

16. 外部割り込み(プロジェクト:int_interrupt)

16.1 概要

本章では、外部からの信号で割り込みをかけて(外部割り込み)、割り込みプログラムを実行する方法を説明します。今回は、INT3端子をP3_3に割り当てて、割り込みを発生させます。

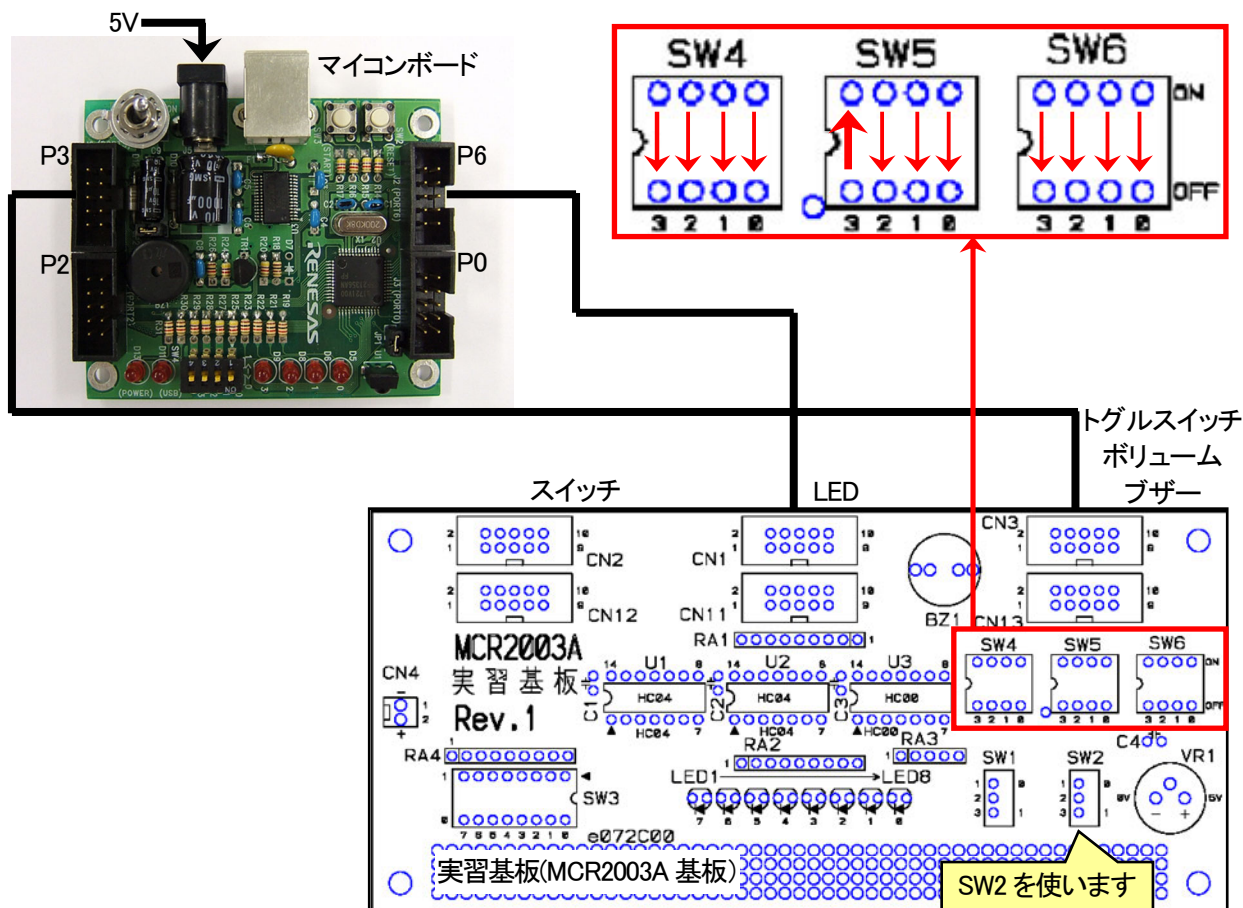
16.2 接続

■使用ポート

マイコンのポート	接続内容
P6 (J2)	実習基板のLED 部など、出力機器を接続します。
P3_3 (J6)	実習基板のトグルスイッチ部など、入力機器を接続します。
P1_3、P1_2、P1_1、P1_0	マイコンボード上のLED です。

