９．割り込みを学ぼう

9.1 外部からの割り込み(SW1を押すことにより割り込みをかける方法）

9.2 タイマ0による割り込み（処理タイミングの管理方法：一定時間毎にLED1, 2, 3を点滅させる方法）

回路製作の詳細は第０章を参照してください．

9.1 外部からの割り込み(SW1を押すことにより割り込みをかける方法)

このソースファイルを打ち込んで下さい．詳細説明はp.11～
注意 CONFIGの前のアンダーバーは2つあります．

SW1を押すと割り込み信号が入り，メインプログラムの処理(LED1を点灯，LED3を消灯)を中断して割り込みプログラムを実行(LED1を消灯，LED3を点灯)し，一定時間後に割込処理を終了してメインプログラムの処理を再開するプログラムです．

```
;Interrupt test program

            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

Memory      EQU       0x0C
WORK1       EQU       Memory+1    ;WORk1 at 0C
TIME1       EQU       Memory+2    ;TIME1 at 0D
TIME2       EQU       Memory+3    ;TIME2 at 0E
TIME3       EQU       Memory+4    ;TIME3 at 0F

            ORG       0
            GOTO      START                 ;Main Program starts at START

            ORG       4                     ;Sub Program
            MOVWF     WORK1                 ;Save the content of Working Register to WORK1
            MOVLW     B'00010000'           ;'00010000' -> (W)
            MOVWF     INTCON                ;(W) -> (INTCON), Inhibit another interruput
            CALL      SUB1                  ;SUB1 call
            MOVLW     B'10010000'           ;'10010000' -> (W)
            MOVWF     INTCON                ;(W)->(INTCON), Enable interrupt
            MOVF      WORK1, 0              ;(WORK1) -> (W)
            RETFIE                          ;Return from interrupt
START
            BSF       STATUS, RP0           ;Select Bank1
            MOVLW     B'00000111'           ;'00000111' -> (W)
            MOVWF     TRISB                 ;RB0-2: Input port; RB3-7: Output port
            MOVLW     B'10010000'           ;'10010000' -> (W)
            MOVWF     INTCON                ;(W) -> Intcon, Enable interrput
            MOVLW     B'00000000'
            MOVWF     OPTION_REG
            BCF       STATUS, RP0           ;Select Bank0
```

