

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、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、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。

H8/300H Tiny シリーズ

内蔵 SSU 通信動作例

要旨

H8/36057 に内蔵されている SSU (Synchronous Serial communication Unit) モジュールを使用した通信動作例をまとめたものです。

動作確認デバイス

H8/36057

目次

1. SSU 通信システム概要	2
2. 使用機能説明	3
3. 動作説明	5
4. アプリケーション例 (仕様 1 : 標準モード)	8
5. アプリケーション例 (仕様 2 : 双方向モード)	20

1. SSU 通信システム概要

SSU はチップセレクト(\overline{SCS} 端子), クロックライン(SSCK 端子), データライン(SSI および SSO 端子), の 4 本のバスを使用して通信を行います。

1.1 標準モード

- (1) 標準モード時のマスタデバイスは、 \overline{SCS} 端子によりチップセレクト信号、SSCK 端子により転送クロック、SSO 端子によりデータをスレーブに送信します。SSI 端子はスレーブからのデータを受信します。
- (2) 標準モード時のスレーブデバイスは、マスタから送信されてきたチップセレクト信号を \overline{SCS} 端子により受信、転送クロックを SSCK 端子により受信、データを SSO 端子により受信し、マスタから送られてくる転送クロックに同期して、SSI 端子によりデータを送信します。

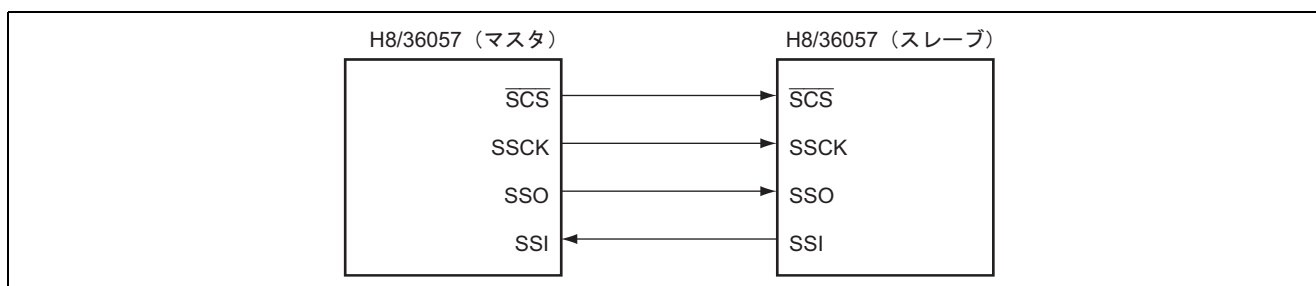


図 1 H8/36057 同士による SSU 通信ブロック図 (標準モード時のマスタ/スレーブ)

1.2 双方向モード

- (1) 双方向モード時のマスタデバイスは、 \overline{SCS} 端子によりチップセレクト信号、SSCK 端子により転送クロックを送信し、SSO 端子によりデータを送受信します。
- (2) 双方向モード時のスレーブデバイスは、マスタから送信されてきたチップセレクト信号を \overline{SCS} 端子により受信、転送クロックを SSCK 端子により受信し、SSO 端子によりマスタから送られてくる転送クロックに同期して、データを送受信します。

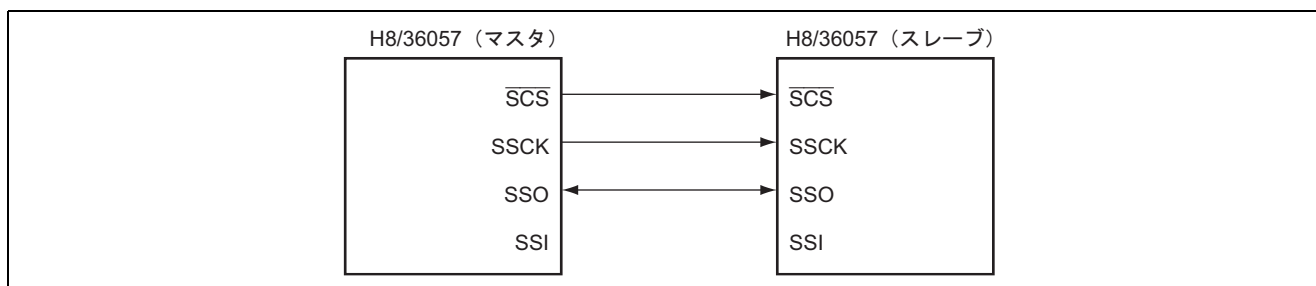


図 2 H8/36057 同士による SSU 通信ブロック図 (双方向モード時のマスタ/スレーブ)

2. 使用機能説明

2.1 送信側

図3に本タスク例で使用するSSUの送信機能ブロック図を示します。

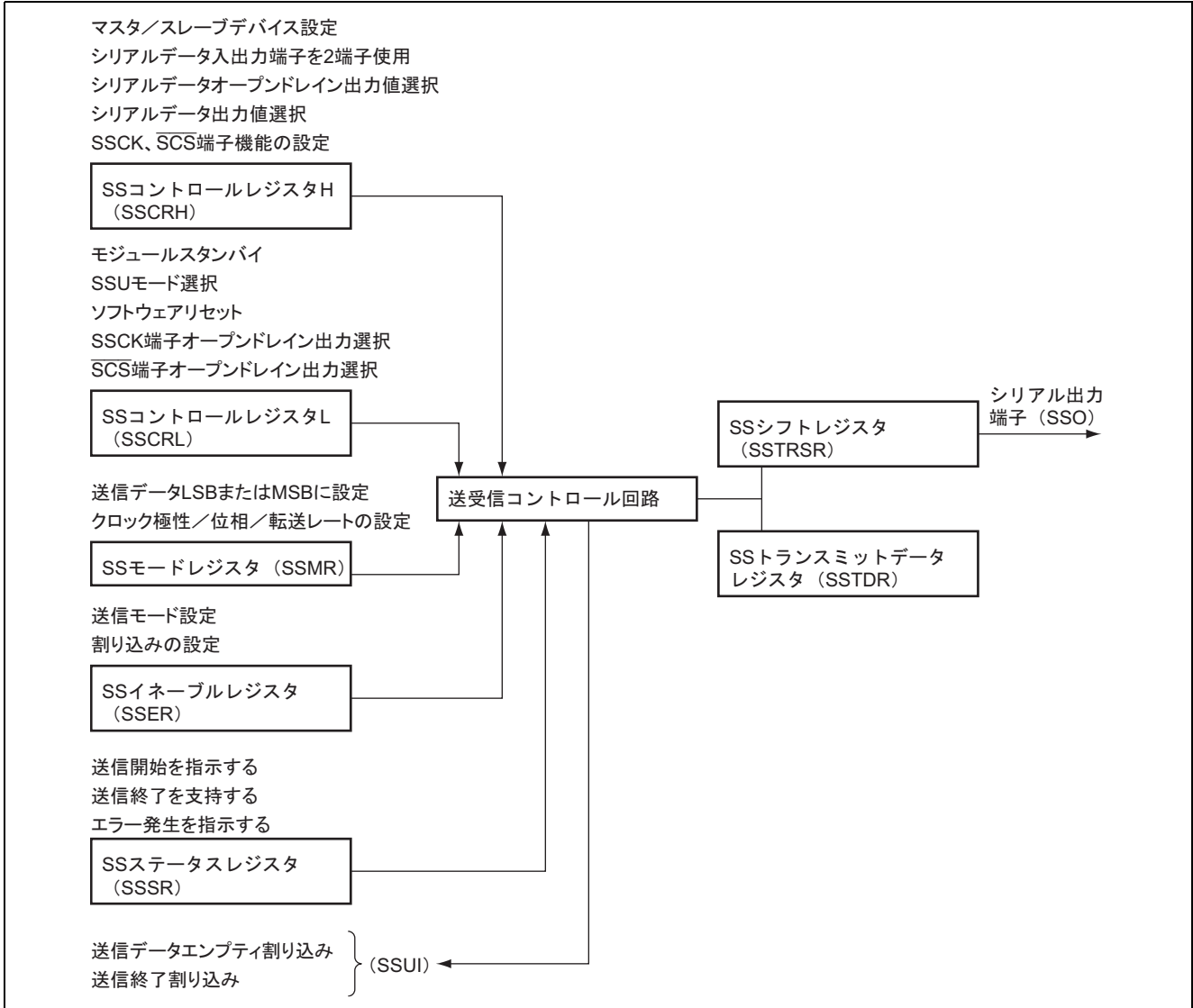


図3 SSU 送信機能ブロック図

2.2 受信側

図 4 に本タスク例で使用する，SSU の受信機能ブロック図を示します。

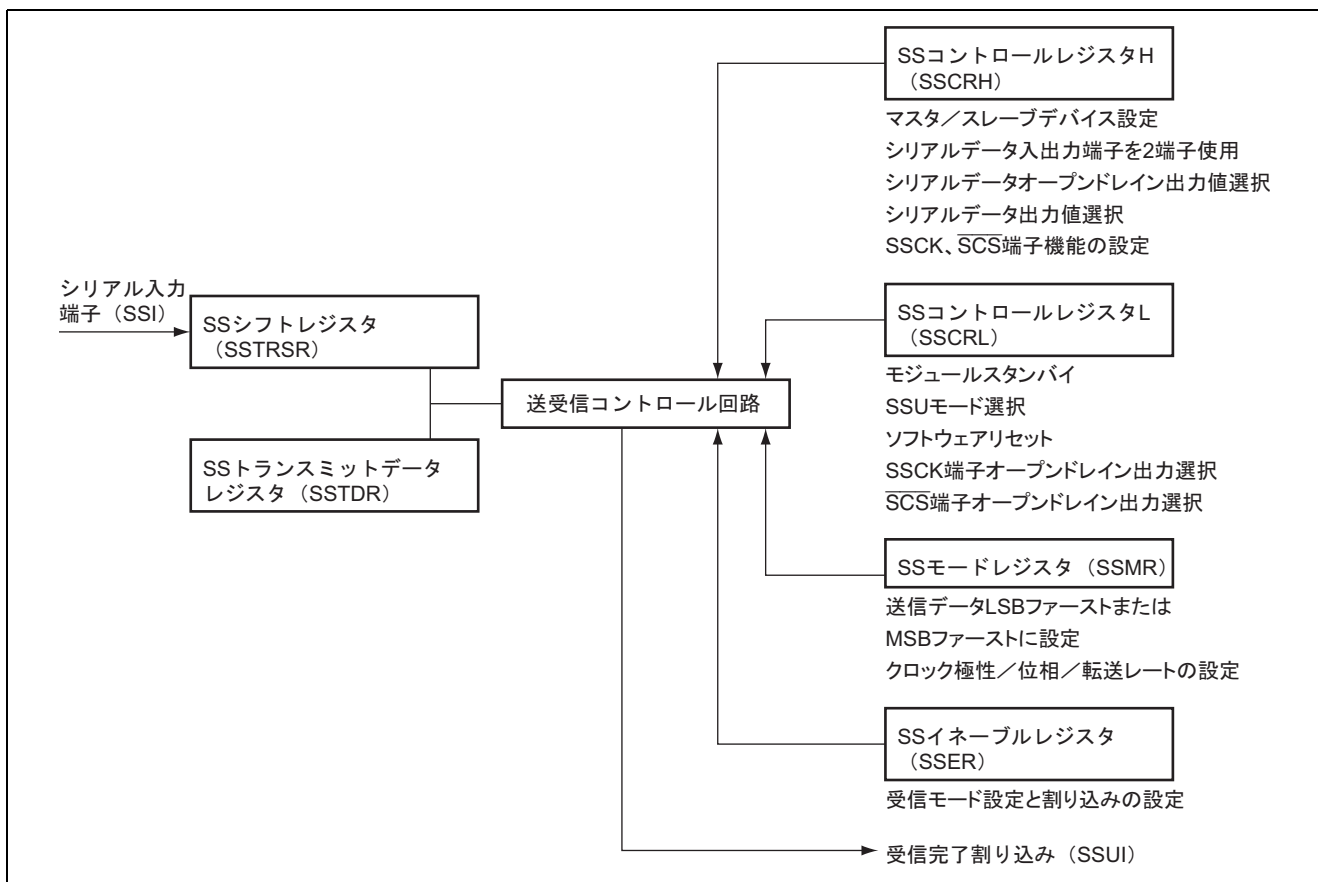


図 4 SSU 受信機能ブロック図

3. 動作説明

3.1 送信動作

(1) 図 5 に H8/36057(マスタ)から，H8/36057(スレーブ)にデータ送信を行なう場合のブロック図を示します。

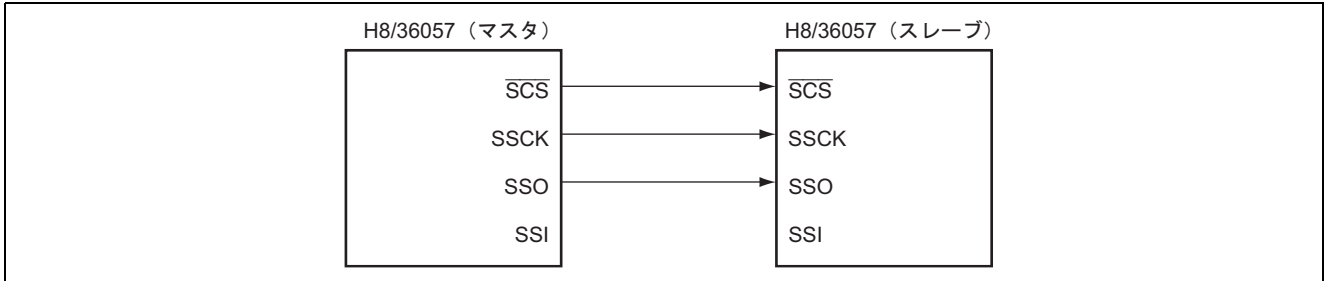


図 5 H8/36057 同士による SSU 通信ブロック図 (マスタ側データ送信時)

