

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。

H8S/20103, H8S/20203, H8S/20223 グループ

ELC を使用した 3 イベント組み合わせ動作

要旨

H8S/20103, H8S/20203, H8S/20223 グループ内蔵の ELC (イベントリンクコントローラ) を使用して, SCI3_1・受信データフル信号で CPU の介在なくタイマ RD_0・チャンネル 0 をスタート, タイマ RD_0・チャンネル 0 と GRA_0 のコンペアマッチ信号で A/D 変換開始動作を実現します。

対象デバイス

H8S/20103 (R4F20103)

H8S/20203 (R4F20203)

H8S/20223 (R4F20223)

動作確認条件

システムクロック $\phi = \phi_{osc} = 20 \text{ MHz}$

目次

| | |
|-------------------|----|
| 1. 仕様 | 2 |
| 2. 使用機能説明 | 4 |
| 3. 動作原理 | 18 |
| 4. ソフトウェア説明 | 21 |
| 5. フローチャート | 25 |
| 6. プログラムリスト | 34 |

1. 仕様

本アプリケーションにおける仕様を以下に説明します。図 1 に ELC を使用した 3 イベント組み合わせ動作の概要を示します。図 2 に ELC を使用した 3 イベント組み合わせの動作概要を示します。

- (1) PB0/AN0 端子に測定電圧を印加します。
- (2) PB0 端子をアナログ入力 AN0 端子に設定し、A/D 変換終了割り込みを許可します。
- (3) タイマ RD_0・チャンネル 0・コンペアマッチ A 信号をイベント信号として、A/D 変換を開始に設定します。(タイマ RD_0・チャンネル 0 と GRA_0 のコンペアマッチ周期が、A/D 変換のサンプリング周期となります。)
- (4) タイマ RD_0・チャンネル 0 をクロックソース $\phi/32$ 、TRDCNT_0 を GRA_0 とのコンペアマッチでクリアに設定します。
- (5) タイマ RD_0・チャンネル 0 のイベント入力時の動作をカウントスタートに設定します。(タイマ RD_0・チャンネル 0 のイベント入力は、本タスクにおいて SCI3_1・受信データフル信号です。)
- (6) SCI3_1・受信データフル信号をイベント信号として、タイマ RD_0・チャンネル 0 を開始に設定します。
- (7) SCI3_1 を調歩同期式通信、ビットレートを 9600 bps、偶数パリティ、1 ストップビットに設定し、受信、受信割り込みを許可します。
- (8) イベント入力を許可します。
- (9) I ビットをクリアし、割り込みを許可します。
- (10) SCI3_1・受信データフル信号が発生すると、CPU の介在なくタイマ RD_0・チャンネル 0 のカウント動作を開始します。(8 バイトの受信が終了すると、受信を禁止します。)
- (11) タイマ RD_0・チャンネル 0 と GRA_0 のコンペアマッチ信号が発生するごとに、CPU の介在なく AN0 端子の A/D 変換を開始します。
- (12) A/D 変換結果は 10 回サンプリングし、最大値と最小値を除いた 8 回の平均値を有効とします。

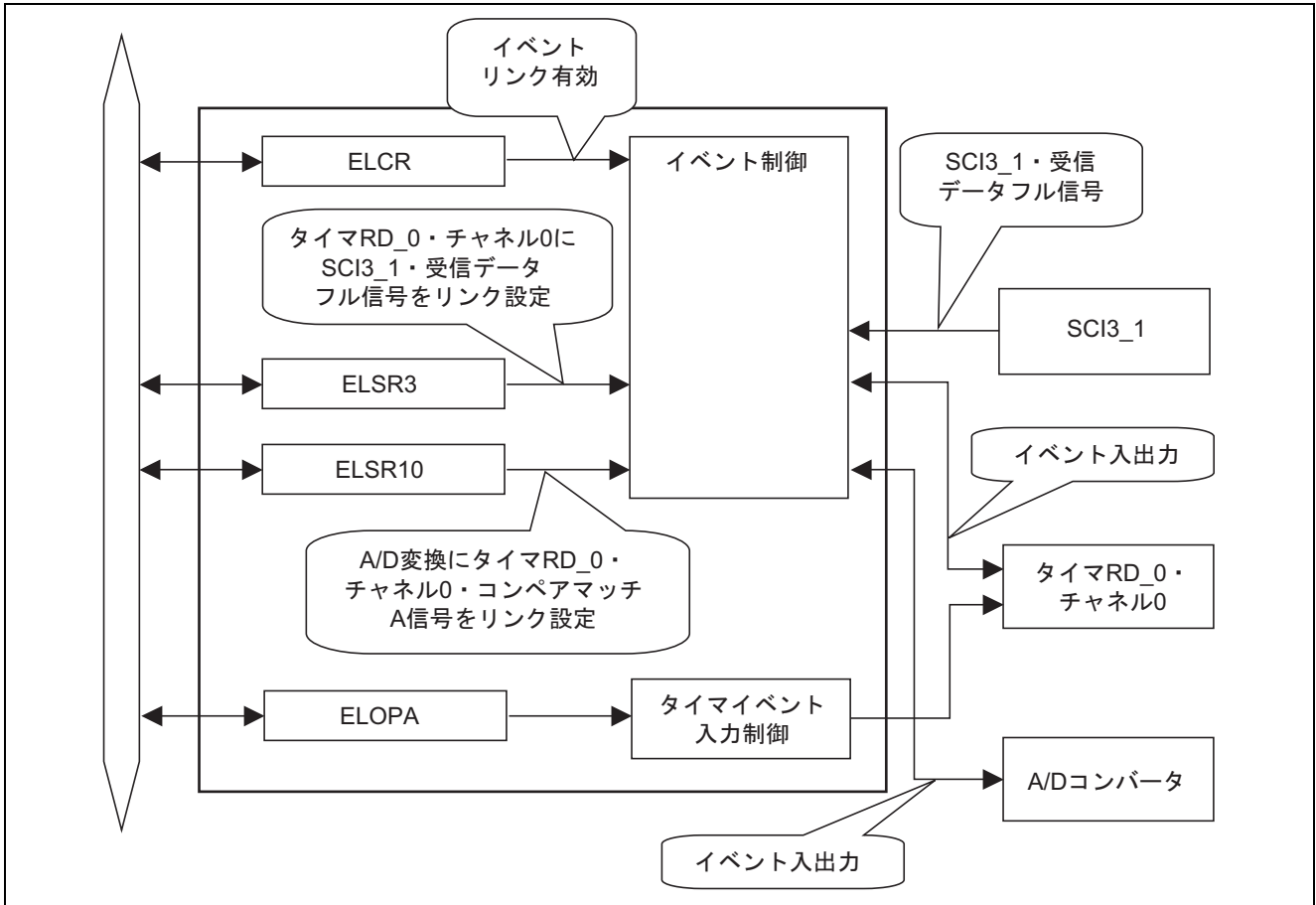


図 1 ELC を使用した 3 イベント組み合わせ動作の概要

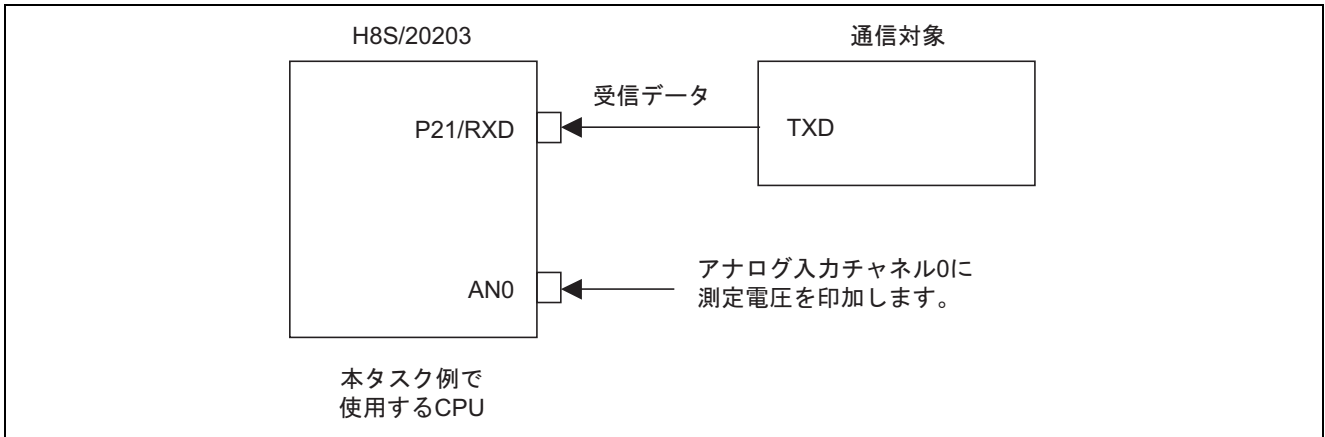


図 2 ELC を使用した 3 イベント組み合わせの動作概要

2. 使用機能説明

2.1 ELC (イベントリンクコントローラ)

以下に、ELC の機能を説明します。イベントリンクコントローラのブロック図を図 3 に示します。

ELC は、各周辺モジュールを出力するイベントをモジュール間で相互に接続 (リンク) します。イベントリンクにより、CPU を介さず直接モジュール間での連携動作が可能になります。

- 59 種類のイベント信号を直接モジュールへリンク可能
- タイマ系のモジュールは、イベント入力時の動作の選択が可能
- ポート 3, ポート 6 へのイベントリンクが可能
- ポートの設定により、ポートでのイベント発生が可能
- イベントを接続するポートは、1 ビット単位または任意の複数ビットをグループ化して指定可能
- イベント発生タイマにより、4 チャンネルの任意設定周期のイベント発生が可能

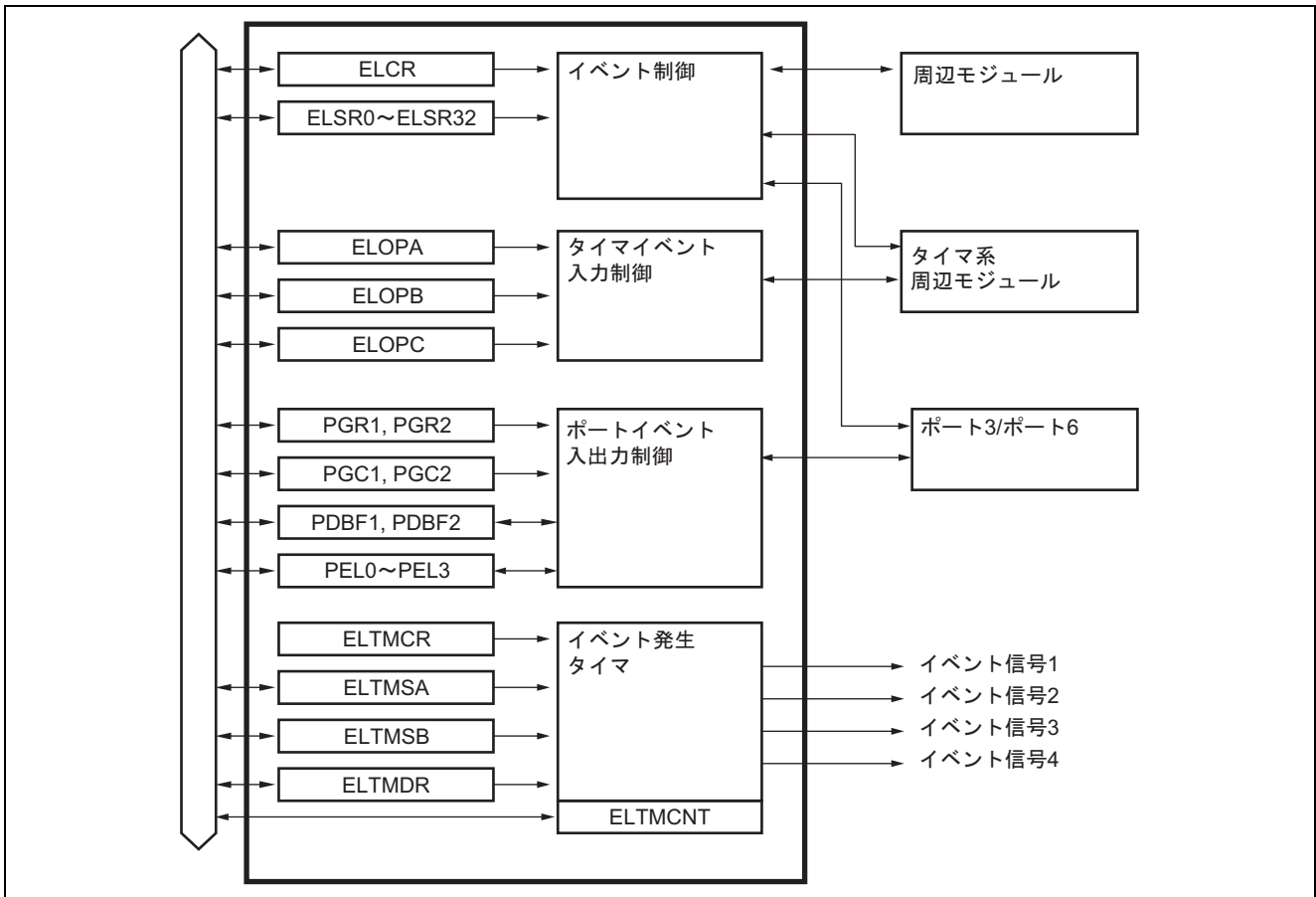


図 3 イベントリンクコントローラのブロック図

2.1.1 タイマ系周辺機能のイベント入力時の動作

ELOP によりイベント入力時の動作を設定します。

(1) カウントスタート動作

イベント入力により、タイマのカウントをスタートし、各タイマの制御レジスタのカウントスタートビット*が、"1"にセットされます。カウントスタートビットが"1"の状態を入力されたイベントは、無効です。

(2) イベントカウンタ動作

タイマクロックソースとして、イベント入力を選択されタイマが動作します。

(3) インプットキャプチャ動作

イベント入力により、キャプチャ動作します。

【注】 *: 各タイマのビット説明を参照してください。

2.1.2 A/D コンバータ, D/A コンバータのイベント入力時の動作

A/D 制御レジスタのスタートビット, D/A 制御レジスタのアウトプットイネーブルビット*が"1"にセットされ, A/D 変換または D/A 変換がスタートします。

【注】 *: A/D コンバータ, D/A コンバータのビット説明を参照してください。

2.2 シリアルコミュニケーションインタフェース 3 (SCI3)

本 LSI は、独立した 3 チャンネルのシリアルコミュニケーションインタフェース 3 (SCI3) を備えています。SCI3 は、調歩同期式とクロック同期式の 2 方式のシリアルデータ通信が可能です。調歩同期方式は Universal Asynchronous Receiver/Transmitter (UART) や、Asynchronous Communication Interface Adapter (ACIA) などの標準の調歩同期式通信用 LSI とのシリアルデータ通信ができます。また、調歩同期方式では複数のプロセッサ間のシリアルデータ通信機能 (マルチプロセッサ通信機能) を備えています。

