

# RX63T グループ

R01AN1240JJ0100

## 2 相励磁方式ステッピングモータ

Rev.1.00

2014.05.14

### 要旨

本アプリケーションノートでは、RX63T グループのマルチファンクションタイマパルスユニット 3、データトランスファコントローラ、コンペアマッチタイマ機能を用いた 2 相ステッピングモータの制御方法について説明します。

なお、本アプリケーションノートでは以下のアプリケーションノートのサンプルコードを使用しています。

- RX63T の初期設定  
「RX63T グループ 初期設定例」 Rev.1.00 (R01AN1252JJ0100)

### 対象デバイス

RX63T グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

### 目次

1. 仕様 .....	2
2. 動作確認条件 .....	3
3. 関連アプリケーションノート .....	3
4. ハードウェア説明 .....	4
5. ソフトウェア説明 .....	5
6. 注意事項 .....	37
7. サンプルコード .....	38
8. 参考ドキュメント .....	38

## 1. 仕様

RX63T グループのマルチファンクションタイムパルスユニット（以下 MTU3）、データトランスファコントローラ（以下 DTC）、コンペアマッチタイマ（以下 CMT）機能を用いて、2 相ステッピングモータを制御します。

なお、本アプリケーションノートのステッピングモータのモータステップ角は 7.5[deg/step]とし、モータドライバに出力するステッピングモータ制御パルスの生成を行います。

- ステッピングモータは 2 相励磁方式で制御し、正転 停止 逆転 停止の動作を繰り返します。
  - ステッピングモータ制御パルスは MTU3 の PWM 出力で行い、DTC 転送により Slow Up および Slow Down 処理を行います。
  - Constant 制御では、CMT で設定した時間分 Slow Up 制御後の PWM 出力を維持します。
  - モータ停止期間は Constant 期間と同じ時間停止します。
  - ステッピングモータの正転と逆転の制御は、モータドライバが出力する B 相、 $\bar{B}$  相出力波形に相当するステッピングモータ制御パルスの波形を反転させることで実現しています。
- なお、ステッピングモータ制御パルスはドライバ保護のため、貫通電流防止期間を設けています。

2 相ステッピングモータ制御の接続図を図 1.1 に示します。

本アプリケーションノートのステッピングモータ制御パルスは MTIOC0A、MTIOC4A、MTIOC0C、MTIOC4C 端子から出力します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
マルチファンクションタイムパルスユニット(MTU3)	ステッピングモータ制御パルス出力
データトランスファコントローラ (DTC)	Slow Up、Slow Down 制御
コンペアマッチタイマ (CMT)	Constant 期間およびモータ停止期間の時間計測

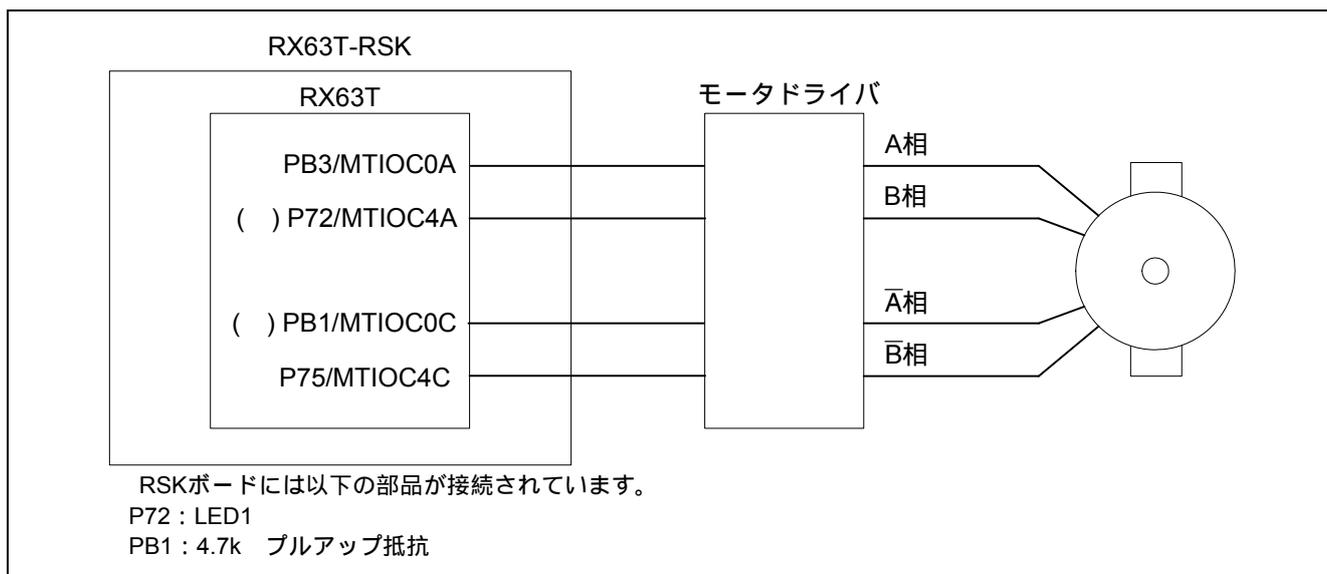


図 1.1 2 相ステッピングモータ制御の接続図

## 2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1 動作確認条件

項目	内容
使用マイコン	R5F563T6EDFM (RX63T グループ)
動作周波数	<ul style="list-style-type: none"> <li>● メインクロック : 16MHz</li> <li>● PLL : 192MHz (メインクロック 12 分周 12 逡倍)</li> <li>● システムクロック (ICLK) : 96MHz (PLL 2 分周)</li> <li>● タイマモジュールクロック (PCLKA) : 96MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック (PCLKB) : 48MHz (PLL 4 分周)</li> <li>● S12AD 用クロック (PCLKD) : 48MHz (PLL 4 分周)</li> <li>● FlashIF クロック (FCLK) : 48MHz (PLL 4 分周)</li> </ul>
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.09.01.007
C コンパイラ	RX Standard Toolchain (V1.2.1.0) RX Family C/C++ Compiler Driver V.1.02.01.000 RX Family C/C++ Compiler V.1.02.01.000 RX Family Assembler V.1.02.00.000 Optimizing Linkage Editor (V.10.02.00.000) RX Family C/C++ Standard Library Generator V.1.02.00.000 コンパイルオプション -cpu=rx600 -output=obj="\$ (CONFIGDIR)¥\$(FILELEAF).obj" -debug -nologo (統合開発環境のデフォルト設定を使用しています)
iodef.h のバージョン	Version 1.0F
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルコードのバージョン	Version 1.00
使用ボード	Renesas Starter Kit for RX63T (製品型名: R0K50563TS000BE)

## 3. 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。

- RX63Tグループ 初期設定例 Rev.1.00 (R01AN1252JJ0100)

## 4. ハードウェア説明

### 4.1 使用端子一覧

表 4.1に使用端子と機能を示します。

表4.1 使用端子と機能

端子名	入出力	内容
PB3/MTIOC0A	出力	ステッピングモータ動作制御出力 (A相)
PB1/MTIOC0C	出力	ステッピングモータ動作制御出力 ( $\bar{A}$ 相)
P72/MTIOC4A	出力	ステッピングモータ動作制御出力 (B相)
P75/MTIOC4C	出力	ステッピングモータ動作制御出力 ( $\bar{B}$ 相)

5. ソフトウェア説明

5.1 動作概要

5.1.1 ステッピングモータの動作原理

a) 2 相励磁方式のステッピングモータの動作概要

2 相ステッピングモータのステップ角 7.5[deg/step]とした場合の 2 相励磁方式で動作させる例を図 5.1 に示します。

- 図 5.1 のようにパルスが High のとき、対応する相を励磁します。
  - まず、 $\bar{B}$  相と A 相を励磁します。このときロータは、 $\bar{B}$  相と A 相の間に位置します。
  - 次に、A 相と B 相を同時に励磁します。このときロータは、A 相と B 相の間に位置します。
- 以下、2 相励磁方式は、隣り合う 2 つの相 ( $\bar{B}$ 、A 相 A、B 相 B、 $\bar{A}$  相  $\bar{A}$ 、 $\bar{B}$  相) の順番に励磁し、ロータを回転させます。
- 逆転動作の場合は、 $\bar{A}$ 、 $\bar{B}$  相 B、 $\bar{A}$  相、A、B 相  $\bar{B}$ 、A 相の順番に励磁することでステッピングモータを回転させます。
  - 停止動作は、正転動作の最後の相または、逆転動作の最後の相を一定時間励磁し続けることで、ステッピングモータを停止させます。

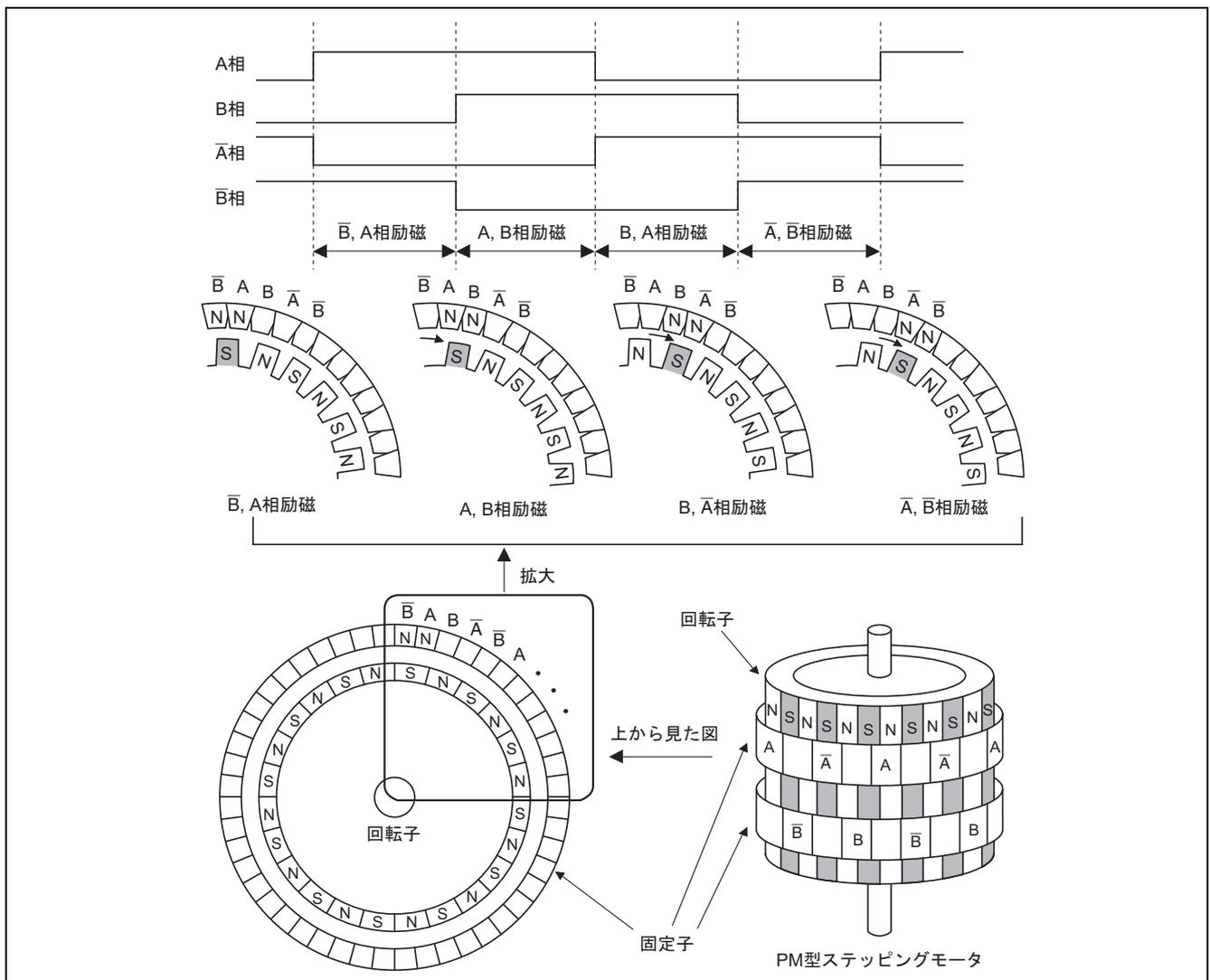


図5.1 ステッピングモータ動作例

## b) 貫通電流防止期間

励磁相の切り替え時に生じるターンオフの遅れにより、ドライバが破壊するのを防止するために貫通電流防止期間  $n$  (ノンオーバーラップ時間) を挿入します。

図 5.2 にノンオーバーラップ時間の出力例を示します。

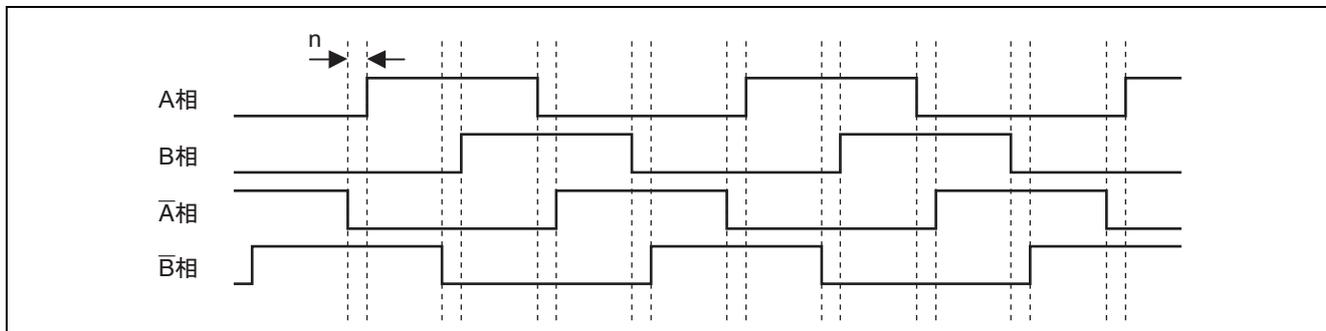


図5.2 ノンオーバーラップ時間の出力例

## c) ステッピングモータの速度制御

ステッピングモータを動作させる際に、急に周期の短いパルスを出力すると、モータはその負荷に追いつけず、回転しなくなります。これをモータの脱調といい、このモータの脱調を防止するため Slow Up、Slow Down 動作を行う必要があります。以下にステッピングモータの速度制御の動作原理を示します。

- ステッピングモータの回転速度を加速するために徐々にパルス周期を短くする。(Slow Up)
- ステッピングモータの回転速度を一定にするためにパルス周期を一定にする。(Constant)
- ステッピングモータの回転速度を減速するために徐々にパルス周期を長くする。(Slow Down)
- ステッピングモータの回転を停止するためにパルス周期を止めて出力状態を保持する。(Stop)

