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ものですが、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て、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。

H8/300L SLP シリーズ

クロック同期式シリアルデータ同時送受信

要旨

クロック同期式シリアル転送機能を使用して, 4 バイトの 8 ビットデータの同時送受信動作を行ないます。データの最下位ビットから受信する LSB ファースト方式による送受信を行ないます。

動作確認デバイス

H8/38024

目次

1. 仕様

(1) 図 1 に示すようにクロック同期式シリアル転送機能を使用して，4 バイトの 8 ビットデータの同時送受信動作を行ないます。
(2) 転送クロックは，内部クロックを使用し 4 µs の転送クロック周期で同時送受信動作を行います。
(3) 送受信するデータのデータ長は 8 ビットで，データの最下位ビットから受信する LSB ファースト方式による送受信を行ないます。

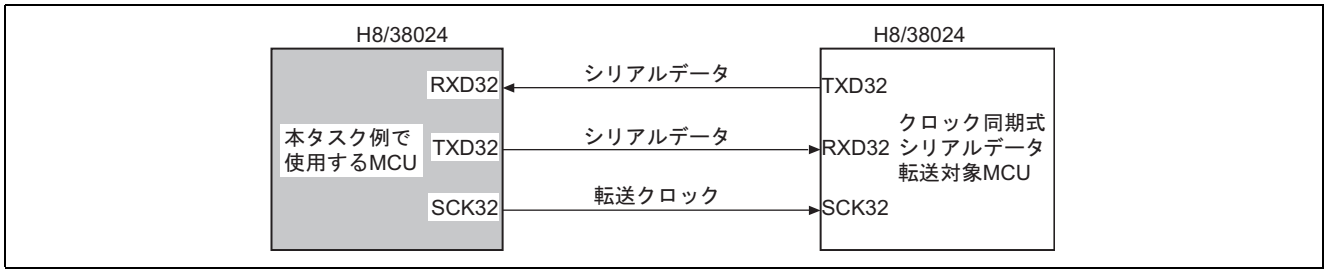
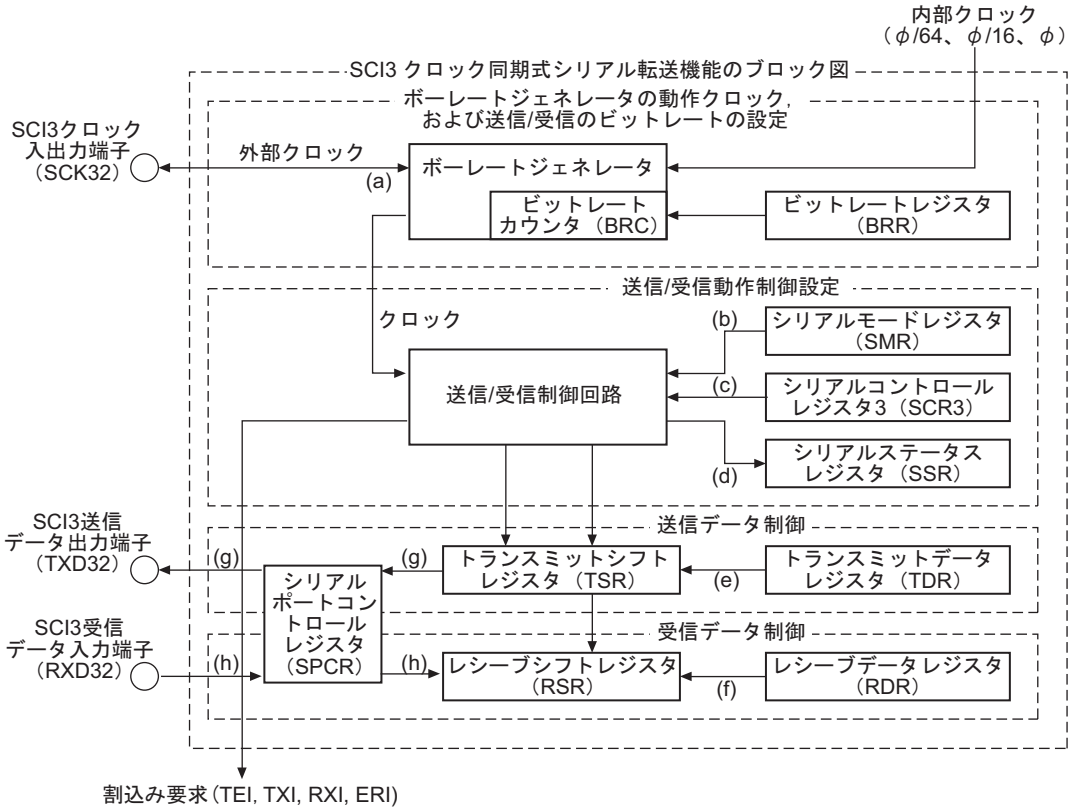


