

# SH7216グループ

RJJ06B1156-0100

Rev. 1.00

2010.05.14

## コントローラエリアネットワーク

### データトランスファコントローラを用いたデータフレーム受信設定例

#### 要旨

本アプリケーションノートでは、コントローラエリアネットワークで受信したデータフレームを、データトランスファコントローラを使用して内蔵 RAM へ格納する設定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.....	19
4. 参考ドキュメント.....	28

## 1. はじめに

### 1.1 仕様

コントローラエリアネットワークがデータフレームを受信するとデータトランスファコントローラを起動し内蔵RAMにメールボックス0の内容を転送します。この動作を5回繰り返します。5回の転送終了後、割り込みが発生し、その処理内でDTCの起動要因を再設定し次のデータフレーム受信に備えます。

### 1.2 使用機能

- コントローラエリアネットワーク
  - 通信速度：1Mbps
  - 受信メールボックス：メールボックス0を使用
  - メールボックス0の設定：ID:H'00A、スタンダードフォーマット
- データトランスファコントローラ
  - 転送モード：ブロック転送モード
  - 転送データのデータサイズ：ワード
  - 起動要因：コントローラエリアネットワーク\_メールボックス0のデータフレーム受信割り込み
  - 転送元：コントローラエリアネットワークのメールボックス0
  - 転送先：内蔵RAM (H'FFF8 C000~H'FFF8 C059、H'FFF8 C100~H'FFF8 C159)
  - ブロックエリア：転送元
  - ブロックサイズ：9ワード
  - 転送回数：5回
  - 転送情報テーブル先頭アドレス：H'FFF8 8800

### 1.3 適用条件

マイコン	SH7216
動作周波数	内部クロック：200MHz バスクロック：50MHz 周辺クロック：50MHz
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Ver.4.05.01
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 Release00
コンパイルオプション	-cpu=sh2afpu -fpu=single -include="\$(WORKSPDIR)¥inc" -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo

### 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

本応用例ではコントローラエリアネットワークで受信したデータフレームを、データトランスファコントローラを使用して内蔵 RAM へ格納します。

### 2.1 使用機能の動作概要

#### (1) コントローラエリアネットワーク

SH7216は CAN2.0B Active と ISO-11898 をサポートするコントローラエリアネットワークを内蔵しています。

コントローラエリアネットワークはプログラム可能な 15 個の送受信用メールボックスおよび 1 個の受信用メールボックス、全てのメールボックスに対応したプログラム可能な受信フィルタマスクを持ち、自由度の高い通信方法を提供します。表 1 にコントローラエリアネットワークの特長を、図 1 にブロック図を示します。また、表 2 に割り込み要因を示します。ダイレクトメモリアクセスコントローラを起動できる割り込み要因はメールボックス 0 のデータフレーム受信割り込みとリモートフレーム受信割り込みのみです。コントローラエリアネットワークについての詳細は、「SH7216グループ ハードウェアマニュアル コントローラエリアネットワーク」の章を参照してください。

表 1 コントローラエリアネットワークの特長

項目	内容
対応規格	CAN 規格 2.0B。ビットタイミングは ISO-11898 規格に準拠
メールボックス	16 個（プログラム可能な 15 個の送受信用メールボックスおよび 1 個の受信用メールボックス）
転送レート	最大 1Mbps
割り込み要因	12
テスト機能	リスンオンリモード、エラーパッシブモードを内蔵

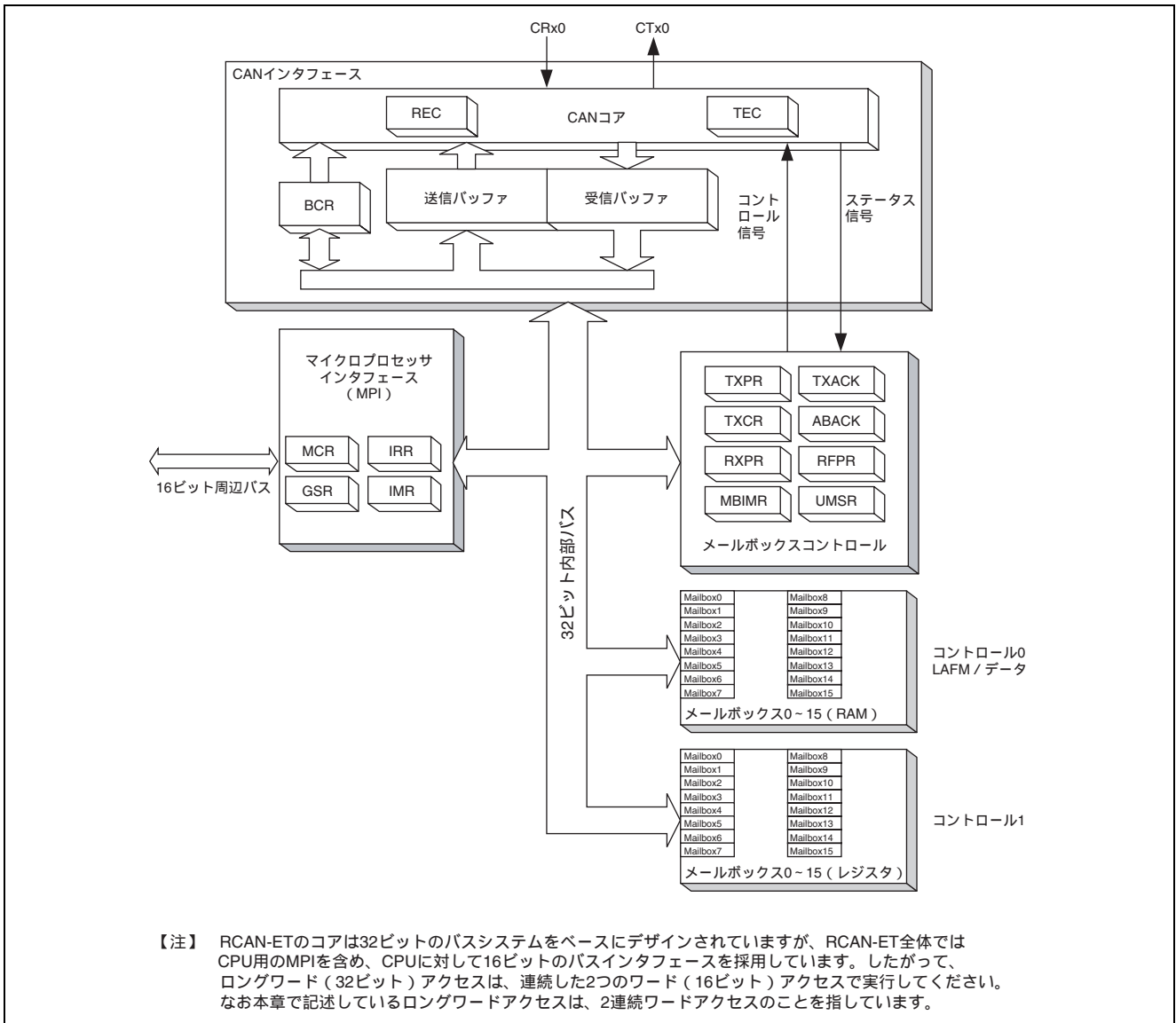


図 1 コントローラエリアネットワークのブロック図

表 2 コントローラエリアネットワークの割り込み要因

名称	要因	割り込み フラグ	データトランスファコントローラ/ ダイレクトメモリアクセスコント ローラの起動
ERS_0	エラーパッシブ	IRR5	不可
	バスオフ/バスオフからの復帰	IRR6	
	エラーワーニング (TEC $\geq$ 96)	IRR3	
	エラーワーニング (REC $\geq$ 96)	IRR4	
OVR_0	メッセージエラー検出	IRR13 <sup>※1</sup>	
	リセット/ホルト/CAN スリープ遷移	IRR0	
	オーバーロードフレーム送信	IRR7	
	未読メッセージのオーバーライト (オーバーラン)	IRR9	
	CAN スリープ中 CAN バス動作の検出	IRR1	
RM0_0 <sup>※2</sup>	データフレーム受信	IRR1 <sup>※3</sup>	
RM1_0 <sup>※2</sup>	リモートフレーム受信	IRR2 <sup>※3</sup>	
SLE_0	メッセージの送信/送信取り消し	IRR8	不可

