

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

お客様各位

資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ (<http://www.renesas.com>)

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

ご注意

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

アプリケーションノート

相補 PWM3 相出力

要旨

H8/3687 に内蔵しているタイマ Z の相補 PWM モードを使用して相補 PWM 波形（正相・逆相）を 3 相出力します。

動作確認デバイス

H8/300 Tiny シリーズ—H8/3687—

目次

ご注意.....	2
1. 仕様.....	3
2. 使用機能説明.....	4
3. 動作説明.....	6
4. ソフトウェア説明.....	10
4.1 モジュール説明.....	10
4.2 引数の説明.....	10
4.3 使用内部レジスタ説明.....	11
4.4 使用 RAM 説明.....	11
5. フローチャート.....	12
6. プログラムリスト.....	14

ご注意

1. 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
2. 本書に記載された情報の使用に際して、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 製品及び製品仕様は予告無く変更する場合がありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書をお求めになりご確認ください。
4. 弊社は品質・信頼性の向上に努めておりますが、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は、事前に弊社営業担当迄ご相談をお願い致します。
5. 設計に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件及びその他諸条件につきましては、弊社保証範囲内でご使用いただきますようお願い致します。保証値を越えてご使用された場合の故障及び事故につきましては、弊社はその責を負いません。また保証値内のご使用であっても半導体製品について通常予測される故障発生率、故障モードをご考慮の上、弊社製品の動作が原因でご使用機器が人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
6. 本製品は耐放射線設計をしておりません。
7. 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
8. 本書をはじめ弊社半導体についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

Copyright©Hitachi, Ltd., 2003. All rights reserved.

1. 仕様

1. 図 1.1 に示すように正相・逆相がノンオーバーラップの関係にある PWM 波形（正相・逆相）を 3 相出力します。
2. デューティはレジスタを設定することにより 0%~100%まで任意に変更できます。

$$\text{デューティ} = \frac{\text{パルスの High 幅}}{\text{パルスの周期}} \times 100\%$$

3. 周期に同期したトグル波形出力を行います。

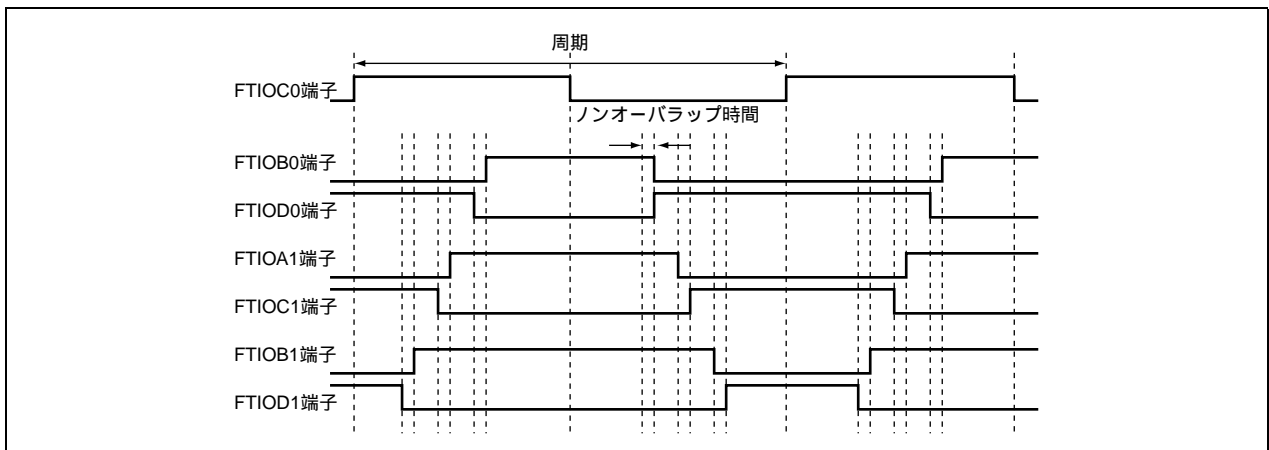


図 1.1 相補 PWM（正相・逆相）3 相出力波形

2. 使用機能説明

1. 本タスク例では TimerZ のチャンネル 0、1 を使用して正相と逆相がノンオーバーラップの関係にある PWM 波形を 3 相出力します。また、PWM 波形の周期に同期したトグル波形出力をします。

(a) 図 2.1 に本タスク例で使用するタイマ Z のブロック図を示します。

また本タスクでは、以下の機能を使用します。

- 正相と逆相がノンオーバーラップの関係にある TimerZ の PWM 波形を 3 相出力する機能（相補 PWM モード）
- コンペアマッチ発生時、バッファレジスタ（GRC0/D0、GRC1/D1）の内容をコンペアレジスタ（GRA0/B0、GRA1/B1）に転送する機能。
- PWM 波形の周期に同期したトグル波形を出力する機能