図 1　クロック同期式シリアルデータ同時送受信

2. 使用機能説明

(1) 本タスク例では，シリアルコミュニケーションインタフェース（SCI：Serial Communication Interface）を使用して，クロック同期式のシリアルデータの送受信同時動作を行ないます。図 2 にクロック同期式シリアルデータ送受信同時動作のブロック図を示します。以下にクロック同期式シリアルデータ同時送受信のブロック図について説明します。

- システムクロック（φ）は，10MHz の OSC クロックで，CPU および周辺機能を動作させるための基準クロックです。
- 受信エラーの検出には，オーバランエラーのみを行います。
- クロック同期式モードではデータ長は 8 ビットになります。
- レシーブシフトレジスタ（RSR）は，シリアルデータを受信するためのレジスタです。RSR に RXD32 端子から入力されたシリアルデータを，LSB（ビット 0）から受信した順にセットしパラレルデータに変換します。1 バイトのデータを受信すると，データは自動的に RDR へ転送されます。CPU から RSR を直接リード/ライトすることはできません。
- レシーブデータレジスタ（RDR）は，受信したシリアルデータを格納する 8 ビットのレジスタです。1 バイトのデータの受信が終了すると，受信したデータを RSR から RDR へ転送し，受信動作を完了します。その後，RSR は受信可能となります。RSR と RDR はダブルバッファになっているため連続した受信動作が可能です。RDR は受信専用レジスタなので CPU からライトできません。
- トランスミットシフトレジスタ（TSR）は，シリアルデータを送信するためのレジスタです。TDR から送信データをいったん TSR に転送し，LSB（ビット 0）から順に TXD32 端子に送出することでシリアルデータ送信を行ないます。1 バイトのデータを送信すると，自動的に TDR から TSR へ次の送信データを転送し，送信を開始します。ただし，TDR にデータが書き込まれていない（TDRE に"1"がセットされている）場合には TDR から TSR へのデータ転送は行ないません。CPU から TSR を直接リード/ライトすることはできません。
- トランスミットデータレジスタ（TDR）は，送信データを格納する 8 ビットのレジスタです。TSR の"空"を検出すると，TDR に書き込まれた送信データを TSR に転送し，シリアルデータ送信を開始します。TSR のシリアルデータ送信中に，TDR に次の送信データをライトしておくと，連続送信が可能です。TDR は，常に CPU によるリード/ライトが可能です。
- シリアルモードレジスタ（SMR）は，シリアルデータ通信フォーマットの設定と，内蔵ボーレートジェネレータのクロックソースを選択するための 8 ビットのレジスタです。
- シリアルコントロールレジスタ 3（SCR3）は，送信/受信動作および送信/受信クロックソースの選択を行なう 8 ビットのレジスタです。
- シリアルステータスレジスタ（SSR）は，SCI3 のステータスフラグと送受信マルチプロセッサビットで構成されています。TDRE，RDRF，OER，PER，FER はクリアのみ可能です。
- 転送クロックは，3 種類の内部クロックと外部クロックから選択できます。内部クロックを選択した場合は，SCK32 端子は出力端子となります。クロック連続出力モードに設定すると選択したクロックを SCK32 端子から連続して出力します。外部クロックを選択した場合は，SCK32 端子はクロック入力端子となります。
- 本タスク例では，転送クロックソースを内蔵ボーレートジェネレータのφクロックにし，転送クロック周期を 4μs に設定しています。
- SCI3 の転送フォーマットは 8 ビットのデータを選択可能です。データの最下位ビットから送受信される LSB ファースト方式による転送を行ないます。送信データは，転送クロックの立ち下がりから次の立ち上がりまで出力されます。また，受信データは転送クロックの立ち上がりで取り込まれます。
- 本タスク例では，動作モードを 8 ビットモードに設定し，8 ビットのデータ送受信を行ないます。
- SCI3 クロック（SCK32）は，SCI3 のクロック入出力端子です。
- SCI3 レシーブデータ入力（RXD32）は，SCI3 の受信データの入力端子です。
- SCI3 トランスミットデータ出力（TXD32）は，SCI3 の送信データの出力端子です。
- シリアルポートコントロールレジスタ（SPCR）は，RXD32 端子，P42/TXD32 端子を制御する 8 ビットのレジスタです。本タスクでは，P42/TXD32 端子を TXD32 出力端子に，RXD32 端子，TXD32 端子の入出力データを反転しないように設定しています。



