

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（CSIO 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の概要

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。

CSIO モードで内部クロック使用時の転送速度は、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	実際のボーレート [bps]
250	256	155	250.40
500	256	77	500.80
1000	256	38	1001.60
2500	32	124	2500.00
5000	8	249	5000.00
10000	8	124	10000.00
25000	8	49	25000.00
50000	1	199	50000.00
100000	1	99	100000.00
250000	1	39	250000.00
500000	1	19	500000.00
1000000	1	9	1000000.00
2500000	1	3	2500000.00
5000000	1	1	5000000.00

CSIO 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。

$$\text{BRG 設定値} = \frac{\text{クロックデバイダカウントソース}}{\text{ボーレート [bps]} \times \text{クロックデバイダ分周値} \times 2} - 1$$

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 を内部クロックの送信、SIO1 を外部クロックの受信に設定し通信を行います。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、単体での CSIO 送受信が行えます。

本参考プログラムでは、通信速度を 500kbps に設定しています。なお、内部クロック選択時は  $f(\text{BCLK})/8$ 、外部クロック選択時は  $f(\text{BCLK})/16$  が最大通信速度となります。

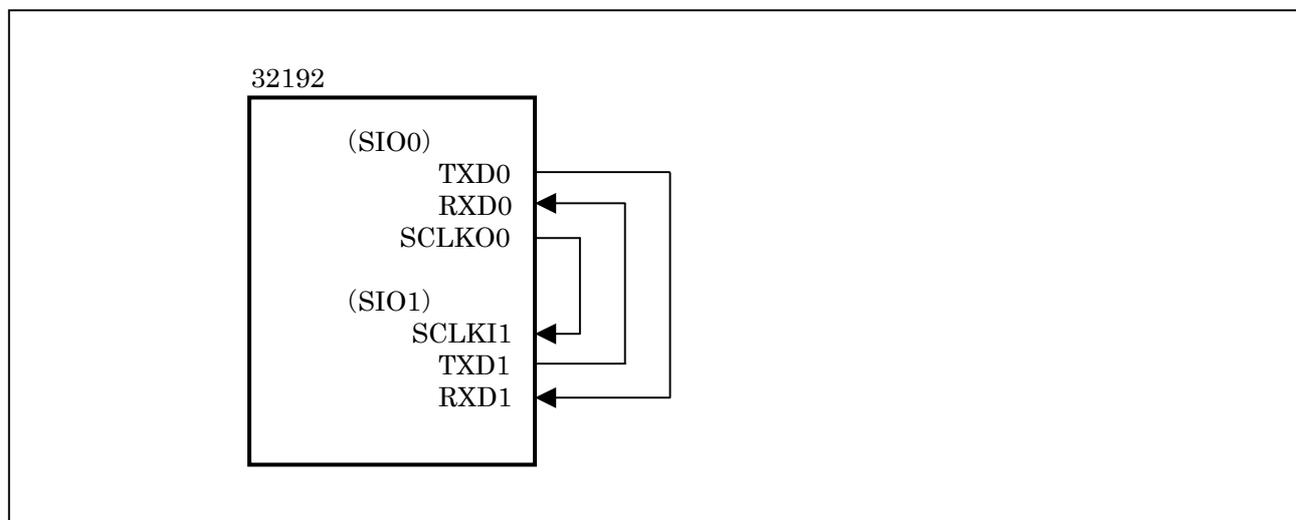


図 4.1.1 結線図

## 4.2 CSIO 送受信処理

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

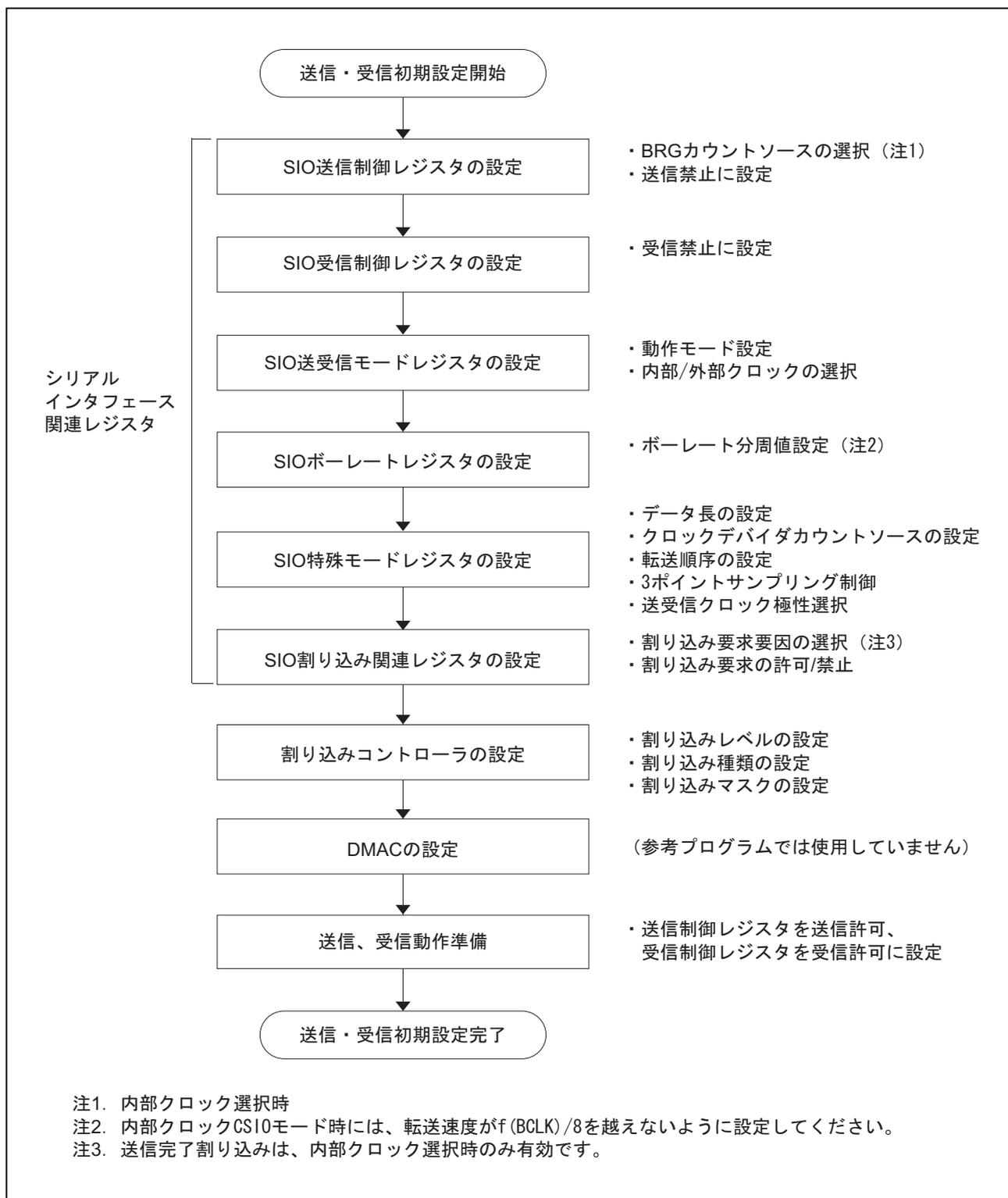


図 4.2.1 CSIO モード送受信初期設定フロー図

### 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 送信処理初期設定関数の呼び出し
- (5) SIO1 受信処理初期設定関数の呼び出し
- (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
  - ・SIO1（受信側）の送信バッファレジスタにダミー書き込み  
（CSIO モードでは受信シフトクロックは送信回路の動作によって得られますので、受信だけを行う場合でも送信動作を実行させる必要があります）
  - ・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
  - ・変数 rx\_cnt1 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 の CSIO 送信の初期設定関数 (SIO0\_CLKTinit())