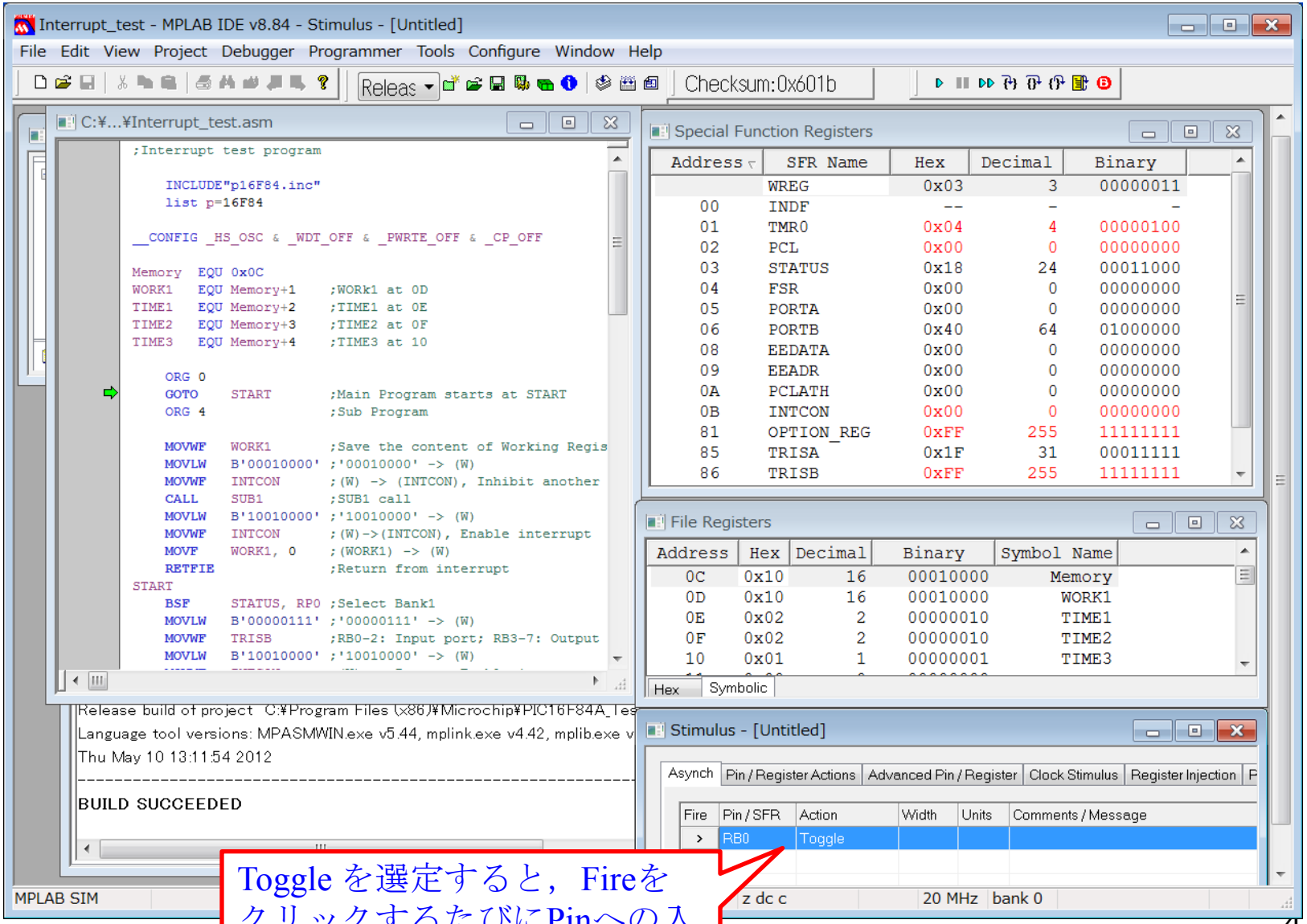
割り込みプログラム

ソースファイル（続き）

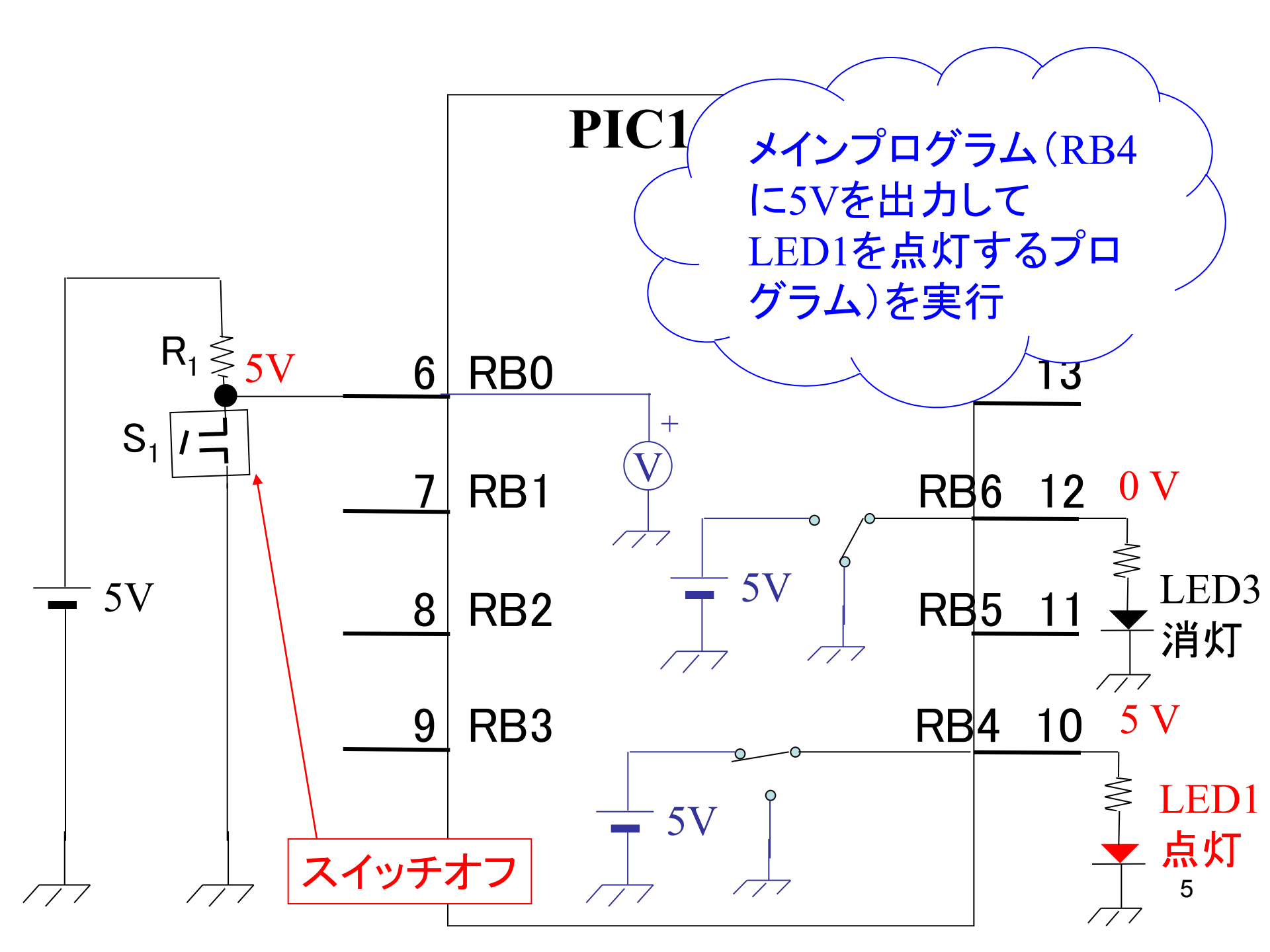
```
;Main Program

            MOVLW     B'00010000'          ;'00010000' -> (W)
STEP1       MOVWF     PORTB                ;(W) -> (PORTB), LED1 on
            GOTO      STEP1


;Sub Program
SUB1        MOVLW     B'01000000'          ;'01000000' -> (W)
            MOVWF     PORTB                ;(W) -> (PORTB), LED3 on
            CALL      COUNT1
            RETURN

COUNT1      MOVLW     0x80
            MOVWF     TIME1
STEPM       MOVWF     TIME2
STEPM1      MOVWF     TIME3
STEPM2      DECFSZ    TIME3,1
            GOTO      STEPM2
            DECFSZ    TIME2,1
            GOTO      STEPM1
            DECFSZ    TIME1,1
            GOTO      STEPM
            RETURN

            END
```

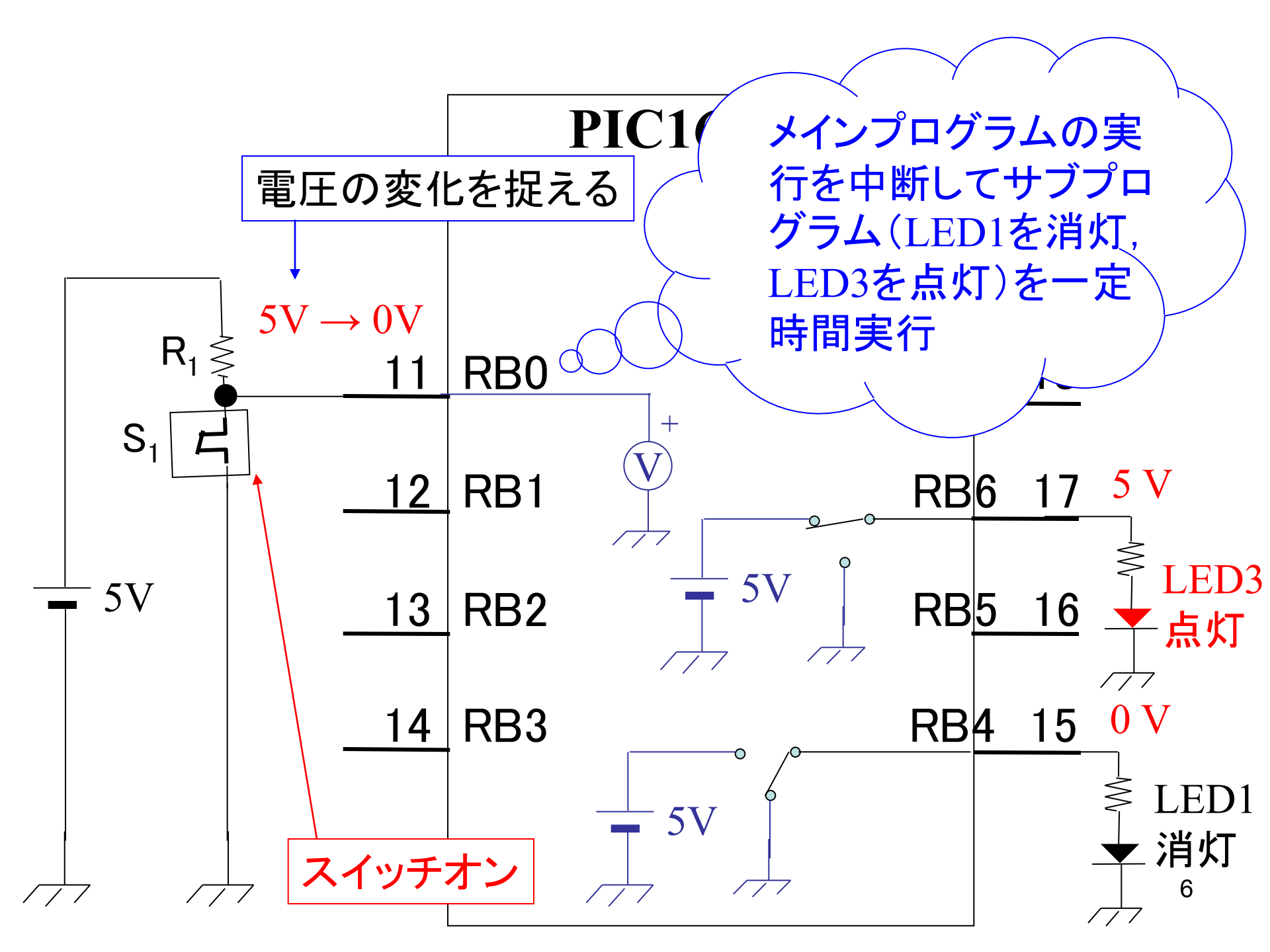
デモプログラムでは80となっていますが，シミュレーションではくり返し回数が多すぎるので0x03を入力してみて下さい．