- (2) 図 6 に本タスクの送信動作を示します。図 6 に示すタイミングでハードウェア、ソフトウェア処理を行ない、H8/36057(SSU)同士の通信を行ないます。
 図 6 の SSCK 端子のクロックの極性は、アイドル時に High 出力、アクティブ時に Low 出力(CPOS=0)、SSCK クロック位相を、最初のエッジでデータ変化(CPHS=0)、8 ビットデータ長(送信データ H'AA)、LSB ファースト送信に設定した場合の動作を示します。

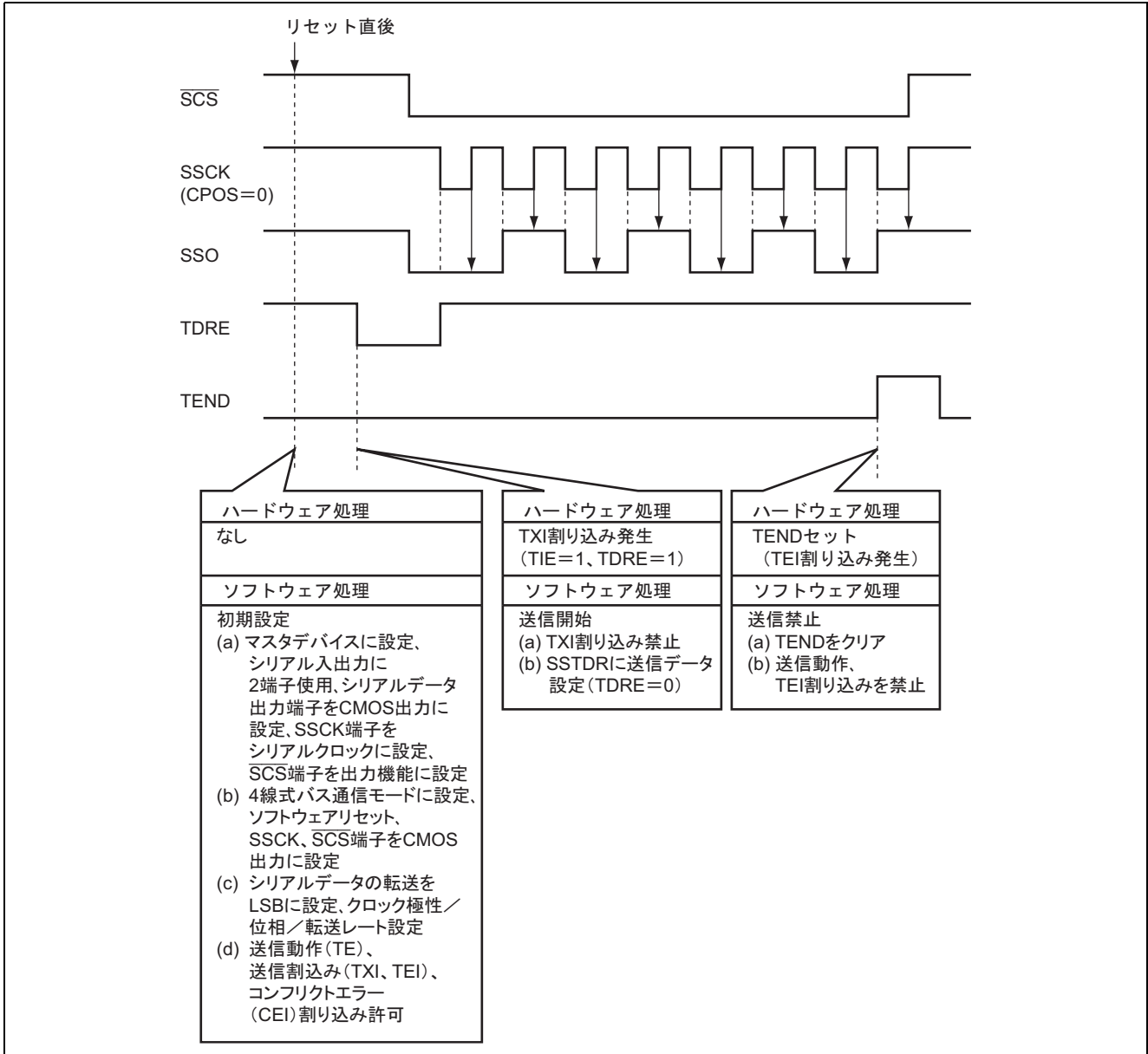


図 6 送信動作

3.2 受信動作

(1) 図 7 に H8/36057(スレーブ)が、H8/36057(マスタ)からデータ受信を行なう場合のブロック図を示します。

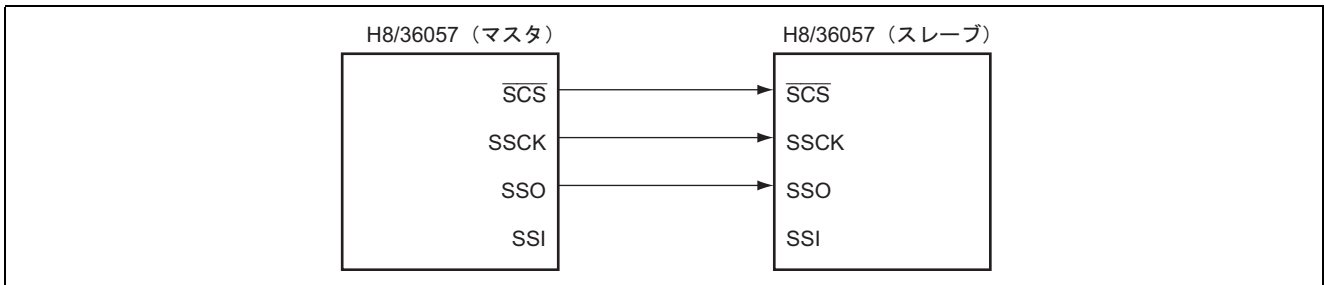


図 7 H8/36057 同士による SSU 通信ブロック図 (スレーブ側データ受信時)

(2) 図 8 に本タスクの受信動作を示します。図 8 に示すタイミングでハードウェア、ソフトウェア処理を行ない、H8/36057(SSU)同士の通信を行ないます。

図 8 は、SSCK 端子のクロックの極性を、アイドル時に High 出力、アクティブ時に Low 出力(CPOS=0)、SSCK クロック位相を最初のエッジでデータ変化(CPHS=0)、8 ビットデータ長(受信データ H'AA)、LSB ファースト受信に設定した場合の動作を示します。

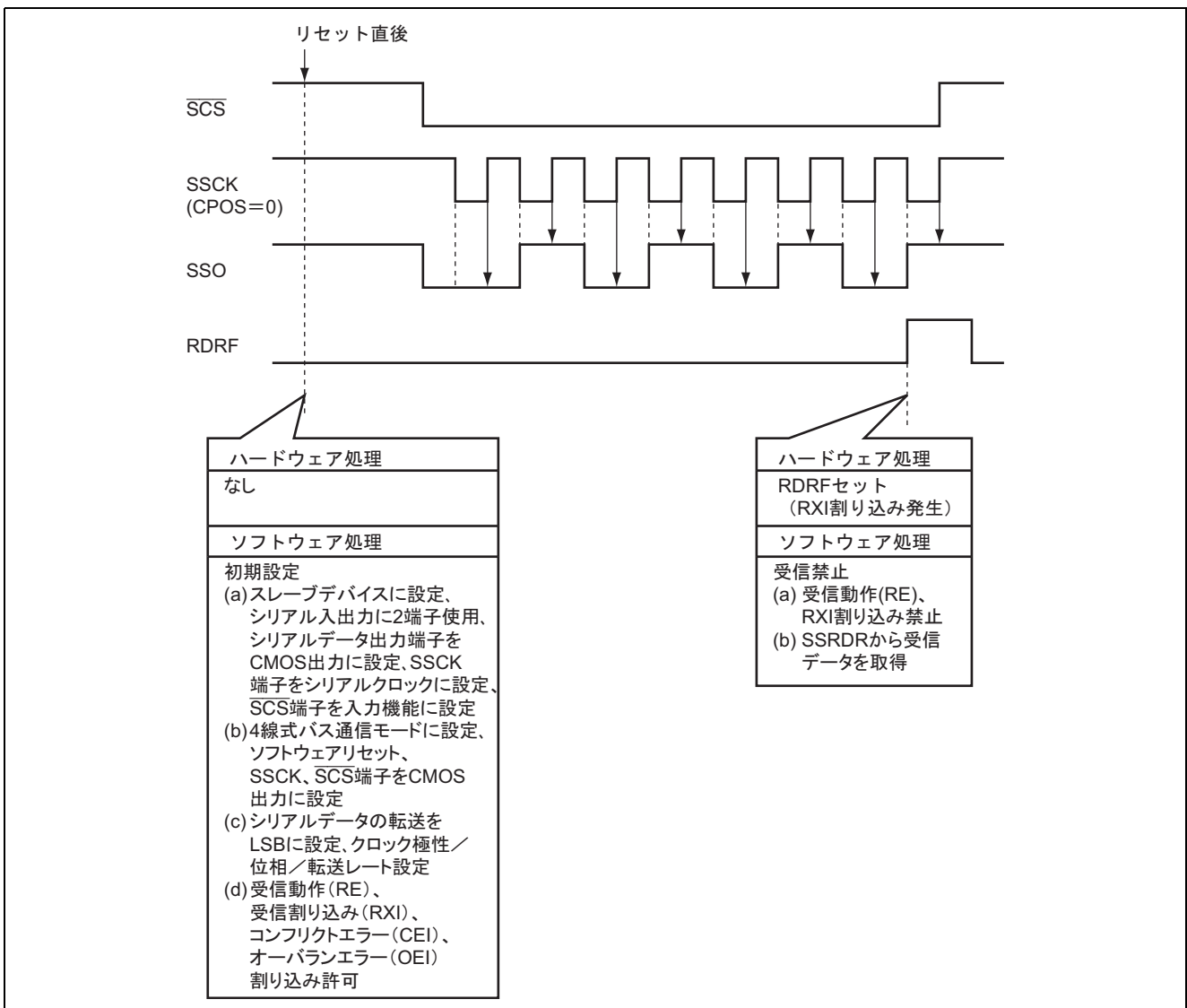


図 8 受信動作

4. アプリケーション例 (仕様 1 : 標準モード)

図 9 に示すように H8/36057($\phi=20\text{MHz}$)同士の間で、データの送受信を標準モードで行ないます(マスタ:送受信, スレーブ:送受信)。

- (1) 転送フォーマットはデータ長 8 ビット(マスタ送信データ H'AA ,スレーブ送信データ H'CC) ,LSB ファースト送信, 転送レートは 1.25Mbps とします。
- (2) シリアルデータの入出力は 2 端子使用(SSO, SSI)します(標準モード)。
- (3) SSCK 端子のクロックの極性を, アイドル時に High 出力, アクティブ時に Low 出力(CPOS=0), SSCK クロック位相は, 最初のエッジでデータ変化(CPHS=0)。

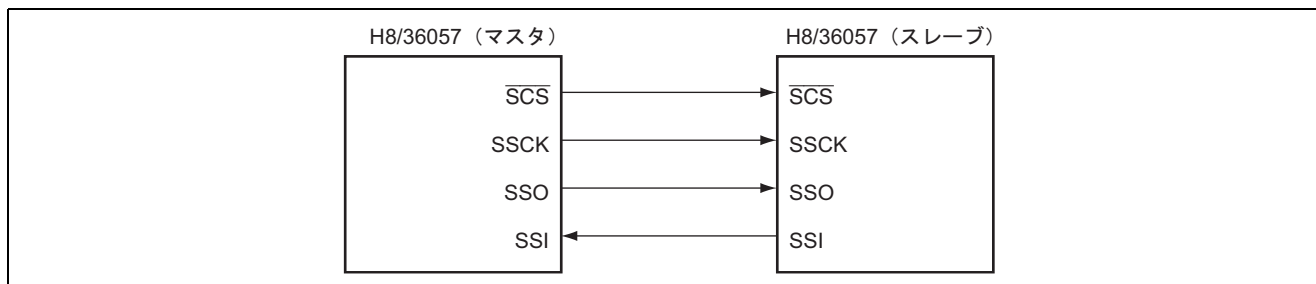


図 9 H8/36057 同士による SSU 通信ブロック図 (マスタ/スレーブ : データ送受信(標準モード))

