

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。

H8/300H Tiny シリーズ

タイマ Z を使用したリセット同期 PWM 制御例

要旨

タイマ Z のリセット同期 PWM モードを使用し、PWM 波形（正相と逆相）を出力します。

動作確認デバイス

H8/36049CPU

目次

1. 仕様	2
2. 使用機能説明	2
3. 動作説明	6
4. ソフトウェア説明	7
5. フローチャート	11
6. プログラムリスト	16

1. 仕様

1. タイマ Z のリセット同期 PWM モードを使用し、PWM 波形（正相と逆相）を出力します。
2. FTIOB0～FTIOD0 および FTIOA1～FTIOD1 の端子は自動的に PWM 出力端子となり、TCNT_0 はアップカウンタとして機能します。
3. GRB_0 の値を GRA_0 の割り込みごとに変更することによって、FTIOB0 および FTIOD0 のデューティを変更します。
4. バッファ動作は未使用とします。
5. TCNT_0 と GRC_0 のコンペアマッチ時にトグル出力を行ないます。
6. TCNT_0 は GRA_0 コンペアマッチでクリアされます。
7. TCNT_1 は TCNT_0 と独立してスタートし、TCNT_1 と GRC_1 のコンペアマッチ時にトグル出力を行ないます。
8. TCNT_1 はフリーランニングカウンタとして使用します。

2. 使用機能説明

1. 本タスク例では、タイマ Z のリセット同期 PWM モードを使用し、PWM 出力端子よりデューティパルスを出力します。タイマ Z のブロック図を図 1 に示します。以下にタイマ Z のブロック図について説明します。
 - システムクロック (ϕ)
20 MHz のクロックで、CPU および周辺機能を動作させるための基準クロックです。
 - タイマスタートレジスタ (TSTR)
TCNT_0、TCNT_1 の動作 / 停止を選択します。本タスク例では、TCNT_0、TCNT_1 とともにカウント動作に設定しています。
 - タイマモードレジスタ (TMDR)
TCNT_0、TCNT_1 のタイマ同期 / 独立を選択します。本タスク例では、TCNT_0 と TCNT_1 は、独立動作に設定しています。それから、バッファ動作は行わず、通常動作に設定しています。
 - タイマファンクションコントロールレジスタ (TFCR)
各動作モードの設定や出力レベルを選択します。本タスク例では、チャンネル 0、チャンネル 1 をリセット同期 PWM モード動作に設定し、初期出力はローレベル、アクティブレベルはハイレベルに設定しています。
 - タイマアウトプットマスタインエーブルレジスタ (TOER)
チャンネル 0、チャンネル 1 の出力許可 / 禁止を選択します。本タスク例では、すべての出力を許可にしています。
 - タイマアウトプットコントロールレジスタ (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します。本タスク例では、すべての初期出力を 0 に設定しています。
 - タイマコントロールレジスタ_0 (TCR_0)
TCNT_0 の入力クロック、クリア方法を選択します。本タスク例では、 ϕ の立ち上がりエッジでカウント、GRA_0 のコンペアマッチで TCNT_0 をクリアする設定にしています。
 - タイマコントロールレジスタ_1 (TCR_1)
TCNT_1 の入力クロック、クリア方法を選択します。本タスク例では、 ϕ の立ち上がりエッジでカウント、TCNT_1 をクリア禁止とする設定にしています。
 - タイマカウンタ_0 (TCNT_0)
16 ビットのリード / ライト可能なアップカウンタです。入力する内部クロック / 外部クロックによりカウントアップされます。本タスク例では、 ϕ の立ち上がりエッジでカウントします。
 - タイマカウンタ_1 (TCNT_1)
16 ビットのリード / ライト可能なアップカウンタです。入力する内部クロック / 外部クロックによりカウントアップされます。本タスク例では、 ϕ の立ち上がりエッジでカウントします。

- ジェネラルレジスタ (GRA_0, GRA_1, GRB_0, GRB_1, GRC_0, GRC_1)
16 ビットのリード/ライト可能なレジスタです。ジェネラルレジスタの内容は TCNT_0 と常に比較されています。
- インพุットキャプチャ/アウトプットコンペア C0 端子 (FTIOC0)
PWM 周期に同期したトグル出力。
- インพุットキャプチャ/アウトプットコンペア B0 端子 (FTIOB0)
PWM 出力 1。
- インพุットキャプチャ/アウトプットコンペア D0 端子 (FTIOD0)
PWM 出力 1 (PWM 出力 1 の逆相波形)。
- インพุットキャプチャ/アウトプットコンペア A1 端子 (FTIOA1)
PWM 出力 2。
- インพุットキャプチャ/アウトプットコンペア C1 端子 (FTIOC1)
PWM 出力 2 (PWM 出力 2 の逆相波形)。
- インพุットキャプチャ/アウトプットコンペア B1 端子 (FTIOB1)
PWM 出力 3。
- インพุットキャプチャ/アウトプットコンペア D1 端子 (FTIOD1)
PWM 出力 3 (PWM 出力 3 の逆相波形)。
- チャンネル 0 割り込み (ITMZ0)
本タスク例では、TCNT_0 と GRC_0 のコンペアマッチによる割り込みを使用しています。
- チャンネル 1 割り込み (ITMZ1)
本タスク例では、TCNT_1 と GRC_1 のコンペアマッチによる割り込みを使用しています。
- I/O ポートの P30, P31 端子 (P30, P31)
P30 に GRC_0 からの割り込みによるトグル出力を、P31 に GRC_1 からの割り込みによるトグル出力を行ないます。