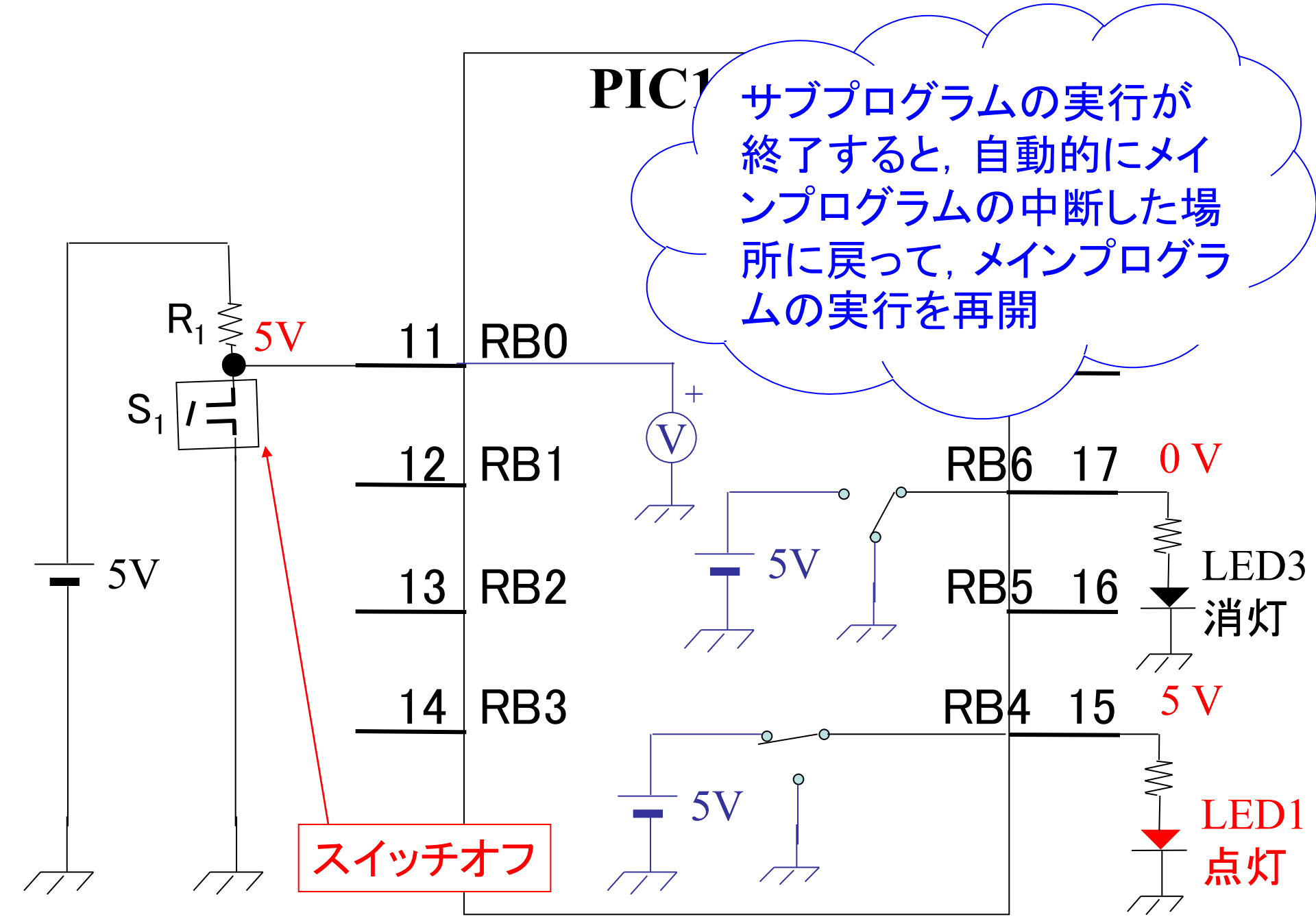
Debugger → Select Tool → MPLAB SIM → Make → View → Special Function Registers → View → File Registers → Stimulus → New Workbook → RB0+Toggle → F7 を押し続けながら ときどき RB0をFire → ・・・



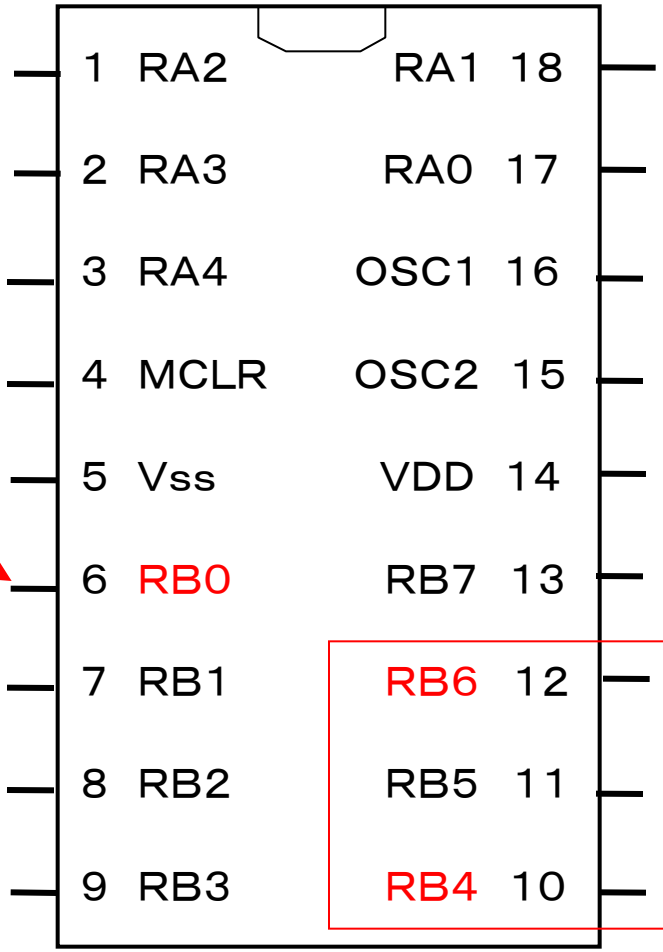
Toggle を選定すると，Fireをクリックするたびにに Pinへの入力が反転する．


PIC1
メインプログラム（RB4に5Vを出力してLED1を点灯するプログラム）を実行
R1
5V
S1
6 RB0
7 RB1
8 RB2
9 RB3
13
RB6 12
RB5 11
RB4 10
V
+
5V
5V
5V
0 V
5 V
LED3
消灯
LED1
点灯
スイッチオフ


PIC1
電圧の変化を捉える
メインプログラムの実行を中断してサブプログラム（LED1を消灯，LED3を点灯）を一定時間実行
5V → 0V
R1
S1
11 RB0
12 RB1
13 RB2
14 RB3
RB6 17
RB5 16
RB4 15
5V
5V
5V
V
+
5 V
0 V
LED3
点灯
LED1
消灯
スイッチオン


PIC1
サブプログラムの実行が終了すると，自動的にメインプログラムの中断した場所に戻って，メインプログラムの実行を再開
R1
5V
S1
11 RB0
12 RB1
13 RB2
14 RB3
RB6 17
RB5 16
RB4 15
0 V
5 V
LED3
消灯
LED1
点灯
V
5V
5V
5V
スイッチオフ

PIC16F84



割り込み入力用端子

出力端子として利用

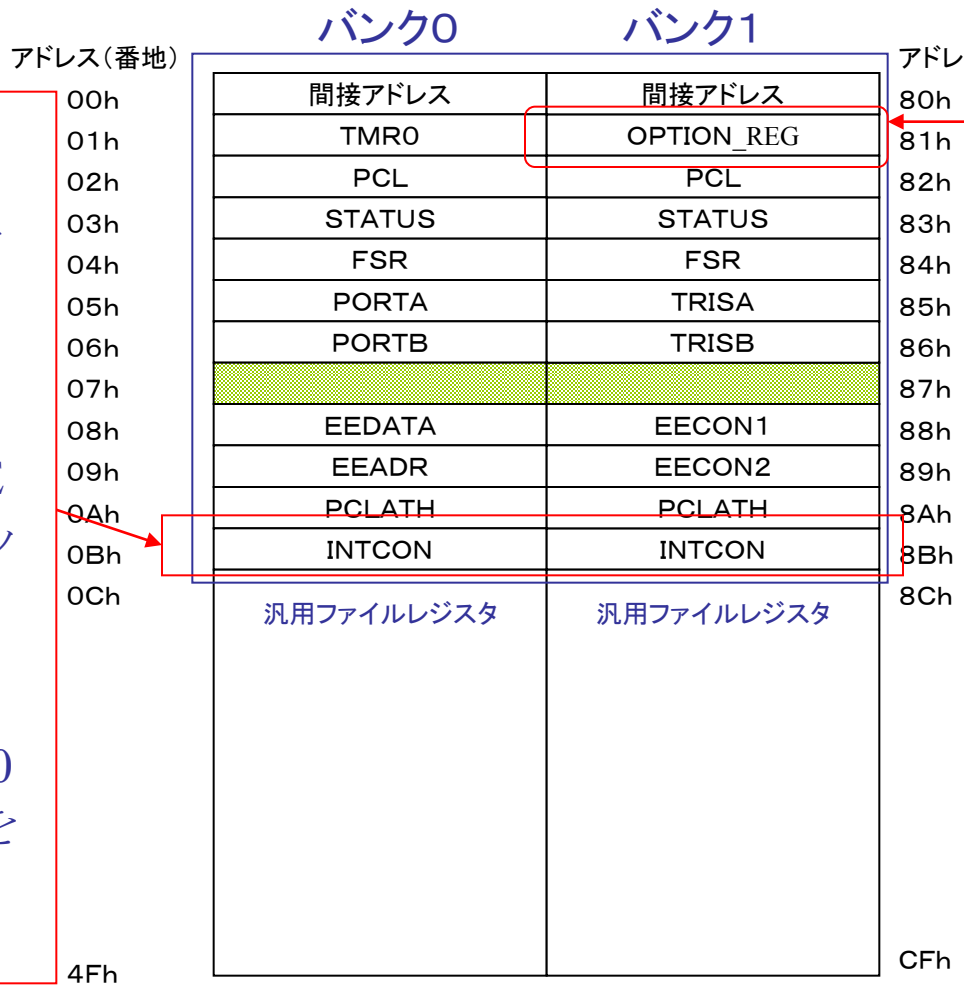
LEDを点灯させる.

