

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
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  
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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。

## SH7730 グループ

### SCIF 調歩同期式 送受信設定例

#### 要旨

この資料は FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF) 機能の調歩同期式モードの送受信設定例を掲載しています。

#### 動作確認デバイス

SH7730

#### 目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムについて.....	4
4. 参考プログラム例.....	29
5. 実行結果.....	60
6. 参考ドキュメント.....	61

## 1. はじめに

### 1.1 仕様

- SCIF チャンネル 0 (SCIF0) を、調歩同期式モードの送信モードとして初期化します。
- SCIF チャンネル 1 (SCIF1) を、調歩同期式モードの受信モードとして初期化します。
- SCIF0 から正常データ、BRK 信号、正常データを順番に送信し、SCIF1 で SCIF0 からの送信データを受信します。

### 1.2 使用機能

- SCIF チャンネル 0
- SCIF チャンネル 1

### 1.3 適用条件

評価ボード:	アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
外付けメモリ (エリア 0):	NOR 型フラッシュメモリ 4 M バイト Spansion 製 S29AL032D70TFI04
(エリア 3):	SDR-SDRAM 32 M バイト (16 M バイト × 2 個) Samsung 製 K4S281632F-UC75
マイコン:	SH7730 (R8A77301)
動作周波数	CPU クロック: 266.66 MHz SuperHyway バスクロック: 133.33 MHz バスクロック: 66.66 MHz 周辺クロック: 33.33 MHz
エリア 0 バス幅:	16 ビット固定 (MD3 端子 = Low レベル)
クロック動作モード:	モード 2 (MD0 端子 = Low レベル, MD1 端子 = High レベル)
エンディアン:	ビッグエンディアン (MD5 端子 = Low レベル)
ツールチェーン:	ルネサス テクノロジ製 SuperH RISC engine Standard Toolchain Ver.9.2.0.0
コンパイルオプション:	High-performance Embedded Workshop でのデフォルト設定 <code>(-cpu=sh4a -include="\$(PROJDIR)¥inc"</code> <code>-object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -optimize=0</code> <code>-gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all</code> <code>-infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)</code>

### 1.4 関連アプリケーションノート

本資料の参考プログラムは、「SH7730 グループ アプリケーションノート SH7730 初期設定例 (RJJ06B0864)」の設定条件で動作確認しています。そちらもあわせてご参照ください。

## 2. 応用例の説明

本応用例では、FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF) を使用して、調歩同期式モードで SCIF0 から SCIF1 へデータを送信し、SCIF1 で SCIF0 からのデータを受信します。

### 2.1 使用機能の動作概要

SCIF の調歩同期式モードでは、通信開始を意味するスタートビットと通信終了を意味するストップビットをデータに付加したキャラクタを送信/受信し、1 キャラクタ単位で同期をとりながら通信を行います。クロックソースとして内部クロックまたは、SCK 端子からの外部クロック入力を選択できます。通信モードとして転送データフォーマット、転送速度などが設定できます。表 1 に調歩同期式モードの概要を示します。SCIF ブロック図については、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 22 章 図 22.1 SCIF のブロック図」を参照ください。

表 1 SCIF (調歩同期式モード) の概要

項目	概要
チャネル数	4 チャネル (SCIF0 ~ SCIF3)
クロックソース	内部クロック: P $\phi$ , P $\phi$ /4, P $\phi$ /16, P $\phi$ /64 P $\phi$ : 内蔵周辺クロック 外部クロック: SCK0 ~ SCK3 端子入力クロック 外部クロックを選択した場合には、ビットレートの 16 倍の周波数のクロックを入力してください
データフォーマット	転送データ長: 7 ビット, または 8 ビット 転送順序: LSB ファースト固定 スタートビット: 1 ビット固定 ストップビット: 1 ビット, または 2 ビット パリティビット: 偶数パリティ, 奇数パリティ, パリティなし
ボーレート	内部クロックを選択時: 62.94bps ~ 1031.25kbps (P $\phi$ = 33.33MHz 動作時) 外部クロックを選択時: 最大 515.62kbps (P $\phi$ = 33.33MHz 外部入力クロック 8.25MHz 動作時)
エラー検出	パリティエラー, フレーミングエラー, オーバランエラー
割り込み要求	送信 FIFO データエンプティ割り込み (TXI) ブレーク割り込み (BRI) 受信 FIFO データフル割り込み (RXI) 受信エラー割り込み (ERI)
その他	<ul style="list-style-type: none"> <li>独立した送信部と受信部を備えているので、送信と受信を同時に行うことが可能</li> <li>送信部および受信部ともに 16 段の FIFO バッファ構造になっているのでシリアルデータの連続送信, 連続受信が可能</li> <li>ブレークの検出が可能</li> <li>消費電力低減のために、未使用チャネルのクロック供給を停止させることが可能</li> <li>モデムコントロール機能 (RTS および CTS) を内蔵 (チャネル 2, 3)</li> <li>送信および受信 FIFO データレジスタ内に格納されている有効データ数, および受信 FIFO データレジスタ内に格納されている受信エラー数を検出可能</li> <li>調歩同期式モードにおいて、受信時、タイムアウトエラー (DR) を検出可能</li> </ul>

【注】 SCIF についての詳細は、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 22 章 FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF)」の章を参照ください。

### 2.2 調歩同期式モードの動作

SCIF の調歩同期式モードの動作については、「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 22 章 FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF) 22.4.2 調歩同期式モードの動作」の章を参照ください。

### 3. 参考プログラムについて

参考プログラムでは、主に以下のことを行います。

- (1) SCIF0 を調歩同期式の送信モード、SCIF1 を調歩同期式の受信モードとして使用します。
- (2) 全送信データを"1234567890ABCDEFGHIJKLMNPOQRSTUVWXYZ" (36 バイト) とします。1 回目 (全送信データの先頭 17 バイト分)、BRK 信号、2 回目 (全送信データの残り 19 バイト分) の順番で SCIF0 からデータ送信を行います。
- (3) 1 回目の送信処理で SCIF0 から全送信データの先頭 17 バイト分を送信します。送信 FIFO (SCFTDR0) の空き数分、送信データを設定します。残りの送信データについては、送信 FIFO データエンプティ割り込み (TXI) 要求のタイミングで、送信 FIFO (SCFTDR0) に順次設定します。
- (4) SCIF1 では、SCIF0 からのデータを順次受信し、受信 FIFO データフル割り込み (RXI) のタイミングで受信 FIFO に格納されているデータを受信データ格納用のデータ領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納します。
- (5) 1 回目の送信処理により、受信データ格納用のデータ領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に SCIF0 から送信されたデータ"1234567890ABCDEFG"が格納されていることを確認します。
- (6) SCIF0 から BRK 信号を送信します。
- (7) SCIF1 で BRK 信号を受信することを確認します。その際に、ブ레이크検出 (SCFSR0.BRK) を参照し、受信データ格納用のデータ領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に、ブ레이크データ (H'00) を格納しないようにします。
- (8) 1 回目の SCIF0 設定条件で再度送信処理を行うために、SCIF0 再送信準備処理を行います。
- (9) 2 回目の送信処理で SCIF0 から残りの送信データ (19 バイト分) を送信します。送信の方法は (3) と同様です。
- (10) 2 回目の送信処理により、受信データ格納用のデータ領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に全送信データ (1 回目の送信データと 2 回目の送信データ"1234567890ABCDEFGHIJKLMNPOQRSTUVWXYZ") が格納されていることを確認します。

#### 3.1 参考プログラム動作環境

以下のように、SCIF0\_TXD 端子と SCIF1\_RXD 端子を接続します。

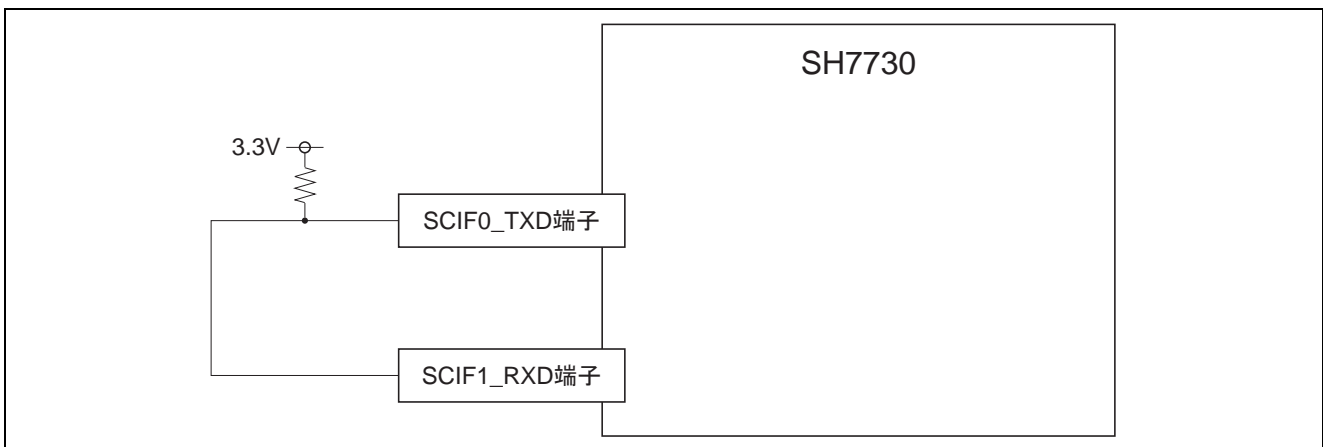


図 1 参考プログラム動作環境

### 3.2 参考プログラムの通信機能設定

表 2 に SCIF0 , 表 3 に SCIF1 の参考プログラムの通信機能設定を示します。

表 2 参考プログラムの通信機能設定 (SCIF0)

通信フォーマット	設定機能
通信モード	調歩同期式
使用チャンネル	チャンネル 0
割り込み	送信 FIFO データエンpty割り込み (TXI) 使用 SCSCR0.TIE = 1 設定 優先度 1
通信速度	115.2kbps
データ長	8 ビットデータ
パリティ	なし
ストップビット	1 ストップビット
モデムコントロール	RTS/CTS 機能無効
トランスミット FIFO データ数トリガ	8 本参考プログラムでは, #define 値 (D_SCIF_DATA_NUM_SND_TRIGGER_CH0) の変更により変更可能です。

表 3 参考プログラムの通信機能設定 (SCIF1)

通信フォーマット	設定機能
通信モード	調歩同期式
使用チャンネル	チャンネル 1
割り込み	受信 FIFO データフル割り込み (RXI) 使用 SCSCR1.RIE = 1 設定 優先度 2 本応用例では ORER (オーバラン) エラーが発生しないように送信処理の SCIF0 より優先度を高くしています。
通信速度	115.2kbps
データ長	8 ビットデータ
パリティ	なし
ストップビット	1 ストップビット
モデムコントロール	RTS/CTS 機能無効
レシーブ FIFO データ数トリガ	4 本参考プログラムでは, #define 値 (D_SCIF_DATA_NUM_RCV_TRIGGER_CH1) の変更により変更可能です。 本応用例では ORER (オーバラン) エラーが発生しないようにトランスミット FIFO データ数トリガより小さい値を設定しています。

### 3.3 参考プログラムの処理フロー

ここでは、本参考プログラムの処理フローを説明します。

#### 3.3.1 参考プログラムメイン処理

メイン処理の処理フローについては、図 2、図 3 を参照ください。



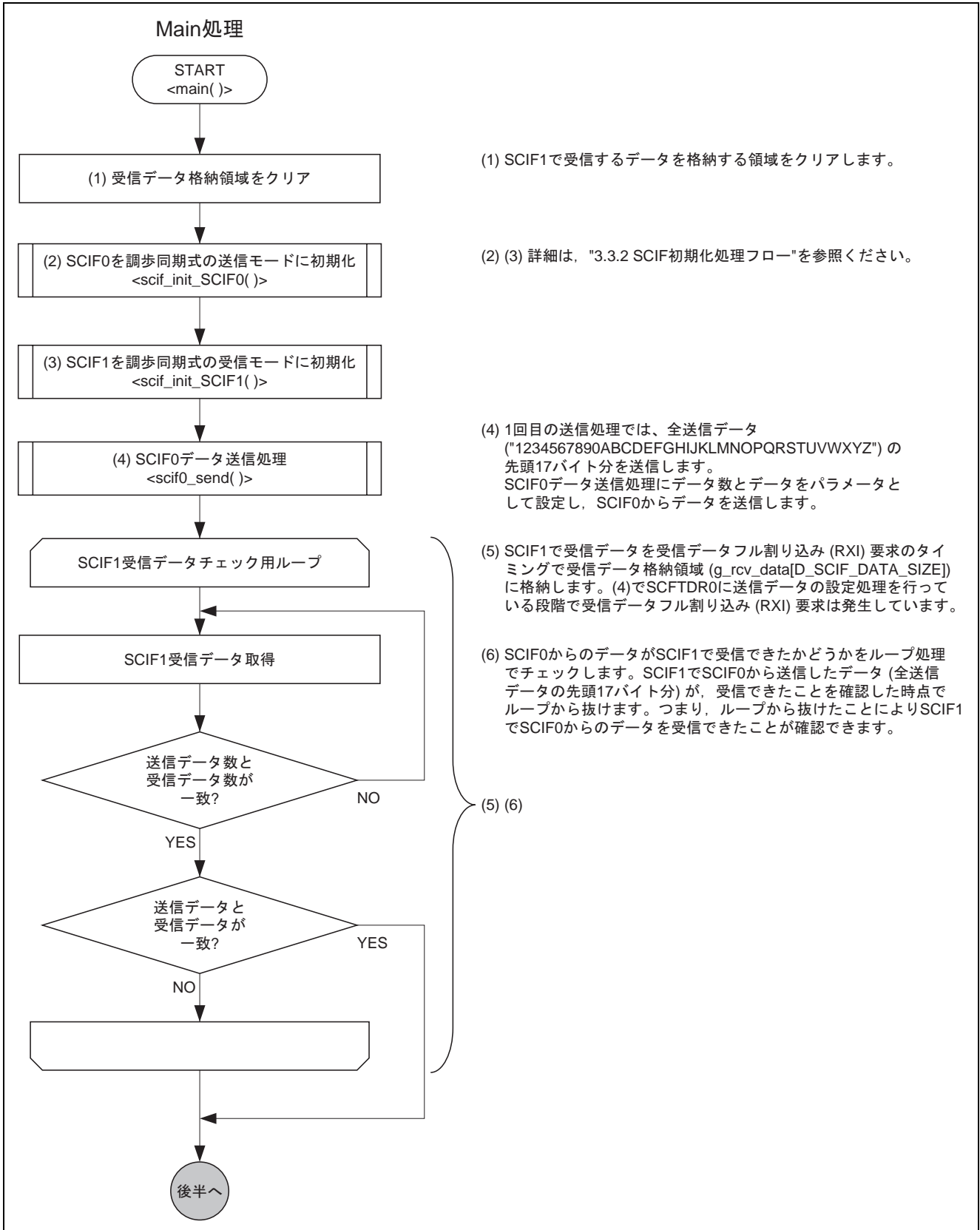


図2 参考プログラムのMain処理フロー (前半)

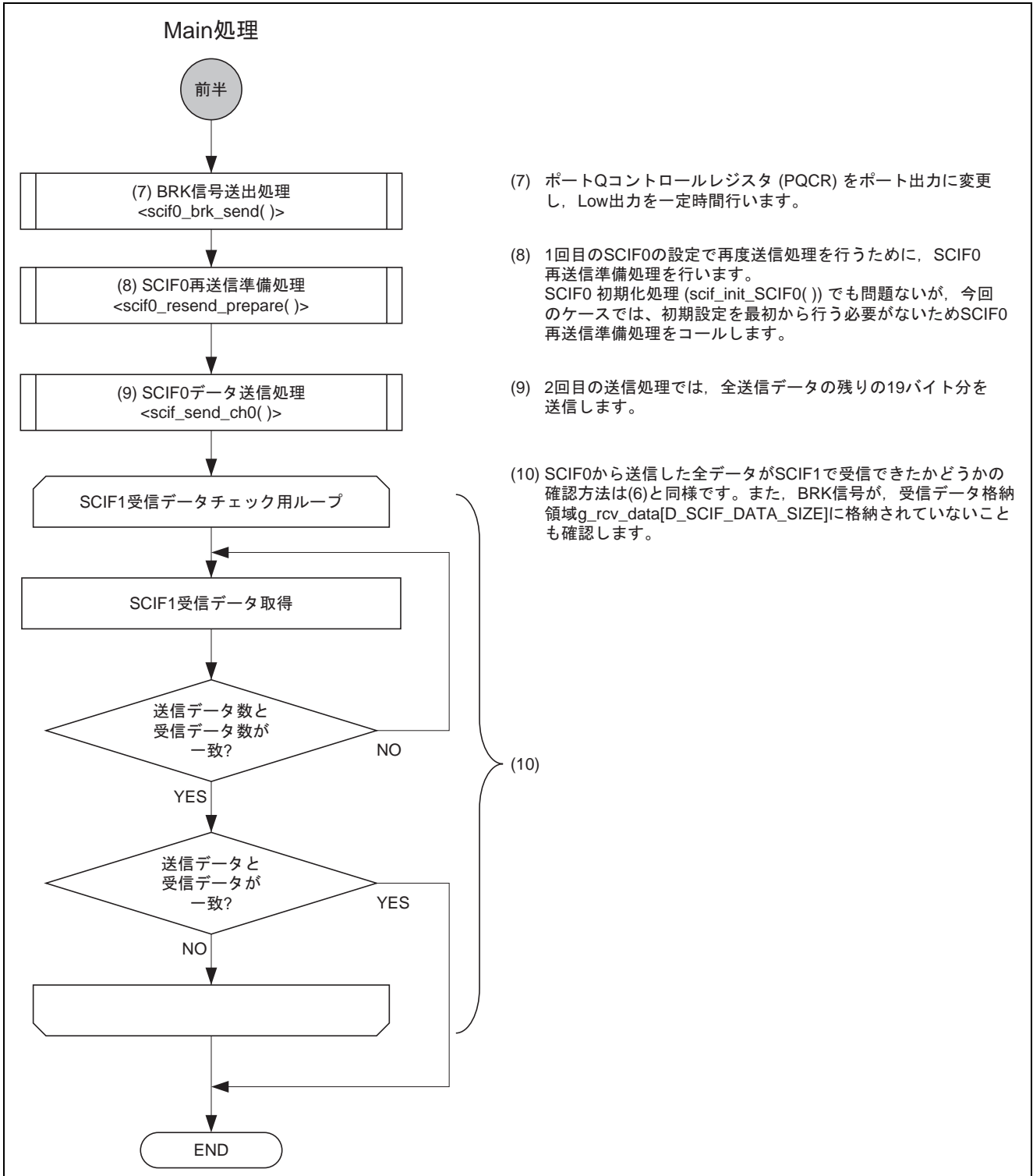


図3 参考プログラムのMain処理フロー (後半)