4.1 機能割り付け

4.1.1 マスタ側

表 1 に本タスクの機能割り付けを示します。H8/36057 の SSU にマスタ機能を割り付け, データの送受信を同時に行ないます。

表 1 H8/36057 機能割り付け (マスタ機能)

H8/36057 機能	機能
SCS	チップセレクト信号を送信。
SSCK	転送クロックを送信。
SSO	H8/36057(スレーブ)へデータを送信。
SSI	H8/36057(スレーブ)からデータを受信。
SSCRH	マスタモードに設定, シリアルデータの入出力を 2 端子使用, シリアルデータを CMOS 出力に設定, $\overline{\text{SCS}}$ 端子=出力機能に設定, SSCK 端子 = シリアルクロック機能に設定。
SSCRL	4 線バス通信モードに設定, SSCK, $\overline{\text{SCS}}$ 端子を CMOS 出力に設定, ソフトウェアリセット。
SSMR	シリアルデータの転送を LSB ファーストに設定, クロック極性・位相・転送レート設定。
SSER	送受信モードの設定, 送受信割込み設定, エラー割込み設定。
SSSR	TDRE, TEND, RDRF ビットにて送受信の状態を示し, CE ビットにてコンフリクトエラー, ORER ビットにてオーバランエラーの発生を示す。
SSTDR	H8/36057(スレーブ)へ送信するデータを設定。
SSRDR	H8/36057(スレーブ)から受信したデータを格納。

4.1.2 スレーブ側

表 2 に本タスクの機能割り付けを示します。H8/36057 の SSU にスレーブ機能を割り付け、データの送受信を同時に行ないます。

表 2 H8/36057 機能割り付け (スレーブ機能)

H8/36057 機能	機能
SCS	チップセレクト信号を受信。
SSCK	転送クロックを受信。
SSO	H8/36057(マスタ)からデータを受信。
SSI	H8/36057(マスタ)へデータを送信。
SSCRH	スレーブモードに設定、シリアルデータの入出力を 2 端子使用、シリアルデータを CMOS 出力に設定、 $\overline{\text{SCS}}$ 端子=入力機能に設定、SSCK 端子 = シリアルクロック機能に設定。
SSCRL	4 線バス通信モードに設定、SSCK, $\overline{\text{SCS}}$ 端子を CMOS 出力に設定、ソフトウェアリセット。
SSMR	シリアルデータの転送を LSB ファーストに設定、クロック極性・位相・転送レート設定。
SSER	送受信モード設定、送受信割込み設定、エラー割込み設定。
SSSR	TDRE, TEND, RDRF ビットにて送受信の状態を示し、CE ビットにてコンフリクトエラー、ORER ビットにてオーバランエラーの発生を示す。
SSTDR	H8/36057(マスタ)へ送信するデータを設定。
SSRDR	H8/36057(マスタ)から受信したデータを格納。

4.2 動作説明 (データ送受信)

図 10 に H8/36057(マスタ、スレーブ)同士の間で、データの送受信を行なう場合のブロック図を示します。(マスタ：送受信，スレーブ：送受信)。

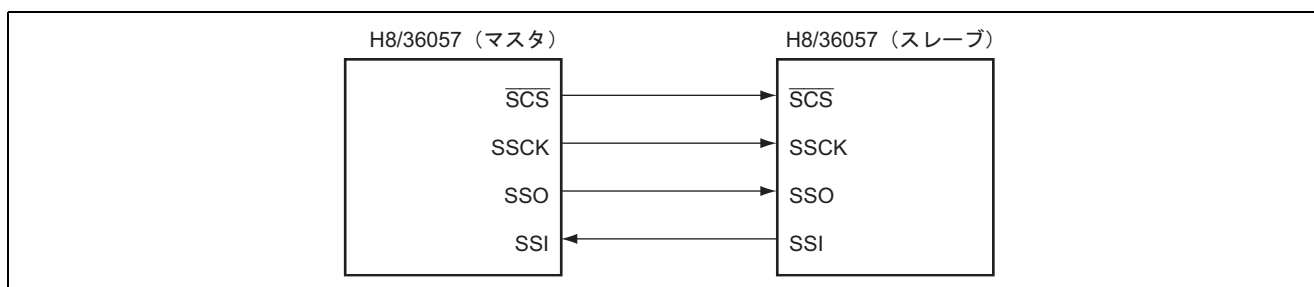


図 10 H8/36057 同士による SSU 通信ブロック図 (マスタ/スレーブ側データ送受信(標準モード時))

図 11 に本タスクの送受信動作を示します。図 11 に示すタイミングでハードウェア、ソフトウェア処理を行ない、H8/36057(SSU)同士の通信を行ないます。

4.2.1 マスタ/スレーブ

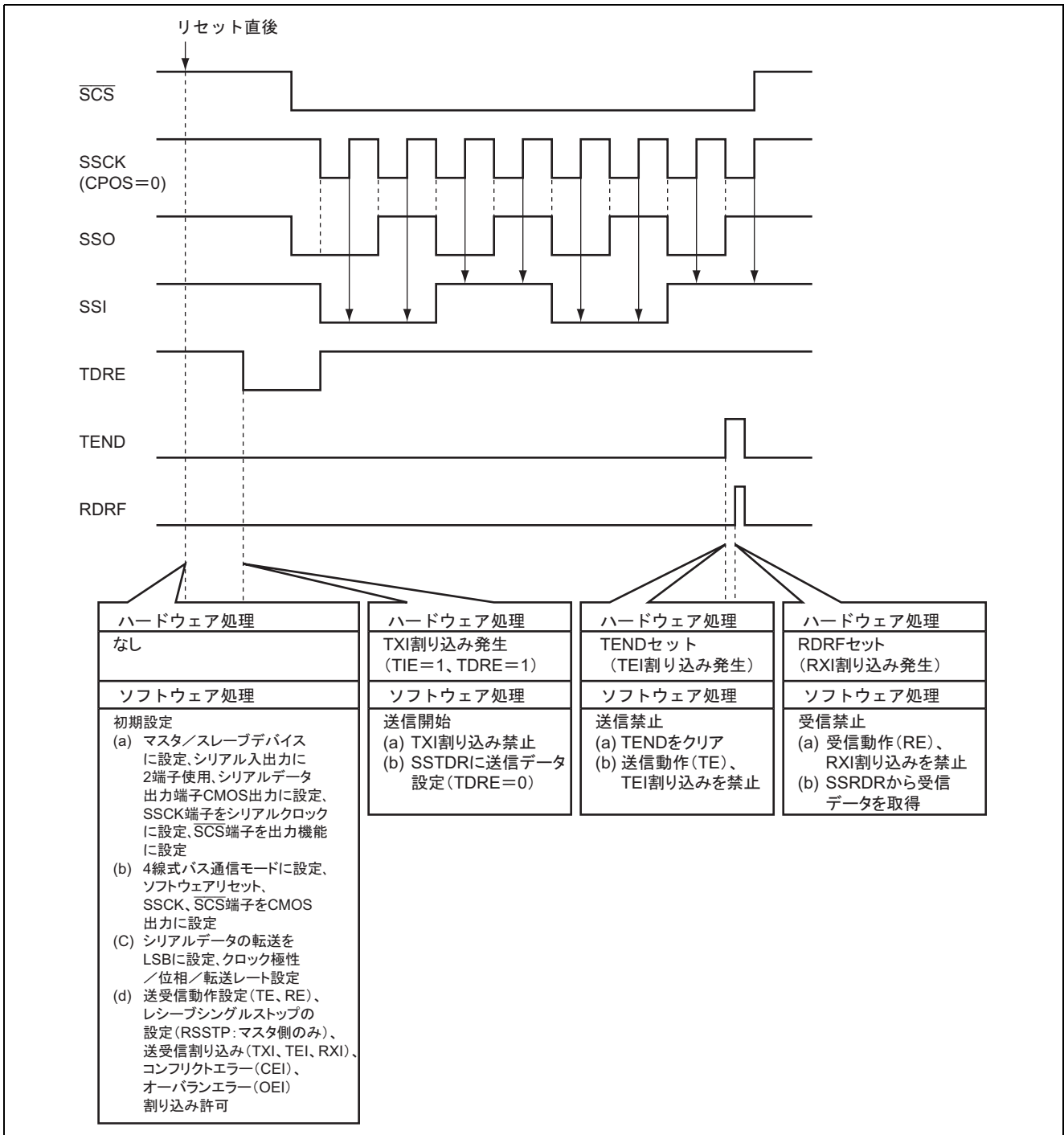


図 11 送受信動作

4.3 ソフトウェア説明(マスタ/スレーブ：送受信機能)

4.3.1 マスタ側

本タスク例のモジュール説明を以下に示します。

(1) 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	マスクレベルを設定します。
初期設定	SSU_init	SSU の初期設定を行ないます。
エラー処理	SSUI_int	CEI 割込みで起動し，エラー処理を行ないます。 OEI 割込みで起動し，エラー処理を行ないます。
データ送信		TXI 割込みで起動し，データ送信を行ないます。
送信終了		TEI 割込みで起動し，データ送信を終了します。
データ受信		RXI 割込みで起動し，データ受信処理を行ないます。

(2) 引数の説明

本タスク例では，引数を使用しません。

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを以下に示します。

レジスタ名	機能	使用モジュール名	設定値
SSCRH	マスタデバイスに設定，シリアル入出力に 2 端子使用，シリアルデータ出力端子，SSCK 端子，SCS 端子の機能を設定します。	初期設定ルーチン	H'8E
SSCRL	バス通信モードを設定，ソフトウェアリセット，SSCK, SCS 端子の機能を設定します。	初期設定ルーチン	H'60
SSMR	シリアルデータの転送を LSB ファーストに設定，クロック極性・位相・転送レート(ϕ クロック入力)の設定をします。	初期設定ルーチン	H'04
SSER	割込み(CEI, OEI, RXI, TEI, TXI)を許可し，SSU の送受信許可/禁止を設定します。	初期設定ルーチン	H'EF
SSTDR	受信局 H8/36057 へ送信するデータを設定します。	データ送信	H'AA
SSRDR	送信局 H8/36057 からのデータを格納します。	データ受信	
SSSR	エラーを検出します。	エラー処理	

(4) 使用 RAM 説明

ラベル名	使用モジュール名	データ長	機能
Rxdata	データ受信	unsigned char	受信したデータを格納します。

4.3.2 スレーブ側

(1) 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	マスクレベルを設定します。
初期設定	SSU_init	SSU の初期設定を行ないます。
エラー処理	SSUI_int	CEI 割込みで起動し，エラー処理を行ないます。 OEI 割込みで起動し，エラー処理を行ないます。
データ受信		RXI 割込みで起動し，データ受信処理を行ないます。
送信終了		TEI 割込みで起動し，データ送信を終了します。
データ送信		TXI 割込みで起動し，データ送信を行ないます。

(2) 引数の説明

本タスク例では，引数を使用しません。

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを以下に示します。

レジスタ名	機能	使用モジュール名	設定値
SSCRH	スレーブデバイスに設定，シリアル入出力に 2 端子使用，シリアルデータ出力端子，SSCK 端子，SCS 端子の機能を設定します。	初期設定ルーチン	H'0D
SSCRL	バス通信モードを設定，ソフトウェアリセット，SSCK，SCS 端子の機能を設定します。	初期設定ルーチン	H'60
SSMR	シリアルデータの転送を LSB に設定，クロック極性・位相・転送レート (φクロック入力)の選択。	初期設定ルーチン	H'04
SSER	割込み(CEI, OEI, RXI, TEI, TXI)を許可し，SSU の送受信許可/禁止を設定します。	初期設定ルーチン	H'CF
SSRDR	送信局 H8/36057 からのデータを設定します。	データ受信	
SSTDR	受信局 H8/36057 へ送信するデータを設定します。	データ送信	H'CC
SSSR	エラーを検出します。	エラー処理	