本章のポイント

INTCONレジスタ

B’10010000’
を書き込む
７ビット目：GIE
割り込み許可ビット
1:許可，0:禁止

４ビット目：RB0
からの割り込みを
許可するビット
1:**許可**，0:**禁止**



本章のポイント

OPTIONレジスタ

B’00000000’を書き込む

6ビット目：割り込みエッジ選択ビット
1:RB0の電圧が0 V→5 Vに変化したとき割り込みをかける.
0: RB0の電圧が5 V→0 Vに変化したとき割り込みをかける.

ファイルレジスタの配置

特殊レジスタ一覧

アドレス	名　称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バンク0									
00h	INDF	FSRの内容のアドレスのデータメモリ（物理的には存在しない）							
01h	TMR0	8ビットリアルタイム・クロック／カウンタ							
02h	PCL	プログラムカウンタ（PC）の下位8ビット							
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	－	－	－	RA4/T0CKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない，「0」としてリードされる							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEPROMアドレスレジスタ							
0Ah	PCLATH	－	－	－	PCの上位5ビットへの書き込みバッファ				
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
バンク1									
80h	INDF	FSRの内容のアドレスのデータメモリ（物理的には存在しない）							
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	プログラムカウンタ（PC）の下位8ビット							
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
84h	FSR	間接データメモリアドレスポインタ							
85h	TRISA	－	－	－	PORTAデータ入出力設定レジスタ				
86h	TRISB	PORTBデータ入出力設定レジスタ							
87h		使用しない，「0」としてリードされる							
88h	EECON1	－	－	－	EEIF	WRERR	WREN	WR	RD
89h	EECON2	EEPROM制御レジスタ2（物理的には存在しない）							
0Ah	PCLATH	－	－	－	PCの上位5ビットへの書き込みバッファ				
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

：バンク０，１で共通

```
;Interrupt test program

            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

Memory      EQU       0x0C
WORK1       EQU       Memory+1    ;WORk1 at 0C
TIME1       EQU       Memory+2    ;TIME1 at 0D
TIME2       EQU       Memory+3    ;TIME2 at 0E
TIME3       EQU       Memory+4    ;TIME3 at 0F

            ORG       0           ; 電源が入るとマイコンはこの番地からプログラム実行を開始する.
            GOTO      START       ;  ただちにSTART番地にジャンプする.

            ORG       4
            MOVWF     WORK1
            MOVLW     B'00010000'
            MOVWF     INTCON
            CALL      SUB1
            MOVLW     B'10010000'
            MOVWF     INTCON
            MOVF      WORK1, 0
            RETFIE

START

            BSF       STATUS, RP0
            MOVLW     B'00000111'
            MOVWF     TRISB
            MOVLW     B'10010000'
            MOVWF     INTCON
            MOVLW     B'00000000'
            MOVWF     OPTION_REG
            BCF       STATUS, RP0
```

p.9ファイルレジスタによるとTRISBがBank 1にあるので，Bank 1を選定している．
Bankの選定は，データシートのSTATUS REGISTERによると，
RP0 = 1　Bank1
　　= 0　Bank0
である．なお，STATUSレジスタは2バンクに共通している．

```
;Interrupt test program

            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

Memory      EQU       0x0C
WORK1       EQU       Memory+1    ;WORk1 at 0C
TIME1       EQU       Memory+2    ;TIME1 at 0D
TIME2       EQU       Memory+3    ;TIME2 at 0E
TIME3       EQU       Memory+4    ;TIME3 at 0F

            ORG       0
            GOTO      START

            ORG       4
            MOVWF     WORK1
            MOVLW     B'00010000'
            MOVWF     INTCON
            CALL      SUB1
            MOVLW     B'10010000'
            MOVWF     INTCON
            MOVF      WORK1, 0
            RETFIE

START

            BSF       STATUS, RP0
            MOVLW     B'00000111'
            MOVWF     TRISB
            MOVLW     B'10010000'
            MOVWF     INTCON
            MOVLW     B'00000000'
            MOVWF     OPTION_REG
            BCF       STATUS, RP0
```

PORT Bの設定

RB0-RB2: 入力ポート

RB3-RB7:出力ポート

```
;Interrupt test program

            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

Memory      EQU       0x0C
WORK1       EQU       Memory+1    ;WORk1 at 0C
TIME1       EQU       Memory+2    ;TIME1 at 0D
TIME2       EQU       Memory+3    ;TIME2 at 0E
TIME3       EQU       Memory+4    ;TIME3 at 0F

            ORG       0
            GOTO      START

            ORG       4
            MOVWF     WORK1
            MOVLW     B'00010000'
            MOVWF     INTCON
            CALL      SUB1
            MOVLW     B'10010000'
            MOVWF     INTCON
            MOVF      WORK1, 0
            RETFIE

START

            BSF       STATUS, RP0
            MOVLW     B'00000111'
            MOVWF     TRISB
            MOVLW     B'10010000'
            MOVWF     INTCON
            MOVLW     B'00000000'
            MOVWF     OPTION_REG
            BCF       STATUS, RP0
```

