

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、誤りが無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 クロック同期式シリアル通信 送受信設定例 (単方向通信)

---

#### 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) 機能のクロック同期式モードの単方向通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.....	24
5. 実行結果.....	50
6. 参考ドキュメント.....	51

## 1. はじめに

### 1.1 仕様

- SCIF チャンネル 0 (SCIF0) を、クロック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、クロック同期式モードの受信モードとして初期化します。
- SCIF0 から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。
- 同期クロックについては、SCIF0 からクロック出力し、SCIF1 で SCIF0 からのクロックを入力します。

### 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、クロック同期式モードで 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 のクロック同期式モードでは、クロックパルスに同期してデータ送受信を行います。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。内部クロックを選択した場合、SCIF はポーレートジェネレータのクロックで動作し、SCK 端子から同期クロックを外部へ出力します。外部クロックを選択した場合は、SCK 端子から入力された外部同期クロックで動作します。通信データフォーマットは、8 ビット固定長です。表 1 にクロック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 (クロック同期式モード) の概要

項目	概要
チャンネル数	4 チャンネル (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
データフォーマット	転送データ長: 8 ビット固定 転送順序: LSB ファースト固定
エラー検出	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) 受信 FIFO データフル割り込み (RXI) ブレイク割り込み (オーバランエラー) (BRI)
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>消費電力低減のために、未使用チャンネルのクロック供給を停止させることが可能</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 クロック同期式モードの動作

SCIF のクロック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) 22.4.3 クロック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 をクロック同期式の送信モード，SCIF1 をクロック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。SCIF0 からデータ送信を行います。
- (3) 送信 FIFO (SCFTDR0) の空き数分，送信データを送信 FIFO (SCFTDR0) に設定します。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) 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 ("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，および SCIF0\_SCK 端子と SCIF1\_SCK 端子を接続します。

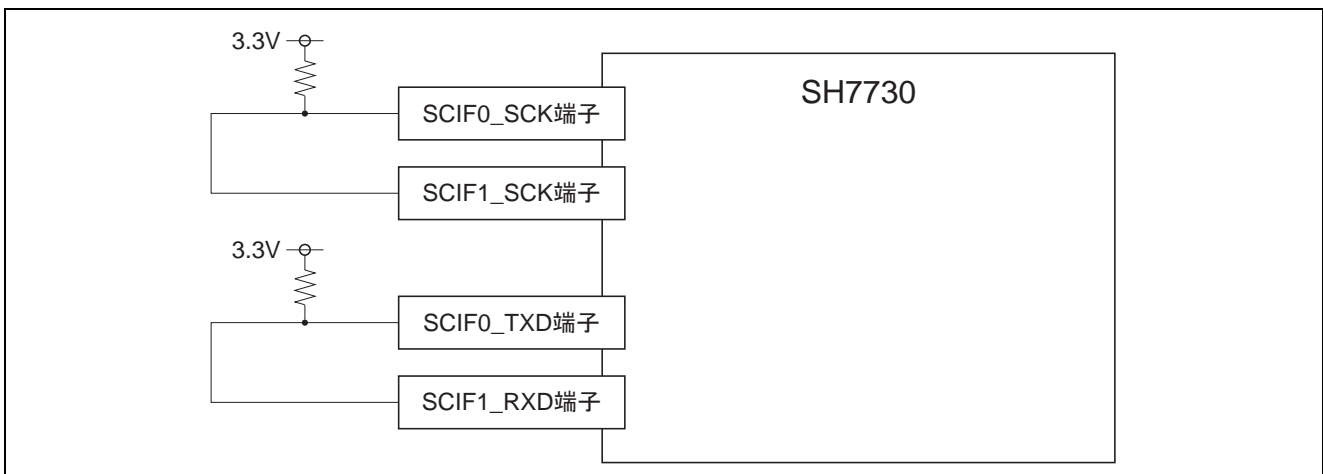


図 1 参考プログラム動作環境

### 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	クロック同期式
使用チャンネル	チャンネル 0
割り込み	送信 FIFO データエンプティ割り込み (TXI) 使用 SCSCR0.TIE = 1 設定 優先度 1
通信速度	100kbps
データ長	8 ビットデータ
ビット順序	LSB ファースト
同期クロック	同期クロック出力
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	クロック同期式
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	100kbps
データ長	8 ビットデータ
ビット順序	LSB ファースト
同期クロック	外部クロック入力
レシーブ FIFO データ数トリガ	2 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。

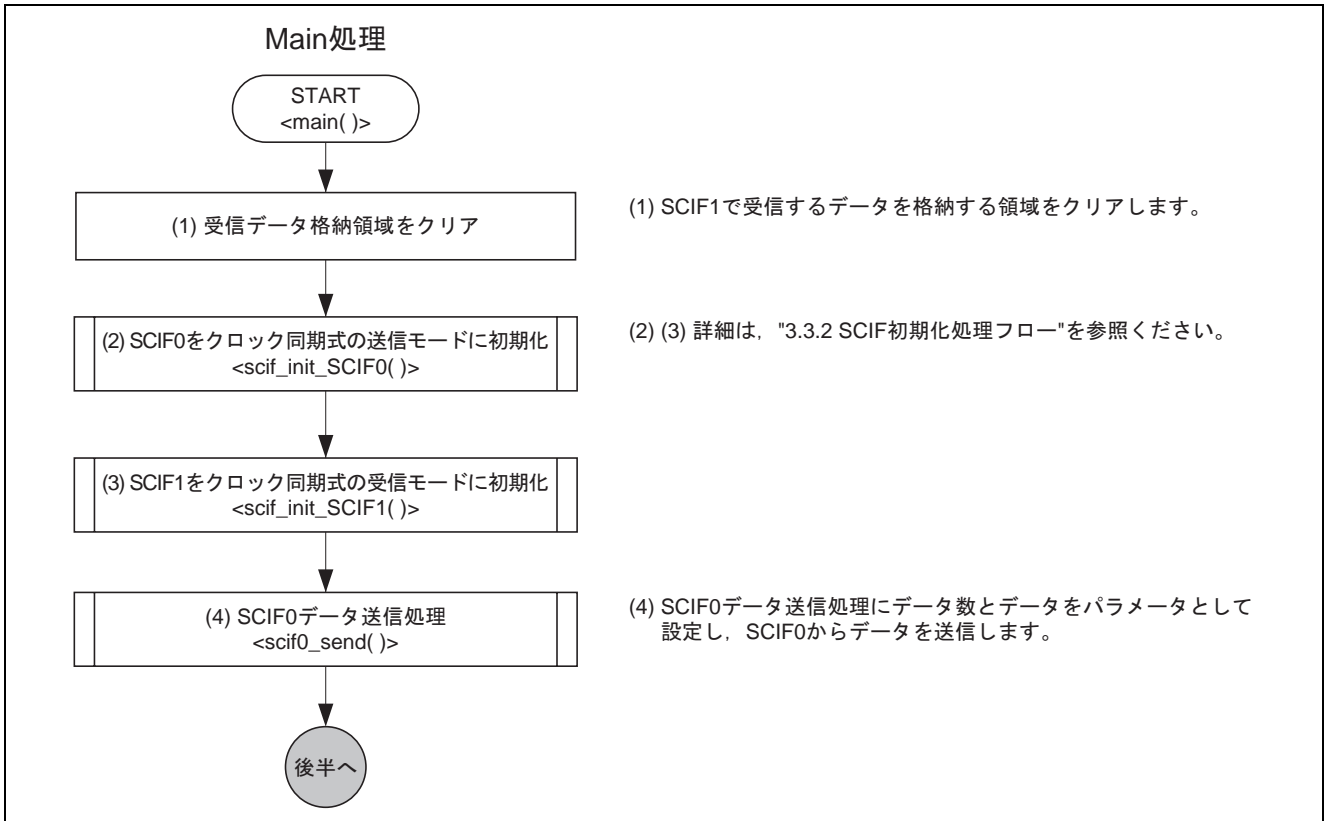


図 2 参考プログラムの Main 処理フロー (前半)



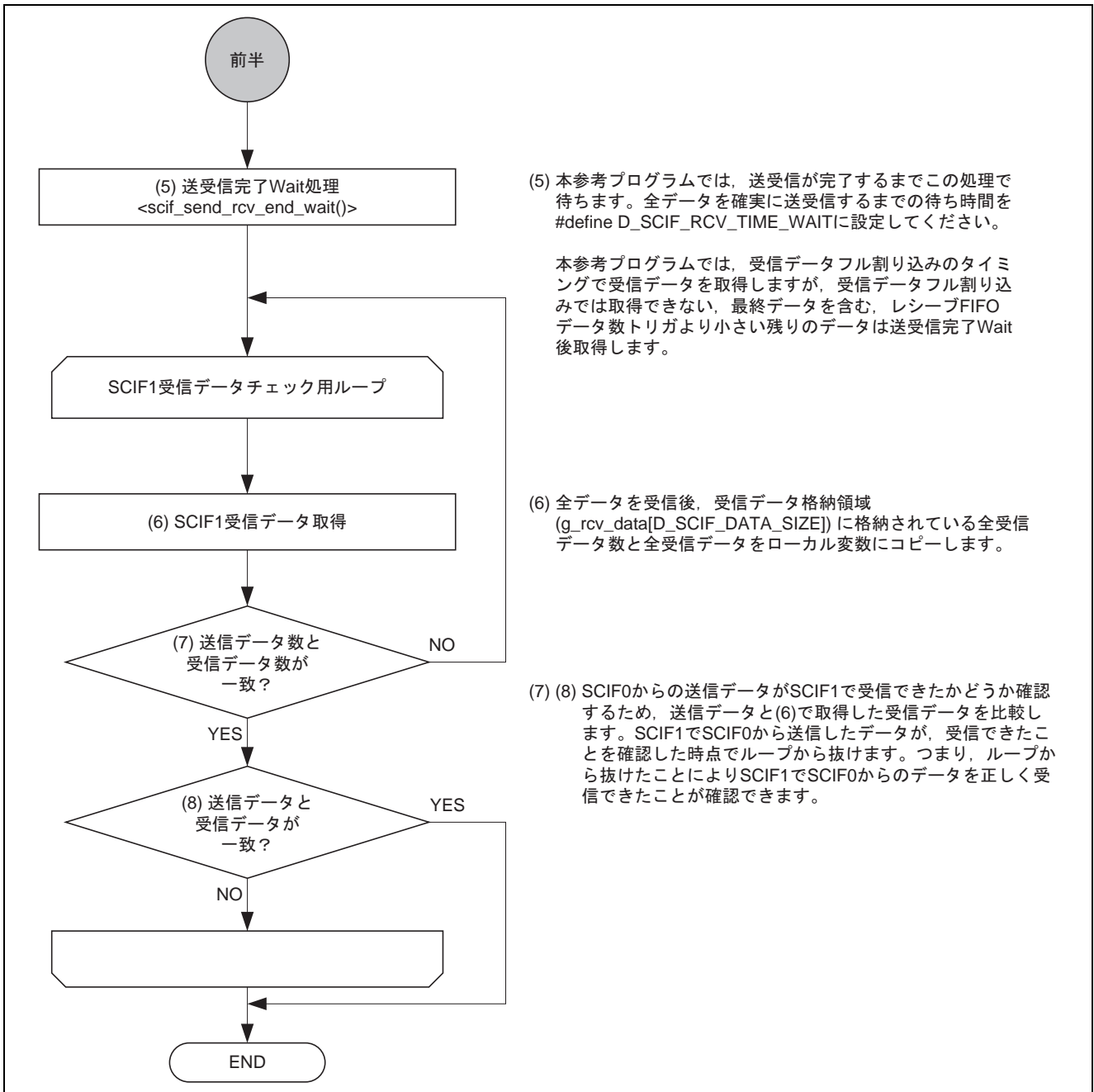


図3 参考プログラムの Main 処理フロー (後半)

3.3.2 SCIF 初期化処理フロー

クロック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。

クロック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。

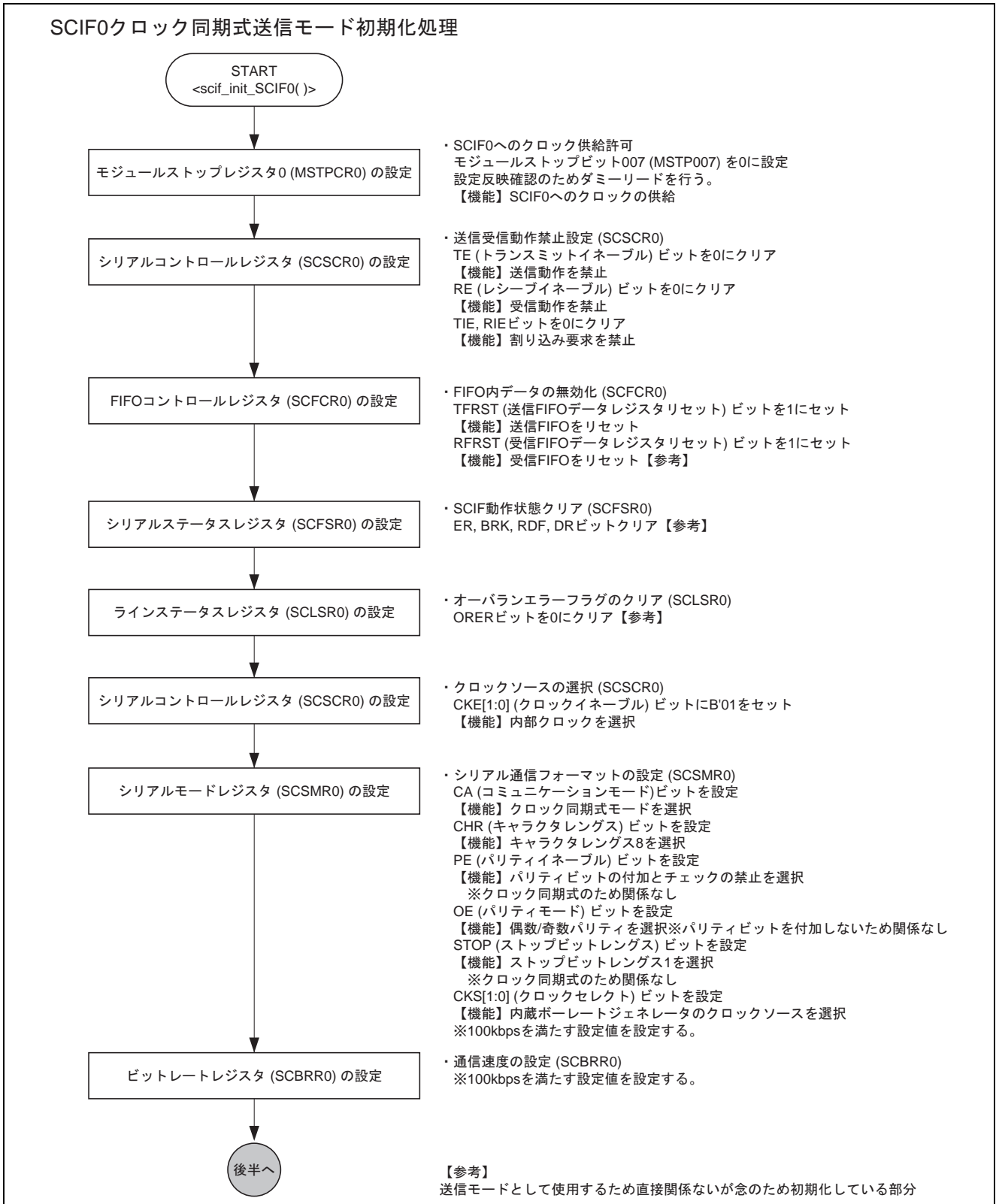


図 4 クロック同期式モード送信初期設定フロー例 (前半)

