

```

1:  /*****
2:  /*  割込み動作確認サンプルプログラム
3:
4:  #include "sfr_r815.h"
5:
6:  void set_MainCLK(void);
7:  void set_int3(void);
8:  void set_LED(void);
9:
10: void main(void)
11: {
12:     long count;
13:     set_MainCLK();
14:     set_LED();
15:     set_int3();
16:     asm( "FSET I" );
17:     while( 1 ){
18:         p1_2=~p1_2;
19:         count = 50000;
20:         while( count-- );
21:     }
22: }
23:
24: void set_MainCLK(void)
25: {
26:     prc0=1;
27:     cm13=1;
28:     cm05=0;
29:     cm06=0;
30:     asm( "nop" );
31:     asm( "nop" );
32:     asm( "nop" );
33:     asm( "nop" );
34:     ocd2=0;
35:     prc0=0;
36: }
37:
38: void set_int3(void)
39: {
40:     tcc03=1;
41:     tcc06=1;
42:     tcc10=1;
43:     tcc11=1;
44:     ilv10_int3ic=1;
45:     ir_int3ic=0;
46: }
47:
48: void set_LED(void)
49: {
50:     drr1=1;
51:     drr2=1;
52:     pd1_1=1;
53:     pd1_2=1;
54:     p1_1=0;
55:     p1_2=0;
56: }
57:
58: #pragma interrupt int3_int( vect=26 )
59: void int3_int(void)
60: {
61:     p1_1 = ~p1_1;
62: }

```

SFR 定義ファイルのインクルード (p1_1 等の定義)

関数のプロトタイプ宣言
関数 int3_int0は呼び出しがないのでプロトタイプ宣言がなくてもよい

main 関数は P218 リスト 19-3

初期設定
asm はアセンブラ命令の実行. FSET I はアセンブラ命令で割込み許可 (P207 図 19-5 の FLG レジスタの I ビットを 1 にセット)

無限ループ
このループで p1_2 の LED を点滅させる
~はビット反転
while(count--) はループ周期を遅くするためのダミーのループ

クロックを 20MHz (試験には出ません)
解説は P199 リスト 18-2

INT3 割込みに関する初期設定 (解説は P217 リスト 19-1)
・ tcc03=1 として立ち下がりエッジとしているのは、スイッチを押した時に電圧が 5V から 0V へ立ち下がるため
・ レジスタの解説は P214 図 19-15, 19-16 参照
・ フィルタが f32 でサンプリングとは、20/32MHz でサンプリングして 3 回連続して 0 が入力されたら 0 に立ち下がったと判断するということ
・ ilv10_int3ic=1 としてレベルを 1 にする (P208 図 19-6 の ILVL と P207 図 19-5 の IPL の比較が 1 > 0 なので割込み許可される)

I/O ポート(p1_1 と p1_2)の初期設定 (P217 リスト 19-2 を変更)
drr を 1 にすることで大電流駆動として LED を点灯可能にする
pd を 1 としてポートを出力に設定
初期値を 0 とすることで最初は LED を点灯とする

この設定によって関数 int3_int が int3 割込み処理関数となる
vect=26 は int3 割込み (P207 表 19-2 参照)

割込みが発生すると p1_1 を反転する
(P219 のリスト 19-4 を変更)

課題

1. 裏面のサンプルプログラムの動作を確認しましょう。

(サンプルでは、1つのLEDがmain関数の無限ループによって点滅し、もうひとつのLEDがスイッチを押すと割込みによって反転するはずです)

2. サンプルプログラムの42, 43行目を

```
tcc10=0;
```

```
tcc11=0;
```

に変更して動作を確認する。動作が不安定となる原因を考える。

3. サンプルを変更し、INT3割込みによって、スイッチを押すたびに点滅しているLEDを変更する(点滅していない方のLEDは消灯とする)(グローバルスコープをもつ変数(関数の外で宣言(プロトタイプ宣言の後あたり))を作成し、割込み処理で変数の値を変える。この変数を見て、main関数の無限ループ内で点滅するLEDを判断するとよいでしょう。)

4. サンプルを変更し、INT3割込みによって、スイッチを押すたびにLEDの点滅速度が変わるようにする(たとえば 低速→中速→高速→低速→・・・とスイッチを押すたびに切り替わる)

5. その他、割込みを使用したプログラム(LEDやブザーを制御)を作成してみる。