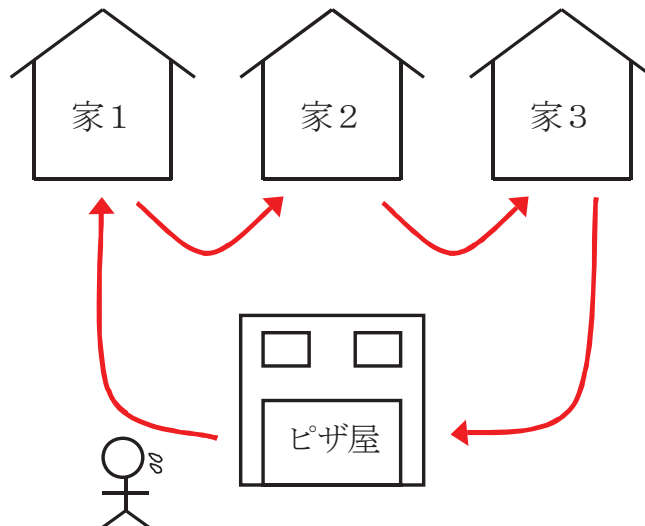


14.5 プログラムの解説

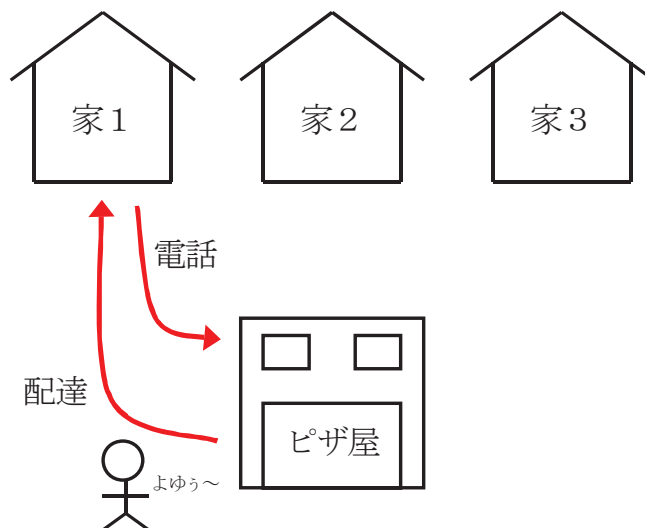
14.5.1 割り込みとは

(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。定期的に聞きに行くことを制御の用語で**ポーリング**といいます。回る間隔が長いと、待たせることになります。また、注文がなければ無駄足になってしまいます(下図)。



そこで、電話で注文を受けることにします。バイト君は、それぞれの家を回る必要がありません。電話の注文が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)。



ただし、電話を用意する必要があります。プログラムに当てはめると、割り込み設定に当たります。さらに、電話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。

まとめると下記ようになります。

・注文がないか聞きに回る

制御の用語で「ポーリング」といいます。定期的に監視しなければいけないので、監視する部分が多いと、監視が遅れたり、監視もれが起きます。

・電話で注文をうける

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ば良いので効率が良いです。ただし、電話の用意、電話の受け答えをする必要があります。

