

---

# RX63N グループ、RX631グループ

R01AN1865JJ0100

Rev.1.00

## DTCaによるクロック同期式 SC1c 送受信(スレーブ)

---

2014.01.06

### 要旨

本アプリケーションノートでは、RX63N グループ、RX631グループのデータトランスファコントローラ(以下、DTC)、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、クロック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

・RX63N グループ	177、176 ピン版	ROM 容量 : 768KB~2MB
・RX63N グループ	145、144 ピン版	ROM 容量 : 768KB~2MB
・RX63N グループ	100 ピン版	ROM 容量 : 768KB~2MB
・RX631 グループ	177、176 ピン版	ROM 容量 : 256KB~2MB
・RX631 グループ	145、144 ピン版	ROM 容量 : 256KB~2MB
・RX631 グループ	100 ピン版	ROM 容量 : 256KB~2MB

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 ハードウェア構成例 .....	5
4.2 使用端子一覧.....	5
5. ソフトウェア説明.....	6
5.1 動作概要.....	8
5.1.1 送信動作 .....	8
5.1.2 受信動作 .....	10
5.1.3 通信準備完了信号について.....	11
5.2 セクション構成 .....	12
5.3 ファイル構成.....	12
5.4 オプション設定メモリ .....	13
5.5 定数一覧.....	14
5.6 構造体/共用体一覧.....	15
5.7 変数一覧.....	16
5.8 関数一覧.....	17
5.9 関数仕様.....	18
5.10 フローチャート .....	22
5.10.1 メイン処理 .....	22
5.10.2 ポート初期化処理.....	23
5.10.3 周辺機能初期設定.....	23
5.10.4 IRQ 初期設定 .....	24
5.10.5 送受信完了処理(コールバック関数).....	24
5.10.6 SCI9 初期設定 .....	25
5.10.7 DTC 初期設定 .....	27
5.10.8 SCI9 送受信開始処理 .....	28
5.10.9 SCI 送受信停止処理 .....	31
5.10.10 通信準備完了信号出力処理.....	31
5.10.11 SCI ステータス取得処理 .....	31
5.10.12 RXI9 割り込み処理 .....	32
5.10.13 TXI9 割り込み処理 .....	32
5.10.14 TEI9 割り込み処理 .....	33
5.10.15 GROUP12 割り込み(ERI9 割り込み)処理 .....	34
6. サンプルコード .....	35
7. 参考ドキュメント.....	35

1. 仕様

SCI を使用してクロック同期式のシリアル送受信を行います。

割り込み要求端子(IRQ15)の立ち下がリエッジを検出すると、送受信許可状態に設定し、PB4 端子から Low(通信準備完了)を出力します。外部デバイスより SCK 端子にクロックが入力されると、クロックに同期し 13 バイトの送受信を行います。送信データは、ROM 領域に保存した文字列(“Hello world!”)を使用します。受信データは、RAM 領域のバッファに保存します。

送信/受信が完了すると、LED0 を反転出力します。受信中にエラーが発生した場合、LED1 を反転出力します。

- 転送フォーマット : 8 ビット長、LSB ファースト
- クロック入出力 : クロック入力(スレーブ)

表 1.1に使用する周辺機能と用途を、図 1.1にブロック図を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
SCIc チャンネル 9 (以下、SCI9)	クロック同期式シリアル送受信
DTCa	送信データを SCI9 へ転送、および SCI9 の受信データを RAM へ転送
IRQ15	シリアル送受信開始トリガ
I/O ポート	LED 点灯 通信準備完了信号

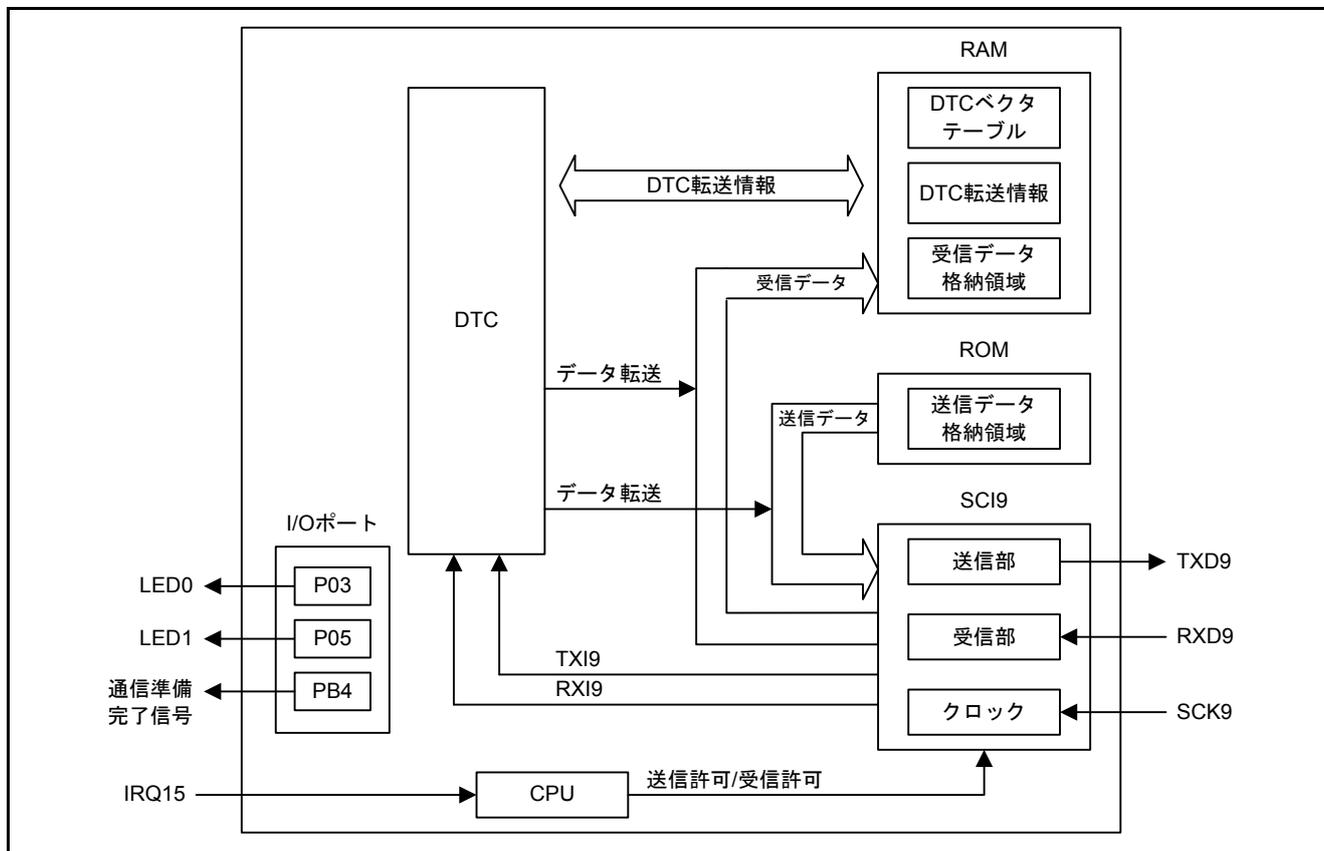


図1.1 ブロック図

## 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	R5F563NBDDFC (RX63N グループ)
動作周波数	メインクロック: 12MHz PLL: 192MHz (メインクロック 1 分周 16 逡倍) 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)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01
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=rx600 -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 1.60A
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 RX63N (製品型名: R0K50563NC000BE)

## 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。

RX63N グループ、RX631 グループ 初期設定例 Rev.1.10 (R01AN1245JJ0110\_RX63N)

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あります。

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に接続例を示します。

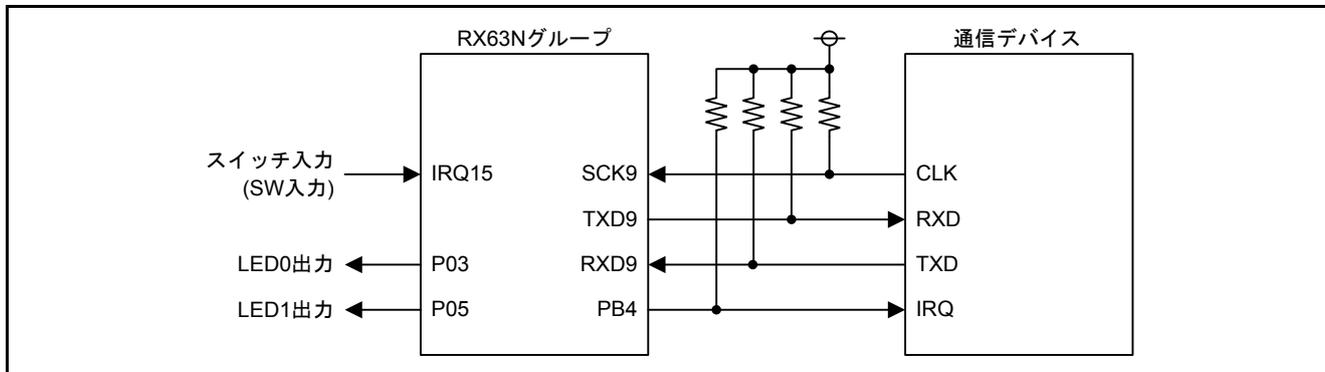


図4.1 接続例

### 4.2 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
P07/IRQ15	入力	送受信開始 SW 入力
PB4	出力	通信準備完了信号
PB5/SCK9	入力	SCI9 のクロック入力
PB6/RXD9	入力	SCI9 の受信データ入力
PB7/TXD9	出力	SCI9 の送信データ出力
P03	出力	LED0 出力(SCI 送信/受信完了)
P05	出力	LED1 出力(SCI 受信エラー)

## 5. ソフトウェア説明

DTC を使用することにより、SCI9 送信/受信を自動的に処理します。SCI 開始関数(api\_sci\_start)の引数で、送受信、送信のみ、受信のみの通信を選択できます。

本サンプルコードでは、送受信開始 SW が押されると、SCI 送信/受信の設定を行います。

リセット解除後、SCI の初期設定および、DTC の初期設定を行います。DTC の転送要因には、TXI9 割り込み、RXI9 割り込みを設定します。

IRQ15 に立ち下がリエッジが入力されたとき、送信完了フラグ(tx\_end\_flag)、受信完了フラグ(rx\_end\_flag)を“0”にし、SCI9 の送受信を許可に設定します。

送信を許可にすると、DTC 転送要因である TXI9 割り込み要求が発生します。DTC により、ROM に配置した送信データが TDR レジスタに転送され、送信許可状態となります。DTC による TDR レジスタの更新後、PCLK で 5 サイクル以上経過したのちに、PB4 端子を Low 出力(通信準備完了)にします。外部デバイスからクロックが入力されると送受信が開始し、送信データが出力されます(注 1)。

受信が完了すると、DTC 転送要因である RXI9 割り込み要求が発生します。DTC により、RDR レジスタの値を受信データ格納領域に転送します。

送信データの転送が指定回数分終わると、CPU への TXI9 割り込み要求が発生します。TXI9 割り込み処理で、TXI9 割り込みを禁止にして、TEI9 割り込みを許可にします。

受信データの転送が転送回数終わると、CPU への RXI9 割り込み要求が発生します。RXI9 割り込み処理で、受信完了フラグ(rx\_end\_flag)を“1”にします。このとき、送信完了フラグが“1”であれば、SCI9 送受信を禁止、PB4 端子を High 出力(通信準備未完了)にします。また、コールバック関数を呼び出します(注 2)。

指定回数分の送信が完了すると、TEI9 割り込み要求が発生します。TEI9 割り込み処理で、送信完了フラグ(tx\_end\_flag)を“1”にします。このとき、受信完了フラグが“1”であれば、SCI9 送受信を禁止、PB4 端子を High 出力(通信準備未完了)にします。また、コールバック関数を呼び出します(注 1)。

オーバランエラーが発生した場合、SCI9 送受信を禁止、PB4 端子を High 出力(通信準備中)にし、コールバック関数を呼び出します(注 1)。

注1. DTC による TDR レジスタ更新後、PCLK で 4 サイクル以内に外部クロックが入力されると、誤動作することがあります。外部デバイスでは、通信準備完了信号を受信してからクロックを出力するようにしてください。

