

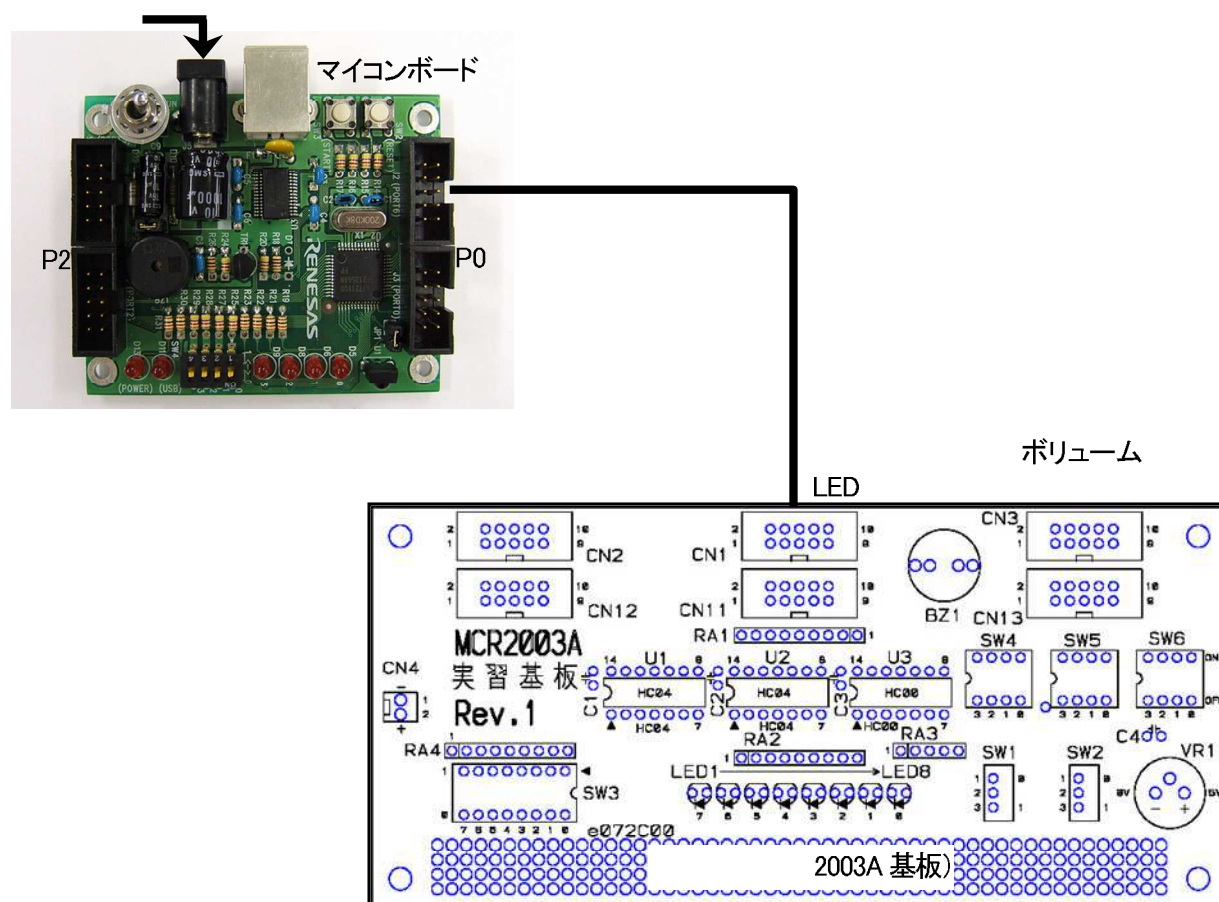
## 14.1 概要

時間を計ります。具体的には、タイマ RB で 1ms ごとに割り込みを発生させ、その回数で時間を計ります。

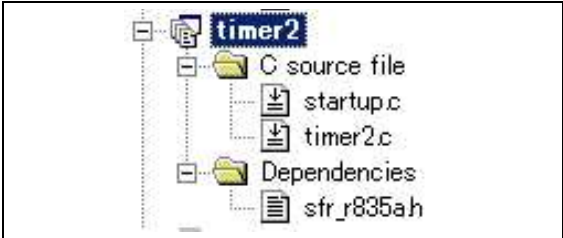
### ■使用ポート

ポート	
(J2)	

実習基板を使ったときの接続例を次に示します。



操作は特にありません。電源を入れると LED が点滅します。LED の点滅の仕方をよく観察してください。



		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。
		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
		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

```
1 :  /*****/
3 :  /* ファイル内容      タイマRB割り込みによるタイマ      */
5 :  /* Date      2010. 04. 19      */
7 :  /*      日立インターメディックス株式会社      */
9 :  /*
11 :
13 :  タイマはタイマRB割り込みによる正確なタイマを使用します。
15 :
17 :  /* インクルード      */
19 :  #include "sfr_r835a.h"      /* R8C/35A SFRの
20 :
22 :  /* シンボル定義      */
24 :
26 :  /* プロトタイプ宣言      */
28 :  void init( void );
30 :
32 :  /* グローバル変数の宣言      */
34 :  unsigned long cnt_rb;      /* タイマRB用      */
```

