

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ひ申し上げます。

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

## オートリロードタイマ機能による割込み周期設定

### 要旨

タイマ B1 のオートリロードタイマ機能を使用して、160 $\mu$ s ごとにタイマ B1 割り込みを発生させます。

### 動作確認デバイス

H8/300H 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.....	7
4.1 モジュール説明.....	7
4.2 引数の説明.....	7
4.3 使用内部レジスタ説明.....	8
4.4 使用 RAM 説明.....	8
5. フローチャート.....	9
6. プログラムリスト.....	11

## 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. タイマ B1 のオートリロードタイマ機能を使用して、160 $\mu$ s ごとにタイマ B1 割り込みを発生させます。
2. タイマ B1 割り込み処理の中で、タイマ B1 割り込みの回数をカウントし、250 回カウントするごとに LED の点灯または消灯を行ないます。

## 2. 使用機能説明

1. 本タスク例では、タイマ B1 オートリロード機能を使用して、160 $\mu$ s ごとにタイマ B1 割り込みを発生させます。

(a) 図 2.1 にタイマ B1 オートリロード機能のブロック図を示します。以下にタイマ B1 オートリロード機能のブロック図について説明します。

- システムクロック ( $\phi$ ) は、10MHz の OSC クロックを 2 分周した 5MHz 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。
- プリスケアラ S (PSS) は、 $\phi$  を入力とする 13 ビットのカウンタで、1 サイクルごとにカウントアップします。
- タイマモードレジスタ B1 (TMB1) は、8 ビットのリード/ライト可能なレジスタで、オートリロード機能の選択、および入力クロックの選択を行ないます。
- タイマカウンタ B1 (TCB1) は、8 ビットのリード可能なアップカウンタで、入力する内部クロック/外部イベントによりカウントアップされます。入力するクロックはシステムクロックの 8192 分周、2048 分周、512 分周、256 分周、64 分周、16 分周、4 分周および外部クロックの計 8 種類のクロックより選択可能です。本タスク例では、TCB1 の入力クロックにシステムクロックの 64 分周のクロックを選択しています。
- タイマロードレジスタ B1 (TLB1) は、8 ビットのライト専用のレジスタで、TCB1 のリロード値を設定します。本タスク例では、160 $\mu$ s で TCB1 がオーバーフローするように TLB1 を H'60 に設定します。
- タイマ B1 割り込み要求フラグ (IRRTB1) は、TCB1 がオーバーフローすることにより“1”にセットされます。IRRTB1 が 1 にセットされていて、割り込みイネーブルレジスタ 2 (IENR2) のタイマ B1 割り込み要求イネーブル (IENTB1) が“1”で、かつコンディションコードレジスタ (CCR) の I ビットが“0”にクリアされている場合にタイマ B1 割り込みが受け付けられ、タイマ B1 割り込み処理を開始します。

