

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。

32192/32196 グループ

シリアルインタフェースを使用した応用例（UART モード）

1. 要約

この資料は 32192/32196 グループのシリアルインタフェースを使用した参考プログラム例を掲載しています。

2. はじめに

この資料で説明する応用例は次のマイコン、条件での利用に適用されます。

- ・マイコン : 32192 グループ (M32192F8VFP、M32192F8UFP、M32192F8TFP)
32196 グループ (M32196F8VFP、M32196F8UFP、M32196F8TFP)
- ・動作周波数 : 128~160MHz (参考プログラムは 160MHz を想定して作成しています)
- ・動作ボード : 32192 グループ用スタータキット

3. 応用技術の説明

3.1 シリアルインタフェース概要

32192/32196 グループは、SIO0～SIO5 の計 6 チャンネルのシリアルインタフェースを内蔵しています。SIO0、SIO1、SIO4、SIO5 は CSIO モード (クロック同期形シリアルインタフェース) と、UART モード (クロック非同期形シリアルインタフェース) を選択できます。SIO2 および SIO3 は UART モード専用です。

- ・ CSIO モード (クロック同期形シリアルインタフェース)

転送クロックに同期して通信を行うモードで、送受信間で同一のクロックを使用します。転送データ長は 8～16 ビットの範囲で選択できます。

- ・ UART モード (クロック非同期形シリアルインタフェース)

任意の転送速度、転送データフォーマットを設定できるモードです。転送データ長は 7 ビット、8 ビット、9 ビットから選択できます。

SIO0～SIO5 は、それぞれ送信 DMA 転送要求と受信 DMA 転送要求を持っています。内蔵 DMAC を用いることにより、高速なシリアル通信が可能となり、また、データ通信にともなう CPU の負荷も低減されます。

UART モード時の転送速度は、クロックデバイダ分周値とボーレートレジスタ設定値 (BRG) で決まります。以下に各ボーレートにおける設定値の例を示します。

表 3.1.1 ボーレートレジスタ設定例

項目 ボーレート [bps]	クロックデバイダカウントソース = 20MHz時			
	クロックデバイダ 分周値 [分周]	BRG 設定値	誤差 [%]	実際のボーレート [bps]
300	32	129	0.16	300.48
600	32	64	0.16	600.96
1200	8	129	0.16	1201.92
2400	8	64	0.16	2403.85
4800	8	32	-1.36	4734.85
9600	1	129	0.16	9615.38
12500	1	99	0.00	12500.00
14400	1	86	-0.22	14367.82
19200	1	64	0.16	19230.77
28800	1	42	0.94	29069.77
31250	1	39	0.00	31250.00
38400	1	32	-1.36	37878.79
57600	1	21	-1.36	56818.18
62500	1	19	0.00	62500.00
115200	1	10	-1.36	113636.36
125000	1	9	0.00	125000.00
250000	1	4	0.00	250000.00
500000	1	2	-16.67	416666.67
625000	1	1	0.00	625000.00
1000000	-	-	-	-
1250000	1	0	0.00	1250000.00

UART モード時のボーレートレジスタ設定値の計算方法は以下のようになります。

$$\text{BRG 設定値} = \frac{\text{クロックデバイダカウントソース}}{\text{ボーレート [bps]} \times \text{クロックデバイダ分周値} \times 16} - 1$$

クロックデバイダカウントソース : f(BCLK)、または f(BCLK)/2
 ボーレートレジスタ設定値(BRG) : H'00 ~ H'FF
 クロックデバイダ分周値 : 1, 8, 32, 256

シリアルインタフェースの詳細は 32192/32196 グループのハードウェアマニュアルを参照してください。

4. シリアルインタフェースを使用した参考プログラム

4.1 参考プログラムの概要

本参考プログラムでは、SIO0 から出力、SIO1 で受信します。共に 8 ビット UART で通信を行います。送信は、引数で指定された送信データ格納領域のアドレスから、送信バイト数分のデータを送信します。受信は割り込み処理によって行います。SIO0 と SIO1 を結線することで、単体で UART による全二重通信が行えます。

