

SH7752 グループ

R01AN1138JJ0101

Rev.1.01

SCIF (SCIF0, 1) ドライバソフトウェア

2012.7.11

要旨

本仕様書は、SH7752 グループの SCIF01 (Serial Communication Interfaces with FIFO 0 and 1) ドライバについて説明します。

動作確認デバイス

SH7752

動作環境

本仕様書に示す SCIF01 ドライバの動作環境を以下に示します。

- 評価ボード SH7752 グループ EVB ボード : ROP7752C00000RZ

- ソフトウェア : High-Performance Embedded Workshop- Ver 4.09.00.007
 : Toolchain - Ver 9.4.1.0
 : OptLinker - Ver 10.01.00
 : SH アセンブラ - Ver 7.01.02
 : SH C/C++コンパイラ- Ver 9.04.01
 : SH C/C++ライブラリジェネレータ - Ver 3.00.03
- エミュレータ : E10A エミュレータ Ver 3.03.00

目次

1.	機能概要.....	3
1.1	主な機能.....	3
1.2	関連ドキュメント.....	3
2.	ドライバ仕様.....	4
2.1	関数一覧.....	4
2.2	コールバック関数仕様一覧.....	4
2.3	エラーコード一覧.....	5
3.	関数.....	6
	R_SCIF01_SetOpr.....	7
	R_SCIF01_ComStop.....	8
	R_SCIF01_ComDataTx.....	9
	R_SCIF01_ComDataRx.....	10
	R_SCIF01_GetStatus.....	11
	R_SCIF01_SetModem.....	12
	R_SCIF01_GetModem.....	13
	R_SCIFn_Interrupt (n=0 or 1).....	14
4.	動作条件.....	15
4.1	ポートについて.....	15
4.1.1	シリアルマルチプレクス.....	15
4.1.2	汎用入出力ポート.....	16
4.2	割り込みについて.....	17
4.3	使用メモリ領域について.....	17
5.	注意事項.....	17
6.	付録.....	18
6.1	ドライバ関数の使用例.....	18
6.1.1	シリアルデータの送信例.....	18
6.1.2	シリアルデータの受信例.....	19
6.1.3	モデム機能 (CTS) を使用したシリアルデータの送信例.....	20
6.1.4	モデム機能 (RTS) を使用したシリアルデータの受信例.....	21
6.1.5	エラー処理例.....	22

1. 機能概要

本デバイスドライバは、SH7752 に搭載されている Serial Communication Interface with FIFO モジュールのチャンネル 0, 1(以下、SCIF01)を使用し、接続デバイスとのシリアルデータ通信処理を実現する関数群で構成されています。

1.1 主な機能

SCIF01 ドライバの主な機能を以下に示す。

- ・ SCIF01 チャンネルの初期化处理
- ・ シリアルデータ通信（データ送受信）
- ・ 受信エラーの通知処理
- ・ SCIF01 チャンネルの終了処理（通信停止）

1.2 関連ドキュメント

SH7752 グループ ユーザーズマニュアル：ハードウェア編 Rev. 1.00

2. ドライバ仕様

2.1 関数一覧

表 1に、SCIF01 ドライバが提供する関数一覧を示す。

表 1. SCIF01 ドライバ関数一覧

分類	関数名	機能概要	スタック サイズ
SCIF01 設定関数	R_SCIF01_SetOpr	チャンネルの初期化处理	20 バイト
	R_SCIF01_ComStop	チャンネルの終了処理	4 バイト
データ通信関数	R_SCIF01_ComDataTx	データの送信処理	16 バイト
	R_SCIF01_ComDataRx	データの受信処理	16 バイト
	R_SCIF01_GetStatus	ステータス情報取得処理	4 バイト
	R_SCIF01_SetModem	モデムステータス情報設定処理	4 バイト
	R_SCIF01_GetModem	モデムステータス情報取得処理	4 バイト
割り込みハンドラ関数	R_SCIFn_Interrupt	SCIF01 割り込み	40 バイト

2.2 コールバック関数仕様一覧

表 2にSCIF01 ドライバのコールバック関数仕様一覧を示す。SCIF01 ドライバのコールバック関数はシリアルデータ受信処理のエラー通知、および、データ送受信完了通知としてコールする。

表 2. SCIF01 ドライバコールバック関数仕様一覧

関数名	コールバック名	仕様内容
R_SCIF01_SetOpr	void (*pv_receive_err_callback) (void)	受信エラー割り込みのコールバック。 (NULL の場合、コールバックなし)
R_SCIF01_SetOpr	void (*pv_modem_callback) (void)	モデムステータス割り込みのコールバック。 (NULL の場合、コールバックなし)
R_SCIF01_ComDataTx	void (*pv_data_tx_end_callback) (void)	送信FIFOへのセット完了コールバック。 (NULL の場合、コールバックなし)
R_SCIF01_ComDataRx	void (*pv_data_rx_end_callback) (void)	データ受信完了時のコールバック。 (NULL の場合、コールバックなし)
R_SCIF01_ComDataRx	void (*pv_timeout_callback) (void)	キャラクタータイムアウト割り込みのコールバック。 (NULL の場合、コールバックなし)

2.3 エラーコード一覧

表 3にSCIF01 ドライバのエラーコード一覧を示す。

表 3. SCIF01 ドライバエラーコード一覧

値	エラーコード (マクロ定義)	エラー内容
0	RET_NORMAL	正常終了
-1	RET_ERR_PARAM1	第 1 引数不正
-2	RET_ERR_PARAM2	第 2 引数不正

