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ものですが、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て、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が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。

M32C/85 グループ

DMAC を使用した多チャネル PWM の応用例

1. 要約

この資料では、DMAC を使用した多チャネルＰWM出力を行う手順と使用例を紹介しています。

2. はじめに

この資料で説明する応用例は、次のマイコン、条件での利用に適用されます。
・マイコン　　:M32C/85 グループ

M32C/85 グループと同様の SFR(周辺装置制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

3. 使用例の説明

ROM 上に PWM 出力パターンを用意しておき、タイマ B0 と DMAC を 4 本使用して、タイマ B0 のアンダフローのタイミング毎にポート P0～P7 から 64 本の PWM 出力を行う応用例を説明します。

使用例：

・ システム条件
VCC1=VCC2=5.0V 、XIN=8MHz 、PLL=4 逓倍 、f1=32MHz

・ DMAC 設定

	DMA 要求要因	転送モード	転送単位	転送方向
DMA0	タイマ B0 割り込み要求	リピート転送	16 ビット	メモリ(順方向)→固定番地(P0) (P0～P1 に 16 ビット単位で出力)
DMA1				メモリ(順方向)→固定番地(P2) (P2～P3 に 16 ビット単位で出力)
DMA2				メモリ(順方向)→固定番地(P4) (P4～P5 に 16 ビット単位で出力)
DMA3				メモリ(順方向)→固定番地(P6) (P6～P7 に 16 ビット単位で出力)

・ TB0 設定
タイマモード、カウントソース=f1(32MHz)、周期(100μs～900μs の範囲で変更する)

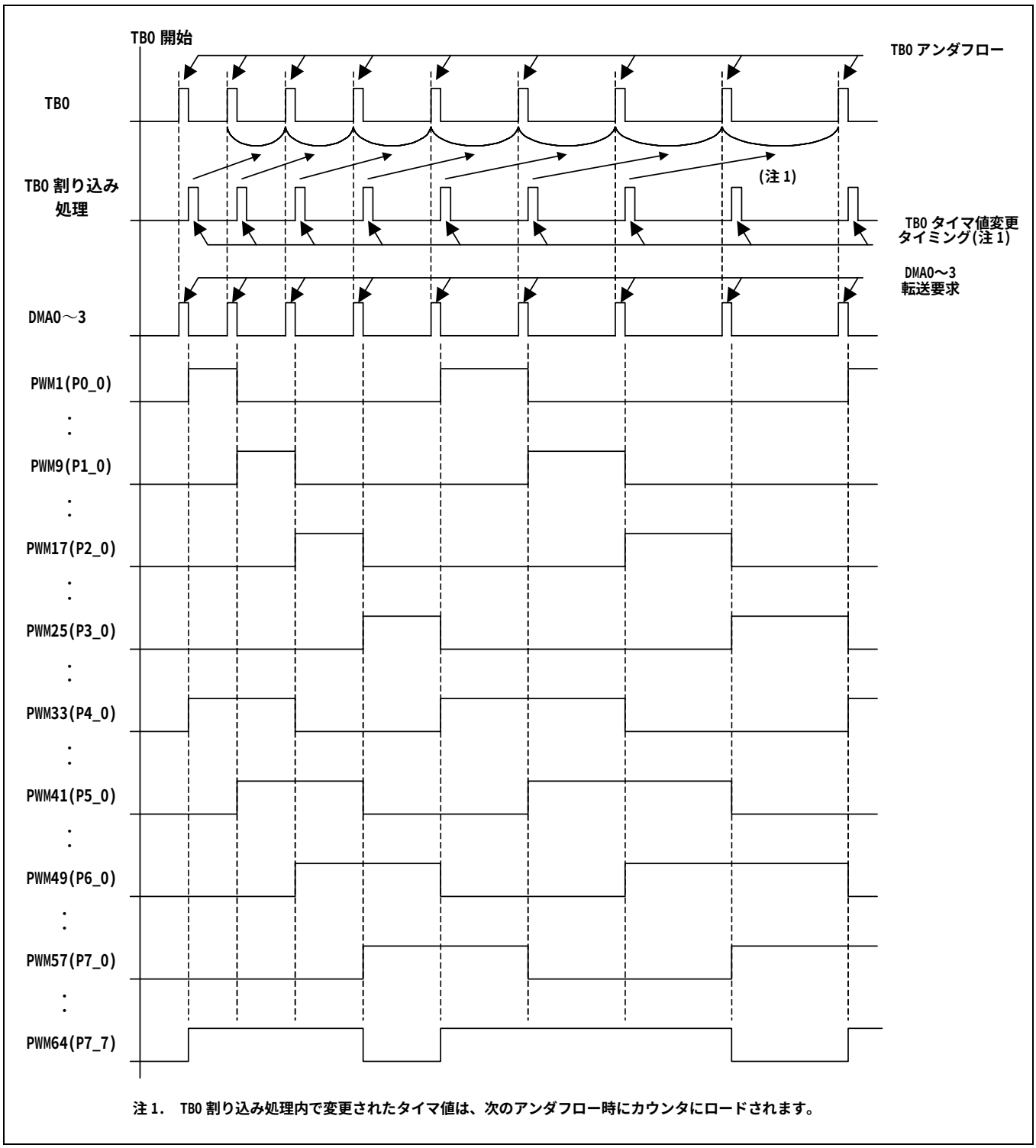
動作：

タイマ B0 のアンダフロー周期毎に、DMA 転送で、出力ポートとして機能している P0～P7 の 64 本のポートに PWM 出力します(P7_0、P7_1 は N チャネルオープンドレイン出力なので、抵抗を介して VCC1 に接続(プルアップ)してください)。
また、タイマ B0 割り込み処理内で、タイマ B0 のタイマ値を変更することで、PWM パルス出力幅を変更します。
図 1 に PWM 出力タイミングチャートを示します。