本参考プログラムでは、通信速度を 62.5kbps に設定しています。

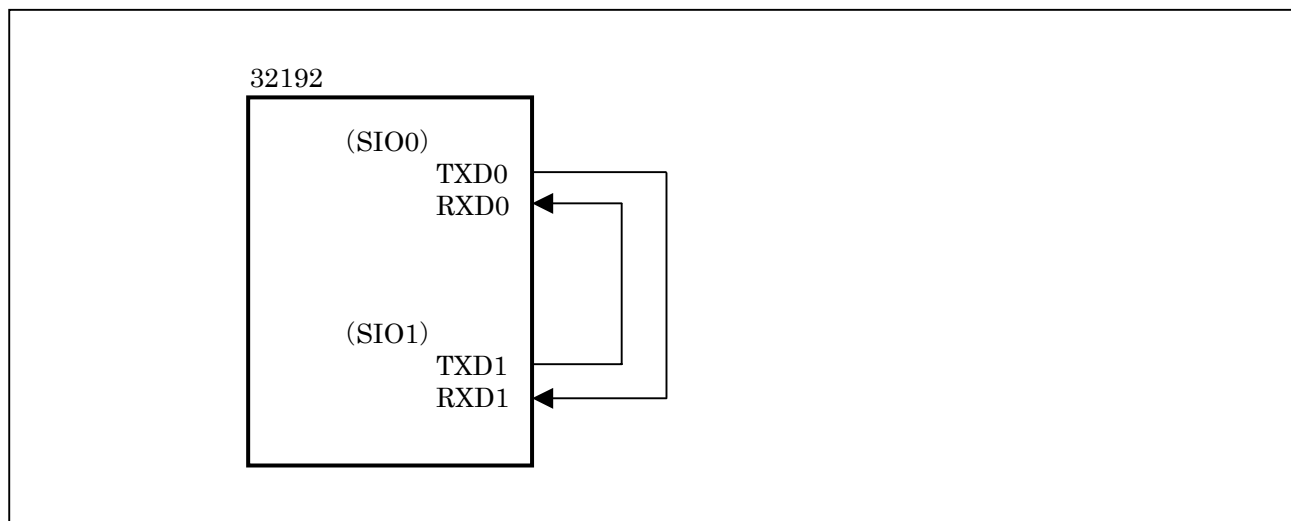


図 4.1.1 結線図

4.2 送受信処理

ここでは、UART モード送受信の初期設定フローを図 4.2.1 に示します

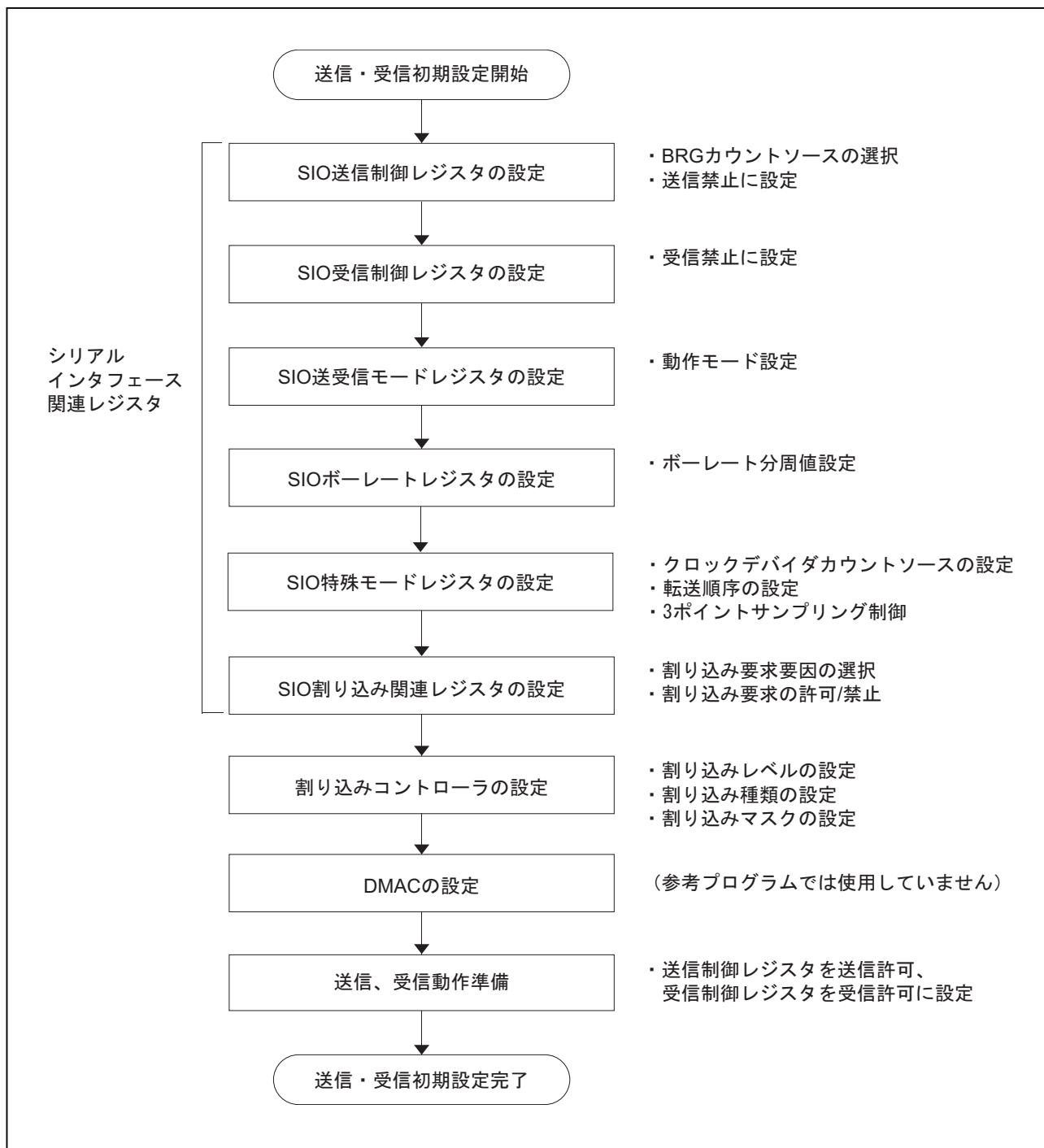


図 4.2.1 UART モード送受信初期設定フロー図

4.3 参考プログラムの解説

注. 使用しているレジスタを (レジスタ名 : ビット名) と表記しています。

4.3.1 各種初期化関数 (init_func())

- (1) ポート初期化関数の呼び出し

4.3.2 ポート初期化関数 (port_init())

- (1) 出力ポートの初期設定
 - ・ポート入力特別機能制御レジスタのポート入力許可ビットを入力許可に設定 (PICNT : PIEN0)
 - ・P11 データレジスタの初期化 (P11DATA)
 - ・P11 方向レジスタを出力モードに設定 (P11DIR)
 - ・P11 動作モードレジスタを汎用ポートに設定 (P11MOD)

注. ・データレジスタの設定を行う前に、方向レジスタを出力に設定すると、データレジスタへ書き込みが行われるまでの間、不定値が出力されます。

4.3.3 メイン関数 (main())

- (1) ローカル変数の初期化
 - ・受信データバッファポインタ (rx_cnt1)、前回受信データバッファポインタ (rx_cnt1_bak)、送信データ (send_data) を"0"に初期化
- (2) 割り込み禁止関数の呼び出し
- (3) 各種初期化関数の呼び出し
- (4) SIO0 送信処理初期化関数の呼び出し
- (5) SIO1 受信処理初期化関数の呼び出し
- (6) 割り込み許可関数の呼び出し
- (7) 送受信処理無限ループ
 - ・SIO0 から変数 send_data の値を 1 バイト送信
 - ・SIO1 の受信割り込み待ち
 - ・SIO1 受信データバッファから受信データの取り出し
 - ・SIO1 受信データバッファポインタがオーバーフローしているなら"0"に再設定
 - ・今回受信したデータバッファポインタを記憶
 - ・SIO0 送信データと SIO1 受信データが同じ値の場合
 - 1) 受信データを LED に表示
 - 2) 送信データをインクリメントして次の送信処理をおこなう
 - ・SIO0 送信データと SIO1 受信データが違う場合
 - 1) 無限ループに入り、LED に H'55 と H'AA を交互に表示する

4.3.4 SIO0 初期化関数 (SIO0_init())

(1) 転送モードの設定

- SIO0 受信制御レジスタの受信許可ビットを受信禁止に設定 (S0RCNT : REN)
- SIO0 送信制御レジスタの設定 (S0TCNT : CDIV、TEN)
BRG カウントソースを f(BCLK)、または f(BCLK)/2、送信禁止に設定
- P8 動作モードレジスタの設定 (P8MOD : P82MOD、P83MOD)
ポート P82 を TXD0、ポート P83 を RXD0 に設定
- SIO0 送受信モードレジスタの設定 (S0MOD : SMOD、CKS、STB、PEN、SEN)
8 ビット UART、内部クロック、1 ストップビット、パリティ禁止、スリープ機能無効に設定
- SIO0 ボーレートレジスタにボーレート分周値を設定 (S0BAUR)
- SIO0 特殊モードレジスタの設定 (S0SMOD : SELCLK、SELFST、SEL3PNT)
クロックデバイダカウントソースに f(BCLK)/2、転送順序を LSB ファースト、
3 ポイントサンプリング無効に設定

(2) 割り込み設定

- SIO0 送信割り込み制御レジスタを割り込み禁止に設定 (ISIO0TXCR : ILEVEL)
- SIO0 受信割り込み制御レジスタを 0 (最優先) に設定 (ISIO0RXCR : ILEVEL)
- SIO03 割り込み要求要因選択レジスタの SIO0 受信割り込み要求要因選択ビットを受信完了
割り込みに設定 (SI03SEL : ISR0)
- SIO03 割り込み要求マスクレジスタの設定 (SI03MASK : T0MASK、R0MASK)
SIO0 送信割り込み禁止、SIO0 受信割り込み許可に設定

(3) 送受信許可設定

- SIO0 送信制御レジスタの送信許可ビットを送信許可に設定 (S0TCNT : TEN)
- SIO0 受信制御レジスタの受信許可ビットを受信許可に設定 (S0RCNT : REN)

4.3.5 SIO1 初期化関数 (SIO1_init())

(1) 転送モードの設定

- SIO1 受信制御レジスタの受信許可ビットを受信禁止に設定 (S1RCNT : REN)
- SIO1 送信制御レジスタの設定 (S1TCNT : CDIV、TEN)
BRG カウントソースを f(BCLK)、または f(BCLK)/2、送信禁止に設定
- P8 動作モードレジスタの設定 (P8MOD : P85MOD、P86MOD)
ポート P85 を TXD1、ポート P86 を RXD1 に設定
- SIO1 送受信モードレジスタの設定 (S1MOD : SMOD、CKS、STB、PEN、SEN)
8 ビット UART、内部クロック、1 ストップビット、パリティ禁止、スリープ機能無効に設定
- SIO1 ボーレートレジスタにボーレート分周値を設定 (S1BAUR)
- SIO1 特殊モードレジスタの設定 (S1SMOD : SELCLK、SELFST、SEL3PNT)
クロックデバイダカウントソースに f(BCLK)/2、転送順序を LSB ファースト、
3 ポイントサンプリング無効に設定