【注】 1. テストモードでのみ有効。

2. RM0\_0 はメールボックス 0 のリモートフレーム受信フラグ (RFPR0[0]) またはデータフレーム受信フラグ (RXPR0[0]) による割り込み、RM1\_0 はメールボックス n (n=1~15) のリモートフレーム受信フラグ (RFPR0[n]) またはデータフレーム受信フラグ (RXPR0[n]) による割り込みです。
3. IRR1 はメールボックス 0~15 のデータフレーム受信フラグ、IRR2 はメールボックス 0~15 のリモートフレーム受信フラグです。
4. RM0\_0 割り込みのみ起動できます。

## (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、割り込み要求によって起動されデータ転送を行います。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。表 3にデータトランスファコントローラの特長を、図 2にブロック図を示します。データトランスファコントローラについての詳細は、「SH7216グループ ハードウェアマニュアル データトランスファコントローラ」の章を参照してください。

表 3 データトランスファコントローラの特長

項目	内容
転送モード	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
データサイズ	バイト、ワード、ロングワード
チェイン転送	あり（チェイン転送は一つの起動要因に対して複数のデータ転送が可能）
割り込み	DTC を起動した割り込みを CPU に要求可能 一回のデータ転送終了後に CPU に対する割り込み要求を発生可能 指定したデータ転送終了後に CPU に対する割り込み要求を発生可能
転送情報のリードスキップ機能	あり
ショートアドレスモード	あり

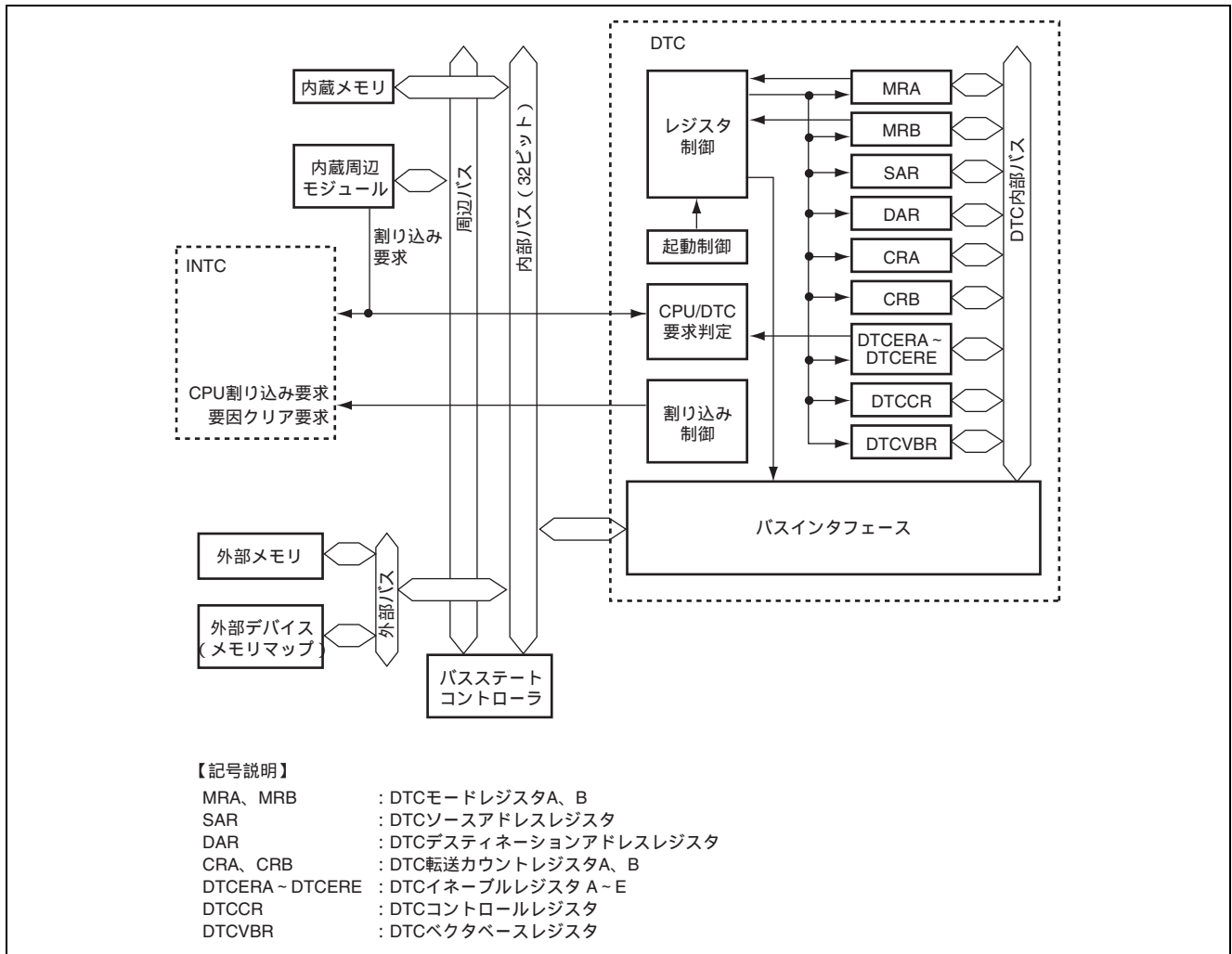


図 2 データトランスファコントローラのブロック図

## 2.2 データトランスファコントローラの転送情報

データトランスファコントローラは、転送情報に従ってデータ転送を行ないます。

転送情報は、リード、ライト可能なRAM上に配置します。転送情報の配置を図3に示します。また、転送情報の先頭アドレスはDTCベクタテーブルで示されます。DTCベクタテーブルと、転送情報の対応を図4に示します。

データトランスファコントローラは、起動要因別にDTCベクタテーブルから転送情報の先頭アドレスをリードし、この先頭アドレスから転送情報をリードします。

転送情報の先頭アドレスは、4バイト境界になるようにDTCベクタテーブルに設定して下さい。DTCベクタテーブルに4バイト境界以外の値を設定した場合、データトランスファコントローラは下位2ビットを無視して転送情報をリードします。

データトランスファコントローラは、転送情報として、転送元アドレス、転送先アドレス、転送回数をそれぞれ、DTCソースアドレスレジスタ(SAR)、DTCデスティネーションアドレスレジスタ(DAR)、DTC転送カウントレジスタA(CRA)に設定します。

データトランスファコントローラは、1回の転送が終了するごとに、DTCソースアドレスレジスタ(SAR)、DTCデスティネーションアドレスレジスタ(DAR)の値を、DTCモードレジスタA(MRA)、DTCモードレジスタB(MRB)の値に従って、インクリメントまたはデクリメント、あるいは保持します。また、DTC転送カウントレジスタA(CRA)の値は、1回の転送が終了するごとにデクリメントされます。

