

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。

SH7145 グループ

SCI のブレーク検出

要旨

SCI(Serial Communication Interface)モジュールを用いた調歩同期式通信時のブレーク検出について述べます。

動作確認デバイス

SH7145F

目次

1. 仕様	2
2. 使用機能説明	3
3. 動作説明	6
4. ソフトウェア説明	7
5. フローチャート	10
6. プログラムリスト	15

1. 仕様

図 1 に示すように、SH7145F の SCI モジュールのチャンネル 0 (ch0) を用いて、ブレークの検出を行います。ブレークの検出は、調歩同期式通信時のみ可能です。SH7145 では、ブレークの検出はハードウェアでは行われませんので、ソフトウェアにより実行します。

ブレークの検出は、フレーミングエラーの際に RxD 端子のレベルの確認により行われ、RxD 端子がローレベルのときブレークとなります。本タスク例では、3 バイトのデータの受信後、ブレークの検出を行います。フレーミングエラー時の RxD 端子レベルの確認を 5msec 間隔で 3 回行い、全てローレベルであるときブレークと認識します。ブレークを検出すると、SCR_0 の RE ビットを 0 にクリアして、SCI の受信動作を終了します。RxD 端子の確認は、コンペアマッチタイマ割り込みを用いて行います。

SCI の通信フォーマットは、19200bps、8 ビット、1 ストップビット、ノンパリティです。

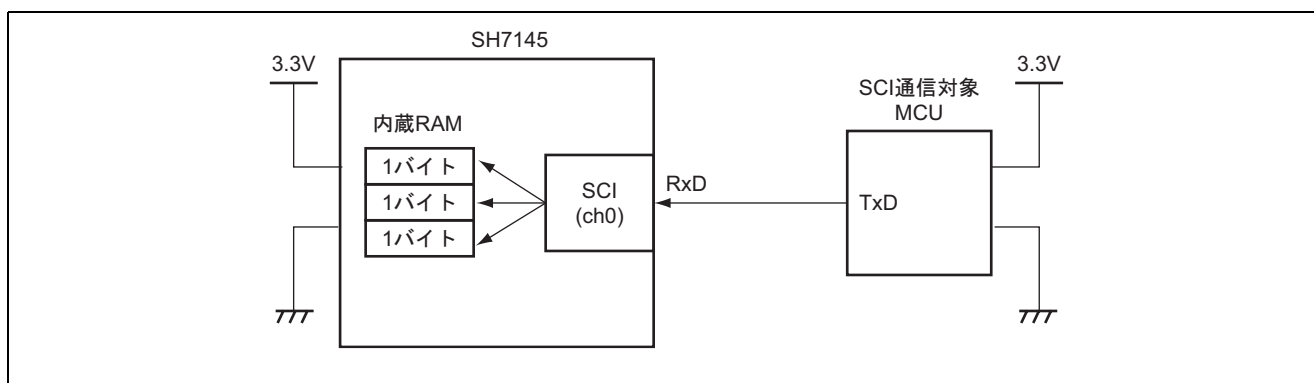


図 1 SH7145 SCI 受信時の接続図

表 1 調歩同期式シリアル送信フォーマット

フォーマット内容	設定
ビットレート	19200bps
データ長	8 ビット
パリティビット	なし
ストップビット	1 ビット
シリアル／パラレル変換フォーマット	LSB ファースト

2. 使用機能説明

本タスク例では、SCI (Serial Communication Interface) と CMT (Compare Match Timer) を使用します。

2.1 SCI (Serial Communication Interface)

本タスク例では、SCI を用いて、調歩同期式シリアルデータ送信を行います。図 2 に SCI モジュールチャネル 0 (ch0) のブロック図を示し、以下に図 2 についての機能説明をします。