注2. コールバック関数を指定していない場合(null 指定の場合)、コールバック関数は呼び出しません。本アプリケーションノートでは、コールバック関数で送信/受信状態を確認し、送信/受信完了していた場合は LED0 を反転出力、オーバランエラーが発生していた場合は LED1 を反転出力します。

使用する周辺機能の設定を以下に示します。

< SCI9 >

- 通信モード : クロック同期式
- SCK9 端子 : 外部クロック入力(スレーブ)
- 通信速度 : 1000000bps (外部デバイスからのクロック入力)
- データ転送方向 : LSB ファースト
- ハードウェア制御 : 使用しない
- 送受信動作 : 送受信許可、送信のみ許可、または受信のみ許可
- 割り込み : 送信終了割り込み(TEI9)を使用(送信許可時のみ)  
: 送信データエンpty割り込み(TXI9)を使用(送信許可時のみ)  
: 受信データフル割り込み(RXI9)を使用(受信許可時のみ)  
: 受信エラー割り込み(ERI9)を使用(受信許可時のみ)

< DTC >

- 転送要因 : TXI9 および RXI9 割り込み要求
- DTC アドレスモード : フルアドレスモード

[TXI9 割り込み要求による DTC 転送設定]

- 転送モード : ノーマル転送モード
- 転送元アドレッシングモード : 転送後 SAR レジスタをインクリメント
- 転送元アドレス : api\_sci\_start 関数で指定した送信データ格納領域の開始アドレス
- 転送先アドレッシングモード : DAR レジスタはアドレス固定
- 転送先アドレス : SCI9.TDR レジスタ
- データ転送単位 : 8 ビット
- 転送回数 : api\_sci\_start 関数で指定した転送回数
- チェーン転送 : 禁止
- 割り込み : 指定されたデータ転送終了後、CPU への割り込みが発生

[RXI9 割り込み要求による DTC 転送設定]

- 転送モード : ノーマル転送モード
- 転送元アドレッシングモード : SAR レジスタはアドレス固定
- 転送元アドレス : SCI9.RDR レジスタ
- 転送先アドレッシングモード : 転送後、DAR レジスタをインクリメント
- 転送先アドレス : api\_sci\_start 関数で指定した受信データ格納領域の先頭アドレス
- データ転送単位 : 8 ビット
- 転送回数 : api\_sci\_start 関数で指定した転送回数
- チェーン転送 : 禁止
- 割り込み : 指定されたデータ転送終了後、CPU への割り込みが発生

< IRQ15 >

- 検出方法 : 立ち下がリエッジ
- デジタルフィルタ : 有効(PCLK/8)
- 割り込み : 使用しない

## 5.1 動作概要

### 5.1.1 送信動作

#### (1) 初期設定

初期設定後、送受信開始 SW を待ちます。

#### (2) 送受信開始 SW 入力の検出

送受信開始 SW 入力を検出したとき、IRQ15 割り込みの IR フラグを“0”にします。SCI ステータスを確認し、SCI の送受信(BUSY)でないことを確認した後、DTC の転送元アドレス、転送回数を設定して、DTC 転送を許可にします。また、送信完了フラグを“0”にして、SCI9.SCR.TEIE、TIE、TE ビットを同時に“1”にして、送信を許可します(受信も許可する場合は、SCI9.SCR.RIE ビット、RE ビットも同時に“1”にします)。このとき、TXI9 割り込みの IR フラグが“1”になります。

#### (3) データ転送開始

TXI9 割り込みを許可にすると、DTC が起動し、送信データ格納領域から SCI.TDR レジスタに、1 バイト目の送信データが転送されます。また、転送時に TXI9 割り込みの IR フラグが“0”になります。

#### (4) 送信許可状態

SCI9.TDR レジスタから SCI9.TSR レジスタにデータが転送されると、TXI9 割り込みの IR フラグが再度“1”になります。TXD9 割り込み要求により DTC が起動し、2 バイト目の送信データが SCI9.TDR レジスタに転送されます。

#### (5) 送信の開始

DTC による SCI9.TDR レジスタへの更新後、PCLK で 5 サイクル以上経過後に PB4 端子を Low(通信準備完了)にします。外部デバイスから SCK9 端子にクロックが入力されると、TXD9 端子から 1 バイト目の送信データが出力されます。(注 1)

#### (6) 1 バイトのデータ送信完了

1 バイトのデータが送信完了したとき、TXI9 割り込みの IR フラグが“1”になります。また、SCI9.TDR レジスタに値が書きこまれていると、再度 SCI9.TSR レジスタにデータが転送されます。

#### (7) TXI9 割り込み

指定した回数分データ転送が完了すると、CPU に TXI9 割り込み要求が発生します。TXI9 割り込み処理で、TXI9 割り込みを禁止にして、TEI9 割り込みを許可にします。

#### (8) TEI9 割り込み

指定した回数分の送信が完了したとき(SCI9.TDR レジスタが更新されておらず、送信データの最終ビットが出力されたとき)、TEI9 割り込み要求が発生します。TEI9 割り込み処理で、TEI9 割り込みを禁止して、送信完了フラグを“1”にします。また、受信完了フラグが“1”の場合、送受信を禁止、PB4 端子を High(通信準備中)にして、コールバック関数を呼び出します。(ただし、コールバック関数が指定されていない場合は、呼び出しません。)

再度、上記(2)から実行されます。

注1. DTC による TDR レジスタ更新後、PCLK で 4 サイクル以内に外部クロックが入力されると、誤動作することがあります。外部デバイスでは、通信準備完了信号を受信してからクロックを出力するようにしてください。

図 5.1に送信動作のタイミング図を示します。

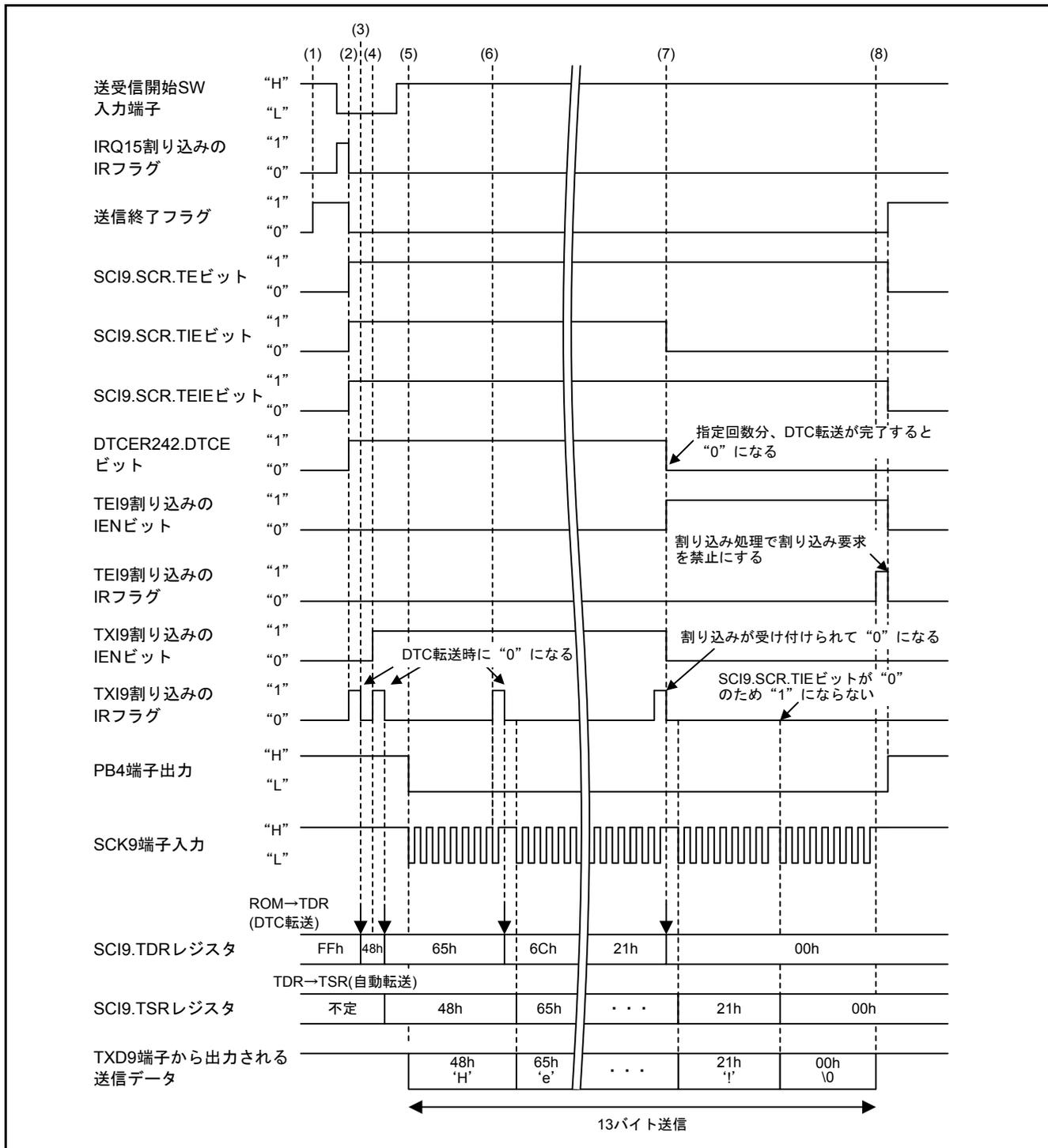


図5.1 送信動作のタイミング図

### 5.1.2 受信動作

#### (1) 初期設定

初期設定後、送受信開始 SW を待ちます。

#### (2) 送受信開始 SW 入力の検出

送受信開始 SW 入力を検出したとき、IRQ15 割り込みの IR フラグを“0”にします。SCI ステータスを確認し、SCI の送受信(BUSY)でないことを確認した後、DTC の転送元アドレス、転送回数を設定して、DTC 転送を許可にします。また、受信完了フラグを“0”にして、SCI9.SCR.RIE、RE ビットを同時に“1”にして、受信を許可します(送信も許可にする場合は、SCI9.SCR.TEIE、TIE、TE ビットも同時に“1”にします)。

#### (3) データ受信開始

受信許可後、PCLK で 5 サイクル以上経過後に PB4 端子を Low(通信準備完了)にします。外部デバイスから SCK9 端子にクロックが入力されると、1 バイト目の送信データが転送されます。クロック出力に同期して、受信動作を行います。

#### (4) RXI9 割り込み

1 バイト目のデータ受信が完了すると、RXI9 割り込みの IR フラグが“1”になります。

#### (5) DTC による受信データの取り込み

RXI9 割り込み要求により、DTC 転送が起動し、SCI9.RDR レジスタから RAM の受信データ格納領域に、受信データが転送されます。また、RXI9 割り込みの IR フラグが“0”になります。

#### (6) 受信完了時

RXI9 割り込み処理で、受信完了フラグを“1”にします。また、送信完了フラグが“1”の場合、送受信を禁止、PB4 端子を High(通信準備中)にして、コールバック関数を呼び出します。(ただし、コールバック関数が指定されていない場合は、呼び出しません。)