データトランスファコントローラは、1回の転送が終了するごとに、更新したこれらのレジスタ情報を転送情報に書き戻します。

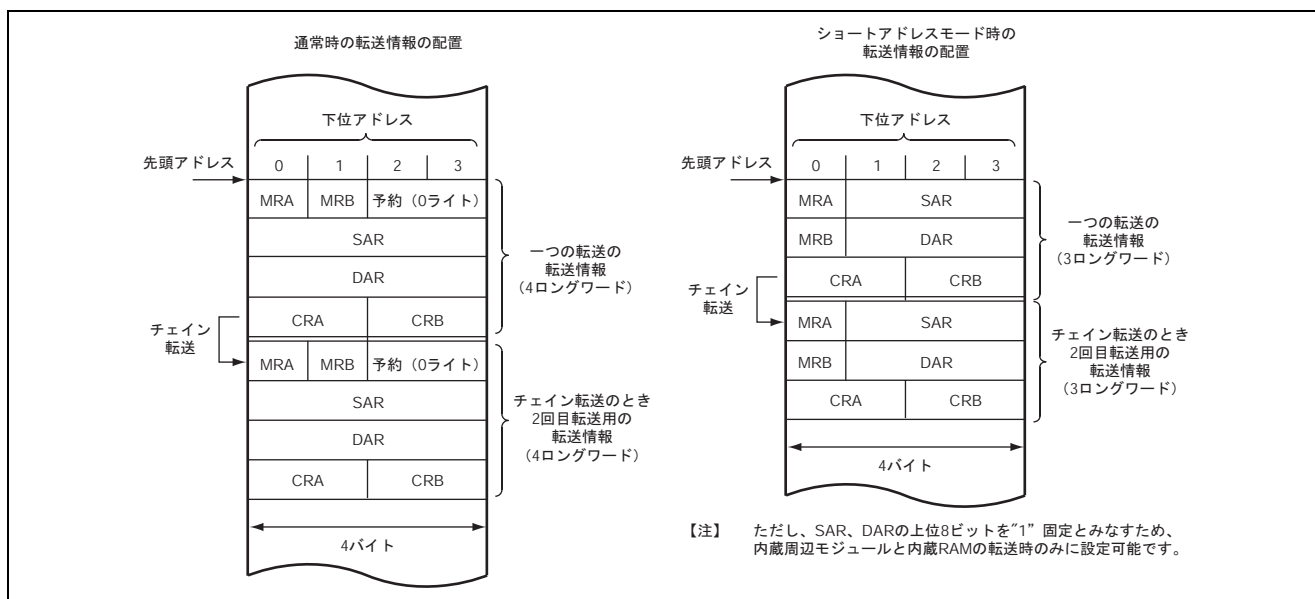


図3 データ領域上での転送情報の配置



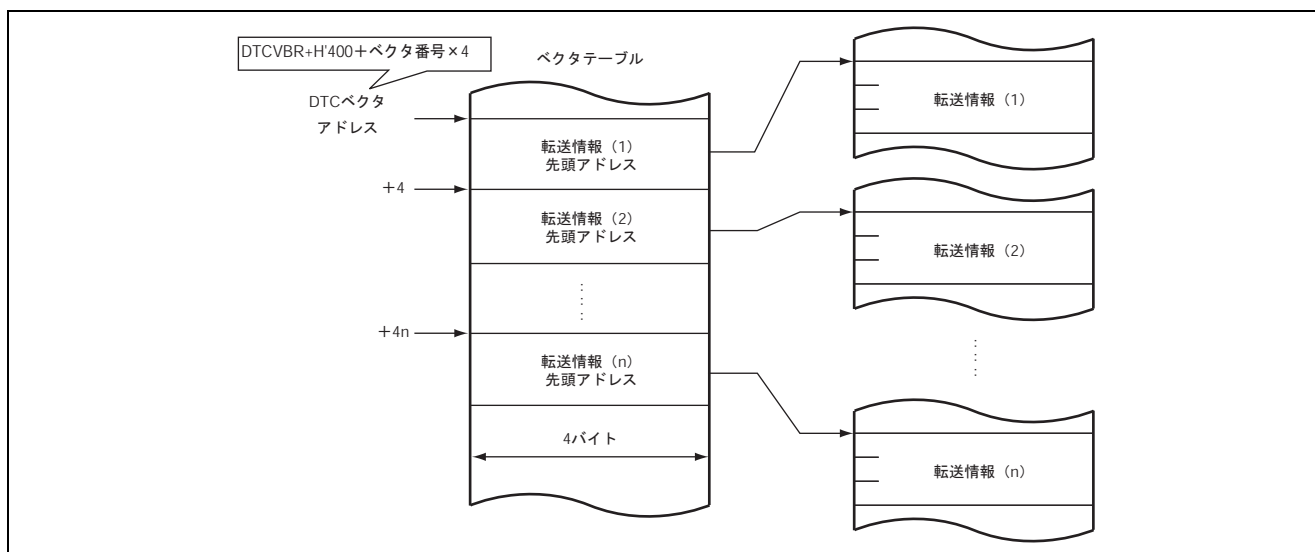


図 4 DTC ベクタテーブルと転送情報の対応

## 2.3 使用機能の設定手順

### (1) コントローラエリアネットワークの設定手順

コントローラエリアネットワークの初期設定はリセットモード（コンフィギュレーションモード）で行います。初期設定終了後、リセットモードを解除することでCANバスアクティビティに参加します。コントローラエリアネットワークのデータフレーム受信割り込み（RM0\_0）でダイレクトメモリアクセスコントローラを起動させる場合、インタラプトマスクレジスタのビット1とメールボックスインタラプトマスクレジスタ0のビット0を割り込み許可に設定します。

図5、図6にコントローラエリアネットワークの初期設定フロー例を示します。なお、各レジスタ設定の詳細は、「SH7216グループ ハードウェアマニュアル」を参照してください。

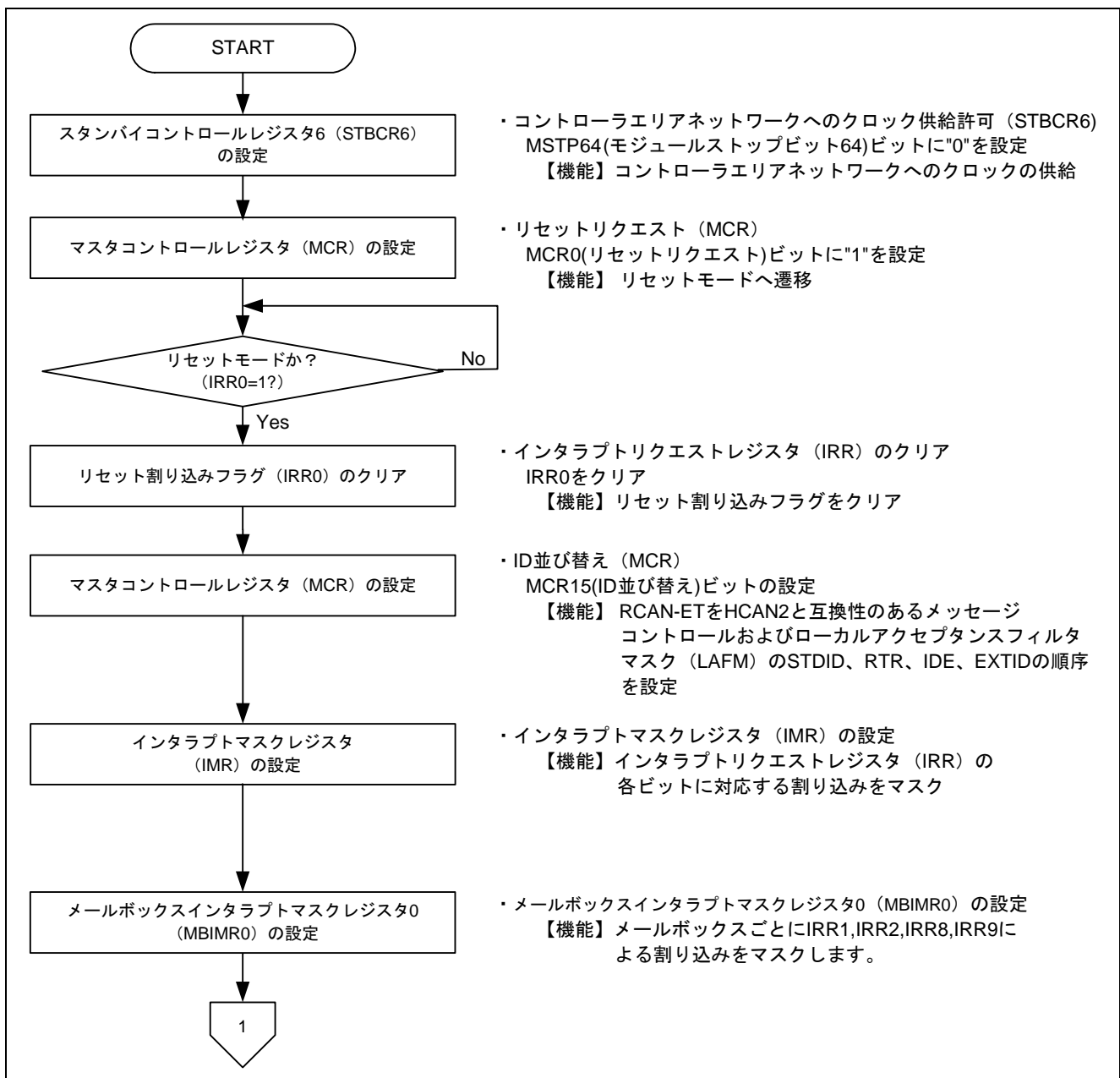


図5 コントローラエリアネットワークの初期設定フロー例 (1)