人で例えましたが、マイコンの場合は下記ようになります。

人間の場合

電話を用意する

→

ベルが鳴る

→

電話対応する

→

マイコンの場合

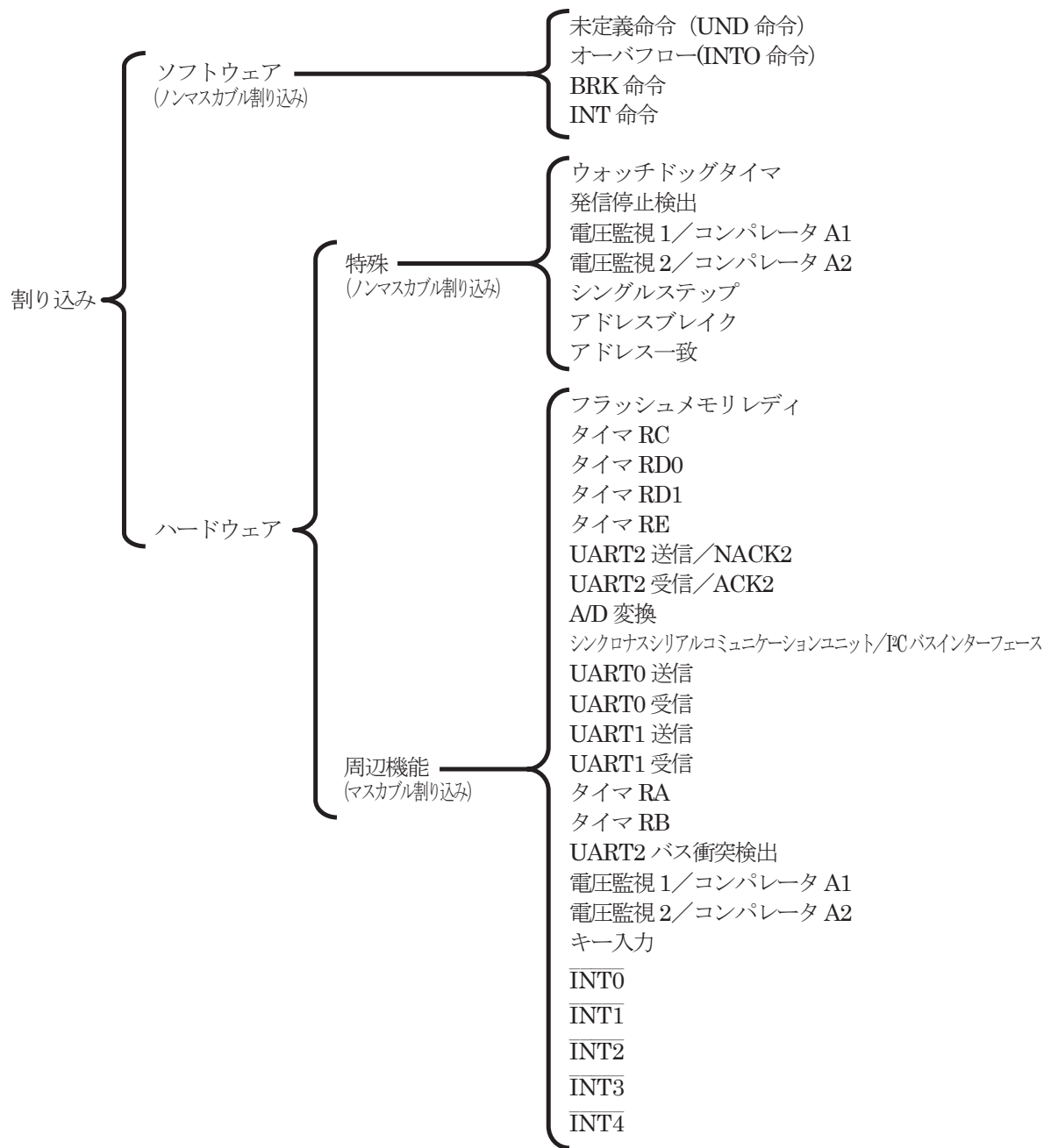
割り込みプログラムを設定する

割り込みが発生する

割り込みプログラムを実行する

(2) 割り込みの種類

R8C/35A の割り込みの種類を、下表に示します。



マスクابل割り込み	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)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
ノンマスクابل割り込み	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)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 不可能

14.5.2 init関数(タイマRBの設定)

タイマ RB を使って、1ms ごとに割り込みを発生させます。

80 :	/* タイマRBの設定 */		
81 :	/* 割り込み周期 = 1 / 20[MHz]	* (TRBPRE+1) * (TRBPR+1)	
82 :	= 1 / (20*10 ⁻⁶) * 200	* 100	
83 :	= 0.001[s] = 1[ms]		
84 :	*/		
85 :	trbmr = 0x00;	/* 動作モード、分周比設定	*/
86 :	trbpre = 200-1;	/* プリスケアラレジスタ	*/
87 :	trbpr = 100-1;	/* プライマリレジスタ	*/
88 :	trbic = 0x07;	/* 割り込み優先レベル設定	*/
89 :	trbcr = 0x01;	/* カウント開始	*/