再度、上記(2)から実行されます。

図 5.2に受信動作のタイミング図(受信のみ動作時)を示します。

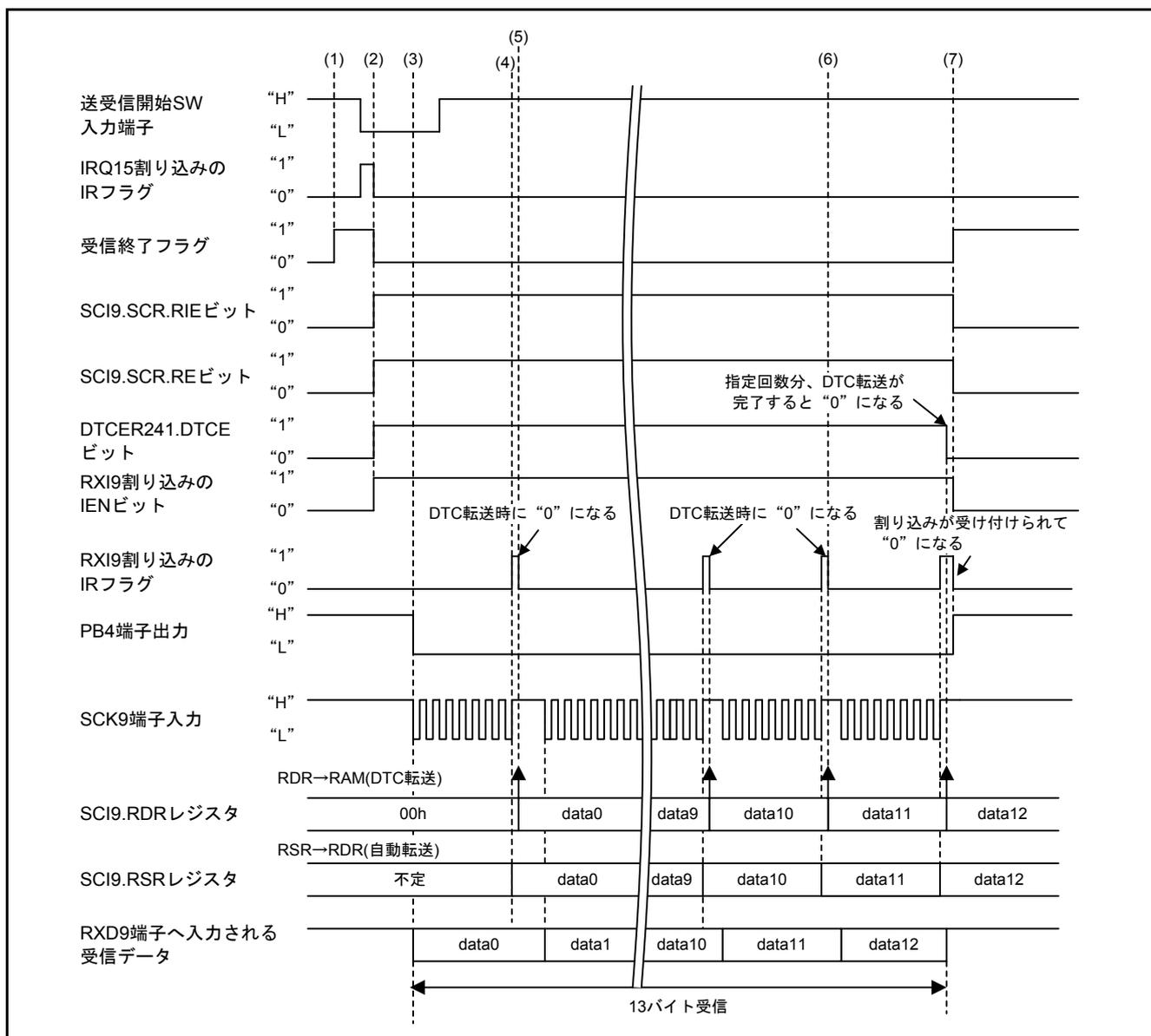


図5.2 受信動作のタイミング図(受信のみ動作時)

### 5.1.3 通信準備完了信号について

DTCによるTDRレジスタ更新後、PCLKで4サイクル以内に外部クロックが入力されると、誤動作することがあります。そのため、スレーブ側でDTCによるTDRレジスタ更新後、PCLKで5サイクル以上待ったのちに、通信準備完了信号をLow(通信準備完了)にしています。外部デバイスでは、通信準備完了信号がLowになったことを確認してから、送信クロックを出力するようにしてください。

また、本サンプルコードでは、送受信完了後、送信完了、または受信完了してから、割り込み処理内で通信準備完了信号がHigh出力(通信準備中)にするまでオーバーヘッドが発生します。外部デバイスでの通信準備完了信号の検出をレベルで行う場合、送受信完了後から一定期間、通信準備完了信号の検出無効時間を設けてください。

通信準備完了信号の検出をエッジで行う場合は、外部デバイス側が起動した状態(エッジ検出できる状態)で、本サンプルコードを動作させてください。

## 5.2 セクション構成

表 5.1にサンプルコードで変更するセクション情報を示します。

セクションの追加/変更および削除の方法は、最新の RX ファミリ C/C++コンパイラパッケージ ユーザーズマニュアルを参照してください。

表5.1 サンプルコードで変更するセクション情報

セクション名	変更	アドレス	内容
DTC_SECTION	追加	0000 3000h	DTC ベクタテーブル

## 5.3 ファイル構成

表 5.2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5.2 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	クロック同期式通信処理	DTC 設定を含む
sci.h	sci.c のヘッダファイル	

#### 5.4 オプション設定メモリ

表 5.3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5.3 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

## 5.5 定数一覧

表 5.4、表 5.5にサンプルコードで使用する定数を示します。

表5.4 サンプルコードで使用する定数(main.c)

定数名	設定値	内容
SW_START	IR(ICU,IRQ15)	送受信開始 SW の IR フラグ
SW_ON	1	SW 入力 ON
SW_OFF	0	SW 入力 OFF
BUF_SIZE	256	受信バッファサイズ
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
LED_ON	0	LED 出力データ: 点灯
LED_OFF	1	LED 出力データ: 消灯

表5.5 サンプルコードで使用する定数(sci.c)

定数名	設定値	内容
SCI_TX_ONLY	1	api_sci_start 関数で指定する送受信設定値 : 送信のみ許可
SCI_RX_ONLY	2	api_sci_start 関数で指定する送受信設定値 : 受信のみ許可
SCI_TX_RX	(SCI_TX_ONLY   SCI_RX_ONLY)	api_sci_start 関数で指定する送受信設定値 : 送信、および受信許可
SCI_RDY	0	api_get_status 関数の戻り値 : SCI 送受信待ち
SCI_BUSY	1	api_get_status 関数、api_sci_start 関数の戻り値 : SCI 送受信中
SCI_ORER	2	api_get_status 関数の戻り値 : オーバランエラー発生
SCI_SUCCESS	0	api_sci_start 関数の戻り値 : SCI 送信/受信開始
SCI_BUF_ERROR	3	api_sci_start 関数の戻り値 : 送信/受信バッファの指定エラー
SCI_MODE_ERROR	4	api_sci_start 関数の戻り値 : 送受信設定エラー
null	(void*)0x00000000	NULL ポインタ