■接続例

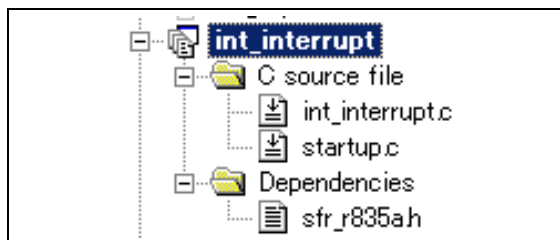
実習基板を使ったときの接続例を次に示します。



■操作方法

実習基板のトグルスイッチ SW2 を上下させると、マイコンボードの LED(D9,D8,D6,D5)の値が「0000」→「0001」→ … 」と、増えていきます(「0」は消灯、「1」は点灯)。その動作と関係なく、ポート 6 に接続している LED が 1 秒ごとに点灯します。ポート 6 に接続している LED の点灯方法は、プロジェクト「timer1」と同じです。

16.3 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	int_interrupt.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。

16.4 プログラム「int_interrupt.c」

```

1 :  /******
2 :  /* 対象マイコン R8C/35A
3 :  /* ファイル内容 INT割り込み
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 : INT3 (P3_3) (実習基板のトグルスイッチ部などチャタリングのない信号)
11 :  出力 : マイコンボードのLED(4bit)
12 :
13 :  INT3 (P3_3) 端子から入力された信号により割り込みプログラムを実行します。
14 :  割り込みは、立ち下がり (1→0の瞬間) でかかるように設定します。
15 :  INT3割り込みの発生ごとに、マイコンボードのLEDが+ 1 していきます。
16 :  メイン関数では、プロジェクト「timer1」のmain関数の内容を実行しています。
17 :  */
18 :
19 :  /*=====*/
20 :  /* インクルード */
21 :  /*=====*/
22 :  #include "sfr_r835a.h" /* R8C/35A SFRの定義ファイル */
23 :
24 :  /*=====*/
25 :  /* シンボル定義 */
26 :  /*=====*/
27 :
28 :  /*=====*/
29 :  /* プロトタイプ宣言 */
30 :  /*=====*/
31 :  void init( void );
32 :  void led_out( unsigned char led );
33 :  void timer( unsigned long timer_set );
34 :
35 :  /*=====*/
36 :  /* グローバル変数の宣言 */