SCI3_2 は IrDA (Infrared Data Association) 規格バージョン 1.0 に基づく IrDA 通信波形の送受信が可能です。

SCI3 のチャンネル構成を表 1 に、ブロック図を図 4、図 5 に示します。3 チャンネル (SCI3, SCI3_2, SCI3_3) とも同一機能です。なお、中ではチャンネルによる区別を省略します。

- シリアルデータ通信フォーマットを調歩同期式またはクロック同期式に設定可能
- 全二重通信が可能
独立した送信部と受信部を備えているので、送信と受信を同時に行うことができます。また、送信部および受信部ともにダブルバッファ構造になっていますので連続送受信が可能です。
- 内蔵ボーレートジェネレータで任意のビットレートを選択可能
- 送受信クロックソースとして内蔵ボーレートジェネレータまたは外部クロックを選択可能
- 6 種類の割り込み要因
送信終了、送信データエンプティ、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの割り込み要因があります。また、送信データエンプティ、受信データフル割り込み要因により、DTC を起動することができます。
- 内部 16 ビットバスによる高速アクセス
TRDCNT, GR の 16 ビットレジスタに対して、16 ビットバスインタフェースによる高速アクセスが可能

調歩同期式モード

- データ長: 7 ビット/8 ビット選択可能
- ストップビット長: 1 ビット/2 ビット選択可能
- パリティ: 偶数パリティ/奇数パリティ/パリティなしから選択可能
- 受信エラーの検出: パリティエラー、オーバランエラー、フレーミングエラー
- ブレークの検出: フレーミングエラー発生時の RXD 端子のレベルを直接読み出すことでブレークを検出可能

クロック同期式モード

- データ長: 8 ビット
- 受信エラーの検出: オーバランエラー

表 1 SCI3 のチャンネル構成

| チャンネル | 略称 | 端子 | レジスタ | レジスタアドレス | ノイズ除去回路 |
|---------|----------------------|--------------------------------------|--------|----------|---------|
| チャンネル 1 | SCI3* ¹ | SCK3 RXD TXD | SMR | H'FF0550 | あり |
| | | | BRR | H'FF0551 | |
| | | | SCR3 | H'FF0552 | |
| | | | TDR | H'FF0553 | |
| | | | SSR | H'FF0554 | |
| | | | RDR | H'FF0555 | |
| | | | RSR | — | |
| | | | TSR | — | |
| | | | SPMR | H'FF0556 | |
| チャンネル 2 | SCI3_2* ² | SCK3_2 RXD_2/IrRxD TXD_2/IrTxD | SMR_2 | H'FF0558 | あり |
| | | | BRR_2 | H'FF0559 | |
| | | | SCR3_2 | H'FF055A | |
| | | | TDR_2 | H'FF055B | |
| | | | SSR_2 | H'FF055C | |
| | | | RDR_2 | H'FF055D | |
| | | | RSR_2 | — | |
| | | | TSR_2 | — | |
| | | | SPMR_2 | H'FF055E | |
| | | | IrCR | H'FF05DE | |
| チャンネル 3 | SCI3_3 | SCK3_3 RXD_3 TXD_3 | SMR_3 | H'FF0560 | あり |
| | | | BRR_3 | H'FF0561 | |
| | | | SCR3_3 | H'FF0562 | |
| | | | TDR_3 | H'FF0563 | |
| | | | SSR_3 | H'FF0564 | |
| | | | RDR_3 | H'FF0565 | |
| | | | RSR_3 | — | |
| | | | TSR_3 | — | |
| | | | SPMR_3 | H'FF0566 | |

- 【注】 1. ブートモードによるオンボードプログラミングモードでは，SCI3 のチャンネル 1 を使用します。
2. SCI3_2 は，IrDA (Infrared Data Association) 規格バージョン 1.0 に基づく IrDA 通信波形の送受信が可能です。

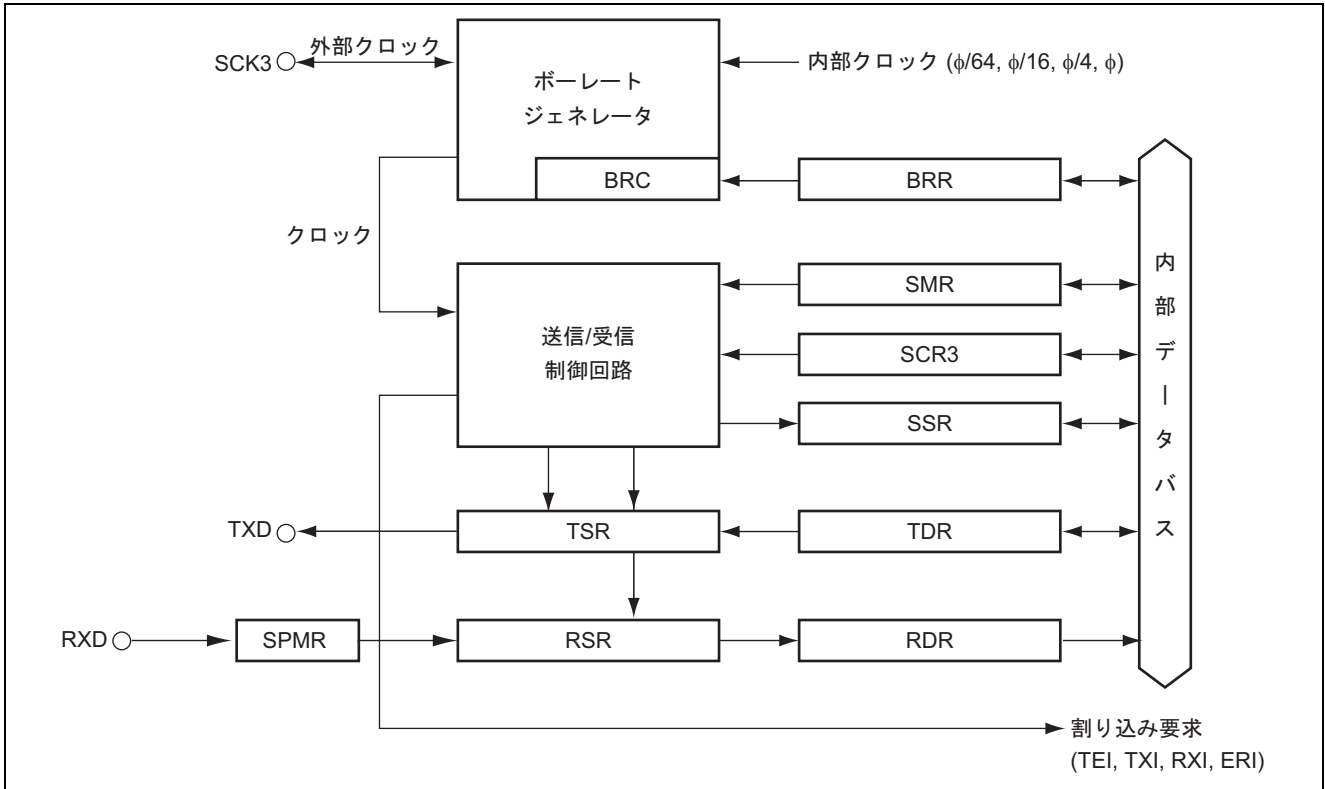


図 4 SCI3, SCI3_3 のブロック図

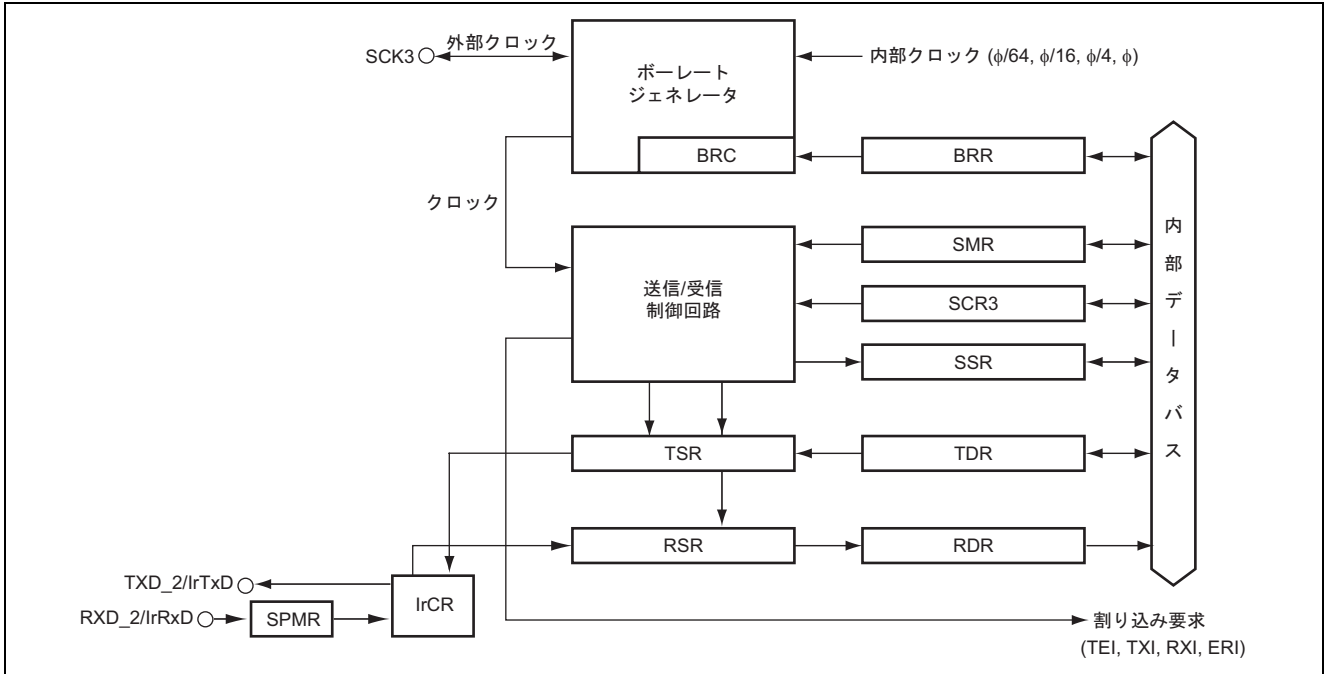


図 5 SCI3_2 のブロック図

2.3 タイマ RD

以下に、タイマ RD の機能を説明します。

タイマ RD は 2 チャンネルの 16 ビットタイマを 2 ユニット (タイマ RD_0, タイマ RD_1) を内蔵しています。タイマ RD の機能を表 2 に、タイマ RD のチャンネル構成を表 3 に示します。また、タイマ RD のブロック図 (全体図) を図 6 に、チャンネル 0 のブロック図を図 7 に、チャンネル 1 のブロック図を図 8 に示します。タイマ RD_0 とタイマ RD_1 は同一です。なお、本文中ではタイマ RD_0, タイマ RD_1 の区別を省略します。(H8S/20103 グループはタイマ RD ユニット 1 を搭載していません。)

- 最大 8 種類の入出力処理が可能
- 各チャンネルに 4 本, 合計 8 本のジェネラルレジスタ (GR) を持ち, 各レジスタは独立にアウトプットコンペア/インプットキャプチャの機能設定が可能
- カウンタ入力クロック: 7 種類
6 種類の内部クロック (ϕ , $\phi/2$, $\phi/4$, $\phi/8$, $\phi/32$, $\phi/40$ (高速音チップオシレータ 40 MHz/32 MHz クロック)) と外部クロックの内から選択可能
- 各チャンネルとも次の動作モードを設定可能
 - タイマモード: アウトプットコンペア機能 (0 出力/1 出力/トグル出力が可能)
 - インプットキャプチャ機能 (立ち上がりエッジ/立ち下がりエッジ/両エッジを検出)
 - 同期動作: タイマカウンタ_0, 1 (TRDCNT_0, TRDCNT_1) への同時書き込みが可能
 - コンペアマッチ/インプットキャプチャによる同時クリアが可能
 - PWM モード: 任意のデューティの PWM 出力が可能
 - 最大 6 相の PWM 出力が可能
 - PWM3 モード: 正相・逆相がノンオーバーラップの関係にある PWM 波形を 1 相出力可能
 - リセット同期 PWM モード: 正相・逆相の PWM 波形を 3 相出力可能
 - 相補 PWM モード: 正相・逆相がノンオーバーラップの関係にある PWM 波形を 3 相出力可能
 - PWM 周期による A/D 変換スタートトリガを設定可能
 - バッファ動作: インプットキャプチャレジスタのダブルバッファ構成が可能
 - アウトプットコンペアレジスタの自動書き換えが可能
- 内部 16 ビットバスによる高速アクセス
TRDCNT, GR の 16 ビットレジスタに対して, 16 ビットバスインタフェースによる高速アクセスが可能
- タイマ出力初期値を任意に設定可能
- 外部トリガによるタイマ出力禁止機能
- 割り込み要因: 11 種類
各チャンネルともコンペアマッチ/インプットキャプチャ兼用割り込み $\times 4$ 要因, オーバフロー割り込みが要求可能。またチャンネル 1 はアンダフロー割り込み設定が可能

表 2 タイマ RD の機能一覧 (1 ユニット)

| 項目 | | チャンネル 0 | チャンネル 1 |
|--|-------|--|---|
| カウントクロック | | 内部クロック: ϕ , $\phi/2$, $\phi/4$, $\phi/8$, $\phi/32$, $\phi/40$ 外部クロック: FTIOA0 (TCLK) | |
| ジェネラルレジスタ (アウトプットコンペア/インプットキャプチャ兼用レジスタ) | | GRA_0, GRB_0, GRC_0, GRD_0 | GRA_1, GRB_1, GRC_1, GRD_1 |
| バッファレジスタ | | GRC_0, GRD_0 | GRC_1, GRD_1 |
| 入出力端子 | | FTIOA0, FTIOB0, FTIOC0, FTIOD0 | FTIOA1, FTIOB1, FTIOC1, FTIOD1 |
| カウンタクリア機能 | | GRA_0/GRB_0/GRC_0/GRD_0 の コンペアマッチまたはインプット キャプチャ | GRA_1/GRB_1/GRC_1/GRD_1 の コンペアマッチまたはインプット キャプチャ |
| コンペアマッチ出力 | 0 出力 | | |
| | 1 出力 | | |
| | トグル出力 | | |
| インプットキャプチャ機能 | | | |
| 同期動作 | | | |
| PWM モード | | | |
| PWM3 モード | | | |
| リセット同期 PWM モード | | | |
| 相補 PWM モード | | | |
| バッファ動作 | | | |
| 割り込み要因 | | コンペアマッチ/ インプットキャプチャ A0 ~ D0 オーバフロー | コンペアマッチ/ インプットキャプチャ A1 ~ D1 オーバフロー アンダフロー |