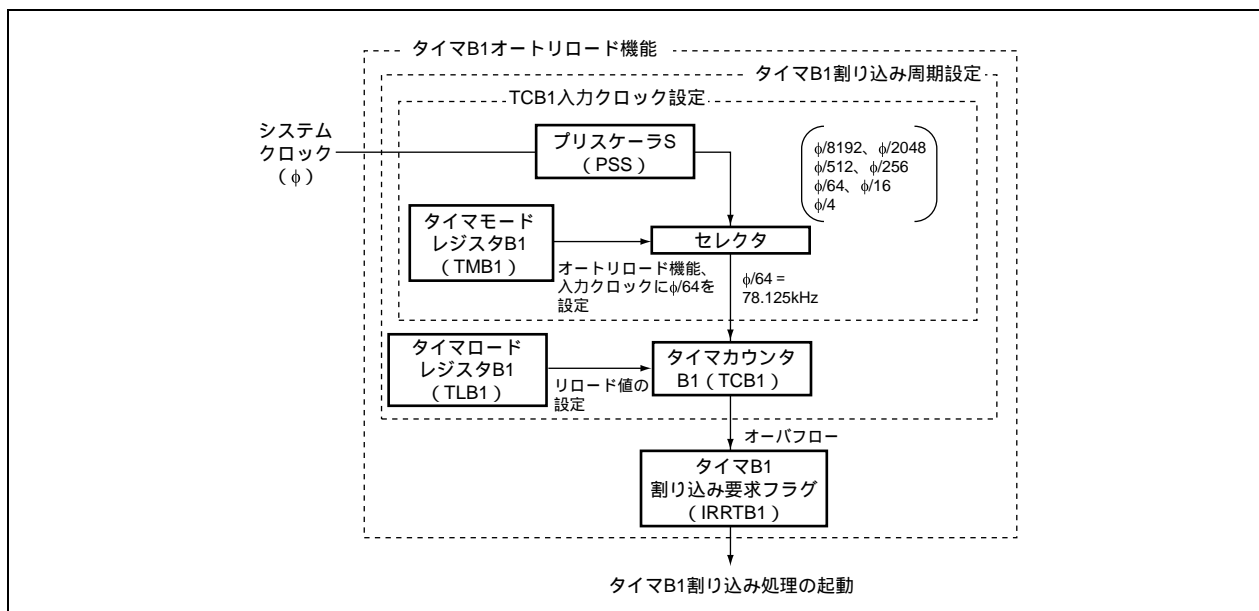


図 2.1 タイマ B1 オートリロード機能ブロック図



- (b) タイマ B1 オートリロード機能による割り込み周期の設定方法について以下に説明します。
- タイマ B1 オートリロード機能による割り込み周期は以下の式によって設定します。  
 タイマ B1 割り込み周期 (s) = (TCB1 入力クロック周期 (s)) × (256 - (リロード設定値))
  - 上記の式によって設定した TCB1 入力クロック周期を TMB1 に、リロード設定値を TLB1 にそれぞれ設定することにより、タイマ B1 割り込み周期を設定します。
2. 表 2.1 に本タスク例の機能割り付けを示します。表 2.1 に示すように機能を割り付け、タイマ B1 オートリロード機能による割り込み周期設定を行いません。

表 2.1 機能割り付け

機能	機能割り付け
PSS	システムクロックを入力とする 13 ビットのカウンタ
TCB1	システムクロックの 64 分周を入力とする 8 ビットのカウンタ
TMB1	オートリロード機能の設定、および TCB1 入力クロックを $\phi/64$ に設定
TLB1	TCB1 のリロード値を設定
IRRTB1	タイマ B1 割り込み要求の有無を反映
PCR7	P73 出力端子機能の設定
PDR7	P73 出力端子のデータの格納
P73	LED 出力

### 3. 動作説明

図 3.1 に動作原理を示します。図 3.1 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タイマ B1 オートリロード機能による割り込み周期設定を行ないます。

