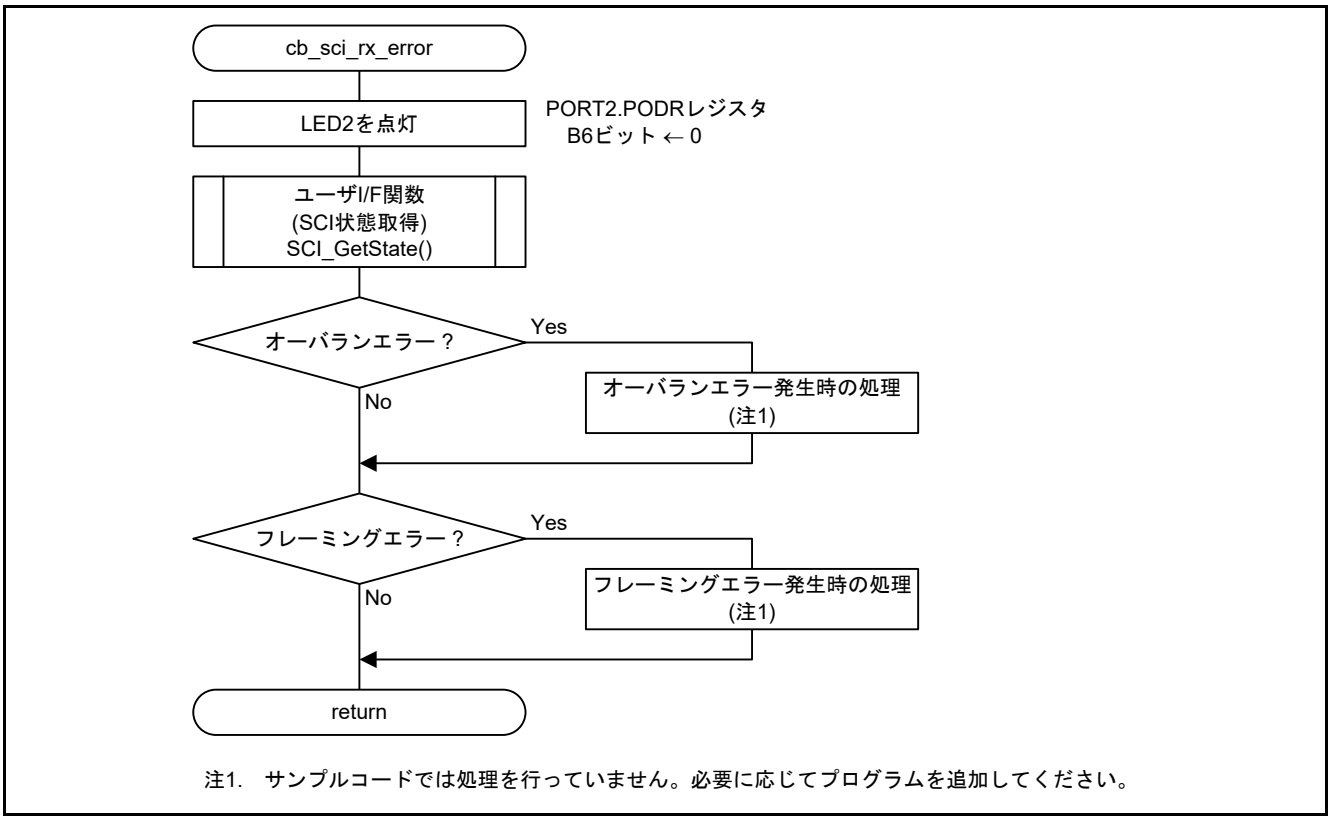


図5.9　コールバック関数(SCI 受信完了)

