

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い申し上げます。

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が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。

SH7780 グループ

SCIF 調歩同期式モード使用例 (シリアルデータ送受信)

要旨

この資料は、SH7780 グループの SCIF 調歩同期式モード使用の設定方法例、および応用例について説明しています。

動作確認デバイス

SH7780 (日立超 LSI システムズ製 Solution Engine MS7780SE03)

目次

1. はじめに	2
2. 応用例の説明	4
3. 参考プログラムリスト	12
4. 参考ドキュメント	23

1. はじめに

1.1 仕様

SCIF チャンネル 0 を調歩同期式モードで使用し、送受信を行ないます。接続先デバイスより受信したデータを、受信完了後に送り返します。

下記に通信の概要と、図 1 に接続例を示します。

- ビットレート 115200 bps
- データ長 8 ビット、パリティビット付加、1 ストップビット

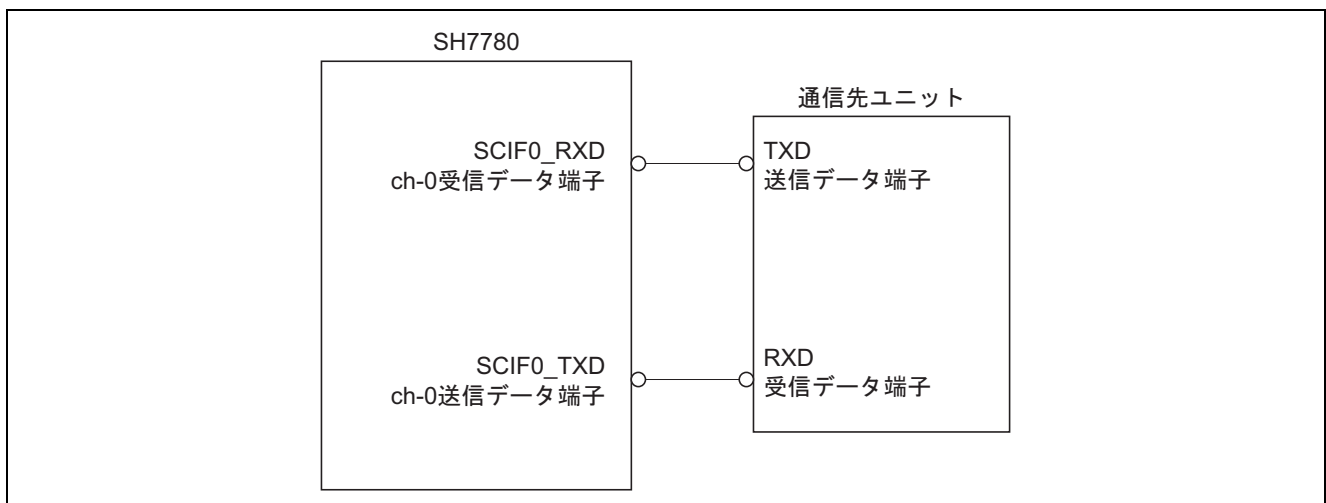


図 1 SCIF (調歩同期式モード) 動作接続例

1.2 使用機能

- SCIF チャンネル 0

1.3 適用条件

- マイコン: SH7780
- 動作周波数:
 - 内部クロック: 400 MHz
 - SuperHyway クロック: 200 MHz
 - 周辺クロック: 33 MHz
 - DDR クロック: 160 MHz
 - 外部バスクロック: 33 MHz
 - PCI バスクロック: 33 MHz
- クロック動作モード: モード 3 (MODE7 = LOW, MODE2 = LOW, MODE1 = High, MODE0 = High)
- データアライメント リトルエンディアン
- アドレスモード 29 ビットアドレスモード
- C コンパイラ: ルネサス テクノロジ製
SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.1.0

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、アプリケーションノート「SH7780 初期設定例」(RJJ06B0860-0100) の設定条件で動作確認をしています。そちらも合わせて参照してください。

2. 応用例の説明

本プログラム例では、SCIF チャンネル 0 を調歩同期式モードで使用して、データ送信とデータ受信を行いません。受信したデータの内容をそのまま送信データとして送り返します。受信処理は、受信データフル割り込みを使って割り込み処理の中で受信データ読み出しを行ない、また送信処理は割り込みを使わずに受信データフル割り込みの中で行ないます。

2.1 参考プログラムの説明

本プログラム例は

- (1) scif.c
- (2) main.c
- (3) intprg.c
- (4) vecttbl.src

の 4 つのソースプログラムから構成されます。

- (1) scif.c は、本プログラムで使用する SCIF の動作設定関数と初期設定例に含まれる intprg.c の割り込み関数よりジャンプする受信データフル割り込み関数、受信エラー割り込み関数、ブ레이크割り込み関数を記述しています。本プログラム例のベースとなる初期設定例には含まれていません。
- (2) main.c では、ステータスレジスタ (SR) の設定と SCIF の動作設定関数コールを行なっています。初期設定例に含まれる main.c を本プログラム例に合わせて変更してください。
- (3) intprg.c では例外/割り込みハンドラよりコールされる割り込みプログラムを記述しています。初期設定例に含まれる intprg.c を本プログラム例に合わせて変更してください。
- (4) vecttbl.src ではリセットベクタテーブル、例外/割り込みベクタテーブル、割り込みマスクテーブルを記述しています。受け付けた割り込みを繰り返し受け付けないように、ステータスレジスタの IMASK ビットに設定する割り込みマスクレベルを割り込みマスクテーブルに記述する必要があります。初期設定例に含まれる vecttbl.src を本プログラム例に合わせて変更してください。

2.2 使用機能の動作概要

SCIF はキャラクタ単位で同期を取りながら通信する調歩同期式モードと、クロックパルスにより同期を取りながら通信するクロック同期式モードの 2 方式でシリアル通信ができます。

送受信おののちに 64 段の FIFO バッファを内蔵しており、効率のよい高速連続通信が可能です。また、チャネル 0 のみモデムコントロール機能 (RTS, CTS) を内蔵しています。

送信フォーマットの選択はシリアルモードレジスタ (SCSMR) で行ないます。また、SCIF のクロックソースは、SCSMR の C/A ビットおよび、シリアルコントロールレジスタ (SCSCR) の CK1, 0 ビットの組み合わせで決まります。