注意事項：

タイマ B と DMAC を組み合わせてタイマパルス出力を行う場合、下記の点に注意してください。

・ DMA 転送優先順位
DMA 要求が同時に発生した場合、DMA0＞DMA1＞DMA2＞DMA3 の優先順位で転送されます。
・ DMA 転送の遅延サイクル数
本使用例の条件(プログラム配置領域=内部 ROM、転送元=内部 ROM、転送先=SFR 領域)で転送を行った場合、CPU クロックの０～５サイクル転送に遅延が発生します。(遅延するサイクル数は、外部メモリを使用した場合など、メモリのウェイト数により変化します)
・ タイマ B スタート時の遅延時間
最初の、タイマ B スタート時のパルス出力は、ポートの方向レジスタを出力設定にしてからタイマ B をスタートさせるまでの命令の実行時間が遅延時間となります。
・ パルス出力の遅延時間
実際のパルス出力は、タイマ B の割り込み要求発生後、DMA 転送サイクル数＋DMA 転送遅延サイクル数分遅れて出力されます。

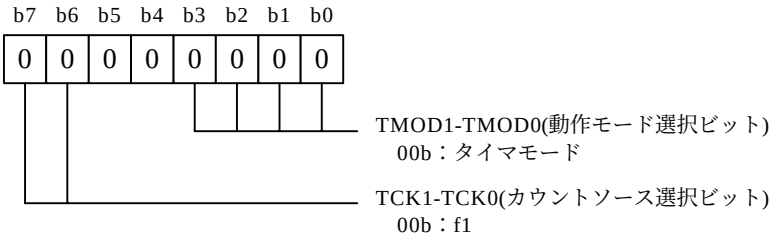


注 1. TB0 割り込み処理内で変更されたタイマ値は、次のアンダフロー時にカウンタにロードされます。

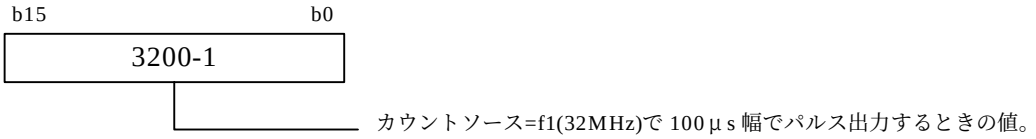
図 1. PWM 出力タイミングチャート

(1)　タイマ B0(DMA0～3 の要求要因)設定

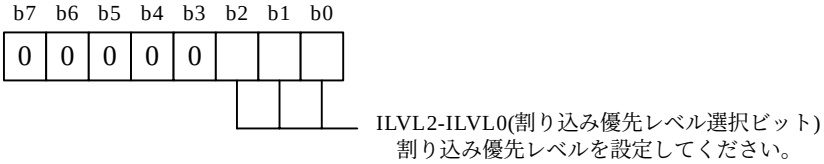
・　TB0MR レジスタ(タイマ B0 モードレジスタ)を設定する。



・　TB0(タイマ B0 レジスタ)に PWM パルス幅初期値を設定する。

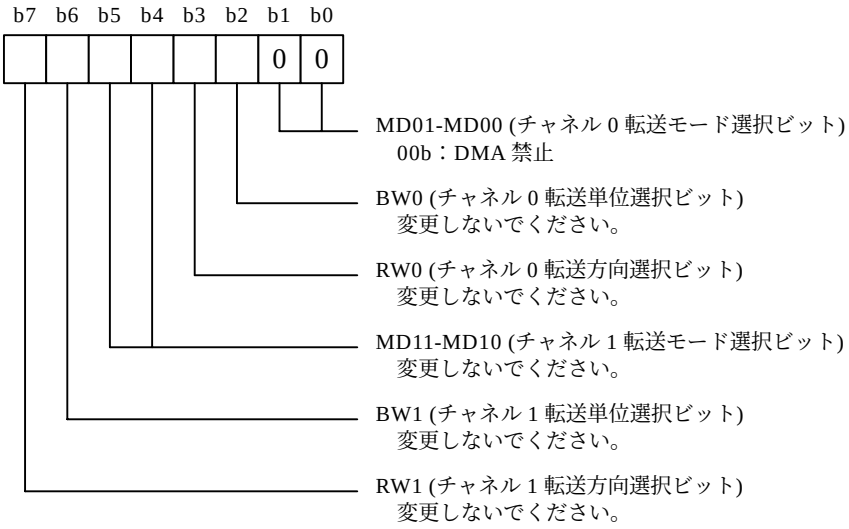


・　TB0IC(タイマ B0 割り込み制御レジスタ)を設定する。

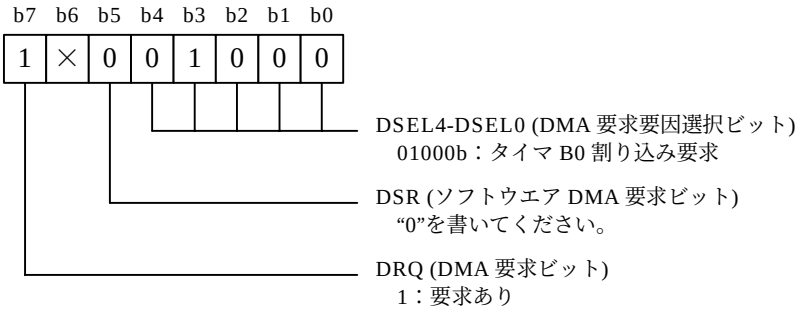


(2) DMA0 設定

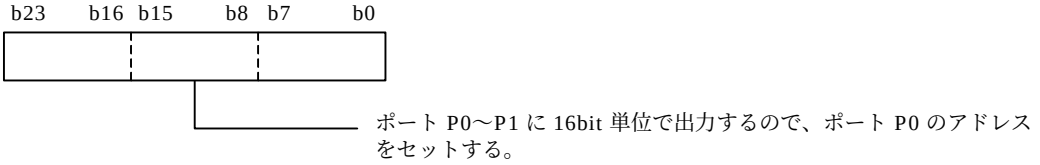
・ DMD0 レジスタ(DMA モードレジスタ 0)を設定(DMA0 を禁止)する。