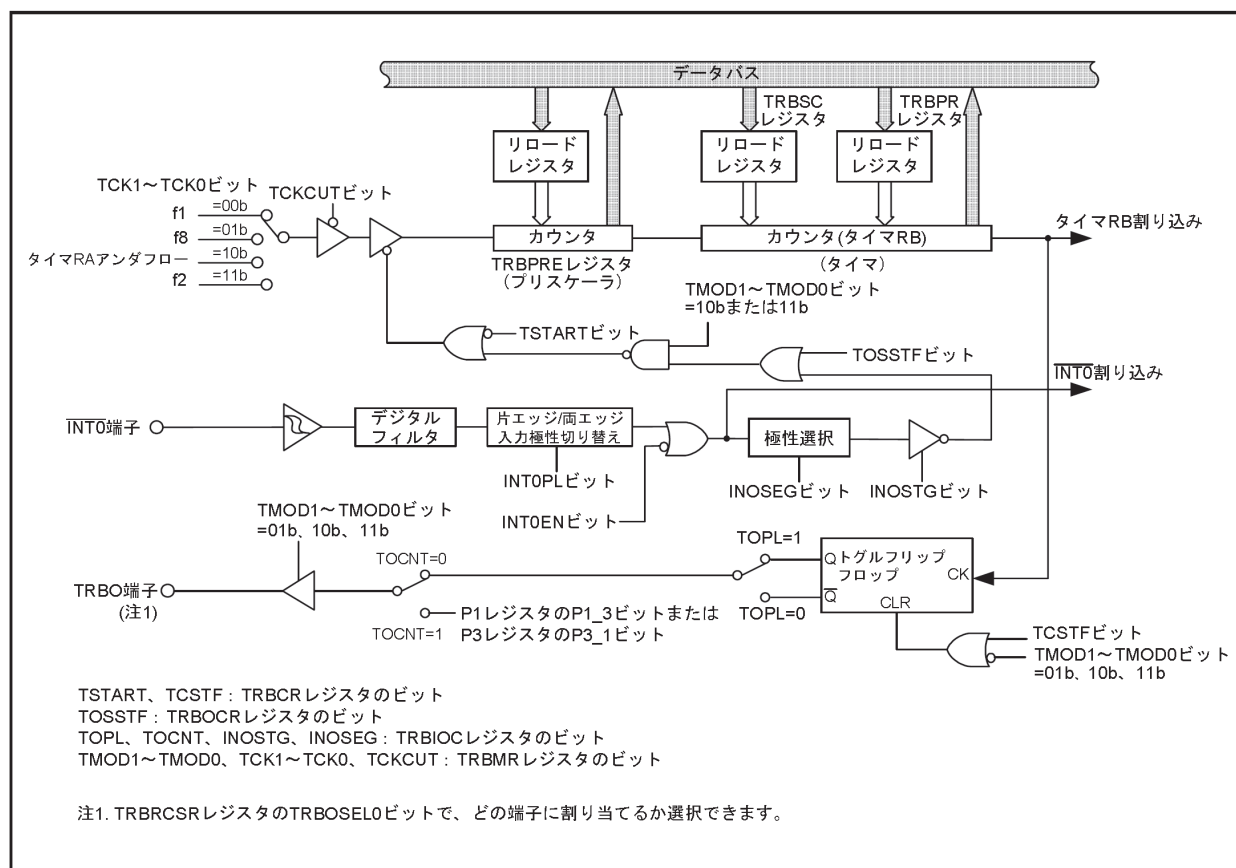
(1) タイマRBとは

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す。

モード	詳細
タイマモード	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)をカウントするモードです。
プログラマブル 波形発生モード	任意のパルス幅を連続して出力するモードです。
プログラマブル ワンショット発生モード	ワンショットパルスを出力するモードです。
プログラマブルウェイト ワンショット発生モード	ディレイドワンショットパルスを出力するモードです。