---

```

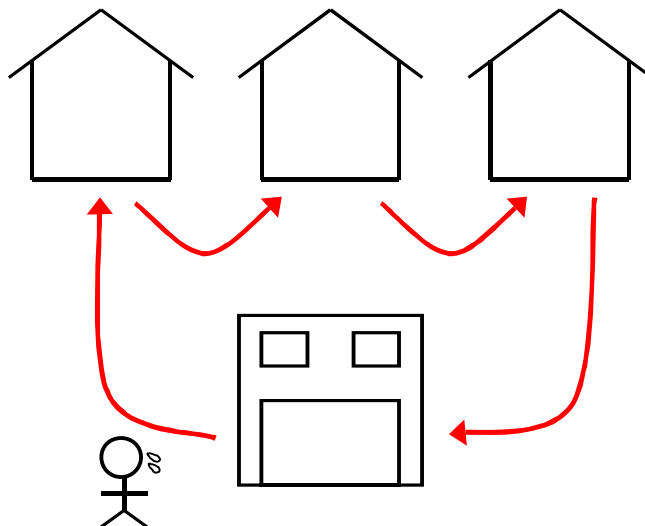
37 : /* メインプログラム */
39 : void main( void )
41 :     init(); /* 初期化 */
43 :
45 :     p6 = 0x55;
47 :     p6 = 0xaa;
49 :     p6 = 0x00;
51 : }
53 :
55 : /* R8C/35A スペシャルファンクションレジスタ (SFR) の初期化 */
57 : void init( void )
59 :     int i;
61 :     /* クロックをXINクロック (20MHz) に変更 */
63 :     cm13 = 1; /* P4_6, P4_7をXI
64 :     cm05 = 0; /* XINクロック発振 */
66 :     ocd2 = 0; /* システムクロックをXIN
67 :     prc0 = 0; /* プロテクトON */
69 :     /* ポートの入出力設定 */
71 :     pd0 = 0xe0; /* 7-5:LED 4:MicroSW 3-0:Sensor */
73 :     pd1 = 0xdf; /* 5:RXD0 4:TXD0 3-0:LED */
75 :     pd3 = 0xfb; /* 4:Buzzer 2:IR */
77 :     pd5 = 0x40; /* 7:DIP SW */
79 :
81 :     /* 割り込み周期 = 1 / 20[MHz] * (TRBPRE+1) * (TRBPR+1)
83 :     = 0.001[s] = 1[ms]
85 :     trbmr = 0x00; /* 動作モード、分周比設定 */
87 :     trbpr = 100-1; /* プライマリレジスタ */
89 :     trbcr = 0x01; /* カウント開始 */
91 :
93 : /* タイマ本体 */
95 : /*****
97 : {
99 :     while( cnt_rb < timer_set );
101 :
103 : /* タイマRB 割り込み処理 */
105 : #pragma interrupt intTRB(vect=24)
107 : {
109 : }
111 : /*****

```

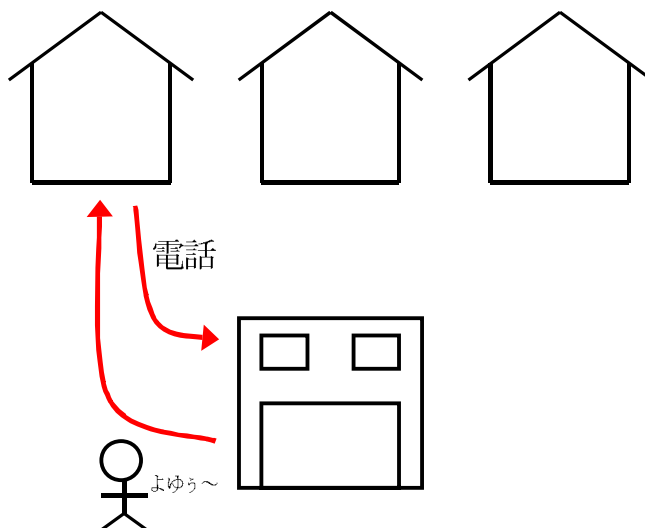
---

### 14.5.1 割り込みとは

例えば、ピザ屋さんが家 1～3 に注文がないか回るとします。バイト君は、定期的に家を回らなければいけません  
**ポリング**  
また、注文がなければ無駄足になってしまいます(下図)。