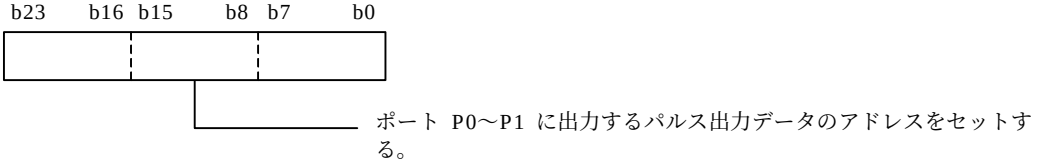
・ DM0SL レジスタ(DMA0 要因選択レジスタ)を設定する。



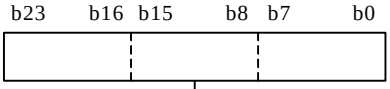
・ DSA0 レジスタ(DMA0 SFR アドレスレジスタ)を設定する。



・ DRA0 レジスタ(DMA0 メモリアドレスリロードレジスタ)を設定する。

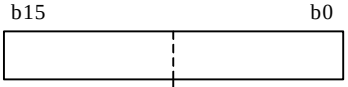


・ DMA0 レジスタ(DMA0 メモリアドレスレジスタ)を設定する。



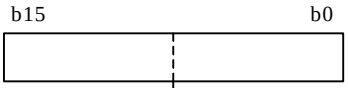
ポート P0～P1 に出力するパルス出力データのアドレスをセットする。

・ DRC0 レジスタ(DMA0 転送カウントリロードレジスタ)を設定する。



パルス出力データ数をセットする。

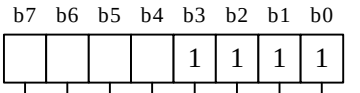
・ DCT0 レジスタ(DMA0 転送カウントレジスタ)を設定する。



パルス出力データ数をセットする。

・ ダミーサイクルを挿入する。
DMOSL レジスタを設定してから、BCLK で 6 サイクル分待ってから DMA を許可してください。ここでは、NOP を 6 個挿入することで、6 サイクル分待ちます。

・ DMD0 レジスタ(DMA モードレジスタ 0)を再設定(DMA0 を許可)する。



MD01-MD00 (チャネル 0 転送モード選択ビット)
11b：リピート転送

BW0 (チャネル 0 転送単位選択ビット)
1：16 ビット

RW0 (チャネル 0 転送方向選択ビット)
1：メモリ(順方向)→固定番地

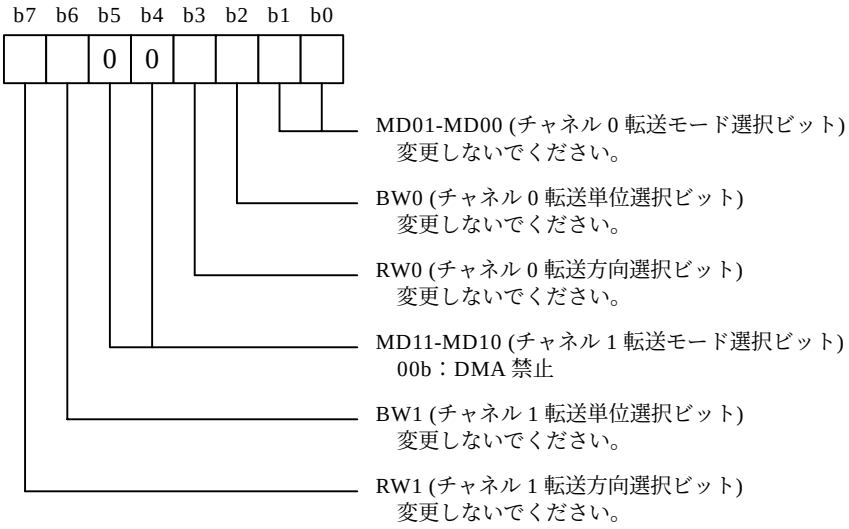
MD11-MD10 (チャネル 1 転送モード選択ビット)
変更しないでください。

BW1 (チャネル 1 転送単位選択ビット)
変更しないでください。

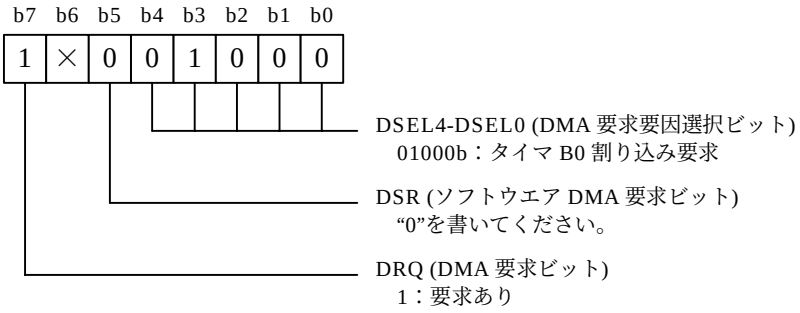
RW1 (チャネル 1 転送方向選択ビット)
変更しないでください。

(3) DMA1 設定

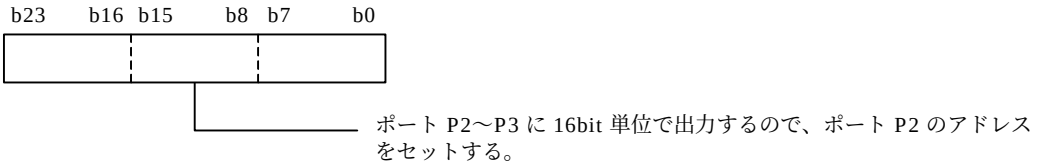
・ DMD0 レジスタ(DMA モードレジスタ 0)を設定(DMA1 を禁止)する。