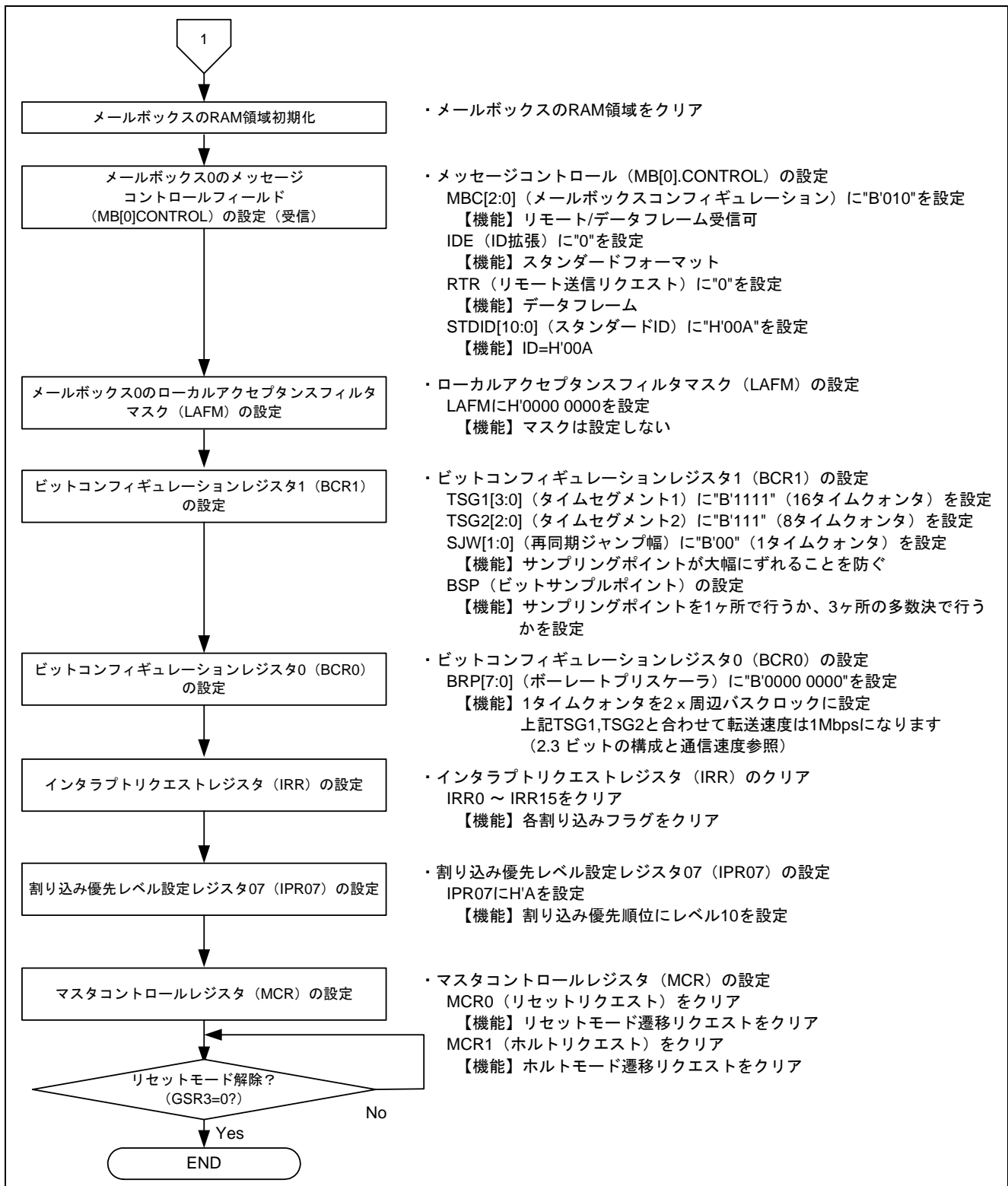


図 6 コントローラエリアネットワークの初期設定フロー例 (2)

## (2) データトランスファコントローラの設定手順

図 7にデータトランスファコントローラの初期設定フロー例を示します。なお、各レジスタ設定の詳細は、「SH7216グループ ハードウェアマニュアル データトランスファコントローラ」の章を参照してください。

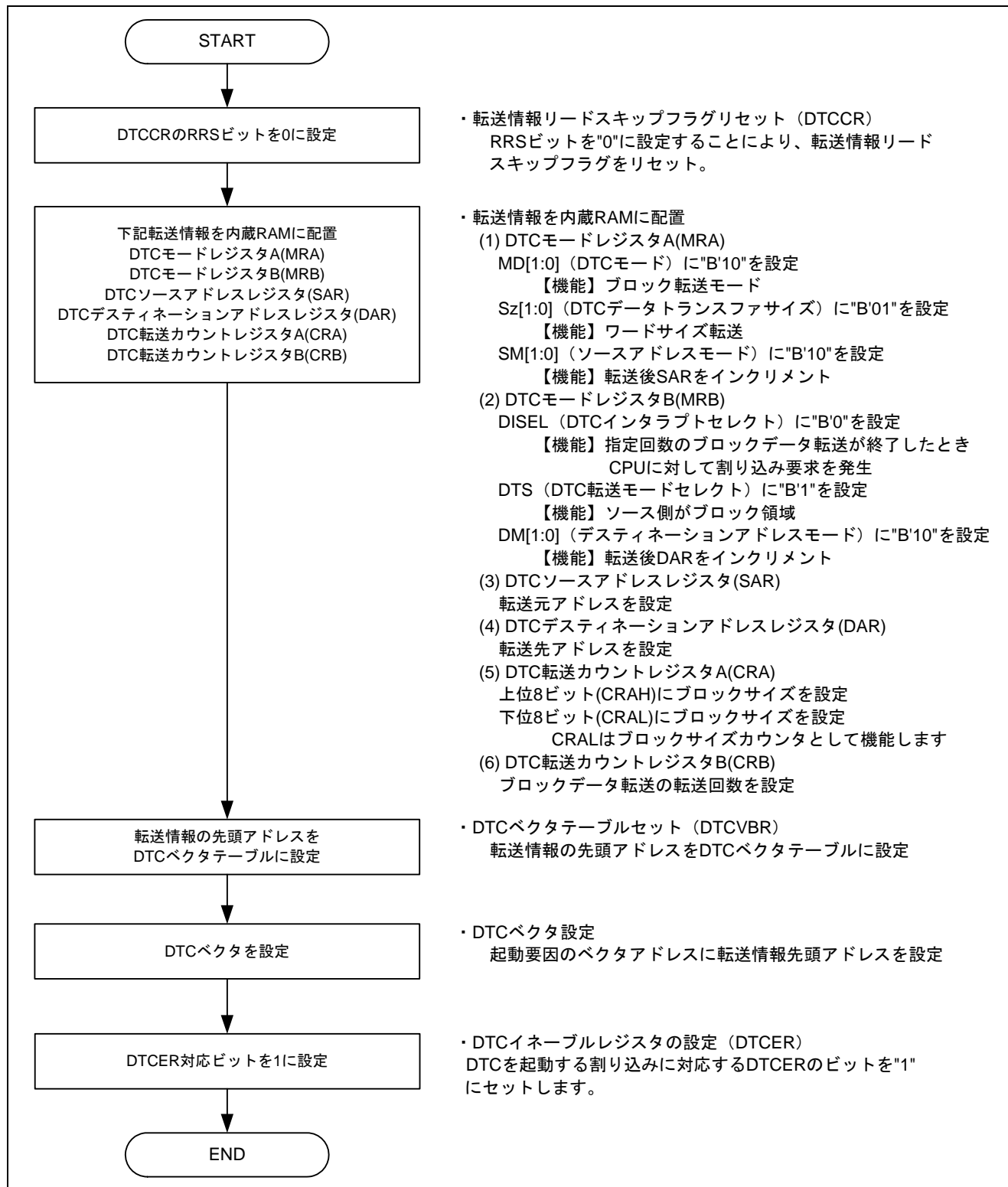


図 7 データトランスファコントローラの初期設定フロー例

## 2.4 コントローラエリアネットワークのビットの構成と通信速度

コントローラエリアネットワークでの1ビットは以下に示す4つのセグメントで構成されています。

- ① シンクロナイゼーションセグメント (SS)
- ② プロパゲーションタイムセグメント (PRSEG)
- ③ フェーズバッファセグメント 1 (PHSEG1)
- ④ フェーズバッファセグメント 2 (PHSEG2)

さらに、各セグメントは  $T_q$  (タイムクオンタ) と呼ばれる基準時間で構成されます。図 8 に  $SS=1T_q$ 、 $PRSEG=8T_q$ 、 $PHSEG1=8T_q$ 、 $PHSEG2=8T_q$  の場合のビット構成例を示します。