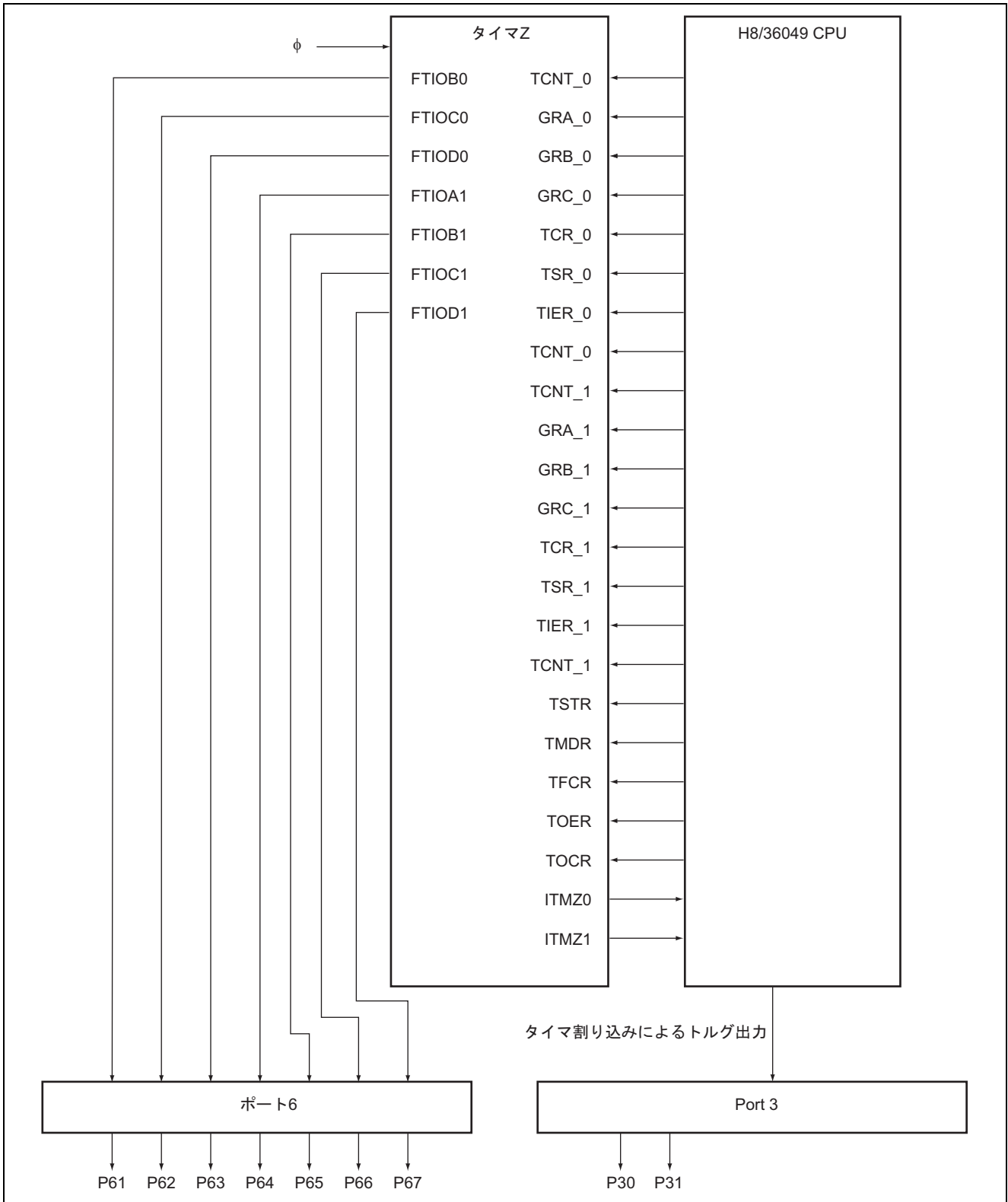


図 1 タイマ Z を使用したリセット同期 PWM 出力のブロック図

2. 本タスク例の機能割り付けを表 1 に示します。表 1 に示すように機能を割り付け、リセット同期 PWM 出力を行いません。

表 1 機能割り付け

機能	割り付け
TSTR	TCNT_0, TCNT_1 の動作 / 停止を選択
TMDR	TCNT_0 と TCNT_1 は、独立動作に設定
TFCR	チャンネル 0, チャンネル 1 をリセット同期 PWM モード動作に設定
TOER	すべての出力を許可
TOCR	すべての初期出力を 0 に設定
TCR_0	TCNT_0 の入力クロック, クリア方法を選択
TCR_1	TCNT_1 の入力クロック, クリア方法を選択
TCNT_0	16 ビットのリード/ライト可能なアップカウンタ
GRA_0	TCNT_0 と常に比較
GRA_1	TCNT_0 と常に比較
GRB_0	TCNT_0 と常に比較
GRB_1	TCNT_0 と常に比較
GRC_0	TCNT_0 と常に比較
GRC_1	TCNT_0 と常に比較
FTIOC0 端子	PWM 周期に同期したトグル出力
FTIOB0 端子	PWM 出力 1
FTIOD0 端子	PWM 出力 1 (PWM 出力 1 の逆相波形)
FTIOA1 端子	PWM 出力 2
FTIOC1 端子	PWM 出力 2 (PWM 出力 2 の逆相波形)
FTIOB1 端子	PWM 出力 3
FTIOD1 端子	PWM 出力 3 (PWM 出力 3 の逆相波形)
ITMZ0	GRC_0 によるチャンネル 0 割り込み
ITMZ1	GRC_1 によるチャンネル 1 割り込み
ポート 3	割り込みによるトグル出力を行いません。