表 3 タイマ RD のチャンネル構成

| 略称 | チャンネル | 端子 | |
|----------------------|----------------------|---------|---------|
| タイマ RD_0 (ユニット 0) | 0 | FTIOA0 | |
| | | FTIOB0 | |
| | | FTIOC0 | |
| | | FTIOD0 | |
| | 1 | FTIOA1 | |
| | | FTIOB1 | |
| | | FTIOC1 | |
| | | FTIOD1 | |
| | チャンネル 0, チャンネル 1 共通 | | TRDOI_0 |
| | タイマ RD_1 (ユニット 1) | 2 | FTIOA2 |
| | | | FTIOB2 |
| | | | FTIOC2 |
| FTIOD2 | | | |
| 3 | | FTIOA3 | |
| | | FTIOB3 | |
| | | FTIOC3 | |
| | | FTIOD3 | |
| チャンネル 2, チャンネル 3 共通 | | TRDOI_1 | |

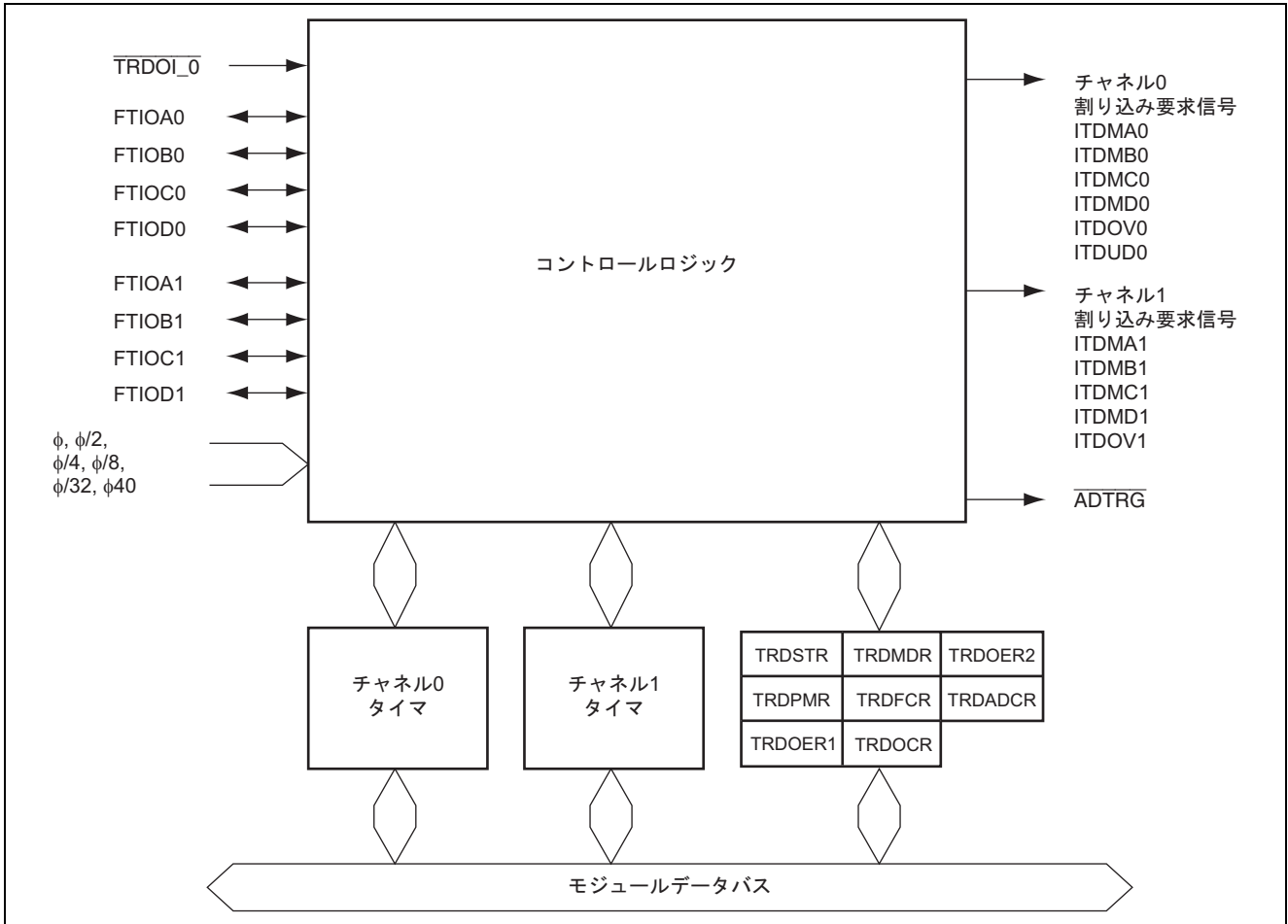


図6 タイマ RD (1 ユニット) のブロック図

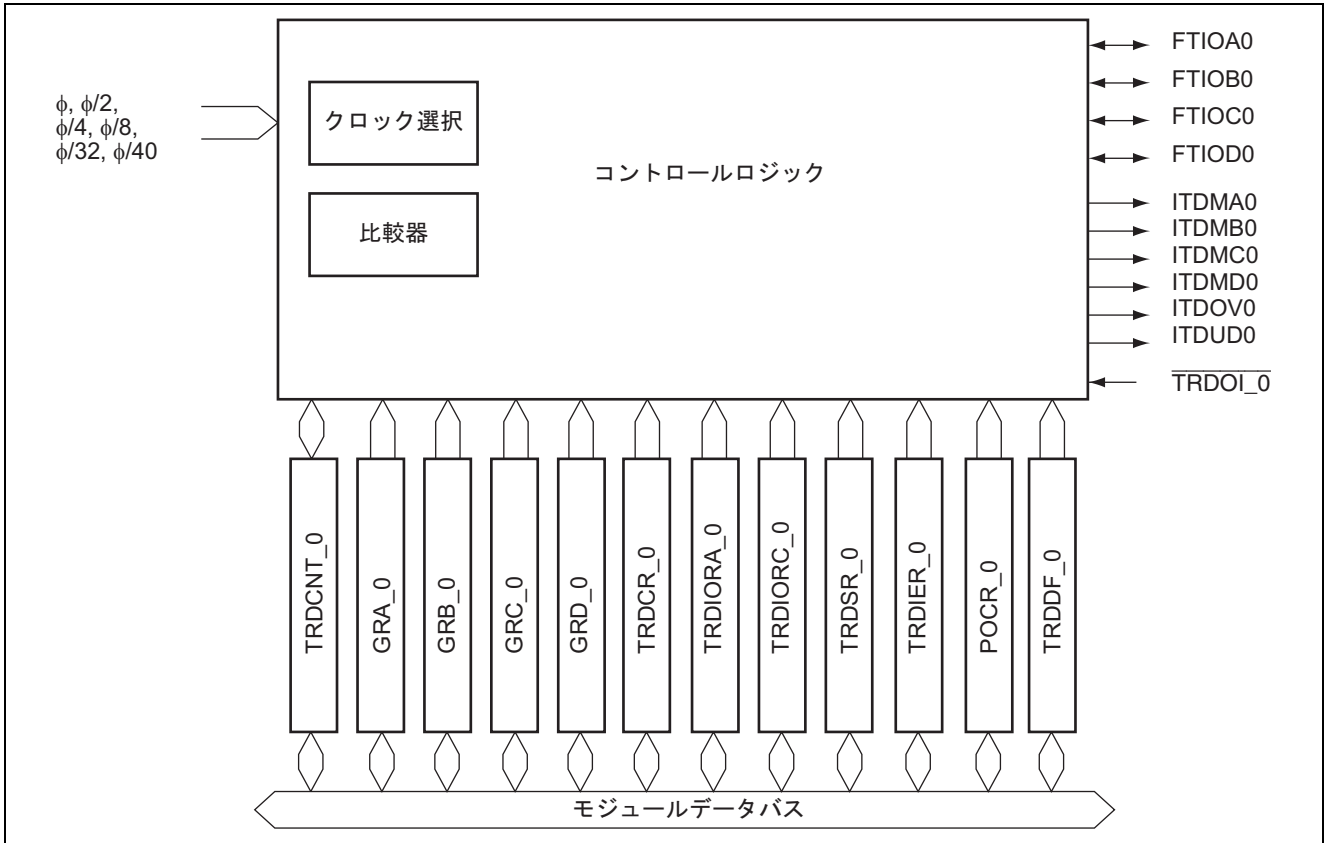


図7 タイマRD (チャンネル0) のブロック図

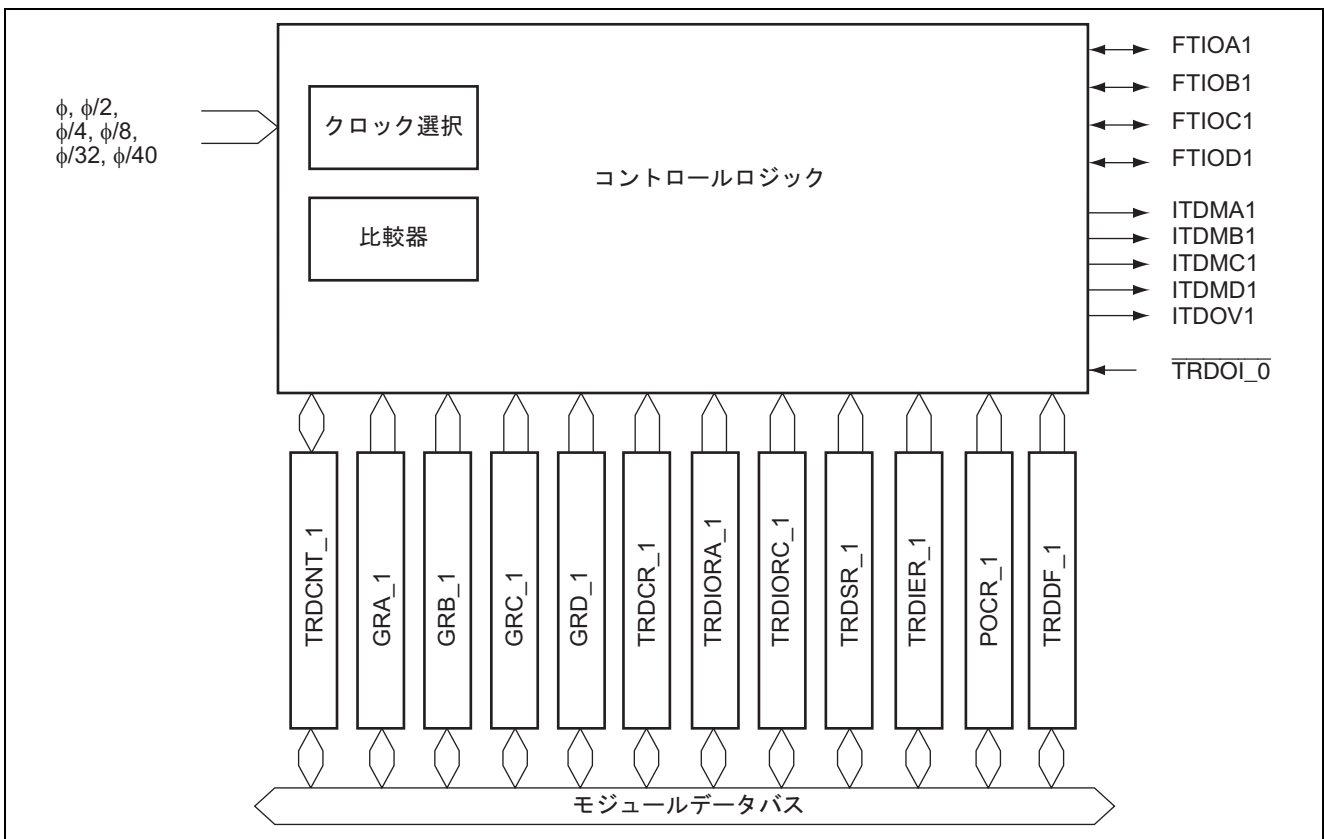


図8 タイマRD (チャンネル1) のブロック図

2.3.1 イベントリンクによる動作

タイマ RD ユニット 0 はイベントリンクコントローラ (ELC) の設定により、他モジュールで発生したイベントによる次の動作が可能です。チャンネル 0、チャンネル 1 はそれぞれ独立に設定可能です。

(1) カウントスタート動作

ELC の ELOPA, ELOPB でタイマ RD のカウントスタート動作を選択します。ELSR3, ELSR4 で指定したイベントが発生すると、TRDSTR の STR[1:0] ビットが"1"にセットされ、タイマ RD のカウントがスタートします。ただし、STR ビットが"1"にセットされた状態で指定したイベントが発生した場合は、そのイベントは無効となります。

(2) イベントカウンタ動作

ELC の ELOPA, ELOPB でタイマ RD のイベントカウンタ動作を選択します。ELSR3, ELSR4 で指定したイベントが発生すると、TRDCR1 の TPSC[2:0] ビットの設定に関係なくそのイベントをカウントソースとして、イベントカウンタ動作を行います。カウント値をリードすると、実際に入力されたイベント数が読み出されます。

(3) インプットキャプチャ動作

ELC の ELOPA, ELOPB でタイマ RD のインプットキャプチャ動作を選択します。ELSR3, ELSR4 で指定したイベントが発生すると、TRDCNT の値を GRD へキャプチャします。イベントリンクによるインプットキャプチャ動作を使用する場合は、タイマ RD の TRDIORC を IOD[3:0] = B'1101 に設定し、TRDSTR の STR ビットを"1"にセットしカウンタをスタートさせてください。このとき、同時に FTIOD 端子の入力も有効になるため、FTIOD 端子の入力を固定するか、PMC にて FTIOD 端子をポートに割り付けない等の対策をしてください。

2.4 A/D コンバータ

以下に、A/D コンバータの機能を説明します。

本 LSI は、逐次比較方式の 10 ビットの A/D コンバータ (1 ユニットまたは 2 ユニット) を内蔵しており、最大合計 16 チャンネルのアナログ入力を選択することができます。A/D コンバータのブロック図を図 9、図 10 に示します。

ユニット 1 とユニット 2 の相違点は、アナログ入力チャンネル数およびデータレジスタの本数のみで、その他の機能は同一です。

- 分解能: 10 ビット
- 入力チャンネル
 ユニット 1: 12 チャンネル (H8S/20223, H8S/20203 グループ), 8 チャンネル (H8S/20103 グループ)
 ユニット 2: 4 チャンネル (H8S/20223 グループ)
- 変換時間: 1 チャンネル当たり 2 μ s (20 MHz 動作時)
- 動作モード: 2 種類
 A/D 変換モード: 選択されたアナログ入力を A/D 変換
 コンペアモード: 選択されたアナログ入力をユーザが設定した電圧と比較
- チャンネル選択モード
 シングルモード: 1 チャンネルの A/D 変換またはコンペア
 スキャンモード: 1~4 チャンネルの連続 A/D 変換または 1~8 チャンネルの連続 A/D 変換
- データレジスタ: 8 本 (ユニット 1), 4 本 (ユニット 2)
 A/D 変換結果は各チャンネルに対応した 16 ビットデータレジスタに保持
- サンプル&ホールド機能付き
- A/D 変換開始方法: 3 種類
 ソフトウェア
 16 ビットタイマ (タイマ RC またはタイマ RD) による変換開始トリガ
 外部トリガ信号
- 割り込み要因
 A/D 変換終了割り込み要求 (ADI) を発生
- コンペア結果が変化した場合、割り込み要求 (CMP1) を発生
- モジュールスタンバイ機能の設定可能