## 5.6 構造体/共用体一覧

図 5.3にサンプルコードで使用する構造体/共用体を示します。

```
/* **** DTC転送情報 **** */
#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 struct
{
    union
    {
        unsigned long LONG;
        struct
        {
            unsigned long MRA_MD :2;
            unsigned long MRA_SZ :2;
            unsigned long MRA_SM :2;
            unsigned long :2;
            unsigned long MRB_CHNE :1;
            unsigned long MRB_CHNS :1;
            unsigned long MRB_DISEL :1;
            unsigned long MRB_DTS :1;
            unsigned long MRB_DM :2;
            unsigned long :2;
            unsigned long :16;
        } BIT;
    } MR;
    void * SAR;
    void * DAR;
    struct
    {
        unsigned long CRA:16;
        unsigned long CRB:16;
    } CR;
}dtc_full_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5.3 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 5.6、表 5.7にstatic 型変数を、表 5.8にconst 型変数を示します。

表5.6 static 型変数(main.c)

型	変数名	内容	使用関数
static uint8_t	rxbuf[BUF_SIZE]	受信データ格納領域	main

表5.7 static 型変数(sci.c)

型	変数名	内容	使用関数
static volatile uint8_t	tx_end_flag	送信完了フラグ	api_sci9_start Excep_SCI9_RXI9 Excep_SCI9_TEI9 Excep_ICU_GROUP12
static volatile uint8_t	rx_end_flag	受信完了フラグ	api_sci9_start Excep_SCI9_RXI9 Excep_SCI9_TEI9 Excep_ICU_GROUP12
static volatile uint8_t	sci_status	SCI ステータス	api_sci9_init api_sci9_start api_get_status Excep_SCI9_RXI9 Excep_SCI9_TEI9 Excep_ICU_GROUP12
static dtc_full_t	dtc_info_rxi	RXI9 の DTC 転送情報	api_dtc_init api_sci9_start
static dtc_full_t	dtc_info_txi	TXI9 の DTC 転送情報	api_dtc_init api_sci9_start
static volatile CallBackFunc	sci_callback	コールバック関数ポインタ	api_sci_start Excep_SCI9_RXI9 Excep_SCI9_TEI9 Excep_ICU_GROUP12