### 3.3.2 SCIF 初期化処理フロー

調歩同期式モードでの送信初期設定フローについては、図4、図5を参照ください。

調歩同期式モードでの受信初期設定フローについては、図6、図7を参照ください。

SCIF0調歩同期式送信モード初期化処理

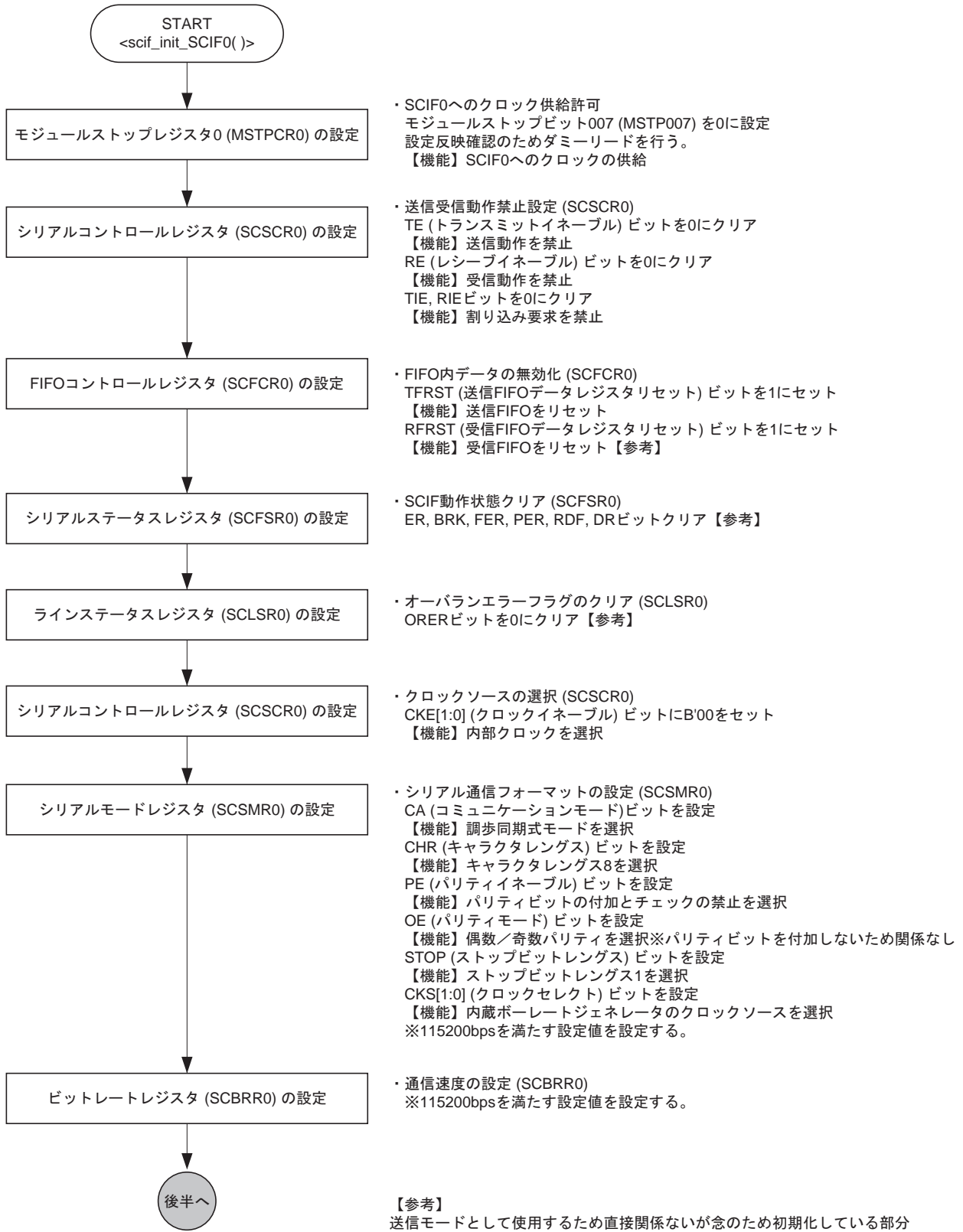


図4 調歩同期式モード送信初期設定フロー例 (前半)

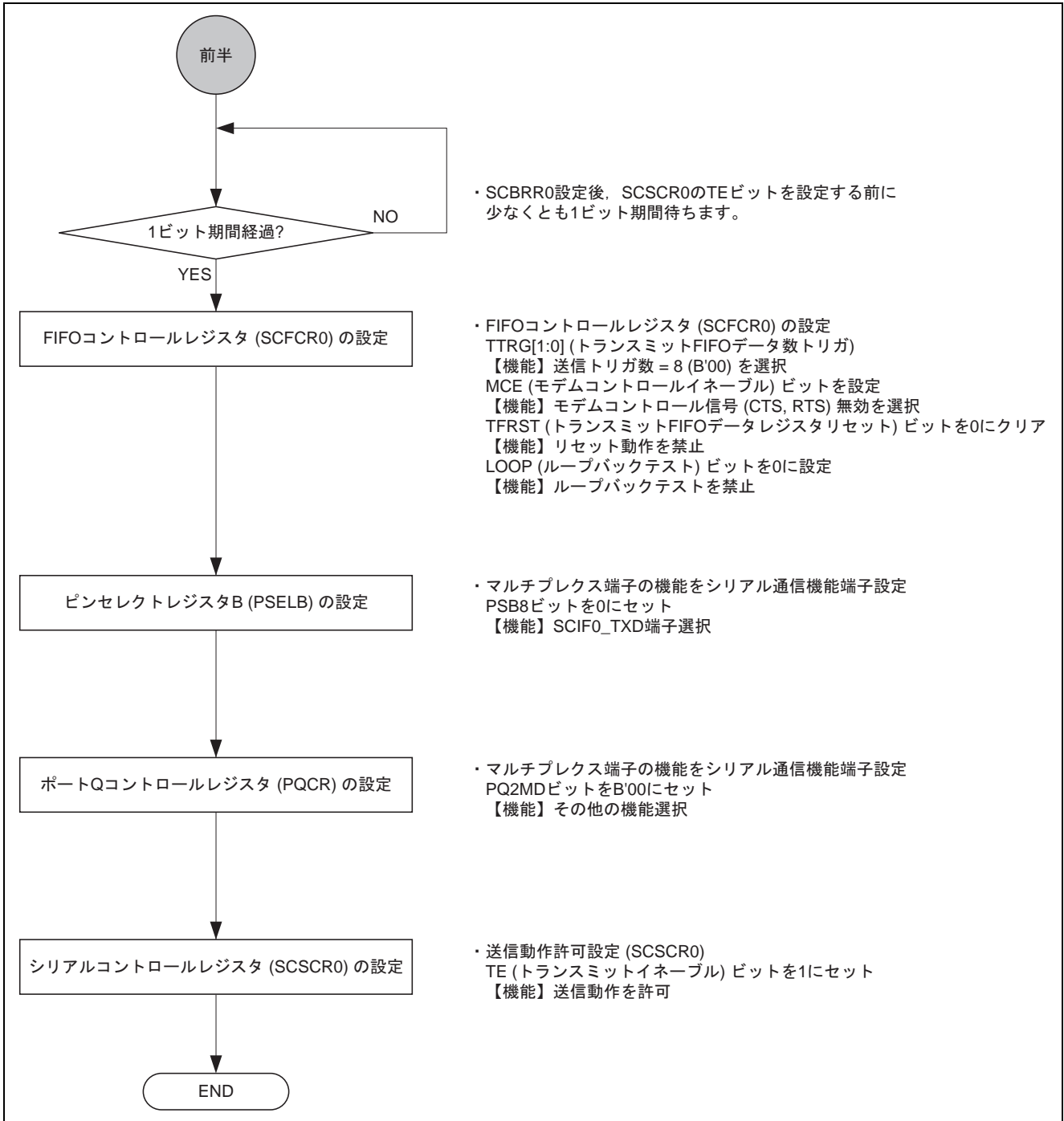


図5 調歩同期式モード送信初期設定フロー例 (後半)

SCIF1 調歩同期式受信モード初期化処理

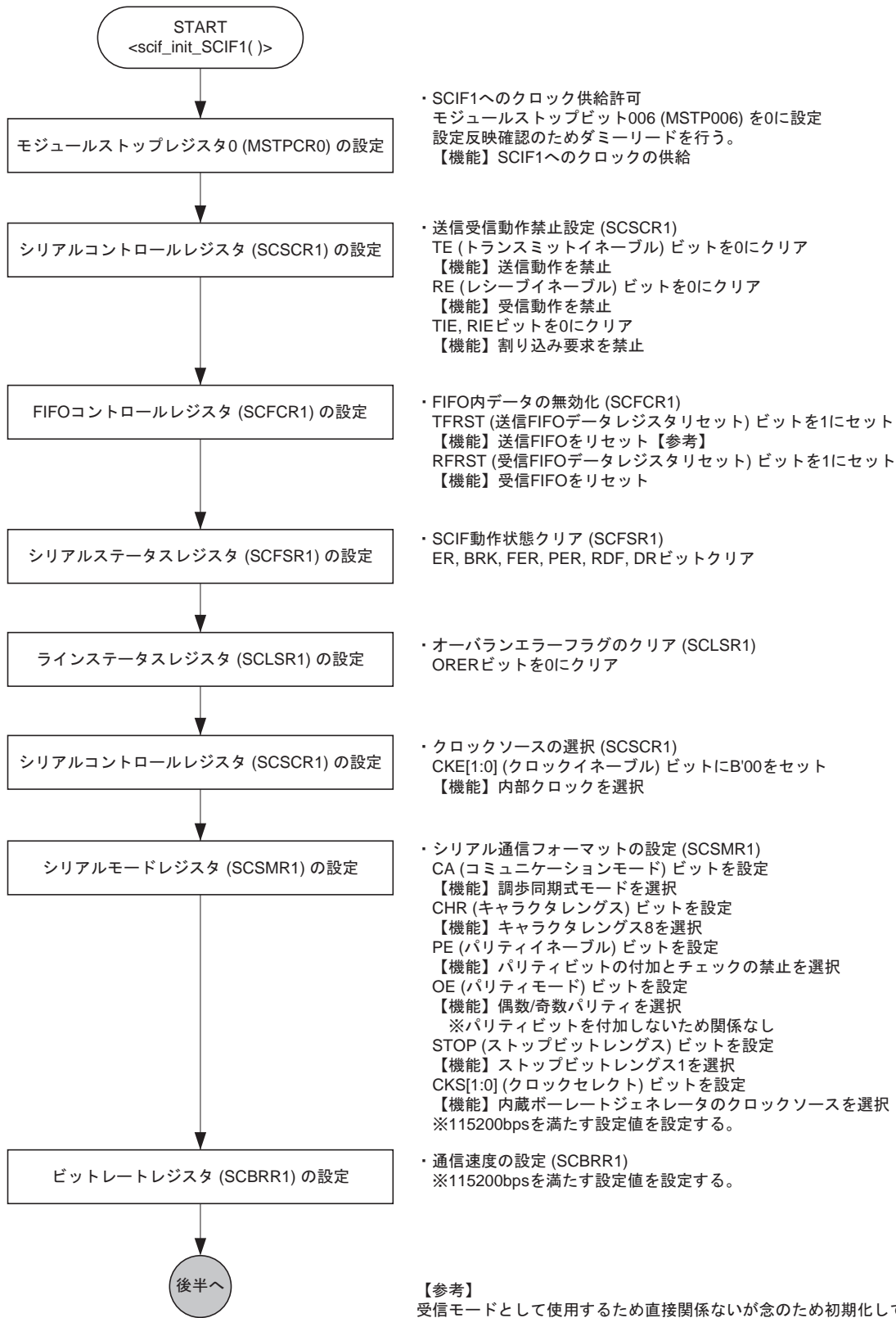


図 6 調歩同期式モード受信初期設定フロー例 (前半)

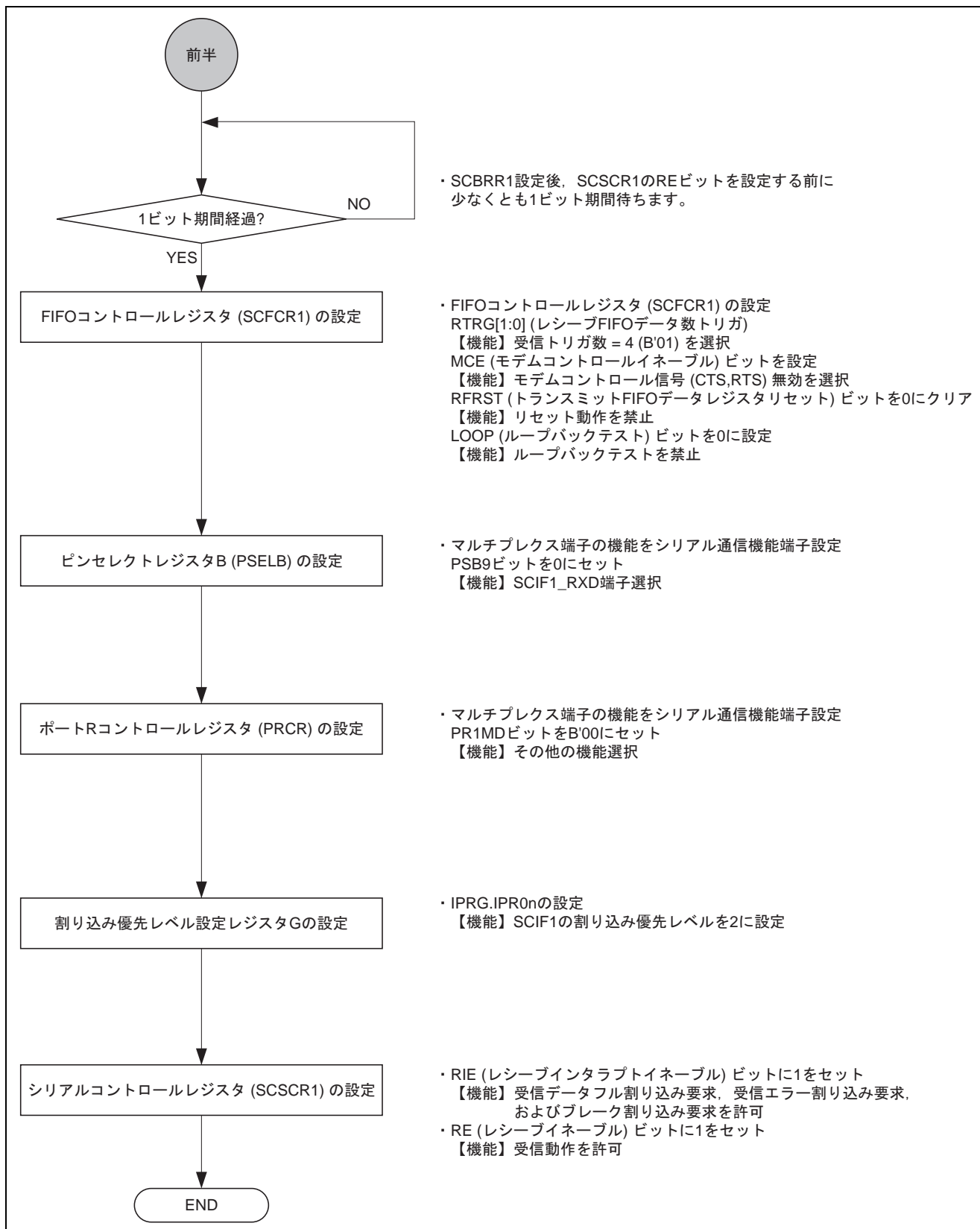


図7 調歩同期式モード受信初期設定フロー例 (後半)

3.3.3 SCIF0 データ送信処理フロー

データ送信処理を開始する際に、パラメータに送信データ数、送信データを設定し SCIF0 データ送信処理 (scif0\_send()) をコールします。(図 8)

SCIF0 データ送信処理 (scif0\_send()) で、SCIF0 送信データ設定処理 (scif0\_setdata()) をコールします。SCIF0 送信データ設定処理 (scif0\_setdata()) で、送信データが存在する場合に、SCFTDR0 データ設定処理 (scif\_setdata\_to\_SCFTDR\_ch0()) をコールします (図 9)。SCFTDR0 データ設定処理 (scif\_setdata\_to\_SCFTDR\_ch0()) で、送信 FIFO の空き数分、送信データを SCFTDR0 に設定します。この時に、送信データを SCFTDR0 に設定後、送信 FIFO データエンブティ割り込み (TXI) 要求を許可にします (図 10)。

送信 FIFO データエンブティ割り込み (TXI) 発生時にコールされる SCIF0 割り込み処理で、SCIF0 送信データ設定処理 (scif0\_setdata()) をコールし、残送信データを順次 SCFTDR0 に設定します。(図 11)