5.10.6　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。

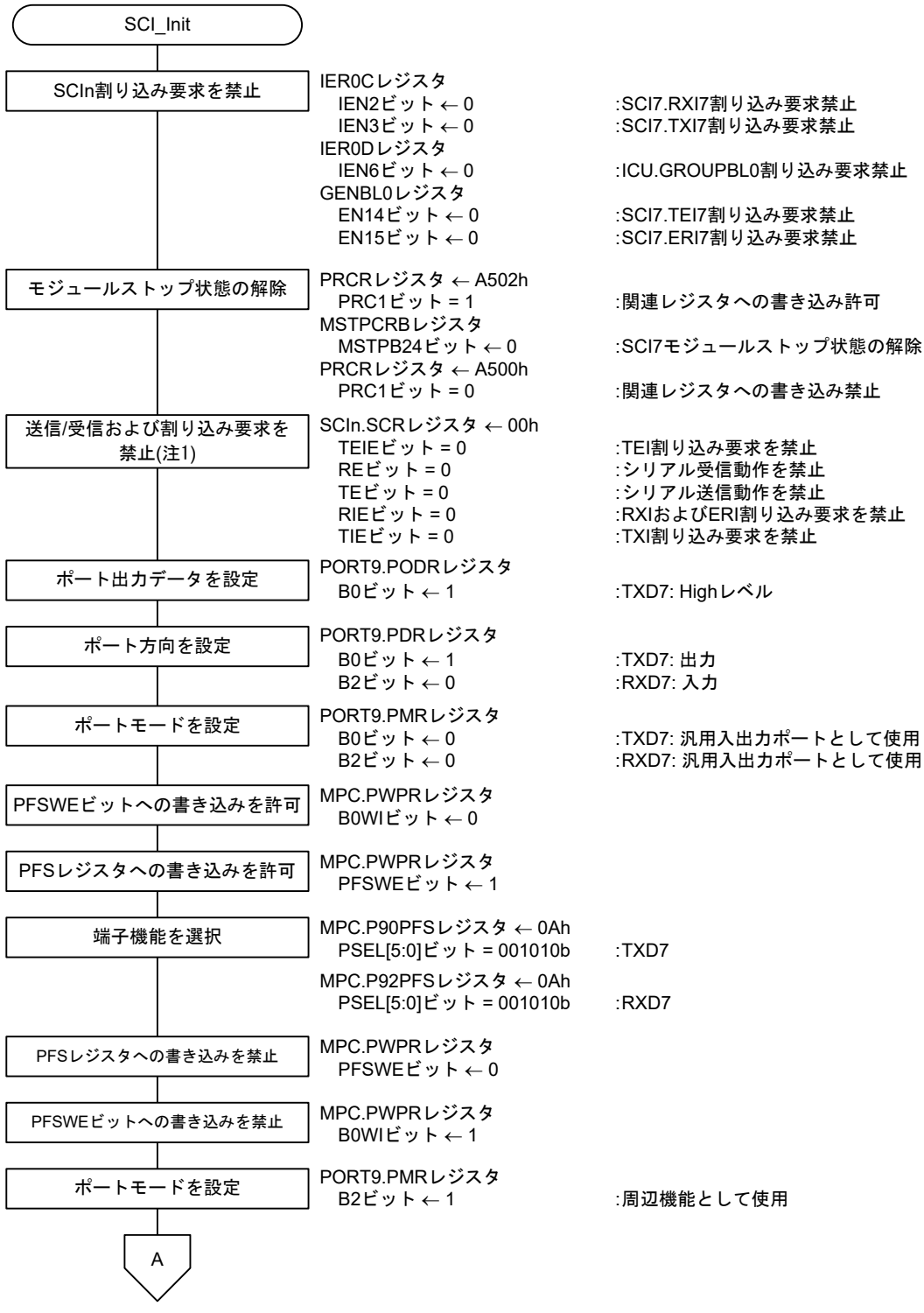


注1.　サンプルコードでは処理を行っていません。必要に応じてプログラムを追加してください。

図5.10　コールバック関数(SCI 受信エラー)

5.10.7　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。



注1.　REビット、TEビット、RIEビット、TIEビットに値を書き込んだ後、書き込んだ値が読めることを確認してください。

図 5.11　ユーザ I/F 関数(SCI 初期設定) (1/2)

注1.　ビットレートモジュレーション機能使用時の設定値です。

図 5.12　ユーザ I/F 関数(SCI 初期設定)　(2/2)

5.10.8　ユーザ I/F 関数(SCI 受信開始)