(2) 割り込み設定

- SIO1 送信割り込み制御レジスタを割り込み禁止に設定 (ISIO1TXCR : ILEVEL)
- SIO1 受信割り込み制御レジスタを 0 (最優先) に設定 (ISIO1RXCR : ILEVEL)
- SIO03 割り込み要求要因選択レジスタの SIO1 受信割り込み要求要因選択ビットを受信完了
割り込みに設定 (SI03SEL : ISR1)
- SIO03 割り込み要求マスクレジスタの設定 (SI03MASK : T1MASK、R1MASK)
SIO1 送信割り込み禁止、SIO1 受信割り込み許可に設定

(3) 送受信許可設定

- SIO1 送信制御レジスタの送信許可ビットを送信許可に設定 (S1TCNT : TEN)
- SIO1 受信制御レジスタの受信許可ビットを受信許可に設定 (S1RCNT : REN)

4.3.6 SIO2 初期化関数 (SIO2_init())

SIO0_init と同様に設定しています。

本参考プログラムでは使用していません。

4.3.7 SIO3 初期化関数 (SIO3_init())

SIO0_init と同様に設定しています。

本参考プログラムでは使用していません。

4.3.8 UART 送信処理関数 (SIO0_Tr(), SIO1_Tr(), SIO2_Tr(), SIO3_Tr())

(1) 送信処理

- ・ SIO 送信バッファエンプティを確認 (S0TCNT : TBE)
- ・ 送信データバッファのデータを送信バッファレジスタに 1 バイト書き込み (S0TXB_L)
- ・ 指定回数送信を繰り返し
- ・ SIO 送信完了待ち (S0TCNT : TSTAT)

SIO1_Tr()、SIO2_Tr()、SIO3_Tr()は本参考プログラムでは使用していません。

4.3.9 SIO 受信割り込みによる UART 受信処理関数 (SIO0_RcvInt(), SIO1_RcvInt(), SIO2_RcvInt(), SIO3_RcvInt())

(1) 受信処理

- ・ SIO 受信制御レジスタ (受信ステータス) 読み出し (S1RCNT)
- ・ SIO 受信バッファレジスタ読み出し (S1RXB_L)
- ・ SIO 受信制御レジスタの再読み出し
受信ステータス読み出しと受信バッファレジスタ読み出しの間に受信するオーバランを検出するため
- ・ SIO 受信エラーサムビットを確認 (S1RCNT : ERS)
 - 1) エラーがない場合、受信データを受信データバッファに格納し、ポインタをインクリメント
 - 2) エラーがある場合、受信許可ビットを受信禁止にし、再度受信許可に設定する
(受信制御レジスタの各ビットがクリアされます)

SIO0_RcvInt()、SIO2_RcvInt()、SIO3_RcvInt()は本参考プログラムでは使用していません。

4.3.10 SIO2,3 の割り込みハンドラ関数 (IntHand_SIO23())

SIO2 と SIO3 割り込みはグループ割り込みとなっているため、どちらの割り込み要求が発生したのかを、SIO23 割り込み要求ステータスレジスタの割り込み要求ステータスビットで判断します。
(SI23STAT : IRQR2、IRQR3)

本参考プログラムでは使用していません。

4.3.11 スタートアップルーチン (startup.ms)

(1) 割り込みの設定

- ・ ICU ベクターテーブルの割り込み要因、SIO1 受信割り込み (H'0000 00CC 番地) に SIO1 受信割り込みルーチン (SIO1_RcvInt0) の先頭アドレスを設定する

4.4 参考プログラム例

下記に UART モードの送受信参考プログラムを示します。

尚、下記の参考プログラム例では、SFR 定義ファイルが必要です。最新の SFR 定義ファイルはホームページよりダウンロードできます。SFR 定義ファイル使用時は、お客様の環境に合わせてパスの設定をおこなってください。

4.4.1 uart_main.c

```

1  /*"FILE COMMENT"*****
2  *      M32R C Programming          Rev. 1.00
3  *      < Sample Program for 32192 >
4  *      < Serial I/O (UART) (main routine) >
5  *
6  *      Copyright (c) 2005 Renesas Technology Corporation
7  *      All Rights Reserved
8  *      *****/
9
10 /******/
11 /*      Include file                */
12 /******/
13
14 #include      ".\inc\sfr32192_pragma.h"
15
16 /******/
17 /*      Function prototype declaration */
18 /******/
19
20 void          main(void);          /* Main function */
21 void          init_func(void);     /* Initial setup function */
22 void          port_init(void);     /* Initialize port */
23
24 /******/
25 /*      Definition of external reference */
26 /******/
27
28 extern void   DisInt( void );      /* Interrupt disable function */
29 extern void   EnInt( void );      /* Interrupt enable function */
30
31 extern void   SIO0_init( void );   /* Initialize SIO0 */
32 extern void   SIO1_init( void );   /* Initialize SIO1 */
33 extern void   SIO0_Tr( UCHAR *TrBuf, ULONG TrNum); /* Transmit from SIO0 */
34 extern void   SIO1_Tr( UCHAR *TrBuf, ULONG TrNum); /* Transmit from SIO1 */
35
36 /******/
37 /*      Global variable                */
38 /******/
39
40 volatile UCHAR RcvBuf0[10];       /* Receive buffer */
41 volatile UCHAR RcvBuf1[10];       /* Receive buffer */
42 volatile UCHAR RcvBuf2[10];       /* Receive buffer */
43 volatile UCHAR RcvBuf3[10];       /* Receive buffer */
44 volatile UCHAR rx_cnt0;           /* Pointer to receive buffer */
45 volatile UCHAR rx_cnt1;           /* Pointer to receive buffer */
46 volatile UCHAR rx_cnt2;           /* Pointer to receive buffer */
47 volatile UCHAR rx_cnt3;           /* Pointer to receive buffer */
48
49 /*"FUNC COMMENT"*****
50 * Function name: init_func()
51 *-----
52 * Description : Call various initialization functions
53 *-----
54 * Argument   : -
55 *-----
56 * Returns    : -
57 *-----
58 * Notes      : -
59 *"FUNC COMMENT END"*****/
60 void init_func(void)
61 {
62     port_init();                  /* Initialize port */
63 }
    
```

```

64
65 /*"FUNC COMMENT"*****
66 * Function name: port_init()
67 *-----
68 * Description : Port initialization
69 *-----
70 * Argument : -
71 *-----
72 * Returns : -
73 *-----
74 * Notes : -
75 /*"FUNC COMMENT END"*****/
76 void port_init(void)
77 {
78     PICNT = PIEN0; /* Enable port input */
79
80 /** LED output port **/
81
82     P11DATA = 0x00; /* Output data (must be set prior to mode) */
83     P11DIR = 0xff; /* P110-P117 : Output mode */
84     P11MOD = 0x00; /* P110-P117 : Input/output port */
85 }
86
87 /*"FUNC COMMENT"*****
88 * Function name: main()
89 *-----
90 * Description : Receives data (increment pattern) from SIO1 after being transmitted in 500 Kbps clock-
synchronized
91 * : serial mode (when the source clock frequency = 10 MHz) from SIO0 and
92 * : if the received data is the same as transmitted, displays it on LED(port 11).
93 * : If unable to receive the same data, it displays 0x55 in blinking inverse video.
94 *-----
95 * Argument : -
96 *-----
97 * Returns : -
98 *-----
99 * Notes : -
100 /*"FUNC COMMENT END"*****/
101 void main(void)
102 {
103 /*
104 *
105 * (SIO0) | TXD0 | --+
106 * | | |
107 * | | |
108 * (SIO1) | RXD1 | <-+
109 * | | |
110 * (SIO1) | TXD1 | --+
111 * | | |
112 * | | | (not use)
113 * (SIO0) | RXD0 | <-+
114 * | | |
115 * +-----+
116 */
117
118     ULONG j;
119     UCHAR send_data;
120     UCHAR rcv_data;
121     UCHAR rx_cnt1_bak;
122
123     rx_cnt1 = 0;
124     rx_cnt1_bak = 0;
125
126     send_data = 0;
127
128     DisInt(); /* Disable interrupt */
129
130     init_func(); /* Initialize microcomputer */
131
132     SIO0_init(); /* Initialize SIO0 */
133     SIO1_init(); /* Initialize SIO1 */
134
135     EnInt(); /* Enable interrupt */
136
137     while(1) {
138         SIO0_Tr( &send_data, 1ul); /* Send data from SIO0 */
139
140         while( rx_cnt1 == rx_cnt1_bak){ /* Wait for data to receive */