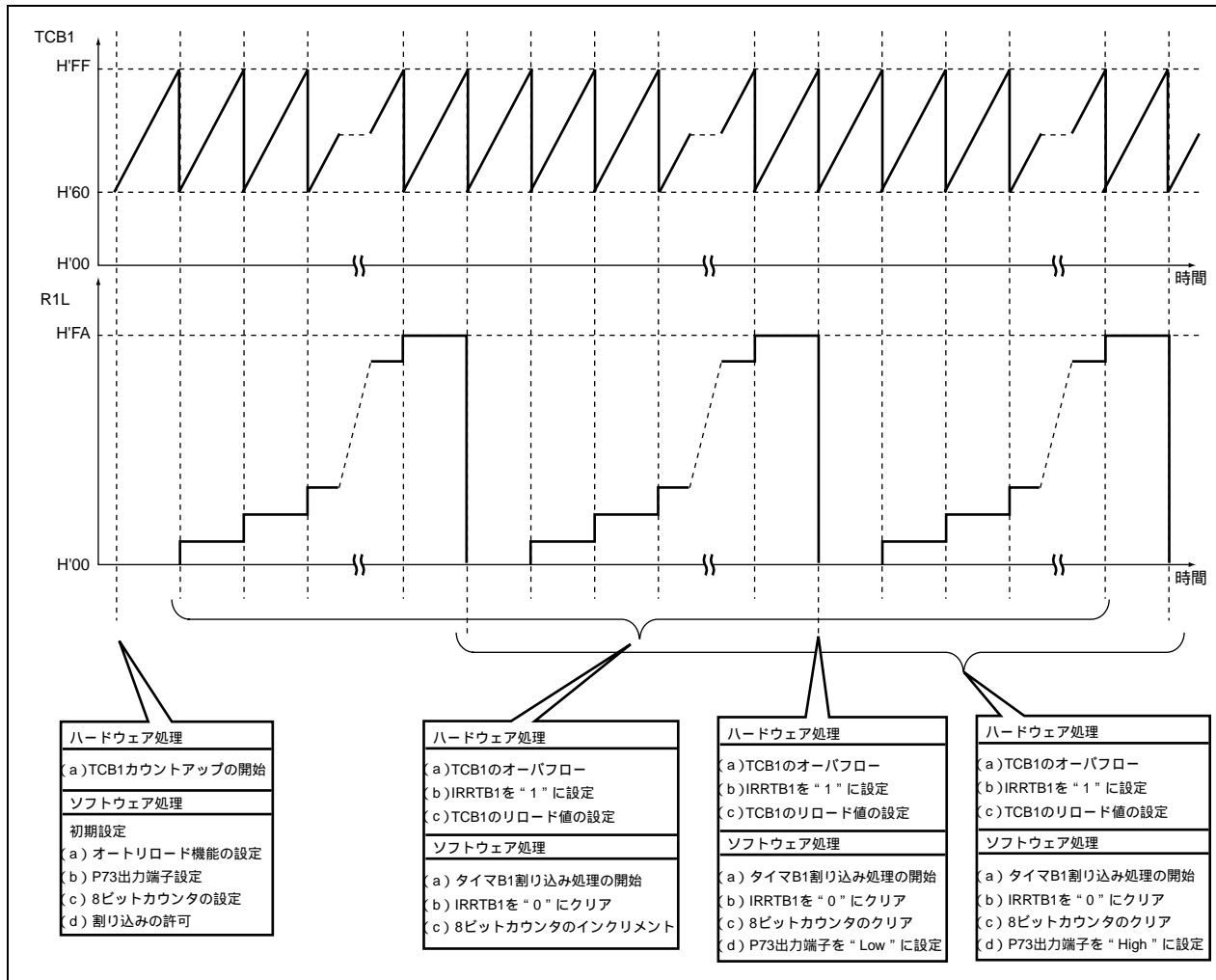


図 3.1 タイマ B1 オートリロード機能による割り込み周期設定の動作原理

## 4. ソフトウェア説明

### 4.1 モジュール説明

本タスク例のモジュールを表 4.1 に示します。

表 4.1 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	スタックポインタのイニシャライズ、オートリロード機能の設定、ポート 73 の設定、8 ビットカウンタの設定、割り込みの許可、LED の点灯／消灯を行なう
カウントアップ	TB1INT	8 ビットカウンタのインクリメント、および 8 ビットカウンタのカウント値が H'FA になった時点で CTEDF を 1 にセットする

### 4.2 引数の説明

本タスク例では、引数を使用しません。

### 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表 4.2 に示します。

表 4.2 使用内部レジスタ説明

レジスタ名		機 能	アドレス	設定値
IRR2	IRRTB1	割り込み要求レジスタ 1 (タイマ B1 割り込み要求フラグ) : IRRTB1=0 のとき、タイマ B1 割り込みが要求されていない : IRRTB1=1 のとき、タイマ B1 割り込みが要求されている	H'FFF7 ビット 5	0
IENR2	IENRB1	割り込みイネーブルレジスタ 1 (タイマ B1 割り込みイネーブル) : IENRB1=1 のとき、タイマ B1 割り込み要求を許可	H'FFF5 ビット 5	1
TMB1		タイマモードレジスタ B1 : TMB1=H'FC のとき、タイマ B1 機能をオートリロード機能に、TCB1 入力クロックにシステムクロックの 64 分周のクロックに設定	H'FFB2	H'FC
TCB1		タイマカウンタ B1 : システムクロックを 64 分周したクロックを入力とする 8 ビットのアップカウンタ	H'FFB3	H'00
TLB1		タイマロードレジスタ B1 : TLB1=H'60 に設定すると、TCB1 は H'60 から カウントアップを開始し、TCB1 がオーバーフローすると TCB1 に H'60 がロードされます	H'FFB3	H'60
PDR7	P73	ポートデータレジスタ 7 (ポートデータレジスタ 73) : P73=0 のとき、P73 端子の出力レベルは Low : P73=1 のとき、P73 端子の出力レベルは High	H'FFDA ビット 3	0
PCR7	PCR73	ポートコントロールレジスタ 7 (ポートコントロールレジスタ 73) : PCR73=1 のとき、P73 端子を出力端子に設定	H'FFEA ビット 3	1