(1) 調歩同期式モード

- データ長: 7 ビット/8 ビットから選択
- データの送受信は LSB が先頭
- パリティビットの付加および、1 ビット/2 ビットのストップビット付加を選択可能
- 受信エラーの検出: フレーミングエラー、パリティエラー、オーバランエラーを検出
- ブレーク信号の検出: フレーミングエラーが発生し、引き続き 1 フレーム長以上スペース 0 (ローレベル) の場合、ブレークを検出
- ブレーク信号の送り出し: SCIF_TXD 端子は SCSPTR の SPB2IO, SPB2DT ビットで入出力条件とレベルを決めることができます。これを使ってブレーク信号の送り出しができます。
シリアル送信時にブレークを送り出したいときは SPB2DT ビットを 0 (ローレベル) にクリアした後、TE ビットを 0 (送信停止) します。TE ビットを 0 にクリアすると現在の送信状態とは無関係に送信部は初期化され、SCIF_TXD 端子からは 0 が出力されます。
- 送受信 FIFO レジスタのおのの格納データ数を表示 (SCTFDR, SCRFDR)
- SCIF のクロックソース: 周辺クロック/SCIF_SCK 端子入力からの選択可能
周辺クロック (Pck) を選択した場合: SCIF はボーレートジェネレータのクロックで動作
SCIF_SCK 端子入力を選択した場合: ビットレートの周波数を 16 倍した周波数でクロックを入力する必要あり

(2) クロック同期式モード

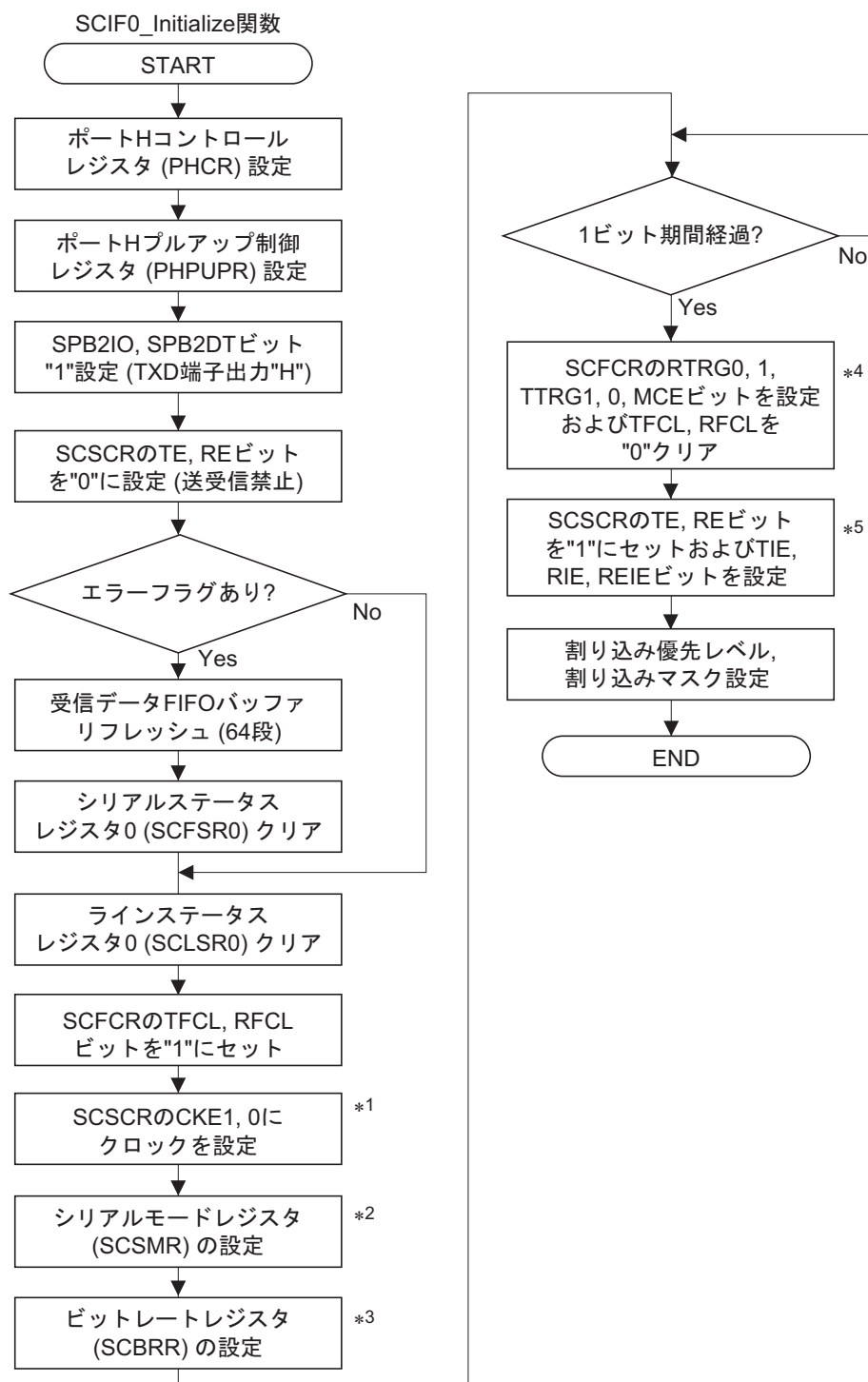
- データ長: 8 ビット
- データの送受信は LSB が先頭
- 受信時にオーバランエラー検出可能
- SCIF のクロックソース: 周辺クロック/SCIF_SCK 端子入力からの選択可能
周辺クロック (Pck) を選択した場合: SCIF はボーレートジェネレータのクロックで動作
SCIF_SCK 端子入力を選択した場合: ビットレートの周波数を 16 倍した周波数でクロックを入力する必要あり

2.3 使用機能の設定手順

ここでは、SCIF チャンネル 0 調歩同期式モード動作の設定手順について説明します。

本プログラムでは、初期設定例のプログラムをベースとすることを前提にメイン関数の始めに本設定を行っています。そのため特権モードで動作していることを前提にしていますので、他のプログラムに流用されるなどの場合には処理モードにご注意の上利用してください。

次ページの図 2 に SCIF チャンネル 0 調歩同期式モード動作の設定のフロー例を示します。各レジスタ設定の詳細は、「SH7780 グループ ハードウェアマニュアル」を参照してください。



- 【注】 *1 SCIFのクロックソース (内部もしくは外部) およびSCIF_SCK端子からのクロック出力について許可/禁止を設定します。今回は内部クロックとして使用し、SCIF_SCK端子はポートとして使用するため、"00を設定してください。
- *2 シリアル通信フォーマットとボーレートジェネレータに入力するクロックソースを設定してください。
- *3 SCBRRにビットレートが目的の値と等しくなるように値を設定してください。
- *4 未送信データ数トリガをD'00, 受信データ数トリガをD'16に設定します。
- *5 受信割り込み, 受信エラー割り込みを使用するため、RIE, REIEビットを割り込み許可とします。またRE, TEビットでシリアル送信動作, 受信動作を許可してください。送信FIFOデータエンプティ割り込みは今回使用しないため、割り込み禁止とします。