【注】

(a) SMR **で選択されたボーレートジェネレータの動作クロック（φ）を出力します。**

(b) **シリアルデータ通信フォーマットの設定と,ボーレートジェネレータのクロックソースを選択します。**

(c) **送信/受信動作，クロック同期式モードでのクロック出力端子の選択をします。**

(d) **ステータスフラグ（トランスミットデータレジスタエンプティ,レシーブデータレジスタフル,オーバランエラー）により** SCI3 **の動作状態を示します。**

(e) TSR **の"空"を検出することにより，**TDR **に書き込まれた送信データを** TSR **に転送します。**

(f) 1 **バイトのデータの受信が終了すると，受信したデータを** RSR **から** RDR **へ転送します。**

(g) **送信データを送信します。**

(h) **受信データを受信します。**

図 2　**クロック同期式シリアルデータ同時送受信のブロック図**

(2) 表 1 に本タスク例の機能割付けを示します。表 1 に示すように機能を割り付け，クロック同期式シリアルデータ受信を行ないます。

表 1　機能割付け

機能	機能割付け
TSR	シリアルデータを送信するためのレジスタ
TDR	送信データを格納するレジスタ
RSR	シリアルデータを受信するためのレジスタ
RDR	受信データを格納するレジスタ
SMR	シリアルデータ通信フォーマット，ボーレートジェネレータのクロックソースの設定
SSR	SCI3 の動作状態を示すステータスフラグ
BRR	送信/受信のビットレートを設定
SCR3	送受信動作の許可，TXD32 出力端子設定，RXD32 入力端子設定，SCK32 の端子機能をクロック出力端子設定
SCK32	SCI3 のクロック出力端子
TXD32	SCI3 の送信データ出力端子
RXD32	SCI3 の受信データ入力端子
SPCR	TXD32 出力端子設定

3. 動作説明

(1) 図 3 に動作説明を示します。図 3 に示すようなハードウェア処理, およびソフトウェア処理によりクロック同期式シリアルデータ同時送受信動作を行ないます。

