

21. タイマRDによるPWM波形出力(リセット同期PWMモード)(プロジェクト:timer_rd_doukipwm)

21.1 概要

本章では、PWM 波形を出力する方法を紹介します。今回は、タイマ RD をリセット同期 PWM モードで使用して、同じ周期の PWM 波形を 3 本、その波形を反転した波形を 3 本、1 周期ごとに反転する波形を 1 本、合計 7 本の波形を出力します。タイマ RD の初期設定後は、プログラムが関与しなくても PWM 波形を出力し続けます。プログラムでは、PWM 波形出力処理以外の処理をすることができます。

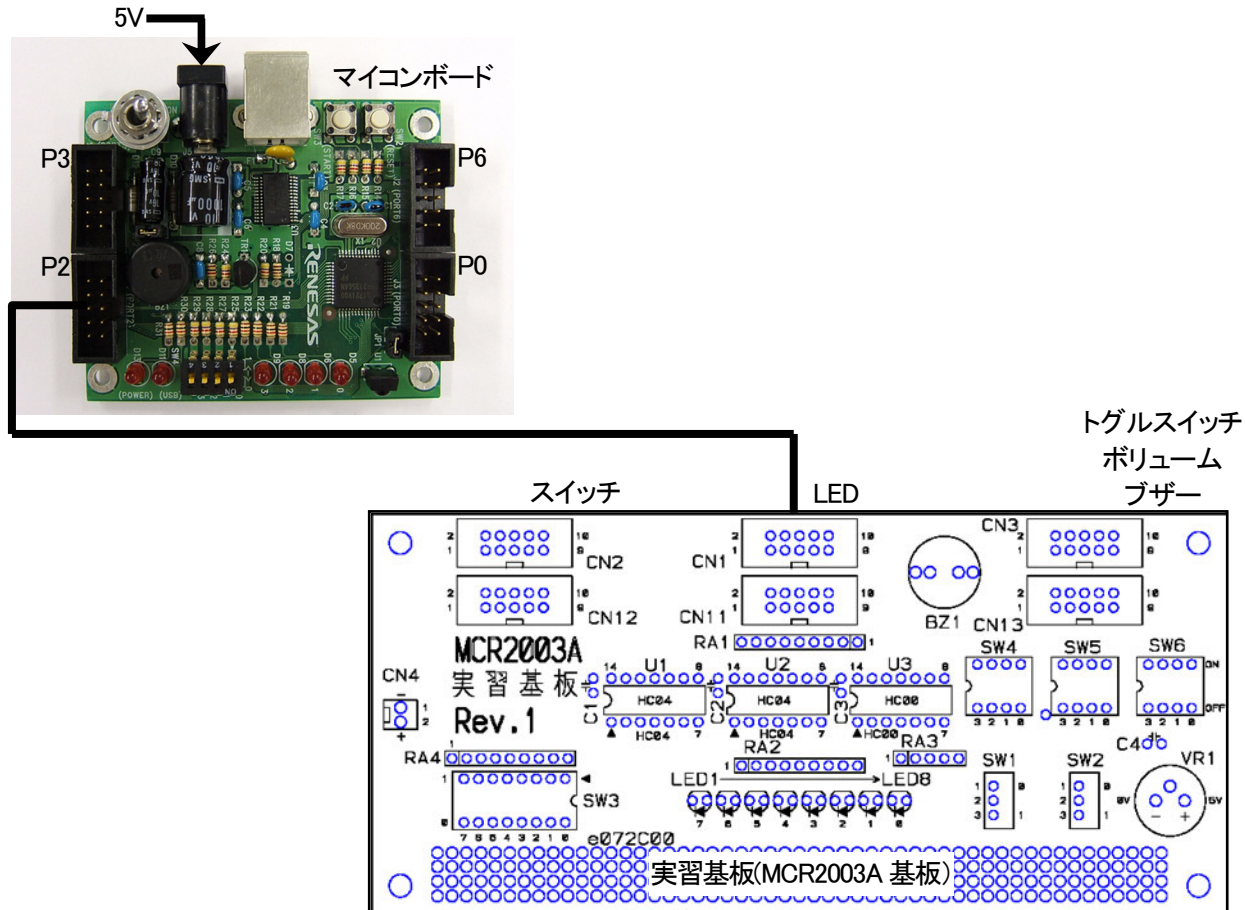
21.2 接続

■使用ポート

マイコンのポート	接続内容
P5_7、P4_5、P4_4、P4_3	マイコンボード上のディップスイッチです。
P2 (J7)	実習基板の LED 部など、出力機器を接続します。

■接続例

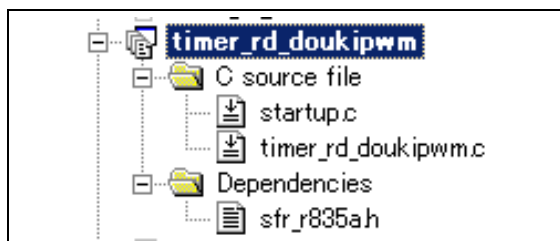
実習基板を使ったときの接続例を次に示します。



■操作方法

マイコンボードのディップスイッチ(SW4)の値 0～15 によって、LED の点灯する明るさが変わります。このとき、どの LED が明るくなって、どの LED が暗くなるか観察してください。

21.3 プロジェクトの構成



	ファイル名	内容
1	startup.c	固定割り込みベクタアドレスの設定、スタートアッププログラム、RAM の初期化(初期値のないグローバル変数、初期値のあるグローバル変数の設定)などを行います。このファイルは共通で、どのプロジェクトもこのファイルから実行されます。
2	timer_rd_doukipwm.c	実際に制御するプログラムが書かれています。R8C/35A の内蔵周辺機能(SFR)の初期化も行います。
3	sfr_r835a.h	R8C/35A マイコンの内蔵周辺機能を制御するためのレジスタ (Special Function Registers)を定義したファイルです。

21.4 プログラム「timer_rd_doukipwm.c」

```

1 :  /******
2 :  /* 対象マイコン R8C/35A
3 :  /* ファイル内容 タイマRDによるリセット同期PWM
4 :  /* バージョン Ver. 1. 20
5 :  /* Date 2010. 04. 19
6 :  /* Copyright ルネサスマイコンカーラリー事務局
7 :  /* 日立インターメディックス株式会社
8 :  /******
9 :  /*
10 :  入力：マイコンボードのディップスイッチ
11 :  出力：P2_1端子～P2_7端子からPWM出力
12 :
13 :  ディップスイッチでデューティ比(ONの割合)を設定し、LEDに出力します。
14 :  */
15 :
16 :  /*=====*/
17 :  /* インクルード */
18 :  /*=====*/
19 :  #include "sfr_r835a.h" /* R8C/35A SFRの定義ファイル */
20 :
21 :  /*=====*/
22 :  /* シンボル定義 */
23 :  /*=====*/
24 :
25 :  /*=====*/
26 :  /* プロトタイプ宣言 */
27 :  /*=====*/
28 :  void init( void );
29 :  unsigned char dipsw_get( void );
30 :

```

```

31 : /*****
32 : /* メインプログラム */
33 : *****/
34 : void main( void )
35 : {
36 :     init(); /* 初期化 */
37 :
38 :     while( 1 ) {
39 :         trdgrd0 = 39998 * dipsw_get() / 15;
40 :     }
41 : }
42 :
43 : /*****
44 : /* R8C/35A スペシャルファンクションレジスタ(SFR)の初期化 */
45 : *****/
46 : void init( void )
47 : {
48 :     int i;
49 :
50 :     /* クロックをXINクロック(20MHz)に変更 */
51 :     prc0 = 1; /* プロテクト解除 */
52 :     cm13 = 1; /* P4_6, P4_7をXIN-XOUT端子にする */
53 :     cm05 = 0; /* XINクロック発振 */
54 :     for(i=0; i<50; i++ ); /* 安定するまで少し待つ(約10ms) */
55 :     ocd2 = 0; /* システムクロックをXINにする */
56 :     prc0 = 0; /* プロテクトON */
57 :
58 :     /* ポートの入出力設定 */
59 :     prc2 = 1; /* PD0のプロテクト解除 */
60 :     pd0 = 0xe0; /* 7-5:LED 4:MicroSW 3-0:Sensor */
61 :     pl = 0x0f; /* 3-0:LEDは消灯 */
62 :     pd1 = 0xdf; /* 5:RXD0 4:TXD0 3-0:LED */
63 :     pd2 = 0xfe; /* 0:PushSW */
64 :     pd3 = 0xff; /* 4:Buzzer 2:IR */
65 :     pd4 = 0x83; /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF */
66 :     pd5 = 0x40; /* 7:DIP SW */
67 :     pd6 = 0xff;
68 :
69 :     /* タイマRD リセット同期PWMモードの設定 */
70 :     trdfer = 0x01; /* リセット同期PWMモードに設定 */
71 :     trdmr = 0xf0; /* バッファレジスタ設定 */
72 :     trdoer1 = 0x01; /* 出力端子の選択 */
73 :     trdpsr0 = 0x68; /* TRDIOB0, C0, D0端子設定 */
74 :     trdpsr1 = 0x55; /* TRDIOA1, B1, C1, D1端子設定 */
75 :     trdcr0 = 0x23; /* ソースカウントの選択:f8 */
76 :     trdgra0 = trdgrc0 = 39999; /* 周期 */
77 :     trdgrb0 = trdgrd0 = 0; /* P2_2端子のON幅設定 */
78 :     trdgra1 = trdgrc1 = 0; /* P2_4端子のON幅設定 */
79 :     trdgrb1 = trdgrd1 = 0; /* P2_5端子のON幅設定 */
80 :     trdstr = 0x0d; /* TRD0カウンタ開始 */
81 : }
82 :
83 : /*****
84 : /* ディップスイッチ値読み込み */
85 : /* 戻り値 スイッチ値 0~15 */
86 : *****/
87 : unsigned char dipsw_get( void )
88 : {
89 :     unsigned char sw, sw1, sw2;
90 :
91 :     sw1 = (p5>>4) & 0x08; /* ディップスイッチ読み込み3 */
92 :     sw2 = (p4>>3) & 0x07; /* ディップスイッチ読み込み2,1,0 */
93 :     sw = sw1 | sw2; /* P5とP4の値を合わせる */
94 :
95 :     return sw;
96 : }
97 :
98 : /*****
99 : /* end of file */
100 : *****/

```

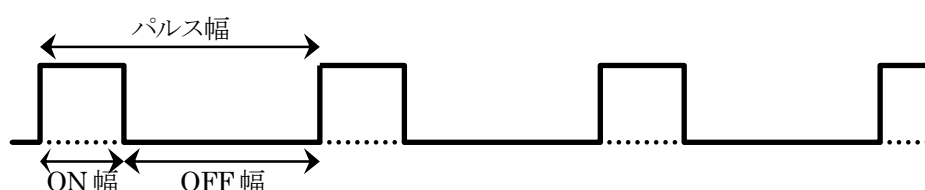
21.5 PWMとは？

モータのスピード制御を考えてみます。

モータを回したければ、電圧を加えます。止めたければ、電圧を加えなければいけません。では、その中間のスピードや10%、20%…など、細かくスピード調整したいときはどうすればよいのでしょうか。

ボリューム(半固定抵抗)を使えば電圧を可変することができます。しかし、モータへは大電流が流れるため、許容電流の大きなボリュームが必要です。また、抵抗で分圧した分は、抵抗の熱となってしまいます。

そこで、スイッチをON/OFFすることを高速に繰り返して、あたかも中間的な電圧が出ているような制御を行います。ON/OFF 信号は、周期を一定にして ON と OFF の比率を変える制御を行います。これを、「パルス幅変調」と呼び、英語では「Pulse Width Modulation」といいます。略して**PWM制御**といえます。パルス幅に対するONの割合のことを**デューティ比**といえます。周期に対する ON 幅を 50%にすると、デューティ比 50%といえます。他にも PWM50%とか、単純にモータ 50%といえます。



デューティ比は下記で表すことができます。

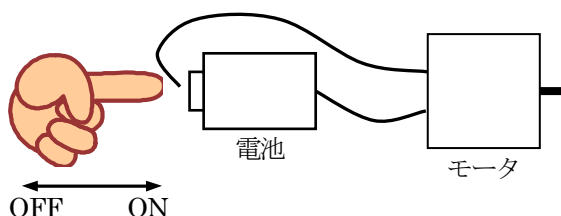
$$\text{デューティ比} = \text{ON 幅} / \text{パルス幅 (ON 幅 + OFF 幅)}$$

例えば、100ms のパルスに対して、ON 幅が 60ms なら、

$$\text{デューティ比} = 60\text{ms} / 100\text{ms} = 0.6 = 60\%$$

となります。すべて ON なら、100%、すべて OFF なら 0%となります。

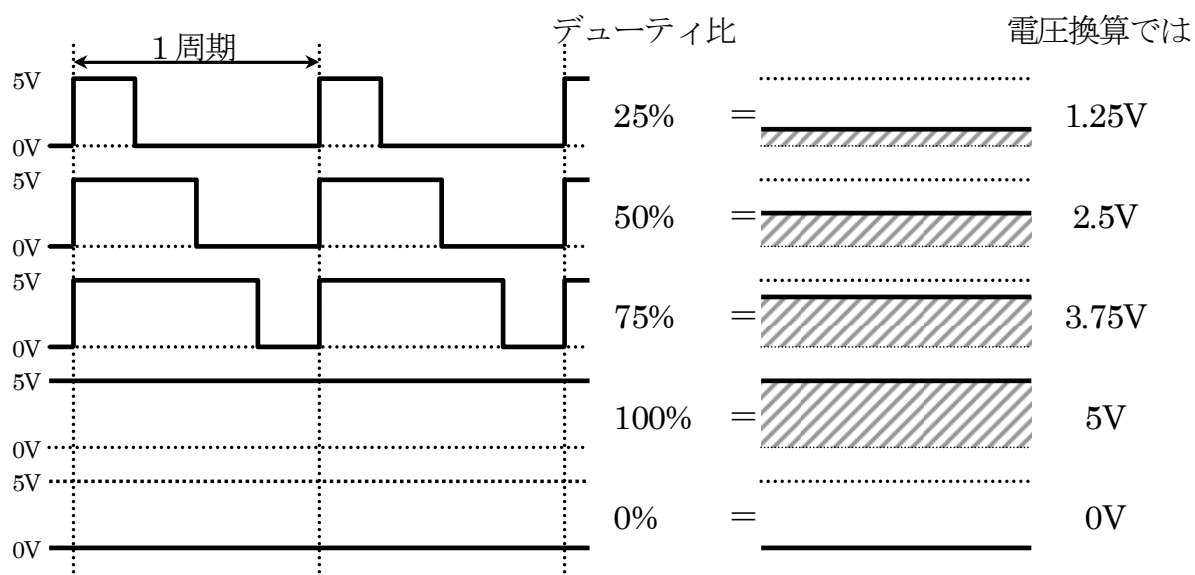
「PWM」と聞くと、何か難しく感じてしまいがちですが、下記のように手でモータと電池の線を「繋ぐ」、「離す」の繰り返し、それも PWM と言えます。繋いでいる時間が長いとモータは速く回ります。離している時間が長いとモータは少ししか回りません。人なら「繋ぐ」、「離す」の動作をコンマ数秒でしか行えませんが、マイコンならマイクロ数、またはミリ秒単位で行うことができます。



下図のように、0Vと5Vを出力するような波形で考えてみます。1周期に対してONの時間が長ければ長いほど平均化した値は大きくなります。すべて5Vにすればもちろん平均化しても5V、これが最大の電圧です。ONの時間を半分の50%にするとどうでしょうか。平均化すると $5V \times 0.5 = 2.5V$ と、あたかも電圧が変わったようになります。

このようにONにする時間を1周期の90%,80%...0%にすると徐々に平均した電圧が下がっていき最後には0Vになります。

この信号をモータに接続すれば、モータの回転スピードも少しずつ変化させることができ、微妙なスピード制御が可能です。LEDに接続すれば、LEDの明るさを変えることができます。マイコンを使えばこの作業をマイクロ秒、ミリ秒単位で行うことができます。このオーダでの制御になると、非常にスムーズなモータ制御が可能です。



なぜ電圧制御ではなく、パルス幅制御でモータのスピードを制御するのでしょうか。マイコンは”0”か”1”かのデジタル値の取り扱いは大変得意ですが、何Vというアナログ的な値は不得意です。そのため、”0”と”1”の幅を変えて、あたかも電圧制御しているように振る舞います。これがPWM制御です。

21.6 プログラムの解説

21.6.1 init関数(タイマRDの設定)

タイマ RD を使い、リセット同期 PWM モードの設定を行います。

```

69 :      /* タイマRD リセット同期PWMモードの設定*/
70 :      trdfcr = 0x01;          /* リセット同期PWMモードに設定 */
71 :      trdmr  = 0xf0;          /* バッファレジスタ設定 */
72 :      trdoerl = 0x01;         /* 出力端子の選択 */
73 :      trdpsr0 = 0x68;         /* TRDIOB0, C0, D0端子設定 */
74 :      trdpsr1 = 0x55;         /* TRDIOA1, B1, C1, D1端子設定 */
75 :      trdcr0  = 0x23;         /* ソースカウントの選択:f8 */
76 :      trdgra0 = trdgrc0 = 39999; /* 周期 */
77 :      trdgrb0 = trdgrd0 = 0;    /* P2_2端子のON幅設定 */
78 :      trdgral = trdgrcl = 0;    /* P2_4端子のON幅設定 */
79 :      trdgrbl = trdgrdl = 0;    /* P2_5端子のON幅設定 */
80 :      trdstr  = 0x0d;          /* TRD0カウント開始 */