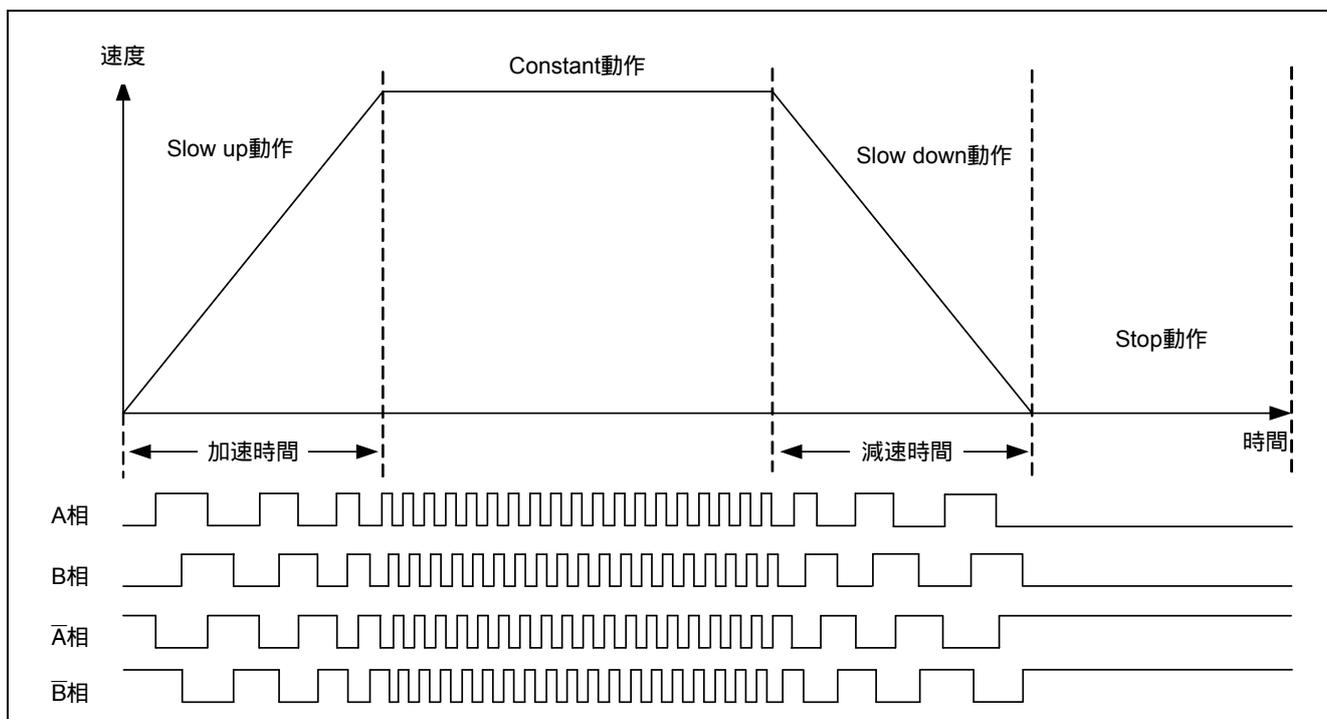


図5.3 ステッピングモータの速度制御の動作

## 5.1.2 ステッピングモータ制御パルス出力制御

本アプリケーションノートでは MTU3 の PWM 出力を使用し、DTC 転送により PWM 出力波形を制御することで、Slow Up、Slow Down を行います。また CMT で設定した時間の間、PWM 出力波形の継続または出力状態を保持することでステッピングモータの Constant と Stop を行います。

## a) ステッピングモータ制御パルスの出力設定

MTU3 のチャンネル 0 (以下 ch0) とチャンネル 4 (以下 ch4) を使用して、モータドライバに必要な A 相、 $\bar{A}$  相、B 相、 $\bar{B}$  相に相当するステッピングモータ制御パルスを出力します。

MTU3 の ch0 と ch4 を同期動作でかつ、それぞれのカウンタを ch0 の周期レジスタ (TGRA0) でクリアに設定し、ch0 と ch4 を同時スタートさせることで同期動作させます。

また、MTU3 の ch0 と ch4 を PWM モード 1 に設定し、表 5.1 に示したステッピングモータ制御パルスをそれぞれ出力します。

表 5.1 MTU3 の PWM 出力とステッピングモータ制御パルスの割り当て

チャンネル	出力端子	各 TGR の出力設定		内容
ch0	MTIOC0A 端子	TGRA0	1 出力	A 相のステッピングモータ制御パルス (PWM) を出力
		TGRB0	0 出力	
	MTIOC0C 端子	TGRC0	1 出力	$\bar{A}$ 相のステッピングモータ制御パルス (PWM) を出力
		TGRD0	0 出力	
ch4	MTIOC4A 端子	TGRA4	1 出力	B 相のステッピングモータ制御パルス (PWM) を出力
		TGRB4	0 出力	
	MTIOC4C 端子	TGRC4	1 出力	$\bar{B}$ 相のステッピングモータ制御パルス (PWM) を出力
		TGRD4	0 出力	

## b) ステッピングモータの Slow Up、Slow Down 制御と正転、逆転制御

ステッピングモータの Slow Up、Slow Down 制御は、ステッピングモータ制御パルスの PWM 波形を DTC 転送により MTU3 の各 TGR の値を更新することで徐々に PWM 波形の周期を変更します。

DTC 転送は、MTU3 の ch0 の周期レジスタ (TGRA0) のコンペアマッチを起動要因とし、図 5.4 で示した MTU3 の各 TGR (TGRA0~TGRD0、TGRA4~TGRD4) の計 8 レジスタに対し、転送要求毎にそれぞれ対応する周期データテーブルの値を順次 DTC のチェーン転送で転送します。

表 5.2 にそれぞれの DTC 転送先に対応した DTC 転送用周期データテーブルを示します。周期データテーブルの詳細については「5.8 データテーブル説明」を参照してください。

表 5.2 DTC 転送用周期データテーブル

MTU3 出力端子	DTC 転送先	DTC 転送用周期データテーブル名	
		正転時	逆転時
MTIOC0A 端子	TGRA0	cyctbl_A0	
	TGRB0	cyctbl_B0	
MTIOC0C 端子	TGRC0	cyctbl_C0	
	TGRD0	cyctbl_D0	
MTIOC4A 端子	TGRA4	cyctbl_A4	cyctbl_D4
	TGRB4	cyctbl_B4	cyctbl_C4
MTIOC4C 端子	TGRC4	cyctbl_C4	cyctbl_B4
	TGRD4	cyctbl_D4	cyctbl_A4

Slow Up 制御では、DTC 転送の転送元である各周期データテーブルの下位アドレスから転送要求毎にインクリメントして順次転送を行い、Slow Down 制御では転送元である各周期データテーブルの上位アドレスから転送要求毎にデクリメントして順次転送を行うことで PWM 出力波形の周期を徐々に可変することで実現します。

図 5.4 にステッピングモータ Slow Up 制御を例としたパルス出力例を示します。

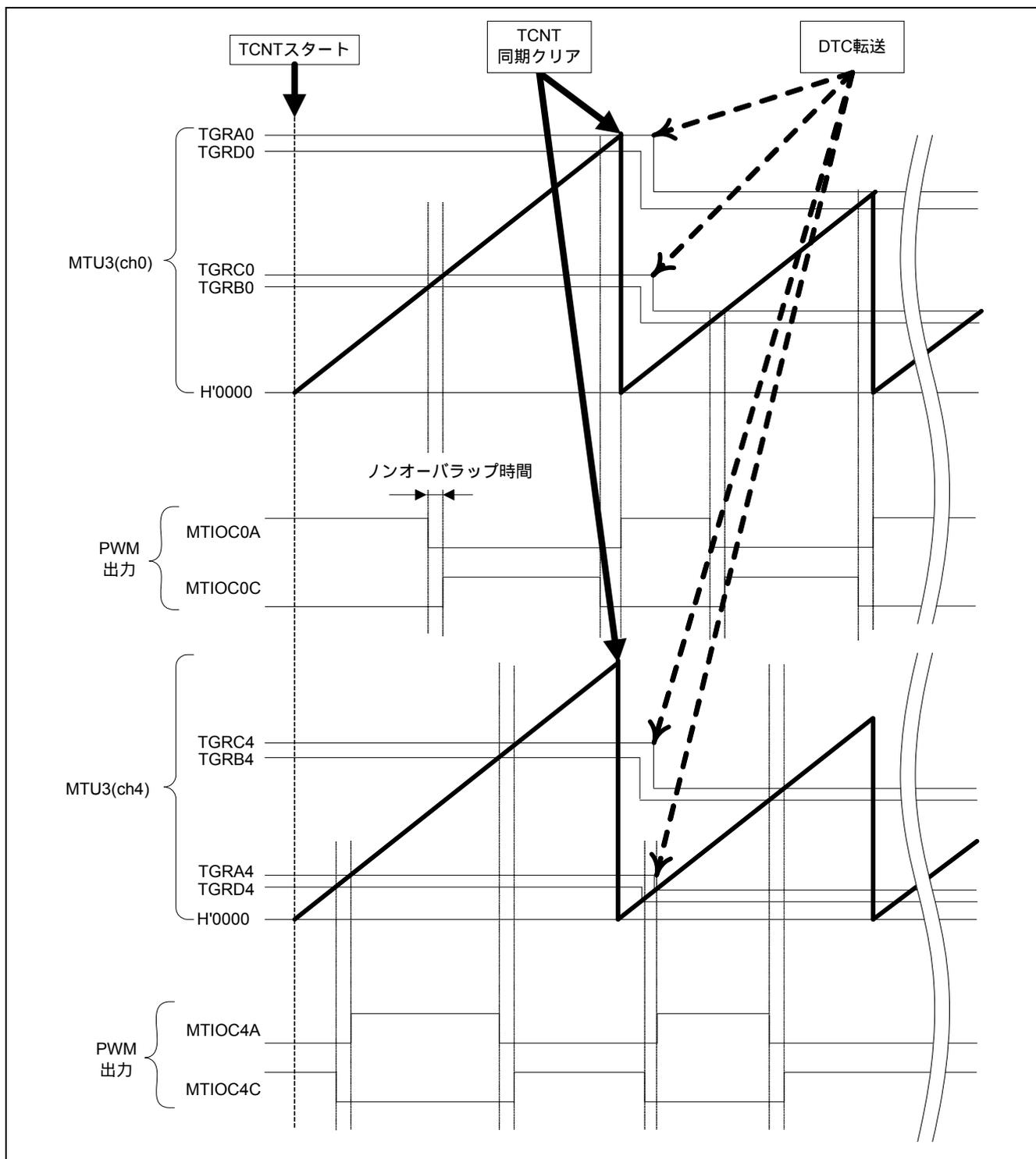


図5.4 ステッピングモータの Slow Up 制御

またステッピングモータの逆転は、DTC 転送元である B 相、 $\bar{B}$  相のステッピングモータ制御パルスの周期データテーブル ( cyctbl\_A4 と cyctbl\_D4、cyctbl\_C4 と cyctbl\_B4 ) の DTC の転送元アドレスを入れ替えることで B 相と  $\bar{B}$  相の出力波形を反転させることで実現します。

図 5.5 に正転、逆転時の DTC チェーン転送の詳細を示します

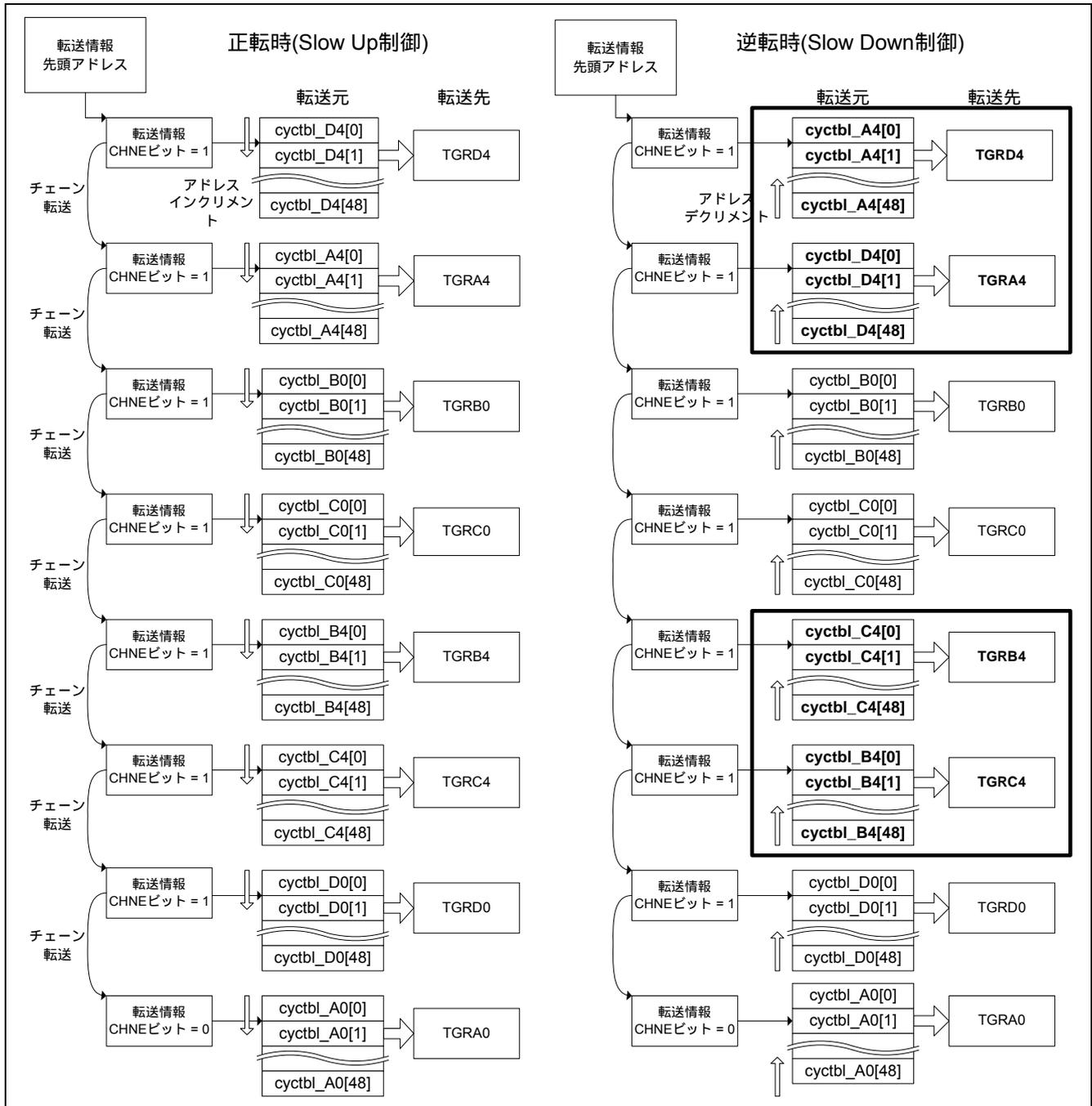


図5.5 チェーン転送

ステッピングモータの逆転は、MTU3 の TGRA4、TGRD4、および TGRB4、TGRC4 で使用した周期データテーブル ( cyctbl\_A4、cyctbl\_D4 および cyctbl\_C4、cyctbl\_B4 ) の DTC 転送元アドレスを入れ替えることで、正転時に出力していた B 相、 $\bar{B}$  相の波形を逆転させることで実現しています。