```

```

141         ;
142     }
143     rcv_data = RcvBuf1[ rx_cnt1 - 1];          /* Read out received data */
144
145     if( rx_cnt1 == 10) {                      /* Check for receive counter overflow */
146         rx_cnt1 = 0;
147     }
148     rx_cnt1_bak = rx_cnt1;
149
150     if( rcv_data == send_data) {
151         P11DATA = rcv_data;                  /* Display received (transmitted) data */
152         send_data++;                          /* Increment transmit data */
153     } else {                                  /* Received data in error */
154         P11DATA = 0x55;
155         while(1) {
156             for( j = 0ul; j < 1000000ul; j++){ /* Wait */
157                 ;
158             }
159             P11DATA ^= 0xffu;                  /* Display 0x55 in blinking inverse video */
160         }
161     }
162 }
163 }
    
```

4.4.2 uart.c

```

1  /*"FILE COMMENT"*****
2  *      M32R C Programming          Rev. 1.00
3  *      < Sample Program for 32192 >
4  *      < Serial Interface (UART) >
5  *
6  *      Copyright (c) 2005 Renesas Technology Corporation
7  *      All Rights Reserved
8  *****/
9
10 *****/
11 /*      Include file                */
12 *****/
13
14 #include      ".\inc\sfr32192_pragma.h"
15
16 *****/
17 /*      Function prototype declaration      */
18 *****/
19
20 void          SIO0_init( void);          /* Initialize SIO0 */
21 void          SIO1_init( void);          /* Initialize SIO1 */
22 void          SIO2_init( void);          /* Initialize SIO2 */
23 void          SIO3_init( void);          /* Initialize SIO3 */
24 void          SIO0_Tr( UCHAR *TrBuf, ULONG TrNum); /* Transmit from SIO0 */
25 void          SIO1_Tr( UCHAR *TrBuf, ULONG TrNum); /* Transmit from SIO1 */
26 void          SIO2_Tr( UCHAR *TrBuf, ULONG TrNum); /* Transmit from SIO2 */
27 void          SIO3_Tr( UCHAR *TrBuf, ULONG TrNum); /* Transmit from SIO3 */
28 void          SIO0_RcvInt( void);        /* SIO0 receive interrupt */
29 void          SIO1_RcvInt( void);        /* SIO1 receive interrupt */
30 void          SIO2_RcvInt( void);        /* SIO2 receive interrupt */
31 void          SIO3_RcvInt( void);        /* SIO3 receive interrupt */
32 void          IntHand_SIO23( void);      /* SIO2,3 transmit/receive interrupt handler */
33
34 *****/
35 /*      Externally referenced variable      */
36 *****/
37
38 extern volatile UCHAR RcvBuf0[];        /* Receive buffer */
39 extern volatile UCHAR RcvBuf1[];        /* Receive buffer */
40 extern volatile UCHAR RcvBuf2[];        /* Receive buffer */
41 extern volatile UCHAR RcvBuf3[];        /* Receive buffer */
42 extern volatile UCHAR rx_cnt0;          /* Pointer to receive buffer */
43 extern volatile UCHAR rx_cnt1;          /* Pointer to receive buffer */
44 extern volatile UCHAR rx_cnt2;          /* Pointer to receive buffer */
45 extern volatile UCHAR rx_cnt3;          /* Pointer to receive buffer */
46
47 *****/
48 /*      Define macro                      */
49 *****/
50
51 /* Setting port operation mode */
52
53 #define P8MOD_SCI0      0x30u            /* 0123 4567 */
54 /* 0011 0000B P8 operation mode register */
55 /* ||| | ||+--- P87 */
56 /* ||| | |+---- P86 */
57 /* ||| | +----- P85 */
58 /* |||+----- RXD0 */
59 /* |+----- TXD0 */
60 /* ++----- don't care */
61
62
63 #define P8MOD_SCI1      0x06u            /* 0123 4567 */
64 /* 0000 0110B P8 operation mode register */
65 /* ||| | ||+--- P87 */
66 /* ||| | |+---- RXD1 */
67 /* ||| | |+---- TXD1 */
68 /* ||| | +----- P84 */
69 /* |||+----- P83 */
70 /* |+----- P82 */
71 /* ++----- don't care */
72
73
74 #define P17MOD_SCI2     0x0cu            /* 0123 4567 */
75 /* 0000 1100B P17 operation mode register */
76 /* ||| | ||+--- P177 */
77 /* ||| | |+---- P176 */

```

```

76          /* |||| |+----- RXD2          */
77          /* |||| +----- TXD2          */
78          /* |||+----- P173          */
79          /* ||+----- P172          */
80          /* ++----- don't care          */
81
82          /* 0123 4567          */
83 #define P17MOD_SCI3      0x03u          /* 0000 0011B P17 operation mode register */
84          /* |||| |||+---- RXD3          */
85          /* |||| ||+----- TXD3          */
86          /* |||| |+----- P175          */
87          /* |||| +----- P174          */
88          /* |||+----- P173          */
89          /* ||+----- P172          */
90          /* ++----- don't care          */
91
92 /* Setting serial IO */
93          /* 0123 4567          */
94 #define SnTCNT_INI      0x00u          /* 0000 0000B SIO0 transmit control register */
95          /* |||| |||+---- Disable transmission */
96          /* |||| +++---- don't care          */
97          /* ||+----- f(BCLK)          */
98          /* ++----- don't care          */
99
100         /* 0123 4567          */
101 #define SnMOD_INI      0x20u          /* 0010 0000B SIO0 mode register */
102         /* |||| |||+---- Sleep function disabled */
103         /* |||| ||+----- Parity inhibited */
104         /* |||| |+----- don't care(odd) */
105         /* |||| +----- 1 stop bit */
106         /* |||+----- Internal clock */
107         /* +++----- 8-bit UART */
108
109         /* 0123 4567          */
110 #define SnSMOD_INI     0x00u          /* 0000 0000B SIO0 special mode register */
111         /* |||| |||+---- transmit at SCLK falling edge */
112         /* |||| ||+----- 3-point sampling valid */
113         /* |||| |+----- LSB first */
114         /* |||| +----- f(BCLK)/2 */
115         /* +++----- don't care          */
116
117 /* Setting interrupt priority level */
118
119 #define SioILEVEL      0x0u          /* Serial IO transmit/receive interrupt priority level
*/
120 #define ILEVEL_7      0x7u          /* Interrupt Disable          */
121
122 /* Setting baud rate (Be sure to check actually set value when using) */
123
124 #define XIN            20          /* 20MHz */
125
126 #define BAUD_62_5      ((XIN / 2) * 2000000 / 16 / 62500 - 1) /* 62.5Kbps */
127 #define BAUD_15_6      ((XIN / 2) * 2000000 / 16 / 15600 - 1) /* 15.6Kbps */
128 #define BAUD_9615      ((XIN / 2) * 2000000 / 16 / 9615 - 1) /* 9615bps */
129
130
131 /*""FUNC COMMENT""*****
132 * Function name: SIO0_init()
133 *-----
134 * Description : Sets SIO0 for 8-bit UART
135 *             : - Reception by interrupt
136 *             : - Program transmission
137 *-----
138 * Argument   : -
139 *-----
140 * Returns    : -
141 *-----
142 * Notes      : Port input function must be enabled
143 *             : Must be executed while interrupts are disabled
144 *""FUNC COMMENT END""*****/
145 void SIO0_init( void)
146 {
147
148 /** Setting transfer mode */
149
150     SORCNT = 0x00;          /* Disable reception */
151     SOTCNT = SnTCNT_INI;   /* f(BCLK) and disable transmission */
152     P8MOD |= P8MOD_SCI0;   /* Set P8 for SCIO mode */

