

SH7216グループ

RJJ06B1173-0100

Rev.1.00

2010.07.30

A/D 変換器、データトランスファコントローラ、 およびシリアルコミュニケーションインタフェースの組み合わせ使用例

要旨

本アプリケーションノートは、A/D 変換器、データトランスファコントローラ、およびシリアルコミュニケーションインタフェースの組み合わせ使用例について説明しています。

動作確認デバイス

SH7216

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムリスト.....	27
4. 参考ドキュメント.....	46

1. はじめに

1.1 仕様

データトランスファコントローラを使用して、A/D 変換結果を以下の手順でシリアル送信します。

- コンペアマッチ割り込みによりデータトランスファコントローラを起動して、あらかじめ内蔵 RAM に格納した A/D 変換開始設定データを A/D コントロールレジスタへ転送し、A/D 変換器を起動します。
- A/D 変換終了割り込みによりデータトランスファコントローラを起動して、ブロック転送モードで A/D 変換結果を内蔵 RAM に転送します。このとき、チェイン転送を使用して、あらかじめ内蔵 RAM に格納した送信データエンプティ割り込み許可設定データをシリアルコントロールレジスタへ転送し、送信データエンプティ割り込みを発生します。
- 送信データエンプティ割り込みによりデータトランスファコントローラを起動して、内蔵 RAM に転送した A/D 変換結果をシリアルコミュニケーションインタフェースへ転送し、調歩同期式モードでシリアル送信します。

1.2 使用機能

- コンペアマッチタイマ (CMT)
- A/D 変換器 (ADC)
- データトランスファコントローラ (DTC)
- シリアルコミュニケーションインタフェース (SCI)

1.3 適用条件

マイコン	SH7216
動作周波数	内部クロック : 200 MHz バスクロック : 50 MHz 周辺クロック : 50 MHz AD クロック : 50 MHz
統合開発環境	ルネサス エレクトロニクス製 High-performance Embedded Workshop Ver.4.07.00
C コンパイラ	ルネサス エレクトロニクス製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.03 Release 00
コンパイルオプション	High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。合わせて参照してください。

- SH7216 グループ 初期設定例

2. 応用例の説明

2.1 使用機能の動作概要

2.1.1 A/D変換器

A/D 変換器は、12 ビットの分解能をもつ 4 チャンネル入力の A/D モジュールを 2 つ (A/D_0、A/D_1) 搭載しています。A/D 変換されたデータは A/D データレジスタ (ADDR) に格納されます。

A/D 変換器の動作モードには 1 サイクルスキャンモードと連続スキャンモードがあります。1 サイクルスキャンモードは、指定した 1 チャンネル以上の A/D 変換を 1 回行った後、A/D 変換待機状態になるモードです。連続スキャンモードは、指定した 1 チャンネル以上の A/D 変換を繰り返し行うモードです。また、A/D 変換終了時には CPU に対し A/D 変換終了割り込みを発生することが可能です。また、A/D 変換終了割り込み発生時にはダイレクトメモリアクセスコントローラ/データトランスファコントローラを起動することが可能です。*

表 1 に A/D 変換器の仕様を、また 図 1 にブロック図を示します。A/D 変換器についての詳細は、「SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編 A/D 変換器 (ADC)」の章を参照してください。

【注】 *ダイレクトメモリアクセスコントローラ起動時は CPU に対する割り込みは発生しません。また、ダイレクトメモリアクセスコントローラを起動できるのは A/D モジュール 0 (A/D_0) のみです。

表 1 A/D 変換器の仕様

項目	内容
分解能	12 ビット
変換スピード	最小変換時間 1 チャンネル当たり 1.0 μ s (A ϕ = 50 MHz 動作時)
モジュール数	2 モジュール (A/D_0、A/D_1)
入力チャンネル数	8 チャンネル (AN0~AN7)
動作モード	<ul style="list-style-type: none"> ● 1 サイクルスキャンモード ● 連続スキャンモード
サンプル&ホールド機能	<ul style="list-style-type: none"> ● チャンネル 0~3 共通 : 1 回路、チャンネル 4~7 共通 : 1 回路 ● チャンネル 0~2 : それぞれ専用に 1 回路 (計 3 回路)
A/D 変換起動要因	<ul style="list-style-type: none"> ● ソフトウェア : ADST ビットの設定 ● タイマ : <ul style="list-style-type: none"> — マルチファンクションタイマパルスユニット 2 の TRGAN、TRG0N、TRG4AN、TRG4BN — マルチファンクションタイマパルスユニット 2S の TRGAN、TRG4AN、TRG4BN ● 外部トリガ : ADTRG

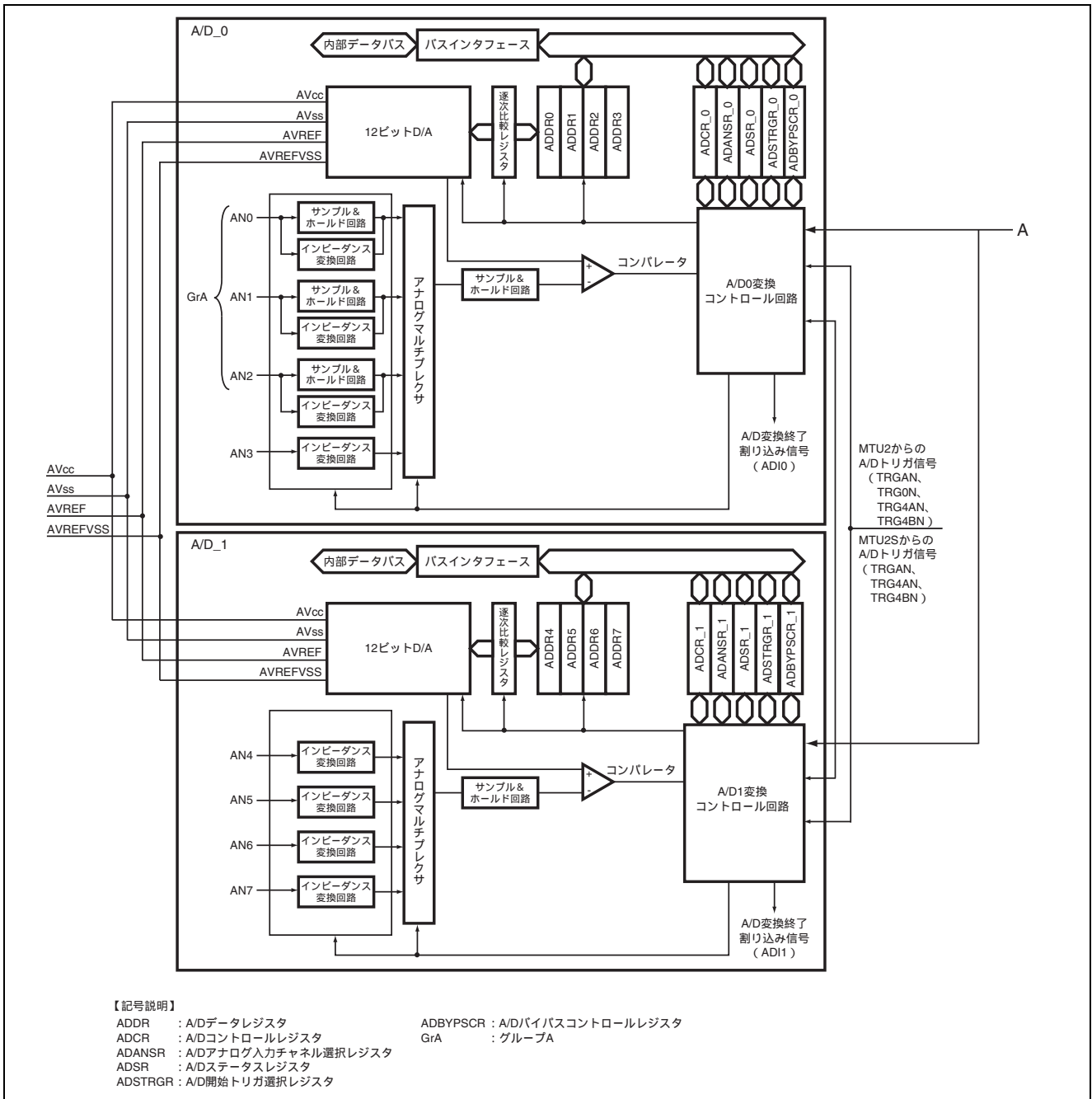


図 1 A/D 変換器のブロック図

2.1.2 データトランスファコントローラ

データトランスファコントローラは、割り込み要求により起動するデータ転送モジュールです。動作モードはノーマル転送モード、リピート転送モード、ブロック転送モードの3種類あり、転送情報をデータ領域に格納することで、任意のチャンネル数のデータ転送を行うことができます。データトランスファコントローラが起動すると、データ領域から転送情報をリードしてデータ転送を行い、データ転送後の転送情報をライトバックします。

データトランスファコントローラについての詳細は、「SH7214 グループ、SH7216 グループ ユーザーズマニュアル ハードウェア編 データトランスファコントローラ (DTC)」の章を参照してください。

表 2 にデータトランスファコントローラの仕様を、また図 2 にブロック図を示します。

表 2 データトランスファコントローラの仕様

項目	内容
チャンネル数	任意のチャンネル数を設定可能
転送モード	<ul style="list-style-type: none"> ノーマル転送モード リピート転送モード ブロック転送モード
転送回数	<ul style="list-style-type: none"> ノーマル転送モード：1~65536 回 リピート転送モード：1~256 回 ブロック転送モード：1~65536 回
転送データサイズ	バイト、ワード (2 バイト)、ロングワード (4 バイト)
CPU 割り込み要求	<ul style="list-style-type: none"> 1 回のデータ転送終了後、CPU に対して割り込み要求可能 指定したデータ転送終了後、CPU に対して割り込み要求可能
起動要因	<ul style="list-style-type: none"> 外部端子 (8 要因) 内蔵周辺モジュール (55 要因)
その他	<ul style="list-style-type: none"> 1 つの起動要因に対して複数のデータ転送が可能 (チェイン転送) 転送情報のリードスキップモードを設定可能 モジュールストップモード設定可能 ショートアドレスモード設定可能 バス権解放タイミングを 3 種類から選択可能 データトランスファコントローラ起動時の優先順位を 2 種類から選択可能

【注】 転送元もしくは転送先の少なくともどちらか一方は必ず内蔵周辺モジュールに設定してください。
外部メモリ、メモリマップト外部デバイス、内蔵メモリ間の転送はできません。

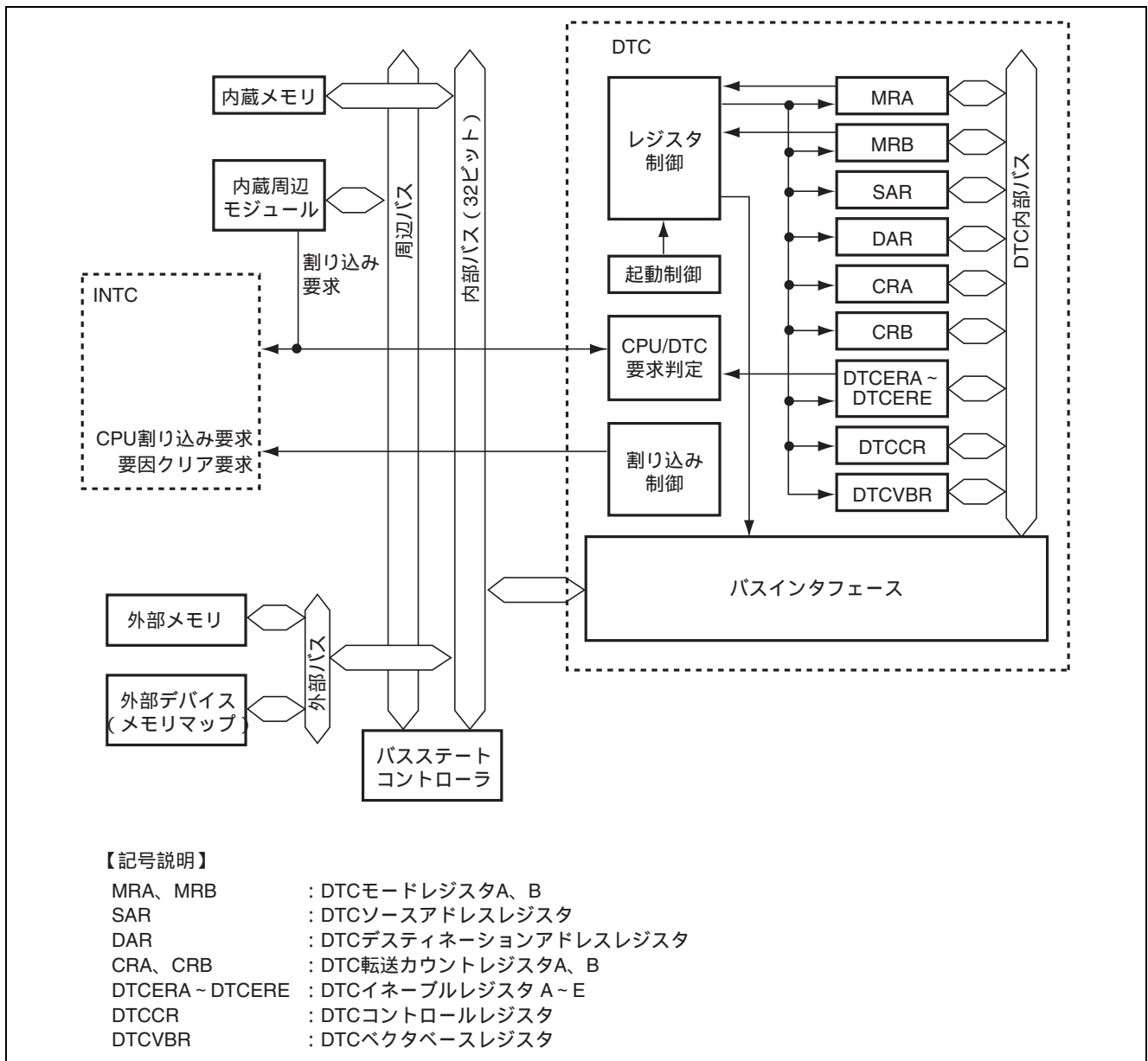


図 2 データトランスファコントローラのブロック図

2.1.3 シリアルコミュニケーションインタフェース（調歩同期式モード）

シリアルコミュニケーションインタフェースの調歩同期式モードでは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータ付加したキャラクタを送信／受信し、1キャラクタ単位で同期を取りながらシリアル通信を行います。また、各チャネルともに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。さらに、送信部と受信部はともにダブルバッファ構造となっているため、シリアルデータの高速度連続送信／受信が可能です。