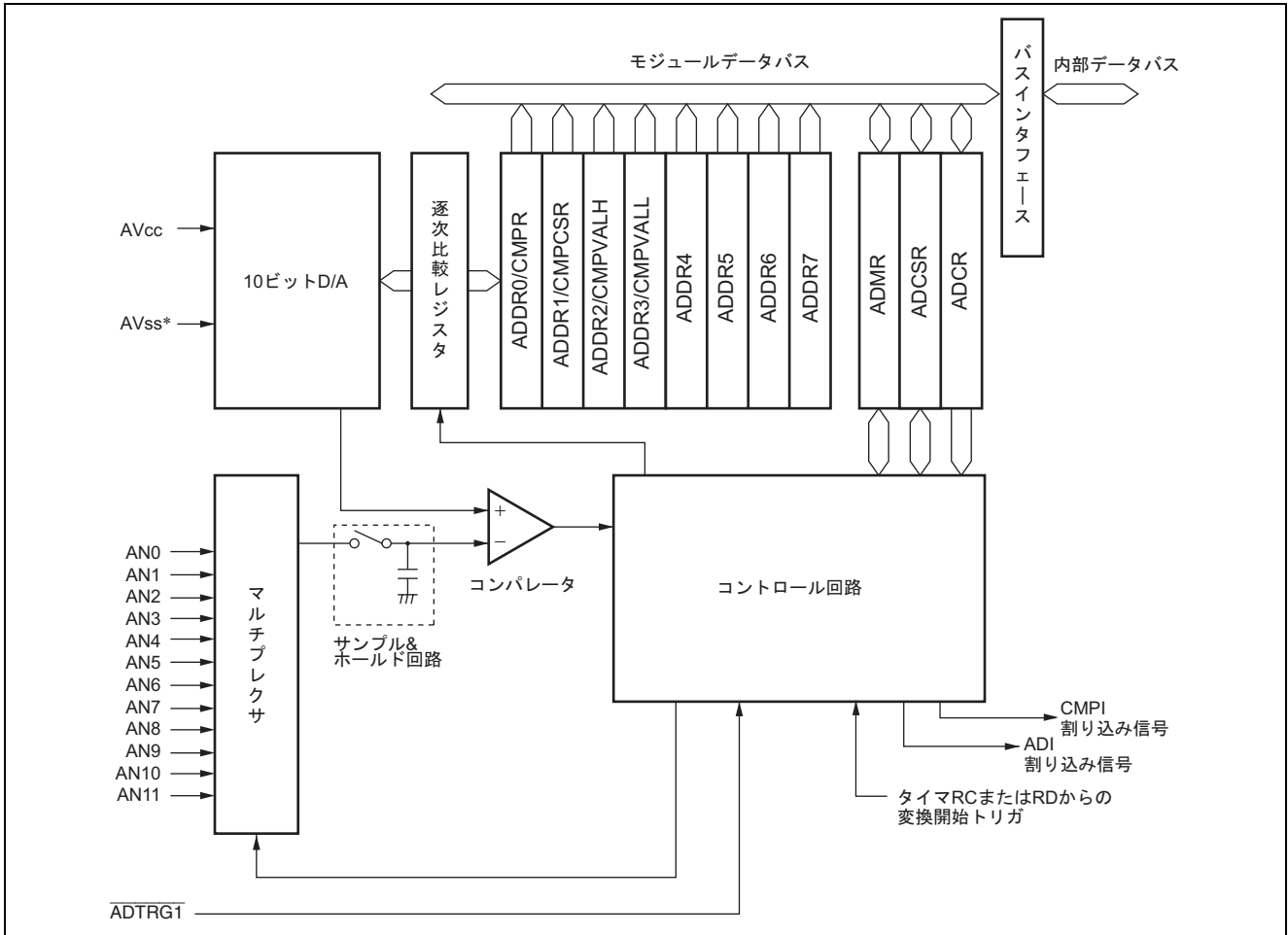


図9 A/Dコンバータ (ユニット1) のブロック図

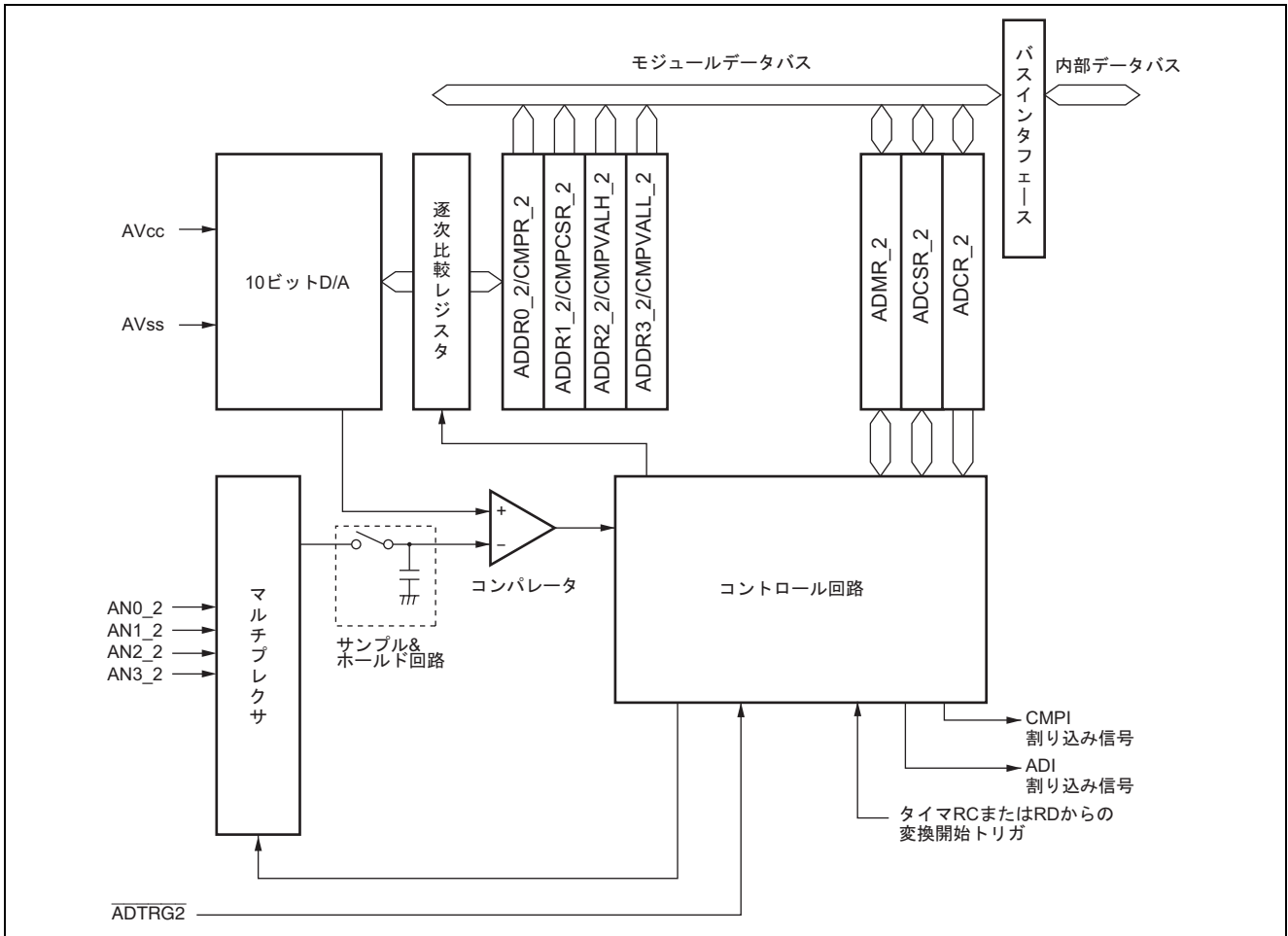


図 10 A/D コンバータ (ユニット 2) のブロック図

3. 動作原理

本タスク例の動作原理を図 11 に示します。図 11、図 12 に示すようなハードウェア処理,およびソフトウェア処理によって,ELC を使用した 3 イベント組み合わせ動作をします。本タスク例の A/D コンバータの動作を図 13 に示します。

RDRF が "1" となり,受信データフル信号が発生すると,CPU の介在なくタイマ RD_0・チャンネル 0 がスタートします。タイマ RD_0・チャンネル 0 と GRA_0 のコンペアマッチが発生すると,CPU の介在なく A/D 変換を開始します。

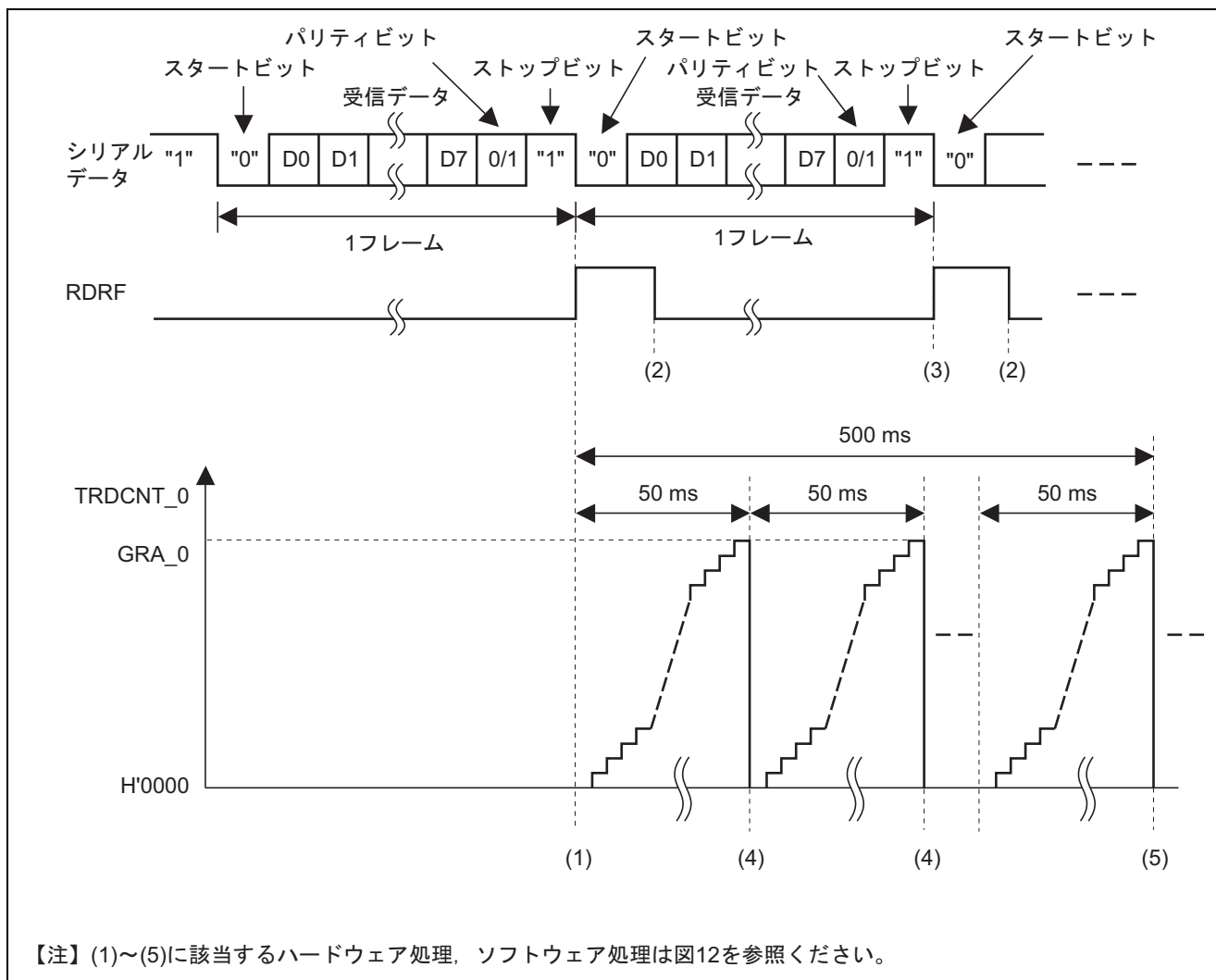


図 11 本タスク例の動作原理

| | | |
|---|---|---|
| <p>初期設定</p> <p>ソフトウェア処理</p> <p>初期設定 (a) A/Dコンバータの設定 (b) タイマRD_0・チャンネル0設定 (c) SCI3の設定 (d) イベントリンク設定 (e) 割り込み許可</p> | <p>(1)</p> <p>ハードウェア処理</p> <p>(a) RSRからRDRにデータ転送 (b) RDRFを"1"にセット (c) タイマRD_0・チャンネル0スタート (d) SCI3_1受信割り込み発生</p> | <p>(2)</p> <p>ハードウェア処理</p> <p>(a) RDRFを"0"にクリア</p> <p>ソフトウェア処理</p> <p>SCI3_1受信割り込み (a) RDRから受信データバッファへ受信データを格納</p> |
| <p>(3)</p> <p>ハードウェア処理</p> <p>(a) RSRからRDRにデータ転送 (b) RDRFを"1"にセット (c) SCI3_1受信割り込み発生</p> | <p>(4)</p> <p>ハードウェア処理</p> <p>(a) TRDCNT_0とGRA_0のコンペアマッチでTRDCNT_0をH'0000にクリア (b) A/D変換スタート (c) A/D変換終了 (d) A/D変換結果をADDR0に格納</p> <p>ソフトウェア処理</p> <p>A/D変換終了割り込み (a) A/D変換結果ADDR0をRAMに格納 (b) 必要であれば, A/D変換データの最大値, 最小値の更新</p> | <p>(5)</p> <p>ハードウェア処理</p> <p>(a) TRDCNT_0とGRA_0のコンペアマッチでTRDCNT_0をH'0000にクリア (b) A/D変換スタート (c) A/D変換終了 (d) A/D変換結果をADDR0に格納</p> <p>ソフトウェア処理</p> <p>A/D変換終了割り込み (a) A/D変換結果ADDR0をRAMに格納 (b) 最大値, 最小値を除いた8回の平均値を求める</p> |