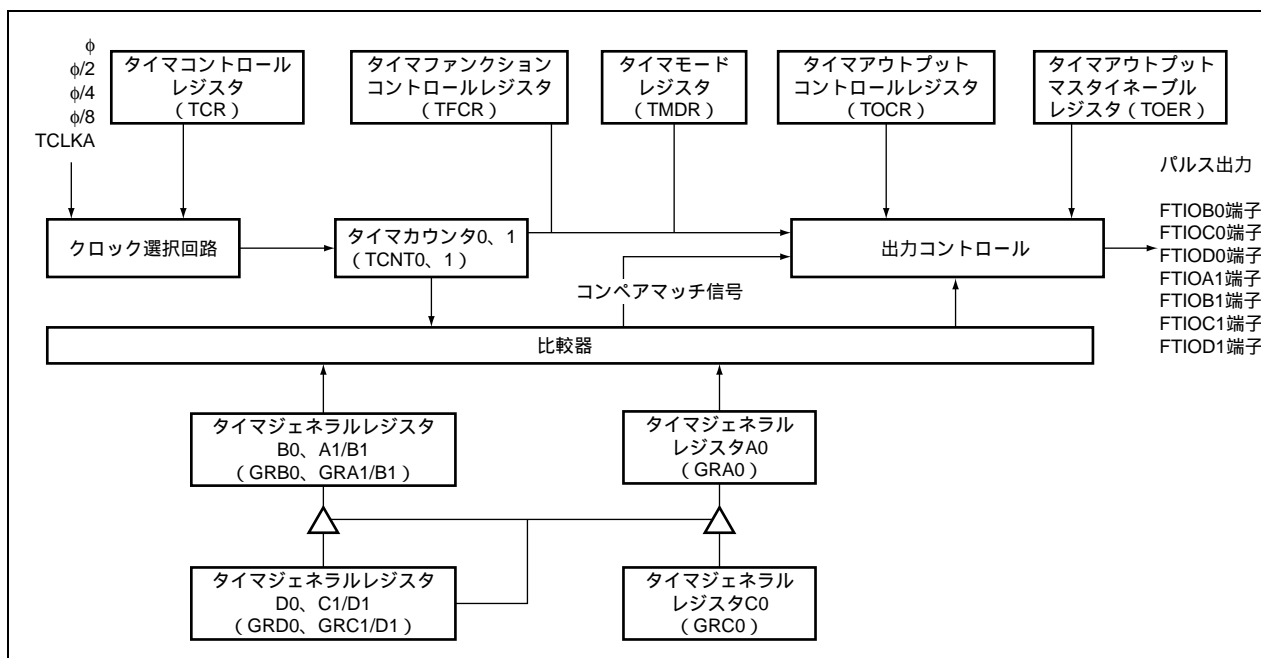


図 2.1 タイマ Z CH0、CH1 ブロック図

2. 表 2.1 に本タスクの機能割り付けを示します。表に示すように TimerZ の機能を割り付け、相補 PWM パルスを出力します。

表 2.1 機能割り付け

端子、レジスタ名	機能割り付け
FTIOC0	PWM に同期したトグル出力
FTIOB0	PWM 出力 1
FTIOD0	PWM 出力 1 とノンオーバーラップの関係にある逆相波形
FTIOA1	PWM 出力 2
FTIOB1	PWM 出力 3
FTIOC1	PWM 出力 2 とノンオーバーラップの関係にある逆相波形
FTIOD1	PWM 出力 3 とノンオーバーラップの関係にある逆相波形
TSTR	TCNT0、TCNT1 の動作／停止を選択
TOCR	コンペアマッチが最初に起こるまでの初期出力を設定
TOER	相補 PWM 出力端子の信号出力の許可／禁止
TMDR	GRC0,GRD0,GRC1,GRD1 をそれぞれ GRA0,GRB0,GRA1,GRB1 のバッファレジスタに設定する
TCR	CH0、CH1 のタイマカウンタのクリア要因と入力クロックを選択
TFCR	Timer Z を相補 PWM モードとして動作
GRA0	PWM 周期の 1/2 を設定
GRC0	GRA0 のバッファレジスタ
GRB0	出力パルスの変化点の設定（コンペアレジスタ）
GRA1	
GRB1	
GRC1	TGR4A のバッファレジスタ
GRD1	TGR4B のバッファレジスタ