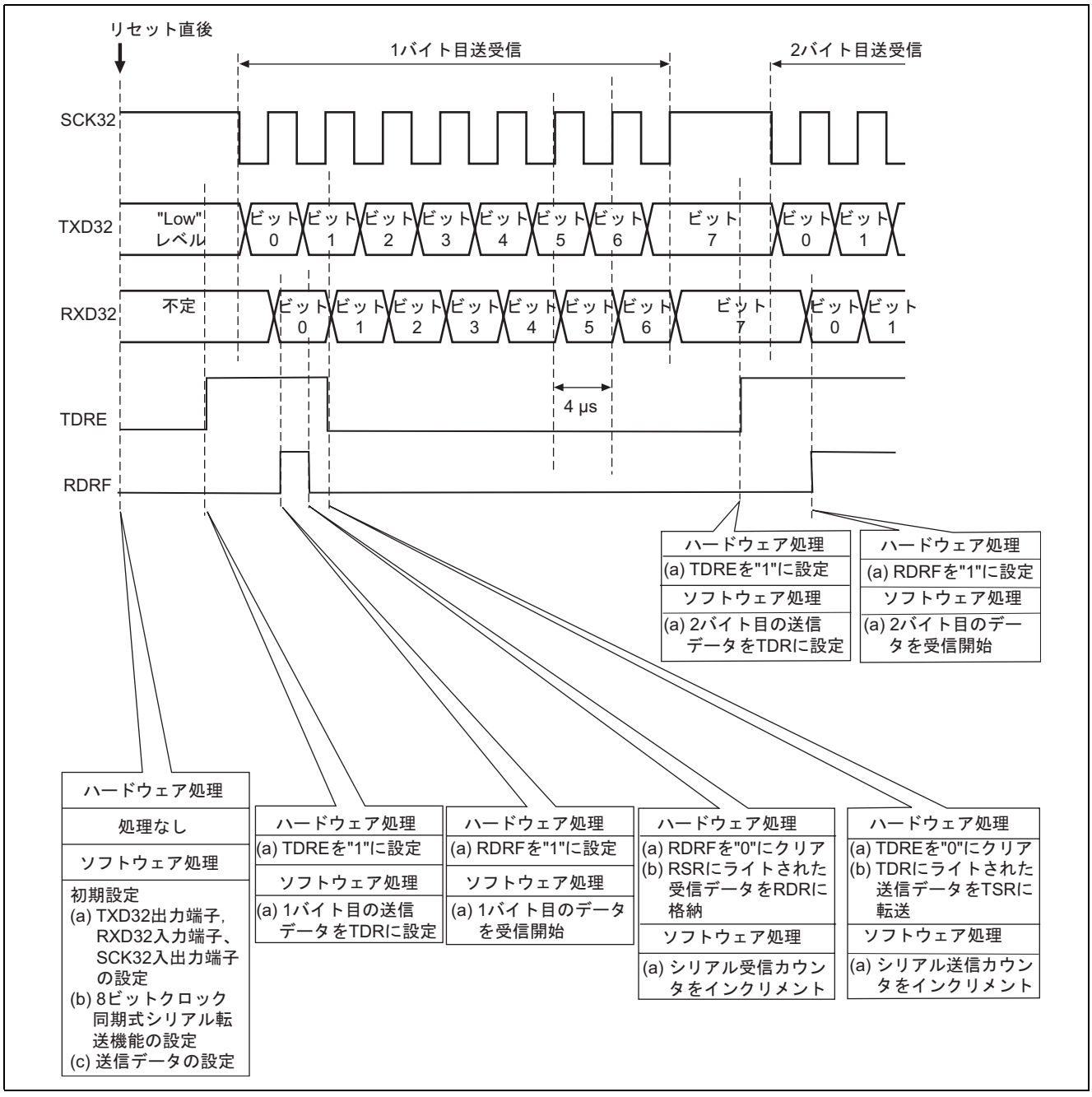


図 3　クロック同期式シリアルデータ同時送受信の動作説明

4. ソフトウェア説明

(1) モジュール説明

本タスク例のモジュールを表 2 に示します。

表 2　モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	転送データの設定，クロック同期式シリアルデータ送受信の設定，受信データを RAM に格納，4 バイトのデータを送受信したところで終了

(2) 引数の説明

本タスク例の引数を表 3 に示します。

表 3　引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD[0]～STD[3]	クロック同期式シリアル送信データ	メインルーチン	1 バイト	入力
SRD[0]～SRD[3]	クロック同期式シリアル受信データ	メインルーチン	1 バイト	出力