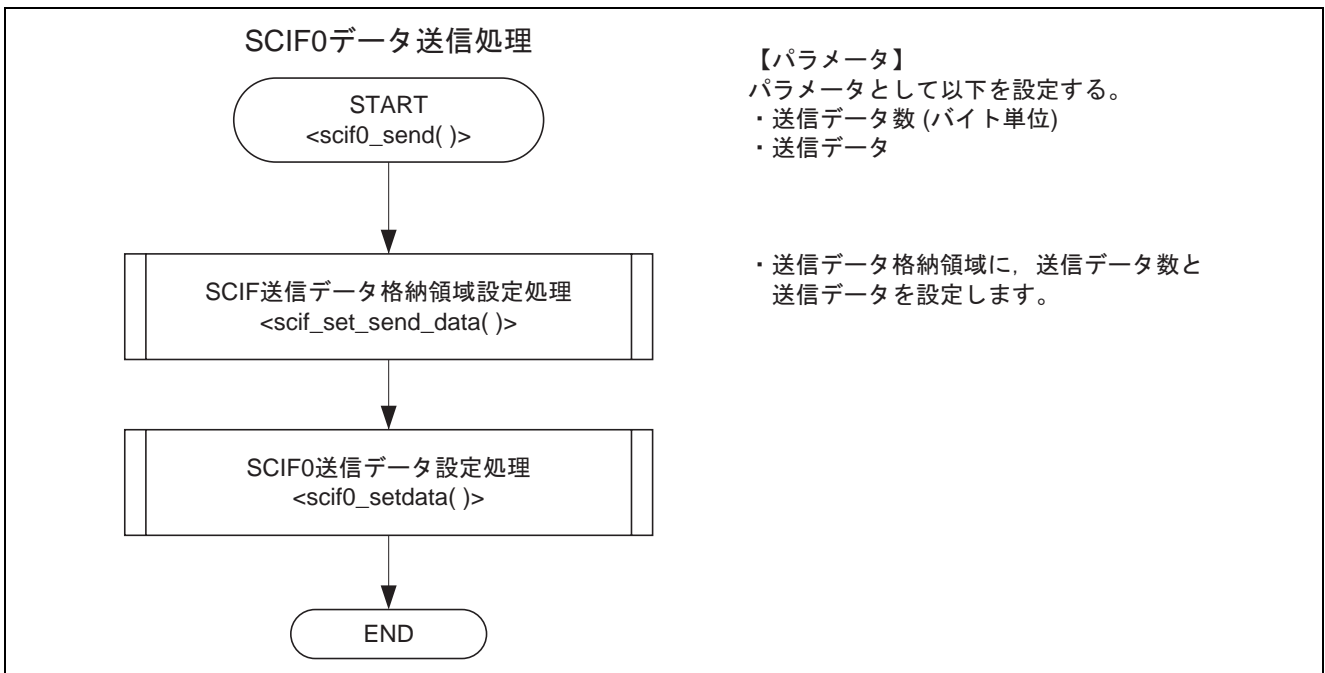


図 8 SCIF0 データ送信処理フロー



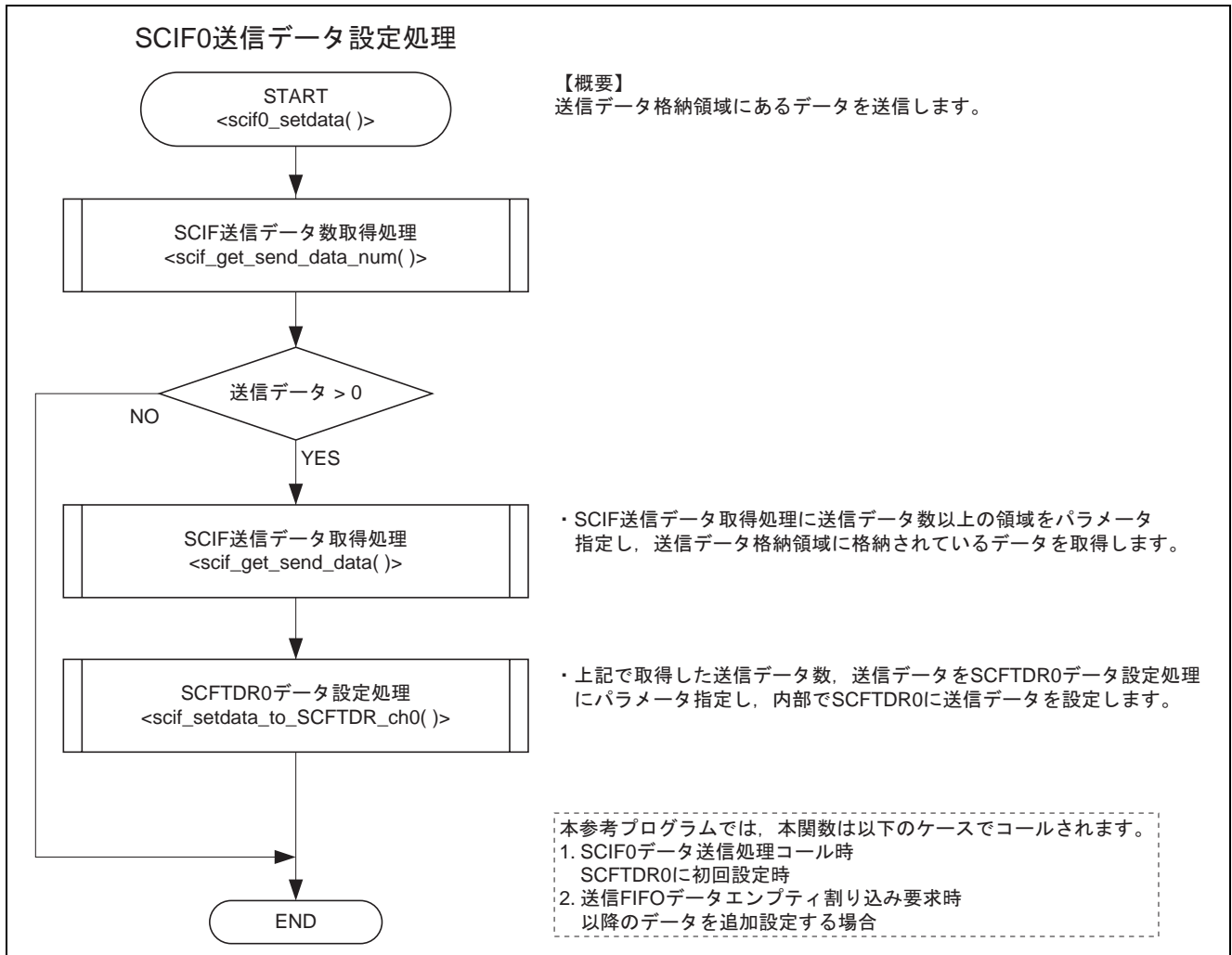


図 9 SCIF0 送信データ設定処理フロー

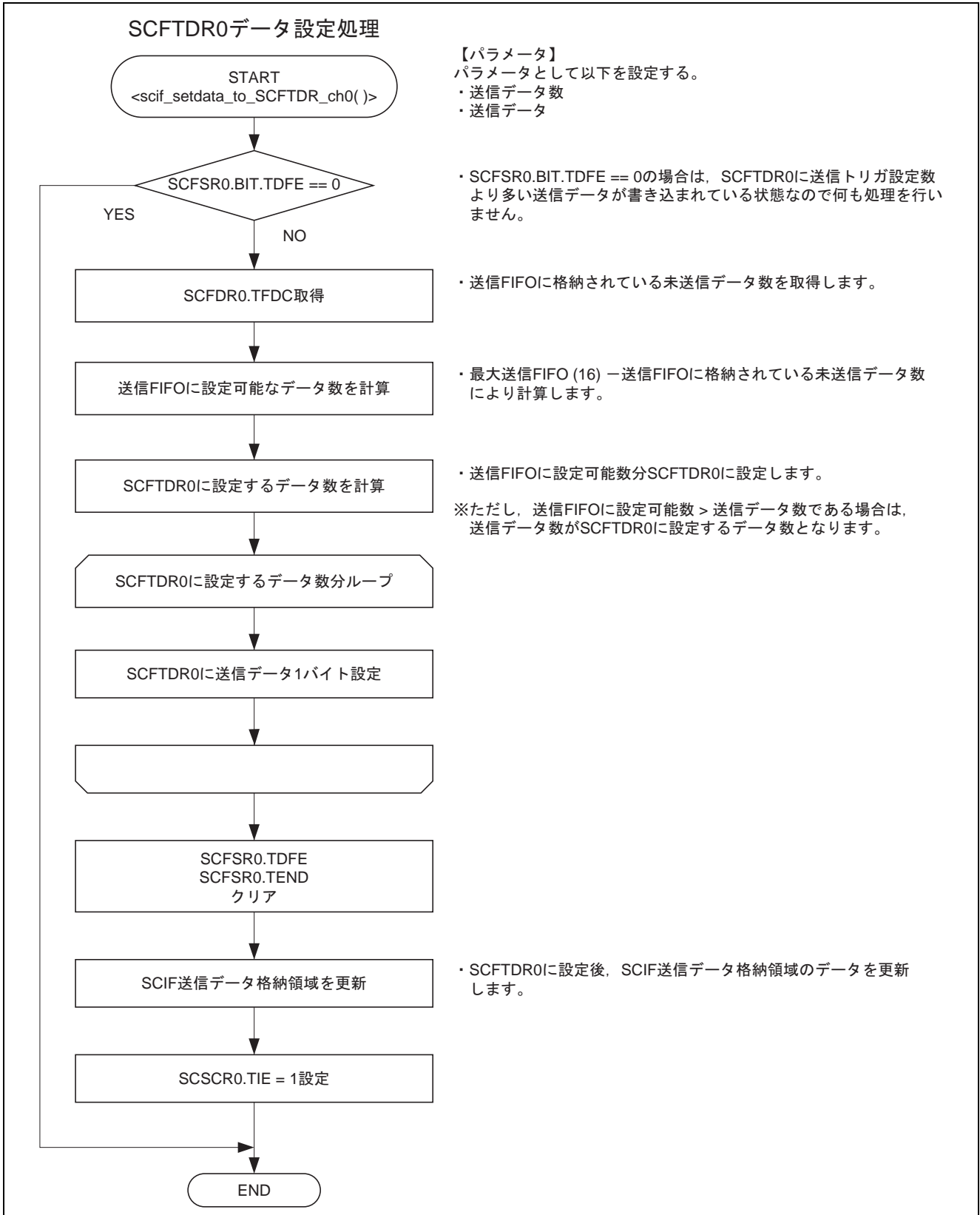


図 10 SCFTDR0 データ設定処理フロー

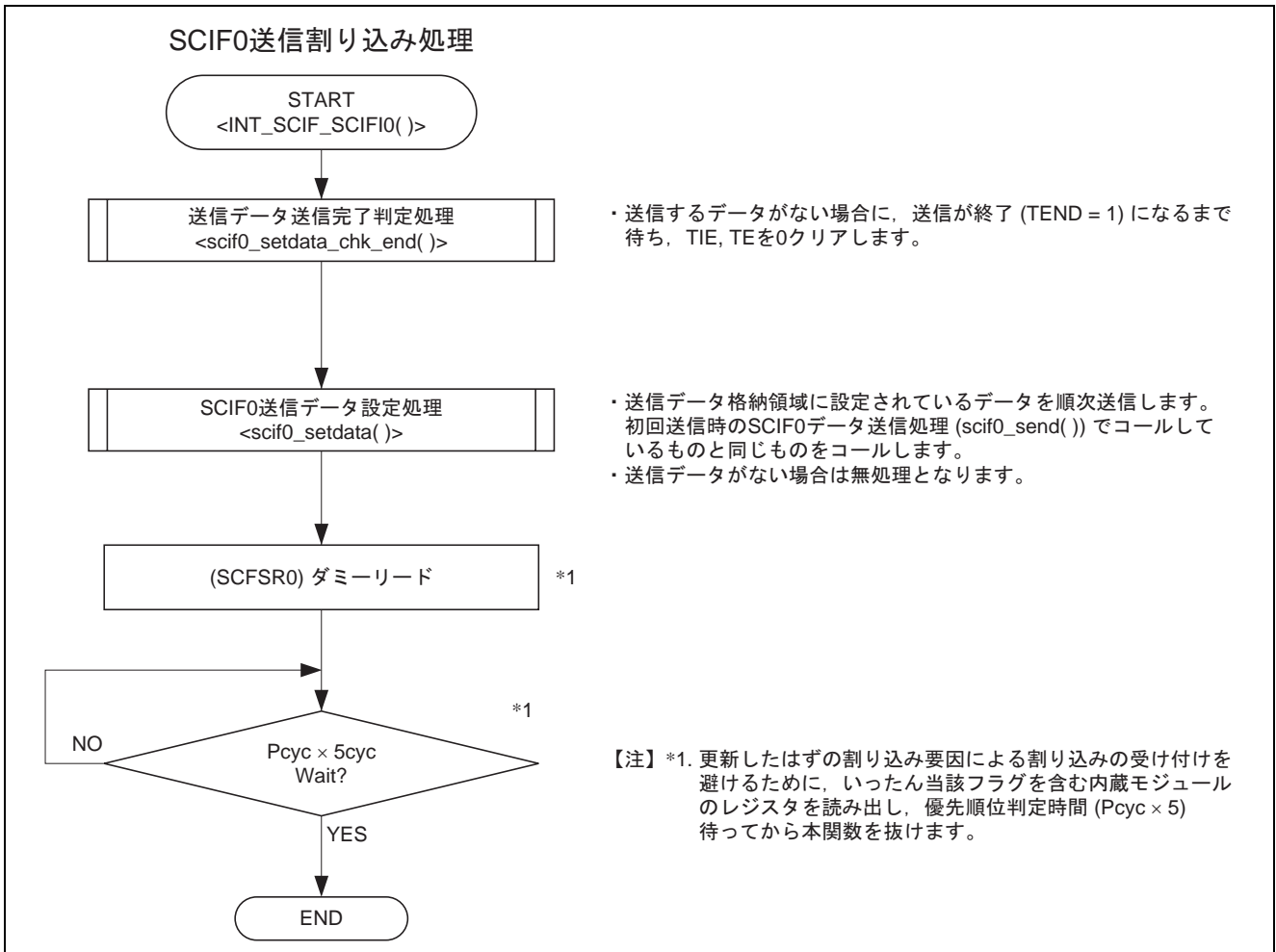


図 11 SCIF0 送信割り込み処理フロー

3.3.4 SCIF1 データ受信処理フロー

受信 FIFO データフル割り込み (RXI) 要求, 受信エラー割り込み (ERI) 要求, ブレーク割り込み (BRI) 要求の発生時に SCIF1 受信割り込み処理が起動されます (図 12)。

SCIF1 受信割り込み処理で SCIF1 データ受信処理をコールします。(図 13)