```

```

153     SOMOD = SnMOD_INI;                /* Set data format */
154     SOBAUR = BAUD_62_5;              /* Set baud rate */
155     SOSMOD = SnSMOD_INI;             /* Set Transmit/Receive clock polarity */
156
157 /*** Interrupt related settings ***/
158
159     ISIO0TXCR = ILEVEL_7;            /* Set SIO0 transmit interrupt priority level */
160     ISIO0RXCR = SioILEVEL;          /* Set SIO0 receive interrupt priority level */
161     SIO3SEL &= ~ISR0;                /* Select receive-finished interrupt */
162     SIO3MASK &= ~TOMASK;             /* Disable SIO0 transmit interrupt request */
163     SIO3MASK |= ROMASK;              /* Enable SIO0 receive interrupt request */
164
165 /*** Starting transmission/reception ***/
166
167     SOTCNT |= TEN;                    /* Enable transmission */
168     SORCNT |= REN;                    /* Enable reception */
169 }
170
171 /*****FUNC COMMENT*****/
172 * Function name: SIO1_init()
173 *-----
174 * Description  : Sets SIO1 for 8-bit UART
175 *               : - Reception by interrupt
176 *               : - Program transmission
177 *-----
178 * Argument    : -
179 *-----
180 * Returns     : -
181 *-----
182 * Notes       : Port input function must be enabled
183 *               : Must be executed while interrupts are disabled
184 /*****FUNC COMMENT END*****/
185 void SIO1_init( void)
186 {
187
188 /*** Setting transfer mode */
189
190     S1RCNT = 0x00;                    /* Disable reception */
191     S1TCNT = SnTCNT_INI;              /* f(BCLK) and disable transmission */
192     P8MOD |= P8MOD_SCI1;              /* Set P8 for SCI1 mode */
193     S1MOD = SnMOD_INI;                /* Set data format */
194     S1BAUR = BAUD_62_5;              /* Set baud rate */
195     S1SMOD = SnSMOD_INI;             /* Set Transmit/Receive clock polarity */
196
197 /*** Interrupt related settings ***/
198
199     ISIO1TXCR = ILEVEL_7;            /* Set SIO1 transmit interrupt priority level */
200     ISIO1RXCR = SioILEVEL;          /* Set SIO1 receive interrupt priority level */
201     SIO3SEL &= ~ISR1;                /* Select receive-finished interrupt */
202     SIO3MASK &= ~TIMASK;             /* Disable SIO1 transmit interrupt request */
203     SIO3MASK |= RIMASK;              /* Enable SIO1 receive interrupt request */
204
205 /*** Starting transmission/reception ***/
206
207     S1TCNT |= TEN;                    /* Enable transmission */
208     S1RCNT |= REN;                    /* Enable reception */
209 }
210
211 /*****FUNC COMMENT*****/
212 * Function name: SIO2_init()
213 *-----
214 * Description  : Sets SIO2 for 8-bit UART
215 *               : - Reception by interrupt
216 *               : - Program transmission
217 *-----
218 * Argument    : -
219 *-----
220 * Returns     : -
221 *-----
222 * Notes       : Port input function must be enabled
223 *               : Must be executed while interrupts are disabled
224 *               : Interrupt priority levels are common to both SIO2,3 transmission and reception
225 /*****FUNC COMMENT END*****/
226 void SIO2_init( void)
227 {
228
229 /*** Setting transfer mode */
230

```



```

231         S2RCNT = 0x00;                               /* Disable reception */
232         S2TCNT = SnTCNT_INI;                         /* f(BCLK) and disable transmission */
233         P17MOD |= P17MOD_SCI2;                      /* Set P17 for SCI2 mode */
234         S2MOD = SnMOD_INI;                          /* Set data format */
235         S2BAUR = BAUD_62_5;                         /* Set baud rate */
236         S2SMOD = SnSMOD_INI;                       /* Set Transmit/Receive clock polarity */
237
238     /*** Interrupt related settings ***/
239
240         ISIO23CR = SioILEVEL;                       /* Set SIO2,3 transmission and reception
interrupt priority level */
241         SIO3SEL &= ~ISR2;                          /* Select receive-finished interrupt */
242         SIO3MASK &= ~T2MASK;                       /* Disable SIO2 transmit interrupt request */
243         SIO3MASK |= R2MASK;                        /* Enable SIO2 receive interrupt request */
244         SI23STAT = (~( IRQT2 | IRQR2)) & 0x0fu;    /* Clear interrupt request */
245
246     /*** Starting transmission/reception ***/
247
248         S2TCNT |= TEN;                               /* Enable transmission */
249         S2RCNT |= REN;                               /* Enable reception */
250     }
251
252     /***FUNC COMMENT***/
253     * Function name: SIO3_init()
254     *-----
255     * Description  : Sets SIO3 for 8-bit UART
256     *               : - Reception by interrupt
257     *               : - Program transmission
258     *-----
259     * Argument    : -
260     *-----
261     * Returns     : -
262     *-----
263     * Notes       : Port input function must be enabled
264     *               : Must be executed while interrupts are disabled
265     *               : Interrupt priority levels are common to both SIO2,3 transmission and reception
266     /***FUNC COMMENT END***/
267 void SIO3_init( void)
268 {
269
270     /*** Setting transfer mode */
271
272         S3RCNT = 0x00;                               /* Disable reception */
273         S3TCNT = SnTCNT_INI;                         /* f(BCLK) and disable transmission */
274         P17MOD |= P17MOD_SCI3;                      /* Set P17 for SCI3 mode */
275         S3MOD = SnMOD_INI;                          /* Set data format */
276         S3BAUR = BAUD_62_5;                         /* Set baud rate */
277         S3SMOD = SnSMOD_INI;                       /* Set Transmit/Receive clock polarity */
278
279     /*** Interrupt related settings ***/
280
281         ISIO23CR = SioILEVEL;                       /* Set SIO2,3 transmission and reception
interrupt priority level */
282         SIO3SEL &= ~ISR3;                          /* Select receive-finished interrupt */
283         SIO3MASK &= ~T3MASK;                       /* Disable SIO3 transmit interrupt request */
284         SIO3MASK |= R3MASK;                        /* Enable SIO3 receive interrupt request */
285         SI23STAT = (~( IRQT3 | IRQR3)) & 0x0fu;    /* Clear interrupt request */
286
287     /*** Starting transmission/reception ***/
288
289         S3TCNT |= TEN;                               /* Enable transmission */
290         S3RCNT |= REN;                               /* Enable reception */
291     }
292
293     /***FUNC COMMENT***/
294     * Function name: SIO0_Tr()
295     *-----
296     * Description  : Transmits data from SIO0
297     *-----
298     * Argument    : unsigned char *TrBuf  Pointer to transmit data buffer
299     *               : unsigned int TrNum  Number of transmit bytes
300     *-----
301     * Returns     : -
302     *-----
303     * Notes       : Do not always need to wait for end of transmission
304     /***FUNC COMMENT END***/
305 void SIO0_Tr( UCHAR *TrBuf, ULONG TrNum)
306 {