図 2 SCIF 調歩同期式モード設定フロー例

2.4 参考プログラムの処理手順

表 1 に SCIF 関連レジスタの設定例を示します。また、図 3~図 6 に参考プログラムのメイン関数、割り込み処理のフロー例を示します。

表 1 SCIF の設定

レジスタ名	アドレス	設定値	機能
割り込み優先順位設定レジスタ 2 (INT2PRI2)	H'FFD40008	H'1F000000	SCIF-ch0 割り込み優先レベル: 31
割り込みマスククリアレジスタ (INT2MSKCR)	H'FFD4003C	H'00000008	SCIF-ch0 割り込みマスククリア
ポート H コントロールレジスタ (PHCR)	H'FFEA000E	H'FC80	PH3~0 SCIF0 モジュール機能
ポート H ブルアップ制御レジスタ (PHPUPR)	H'FFEA004E	H'0000	PH0~PH7 ブルアップオフ
シリアルモードレジスタ (SCSMR)	H'FFD80004	H'0020	調歩同期式モード 8 ビットデータ パリティ許可 (偶数パリティ) 1 ストップビット クロックソース: Pck
シリアルコントロールレジスタ (SCSCR)	H'FFD80008	H'0078	内部クロック動作 RXI, ERI, BRI 割り込みイネーブル SCK 端子はポートとして使用
ビットレートレジスタ (SCBBR)	H'FFD8000C	H'08	
FIFO コントロールレジスタ (SCFCR)	H'FFD80010	H'0070	RDF フラグトリガ数: 16