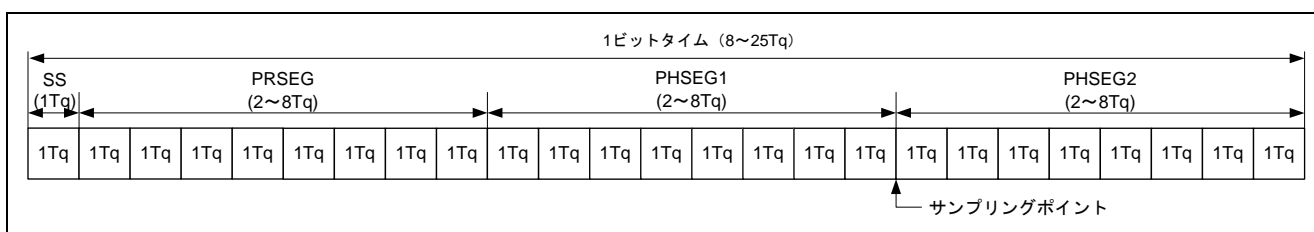


図 8 1 ビットの構成図

CAN では  $1T_q = \frac{2 \times (\text{BRP}[7:0] + 1)}{\text{周辺バスクロック}}$  と定義しており、通信速度は以下の通り計算されます。

$$\begin{aligned} \text{通信速度} &= \frac{\text{周辺バスクロック}}{(2 \times (\text{BRP}[7:0] + 1) \times 1 \text{ビットの} T_q \text{数})} \\ &= \frac{\text{周辺バスクロック}}{(2 \times (\text{BRP}[7:0] + 1) \times ((\text{TSG1}[3:0] + 1) + (\text{TSG2}[2:0] + 1) + 1))} \end{aligned}$$

コントローラエリアネットワークでは、PRSEG+PHSEG1 の  $T_q$  数をビットコンフィギュレーションレジスタ 1 (BCR1) の TSG1[3:0] に、PHSEG2 の  $T_q$  数を TSG2[2:0] に設定します (設定値+1 が  $T_q$  数)。また、 $1T_q$  に対応する周辺バスクロック数をビットコンフィギュレーションレジスタ 0 (BCR0) の BRP[7:0] に設定します。

以下の説明では BRP[7:0]、TSG1[3:0]、TSG2[2:0] はレジスタ設定値、BRP、TSEG1、TSEG2、SJW はレジスタ設定値に対応する値を示します。レジスタ設定値に対応する値は「SH7216 グループ ハードウェアマニュアル コントローラエリアネットワーク」の章を参照してください。

ビットコンフィギュレーションレジスタ設定上の制限事項を以下に示します。

$$\text{TSEG1 (Min.)} > \text{TSEG2} \geq \text{SJW (Max)} \quad (\text{SJW}=1 \sim 4)$$

SJW: 再同期ジャンプ幅。位相誤差を補正するためにフェーズバッファセグメント 1 を延長、または、フェーズバッファセグメント 2 を短縮するセグメントです。

$$8 \leq \text{TSEG1} + \text{TSEG2} + 1 \leq 25 \text{ タイムクオンタ}$$

$$\text{TSEG2} \geq 2$$

本参考プログラムでは、周辺バスクロック=50MHz、BRP[7:0]=0、TSG1[3:0]=15、TSG2[2:0]=7 に設定していますので通信速度は下記のようになります。

$$\text{通信速度} = \frac{50\text{M}}{2 \times (0+1) \times \{(15+1) + (7+1) + 1\}} = 1\text{M} \cdot \cdot \cdot 1\text{Mbps}$$

## 2.5 参考プログラムの動作

本参考プログラムでは、メールボックス 0 に ID : H'00A のスタンダードフォーマットのデータフレームを受信し割り込みを発生します。このデータフレーム受信割り込み (RM0\_0) を起動要因としデータトランスファコントローラでメールボックス 0 の内容を内蔵 RAM にブロック転送します。指定回数のブロック転送が終了すると CPU に割り込みを発生し、割り込み処理ではデータトランスファコントローラの再設定を行います。転送先内蔵 RAM 領域は 2 領域用意し、割り込み処理ごとに領域を切り替えます。

注： データフレーム受信割り込み (RM0\_0) を起動要因としデータトランスファコントローラでメールボックス 0 の内容を転送する場合、メールボックス 0 のメッセージコントロールフィールド 0 (CONTROL0) からメッセージコントロールフィールド 1 (CONTROL1) までのデータを転送する必要があります。