3. 動作説明

1. 図 3.1 に動作原理を示します。H8/3687 のハードウェア処理およびソフトウェア処理により、相補 PWM 波形出力をします。

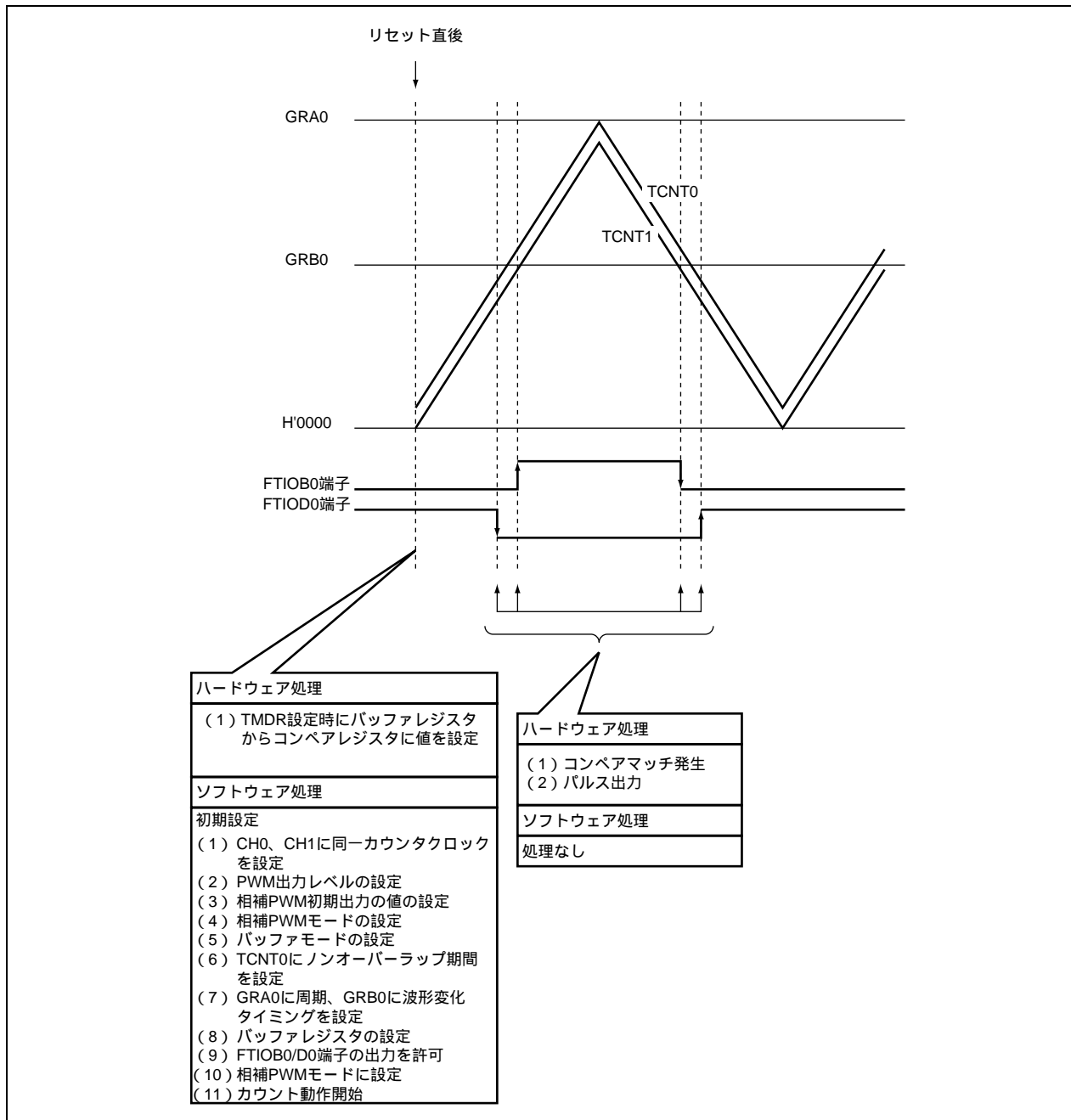


図 3.1 相補 PWM1 相分の波形出力の動作原理

2. 図 3.2 に動作原理を示します。H8/3687 のハードウェア処理およびソフトウェア処理により、相補 PWM 波形出力をします。ただし、転送モードは、谷でデータを変更するモードに選択してあります。