図 12 本タスク例のハードウェア処理, ソフトウェア処理

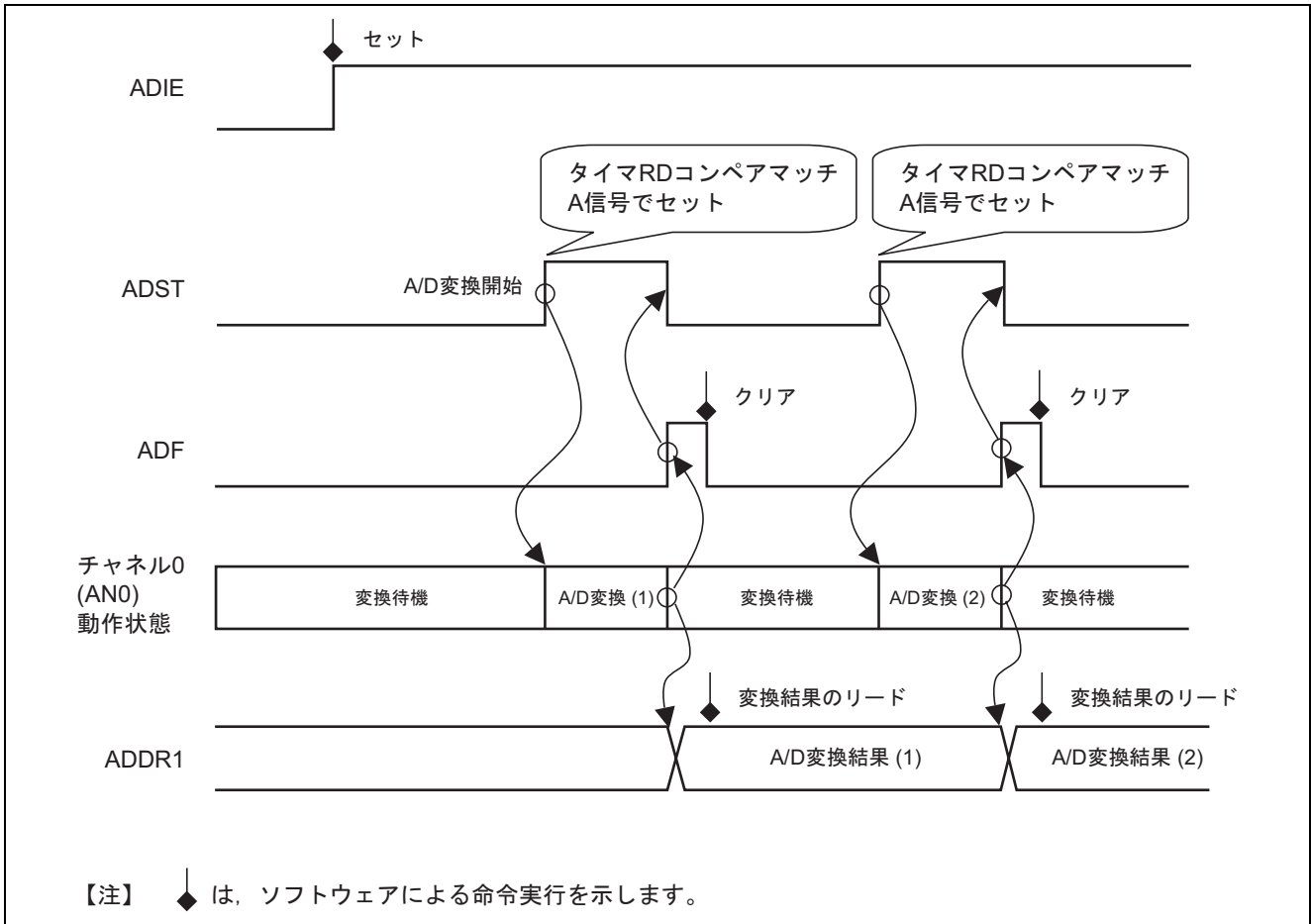


図 13 本タスク例の A/D コンバータの動作

4. ソフトウェア説明

4.1 モジュール説明

表 4 に本タスク例におけるモジュール説明を示します。

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 | 各関数の呼び出し, 割り込み許可 |
| システムの初期化ルーチン | h8s_sysinit | モジュールスタンバイの設定, WDT の停止, システムクロックの設定, バスマスタ動作クロックの設定 |
| A/D コンバータ設定ルーチン | init_ad | A/D コンバータの設定, ELC の設定 |
| タイマ RD 設定ルーチン | init_tmrd | タイマ RD_0・チャンネル 0 設定, ELC の設定 |
| SCI3_1 設定ルーチン | init_sci31 | SCI3 チャンネル 1 の設定 |
| A/D コンバータ終了割り込みルーチン | INT_IADEND_AD1 | A/D 変換結果を RAM に格納, A/D 変換結果の有効値を算出 |
| SCI3_1 受信エラー割り込みルーチン | INT_ERI_SCI31 | 受信エラーフラグのクリア |
| SCI3_1 受信割り込みルーチン | INT_RXI_SCI31 | 受信データのリードおよび格納 受信終了の場合は受信禁止 |

4.2 引数の説明

本タスク例では, 引数を使用していません。

4.3 使用内部レジスタ説明

表 5 に本タスク例における使用内部レジスタを説明します。

表 5 使用内部レジスタ説明

| レジスタ名 | シンボル | 機能 | アドレス | 設定値 |
|-------|-----------|-----------------------------------|----------|------|
| PMR2 | PMR21 | P21 端子を RXD 端子に設定 | H'FF0001 | 1 |
| PMRA | PMRA2 | PB0 端子を AN0 端子に設定 | H'FF0009 | 1 |
| PMRJ | PMRJ[1:0] | PJ0, PJ1 端子を OSC1, OSC2 端子に設定 | H'FF000C | B'11 |
| PUCR2 | PUCR21 | P21 端子のプルアップ MOS をオンに設定 | H'FF0011 | 1 |
| SMR | COM | 調歩同期式モードに設定 | H'FF0550 | 0 |
| | CHR | データ長を 8 ビットのフォーマットで送受信 | | 0 |
| | PE | 送信時はパリティビットを付加し, 受信時はパリティチェックを行う | | 1 |
| | PM | 偶数パリティで送受信に設定 | | 0 |
| | STOP | 1 ストップビットに設定 | | 0 |
| | MP | マルチプロセッサ通信機能を禁止に設定 | | 0 |
| | CKS[1:0] | BRR と合わせて 9600 bps に設定 | | B'00 |
| BRR | | SMR の CKS[1:0] と合わせて 9600 bps に設定 | H'FF0551 | 64 |

| レジスタ名 | シンボル | 機能 | アドレス | 設定値 |
|----------|-------------|---|----------|--------|
| SCR3 | TIE | TXI 割り込み要求を禁止 | H'FF0552 | 0 |
| | RIE | RXI および ERI 割り込み要求を許可 | | 1 |
| | TE | 送信動作を禁止 | | 0 |
| | RE | 受信動作を許可 | | 1 |
| | TEIE | TEI 割り込み要求を禁止 | | 0 |
| | CKE[1:0] | クロックソースを内部ポーレートジェネレータに設定 | | B'00 |
| SSR | RDRF | [1 になる条件] • 受信が正常終了し, RSR から RDR へ受信データが転送されたとき [0 になる条件] • CPU が "1" の状態をリードした後, "0" をライトしたとき • CPU が RDR のデータをリードしたとき • DTC で RDR から受信データを転送, かつ DTC の設定がフラグのクリア条件を満たしたとき* | H'FF0554 | 0 or 1 |
| RDR | | 受信データをリード | H'FF0555 | 不定 |
| ADDR0 | | A/D 変換 0 データのリード | H'FF05E0 | 不定 |
| ADCSR | ADIE | A/D 変換終了割り込みを許可 | H'FF05F0 | 1 |
| | ADST | A/D 変換を停止 | | 0 |
| | CH[3:0] | A/D 変換チャンネルを AN0 に設定 | | B'0000 |
| ADCR | TRGS[1:0] | EXTRGS と組み合わせ, 外部トリガによる A/D 変換開始を禁止 | H'FF05F1 | B'00 |
| | SCANE | シングルモードに設定 | | 0 |
| | CKS[1:0] | A/D 変換時間を 84 ステートに設定 | | B'10 |
| | EXTRGS | TRGS[1:0] と組み合わせ, 外部トリガによる A/D 変換開始を禁止 | | 0 |
| ADMR | ADM1 | A/D 変換モードに設定 | H'FF05F4 | 0 |
| ELSR3 | | タイマ RD_0・チャンネル 0 と SCI3_1・受信データフル信号をリンク設定 | H'FF0683 | H'4C |
| ELSR10 | | A/D 変換とタイマ RD_0・チャンネル 0・コンペアマッチ A 信号をリンク設定 | H'FF068A | H'09 |
| ELOPA | TMRD1M[2:1] | イベント入力時, タイマ RD_0・チャンネル 0 動作をカウントスタートに設定 | H'FF06B5 | B'00 |
| ELCR | ELCON | 全イベントリンク有効に設定 | H'FF06BC | 1 |
| SYSCCR | PHIHSEL | ϕ_{high} クロックソースを ϕ_{osc} に設定 | H'FF06D0 | 1 |
| LPCR1 | PSCSTP | PSC 分周回路動作 | H'FF06D1 | 0 |
| | PHIBSEL | ϕ_{base} クロックソースを ϕ_{high} に設定 | | 1 |
| LPCR2 | PHI[2:0] | システムクロック ϕ を ϕ_{base} に設定 | H'FF06D2 | B'000 |
| LPCR3 | PHIS[2:0] | バスマスタ動作クロック ϕ_s を ϕ に設定 | H'FF06D3 | B'000 |
| OSCCSR | | ϕ_{osc} 発振安定時間設定 | H'FF06D5 | H'0E |
| TMWD | | WDT にクロック入力禁止 | H'FFFF99 | H'F7 |
| TCSRWD | | TMWD 書き込み制御 | H'FFFF9A | H'A3 |
| TRDCNT_0 | | TRDCNT_0 を初期化 | H'FFFFB0 | H'0000 |

| レジスタ名 | シンボル | 機能 | アドレス | 設定値 |
|--------|-----------|--|------------|--------|
| GRA_0 | | タイマ RD_0・チャンネル 0 周期の設定 (A/D 変換周期) | H'FFFFFFB2 | H'7A11 |
| TRDCR | CCLR[2:0] | GRA_0 とコンペアマッチでクリアに設定 | H'FFFFFFC4 | B'001 |
| | CKEG[1:0] | 立ち上がりエッジカウントに設定 | | B'00 |
| | TPSC[2:0] | 内部クロックφ/32 でカウントに設定 | | B'100 |
| TRDSTR | CSTPN0 | TRDCNT_0 と GRA_0 のコンペアマッチが起こったとき, カウントを継続に設定 | H'FFFFFFD2 | 1 |
| | STR0 | TRDCNT_0 はカウント停止に設定 | | 0 |
| MSTCR1 | MSTWDT | WDT モジュールスタンバイ解除 | H'FFFFFFDC | 0 |
| | MSTAD1 | A/D コンバータユニット 1 モジュールスタンバイ解除 | | 0 |
| MSTCR2 | MSTSCI3_1 | SCI3_1 モジュールスタンバイ解除 | H'FFFFFFDD | 0 |
| MSTCR3 | MSTTMRD1 | タイマ RD ユニット 0 モジュールスタンバイ解除 | H'FFFFFFDE | 0 |

【注】 *: DTC は以下の 3 つの条件をすべて満足したときに, 周辺モジュールフラグをクリアします。

1. DIESEL ビットが 0 のとき
2. 転送回転カウンタが 0 でないとき (ノーマルおよびリピートモードの場合: CRA, ブロックモードの場合: CRB)
3. チェイン転送でないとき

4.4 使用 RAM 説明

表 6 に本タスク例における使用 RAM 説明を示します。

表 6 使用 RAM 説明

| ラベル名 | 機能 | メモリ使用量 | 使用モジュール名 |
|-----------|------------------------|-----------|---------------------------|
| ad_sum | A/D 変換結果の総和データを格納 | 4 バイト | init_ad, INT_IADEND_AD1 |
| ad_data[] | A/D 変換結果を格納 | 2 バイト | INT_IADEND_AD1 |
| ad_fix | A/D 変換の有効データを格納 | 2 バイト | INT_IADEND_AD1 |
| ad_max | A/D 変換結果の最大値を格納 | 2 バイト | INT_IADEND_AD1 |
| ad_min | A/D 変換結果の最小値を格納 | 2 バイト | INT_IADEND_AD1 |
| rcv_buf[] | SCI3_1 受信データバッファ | 1 バイト × 8 | INT_RXI_SCI31 |
| index_ad | A/D 変換結果データのインデックス | 1 バイト | init_ad, INT_IADEND_AD1 |
| rcv_index | SCI3_1 受信データバッファインデックス | 1 バイト | init_sci31, INT_RXI_SCI31 |