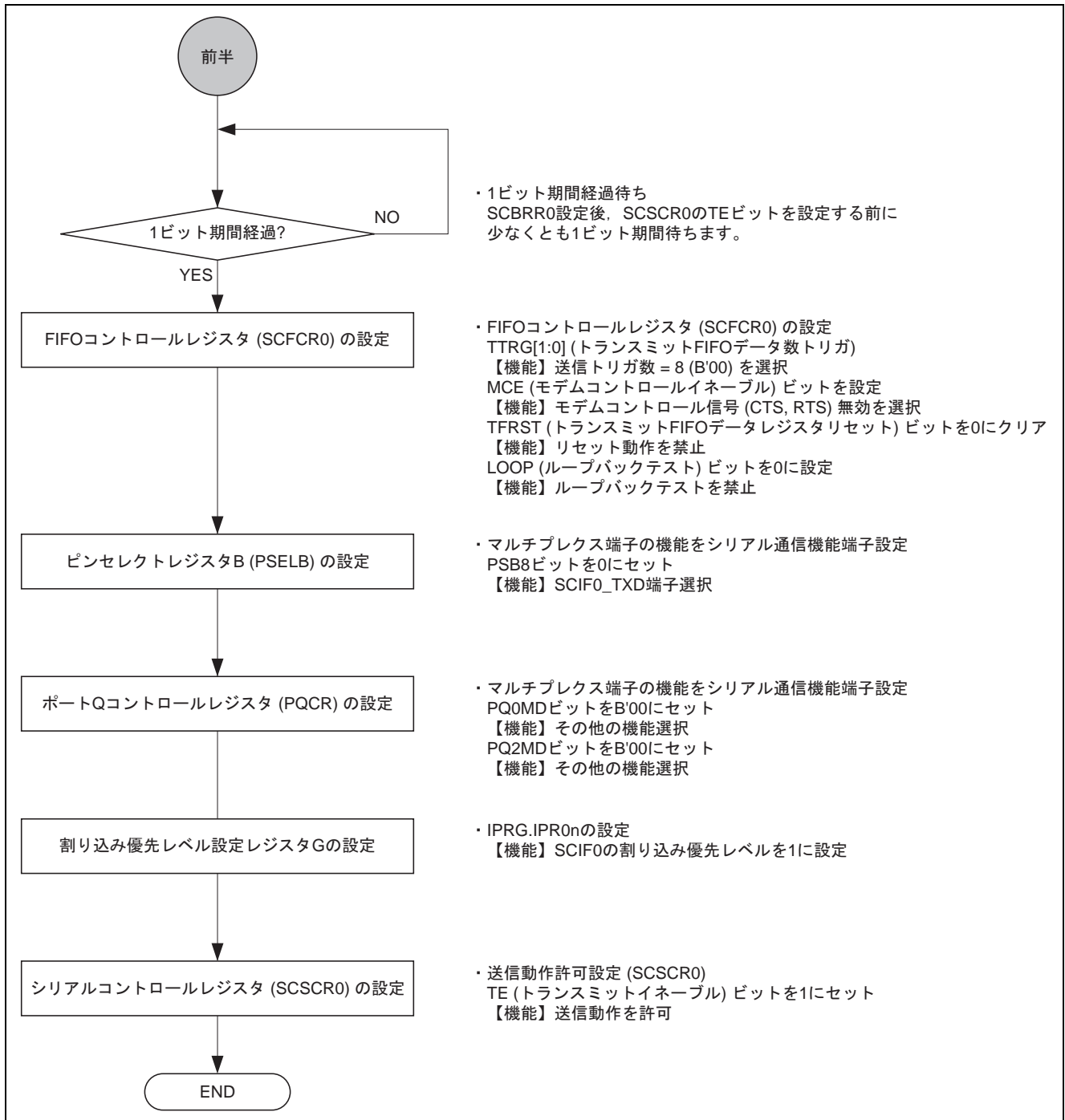


図5 クロック同期式モード送信初期設定フロー例 (後半)

SCIF1クロック同期式受信モード初期化処理

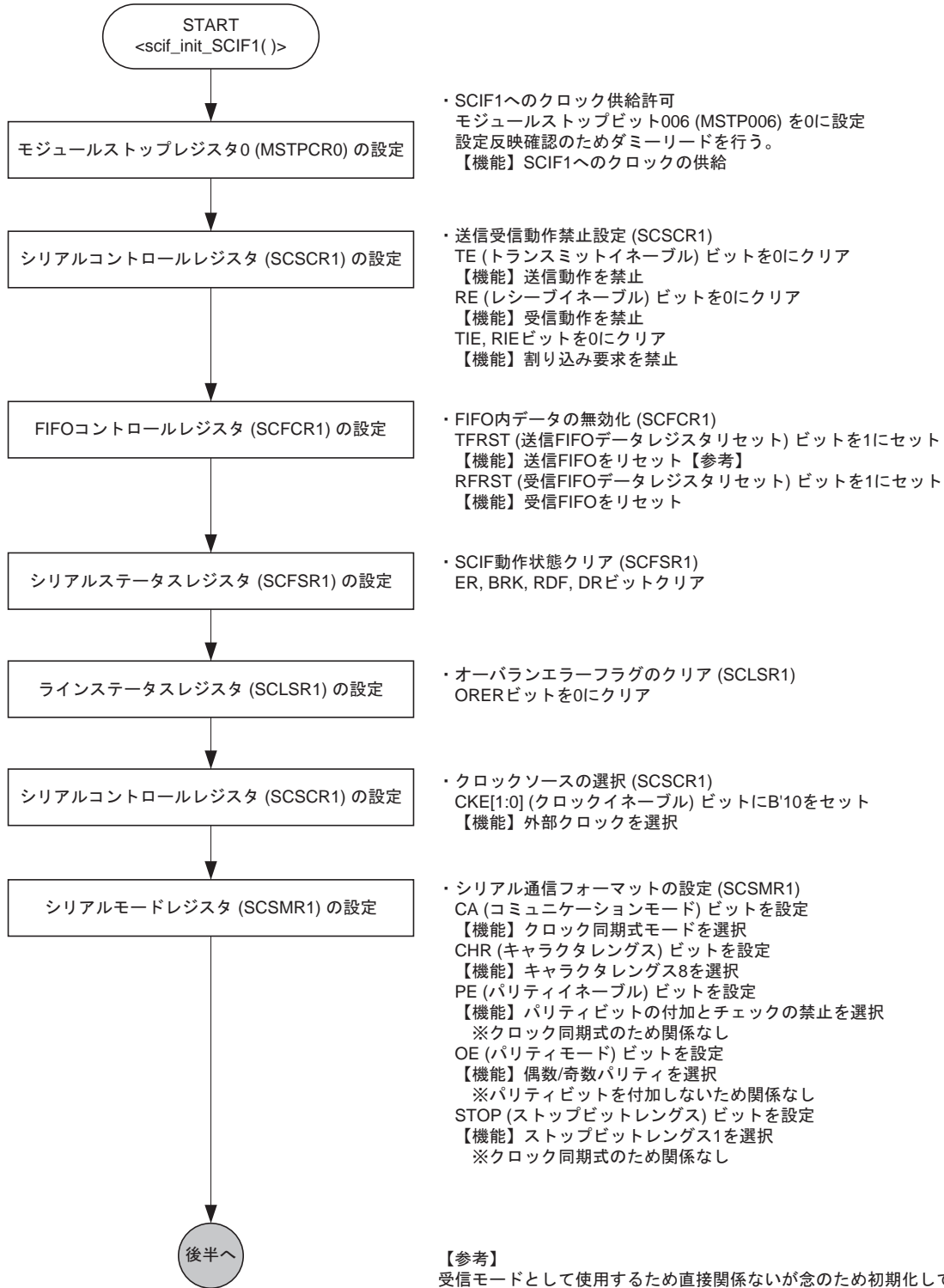


図 6 クロック同期式モード受信初期設定フロー例 (前半)

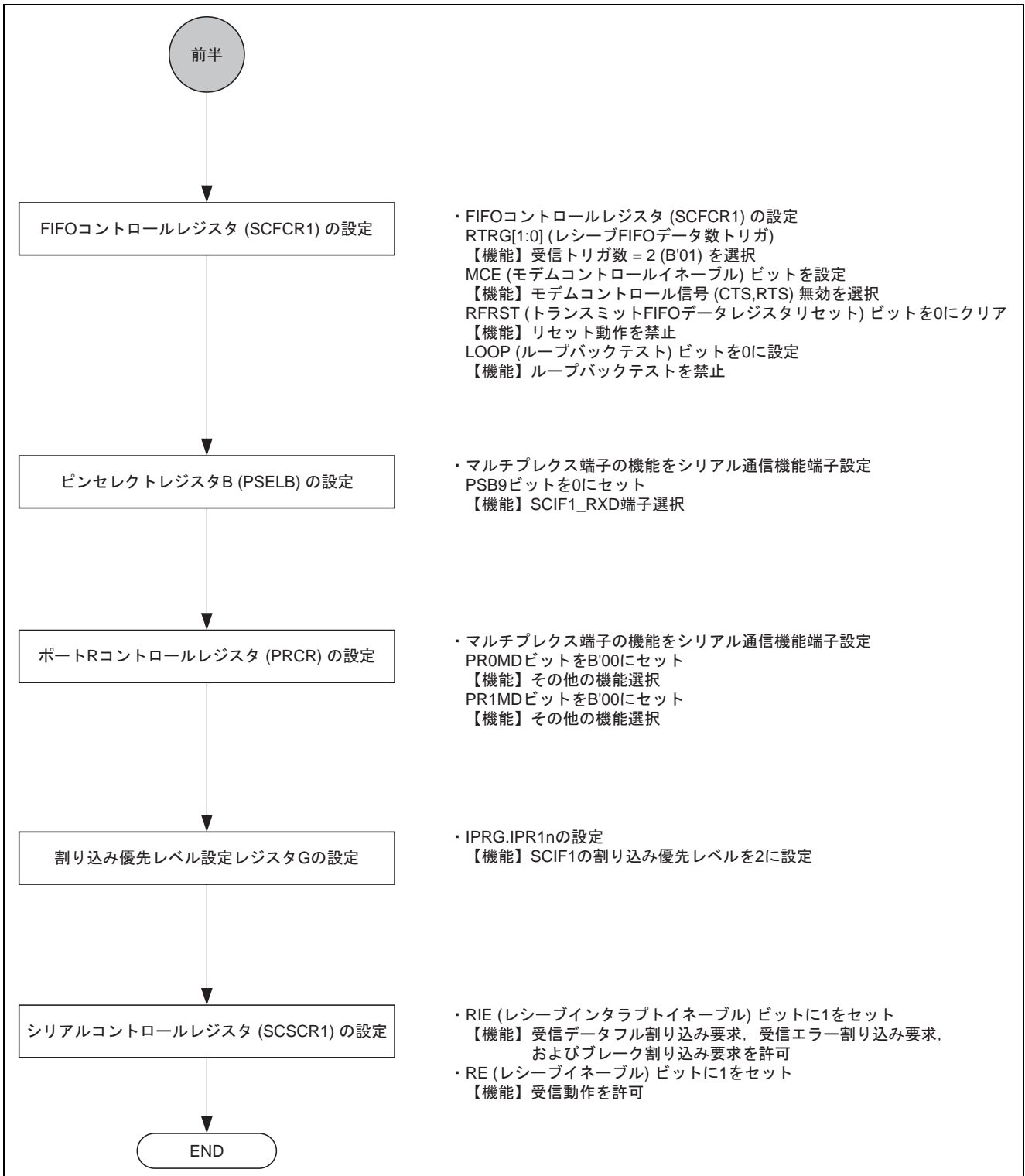


図7 クロック同期式モード受信初期設定フロー例 (後半)

### 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pty割り込み (TXI) 要求を許可にします (図 10)。

送信 FIFO データエンpty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)