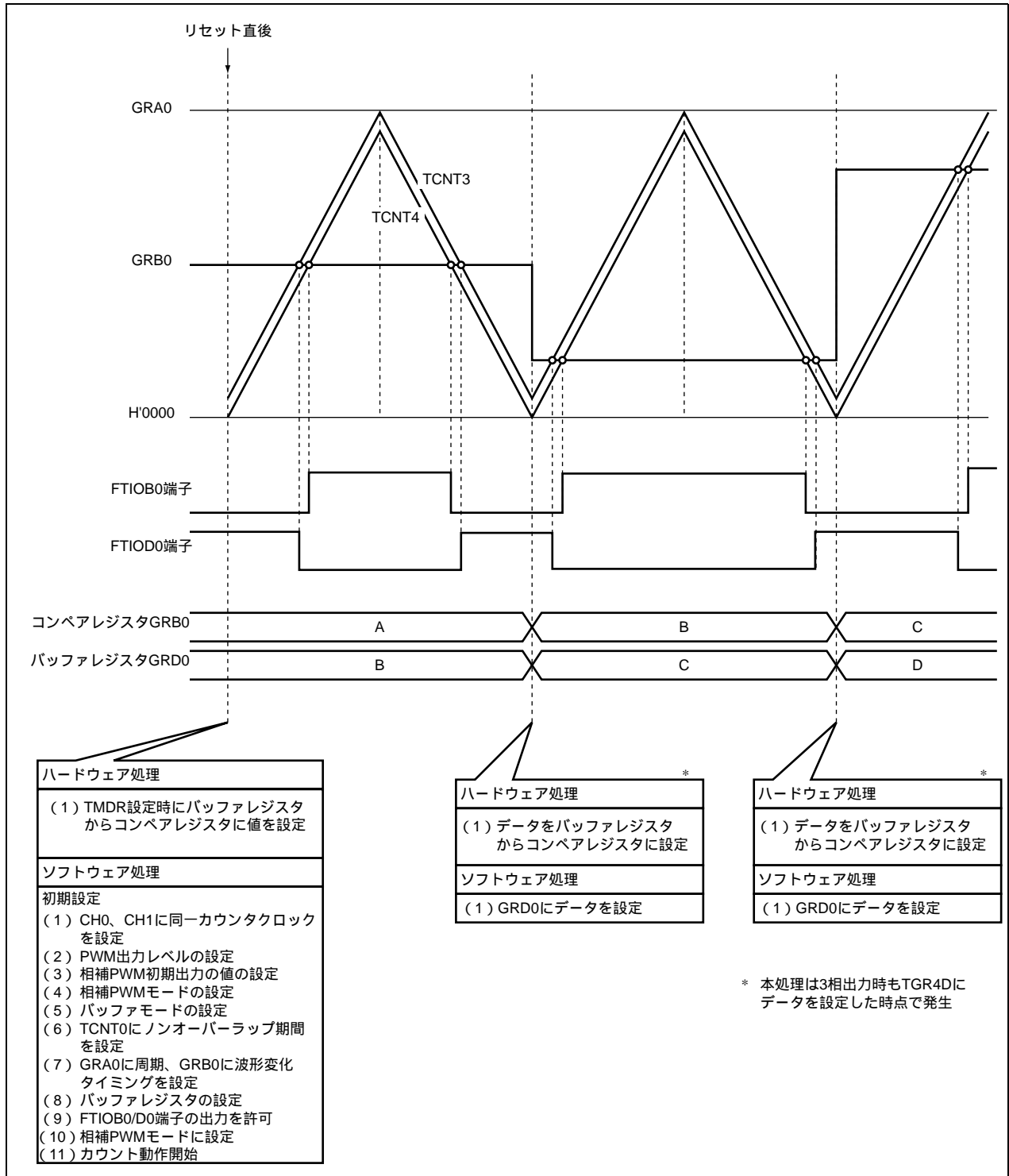


図 3.2 相補 PWM1 相波形出力の動作原理

3. 図 3.3 に動作原理を示します。H8/3687 のハードウェア処理およびソフトウェア処理により、PWM 周期に同期したトグル出力を行います。

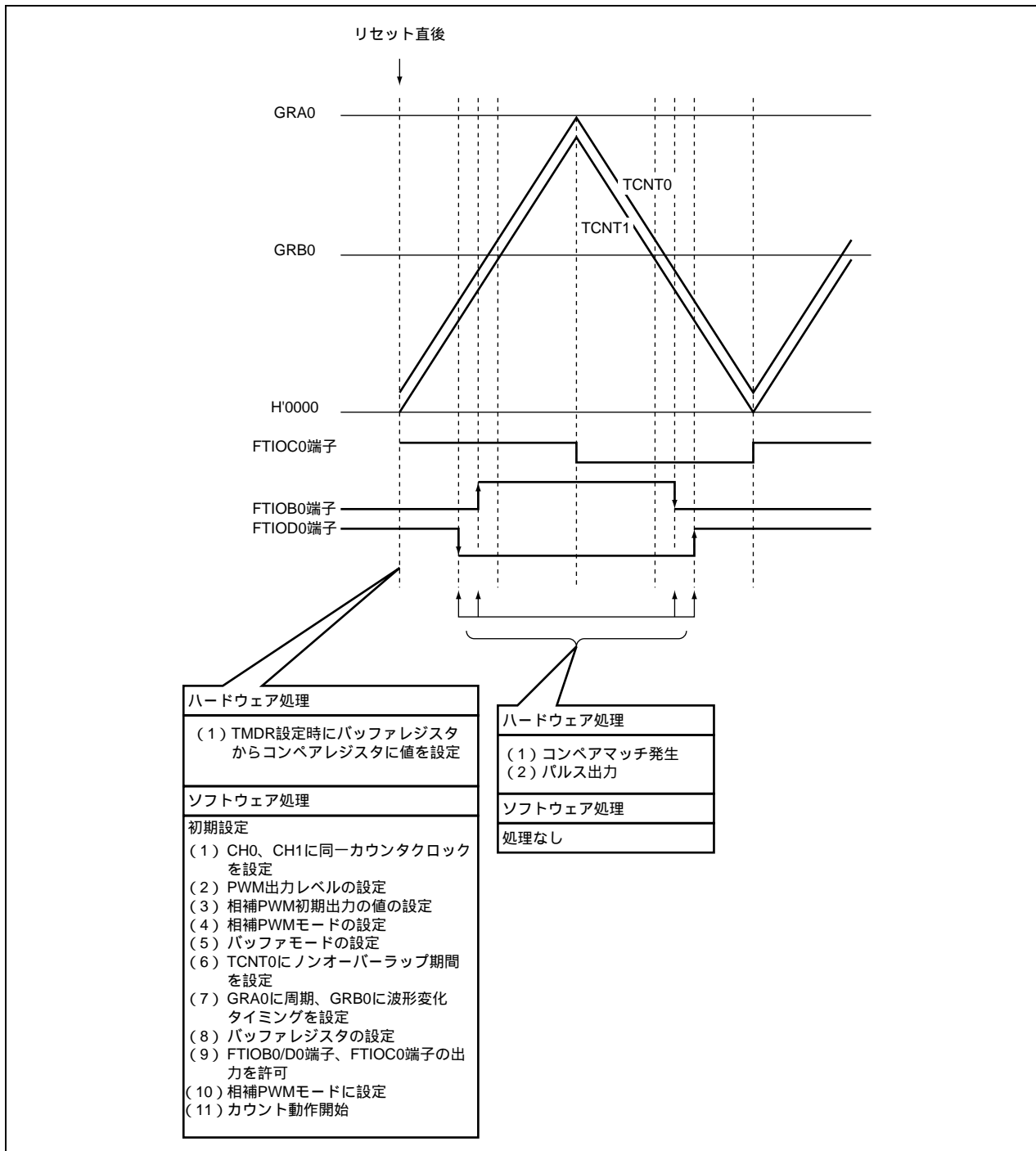


図 3.3 PWM 周期に同期したトグル波形出力の動作原理

4. 図 3.4 に動作原理を示します。図 3.4 に示すように H8/3687 のハードウェア処理およびソフトウェア処理により CH0、CH1 の各 PWM 出力端子 (FTIOB0/D0、FTIOA1/B1/C1/D1) から 6 相の PWM 出力をします。