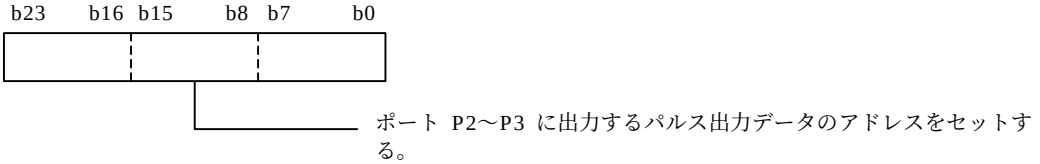
・ DM1SL レジスタ(DMA1 要因選択レジスタ)を設定する。



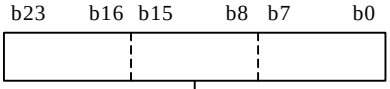
・ DSA1 レジスタ(DMA1 SFR アドレスレジスタ)を設定する。



・ DRA1 レジスタ(DMA1 メモリアドレスリロードレジスタ)を設定する。

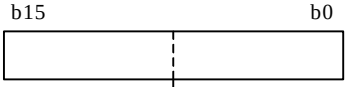


・ DMA1 レジスタ(DMA1 メモリアドレスレジスタ)を設定する。



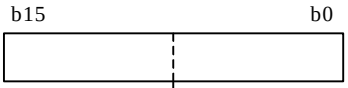
ポート P2～P3 に出力するパルス出力データのアドレスをセットする。

・ DRC1 レジスタ(DMA1 転送カウントリロードレジスタ)を設定する。



パルス出力データ数をセットする。

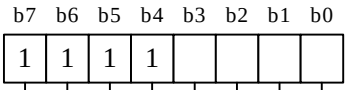
・ DCT1 レジスタ(DMA1 転送カウントレジスタ)を設定する。



パルス出力データ数をセットする。

・ ダミーサイクルを挿入する。
DM1SL レジスタを設定してから、BCLK で 6 サイクル分待ってから DMA を許可してください。ここでは、NOP を 6 個挿入することで、6 サイクル分待ちます。

・ DMD0 レジスタ(DMA モードレジスタ 0)を再設定(DMA1 を許可)する。



MD01-MD00 (チャネル 0 転送モード選択ビット)
変更しないでください。

BW0 (チャネル 0 転送単位選択ビット)
変更しないでください。

RW0 (チャネル 0 転送方向選択ビット)
変更しないでください。

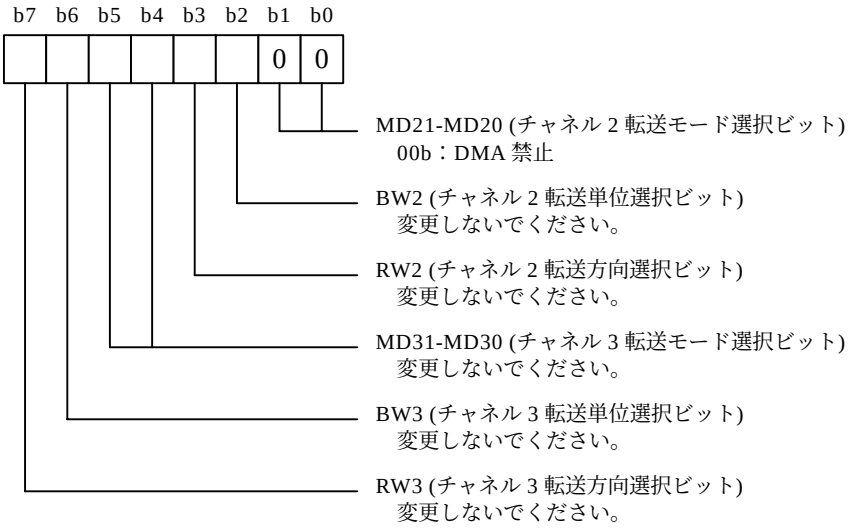
MD11-MD10 (チャネル 1 転送モード選択ビット)
11b：リピート転送

BW1 (チャネル 1 転送単位選択ビット)
1：16 ビット

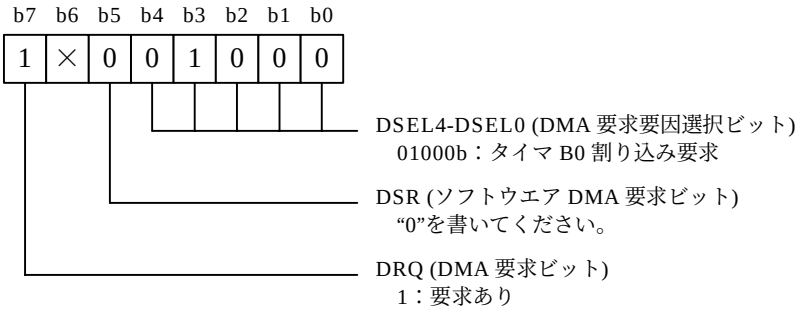
RW1 (チャネル 1 転送方向選択ビット)
1：メモリ(順方向)→固定番地

(4) DMA2 設定

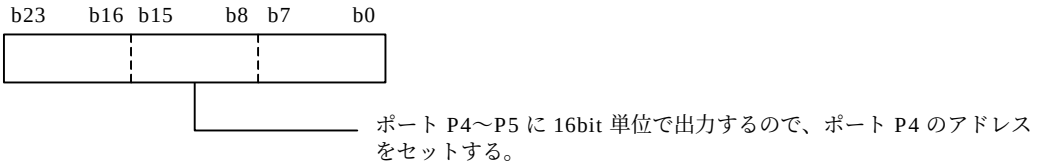
・ DMD1 レジスタ(DMA モードレジスタ 1)を設定(DMA2 を禁止)する。