3. 動作説明

図 2 に動作説明を示します。図 2 に示すようなハードウェア処理、およびソフトウェア処理によりリセット同期 PWM 出力を行ないます。

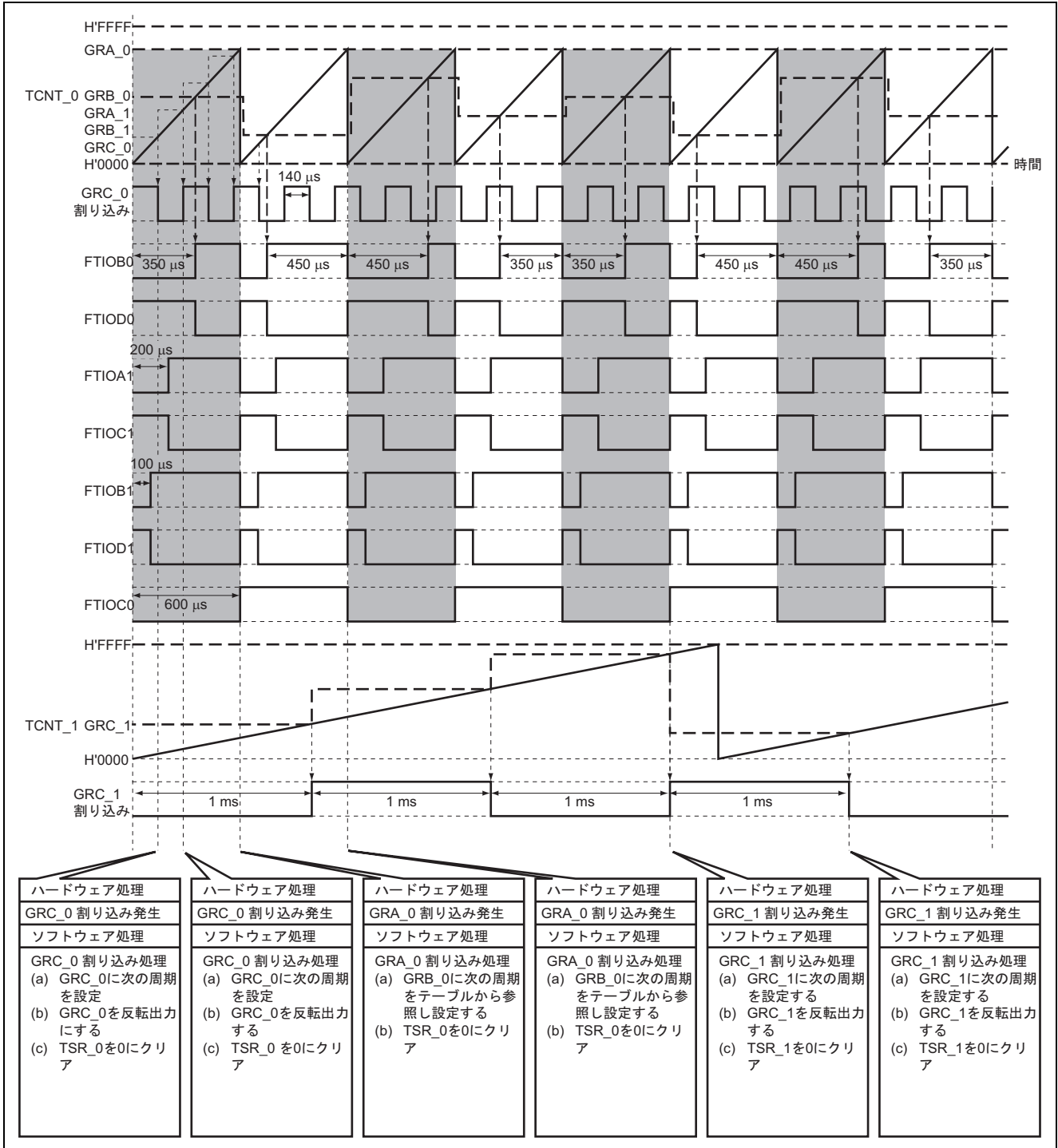


図 2 タイマ Z を使用したリセット同期 PWM 出力の動作説明

4. ソフトウェア説明

4.1 モジュール説明

表 2 に本タスク例におけるモジュール説明を示します。

表 2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	タイマ Z コンペアマッチ機能の設定, カウンタ開始, コンペアマッチ出力端子の設定を行なう。
タイマ Z0 割り込み処理ルーチン	tmrz0	GRC_0 による割り込み時 : 次の割り込みポイントを設定する。トグル出力を行なう。 GRA_0 による割り込み時 : 次の割り込みポイントを設定する。
タイマ Z1 割り込み処理ルーチン	tmrz1	次の割り込みポイントを設定する。トグル出力を行なう。

4.2 引数説明

本タスク例では, 引数を使用しておりません。

4.3 使用内部レジスタ説明

本タスク例の使用内部レジスタを以下に示します。

- TSTR タイマスタートレジスタ アドレス: 0xFFFF720

ビット	ビット名	設定値	機能
1	STR1	0/1	チャンネル 1 カウンタスタート 0: TCNT_1 はカウント動作停止 1: TCNT_1 はカウント動作
0	STR0	0/1	チャンネル 0 カウンタスタート 0: TCNT_0 はカウント動作停止 1: TCNT_0 はカウント動作

- TMDR タイマモードレジスタ アドレス: 0xFFFF721

ビット	ビット名	設定値	機能
7	BFD1	0	バッファ動作 D1 0: GRD_1 は通常動作
6	BFC1	0	バッファ動作 C1 0: GRC_1 は通常動作
5	BFD0	0	バッファ動作 D0 0: GRD_0 は通常動作
4	BFC0	0	バッファ動作 C0 0: GRC_0 は通常動作
0	SYNC	0	タイマ同期 0: TCNT_1, TCNT_0 は独立動作

● TFCR タイマファンクションコントロールレジスタ アドレス：0xFF723

ビット	ビット名	設定値	機能
3	OLS1	1	出力レベルセレクト 1 1：初期出力はローレベル，アクティブレベルはハイレベル
2	OLS0	1	出力レベルセレクト 0 1：初期出力はローレベル，アクティブレベルはハイレベル
1 0	CMD1 CMD0	CMD1 = 0 CMD0 = 1	コンビネーションモード 1~0 CMD1 = 0, CMD0 = 1：チャンネル 0, 1 は，リセット同期 PWM モード

● TOER タイマアウトプットマスタイネーブルレジスタ アドレス：0xFF724

ビット	ビット名	設定値	機能
7	ED1	0	マスタイネーブル D1 0：FTIOD1 端子の出力を許可
6	EC1	0	マスタイネーブル C1 0：FTIOC1 端子の出力を許可
5	EB1	0	マスタイネーブル B1 0：FTIOB1 端子の出力を許可
4	EA1	0	マスタイネーブル A1 0：FTIOA1 端子の出力を許可
3	ED0	0	マスタイネーブル D0 0：FTIOD0 端子の出力を許可
2	EC0	0	マスタイネーブル C0 0：FTIOC0 端子の出力を許可
1	EB0	0	マスタイネーブル B0 0：FTIOB0 端子の出力を許可
0	EA0	0	マスタイネーブル A0 0：FTIOA0 端子の出力を許可

● TOCR タイマアウトプットコントロールレジスタ アドレス：0xFF725

ビット	ビット名	設定値	機能
7	TOD1	0	出力レベルセレクト D1 TOD1 = 0：FTIOD1 端子の初期出力を 0 に設定
6	TOC1	0	出力レベルセレクト C1 TOC1 = 0：FTIOC1 端子の初期出力を 0 に設定
5	TOB1	0	出力レベルセレクト B1 TOB1 = 0：FTIOB1 端子の初期出力を 0 に設定
4	TOA1	0	出力レベルセレクト A1 TOA1 = 0：FTIOA1 端子の初期出力を 0 に設定
3	TOD0	0	出力レベルセレクト D0 TOD0 = 0：FTIOD0 端子の初期出力を 0 に設定
2	TOC0	0	出力レベルセレクト C0 TOC0 = 0：FTIOC0 端子の初期出力を 0 に設定
1	TOB0	0	出力レベルセレクト B0 TOB0 = 0：FTIOB0 端子の初期出力を 0 に設定
0	TOA0	0	出力レベルセレクト A0 TOA0 = 0：FTIOA0 端子の初期出力を 0 に設定