図 9 に参考プログラムの動作概要を、図 10 に参考プログラムの転送領域を示します。

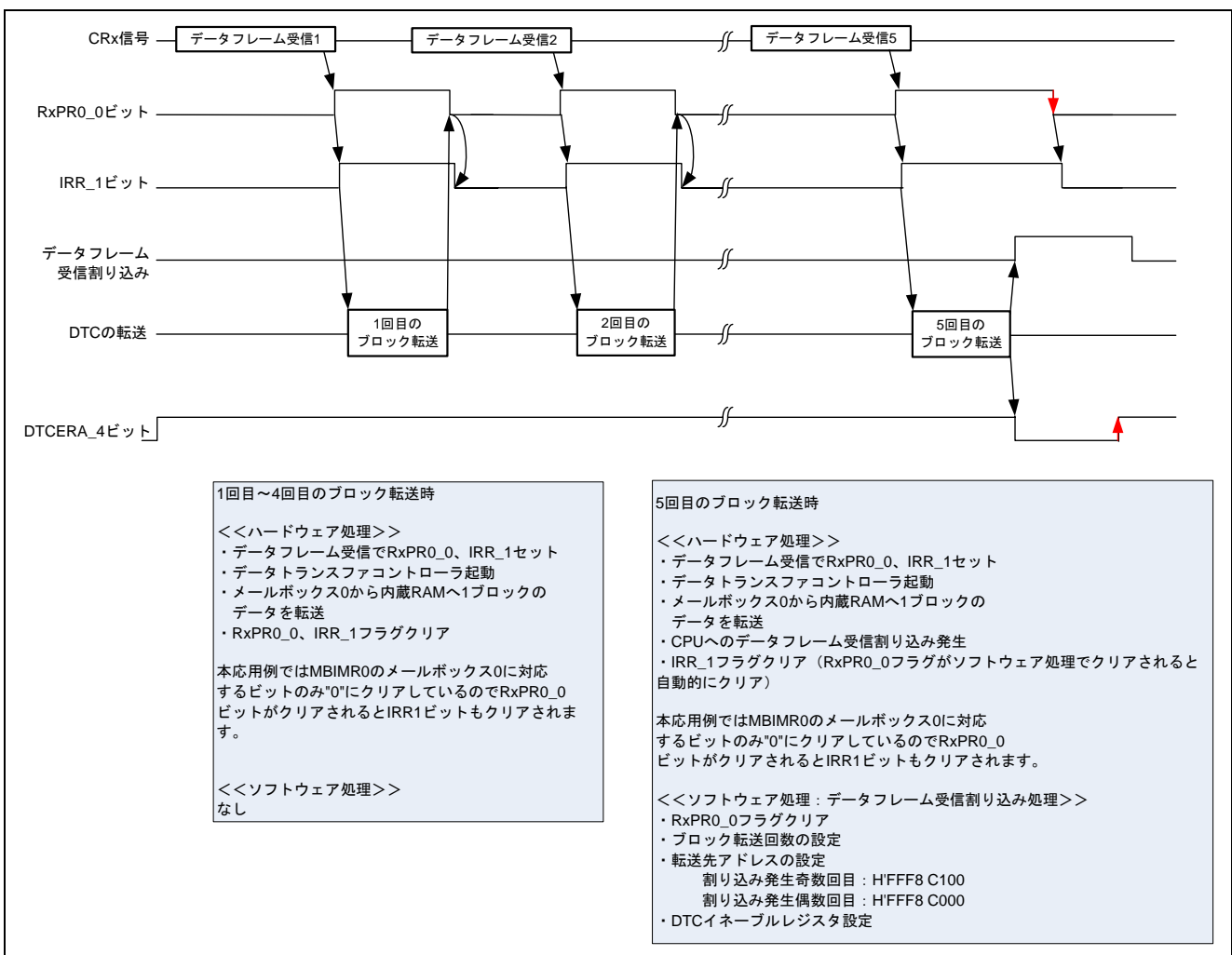


図 9 参考プログラムの動作概要

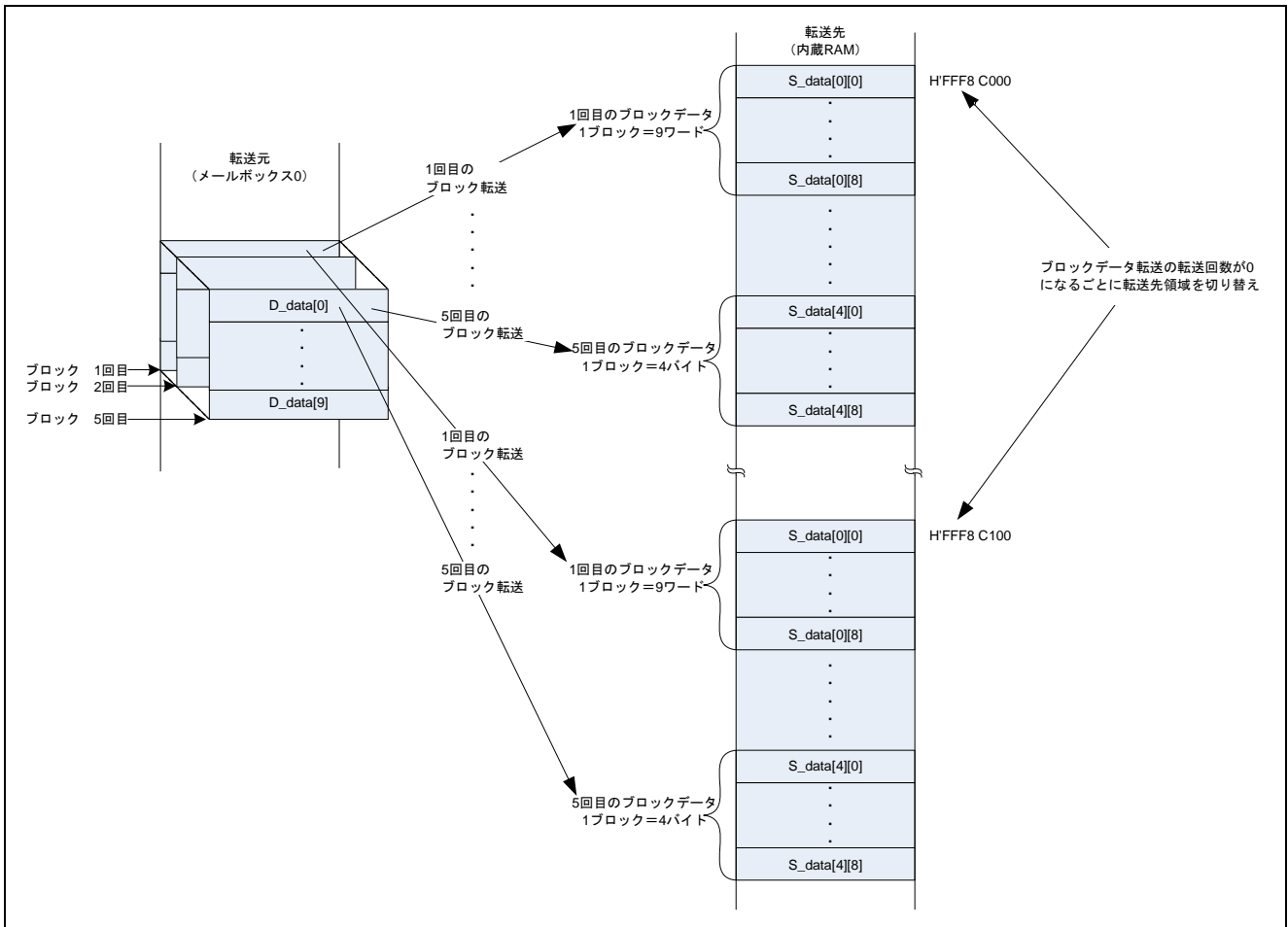


図 10 参考プログラムの転送領域

## 2.6 参考プログラムの処理手順

表 4 にコントローラエリアネットワークの設定例を、表 5 にデータトランスファコントローラの設定例を示します。また、図 11 に本参考プログラムのフローを示します。

表 4 コントローラエリアネットワークの設定

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 (STBCR6)	H'FFFE 041C	H'8F	MSTP64="0" : コントローラエリアネットワークは動作
マスタコントロールレジスタ (MCR)	H'FFFF D000	H'0001	MCR0="1" : リセットモード遷移リクエスト
		H'8001	MCR15="1" : RCAN-ET と HCAN2 は異なる順序
		H'8000	MCR0="0" : リセットモード解除
メールボックスインタラプトマスクレジスタ 0 (MBIMR0)	H'FFFF D052	H'FFFE	MBIMR00="0" : メールボックス 0 の受信割り込みを許可
インタラプトマスクレジスタ (IMR)	H'FFFF D00A	H'FFFD	データフレーム受信割り込み許可
ビットコンフィギュレーションレジスタ 1 (BCR1)	H'FFFF D004	H'F700	TSG1[3:0]="B'1111" : PRSEG + PHSEG1 = 16Tq  TSG2[2:0]="B'111" : PHSEG2 = 8Tq SJW="0" : SJW=1Tq BSP="0" : 1 サンプルング
ビットコンフィギュレーションレジスタ 0 (BCR0)	H'FFFF D006	H'0000	BRP[7:0]= "0" : 1Tq=2xPφ
メールボックス 0 メッセージコントロールフィールド (MB[0].CONTROL1)	H'FFFF D110	H'0200	MBC[2:0]="B'010" : データフレームとリモートフレームの受信可
メールボックス 0 コントロールフィールド (MB[1].CONTROL0)	H'FFFF D120	H'0028 0000	IDE="0" : スタンダードフォーマット RTR="0" : データフレーム STDID[10:0]= " H'00A " : スタンダード ID= H'00A
メールボックス 0 ローカルアクセプタンスフィルタマスク (MB[0].LAFM)	H'FFFF D104	H'0000 0000	クリア : MASK は設定しない



表 5 データトランスファコントローラ (DTC) の設定

レジスタ名	アドレス	設定値	機能
DTC コントロールレジスタ (DTCCR)	H'FFFE 6010	H'00	RRS="0" : DTC 転送情報リードスキップ ディセーブル
DTC ベクタベースレジスタ (DTCVBR)	H'FFFE 6014	H'FFF8 8000	DTC ベクタテーブルの先頭アドレス : H'FFFF A000
DTC イネーブルレジスタ A (DTCERA)	H'FFFE 6000	H'0010	RM0_0="1" : DTC 起動要因を RM0_0 に 設定

表 6 転送情報の設定

レジスタ名	設定値	機能
DTC モードレジスタ A (MRA)	H'98	<ul style="list-style-type: none"> <li>MD="B'10" : ブロック転送モード</li> <li>Sz="B'01" : ワードサイズ転送</li> <li>SM="B'10" : 転送後 SAR をインクリメント</li> </ul>
DTC モードレジスタ B (MRB)	H'18	<ul style="list-style-type: none"> <li>CHNE="0" : チェイン転送禁止</li> <li>DISEL="0" : 指定回数の転送終了時のみ割り込み</li> <li>DTS="B'1" : ソース側がブロック領域</li> <li>DM="B'10" : 転送後 DAR をインクリメント</li> </ul>
DTC ソースアドレスレジスタ (SAR)	H'FFFF D100	転送元の先頭アドレス
DTC デスティネーションアドレスレジスタ (DAR)	H'FFF8 C000 H'FFF8 C100	転送先の先頭アドレス
DTC 転送カウントレジスタ A (CRA)	H'0909	ブロックサイズ : 9
DTC 転送カウントレジスタ B (CRB)	H'0005	ブロック転送回数 : 5

表 7 使用変数

変数名	型	機能	モジュール名
SrcData [9]	unsigned char	DTC 転送元データの格納	main 関数
DstData_even[5] [9]	unsigned short	DTC 転送先データの格納 (偶数回目のブロックデータ転送時)	main 関数
DstData_odd[5] [9]	unsigned short	DTC 転送先データの格納 (奇数回目のブロックデータ転送時)	main 関数