```

```

37 : /*=====*/
38 : unsigned char cnt_int3;          /* INT3割り込みごとに+1 */
39 :
40 : /*=====*/
41 : /* メインプログラム */
42 : /*=====*/
43 : void main( void )
44 : {
45 :     unsigned char d;
46 :
47 :     init();                      /* 初期化 */
48 :     asm(" fset I ");            /* 全体の割り込み許可 */
49 :
50 :     while( 1 ) {
51 :         p6 = 0x55;
52 :         timer( 1000 );
53 :         p6 = 0xaa;
54 :         timer( 1000 );
55 :         p6 = 0x00;
56 :         timer( 1000 );
57 :     }
58 : }
59 :
60 : /*=====*/
61 : /* R8C/35A スペシャルファンクションレジスタ(SFR)の初期化 */
62 : /*=====*/
63 : void init( void )
64 : {
65 :     int i;
66 :
67 :     /* クロックをXINクロック(20MHz)に変更 */
68 :     prc0 = 1;                   /* プロテクト解除 */
69 :     cm13 = 1;                   /* P4_6, P4_7をXIN-XOUT端子にする */
70 :     cm05 = 0;                   /* XINクロック発振 */
71 :     for(i=0; i<50; i++ );       /* 安定するまで少し待つ(約10ms) */
72 :     ocd2 = 0;                   /* システムクロックをXINにする */
73 :     prc0 = 0;                   /* プロテクトON */
74 :
75 :     /* ポートの入出力設定 */
76 :     prc2 = 1;                   /* PD0のプロテクト解除 */
77 :     pd0 = 0xe0;                 /* 7-5:LED 4:MicroSW 3-0:Sensor */
78 :     p1 = 0xf0;                  /* 3-0:LEDは消灯 */
79 :     pd1 = 0xdf;                 /* 5:RXD0 4:TXD0 3-0:LED */
80 :     pd2 = 0xfe;                 /* 0:PushSW */
81 :     pd3 = 0xf3;                 /* 4:Buzzer 3:INT3 2:IR */
82 :     pd4 = 0x80;                 /* 7:XOUT 6:XIN 5-3:DIP SW 2:VREF */
83 :     pd5 = 0x40;                 /* 7:DIP SW */
84 :     pd6 = 0xff;                 /* LEDなど出力 */
85 :
86 :     /* INT0~4割り込み設定(今回はINT3を設定) */
87 :     intsr = 0x00;               /* INT1~3の入力端子設定 */
88 :     inten = 0x40;               /* INT0~3の外部入力許可設定 */
89 :     inten1 = 0x00;              /* INT4の外部入力許可設定 */
90 :     intf = 0xc0;                /* INT0~3の入力フィルタ選択 */
91 :     intf1 = 0x00;               /* INT4の入力フィルタ選択 */
92 :
93 :     int0ic = 0x00;              /* INT0割り込み優先レベル設定 */
94 :     int1ic = 0x00;              /* INT1割り込み優先レベル設定 */
95 :     int2ic = 0x00;              /* INT2割り込み優先レベル設定 */
96 :     int3ic = 0x07;              /* INT3割り込み優先レベル設定 */
97 :     int4ic = 0x00;              /* INT4割り込み優先レベル設定 */
98 : }
99 :
100 : /*=====*/
101 : /* マイコン部のLED出力 */
102 : /* 引数 スイッチ値 0~15 */
103 : /*=====*/
104 : void led_out( unsigned char led )
105 : {
106 :     unsigned char data;
107 :
108 :     led = ~led;
109 :     led &= 0xf0;
110 :     data = p1 & 0xf0;
111 :     p1 = data | led;
112 : }
113 :
114 : /*=====*/
115 : /* タイマ本体 */
116 : /* 引数 タイマ値 1=1ms */
117 : /*=====*/
118 : void timer( unsigned long timer_set )
119 : {
120 :     int i;
121 :
122 :     do {
123 :         for( i=0; i<1240; i++ );
124 :     } while( timer_set-- );
125 : }
126 :
127 : /*=====*/

```

```

128 : /* INT3 割り込み処理 */
129 : /***** */
130 : #pragma interrupt intINT3(vect=26)
131 : void intINT3( void )
132 : {
133 :     cnt_int3++;
134 :     led_out( cnt_int3 );
135 : }
136 :
137 : /***** */
138 : /* end of file */
139 : /***** */

```

16.5 プログラムの解説

今回は、INT3端子を P3_3 に割り当てて使用します。エッジは、立ち下がりエッジとします。

16.5.1 init関数(INT3割り込みの設定)

(1) INT0～INT4割り込み

INT0～INT4割り込みとは、INT0～INT4端子から入力されるエッジによって発生する割り込みのことです。エッジの種類を、下表に示します。

立ち上がりエッジ	"0"→"1"になる瞬間を検出することです。
立ち下がりエッジ	"1"→"0"になる瞬間を検出することです。
両エッジ	立ち上がりエッジと立ち下がりエッジの両方を検出することです。