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを表 4 に示します。

表 4　使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
SPCR	SPC32	シリアルポートコントロールレジスタ （P42/TXD32 端子機能切り替え） ：SPC32="0"のとき，P42/TXD32 端子を P42 端子機能に設定 ：SPC32="1"のとき，P42/TXD32 端子を TXD32 端子機能に設定	H'FF91 ビット 5	1
	SCINV3	シリアルポートコントロールレジスタ （TXD32 端子出力データ反転切り替え） ：SCINV3="0"のとき，TXD32 の出力データを反転しない ：SCINV3="1"のとき，TXD32 の出力データを反転する。	H'FF91 ビット 3	0
	SCINV2	シリアルポートコントロールレジスタ （RXD32 端子入力データ反転切り替え） ：SCINV2="0"のとき，RXD32 の入力データを反転しない ：SCINV2="1"のとき，RXD32 の入力データを反転する。	H'FF91 ビット 2	0
SMR	COM	シリアルモードレジスタ（コミュニケーションモード） ：COM="0"のとき，コミュニケーションモードを調歩同期式モードに設定 ：COM="1"のとき，コミュニケーションモードをクロック同期式モードに設定	H'FFA8 ビット 7	1
	MP	シリアルモードレジスタ（マルチプロセッサモード） ：クロック同期式モードではこのビットを"0"に設定する。	H'FFA8 ビット 2	0
	CKS1 CKS0	シリアルモードレジスタ（クロックセレクト 1，0） ：CKS1="0"，CKS0="0"のとき，内蔵ボーレートジェネレータのクロックソースをφに設定	H'FFA8 ビット 1 ビット 0	CKS1="0" CKS0="0"
BRR		ビットレートレジスタ ：BRR=H'04 のとき，外部動作クロックとあわせた送信のビットレートを 250k（bit/s）に設定	H'FFA9	H'04

表 4　使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
SCR3	TE	シリアルコントロールレジスタ 3(トランスミットイネーブル) ：TE="0"のとき，送信動作を禁止 ：TE="1"のとき，送信動作を許可	H'FFAA ビット 5	0
	RE	シリアルコントロールレジスタ 3（レシーブイネーブル） ：RE="0"のとき，受信動作を禁止 ：RE="1"のとき，受信動作を許可	H'FFAA ビット 4	0
	CKE1 CKE0	シリアルコントロールレジスタ 3（クロックイネーブル 1，0） ：CKE1="0"，CKE0="0"のとき，クロック同期式モードにおいてクロックソースを内部クロック,SCK32 端子機能をクロック出力に設定	H'FFAA ビット 1 ビット 0	CKE1="0" CKE0="0"
TDR		トランスミットデータレジスタ ：送信データを格納する 8 ビットのレジスタ	H'FFAB	—
SSR	TDRE	シリアルステータスレジスタ （トランスミットデータレジスタエンプティ） ：TDRE="0"のとき，TDR にライトされた送信データが TSR に転送されていないことを示す ：TDRE="1"のとき，TDR に送信データがライトされていない，または TDR にライトされた送信データが TSR に転送されたことを示す	H'FFAC ビット 7	1
	RDRF	シリアルステータスレジスタ（レシーブデータレジスタフル） ：RDRF="0"のとき，RDR に受信データが格納されていない ：RDRF="1"のとき，RDR に受信データが格納されている	H'FFAC ビット 6	0
	OER	シリアルステータスレジスタ（オーバランエラー） ：OER="0"のとき，受信中，または受信を完了 ：OER="1"のとき，受信時にオーバランエラーが発生	H'FFAC ビット 5	0
	TEND	シリアルステータスレジスタ（トランスミットエンド） ：TEND="0"のとき，送信中であることを示す ：TEND="1"のとき，送信を終了したことを示す	H'FFAC ビット 2	—
RDR		レシーブデータレジスタ ：受信データを格納する 8 ビットのレジスタ	H'FFAD	—