## c) ステッピングモータの Constant 制御と Stop 制御

ステッピングモータの Constant 制御は、Slow Up 制御終了時点での PWM 波形を継続することで実現します。

MTU3 の PWM 出力は Slow Up 制御終了時 (DTC 転送終了時に転送した各 TGR の値) の PWM 出力波形を継続して出力します。これを利用して本アプリケーションノートでは DTC 転送終了後に CMT をスタートさせて、CMT の割り込みが発生するまでこの PWM 出力波形 (Constant) を継続します。

CMT 割り込みが発生すると CMT 割り込み処理内で CMT 動作の停止とステッピングモータの Slow Down 制御に必要な DTC の再設定を行い、Slow Down 制御に移行します。

ステッピングモータの Stop 制御では、Slow Down 制御終了後 (DTC 転送終了時) に MTU3 の動作を停止することで実現します。

MTU3 の動作を停止すると、PWM 出力は MTU3 を停止した時点での出力状態を保持します。

これを利用してステッピングモータの Constant 制御同様、MTU3 停止後に CMT をスタートさせて、CMT の割り込みが発生するまでステッピングモータの Stop 状態を継続します。

CMT 割り込みが発生すると CMT 割り込み処理内で CMT 動作の停止、ステッピングモータの Slow Up 制御に必要な DTC の再設定および MTU3 の再スタート処理を行い、Slow Up 制御に移行します。

## 5.1.3 DTC の設定

本アプリケーションノートの DTC 転送条件を表 5.3 に示します。

表5.3 DTC の転送条件

条件	正(逆)転 Slow Up DTC 転送条件	正(逆)転 Slow Down DTC 転送条件
転送情報	フルアドレスモード	フルアドレスモード
転送モード	ノーマルモード	ノーマルモード
転送回数	49 回	49 回
転送データ	サイズ：ワード(2Byte)	サイズ：ワード(2Byte)
	データ内容：タイマジェネラルレジスタ (TGR)の値を変更	データ内容：タイマジェネラルレジスタ (TGR)の値を変更
転送元	内蔵 ROM	内蔵 ROM
転送先	タイマジェネラルレジスタ(TGR) MTU3 の TGRA ~ TGRD(ch0,4)	タイマジェネラルレジスタ(TGR) MTU3 の TGRA ~ TGRD(ch0,4)
転送元アドレス	転送後に転送元アドレスを インクリメント	転送後に転送元アドレスを デクリメント
転送先アドレス	転送先は固定	転送先は固定
起動要因	MTU3 の TGIA0 コンペアマッチ割り込み で起動	MTU3 の TGIA0 コンペアマッチ割り込み で起動
割り込み	DTC データ転送の度に、CPU への割り込み が発生	DTC データ転送の度に、CPU への割り込み が発生

## 5.2 必要メモリサイズ

表 5.4 に DTC テーブルを示します。

表5.4 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	4271 バイト	うち 784 バイトはデータテーブルで使用
RAM	2960 バイト	
最大使用ユーザスタック	52 バイト	
最大使用割り込みスタック	52 バイト	

【注】 必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。

## 5.3 ファイル構成

表 5.5 にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表5.5 サンプルコードで使用するファイル

ファイル名	概要	備考
r_init_stop_module.c	RX63T グループ 初期設定例のプログラム	詳細は RX63T グループ初期設定例のアプリケーションノートを参照してください。
r_init_stop_module.h		
r_init_clock.c		
r_init_clock.h		
r_init_non_existent_port.c		
r_init_non_existent_port.h		
main.c	2 相励磁ステッピングモータ動作の初期設定プログラム	
timer_int.c	割り込み処理プログラム	
motor.h	2 相励磁ステッピングモータ動作関連のインクルードヘッダ	
intprg.c	ベクタ関数の定義 MTU3 割り込み関数、CMT 割り込み関数を追加	

## 5.4 オプション設定メモリ

表 5.6 にサンプルコードで使用するオプション設定メモリの状態を示します。必要に応じて、お客様のシステムに最適な値を設定してください。

表5.6 サンプルコードで使用するオプション設定メモリ

シンボル	アドレス	設定値	内容
OFS0	FFFF FF8Fh ~ FFFF FF8Ch	FFFF FFFFh	リセット後、IWDT は停止 リセット後、WDT は停止
OFS1	FFFF FF8Bh ~ FFFF FF88h	FFFF FFFFh	リセット後、電圧監視 0 リセット無効
MDES	FFFF FF83h ~ FFFF FF80h	FFFF FFFFh FFFF FFF8h	(シングルチップモード時) リトルエンディアン ビッグエンディアン

【注】 \*1 本サンプルコードの設定はリトルエンディアンです。エンディアンの切り替えは6.2エンディアンの設定を参照ください。

## 5.5 定数一覧

表 5.7 にサンプルコードで使用する定数を示します。

表5.7 サンプルコードで使用する定数

定数名	設定値	内容
UPTIME	49	Slow Up/Slow Down 制御時の DTC 転送回数
CNSTTIME	48605	Constant 制御時の設定時間
NON_OVR	100	貫通電流防止期間(ノンオーバーラップ時間)
STOPTIME	48605	Stop の設定時間
CYCTIME	15000	データテーブルの周期設定時間
CHGTIME	200	データテーブルの周期変化値

## 5.6 構造体/共用体一覧

図 5.6にサンプルコードで使用する構造体/共用体を示します。

```
#pragma bit_order left
#pragma unpack

struct st_dtc_full{
  union{
    unsigned long LONG;
    struct{
      unsigned long MRA_MD      :2; /* MRA.MDビット */
      unsigned long MRA_SZ      :2; /* MRA.SZビット */
      unsigned long MRA_SM      :2; /* MRA.SMビット */
      unsigned long              :2;
      unsigned long MRB_CHNE    :1; /* MRB.CHNEビット */
      unsigned long MRB_CHNS    :1; /* MRB.CHNSビット */
      unsigned long MRB_DISEL    :1; /* MRB.DISELビット */
      unsigned long MRB_DTS     :1; /* MRB.DTSビット */
      unsigned long MRB_DM      :2; /* MRB.DMビット */
      unsigned long              :2;
      unsigned long              :16;
    }BIT;
  }MR;
  void * SAR; /* SARレジスタ */
  void * DAR; /* DARレジスタ */
  struct{
    unsigned long CRA:16; /* CRAレジスタ */
    unsigned long CRB:16; /* CRBレジスタ */
  }CR;
};

#pragma bit_order
#pragma packoption
```

図5.6 サンプルコードで使用する構造体/共用体

【注】 「#pragma unpack」で DTC 転送情報のアライメント数を 4 としています。

## 5.7 変数一覧

表 5.8にグローバル変数を、表 5.9にconst 型変数を、表 5.10に列挙型を示します。

表5.8 グローバル変数

型	変数名	内容	使用関数
st_dtc_full	dtc_tbl	MTU の TGRA0 ~ TGRD0、TGRA4 ~ TGRD4 にデータを転送するための構造体変数	dtc_init、fslowup0、fslowdown0、rslowup0、rslowdown0
void*	dtc_table[256]	DTC 転送情報の dtc_tbl[0]のアドレスを割り当てた DTC ベクタテーブル(開始アドレスは 0x00000000 に設定しています。)	dtc_init

表5.9 const 型変数

型	変数名	内容	使用関数
unsigned short	cyctbl_A0	TGR に転送するデータテーブル	dtc_init、 mtu3_init、 fslowup0、 fslowdown0、 rslowup0、 rslowdown0
unsigned short	cyctbl_A4		
unsigned short	cyctbl_B0		
unsigned short	cyctbl_B4		
unsigned short	cyctbl_C0		
unsigned short	cyctbl_C4		
unsigned short	cyctbl_D0		
unsigned short	cyctbl_D4		

表5.10 列挙型

型	変数名	内容	使用関数
motor_mode_t	nextmode0	モータの各制御を切り替える F_SLOWUP_MODE : 正転時 Slow Up 制御 F_CONSTANT_MODE : 正転時 Constant 制御 F_SLOWDOWN_MODE : 正転時 Slow Down 制御 F_STOP_MODE : Stop 制御 R_SLOWUP_MODE : 逆転時 Slow Up 制御 R_CONSTANT_MODE : 逆転時 Constant 制御 R_SLOWDOWN_MODE : 逆転時 Slow Down 制御 R_STOP_MODE : Stop 制御	main、 tgia0_int、 cmi0_int
motor_stop_seq_t;	fstop_seq	正転 Stop 移行フラグ STOP_DTCEND_SEQ : DTC 転送終了 STOP_MTUCMP_SEQ : コンペアマッチ発生終了	tgia0_int
motor_stop_seq_t;	rstop_seq	逆転 Stop 移行フラグ STOP_DTCEND_SEQ : DTC 転送終了 STOP_MTUCMP_SEQ : コンペアマッチ発生終了	tgia0_int
		DTC 転送番号(チェーン転送の順番) DTCTBL_MTU4_TGRD : 第 1 のデータ転送 DTCTBL_MTU4_TGRA : 第 2 のデータ転送 DTCTBL_MTU0_TGRB : 第 3 のデータ転送 DTCTBL_MTU0_TGRC : 第 4 のデータ転送 DTCTBL_MTU4_TGRB : 第 5 のデータ転送 DTCTBL_MTU4_TGRC : 第 6 のデータ転送 DTCTBL_MTU0_TGRD : 第 7 のデータ転送 DTCTBL_MTU0_TGRA : 第 8 のデータ転送	dtc_init、 rslowdown0、 rslowup0、 fslowdown0、 fslowup0

## 5.8 データテーブル説明

ステッピングモータ励磁出力するための周期データテーブルです。

本アプリケーションノートでは、タイマジェネラルレジスタ(TGR)のコンペアマッチにより、出力波形を変化させています。

cyctbl\_A0 と cyctbl\_B0 は、A 相の波形を出力するためのデータテーブルです。

cyctbl\_C0 と cyctbl\_D0 は、 $\bar{A}$  相の波形を出力するためのデータテーブルです。

cyctbl\_A4 と cyctbl\_B4 は、B 相の波形を出力するためのデータテーブルです。

cyctbl\_C4 と cyctbl\_D4 は、 $\bar{B}$  相の波形を出力するためのデータテーブルです。

下図は UPTIME = 49 の場合のデータテーブルとなります。UPTIME の値を変更される場合は計算式の追加 / 削除を行ってデータテーブルを変更してください。

```

cyctbl_A0[49] = {(4 * CYCTIME),
                 (4 * CYCTIME) - 1 * (4 * CHGTIME),
                 ...,
                 (4 * CYCTIME) - 48 * (4 * CHGTIME)};

cyctbl_B0[49] = {(2 * CYCTIME) - NON_OVR,
                 (2 * CYCTIME) - 1 * (2 * CHGTIME) - NON_OVR,
                 ...,
                 (2 * CYCTIME) - 48 * (2 * CHGTIME) - NON_OVR};

cyctbl_C0[49] = {(2 * CYCTIME),
                 (2 * CYCTIME) - 1 * (2 * CHGTIME),
                 ...,
                 (2 * CYCTIME) - 48 * (2 * CHGTIME)};

cyctbl_D0[49] = {(4 * CYCTIME) - NON_OVR,
                 (4 * CYCTIME) - 1 * (4 * CHGTIME) - NON_OVR,
                 ...,
                 (4 * CYCTIME) - 48 * (4 * CHGTIME) - NON_OVR};

cyctbl_A4[49] = {CYCTIME,
                 CYCTIME - 1 * CHGTIME,
                 ...,
                 CYCTIME - 48 * CHGTIME};

cyctbl_B4[49] = {(3 * CYCTIME) - NON_OVR,
                 (3 * CYCTIME) - 1 * (3 * CHGTIME) - NON_OVR,
                 ...,
                 (3 * CYCTIME) - 48 * (3 * CHGTIME) - NON_OVR};

cyctbl_C4[49] = {(3 * CYCTIME),
                 (3 * CYCTIME) - 1 * (3 * CHGTIME),
                 ...,
                 (3 * CYCTIME) - 48 * (3 * CHGTIME)};

cyctbl_D4[49] = {CYCTIME - NON_OVR,
                 CYCTIME - 1 * CHGTIME - NON_OVR,
                 ...,
                 CYCTIME - 48 * CHGTIME - NON_OVR};

```

図5.7 周期データテーブル

- 2 相励磁方式の 4 相パルス出力のデータテーブル作成方法  
速度を変化させるデータテーブルの作成方法を以下に記します。  
最初にコンペアマッチを発生させるタイマジェネラルレジスタ(TGR)の値を決めます。この時の値を「C」とします。  
次に、周期を変化させる値を決めます。この値を「T」とし、貫通電流防止期間の値を「N」とします。  
図 5.8 に速度を変化させた時の波形を示します。