• TCR_0 タイマコントロールレジスタ_0 アドレス：0xFF700

ビット	ビット名	設定値	機能
7	CCLR2	CCLR2 = 0	カウンタクリア 2~0 CCLR2 = 0, CCLR1 = 0, CCLR0 = 1 : GRA0 のコンペアマッチで TCNT0 をクリア
6	CCLR1	CCLR1 = 0	
5	CCLR0	CCLR0 = 1	
4	CKEG1	CKEG1 = 0	クロックエッジ 1~0 CKEG1 = 0, CKEG0 = 0 : 立ち上がりエッジでカウント
3	CKEG0	CKEG0 = 0	
2	TPSC2	TPSC2 = 0	タイマプリスケラ 2~0 TPSC2 = 0, TPSC1 = 0, TPSC0 = 0 : ϕ でカウント
1	TPSC1	TPSC1 = 0	
0	TPSC0	TPSC0 = 0	

• TCR_1 タイマコントロールレジスタ_1 アドレス：0xFF710

ビット	ビット名	設定値	機能
7	CCLR2	CCLR2 = 0	カウンタクリア 2~0 CCLR2 = 0, CCLR1 = 0, CCLR0 = 0 : TCNT1 のクリア禁止
6	CCLR1	CCLR1 = 0	
5	CCLR0	CCLR0 = 0	
4	CKEG1	CKEG1 = 0	クロックエッジ 1~0 CKEG1 = 0, CKEG0 = 0 : 立ち上がりエッジでカウント
3	CKEG0	CKEG0 = 0	
2	TPSC2	TPSC2 = 0	タイマプリスケラ 2~0 TPSC2 = 0, TPSC1 = 0, TPSC0 = 0 : ϕ でカウント
1	TPSC1	TPSC1 = 0	
0	TPSC0	TPSC0 = 0	

• TCNT_0 タイマカウンタ_0 アドレス：0xFF706

機能： ϕ の立ち上がりエッジでカウントする 16 ビットのアップカウンタ
設定値：0

• TCNT_1 タイマカウンタ_1 アドレス：0xFF716

機能： ϕ の立ち上がりエッジでカウントする 16 ビットのアップカウンタ
設定値：0

• GRA_0 ジェネラルレジスタ A_0 アドレス：0xFF708

機能：GRA_0 の設定値と TCNT_0 のカウンタ値が一致すると、コンペアマッチが発生
設定値：12000

• GRA_1 ジェネラルレジスタ A_1 アドレス：0xFF718

機能：GRA_1 の設定値と TCNT_0 のカウンタ値が一致すると、コンペアマッチが発生
設定値：4000

• GRB_0 ジェネラルレジスタ B_0 アドレス：0xFF70A

機能：GRB_0 の設定値と TCNT_0 のカウンタ値が一致すると、コンペアマッチが発生
設定値：GRA_0 の割り込みごとに GRB_DATA 配列内の 1 データを設定
(本タスクでは 4 つのデータを繰り返す)