```

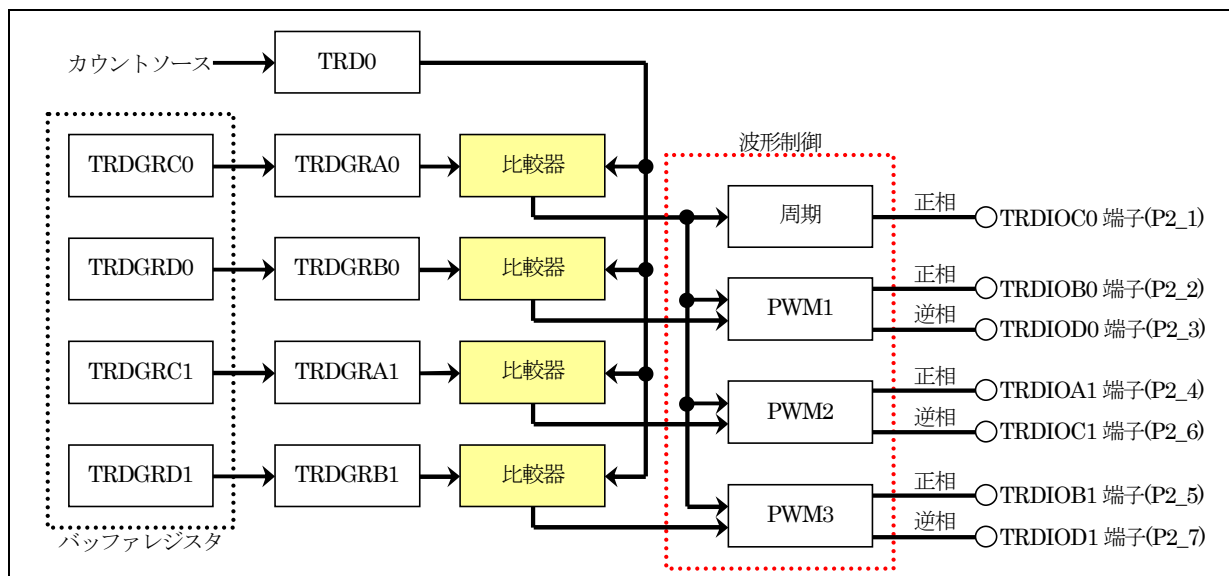
(1) タイマRDとは

R8C/35A には、タイマ RD というタイマが 2 個内蔵されています。タイマ RD には、次の 5 種類のモードがあります。今回は、「**リセット同期 PWM モード**」を使います。

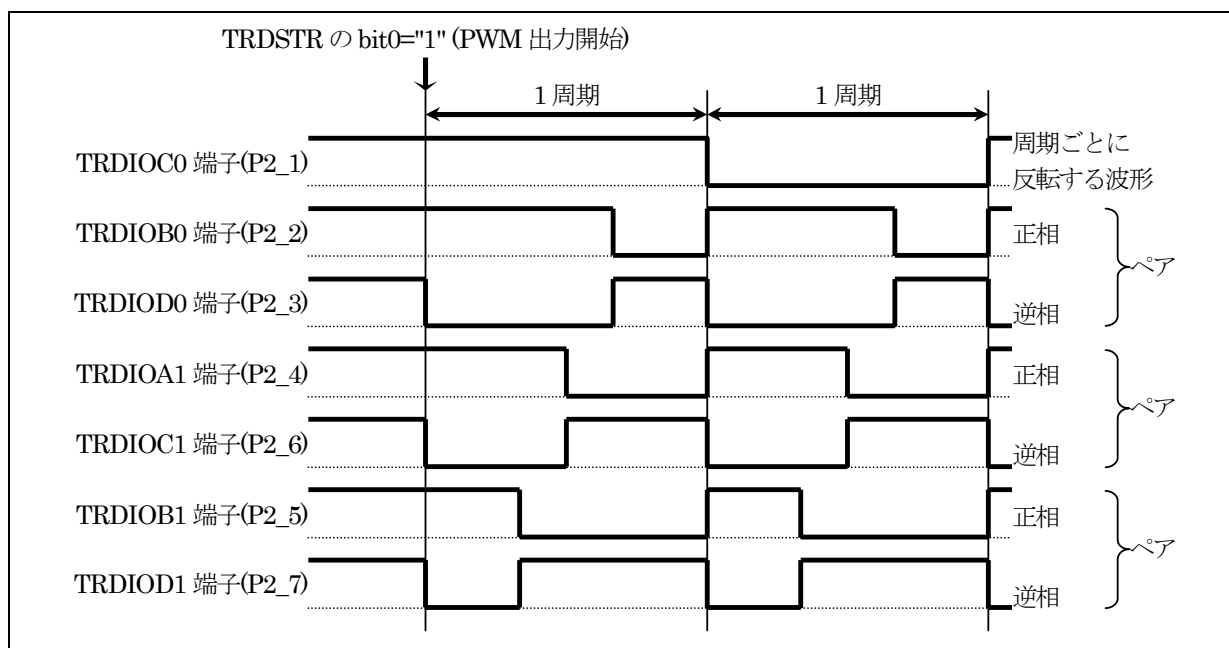
モード	詳細
タイマモード	タイマモードには、次の 2 つの機能があります。 ・インプットキャプチャ機能 外部信号をトリガにしてカウンタの値をレジスタに取り込む機能 ・アウトプットコンペア機能 カウンタとレジスタの値の一致を検出する機能(検出時に端子出力変更可能)
PWM モード	任意の幅のパルスを連続して出力するモード
リセット同期 PWM モード	鋸波変調、短絡防止時間なしの三相波形(6 本)を出力するモード
相補 PWM モード	三角波変調、短絡防止時間ありの三相波形(6 本)を出力するモード
PWM3 モード	同一周期の PWM 波形(2 本)を出力するモード

(2) タイマRDのブロック図

リセット同期 PWM モードのブロック図を下記に示します。リセット同期 PWM モードは、タイマ RD のチャンネル 0 とチャンネル 1 を組み合わせて使います。

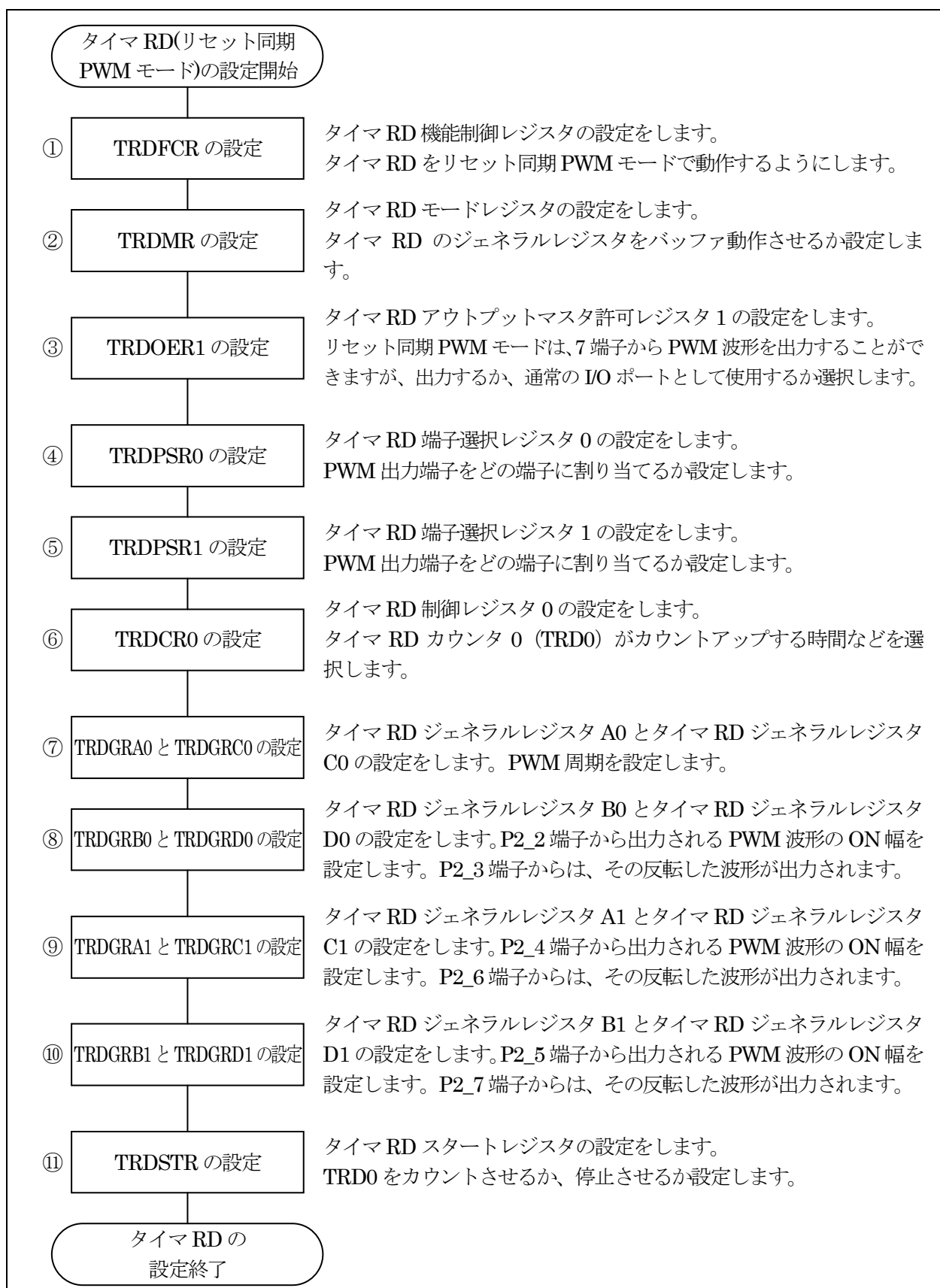


PWM 波形の出力例を下記に示します。



(3) タイマRDの設定(リセット同期PWMモード)

今回は、タイマ RD をリセット同期式 PWM モードで使用して、PWM 波形を正相 3 本、逆相 3 本、周期ごとに反転する波形の合計 7 本出力します。レジスタの設定手順を下記に示します。



①タイマ RD 機能制御レジスタ (TRDFCR:Timer RD function control register)の設定

タイマ RD の機能を設定します。

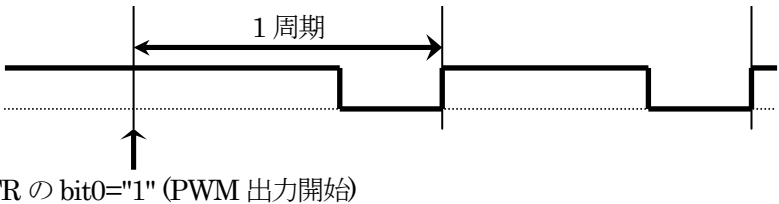
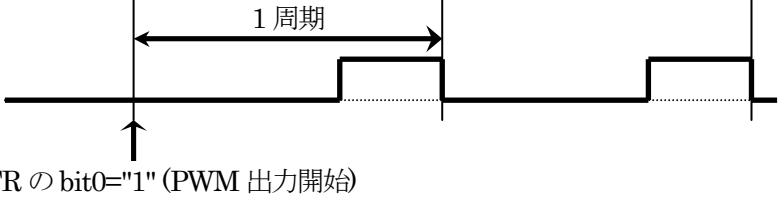
設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7		"0"を設定	0
bit6	外部クロック入力選択ビット stclk_trdfer	0:外部クロック入力無効 1:外部クロック入力有効 今回は、内部のクロックを使用しますので、無効を選択します。	0
bit5,4		"00"を設定	00
bit3	逆相出力レベル選択ビット (リセット同期 PWM モードまたは相補 PWM モード時) ols1_trdfer	0:初期出力"H"、アクティブレベル"L" 1:初期出力"L"、アクティブレベル"H" 今回は、"0"を設定します。	0
bit2	正相出力レベル選択ビット (リセット同期 PWM モードまたは相補 PWM モード時) ols0_trdfer	0:初期出力"H"、アクティブレベル"L" 1:初期出力"L"、アクティブレベル"H" 今回は、"0"を設定します。	0
bit1,0	コンビネーションモード選択ビット bit1:cmd1_trdfer bit0:cmd0_trdfer	リセット同期 PWM モードでは"01"(リセット同期 PWM モード)にしてください	01

タイマ RD 機能制御レジスタ (TRDFCR)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	0	0	0	1
16 進数	0				1			

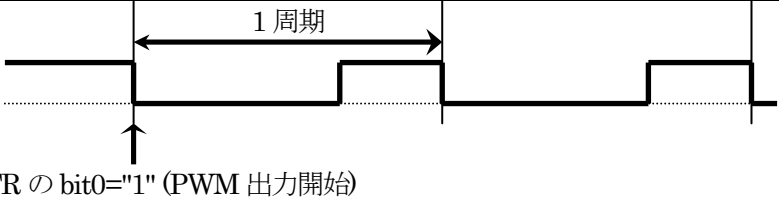
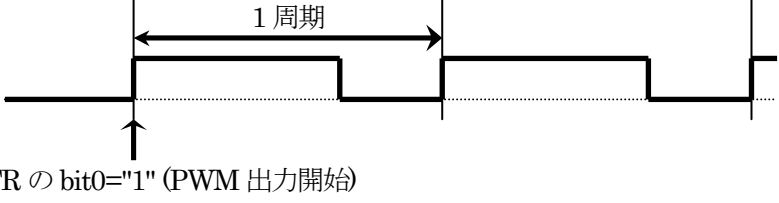
※正相出力レベル選択ビットと出力波形の関係

正相出力とは、P2_2 端子、P2_4 端子、P2_5 端子から出力される波形のことです。この端子から出力される波形のレベルをどうするか設定します。

bit2	波形	説明
0 (標準)	 <p>出力波形</p> <p>TRDSTR の bit0="1" (PWM 出力開始)</p>	"1" からスタートする設定です。
1	 <p>出力波形</p> <p>TRDSTR の bit0="1" (PWM 出力開始)</p>	"0" からスタートする設定です。

※逆相出力レベル選択ビットと出力波形の関係

逆相出力とは、P2_3 端子、P2_6 端子、P2_7 端子から出力される波形のことです。この端子から出力される波形のレベルをどうするか設定します。

bit3	波形	説明
0 (標準)	 <p>出力波形</p> <p>TRDSTR の bit0="1" (PWM 出力開始)</p>	"0" からスタートする設定です。
1	 <p>出力波形</p> <p>TRDSTR の bit0="1" (PWM 出力開始)</p>	"1" からスタートする設定です。

②タイマ RD モードレジスタ(TRDMR:Timer RD mode register)の設定

タイマ RD のジェネラルレジスタをバッファレジスタとして使用するかどうか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	TRDGRD1 レジスタ機能選択 ビット bfd1_trdmr	0:ジェネラルレジスタ 1:TRDGRB1 レジスタのバッファレジスタ バッファレジスタとして使用します。1 を設定します。	1
bit6	TRDGRC1 レジスタ機能選択 ビット bfc1_trdmr	0:ジェネラルレジスタ 1:TRDGRA1 レジスタのバッファレジスタ バッファレジスタとして使用します。1 を設定します。	1
bit5	TRDGRD0 レジスタ機能選択 ビット bfd0_trdmr	0:ジェネラルレジスタ 1:TRDGRB0 レジスタのバッファレジスタ バッファレジスタとして使用します。1 を設定します。	1
bit4	TRDGRC0 レジスタ機能選択 ビット bfc0_trdmr	0:ジェネラルレジスタ 1:TRDGRA0 レジスタのバッファレジスタ バッファレジスタとして使用します。1 を設定します。	1
bit3～0		"0000"を設定	0000

バッファレジスタについては、後述します。

タイマ RD モードレジスタ(TRDMR)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	1	1	1	1	0	0	0	0
16 進数	F				0			

③タイマ RD アウトプットマスタ許可レジスタ 1(TRDOER1:Timer RD output master enable register 1)の設定

リセット同期 PWM モードは 7 端子から PWM 波形を出力することができますが、出力するか、通常の I/O ポートとして使用するかを選択できます。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	TRDIOD1(P2_7)出力禁止ビット ed1_trdoer1	0:出力許可 1:出力禁止(TRDIOD1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit6	TRDIOC1(P2_6)出力禁止ビット ec1_trdoer1	0:出力許可 1:出力禁止(TRDIOC1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit5	TRDIOB1(P2_5)出力禁止ビット eb1_trdoer1	0:出力許可 1:出力禁止(TRDIOB1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit4	TRDIOA1(P2_4)出力禁止ビット ea1_trdoer1	0:出力許可 1:出力禁止(TRDIOA1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit3	TRDIOD0(P2_3)出力禁止ビット ed0_trdoer1	0:出力許可 1:出力禁止(TRDIOD0 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit2	TRDIOC0(P2_1)出力禁止ビット ec0_trdoer1	0:出力許可 1:出力禁止(TRDIOC0 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit1	TRDIOB0(P2_2)出力禁止ビット eb0_trdoer1	0:出力許可 1:出力禁止(TRDIOB0 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit0		"1"を設定	1

タイマ RD アウトプットマスタ許可レジスタ 1(TRDOER1)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	0	0	0	1
16 進数	0				1			

④タイマ RD 端子選択レジスタ 0(TRDPSR0:Timer RD function select register 0)の設定

PWM 波形の出力端子をどのポートに割り当てるか設定します。タイマ RD 端子選択レジスタ 0(TRDPSR0)では、TRDIOD0 端子、TRDIOC0 端子、TRDIOB0 端子の割り当てを設定します。

R8C/35A では、割り当てる端子は決まっております端子を変更することができません。PWM 端子として割り当てるか、割り当てないか(通常の I/O ポートとして使用するか)を設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7		"0"を設定	0
bit6	TRDIOD0 端子選択ビット trdiiod0sel0	0:TRDIOD0 端子は使用しない 1:P2_3 に割り当てる 今回は、P2_3 に割り当てて、この端子から PWM 波形を出力します。	1
bit5,4	TRDIOC0 端子選択ビット bit5:trdioc0sel1 bit4:trdioc0sel0	00:TRDIOC0 端子は使用しない 01:設定しないでください 10:P2_1 に割り当てる 11:設定しないでください 今回は、P2_1 に割り当てて、この端子から PWM 波形(周期の波形)を出力します。	10
bit3,2	TRDIOB0 端子選択ビット bit3:trdiob0sel1 bit2:trdiob0sel0	00:TRDIOB0 端子は使用しない 01:設定しないでください 10:P2_2 に割り当てる 11:設定しないでください 今回は、P2_2 に割り当てて、この端子から PWM 波形を出力します。	10
bit1		"0"を設定	0
bit0	TRDIOA0/TRDCLK 端子選択ビット trdioa0sel0	0:TRDIOA0/TRDCLK 端子は使用しない 1:P2_0 に割り当てる リセット同期 PWM モードを使用するときは"0"を設定します。	0