図5.13にユーザ/F関数(SCI受信開始)のフローチャートを示します。

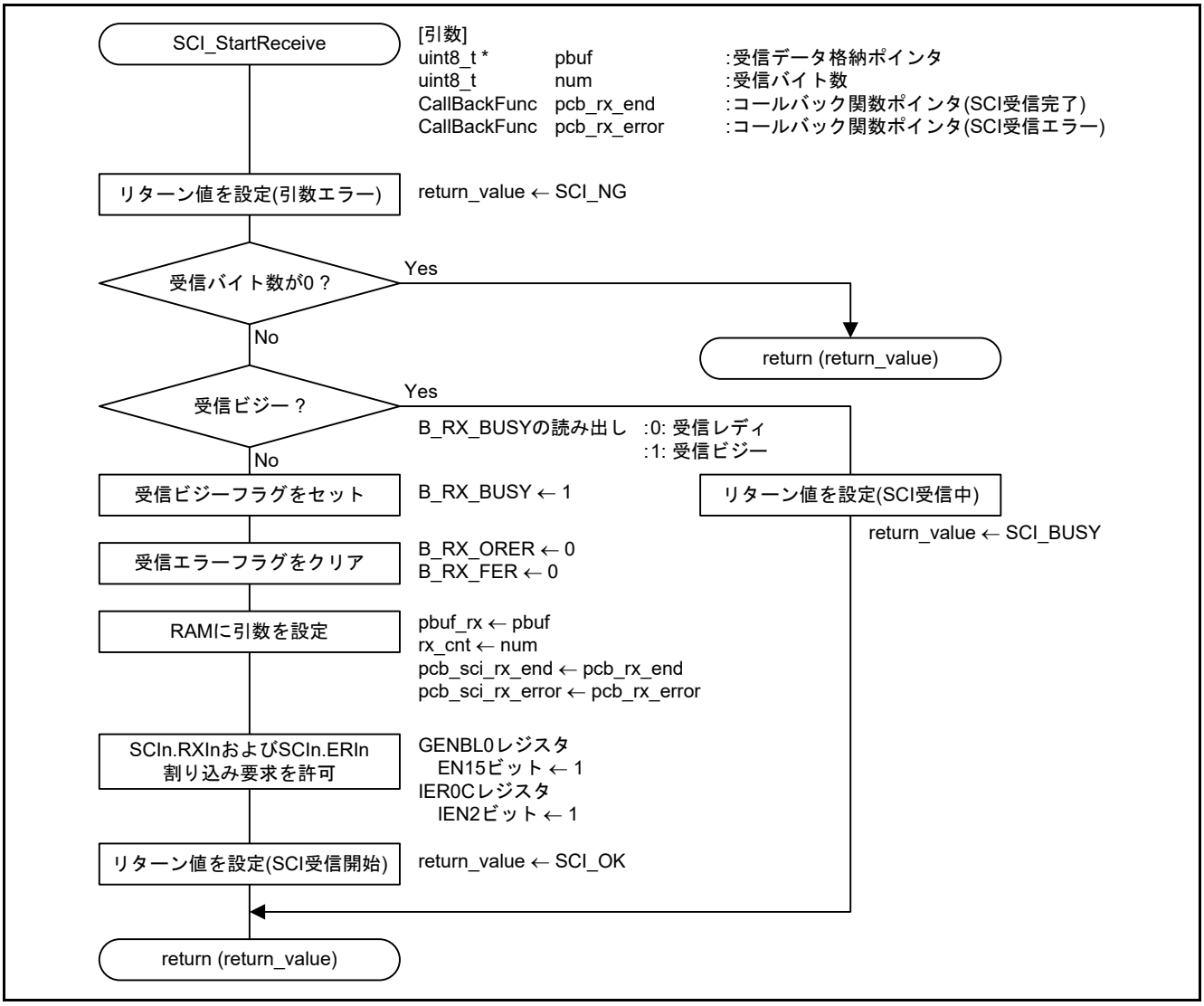
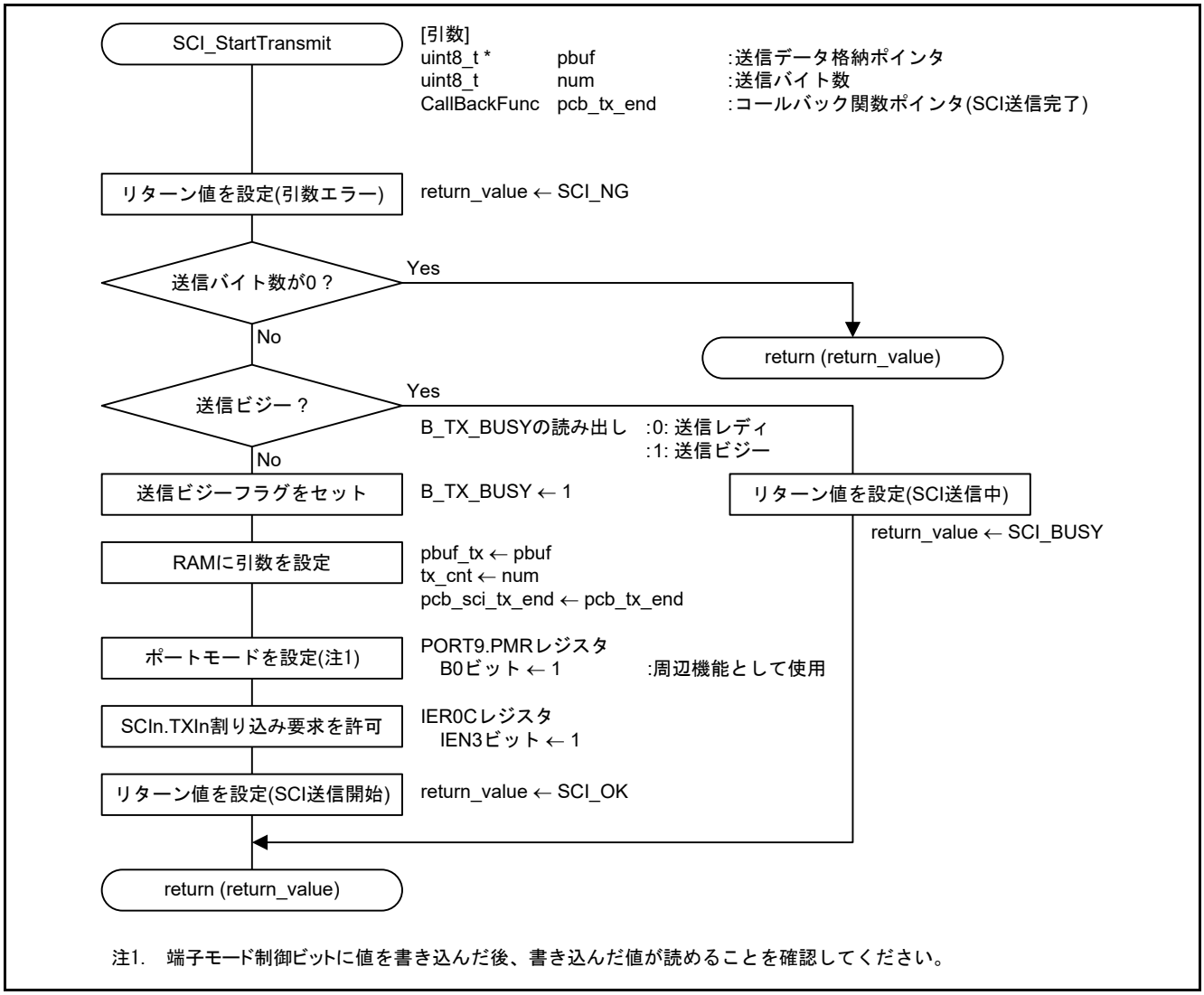


図 5.13　ユーザ/F 関数(SCI 受信開始)

5.10.9　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。



注1.　端子モード制御ビットに値を書き込んだ後、書き込んだ値が読めることを確認してください。

図 5.14　ユーザ I/F 関数(SCI 送信開始)

5.10.10　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。

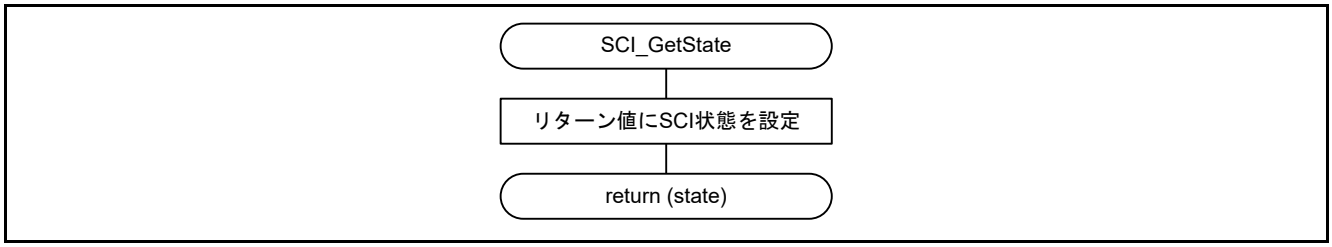


図 5.15　ユーザ I/F 関数(SCI 状態取得)