• GRB_1 ジェネラルレジスタ B_1 アドレス：0xFF71A

機能：GRB_1 の設定値と TCNT_0 のカウンタ値が一致すると、コンペアマッチが発生
設定値：2000

• GRC_0 ジェネラルレジスタ C_0 アドレス：0xFF70C

機能：GRC_0 の設定値と TCNT_0 のカウンタ値が一致すると、コンペアマッチが発生
設定値：2800

• GRC_1 ジェネラルレジスタ C_1 アドレス：0xFF71C

機能：GRC_1 の設定値と TCNT_1 のカウンタ値が一致すると、コンペアマッチが発生
設定値：20000

4.4 使用 RAM 説明

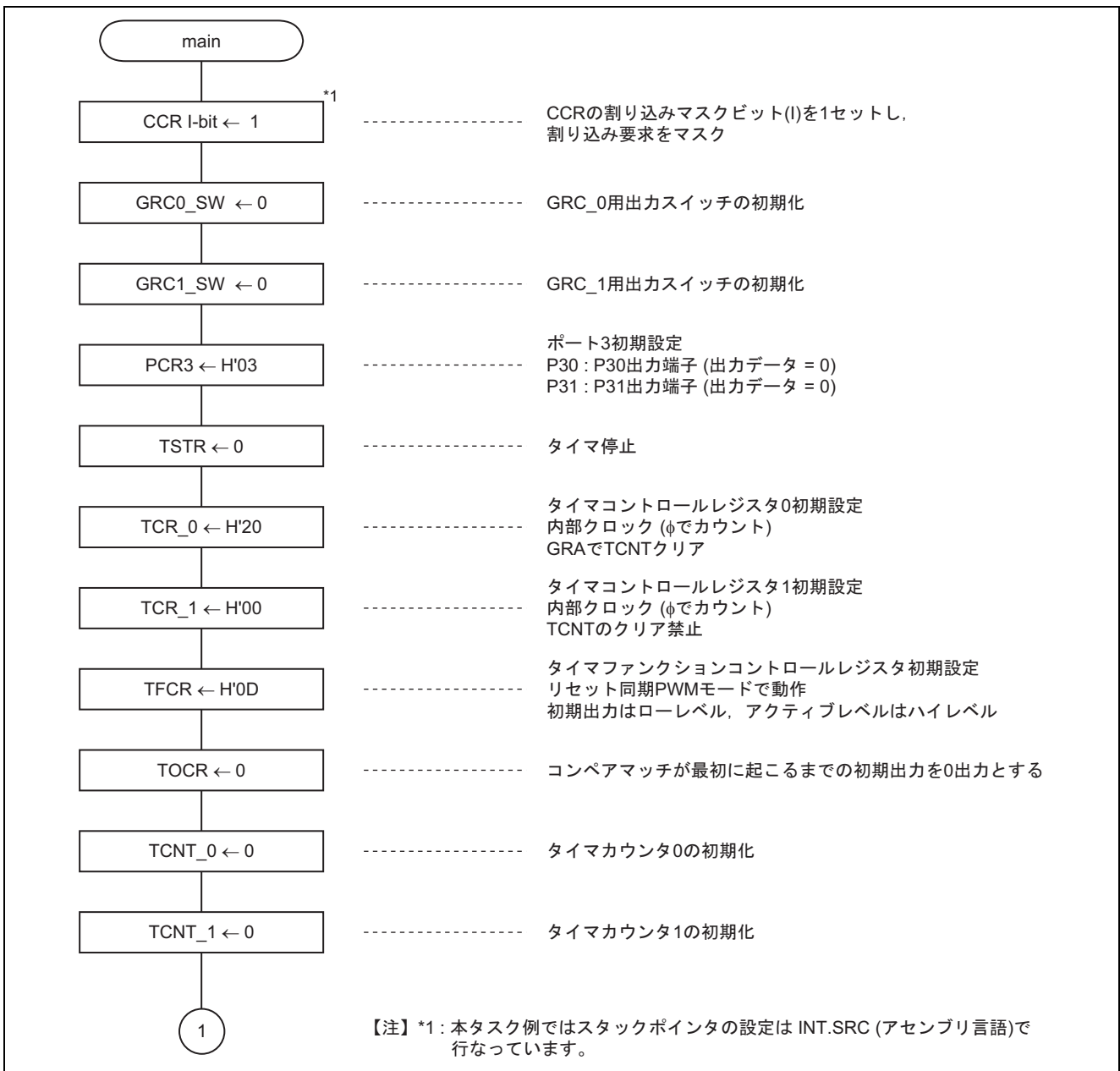
表 3 に本タスク例における使用 RAM 説明を示します。

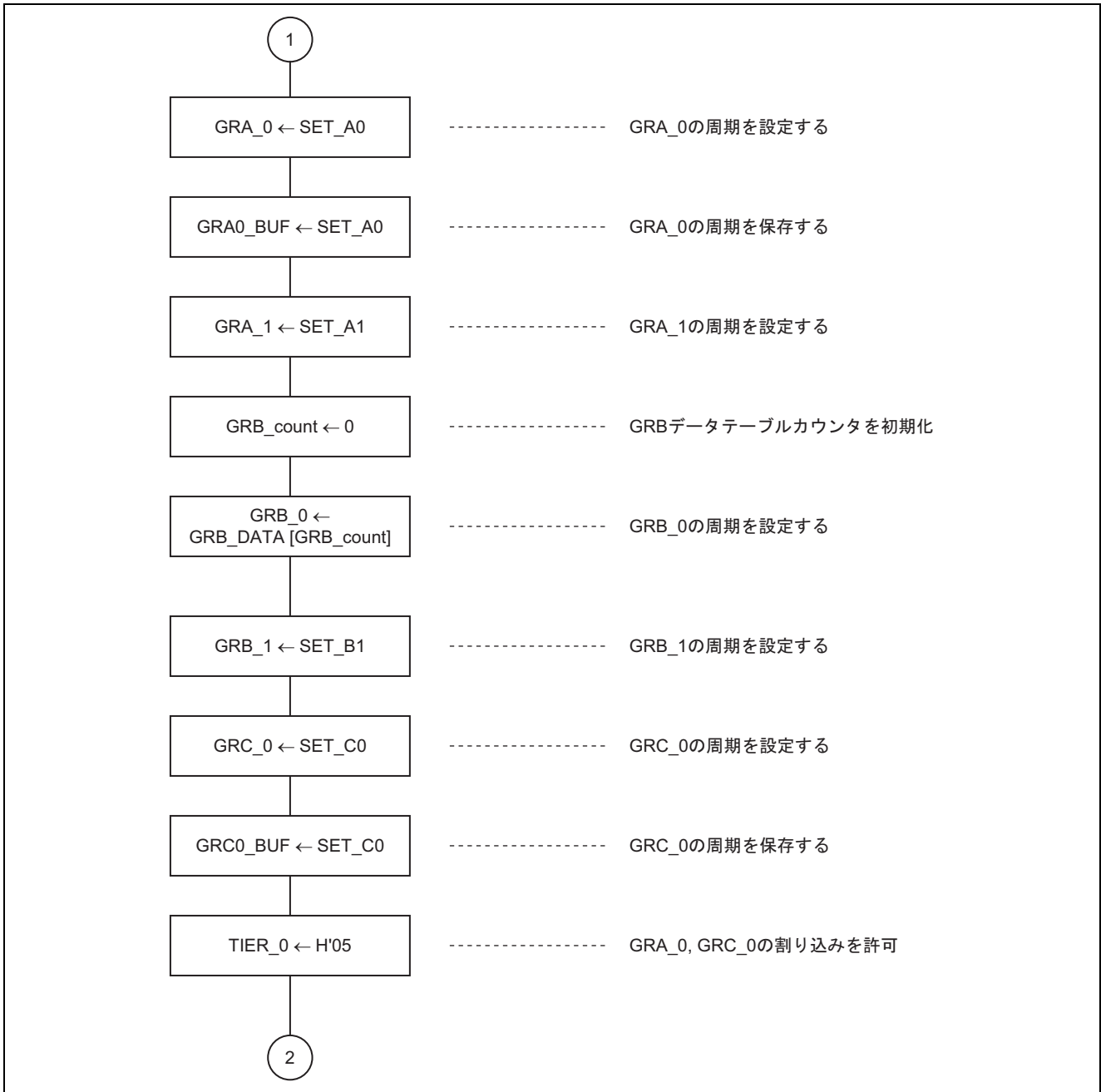
表 3 使用 RAM 説明

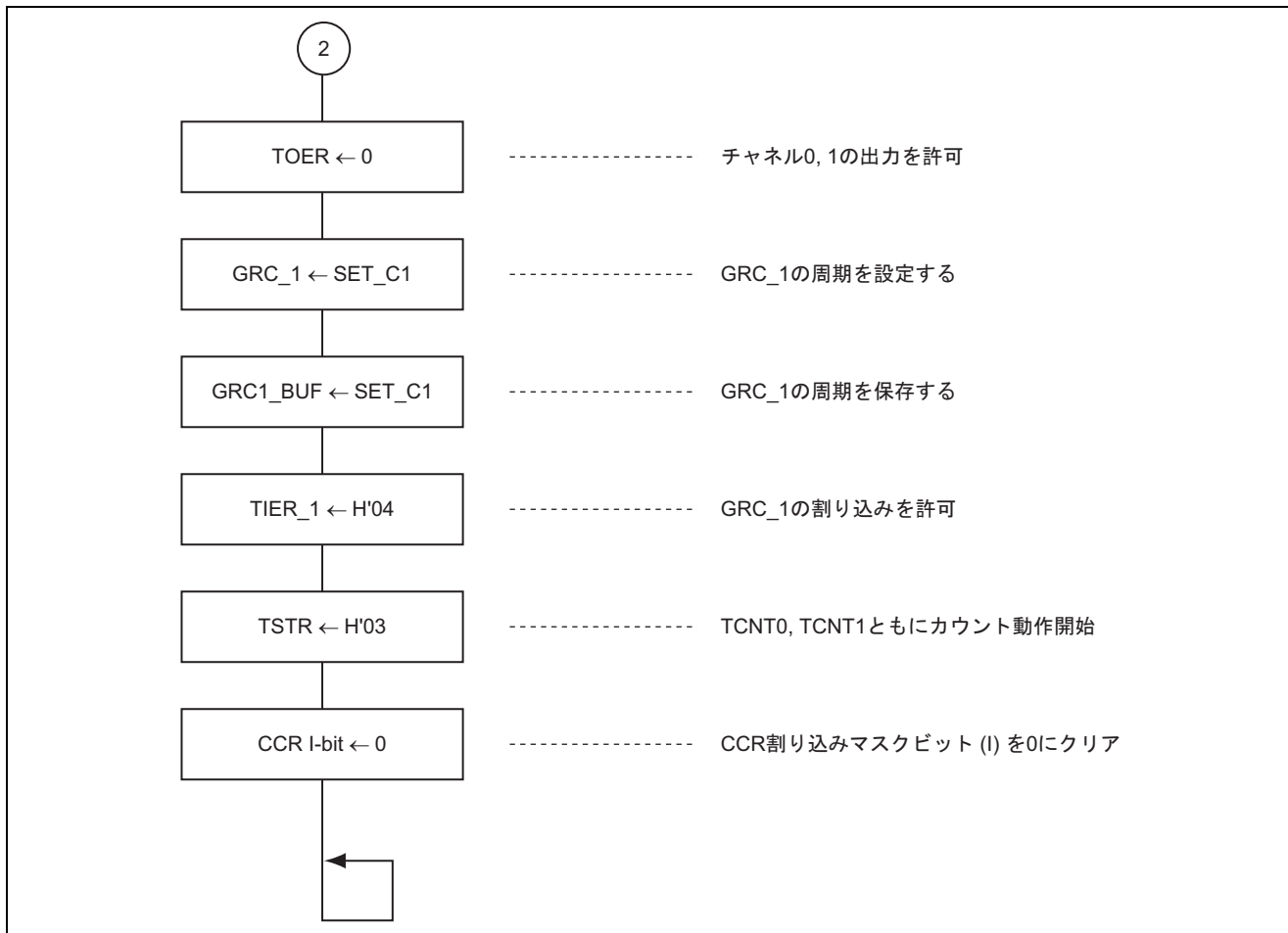
ラベル名	機能	アドレス	使用モジュールラベル名
TSR_SV	TSR_0 の値を保存	H'FFE812	tmrz0
GRB_count	GRB_0 に出力するデータへのカウンタ	H'FFE800	main, tmrz0
GRA0_BUF	GRA_0 の初期設定データを保存	H'FFE802	main, tmrz0
GRC0_BUF	GRC_0 の出力データを保存	H'FFE806	main, tmrz0
GRC1_BUF	GRC_1 の出力データを保存	H'FFE80A	main, tmrz1
GRC0_SW	GRC_0 のトグル出力用スイッチ	H'FFE80E	main, tmrz0
GRC1_SW	GRC_1 のトグル出力用スイッチ	H'FFE810	main, tmrz1