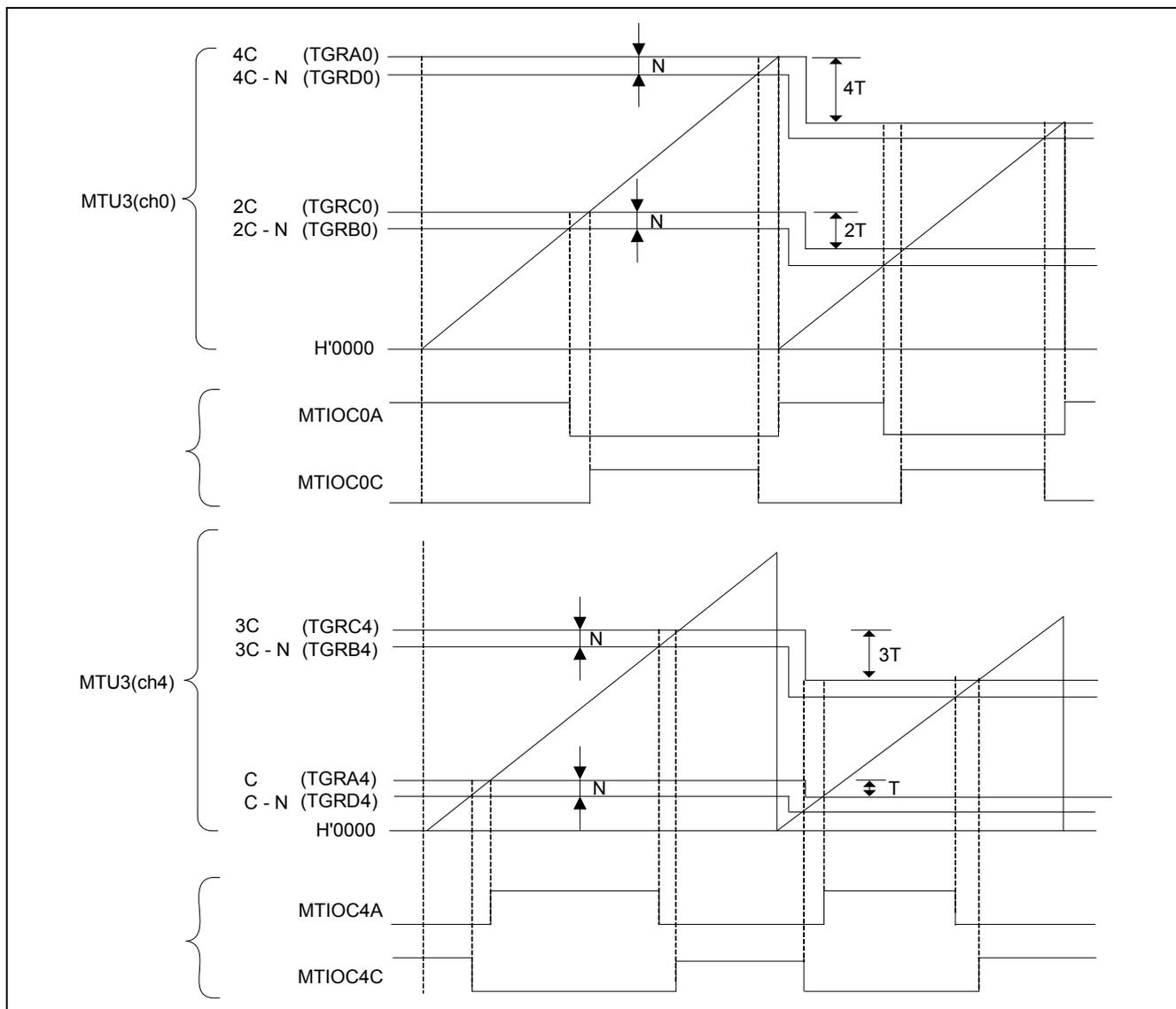


図5.8 周期データ変更

ここで記載しています C、T、N、n はサンプルコードではそれぞれ CYCTIME、CHGTIME、NON\_OVR、UPTIME と定義しています。

データテーブルは以下の計算式で作成することができます。n は配列の要素数です。(n = 0 ~ 48)

$$\text{cycTbl\_A4}[n] = (C - n \times T)$$

$$\text{cycTbl\_C0}[n] = (2C - n \times 2T)$$

$$\text{cycTbl\_C4}[n] = (3C - n \times 3T)$$

$$\text{cycTbl\_A0}[n] = (4C - n \times 4T)$$

貫通電流防止期間挿入用のデータテーブル

$$\text{cycctl\_D4}[n] = (C - n \times T) - N$$

$$\text{cycctl\_B0}[n] = (2C - n \times 2T) - N$$

$$\text{cycctl\_B4}[n] = (3C - n \times 3T) - N$$

$$\text{cycctl\_D0}[n] = (4C - n \times 4T) - N$$

本アプリケーションノートでの一例を示します。

最初にコンペアマッチする値は、cycctl\_A4[0]の 0x3A98(D'15000)となります。

C = 15000、T = 200、N = 100 と設定した場合、

$$\text{cycctl\_A4}[2] = 15000 - 2 \times 200 = 14600 \quad (n = 2)$$

貫通電流防止期間挿入用のデータテーブル

$$\text{cycctl\_D0}[47] = \{ (4 \times 15000) - 47 \times (4 \times 200) \} - 100 = 22300 \quad (n = 47)$$

となります。

上記の様に計算を行い、データテーブルを作成します。本アプリケーションノートで作成したデータテーブルの一例を図 5.9 に示します。

cycctl_A4[49]				cycctl_D0[49]			
要素数(n)	16進数(C)	10進数(C)	変更値(T)	要素数(n)	16進数(C - N)	10進数(C - N)	変更値(4T)
0	3A98	15000	-	0	E9FC	59900	-
1	39D0	14800	200	1	E6DC	59100	800
2	3908	14600	200	2	E3BC	58300	800
...	...	...	...	...	...	...	...
47	15E0	5600	200	47	571C	22300	800
48	1518	5400	200	48	53FC	21500	800

図5.9 Slow Up、Slow Down のデータテーブル設定

## 5.9 関数仕様

サンプルコードの関数仕様を示します。

HardwareSetup	
概要	初期化处理
ヘッダ	iodefine.h
宣言	void HardwareSetup(void)
説明	初期化处理の設定を行います。 <ul style="list-style-type: none"> <li>存在しないポートの初期設定</li> <li>システムクロック、周辺モジュールクロックの設定</li> <li>モジュールコントロールレジスタの設定</li> </ul>
引数	なし
リターン値	なし

main	
概要	メイン処理
ヘッダ	iodefine.h
宣言	void main(void)
説明	<ul style="list-style-type: none"> <li>正転 Constant 制御に設定</li> <li>DTC の初期設定</li> <li>MTU3 の初期設定</li> <li>CMT の初期設定</li> <li>MPC の初期設定</li> <li>PMR の初期設定</li> <li>ICU の初期設定</li> <li>タイマカウントスタート</li> </ul>
引数	なし
リターン値	なし

icu_init	
概要	ICU 初期設定
ヘッダ	iodefine.h
宣言	void icu_init(void)
説明	ICU の初期設定を行います。 <ul style="list-style-type: none"> <li>MTU3 の TGIA0 を DTC の起動要因に設定</li> <li>MTU3 の TGIA0 と CMI のチャンネル 0 の割り込み優先レベルの設定</li> <li>MTU3 の TGIA0 の割り込み要求許可レジスタ許可</li> </ul>
引数	なし
リターン値	なし

dtc_init	
概要	DTC 初期設定
ヘッダ	motor.h, iodefne.h
宣言	void dtc_init(void)
説明	DTC の初期設定を行います。 <ul style="list-style-type: none"> <li>• DTC 起動要因の転送情報アドレスを設定</li> <li>• DTC 転送情報は正転 Slow Up 制御用に設定</li> <li>• DTC ベクタベースレジスタを設定</li> <li>• DTC 転送情報を設定した後、DTC モジュール制御に設定</li> </ul>
引数	なし
リターン値	なし

mtu3_init	
概要	MTU3 の初期設定
ヘッダ	iodefne.h
宣言	void mtu3_init(void)
説明	MTU3 の初期設定を行います。 <ul style="list-style-type: none"> <li>• タイマのチャンネル 0 と 4 を同期動作に設定し、チャンネル 0 の TGRA コンペアマッチで TCNT0 と TCNT4 を同時クリア</li> <li>• チャンネル 0 と 4 を PWM モード 1 に設定し、波形の出力を行う</li> <li>• 正転 Slow Up 制御用に TIOR を設定</li> <li>• 周期データテーブルをチャンネル 0 の TGRA0 ~ TGRD0、チャンネル 4 の TGRA4 ~ TGRD4 に設定</li> <li>• MTU3 モジュールの TGIA の割り込み要求を許可し、チャンネル 0 とチャンネル 4 のタイマステータスフラグ (TSR) をクリア</li> </ul>
引数	なし
リターン値	なし

cmt0_init	
概要	CMT0 初期設定
ヘッダ	iodefne.h
宣言	void cmt0_init(void)
説明	CMT0 の初期設定を行います。 <ul style="list-style-type: none"> <li>• クロックの設定、コンペアマッチ割り込みを許可</li> </ul>
引数	なし
リターン値	なし

mpc_init	
概要	MPC 初期設定
ヘッダ	iodefine.h
宣言	void mpc_init(void)
説明	MPC を下記の機能に選択します。 <ul style="list-style-type: none"> <li>• PB3 → MTIOC0A</li> <li>• PB1 → MTIOC0C</li> <li>• P72 → MTIOC4A</li> <li>• P75 → MTIOC4C</li> </ul>
引数	なし
リターン値	なし

pmr_init	
概要	PMR 初期設定
ヘッダ	iodefine.h
宣言	void pmr_init(void)
説明	PMR の初期設定を行います。 <ul style="list-style-type: none"> <li>• PB1、PB3、P72、P75 を周辺機能として使用</li> </ul>
引数	なし
リターン値	なし

tgia0_int	
概要	TGIA0 割り込み
ヘッダ	motor.h, iodefine.h
宣言	void tgia0_int(void)
説明	TGIA0 割り込み内で Slow Up 制御または Slow Down 制御を終了します。 <ul style="list-style-type: none"> <li>• 割り込みに遷移した後、MTU3 のチャンネル 0 の TGFA フラグをクリア  (A)指定した DTC 転送回数が終了した場合、モータの制御を変更します。  モータ制御内容  (1)nextmode0 = F_CONSTANT_MODE の場合  正転 Constant 制御後、正転 Slow Down 制御に移行  (2)nextmode0 = F_STOP_MODE の場合  Stop 制御後、逆転 Slow Up 制御に移行  (3)nextmode0 = R_CONSTANT_MODE の場合  逆転 Constant 制御後、逆転 Slow Down 制御に移行  (4)nextmode0 = R_STOP_MODE の場合  Stop 制御後、正転 Slow Up 制御に移行  (B)指定した DTC 転送回数が終了しなかった場合、モータの制御は変更しません。</li> </ul>
引数	なし
リターン値	なし

cmi0_int	
概要	CMI0 割り込み
ヘッダ	motor.h, iodefne.h
宣言	void cmi0_int(void)
説明	CMT 割り込み処理内で Constant 制御または Stop 制御を終了します。 <ul style="list-style-type: none"> <li>• CMT0 のカウントをストップ モータの制御を変更します。 <ol style="list-style-type: none"> <li>(1) nextmode0 = F_SLOWUP_MODE の場合 正転 Slow Up 制御後、正転 Constant 制御に移行</li> <li>(2) nextmode0 = F_SLOWDOWN_MODE の場合 正転 Slow Down 制御後、Stop 制御に移行</li> <li>(3) nextmode0 = R_SLOWUP_MODE の場合 逆転 Slow Up 制御後、逆転 Constant 制御に移行</li> <li>(4) nextmode0 = R_SLOWDOWN_MODE の場合 逆転 Slow Down 制御後、Stop 制御に移行</li> </ol> </li> <li>• DTC モジュール動作に設定</li> <li>• ICU の IR フラグをクリア</li> <li>• MTU3 の TGIA0 割り込み要求許可レジスタを許可</li> </ul>
引数	なし
リターン値	なし
fslowup0	
概要	正転 Slow Up 制御
ヘッダ	iodefne.h
宣言	void fslowup0(void)
説明	DTC 転送情報を正転 Slow Up 制御用に設定します。 <ul style="list-style-type: none"> <li>• MTU3 の動作停止</li> <li>• TIOR の設定</li> <li>• DTC レジスタ転送情報を更新</li> <li>• MTU3 の動作開始</li> </ul>
引数	なし
リターン値	なし
frconst0	
概要	正(逆)転 Constant 制御
ヘッダ	iodefne.h
宣言	void frconst0(void)
説明	Constant 制御を行います。 <ul style="list-style-type: none"> <li>• CMT 動作を開始</li> <li>• MTU3 の TGIA0 割り込み要求許可レジスタを禁止</li> </ul>
引数	なし
リターン値	なし

---

**fslowdown0**

---

概要	正転 Slow Down 制御
ヘッダ	iodefine.h
宣言	void fslowdown0(void)
説明	DTC 転送情報を正転 Slow Down 制御用に設定します。 <ul style="list-style-type: none"><li>• DTC レジスタ転送情報を更新</li></ul>
引数	なし
リターン値	なし

---

**rslowup0**

---

概要	逆転 Slow Up 制御
ヘッダ	iodefine.h
宣言	void rslowup0 (void)
説明	DTC 転送情報を逆転 Slow Up 制御用に設定します。 <ul style="list-style-type: none"><li>• MTU3 の動作停止</li><li>• TIOR の設定</li><li>• DTC レジスタ転送情報を更新</li><li>• MTU3 の動作開始</li></ul>
引数	なし
リターン値	なし

---

**rslowdown0**

---

概要	逆転 Slow Down 制御
ヘッダ	iodefine.h
宣言	void rslowdown0(void)
説明	DTC 転送情報を逆転 Slow Down 制御用に設定します。 <ul style="list-style-type: none"><li>• DTC レジスタ転送情報を更新</li></ul>
引数	なし
リターン値	なし

---

**frstop0**

---

概要	Stop 制御
ヘッダ	iodefine.h
宣言	void frstop0(void)
説明	Stop 制御を行います。 <ul style="list-style-type: none"><li>• MTU3 の動作停止</li><li>• CMT0 の動作開始</li><li>• MTU3 の TGIA0 割り込み要求許可レジスタを禁止</li></ul>
引数	なし
リターン値	なし

---

CmtStart	
概要	CMT 動作開始
ヘッダ	iodefine.h
宣言	void CmtStart(unsigned short cmt_cnt)
説明	CMT 動作の開始を行います。 <ul style="list-style-type: none"><li>• CMIO の割り込み要求許可レジスタを許可</li><li>• CMCOR の時間設定</li><li>• CMT0 動作開始</li></ul>
引数	第一引数 : cmt_cnt            Constant 制御時間または Stop 制御時間
リターン値	なし

---

CmtStop	
概要	CMT 動作停止
ヘッダ	iodefine.h
宣言	void CmtStop(void)
説明	CMT 動作を停止します。 <ul style="list-style-type: none"><li>• CMT0 動作停止</li><li>• CMIO の割り込み要求許可レジスタを禁止</li></ul>
引数	なし
リターン値	なし