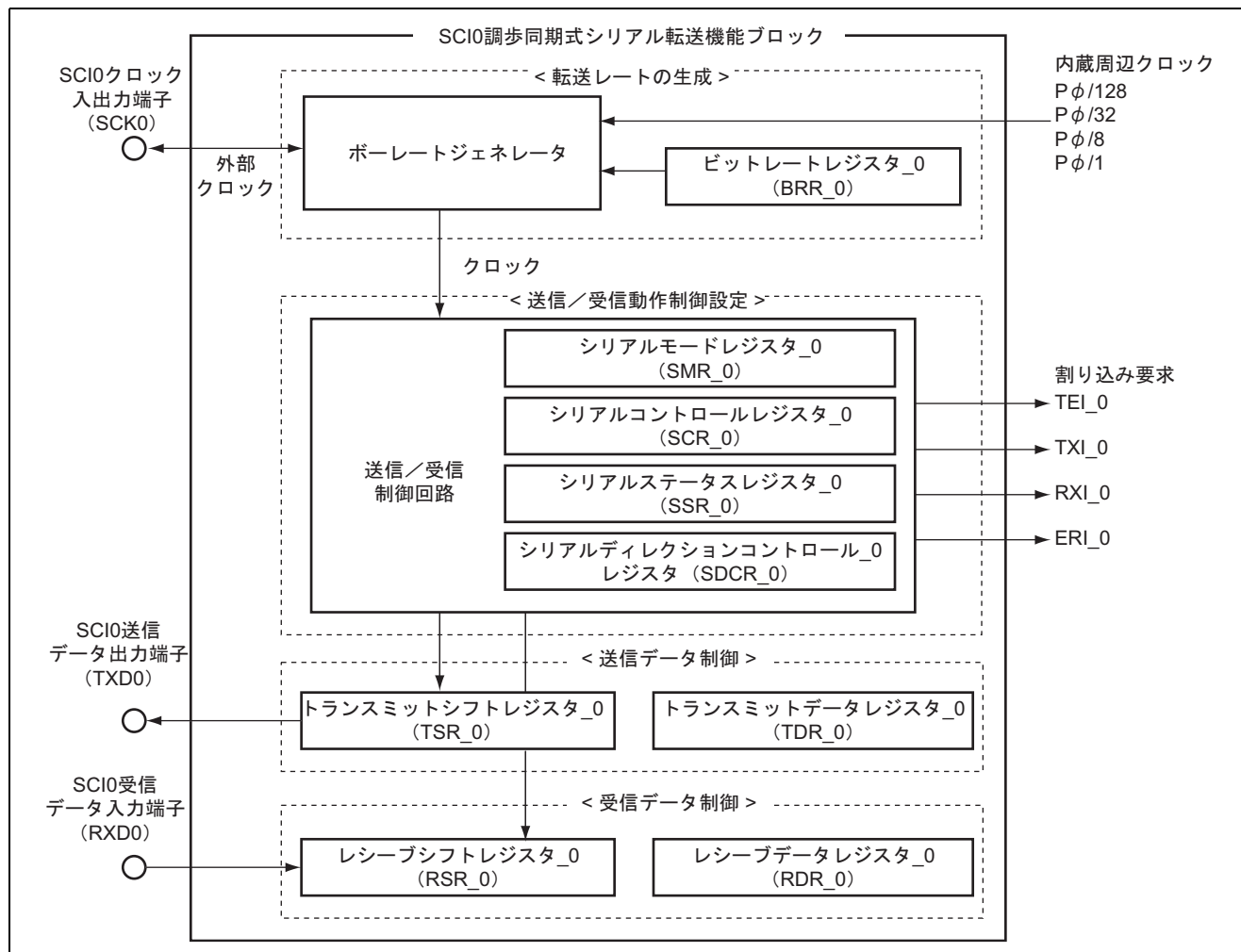


図 2 SCI (ch0) のブロック図

- 調歩同期式モードでは、キャラクタ単位で同期をとり、シリアルデータ通信を行い、Universal Asynchronous Receiver/Transmitter(UART)や、Asynchronous Communication Interface Adapter(ACIA)等の標準の調歩同期式通信専用 LSI とのシリアル通信が可能です。また、調歩同期式モードでは、複数のプロセッサ間のシリアル通信機能（マルチプロセッサ通信機能）を備えています。
- 内蔵周辺クロック P_φ は、内蔵周辺機能を動作させるための基準クロックであり、クロック発振器により生成されます。
- レシーブシフトレジスタ(RSR_0)は、シリアルデータを受信するためのレジスタです。RSR_0 は、RxD_0 端子からシリアルデータが入力され、1 フレーム分のデータを受信すると、データは自動的にレシーブデータレジスタ(RDR_0)へと転送されます。CPU からのアクセスは行えません。
- レシーブデータレジスタ(RDR_0)は、受信データを格納する 8 ビットのレジスタです。1 フレーム分のデータを受信すると、自動的に RSR_0 からデータが転送されます。RSR_0 と RDR_0 は、ダブルバッファ構造ですので、連続受信動作が可能です。RDR_0 は、受信専用レジスタのため、CPU からのリードのみ可能です。
- トランスミットシフトレジスタ(TSR_0)は、シリアルデータを送信するためのレジスタです。送信の際には、トランスミットデータレジスタ(TDR_0)から TSR_0 へと転送され、TxD_0 端子から送信データが出力されます。CPU からの直接アクセスは行えません。
- トランスミットデータレジスタ(TDR_0)は、送信データを格納する 8 ビットのレジスタです。TDR_0 に空きを検出すると、TDR_0 にライトしたデータは、自動的に TSR_0 に転送されます。また、TDR_0 と TSR_0 はダブルバッファ構造ですので、1 フレーム分のデータを送信したときに TDR_0 に次のデータがライトされていると、TSR_0 へと転送され連続送信が可能です。TDR は、常に CPU からのリード/ライトが可能です。ライトは、シリアルステータスレジスタ(SSR_0)の TDRE ビットが 1 であることを確認して行ってください。
- シリアルモードレジスタ(SMR_0)は、シリアルデータ通信フォーマットと、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(SCR_0)は、送受信制御と割り込み制御、および送受信クロックソースの選択を行うためのレジスタです。
- シリアルステータスレジスタ(SSR_0)は、SCI1 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、ORER、PER、FER はクリアのみ可能です。
- シリアルディレクションコントロールレジスタ(SDCR_0)は、LSB/MSB ファーストの選択を行います。8 ビット長通信の場合は LSB/MSB ファーストのどちらも選択可能ですが、7 ビット長通信の場合は LSB ファーストを選択してください。
- ビットレートレジスタ(BRR_0)は、ビットレートを調整するための 8 ビットのレジスタです。SCI は、チャネルごとにボーレートジェネレータが独立しているため、それぞれ異なるビットレートを設定可能です。設定値、実行レートの関係等、詳細はハードウェアマニュアルを参照ください。

表 2 に本タスク例の機能割付けを示します。

表 2 機能割付け

機能	分類	機能割付け
TXD0	端子	チャンネル 0 の送信データ出力端子
RXD0	端子	チャンネル 0 の送信データ入力端子
SMR_0	SCI0	通信フォーマットを調歩同期式モードに設定
SCR_0	SCI0	送信動作許可
SSR_0	SCI0	SCI0 の動作状態を示すステータスフラグ
SDCR_0	SCI0	LSB ファーストに設定
BRR_0	SCI0	通信のビットレートを設定
TSR_0	SCI0	シリアルデータを送信するためのレジスタ
TDR_0	SCI0	送信データを格納するレジスタ
RSR_0	SCI0	シリアルデータを受信するためのレジスタ
RDR_0	SCI0	受信データを格納するレジスタ