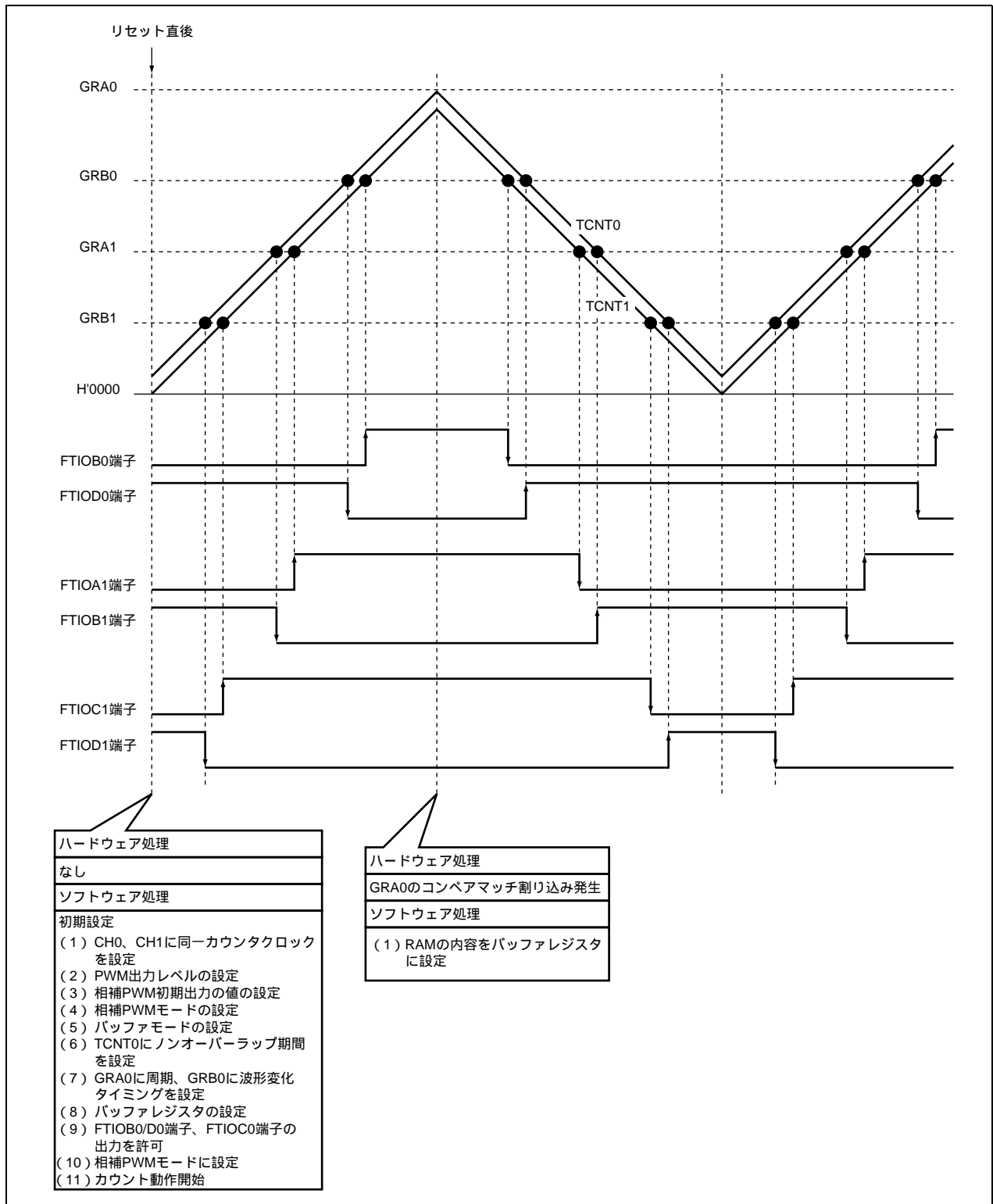


図 3.4 PWM 波形の動作原理

4. ソフトウェア説明

4.1 モジュール説明

モジュール名	ラベル名	機能割り付け
メインルーチン	main	相補 PWM 出力の設定
データ設定	setdata	バッファレジスタに波形変化タイミングを設定

4.2 引数の説明

ラベル名、レジスタ名	機能割り付け	データ長	使用モジュール名	入出力
pul_cyc	パルスの 1/2 周期を設定 パルスの周期は以下の式にて算出 パルス周期 (ns) = タイマ値 × φ周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力
duty_1	TIOC 端子から出力される波形変化タイミングを設定			
duty_2				
duty_3				
dead_time	ノンオーバーラップ期間を設定	メインルーチンデータ設定		

4.3 使用内部レジスタ説明

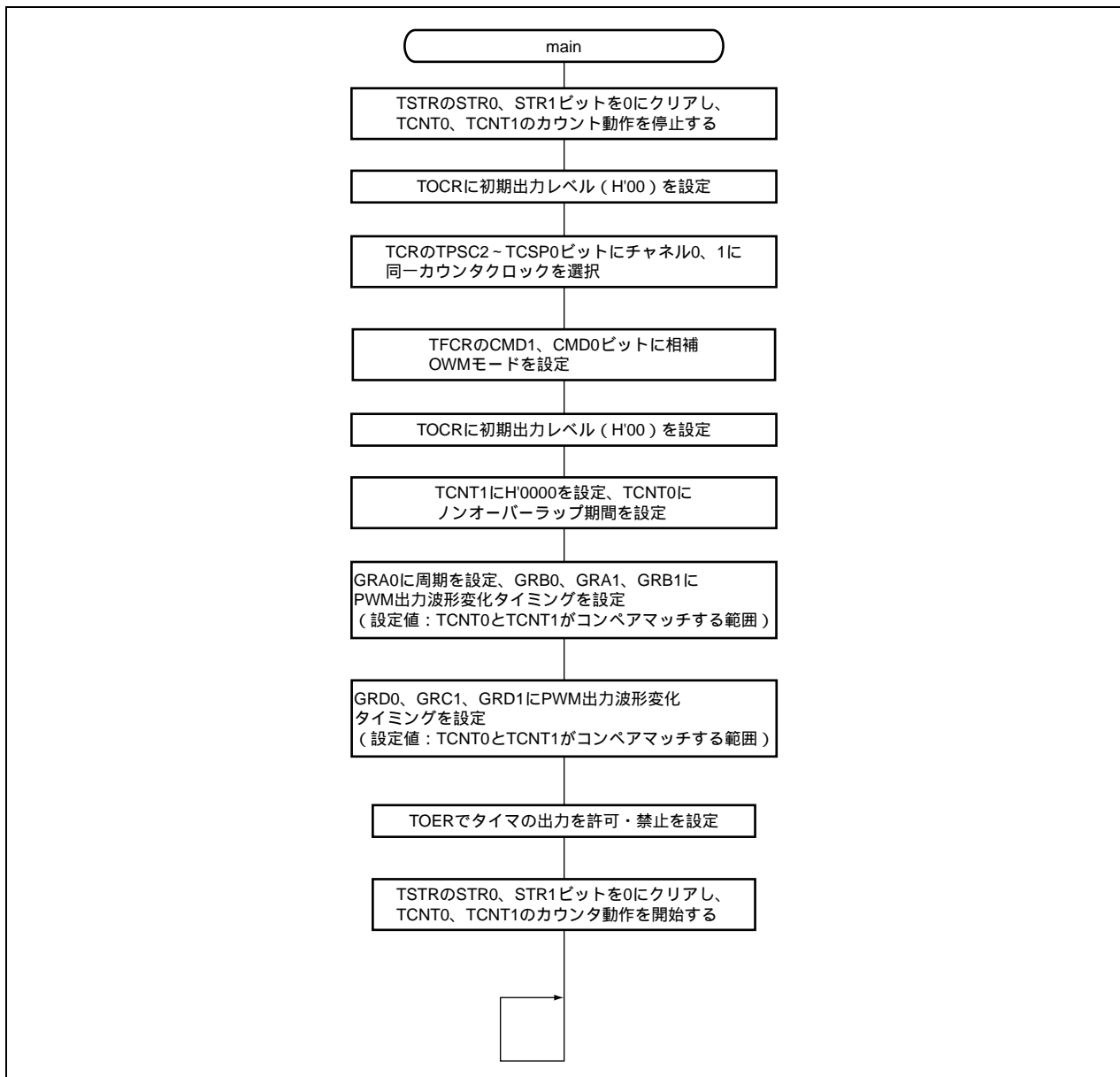
レジスタ名	機能割り付け	使用モジュール名	
TSTR	タイマカウントスタートを実行	メインルーチン	
TOCR	コンペアマッチ発生までの初期出力を設定		