設定したエッジの信号があると、割り込みが発生します。

(2) INT0～INT4割り込み端子

INT0～INT4割り込み端子に割り当てることのできる端子を下表に示します。

端子名	割り当てることのできる端子
<u>INT0</u> 端子	P4_5 に設定可能です。
<u>INT1</u> 端子	P1_5、P1_7、P2_0、P3_2、P3_6 のどれかに設定可能です。 設定は、INT 割り込み入力端子選択レジスタ(INTSR)で行います。
<u>INT2</u> 端子	P3_2、P6_6 のどれかに設定可能です。 設定は、INT 割り込み入力端子選択レジスタ(INTSR)で行います。
<u>INT3</u> 端子	P3_3、P6_7 のどれかに設定可能です。 設定は、INT 割り込み入力端子選択レジスタ(INTSR)で行います。
<u>INT4</u> 端子	P6_5 に設定可能です。

(3) INT3割り込みの設定

レジスタの設定手順を下記に示します。



①INT 割り込み入力端子選択レジスタ(INTSR:INT function select register)の設定

INT 割り込み入力端子選択レジスタを設定します。 $\overline{\text{INT1}}$ ～ $\overline{\text{INT3}}$ をどの端子に割り当てるか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7,6	$\overline{\text{INT3}}$ 端子選択ビット bit7: int3sel1 bit6: int3sel0	00:P3_3 に割り当てる 01:設定しないでください 10:P6_7 に割り当てる 11:設定しないでください 今回は、P3_3 に割り当てるので"00"に設定します。	00
bit5		"0"を設定	0
bit4	$\overline{\text{INT2}}$ 端子選択ビット int2sel0	0:P6_6 に割り当てる 1:P3_2 に割り当てる 今回、 $\overline{\text{INT2}}$ は使用しませんが"0"を設定しておきます。	0
bit3,2,1	$\overline{\text{INT1}}$ 端子選択ビット bit3: int1sel2 bit2: int1sel1 bit1: int1sel0	000:P1_7 に割り当てる 001:P1_5 に割り当てる 010:P2_0 に割り当てる 011:P3_6 に割り当てる 100:P3_2 に割り当てる 上記以外:設定しないでください 今回、 $\overline{\text{INT1}}$ は使用しませんが"000"を設定しておきます。	000
bit0		"0"を設定	0

INT 割り込み入力端子選択レジスタ(INTSR)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			

②外部入力許可レジスタ 0(INTEN: External interrupt enable register)の設定

外部入力許可レジスタ 0 設定をします。 $\overline{\text{INT0}} \sim \overline{\text{INT3}}$ を許可するか、エッジをどうするか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7	$\overline{\text{INT3}}$ 入力極性選択ビット(注 1、2) int3pl	0:片エッジ 1:両エッジ 今回は、片エッジを選択します。立ち上がりか立ち下 がりかは、INT3IC で設定します。	0
bit6	$\overline{\text{INT3}}$ 入力許可ビット int3en	0:禁止 1:許可 今回は $\overline{\text{INT3}}$ を使いますので、許可します。	1
bit5	$\overline{\text{INT2}}$ 入力極性選択ビット(注 1、2) int2pl	0:片エッジ 1:両エッジ	0
bit4	$\overline{\text{INT2}}$ 入力許可ビット int2en	0:禁止 1:許可 今回は $\overline{\text{INT2}}$ を使いませんので禁止にしておきます。	0
bit3	$\overline{\text{INT1}}$ 入力極性選択ビット(注 1、2) int1pl	0:片エッジ 1:両エッジ	0
bit2	$\overline{\text{INT1}}$ 入力許可ビット int1en	0:禁止 1:許可 今回は $\overline{\text{INT1}}$ を使いませんので禁止にしておきます。	0
bit1	$\overline{\text{INT0}}$ 入力極性選択ビット(注 1、2) int0pl	0:片エッジ 1:両エッジ	0
bit0	$\overline{\text{INT0}}$ 入力許可ビット int0en	0:禁止 1:許可 今回は $\overline{\text{INT0}}$ を使いませんので禁止にしておきます。	0