2.2 CMT (Compare Match Timer)

設定した周期ごとに割り込みを発生させます。図 3 に CMT モジュールチャンネル 0 (ch0) のブロック図を示し、以下に図 3 の機能説明をします。

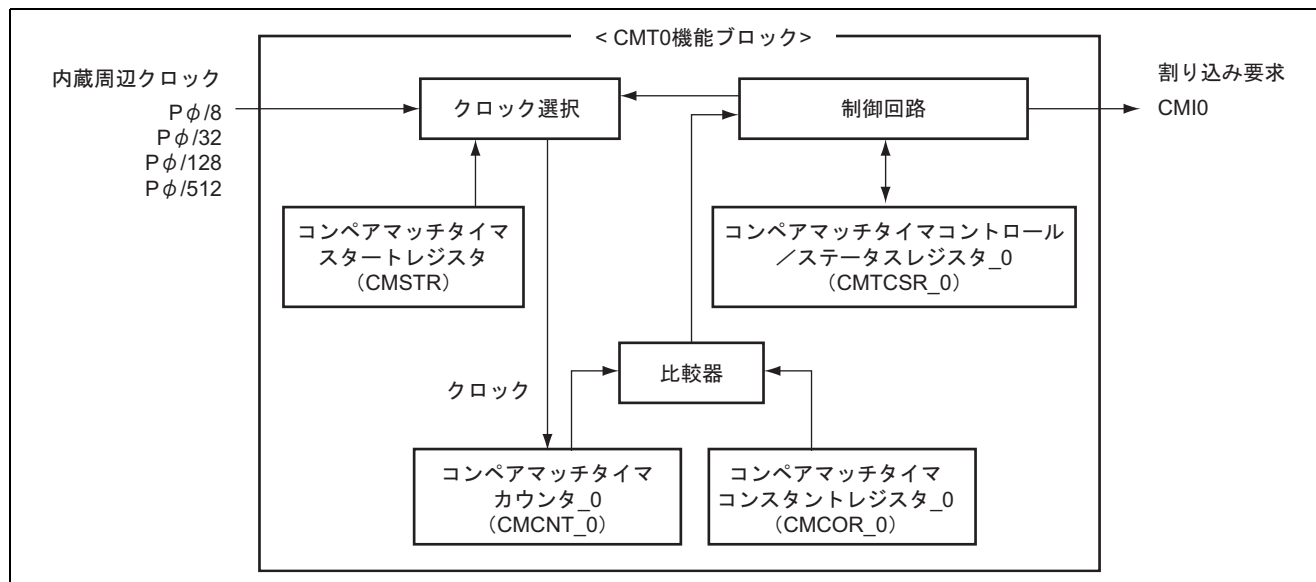


図 3 CMT(ch0)のブロック図

- CMT は、16 ビットのカウンタを持ち、設定した周期ごとの割り込みの発生が可能です。
- 内蔵周辺クロック Pφ を分周したクロックを選択することができます。そして、選択したクロックにより、カウントアップを行います。
- コンペアマッチタイマスタートレジスタ(CMSTR_0)は、カウントの開始および停止の設定を行います。
- コンペアマッチタイマコントロール/ステータスレジスタ(CMCSR_0)は、コンペアマッチ発生の表示、割り込みの設定、カウントアップ用クロックの選択を行います。
- コンペアマッチタイマカウンタ(CMCNT_0)は、割り込み要求を発生させるためのアップカウンタです。
- コンペアマッチタイマコンスタントレジスタ(CMCOR_0)は、コンペアマッチ周期を設定します。

3. 動作説明

図 4 に本タスク例における調歩同期式モードの受信時の動作を示します。また、表 3 に図 4 の説明として、ソフトウェアおよびハードウェア処理の内容を示します。

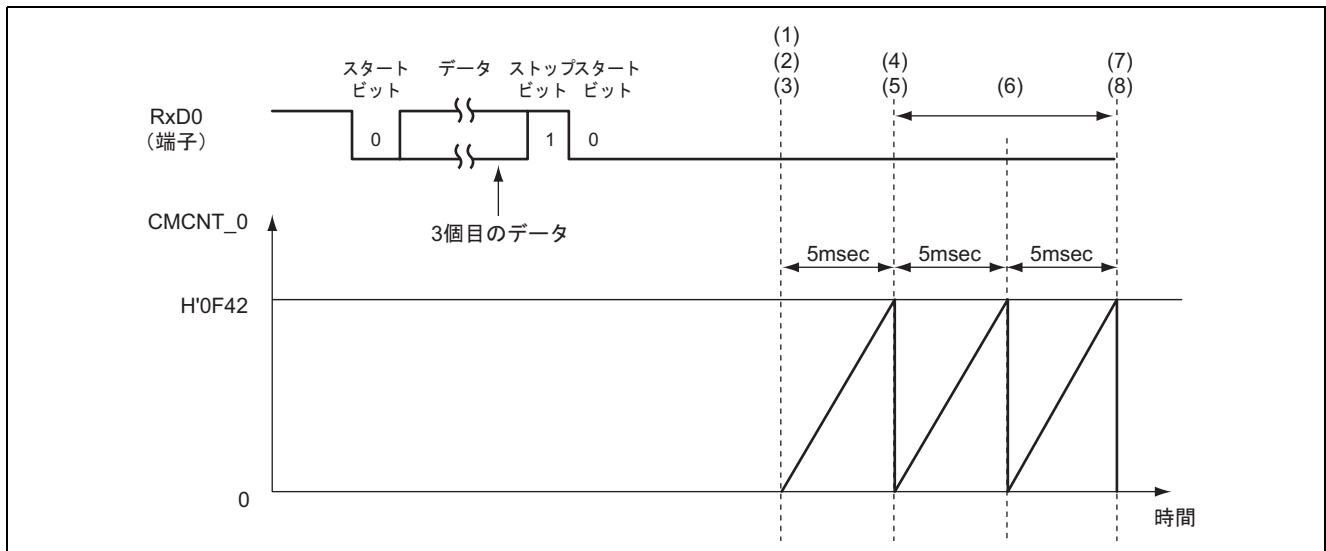


図 4 データ受信時の動作

表 3 処理内容

	《ソフトウェア処理》	《ハードウェア処理》
(1)	フレーミングエラー処理	SSR_0 の FER ビットを 1 にセット
(2)	CMSTR の STR0 ビットを 1 にセット	CMT_0 のカウントスタート
(3)	—	CMF フラグセット (コンペアマッチ割り込み発生)
(4)	CMF フラグを 0 クリア	CMT_0 のカウントスタート
(5)	RxD 端子レベルをリード	—
(6)	(3)~(5)を 2 回繰り返す	(3)~(5)を 2 回繰り返す
(7)	CMSTR の STR0 ビットを 0 クリア	CMT_0 の動作停止
(8)	SCR_0 の RE ビットを 0 クリア	SCI_0 の受信動作停止