調歩同期式シリアル通信では、通信回線は通常、マーク状態（ハイレベル）に保たれています。シリアルコミュニケーションインタフェースは通信回線を監視し、スペース（ローレベル）になったところをスタートビットとみなしてシリアル通信を開始します。

シリアル通信の1キャラクタは、スタートビット（ローレベル）から始まり、データ（LSB ファースト：最下位ビットから）、パリティビット（ハイ／ローレベル）、最後にストップビット（ハイレベル）の順で構成されています。

シリアルコミュニケーションインタフェースについての詳細は、「SH7214 グループ、SH7216 グループユーザーズマニュアル：ハードウェア編 シリアルコミュニケーションインタフェース（SCI）」の章を参照してください。

表 3に、調歩同期式シリアル通信の仕様を、また図 3にシリアルコミュニケーションインタフェースのブロック図を示します。

表 3 調歩同期式シリアル通信の仕様

項目	内容
チャネル数	4チャネル（SCI0、SCI1、SCI2、SCI4）
クロックソース	内部クロック／外部クロックから選択可能： <ul style="list-style-type: none"> 内部クロック：Pϕ、Pϕ/4、Pϕ/16、Pϕ/64 — SCIはボーレートジェネレータのクロックで動作し、ビットレートの16倍の周波数クロックを出力することが可能。 外部クロック：SCK端子入力クロック — ビットレートの16倍の周波数クロックを入力することが必要。 (内部ボーレートジェネレータを使用しない)
データフォーマット	<ul style="list-style-type: none"> 転送データ長：7ビット／8ビットから選択 転送順序：LSBファースト／MSBファースト選択可能 (7ビットデータ長の場合はLSBファーストのみ)
ボーレート	<ul style="list-style-type: none"> 内部クロック選択時：110 bps ~ 1.5625 Mbps（Pϕ=50 MHz動作時） 外部クロック選択時：最大781250 bps (Pϕ=50 MHz外部クロック入力15 MHz動作時)
エラーの検出	フレーミングエラー、パリティエラー、オーバランエラー、およびブレイク検出可能
割り込み要求	<ul style="list-style-type: none"> 送信データエンプティ割り込み（TXI） 受信データフル割り込み（RXI） 受信エラー割り込み（ERI） 送信終了割り込み（TEI）

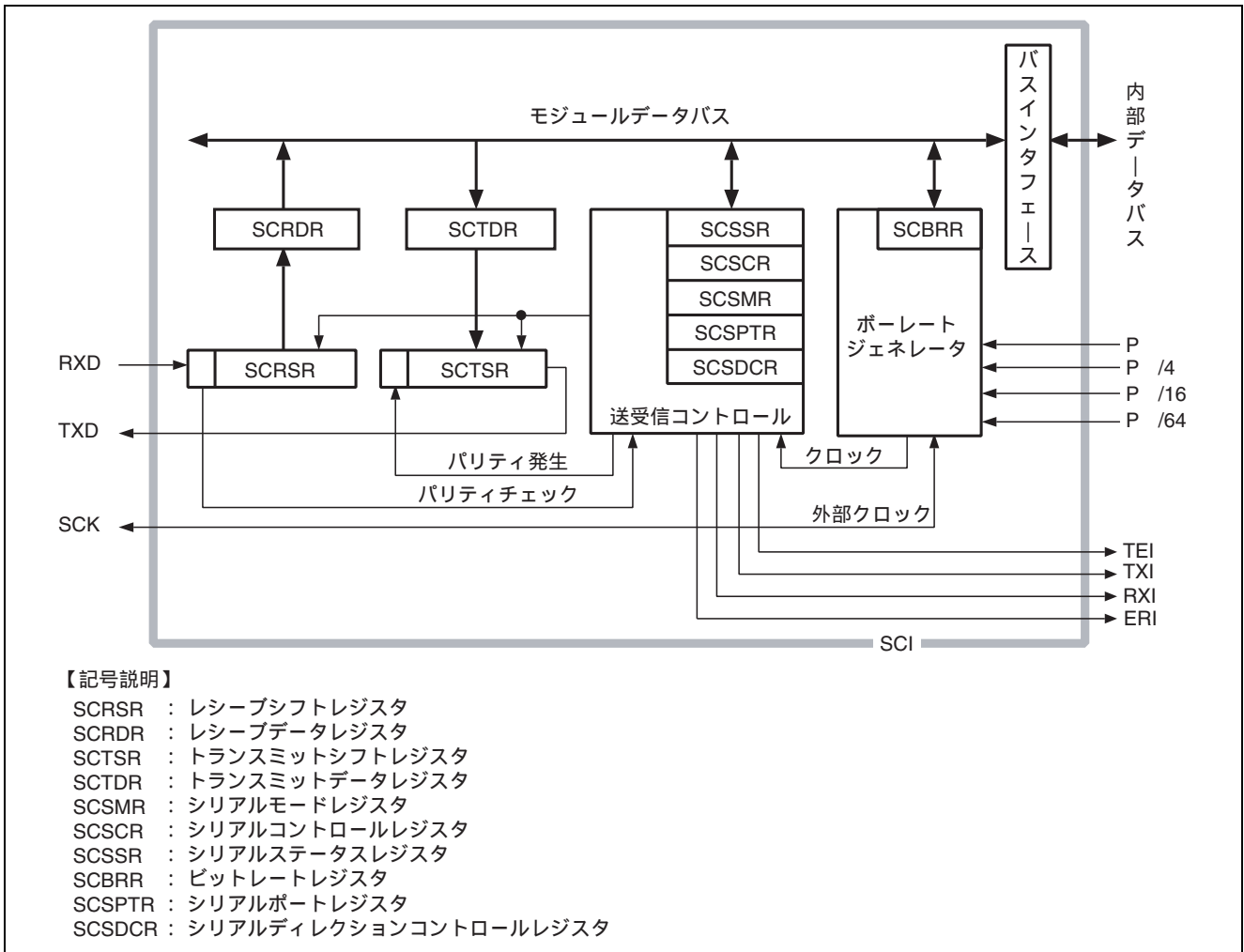


図 3 シリアルコミュニケーションインタフェースのブロック図

2.2 使用機能の設定手順

2.2.1 A/D変換器の設定手順

図 4に、本応用例で使用するA/D変換器の初期設定フローを示します。なお、各レジスタ設定の詳細は、「SH7214 グループ、SH7216 グループユーザズマニュアル：ハードウェア編」を参照してください。

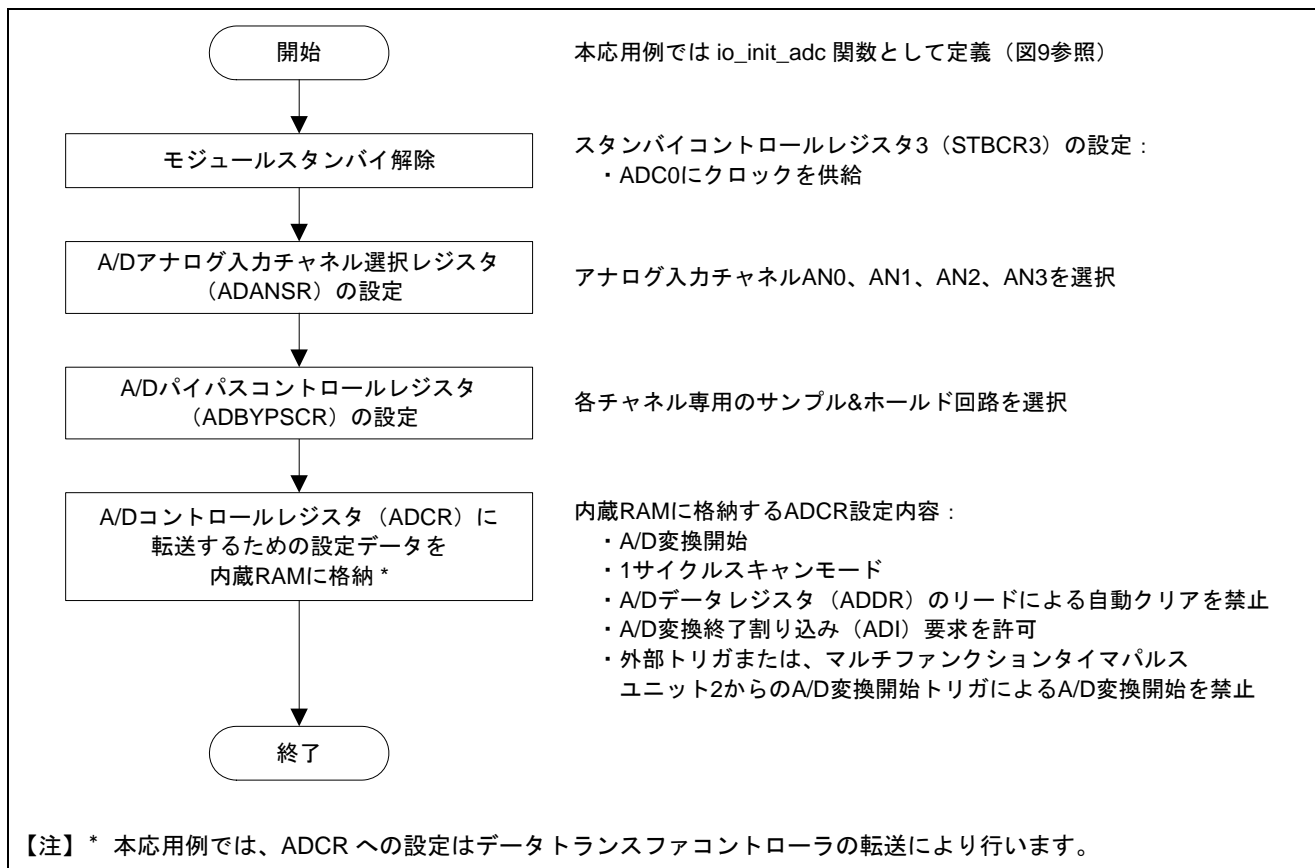


図 4 A/D 変換器の初期設定フロー

2.2.2 データトランスファコントローラの設定手順

図 5に、本応用例で使用するデータトランスファコントローラの初期設定フローを示します。なお、各レジスタ設定の詳細は、「SH7214 グループ、SH7216 グループ ユーザーズマニュアル：ハードウェア編」を参照してください。

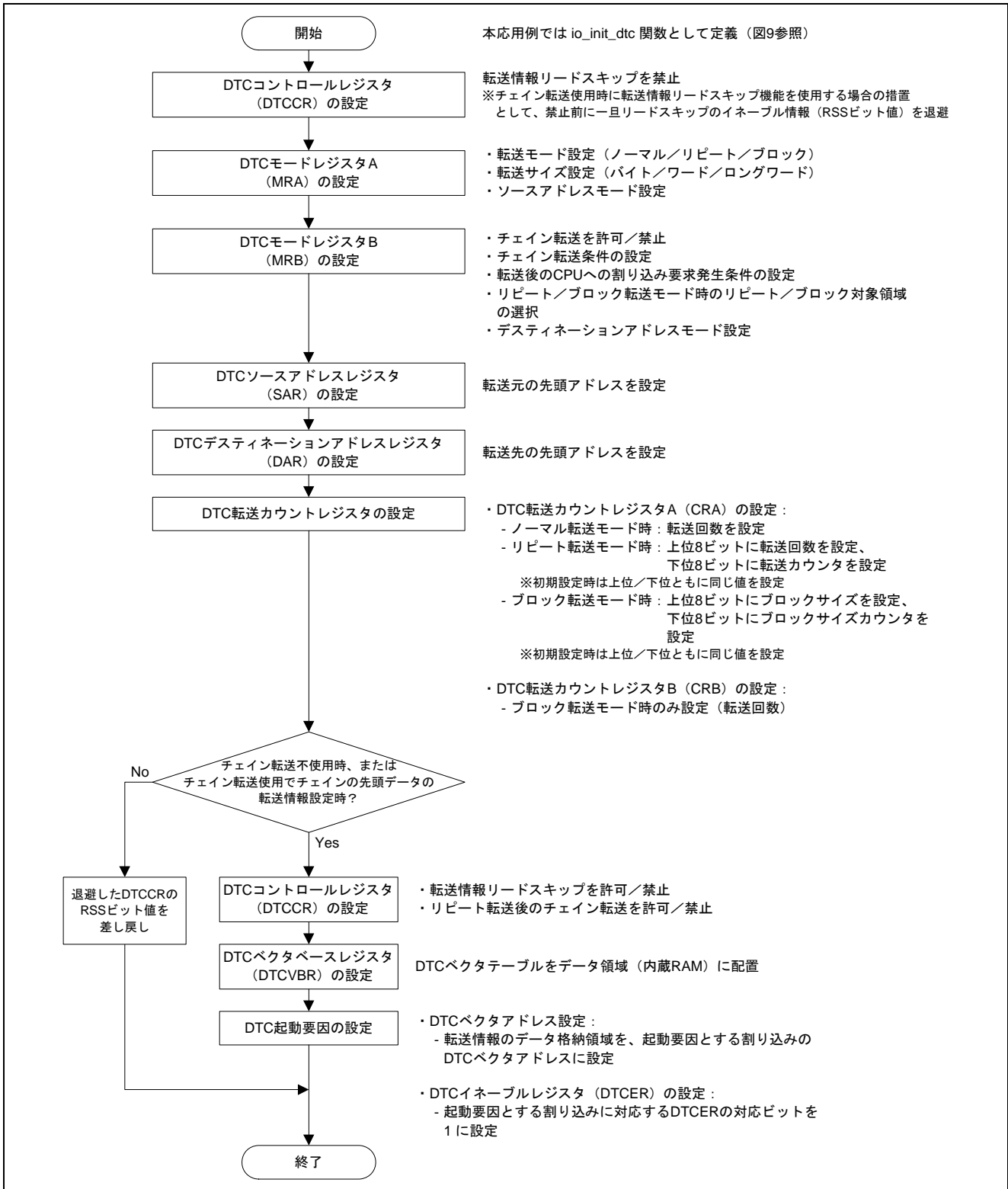


図 5 データトランスファコントローラの初期設定フロー

2.2.3 シリアルコミュニケーションインタフェースの設定手順

図 6に本応用例で使用する、シリアルコミュニケーションインタフェースの初期設定フローを示します。なお、各レジスタ設定の詳細は、「SH7214 グループ、SH7216 グループユーザーズマニュアル：ハードウェア編」を参照してください。