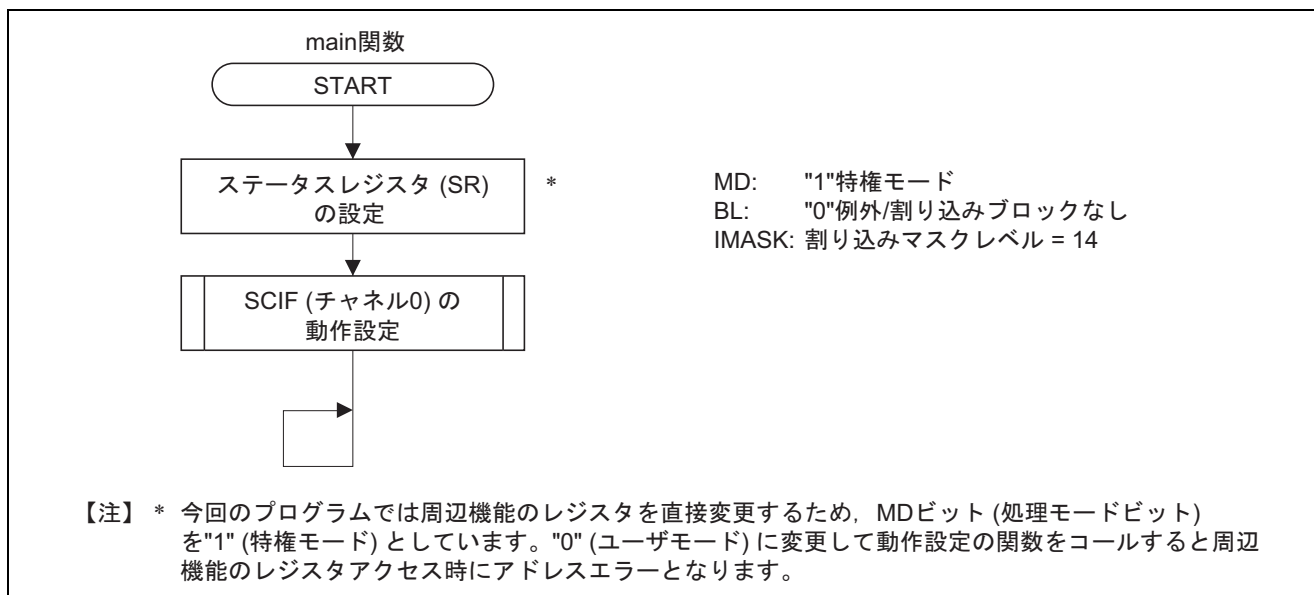
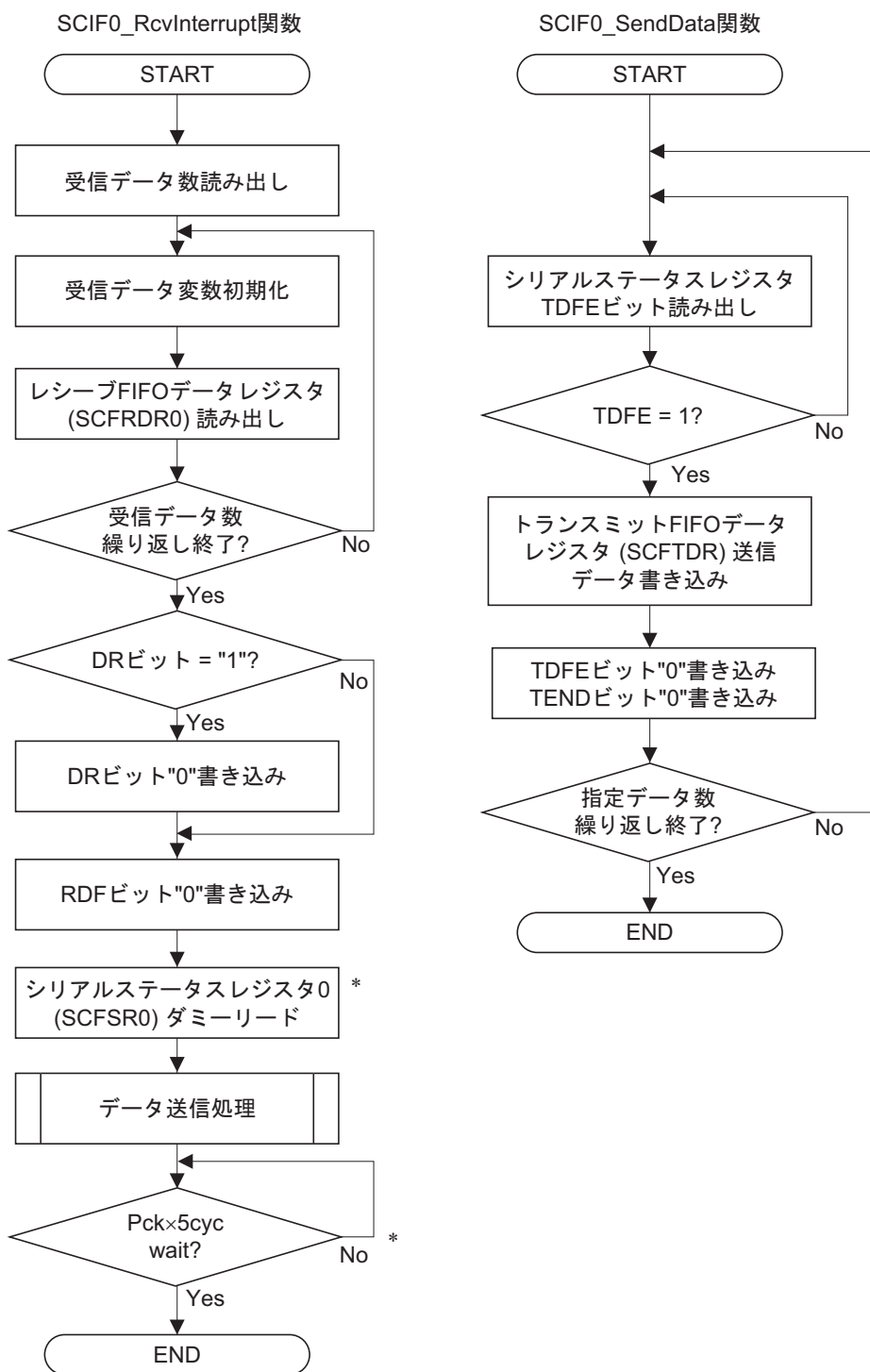
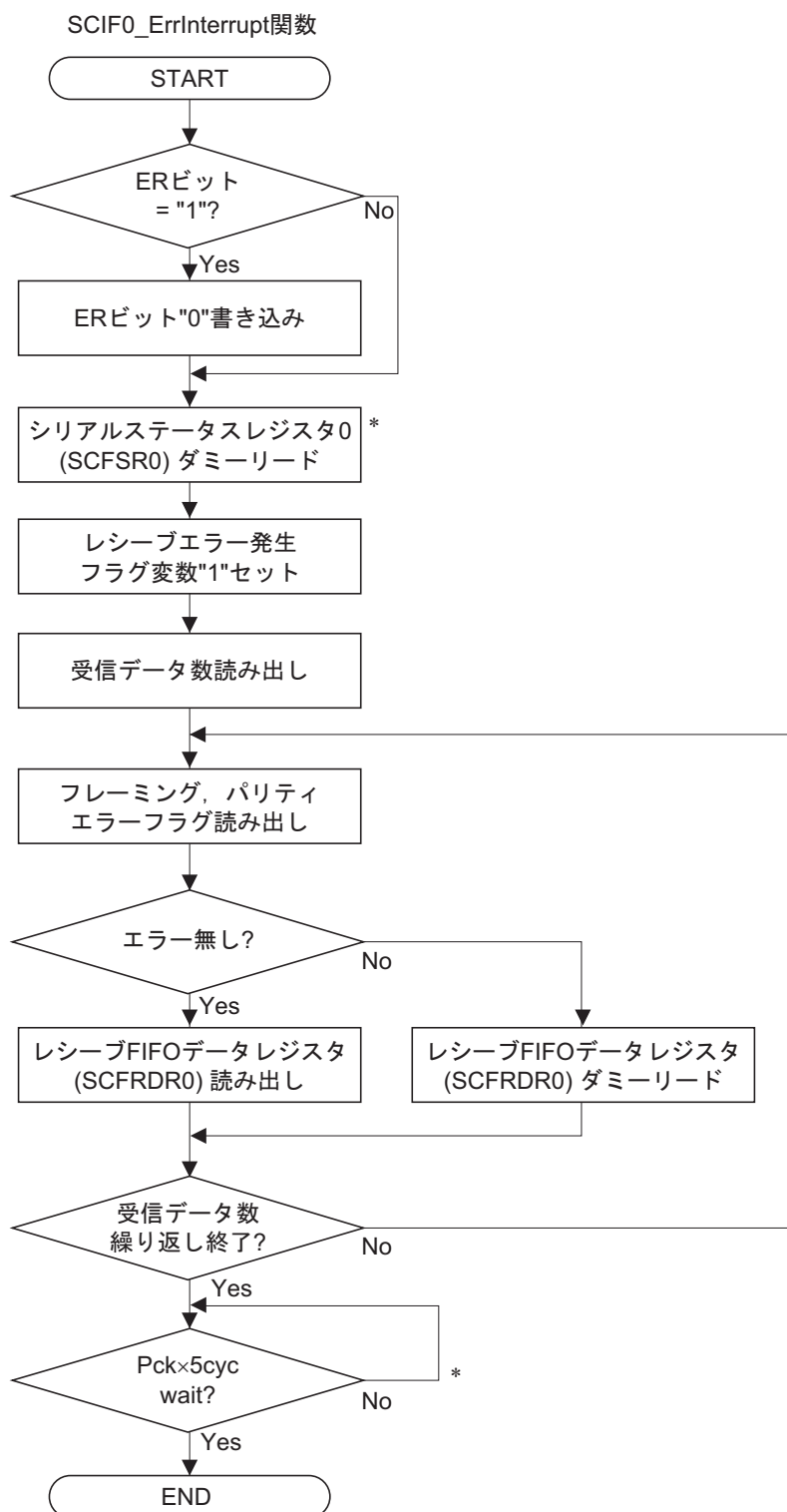