本プロジェクトでは、タイマモードを使い、1ms ごとに割り込みを発生させます。

(2) タイマRBのブロック図



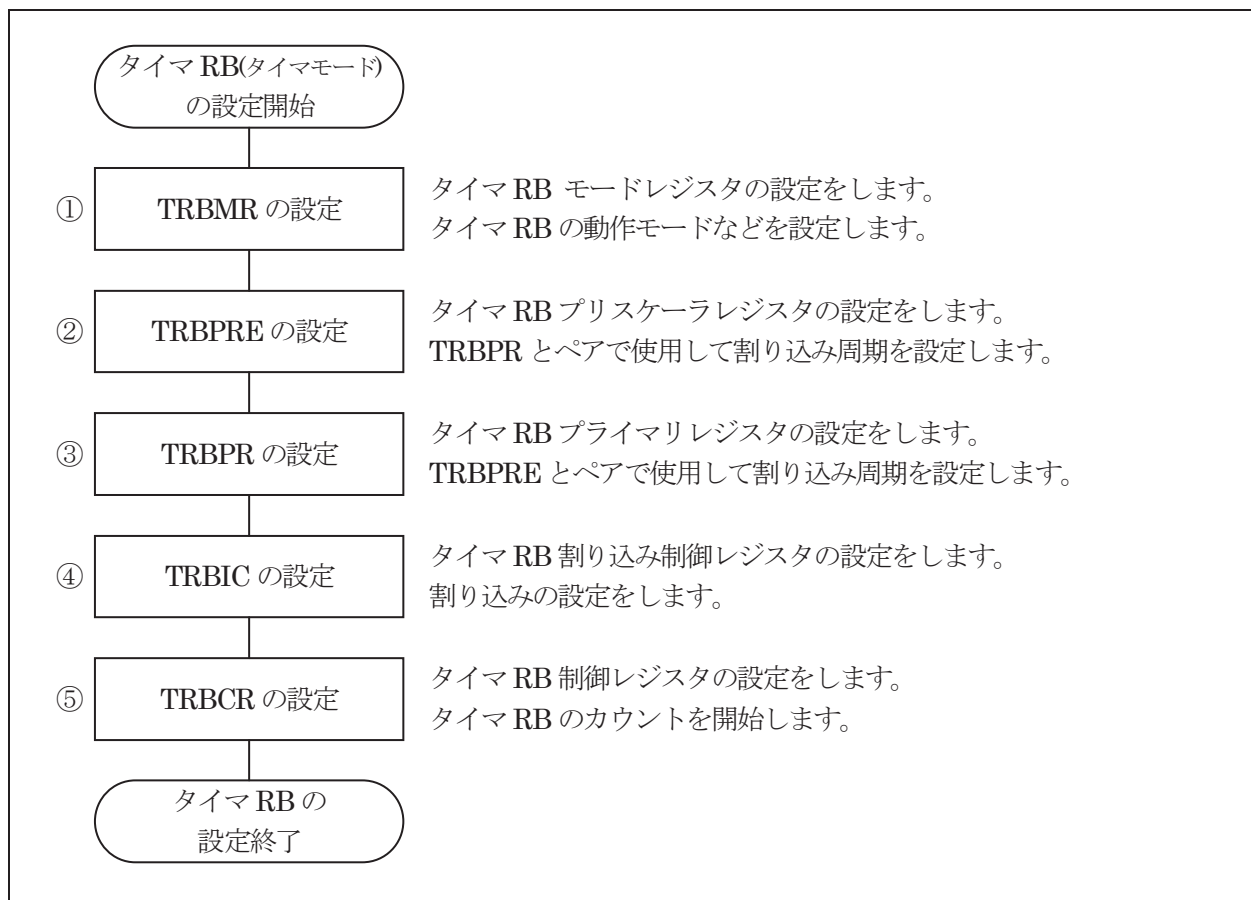
※タイマRBの端子構成

端子名	割り当てる端子	入出力	機能
TRBO	P1_3 または P3_1	出力	パルス出力(プログラマブル波形発生モード、プログ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)

今回は、1ms ごとの割り込みを発生させるだけなので、端子は使いません。

(3) タイマRBの設定(タイマモード)

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。レジスタの設定手順を下記に示します。



①タイマ RB モードレジスタ (TRBMR:Timer RB mode register)の設定

タイマ RB のモードを設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	タイマ RB カウントソース遮断 ビット(注 1) tckcut_trbmr	0:カウントソース供給 1:カウントソース遮断 供給するので"0"を設定します。	0
bit6		"0"を設定	0
bit5,4	タイマ RB カウントソース選択 ビット(注 1) bit5:tck1_trbmr bit4:tck0_trbmr	00:f1 (1/20MHz=50ns) 01:f8 (8/20MHz=400ns) 10:タイマ RA のアンダフロー 11:f2 (2/20MHz=100ns) f1 を選択します。	00
bit3	タイマ RB 書き込み制御ビット (注 2) twrc_trbmr	0:リロードレジスタとカウンタへの書き込み 1:リロードレジスタのみ書き込み "0"を設定します。	0
bit2		"0"を設定	0
bit1,0	タイマ RB 動作モード選択ビッ ト(注 1) bit1:tmod1_trbmr bit0:tmod0_trbmr	00:タイマモード 01:プログラマブル波形発生モード 10:プログラマブルワンショット発生モード 11:プログラマブルウェイト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。	00

注 1. TMOD1～TMOD0 ビット、TCK1～TCK0 ビット、TCKCUT ビットは、TRBCR レジスタの TSTART ビットと 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。

注 2. TWRC ビットは、タイマモードのとき"0"または"1"が選択できます。プログラマブル波形発生モード、プログ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み書き込み)にしてください。

タイマ RB モードレジスタ (TRBMR)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			

※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設定方法

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 プリスケールレジスタ(TRBPRES)、タイマ RB プライマリレジスタ(TRBPR)がどのくらいの間隔で+1 するか設定します。

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示します。

TRBMR bit5,4	内容
00	<p>タイマ RB プリスケールレジスタ(TRBPRES)がカウントアップする時間を、f1 に設定します。時間は、 $f1/20\text{MHz}=1/20\text{MHz}=50\text{ns}$ 設定できる割り込み間隔の最大は、 $50\text{ns} \times 65,536 = \mathbf{3.2768\text{ms}}$ よって、この時間以内の割り込み間隔を設定する場合は“00”を設定、これ以上の割り込み間隔を設定したい場合は次以降の値を検討します。</p>
11	<p>タイマ RB プリスケールレジスタ(TRBPRES)がカウントアップする時間を、f2 に設定します。時間は、 $f2/20\text{MHz}=2/20\text{MHz}=100\text{ns}$ 設定できる割り込み間隔の最大は、 $100\text{ns} \times 65,536 = \mathbf{6.5536\text{ms}}$ よって、この時間以内の割り込み間隔を設定する場合は“11”を設定、これ以上の割り込み間隔を設定したい場合は次以降の値を検討します。</p>
01	<p>タイマ RB プリスケールレジスタ(TRBPRES)がカウントアップする時間を、f8 に設定します。時間は、 $f8/20\text{MHz}=8/20\text{MHz}=400\text{ns}$ 設定できる割り込み間隔の最大は、 $400\text{ns} \times 65,536 = \mathbf{26.2144\text{ms}}$ よって、この時間以内の割り込み間隔を設定する場合は“01”を設定します。これ以上の割り込み間隔を設定することはできません。これ以上の割り込み間隔を設定しなくても良いように、プログラム側で工夫してください。</p>