void*	dtc_vect_table[256]	DTC ベクタテーブル	api_dtc_init

表5.8 const 型変数

型	変数名	内容	使用関数
static const uint8_t	txbuf[]	送信データ格納領域 (文字列: "Hello world!")	main

## 5.8 関数一覧

表 5.9に関数を示します。

表5.9 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
irq_init	IRQ15 初期設定
communication_end	送受信完了処理(コールバック関数)
api_sci9_init	SCI9 初期設定
api_dtc_init	DTC 初期設定
api_sci9_start	SCI9 送受信開始処理
api_sci9_stop	SCI9 送受信停止処理
api_output_rdy	通信準備完了信号出力処理
api_get_status	SCI ステータス取得処理
Excep_SCI9_RXI9	RXI9 割り込み処理
Excep_SCI9_TXI9	TXI9 割り込み処理
Excep_SCI9_TEI9	TEI9 割り込み処理
Excep_ICU_GROUP12	GROUP12 割り込み(ERI9 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、送受信開始 SW の入力を検出すると、SCI9 送信/受信を開始します。
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	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「RX63N グループ、RX631 グループ 初期設定例 Rev.1.10」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、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「RX63N グループ、RX631 グループ 初期設定例 Rev.1.10」を参照してください。

---

<b>R_INIT_Clock</b>	
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「RX63N グループ、RX631 グループ 初期設定例 Rev.1.10」を参照してください。

---



---

<b>peripheral_init</b>	
概要	周辺機能初期設定
ヘッダ	なし
宣言	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

---



---

<b>irq_init</b>	
概要	IRQ15 初期設定
ヘッダ	なし
宣言	void irq_init(void)
説明	IRQ15 の初期設定を行います。
引数	なし
リターン値	なし

---



---

<b>communication_end</b>	
概要	送受信完了処理(コールバック関数)
ヘッダ	なし
宣言	static void communication_end(void)
説明	SCI ステータスを確認し、送信/受信完了していた場合は LED0 を反転出力、受信エラーが発生していた場合は、LED1 を反転出力します。
引数	なし
リターン値	なし

---



---

<b>api_sci9_init</b>	
概要	SCI9 初期設定
ヘッダ	sci.h
宣言	void api_sci9_init(void)