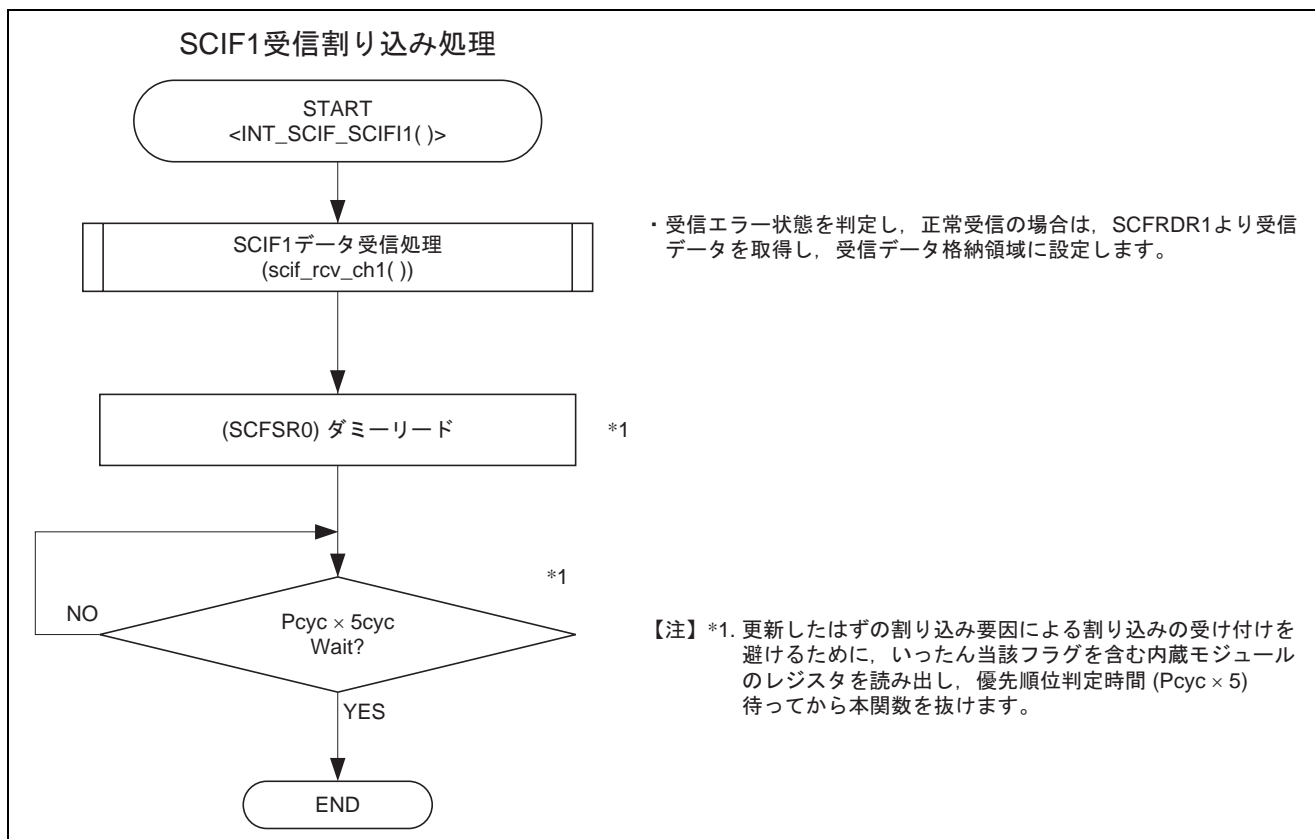


図 12 SCIF1 データ受信割り込み処理フロー

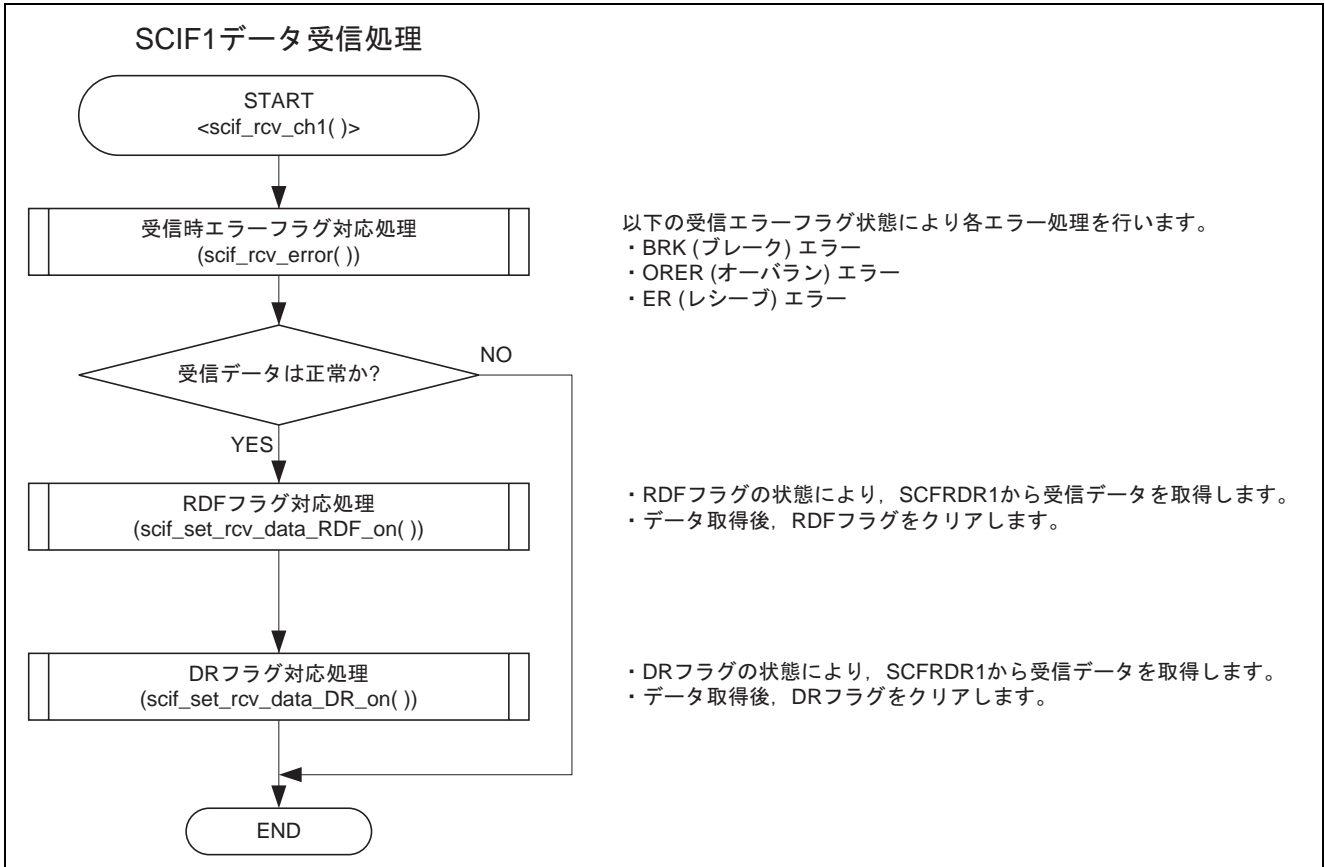


図 13 SCIF1 データ受信処理フロー

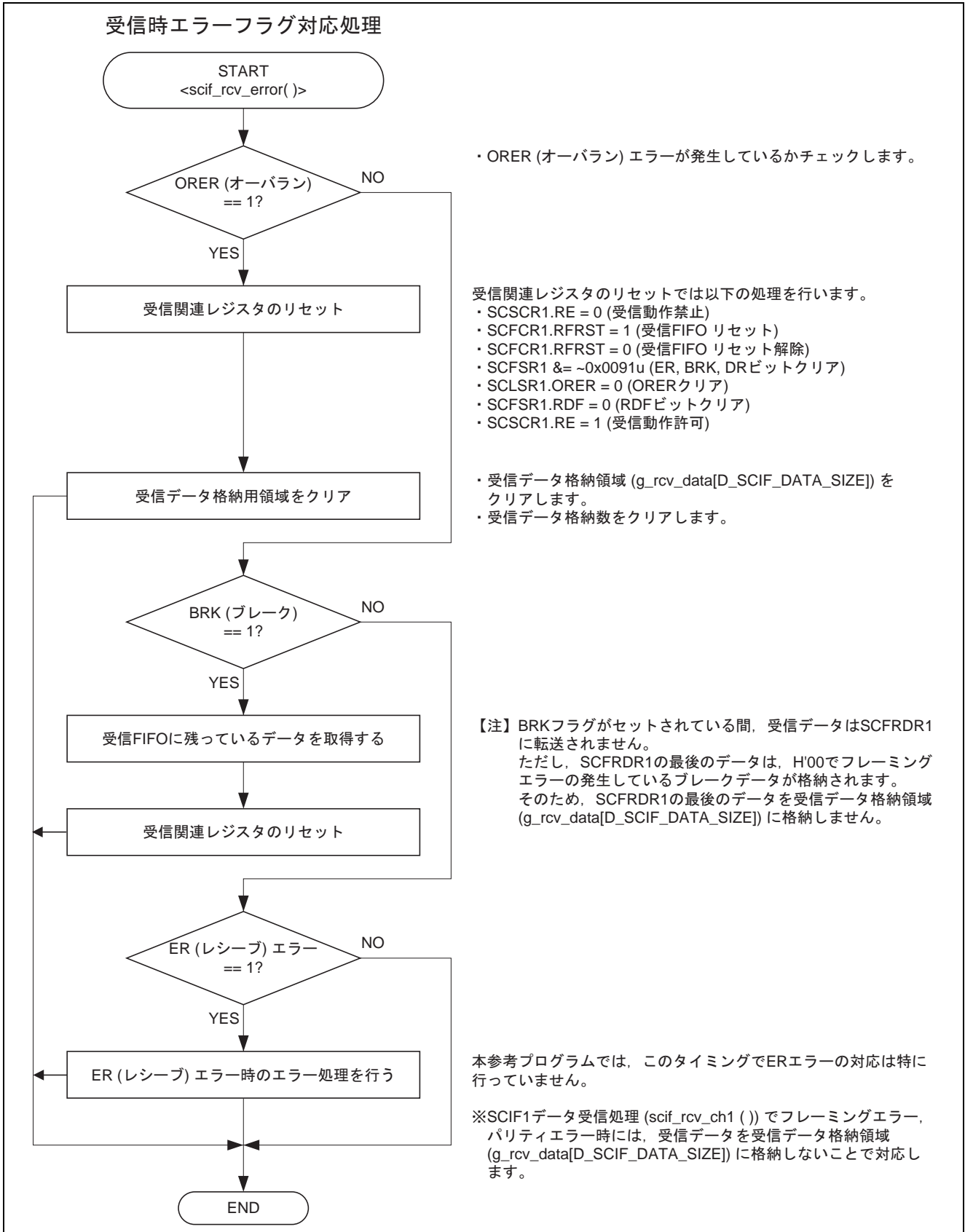


図 14 SCIF1 受信時エラーフラグ対応処理フロー

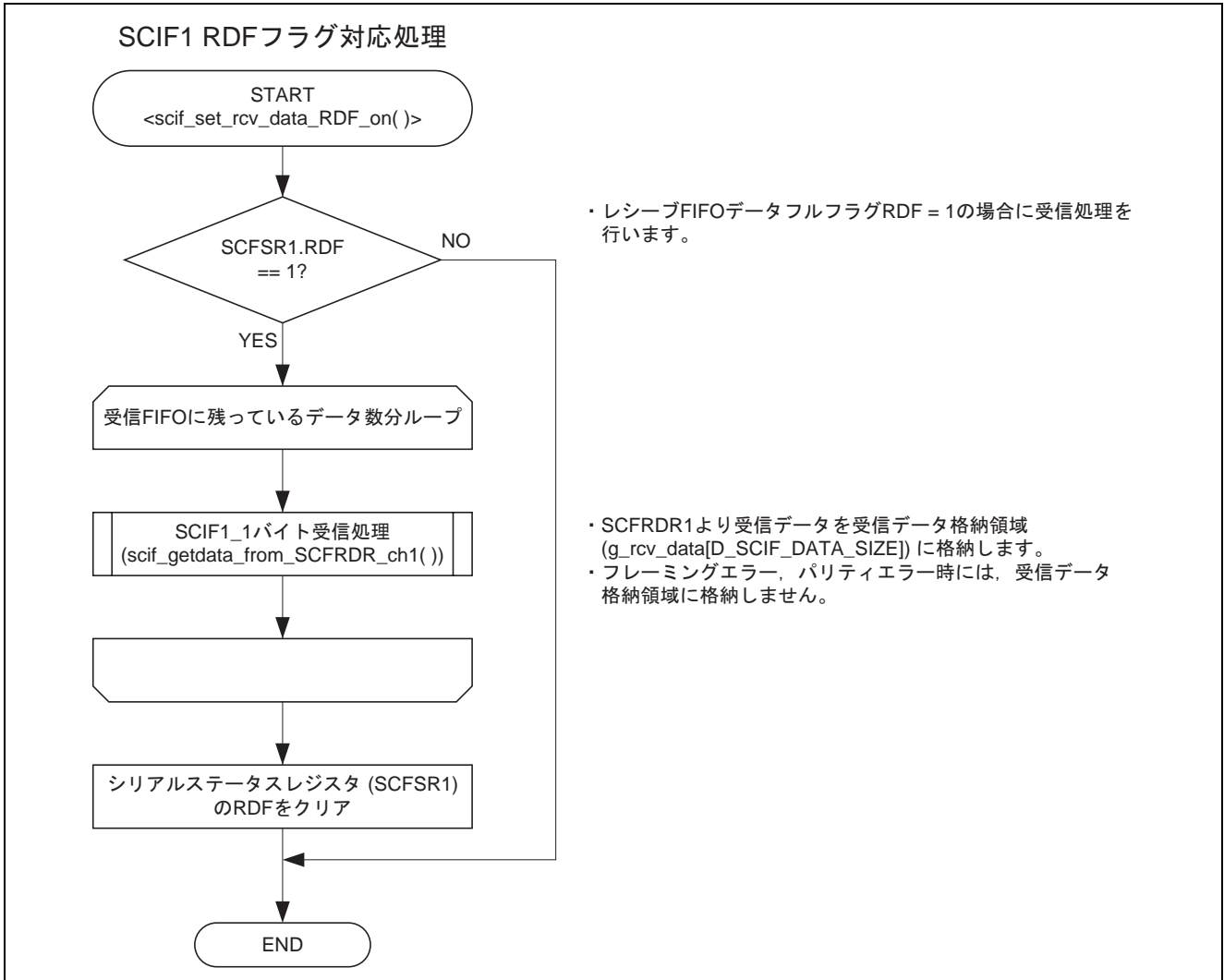


図 15 SCIF1 RDF フラグ対応処理フロー

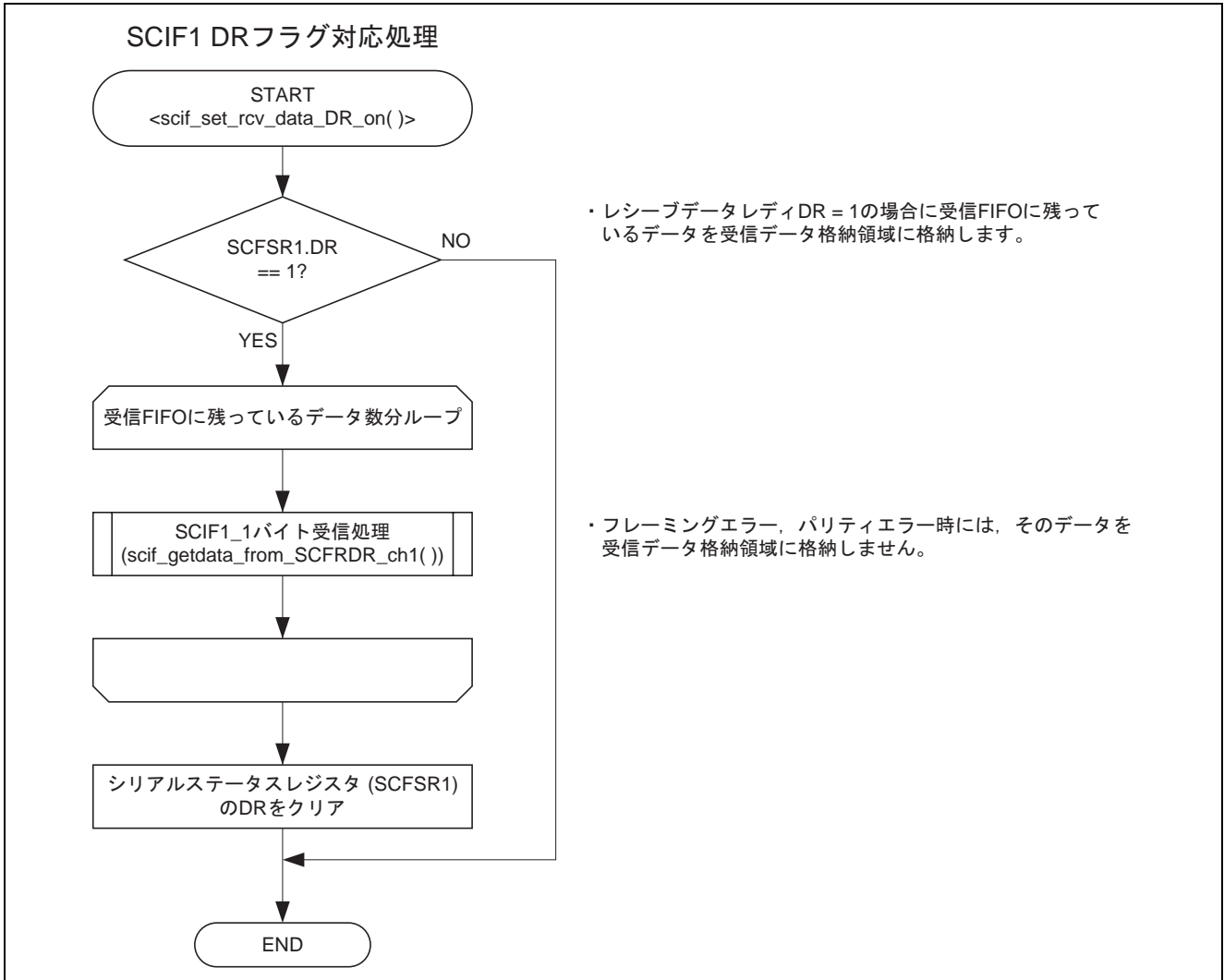


図 16 SCIF1DR フラグ対応処理フロー



### 3.4 参考プログラムのインタフェース

本参考プログラムでは、SCIF でデータ送受信するために、主なインタフェースとして以下を作成します。

#### (1) SCIF0 初期化処理 (scif\_init\_SCIF0())

##### 【概要】

SCIF0 を調歩同期式の送信モードに初期化します。  
 レジスタの設定値に関しては、表 4 を参照ください。  
 処理フローについては、図 4、図 5 を参照ください。

##### 【パラメータ】

なし。

#### (2) SCIF1 初期化処理 (scif\_init\_SCIF1())

##### 【概要】

SCIF1 を調歩同期式の受信モードに初期化します。  
 レジスタの設定値に関しては、表 5 を参照ください。  
 処理フローについては、図 6、図 7 を参照ください。

##### 【パラメータ】

なし。

#### (3) SCIF0 データ送信処理 (scif0\_send())

##### 【概要】

SCIF0 よりデータ送信開始する際に使用します。  
 事前に SCIF0 初期化処理 (scif\_init\_SCIF0()) をコールしておく必要があります。  
 処理フローについては、図 8 を参照ください。

##### 【処理内容】

- 送信データ格納領域をクリア後、送信データ格納領域にパラメータで設定した送信データ数と送信データを設定します。ただし、送信データサイズが、D\_SCIF\_DATA\_SIZE より大きい場合には設定を行いません。より大きいサイズのデータを送信したい場合は D\_SCIF\_DATA\_SIZE の値を変更してください。
- SCIF0 送信データ設定処理 (scif0\_setdata()) をコールします。

##### 【パラメータ】

型	変数	内容
unsigned long	i_count	送信データ数 (バイト単位)
unsigned char*	i_ptr	送信データ

## (4) SCIF0 送信データ設定処理 (scif0\_setdata())

**【概要】**

送信データ格納領域に設定されている送信データを SCFTDR0 に設定する際にコールします。

SCIF0 データ送信処理 (scif0\_send()) (図 8) と送信 FIFO データエンプティ割り込み (TXI) 発生時にコールされる SCIF0 割り込み処理 (図 11) からコールされます。

事前に送信データ格納領域に送信データが設定されている必要があります。送信データ格納領域に送信データがない場合は、無処理となります。

処理フローについては、図 9 を参照ください。

**【処理内容】**

1. 送信 FIFO の空き数分、送信データ格納領域に設定されている送信データを SCFTDR0 に設定します。
2. 送信 FIFO データエンプティ割り込み (TXI) 要求を許可にします。

**【パラメータ】**

なし。

## (5) SCIF1 データ受信処理 (scif\_rcv\_ch1())

**【概要】**

SCIF1 よりデータ受信する際に使用します。

レシーブインタラプトイネーブルの割り込み処理でコールされます。(図 12)

受信データを受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納します。

ただし受信したデータが D\_SCIF\_DATA\_SIZE より大きくなる場合は、それ以降のデータは格納しません。

事前に SCIF1 初期化処理 (scif\_init\_SCIF1()) をコールしておく必要があります。

処理フローについては、図 13 を参照ください。

**【処理内容】**

1. 受信データのエラーチェックを行い、エラーの際にはおのこのエラーケースの処理を行います。(詳細は、受信時エラーフラグ対応処理 (scif\_rcv\_error()) を参照ください。)
2. 1.の処理で正常データと判断された場合、受信 FIFO に設定されている分のデータを SCFRDR1 から取得します。
3. SCFRDR1 に受信トリガ設定数未満のデータがあり、かつ最後に受信したデータのストップビットから一定時間以上、次のデータを受信しない場合、SCFSR.DR = 1 となります。SCFSR.DR = 1 となった場合に、受信 FIFO に残っているデータを受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納しません。
4. フレーミングエラー、パリティエラー時には、受信データを受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納しません。

**【パラメータ】**

なし。

## (6) 受信時エラーフラグ対応処理 (scif\_rcv\_error())

**【概要】**

SCIF1 データ受信処理 (scif\_rcv\_ch1()) からコールされます。(図 13)

データ受信時の各種エラー状態を判定し、エラー時には D\_SCIF\_RET\_NG を返します。SCIF1 データ受信処理 (scif\_rcv\_ch1()) では、本関数が、D\_SCIF\_RET\_NG と返した際には、以降の SCIF1 データ受信処理での受信処理を行いません。

処理フローについては、図 14 を参照ください。

**【処理内容】**

1. ORER (オーバラン) エラー発生時
  - 受信関連のレジスタをリセットします。
  - 受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) をクリアします。
2. BRK (ブ레이크) エラー発生時
  - 受信関連のレジスタをリセットします。
  - SCFRDR1 の最後のデータは、H'00 でフレーミングエラーの発生しているブ레이크データが格納されます。そのため、SCFRDR1 の最後のデータを受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納しません。SCFRDR1 の最後のデータ以外の受信 FIFO に残っているデータを受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納します。
3. ER (レシープ) エラー発生時
  - フレーミングエラー、パリティエラー時には、SCIF1 データ受信処理 (scif\_rcv\_ch1()) で受信データを受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) に格納しないことで対応します。

**【パラメータ】**

なし。

**【戻り値】**

受信時エラーフラグ状態正常 D\_SCIF\_RET\_OK

受信時エラーフラグ状態異常 D\_SCIF\_RET\_NG

### 3.5 参考プログラムのレジスタ設定

表 4 に SCIF0 , 表 5 に SCIF1 の参考プログラムのレジスタ設定を示します。

表 4 参考プログラムのレジスタ設定 (SCIF0)