4.5 使用定義説明

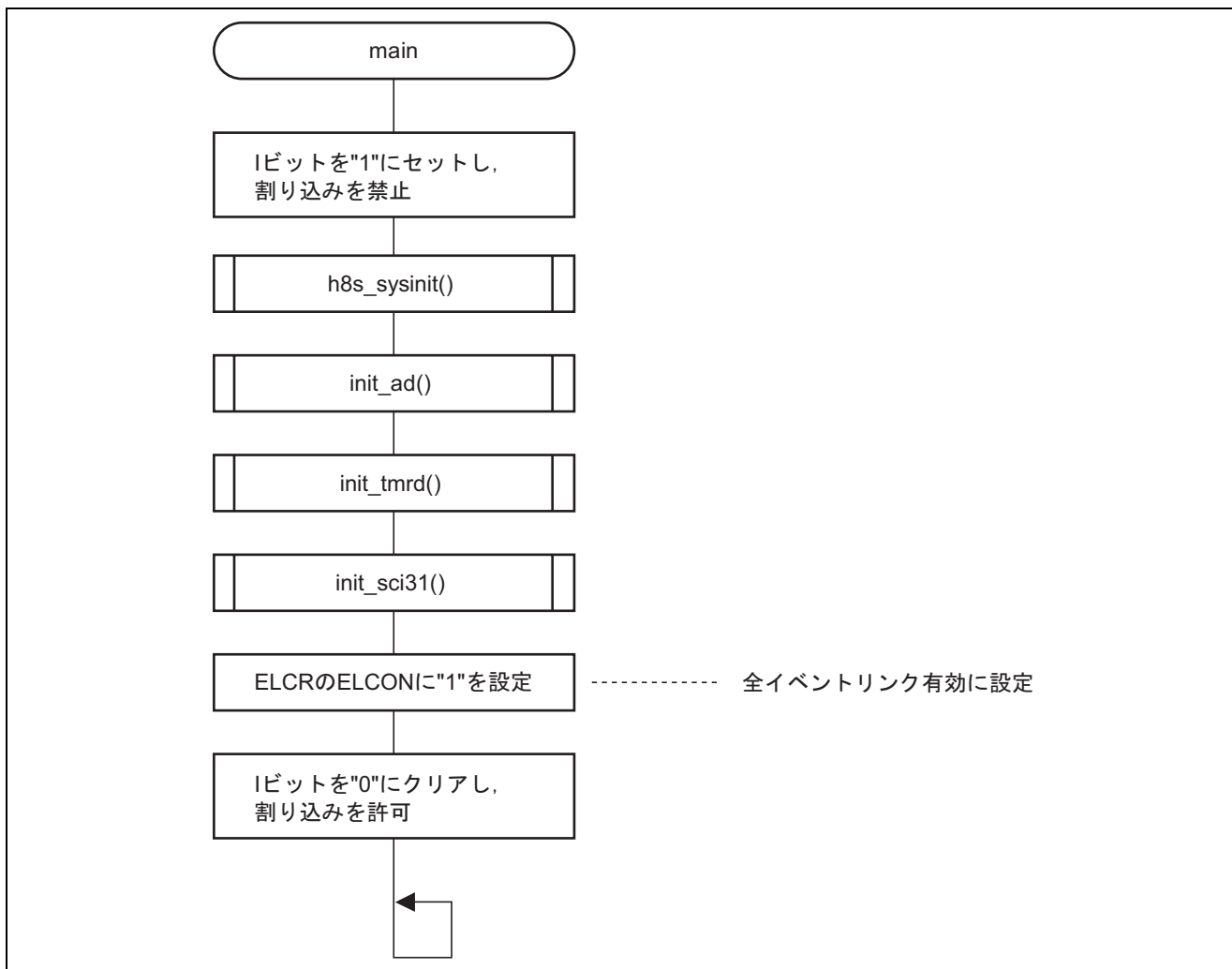
表 7 に本タスク例における使用定義説明を示します。

表 7 使用定義説明

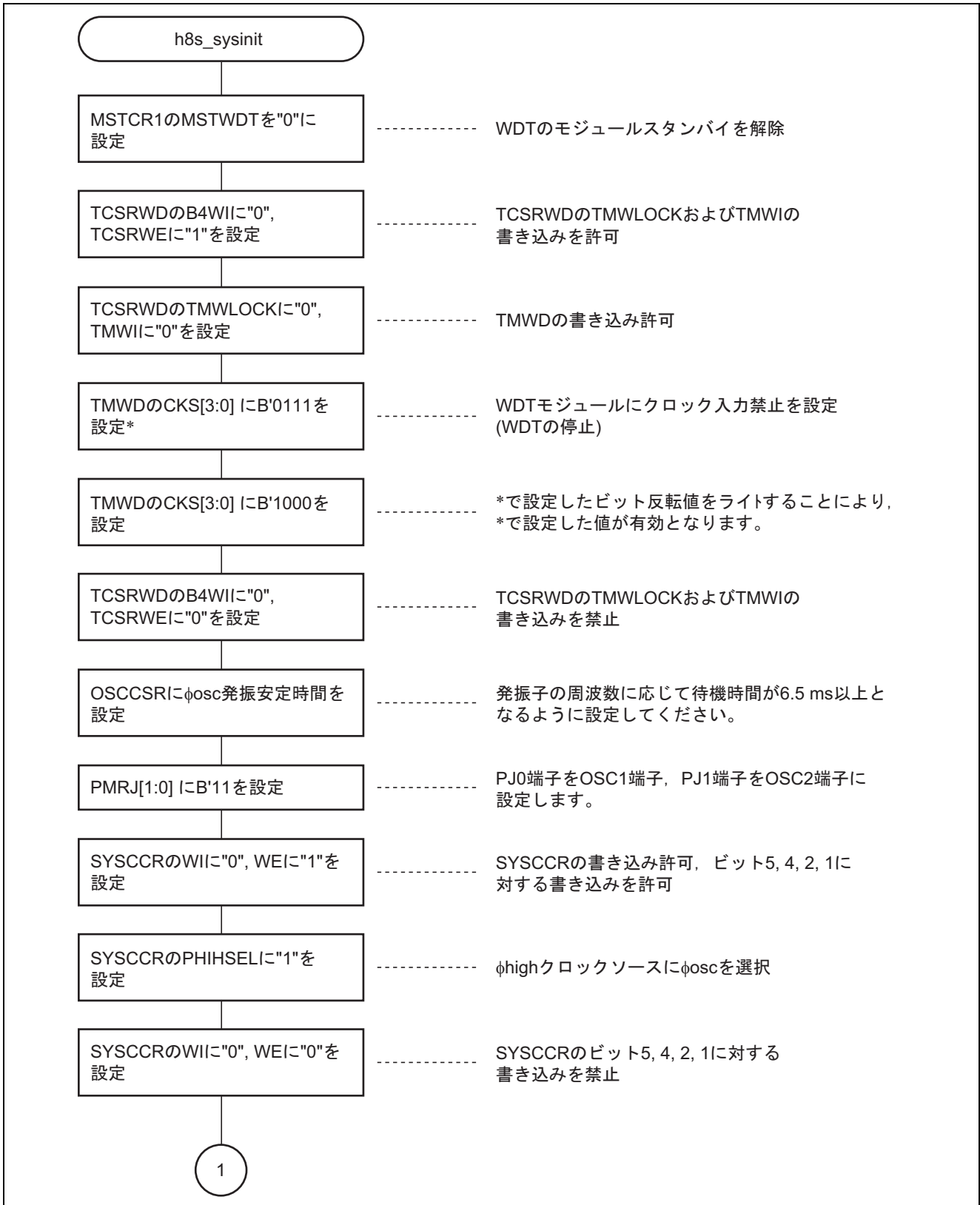
| ラベル名 | 説明 | 定義値 |
|------------|---|--------|
| AD_CYCLE | タイマ RD の $\phi = 20$ MHz, 32 分周時, 50 ms 生成値 | H'7A11 |
| SAMPLING | A/D 変換サンプリング回数 (10 - 1) | 9 |
| FRAME_SIZE | フレーム受信サイズの設定 | 8 |