タイマ RD 端子選択レジスタ 0(TRDPSR0)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	1	1	0	1	0	0	0
16 進数	6				8			

⑤タイマ RD 端子選択レジスタ 1(TRDPSR1:Timer RD function select register 1)の設定

PWM 波形の出力端子をどのポートに割り当てるか設定します。タイマ RD 端子選択レジスタ 1(TRDPSR1)では、TRDIOD1 端子、TRDIOC1 端子、TRDIOB1 端子、TRDIOA1 端子の割り当てを設定します。

R8C/35A では、割り当てる端子は決まっており端子を変更することができません。PWM 端子として割り当てるか、割り当てないか(通常の I/O ポートとして使用するか)を設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7		"0"を設定	0
bit6	TRDIOD1 端子選択ビット trdioid1sel0	0:TRDIOD1 端子は使用しない 1:P2_7 に割り当てる 今回は、P2_7に割り当てて、この端子からPWM 波形を出力します。	1
bit5		"0"を設定	0
bit4	TRDIOC1 端子選択ビット trdioc1sel0	0:TRDIOC1 端子は使用しない 1:P2_6 に割り当てる 今回は、P2_6に割り当てて、この端子からPWM 波形を出力します。	1
bit3		"0"を設定	0
bit2	TRDIOB1 端子選択ビット trdiob1sel0	0:TRDIOB1 端子は使用しない 1:P2_5 に割り当てる 今回は、P2_5に割り当てて、この端子からPWM 波形を出力します。	1
bit1		"0"を設定	0
bit0	TRDIOA1 端子選択ビット trdioa1sel0	0:TRDIOA1 端子は使用しない 1:P2_4 に割り当てる 今回は、P2_4に割り当てて、この端子からPWM 波形を出力します。	1

タイマ RD 端子選択レジスタ 1(TRDPSR1)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	1	0	1	0	1	0	1
16 進数	5				5			

⑥タイマ RD 制御レジスタ 0(TRDCR0:Timer RD control register 0)の設定

タイマ RD カウンタ 0 (TRD0) がカウントアップする時間などを選択します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～5	TRD0 カウンタクリア選択ビット bit7:cclr2_trdcr0 bit6:cclr1_trdcr0 bit5:cclr0_trdcr0	リセット同期 PWM モードの場合は、“001”(TRDGRA0 とのコンペアー一致で TRD0 レジスタクリア)に設定	001
bit4,3	外部クロックエッジ選択ビット (注 3) bit4:ckeg1_trdcr0 bit3:ckeg0_trdcr0	00:立ち上がりエッジでカウント 01:立ち下がりエッジでカウント 10:両エッジでカウント 11:設定しないでください 外部クロックは使いませんので何を設定しても変化ありません。今回は“00”を設定します。	00
bit2～0	カウントソース選択ビット bit2:tck2_trdcr0 bit1:tck1_trdcr0 bit0:tck0_trdcr0	000:f1 (1/20MHz=50ns) 001:f2 (2/20MHz=100ns) 010:f4 (4/20MHz=200ns) 011:f8 (8/20MHz=400ns) 100:f32 (32/20MHz=1600ns) 101:TRDCLK 入力(注 1)または fC2 (注 2) fC2 = 2/XCIN クロック=今回は未接続 110:fOCO40M (高速オンチップオシレータ 40MHz= 今回は未接続) 111:fOCO-F(注 4) (高速オンチップオシレータを FRA2 で分周したクロック=今回は未接続) タイマ RD カウンタ 0 (TRD0) がカウントアップする時間を 設定します。今回は“011”を設定します。TRD0 は、 400ns ごとに+1 していきます。	011

注 1. TRDECR レジスタの ITCLK0 ビットが“0”(TRDCLK 入力)かつ TRDFCR レジスタの STCLK ビットが“1”(外部クロック入力有効)のとき、有効です。

注 2. タイマモードで、TRDECR レジスタの ITCLK0 ビットが“1”(fC2)のとき有効です。

注 3. TCK2～TCK0 ビットが“101”(TRDCLK 入力または fC2)、TRDECR レジスタの ITCLK0 ビットが“0”(TRDCLK 入力)、かつ TRDFCR レジスタの STCLK ビットが“1”(外部クロック入力有効)のとき、有効です。

注 4. fOCO-F を選択するとき、CPU クロックより速いクロック周波数に fOCO-F を設定してください。

タイマ RD 制御レジスタ 0(TRDCR0)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	1	0	0	0	1	1
16 進数	2				3			

※タイマ RD 制御レジスタ 0(TRDCR0)カウンタソース選択ビットの設定方法

タイマ RD 制御レジスタ 0(TRDCR0)のカウンタソース選択ビット(bit2～0)で、タイマ RD カウンタ 0(TRD0)がどのくらいの間隔で+1 するか設定します。TRD0 は、0 からスタートして最大 65,535 までカウントアップします。65,535 の次は 0 に戻ります。PWM の周期や ON 幅は TRD0 の値を基準にするので、カウントアップする時間×65,536 以上の時間を設定することができません。

タイマ RD 制御レジスタ 0(TRDCR0)のカウンタソース選択ビットの値と、周期の関係を下記に示します。

bit2～0	内容
000	タイマ RD カウンタ 0(TRD0)がカウントアップする時間を、f1 に設定します。時間は、 $f1/20\text{MHz}=1/20\text{MHz}=50\text{ns}$ 設定できる PWM 周期の最大は、 $50\text{ns} \times 65,536 = \mathbf{3.2768\text{ms}}$ よって、この時間以内の PWM 周期を設定する場合は“000”を設定、これ以上の PWM 周期を設定したい場合は次以降の値を検討します。
001	タイマ RD カウンタ 0(TRD0)がカウントアップする時間を、f2 に設定します。時間は、 $f2/20\text{MHz}=2/20\text{MHz}=100\text{ns}$ 設定できる PWM 周期の最大は、 $100\text{ns} \times 65,536 = \mathbf{6.5536\text{ms}}$ よって、この時間以内の PWM 周期を設定する場合は“001”を設定、これ以上の PWM 周期を設定したい場合は次以降の値を検討します。
010	タイマ RD カウンタ 0(TRD0)がカウントアップする時間を、f4 に設定します。時間は、 $f4/20\text{MHz}=4/20\text{MHz}=200\text{ns}$ 設定できる PWM 周期の最大は、 $200\text{ns} \times 65,536 = \mathbf{13.1072\text{ms}}$ よって、この時間以内の PWM 周期を設定する場合は“010”を設定、これ以上の PWM 周期を設定したい場合は次以降の値を検討します。
011	タイマ RD カウンタ 0(TRD0)がカウントアップする時間を、f8 に設定します。時間は、 $f8/20\text{MHz}=8/20\text{MHz}=400\text{ns}$ 設定できる PWM 周期の最大は、 $400\text{ns} \times 65,536 = \mathbf{26.2144\text{ms}}$ よって、この時間以内の PWM 周期を設定する場合は“011”を設定、これ以上の PWM 周期を設定したい場合は次以降の値を検討します。
100	タイマ RD カウンタ 0(TRD0)がカウントアップする時間を、f32 に設定します。時間は、 $f32/20\text{MHz}=32/20\text{MHz}=1600\text{ns}$ 設定できる PWM 周期の最大は、 $1600\text{ns} \times 65,536 = \mathbf{104.8576\text{ms}}$ よって、この時間以内の PWM 周期を設定する場合は“100”を設定します。 これ以上の PWM 周期を設定することはできません。 これ以上の PWM 周期を設定しなくても良いように、回路側を工夫してください。

今回は、PWM 波形の周期を 16ms にするので、

- ①“000”の設定…最大の PWM 周期は 3.2768ms、今回設定したい 16ms の周期を設定できないので不可
- ②“001”の設定…最大の PWM 周期は 6.5536ms、今回設定したい 16ms の周期を設定できないので不可
- ③“010”の設定…最大の PWM 周期は 13.1072ms、今回設定したい 16ms の周期を設定できないので不可
- ④“011”の設定…最大の PWM 周期は 26.2144ms、今回設定したい 16ms の周期を設定できるので OK

よって、“011”を設定します。

⑦タイマ RD ジェネラルレジスタ A0(TRDGRA0:Timer RD General register A0)、
タイマ RD ジェネラルレジスタ C0(TRDGRC0:Timer RD General register C0)の設定

タイマ RD ジェネラルレジスタ A0(TRDGRA0)に値を設定することによって、出力するPWM 波形の周期を設定します。PWM 周期は、下記の式で決まります。

$$\text{PWM 周期} = \text{タイマ RD カウンタ 0 のカウントソース} \times (\text{TRDGRA0} + 1)$$

TRDGRA0 を左辺に移動して、TRDGRA0 を求める式に変形します。

$$\text{TRDGRA0} = \text{PWM 周期} / \text{タイマ RD カウンタ 0 のカウントソース} - 1$$

今回、PWM 波形の周期を 16ms にします。タイマ RD カウンタ 0 のカウントソースとは、タイマ RD 制御レジスタ 0(TRDCR0)の bit2～0 で設定した時間のことで、今回は 400ns に設定しています。よって、タイマ RD ジェネラルレジスタ A0(TRDGRA0)は次のようになります。

$$\begin{aligned} \text{TRDGRA0} &= \text{周期} / \text{カウントソース} - 1 \\ \text{TRDGRA0} &= (16 \times 10^{-3}) / (400 \times 10^{-9}) - 1 \\ \text{TRDGRA0} &= 40000 - 1 = 39999 \end{aligned}$$

TRDGRA0 の値は、65,535 以下にする必要があります。今回の結果は、65,535 以下なので、400ns の設定で大丈夫です。65,536 以上になった場合、タイマ RD 制御レジスタ 0(TRDCR0)のカウントソース選択ビット(bit2～0)の設定を大きい時間にしてください。

■設定のポイント

- ※1…タイマ RD ジェネラルレジスタ A0(TRDGRA0)を使うときは、タイマ RD ジェネラルレジスタ C0(TRDGRC0)をバッファレジスタに設定して、ペアで使用してください。
- ※2…タイマ RD ジェネラルレジスタ A0(TRDGRA0)の設定は、イニシャライズ時に 1 回だけ行ってください。2 回目以降、周期を変更したい場合は、タイマ RD ジェネラルレジスタ C0(TRDGRC0)に値を設定してください。
- ※3…タイマ RD ジェネラルレジスタ C0(TRDGRC0)のイニシャライズ時に設定する値は、タイマ RD ジェネラルレジスタ A0(TRDGRA0)と同じ値にしてください。

プログラム例を下記に示します。

```
void main( void )
{
    // メインプログラム
    init();                               /* レジスタの初期化 */
    ~~~~~
    trdgrc0 = 19999;                       /* 2 回目以降は必ず TRDGRC0 へ行う */
    ~~~~~
    trdgrc0 = 9999;                       /* 2 回目以降は必ず TRDGRC0 へ行う */
}

void init( void )
{
    // レジスタの初期化
    ~~~~~
    trdgra0 = trdgrc0 = 39999;            /* 1 回だけ TRDGRA0 に値を設定 */
    ~~~~~                                /* TRDGRC0 にも同じ値を設定する */
}
```

詳しくは、後述するバッファ動作で説明します。

⑧タイマ RD ジェネラルレジスタ B0(TRDGRB0:Timer RD General register B0)、
タイマ RD ジェネラルレジスタ D0(TRDGRD0:Timer RD General register D0)の設定

タイマ RD ジェネラルレジスタ B0(TRDGRB0)に値を設定することによって、P2_2 端子から出力される PWM 波形の ON 幅を設定します。P2_3 端子からは、その反転した波形が出力されます。

P2_2 端子から出力される PWM 波形の ON 幅は、下記の式で決まります。

$$\text{P2}_2 \text{ 端子から出力される PWM 波形の ON 幅} = \text{タイマ RD カウンタ 0 のカウントソース} \times (\text{TRDGRB0} + 1)$$

TRDGRB0 を左辺に移動して、TRDGRB0 を求める式に変形します。

$$\text{TRDGRB0} = \text{P2}_2 \text{ 端子から出力される PWM 波形の ON 幅} / \text{タイマ RD カウンタ 0 のカウントソース} - 1$$

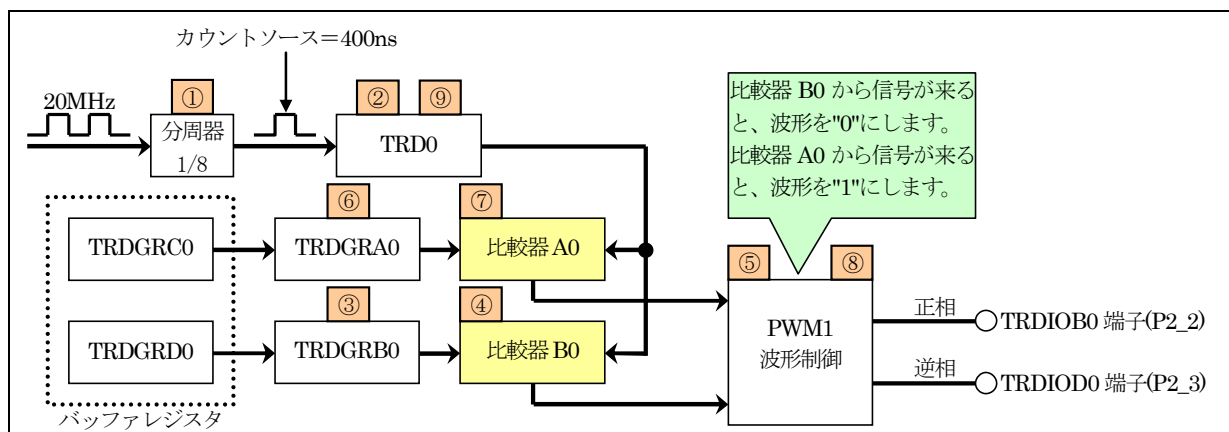
今回、タイマ RD カウンタ 0 のカウントソースは 400ns です。例えば ON 幅を 1ms にするなら、TRDGRB0 の値は次のようになります。よって、タイマ RD ジェネラルレジスタ B0(TRDGRB0)は次のようになります。

$$\text{TRDGRB0} = \text{ON 幅} / \text{カウントソース} - 1$$

$$\text{TRDGRB0} = (1 \times 10^{-3}) / (400 \times 10^{-9}) - 1$$

$$\text{TRDGRB0} = 2500 - 1 = 2499$$

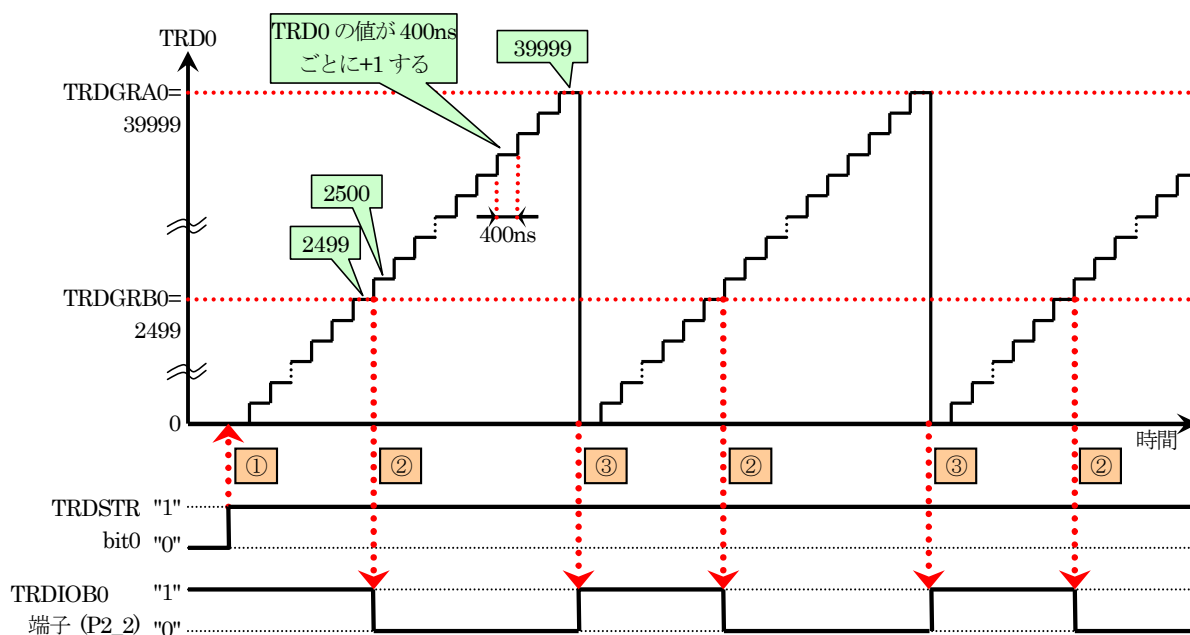
■仕組み



①	分周器には、クリスタルの 20MHz のパルスが入力されます。分周器は、タイマ RD 制御レジスタ 0 (TRDCR0) のカウントソース選択ビットで何分周にするか設定します。今回は f8 を選択しています。これは、8 分周することです。よって分周器からの出力は、 $8 / 20\text{MHz} = 2.5\text{MHz}$ のパルスが出力されます。時間は、 $1 / 2.5\text{M} = 400\text{ns}$ となります。
②	TRD0 は、400ns ごとにカウントアップ(+1 ずつ)します。TRD0 の最大値は 65,535 で、その次の値は 0 に戻りカウントアップし続けます。