来ればその家に届けばよいので作業効率が良いです(下図)。



の受け答えをする必要があります。割り込みプログラムに当たります。

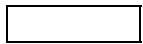
---

・注文がないか聞きに回る

視が遅れたり、監視もれが起こります。

制御の用語(でもないですが)で「割り込み」といいます。電話のように、きっかけがあったときにだけ対処すれ

人で例えましたが、マイコンの場合は下記のようになります。

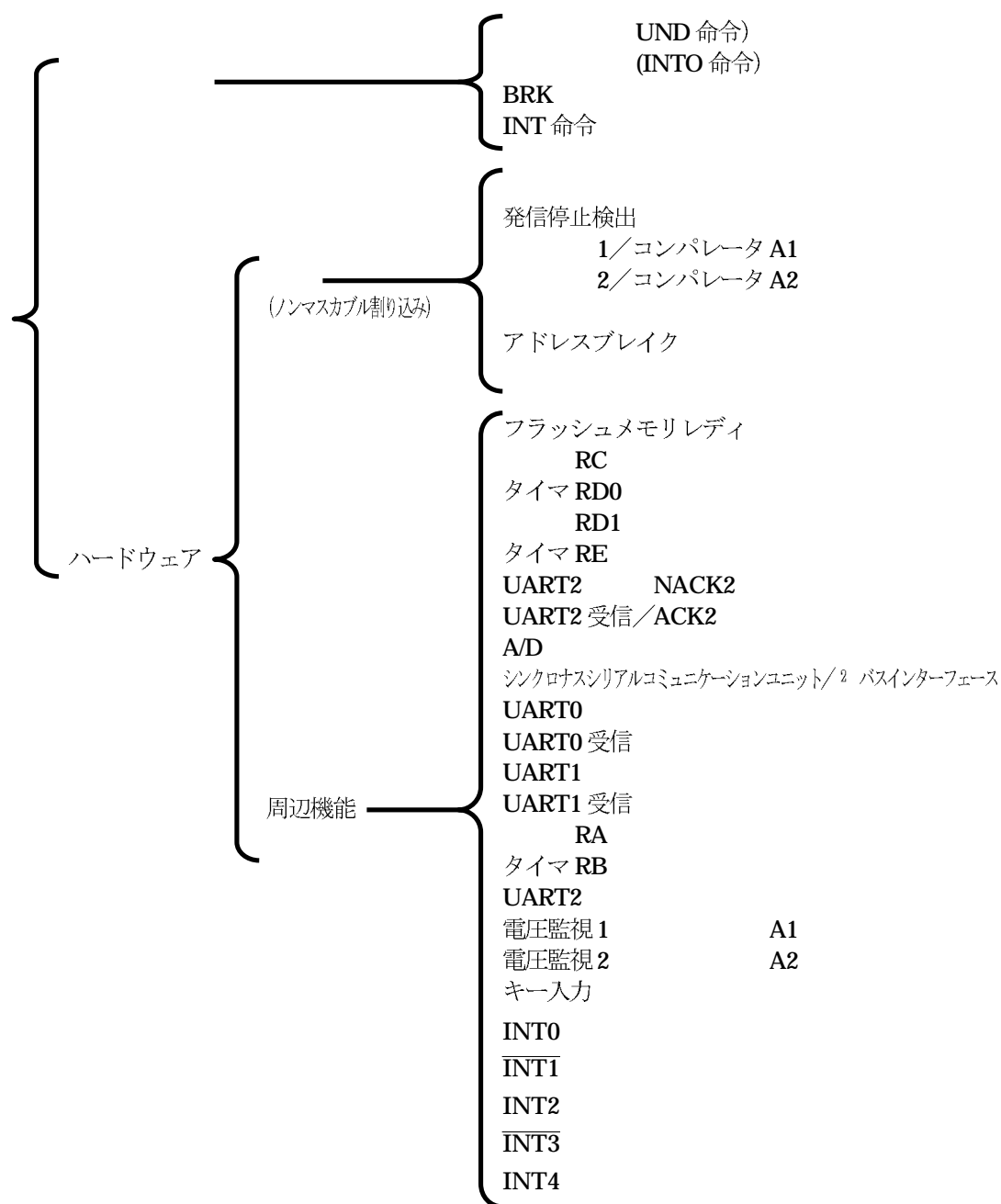


ベルが鳴る

→

割り込みが発生する

R8C/35A の割り込みの種類を、下表に示します。



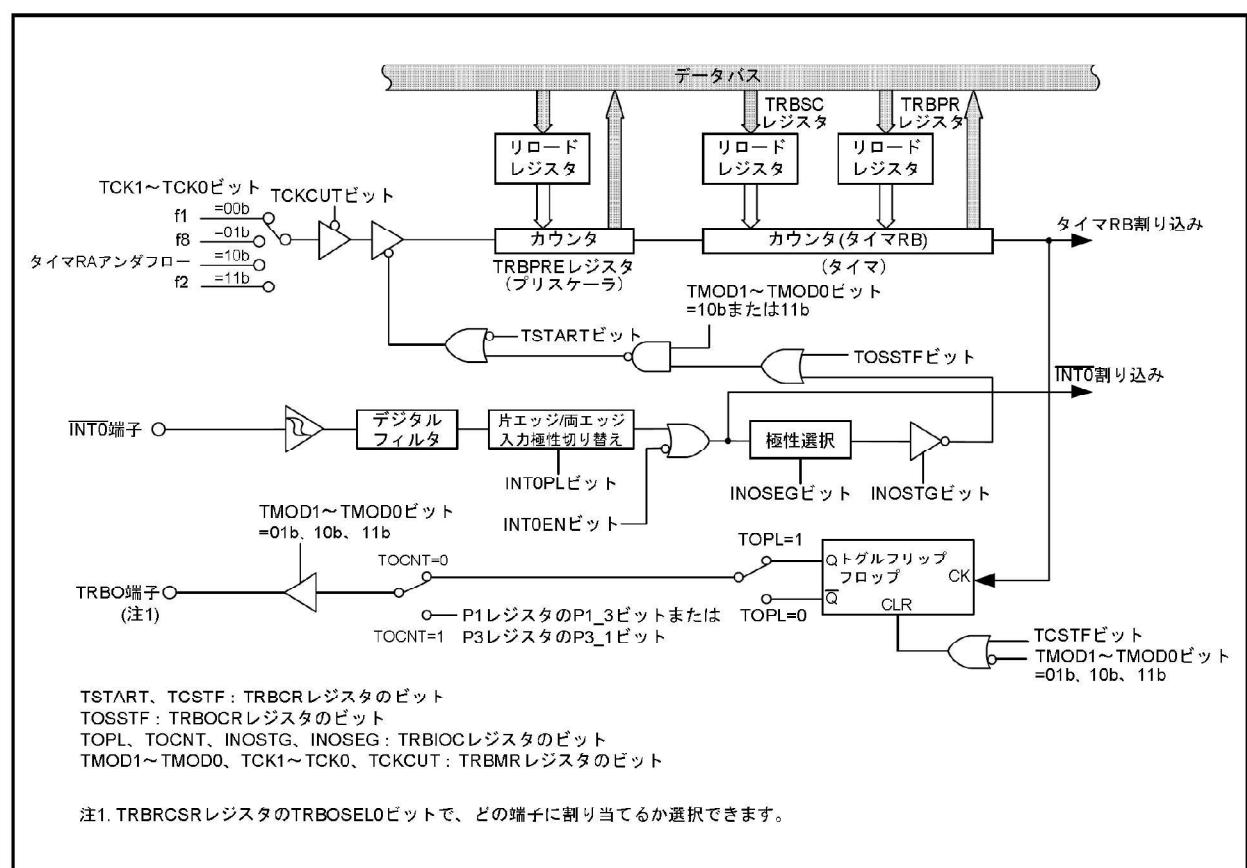
マスカブル割り込み	や割り込み優先レベルによる割り込み優先順位の変更が可能
	フラグレジスタ(FLG)の割り込み許可フラグ(I フラグ)による割り込みの許可(禁止) 不可能

タイマ RB を使って、1ms ごとに割り込みを発生させます。

81 :	/* 割り込み周期 = 1 / 20[MHz]      * (TRBPRE+1) * (TRBPR+1)	
83 :	= 0.001[s] = 1[ms]	
85 :	trbmr = 0x00;	/* 動作モード、分周比設定      */
87 :	trbpr = 100-1;	/* プライマリレジスタ      */
89 :	trbcr = 0x01;	/* カウント開始      */

R8C/35A には、タイマ RB というタイマが 1 個内蔵されています。タイマ RB には、次の 4 種類のモードがありま

モード	
	内部カウントソース(周辺機能クロックまたはタイマ RA のアンダフロー)をカウントする
波形発生モード	
ワンショット発生モード	
ワンショット発生モード	