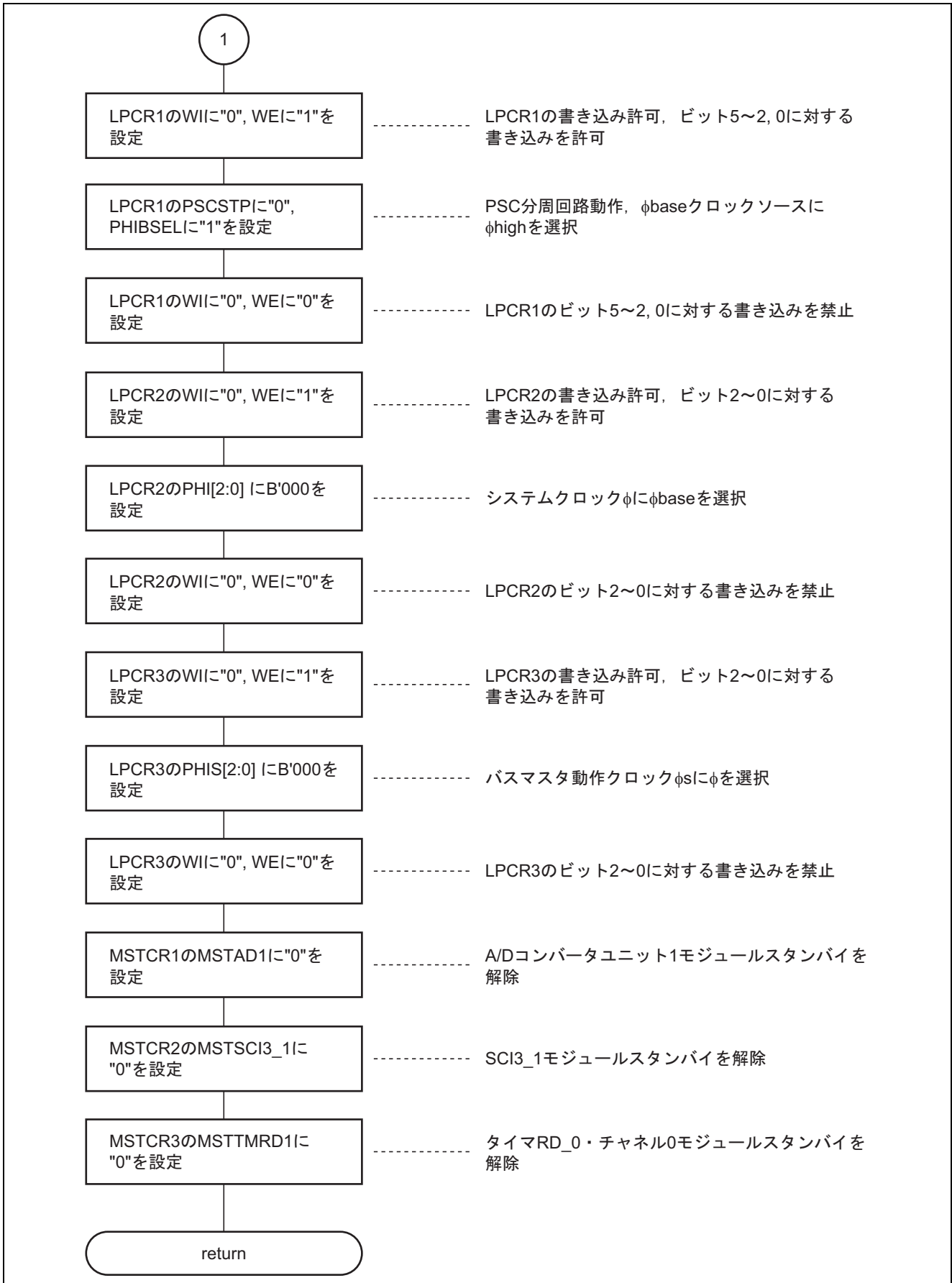
5. フローチャート

5.1 メインルーチン

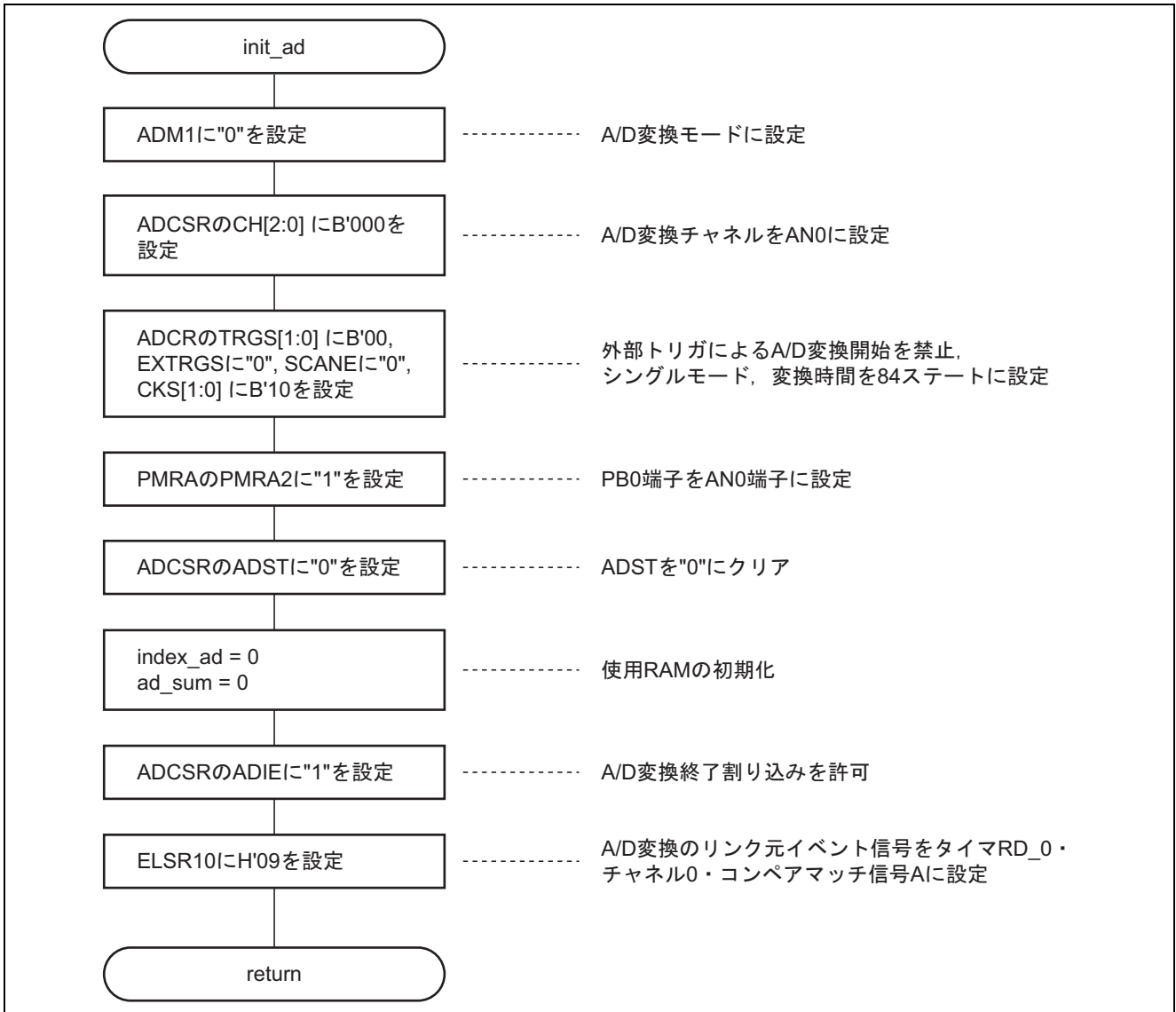


5.2 システムの初期化ルーチン

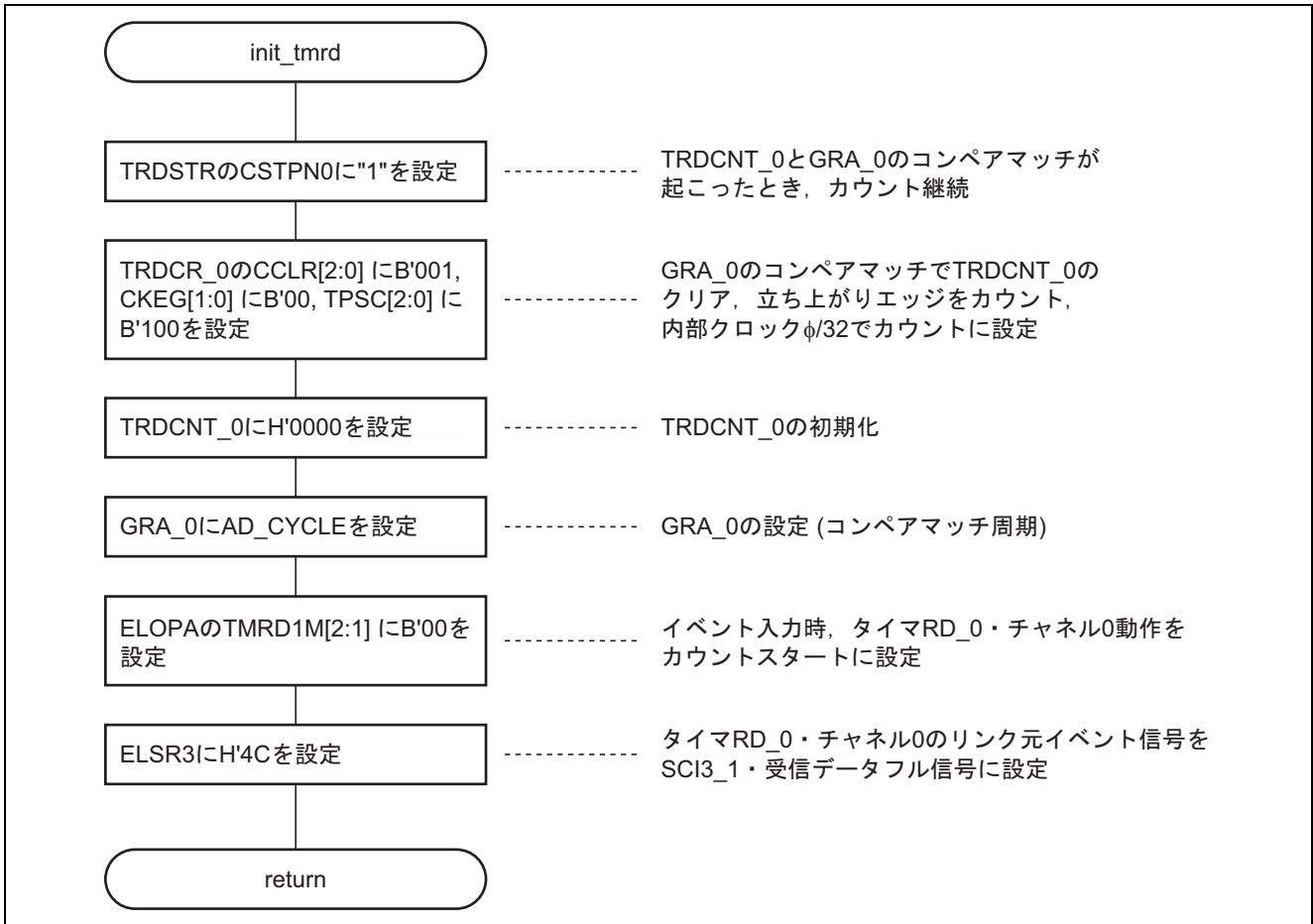




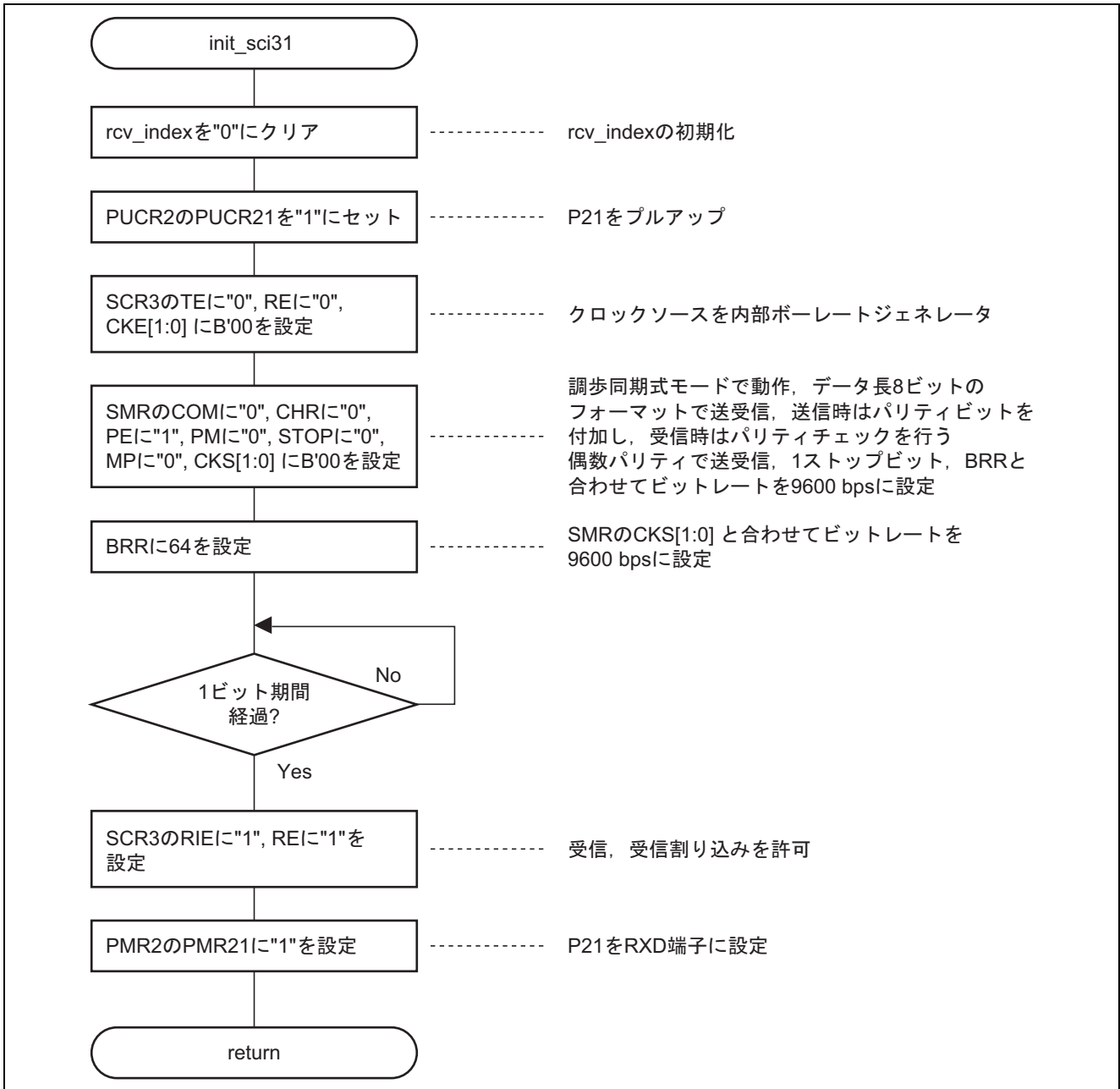
5.3 A/D コンバータ設定ルーチン



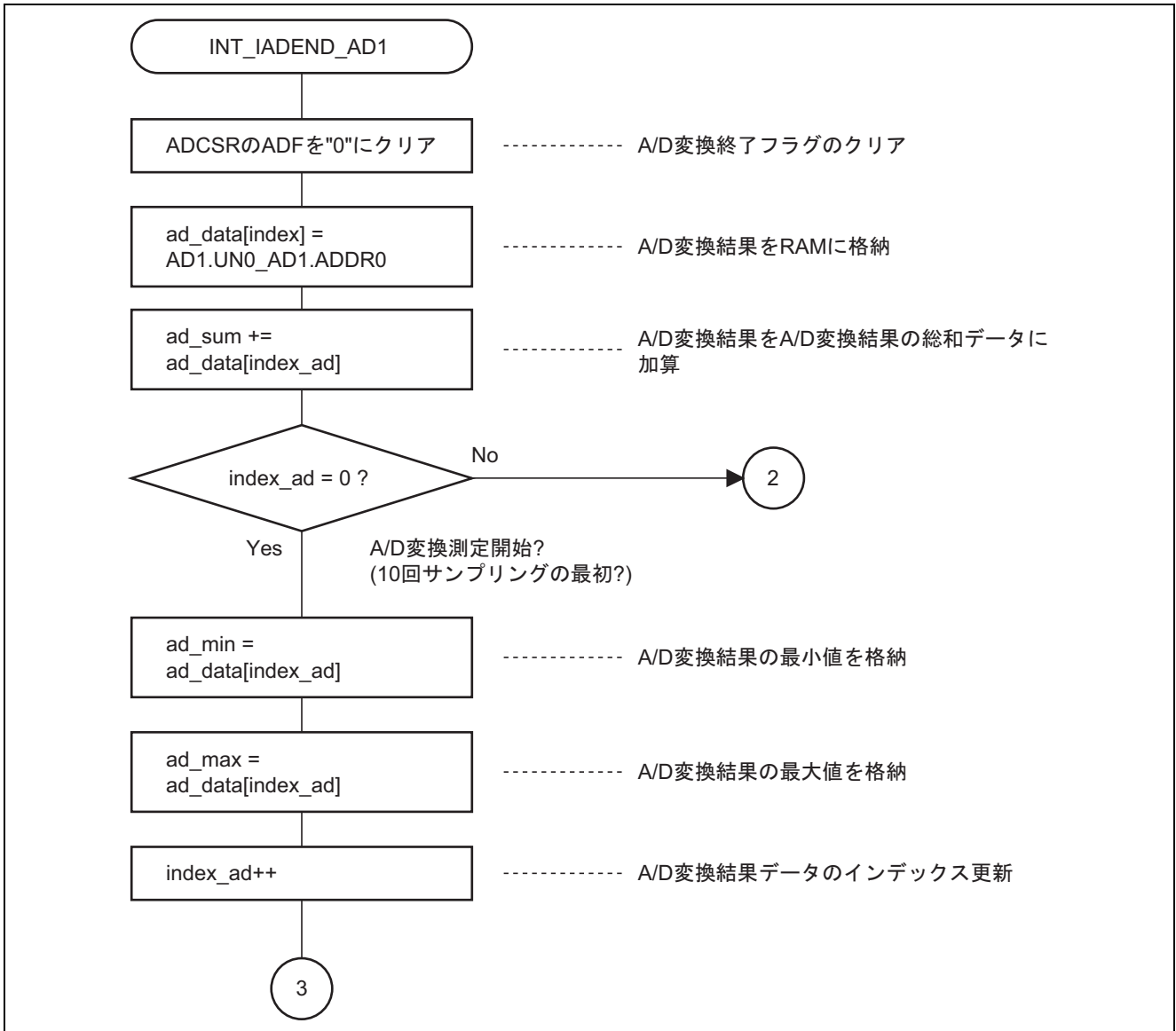
5.4 タイマ RD 設定ルーチン

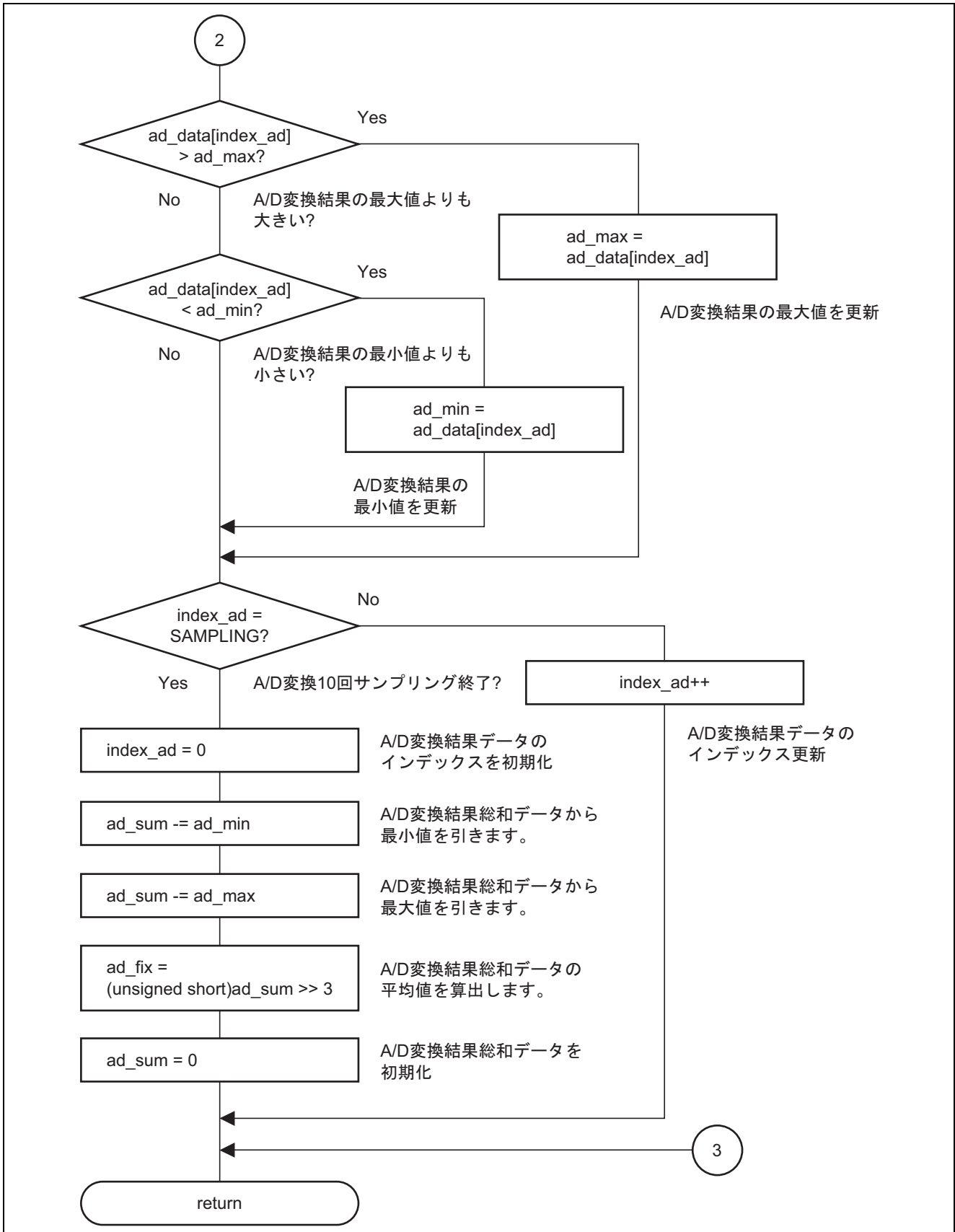


5.5 SCI3_1 設定ルーチン

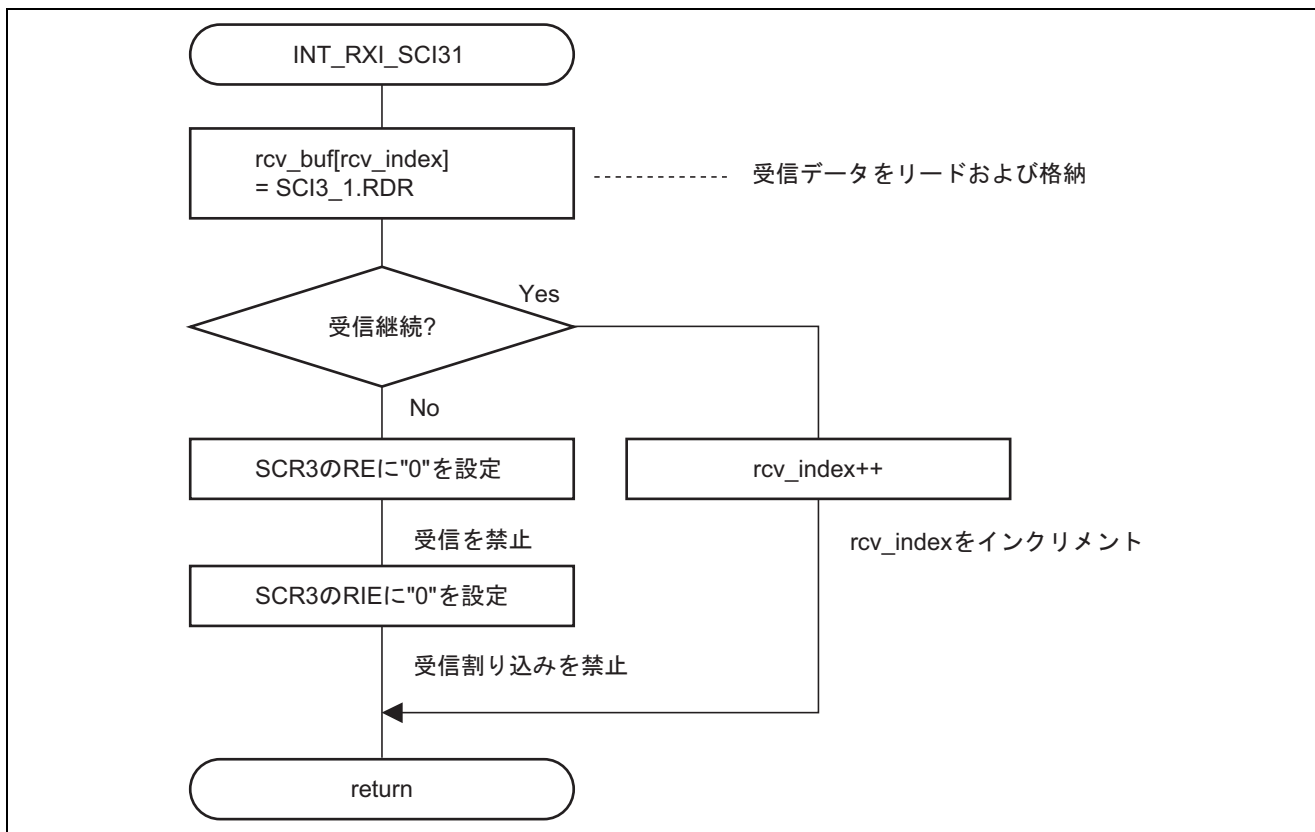


5.6 A/D 変換終了割り込みルーチン

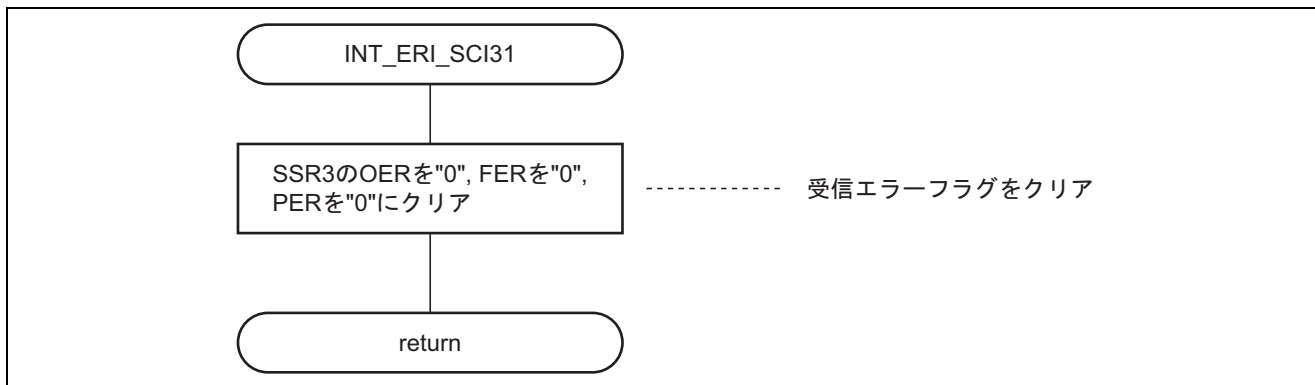




5.7 SCI3_1 受信割り込みルーチン



5.8 SCI3_1 受信エラー割り込みルーチン



6. プログラムリスト

```

<sci3_trd_adc.c>
/*****
/* H8S/2000 Tiny Series -H8S/20203-                               */
/* Application Note                                             */
/*                                                                 */
/* 3 event combination by ELC                                   */
/*                                                                 */
/* Function                                                    */
/* : 3 event combination by ELC (start A/D converter by SCI3_1  */
/*   recieve data full, Timer RD unit0 channel 0 compare match A) */
/*   Event Link A/D converter, Timer RD and SCI3_1             */
/*                                                                 */
/* External Clock : 20MHz                                       */
/* Internal Clock : 20MHz                                       */
*****/
#include <machine.h>
#include "iodefine.h"

/* Timer RD */
/* unit0 (Phi=20MHz, divide 32) */
#define AD_CYCLE          0x7A11                               /* Set GRA_0 (50ms) */

#define SAMPLING          9                                   /* Set A/D sampling (SAMPLING+1) */

/* SCI3_1 */
#define FRAME_SIZE       8                                   /* frame size */

/*****
/*Declaration of function prototype                               */
*****/
void main(void);
void init_ad(void);
void init_tmr(void);
void init_sci31(void);
void h8s_sysinit(void);

```

```

/*****/
/*Definition of RAM area */
/*****/
unsigned long ad_sum; /* A/D sum data */
unsigned short ad_data[SAMPLING+1]; /* save A/D data AN0 */
unsigned short ad_fix; /* save A/D fix data */
unsigned short ad_max; /* save maximum A/D data AN0 */
unsigned short ad_min; /* save minimum A/D data AN0 */
unsigned char index_ad; /* index of A/D data buffer */
unsigned char rcv_buf[FRAME_SIZE]; /* receive buffer */
unsigned char rcv_index; /* receive index */

/*****/
/*Name: main */
/*Parameters: None */
/*Returns: None */
/*Description: User main */
/*****/
void main(void)
{
    set_ccr(0x80); /* set CCR-Ibit */

    h8s_sysinit(); /* initialize system */

    init_ad(); /* initialize A/D converter */

    init_tmr(); /* initialize timer RD */

    init_sci31(); /* initialize SCI3_1 */

    set_imask_ccr(0); /* interrupt enable */

    ELC.ELCR.BIT.ELCON = 1; /* event link enable */

    while(1);
}

```

```

/*****/
/*Name:      init_sci31          */
/*Parameters: None              */
/*Returns:   None               */
/*Description: initialize SCI3_1 */
/*****/
void init_sci31(void)
{
    unsigned short wait;

    rcv_index = 0;                /* clear rcv_index */

    /* port pull up */
    PUCR2.BYTE |= 0x02;          /* pull up P21 */

    SCI3_1.SCR3.BYTE = 0x00;      /* clear TE, RE */
                                /* internal baud rate generator */
    SCI3_1.SMR.BYTE = 0x20;       /* select asynchronous mode */
                                /* even parity, 1 stop bit */
    SCI3_1.BRR = 64;             /* bitrate => 9600 bit/s */

    /* 1bit wait */
    for( wait=0; wait<420; wait++ );

    SCI3_1.SCR3.BYTE |= 0x50;     /* set RIE, RE */
    PMR2.BIT.PMR21 = 1;          /* P21=>RXD */
}

/*****/
/*Name:      init_tmrD          */
/*Parameters: None              */
/*Returns:   None               */
/*Description: initialize timer RD */
/*****/
void init_tmrD(void)
{
    /* unit 0 */
    TRD01.TRDSTR.BYTE = 0xFC;     /* continue count when TRDCNT_0 compare match GRA_0, TRDCNT_0 stop */
}

```

```

TRD0.TRDCR.BYTE = 0x24;          /* clear TRDCNT_0 when compare match GRA_0 */
                                /* select clock Phi/32 */
TRD0.TRDCNT = 0x0000;          /* clear TRDCNT_0 */
TRD0.GRA = AD_CYCLE;          /* set A/D convert cycle */

/* Set event link, SCI3_1 receive data full */
ELC.ELOPA.BYTE = 0xFC;        /* Timer RD_0 count start when event input */
ELC.ELSR3.BYTE = 0x4C;
}

/*****/
/*Name:      init_ad          */
/*Parameters: None          */
/*Returns:   None           */
/*Description: initialize A/D */
/*****/
void init_ad(void)
{
    AD1.ADMR.BYTE = 0x00;        /* select A/D converter mode */
    AD1.ADCSR.BYTE = 0x00;      /* channel select AN0 */
    AD1.ADCR.BYTE = 0x08;      /* single mode CH1 */
                                /* A/D convert time => 84 state */
    PMRA.BIT.PMRA2 = 1;        /* PB0 => AN0 */

    AD1.ADCSR.BIT.ADST = 0;     /* clear ADST */

    index_ad = 0;              /* clear index_ad */
    ad_sum = 0;                /* clear ad_sum */

    AD1.ADCSR.BIT.ADIE = 1;     /* A/D interrupt enable */

    /* Set event link, Timer RD unit0 channel 0 compare match A */