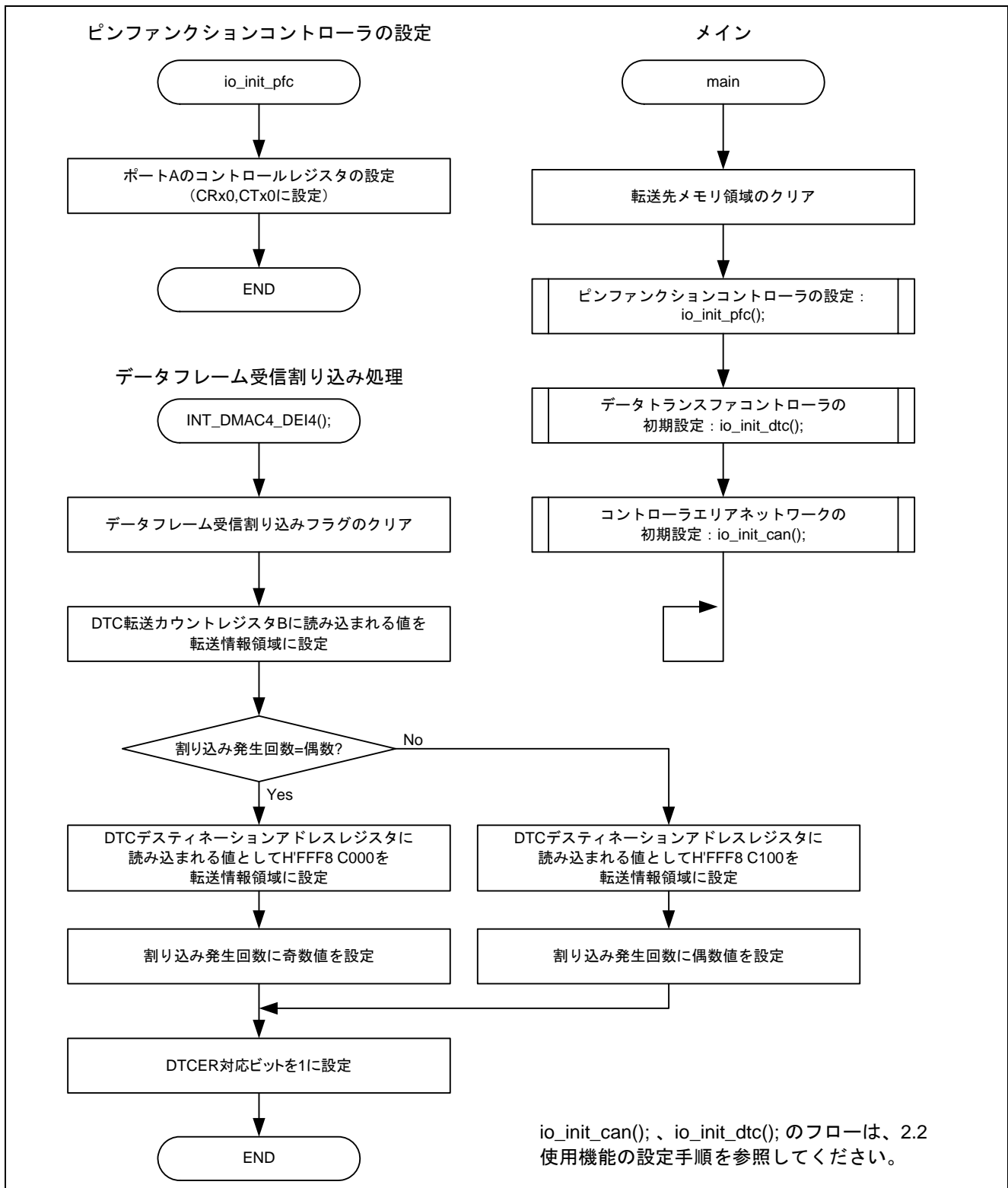


図 11 参考プログラムの処理フロー

### 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 *   (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *   "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7216 Sample Program
31 *   File Name   : main.c
32 *   Abstract    : DTC+CAN Module Application (Data Frame Receive)
33 *   Version     : 1.00.00
34 *   Device      : SH7216
35 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *               : C/C++ compiler package for the SuperH RISC engine family
37 *               :                               (Ver.9.03 Release00).
38 *   OS          : None
39 *   H/W Platform: R0K572167 (CPU board)
40 *   Description :
41 *****/
42 *   History     : Apr.28,2010 Ver.1.00.00
43 *   "FILE COMMENT END"*****/
44 #include "iodefine.h"
45 #include "dtc.h"
46
```

## 3.2 サンプルプログラムリスト "main.c" (2)

```
47  /* ---- prototype declaration ---- */
48  void main(void);
49  void io_init_pfc(void);
50  void io_init_dtc(unsigned long sar, unsigned long dar, unsigned char block_size, unsigned short count);
51  void io_init_can(void);
52
53  /* ---- Global variable ---- */
54  #pragma section DTC_VECT_TABLE
55  unsigned long dtc_vect[512]; /* ベクタテーブル領域 */
56
57  #pragma section D_DATA_O
58  unsigned short DstData_odd[DTC_COUNT][DTC_BLOCK LENG]; /* 転送先メモリ領域 */
59
60  #pragma section D_DATA_E
61  unsigned short DstData_even[DTC_COUNT][DTC_BLOCK LENG]; /* 転送先メモリ領域 */
62
63  #pragma section
64  int Int_Flag_EvenOdd = 1; /* 割り込み発生奇数回_偶数回フラグ */
65
66  /*"FUNC COMMENT"*****
67  * ID      :
68  * Outline  : Sample program main
69  *-----
70  * Include  : "iodefine.h"
71  *-----
72  * Declaration : void main(void);
73  *-----
74  * Description : PFC の設定、DTC の初期設定と CAN の初期設定を行います。
75  *-----
76  * Argument   : void
77  *-----
78  * Return Value : void
79  *-----
80  * Note       : None
81  *"FUNC COMMENT END"*****
82  void main(void)
83  {
84      int i,j;
85
86      /* ==== 転送先メモリ領域のクリア ==== */
87      for(i = 0; i < DTC_COUNT; i++){
88          for(j = 0; j < DTC_BLOCK LENG; j++){
89              DstData_odd[i][j] = 0x0000;
90              DstData_even[i][j] = 0x0000;
91          }
92      }
93
94      /* ==== ベクタテーブル_転送情報領域のクリア ==== */
95      for(i = 0; i < 512; i++){
96          dtc_vect[i] = 0x00000000;
97      }
```

## 3.3 サンプルプログラムリスト "main.c" (3)

```
98
99     /* ==== PFC の設定 ==== */
100    io_init_pfc();
101
102    /* ==== DTC の初期設定 ==== */
103    io_init_dtc((unsigned long)RCANET.MB, (unsigned long)DstData_odd, DTC_BLOCK LENG, DTC_COUNT);
104
105    /* ==== CAN の初期設定 ==== */
106    io_init_can();
107
108    while(1){
109        if(Int_Flag_EvenOdd == 0){
110            /* ---- DTC 終了割り込み奇数回終了 ---- */
111            while(Int_Flag_EvenOdd == 0){
112            }
113        }
114        else{
115            /* ---- DTC 終了割り込み偶数回終了 ---- */
116            while(Int_Flag_EvenOdd == 1){
117            }
118        }
119    }
120 }
121
122 /*"FUNC COMMENT"*****
123 * ID      :
124 * Outline : PFC setting
125 *-----
126 * Include : "iodefine.h"
127 *-----
128 * Declaration : void io_init_pfc(void);
129 *-----
130 * Description : 端子機能の設定を行います。
131 *              : CRx0 input、CTx0 output
132 *-----
133 * Argument  : void
134 *-----
135 * Return Value : void
136 *-----
137 * Note      : None
138 *"FUNC COMMENT END"*****
139 void io_init_pfc(void)
140 {
141     /* ==== PFC の設定 ==== */
142     PFC.PACRL1.BIT.PA0MD = 0x5;          /* Set CRx0 */
143     PFC.PACRL1.BIT.PA1MD = 0x5;          /* Set CTx0 */
144 }
145
```

## 3.4 サンプルプログラムリスト "main.c" (4)