4. ソフトウェア説明

4.1 モジュール説明

表 4 に本タスク例のモジュールを示します。

表 4 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	各種モジュールの呼び出し
SCI ルーチン	init_sci	SCI0 と CMT_0 の初期設定
受信ルーチン	rcv_sci	シリアルデータの受信
エラー処理ルーチン	err_int	各受信エラー処理
CMT0 割り込みルーチン	cmt_int	5msec 間隔で RxD 端子レベルのリード

4.2 内部レジスタ説明

表 5 に本タスク例で使用する内部レジスタを示します。なお、設定値は本タスク例において使用している値であり、初期値とは異なります。

表 5 使用内部レジスタ説明 (1)

レジスタ名		設定値	機能
ビット	ビット名		
MSTCR1			モジュールスタンバイコントロールレジスタ 1
0	MSTP16	0	SCI0 のスタンバイ制御ビット MSTP16 = 1 のときスタンバイ解除
MSTCR2			モジュールスタンバイコントロールレジスタ 2
12	MSTP12	0	CMT のスタンバイ制御ビット MSTP12 = 0 のとき CMT のスタンバイ解除
SCR_0		H'10	シリアルコントロールレジスタ_0(SCI_0) 送受信制御、割り込み制御、送受信クロックソースの選択
7	TIE	0	トランスミットインタラプトイネーブル 1 にセットすると、TXI 割り込み要求がイネーブルになる
6	RIE	0	レシーブインタラプトイネーブル 1 にセットすると、RXI および ERI 割り込み要求がイネーブルになる
5	TE	0	トランスミットイネーブル 1 にセットすると、送信動作が可能
4	RE	1	レシーブイネーブル 1 にセットすると、受信動作が可能
3	MPIE	0	マルチプロセスインタラプトイネーブル (調歩同期式モードで SMR の MP=1 のとき有効) 本タスク例では、MP=0 なので無効
2	TEIE	0	トランスミットエンドインタラプトイネーブル 1 にセットすると、TEI 割り込み要求がイネーブルになる
1 0	CKE1 CKE2	0 0	クロックイネーブル 1~0 クロックソースおよび SCK 端子機能を選択 本タスク例では、クロックソースは内部クロック、SCK 端子は未使用

表 5 使用内部レジスタ説明 (2)

レジスタ名		設定値	機能
ビット	ビット名		
SMR_0		H'00	シリアルモードレジスタ 0 通信フォーマット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レジスタ
7	C/ $\bar{A}$	0	コミュニケーションモード 0 のとき、調歩同期式モードで動作
6	CHR	0	キャラクタレングス（調歩同期式モードのみ有効） 0 のとき、データ長 8 ビットで送受信
5	PE	0	パリティイネーブル（調歩同期式モードのみ有効） 0 のとき、ノンパリティで送受信を行う
4	O/ $\bar{E}$	0	パリティモード（調歩同期式モードで PE=1 のとき有効） 本タスク例では、PE=0 なので本ビットは無効
3	STOP	0	ストップビットレングス（調歩同期式モードのみ有効） 0 のとき、1 ストップビットで送信を行う
2	MP	0	マルチプロセッサモード（調歩同期式モードのみ有効） 0 のとき、マルチプロセッサ通信機能は無効
1 0	CKS1 CKS2	0 0	クロックセレクト 1~0 00 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、 $P\phi$ クロックに設定
BRR_0		H'40	ビットレートレジスタ 0 ビットレートを調整するための 8 ビットのレジスタ
SDCR_0		H'F2	シリアルディレクションコントロールレジスタ 0 DIR ビット（ビット 3）により LSB/MSB ファーストを選択する 本タスク例では、DIR=0（LSB ファースト）に設定
SSR_0		H'xx	シリアルステータスレジスタ 0 SCI1 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 ステータスフラグは、フラグクリアのための 0 のみ書き込み可能
7	TDRE	※	トランスミットデータレジスタエンプティ（ステータスフラグ）
6	RDRF	※	レシーブデータレジスタフル（ステータスフラグ）
5	ORER	※	オーバーランエラー（ステータスフラグ）
4	FER	※	フレーミングエラー（ステータスフラグ）
3	PER	※	パリティエラー（ステータスフラグ）
2	TEND	※	トランスミットエンド（ステータスフラグ）
1	MPB	0	マルチプロセッサビット
0	MPBT	0	マルチプロセッサビットトランスファ
PACRL2			ポート A コントロールレジスタ L2
0	PA0MD	1	PA0 モードビット ポート A のマルチプレクス端子 PA0 の機能設定（Rx/D0）

【注】※0 クリアのみ可能、1 のセットはハード的に行われます。

表 5 使用内部レジスタ説明 (3)