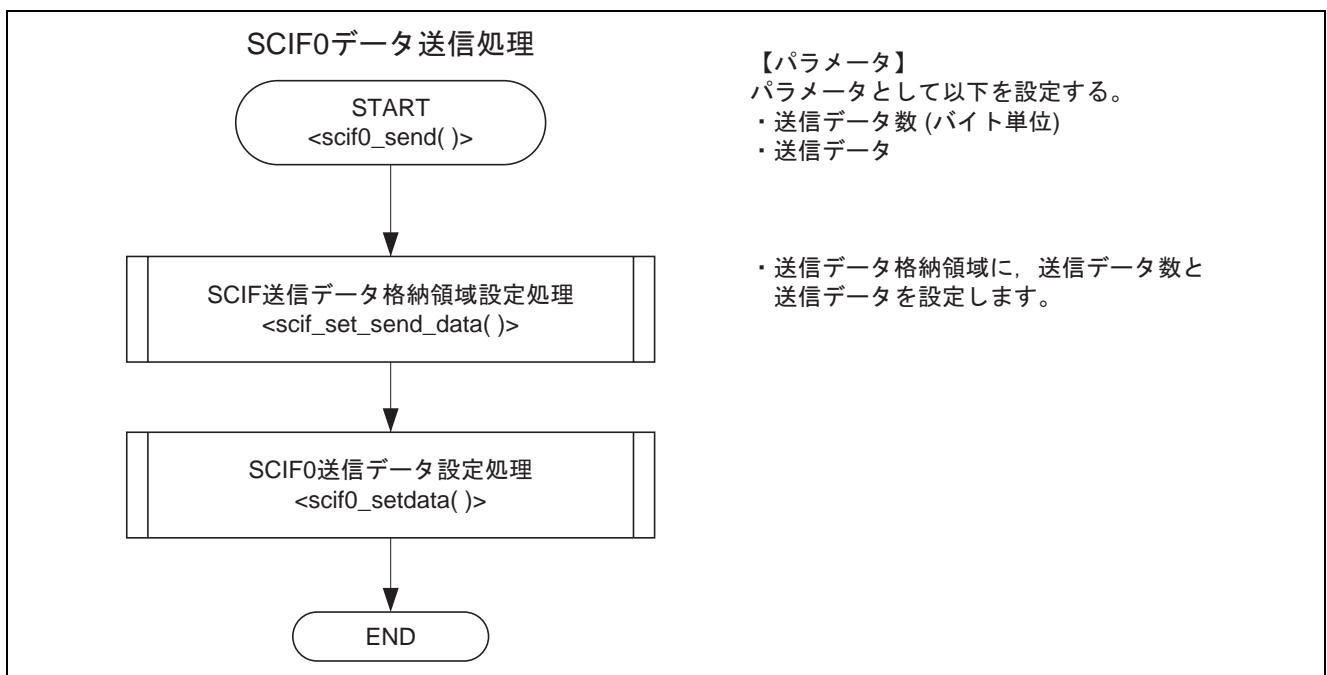


図 8 SCIF0 データ送信処理フロー

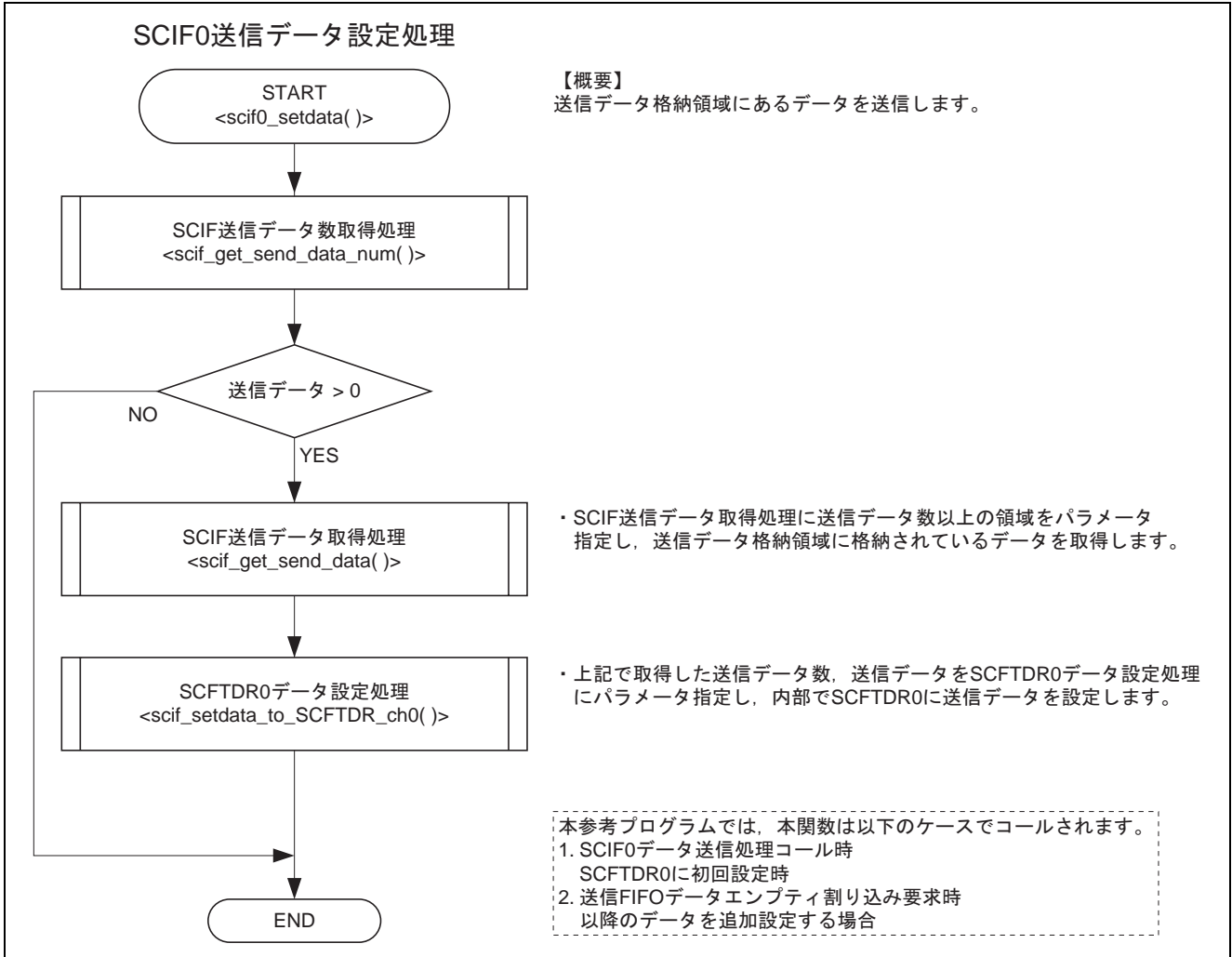


図 9 SCIF0 送信データ設定処理フロー

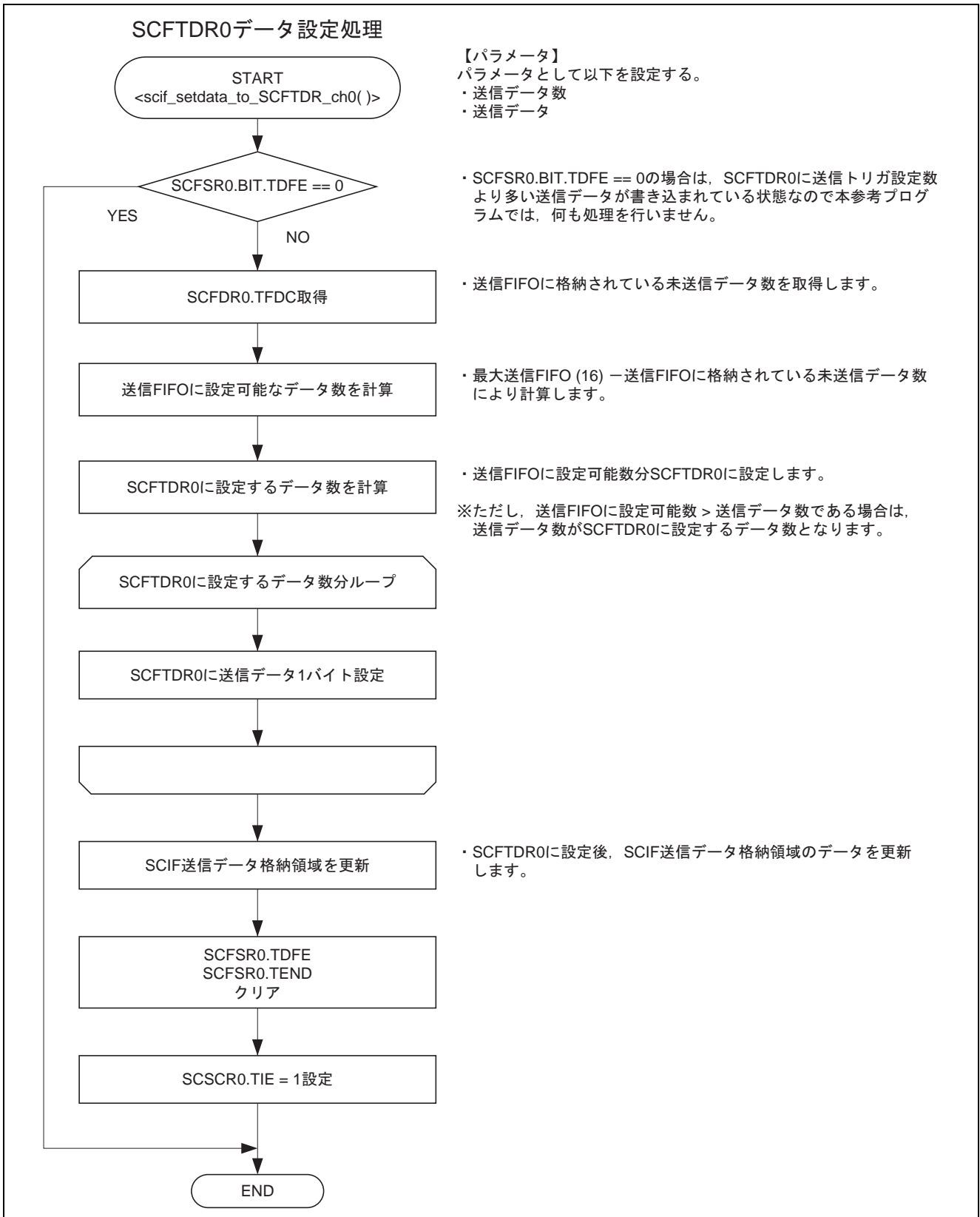


図 10 SCFTDR0 データ設定処理フロー



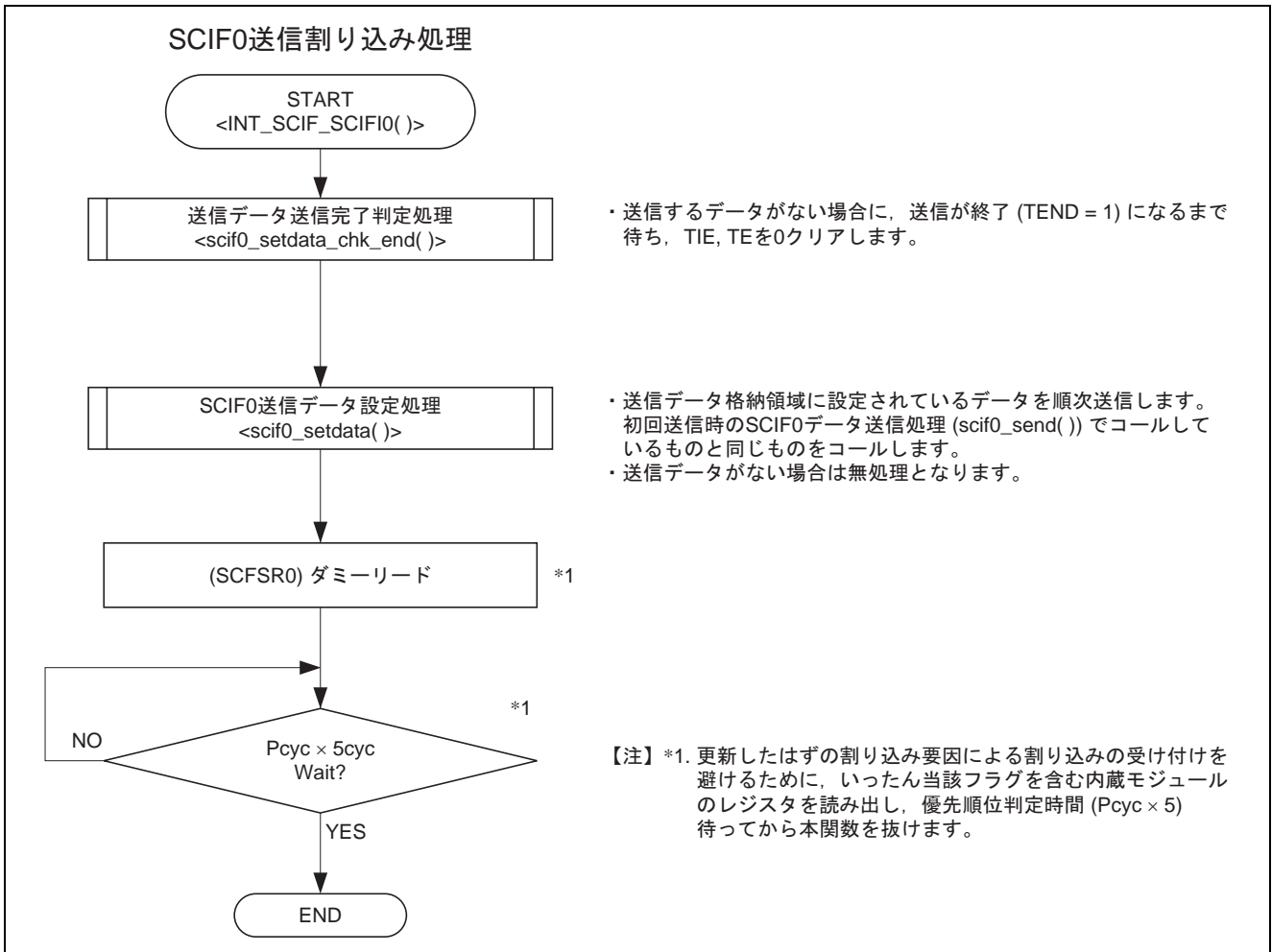


図 11 SCIF0 送信割り込み処理フロー

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, 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)