```
146  /*"FUNC COMMENT"*****
147  * ID      :
148  * Outline  : データトランスファコントローラの設定
149  *-----
150  * Include  : "iodefine.h"
151  *-----
152  * Declaration : void io_init_dtc(void);
153  *-----
154  * Description : データトランスファコントローラの初期設定を行います。
155  *             : ブロック転送モードでメールボックス 0(サイズ 9 のブロック)を 5 回
156  *             : 内蔵 RAM へ転送するように設定します。
157  *-----
158  * Arguments  : unsigned long sar   : 転送元アドレス
159  *             : unsigned long dar   : 転送先アドレス
160  *             : unsigned short num  : ブロックサイズ
161  *             : unsigned short num2 : 転送回数
162  *-----
163  * Return Value : void
164  *-----
165  * Note        : None
166  *"FUNC COMMENT END"*****/
167 void io_init_dtc(unsigned long sar, unsigned long dar, unsigned char block_size, unsigned short count)
168 {
169     /* ---- DTC コントロールレジスタの設定 ---- */
170     DTC_DTCCR.BYTE = 0x00;          /* 転送情報リードスキップ無し */
171
172     /* ==== DTC 転送情報の設定 ==== */
173     /* ---- DTC モードレジスタ A の設定 ---- */
174     DTC_REG.MRA = 0x98;             /* ブロック転送モード          */
175                                     /* ワードサイズ転送          */
176                                     /* 転送後 SAR をインクリメント */
177     /* ---- DTC モードレジスタ B の設定 ---- */
178     DTC_REG.MRB = 0x18;             /* チェイン転送禁止          */
179                                     /* 指定回数転送終了時に割り込み */
180                                     /* ソース側 がブロック領域    */
181                                     /* 転送後 DAR をインクリメント */
182     /* ---- DTC ソースアドレスレジスタの設定 ---- */
183     DTC_REG.SAR = (unsigned long)sar;
184     /* ---- DTC デスティネーションアドレスレジスタの設定 ---- */
185     DTC_REG.DAR = (unsigned long)dar;
186     /* ---- DTC 転送カウンタレジスタの設定 ---- */
187     DTC_REG.CRA.BYTE.H = block_size; /* ブロックサイズ          */
188     DTC_REG.CRA.BYTE.L = block_size; /* ブロックサイズカウンタ */
189     DTC_REG.CRB = count;           /* 転送回数                */
```

## 3.5 サンプルプログラムリスト "main.c" (5)

```
190 /* ---- DTC ベクタベースレジスタの設定 ---- */
191 DTC.DTCVBR = DTC_VECT_BASE;
192
193 /* ---- DTC ベクタの設定 ---- */
194 dtc_vect[0x5a8/sizeof(unsigned long)] = (unsigned long)&DTC_REG;
195
196 /* ---- DTC イネーブルレジスタ A の設定 ---- */
197 DTC.DTCERA.BIT.RM0 = 1; /* interrupt source RM0 */
198 }
199
200 /*"FUNC COMMENT"*****
201 * ID :
202 * Outline : Controller area network setting
203 *-----
204 * Include : "iodefine.h"
205 *-----
206 * Declaration : void io_init_can(void);
207 *-----
208 * Description : Controller area network の初期設定を行います。
209 * : 転送レートを 1Mbps に設定、メールボックス 0 を設定します。
210 * : メールボックス 0 のデータフレーム受信を許可します。
211 *-----
212 * Argument : void
213 *-----
214 * Return Value : void
215 *-----
216 * Note : None
217 *"FUNC COMMENT END"*****/
218 void io_init_can(void)
219 {
220 int i,j;
221
222 /* ==== スタンバイコントロールレジスタ 6 の設定 ==== */
223 STB.CR6.BYTE = 0x8f; /* RCAN のモジュールスタンバイ解除 */
224
225 /* ==== マスターコントロールレジスタの設定 ==== */
226 RCANET.MCR.WORD = 0x0001; /* リセットモードの設定 */
227 while((RCANET.IRR.WORD & CAN_IRR0) != CAN_IRR0){
228 /* リセットモード移行完了待ち */
229 }
230 /* ==== IRR = 1, GSR = 1 (自動的に設定) ==== */
231
232 /* ---- リセット割り込みフラグのクリア ---- */
233 RCANET.IRR.WORD = 0x0001;
234
235 /* ---- マスタコントロールレジスタの設定 ---- */
236 RCANET.MCR.WORD |= 0x8000; /* RCAN-ET は HCAN2 と別フォーマット */
237
```

## 3.6 サンプルプログラムリスト "main.c" (6)

```
238  /* ---- インタラプトマスクレジスタの設定 ---- */
239  RCANET.IMR.WORD = 0xfffd;
240
241  /* ---- メールボックスインタラプトマスクレジスタ 0 の設定 ---- */
242  RCANET.MBIMR0.WORD = 0xfffe;
243
244  /* ----メールボックスの RAM 領域をクリア ---- */
245  for(i = 0; i < 16; i++){
246      RCANET.MB[i].CTRL0.LONG = 0x00000000;
247      RCANET.MB[i].LAFM.LONG = 0x00000000;
248      for(j = 0; j < 8; j++){
249          RCANET.MB[i].MSG_DATA[j] = 0x00;
250      }
251  }
252
253  /* ---- メールボックス 0 の設定 ---- */
254  RCANET.MB[0].CTRL1.WORD = 0x0200;          /* MBC=2, dlc=0 */
255  RCANET.MB[0].CTRL0.LONG = 0x00280000;     /* standard data frame, id=0x00a */
256  RCANET.MB[0].LAFM.LONG = 0x00000000;
257  for(i = 0; i < 8; i++){                   /* データクリア */
258      RCANET.MB[0].MSG_DATA[i] = 0x00;
259  }
260
261  /* ---- ビットコンフィギュレーションレジスタの設定 ---- */
262  RCANET.BCR1.WORD = 0xf700;                 /* tsg1=15(16bit),tsg2=7(8bit),sjw=0(1bit),bsp=0 */
263  RCANET.BCR0.WORD = 0x0000;                 /* 1Mbps */
264
265  /* ---- インタラプトリクエストレジスタの設定 ---- */
266  RCANET.IRR.WORD = 0xffff;
267
268  /* =====割り込み優先レベル設定レジスタ 07 (IPR07) の設定===== */
269  INTC.IPR18.BIT._RCAN = 0xa;
270
271  /* ---- マスタコントロールレジスタの設定 ---- */
272  RCANET.MCR.WORD &= 0xf8fc;                 /* MCR0,MCR1 クリア */
273  while( (RCANET.GSR.WORD & 0x0008) != 0x0000 ){
274      /* リセットモード終了待ち */
275  }
276  }
277
278  /* End of File */
```



## 3.7 サンプルプログラムリスト "intprg.c"

```
==== 省略 ====

267
268 void INT_RCANET0_RM01_0(void)
269 {
270     /* ---- データフレーム受信割り込みフラグのクリア ---- */
271     RCANET.RXPR0.BIT.MB0 = 0x1;
272
273     DTC_REG.CRB = DTC_COUNT;          /* transfer counter */
274
275     if(Int_Count==0){
276         /* ---- DTC destination address register ---- */
277         DTC_REG.DAR = (unsigned long)DstData_odd;
278         Int_Count=1;
279     }
280     else{
281         /* ---- DTC destination address register ---- */
282         DTC_REG.DAR = (unsigned long)DstData_even;
283         Int_Count=0;
284     }
285
286     DTC.DTCERA.BIT.RM0 = 1;          /* interrupt source RM0 */
    }

==== 省略 ====
```

## 3.8 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 *  (C) 2010 Renesas Electronics Corporation. All rights reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7216 Sample Program
31 *  File Name   : dtc.h
32 *  Abstract    : DMAC+CAN Module Application (Data Frame Receive)
33 *  Version     : 1.00.00
34 *  Device      : SH7216
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                               (Ver.9.03 Release00).
38 *  OS          : None
39 *  H/W Platform: R0K572167 (CPU board)
40 *  Description :
41 *****/
42 *  History     : Apr.28,2010 Ver.1.00.00
43 *  "FILE COMMENT END"*****/
44
```

## 3.9 サンプルプログラムリスト "dtc.h" (2)

```
45  /* ---- structure definition ---- */
46  struct st_dtc_info{
47      unsigned char MRA;          /* DTC mode register A          */
48      unsigned char MRB;          /* DTC mode register B          */
49      unsigned char dummy1;       /* reserved                      */
50      unsigned char dummy2;       /* reserved                      */
51      unsigned long SAR;          /* DTC source address register  */
52      unsigned long DAR;          /* DTC destination address register */
53      union{
54          unsigned short WORD;
55          struct {
56              unsigned char H;
57              unsigned char L;
58          } BYTE;
59      } CRA;          /* DTC transfer count register A */
60      unsigned short CRB;        /* DTC transfer count register B */
61  };
62
63  /* ---- symbol definition ---- */
64  #define CAN_IRR0    0x0001
65  #define DTC_COUNT   5          /* DTC transfer count          */
66  #define DTC_BLOCK_LENG 0x09    /* DTC block size              */
67
68  #define DTC_REG(*(volatile struct st_dtc_info*)0xffff88800) /* DTC transfer information */
69  #define DTC_VECT_BASE 0xffff88000 /* DTC vector base address */
70
71  /* 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  
SH7216 グループ ハードウェアマニュアル Rev.1.01  
(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	2010.05.14	—	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くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載され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>