    ELC.ELSR10.BYTE = 0x09;
}

```

```

/*****/
/*Name:      h8s_sysinit          */
/*Parameters: None                */
/*Returns:   None                 */
/*Description: initialize H8S/20203 */
/*****/
void h8s_sysinit(void)
{
    MSTCR1.BIT.MSTWDT = 0;          /* WDT module standby off */

    /* stop WDT */
    WDT.TCSRWD.BYTE = 0x97;        /* write enable TMWLOCK, TMWI */
    WDT.TCSRWD.BYTE = 0xA3;        /* write enable TMWD */
    WDT.TMWD.BYTE = 0xF7;         /* Not select clock source */
    WDT.TMWD.BYTE = 0xF8;         /* write bit inversion */
    WDT.TCSRWD.BYTE = 0x87;        /* write disable TMWLOCK, TMWI */

    CPG.OSCCSR.BYTE = 0x0E;        /* wait over 6.5ms, Phi_osc=20MHz */
    PMRJ.BYTE = 0x03;             /* select OSC1,OSC2 */

    CPG.SYSCCR.BYTE = (CPG.SYSCCR.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.SYSCCR.BYTE = 0x60;        /* high=Phi_osc, Phi_low=Phi_loco */
    CPG.SYSCCR.BYTE = CPG.SYSCCR.BYTE & 0x3F; /* WI=0, WE=0 */

    CPG.LPCR1.BYTE = (CPG.LPCR1.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.LPCR1.BYTE = 0x41;         /* PSC on, Phi_base=Phi_high */
    CPG.LPCR1.BYTE = CPG.LPCR1.BYTE & 0x3F; /* WI=0, WE=0 */

    CPG.LPCR2.BYTE = (CPG.LPCR2.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.LPCR2.BYTE = 0x40;         /* select system clock */
    CPG.LPCR2.BYTE = CPG.LPCR2.BYTE & 0x3F; /* WI=0, WE=0 */

    CPG.LPCR3.BYTE = (CPG.LPCR3.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.LPCR3.BYTE = 0x40;         /* select clock of bus master */
    CPG.LPCR3.BYTE = CPG.LPCR3.BYTE & 0x3F; /* WI=0, WE=0 */

    /* module standby off */
    MSTCR1.BIT.MSTAD1 = 0;         /* A/D module standby off */
    MSTCR2.BIT.MSTSCI3_1 = 0;     /* SCI3_1 module standby off */
    MSTCR3.BIT.MSTTMRD1 = 0;     /* Timer RD unit0 module standby off */
}

```



```

<intrpg.c>
/*****/
/*Definition of const data */
/*****/
#define SAMPLING          9          /* Set A/D sampling (SAMPLING+1) */

/* SCI3_1 */
#define FRAME_SIZE       8          /* frame size */

/*****/
/*Extern Definition of RAM area */
/*****/
extern unsigned long ad_sum;          /* A/D sum data */
extern unsigned short ad_data[SAMPLING+1]; /* save A/D data AN0 */
extern unsigned short ad_fix;          /* save A/D fix data */
extern unsigned short ad_max;          /* save maximum A/D data AN0 */
extern unsigned short ad_min;          /* save minimum A/D data AN0 */
extern unsigned char index_ad;          /* index of A/D data buffer */
extern unsigned char rcv_buf[FRAME_SIZE]; /* receive buffer */
extern unsigned char rcv_index;          /* receive index */

// vector 30 END AD1
__interrupt(vect=30) void INT_IADEND_AD1(void) {
    AD1.ADCSR.BIT.ADF = 0;          /* clear ADF */

    ad_data[index_ad] = AD1.UN0_AD1.ADDR0; /* read ADDR0 */
    ad_sum += ad_data[index_ad];          /* add to ad_sum */

    /* start sampling ? */
    if( index_ad == 0 ){
        ad_min = ad_data[index_ad];          /* update ad_min */
        ad_max = ad_data[index_ad];          /* update ad_max */
        index_ad++;
    }
    else{
        /* ad_max over ? */
        if ( ad_data[index_ad] > ad_max ){
            ad_max = ad_data[index_ad];          /* update ad_max */
        }
        /* ad_min under ? */
        else if( ad_data[index_ad] < ad_min ){
            ad_min = ad_data[index_ad];          /* update ad_min */
        }
    }

    /* sampling end ? */
    if( index_ad == SAMPLING ){

```

```

        index_ad = 0;
        ad_sum -= ad_min;
        ad_sum -= ad_max;

        /* calculate average ad_data */
        ad_fix = (unsigned short)(ad_sum >> 3);
        ad_sum = 0;
    }
    else{
        index_ad++;
    }
}

// vector 37 ERI SCI31
__interrupt(vect=37) void INT_ERI_SCI31(void) {
    /* clear receive error flag */
    SCI3_1.SSR.BYTE &= ~0x38;
}

// vector 38 RXI SCI31
__interrupt(vect=38) void INT_RXI_SCI31(void) {
    /* read receive data */
    rcv_buf[rcv_index] = SCI3_1.RDR;

    /* receive continue */
    if ( rcv_index < (FRAME_SIZE-1) ){
        rcv_index++;
    }
    /* frame end */
    else{
        SCI3_1.SCR3.BIT.RE = 0;          /* receive disable */
        SCI3_1.SCR3.BIT.RIE = 0;       /* receive data hull interrupt disable */
    }
}

```

6.1 リンクアドレス指定

| セクション名 | アドレス |
|------------------------|----------|
| PResetPRG, PIntPRG | H'000400 |
| P, C\$DSEC, C\$BSEC, D | H'000800 |
| B, R | H'FFDF80 |
| S | H'FFFD80 |

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://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

| Rev. | 発行日 | 改訂内容 | |
|------|------------|------|------|
| | | ページ | ポイント |
| 1.00 | 2008.12.24 | — | 初版発行 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

すべての商標および登録商標は、それぞれの所有者に帰属します。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事事業の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますと、事前に弊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、弊社ホームページ (<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。
5. 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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等については弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444