3. 関数

本章では、SCIF01 ドライバの各関数仕様詳細を示す。各関数詳細の読み方は以下のとおりです。

関数名		分類
機能概要		
書式	関数の呼び出し形式を示します。 <code>#include</code> “ヘッダファイル” で示すヘッダファイルは、この関数の実行に必要な標準ヘッダファイルで、必ずインクルードする。	
引数	I, 0 は、引数がそれぞれ入力データ、出力データであることを意味する。	
戻り値	関数の戻り値を示す。	
解説	関数の仕様について説明する。	
注意事項	注意事項があればここに示す。	

R_SCIF01_SetOpr

SCIF01 設定関数

チャンネルの初期設定処理

```
書式      #include "r_common.h"
          #include "r_SH7752.h"
          #include "r_scif01_if.h"
          char_t R_SCIF01_SetOpr ( uchar_t uc_ch_num, scif01_opr_t *pst_scif01_opr );
```

```
引数      uchar_t uc_ch_num          I   SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1)
          SCIF01_opr_t *pst_SCIF01_opr  I   SCIF01 オペレーション構造体へのポインタ
```

```
戻り値    RET_NORMAL          正常終了
          RET_ERR_PARAM1     第 1 引数不正
          RET_ERR_PARAM2     第 2 引数不正
```

解説 本関数は、SCIF01 の初期設定を行います。各設定内容は【SCIF01 オペレーション構造体】を参照してください。また、送信 FIFO、受信 FIFO を有効にします。

受信エラー、モデムステータス割り込みを通知する場合、各メンバ変数 (pv_receive_err_callback、pv_modem_callback) に指定したユーザ定義関数をコールすることで通知します。

【SCIF01 オペレーション構造体】 pst_scif01_opr

メンバ	I/O	機能
uchar_t uc_divisor_clock_sel;	I	ボーレートジェネレータのクロックソース選択 0:Pck/26 (Pck の 26 分周クロック) 1:Pck/13 (Pck の 13 分周クロック) 2:Pck/2 (Pck の 2 分周クロック)
uchar_t uc_receive_ff_trigger;	I	受信 FIFO 割り込みトリガ選択 1 : 1 データ 4 : 4 データ 8 : 8 データ 14: 14 データ
ushort_t us_bitrate;	I	ボーレートジェネレータの設定値 (ディバイザラッチ [FDLH , FDLL] H' 0001 ~H' FFFE
uchar_t uc_data_length;	I	データ長選択 5 : 5 ビット長 6 : 6 ビット長 7 : 7 ビット長 8 : 8 ビット長
uchar_t uc_parity;	I	パリティ選択 0: パリティなし 1: 偶数パリティ 2: 奇数パリティ
uchar_t uc_stop_bits;	I	ストップビット 1: 1 ストップビット 2: 2 ストップビット (データ長が 5 ビットの場合、1.5 ストップビット)
void (*pv_receive_err_callback) (void);	I	受信エラー割り込みのコールバック 受信エラー割り込みのコールバックアドレス (NULL の場合、コールバックなし)
void (*pv_modem_callback) (void);	I	モデムステータス割り込みのコールバック モデムステータス割り込みのコールバックアドレス (NULL の場合、コールバックなし)

注意 ビットレート値の計算方法は、SH7752 グループユーザズマニュアル ハードウェア編「27 章 SCIF0, SCIF1 のディバイザラッチ H, L (FDLH, FDLL)」を参照してください。

R_SCIF01_ComStop

SCIF01 設定関数

チャンネルの終了処理

書式

```
#include "r_common.h"
#include "r_SH7752.h"
#include "r_scif01_if.h"
char_t R_SCIF01_ComStop( uchar_t uc_ch_num );
```

引数

uchar_t uc_ch_num	I	SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1)
-------------------	---	-----------------------------------

戻り値

RET_NORMAL	正常終了
RET_ERR_PARAM	第 1 引数不正

解説

本関数は、SCIF01 のチャンネル番号で選択したチャンネルの終了処理を行います。終了処理は、送受信停止、FIFO データレジスタの初期化、及び割り込み要求フラグを無効に設定にします。

シリアル通信の終了および、中断する時、本関数を使用してください。

注意

本関数を使用した後、シリアル通信を再開する場合、R_SCIF01_SetOpr () 関数で再設定してください。

R_SCIF01_ComDataTx

データ通信関数

データの送信処理

書式	<pre>#include "r_common.h" #include "r_SH7752.h" #include "r_scif01_if.h" char_t R_SCIF01_ComDataTx (uchar_t uc_ch_num, uchar_t *puc_data, ulong_t ul_size, void (*pv_data_tx_end_callback) (void));</pre>		
引数	uchar_t uc_ch_num	I	SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1)
	uchar_t *puc_data	I	送信するデータを格納する領域の先頭アドレス
	ulong_t ul_size	I	送信するデータ数
	void (*pv_data_tx_end_callback) (void)	I	送信 FIFO へのセット完了コールバックアドレス (NULL の場合、コールバックなし)
戻り値	RET_NORMAL		正常終了
	RET_ERR_PARAM1		第 1 引数不正
解説	<p>本関数は、SCIF01 のチャンネル番号で選択したチャンネルからデータを送信します。送信データ格納領域の先頭アドレスから、データサイズで指定した数送信します。送信 FIFO へのセットは pv_data_tx_end_callback に指定したユーザ定義関数をコールすることで通知します。送信の完了は R_SCIF01_GetStatus() 関数のトランスミッタエンプティで判断してください。</p>		
注意	<p>本関数を実行する前に、選択したチャンネルのデータ送信が完了していることを確認してください。</p>		