レジスタ名	アドレス	設定値	機能と設定値
シリアルコントロールレジスタ (SCSCR0)	H'FFE00008	H'0000	<ul style="list-style-type: none"> <li>初期化处理(1)</li> <li>TE = "0": 送信動作を禁止</li> <li>RE = "0": 受信動作を禁止</li> </ul>
		H'0020	<ul style="list-style-type: none"> <li>初期化处理(2)</li> <li>CKE[1:0] = "B'00": 内部クロック/SCK 端子は入力端子</li> <li>TE = "1": 送信動作を許可</li> </ul>
		H'0000	<ul style="list-style-type: none"> <li>送信完了時</li> <li>TE = "0": 送信動作を禁止</li> </ul>
FIFO コントロールレジスタ (SCFCR0)	H'FFE00018	H'0006	<ul style="list-style-type: none"> <li>初期化处理(1)</li> <li>TFRST = "1": トランスミット FIFO データレジスタリセット動作を許可</li> <li>RFRST = "1": レシーブ FIFO データレジスタリセット動作を許可</li> </ul>
		H'0000	<ul style="list-style-type: none"> <li>初期化处理(2)</li> <li>TFRST = "0": トランスミット FIFO データレジスタリセット動作を禁止</li> <li>RFRST = "0": レシーブ FIFO データレジスタリセット動作を禁止</li> <li>TTRG[1:0] = "B'00": 送信 FIFO データ数のトリガ送信トリガ数 = 8 #define により変更可能</li> </ul>
シリアルステータスレジスタ (SCFSR0)	H'FFE00010	H'0060	<ul style="list-style-type: none"> <li>初期化处理</li> <li>ER, BRK, FER, PER, RDF, DR ビットクリア</li> </ul>
ラインステータスレジスタ (SCLSR0)	H'FFE00024	H'0000	<ul style="list-style-type: none"> <li>初期化处理</li> <li>ORER ビットクリア</li> </ul>
シリアルモードレジスタ (SCSMR0)	H'FFE00000	H'0000	<ul style="list-style-type: none"> <li>初期化处理</li> <li>CA = "0": 調歩同期式モード</li> <li>CHR = "0": 8 ビットデータ</li> <li>PE = "0": パリティビットの付加を禁止</li> <li>STOP = "0": 1 ストップビット</li> <li>CKS[1:0] = "0": P<math>\phi</math>クロック</li> <li>ビットレートレジスタの設定値に依存</li> </ul>
ビットレートレジスタ (SCBRR0)	H'FFE00004	H'0008	<ul style="list-style-type: none"> <li>初期化处理</li> <li>115.2kbps 設定</li> </ul>
ピンセレクトレジスタ B (PSELB)	H'A4050102	H'00	<ul style="list-style-type: none"> <li>初期化处理</li> <li>PSB8 = "0": SCIF0_TXD の選択</li> </ul>
割り込み優先レベル設定レジスタ G (IPRG)	H'A4080018	H'1000	<ul style="list-style-type: none"> <li>初期化处理</li> <li>IPR0 = "B'0001": 優先レベル 1 設定</li> </ul>
ポート Q コントロールレジスタ (PQCR)	H'A405001A	H'000A	<ul style="list-style-type: none"> <li>初期化处理</li> <li>PQ2MD[1:0] = "B'00": その他の機能選択</li> </ul>
		H'001A	<ul style="list-style-type: none"> <li>BRK 信号送信時</li> <li>PQ2MD[1:0] = "B'01": ポート出力</li> </ul>

表 5 参考プログラムのレジスタ設定 (SCIF1)

レジスタ名	アドレス	設定値	機能と設定値
シリアルコントロールレジスタ (SCSCR1)	H'FFE10008	H'0000	<ul style="list-style-type: none"> <li>初期化処理(1)</li> <li>TE = "0": 送信動作を禁止</li> <li>RE = "0": 受信動作を禁止</li> </ul>
		H'0050	<ul style="list-style-type: none"> <li>初期化処理(2)</li> <li>CKE[1:0] = "B'00": 内部クロック/SCK 端子は入力端子</li> <li>RIE = "1": レシーブインタラプトイネーブルを許可</li> <li>RE = "1": 受信動作を許可</li> </ul>
FIFO コントロールレジスタ (SCFCR1)	H'FFE10018	H'0006	<ul style="list-style-type: none"> <li>初期化処理(1)</li> <li>RFRST = "1": レシーブFIFOデータレジスタリセット動作を許可</li> <li>TFRST = "1": トランスミットFIFOデータレジスタリセット動作を許可</li> </ul>
		H'0040	<ul style="list-style-type: none"> <li>初期化処理(2)</li> <li>RFRST = "0": レシーブFIFOデータレジスタリセット動作を禁止</li> <li>TFRST = "0": トランスミットFIFOデータレジスタリセット動作を禁止</li> <li>RTRG[1:0] = "B'01": レシーブFIFOデータ数のトリガ</li> <li>受信トリガ数 = 4 #define により変更可能</li> </ul>
ラインステータスレジスタ (SCLSR1)	H'FFE00024	H'0000	<ul style="list-style-type: none"> <li>初期化処理</li> <li>ORER ビットクリア</li> </ul>
シリアルモードレジスタ (SCSMR1)	H'FFE10000	H'0000	<ul style="list-style-type: none"> <li>初期化処理</li> <li>CA = "0": 調歩同期式モード</li> <li>CHR = "0": 8 ビットデータ</li> <li>PE = "0": パリティビットの付加を禁止</li> <li>STOP = "0": 1 ストップビット</li> <li>CKS[1:0] = "0": P<math>\phi</math>クロック</li> <li>ビットレートレジスタの設定値に依存</li> </ul>
ビットレートレジスタ (SCBRR1)	H'FFE10004	H'0008	<ul style="list-style-type: none"> <li>初期化処理</li> <li>115.2kbps 設定</li> </ul>
シリアルステータスレジスタ (SCFSR1)	H'FFE10010	H'0060	<ul style="list-style-type: none"> <li>初期化処理</li> <li>ER, BRK, FER, PER, RDF, DR ビットクリア</li> </ul>
ピンセレクトレジスタ B (PSELB)	H'A4050102	H'00	<ul style="list-style-type: none"> <li>初期化処理</li> <li>PSB9 = "0": SCIF1_RXD の選択</li> </ul>
ポート R コントロールレジスタ (PRCR)	H'A405001C	H'0022	<ul style="list-style-type: none"> <li>初期化処理</li> <li>PR1MD[1:0] = "0": その他の機能選択</li> </ul>
割り込み優先レベル設定レジスタ G (IPRG)	H'A4080018	H'0200	<ul style="list-style-type: none"> <li>初期化処理</li> <li>IPR1 = "B'0002": 優先レベル 2 設定</li> </ul>

### 3.6 参考プログラムのマクロ定義

表 6 に参考プログラムのマクロ定義を示します。

表 6 参考プログラムで使用しているマクロ定義

マクロ定義	設定値	機能
D_SCIF_FIFO_NUM	16	FIFO 設定可能数: 16
D_SCIF_DATA_SIZE	50	送信・受信データ格納領域サイズ: 50 格納領域のサイズを変更できます。
D_SCIF_DATA_NUM_SND_TRIGGER_CH0	8	トランスミット FIFO データ数トリガの設定値 設定値を以下 D_SCIF_SEND_FIFO_TRIGGER 0, 2, 4, 8 から選択可能。 本参考プログラムでは D_SCIF_SEND_FIFO_TRIGGER8 を設定
D_SCIF_DATA_NUM_RCV_TRIGGER_CH1	4	レシーブ FIFO データ数トリガの設定値 設定値を以下 D_SCIF_RCV_FIFO_TRIGGER 1, 4, 8, 14 から選択可能。 本参考プログラムでは D_SCIF_RCV_FIFO_TRIGGER4 を設定
D_SCIF_SEND_FIFO_TRIGGER0	0	トランスミット FIFO データ数トリガ: 0
D_SCIF_SEND_FIFO_TRIGGER2	2	トランスミット FIFO データ数トリガ: 2
D_SCIF_SEND_FIFO_TRIGGER4	4	トランスミット FIFO データ数トリガ: 4
D_SCIF_SEND_FIFO_TRIGGER8	8	トランスミット FIFO データ数トリガ: 8
D_SCIF_RCV_FIFO_TRIGGER1	1	レシーブ FIFO データ数トリガ: 1
D_SCIF_RCV_FIFO_TRIGGER4	4	レシーブ FIFO データ数トリガ: 4
D_SCIF_RCV_FIFO_TRIGGER8	8	レシーブ FIFO データ数トリガ: 8
D_SCIF_RCV_FIFO_TRIGGER14	14	レシーブ FIFO データ数トリガ: 14
D_SCIF_SCBRR_1200	0	ビットレート指定値: 1200bps *
D_SCIF_SCBRR_2400	1	ビットレート指定値: 2400bps *
D_SCIF_SCBRR_4800	2	ビットレート指定値: 4800bps *
D_SCIF_SCBRR_9600	3	ビットレート指定値: 9600bps *
D_SCIF_SCBRR_19200	4	ビットレート指定値: 19200bps *
D_SCIF_SCBRR_31250	5	ビットレート指定値: 31250bps *
D_SCIF_SCBRR_38400	6	ビットレート指定値: 38400bps *
D_SCIF_SCBRR_57600	7	ビットレート指定値: 57600bps *
D_SCIF_SCBRR_115200	8	ビットレート指定値: 115200bps *

【注】 \* ビットレート指定値のビットレートを満たす SCSMR の CKS[1:0], SCBRR 設定値が, あらかじめ T\_SCIF\_CKS\_SCBRR\_SET\_INFO scif\_cks\_scbrr\_tbl[] テーブルに格納されています。

テーブルの設定値は,  $P\phi = 33.33\text{MHz}$  で動作する場合を前提としております。そのため,  $P\phi$  を変更する場合はテーブルの設定値も変更してください。

このテーブルに格納されている SCSMR の CKS[1:0], SCBRR 設定値については, 「SH7730 グループ ハードウェアマニュアル (RJJ09B0339) 22 章 FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF) 22.3.8 ビットレートレジスタ (SCBRR)」を参照ください。

#### 4. 参考プログラム例

- (1) サンプルプログラムリスト "sh7730.c"  
 SCIF 処理のメインプログラムです。

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. and is protected under
8  * all applicable laws, including copyright laws.
9
10 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 * PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 * DISCLAIMED.
15
16 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22 * Renesas reserves the right, without notice, to make changes to this
23 * software and to discontinue the availability of this software.
24 * By using this software, you agree to the additional terms and
25 * conditions found by accessing the following link:
26 * http://www.renesas.com/disclaimer
27 *****/
28 /* Copyright (C) 2008. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : sh7730.c
32 * Abstract : SH7730 SCIF 調歩同期式 送受信設定例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 FIFO 内蔵シリアルコミュニケーション
40 * : インターフェース(SCIF)
41 * : 調歩同期式 送受信サンプルプログラム
42 * :
43 * Operation :
44 * Limitation :
45 * :
46 *****/
47 * History : 27.Oct.2008 Ver. 1.00 First Release
48 /*"FILE COMMENT END"*****/
49
50 #include <machine.h>
    
```

```

51  #include "iodefine.h"
52  #include "scif.h"
53
54  void main(void);
55
56  /*"FUNC COMMENT"*****
57  * ID          :
58  * Outline     : サンプルプログラムメイン
59  *             : (調歩同期式送信・受信処理)
60  * Include     :
61  * Declaration : void main(void)
62  * Description : 調歩同期式送受信設定例を示します。
63  *             :
64  *             : SCIF0 からデータを送信し SCIF1 でデータが
65  *             : 正しく受信できていることを確認します。
66  *             :
67  *             : 1.SCIF0 を送信モードで初期化します。
68  *             : 2.SCIF1 を受信モードで初期化します。
69  *             : 3.1 回目データを SCIF0 から順次送信します。
70  *             : 4.SCIF1 の受信データ格納領域のデータと
71  *             :   1 回目送信データが等しいことを確認します。
72  *             : 5.BRK 信号を SCIF0 から送信します。
73  *             : 6.SCIF0 の再送信準備処理を行います。
74  *             : 7.2 回目データを SCIF0 から順次送信します。
75  *             : 8.SCIF1 の受信データ格納領域のデータと
76  *             :   全送信データ(1 回目+2 回目)が等しい
77  *             :   ことを確認します。
78  *             :   BRK 信号が受信データ格納領域に含まれて
79  *             :   いないことも確認します。
80  *             :
81  * Argument    : none
82  * Return Value : none
83  * Calling Functions :
84  *"FUNC COMMENT END"*****/
85  void main(void)
86  {
87      unsigned char  senddata[] = "1234567890ABCDEFGHIJKLMNQRSTUUVWXYZ"; /* 全送信データ */
88      unsigned long  send_num1 = 17; /* 1 回目の送信データ数 */
89      unsigned long  send_num2 = 19; /* 2 回目の送信データ数 */
90
91      unsigned char  *ptr;
92
93      unsigned char  rcvdata[D_SCIF_DATA_SIZE];
94      unsigned long  rcv_num = 0;
95      int            result = 0;
96
97      memset(rcvdata, 0x00, sizeof(rcvdata));
98
99      /* 受信データ格納領域をクリアする */
100     scif_clear_rcvdata();
101
102     /* ==== SCIF0 を調歩同期式の送信モードに初期化 ==== */
103     scif_init_SCIF0();
104
105     /* ==== SCIF1 を調歩同期式の受信モードに初期化 ==== */
106     scif_init_SCIF1();
107

```



```

108     ptr = senddata;
109
110     /* ==== SCIF0 データ送信処理 ==== */
111     scif0_send(send_num1, ptr);          /* 送信データ数 */
112                                         /* 送信データ */
113
114     /* 1回目データ確認 */
115     while (1)
116     {
117         /* 受信データ格納領域よりデータを取得 */
118         scif_get_rcvdata(&rcv_num, rcvdata);
119
120         /* 送信データ数 == 受信データ数となった時 */
121         if(send_num1 == rcv_num)
122         {
123             /* 送信データと受信データを比較する */
124             result = memcmp(senddata, rcvdata, rcv_num);
125
126             /* 一致した場合ループを抜ける */
127             if(result == 0)
128             {
129                 break;
130             }
131
132         }
133     }
134
135     /* BRK 信号送出処理 */
136     scif0_brk_send();
137
138     /* 2回目送信処理を開始する */
139 #if 0 /* 初期設定から行う場合 */
140
141     /* ==== SCIF0 を調歩同期式の送信モードに初期化 ==== */
142     scif_init_SCIF0();
143
144 #else /* 1回目の送信処理と同じ設定で再送信する場合 */
145
146     /* SCIF0 再送信準備処理 */
147     scif0_resend_prepare();
148
149 #endif
150
151     /* 2回目送信データ位置設定 */
152     ptr = senddata + send_num1;
153
154     /* ==== SCIF0 データ送信処理 ==== */
155     scif0_send(send_num2, ptr);
156
157     /* 全データ確認 */
158     while (1)
159     {
160         /* 受信データ格納領域よりデータを取得 */
161         scif_get_rcvdata(&rcv_num, rcvdata);
162
163         /* 送信データ数 == 受信データとなった時 */
164         if((send_num1 + send_num2) == rcv_num)
    
```

```

165     {
166         /* 送信データと受信データを比較する */
167         result = memcmp(senddata, rcvdata, rcv_num);
168
169         /* 一致した場合ループを抜ける */
170         if(result == 0)
171         {
172             break;
173         }
174
175     }
176 }
177
178 /* ここまで処理が実行されたことにより正しく SCIF1 で、データ受信できたことが確認できます */
179 while (1)
180 {
181
182 }
183
184 }
    