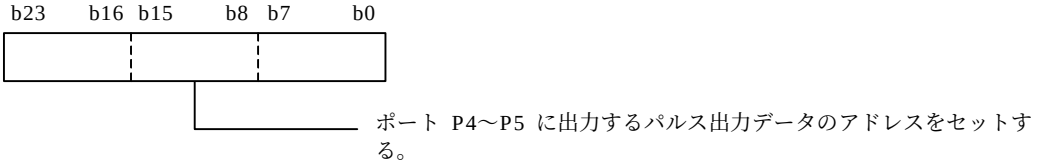
・ DM2SL レジスタ(DMA2 要因選択レジスタ)を設定する。



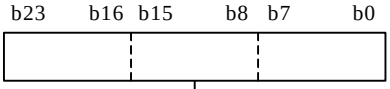
・ DSA2 レジスタ(DMA2 SFR アドレスレジスタ)を設定する。



・ DRA2 レジスタ(DMA2 メモリアドレスリロードレジスタ)を設定する。

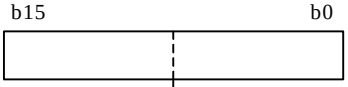


・ DMA2 レジスタ(DMA2 メモリアドレスレジスタ)を設定する。



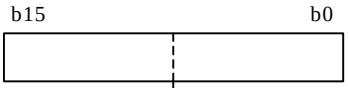
ポート P4～P5 に出力するパルス出力データのアドレスをセットする。

・ DRC2 レジスタ(DMA2 転送カウントリロードレジスタ)を設定する。



パルス出力データ数をセットする。

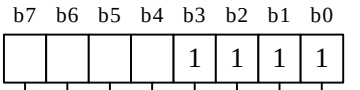
・ DCT2 レジスタ(DMA2 転送カウントレジスタ)を設定する。



パルス出力データ数をセットする。

・ ダミーサイクルを挿入する。
DM2SL レジスタを設定してから、BCLK で 6 サイクル分待ってから DMA を許可してください。ここでは、NOP を 6 個挿入することで、6 サイクル分待ちます。

・ DMD1 レジスタ(DMA モードレジスタ 1)を再設定(DMA2 を許可)する。



MD21-MD20 (チャネル 2 転送モード選択ビット)
11b：リピート転送

BW2 (チャネル 2 転送単位選択ビット)
1：16 ビット

RW2 (チャネル 2 転送方向選択ビット)
1：メモリ(順方向)→固定番地

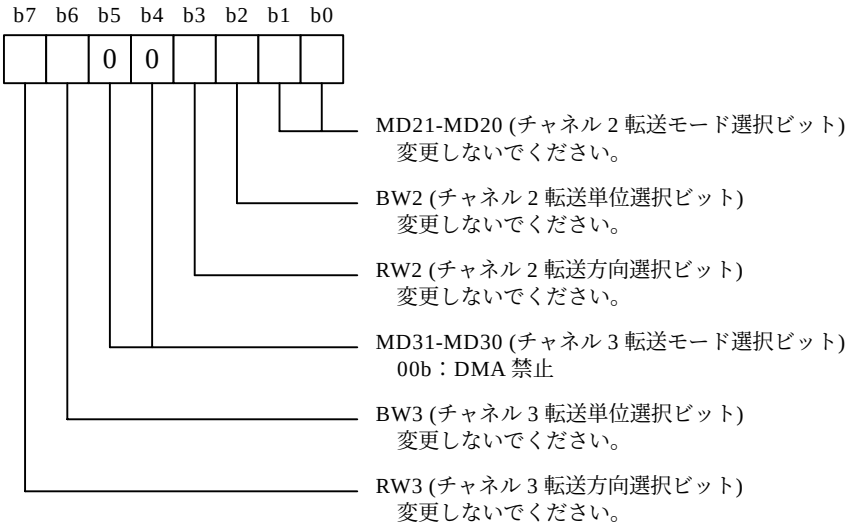
MD31-MD30 (チャネル 3 転送モード選択ビット)
変更しないでください。

BW3 (チャネル 3 転送単位選択ビット)
変更しないでください。

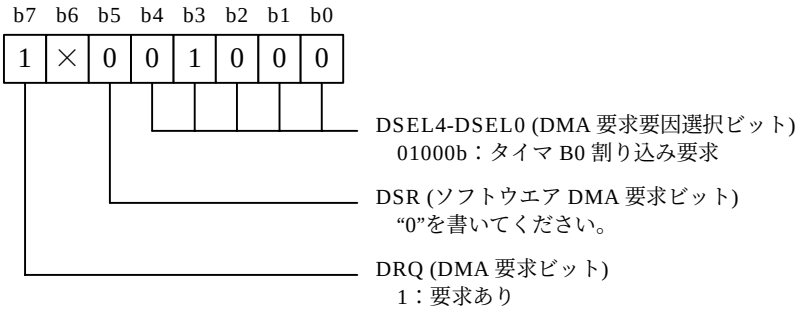
RW3 (チャネル 3 転送方向選択ビット)
変更しないでください。

(5) DMA3 設定

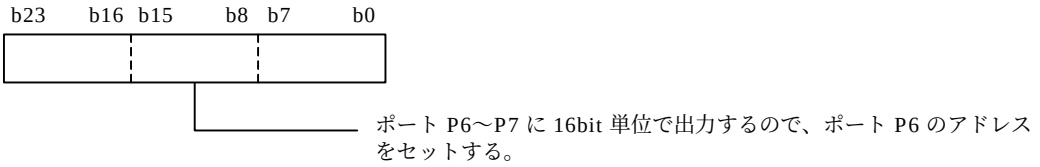
・ DMD1 レジスタ(DMA モードレジスタ 1)を設定(DMA3 を禁止)する。