<p>③</p> <p>④</p>	<p>TRDGRB0 には、波形の ON 幅を設定します。 TRDGRB0 と TRD0 の値は、常に比較器 B0 によって比較されています。次の式が成り立つと、PWM1 波形制御回路へ信号を送ります。</p> $\text{TRDGRB0} + 1 = \text{TRD0}$ <p>TRD0 は 400ns ごとに +1 します。TRDGRB0 には 2499 が設定されているとします。マイコンが起動したとき、TRD0 は 0 なので</p> $2499(\text{TRDGRB0}) + 1 \neq 0(\text{TRD0})$ <p>となり、成り立ちません。400ns 後、TRD0 は +1 されて 1 になります。</p> $2499(\text{TRDGRB0}) + 1 \neq 1(\text{TRD0})$ <p>まだ成り立ちません。式が成り立つ TRD0 が 2500 になるには $2500 \times 400\text{ns} = 1\text{ms}$ です。よって、1ms 経つと</p> $2499(\text{TRDGRB0}) + 1 = 2500(\text{TRD0})$ <p>が成り立ち、比較器 B0 から信号が PWM1 波形制御部分へ出力されます。</p>
<p>⑤</p>	<p>PWM 波形制御部分は、比較器 B0 から信号が送られてくると、波形を“0”にします。</p>
<p>⑥</p> <p>⑦</p>	<p>TRDGRA0 には、波形の周期を設定します。 TRDGRA0 と TRD0 の値は、常に比較器 A0 によって比較されています。次の式が成り立つと、PWM1 波形制御回路へ信号を送ります。</p> $\text{TRDGRA0} + 1 = \text{TRD0}$ <p>TRD0 は 400ns ごとに +1 します。TRDGRA0 には 39999 が設定されているとします。マイコンが起動したとき、TRD0 は 0 なので</p> $39999(\text{TRDGRA0}) + 1 \neq 0(\text{TRD0})$ <p>となり、成り立ちません。400ns 後、TRD0 は +1 されて 1 になります。</p> $39999(\text{TRDGRA0}) + 1 \neq 1(\text{TRDGRA0})$ <p>まだ成り立ちません。式が成り立つ TRD0 が 40000 になるには $40000 \times 400\text{ns} = 16\text{ms}$ です。よって、16ms 経つと</p> $39999(\text{TRDGRA0}) + 1 = 40000(\text{TRD0})$ <p>が成り立ち、比較器 A0 から信号が PWM1 波形制御部分へ出力されます。</p>
<p>⑧</p> <p>⑨</p>	<p>PWM 波形制御部分は、比較器 A0 から信号が送られてくると、波形を“1”にします。 これと同時に、比較器 A0 は TRD0 の値を 0 にクリアします。TRD0 は、0 からカウントし直します。</p>

TRD0 の値と波形の関係を図解すると下記ようになります。

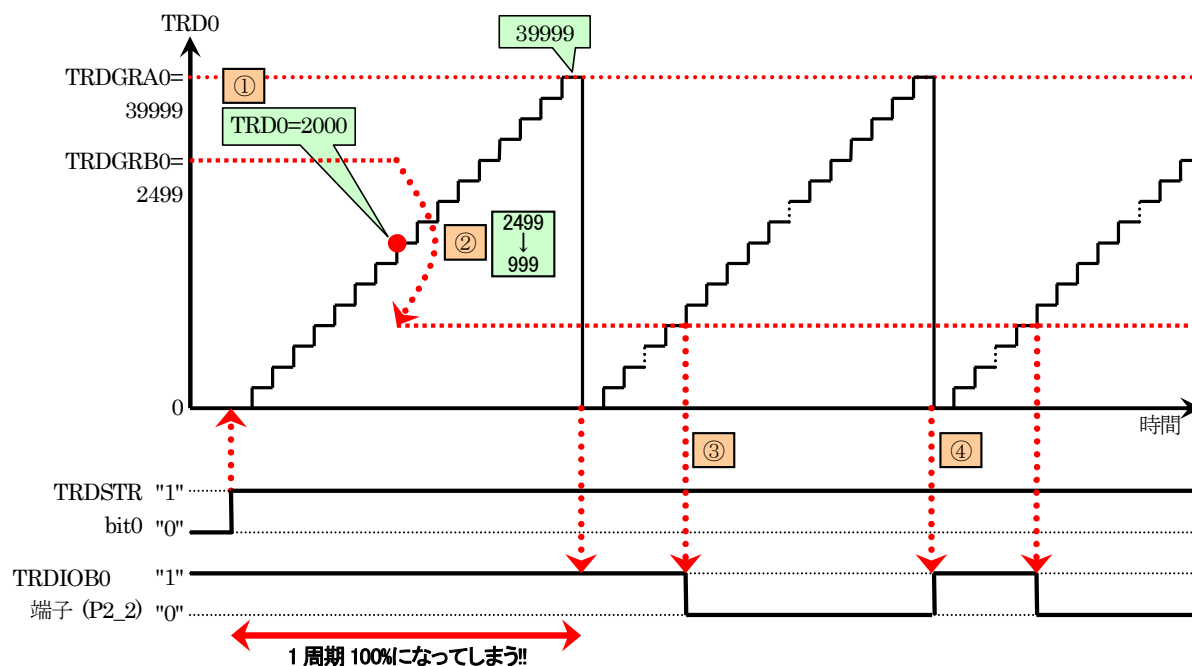


①	TRDSTR の bit0 を"1"にすると、TRD0 のカウントが開始されます。
②	<p>TRD0=(TRDGRB0+1)になると、P2_2 端子が"0"になります。 TRDGRB0 の値は 2499、TRD0 が+1 する間隔は 400ns です。 よって、P2_2 端子が"1"の時間は、 P2_2 端子が"1"の時間 = (TRDGRB0+1)×TRD0 が+1 する間隔 = (2499+1)×400ns = 1,000,000 [ns] = 1[ms]</p>
③	<p>TRD0=(TRDGRA0+1)になると、P2_2 端子が"1"になります。同時に TRD0 が 0 にクリアされ、また 0 から値が増えていきます。 TRDGRA0 の値は 39999、TRD0 が+1 する間隔は 400ns です。 よって、P2_2 端子の PWM 周期は、 P2_2 端子の PWM 周期 = (TRDGRA0+1)×TRD0 が+1 する間隔 = (39999+1)×400ns = 16,000,000 [ns] = 16[ms]</p> <p>ちなみに、 「PWM 周期」=「ON("1")の時間」+「OFF("0")の時間」 ですので、"0"の時間は、 「OFF("0")の時間」=「PWM 周期」-「ON("1")の時間」 = 16[ms] - 1[ms] = 15[ms] となります。</p>

■バッファレジスタはなぜ必要か

リセット同期 PWM モードを使用する場合、バッファレジスタというレジスタを使います。なぜバッファレジスタを使わなければならないのでしょうか。

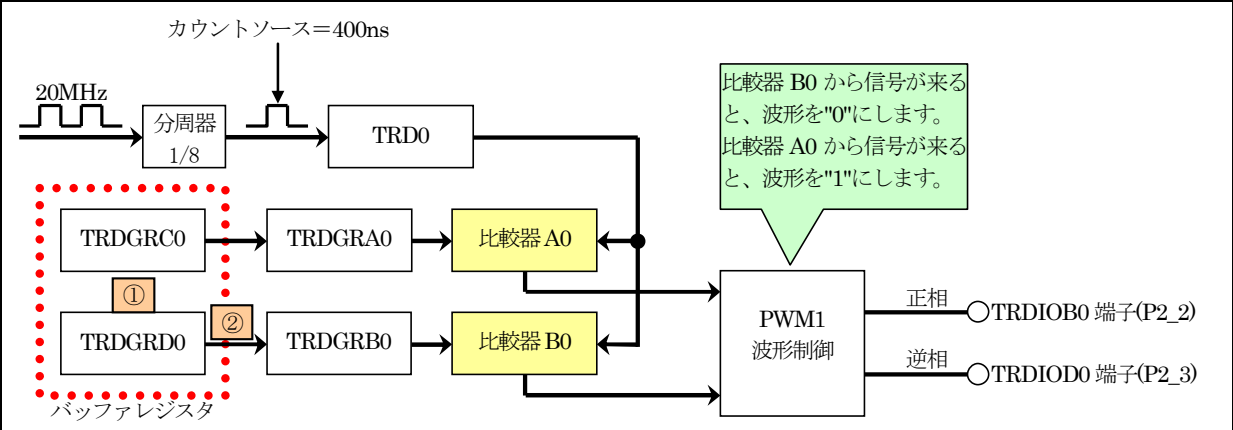
例えば、TRDGRB0=2499、TRD0=2000 のときに、プログラムで TRDGRB0 の値を 999 に変更したとします。



① ②	<p>TRD0 が 2000 のときに(①)、プログラムで TRDGRB0 の値を 2499 から 999 に変更します(②)。</p> <p>TRD0 の値が 400ns ごとに増えていき 40000 になりました。40000 になると「$TRD0 = (TRDGRA0 + 1)$」が成り立ち、P2_2 端子が「1」になります。同時に TRD0 が 0 にクリアされます。この間、「$TRD0 = (TRDGRB0 + 1)$」が成り立つ条件がありませんでした。そのため、今回の 1 周期分すべてが「1」となり 100%出力になってしまいます。</p> <p>今回のように、「$TRD0 = (TRDGRB0 + 1)$」が成り立つ前に TRDGRB0 の値を TRD0 より小さい値に変更すると、PWM100%という想定外の値になります。</p>
③	次の周期は、 $TRD0 = (TRDGRB0 + 1)$ になるため、P2_2 端子が「0」になります。
④	$TRD0 = (TRDGRA0 + 1)$ になるため、P2_2 端子が「1」になります。同時に TRD0 が 0 になります。

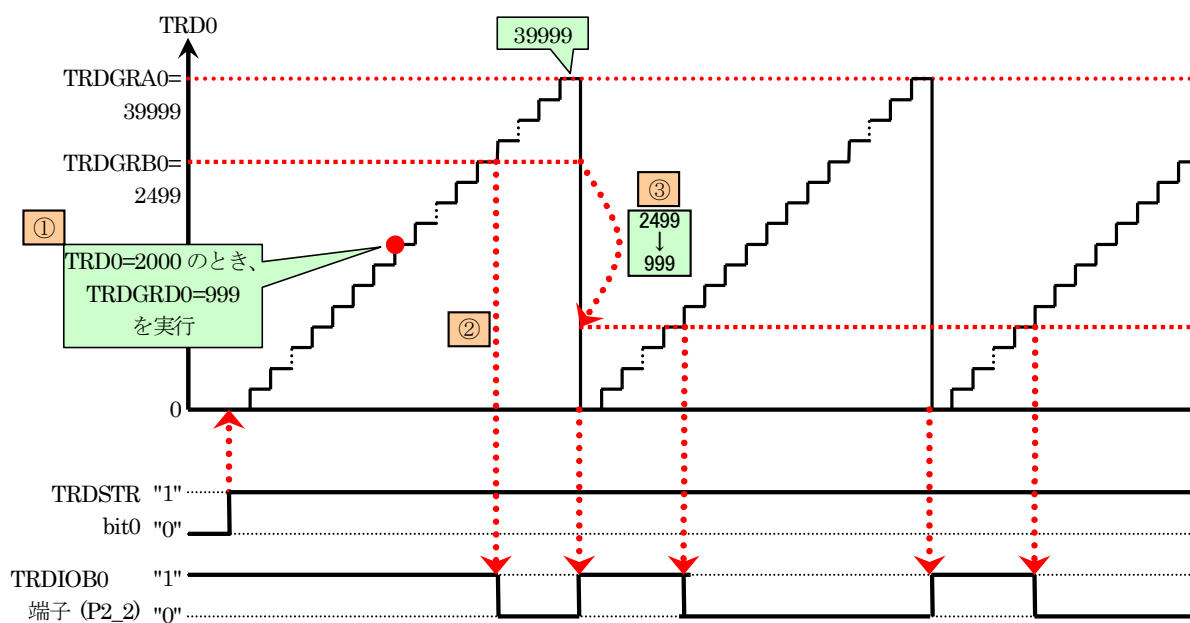
このように、TRDGRB0 の値を変えるタイミングによっては PWM100%という想定外の波形になってしまいます。モータを制御していたら、一瞬ですが 100%の回転となってしまう、暴走してしまうかもしれません。

そこで、バッファレジスタというのを使います(点線で囲った部分)。



①	<p>TRDGRB0 のバッファレジスタは、TRDGRD0 です。</p> <p>P2.2 端子の ON 幅の設定は、TRDGRB0 ではなく TRDGRD0 に値を設定することにより行います。</p> <p>プログラム例) TRDGRD0 = 2499;</p> <p>プログラムで TRDGRB0 の値を、直接変更しません。直接変更すると、前記のとおり 100% の波形が出力されることがあります。ただし、最初は TRDGRB0 に値が設定されていけませんので、イニシャライズ時に 1 回だけ値を設定してください。このとき、TRDGRB0 と TRDGRD0 には同じ値を設定します。</p>
②	<p>「TRD0=(TRDGRA0+1)」が成り立つと、下記が実行されます。</p> <ul style="list-style-type: none">•P2.2 端子が"1"になります。•TRD0 の値が 0 にクリアされます。•TRDGRD0 の値が、TRDGRB0 に転送されます。 <p>TRDGRB0 の値は、TRD0 の値が 0 になると同時に更新されます。これ以外のタイミングで TRDGRB0 の値が変わることはありません。そのため、前記の不具合が発生しないことになります。これがバッファレジスタを使う目的です。</p>

TRD0 の値と波形の関係を図解すると下記ようになります。



①	TRD0 が 2000 のときに、プログラムで TRDGRD0 の値を 2499 から 999 に変更します。
②	TRDGRB0 の値は変わっていません。そのため、「TRD0=(TRDGRB0+1)」が成り立ち、P2_2 端子が"0"になります。
③	<p>「TRD0=(TRDGRA0+1)」が成り立つと、下記が実行されます。</p> <ul style="list-style-type: none"> ・P2_2 端子が"1"になります。 ・TRD0 の値が 0 になります。 ・TRDGRD0 の値が、TRDGRB0 に転送されます。 <p>このタイミングで、TRDGRB0 の値が 999 に変更されます。次に「TRD0=(TRDGRB0+1)」が成り立つタイミングで、P2_2 端子が"0"に成ります。</p>

このように、バッファレジスタを使うと PWM 波形が 100%になるという不具合が発生しなくなります。

■0%出力

P2_2 端子が“0”になる時間、すなわち P2_2 端子が“1”の時間は、下記のようにした。

$$\text{P2_2 端子が“1”の時間} = (\text{TRDGRB0} + 1) \times \text{TRD0 が+1 する間隔}$$

0%出力にするには、“1”の時間を 0 にすれば良いことになります。すなわち TRDGRB0 の値を 0 にします。

$$\begin{aligned} \text{P2_2 端子が“1”の時間} &= (0 + 1) \times 400\text{ns} \\ &= 1 \times 400\text{ns} = 400\text{ns} \end{aligned}$$

このように、TRDGRB0 に 0 を代入しても完全な“0”になりません。400ns の時間だけ“1”になります。

リセット同期 PWM モードでは特殊な方法で 0%にします。それは、TRDGRB0 の値を TRDGRA0 と同じ値にします。そうすると、次のことが同時に起こります。

- ・「TRD0=(TRDGRB0+1)」が成り立ち、P2_2 端子を“0”にします。
- ・「TRD0=(TRDGRA0+1)」が成り立ち、P2_2 端子を“1”にします。

相反することが同時に起こります。**この場合 R8C/35A は、P2_2 端子を“0”にすることを優先させます。**

よって、TRDGRB0=TRDGRA0 にすると端子を“0”にすることが優先され、端子を“1”にする動作を行いません。

PWM 出力を 0%にしたい場合、次のようにプログラムします。設定は、TRDGRB0 ではなく、バッファレジスタである TRDGRD0 に設定します。

```
trdgrd0 = trdgra0;           // バッファレジスタに設定する
```

■100%出力

「TRD0=(TRDGRB0+1)」が成り立つと、P2_2 端子が“0”になります。100%出力にするには、「TRD0=(TRDGRB0+1)」が成り立たない値を TRDGRB0 に設定すれば良いことになります。TRD0 は、(TRDGRA0+1)と一致すると 39999→0 になります。すなわち、TRDGRA0 より大きい値を設定します。今回は、TRDGRA0 より 1 つ大きい値を設定することになります。

PWM 出力を 100%にしたい場合、次のようにプログラムします。

```
trdgrd0 = 40000;           // バッファレジスタに設定する
```

ただし、上記プログラムは、TRDGRA0 が 39999 (周期が 16ms) のときだけです。TRDGRA0 がどのような値でも 100%にするために、次のプログラムの方が良いでしょう。

```
trdgrd0 = trdgra0 + 1;     // バッファレジスタに設定する
```

このとき、TRDGRA0 が 65535 のときは、TRDGRD0 に 65536 を設定することはできないので、100%出力はできません。

■設定のポイント

- ※1…タイマ RD ジェネラルレジスタ B0(TRDGRB0)を使うときは、タイマ RD ジェネラルレジスタ D0(TRDGRD0)をバッファレジスタに設定して、ペアで使用してください。
- ※2…タイマ RD ジェネラルレジスタ B0(TRDGRB0)の設定は、イニシャライズ時に 1 回だけ行ってください。2 回目以降、ON 幅を変更したい場合は、タイマ RD ジェネラルレジスタ D0(TRDGRD0)に値を設定してください。
- ※3…タイマ RD ジェネラルレジスタ D0(TRDGRD0)のイニシャライズ時に設定する値は、タイマ RD ジェネラルレジスタ B0(TRDGRB0)と同じ値にしてください。

- ⑨タイマ RD ジェネラルレジスタ A1(TRDGRA1:Timer RD General register A1)、
タイマ RD ジェネラルレジスタ C1(TRDGRC1:Timer RD General register C1)の設定

タイマ RD ジェネラルレジスタ A1(TRDGRA1)に値を設定することによって、P2_4 端子から出力される PWM 波形の ON 幅を設定します。P2_6 端子からは、その反転した波形が出力されます。

P2_4 端子から出力される PWM 波形の ON 幅は、下記の式で決まります。

$$\text{P2}_4 \text{ 端子から出力される PWM 波形の ON 幅} = \text{タイマ RD カウンタ 0 のカウントソース} \times (\text{TRDGRA1} + 1)$$

TRDGRA1 を左辺に移動して、TRDGRA1 を求める式に変形します。

$$\text{TRDGRA1} = \text{P2}_4 \text{ 端子から出力される PWM 波形の ON 幅} / \text{タイマ RD カウンタ 0 のカウントソース} - 1$$