説明	SCI9 の初期設定を行います。
引数	なし
リターン値	なし

---

---

<b>api_dtc_init</b>	
概要	DTC 初期設定
ヘッダ	なし
宣言	static void api_dtc_init(void)
説明	DTC の初期設定を行います。
引数	なし
リターン値	なし

---



---

<b>api_sci9_start</b>	
概要	SCI9 送受信開始処理
ヘッダ	sci.h
宣言	uint8_t api_sci9_start( uint8_t sci_mode, void* p_tx_buf, void* p_rx_buf, uint16_t cnt, CallbackFunc cb_func )
説明	
引数	uint8_t sci_mode           SCI 送信/受信設定 SCI_SEND_ONLY : 送信のみ SCI_RECV_ONLY : 受信のみ SCI_SEND_RECV : 送受信 void* p_tx_buf           送信データ格納バッファのポインタ(注 1) void* p_rx_buf           受信データ格納バッファのポインタ(注 2) uint16_t cnt            送信/受信バイト数 CallbackFunc cb_func   コールバック関数ポインタ(注 3)
リターン値	SCI_SUCCESS            : 送信/受信開始 SCI_BUSY               : 送信/受信 SCI_BUF_ERROR        : 送信/受信データ格納バッファの指定エラー SCI_MODE_ERROR       : SCI 送信/受信設定エラー

(注 1) SCI 送信/受信設定に受信のみを設定した場合、設定値は無効です。“null”を指定してください。

(注 2) SCI 送信/受信設定に送信のみを設定した場合、設定値は無効です。“null”を指定してください。

(注 3) 送信/受信が完了したとき、またはオーバランエラーが発生したときに呼び出されます。

コールバック関数を使用しない場合は、“null”を設定してください。

---

<b>api_sci9_stop</b>	
概要	SCI9 送受信停止処理
ヘッダ	なし
宣言	static void api_sci9_stop(void)
説明	SCI9 の送受信を停止します。
引数	なし
リターン値	なし

---



---

<b>api_output_rdy</b>	
概要	通信準備完了信号出力処理
ヘッダ	なし
宣言	static void api_output_rdy(void)
説明	PCLK で 5 サイクル以上待ち、通信準備完了信号を出力します。
引数	なし
リターン値	なし

---

---

<b>api_get_status</b>	
概要	SCI ステータス取得処理
ヘッダ	sci.h
宣言	uint8_t api_get_status( void )
説明	SCI9 の送受信状態を取得します。
引数	なし
リターン値	SCI_RDY : SCI 送受信待ち SCI_BUSY : SCI 送受信中 SCI_ORER : オーバランエラー発生

---

<b>Excep_SCI9_RXI9</b>	
概要	RXI9 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCI9_RXI9(void)
説明	(受信のみ許可時)最終データ受信時、受信完了フラグ、送信完了フラグを“1”にします。SCI9 受信を停止し、受信データを取り込んだのち、コールバック関数を呼び出します。 (送信も許可時)受信完了フラグを“1”にします。送信完了フラグが“1”の場合、SCI9 送受信を停止し、コールバック関数を呼び出します。