注 1. INTiPL ビット(i=0~3)を“1”(両エッジ)にする場合、INTiIC レジスタの POL ビットを“0”(立ち下がりエッジを
選択)にしてください。

注 2. INTiPL ビットを変更すると、INTiIC レジスタの IR ビットが“1”(割り込み要求あり)になることがあります。詳し
くはハードウェアマニュアルの「11.8.4 割り込み要因の変更」を参照してください。

外部入力許可レジスタ 0(INTEN)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	0	0	0	0	0
16 進数	4				0			

③外部入力許可レジスタ 1(INTEN1:external input enable register 1)の設定

外部入力許可レジスタ 1 設定をします。 $\overline{\text{INT4}}$ を許可するか、エッジをどうするか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～2		"000000"を設定	0000 00
bit1	$\overline{\text{INT4}}$ 入力極性選択ビット(注 1、2) int4pl	0:片エッジ 1:両エッジ	0
bit0	$\overline{\text{INT4}}$ 入力許可ビット int4en	0:禁止 1:許可 今回は $\overline{\text{INT4}}$ を使いませんので禁止にしておきます。	0

注 1. INT4PL ビットを“1”(両エッジ)にする場合、INT4IC レジスタの POL ビットを“0”(立ち下がりエッジを選択)にしてください。

注 2. INT4PL ビットを変更すると、INT4IC レジスタの IR ビットが“1”(割り込み要求あり)になることがあります。詳しくはハードウェアマニュアルの「11.8.4 割り込み要因の変更」を参照してください。

外部入力許可レジスタ 1(INTEN1)の設定値を下記に示します。

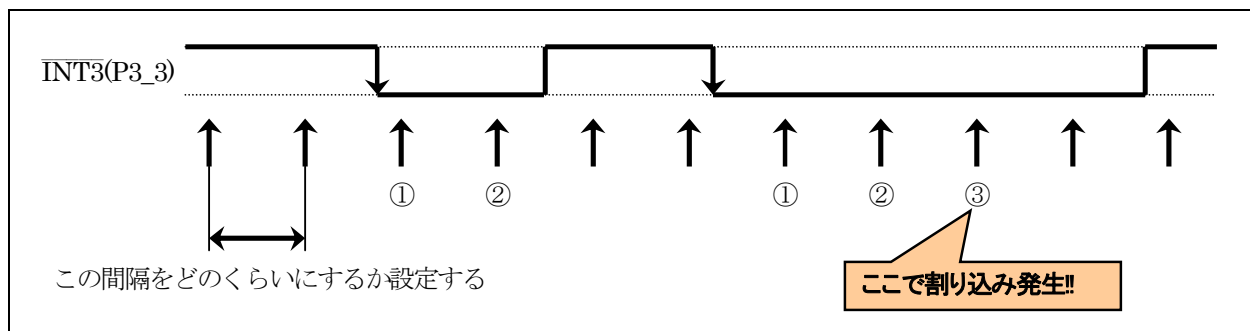
bit	7	6	5	4	3	2	1	0
設定値	0	0	0	0	0	0	0	0
16 進数	0				0			