5.10.11　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。

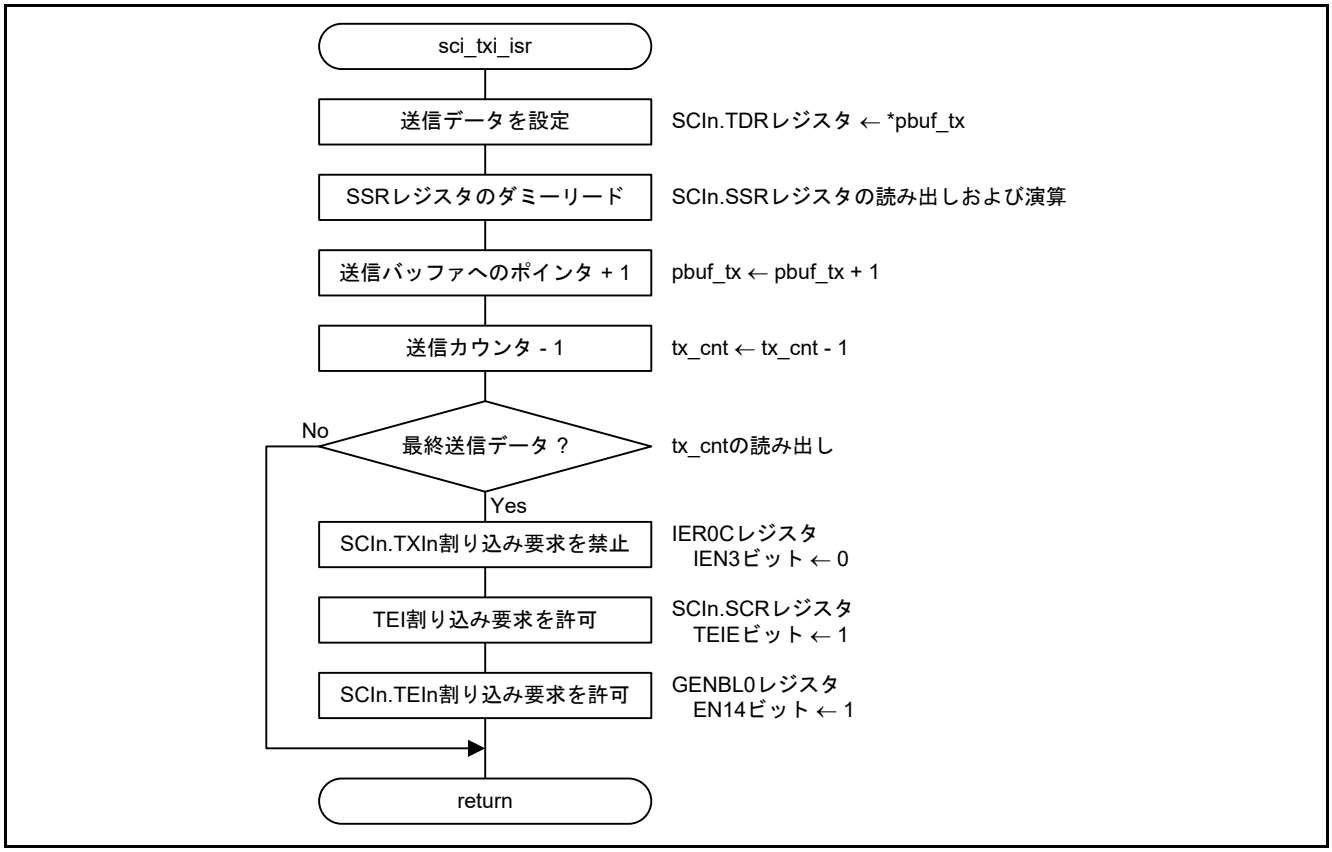


図 5.16　送信データエンプティ割り込み

5.10.12　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。