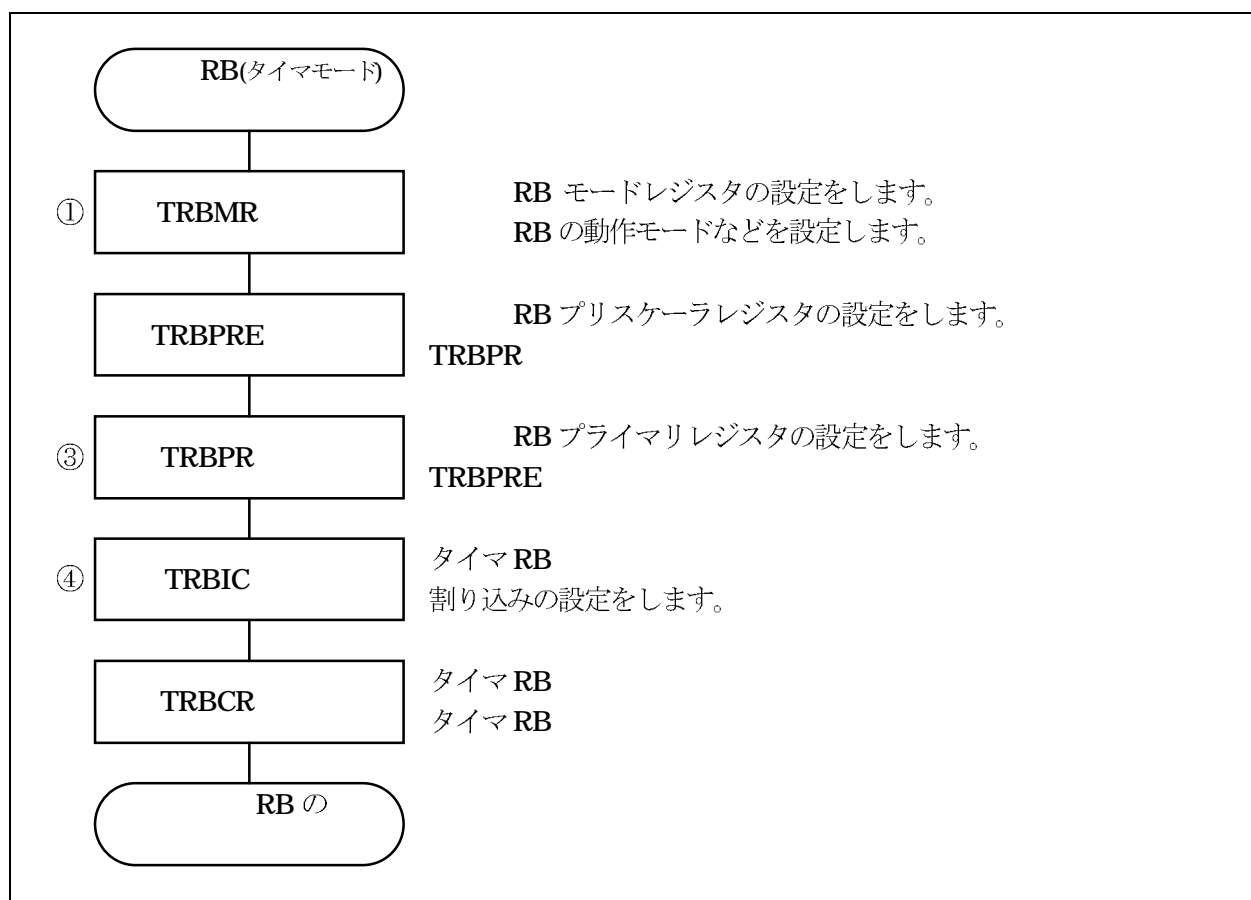


# ※タイマ RB の端子構成

		出力	プログラマブルワンショット発生モード、



今回は、タイマ RB をタイマモードで使用して、1ms ごとに割り込みを発生させるように設定にします。レジスタの



タイマ RB のモードを設定します。

	上:ビット名	内容	内容
	タイマ RB カウントソース遮断 tckcut_trbmr	1:カウントソース遮断	0
		"0"を設定	
	タイマ RB カウントソース選択 bit5:tck1_trbmr	00:f1 (1/20MHz=50ns) 10:タイマ RA のアンダフロー  f1 を選択します。	
	タイマ RB 書き込み制御ビット twrc_trbmr	1:リロードレジスタのみ書き込み	0
		"0"を設定	
	タイマ RB 動作モード選択ビット bit1:tmod1_trbmr	00:タイマモード 10:プログラマブルワンショット発生モード  タイマモードで動作させるので"00"を設定します。	

TCSTF ビットが共に"0"(カウント停止)のときに変更してください。

ラマブルワンショット発生モード、プログラマブルウェイトワンショット発生モードでは"1"(リロードレジスタの

タイマ RB モードレジスタ(TRBMR)の設定値を下記に示します。

	0		0	0	0		0	0
	0							

タイマ RB モードレジスタ (TRBMR) のタイマ RB カウントソース選択ビット (bit5,4) で、タイマ RB プリスケアラレジ  
タイマ RB モードレジスタ (TRBMR) のタイマ RB カウントソース選択ビットの値と、割り込み間隔の関係を下記に

TRBMR	内容
	<p>タイマ RB プリスケアラレジスタ (TRBPRES) がカウントアップする時間を、f1 に設定します。時間は、設定できる割り込み間隔の最大は、 <b>3.2768ms</b></p> <p>設定したい場合は次以降の値を検討します。</p>
	<p>タイマ RB プリスケアラレジスタ (TRBPRES) がカウントアップする時間を、f2 に設定します。時間は、設定できる割り込み間隔の最大は、 <b>6.5536ms</b></p> <p>設定したい場合は次以降の値を検討します。</p>
	<p>タイマ RB プリスケアラレジスタ (TRBPRES) がカウントアップする時間を、f8 に設定します。時間は、設定できる割り込み間隔の最大は、 <b>26.2144ms</b></p> <p><b>これ以上の割り込み間隔を設定することはできません</b> ム側で工夫してください。</p>