5. フローチャート

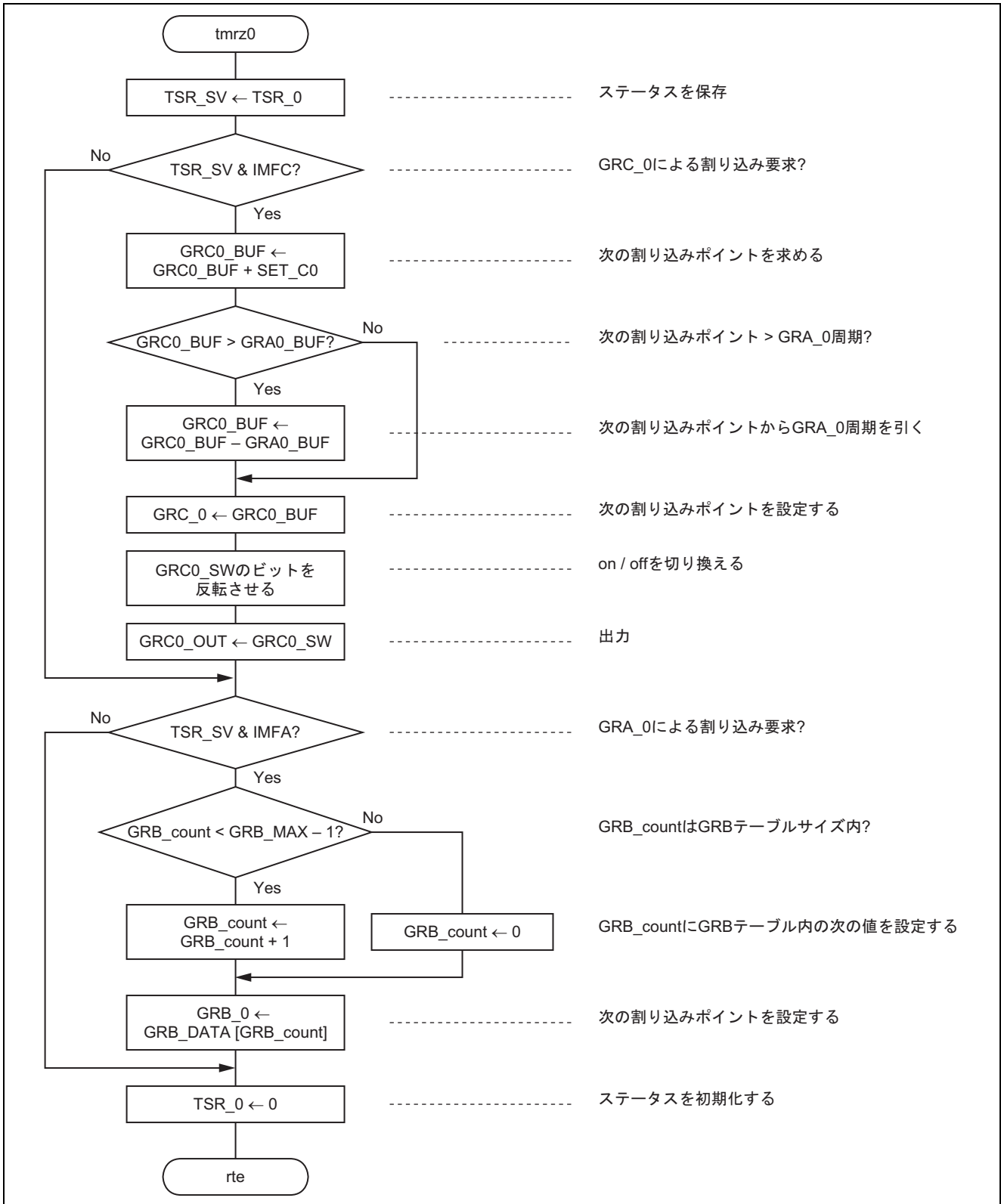
(1) メインルーチン (main)