TCR0	入力クロックを選択 (TCR1 と同じクロックを選択)		
TCR1	入力クロックを選択 (TCR0 と同じクロックを選択)		
TFCR	相補PWMモードに設定		
TMDR	GRC0、GRD0 を GRA0、GRB0 及び GRC1、GRD1 を GRA1、GRB1 のバッファレジスタに設定		
TCNT0	CH0 のタイマカウンタ		
TCNT1	CH1 のタイマカウンタ		
GRA0	キャリア周期の 1/2 の値を設定		メインルーチン
GRB0	FTIOB0、FTIOD0 から出力させるタイマカウンタ値を設定		データ設定
GRC0	GRA0 のバッファレジスタ	メインルーチンおよび 割り込みルーチンデータ設定	
GRD0	GRB0 のバッファレジスタ		
GRA1	FTIOA1、FTIOC1 から出力させるタイマカウンタ値を設定	メインルーチン	
GRB1	FTIOB1、FTIOD1 から出力させるタイマカウンタ値を設定	データ設定	
GRC1	GRA1 のバッファレジスタ	メインルーチンおよび 割り込みルーチンデータ設定	
GRD1	GRB1 のバッファレジスタ		
TOER	タイマ出力の許可を設定	メインルーチン	
TIER0	GRA0 の割り込みを許可		