今回は、割り込み間隔を 1ms にします。

“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

よって、“00”を設定します。

②タイマ RB プリスケアラレジスタ(TRBPRES:Timer RB prescaler register)の設定

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

タイマ RB プリスケアラレジスタ(TRBPRES)とタイマ RB プライマリレジスタ(TRBPR)はペアで使い、割り込み周期を設定します。

TRBPRES と TRBPR の値を計算する式を、下記に示します。

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

TRBPRES と TRBPR を左辺に移動します。

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

今回、設定する割り込み周期は 1ms です。タイマ RB カウントソースとは、タイマ RB モードレジスタ(TRBMR)の bit5,4 に設定している内容で、今回は f1 (50ns) です。よって、

$$(\text{TRBPRES} + 1) \times (\text{TRBPR} + 1) = (1 \times 10^{-3}) / (50 \times 10^{-9})$$

$$\underbrace{(\text{TRBPRES} + 1)}_B \times \underbrace{(\text{TRBPR} + 1)}_C = \underbrace{20,000}_A$$

次の条件になるよう、A、B、C 部分を設定してください。

A…65,536 以下にする必要があります。65,537 以上の場合、カウントソースを長い時間に設定し直してください。

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。

C…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。

B、C を決める公式はありません。B×C が、20,000 になるような数字を見つけてください。

例えば、B=200 とすると、

$$200 \times C = 20,000$$

$$\therefore C = 100$$

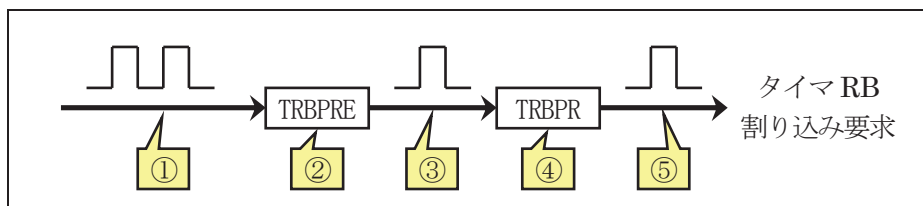
となります。

$$B = \text{TRBPRES} + 1 \quad \therefore \text{TRBPRES} = 200 - 1 = \mathbf{199}$$

$$C = \text{TRBPR} + 1 \quad \therefore \text{TRBPR} = 100 - 1 = \mathbf{99}$$

を設定します。

このときの動作を下記に示します。



①	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 を選択しているので、次のパルスが入力されます。 f1=1/20MHz=50ns
②	タイマ RB プリスケールレジスタ (TRBPRES) はダウンカウントです。設定した値からスタートし、1 つずつ値が減っていき 0 の次は設定値になります。例えば 9 を設定したなら 9→8→7→6→5→4→3→2→1→0→9→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 を設定します。 今回は、199 を設定します。そのため、199→198→・・・→2→1→0→199→198・・・、とカウントされます。
③	タイマ RB プリスケールレジスタ (TRBPRES) が 0→199 になった瞬間、1 パルス出力されます。これは TRBPRES に 200 パルス入ると 1 パルス出力されるということです。パルスが出力される間隔は、次のようになります。 $50\text{ns}(\text{入力されるパルスの間隔}) \times 200 = 10,000\text{ns} = 10\mu\text{s}$
④	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 の次は設定値になります。 今回は 99 を設定します。そのため、99→98→・・・→2→1→0→99→98・・・、とカウントされます。TRBPR には $10\mu\text{s}$ ごとにパルスが入力されます。要は、 $10\mu\text{s}$ ごとに、TRBPR が-1 されます。
⑤	タイマ RB プライマリレジスタ (TRBPR) が 0→99 になった瞬間、1 パルス出力されます。これは TRBPR に 100 パルス入ると 1 パルス出力されるということです。パルスが出力される間隔は、次のようになります。 $10\mu\text{s}(\text{入力されるパルスの間隔}) \times 100 = 1,000\mu\text{s} = 1\text{ms}$

このように、TRBPR から 1ms ごとにパルスが出力されます。**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。**要は、タイマ RB によって 1ms ごとに割り込みを発生させます。

タイマ RB プリスケールレジスタ (TRBPRES) 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 9 9							

タイマ RB プライマリレジスタ (TRBPR) の設定値を下記に示します。

bit	7	6	5	4	3	2	1	0
設定値	9 9							

④タイマ RB 割り込み制御レジスタ(TRBIC:Timer RB interrupt control register)の設定

タイマ RB の割り込み関係の設定をします。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～4		"0"を設定	0000
bit3	割り込み要求ビット ir_trbic	0:割り込み要求なし 1:割り込み要求あり 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。	0
bit2～0	割り込み優先レベル選択ビット bit2: ilvl2_trbic bit1: ilvl1_trbic bit0: ilvl0_trbic	000:レベル 0 (割り込み禁止) 001:レベル 1 010:レベル 2 011:レベル 3 100:レベル 4 101:レベル 5 110:レベル 6 111:レベル 7 他の割り込みが同時に発生した場合、どの割り込みを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れます。割り込みを2つ以上使う場合は、どれを優先させるかここで決めます。今回の割り込みは、タイマ RB だけなのでレベル 1～7 のどれを設定しても構いません。一応、レベルのいちばん高い"111"を設定します。	111