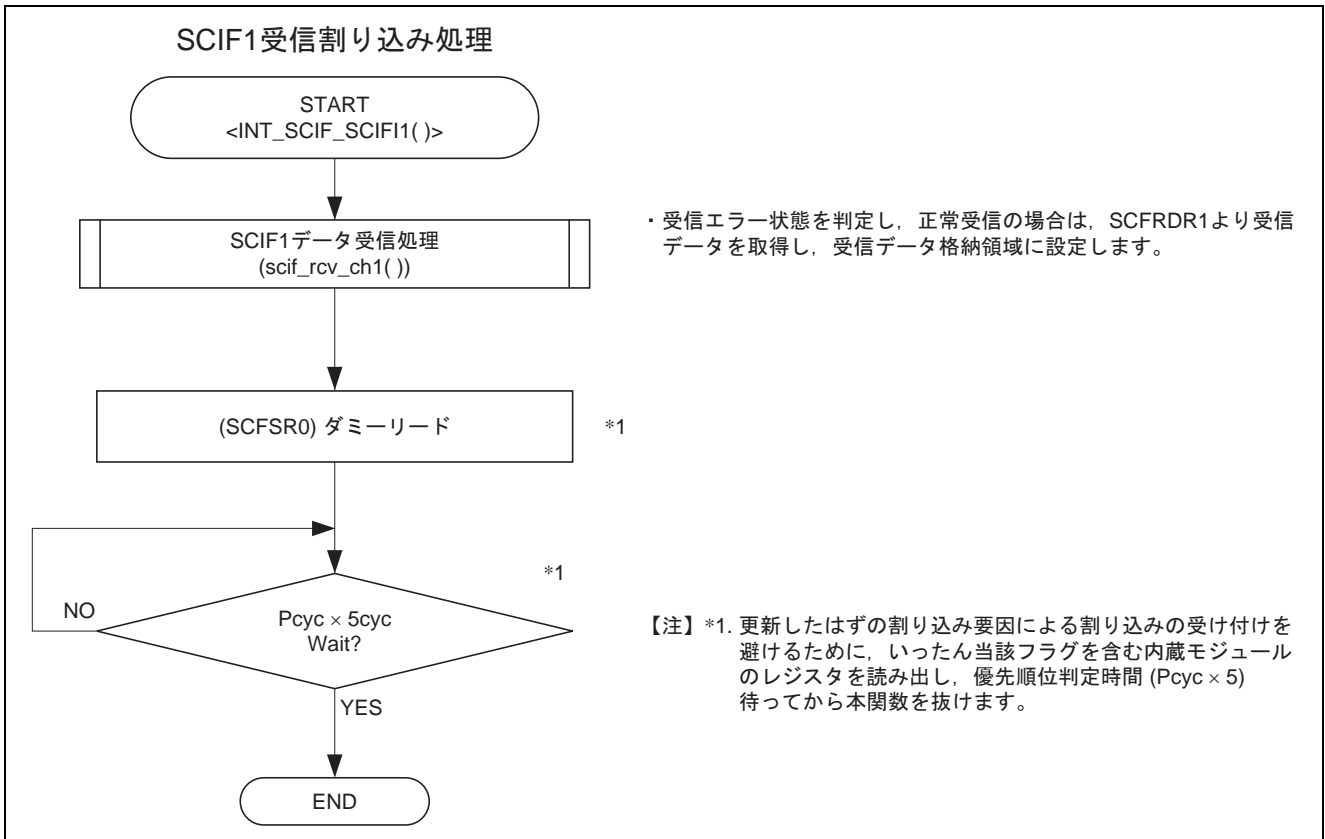


図 12 SCIF1 データ受信割り込み処理フロー

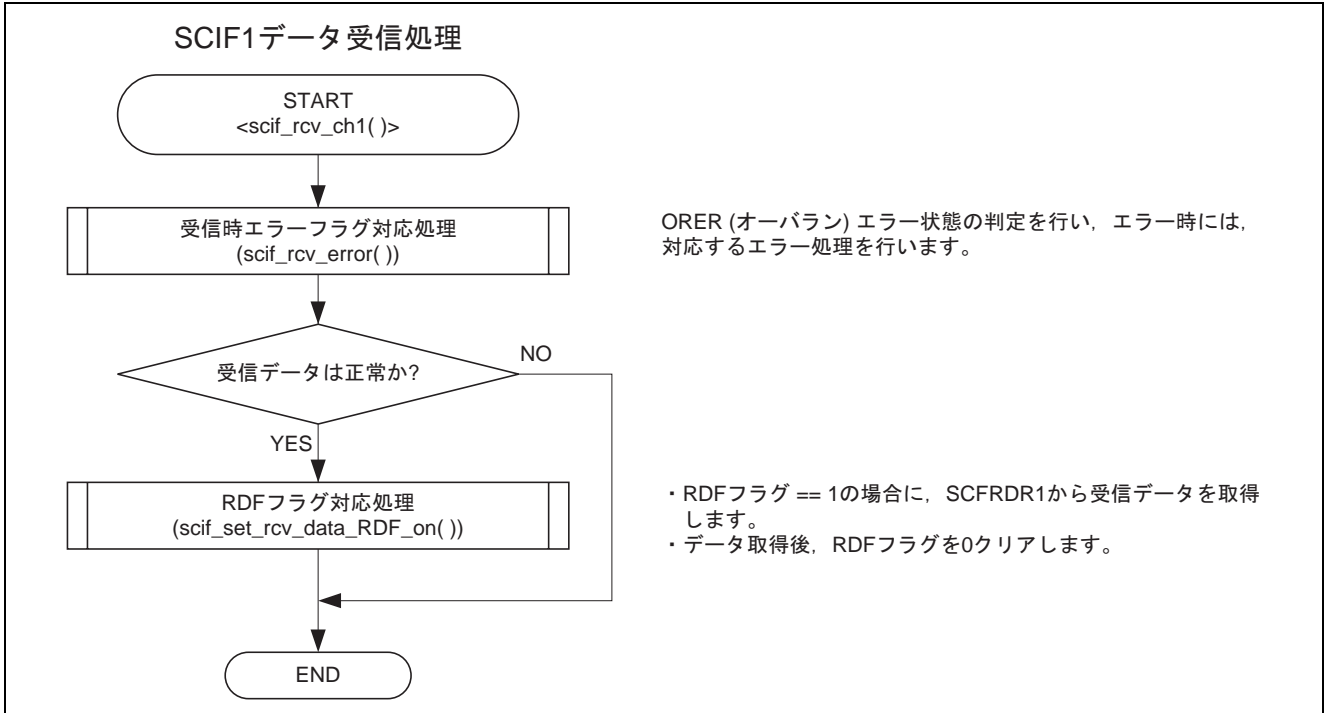


図 13 SCIF1 データ受信処理フロー

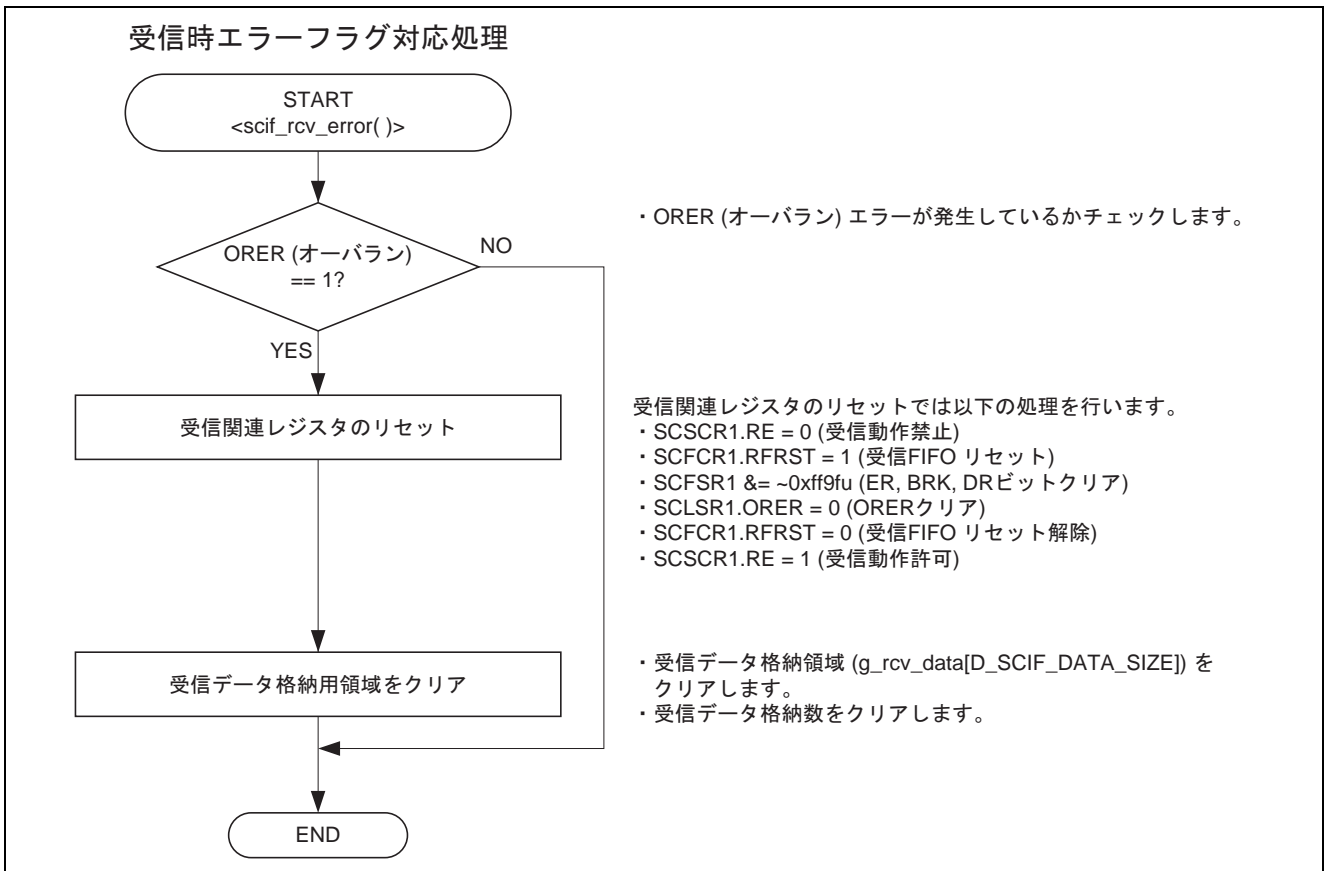


図 14 SCIF1 受信時エラーフラグ対応処理フロー

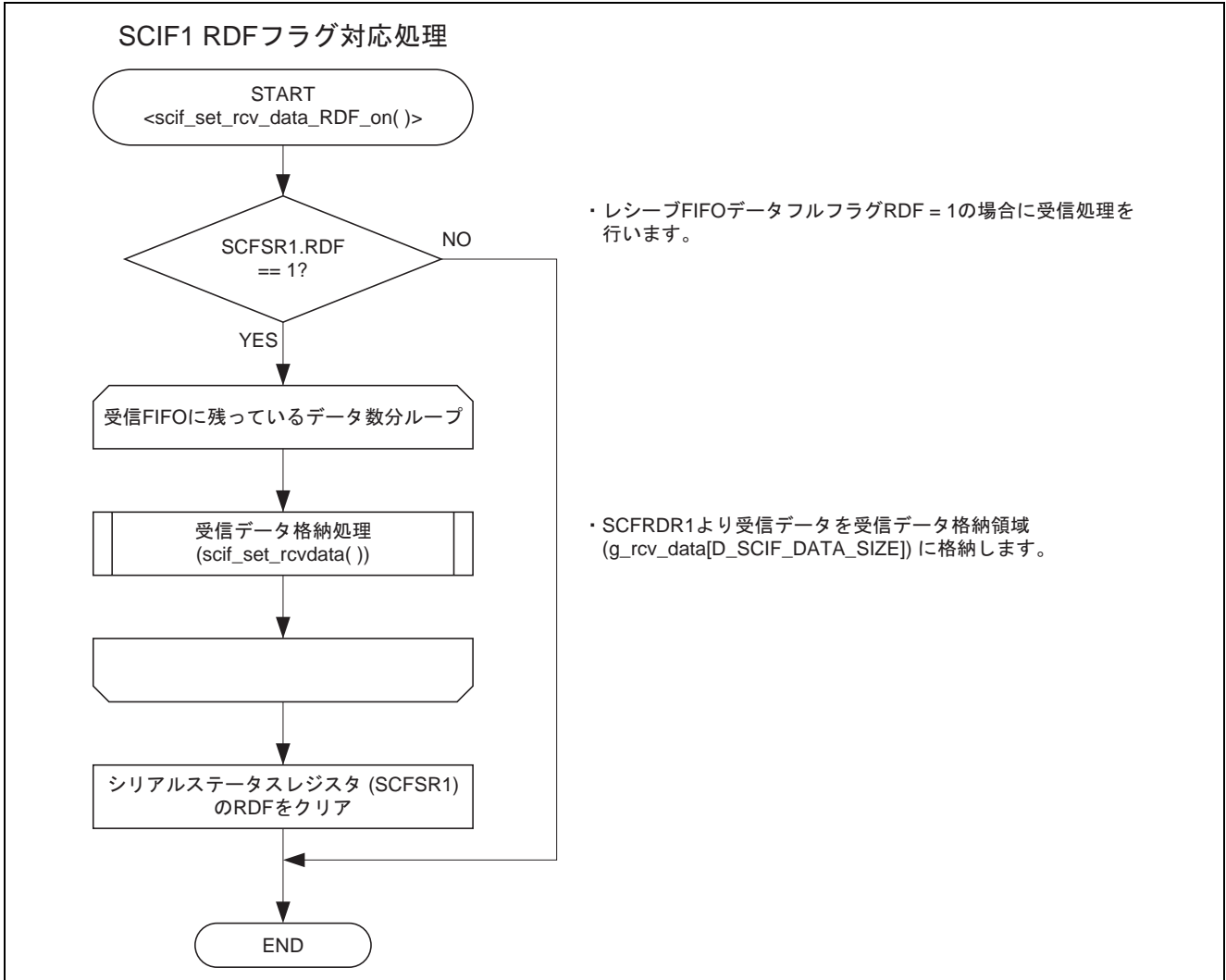


図 15 SCIF1 RDF フラグ対応処理フロー

### 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 をクロック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 をクロック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。