4.4 使用 RAM 説明

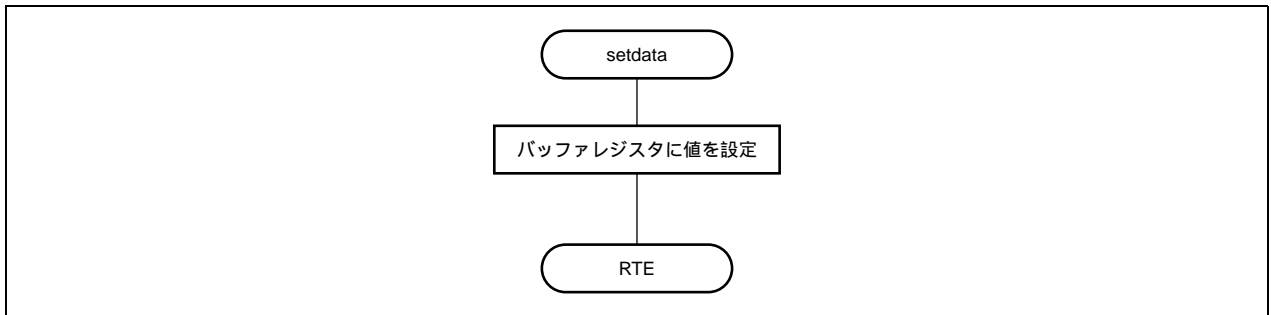
本アプリケーション例では引数以外の RAM は使用していません。

5. フローチャート

(1) メインルーチン



(2) 割り込みルーチン



6. プログラムリスト

```

/*****
/*
/* H8/300HN Series -H8/3687-
/* Application Note
/*
/* 'Complementary PWM Mode function'
/*
/* Function
/* :Timer Z Complementary PWM Mode
/*
/* External Clock : 16MHz
/* Internal Clock : 16MHz
/* Sub Clock : 32.768kHz
/*
*****/
#include <C:\ch38\include\machine.h>

/*****
/* Symbol Definition
*****/

struct BIT {
    unsigned char b7:1; /* bit7 */
    unsigned char b6:1; /* bit6 */
    unsigned char b5:1; /* bit5 */
    unsigned char b4:1; /* bit4 */
    unsigned char b3:1; /* bit3 */
    unsigned char b2:1; /* bit2 */
    unsigned char b1:1; /* bit1 */
    unsigned char b0:1; /* bit0 */
};

#define TCR0 *(volatile unsigned char *)0xF700 /* Timer Control Register 0 */
#define TIORA0 *(volatile unsigned char *)0xF701 /* Timer I/O Control Register A0 */
#define TIORC0 *(volatile unsigned char *)0xF702 /* Timer I/O Control Register C0 */
#define TSR0 *(volatile unsigned char *)0xF703 /* Timer Status Register 0 */
#define TSR0_BIT (*(struct BIT *)0xF703) /* Timer Status Register 0 */
#define IMIFA_0 TSR0_BIT.b0 /* Input Capture/Compare match Flag A */
#define TIER0 *(volatile unsigned char *)0xF704 /* Timer Interrupt Enable Register 0 */
#define POCR0 *(volatile unsigned char *)0xF705 /* Port Output Level Control Register */
#define TCNT0 *(volatile unsigned short *)0xF706 /* Timer Counter 0 */
#define GRA0 *(volatile unsigned short *)0xF708 /* General Register A0 */