タイマ RB 割り込み制御レジスタ(TRBIC)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	1	1	1
16 進数	0				7			

⑤タイマ RB 制御レジスタ (TRBCR: Timer RB Control Register) の設定

タイマ RB のカウント動作を開始するよう設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～3		”00000”を設定	00000
bit2	タイマ RB カウント強制停止ビット(注 1、2) tstop_trbcr	“1”を書くとカウントが強制停止します。 読んだ場合、その値は“0”になります。	0
bit1	タイマ RB カウントステータスフラグ(注 1) tcstf_trbcr	0:カウント停止 1:カウント中(注 3) カウント中かどうかチェックするフラグです。書き込みは無効です。書き込むときは“0”を設定します。	0
bit0	タイマ RB カウント開始ビット(注 1) tstart_trbcr	0:カウント停止 1:カウント開始 タイマ RB のカウントを開始するので“1”を設定します。 設定した瞬間から、カウントが開始されます。	1

注 1. TSTART、TCSTF、TSTOP ビットの使用上の注意事項については、ハードウェアマニュアルの「18.7 タイマ RB 使用上の注意」を参照してください。

注 2. TSTOP ビットに“1”を書くと、TRBPRES レジスタ、TRBSC レジスタ、TRBPR レジスタ、TSTART ビット、TCSTF ビット、TRBOCR レジスタの TOSSTF ビットがリセット後の値になります。

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、ワンショットパルスのトリガを受け付けられることを示します。

タイマ RB 制御レジスタ (TRBCR) 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			

14.5.3 intTRB関数(1msごとに実行される関数)

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、この割り込みが発生したときに実行される関数です。

```

105 : #pragma interrupt intTRB(vect=24)
106 : void intTRB( void )
107 : {
108 :     cnt_rb++;
109 : }
```

105 行	<p>#pragma interrupt 割り込み処理関数名(vect=ソフトウェア割り込み番号) とすることで、ソフトウェア割り込み番号の割り込みが発生したとき、割り込み処理関数名を実行します。 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。 よって、24 番の割り込みが発生したときに intTRB 関数を実行するよう、「#pragma interrupt」で設定します。</p>
106 行	<p>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はできません。すなわち、「void 関数名(void)」である必要があります。</p>
108 行	<p>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 されることになります。</p>

※ソフトウェア割り込み番号

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。

割り込み要因	ベクタ番地(注1) 番地(L)～番地(H)	ソフトウェア 割り込み番号	割り込み制御 レジスタ	参照先
BRK 命令(注3)	+0～+3(0000h～0003h)	0	—	R8C/Tinyシリーズ ソフトウェアマニュアル
フラッシュメモリレディ	+4～+7(0004h～0007h)	1	FMRDYIC	32. フラッシュメモリ
—(予約)		2～5	—	—
INT4	+24～+27(0018h～001BFh)	6	INT4IC	11.4 INT割り込み
タイマRC	+28～+31(001Ch～001Fh)	7	TRCIC	19. タイマRC
タイマRD0	+32～+35(0020h～0023h)	8	TRD0IC	20. タイマRD
タイマRD1	+36～+39(0024h～0027h)	9	TRD1IC	
タイマRE	+40～+43(0028h～002Bh)	10	TREIC	21. タイマRE
UART2送信/NACK2	+44～+47(002Ch～002Fh)	11	S2TIC	23. シリアルインタフェース (UART2)
UART2受信/ACK2	+48～+51(0030h～0033h)	12	S2RIC	
キー入力	+52～+55(0034h～0037h)	13	KUPIC	11.5 キー入力割り込み
A/D変換	+56～+59(0038h～003Bh)	14	ADIC	28. A/Dコンバータ
シンクロナスシリアルコミュニ ケーションユニット/I ² Cバ スインタフェース(注2)	+60～+63(003Ch～003Fh)	15	SSUIC/ IICIC	25. シンクロナスシリアルコミュニ ケーションユニット(SSU)、 26. I ² Cバスインタフェース
—(予約)		16	—	—
UART0送信	+68～+71(0044h～0047h)	17	S0TIC	22. シリアルインタフェース (UARTi (i=0～1))
UART0受信	+72～+75(0048h～004Bh)	18	S0RIC	
UART1送信	+76～+79(004Ch～004Fh)	19	S1TIC	
UART1受信	+80～+83(0050h～0053h)	20	S1RIC	
INT2	+84～+87(0054h～0057h)	21	INT2IC	11.4 INT割り込み
タイマRA	+88～+91(0058h～005Bh)	22	TRAIC	17. タイマRA
—(予約)		23	—	—
タイマRB	+96～+99(0060h～0063h)	24	TRBIC	18. タイマRB
INT1	+100～+103(0064h～0067h)	25	INT1IC	11.4 INT割り込み
INT3	+104～+107(0068h～006Bh)	26	INT3IC	
—(予約)		27	—	—
—(予約)		28	—	—
INT0	+116～+119(0074h～0077h)	29	INT0IC	11.4 INT割り込み
UART2バス衝突検出	+120～+123(0078h～007Bh)	30	U2BCNIC	23. シリアルインタフェース (UART2)
—(予約)		31	—	—
ソフトウェア(注3)	+128～+131(0080h～0083h)～ +164～+167(00A4h～00A7h)	32～41	—	R8C/Tinyシリーズ ソフトウェアマニュアル
—(予約)		42～49	—	—
電圧監視1/コンパレータA1	+200～+203(00C8h～00CBh)	50	VCMP1IC	6. 電圧検出回路
電圧監視2/コンパレータA2	+204～+207(00CCh～00CFh)	51	VCMP2IC	30. コンパレータA
—(予約)		52～55	—	—
ソフトウェア(注3)	+224～+227(00E0h～00E3h)～ +252～+255(00FCh～00FFh)	56～63		R8C/Tinyシリーズ ソフトウェアマニュアル