レジスタ名		設定値	機能
ビット	ビット名		
MSTCR2			モジュールスタンバイコントロールレジスタ 2
12	MSTP12	0	CMT のスタンバイ制御ビット MSTP12=0 のとき CMT のスタンバイ解除
CMSTR		H'01	コンペアマッチタイマスタートレジスタ
15-2	—	0	リザーブビット
1	STR1	0	カウントスタート 1 STR1=0 で CMCNT_1 はカウント動作停止
0	STR0	1	カウントスタート 0 STR0=1 で CMCNT_0 はカウント動作
CMCSR_0			コンペアマッチタイマコントロール/ステータスレジスタ_0
15-8	—	0	リザーブビット
7	CMF	※	コンペアマッチフラグ CMF=1 のとき CMCNT と CMCOR の値が一致
6	CMIE	1	コンペアマッチ割り込みイネーブル コンペアマッチ割り込みの許可と禁止の選択 CMIE=1 のときコンペアマッチ割り込み許可
5-2	—	0	リザーブビット
1-0	CKS1 CKS0	0 1	CMCNT_0 に入力するクロックの選択 本タスク例では Pφ/32 を選択
CMCNT_0		—	コンペアマッチタイマカウンタ_0 割り込み要求を発生させるためのアップカウンタ
CMCOR_0		H'0F4 2	コンペアマッチタイマコンスタントレジスタ CMCNT_0 とのコンペアマッチ周期
IPRG		H'00F 0	インタラプトプライオリティレジスタ G 割り込み要因の優先順位の設定
7-4	IPR7 IPR6 IPR5 IPR4	1 1 1 1	CMT_0 の優先順位 (0-15) の設定

【注】※ 0 クリアのみ可能で、1 のセットはハードウェアにより自動的に行われます。

4.3 引数の説明

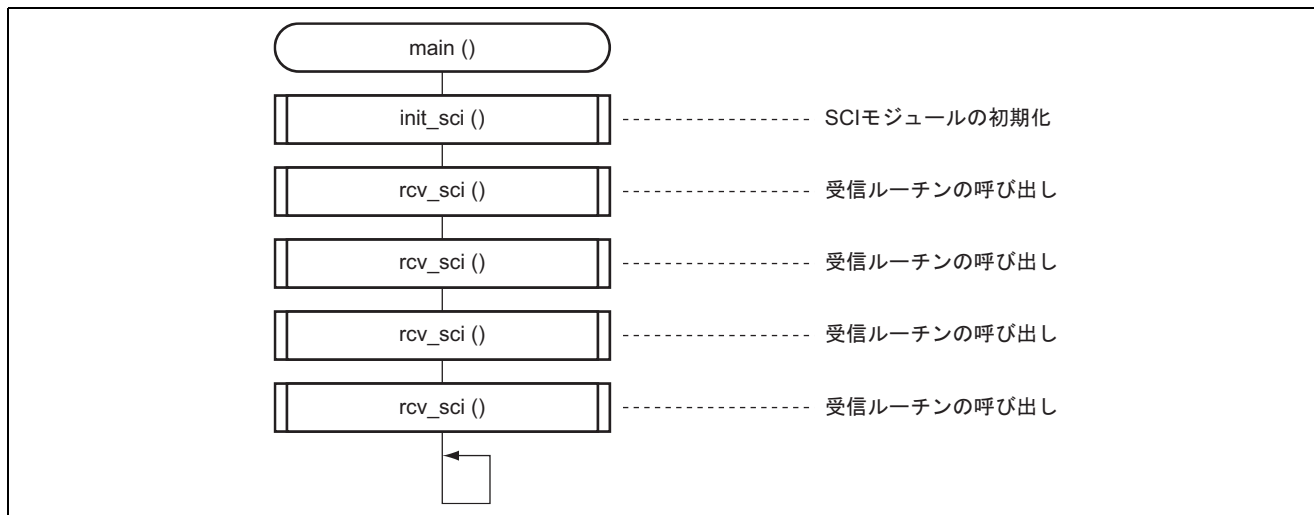
表 6 に本タスク例で使用する引数を示します。

表 6 引数説明

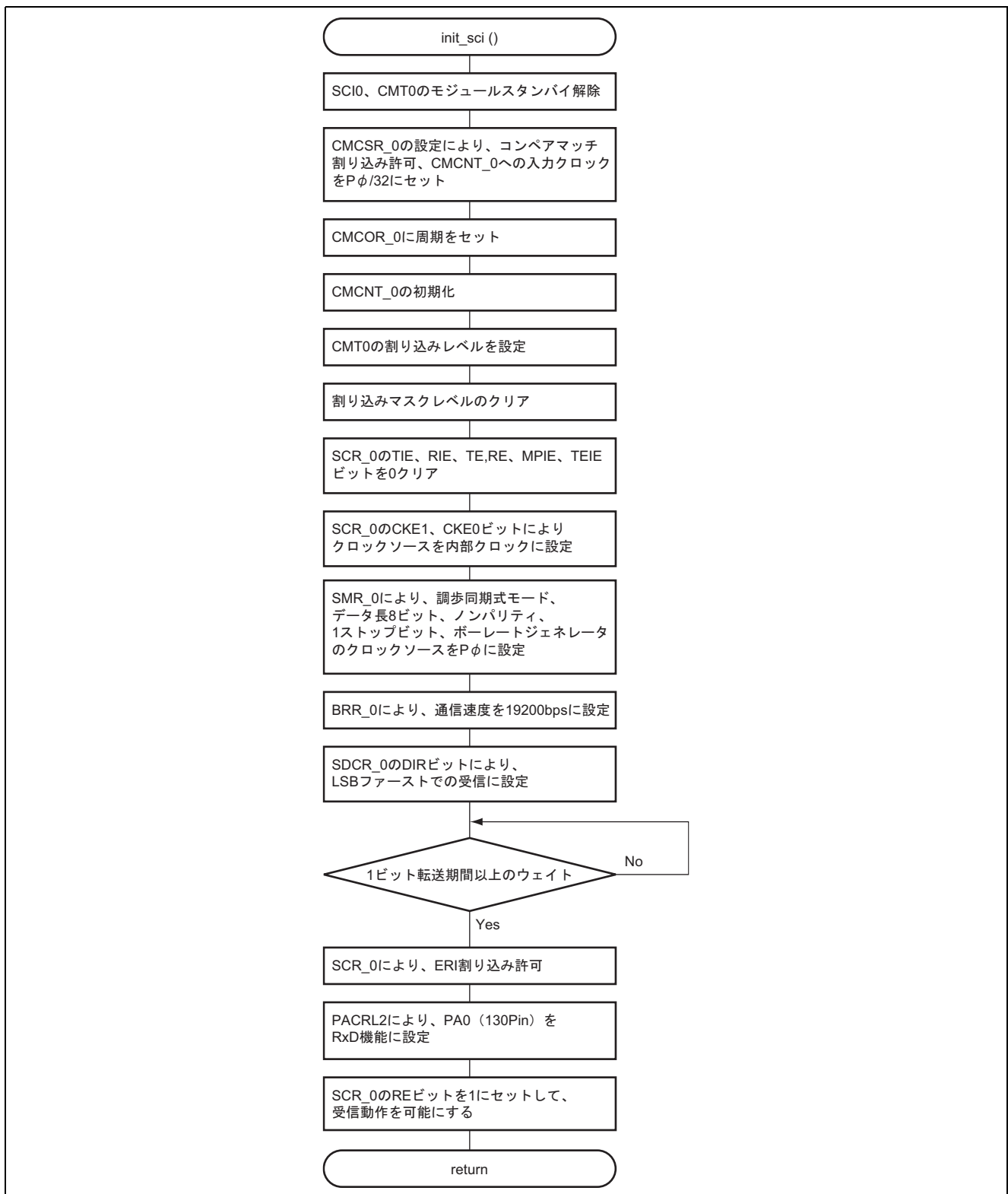
引数名	機能	使用モジュール名
Rev_data[0~2]	SCI0 の受信データの格納	受信ルーチン
T_count	CMT0 割り込みの回数のカウント	CMT_0 割り込みルーチン エラー処理ルーチン
Rd_count	RxD 端子レベルリード時ローレベルであるときの回数	CMT_0 割り込みルーチン エラー処理ルーチン