---

5.10 フローチャート

5.10.1 初期化処理

図 5.10に初期化処理のフローチャートを示します。

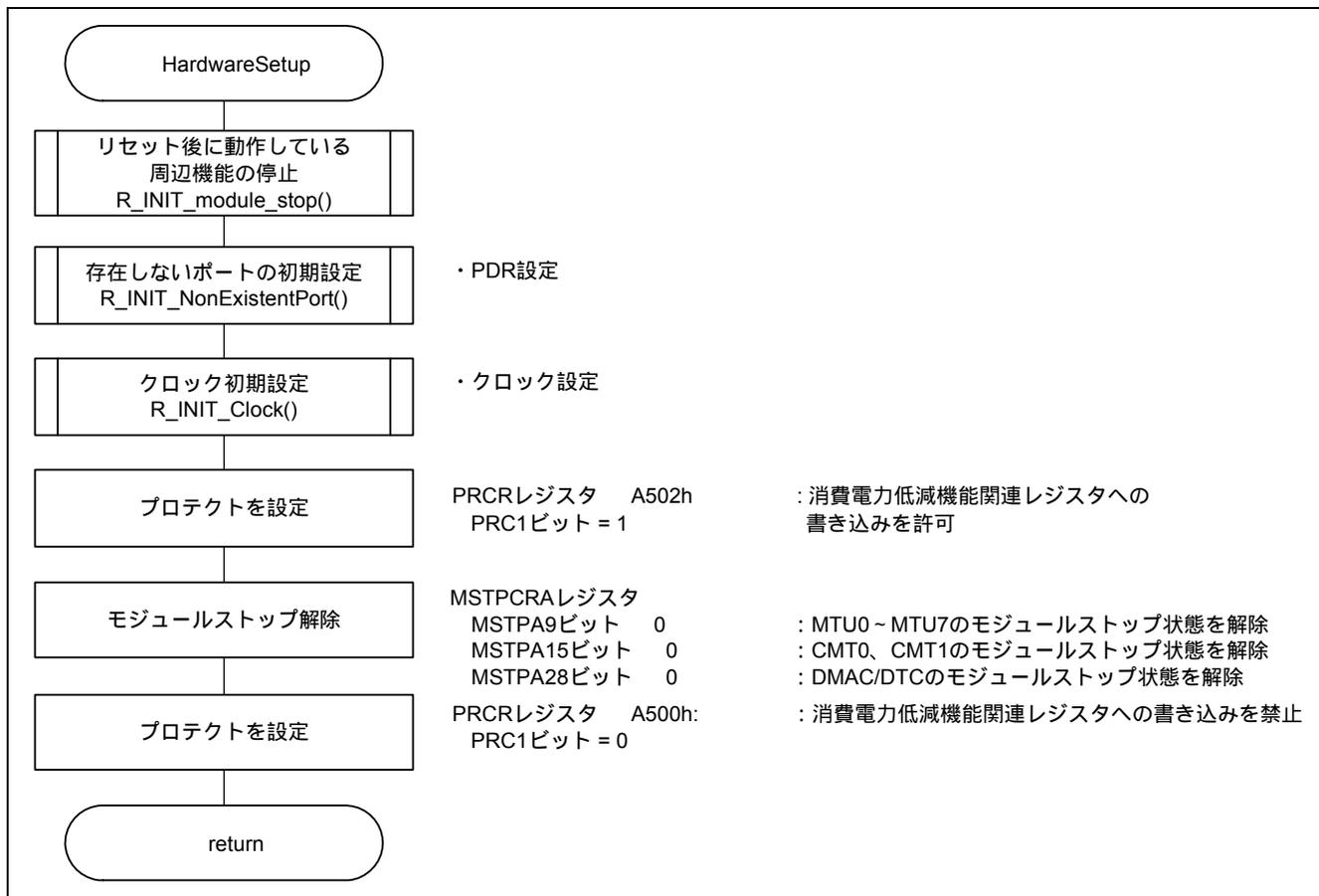


図5.10 初期化処理

5.10.2 メイン処理

図 5.11にメイン処理のフローチャートを示します。

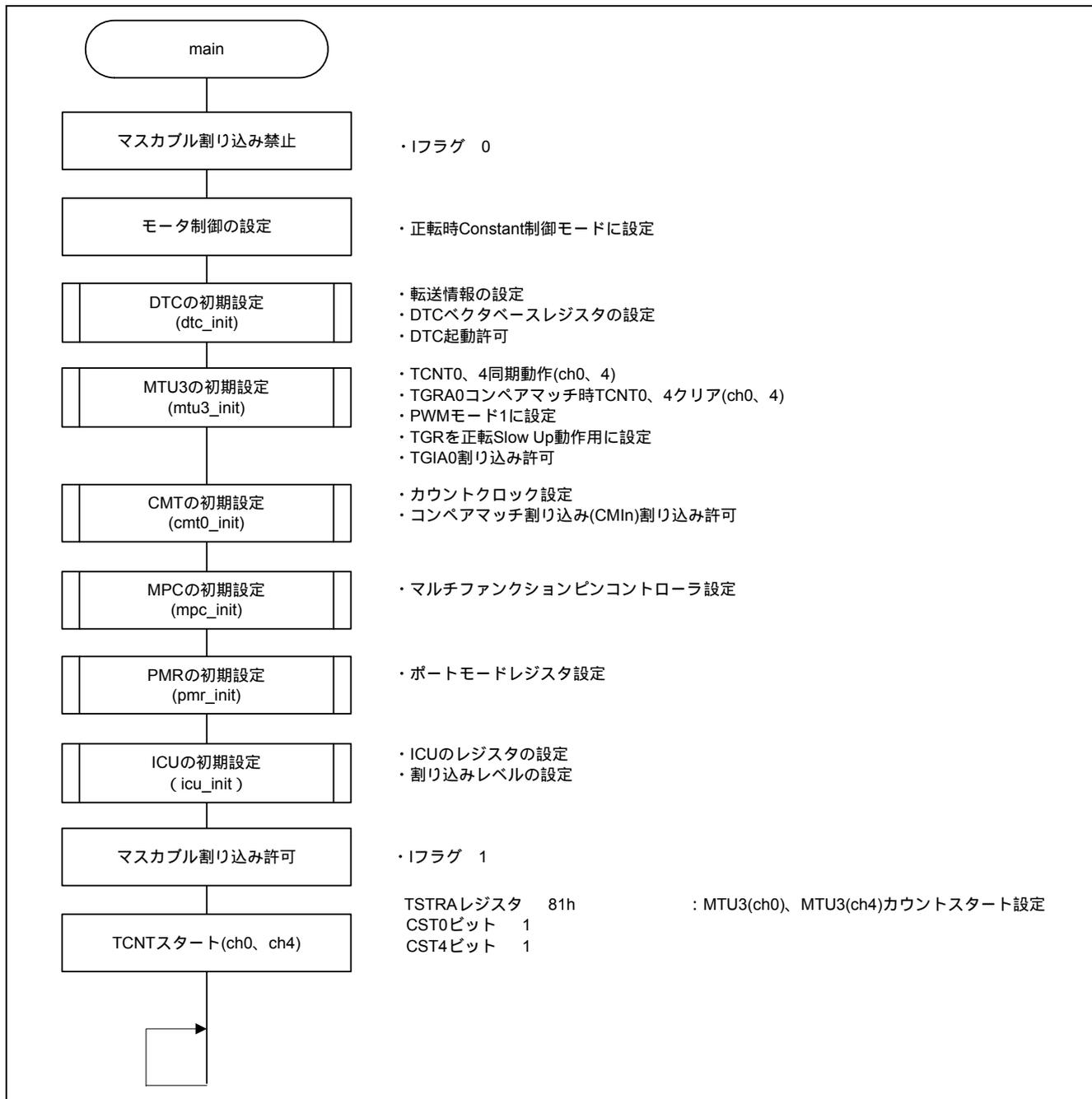


図5.11 メイン処理

5.10.3 ICU 初期設定

図 5.12に ICU 初期設定のフローチャートを示します。

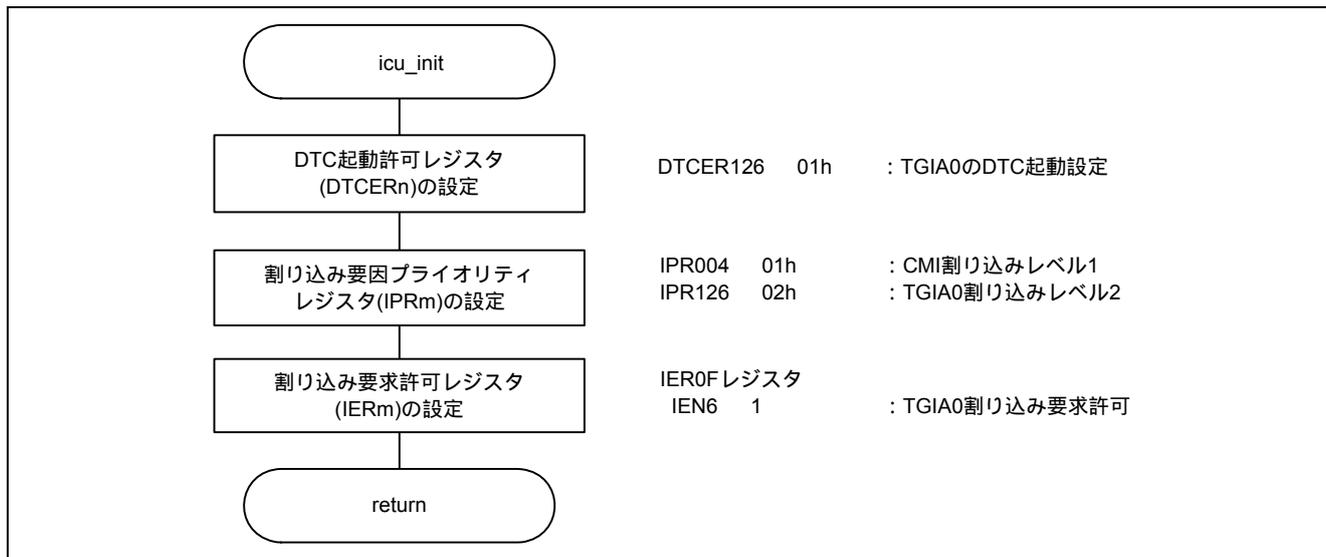


図5.12 ICU 初期設定

5.10.4 DTC 初期設定

図 5.13に DTC 初期設定を示します。

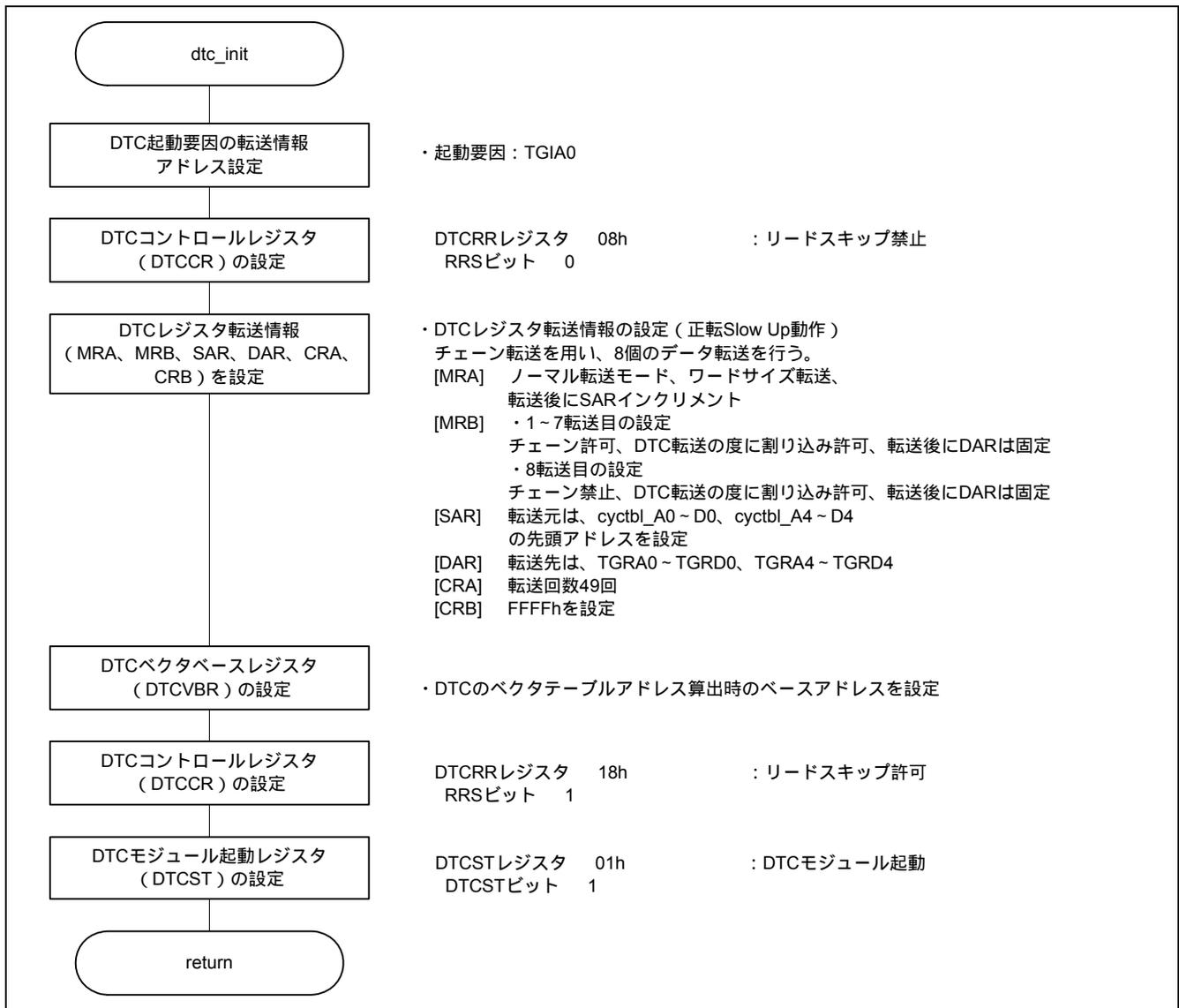


図5.13 DTC 初期設定

5.10.5 MTU3 初期設定

図 5.14に MTU3 初期設定のフローチャートを示します。

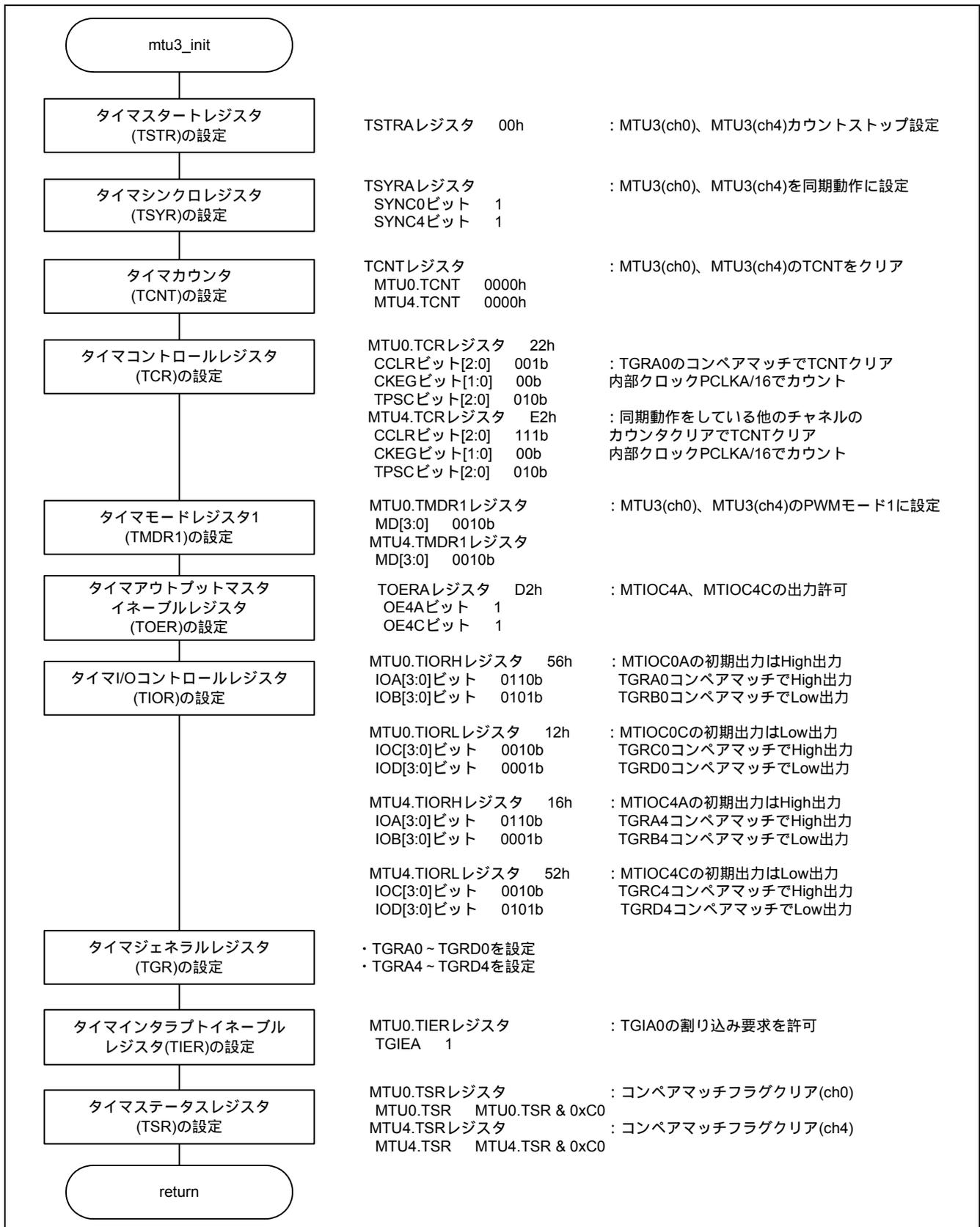


図5.14 MTU3 初期設定

5.10.6 CMT0 初期設定

図 5.15に CMT0 初期設定のフローチャートを示します。

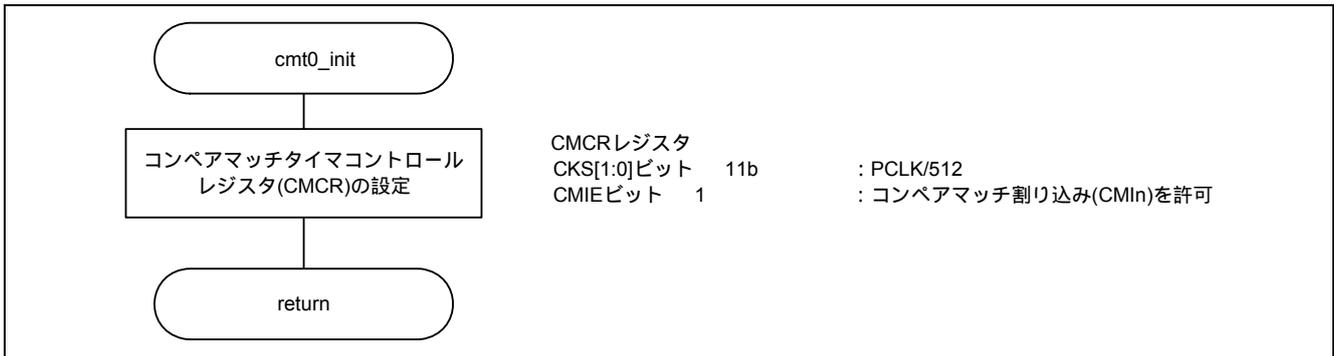


図5.15 CMT0 初期設定

5.10.7 MPC 初期設定

図 5.16 に MPC 初期設定のフローチャートを示します。

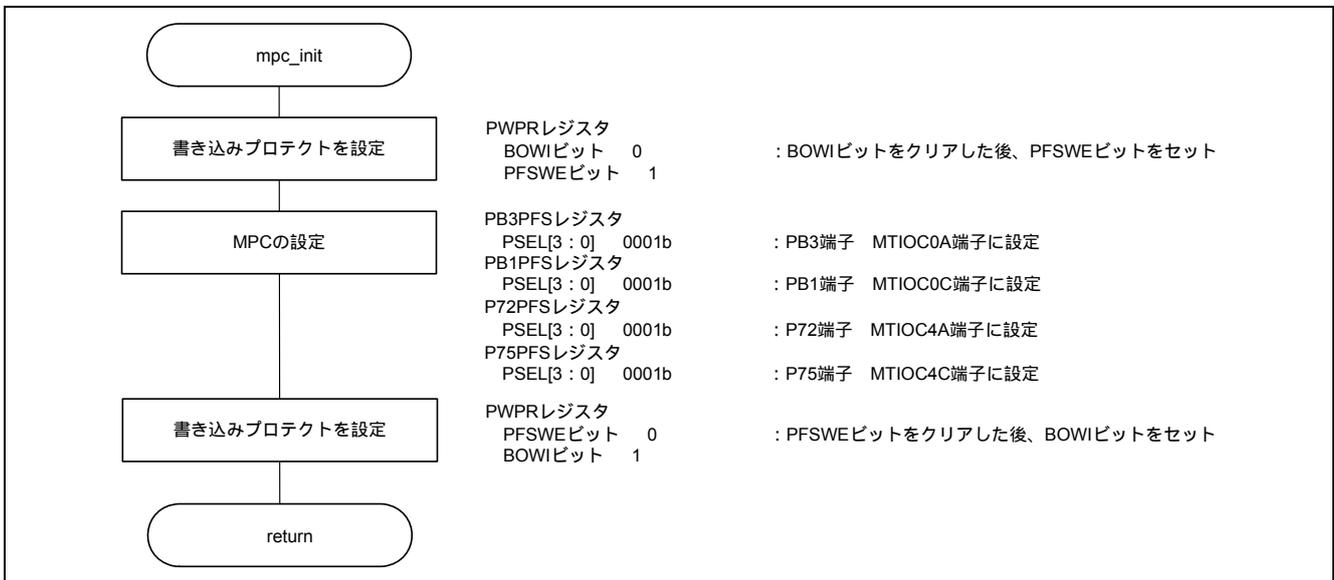


図5.16 MPC 初期設定

5.10.8 PMR 初期設定

図 5.17 に PMR 初期設定のフローチャートを示します。

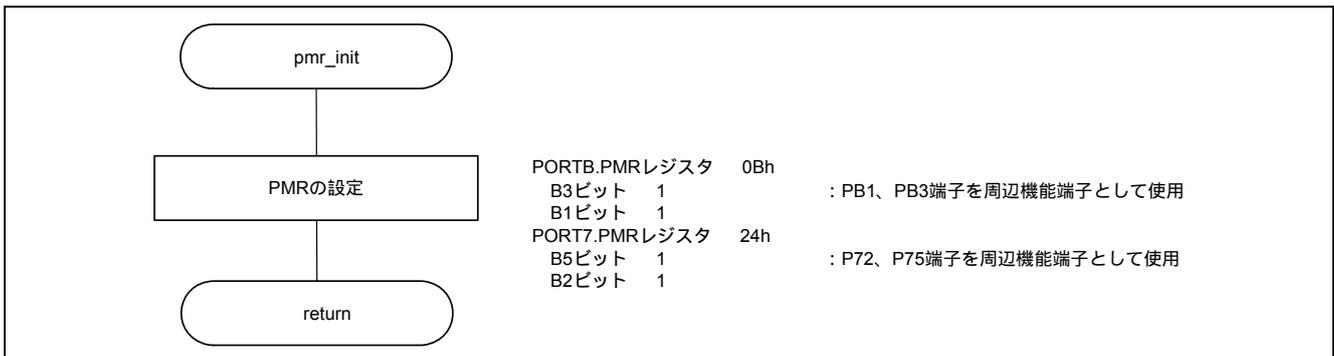


図5.17 PMR 初期設定

5.10.9 TGIA0 割り込み処理

図 5.18に TGIA0 割り込み処理フローチャートを示します。

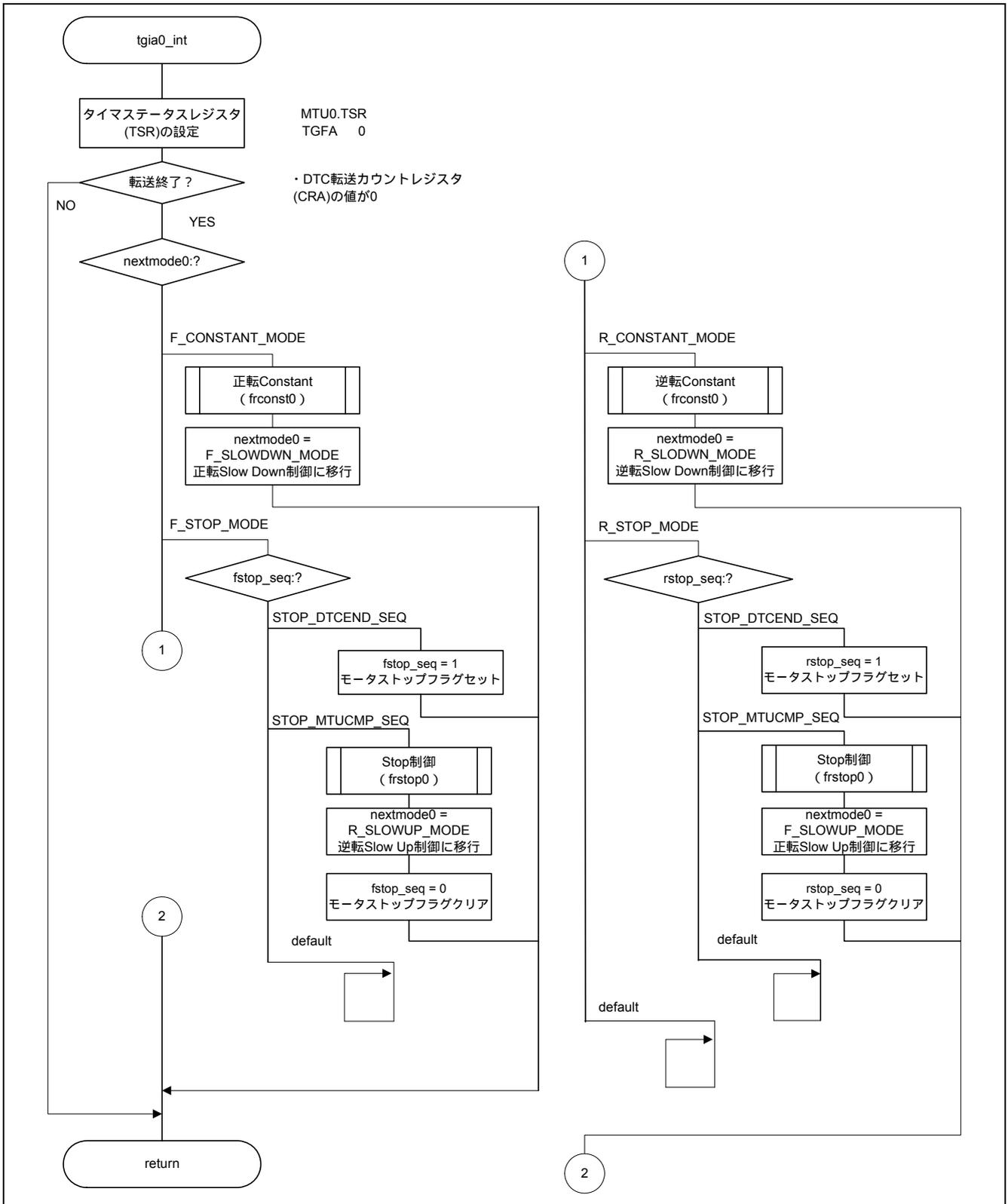


図5.18 TGIA0 割り込み