注1. INTBレジスタが示す番地からの相対番地です。

注2. SSUIICSRレジスタのIICSELビットで選択できます。

注3. Iフラグによる禁止はできません。

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。

次のように、`#pragma interrupt` 命令を記述して、「ソフトウェア割り込み番号 24 番が発生したときに、実行する関数は `割り込み処理関数名` です」ということを、宣言します。

```
#pragma interrupt 割り込み処理関数名(vect=24)
```

関数名のプログラムを記述して、割り込みが発生したときに実行するプログラムを作成します。

```
void 割り込み処理関数名( void )
{
    プログラム
}
```

14.5.4 timer関数(割り込みを使った時間稼ぎ)

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。そのため、for 文で使った 1240 という数値を見つけるのは大変です。また、コンパイラのバージョンの違いやツ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いるため、正確な計測が可能です。

```
92 : /*****
93 :  /* タイマ本体                                */
94 :  /* 引数   タイマ値 1=1ms                        */
95 : *****/
96 : void timer( unsigned long timer_set )
97 : {
98 :     cnt_rb = 0;
99 :     while( cnt_rb < timer_set );
100 : }
```

98 行	cnt_rb を 0 にクリアします。
99 行	<p>cnt_rb が timer_set より小さいなら、99 行を繰り返し続けます。 cnt_rb は割り込みプログラムで 1ms ごとに+1 されます。timer_set は、timer 関数を実行したときに引数でセットした値です。 例えば、</p> <pre>timer(500);</pre> <p>と実行したなら、timer_set には 500 が入ります。 timer 関数を実行したいちばん最初は、</p> <pre>while(0 < 500);</pre> <p>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、</p> <pre>while(1 < 500);</pre> <p>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 たったなら、割り込みプログラムも 500 回実行され、cnt_rb は 500 となります。</p> <pre>while(500 < 500);</pre> <p>成り立たなくなり、次の行へ進みます。よって、99 行を実行し終わるまで 500ms かかることとなります。</p>