R_SCIF01_ComDataRx

データ通信関数

データの受信処理

書式	<pre>#include "r_common.h" #include "r_SH7752.h" #include "r_scif01_if.h" char_t R_SCIF01_ComDataRx (uchar_t uc_ch_num, uchar_t *puc_data,ulong_t ul_size, void (*pv_data_rx_end_callback)(void), void (*pv_timeout_callback)(void));</pre>		
引数	<pre>uchar_t uc_ch_num uchar_t *puc_data ulong_t ul_size void (*pv_data_rx_end_callback)(void) void (*pv_timeout_callback)(void)</pre>	<pre>I I I I I</pre>	<pre>SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1) 受信したデータを格納する領域の先頭アドレス 受信データ数 データ受信完了時のコールバックアドレス (NULL の場合、コールバックなし) キャラクタタイムアウトエラーのコールバックアドレス (NULL の場合、コールバックなし)</pre>
戻り値	<pre>RET_NORMAL RET_ERR_PARAM1</pre>	<pre>正常終了 第 1 引数不正</pre>	

解説 本関数は、SCIF01 のチャンネル番号で選択したチャンネルからデータを受信します。受信データ格納領域の先頭アドレスから、データサイズで指定した数、受信します。受信完了は pv_data_rx_end_callback、キャラクタタイムアウトエラーは pv_timeout_callback に指定したユーザ定義関数をコールすることで通知します。

注意 本関数を実行する前に、選択したチャンネルのデータ受信が完了していることを確認してください。本関数は、R_SCIF01_SetOpr () 関数の「受信 FIFO 割り込みトリガ選択」に設定した単位で受信するため、受信データが受信 FIFO 割り込みトリガに満たない場合、受信完了を通知できません。「受信データ数」は R_SCIF01_SetOpr () 関数の「受信 FIFO 割り込みトリガ選択」で指定した単位に合わせてください。

R_SCIF01_GetStatus

データ通信関数

ステータス情報取得処理

書式

```
#include "r_common.h"
#include "r_SH7752.h"
#include "r_scif01_if.h"
short_t R_SCIF01_GetStatus ( uchar_t uc_ch_num );
```

引数 uchar_t uc_ch_num **I** SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1)

戻り値 H'0000~H'00FF ステータス情報
RET_ERR_PARAM1 第1引数不正

解説 本関数は、SCIF01 のチャンネル番号で選択したチャンネルのステータス情報を取得します。

【ステータス情報詳細】

bit[15:8]	リザーブビット	0	を返します。
bit[7]	受信 FIFO エラー(注)	0:なし	1:発生中
bit[6]	トランスミッタエンプティ	0:送信中	1:送信完了
bit[5]	送信 FIFO (FTHR) エンプティ	0:データあり	1:データなし (送信 FIFO 空)
bit[4]	ブ레이크信号(注)	0:なし	1:発生中
bit[3]	フレーミングエラー(注)	0:なし	1:発生中
bit[2]	パリティエラー(注)	0:なし	1:発生中
bit[1]	オーバランエラー(注)	0:なし	1:発生中
bit[0]	データレディ	0:受信データなし	1:受信データあり

注. 本関数をコールするとエラーフラグはクリアされます。

注意 なし

R_SCIF01_SetModem

データ通信関数

モデムステータス制御設定処理

書式	<pre>#include "r_common.h" #include "r_SH7752.h" #include "r_scif01_if.h" char_t R_SCIF01_SetModem (uchar_t uc_ch_num, uchar_t uc_modem_con);</pre>		
引数	uchar_t uc_ch_num	I	SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1)
	uchar_t uc_modem_con	I	RTS、DTR の出力レベル選択
戻り値	RET_ERR_PARAM1		第 1 引数不正
	RET_ERR_PARAM2		第 2 引数不正
解説	<p>本関数は、SCIF01 のチャンネル番号で選択したチャンネルの RTS 及び DTR の出力レベルを選択します。</p> <p>【出力レベル選択の詳細】 uchar_t uc_modem_con</p> <p>bit[7:2] リザーブビット “0” を設定してください。</p> <p>bit[1] RTS (リクエストトウセンド) 0: H レベル 1: L レベル</p> <p>bit[0] DTR (データターミナルレデイ) 0: H レベル 1: L レベル</p>		
注意	なし		

R_SCIF01_GetModem

データ通信関数

モデムステータス情報取得処理

書式 `#include "r_common.h"`
 `#include "r_SH7752.h"`
 `#include "r_scif01_if.h"`
 `short_t R_SCIF01_GetStatus (uchar_t uc_ch_num);`

引数 `uchar_t uc_ch_num` **I** SCIF01 のチャンネル番号 (0:SCIF0 1:SCIF1)