**【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 ORER (オーバラン) 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。
  - 受信データ数、受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE]) をクリアします。

**【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'0021	<ul style="list-style-type: none"> <li>初期化処理(2)</li> <li>CKE[1:0] = "B'01": 内部クロック/SCK 端子は同期クロック出力</li> <li>TE = "1": 送信動作を許可</li> </ul>
		H'0001	<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>初期化処理(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>初期化処理(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>
シリアルモードレジスタ (SCSMR0)	H'FFE00000	H'0080	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>CA = "1": クロック同期モード</li> <li>CKS[1:0] = "0": Pφ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>
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>初期化処理時</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>初期化処理時</li> <li>ORER ビットクリア</li> </ul>
ビットレートレジスタ (SCBRR0)	H'FFE00004	H'0052	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>100kbps 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>初期化処理時</li> <li>PSB8 = "0": SCIF0_TXD の選択</li> </ul>
ポート Q コントロールレジスタ (PQCR)	H'A405001A	H'0008	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>PQ0MD[1:0] = "0": その他の機能選択</li> <li>PQ2MD[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'1000	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>IPR0 = "B'0001": 優先レベル 1 設定</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'0052	<ul style="list-style-type: none"> <li>初期化処理(2)</li> <li>CKE[1:0] = "B'10": 外部クロック/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 = 2 #define により変更可能</li> </ul>
シリアルモードレジスタ (SCSMR1)	H'FFE10000	H'0080	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>CA = "1": クロック同期式モード</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>初期化処理時</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 (SCLSR1)	H'FFE10024	H'0000	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>ORER 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>初期化処理時</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'0020	<ul style="list-style-type: none"> <li>初期化処理時</li> <li>PR0MD[1:0] = "0": その他の機能選択</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>初期化処理時</li> <li>IPR1 = "B'0010": 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	2	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, 2, 8, 14 から選択可能。 本参考プログラムでは D_SCIF_RCV_FIFO_TRIGGER2 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_TRIGGER2	2	レシーブ FIFO データ数トリガ: 2
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_1K	0	ビットレート指定値: 1kbps *
D_SCIF_SCBRR_2_5K	1	ビットレート指定値: 2.5kbps *
D_SCIF_SCBRR_5K	2	ビットレート指定値: 5kbps *
D_SCIF_SCBRR_10K	3	ビットレート指定値: 10kbps *
D_SCIF_SCBRR_25K	4	ビットレート指定値: 25kbps *
D_SCIF_SCBRR_50K	5	ビットレート指定値: 50kbps *
D_SCIF_SCBRR_100K	6	ビットレート指定値: 100kbps *
D_SCIF_SCBRR_250K	7	ビットレート指定値: 250kbps *
D_SCIF_SCBRR_500K	8	ビットレート指定値: 500kbps *

【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 gc\_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) 2009. 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 クロック同期式 単方向送受信設定例
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 * : クロック同期式単方向通信サンプルプログラム
42 * Operation :
43 * Limitation :
44 * :
45 *****/
46 * History : 27.Feb.2009 Ver. 1.00 First Release
47 *"FILE COMMENT END"*****
48
49 #include <machine.h>
50 #include "iodefine.h"
51 #include "scif.h"
    
```

```

52
53 void main(void);
54
55 /*"FUNC COMMENT"*****
56 * ID :
57 * Outline : サンプルプログラムメイン
58 * : (クロック同期式シリアル送信・受信処理)
59 * Include :
60 * Declaration : void main(void)
61 * Description : クロック同期式シリアル単方向通信例を示します。
62 * :
63 * : SCIF0 からデータを送信し SCIF1 でデータが
64 * : 正しく受信できていることを確認します。
65 * :
66 * : 1.SCIF0 を送信モードで初期化します。
67 * : 2.SCIF1 を受信モードで初期化します。
68 * : 3.データを SCIF0 から順次送信します。
69 * : 4.データを SCIF1 から順次受信します。
70 * : 5.全データを受信するまで待ちます。
71 * : 6.SCIF1 の受信データ格納領域のデータと
72 * : 送信データが等しいことを確認します。
73 * :
74 * Argument : none
75 * Return Value : none
76 * Calling Functions :
77 /*"FUNC COMMENT END"*****/
78 void main(void)
79 {
80     unsigned char senddata[] =
81         "1234567890ABCDEFGHIJKLMNQRSTUWXYZ"; /* 全送信データ */
82     unsigned long send_num = 36; /* 送信データ数 */
83     unsigned char *ptr;
84     unsigned char rcvdata[D_SCIF_DATA_SIZE]; /* ローカル受信データ用 */
85     unsigned long rcv_num = 0; /* ローカル受信データ数 */
86     int result = 0;
87
88
89     memset(rcvdata, 0x00, sizeof(rcvdata));
90
91     /* 受信データ格納領域をクリアする */
92     scif_clear_rcvdata();
93
94     /* ==== SCIF0 をクロック同期式の送信モードに初期化 ==== */
95     scif_init_SCIF0();
96
97     /* ==== SCIF1 をクロック同期式の受信モードに初期化 ==== */
98     scif_init_SCIF1();
99
100    ptr = senddata;
101
102    /* ==== SCIF0 データ送信処理 ==== */
103    scif0_send(send_num, ptr); /* 送信データ数 */
104                                /* 送信データ */
105
106
107    /* 送受信完了 wait 処理 */
108    /* 本参考プログラムでは割り込み処理で送受信が完全に終了するまでここで wait します */

```

```

109     /* Wait 完了後，受信 FIFO の残データを受信データ格納領域に設定します */
110     scif_send_rcv_end_wait();
111
112
113     /* 受信データ確認用ループ */
114     while (1)
115     {
116         /* 受信データ格納領域よりデータを取得 */
117         scif_get_rcvdata(&rcv_num, rcvdata);
118
119         /* 送信データ数 == 受信データ数となった時 */
120         if(send_num == rcv_num)
121         {
122             /* 送信データと受信データを比較する */
123             result = memcmp(senddata, rcvdata, rcv_num);
124
125             /* 一致した場合ループを抜ける */
126             if(result == 0)
127             {
128                 break;
129             }
130         }
131
132     }
133
134     /* 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 */
135     while (1)
136     {
137
138     }
139 }
    