B’10010000’をINTCON (Interrupt Control)レジスタに書き込む

7ビット目：割り込み許可ビット
1:許可，0:禁止

4ビット目：RB0からの割り込みを許可するビット
1:許可，0:禁止

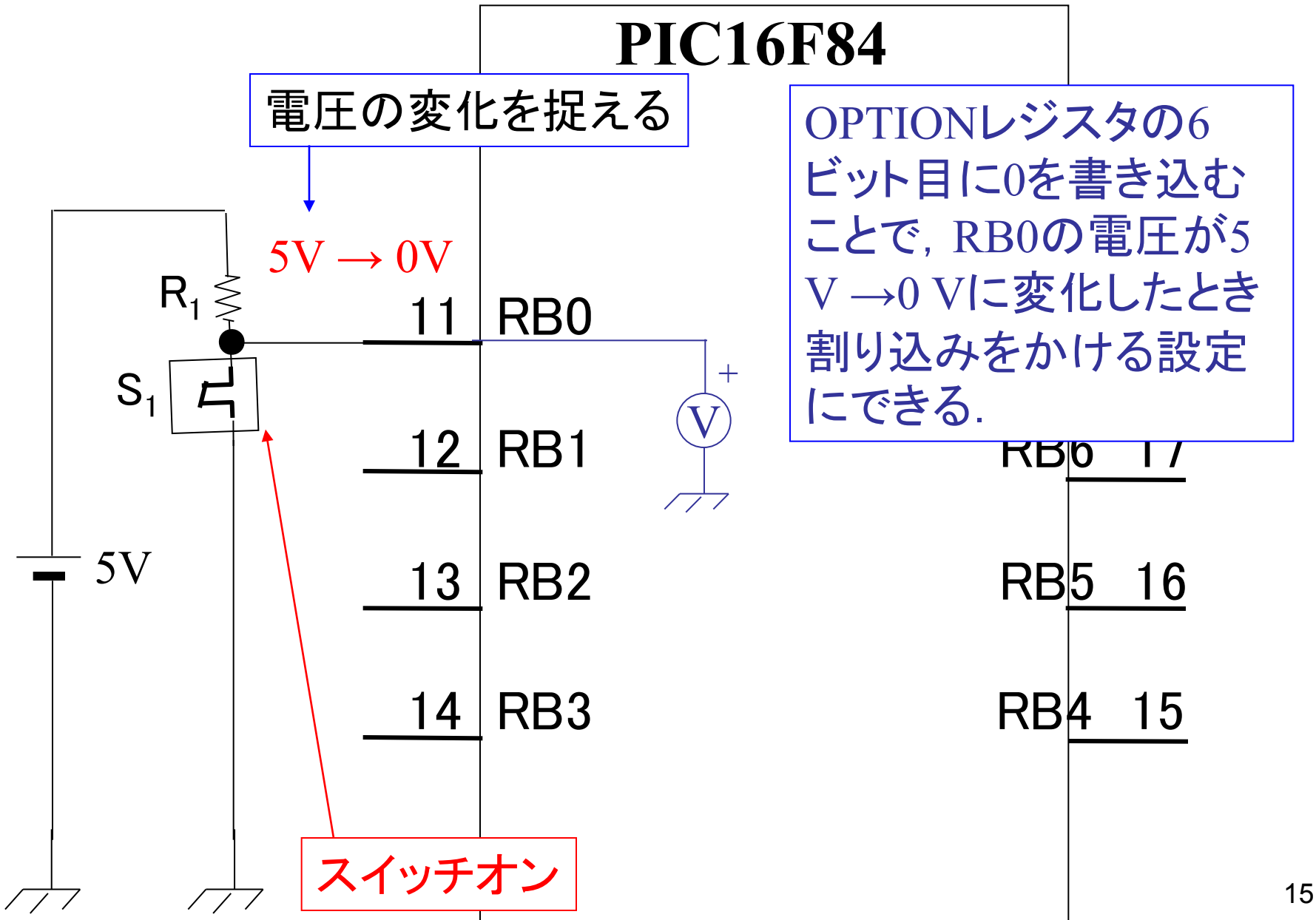
```
;Interrupt test program

            INCLUDE"p16F84.inc"
            list p=16F84

            __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

Memory      EQU         0x0C
WORK1       EQU         Memory+1    ;WORk1 at 0C
TIME1       EQU         Memory+2    ;TIME1 at 0D
TIME2       EQU         Memory+3    ;TIME2 at 0E
TIME3       EQU         Memory+4    ;TIME3 at 0F

            ORG         0
            GOTO        START

            ORG         4
            MOVWF       WORK1
            MOVLW       B'00010000'
            MOVWF       INTCON
            CALL        SUB1
            MOVLW       B'10010000'
            MOVWF       INTCON
            MOVF        WORK1, 0
            RETFIE

START

            BSF         STATUS, RP0
            MOVLW       B'00000111'
            MOVWF       TRISB
            MOVLW       B'10010000'
            MOVWF       INTCON
            MOVLW       B'00000000'
            MOVWF       OPTION_REG
            BCF         STATUS, RP0
```

OPTIONレジスタに　B’00000000’を書き込む

6ビット目 : 割り込みエッジ選択ビット

1:RB0の電圧が0 V →5 Vに変化したとき割り込みをかける.

0: RB0の電圧が5 V →0 Vに変化したとき割り込みをかける


PIC16F84
電圧の変化を捉える
OPTIONレジスタの6ビット目に0を書き込むことで，RB0の電圧が5 V →0 Vに変化したとき割り込みをかける設定にできる.
5V → 0V
R1
S1
5V
11 RB0
12 RB1
13 RB2
14 RB3
RB6 17
RB5 16
RB4 15
V
+
スイッチオン

ソースファイル（続き）

```
;Main Program

        MOVLW   B'00010000'
STEP1   MOVWF   PORTB
        GOTO    STEP1
```

メインプログラム
LED1を点灯するという信号を
PORTBに出し続けるプログラム

このメインプログラムを実行中にRB0に割り込み信号が入ると，そのときのプログラムカウンタの値がスタックに格納され，プログラムカウンタには0004番地が書き込まれて，4番地から書かれているプログラムが実行される．

```
ORG     4
MOVWF   WORK1
MOVLW   B'00010000'
MOVWF   INTCON
CALL    SUB1
MOVLW   B'10010000'
MOVWF   INTCON
MOVF    WORK1, 0
RETFIE
```

割り込みプログラム.

RB0に割り込み信号が入ると，そのときのプログラムカウンタの値がスタックに格納され，プログラムカウンタには0004番地が書き込まれて，４番地から書かれているこのプログラムが実行される.

RETFIE（割り込みからのReturn）が実行されると，スタックに格納した番地がプログラムカウンタに戻され，割り込みがかかった時のメインプログラムに戻る.

割り込み時のメインプログラムのデータの退避場所.

```
WORK1       EQU         Memory+1    ;WORk1 at 0C
```

```
            ORG         4
            MOVWF       WORK1
            MOVLW       B'00010000'
            MOVWF       INTCON
            CALL        SUB1
            MOVLW       B'10010000'
            MOVWF       INTCON
            MOVF        WORK1, 0
            RETFIE
```

Wレジスタの内容をWORK1に退避させる. サブプログラムもWレジスタを使用するので, メインプログラムで実行していた値を退避させておかないと, 書き変えられてしまうため.

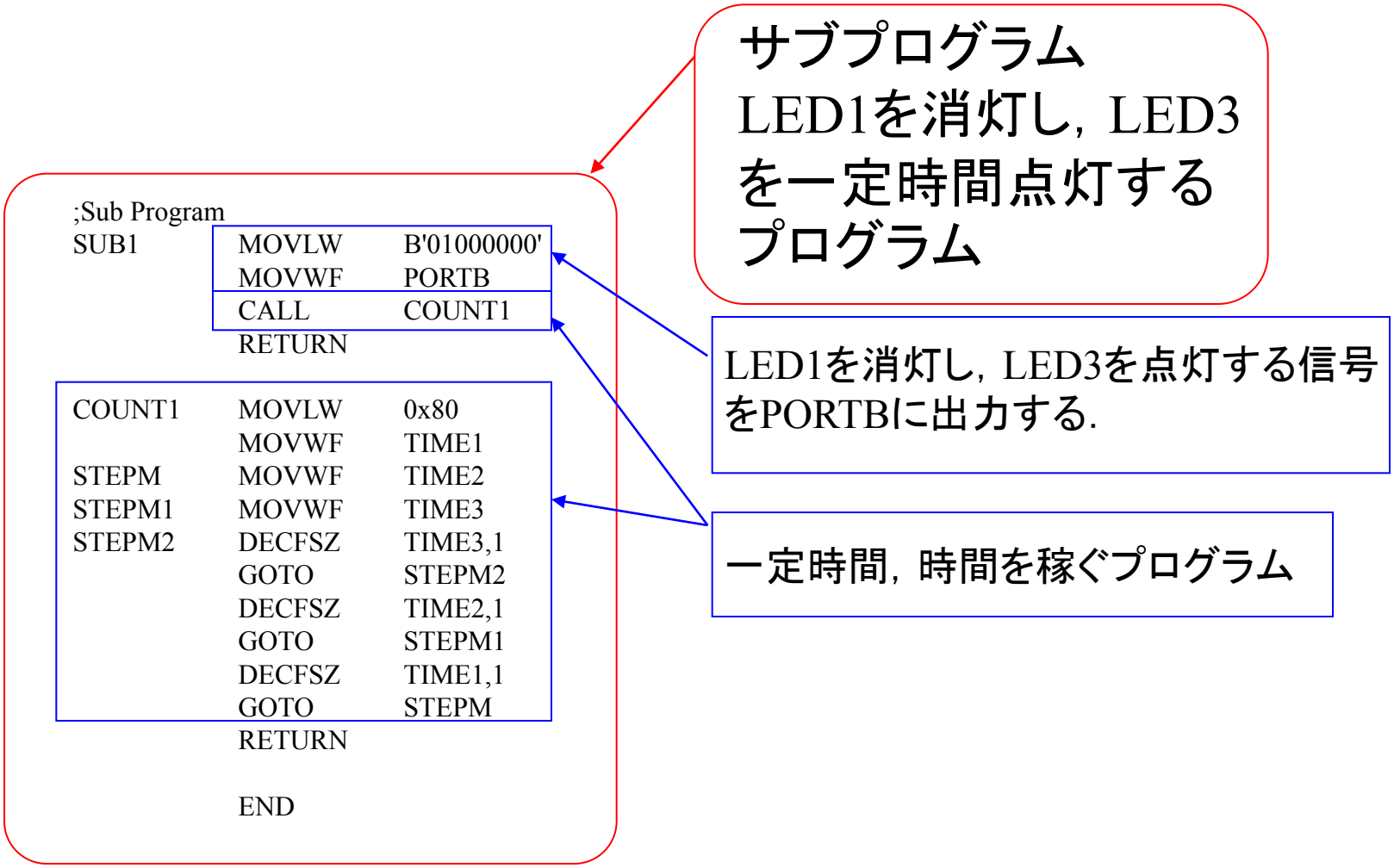
この他, Wレジスタに限らず, サブプログラムに書き換えられては困るものがある場合は, WORK2, WORK3等をファイルレジスタに定義しておいて, 割り込みがかかったときには, ここに退避させるようにする.