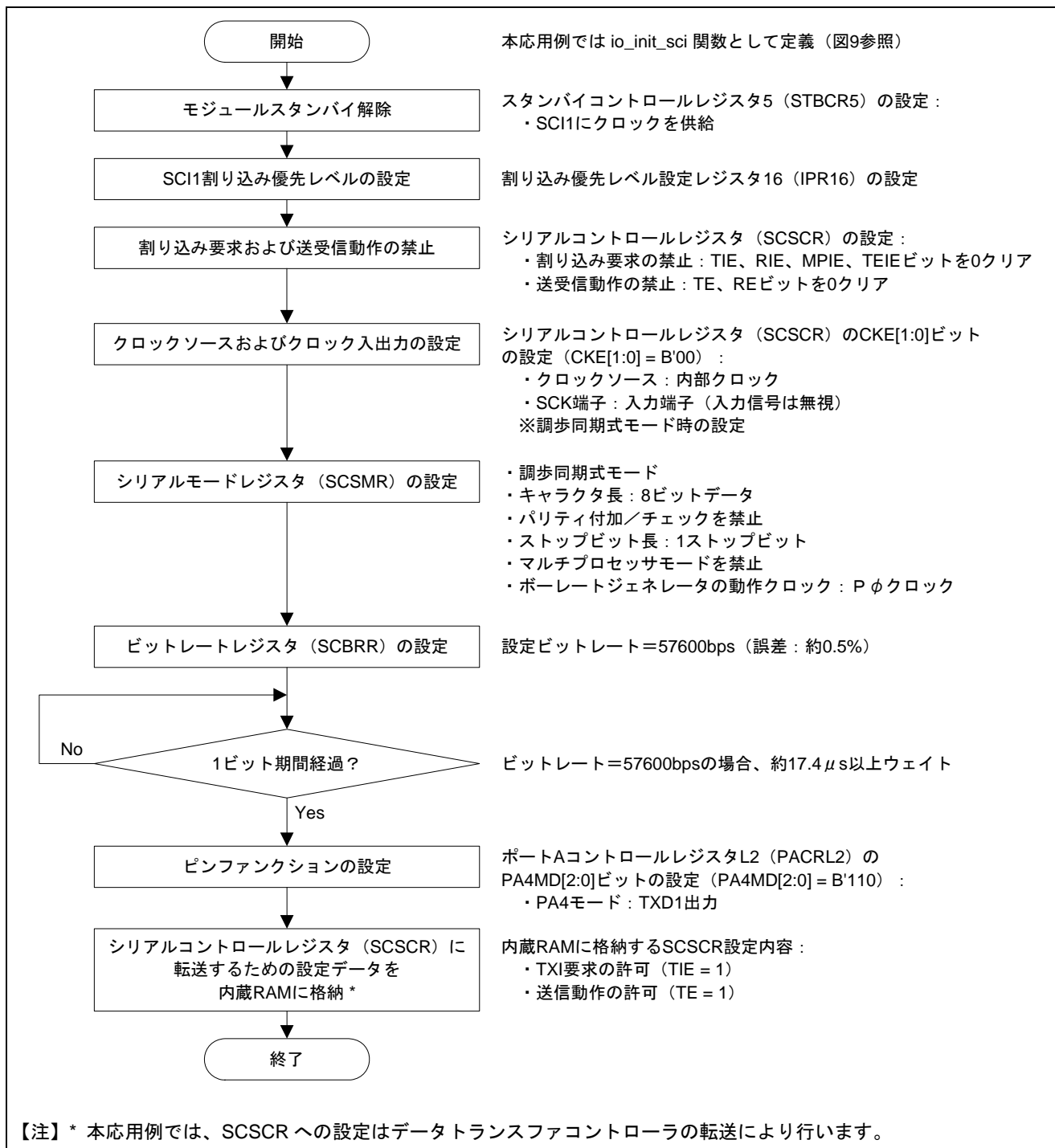


図 6 シリアルコミュニケーションインタフェースの初期設定フロー

2.3 参考プログラムの処理手順

2.3.1 参考プログラムの動作概要

参考プログラムでは、3種類の割り込み起動要因を使用してデータトランスファコントローラを連動的に起動し、4チャンネル分のA/D変換結果を内蔵RAMへ格納後、シリアルコミュニケーションインタフェースに転送して連続送信を行います。この一連の動作は5ms周期ごとに計10回行います。

以下に、参考プログラムで使用するデータトランスファコントローラの3種類の割り込み起動要因とそれぞれの転送内容について説明します。また、表4に、参考プログラムにおけるデータトランスファコントローラの主な動作仕様を示します。

1. コンペアマッチ割り込み (CMIO)

CMIOを起動要因とするデータトランスファコントローラでは、あらかじめ内蔵RAMに格納したA/D変換開始設定データを、A/Dコントロールレジスタに転送します。この転送が終了すると、A/D変換器が起動して4チャンネル分 (AN0~AN3) のA/D変換を行います。A/D変換終了後は、A/D変換終了割り込み (ADIO)が発生します。

2. A/D変換終了割り込み (ADIO)

ADIOを起動要因とするデータトランスファコントローラでは、チェーン転送を使用して、以下の2種類のデータ転送を行います。

- (1) あらかじめ内蔵RAMに格納した送信データエンティ割り込み (TXI1) 許可設定データをシリアルコントロールレジスタへ転送。
- (2) ブロック転送モードを使用して、4チャンネル分のA/D変換結果 (A/Dデータレジスタ0~3) を内蔵RAMへ転送。

(1)の転送が終了すると、TXI1が発生しますが、TXI1を起動要因とするデータトランスファコントローラ転送は、(1)および(2)の転送終了後に行われます。

3. 送信データエンティ割り込み (TXI1)

TXI1を起動要因とするデータトランスファコントローラでは、内蔵RAMに転送した4チャンネル分のA/D変換結果 (計8バイト) をトランスミットデータレジスタに転送し、シリアル送信を行います。

表4 参考プログラムにおけるデータトランスファコントローラの主な動作仕様

起動要因	転送元	転送先	転送回数	備考
コンペアマッチ 割り込み (CMIO)	内蔵 RAM	A/D コントロール レジスタ (ADCR_0)	10	転送元のデータは、 コンペアマッチタイマの 初期設定時に内蔵 RAM に格納
A/D 変換終了割り込み (ADIO)	内蔵 RAM	シリアルコントロール レジスタ (SCSCR_1)	10	<ul style="list-style-type: none"> ● 転送元のデータは、 A/D 変換器の初期設定時に 内蔵 RAM に格納 ● チェーン転送使用
	A/D データ レジスタ (ADDR0~3)	内蔵 RAM	10	ブロック転送モード使用 ※ブロックサイズ: 4ワード
送信データエンティ 割り込み (TXI1)	内蔵 RAM	トランスミットデータ レジスタ (SCTDR_1)	8	転送回数は、4チャンネル分の A/D 変換結果 (計8バイト) の 連続シリアル送信回数に相当

参考プログラムは、CPU リセット後、メイン処理にてコンペアマッチタイマ、A/D 変換器、データトランスファコントローラ、およびシリアルコミュニケーションインタフェースの初期設定を行います。

A/D 変換器の初期設定では、A/D 変換開始および A/D 変換終了割り込み (ADIO) を許可するための A/D コントロールレジスタ (ADCR_0) への設定データを内蔵 RAM に格納します。データトランスファコントローラの初期設定では、使用する 3 種類の起動要因それぞれに対し転送情報の設定を行います。シリアルコミュニケーションインタフェースの初期設定では、送信データエンpty割り込み (TXI1) および送信動作を許可するためのシリアルコントロールレジスタ (SCSCR_1) への設定データを内蔵 RAM に格納します。

上記の初期設定終了後、参考プログラムはコンペアマッチタイマのカウンタ動作を開始します。カウンタ動作開始後、5 ms 周期ごとにコンペアマッチ割り込み (CMIO) 要求が発生します。この割り込み要求を最初のデータトランスファコントローラの起動要因として、内蔵 RAM に格納した A/D 変換開始および ADIO を許可するための設定データを ADCR_0 へ転送します。

上記の転送終了後、A/D 変換器が起動して 4 チャンネル分 (AN0~AN3) の A/D 変換を行います。A/D 変換終了後は ADIO 要求が発生します。この割り込み要求を 2 番目のデータトランスファコントローラの起動要因として、内蔵 RAM に格納した TXI1 および送信動作を許可するための設定データを SCSCR_1 へ転送します。このとき、さらにブロック転送モードに設定したチェイン転送を使用して、4 チャンネル分の A/D 変換結果を内蔵 RAM へ転送します。

上記の転送終了後、TXI1 要求 (計 8 回) を 3 番目のデータトランスファコントローラの起動要因として、内蔵 RAM に転送した 4 チャンネル分の A/D 変換結果 (計 8 バイト) をトランスミットデータレジスタ (SCTDR_1) へ転送し、シリアル送信出力を行います。

4 チャンネル分の A/D 変換結果のシリアル送信出力が終了すると、CPU に対し TXI1 要求が発生します。参考プログラムでは、TXI1 の割り込み処理内で、データトランスファコントローラの転送情報を再設定します。これにより、5 ms 周期で CMIO 要求が発生するたびに、参考プログラムは上記の一連の転送処理を繰り返します。

参考プログラムは、上記の一連の転送処理を、CMIO および ADIO の起動要因で使用するデータトランスファコントローラの転送回数 (10 回) 分行います。10 回分の転送処理を終了すると、参考プログラムは TXI1 の割り込み処理内でトランスミットエンド (TEND) フラグを 0 クリアし、送信終了割り込み (TEI1) 要求を許可に設定した後、TEI1 の割り込み処理内で送信動作を禁止に設定します。*

上記の処理をすべて終了すると、CPU に対し CMIO 要求が発生します。参考プログラムでは、CMIO の割り込み処理内でカウンタ動作を停止します。

図 7 および 図 8 に、参考プログラムの動作概要を示します。

【注】 * TXI を割り込み要因としてデータトランスファコントローラを起動し、SCTDR ヘデータを書き込んだ場合、TEND フラグは不定となるため、TEND フラグを送信終了フラグとして使用できません。このため、参考プログラムでは TEI を使用して送信終了を判断しています。