5.10.10 CMI0 割り込み

図 5.19に CMI0 割り込み処理フローチャートを示します。

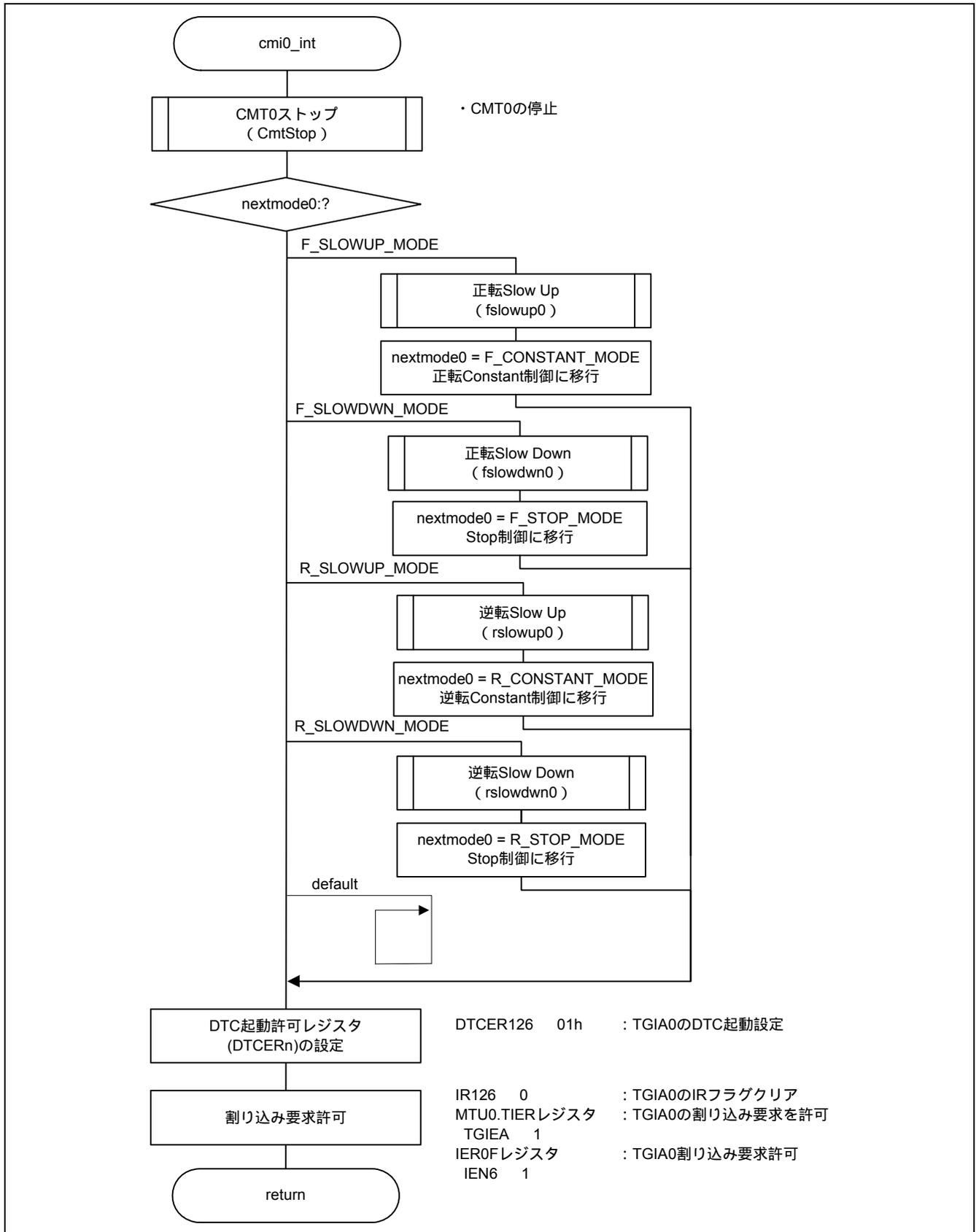


図5.19 CMI0 割り込み

5.10.11 正転 Slow Up 処理

図 5.20に正転 Slow Up 処理のフローチャートを示します。

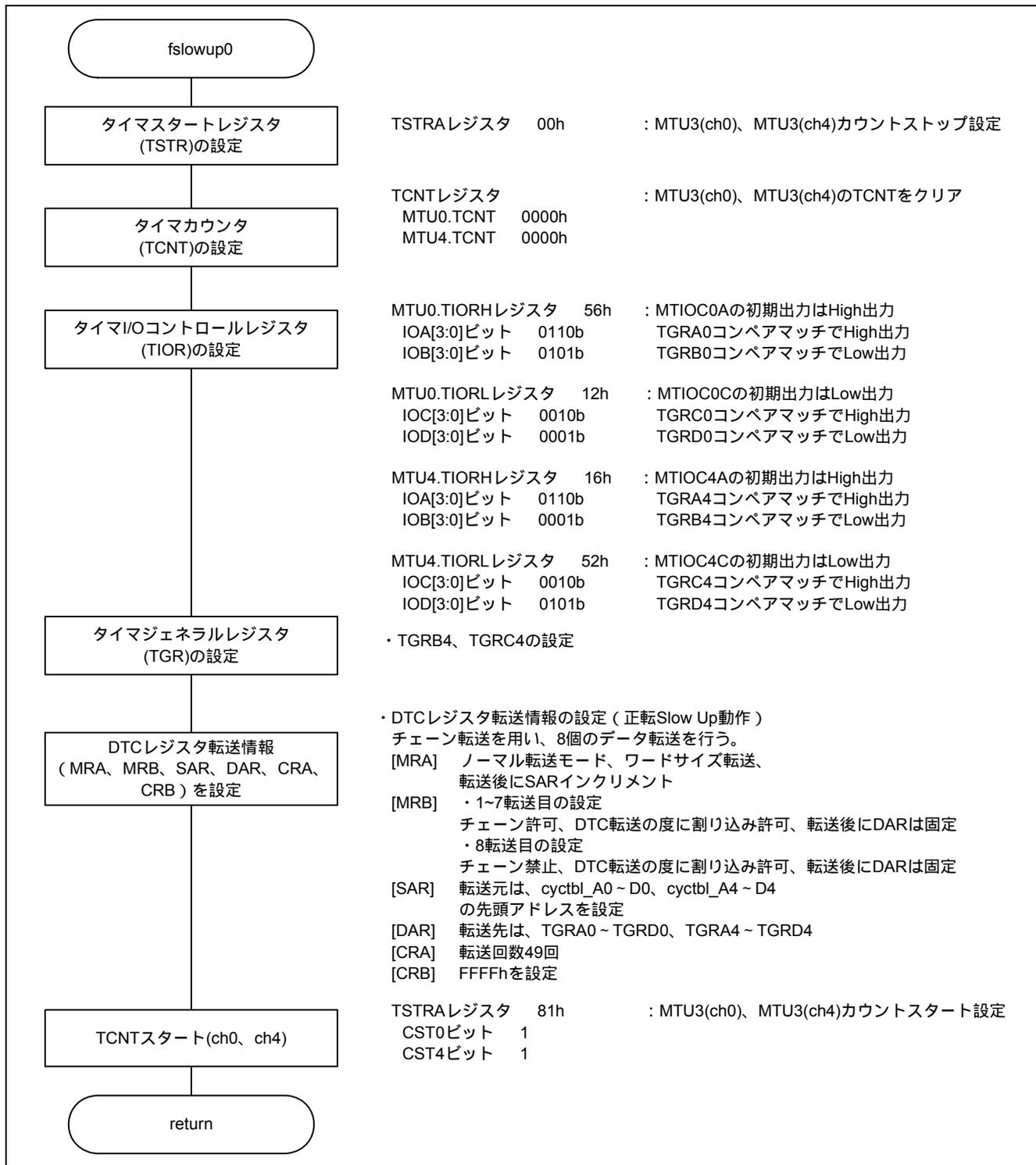


図5.20 正転 Slow Up 処理

## 5.10.12 正逆転 Constant 処理

図 5.21に正逆転 Constant 処理のフローチャートを示します。

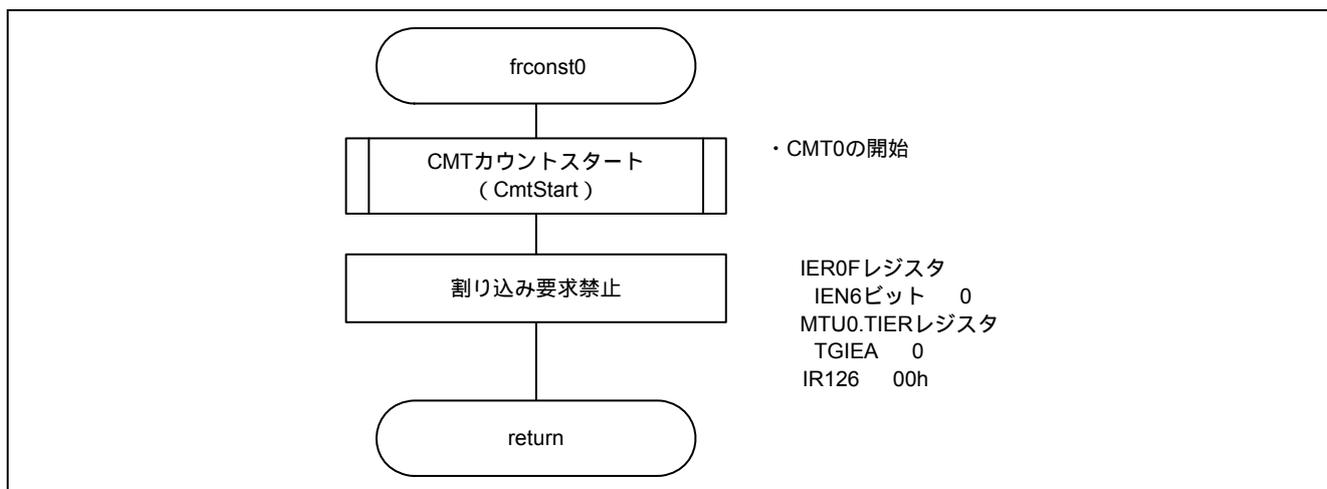


図5.21 正逆転 Constant 処理

## 5.10.13 正転 Slow Down 処理

図 5.22に正転 Slow Down 処理のフローチャートを示します。

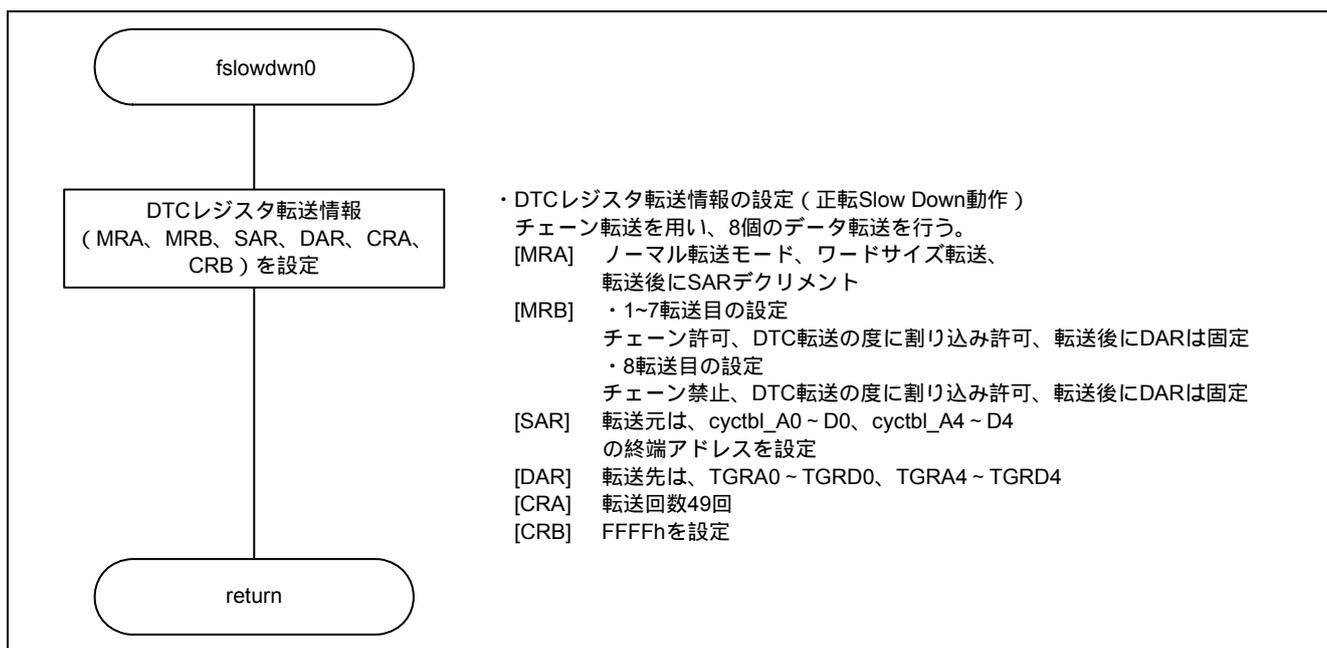


図5.22 正転 Slow Down 処理

5.10.14 逆転 Slow Up 処理

図 5.23に逆転 Slow Up 処理のフローチャートを示します。

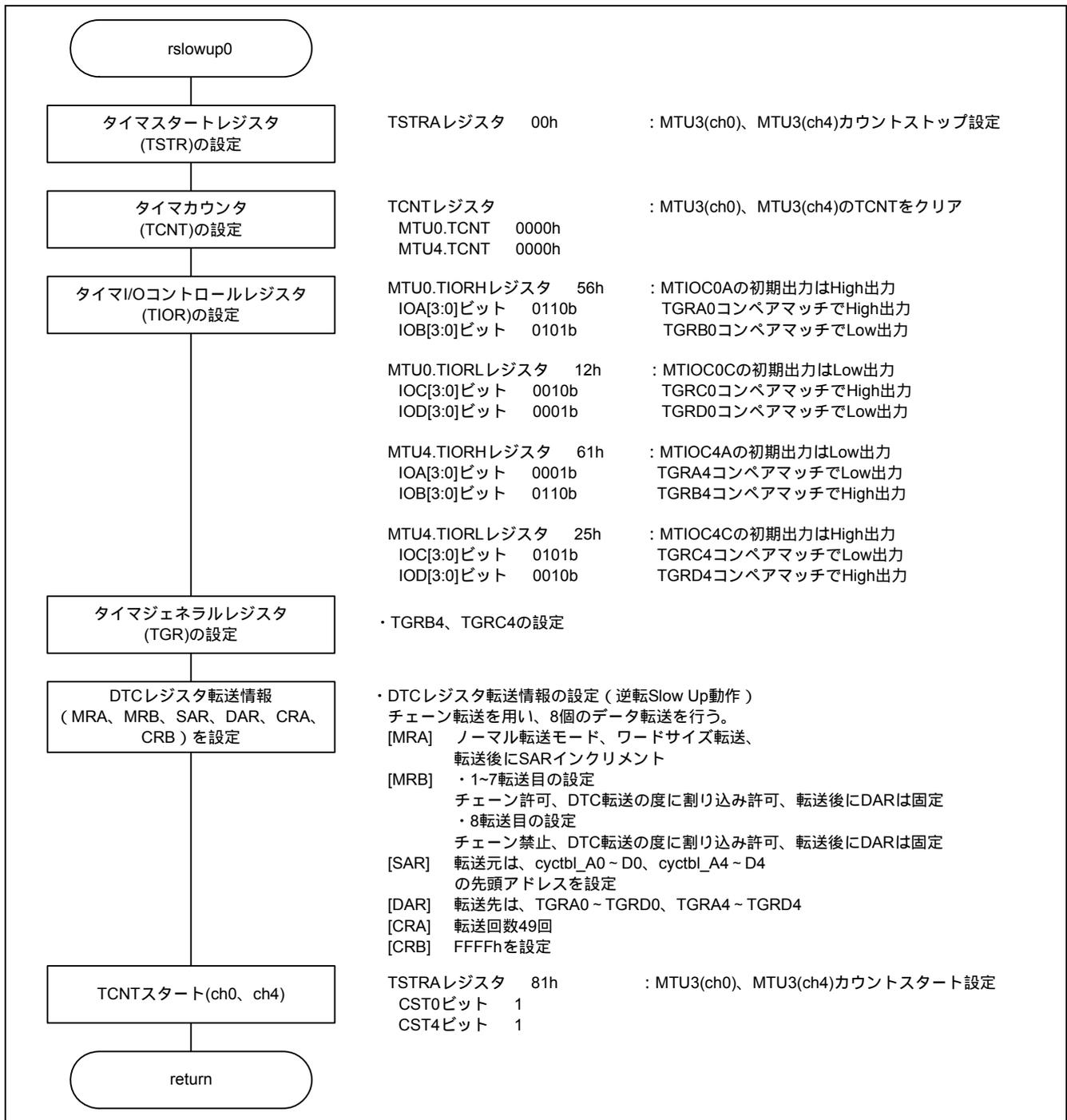


図5.23 逆転 Slow Up 処理

5.10.15 逆転 Slow Down 処理

図 5.24に逆転 Slow Down 処理のフローチャートを示します。

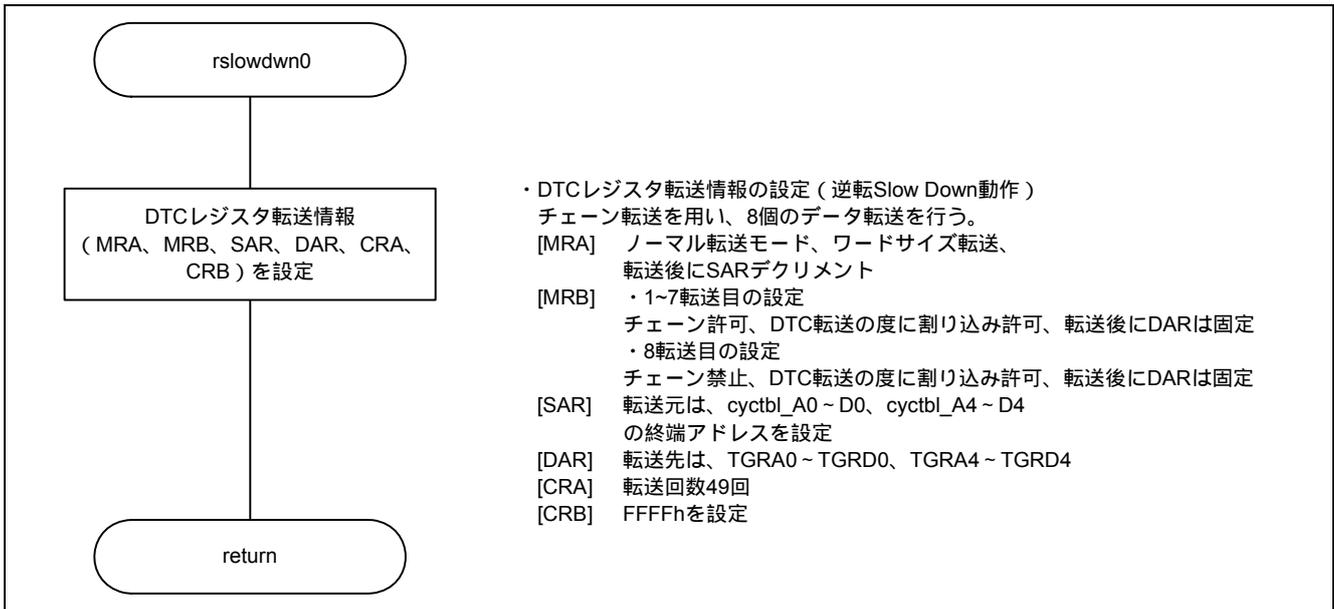


図5.24 逆転 Slow Down 処理

5.10.16 Stop 処理

図 5.25に Stop 処理のフローチャートを示します。

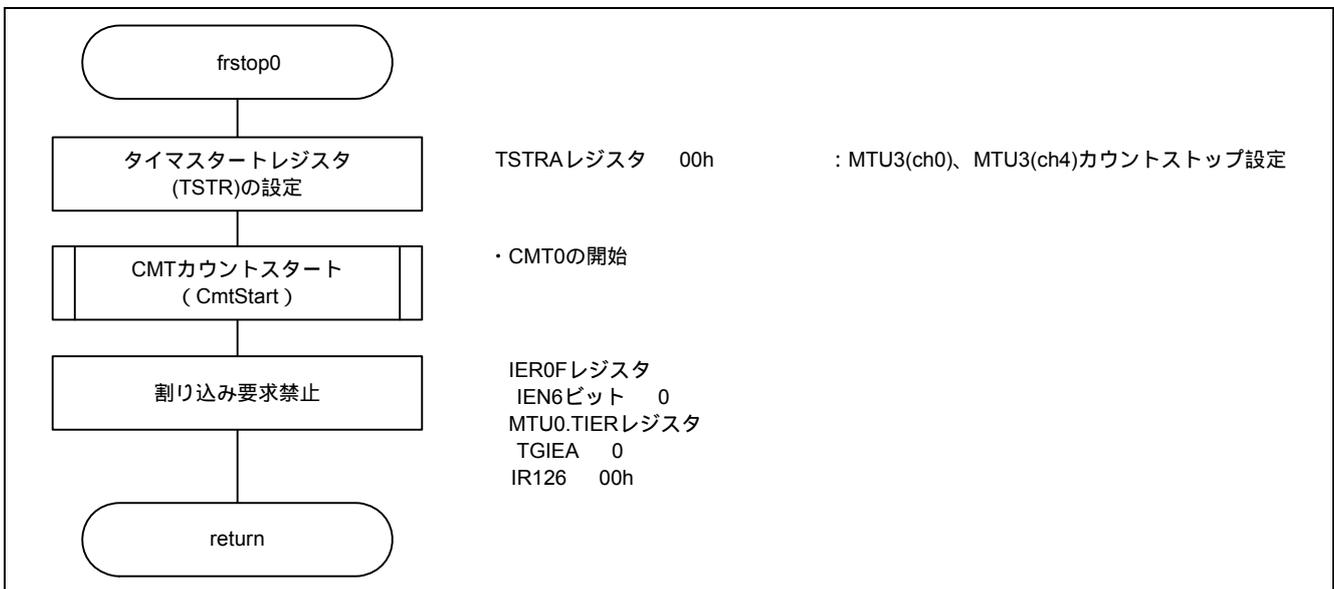


図5.25 STOP 処理

5.10.17 CMT スタート処理

図 5.26に CMT スタート処理のフローチャートを示します。

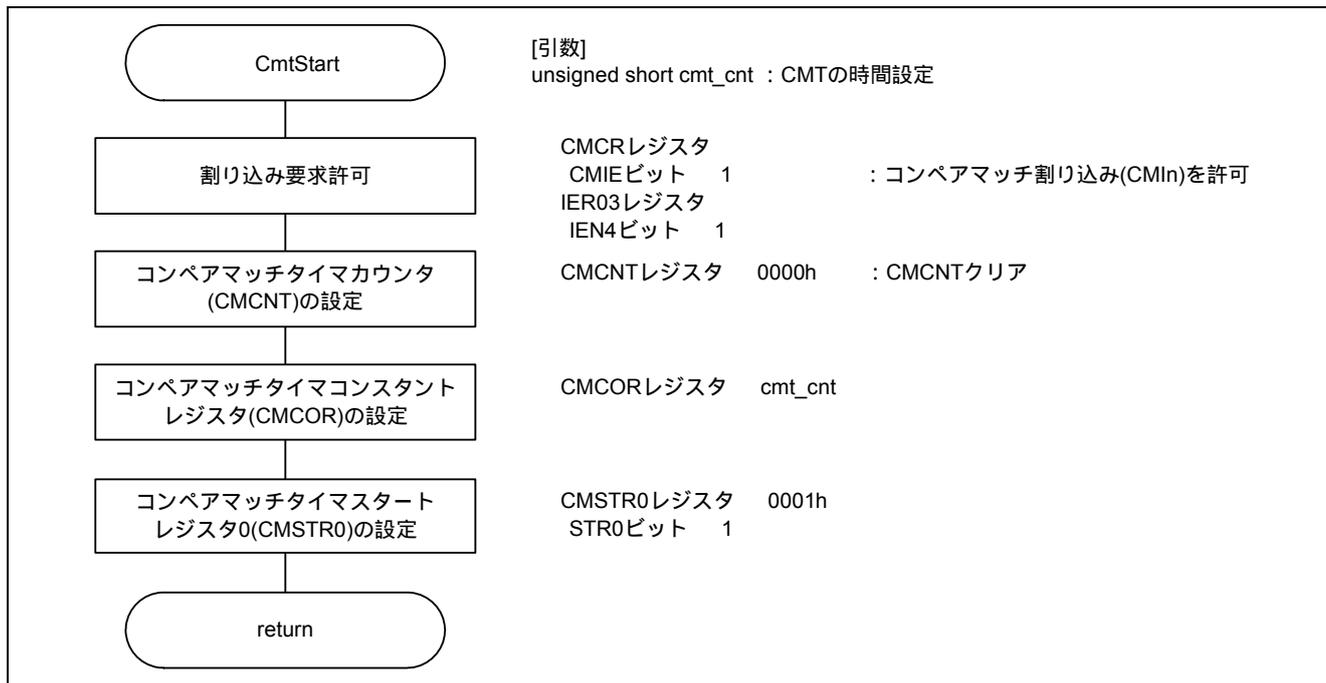


図5.26 CMT 動作開始