##### (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、P84MOD)  
ポート P82 を TXD0、ポート P84 を SCLKI0/SCLKO0 に設定
- SIO0 送受信モードレジスタを CSIO モード、内部クロックに設定 (S0MOD : SMOD、CKS)
- SIO0 ボーレートレジスタのボーレート分周値を設定 (S0BAUR)  
(500kbps@CPU クロック=160MHz 時)
- SIO0 特殊モードレジスタの設定 (S0SMOD : CSIBL、SELCLK、SELFST、SEL3PNT、CKPOL)  
CSIO ビット長を 8 ビット、クロックデバイダカウントソースを 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、送受信クロック極性を SCLK の立ち下がりで送信データ出力、立ち上がりで受信データ取り込みに設定

##### (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割り込み禁止に設定 (ISIO0RXCR : ILEVEL)
- SIO03 割り込み要求マスクレジスタの設定 (SIO03MASK : T0MASK、R0MASK)  
SIO0 送信割り込み禁止、SIO0 受信割り込み禁止に設定

##### (3) 送信許可設定

- 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)

#### 4.3.5 SIO1 の CSIO 受信の初期設定関数 (SIO1\_CLKRinit())

##### (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 : P86MOD、P87MOD)  
ポート P86 を RXD1、ポート P87 を SCLKI1/SCLKO1 に設定
- SIO1 送受信モードレジスタを CSIO モード、外部クロックに設定 (S1MOD : SMOD、CKS)
- SIO1 特殊モードレジスタの設定 (S1SMOD : CSIBL、SELCLK、SELFST、SEL3PNT、CKPOL)  
CSIO ビット長を 8 ビット、クロックデバイダカウントソースを 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、送受信クロック極性を SCLK の立ち下がりで送信データ出力、立ち上がりで受信データ取り込みに設定

##### (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 (SIO03SEL : ISR1)
- SIO03 割り込み要求マスクレジスタの設定 (SIO03MASK : 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 SIO0 の CSIO 送信処理関数（SIO0\_Tr()）

##### (1) 送信処理

- ・ SIO0 送信バッファエンプティを確認（S0TCNT : TBE）
- ・ 送信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
- ・ SIO0 送信完了待ち（S0TCNT : TSTAT）

#### 4.3.7 SIO1 受信割り込みによる CSIO 受信処理関数（SIO1\_RcvInt()）

##### (1) 受信処理

- ・ SIO1 受信制御レジスタ（受信ステータス）読み出し（S1RCNT）
- ・ SIO1 受信バッファレジスタ読み出し（S1RXB\_L）
- ・ SIO1 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発生するオーバーランを検出するため
- ・ SIO1 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)

#### 4.3.8 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 CSIO 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 csio\_main.c

```

1  /*"FILE COMMENT"*****
2  *      M32R C Programming          Rev. 1.00
3  *      < Sample Program for 32192 >
4  *      < Serial Interface (CSIO) (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 /*      Externally referenced variable */
26 /******/
27
28 extern void   DisInt( void );     /* Interrupt disable function */
29 extern void   EnInt( void );     /* Interrupt enable function */
30
31 extern void   SIO0_CLKTinit( void); /* Initialize SIO0 */
32 extern void   SIO1_CLKRinit( void); /* Initialize SIO1 */
33 extern void   SIO0_Tr( UCHAR *, ULONG); /* Transmit from SIO0 */
34
35 /******/
36 /*      Global variable                */
37 /******/
38
39 volatile UCHAR RcvBuf1[10];     /* Receive buffer */
40 volatile UCHAR rx_cnt1;        /* Pointer to receive buffer */
41
42 /*"FUNC COMMENT"*****
43 * Function name: init_func()
44 *-----
45 * Description : Call various initialization functions
46 *-----
47 * Argument   : -
48 *-----
49 * Returns    : -
50 *-----
51 * Notes      :
52 *"FUNC COMMENT END"*****
53 void init_func(void)
54 {
55     port_init();                /* Initialize port */
56 }
57
58 /*"FUNC COMMENT"*****
59 * Function name: port_init()
60 *-----
61 * Description : Initialize port
62 *-----
63 * Argument   : -