戻り値 `H'0000~H'00FF` ステータス情報
 `RET_ERR_PARAM1` 第 1 引数不正

解説 本関数は、SCIF01 のチャンネル番号で選択したチャンネルのモデムステータス情報を取得します。

【ステータス情報詳細】各ビットが示す内容

<code>bit[15:8]</code>	リザーブビット	0 を返します。	
<code>bit[7]</code>	DCD	0: H レベル	1: L レベル
<code>bit[6]</code>	RI	0: H レベル	1: L レベル
<code>bit[5]</code>	DSR	0: H レベル	1: L レベル
<code>bit[4]</code>	CTS	0: H レベル	1: L レベル
<code>bit[3]</code>	DDCD(注)	0: 信号変化なし	1: 信号変化を検出
<code>bit[2]</code>	TERI(注)	0: 立ち上がり未検出	1: 立ち上がりを検出
<code>bit[1]</code>	DDSR(注)	0: 信号変化なし	1: 信号変化を検出
<code>bit[0]</code>	DCTS(注)	0: 信号変化なし	1: 信号変化を検出

注. 本関数をコールするとフラグはクリアされます。

注意 なし

R_SCIFn_Interrupt (n=0 or 1)

割り込みハンドラ関数

SCIF01 割り込みハンドラ

書式

```
#include "r_common.h"
#include "r_SH7752.h"
#include "r_scif01_if.h"
void R_SCIFn_ERIInterrupt ( void );
```

引数 なし

戻り値 なし

解説

本関数は、SCIF01 割り込み内での処理を行う。
本割り込みハンドラ関数で、該当コールバック関数をコールする。

注意

本関数は、SCIF01 割り込みのハンドラ内からコールすること。

4. 動作条件

4.1 ポートについて

本デバイスドライバ使用時は、使用するチャンネルと接続ポートの組み合わせをシリアルマルチプレクス（以降 SerMux）にて設定する必要があります。さらに、SerMux により接続されたポートに対して、汎用入出力ポート（以降 GPIO）の設定が必要になります。

4.1.1 シリアルマルチプレクス

表 4. シリアルマルチプレクス対応一覧

チャンネル	SerMuxモード設定														接続ポート
	0	1	2	3	4	5	6	無効	8	9	A	B	C	D	
SCIF0	○	○	○	○											COM1
					○		○								COM2
						○		○	○	○					RAC(SCIF0)
SCIF1															COM1
			○	○				○	○		○		○		COM2(SCIF1)
							○								RAC
SCIF2	○		○		○										SCIF2
															COM1
						○									COM2
											○	○			RAC
SCIF3		○		○			○	○							SCIF3
															COM1
	○														COM2
															RAC
SCIF4	SerMuxに影響なし														SCIF4

※： SCIF0, SCIF1 を使用する場合、○印が付いているモードを選択してください。

4.1.2 汎用入出力ポート

表 5. 汎用入出力ポート設定一覧

接続ポート	GPIO設定			
	レジスタ	ビット	設定	
COM1	POCR	P03MD [1:0]	00: COM1-TXD, SGPI02-CLK機能を選択する	
		P02MD [1:0]	00: COM1-RXD, SGPI02-LOAD機能を選択する	
		P01MD [1:0]	00: COM1-RTS, SGPI02-DI機能を選択する	
		P00MD [1:0]	00: COM1-CTS, SGPI02-D0機能を選択する	
	PSEL5	P03SEL	1: COM1-TXD機能を選択する	
		P02SEL	1: COM1-RXD機能を選択する	
		P01SEL	1: COM1-RTS機能を選択する	
		P00SEL	1: COM1-CTS機能を選択する	
COM2	PKGR	PK7MD [1:0]	00: COM2-TXD, SCK2機能を選択する	
		PK6MD [1:0]	00: COM2-RXD機能を選択する	
		PK5MD [1:0]	00: COM2-RTS機能を選択する	
		PK4MD [1:0]	00: COM2-CTS機能を選択する	
		PK3MD [1:0]	00: COM2-DTR機能を選択する	
		PK2MD [1:0]	00: COM2-DSR, SCK4機能を選択する	
		PK1MD [1:0]	00: COM2-DCD, SCK3機能を選択する	
	PVCR	PV7MD [1:0]	00: COM2-RI, A23機能を選択する	
	PSEL3	PK7SEL	0: COM2-TXD/TXD1機能を選択する	
		PK2SEL	0: COM2-DSR/DSR1機能を選択する	
		PK1SEL	0: COM2-DCD/DCD1機能を選択する	
	PSEL6	PV7SEL	1: COM2-RI/RI1機能を選択する。	
	RAC	PLCR	PL6MD [1:0]	00: RAC-RXD, RXD2機能を選択する
			PL5MD [1:0]	00: RAC-RTS, CS5機能を選択する
PL4MD [1:0]			00: RAC-CTS, CS6機能を選択する	
PL3MD [1:0]			00: RAC-DTR機能を選択する	
PL2MD [1:0]			00: RAC-DSR, AUDSYNC機能を選択する	
PL1MD [1:0]			00: RAC-DCD, AUDCK機能を選択する	
PL0MD [1:0]			00: RAC-TXD, TXD2機能を選択する	
PFCR		PF2MD [1:0]	00: RAC-RI, RMI11-RXD1機能を選択する	
PSEL4		PL6SEL	0: RAC-RXD/RXD0機能を選択する	
		PL5SEL	0: RAC-RTS/RTS0機能を選択する	
		PL4SEL	0: RAC-CTS/CTS0機能を選択する	
		PL2SEL	0: RAC-DSR/DSR0機能を選択する	
		PL1SEL	0: RAC-DCD/DCD0機能を選択する	
PSEL1		PLOSEL	0: RAC-TXD/TXD0機能を選択する	
		PF2SEL	1: RAC-RI/RIO機能を選択する	