割り込み処理の終了時には, 退避させていた, 値を元のWレジスタにもどしてから, メインプログラムに帰っていく.

```
ORG         4
MOVWF       WORK1
MOVLW       B'00010000'
MOVWF       INTCON
CALL        SUB1
MOVLW       B'10010000'
MOVWF       INTCON
MOVF        WORK1, 0
RETFIE
```

INTCONレジスタの7ビット目を0とすることで，割り込み処理中に，RB0に割り込み信号が入っても，割り込み処理に割り込み処理が入らないようにする.

サブプログラムをコール.


;Sub Program
SUB1 MOVLW B'01000000'
MOVWF PORTB
CALL COUNT1
RETURN
COUNT1 MOVLW 0x80
MOVWF TIME1
STEPM MOVWF TIME2
STEPM1 MOVWF TIME3
STEPM2 DECFSZ TIME3,1
GOTO STEPM2
DECFSZ TIME2,1
GOTO STEPM1
DECFSZ TIME1,1
GOTO STEPM
RETURN
END
サブプログラム
LED1を消灯し，LED3を一定時間点灯するプログラム
LED1を消灯し，LED3を点灯する信号をPORTBに出力する.
一定時間，時間を稼ぐプログラム

ソースファイル（続き）

```
;Main Program

                MOVLW     B'00010000'
STEP1           MOVWF     PORTB
                GOTO      STEP1


;Sub Program
SUB1            MOVLW     B'01000000'
                MOVWF     PORTB
                CALL      COUNT1
                RETURN

COUNT1          MOVLW     0x80
                MOVWF     TIME1
STEPM           MOVWF     TIME2
STEPM1          MOVWF     TIME3
STEPM2          DECFSZ    TIME3,1
                GOTO      STEPM2
                DECFSZ    TIME2,1
                GOTO      STEPM1
                DECFSZ    TIME1,1
                GOTO      STEPM
                RETURN

                END
```

サブプログラム
LED1を消灯し，LED3を一定時間点灯するプログラム

初めにTIME1,2,3に0x80を入れる．TIME3を一つずつ減らしていき，0になったら，TIME2を一つ減らして，TIME3に0x80を入れて，再び，TIME3を一つずつ減らしていき，0になったら，TIME2を一つ減らして，TIME3に0x80を入れて，再び‥‥と，くり返し，やがて，TIME2が0になったら，TIME1を一つ減らしてTIME2, 3に0x80を入れて‥‥‥とくり返す．TIME1が0になったら終了．
全部で80×80×80回のくり返し演算を行う．

```
ORG         4
MOVWF       WORK1
MOVLW       B'00010000'
MOVWF       INTCON
CALL        SUB1
MOVLW       B'10010000'
MOVWF       INTCON
MOVF        WORK1, 0
RETFIE
```

割り込み処理が終了したので，INTCONレジスタの7ビット目を1として，割り込み処理を可として，メインプログラムに戻るようにする．

割り込み処理の終了時には，退避させていた，値を元のWレジスタにもどしてから，メインプログラムに帰っていく．

9.2 タイマ0による割り込み

本節では，一定時間毎にLED1, 2, 3を点滅させるプログラムを紹介します．

LEDの点滅に限らず，決まった時間間隔で何らかの処理をさせたい場合に必須の方法です．

例）

ステッピングモータの制御周期管理（第8α章）

データのサンプリング周期管理

（モータドライブノート第１章　1.3.4項）

etc.

9.2 タイマ0による割り込み（処理タイミングの管理方法：一定時間毎にLED1, 2, 3を点滅させる方法）

このソースファイルを打ち込んで下さい．詳細説明はp.26から

```
;タイマ0による割り込みプログラム
                    INCLUDE"P16F84A.INC"
                    list p=16F84A

                    __CONFIG  _HS_OSC  &  _WDT_OFF  &  _PWRTE_OFF  &  _CP_OFF

WORK1       EQU     0x0C                ;WORk1 at 0D

            ORG     0                   ;電源が入るとこの番地からプログラム実行を開始する.
            GOTO    START

            ORG     4                   ;割り込みがかかるとこの番地から開始する.
            MOVWF   WORK1               ;wレジスタの内容をWORK1に退避させる
            CALL    Timer0_interrupt
            MOVF    WORK1, 0            ;(WORK1) -> (W)
            BCF     INTCON, T0IF        ;TMR0による再割り込みを可能とする.
            RETFIE                      ;割り込み処理ルーチンからメインプログラムへ復帰

START
            ;ポートBの設定
            BSF     STATUS,RP0          ;バンク1の選択
            MOVLW   B'00000000'
            MOVWF   TRISB               ;RB0-7を出力ポートに設定

            ;タイマ0の設定
            BCF     OPTION_REG, T0CS    ;システムクロック(FOSC)を選択．実際には　FOSC/4 = 4MHz/4 = 1MHz
            BCF     OPTION_REG, PSA     ;プリスケーラをタイマ0用に設定．（PSA = 1とすると，プリスケーラはWDT専用となる．）
            BSF     OPTION_REG, PS2     ;
            BSF     OPTION_REG, PS1     ;
            BSF     OPTION_REG, PS0     ;PS2 PS1 PS0 = 111 とすることでタイマ0のクロックをFOSC/4/256 と設定.

            ;割込みの設定
            BSF     INTCON, GIE         ;マスクされていない全ての割込みを可とする.
            BSF     INTCON, T0IE        ;タイマ0の割込みを可とする.
            BCF     INTCON, T0IF        ;タイマ0の割込みフラグをクリアする.

            BCF     STATUS,RP0          ;バンク0の選択
```

```
                ;メインルーチン
                STEP1           GOTO            STEP1           ;STEP1にジャンプすることを繰り返す永久ループ


                ;割り込み処理サブルーチン
Timer0_interrupt
                MOVLW           B'00000000'                     ;00000000 ->(W)
                                                                ;タイマ0はこの値を初期値としてカウントアップする.
                                                                ;（次の割り込みは11111111 -> 00000000 となるタイミング）
                MOVWF           TMR0                            ;(W) -> TMR0

                COMF            PORTB,1                         ;PORBの各ビットを1/0反転させる. これにより割り込みがかかるたびに
                                                                :PORTBの出力を反転し, LED1, 2, 3を点滅させる.

                RETURN

                END
```