今回、タイマ RD カウンタ 0 のカウントソースは 400ns です。ON 幅を 10ms にするなら、タイマ RD ジェネラルレジスタ A1(TRDGRA1)は次のようになります。

$$\begin{aligned} \text{TRDGRA1} &= \text{ON 幅} / \text{カウントソース} - 1 \\ \text{TRDGRA1} &= (10 \times 10^{-3}) / (400 \times 10^{-9}) - 1 \\ \text{TRDGRA1} &= 25000 - 1 = 24999 \end{aligned}$$

■設定のポイント

- ※1…タイマ RD ジェネラルレジスタ A1(TRDGRA1)を使うときは、タイマ RD ジェネラルレジスタ C1(TRDGRC1)をバッファレジスタに設定して、ペアで使用してください。
- ※2…タイマ RD ジェネラルレジスタ A1(TRDGRA1)の設定は、イニシャライズ時に 1 回だけ行ってください。2 回目以降、ON 幅を変更したい場合は、タイマ RD ジェネラルレジスタ C1(TRDGRC1)に値を設定してください。
- ※3…タイマ RD ジェネラルレジスタ C1(TRDGRC1)のイニシャライズ時に設定する値は、タイマ RD ジェネラルレジスタ A1(TRDGRA1)と同じ値にしてください。

- ⑩タイマ RD ジェネラルレジスタ B1(TRDGRB1:Timer RD General register B1)、
タイマ RD ジェネラルレジスタ D1(TRDGRD1:Timer RD General register D1)の設定

タイマ RD ジェネラルレジスタ B1(TRDGRB1)に値を設定することによって、P2_5 端子から出力される PWM 波形の ON 幅を設定します。P2_7 端子からは、その反転した波形が出力されます。

P2_5 端子から出力される PWM 波形の ON 幅は、下記の式で決まります。

$$\text{P2}_5 \text{ 端子から出力される PWM 波形の ON 幅} = \text{タイマ RD カウンタ 0 のカウントソース} \times (\text{TRDGRB1} + 1)$$

TRDGRB1 を左辺に移動して、TRDGRB1 を求める式に変形します。

$$\text{TRDGRB1} = \text{P2}_5 \text{ 端子から出力される PWM 波形の ON 幅} / \text{タイマ RD カウンタ 0 のカウントソース} - 1$$

今回、タイマ RD カウンタ 0 のカウントソースは 400ns です。ON 幅を 15ms にするなら、タイマ RD ジェネラルレジスタ B1(TRDGRB1)は次のようになります。

$$\text{TRDGRB1} = \text{ON 幅} / \text{カウントソース} - 1$$

$$\text{TRDGRB1} = (15 \times 10^{-3}) / (400 \times 10^{-9}) - 1$$

$$\text{TRDGRB1} = 37500 - 1 = 37499$$

■設定のポイント

- ※1…タイマ RD ジェネラルレジスタ B1(TRDGRB1)を使うときは、タイマ RD ジェネラルレジスタ D1(TRDGRD1)をバッファレジスタに設定して、ペアで使用してください。
- ※2…タイマ RD ジェネラルレジスタ B1(TRDGRB1)の設定は、イニシャライズ時に 1 回だけ行ってください。2 回目以降、ON 幅を変更したい場合は、タイマ RD ジェネラルレジスタ D1(TRDGRD1)に値を設定してください。
- ※3…タイマ RD ジェネラルレジスタ D1(TRDGRD1)のイニシャライズ時に設定する値は、タイマ RD ジェネラルレジスタ B1(TRDGRB1)と同じ値にしてください。

⑪タイマ RD スタートレジスタ (TRDSTR: Timer RD start register) の設定

TRD0 をカウントさせるか、停止させるか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～4		"0000"を設定	0000
bit3	TRD1 カウント動作選択ビット csel1_trdstr	0: TRDGRA1 レジスタとのコンペアー一致でカウント停止 1: TRDGRA1 レジスタとのコンペアー一致後もカウント継続 今回は"1"を設定します。	1
bit2	TRD0 カウント動作選択ビット csel0_trdstr	0: TRDGRA0 レジスタとのコンペアー一致でカウント停止 1: TRDGRA0 レジスタとのコンペアー一致後もカウント継続 今回は"1"を設定します。	1
bit1	TRD1 カウント開始フラグ(注4) tstart1_trdstr	0: カウント停止(注2) 1: カウント開始 設定した瞬間から、TRD1 のカウントが開始されます。リセット同期 PWM モードでは TRD1 は使いませんので"0"を設定します。	0
bit0	TRD0 カウント開始フラグ(注3) tstart0_trdstr	0: カウント停止(注1) 1: カウント開始 設定した瞬間から、TRD0 のカウントが開始されます。 "1"を設定します。	1

注 1. bit2 が"1"に設定されているとき、bit0 へ"0"を書いてください。

注 2. bit3 が"1"に設定されているとき、bit1 へ"0"を書いてください。

注 3. bit2 が"0"でコンペアー一致信号(TRDIOA0)が発生したとき、"0"(カウント停止)になります。

注 4. bit3 が"0"でコンペアー一致信号(TRDIOA1)が発生したとき、"0"(カウント停止)になります。

タイマ RD スタートレジスタ (TRDSTR) の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	1	1	0	1
16 進数	0				d			

21.6.2 main関数

P2_2 端子に繋がっている LED などへ PWM 信号を出力します。PWM のデューティ比は、マイコンボードのディップスイッチで設定します。

```

34 : void main( void )
35 : {
36 :     init();                /* 初期化                */
37 :
38 :     while( 1 ) {
39 :         trdgrd0 = 39998 * dipsw_get() / 15;
40 :     }
41 : }
```

マイコンボードのディップスイッチの値に応じて、タイマ RD ジェネラルレジスタ D0(TRDGRD0)に値を設定しています。ディップスイッチとデューティ比の関係を、下表に示します。

ディップスイッチの値 (dipsw_get 関数)	TRDGRD0 の値 (A)	TRD0 と一致する値 (A)+1	デューティ比 ((A)+1) / 40000
0	$39998 * 0 / 15 = 0$	1	0.0025%
1	$39998 * 1 / 15 = 2666$	2667	6.6675%
2	$39998 * 2 / 15 = 5333$	5334	13.3350%
3	$39998 * 3 / 15 = 7999$	8000	20.0000%
4	$39998 * 4 / 15 = 10666$	10667	26.6675%
5	$39998 * 5 / 15 = 13332$	13333	33.3325%
6	$39998 * 6 / 15 = 15999$	16000	40.0000%
7	$39998 * 7 / 15 = 18665$	18666	46.6650%
8	$39998 * 8 / 15 = 21332$	21333	53.3325%
9	$39998 * 9 / 15 = 23998$	23999	59.9975%
10	$39998 * 10 / 15 = 26665$	26666	66.6650%
11	$39998 * 11 / 15 = 29331$	29332	73.3300%
12	$39998 * 12 / 15 = 31998$	31999	79.9975%
13	$39998 * 13 / 15 = 34664$	34665	86.6625%
14	$39998 * 14 / 15 = 37331$	37332	93.3300%
15	$39998 * 15 / 15 = 39998$	39999	99.9975%

39 行目だけで 0%～100%まで対応させる、簡単な計算式です。ただし、0%は完全な 0%ではなく、400ns だけ“1”になります。100%も完全な 100%ではなく、400ns だけ“0”になります。

完全な 0%、100%にも対応するには、下記プログラムのように 0%、100%、それ以外で場合分けします。

```
void main( void )
{
    init();                      /* 初期化                      */

    while( 1 ) {
        if( dipsw_get() == 0 ) {
            // 0%
            trdgrd0 = trdgra0;
        } else if( dipsw_get() == 15 ) {
            // 100%
            trdgrd0 = trdgra0 + 1;
        } else {
            // 0%、100%以外
            trdgrd0 = 39999 * dipsw_get() / 15;
        }
    }
}
```

ディップスイッチとデューティ比の関係を、下表に示します。

ディップスイッチの値 (dipsw_get 関数)	TRDGRD0 の値 (A)	TRD0 と一致する値 (A)+1	デューティ比 ((A)+1)/40000
0	39999 を直接指定	40000 (“0”と“1”が同時に起こり、 “0”が優先される)	0%
1	$39999 * 1 / 15 = 2666$	2667	6.6675%
2	$39999 * 2 / 15 = 5333$	5334	13.3350%
3	$39999 * 3 / 15 = 7999$	8000	20.0000%
4	$39999 * 4 / 15 = 10666$	10667	26.6675%
5	$39999 * 5 / 15 = 13332$	13333	33.3325%
6	$39999 * 6 / 15 = 15999$	16000	40.0000%
7	$39999 * 7 / 15 = 18665$	18666	46.6650%
8	$39999 * 8 / 15 = 21332$	21333	53.3325%
9	$39999 * 9 / 15 = 23998$	23999	59.9975%
10	$39999 * 10 / 15 = 26665$	26666	66.6650%
11	$39999 * 11 / 15 = 29331$	29332	73.3300%
12	$39999 * 12 / 15 = 31998$	31999	79.9975%
13	$39999 * 13 / 15 = 34664$	34665	86.6625%
14	$39999 * 14 / 15 = 37331$	37332	93.3300%
15	40000 を直接指定	一致せず	100%

21.7 演習

- (1) 周期が出力される端子(P2_1)の PWM 出力を禁止して、通常の I/O ポートにしない。このとき、この端子は出力端子として"0"を出力しない。
- (2) 逆相の端子(P2_3、P2_6、P2_7 の 3 つとも)の PWM 出力を禁止して、通常の I/O ポートにしない。このとき、この端子は出力端子として"0"を出力しない。
※プログラムは、(1)の状態から改造するものとする。
- (3) P2_4 端子も P2_2 端子と同様に、ディップスイッチに合わせて PWM 信号が出力されるようにしない。出力の仕方は、サンプルプログラム 39 行目の「39998 * dipsw_get() / 15」を使用することとする。
※プログラムは、(2)の状態から改造するものとする。
- (4) P2_5 端子も P2_2 端子と同様に、ディップスイッチに合わせて PWM 信号が出力されるようにしない。出力の仕方は、サンプルプログラム 39 行目の「39998 * dipsw_get() / 15」を使用することとする。
※プログラムは、(3)の状態から改造するものとする。
- (5) P2_4 端子、P2_5 端子の PWM 出力を禁止して、通常の I/O ポートにしない。このとき、この端子は出力端子として"0"を出力しない。
※プログラムは、(4)の状態から改造するものとする。

22. タイマRDによるPWM波形出力(PWMモード) (プロジェクト:timer_rd_pwm6)

22.1 概要

本章では、PWM 波形を出力する方法を紹介します。今回は、タイマ RD を PWM モードで使用して、同じ周期の PWM 波形を 6 本出力します。タイマ RD の初期設定後は、プログラムが関与しなくても PWM 波形を出力し続けます。プログラムでは、PWM 波形出力処理以外の処理をすることができます。ただし、リセット同期 PWM モードで設定していたバッファレジスタは PWM モードでは使えないため、PWM の ON 幅を変更するときはプログラムでタイミングを計ります。この方法も解説します。

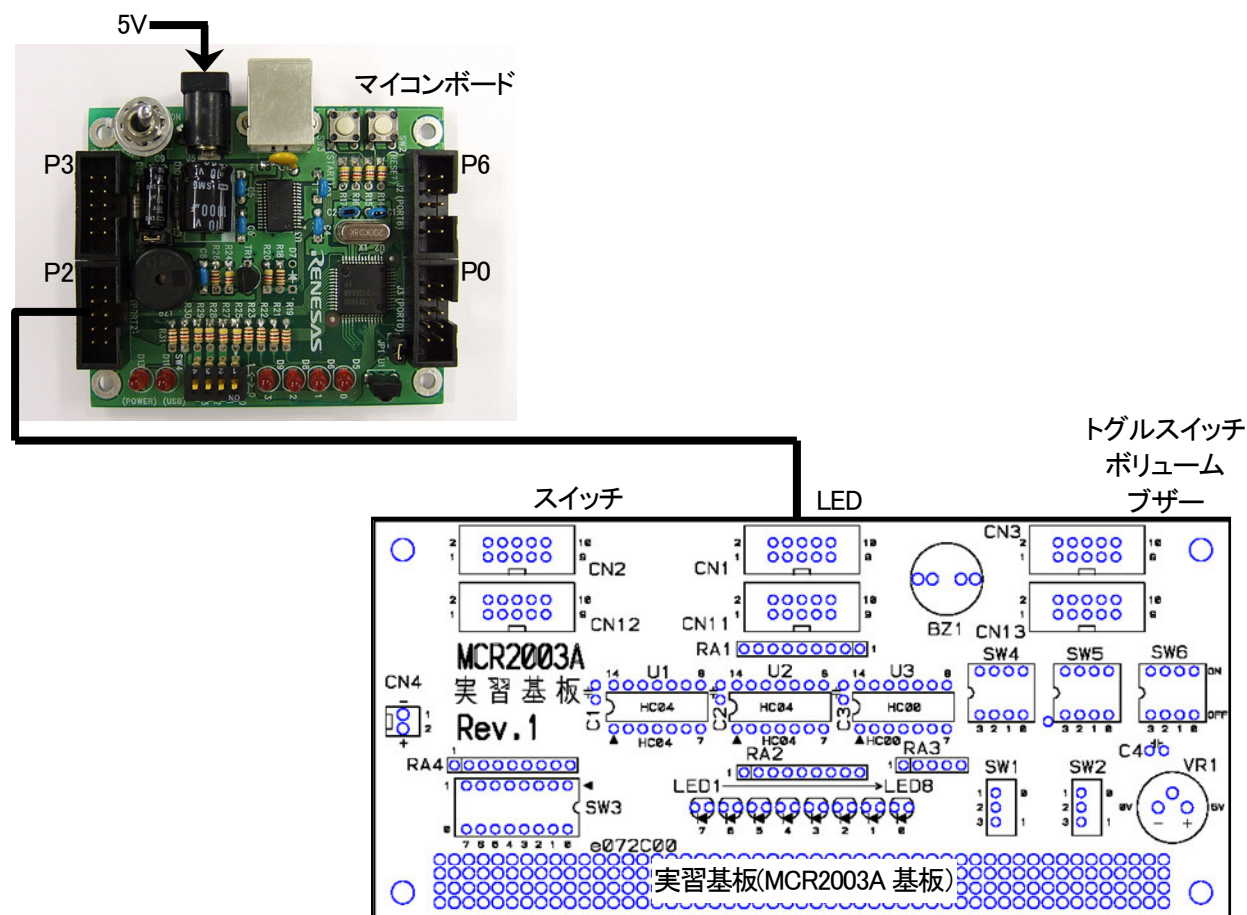
22.2 接続

■使用ポート

マイコンのポート	接続内容
P5_7、P4_5、P4_4、P4_3	マイコンボード上のディップスイッチです。
P2 (J7)	実習基板の LED 部など、出力機器を接続します。

■接続例

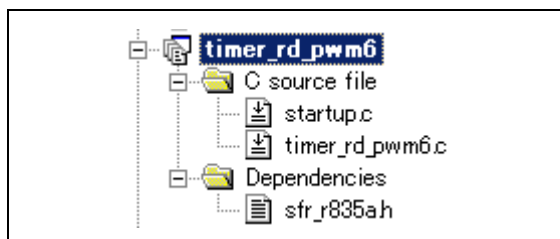
実習基板を使ったときの接続例を次に示します。



■操作方法

マイコンボードのディップスイッチ(SW4)の値 0～15 によって、LED の点灯する明るさが変わります。このとき、どの LED が明るくなるか観察してください。

22.3 プロジェクトの構成



	ファイル名	内容
1	startup.c	固定割り込みベクタアドレスの設定、スタートアッププログラム、RAM の初期化(初期値のないグローバル変数、初期値のあるグローバル変数の設定)などを行います。このファイルは共通で、どのプロジェクトもこのファイルから実行されます。
2	timer_rd_pwm6.c	実際に制御するプログラムが書かれています。R8C/35A の内蔵周辺機能(SFR)の初期化も行います。
3	sfr_r835a.h	R8C/35A マイコンの内蔵周辺機能を制御するためのレジスタ (Special Function Registers)を定義したファイルです。

22.4 プログラム「timer_rd_pwm6.c」

```

1 :  /******
2 :  /* 対象マイコン R8C/35A
3 :  /* ファイル内容 タイマRDのPWMモードを使ってPWM6本出力
4 :  /* バージョン Ver. 1. 20
5 :  /* Date 2010. 04. 19
6 :  /* Copyright ルネサスマイコンカーラー事務局
7 :  /* 日立インターメディックス株式会社
8 :  /******
9 :  /*
10 :  入力 : マイコンボードのディップスイッチ(4bit)
11 :  出力 : P2_1, P2_2, P2_3, P2_5, P2_6, P2_7端子からPWM出力
12 :
13 :  マイコンボードのディップスイッチ(4bit)から入力した割合により、
14 :  ポート2の6端子からPWM波形を出力します。
15 :  */
16 :
17 :  /*=====*/
18 :  /* インクルード */
19 :  /*=====*/
20 :  #include "sfr_r835a.h" /* R8C/35A SFRの定義ファイル */
21 :
22 :  /*=====*/
23 :  /* シンボル定義 */
24 :  /*=====*/
25 :
26 :  /*=====*/
27 :  /* プロトタイプ宣言 */
28 :  /*=====*/
29 :  void init( void );
30 :  unsigned char dipsw_get( void );
31 :
32 :  /******
33 :  /* メインプログラム */
34 :  /******
35 :  void main( void )
36 :  {
37 :      unsigned int pwm;
38 :