図 3 参考プログラム (調歩同期式モードシリアルデータ送信) メイン関数の処理フロー例



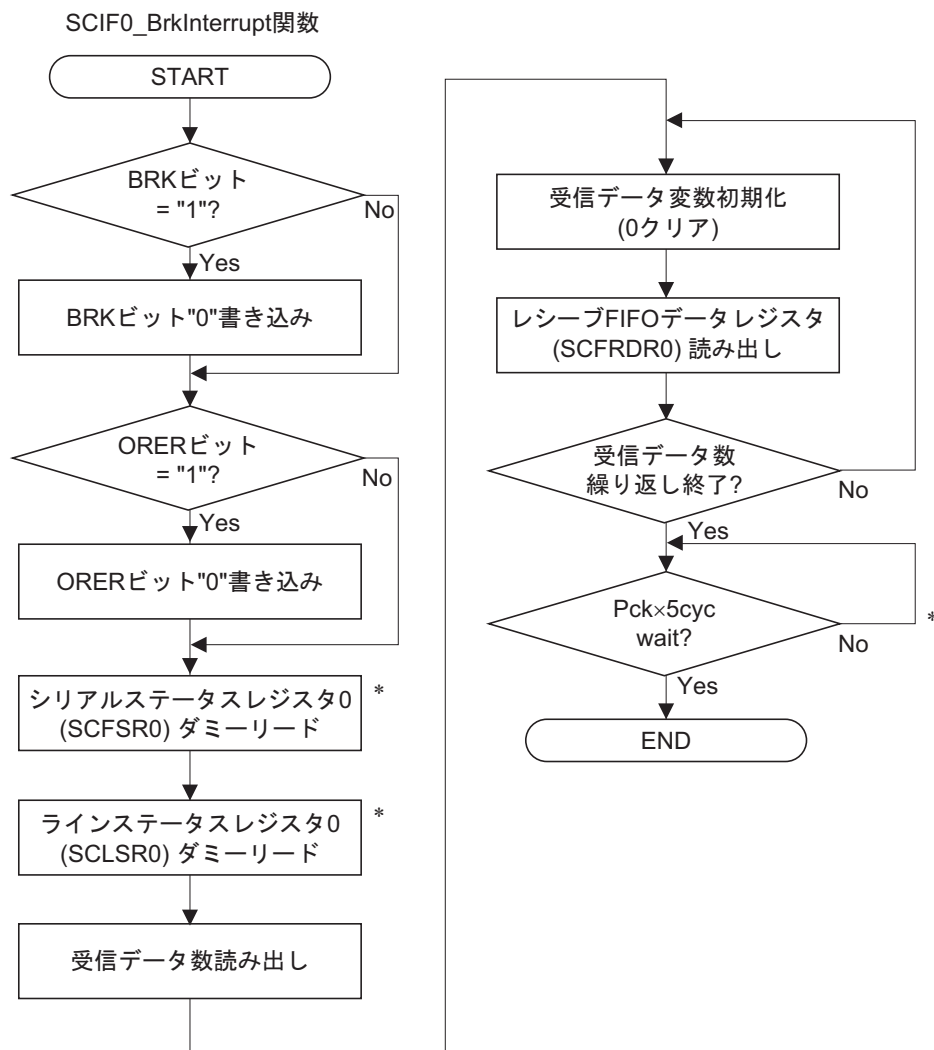
【注】 * 更新したはずの割り込み要因による割り込みの受け付けを避けるために、いったん該当フラグを含む内蔵モジュールのレジスタを読み出し、優先順位判定時間 (Pck×5cyc) 待ってからBLビットを"0"にしています。(ただし、今回のプログラムでは割り込み要因フラグのクリアを行なってから、データ送信処理を行なっているため、優先順位判定時間以上の時間が経過しており、コメントアウトすることも可能です。)

図 4 RXI (受信データフル) 割り込み処理，データ送信処理フロー例



【注】 * 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ために、いったん該当フラグを含む内蔵モジュールのレジスタを読み出し、優先順位判定時間 (Pck×5cyc) 待ってからBLビットを"0"にしています。(ただし、今回のプログラムでは割り込み要因フラグのクリアを行ってから、データ送信処理を行なっているため、優先順位判定時間以上が経過しており、コメントアウトすることも可能です。)

図5 ERI (受信エラー) 割り込み処理フロー例



【注】 * 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ために、いったん該当フラグを含む内蔵モジュールのレジスタを読み出し、優先順位判定時間 (Pck×5cyc) 待ってからBRビットを"0"にしています。(ただし、今回のプログラムでは割り込み要因フラグのクリアを行なってから、データ送信処理を行なっているため、優先順位判定時間以上の時間が経過しており、コメントアウトすることも可能です。)

図 6 BRK (ブレーク) 割り込み処理フロー例

3. 参考プログラムリスト

1. サンプルプログラムリスト"main.c"

```

1.  /*""FILE COMMENT""*****
2.      *      System Name:   SH7780 Sample Program
3.      *      File Name    :   main.c
4.      *      Version     :   1.00.00
5.      *      Contents    :   SH7780 Initialize Program
6.      *      Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7.      *      CPU         :   SH7780
8.      *      Compiler    :   SHC.9.1.00
9.      *      OS          :   none
10.     *
11.     note           :   < Caution >
12.                     This sample program is provided simply as a reference and
13.                     its operation is not guaranteed.
14.                     Use this sample program as a technical reference when
15.                     developing software.
16.     *
17.     Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
18.     *
19.     History        :   2007/12/26 ver 1.00.00
20.     *
21.     *****/
22. #include <machine.h>
23.
24. /* --- Function Definition(internal) --- */
25.
26. /* --- Symbol Definition --- */
27. #define SR_Init      0x400000e0      /* Privileged mode, RB,BL=0, IMASK level 14 */
28.
29. /***** Function Comment *****/
30. * Outline           : main
31. *-----
32. * Declaration       : void main(void)
33. *-----
34. * Functional description:
35. *         main function
36. *-----
37. * Return Value      : -
38. * Argument          : -
39. *-----
40. * Input             : -
41. * Output            : -
42. *-----
43. * Notes              : -
44. *****/ Function Comment End *****/
45. void main(void)
46. {
47.     set_cr(SR_Init);      /* Set SR "Privileged mode, RB,BL=0, IMASK level 14" */
48.
49.     SCIF0_Initialize();    /* SCIF0 Initialize (additional part from Initialize program)*/
50.
51.     while(1)
52.     {
53.     }
54. }
```

2. サンプルプログラムリスト"scif.c" (1)

```

1  /*"FILE COMMENT"*****
2  *      System Name:   SH7780 Sample Program
3  *      File Name    :   scif.c
4  *      Version      :   1.00.00
5  *      Contents     :   SH7780 SCIF0 transmit Program
6  *      Model        :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7  *      CPU          :   SH7780
8  *      Compiler     :   SHC.9.1.00
9  *      OS           :   none
10 *
11 *      note          :   < Caution >
12 *                      This sample program is provided simply as a reference and
13 *                      its operation is not guaranteed.
14 *                      Use this sample program as a technical reference when
15 *                      developing software.
16 *
17 *      Copyright (C) 2007 Renesas Technology Corp. All Rights Reserved
18 *
19 *      History       :   2007/12/26 ver 1.00.00
20 *
21 *****/
22 #include <machine.h>
23
24 /* --- Function Definition(internal) --- */
25 void SCIF0_Initialize(void);
26 void SCIF0_SendData(void);
27 void SCIF0_RcvInterrupt(void);
28 void SCIF0_ErrInterrupt(void);
29 void SCIF0_BrkInterrupt(void);
30 static void int_responstime_wait(unsigned int wait_time);
31
32 /* --- Symbol Definition --- */
33 union st_scsmr{                                /* struct SCSMR*/
34     unsigned short    WORD;                    /* Word Access */
35     struct{
36         unsigned short    :8;                /*          */
37         unsigned short CA    :1;            /*      CA      */
38         unsigned short CHR   :1;            /*      CHR     */
39         unsigned short PE    :1;            /*      PE      */
40         unsigned short OE    :1;            /*      OE      */
41         unsigned short STOP  :1;            /*      STOP    */
42         unsigned short      :1;            /*          */
43         unsigned short CKS   :2;            /*      CKS     */
44     }BIT;
45 };
46
47 union st_scscr{                                /* struct SCSCR*/
48     unsigned short    WORD;                    /* Word Access */
49     struct{
50         unsigned short    :8;                /*          */
51         unsigned short TIE   :1;            /*      TIE     */
52         unsigned short RIE   :1;            /*      RIE     */
53         unsigned short TE    :1;            /*      TE      */
54         unsigned short RE    :1;            /*      RE      */
55         unsigned short REIE  :1;            /*      REIE    */
56         unsigned short      :1;            /*          */