“00”の設定…最大の割り込み間隔は 3.2768ms、今回設定したい 1ms の割り込み間隔を設定できるので OK

---

③タイマ RB プライマリレジスタ(TRBPR:Timer RB Primary Register)の設定

を設定します。

$$\text{タイマ RB 割り込み要求周期} = \text{タイマ RB カウントソース} \times (\text{TRBPRE} + 1) \times (\text{TRBPR} + 1)$$

$$(\text{TRBPRE} + 1) \times (\text{TRBPR} + 1) = \text{タイマ RB 割り込み要求周期} / \text{タイマ RB カウントソース}$$

bit5,4 に設定している内容で、今回は f1 (50ns) です。よって、

$$\frac{(\text{TRBPRE} + 1)}{\text{-----}} \quad \text{-----} \quad \text{-----}$$

-3                      -9

次の条件になるよう、A、B、C 部分を設定してください。

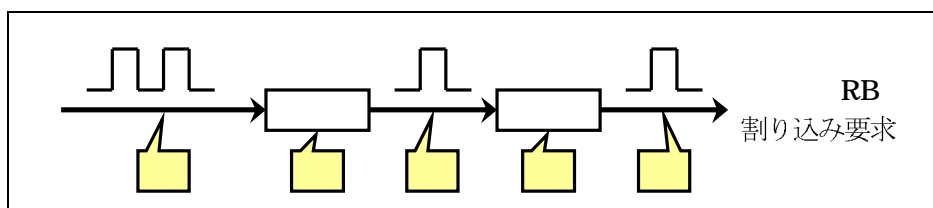
B…1～256 以下になるよう、値を設定してください。値は整数です。

今回、A は 20,000 なので、A の条件は満たしています。

例えば、B=200 とすると、

$$\therefore C = 100$$

$$B = \text{TRBPRE} + 1 \quad \therefore \text{TRBPRE} \quad \mathbf{199}$$
$$\quad \quad \quad \text{TRBPR} \quad \mathbf{99}$$



	タイマRBモードレジスタ(TRBMR)のRBカウントソース選択ビットで設定したパルスが入力されます。今回は $f1=1/20\text{MHz}=50\text{ns}$
	タイマ RB プリスケーラレジスタ (TRBPRES) はダウンカウントです。設定した値からスタートし、1 つずつ値が 8...となります。このように 0 も含めてカウントするため、10 回カウントしたければ、1 小さい値の 9 を設定し 今回は、199 を設定します。そのため、 $199 \rightarrow 198 \rightarrow \dots \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow 199 \rightarrow 198 \dots$ 、とカウントされます。
	タイマRBプリスケーラレジスタ (TRBPRES) が $0 \rightarrow 199$ になった瞬間、1 パルス出力されます。これは TRBPRES $50\text{ns}(\text{入力されるパルスの間隔}) \times 200 = 10,000\text{ns} = 10 \mu\text{s}$
	タイマRBプライマリレジスタ (TRBPR) はダウンカウントです。設定値した値からスタート、0 の次は設定値に 今回は 99 を設定します。そのため、 $99 \rightarrow 98 \rightarrow \dots \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow 99 \rightarrow 98 \dots$ 、とカウントされます。TRBPR に
	タイマRBプライマリレジスタ (TRBPR) が $0 \rightarrow 99$ になった瞬間、1 パルス出力されます。これは TRBPR に 100 $10 \mu\text{s}(\text{入力されるパルスの間隔}) \times 100 = 1,000 \mu\text{s} = 1\text{ms}$

このパルスが割り込みを発生させるきっかけになり

ます

タイマ RB プリスケーラレジスタ (TRBPRES) の設定値を下記に示します。

	1 9 9							

bit	7	6	5	4	3	2	1	0
	9 9							

タイマ RB の割り込み関係の設定をします。

	上:ビット名	内容	内容
		"0"を設定	
	割り込み要求ビット	0:割り込み要求なし  <b>割り込みが発生すると自動で"1"になります。割り込みプログラムプログラムを実行すると自動的に"0"になります。設定は、"0"にします。</b>	
	割り込み優先レベル選択ビット  bit2: ilvl2_trbic  bit0: ilvl0_trbic	001:レベル 1  011:レベル 3  101:レベル 5  111:レベル 7  優先させるか設定します。レベルの高い割り込みが優先させるかここで決めます。今回の割り込みは、タイマせん。一応、レベルのいちばん高い"111"を設定しま	111

bit	7	6	5	4	3	2	1	0
		0	0		0	1	1	1
	0							

タイマ RB のカウント動作を開始するよう設定します。

	上:ビット名	内容	内容
		"00000"を設定	
	タイマ RB カウント強制停止ビット tstop_trbcr	読んだ場合、その値は"0"になります。	
	タイマRB カウントステータス tcstf_trbcr	1:カウント中(注 3)  無効です。書き込むときは"0"を設定します。	
	タイマ RB カウント開始ビット tstart_trbcr	1:カウント開始  設定した瞬間から、カウントが開始されます。	