```



```

39 :      init();                                /* 初期化 */
40 :
41 :      while( 1 ) {
42 :          if( dipsw_get() == 0 ) {
43 :              // trdgra0, trdgralと同じにすると0%出力
44 :              pwm = 39999;
45 :          } else if( dipsw_get() == 15 ) {
46 :              // trdgra0, trdgralより大きくすると100%出力
47 :              pwm = 40000;
48 :          } else {
49 :              // 1~14なら割合に応じてPWM出力する
50 :              pwm = (long)39999 * dipsw_get() / 15;
51 :          }
52 :
53 :          trdgrb0 = pwm;                        /* P2_2のON幅設定 */
54 :          trdgrc0 = pwm;                        /* P2_1のON幅設定 */
55 :          trdgrd0 = pwm;                        /* P2_3のON幅設定 */
56 :          trdgrb1 = pwm;                        /* P2_5のON幅設定 */
57 :          trdgrc1 = pwm;                        /* P2_6のON幅設定 */
58 :          trdgrd1 = pwm;                        /* P2_7のON幅設定 */
59 :      }
60 : }
61 :
62 : /*****
63 : /* R8C/35A スペシャルファンクションレジスタ (SFR)の初期化
64 : /*****
65 : void init( void )
66 : {
67 :     int i;
68 :
69 :     /* クロックをXINクロック (20MHz)に変更 */
70 :     prc0 = 1;                                /* プロテクト解除 */
71 :     cm13 = 1;                                /* P4_6, P4_7をXIN-XOUT端子にする */
72 :     cm05 = 0;                                /* XINクロック発振 */
73 :     for(i=0; i<50; i++ );                    /* 安定するまで少し待つ(約10ms) */
74 :     ocd2 = 0;                                /* システムクロックをXINにする */
75 :     prc0 = 0;                                /* プロテクトON */
76 :
77 :     /* ポートの入出力設定 */
78 :     prc2 = 1;                                /* PD0のプロテクト解除 */
79 :     pd0 = 0x00;                              /* スイッチなど入力 */
80 :     p1 = 0x0f;                               /* 3-0:LEDは消灯 */
81 :     pd1 = 0xdf;                              /* 5:RXD0 4:TXD0 3-0:LED */
82 :     pd2 = 0xfe;                              /* 0:PushSW */
83 :     pd3 = 0xfb;                              /* 4:Buzzer 2:IR */
84 :     pd4 = 0x80;                              /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF */
85 :     pd5 = 0x40;                              /* 7:DIP SW */
86 :     pd6 = 0xff;                              /* LEDなど出力 */
87 :
88 :     /* タイマRD PWMモードの設定 (6本のPWM出力) */
89 :     trdpmr = 0x77;                          /* 各端子PWMモードにするかしないか */
90 :     trdfcr = 0x80;                          /* PWMモードに設定 */
91 :     trdoerl = 0x11;                         /* 各端子出力許可するかしないか */
92 :     trdpsr0 = 0x68;                         /* TRDIOB0, C0, D0端子設定 */
93 :     trdpsr1 = 0x54;                         /* TRDIOB1, C1, D1端子設定 */
94 :     trdocr = 0x00;                          /* 各端子初期出力設定 */
95 :     trdcr0 = 0x23;                          /* TRD0ソースカウンタの選択:f8 */
96 :     trdcr1 = 0x23;                          /* TRD1ソースカウンタの選択:f8 */
97 :     trdpocr0 = 0x00;                        /* B0~D0のアクティブレベル設定 */
98 :     trdpocr1 = 0x00;                        /* B1~D1のアクティブレベル設定 */
99 :     trdgra0 = 39999;                        /* TRD0の周期設定 */
100 :    trdgral = 39999;                        /* TRD1の周期設定 */
101 :    trdgrb0 = 0;                            /* P2_2のON幅設定 */
102 :    trdgrc0 = 0;                            /* P2_1のON幅設定 */
103 :    trdgrd0 = 0;                            /* P2_3のON幅設定 */
104 :    trdgrb1 = 0;                            /* P2_5のON幅設定 */
105 :    trdgrc1 = 0;                            /* P2_6のON幅設定 */
106 :    trdgrd1 = 0;                            /* P2_7のON幅設定 */
107 :    trdstr = 0x0f;                          /* TRD0, TRD1カウンタ開始 */
108 : }
109 :
110 : /*****
111 : /* ディップスイッチ値読み込み
112 : /* 戻り値 スイッチ値 0~15
113 : /*****
114 : unsigned char dipsw_get( void )
115 : {
116 :     unsigned char sw, sw1, sw2;
117 :
118 :     sw1 = (p5>>4) & 0x08;                /* ディップスイッチ読み込み3 */
119 :     sw2 = (p4>>3) & 0x07;                /* ディップスイッチ読み込み2, 1, 0 */
120 :     sw = sw1 | sw2;                       /* P5とP4の値を合わせる */
121 :
122 :     return sw;
123 : }
124 :
125 : /*****
126 : /* end of file
127 : /*****

```

22.5 プログラムの解説

22.5.1 init関数(タイマRDの設定)

タイマ RD を使った PWM モードの設定を行います。

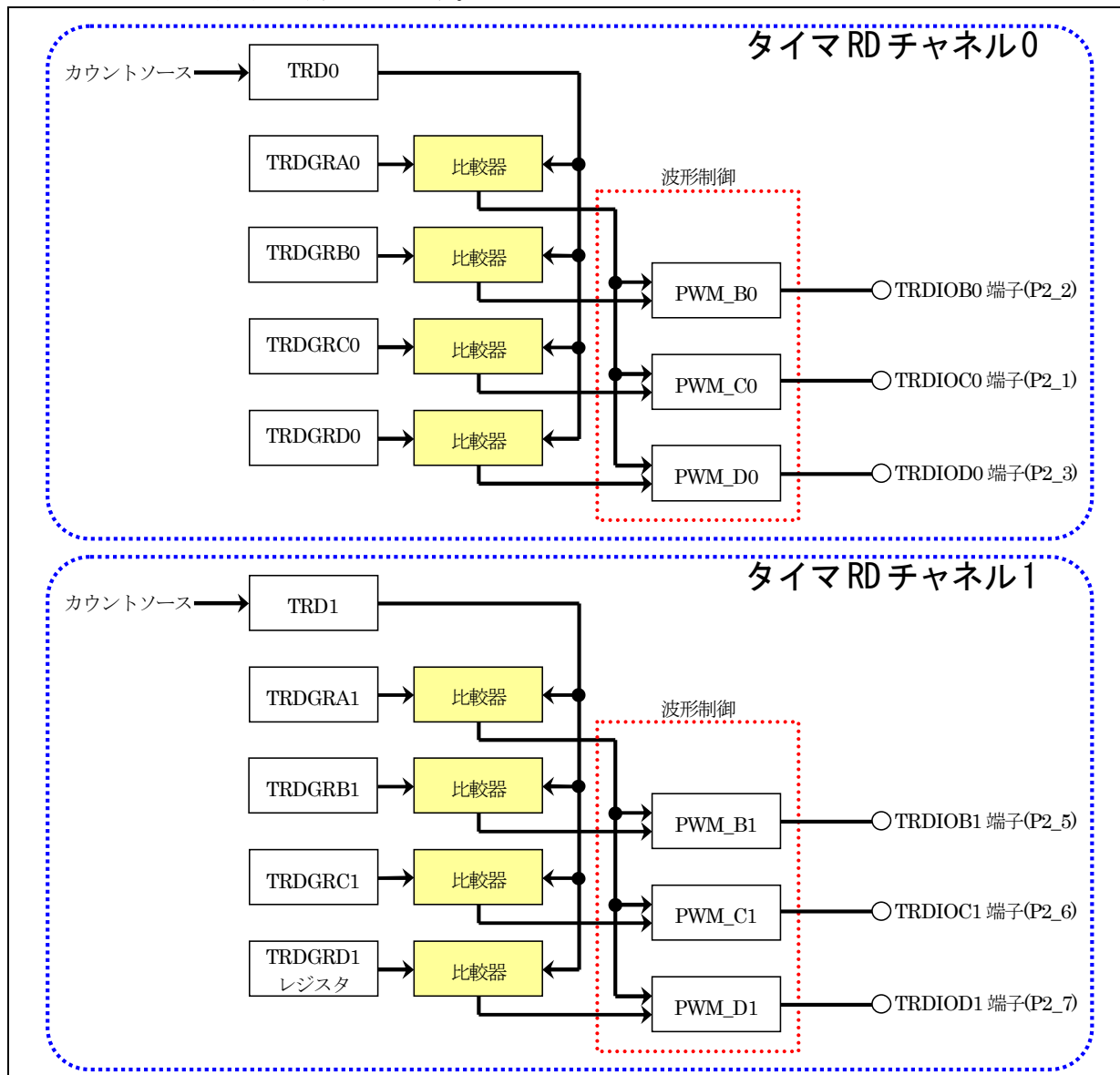
```

88 :      /* タイマRD PWMモードの設定(6本のPWM出力) */
89 :      trdpmr  = 0x77;          /* 各端子PWMモードにするかしないか */
90 :      trdfcr  = 0x80;          /* PWMモードに設定 */
91 :      trdoerl  = 0x11;         /* 各端子出力許可するかしないか */
92 :      trdpsr0  = 0x68;         /* TRDIOB0, C0, D0端子設定 */
93 :      trdpsr1  = 0x54;         /* TRDIOB1, C1, D1端子設定 */
94 :      trdocr   = 0x00;         /* 各端子初期出力設定 */
95 :      trdcr0   = 0x23;         /* TRD0ソースカウンタの選択:f8 */
96 :      trdcr1   = 0x23;         /* TRD1ソースカウンタの選択:f8 */
97 :      trdpocr0  = 0x00;         /* B0~D0のアクティブレベル設定 */
98 :      trdpocr1  = 0x00;         /* B1~D1のアクティブレベル設定 */
99 :      trdgra0  = 39999;        /* TRD0の周期設定 */
100 :      trdgra1  = 39999;        /* TRD1の周期設定 */
101 :      trdgrb0  = 0;            /* P2_2のON幅設定 */
102 :      trdgrc0  = 0;            /* P2_1のON幅設定 */
103 :      trdgrd0  = 0;            /* P2_3のON幅設定 */
104 :      trdgrb1  = 0;            /* P2_5のON幅設定 */
105 :      trdgrc1  = 0;            /* P2_6のON幅設定 */
106 :      trdgrd1  = 0;            /* P2_7のON幅設定 */
107 :      trdstr   = 0x0f;         /* TRD0, TRD1カウンタ開始 */

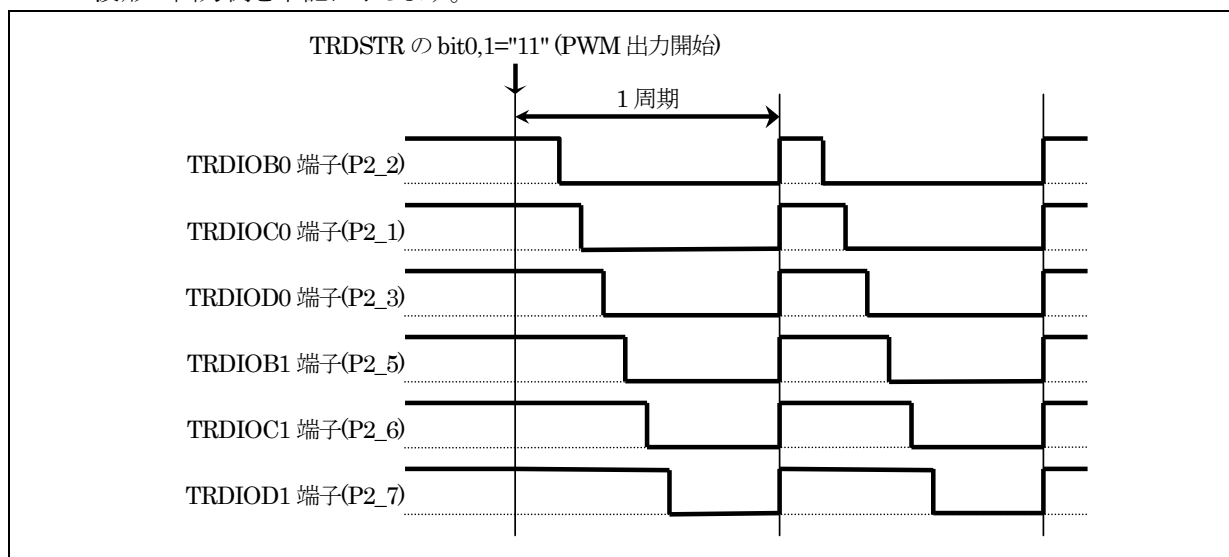
```

(1) タイマRDのブロック図

PWM モードのブロック図を下記に示します。



PWM 波形の出力例を下記に示します。



(2) タイマRDの設定(PWMモード)

今回は、タイマ RD を PWM モードで使用して、PWM 波形を 6 本出力します。レジスタの設定手順を下記に示します。





①タイマ RD PWM モードレジスタ (TRDPMR: Timer RD PWM mode register) の設定

タイマ RD を PWM モードで動作するようにします。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7		"0"を設定	0
bit6	TRDIOD1 PWM モード選択 ビット pwmd1_trdpmr	0:タイマモード 1:PWM モード TRDIOD1 を PWM として使うか、タイマとして使うかを 設定します。PWM モードを設定します。	1
bit5	TRDIOC1 PWM モード選択 ビット pwmc1_trdpmr	0:タイマモード 1:PWM モード TRDIOC1 を PWM として使うか、タイマとして使うかを 設定します。PWM モードを設定します。	1
bit4	TRDIOB1 PWMモード選択ビ ット pwmb1_trdpmr	0:タイマモード 1:PWM モード TRDIOB1をPWMとして使うか、タイマとして使うかを設 定します。PWM モードを設定します。	1
bit3		"0"を設定	0
bit2	TRDIOD0 PWM モード選択 ビット pwmd0_trdpmr	0:タイマモード 1:PWM モード TRDIOD0 を PWM として使うか、タイマとして使うかを 設定します。PWM モードを設定します。	1
bit1	TRDIOC0 PWM モード選択 ビット pwmc0_trdpmr	0:タイマモード 1:PWM モード TRDIOC0 を PWM として使うか、タイマとして使うかを 設定します。PWM モードを設定します。	1
bit0	TRDIOB0 PWMモード選択ビ ット pwmb0_trdpmr	0:タイマモード 1:PWM モード TRDIOB0をPWMとして使うか、タイマとして使うかを設 定します。PWM モードを設定します。	1

タイマ RD PWM モードレジスタ (TRDPMR) の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	1	1	1	0	1	1	1
16 進数	7				7			

②タイマ RD 機能制御レジスタ (TRDFCR:Timer RD function control register)の設定

タイマ RD を PWM モードで動作するようにします。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	PWM3 モード選択ビット pwm3_trdfer	"1"を設定 ※"1"は PWM3 モードが無効になる設定です。	1
bit6	外部クロック入力選択ビット stclk_trdfer	0:外部クロック入力無効 1:外部クロック入力有効 今回は、内部のクロックを使用しますので、無効を選択します。	0
bit5～2		"0000"を設定	0000
bit1,0	コンビネーションモード選択 ビット bit1:cmd1_trdfer bit0:cmd0_trdfer	PWM モードでは"00"(タイマモード、PWM モード、 PWM3 モード)にしてください	00

タイマ RD 機能制御レジスタ (TRDFCR)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	1	0	0	0	0	0	0	0
16 進数	8				0			

③タイマ RD アウトプットマスタ許可レジスタ 1(TRDOER1:Timer RD output master enable register 1)の設定

PWM モードは 6 端子から PWM 波形を出力することができますが、出力するか、通常の I/O ポートとして使用するかを選択します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	TRDIOD1(P2_7)出力禁止ビット ed1_trdoer1	0:出力許可 1:出力禁止(TRDIOD1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit6	TRDIOC1(P2_6)出力禁止ビット ec1_trdoer1	0:出力許可 1:出力禁止(TRDIOC1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit5	TRDIOB1(P2_5)出力禁止ビット eb1_trdoer1	0:出力許可 1:出力禁止(TRDIOB1 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit4	TRDIOA1 出力禁止ビット ea1_trdoer1	PWM モード時、“1”に設定	1
bit3	TRDIOD0(P2_3)出力禁止ビット ed0_trdoer1	0:出力許可 1:出力禁止(TRDIOD0 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit2	TRDIOC0(P2_1)出力禁止ビット ec0_trdoer1	0:出力許可 1:出力禁止(TRDIOC0 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit1	TRDIOB0(P2_2)出力禁止ビット eb0_trdoer1	0:出力許可 1:出力禁止(TRDIOB0 端子はプログラマブル入出力ポート) 今回は、出力を許可します。	0
bit0	TRDIOA0 出力禁止ビット ea0_trdoer1	PWM モード時、“1”に設定	1

タイマ RD アウトプットマスタ許可レジスタ 1(TRDOER1)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	1	0	0	0	1
16 進数	1				1			

④タイマ RD 端子選択レジスタ 0(TRDPSR0:Timer RD function select register 0)の設定

PWM 波形の出力端子をどのポートに割り当てるか設定します。タイマ RD 端子選択レジスタ 0(TRDPSR0)では、TRDIOD0 端子、TRDIOC0 端子、TRDIOB0 端子の割り当てを設定します。

R8C/35A では、割り当てる端子は決まっており端子を変更することができません。PWM 端子として割り当てるか、割り当てないか(通常の I/O ポートとして使用するか)を設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7		"0"を設定	0
bit6	TRDIOD0 端子選択ビット trdiod0sel0	0:TRDIOD0 端子は使用しない 1:P2_3 に割り当てる 今回は、P2_3 に割り当てて、この端子から PWM 波形を出力します。	1
bit5,4	TRDIOC0 端子選択ビット bit5:trdioc0sel1 bit4:trdioc0sel0	00:TRDIOC0 端子は使用しない 01:設定しないでください 10:P2_1 に割り当てる 11:設定しないでください 今回は、P2_1 に割り当てて、この端子から PWM 波形を出力します。	10
bit3,2	TRDIOB0 端子選択ビット bit3:trdiob0sel1 bit2:trdiob0sel0	00:TRDIOB0 端子は使用しない 01:設定しないでください 10:P2_2 に割り当てる 11:設定しないでください 今回は、P2_2 に割り当てて、この端子から PWM 波形を出力します。	10
bit1,0		"00"を設定	00

タイマ RD 端子選択レジスタ 0(TRDPSR0)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	1	1	0	1	0	0	0
16 進数	6				8			

⑤タイマ RD 端子選択レジスタ 1(TRDPSR1:Timer RD function select register 1)の設定

PWM 波形の出力端子をどのポートに割り当てるか設定します。タイマ RD 端子選択レジスタ 1(TRDPSR1)では、TRDIOD1 端子、TRDIOC1 端子、TRDIOB1 端子、TRDIOA1 端子の割り当てを設定します。