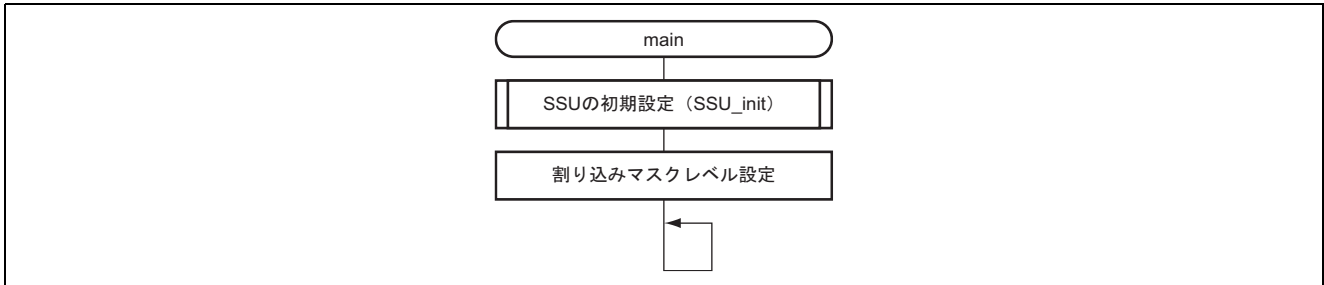
(4) 使用 RAM 説明

ラベル名	使用モジュール名	データ長	機能
Rxdata	データ受信	unsigned char	受信したデータを格納します。

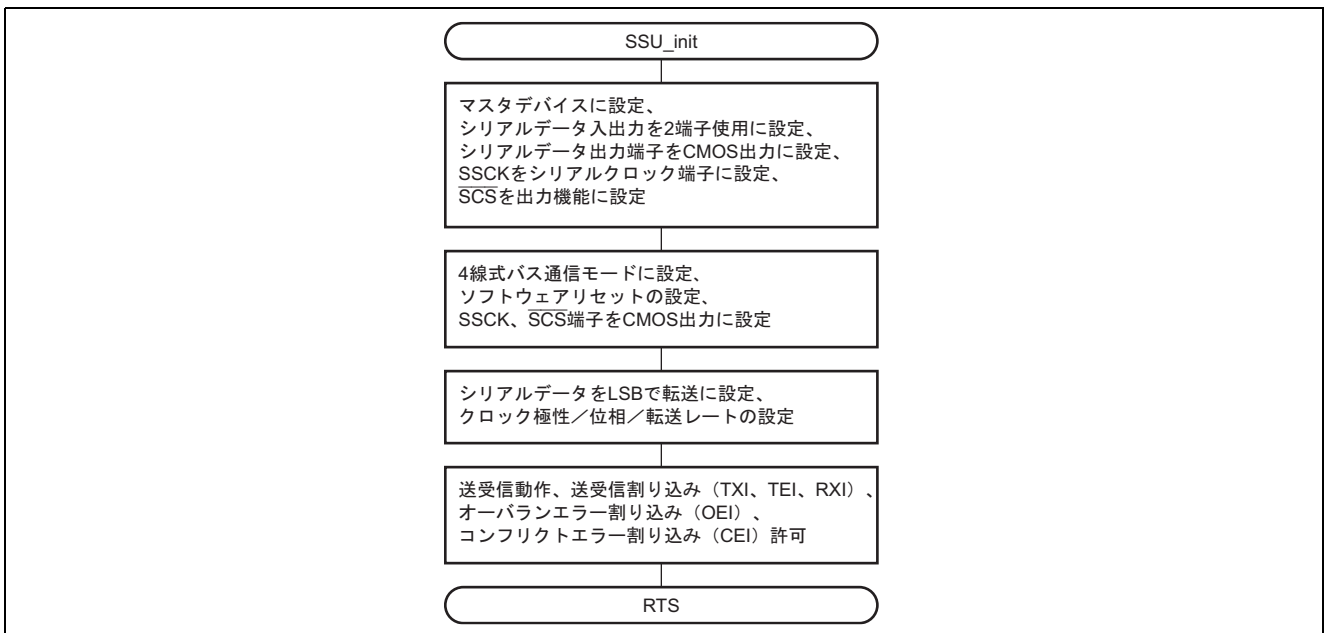
4.4 フローチャート

4.4.1 マスタ側

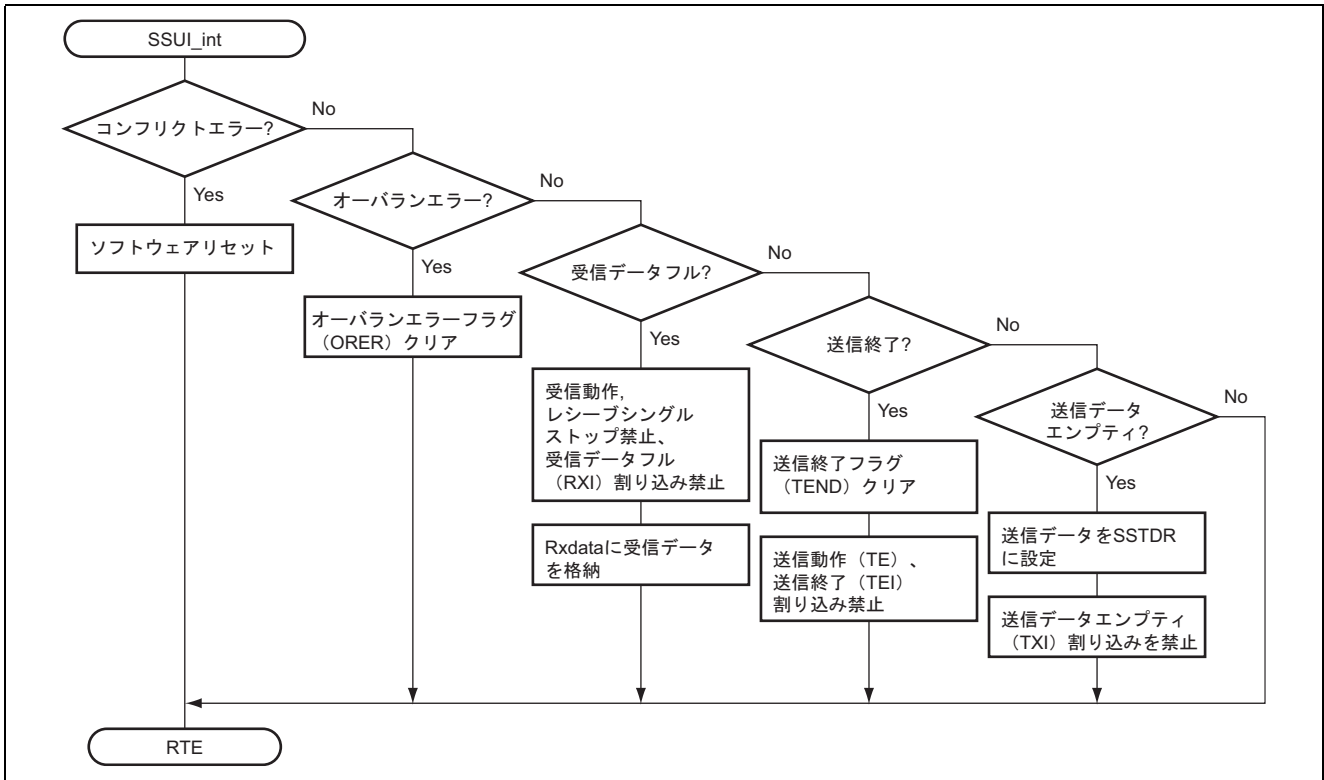
(1) メインルーチン



(2) 初期設定

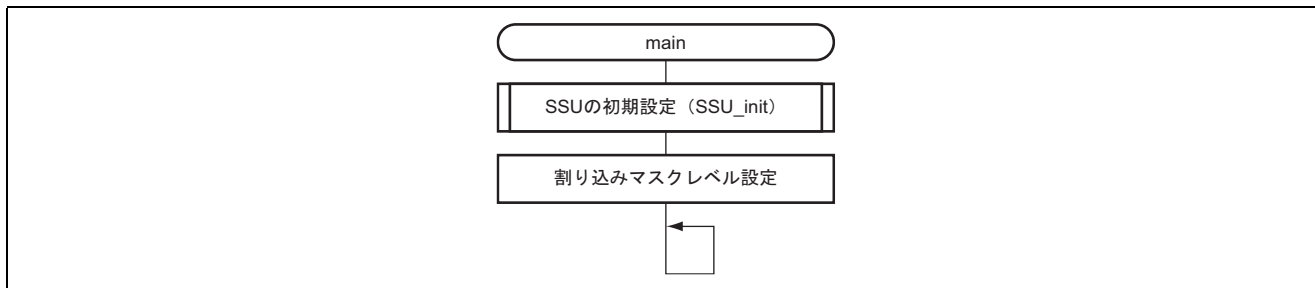


(3) エラー発生，データ受信，データ送信

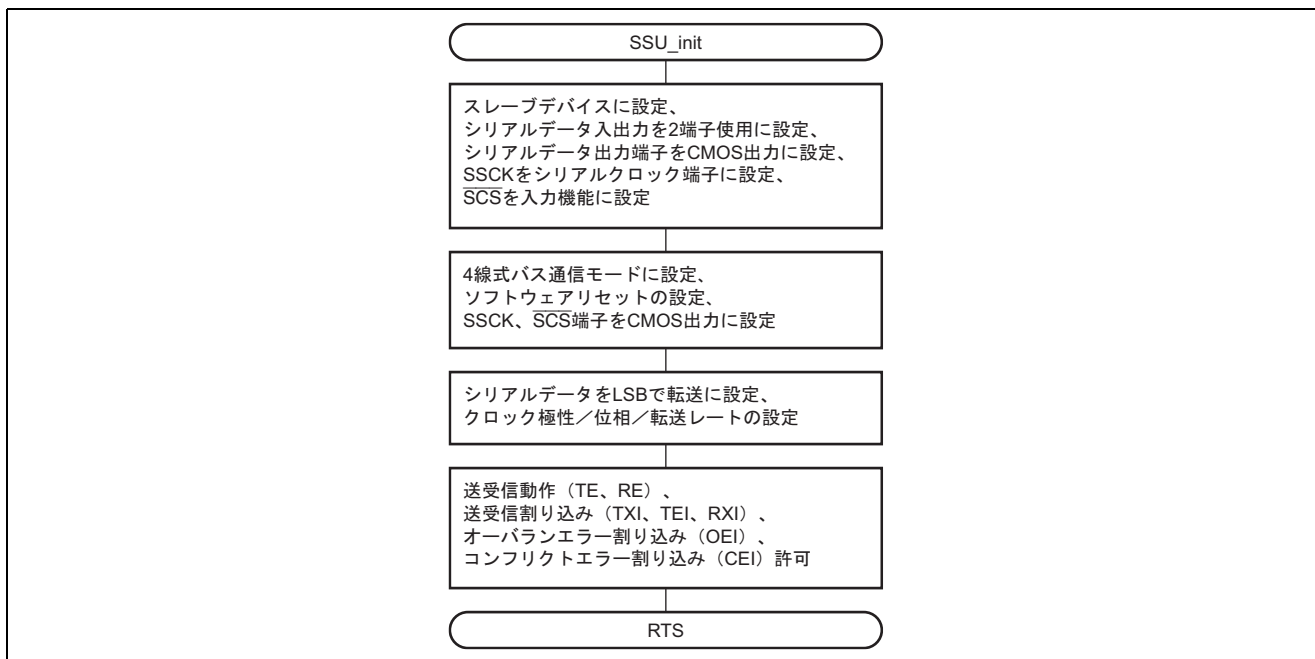


4.4.2 スレーブ側

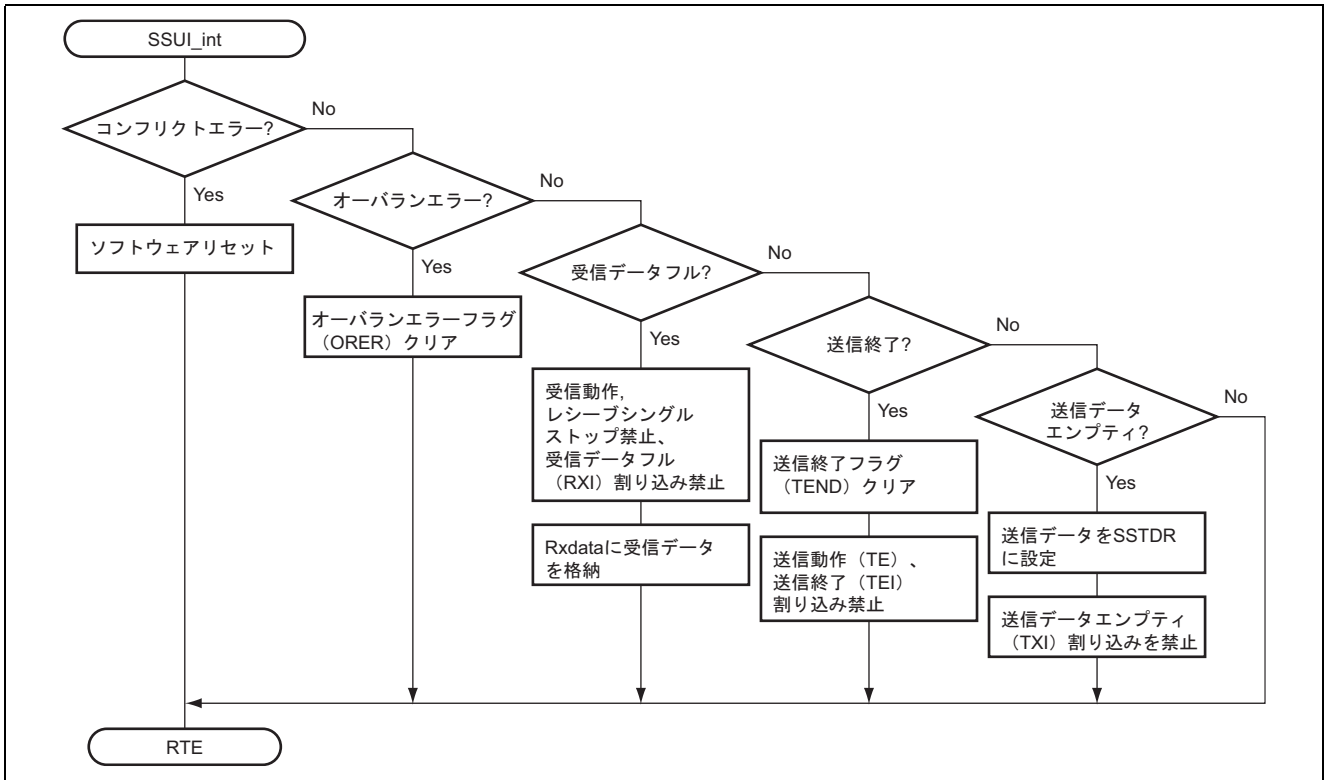
(1) メインルーチン



(2) 初期設定



(3) エラー発生, データ受信, データ送信



4.5 プログラムリスト

4.5.1 マスタ側

```

#include <machine.h>
#include "H8_36057.h"
void main (void);
void SSU_init (void);
unsigned char Rxdata;
/*****
/* Main Routine
*****/
void main (void)
#pragma asm
    mov.l #H'FFFFFF7C,SP
#pragma endasm
{
    SSU_init();
    set_imask_ccr(0);
    while(1);
}
/*****
/* SSU Module Initialize Routine
*****/
void SSU_init(void)
{
    SSU.SSER.BYTE = 0x00;
    SSU.SSCRL.BYTE = 0x00;
    SSU.SSCRH.BYTE = 0x8E;
    SSU.SSCRL.BYTE = 0x60;
    SSU.SSMR.BYTE = 0x04;
    SSU.SSSR.BYTE &= 0x04;
    SSU.SSER.BYTE = 0xEF;
}
/*****
#pragma interrupt (SSUI_int)      /* エラー割込み,受信割込み,送信割込み
*****/
void SSUI_int(void)
{
    if(SSU.SSSR.BIT.CE && SSU.SSER.BIT.CEIE){
        SSU.SSSR.BIT.CE = 0;
        SSU.SSCRL.BIT.SRES = 1;
    }
    else if(SSU.SSSR.BIT.ORER && SSU.SSER.BIT.RIE){
        SSU.SSSR.BIT.ORER = 0;
    }
    else if(SSU.SSSR.BIT.RDRF && SSU.SSER.BIT.RIE){
        SSU.SSER.BYTE &= 0x8D;
        Rxdata = SSU.SSRDR;
    }
    else if(SSU.SSSR.BIT.TEND && SSU.SSER.BIT.TEIE){
        SSU.SSSR.BIT.TEND = 0;
        SSU.SSER.BYTE &= 0x63;
    }
    else if(SSU.SSSR.BIT.TDRE && SSU.SSER.BIT.TIE){
        SSU.SSTDR = 0xAA;
        SSU.SSER.BIT.TIE = 0;
    }
}
}

```

4.5.2 スレーブ側

```

#include <machine.h>
#include "H8_36057.h"
void main (void);
void SSU_init (void);
unsigned char Rxdata;
/*****
/* Main Routine
*****/
void main (void)
#pragma asm
    mov.l #H'FFFFFF7C,SP
#pragma endasm
{
    SSU_init();
    set_imask_ccr(0);
    while(1);
}
/*****
/* SSU Module Initialize Routine
*****/
void SSU_init(void)
{
    SSU.SSER.BYTE = 0x00;
    SSU.SSCRL.BYTE = 0x00;
    SSU.SSCRH.BYTE = 0x0D;
    SSU.SSCRL.BYTE = 0x60;
    SSU.SSMR.BYTE = 0x04;
    SSU.SSSR.BYTE &= 0x04;
    SSU.SSER.BYTE = 0xCF;
}
/*****
#pragma interrupt (SSUI_int)      /* エラー割込み,受信割込み,送信割込み
*****/
void SSUI_int(void)
{
    if(SSU.SSSR.BIT.CE && SSU.SSER.BIT.CEIE){
        SSU.SSSR.BIT.CE = 0;
        SSU.SSCRL.BIT.SRES = 1;
    }
    else if(SSU.SSSR.BIT.ORER && SSU.SSER.BIT.RIE){
        SSU.SSSR.BIT.ORER = 0;
    }
    else if(SSU.SSSR.BIT.RDRF && SSU.SSER.BIT.RIE){
        SSU.SSER.BYTE &= 0x8D;
        Rxdata = SSU.SSRDR;
    }
    else if(SSU.SSSR.BIT.TEND && SSU.SSER.BIT.TEIE){