```

(2) サンプルプログラムリスト "scif.c"  
 SCIF 機能が実装されているファイルです。

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. and is protected under
8  * all applicable laws, including copyright laws.
9
10 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 * PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 * DISCLAIMED.
15
16 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22 * Renesas reserves the right, without notice, to make changes to this
23 * software and to discontinue the availability of this software.
24 * By using this software, you agree to the additional terms and
25 * conditions found by accessing the following link:
26 * http://www.renesas.com/disclaimer
27 *****/
28 /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : scif.c
32 * Abstract : SH7730 SCIF 調歩同期式 送受信設定例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 FIFO 内蔵シリアルコミュニケーション
40 * : インタフェース(SCIF)
41 * : 調歩同期式 送受信サンプルプログラム
42 * :
43 * Operation :
44 * Limitation :
45 * :
46 *****/
47 * History : 27.Oct.2008 Ver. 1.00 First Release
48 /*"FILE COMMENT END"*****
49
50 #include <machine.h>
51 #include <stdio.h>
52 #include <stdlib.h>
53 #include "iodefine.h"
    
```

```

54  #include "scif.h"
55
56  /* ==== 受信データ格納用 ==== */
57  unsigned char   g_send_data[D_SCIF_DATA_SIZE]; /* 送信データ格納領域 */
58  unsigned long   g_send_count = 0;             /* 送信データ数 */
59  unsigned char   g_rcv_data[D_SCIF_DATA_SIZE]; /* 受信データ格納領域 */
60  unsigned long   g_rcv_count = 0;             /* 受信データ数 */
61
62  /* ==== レジスタ(CKS, SCBRR)設定値テーブル ==== */
63  /* 注意
64     以下の設定値は I =266.66MHz, P =33.33MHz で動作している場合の設定値
65     となります。I, P を変更する場合は以下の設定値も変更してください
66  */
67  const static T_SCIF_CKS_SCBRR_SET_INFO gc_scif_cks_scbrr_tbl[] =
68  {
69      /* SCBRR, CKS, lbitwait */
70      { 216, 1, 222222}, /* 1200bps */
71      { 108, 1, 111111}, /* 2400bps */
72      { 216, 0, 55556}, /* 4800bps */
73      { 108, 0, 27778}, /* 9600bps */
74      { 53, 0, 13889}, /* 19200bps */
75      { 32, 0, 8533}, /* 31250bps */
76      { 26, 0, 6944}, /* 38400bps */
77      { 17, 0, 4630}, /* 57600bps */
78      { 8, 0, 2315} /*115200bps */
79  };
80
81  /* ==== レジスタ(SCFCR.TTRG) 設定値テーブル ==== */
82  const static T_SCIF_SCFCR_TTRG_SET gc_scif_ttrg_tbl[] =
83  {
84      { D_SCIF_SEND_FIFO_TRIGGER8, 0x00}, /* トランスミット FIFO データ数トリガ = 8 */
85      { D_SCIF_SEND_FIFO_TRIGGER4, 0x01}, /* トランスミット FIFO データ数トリガ = 4 */
86      { D_SCIF_SEND_FIFO_TRIGGER2, 0x02}, /* トランスミット FIFO データ数トリガ = 2 */
87      { D_SCIF_SEND_FIFO_TRIGGER0, 0x03} /* トランスミット FIFO データ数トリガ = 0 */
88  };
89
90  /* ==== レジスタ(SCFCR.RTRG) 設定値テーブル ==== */
91  const static T_SCIF_SCFCR_RTRG_SET scif_rtrg_tbl[] =
92  {
93      { D_SCIF_RCV_FIFO_TRIGGER1, 0x00}, /* レシーブ FIFO データ数トリガ = 1 */
94      { D_SCIF_RCV_FIFO_TRIGGER4, 0x01}, /* レシーブ FIFO データ数トリガ = 4 */
95      { D_SCIF_RCV_FIFO_TRIGGER8, 0x02}, /* レシーブ FIFO データ数トリガ = 8 */
96      { D_SCIF_RCV_FIFO_TRIGGER14, 0x03} /* レシーブ FIFO データ数トリガ = 14 */
97  };
98
99
100 /* ==== プロトタイプ宣言 ==== */
101 static unsigned char scif_get_clock_select(T_SCIF_bit_rate_type i_type_bps);
102 static unsigned char scif_get_SCBRR(T_SCIF_bit_rate_type i_type_bps);
103 static void scif_setdata_to_SCFSTR_ch0(unsigned long i_num, unsigned char *i_pdata);
104 static void scif_set_rcvdata(unsigned char *i_pRcv_data);
105 static void scif_getdata_from_SCFSTR_ch1(void);
106 static void scif_reset_rcvdata_status(void);
107 static void scif_set_rest_data(unsigned char i_rest_data);
108 static void scif_set_rcv_data_DR_on(void);
109 static void scif_set_rcv_data_RDF_on(void);
110 static int scif_rcv_error();
    
```

```

111 static unsigned char scif_get_snd_trigger_num(T_SCIF_send_fifo_trigger i_trigger_num);
112 static unsigned char scif_get_rcv_trigger_num(T_SCIF_rcv_fifo_trigger i_trigger_num);
113 static void scif_wait_lbit(unsigned long i_bit_rate);
114
115 /*"FUNC COMMENT"*****
116 * ID :
117 * Outline : SCIF0 初期化処理
118 * Include :
119 * Declaration : void scif_init_SCIF0(void)
120 * Description : SCIF0 を送信モードで以下のように初期化
121 * : します。
122 * : SCSMR
123 * : CA = 調歩同期式
124 * : CHR = 8 ビットデータ
125 * : PE = パリティ無し
126 * : OE = パリティ無しのため関係なし
127 * : STOP = 1 ストップビット
128 * : CKS = gc_scif_cks_scbrrr_tbl[]より設定
129 * : SCSCR
130 * : TIE = 割り込み要求禁止
131 * : RIE = 割り込み要求禁止
132 * : TE = 送信動作を禁止 許可
133 * : RE = 受信動作を禁止
134 * : REIE = 割り込み要求禁止
135 * : CKE = 内部クロック
136 * : SCFSR
137 * : ER, TEND, TDFE, BRK, RDF, DR ビットクリア
138 * : SCBRR
139 * : gc_scif_cks_scbrrr_tbl[]より設定
140 * : 本参考プログラムでは、ビットレートは、
141 * : 115200bps で指定します。
142 * : SCFCR
143 * : RSTRG = モデム信号無効のため関係なし
144 * : RTRG = 送信モードのため関係なし
145 * : TTRG = トランスミット FIFO データ数トリガ 8
146 * : MCE = モデム信号を無効
147 * : TFRST = リセット動作を許可 禁止
148 * : RFRST = リセット動作を許可 禁止
149 * : LOOP = ループバックテストを禁止
150 * : SCLSR
151 * : ORER = オーバランエラーフラグクリア
152 * :
153 * Limitation : ボーレート設定値は内部クロックを使った
154 * : 周辺モジュール用動作周波数 P =33.33MHz
155 * : の場合です。
156 * : 他のクロックを使用する場合はビットレート
157 * : 設定値を変更してください。
158 * : gc_scif_cks_scbrrr_tbl[]の設定値を
159 * : 修正してください。
160 * :
161 * Argument : none
162 * Return Value : none
163 * Calling Functions :
164 *"FUNC COMMENT END"*****/
165 void scif_init_SCIF0(void)
166 {
167     unsigned long dummy;
    
```

```

168
169     /* ==== SCIF0 の初期設定 ==== */
170     /* ==== モジュールストップレジスタの設定 ==== */
171     LOWP.MSTPCR0 &= ~0x00000080;
172
173     dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
174
175     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
176     SCIF0.SCSCR.WORD = 0x0000; /* SCIF0 送/受信動作停止 */
177
178     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
179     SCIF0.SCFCR.BIT.TFRST = 1; /* 送信 FIFO リセット */
180     SCIF0.SCFCR.BIT.RFRST = 1; /* 受信 FIFO リセット */
181
182     /* ==== シリアルステータスレジスタ(SCFSR)の初期化 ==== */
183     SCIF0.SCFSR.WORD &= ~0xff9f; /* ER,BRK,FER,PER,RDF,DR ビットクリア */
184
185     /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
186     SCIF0.SCLSR.BIT.ORER = 0; /* OREO ビットクリア */
187
188     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
189     SCIF0.SCSCR.BIT.CKE = 0; /* B'00:内部クロック */
190
191     /* ==== シリアルモードレジスタ(SCSMR)の設定 ==== */
192     /* コミュニケーションモード : 調歩同期式 */
193     /* キャラクタレングス : 8 ビットデータ */
194     /* パリティイネーブル : 付加とチェックを禁止 */
195     /* パリティモード : 偶数パリティ */
196     /* ストップビットレングス : 1 ストップビット */
197     SCIF0.SCSMR.WORD = 0x00;
198
199     /* ==== クロックセレクト : ビットレートの設定値により決定 ==== */
200     SCIF0.SCSMR.BIT.CKS = scif_get_clock_select(D_SCIF_SCBRR_ACT);
201
202     /* ==== ビットレートレジスタ(SCBRR)の設定 : ビットレートの設定値により決定 ==== */
203     SCIF0.SCBRR = scif_get_SCBRR(D_SCIF_SCBRR_ACT);
204
205     /* ==== 1 ビット期間経過待ち ==== */
206     scif_wait_lbit(D_SCIF_SCBRR_ACT);
207
208     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
209     /* モデムコントロールイネーブル : 禁止 */
210     /* 送信 FIFO データレジスタリセット : 禁止 */
211     /* ループバックテスト : 禁止 */
212     SCIF0.SCFCR.WORD = 0x0000;
213
214     /* 送信 FIFO データ数トリガ : データ数 8 */
215     /* 送信 FIFO データ数トリガは D_SCIF_DATA_NUM_SND_TRIGGER_CH0 の設定値になります */
216     SCIF0.SCFCR.BIT.TTRG = scif_get_snd_trigger_num(D_SCIF_DATA_NUM_SND_TRIGGER_CH0);
217
218     /* ==== ピンセレクトレジスタ B (PSELB) ==== */
219     PFC.PSELB.BIT.PSB8 = 0; /* SCIF0_TXD の選択 */
220
221     /* ==== ポート Q コントロールレジスタ (PQCR) の設定 ==== */
222     PFC.PQCR.BIT.PQ2MD = 0; /* その他の機能選択 */
223
224     /* ==== 割り込み優先レベル設定 (優先レベル 1) ==== */
    
```

```

225     INTC0.IPRG = INTC0.IPRG | 0x1000;
226
227     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
228     SCIF0.SCSCR.BIT.TE = 1;          /* SCIF0 送信動作を許可する */
229
230 }
231
232 /*"FUNC COMMENT"*****
233 * ID          :
234 * Outline    : SCIF1 初期化处理
235 * Include    :
236 * Declaration : void scif_init_SCIF1(void)
237 * Description : SCIF1 を受信モードで以下のように初期化
238 *             : します。
239 *             :   SCSMR
240 *             :   CA = 調歩同期式
241 *             :   CHR = 8 ビットデータ
242 *             :   PE = パリティ無し
243 *             :   OE = パリティ無しのため関係なし
244 *             :   STOP = 1 ストップビット
245 *             :   CKS = gc_scif_cks_scbrrr_tbl[]より設定
246 *             :   SCSCR
247 *             :   TIE = 割り込み要求禁止
248 *             :   RIE = 割り込み要求許可
249 *             :   TE = 送信動作を禁止
250 *             :   RE = 受信動作を禁止 許可
251 *             :   REIE = 割り込み要求禁止
252 *             :   CKE = 内部クロック
253 *             :   SCFSR
254 *             :   ER, TEND, TDFE, BRK, RDF, DR ビットクリア
255 *             :   SCBRR
256 *             :   gc_scif_cks_scbrrr_tbl[]より設定
257 *             :   本参考プログラムでは、ビットレートは、
258 *             :   115200bps で指定します。
259 *             :   SCFCR
260 *             :   RSTRG = モデム信号無効のため関係なし
261 *             :   RTRG = レシーブ FIFO データ数トリガ 4 を設定
262 *             :   TTRG = 受信モードのため関係なし
263 *             :   MCE = モデム信号を無効
264 *             :   TFRST = リセット動作を許可 禁止
265 *             :   RFRST = リセット動作を許可 禁止
266 *             :   LOOP = ループバックテストを禁止
267 *             :   SCLSR
268 *             :   ORER = オーバランエラーフラグクリア
269 *             :
270 * Limitation : ボーレート設定値は内部クロックを使った
271 *             : 周辺モジュール用動作周波数 P = 33MHz
272 *             : の場合です。
273 *             : 他のクロックを使用する場合はビットレート
274 *             : 設定値を変更してください。
275 *             : gc_scif_cks_scbrrr_tbl[]の設定値を
276 *             : 修正してください。
277 *             :
278 * Argument   : none
279 * Return Value : none
280 * Calling Functions :
281 * "FUNC COMMENT END"*****
    
```

```

282 void scif_init_SCIF1(void)
283 {
284     unsigned long dummy;
285
286     /* ==== SCIF1 の初期設定 ==== */
287     /* ==== モジュールストップレジスタの設定 ==== */
288     LOWP.MSTPCR0 &= ~0x00000040;
289
290     dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
291
292     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
293     SCIF1.SCSCR.WORD = 0x0000; /* SCIF1 送/受信動作停止 */
294
295     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
296     SCIF1.SCFCR.BIT.TFRST = 1; /* 送信 FIFO リセット */
297     SCIF1.SCFCR.BIT.RFRST = 1; /* 受信 FIFO リセット */
298
299     /* ==== シリアルステータスレジスタ(SCFSR)の設定 ==== */
300     SCIF1.SCFSR.WORD &= ~0xff9f; /* ER,BRK,FER,PER,RDF,DR ビットクリア */
301
302     /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
303     SCIF1.SCLSR.BIT.ORER = 0; /* ORER ビットクリア */
304
305     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
306     SCIF1.SCSCR.BIT.CKE = 0; /* B'00:内部クロック */
307
308     /* ==== シリアルモードレジスタ(SCSMR)の設定 ==== */
309     /* コミュニケーションモード : 調歩同期式 */
310     /* キャラクタレングス : 8 ビットデータ */
311     /* パリティイネーブル : 付加とチェックを禁止 */
312     /* パリティモード : 偶数パリティ */
313     /* ストップビットレングス : 1 ストップビット */
314     SCIF1.SCSMR.WORD = 0x00;
315
316     /* ==== クロックセレクト : ビットレートの設定値により決定 ==== */
317     SCIF1.SCSMR.BIT.CKS = scif_get_clock_select(D_SCIF_SCBRR_ACT);
318
319     /* ==== ビットレートレジスタ(SCBRR)の設定 : ビットレートの設定値により決定 ==== */
320     SCIF1.SCBRR = scif_get_SCBRR(D_SCIF_SCBRR_ACT);
321
322     /* ==== 1 ビット期間経過待ち ==== */
323     scif_wait_lbit(D_SCIF_SCBRR_ACT);
324
325     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
326     /* モデムコントロールイネーブル : 禁止 */
327     /* 送信 FIFO データレジスタリセット : 禁止 */
328     /* ループバックテスト : 禁止 */
329     SCIF1.SCFCR.WORD = 0x0000;
330
331     /* 受信 FIFO データ数トリガ : データ数 4 */
332     /* 受信 FIFO データ数トリガは D_SCIF_DATA_NUM_RCV_TRIGGER_CH1 の設定値になります */
333     SCIF1.SCFCR.BIT.RTRG = scif_get_rcv_trigger_num(D_SCIF_DATA_NUM_RCV_TRIGGER_CH1);
334
335     /* ==== ピンセレクトレジスタ B (PSELB) ==== */
336     PFC.PSELB.BIT.PSB9 = 0; /* SCIF1_RXD の選択 */
337
338     /* ==== ポート R コントロールレジスタ (PRCR) の設定 ==== */
    
```



```

339     PFC.PRCR.BIT.PR1MD = 0;           /* その他の機能選択 */
340
341     /* ==== 割り込み優先レベル設定 (優先レベル 2) ==== */
342     INTC0.IPRG = INTC0.IPRG | 0x0200;
343
344     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
345     SCIF1.SCSCR.BIT.RIE = 1;         /* 受信データフル割り込み有効 */
346
347     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
348     SCIF1.SCSCR.BIT.RE = 1;         /* SCIF0 受信動作を許可する */
349
350 }
351
352 /*"FUNC COMMENT"*****
353 * ID :
354 * Outline : クロックセレクト設定値取得
355 * Include :
356 * Declaration : unsigned char scif_get_clock_select
357 * : (T_SCIF_bit_rate_type i_type_bps)
358 * Description : ビットレートに対応するクロックセレクト
359 * : CKS[1:0]設定値を取得する。
360 * :
361 * Argument : T_SCIF_bit_rate_type i_type_bps :
362 * : ビットレート指定値
363 * Return Value : unsigned char : クロックセレクト設定値
364 * Calling Functions :
365 *"FUNC COMMENT END"*****/
366 unsigned char scif_get_clock_select(T_SCIF_bit_rate_type i_type_bps)
367 {
368     return gc_scif_cks_scbrt_tbl[i_type_bps].mcsmr_cks;
369 }
370
371
372 /*"FUNC COMMENT"*****
373 * ID :
374 * Outline : ビットレートレジスタ設定値(SCBRR)取得
375 * Include :
376 * Declaration : unsigned char scif_get_SCBRR
377 * : (T_SCIF_bit_rate_type i_type_bps)
378 * Description : ビットレートに対応するビットレート
379 * : レジスタ設定値を取得します。
380 * :
381 * Argument : T_SCIF_bit_rate_type i_type_bps :
382 * : ビットレート指定値
383 * Return Value : unsigned char : ビットレートレジスタ設定値
384 * Calling Functions :
385 *"FUNC COMMENT END"*****/
386 unsigned char scif_get_SCBRR(T_SCIF_bit_rate_type i_type_bps)
387 {
388     return gc_scif_cks_scbrt_tbl[i_type_bps].mscbrt;
389 }
390
391
392 /*"FUNC COMMENT"*****
393 * ID :
394 * Outline : SCIF0 データ送信処理
395 * Include :
    
```

```

396 * Declaration          : void scif0_send
397 *                      : (unsigned long i_count,
398 *                      :   unsigned char *i_ptr)
399 *                      :
400 * Description          : 送信データを送信データ格納領域に保存し
401 *                      : SCIF0 送信データ設定処理をコールします。
402 *                      :
403 * Argument             : unsigned long i_count :
404 *                      :   送信データ数
405 *                      : unsigned char *i_ptr :
406 *                      :   送信データ
407 * Return Value         : none
408 * Calling Functions    :
409 * "FUNC COMMENT END"*****/
410 void scif0_send(unsigned long i_count, unsigned char *i_ptr)
411 {
412     /* SCIF 送信データ格納領域設定処理 */
413     scif_set_send_data(i_count, i_ptr);
414
415     /* SCIF0 送信データ設定処理 */
416     scif0_setdata();
417 }
418 }
419
420 /*"FUNC COMMENT"*****
421 * ID                    :
422 * Outline                : 送信データ送信完了判定処理
423 * Include                :
424 * Declaration            : void scif0_setdata_chk_end(void)
425 *                      :
426 * Description            : 全送信データを送信したか判定し、送信完了
427 *                      : した場合は SCSCR の TIE, TE = 0 を
428 *                      : クリアします。
429 *                      :
430 * Argument                : none
431 *                      :
432 * Return Value            : none
433 * Calling Functions      :
434 * "FUNC COMMENT END"*****/
435 void scif0_setdata_chk_end(void)
436 {
437     unsigned long     send_num = 0;
438
439     /* SCIF 送信データ数取得処理 */
440     scif_get_send_data_num(&send_num);
441
442     /* 全てのデータを送信完了した場合 */
443     if(send_num == 0)
444     {
445         /* ==== 送信完了判定 ==== */
446         while(SCIF0.SCFISR.BIT.TEND == 0)
447         {
448
449         }
450
451         /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
452         SCIF0.SCSCR.BIT.TIE = 0; /* 送信データエンプティ割り込み無効 */
    
```

```

453
454     /* ==== SCIF0 送信動作を無効にする ==== */
455     SCIF0.SCSCR.BIT.TE = 0;
456
457     }
458
459     }
460
461     /*"FUNC COMMENT"*****
462     * ID           :
463     * Outline      : SCIF0 送信データ設定処理
464     * Include      :
465     * Declaration  : void scif0_setdata(void)
466     *              :
467     * Description  : 送信データ格納領域から送信データを取得し
468     *              : SCFTDR0 データ設定処理をコールします。
469     *              :
470     * Argument     : none
471     *              :
472     * Return Value : none
473     * Calling Functions :
474     *"FUNC COMMENT END"*****/
475     void scif0_setdata(void)
476     {
477         unsigned long    send_num = 0;
478         unsigned char    send_data[D_SCIF_DATA_SIZE];
479
480         memset(send_data, 0x00, sizeof(send_data));
481
482         /* SCIF 送信データ数取得処理 */
483         scif_get_send_data_num(&send_num);
484
485         /* 送信データがなければ無処理 */
486         if(send_num == 0)
487         {
488             return;
489         }
490
491         /* SCIF 送信データ取得処理 */
492         scif_get_send_data(send_num, send_data);
493
494         /* SCFTDR0 データ設定処理 */
495         scif_setdata_to_SCFTDR_ch0(send_num, send_data);
496
497     }
498
499     /*"FUNC COMMENT"*****
500     * ID           :
501     * Outline      : SCFTDR0 データ設定処理
502     * Include      :
503     * Declaration  : void scif_setdata_to_SCFTDR_ch0
504     *              : (unsigned long i_num,
505     *              :   unsigned char *i_pdata)
506     *              :
507     * Description  : TDFE の状態を確認し送信 FIFO データエンプティ
508     *              : であることを確認します。
509     *              : SCFDR.TFDC を確認して設定可能な数の
    
```

```

510 *           : 送信データを SCFTDR0 に設定します。
511 *           : データ設定後 TDFE, TEND をクリアします。
512 *           : SCFTDR0 に送信データを設定後, 送信データ
513 *           : 格納領域を更新します。
514 *           : SCFTDR0 に設定していない送信データが存在
515 *           : する場合に, 送信データエンプティ割り込み
516 *           : を有効にします。
517 *           :
518 * Argument   : unsigned long i_num :
519 *           : 送信データ数
520 *           : unsigned char *i_pdata :
521 *           : 送信データ
522 *           :
523 * Return Value : none
524 * Calling Functions :
525 * "FUNC COMMENT END"*****/
526 void scif_setdata_to_SCFTDR_ch0(unsigned long i_num, unsigned char *i_pdata)
527 {
528     int          i = 0;
529     unsigned long set_send_fifo_num = 0;
530
531     /* ==== シリアルステータスレジスタ(SCSFR0)の
532     送信 FIFO データエンプティを確認 (TDFE フラグ) ==== */
533     if(SCIF0.SCF SR.BIT.TDFE == 0){
534         return;
535     }
536
537     /* 未送信データ数取得 */
538     set_send_fifo_num = SCIF0.SCFDR.BIT.TFDC;
539
540     /* 送信 FIFO に設定可能なデータ数を計算 */
541     set_send_fifo_num = D_SCIF_FIFO_NUM - set_send_fifo_num;
542
543     /* 送信 FIFO 設定数 */
544     if(set_send_fifo_num > i_num)
545     {
546         set_send_fifo_num = i_num;
547     }
548
549     /* ==== 送信 FIFO データレジスタ(SCFTDR0)に送信データを書き込む ==== */
550     for(i = 0; i < set_send_fifo_num; i++)
551     {
552         i_num--;
553         SCIF0.SCFTDR = *i_pdata++;
554     }
555
556
557     /* ==== シリアルステータスレジスタ(SCFSR0)の TDFE, TEND を
558     読出し後にクリアする ==== */
559     SCIF0.SCF SR.BIT.TDFE = 0;
560     SCIF0.SCF SR.BIT.TEND = 0;
561
562     /* SCIF 送信データ格納領域を更新する */
563     scif_set_send_data(i_num, i_pdata);
564
565     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
566     SCIF0.SCSCR.BIT.TIE = 1; /* 送信データエンプティ割り込み有効 */
    
```

```

567
568 }
569
570 /*"FUNC COMMENT"*****
571 * ID :
572 * Outline : SCIF 送信データ格納領域設定処理
573 * Include :
574 * Declaration : void scif_set_send_data
575 * : (unsigned long i_count,
576 * : unsigned char *i_ptr)
577 * Description : 送信データを送信データ格納領域に設定します。
578 * :
579 * Limitation : 送信データ格納領域より大きいデータの場合
580 * : 設定処理は行ないません。
581 * Argument : unsigned long i_count :
582 * : 送信データ数
583 * : unsigned char *i_ptr :
584 * : 送信データ
585 * Return Value : none
586 * Calling Functions :
587 /*"FUNC COMMENT END"*****/
588 void scif_set_send_data(unsigned long i_count, unsigned char *i_ptr)
589 {
590     /* 設定データチェック */
591     if(i_count > sizeof(g_send_data))
592     {
593         return;
594     }
595
596     /* 送信データ領域のクリア */
597     scif_clear_senddata();
598
599     if(i_count > 0)
600     {
601         /* 送信データ数設定 */
602         g_send_count = i_count;
603
604         /* 送信データを設定 */
605         memcpy(g_send_data, i_ptr, g_send_count);
606     }
607
608 }
609
610 /*"FUNC COMMENT"*****
611 * ID :
612 * Outline : SCIF 送信データ数取得処理
613 * Include :
614 * Declaration : void scif_get_send_data_num
615 * : (unsigned long *o_pnum)
616 * Description : 送信データ格納領域より送信データ数を
617 * : 取得します。
618 * :
619 * Argument : unsigned long *o_pnum :
620 * : 送信データ数
621 * Return Value : none
622 * Calling Functions :
623 /*"FUNC COMMENT END"*****/