```

```

#define GRB0      *(volatile unsigned short *)0xF70A    /* General Register B0 */
#define GRC0      *(volatile unsigned short *)0xF70C    /* General Register C0 */
#define GRD0      *(volatile unsigned short *)0xF70E    /* General Register D0 */

#define TCR1      *(volatile unsigned char *)0xF710     /* Timer Control Register 1 */
#define TIORAL    *(volatile unsigned char *)0xF711     /* Timer I/O Control Register A1 */
#define TIORC1    *(volatile unsigned char *)0xF712     /* Timer I/O Control Register C1 */
#define TSR1      *(volatile unsigned char *)0xF713     /* Timer Status Register 1 */
#define TIER1     *(volatile unsigned char *)0xF714     /* Timer Interrupt Enable Register 0 */
#define POCR1     *(volatile unsigned char *)0xF715     /* Port Output Level Control Register */
#define TCNT1     *(volatile unsigned short *)0xF716    /* Timer Counter 1 */
#define GRA1      *(volatile unsigned short *)0xF718    /* General Register A1 */
#define GRB1      *(volatile unsigned short *)0xF71A    /* General Register B1 */
#define GRC1      *(volatile unsigned short *)0xF71C    /* General Register C1 */
#define GRD1      *(volatile unsigned short *)0xF71E    /* General Register D1 */

#define TSTR      *(volatile unsigned char *)0xF720     /* Timer Start Register */
#define TMDR      *(volatile unsigned char *)0xF721     /* Timer Mode Register */
#define TPMR      *(volatile unsigned char *)0xF722     /* Timer PWM Mode Register */
#define TFCR      *(volatile unsigned char *)0xF723     /* Timer Function Control Register */
#define TOER      *(volatile unsigned char *)0xF724     /* Timer Output Master Enable Register */
#define TOCR      *(volatile unsigned char *)0xF725     /* Timer Output Master Enable Register */

#pragma interrupt (setdata)

/*****
/* 関数定義
*****/

extern void INIT ( void ); /* SP Set */
void main ( void );
void setdata ( void );

/*****
/* RAM define
*****/

unsigned char duty_1; /* Duty Setting Buffer 1 */
unsigned char duty_2; /* Duty Setting Buffer 2 */
unsigned char duty_3; /* Duty Setting Buffer 3 */
unsigned short pul_cyc; /* PWM cycle Data Set */
unsigned short dead_time; /* Dead Time Data Set

extern void _INITSCT();

```

```

/*****
/*   Vector Address
/*****
#pragma section V1 /* VECTOR SECTOIN SET
void (*const VEC_TBL1[])(void) = { /* 0x00 - 0x0f
    INIT /* 00 Reset
};

#pragma section V2 /* VECTOR SECTOIN SET
void (*const VEC_TBL2[])(void) =
    setdata /* 34 Timer z(CH0) Interrupt
};

#pragma section /* P
/*****
/*   Main Program
/*****
void main ( void )
{
    _INITSCT();

    set_imask_ccr(1); /* Interrupt Disable
    TSTR = 0xFC; /* Timer Stop
    TOCR = 0x00; /* PWM initial output "0"
    TCR0 = 0x00; /* No Clear Mode
    TCR1 = 0x00; /* No Clear Mode
    TFCR = 0x0A; /* Complementary PWM Mode
    TMDR = 0xEE; /* GRB0,GRA1,GRB1 Buffer Mode

    dead_time = 0x0000;

    TCNT0 = dead_time; /* No Dead Time
    TCNT1 = 0x0000;
    pul_cyc = 0x190; /* Carieer Cycle 50us
    duty_1 = 0x78; /* Duty 30%
    duty_2 = 0xC8; /* Duty 50%
    duty_3 = 0x140; /* Duty 80%
    GRA0 = pul_cyc; /* Carieer Cycle 50us
    GRB0 = duty_1; /* Duty 30%
    GRA1 = duty_2; /* Duty 50%
    GRB1 = duty_3; /* Duty 80%

    GRD0 = duty_1; /* Duty 30%
    GRC1 = duty_2; /* Duty 50%

```

```

GRD1 = duty_3;                /* Duty 80%                */
                                */

TOER = 0x00;                  /* FTIOB0,FTIO00,FTIOA1,FTIOB1,FTIOC1 */
                                */
                                /* FTIOD1 Output Enable        */
                                */

TIER0 = 0x01;                 /* Interrupt Enable Disable    */
                                */

TSTR = 0xFF;                  /* TCNT0,TCNT1 Start          */
                                */

set_imask_ccr(0);             /* Interrupt Enable            */
                                */

while(1){
    ;
}

/*****
/*   Timerz Interrupt
*****/

void setdata ( void )
{
    TSR = (TSR & 0xF7);        /* Clear IMIFA_0 to 0        */
                                */

    GRD0 = duty_1;
    GRC1 = duty_2;
    GRD1 = duty_3;
}

```

INIT.SRC (プログラムリスト)

```

        .EXPORT      _INIT
        .IMPORT      _main
;

        .SECTION    P, CODE

_INIT:
        MOV.W        #H'FF80,R7
        LDC.B        #B'10000000,CCR
        JMP          @_main
;

        .END

```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'0036
P	H'0100
B	H'F780