引数	なし
リターン値	なし

---

<b>TXI9 割り込み処理</b>	
概要	TXI9 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCI9_TXI9(void)
説明	TXI9 割り込みを禁止し、TEI 割り込みを許可にします。
引数	なし
リターン値	なし

---

<b>Excep_SCI9_TEI9</b>	
概要	TEI9 割り込み処理
ヘッダ	なし
宣言	static void Excep_SCI9_TEI9(void)
説明	TEI9 割り込みを禁止にし、送信完了フラグを“1”にします。受信完了フラグが“1”の場合、SCI9 送受信を停止し、コールバック関数を呼び出します。
引数	なし
リターン値	なし

---

<b>Excep_ICU_GROUP12</b>	
概要	GROPU12 割り込み(ERI9 割り込み)処理
ヘッダ	なし
宣言	static void Excep_ICU_GROUP12(void)
説明	SCI9 の送受信を停止し、初期化を行います。また、コールバック関数を呼び出します。
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 5.4にメイン処理のフローチャートを示します。

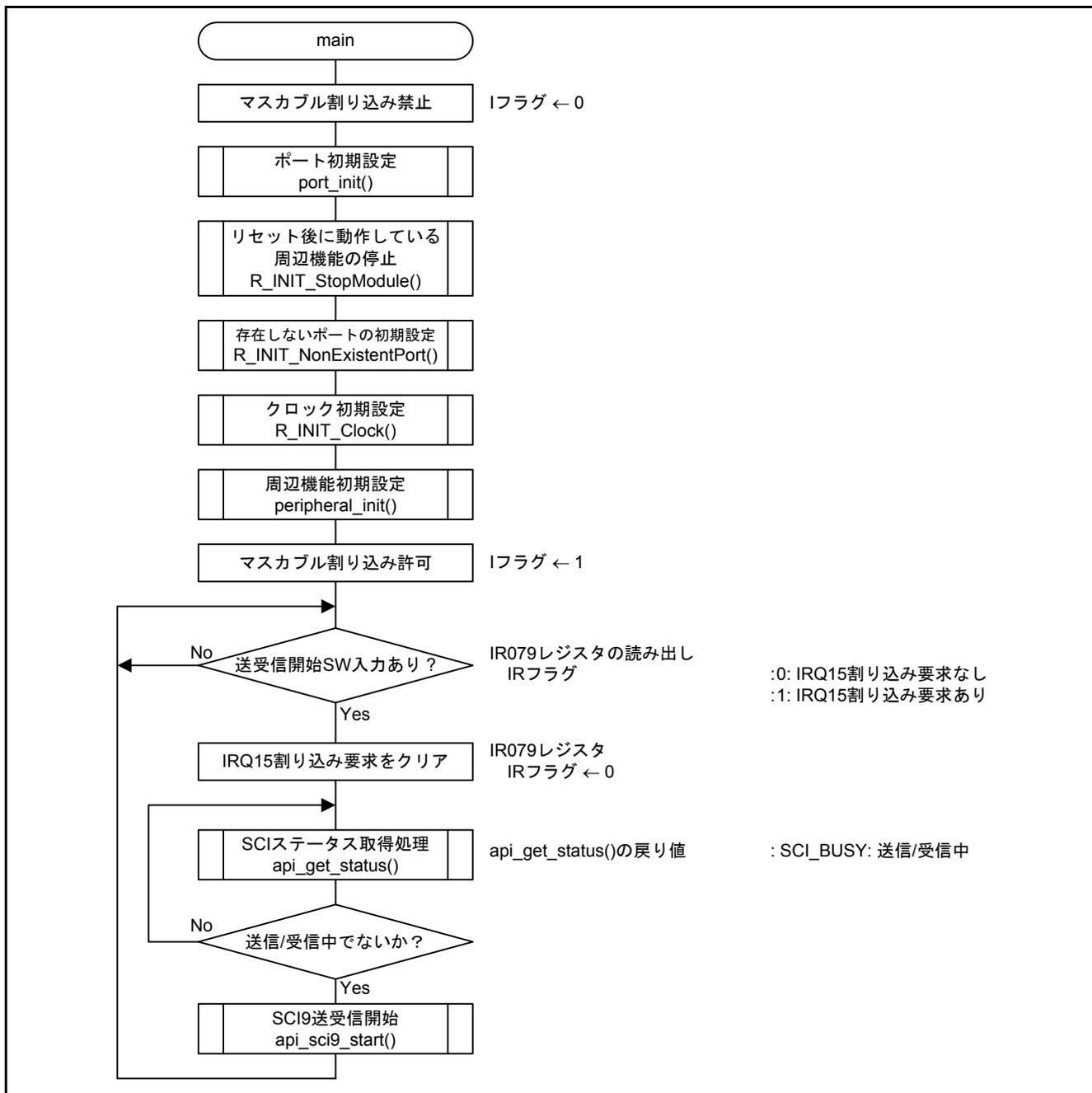


図5.4 メイン処理

### 5.10.2 ポート初期化処理

図 5.5にポート初期化処理のフローチャートを示します。

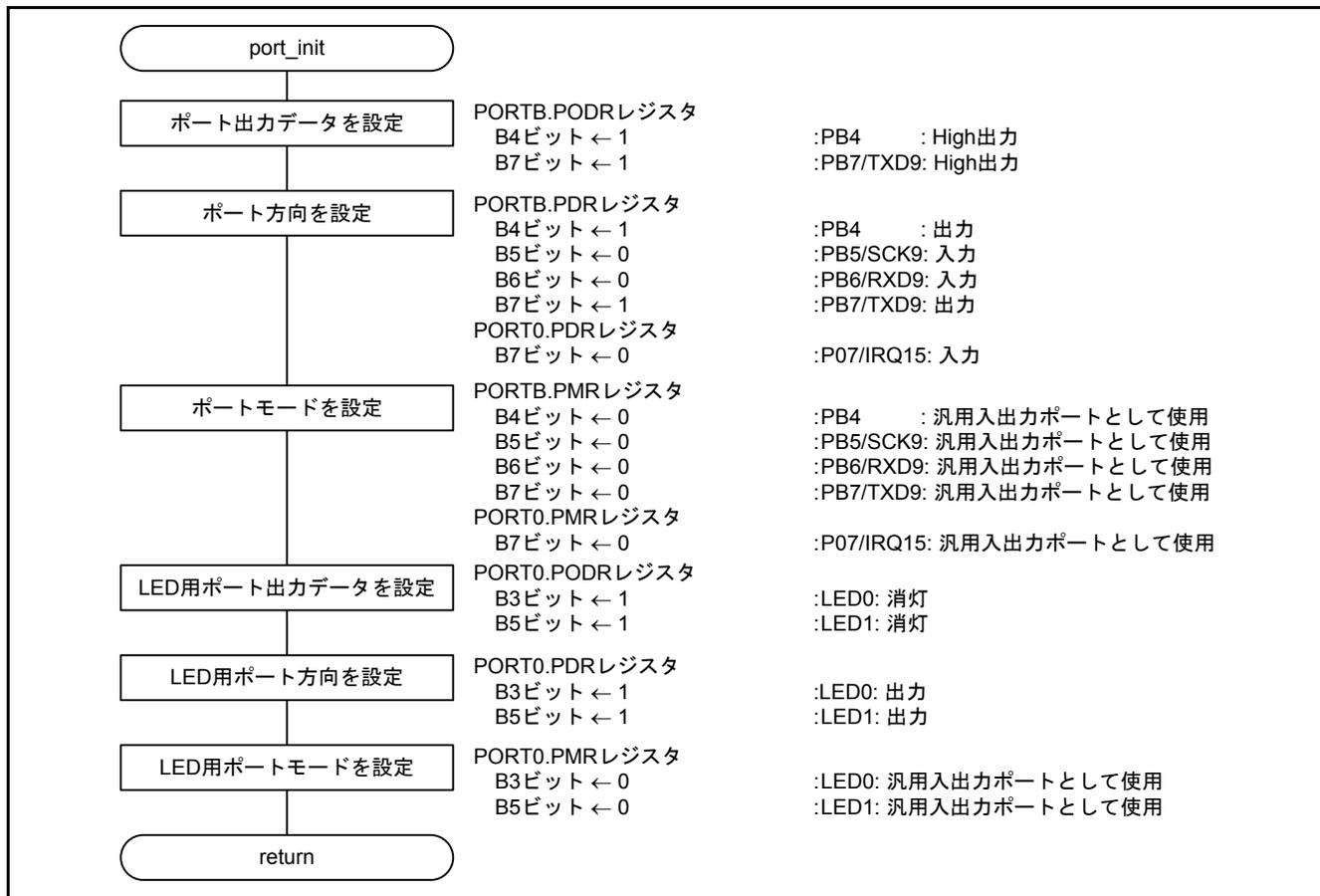


図5.5 ポート初期化処理

### 5.10.3 周辺機能初期設定

図 5.6に周辺機能初期設定のフローチャートを示します。

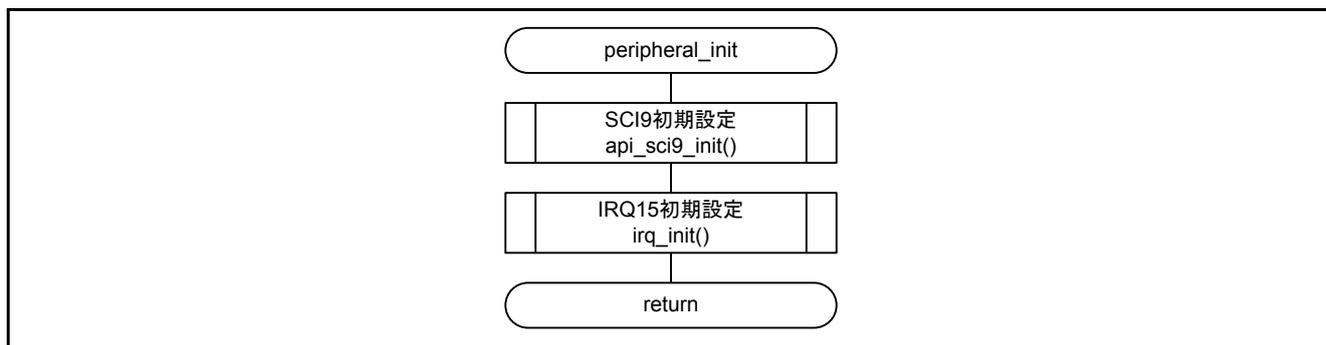


図5.6 周辺機能初期設定

### 5.10.4 IRQ 初期設定

図 5.7にIRQ 初期設定のフローチャートを示します。

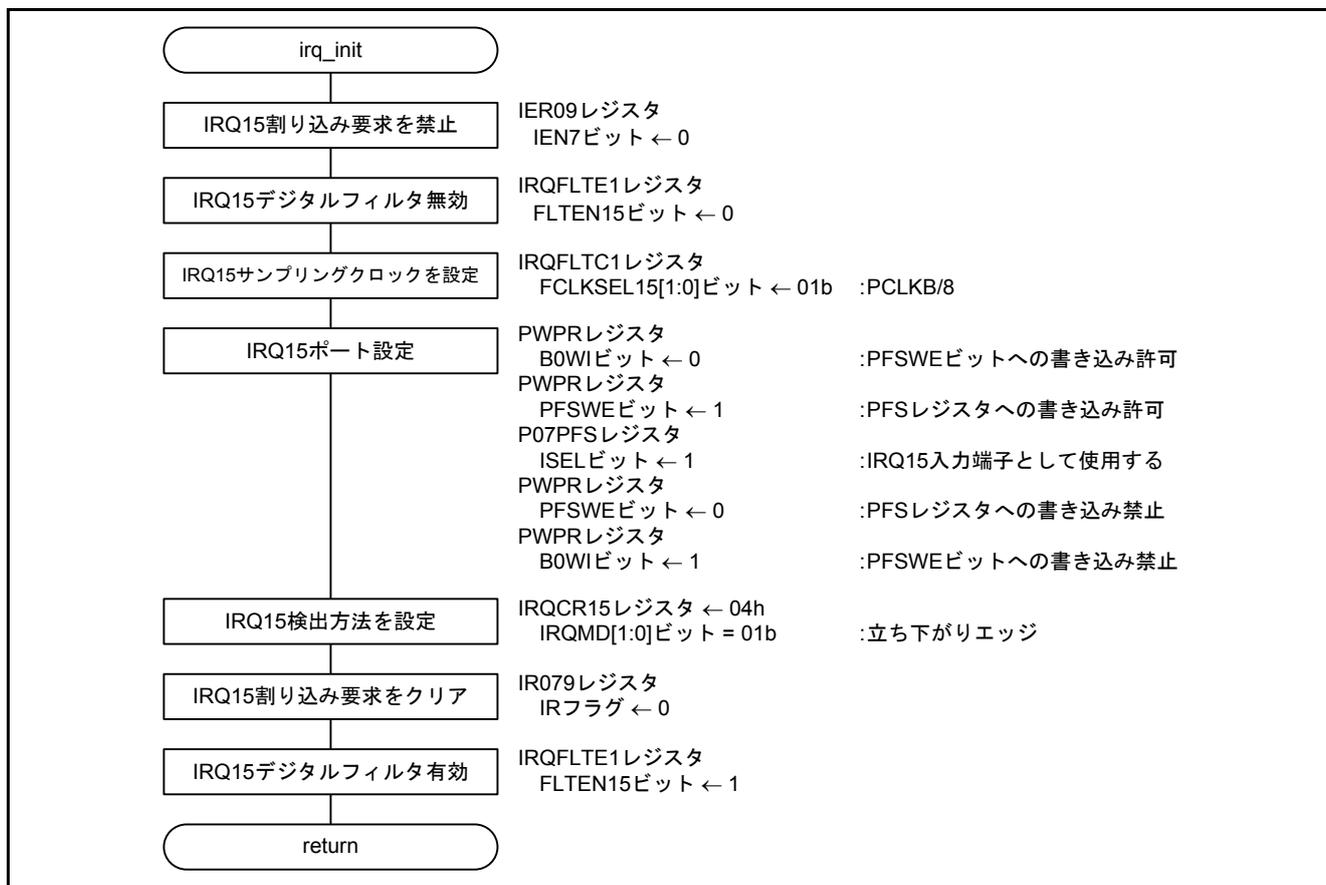


図5.7 IRQ 初期設定

### 5.10.5 送受信完了処理(コールバック関数)

図 5.8に送受信完了処理(コールバック関数)のフローチャートを示します。

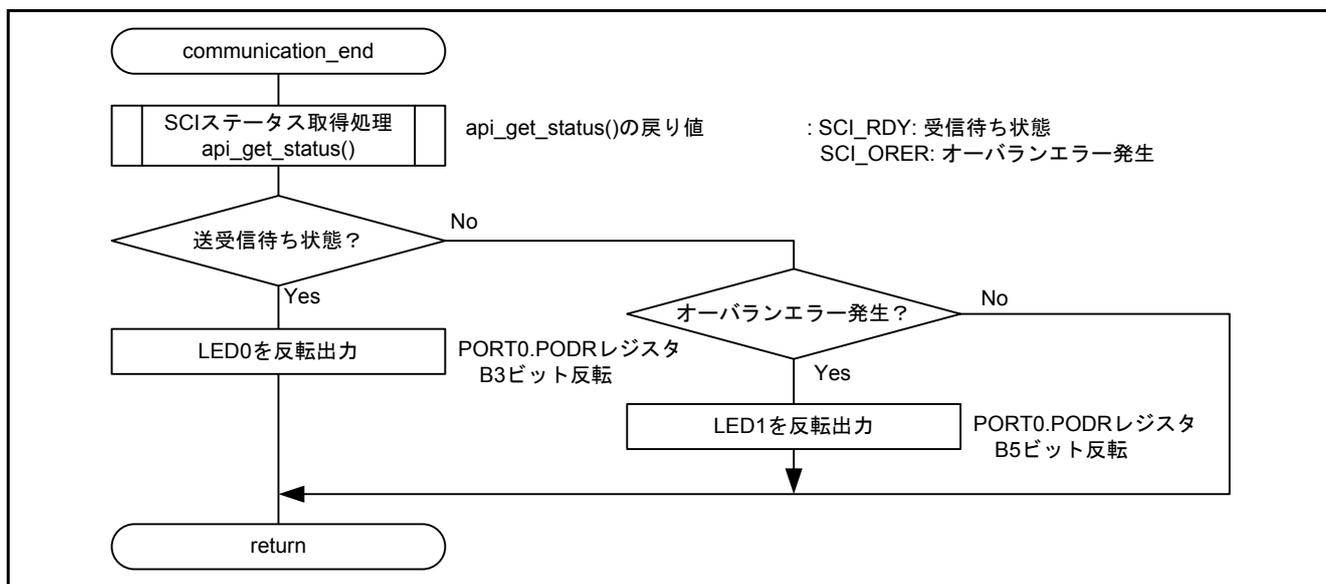


図5.8 送受信完了処理(コールバック関数)

5.10.6 SCI9 初期設定

図 5.9～図 5.10にSCI9 初期設定のフローチャートを示します。

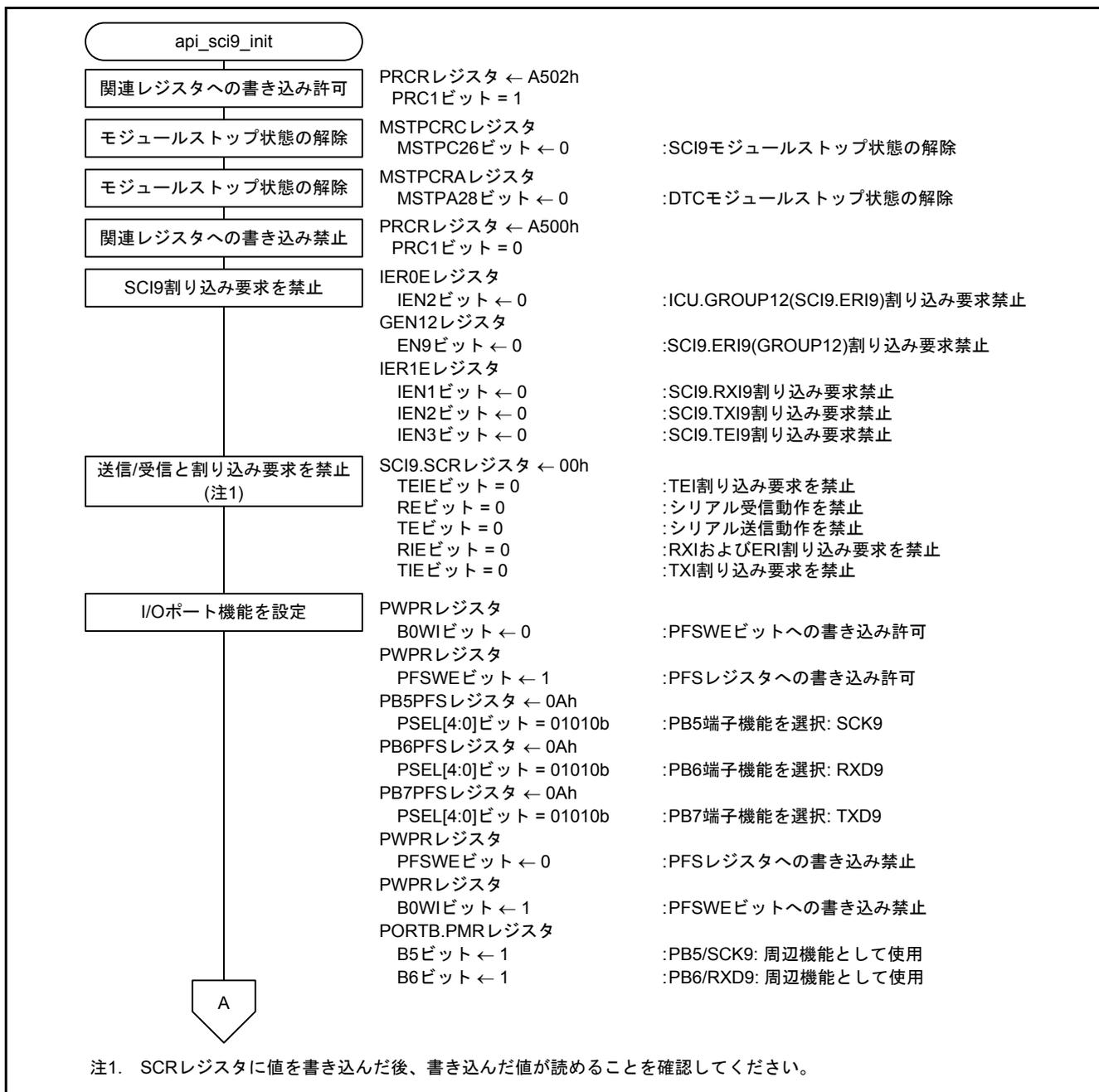


図5.9 SCI9 初期設定(1/2)

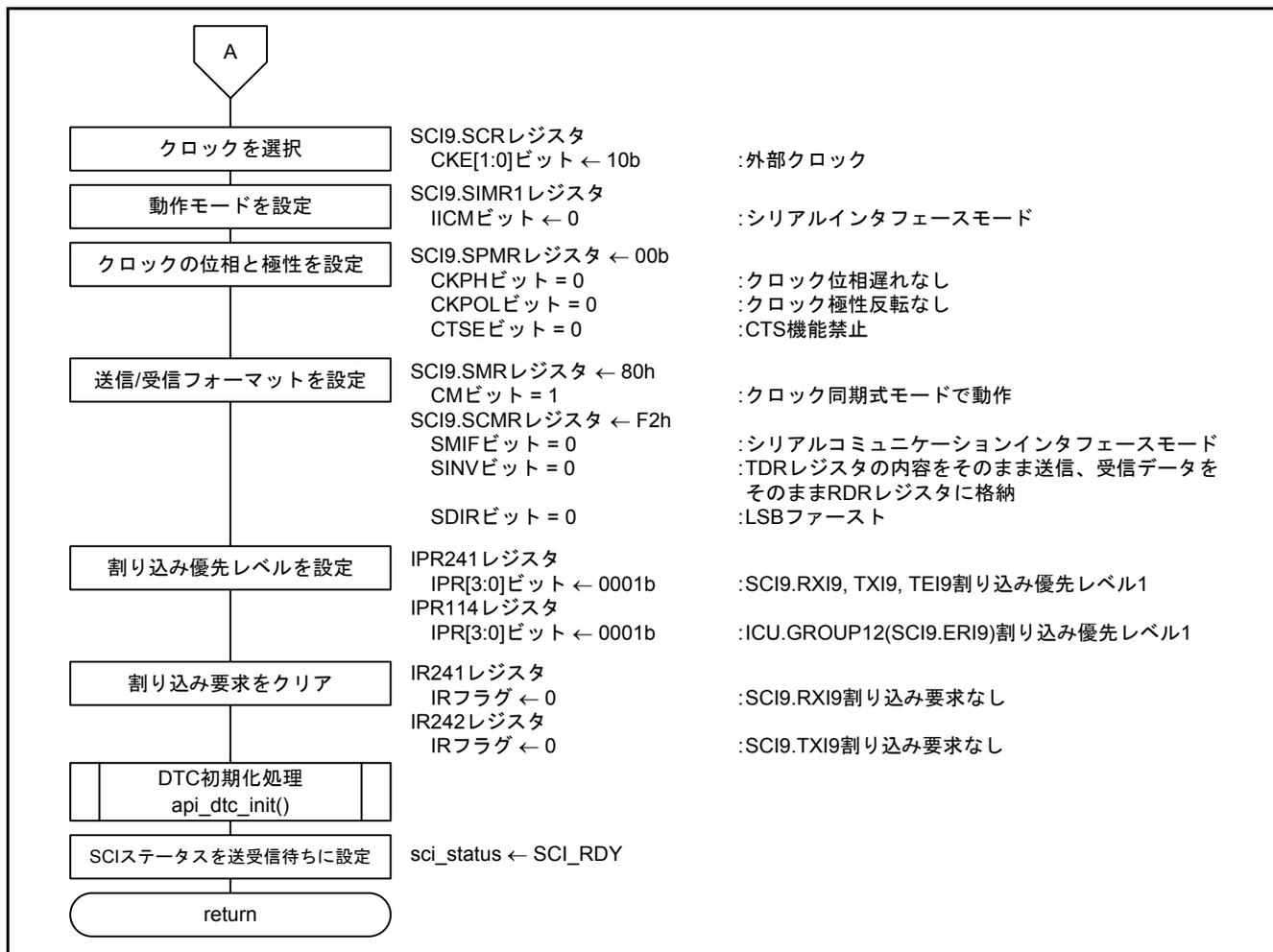


図5.10 SCI 初期設定(2/2)

5.10.7 DTC 初期設定

図 5.11にDTC 初期設定のフローチャートを示します。

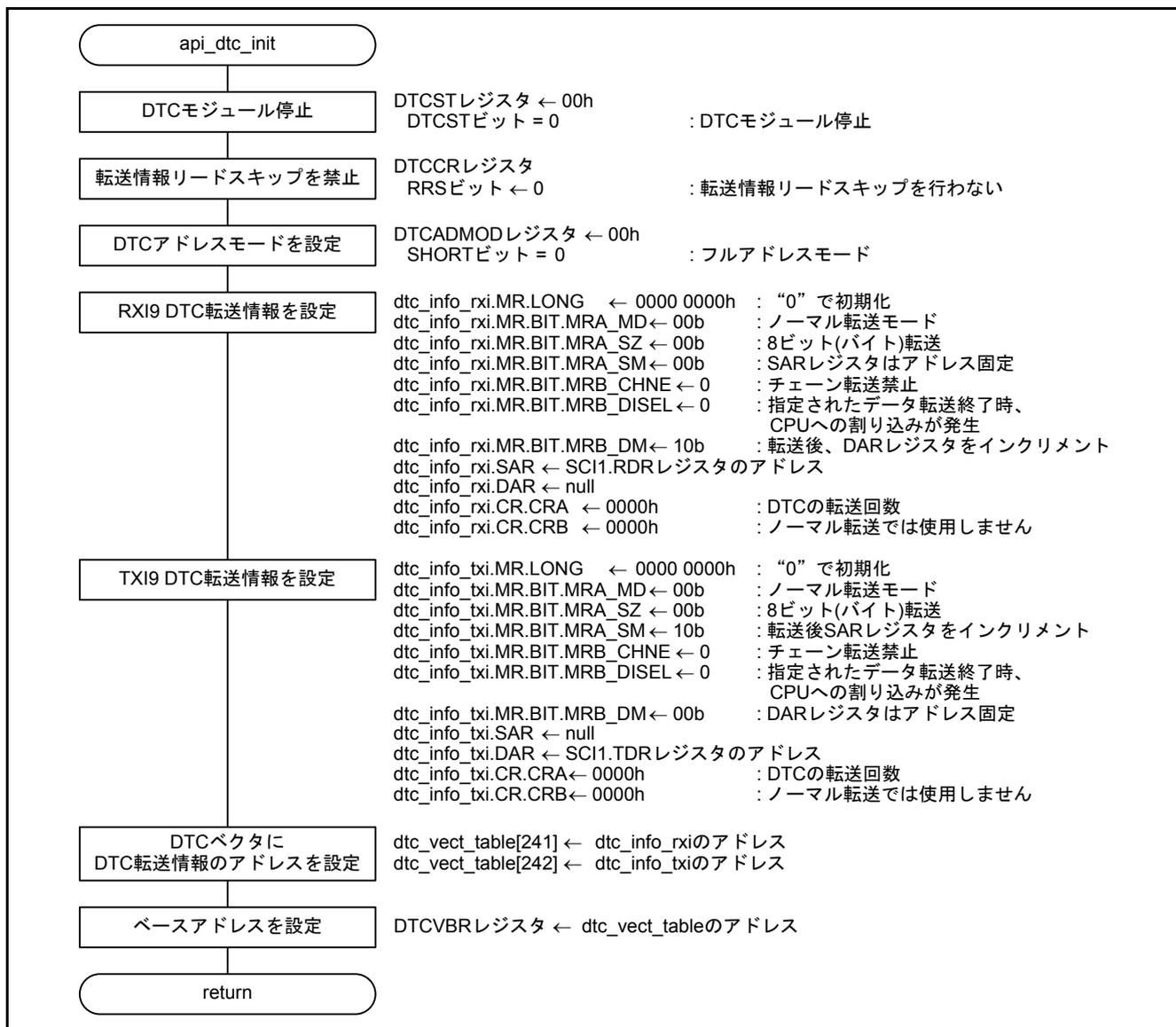


図5.11 DTC 初期設定

5.10.8 SCI9 送受信開始処理

図 5.12~図 5.14にSCI9 送受信開始処理のフローチャートを示します。

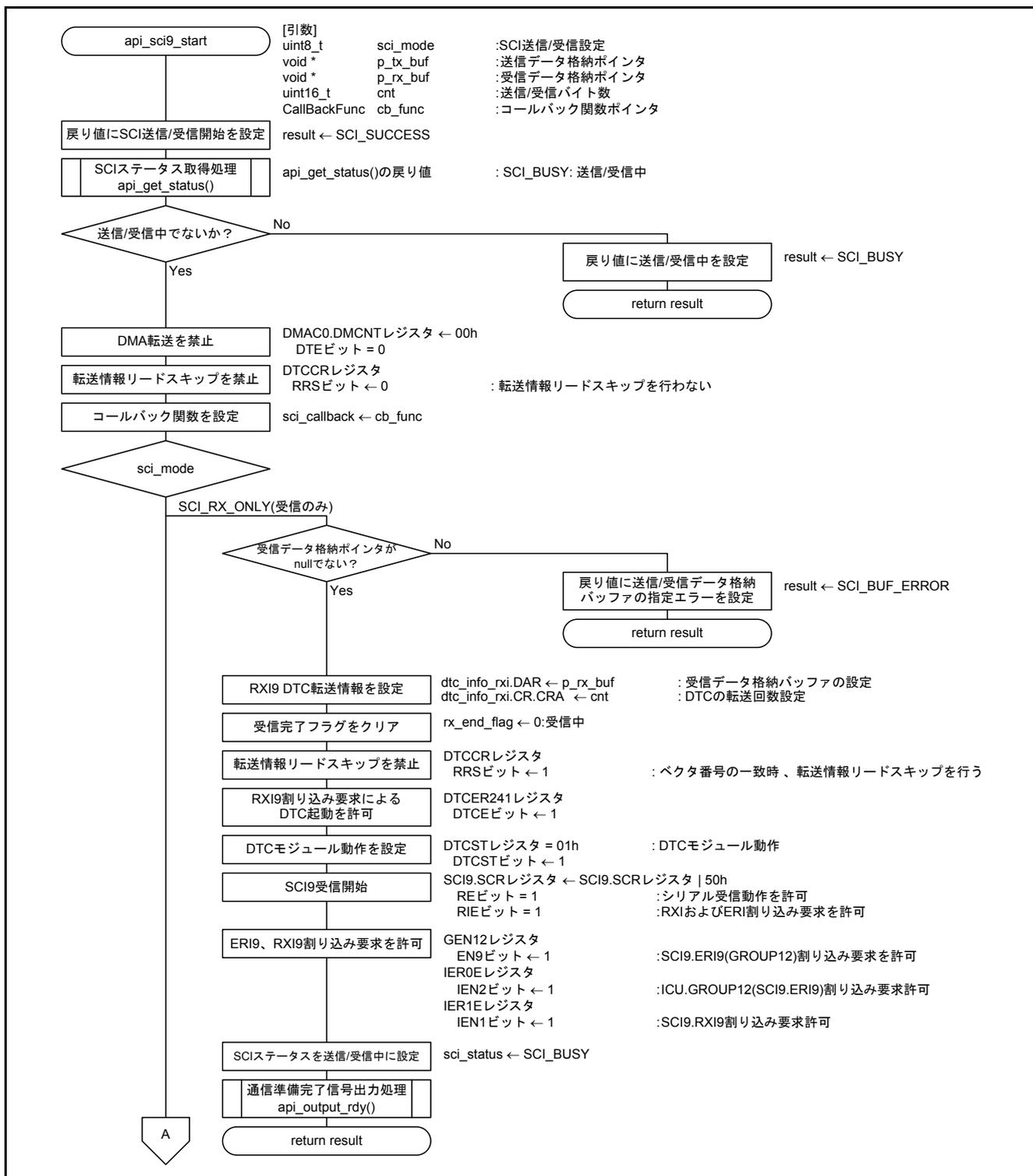


図5.12 SCI9 送受信開始処理(1/3)

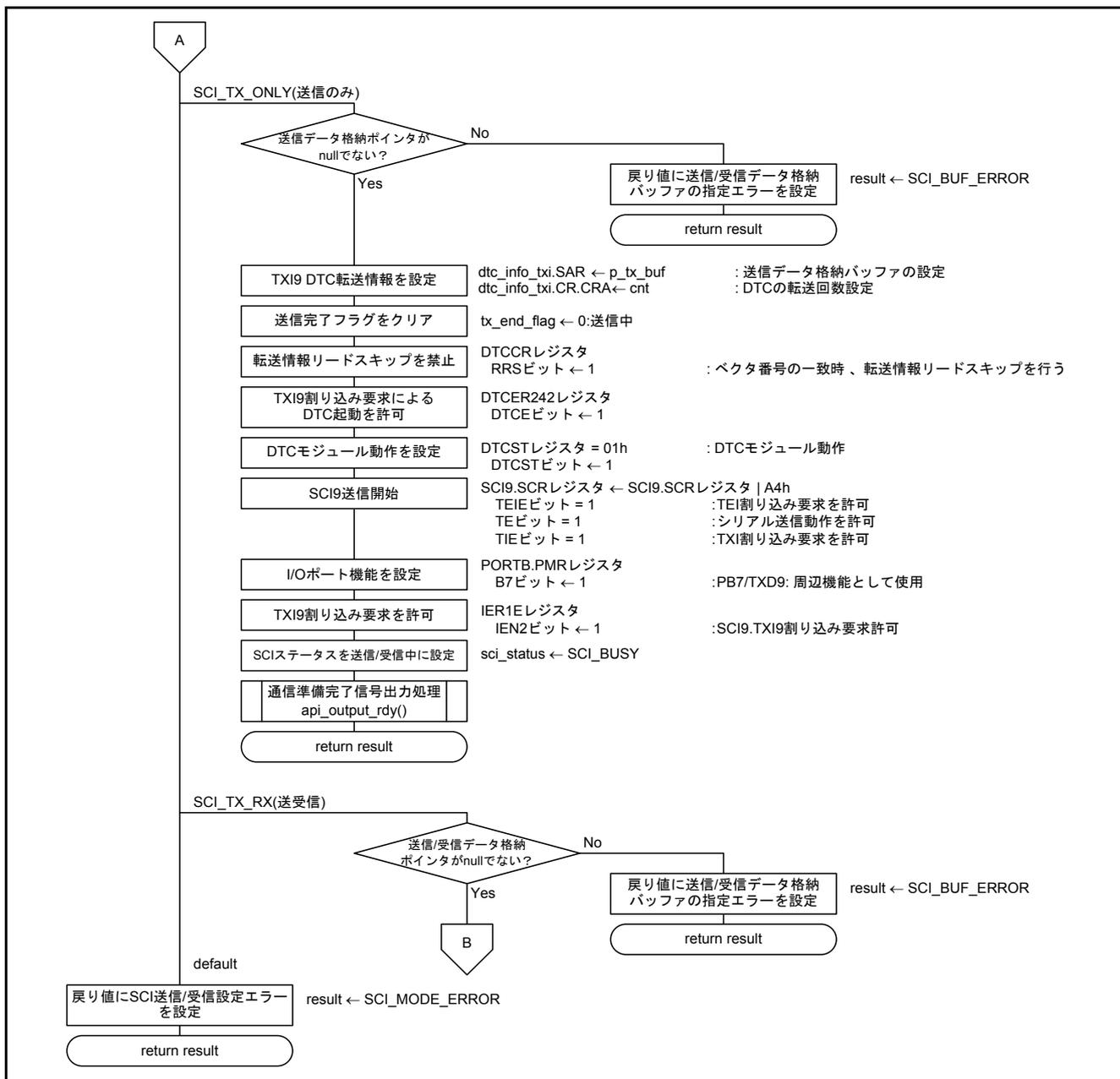