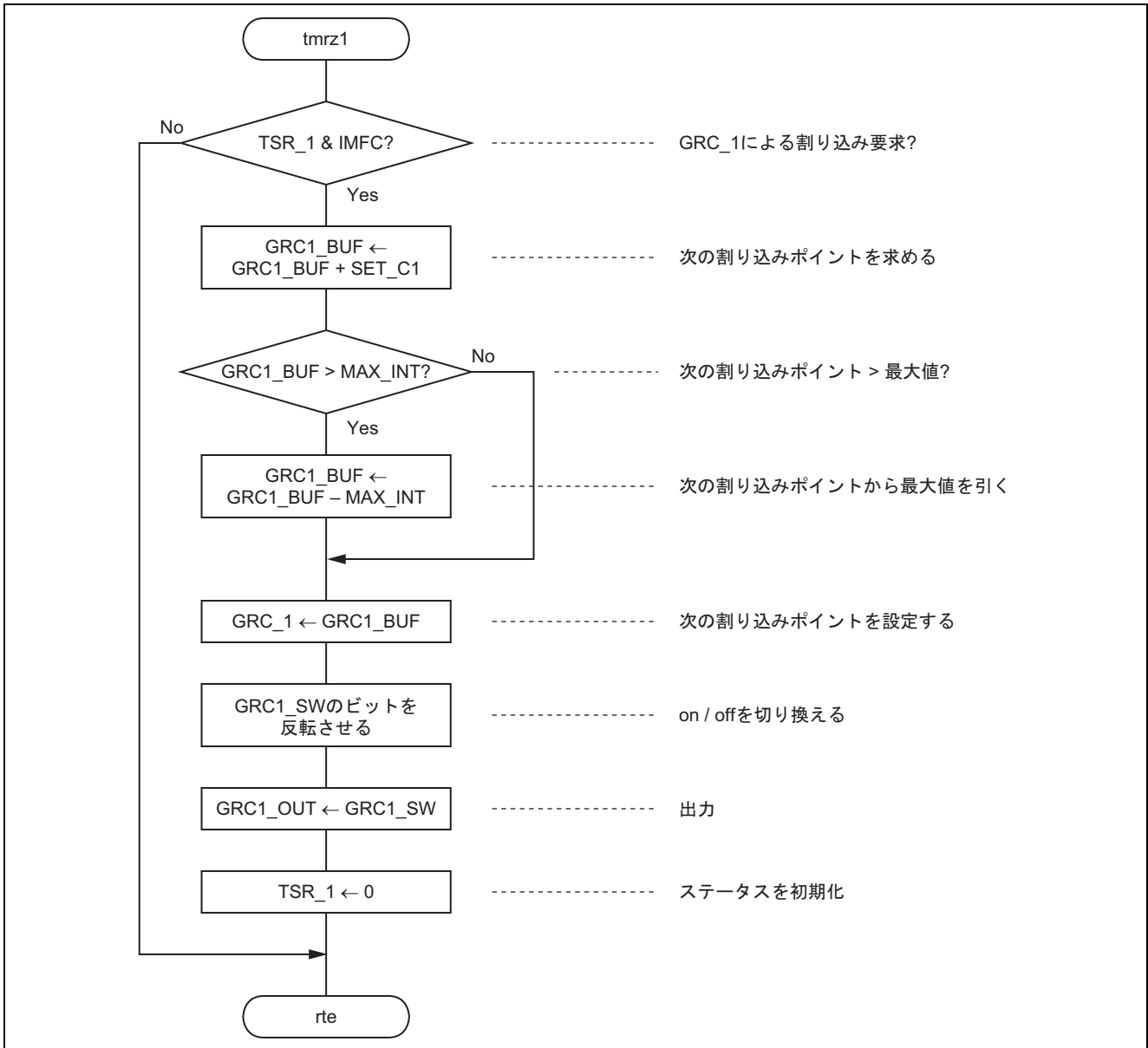




(2) タイマ Z0 割り込み処理ルーチン (tmrz0)



(3) タイマ Z1 割り込み処理ルーチン (tmrz1)



6. プログラムリスト

INIT.SRC (プログラムリスト)

```

.export      _INIT
.import      _main
;
.section     P, CODE
_INIT:
    mov.l     #h'fff000, sp
    ldc.b     #b'10000000, ccr
    jmp       @_main
;
.end
    
```

```

/* H8/300H Tiny Series -H8/36049- Application Note */
/* Application Version */
/* Usage Example of Internal Timer Z */

#include <machine.h>

/* Symbol definitions */
struct BIT {
    unsigned char b7:1;      /* bit 7 */
    unsigned char b6:1;      /* bit 6 */
    unsigned char b5:1;      /* bit 5 */
    unsigned char b4:1;      /* bit 4 */
    unsigned char b3:1;      /* bit 3 */
    unsigned char b2:1;      /* bit 2 */
    unsigned char b1:1;      /* bit 1 */
    unsigned char b0:1;      /* bit 0 */
};

#define PDR3      *(volatile unsigned char *)0xFFFFD6 /* Port Data Register 3 */
#define PDR3_BIT *(struct BIT *)0xFFFFD6 /* Port Data Register 3 */
#define GRC0_OUT  PDR3_BIT.b0 /* Output it by interrupt from GRC0 */
#define GRC1_OUT  PDR3_BIT.b1 /* Output it by interrupt from GRC1 */
#define PCR3      *(volatile unsigned char *)0xFFFFE6 /* Port Control Register 3 */

/* COMMON */
#define TSTR      *(volatile unsigned char *)0xFFF720 /* Timer Start Register */
#define TMDR      *(volatile unsigned char *)0xFFF721 /* Timer Mode Register */
#define TPMR      *(volatile unsigned char *)0xFFF722 /* Timer PWM Mode Register */
#define TFCR      *(volatile unsigned char *)0xFFF723 /* Timer Function Control Register */
#define TOER      *(volatile unsigned char *)0xFFF724 /* Timer Output Master Enable Register */
#define TOCR      *(volatile unsigned char *)0xFFF725 /* Timer Output Control Register */

/* CHANNEL 0 */
#define TCR_0     *(volatile unsigned char *)0xFFF700 /* Timer Control Register_0 */
#define TIORA_0   *(volatile unsigned char *)0xFFF701 /* Timer I/O Control RegisterA_0 */
#define TIORC_0   *(volatile unsigned char *)0xFFF702 /* Timer I/O Control RegisterC_0 */
#define TSR_0     *(volatile unsigned char *)0xFFF703 /* Timer Status Register_0 */
#define TIER_0    *(volatile unsigned char *)0xFFF704 /* Timer Interrupt Enable Register_0 */
#define POCR_0    *(volatile unsigned char *)0xFFF705 /* PWM Mode Output Level Control Register_0 */
/*
#define TCNT_0    *(volatile unsigned int *)0xFFF706 /* Timer Counter_0 */
#define GRA_0     *(volatile unsigned int *)0xFFF708 /* General Register A_0 */
#define GRB_0     *(volatile unsigned int *)0xFFF70A /* General Register B_0 */
#define GRC_0     *(volatile unsigned int *)0xFFF70C /* General Register C_0 */
    
```

```

#define GRD_0    *(volatile unsigned int *)0xFFFF70E    /* General Register D_0 */

/* CHANNEL 1 */
#define TCR_1    *(volatile unsigned char *)0xFFFF710  /* Timer Control Register_1 */
#define TIORA_1 *(volatile unsigned char *)0xFFFF711  /* Timer I/O Control RegisterA_1 */
#define TIORC_1 *(volatile unsigned char *)0xFFFF712  /* Timer I/O Control RegisterC_1 */
#define TSR_1    *(volatile unsigned char *)0xFFFF713  /* Timer Status Register_1 */
#define TIER_1   *(volatile unsigned char *)0xFFFF714  /* Timer Interrupt Enable Register_1 */
#define POCR_1   *(volatile unsigned char *)0xFFFF715  /* PWM Mode Output Level Control Register_1 */
/*
#define TCNT_1   *(volatile unsigned int *)0xFFFF716   /* Timer Counter_1 */
#define GRA_1    *(volatile unsigned int *)0xFFFF718   /* General Register A_1 */
#define GRB_1    *(volatile unsigned int *)0xFFFF71A   /* General Register B_1 */
#define GRC_1    *(volatile unsigned int *)0xFFFF71C   /* General Register C_1 */
#define GRD_1    *(volatile unsigned int *)0xFFFF71E   /* General Register D_1 */

#define IMFA     0x01    /* bit position of IMFA */
#define IMFC     0x04    /* bit position of IMFC */
#define GRB_MAX  4      /* GRB table size */

#define SET_A0   12000   /* setting value for GRA_0 */
#define SET_A1   4000    /* setting value for GRA_1 */
#define SET_B1   2000    /* setting value for GRB_1 */
#define SET_C0   2800    /* setting value for GRC_0 */
#define SET_C1   20000   /* setting value for GRC_1 */

#define MAX_INT  65535   /* integer max value */

#pragma interrupt (tmrz0)
#pragma interrupt (tmrz1)

/* function definition */
extern void INIT(void);    /* Stack pointer set */
void main(void);          /* main routine */
void tmrz0(void);         /* Timer Z0 interrupt routine */
void tmrz1(void);         /* Timer Z1 interrupt routine */

/* Data table */
const unsigned int GRB_DATA[GRB_MAX] =    /* GRB table */
{
    7000,
    3000,
    9000,
    5000
};

/* RAM definition */
unsigned char TSR_SW;    /* Save TSR */
int GRB_count;          /* GRB counter */
unsigned long GRA0_BUF; /* Buffer for GRA_0 */
unsigned long GRC0_BUF; /* Buffer for GRC_0 */
unsigned long GRC1_BUF; /* Buffer for GRC_1 */
int GRC0_SW;            /* Inverse switch for GRC_0 */
int GRC1_SW;            /* Inverse switch for GRC_1 */

/* Vector address */
#pragma section V1    /* Vector section set */
void (*const VEC_TBL1[])(void) = {
    INIT    /* H'0000 Reset vector */
};
#pragma section V2    /* Vector section set */
void (*const VEC_TBL2[])(void) = {