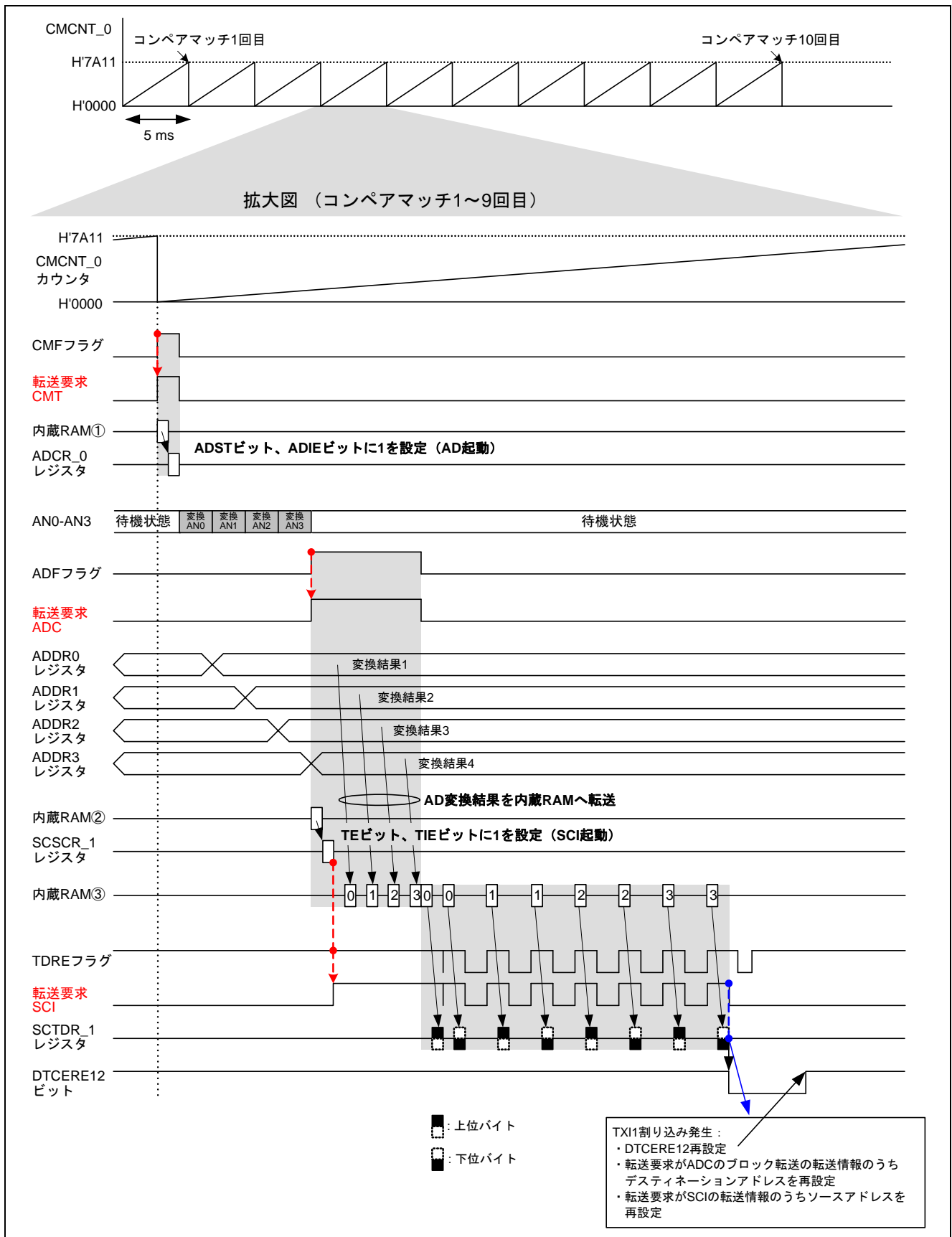


図 7 参考プログラムの動作概要 (1)

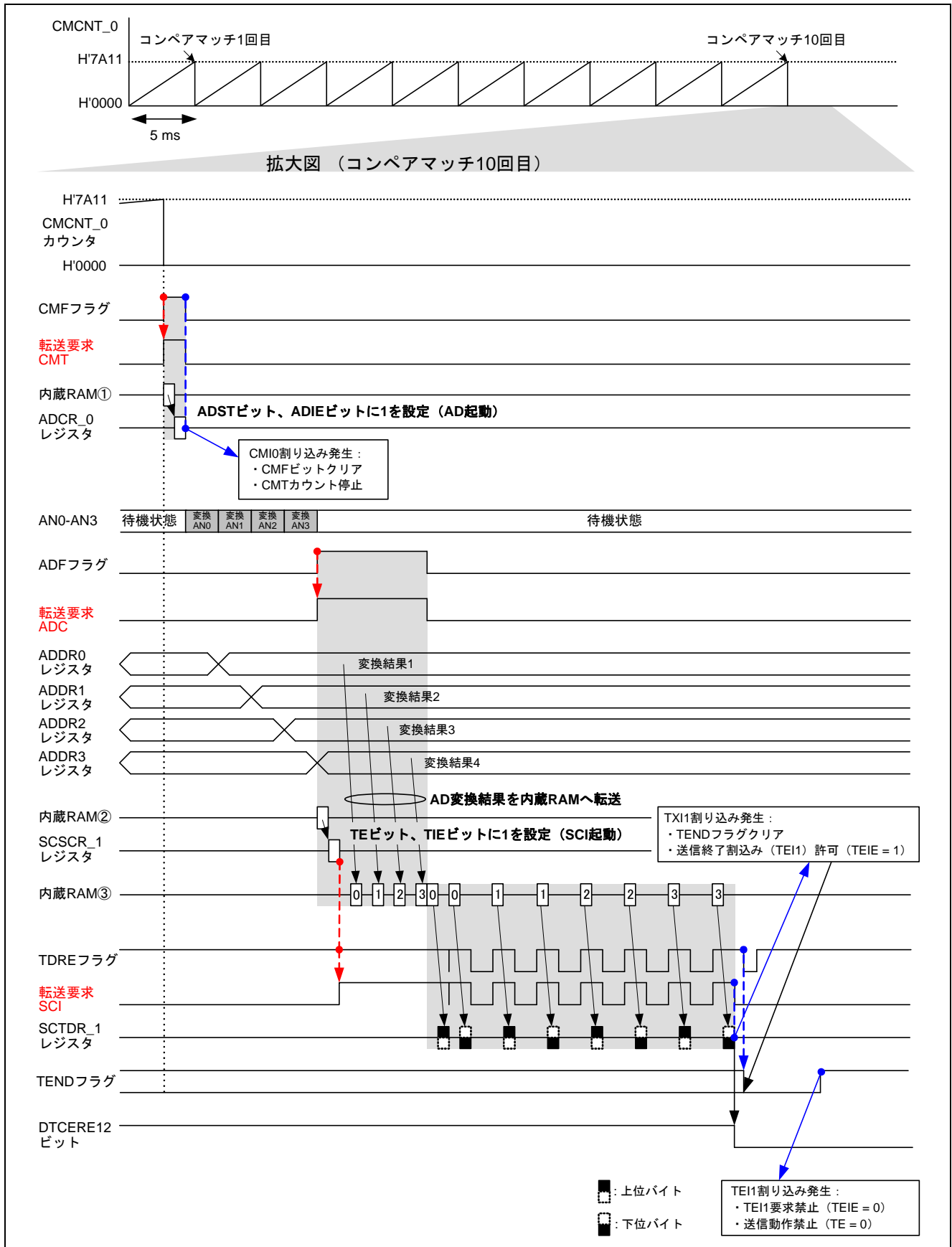


図 8 参考プログラムの動作概要 (2)

2.3.2 A/D変換器のレジスタ設定

表 5に、A/D変換器のレジスタ設定例を示します。

なお、本設定では、A/D 変換終了割り込み発生を許可していますが、データトランスファコントローラの起動時にチェーン転送を行うため、CPU に対する割り込みは発生しません。

表 5 A/D 変換器のレジスタ設定例

レジスタ名	アドレス	設定値	機能
スタンバイコントロール レジスタ 3 (STBCR3)	H'FFFE 0408	H'7A	<ul style="list-style-type: none"> • MSTP32 = "0" : A/D_0 にクロック供給
A/D コントロールレジスタ_0 (ADCR_0) *	H'FFFF E800	H'90*	<ul style="list-style-type: none"> • ADST = "1" : A/D 変換開始 • ADCS = "0" : 1 サイクルスキャンモードを選択 • ACE = "0" : ADDR レジスタのリードによる ADDR レジスタの自動クリアを禁止 • ADIE = "1" : A/D 変換終了割り込み発生を許可 • TRGE = "0" : 外部トリガまたは、マルチ ファンクションタイマパルスユニット 2 からの A/D 変換開始トリガによる A/D 変換の開始を禁止 • EXTRG = "0" : マルチファンクションタイマパルス ユニット 2 からの A/D 変換開始トリガ により A/D 変換器を起動
A/D アナログ入力チャネル 選択レジスタ_0 (ADANSR_0)	H'FFFF E820	H'0F	<ul style="list-style-type: none"> • ANS3 = "1" : • ANS2 = "1" : • ANS1 = "1" : • ANS0 = "1" : アナログ入力端子 AN0~AN3 を選択
A/D バイパスコントロール レジスタ_0 (ADBYPSR_0)	H'FFFF E830	H'01	<ul style="list-style-type: none"> • SH = "1" : 各チャネル専用のサンプル&ホールド 回路を選択

【注】 * 本応用例では初期化時に設定値を内蔵 RAM に格納し、レジスタへの設定は CMI0 を起動要因とするデータトランスファコントローラの転送により行います。

2.3.3 データトランスファコントローラのレジスタ設定

表 6～表 10に、データトランスファコントローラのレジスタ設定例を示します。

表 6 データトランスファコントローラのレジスタ設定例（共通）

レジスタ名	アドレス	設定値	機能
DTC イネーブルレジスタ A (DTCERA)	H'FFFE 6000	H'0088	<ul style="list-style-type: none"> DTCE7 = "1" : 割り込み起動要因を ADI0 に設定 DTCE3 = "1" : 割り込み起動要因を CMIO に設定
DTC イネーブルレジスタ E (DTCERE)	H'FFFE 6008	H'1000	<ul style="list-style-type: none"> DTCE12 = "1" : 割り込み起動要因を TXI1 に設定
DTC コントロールレジスタ (DTCCR)	H'FFFE 6010	H'00	<ul style="list-style-type: none"> RRS = "0" : 転送情報リードスキップを行わない RCHNE = "0" : リピート転送後のチェイン転送を禁止
DTC ベクタベースレジスタ (DTCVBR)	H'FFFE 6014	H'FFF8 5000	DTC ベクタテーブルアドレス算出時の ベースアドレス

表 7 データトランスファコントローラのレジスタ設定例（起動要因：CMIO）

レジスタ名	アドレス*	設定値	機能
DTC モードレジスタ A (MRA)	H'FFF8 4000	H'00	<ul style="list-style-type: none"> MD[1:0] = "B'00" : ノーマル転送モード Sz[1:0] = "B'00" : バイトサイズ転送 SM[1:0] = "B'00" : 転送後 SAR は固定
DTC モードレジスタ B (MRB)	H'FFF8 4001	H'00	<ul style="list-style-type: none"> CHNE = "0" : チェイン転送禁止 CHNS = "0" : 連続してチェイン転送を行う DISEL = "0" : 指定した回数分のデータ転送後、 CPU に対し割り込み要求を発生 DTS = "0" : デスティネーション側がリピート領域 またはブロック領域 DM[1:0] = "B'00" : 転送後 DAR は固定
DTC ソースアドレスレジスタ (SAR)	H'FFF8 4004	H'FFF8 4108	転送元のアドレス : A/D_0 の変換開始設定データ (H'90) を格納した内蔵 RAM 上のアドレス
DTC デスティネーション アドレスレジスタ (DAR)	H'FFF8 4008	H'FFFF E800	転送先のアドレス : A/D コントロールレジスタ_0 (ADCR_0)
DTC 転送カウントレジスタ A (CRA)	H'FFF8 400C	H'000A	転送回数 : 10 回

【注】 * 本応用例における転送情報のメモリ配置先であり、ユーザ側の設定に依存します。

表 8 データトランスファコントローラのレジスタ設定例（起動要因：ADI0）(1)

レジスタ名	アドレス*	設定値	機能
DTC モードレジスタ A (MRA)	H'FFF8 4010	H'00	<ul style="list-style-type: none"> MD[1:0] = "B'00" : ノーマル転送モード Sz[1:0] = "B'00" : バイトサイズ転送 SM[1:0] = "B'00" : 転送後 SAR は固定
DTC モードレジスタ B (MRB)	H'FFF8 4011	H'80	<ul style="list-style-type: none"> CHNE = "1" : チェイン転送許可 CHNS = "0" : 連続してチェイン転送を行う DISEL = "0" : 指定した回数分のデータ転送後、 CPU に対し割り込み要求を発生 DTS = "0" : デスティネーション側がリピート領域 またはブロック領域 DM[1:0] = "B'00" : 転送後 DAR は固定
DTC ソースアドレスレジスタ (SAR)	H'FFF8 4014	H'FFF8 410C	転送元のアドレス : TX11 の許可設定データ (H'A0) を格納した内蔵 RAM 上のアドレス
DTC デスティネーション アドレスレジスタ (DAR)	H'FFF8 4018	H'FFFF 8804	転送先のアドレス : シリアルコントロールレジスタ_1 (SCSCR_1)
DTC 転送カウントレジスタ A (CRA)	H'FFF8 401C	H'000A	転送回数 : 10 回

【注】 * 本応用例における転送情報のメモリ配置先であり、ユーザ側の設定に依存します。

表 9 データトランスファコントローラのレジスタ設定例（起動要因：ADI0）(2)

レジスタ名	アドレス*	設定値	機能
DTC モードレジスタ A (MRA)	H'FFF8 4020	H'98	<ul style="list-style-type: none"> MD[1:0] = "B'10" : ブロック転送モード Sz[1:0] = "B'01" : ワードサイズ転送 SM[1:0] = "B'10" : 転送後 SAR をインクリメント
DTC モードレジスタ B (MRB)	H'FFF8 4021	H'18	<ul style="list-style-type: none"> CHNE = "0" : チェイン転送禁止 CHNS = "0" : 連続してチェイン転送を行う DISEL = "0" : 指定した回数分のデータ転送後、 CPU に対し割り込み要求を発生 DTS = "1" : ソース側がブロック領域 DM[1:0] = "B'10" : 転送後 DAR をインクリメント
DTC ソースアドレスレジスタ (SAR)	H'FFF8 4024	H'FFFF E840	転送元のアドレス : A/D データレジスタ 0 (ADDR0)
DTC デスティネーション アドレスレジスタ (DAR)	H'FFF8 4028	H'FFF8 4102	転送先のアドレス : A/D 変換結果 (AN0~AN3) の格納先 (内蔵 RAM) の先頭アドレス
DTC 転送カウントレジスタ A (CRA)	H'FFF8 402C	H'0404	<ul style="list-style-type: none"> CRAH (上位 8 ビット) : ブロックサイズ=4 ワード CRAL (下位 8 ビット) : ブロックサイズカウンタ=4 ※初期設定時はブロックサイズと同じ 値
DTC 転送カウントレジスタ B (CRB)	H'FFF8 402E	H'000A	転送回数 : 10 回