```

3. サンプルプログラムリスト"scif.c" (2)

```

57     unsigned short CKE           :2;    /*    CKE    */
58     }BIT;
59 };
60
61 union st_scfsr{                  /* struct SCFSR*/
62     unsigned short    WORD;      /* Word Access */
63     struct{              /* Bit Access */
64         unsigned short      :8;  /*          */
65         unsigned short ER    :1;  /*    ER    */
66         unsigned short TEND  :1;  /*    TEND  */
67         unsigned short TDFE  :1;  /*    TDFE  */
68         unsigned short BRK   :1;  /*    BRK   */
69         unsigned short FER   :1;  /*    FER   */
70         unsigned short PER   :1;  /*    PER   */
71         unsigned short RDF   :1;  /*    RDF   */
72         unsigned short DR    :1;  /*    DR    */
73     }BIT;
74 };
75
76 union st_scfc{                  /* struct SCFCR*/
77     unsigned short    WORD;      /* Word Access */
78     struct{              /* Bit Access */
79         unsigned short      :5;  /*          */
80         unsigned short RSTRG  :3;  /*    RSTRG  */
81         unsigned short RTRG   :2;  /*    RTRG   */
82         unsigned short TTRG   :2;  /*    TTRG   */
83         unsigned short MCE    :1;  /*    MCE    */
84         unsigned short TFCL   :1;  /*    TFCL   */
85         unsigned short RFCL   :1;  /*    RFCL   */
86         unsigned short LOOP   :1;  /*    LOOP   */
87     }BIT;
88 };
89
90 union st_scsptr{                /* struct SCSPTR */
91     unsigned short    WORD;      /* Word Access */
92     struct{              /* Bit Access */
93         unsigned short      :8;  /*          */
94         unsigned short RTSIO  :1;  /*    RTSIO  */
95         unsigned short RTSDT  :1;  /*    RTSDT  */
96         unsigned short CTSIO  :1;  /*    CTSIO  */
97         unsigned short CTSDT  :1;  /*    CTSDT  */
98         unsigned short SCKIO  :1;  /*    SCKIO  */
99         unsigned short SCKDT  :1;  /*    SCKDT  */
100        unsigned short SPB2IO  :1;  /*    SPB2IO  */
101        unsigned short SPB2DT  :1;  /*    SPB2DT  */
102    }BIT;
103 };
104
105 union st_sclsr{                 /* struct SCLSR*/
106     unsigned short    WORD;      /* Word Access */
107     struct{              /* Bit Access */
108         unsigned short      :15;  /*          */
109         unsigned short ORER    :1;  /*    ORER   */
110     }BIT;
111 };
112

```

4. サンプルプログラムリスト"scif.c" (3)

```

1  union st_screr{                                /* struct SCRER*/
2      unsigned short    WORD;                    /* Word Access */
3      struct{
4          unsigned short :2; /* */
5          unsigned short PER :6; /* PER */
6          unsigned short :2; /* */
7          unsigned short FER :6; /* FER */
8      }BIT;
9  };
10
11 #define SCSMR0    (*(volatile union st_scsmr    *)0xFFE00000) /* SCSMR0 Address */
12 #define SCBRR0    (*(volatile unsigned char     *)0xFFE00004) /* SCBRR0 Address */
13 #define SCSCR0    (*(volatile union st_scscr    *)0xFFE00008) /* SCSCR0 Address */
14 #define SCFTDR0    (*(volatile unsigned char     *)0xFFE0000C) /* SCFTDR0 Address */
15 #define SCFSR0    (*(volatile union st_scfsr    *)0xFFE00010) /* SCFSR0 Address */
16 #define SCFRDR0    (*(volatile unsigned char     *)0xFFE00014) /* SCFRDR0 Address */
17 #define SCFCR0    (*(volatile union st_scfcr    *)0xFFE00018) /* SCFCR0 Address */
18 #define SCRFDR0    (*(volatile unsigned short    *)0xFFE00020) /* SCRFDR0 Address */
19 #define SCLSR0    (*(volatile union st_sclsr    *)0xFFE00028) /* SCLSR0 Address */
20 #define SCSPTR0    (*(volatile union st_scsptr   *)0xFFE00024) /* SCSPTR0 Address */
21
22 #define PHCR      (*(volatile unsigned short    *)0xFFEA000E) /* PHCR Address */
23 #define PHPUPR    (*(volatile unsigned char     *)0xFFEA004E) /* PHPUPR Address */
24 #define INT2PRI2  (*(volatile unsigned int      *)0xFFD40008) /* INT2PRI2 Address */
25 #define INT2MSKCR (*(volatile unsigned int      *)0xFFD4003C) /* INT2MSKCR Address */
26
27 #define NUM_RCV_DATABUF    (0x40)
28
29 #define INTC_RESPONSEWAIT  (0x00000014) /* INT response wait Pck 5cycle
30                                         H'14 = (1/Pck*5cyc) / (1/Ick*3cyc) */
31
32 /* --- RAM allocation variable declaration --- */
33 volatile unsigned char u1SCIF0Rcvdata[16];
34 volatile unsigned char u1FlgSCIF0BrkInt;
35 volatile unsigned char u1FlgSCIF0RcvErrInt;
36 volatile unsigned short u2NumRcvData;
37
38 /***** Function Comment *****/
39 * Outline      : SCIF0_Initialize
40 *-----
41 * Declaration  : void SCIF0_Initialize(void)
42 *-----
43 * Functional description:
44 *             SCIF0 Initialize
45 *-----
46 * Return Value : -
47 * Argument     : -
48 *-----
49 * Input        : -
50 * Output       : -
51 *-----
52 * Notes       : -
53 *****/
54 void SCIF0_Initialize(void)
55 {
56     volatile unsigned short    u2loop;