```

## (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 クロック同期式 単方向送受信設定例
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 *             : クロック同期式単方向通信サンプルプログラム
42 * Operation   :
43 * Limitation  :
44 *             :
45 *****/
46 * History     : 27.Feb.2009 Ver. 1.00 First Release
47 /*"FILE COMMENT END"*****
48
49 #include <machine.h>
50 #include <stdio.h>
51 #include <stdlib.h>
52 #include "iodefine.h"
53 #include "scif.h"
    
```

```

54
55  /*"VAL DEF"*/
56  /* ===== 送信データ格納用 ===== */
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
60  /* ===== 受信データ格納用 ===== */
61  unsigned char   g_rcv_data[D_SCIF_DATA_SIZE];           /* 受信データ格納領域 */
62  unsigned long   g_rcv_count = 0;                       /* 受信データ数 */
63
64  /* ===== レジスタ(SCBRR, CKS)設定値テーブル ===== */
65  /* 注意
66     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
67     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
68  */
69  const static T_SCIF_CKS_SCBRR_SET_INFO gc_scif_cks_scbrr_tbl[] =
70  {
71      /* SCBRR, CKS, lbitwait */
72      { 129, 3, 266667}, /* 1Kbps */
73      { 52, 3, 106667}, /* 2.5Kbps */
74      { 104, 2, 53333}, /* 5Kbps */
75      { 52, 2, 26667}, /* 10Kbps */
76      { 83, 1, 10667}, /* 25Kbps */
77      { 166, 0, 5333}, /* 50Kbps */
78      { 82, 0, 2667}, /* 100Kbps */
79      { 33, 0, 1067}, /* 250Kbps */
80      { 16, 0, 533} /* 500Kbps */
81  };
82
83  /* ===== レジスタ(SCFCR.TTRG)設定値テーブル ===== */
84  const static T_SCIF_SCFCR_TTRG_SET gc_scif_ttrg_tbl[] =
85  {
86      { D_SCIF_SEND_FIFO_TRIGGER8, 0x00}, /* トランスミット FIFO データ数トリガ = 8 */
87      { D_SCIF_SEND_FIFO_TRIGGER4, 0x01}, /* トランスミット FIFO データ数トリガ = 4 */
88      { D_SCIF_SEND_FIFO_TRIGGER2, 0x02}, /* トランスミット FIFO データ数トリガ = 2 */
89      { D_SCIF_SEND_FIFO_TRIGGER0, 0x03} /* トランスミット FIFO データ数トリガ = 0 */
90  };
91
92  /* ===== レジスタ(SCFCR.RTRG)設定値テーブル ===== */
93  const static T_SCIF_SCFCR_RTRG_SET gc_scif_rtrg_tbl[] =
94  {
95      { D_SCIF_RCV_FIFO_TRIGGER1, 0x00}, /* レシーブ FIFO データ数トリガ = 1 */
96      { D_SCIF_RCV_FIFO_TRIGGER2, 0x01}, /* レシーブ FIFO データ数トリガ = 2 */
97      { D_SCIF_RCV_FIFO_TRIGGER8, 0x02}, /* レシーブ FIFO データ数トリガ = 8 */
98      { D_SCIF_RCV_FIFO_TRIGGER14, 0x03} /* レシーブ FIFO データ数トリガ = 14 */
99  };
100 /*"VAL DEF END"*/
101
102 /* ===== プロトタイプ宣言 ===== */
103 static unsigned char scif_get_clock_select(T_SCIF_bit_rate_type i_type_bps);
104 static unsigned char scif_get_SCBRR(T_SCIF_bit_rate_type i_type_bps);
105 static void scif_setdata_to_SCFTDR_ch0(unsigned long i_num, unsigned char *i_pdata);
106 static void scif_set_rcvdata(unsigned char *i_pRcv_data);
107 static void scif_reset_rcvdata_status(void);
108 static void scif_set_rcv_data_RDF_on(void);
109 static int scif_rcv_error(void);
110 static unsigned char scif_get_snd_trigger_num(T_SCIF_send_fifo_trigger i_trigger_num);
    
```

```

111 static unsigned char scif_get_rcv_trigger_num(T_SCIF_rcv_fifo_trigger i_trigger_num);
112 static void scif_wait_lbit(unsigned long i_bit_rate);
113
114 /*"FUNC COMMENT"*****
115 * ID :
116 * Outline : SCIF0 初期化处理
117 * Include :
118 * Declaration : void scif_init_SCIF0(void)
119 * Description : SCIF0 を送信モードで以下のように初期化
120 * : します。
121 * : SCSMR
122 * : CA = クロック同期式
123 * : CHR = クロック同期式は 8 ビットデータ固定
124 * : PE = クロック同期式のため関係なし
125 * : OE = クロック同期式のため関係なし
126 * : STOP = クロック同期式のため関係なし
127 * : CKS = gc_scif_cks_scbrr_tbl[]より設定
128 * : SCSCR
129 * : TIE = 割り込み要求禁止
130 * : RIE = 割り込み要求禁止
131 * : TE = 送信動作を禁止 許可
132 * : RE = 受信動作を禁止
133 * : REIE = 割り込み要求禁止
134 * : CKE = 内部クロック
135 * : SCFSR
136 * : ER, TEND, TDFE, BRK, RDF, DR ビットクリア
137 * : SCBRR
138 * : gc_scif_cks_scbrr_tbl[]より設定
139 * : ビットレートは, 100kbps で指定します。
140 * : SCFCR
141 * : RSTRG = モデム信号無効のため関係なし
142 * : RTRG = 送信モードのため関係なし
143 * : TTRG = トランスミット FIFO データ数トリガ 8
144 * : MCE = モデム信号を無効
145 * : TFRST = リセット動作を許可 禁止
146 * : RFRST = リセット動作を許可 禁止
147 * : LOOP = ループバックテストを禁止
148 * : SCLSR
149 * : ORER = オーバランエラーフラグクリア
150 * :
151 * Limitation : ポーレート設定値は内部クロックを使った
152 * : 周辺モジュール用動作周波数 P = 33.33MHz
153 * : の場合です。
154 * : 他のクロックを使用する場合はビットレート
155 * : 設定値を変更してください。
156 * : gc_scif_cks_scbrr_tbl[]の設定値も
157 * : 変更してください。
158 * :
159 * Argument : none
160 * Return Value : none
161 * Calling Functions :
162 /*"FUNC COMMENT END"*****/
163 void scif_init_SCIF0(void)
164 {
165     unsigned long dummy;
166
167     /* ==== SCIF0 の初期設定 ==== */
    
```

```

168     /* ==== モジュールストップレジスタの設定 ==== */
169     LOWP.MSTPCR0 &= ~0x00000080;
170
171     dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
172
173     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
174     SCIF0.SCSCR.WORD = 0x0000; /* SCIF0 送/受信動作停止 */
175
176     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
177     SCIF0.SCFCR.BIT.TFRST = 1; /* 送信 FIFO リセット */
178     SCIF0.SCFCR.BIT.RFRST = 1; /* 受信 FIFO リセット */
179
180     /* ==== シリアルステータスレジスタ(SCFSR)の初期化 ==== */
181     SCIF0.SCFSR.WORD &= ~0xff9f; /* ER,BRK,RDF ビットクリア */
182
183     /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
184     SCIF0.SCLSR.BIT.ORER = 0; /* OREO ビットクリア */
185
186     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
187     SCIF0.SCSCR.BIT.CKE = 1; /* B'01:内部クロック 同期クロック出力 */
188
189     /* ==== シリアルモードレジスタ(SCSMR)の設定 ==== */
190     /* コミュニケーションモード : クロック同期式 */
191     SCIF0.SCSMR.WORD = 0x0080;
192
193     /* ==== クロックセレクト : ビットレートの設定値により決定 ==== */
194     SCIF0.SCSMR.BIT.CKS = scif_get_clock_select(D_SCIF_SCBRR_100K);
195
196     /* ==== 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 ==== */
197     SCIF0.SCBRR = scif_get_SCBRR(D_SCIF_SCBRR_100K);
198
199     /* 1 ビット期間経過待ち */
200     scif_wait_lbit(D_SCIF_SCBRR_100K);
201
202     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
203     /* 送信 FIFO データレジスタリセット : 禁止 */
204     /* ループバックテスト : 禁止 */
205     SCIF0.SCFCR.WORD = 0x0000;
206
207     /* 送信 FIFO データ数トリガ : データ数 8 */
208     /* 送信 FIFO データ数トリガは gc_scif_ttrg_tbl[] の設定値になります */
209     SCIF0.SCFCR.BIT.TTRG = scif_get_snd_trigger_num(D_SCIF_DATA_NUM_SND_TRIGGER_CH0);
210
211     /* ==== ピンセレクトレジスタ B (PSELB) ==== */
212     PFC.PSELB.BIT.PSB8 = 0; /* SCIF0_TXD の選択 */
213
214     /* ==== ポート Q コントロールレジスタ (PQCR) の設定 ==== */
215     PFC.PQCR.BIT.PQ2MD = 0; /* その他の機能選択 */
216     PFC.PQCR.BIT.PQ0MD = 0; /* その他の機能選択 */
217
218     /* ==== 割り込み優先レベル設定 (優先レベル 1) ==== */
219     INTC0.IPRG = INTC0.IPRG | 0x1000;
220
221     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
222     SCIF0.SCSCR.BIT.TE = 1; /* SCIF0 送信動作を許可する */
223
224 }
    
```



```

225
226 /*"FUNC COMMENT"*****
227 * ID :
228 * Outline : SCIF1 初期化处理
229 * Include :
230 * Declaration : void scif_init_SCIF1(void)
231 * Description : SCIF1 を受信モードで以下のように初期化
232 * : します。
233 * : SCSMR
234 * : CA = クロック同期式
235 * : CHR = クロック同期式は 8 ビットデータ固定
236 * : PE = クロック同期式のため関係なし
237 * : OE = クロック同期式のため関係なし
238 * : STOP = クロック同期式のため関係なし
239 * : SCSCR
240 * : TIE = 割り込み要求禁止
241 * : RIE = 割り込み要求許可
242 * : TE = 送信動作を禁止
243 * : RE = 受信動作を禁止 許可
244 * : REIE = 割り込み要求禁止
245 * : CKE = 外部クロック
246 * : SCFSR
247 * : ER, TEND, TDFE, BRK, RDF, DR ビットクリア
248 * : SCFCR
249 * : RSTRG = モデム信号無効のため関係なし
250 * : RTRG = レシーブ FIFO データ数トリガ 2 を設定
251 * : TTRG = 受信モードのため関係なし
252 * : MCE = モデム信号を無効
253 * : TFRST = リセット動作を許可 禁止
254 * : RFRST = リセット動作を許可 禁止
255 * : LOOP = ループバックテストを禁止
256 * : SCLSR
257 * : ORER = オーバランエラーフラグクリア
258 * :
259 * Argument : none
260 * Return Value : none
261 * Calling Functions :
262 /*"FUNC COMMENT END"*****/
263 void scif_init_SCIF1(void)
264 {
265     unsigned long dummy;
266
267     /* ==== SCIF1 の初期設定 ==== */
268     /* ==== モジュールストップレジスタの設定 ==== */
269     LOWP.MSTPCR0 &= ~0x00000040;
270
271     dummy = LOWP.MSTPCR0; /* 設定反映確認のためダミリード */
272
273     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
274     SCIF1.SCSCR.WORD = 0x0000; /* SCIF1 送/受信動作停止 */
275
276     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
277     SCIF1.SCFCR.BIT.TFRST = 1; /* 送信 FIFO リセット */
278     SCIF1.SCFCR.BIT.RFRST = 1; /* 受信 FIFO リセット */
279
280     /* ==== シリアルステータスレジスタ(SCFSR)の設定 ==== */
281     SCIF1.SCFSR.WORD &= ~0xff9f; /*各フラグクリア */
    
```

```

282
283     /* ==== ラインステータスレジスタ(SCLSR)の設定 ==== */
284     SCIF1.SCLSR.BIT.orer = 0;      /* ORER ビットクリア */
285
286     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
287     SCIF1.SCSCR.BIT.cke = 2;      /* B'10:外部クロック */
288
289     /* ---- シリアルモードレジスタ(SCSMR)の設定 ---- */
290     /* コミュニケーションモード : クロック同期式 */
291     SCIF1.SCSMR.WORD = 0x0080;
292
293     /* ==== FIFO コントロールレジスタ(SCFCR)の設定 ==== */
294     /* 受信 FIFO データレジスタリセット : 禁止 */
295     /* ループバックテスト : 禁止 */
296     SCIF1.SCFCR.WORD = 0x0000;
297
298     /* 受信 FIFO データ数トリガ : データ数 2 */
299     /* 受信 FIFO データ数トリガは gc_scif_rtrg_tbl[] の設定値になります */
300     SCIF1.SCFCR.BIT.rtrg = scif_get_rcv_trigger_num(D_SCIF_DATA_NUM_RCV_TRIGGER_CH1);
301
302     /* ==== ピンセレクトレジスタ B (PSELB) ==== */
303     PFC.PSELB.BIT.psb9 = 0;      /* SCIF1_RXD の選択 */
304
305     /* ==== ポート R コントロールレジスタ (PRCR) の設定 ==== */
306     PFC.PRCR.BIT.pr1md = 0;      /* その他の機能選択 */
307     PFC.PRCR.BIT.promd = 0;      /* その他の機能選択 */
308
309     /* ==== 割り込み優先レベル設定 (優先レベル 2) ==== */
310     INTC0.IPRG = INTC0.IPRG | 0x0200;
311
312     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
313     SCIF1.SCSCR.BIT.rie = 1;      /* 受信データフル割り込み有効 */
314     SCIF1.SCSCR.BIT.re = 1;      /* SCIF1 受信動作を許可する */
315
316 }
317
318 /*"FUNC COMMENT"*****
319 * ID :
320 * Outline : クロックセレクト設定値取得
321 * Include :
322 * Declaration : unsigned char scif_get_clock_select
323 * : (T_SCIF_bit_rate_type i_type_bps)
324 * Description : ビットレートに対応するクロックセレクト
325 * : CKS[1:0]設定値を取得する。
326 * :
327 * Argument : T_SCIF_bit_rate_type i_type_bps :
328 * : ビットレート指定値
329 * Return Value : unsigned char : クロックセレクト設定値
330 * Calling Functions :
331 *"FUNC COMMENT END"*****/
332 unsigned char scif_get_clock_select(T_SCIF_bit_rate_type i_type_bps)
333 {
334     return gc_scif_cks_scbrtbl[i_type_bps].mcsmr_cks;
335 }
336
337 /*"FUNC COMMENT"*****
338 * ID :
    
```

```

339 * Outline           : ビットレートレジスタ設定値(SCBRR)取得
340 * Include           :
341 * Declaration       : unsigned char scif_get_SCBRR
342 *                   : (T_SCIF_bit_rate_type i_type_bps)
343 * Description       : ビットレートに対応するビットレート
344 *                   : レジスタ設定値を取得します。
345 *                   :
346 * Argument          : T_SCIF_bit_rate_type i_type_bps :
347 *                   : ビットレート指定値
348 * Return Value      : unsigned char : ビットレートレジスタ設定値
349 * Calling Functions :
350 * "FUNC COMMENT END"*****/
351 unsigned char scif_get_SCBRR(T_SCIF_bit_rate_type i_type_bps)
352 {
353     return      gc_scif_cks_scbrr_tbl[i_type_bps].mscbrr;
354 }
355
356 /*"FUNC COMMENT"*****
357 * ID                 :
358 * Outline            : SCIF0 データ送信処理
359 * Include            :
360 * Declaration        : void scif0_send
361 *                   : (unsigned long i_count,
362 *                   : unsigned char *i_ptr)
363 *                   :
364 * Description        : 送信データを送信データ格納領域に保存し
365 *                   : SCIF0 送信データ設定処理をコールします。
366 *                   :
367 * Argument           : unsigned long i_count :
368 *                   : 送信データ数
369 *                   : unsigned char *i_ptr :
370 *                   : 送信データ
371 * Return Value       : none
372 * Calling Functions :
373 * "FUNC COMMENT END"*****/
374 void scif0_send(unsigned long i_count, unsigned char *i_ptr)
375 {
376     /* SCIF 送信データ格納領域設定処理 */
377     scif_set_send_data(i_count, i_ptr);
378
379     /* SCIF0 送信データ設定処理 */
380     scif0_setdata();
381
382 }
383
384 /*"FUNC COMMENT"*****
385 * ID                 :
386 * Outline            : 送信データ送信完了判定処理
387 * Include            :
388 * Declaration        : void scif0_setdata_chk_end(void)
389 *                   :
390 * Description        : 送信データ格納領域の送信データがあるか
391 *                   : どうか確認し、送信データがない場合には
392 *                   : TEND が 1 になるまで待ち
393 *                   : SCSCR の TIE , TE を 0 クリアします。
394 *                   :
395 * Argument           : none
    
```

```

396 * Return Value          : none
397 * Calling Functions     :
398 * "FUNC COMMENT END"*****/
399 void scif0_setdata_chk_end(void)
400 {
401     unsigned long        send_num = 0;
402
403     /* SCIF 送信データ数取得処理 */
404     scif_get_send_data_num(&send_num);
405
406     /* 全てのデータを送信完了した場合 */
407     if(send_num == 0)
408     {
409         /* ==== 送信完了判定 ==== */
410         while(SCIF0.SCFSR.BIT.TEND == 0)
411         {
412
413         }
414
415         /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
416         SCIF0.SCSCR.BIT.TIE = 0; /* 送信データエンプティ割り込み無効 */
417
418         /* ==== SCIF0 送信動作を無効にする ==== */
419         SCIF0.SCSCR.BIT.TE = 0;
420
421     }
422
423 }
424
425 /* "FUNC COMMENT"*****
426 * ID                      :
427 * Outline                 : SCIF0 送信データ設定処理
428 * Include                 :
429 * Declaration             : void scif0_setdata(void)
430 *
431 * Description             : 送信データ格納領域から送信データを取得し
432 *                          : SCFTDR0 データ設定処理をコールします。
433 *
434 * Argument                : none
435 *
436 * Return Value            : none
437 * Calling Functions       :
438 * "FUNC COMMENT END"*****/
439 void scif0_setdata(void)
440 {
441     unsigned long        send_num = 0;
442     unsigned char        send_data[D_SCIF_DATA_SIZE];
443
444     memset(send_data, 0x00, sizeof(send_data));
445
446     /* SCIF 送信データ数取得処理 */
447     scif_get_send_data_num(&send_num);
448
449     /* 送信データがなければ無処理 */
450     if(send_num <= 0)
451     {
452         return;

```

```

453     }
454
455     /* SCIF 送信データ取得処理 */
456     scif_get_send_data(send_num, send_data);
457
458     /* SCFTDR0 データ設定処理 */
459     scif_setdata_to_SCFTDR_ch0(send_num, send_data);
460
461 }
462
463 /*"FUNC COMMENT"*****
464 * ID          :
465 * Outline     : SCFTDR0 データ設定処理
466 * Include     :
467 * Declaration : void scif_setdata_to_SCFTDR_ch0
468 *             : (unsigned long i_num,
469 *             :   unsigned char *i_pdata)
470 *             :
471 * Description : TDFE の状態を確認し送信 FIFO データエンブティ
472 *             : であることを確認します。
473 *             : SCFDR.TFDC を確認して設定可能な数の
474 *             : 送信データを SCFTDR0 に設定します。
475 *             : データ設定後 TDFE, TEND をクリアします。
476 *             : SCFTDR0 に送信データを設定後, 送信データ
477 *             : 格納領域を更新します。
478 *             : SCFTDR0 に送信データを設定後,
479 *             : 送信データエンブティ割り込みを有効にします。
480 *             :
481 * Argument    : unsigned long i_num :
482 *             :   送信データ数
483 *             : unsigned char *i_pdata :
484 *             :   送信データ
485 *             :
486 * Return Value : none
487 * Calling Functions :
488 *"FUNC COMMENT END"*****/
489 void scif_setdata_to_SCFTDR_ch0(unsigned long i_num, unsigned char *i_pdata)
490 {
491     int          i = 0;
492     unsigned long set_send_fifo_num = 0;
493
494     /* ==== シリアルステータスレジスタ(SCSFR0)の
495     送信 FIFO データエンブティを確認 (TDFE フラグ) ==== */
496     if(SCIF0.SCFDR.BIT.TFDC == 0)
497     {
498         return;
499     }
500
501     /* 未送信データ数取得 */
502     set_send_fifo_num = SCIF0.SCFDR.BIT.TFDC;
503
504     /* 送信 FIFO に設定可能なデータ数を計算 */
505     set_send_fifo_num = D_SCIF_FIFO_NUM - set_send_fifo_num;
506
507     /* 送信 FIFO 設定数 */
508     if(set_send_fifo_num > i_num)
509     {