4.2 割り込みについて

SH7752 グループのSCIF01 は、チャンネル毎に1つの割り込みを使用できます。表 6に登録が必要な割り込みハンドラ関数を示す。

表 6. 割り込みベクタテーブル設定関数一覧

チャンネル	割り込み要因番号	割り込み要因	割り込みハンドラ関数
SCIF0	0xB40	FINT0	R_SCIF0_Interrupt
SCIF1	0xB60	FINT1	R_SCIF1_Interrupt

4.3 使用メモリ領域について

本デバイスドライバで使用するメモリ領域を表 7に示す。

表 7. メモリ使用量一覧

内容	セクション	属性	バイト数
プログラムコード	P	code, align=4	3276 バイト
定数データ	C	data, align=4	0 バイト
初期値ありデータ	D	data, align=4	0 バイト
初期値なしデータ	B	data, align=4	80 バイト

5. 注意事項

- ・ 本ドライバはループバックテスト機能、ブレーク発生機能に対応していません。

6. 付録

6.1 ドライバ関数の使用例

SCIF01 ドライバ関数を使用したシリアルデータ通信の手順を以下に示す。

6.1.1 シリアルデータの送信例

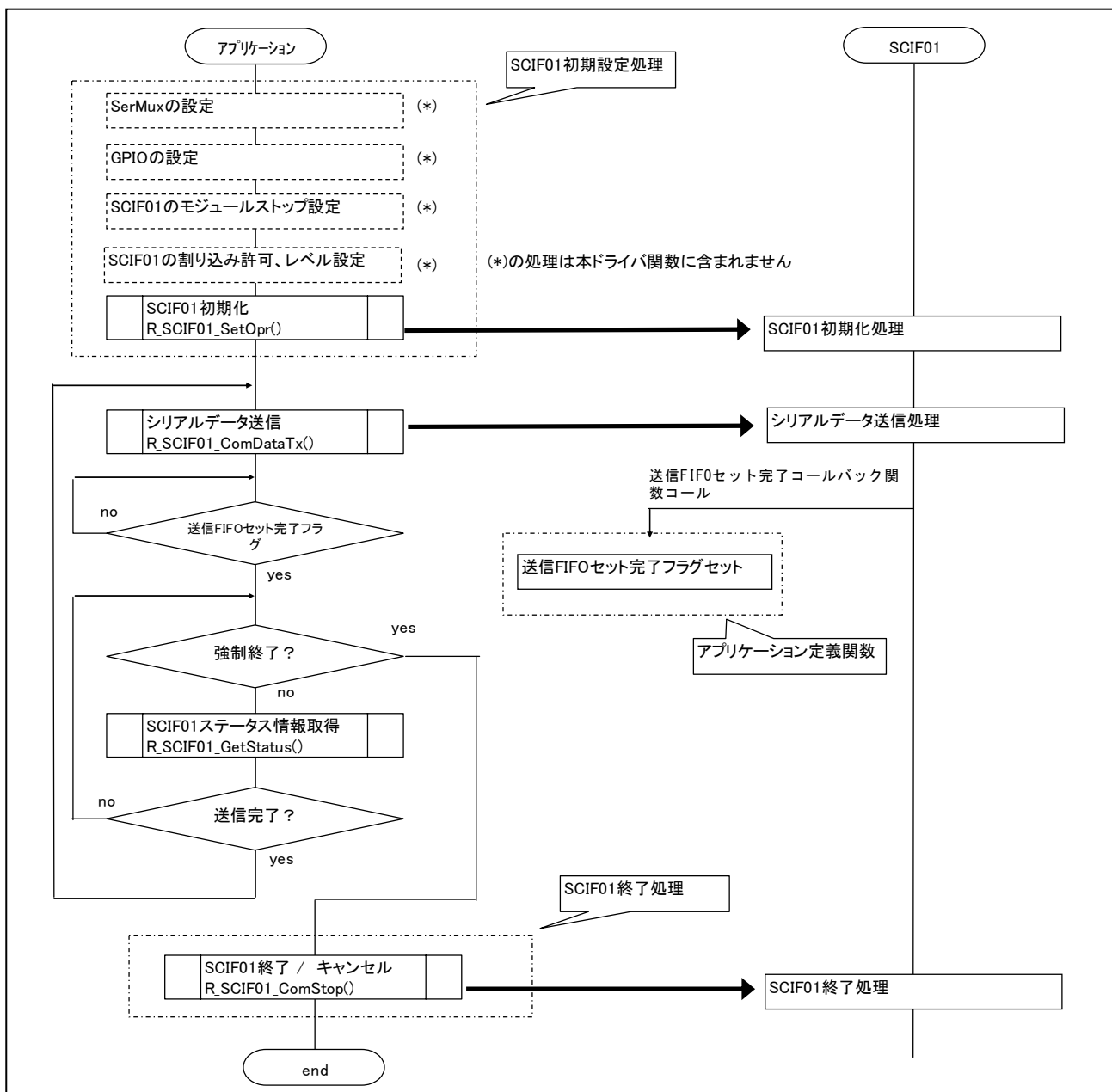


図 1. シリアルデータの送信例図

6.1.2 シリアルデータの受信例

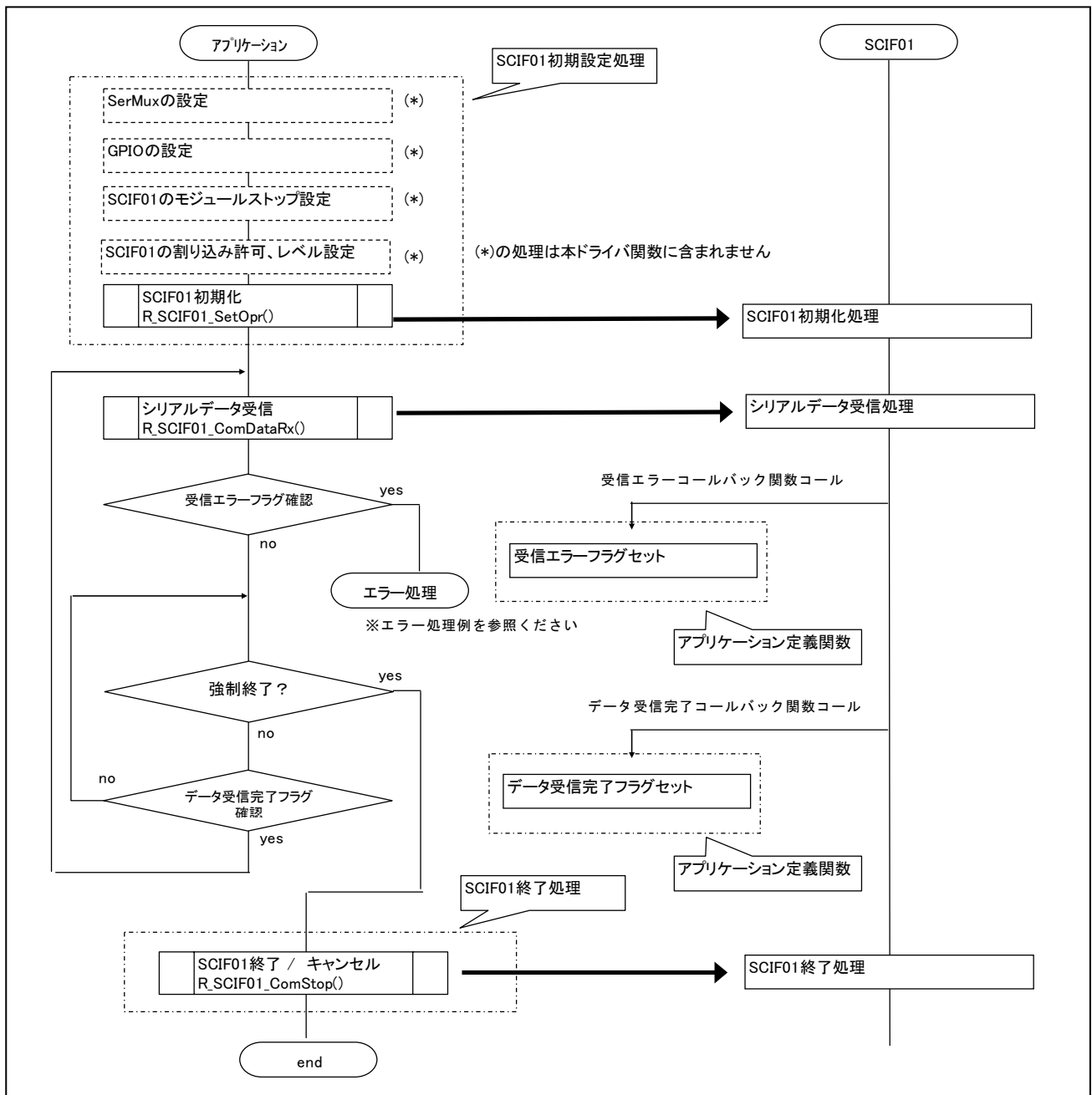


図 2. シリアルデータの受信例図

6.1.3 モデム機能 (CTS) を使用したシリアルデータの送信例

CTS 信号を使用した送信例を示します。CTS 信号を使用する場合、送信するデータ数は” 1 ” (R_SCIF01_ComDataTx() の ul_size) にしてください。