```

```

64 *-----
65 * Returns      : -
66 *-----
67 * Notes       :
68 *""FUNC COMMENT END""*****/
69 void port_init(void)
70 {
71     PICNT = PIENO;                               /* Enable port input */
72
73     /*** LED output port ***/
74
75     P11DATA = 0x00;                               /* Output data (must be set prior to mode) */
76     P11DIR  = 0xff;                               /* P110-P117 : Output mode */
77     P11MOD  = 0x00;                               /* P110-P117 : Input/output port */
78 }
79
80 /""FUNC COMMENT""*****
81 * Function name: main()
82 *-----
83 * Description  : Input/output portReceives data (increment pattern) from SIO1 after being transmitted in 500
Kbps
84 *              : clock-synchronized serial mode (when the source clock frequency = 10 MHz) from SIO0 and
85 *              : if the received data is the same as transmitted, displays it on LED(port 11).
86 *              : If unable to receive the same data, it displays 0x55 in blinking inverse video.
87 *-----
88 * Argument    : -
89 *-----
90 * Returns     : -
91 *-----
92 * Notes       : Dummy transmission is required for data to be received from SIO1
93 *""FUNC COMMENT END""*****/
94 void main(void)
95 {
96     /*
97     *      +-----+
98     *      | TXD0 | ---+
99     * (SIO0) | RXD0 | <-|----+
100    *      | SCLK00| -----|---+
101    *      |      | | | | |
102    *      |      | | | | |
103    *      | TXD1 | --|----+ |
104    * (SIO1) | RXD1 | <-+   |
105    *      | SCLK11| <-----+
106    *      |      | | | | |
107    *      |      | | | | |
108    *      +-----+
109    */
110
111     ULONG j;
112     UCHAR send_data;
113     UCHAR rcv_data;
114     UCHAR rx_cnt1_bak;
115
116     rx_cnt1 = 0;
117     rx_cnt1_bak = 0;
118
119     send_data = 0;
120
121     DisInt();                                     /* Disable interrupt */
122
123     init_func();                                 /* Initialize microcomputer */
124
125     SIO0_CLKTinit();                             /* Initialize SIO0 */
126     SIO1_CLKRinit();                             /* Initialize SIO1 */
127
128     EnInt();                                     /* Enable interrupt */
129
130     while(1) {
131         S1TXB_L = rx_cnt1;                       /* Dummy write */
132         SIO0_Tr( &send_data, 1ul);             /* Send data from SIO0 */
133
134         while( rx_cnt1 == rx_cnt1_bak){         /* Wait for data to receive */
135             ;
136         }
137         rcv_data = RcvBuf1[ rx_cnt1 - 1];       /* Read out received data */
138
139         if( rx_cnt1 == 10) {                    /* Check for receive counter overflow */
140             rx_cnt1 = 0;

```

```

141     }
142     rx_cnt1_bak = rx_cnt1;
143
144     if( rcv_data == send_data) {
145         P11DATA = rcv_data;           /* Display received (transmitted) data */
146         send_data++;                 /* Increment transmit data */
147     } else {                         /* Received data in error */
148         P11DATA = 0x55;
149         while(1) {
150             for( j = 0ul; j < 1000000ul; j++){ /* Wait */
151                 ;
152             }
153             P11DATA ^= 0xffu;         /* Display 0x55 in blinking inverse video */
154         }
155     }
156 }
157 }