R8C/35A では、割り当てる端子は決まっており端子を変更することができません。PWM 端子として割り当てるか、割り当てないか(通常の I/O ポートとして使用するか)を設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7		"0"を設定	0
bit6	TRDIOD1 端子選択ビット trdioid1sel0	0:TRDIOD1 端子は使用しない 1:P2_7 に割り当てる 今回は、P2_7に割り当てて、この端子からPWM 波形を出力します。	1
bit5		0を設定	0
bit4	TRDIOC1 端子選択ビット trdioc1sel0	0:TRDIOC1 端子は使用しない 1:P2_6 に割り当てる 今回は、P2_6に割り当てて、この端子からPWM 波形を出力します。	1
bit3		"0"を設定	0
bit2	TRDIOB1 端子選択ビット trdiob1sel0	0:TRDIOB1 端子は使用しない 1:P2_5 に割り当てる 今回は、P2_5に割り当てて、この端子からPWM 波形を出力します。	1
bit1		"0"を設定	0
bit0	TRDIOA1 端子選択ビット trdioa1sel0	PWM モードでは"0"を設定	0

タイマ RD 端子選択レジスタ 1(TRDPSR1)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	1	0	1	0	1	0	0
16 進数	5				4			

⑥タイマ RD アウトプット制御レジスタ 0(TRDOCR:Timer RD output control register)の設定

PWM 出力端子の初期出力を設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	TRDIOD1 初期出力レベル選 択ビット(注 1) tod1_trdocr	0:アクティブでないレベル 1:アクティブレベル 初期出力はアクティブでないレベルにします。	0
bit6	TRDIOC1 初期出力レベル選 択ビット(注 1) toc1_trdocr	0:アクティブでないレベル 1:アクティブレベル 初期出力はアクティブでないレベルにします。	0
bit5	TRDIOB1 初期出力レベル選 択ビット(注 1) tob1_trdocr	0:アクティブでないレベル 1:アクティブレベル 初期出力はアクティブでないレベルにします。	0
bit4	TRDIOA1 初期出力レベル選 択ビット(注 1) toa1_trdocr	PWM モードでは"0"を設定	0
bit3	TRDIOD0 初期出力レベル選 択ビット(注 1) tod0_trdocr	0:アクティブでないレベル 1:アクティブレベル 初期出力はアクティブでないレベルにします。	0
bit2	TRDIOC0 初期出力レベル選 択ビット(注 1) toc0_trdocr	0:アクティブでないレベル 1:アクティブレベル 初期出力はアクティブでないレベルにします。	0
bit1	TRDIOB0 初期出力レベル選 択ビット(注 1) tob0_trdocr	0:アクティブでないレベル 1:アクティブレベル 初期出力はアクティブでないレベルにします。	0
bit0	TRDIOA0 初期出力レベル選 択ビット(注 1) toa0_trdocr	PWM モードでは"0"を設定	0

注 1. 端子の機能が波形出力の場合(「ハードウェアマニュアル 7.5 ポートの設定」参照)、TRDOCR レジスタを設定したとき、初期出力レベルが出力されます。

タイマ RD 制御レジスタ 0(TRDOCR)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	0	0	0	0
16 進数	0				0			

⑦タイマ RD 制御レジスタ 0(TRDCR0:Timer RD control register 0)の設定

タイマ RD カウンタ 0 (TRD0) がカウントアップする時間などを選択します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～5	TRD0 カウンタクリア選択ビット bit7:cclr2_trdcr0 bit6:cclr1_trdcr0 bit5:cclr0_trdcr0	PWM モードの場合は、“001”(TRDGRA0 とのコンペア一致で TRD0 レジスタクリア)に設定	001
bit4,3	外部クロックエッジ選択ビット (注 3) bit4:ckeg1_trdcr0 bit3:ckeg0_trdcr0	00:立ち上がりエッジでカウント 01:立ち下がりエッジでカウント 10:両エッジでカウント 11:設定しないでください 外部クロックは使いませんので何を設定しても変化ありません。今回は“00”を設定します。	00
bit2～0	カウントソース選択ビット bit2:tck2_trdcr0 bit1:tck1_trdcr0 bit0:tck0_trdcr0	000:f1 (1/20MHz=50ns) 001:f2 (2/20MHz=100ns) 010:f4 (4/20MHz=200ns) 011:f8 (8/20MHz=400ns) 100:f32 (32/20MHz=1600ns) 101:TRDCLK 入力(注 1)または fC2 (注 2) fC2 = 2/XCIN クロック=今回は未接続 110:fOCO40M (高速オンチップオシレータ 40MHz=今回は未接続) 111:fOCO-F(注 4) (高速オンチップオシレータを FRA2 で分周したクロック=今回は未接続) タイマ RD カウンタ 0 (TRD0) がカウントアップする時間を設定します。今回は“011”を設定します。TRD0 は、400ns ごとに+1 していきます。	011

注 1. TRDECR レジスタの ITCLK0 ビットが“0”(TRDCLK 入力)かつ TRDFCR レジスタの STCLK ビットが“1”(外部クロック入力有効)のとき、有効です。

注 2. タイマモードで、TRDECR レジスタの ITCLK0 ビットが“1”(fC2)のとき有効です。

注 3. TCK2～TCK0 ビットが“101”(TRDCLK 入力または fC2)、TRDECR レジスタの ITCLK0 ビットが“0”(TRDCLK 入力)、かつ TRDFCR レジスタの STCLK ビットが“1”(外部クロック入力有効)のとき、有効です。

注 4. fOCO-F を選択するとき、CPU クロックより速いクロック周波数に fOCO-F を設定してください。

タイマ RD 制御レジスタ 0(TRDCR0)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	1	0	0	0	1	1
16 進数	2				3			

⑧タイマ RD 制御レジスタ 1(TRDCR1:Timer RD control register 1)の設定

タイマ RD カウンタ 1 (TRD1) がカウントアップする時間などを選択します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～5	TRD1 カウンタクリア選択ビット bit7:cclr2_trdcr1 bit6:cclr1_trdcr1 bit5:cclr0_trdcr1	PWM モードの場合は、“001”(TRDGRA1 とのコンペア一致で TRD1 レジスタクリア)に設定	001
bit4,3	外部クロックエッジ選択ビット (注 3) bit4:ckeg1_trdcr1 bit3:ckeg0_trdcr1	00:立ち上がりエッジでカウント 01:立ち下がりエッジでカウント 10:両エッジでカウント 11:設定しないでください 外部クロックは使いませんので何を設定しても変化ありません。今回は“00”を設定します。	00
bit2～0	カウントソース選択ビット bit2:tck2_trdcr1 bit1:tck1_trdcr1 bit0:tck0_trdcr1	000:f1 (1/20MHz=50ns) 001:f2 (2/20MHz=100ns) 010:f4 (4/20MHz=200ns) 011:f8 (8/20MHz=400ns) 100:f32 (32/20MHz=1600ns) 101:TRDCLK 入力(注 1)または fC2 (注 2) fC2 = 2/XCIN クロック=今回は未接続 110:fOCO40M (高速オンチップオシレータ 40MHz=今回は未接続) 111:fOCO-F(注 4) (高速オンチップオシレータを FRA2 で分周したクロック=今回は未接続) タイマ RD カウンタ 1 (TRD1) がカウントアップする時間を設定します。今回は“011”を設定します。TRD1 は、400ns ごとに+1 していきます。	011

注 1. TRDECR レジスタの ITCLK1 ビットが“0”(TRDCLK 入力)かつ TRDFCR レジスタの STCLK ビットが“1”(外部クロック入力有効)のとき、有効です。

注 2. タイマモードで、TRDECR レジスタの ITCLK1 ビットが“1”(fC2)のとき有効です。

注 3. TCK2～TCK0 ビットが“101”(TRDCLK 入力または fC2)、TRDECR レジスタの ITCLK1 ビットが“0”(TRDCLK 入力)、かつ TRDFCR レジスタの STCLK ビットが“1”(外部クロック入力有効)のとき、有効です。

注 4. fOCO-F を選択するとき、CPU クロックより速いクロック周波数に fOCO-F を設定してください。

タイマ RD 制御レジスタ 1(TRDCR1)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	1	0	0	0	1	1
16 進数	2				3			

⑨タイマ RD PWM モードアウトプットレベル制御レジスタ 0

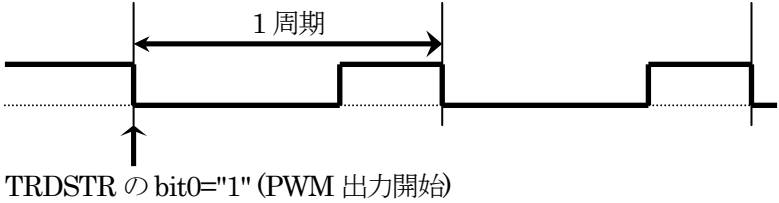
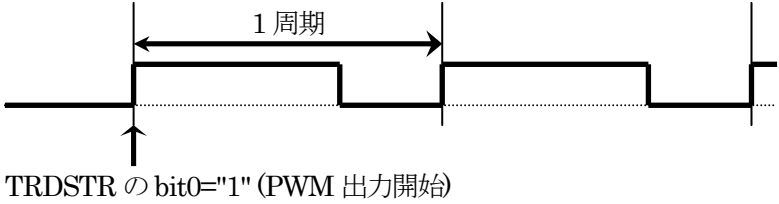
(TRDPOCR0: Timer RD PWM mode output level control register 0)の設定

タイマ RD PWM モードアウトプットレベル制御レジスタ 0(TRDPOCR0)の設定をします。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～3		"00000"を設定	00000
bit2	PWM モードアウトプットレベル制御ビット D pold_trdpocr0	0:TRDIOD0 の出力レベルは“L”アクティブ 1:TRDIOD0 の出力レベルは“H”アクティブ 出力レベルは“L”アクティブに設定	0
bit1	PWM モードアウトプットレベル制御ビット C polc_trdpocr0	0:TRDIOC0 の出力レベルは“L”アクティブ 1:TRDIOC0 の出力レベルは“H”アクティブ 出力レベルは“L”アクティブに設定	0
bit0	PWM モードアウトプットレベル制御ビット B polb_trdpocr0	0:TRDIOB0 の出力レベルは“L”アクティブ 1:TRDIOB0 の出力レベルは“H”アクティブ 出力レベルは“L”アクティブに設定	0

※出力レベルと出力波形の関係

出力レベルによって、下記のように出力波形のレベルが変わります。

出力 レベル	波形	説明
“H”アク ティブ		“0”からスタートする設定です。
“L”アク ティブ		“1”からスタートする設定です。

タイマ RD PWM モードアウトプットレベル制御レジスタ 0(TRDPOCR0)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	0	0	0	0
16 進数	0				0			

⑩タイマ RD PWM モードアウトプットレベル制御レジスタ 1

(TRDPOCR1: Timer RD PWM mode output level control register 1)の設定

タイマ RD PWM モードアウトプットレベル制御レジスタ 1(TRDPOCR1)の設定をします。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～3		"00000"を設定	00000
bit2	PWM モードアウトプットレベル制御ビット D pold_trdpocr1	0:TRDIOD1 の出力レベルは“L”アクティブ 1:TRDIOD1 の出力レベルは“H”アクティブ 出力レベルは“L”アクティブに設定	0
bit1	PWM モードアウトプットレベル制御ビット C polc_trdpocr1	0:TRDIOC1 の出力レベルは“L”アクティブ 1:TRDIOC1 の出力レベルは“H”アクティブ 出力レベルは“L”アクティブに設定	0
bit0	PWM モードアウトプットレベル制御ビット B polb_trdpocr1	0:TRDIOB1 の出力レベルは“L”アクティブ 1:TRDIOB1 の出力レベルは“H”アクティブ 出力レベルは“L”アクティブに設定	0

※出力レベルと出力波形の関係

出力レベルによって、下記のように出力波形のレベルが変わります。

出力 レベル	波形	説明
“H”アク ティブ		“0”からスタートする設定です。
“L”アク ティブ		“1”からスタートする設定です。

タイマ RD PWM モードアウトプットレベル制御レジスタ 1(TRDPOCR1)の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	0	0	0	0
16 進数	0				0			

⑪タイマ RD ジェネラルレジスタ A0(TRDGRA0:Timer RD General register A0)の設定

タイマ RD ジェネラルレジスタ A0(TRDGRA0)に値を設定することによって、TRDIOB0(P2_2) 端子,TRDIOC0(P2_1)端子,TRDIOD0(P2_3)端子から出力する PWM 波形の周期を設定します。PWM 周期は、下記の式で決まります。

$$\text{PWM 周期} = \text{タイマ RD カウンタ 0 のカウントソース} \times (\text{TRDGRA0} + 1)$$

TRDGRA0 を左辺に移動して、TRDGRA0 を求める式に変形します。

$$\text{TRDGRA0} = \text{PWM 周期} / \text{タイマ RD カウンタ 0 のカウントソース} - 1$$

今回、PWM 波形の周期を 16ms にします。タイマ RD カウンタ 0 のカウントソースとは、タイマ RD 制御レジスタ 0(TRDCR0)のカウントソース選択ビット(bit2~0)で設定した時間のことで、今回は 400ns に設定しています。よって、タイマ RD ジェネラルレジスタ A0(TRDGRA0)は次のようになります。

$$\begin{aligned}\text{TRDGRA0} &= \text{周期} / \text{カウントソース} - 1 \\ \text{TRDGRA0} &= (16 \times 10^{-3}) / (400 \times 10^{-9}) - 1 \\ \text{TRDGRA0} &= 40000 - 1 = 39999\end{aligned}$$

TRDGRA0 の値は、65,535 以下にする必要があります。今回の結果は、65,535 以下なので、400ns の設定で大丈夫です。65,536 以上になった場合、タイマ RD 制御レジスタ 0(TRDCR0)のカウントソース選択ビット(bit2~0)の設定を大きい時間にしてください。

⑫タイマ RD ジェネラルレジスタ A1(TRDGRA1:Timer RD General register A1)の設定

タイマ RD ジェネラルレジスタ A1(TRDGRA1)に値を設定することによって、TRDIOB1(P2_5) 端子,TRDIOC1(P2_6)端子,TRDIOD1(P2_7)端子から出力する PWM 波形の周期を設定します。PWM 周期は、下記の式で決まります。

$$\text{PWM 周期} = \text{タイマ RD カウンタ 1 のカウントソース} \times (\text{TRDGRA1} + 1)$$

TRDGRA1 を左辺に移動して、TRDGRA1 を求める式に変形します。

$$\text{TRDGRA1} = \text{PWM 周期} / \text{タイマ RD カウンタ 1 のカウントソース} - 1$$

今回、PWM 波形の周期を 16ms にします。タイマ RD カウンタ 1 のカウントソースとは、タイマ RD 制御レジスタ 1(TRDCR1)のカウントソース選択ビット(bit2~0)で設定した時間のことで、今回は 400ns に設定しています。よって、タイマ RD ジェネラルレジスタ A1(TRDGRA1)は次のようになります。

$$\begin{aligned}\text{TRDGRA1} &= \text{PWM 周期} / \text{カウントソース} - 1 \\ \text{TRDGRA1} &= (16 \times 10^{-3}) / (400 \times 10^{-9}) - 1 \\ \text{TRDGRA1} &= 40000 - 1 = 39999\end{aligned}$$

TRDGRA1 の値は、65,535 以下にする必要があります。今回の結果は、65,535 以下なので、400ns の設定で大丈夫です。65,536 以上になった場合、タイマ RD 制御レジスタ 1(TRDCR1)のカウントソース選択ビット(bit2~0)の設定を大きい時間にしてください。

- ⑬タイマ RD ジェネラルレジスタ B0(TRDGRB0:Timer RD General register B0)の設定
 ⑭タイマ RD ジェネラルレジスタ C0(TRDGRC0:Timer RD General register C0)の設定
 ⑮タイマ RD ジェネラルレジスタ D0(TRDGRD0:Timer RD General register D0)の設定
 ⑯タイマ RD ジェネラルレジスタ B1(TRDGRB1:Timer RD General register B1)の設定
 ⑰タイマ RD ジェネラルレジスタ C1(TRDGRC1:Timer RD General register C1)の設定
 ⑱タイマ RD ジェネラルレジスタ D1(TRDGRD1:Timer RD General register D1)の設定

各端子から PWM 波形を出力するときの、出力波形の ON 幅を設定します。

番号	レジスタ名	レジスタ名 略称	出力 端子	ON 幅の計算
⑬	タイマ RD ジェネラル レジスタ B0	TRDGRB0	P2_2	$\text{TRDGRB0} = \text{ON 幅} \times \text{タイマ RD カウンタ 0 のカウントソース} - 1$
⑭	タイマ RD ジェネラル レジスタ C0	TRDGRC0	P2_1	$\text{TRDGRC0} = \text{ON 幅} \times \text{タイマ RD カウンタ 0 のカウントソース} - 1$
⑮	タイマ RD ジェネラル レジスタ D0	TRDGRD0	P2_3	$\text{TRDGRD0} = \text{ON 幅} \times \text{タイマ RD カウンタ 0 のカウントソース} - 1$
⑯	タイマ RD ジェネラル レジスタ B1	TRDGRB1	P2_5	$\text{TRDGRB1} = \text{ON 幅} \times \text{タイマ RD カウンタ 1 のカウントソース} - 1$
⑰	タイマ RD ジェネラル レジスタ C1	TRDGRC1	P2_6	$\text{TRDGRC1} = \text{ON 幅} \times \text{タイマ RD カウンタ 1 のカウントソース} - 1$
⑱	タイマ RD ジェネラル レジスタ D1	TRDGRD1	P2_7	$\text{TRDGRD1} = \text{ON 幅} \times \text{タイマ RD カウンタ 1 のカウントソース} - 1$

例として、タイマ RD ジェネラルレジスタ B0(TRDGRB0)を計算してみます。TRDGRB0 を設定することにより、P2_2 端子から出力される PWM 波形の ON 幅を設定します。

P2_2 端子から出力される PWM 波形の ON 幅は、下記の式で決まります。

$$\text{TRDGRB0} = \text{P2_2 端子から出力される PWM 波形の ON 幅} / \text{タイマ RD カウンタ 0 のカウントソース} - 1$$