注1.　端子モード制御ビット、TEIEビットに値を書き込んだ後、書き込んだ値が読めることを確認してください。

図 5.17　送信終了割り込み

5.10.13　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。

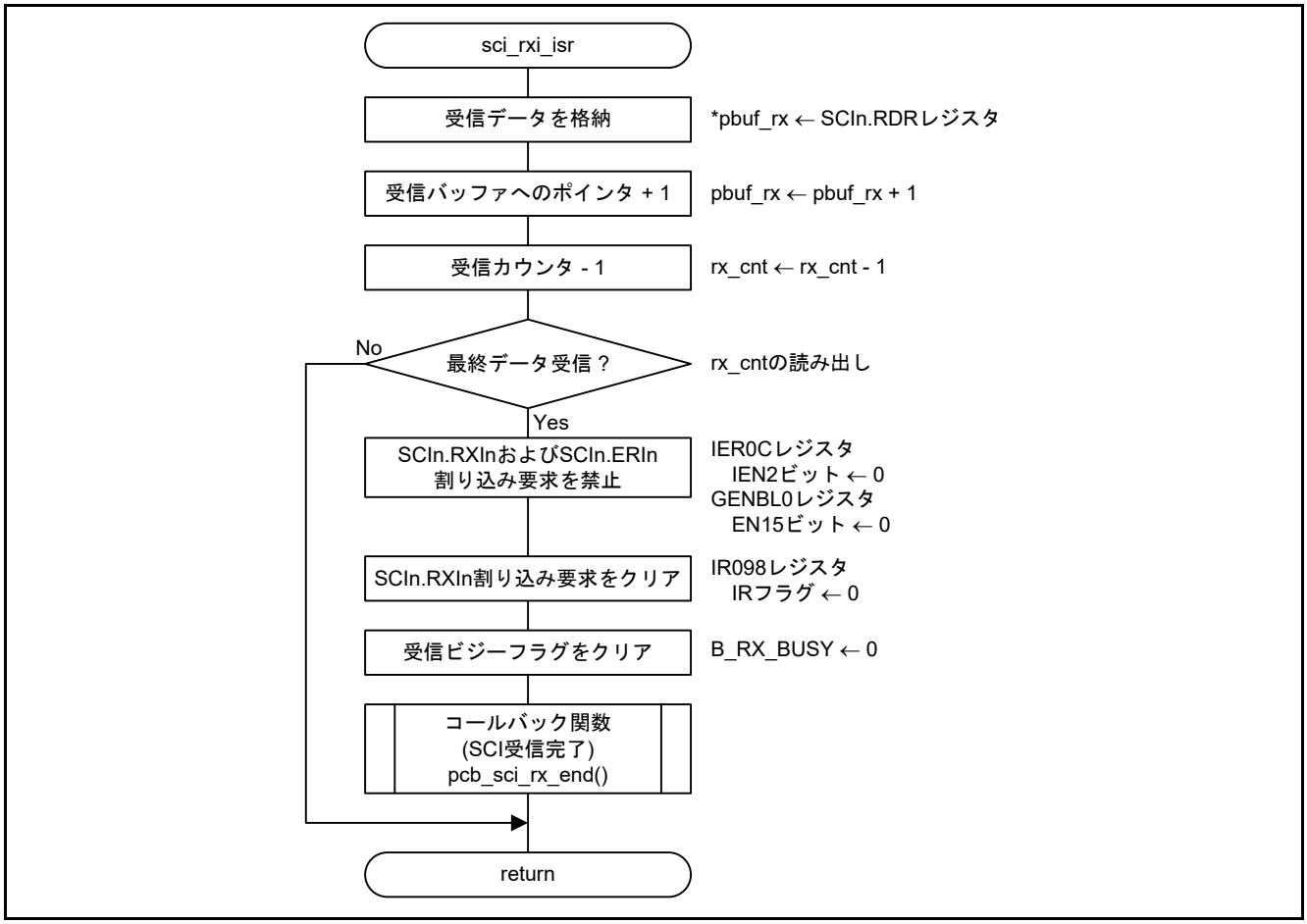
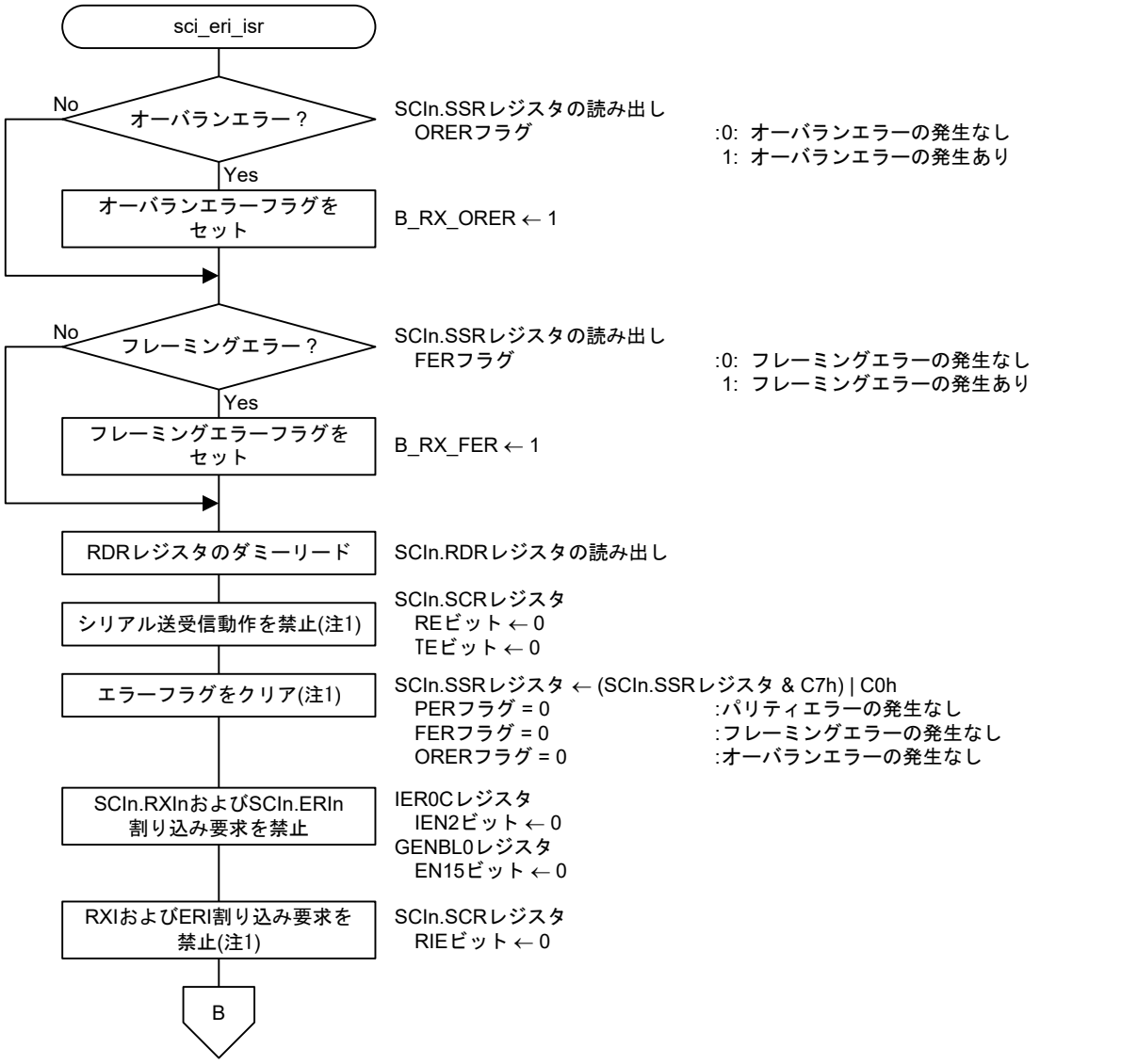