        SSU.SSSR.BIT.TEND = 0;
        SSU.SSER.BYTE &= 0x63;
    }
    else if(SSU.SSSR.BIT.TDRE && SSU.SSER.BIT.TIE){
        SSU.SSTDR = 0xCC;
        SSU.SSER.BIT.TIE = 0;
    }
}
}
    
```

4.6 マスタ：送信/スレーブ：受信動作時のプログラム

4.6.1 マスタ側設定

マスタ側より 8bit データを 1 回のみ送信するプログラムは、前述マスタ側プログラムリスト内のラベル名 main, SU_init(SSER を H'8D に設定変更), SSUI_int(CEI, EI, XI)を使用します。

4.6.2 スレーブ側設定

スレーブ側で 8bit データを 1 回受信のみするプログラムは、前述スレーブ側プログラムリスト内のラベル名 main, SSU_int(SSER を H'43 に設定変更), SUI_int(CEI, EI, XI)を使用します。

4.7 マスタ： \overline{SCS} , SSCK 送信, データ受信, スレーブ：データ送信時のプログラム

4.7.1 マスタ側設定

マスタ側より SSCK, \overline{SCS} のみを送信(8bit データを 1 回受信するため)するプログラムは、前述マスタ側プログラムリスト内のラベル名 main, SSU_init(SSER を H'63 に設定更) ,SSUI_int(CEI, OEI, RXI)を使用します。

4.7.2 スレーブ側設定

スレーブ側でデータ送信(8bit データ)のみをするプログラムは、前述スレーブ側プログラムリスト内のラベル名 main, SSU_init(SSER を H'8D に設定変更), SSUI_int(CEI, TEI, TXI)を使用します。

5. アプリケーション例 (仕様 2 : 双方向モード)

図 12 に示すように H8/36057($\phi=20\text{MHz}$)同士の間で、データの送受信を双方向モードで行ないます(マスタ：送受信，スレーブ：送受信)。

- 転送フォーマットはデータ長 16 ビット(マスタ/スレーブ送信データ H'AA, H'55)MSB ファースト送信，転送レートは 0.08Mbps とします。
- シリアルデータの入出力は 1 端子使用(SSO)します(双方向モード)。
- SSCK 端子のクロックの極性を，アイドル時に Low 出力，アクティブ時に High 出力(CPOS=1)，SSCK クロック位相を，最初のエッジでデータラッチ(CPHS=1)。

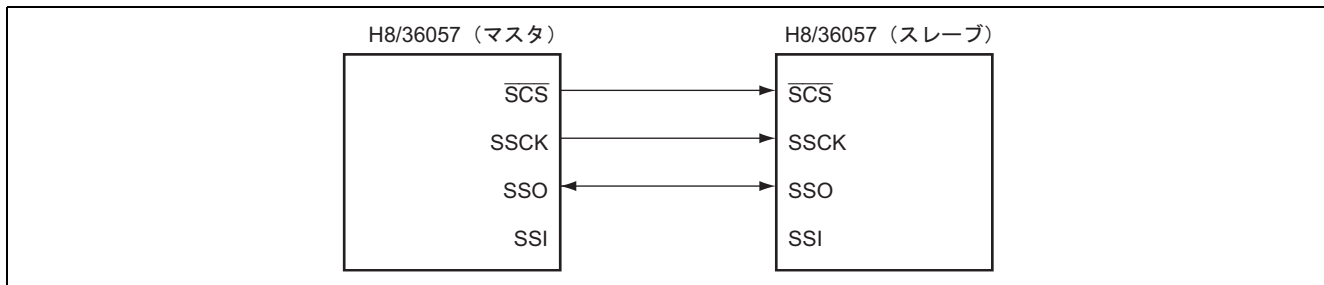


図 12 H8/36057 同士による SSU 通信ブロック図 (マスタ/スレーブ：データ送受信(双方向モード))

5.1 機能割り付け

5.1.1 マスタ側

表 3 に本タスクの機能割り付けを示します。表 3 は、H8/36057 の SSU にマスタ機能を割り付け、データの送受信を行ないます。

表 3 H8/36057 機能割り付け (マスタ機能)

H8/36057 機能	機能
SCS	チップセレクト信号を送信。
SSCK	転送クロックを送信。
SSO	H8/36057(スレーブ)とのデータ送受信。
SSCRH	マスタモードに設定、シリアルデータの入出力を 1 端子使用、シリアルデータを CMOS 出力に設定、SCS 端子 = 出力機能に設定、SSCK 端子 = シリアルクロック機能に設定。
SSCRL	4 線バス通信モードに設定、SSCK、SCS 端子を CMOS 出力に設定、ソフトウェアリセット。
SSMR	シリアルデータの転送を MSB ファーストに設定、クロック極性・位相・転送レートを設定。
SSER	送受信モードの設定、送受信割込み設定、エラー割込み設定。
SSSR	TDRE、TEND、RDRF ビットにて送受信の状態を示し、CE ビットにてコンフリクトエラー、ORER ビットにてオーバランエラーの発生を示す。
SSTDR	H8/36057(スレーブ)へ送信するデータを設定。
SSRDR	H8/36057(スレーブ)から受信したデータを格納。

5.1.2 スレーブ側

表 4 に本タスクの機能割り付けを示します。H8/36057 の SSU にスレーブ機能を割り付け、データの送受信を行ないます。

表 4 H8/36057 機能割り付け (スレーブ機能)

H8/36057 機能	機能
SCS	チップセレクト信号を受信。
SSCK	転送クロックを受信。
SSO	H8/36057(マスタ)とのデータを送受信。
SSCRH	スレーブモードに設定、シリアルデータの入出力を 1 端子使用、シリアルデータを CMOS 出力に設定、SCS 端子 = 入力機能に設定、SSCK 端子 = シリアルクロック機能に設定。
SSCRL	4 線バス通信モードに設定、SSCK、SCS 端子を CMOS 出力に設定、ソフトウェアリセット。
SSMR	シリアルデータの転送を MSB ファーストに設定、クロック極性・位相・転送レート設定。
SSER	送受信モード設定、送受信割込み設定、エラー割込み設定。
SSSR	TDRE、TEND、RDRF ビットにて送受信の状態を示し、CE ビットにてコンフリクトエラー、ORER ビットにてオーバランエラーの発生を示す。
SSTDR	H8/36057(マスタ)へ送信するデータを設定。
SSRDR	H8/36057(マスタ)から受信したデータを格納。

5.2 動作説明 (データ送受信)

図 13 に H8/36057(マスタ, スレーブ)同士の間で, データの送受信を行なう場合のブロック図を示します。(マスタ: 送受信, スレーブ: 送受信)。

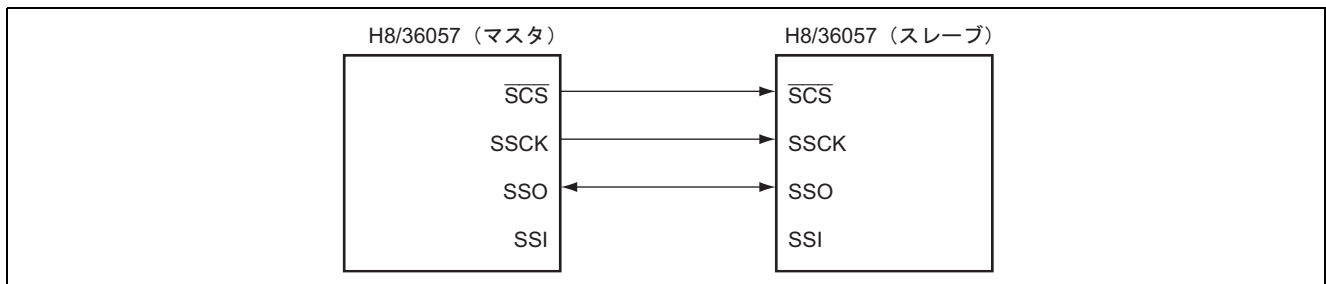


図 13 H8/36057 同士による SSU 通信ブロック図 (マスタ, スレーブ側データ送受信(双方向モード時))