### 4.4 使用 RAM 説明

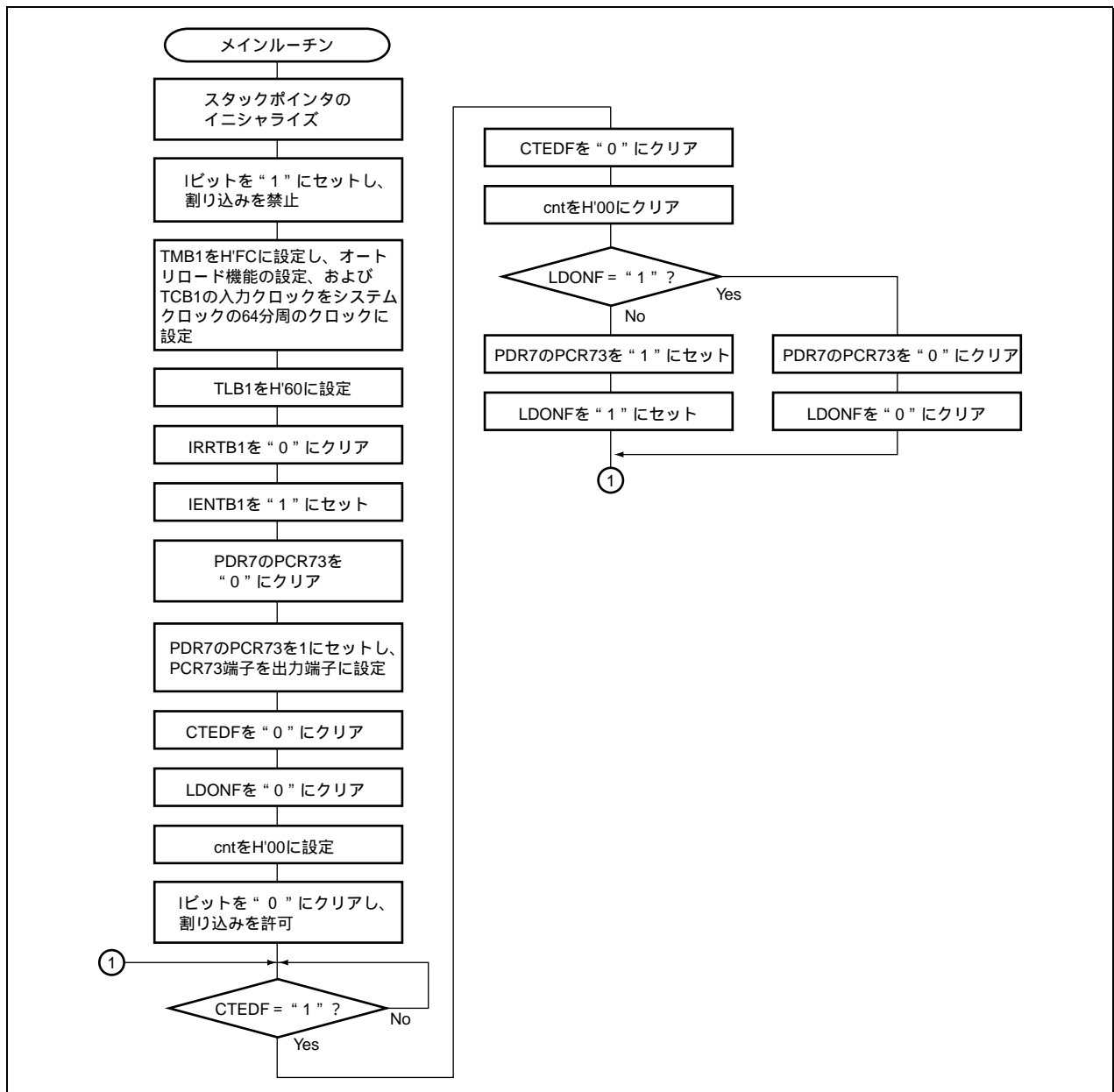
表 4.3 に本タスク例における使用 RAM 説明を示します。

表 4.3 使用 RAM 説明

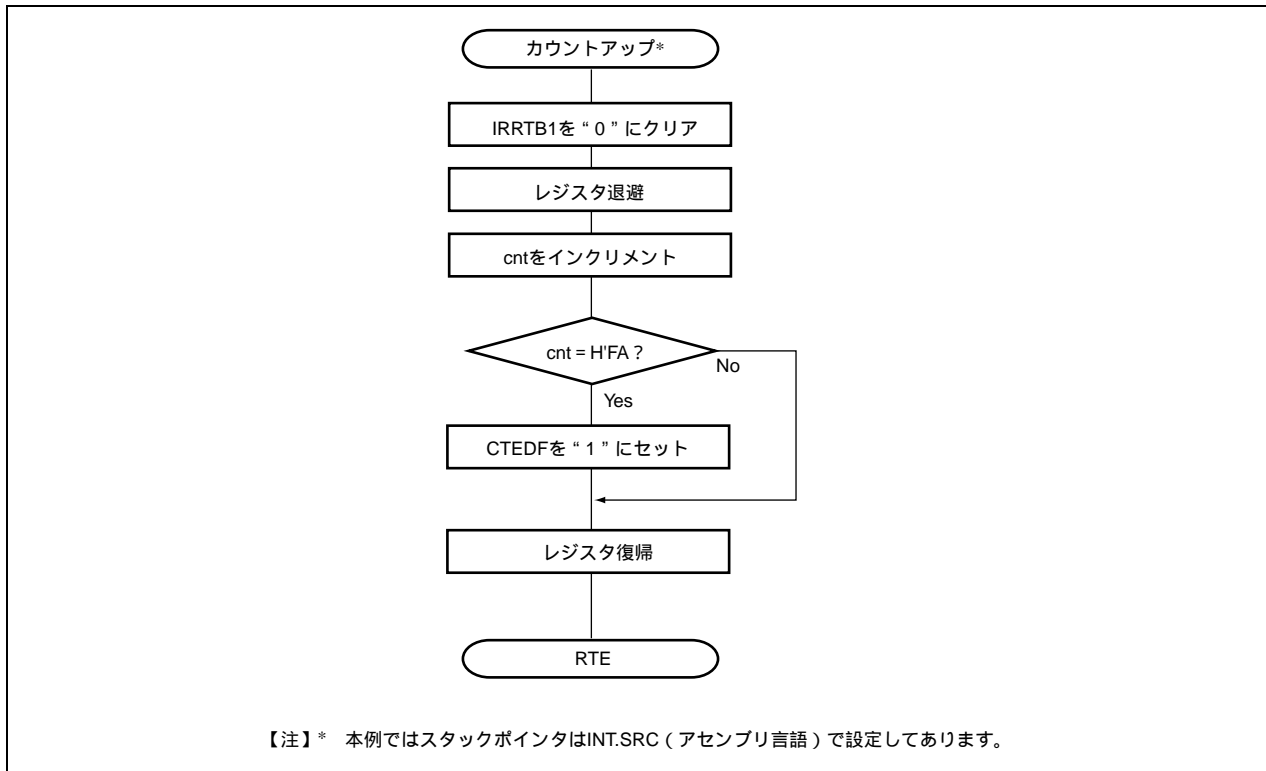
ラベル名		機 能	アドレス	使用モジュール名
USRF	CTEDF	8 ビットカウンタのカウンタ値が H'FA に なったかを判定するフラグ	H'FB80 ビット 0	メインルーチン カウントアップ
	LDONF	LED の ON/OFF を判定するフラグ	H'FB80 ビット 1	メインルーチン
cnt		8 ビットカウンタ	H'FB81	カウントアップ

## 5. フローチャート

### (a) メインルーチン



## (b) タイマ B1 割り込み処理ルーチン



## 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.W   #H'FF80,R7
        LDC.B   #B'10000000,CCR
        JMP     @_main

;

        .END

```

```

/*****
/*
/*      H8/300HN Series -H8/3687-
/*      Application Note
/*
/*      'Interrupt Period Setting by Auto Reload Timer Function'
/*
/*      Function
/*      :Timer B1 Auto Reload
/*
/*      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 TMB                *(volatile unsigned char *)0xF760 /* Timer Mode Register B1 */
#define TCB1               *(volatile unsigned char *)0xF761 /* Timer Counter B1 */
#define TLB1               *(volatile unsigned char *)0xF761 /* Timer Load Register */