(4) 使用 RAM 説明

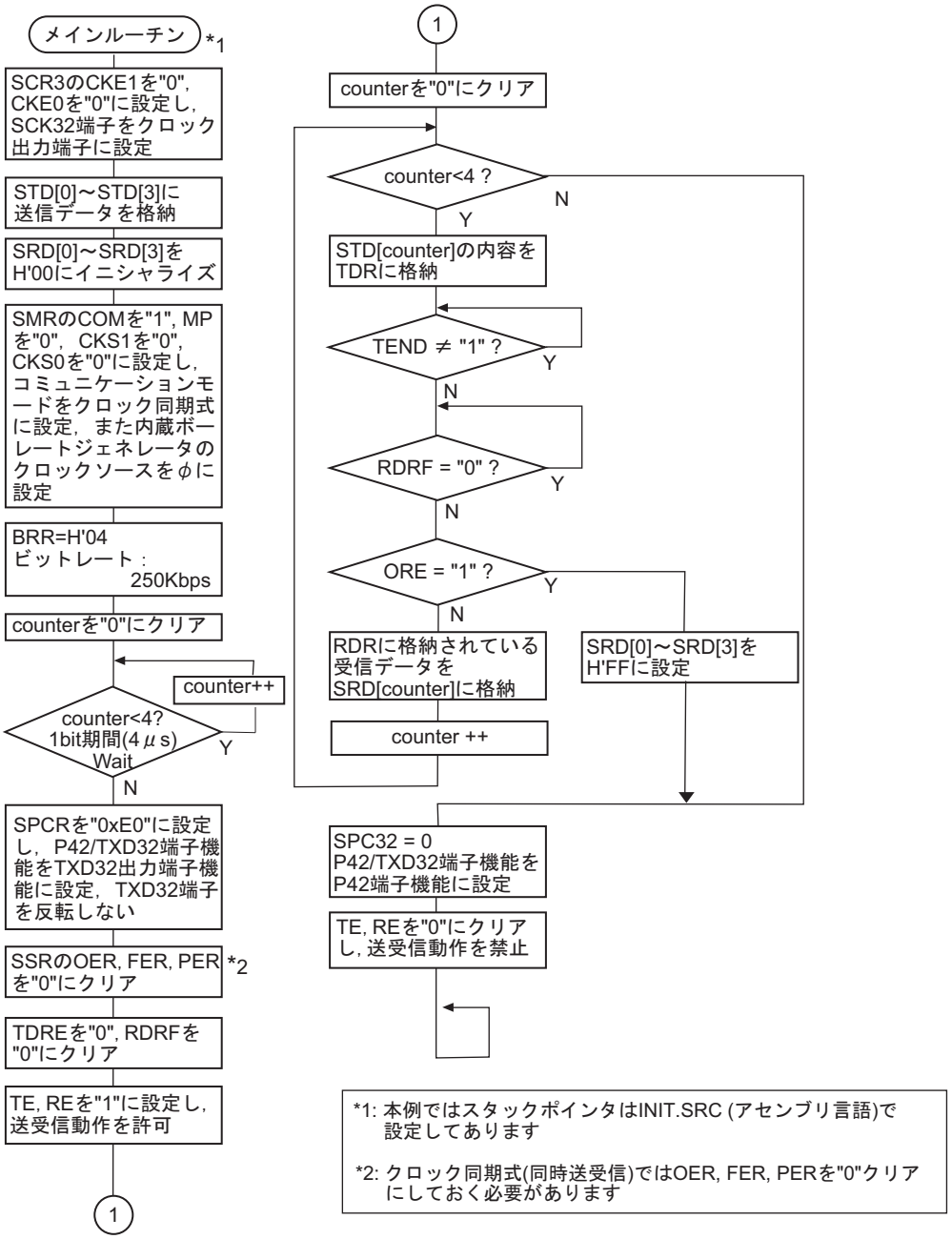
本タスク例の使用 RAM を表 5 に示します。

表 5　使用 RAM 説明

ラベル名	機能	アドレス	使用モジュール名
STD[0]	クロック同期式シリアル送信データの 1 バイト目を格納	H'FB80	メインルーチン
STD[1]	クロック同期式シリアル送信データの 2 バイト目を格納	H'FB81	メインルーチン
STD[2]	クロック同期式シリアル送信データの 3 バイト目を格納	H'FB82	メインルーチン
STD[3]	クロック同期式シリアル送信データの 4 バイト目を格納	H'FB83	メインルーチン
SRD[0]	クロック同期式シリアル受信データの 1 バイト目を格納	H'FB84	メインルーチン
SRD[1]	クロック同期式シリアル受信データの 2 バイト目を格納	H'FB85	メインルーチン
SRD[2]	クロック同期式シリアル受信データの 3 バイト目を格納	H'FB86	メインルーチン
SRD[3]	クロック同期式シリアル受信データの 4 バイト目を格納	H'FB87	メインルーチン