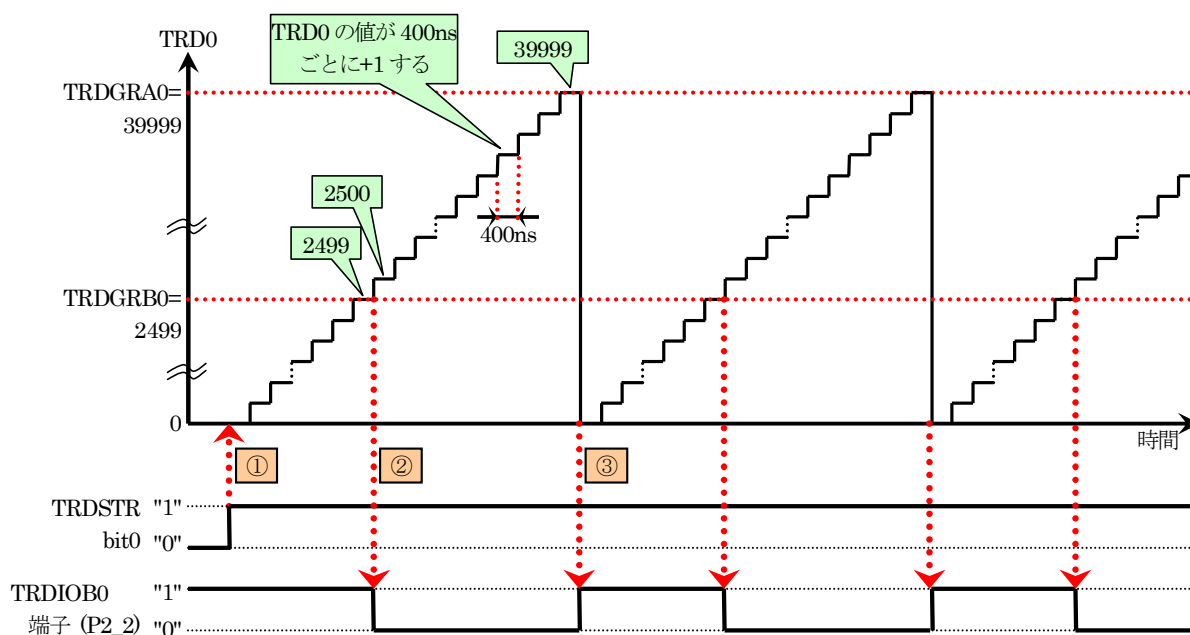
今回、タイマ RD カウンタ 0 のカウントソースは 400ns です。例えば ON 幅を 1ms にするなら、タイマ RD ジェネラルレジスタ B0(TRDGRB0)の値は次のようになります。

$$\text{TRDGRB0} = \text{ON 幅} / \text{カウントソース} - 1$$

$$\text{TRDGRB0} = (1 \times 10^{-3}) / (400 \times 10^{-9}) - 1$$

$$\text{TRDGRB0} = 2500 - 1 = 2499$$

TRD0、TRDGRA0、TRDGRB0 の値と波形の関係を図解すると下記ようになります。



①	TRDSTR の bit0 を"1"にすると、TRD0 のカウントが開始されます。
②	<p>TRD0=(TRDGRB0+1)になると、P2_2 端子が"0"になります。 TRDGRB0 の値は 2499、TRD0 が+1 する間隔は 400ns です。 よって、P2_2 端子が"1"の時間は、 P2_2 端子が"1"の時間 = (TRDGRB0+1)×TRD0 が+1 する間隔 = (2499+1)×400ns = 1,000,000 [ns] = 1[ms]</p>
③	<p>TRD0=(TRDGRA0+1)になると、P2_2 端子が"1"になります。同時に TRD0 が 0 にクリアされ、また 0 から値が増えていきます。 TRDGRA0 の値は 39999、TRD0 が+1 する間隔は 400ns です。 よって、P2_2 端子の PWM 周期は、 P2_2 端子の PWM 周期 = (TRDGRA0+1)×TRD0 が+1 する間隔 = (39999+1)×400ns = 16,000,000 [ns] = 16[ms]</p> <p>ちなみに、 「PWM 周期」=「ON("1")の時間」+「OFF("0")の時間」 ですので、"0"の時間は、 「OFF("0")の時間」=「PWM 周期」-「ON("1")の時間」 = 16[ms] - 1[ms] = 15[ms] となります。</p>

⑩タイマ RD スタートレジスタ (TRDSTR: Timer RD start register) の設定

TRD0 をカウントさせるか、停止させるか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～4		"0000"を設定	0000
bit3	TRD1 カウント動作選択ビット csel1_trdstr	0: TRDGRA1 レジスタとのコンペア一致でカウント停止 1: TRDGRA1 レジスタとのコンペア一致後もカウント継続 "1"を設定します。	1
bit2	TRD0 カウント動作選択ビット csel0_trdstr	0: TRDGRA0 レジスタとのコンペア一致でカウント停止 1: TRDGRA0 レジスタとのコンペア一致後もカウント継続 "1"を設定します。	1
bit1	TRD1 カウント開始フラグ(注 4) tstart1_trdstr	0: カウント停止(注 2) 1: カウント開始 設定した瞬間から、TRD1 のカウントが開始されます。 "1"を設定します。	1
bit0	TRD0 カウント開始フラグ(注 3) tstart0_trdstr	0: カウント停止(注 1) 1: カウント開始 設定した瞬間から、TRD0 のカウントが開始されます。 "1"を設定します。	1

注 1. bit2 が“1”に設定されているとき、bit0 へ“0”を書いてください。

注 2. bit3 が“1”に設定されているとき、bit1 へ“0”を書いてください。

注 3. bit2 が“0”でコンペア一致信号(TRDIOA0)が発生したとき、“0”(カウント停止)になります。

注 4. bit3 が“0”でコンペア一致信号(TRDIOA1)が発生したとき、“0”(カウント停止)になります。

タイマ RD スタートレジスタ (TRDSTR) の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	1	1	1	1
16 進数	0				f			

22.5.2 main関数

マイコンボードのディップスイッチの値 0～15 に応じて、PWM 端子から 0%～100%の PWM 波形を出力します。

```

35 : void main( void )
36 : {
37 :     unsigned int    pwm;
38 :
39 :     init();          /* 初期化                      */
40 :
41 :     while( 1 ) {
42 :         if( dipsw_get() == 0 ) {
43 :             // trdgra0, trdgra1と同じにすると0%出力
44 :             pwm = 39999;
45 :         } else if( dipsw_get() == 15 ) {
46 :             // trdgra0, trdgra1より大きくすると100%出力
47 :             pwm = 40000;
48 :         } else {
49 :             // 1～14なら割合に応じてPWM出力する
50 :             pwm = (long)39999 * dipsw_get() / 15;
51 :         }
52 :
53 :         trdgrb0 = pwm;      /* P2_2のON幅設定          */
54 :         trdgrc0 = pwm;      /* P2_1のON幅設定          */
55 :         trdgrd0 = pwm;      /* P2_3のON幅設定          */
56 :         trdgrb1 = pwm;      /* P2_5のON幅設定          */
57 :         trdgrc1 = pwm;      /* P2_6のON幅設定          */
58 :         trdgrd1 = pwm;      /* P2_7のON幅設定          */
59 :     }
60 : }
```

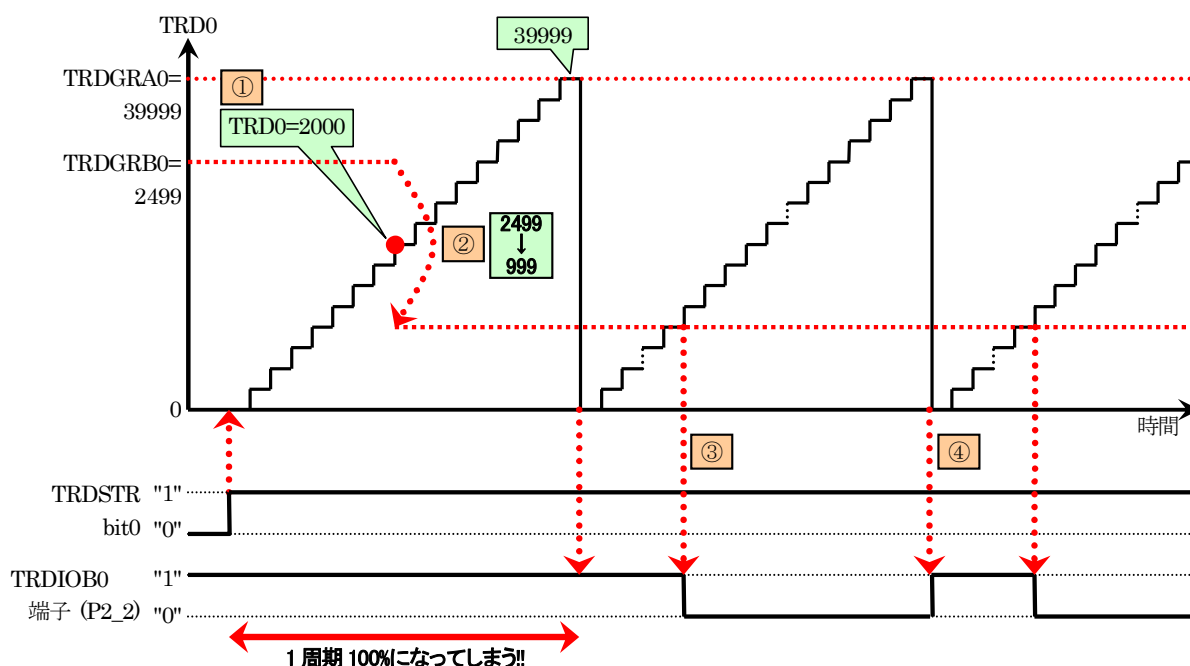
42 行	ディップスイッチの値が 0 かどうかチェックします。0 なら、44 行を実行します。
44 行	PWM を 0%出力するために、pwm 変数に TRDGRA0 (TRDGRA1)と同じ値の 39999 を設定します。周期を替えた場合は、TRDGRA0 (TRDGRA1)の値は 39999 では無くなりますので、TRDGRA0 (TRDGRA1)と同じ値を設定してください。 詳しくは、「21. タイマRDによるPWM波形出力(リセット同期PWMモード) (プロジェクト: timer_rd_doukipwm)」を参照してください。
45 行	ディップスイッチの値が 15 かどうかチェックします。15 なら、47 行を実行します。
47 行	PWM を 100%出力するために、pwm 変数に TRDGRA0 (TRDGRA1)より 1 つ大きい値の 40000 を設定します。周期を替えた場合は、TRDGRA0 (TRDGRA1)の値は 40000 では無くなりますので、TRDGRA0+1 (TRDGRA1+1)の値を設定してください。 詳しくは、「21. タイマRDによるPWM波形出力(リセット同期PWMモード) (プロジェクト: timer_rd_doukipwm)」を参照してください。
50 行	ディップスイッチの値が 1～14 の場合、ディップスイッチの割合に応じて PWM 値を計算します。
53 行～ 58 行	各ジェネラルレジスタに pwm 変数の値を設定して、PWM 波形の ON 幅を設定します。

マイコンボードのディップスイッチとデューティ比の関係を、下表に示します。

ディップスイッチの値 (dipsw_get 関数)	TRDGRD0 の値 (A)	TRD0 と一致する値 (A)+1	デューティ比 ((A)+1)/40000
0	39999 を直接指定	40000 (2 つの一致が 同時に起こり、 "0"が優先される)	0%
1	$39999 * 1 / 15 = 2666$	2667	6.6675%
2	$39999 * 2 / 15 = 5333$	5334	13.3350%
3	$39999 * 3 / 15 = 7999$	8000	20.0000%
4	$39999 * 4 / 15 = 10666$	10667	26.6675%
5	$39999 * 5 / 15 = 13332$	13333	33.3325%
6	$39999 * 6 / 15 = 15999$	16000	40.0000%
7	$39999 * 7 / 15 = 18665$	18666	46.6650%
8	$39999 * 8 / 15 = 21332$	21333	53.3325%
9	$39999 * 9 / 15 = 23998$	23999	59.9975%
10	$39999 * 10 / 15 = 26665$	26666	66.6650%
11	$39999 * 11 / 15 = 29331$	29332	73.3300%
12	$39999 * 12 / 15 = 31998$	31999	79.9975%
13	$39999 * 13 / 15 = 34664$	34665	86.6625%
14	$39999 * 14 / 15 = 37331$	37332	93.3300%
15	40000 を直接指定	一致せず	100%

※問題点

例えば、TRDGRB0=2499、TRD0=2000 のときに、プログラムで TRDGRB0 の値を 999 に変更したとします。



① ②	<p>TRD0 が 2000 のときに(①)、プログラムで TRDGRB0 の値を 2499 から 999 に変更します(②)。 TRD0 の値が 400ns ごとに増えていき 40000 になりました。40000 になると「TRD0=(TRDGRA0+1)」が成り立ち、P2_2 端子が「1」になります。同時に TRD0 が 0 にクリアされます。この間、「TRD0=(TRDGRB0+1)」が成り立つ条件がありませんでした。そのため、今回の 1 周期分すべてが「1」となり 100%出力になってしまいます。</p> <p>今回のように、「TRD0=(TRDGRB0+1)」が成り立つ前に TRDGRB0 の値を TRD0 より小さい値に変更すると、PWM100%という想定外の値になります。</p>
③	<p>TRD0=(TRDGRB0+1)になるため、P2_2 端子が「0」になります。</p>
④	<p>TRD0=(TRDGRA0+1)になるため、P2_2 端子が「1」になります。</p>

このように、TRDGRB0 の値を変えるタイミングによっては PWM100%という想定外の波形になってしまいます。モータを制御していたら、一瞬ですが 100%の回転となってしまう、暴走してしまうかもしれません。

リセット同期 PWM モードの場合、バッファレジスタがありましたが、PWM モードにはバッファレジスタの機能がありません。そのため、プログラムでこのような不具合が起こらないようにします。

各ジェネラルレジスタを設定するとき、下記のようにプログラムします。

```
// trdgrb0を設定するとき
while( trd0 <= trdgrb0 && trdgra0 > trdgrb0 );
trdgrb0 = pwm;

// trdgrc0を設定するとき
while( trd0 <= trdgrc0 && trdgra0 > trdgrc0 );
trdgrc0 = pwm;

// trdgrd0を設定するとき
while( trd0 <= trdgrd0 && trdgra0 > trdgrd0 );
trdgrd0 = pwm;

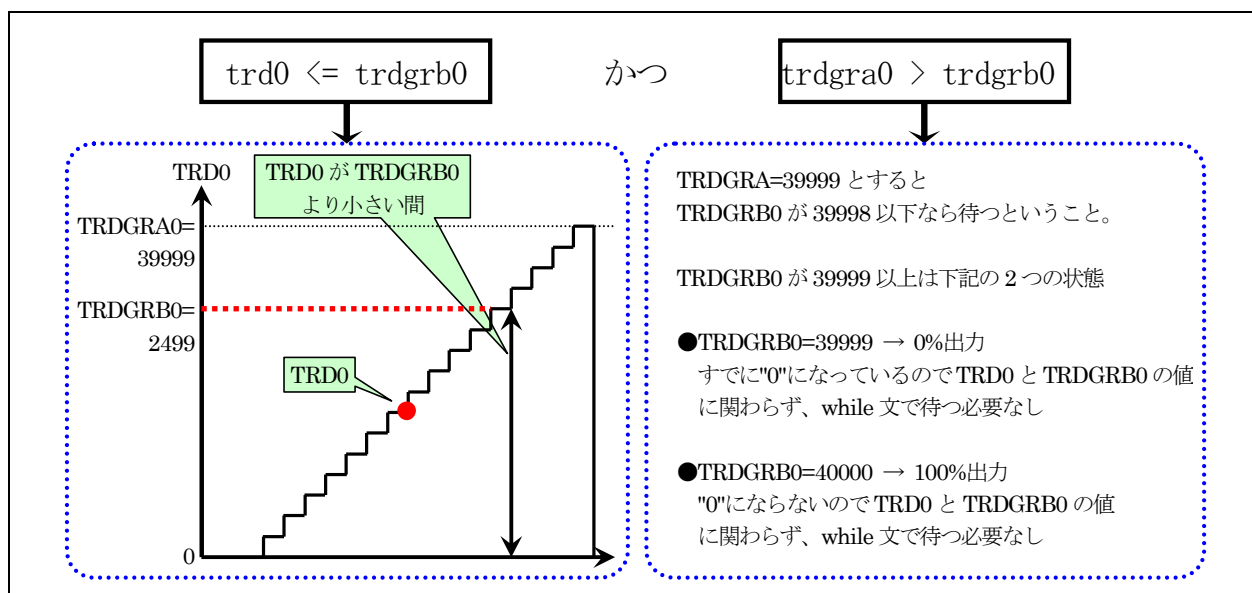
// trdgrb1を設定するとき
while( trd1 <= trdgrb1 && trdgra1 > trdgrb1 );
trdgrb1 = pwm;

// trdgrc1を設定するとき
while( trd1 <= trdgrc1 && trdgra1 > trdgrc1 );
trdgrc1 = pwm;

// trdgrd1を設定するとき
while( trd1 <= trdgrd1 && trdgra1 > trdgrd1 );
trdgrd1 = pwm;
```

不具合が発生するタイミングかどうか、while 文でチェックします。不具合が発生するタイミングの場合、while 文の行で不具合が発生しなくなるタイミングまで待ちます。そのため、待つ時間が長くなるとバッファレジスタを使うよりプログラム実行時間が長くなります。

while 文のカッコの中の意味を下記に示します。



22.6 演習

- (1) 問題点を解決したプログラムに改造しなさい。
- (2) TRDGRC0(P2_1)=0%、TRDGRB0(P2_2)=20%、TRDGRD0(P2_3)=40%、TRDGRB1=60%、TRDGRC1(P2_6)=80%、TRDGRD1(P2_7)=100%を初期 PWM 値として、1 秒ごとに 20%ずつ PWM 値を増やすプログラムを作成しなさい。ただし、100%の次は 0%とする。
- (3) (2)の初期 PWM 値から初めて、1 秒ごとに 10%ずつ PWM 値を減らすプログラムを作成しなさい。ただし、0%の次は 100%とする。