```

## 4.4.2 csio.c

```

1  /*"FILE COMMENT"*****
2  *      M32R C Programming          Rev. 1.00
3  *      < Sample Program for 32192 >
4  *      < Serial Interface (CSIO) >
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_CLKTinit(void);          /* Initialize SIO0 */
21 void          SIO1_CLKRinit(void);         /* Initialize SIO1 */
22 void          SIO0_Tr( UCHAR *TrBuf, ULONG TrNum); /* Send data from SIO0 */
23 void          SIO1_RcvInt(void);          /* SIO1 receive interrupt */
24
25 /******/
26 /*      Externally referenced variable */
27 /******/
28
29 extern volatile UCHAR RcvBuf1[];          /* Receive buffer */
30 extern volatile UCHAR rx_cnt1;           /* Pointer to receive buffer */
31
32 /******/
33 /*      Define macro                */
34 /******/
35
36 /* Setting port operation mode */
37
38 #define P8MOD_SCI0ClkTr          0x28u          /* 0123 4567 */
39                                     /* 0010 1000B P8 operation mode register */
40                                     /* |||| |||+---- P87 */
41                                     /* |||| ||+---- P86 */
42                                     /* |||| |+----- P85 */
43                                     /* |||| +----- SCLKI0/SCLKO0 */
44                                     /* |||+----- P83 */
45                                     /* ||+----- TXD0 */
46                                     /* ++----- don't care */
47
48 #define P8MOD_SCI1ClkRcv          0x03u          /* 0123 4567 */
49                                     /* 0000 0011B P8 operation mode register */
50                                     /* |||| |||+---- SCLKI1/SCLKO1 */
51                                     /* |||| ||+---- RXD1 */
52                                     /* |||| |+----- P85 */
53                                     /* |||| +----- P84 */
54                                     /* |||+----- P83 */
55                                     /* ||+----- P82 */
56                                     /* ++----- don't care */
57
58 /* Setting serial I/O */
59
60 #define SnTCNT_INI                0x00u          /* 0123 4567 */
61                                     /* 0000 0000B SIOn transmit control register */
62                                     /* |||| |||+---- Disable transmission */
63                                     /* |||| +++----- don't care */
64                                     /* ||+----- f(BCLK)/2 */
65                                     /* ++----- don't care */
66
67 #define SnMOD_CLK_INI              0x80u          /* 0123 4567 */
68                                     /* 1000 0000B SIOn mode register */
69                                     /* |||| +++----- don't care ( UART mode data ) */
70                                     /* |||+----- Select clock(specified separately) */
71                                     /* +++----- CSIO */
72
73 #define SnSMOD_INI                 0x00u          /* 0123 4567 */
74                                     /* 0000 0000B SIOn special mode register */
75                                     /* |||| |||+---- transmit at SCLK falling edge */
    
```

```

75                                     /* |||| |+- 3-point sampling valid */
76                                     /* |||| |+- LSB first */
77                                     /* |||| +----- f(BCLK)/2 */
78                                     /* +----- 8-bit CSIO */
79
80 /* Setting interrupt priority level */
81
82 #define SioILEVEL      0x0           /* Serial IO transmit/receive interrupt priority level */
83 #define ILEVEL_7      0x7u         /* Interrupt Disable */
84
85 /* Setting baud rate (Be sure to check actually set value when using) */
86
87 #define XIN            20            /* 20MHz */
88 #define ClkBAUD_500    ((XIN / 2) * 2000000 / 2 / 500000 - 1) /* 500Kbps */
89
90 /*"FUNC COMMENT"*****
91 * Function name: SIO0_CLKTinit()
92 *-----
93 * Description : Sets SIO0 for synchronous serial I/O transmission
94 *             : - Program transmission
95 *             : - Internal clock selected (clock output)
96 *-----
97 * Argument   : -
98 *-----
99 * Returns    : -
100 *-----
101 * Notes      : Must be executed while interrupts are disabled
102 *"FUNC COMMENT END"*****/
103 void SIO0_CLKTinit( void)
104 {
105
106 /*** Setting transfer mode ***/
107
108     SORCNT = 0x00;                /* Disable reception */
109     SOTCNT = SnTCNT_INI;          /* f(BCLK), disable transmission */
110     P8MOD |= P8MOD_SCI0ClkTr;    /* Set P8 for CSIO mode */
111     SOMOD = SnMOD_CLK_INI;        /* Set data format */
112     SOBAUR = ClkBAUD_500;        /* Set baud rate */
113     SOSMOD = SnSMOD_INI;         /* Set Transmit/Receive clock polarity */
114
115 /*** Interrupt related settings ***/
116
117     ISIO0TXCR = ILEVEL_7;        /* Set SIO0 transmit interrupt priority level */
118     ISIO0RXCR = ILEVEL_7;        /* Set SIO0 receive interrupt priority level */
119     SIO3MASK &= ~T0MASK;        /* Disable SIO0 transmit interrupt request */
120     SIO3MASK &= ~R0MASK;        /* Disable SIO0 receive interrupt request */
121
122 /*** Starting transmission/reception ***/
123
124     SOTCNT |= TEN;               /* Enable transmission */
125 }
126
127 /*"FUNC COMMENT"*****
128 * Function name: SIO1_CLKRinit()
129 *-----
130 * Description : Enable transmissionSets SIO1 for synchronous serial I/O reception
131 *             : - Reception by interrupt
132 *             : - External clock selected (clock input)
133 *-----
134 * Argument   : -
135 *-----
136 * Returns    : -
137 *-----
138 * Notes      : Port input function must be enabled
139 *             : To receive, it is necessary to enable transmission and write dummy data to transmit buffer
140 *             : Must be executed while interrupts are disabled
141 *"FUNC COMMENT END"*****/
142 void SIO1_CLKRinit( void)
143 {
144
145 /*** Setting transfer mode ***/
146
147     S1RCNT = 0x00;                /* Disable reception */
148     S1TCNT = SnTCNT_INI;          /* f(BCLK), disable transmission */
149     P8MOD |= P8MOD_SCI1ClkRcv;    /* Set P8 for CSIO mode */
150     S1MOD = SnMOD_CLK_INI | CKS;  /* Set data format */
151     S1SMOD = SnSMOD_INI;         /* Set Transmit/Receive clock polarity */