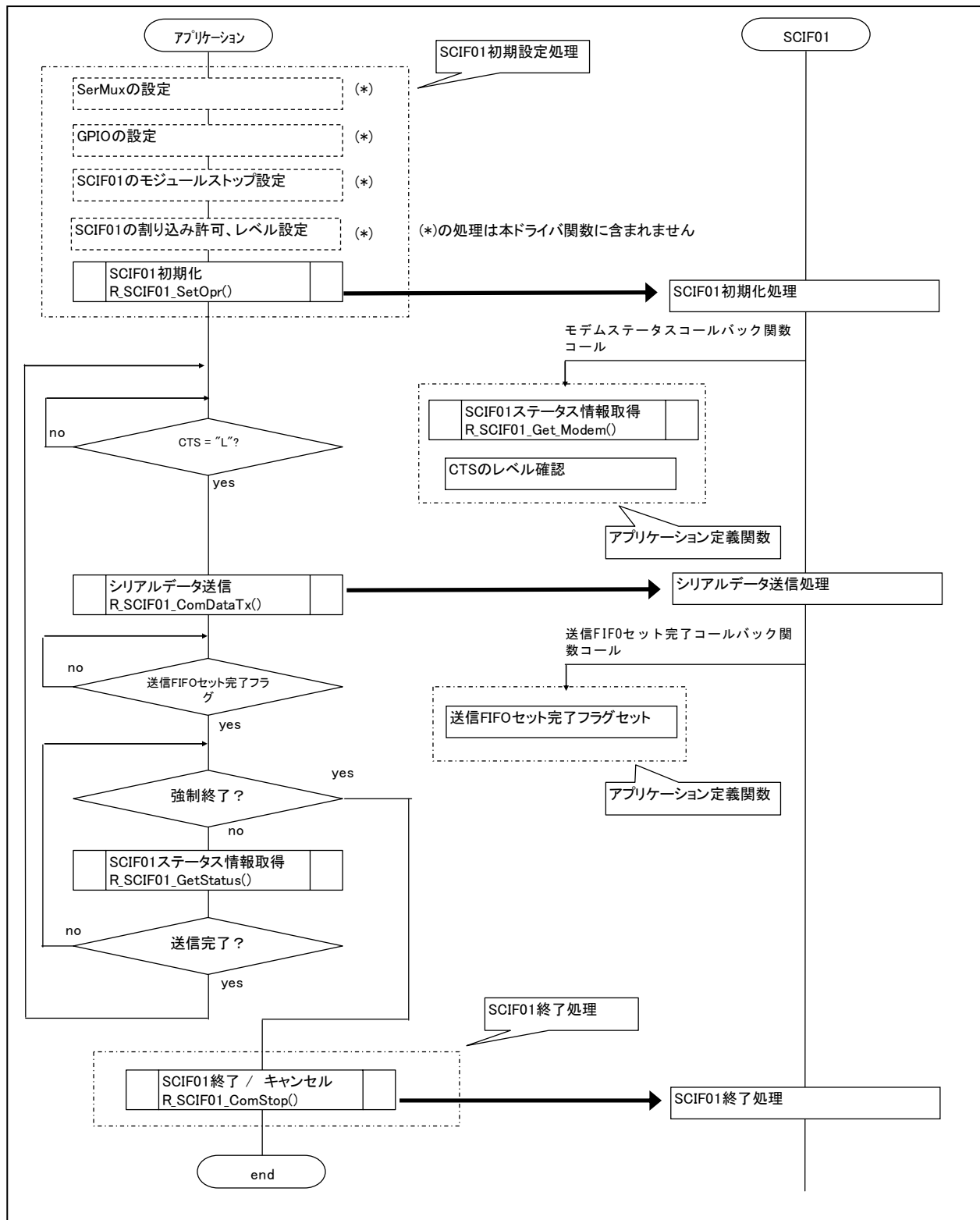
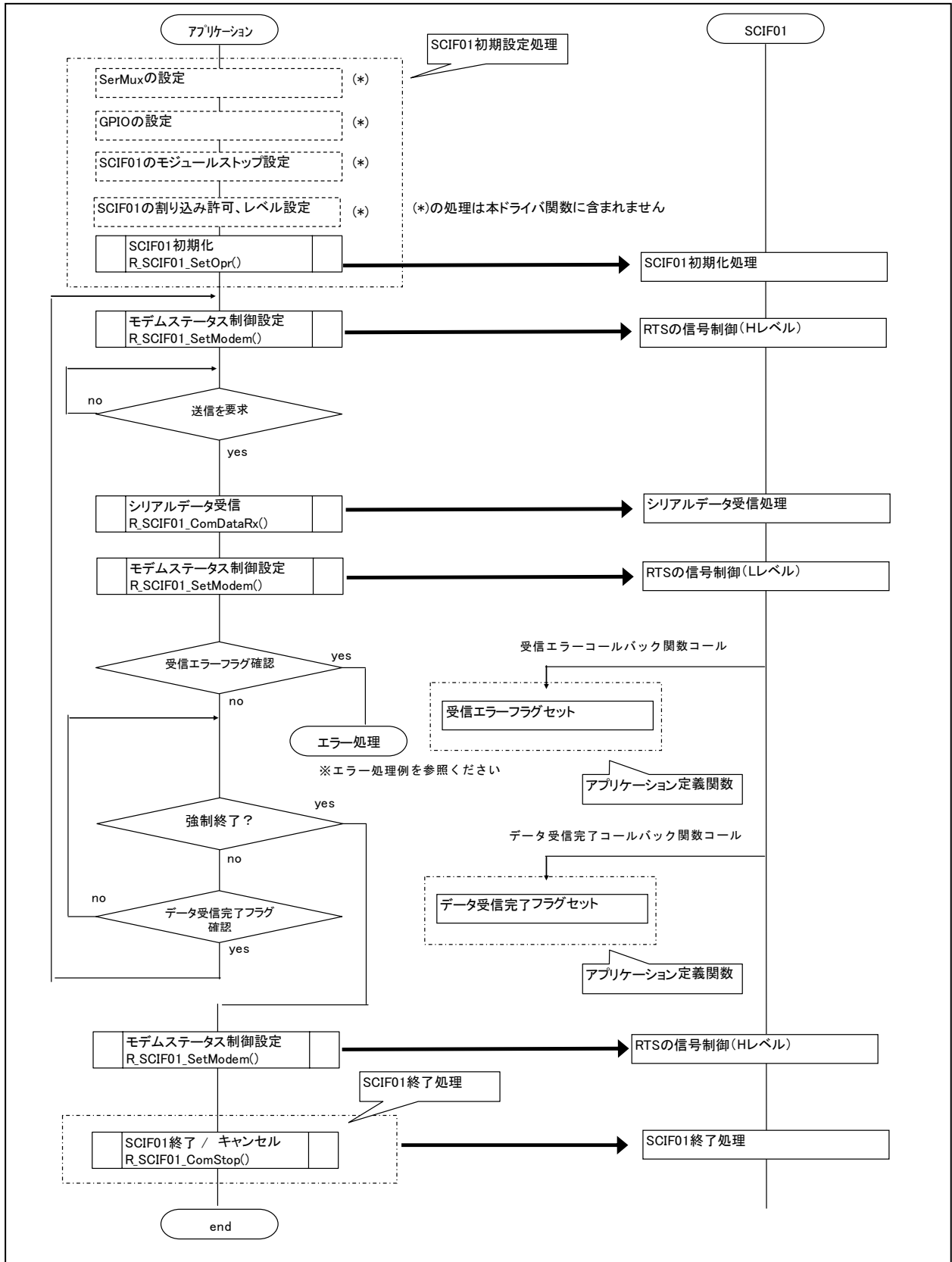