マ RB 使用上の注意」を参照してください。

CSTF

注 3. タイマモード、プログラマブル波形発生モードでは、カウント中を示します。プログラマブルワンショット発生を示します。

bit	7	6	5	4	3	2	1	0
		0	0		0		0	1
	0							

先の設定で、タイマ RB を 1ms ごとに割り込みを発生させる設定にしました。intTRB 関数は、この割り込みが発

```
105 : #pragma interrupt intTRB(vect=24)
107 : {
109 : }
```

	<pre>#pragma interrupt [ ] (vect=[ ]) [ ]</pre> <p>ソフトウェア割り込み番号の表を次ページに示します。タイマ RB 割り込みは表より、24 番です。</p> <p>定めます。</p> <pre>#pragma interrupt</pre>
	タイマ RB 割り込みにより実行する関数です。割り込み関数は、引数、戻り値ともに指定することでは void 関数名 ( void )
	cnt_rb 変数を+1します。この関数は 1ms ごとに実行されるので、cnt_rb は 1ms ごとに+1 されることに



---

割り込み要因とソフトウェア割り込み番号の関係は、下記のとおりです。



今回は、タイマ RB を使用して割り込みを発生させるので、表より番号は 24 番となります。

数は「割り込み処理関数名」ですよということを、宣言します。

<div></div>
-------------

<div>void 割り込み処理関数名( void )</div> <div>プログラム</div>
--

timer 関数は、実行した行で時間稼ぎをする関数です。プロジェクト「timer1」はソフトウェアによるタイマでした。  
ールチェーンの設定により、時間が変わる可能性があります。今回の timer 関数は、クリスタルの値を基準として

<div>92 :  /*****</div> <div>94 :  /* 引数   タイマ値 1=1ms                               */</div> <div>96 :  void timer( unsigned long timer_set )</div> <div>98 :          cnt_rb = 0;</div> <div>100 :  }</div>
--

	<div>cnt_rb が timer_set より小さいなら、99 行を繰り返し続けます。</div> <div>数でセットした値です。</div> <div>timer( 500 );</div> <div>timer 関数を実行したいちばん最初は、</div> <div>となり、成り立つので 99 行を繰り返します。1ms 後は、割り込みプログラムで cnt_rb が+1 されるので、</div> <div>となります。まだ成り立つので、99 行目を繰り返します。timer 関数を実行してから 500ms たったな</div> <div>while( 500 &lt; 500 );</div> <div>す。</div>

---

```
39 : void main( void )  
  
41 :     init();                /* 初期化                */  
  
43 :  
  
45 :         p6 = 0x55;  
  
47 :         p6 = 0xaa;  
  
49 :         p6 = 0x00;  
  
51 :     }
```

	全体の割り込みを許可する命令です。
--	-------------------

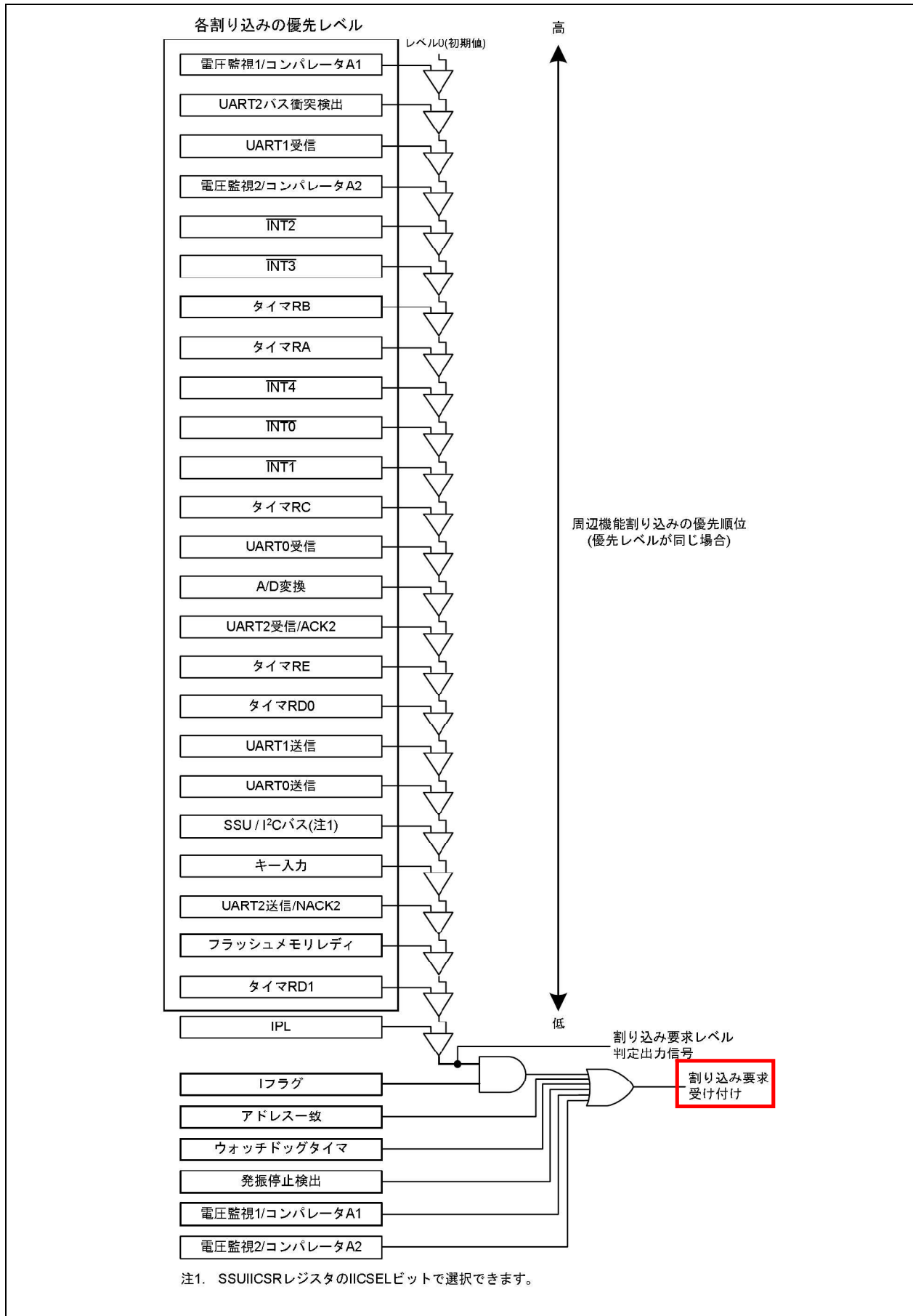
	は発生しません。全体の割り込みを許可する命令は、C 言語で記述することができないため、asm
--	--