5. フローチャート

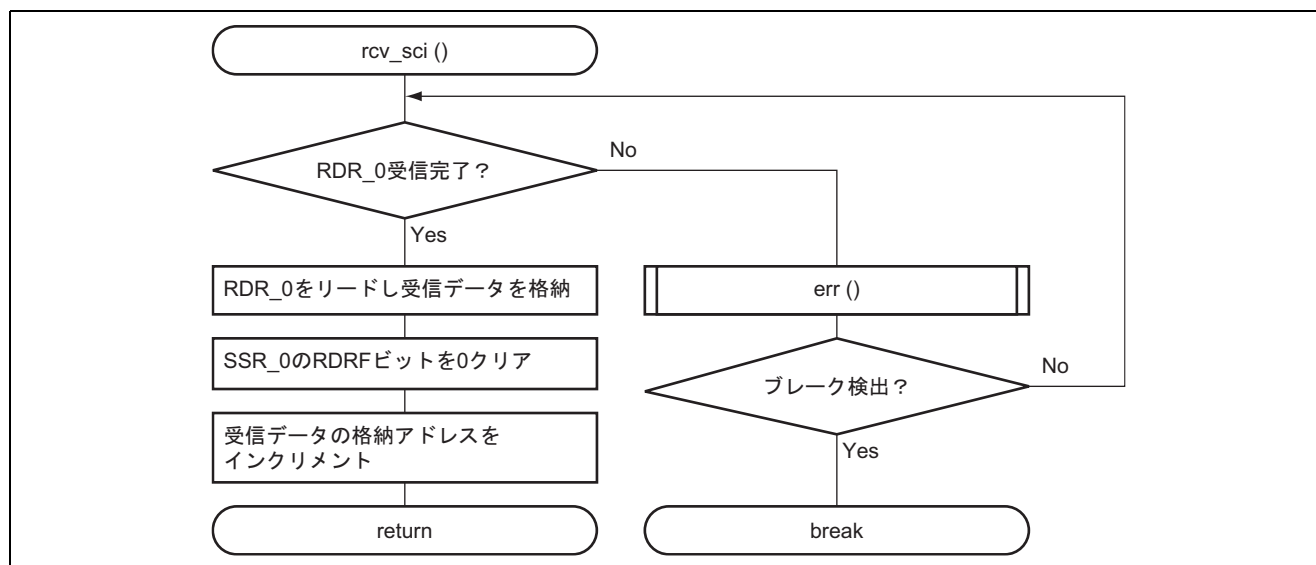
5.1 メインルーチン



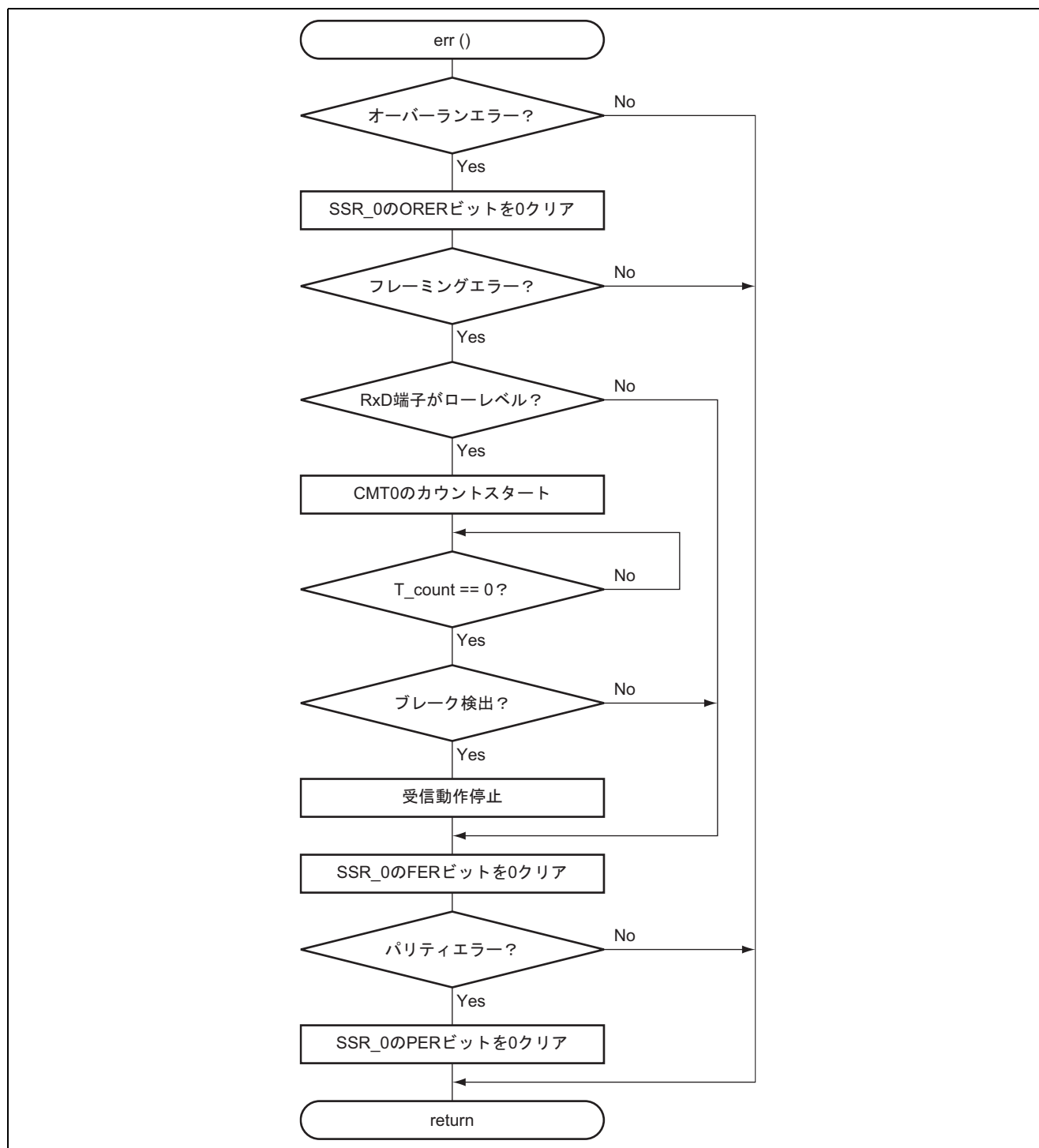
5.2 SCI ルーチン



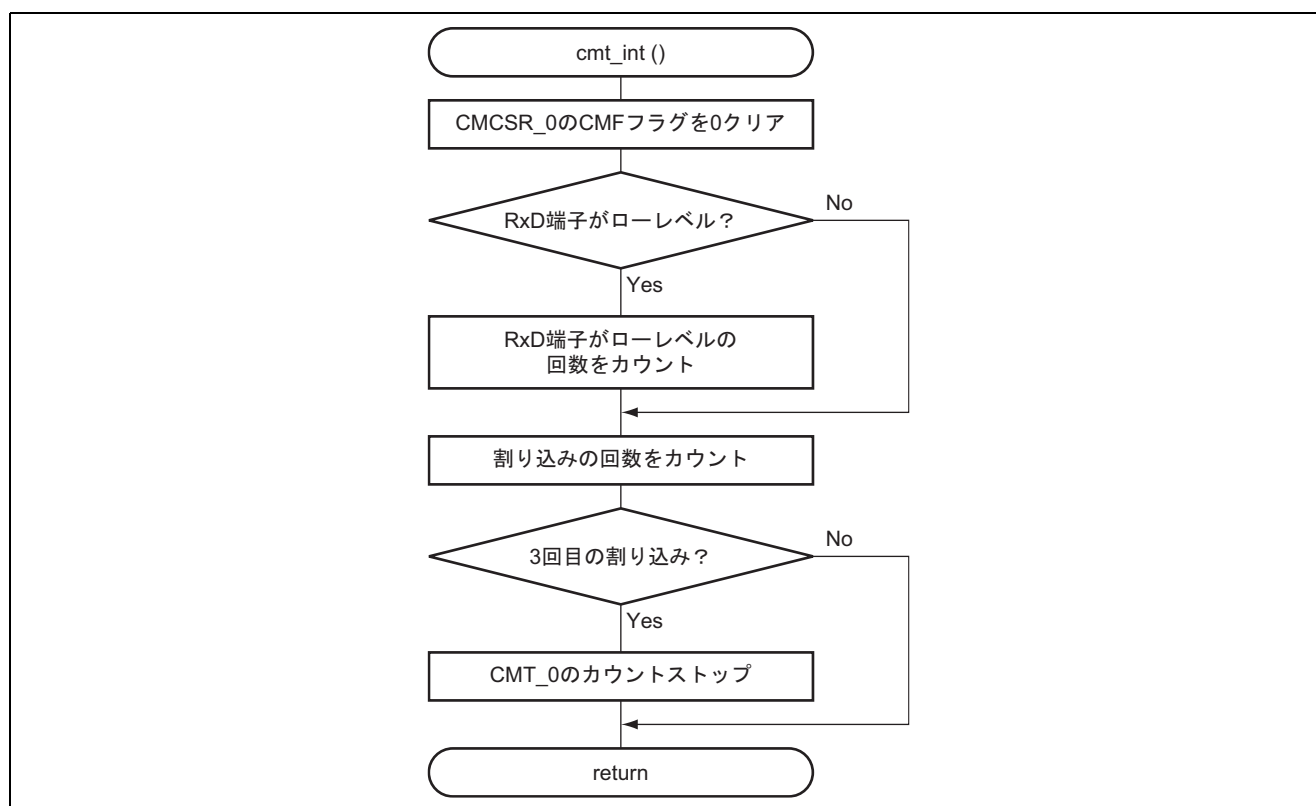
5.3 受信ルーチン



5.4 エラー処理ルーチン



5.5 CMT0 割り込みルーチン



6. プログラムリスト

```

/*****
/* SH7145F  Application Note
/*
/* Function
/* :SCI0
/* :Asynchronous Receive Mode(break)
/*
/* External input clock      :12.5MHz
/* Internal CPU clock        :50MHz
/* Internal peripheral clock  :25MHz
/*
/* Written      :2003/12  Rev.1.0
*****/

#include "iodef.h"
#include <machine.h>

/*----- Symbol Definition -----*/

#define COUNT 3

/*----- Function Definition -----*/
void main(void);

void init_sci(void);
unsigned char rcv_sci(unsigned char);

void err(void);
void cmt_int(void);
void dummy_f(void);

/*----- RAM allocation Definition -----*/

unsigned char T_count;          /* timer interrupt count */
unsigned char Rd_Count;         /* RxD low level count */

volatile unsigned char Rev_data[COUNT]; /* receive data */

/*****
/* main Program
*****/
void main( void )
{
    unsigned char i = 0;

    init_sci();          /* SCI initialize routine */

    i = rcv_sci(i);
    i = rcv_sci(i);
    i = rcv_sci(i);
    i = rcv_sci(i);

    while(1);
}