図 5.18　受信データフル割り込み

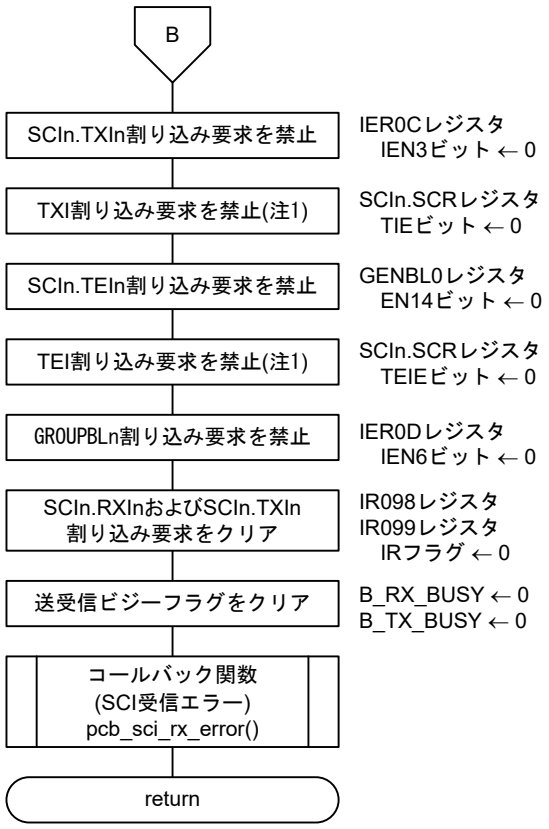
5.10.14　受信エラー割り込み

図5.19、図5.20に受信エラー割り込みのフローチャートを示します。



注1.　REビット、RIEビット、PERフラグ、FERフラグ、ORERフラグに値を書き込んだ後、書き込んだ値が読めることを確認してください。

図 5.19　受信エラー割り込み(1/2)



注1.　TIEビット、TEIEビットに値を書き込んだ後、書き込んだ値が読めることを確認してください。

図 5.20　受信エラー割り込み (2/2)