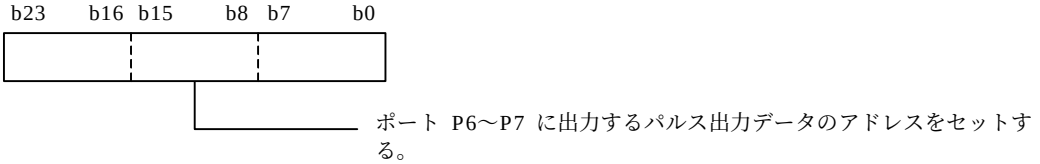
・ DM3SL レジスタ(DMA3 要因選択レジスタ)を設定する。



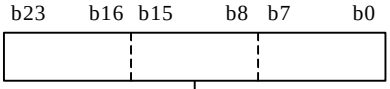
・ DSA3 レジスタ(DMA3 SFR アドレスレジスタ)を設定する。



・ DRA3 レジスタ(DMA3 メモリアドレスリロードレジスタ)を設定する。

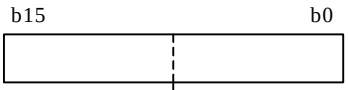


・ DMA3 レジスタ(DMA3 メモリアドレスレジスタ)を設定する。



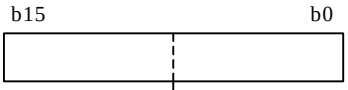
ポート P6～P7 に出力するパルス出力データのアドレスをセットする。

・ DRC3 レジスタ(DMA3 転送カウントリロードレジスタ)を設定する。



パルス出力データ数をセットする。

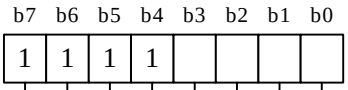
・ DCT3 レジスタ(DMA3 転送カウントレジスタ)を設定する。



パルス出力データ数をセットする。

・ ダミーサイクルを挿入する。
DM3SL レジスタを設定してから、BCLK で 6 サイクル分待ってから DMA を許可してください。ここでは、NOP を 6 個挿入することで、6 サイクル分待ちます。

・ DMD1 レジスタ(DMA モードレジスタ 1)を再設定(DMA3 を許可)する。



MD21-MD20 (チャネル 2 転送モード選択ビット)
変更しないでください。

BW2 (チャネル 2 転送単位選択ビット)
変更しないでください。

RW2 (チャネル 2 転送方向選択ビット)
変更しないでください。

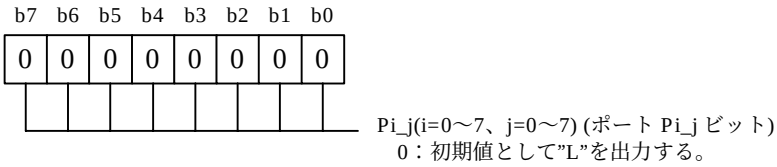
MD31-MD30 (チャネル 3 転送モード選択ビット)
11b：リピート転送

BW3 (チャネル 3 転送単位選択ビット)
1：16 ビット

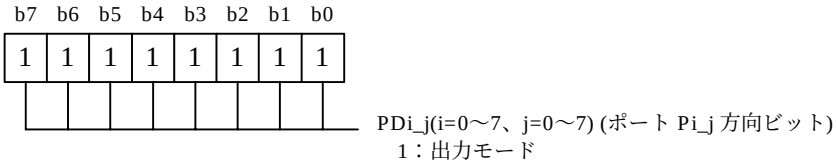
RW3 (チャネル 3 転送方向選択ビット)
1：メモリ(順方向)→固定番地

(6) PWM 出力用ポートを出力ポートに設定する。

・ P0～P7 レジスタ(ポート P0～P7 レジスタ)に初期値を設定する。



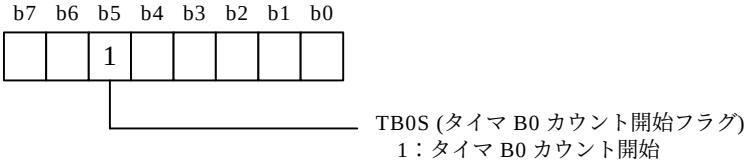
・ PD0～PD7 レジスタ(ポート PD0～PD7 方向レジスタ)で出力ポートに設定する。