```

```

624 void scif_get_send_data_num(unsigned long *o_pnum)
625 {
626
627     /* パラメータチェック */
628     if(NULL == o_pnum)
629     {
630         return;
631     }
632
633     /* 送信データ数取得 */
634     *o_pnum = g_send_count;
635
636 }
637
638 /*"FUNC COMMENT"*****
639 * ID          :
640 * Outline     : SCIF 送信データ取得処理
641 * Include     :
642 * Declaration : void scif_get_send_data
643 *             : (unsigned long i_num,
644 *             : unsigned char *o_ptr)
645 * Description : 送信データを送信データ格納領域より
646 *             : 取得します。
647 *             :
648 * Limitation  : *o_ptr は scif_get_send_data_num()
649 *             : で取得した領域以上分を指定してください。
650 *             :
651 * Argument    : unsigned long i_num :
652 *             : 取得送信データ数
653 *             : unsigned char *o_ptr :
654 *             : 送信データ
655 * Return Value : none
656 * Calling Functions :
657 *"FUNC COMMENT END"*****/
658 void scif_get_send_data(unsigned long i_num, unsigned char *o_ptr)
659 {
660     /* パラメータチェック */
661     if((NULL == o_ptr) || (i_num > sizeof(g_send_data)))
662     {
663         return;
664     }
665
666     /* 送信データを取得 */
667     memcpy(o_ptr, g_send_data, i_num);
668
669 }
670
671 /*"FUNC COMMENT"*****
672 * ID          :
673 * Outline     : SCIF1 データ受信処理
674 * Include     :
675 * Declaration : void scif_rcv_ch1(void)
676 * Description : 本関数は、レシーブインタラプトイネーブル
677 *             : の割り込み処理でコールされます。
678 *             : SCIF1 の受信時のエラー状態を判定し、
679 *             : 受信可能であるか決定します。
680 *             : ; 正常データである場合には、
    
```

```

681 *           : データ読み出し時に, RDF, DR ビットを
682 *           : 参照し, 受信処理を行います。
683 *           :
684 * Argument       : none
685 * Return Value   : none
686 * Calling Functions :
687 * "FUNC COMMENT END"*****/
688 void scif_rcv_ch1(void)
689 {
690
691     do
692     {
693         /* 受信時エラーフラグ対応処理 */
694         if(D_SCIF_RET_OK != scif_rcv_error())
695         {
696             break;
697         }
698
699         /* ==== RDF フラグ対応処理 ==== */
700         scif_set_rcv_data_RDF_on();
701
702         /* ==== DR フラグ対応処理 ==== */
703         scif_set_rcv_data_DR_on();
704
705     }while(0);
706
707 }
708
709 /* "FUNC COMMENT"*****
710 * ID           :
711 * Outline      : 受信時エラーフラグ対応処理
712 * Include      :
713 * Declaration  : int scif_rcv_error(void)
714 * Description  : データ受信時に変化するエラーフラグ
715 *              : (BRK, ORER, ER)を参照し, 各々のエラー時
716 *              : の処理を行います。
717 *              :   BRK(ブレーク)エラーの時
718 *              :   受信 FIFOに残っているデータを取得
719 *              :   最後に格納されているデータは受信しない。
720 *              :   ORER(オーバーラン)エラーの時
721 *              :   受信ステータスをリセットする。
722 *              :   ER(レシーブ)エラーの時
723 *              :   コメント参照。
724 *              :
725 * Argument     : none
726 * Return Value : int : エラー発生有無
727 * Calling Functions :
728 * "FUNC COMMENT END"*****/
729 int scif_rcv_error(void)
730 {
731     int     ret = D_SCIF_RET_OK;
732
733     do
734     {
735         /* ==== ORER(オーバーラン)エラーの時 ==== */
736         /* ORER(オーバーラン)エラーの時にやりたい処理を定義してください */
737         if(SCIF1.SCLSR.BIT.ORER == 1)
    
```

```

738     {
739         /* 受信ステータスをリセット */
740         scif_reset_rcvdata_status();
741
742         /* 受信データ格納用領域をクリア */
743         scif_clear_rcvdata();
744
745         ret = D_SCIF_RET_NG;
746
747         break;
748     }
749
750     /* ==== BRK(ブレーク)エラーの時 ==== */
751     /* BRK(ブレーク)エラーの時にやりたい処理を定義してください */
752     if(SCIF1.SCFSR.BIT.BRK == 1)
753     {
754         if(SCIF1.SCFDR.BIT.RFDC > 0)
755         {
756             /* 受信 FIFOに残っているデータを取得 */
757             /* BRK 状態で最後に格納されているデータはフレーミング */
758             /* エラーのため格納しない */
759             scif_set_rest_data(SCIF1.SCFDR.BIT.RFDC - 1);
760         }
761
762         /* 受信ステータスをリセット */
763         scif_reset_rcvdata_status();
764
765         ret = D_SCIF_RET_NG;
766
767         break;
768     }
769
770     /* ==== ER(レシーブ)エラーの時 ==== */
771     /* ER(レシーブ)エラーの時にやりたい処理を定義してください */
772     if(SCIF1.SCFSR.BIT.ER == 1)
773     {
774         /*
775          * 本参考プログラムでは、フレーミングエラー、パリティエラー時には、
776          * SCIF1 データ受信処理 (scif_rcv_ch1 ( )) で、受信データを
777          * 受信データ格納領域 (g_rcv_data[D_SCIF_DATA_SIZE]) に
778          * 格納しないことに対応します。
779          */
780     }
781
782     }while(0);
783
784     return  ret;
785
786 }
787
788 /*"FUNC COMMENT"*****
789 * ID           :
790 * Outline      : RDF フラグ対応処理
791 * Include      :
792 * Declaration  : void scif_set_rcv_data_RDF_on(void)
793 * Description  : 受信データ数がトリガー数以上になった時の
794 *               : 受信処理を行います。
    
```



```

795 *           :
796 * Argument       : none
797 * Return Value   : none
798 * Calling Functions :
799 * "FUNC COMMENT END"*****/
800 void scif_set_rcv_data_RDF_on(void)
801 {
802     int     i = 0;
803
804     /* ==== 受信データ数がトリガー数以上になった時 ==== */
805     if(SCIF1.SCFSR.BIT.RDF == 1)
806     {
807         /* 受信 FIFO に残っているデータを取得する */
808         scif_set_rest_data(SCIF1.SCFDR.BIT.RFDC);
809
810         /* ==== シリアルステータスレジスタ(SCFSR1)の RDF をクリアする ==== */
811         /* RDF=1 を読み出した後, SCFRDR1 内の受信データ数が受信トリガ設定数より */
812         /* 少なくなるまで SCFRDR1 を読み出し, 0 を書き込んだときクリアされる */
813         SCIF1.SCFSR.BIT.RDF = 0; /* RDF ビットクリア */
814     }
815 }
816
817 /* "FUNC COMMENT"*****
818 * ID           :
819 * Outline      : DR フラグ対応処理
820 * Include      :
821 * Declaration  : void scif_set_rcv_data_DR_on(void)
822 * Description  : 受信 FIFO にデータが残っており, 一定時間
823 *              : 以上データがこない場合
824 *              : に受信 FIFO にある残データを取得します。
825 *              :
826 * Argument     : none
827 * Return Value : none
828 * Calling Functions :
829 * "FUNC COMMENT END"*****/
830 void scif_set_rcv_data_DR_on(void)
831 {
832     /* ==== 受信 FIFO にデータが残っており, 一定時間以上データがこない場合 ====*/
833     if(SCIF1.SCFSR.BIT.DR == 1)
834     {
835         /* 残りデータを取得 */
836         scif_set_rest_data(SCIF1.SCFDR.BIT.RFDC);
837
838         /* ==== FIFO 受信データを読み出し後, DR をクリア ==== */
839         SCIF1.SCFSR.BIT.DR = 0; /* FIFO が空の場合 DR ビットはクリアされます */
840
841     }
842 }
843
844 /* "FUNC COMMENT"*****
845 * ID           :
846 * Outline      : 受信 FIFO 残データ取得処理
847 * Include      :
848 * Declaration  : void scif_set_rest_data
849 *              : (unsigned char rest_data)
850 * Description  : 受信 FIFO 残データ数を指定し,
851 *              : 受信 FIFO に残っているデータを取得する。
    
```

```

852 *           :
853 * Argument       : unsigned char i_rest_data :
854 *           : 受信 FIFO 残データ数
855 * Return Value   : none
856 * Calling Functions :
857 * "FUNC COMMENT END"*****/
858 void scif_set_rest_data(unsigned char i_rest_data)
859 {
860     int     i = 0;
861
862     /* 受信 FIFO に残っているデータを取得する */
863     for(i = 0; i < i_rest_data; i++)
864     {
865         scif_getdata_from_SCFRDR_ch1();
866     }
867
868 }
869
870 /* "FUNC COMMENT"*****
871 * ID           :
872 * Outline      : 受信エラー時初期化処理
873 * Include      :
874 * Declaration  : void scif_reset_rcvdata_status(void)
875 * Description  : 受信エラー時に受信関連のビットを初期化し,
876 *           : FIFO のデータもクリアする。
877 *           :
878 * Argument     : none
879 * Return Value : none
880 * Calling Functions :
881 * "FUNC COMMENT END"*****/
882 void scif_reset_rcvdata_status(void)
883 {
884     /* 受信エラー処理(受信データの破棄) */
885     SCIF1.SCSCR.BIT.RE = 0;          /* 受信禁止 */
886     SCIF1.SCFRCR.BIT.RFRST = 1;     /* 受信 FIFO リセット */
887     SCIF1.SCFRCR.BIT.RFRST = 0;     /* 受信 FIFO リセット解除 */
888     SCIF1.SCFSR.WORD &= ~0x0091u; /* ER,BRK,DR ビットクリア */
889     SCIF1.SCLSR.BIT.ORER = 0;       /* OREER クリア */
890     SCIF1.SCFSR.BIT.RDF = 0;        /* RDF ビットクリア */
891     SCIF1.SCSCR.BIT.RE = 1;         /* 受信許可 */
892
893 }
894
895 /* "FUNC COMMENT"*****
896 * ID           :
897 * Outline      : SCIF1_1 バイト受信処理
898 * Include      :
899 * Declaration  : void scif_getdata_from_SCFRDR_ch1(void)
900 * Description  : 次ぎに FIFO から読み出すデータが,
901 *           : フレーミングエラー, パリティエラー時は
902 *           : データを受信データ格納領域に設定しません。
903 *           : 正常なデータについて, 受信データ格納領域に
904 *           : 設定処理を行います。
905 * Argument     : none
906 * Return Value : none
907 * Calling Functions :
908 * "FUNC COMMENT END"*****/
    
```

```

909 void scif_getdata_from_SCFRDR_ch1(void)
910 {
911     unsigned char   rcv_data = 0;
912
913     /* ==== フレーミングエラー , パリティエラー時データを受信しない ==== */
914     if(SCIF1.SCFSR.BIT.FER == 1 || SCIF1.SCFSR.BIT.PER == 1)
915     {
916         /* 次の受信データを取得するため ,
917            現在のエラー受信データを一次的に取得する */
918         rcv_data = SCIF1.SCFRDR;
919
920         SCIF1.SCFSR.BIT.ER = 0;
921
922         return;
923     }
924
925     /* ==== 受信 FIFO データレジスタ (SCFRDR1) から受信データを読み出す ==== */
926     scif_set_rcvdata(&SCIF1.SCFRDR);
927
928 }
929
930 /*"FUNC COMMENT"*****
931 * ID          :
932 * Outline     : 受信データ格納処理
933 * Include     :
934 * Declaration : void set_rcvdata(unsigned char *i_pRcv_data)
935 * Description : 引数で指定した受信データを受信データ
936 *             : 格納領域に設定します。
937 *             :
938 * Argument    : unsigned char *i_pRcv_data : 受信データ
939 * Return Value : none
940 * Calling Functions :
941 *"FUNC COMMENT END"*****/
942 void scif_set_rcvdata(unsigned char *i_pRcv_data)
943 {
944     /* 受信データ格納領域に受信データを設定する */
945     if(g_rcv_count < sizeof(g_rcv_data))
946     {
947         g_rcv_data[g_rcv_count] = *i_pRcv_data;
948         g_rcv_count++;
949     }
950
951 }
952
953 /*"FUNC COMMENT"*****
954 * ID          :
955 * Outline     : 送信データ格納領域クリア処理
956 * Include     :
957 * Declaration : void scif_clear_senddata(void)
958 * Description : 送信データ格納領域をクリアする。
959 *             :
960 * Argument    : none
961 * Return Value : none
962 * Calling Functions :
963 *"FUNC COMMENT END"*****/
964 void scif_clear_senddata(void)
965 {

```

```

966     g_send_count = 0;
967     memset(g_send_data, 0x00, sizeof(g_send_data));
968 }
969
970 /*"FUNC COMMENT"*****
971 * ID          :
972 * Outline     : 受信データ格納領域クリア処理
973 * Include     :
974 * Declaration : void scif_clear_rcvdata(void)
975 * Description : 受信データ格納領域をクリアする。
976 *            :
977 * Argument    : none
978 * Return Value : none
979 * Calling Functions :
980 *"FUNC COMMENT END"*****/
981 void scif_clear_rcvdata(void)
982 {
983     g_rcv_count = 0;
984     memset(g_rcv_data, 0x00, sizeof(g_rcv_data));
985 }
986
987 /*"FUNC COMMENT"*****
988 * ID          :
989 * Outline     : 格納受信データ取得処理
990 * Include     :
991 * Declaration : void scif_get_rcvdata
992 *            : (unsigned long *o_pRcvdata_num,
993 *            : unsigned char *o_pRcvdata)
994 * Description : 格納受信データを取得する。
995 *            :
996 * Argument    : unsigned long *o_pRcvdata_num :
997 *            : 格納受信データ数
998 *            : unsigned char *o_pRcvdata :
999 *            : 格納受信データ
1000 * Return Value : none
1001 * Calling Functions :
1002 *"FUNC COMMENT END"*****/
1003 void scif_get_rcvdata(unsigned long *o_pRcvdata_num, unsigned char *o_pRcvdata)
1004 {
1005     if((o_pRcvdata_num == NULL) || (o_pRcvdata == NULL))
1006     {
1007         return;
1008     }
1009
1010     *o_pRcvdata_num = g_rcv_count;
1011
1012     memcpy(o_pRcvdata, g_rcv_data, g_rcv_count);
1013
1014     return;
1015 }
1016
1017 /*"FUNC COMMENT"*****
1018 * ID          :
1019 * Outline     : 送信データ数トリガ対応設定値取得処理
1020 * Include     :
1021 * Declaration : unsigned char scif_get_snd_trigger_num
1022 *            : (T_SCIF_send_fifo_trigger i_trigger_num)