【注】 * 本応用例における転送情報のメモリ配置先であり、ユーザ側の設定に依存します。

表 10 データトランスファコントローラのレジスタ設定例（起動要因：TXI1）

レジスタ名	アドレス*	設定値	機能
DTC モードレジスタ A (MRA)	H'FFF8 4030	H'08	<ul style="list-style-type: none"> MD[1:0] = "B'00" : ノーマル転送モード Sz[1:0] = "B'00" : バイトサイズ転送 SM[1:0] = "B'10" : 転送後 SAR をインクリメント
DTC モードレジスタ B (MRB)	H'FFF8 4031	H'00	<ul style="list-style-type: none"> CHNE = "0" : チェイン転送禁止 CHNS = "0" : 連続してチェイン転送を行う DISEL = "0" : 指定した回数分のデータ転送後、 CPU に対し割り込み要求を発生 DTS = "0" : デスティネーション側がリピート領域 またはブロック領域 DM[1:0] = "B'00" : 転送後 DAR は固定
DTC ソースアドレスレジスタ (SAR)	H'FFF8 4034	H'FFF8 4102	転送元のアドレス : アナログ入力チャネル AN0~AN3 の A/D 変換結果を格納した内蔵 RAM 上の 先頭アドレス
DTC デスティネーション アドレスレジスタ (DAR)	H'FFF8 4038	H'FFFF 8806	転送先のアドレス : トランスミットデータレジスタ_1 (SCTDR_1)
DTC 転送カウンタレジスタ A (CRA)	H'FFF8 403C	H'0008	転送回数 : 8 回

【注】 * 本応用例における転送情報のメモリ配置先であり、ユーザ側の設定に依存します。

2.3.4 シリアルコミュニケーションインタフェースのレジスタ設定

表 11に、シリアルコミュニケーションインタフェースのレジスタ設定例を示します。

表 11 シリアルコミュニケーションインタフェースのレジスタ設定例

レジスタ名	アドレス	設定値	機能
スタンバイコントロール レジスタ 5 (STBCR5)	H'FFFE 0418	H'BF	<ul style="list-style-type: none"> MSTP56 = "0" : SCI1 ヘクロック供給
割り込み優先レベル設定 レジスタ (IPR16)	H'FFFE 0C14	H'0F00	SCI1 割り込み優先レベル : 15
ポート A コントロール レジスタ L2 (PACRL2)	H'FFFE 3814	H'0006	<ul style="list-style-type: none"> PA4MD[2:0] = " B'110" : PA4 端子を TXD1 出力に設定
シリアルモードレジスタ_1 (SCSMR_1)	H'FFFF 8800	H'00	<ul style="list-style-type: none"> C/A = "0" : 調歩同期式モード CHR = "0" : 8 ビットデータ PE = "0" : パリティビットの付加、およびチェック を禁止 STOP = "0" : 1 ストップビット MP = "0" : マルチプロセッサモードを禁止 CKS[1:0] = "B'00" : Pφクロック
ビットレートレジスタ_1 (SCBRR_1)	H'FFFF 8802	H'1A	設定ビットレート : 57870.37 bps (想定ビットレートは 57600 bps)
シリアルコントロール レジスタ_1 (SCSCR_1) *	H'FFFF 8804	H'A0*	<ul style="list-style-type: none"> TIE = "1" : 送信データエンプティ割り込み (TXI) 要求を許可 TE = "1" : 送信動作を許可 CKE[1:0] = "B'00" : 内部クロック/SCK 端子は入力端子 (入力信号は無視)

【注】 * 本応用例では初期化時に設定値を内蔵 RAM に格納し、レジスタへの設定は ADIO を起動要因とするデータトランスファコントローラの転送により行います。