図5.13 SCI 送受信開始処理(2/3)

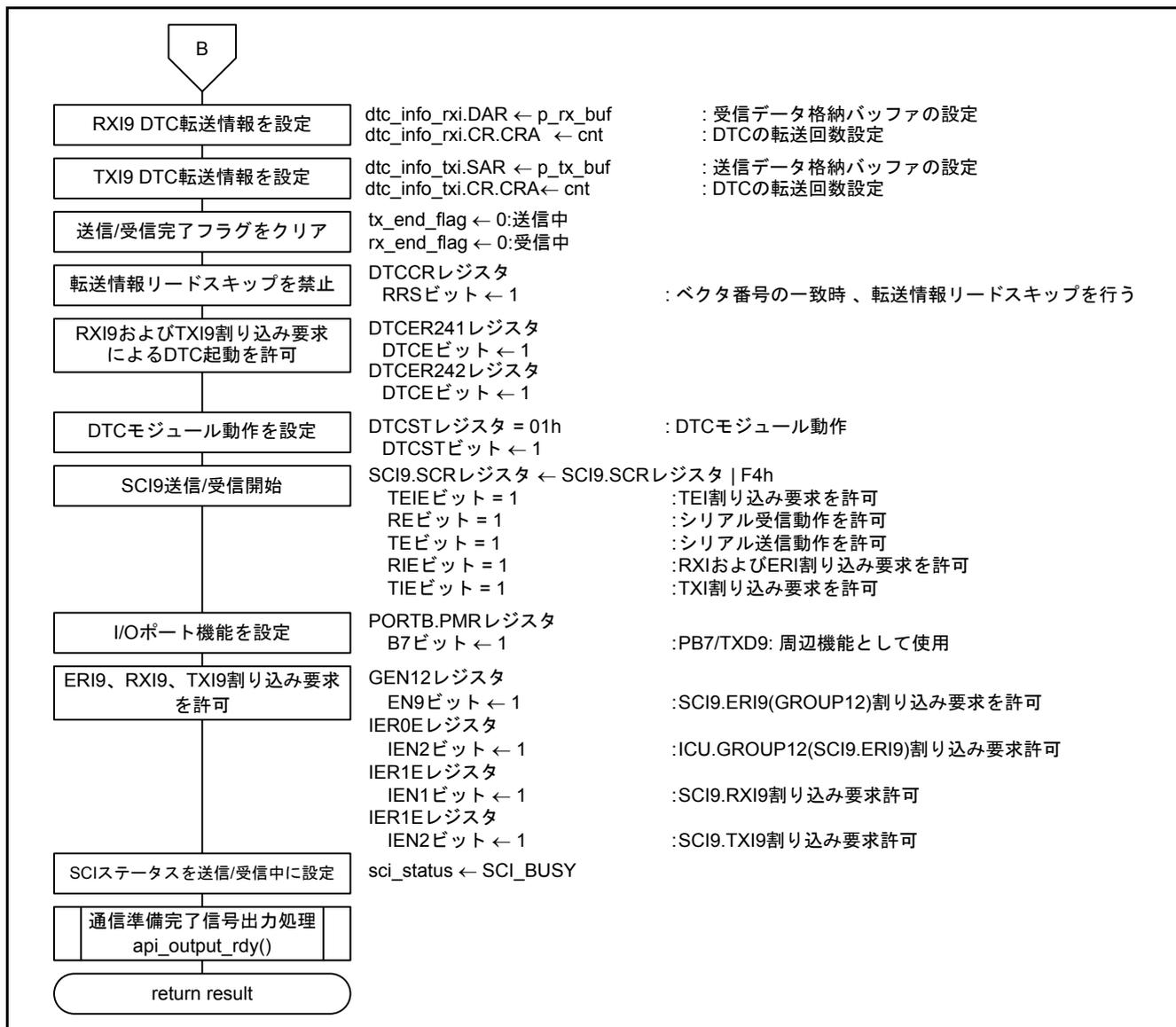


図5.14 SC1 送受信開始処理(3/3)

5.10.9 SCI 送受信停止処理

図 5.15にSCI 送受信停止処理のフローチャートを示します。

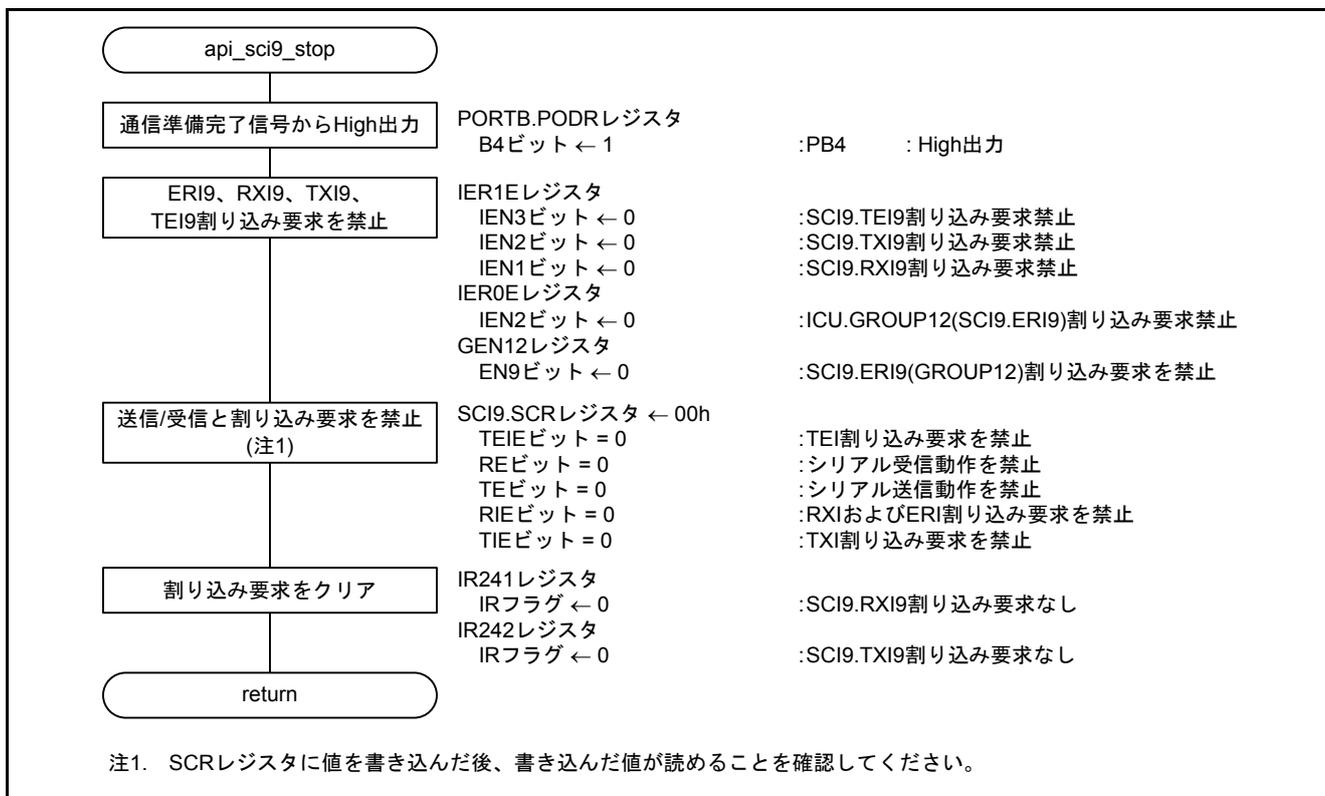


図5.15 SCI 送受信停止処理

5.10.10 通信準備完了信号出力処理

図 5.16に通信準備完了信号出力処理のフローチャートを示します。

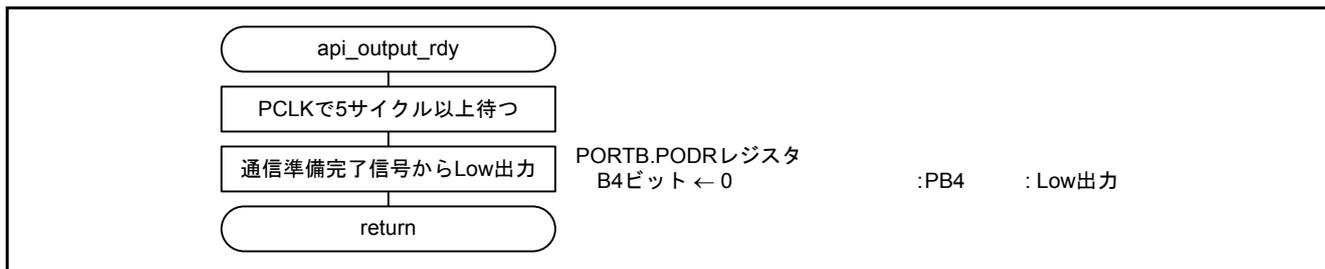


図5.16 通信準備完了信号出力処理

5.10.11 SCI ステータス取得処理

図 5.17にSCI ステータス取得処理のフローチャートを示します。

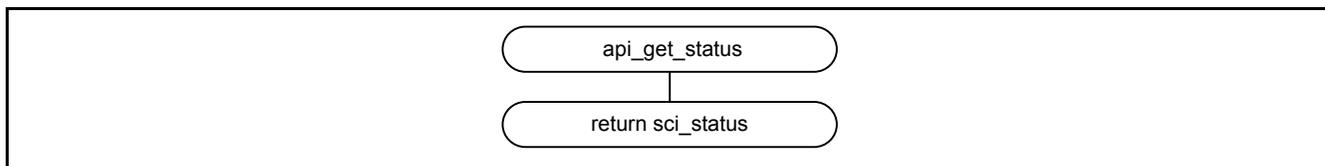


図5.17 SCI ステータス取得処理

5.10.12 RXI9 割り込み処理

図 5.18にRXI9 割り込み処理のフローチャートを示します。

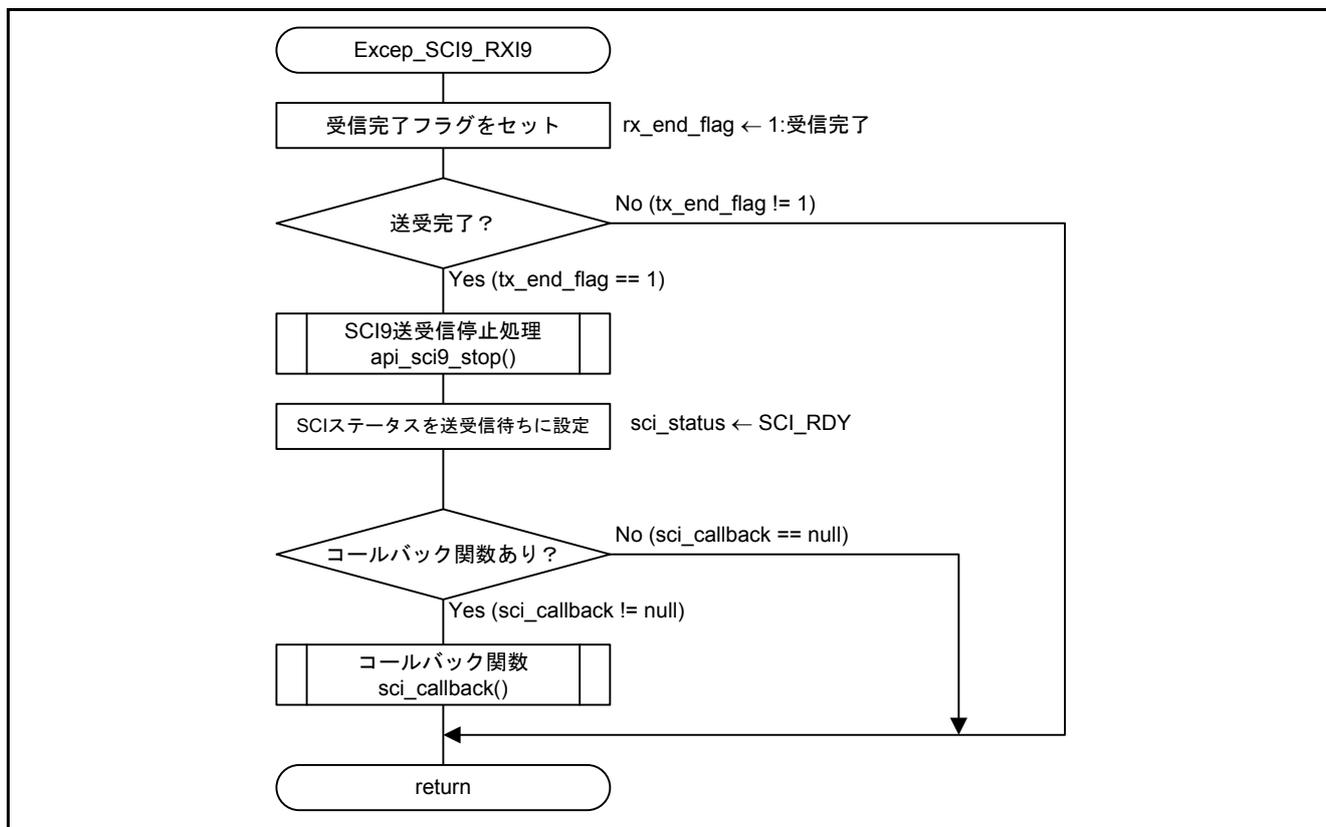


図5.18 RXI9 割り込み処理

5.10.13 TXI9 割り込み処理

図 5.19にTXI9 割り込み処理のフローチャートを示します。

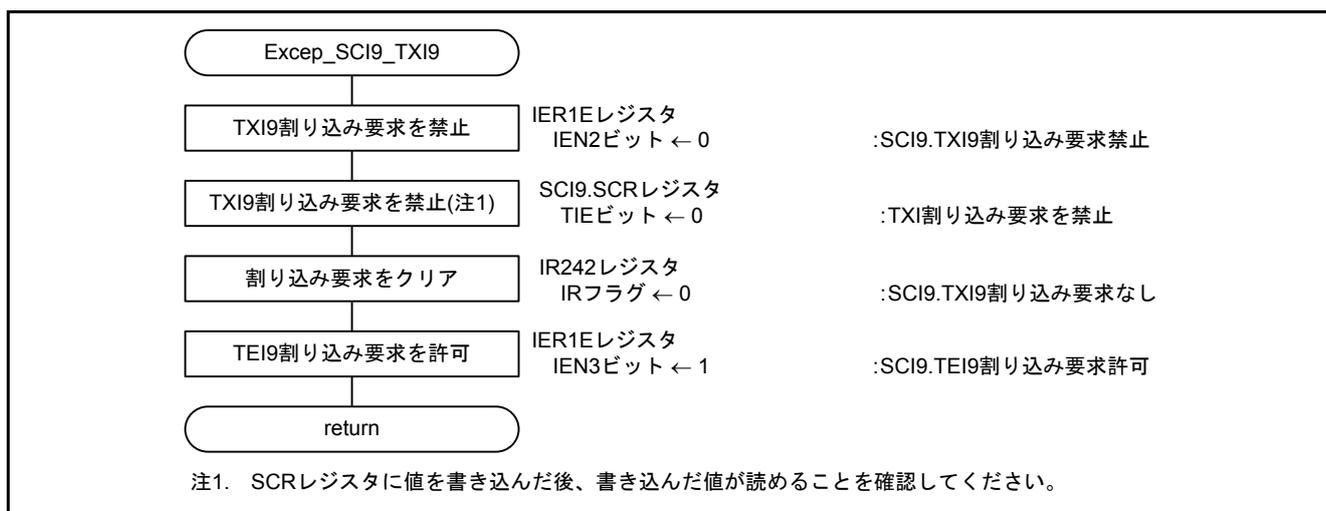


図5.19 TXI9 割り込み処理

5.10.14 TEI9 割り込み処理

図 5.20にTEI9 割り込み処理のフローチャートを示します。

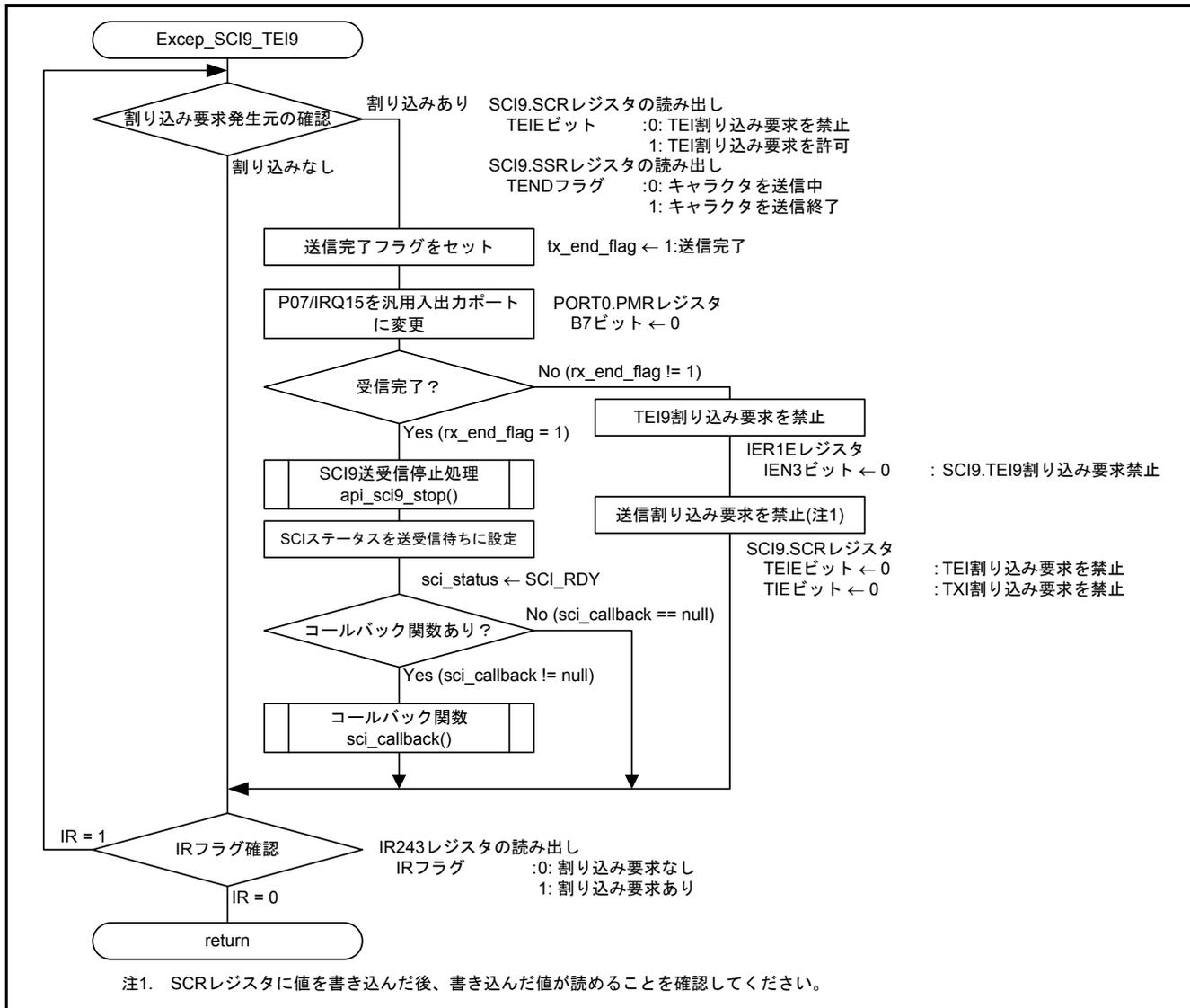


図5.20 TEI9 割り込み処理

5.10.15 GROUP12 割り込み(ERI9 割り込み)処理

図 5.21にGROUP12 割り込み(ERI9 割り込み)処理のフローチャートを示します。

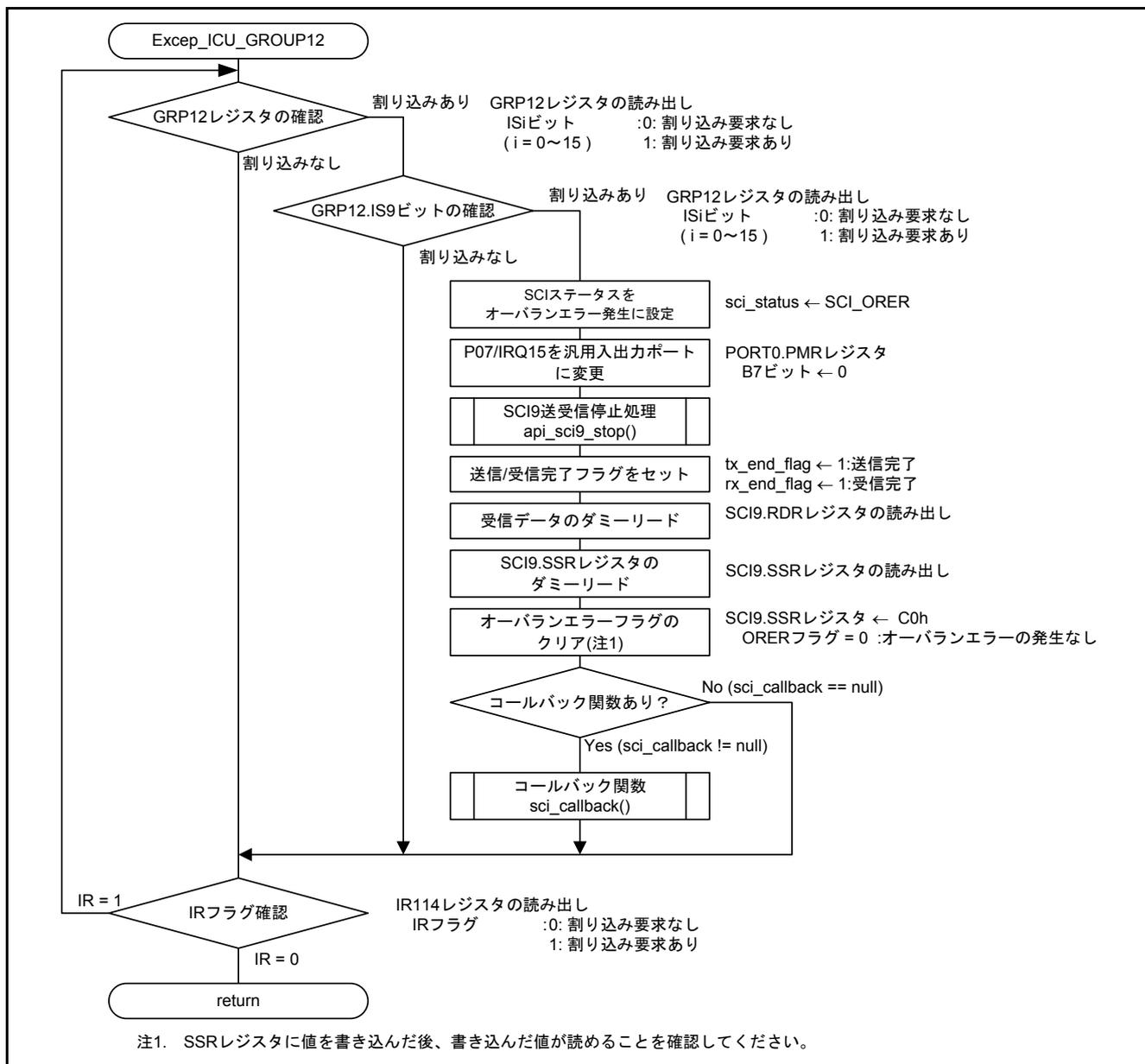


図5.21 GROUP12 割り込み(ERI9 割り込み)処理

## 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

RX63N グループ、RX631グループ ユーザーズマニュアル ハードウェア編 Rev.1.70 (R01UH0041JJ)  
(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>

お問合せ先

<http://japan.renesas.com/contact/>

改訂記録	RX63N グループ、RX631 グループ アプリケーションノート DTC によるクロック同期式 SCI 送受信(スレーブ)
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.01.06	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 製品ご使用上の注意事項

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

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

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

### 3. リザーブアドレスのアクセス禁止

【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/>