```

5. サンプルプログラムリスト"scif.c" (4)

```

169 volatile unsigned char u1dummy;
170 volatile unsigned char u1ErrFlg;
171
172 PHCR = 0xFC80;          /* PH4-PH0 mode select "SCIF0" */
173 PHPUPR = 0x00;         /* set PH7-PH0 pullup off */
174
175 SCSPTR0.BIT.SPB2IO = 1; /* SPB2DT bit value is output to the SCIF_TXD pin */
176 SCSPTR0.BIT.SPB2DT = 1; /* output data is Hi-level */
177
178 SCSCR0.WORD = 0x0000;    /* TE,RE bit clear */
179 u1ErrFlg = ((SCFSR0.BIT.ER)|| (SCFSR0.BIT.DR)|| (SCFSR0.BIT.BRK)|| (SCLSR0.BIT.0RER));
/* read error flag */

180
181 if(u1ErrFlg == 0x01)
182 {
183     for(u2loop = 0; u2loop < NUM_RCV_DATABUF ; u2loop++)
184     {
185         u1dummy = SCFRDR0; /* receive data FIFO buffer refresh */
186     }
187     SCFSR0.WORD &= 0x0000; /* clear all status bit */
188     SCLSR0.BIT.0RER &= 0x00; /* clear 0RER bit */
189 }
190
191 SCFCR0.WORD = 0x0006;    /* set TFCL, RFCL bit, loopback disable, modem control disable */
192 SCSCR0.WORD = 0x0000;    /* internal clock, SCIF_SCK general port */
193 SCSMR0.WORD = 0x0020;    /* UART, 8bit, parity enable, even parity, 1stopbit, Pck */
194 SCBR0 = 0x08;           /* baud rate 115200bps */
195
196 for(u2loop = 0; u2loop < 0x0d90 ; u2loop++) /* 1-bit interval elapsed? */
197 {
198     nop();
199 }
200
201 SCFCR0.WORD = 0x0070;    /* clear TFCL,RFCL bit,RTRG0,1,TTRG */
202 SCSCR0.WORD = 0x0078;    /* transmit enable, receive enable, receive interrupt enable */
203
204 INT2PRI2 |= 0x1F000000; /* SCIF ch0 interrupt level 31 */
205 INT2MSKCR = 0x00000008; /* SCIF ch0 interrupt mask clear */
206 }
207
208 /***** Function Comment *****/
209 * Outline      : SCIF0_ReceiveInterrupt
210 *-----
211 * Declaration  : void SCIF0_RcvInterrupt(void)
212 *-----
213 * Functional description:
214 *             SCIF0 Receive Data full interrupt
215 *-----
216 * Return Value : -
217 * Argument     : -
218 *-----
219 * Input        : -
220 * Output       : -
221 *-----
222 * Notes       : -
223 *****/

```

6. サンプルプログラムリスト"scif.c" (5)

```

224 void SCIF0_RcvInterrupt(void)
225 {
226     volatile unsigned short    u2loop;
227     volatile unsigned short    u2dummy;
228
229     u2NumRcvData = SCRFDR0;
230
231     for(u2loop = 0; u2loop < u2NumRcvData ; u2loop++)
232     {
233         u1SCIF0Rcvdata[u2loop] = 0x00;
234         u1SCIF0Rcvdata[u2loop] = SCFRDR0;    /* read receive data    */
235     }
236
237     if(SCFSR0.BIT.DR == 0x01)
238     {
239         SCFSR0.BIT.DR &= 0x00;    /* clear DR bit    */
240     }
241     SCFSR0.BIT.RDF &= 0x00;    /* clear RDF bit    */
242
243     u2dummy = SCFSR0.WORD;    /* dummy read    */
244
245     SCIF0_SendData();
246
247     int_responstime_wait(INTC_RESPONSEWAIT); /* 5cyc(Pck=33MHz) wait    */
248 }
249
250 /***** Function Comment *****/
251 * Outline      : SCIF0_SendData
252 *-----
253 * Declaration  : void SCIF0_SendData(void)
254 *-----
255 * Functional description:
256 *       SCIF0 Send data
257 *-----
258 * Return Value : -
259 * Argument     : -
260 *-----
261 * Input        : -
262 * Output       : -
263 *-----
264 * Notes       : -
265 *****/ Function Comment End *****/
266 void SCIF0_SendData(void)
267 {
268     volatile unsigned short u2loop;
269
270     for(u2loop = 0; u2loop < u2NumRcvData ; u2loop++)
271     {
272         while(SCFSR0.BIT.TDFE != 1)
273         {
274             nop();
275         }
276
277         SCFTDR0 = u1SCIF0Rcvdata[u2loop];    /* set Send Data    */
278         SCFSR0.WORD &= 0x9f;    /* clear TDFE,TEND bit */
279     }

```

7. サンプルプログラムリスト"scif.c" (6)

```

280 }
281
282 /***** Function Comment *****/
283 * Outline      : SCIF0_ErrInterrupt
284 *-----
285 * Declaration  : void SCIF0_ErrInterrupt(void)
286 *-----
287 * Functional description:
288 *           SCIF0 Receive error interrupt
289 *-----
290 * Return Value : -
291 * Argument     : -
292 *-----
293 * Input        : -
294 * Output       : -
295 *-----
296 * Notes       : -
297 *****/ Function Comment End *****/
298 void SCIF0_ErrInterrupt(void)
299 {
300     volatile unsigned short    u2loop;
301     volatile unsigned char u1ErrFlg;
302     volatile unsigned char u1dummy;
303     volatile unsigned short    u2dummy;
304
305     if(SCFSR0.BIT.ER == 0x01)
306     {
307         SCFSR0.BIT.ER &= 0x00;          /* clear ER bit */
308     }
309
310     u2dummy = SCFSR0.WORD;              /* dummy read */
311
312     u1FlgSCIF0BrkInt = 1;              /* set Receive Error Interrupt flag */
313
314     u2NumRcvData = SCFRDR0;
315
316     for(u2loop = 0 ; u2loop < u2NumRcvData ; u2loop++)
317     {
318         u1ErrFlg = ((SCFSR0.BIT.FER)|| (SCFSR0.BIT.PER));
319
320         if(u1ErrFlg == 0x00)
321         {
322             u1SCIF0Rcvdata[u2loop] = SCFRDR0; /* read receive data */
323         }
324         else
325         {
326             u1SCIF0Rcvdata[u2loop] = 0x00;
327             u1dummy = SCFRDR0;          /* dummy read */
328         }
329     }
330
331     int_responstime_wait(INTC_RESPONSEWAIT); /* 5cyc(Pck=33MHz) wait */
332 }
333