(7) 割り込みを許可(I フラグ＝"1")する。

(8) タイマ B0 開始

・ TABSR レジスタ(カウント開始フラグ)を設定する。



(9) タイマ B0 割り込み処理

PWM パルス出力幅を変更するために、TB0 レジスタ(タイマ B0 レジスタ)の値を変更する。

4. 参考プログラム

タイマ B0 と DMA0～3 を使用して、100us～900us の周期で、64 本の PWM 出力を行うプログラム例を以下に示します。

動作条件： VCC1=VCC2=5.0V、XIN=8MHz、PLL=4 逓倍、f1=32MHz

```
/***********************************************************************/
/*                                                                     */
/*  M32C/85 Group Program Collection                                   */
/*                                                                     */
/*  FILE NAME : rjj05b0579_src.c                                       */
/*  CPU       : M32C/85 Group                                          */
/*  FUNCTION  : The sample program of the 64ch multiplex PWM output    */
/*              using DMAC.                                            */
/*  HISTORY   : 2004.09.15 Ver 1.00                                    */
/*                                                                     */
/*  Copyright (C) 2004. Renesas Technology Corp.                       */
/*  Copyright (C) 2004. Renesas Solutions Corp.                        */
/*  All right reserved.                                                */
/*                                                                     */
/***********************************************************************/

/*****************************************/
/*    include file                       */
/*****************************************/
#include "sfr32c8586.h"     // Special Function Register Header File

/*****************************************/
/*    Function declaration               */
/*****************************************/
void tb0_int(void);         // TB0 interrupt routine

/*****************************************/
/*    Global variable declaration        */
/*****************************************/
                            // PWM ch0-ch15 output data.
const unsigned short     pwm0_data[8] =
        {0x0101, 0x0202, 0x0404, 0x0808, 0x1010, 0x2020, 0x4040, 0x8080 };
const unsigned short     pwm1_data[8] =
        {0x0303, 0x0606, 0x0c0c, 0x1818, 0x3030, 0x6060, 0xc0c0, 0x8181 };
const unsigned short     pwm2_data[8] =
        {0x0707, 0x0e0e, 0x1c1c, 0x3838, 0x7070, 0xe0e0, 0xc1c1, 0x8383 };
const unsigned short     pwm3_data[8] =
        {0x0f0f, 0x1e1e, 0x3c3c, 0x7878, 0xf0f0, 0xe1e1, 0xc3c3, 0x8787 };

unsigned short           tb_value;      // TB0 current value

/*****************************************/
/*    #define declaration                */
/*****************************************/
#define TB_INI_VALUE    3200-1   // TB0 initial value
#define TB_MAX_VALUE    32000-1  // TB0 maximum value
#define TB_UP_VALUE     3200     // TB0 incremental value

/*****************************************/
/*    DMAC register declaration          */
/*****************************************/
```

```
                                // CPU internal registor
unsigned short  dmd0;
#pragma DMAC    dmd0    DMD0    // DMD0(DMA mode register0)
unsigned short  dct0;
#pragma DMAC    dct0    DCT0    // DCT0(DMA0 transfer count register)
unsigned short  drc0;
#pragma DMAC    drc0    DRC0    // DRC0(DMA0 transfer count reload register)
void _far       *dma0;
#pragma DMAC    dma0    DMA0    // DMA0(DMA0 memory address register)
void _far       *dsa0;
#pragma DMAC    dsa0    DSA0    // DSA0(DMA0 SFR address register)
void _far       *dra0;
#pragma DMAC    dra0    DRA0    // DRA0(DMA0 memory address reload register)

unsigned short  dmd1;
#pragma DMAC    dmd1    DMD1    // DMD1(DMA mode register1)
unsigned short  dct1;
#pragma DMAC    dct1    DCT1    // DCT1(DMA1 transfer count register)
unsigned short  drc1;
#pragma DMAC    drc1    DRC1    // DRC1(DMA1 transfer count reload register)
void _far       *dma1;
#pragma DMAC    dma1    DMA1    // DMA1(DMA1 memory address register)
void _far       *dsa1;
#pragma DMAC    dsa1    DSA1    // DSA1(DMA1 SFR address register)
void _far       *dra1;
#pragma DMAC    dra1    DRA1    // DRA1(DMA1 memory address reload register)

/*************************************/
/*    SFR declaration                */
/*************************************/
#pragma ADDRESS     plc0_w      0026H   // PLL control register 0 & 1
unsigned short      plc0_w;

/*************************************/
/*    Main Program                   */
/*************************************/
void main(void)
{
    short   dmd0_tmp;                   // DMD0 register temp
    short   dmd1_tmp;                   // DMD1 register temp
    short   pll_wait;                   // PLL wait counter

    prcr = 3;
    plc0_w = 0x0254;                    // PLL clock = Main clock x4.
    plc0   = 0xd4;                      // Start PLL.
                                        // It waits until a PLL clock is stabilized.
    for (pll_wait=0;pll_wait<4500;pll_wait++);
    cm17 = 1;                           // Main clock = PLL clock.
    mcd = 0x12;                         // Set main-clock no division mode.
    prcr = 0;

    // A setup a TB0(For DMA request cause).
    tb0mr = 0x00;                       // Set TB0MR register.
                                        //  <TMOD1-0> : timer mode
                                        //  <TCK1-0>  : f1

    tb_value = TB_INI_VALUE;
    tb0 = tb_value;                     // Set TB0 register.
                                        //  Pulse output cycle=1ms(XIN=32MHz)
```

```
tb0ic = 6;                      // Set TB0 interrupt priority level = 6.

// A setup of DMA0.
dmd0_tmp =  dmd0;               // DMA0 inhibit(DMD0)
dmd0_tmp &= 0x00fc;             //  <MD01-00> : DMA0 inhibit
dmd0     =  dmd0_tmp;           //

dm0sl = 0x88;                   // Set DM0SL register.
                                //  <DSEL4-0> : TB0
                                //  <DRQ>     : DMA requested

dsa0  = &p0;                    // Set DSA0 register.
dma0  = &pwm0_data[0];          // Set DMA0 register.