④INT 入力フィルタ選択レジスタ 0 (INTF:INT0 input filter select register)の設定

INT 入力フィルタ選択レジスタ 0 の設定をします。 $\overline{\text{INT0}} \sim \overline{\text{INT3}}$ のフィルタをどうするか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7,6	$\overline{\text{INT3}}$ 入力フィルタ選択ビット bit7:int3f1 bit6:int3f0	00:フィルタなし 01:フィルタあり、f1(1/20MHz=50ns)でサンプリング 10:フィルタあり、f8(8/20MHz=400ns)でサンプリング 11:フィルタあり、f32(32/20MHz=1600ns)でサンプリング 今回は、f32 でサンプリングします。	11
bit5,4	$\overline{\text{INT2}}$ 入力フィルタ選択ビット bit5:int2f1 bit4:int2f0	00:フィルタなし 01:フィルタあり、f1(1/20MHz=50ns)でサンプリング 10:フィルタあり、f8(8/20MHz=400ns)でサンプリング 11:フィルタあり、f32(32/20MHz=1600ns)でサンプリング	00
bit3,2	$\overline{\text{INT1}}$ 入力フィルタ選択ビット bit3:int1f1 bit2:int1f0	00:フィルタなし 01:フィルタあり、f1(1/20MHz=50ns)でサンプリング 10:フィルタあり、f8(8/20MHz=400ns)でサンプリング 11:フィルタあり、f32(32/20MHz=1600ns)でサンプリング	00
bit1,0	$\overline{\text{INT0}}$ 入力フィルタ選択ビット bit1:int0f1 bit0:int0f0	00:フィルタなし 01:フィルタあり、f1(1/20MHz=50ns)でサンプリング 10:フィルタあり、f8(8/20MHz=400ns)でサンプリング 11:フィルタあり、f32(32/20MHz=1600ns)でサンプリング	00

入力フィルタとは、設定した間隔で信号を読み込んで、3 回一致した時点で割り込みを発生させる機能です。



今回、「f32でサンプリング」にしました。これは、1600nsごとに読み込んで3回連続して“0”なら、信号が“0”になったと判断して割り込みを発生させます。よって、立ち下がりエッジから約 4800ns(1600ns×3)後に割り込みがかかります。

INT 入力フィルタ選択レジスタ 0 (INTF)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	0	0	0	0	0	0
16 進数	C				0			

⑤INT 入力フィルタ選択レジスタ 1 (INTF1:INT input filter select register 1)の設定

INT 入力フィルタ選択レジスタ 1 の設定をします。 $\overline{\text{INT4}}$ のフィルタをどうするか設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～2		"000000"を設定	000000
bit1,0	$\overline{\text{INT4}}$ 入力フィルタ選択ビット bit1: int4f1 bit0: int4f0	00:フィルタなし 01:フィルタあり、f1(1/20MHz=50ns)でサンプリング 10:フィルタあり、f8(8/20MHz=400ns)でサンプリング 11:フィルタあり、f32(32/20MHz=1600ns)でサンプリング	00

INT 入力フィルタ選択レジスタ 1 (INTF1)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			

⑨INT3割り込み制御レジスタ(INT3IC:INT3 interrupt control register)の設定

INT3割り込み制御レジスタの設定をします。INT3の割り込み優先レベル、極性切り替えなどを設定します。

設定 bit	上:ビット名 下:シンボル	内容	今回の 内容
bit7～5		"000"を設定	000
bit4	極性切り替えビット(注 3) pol_int1ic	0:立ち下がりエッジを選択 1:立ち上がりエッジを選択(注 2) 立ち下がりエッジを選択します。	0
bit3	割り込み要求ビット(注 1) ir_int1ic	0:割り込み要求なし 1:割り込み要求あり 書き込みは"0"のみです。	0
bit2～0	割り込み優先レベル選択ビット bit2:ilvl2_int1ic bit1:ilvl1_int1ic bit0:ilvl0_int1ic	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、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、どちらを優先させるか決めて設定します。今回はINT3割り込みだけなのでレベル 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

注 1. IR ビットは"0"のみ書けます("1"を書かないでください)。

注 2. INTEN レジスタの INTiPL ビットが"1"(両エッジ)の場合、POL ビットを"0"(立ち下がりエッジを選択)にしてください。

注 3. POL ビットを変更すると、IR ビットが"1"(割り込み要求あり)になることがあります。詳しくは、ハードウェアマニュアルの「11.8.4 割り込み要因の変更」を参照してください。

割り込み制御レジスタの変更は、そのレジスタに対応する割り込み要求が発生しない箇所で行ってください。詳しくは、ハードウェアマニュアルの「11.8.5 割り込み制御レジスタの変更」を参照してください。