タイマ0による割り込みプログラム(本章p.24, 25)の詳細説明(1)

```
        ORG     0               ;電源が入るとこの番地からプログラム実行を開始する.
        GOTO    START
```

```
START
        ;ポートBの設定
        BSF     STATUS,RP0      ;バンク1の選択
        MOVLW   B'00000000'
        MOVWF   TRISB           ;RB0-7を出力ポートに設定
```

タイマ0による割り込みプログラム(本章p.24, 25)の詳細説明(2)

```
;タイマ0の設定
        BCF     OPTION_REG, T0CS  ;システムクロック(FOSC)を選択.
                                  ;実際には  FOSC/4 = 4MHz/4 = 1MHz
        BCF     OPTION_REG, PSA   ;プリスケーラをタイマ0用に設定.
                                  ;(PSA = 1とすると, プリスケーラはWDT専用となる. )
        BSF     OPTION_REG, PS2    ; PS2 PS1 PS0 = 111 とすることで
        BSF     OPTION_REG, PS1    ;タイマ0のクロックをFOSC/4/256 と設定.
        BSF     OPTION_REG, PS0    ;
```

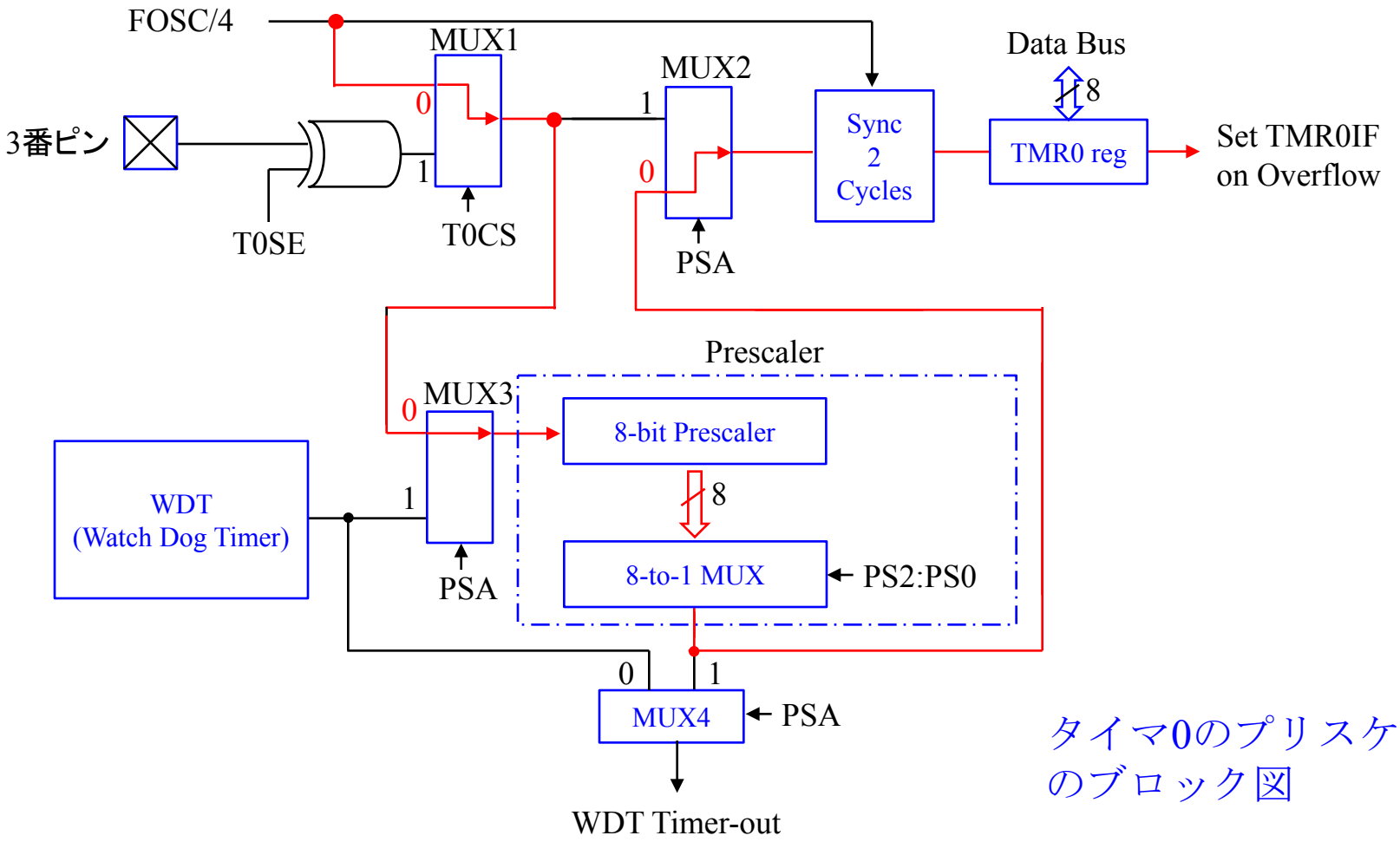
データシートによるとOPTION REGISTER (p.28) の各ビットによりタイマ0のクロックの選定, タイマ0のプリスケーラの設定ができる.
上のプログラムの設定によりタイマ0の入力クロックは, セラミック発振子に4 [MHz]のものを用いた場合, 4 [MHz]/4/256 = 3.906 [kHz]となる.

このクロックによりタイマ0をカウントアップして, タイマ0がオーバフロー(B'11111111' → B'00000000')するタイミングで 割り込みをかけることができる (ORG 4からのプログラムを実行する) . 割り込み処理プログラムの中でタイマ0の値を例えばB'00000000' と再設定すれば,
(B'100000000' - B'00000000')/3.906[kHz] = (256 – 0)/3.906[kHz] = 0.0655 [s]
後に再び割り込みがかけられる.

特殊レジスタ一覧

アドレス	名　称	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
バンク0									
00h	INDF	FSRの内容のアドレスのデータメモリ（物理的には存在しない）							
01h	TMR0	8ビットリアルタイム・クロック／カウンタ							
02h	PCL	プログラムカウンタ（PC）の下位8ビット							
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
04h	FSR	間接データメモリアドレスポインタ							
05h	PORTA	－	－	－	RA4/T0CKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07h		使用しない，「0」としてリードされる							
08h	EEDATA	EEDATAEEPROMデータレジスタ							
09h	EEADR	EEADREEPROMアドレスレジスタ							
0Ah	PCLATH	－	－	－	PCの上位5ビットへの書き込みバッファ				
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
バンク1									
80h	INDF	FSRの内容のアドレスのデータメモリ（物理的には存在しない）							
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
82h	PCL	プログラムカウンタ（PC）の下位8ビット							
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
84h	FSR	間接データメモリアドレスポインタ							
85h	TRISA	－	－	－	PORTAデータ入出力設定レジスタ				
86h	TRISB	PORTBデータ入出力設定レジスタ							
87h		使用しない，「0」としてリードされる							
88h	EECON1	－	－	－	EEIF	WRERR	WREN	WR	RD
89h	EECON2	EEPROM制御レジスタ2（物理的には存在しない）							
0Ah	PCLATH	－	－	－	PCの上位5ビットへの書き込みバッファ				
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

：バンク０，１で共通



タイマ0のプリスケーラのブロック図

MUX:マルチプレクサ(Multiplexer)，複数の入力のいずれかを選んで出力する，入力切り替え器
例えばMUX1はT0CS=0のとき入力のFOSC/4を出力する．T0CS=1のときはXORの値を出力する．
Prescaler: プリスケーラ，クロックの分周器．
上図の設定例ではタイマ0に入る前のクロックを$(1/2)^n$倍($n = 1 \sim 8$)する．nはPS2～PS0により設定できる．
WDT: ウォッチドッグタイマ(Watch Dog Timer)，コンフィギュレーションにて(__CONFIG _WDT_ON)とすると，WDTが起動する．設定時間内にプログラムがCLRWDT命令を実行して，WDTとそのWDT prescalerをクリア(0を代入)しないとプログラム実行が強制リセットされる．プログラムが暴走した場合などにWDTは有効．番犬タイマという意味．
Sync 2 Cycles: Data BusからTMR0 registerに値を書き込む際に，FOSC/4の2クロックの間，TMR0 registerの入力(MUX2の出力）は無視される．