```

```

510     set_send_fifo_num = i_num;
511     }
512
513     /* ==== 送信 FIFO データレジスタ(SCFTDR0)に送信データを書き込む ==== */
514     for(i = 0; i < set_send_fifo_num; i++)
515     {
516         i_num--;
517         SCIF0.SCFTDR = *i_pdata++;
518     }
519
520
521     /* SCIF 送信データ格納領域を更新する */
522     scif_set_send_data(i_num, i_pdata);
523
524     /* ==== シリアルステータスレジスタ(SCFSR0)の TDFE, TEND を
525     読出し後にクリアする ==== */
526     SCIF0.SCFSR.BIT.TDFE = 0;
527     SCIF0.SCFSR.BIT.TEND = 0;
528
529     /* ==== シリアルコントロールレジスタ(SCSCR)の設定 ==== */
530     SCIF0.SCSCR.BIT.TIE = 1; /* 送信データエンベティ割り込み有効 */
531
532 }
533
534 /*"FUNC COMMENT"*****
535 * ID :
536 * Outline : SCIF 送信データ格納領域設定処理
537 * Include :
538 * Declaration : void scif_set_send_data
539 * : (unsigned long i_count,
540 * : unsigned char *i_ptr)
541 * Description : 送信データを送信データ格納領域に設定します。
542 * :
543 * Limitation : 送信データ格納領域より大きいデータの場合
544 * : 設定処理は行ないません。
545 * Argument : unsigned long i_count :
546 * : 送信データ数
547 * : unsigned char *i_ptr :
548 * : 送信データ
549 * Return Value : none
550 * Calling Functions :
551 *"FUNC COMMENT END"*****/
552 void scif_set_send_data(unsigned long i_count, unsigned char *i_ptr)
553 {
554     /* 設定データチェック */
555     if(i_count > sizeof(g_send_data))
556     {
557         return;
558     }
559
560     /* 送信データ領域のクリア */
561     scif_clear_senddata();
562
563     if(i_count > 0)
564     {
565         /* 送信データ数設定 */
566         g_send_count = i_count;

```

```

567
568     /* 送信データを設定 */
569     memcpy(g_send_data, i_ptr, g_send_count);
570 }
571
572 }
573
574 /*"FUNC COMMENT"*****
575 * ID          :
576 * Outline     : SCIF 送信データ数取得処理
577 * Include     :
578 * Declaration : void scif_get_send_data_num
579 *             : (unsigned long *o_pnum)
580 * Description : 送信データ格納領域より送信データ数を
581 *             : 取得します。
582 *             :
583 * Argument    : unsigned long *o_pnum :
584 *             : 送信データ数
585 * Return Value : none
586 * Calling Functions :
587 /*"FUNC COMMENT END"*****/
588 void scif_get_send_data_num(unsigned long *o_pnum)
589 {
590
591     /* パラメータチェック */
592     if(NULL == o_pnum)
593     {
594         return;
595     }
596
597     /* 送信データ数取得 */
598     *o_pnum = g_send_count;
599
600 }
601
602 /*"FUNC COMMENT"*****
603 * ID          :
604 * Outline     : SCIF 送信データ取得処理
605 * Include     :
606 * Declaration : void scif_get_send_data
607 *             : (unsigned long i_num,
608 *             : unsigned char *o_ptr)
609 * Description : 送信データを送信データ格納領域より
610 *             : 取得します。
611 *             :
612 * Limitation  : *o_ptr は scif_get_send_data_num()
613 *             : で取得した領域以上分を指定してください。
614 *             :
615 * Argument    : unsigned long i_num :
616 *             : 取得送信データ数
617 *             : unsigned char *o_ptr :
618 *             : 送信データ
619 * Return Value : none
620 * Calling Functions :
621 /*"FUNC COMMENT END"*****/
622 void scif_get_send_data(unsigned long i_num, unsigned char *o_ptr)
623 {

```

```

624     /* パラメータチェック */
625     if((NULL == o_ptr) || (i_num > sizeof(g_send_data)))
626     {
627         return;
628     }
629
630     /* 送信データを取得 */
631     memcpy(o_ptr, g_send_data, i_num);
632
633 }
634
635 /*"FUNC COMMENT"*****
636 * ID          :
637 * Outline     : SCIF1 データ受信処理
638 * Include     :
639 * Declaration : void scif_rcv_ch1(void)
640 * Description : 本関数は、レシーブインタラプトイネーブル
641 *             : の割り込み処理でコールされます。
642 *             : SCIF1 の受信時のエラー状態を判定し、
643 *             : 受信可能であるか決定します。
644 *             ; 正常データである場合には、
645 *             : データ読み出し時に、RDF、DR ビットを
646 *             : 参照し、受信処理を行います。
647 *             :
648 * Argument    : none
649 * Return Value : none
650 * Calling Functions :
651 *"FUNC COMMENT END"*****/
652 void scif_rcv_ch1(void)
653 {
654     /* 受信時エラーフラグ対応処理 */
655     if(D_SCIF_RET_OK != scif_rcv_error())
656     {
657         return;
658     }
659
660     /* ==== RDF フラグ対応処理 ====*/
661     scif_set_rcv_data_RDF_on();
662
663 }
664
665 /*"FUNC COMMENT"*****
666 * ID          :
667 * Outline     : 受信時エラーフラグ対応処理
668 * Include     :
669 * Declaration : int scif_rcv_error(void)
670 * Description : エラーフラグ(ORER)を参照し、
671 *             : エラー時の処理を行います。
672 *             : ORER(オーバーラン)エラーの時
673 *             : 受信関連レジスタをリセットし、
674 *             : 受信データ格納領域をクリアします。
675 *             :
676 * Argument    : none
677 * Return Value : int : エラー発生有無
678 * Calling Functions :
679 *"FUNC COMMENT END"*****/
680 int scif_rcv_error(void)
    
```



```

681  {
682      int      ret = D_SCIF_RET_OK;
683
684      /* ==== ORER(オーバーラン)エラーの時 ==== */
685      /* 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 */
686      if(SCIF1.SCLSR.BIT.ORER == 1)
687      {
688          /* 受信ステータスをリセット */
689          scif_reset_rcvdata_status();
690
691          /* 受信データ格納用領域をクリア */
692          scif_clear_rcvdata();
693
694          ret = D_SCIF_RET_NG;
695      }
696  }
697
698  return  ret;
699  }
700
701  /*"FUNC COMMENT"*****
702  * ID          :
703  * Outline     : RDF フラグ対応処理
704  * Include     :
705  * Declaration : void scif_set_rcv_data_RDF_on(void)
706  * Description : 受信データ数がトリガー数以上になった時の
707  *              : 受信処理を行います。
708  *              :
709  * Argument    : none
710  * Return Value : none
711  * Calling Functions :
712  *"FUNC COMMENT END"*****/
713  void scif_set_rcv_data_RDF_on(void)
714  {
715      int      i = 0;
716
717      /* ==== 受信データ数がトリガー数以上になった時 ==== */
718      if(SCIF1.SCFDR.BIT.RDF == 1)
719      {
720          /* 受信 FIFO に残っているデータを取得する */
721          scif_set_rest_data(SCIF1.SCFDR.BIT.RFDC);
722
723          /* ==== シリアルステータスレジスタ(SCFSR1)の RDF をクリアする ==== */
724          /* 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 */
725          /* 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 */
726          SCIF1.SCFDR.BIT.RDF = 0; /* RDF ビットクリア */
727      }
728  }
729
730  /*"FUNC COMMENT"*****
731  * ID          :
732  * Outline     : 受信 FIFO 残データ取得処理
733  * Include     :
734  * Declaration : void scif_set_rest_data
735  *              : (unsigned char rest_data)
736  * Description : 受信 FIFO 残データ数を指定し,
737  *              : 受信 FIFO に残っているデータを取得する。
    
```

```

738 *           :
739 * Argument   : unsigned char i_rest_data :
740 *           : 受信 FIFO 残データ数
741 * Return Value : none
742 * Calling Functions :
743 * "FUNC COMMENT END"*****/
744 void scif_set_rest_data(unsigned char i_rest_data)
745 {
746     int    i = 0;
747
748     /* 受信 FIFO に残っているデータを取得する */
749     for(i = 0; i < i_rest_data; i++)
750     {
751         /* ==== 受信 FIFO データレジスタ(SCFRDR1)から受信データを読み出す ==== */
752         scif_set_rcvdata(&SCIF1.SCFRDR);
753     }
754
755 }
756
757 /*"FUNC COMMENT"*****
758 * ID           :
759 * Outline      : 受信エラー時初期化処理
760 * Include      :
761 * Declaration  : void scif_reset_rcvdata_status(void)
762 * Description  : 受信エラー時に受信関連のビットを初期化し,
763 *              : FIFO のデータもクリアする。
764 *              :
765 * Argument     : none
766 * Return Value : none
767 * Calling Functions :
768 * "FUNC COMMENT END"*****/
769 void scif_reset_rcvdata_status(void)
770 {
771     /* 受信エラー処理(受信データの破棄) */
772     SCIF1.SCSCR.BIT.RE = 0;          /* 受信禁止 */
773     SCIF1.SCFCR.BIT.RFRST = 1;      /* 受信 FIFO リセット */
774     SCIF1.SCFSR.WORD &= ~0xff9fu; /* 各フラグビットクリア */
775     SCIF1.SCLSR.BIT.ORER = 0;       /* OREER クリア */
776     SCIF1.SCFCR.BIT.RFRST = 0;      /* 受信 FIFO リセット解除 */
777     SCIF1.SCSCR.BIT.RE = 1;         /* 受信許可 */
778
779 }
780
781 /*"FUNC COMMENT"*****
782 * ID           :
783 * Outline      : 受信データ格納処理
784 * Include      :
785 * Declaration  : void set_rcvdata(unsigned char *i_pRcv_data)
786 * Description  : 引数で指定した受信データを受信データ
787 *              : 格納領域に設定します。
788 *              :
789 * Argument     : unsigned char *i_pRcv_data : 受信データ
790 * Return Value : none
791 * Calling Functions :
792 * "FUNC COMMENT END"*****/
793 void scif_set_rcvdata(unsigned char *i_pRcv_data)
794 {
    
```

```

795     /* 受信データ格納領域に受信データを設定する */
796     if(g_rcv_count < sizeof(g_rcv_data))
797     {
798         g_rcv_data[g_rcv_count] = *i_pRcv_data;
799         g_rcv_count++;
800     }
801
802 }
803
804 /*"FUNC COMMENT"*****
805 * ID          :
806 * Outline     : 送信データ格納領域クリア処理
807 * Include     :
808 * Declaration : void scif_clear_senddata(void)
809 * Description : 送信データ格納領域をクリアする。
810 *
811 * Argument    : none
812 * Return Value : none
813 * Calling Functions :
814 *"FUNC COMMENT END"*****/
815 void scif_clear_senddata(void)
816 {
817     g_send_count = 0;
818     memset(g_send_data, 0x00, sizeof(g_send_data));
819 }
820
821 /*"FUNC COMMENT"*****
822 * ID          :
823 * Outline     : 受信データ格納領域クリア処理
824 * Include     :
825 * Declaration : void scif_clear_rcvdata(void)
826 * Description : 受信データ格納領域をクリアする。
827 *
828 * Argument    : none
829 * Return Value : none
830 * Calling Functions :
831 *"FUNC COMMENT END"*****/
832 void scif_clear_rcvdata(void)
833 {
834     g_rcv_count = 0;
835     memset(g_rcv_data, 0x00, sizeof(g_rcv_data));
836 }
837
838 /*"FUNC COMMENT"*****
839 * ID          :
840 * Outline     : 格納受信データ取得処理
841 * Include     :
842 * Declaration : void scif_get_rcvdata
843 *              : (unsigned long *o_pRcvdata_num,
844 *              : unsigned char *o_pRcvdata)
845 * Description : 格納受信データを取得する。
846 *
847 * Argument    : unsigned long *o_pRcvdata_num :
848 *              : 格納受信データ数
849 *              : unsigned char *o_pRcvdata :
850 *              : 格納受信データ
851 * Return Value : none
    
```

```

852  * Calling Functions      :
853  * "FUNC COMMENT END"*****/
854  void scif_get_rcvdata(unsigned long *o_pRcvdata_num, unsigned char *o_pRcvdata)
855  {
856      if((o_pRcvdata_num == NULL) || (o_pRcvdata == NULL))
857      {
858          return;
859      }
860
861      *o_pRcvdata_num = g_rcv_count;
862
863      memcpy(o_pRcvdata, g_rcv_data, g_rcv_count);
864
865      return;
866  }
867
868  /*"FUNC COMMENT"*****
869  * ID                      :
870  * Outline                  : SCFCR.TTRG 設定値取得処理
871  * Include                  :
872  * Declaration              : unsigned char scif_get_snd_trigger_num
873  *                          : (T_SCIF_send_fifo_trigger i_trigger_num)
874  * Description              : 引数で指定したデータ数トリガに対応する
875  *                          : SCFCR.TTRG 設定値を取得する。
876  *                          :
877  * Argument                 : T_SCIF_send_fifo_trigger i_trigger_num
878  *                          : データ数トリガ
879  * Return Value             : unsigned char :
880  *                          : レジスタ設定値(SCFCR.TTRG に設定)
881  * Calling Functions        :
882  * "FUNC COMMENT END"*****/
883  unsigned char scif_get_snd_trigger_num(T_SCIF_send_fifo_trigger i_trigger_num)
884  {
885      unsigned char  ret = 0;
886      int            i = 0;
887      int            count = 0;
888
889      count = sizeof(gc_scif_ttrg_tbl) / sizeof(T_SCIF_SCFCR_TTRG_SET);
890
891      for(i = 0; i < count; i++)
892      {
893          if(gc_scif_ttrg_tbl[i].mtrigger_num == i_trigger_num)
894          {
895              /* ==== SCFCR.TTRG 設定 ==== */
896              ret = gc_scif_ttrg_tbl[i].mttrg;
897              break;
898          }
899      }
900
901      return ret;
902  }
903
904  /*"FUNC COMMENT"*****
905  * ID                      :
906  * Outline                  : SCFCR.RTRG 設定値取得処理
907  * Include                  :
908  * Declaration              : unsigned char scif_get_rcv_trigger_num