5.10.18 CMT ストップ処理

図 5.27に CMT ストップ処理のフローチャートを示します。

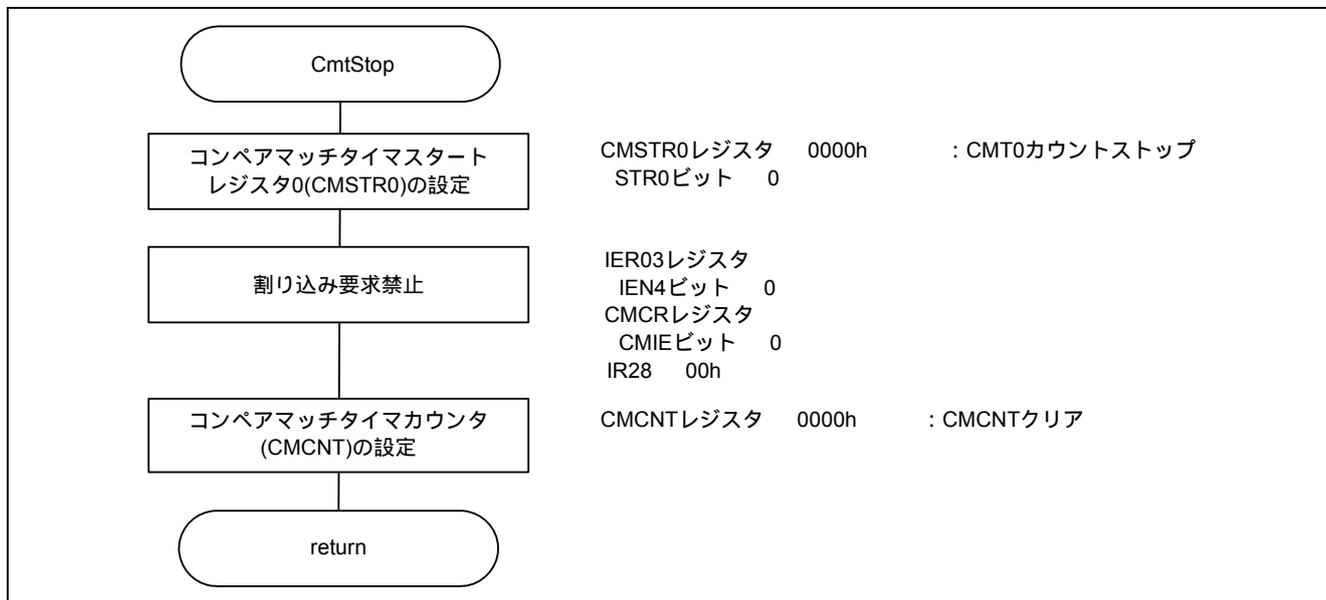


図5.27 CMT 動作停止

## 6. 注意事項

### 6.1 動作モードの設定

本アプリケーションノートでは、モード端子を MD=High に設定し動作モードをシングルチップモードに、システムコントロールレジスタ 0(SYSCR0)の ROME ビットを 1 に設定し内蔵 ROM を有効にそれぞれ設定しています。

本アプリケーションノートにおける動作モードの設定を表 6.1 に示します。

表6.1 動作モードの設定

モード設定端子	SYSCR0 レジスタ	動作モード	内蔵 ROM
MD	ROME		
High	1	シングルチップモード	有効

【注】 SYSCR0 レジスタの ROME ビットの初期値は、SYSCR0.ROME=1 のため、プログラム中で SYSCR0 レジスタの設定は行っておりません。

### 6.2 エンディアンの設定

本アプリケーションノートは、ビッグエンディアン / リトルエンディアンの両方に対応しています。

#### 6.2.1 リトルエンディアン使用時

リトルエンディアンで動作する場合は以下の設定を行ってください。

コンパイラオプションのエンディアンの設定で “ Little-endian データ ” を指定してください。5.4 オプション設定メモリの MDES はリトルエンディアンの値になります。

#### 6.2.2 ビッグエンディアン使用時

ビッグエンディアンで動作する場合は以下の設定を行ってください。

コンパイラオプションのエンディアンの設定で “ Big-endian データ ” を指定してください。5.4 オプション設定メモリの MDES はビッグエンディアンの値になります。

## 7. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

## 8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RX63Tグループ ユーザーズマニュアル ハードウェア編 Rev.1.00 (R01UH0238JJ)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：開発環境

RX ファミリ C/C++コンパイラパッケージ V.1.01 ユーザーズマニュアル Rev.1.00( V.1.02 添付資料含む)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

アプリケーションノート RX63Tグループ 初期設定例 Rev.1.00

(最新版をルネサス エレクトロニクス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>

お問い合わせ先

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

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	2014.05.14	—	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、本ドキュメントおよびテクニカルアップデートを参照してください。

### 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. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っていません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。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。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所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 エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<http://japan.renesas.com/contact/>