```

```

/*****/
/* function : init_sci */
/* operation : Initialize SCI0 */
/* Asynchronous Receive Mode */
/* -data length :8bit */
/* -stop bit :1bit */
/* -parity bit :non-parity bit */
/*****/
void init_sci(void)
{
    unsigned long i;

    T_count = 0;
    Rd_count = 0;

    P_STBY.MSTCR1.BIT.MSTP16 = 0; /* disable SCI_0 standby mode */
    P_STBY.MSTCR2.BIT.MSTP12 = 0; /* disable CMT standby mode */

    P_CMT.CMCSR_0.WORD = 0x0041; /* initialize CMCSR_0 */
        // [15-8] = 0
        // [7]CMF = 0
        // [6]CMIE = 1 CMT_0 interrupt enable
        // [5-2] = 0
        // [1]CKS1 = 0
        // [0]CKS0 = 1 count clock = Pφ/32
    P_CMT.CMCOR_0 = 0x0F42; /* set CMCOR_0 (5msec) */
    P_CMT.CMCNT_0 = 0; /* initialize CMCNT_0 */
    P_INTC.IPRG.BIT.CMT0 = 0xF; /* set CMT_0 interrupt level */
    set_imask(0); /* clear interrupt mask level */

    /* Initialize SCI Asynchronous mode */
    P_SCI0.SCR_0.BYTE &= 0x03; /* clear TIE,RIE,TE,RE,MPIE,TEIE bit */
    P_SCI0.SCR_0.BIT.CKE = 0; /* clock:internal,SCK:output */
    P_SCI0.SMR_0.BYTE = 0x00; /* 8bit,No parity,1stop bit */
        // CA = 0; /* asynchronous mode */
        // CHR = 0; /* data length 8 bits */
        // PE = 0; /* non-parity */
        // OE = 0; /* (=0)even parity */
        // STOP = 0; /* 1 stop bit */
        // CKS = 0; /* clock source = Pφ(25MHz) */

    P_SCI0.BRR_0 = 40; /* 19200bps@25MHz */
    P_SCI0.SDCR_0.BIT.DIR = 0; /* LSB first */

    for( i=0; i < 0x0400 ; i++); /* wait 1 bit */

    /* Initialize SCI0 PORT */
    P_PORTA.PACRL2.BIT.PA0MD = 1; /* set RXD0(PA1:130pin@SH7145) */

    P_SCI0.SCR_0.BIT.RE = 1; /* RE=1, receive enable */
}

/*****/
/* function: rcv_sci */
/* operation : receive serial data(SCI0) */
/* Asynchronous Receive Mode */

```

```

/*****/
unsigned char rcv_sci(unsigned char rev_count )
{
    while(P_SCI0.SSR_0.BIT.RDRF == 0){ /* wait until RDRF flag high level */
        err(); /* error judging */

        /* judging break detect */
        if(P_SCI0.SCR_0.BIT.RE == 0) /* break detect */
            break;
    }

    Rev_data[rev_count] = P_SCI0.RDR_0; /* store receive data */

    P_SCI0.SSR_0.BIT.RDRF = 0; /* clear RDRF flag */

    rev_count++; /* storing address increment */

    return(rev_count);
}

/*****/
/* function: err */
/* operation : error judging(SCI0) */
/* Asynchronous Receive Mode */
/*****/
void err(void)
{
    if(P_SCI0.SSR_0.BIT.Over == 1){ /* overrun error */
        P_SCI0.SSR_0.BIT.Over = 0; /* clear Over flag */
    }

    if(P_SCI0.SSR_0.BIT.FER == 1){ /* framing error */

        if(P_PORTA.PADRL.BIT.PA0DR == 0){
            P_CMT.CMSTR.BIT.STR = 1; /* CMT_0 count start */

            while(T_count != 0); /* wait until T_count = 0 */

            if(Rd_count == 3){ /* break detect (Rd_count==3) */
                P_SCI0.SCR_0.BIT.RE = 0; /* RE=0, receive disable */
            }
        }

        P_SCI0.SSR_0.BIT.FER = 0; /* clear FER flag */
    }

    if(P_SCI0.SSR_0.BIT.PER == 1){ /* parity error */
        P_SCI0.SSR_0.BIT.PER = 0; /* clear PER flag */
    }
}

/*****/
/* Interruption Program */
/*****/
/*****/
/* CMT0 Interruption Program */
/*****/
#pragma interrupt(cmt_int)

```

```

void cmt_int(void)
{
    P_CMT.CMCSR_0.BIT.CMF = 0;          /* clear CMF flag */

    /* read RxD-pin level */
    if(P_PORTA.PADRL.BIT.PA0DR == 0){    /* RxD-pin level is low */
        Rd_count++;                      /* count */
    }

    T_count--;                          /* count interrupt times */

    /* judging interrupt times */
    if(T_count == 0){                   /* 3 times interrupt */
        P_CMT.CMSTR.BIT.STR = 0;        /* CMT_0 count stop */
    }
}
/*****
/* other Interruption Program */
*****/
#pragma interrupt(dummy_f)
void dummy_f(void)
{
    /* Other Interrupt */
}

```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.09.15	—	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注意ください。

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、その他お気づ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。