```

```

1023 * Description      : 引数で指定したデータ数トリガに対応する
1024 *                  : SCFCR.TTRG 設定値を取得する。
1025 *                  :
1026 * Argument        : T_SCIF_send_fifo_trigger i_trigger_num
1027 *                  : データ数トリガ
1028 * Return Value    : unsigned char :
1029 *                  : レジスタ設定値(SCFCR.TTRG に設定)
1030 * Calling Functions :
1031 * "FUNC COMMENT END"*****/
1032 unsigned char scif_get_snd_trigger_num(T_SCIF_send_fifo_trigger i_trigger_num)
1033 {
1034     unsigned char  ret = 0;
1035     int            i = 0;
1036     int            count = 0;
1037
1038     count = sizeof(gc_scif_ttrg_tbl) / sizeof(T_SCIF_SCFCR_TTRG_SET);
1039
1040     for(i = 0; i < count; i++)
1041     {
1042         if(gc_scif_ttrg_tbl[i].mtrigger_num == i_trigger_num)
1043         {
1044             /* ==== SCFCR.TTRG 設定 ==== */
1045             ret = gc_scif_ttrg_tbl[i].mttrg;
1046             break;
1047         }
1048     }
1049
1050     return ret;
1051 }
1052
1053 /* "FUNC COMMENT"*****
1054 * ID                  :
1055 * Outline             : 受信データ数トリガ対応設定値取得処理
1056 * Include             :
1057 * Declaration         : unsigned char scif_get_rcv_trigger_num
1058 *                     : (T_SCIF_rcv_fifo_trigger i_trigger_num)
1059 * Description         : 引数で指定したデータ数トリガに対応する
1060 *                     : SCFCR.RTRG 設定値を取得する。
1061 *                     :
1062 * Argument            : T_SCIF_rcv_fifo_trigger i_trigger_num :
1063 *                     : データ数トリガ
1064 * Return Value        : unsigned char :
1065 *                     : レジスタ設定値(SCFCR.RTRG に設定)
1066 * Calling Functions   :
1067 * "FUNC COMMENT END"*****/
1068 unsigned char scif_get_rcv_trigger_num(T_SCIF_rcv_fifo_trigger i_trigger_num)
1069 {
1070     unsigned char  ret = 0;
1071     int            i = 0;
1072     int            count = 0;
1073
1074     count = sizeof(scif_rtrg_tbl) / sizeof(T_SCIF_SCFCR_RTRG_SET);
1075
1076     for(i = 0; i < count; i++)
1077     {
1078         if(scif_rtrg_tbl[i].mtrigger_num == i_trigger_num)
1079         {
    
```

```

1080         /* ==== SCFCR.RTRG 設定 ==== */
1081         ret = scif_rtrg_tbl[i].mrtrg;
1082         break;
1083     }
1084 }
1085
1086     return ret;
1087
1088 }
1089
1090 /*"FUNC COMMENT"*****
1091 * ID          :
1092 * Outline     : 1ビット期間経過待ち処理
1093 * Include     :
1094 * Declaration : void scif_wait_lbit(
1095 *             : unsigned long i_bit_rate)
1096 *             :
1097 * Description  : 指定したビットレートに対応した
1098 *             : 1ビット期間経過後に処理を抜けます。
1099 *             :
1100 *             : gc_scif_cks_scbrrr_tbl テーブルに設定され
1101 *             : ている cpu 実行回数分 nop 処理を行い、
1102 *             : 1ビット期間の経過を待ちます。
1103 *             :
1104 * Limitation  : I を変更した場合は、
1105 *             : gc_scif_cks_scbrrr_tbl テーブルの設定値
1106 *             : も修正ください。
1107 *             :
1108 * Argument    : long i_bit_rate : 動作ビットレート
1109 *             :
1110 * Return Value : none
1111 * Calling Functions :
1112 *"FUNC COMMENT END"*****/
1113 static void scif_wait_lbit(unsigned long i_bit_rate)
1114 {
1115     unsigned long  cpu_count;
1116     unsigned long  i;
1117
1118     /* 1ビット期間待ち分の cpu 実行回数を取得 */
1119     cpu_count = gc_scif_cks_scbrrr_tbl[i_bit_rate].cpu_count_lbit;
1120
1121     for(i = 0; i < cpu_count; i++)
1122     {
1123         nop();
1124     }
1125
1126     return;
1127 }
1128
1129 /*"FUNC COMMENT"*****
1130 * ID          :
1131 * Outline     : SCIF0 再送信準備処理
1132 * Include     :
1133 * Declaration : void scif0_resend_prepare(void)
1134 *             :
1135 * Description  : 送信処理が終了し、SCSCR.TE = 0 にした状態
1136 *             : から再度前回と同じ設定で、送信処理を開始
    
```

```

1137 *           : したい場合にコールしてください。
1138 *           :
1139 *           : 設定を変更する場合は、初期化処理を行って
1140 *           : ください。
1141 *           :
1142 * Argument   : none
1143 *           :
1144 * Return Value : none
1145 *           :
1146 * Calling Functions :
1147 * "FUNC COMMENT END"*****/
1148 void scif0_resend_prepare(void)
1149 {
1150     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
1151     SCIF0.SCSCR.BIT.TE = 0;      /* SCIF0 送信動作を禁止する */
1152
1153     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
1154     SCIF0.SCFCR.BIT.TFRST = 1;  /* 送信 FIFO リセット許可 */
1155
1156     /* ==== シリアルステータスレジスタ(SCFSR)の初期化 ==== */
1157     SCIF0.SCFSR.WORD &= ~0xff9f; /* ER,BRK,FER,PER,RDF,DR ビットクリア */
1158
1159     /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
1160     SCIF0.SCLSR.BIT.ORER = 0;   /* ORER ビットクリア */
1161
1162     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
1163     SCIF0.SCFCR.BIT.TFRST = 0;  /* 送信 FIFO リセット禁止 */
1164
1165     /* ==== ピンセレクトレジスタ B (PSELB) ==== */
1166     PFC.PSELB.BIT.PSB8 = 0;     /* SCIF0_TXD の選択 */
1167
1168     /* ==== ポート Q コントロールレジスタ (PQCR) の設定 ==== */
1169     PFC.PQCR.BIT.PQ2MD = 0;     /* その他の機能選択 */
1170
1171     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
1172     SCIF0.SCSCR.BIT.TE = 1;     /* SCIF0 送信動作を許可する */
1173
1174 }
1175
1176 /* "FUNC COMMENT"*****
1177 * ID           :
1178 * Outline      : BRK 信号送出処理
1179 * Include      :
1180 * Declaration  : void scif0_brk_send(void)
1181 *           :
1182 * Description  : ポート Q コントロールレジスタをポート出力
1183 *           : に変更して一定期間 Low を出力します。
1184 *           :
1185 * Argument     : none
1186 *           :
1187 * Return Value : none
1188 *           :
1189 * Calling Functions :
1190 * "FUNC COMMENT END"*****/
1191 void scif0_brk_send(void)
1192 {
1193     volatile unsigned int cnt;
    
```

```

1194
1195     /* ==== ポート Q コントロールレジスタ (PQCR) の設定 ==== */
1196     PFC.PQCR.BIT.PQ2MD = 1;          /* ポート出力 */
1197
1198     /* Low 出力 */
1199     PORT.PQDR.BIT.B2 = 0;
1200
1201     for ( cnt = 0x80000u; cnt > 0x0u; cnt-- ){
1202         /* ソフトウェアウェイト */
1203     }
1204
1205     return;
1206 }
1207
1208 /* End of File */
    
```



(3) サンプルプログラムリスト "scif.h"  
 SCIF 機能に関連するヘッダファイルです。

```

1  /*****
2  * DISCLAIMER
3
4  * This software is supplied by Renesas Technology Corp. and is only
5  * intended for use with Renesas products. No other uses are authorized.
6
7  * This software is owned by Renesas Technology Corp. and is protected under
8  * all applicable laws, including copyright laws.
9
10 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 * PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 * DISCLAIMED.
15
16 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22 * Renesas reserves the right, without notice, to make changes to this
23 * software and to discontinue the availability of this software.
24 * By using this software, you agree to the additional terms and
25 * conditions found by accessing the following link:
26 * http://www.renesas.com/disclaimer
27 *****/
28 /* Copyright (C) 2008. Renesas Technology Corp., All Rights Reserved. */
29 /*"FILE COMMENT"***** Technical reference data *****/
30 * System Name : SH7730 Sample Program
31 * File Name : scif.h
32 * Abstract : SH7730 SCIF 調歩同期式 送受信設定例
33 * Version : Ver 1.00
34 * Device : SH7730
35 * Tool-Chain : High-performance Embedded Workshop (Version 4.04.01.001)
36 * : C/C++ Compiler Package for SuperH Family (V.9.02release00)
37 * OS : None
38 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
39 * Description : SH7730 FIFO 内蔵シリアルコミュニケーション
40 * : インタフェース(SCIF)
41 * : 調歩同期式 送受信サンプルプログラム
42 * :
43 * Operation :
44 * Limitation :
45 * :
46 *****/
47 * History : 27.Oct.2008 Ver. 1.00 First Release
48 /*"FILE COMMENT END"*****/
49
50 #ifndef __SCIF_DEF_H__
51 #define __SCIF_DEF_H__
52
53 /* ==== マクロ定義 ==== */
    
```

```

54
55  /* 戻り値 */
56  #define D_SCIF_RET_NG      -1
57  #define D_SCIF_RET_OK      0
58
59  /* FIFO 設定可能数 */
60  #define D_SCIF_FIFO_NUM    16
61
62  /* 送信・受信データ格納領域サイズ */
63  #define D_SCIF_DATA_SIZE   50
64
65  /* 動作ビットレートの指定 */
66  /* ビットレートを変更する場合は T_SCIF_bit_rate_type から指定してください。 */
67  #define D_SCIF_SCBRR_ACT   D_SCIF_SCBRR_115200
68
69  /* ==== FIFO データ数トリガ ==== */
70  /* 動作させたい FIFO のトリガ数を以下の定義している */
71  /* デファインから設定してください */
72  #define D_SCIF_DATA_NUM_SND_TRIGGER_CH0    D_SCIF_SEND_FIFO_TRIGGER8 /* 設定可能値 0,2,4,8 */
73  #define D_SCIF_DATA_NUM_RCV_TRIGGER_CH1    D_SCIF_RCV_FIFO_TRIGGER4 /* 設定可能値
74  1,4,8,14 */
75
76  /* 送信 FIFO データ数トリガ */
77  typedef enum
78  {
79      D_SCIF_SEND_FIFO_TRIGGER0 = 0, /* 送信 FIFO データトリガ数 0 */
80      D_SCIF_SEND_FIFO_TRIGGER2 = 2, /* 送信 FIFO データトリガ数 2 */
81      D_SCIF_SEND_FIFO_TRIGGER4 = 4, /* 送信 FIFO データトリガ数 4 */
82      D_SCIF_SEND_FIFO_TRIGGER8 = 8, /* 送信 FIFO データトリガ数 8 */
83  } T_SCIF_send_fifo_trigger;
84
85  /* 受信 FIFO データ数トリガ */
86  typedef enum
87  {
88      D_SCIF_RCV_FIFO_TRIGGER1 = 1, /* 受信 FIFO データトリガ数 1 */
89      D_SCIF_RCV_FIFO_TRIGGER4 = 4, /* 受信 FIFO データトリガ数 4 */
90      D_SCIF_RCV_FIFO_TRIGGER8 = 8, /* 受信 FIFO データトリガ数 8 */
91      D_SCIF_RCV_FIFO_TRIGGER14 = 14 /* 受信 FIFO データトリガ数 14 */
92  } T_SCIF_rcv_fifo_trigger;
93
94  /* ==== ビットレート設定タイプ ==== */
95  typedef enum
96  {
97      D_SCIF_SCBRR_1200,
98      D_SCIF_SCBRR_2400,
99      D_SCIF_SCBRR_4800,
100     D_SCIF_SCBRR_9600,
101     D_SCIF_SCBRR_19200,
102     D_SCIF_SCBRR_31250,
103     D_SCIF_SCBRR_38400,
104     D_SCIF_SCBRR_57600,
105     D_SCIF_SCBRR_115200
106  } T_SCIF_bit_rate_type;
107
108  /* ==== 構造体定義 ==== */
109  /* SCBRR 設定情報 */
110  typedef struct {
    
```

```

111     unsigned char   mscbrr;
112     unsigned char   mscsmr_cks;
113     unsigned long   cpu_count_lbit;
114 } T_SCIF_CKS_SCBRR_SET_INFO;
115
116 /* SCFCR.TTRG 設定用 */
117 typedef struct {
118     unsigned char   mtrigger_num;
119     unsigned char   mttrg;
120 } T_SCIF_SCFCR_TTRG_SET;
121
122 /* SCFCR.RTRG 設定用 */
123 typedef struct {
124     unsigned char   mtrigger_num;
125     unsigned char   mrtrg;
126 } T_SCIF_SCFCR_RTRG_SET;
127
128 /* ==== 関数宣言 ==== */
129 void scif_init_SCIF0(void);
130 void scif_init_SCIF1(void);
131 void scif0_send(unsigned long i_count, unsigned char *i_ptr);
132 void scif0_setdata(void);
133 void scif_set_send_data(unsigned long i_count, unsigned char *i_ptr);
134 void scif_get_send_data_num(unsigned long *o_pnum);
135 void scif_get_send_data(unsigned long i_num, unsigned char *o_ptr);
136 void scif_rcv_ch1(void);
137 void scif0_setdata_chk_end(void);
138 void scif_clear_senddata(void);
139 void scif_clear_rcvdata(void);
140 void scif_get_rcvdata(unsigned long *o_pRcvdata_num, unsigned char *o_pRcvdata);
141 void scif0_brk_send(void);
142 void scif0_resend_prepare(void);
143
144 #endif /* __SCIF_DEF_H__ */
    
```

(4) サンプルプログラムリスト "intprg.c"  
 SCIF1 に関連する割り込み処理を実装しています。

```

1  /*
2  以下の INTC_RESPONSEWAIT は、周辺モジュールの優先順位判定時間待ち用の値です。
3  int_responstime_wait() にパラメータ指定して使用ください。
4  int_responstime_wait() については、common.src を参照ください。
5
6  周辺モジュールの優先順位判定時間は、5Pcyc となります。
7  int_responstime_wait() では、Icyc を 3cyc 実行する処理を、指定されたパラメータの回数
8  実行することで wait 処理を行います。
9
10 int_responstime_wait() のパラメータ指定値計算
11 H'0E >= (1/Pcyc*5cyc)/(1/Icyc*3cyc)
12 念のため 1 回多く実行します。
13
14 注)
15 I =266.66MHz、P =33.33MHz 動作時の設定となっています。
16 これらを変更する場合はこちらの設定値も見直してください。
17 */
18
19 #define INTC_RESPONSEWAIT (0x0000000E)
20
21 ...途中省略...
22
23 /* H'C00 SCIF SCIFIO */
24 void INT_SCIF_SCIFIO(void)
25 {
26     unsigned short dummy;
27
28     /* 送信データ送信完了判定処理 */
29     scif0_setdata_chk_end();
30
31     /* SCIF0 送信データ設定処理 */
32     scif0_setdata();
33
34     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
35     dummy = SCIF0.SCF0SR.WORD;
36     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間分待ち */
37
38 }
39
40 /* H'C20 SCIF SCIF11 */
41 void INT_SCIF_SCIF11(void)
42 {
43
44     unsigned short dummy;
45
46     /* SCIF1 受信処理 */
47     scif_rcv_ch1();
48
49     /* 更新したはずの割り込み要因による割り込みの受け付けを避ける対応 */
50     dummy = SCIF1.SCF1SR.WORD;
51     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間分待ち */
52
53 }
54
55 ...途中省略...

```

## (5) サンプルプログラムリスト "vecttbl.src"

SCIF0, SCIF1 に関連する割り込み実行時の割り込み優先度を設定しています。

SCIF0 に関連する割り込みの優先度を 1 に設定しているため、SCIF0 に関連する割り込み中に新たな SCIF0 に関連する割り込みが発生しないように優先度に 1 を設定しています。

SCIF1 に関連する割り込みの優先度を 2 に設定しているため、SCIF1 に関連する割り込み中に新たな SCIF1 に関連する割り込みが発生しないように優先度に 2 を設定しています。

```

1  ...途中省略...
2
3  ;SCIF
4          ;H'C00      SCIF SCIF0
5          .data.b      H'10
6          ;H'C20      SCIF SCIF1
7          .data.b      H'20
8
9  ...途中省略...
```

5. 実行結果

上記サンプルプログラムの送信受信の実行結果については、送信データと受信データを比較処理するループを抜けているため、SCIF0 から送信したデータが SCIF1 で受信できていることが確認できます。

また、図 17 のように、SCIF1 受信データ格納領域 (g\_rcv\_data [D\_SCIF\_DATA\_SIZE]) を High-performance Embedded Workshop で出力しても SCIF0 から送信したデータが SCIF1 で受信できていることが確認できます。

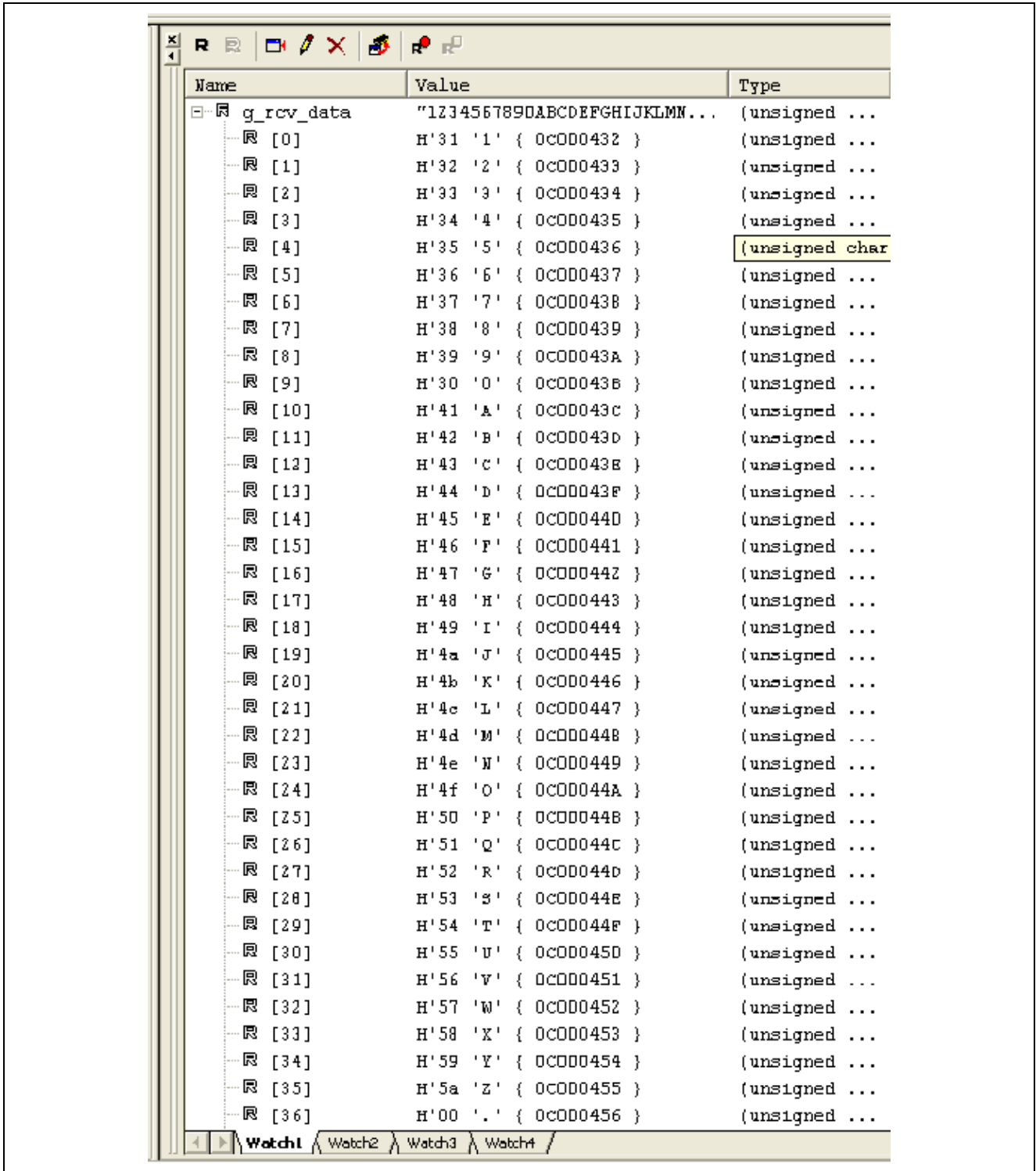


図 17 SCIF1 受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE])

## 6. 参考ドキュメント

- ソフトウェアマニュアル  
SH-4A ソフトウェアマニュアル (RJJ09B0090)  
(最新版をルネサス テクノロジホームページから入手してください。)
- ハードウェアマニュアル  
SH7730 グループ ハードウェアマニュアル (RJJ09B0339)  
(最新版をルネサス テクノロジ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 テクノロジホームページ

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

お問合せ先

<http://japan.renesas.com/inquiry>[csc@renesas.com](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.05.28	—	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。



## 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事事務の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認いただきますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、社会的損害など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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
  - 1 1. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
  - 1 2. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
  - 1 3. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444