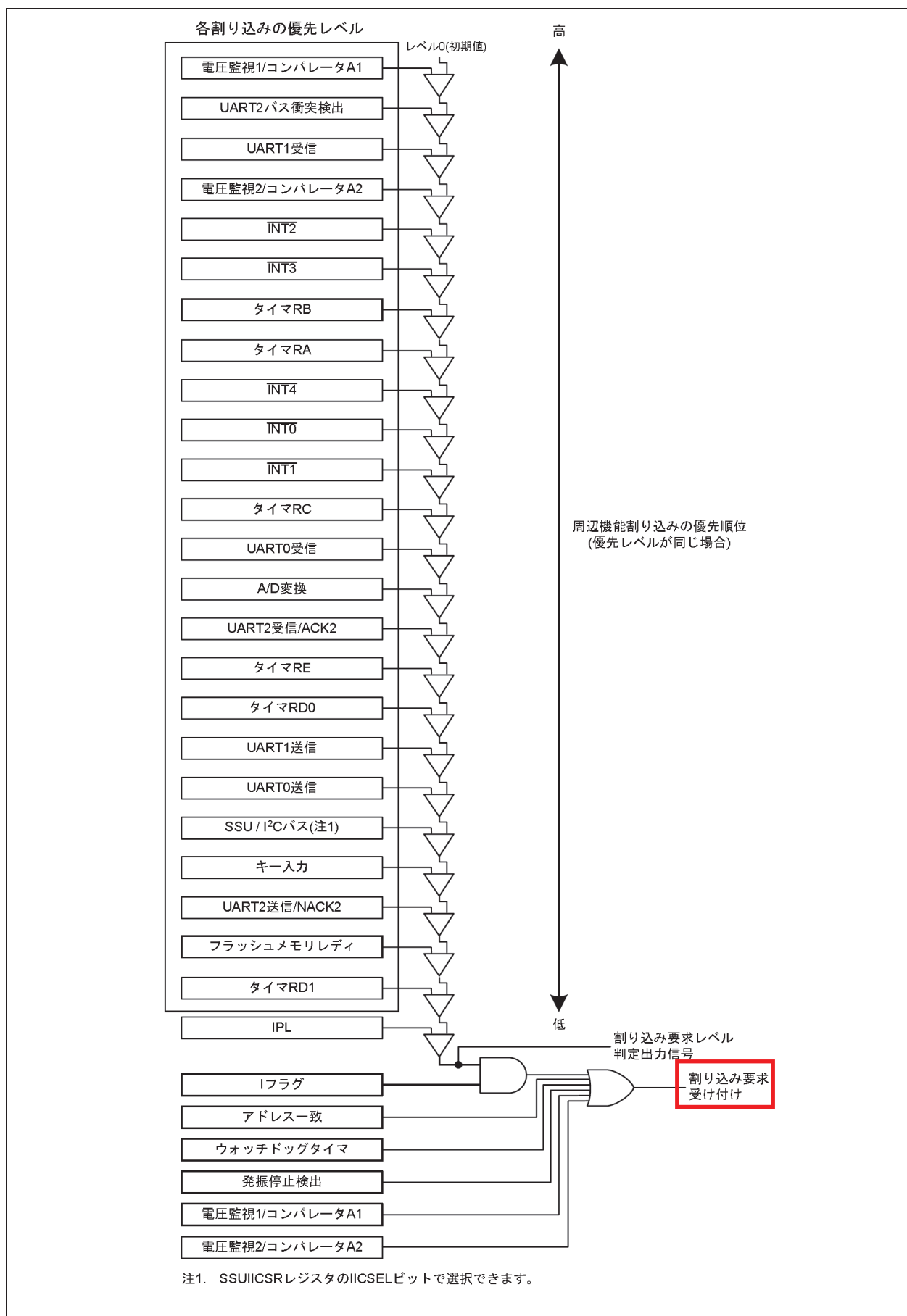
14.5.5 main関数

```
39 : void main( void )
40 : {
41 :     init();                      /* 初期化                */
42 :     asm( " fset I ");           /* 全体の割り込み許可    */
43 :
44 :     while( 1 ) {
45 :         p6 = 0x55;
46 :         timer( 1000 );
47 :         p6 = 0xaa;
48 :         timer( 1000 );
49 :         p6 = 0x00;
50 :         timer( 1000 );
51 :     }
52 : }
```

42 行

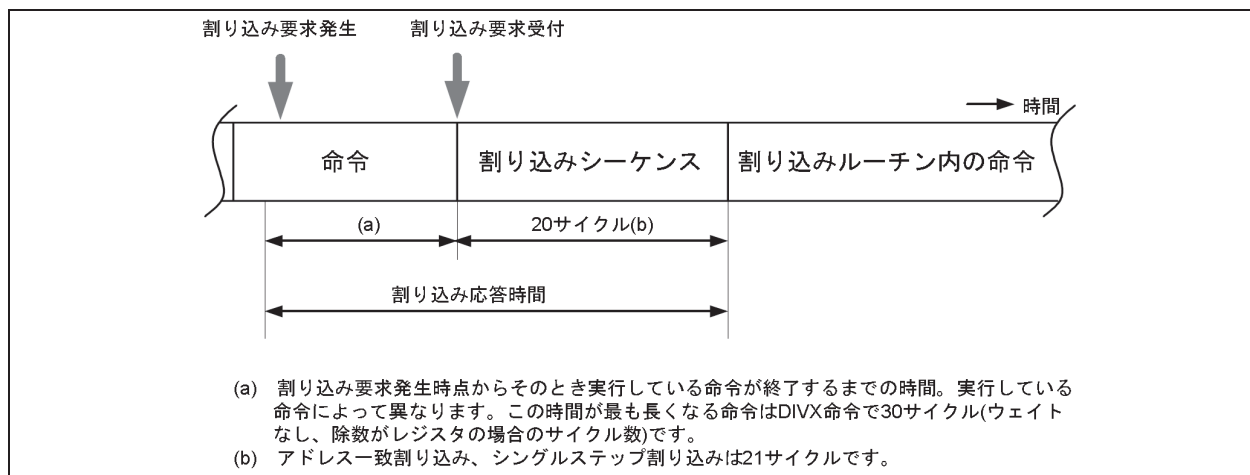
全体の割り込みを許可する命令です。
init 関数内でタイマ RB の割り込みを許可していますが、全体の割り込みを許可しなければ割り込みは発生しません。全体の割り込みを許可する命令は、C 言語で記述することができないため、asm 命令を使ってアセンブリ言語で割り込みを許可する命令を記述しています。

「割り込み要求があるかチェック」の部分は、下記の「割り込み要求受け付け」信号をチェック、「1」なら割り込みありと判断します。



「割り込み要求受け付け」が"1"なら、割り込みが発生した割り込みプログラムを実行します。

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内の最初の命令を実行するまでの時間です。この時間は、割り込み要求発生時点から、そのとき実行している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。



本マイコンボードは、20MHz のクリスタルで動作しています。1 サイクルは、

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

です。割り込みシーケンスの時間は、20 サイクルですので、

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

となります。

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、次の命令から実行を再開します。