```

8. サンプルプログラムリスト"scif.c" (7)

```

334 /***** Function Comment *****/
335 * Outline      : SCIF0_BrkInterrupt
336 *-----
337 * Declaration  : void SCIF0_BrkInterrupt(void)
338 *-----
339 * Functional description:
340 *             SCIF0 Break interrupt
341 *-----
342 * Return Value : -
343 * Argument     : -
344 *-----
345 * Input        : -
346 * Output       : -
347 *-----
348 * Notes       : -
349 *****/
350 void SCIF0_BrkInterrupt(void)
351 {
352     volatile unsigned short    u2loop;
353     volatile unsigned short    u2dummy;
354
355     if(SCFSR0.BIT.BRK == 0x01)
356     {
357         SCFSR0.BIT.BRK &= 0x00;          /* clear BRK bit */
358     }
359     else if(SCLSR0.BIT.ORER == 0x01)
360     {
361         SCLSR0.BIT.ORER &= 0x00;          /* clear ORER bit */
362     }
363
364     u2dummy = SCFSR0.WORD;                /* dummy read */
365     u2dummy = SCLSR0.WORD;                /* dummy read */
366
367     u1FlgSCIF0BrkInt = 1;                /* set BreakInterrupt flag */
368
369     u2NumRcvData = SCRFDR0;
370     for(u2loop = 0 ; u2loop < u2NumRcvData ; u2loop++)
371     {
372         u1SCIF0Rcvdata[u2loop] = 0x00;    /* receive data initial */
373         u1SCIF0Rcvdata[u2loop] = SCFRDR0; /* read receive data */
374     }
375
376     int_responstime_wait(INTC_RESPONSEWAIT); /* 5cyc(Pck=33MHz) wait */
377 }
378

```

9. サンプルプログラムリスト"scif.c" (8)

```

379 /***** Function Comment *****/
380 * Outline      : SCIF0_SendBrkSignal
381 *-----
382 * Declaration  : void SCIF0_SendBrkSignal(void)
383 *-----
384 * Functional description:
385 *             SCIF0 Send Break Signal
386 *-----
387 * Return Value : -
388 * Argument     : -
389 *-----
390 * Input        : -
391 * Output       : -
392 *-----
393 * Notes       : -
394 *****/ Function Comment End *****/
395 void SCIF0_SendBrkSignal(void)
396 {
397     volatile unsigned short u2loop;
398
399     SCSPTR0.BIT.SPB2IO = 1;          /* SPB2DT bit value is output to the SCIF_TXD pin */
400     SCSPTR0.BIT.SPB2DT = 0;          /* output data is low-level */
401
402     SCSCR0.BIT.TE = 0;               /* clear TE bit */
403
404     for(u2loop = 0 ; u2loop < 0x87a2 ; u2loop++) /* 10bit(S+1byte+P+1STOP) interval elapsed? */
405     {
406         nop();
407     }
408
409     SCSPTR0.BIT.SPB2IO = 0;          /* SPB2DT bit value is not output to the SCIF_TXD pin */
410     SCSCR0.BIT.TE = 1;               /* set TE bit */
411 }
412
413 #pragma inline_asm(int_responstime_wait)
414 static void int_responstime_wait(unsigned int wait_time)
415 {
416     ?0001:
417         DT      R4
418         BF      ?0001
419         NOP
420 }

```

10. サンプルプログラムリスト"intprg.c"

```

1  /*"FILE COMMENT"*****
2  *      System Name:   SH7780 Sample Program
3  *      File Name    :   intprg.c
4  *      Version      :   1.00.00
5  *      Contents     :   SH7780 Initialize Program
6  *      Model        :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7  *      CPU          :   SH7780
8  *      Compiler     :   SHC.9.1.00
9  *      OS           :   none
10 *
11 *      note          :   < Caution >
12 *                      This sample program is provided simply as a reference and
13 *                      its operation is not guaranteed.
14 *                      Use this sample program as a technical reference when
15 *                      developing software.
16 *
17 *      Copyright(C) 2007 Renesas Technology Corp. All Rights Reserved
18 *
19 *      History       :   2007/12/26 ver 1.00.00
20 *
21 *****/
22 #include <machine.h>
23
24 /* --- Function Definition(internal) --- */
25 static void int_responstime_wait(unsigned int wait_time);
26
27 /* --- Symbol Definition --- */
28 #define INTC_RESPONSEWAIT (0x00000014) /* INT response wait Pck 5cycle
29                                     H'14 = (1/Pck*5cyc) / (1/Ick*3cyc) */
30
31 /* --- RAM allocation variable declaration --- */
32
33 #pragma section IntPRG

```



```

240 /* H'700 SCIF ch-0 receive error interrupt */
241 void INT_SCIF0_ERI0(void)
242 {
243     SCIF0_ErrInterrupt(); /* (additional part from Initialize program) */
244 }
245 /* H'720 SCIF ch-0 receive FIFO data full or receive data ready interrupt */
246 void INT_SCIF0_RXI0(void)
247 {
248     SCIF0_RcvInterrupt(); /* (additional part from Initialize program) */
249 }
250 /* H'740 SCIF ch-0 break or overrun error interrupt */
251 void INT_SCIF0_BRI0(void)
252 {
253     SCIF0_BrkInterrupt(); /* (additional part from Initialize program) */
254 }

```

11. サンプルプログラムリスト"vecttbl.src"

```

1  ;*****"FILE COMMENT"*****
2  ;      System Name:   SH7780 Sample Program
3  ;      File Name    :   vecttbl.src
4  ;      Version      :   1.00.00
5  ;      Contents     :   SH7780 Initialize Program
6  ;      Model        :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7  ;      CPU          :   SH7780
8  ;      Compiler     :   SHC.9.1.00
9  ;      OS           :   none
10 ;
11 ;      note          :   < Caution >
12 ;                      This sample program is provided simply as a reference and
13 ;                      its operation is not guaranteed.
14 ;                      Use this sample program as a technical reference when
15 ;                      developing software.
16 ;
17 ;      Copyright(C) 2007 Renesas Technology Corp. All Rights Reserved
18 ;
19 ;      History       :   2007/12/26 ver 1.00.00
20 ;
21 ;*****/
22
23      .include      "vect.inc"
24
25      .section      VECTTBL,data
26      .export       _RESET_VECTORS

333 ;SCIF-ch0
334      ;H'700      ERI0
335      .data.b      H'F0      /* (change part from Initialize program) */
336      ;H'720      RXI0
337      .data.b      H'F0      /* (change part from Initialize program) */
338      ;H'740      BRI0
339      .data.b      H'F0      /* (change part from Initialize program) */
340      ;H'760      TXI0
341      .data.b      H'00

```

4. 参考ドキュメント

- ハードウェアマニュアル
SH7780 ハードウェアマニュアル
(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。)
- ソフトウェアマニュアル
SH-4A ソフトウェアマニュアル
(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

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.03.21	—	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は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。
11. 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くよう、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。
12. 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。
13. 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