2.3.5 参考プログラムの動作フロー

図 9～図 14に、参考プログラムの動作フローを示します。

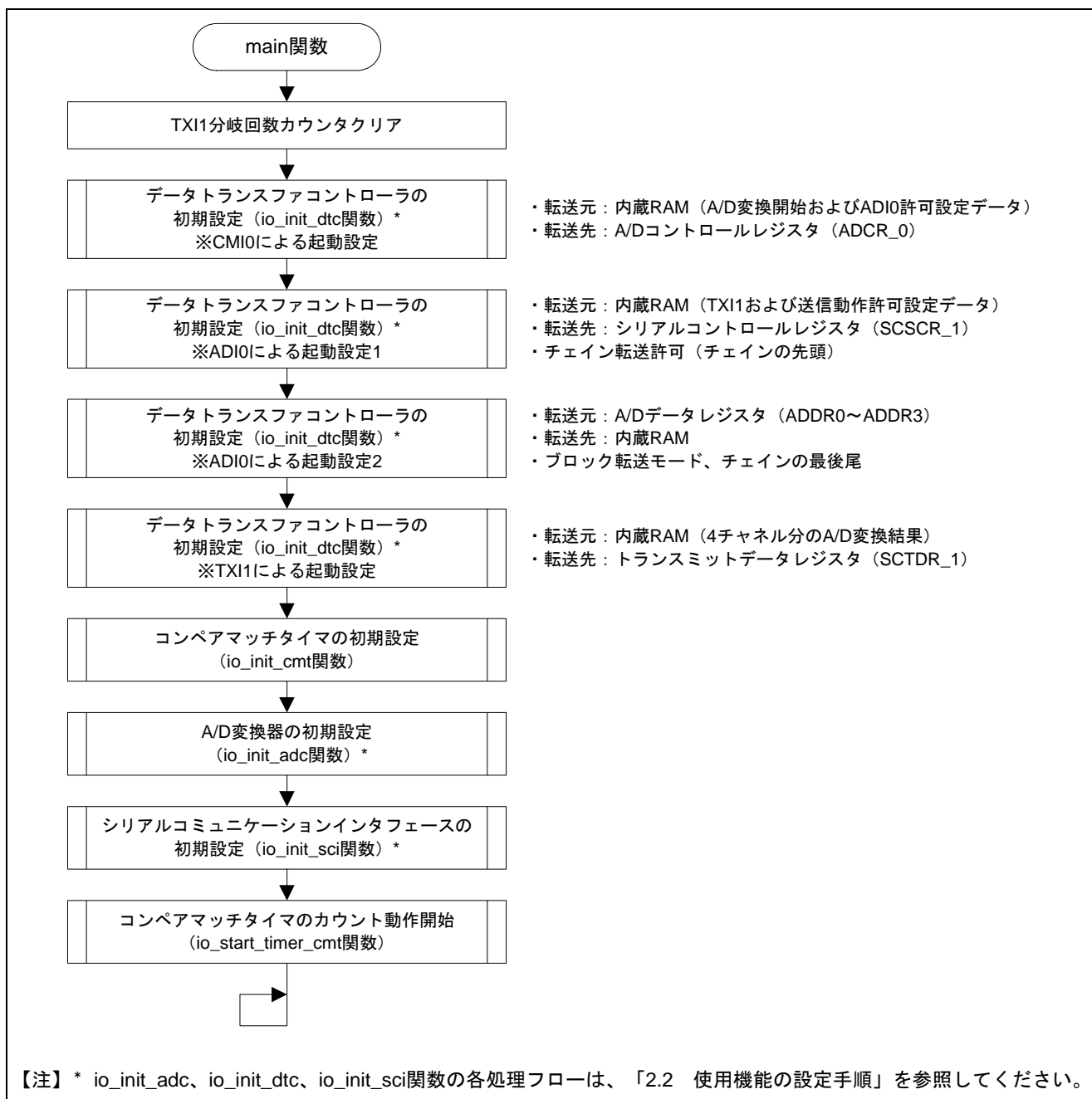


図 9 メイン処理フロー

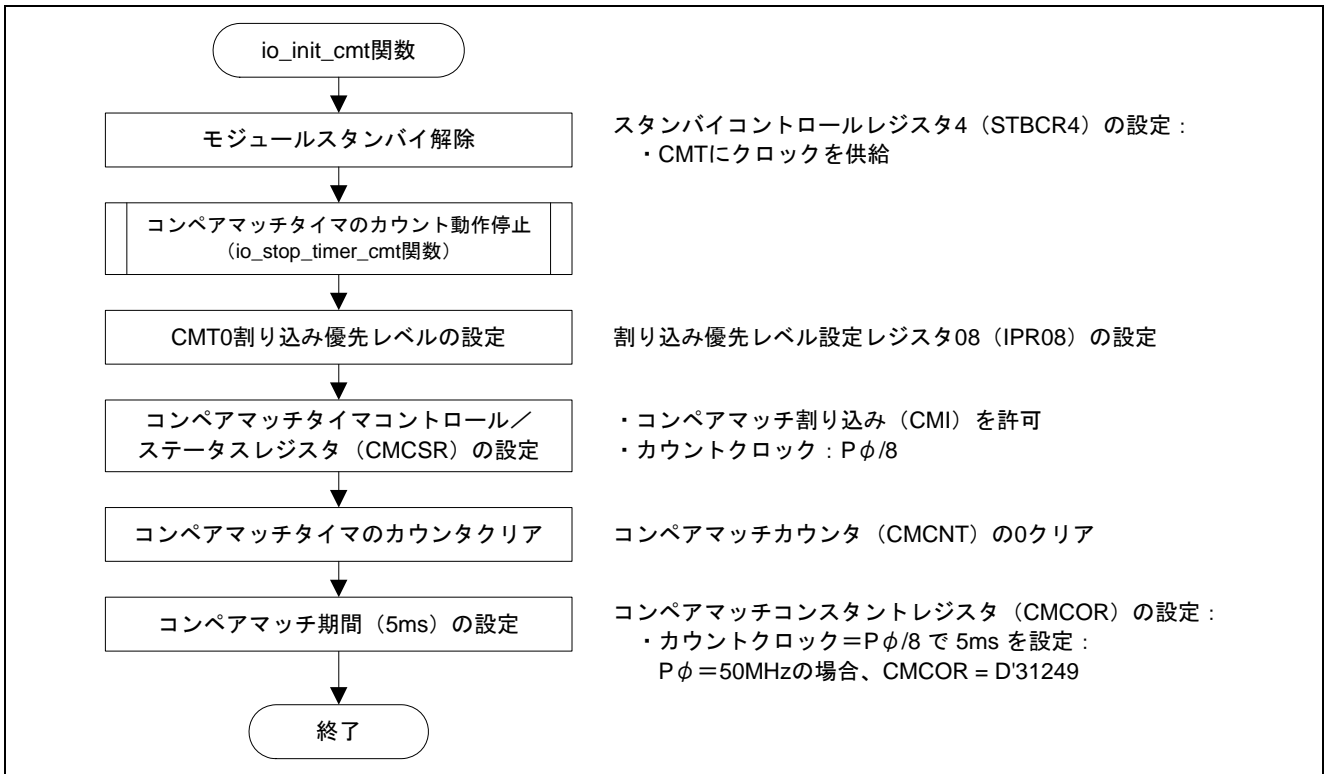


図 10 コンペアマッチタイマの初期設定フロー

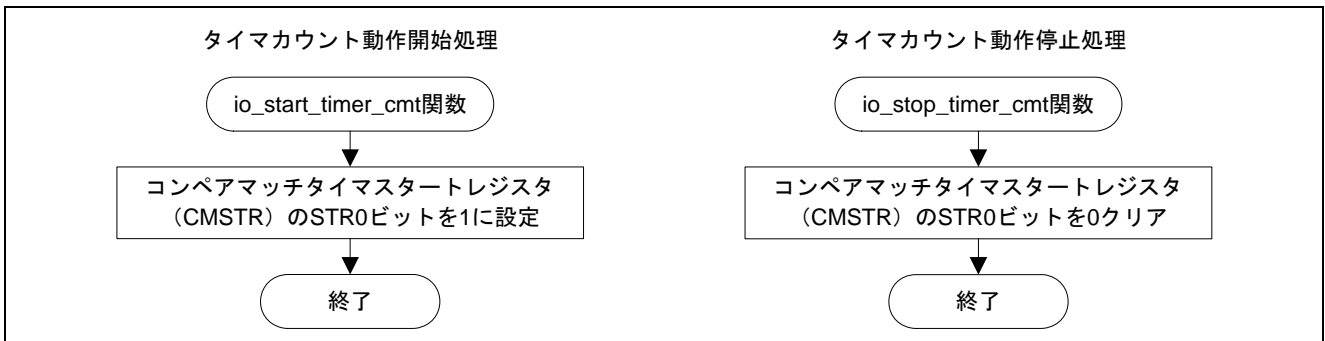


図 11 コンペアマッチタイマのカウンタ動作制御処理フロー

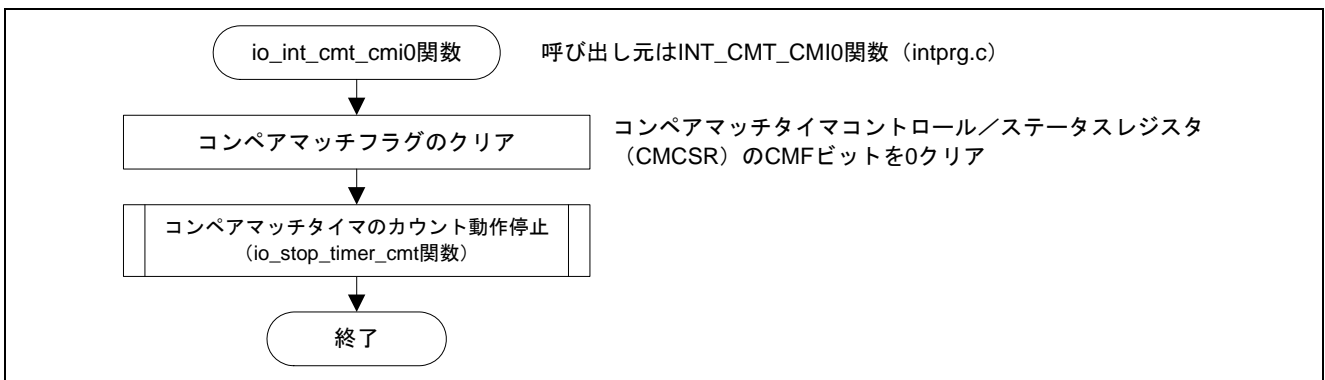


図 12 コンペアマッチ割り込み (CMI0) 処理フロー

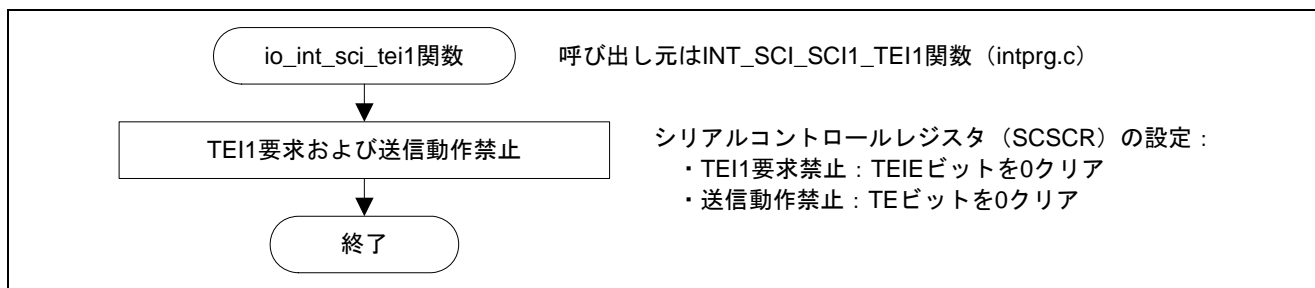


図 14 送信終了割り込み (TEI1) 処理フロー

3. 参考プログラムリスト

3.1 サンプルプログラムリスト "main.c" (1)

```
1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corp. and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corp. and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 *   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 /*"FILE COMMENT"***** Technical reference data *****
31 *   System Name : SH7216 Sample Program
32 *   File Name   : main.c
33 *   Abstract    : ADC+DTC+SCI Module Application
34 *   Version     : 1.00.00
35 *   Device      : SH7216
36 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37 *                : C/C++ compiler package for the SuperH RISC engine family
38 *                :                               (Ver.9.03 Release00).
39 *   OS          : None
40 *   H/W Platform: R0K572167 (CPU board)
41 *   Description :
42 *****/
43 *   History     : Jul.30,2010 Ver.1.00.00
44 *"FILE COMMENT END"*****
45 #include <machine.h>
46 #include "iodefine.h"
47 #include "dtc.h"
48
```

3.2 サンプルプログラムリスト "main.c" (2)

```
49  /* ==== Prototype declaration ==== */
50  void main(void);
51  void io_init_adc(void);
52  void io_init_dtc(DTC_TRANS_INFO *dtc_info, unsigned long src,
53                unsigned long dst, unsigned long count, unsigned long mode);
54  void io_init_sci(void);
55  void io_init_cmt(void);
56  void io_start_timer_cmt(void);
57  void io_stop_timer_cmt(void);
58  void io_int_cmt_cmi0(void);
59  void io_int_sci_txil(void);
60  void io_int_sci_teil(void);
61
62  /* ==== Global variable ==== */
63  int gCnt_TXI; /* TXI1 分岐回数カウンタ */
64  unsigned long *gData_DTCvect = (unsigned long *)DTC_VECT_BASE; /* DTC ベクタテーブル */
65  /* ---- DTC 転送用データ格納先(URAM) ---- */
66  #pragma section DTCDATA
67  unsigned char gData_ADstart; /* A/D 変換開始データ */
68  unsigned char gData_TXIenable; /* TXI 許可データ */
69  unsigned short gData_ADconv[4]; /* A/D 変換データ(偶数番地に配置) */
70  #pragma section
71
72  /*"FUNC COMMENT"*****
73  * ID          :
74  * Outline     : サンプルプログラムメイン
75  *-----
76  * Include     : "dtc.h"
77  *-----
78  * Declaration : void main(void);
79  *-----
80  * Description : サンプルプログラムメイン
81  *-----
82  * Argument    : void
83  *-----
84  * Return Value : void
85  *-----
86  * Note        : None
87  *"FUNC COMMENT END"*****/
88  void main(void)
89  {
90    /* ==== TXI1 分岐回数カウンタクリア ==== */
91    gCnt_TXI = 0;
92
```

3.3 サンプルプログラムリスト "main.c" (3)

```
93      /* ==== DTC 設定 ==== */
94      /* ---- CMI0 による起動設定 ---- */
95      io_init_dtc(&DTC_CMI0,          /* 転送情報格納領域の参照先アドレス */
96                (unsigned long)&gData_ADstart,    /* 転送元: URAM */
97                (unsigned long)&ADC0.ADCR,        /* 転送先: ADCR */
98                DTC_COUNT_LINK,                /* DTC 連動回数 */
99                (DTC_TRG_CMI0 | DTC_MODE_NORMAL | DTC_SIZE_BYTE)
100             /* 起動要因 CMI0、ノーマル転送モード、バイトサイズ転送 */
101             );
102      /* ---- ADI0 による起動設定 (チェーン転送) ---- */
103      io_init_dtc(&DTC_ADI0_CHN1,      /* 転送情報格納領域の参照先アドレス */
104                (unsigned long)&gData_TXIenable,  /* 転送元: URAM */
105                (unsigned long)&SCI1.SCSCR,        /* 転送先: SCSCR */
106                DTC_COUNT_LINK,                /* DTC 連動回数 */
107                (DTC_TRG_ADI0 | DTC_MODE_NORMAL | DTC_SIZE_BYTE | DTC_CHN_INITIAL)
108             /* 起動要因 ADI0、ノーマル転送モード、バイトサイズ転送、 */
109             /* チェイン転送使用 (チェーンの先頭) */
110             );
111      io_init_dtc(&DTC_ADI0_CHN2,      /* 転送情報格納領域の参照先アドレス */
112                (unsigned long)&ADC0.ADDR0,        /* 転送元: ADDR0-ADDR3 */
113                (unsigned long)&gData_ADconv[0],   /* 転送先: URAM */
114                (DTC_COUNT_LINK | DTC_COUNT_BLKSIZE), /* DTC 連動回数、ブロックサイズ指定 */
115                (DTC_TRG_ADI0 | DTC_MODE_BLOCK | DTC_SIZE_WORD | DTC_SRC_UP | DTC_DST_UP |
116                 DTC_TARGET_SRC | DTC_CHN_FINAL)
117             /* 起動要因 ADI0、ブロック転送モード、ワードサイズ転送、 */
118             /* 転送後 SAR, DAR とともにインクリメント、ソース側がブロック領域、 */
119             /* チェイン転送使用 (チェーンの最後尾) */
120             );
121      /* ---- TXI1 による起動設定 ---- */
122      io_init_dtc(&DTC_TXI1,          /* 転送情報格納領域の参照先アドレス */
123                (unsigned long)&gData_ADconv[0],  /* 転送元: URAM */
124                (unsigned long)&SCI1.SCTDR,        /* 転送先: SCTDR */
125                DTC_COUNT_SCI,                /* シリアルデータ転送回数 */
126                (DTC_TRG_TXI1 | DTC_MODE_NORMAL | DTC_SIZE_BYTE | DTC_SRC_UP)
127             /* 起動要因 TXI1、ノーマル転送モード、バイトサイズ転送、 */
128             /* 転送後 SAR をインクリメント */
129             );
130
131      /* ==== CMT 設定 ==== */
132      io_init_cmt();
133
134      /* ==== ADC 設定 ==== */
135      io_init_adc();
136
137      /* ==== SCI 設定 ==== */
138      io_init_sci();
139
```

3.4 サンプルプログラムリスト "main.c" (4)

```
140     /* ==== CMT タイマカウンタ動作開始 ==== */
141     io_start_timer_cmt();
142
143     while(1){
144         /* Dead loop */
145     }
146 }
147
148 /*"FUNC COMMENT"*****
149 * ID          :
150 * Outline     : ADC 設定
151 *-----
152 * Include     : "iodefine.h"
153 *-----
154 * Declaration : void io_init_adc(void);
155 *-----
156 * Description : ADC0 の初期化処理を行います。
157 *              : ・動作モード : 1 サイクルスキャン
158 *              : ・アナログ入力チャネル : AN0, AN1, AN2, AN3
159 *              : ・変換回路 : サンプル&ホールド回路
160 *              :
161 *              : サンプルプログラムでは、ADCR への設定 (A/D 変換開始および
162 *              : ADI 要求許可) は DTC 転送により行います。
163 *              : このため本関数にて転送元の設定データを URAM に格納しています。
164 *-----
165 * Argument    : void
166 *-----
167 * Return Value : void
168 *-----
169 * Note        : None
170 *"FUNC COMMENT END"*****/
171 void io_init_adc(void)
172 {
173     /* ==== モジュールスタンバイ解除 ==== */
174     STB.CR3.BIT._ADC0 = 0;          /* ADC0 にクロックを供給 */
175
176     /* ==== A/D アナログ入力チャネル選択レジスタ (ADANSR) 設定 ==== */
177     ADC0.ADANSR.BYTE = 0x0f;        /* アナログ入力チャネル AN0~AN3 を選択 */
178
179     /* ==== A/D バイパスコントロールレジスタ (ADBYPSCR) 設定 ==== */
180     ADC0.ADBYPSCR.BIT.SH = 1;       /* サンプル&ホールド回路を選択 */
181
```

3.5 サンプルプログラムリスト "main.c" (5)

```

182     /* ==== A/D コントロールレジスタ (ADCR) に転送するための設定データを URAM に格納 ==== */
183     gData_ADstart = 0x90;
184     /*
185         bit7 : ADST = 1 ---- A/D 変換開始
186         bit6 : ADCS = 0 ---- 1 サイクルスキャン
187         bit5 : ACE = 0 ----- ADDR のリードによる ADDR の自動クリアを禁止
188         bit4 : ADIE = 1 ---- A/D 変換終了割り込み (ADI) 要求の許可
189         bit3-2: Reserved(0)
190         bit1 : TRGE = 0 ---- 外部トリガまたは、MTU2/MTU2S からの A/D 変換
191                 :                               開始トリガによる A/D 変換の開始を禁止
192         bit0 : EXTRG = 0 --- MTU2/MTU2S からの A/D 変換開始トリガにより
193                 :                               A/D 変換器を起動
194     */
195 }
196
197 /*"FUNC COMMENT"*****
198 * ID          :
199 * Outline     : DTC 設定
200 *-----
201 * Include     : "iodef.h" and "dtc.h"
202 *-----
203 * Declaration : void io_init_dtc(DTC_TRANS_INFO *dtc_info,
204 *                               :                               unsigned long src,
205 *                               :                               unsigned long dst,
206 *                               :                               unsigned long count,
207 *                               :                               unsigned long mode);
208 *-----
209 * Description : DTC の初期化処理を行います。
210 *             : 引数 src,dst,count,mode に指定した転送情報を、引数 *dtc_info
211 *             : に指定した DTC 転送情報レジスタ (MRA,MRB,SAR,DAR,CRA,CRB) 配置先へ
212 *             : 設定します。
213 *             : チェイン転送不使用時、またはチェイン転送使用でチェインの先頭
214 *             : データの転送情報設定時は、引数 *dtc_info に指定した転送情報
215 *             : 配置先の参照アドレスを、使用する割り込み起動要因の DTC ベクタ
216 *             : テーブルに登録します。
217 *-----
218 * Argument    : ・DTC_TRANS_INFO *dtc_info ; I/O : DTC 転送情報レジスタ配置先へ
219 *             :                               : の参照アドレス
220 *             : ・unsigned long src      ; I   : SAR に設定するアドレス
221 *             : ・unsigned long dst      ; I   : DAR に設定するアドレス
222 *             : ・unsigned long count    ; I   : 転送回数
223 *             : ・unsigned long mode     ; I   :
224 *             :   転送モード/転送サイズ/起動要因トリガ等の DTC 設定情報を
225 *             :   サンプルプログラムのヘッダファイル "dtc.h" にてマクロ定義
226 *             :   しているパラメータで指定します。
227 *             :   パラメータを複数指定する場合は、OR 演算子 "|" で区切って
228 *             :   指定してください。
229 *-----
230 * Return Value : void
231 *-----