図 14, 15 に本タスクの送受信動作原理を示します。図 14, 15 に示すタイミングでハードウェアソフトウェア処理を行ない, H8/36057(SSU)同士の通信を行ないます。

5.2.1 マスタ側 (*スレーブ側処理)

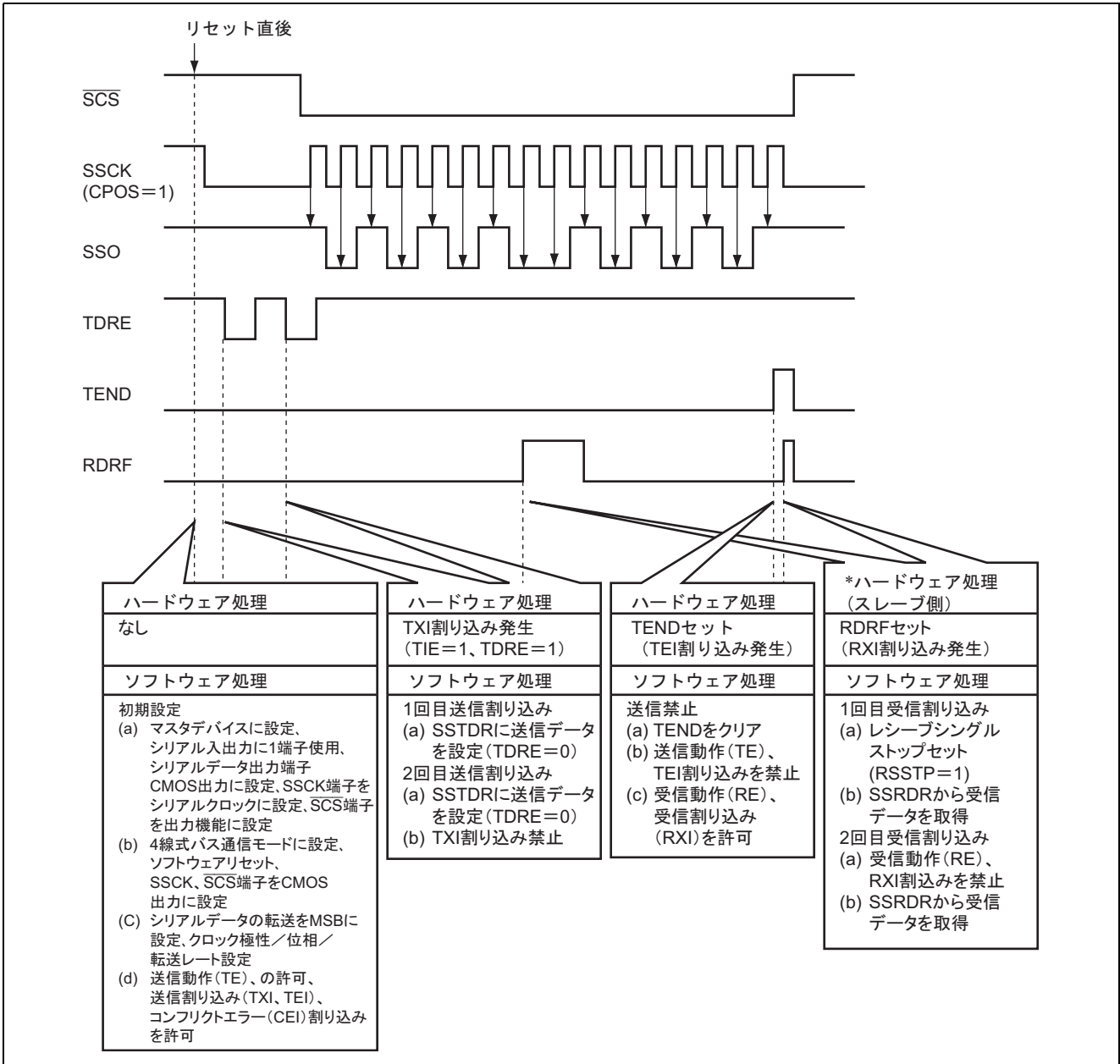


図 14 マスタ側送受信動作原理 (双方向モード時)

5.2.2 スレーブ側 (*マスタ側処理)

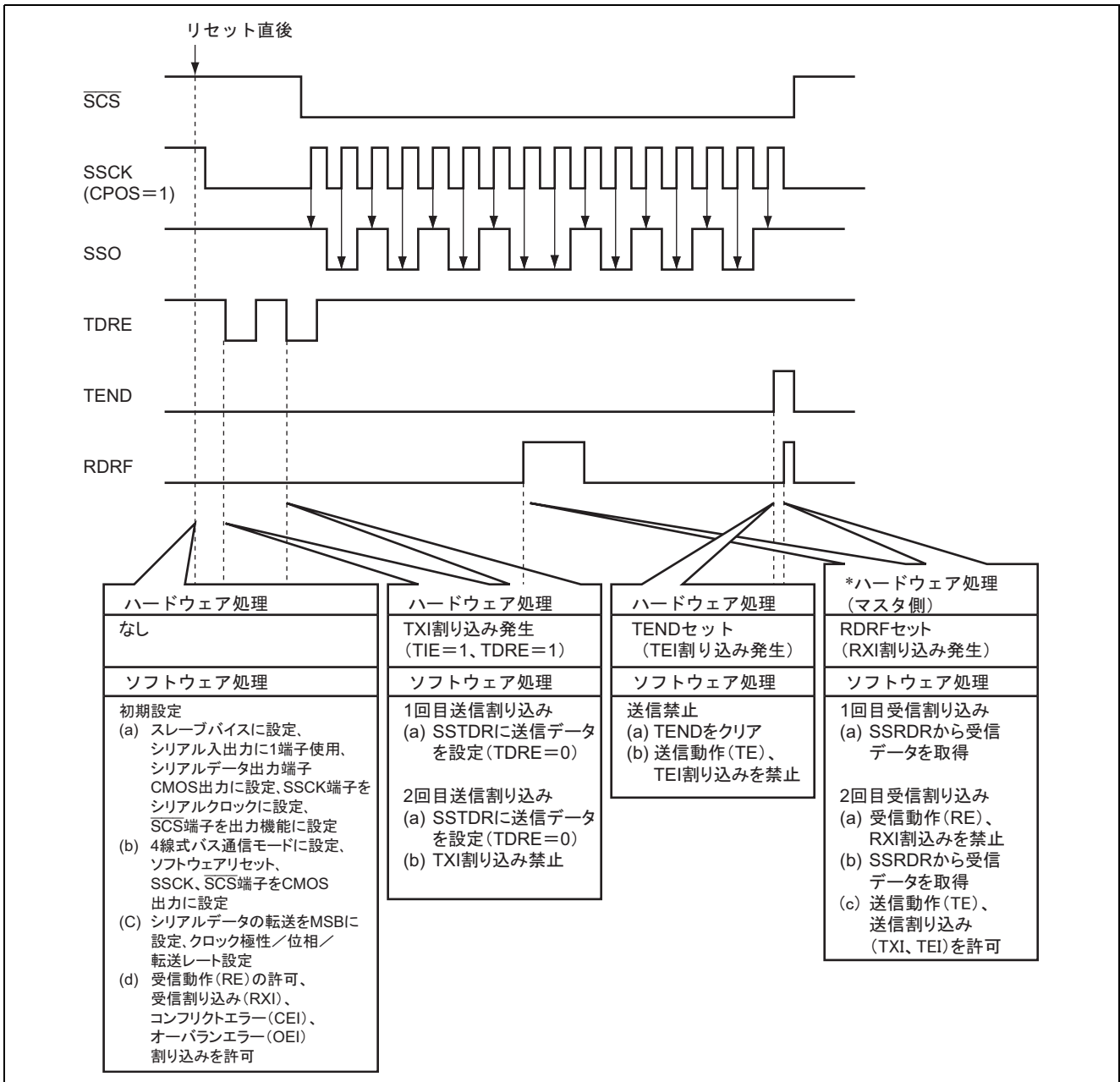


図 15 スレーブ側送受信動作原理 (双方向モード時)

5.3 ソフトウェア説明 (マスタ/スレーブ : 送受信機能)

5.3.1 マスタ側

本タスク例のモジュール説明を以下に示します。

(1) 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	マスクレベルを設定します。
初期設定	SSU_init	SSU の初期設定を行ないます。
変数初期化	SSU_counter_init	送受信データのデータレングスをカウントします。
エラー処理 データ受信 送信終了 データ送信	SSUI_int	CEI 割込みで起動し, エラー処理を行ないます。 OEI 割込みで起動し, エラー処理を行ないます。 RXI 割込みで起動し, データ受信処理を行ないます。 TEI 割込みで起動し, データ送信を終了します。 TXI 割込みで起動し, データ送信を行ないます。

(2) 引数の説明

本タスク例では, 引数を使用しません。

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを以下に示します。

レジスタ名	機能	使用モジュール名	設定値
SSCRH	マスタモード, シリアルデータ入出力を 1 端子使用して通信, シリアルデータ出力端子, SSCK 端子, SCS 端子の機能を設定します。	初期設定ルーチン	H'CE
SSCRL	バス通信モードを設定, ソフトウェアリセット, SSCK, SCS 端子の機能を設定します。	初期設定ルーチン	H'60
SSMR	シリアルデータの転送を MSB ファースト設定, クロック極性・位相・転送レート(ϕ クロック入力)の設定をします。	初期設定ルーチン	H'E0
SSER	割込み(CEI, OEI, RXI, TEI, TXI)を許可し, SSU の送受信許可/禁止を設定します。	初期設定ルーチン	H'8D
SSTDR	受信局 H8/36057 へ送信するデータを設定します。	データ送信	H'AA, H'55
SSRDR	送信局 H8/36057 からのデータを格納します。	データ受信	
SSSR	エラーを検出します。	エラー処理	

(4) 使用 RAM 説明

ラベル名	使用モジュール名	データ長	機能
Rxdata[0 ~ 1]	データ受信	unsigned char	受信したデータを格納します。
T_counter	データ送信	unsigned char	送信データのデータレングスをカウントします。
R_counter	データ受信	unsigned char	受信データのデータレングスをカウントします。

5.3.2 スレーブ側

(1) 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	マスクレベルを設定します。
初期設定	SSU_init	SSU の初期設定を行ないます。
変数初期化	SSU_counter_init	送受信データのデータレングスをカウントします。
エラー処理	SSUI_int	CEI 割込みで起動し、エラー処理を行ないます。 OEI 割込みで起動し、エラー処理を行ないます。 RXI 割込みで起動し、データ受信処理を行ないます。 TEI 割込みで起動し、データ送信を終了します。 TXI 割込みで起動し、データ送信を行ないます。
データ受信 送信終了 データ送信		

(2) 引数の説明

本タスク例では、引数を使用しません。

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを以下に示します。

レジスタ名	機能	使用モジュール名	設定値
SSCRH	スレーブデバイスに設定、シリアルデータに 1 端子使用、シリアルデータの出力端子、SSCK 端子、SCS 端子の機能を設定します。	初期設定ルーチン	H'4D
SSCRL	バス通信モードを設定、ソフトウェアリセット、SSCK、SCS 端子の機能を設定します。	初期設定ルーチン	H'60
SSMR	シリアルデータの転送を MSB ファーストに設定、クロック極性・位相・転送レート(φクロック入力)を設定します。	初期設定ルーチン	H'E0
SSER	割込み(CEI, OEI, RXI)を許可し、SSU の受信許可/禁止を設定します。	初期設定ルーチン	H'43
SSRDR	送信局 H8/36057 からのデータを設定します。	データ受信	
SSTDR	受信局 H8/36057 へ送信するデータを設定します。	データ送信	H'AA, H'55
SSSR	エラーを検出します。	エラー処理	