INT3割り込み制御レジスタ(INT3IC)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			

- ⑥ $\overline{\text{INT0}}$ 割り込み制御レジスタ(INT0IC)の設定
 ⑦ $\overline{\text{INT1}}$ 割り込み制御レジスタ(INT1IC)の設定
 ⑧ $\overline{\text{INT2}}$ 割り込み制御レジスタ(INT2IC)の設定
 ⑩ $\overline{\text{INT4}}$ 割り込み制御レジスタ(INT4IC)の設定

$\overline{\text{INT0}}$ 、 $\overline{\text{INT1}}$ 、 $\overline{\text{INT2}}$ 、 $\overline{\text{INT4}}$ 割り込み制御レジスタの設定をします。設定内容は INT3IC と同じです。今回は $\overline{\text{INT0}}$ 、 $\overline{\text{INT1}}$ 、 $\overline{\text{INT2}}$ 、 $\overline{\text{INT4}}$ は使いませんので、レベルは 0(割り込み禁止)に設定します。

$\overline{\text{INT0}}$ 割り込み制御レジスタ(INT0IC)、 $\overline{\text{INT1}}$ 割り込み制御レジスタ(INT1IC)、 $\overline{\text{INT2}}$ 割り込み制御レジスタ(INT2IC)、 $\overline{\text{INT4}}$ 割り込み制御レジスタ(INT4IC)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			

16.5.2 intINT3 関数 ($\overline{\text{INT3}}$ 端子に立ち下がりエッジの信号が入力されたときに実行される関数)

先の設定で、 $\overline{\text{INT3}}$ 端子(P3_3)に立ち下がりエッジの信号が入力されると、割り込みを発生させる設定にしました。intINT3 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。

```

127 :  /*****
128 :  /* INT3 割り込み処理                                     */
129 :  *****/
130 :  #pragma interrupt intINT3(vect=26)
131 :  void intINT3( void )
132 :  {
133 :      cnt_int3++;
134 :      led_out( cnt_int3 );
135 :  }
```

130 行	#pragma interrupt <u>割り込み処理関数名</u> (vect= <u>ソフトウェア割り込み番号</u>) とすることで、 <u>ソフトウェア割り込み番号</u> の割り込みが発生したとき、 <u>割り込み処理関数名</u> を実行します。 ソフトウェア割り込み番号の表より、 $\overline{\text{INT3}}$ 割り込みは、26 番です。 よって、26 番の割り込みが発生したときに intINT3 関数を実行するよう、「#pragma interrupt」で設定します。
131 行	$\overline{\text{INT3}}$ 割り込みにより実行する関数です。今回の設定では、P3_3 の立ち下がりエッジ信号ごとに実行されます。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。
133 行	変数 cnt_int3 を+1 します。
134 行	マイコンボード上の LED へ変数 cnt_int3 の値を出力します。

※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. SSUICSR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、INT3を使用して割り込みを発生させるので、表より番号は26番となります。

16.5.3 main関数

```

40 :  /*****
41 :  /* メインプログラム
42 :  *****/
43 :  void main( void )
44 :  {
45 :      unsigned char d;
46 :
47 :      init();                /* 初期化
48 :      asm( " fset I ");      /* 全体の割り込み許可
49 :
50 :      while( 1 ) {
51 :          p6 = 0x55;
52 :          timer( 1000 );
53 :          p6 = 0xaa;
54 :          timer( 1000 );
55 :          p6 = 0x00;
56 :          timer( 1000 );
57 :      }
58 :  }

```

48 行	全体の割り込みを許可する命令です。 init 関数内でINT3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。
50 行 ～ 57 行	プロジェクト「timer1」と同じプログラムです。プロジェクト「timer1」の解説を参照してください。

16.6 演習

- (1) $\overline{\text{INT3}}$ 端子に立ち上がりエッジの信号が入力されると、割り込みが発生するようにしなさい。
- (2) $\overline{\text{INT1}}$ 割り込みを P3.2 端子に設定して、立ち下がりエッジで割り込みがかかるようにしなさい。割り込みプログラムはサンプルプログラムと同じとする。
(実習基板の SW5 は、左から「下上下上」にしてください)