#define IEGR1_BIT          (*(struct BIT *)0xFFF2) /*Interrupt Edge Select Register 1*/
#define IEG3               IEGR1_BIT.b3 /* IRQ3 Edge Select */
#define IEG2               IEGR1_BIT.b2 /* IRQ2 Edge Select */
#define IEG1               IEGR1_BIT.b1 /* IRQ1 Edge Select */
#define IEG                 IEGR1_BIT.b0 /* IRQ0 Edge Select */
#define IENR1              *(volatile unsigned char *)0xFFF4 /* Interrupt Enable Register 1 */
#define IENR1_BIT          (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 1 */
#define IEN3               IENR1_BIT.b3 /* IRQ3 Interrupt Enable */
#define IEN2               IENR1_BIT.b2 /* IRQ2 Interrupt Enable */
#define IEN1               IENR1_BIT.b1 /* IRQ1 Interrupt Enable */
#define IEN0               IENR1_BIT.b0 /* IRQ0 Interrupt Enable */
#define IENR2              *(volatile unsigned char *)0xFFF5 /* Interrupt Enable Register 1 */
#define IENR2_BIT          (*(struct BIT *)0xFFF5) /* Interrupt Enable Register 1 */
#define IENTB1             IENR2_BIT.b5 /* IRQ3 Interrupt Enable */
#define IRR1               *(volatile unsigned char *)0xFFF6 /* Interrupt Flag Register 1 */
#define IRR1_BIT           (*(struct BIT *)0xFFF6) /* Interrupt Flag Register 1 */
#define IRRI3              IRR1_BIT.b3 /* IRQ3 Interrupt Request Flag */
#define IRRI2              IRR1_BIT.b2 /* IRQ2 Interrupt Request Flag */
#define IRRI1              IRR1_BIT.b1 /* IRQ1 Interrupt Request Flag */
#define IRRI0              IRR1_BIT.b0 /* IRQ0 Interrupt Request Flag */
#define IRR2               *(volatile unsigned char *)0xFFF7 /* Interrupt Flag Register 2 */
#define IRR2_BIT           (*(struct BIT *)0xFFF7) /* Interrupt Flag Register 2 */
#define IRRTB1             IRR2_BIT.b5 /* TMB1 Interrupt Request Flag */
#define PDR7               *(volatile unsigned char *)0xFFDA /* Port7 Data Register */
#define PDR7_BIT           (*(struct BIT *)0xFFDA) /* Port7 Data Register */
#define P7B3               PDR7_BIT.b3 /* Port Data Register 73 */
#define PCR7_BIT           (*(struct BIT *)0xFFDA) /* Port7 Control Register */
#define PCR73              PDR7_BIT.b3 /* Port73 Control Register */

#pragma interrupt          (TB1INT)

```



```

/*****
/*      関数定義
/*****
extern      void          INIT ( void );          /* SP Set          */
void        mai          ( void );
void        TBLINT       ( void );

/*****
/*      RAM define
/*****
unsigned char USRF;          /* User Flag Area    */
unsigned char USRF_1;      /* User Flag Area    */
unsigned char cnt;         /* Counter           */

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) =
          TBLINT          /* 29 Timer B1 Interrupt */
};

/*****
/*
/*      H8/300H Tiny Series -H8/3664-
/*      Application Note
/*
/*      'Port B 8 bit Parallel Input Port Read
/*
/*      Function
/*      : Port B Read
/*
/*      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    ADDRA    *(volatile unsigned int *)0xFFB0    /*
A/D Data Register A    */
#define    ADDR8    *(volatile unsigned int *)0xFFB2    /*
A/D Data Register B    */
#define    ADDR4    *(volatile unsigned int *)0xFFB4    /*
A/D Data Register C    */
#define    ADDR0    *(volatile unsigned int *)0xFFB6    /*
A/D Data Register D    */
#define    ADCSR    *(volatile unsigned char *)0xFFB8    /*
A/D Control/Status Register    */
#define    ADCSR_BIT    (*(struct BIT *)0xFFB8) /* A/D Control/Status Register    */
#define    ADF    ADCSR_BIT.b7    /* A/D END Flag    */
#define    ADIE    ADCSR_BIT.b6    /* A/D Interrupt Enable    */
#define    ADST    ADCSR_BIT.b5    /* A/D Start    */
#define    SCAN    ADCSR_BIT.b4    /* A/D Scan Mode    */
#define    CKS    ADCSR_BIT.b3    /* A/D Clock Select    */
#define    CH2    ADCSR_BIT.b2    /* Channel Select 2    */
#define    CH1    ADCSR_BIT.b1    /* Channel Select 1    */
#define    CH0    ADCSR_BIT.b0    /* Channel Select 0    */
#define    PDRB    *(volatile unsigned char *)0xFFDD    /*
Port Data Register B    */