図 3. モデム機能 (CTS) を使用したシリアルデータの送信例図

6.1.4 モデム機能 (RTS) を使用したシリアルデータの受信例

RTS 信号を使用した受信例を示します。RTS 信号を使用する場合、受信するデータ数は” 1 ” (R_SCIF01_ComDataRx () の ul_size) にしてください。



6.1.5 エラー処理例

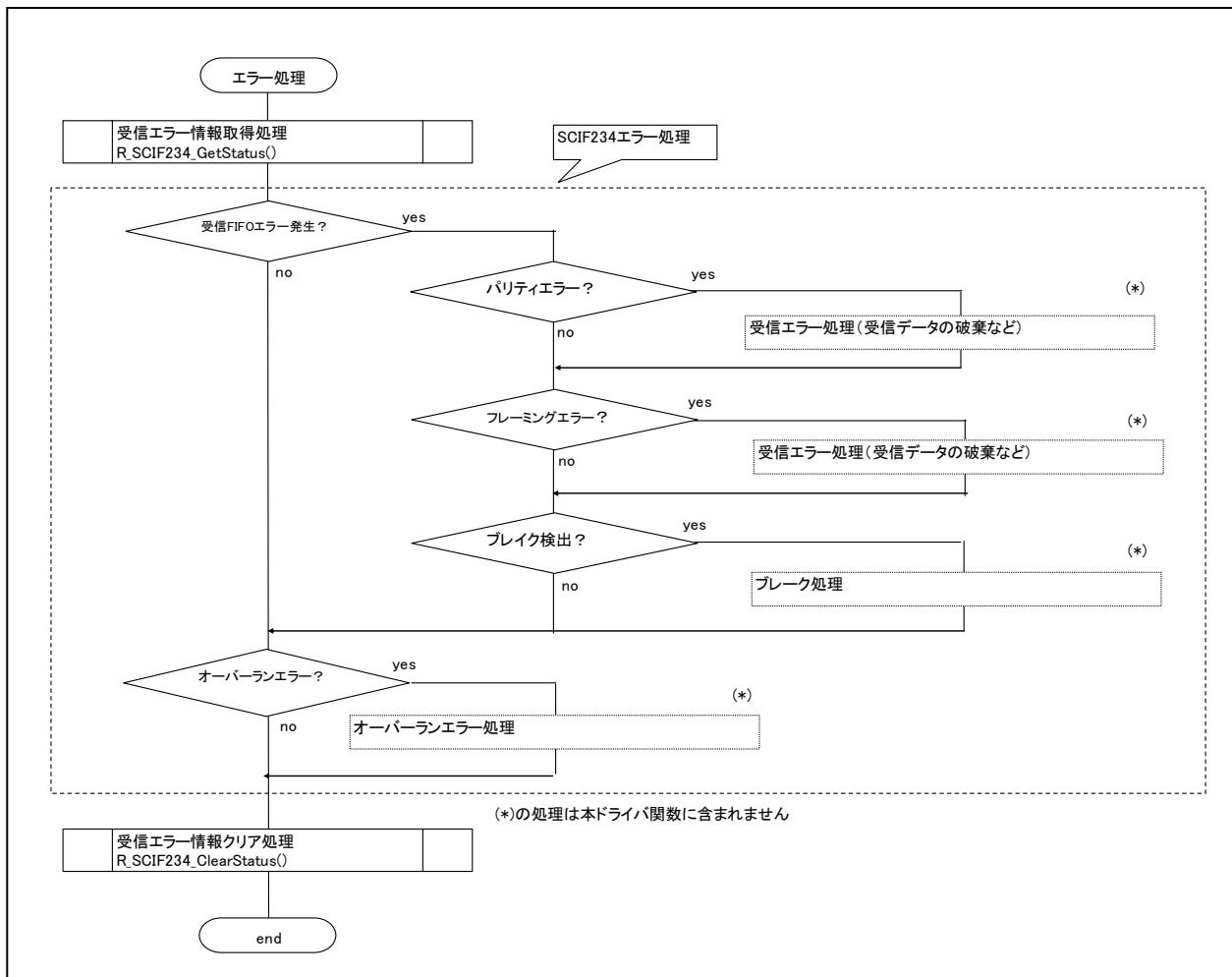


図 5. エラー処理例図

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
<http://japan.renesas.com/>
- お問い合わせ先
<http://japan.renesas.com/inquiry>

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 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・システムの設計において、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て、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。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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い合わせください。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。
9. 本資料に記載されている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・システムに使用することはできません。また、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情報に使用しないで行ってください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>