タイマ0による割り込みプログラム(本章p.24, 25)の詳細説明(3)

```
;割込みの設定
        BSF     INTCON, GIE     ;マスクされていない全ての割込みを可とする.
        BSF     INTCON, T0IE    ;タイマ0の割込みを可とする.
        BCF     INTCON, T0IF    ;タイマ0の割込みフラグをクリアする.

        BCF     STATUS,RP0      ;バンク0の選択
```

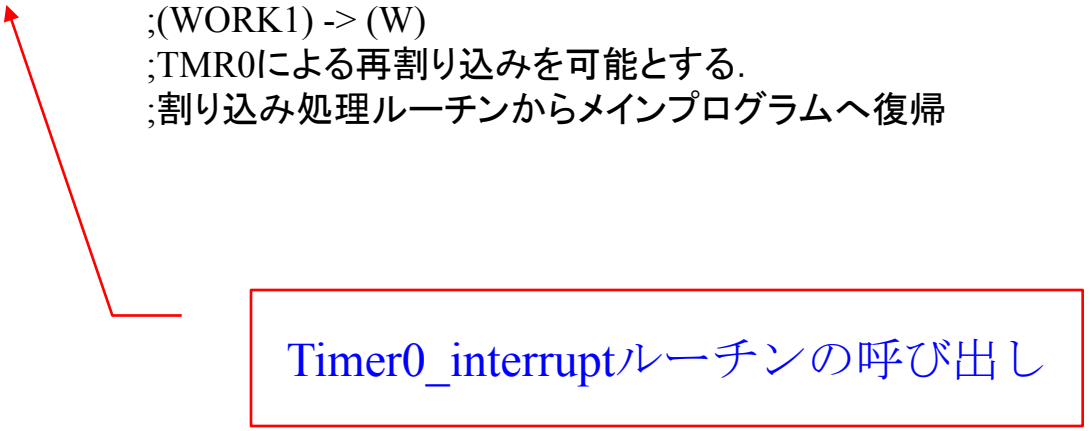
p.28よりPORTBはBank 0にある．割り込み処理ルーチンにてPORTBを利用するので，Bank 0を選択しておく．

データシートによるとINTCON Register (p.28 Bank0, 1に共通）の各ビットによりタイマ0の割り込みを設定できる．

上記のGIE, T0IEのビットを1にセットし，T0IFを0にクリアすることで，タイマ0がオーバフローしたときに，割り込みをかけることができる．なお，タイマ0がオーバフローした際にT0IFはセットされるので，割り込み処理プログラムの中で， T0IFを再びクリアしておく必要がある．これによりタイマ0による割り込みを再びかけることができる．

割り込み処理ルーチン

```
ORG     4               ;割り込みがかかるとこの番地からプログラム実行を開始する.
MOVWF   WORK1           ;wレジスタの内容をWORK1に退避させる
CALL    Timer0_interrupt
MOVF    WORK1, 0        ;(WORK1) -> (W)
BCF     INTCON, T0IF    ;TMR0による再割り込みを可能とする.
RETFIE                  ;割り込み処理ルーチンからメインプログラムへ復帰
```

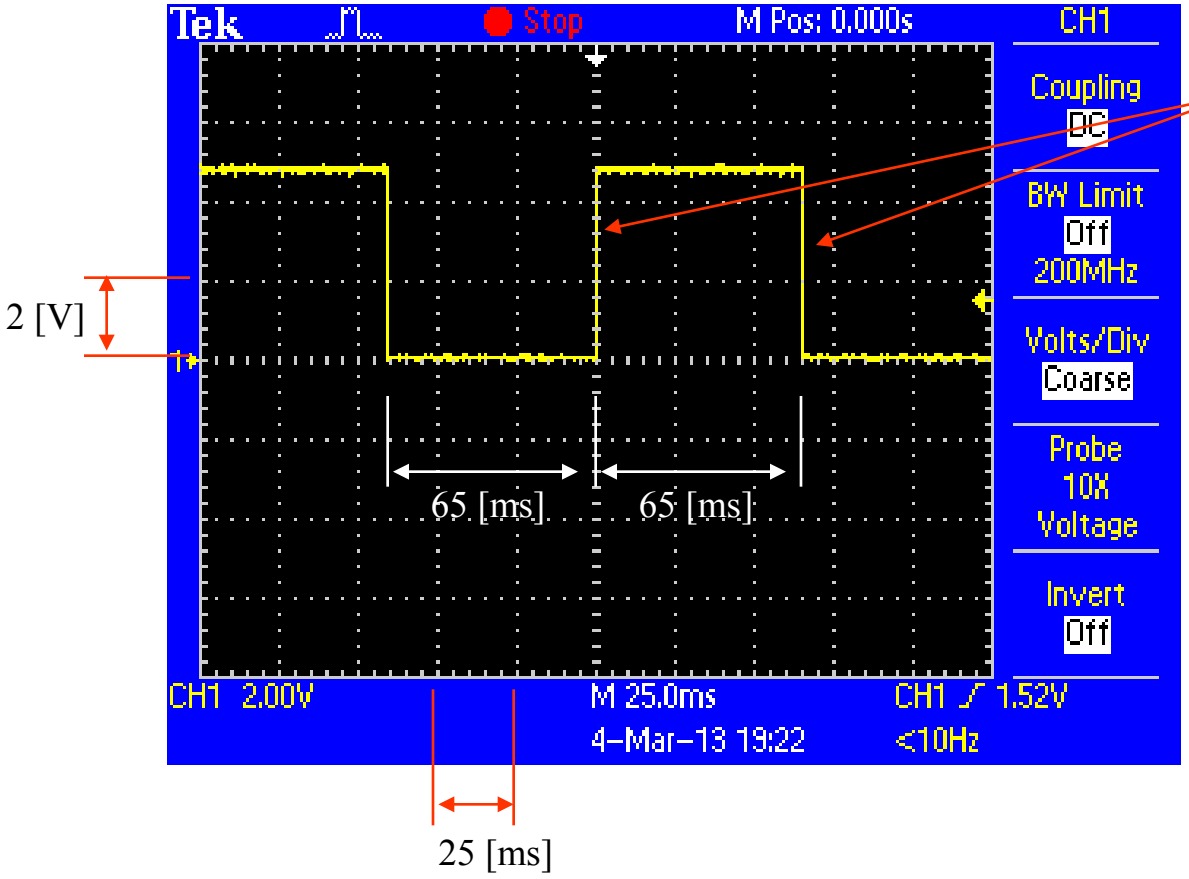


```
;割り込み処理サブルーチン
Timer0_interrupt
        MOVLW   B‘00000000‘     ;00000000 ->(W)
                                ;タイマ0はこの値を初期値としてカウントアップする.
                                ;（次の割り込みは11111111 -> 00000000 となるタイミング）
        MOVWF   TMR0            ;(W) -> TMR0

        COMF    PORTB,1         ;PORBの各ビットを1/0反転させる. これにより割り込みがかか
                                ;るたびにPORTBの出力を反転し, LED1, 2, 3を点滅させる.

        RETURN
```

タイマ0の初期値をB‘00000000’＝0に設定している．この値からカウントアップして，8ビットタイマーがオーバフロー (B‘11111111’→B‘00000000’) するタイミングで 割り込みをかけることができる．割り込み周期は

(B‘100000000’ - B‘00000000’)/3.906[kHz]
= (256 – 0)/3.906[kHz] = 0.0655 [ss]

となる．



COMF PORTB,1

65 [ms]毎に割り込みがかかり，出力電圧が5[V] ⇔ 0[V]で反転する．

タイマ0による割り込みプログラム
実行時の10番ピンの出力波形

2004年8月
2013年3月(9.2節)

著者： 古橋武
名古屋大学工学研究科計算理工学専攻
furuhashi@cse.nagoya-u.ac.jp