```

```

307     ULONG   i;
308
309     for( i = 0ul; i < TrNum; i++) {
310         while(( S0TCNT & TBE) == 0u){           /* Wait until transmit buffer is empty */
311             ;
312         }
313         S0TXB_L = *TrBuf++;                       /* Transfer return data */
314     }
315     while(( S0TCNT & TSTAT) != 0u){             /* Wait for end of transmission */
316         ;
317     }
318 }
319
320 /*"FUNC COMMENT"*****
321 * Function name: SIO1_Tr()
322 *-----
323 * Description   : Transmits data from SIO1
324 *-----
325 * Argument      : unsigned char *TrBuf  Pointer to transmit data buffer
326 *                : unsigned int TrNum   Number of transmit bytes
327 *-----
328 * Returns       : -
329 *-----
330 * Notes         : Do not always need to wait for end of transmission
331 *"FUNC COMMENT END"*****/
332 void SIO1_Tr( UCHAR *TrBuf, ULONG TrNum)
333 {
334     ULONG   i;
335
336     for( i = 0ul; i < TrNum; i++) {
337         while(( S1TCNT & TBE) == 0u){           /* Wait until transmit buffer is empty */
338             ;
339         }
340         S1TXB_L = *TrBuf++;                       /* Transfer return data */
341     }
342     while(( S1TCNT & TSTAT) != 0u){             /* Wait for end of transmission */
343         ;
344     }
345 }
346
347 /*"FUNC COMMENT"*****
348 * Function name: SIO2_Tr()
349 *-----
350 * Description   : Transmits data from SIO2
351 *-----
352 * Argument      : unsigned char *TrBuf  Pointer to transmit data buffer
353 *                : unsigned int TrNum   Number of transmit bytes
354 *-----
355 * Returns       : -
356 *-----
357 * Notes         : Do not always need to wait for end of transmission
358 *"FUNC COMMENT END"*****/
359 void SIO2_Tr( UCHAR *TrBuf, ULONG TrNum)
360 {
361     ULONG   i;
362
363     for( i = 0ul; i < TrNum; i++) {
364         while(( S2TCNT & TBE) == 0u){           /* Wait until transmit buffer is empty */
365             ;
366         }
367         S2TXB_L = *TrBuf++;                       /* Transfer return data */
368     }
369     while(( S2TCNT & TSTAT) != 0u){             /* Wait for end of transmission */
370         ;
371     }
372 }
373
374 /*"FUNC COMMENT"*****
375 * Function name: SIO3_Tr()
376 *-----
377 * Description   : Transmits data from SIO3
378 *-----
379 * Argument      : unsigned char *TrBuf  Pointer to transmit data buffer
380 *                : unsigned int TrNum   Number of transmit bytes
381 *-----
382 * Returns       : -
383 *-----
384 * Notes         : Do not always need to wait for end of transmission

```

```

385  *""FUNC COMMENT END""*****/
386 void SIO3_Tr( UCHAR *TrBuf, ULONG TrNum)
387 {
388     ULONG i;
389
390     for( i = 0ul; i < TrNum; i++) {
391         while(( S3TCNT & TBE) == 0u){           /* Wait until transmit buffer is empty */
392             ;
393         }
394         S3TXB_L = *TrBuf++;                       /* Transfer return data */
395     }
396     while(( S3TCNT & TSTAT) != 0u){             /* Wait for end of transmission */
397         ;
398     }
399 }
400
401 /""FUNC COMMENT""*****
402 * Function name: SIO0_RcvInt()
403 *-----
404 * Description : Reads out received data
405 *-----
406 * Argument    : -
407 *-----
408 * Returns     : -
409 *-----
410 * Notes       : -
411 *""FUNC COMMENT END""*****/
412 void SIO0_RcvInt(void)
413 {
414     UCHAR data;
415     UCHAR status;
416
417     status = S0RCNT;                               /* Read receive status */
418     data = S0RXB_L;                                /* Read out received data */
419     status |= S0RCNT;                              /* Handle overrun */
420
421     if(( status & ERS) == 0u) {                   /* Check receive error sum */
422         RcvBuf0[ rx_cnt0++] = data;              /* Store received data */
423     } else {                                       /* Disable reception */
424
425         /* Error processing by user */
426
427         S0RCNT &= ~REN;                            /* Disable reception */
428         S0RCNT |= REN;                             /* Enable reception */
429     }
430 }
431 }
432
433 /""FUNC COMMENT""*****
434 * Function name: SIO1_RcvInt()
435 *-----
436 * Description : Reads out received data
437 *-----
438 * Argument    : -
439 *-----
440 * Returns     : -
441 *-----
442 * Notes       :
443 *""FUNC COMMENT END""*****/
444 void SIO1_RcvInt(void)
445 {
446     UCHAR data;
447     UCHAR status;
448
449     status = S1RCNT;                               /* Read receive status */
450     data = S1RXB_L;                                /* Read out received data */
451     status |= S1RCNT;                              /* Handle overrun */
452
453     if(( status & ERS) == 0u) {                   /* Check receive error sum */
454         RcvBuf1[ rx_cnt1++] = data;              /* Store received data */
455     } else {                                       /* Disable reception */
456
457         /* Error processing by user */
458
459         S1RCNT &= ~REN;                            /* Disable reception */
460         S1RCNT |= REN;                             /* Enable reception */
461     }
462 }

```

```

463 }
464
465 /*"FUNC COMMENT"*****
466 * Function name: SIO2_RcvInt()
467 *-----
468 * Description : Reads out received data
469 *-----
470 * Argument : -
471 *-----
472 * Returns : -
473 *-----
474 * Notes : -
475 *"FUNC COMMENT END"*****/
476 void SIO2_RcvInt(void)
477 {
478     UCHAR data;
479     UCHAR status;
480
481     status = S2RCNT; /* Read receive status */
482     data = S2RXB L; /* Read out received data */
483     status |= S2RCNT; /* Handle overrun */
484
485     if(( status & ERS) == 0u) { /* Check receive error sum */
486         RcvBuf2[ rx_cnt2++] = data; /* Store received data */
487     } else { /* Disable reception */
488
489         /* Error processing by user */
490
491         S2RCNT &= ~REN; /* Disable reception */
492         S2RCNT |= REN; /* Enable reception */
493     }
494 }
495 }
496
497 /*"FUNC COMMENT"*****
498 * Function name: SIO3_RcvInt()
499 *-----
500 * Description : Reads out received data
501 *-----
502 * Argument : -
503 *-----
504 * Returns : -
505 *-----
506 * Notes : -
507 *"FUNC COMMENT END"*****/
508 void SIO3_RcvInt(void)
509 {
510     UCHAR data;
511     UCHAR status;
512
513     status = S3RCNT; /* Read receive status */
514     data = S3RXB L; /* Read out received data */
515     status |= S3RCNT; /* Handle overrun */
516
517     if(( status & ERS) == 0u) { /* Check receive error sum */
518         RcvBuf3[ rx_cnt3++] = data; /* Store received data */
519     } else { /* Disable reception */
520
521         /* Error processing by user */
522
523         S3RCNT &= ~REN; /* Disable reception */
524         S3RCNT |= REN; /* Enable reception */
525     }
526 }
527 }
528
529 /*"FUNC COMMENT"*****
530 * Function name: IntHand_SIO23()
531 *-----
532 * Description : Processing by SIO2,3 transmit/receive interrupt handlers
533 *-----
534 * Argument : -
535 *-----
536 * Returns : -
537 *-----
538 * Notes : Transmit interrupt is not considering
539 *"FUNC COMMENT END"*****/
540 void IntHand_SIO23(void)