5.10.15　SCI.ERI 割り込み処理

図5.21にSCI.ERI割り込み処理のフローチャートを示します。

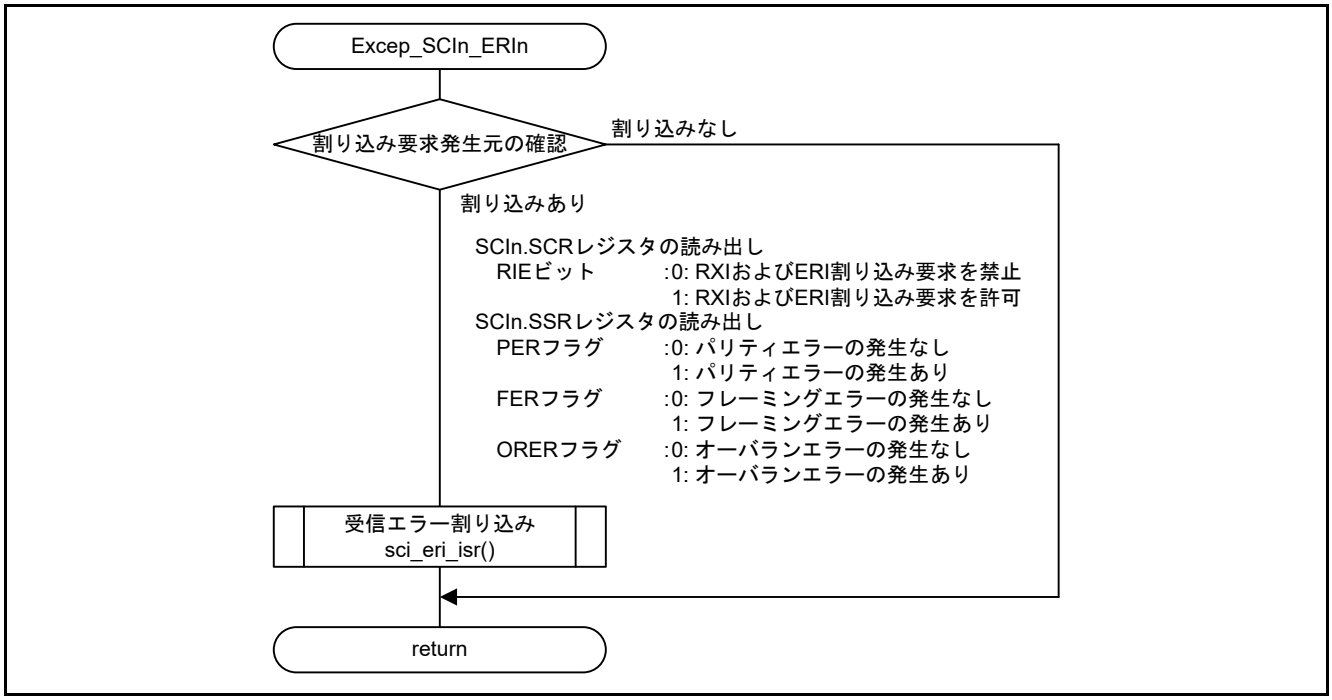


図 5.21　SCI.ERI 割り込み処理

5.10.16　SCI.RXI 割り込み処理

図5.22にSCI.RXI割り込み処理のフローチャートを示します。

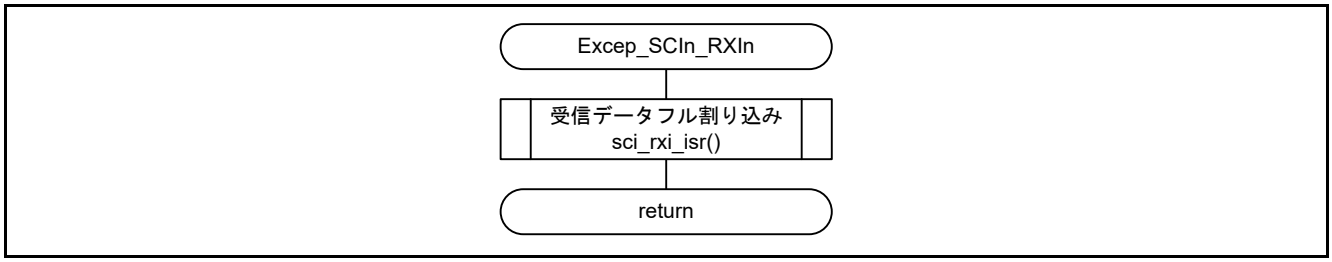


図 5.22　SCI.RXI 割り込み処理

5.10.17　SCI.TXI 割り込み処理

図5.23にSCI.TXI割り込み処理のフローチャートを示します。

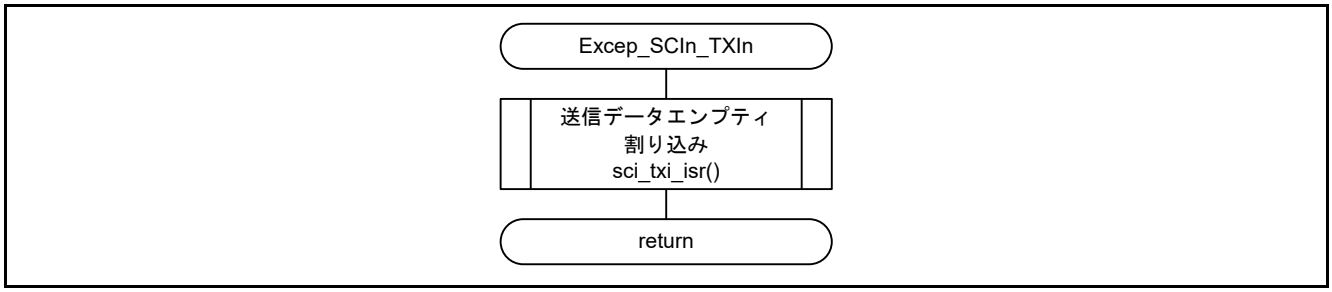


図 5.23　SCI.TXI 割り込み処理

5.10.18　SCI.TEI 割り込み処理

図5.24にSCI.TEI割り込み処理のフローチャートを示します。

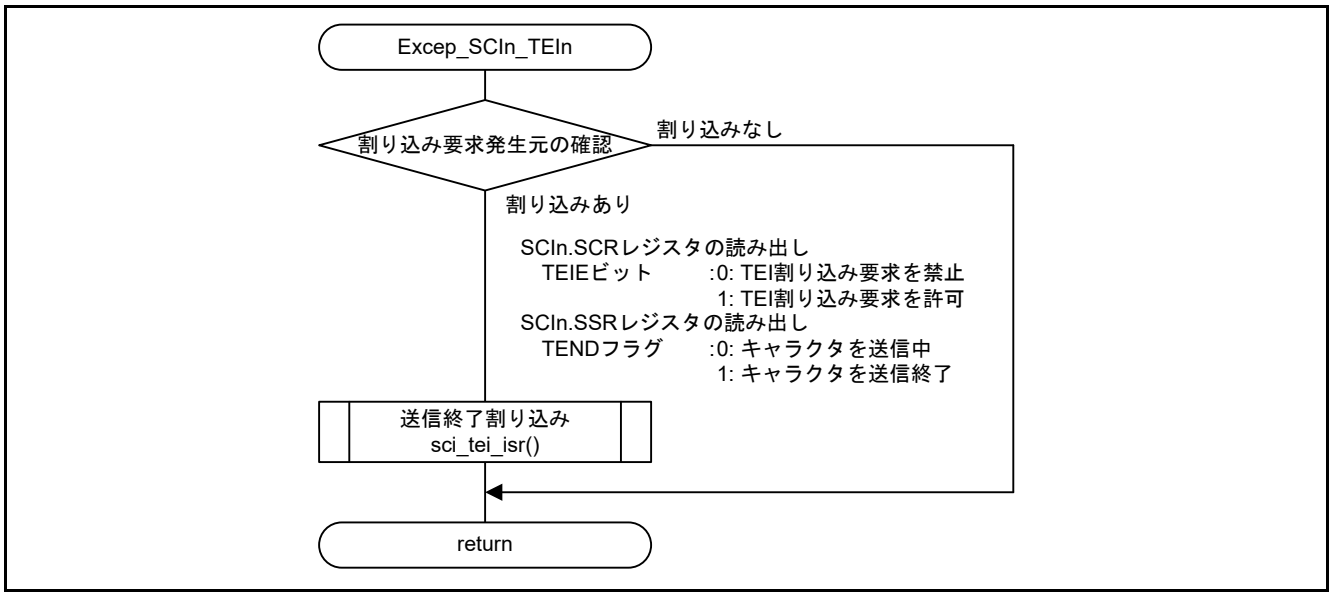


図 5.24　SCI.TEI 割り込み処理

5.10.19　グループ BL0 割り込み処理

図5.25にグループBL0割り込み処理のフローチャートを示します。

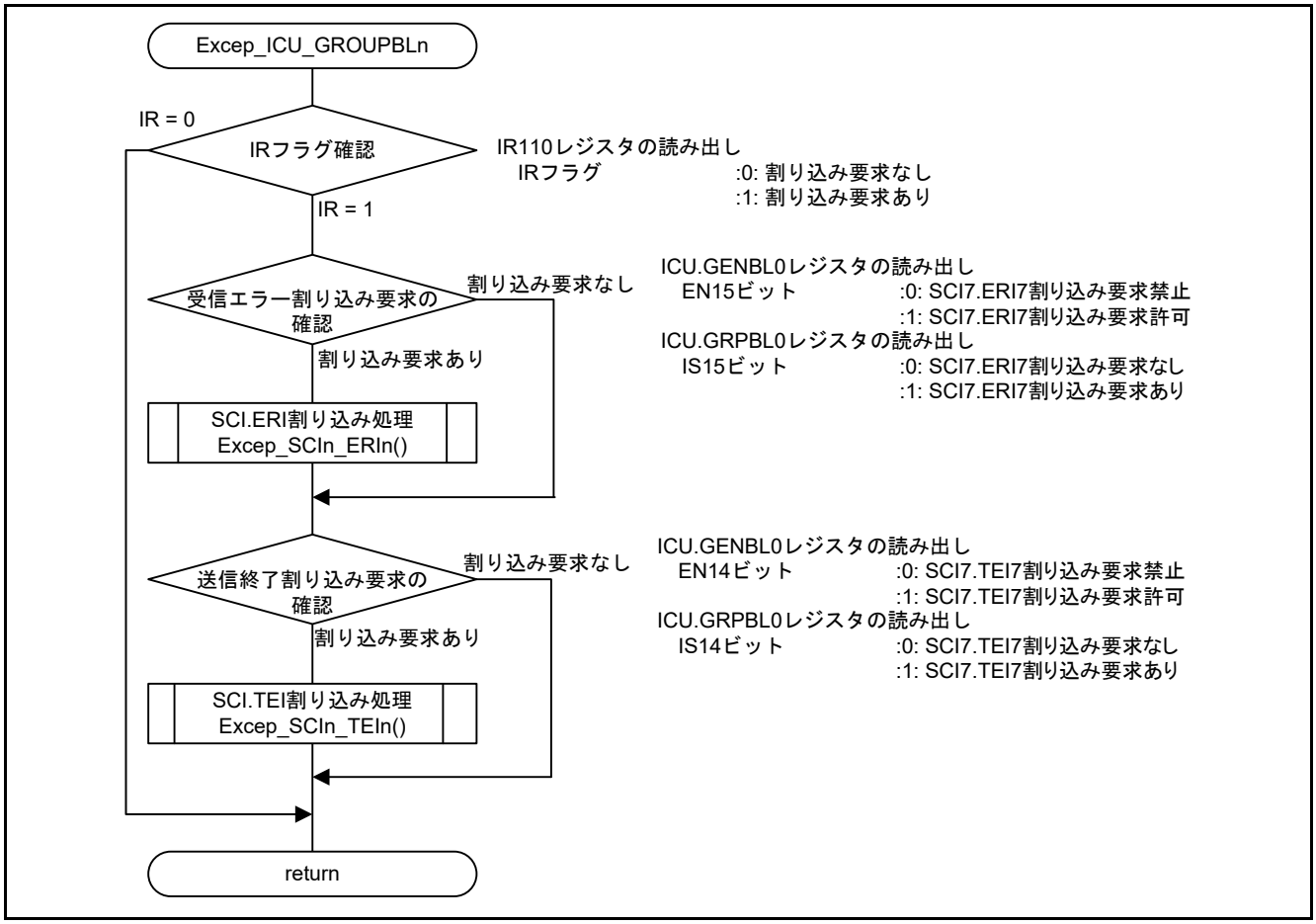


図 5.25　グループ BL0 割り込み処理

6. 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。

7. 参考ドキュメント

ユーザーズマニュアル：ハードウェア
RX64M グループ ユーザーズマニュアル ハードウェア編 Rev.1.00 （R01UH0377JJ）
RX71M グループ ユーザーズマニュアル ハードウェア編 Rev.1.00 （R01UH0493JJ）
（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 コンパイラ CC-RX V2.01.00 ユーザーズマニュアル RX コーディング編 Rev.1.00
（R20UT2748JJ）
（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

お問合せ先
http://japan.renesas.com/contact/

改訂記録	RX64M、RX71Mグループ アプリケーションノート SCIg ビットレートモジュレーション機能の使い方

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.10.01	—	初版発行
1.01	2015.11.02	—	対応デバイスに RX71M グループを追加
		25	図 5.5 「SCI 送信開始?」を「SCI 送信完了?」に修正
			図 5.5 「SCI 受信開始?」を「SCI 受信完了?」に修正
1.10	2020.12.21	—	toolchain version の更新

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. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   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、将来の拡張機能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。
8. 製品間の相違について
   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うと、フラッシュメモリ、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、ノイズ幅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生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。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. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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が直接的、間接的に支配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。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。