```

```

152
153 /*** Interrupt related settings ***/
154
155     ISIO1TXCR = ILEVEL_7;                /* Set SIO1 transmit interrupt priority level */
156     ISIO1RXCR = SioILEVEL;              /* Set SIO1 receive interrupt priority level */
157     SIO3SEL &= ~ISR1;                   /* Select reception-finished interrupt */
158     SIO3MASK &= ~T1MASK;                /* Disable SIO1 transmit interrupt request */
159     SIO3MASK |= R1MASK;                 /* Disable SIO1 receive interrupt request */
160
161 /*** Starting transmission/reception ***/
162
163     S1TCNT |= TEN;                       /* Enable transmission */
164     S1RCNT |= REN;                       /* Enable reception */
165 }
166
167 /*"FUNC COMMENT"*****
168 * Function name: SIO0_Tr()
169 *-----
170 * Description  : Transmits data from SIO0
171 *-----
172 * Argument    : unsigned char *TrBuf  Pointer to transmit data buffer
173 *              : unsigned long TrNum  Number of transmit bytes
174 *-----
175 * Returns     : -
176 *-----
177 * Notes       : Do not always need to wait for end of transmission
178 *"FUNC COMMENT END"*****/
179 void SIO0_Tr( UCHAR *TrBuf, ULONG TrNum)
180 {
181     ULONG  i;
182
183     for( i = 0ul; i < TrNum; i++) {
184         while(( SOTCNT & TBE) == 0u){    /* Wait until transmit buffer is empty */
185             ;
186         }
187         SOTXB_L = *TrBuf++;              /* Transfer return data */
188     }
189     while(( SOTCNT & TSTAT) != 0u){     /* Wait for end of transmission */
190         ;
191     }
192 }
193
194 /*"FUNC COMMENT"*****
195 * Function name: SIO1_RcvInt()
196 *-----
197 * Description  : Reads out received data
198 *-----
199 * Argument    : -
200 *-----
201 * Returns     : -
202 *-----
203 * Notes       :
204 *"FUNC COMMENT END"*****/
205 void SIO1_RcvInt(void)
206 {
207     UCHAR  data;
208     UCHAR  status;
209
210     status = S1RCNT;                     /* Read receive status */
211     data = S1RXB_L;                       /* Read out received data */
212     status |= S1RCNT;                     /* Handle overrun */
213
214     if(( status & ERS) == 0u) {           /* Check receive error sum */
215         RcvBuf1[ rx_cntl++] = data;      /* Store received data */
216     } else {                              /* Process receive errors */
217
218         /* Error processing by user */
219
220         S1RCNT &= ~REN;                   /* Disable reception */
221         S1RCNT |= REN;                   /* Enable reception */
222     }
223 }
224 }
225