```

```

541 {
542     if(( SI23STAT & IRQR3) != 0u) {                               /* Judge interrupt request */
543         SI23STAT = (~IRQR3) & 0x0fu;                             /* Clear interrupt request status */
544         SIO3_RcvInt();                                           /* Process SIO3 receive interrupt */
545     }
546
547     if(( SI23STAT & IRQR2) != 0u) {                               /* Judge interrupt request */
548         SI23STAT = (~IRQR2) & 0x0fu;                             /* Clear interrupt request status */
549         SIO2_RcvInt();                                           /* Process SIO2 receive interrupt */
550     }
551 }
552

```

4.4.3 startup.ms (一部を抜粋)

(省略)

```

72 ;*****
73 ; ICU Vector Table
74 ;*****
75 ;
76     .SECTION          ICUVECT, DATA, ALIGN=4
77 ;
78     .IMPORT          $$SIO0_RcvInt, $$SIO1_RcvInt
79     .IMPORT          $IntHand_SIO23
80 ;
81 vectbl:
82     .DATA.W          EIT_reset ; H'0000 0094 MJT Input Interrupt 4:TIN3-TIN6
83     .DATA.W          EIT_reset ; H'0000 0098 MJT Input Interrupt 3:TIN20-TIN27
84     .DATA.W          EIT_reset ; H'0000 009C MJT Input Interrupt 2:TIN16-TIN19
85     .DATA.W          EIT_reset ; H'0000 00A0 MJT Input Interrupt 1:TIN0
86     .DATA.W          EIT_reset ; H'0000 00A4 MJT Input Interrupt 0:TIN7-TIN10
87     .DATA.W          EIT_reset ; H'0000 00A8 MJT Output Interrupt 7:TMS0,TMS1
88     .DATA.W          EIT_reset ; H'0000 00AC MJT Output Interrupt 6:TOP8, TOP9
89     .DATA.W          EIT_reset ; H'0000 00B0 MJT Output Interrupt 5:TOP10
90     .DATA.W          EIT_reset ; H'0000 00B4 MJT Output Interrupt 4:TIO4-TIO7
91     .DATA.W          EIT_reset ; H'0000 00B8 MJT Output Interrupt 3:TIO8,TIO9
92     .DATA.W          EIT_reset ; H'0000 00BC MJT Output Interrupt 2:TOP0-TOP5
93     .DATA.W          EIT_reset ; H'0000 00C0 MJT Output Interrupt 1:TOP6, TOP7
94     .DATA.W          EIT_reset ; H'0000 00C4 MJT Output Interrupt 0:TIO0-TIO3
95     .DATA.W          EIT_reset ; H'0000 00C8 DMAC0-4 Interrupt:DMA0-DMA4
96     .DATA.W          $$SIO1_RcvInt ; H'0000 00CC SIO1 Receive Interrupt
97     .DATA.W          EIT_reset ; H'0000 00D0 SIO1 Transmit Interrupt
98     .DATA.W          $$SIO0_RcvInt ; H'0000 00D4 SIO0 Receive Interrupt
99     .DATA.W          EIT_reset ; H'0000 00D8 SIO0 Transmit Interrupt
100    .DATA.W          EIT_reset ; H'0000 00DC A/D0 Conversion Interrupt
101    .DATA.W          EIT_reset ; H'0000 00E0 TID0 Output Interrupt
102    .DATA.W          EIT_reset ; H'0000 00E4 TOD0 Output Interrupt
103    .DATA.W          EIT_reset ; H'0000 00E8 DMAC5-9 Interrupt:DMA5-DMA9
104    .DATA.W          $IntHand_SIO23 ; H'0000 00EC SIO2,3 Transmit/Receive Interrupt
105    .DATA.W          EIT_reset ; H'0000 00F0 RTD Interrupt
106    .DATA.W          EIT_reset ; H'0000 00F4 TID1 Output Interrupt
107    .DATA.W          EIT_reset ; H'0000 00F8 TOU1 Output Interrupt:TOU1_0-TOU1_7
108    .DATA.W          EIT_reset ; H'0000 00FC SIO4,5 Transmit/Receive Interrupt
109    .DATA.W          EIT_reset ; H'0000 0100 Reserved
110    .DATA.W          EIT_reset ; H'0000 0104 Reserved
111    .DATA.W          EIT_reset ; H'0000 0108 TML1 Input Interrupt:TIN30-TIN33
112    .DATA.W          EIT_reset ; H'0000 010C CAN0 Transmit/Receive & Error Interrupt
113    .DATA.W          EIT_reset ; H'0000 0110 CAN1 Transmit/Receive & Error Interrupt
114    .DATA.W          EIT_reset ; H'0000 0114 DRI Transfer Interrupt
115    .DATA.W          EIT_reset ; H'0000 0118 DRI Counter Interrupt:DEC0-DEC4
116    .DATA.W          EIT_reset ; H'0000 011C DRI Event Detection Interrupt:DIN0-DIN5
117    .DATA.W          EIT_reset ; H'0000 0120 CAN0 Transmit/Receive Completion
Interrupt
118    .DATA.W          EIT_reset ; H'0000 0124 CAN0 Single-Shot Interrupt
119    .DATA.W          EIT_reset ; H'0000 0128 CAN0 Error Interrupt
120    .DATA.W          EIT_reset ; H'0000 012C CAN1 Transmit/Receive Completion
Interrupt
121    .DATA.W          EIT_reset ; H'0000 0130 CAN1 Single-Shot Interrupt
122    .DATA.W          EIT_reset ; H'0000 0134 CAN1 Error Interrupt
123    .DATA.W          EIT_reset ; H'0000 0138 RAM Write Monitor Interrupt
124 ;

```

(以下省略)