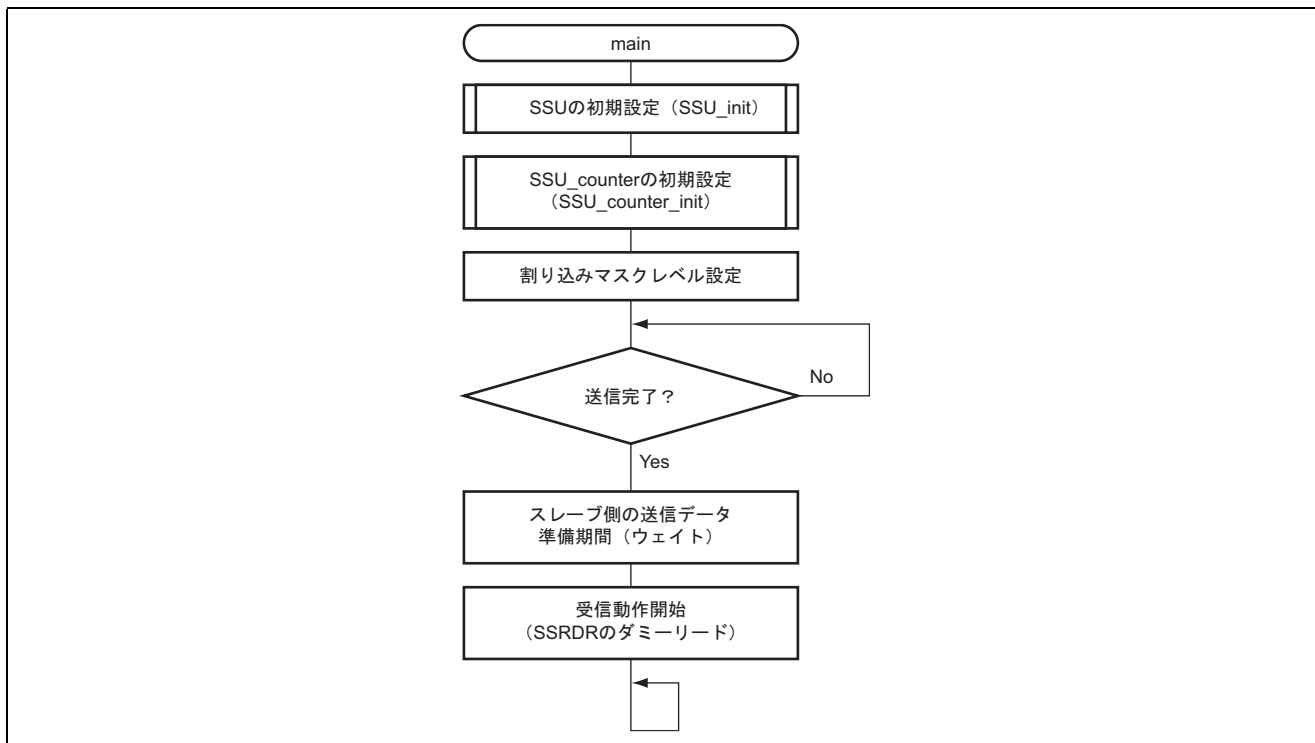
(4) 使用 RAM 説明

ラベル名	使用モジュール名	データ長	機能
Rxdata[0 ~ 1]	データ受信	unsigned char	受信したデータを格納します。
T_counter	データ送信	unsigned char	送信データのデータレングスをカウントします。
R_counter	データ受信	unsigned char	受信データのデータレングスをカウントします。