```

3.6 サンプルプログラムリスト "main.c" (6)

```

232 * Note      : ・本関数を実行する際、使用する起動要因モジュールの割り込みは
233 *           :   禁止にしてください。
234 *           : ・引数 *dtc_info に指定する転送情報レジスタ配置先の
235 *           :   参照アドレスは、4 バイトアライメントにしてください。
236 *           : ・ブロック転送モードを使用する場合は、引数 count の
237 *           :   bit23～bit16 にブロックサイズを指定し、さらに
238 *           :   下位 16 ビット(bit15～bit0)に転送回数を指定してください。
239 * "FUNC COMMENT END"*****
240 void io_init_dtc(DTC_TRANS_INFO *dtc_info, unsigned long src,
241                unsigned long dst, unsigned long count, unsigned long mode)
242 {
243     unsigned long chain_info;
244     unsigned char set_mra, set_mrb, set_dtccr, f_rrs;
245
246     /* ---- DTC 設定情報を第 5 引数より抽出 ---- */
247     set_mra = (unsigned char)((mode >> 8) & 0x000000fc); /* MRA 設定情報 */
248     set_mrb = (unsigned char)(mode & 0x000000fc); /* MRB 設定情報 */
249     set_dtccr = (unsigned char)((mode >> 24) & 0x00000018); /* DTCCR 設定情報 */
250     chain_info = mode & 0x00ffc0c3; /* チェイン転送情報 */
251
252     /* **** 転送情報設定 **** */
253     /* ==== DTC コントロールレジスタ (DTCCR)設定 ==== */
254     /* ---- 転送情報リードスキップイネーブル情報回避(チェイン転送使用時の措置) ---- */
255     f_rrs = DTC.DTCCR.BIT.RRS;
256     /* ---- 転送情報リードスキップ禁止 ---- */
257     DTC.DTCCR.BIT.RRS = 0;
258     /* ==== DTC モードレジスタ A (MRA)設定 ==== */
259     dtc_info->MRA = set_mra;
260     /* ==== DTC モードレジスタ B (MRB)設定 ==== */
261     dtc_info->MRB = set_mrb;
262     /* ==== DTC ソースアドレスレジスタ (SAR)設定 ==== */
263     dtc_info->SAR = src;
264     /* ==== DTC デスティネーションアドレスレジスタ (DAR)設定 ==== */
265     dtc_info->DAR = dst;
266     /* ==== DTC 転送カウントレジスタ A,B (CRA,CRB)設定 ==== */
267     if((chain_info & DTC_MODE_REPEAT) != 0){
268         /* ---- リピート転送モード時 ---- */
269         dtc_info->CRA.BYTE.H = (unsigned char)(count & 0x000000ff); /* 転送回数 */
270         dtc_info->CRA.BYTE.L = dtc_info->CRA.BYTE.H; /* 転送カウンタ */
271     }
272     else if((chain_info & DTC_MODE_BLOCK) != 0){
273         /* ---- ブロック転送モード時 ---- */
274         dtc_info->CRA.BYTE.H = (unsigned char)((count & 0x00ff0000) >> 16); /* ブロックサイズ */
275         dtc_info->CRA.BYTE.L = dtc_info->CRA.BYTE.H; /* ブロックサイズカウンタ */
276         dtc_info->CRB = (unsigned short)(count & 0x0000ffff); /* 転送回数 */
277     }
278     else{
279         /* ---- ノーマル転送モード時 ---- */
280         dtc_info->CRA.WORD = (unsigned short)(count & 0x0000ffff); /* 転送回数 */
281     }
282

```


3.7 サンプルプログラムリスト "main.c" (7)

```
283     if(((chain_info & 0x00000083) == DTC_CHN_DISABLE) ||
284        ((chain_info & 0x00000083) == DTC_CHN_INITIAL)){
285         /* ---- チェイン転送不使用時、または
286            チェイン転送使用でチェインの先頭データの転送情報設定時 ---- */
287
288         /* **** DTC コントロールレジスタ (DTCCR)設定 **** */
289         DTC.DTCCR.BYTE = set_dtccr;
290         /*
291            bit7-5: Reserved(0)
292            bit4  : RRS = x ----- 転送情報リードスキップを許可/禁止
293                   :                (デフォルト禁止)
294            bit3  : RCHNE = x --- リポート転送後のチェイン転送を許可/禁止
295                   :                (デフォルト禁止)
296            bit2-1: Reserved(0)
297            bit0  : ERR = 0 ----- 転送停止フラグ
298         */
299
300         /* **** DTC ベクタベースレジスタ (DTCVBR)設定 **** */
301         DTC.DTCVBR = DTC_VECT_BASE;
302
303         /* **** DTC ベクタアドレス/起動要因設定 **** */
304         switch(chain_info & 0x00ff0000){
305         case 0x00010000: /* DTC ベクタアドレス設定: CMI0 */
306             gData_DTCvect[DTC_VECT_CMI0/sizeof(unsigned long)] = (unsigned long)dtc_info;
307             DTC.DTCERA.BIT.CMI0 = 1; /* DTC 起動イネーブル: CMI0 */
308             break;
309         case 0x00020000: /* DTC ベクタアドレス設定: ADI0 */
310             gData_DTCvect[DTC_VECT_ADI0/sizeof(unsigned long)] = (unsigned long)dtc_info;
311             DTC.DTCERA.BIT.ADI0 = 1; /* DTC 起動イネーブル: ADI0 */
312             break;
313         case 0x00030000: /* DTC ベクタアドレス設定: TXI1 */
314             gData_DTCvect[DTC_VECT_TXI1/sizeof(unsigned long)] = (unsigned long)dtc_info;
315             DTC.DTCERE.BIT.TXI1 = 1; /* DTC 起動イネーブル: TXI1 */
316             break;
317         default:
318             break;
319         }
320     }
321     else{
322         /* ---- チェイン転送使用でチェインの中間または最後尾データの
323            転送情報設定時 ---- */
324         DTC.DTCCR.BIT.RRS = f_rrs; /* 転送情報リードスキップイネーブル情報差し戻し */
325     }
326 }
327
```

3.8 サンプルプログラムリスト "main.c" (8)

```
328 /*"FUNC COMMENT"*****
329 * ID      :
330 * Outline : SCI 設定
331 *-----
332 * Include : "iodefine.h"
333 *-----
334 * Declaration : void io_init_sci(void);
335 *-----
336 * Description : SCI1 の初期化処理を行います。
337 *             : ・調歩同期モード
338 *             : ・キャラクタ長:8 ビットデータ
339 *             : ・パリティ無し
340 *             : ・ストップビット長:1 ストップビット
341 *             : ・ビットレート:57600bps
342 *             :
343 *             : ピンファンクションは PA4 を TXD1(出力)に設定しています。
344 *             :
345 *             : サンプルプログラムでは、SCSCR への設定(TXI 要求許可および
346 *             : 送信動作許可)は DTC 転送により行います。
347 *             : このため本関数にて転送元の設定データを URAM に格納しています。
348 *-----
349 * Argument : void
350 *-----
351 * Return Value : void
352 *-----
353 * Note      : None
354 *"FUNC COMMENT END"*****/
355 void io_init_sci(void)
356 {
357     volatile int CntWait_1bit = 600; /* 1ビット期間ウェイト用カウンタ@200MHz */
358
359     /* ==== モジュールスタンバイ解除 ==== */
360     STB.CR5.BIT._SCI1 = 0; /* SCI1 にクロックを供給 */
361
362     /* ==== SCI1 割り込み優先レベル設定 ==== */
363     INTC.IPR16.BIT._SCI1 = 0xf; /* Level=15 */
364
365     /* ==== シリアルコントロールレジスタ(SCSCR)設定 ==== */
366     /* ---- 割り込み要求禁止/送受信動作禁止 ---- */
367     SCI1.SCSCR.BYTE = 0x00;
368     /* ---- クロックソース/クロック入出力設定 ---- */
369     SCI1.SCSCR.BIT.CKE = 0; /* 内部クロック/SCK 端子は入力端子(入力信号は無視) */
370
```

3.9 サンプルプログラムリスト "main.c" (9)

```
371      /* ==== シリアルモードレジスタ (SCSMR)設定 ==== */
372      SCI1.SCSMR.BYTE = 0x00;
373      /*
374          bit7 : C/A# = 0 ----- 調歩同期式モード
375          bit6 : CHR = 0 ----- キャラクタ長: 8 ビットデータ
376          bit5 : PE = 0 ----- パリティビットの付加/チェックを禁止
377          bit4 : O/E# = 0 ----- 不使用
378          bit3 : STOP = 0 ----- ストップビット長: 1 ストップビット
379          bit2 : MP = 0 ----- マルチプロセッサモードを禁止
380          bit1-0: CKS[1:0] = B'00 --- ボーレートジェネレータの動作クロック: Pφ
381      */
382
383      /* ==== ビットレートレジスタ (SCBRR)設定 ==== */
384      SCI1.SCBRR = 0x1a;          /* 設定ビットレート=57870.37bps@Pφ=50MHz */
385                                /* (想定ビットレート=57600bps) */
386
387      /* ==== 1ビット期間ウェイト ==== */
388      while(CntWait_1bit-- > 0){
389      }
390
391      /* ==== ピンファンクション設定 ==== */
392      PFC.PACRL2.BIT.PA4MD = 6;  /* PA4 端子機能を TXD1(出力)に設定 */
393
394      /* ==== SCSCR に設定(転送)するためのデータ値を URAM に格納 ==== */
395      gData_TXIenable = 0xa0;
396      /*
397          bit7 : TIE = 1 ----- TXI 要求を許可
398          bit6 : RIE = 0 ----- RXI, ERI 要求を禁止
399          bit5 : TE = 1 ----- 送信動作を許可
400          bit4 : RE = 0 ----- 受信動作を禁止
401          bit3 : MPiE = 0 ----- マルチプロセッサ機能ディスエーブル
402          bit2 : TEiE = 0 ----- TEI 要求を禁止
403          bit1-0: CKE[1:0] = B'00 --- 内部クロック/SCK 端子は入力端子
404              :                    (入力信号は無視)
405      */
406  }
407
```

3.10 サンプルプログラムリスト "main.c" (10)

```
408  /*"FUNC COMMENT"*****
409  * ID      :
410  * Outline : CMT 設定
411  *-----
412  * Include : "iodefine.h"
413  *-----
414  * Declaration : void io_init_cmt(void);
415  *-----
416  * Description : CMT0 を 5msec の定周期タイマとして設定します。
417  *-----
418  * Argument   : void
419  *-----
420  * Return Value : void
421  *-----
422  * Note       : None
423  *"FUNC COMMENT END"*****/
424 void io_init_cmt(void)
425 {
426     /* ==== モジュールスタンバイ解除 ==== */
427     STB.CR4.BIT._CMT = 0;          /* CMT にクロックを供給 */
428     /* ==== CMT カウント動作停止 ==== */
429     io_stop_timer_cmt();
430     /* ==== CMT0 割り込み優先レベル設定 ==== */
431     INTC.IPR08.BIT._CMT0 = 0xf;   /* Level=15 */
432     /* ==== CMT コントロール/ステータスレジスタ (CMCSR) 設定 ==== */
433     CMT0.CMCSR.WORD = 0x0040;     /* CMI 要求許可、カウントクロック : Pφ/8 */
434     /* ==== CMT カウンタクリア ==== */
435     CMT0.CMCNT = 0x0000;
436     /* ==== コンペアマッチ期間の設定 ==== */
437     CMT0.CMCOR = 31250 - 1;       /* 5msec@Pφ/8 (Pφ=50MHz) */
438 }
439
```

3.11 サンプルプログラムリスト "main.c" (11)

```
440  /*"FUNC COMMENT"*****
441  * ID      :
442  * Outline  : CMT タイマカウント動作開始
443  *-----
444  * Include  : "iodefine.h"
445  *-----
446  * Declaration : void io_start_timer_cmt(void);
447  *-----
448  * Description : CMT のタイマカウント動作を開始します。
449  *-----
450  * Argument   : void
451  *-----
452  * Return Value : void
453  *-----
454  * Note       : None
455  *"FUNC COMMENT END"*****/
456 void io_start_timer_cmt(void)
457 {
458     CMT.CMSTR.BIT.STR0 = 1;    /* CMCNT_0 カウント動作開始 */
459 }
460
461 /*"FUNC COMMENT"*****
462 * ID      :
463 * Outline  : CMT タイマカウント動作停止
464 *-----
465 * Include  : "iodefine.h"
466 *-----
467 * Declaration : void io_stop_timer_cmt(void);
468 *-----
469 * Description : CMT のタイマカウント動作を停止します。
470 *-----
471 * Argument   : void
472 *-----
473 * Return Value : void
474 *-----
475 * Note       : None
476 *"FUNC COMMENT END"*****/
477 void io_stop_timer_cmt(void)
478 {
479     CMT.CMSTR.BIT.STR0 = 0;    /* CMCNT_0 カウント動作停止 */
480 }
481
```

3.12 サンプルプログラムリスト "main.c" (12)

```
482  /*"FUNC COMMENT"*****
483  * ID      :
484  * Outline : CMT コンペアマッチ割り込み(CMI0)処理
485  *-----
486  * Include : "iodefine.h"
487  *-----
488  * Declaration : void io_int_cmt_cmi0(void);
489  *-----
490  * Description : コンペアマッチフラグをクリア後、CMT カウント動作を停止します。
491  *-----
492  * Argument   : void
493  *-----
494  * Return Value : void
495  *-----
496  * Note       : None
497  *"FUNC COMMENT END"*****/
498 void io_int_cmt_cmi0(void)
499 {
500     unsigned short dummy;
501
502     /* ==== コンペアマッチフラグクリア ==== */
503     CMT0.CMCSR.BIT.CMF &= 0;
504     /* ==== CMT カウント動作停止 ==== */
505     io_stop_timer_cmt();
506
507     dummy = CMT0.CMCSR.WORD; /* dummy read */
508 }
509
```