```

```

909 *          : (T_SCIF_rcv_fifo_trigger i_trigger_num)
910 * Description      : 引数で指定したデータ数トリガに対応する
911 *          : SCFCR.RTRG 設定値を取得する。
912 *          :
913 * Argument        : T_SCIF_rcv_fifo_trigger i_trigger_num :
914 *          : データ数トリガ
915 * Return Value    : unsigned char :
916 *          : レジスタ設定値(SCFCR.RTRG に設定)
917 * Calling Functions :
918 * "FUNC COMMENT END"*****/
919 unsigned char scif_get_rcv_trigger_num(T_SCIF_rcv_fifo_trigger i_trigger_num)
920 {
921     unsigned char  ret = 0;
922     int            i = 0;
923     int            count = 0;
924
925     count = sizeof(gc_scif_rtrg_tbl) / sizeof(T_SCIF_SCFCR_RTRG_SET);
926
927     for(i = 0; i < count; i++)
928     {
929         if(gc_scif_rtrg_tbl[i].mtrigger_num == i_trigger_num)
930         {
931             /* ==== SCFCR.RTRG 設定 ==== */
932             ret = gc_scif_rtrg_tbl[i].mrtrg;
933             break;
934         }
935     }
936
937     return ret;
938 }
939
940
941 /*"FUNC COMMENT"*****
942 * ID          :
943 * Outline     : 1 ビット期間経過待ち処理
944 * Include     :
945 * Declaration : void scif_wait_lbit(
946 *          : unsigned long i_bit_rate)
947 *          :
948 * Description : 指定したビットレートに対応した
949 *          : 1 ビット期間経過後に処理を抜けます。
950 *          :
951 *          : gc_scif_cks_scbrr_tbl テーブルに設定され
952 *          : ている cpu 実行回数分 nop 処理を行い、
953 *          : 1 ビット期間の経過を待ちます。
954 *          :
955 * Limitation  : I を変更した場合は、
956 *          : gc_scif_cks_scbrr_tbl テーブルの設定値
957 *          : も修正ください。
958 *          :
959 * Argument    : long i_bit_rate : 動作ビットレート
960 *          :
961 * Return Value : none
962 * Calling Functions :
963 * "FUNC COMMENT END"*****/
964 static void scif_wait_lbit(unsigned long i_bit_rate)
965 {
    
```

```

966     unsigned long   cpu_count;
967     unsigned long   i;
968
969     /* 1ビット期間待ち分のcpu実行回数を取得 */
970     cpu_count = gc_scif_cks_scbrr_tbl[i_bit_rate].cpu_count_1bit;
971
972     for(i = 0; i < cpu_count; i++)
973     {
974         nop();
975     }
976
977     return;
978 }
979
980 /*"FUNC COMMENT"*****
981 * ID           :
982 * Outline      : 送受信完了 Wait 処理
983 * Include      :
984 * Declaration  : scif_send_rcv_end_wait()
985 * Description  : 本参考プログラムでは、送受信が
986 *              : 完了するまでこの処理で待ちます。
987 *              : 全データを確実に送受信するまでの待ち時間
988 *              : を D_SCIF_RCV_TIME_WAIT に設定してください。
989 * Description  : Wait 時間経過後に
990 *              : 受信 FIFO に残っているデータを取得します。
991 *              :
992 * Argument     : none
993 * Return Value : none
994 * Calling Functions :
995 *"FUNC COMMENT END"*****/
996 void scif_send_rcv_end_wait(void)
997 {
998     unsigned long   alldata_rcv_time = D_SCIF_RCV_TIME_WAIT;
999
1000     while(alldata_rcv_time > 0)
1001     {
1002         --alldata_rcv_time;
1003     }
1004
1005     /* 受信 FIFO に残っているデータを取得する */
1006     scif_set_rest_data(SCIF1.SCFDR.BIT.RFDC);
1007
1008 }
1009
1010 /* 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) 2009. 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 クロック同期式 単方向送受信設定例
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 * : クロック同期式単方向通信サンプルプログラム
42 * Operation :
43 * Disclaimer :
44 * :
45 * Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
46 *
47 *****/
48 * History : 27.Feb.2009 Ver. 1.00 First Release
49 *"FILE COMMENT END"*****
50
51 #ifndef __SCIF_DEF_H__
52 #define __SCIF_DEF_H__
53

```

```

54  /* ==== マクロ定義 ==== */
55
56  /* 戻り値 */
57  #define D_SCIF_RET_NG    -1
58  #define D_SCIF_RET_OK    0
59
60  /* FIFO 設定可能数 */
61  #define D_SCIF_FIFO_NUM    16
62
63  /* 送信・受信データ格納領域サイズ */
64  #define D_SCIF_DATA_SIZE  50
65
66  /* 送受信完了待ち時間設定値 */
67  /* 本参考プログラムの送受信が確実に完了する時間を設定してください */
68  /* この時間経過後、送信と受信のデータ比較を行います。 */
69  #define D_SCIF_RCV_TIME_WAIT    10000000
70
71  /* ==== FIFO データ数トリガ ==== */
72  /* 動作させたい FIFO のトリガ数を以下の定義している */
73  /* デファインから設定してください */
74  #define D_SCIF_DATA_NUM_SND_TRIGGER_CH0    D_SCIF_SEND_FIFO_TRIGGER8
75  /* 設定可能値 0,2,4,8 */
76
77  #define D_SCIF_DATA_NUM_RCV_TRIGGER_CH1    D_SCIF_RCV_FIFO_TRIGGER2
78  /* 設定可能値 1,2,8,14 */
79
80  /* 送信 FIFO データ数トリガ */
81  typedef enum
82  {
83      D_SCIF_SEND_FIFO_TRIGGER0 = 0,    /* 送信 FIFO データトリガ数 0 */
84      D_SCIF_SEND_FIFO_TRIGGER2 = 2,    /* 送信 FIFO データトリガ数 2 */
85      D_SCIF_SEND_FIFO_TRIGGER4 = 4,    /* 送信 FIFO データトリガ数 4 */
86      D_SCIF_SEND_FIFO_TRIGGER8 = 8     /* 送信 FIFO データトリガ数 8 */
87  } T_SCIF_send_fifo_trigger;
88
89  /* 受信 FIFO データ数トリガ */
90  typedef enum
91  {
92      D_SCIF_RCV_FIFO_TRIGGER1 = 1,    /* 受信 FIFO データトリガ数 1 */
93      D_SCIF_RCV_FIFO_TRIGGER2 = 2,    /* 受信 FIFO データトリガ数 2 */
94      D_SCIF_RCV_FIFO_TRIGGER8 = 8,    /* 受信 FIFO データトリガ数 8 */
95      D_SCIF_RCV_FIFO_TRIGGER14 = 14   /* 受信 FIFO データトリガ数 14 */
96  } T_SCIF_rcv_fifo_trigger;
97
98  /* ==== ビットレート設定タイプ ==== */
99  typedef enum
100 {
101     D_SCIF_SCBRR_1K,
102     D_SCIF_SCBRR_2_5K,
103     D_SCIF_SCBRR_5K,
104     D_SCIF_SCBRR_10K,
105     D_SCIF_SCBRR_25K,
106     D_SCIF_SCBRR_50K,
107     D_SCIF_SCBRR_100K,
108     D_SCIF_SCBRR_250K,
109     D_SCIF_SCBRR_500K
110 } T_SCIF_bit_rate_type;
    
```



```

111
112 /* ==== 構造体定義 ==== */
113 /* SCBRR,SCSMR.CKS,lbitwait 期間 設定情報 */
114 typedef struct {
115     unsigned char  mscbrr;
116     unsigned char  mscsmr_cks;
117     unsigned long  cpu_count_lbit;
118 } T_SCIF_CKS_SCBRR_SET_INFO;
119
120 /* SCFCR.TTRG 設定用 */
121 typedef struct {
122     unsigned char  mtrigger_num;
123     unsigned char  mttrg;
124 } T_SCIF_SCFCR_TTRG_SET;
125
126 /* SCFCR.RTRG 設定用 */
127 typedef struct {
128     unsigned char  mtrigger_num;
129     unsigned char  mrtrg;
130 } T_SCIF_SCFCR_RTRG_SET;
131
132 /* ==== 関数宣言 ==== */
133 void scif_init_SCIF0(void);
134 void scif_init_SCIF1(void);
135 void scif0_send(unsigned long i_count, unsigned char *i_ptr);
136 void scif0_setdata(void);
137 void scif_set_send_data(unsigned long i_count, unsigned char *i_ptr);
138 void scif_get_send_data_num(unsigned long *o_pnum);
139 void scif_get_send_data(unsigned long i_num, unsigned char *o_ptr);
140 void scif_rcv_ch1(void);
141 void scif0_setdata_chk_end(void);
142 void scif_clear_senddata(void);
143 void scif_clear_rcvdata(void);
144 void scif_set_rest_data(unsigned char i_rest_data);
145 void scif_get_rcvdata(unsigned long *o_pRcvdata_num, unsigned char *o_pRcvdata);
146 void scif_rcv_alldata_wait(void);
147
148 #endif /* __SCIF_DEF_H__ */
    
```

## (4) サンプルプログラムリスト "intprg.c"

SCIF0, 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.SCFSR.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 SCIFI1 */
41 void INT_SCIF_SCIFI1(void)
42 {
43     unsigned short dummy;
44
45     /* SCIF1 受信処理 */
46     scif_rcv_ch1();
47
48     /* 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 */
49     dummy = SCIF1.SCFSR.WORD;
50     int_responstime_wait(INTC_RESPONSEWAIT); /* 優先順位判定時間分待ち */
51
52 }
53
54 ...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 SCIFIO
5          .data.b   H'10
6          ;H'C20    SCIF SCIF11
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 16 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。

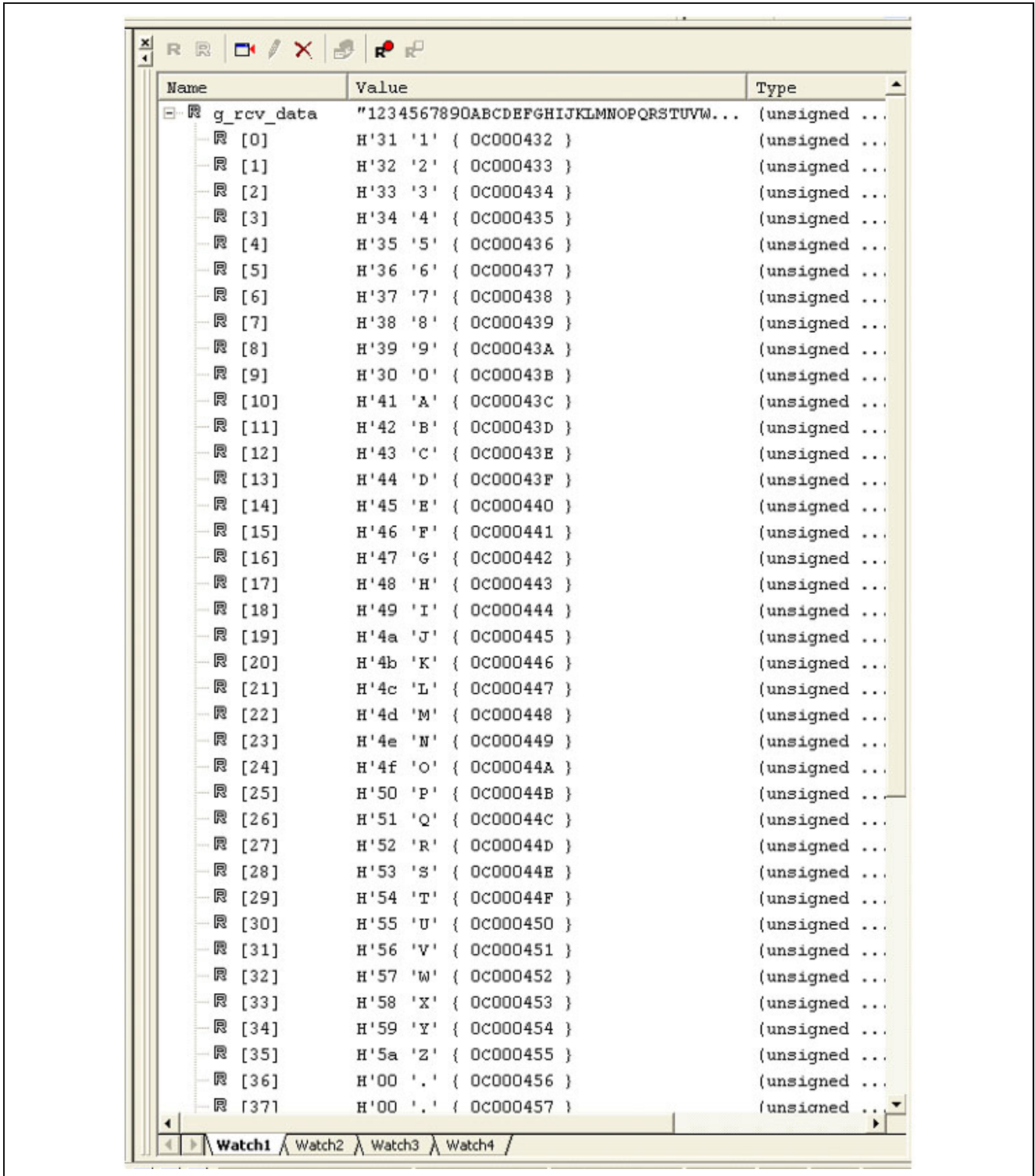


図 16 SCIF1 受信データ格納領域 (g\_rcv\_data[D\_SCIF\_DATA\_SIZE])

## 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