```

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

```

（以下省略）

### 4.5 シリアルインタフェースの動作タイミング

下記に CSIO モードの送信、受信タイミング図を示します。

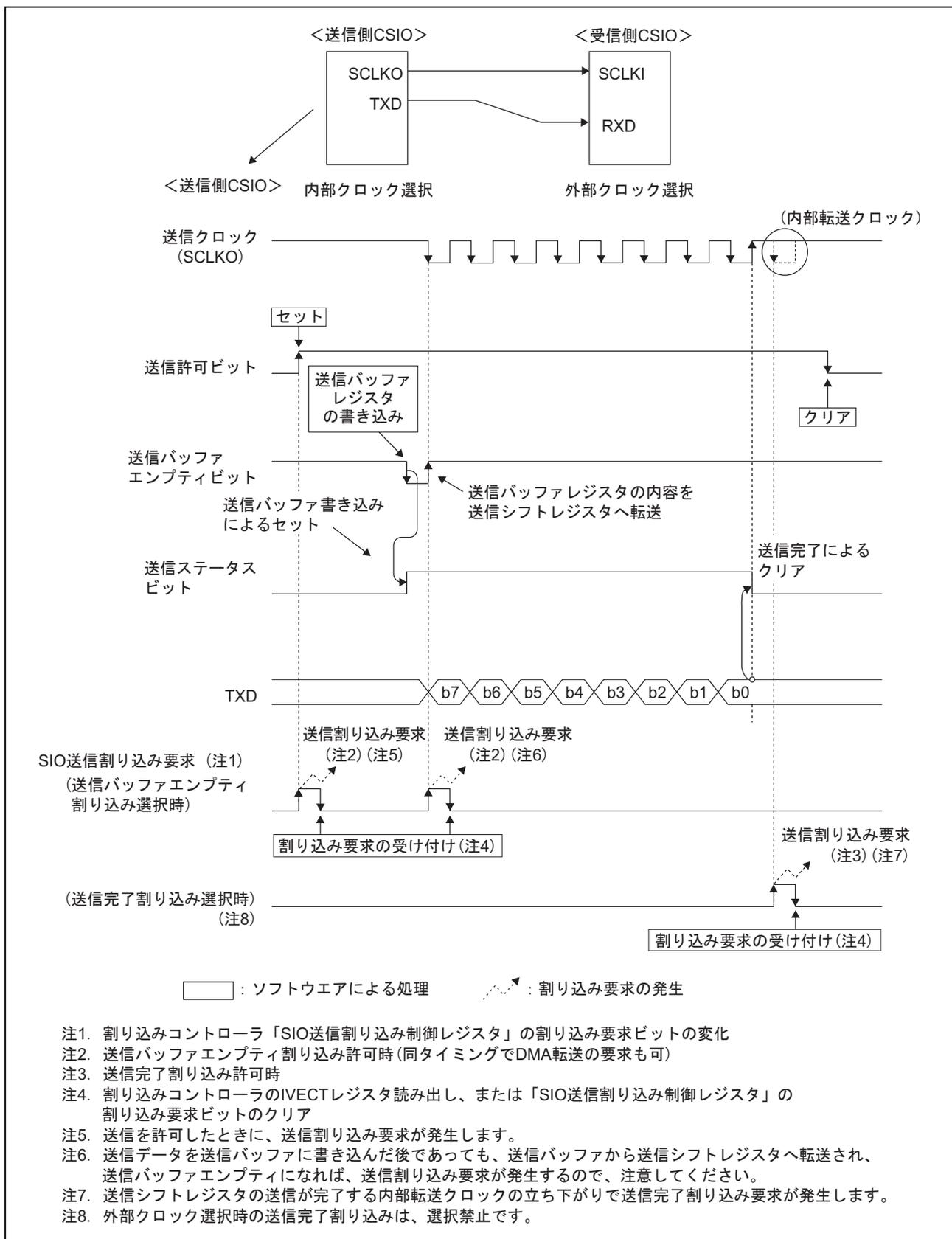


図 4.5.1 CSIO モードの送信タイミング図

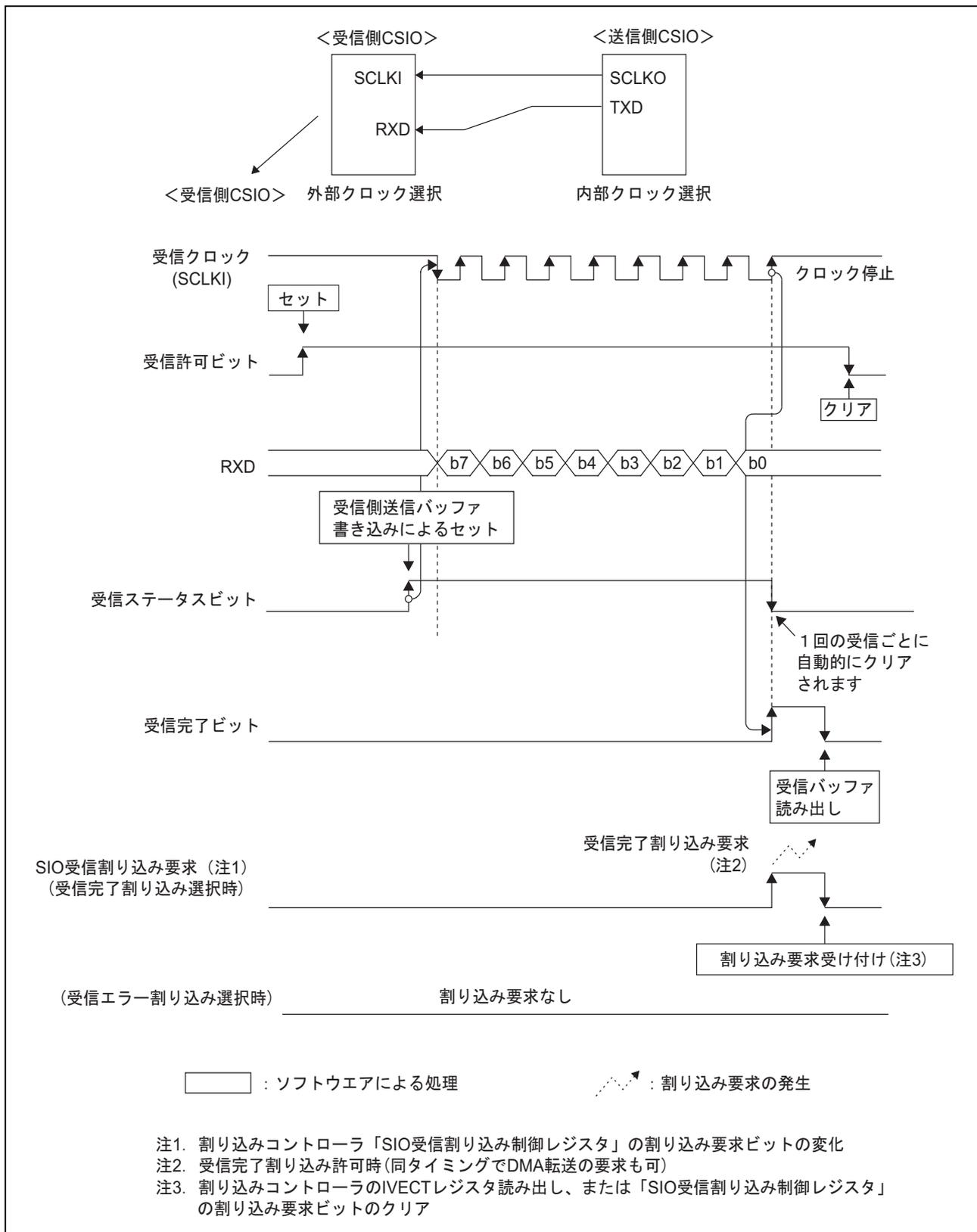


図 4.5.2 CSIO モードの受信タイミング図

## 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](mailto: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（CSIO 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。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。