4.5 UART の動作タイミング

下記に UART モードの送信、受信タイミング図を示します。

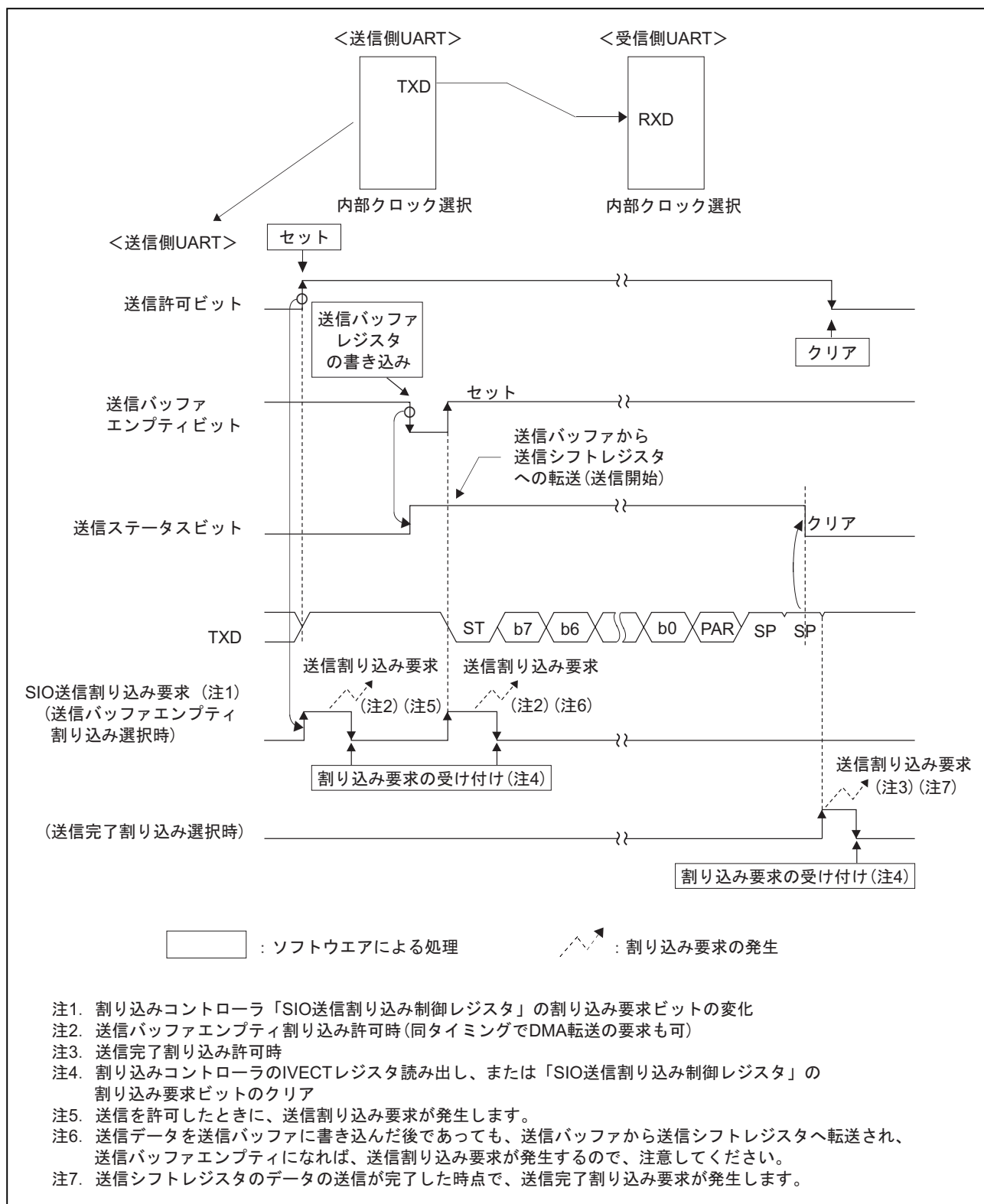


図 4.5.1 UART モードの送信タイミング図

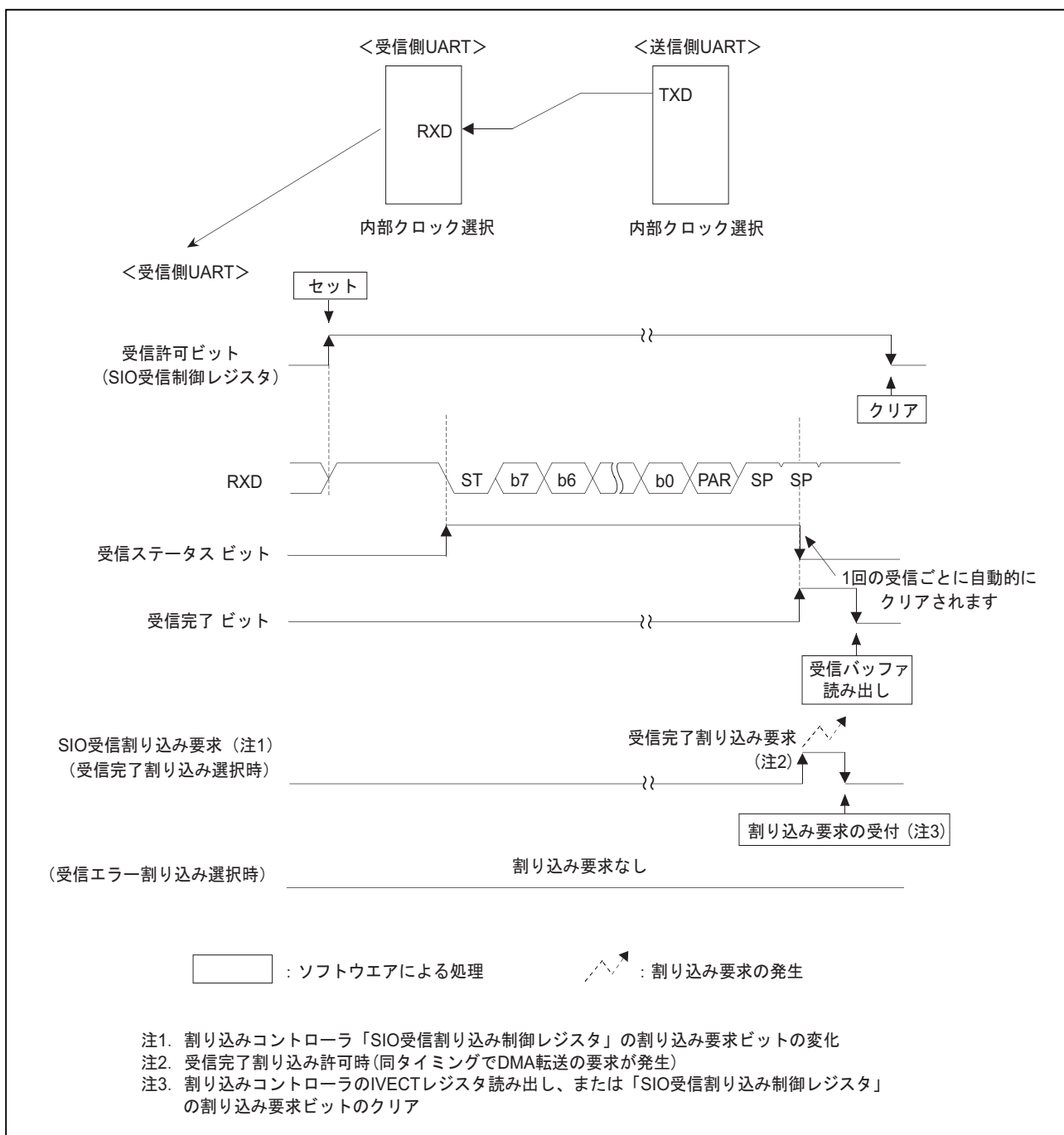


図 4.5.2 UART モードの受信タイミング図

5. 参考ドキュメント

- 32192 グループ データシート Rev.1.01
- 32196 グループ データシート Rev.1.00
- 32192/32196 グループ ハードウェアマニュアル Rev.1.00
- M3T-CC32R V.4.30 ユーザーズマニュアル (C コンパイラ編)
- M3T-AS32R V.4.30 ユーザーズマニュアル (アセンブラ編)
- M32R-FPU ソフトウェアマニュアル Rev.1.01
(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。)

6. 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://www.renesas.com/>

ルネサス製品全般に関するお問合せと M32R ファミリに関する技術的なお問合せ先
カスタマサポートセンタ : csc@renesas.com

改訂記録	32192/32196 グループ シリアルインタフェースを使用した応用例（UART モード） アプリケーションノート
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.06.29	-	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すは、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。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、アルゴリズム単体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を負いません。
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。