dra0  = &pwm0_data[0];          // Set DRA0 register.
dct0  = 8;                      // Set DCT0 register.
drc0  = 8;                      // Set DRC0 register.
                                // Dummy cycle insertion
asm("NOP    ");                 //  It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd0_tmp |= 0x0f;               // DMA0 permission(DMD0)
                                //  <MD01-00> : repeat transfer
                                //  <BW0>     : 16bit
                                //  <RW0>     : Memory to Fixed address
dmd0      = dmd0_tmp;

// A setup of DMA1.
dmd0_tmp =  dmd0;               // DMA1 inhibit(DMD0)
dmd0_tmp &= 0x00cf;             //  <MD11-10> : DMA1 inhibit
dmd0     =  dmd0_tmp;           //

dm1sl = 0x88;                   // Set DM1SL register.
                                //  <DSEL4-0> : TB0
                                //  <DRQ>     : DMA requested

dsa1  = &p2;                    // Set DSA1 register.
dma1  = &pwm1_data[0];          // Set DMA1 register.
dra1  = &pwm1_data[0];          // Set DRA1 register.
dct1  = 8;                      // Set DCT1 register.
drc1  = 8;                      // Set DRC1 register.
                                // Dummy cycle insertion
asm("NOP    ");                 //   It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd0_tmp |= 0xf0;               // DMA1 permission(DMD0)
                                //  <MD11-10> : repeat transfer
                                //  <BW1>     : 16bit
                                //  <RW1>     : Memory to Fixed address
dmd0      = dmd0_tmp;

// (3) A setup of DMA2.
dmd1_tmp =  dmd1;               // DMA2 inhibit(DMD1)
```

```
dmd1_tmp &= 0x00fc;                 //  <MD21-20> : DMA2 inhibit
dmd1     =  dmd1_tmp;               //


asm("mov.b  #088h, _dm2sl_addr");   // Set DM2SL register.
                                    //  <DSEL4-0> : TB0
                                    //  <DRQ>     : DMA requested
asm("fclr   I");                    // Interrupt disabled.
asm("fset   B");                    // Register-bank1 enable
asm("ldc    #_p4_addr, sb");        // Set DSA2(SB) regisster.
asm("ldc    #_pwm2_data, svp");     // Set DRA2(SVP) register.
asm("mov.l  #_pwm2_data, a0");      // Set DMA2(A0) register.
asm("mov.w  #8, r2");               // Set DRC2(R2) register.
asm("mov.w  #8, r0");               // Set DCT2(R0) register.
asm("fclr   B");                    // Register-bank1 disable
                                    // Dummy cycle insertion
asm("NOP    ");                     //   It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd1_tmp |= 0x0f;                   // DMA2 permission(DMD1)
                                    //  <MD21-20> : repeat transfer
                                    //  <BW0>     : 16bit
                                    //  <RW0>     : Memory to Fixed address
dmd1      = dmd1_tmp;


// (4) A setup of DMA3.
dmd1_tmp =  dmd1;                   // DMA3 inhibit(DMD1)
dmd1_tmp &= 0x00cf;                 //  <MD31-30> : DMA3 inhibit
dmd1     =  dmd1_tmp;               //


asm("mov.b  #088h, _dm3sl_addr");   // A setup of DM3SL(DRQ=1, TA1)
                                    //  <DSEL4-0> : TB0
                                    //  <DRQ>     : DMA requested
asm("fclr   I");                    // Interrupt disabled.
asm("fset   B");                    // Register-bank1 enable
asm("ldc    #_p6_addr, fb");        // Set DSA3(FB) register.
asm("ldc    #_pwm3_data, vct");     // Set DRA3(VCT) register.
asm("mov.l  #_pwm3_data, a1");      // Set DMA3(A1) register.
asm("mov.w  #8, r3");               // Set DRC3(R3) register.
asm("mov.w  #8, r1");               // Set DCT3(R1) register.
asm("fclr   B");                    // Register-bank1 disable
                                    // Dummy cycle insertion
asm("NOP    ");                     //   It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd1_tmp |= 0xf0;                   // DMA3 permission(DMD1)
                                    //  <MD11-10> : repeat transfer
                                    //  <BW1>     : 16bit
                                    //  <RW1>     : Memory to Fixed address
dmd1      = dmd1_tmp;
```

```
    // A setup a Port(For PWM output).
    p0  = 0;
    p1  = 0;
    p2  = 0;
    p3  = 0;
    p4  = 0;
    p5  = 0;
    p6  = 0;
    p7  = 0;
    pd0 = 0xff;                         // P0 is output port.
    pd1 = 0xff;                         // P1 is output port.
    pd2 = 0xff;                         // P2 is output port.
    pd3 = 0xff;                         // P3 is output port.
    pd4 = 0xff;                         // P4 is output port.
    pd5 = 0xff;                         // P5 is output port.
    pd6 = 0xff;                         // P6 is output port.
    pd7 = 0xff;                         // P7 is output port.

    asm("fset i");                      // Interrupt enabled.

    tb0s = 1;                           // TB0 start.

    while(1);

}

/****************************************/
/*  TB0 interrupt routine               */
/****************************************/
#pragma INTERRUPT/B tb0_int
// "/B" = Instead of saving the registers to the stack,
//        you can switch to the alternate registers.

void tb0_int(void)
{
                                        // A next timer value is calculated.
    tb_value += TB_UP_VALUE;
    if (tb_value >= TB_MAX_VALUE) tb_value = TB_INI_VALUE;

    tb0 = tb_value;                     // Next timer value set.

}
```

5. 参考ドキュメント

ハードウエアマニュアル

M32C/85 グループハードウエアマニュアル
（最新版をルネサス テクノロジホームページから入手してください。）

6. ホームページとサポート窓口

ルネサス テクノロジホームページ

http://www.renesas.com/jpn/

M16C ファミリ MCU 技術サポート窓口

E-mail:support_apl@renesas.com

改訂記録

Rev.	発行日		改訂内容
		ページ	ポイント
1.00	2004.09.15	-	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が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は負いません。
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。