5. フローチャート

（1）メインルーチン



6. プログラムリスト

6.1 INIT.SRC（プログラムリスト）

```
      .EXPORT   _INIT
      .IMPORT   _main
  ;
      .SECTION  P,CODE
  _INIT:
      MOV.W     #H'FF80,R7
      LDC.B     #B'10000000,CCR
      JMP       @_main
  ;
      .END
```

```
/************************************************************/
/*                                                          */
/*  H8/300L Super Low Power Series                          */
/*     -H8/38024 Series-                                    */
/*  Application Note                                        */
/*                                                          */
/*  'Synchronous Serial Data Simultaneous                   */
/*   Transmission/Reception'                                */
/*                                                          */
/*  Function                                                */
/*  : Serial Communication Interface                        */
/*    Synchronous Serial Interface                          */
/*    -Transmitting/Receiving                               */
/*                                                          */
/*  External Clock : 10MHz                                  */
/*  Internal Clock : 5MHz                                   */
/*  Sub Clock      : 32.768kHz                              */
/*                                                          */
/************************************************************/

#include   <machine.h>

/************************************************************/
/*  Symbol Definition                                       */
/************************************************************/
struct BIT {
   unsigned char   b7:1;      /* bit7 */
   unsigned char   b6:1;      /* bit6 */
   unsigned char   b5:1;      /* bit5 */
   unsigned char   b4:1;      /* bit4 */
   unsigned char   b3:1;      /* bit3 */
   unsigned char   b2:1;      /* bit2 */
   unsigned char   b1:1;      /* bit1 */
   unsigned char   b0:1;      /* bit0 */
};

#define    SMR_BIT    (*(struct BIT *)0xFFA8)          /* Serial Mode Register           */
#define    COM        SMR_BIT.b7                      /* Communication Mode             */
#define    CHR        SMR_BIT.b6                      /* Character Length               */
#define    PE         SMR_BIT.b5                      /* Parity Enable                  */
#define    PM         SMR_BIT.b4                      /* Parity Mode                    */
#define    STOP       SMR_BIT.b3                      /* Stop Bit Length                */
#define    MP         SMR_BIT.b2                      /* Multiprocesor Mode             */
```

```
#define    CKS1      SMR_BIT.b1                    /* Clock Select 1               */
#define    CKS0      SMR_BIT.b0                    /* Clock Select 0               */
#define    BRR       *(volatile unsigned char *)0xFFA9   /* Bit Rate Register            */
#define    SCR3      *(volatile unsigned char *)0xFFAA   /* Serial Control Register 3      */
#define    SCR3_BIT   (*(struct BIT *)0xFFAA)          /* Serial Control Register 3      */
#define    TIE       SCR3_BIT.b7                   /* Transmit Interrupt Enable      */
#define    RIE       SCR3_BIT.b6                   /* Receive Interrupt Enable       */
#define    TE        SCR3_BIT.b5                   /* Transmit Enable              */
#define    RE        SCR3_BIT.b4                   /* Receive Enable               */
#define    MPIE      SCR3_BIT.b3                    /* Multiprocessor Interrupt Enable  */
#define    TEIE      SCR3_BIT.b2                    /* Transmit End Interrupt Enable   */
#define    CKE1      SCR3_BIT.b1                    /* Clock Enable 1               */
#define    CKE0      SCR3_BIT.b0                    /* Clock Enable 0               */
#define    TDR       *(volatile unsigned char *)0xFFAB   /* Transmit Data Register         */
#define    SSR       *(volatile unsigned char *)0xFFAC   /* Serial Status Register         */
#define    SSR_BIT    (*(struct BIT *)0xFFAC)          /* Serial Status Register         */
#define    TDRE      SSR_BIT.b7                    /* Transmit Data Register Empty    */
#define    RDRF      SSR_BIT.b6                    /* Receive Data Register Full     */
#define    OER       SSR_BIT.b5                    /* Overrun Erorr                */
#define    FER       SSR_BIT.b4                    /* Framing Erorr                */
#define    PER       SSR_BIT.b3                    /* Parity Erorr                 */
#define    TEND      SSR_BIT.b2                    /* Transmit End                 */
#define    MPBR      SSR_BIT.b1                    /* Multiprocessor Bit Receive     */
#define    MPBT      SSR_BIT.b0                    /* Multiprocessor Bit Transfer     */
#define    SPCR      *(volatile unsigned char *)0xFF91   /* Transmit Data Register         */
#define    SPCR_BIT   (*(struct BIT *)0xFF91)          /* Port Mode Register 1           */
#define    SPC32      SPCR_BIT.b5                    /* TXD Output Terminal            */
#define    RDR       *(volatile unsigned char *)0xFFAD   /* Receive Data Register          */