main 関数の while(1)のカッコ内の C 言語をアセンブリ言語に変換すると、下記のようになります。

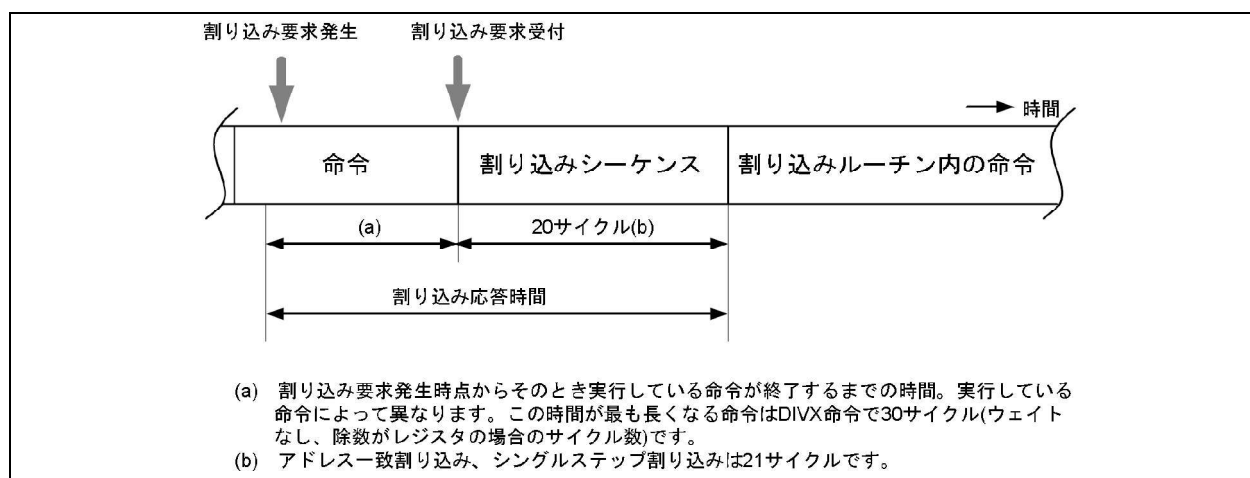
<p>； 変換した内容</p> <pre>mov.w    #0001H, R0  mov.b    #55H, _p6_addr  push.w   #03e8H  add.b    #04H, SP</pre> <p>(右上へ続く)</p>	<pre>push.w   #0000H  jsr      _timer  mov.b    #00H, _p6_addr  push.w   #03e8H  add.b    #04H, SP  L3:</pre>
--	---

イメージを下記に示します。

<p>； 変換した内容</p> <pre>mov.w    #0001H, R0</pre> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p>	<p>(左下から続き)</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>割り込み要求があるかチェック</p> <p>L3:</p>
--	--



割り込み応答時間を下記に示します。割り込み応答時間は、割り込み要求が発生してから割り込みルーチン命令が終了するまでの時間(a)と割り込みシーケンスを実行する時間(20 サイクル(b))で構成されます。



$$1/20\text{MHz}=50\text{ns}$$

$$\text{割り込みシーケンスの時間} = 1 \text{ サイクル} \times 20 = 50\text{ns} \times 20 = 1000\text{ns} = 1 \mu\text{s}$$

割り込みルーチンの処理が終わったら、割り込みシーケンス発生前に実行していた命令の、次の命令から実