```

```

    tmrz0                                /* H'0068 Timer Z0 interrupt vector */
};
#pragma section V3                       /* Vector section set */
void (*const VEC_TBL3[])(void) = {
    tmrz1                                /* H'006C Timer Z1 interrupt vector */
};
#pragma section                          /* P */

/*****
/* Main program                          */
*****/
void main(void)
{
    set_imask_ccr(1);                    /* CCR I-bit = 1 */

    GRC0_SW = 0;                          /* initialize GRC_0 switch */
    GRC1_SW = 0;                          /* initialize GRC_1 switch */

    PCR3 = 0x03;                          /* initialize Port 3 */

    TSTR = 0;                             /* stop timer */
    TCR_0 = 0x20;                          /* select Internal clock(φ), Clears TCNT by
GRA */
    TCR_1 = 0x00;                          /* select Internal clock(φ), Disables TCNT
clearing */
    TFCR = 0x0D;                          /* operate in reset synchronous PWM mode */
                                           /* initial output is low */
    TOCR = 0;                              /* selects the initial outputs */
    TCNT_0 = 0;                            /* clear timer counter */
    TCNT_1 = 0;                            /* clear timer counter */
    GRA_0 = SET_A0;                         /* set period */
    GRA0_BUF = SET_A0;                     /* keep period */
    GRA_1 = SET_A1;                         /* set period */
    GRB_count = 0;                         /* init counter */
    GRB_0 = GRB_DATA[GRB_count];          /* set period */
    GRB_1 = SET_B1;                         /* set period */

    GRC_0 = SET_C0;                         /* set period */
    GRC0_BUF = SET_C0;                     /* keep period */

    TIER_0 = 0x05;                          /* enable GRA_0,GRC_0 interrupt */

    TOER = 0;                              /* enable output */

    GRC_1 = SET_C1;                         /* set period */
    GRC1_BUF = SET_C1;                     /* keep period */
    TIER_1 = 0x04;                          /* enable GRC_1 interrupt */

    TSTR = 0x03;                          /* start timer */

    set_imask_ccr(0);                      /* CCR I-bit = 0 */

    while(1);
}

/*****
/* Timer Z0 Interrupt                      */
*****/
void tmrz0(void)
{
    TSR_SV = TSR_0;                        /* save status */

```

```

/* interrupt by GRC_0 */
if(TSR_SV & IMFC) {
    GRC0_BUF += SET_C0; /* get next period */
    if(GRC0_BUF > GRA0_BUF) {
        GRC0_BUF -= GRA0_BUF;
    }
    GRC_0 = GRC0_BUF; /* set next period */

    GRC0_SW ^= 1; /* reverse switch */
    GRC0_OUT = GRC0_SW; /* output signal */
}

/* interrupt by GRA_0 */
if(TSR_SV & IMFA) {
    if(GRB_count < (GRB_MAX - 1)) /* get next index */
        GRB_count++;
    else
        GRB_count = 0;
    GRB_0 = GRB_DATA[GRB_count]; /* set next period from table */
}
TSR_0 = 0; /* clear status */
}

/*****/
/* Timer Z1 Interrupt */
/*****/
void tmrz1(void)
{
    /* interrupt by GRC_1 */
    if(TSR_1 & IMFC) {
        GRC1_BUF += SET_C1; /*get next period */
        if(GRC1_BUF > MAX_INT) {
            GRC1_BUF -= MAX_INT;
        }
        GRC_1 = GRC1_BUF; /*set next period */

        GRC1_SW ^= 1; /* reverse switch */
        GRC1_OUT = GRC1_SW; /* output signal */
    }
    TSR_1 = 0; /* clear status */
}

```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.08.12	—	初版発行

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すは、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。