/************************************************************/
/*  Function define                                         */
/************************************************************/
extern void INIT ( void );                          /* SP Set                       */
void       main ( void );

/************************************************************/
/*  RAM define                                             */
/************************************************************/
unsigned char   STD[4];
unsigned char   SRD[4];

/************************************************************/
/*  Vector Address                                          */
/************************************************************/
#pragma section    V1                                /* VECTOR SECTOIN SET            */
void (*const VEC_TBL1[])(void) = {
/* 0x00 - 0x0f */
   INIT                                          /* 00 Reset                     */
};

#pragma section                                      /* P                            */
/************************************************************/
/*  Main Program                                            */
/************************************************************/
void    main ( void )
{
   unsigned char   counter;
```

```
CKE1 = 0;                              /* Initialize Clock Enable 1 Output          */
CKE0 = 0;                              /* Initialize Clock Enable 0 Output          */

STD[0] = 0x99;                          /* Set Serial Transfer Data 0               */
STD[1] = 0x55;                          /* Set Serial Transfer Data 1               */
STD[2] = 0xAA;                          /* Set Serial Transfer Data 2               */
STD[3] = 0xFF;                          /* Set Serial Transfer Data 3               */

SRD[0] = 0x00;                          /* Initialize Serial Receiving Data 0        */
SRD[1] = 0x00;                          /* Initialize Serial Receiving Data 1        */
SRD[2] = 0x00;                          /* Initialize Serial Receiving Data 2        */
SRD[3] = 0x00;                          /* Initialize Serial Receiving Data 3        */

COM = 1;                               /* Initialize Communication Mode             */
MP = 0;                                /* Initialize Multiprocesor Mode             */
CKS1 = 0;                              /* Initialize Clock Select 1 ϕ               */
CKS0 = 0;                              /* Initialize Clock Select 0 ϕ               */

BRR = 4;
for(counter = 0; counter < 4; counter++);   /* Serial Transmitting Data Counter 4 Loop     */

SPCR = 0xE0;                            /* Initialize Output Port TXD                */

OER = 0;                               /* Clear OER                                 */
FER = 0;                               /* Clear FER                                 */
PER = 0;                               /* Clear PER                                 */

TDRE = 0;                              /* Clear TDRE                                */
RDRF = 0;                              /* Clear RDRF                                */

SCR3 = 0x30;                            /* Start Serial Transmitting / Receiving       */

for(counter = 0; counter < 4; counter++){   /* Serial Transmitting Data Counter 4 Loop     */

   TDR = STD[counter];                  /* Save Serial Transmitting Data             */

   while(TEND != 1){                    /* End Serial Transmitting                   */
      ;
   }

   while(RDRF == 0){                    /* End Serial Receive End ?                  */
      ;
   }

   if (OER == 1){                       /* Overrun Erorr Flag = 1 ?                  */
      SRD[0] = 0xFF;                    /* Overrun Erorr 0                           */
      SRD[1] = 0xFF;                    /* Overrun Erorr 1                           */
      SRD[2] = 0xFF;                    /* Overrun Erorr 2                           */
      SRD[3] = 0xFF;                    /* Overrun Erorr 3                           */
      break;
   }
   else{
      SRD[counter] = RDR;               /* Save Serial Receiving Data                */
   }
}
```

```
    SPC32 = 0;
    SCR3 = 0x00;                          /* Initialize Transmitting / Receiving Enable   */

    while(1){
        ;
    }
}
```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
P	H'0100
B	H'FB80

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.19	—	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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ルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾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の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 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全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(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。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位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。ルネサス テクノロジは、適用可否に対する責任は負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。