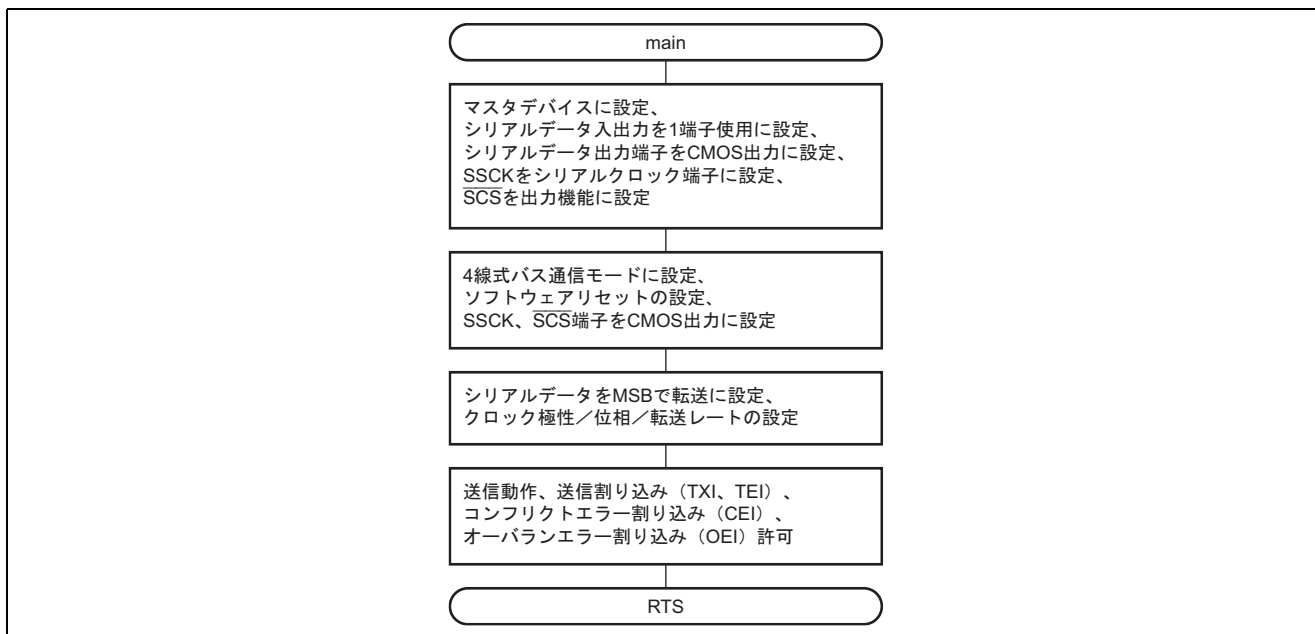
5.4 フローチャート

5.4.1 マスタ側

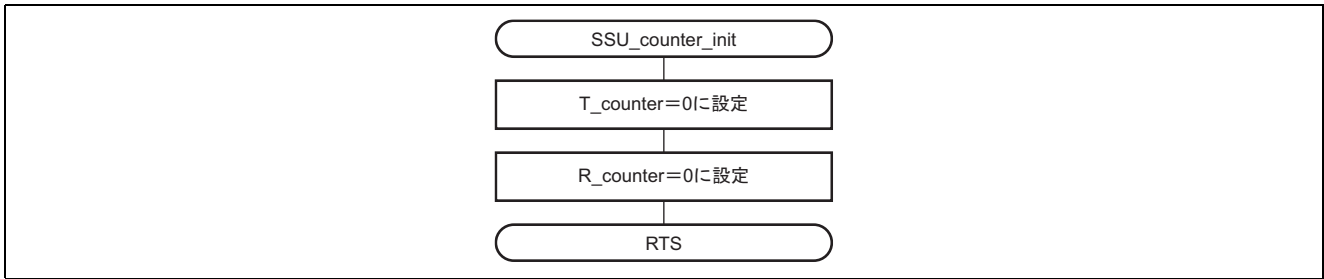
(1) メインルーチン



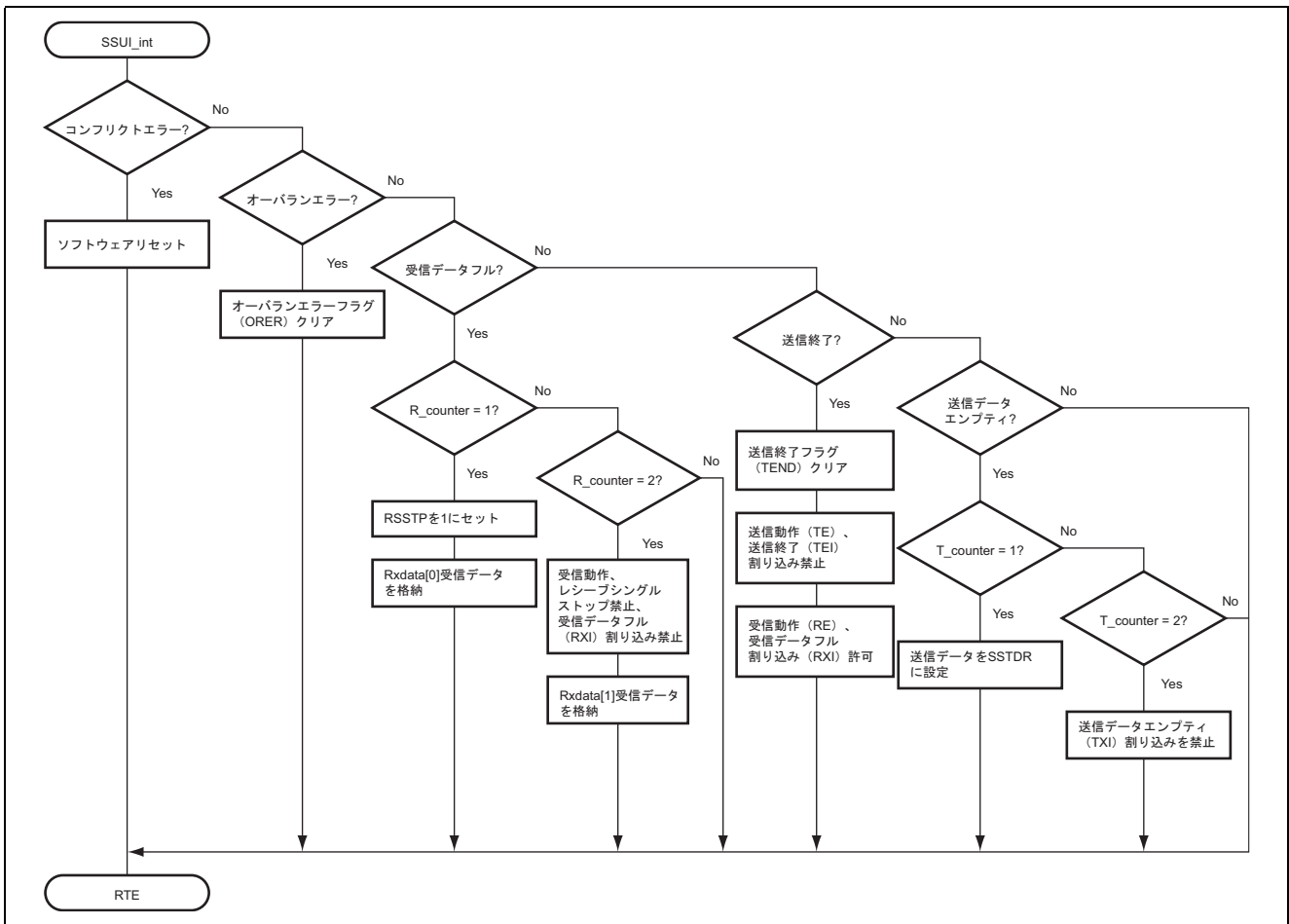
(2) 初期設定



(3) 変数初期化

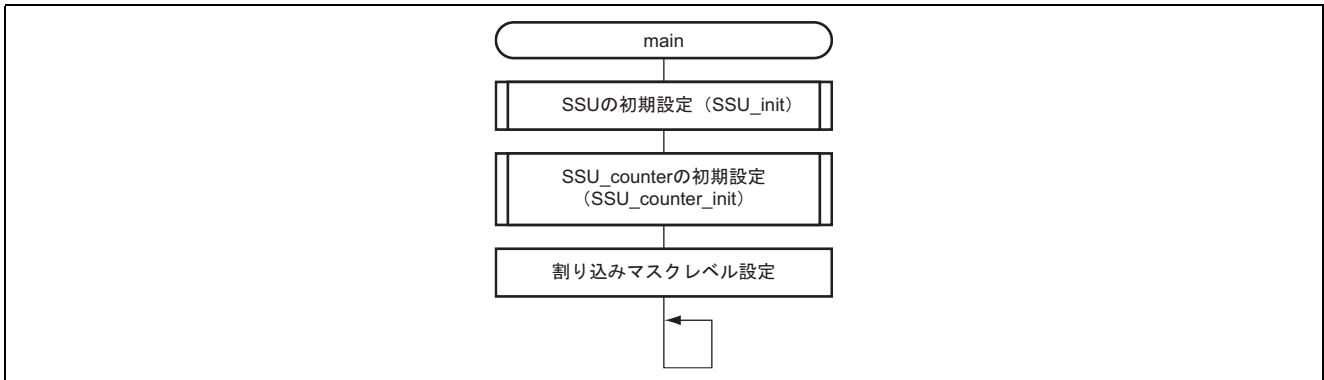


(4) エラー発生，データ受信，データ送信

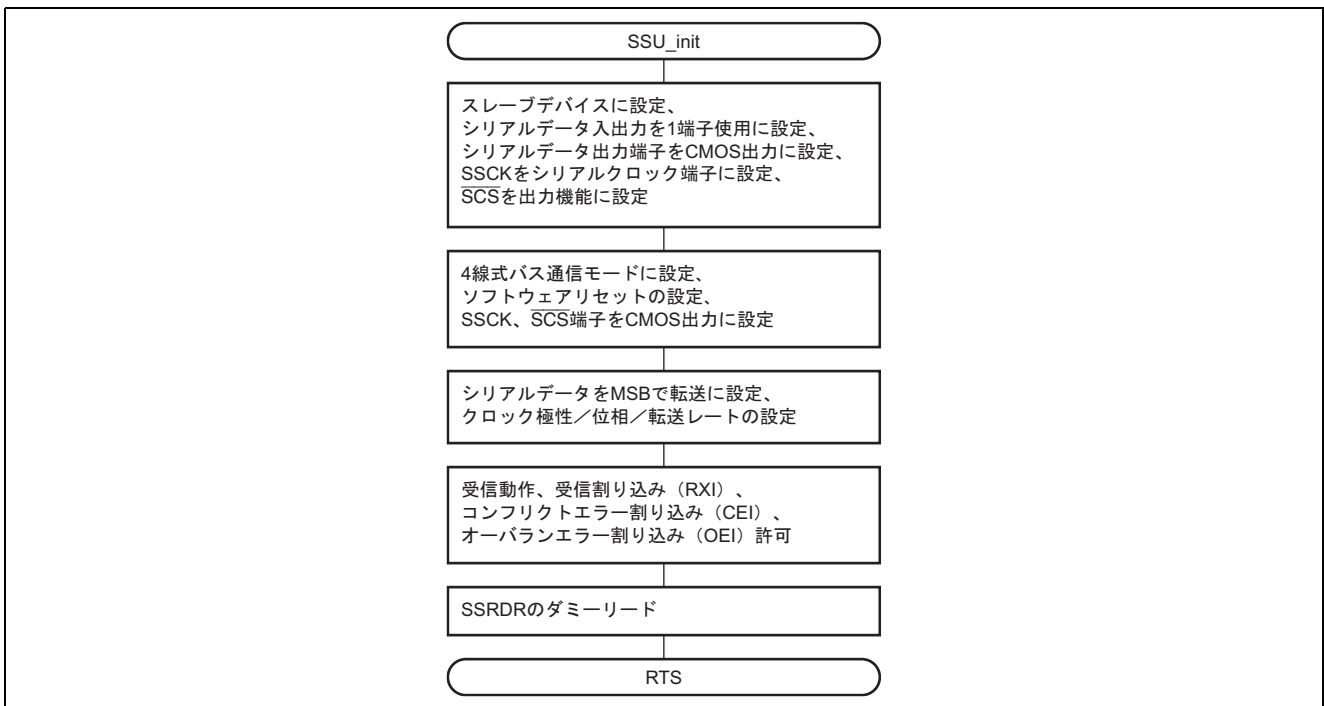


5.4.2 スレーブ側

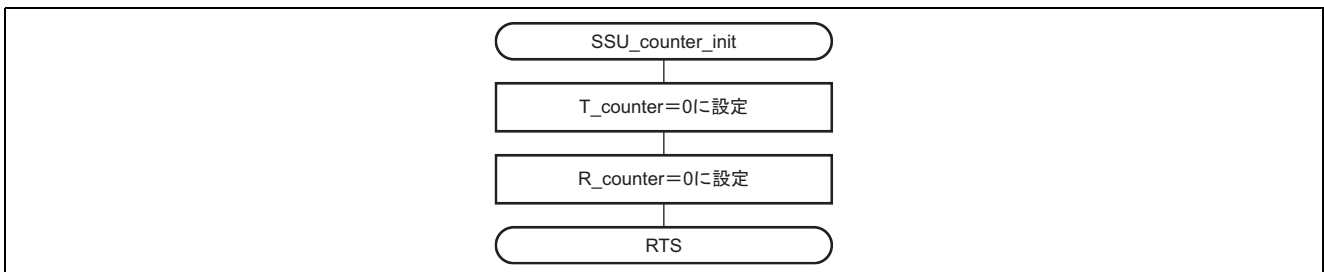
(1) メインルーチン



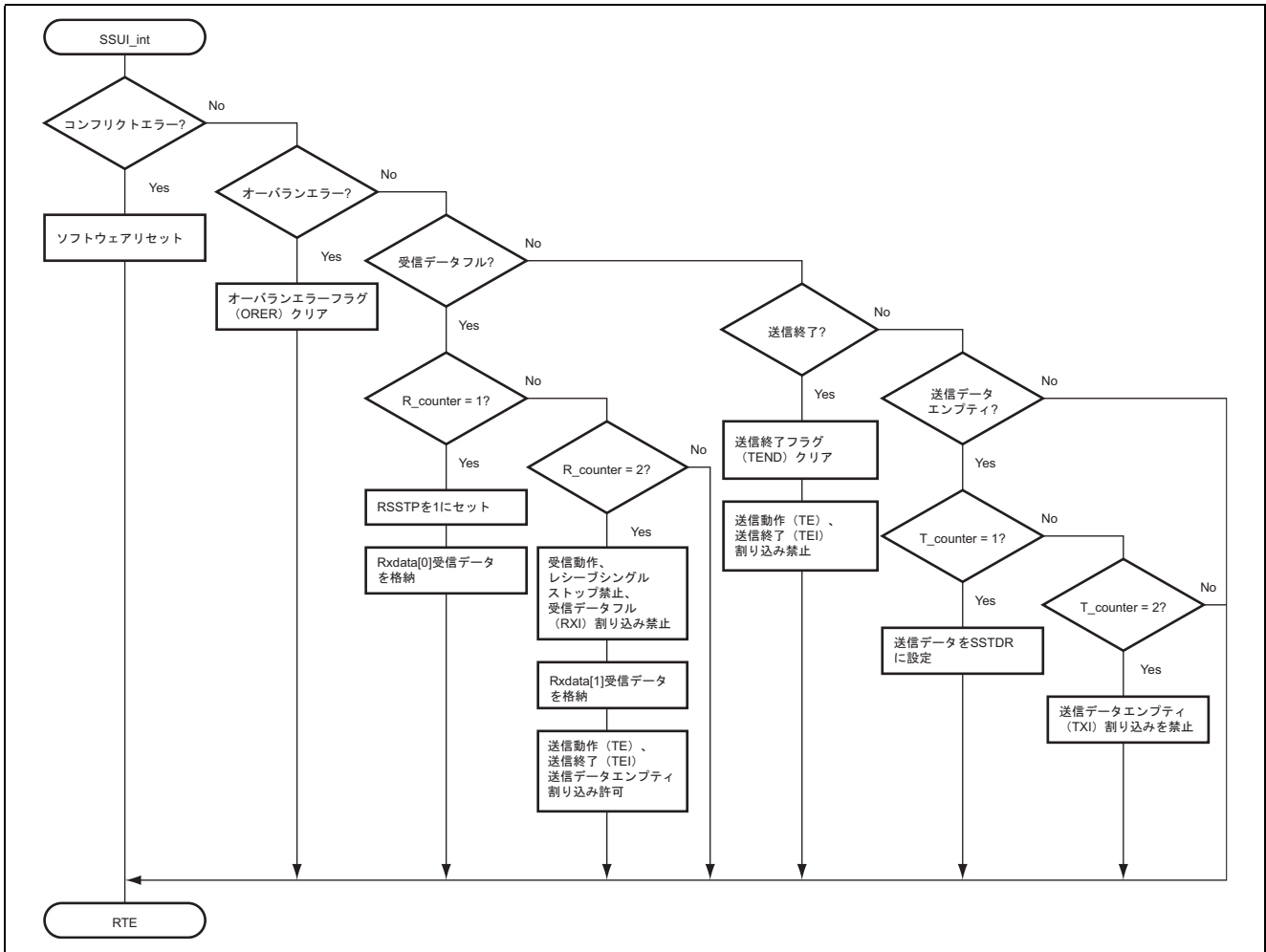
(2) 初期設定



(3) 変数初期化



(4) エラー発生，データ受信，データ送信



5.5 プログラムリスト

5.5.1 マスタ側

```
#include <machine.h>
#include "H8_36057.h"

void main (void);
void SSU_init (void);
void SSU_counter_init(void);

unsigned char T_counter;
unsigned char R_counter;
unsigned char Rxdata[2];
/*****
/* Main Routine */
*****/
void main (void)
#pragma asm
    mov.l #H'FFFFFF7C,SP
#pragma endasm
{
    signed int lp;
    signed int A;
    SSU_init();
    SSU_counter_init();
    set_imask_ccr(0);

    while(SSU.SSER.BIT.TE);
    for(lp=50;lp>0;lp--);
    A = SSU.SSRDR;
    while(1);
}
/*****
/* SSU_counter Initialize Routine */
*****/
void SSU_counter_init(void)
{
    T_counter = 0;
    R_counter = 0;
}
/*****
/* SSU Module Initialize Routine */
*****/
void SSU_init(void)
{
    SSU.SSER.BYTE = 0x00;
    SSU.SSCL.BYTE = 0x00;
    SSU.SSCRH.BYTE = 0xCE;
    SSU.SSCL.BYTE = 0x63;
    SSU.SSMR.BYTE = 0xE0;
    SSU.SSSR.BYTE &= 0x04;
    SSU.SSER.BYTE = 0x8D;
}
```

```

/*****
#pragma interrupt (SSUI_int) /* エラー割込み,受信割込み,送信割込み */
/*****
void SSUI_int(void)
{
    if(SSU.SSSR.BIT.CE && SSU.SSER.BIT.CEIE){
        SSU.SSSR.BIT.CE = 0;
        SSU.SSCRL.BIT.SRES = 1;
    }
    else if(SSU.SSSR.BIT.orer && SSU.SSER.BIT.RIE){
        SSU.SSSR.BIT.orer = 0;
    }
    if(SSU.SSSR.BIT.RDRF && SSU.SSER.BIT.RIE){
        R_counter++;
        if(R_counter == 1){
            SSU.SSER.BIT.RSSTP = 1;
            Rxddata[0] = SSU.SSRDR;
        }
        else if(R_counter == 2){
            SSU.SSER.BYTE &= 0x8D;
            Rxddata[1] = SSU.SSRDR;
        }
    }
    else if(SSU.SSSR.BIT.TEND && SSU.SSER.BIT.TEIE){
        SSU.SSSR.BIT.TEND = 0;
        SSU.SSER.BYTE &= 0x63;
        SSU.SSER.BYTE = 0x43;
    }
    else if(SSU.SSSR.BIT.TDRE && SSU.SSER.BIT.TIE){
        T_counter++;
        if(T_counter == 1){
            SSU.SSTDR = 0xAA;
        }
        else if(T_counter == 2){
            SSU.SSTDR = 0x55;
            SSU.SSER.BIT.TIE = 0;
        }
    }
}

```

5.5.2 スレーブ側

```

#include <machine.h>
#include "H8_36057.h"

void main (void);
void SSU_init (void);
void SSU_counter_init(void);

unsigned char T_counter;
unsigned char R_counter;
unsigned char Rxdata[2];
/*****
/* Main Routine */
*****/
void main (void)
#pragma asm
mov.l #H'FFFFFF7C,SP
#pragma endasm
{
    SSU_init();
    SSU_counter_init();
    set_imask_ccr(0);
    while(1);
}
/*****
/* SSU_counter Initialize Routine */
*****/
void SSU_counter_init(void)
{
    R_counter = 0;
    T_counter = 0;
}
/*****
/* SSU Module Initialize Routine */
*****/
void SSU_init(void)
{
    unsigned char A;
    SSU.SSER.BYTE = 0x00;
    SSU.SSCL.BYTE = 0x00;
    SSU.SSCRH.BYTE = 0x4D;
    SSU.SSCL.BYTE = 0x63;
    SSU.SSMR.BYTE = 0xE0;
    SSU.SSSR.BYTE &= 0x04;
    SSU.SSER.BYTE = 0x43;
    A = SSU.SSRDR;
}
    
```

```

/*****
#pragma interrupt (SSUI_int) /* エラー割込み,受信割込み,送信割込み */
/*****
void SSUI_int(void)
{
    if(SSU.SSSR.BIT.CE && SSU.SSER.BIT.CEIE){
        SSU.SSSR.BIT.CE = 0;
        SSU.SSCRL.BIT.SRES = 1;
    }
    else if(SSU.SSSR.BIT.OPER && SSU.SSER.BIT.RIE){
        SSU.SSSR.BIT.OPER = 0;
    }
    else if(SSU.SSSR.BIT.RDRF && SSU.SSER.BIT.RIE){
        R_counter++;
        if(R_counter == 1){
            Rxdata[0] = SSU.SSRDR;
        }
        else if(R_counter == 2){
            SSU.SSER.BYTE &= 0x8D;
            Rxdata[1] = SSU.SSRDR;
            SSU.SSER.BYTE = 0x8D;
        }
    }
    else if(SSU.SSSR.BIT.TEND && SSU.SSER.BIT.TEIE){
        SSU.SSSR.BIT.TEND = 0;
        SSU.SSER.BYTE &= 0x63;
    }
    else if(SSU.SSSR.BIT.TDRE && SSU.SSER.BIT.TIE){
        T_counter++;
        if(T_counter == 1){
            SSU.SSTDR = 0xAA;
        }
        else if(T_counter == 2){
            SSU.SSTDR = 0x55;
            SSU.SSER.BIT.TIE = 0;
        }
    }
}
}

```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.09.13	—	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。