3.13 サンプルプログラムリスト "main.c" (13)

```
510  /*"FUNC COMMENT"*****
511  * ID      :
512  * Outline : SCI 送信データエンプティ割り込み(TXI1)処理
513  *-----
514  * Include : "iodef.h"
515  *-----
516  * Declaration : void io_int_sci_txil(void);
517  *-----
518  * Description : TXI1 要求を禁止に設定し、TDRE フラグをクリアした後、
519  *              : TXI1 への分岐回数別に以下の処理を行います。
520  *              :
521  *              : ・TXI1 への分岐回数が、DTC 初期設定時に指定した連動回数より
522  *              :   少ない場合、以下の転送情報を初期化した後、起動要因 TXI1 を
523  *              :   イネーブルに設定します。
524  *              :   - 起動要因 ADI0 におけるブロック転送用の転送先アドレス(DAR)
525  *              :   - 起動要因 TXI1 における転送元アドレス(SAR)および
526  *              :     シリアルデータ転送回数(CRA)
527  *              :
528  *              : ・TXI1 への分岐回数が、DTC 初期設定時に指定した連動回数に
529  *              :   達した場合、TEND フラグをクリア(TDRE ビットを1リード後に
530  *              :   0 ライト)した後に、SCI 送信終了割り込み(TEI1)要求を許可に設定
531  *              :   します。
532  *-----
533  * Argument : void
534  *-----
535  * Return Value : void
536  *-----
537  * Note      : None
538  *"FUNC COMMENT END"*****/
539 void io_int_sci_txil(void)
540 {
541     /* ==== TXI1 分岐回数カウンタインクリメント ==== */
542     gCnt_TXI++;
543
544     /* ==== TXI1 要求禁止 ==== */
545     SCI1.SCSCR.BIT.TIE = 0;
546
547     /* ==== TDRE フラグクリア ==== */
548     SCI1.SCSSR.BIT.TDRE = 0;
549
```

3.14 サンプルプログラムリスト "main.c" (14)

```
550     if(gCnt_TXI < DTC_COUNT_LINK){
551         /* ==== DTC 転送情報/起動要因再設定 ==== */
552         /* ---- 起動要因 ADI0 ---- */
553         DTC_ADI0_CHN2.DAR = (unsigned long)&gData_ADconv[0]; /* ブロック転送用 DAR */
554         /* ---- 起動要因 TXI1 ---- */
555         DTC_TXI1.SAR = (unsigned long)&gData_ADconv[0];      /* SAR */
556         DTC_TXI1.CRA.WORD = DTC_COUNT_SCI;      /* シリアルデータ転送回数 */
557         DTC.DTCERE.BIT.TXI1 = 1; /* DTC 起動イネーブル: TXI1 */
558     }
559     else{
560         /* ==== TXI1 分岐回数カウンタクリア ==== */
561         gCnt_TXI = 0;
562         /* ==== TEND フラグクリア ==== */
563         SCI1.SCSSR.BIT.TDRE = 0; /* TDRE ビットを1リード後に0ライト */
564         /* ==== TEI1 要求許可 ==== */
565         SCI1.SCSCR.BIT.TEIE = 1;
566     }
567 }
568
569 /*"FUNC COMMENT"*****
570 * ID          :
571 * Outline     : SCI 送信終了割り込み(TEI1)処理
572 *-----
573 * Include     : "iodef.h"
574 *-----
575 * Declaration : void io_int_sci_teil(void);
576 *-----
577 * Description : TEI1 要求および送信動作を禁止します。
578 *-----
579 * Argument    : void
580 *-----
581 * Return Value : void
582 *-----
583 * Note        : None
584 *"FUNC COMMENT END"*****/
585 void io_int_sci_teil(void)
586 {
587     unsigned char dummy;
588
589     /* ==== TEI1 要求禁止 ==== */
590     SCI1.SCSCR.BIT.TEIE = 0;
591     /* ==== 送信動作禁止 ==== */
592     SCI1.SCSCR.BIT.TE = 0;
593
594     dummy = SCI1.SCSCR.BYTE; /* dummy read */
595 }
596
597 /* End of File */
```


3.15 サンプルプログラムリスト "intprg.c" (1)

```
1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corp. and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corp. and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 *   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 /*"FILE COMMENT"***** Technical reference data *****
31 *   System Name : SH7216 Sample Program
32 *   File Name   : intprg.c
33 *   Abstract    : Interrupt Functions
34 *   Version     : 1.00.00
35 *   Device      : SH7216
36 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37 *                : C/C++ compiler package for the SuperH RISC engine family
38 *                :                (Ver.9.03 Release00).
39 *   OS          : None
40 *   H/W Platform: R0K572167 (CPU board)
41 *   Description :
42 *****/
43 *   History     : Jul.30,2010 Ver.1.00.00
44 *"FILE COMMENT END"*****
```

3.16 サンプルプログラムリスト "intprg.c" (2)

```
45  #include <machine.h>
46  #include "vect.h"
47
48  extern void io_int_cmt_cmi0(void);
49  extern void io_int_txil(void);
50  extern void io_int_teil(void);
51
52  #pragma section IntPRG
53
54  // 4 Illegal code
55  void INT_Illegal_code(void){/* sleep(); */}
56  // 5 Reserved
57
58  // 6 Illegal slot
59  void INT_Illegal_slot(void){/* sleep(); */}
60
61  ...
62  . . .
63
64  // 140 CMT CMI0
65  void INT_CMT_CMI0(void)
66  {
67      /* ==== CMT コンペアマッチ割り込み(CMI0)処理 ==== */
68      io_int_cmt_cmi0();
69  }
70
71  ...
72  . . .
73
74  // 246 SCI SCI1 TXI1
75  void INT_SCI_SCI1_TXI1(void)
76  {
77      /* ==== SCI 送信データエンプティ割り込み(TXI1)処理 ==== */
78      io_int_sci_txil();
79  }
80  // 247 SCI SCI1 TEI1
81  void INT_SCI_SCI1_TEI1(void)
82  {
83      /* ==== SCI 送信終了割り込み(TEI1)処理 ==== */
84      io_int_sci_teil();
85  }
86
87  ...
88  . . .
89
90  // 254 SCIF SCIF3 RXI3
91  void INT_SCIF_SCIF3_RXI3(void){/* sleep(); */}
92  // 255 SCIF SCIF3 TXI3
93  void INT_SCIF_SCIF3_TXI3(void){/* sleep(); */}
94  // Dummy
95  void Dummy(void){/* sleep(); */}
96
97  572
98  573  /* End of File */
```

3.17 サンプルプログラムリスト "dtc.h" (1)

```
1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corp. and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corp. and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 *   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *****/
30 /*"FILE COMMENT"***** Technical reference data *****
31 *   System Name : SH7216 Sample Program
32 *   File Name   : dtc.h
33 *   Abstract    : Macro definition for DTC setting
34 *   Version     : 1.00.00
35 *   Device      : SH7216
36 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37 *                : C/C++ compiler package for the SuperH RISC engine family
38 *                :                               (Ver.9.03 Release00).
39 *   OS          : None
40 *   H/W Platform: R0K572167 (CPU board)
41 *   Description :
42 *****/
43 *   History     : Jul.30,2010 Ver.1.00.00
44 *"FILE COMMENT END"*****
45 #ifndef _DTC_H_
46 #define _DTC_H_
47
```

3.18 サンプルプログラムリスト "dtc.h" (2)

```
48  /* **** Structure definition **** */
49  /* ==== DTC 転送情報用レジスタ定義 ==== */
50  typedef struct {
51      unsigned char MRA;          /* DTC モードレジスタ A          */
52      unsigned char MRB;          /* DTC モードレジスタ B          */
53      unsigned char dummy1;       /* Reserved                       */
54      unsigned char dummy2;       /* Reserved                       */
55      unsigned long SAR;           /* DTC ソースアドレスレジスタ   */
56      unsigned long DAR;          /* DTC デスティネーションアドレスレジスタ */
57      union{
58          unsigned short WORD;
59          struct {
60              unsigned char H;
61              unsigned char L;
62          } BYTE;
63      } CRA;                       /* DTC 転送カウントレジスタ A   */
64      unsigned short CRB;          /* DTC 転送カウントレジスタ B   */
65  } DTC_TRANS_INFO;
66
67
68  /* **** Macro definition **** */
69  /* ==== 転送情報格納領域 ==== */
70  #define DTC_CMI0    (*(volatile DTC_TRANS_INFO *)0xffff8400) /* CMI0 起動用 */
71  #define DTC_ADI0_CHN1 (*(volatile DTC_TRANS_INFO *)0xffff8401) /* ADI0 起動用 1 */
72  #define DTC_ADI0_CHN2 (*(volatile DTC_TRANS_INFO *)0xffff8402) /* ADI0 起動用 2 */
73  #define DTC_TXI1    (*(volatile DTC_TRANS_INFO *)0xffff8403) /* TXI1 起動用 */
74
75  /* ==== DTC ベクタ設定用パラメータ ==== */
76  /* ---- DTC ベクタテーブル用ベースアドレス ---- */
77  #define DTC_VECT_BASE 0xffff8500
78  /* ---- DTC 起動要因ベクタアドレスオフセット ---- */
79  #define DTC_VECT_ADI0 0x0570 /* ADI0 */
80  #define DTC_VECT_CMI0 0x0630 /* CMI0 */
81  #define DTC_VECT_TXI1 0x07d8 /* TXI1 */
82
83  /* ==== io_init_dtc 関数の第 4 引数(転送回数)設定用パラメータ ==== */
84  #define DTC_COUNT_LINK 10 /* DTC 連動回数 */
85  #define DTC_COUNT_BLKSIZE 0x00040000 /* ブロック転送時のブロックサイズ(bit23-16) */
86  #define DTC_COUNT_SCI 8 /* シリアルデータ転送回数 */
87
```

3.19 サンプルプログラムリスト "dtc.h" (3)

```
88  /* ==== io_init_dtc 関数の第 5 引数(mode)設定用パラメータ ==== */
89  /* ---- 起動要因トリガ ---- */
90  #define DTC_TRG_CMI0    0x00010000 /* CMI0 */
91  #define DTC_TRG_ADI0    0x00020000 /* ADI0 */
92  #define DTC_TRG_TXI1    0x00030000 /* TXI1 */
93  /* ---- 転送モード ---- */
94  #define DTC_MODE_NORMAL  0x00000000 /* ノーマル転送モード */
95  #define DTC_MODE_REPEAT  0x00004000 /* リピート転送モード */
96  #define DTC_MODE_BLOCK   0x00008000 /* ブロック転送モード */
97  /* ---- 転送サイズ ---- */
98  #define DTC_SIZE_BYTE    0x00000000 /* バイトサイズ転送 */
99  #define DTC_SIZE_WORD    0x00001000 /* ワードサイズ転送 */
100 #define DTC_SIZE_LONG    0x00002000 /* ロングワードサイズ転送 */
101 /* ---- 転送元/転送先アドレスモード ---- */
102 #define DTC_SRC_FIX       0x00000000 /* 転送後 SAR は固定 */
103 #define DTC_SRC_UP        0x00000800 /* 転送後 SAR をインクリメント */
104 #define DTC_SRC_DOWN      0x00000c00 /* 転送後 SAR をデクリメント */
105 #define DTC_DST_FIX       0x00000000 /* 転送後 DAR は固定 */
106 #define DTC_DST_UP        0x00000008 /* 転送後 DAR をインクリメント */
107 #define DTC_DST_DOWN      0x0000000c /* 転送後 DAR をデクリメント */
108 /* ---- チェイン転送関連 ---- */
109 #define DTC_CHN_DISABLE   0x00000000 /* チェイン転送不使用 */
110 #define DTC_CHN_INITIAL   0x00000080 /* チェイン転送使用(チェインの先頭) */
111 #define DTC_CHN_MIDDLE    0x00000081 /* チェイン転送使用(チェインの中間) */
112 #define DTC_CHN_FINAL     0x00000002 /* チェイン転送使用(チェインの最後尾) */
113 #define DTC_CHN_CONTINUE  0x00000000 /* 連続してチェイン転送実施 */
114 #define DTC_CHN_COUNTER0  0x00000040 /* 転送回数ゼロ時のみチェイン転送実施 */
115 /* ---- CPU への割り込み要求発生タイミング ---- */
116 #define DTC_INT_TRANSTIME 0x00000000 /* 転送回数ごとに発生 */
117 #define DTC_INT_EVERYTIME 0x00000020 /* 1 回の転送ごとに発生 */
118 /* ---- リピート/ブロック転送モード時の対象領域指定 ---- */
119 #define DTC_TARGET_DST    0x00000000 /* デスティネーション側がリピート/ブロック領域 */
120 #define DTC_TARGET_SRC    0x00000010 /* ソース側がリピート/ブロック領域 */
121 /* ---- DTCCR 設定用 ---- */
122 #define DTC_CR_RRS_DISABLE 0x00000000 /* 転送情報リードスキップ禁止 */
123 #define DTC_CR_RRS_ENABLE  0x10000000 /* 転送情報リードスキップ許可 */
124 #define DTC_CR_RCHNE_DISABLE 0x00000000 /* リピート転送後のチェイン転送禁止 */
125 #define DTC_CR_RCHNE_ENABLE 0x08000000 /* リピート転送後のチェイン転送許可 */
126
127
128 #endif /* _DTC_H_ */
129
130 /* End of File */
```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A、SH2A-FPU ソフトウェアマニュアル Rev.3.00
(最新版をルネサス エレクトロニクスのホームページから入手してください。)
- ハードウェアマニュアル
SH7214 グループ、SH7216 グループ ユーザーズマニュアル：ハードウェア編 Rev.2.00
(最新版をルネサス エレクトロニクスのホームページから入手してください。)

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/>

お問合せ先

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

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.07.30	—	初版発行

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っていると、内部メモリ、レイアウトパターンなどの相違などにより、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て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きを行ってください。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。



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/inquiry>