/*****
/*          関数定義                                         */
*****/

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

```

```

/*****
/*      Vector Address      */
/*****

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

/*****
/*      Main Program      */
/*****

void main ( void )
{
    unsigned char    buffer;
    unsigned char    PB_Data;

    ADCSR = 0x00;          /* A/D Enable Channel Initialize */
    buffer = PDRB;        /* Read Port B */
    ADCSR = 0x01;          /* A/D Enable Channel change */
    buffer |= PDRB;       /* Read Port B */
    PB_Data = buffer;     /* PB_Data <- Port B Status */

    while(1){
        ;
    }
}
#pragma section          /* P          */

```

```

/*****
/*  Main Program
/*****

void main ( void )
{
    _INITSCT();

    set_imask_ccr(1);                /* Interrupt Disable                */

    TMB1 = 0xFC;                    /* Initialize Timer B1 Function & Input Clock */
    TLB1 = 0x60;                    /* Initialize TCBl reload Value          */
    IRRTB1 = 0;                    /* Interrupt Flag Initialize            */
    IENTB1 = 1;                    /* Interrupt Enable Flag Set            */

    P7B3 = 0;
    PCR73 = 1;                    /* Initialize P73 Output Terminal Function */

    USRF = 0x00;                    /* Initialize User Flag Erea            */
    USRF_1 = 0x00;                /* Initialize User Flag Erea            */
    cnt = 0x00;                    /* Initialize 8 bit counter              */

    set_imask_ccr(0);                /* Interrupt Enable                    */

    while(1)
    {

        while(USRF == 0x01)          /* CTEDF = "1"?                        */
        {
            USRF = 0x00;            /* Clear CTEDF                          */

            cnt = 0x00;            /* Initialize 8 bit counter              */

            if (USRF_1 == 0x00)      /* LDONF = "1"?                          */
            {
                P7B3 = 1;          /* Turn on LED                            */
                USRF_1 = 0x01;      /* Set LDONF                              */
            }
            else
            {
                P7B3 = 0;          /* Turn off LED                            */
                USRF_1 = 0x00;      /* Clear LDONF                            */
            }
        }
    }
}

```

```

    }
}

/*****
/* TimerB1 Interrupt
*****/

void TB1INT( void )
{
    IRRTB1 = 0;                /* Clear IRRTB1 */
    cnt = cnt+1;              /* Increment 8 bit Counter */

    if(cnt == 0xFA)          /* 8 bit Counter = H'FA */
    {
        USRF = 0x01;        /* Set CTEDF */
    }
}

#pragma section              /* P */
/*****
/* Main Program
*****/

void main ( void )
{
    _INITSCT();

    set_imask_ccr(1);        /* Interrupt Disable */

    TMB1 = 0xFC;            /* Initialize Timer B1 Function & Input Clock */
    TLB1 = 0x60;            /* Initialize TCB1 reload Value */
    IRRTB1 = 0;            /* Interrupt Flag Initialize */
    IENTB1 = 1;            /* Interrupt Enable Flag Set */

    P7B3 = 0;
    PCR73 = 1;              /* Initialize P73 Output Terminal Function */

    USRF = 0x00;            /* Initialize User Flag Area */
    USRF_1 = 0x00;         /* Initialize User Flag Area */
    cnt = 0x00;            /* Initialize 8 bit counter */

    set_imask_ccr(0);        /* Interrupt Enable */

    while(1)
    {

```

```

while(USRF == 0x01)                                /* CTEDF = "1"? */
{
    USRF = 0x00;                                    /* Clear CTEDF */

    cnt = 0x00;                                     /* Initialize 8 bit counter */

    if (USRF_1 == 0x00)                            /* LDONF = "1"? */
    {
        P7B3 = 1;                                  /* Turn on LED */
        USRF_1 = 0x01;                             /* Set LDONF */
    }
    else
    {
        P7B3 = 0;                                  /* Turn off LED */
        USRF_1 = 0x00;                             /* Clear LDONF */
    }
}
}

/*****
/*   TimerB1 Interrupt
*****/

void TB1INT( void )
{
    IRRTB1 = 0;                                     /* Clear IRRTB1 */
    cnt = cnt+1;                                    /* Increment 8 bit Counter */

    if(cnt == 0xFA)                                 /* 8 bit Counter = H'FA */
    {
        USRF = 0x01;                               /* Set CTEDF */
    }
}

```

## リンクアドレス指定

セクション名	アドレス
CV1	0x0000
P	0x0100
B	0xFB80